विश्वविद्यालय अनुदान आयोग
University Grants Commission
Quality higher education for all

आर्टिफिशियल इंटेलिजेंस का परिचय

अध्याय 1

आर्टिफिशियल इंटेलिजेंस का परिचय

1.1 परिचय: इलेक्ट्रॉनिक उपकरणों या मशीनों के विकास ने उनकी क्षमता को विभिन्न कार्यों को करने की आंतरवृद्धि की है। मानवों ने इलेक्ट्रॉनिक उपकरणों की क्षमता को उनके विभिन्न संचालन क्षेत्रों, उनके छोटे होते जाने और स्पीड में बढ़ते जाने के संदर्भ में विकसित किया है। पिछले दशक में, तार्किक निर्णय लेने की प्रक्रिया के लिए सूचना का समर्थन कर सके। सूचना का महत्व कुछ नहीं है जब तक वह प्रसंस्कृत नहीं होती और कुछ सूचना नहीं देती है। डेटा तेजी से बढ़ रहा है, और विभिन्न क्षेत्रों के लोगों को इस डेटा से ज्ञान की आवश्यकता होती है। इसलिए, हमें डेटा और ज्ञान के बीच एक सेतु बनानी होगी और इसे बेहतर ढंग से प्रसंस्कृत करके और समझकर ही प्राप्त किया जा सकता है। कंप्यूटर विज्ञान की शाखा आर्टिफिशियल इंटेलिजेंस, जो मानव से अधिक इंटेलिजेंस वाले कंप्यूटर का अध्ययन और निर्माण करने के लिए उपयोग की जाती है। आर्टिफिशियल इंटेलिजेंस कंप्यूटर विज्ञान में एक लोकप्रिय क्षेत्र बन रही है क्योंकि इसने कई क्षेत्रों में मानव जीवन की मानकों को बढ़ा दिया है। आर्टिफिशियल इंटेलिजेंस एक विज्ञान और प्रौद्योगिकी है जो कंप्यूटर विज्ञान, मनोविज्ञान, जीवविज्ञान (जेनोम, मानचित्रण, प्रोटीन बँधन) की शाखाओं पर आधारित है, इंजीनियरिंग (टर्बो-फैन डिजाइन, ऑटोमोबाइल डिजाइन) और ऊच्च-ऊर्जा भौतिकी (पारमाणविक-शस्त्र नकल)। AI का एक मुख्य ध्यान वहां है जहां मानव इंटेलिजेंस से जुड़े कंप्यूटर कार्यों के विकास में है, जैसे तर्क, सीखना, और समस्या का हल निकालना। पिछले 20 वर्षों में, आर्टिफिशियल इंटेलिजेंस ने विभिन्न उद्योगिक आधारित सिस्टमों के प्रदर्शन में बड़ी सुधार की है। आर्टिफिशियल इंटेलिजेंस एक तरीका है जिसमें एक इंटेलिजेंसमान आधारित रोबोट, एक संकेत प्रसंस्करण-नियंत्रित यंत्र, या कोई सॉफ्टवेयर हो सकता है जो इंटेलिजेंसमान इंसानों की तरह सोचता है। आर्टिफिशियल इंटेलिजेंस दो शब्दों आर्टिफिशियल और इंटेलिजेंस का मिश्रण है।

AI= ARTIFICIAL + INTELLIGENCE

सबसे पहले, 'आर्टिफिशियल' शब्द मानव के कार्य के परिणामस्वरूप बनाया गया है और 'इंटेलिजेंस' उच्च मानसिक प्रक्रियाओं से संबंधित है, जैसे कि रचनात्मकता, समस्याओं का पता लगाना, पैटर्न पहचान, वर्गीकरण, सीखना, प्रवेश, निष्कर्षण, उपमान निर्माण, अनुकरण, भाषा प्रसंस्करण, सूचना और कई अन्य ज्ञान। इंटेलिजेंस लक्ष्यों को प्राप्त करने की क्षमता की कार्यप्रणाली है।

आर्टिफिशियल इंटेलिजेंस के दृष्टिकोण को चार श्रेणियों में विभाजित किया जाता है:

इंसानों की तरह सोचो	तर्कसंगत सोचो
इंसानों की तरह व्यवहार करें	तर्कसंगत व्यवहार करें

संक्षेप में, हम मानव इंटेलिजेंस और आर्टिफीसियल इंटेलिजेंस के बीच अंतर करने में सक्षम हैं जैसा कि तालिका में दिखाया गया है:

मानव इंटेलिजेंस	आर्टिफीसियल इंटेलिजेंस
प्राकृतिक या दैवीय इंटेलिजेंस	मानव खुफिया द्वारा प्रोग्राम किया गया
विशेषज्ञता के साथ बढ़ता है और संयुक्त रूप से वंशानुगत होता है।	वंशानुगत के रूप में कुछ भी नहीं है, हालांकि सिस्टम विशेषज्ञता से सीखते हैं;
आउटपुट प्राप्त करने के लिए अत्यधिक परिष्कृत और बाहर से कोई बिजली की आवश्यकता नहीं है। बल्कि इंटेलिजेंस के लिए डेटा अच्छा है।	यह कंप्यूटर विज्ञान आधारित प्रणाली में है और हम चाहते हैं कि वर्तमान आउटपुट का आग्रह करे। आउटपुट प्राप्त करने के लिए मानसिक वस्तु की आवश्यकता होती है।
कम सामाजिक कौशल	उत्कृष्ट सामाजिक कौशल
नवीनता	अनुकूलन
कम सटीकता	उच्च सटीकता
धीमी प्रक्रिया	तेज प्रक्रिया
25 वाट का उपयोग करता है	2 वाट का उपयोग करता है

तालिका 1.1 मानव इंटेलिजेंस और आर्टिफीसियल इंटेलिजेंस

1.1.1 AI के प्रकार: "इंटेलिजेंस" शब्द के संबंध में उलझन, इसकी गलत परिभाषा और बहुत व्यापक क्षेत्र ने लोगों को AI को दो श्रेणियों में विभाजित करने पर मजबूर किया है:

(i) मजबूत AI (Strong AI)

(ii) कमजोर AI (Weak AI)

• **मजबूत AI**: यह दृढ़ दावा करता है कि एक कंप्यूटर मानवता की संख्या पर सोचने की क्षमता के साथ बनाया जाता है। मजबूत AI शोध कंप्यूटर के बारे में होती है, जिसके द्वारा वास्तविकता से तर्क किया जा सकता है और समस्याओं का हल निकाला जा सकता है। मजबूत AI की परामर्श करने वाले लोग यह मानते हैं कि यह अंततः कंप्यूटर को उन मानवों की इंटेलिजेंस से बहुत अधिक बना देगा।

• **कमजोर AI:** सिर्फ यह कहता है कि कुछ "चिंतन-जीव" फीचर्स कंप्यूटरों में जोड़े जाते हैं ताकि उन्हें एक और सहायक औज़ार के रूप में सुगमता मिले। कमजोर AI शोध कंप्यूटर के बारे में होती है, जिसके द्वारा सीमित क्षेत्र में तर्क किया और समस्याओं का हल निकाला जा सकता है। इसलिए, ऐसी मशीन का व्यवहार कुछ ऐसा होगा जिसे ऐसा लगेगा कि वह इंटेलिजेंसमान है, लेकिन वास्तविक इंटेलिजेंस नहीं होगी। उदाहरण के रूप में, "मदद" को कमजोर AI कहा जाता है और "करना" को अक्सर मजबूत AI कहा जाता है।

1.2 AI की परिभाषाएँ

• "मशीनों को बनाने की कला जो कार्य करती हैं जो मानवों द्वारा किए जाने पर इंटेलिजेंसमानता की आवश्यकता होती है।" (कुर्जवेल, 1990)

• "जो कंप्यूटर विज्ञान की शाखा है और इंटेलिजेंसमान व्यवहार के स्वचालन के साथ संबंधित है।" (लुगर और स्टबलफील्ड, 1993)

• "हम मानव सोच से जुड़ी गतिविधियों की स्वचालन करते हैं, जैसे निर्णय लेना, समस्या का हल ढूंढ़ना, सीखना," (बेलमैन, 1978)

• "AI आमतौर पर मानवों की तरह इंटेलिजेंसमानता की आवश्यकता रखने वाली चीज़ों को करने के लिए कंप्यूटर को बनाने का विज्ञान है।" (मिंस्की द्वारा परिभाषित)

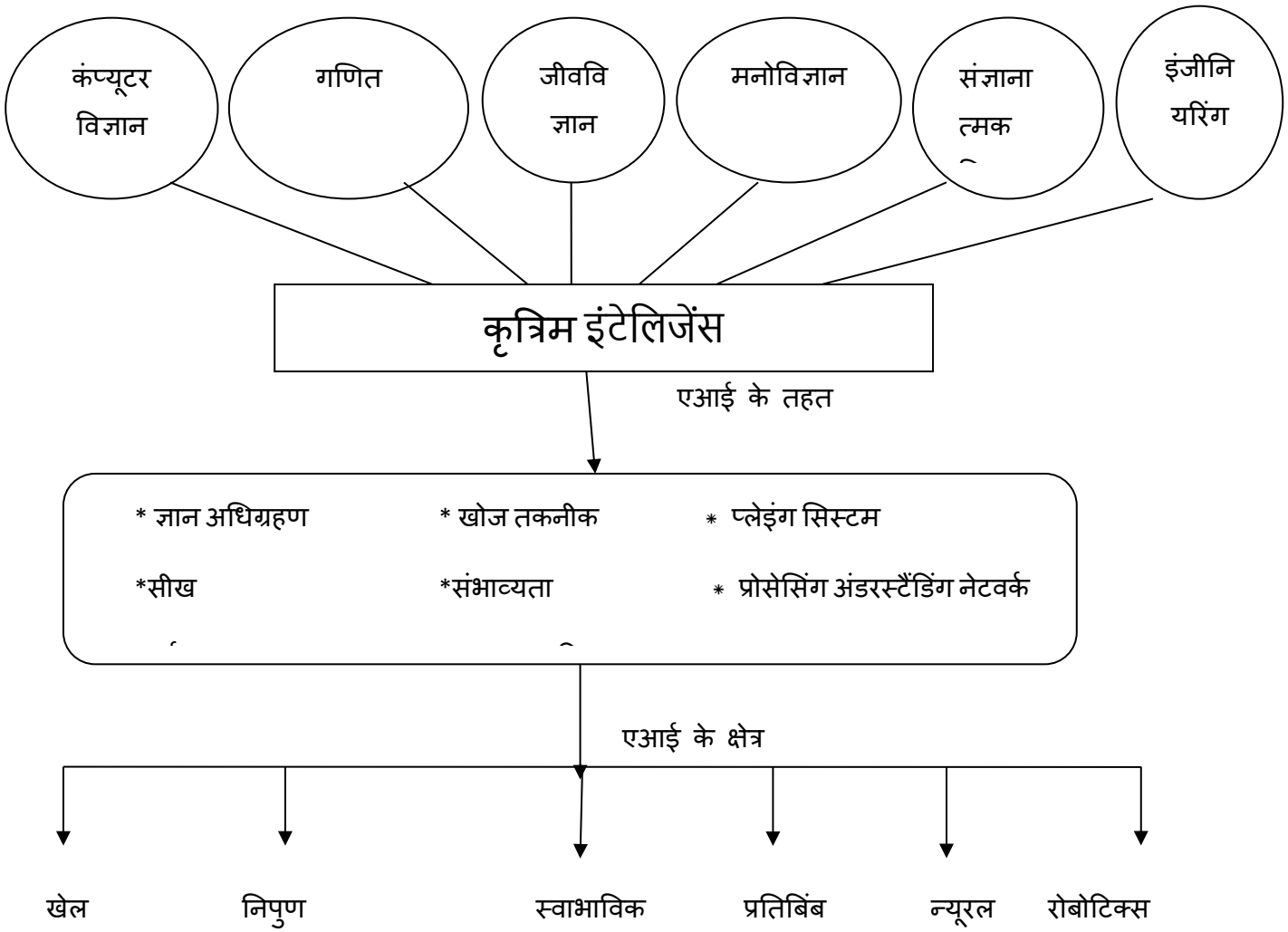
• "AI वह अध्ययन है जिसका मकसद यह है कि कंप्यूटर को वे काम करवाया जा सके जो फिलहाल मानवों को बेहतर से करना होता है।" (रिच और नाइट, 1991)

• "AI कंप्यूटर विज्ञान का एक हिस्सा है जो मनुष्य के व्यवहार में इंटेलिजेंसमानता से जुड़ी विशेषताओं का प्रदर्शन करने वाले इंटेलिजेंसमान कंप्यूटर सिस्टम डिज़ाइन के साथ संबंधित है।" (बार और एडवर्ड, 1981 - 82)

• "कंप्यूटेशनल मॉडल के उपयोग से मानसिक क्षमताओं का अध्ययन।" (चर्नियाक + मैकडर्मोट, 1985)

1.3 AI की शाखाएँ: AI की शाखा को कंप्यूटर विज्ञान, गणित और आँकड़े, जीवविज्ञान, मनोविज्ञान, संज्ञानशास्त्र और इंजीनियरिंग से एक महत्वपूर्ण ज्ञान की अनेक शाखाओं ने समृद्ध किया है। एक नज़र में चित्र 1.1 भी AI के विषयों और इससे संबंधित क्षेत्रों की विषयक्रम को दर्शाता है।

- **एआई की शाखाएं**



चित्र 1.1: एआई की शाखाएं और अनुप्रयोग क्षेत्र

* **सीखने की प्रणाली:** प्रणाली में विशिष्ट लक्षणों के संगठन को एक साधारण लक्ष्य प्राप्त करने के लिए जोड़ा गया होता है और सीखना कार्यक्रम वह चीजें हैं जिन्हें हमें फैक्ट या व्यवहार के अध्ययन से प्राप्त करना होगा। सीखने का अर्थ है कि प्रणाली के अंदर परिवर्तन होते हैं जो प्रणाली को समायोजनशील बनाते हैं, अर्थात इससे प्रणाली अगली बार एक ही कार्य(ओं) को अधिक दक्षता से कर

सकती है। सीखने के लिए कुछ नियम तर्क में व्यक्त किए जाते हैं। कार्यक्रम सिर्फ वही बातें सीख सकते हैं जिनके तथ्य या व्यवहार को उनके फॉर्मलिज़्म को प्रदर्शित कर सकते हैं, और दुर्भाग्य से सीखने की प्रणाली बहुत ही सीमित क्षमताओं पर आधारित होती है जो ज्ञान को प्रदर्शित करने की क्षमता को देखते हैं। उदाहरण के लिए, एक भारी इंटरसेक्शन में यातायात पैटर्न, हम उसे एक सीखने की प्रणाली के माध्यम से चला सकते हैं जिसमें पिछले यातायात पैटर्न के बारे में ज्ञान (अनुभव) होता है, और यदि यह सफलतापूर्वक "सीखता" है, तो यह भविष्य के यातायात पैटर्न को बेहतरीन तरीके से पूर्वानुमान करेगा।

* **नॉलेज सब्स्टिट्यूशन एंड रीजनिंग:** यह AI का क्षेत्र है जो मानव स्तर से सबसे दूर है, यद्यपि यह 1950 के दशक से एक सक्रिय अनुसंधान क्षेत्र रहा है। हालांकि, गैर-अवादानी तर्कशास्त्र और कार्य के सिद्धांतों की विकास में व्यापक प्रगति हुई है, फिर भी नई विचारों की आवश्यकता है। कुछ तथ्यों से भी इन्फर किए जा सकते हैं। कुछ कार्यों के लिए गणितीय तर्क पर्याप्त है, लेकिन 1970 के दशक से तर्क के गैर-अवादानी तर्क के नए तरीके तर्क में शामिल किए गए हैं। गैर-अवादानी तर्क का सबसे अच्छा प्रकार डिफ़ॉल्ट तर्क है, जिसमें एक नतीजा डिफ़ॉल्ट रूप से अभिप्रेत होना है। हालांकि, यदि विपरीत प्रमाण हो तो नतीजा वापस लिया जा सकता है। उदाहरण के रूप में, हम पक्षी के बारे में सुनते हैं, तो हम यह अनुमान लगाते हैं कि यह उड़ सकता है, लेकिन जब हमें यह पता चलता है कि यह एक पेंग्विन है, तो यह नतीजा उलट सकता है। यह नतीजा वापस लेने की संभावना है जो तर्क की गैर-अवादानी चरित्रिता का निरूपण करती है। पारंपरिक तर्कशास्त्र एक-विधि है, जिसमें पर्याप्तता के दृष्टिकोण से निर्धारित परिणाम सेट पर्याप्तता वाले परिणामों का उत्पादन कर सकता है, अर्थात परिणामों के प्रेमी की वृद्धि आधारित बद्धता। प्रतिबंध एक और गैर-अवादानी तर्क का प्रकार है।

* **योजना:** योजना में सामान्य तथ्यों के बारे में जानकारी, वास्तविक स्थिति के बारे में जानकारी और एक लक्ष्य का एक बयान शुरू होता है। इनसे, योजना कार्यक्रम लक्ष्य को प्राप्त करने के लिए एक रणनीति उत्पन्न करते हैं। सबसे सामान्य मामलों में, रणनीति केवल एक क्रियाओं की एक क्रमशः होती है। दूसरे शब्दों में कहा जा सकता है, योजना विषयों के तरह की चीजों का अध्ययन है जो मौजूद होती हैं। AI में कार्यक्रम और वाक्यों में विभिन्न प्रकार के वस्तुओं का सम्प्रदाय किया जाता है और हम यह अध्ययन करते हैं कि ये प्रकार क्या हैं और उनकी मूलभूत गुणधर्म क्या हैं।

* **सर्च समस्याएँ:** सर्च समस्याएँ जिन्हें हम अक्सर इंजीनियरिंग में देखते हैं, वे निश्चिततापूर्ण होती हैं, अर्थात सर्च स्थान के तत्वों के दौरे की यात्रा के आदेश को जाना जाता है। हालांकि, सर्च समस्याएँ, जिन्हें हम AI में देखेंगे, अनिश्चित होती हैं और सर्च स्थान के तत्वों में यात्रा के क्रम पूरी तरह से ज्ञान सेट पर निर्भर करती है। Artificial Intelligence कार्यक्रम आमतौर पर बड़ी संख्या में संभावनाओं का परीक्षण करते हैं - उदाहरण के लिए, शतरंज में चालें और एक स्थानांतरण के द्वारा प्रमाण के द्वारा निष्कर्षण। विभिन्न क्षेत्रों में इसे अधिक सक्रिय ढंग से करने के बारे में अक्सर खोजें की जाती हैं। सर्च एल्गोरिदम की विस्तृत चर्चा बाद में की जाएगी।

* **लॉजिक प्रोग्रामिंग:** सदीयों से अधिक समय के लिए, गणितज्ञों और तर्कशास्त्री लॉजिकल बयानों को प्रतीकात्मक ऑपरेटरों द्वारा दर्शाने के लिए विभिन्न उपकरण बना रहे थे। ऐसे प्रयासों की एक उत्पादन

विज्ञान तर्क है, जो गणितीय ऑपरेटरों द्वारा जुड़ी बाइनरी बयान (प्रस्तावनाएं) का समूह है। प्रस्तावों की तर्क की तर्क विधि में सरल संख्या शामिल है, जो वास्तविक दुनिया के अधिक जटिल स्थितियों को हैंडल करने के लिए धीमी धीमी गई। एक प्रकार की प्रतिष्ठा की बात की जा सकती है, जिसमें डिफ़ॉल्ट तर्क द्वारा एक निष्कर्ष अवधारित करना होता है।

* **सॉफ्ट कंप्यूटिंग:** सॉफ्ट कंप्यूटिंग एक कंप्यूटिंग उपकरणों और तकनीकों का संग्रह है, जिन्हें संख्यात जुड़ी विज्ञानों में साझा किया जाता है, जिनमें समयोज्यता तार्किक, आर्टिफीसियल स्नायु नेटवर्क, जीनेटिक एल्गोरिदम, विश्वास गणना और अनुक्रमणिकीय तार्किक के नामकरण जैसे कुछ पहलुओं को शामिल किया गया है। इन उपकरणों का अलग-अलग और साथ-साथ उपयोग किया जाता है जो अनुप्रयोग के क्षेत्र पर निर्भर करता है। पहले तीन उपकरणों की विस्तारित सीमा निम्नलिखित रूप से व्यक्त की गई है।

* **फजी लॉजिक:** फजी लॉजिक वास्तविक दुनिया की मानव जैसी तर्कशास्त्रीय समस्याओं का मॉडल बनाने के लिए फजी सेट्स और तार्किक संयोजकों के साथ संघटित होता है। एक फजी सेट, सामान्य सेटों के विपरीत होता है, जिसमें डोमेन के यूनिवर्सल सेट के सभी तत्व शामिल होते हैं, लेकिन [0,1] अवधारित सदस्यता मानों के साथ। यह ध्यान देना चाहिए कि एक सामान्य सेट अपने सदस्यों को सदस्यता के मूल्य के समकक्ष एक वार्ता में शामिल करता है, क्योंकि वे शून्य सदस्यता होते हैं। फजी सेटों पर लागू किए जाने वाले सबसे सामान्य ऑपरेटर एंड (न्यूनतम), और (अधिकतम) और प्रतिस्थापन (पूरकता) हैं, जहां एंड और और बाइनरी आपत्ति होती है, जबकि प्रतिस्थापन केवल एक आपत्ति होती है। फजी सेटों की तार्किक तर्कशास्त्र में अवधारित की गई थी, जिसने सिस्टम के लिए प्रस्तावनात्मक तार्क में इसे प्रस्तावित किया, और बाद में इसे व्यापक सिस्टमों में लागू विचारशीलता के लिए बढ़ाया गया।

* **आर्टिफीसियल न्यूरल नेटवर्क:** आर्टिफीसियल न्यूरल नेटवर्क (ANN) जैविक स्नायु जाल की विद्युत आनुलोमिकों के समानांतर हैं। जैविक नस्यों को न्यूरॉन कहा जाता है, जो आस-पासी न्यूरॉनों या संवेदकों से प्राप्त संकेतों को डेंड्राइट्स के माध्यम से प्राप्त करते हैं, यहां तक कि तारीक तत्व में प्राप्त विद्युतीय इम्पल्स को प्रसंस्करण करते हैं और एक मोटी और मोटी फाइबर, जिसे नस्य फाइबर कहा जाता है, के माध्यम से संकेतों को प्रेषित करते हैं। एक विधुतीय नस्य कोशिका का विद्युतीय मॉडल एक सीधी रेखा के माध्यम से गणना किया जा सकता है, जो वेटेड प्रवर्धन की कुल मानकों का होगा, जबकि गैर-लीनियर विकुलनीयता संकेतों के कुल स्तरों की प्रतिरोध करने की कोशिश करेगी। एक आर्टिफीसियल स्नायु नेटवर्क का सबसे सामान्य उपयोग मशीन लर्निंग में होता है। एक लर्निंग समस्या में, आर्टिफीसियल स्नायु नेटवर्क में वजन और / या गैर-लीनियरिटी एक समायोजन चक्र के माध्यम से होती है। विभिन्न प्रकार के मशीन लर्निंग में सफलतापूर्वक लागू किए गए निर्देशित लर्निंग एल्गोरिदम के साथ आर्टिफीसियल स्नायु नेटवर्क हैं। उपन्यास लर्निंग एल्गोरिदम से डिज़ाइन किए गए अनुप्रयोगों में, उपयोगकर्ता को निर्देशित बनाने, स्वचालितीकरण, रोबोटिक्स और कंप्यूटर दृष्टिकोण में ज्ञान प्राप्ति के लिए आवेदित किया जाता है। प्रत्युपदेशित लर्निंग एल्गोरिदम के साथ बनाए गए आर्टिफीसियल स्नायु नेटवर्क को योग्यता, डेटा प्राप्ति और ज्ञान का एनालॉग सूचना में अनुक्रमणिकीय रूप में अनुप्रयोग किया जाता है।

* **जीनेटिक एल्गोरिदम:** जीनेटिक एल्गोरिदम (GA) एक यादृच्छिक एल्गोरिदम है जो जैविक विकास की गतिविधि की अनुकरण करता है। यह जाति की गतिविधि के प्रक्रिया में "उत्कृष्टता के अवशेष" के सिद्धांत पर आधारित है। जीनेटिक एल्गोरिदम इंटेलिजेंस सर्च, मशीन लर्निंग और अनुक्रमणिकी समस्याओं में व्यापक उपयोग की पात्र होते हैं। एक जीनेटिक एल्गोरिदम में मुद्दों को क्रोमोसोम द्वारा दर्शाया जाता है, जो आमतौर पर बाइनरी स्ट्रिंग के द्वारा प्रतिष्ठित होते हैं। जीनेटिक एल्गोरिदम में प्रयुक्त सबसे सामान्य ऑपरेटर्स क्रॉसओवर और म्युटेशन होते हैं। एक जीनेटिक एल्गोरिदम में विकास चक्र में निम्नलिखित 3 क्रमशः चरण होते हैं। a) प्रजनन की पीढ़ी (मुद्दे द्वारा प्रतिष्ठित किए गए क्रोमोसोमों को जन्म देते हैं)। b) क्रॉसओवर और म्युटेशन के माध्यम से जीनेटिक विकास। c) उत्कृष्ट उम्मीदवार राजनीतिक प्रजनन से उपजे मुद्दों में से चयन करें। उपरोक्त चरणों को निर्दिष्ट संख्या में बार बार दोहराया जाता है तक जवाब प्राप्त करने के लिए। जीनेटिक प्रोग्रामिंग एक समस्या के एक उच्च-स्तरीय संकेत के साथ कार्यरत की गई एक कार्यक्षम रूप से है। जीनेटिक प्रोग्रामिंग एक समस्या को हल करने के लिए स्वचालित रूप से एक कार्यरत बनाता है।

1.4 एआई तकनीकों के अनुप्रयोग:

एक्सपर्ट सिस्टम: एक एक्सपर्ट सिस्टम, जिसे आपूर्ति श्रृंखला की एक शाखा भी कहा जाता है, विभिन्न प्रकार के डेटा, संपत्ति और संसाधनों के प्रबंधन को सुनिश्चित करने के लिए AI तकनीकों का उपयोग करती है। यह प्रणाली संगठनों को डेटा की व्याख्या करने, आवश्यकताओं को आकलन करने, आपूर्ति लगत को कम करने, ग्राहकों की मांग को पूरा करने और विनिर्माण और वितरण की प्रक्रियाओं को समर्थन करने में मदद करती है। उदाहरण के रूप में, एक आपूर्ति प्रणाली के माध्यम से स्टॉक मैनेजमेंट, विपणन योजना, भविष्य की मांग का अनुमान लगाना और आपूर्ति श्रृंखला की अवधारणा करना संभव होता है।

नतीजे के रूप में, आर्टिफिशियल इंटेलिजेंस (AI) के प्रमुख विभागों में से कुछ वर्गीकृत क्षेत्रों का उल्लेख किया गया है। यहां उदाहरण दिए गए हैं, लेकिन यह अंतिम नहीं हैं। AI का विस्तारित उपयोग बहुत सी अन्य क्षेत्रों में भी होता है और नवीनतम तकनीकी विकास के साथ उनकी संभावनाओं का गहरा अध्ययन जारी है। AI ने मानवीय संदर्भों में रोचकता, सुरक्षा, स्वास्थ्य देखभाल, संचार, वाणिज्यिकी, संगठन, विज्ञान और और भी बहुत कुछ में महत्वपूर्ण योगदान दिया है।

Expert System = Knowledge + Approaches of problem-solving.

एक नॉलेज बेस जो डोमेन-विशिष्ट जानकारी को कैप्चर करता है जो एक अनुमान से संबंधित होता है। इंजन इसमें सिस्टम को दी गई उलझन को हल करने के लिए नॉलेज बेस के भीतर वर्णित जानकारी में हेरफेर करने के लिए एल्गोरिदम शामिल हैं।

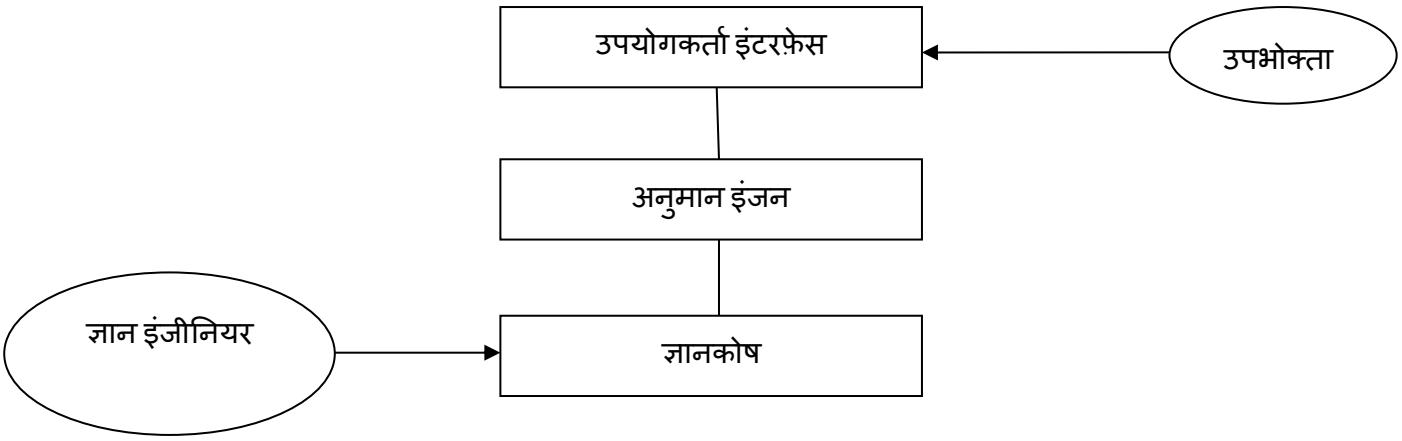
विशेषज्ञ प्रणालियों की मुख्य विशेषताएं

- उच्च प्रदर्शन
- विश्वसनीयता
- स्थिरता के तहत

- उच्च स्तरीय प्रतिक्रियाएं

विशेषज्ञ प्रणालियों के मुख्य घटक

- उपयोगकर्ता इंटरफ़ेस
- अनुमान इंजन
- नॉलेज-बेस

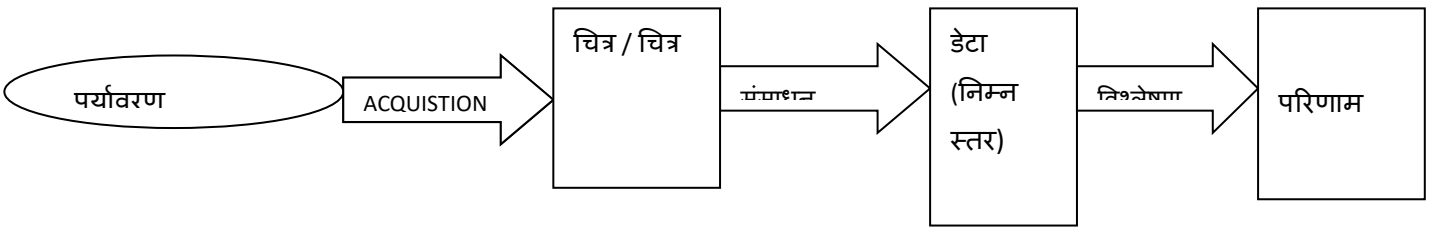


चित्र 1.2: विशेषज्ञ प्रणालियों के घटक

2. कंप्यूटर विजन: कंप्यूटर विजन आर्टिफिशियल इंटेलिजेंस और प्रौद्योगिकी का एक क्षेत्र हो सकता है जिसका उद्देश्य कंप्यूटर को दुनिया की एक दृश्य मान समझ देना है। कंप्यूटर विजन का लक्ष्य तीन मुख्य प्रक्रिया भागों के माध्यम से मानव दृष्टि पीड़ित डिजिटल चित्रों का अनुकरण करना है:

1. छवि अधिग्रहण (Image acquisition)
2. छवि प्रक्रिया (Image process)
3. छवि विश्लेषण और समझ (Image analysis and understanding)

जैसा कि दुनिया की हमारी मानवीय दृश्य समझ हमारे द्वारा देखे जाने वाले कार्यों के माध्यम से चयन करने की हमारी क्षमता में प्रतिबिंबित होती है, कंप्यूटर को इस तरह की दृश्यमान समझ प्रदान करने से उन्हें समान शक्ति मिलेगी:



चित्र 1.3: कंप्यूटर दृष्टि का प्रवाह

1. छवि अधिग्रहण:

छवि अधिग्रहण यह है कि हमारे चारों ओर एनालॉग दुनिया को शून्य और डिजिटल चित्रों से बने द्विआधारी ज्ञान में अनुवाद करने की विधि है।

ऐसे डेटासेट बनाने के लिए विभिन्न उपकरण बनाए जाते हैं:

1. वेबकैम और सिग्नल सेंसर आधारित कैमरे
2. डिजिटल कैमरा
3. DSLR

2. छवि प्रसंस्करण:

कंप्यूटर विजन का दूसरा हिस्सा चित्रों की निम्न-स्तरीय प्रक्रिया है। एल्गोरिदम वर्ग।

छवि के घटकों पर निम्न स्तर की जानकारी का अनुमान लगाने के लिए प्रारंभ के भीतर प्राप्त बाइनरी डेटा पर लागू माप।

निम्न-स्तरीय छवि प्रक्रिया एल्गोरिदम में शामिल हैं:

1. छवि वृद्धि
2. एज डिटेक्शन
3. छवि वर्गीकरण
4. छवि सुविधा मिलान

3. छवि विश्लेषण और समझ

कंप्यूटर विजन पाइपलाइन का अंतिम चरण यदि जानकारी का विशेष विश्लेषण है, जो विकल्प बनाने की अनुमति दे सकता है। उच्च-स्तरीय एल्गोरिदम वर्ग माप लागू करते हैं, प्रत्येक छवि ज्ञान और पिछले चरणों में गणना की गई निम्न-स्तरीय जानकारी को भी प्रभावित करते हैं।

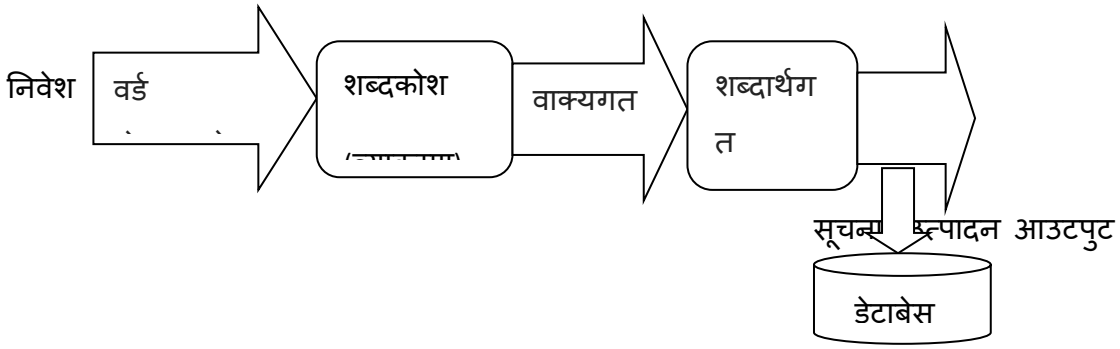
उच्च-स्तरीय छवि विश्लेषण के उदाहरण हैं:

1. 3D ऑब्जेक्ट्स
2. चेहरे की पहचान

3. रोबोटिक्स: रोबोटिक्स एक इंजीनियरिंग की शाखा है जो रोबोट्स के निर्माण और उनके संचालन और प्रसंस्करण के लिए कंप्यूटर प्रणाली का उपयोग करती है। रोबोट उद्योगों में उपयोग होते हैं ताकि उत्पादन प्रक्रिया को तेजी से बढ़ावा मिल सके। वे स्वास्थ्य देखभाल, परमाणु विज्ञान, डेटा संचार, विभिन्न प्रकार की सेवाएं, फार्मेसी, सुरक्षा उपकरण, सॉफ़्टवेयर इंजीनियरिंग, डेटा माइनिंग और वेयरहाउसिंग आदि के क्षेत्र में उपयोग होते हैं। आर्टिफीसियल इंटेलिजेंस के लिए कंप्यूटर-आधारित एकीकृत इलेक्ट्रॉनिक्स और संचार इंजीनियरिंग, उत्पादन

इंजीनियरिंग, जैविक मैकेनिक्स, डेटा माइनिंग इंजीनियरिंग का इस्तेमाल किया जाता है। स्वचालन और आर्टिफिशियल इंटेलिजेंस इंजीनियरिंग माल-सेवा और उत्पादों के निर्माण में मानव के अवश्यकता को कम करने के लिए नियंत्रण प्रणाली और सॉफ़्टवेयर इंजीनियरिंग का उपयोग करती है।

4. **नेचुरल लैंग्वेज अंडरस्टैंडिंग:** नेचुरल लैंग्वेज प्रोसेसिंग प्रणालियाँ वाक्यों की स्ट्रिंग (वाक्य) को अपनी इनपुट के रूप में लेती हैं और इनपुट के अर्थ को पकड़ती हैं। इसका आउटपुट बहुत अधिक मायने रखता है टास्क पर। एक डेटाबेस के रूप में काम करने वाली प्राकृतिक भाषा समझ प्रणाली इंग्लिश में प्रश्नों को स्वीकार कर सकती है जो डेटाबेस द्वारा प्रायोगिक हो जाते हैं। इस मामले में इनपुट की रूपांतरण (सिस्टम का आउटपुट) SQL प्रश्नों के संरचित प्रस्ताव में किया जा सकता है, जिन्हें सीधे डेटाबेस में प्रस्तुत किया जा सकता है। नेचुरल लैंग्वेज प्रोसेसिंग के चार विभिन्न चरणों को सरल रूप से दिखाया जाता है।



चित्र 1.4: एनएलपी का एक सरलीकृत दृश्य

5. **मशीन लर्निंग:** मशीन लर्निंग, कंप्यूटर और डेटा सिस्टम को अनुभव के साथ सहज करने और सुधार करने की अध्ययन है। यह एक AI का उपसंग है और डेटा माइनिंग, जानकारी सिद्धांत और संभावना संयोजन के साथ परस्पर्तिक्ष करता है। मशीन लर्निंग के प्रकार:

   प्रशिक्षणसंबंधी लर्निंग (शिक्षक-उन्मुख शिक्षा): प्रशिक्षित (लेबलयुक्त) प्रशिक्षण डेटा से किसी समाधान की कार्यकारी फ़ंक्शन का निष्पादन करने का मशीन लर्निंग का कार्य है।

   अशिक्षित लर्निंग: अशिक्षित लर्निंग अध्ययन करता है कि सिस्टम कैसे सीख सकते हैं कि किसी ऐसे प्रविष्टि पैटर्न को प्रतिष्ठित करें जो समग्र प्रविष्टि पैटर्न की सांख्यिकीय संरचना को प्रतिबिंबित करता है।

   रिएंफोर्समेंट लर्निंग: पुनर्प्रोत्साहन लर्निंग मशीन लर्निंग का एक उपक्षेत्र है; हालांकि, यह मशीन नियंत्रित निर्णय और AI के लिए सामान्य उद्देश्य सम्मिश्रण भी है।

पुनर्प्रोत्साहन लर्निंग, मशीन लर्निंग में एक अध्ययन मॉडल के रूप में ज्ञात होता है, जो एक एजेंट को कार्रवाई लेने और पर्यावरण के साथ एक्स करने की अनुमति देने के लिए एक तरीका है ताकि कुल प्रतिफल को अधिकतम किया जा सके। RL आमतौर पर एक मार्कोव निर्णय प्रक्रिया (MDP) के रूप में मॉडल किया जाता है।

6. **डेटा विज्ञान:** डेटा विज्ञान उन्हें सिस्टम के रूप में सोचने का एक तरीका है जो विश्वव्यापी और व्यापक है, विशेष समस्याओं पर केंद्रित नहीं है। डेटा विज्ञान एक विभिन्न उपकरण, एल्गोरिदम और मशीन लर्निंग के सिद्धांतों का संग्रह है जिसका लक्ष्य डेटा से छिपे हुए पैटर्न सर्चना है। डेटा साइंस संरचित और असंरचित दोनों प्रकार के डेटा का उपयोग करता है। डेटा विज्ञान के चरण:
   व्यापार समझ
   डेटा समझ
   डेटा तैयारी
   डेटा मॉडलिंग
   मूल्यांकन
   डिप्लॉयमेंट
   आज डेटा साइंस क्यों इतना प्रसिद्ध है?
   डेटा हर जगह है
   उपकरण मजबूत और सस्ते हो गए हैं: पिछले दशक में मुद्रित संग्रह की कीमत छः गुना से अधिक कम हो गई है और जीपीयू बारह सौ गुना अधिक क्षम हो गए हैं।
   सॉफ्टवेयर और हार्डवेयर का जनतंत्रीकरण: Tensorflow और Pytorch जैसे लोकप्रिय ओपन-स्रोत के फ़्रेमवर्क आसान इंटरफ़ेस प्रदान करते हैं और हार्डवेयर पर संकलन और अनुकूलन जैसी जटिलताओं को छिपाने के साथ-साथ प्रकाशित करते हैं।

7. **आर्टिफीसियल न्यूरल नेटवर्क:** आर्टिफीसियल न्यूरल नेटवर्क (ANN) जैविक सरंचना नेट के विद्युत तुलनात्मक रूप हैं। जैविक नस्य शिशु, जिन्हें न्यूरॉन कहा जाता है, पड़ोसी न्यूरॉनों या रिसीवर के माध्यम से इलेक्ट्रिक पल्स प्राप्त करते हैं, इसे सेल बॉडी में प्राप्त इलेक्ट्रिक पल्स को प्रसंस्करण करते हैं और एक बड़ी और मोटी रेशा के माध्यम से इलेक्ट्रिक पल्स प्रसारित करते हैं, जिसे न्यूरॉन फाइबर कहा जाता है। एक न्यूरॉन के विद्युतीय मॉडल में एक रैखिक पदार्थ, जिसके बाद एक गैर-रैखिक रोकने वाला कार्य का सामरिक रूप है, होता है। रैखिक सक्रियता कार्य वज़नित इनपुट उत्तेजना की कुल कोशिका में योगदान की जांच करता है, जबकि गैर-रैखिक रोकने वाला कार्य प्रतिष्ठान की संख्या के साथ शब्द सीमाओं के बीच भेदभाव करने में मदद करता है।

अक्सर, छिपे हुए मार्कव मॉडल अधिकतम तारीख़ी रूप में भी प्रायोजकीय संरचना को मान्यता देने के लिए उपयोग किए जाते हैं।

8. **लैंग्वेज कॉग्निशन:** लैंग्वेज कॉग्निशन मशीन या कार्यक्रम की क्षमता है शब्दों और वाक्यांशों की पहचान करने और उन्हें मशीन-योग्य प्रारूप में रूपांतरित करने की। आदिकालिक भाषा संज्ञान सॉफ्टवेयर का एक प्रतिबंधित शब्दकोष होता है और यह केवल उसी को पहचान सकता है यदि वे बहुत स्पष्टता से कहे जाते हैं। अधिक परिपूर्ण सॉफ्टवेयर के पास संरचित SQL प्रश्नों के रूप में व्यक्तिगत के बारे में प्रश्न स्वीकार करने की क्षमता होती है।

9. **ह्यूरिस्टिक वर्गीकरण:** AI की इस ज्ञान के माध्यम से स्किल्ड सिस्टम के एक संभावित रूप में, कई स्रोतों की सूचना का उपयोग करके कुछ जानकारी को विभिन्न वर्गों में रखा जा सकता है। एक उदाहरण है कि क्या किसी प्रस्तावित क्रेडिट कार्ड खरीद को स्वीकार किया जाए। क्रेडिट कार्ड के मालिक के बारे में जानकारी, उसके भुगतान का रिकॉर्ड और उस आइटम के बारे में और उस संस्थान के बारे में जहां से वह खरीदारी कर रहा है, उपलब्ध होती है।

10. **गेम खेलना:** कंप्यूटर गेम क्षेत्र में आर्टिफिशियल इंटेलिजेंस महत्वपूर्ण भूमिका निभाता है, जैसे शतरंज, पोकर और टिक-टैक-टो आदि में। यहां मशीनें ह्यूरिस्टिक डेटा पर आधारित संभावित स्थितियों की संभावना का ध्यान रख सकती हैं। कई सौ डॉलर की मास्टर स्तर की चेस खेल सकने वाली मशीनें हैं। उनमें कुछ आर्टिफिशियल इंटेलिजेंस होती है, हालांकि वे मुख्य रूप से ब्रूट फोर्स कम्प्यूटेशन के माध्यम से लोगों के खिलाफ अच्छी तरह से खेलती हैं - हजारों पदों पर आधारित। विश्व चैम्पियन को ब्रूट फोर्स से हराने के लिए विश्वसनीय ह्यूरिस्टिक्स की आवश्यकता होती है, जिसके लिए सेकंड में 200 मिलियन पदों को देखने की क्षमता होनी चाहिए। IBM सुपरकंप्यूटर डीप ब्लू ने एक प्रदर्शनी मुकाबले में विश्व चैम्पियन गैरी कस्पारोव को 3.5 से 2.5 के स्कोर पर हराने में भारी योगदान दिया। (गूडमैन और कीन, 1997)।

1.5 एआई का संक्षिप्त इतिहास

1.5.1 क्लासिकल अवधि (1943 - 1956)

o 1943 वरेन एस. मक्कुलोच और वाल्टर पिट्स ने 1943 में पहले काम किया था जो वर्तमान में AI के रूप में मान्यता प्राप्त कर रहा है। उन्होंने "कंप्यूटर आधारित आर्टिफीसियल न्यूरल नेटवर्क" के लिए प्रेरणा का मॉडल प्रस्तावित किया।

o 1949 डॉनाल्ड हेब द्वारा विकसित हेबियन सीखने का नियम। यह एक सीखने का नियम है जो न्यूरॉनिक गतिविधियों को न्यूरॉन के बीच संबंध पर कैसे प्रभाव डालती है। यह न्यूरल नेटवर्क के भीतर न्यूरॉनिक संबंध के वजन को अद्यतित करने के लिए एक नियम प्रदान करता है।

o 1950 इस साल में सबसे मुख्य काम शामिल हैं खेल सिद्धांत और टर्निंग सिद्धांत प्रमाणित करना। ट्यूरिंग की "परीक्षा", जो मशीन इंटेलिजेंस की परीक्षण करने के लिए एक उपयोगी उपकरण है, इस साल की मूलभूत उपलब्धि है। एलन ट्यूरिंग ने "कंप्यूटिंग मशीनरी और इंटेलिजेंस" प्रकाशित की है जिसमें उन्होंने एक परीक्षा का प्रस्ताव किया है।

o 1951 डीन एडमंड्स और मार्विन मिंस्की ने वस्त्रीय न्यूरॉन एनालॉग प्रोत्साहन कैलकुलेटर, पहला आर्टिफीसियल न्यूरॉन नेटवर्क (ANN) तैयार किया, जिसमें 40 न्यूरॉन के नेटवर्क को अनुकरण करने के लिए तीन हजार वैक्यूम ट्यूब का उपयोग किया गया।

o 1955 एलन न्यूएल और हर्बर्ट ए. साइमन ने पहले आर्टिफीसियल इंटेलिजेंस कार्यक्रम विकसित किया, जिसका नाम "लॉजिक थियोरिस्ट" था।

o 1956 अमेरिकी कंप्यूटर वैज्ञानिक जॉन मैकार्थी ने दर्तमूथ में 1956 के गर्मियों में दो महीने का कार्यशाला आयोजित की थी, जहां पहली बार "आर्टिफीसियल इंटेलिजेंस" का शब्द प्रयुक्त किया गया।

1.5.2 स्वर्णिम काल (1956 - 1974)

1961: जॉर्ज डेवोल द्वारा बनाई गई पहली औद्योगिक रोबोट, यूनिमेट, न्यू जर्सी के ईविंग टाउनशिप में एक जनरल मोटर्स प्लांट की एक संचालन रेखा पर काम करना शुरू हुआ।

1964: डेनियल जी. बोब्रो ने एक प्राकृतिक भाषा समझने वाला कंप्यूटर प्रोग्राम साबित किया और बीजगणित शब्द समस्याओं को हल किया।

1965: हुबर्ट ड्रेफस ने "Alchemy and AI" नामक एक पेपर प्रकाशित किया, जिसमें उत्पन्न करने के लिए पुराना खेल प्रकाशित किया जाता है। आई.जे. गुड ब्रिटिश गणितज्ञ, जिन्होंने पहली अल्ट्रा इंटेलिजेंट मशीन काम किया।

o 1966: जोसेफ वैजेनबॉम ने पहली बार चैटबॉट ELIZA बनाया, जो मानव और मशीन के बीच संवाद की सतहिता को प्रदर्शित करता है।

o 1969: यू-ची हो और आर्थर ब्राइसन ने पीछे की ओर जाने की व्याख्या की, जो एक स्तरित गतिशील प्रणाली के रूप में एक प्रोत्साहन विधि है। आज, बहुस्तरीय आर्टिफीसियल न्यूरल नेटवर्क, इसने गहरे सीखने के सफलता में महत्वपूर्ण योगदान दिया है।

o 1970: जापान के वासेदा विश्वविद्यालय में WABOT - 1, पहला इंटेलिजेंस युक्त इंसानोइड रोबोट बनाया गया।

o 1972: स्टैनफोर्ड यूनिवर्सिटी में माइक्रोबियल संक्रमण का विवेचक और एंटीबायोटिक की सिफारिश के लिए एक पहले विशेषज्ञ प्रणाली MYCIN विकसित की गई।

1.5.3 आधुनिक काल (1974 - आज तक)

o 1976: कम्प्यूटर वैज्ञानिक राज रेड्डी ने एक अनुवेदक पेपर "मशीन द्वारा भाषा पहचान: एक समीक्षा" प्रकाशित किया, जिसमें NLP पर काम के बारे में व्याख्या की गई।

o 1979: स्टैनफोर्ड कार्ट ने बिना मानव हस्तक्षेप के एक कुर्सी भरे हुए क्षेत्र को पार किया, जिससे यह एकत्रित तीन घंटे के भीतर आता है, यह एकत्रित तीन घंटे के भीतर एकत्रित हुआ था, जो स्वतंत्र वाहनों की पहली मिसालों में से एक बन गया।

o 1980: वासेदा विश्वविद्यालय ने WABOT-2 नामक एक म्यूजिशियन रोबोट निर्मित किया, जो एक इंसान के साथ संवाद करने, एक शीट संगीत को पढ़ने और एक इलेक्ट्रिक ऑर्गन पर सादा मुश्किल की धुनें बजाने की क्षमता रखता है।

o 1986: मर्सिडीज-बेंज वैन था पहली ड्राइवरलेस कार, यह कैमरों और सेंसर्स से सुसज्जित थी, जो एन्र्स्ट डिकमैन्स के मार्गदर्शन में म्यूनिख के बंडेसवहर यूनिवर्सिटी में निर्मित हुई, खाली सड़कों पर 55 मील प्रति घंटे तक चल सकती थी।

o 1988: रोलो कारपेंटर ने चैटबॉट जैबरवैकी को निर्मित किया, जो "विचित्र, मनोहारी और हास्यास्पद तरीके में मानव चैट को नकल करने" की कोशिश है। यह मानव संवाद के माध्यम से AI के निर्माण का एक पहल है।

o 1995: रिचर्ड वॉलेस ने चैटबॉट A.L.I.C.E (Artificial Linguistic Internet Computer Entity) विकसित की, जो इंटरनेट के आगमन के साथ अद्वितीय स्तर पर प्राकृतिक भाषा नमूना डेटा का संग्रह करने की सुविधा प्रदान करता है।

o 1997: आईबीएम सुपरकंप्यूटर डीप ब्लू ने विश्व शतरंज चैंपियन गैरी कस्पारोव को एक छः खेल मैच में हरा दिया।

o 2000: मासाचुसेट्स इंस्टीट्यूट ऑफ टेक्नोलॉजी (एमआईटी) के डॉ. सिंथिया ब्रीजल ने Kismet विकसित किया, जो भावनाओं को पहचान सकता है और उन्हें अनुकरण कर सकता है।

o 2002: Roomba एक वैक्यूम क्लीनर है; AI घर में स्वतंत्र रूप से कार्यरत रोबोटिक वैक्यूम क्लीनरों के रूप में दाखिल हुआ।

o 2006: वर्तमान तक वेब उद्योग में AI। कंपनियों जैसे Google, Amazon, Facebook और Netflix ने भी आर्टिफीसियल इंटेलिजेंस का उपयोग करना शुरू किया।

o 2007: प्रिंसटन विश्वविद्यालय ने ImageNet का निर्माण शुरू किया, जो छवियों की एक विस्तृत डेटाबेस है और यह विजुअल वस्तु पहचान सॉफ्टवेयर अनुसंधान में मदद करने के लिए डिज़ाइन किया गया है।

o 2011: आईबीएम का Watson जेपर्डी जीत गया, एक क्विज शो, जहां इसे जटिल सवालों और पहेलियों को हल करना पड़ा। Watson ने साबित किया कि यह भाषा को समझ सकता है और कठिन सवालों को तेजी से हल कर सकता है।

o 2012: Google ने एंड्रॉयड ऐप फीचर "Google Now" शुरू किया, जो उपयोगकर्ता को एक पूर्वानुमान के रूप में जानकारी प्रदान कर सकता है।

o 2014: चैटबॉट "युजीन गूस्टमैन" ने "ट्यूरिंग टेस्ट" में एक प्रतियोगिता जीती।

o 2018: आईबीएम के "प्रोजेक्ट डीबेटर" ने दो मास्टर डीबेटर्स के साथ जटिल विषयों पर वाद-विवाद किया और बहुत अच्छा प्रदर्शन किया।

1.6 आर्टिफिशियल इंटेलिजेंस की वर्तमान स्थिति

पिछले कुछ वर्षों में, आर्टिफिशियल इंटेलिजेंस (एआई) प्रौद्योगिकी में एक अत्यधिक गतिशील और रोमांचकारी क्षेत्र के रूप में उभरा है। एआई में प्रगति ने विश्व स्तर पर कई उद्योगों और समाजों में उत्साह और आशंका दोनों उत्पन्न की है। यह खंड आर्टिफीसियल इंटेलिजेंस की वर्तमान स्थिति का एक व्यापक सारांश प्रस्तुत करता है, इसकी उपलब्धियों, बाधाओं और संभावित दृष्टिकोण पर जोर देता है।

- **मशीन लर्निंग और डीप लर्निंग:** मशीन लर्निंग और डीप लर्निंग के क्षेत्र ने एआई की प्रमुख शाखाओं के रूप में उल्लेखनीय प्रगति का अनुभव किया है। मशीन लर्निंग एल्गोरिदम कंप्यूटर को पैटर्न सीखने और डेटा के आधार पर भविष्यवाणियां करने के लिए सशक्त बनाते हैं, जबकि

मानव मस्तिष्क से प्रेरित डीप लर्निंग एल्गोरिदम जटिल कार्यों को संभालने के लिए तंत्रिका नेटवर्क को नियोजित करते हैं। इन तकनीकों को छवि और भाषण पहचान, प्राकृतिक भाषा प्रसंस्करण और सिफारिश प्रणालियों सहित विभिन्न डोमेन में प्रभावी ढंग से लागू किया गया है।

○ **ऑटोनोमस सिस्टम और रोबोटिक्स:** ऑटोनोमस सिस्टम और रोबोटिक्स के डोमेन ने हाल के वर्षों में महत्वपूर्ण प्रगति की है। विशेष रूप से, स्व-ड्राइविंग कारों ने टेस्ला और वेमो जैसी कंपनियों के नेतृत्व में प्रयोगात्मक प्रोटोटाइप से वास्तविक दुनिया की तैनाती में संक्रमण किया है। इसके अतिरिक्त, रोबोट तेजी से परिष्कृत हो गए हैं, जो विनिर्माण, स्वास्थ्य देखभाल और कृषि जैसे विविध उद्योगों में अपनी क्षमताओं का प्रदर्शन करते हैं। ये प्रगति उद्योगों को नया रूप दे रही हैं और परिवहन और श्रम बाजारों में क्रांति लाने की क्षमता रखती हैं।

○ **नेचुरल लैंग्वेज प्रोसेसिंग:** नेचुरल लैंग्वेज प्रोसेसिंग (एनएलपी) का उद्देश्य कंप्यूटर को मानव भाषा को समझने, व्याख्या करने और उत्पन्न करने में सक्षम बनाना है। एनएलपी ने सिरी और एलेक्सा जैसे आभासी सहायकों के साथ-साथ भाषा अनुवाद सेवाओं में अनुप्रयोगों को सर्चते हुए पर्याप्त प्रगति हासिल की है। ओपनएआई के जीपीटी -3 जैसे अत्याधुनिक मॉडल ने सुसंगत और प्रासंगिक रूप से प्रासंगिक पाठ उत्पन्न करने की क्षमता का प्रदर्शन किया है, सामग्री निर्माण के लिए रोमांचक संभावनाएं प्रस्तुत की हैं और मानव-कंप्यूटर इंटरैक्शन को बढ़ाया है।

○ **हेल्थकेयर में एआई:** एआई के एकीकरण के कारण स्वास्थ्य सेवा क्षेत्र में पर्याप्त प्रगति देखी जा रही है। एआई में निदान, दवा की सर्च और व्यक्तिगत चिकित्सा सहित स्वास्थ्य सेवा के विभिन्न पहलुओं को बढ़ाने की क्षमता है। मशीन लर्निंग एल्गोरिदम रोगी डेटा की विशाल मात्रा का विश्लेषण कर सकते हैं, जिससे पैटर्न की पहचान और रोग के परिणामों की भविष्यवाणी हो सकती है। इसके अतिरिक्त, एआई-संचालित सिस्टम चिकित्सा इमेजिंग के विश्लेषण में सहायता कर सकते हैं, जिससे कैंसर जैसी बीमारियों का जल्दी पता लगाने की सुविधा मिलती है। बहरहाल, संबोधित करने के लिए महत्वपूर्ण चुनौतियां हैं, जैसे कि डेटा गोपनीयता सुनिश्चित करना, नियामक अनुपालन, और एआई-संचालित स्वास्थ्य देखभाल प्रथाओं में मानव निरीक्षण बनाए रखने की आवश्यकता है।

○ **निर्णय लेने में एआई:** क्रेडिट स्कोरिंग, भर्ती प्रथाओं और आपराधिक न्याय जैसे महत्वपूर्ण निर्णय लेने वाले कार्यों में एआई का उपयोग बढ़ रहा है। एआई एल्गोरिदम की व्याख्या और निष्पक्षता निरंतर जांच और आशंका के विषय हैं।

- **एआई और डेटा:** बड़े डेटा के प्रसार और डेटा भंडारण और प्रसंस्करण में प्रगति ने एआई प्रगति को आगे बढ़ाने में महत्वपूर्ण भूमिका निभाई है। एआई सिस्टम प्रशिक्षण और चल रही सीखने की प्रक्रियाओं के लिए व्यापक डेटासेट पर बहुत अधिक निर्भर करते हैं।

1.7 इंटेलीजेंट एजेंट

एजेंट का अर्थ होता है, जैसे मानव, मशीन, रोबोट प्रणाली जो प्रोग्राम की जाती है।

आर्किटेक्चर प्रोग्राम और एजेंट के बीच संबंध की रिपोर्ट की जा सकती है।

एजेंट = आर्किटेक्चर + प्रोग्राम

साधन एजेंट वास्तुकला और कार्यक्रम का संयोजन हैं।

Agent → Percept (Sensors) ⟶ Decision (Agent Program) ⟶ Actions (Effectors)

एक एजेंट कुछ भी है जो सेंसर के माध्यम से अपने पर्यावरण को समझ सकता है और अपने कार्यक्रम के माध्यम से निर्णय ले सकता है और फिर प्रभावकों के माध्यम से उस वातावरण पर कार्य कर सकता है।

उदाहरण: मान लीजिए कि एक एजेंट है और वह बाहर जाता है। लेकिन ध्यान दें कि आज बारिश जैसा वातावरण है। मतलब वह अपनी आंखों से पर्यावरण को महसूस करता है कि आज पानी गिरने वाला है। तो उसके दिमाग में विचार आता है कि आज मैं उसके साथ छाता लेकर जाऊं नहीं तो भीग जाऊंगा। इसलिए वह छाता अपने हाथ में लेता है और फिर से उसी वातावरण में निकल जाता है।

1.7.1 एजेंटों में कुछ शब्दावली का उपयोग किया जाता है:

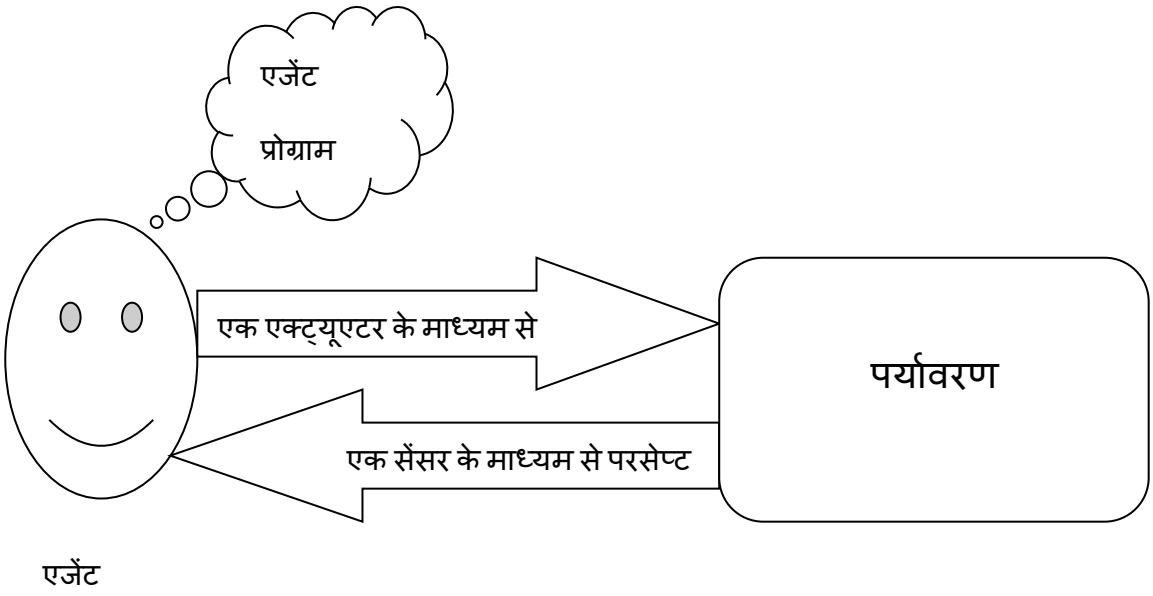
1. **धारणा:**
   दो प्रकार के होते हैं
   1.1 वर्तमान धारणा
   1.2 ऐतिहासिक धारणा
2. **सेंसर:**
   2.1 मानव एजेंट- त्वचा, आंख, नाक, कान।
   2.2 रोबोटिक्स एजेंट- कैमर आईआर (दूरी माप का उपयोग करके)
   2.3 सॉफ्टवेयर एजेंट- अल्ट्रासोनिक सेंसर (ध्वनि आवृत्ति), फोटो प्रतिरोधक (प्रकाश सेंसर)

1.7.2 एजेंट के लक्ष्य

1. उच्च प्रदर्शन
2. अनुकूलित परिणाम
3. तर्कसंगत कार्रवाई

1.7.3 सामान्य एजेंट की वास्तुकला

एजेंट प्रोग्राम एक ऐसा फ़ंक्शन है जो एजेंट मैपिंग को लागू करता है। यह प्रोग्राम डिवाइस चलाता है जिसे आर्किटेक्चर कहा जाता है।



चित्र: 1.5 एजेंट की संरचना

एजेंट दो चीजों को दर्शाता है:

1. **एजेंट फ़ंक्शन:** क्रिया के लिए एक धारणा का मानचित्रण (इसका अर्थ इनपुट की ओर से आउटपुट की योजना बनाना है)
2. **एजेंट प्रोग्राम:** एजेंट फ़ंक्शन का कार्यान्वयन। (कुछ प्रोग्रामिंग भाषा का उपयोग करके)

1.7.4 पीईएएस प्रतिनिधित्व

पहले इंटेलीजेंट एजेंट डिजाइन के लिए सेटिंग निर्दिष्ट कर रहे हैं। कहां

P - प्रदर्शन के उपाय

E - पर्यावरण

A - एक्ट्यूएटर

S - सेंसर

पीईएएस प्रतिनिधित्व का उपयोग करते हुए कुछ एजेंट उदाहरण:

1. **एजेंट: स्वचालित टैक्सी चालक**
    - **प्रदर्शन माप:** सुरक्षित, तेज, आरामदायक यात्रा, कम कीमत
    - **पर्यावरण:** सड़कों की विविधता, अन्य यातायात, विभिन्न प्रकार के यात्री
    - **एक्ट्यूएटर:** स्टीयरिंग व्हील, त्वरक, ब्रेक, सिग्नल, हॉर्न
    - **सेंसर:** कैमरा, स्पीडोमीटर, जीपीएस, सोनार।

2. **एजेंट: चिकित्सा निदान प्रणाली**
    - **प्रदर्शन के उपाय:** स्वस्थ रोगियों, लागत को कम करें
    - **पर्यावरण:** मरीज, कर्मचारी, अस्पताल
    - **एक्ट्यूएटर:** स्क्रीन डिस्प्ले (प्रश्न परीक्षण, उपचार)
    - **सेंसर:** कुंजीपटल (लक्षणों में प्रवेश)

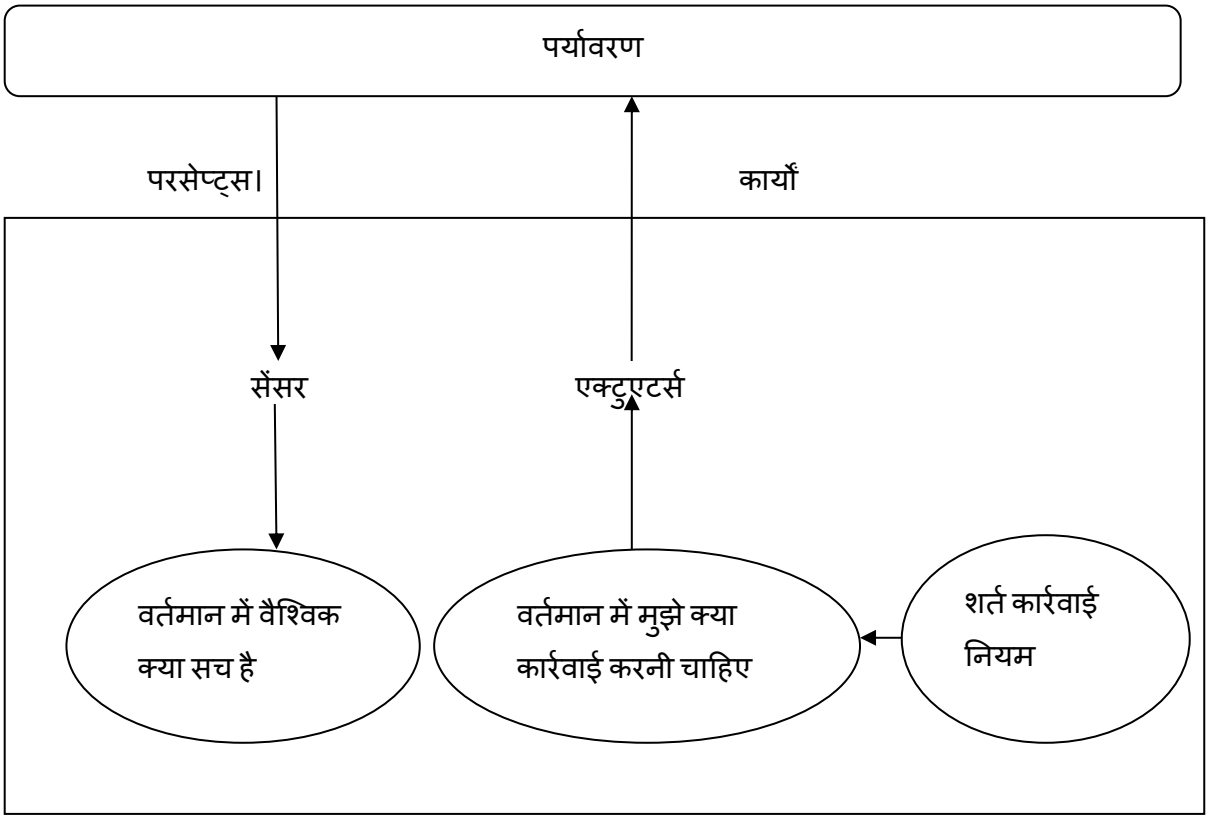
1.7.5 इंटेलीजेंट एजेंटों का वर्गीकरण:

इंटेलीजेंट एजेंट चार प्रकार के होते हैं:

- सरल रिफ्लेक्स एजेंट
- मॉडल-आधारित रिफ्लेक्स एजेंट
- लक्ष्य-आधारित रिफ्लेक्स एजेंट
- उपयोगिता-आधारित रिफ्लेक्स एजेंट

1. **सरल रिफ्लेक्स एजेंट:**

    एक साधारण रिफ्लेक्स एजेंट को शिक्षक उन्मुख एजेंट के रूप में भी जाना जाता है और केवल वर्तमान धारणा के आधार पर कार्य करता है, बाकी परसेप्ट इतिहास को अनदेखा करता है। इसका मतलब है कि एजेंट फ़ंक्शन स्थिति-कार्रवाई नियम पर आधारित है: यदि स्थिति तो कार्रवाई।



चित्र 1.6 सरल रिफ्लेक्स एजेंट

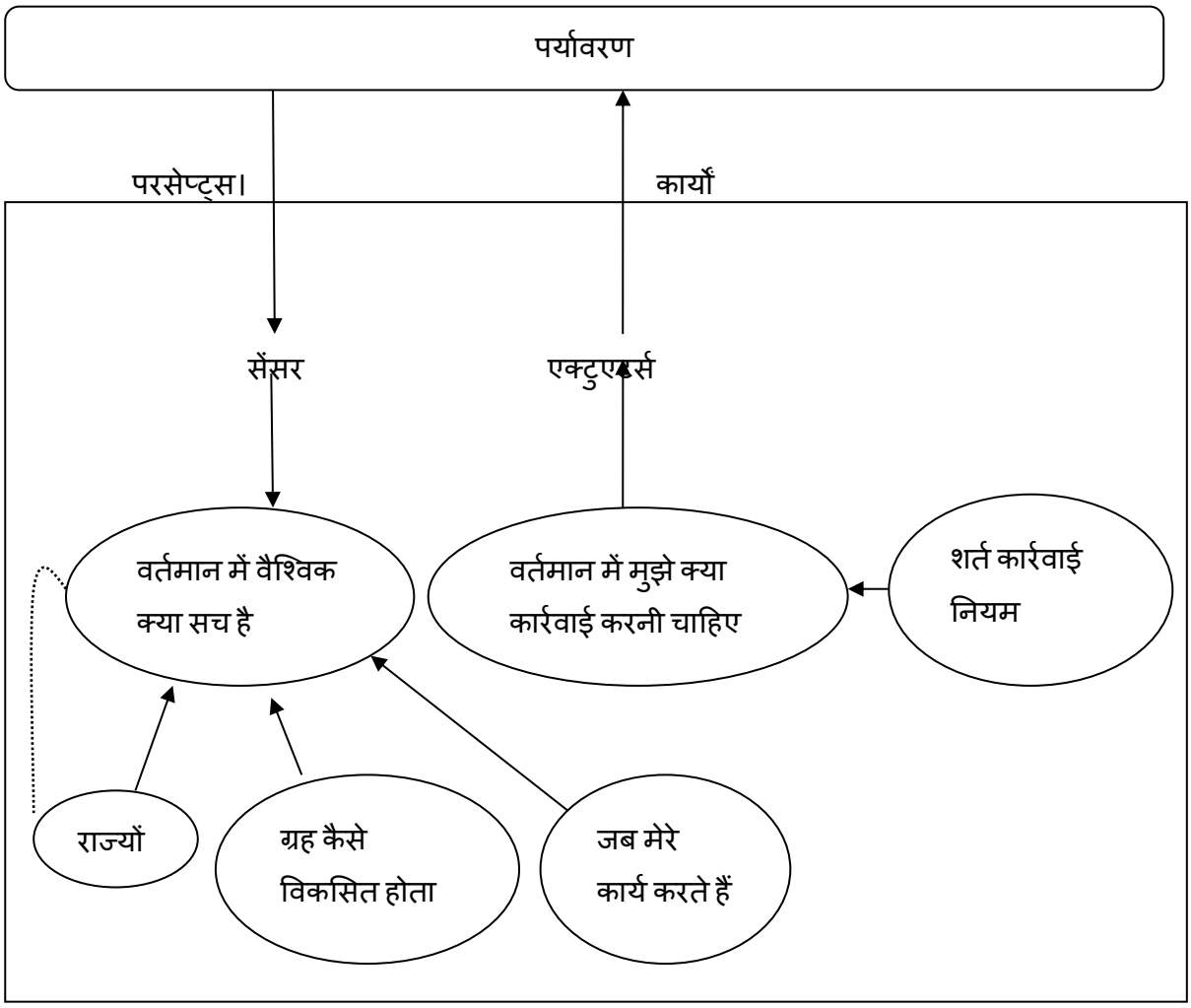
उदाहरण: एक एजेंट है जो एक कमरे के अंदर है और हमने उसे एक शर्त दी है कि यदि तापमान 45 डिग्री सेल्सियस **>** है, तो एसी चालू करें। लेकिन अगर कमरे में कोई व्यक्ति है, तो एजेंट एयर कंडीशनर शुरू करता है अन्यथा ऐसा नहीं होता है। इसलिए इसमें इस दृष्टिकोण का उपयोग नहीं किया जा सकता है, तो हमें इसे संशोधित करना होगा। क्योंकि उसके अंदर ऐसी कोई याद नहीं है कि वह इतिहास की जांच कर सके।

2. मॉडल आधारित रिफ्लेक्स एजेंट:

एक मॉडल आधारित रिफ्लेक्स एजेंट को ज्ञान आधारित एजेंट के रूप में भी जाना जाता है क्योंकि यह इतिहास धारणा द्वारा बनाया गया है।

एक मॉडल आधारित एजेंट आंशिक रूप से देखने योग्य वातावरण को भी संभाल सकता है। इस प्रकार का एजेंट सबसे पहले वर्तमान स्थिति को संग्रहीत करता है। उसके बाद इतिहास की जांच की जाती है। फिर उसके बाद तय करें कि क्या कार्रवाई करनी है।

उदाहरण: **वेमो एलएलसी** एक है मॉडल आधारित रिफ्लेक्स एजेंट का उदाहरण यह अमेरिकी है स्वायत्त ड्राइविंग टेक्नोलॉजी। यह Google की एक परियोजना के रूप में उत्पन्न हुआ है। जिसमें हाई लेवल सेंसर काम करते हैं। मतलब है कि स्वयं से सब कुछ प्रबंधित करता है। चाहे सड़क की जानकारी हो या यातायात की जानकारी, यह सब कुछ संभालती है।

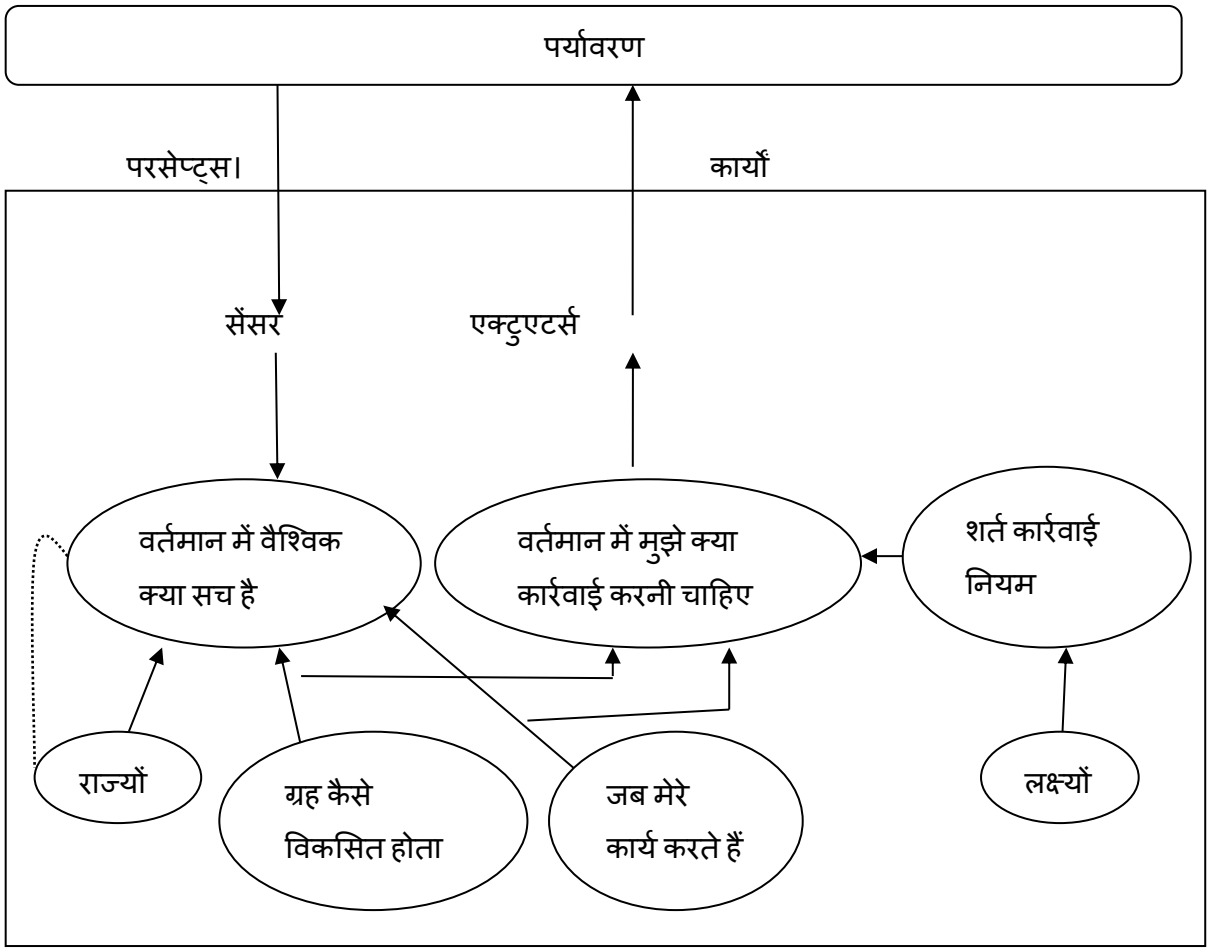


चित्र 1.7 मॉडल-आधारित रिफ्लेक्स एजेंट

3. लक्ष्य-आधारित रिफ्लेक्स एजेंट:

लक्ष्य आधारित एजेंट मॉडल आधारित एजेंटों की क्षमताओं पर और विस्तार करते हैं; यह वांछनीय स्थिति (लक्ष्य) पर आधारित है। यह सर्च और योजना बनाने के दो तरीकों पर आधारित है। यह एजेंटों को कई संभावनाओं के बीच चयन करने का एक तरीका देता है जो एक लक्ष्य स्थिति तक पहुंचता है।

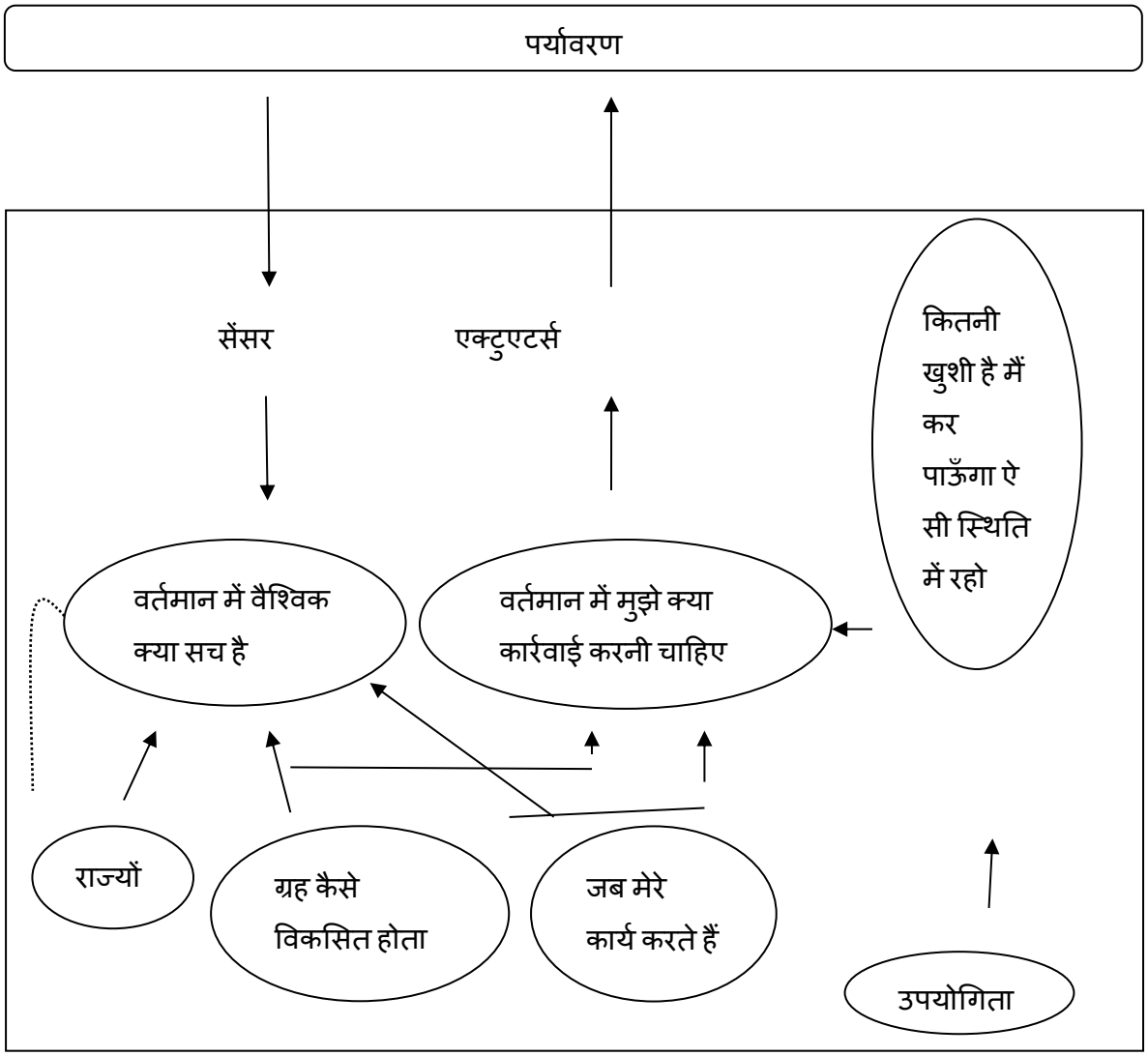
उदाहरण: अली-बाबा जी-प्लस रोबोट ऑनलाइन पैकेट वितरित करने के लिए उपयोग किए जाने वाले लक्ष्य-आधारित रिफ्लेक्स एजेंट का एक वास्तविक उदाहरण है। यह खुद तय करता है कि पैकेट को कैसे पैक करना है और इसे कैसे वितरित करना है।



चित्र 1.8 लक्ष्य-आधारित रिफ्लेक्स एजेंट

4. **उपयोगिता आधारित रिफ्लेक्स एजेंट:** उपयोगिता आधारित रिफ्लेक्स एजेंट उपयोगिता पर ध्यान केंद्रित करता है न कि लक्ष्य पर। एक उपयोगिता को उपयोगिता फ़ंक्शन द्वारा मापा जाता है,

जिसका अर्थ है यू स्टेट जो किसी दिए गए स्टेट में सफलता या खुशी के माप को दर्शाता है। एजेंट किस स्टेट में की जा रही कार्रवाई को पसंद करता है? सुखी या अप्रसन्न अवस्था।



चित्र 1.9 उपयोगिता-आधारित रिफ्लेक्स एजेंट

1.8 इंटेलीजेंट वातावरण

इंटेलीजेंट वातावरण एक भौतिक को संदर्भित करता है मकान जिसके भीतर सूचना प्रौद्योगिकी और अलग व्यापक कंप्यूटिंग तकनीक है इंटरलेसिंग और अभ्यस्त के माध्यम से आओ उपयोगकर्ता के लिए विशिष्ट लक्ष्य, वायुमण्डल नहीं तो हर.

इंटेलीजेंट वातावरण के प्रकार:

1. **पहुँच योग्य / पहुँच योग्य नहीं**.
    - यदि किसी एजेंट के सेंसर इसे एएन कार्रवाई पर निर्णय लेने के लिए आवश्यक वायुमंडल की पूरी स्थिति तक पहुंच प्रदान करते हैं, तो वातावरण सुलभ है।
    - ऐसे वातावरण सुविधाजनक हैं, क्योंकि एजेंट वातावरण के भीतर परिवर्तनों का ट्रैक रखने के कार्य से मुक्त है।

2. नियतात्मक/गैर-निर्धारक

- एक वातावरण तब तय होता है जब वायुमंडल की भविष्य की स्थिति पूरी तरह से वायुमंडल की वर्तमान स्थिति और एजेंट की कार्रवाई से निर्धारित होती है।
- एक सुलभ और व्यवस्थित वातावरण में, एजेंट को अनिश्चितता को प्रभावित करने की आवश्यकता नहीं है।

3. एपिसोडिक/अनुक्रमिक

- एक एपिसोडिक वातावरण का मतलब है कि सीक्वेंट एपिसोड पिछले एपिसोड में हुई कार्रवाइयों पर भरोसा नहीं करते हैं।
- ऐसे वातावरण को एजेंट को आगे की व्यवस्था करने की आवश्यकता नहीं है।

4. स्थैतिक / गतिशील

- एक स्थिर वातावरण संशोधित नहीं होता है जबकि एजेंट सोच रहा है।
- एएन एजेंट के रूप में आपके समय का बीतना अनुपयुक्त है।
- एजेंट को पूरे विचार-विमर्श के दौरान ग्रह का निरीक्षण नहीं करना चाहिए।

5. असतत/निरंतर

- यदि अलग-अलग धारणाओं और कार्यों की मात्रा निषिद्ध है, तो वातावरण क्या है? असतत, अन्यथा यह निरंतर है।

6. इंटेलिजेंसमान विरोधियों के साथ /

- यदि वातावरण में इंटेलिजेंसमान, प्रतिकूल एजेंट होते हैं, तो एजेंट को वातावरण के रणनीतिक, खेल-सैद्धांतिक पहलुओं से संबंधित शामिल होना चाहिए।
- अधिकांश इंजीनियरिंग वातावरण में तर्कसंगत विरोधी नहीं होते हैं, जबकि अधिकांश सामाजिक और आर्थिक प्रणालियां (अधिक या कम) तर्कसंगत एजेंटों की बातचीत से अपनी गुणवत्ता प्राप्त करती हैं।

1.9 आर्टिफिशियल इंटेलिजेंस में समस्या सूत्रीकरण

इंटेलीजेंट कंप्यूटर द्वारा समस्या समाधान एक एजेंट एक ऐसी चीज है जिसे सेंसर के माध्यम से अपने वातावरण को समझने और एक्ट्यूएटर के माध्यम से उस वातावरण पर कार्य करने के रूप में देखा जा सकता है। डाउनसाइड रिज़ॉल्यूशन एजेंट एक यथोचित लक्ष्य आधारित एजेंट है जो यह तय करता है कि आकर्षक स्टेट में समाप्त होने वाले कार्यों के अनुक्रमों को ढूंढकर क्या करना है और क्या करना है। यह प्रारंभिक रूप से एक लक्ष्य और एक दौड़ तैयार करता है। इंटेलिजेंसमान एजेंटों में ज्ञान डोमेन वायुमंडल से मेल खाता है, ऑपरेटर सेंसर के अनुरूप होते हैं और इसलिए सर्च तकनीक एक्ट्यूएटर को मापती है। इसलिए, नर्सिंग इंटेलिजेंसमान एजेंट में एसोसिएट के 3 पैरामीटर हैं:

• **नॉलेज बेस (डेटाबेस):** ज्ञान डोमेन इस कार्य डोमेन का वर्णन करता है और इसलिए लक्ष्य है.
•**ऑपरेटरों:** ऑपरेटर वर्ग माप ज्ञान डोमेन में हेरफेर नहीं करेगा।
• **नियंत्रण रणनीति (सर्च तकनीक):** प्रबंधन रणनीति तय करती है कि ऑपरेटरों को क्या उपयोग करना है और कहां करना है। नर्सिंग सर्च तकनीक में एसोसिएट का उद्देश्य यह है कि लक्ष्य को महसूस करने के लिए प्रारंभिक अवस्था में ऑपरेटरों के लागू अनुक्रम में नर्सिंग में एसोसिएट का अनुप्रयोग।

उद्देश्यों को अक्सर 2 तरीकों से प्राप्त किया जाता है:

(1) फॉरवर्ड रीजनिंग - यह ज्ञान डोमेन के भीतर उन संरचनाओं के लिए ऑपरेटरों के आवेदन को संदर्भित करता है जो कार्य डोमेन का वर्णन करता है ताकि एक बदली हुई स्थिति प्रदान की जा सके। इस तरह के तरीके को बॉटम-अप या डेटा-संचालित तर्क के रूप में भी टिप्पणी की जाती है।

(2) बैकवर्ड रीजनिंग - यह लक्ष्य (समस्या) कथन को उप लक्ष्यों (समस्याओं) में विभाजित करता है जो वर्ग मापना आसान है और जिनके समाधान वर्ग माप पहले नकारात्मक पक्ष को उजागर करने के

लिए पर्याप्त हैं। समस्या निर्माण

इससे पहले कि कोई जवाब अक्सर पाया जाता है, महत्वपूर्ण उद्देश्य यह है कि मामले को बहुत सटीक रूप से रेखांकित किया जाना चाहिए। एक दौड़ को व्यवस्थित करने के लिए निम्नलिखित चरणों के वर्ग माप की आवश्यकता होती है:

(i) समस्या परिभाषा - मामले को ठीक से रेखांकित करें, प्रारंभिक स्थिति (ओं) और इसलिए अंतिम स्थिति (ओं) के विनिर्देशों को देते हुए। इनपुट और स्वीकार्य समाधान के विस्तृत विनिर्देश।

(ii) समस्या विश्लेषण - कुछ महत्वपूर्ण विकल्पों के परिणामस्वरूप मामले का विश्लेषण करने से मामले के समाधान के लिए बाजार पर विभिन्न तकनीकों की गुणवत्ता पर भारी प्रभाव पड़ेगा।

(iii) ज्ञान चित्रण - उस जानकारी का पूरी तरह से प्रतिनिधित्व करें जो अत्यधिक दिए गए डोमेन में नकारात्मक पक्ष को उजागर करने के लिए महत्वपूर्ण है।

(iv) समस्या समाधान - सबसे प्रभावी तकनीक (ओं) का चयन करता है और इसे (उन्हें) वास्तविक नकारात्मक पक्ष पर लागू करता है। नकारात्मक पक्ष के घटक को वास्तविक संभावित अवस्थाओं में पुनर्जन्म लेना चाहिए जो वर्ग माप स्पष्ट रूप से समझ में आता है। ये स्टेट ऑपरेटरों के एक संग्रह द्वारा संचालित वर्ग माप पर संचालित होते हैं और इसलिए उस ऑपरेटर के आह्वान को एक बार और जहां भी उपयोग करना है, प्रबंधन रणनीति द्वारा तय किया जाता है।

समस्याओं को अक्सर निम्नलिखित घटकों द्वारा औपचारिक रूप से रेखांकित किया जाता है:

1. **प्रारंभिक स्थिति:** जहां से भी जवाब शुरू किया जाता है।
2. **ऑपरेटर या उत्तराधिकारी फ़ंक्शन**: एक स्टेट से दूसरे स्टेट में पैंतरेबाज़ी करने के नियम।
3. **स्टेट स्थान**: स्थान या दायरा जिसके दौरान नकारात्मक पक्ष को हल किया जाना है।
4. **पथ:** आंदोलनों की दिशा के लिए व्यावहारिक उत्तर।
5. **पथ लागत:** एक अत्यधिक विशिष्ट दिशा में पूर्ण आंदोलन।
6. **लक्ष्य:** अंतिम स्थिति (वांछित स्थिति जिसे प्राप्त किया जाना है)।

एआई में मुद्दों के प्रकार आर्टिफिशियल इंटेलिजेंस में मुद्दों की 3 शैलियों को मापते हैं:

1. **अनदेखा करने योग्य:** जिसके दौरान उत्तर चरणों पर अक्सर ध्यान नहीं दिया जाता है।
2. **पुनर्प्राप्ति योग्य:** जिसके दौरान उत्तर चरण अक्सर पूर्ववत होते हैं (बैकट्रैक)।
3. **बेवसूल:** उत्तर चरणों को पूर्ववत नहीं किया जा सकता.

उत्पादन प्रणाली: 1972 में न्यूवेल और सिमंस द्वारा मानव संकल्प व्यवहार मॉडलिंग के लिए उत्पादन प्रणालियों का अनुमान लगाया गया था। उत्पादन प्रणाली वर्ग माप को अक्सर विचारशील प्रणालियों, नियम आधारित प्रणालियों या केवल प्रस्तुतियों के रूप में टिप्पणी की जाती है। उत्पादन प्रणालियों में उत्पादन शब्द एक स्थिति-क्रिया नियम को दर्शाता है।

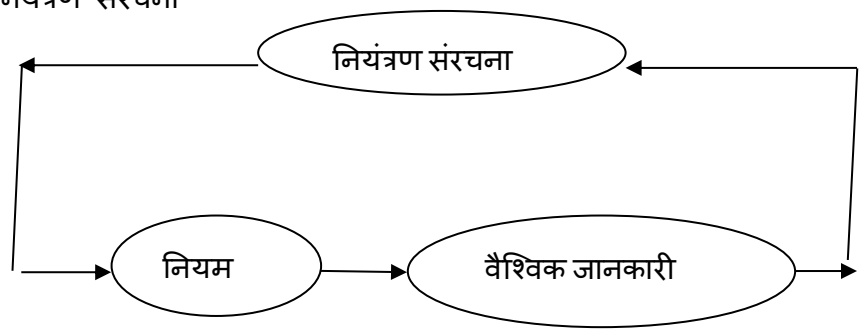
यदि कोई उत्पादन नियमों और 'नियम-दुभाषिया' के साथ एक प्रणाली को अपनाता है, तो प्रणाली को उत्पादन प्रणाली के रूप में समझा जाता है। उत्पादन प्रणालियों के नियम

1. एक मजबूत सूचना चित्रण विषय - उत्पादन प्रणाली न केवल जानकारी का प्रतिनिधित्व करती है, बल्कि कार्रवाई भी करती है।
2. एआई विश्लेषण को कुशल प्रणालियों से जोड़ने वाला पुल - उत्पादन प्रणाली एक ऐसी भाषा प्रदान करती है जिसके दौरान कुशल जानकारी (एक डोमेन में) का चित्रण बेहद स्वाभाविक है।
3. वे मानव व्यवहार के लिए एक अनुमानी मॉडल प्रदान करते हैं।
4. वे इंटेलिजेंसमान कार्रवाई की मजबूत डेटा-संचालित प्रकृति को मॉडल करने का शानदार तरीका है: जैसे ही नए इनपुट जानकारी दर्ज करते हैं, सिस्टम का व्यवहार बदल जाता है।
5. नए नियम बस नई चीजों के लिए साइड में होंगे, जबकि सिस्टम के बाकी हिस्सों को परेशान नहीं करेंगे।

उत्पादन प्रणाली की वास्तुकला

एक उत्पादन प्रणाली का एक विशिष्ट डिजाइन चित्र 1.10 में दिखाया गया है। इसमें 3 मुख्य घटक होते हैं:

1. नियम आधार
2. वैश्विक जानकारी
3. नियंत्रण संरचना



चित्र 1.10: एक उत्पादन के घटक

असेंबली सिस्टम के इन हिस्सों में दोष को बदलना आमतौर पर समस्या चित्रण के रूप में जाना जाता है।

1. नियम आधार: उत्पादन नियम वर्ग सशर्त आईएफ-तब शाखाओं को मापते हैं। उत्पादन प्रणालियों में सूचना चित्रण को एक बहुत ही घोषणात्मक में डिकोड किया जाता है जिसमें नियमों का एक संग्रह होता है जो फॉर्म को मापता है:

स्थिति परिदृश्य कार्रवाई समस्या स्थिति-कार्रवाई नियम वर्ग माप को आमतौर पर उत्पादन-नियम या आईएफ-थेन नियमों के रूप में जाना जाता है। सिद्धांतों का बाद का सामान्य रूप है:

अगर

< पूर्ववर्ती एक **>**

< पूर्ववर्ती **>** की एक जोड़ी

< पूर्ववर्ती और **>**

तब

< परिणाम एक **>** | (निश्चितता के साथ C1)

< परिणाम **>** की एक जोड़ी | (निश्चितता C2 के साथ)

< परिणाम तीन **>** | (निश्चितता के साथ)

उदाहरण:

यदि स्टार्टर काम नहीं करता है

और हेड लाइट स्क्वायर माप मंद है।

और इग्निशन और दबाव हल्का वापस नहीं आता है।

तब बैटरी अच्छी तरह से सपाट हो सकती है।

फिर बैटरी अच्छी तरह से सपाट हो सकती है या बैटरी संबद्धता खराब हो सकती है।

पूर्ववर्ती आधे हिस्से को पंजे के पहलू (एलएचएस) के एक हिस्से के रूप में भी जाना जाता है।

परिणाम आधे को अतिरिक्त रूप से एक्शन हाफ या राइट पहलू (आरएचएस) के रूप में जाना जाता है।

(II) वैश्विक सूचना: विश्व डेटाबेस वह केंद्रीय संगठन है जिसका उपयोग असेंबली सिस्टम द्वारा किया जाता है। दुनिया की जानकारी सभी सिद्धांतों द्वारा एक्सेस की जाएगी। प्रत्येक नियम में एक पूर्व शर्त शामिल होती है जो या तो खुश होती है या दुनिया की जानकारी से नहीं होती है। यदि पूर्व शर्त खुश है, तो नियम लागू किया जाएगा।

(iii) प्रबंधन संरचना या नियंत्रण कार्यनीति: यह मुख्य रूप से एक दुभाषिया कार्यक्रम है जो उस क्रम को प्रबंधित करता है जिसके भीतर असेंबली नियम वर्ग को मापते हैं और संघर्षों को हल करते हैं, अगर एक ही समय में एक से अधिक नियम लागू हो जाते हैं। उन सिद्धांतों को विशिष्ट करने की विधि जो

वर्ग मापती हैं, उनमें से कुछ अनुक्रम तक कोशिश की जाती है यदि पाया जाता है जो समाप्ति की स्थिति (लक्ष्य) को संतुष्ट करने वाली जानकारी उत्पन्न करता है, तो SEARCH विधि कहा जाता है।

1.10 स्टेट स्पेस रिप्रजेंटेशन

इस विषय में हमारे पास सर्च की अटकलों को पेश करने की प्रवृत्ति है। जिन प्रश्नों पर हम {जा रहे हैं} के पते पर जा रहे हैं, उनमें निम्नलिखित हैं:

- क्या सर्च एक समाधान को नोटिस करना सुनिश्चित करती है?
- क्या सर्च हमेशा के लिए समाप्त हो जाएगी (या यह अंतहीन लूप कर सकती है)?
- अगर जवाब मिल जाए, तो क्या वह सबसे अच्छा होगा?
- यह सर्च प्रक्रिया कितनी उन्नत है?
- सर्च की जटिलता को कैसे कम किया जाएगा?
- सर्च को चित्रण भाषा का उपयोग करने के लिए कैसे डिज़ाइन किया जाएगा?

1.11 सारांश

आर्टिफिशियल इंटेलिजेंस एक खुफिया आधारित रोबोट, एक सिग्नल प्रोसेसिंग-नियंत्रित तंत्र, या एक कंप्यूटर कोड बनाने का एक तरीका हो सकता है, जो इंटेलिजेंस को दर्शाता है, उसी तरह से इंटेलिजेंसमान मनुष्य मानते हैं।. आर्टिफिशियल इंटेलिजेंस कंप्यूटर विज्ञान, मनोविज्ञान, जीव विज्ञान, इंजीनियरिंग और उच्च ऊर्जा भौतिकी जैसे विषयों पर आधारित एक विज्ञान और प्रौद्योगिकी है, एआई का एक प्रमुख जोर मानव इंटेलिजेंस से जुड़े कंप्यूटर कार्यों के विकास में है, जैसे तर्क, सीखना और समस्या को हल करना। इस अध्याय में आज की दुनिया की प्रौद्योगिकी में आर्टिफिशियल इंटेलिजेंस के कुछ उदाहरण और अनुप्रयोग हैं जैसे मशीन लर्निंग (पर्यवेक्षित, असुरक्षित और सुदृढीकरण सीखना), कंप्यूटर दृष्टि, प्राकृतिक भाषा प्रसंस्करण, रोबोट, गेम प्लेइंग, डेटा विज्ञान आदि। एजेंट का अर्थ होता है, जैसे मानव, मशीन, रोबोट प्रणाली जो प्रोग्राम की जाती है। एक एजेंट कुछ भी है जो सेंसर के माध्यम से अपने पर्यावरण को समझ सकता है और अपने कार्यक्रम के माध्यम से निर्णय ले सकता है और फिर प्रभावकों के माध्यम से उस वातावरण पर कार्य कर सकता है। परसेप्ट दो प्रकार के होते हैं जैसे ऐतिहासिक और वर्तमान आधारित धारणा। आर्टिफीसियल एजेंट चार प्रकार के होते हैं जैसे कि पहले एक साधारण रिफ्लेक्स एजेंट को शिक्षक उन्मुख एजेंट के रूप में भी जाना जाता है और केवल वर्तमान धारणा के आधार पर कार्य करता है, बाकी परसेप्ट इतिहास को अनदेखा करता है। दूसरा, एक मॉडल आधारित एजेंट आंशिक रूप से अवलोकन योग्य वातावरण को भी संभाल सकता है। तीसरा लक्ष्य आधारित एजेंट मॉडल आधारित एजेंटों की क्षमताओं पर और विस्तार करते हैं; यह सर्च और योजना बनाने के दो तरीकों पर आधारित है। और अंत में उपयोगिता आधारित रिफ्लेक्स एजेंट

उपयोगिता पर ध्यान केंद्रित करता है, न कि लक्ष्य का मतलब खुश और गैर-खुश स्थिति को मापता है। इंटेलिजेंसमान वातावरण एक भौतिक को संदर्भित करता है मकान जिसके भीतर सूचना प्रौद्योगिकी और अलग व्यापक कंप्यूटिंग तकनीक है इंटरलेसिंग और अभ्यस्त के माध्यम से आओ उपयोगकर्ता के लिए विशिष्ट लक्ष्य, वायुमण्डल नहीं तो हर. और एक एजेंट के रूप में इंटेलिजेंट कंप्यूटर द्वारा समस्या समाधान पर चर्चा करें। इसलिए, नर्सिंग इंटेलिजेंसमान एजेंट में एसोसिएट के 3 पैरामीटर हैं: ज्ञान आधार, ऑपरेटर और नियंत्रण रणनीति। उत्पादन प्रणाली वर्ग माप को अक्सर विचारशील प्रणालियों, नियम आधारित प्रणालियों या केवल प्रस्तुतियों के रूप में टिप्पणी की जाती है। उत्पादन प्रणालियों में उत्पादन शब्द एक स्थिति-क्रिया नियम को दर्शाता है। एक उत्पादन प्रणाली का एक विशिष्ट डिजाइन इसमें 3 मुख्य घटक होते हैं: नियम आधार, वैश्विक सूचना और नियंत्रण संरचना। स्टेट अंतरिक्ष प्रतिनिधित्व स्टेट हाउस सर्च प्रक्रिया की अटकलों को पेश करने की प्रवृत्ति है। स्टेट हाउस या ड्राबैक हाउस स्टेट का एक समूह हो सकता है और मामले का प्रतिनिधित्व करने के लिए बीच के संबंध भी हो सकते हैं। ड्राबैक हाउस ग्राफ को एक उलझे हुए घर का प्रतिनिधित्व करने के लिए नियोजित किया जाता है।

प्रमुख शब्द:

- आर्टिफीसियल
- इंटेलिजेंस
- एजेंट
- दृष्टि
- धारणा
- सेंसर
- मशीन
- सीखना
- एक्चुएटर्स
- प्रभावकारक
- मानव मस्तिष्क
- कंप्यूटर
- विशेषज्ञ प्रणाली
- डेटा विज्ञान
- दृष्टि

- छवि अधिग्रहण
- छवि प्रक्रिया
- छवि विश्लेषण और समझ
- तंत्रिका नेटवर्क पागल
- इंटेलिजेंसमान वातावरण
- नर्सिंग
- समस्या समाधान
- समस्या समाधान
- आगे का तर्क
- पिछड़ा तर्क
- उत्पादन प्रणाली
- प्रारंभिक
- लक्ष्य
- पथ

अभ्यास

1. बहुत छोटे प्रकार के प्रश्न (2 अंक):

a) आर्टिफिशियल इंटेलिजेंस से आप क्या समझते हैं?

b) डेटाबेस और नॉलेज बेस के बीच अंतर

c) मजबूत एआई और कमजोर एआई के बीच अंतर

d) फॉरवर्ड रीजनिंग और बैकवर्ड रीजनिंग में अंतर है।

e) पर्यवेक्षित और असुरक्षित सीखने के बीच अंतर करें।

f) एजेंट के परसेप्ट अनुक्रम को परिभाषित करें।

g) एक विशेषज्ञ प्रणाली क्या है?

h) आर्टिफिशियल इंटेलिजेंस का इतिहास लिखिए।

i) प्राकृतिक भाषा प्रसंस्करण को परिभाषित करें।

j) सुदृढीकरण सीखने को परिभाषित करें।

k) उपयुक्त उदाहरण के साथ एजेंट प्रोग्राम की संरचना का वर्णन करें।

l) कंप्यूटर विजन क्या है?

m) "सैटेलाइट एजेंट" के लिए पीईएएस उपाय डिजाइन करें।

n) आर्किटेक्चर की मदद से लर्निंग एजेंट को परिभाषित करें।

o) मानव दृष्टि और कंप्यूटर दृष्टि के बीच अंतर क्या है?

p) मशीन लर्निंग क्या है?

q) इंटेलिजेंसमान एजेंट क्या है?

r) धारणा क्या है?

s) स्टेट स्थान क्या है?

t) उत्पादन प्रणाली को परिभाषित करें। एक अच्छी नियंत्रण रणनीति की आवश्यकताएं क्या हैं?

2. लघु प्रकार के प्रश्न (5 अंक):

a) आर्टिफिशियल इंटेलिजेंस का मतलब क्या है? यह मानव इंटेलिजेंस से कैसे भिन्न है?

b) विशेषज्ञ प्रणाली पर एक संक्षिप्त टिप्पणी लिखिए।

c) विभिन्न एआई तकनीकों की व्याख्या करें।

d) एयरोस्पेस सिस्टम के लिए पीईएएस क्या निर्दिष्ट करता है।

e) एआई के विभिन्न कार्य डोमेन की सूची को स्पष्ट कीजिए।

f) एआई की नींव पर एक संक्षिप्त नोट लिखें।

g) आर्टिफीसियल तकनीक से क्या तात्पर्य है?

h) एक उदाहरण के साथ सरल रिफ्लेक्स एजेंट की अवधारणा को समझाएं।

i) प्राकृतिक भाषा प्रसंस्करण में शामिल चरणों की व्याख्या करें।

j) किसी भी एआई समस्या को हल करने के लिए स्टेट अंतरिक्ष दृष्टिकोण की व्याख्या करें।

k) उत्पादन प्रणाली को स्पष्ट कीजिए।

l) एआई में कंप्यूटर विजन की भूमिका का वर्णन करें।

m) पीईएएस को परिभाषित करें: स्वचालित टैक्सी चालक और शतरंज खिलाड़ी का एक उदाहरण.

3. लंबे प्रकार के प्रश्न (10 अंक):

a) प्राकृतिक इंटेलिजेंस और आर्टिफीसियल इंटेलिजेंस के बीच अंतर क्या है? समझाएं कि एआई तकनीकों का प्रतिनिधित्व कैसे किया जा सकता है।

b) आर्टिफीसियल इंटेलिजेंस को परिभाषित करें। उस क्षेत्र की चर्चा करें जिसमें एआई के अनुप्रयोग का उपयोग किया जाता है।

c) नॉलेज रिप्रजेंटेशन और ज्ञान अधिग्रहण के बीच अंतर करें।

d) आर्टिफीसियल इंटेलिजेंस के अनुप्रयोग क्या हैं?

e) इंटेलिजेंट एजेंट क्या है? एजेंट प्रोग्राम के बुनियादी प्रकार का वर्णन करें।

f) एजेंट प्रोग्राम से क्या तात्पर्य है? आप कैसे आश्वस्त करते हैं कि एक एजेंट प्रोग्राम एक इंटेलिजेंसमान एजेंट प्रोग्राम है?

g) आर्टिफीसियल इंटेलिजेंस का अनुप्रयोग लिखिए। इंटेलिजेंसमान एजेंटों को परिभाषित करें। इंटेलिजेंसमान एजेंटों की संरचना का वर्णन कीजिए।

(या)

इंटेलिजेंसमान एजेंट को परिभाषित करें। उपयुक्त उदाहरण के साथ विभिन्न प्रकार के एजेंट कार्यक्रमों की व्याख्या करता है।

h) घटकों और विशेषताओं के साथ उत्पादन प्रणाली की व्याख्या करें। अच्छी नियंत्रण रणनीतियों की आवश्यकताओं को सूचीबद्ध करें।

i) स्टेट अंतरिक्ष सर्च के रूप में किसी समस्या को कैसे परिभाषित करें? एक उदाहरण की सहायता से चर्चा कीजिए।

ग्रन्थसूची

अरूप एट अल., 1994

आरूप, एम., एरेन्टोफ्ट, एम.एम., पैरोड, वाई., स्टैडर, जे., और स्टोक्स, आई. (1994)। ऑप्टिमम-एआईवी: अंतरिक्ष यान एआईवी के लिए एक ज्ञान-आधारित योजना और शेड्यूलिंग प्रणाली। फ़ॉक्स में, एम. और ज़्वेबेन, एम., संपादक, ज्ञान आधारित शेड्यूलिंग। मॉर्गन कॉफ़मैन, सैन मेटो, कैलिफ़ोर्निया।

अबू-मुस्तफा और साल्टिस, 1987

अबू-मुस्तफा, वाई.एस. और साल्टिस, डी. (1987)। ऑप्टिकल न्यूरल कंप्यूटर. साइंटिफिक अमेरिकन, 256:88-95.

आचार्य एट अल., 1992

आचार्य, ए., तांबे, एम., और गुप्ता, ए. (1992)। संदेश भेजने वाले कंप्यूटरों पर उत्पादन प्रणालियों का कार्यान्वयन। समानांतर और वितरित प्रणालियों पर आईईईई लेनदेन, 3(4):477-487।

बैचस, 1990

बैचस, एफ. (1990)। संभाव्य ज्ञान के साथ प्रतिनिधित्व और तर्क। एमआईटी प्रेस, कैम्ब्रिज, मैसाचुसेट्स।

बैचस एट अल., 1992

बैचस, एफ., ग्रोव, ए., हेल्पर, जे.वाई., और कोल्लर, डी. (1992)। आंकड़ों से लेकर मान्यताओं तक. आर्टिफिशियल इंटेलिजेंस (एएएआई-92) पर दसवें राष्ट्रीय सम्मेलन की कार्यवाही में, पृष्ठ 602-608, सैन जोस, कैलिफोर्निया। एएएआई प्रेस।

कैयानेलो, 1961

कैयानेलो, ई.आर. (1961)। विचार और विचार मशीनों के सिद्धांत की रूपरेखा। सैद्धांतिक जीवविज्ञान जर्नल, 1:204-235।

कैंपबेल एट अल., 1991

कैंपबेल, पी.के., जॉन्स, के.ई., ह्यूबर, आर.जे., हॉर्च, के.डब्ल्यू., और नॉर्मन, आर.ए. (1991)। एक सिलिकॉन-आधारित, 3-आयामी तंत्रिका इंटरफ़ेस: एक इंट्राकॉर्टिकल इलेक्ट्रोड सरणी के लिए विनिर्माण प्रक्रियाएं। बायोमेडिकल इंजीनियरिंग पर आईईईई लेनदेन, 38(8):758-768

हास, 1986

हास, ए. (1986)। विश्वास और क्रिया का एक वाक्यात्मक सिद्धांत। आर्टिफिशियल इंटेलिजेंस, 28(3):245-292.

इंगरमैन, 1967

इंगरमैन, पी.जेड. (1967)। पाणिनी-बैकस फॉर्म का सुझाव दिया गया। कंप्यूटिंग मशीनरी एसोसिएशन के संचार, 10(3):137।

अध्याय - दो

इंटेलीजेंट सर्च द्वारा समस्या समाधान - I

2.1 परिचय: किसी भी समस्या के समाधान के लिए दो मुख्य मापदंडों की आवश्यकता होती है: एक उचित ढंग से संरचित स्थिति ग्रहण करके समस्या का प्रारंभिक चित्रण और इसके बाद स्थिति ग्रहण की मौजूदगी का परीक्षण। दिए गए प्रारंभिक स्थिति से गोल स्थिति तक जाने के लिए गणना और आवश्यकतानुसार अवस्थाओं के माध्यम से श्रेष्ठ मार्ग की निर्धारण, इस अध्याय में सवालों को विस्तृत रूप से विचार किया जाता है। इसलिए, यह अध्याय कुछ प्रसिद्ध सर्च एल्गोरिदमों के साथ शुरू होता है, जैसे कि गहराई पहले और फिर प्रश्नों के समय और स्थानीय जटिलता पर विशेष ध्यान देता है। फिर यह धीरे-धीरे 'ह्यूरिस्टिक सर्च' एल्गोरिदमों की सर्च करता है, जहां सर्च स्थान में राजमार्ग का क्रम हुक निर्धारित करने के लिए हीउरिस्टिक्स नामक नियमों पर आधारित होता है, और इनके उपयोग को प्रदर्शित करता है उच्चकोटि समस्या के समाधान में। हमने पहले ही इंटेलिजेंसमान सर्च द्वारा हल किए जा सकने वाली कुछ समस्याओं के बारे में बात कर ली हैं। उदाहरण के लिए, प्रसिद्ध जल-पेटी समस्या, संख्या पहेली समस्या और धर्मियों-शिकारियों समस्या इंटेलिजेंसमान सर्च द्वारा हल की जा सकने वाली आदर्श उदाहरण हैं।

2.2 सर्च ग्राफ और सर्च ट्री की अवधारणा:

सामान्य ज्ञान दिखाता है कि सर्च समस्या के दो महत्वपूर्ण मुद्दे होते हैं: पहला 'सर्च क्या करें' और दूसरा 'सर्च कहाँ करें'। पहला आमतौर पर 'मुद्दा' कहा जाता है, जबकि दूसरा 'सर्च स्थान' कहलाता है। आर्टिफीसियल इंटेलिजेंस में सर्च स्थान को आमतौर पर स्टेट का एक समूह माना जाता है और इसलिए इसे स्टेट स्थान कहा जाता है। सामान्य सर्च स्थान के विपरीत, इंटेलिजेंसमानता में सभी समस्याओं में स्टेट स्थान पूरी तरह से नहीं ज्ञात होता है, समस्या के हल तक। स्टेट स्थान के विकास और उसके पश्चात स्टेट के यात्रा के क्रम के आधार पर, AI में सर्च समस्याएं अलग-अलग नामों से ज्ञात होती हैं। उदाहरण के लिए, एक ट्री के रूप में जोर करने वाले एक हौल के स्टेट स्थान की गणना करें। अब, यदि हम ट्री के हर स्तर पर गोल सर्चते हैं, जिसकी शुरुआत मूल स्थान से होती है और मुख्य गहराई तक जाती है, तो हम इसे चौड़ी पहली सर्च कहते हैं। दूसरी ओर, हम कभी-कभी ट्री के मुख्य गहराई पर गोल सर्चते हैं, और गहराई में और तर्क नहीं कर सकते हैं। हम फिर उस नोड (स्टेट) के विभाजित्पुत्रों की कई संभावित स्टेट का पता लगाने का प्रयास करते हैं जिसे हमने अंतिम रूप से यात्रा की है। यदि हम ट्री के नोड्स का अनुकरण उपरोक्त सिद्धांतों के माध्यम से सर्चने के लिए करते हैं, तो यह चौड़ी पहली यात्रा कहलाती है और इसलिए सर्च रणनीति को गहराई पहली सर्च कहा जाता है।

2.3 <u>सामान्य समस्या समाधान दृष्टिकोण</u>

एआई में नकारात्मक सर्च तकनीकों की काफी बड़ी मात्रा मौजूद है जो सर्च में विश्वास रखती हैं। उनमें से सबसे अच्छा यह है कि उत्पादन और जांच पद्धति है। उत्पन्न और जांच पद्धति के लिए सूत्र औपचारिक रूप से निम्नानुसार स्पष्ट है:

- **प्रक्रिया: उत्पन्न करें और जांचें;**

- **निकल**
- **दोहराना**

एक प्रतिस्थापन स्थिति उत्पन्न करें और इसे वर्तमान-स्थिति का निर्णय लें;

- **तक** वर्तमान-स्थिति = लक्ष्य;
- **रुको।**

सूत्र से उच्च से स्पष्ट है कि सूत्र दोहराव के हर पुनरावृत्ति में एक प्रतिस्थापन स्थिति की सर्च करने का मौका जारी रखता है- जब तक कि लूप और बाहर नहीं निकलता केवल वर्तमान स्थिति लक्ष्य में सक्षम है। सूत्र के भीतर सबसे महत्वपूर्ण आधा प्रतिस्थापन स्थिति प्राप्त करना है। यह एक सीधा काम नहीं हो सकता है। यदि हाल के स्टेट का निर्माण संभव नहीं है, तो सूत्र को समाप्त कर दिया जाना चाहिए। हमारे सरल सूत्र में, हम, हालांकि, इसे सरल बनाए रखने के लिए पसंद से इसे मूर्त रूप देने में विफल रहे।

लेकिन क्या मैं एक समस्या की स्थिति उत्पन्न करता हूं? इसे औपचारिक रूप देने के लिए, हम एक चार टपल को रेखांकित करते हैं, जिसे स्टेट स्थान के रूप में संदर्भित किया जाता है, जिसे

{नोड्स, आर्क, लक्ष्य, वर्तमान}

कहां

नोड्स सर्च स्थान के भीतर मौजूदा स्टेट के सेट का प्रतिनिधित्व करें;
एक **चाप** एसोसिएट डिग्री ऑपरेटर को दर्शाता है जो एसोसिएट डिग्री मौजूदा स्टेट के लिए लागू होता है जो संक्रमण का कारण बनता है [एक अलग स्टेट के अलावा एसोसिएट डिग्री में);
लक्ष्य नोड्स के भीतर ज्ञात होने के लिए आवश्यक स्थिति को दर्शाता है; और **प्रवाह** स्टेट का प्रतिनिधित्व करता है, जो वर्तमान में लक्ष्य के साथ मिलान के लिए उत्पन्न होता है।
सर्च विधियों को आम तौर पर दो श्रेणियों में विभाजित किया जाता है: यूनिफार्म सर्च और सूचित सर्च।

यूनिफार्म सर्च: इस सर्च को ब्रूट फोर्स सर्च या ब्लाइंड सर्च या संपूर्ण सर्च भी कहा जाता है। इसे ब्लाइंड सर्च कहा जाता है क्योंकि जिस तरह से सर्च स्पेस के बारे में किसी भी जानकारी का उपयोग किए बिना सर्च ट्री की सर्च की जाती है। यह इतना प्रभावी नहीं है क्योंकि किसी भी डोमेन विशिष्ट ज्ञान का उपयोग किए बिना सर्च ट्री का मतलब है कि लक्ष्य पाए जाते हैं लेकिन मार्ग परिभाषित नहीं किया जाता है, इस शब्द को यूनिफार्म सर्च परिभाषित किया गया है। यूनिफार्म सर्च मुख्य रूप से दो निम्नलिखित तकनीकों का उपयोग करके:

1. ब्रेड्थ फर्स्ट सर्च (स्तर क्रम)
2. डेप्थ फर्स्ट सर्च (प्रीऑर्डर)

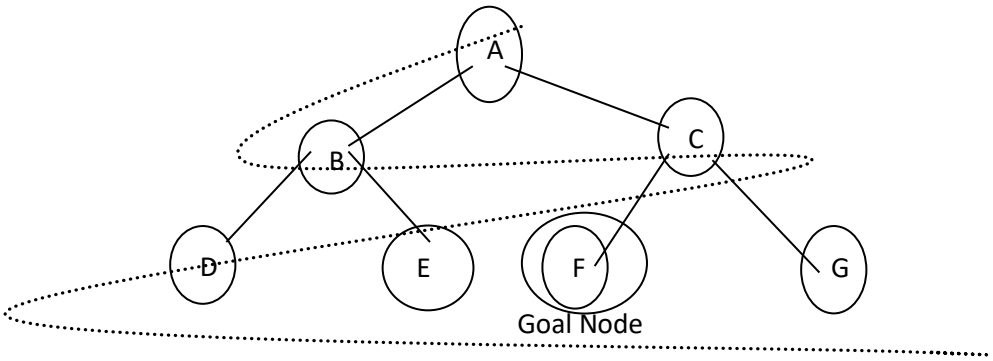
2.3.1 <u>ब्रेड्थ फर्स्ट सर्च</u>

ब्रेड्थ फर्स्ट सर्च एक आसान रणनीति हो सकती है जिसके दौरान आधार नोड प्रारंभिक रूप से समाप्त हो जाता है, फिर आधार नोड क्षेत्र इकाई के सभी उत्तराधिकारी समाप्त हो जाते हैं, फिर उनके उत्तराधिकारी। FIFI (फर्स्ट इन फर्स्ट आउट) ऑर्डर के दौरान नोड्स समाप्त होने जा रहे हैं।

कलन विधि:

- रूट नोड को कतार में डालें।
- यदि कतार खाली है, तो विफलता और रुक जाओ।
- यदि क्यू कतार पर प्रारंभिक तत्व एक लक्ष्य नोड हो सकता है, तो सफलता और अन्यथा
- प्रारंभिक तत्व को हटाएं और विस्तार करें और बच्चों को कतार के अंतिम में रखें।
- चरण - 2 वापस करें (दोहराएं चरण - 2 लक्ष्य मिल जाएगा)

उदाहरण:



चित्र 2.1: ब्रीथ फर्स्ट सर्च का उदाहरण

समझाना:

खुला (A)

व्यय (X) = A बंद करें (A)

खुला (बी, सी)

व्यय (B) = (X) बंद करें (A)

खुला (सी, डी, ई)

व्यय (X) = C बंद करें (C)

खुला (डी, ई, एफ, जी)

व्यय (X) = D बंद करें (D)

खुला (ई, एफ, जी)

व्यय (X) = E बंद करें (E)

खुला (F, G)

व्यय (F) लक्ष्य नोड = उत्तराधिकारी

पूर्णता और ऑप्टिमालित्य

कुछ बहुत अच्छी खबर ब्रीथ फर्स्ट पहली सर्च:

- ब्रीथ फर्स्ट सर्च पूर्णता की गारंटी देती है: यदि कोई मौजूद है एक जवाब यह होगा अंत में पाया जा सकता है।
- ब्रीथ फर्स्ट सर्च इसके अतिरिक्त ऑप्टिमालित्य की गारंटी देता है: प्राथमिक प्रस्ताव वापस आ गए होने जा रहे हैं जितना छोटा प्राप्य. (टिप्पणी: इष्टतमता की यह व्याख्या मानती है कि हर एक चलना पास एक कीमत का एक. वास्तविक के साथ दाम यह कार्य करता है मर्जी बनना थोड़ा बहुत कुछ चिंतित.) उस गहराई-पहली सर्च को याद करें। नहीं जिम्मेदारी या तो पूर्णता या इष्टतमता।

ब्रीथ फर्स्ट सर्च का जटिलता विश्लेषण

1. **समय जटिलता:** मान लीजिए कि गहराई क्या है? प्राथमिक प्रस्ताव और बी को शाखा बनने दें। समस्या (फिर, सादगी के लिए स्थिर माना जाता है)।
2. तब **सबसे खराब स्थिति समय जटिलता ओ (बी^{d}).**
3. **अंतरिक्ष जटिलता:** विशाल अंतर; फिलहाल हमारे पास है स्टोर करने के लिए हर रास्ता पहले भी देखा जा चुका है, जबकि गहराई के लिए- सबसे पहले। हम करते हैं केवल था रहने के लिए एक स्मृति में शाखा। अतः **अन्तरिक्ष जटिलता यह भी है O (b^{d}).**

ब्रीथ फर्स्ट सर्च के लाभ

1. विस्तार पहला ढूँढ सकना नहीं। मिलना घुसना हमेशा के लिए बेकार रास्ते की सर्च करना।
2. अगर वहाँ है एक जवाबविस्तार पहला ढूँढ निश्चित रूप से होगा सूचना यह बाहर है.
3. अगर वहाँ है पूर्णतया एक प्रस्ताव फिर चौड़ाई पहला सर्च की जाएगी सूचना वही सीमारेखा एक उस जरूरतों कम विविधता चरणों की संख्या।

ब्रीथ फर्स्ट सर्च के नुकसान

1. सबसे ज्यादा नुक्सान चौड़ाई की संख्या पहला सर्च इसकी स्मृति है। मांग. क्योंकि हर ट्री का स्तर चाहिए बचाया जाए इस प्रकार प्राप्त करने के लिए आगामी स्तर और इसलिए परिमाण स्मृति का समानुपाती क्या है? मात्रा नोड्स की संख्या रुकोवही क्षेत्र जटिलता। चौड़ाई की संख्या पहला सर्च O(b^{d}). नतीजतन, चौड़ाई पहला सर्च गंभीर रूप से अंतरिक्ष-बाध्य है परिशीलन कर इस प्रकार सकना स्मृति समाप्त हो जाती है बाहर वहाँ विशिष्ट कंप्यूटरों पर एक अत्यधिक में मिनटों की बात।
2. अगर जवाब दूर है दूर से आधारसाँस पहला ढूँढ सकना खाना स्वर अपने समय के बारे में.

2.3.2. डेप्थ फर्स्ट सर्च

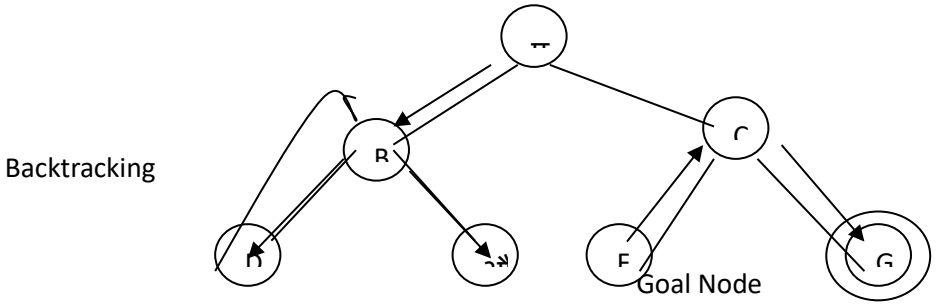
यह देखते हुए कि एक पथ दूसरे के रूप में अच्छा है, एक योजना प्रत्येक नोड पर विकल्प चुनना और उस विकल्प से आगे काम करना है। एक बार माता-पिता के पास जाने के बाद, एल्गोरिथम प्रोग्राम मूल नोड की एक नई संतान

प्राप्त करने का प्रयास करता है। सिद्धांत से उच्च का उपयोग अत्यधिक गहराई वाली प्रारंभिक सर्च में ट्री के हर नोड के लिए पुनरावर्ती रूप से किया जाता है। गहराई से पहले सर्च एल्गोरिथम प्रोग्राम के भीतर सूत्र को समझने के लिए एक आसान चीजें स्टैक का उपयोग करना है।

कलन विधि:

- स्टैक पर शुरुआती नोड्स रखें (डेटा संरचना का साधन)
- यदि स्टैक खाली है, तो विफलता वापस करें और बंद करें
- यदि एस स्टैक पर प्रारंभिक तत्व एक लक्ष्य नोड हो सकता है, तो सफलता और अन्यथा
- प्रारंभिक तत्व को हटाएं और विस्तार करें और बच्चों को कतार के बिंदु पर रखें।
- चरण - 2 वापस करें (दोहराएं चरण - 2 लक्ष्य मिल जाएगा)

उदाहरण:



चित्र 2.2: गहराई प्रथम सर्च का उदाहरण

समझाना:

गहराई प्रथम सर्च (DFS) ट्रैवर्सल पथ ABDECFG है।

1. स्पेस कम्प्लेक्सिटी (जटिलता)

गहराई में अधिकतम स्मृति की आवश्यकता होती है, पहली सर्च के बाद, जब हम प्राथमिक समय में सबसे बड़ी गहराई तक पहुंचते हैं। यह मानते हुए कि प्रत्येक नोड में एक ब्रांचिंग इश्यू बी शामिल है, एक बार जब गहराई डी पर एक नोड की जांच की जाती है, तो मेमोरी स्क्वायर में सहेजे गए नोड्स की मात्रा गहराई डी तक सभी अविस्तारित नोड्स को मापती है और नोड की जांच की जाती है। चूंकि हर स्तर पर वर्ग माप (बी -1) अविस्तारित नोड्स होते हैं, इसलिए आवश्यक मेमोरी की पूरी श्रृंखला = डी (बी -1) + 1।

इसलिए गहराई से पहली सर्च की घर की जटिलता बी का एक रैखिक संचालन हो सकती है, डेप्थ फर्स्ट सर्च के विपरीत, जहां यह बी के गणितीय कार्य को जोड़ता है। यह, वास्तव में, गहराई से पहली सर्च का सबसे उपयोगी पहलू है।

2. समय जटिलता:

यदि हम गहराई d पर बाईं स्थिति में लक्ष्य का पता लगाते हैं, तो जांच किए गए नोड्स की मात्रा = (d +1)। दूसरी ओर, यदि हम गहराई डी पर तीव्र दाईं ओर लक्ष्य का पता लगाते हैं, तो जांच किए गए नोड्स की मात्रा ट्री के भीतर सभी नोड्स का प्रतीक है, जो

$$1+b+b^2+b^3+...+b^d = (\text{बी})^{d+1} -1) / (\text{बी} -1)$$

तो, एक औसत मामले में नोड्स की पूरी श्रृंखला की जांच की गई।

$$= (d +1) /2 + (b^{d+1} -1) / 2 (\text{बी} -1)$$

$$= b(b)^d + d) / 2 (b -1)$$

यह गहराई 1 सर्च सूत्र की सामान्य केस समय जटिलता है। चूंकि विशाल गहराई डी के लिए, गहराई 1 सर्च को काफी विशाल रनटाइम की आवश्यकता होती है, इसलिए मामले को हल करने के लिए वैकल्पिक धन्यवाद सर्च ट्री की गहराई से प्रमुख है।

इस तरह के सूत्र, जहां भी उपयोगकर्ता हर पुनरावृत्ति पर प्रारंभिक गहराई कट-ऑफ का उल्लेख करता है, को एक पुनरावर्ती गहराई गहराई पहले सर्च या सिर्फ एक पुनरावृत्ति गहन सर्च कहा जाता है।

गहराई-पहली सर्च का जटिलता विश्लेषण

1. समय जटिलता:

- जैसा कि अनंत लूप हो सकते हैं, सबसे खराब स्थिति में, सीधी गहराई-पहले एल्गोरिथम नियम को रोका नहीं जा सकता है। इसलिए, हमारे पास इसके बजाय गहराई से बंधे गहराई-पहले सर्च का विश्लेषण करने के लिए प्रगति को मापने की प्रवृत्ति है।
- शीर्ष गहराई की अनुमति दी जाए। (यदि हमारे पास यह समझने की प्रवृत्ति है कि ट्री के भीतर कोई शाखा डी से लंबी नहीं हो सकती है, तो हमारा विश्लेषण विपरीत दो गहराई-पहले सर्च एल्गोरिदम पर भी लागू होगा।
- सरलता के लिए, मान लें कि प्रत्येक संभावित स्टेट के लिए वर्ग माप सटीक बी संभावित अनुवर्ती स्टेट को मापते हैं। यही है, बी सर्च ट्री की शाखाओं का मुद्दा है।

2. सबसे खराब स्थिति में, प्रत्येक शाखा की लंबाई डीमैक्स (या अधिक) होती है और लक्ष्य स्थिति के साथ लेबल किया गया एकमात्र नोड यह है कि दाईं शाखा पर अंतिम नोड है। इसलिए, गहराई-पहली सर्च उत्तर सर्चने से पहले ट्री के भीतर सभी नोड्स (गहराई डीमैक्स तक) का दौरा कर सकती है। **सबसे खराब स्थिति में गहराई से सर्च की समय गुणवत्ता O(b) में है।^{d}).**

3. स्पेस कम्प्लेक्सिटी:

अच्छी खबर यह है कि गहराई-पहली सर्च इसकी स्मृति आवश्यकताओं की दृष्टि में बेहद किफायती है:

- समय में किसी भी उद्देश्य पर, हमारे पास केवल निशान को आधार से वर्तमान नोड तक स्मृति में रखने की प्रवृत्ति है, और - सटीक कार्यान्वयन के आधार पर - संभवतः प्रत्येक नोड्स के लिए सभी एसआईबी नोड्स को संयुक्त रूप से जोड़ा जाता है।
- इसलिए, सबसे खराब स्थिति में घर की गुणवत्ता O(b^d). यही है, गुणवत्ता डी में रैखिक है। वास्तव में, प्रोग्रामिंग भाषा बैकट्रैकिंग का उपयोग करने के परिणामस्वरूप, एसआईबी नोड्स को स्पष्ट रूप से स्मृति में अखंड होने की आवश्यकता नहीं है। इसलिए, अंतरिक्ष जटिलता भी ओ (डी) तक कम हो जाती है। **गहराई-पहली सर्च की अंतरिक्ष जटिलता ओ (डी) में है।**

गहराई-सीमित सर्च

गहराई-पहली सर्च के साथ समस्या यह है कि सर्च नर्सिंग अनंत शाखा में एसोसिएट से नीचे जाएगी और इसलिए वापस नहीं आएगी। गहराई-सीमित सर्च एक गहराई सीमा लागू करके इस दोष से बचती है जो गहराई पर सर्च को प्रभावी ढंग से समाप्त कर देती है।

गहराई पैरामीटर का चुनाव एक महत्वपूर्ण मुद्दा हो सकता है। जब तक गहराई पैरामीटर के रूप में, मैं 'काफी गहरा' तैयार हूं, तब तक हमारे पास बाहर निकलने पर उत्तर का एहसास करने के लिए निश्चित रूप से मापने की प्रवृत्ति है। इसलिए, यह तब तक पूरा होता है जब तक

और > = डी

(जहां डी है वह समाधान की गहराई है)। यदि यह शर्त पूरी नहीं होती है, तो गहराई प्रतिबंधित सर्च पूरी नहीं होती है।

वही **स्पेस जटिलता** गहराई-सीमित सर्च वर्ग माप के लिए गहराई-पहली सर्च के समान है, अर्थात, **ओ (बीएल)। समय की गुणवत्ता O(bl) है।**

गहराई-पहली सर्च के लाभ

1. ब्रेड्थ फर्स्ट सर्च के बजाय कम समय और स्थान जटिलता.
2. समाधान को स्वीकार किया जाएगा, जबकि अधिक सर्च नहीं की जाएगी।

गहराई के नुकसान - फर्स्ट सर्च

1. यह आवश्यक नहीं है कि यह आपका समाधान प्रदान करेगा।
2. कट-ऑफ गहराई छोटी है इसलिए समय जटिलता अधिक है।

2.3.3 पुनरावृत्त डेप्थ सर्च

जब प्रारंभिक गहराई कट-ऑफ एक होता है, तो यह पूरी तरह से नींव नोड उत्पन्न करता है और इसकी जांच करता है। यदि नींव नोड लक्ष्य नहीं है, तो गहराई कट-ऑफ लगभग दो है और गहराई 2 तक के ट्री को विशिष्ट गहराई 1 सर्च के शिकार के रूप में उत्पन्न किया जाता है। इसी तरह, एक बार जब गहराई कट-ऑफ लगभग मीटर हो जाती है, तो ट्री को गहराई से मीटर तक बनाया जाता है। इसलिए कोई आश्चर्यचकित हो सकता है कि एक पुनरावृत्ति गहन सर्च में, किसी को वर्तमान गहराई कट-ऑफ पर परिधि नोड्स को छोड़कर सभी नोड्स को पुनर्जीवित करने की आवश्यकता है। चूँकि गहराई से उत्पन्न नोड्स की मात्रा, गहराई एच तक पहली सर्च कितनी है?

(ख)$^{h+1}$(बी -1),

पुनरावृत्त गहन सर्च द्वारा असफल खोजों में विस्तारित नोड्स की कुल संख्या क्या होगी?

$$\{1 / (b-1)\} \sum_{h=0}^{(D-1)} (\text{ख})^{h+1} - 1)$$

≅ b(b)^{d} (b-1)^{2}.

एल्गोरिथ्म में अंतिम पास के परिणामस्वरूप गहराई डी पर एक सफल नोड होता है, जिसकी औसत समय जटिलता विशिष्ट गहराई से पहली सर्च द्वारा दी जाती है। **b(b)^{d} + डी) / 2 (बी -1)।**

इस प्रकार कुल औसत समय जटिलता किसके द्वारा दी गई है? **b(b)^{d} (b-1)^{2} + बी (बी)^{d} + डी) / 2 (बी -1)।**

≅ (बी +1) बी$^{d+1}$ / 2 (बी -1)^{2}.

नतीजतन, पुनरावृत्ति गहन सर्च की औसत समय जटिलता का गहराई से पहली सर्च का अनुपात किसके द्वारा दिया गया है?

{(b+1) b^{d+1} / 2 (b -1) } : { b^{2} / 2 (बी -1)$^{d+1}$}

वही प्रतिशोधी सर्च को गहरा करना इसलिए नहीं लेना प्रचुर नियमित समय से परे, तुलनात्मक रूप से तक हर रोज गहराई प्रारंभिक ढूँढ। वही मुफ्त वृद्धि का पूरा गहराई से ट्री प्रारंभिक सर्च, इस प्रकार, है इससे बचा गया प्रतिशोधी मजबूत बनाने। एक उचित एल्गोरिथम प्रोग्राम का प्रतिशोधी गहरा हो रहा है प्रदत्त नीचे

पुनरावृत्ति-गहनीकरण प्रक्रिया

शुरू करना वर्तमान गहराई कटऑफ सेट करें = 1;
स्थान स्टैक में प्रारंभिक नोड, स्टैक-टॉप द्वारा इंगित किया गया;
कहाँ स्टैक। नहीं खाली और यह भी गहराई क्या है? अंदर दी गई गहराई कट-ऑफ क्या है।
शुरू करना पॉप स्टैक आग्रह करना स्टैक-टॉप तत्व;
यदि स्टैक-टॉप घटक = लक्ष्य, वापस आ जाओ और इसे रोकें।
अन्यथा धक्का बच्चों स्टैक में किसी भी क्रम में स्टैक-टॉप का;
अंत में;
गहराई कट-ऑफ में कितनी वृद्धि करें? एक और चरण 2 के माध्यम से दोहराएं;
समाप्ति।
चौड़ाई प्रारंभिकगहराई प्रारंभिक और यह भी प्रतिशोधी सर्च को गहरा करना है जनरेट के लिए समान रूप से उपयोग किया जाता है और एक नजर डालिए प्रकार एल्गोरिदम। फिर भी जबकि चौड़ाई प्रारंभिक ढूँढ जरूरतों एक चरघातांकी परिमाण स्मृति की, गहराई प्रारंभिक ढूँढ चिंताओं स्मृति किसके समानुपाती है? सबसे बड़ा ट्री की गहराई। वही प्रतिशोधी गहराहो रहा है, चल रहा है इसके विपरीत। हाथ का फायदा है बाहर देख रहे हैं गहराई प्रारंभिक तरीके से। परिवेश ट्री की नियंत्रित गहराई।

2.3.4 <u>हिल क्लाइंब सर्च</u>

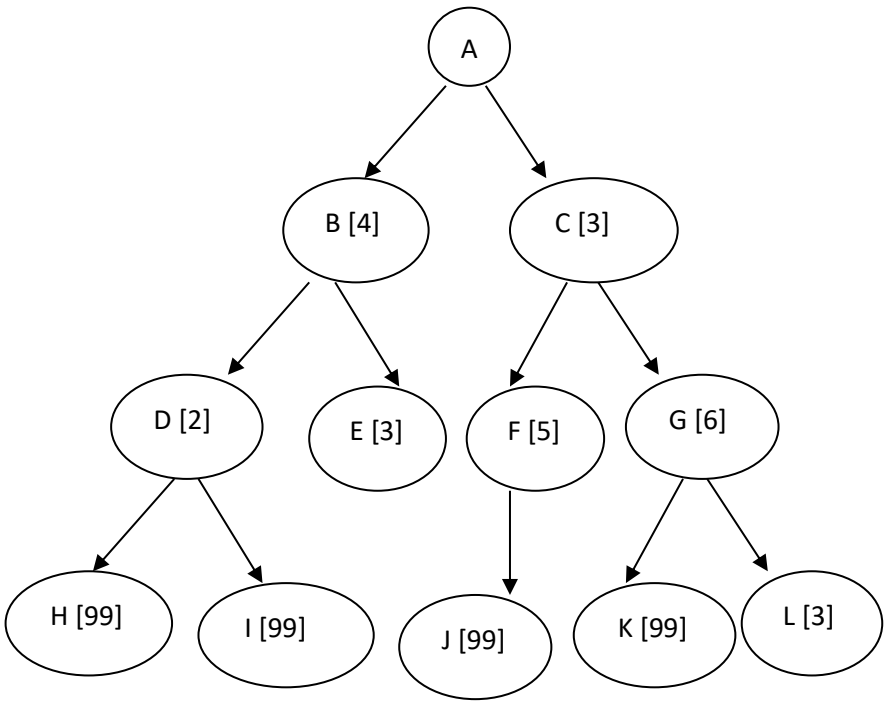
पहाड़ चढ़ाई आमतौर पर प्रयुक्त एक बार एक ईमानदार स्वानुभविक पालन करना पेश किया जाता है। स्टेट के मूल्यांकन के लिए फिर भी एक बार नहीं अलग उपयोगी डेटा पेश किया जाता है।. एक बार पहाड़ी चढ़ाई, एक नोड का पालन करने के लिए चुना गया है जो जारी है कोशिश करने के लिए समग्र प्रारंभिक रीति यद्यपि अनुमानी योग्य नोड की संख्या बच्चे वर्ग माप; सबसे अच्छा से अधिक मूल्य पूर्ववर्ती रूप से नोड्स का दौरा किया हर मंच चयन सबसे प्रभावी एक इस बिंदु तक.

उदाहरण के लिए, पहाड़ी चढ़ाई होगा यात्रा पर लागू विक्रेता समस्याएं. यह है सरल सर्च करने के लिए एसोसिएट डिग्री प्रारंभिक प्रस्ताव जो सभी शहरों का दौरा करता है फिर भी आर बहुत गरीबों की तुलना में गरीब सर्वोत्तम प्रस्ताव. वही नियम ऐसे शुरू करें एक जवाब और बनाता है छोटा एन्हांसमेंट्स उस पर, पसंद परिवर्तन आदेश जिसके भीतर दो शहरों वर्ग माप; दौरा। अंत में एक तरह से छोटा मार्ग शायद जा रहा है प्राप्त किया जाना है।

उदाहरण: इसके बारे में सोचो आकृति में ट्री । यहाँ हेरिस्टिक है वही का अनुमान अंतरिक्ष एक लक्ष्य के लिए, इसलिए कम योग्य उच्च।. मूल्य हेरिस्टिक के बारे में। पालन करना भीतर दिया गया है
वही कोष्ठक।

पहाड़ी चढ़ाई:

→ A से शुरू करें
→ A के बच्चे = [B(4), C(3)]
→ सर्वोत्तम बच्चा C(3)
→c का बच्चा = [F(5), G(6)]
→सर्वोत्तम बच्चा F(5)
→ f के बच्चे = [J(99)] – लक्ष्य नहीं
→ g से c तक बैक ट्रैक करें
→ g के बच्चे = [K(99), L(3)]
→सर्वोत्तम बच्चा L→ लक्ष्य है



चित्र 2.3: हिल क्लाइम्बिंग में आशाजनक नोड्स की सर्च करें

वेरिएंट:

सरल हिल राइजिंग

एल्गोरिदम: सरल हिल राइजिंग

प्रारंभिक स्थिति: दिए गए आरंभिक समस्या के लिए एक शुरुआती हल स्थिति का चयन करें।

प्रक्रिया:

1. मूल समस्या के लिए एक मूल्यांकन फ़ंक्शन का निर्धारण करें। इस फ़ंक्शन का उपयोग करके आपकोडेड समाधान को एक समर्थकता स्कोर दिया जाएगा।
2. एक प्रारंभिक समाधान चयन करें और उसे मूल्यांकन फ़ंक्शन पर प्रभावित करें।
3. समर्थकता स्कोर की प्राप्ति करें। यदि समर्थकता स्कोर सीमा मान से ऊपर है, तो वर्तमान समाधान को स्वीकार करें और प्रक्रिया को समाप्त करें।
4. अन्यथा, नए समाधान के लिए एक ग्राडियेंट या अद्यतित करने वाली विधि का उपयोग करें।
5. समाधान को अद्यतित करें और 2 कदम से वापस जाएं।

यह प्रक्रिया समर्थकता स्कोर की सीमा तक जारी रखेगी और समाधान को धीमे-धीमे समस्या के सटीक समाधान की ओर ले जाएगी। एल्गोरिदम को अच्छे समाधान तक पहुंचने के लिए पुनरावृत्ति किया जाता है।

स्टीपेस्ट-आसेंट हिल क्लाइंबिंग

स्टीपेस्ट-आसेंट हिल क्लाइंबिंग एक आपूर्ति-बहाव एल्गोरिदम है जिसका उपयोग उच्च स्तरीय समस्याओं के समाधान के लिए किया जाता है। यह एक आपूर्ति-बहाव तकनीक है जो समस्या के दीर्घायु समाधान के प्रत्येक पथ का मूल्यांकन करती है और उच्चतम मूल्यांकन स्कोर वाले पथ को चुनती है।

इस एल्गोरिदम में, प्रारंभिक समाधान का चयन किया जाता है और उसे मूल्यांकन फ़ंक्शन पर प्रभावित किया जाता है। फिर, सभी संभावित नवीनतम या अद्यतित समाधानों की समीक्षा की जाती है और उनके लिए समर्थकता स्कोर की प्राप्ति की जाती है। अगले कदम में, सभी संभावित समाधानों के बीच से सबसे उच्च समर्थकता स्कोर वाला समाधान चुना जाता है। चयनित समाधान को अद्यतित किया जाता है और प्रक्रिया दोहराई जाती है, जहां नए समाधान के लिए मूल्यांकन फ़ंक्शन का फिर से प्रयोग किया जाता है।

यह प्रक्रिया समाधान को स्वर्गीय समाधान तक पहुंचने में मदद करती है, जहां समर्थकता स्कोर को सर्वोच्च सीमा तक ले जाया जाता है। इसका उद्देश्य समस्या के निर्धारित की ओर आगे बढ़ना होता है, जिससे समस्या का समाधान प्राप्त हो सके।

स्टीपेस्ट-आसेंट हिल क्लाइंबिंग एल्गोरिदम निरंतर पुनरावृत्ति और मूल्यांकन के माध्यम से समस्या के उत्तम समाधान की सर्च करने में मदद करता है। इसका उपयोग किसी भी संदर्भ में समस्याओं के समाधान के लिए किया जा सकता है, जहां समस्या स्थानिक चर में नियमित होती है और उच्च स्तरीय समाधान चाहिए।

एल्गोरिदम: Steepest–Ascent Hill Climbing (स्टीपेस्ट-आसेंट हिल क्लाइंबिंग)
प्रारंभिक स्थिति: दिए गए आरंभिक समस्या के लिए एक शुरुआती हल स्थिति का चयन करें।
प्रक्रिया:

1. मूल समस्या के लिए एक मूल्यांकन फ़ंक्शन का निर्धारण करें। इस फ़ंक्शन का उपयोग करके आपकोडेड समाधान को एक समर्थकता स्कोर दिया जाएगा।
2. एक प्रारंभिक समाधान चयन करें और उसे मूल्यांकन फ़ंक्शन पर प्रभावित करें।
3. समर्थकता स्कोर की प्राप्ति करें। यदि समर्थकता स्कोर सीमा मान से ऊपर है, तो वर्तमान समाधान को स्वीकार करें और प्रक्रिया को समाप्त करें।
4. अन्यथा, सभी संभावित नवीनतम या अद्यतित समाधानों की समीक्षा करें।
5. सभी संभावित समाधानों के लिए समर्थकता स्कोर की प्राप्ति करें।
6. सबसे उच्च समर्थकता स्कोर वाले समाधान को चुनें। यदि ऐसा समाधान एकाधिक है, तो कोई एक विकल्प चुनें।
7. चयनित समाधान को अद्यतित करें और 4 कदम से वापस जाएं।

यह प्रक्रिया समर्थकता स्कोर की सीमा तक जारी रखेगी और समाधान को धीमे-धीमे समस्या के सटीक समाधान की ओर ले जाएगी। एल्गोरिदम को अच्छे समाधान तक पहुंचने के लिए पुनरावृत्ति किया जाता है।

स्टोकेस्टिक पहाड़ी चढ़ाई

स्टोकास्टिक हिल क्लाइंबिंग एक आपूर्ति-बहाव एल्गोरिदम है जिसका उपयोग समस्याओं के समाधान के लिए किया जाता है। यह एक आपूर्ति-बहाव तकनीक है जो एक मूल्यांकन स्कोर का उपयोग करके समाधान की गुणवत्ता को सुधारने की कोशिश करती है।
स्टोकास्टिक हिल क्लाइंबिंग में, प्रारंभिक समाधान का चयन किया जाता है और उसे मूल्यांकन स्कोर के लिए प्रभावित किया जाता है। फिर, समाधान के पास उपलब्ध सभी संभावित नवीनतम समाधानों की समीक्षा की जाती है।
स्टोकास्टिक तत्व का इस्तेमाल करते हुए, चयनित समाधानों में से एक का रंडम चयन किया जाता है। यह चयन प्रमुखतः तत्विक होता है, जिससे एक समाधान की प्राथमिकता निर्धारित की जाती है।
चयनित समाधान को अद्यतित करें और मूल्यांकन स्कोर की प्राप्ति करें। अगर अद्यतित समाधान का मूल्यांकन स्कोर सीमा से अधिक है, तो उसे स्वीकार किया जाता है और प्रक्रिया समाप्त होती है। अन्यथा, पुनरावृत्ति की प्रक्रिया जारी रहती है, जहां नए समाधान के लिए मूल्यांकन स्कोर का फिर से प्राप्ति की जाती है।
स्टोकास्टिक हिल क्लाइंबिंग एल्गोरिदम रंगीन तत्वों का उपयोग करके एक समाधान के समाधान की सर्च करता है। इस तरीके में, विभिन्न समाधानों को चालू रखने और समस्या के समाधान की गुणवत्ता

को सुधारने का प्रयास किया जाता है। स्टोकास्टिक हिल क्लाइंबिंग का उपयोग विभिन्न उद्देश्यों के लिए किया जा सकता है, जैसे ऑप्टिमाइज़ेशन, मशीन लर्निंग, और अन्य संदर्भों में।

स्टोकास्टिक हिल क्लाइंबिंग का उदाहरण:

चलिए एक उदाहरण के माध्यम से स्टोकास्टिक हिल क्लाइंबिंग को समझते हैं। सोचें कि हमें एक प्रोब्लेम है जहां हम एक संख्या का उचित मान ढूंढ़ना चाहते हैं। हम यह निश्चित करना चाहते हैं कि उस संख्या का मान क्या है जिसके लिए हमारा मूल्यांकन स्कोर सबसे अधिक होगा।

1. प्रारंभिक चरण में, हम एक रैंडम संख्या का उचित मान चुनते हैं।
2. चयनित मान के लिए मूल्यांकन स्कोर की प्राप्ति करें। यह स्कोर हमारी मान्यता फ़ंक्शन द्वारा निर्धारित होता है।
3. एक नया रैंडम मान चुनें और उसके लिए मूल्यांकन स्कोर प्राप्त करें।
4. यदि नया मान का मूल्यांकन स्कोर पहले के मान से अधिक है, तो नया मान को स्वीकार करें और प्रक्रिया को समाप्त करें।
5. अन्यथा, पुनरावृत्ति की प्रक्रिया जारी रखें और चयनित मान को नया मान बनाएं।

इस प्रक्रिया को कई बार दोहराने से संभवतः हम उच्च स्कोर वाले मान तक पहुँच सकते हैं। स्टोकास्टिक हिल क्लाइंबिंग आपूर्ति-बहाव एल्गोरिदम होता है क्योंकि इसमें हम प्रत्येक चरण में एक यादृच्छिकता का उपयोग करते हैं, जो समस्या के समाधान की गुणवत्ता को सुधारने में मदद करता है।

एल्गोरिदम: स्टोकास्टिक हिल क्लाइंबिंग

प्रारंभिक स्थिति: दिए गए आरंभिक समस्या के लिए एक शुरुआती हल स्थिति का चयन करें।
प्रक्रिया:

1. मूल समस्या के लिए एक मूल्यांकन फ़ंक्शन का निर्धारण करें। इस फ़ंक्शन का उपयोग करके आपकोडेड समाधान को एक समर्थकता स्कोर दिया जाएगा।
2. एक प्रारंभिक समाधान चयन करें और उसे मूल्यांकन फ़ंक्शन पर प्रभावित करें।
3. समर्थकता स्कोर की प्राप्ति करें। यदि समर्थकता स्कोर सीमा मान से ऊपर है, तो वर्तमान समाधान को स्वीकार करें और प्रक्रिया को समाप्त करें।
4. अन्यथा, नए समाधान के लिए एक संभावित मान चुनें। इसमें रैंडम शामिल होगी।
5. नए समाधान को मूल्यांकन फ़ंक्शन पर प्रभावित करें और समर्थकता स्कोर प्राप्त करें।
6. पुराने समाधान और नए समाधान के समर्थकता स्कोर की तुलना करें। यदि नए समाधान का स्कोर अधिक है, तो नए समाधान को स्वीकार करें।
7. चयनित समाधान को अद्यतित करें और 4 कदम से वापस जाएं।

यह प्रक्रिया रैंडम का उपयोग करके समस्या के समाधान की ओर प्रगति करेगी। एल्गोरिदम को अच्छे समाधान तक पहुँचने के लिए पुनरावृत्ति किया जाता है। इस तकनीक का उपयोग समस्या के उच्चतम समर्थकता स्कोर को प्राप्त करने के लिए किया जाता है।

रैंडम-रिस्टार्ट हिल क्लाइंबिंग या शॉटगन हिल क्लाइंबिंग

रैंडम-रिस्टार्ट हिल क्लाइंबिंग, जिसे शॉटगन हिल क्लाइंबिंग भी कहा जाता है, एक आपूर्ति-बहाव एल्गोरिदम है जिसका उपयोग बड़े स्थानिक मैक्सिमम समस्याओं के समाधान के लिए किया जाता है। यह एक आपूर्ति-बहाव तकनीक है जो अनुकरण और नवीनतम समाधानों की एक संख्या बनाकर विचार करती है।
इस तकनीक में, अनुकरण के लिए एक संख्या चुना जाता है और उसे मूल्यांकन फ़ंक्शन पर प्रभावित किया जाता है।

1. यदि मूल्यांकन स्कोर सीमा से ऊपर है, तो अनुकरण को स्वीकार किया जाता है और प्रक्रिया समाप्त हो जाती है।
2. अन्यथा, एक नवीनतम समाधान चुना जाता है और उसे मूल्यांकन फ़ंक्शन पर प्रभावित किया जाता है।
3. मूल्यांकन स्कोर की प्राप्ति की जाती है।
4. यदि नवीनतम समाधान का मूल्यांकन स्कोर सीमा से ऊपर है, तो नवीनतम समाधान को स्वीकार किया जाता है और प्रक्रिया समाप्त हो जाती है।
5. अन्यथा, पुनरावृत्ति की प्रक्रिया जारी रखें और नवीनतम समाधान को अनुकरण बनाएं।

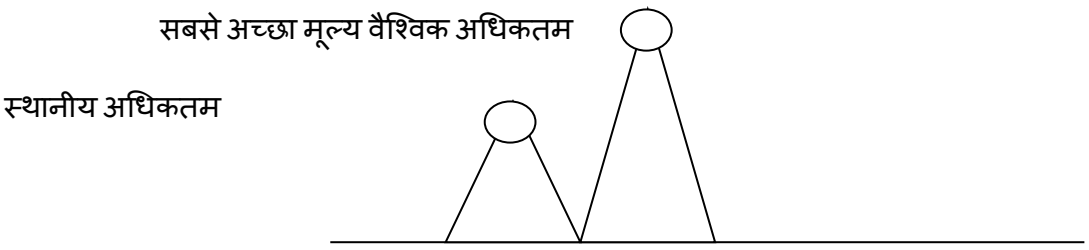
यदि रैंडम-रिस्टार्ट हिल क्लाइंबिंग एल्गोरिदम सक्षम समाधान को प्राप्त नहीं करता है, तो प्रक्रिया को पुनः प्रारंभ करने के लिए नए शुरुआती स्थिति से दोहराया जाता है।
रैंडम-रिस्टार्ट हिल क्लाइंबिंग एल्गोरिदम एक वैकल्पिक संगठन की प्राथमिकता देता है, जिसमें कुछ समाधान नहीं पाए जाते हैं, लेकिन प्रायः अधिक समय नहीं लगता है। यह अल्गोरिदम विशेष रूप से संख्या की सर्च, प्रूफ ऑफ कंसेप्ट, और अन्य संदर्भों में उपयोगी हो सकता है।

हिल क्लाइंबिंग की समस्याएँ

हिल क्लाइंबिंग एक आपूर्ति-बहाव एल्गोरिदम है जो संभावित समाधानों की सर्च करता है और उच्चतम समाधान तक पहुंचने का प्रयास करता है। हालांकि, हिल क्लाइंबिंग के साथ कुछ सामान्य समस्याएं जुड़ी हो सकती हैं। यहां कुछ मुख्य समस्याओं के उदाहरण हैं:
1. लोकल मैक्सिमम पर अटक जाना: हिल क्लाइंबिंग उच्चतम समाधान तक पहुंचने का प्रयास करता है, लेकिन कभी-कभी यह लोकल मैक्सिमम पर अटक जा सकता है। इसका कारण है कि हिल क्लाइंबिंग

केवल प्रत्येक चरण में सबसे अच्छे आपूर्तियों का चयन करता है और यह उसे सुधारने की क्षमता खो सकता है।

उदाहरण: सोचें कि आप एक पहाड़ी पर हैं और उच्चतम शिखर को पहुंचने का प्रयास कर रहे हैं। आपके पास कई रास्ते हैं, लेकिन हर बार आपका पथ उच्चतम बिंदु पर नहीं पहुंचता है, बल्कि आप किसी निकटवर्ती शिखर पर अटक जाते हैं।



चित्र 2.4: स्थानीय अधिकतम का उदाहरण

2. **समाधान की प्रारंभिक स्थिति:** हिल क्लाइंबिंग का परिणाम प्रारंभिक समाधान के प्रभाव पर निर्भर करता है। यदि प्रारंभिक समाधान गलत या अच्छा नहीं है, तो हिल क्लाइंबिंग समाधान की गुणवत्ता को सुधारने में असमर्थ हो सकता है।

उदाहरण: आपको एक विद्यालय में बिल्डिंग का सबसे ऊँचा मंजिल सर्चना है। अगर आप गलत तरीके से पहले मंजिल का चयन करते हैं, जो आपको कम ऊँचाई पर ले जाता है, तो हिल क्लाइंबिंग आपको उच्चतम मंजिल तक पहुंचने में असफल होगी।

3. **खरंगी या उच्चतम समाधान:** हिल क्लाइंबिंग में खरंगी या उच्चतम समाधान की समस्या हो सकती है। इसका मतलब है कि हिल क्लाइंबिंग समाधान नहीं पाएगा जो सभी संभावित समाधानों की तुलना में बेहतर हो।

उदाहरण: आप एक खेल में उच्चतम स्कोर प्राप्त करने का प्रयास कर रहे हैं। यदि आप केवल लगातार एक ही स्थानिक स्थान पर कदम बढ़ाते रहते हैं और किसी खरंगी को पास नहीं करते हैं, तो आप उच्चतम स्कोर प्राप्त करने में असमर्थ हो सकते हैं।

हिल क्लाइंबिंग के साथ संबंधित इन समस्याओं को समझना महत्वपूर्ण है ताकि हम इस तकनीक को उच्चतम समाधान तक पहुंचाने में सफल हो सकें।

हिल क्लाइंबिंग की समस्याओं को कैसे पार करें:

1. **रिस्टार्ट टेकनीक:** एक सामान्य तकनीक है कि हम रिस्टार्ट टेकनीक का उपयोग करके हिल क्लाइंबिंग की समस्याओं को पार कर सकते हैं। इसके लिए, हम प्रारंभिक समाधान को रिस्टार्ट करके नए शुरुआती स्थिति से प्रारंभ करते हैं, ताकि हम उच्चतम समाधान तक पहुंचने की कोशिश कर सकें। यह हमें लोकल मैक्सिमम पर अटकने से बचाता है और समाधान की गुणवत्ता को सुधारने का मौका देता है।

2. **स्थिरता समस्याओं को उपेक्षा करें:** हिल क्लाइंबिंग के दौरान, कुछ स्थिरता समस्याएं आ जाती हैं जो हमें उच्चतम समाधान तक पहुंचने में बाधा डाल सकती हैं। ऐसे समस्याओं को उपेक्षा करें और अल्गोरिदम को संक्षिप्त करें ताकि हम स्थानिक निष्क्रियता को पार कर सकें।
3. **बदलती प्रारंभिक स्थिति:** हिल क्लाइंबिंग की समस्याओं को पार करने का एक तरीका है प्रारंभिक स्थिति को बदलना। यदि हमें प्रारंभिक समाधान से संबंधित समस्या है, तो हमें एक नया और उपयुक्त प्रारंभिक समाधान चुनना चाहिए जो हमें उच्चतम समाधान तक पहुंचाने में मदद कर सके।
4. **मेटाहीलिंग तकनीक:** मेटाहीलिंग एक तकनीक है जिसमें हम अलग-अलग तकनीकों का उपयोग करके हिल क्लाइंबिंग की समस्याओं को पार करते हैं। हम इसके लिए विभिन्न स्थानिक आपूर्तियों के साथ प्रयोग करके बेहतर समाधान तक पहुंचने की कोशिश करते हैं। इससे हमें विभिन्न खरंगियों को छोड़कर बेहतर समाधान मिल सकता है।
5. **एकीकृत करना:** हिल क्लाइंबिंग के समस्याओं को पार करने के लिए एकीकृत तकनीकों का उपयोग करें। इसमें कई तकनीकों को संयोजित करना शामिल होता है जैसे कि रैंडम-रिस्टार्ट, स्टोकास्टिक हिल क्लाइंबिंग, या सिमुलेटेड एनीलिंग। इससे हमें बेहतर समाधान तक पहुंचने की और मदद मिलती है।

ये कुछ तकनीकें हैं जो हिल क्लाइंबिंग की समस्याओं को पार करने के लिए उपयोगी हो सकती हैं। हालांकि, इन तकनीकों का उपयोग समस्या के प्रकार और परियोजना की आवश्यकताओं के अनुसार विचार करना चाहिए।

2.3.5 <u>सिम्युलेटेड एनीलिंग</u>

सिम्युलेटेड एनीलिंग एक प्रोबेबिलिस्टिक आपूर्ति-बहाव एल्गोरिदम है जिसका उपयोग कम से कम एक भूकंप के बाद उच्चतम समाधान की सर्च में किया जाता है। यह एक प्रकार का मैटलर्जीकल प्रक्रिया है जो उच्चतम समाधान की ओर समय के साथ प्रगति करती है। इसका नाम मैटलर्जीकल प्रक्रिया से प्राप्त हुआ है, जिसमें धातु को धीरे-धीरे ठंडा किया जाता है ताकि उसकी रचनात्मकता सुधार सके।
सिम्युलेटेड एनीलिंग अपने नाम के अनुसार विज्ञानिक प्रक्रिया से प्रभावित हुआ है जो शुरुआत में एक उच्च तापमान पर रखी गई चीज़ को धीरे-धीरे ठंडा करके उसकी अणुओं को स्थिर करती है।
सिम्युलेटेड एनीलिंग एक विभाजन और परिखा तकनीक है, जिसमें समाधान के स्थानिक स्थानों के आसपास घूमने के माध्यम से एक गड्ढा ढूंढ़ा जाता है। यह आपूर्ति-बहाव पदार्थ के तापमान को उच्चतम समाधान की तुलना में बढ़ाने के लिए उपयोग करता है। जैसे कि धातु को ठंडा करने से उसकी रचनात्मकता में सुधार होती है, वैसे ही सिम्युलेटेड एनीलिंग में भी समाधान की गुणवत्ता को सुधारा जाता है।

इस तकनीक में, एक तापमान शेड्यूल निर्धारित किया जाता है जो समय के साथ बदलता है। शुरुआती में तापमान ऊँचा होता है और धीरे-धीरे तापमान कम होता है ताकि प्रक्रिया समय के साथ ठंडा हो सके। इसके दौरान, उच्चतम समाधान के लिए स्थानिक स्थानों की चयनितता पर प्राधिकार रखा जाता है, जो कि गड्ढा के आसपास आपूर्तियों को सर्चने की अनुमति देता है। यदि कोई बेहतर समाधान मिलता है, तो उसे स्वीकार किया जाता है, वरना प्राथमिकता दी जाती है।
सिम्युलेटेड एनीलिंग विभिन्न व्यापारिक, अणुसंचार, विज्ञान, और इंजीनियरिंग क्षेत्रों में उपयोग होता है। इसका मुख्य लाभ यह है कि यह उच्चतम समाधान की सर्च में अधिकतम तापमान का उपयोग करता है और विभिन्न समस्याओं को समाधान करने में मदद करता है जहां दूसरे आपूर्ति-बहाव एल्गोरिदम कठिनाईयों का सामना करते हैं।

2.4 सारांश

आर्टिफीसियल इंटेलिजेंस या आर्टिफिशियल इंटेलिजेंस में, समस्या हल करने के लिए इंटेलिजेंसमान सर्च विधियों का उपयोग किया जाता है। ये विधियाँ समस्याओं को अद्यतित करने, विभिन्न विकल्पों को मूल्यांकन करने और सर्वोत्तम समाधान की सर्च में मदद करती हैं। यहां इंटेलिजेंट सर्च के कुछ महत्वपूर्ण पहलुओं का सारांश है:

1. सर्च एल्गोरिदम: अलग-अलग प्रकार के सर्च एल्गोरिदम, जैसे कि प्रतिष्ठित ब्रीद्सडथ, प्रतिष्ठित सिम्युलेटेड एनीलिंग और प्रतिष्ठित जेनेटिक एल्गोरिदम, का उपयोग किया जाता है। ये एल्गोरिदम समस्या के आधार पर चरणों को प्रगतिशील रूप से अद्यतित करते हैं और उच्चतम समाधान की सर्च में मदद करते हैं।
2. प्रारंभिक स्थिति: समस्या का उपयुक्त प्रारंभिक स्थिति का चयन करना महत्वपूर्ण है। एक सही और उपयुक्त प्रारंभिक समाधान विधियाँ को समस्या के विश्लेषण में मदद करती हैं और सर्वोत्तम समाधान की ओर दिशा निर्देश करती हैं।
3. विभिन्न तकनीकों का उपयोग: इंटेलिजेंसमान सर्च में विभिन्न तकनीकों का उपयोग किया जाता है जैसे कि गहराई सर्च, आर्की, जेनेटिक एल्गोरिदम, स्वार्थी सर्च, और स्थैतिक और गतिशील निर्धारण। ये तकनीकें समस्या के स्वरूप और परियोजना की मांगों के आधार पर चयनित की जाती हैं।
4. मानकीकरण और मूल्यांकन: इंटेलिजेंट सर्च में मानकीकरण और मूल्यांकन का महत्वपूर्ण स्थान है। समस्या के अलग-अलग विकल्पों को मूल्यांकित करके, उनकी उपयुक्तता को मापकर और मान्यता से चुनकर हम सर्वोत्तम समाधान की सर्च कर सकते हैं।

इंटेलिजेंसमान सर्च विधियाँ आर्टिफिशियल इंटेलिजेंस के विभिन्न क्षेत्रों में उपयोगी हैं जैसे कि मशीन लर्निंग, डेटा माइनिंग, रोबोटिक्स, और व्यापारिक समस्याएं समाधान करने के लिए। ये विधियाँ समस्याओं को नवीनतम और सर्वोत्तम समाधानों की ओर ले जाने में मदद करती हैं।

प्रमुख शब्द:

- ग्राफ़
- ट्री
- ढूँढ
- कलन विधि
- सूचित सर्च
- बिना सूचना के सर्च
- अंधा
- चौड़ाई सबसे पहले
- गहराई सबसे पहले
- अन्तरिक्ष
- समय
- सबसे खराब
- जटिलता
- गहराई सीमित
- चलने
- पहाड़ी पर चढ़ना
- सबसे तेज चढ़ाई - चढ़ाई
- स्थानीय अधिकतम
- पठार
- कटक
- एनीलिंग

अभ्यास

1. **बहुत छोटे प्रकार के प्रश्न (2 अंक):**

   a) इंटेलीजेंट सर्च क्या है?

   b) हिल क्लाइंबिंग क्या है?

   c) ब्रेथ-फर्स्ट सर्च क्या होता है?

   d) ब्रेथ-फर्स्ट सर्च के लाभ क्या हैं?

   e) ब्रेथ-फर्स्ट सर्च के दुष्प्रभाव क्या हो सकते हैं?

   f) डेप्थ-फर्स्ट सर्च क्या होता है?

   g) डेप्थ-फर्स्ट सर्च के लाभ क्या हैं?

   h) डेप्थ-फर्स्ट सर्च कहां उपयोगी होता है?

   i) हिल क्लाइंबिंग कहां उपयोगी होता है?

   j) इटेरेटिव डीपनिंग सर्च क्या होता है?

   k) स्टीपेस्ट हिल क्लाइंबिंग क्या होता है?

2. लघु प्रकार के प्रश्न: [5 अंक]

a) डेप्थ-फर्स्ट सर्च कैसे काम करता है?

b) ब्रेथ-फर्स्ट सर्च कैसे काम करता है?

c) डेप्थ-फर्स्ट सर्च की सीमाएं क्या हैं?

d) ब्रेथ--फर्स्ट सर्च की सीमाएं क्या हैं?

e) सिंपल हिल क्लाइंबिंग एल्गोरिदम के बारे में बताएं।

f) हिल क्लाइंबिंग कैसे काम करता है?

g) हिल क्लाइंबिंग के लाभ क्या हैं?

h) स्टीपेस्ट हिल क्लाइंबिंग कैसे काम करता है?

i) इटेरेटिव डीपनिंग सर्च कैसे काम करता है?

3. दीर्घ प्रकार के प्रश्न: [10 अंक]

a) निम्नलिखित खोज तकनीक की चर्चा करें और उदाहरण की मदद से उनके लाभ और कमियों पर चर्चा करें।

I. ब्रेड्थ फर्स्ट सर्च (BFS)

II. डेप्थ फर्स्ट सर्च (DFS)

b) एक ऐसी समस्या का उदाहरण दें जिसमें ब्रेड्थ फर्स्ट सर्च डेप्थ फर्स्ट सर्च से बेहतर काम करेगा।

c) स्टीपेस्ट हिल क्लाइंबिंग के एल्गोरिदम की व्याख्या करें।

d) डेप्थ-फर्स्ट सर्च और हिल क्लाइंबिंग के बीच अंतर स्पष्ट करें?
e) डेप्थ-फर्स्ट सर्च और ब्रेथ--फर्स्ट सर्च के बीच अंतर स्पष्ट करें?
f) आर्टिफीसियल इंटेलीजेंट में सिम्युलेटेड एनीलिंग खोज की व्याख्या करें?
g) आर्टिफीसियल इंटेलीजेंट में रैंडम-रीस्टार्ट हिल क्लाइंबिंग के बारे में बताएं?

ग्रन्थसूची

एग्नर, एम., कॉम्बिनेटोरियल सर्च, जॉन विले एंड संस, न्यूयॉर्क, 1988।

बर्र, ए. और फीगेनबाम, ई., द हैंडबुक ऑफ़ आर्टिफिशियल इंटेलिजेंस, खंड 1. विलियम कॉफ़मैन इंक., लॉस अल्टोस, कैलिफ़ोर्निया, पृष्ठ 190-191, 1981।

बेंडर, ई., आर्टिफिशियल इंटेलिजेंस में गणितीय तरीके, आईईईई कंप्यूटर साइंस प्रेस, लॉस एलामिटोस, सीए, 1996।

ब्रैडली, एस.पी., हैक्स, ए.सी., और मैग्नंती, टी.एल., एप्लाइड गणितीय प्रोग्रामिंग, एडिसन वेस्ले पब्लिशिंग कंपनी, रीडिंग, मैसाचुसेट्स, 1977।

कैगन, जे. और मिशेल, डब्ल्यू., "ऑप्टिमली डायरेक्टेड शेप जेनरेशन बाई शेप एनीलिंग," पर्यावरण और योजना बी, 1992।

गेरो, जे. और सुडवीक्स, एफ. (संपादक), आर्टिफिशियल इंटेलिजेंस इन डिज़ाइन' 96, क्लूवर एकेडमिक पब्लिशर्स, 1996

हॉलैंड, जे.एच., प्राकृतिक और आर्टिफीसियल प्रणालियों में अनुकूलन, विश्वविद्यालय। मिशिगन प्रेस, एन आर्बर, 1975 [एमआईटी प्रेस संस्करण]

कनाल, एल. और कुमार, वी. (संपादक), सर्च इन आर्टिफिशियल इंटेलिजेंस, स्प्रिंगर वेरलाग, 1988।

लैयर्ड, जे. और नेवेल, ए., ए यूनिवर्सल वीक मेथड, तकनीकी रिपोर्ट संख्या: सीएमयू-सीएस-83-141, कंप्यूटर विज्ञान विभाग, कार्नेगी-मेलन विश्वविद्यालय, पिट्सबर्ग, पीए 15213, 1983।

माइकलेविक, जेड. जेनेटिक एल्गोरिदम + डेटा स्ट्रक्चर्स = इवोल्यूशन प्रोग्राम्स, स्प्रिंगर वेरलाग, 1992।

अध्याय - तीन

इंटेलीजेंट सर्च द्वारा समस्या समाधान - II

3.1 हेरिस्टिक सर्च

ह्यूरिस्टिक खोज एक विधि है जो समस्या समाधान के लिए एक प्राथमिकता आधारित खोज करती है। यह एक अनुकूलन तकनीक है जो किसी समस्या को समाधान करने के लिए योग्य रूप से डिज़ाइन किए गए तरीकों का उपयोग करती है। ह्यूरिस्टिक खोज का मुख्य उद्देश्य समस्या के समाधान तक पहुंचने के लिए सामरिक और सत्यापनीय पथ का चयन करना होता है।
ह्यूरिस्टिक खोज का उपयोग कई विभिन्न एप्लीकेशन में किया जाता है, जैसे कि विज्ञान्य कंप्यूटिंग, ऑप्टिमाइजेशन, निर्माण यंत्र और अतिरिक्त ज्ञान की खोज। इस तकनीक का उपयोग ज्ञान आधारित सिस्टम में सुधार, रोबोटिक्स, रेलवे नेटवर्क नियोजन, विमानन योजना, और खेलों के विकास में भी किया जाता है।
ह्यूरिस्टिक खोज के उदाहरण में शामिल हैं A* खोज, Best-First खोज, एवरेजज़ड खोज, ग्रीड खोज, और सिमुलेटेड एनीलिंग। इन तकनीकों का उपयोग करके, हम संबंधित समस्या के लिए समाधान की खोज करते हैं और बेहतर परिणाम प्राप्त करने के लिए उपयुक्त हैरिस्टिक्स का उपयोग करते हैं।

ह्यूरिस्टिक खोज के अनुप्रयोग और उदाहरणों की कई संभावित विधाएं हैं। यहां कुछ मुख्य एप्लीकेशन और उदाहरण हैं जहां ह्यूरिस्टिक खोज का उपयोग किया जाता है:

1. विज्ञान्य कंप्यूटिंग: ह्यूरिस्टिक खोज का उपयोग कंप्यूटर विज्ञान में अनुकूलन और प्राकृतिक भूमिका मॉडलिंग में किया जाता है। इसमें क्षेत्र संग्रह का उपयोग करके बेहतरीन समाधान प्राप्त करने की कोशिश की जाती है।
2. ऑप्टिमाइजेशन: ह्यूरिस्टिक खोज का उपयोग विभिन्न ऑप्टिमाइजेशन समस्याओं में किया जाता है, जैसे कि विमानन योजना, रेलवे नेटवर्क नियोजन, व्यापार योजना, और उत्पादन नियोजन। यहां हमें समाधान की खोज करने के लिए उपयुक्त हैरिस्टिक्स का उपयोग करना होता है।
3. रोबोटिक्स: ह्यूरिस्टिक खोज रोबोटिक्स में अहम भूमिका निभाता है, जैसे कि रोबोटिक पथ निर्धारण, स्वरूपण, और निर्धारित कार्य पूरा करने के लिए उपयुक्त नये तरीकों का उपयोग करना।

4. खेलों का विकास: ह्यूरिस्टिक खोज विभिन्न वीडियो गेम्स और रियल-टाइम सिमुलेशन में खेलों के विकास में उपयोगी होता है। इसके द्वारा हमें खेल में सबसे संभावित समाधान प्राप्त करने के लिए उपयुक्त कदम लेने की संभावना होती है।
5. ज्ञान आधारित सिस्टम: ह्यूरिस्टिक खोज ज्ञान आधारित सिस्टमों में सुधार करने के लिए उपयोगी होता है। इससे हमें अद्यतन और अनुकूलन करके उच्चतम स्तर के समाधान तक पहुंचने में मदद मिलती है।

निम्नलिखित ह्यूरिस्टिक खोज विधियों की सूची है।

1. बेस्ट फर्स्ट सर्च (Best First Search)
2. ओआर ग्राफ के लिए ह्यूरिस्टिक खोज (Heuristic Search for OR Graphs)
3. इतरत्वे डीपेनिंग एस्टरिक एल्गोरिदम (Iterative Deepening A* Algorithm)
4. एंड-ओआर ग्राफ पर ह्यूरिस्टिक खोज (Heuristic Search on AND-OR Graphs)

ये विभिन्न ह्यूरिस्टिक खोज विधियाँ हैं जो समस्या समाधान में विशेषता के अनुसार उपयोग की जाती हैं।

3.1.1 बेस्ट फर्स्ट सर्च (Best First Search)

बेस्ट फर्स्ट सर्च एक प्रकार की खोज तकनीक है जो सर्वोत्तम समाधान की प्राप्ति के लिए ह्यूरिस्टिक का उपयोग करती है। इस तकनीक में, हम सभी अनुवर्ती स्थितियों को मूल्यांकन करते हैं और सबसे मूल्यांकित स्थिति को प्राथमिकता से खोजते हैं। इसे "बेस्ट फर्स्ट" कहा जाता है क्योंकि यह वर्तमान में सबसे बेहतर मूल्यांकित स्थिति का चयन करता है।

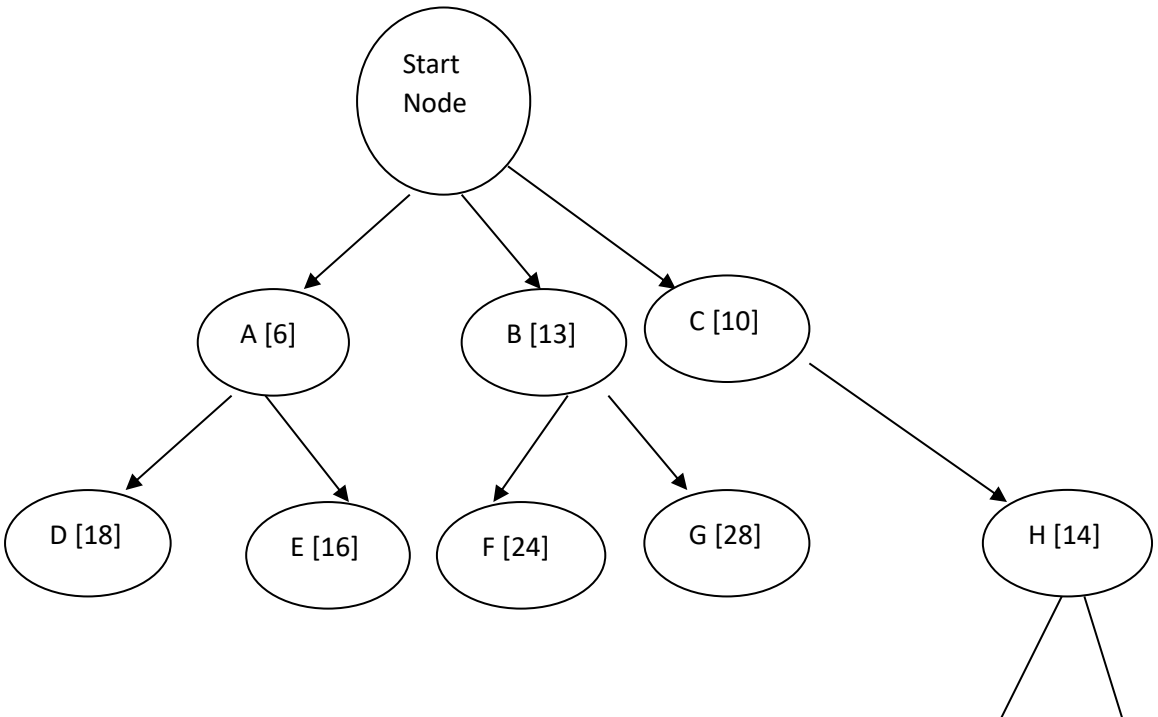
बेस्ट फर्स्ट सर्च का मुख्य उद्देश्य सर्वोत्तम समाधान तक पहुंचना होता है। इसमें हमें सभी संभावित स्थितियों की मूल्यांकन करनी होती है और उन्हें सबसे उच्च मूल्यांकित से नीचे आदेश में खोजना होता है। हम ह्यूरिस्टिक्स का उपयोग करके प्रत्येक स्थिति की मूल्यांकन करते हैं ताकि हम आगे बढ़ने के लिए सबसे प्रमुख स्थिति का चयन कर सकें। यह हमें समय और संसाधनों को बचाने में मदद करता है और अधिकतम मूल्यवान स्थिति के प्राप्ति के लिए अधिक संभावनाएं प्रदान करता है।

बेस्ट फर्स्ट सर्च विभिन्न डोमेन में उपयोगी होता है, जैसे कि निर्माण यंत्र, विमानन योजना, खेलों का विकास, और संख्यात्मक विज्ञान। इसका उपयोग उन समस्याओं के समाधान में किया जाता है जहां हमें सर्वोत्तम समाधान की खोज करनी हो और सामरिक अनुप्रयोगों में अद्यतन और संशोधन करने की आवश्यकता हो।

बेस्ट फर्स्ट सर्च के अनुप्रयोग और उदाहरणों की कई संभावित विधाएं हैं। यहां कुछ मुख्य एप्लीकेशन और उदाहरण हैं जहां बेस्ट फर्स्ट सर्च का उपयोग किया जाता है:

1. विमानन योजना: बेस्ट फर्स्ट सर्च का उपयोग विमानन योजना में फ्लाइट पथ निर्धारण, उड़ान योजना और विमानन नेटवर्क प्लानिंग में किया जाता है। यह सर्वोत्तम फ्लाइट पथ का चयन करने में मदद करता है जो समय, इंजन उपयोग और ईंधन की बचत के साथ सबसे कम अवरोध और सबसे अच्छा प्रदर्शन प्रदान करता है।
2. निर्माण यंत्र: बेस्ट फर्स्ट सर्च का उपयोग निर्माण यंत्र में विभिन्न कार्यों की अनुकूलन और संशोधन के लिए किया जाता है। इससे सर्वोत्तम सामग्री व्यवस्था, मशीनरी निर्माण और कार्यशालाओं की योजना आदि में मदद मिलती है।
3. खेलों का विकास: बेस्ट फर्स्ट सर्च वीडियो गेम्स और रियल-टाइम सिमुलेशन में खेलों के विकास में उपयोगी होता है। यहां हमें अगले कदमों का चयन करने में मदद मिलती है जो सबसे संभावित समाधान प्राप्त करने के लिए उपयुक्त होते हैं और खेल में सबसे उच्च स्तर की प्रदर्शन प्राप्त करने के लिए संभावनाएं प्रदान करती हैं।
4. न्यूरल नेटवर्क्स: बेस्ट फर्स्ट सर्च का उपयोग न्यूरल नेटवर्क्स के प्रशिक्षण और समस्या समाधान में किया जाता है। इससे हमें नेटवर्क के अच्छे और उत्कृष्ट प्रशिक्षण पथ का चयन करने में मदद मिलती है जो अधिकतम निर्धारित लक्ष्य प्राप्ति के लिए सर्वोत्तम रूप से काम करेगा।

ये केवल कुछ उदाहरण हैं, बेस्ट फर्स्ट सर्च के अनुप्रयोग और उदाहरण विभिन्न क्षेत्रों में व्याप्त हैं और विशेषता के आधार पर विविधता देखी जा सकती है।



मूल्यांकन फ़ंक्शन मान लक्ष्य नोड

चित्र 2.7: बेस्ट फर्स्ट सर्च का उदाहरण

क़दम #	नोड का विस्तार किया जा रहा है	बच्चे	उपलब्ध नोड्स	नोड चुना गया
1	Start Node	(a:6), (b:13), (c:10)	(a:6), (b:13), (c:10)	(a:6)
2	A	(d:18), (e:16)	(b:13), (c:10), (d:18), (e:16)	(c:10)
3	C	(h:14)	(b:13), (d:18), (e:16), (h:14)	(b:13)
4	B	(f:24), (g:28)	(d:18), (e:16), (h:14), (f:24), (g:28)	(h:14)
5	H	(i:10), (j:12)	(D:18), (E:16), (F:24), (G:28), (I:10), (J:12)	(i:10)
6	M	(k:2), (l:0), (m:4)	(D:18), (E:16), (H:14), (F:24), (G:28), (J:12), (K:2), (L:0), (M:4)	सर्च रुक जाती है क्योंकि लक्ष्य 1 तक पहुंच गया है

जैसा कि हम देख सकते हैं, बेस्ट फर्स्ट सर्च खोज ग्राफ में "अतिरिक्त उछलना" करता है ताकि न्यूनतम विश्लेषण कार्यक्षमता वाले नोड को पहचाना जा सके। हिल क्लाइंबिंग और बेस्ट फर्स्ट सर्च के बीच केवल एक छोटा अंतर होता है, पहले में, हम पहले उत्पन्न होने वाले नोड के बच्चों को क्रमबद्ध करते हैं, और दूसरे में, हमें पूरी सूची को क्रमबद्ध करके आगे बढ़ने वाले नोड को पहचानना होता है।

एल्गोरिथ्म: बेस्ट फर्स्ट सर्च

बेस्ट फर्स्ट सर्च के लिए निम्नलिखित चरण होते हैं:

1. स्थिति की मूल्यांकन: प्रारंभिक स्थिति को मूल्यांकित करें उपयुक्त हैरिस्टिक या विश्लेषण कार्यक्षमता का उपयोग करके।
2. प्रारंभिक स्थिति को सीमित करें: यदि प्रारंभिक स्थिति समाधान है, तो काम पूरा हो गया है। अन्यथा, आगे बढ़ने के लिए स्थिति को बचाने के लिए एक प्रियतमता सूची बनाएं।
3. बचाने के लिए स्थिति की कतार तैयार करें: सभी संभावित स्थितियों को उनकी प्राथमिकता के आधार पर क्रमबद्ध करें।
4. नवीनतम स्थिति का चयन करें: प्रियतम सूची से प्राथमिकता वाली सबसे नवीन स्थिति का चयन करें।
5. स्थिति की विश्लेषण: चयनित स्थिति को विश्लेषण करें और उससे नये स्थितियों को उत्पन्न करें।
6. समाधान की जांच: क्या चयनित स्थिति समाधान है या नहीं, यदि हां, तो काम पूरा हो गया है। अन्यथा, प्राथमिकता सूची में आगे बढ़ने के लिए नए स्थितियों को जोड़ें।
7. पुनरावृत्ति: 2 से 6 चरणों को पुनरावृत्ति करें जब तक समाधान नहीं मिल जाता है या सीमा को पूरा नहीं किया जाता है।

यह चरण सभी बेस्ट फर्स्ट सर्च एल्गोरिदम में समान रूप से होते हैं, हालांकि विशेषता और परिस्थितियों के आधार पर चरणों में बदलाव हो सकता है।

3.1.2 पुनरावृत्ति गहरा ए * एल्गोरिथ्म।

इतरत्वे डीपेनिंग ए* एल्गोरिदम एक खोज एल्गोरिदम है जो मुख्य लक्ष्य के प्राप्ति के लिए A* एल्गोरिदम का उपयोग करता है, लेकिन यह गहराई प्रतिबंध की सीमा के साथ काम करता है। इस एल्गोरिदम में, हम प्रारंभिक गहराई सीमा के साथ A* खोज करते हैं, और फिर इसे गहराई सीमा को बढ़ाने के लिए बार-बार पुनरावृत्ति करते हैं जब तक समाधान नहीं मिल जाता है।

इतरत्वे डीपेनिंग ए* एल्गोरिदम का मुख्य लक्ष्य है कि यह पूरी समस्या गहराई के बिना हल कर सके, इसलिए यह ज्यादातर सामरिक और अवधारणात्मक समस्याओं के लिए उपयुक्त होता है जहां समाधान की गहराई पर विवेचना करना महत्वपूर्ण होता है।

इतरत्वे डीपेनिंग ए* एल्गोरिदम के चरणों में इस तरह की क्रियाएं शामिल हो सकती हैं:

1. आवर्ती के प्रारंभ में गहराई सीमा सेट करें।
2. A* खोज चलाएं जिसमें हम गहराई सीमा के अंदर ही संभव सभी स्थितियों की जांच करें।
3. समाधान मिलने पर उसे लौटाएं।
4. गहराई सीमा बढ़ाएं और चरण 2 और 3 को दोहराएं।

5. समाधान मिलने या गहराई सीमा के अधीन आने तक चरण 2 से 4 को पुनरावृत्ति करें।

इतरत्वे डीपेनिंग ए* एल्गोरिदम के द्वारा, हमें समाधान की गहराई तक पहुंचने का लाभ मिलता है और यह सुनिश्चित करता है कि हमें न केवल सबसे संभावित समाधान मिलता है, बल्कि उसकी गहराई भी अधिकतम होती है।

इतरत्वे डीपेनिंग ए* एल्गोरिदम के अभ्यास और उदाहरणों की कई संभावित विधाएं हैं। यहां कुछ मुख्य उदाहरण और एप्लिकेशन हैं:

1. चेस: इतरत्वे डीपेनिंग ए* एल्गोरिदम चेस के खेल के समाधान में उपयोगी होता है। यह खेल की पोजीशन के ब्रांचिंग फैक्टर को नियंत्रित करने में मदद करता है और उच्चतम संभावित समाधान की प्राप्ति के लिए स्थानीय गहराई का विचार करता है।
2. अभियांत्रिकी: इतरत्वे डीपेनिंग ए* एल्गोरिदम अभियांत्रिकी के कई अनुप्रयोगों में उपयोगी होता है। इसका उपयोग नक्शा तैयार करने, क्षेत्रों के अध्ययन और निर्माण कार्यों की योजना बनाने में किया जाता है। इससे समाधान की गहराई पर विचार करते हुए सबसे अच्छा संभावित समाधान प्राप्त करने में मदद मिलती है।
3. रोबोटिक्स: इतरत्वे डीपेनिंग ए* एल्गोरिदम रोबोटिक्स में पथ निर्धारण और स्थानांतरण के लिए उपयोगी होता है। इसका उपयोग रोबोट को बाधाओं को पार करने और सबसे अच्छा रास्ता चुनने में मदद करता है जिससे संसाधनों की बचत होती है और नजदीकी अच्छी संभावित समाधान प्राप्त होता है।
4. स्वरूपरेखा निर्धारण: इतरत्वे डीपेनिंग ए* एल्गोरिदम स्वरूपरेखा निर्धारण के लिए उपयोगी होता है, जहां हमें सबसे अच्छी स्वरूपरेखा का चयन करने की जरूरत होती है। इसका उपयोग बनावटी डिजाइन, विकास और विमानन निर्माण में किया जाता है ताकि सबसे उच्च स्थानीय गहराई वाली स्वरूपरेखा चुनी जा सके।

ये केवल कुछ उदाहरण हैं, इतरत्वे डीपेनिंग ए* एल्गोरिदम के उदाहरण और अनुप्रयोगों की विभिन्न विधाएं विभिन्न क्षेत्रों में व्याप्त हो सकती हैं और उनके आधार पर विविधता देखी जा सकती है।

3.1.3 और-या ग्राफ़ पर हेरिस्टिक सर्च

एंड-और ग्राफ पर ह्यूरिस्टिक खोज एक खोज तकनीक है जो एंड-और ग्राफ की समस्याओं के समाधान के लिए ह्यूरिस्टिक्स का उपयोग करती है। एंड-और ग्राफ विशेष रूप से निर्णय विज्ञान और आर्टिफीसियल इंटेलिजेंस क्षेत्र में उपयोगी होते हैं, जहां निर्णय लेने और कार्रवाई करने के लिए एक विकल्पों का समाधान करना होता है।

इस तकनीक में, हम एंड-और ग्राफ की स्थितियों को मूल्यांकन करते हैं और उन्हें उच्चतम मूल्यांकित से नीचे आदेश में खोजते हैं। हमें हर स्थिति पर ह्यूरिस्टिक्स का उपयोग करके मूल्यांकन करना होता है, जिससे हम आगे बढ़ने के लिए सबसे प्रमुख स्थिति का चयन कर सकें। यह हमें समय और संसाधनों

को बचाने में मदद करता है और अधिकतम मूल्यवान स्थिति के प्राप्ति के लिए अधिक संभावनाएं प्रदान करता है।

एंड-और ग्राफ पर ह्यूरिस्टिक खोज के लाभों में शामिल हैं:

- इसका उपयोग अधिकतम संभावित समाधान की प्राप्ति के लिए किया जा सकता है।
- इसे विभिन्न विधाओं में उपयोगी होता है, जैसे कि निर्णय विज्ञान, आर्टिफीसियल इंटेलिजेंस, विमानन योजना, और संख्यात्मक विज्ञान।
- यह संभावनाएं प्रदान करता है जो सबसे उच्च मूल्यांकित स्थिति के प्राप्ति के लिए उपयुक्त होती हैं।

एंड-और ग्राफ पर ह्यूरिस्टिक खोज के उदाहरणों में शामिल हो सकते हैं:

- वाहन निर्देशिका प्रणाली: यह उदाहरण आपको सबसे संभावित यात्रा मार्ग का चयन करने में मदद करता है, जो आपको सबसे कम समय में गंतव्य तक पहुंचाता है।
- नविकरण सिस्टम: यह उदाहरण आपको सबसे अच्छा और सरल नेविगेशन पथ का चयन करने में मदद करता है, जिससे आप अनुकूल यात्रा कर सकते हैं।
- समय-बनावटी निर्माण: यह उदाहरण आपको समय के प्रत्येक अनुभाग को सबसे प्रभावी रूप से नियंत्रित करने में मदद करता है, जो आपको उचित रूप से संसाधनों का उपयोग करने की अनुमति देता है।

एंड-और ग्राफ पर ह्यूरिस्टिक खोज एक व्यापक खोज तकनीक है जिसे विभिन्न क्षेत्रों में समस्याओं के समाधान के लिए उपयोग किया जा सकता है। यह हमें सबसे उच्च मूल्यवान स्थिति की प्राप्ति के लिए उपयुक्त स्थानीय गहराई की प्राथमिकता के साथ काम करने में मदद करता है।

एंड-और ग्राफ पर ह्यूरिस्टिक खोज एल्गोरिदम के चरणों को निम्नलिखित तरीके से व्यक्त किया जा सकता है:

1. प्रारंभिक स्थिति तय करें: शुरुआती स्थिति को निर्धारित करें और उसे मूल्यांकन करें उपयुक्त ह्यूरिस्टिक्स का उपयोग करके।
2. स्थिति का मूल्यांकन करें: वर्तमान स्थिति को मूल्यांकित करें और उसे उच्चतम मूल्यांकित से नीचे क्रमबद्ध करें।
3. स्थितियों का चयन करें: प्राथमिकता क्रमबद्ध स्थितियों के आधार पर एंड और और स्थितियों को चयनित करें।
4. स्थिति के नए उत्पादन करें: चयनित स्थितियों को विश्लेषण करें और नए स्थितियों को उत्पन्न करें।
5. समाधान की जांच करें: क्या चयनित स्थिति समाधान है या नहीं, यदि हां, तो काम पूरा हो गया है। अन्यथा, नए स्थितियों को प्राथमिकता क्रमबद्ध सूची में जोड़ें और चरण 3 से 5 को दोहराएं।
6. सीमा पर पहुंचने या समाधान मिलने तक चरण 3 से 5 को पुनरावृत्ति करें।

यह चरण एंड-और ग्राफ पर ह्यूरिस्टिक खोज एल्गोरिदम के बुनियादी चरण हैं, हालांकि विशेषताएं और अनुभव के आधार पर चरणों में बदलाव किया जा सकता है।

एंड-और ग्राफ पर ह्यूरिस्टिक खोज का उदाहरण

एंड-और ग्राफ पर ह्यूरिस्टिक खोज को एक उदाहरण के साथ समझाते हैं। यहां एक प्राचीन मेयर्स माजिक स्क्वेयर पजल खेल का उदाहरण है जिसमें एंड-और ग्राफ पर ह्यूरिस्टिक खोज का उपयोग किया जा सकता है।
मायर्स माजिक स्क्वेयर पजल एक चार चौकोर ग्रिड होता है, जिसमें हर ग्रिड क्षेत्र में एक प्राकृतिक संख्या होती है। उदाहरण के लिए, हमें निम्नलिखित पजल खेल को हल करना है:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 -
हर एक ग्रिड क्षेत्र में एक संख्या होती है और रिक्त स्थान (-) होता है। खेल का लक्ष्य है कि हमें संख्याओं को सही क्रम में सजाना होता है।
एंड-और ग्राफ पर ह्यूरिस्टिक खोज का उपयोग करने के लिए हम निम्नलिखित चरणों का पालन कर सकते हैं:

1. प्रारंभिक स्थिति को निर्धारित करें: शुरुआती स्थिति में, यानी शुरुआती पजल स्थिति में आवर्तन करें।
2. स्थिति को मूल्यांकन करें: हर पजल स्थिति को मूल्यांकित करें उपयुक्त ह्यूरिस्टिक्स का उपयोग करके। यह हमें एक स्थिति की गुणवत्ता और मान्यता को मापने में मदद करता है।
3. स्थितियों का चयन करें: उच्चतम मूल्यांकित से नीचे क्रमबद्ध करें और आवर्ती प्राथमिकता आदेश में स्थितियों का चयन करें।
4. स्थिति को विश्लेषण करें: चयनित स्थिति को विश्लेषण करें और नए स्थितियों को उत्पन्न करें।
5. समाधान की जांच करें: क्या चयनित स्थिति समाधान है या नहीं, यदि हां, तो हमने खेल को सही क्रम में सजाया है। अन्यथा, चरण 3 से 5 को दोहराएं और समाधान या सीमा तक पहुंचने तक चरणों को दोहराएं।

इस तरह, एंड-और ग्राफ पर ह्यूरिस्टिक खोज एल्गोरिदम के द्वारा हम आवर्तन करके उचित ग्रिड स्थिति का चयन करते हैं, जो हमें दी गई पजल को सही क्रम में सजाने में मदद करता है।

3.2 विरोधी सर्च (Adversary Search)

विरोधी खोज, कंप्यूटर प्रोग्राम को एक प्रतियोगी खिलाड़ी के साथ प्रतिस्पर्धा करने के लिए एक खोज

तकनीक है। इस तकनीक का उपयोग किया जाता है विशेष रूप से विकासशील गेम खेलों में, जहां कंप्यूटर एक विरोधी खिलाड़ी के खिलाफ खेलता है और विजय प्राप्त करने की कोशिश करता है।
विरोधी खोज के द्वारा कंप्यूटर प्रोग्राम एक विरोधी खिलाड़ी के पोसीशन, कार्रवाई और निर्णयों को विश्लेषण करता है और विभिन्न गतिविधियों के लिए संभावित पथों की खोज करता है। इसका लक्ष्य है अगले सर्वोत्तम निर्णय या पहल को खोजना जिससे कि वह विरोधी खिलाड़ी को प्रतिस्पर्धा में हराने की संभावना बढ़ा सके।
विरोधी खोज के उदाहरणों में शतरंज, टिकटैक्टो, चेकर्स, गो और पोकर जैसे खेल शामिल हो सकते हैं। यह तकनीक न केवल विकसित गेम खेलों में बल्कि विविध विज्ञान क्षेत्रों, वित्तीय प्रणालियों, रणनीतिक निर्माण और रोबोटिक्स में भी उपयोगी होती है। विरोधी खोज तकनीक का उपयोग करके, संघर्ष स्थितियों को बेहतरीन ढंग से प्रबंधित किया जा सकता है और संघर्षों की समाधान के लिए विशेषज्ञता दिखा सकते हैं।

3.2.1 <u>मिनिमैक्स एल्गोरिथ्म</u>

मिनिमैक्स एल्गोरिदम एक खोज तकनीक है जो विकासशील गेम खेलों में उपयोग किया जाता है। इसका उपयोग कंप्यूटर प्रोग्राम द्वारा प्रतिस्पर्धी खिलाड़ी के मुकाबले सर्वोत्तम निर्णयों का खोज करने के लिए किया जाता है।
मिनिमैक्स एल्गोरिदम का मुख्य उद्देश्य होता है संघर्ष स्थितियों को प्रबंधित करना, जहां दो प्रतिस्पर्धी खिलाड़ियों के बीच एक कंप्यूटर के निर्णय लेने की आवश्यकता होती है। एल्गोरिदम का उपयोग एक डेसीजन प्रोसेस में भी किया जा सकता है, जहां विभिन्न निर्णयों की गुणवत्ता का मूल्यांकन करना होता है।
मिनिमैक्स एल्गोरिदम की कार्यप्रणाली निम्नांकित चरणों पर आधारित होती है:

1. प्रारंभिक स्थिति का चयन करें: खेल की शुरुआती स्थिति को चुनें, जहां कंप्यूटर को निर्णय लेना होगा।
2. मूल्यांकन का प्रयास करें: प्रत्येक संघर्ष स्थिति को मूल्यांकित करें, जिसमें कंप्यूटर के निर्णयों का प्रभावी मूल्यांकन किया जाता है।
3. विभाजन या खोज: कंप्यूटर और प्रतिस्पर्धी खिलाड़ी के लिए संभावित पथों की खोज करें, जहां कंप्यूटर की प्राथमिकता के आधार पर उच्चतम मूल्यांकित स्थिति का चयन किया जाता है।
4. मूल्यांकन लागू करें: प्रत्येक संभावित पथ को मूल्यांकित करें और सर्वोत्तम निर्णय के लिए मिनिमम या मैक्सिमम मूल्यांकित स्थिति का चयन करें।
5. सर्वोत्तम निर्णय चुनें: संभावित पथों के मूल्यांकन के आधार पर सर्वोत्तम निर्णय चुनें, जो कंप्यूटर के लिए सर्वोत्तम परिणाम देने की संभावना होती है।

3.2.2 <u>अल्फा-बीटा कटऑफ प्रक्रिया</u>

आल्फा-बीटा कटऑफ प्रक्रिया एक खोज तकनीक है जो मिनिमैक्स एल्गोरिदम के साथ उपयोग किया जाता है। यह तकनीक खोज प्रक्रिया को प्रभावी ढंग से संक्षिप्त करने का उद्देश्य रखती है और निर्णय लेने की क्षमता को बढ़ाती है। आल्फा-बीटा कटऑफ प्रक्रिया का उपयोग किया जाता है जब कंप्यूटर प्रोग्राम विरोधी खिलाड़ी के मुकाबले खुद के निर्णय को बनाने के लिए विभिन्न मूल्यांकन नोडों का खोज करता है।

आल्फा-बीटा कटऑफ प्रक्रिया में निम्नलिखित चरण होते हैं:

1. प्रारंभिक स्थिति को निर्धारित करें: खेल की शुरुआती स्थिति को चुनें, जिसमें कंप्यूटर को निर्णय लेना होता है।
2. आल्फा और बीटा मान सेट करें: आल्फा और बीटा मानों को प्राथमिक रूप से सेट करें। आल्फा मान आरंभिक रूप से असीमित होता है और बीटा मान आरंभिक रूप से असीमित होता है।
3. मूल्यांकन नोडों का खोज करें: कंप्यूटर और प्रतिस्पर्धी खिलाड़ी के लिए संभावित पथों की खोज करें।
4. आल्फा-बीटा कटऑफ नियम लागू करें: मूल्यांकित नोडों के आधार पर, आल्फा और बीटा मानों की जांच करें। अगर किसी नोड का मूल्यांकन आल्फा मान से ऊपर जाता है, तो आल्फा मान अपडेट करें। अगर किसी नोड का मूल्यांकन बीटा मान से नीचे जाता है, तो बीटा मान अपडेट करें।
5. प्रूनिंग करें: आल्फा और बीटा मानों के आधार पर कुछ नोडों को प्रून करें, जो कंप्यूटर के लिए अनुचित होते हैं। यदि आल्फा मान बीटा मान से बड़ा होता है, तो विचारशील खोज को प्रभावी ढंग से कटऑफ किया जा सकता है।
6. सर्वोत्तम निर्णय चुनें: संभावित पथों के मूल्यांकन के आधार पर सर्वोत्तम निर्णय चुनें, जो कंप्यूटर के लिए सर्वोत्तम परिणाम देने की संभावना होती है।

आल्फा-बीटा कटऑफ प्रक्रिया का उपयोग मिनिमैक्स एल्गोरिदम के साथ खेल खेलने और निर्णय लेने की गतिविधियों को संक्षिप्त करने के लिए किया जाता है। यह प्रक्रिया कंप्यूटर को संघर्ष स्थितियों को प्रबंधित करने में मदद करती है और निर्णय लेने की क्षमता को बढ़ाती है।

3.2.3 - पजल समस्या:

वही आठ पहेलियों में आठ क्रमांकित, जंगम टाइलें सेट शामिल हैं। बहुत ही में 3x3 फ्रेम. फ्रेम का एक सेल अक्सर होता है खाली इसलिए बनाने यह प्राप्य पैंतरेबाज़ी करने के लिए। साथी खाली सेल में आसन्न क्रमांकित टाइल। इस तरह की एक पहेली निम्नलिखित आरेख में चित्रित की गई है।

1	2	3

4	8	-
7	6	5

स्थिति प्रारंभ करें

1	2	3
4	5	6
7	8	-

लक्ष्य की स्थिति

केवल चलना = "नीचे", "ऊपर", "बाएं" और "दाएं"

प्रारंभ स्थिति = {(1, 2, 3), (4, 8, -), (7, 6, 5)}

नीचे जाना = {(1, 2, 3), (4, 8, 5), (7, 6, -)}

1	2	3
4	8	5
7	6	-

बाएं चलना = { (1,2,3), (4, 8, 5), (7, -, 6)}

1	2	3
4	8	5
7	-	6

ऊपर बढ़ें = { (1,2,3), (4, -, 5), (7,8, 6)}

1	2	3
4	-	5
7	8	6

दायें चलना = { (1,2,3), (4, 5, -), (7,8, 6)}

1	2	3

4	5	-
7	8	6

नीचे जाना = { (1,2,3), (4, 5,6), (7,8, 6)}

1	2	3
4	5	6
7	8	-

कुल पथ लागत = 5 (नीचे, बाएं, ऊपर, दाएं, नीचे)

3.3 सारांश:

हेरिस्टिक सर्च को सूचित सर्च के रूप में भी जाना जाता है। एक हेरिस्टिक सर्च के पीछे विचार यह है कि हम उस नोड का पता लगाते हैं जो संभवतः लक्ष्य स्थिति के सबसे करीब है। सबसे अच्छी पहली सर्च में, एक चाल चुनी जाती है, हालांकि अन्य वर्ग माप अखंड होते हैं, इसलिए यदि चुनाव पथ कम आशाजनक हो जाता है तो उन्हें बाद में फिर से देखा जाएगा। सबसे अच्छा पहला सर्च एल्गोरिथ्म सूत्र 2 सूचियों को बनाए रखता है, एक में अभी भी तलाशने के लिए उम्मीदवारों की एक सूची होती है (ओपन), और एक में विज़िट किए गए नोड्स (बंद) की एक सूची होती है। ए * एल्गोरिदम का उद्देश्य किसी दिए गए प्रारंभिक नोड से विशेष लक्ष्य तक कम से कम लागत वाले पथ की सर्च करना है। दूसरा शास्त्रीय नकारात्मक पक्ष, जहां भी हेरिस्टिक सर्च लागू होती है, यह है कि एएनडी-ओआर ग्राफ़ पर लागू पिछड़ा तर्क नकारात्मक पक्ष है। मिनिमैक्स फॉर्मूला किसी दिए गए स्टेट से स्टेट संक्रमण के पूर्ण जोखिम पर विचार करता है और इसके परिणामस्वरूप पूरे घर को कवर करता है। अल्फा बीटा छंटाई प्रतिकूल सर्च की एक तकनीक है जो मिनिमैक्स रणनीति में खोजे गए नोड्स की संख्या को कम करती है। आठ पहेलियों में आठ क्रमांकित, मूवेबल टाइलें होती हैं जो एक बहुत ही 3x3 फ्रेम में सेट होती हैं और केवल "नीचे", "ऊपर", "बाएं" और "दाएं" चलती हैं।

प्रमुख शब्द:

- स्वानुभविक
- सबसे अच्छा पहला
- और-या
- A * एल्गोरिथ्म।
- लक्ष्य
- गांठ
- प्रतिकूल
- मिनिमैक्स
- अल्फा-बीटा कटऑफ
- 8 पजल
- स्टेट

अभ्यास

1. बहुत संक्षिप्त प्रकार के प्रश्न:[2 अंक]

a) उदाहरण के साथ AO * एल्गोरिथ्म की व्याख्या करें।]

b) उदाहरण के साथ AO * एल्गोरिथ्म की व्याख्या करें।

c) बेस्ट फर्स्ट सर्च का वर्णन करें।

d) स्थानीय न्यूनतम समस्या क्या है?

e) बिना सूचना वाली सर्च रणनीतियों की व्याख्या करें.

f) खेल सर्च के साथ स्थानीय सर्च को अलग करें।

g) खेल में सर्च के उपयोग को सही ठहराएं।

h) हेरिस्टिक फ़ंक्शन से आपका क्या मतलब है?

i) क्या मिनिमैक्स प्रक्रिया तीन से चार खिलाड़ियों के लिए उपयुक्त है? अपने जवाब को सही ठहराएं।

j) अल्फा बीटा छंटाई तकनीक कैसे काम करती है? समझाना

2. लघु प्रकार के प्रश्न: [5 अंक]

a) सर्च तकनीक का मूल्यांकन करने के लिए विभिन्न मापदंडों का उपयोग क्या किया जाता है?

b) A* सर्च तकनीक का वर्णन करें। साबित करें कि A * पूर्ण और इष्टतम है।

c) साबित करें कि चौड़ाई पहली सर्च समान लागत सर्च का विशेष मामला है।

d) हिल क्लाइम्बिंग से जुड़ी विभिन्न समस्याएं क्या हैं? उनका संभावित समाधान बताएं।

e) आप एओ * सर्च में हल किए गए या अनसुलझे नोड को कैसे लेबल करते हैं?

f) क्यों हेरिस्टिक सर्च अंधा सर्च से बेहतर है?

g) गेम ट्री का एक उदाहरण दें। एक गेम ट्री में MINMAX प्रक्रिया का उद्देश्य क्या है?

h) AO* सर्च तकनीक का वर्णन करें।

i) संपूर्ण सर्च पर इंटेलिजेंसमान सर्च के पेशेवरों और विपक्ष क्या हैं, प्रत्येक के लिए दो उदाहरण दें?

j) AND-OR ग्राफ़ के संबंध में समस्या न्यूनीकरण की व्याख्या कीजिए।

k) हिल क्लाइम्बिंग एल्गोरिथ्म के बारे में इसकी कमी के साथ समझाएं और इसे कैसे दूर किया जा सकता है?

l) एक उदाहरण के साथ सर्वश्रेष्ठ पहली सर्च एल्गोरिथ्म की व्याख्या करें।

m) हेरिस्टिक फ़ंक्शन क्या है? ब्लाइंड सर्च और हेरिस्टिक सर्च रणनीतियों के बीच अंतर क्या है?

n) हिल क्लाइम्बिंग एल्गोरिथ्म की समस्याओं पर चर्चा करें?

o) एओ * एल्गोरिदम को चक्रों वाले ग्राफ़ पर ठीक से काम क्यों करना चाहिए, आपके उत्तर को सही ठहराता है।

p) 8 पहेली समस्या के आधार पर इंटरैक्टिव समझाएं।

q) मिनिमैक्स सर्च प्रक्रिया क्या है? अल्फा बीटा छंटाई को एक उपयुक्त उदाहरण के साथ समझाएं।

3. दीर्घ प्रकार के प्रश्न: [10 अंक]

a) निम्नलिखित मानदंडों के आधार पर किसी भी दो अज्ञात सर्च रणनीतियों की तुलना करें: पूर्णता, अंतरिक्ष जटिलता और इष्टतमता।

b) दिखाएँ कि DFS न तो पूर्ण है और न ही इष्टतम है।

c) हिल क्लाइंब सर्च की व्याख्या करें। हिल क्लाइम्बिंग में पठार और रिज की भूमिका क्या है?

d) एक उपयुक्त उदाहरण पर विचार करके स्थानीय सर्च एल्गोरिथ्म की व्याख्या करें।

e) स्टेट अंतरिक्ष सर्च के रूप में किसी समस्या को कैसे परिभाषित करें? एक उदाहरण की सहायता से चर्चा कीजिए।

f) बीएफएस और डीएफएस सर्च तकनीक को विस्तार से समझाएं। उपयुक्त उदाहरण के साथ एक * सर्च तकनीक का वर्णन करें।

g) शुरुआती नोड से एक लक्ष्य नोड का पता लगाने के लिए सबसे तेज चढ़ाई हिल क्लाइम्बिंग तकनीक के लिए एल्गोरिदम लिखें। क्या यह हमेशा एक लक्ष्य पाता है?

h) निम्नलिखित कथनों में से प्रत्येक को सिद्ध कीजिए, या एक प्रतिउदाहरण दीजिए।

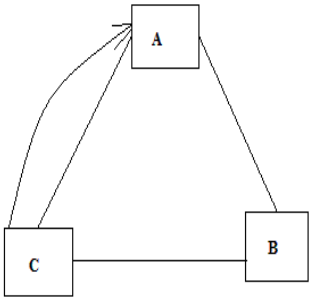
(i) चौड़ाई - पहली सर्च एक समान लागत वाली सर्च का एक विशेष मामला है।

(ii) गहराई - पहली सर्च सर्वश्रेष्ठ-प्रथम ट्री सर्च का एक विशेष मामला है।

(iii) वर्दी - लागत सर्च ए * सर्च का एक विशेष मामला है।

i) हिल क्लाइम्बिंग का उपयोग करके 8 पहेली को हल करने की कोशिश करने पर विचार करें। हेरिस्टिक फ़ंक्शन ढूंढें जो पहेली को काम करता है।

j) वर्णन करें कि चक्र वाले ग्राफ़ पर काम करने के लिए एओ * एल्गोरिथ्म को कैसे संशोधित किया जा सकता है। एल्गोरिथ्म में क्या बदलाव किए जाने चाहिए? सुनिश्चित करें कि नोड सी की लागत बदलने पर आप नीचे चित्र -1 के रूप में दिखाए गए ग्राफ को सही ढंग से संभाल सकते हैं।



ग्रन्थसूची

अग्रे, पी.ई. (1988)।
रोजमर्रा की जिंदगी की गतिशील संरचना।
पीएचडी थीसिस, मैसाचुसेट्स इंस्टीट्यूट ऑफ टेक्नोलॉजी, कैम्ब्रिज, एमए।
एआई-टीआर 1085, एमआईटी आर्टिफीसियल इंटेलिजेंस प्रयोगशाला।
एग्रे, पी.ई. और चैपमैन, डी. (1990)।
रोबोटिक्स और स्वायत्त प्रणालियाँ, 6:17-34।
एल्बस, जे.एस. (1971)।
अनुमस्तिष्क कार्य का एक सिद्धांत.
गणितीय जीवविज्ञान, 10:25-61.
बेयर्ड, एल.सी. (1995)।
अवशिष्ट एल्गोरिदम: फ़ंक्शन सन्निकटन के साथ सुदृढीकरण सीखना।
प्रीडाइटिस में, ए. और रसेल, एस., संपादक, मशीन लर्निंग पर बारहवें अंतर्राष्ट्रीय सम्मेलन की कार्यवाही, पृष्ठ 30-37, सैन फ्रांसिस्को, सीए। मॉर्गन कॉफ़मैन.
बाओ, जी., कैसेंड्रास, सी.जी., जैफेरिस, टी.ई., गांधी, ए.डी., और लूज़, डी.पी. (1994)।
डाउन पीक ट्रैफिक के लिए लिफ्ट डिस्पैचर।
तकनीकी रिपोर्ट, ईसीई विभाग, मैसाचुसेट्स विश्वविद्यालय।
कोलोम्बेट्टी, एम. और डोरिगो, एम. (1994)।

अनुक्रमिक व्यवहार करने के लिए प्रशिक्षण एजेंट।
अनुकूली व्यवहार, 2(3):247-275.
कॉनेल, जे. (1989)।
एक आर्टिफीसियल प्राणी के लिए एक कॉलोनी वास्तुकला।
तकनीकी रिपोर्ट एआई-टीआर-1151, एमआईटी आर्टिफिशियल इंटेलिजेंस प्रयोगशाला, कैम्ब्रिज, एमए।
कनेर्वा, पी. (1993)।
विरल वितरित मेमोरी और संबंधित मॉडल।
हसौं में, एम. एच., संपादक, एसोसिएटिव न्यूरल मेमोरीज़: थ्योरी एंड इम्प्लीमेंटेशन, पृष्ठ 50-76। ऑक्सफोर्ड यूनिवर्सिटी प्रेस, एनवाई।
कश्यप एट अल., 1970
कश्यप, आर.एल., ब्लेडन, सी.सी., और फू, के.एस. (1970)।

अध्याय - चार

नॉलेज रिप्रजेंटेशन के लिए संरचित दृष्टिकोण

4.1 परिचय

यह अध्याय संरचित मॉडल द्वारा नॉलेज रिप्रजेंटेशन के लिए एक अलग दृष्टिकोण प्रदान करता है। इन मॉडलों का बाद की गणना ओं द्वारा गैर-संरचित मॉडल पर बड़ा महत्व है। संरचनात्मक मॉडल के कई मॉड्यूल एक ही समय में सक्रिय बनाए जाते हैं, इसलिए तर्क विधि के भीतर भारी समानता के लिए एक गुंजाइश प्रदान करते हैं। तर्क इंजन की समय शक्ति इसलिए संरचित मॉडल के साथ सुधार किया जाता है।

इस अध्याय में, नॉलेज रिप्रजेंटेशन के लिए संरचित मॉडल की विविधता को पंक्तिबद्ध किया जाएगा। इन संरचनाओं में, कुछ घटनाओं के बीच कठोर निर्भरता संबंध हैं। इन संरचित मॉडलों को मजबूत भराव और स्लॉट विधियों के रूप में जाना जाता है। तार्किक निर्भरता और स्क्रिप्ट ऐसे संरचित मॉडल के आदर्श नमूने हैं। संरचनात्मक मॉडल का विपरीत रूप घटनाओं के बीच निर्भरता संबंधों पर प्रचुर मात्रा में प्रतिबंध नहीं लगाता है और इसलिए इसे कमजोर भराव के रूप में जाना जाता है।

4.2. सिमेंटिक नेट

सिमेंटिक नेट, जिसे सिमेंटिक नेटवर्क के रूप में भी जाना जाता है, आर्टिफीसियल इंटेलिजेंस के क्षेत्र में नॉलेज रिप्रजेंटेशन के लिए एक संरचित दृष्टिकोण है। उन्हें 1960 के दशक के अंत में संज्ञानात्मक मनोवैज्ञानिक रॉस क्विलियन द्वारा मानव सिमेंटिक मेमोरी को मॉडल करने के तरीके के रूप में पेश किया गया था।

एक सिमेंटिक नेट एक नेटवर्क या ग्राफ के रूप में ज्ञान का प्रतिनिधित्व करता है, जहां नोड्स अवधारणाओं या वस्तुओं का प्रतिनिधित्व करते हैं, और किनारे उनके बीच संबंधों का प्रतिनिधित्व करते हैं। नेटवर्क में प्रत्येक नोड एक अवधारणा से मेल खाती है, और प्रत्येक किनारा अवधारणाओं के बीच एक विशिष्ट संबंध का प्रतिनिधित्व करता है।

एक सिमेंटिक नेट के मूल तत्व हैं:

1. नोड्स या अवधारणाएं: ये नेटवर्क के निर्माण खंड हैं और व्यक्तिगत अवधारणाओं या वस्तुओं का प्रतिनिधित्व करते हैं। उदाहरण के लिए, "बिल्ली," "कुत्ता," या "कार" एक सिमेंटिक नेट में नोड्स हो सकते हैं।
2. लिंक या रिश्ते: ये नोड्स के बीच कनेक्शन हैं जो अवधारणाओं के बीच संबंधों का प्रतिनिधित्व करते हैं। लिंक को संबंध के प्रकार को निर्दिष्ट करने के लिए लेबल किया जा सकता है। उदाहरण के लिए, "is-a" लेबल वाला एक लिंक "बिल्ली" का प्रतिनिधित्व करने वाले नोड को "जानवर" का प्रतिनिधित्व करने वाले नोड से जोड़ सकता है, यह दर्शाता है कि एक बिल्ली एक प्रकार का जानवर है।
3. विशेषताएँ या गुण: नोड्स में उनके साथ जुड़े गुण या गुण भी हो सकते हैं। ये गुण उन अवधारणाओं के बारे में अतिरिक्त जानकारी प्रदान करते हैं जिनका वे प्रतिनिधित्व करते हैं। उदाहरण के लिए, "कार" का प्रतिनिधित्व करने वाले नोड में "रंग," "बनाओ," या "मॉडल" जैसी विशेषताएं हो सकती हैं।

सिमेंटिक नेट का उपयोग आमतौर पर उन डोमेन में ज्ञान का प्रतिनिधित्व करने के लिए किया जाता है जहां अवधारणाओं के बीच संबंध महत्वपूर्ण हैं। वे ज्ञान को पकड़ने और व्यवस्थित करने के लिए एक ग्राफिकल और सहज ज्ञान युक्त तरीका प्रदान करते हैं। एक सिमेंटिक नेट की संरचना कुशल क्वेरी और अनुमान की अनुमति देती है, क्योंकि अवधारणाओं के बीच संबंधों को आसानी से पार किया जा सकता है।

सिमेंटिक नेट की सीमाओं में से एक उनकी स्केलेबिलिटी और जटिल ज्ञान डोमेन को संभालने में कठिनाई है। जैसे-जैसे अवधारणाओं और रिश्तों की संख्या बढ़ती है, नेटवर्क अनियंत्रित और प्रबंधित करना मुश्किल हो सकता है। हालांकि, इन चुनौतियों का सामना करने के लिए सिमेंटिक नेट के विविधताएं और विस्तार विकसित किए गए हैं, जैसे कि फ्रेम-आधारित सिस्टम और ओन्टोलॉजी।

4.3 सिमेंटिक नेट में वंशानुक्रम: आर्टिफीसियल इंटेलिजेंस में सिमेंटिक नेट का उपयोग करके नॉलेज रिप्रजेंटेशन के संरचित दृष्टिकोण में वंशानुक्रम एक महत्वपूर्ण अवधारणा है। यह नेटवर्क के भीतर अधिक सामान्य अवधारणाओं से अधिक विशिष्ट लोगों तक विशेषताओं और संबंधों के प्रसार की अनुमति देता है।

एक सिमेंटिक नेट में, वंशानुक्रम को आमतौर पर नोड्स के बीच "इस-ए" या "सबक्लास" संबंधों के उपयोग के माध्यम से दर्शाया जाता है। इन संबंधों से संकेत मिलता है कि एक अवधारणा किसी अन्य

अवधारणा का उपप्रकार या विशेषज्ञता है। उदाहरण के लिए, जानवरों का प्रतिनिधित्व करने वाले एक सिमेंटिक नेट में, "पशु" के लिए एक नोड और "बिल्ली," "कुत्ता" और "पक्षी" के लिए सबनोड हो सकते हैं।

जब एक अवधारणा किसी अन्य अवधारणा से विरासत में मिलती है, तो इसका मतलब है कि यह अधिक सामान्य अवधारणा से जुड़े गुणों, विशेषताओं और संबंधों को प्राप्त करता है। उदाहरण के लिए, यदि "पशु" नोड में "4" के मान के साथ "हैसलेग्स" के लिए एक विशेषता है, तो "बिल्ली" और "कुत्ता" नोड्स इस विशेषता और मूल्य को विरासत में प्राप्त करेंगे। इसी तरह, यदि "जानवर" और "निवास स्थान" के बीच कोई संबंध है जो दर्शाता है कि जानवर कहाँ रहते हैं, तो सबनोड "बिल्ली" और "कुत्ते" में भी यह संबंध होगा।

वंशानुक्रम एक पदानुक्रमित तरीके से संगठन और ज्ञान साझा करने की अनुमति देता है। यह अनावश्यक जानकारी से बचने और अमूर्तता के उच्च स्तर पर सामान्य विशेषताओं को कैप्चर करके समान अवधारणाओं के प्रतिनिधित्व को सरल बनाता है। यह विरासत में मिले ज्ञान के आधार पर सामान्यीकरण और कटौती की अनुमति देकर कुशल क्वेरी और तर्क को भी सक्षम बनाता है।

4.4 सिमेंटिक नेट में मोनोटोनिक और डिफ़ॉल्ट इनहेरिटेंस में हेरफेर करना:

सिमेंटिक नेट का उपयोग करके नॉलेज रिप्रजेंटेशन के संरचित दृष्टिकोण में, मोनोटोनिक और डिफ़ॉल्ट वंशानुक्रम में हेरफेर करना ज्ञान के साथ कैप्चर और तर्क के लिए एक महत्वपूर्ण पहलू है। मोनोटोनिक वंशानुक्रम अधिक सामान्य अवधारणाओं से अधिक विशिष्ट अवधारणाओं तक विशेषताओं और संबंधों के सीधे प्रसार को संदर्भित करता है, जबकि डिफ़ॉल्ट वंशानुक्रम अपवादों और डिफ़ॉल्ट मूल्यों को संभालने से संबंधित है।

आइए इन अवधारणाओं का आगे पता लगाएं:

1. मोनोटोनिक वंशानुक्रम: मोनोटोनिक वंशानुक्रम मानता है कि वंशानुक्रम पदानुक्रम सख्ती से पदानुक्रमित है और अपवाद या विरोधाभासों की अनुमति नहीं देता है। इसका मतलब है कि अधिक सामान्य अवधारणा के लिए परिभाषित सभी विशेषताओं और संबंधों को संशोधन के बिना इसकी उप-अवधारणाओं द्वारा विरासत में मिला है। उदाहरण के लिए, यदि अवधारणा "पशु" में "4" के मान के साथ "हैसलेग्स" विशेषता है, तो "बिल्ली" या "कुत्ता" जैसी सभी उपअवधारणाएं इस विशेषता को समान मूल्य के साथ विरासत में प्राप्त करेंगी।
2. डिफ़ॉल्ट वंशानुक्रम: डिफ़ॉल्ट वंशानुक्रम वंशानुक्रम पदानुक्रम के भीतर अपवादों या डिफ़ॉल्ट मानों को संभालने का एक तरीका प्रदान करता है। यह मानता है कि कुछ विशेषताएं या संबंध सभी उप-अवधारणाओं के लिए सार्वभौमिक रूप से नहीं हो सकते हैं। इसके बजाय, ऐसे मामलों को संभालने के लिए डिफ़ॉल्ट मान या अपवाद निर्दिष्ट किए जा सकते हैं। उदाहरण के लिए,

यदि अधिकांश जानवरों के पास "4" के मान के साथ "हैसलेग्स" हैं, लेकिन बिना पैरों वाले "सांप" या "मछली" जैसे कुछ अपवाद हैं, तो उन मामलों के लिए "0" का डिफ़ॉल्ट मान या एक अपवाद नियम परिभाषित किया जा सकता है।

डिफ़ॉल्ट वंशानुक्रम ज्ञान के अधिक लचीले और सूक्ष्म प्रतिनिधित्व की अनुमति देता है। यह वंशानुक्रम पदानुक्रम के भीतर भिन्नताओं और अपवादों को पकड़ने में सक्षम बनाता है और उन मामलों को समायोजित करता है जहां विशिष्ट अवधारणाएं सामान्य पैटर्न से विचलित होती हैं।

सिमेंटिक नेट में मोनोटोनिक और डिफ़ॉल्ट वंशानुक्रम में हेरफेर करने में नेटवर्क के भीतर विशेषताओं, रिश्तों और अपवाद नियमों को परिभाषित और निर्दिष्ट करना शामिल है। यह अतिरिक्त नोड्स या लिंक के माध्यम से प्राप्त किया जा सकता है जो डिफ़ॉल्ट मानों, अपवादों, या विशेष मामलों का प्रतिनिधित्व करते हैं। इन तत्वों को शामिल करके, सिमेंटिक नेट एक अधिक मजबूत प्रतिनिधित्व बन जाता है जो विरासत में मिले ज्ञान में भिन्नताओं और विरोधाभासों को संभाल सकता है।

मोनोटोनिक और डिफ़ॉल्ट वंशानुक्रम से जुड़े सिमेंटिक नेट में तर्क और अनुमान नेटवर्क को पार करने, विशेषताओं और रिश्तों का प्रचार करने और आवश्यकतानुसार अपवादों और डिफ़ॉल्ट मूल्यों पर विचार करने पर निर्भर करते हैं। यह नेटवर्क में कैप्चर किए गए ज्ञान के आधार पर कुशल क्वेरी, कटौती और निर्णय लेने की अनुमति देता है।

यह ध्यान रखना महत्वपूर्ण है कि सिमेंटिक नेट में मोनोटोनिक और डिफ़ॉल्ट वंशानुक्रम में हेरफेर करने के लिए विशिष्ट तंत्र और सम्मेलन विशेष ज्ञान, प्रतिनिधित्व ढांचे या उपयोग की जा रही प्रणाली के आधार पर भिन्न हो सकते हैं। विभिन्न दृष्टिकोण, जैसे फ्रेम-आधारित सिस्टम या ओन्टोलॉजी, वंशानुक्रम और अपवादों को संभालने के लिए अतिरिक्त तंत्र प्रदान कर सकते हैं।

4.5 सिमेंटिक नेट में अव्यावहारिक तर्क:

आर्टिफीसियल इंटेलिजेंस में सिमेंटिक नेट का उपयोग करके नॉलेज रिप्रजेंटेशन के लिए संरचित दृष्टिकोण में अव्यवहार्य तर्क एक महत्वपूर्ण अवधारणा है। यह गैर-मोनोटोनिक या अव्यवहार्य ज्ञान के प्रतिनिधित्व और हैंडलिंग की अनुमति देता है, जिसका अर्थ है कि ज्ञान से निकाले गए निष्कर्षों को नई जानकारी या अपवादों की उपस्थिति में ओवरराइड या संशोधित किया जा सकता है।

सिमेंटिक नेट में, अनिश्चित या संदर्भ-निर्भर ज्ञान को पकड़ने और तर्क करने के लिए अव्यावहारिक तर्क का उपयोग किया जाता है। यह उन नियमों या मान्यताओं के प्रतिनिधित्व की अनुमति देता है जो आम तौर पर सच होते हैं लेकिन विशिष्ट स्थितियों या अपवादों द्वारा ओवरराइड या पराजित किए जा सकते हैं।

सिमेंटिक नेट में अव्यवहार्य तर्क के कुछ प्रमुख पहलू यहां दिए गए हैं:

अव्यवहार्य नियम: अव्यवहार्य नियम ऐसे नियम हैं जो सामान्य पैटर्न या अनुमानों को पकड़ते हैं लेकिन अपवाद या पराजय के अधीन होते हैं। उन्हें सिमेंटिक नेट में नोड्स के बीच संबंधों या लिंक के रूप में दर्शाया जाता है। उदाहरण के लिए, एक अव्यवहार्य नियम यह कह सकता है कि "यदि किसी जानवर के पंख हैं, तो वह उड़ सकता है।

पराजित: पराजित करने वाले जानकारी या शर्तों के टुकड़े हैं जो अव्यवहार्य नियमों से निकाले गए निष्कर्षों को ओवरराइड या पराजित कर सकते हैं। वे अपवाद या विरोधाभासी साक्ष्य का प्रतिनिधित्व करते हैं जो नियम को कमजोर या अमान्य करते हैं। उदाहरण के लिए, यदि सिमेंटिक नेट में एक और नोड है जो "पेंगुइन" का प्रतिनिधित्व करता है और यह नियम से जुड़ा हुआ है, यह दर्शाता है कि पेंगुइन के पंख हैं लेकिन उड़ नहीं सकते हैं, तो यह एक पराजय के रूप में कार्य करता है।

प्रासंगिक अनुमान: अव्यावहारिक तर्क अधिक सूक्ष्म निष्कर्ष बनाने के लिए प्रासंगिक जानकारी और हारने वालों की उपस्थिति को ध्यान में रखता है। यह मानता है कि कुछ नियम सार्वभौमिक रूप से नहीं हो सकते हैं और विशिष्ट संदर्भों या स्थितियों में अतिरंजित हो सकते हैं।

संघर्ष समाधान: जब परस्पर विरोधी या विरोधाभासी अव्यवहार्य नियम या पराजय होते हैं, तो सबसे उपयुक्त निष्कर्ष निर्धारित करने के लिए एक संघर्ष समाधान तंत्र नियोजित किया जाता है। इसमें नियमों की ताकत या विशिष्टता को तौलना, संदर्भ पर विचार करना, या संघर्षों को हल करने के लिए प्राथमिकता या वरीयता रैंकिंग लागू करना शामिल हो सकता है।

अव्यावहारिक तर्क अधिक लचीला और संदर्भ-जागरूक नॉलेज रिप्रजेंटेशन और अनुमान की अनुमति देता है। यह उन स्थितियों को समायोजित करता है जहां सामान्य नियमों से निकाले गए निष्कर्षों को विशिष्ट अपवादों या पराजय के आधार पर संशोधित किया जा सकता है। यह अनिश्चित या अधूरी जानकारी के साथ तर्क का समर्थन करता है और एआई सिस्टम में अधिक परिष्कृत निर्णय लेने में सक्षम बनाता है।

यह ध्यान रखना महत्वपूर्ण है कि सिमेंटिक नेट में अव्यवहार्य तर्क के लिए कार्यान्वयन और तकनीक विशिष्ट नॉलेज रिप्रजेंटेशन ढांचे या उपयोग की जा रही प्रणाली के आधार पर भिन्न हो सकती हैं। विभिन्न दृष्टिकोण, जैसे नियम-आधारित सिस्टम, गैर-मोनोटोनिक लॉजिक्स, या तर्क ढांचे, सिमेंटिक नेट में अव्यवहार्य तर्क को संभालने के लिए अतिरिक्त तंत्र और औपचारिकताएं प्रदान कर सकते हैं।

4.6 फ्रेम: फ्रेम्स आर्टिफीसियल इंटेलिजेंस के क्षेत्र में नॉलेज रिप्रजेंटेशन के लिए एक संरचित दृष्टिकोण है। 1970 के दशक में मार्विन मिंस्की द्वारा पेश किया गया, फ्रेम एक पदानुक्रमित और संरचित तरीके से ज्ञान को व्यवस्थित और प्रतिनिधित्व करने का एक तरीका प्रदान करते हैं।

एक फ्रेम एक वैचारिक इकाई या वस्तु का प्रतिनिधित्व करता है और इसमें स्लॉट और मान होते हैं। प्रत्येक स्लॉट ऑब्जेक्ट की एक विशेषता या गुण का प्रतिनिधित्व करता है, और संबद्ध मान उस

विशेषता के लिए विशिष्ट मान या जानकारी का प्रतिनिधित्व करता है। फ्रेम वस्तुओं की संरचना और गुणों को कैप्चर करते हैं, जिससे ज्ञान के कुशल प्रतिनिधित्व और हेरफेर की अनुमति मिलती है।

नॉलेज रिप्रजेंटेशन के लिए एक संरचित दृष्टिकोण के रूप में फ्रेम के कुछ प्रमुख पहलू यहां दिए गए हैं:

1. पदानुक्रमित संरचना: फ्रेम को एक पदानुक्रमित तरीके से व्यवस्थित किया जाता है, जहां फ्रेम उच्च-स्तरीय फ्रेम से स्लॉट और मान प्राप्त कर सकते हैं। यह संबंधित वस्तुओं में सामान्य विशेषताओं और संबंधों के साझाकरण और प्रचार के लिए अनुमति देता है। फ्रेम में वंशानुक्रम एक डोमेन के भीतर सामान्यीकरण और विशेषज्ञता के प्रतिनिधित्व को सक्षम बनाता है।
2. स्लॉट और मान: स्लॉट एक फ्रेम की विशेषताओं या गुणों को परिभाषित करते हैं। प्रत्येक स्लॉट का एक नाम होता है, जो विशेषता का प्रतिनिधित्व करता है, और इसमें संबंधित मान हो सकते हैं, जो उस विशेषता के लिए विशिष्ट जानकारी या डेटा का प्रतिनिधित्व करते हैं। उदाहरण के लिए, "कार" का प्रतिनिधित्व करने वाले फ्रेम में "बनाओ," "मॉडल" और "रंग" के लिए स्लॉट हो सकते हैं, जिसमें "टोयोटा," "कैमरी" और "लाल" जैसे संबंधित मान हो सकते हैं।
3. डिफ़ॉल्ट मान: फ़्रेम में स्लॉट के लिए डिफ़ॉल्ट मान शामिल हो सकते हैं, जो विशेषताओं के लिए विशिष्ट या डिफ़ॉल्ट जानकारी का प्रतिनिधित्व करते हैं. ये डिफ़ॉल्ट मान तब उपयोग किए जाते हैं जब विशिष्ट मान स्पष्ट रूप से प्रदान नहीं किए जाते हैं। डिफ़ॉल्ट मान अतिरेक को कम करने और डोमेन के भीतर सामान्य पैटर्न कैप्चर करने में मदद करते हैं।
4. वंशानुक्रम और ओवरराइडिंग: फ्रेम वंशानुक्रम का समर्थन करते हैं, जिससे उच्च-स्तरीय फ्रेम से निम्न-स्तरीय फ्रेम तक स्लॉट मूल्यों के प्रसार की अनुमति मिलती है। हालांकि, फ्रेम निचले स्तर पर विरासत में मिले मूल्यों को ओवरराइड करने के लिए लचीलापन भी प्रदान करते हैं, जो अपवादों या विशिष्ट उदाहरणों के प्रतिनिधित्व को सक्षम करते हैं जो सामान्य पैटर्न से विचलित होते हैं।
5. फ़्रेम संबंध: फ़्रेम रिश्तों के माध्यम से अन्य फ़्रेम से संबंधित हो सकते हैं, जैसे कि भाग-संपूर्ण संबंध, पदानुक्रमित संबंध या एसोसिएशन संबंध। ये संबंध फ्रेम के बीच कनेक्शन और निर्भरता को पकड़ते हैं और जटिल ज्ञान संरचनाओं के प्रतिनिधित्व को सक्षम करते हैं।

फ्रेम्स संरचित ज्ञान का प्रतिनिधित्व करने और आर्टिफीसियल इंटेलिजेंस प्रणालियों में तर्क और अनुमान का समर्थन करने का एक शक्तिशाली तरीका प्रदान करते हैं। वे कुशल क्वेरी, हेरफेर और जानकारी की पुनर्प्राप्ति की सुविधा प्रदान करते हैं। फ्रेम्स का व्यापक रूप से विभिन्न एआई अनुप्रयोगों में उपयोग किया गया है, जिसमें विशेषज्ञ प्रणाली, प्राकृतिक भाषा समझ और संज्ञानात्मक मॉडलिंग शामिल हैं।

यह ध्यान रखना महत्वपूर्ण है कि एक ज्ञान डोमेन की विशिष्ट आवश्यकताओं के आधार पर फ्रेम को बढ़ाया और अनुकूलित किया जा सकता है। फ्रेम के विभिन्न विविधताएं और विस्तार, जैसे कि फ्रेम-आधारित भाषाएं, ओन्टोलॉजी, या सिमेंटिक नेटवर्क, विभिन्न एआई संदर्भों में फ्रेम के प्रतिनिधित्व और तर्क क्षमताओं को बढ़ाने के लिए विकसित किए गए हैं।

4.7 उलझे हुए फ्रेम में वंशानुक्रम: उलझे हुए फ्रेम का उपयोग करके नॉलेज रिप्रजेंटेशन, वंशानुक्रम परस्पर जुड़े फ्रेम के बीच विशेषताओं और संबंधों के प्रसार को संदर्भित करता है। उलझे हुए फ्रेम पारंपरिक फ्रेम-आधारित दृष्टिकोण का एक रूपांतर हैं जहां फ्रेम को सख्त पदानुक्रमित संरचना में व्यवस्थित नहीं किया जाता है, बल्कि इसके बजाय परस्पर संबंधों का एक जटिल वेब बनाते हैं।

उलझे हुए फ्रेम में वंशानुक्रम पारंपरिक फ्रेम में पाए जाने वाले पदानुक्रमित वंशानुक्रम से अलग तरह से संचालित होता है। स्पष्ट उपप्रकारों और सुपरटाइप्स के साथ एक स्पष्ट पदानुक्रमित संरचना के बजाय, उलझे हुए फ्रेम कई और अतिव्यापी वंशानुक्रम संबंधों की अनुमति देते हैं। इस दृष्टिकोण में, फ्रेम कई फ्रेम से विशेषताओं और संबंधों को विरासत में ले सकते हैं, और फ्रेम को विभिन्न प्रकार के रिश्तों के माध्यम से एक दूसरे से जोड़ा जा सकता है।

उलझे हुए फ्रेम में विरासत के कुछ प्रमुख पहलू यहां दिए गए हैं:

मल्टीपल इनहेरिटेंस: टैंगल्ड फ्रेम कई वंशानुक्रम का समर्थन करते हैं, जिससे एक फ्रेम को एक साथ कई अन्य फ्रेम से विशेषताओं और रिश्तों को विरासत में लेने की अनुमति मिलती है। यह ज्ञान डोमेन के भीतर वस्तुओं के जटिल संबंधों और विशेषताओं को पकड़ने में अधिक लचीलेपन की अनुमति देता है।

अतिव्यापी वंशानुक्रम: उलझे हुए फ्रेम में, फ्रेम उन विशेषताओं और रिश्तों के संदर्भ में ओवरलैप हो सकते हैं जो उन्हें विरासत में मिलते हैं। इसका मतलब यह है कि कई फ्रेम सामान्य विशेषताओं और संबंधों को साझा कर सकते हैं, जिससे जटिल और अंतर्निहित ज्ञान संरचनाओं का प्रतिनिधित्व होता है।

इंटरकनेक्टेड रिलेशनशिप: उलझे हुए फ्रेम रिश्तों के माध्यम से फ्रेम के परस्पर संबंध पर जोर देते हैं। फ़्रेम को विभिन्न प्रकार के संबंधों के माध्यम से एक साथ जोड़ा जा सकता है, जैसे कि भाग-संपूर्ण संबंध, सहयोगी संबंध, या यहां तक कि अधिक जटिल कनेक्शन। ये संबंध परस्पर जुड़े फ्रेम में ज्ञान के ट्रैवर्सल और अन्वेषण को सक्षम करते हैं।

संदर्भ-विशिष्ट वंशानुक्रम: उलझे हुए फ्रेम में वंशानुक्रम संदर्भ-विशिष्ट हो सकता है, जिससे फ्रेम को विशिष्ट प्रासंगिक स्थितियों या परिदृश्यों के आधार पर विशेषताओं और संबंधों को विरासत में लेने की अनुमति मिलती है। यह ज्ञान के प्रतिनिधित्व को सक्षम करता है जो विशिष्ट संदर्भों या स्थितियों पर निर्भर है।

उलझे हुए फ्रेम में वंशानुक्रम उन डोमेन में ज्ञान का प्रतिनिधित्व करने के लिए एक लचीला और अभिव्यंजक तरीका प्रदान करता है जहां जटिल और अतिव्यापी संबंध मौजूद हैं। यह समृद्ध और परस्पर ज्ञान संरचनाओं को पकड़ने की अनुमति देता है जो पदानुक्रमित संगठन में बड़े करीने से फिट नहीं हो सकते हैं। हालांकि, उलझे हुए फ्रेम के प्रबंधन की जटिलता बढ़ जाती है क्योंकि फ्रेम और रिश्तों

की संख्या बढ़ती है, उलझी हुई संरचना के भीतर क्वेरी और तर्क के लिए कुशल एल्गोरिदम और तंत्र की आवश्यकता होती है।

यह ध्यान रखना महत्वपूर्ण है कि उलझे हुए फ्रेम नॉलेज रिप्रजेंटेशन के लिए फ्रेम-आधारित दृष्टिकोण का सिर्फ एक बदलाव हैं, और विशिष्ट कार्यान्वयन और तकनीक ेंं नॉलेज रिप्रजेंटेशन ढांचे या सिस्टम के आधार पर भिन्न हो सकती हैं।

4.8 पेट्री नेट: पेट्री नेट आर्टिफीसियल इंटेलिजेंस के क्षेत्र में नॉलेज रिप्रजेंटेशन के लिए एक संरचित दृष्टिकोण है, विशेष रूप से मॉडलिंग और समवर्ती प्रणालियों का विश्लेषण करने के क्षेत्र में। हालांकि वे मुख्य रूप से सामान्य नॉलेज रिप्रजेंटेशन के लिए उपयोग नहीं किए जाते हैं, वे व्यापक रूप से मॉडलिंग और प्रक्रियाओं, वर्कफ़्लो और समवर्ती व्यवहार का विश्लेषण करने के लिए नियोजित होते हैं।

पेट्री नेट निर्देशित ग्राफ़ का उपयोग करके सिस्टम का एक ग्राफिकल प्रतिनिधित्व प्रदान करते हैं, जिसमें नोड्स शामिल होते हैं जिन्हें स्थान और संक्रमण कहा जाता है, साथ ही आर्क जो उन्हें जोड़ते हैं। स्थान स्टेट या स्थितियों का प्रतिनिधित्व करते हैं, जबकि संक्रमण उन घटनाओं या कार्यों का प्रतिनिधित्व करते हैं जो हो सकते हैं। आर्क स्थानों और संक्रमणों के बीच टोकन के प्रवाह को परिभाषित करते हैं, जो सिस्टम के भीतर नियंत्रण या डेटा के प्रवाह का प्रतिनिधित्व करते हैं।

मॉडलिंग और विश्लेषण प्रणालियों के लिए एक संरचित दृष्टिकोण के रूप में पेट्री नेट के कुछ प्रमुख पहलू यहां दिए गए हैं:

स्थान: पेट्री नेट में स्थान एक प्रणाली के स्टेट या स्थितियों का प्रतिनिधित्व करते हैं। वे मॉडलकिए जा रहे डोमेन के लिए प्रासंगिक संसाधनों, वस्तुओं की स्थिति, या अधिक अमूर्त स्टेट का प्रतिनिधित्व कर सकते हैं। स्थान टोकन रखते हैं, जो एक इकाई की उपस्थिति या एक शर्त की पूर्ति का संकेत देते हैं।

संक्रमण: पेट्री नेट में संक्रमण उन घटनाओं या कार्यों का प्रतिनिधित्व करते हैं जो सिस्टम में हो सकते हैं। वे प्रक्रियाओं, संचालन, या स्टेट परिवर्तनों का प्रतिनिधित्व कर सकते हैं। संक्रमण तब सक्षम या निकाले जाते हैं जब इनपुट स्थानों में सभी आवश्यक टोकन मौजूद होते हैं।

आर्क: आर्क स्थानों और संक्रमणों को जोड़ते हैं और उनके बीच टोकन के प्रवाह का प्रतिनिधित्व करते हैं। वे स्थानों और संक्रमणों के बीच संबंधों और निर्भरताओं को परिभाषित करते हैं। आर्क या तो इनपुट आर्क, आउटपुट आर्क या दोनों हो सकते हैं, जो संक्रमण के अंदर और बाहर टोकन के प्रवाह को इंगित करते हैं।

टोकन आंदोलन: संक्रमण आग लगने पर टोकन पेट्री नेट के भीतर चलते हैं। एक संक्रमण को फायरिंग करने से इनपुट स्थानों से टोकन की खपत होती है और आउटपुट स्थानों में टोकन का उत्पादन होता है, जो सिस्टम की स्थिति में परिवर्तन या किसी कार्रवाई की घटना को दर्शाता है।

पेट्री नेट सिस्टम का एक संरचित और दृश्य प्रतिनिधित्व प्रदान करते हैं, जिससे उनके व्यवहार, प्रदर्शन और संयोजन के विश्लेषण की अनुमति मिलती है। वे एक सिस्टम के भीतर जटिल इंटरैक्शन, निर्भरता और संघर्षों को पकड़ सकते हैं, जिससे उन्हें मॉडलिंग और विभिन्न प्रक्रियाओं और वर्कफ़्लो को समझने के लिए उपयोगी बना दिया जाता है।

जबकि पेट्री नेट का उपयोग मुख्य रूप से प्रक्रिया मॉडलिंग और विश्लेषण के क्षेत्र में किया गया है, उनके सिद्धांतों और अवधारणाओं ने नॉलेज रिप्रजेंटेशन और आर्टिफीसियल इंटेलिजेंस के अन्य क्षेत्रों को भी प्रभावित किया है। उदाहरण के लिए, पेट्री नेट के वेरिएंट और एक्सटेंशन, जैसे रंगीन पेट्री नेट और उच्च-स्तरीय पेट्री नेट, अधिक जटिल पहलुओं को पकड़ने और उन्नत विश्लेषण तकनीकों का समर्थन करने के लिए विकसित किए गए हैं।

कुल मिलाकर, पेट्री नेट समवर्ती प्रणालियों के मॉडलिंग और विश्लेषण के लिए एक संरचित दृष्टिकोण प्रदान करते हैं, एक ग्राफिकल प्रतिनिधित्व प्रदान करते हैं जो सिस्टम व्यवहार की समझ और मूल्यांकन की सुविधा प्रदान करता है।

4.9 वैचारिक निर्भरता: वैचारिक निर्भरता आर्टिफीसियल इंटेलिजेंस के क्षेत्र में नॉलेज रिप्रजेंटेशन के लिए एक संरचित दृष्टिकोण है। यह 1970 के दशक में रोजर शैंक और उनके सहयोगियों द्वारा प्राकृतिक भाषा वाक्यों के अर्थ का प्रतिनिधित्व करने और प्राकृतिक भाषा समझ का समर्थन करने के साधन के रूप में विकसित किया गया था।

वैचारिक निर्भरता (सीडी) ज्ञान को परस्पर वैचारिक इकाइयों के एक सेट के रूप में दर्शाती है जिसे आदिम कहा जाता है। ये आदिम अवधारणाओं, रिश्तों, कार्यों और घटनाओं का प्रतिनिधित्व करके प्राकृतिक भाषा अभिव्यक्तियों के शब्दार्थ को पकड़ते हैं।

नॉलेज रिप्रजेंटेशन के लिए एक संरचित दृष्टिकोण के रूप में वैचारिक निर्भरता के कुछ प्रमुख पहलू यहां दिए गए हैं:

आदिम: आदिम वैचारिक निर्भरता के बुनियादी निर्माण खंड हैं। वे मौलिक अवधारणाओं, रिश्तों और कार्यों का प्रतिनिधित्व करते हैं जिनका उपयोग अधिक जटिल ज्ञान संरचनाओं के निर्माण के लिए किया जाता है। आदिमों के उदाहरणों में "वस्तु," "एजेंट", और "कार्रवाई" जैसी अवधारणाएं शामिल हैं, साथ ही साथ "पार्ट-ऑफ," "हैस", और "कारण" जैसे संबंध शामिल हैं।

वैचारिक संरचनाएं: वैचारिक निर्भरता प्राकृतिक भाषा वाक्यों के अर्थ का प्रतिनिधित्व करने के लिए वैचारिक संरचनाओं का उपयोग करती है। इन संरचनाओं में रिश्तों से जुड़े आदिम ो के संग्रह शामिल हैं। वैचारिक संरचनाएं अवधारणाओं के बीच संबंधों को पकड़ती हैं और एक वाक्य या प्रवचन की शब्दार्थ सामग्री का प्रतिनिधित्व प्रदान करती हैं।

निर्भरता संबंध: वैचारिक निर्भरता में निर्भरता संबंध अवधारणाओं के बीच संबंधों को व्यक्त करते हैं। वे वर्णन करते हैं कि अवधारणाएं एक-दूसरे से कैसे संबंधित हैं और वे कैसे बातचीत करते हैं। उदाहरण के लिए, एक निर्भरता संबंध एक एजेंट अवधारणा को एक कार्रवाई अवधारणा के साथ जोड़ सकता है ताकि यह इंगित किया जा सके कि एजेंट कार्रवाई करता है।

अनुमान और तर्क: वैचारिक निर्भरता नए ज्ञान को हेरफेर करने और प्राप्त करने के लिए अनुमान नियमों को लागू करके तर्क और अनुमान की अनुमति देती है। आदिम और विशिष्ट अवधारणाओं के गुणों के बीच संबंधों के आधार पर निष्कर्ष निकाले जा सकते हैं।

वैचारिक निर्भरता एक संरचित तरीके से प्राकृतिक भाषा अभिव्यक्तियों के अर्थ के बारे में प्रतिनिधित्व और तर्क करने का एक तरीका प्रदान करती है। इसका उद्देश्य भाषा के शब्दार्थ को पकड़ना और पाठ या बोली जाने वाली भाषा की समझ और व्याख्या का समर्थन करना है। दृष्टिकोण का उपयोग विभिन्न प्राकृतिक भाषा प्रसंस्करण कार्यों में किया गया है, जैसे कि प्रश्न उत्तर, पाठ योग, और मशीन अनुवाद।

जबकि वैचारिक निर्भरता ने प्राकृतिक भाषा की समझ के क्षेत्र में महत्वपूर्ण योगदान दिया है, अन्य नॉलेज रिप्रजेंटेशन दृष्टिकोणों ने हाल के वर्षों में अधिक प्रमुखता प्राप्त की है, जैसे कि सिमेंटिक नेटवर्क, फ्रेम और ओन्टोलॉजी। ये दृष्टिकोण अक्सर ज्ञान के अधिक स्पष्ट और औपचारिक प्रतिनिधित्व प्रदान करते हैं, जो विभिन्न एआई अनुप्रयोगों की जरूरतों को पूरा करते हैं।

4.10 स्क्रिप्ट: स्क्रिप्ट आर्टिफीसियल इंटेलिजेंस के क्षेत्र में नॉलेज रिप्रजेंटेशन के लिए एक संरचित दृष्टिकोण है। 1970 के दशक में रोजर शैंक और उनके सहयोगियों द्वारा घटनाओं, स्थितियों या दिनचर्या के बारे में रूढ़िवादी ज्ञान का प्रतिनिधित्व करने के तरीके के रूप में स्क्रिप्ट पेश की गई थी।

नॉलेज रिप्रजेंटेशन के संदर्भ में, एक स्क्रिप्ट को क्रियाओं या घटनाओं के पूर्वनिर्धारित अनुक्रम के रूप में देखा जा सकता है जो किसी विशेष परिदृश्य या स्थिति से जुड़े होते हैं। स्क्रिप्ट आमतौर पर किसी दिए गए स्थिति में क्या होता है, इसके बारे में ज्ञान को कैप्चर करती है, जिसमें घटनाओं का अपेक्षित अनुक्रम, प्रतिभागियों की भूमिकाएं और विशिष्ट परिणाम शामिल हैं।

नॉलेज रिप्रजेंटेशन के लिए एक संरचित दृष्टिकोण के रूप में स्क्रिप्ट के कुछ प्रमुख पहलू यहां दिए गए हैं:

ईवेंट और क्रियाएँ: स्क्रिप्ट विशिष्ट स्थितियों में होने वाली घटनाओं और क्रियाओं के बारे में ज्ञान का प्रतिनिधित्व करती हैं। घटनाएं प्रतिभागियों द्वारा किए गए कार्य, पर्यावरण में परिवर्तन, या कोई अन्य प्रासंगिक घटनाएं हो सकती हैं। क्रियाएँ विशिष्ट चरणों या व्यवहारों का प्रतिनिधित्व करती हैं जो आमतौर पर किसी विशेष घटना या स्थिति से जुड़ी होती हैं।

भूमिकाएँ और प्रतिभागी: स्क्रिप्ट अक्सर घटनाओं या कार्यों में शामिल भूमिकाओं और प्रतिभागियों को परिभाषित करती हैं। भूमिकाएं किसी दिए गए स्थिति में प्रतिभागियों के अपेक्षित कार्यों या जिम्मेदारियों का प्रतिनिधित्व करती हैं। प्रतिभागी वे संस्थाएं या व्यक्ति हैं जो उन भूमिकाओं को पूरा करते हैं।

अनुक्रम और क्रम: स्क्रिप्ट किसी दिए गए स्थिति में घटनाओं और क्रियाओं के विशिष्ट अनुक्रम या क्रम को कैप्चर करती हैं। वे घटनाओं और कार्यों के अपेक्षित प्रवाह या प्रगति को निर्दिष्ट करते हैं, जो ज्ञान के लौकिक संगठन का प्रतिनिधित्व करते हैं।

पूर्व शर्तें और पोस्टकंडीशंस: स्क्रिप्ट में घटनाओं या क्रियाओं से जुड़ी पूर्व शर्तें और पोस्टकंडीशन भी शामिल हो सकती हैं। पूर्व शर्तें उन शर्तों को परिभाषित करती हैं जिन्हें किसी घटना या कार्रवाई के लिए पूरा किया जाना चाहिए, जबकि पोस्टकंडीशन किसी घटना या कार्रवाई के होने के बाद अपेक्षित स्थिति या परिणाम का वर्णन करते हैं।

स्क्रिप्ट नियमित या रूढ़िवादी स्थितियों के बारे में ज्ञान का एक संरचित प्रतिनिधित्व प्रदान करती हैं। वे विशिष्ट परिदृश्यों या घटनाओं से जुड़े सामान्य पैटर्न और अपेक्षाओं को पकड़ते हैं। स्क्रिप्ट का उपयोग विभिन्न एआई अनुप्रयोगों में समझ, अनुमान और भविष्यवाणी का समर्थन करने के लिए किया जाता है, जैसे कि प्राकृतिक भाषा समझ, प्रश्न उत्तर और योजना।

यह ध्यान रखना महत्वपूर्ण है कि स्क्रिप्ट विशिष्ट स्थितियों या घटनाओं के बारे में ज्ञान का प्रतिनिधित्व करने तक सीमित हैं और वास्तविक दुनिया में होने वाली परिवर्तनशीलता और अपवादों की पूरी श्रृंखला को कैप्चर नहीं कर सकती हैं। अन्य नॉलेज रिप्रजेंटेशन दृष्टिकोण, जैसे कि फ्रेम, ओन्टोलॉजी, या संभाव्य मॉडल, अक्सर अधिक विविध और जटिल ज्ञान डोमेन को संभालने के लिए स्क्रिप्ट के साथ संयुक्त होते हैं।

कुल मिलाकर, स्क्रिप्ट विशिष्ट स्थितियों में विशिष्ट घटनाओं, कार्यों और घटनाओं के अनुक्रमों के बारे में प्रतिनिधित्व करने और तर्क करने के लिए एक संरचित दृष्टिकोण प्रदान करती है, जो एआई सिस्टम में मॉडलिंग नियमित ज्ञान के लिए एक आधार प्रदान करती है।

सारांश:

आर्टिफीसियल इंटेलिजेंस में नॉलेज रिप्रजेंटेशन के लिए एक संरचित दृष्टिकोण में व्यवस्थित और संगठित तरीके से ज्ञान का आयोजन और प्रतिनिधित्व करना शामिल है। संरचित नॉलेज रिप्रजेंटेशन के लिए कई तकनीकों और रूपरेखाओं को विकसित किया गया है, जिसमें सिमेंटिक नेट, फ्रेम, पेट्री नेट, वैचारिक निर्भरता और स्क्रिप्ट शामिल हैं।

सिमेंटिक नेट, या सिमेंटिक नेटवर्क, अवधारणाओं और किनारों का प्रतिनिधित्व करने वाले नोड्स के साथ एक ग्राफ-आधारित संरचना का उपयोग करते हैं जो उनके बीच संबंधों को दर्शाते हैं। वे ज्ञान को पकड़ने और कुशल क्वेरी और अनुमान का समर्थन करने के लिए एक सहज ज्ञान युक्त तरीका प्रदान करते हैं।

फ़्रेम ज्ञान को पदानुक्रमित संरचनाओं में व्यवस्थित करते हैं, जिसमें वैचारिक संस्थाओं का प्रतिनिधित्व करने वाले फ्रेम और विशेषताओं या गुणों का प्रतिनिधित्व करने वाले स्लॉट होते हैं। फ़्रेम विरासत, डिफ़ॉल्ट मूल्यों और फ़्रेम के बीच संबंधों का समर्थन करते हैं, जिससे कुशल प्रतिनिधित्व और ज्ञान के हेरफेर को सक्षम किया जा सकता है।

पेट्री नेट ग्राफिकल मॉडल हैं जिनका उपयोग समवर्ती प्रणालियों का प्रतिनिधित्व और विश्लेषण करने के लिए किया जाता है। उनमें स्टेट का प्रतिनिधित्व करने वाले स्थान, घटनाओं या कार्यों का प्रतिनिधित्व करने वाले संक्रमण और उनके बीच टोकन के प्रवाह का प्रतिनिधित्व करने वाले आर्क शामिल हैं। पेट्री नेट मॉडलिंग प्रक्रियाओं और वर्कफ़्लो के लिए विशेष रूप से उपयोगी हैं।

वैचारिक निर्भरता ज्ञान को परस्पर जुड़े आदिमों के एक सेट के रूप में दर्शाती है, अवधारणाओं, रिश्तों, कार्यों और घटनाओं को कैप्चर करती है। इसका उद्देश्य प्राकृतिक भाषा वाक्यों के अर्थ का प्रतिनिधित्व करना और प्राकृतिक भाषा समझ का समर्थन करना है।

स्क्रिप्ट घटनाओं, स्थितियों या दिनचर्या के बारे में रूढ़िवादी ज्ञान का प्रतिनिधित्व करती हैं। वे भूमिकाओं, प्रतिभागियों और पूर्व शर्तों / पोस्टकंडीशन के साथ किसी दिए गए परिदृश्य में कार्यों या घटनाओं के अपेक्षित अनुक्रम को कैप्चर करते हैं।

प्रत्येक संरचित दृष्टिकोण की अपनी ताकत है और विभिन्न प्रकार के ज्ञान और समस्या डोमेन के लिए अनुकूल है। ये दृष्टिकोण कुशल प्रतिनिधित्व, हेरफेर, तर्क और ज्ञान के अनुमान की सुविधा प्रदान करते हैं, प्राकृतिक भाषा प्रसंस्करण, विशेषज्ञ प्रणालियों और प्रक्रिया मॉडलिंग जैसे विभिन्न एआई अनुप्रयोगों में सहायता करते हैं।

हालांकि ये संरचित दृष्टिकोण प्रभावशाली रहे हैं, यह ध्यान रखना महत्वपूर्ण है कि नॉलेज रिप्रजेंटेशन तकनीक विकसित होती रहती है, और आधुनिक दृष्टिकोण अक्सर आर्टिफीसियल इंटेलिजेंस प्रणालियों में जटिल और विविध ज्ञान डोमेन को पकड़ने और प्रतिनिधित्व करने के लिए अन्य तकनीकों के साथ इन संरचित दृष्टिकोणों के संयोजन को शामिल करते हैं, जैसे कि संभाव्य मॉडल और ओन्टोलॉजी।

प्रमुख शब्द:

- सिमेंटिक नेट
- सिमेंटिक नेट में वंशानुक्रम
- सिमेंटिक नेट में मोनोटोनिक और डिफ़ॉल्ट इनहेरिटेंस में हेरफेर
- सिमेंटिक नेट में अव्यवहार्य तर्क
- फ्रेम
- उलझे हुए फ्रेम में वंशानुक्रम
- पेट्री नेट

- वैचारिक निर्भरता
- स्कॉपिटस

अभ्यास

1. बहुत संक्षिप्त प्रकार के प्रश्न: [2 अंक]

a) संरचित नॉलेज रिप्रजेंटेशन क्या है?

b) नॉलेज रिप्रजेंटेशन के दृष्टिकोण क्या हैं?

c) नॉलेज रिप्रजेंटेशन की दो विधियाँ क्या हैं?

d) प्रथम-क्रम तर्क के बारे में विस्तार से समझाएँ?

e) उपयोगिता शब्द को परिभाषित करें?

2. लघु प्रकार के प्रश्न: [5 अंक]

a) उपयुक्त उदाहरण के साथ आगे और पीछे की श्रृंखला के बारे में विस्तार से वर्णन करें।

b) नॉलेज रिप्रजेंटेशन के प्रकारों के बारे में वर्णन करें?

c) अधिकतम संभावना पैरामीटर सीखने के तरीके क्या हैं?

d) तर्क के विभिन्न तरीकों की सूची बनाएं?

e) नियम आधारित प्रणालियों के बारे में विस्तार से चर्चा करें

f) सेमांटिक नेट्स का उपयोग कहां होता है?

g) डिफ़ीज़िबल रीज़निंग का उपयोग किसलिए किया जाता है?

3. दीर्घ प्रकार के प्रश्न: [10 अंक]

a) सेमांटिक नेट्स कैसे कार्य करता है?

b) सेमांटिक नेट्स में डिफ़ीज़िबल रीज़निंग के लाभ और सीमाएँ क्या हैं?

c) फ़्रेम के गुण और मान्यताएं सेमांटिक नेट्स में कैसे निर्धारित की जाती हैं?

d) सेमांटिक नेट्स में फ़्रेम का प्रयोग किसलिए किया जाता है?

e) सेमांटिक नेट्स में फ़्रेम के लाभ और सीमाएँ क्या हो सकती हैं?

f) सेमांटिक नेट्स में पेट्री नेट्स की सीमाएँ क्या हो सकती हैं?

g) पेट्री नेट्स के अन्य संरचनात्मक मॉडलों के साथ तुलना में सेमांटिक नेट्स में इसकी विशेषताएँ क्या हो सकती हैं?

h) सेमांटिक नेट्स में स्क्रिप्ट्स कैसे प्रदर्शित किए जाते हैं?

i) सेमांटिक नेट्स में स्क्रिप्ट्स का प्रयोग किसलिए किया जाता है?

j) सेमांटिक नेट्स में स्क्रिप्ट्स के लाभ और सीमाएँ क्या हो सकती हैं?

ग्रन्थसूची

ऑन्टोलॉजी अनुसंधान पर विशेष अंक। एआई मैगज़ीन, 24(3), फ़ॉल 2003।

नॉलेज रिप्रजेंटेशन और तर्क प्रोग्रामिंग पर विशेष अंक। आर्टिफिशियल इंटेलिजेंस, 138(1-2), जून 2002।

KEE सॉफ़्टवेयर विकास उपयोगकर्ता मैनुअल। इंटेलीकॉर्प, माउंटेन व्यू, सीए, 1985।

डेम्पस्टर-शेफ़र सिद्धांत, कार्यप्रणाली और अनुप्रयोगों पर विशेष अंक। इंटरनेशनल जर्नल ऑफ़ एप्रोक्सिमेट रीज़निंग, 31(1-2), 2002।

दिमित्रिस अक्लिओप्टास। अंतर समीकरणों के माध्यम से यादृच्छिक 3-सैट के लिए निचली सीमाएं। सैद्धांतिक कंप्यूटर विज्ञान, 265(1-2):159-185, 2001।

शेल्डन बी. एकर्स, जूनियर बाइनरी निर्णय आरेख। कंप्यूटर पर आईईईई लेनदेन, सी-27(6):509-516, 1978।

कार्लोस ई. अल्चौरोन और डेविड माकिंसन। सिद्धांत परिवर्तन का तर्क: संकुचन कार्य और उनके संबंधित पुनरीक्षण कार्य। थियोरिया, 48:14-37, 1982।

कार्लोस ई. अल्चौरोन और डेविड माकिंसन। सिद्धांत परिवर्तन के तर्क पर: सुरक्षित संकुचन। स्टूडियो लॉजिका, 44:405-422, 1985।

कार्लोस ई. अल्चौरोन और डेविड माकिंसन। कुछ भिन्न प्रकार के संकुचन कार्यों के बीच मानचित्र: परिमित मामला। स्टूडियो लॉजिका, 45:187-198, 1986।

कार्लोस ई. अल्चौरोन, पीटर गार्डनफोर्स, और डेविड माकिंसन। सिद्धांत परिवर्तन के तर्क पर: आंशिक संकुचन और पुनरीक्षण कार्य मिलते हैं। जर्नल ऑफ़ सिम्बोलिक लॉजिक, 50(2):510-530, 1985।

डीन अल्लेमांग, माइकल टान्नर, थॉमस बायलैंडर, और जॉन जोसेफसन। परिकल्पना संयोजन की कम्प्यूटेशनल जटिलता। आर्टिफिशियल इंटेलिजेंस, मिलान, इटली पर दसवें अंतर्राष्ट्रीय संयुक्त सम्मेलन की कार्यवाही में, पृष्ठ 1112-1117। मॉर्गन कॉफ़मैन, सैन मेटो, सीए, 1987।

जेम्स एलन, जेम्स हेंडलर, और ऑस्टिन टेट, संपादक। योजना में रीडिंग. मॉर्गन कॉफ़मैन, सैन मेटो, सीए, 1990।

अध्याय - पांच

प्रोबबिलिस्टिक्स रीजनिंग

5.1 परिचय

प्रायोगिक तर्कसंबंधीता (Probabilistic Reasoning) एक प्रभावी तकनीक है जो विभिन्न क्षेत्रों में इस्तेमाल होती है, जैसे कि कंप्यूटर विज्ञान, ज्ञान इंजीनियरिंग, संज्ञानात्मक संगणना, मशीन शिक्षा, और निर्णय विज्ञान.

यह संबंधितता, अनिश्चितता, और प्रायिकता को मॉडल करने का एक तरीका है। प्रायोगिक तर्कसंबंधीता विज्ञान के क्षेत्र में, हम चुनौतियों के सामर्थ्य और निर्णय लेने के लिए अनिश्चितता को मान्यता देते हैं। यह इस्तेमाल करके हम विभिन्न प्रकार के स्थितिगत और गैर-स्थितिगत प्रतीकों के लिए मॉडल, अनुमान और निर्णय बना सकते हैं।

प्रायोगिक तर्कसंबंधीता आधारित संगणक मॉडल विज्ञान के मुख्य संदर्भों में से एक है। यह विभिन्न प्रतीकों के साथ उनके अवसरों, प्रायिकताओं और नतीजों के मान्यता स्तर को प्रतिष्ठानित करता है। यह सामरिक विज्ञान, डेटा विज्ञान, मशीन शिक्षा और बड़ा डेटा विज्ञान में महत्वपूर्ण भूमिका निभाता है।

प्रायोगिक तर्कसंबंधीता में मुख्य अवधारणाएं बायेस थियोरिम, मार्कोव चेन, बेल लॉजिक और संख्यात्मक विज्ञान के साथ जुड़ी होती हैं। यह संगणना और संगणक मॉडलिंग के क्षेत्र में अनुप्रयोगी निर्णयों का निर्माण करने में मदद करता है।

प्रायोगिक तर्कसंबंधीता का उपयोग डेटा के साथ व्याख्यात्मक, विश्लेषकी और निर्णयात्मक कार्यों में किया जाता है, जहां हमें अस्थायीता और अनिश्चितता के साथ दिये गए डेटा से संबंधित निर्णय लेना होता है। यह हमें नए ज्ञान का प्रस्तावना करने और नई सूचना का संगठन करने में मदद करता है, जिससे हम बेहतर निर्णय ले सकते हैं।

प्रायोगिक तर्कसंबंधीता का विस्तृत अनुप्रयोग कई क्षेत्रों में है। यहां कुछ मुख्य उदाहरण दिए गए हैं:

1. मशीन शिक्षा: प्रायोगिक तर्कसंबंधीता मशीन शिक्षा में व्यापक रूप से इस्तेमाल होती है। इसके द्वारा, एक मॉडल आधारित संभावनाओं और पूर्वानुमानों का मूल्यांकन करके अद्यतित किया जा सकता है, जिससे आपके प्रशिक्षण और निर्णय लेने की क्षमता में सुधार होती है।
2. वाणिज्यिक संगणना: प्रायोगिक तर्कसंबंधीता वाणिज्यिक संगणना में भी महत्वपूर्ण भूमिका निभाती है। व्यापारिक और आर्थिक निर्णयों को लेने में, हमें अनिश्चितता को मध्यस्थ करने के लिए व्यापारिक डेटा का उपयोग करना पड़ता है। प्रायोगिक तर्कसंबंधीता के आधार पर, हम व्यापारिक प्रतिस्पर्धा, मार्केट विभाजन, उत्पादों की मूल्यांकन आदि में समझदारीपूर्ण निर्णय ले सकते हैं।
3. रोबोटिक्स: प्रायोगिक तर्कसंबंधीता रोबोटिक्स और स्वयं चालित यांत्रिकी में भी महत्वपूर्ण है। रोबोट को नए और अज्ञात परिस्थितियों में समझने और प्रवर्तित करने के लिए प्रायोगिक तर्कसंबंधीता का उपयोग किया जाता है। यह रोबोट को अपने पर्यावरण की समझ, अनुमान, और निर्णय लेने की क्षमता प्रदान करता है।

4. निर्णय विज्ञान: प्रायोगिक तर्कसंबंधीता निर्णय विज्ञान में महत्वपूर्ण रूप से उपयोग होती है। यह अस्थायीता और अनिश्चितता को ध्यान में रखकर निर्णय लेने की क्षमता प्रदान करती है। यहां प्रायोगिक तर्कसंबंधीता के माध्यम से उपयोगकर्ताओं को उपयुक्त निर्णय लेने में मदद मिलती है, चाहे वे स्वास्थ्य सेवाओं, वित्तीय नियम, या सार्वजनिक नीतियों के संबंध में हों।

इन उदाहरणों के अलावा, प्रायोगिक तर्कसंबंधीता कई अन्य क्षेत्रों में भी उपयोगी है, जैसे कि विज्ञान, आर्थिक मॉडेलिंग, संगणना नेटवर्क, साहित्यिक विज्ञान, राजनीति विज्ञान, और क्रिमिनोलॉजी।

5.2 प्रायोगिक (Probabilistic)

प्रायोगिक (Probabilistic) शब्द संभावनात्मक या संभावनावादी को दर्शाता है। यह एक गणितीय अभिप्रेत शब्द है जिसका अर्थ होता है कि यह किसी घटना, परिस्थिति, या प्रयोग के लिए संभावनाएं या प्रायिकताएं मापने या प्रतिस्पर्धा करने की क्षमता होती हैं।

प्रायोगिक अभिप्रेतता साधारित रूप से उपयोग की जाती है जब हम किसी निर्णय लेने के लिए उपलब्ध जानकारी के आधार पर संभावनाओं की गणना या मूल्यांकन करते हैं। इसका उपयोग विज्ञान, अध्ययन, निर्माण, व्यापार, और निर्णय विज्ञान के विभिन्न क्षेत्रों में होता है।

प्रायोगिक अभिप्रेतता एक माप तकनीक है जो विभिन्न प्रतीकों, तथ्यों और आँकड़ों के साथ संबंधित अनिश्चितताओं को मापने में मदद करती है। इसका उपयोग करके हम नए डेटा के आधार पर अनुमान बना सकते हैं, संभावनाएं प्रतिस्पर्धात्मक रूप से मूल्यांकन कर सकते हैं और निर्णय लेने के लिए समर्थ दृष्टिकोण प्रदान कर सकते हैं।

प्रायोगिक अभिप्रेतता का उपयोग डेटा विज्ञान, मशीन शिक्षा, संगणना, पैटर्न पहचान, संगणना भूगोल, वित्तीय नियमन, आर्थिक मॉडेलिंग, रोबोटिक्स, और व्यापारिक निर्णय निर्माण में किया जाता है। यह तकनीकी निर्णय लेने में मदद करती है और अवसरों और जोखिमों को समझने में मदद करती है।

5.2.1 शर्तगत संभावना (Conditional Probability):

शर्तगत संभावना (Conditional Probability) गणितीय तत्व है जिसका उपयोग दो या अधिक घटनाओं के बीच संबंध को मापने के लिए किया जाता है। इसमें एक घटना होने पर दूसरी घटना के होने की संभावना निर्धारित की जाती है।

शर्तगत संभावना को P(A|B) से प्रकट किया जाता है, जहां A और B घटनाएं होती हैं और B होने की जानकारी के बाद A की संभावना मापी जाती है। शर्तगत संभावना को निम्नलिखित सूत्र से प्राप्त किया जा सकता है:

$$P(A|B) = P(A \text{ और } B) / P(B)$$

यहां, P(A और B) दोनों घटनाओं A और B का संयोजन संभावना है, और P(B) B की संभावना है। शर्तगत संभावना घटनाओं के बीच संबंध को व्यक्त करने में मदद करती है, जैसे कि एक घटना के पूर्ववाद के आधार पर एक दूसरी घटना की संभावना का मूल्यांकन करना। शर्तगत संभावना के उदाहरण शामिल हो सकते हैं:

- एक सिटी में बारिश होने की संभावना P(Rain) है। अगर एक व्यक्ति अपनी उम्र के हिसाब से जानना चाहता है कि बारिश होने के बाद उस दिन पार्क में जाने की संभावना क्या है, तो यह P(Park|Rain) होगी।
- एक डायग्नोस्टिक टेस्ट में एक व्यक्ति को बीमारी के लिए पॉजिटिव रिजल्ट मिलता है। अगर हम जानना चाहते हैं कि यह टेस्ट सही रूप से काम कर रहा है और व्यक्ति वास्तविक रूप से बीमार है, तो यह P(Disease|Positive) होगी।

शर्तगत संभावना व्यापार, विज्ञान, संगणना, संख्यात्मक विज्ञान, वित्तीय मॉडेलिंग, बायोस्टैटिस्टिक्स, और मार्केटिंग जैसे क्षेत्रों में व्यापक रूप से उपयोग की जाती है।

उदाहरण 1:

हम एक उदाहरण के माध्यम से शर्तगत संभावना (Conditional Probability) को समझें। मान लें कि एक विमान पायलट के लिए दो परीक्षाएं हैं - एक लिखित परीक्षा (A) और एक फिजिकल टेस्ट (B)। विमान पायलट बनने के लिए पास होने के लिए, पायलट को दोनों परीक्षाओं में पास होना आवश्यक है। ज्ञात हो कि:

- लिखित परीक्षा (A) पास करने की संभावना 0.75 है। इसका अर्थ है कि 75% पायलट लिखित परीक्षा में पास होते हैं।
- फिजिकल टेस्ट (B) पास करने की संभावना 0.60 है। इसका अर्थ है कि 60% पायलट फिजिकल टेस्ट में पास होते हैं।

अब हमें जानना है कि एक व्यक्ति को लिखित परीक्षा (A) पास होने के बाद फिजिकल टेस्ट (B) पास होने की संभावना क्या होगी। इसको हम P(B|A) के रूप में प्रदर्शित कर सकते हैं।

P(B|A) = (P(A और B)) / P(A)

यहां, P(A और B) लिखित परीक्षा (A) और फिजिकल टेस्ट (B) दोनों पास होने की संभावना है और P(A) लिखित परीक्षा (A) पास होने की संभावना है।

अब, हम इन तथ्यों को उपयोग करके शर्तगत संभावना की गणना कर सकते हैं:

P(A और B) = P(A) * P(B) = 0.75 * 0.60 = 0.45

और

P(A) = 0.75

इसे लगातार रखते हुए, हम शर्तगत संभावना की गणना कर सकते हैं:

P(B|A) = (P(A और B)) / P(A) = 0.45 / 0.75 = 0.6

इसलिए, एक व्यक्ति को लिखित परीक्षा (A) पास होने के बाद फिजिकल टेस्ट (B) पास होने की संभावना 0.6 होगी।

उदाहरण 2:

एक आर्टिफिशियल इंटेलिजेंस (Artificial Intelligence) के संदर्भ में शर्तगत संभावना (Conditional Probability) का एक उदाहरण देखें।

मान लें कि एक आर्टिफिशियल इंटेलिजेंस सिस्टम को एक प्रमाणित ईमेल चिट्ठी (Email) क्लासिफाई करने की क्षमता दी गई है। यह सिस्टम स्पैम (Spam) ईमेल और अवरोधित (Ham) ईमेल के बीच अंतर करना होगा। हमारे पास निम्नलिखित ज्ञातियाँ हैं:

- प्रमाणित ईमेल का 80% स्पैम होता है (P(Spam) = 0.8)
- प्रमाणित ईमेल का 20% अवरोधित होता है (P(Ham) = 0.2)
- यदि ईमेल स्पैम होता है, तो वह 90% यथार्थ (सही) स्पैम क्लासिफाई किया जाएगा (P(Correct|Spam) = 0.9)
- यदि ईमेल अवरोधित होता है, तो वह 95% यथार्थ (सही) अवरोधित क्लासिफाई किया जाएगा (P(Correct|Ham) = 0.95)

अब हमें जानना है कि ईमेल वास्तव में स्पैम होने की संभावना क्या है, जब यह सिस्टम उसे स्पैम क्लासिफाई करता है। इसे हम P(Spam|Correct) के रूप में प्रदर्शित कर सकते हैं।

P(Spam|Correct) = (P(Correct|Spam) * P(Spam)) / P(Correct)

यहां, P(Correct|Spam) स्पैम ईमेल को सही स्पैम क्लासिफाई किया जाने की संभावना है, P(Spam) स्पैम होने की संभावना है, और P(Correct) सही क्लासिफाई की संभावना है।

अब, हम इन तथ्यों को उपयोग करके शर्तगत संभावना की गणना कर सकते हैं:

P(Correct) = (P(Correct|Spam) * P(Spam)) + (P(Correct|Ham) * P(Ham)) = (0.9 * 0.8) + (0.95 * 0.2) = 0.72 + 0.19 = 0.91

और

P(Spam|Correct) = (P(Correct|Spam) * P(Spam)) / P(Correct) = (0.9 * 0.8) / 0.91 = 0.72 / 0.91 = 0.791

इसलिए, जब आर्टिफिशियल इंटेलिजेंस सिस्टम ईमेल को स्पैम क्लासिफाई करता है और ईमेल सही रूप से स्पैम क्लासिफाई होता है, तो संभावना है कि वास्तव में ईमेल स्पैम होने की संभावना 0.791 होगी।

इस उदाहरण से देखा जा सकता है कि शर्तगत संभावना कैसे आर्टिफिशियल इंटेलिजेंस में निर्णय लेने और विश्लेषण करने में मदद करती है।

5.2.2 बायेस सूत्र (Bayes' Rule)

बायेस का सूत्र (Bayes' Rule) एक महत्वपूर्ण गणितीय सूत्र है जो आर्टिफिशियल इंटेलिजेंस (Artificial Intelligence) में व्यापक रूप से उपयोग होता है। यह सूत्र शर्तगत संभावना (Conditional Probability) के संबंध में विशेषज्ञता प्रदान करता है और नए ज्ञान के आधार पर प्रायोगिक प्रदर्शन को अद्यतित करने में मदद करता है।

बायेस का सूत्र P(A|B) = (P(B|A) * P(A)) / P(B) होता है, जहां A और B दो घटनाएं होती हैं। इसके द्वारा हम एक घटना के होने के बाद दूसरी घटना की संभावना को अद्यतित कर सकते हैं।

आईए एक उदाहरण के माध्यम से बायेस के सूत्र को समझें:

मान लें कि हमारे पास एक आर्टिफिशियल इंटेलिजेंस सिस्टम है जो टेक्स्ट संदेशों को स्पैम या अवरोधित (हैम) क्लासिफाई करने की क्षमता रखता है। हमारे पास निम्नलिखित ज्ञातियां हैं:

- टेक्स्ट संदेश का 60% स्पैम होता है (P(Spam) = 0.6)
- टेक्स्ट संदेश का 40% अवरोधित होता है (P(Ham) = 0.4)
- यदि संदेश स्पैम है, तो यह 90% सही ढंग से स्पैम क्लासिफाई होता है (P(Correct|Spam) = 0.9)
- यदि संदेश अवरोधित है, तो यह 95% सही ढंग से अवरोधित क्लासिफाई होता है (P(Correct|Ham) = 0.95)

अब हमें जानना है कि जब सिस्टम एक संदेश को स्पैम क्लासिफाई करता है, तो यह वास्तव में स्पैम होने की संभावना क्या है। इसे हम P(Spam|Correct) के रूप में प्रदर्शित कर सकते हैं।

P(Spam|Correct) = (P(Correct|Spam) * P(Spam)) / P(Correct)

यहां, P(Correct|Spam) संदेश को सही ढंग से स्पैम क्लासिफाई होने की संभावना है, P(Spam) संदेश स्पैम होने की संभावना है, और P(Correct) सही क्लासिफाई होने की संभावना है।

अब, हम इन तथ्यों को उपयोग करके बायेस के सूत्र की गणना कर सकते हैं:

P(Correct) = (P(Correct|Spam) ∗ P(Spam)) + (P(Correct|Ham) ∗ P(Ham)) = (0.9 ∗ 0.6) + (0.95 ∗ 0.4) = 0.54 + 0.38 = 0.92

और

P(Spam|Correct) = (P(Correct|Spam) ∗ P(Spam)) / P(Correct) = (0.9 ∗ 0.6) / 0.92 = 0.54 / 0.92 ≈ 0.587

इसलिए, जब आर्टिफिशियल इंटेलिजेंस सिस्टम एक संदेश को स्पैम क्लासिफाई करता है और संदेश सही ढंग से स्पैम क्लासिफाई होता है, तो संभावना है कि वास्तव में संदेश स्पैम होने की संभावना लगभग 0.587 होगी।

बायेस सूत्र (Bayes' Rule) के अनुप्रयोग

आर्टिफिशियल इंटेलिजेंस (Artificial Intelligence) में बायेस के सूत्र (Bayes' Rule) के कई अभियांत्रिकी और अनुप्रयोग हैं। इसका उपयोग ज्ञान अद्यतन, संभावना मूल्यांकन, समस्या समाधान, फ़िल्टरिंग, क्लासिफाईकर, निर्णय समर्थन, और डेटा माइनिंग में किया जाता है। यहां कुछ आर्टिफिशियल इंटेलिजेंस में बायेस के सूत्र के अभियांत्रिकी अनुप्रयोगों के उदाहरण हैं:

1. स्पैम फ़िल्टरिंग: बायेस के सूत्र का उपयोग स्पैम फ़िल्टरिंग में किया जाता है। यहां संदेश के विभिन्न विशेषताओं के आधार पर संदेश को स्पैम या अवरोधित (हैम) के रूप में श्रेणीबद्ध किया जाता है। बायेस के सूत्र का उपयोग करके, प्रशिक्षित किए गए मॉडल ज्ञात डेटा के आधार पर नए संदेश को स्पैम या अवरोधित क्लासिफाई करते हैं।
2. निर्णय समर्थन: बायेस के सूत्र का उपयोग निर्णय समर्थन में किया जाता है, जहां यथार्थता और संभावनाओं के आधार पर निर्णय लिया जाता है। इसका उपयोग अनुकरण की समस्याओं, फ़िल्टरिंग, विचार-क्रिया निर्णय और व्यापारिक निर्णय लेने में किया जाता है।
3. प्रायिकतावादी मॉडल: बायेस के सूत्र के आधार पर, प्रायिकतावादी मॉडल विकसित किए जाते हैं जो नए ज्ञान और डेटा पर आधारित होते हैं। इन मॉडलों में, प्राथमिकताओं, साक्ष्य, और पूर्वज्ञान को अद्यतित करने के लिए बायेस का सूत्र उपयोग किया जाता है।
4. विज्ञानात्मक माइनिंग: बायेस के सूत्र का उपयोग डेटा माइनिंग में किया जाता है, जहां आर्टिफिशियल इंटेलिजेंस सिस्टम नए नियम, अनुमान, और प्रायिकताएं खोजने के लिए पर्याप्त ज्ञात डेटा का उपयोग करता है। बायेस के सूत्र अनुसार, डेटा के प्राथमिकताओं और साक्ष्यों का मूल्यांकन किया जाता है, जिससे नई संभावनाएं और नियमों की खोज की जा सकती है।
5. पैटर्न पहचान: बायेस के सूत्र का उपयोग पैटर्न पहचान में किया जाता है। जब किसी पैटर्न को नए डेटा में पहचाना जाना हो, तो प्रशिक्षित आर्टिफिशियल इंटेलिजेंस सिस्टम बायेस के सूत्र के आधार पर प्राथमिकताओं और साक्ष्यों का उपयोग करता है। इसके बाद, सिस्टम नए डेटा को इन पैटर्नों में श्रेणीबद्ध करने में सक्षम होता है।

ये केवल कुछ उदाहरण हैं जिनमें बायेस के सूत्र का उपयोग आर्टिफिशियल इंटेलिजेंस में किया जाता है।

5.2.3 बायेसियन नेटवर्क - प्रतिनिधित्व, निर्माण और अनुमान

बायेसियन नेटवर्क्स (Bayesian Networks) आर्टिफिशियल इंटेलिजेंस (Artificial Intelligence) में विशेष रूप से इनफ़रेंस और प्रोबेबिलिस्टिक रीज़निंग के लिए उपयोग होते हैं। ये नेटवर्क्स प्राथमिकताओं, साक्ष्यों और निर्णयों के बीच आपसी संबंधों को प्रतिष्ठान देते हैं। यहां हम बायेसियन नेटवर्क्स के विस्तार, निर्माण और इनफ़रेंस के संदर्भ में बात करेंगे:

1. प्रतिष्ठान (Representation): बायेसियन नेटवर्क्स को ग्राफिकल माध्यम द्वारा प्रतिष्ठित किया जाता है, जहां ग्राफ के नोड्स ज्ञात परिप्रेक्ष्यों को और बायेसियन नेटवर्क्स के बायेसियन संबंधों को प्रतिष्ठित करते हैं। नोड्स वास्तविक या अस्पष्ट संगठन को दर्शाते हैं, और इन्फ़ोर्मेशन को संगठित रूप में प्रस्तुत करते हैं।
2. निर्माण (Construction): बायेसियन नेटवर्क्स को निर्माण करने के लिए प्राथमिकताओं, साक्ष्यों और संभाव्य संबंधों को व्यक्त किया जाता है। यहां कई तरीके हैं, जैसे बायेसियन पैरामीटर अद्यतन, माकुलीय अद्यतन, ग्राफ संरचना अद्यतन, और डेटा माइनिंग के माध्यम से नेटवर्क्स को निर्मित किया जा सकता है।
3. इनफ़रेंस (Inference): बायेसियन नेटवर्क्स का मुख्य उद्देश्य प्राथमिकताओं और साक्ष्यों के आधार पर नई संभाव्यताएं या निर्णयों को निकालना है। ये नेटवर्क्स प्रदर्शित संभावनाओं, परिप्रेक्ष्यों और निष्कर्षों को साझा करने की क्षमता रखते हैं। प्राथमिकताएं और साक्ष्यों के आधार पर, बायेसियन नेटवर्क्स के माध्यम से हम अनुमान, पूर्वाभास और अधिकारण निकाल सकते हैं।

बायेसियन नेटवर्क्स विशेष रूप से प्रायोगिक निर्णय समर्थन, संभावनाओं के मूल्यांकन, पूर्वज्ञान का उपयोग, और पैटर्न पहचान में उपयोग होते हैं। इन नेटवर्क्स का उपयोग करके, आर्टिफिशियल इंटेलिजेंस सिस्टम अद्यतित निर्णय लेते हैं, अनुमान लगाते हैं और नए ज्ञान का विकास करते हैं।

बायेसियन नेटवर्क – प्रतिनिधित्व

बायेसियन नेटवर्क्स (Bayesian Networks) का प्रतिष्ठान आर्टिफिशियल इंटेलिजेंस में ग्राफिकल रूप में किया जाता है। इसके लिए यह एक प्रकार का अभियांत्रिकी आदान-प्रदान करता है जो प्राथमिकताओं, साक्ष्यों और संभाव्य संबंधों को दर्शाता है। बायेसियन नेटवर्क्स ग्राफ के रूप में प्रस्तुत होता है, जहां नोड्स (वर्टेक्स) और बायेसियन बांध्यक (एड्ज) प्रतिष्ठित किए जाते हैं।

यहां कुछ मुख्य प्रतीकों को बयान किया गया है जो बायेसियन नेटवर्क्स के रूप में प्रतिष्ठित किए जाते हैं:

1. नोड्स (Nodes): नोड्स ग्राफ के वर्टेक्स के रूप में प्रतिष्ठित होते हैं। ये प्राथमिकताएं, साक्ष्य या बायेसियन परिप्रेक्ष्यों को दर्शाते हैं। प्रत्येक नोड पर एक नाम और मूल्यांकन का सेट होता है।

2. बायेसियन बांध्यक (Bayesian Edges): बांध्यक ग्राफ के एड्ज के रूप में प्रतिष्ठित होते हैं। ये नोड्स के बीच संबंध को प्रदर्शित करते हैं। एक बांध्यक एड्ज द्वारा एक नोड से दूसरे नोड की प्राथमिकता या संभावना का प्रभाव प्रकट होता है।
3. संभावनाओं के नियम (Probability Tables): प्रत्येक नोड के लिए एक संभावना सारणी होती है, जो नोड के मूल्यांकन की संभावनाएं प्रदर्शित करती है। ये सारणियाँ उदाहरण और पहले के ज्ञान के आधार पर निर्मित की जाती हैं।
4. संघातिक और असंघातिक नेटवर्क्स (Directed vs Undirected Networks): बायेसियन नेटवर्क्स में एड्ज एक संघातिक या असंघातिक नेटवर्क का प्रतीक हो सकते हैं। संघातिक नेटवर्क में एड्ज दिशा धारण करते हैं, जबकि असंघातिक नेटवर्क में एड्ज इशारों द्वारा जुड़े होते हैं।

बायेसियन नेटवर्क – निर्माण

आर्टिफिशियल इंटेलिजेंस में बायेसियन नेटवर्क्स (Bayesian Networks) का निर्माण विभिन्न तकनीकों और प्रक्रियाओं का उपयोग करके किया जाता है। यहां हम बायेसियन नेटवर्क्स के निर्माण की कुछ मुख्य प्रक्रियाएं और तकनीकों के बारे में चर्चा करेंगे:

1. श्रेणीबद्ध नेटवर्क्स (Structured Networks): श्रेणीबद्ध नेटवर्क्स निर्माण की प्रक्रिया में एक पहले से निर्धारित नेटवर्क संरचना का उपयोग किया जाता है। इसमें प्राथमिकताएं, साक्ष्य और संभावनाएं पहले से ही निर्धारित होती हैं और नेटवर्क में संबंधित बायेसियन बांध्यक जोड़े जाते हैं।
2. आपूर्ति-मांग नेटवर्क्स (Supply-Demand Networks): इस प्रक्रिया में, आपूर्ति और मांग के बीच के संबंधों को मॉडल करने के लिए बायेसियन नेटवर्क्स का निर्माण किया जाता है। इसमें उत्पादन, आपूर्ति, मांग, खाली स्थान, अपारदर्शिता आदि के नोड्स और उनके बीच के संबंधों को बायेसियन बांध्यक के माध्यम से प्रतिष्ठित किया जाता है।
3. अद्यतन-आधारित नेटवर्क्स (Update-Based Networks): यह प्रक्रिया बायेसियन नेटवर्क का निर्माण करने के लिए माकुलीय अद्यतन की तकनीक का उपयोग करती है। पहले, शुरुआती ज्ञान के आधार पर नेटवर्क निर्मित किया जाता है और फिर नए साक्ष्यों और अद्यतित ज्ञान के साथ नेटवर्क अपडेट किया जाता है।
4. डेटा-माइन्ड नेटवर्क्स (Data-Mined Networks): यह प्रक्रिया डेटा माइनिंग के तकनीकों का उपयोग करती है और डेटा के प्राथमिकताओं और साक्ष्यों के आधार पर नेटवर्क निर्मित करती है। इसमें डेटा पर्याप्तता, फ़िल्टरिंग, अनुकरण और प्रतिभाजन के माध्यम से नेटवर्क निर्मित किया जाता है।

ये केवल कुछ नेटवर्क निर्माण की प्रक्रियाएं हैं जो आर्टिफिशियल इंटेलिजेंस में बायेसियन नेटवर्क्स के निर्माण के लिए उपयोग होती हैं। इन प्रक्रियाओं का उपयोग करके, हम प्राथमिकताओं, साक्ष्यों और निष्कर्षों के बीच के संबंधों को मॉडल करके और नए डेटा के साथ नेटवर्क को अद्यतित करके बायेसियन नेटवर्क्स का निर्माण कर सकते हैं।

बायेसियन नेटवर्क – इनफ़रेंस (Inference)

आर्टिफिशियल इंटेलिजेंस (Artificial Intelligence) में बायेसियन नेटवर्क्स (Bayesian Networks) के माध्यम से इनफ़रेंस (Inference) करने का मतलब होता है प्राथमिकताओं, साक्ष्यों और निष्कर्षों के आधार पर नए संभाव्यताएं या निर्णयों को निकालना। यह नेटवर्क प्रदर्शित संभावनाओं, परिप्रेक्ष्यों और निष्कर्षों को साझा करने की क्षमता रखता है।

आर्टिफिशियल इंटेलिजेंस में बायेसियन नेटवर्क्स के इनफ़रेंस के लिए निम्नलिखित प्रक्रियाएं हो सकती हैं:

1. अद्यतन (Updating): नए साक्ष्यों के आधार पर बायेसियन नेटवर्क को अद्यतित करके नई संभाव्यताएं या निष्कर्ष निकालने का अद्यतित ज्ञान प्राप्त किया जा सकता है। इसमें नए साक्ष्यों के मूल्यांकन का उपयोग किया जाता है और नेटवर्क के बायेसियन संबंधों को अद्यतित किया जाता है।
2. पूर्वाभास (Prediction): प्राथमिकताओं और साक्ष्यों के आधार पर, बायेसियन नेटवर्क के माध्यम से हम भविष्यवाणी कर सकते हैं। यह नई संभाव्यताएं या पूर्वाभास की रूपरेखा के रूप में प्रस्तुत होती है जो आगामी घटनाओं की संभावना दर्शाती हैं।
3. निर्णय समर्थन (Decision Support): बायेसियन नेटवर्क्स का उपयोग निर्णय समर्थन करने में किया जा सकता है, जहां नेटवर्क प्राथमिकताओं, साक्ष्यों और निर्णयों के बीच के संबंध को दर्शाता है। यह निर्णय लेने में मदद कर सकता है जैसे कि किसी निष्कर्ष की प्राथमिकता की जांच, विभाजन निर्णय, और रिस्क विश्लेषण आदि।
4. अभिप्राय विश्लेषण (Causal Analysis): बायेसियन नेटवर्क्स के माध्यम से हम अभिप्राय विश्लेषण कर सकते हैं, जिसमें नेटवर्क के बायेसियन संबंधों का उपयोग करके एक प्राथमिकता या साक्ष्य के पीछे के कारणों की खोज की जाती है। यह हमें प्रभाव-प्रतिक्रिया संबंध, कारण-परिणाम संबंध और सामंजस्यिक संबंधों की समझ में मदद करता है।

इन तकनीकों का उपयोग करके, बायेसियन नेटवर्क्स आर्टिफिशियल इंटेलिजेंस के कई क्षेत्रों में इनफ़रेंस के लिए प्रयोग होते हैं, जैसे कि रोबोटिक्स, विश्लेषण, निर्णय समर्थन, मेडिकल डायग्नोस्टिक्स, और वाणिज्यिक निर्णय आदि।

5.3 अस्थायी मॉडल (Temporal Model)

आर्टिफिशियल इंटेलिजेंस में टेम्पोरल मॉडल (Temporal Model) एक मॉडल है जो समय के प्रवाह और कार्रवाई को मॉडल करता है। यह विशेष रूप से समय संबंधी डेटा और इवेंट्स के बारे में सूचना संग्रहित करता है और इसका उपयोग कई क्षेत्रों में किया जाता है। यह मॉडल दृश्यमानता, पूर्वाभास, विश्लेषण, और निर्णय समर्थन में मदद करता है।

टेम्पोरल मॉडल की कुछ मुख्य विशेषताएं शामिल हो सकती हैं:

1. समय-स्थायीता (Temporal Persistence): टेम्पोरल मॉडल समय के प्रवाह को मॉडल करता है और डेटा और इवेंट्स की स्थायित्व को संग्रहित करता है। यह संघातिक और असंघातिक इवेंट्स के बीच समयी संबंधों को दर्शाता है।
2. टेम्पोरल लॉजिक (Temporal Logic): टेम्पोरल मॉडल में टेम्पोरल लॉजिक का उपयोग किया जाता है जो समय के प्रवाह के प्रमाणीकरण और संबंधों को व्यक्त करता है। यह नियम, प्राथमिकताएं, और विशेष लॉजिकल संबंधों के बीच समय के प्रवाह को संबोधित करता है।
3. टेम्पोरल अभिप्राय (Temporal Abstraction): टेम्पोरल मॉडल अभिप्राय तकनीकों का उपयोग करके डेटा को टेम्पोरल इवेंट्स और कार्रवाई के स्तर से उच्च स्तरीय ट्रेंड और पैटर्न में विश्लेषण कर सकता है। यह अनुकरण, रूट कॉज, और अनुमान जैसी कार्रवाई को समझने में मदद करता है।
4. टेम्पोरल निर्णय (Temporal Reasoning): टेम्पोरल मॉडल टेम्पोरल निर्णय करने की क्षमता प्रदान करता है, जिसमें समय संबंधी डेटा और इवेंट्स के आधार पर पूर्वाभास, भविष्यवाणी, समयी संबंधों की पहचान और रिस्क विश्लेषण करना शामिल होता है।

टेम्पोरल मॉडल आर्टिफिशियल इंटेलिजेंस के कई क्षेत्रों में उपयोग होता है, जैसे कि संघातिक विश्लेषण, वाणिज्यिक विश्लेषण, सामंजस्यिक रोबोटिक्स, वाहन निर्णय, बायोमेडिकल प्रभाव और परिणाम पठन, और विधानसभा विश्लेषण आदि।

टेम्पोरल मॉडल के अनुप्रयोग

आर्टिफिशियल इंटेलिजेंस में टेम्पोरल मॉडल के कई उपयोग होते हैं। यहां कुछ महत्वपूर्ण एप्लिकेशन की उल्लेख की गई है:

1. समयी संख्यात्मक विश्लेषण (Time Series Analysis): टेम्पोरल मॉडल द्वारा समयी संख्यात्मक डेटा के विश्लेषण की जाती है। यह डेटा में समय के प्रवाह के आधार पर पैटर्न, रुचि और अनुकरणों का पता लगाने में मदद करता है। इसे आर्टिफिशियल इंटेलिजेंस के कई क्षेत्रों में उपयोग किया जाता है, जैसे कि आर्थिक विश्लेषण, मौसम पूर्वानुमान, और स्वास्थ्य सेवाओं के लिए रोग पूर्वानुमान।
2. आवृत्ति विश्लेषण (Recurrent Analysis): टेम्पोरल मॉडल का उपयोग आवृत्ति विश्लेषण के लिए किया जाता है, जहां पश्चात्य संदर्भों के आधार पर गतिविधियों, इवेंट्स और समय संबंधी आवृत्तियों का पता लगाया जाता है। यह स्थानीय रोबोटिक्स, भाषा प्रसंस्करण, और संगीत समझ में उपयोगी होता है।
3. समयी निर्णय समर्थन (Temporal Decision Support): टेम्पोरल मॉडल का उपयोग निर्णय समर्थन के लिए किया जाता है, जहां यह समय के प्रवाह के आधार पर समयी संबंधी डेटा और इवेंट्स का उपयोग करके निर्णय लेने में मदद करता है। यह आर्थिक निर्णय, संचालन विश्लेषण, यात्रा निर्णय और वित्तीय विश्लेषण में उपयोगी होता है।

4. समयी संबंधित भविष्यवाणी (Temporal Forecasting): टेम्पोरल मॉडल का उपयोग समयी संबंधित भविष्यवाणी करने के लिए किया जाता है। यह मॉडल विश्लेषण और पूर्वाभास के माध्यम से समय के प्रवाह को जांचता है और आगामी घटनाओं की संभावनाओं को अनुमानित करता है। यह वित्तीय भविष्यवाणी, मार्केटिंग ट्रेंड्स, और विशेष आपूर्ति और मांग विश्लेषण में उपयोगी होता है।

इन एप्लिकेशन के अलावा, टेम्पोरल मॉडल का उपयोग बायोमेडिकल प्रभाव, प्राकृतिक भूकंप पूर्वानुमान, उत्पादन प्रबंधन, वाहन निर्णय, और नेटवर्क विश्लेषण जैसे अन्य क्षेत्रों में भी होता है।

उदाहरण

सोचिए कि आप एक इलेक्ट्रॉनिक्स कंपनी के लिए एक प्रोडक्ट लॉन्च कर रहे हैं। आपके पास पिछले 5 वर्षों के इलेक्ट्रॉनिक्स प्रोडक्ट्स की बिक्री डेटा है, जिसमें हर वर्ष के महीनों के लिए इकाइयों में बिक्री दी गई है। आपको यह जानना है कि किस महीने आपका नया प्रोडक्ट लॉन्च करना चाहिए ताकि आप अधिक संभावित हों कि आपके नए प्रोडक्ट की बिक्री अधिक होगी।
यहां पिछले 5 वर्षों के इलेक्ट्रॉनिक्स प्रोडक्ट्स की मासिक बिक्री डेटा है:

वर्ष 1:
जनवरी: 1000 इकाइयां
फरवरी: 1200 इकाइयां
मार्च: 1500 इकाइयां
अप्रैल: 1800 इकाइयां
मई: 2000 इकाइयां
जून: 2200 इकाइयां
जुलाई: 2300 इकाइयां
अगस्त: 2200 इकाइयां
सितंबर: 2000 इकाइयां
अक्टूबर: 1800 इकाइयां
नवंबर: 1500 इकाइयां
दिसंबर: 1200 इकाइयां

वर्ष 2:
जनवरी: 1300 इकाइयां
फरवरी: 1400 इकाइयां
मार्च: 1600 इकाइयां
अप्रैल: 1900 इकाइयां

मई: 2100 इकाइयां
जून: 2300 इकाइयां
जुलाई: 2400 इकाइयां
अगस्त: 2300 इकाइयां
सितंबर: 2100 इकाइयां
अक्टूबर: 1900 इकाइयां
नवंबर: 1600 इकाइयां
दिसंबर: 1400 इकाइयां

वर्ष 3:
जनवरी: 1100 इकाइयां
फरवरी: 1300 इकाइयां
मार्च: 1400 इकाइयां
अप्रैल: 1700 इकाइयां
मई: 1900 इकाइयां
जून: 2100 इकाइयां
जुलाई: 2200 इकाइयां
अगस्त: 2100 इकाइयां
सितंबर: 1900 इकाइयां
अक्टूबर: 1700 इकाइयां
नवंबर: 1400 इकाइयां
दिसंबर: 1300 इकाइयां

वर्ष 4:
जनवरी: 1500 इकाइयां
फरवरी: 1700 इकाइयां
मार्च: 1800 इकाइयां
अप्रैल: 2000 इकाइयां
मई: 2200 इकाइयां
जून: 2400 इकाइयां
जुलाई: 2500 इकाइयां
अगस्त: 2400 इकाइयां

सितंबर: 2200 इकाइयां
अक्टूबर: 2000 इकाइयां
नवंबर: 1800 इकाइयां
दिसंबर: 1700 इकाइयां

वर्ष 5:
जनवरी: 1800 इकाइयां
फरवरी: 2000 इकाइयां
मार्च: 2200 इकाइयां
अप्रैल: 2400 इकाइयां
मई: 2600 इकाइयां
जून: 2800 इकाइयां
जुलाई: 2900 इकाइयां
अगस्त: 2800 इकाइयां
सितंबर: 2600 इकाइयां
अक्टूबर: 2400 इकाइयां
नवंबर: 2200 इकाइयां
दिसंबर: 2000 इकाइयां

इस उदाहरण में, आपको प्रतिमाह बिक्री डेटा के आधार पर पट्टों और बिक्री ट्रेंड को विश्लेषण करना होगा। यह विश्लेषण आपको बताएगा कि किस महीने बिक्री सबसे अधिक होती है और किस महीने बिक्री कम होती है। आप इस विश्लेषण के आधार पर नए प्रोडक्ट के लॉन्च का फैसला ले सकते हैं। यह एक साधारण उदाहरण है जिसमें टेम्पोरल मॉडल का उपयोग किया जाता है ताकि आप व्यापारिक निर्णय ले सकें। वास्तविक जीवन में, यहां बहुत सारे तकनीकी विवरण और एल्गोरिदम्स शामिल हो सकते हैं, जो अधिक उन्नत टेम्पोरल मॉडलिंग को विश्लेषण कर सकते हैं।

5.4 हिडन मार्कॉव मॉडल

हिडन मार्कॉव मॉडल (Hidden Markov Model या HMM) एक संदर्भी प्रकार का प्राकृतिक भाषा प्रसंस्करण मॉडल है जिसका उपयोग आर्टिफिशियल इंटेलिजेंस में किया जाता है। HMM वहाँ संभाव्य स्थितियों के एक प्रवाह को प्रतिष्ठित करता है, जहाँ प्रत्येक स्थिति संबंधित प्राकृतिक भाषा में एक निर्दिष्ट विधि के साथ जुड़ी होती है।
हिडन मार्कॉव मॉडल के अंतर्गत, दो प्रमुख प्रतिष्ठित अंश होते हैं:

1. संदर्भी स्थिति (Hidden States): यह संभाव्य स्थितियों का सेट होता है, जिनकी दृष्टि से हमें विवेकपूर्वक जानकारी नहीं होती है। इन स्थितियों को वास्तविकता में नहीं देखा जा सकता है, लेकिन हमें उनके बारे में कुछ निश्चित जानकारी होती है, जैसे कि उनकी संभावनाओं का सेट।
2. वीज़ीबल संकेत (Visible Symbols): यह संकेतों का सेट होता है, जिन्हें हम संदर्भी स्थितियों के अनुसार देख सकते हैं। इन संकेतों को हम निदर्शित करते हैं और उन्हें अनुमानित या विश्लेषित करते हैं।

हिडन मार्कोव मॉडल का उपयोग वाणिज्यिक विश्लेषण, भाषा प्रसंस्करण, वाणिज्यिक निर्णय समर्थन, वॉयस व्यवहारी ईंधन, और आंतरिक प्रवृत्तियों के प्रकटीकरण में किया जाता है। यह भाषा अनुवाद, निर्णय समर्थन प्रणालियों, हाथों के चित्रकारी, मार्केटिंग ट्रेंड विश्लेषण, और स्वचालित वाणिज्यिक प्रणालियों में उपयोगी होता है।

हिडन मार्कोव मॉडल के अनुप्रयोग

हिडन मार्कोव मॉडल (Hidden Markov Model या HMM) आर्टिफिशियल इंटेलिजेंस में कई क्षेत्रों में उपयोगी होता है। यहां कुछ महत्वपूर्ण उदाहरण दिए गए हैं:

1. भाषा प्रसंस्करण (Language Processing): HMM का उपयोग भाषा प्रसंस्करण के क्षेत्र में होता है, जहां इसका उपयोग वाक्य रचना, शब्दों का टैगिंग, भाषा अनुवाद, और भाषाई संज्ञानात्मक विश्लेषण में किया जाता है। HMM मॉडल वाक्य में छुपे अद्यावधिकता और भाषाई संरचना को अधिक समझने में मदद करता है।
2. रोबोटिक्स (Robotics): HMM रोबोटिक्स के क्षेत्र में आपूर्ति चयन, स्थानांतरण विन्यास, और नेविगेशन में उपयोगी होता है। इसका उपयोग रोबोटिक नेविगेशन में स्थान संदर्भ, अवकाश मॉडल, और अद्यावधिकता संज्ञानात्मक विश्लेषण के लिए किया जाता है।
3. वाणिज्यिक निर्णय समर्थन (Business Decision Support): HMM का उपयोग वाणिज्यिक निर्णय समर्थन के क्षेत्र में विक्रय, पूर्ति चयन, बाजारी विश्लेषण, और ग्राहक व्यवहार की भविष्यवाणी में किया जाता है। HMM मॉडल व्यापारिक प्रवृत्तियों के पट्टों का विश्लेषण करके निर्णय समर्थन में मदद करता है।
4. वॉयस रिकग्निशन (Speech Recognition): HMM वॉयस रिकग्निशन में विक्रय, शब्द तालिका, और ध्वनिमती विश्लेषण में उपयोगी होता है। इसका उपयोग भाषा विन्यास, शब्दों के मूल्यांकन, और उच्चतम संभाव्य पदों की खोज करने में किया जाता है।

इनके अलावा, HMM का उपयोग ऑटोमेटेड वाहन, संकेत संग्रह, मूवी रेकमेंडेशन, वायरल मार्केटिंग, और फाइनांशियल मॉडेलिंग जैसे क्षेत्रों में भी किया जाता है। HMM उपयोगकर्ता के साथ बातचीत और विचारों को समझने के लिए भी उपयोगी होता है, जैसे कि शाखा नेविगेशन और चैटबॉट्स।

उदाहरण

मान लीजिए कि आप एक अच्छी गुणवत्ता वाले चाय बनाने वाली कंपनी चला रहे हैं। आपके पास तीन प्रकार की चाय है: ग्रीन चाय, ब्लैक चाय, और इलायची चाय। आप एक प्रकाशक डिवाइस का उपयोग करके चाय की गुणवत्ता को मापते हैं और चाय के गुणवत्ता स्तर के आधार पर चाय का प्रकार निर्णय करना चाहते हैं।

यहां चाय की गुणवत्ता को दर्शाने वाला डेटा है:

- ग्रीन चाय के लिए गुणवत्ता स्तर: 80, 85, 90, 75, 70
- ब्लैक चाय के लिए गुणवत्ता स्तर: 60, 65, 70, 55, 50
- इलायची चाय के लिए गुणवत्ता स्तर: 90, 95, 80, 85, 80

अब, हम HMM मॉडल का उपयोग करके चाय के प्रकार का पता लगाने का प्रयास करेंगे। यहां, हम संदर्भी स्थितियों के रूप में चाय के प्रकार का उपयोग करेंगे: संदर्भी स्थिति "H1" ग्रीन चाय, "H2" ब्लैक चाय और "H3" इलायची चाय को दर्शाएगी। संदर्भी स्थितियों के बीच के रूप में हम HMM मॉडल का उपयोग करेंगे।

आइए, हम HMM मॉडल को परिभाषित करें:

1. संदर्भी स्थितियां (Hidden States): H1 (ग्रीन चाय), H2 (ब्लैक चाय), H3 (इलायची चाय)
2. वीज़ीबल संकेत (Visible Symbols): चाय की गुणवत्ता स्तर
3. पारंपरिक प्रवाह (Transition Probabilities): चाय के प्रकार के बीच के संदर्भी स्थितियों के परामर्शिक मान। इसे आप अपनी विश्लेषण के आधार पर निर्धारित कर सकते हैं।
4. आरंभिक स्थिति (Initial State): आरंभिक संदर्भी स्थिति की संभावना।

अब हम HMM मॉडल का उपयोग करके चाय के प्रकार का पता लगाने के लिए आरंभिक स्थिति, संदर्भी स्थितियों, और पारंपरिक प्रवाह को जानने का प्रयास करेंगे।

यह एक साधारण उदाहरण है, जिसमें HMM का उपयोग चाय के प्रकार का पता लगाने के लिए किया जाता है। वास्तविक जीवन में, यहां अधिक विवरण और परिभाषित डेटा सेट हो सकते हैं, जो उपयोगकर्ता के विशेष उद्देश्यों और विश्लेषण के लिए विविध तकनीकों का प्रयोग कर सकते हैं।

5.5 सारांश

प्रायोगिक तर्कशास्त्र (Probabilistic Reasoning) आर्टिफिशियल इंटेलिजेंस में एक महत्वपूर्ण और प्रभावी उपकरण है जो हमें अज्ञात और अनिश्चितता के साथ व्यवहार करने में मदद करता है। यह एक प्रायोगिक दृष्टिकोण है जो हमें आधारभूत संभावनाओं, विश्लेषण, और निर्णय लेने में सहायता प्रदान करता है।

प्रायोगिक तर्कशास्त्र का मूल अवधारणा है कि हमारे पास ज्ञात और अज्ञात साक्ष्य (डेटा) होता है और हमें इस डेटा के माध्यम से संभाव्य परिणामों की संभावनाओं का मूल्यांकन करना होता है। यह हमें व्यक्तिगत निर्णय लेने, भविष्यवाणी करने, और समस्या समाधान करने में मदद करता है।

प्रायोगिक तर्कशास्त्र के मुख्य उपकरणों में सम्मिलित हैं:

1. प्रायोगिक मान्यताएँ (Probabilistic Assumptions): यह हमें विभिन्न घटनाओं की संभावनाओं को मान्यता के रूप में देखने में मदद करता है। इसमें आपातकालीनता, स्थितिगतता, और संयुक्त सापेक्षता जैसी मान्यताएँ शामिल हो सकती हैं।
2. संयोगीता (Independence): यह हमें अलग-अलग घटनाओं के बीच स्थायित्व के आधार पर संभावनाएं निर्धारित करने में मदद करता है।
3. बयेस के नियम (Bayes' Rule): यह हमें पूर्व ज्ञान, डेटा और संभावनाएं का मिश्रण करके परिणामों की संभावनाओं को अद्यतन करने में मदद करता है।
4. बयेसियन नेटवर्क (Bayesian Network): यह हमें संबंधित घटनाओं की प्रतिष्ठित संभावनाओं को दर्शाने वाले ग्राफिकल मॉडल के माध्यम से संभावनाएं और प्रतिष्ठितता को प्रदर्शित करने में मदद करता है।

प्रायोगिक तर्कशास्त्र आर्टिफिशियल इंटेलिजेंस के कई क्षेत्रों में उपयोगी है, जैसे कि निर्णय समर्थन सिस्टम, रोबोटिक्स, वाणिज्यिक विश्लेषण, वॉयस रिकग्निशन, और रोग परीक्षण आदि। प्रायोगिक तर्कशास्त्र हमें अज्ञातता और अनिश्चितता के साथ व्यवहार करने की क्षमता प्रदान करता है और समय, संसाधन, और निर्णय के मामलों में हमारी संभावितता को बेहतर ढंग से प्रबंधित करने में मदद करता है।

प्रमुख शब्द:

- संभावनाएं (Probabilities)
- बयेसियन नेटवर्क (Bayesian Network)
- बयेस के नियम (Bayes' Rule)
- संयोगीता (Independence)
- परिभाषित और अपरिभाषित संदर्भी स्थितियाँ (Defined and Undefined Hidden States)
- हिडन मार्कॉव मॉडल
- टेम्पोरल मॉडल

अभ्यास

1. बहुत संक्षिप्त प्रकार के प्रश्न: [2 अंक]

a) प्रोबबिलिस्टिक्स रीजनिंग क्या है?

b) बयेसियन नेटवर्क क्या है

c) बयेसियन नेटवर्क क्या उपयोग है?

d) संयोगीता क्या होती है?

e) बयेस के नियम क्या है?

f) हिडन मार्कोव मॉडल क्या है?

g) टेम्पोरल मॉडल क्या है?

2. लघु प्रकार के प्रश्न: [5 अंक]

a) बयेस के नियम कैसे काम करता है?

b) अनिश्चितता में संभाव्यता सिद्धांत की क्या आवश्यकता है?

c) संभाव्य अनुमान (Probabilistic Inference) का मूल कार्य क्या है?

d) बायेसियन नेटवर्क के बारे में विस्तार से बताएं।

e) एआई में बायेसियन संभाव्य अनुमान क्या है?

f) टेम्पोरल मॉडल का उपयोग किसलिए किया जाता है?

3. दीर्घ प्रकार के प्रश्न: [10 अंक]

a) बयेसियन नेटवर्क के गुण और मान्यताएं सेमांटिक नेट्स में कैसे निर्धारित की जाती हैं?

b) प्रोबबिलिस्टिक्स रीजनिंग के लाभ और सीमाएँ क्या हैं?

c) बायेसियन नेटवर्क के लाभ और सीमाएँ क्या हो सकती हैं?

d) संभाव्य अनुमान (Probabilistic Inference) का प्रयोग किसलिए किया जाता है?

e) अनिश्चितता में संभाव्यता सिद्धांत की सीमाएँ क्या हो सकती हैं?

f) एआई में बायेसियन संभाव्य विशेषताएँ क्या हो सकती हैं?

g) टेम्पोरल मॉडल कैसे प्रदर्शित किए जाते हैं?

ग्रन्थसूची

नीपोलिटन, रिचर्ड ई., "विशेषज्ञ प्रणालियों में संभाव्य तर्क:
सिद्धांत और एल्गोरिदम", जॉन विले एंड संस, 1990। [प्रैक्टिकल गाइड
कार्यान्वयन के लिए.]

ओलिवर, रॉबर्ट एम., और स्मिथ, जेम्स क्यू., संपादक, "इन्फ्लुएंस डायग्राम्स,
विश्वास जाल और निर्णय विश्लेषण", जॉन विले एंड संस, 1990।

पर्ल, जुडिया, "इंटेलिजेंट सिस्टम में संभाव्य तर्क:
प्रशंसनीय अनुमान के नेटवर्क", मॉर्गन कॉफ़मैन, सैन मेटो,
कैलिफोर्निया, 1988. [बायेसियन नेटवर्क के लिए सैद्धांतिक रूपरेखा]

शेफ़र, ग्लेन, और पर्ल, जुडिया, "रीडिंग्स इन अनसर्टेन रीज़निंग",
मॉर्गन कॉफ़मैन, सैन मेटो, कैलिफ़ोर्निया, 1990। 768 पृष्ठ,
आईएसबीएन 1-55860-125-2 पेपर ($49.95)।

आर.ओ. डूडा, पी.ई. हार्ट, और एन.जे. निल्सन, "सब्जेक्टिव बायेसियन मेथड्स
नियम-आधारित अनुमान प्रणालियों के लिए", 1976 की राष्ट्रीय कार्यवाही में
कंप्यूटर सम्मेलन, पृष्ठ 1075-1082, एएफआईपीएस, 1976।

चार्नियाक, यूजीन, "बायेसियन नेटवर्क्स विदाउट टीयर्स", एआई मैगज़ीन
12(4):50-63, शीतकालीन 1991। [शुरुआती लोगों के लिए बायेसियन नेटवर्क का परिचय]

अध्याय 6
मार्कोव निर्णय प्रक्रिया

6.1 परिचय

यह अध्याय विकल्प के मुद्दों की एक श्रेणी का परिचय देता है जिसे अनिश्चितता के साथ अनुक्रमिक निर्णय समस्याओं के रूप में जाना जाता है। वर्तमान निर्णय और भविष्य के निर्णयों के बीच संबंध ऐसे मुद्दों की प्राथमिक विशेषता है। वास्तव में, इन मुद्दों को केवल एक के बजाय पसंद की समस्याओं की एक श्रृंखला के रूप में प्रस्तुत किया जाता है। सुदृढीकरण सीखने का प्रतिमान, जो पर्यवेक्षित सीखने और असुरक्षित सीखने के बीच बैठता है, कम प्रतिक्रिया के साथ अनुक्रमिक निर्णय लेने के कार्यों में सीखने से संबंधित है। यह प्रतिमान राज्यों के संग्रह के रूप में एक पर्यावरण का प्रतिनिधित्व करता है जिसे कार्यों द्वारा नियंत्रित किया जा सकता है। उद्देश्य सिस्टम में हेरफेर करना है ताकि एक निश्चित प्रदर्शन मानदंड अधिकतम हो। (स्टोकेस्टिक) नियोजन मुद्दों, रोबोट नियंत्रण सीखने और गेम-प्लेइंग मुद्दों सहित कई मुद्दों को एमडीपी का उपयोग करके सफलतापूर्वक दर्शाया गया है। वास्तव में, एमडीपी ने अनुक्रमिक निर्णय लेने के शिक्षण के लिए वास्तविक उद्योग मानक के रूप में पदभार संभाला है। एजेंट (अभिनेता या निर्णय निर्माता) को इस प्रक्रिया के प्रत्येक चरण में कार्रवाई के अगले पाठ्यक्रम का चयन करना चाहिए, यह विचार करके कि यह कैसे प्रभावित करेगा कि भविष्य की चुनौतियों को कैसे हल किया जाता है। यह अनुक्रमिक विशेषता एआई में नियोजन के मुद्दों की भी विशेषता है, और यह अक्सर ग्राफ सिद्धांत के सबसे छोटे पथ एल्गोरिदम से जुड़ा होता है। सभी संभावित निर्णयों के परिणामों के आसपास की अनिश्चितता इन पृष्ठों (कार्यों) में खोजे गए मुद्दों का दूसरा पहलू है। एजेंट के पास पहले से अपने निर्णयों के परिणामों के बारे में ज्ञान तक नियतात्मक पहुंच नहीं है। नतीजतन, इस विषय में विविध औपचारिकरण और समाधान रणनीतियां शामिल हैं जो अनिश्चितता के तहत निर्णय लेने के सिद्धांतों द्वारा सुझाई जाती हैं। हमें विशेष रूप से इन रणनीतियों के बीच प्रत्याशित उपयोगिता को अधिकतम करने की पारंपरिक धारणा को कहना चाहिए।

अनिश्चितता के तहत अनुक्रमिक निर्णय और निर्णय के दो मुद्दे इसलिए अनिश्चितता के तहत अनुक्रमिक निर्णय की कठिनाइयों के साथ युग्मित होते हैं। स्टोकेस्टिक स्थितियों में सबसे छोटे पथ के मुद्दों का वर्णन करने के लिए एक सामान्य गणितीय प्रतिमान को मार्कोव निर्णय समस्या (एमडीपी) कहा जाता है। मार्कोव निर्णय प्रक्रियाओं के सिद्धांतों के आधार पर, यह औपचारिकता (एमडीपी के रूप में भी लिखी गई)। मार्कोव निर्णय प्रक्रिया राज्य की अवधारणाओं पर निर्भर करती है, जो एजेंट की वर्तमान स्थिति, कार्रवाई (या निर्णय) का वर्णन करती है, जो प्रक्रिया की गतिशीलता और इनाम को प्रभावित करती है, जो राज्यों के बीच प्रत्येक संक्रमण के लिए देखी जाती है। इस तरह की प्रक्रिया इस संभावना का वर्णन करती है कि राज्यों में निर्णय लेने से राज्यों में संक्रमण होगा और परिणामस्वरूप एक निश्चित इनाम होगा। नतीजतन, एक एमडीपी को एक नियंत्रित मार्कोव श्रृंखला के रूप में सोचा जा सकता है, जहां चयनित कार्रवाई द्वारा प्रत्येक चरण में नियंत्रण प्रदान किया जाता है। प्रक्रिया तब राज्यों की एक श्रृंखला का दौरा करती है, और देखे गए पुरस्कारों का उपयोग इसका आकलन करने के लिए किया जा सकता है। इष्टतम व्यवहार प्राप्त करने के लिए एजेंट को नियंत्रित करना, या अपने समग्र राजस्व को अधिकतम करना, एमडीपी को हल करने की कुंजी है। सर्वोत्तम नियंत्रण रणनीति को हमेशा कार्यों के एकल सेट के रूप में मॉडल नहीं किया जा सकता है क्योंकि कार्रवाई प्रभाव स्टोकेस्टिक होते हैं और इसलिए, निम्नलिखित निर्णय लेने के चरण में विभिन्न वैकल्पिक राज्यों को जन्म दे सकते हैं। इसलिए, एमडीपी के उत्तर आमतौर पर सामान्य योजनाओं या नीतियों (रणनीतियों या निर्णय नियमों) के रूप में प्रदान किए जाते हैं जो बताते हैं कि निर्णय लेने की प्रक्रिया के प्रत्येक चरण में क्या करना है और एजेंट हर संभावित स्थिति तक पहुंच सकता है। किसी दी गई नीति को लागू करने से राज्यों या कार्यों के कई अनुक्रम हो सकते हैं क्योंकि इस बात में अस्पष्टता है कि कार्रवाई कैसे होगी।

.

6.2 एमडीपी सूत्रीकरण

एक नीति का विचार, जो सभी राज्यों से अनुकूलन किए जाने वाले कुछ लक्ष्यों के संबंध में इष्टतमता के निर्णय-सैद्धांतिक उपायों के आधार पर (इष्टतम) कार्रवाई के लिए मानचित्रण है, एआई योजना में एक योजना के विचार का विस्तार है, जो एक प्रारंभिक राज्य से लक्ष्य राज्य तक कार्यों की एक श्रृंखला है।
एमडीपी राज्यों, कार्यों, राज्य संक्रमण और इनाम समारोह के एक विनिर्देश से बने होते हैं। हम बारी-बारी से हर एक के बारे में सोचते हैं।

राज्यों: पर्यावरणीय परिस्थितियों का समूह जब राज्य स्थान का आकार N होता है, याS| = N, S को परिमित सेट s के रूप में परिभाषित किया गया है।[1]..... s^N. एक राज्य प्रतिनिधित्व की गई समस्या की किसी दी गई स्थिति में महत्वपूर्ण सब कुछ का एक विशिष्ट विवरण है। उदाहरण के लिए, लूडो गेम में चार रंगों के बोर्ड टुकड़ों का एक पूरा कॉन्फ़िगरेशन, एक राज्य है।

कार्यों: चरणों की श्रृंखला जब क्रिया स्थान का आकार K होता है, या कब |A| = K, A को परिमित सेट a के रूप में परिभाषित किया गया है[1]..... एक[K]. सिस्टम स्थिति को क्रियाओं द्वारा प्रबंधित किया जा सकता है। ए (एस), जहां ए (एस) ए, उन कार्यों का संग्रह है जिनका उपयोग एक विशिष्ट राज्य में किया जा सकता है जहां एस ∈ S. यद्यपि सभी प्रणालियों में हर राज्य में सभी क्रियाओं का उपयोग नहीं किया जा सकता है, हम आम तौर पर यह मान लेंगे कि सभी के लिए ए (एस) = ए ∈ S. एक पूर्व शर्त फ़ंक्शन पूर्व: एस ए सही, गलत का उपयोग अधिक संरचित प्रतिनिधित्व (जैसे सुविधाओं का उपयोग करने वाले) में किया जाता है ताकि वास्तविकता को पकड़ा जा सके कि कुछ क्रियाएं सभी राज्यों में लागू नहीं होती हैं। यह फ़ंक्शन निर्धारित करता है कि क्या क्रिया a ∈ A राज्य में लागू होता है ∈ S.

संक्रमण समारोह: सभी संभावित संक्रमणों में संभाव्यता वितरण के आधार पर, सिस्टम राज्य से एक नए राज्य में बदल जाता है। ∈ राज्य में कार्रवाई ए ए लागू करके एस ∈ S. राज्य में क्रिया करने के बाद अवस्था 's' पर पहुंचने की संभावना संकेतन T (s, a, s') द्वारा दी जाती है, और संक्रमण फ़ंक्शन T को T: S A S [0, 1] के रूप में परिभाषित किया जाता है। यह आवश्यक है कि T(s, a, s') 0 और T(s, a, s' P) 1 सभी क्रियाओं a और सभी अवस्थाओं s और s ' के लिए हो। इसके अतिरिक्त, s' ∈ सभी राज्यों और क्रियाओं के लिए S T (s, a, s') = 1, अर्थात T संभावित अगले राज्यों पर एक उपयुक्त संभाव्यता वितरण को परिभाषित करता है।
मार्कोवियन गतिशीलता के अनुसार, यह अप्रासंगिक है कि सबसे अच्छा विकल्प बनाने के लिए वर्तमान राज्यों से पहले जो भी राज्य और कार्य मौजूद थे। इसे रखने का एक अलग तरीका यह कहना है कि यदि आप एक कार्रवाई चुनते हैं, तो अगले राज्यों में इसकी संभावना वितरण वही होगा जो पिछली बार आपने उस राज्य में किया था। के-मार्कोव होने की संपत्ति, या अंतिम के राज्य जो मार्कोव को वास्तव में 1-मार्कोव बनाने के लिए पर्याप्त हैं, का उपयोग अधिक सामान्य मॉडल का वर्णन करने के लिए किया जा सकता है। हालांकि, कोई हर के-मार्कोव समस्या को संबंधित मार्कोव समस्या में परिवर्तित कर सकता है।

पुरस्कारो का कार्य: जब किसी राज्य में कोई राज्य या कार्रवाई की जाती है, तो पुरस्कार पुरस्कार फ़ंक्शन द्वारा निर्दिष्ट किए जाते हैं।[2]. राज्यों में अर्जित इनाम को राज्य इनाम समारोह द्वारा वर्णित किया जाता है, जिसमें सूत्र आर: एस → आर है। हालांकि, दो और परिभाषाएं हैं। R को या तो R : S × A → R या R : S × A × S → R के रूप में परिभाषित किया जा सकता है। पहले में, एक राज्य में रहते हुए किए गए कार्यों को पुरस्कृत किया जाता है, जबकि दूसरे में, राज्यों के बीच विशिष्ट संक्रमण को पुरस्कृत किया जाता है।
एमडीपी का इनाम समारोह एक महत्वपूर्ण घटक है जो अप्रत्यक्ष रूप से सीखने के उद्देश्य को बताता है। यह निर्धारित करने के लिए कि सिस्टम, या एमडीपी को कैसे नियंत्रित किया जाना चाहिए, इनाम फ़ंक्शन का उपयोग किया जाता है।

मार्कोव निर्णय प्रक्रिया निम्नलिखित परिभाषा के साथ एक टपल है: एस राज्यों का एक परिमित संग्रह है, ए क्रियाओं का एक परिमित सेट है, और टी: एस ×ए×एस → [0, 1] एक संक्रमण फ़ंक्शन है। और एक इनाम फ़ंक्शन आर जिसे आर द्वारा निरूपित किया जाता है: एस × ए × एस → आर।
एमडीपी मॉडल को संक्रमण फ़ंक्शन टी और इनाम फ़ंक्शन आर के संयोजन से परिभाषित किया गया है। उदाहरण के लिए, एमडीपी को कभी-कभी राज्य संक्रमण ग्राफ़ के रूप में दिखाया जाता है, जहां नोड्स राज्यों का प्रतिनिधित्व करते हैं और (निर्देशित) किनारे संक्रमण का प्रतिनिधित्व करते हैं।

कार्रवाई नीतियां: जब एक स्टोकेस्टिक सिस्टम को एक एजेंट द्वारा नियंत्रित किया जाता है जो प्रत्येक बार चरण टी पर क्रियाओं को चुनता है और निष्पादित करता है, तो मार्कोव निर्णय प्रक्रियाएं हमें स्टोकेस्टिक सिस्टम की राज्य विकास गतिशीलता का अनुकरण करने में सक्षम बनाती हैं। एक कार्रवाई नीति, जिसे एक रणनीति के रूप में भी जाना जाता है, ऐसे कार्यों का चयन करने की प्रक्रिया है और प्रतीक π द्वारा दर्शाया जाता है। एक नीति विभिन्न लागू क्रियाओं पर संभाव्यता वितरण निर्दिष्ट कर सकती है या नियतात्मक रूप से चुन सकती है कि कौन सी कार्रवाई करनी है। फिर, एक नीति प्रक्रिया की वर्तमान स्थिति पर आधारित हो सकती है, जिसे इतिहास-निर्भर होने के रूप में जाना जाता है, या यह प्रक्रिया के पूरे इतिहास को ध्यान में रख सकता है।

कार्रवाई नीति P_t	एमडीपी	इतिहास पर निर्भर
नियतात्मक प्रणाली	$s_t \dashrightarrow एक_t$	$h_t \dashrightarrow एक_t$
स्टोकेस्टिक सिस्टम	$एक_t, s_t \dashrightarrow$ [0, 1]	$h_t, s_t \dashrightarrow$ [0, 1]

तालिका: एमडीपी के परिवार

नीति, जो एजेंट का एक घटक है, का उद्देश्य एमडीपी-मॉडलिंग वातावरण को विनियमित करना है। एक निश्चित नीति एमडीपी में एक स्थिर संक्रमण वितरण उत्पन्न करती है जिसे मार्कोव सिस्टम में परिवर्तित किया जा सकता है।[3] (S', T)' i जहां S ' = S और T ' (s, s') = T(s, a, s') जब भी π(s) = a होता है।

6.3 उपयोगिता सिद्धांत:

किसी भी एआई प्रणाली की निर्णय लेने की प्रक्रिया इसकी नींव है। उपयोगिता-आधारित दृष्टिकोण में, प्रत्येक कल्पनीय कार्रवाई को एक बार में स्कोर किया जाता है, और उच्चतम स्कोरिंग एक्शन का चयन किया जाता है। यह अपने आप में एक बुनियादी और सरल रणनीति है। उपयोगिता सिद्धांत का विचार वीडियो गेम या यहां तक कि कंप्यूटर से बहुत पहले मौजूद था। कई अन्य विषयों के अलावा, यह खेल सिद्धांत और अर्थशास्त्र में नियोजित किया गया है। उपयोगिता सिद्धांत के केंद्रीय सिद्धांत के अनुसार, किसी दिए गए मॉडल के अंदर हर संभावित कार्रवाई या स्थिति को पर्याप्त रूप से एकल, समान मूल्य द्वारा चिह्नित किया जा सकता है। दिए गए संदर्भ के भीतर एक कार्रवाई की उपयोगिता को इस मान द्वारा वर्णित किया गया है, जिसे आमतौर पर उपयोगिता के रूप में जाना जाता है।
यह समझना महत्वपूर्ण है कि उपयोगिता और मूल्य एक ही चीज नहीं हैं। मूल्य एक मात्रात्मक अवधारणा है। हम किसी चीज की कितनी इच्छा करते हैं, यह उसकी उपयोगिता से मापा जाता है। स्थिति या व्यक्ति के आधार पर, यह बदल सकता है। ₹ 200 के लिए बिल्ली का खिलौना शायद आपकी पसंद होगा यदि आप अमीर थे क्योंकि आप पैसे पर समय की प्राथमिकता कर सकते हैं। आपकी ₹20.00 की बचत महत्वहीन है। हालांकि, यदि आप बेहद गरीब थे, तो आप शायद कम महंगी वस्तु का चयन करेंगे और लंबे समय तक इंतजार करेंगे क्योंकि वह अतिरिक्त ₹ 20 आपके लिए बहुत आवश्यक है। उस पैसे की उपयोगिता उस संदर्भ के आधार पर भिन्न होती है जिसमें इसे देखा जा रहा है, हालांकि एक ही मूल्य है।
उपयोगिता सिद्धांत तब निम्नलिखित को पूर्ववत करता है:
1. पूर्णता: प्रत्येक कल्पनीय बंडल को व्यक्तियों द्वारा क्रम में रैंक किया जा सकता है। रैंक ऑर्डरिंग से पता चलता है कि सिद्धांत यह मानता है कि प्रत्येक व्यक्ति हमेशा उन्हें वरीयताओं के आधार पर कुछ क्रम में रैंक कर सकता है,

भले ही खपत बंडलों के कितने रूप उन्हें प्रस्तुत किए जाएं। नतीजतन, लोग किसी भी बंडल को उसके द्वारा प्रदान किए जाने वाले आनंद के स्तर के अनुसार रैंक कर सकते हैं और इसकी तुलना किसी अन्य बंडल से कर सकते हैं। नतीजतन, हमारे उदाहरण में, भोजन और कपड़ों के आधे मूल्य वाले सप्ताह की तुलना एक सप्ताह के भोजन, एक सप्ताह के कपड़े या किसी अन्य संयोजन से की जा सकती है। वरीयताओं की पूर्णता संपत्ति उस संपत्ति के लिए गणितीय नाम है जो किसी व्यक्ति को किसी भी दिए गए बंडल की तुलना किसी अन्य बंडल के साथ करने की अनुमति देता है।
2. अधिक बेहतर है: मान लें कि कोई बंडल बी पर उत्पादों के बंडल ए का उपभोग करना चुनेगा। फिर उसे एक और बंडल प्रस्तुत किया जाता है जिसमें बंडल ए में सब कुछ अधिक होता है; दूसरे शब्दों में, नए बंडल को ए द्वारा निरूपित किया जाता है जहां = 1। बेहतर आधार के अनुसार, लोग ए के अन्य रूपों की तुलना में बी के साथ-साथ ए को पसंद करते हैं। हमारे उदाहरण के लिए, यदि भोजन के एक सप्ताह के पैकेज की तुलना में दो सप्ताह के भोजन के पैकेज को प्राथमिकता दी जाती है, तो भोजन के दो सप्ताह के पैकेज के लिए एक सप्ताह के भोजन के पैकेज को प्राथमिकता दी जाती है। वरीयताओं पर मोनोटोनिटी धारणा गणित में अधिक-से-बेहतर आधार को दिया गया नाम है। यह मुकाबला करना हमेशा संभव है कि यह अनुमान आमतौर पर विफल रहता है। मान लीजिए कि व्यक्ति के पास किसी अन्य व्यक्ति या उनकी पसंद के दान को अतिरिक्त भोजन दान करने का मौका है। यहां तक कि जब व्यक्ति को इस स्थिति में पर्याप्त भोजन मिला है, तब भी वे अधिक तरसते हैं। एक छिपी हुई संपत्ति किसी भी बंडल की अधिशेष मात्रा के लागत-मुक्त निपटान को मोनोटोनिसिटी धारणा के तहत सक्षम बनाती है।
3. मिक्स-इज़-बेटर: मान लें कि किसी व्यक्ति को एक सप्ताह के भोजन और एक सप्ताह के कपड़ों के बीच कोई प्राथमिकता नहीं है। इसलिए, कोई भी विकल्प अपने दम पर दूसरे के ऊपर नहीं चुना जाता है। "मिक्स-इज़-बेटर" वरीयता सिद्धांत के अनुसार, दोनों के संयोजन को दोनों स्टैंड-अलोन विकल्पों के लिए पसंद किया जाएगा, जैसे कि आधे सप्ताह के कपड़ों के साथ संयुक्त आधे सप्ताह का भोजन।
तर्कसंगतता: उपयोगिता सिद्धांत का सबसे महत्वपूर्ण और विवादास्पद अंतर्निहित आधार। यदि लोग तर्कसंगत हैं, तो उनकी प्राथमिकताएं किसी भी प्रकार की परिपत्रता से बचेंगी; उदाहरण के लिए, यदि बंडल A को बंडल B पर चुना जाता है और बंडल B को बंडल C पर प्राथमिकता दी जाती है, तो A को भी C पर प्राथमिकता दी जाती है। किसी भी परिस्थिति में व्यक्ति A के स्थान पर C को नहीं चुनेगा। आप शायद समझ सकते हैं कि यह अनुमान बहस योग्य क्यों है। यह इस विचार पर आधारित है कि आंतरिक प्राथमिकताएं, या वस्तुओं के बंडलों के रैंक ऑर्डर, संदर्भों और समय में स्थिर रहते हैं।
उपयोगिता स्कोर निर्धारित करते समय स्थिरता महत्वपूर्ण है। उपयोगिता स्कोर पूरे सिस्टम में एक ही पैमाने पर होना चाहिए क्योंकि अंतिम निर्धारण करने के लिए उनकी तुलना एक दूसरे से की जाती है। नतीजतन, सामान्यीकृत स्कोर (संख्या जो 0 से 1 तक होती है) से शुरू करना एक उचित विचार है। संख्याओं की एक पूर्व निर्धारित सीमा के भीतर किसी भी मान का उपयोग सामान्यीकृत स्कोर की गणना करने के लिए किया जा सकता है, जो औसत के लिए भी बहुत सरल हैं और तुलना करने के लिए सरल हैं क्योंकि वे एक ही पैमाने पर हैं। यह याद रखना महत्वपूर्ण है कि कोई भी मूल्य सीमा तब तक कार्य करेगी जब तक विभिन्न चर सुसंगत हैं।
निर्णय लेने के लिए उपयोगिता-आधारित एआई को नियोजित करने का रहस्य प्रत्येक कार्रवाई के लिए एक उपयोगिता स्कोर (जिसे वजन के रूप में भी जाना जाता है) असाइन करना है जो एआई एजेंट उच्चतम मूल्य वाले एक का चयन करने से पहले कर सकता है। अधिकांश गेमिंग दुनिया के गैर-निर्धारण वाद के कारण, सटीक उपयोगिता निर्धारित करना अक्सर असंभव होता है। यदि आप किसी गतिविधि के परिणामों की भविष्यवाणी नहीं कर सकते हैं, तो यह तय करना मुश्किल है कि यह बेहतर होगा या नहीं। जहां उपयोगिता सिद्धांत सबसे अधिक सहायक है, वह इसके मूल में है। उपयोगिता-आधारित एआई अपर्याप्त ज्ञान के आधार पर "सबसे अच्छा अनुमान" बनाने में अत्यधिक अच्छा है, एक उचित ग्रेडिंग सिस्टम दिया गया है। सबसे आम विधि प्रत्येक कल्पनीय परिणाम के उपयोगिता मूल्य को उसकी संभावना के अनुसार वजन करना है, फिर इन भारित स्कोर को जोड़ें।

अपेक्षित उपयोगिता (EU) = $\sum_{i=1}^{n} Di\ \ (Desire\ Outputs)\ Pi\ (Probability)$

अधिकतम अपेक्षित उपयोगिता (एमईयू) सिद्धांत के अनुसार, एक तर्कसंगत एजेंट को कार्रवाई के पाठ्यक्रम का चयन करना चाहिए जो उनकी अपेक्षित उपयोगिता को अधिकतम करेगा।

क्रिया = argmax यूरोपीय संघ

उदाहरण: सप्ताह में पांच दिन अंकगणित में आपकी सहायता के लिए एक ट्यूटर को काम पर रखने पर विचार करें। लेकिन पहली योजना के विपरीत, ट्यूटर हर हफ्ते कम से कम दो या तीन बार सुलभ नहीं है। क्या यह आपको खुश करेगा? यदि ट्यूटर आपकी अपेक्षाओं को पूरा करता है, तो क्या आप उसी राशि का भुगतान करने के लिए तैयार होंगे? आमतौर पर, जवाब नहीं है। कोई व्यक्ति जो सप्ताह में पांच दिनों के लिए ट्यूशन फीस का भुगतान करता है, वह ट्यूशन के घंटों की संख्या प्राप्त करने की उम्मीद कर सकता है। उपयोगिता अधिकतमकरण इसे संदर्भित करता है।
प्रदर्शन उपायों की अवधारणा और एमईयू सिद्धांत स्पष्ट रूप से संबंधित हैं। उन संभावित परिस्थितियों के बारे में सोचें जिनके परिणामस्वरूप एक एजेंट के पास एक विशिष्ट परसेप्ट इतिहास हो सकता है, साथ ही साथ विभिन्न एजेंट जो हम बना सकते हैं। एक एजेंट को अधिकतम संभव प्रदर्शन स्कोर मिलेगा यदि वे एक उपयोगिता फ़ंक्शन को अधिकतम करने के लिए कार्रवाई करते हैं जो प्रदर्शन माप (सभी संभावित वातावरणों पर औसत) को सटीक रूप से दर्शाता है। यह एमईयू आधार की मुख्य रक्षा के रूप में कार्य करता है। यह कथन एक टॉटोलॉजी प्रतीत हो सकता है, लेकिन यह वास्तव में तर्कसंगतता के एक वैश्विक, बाहरी मानक से एक महत्वपूर्ण बदलाव का प्रतिनिधित्व करता है - पर्यावरणीय इतिहास में प्रदर्शन माप - एक स्थानीय, आंतरिक मानक - अगले राज्य पर लागू उपयोगिता फ़ंक्शन का अधिकतम।

6.4 उपयोगिता फ़ंक्शन: आर्टिफिशियल इंटेलिजेंस (एआई) के संदर्भ में, उपयोगिता कार्य निर्णय लेने की प्रक्रियाओं में महत्वपूर्ण भूमिका निभाते हैं, विशेष रूप से मार्कोव निर्णय प्रक्रियाओं (एमडीपी) के ढांचे में। एमडीपी गणितीय मॉडल हैं जिनका उपयोग उन स्थितियों में निर्णय लेने की समस्याओं का प्रतिनिधित्व करने के लिए किया जाता है जहां परिणाम कार्यों और स्टोकेस्टिक घटनाओं दोनों से प्रभावित होते हैं।

एमडीपी के संदर्भ में एक उपयोगिता फ़ंक्शन पर्यावरण में किसी परिणाम या राज्य की वांछनीयता या अच्छाई को निर्धारित करता है। यह प्रत्येक राज्य या राज्य-कार्रवाई जोड़ी को एक संख्यात्मक मूल्य प्रदान करता है, जो उस स्थिति में होने या उस कार्रवाई को करने की अपेक्षित दीर्घकालिक वांछनीयता या उपयोगिता को दर्शाता है। उपयोगिता फ़ंक्शन एजेंट की प्राथमिकताओं के माप के रूप में कार्य करता है और अपेक्षित उपयोगिता को अधिकतम करके इसकी निर्णय लेने की प्रक्रिया का मार्गदर्शन करता है।

उपयोगिता फ़ंक्शन एक एजेंट को विभिन्न परिणामों के समग्र मूल्य या वांछनीयता के संदर्भ में अपने कार्यों के परिणामों का मूल्यांकन करने की अनुमति देता है। अधिक अनुकूल परिणामों के लिए उच्च उपयोगिता मूल्यों को असाइन करके, एजेंट सूचित निर्णय ले सकता है जो समय के साथ इसकी अपेक्षित उपयोगिता को अधिकतम करता है। अपेक्षित उपयोगिता को अधिकतम करने की यह धारणा कई एआई एल्गोरिदम और निर्णय लेने के ढांचे के लिए केंद्रीय है।

एमडीपी को हल करने के लिए, एजेंट आमतौर पर मूल्य पुनरावृत्ति या नीति पुनरावृत्ति जैसे एल्गोरिदम को नियोजित करता है। ये एल्गोरिदम विभिन्न कार्यों से जुड़े संक्रमण संभावनाओं और तत्काल पुरस्कारों को ध्यान में रखते हुए, उनके उत्तराधिकारी राज्यों की अपेक्षित उपयोगिताओं के आधार पर राज्यों के उपयोगिता मूल्यों को पुन: अपडेट करते हैं। अंतिम स्थिति या लक्ष्य स्थिति (ओं) से पीछे उपयोगिता मूल्यों का प्रचार करके, एजेंट इष्टतम नीति की गणना कर सकता है - कार्यों का अनुक्रम जो लंबे समय में अपेक्षित उपयोगिता को अधिकतम करता है।

एक बार उपयोगिता फ़ंक्शन स्थापित हो जाने और इष्टतम नीति प्राप्त होने के बाद, एजेंट वास्तविक समय परिदृश्यों में निर्णय लेने के लिए इसका उपयोग कर सकता है। वर्तमान स्थिति के आधार पर विभिन्न क्रियाओं की अपेक्षित उपयोगिता का मूल्यांकन करके, एजेंट दिए गए वातावरण में अपने व्यवहार को अनुकूलित करते हुए, उच्चतम अपेक्षित उपयोगिता के साथ कार्रवाई का चयन कर सकता है।

यह ध्यान देने योग्य है कि उपयोगिता कार्यों को एजेंट या कार्य की विशिष्ट वरीयताओं और उद्देश्यों के आधार पर अनुकूलित किया जा सकता है। उपयोगिता फ़ंक्शन की पसंद समस्या डोमेन और इसमें शामिल विशिष्ट ट्रेड-ऑफ और विचारों पर निर्भर करती है। विभिन्न उपयोगिता फ़ंक्शन वांछनीयता के विभिन्न पहलुओं को पकड़ सकते हैं, जैसे कि जोखिम विचलन, समय प्राथमिकताएं, या कार्य-विशिष्ट मानदंड।

6.5 मान पुनरावृत्ति: मूल्य पुनरावृत्ति एक एल्गोरिथ्म है जिसका उपयोग आमतौर पर मार्कोव निर्णय प्रक्रियाओं (एमडीपी) को हल करने के लिए आर्टिफिशियल इंटेलिजेंस (एआई) में किया जाता है। एमडीपी गणितीय मॉडल हैं जो उन स्थितियों में निर्णय लेने की समस्याओं का प्रतिनिधित्व करते हैं जहां परिणाम कार्यों और स्टोकेस्टिक घटनाओं दोनों से प्रभावित होते हैं।

मूल्य पुनरावृत्ति का लक्ष्य इष्टतम मूल्य फ़ंक्शन को ढूंढना है, जो एक इष्टतम नीति के तहत किसी विशेष राज्य में होने की अपेक्षित दीर्घकालिक उपयोगिता का प्रतिनिधित्व करता है। मान फ़ंक्शन प्रत्येक राज्य को एक संख्यात्मक मान प्रदान करता है, जो उस स्थिति में होने की अपेक्षित उपयोगिता को दर्शाता है।

मान पुनरावृत्ति एल्गोरिथ्म पुनरावर्ती रूप से मान फ़ंक्शन को अद्यतन करता है जब तक कि यह इष्टतम मानों में अभिसरण न हो जाए। एल्गोरिथ्म निम्नानुसार काम करता है:

1. मान फ़ंक्शन प्रारंभ करें: MDP में प्रत्येक राज्य के लिए एक प्रारंभिक मान सेट करें।
2. अभिसरण तक पुनरावृत्ति करें: ए। एमडीपी में प्रत्येक राज्य के लिए:
    - अपेक्षित तत्काल इनाम और परिणामी राज्यों की अपेक्षित भविष्य की उपयोगिता पर विचार करके प्रत्येक संभावित कार्रवाई के मूल्य की गणना करें।
    - तत्काल इनाम और अपेक्षित भविष्य की उपयोगिता के योग को अधिकतम करने वाली कार्रवाई का चयन करके राज्य के मूल्य को अपडेट करें।

    b. अभिसरण की जाँच करें: सभी राज्यों के लिए अद्यतन मानों और पिछले मानों के बीच अधिकतम अंतर की गणना करें। यदि यह अंतर पूर्व निर्धारित सीमा से नीचे आता है, तो एल्गोरिथ्म अभिसरण हो गया है, और इष्टतम मान फ़ंक्शन पाया गया है।
3. एक बार जब मूल्य पुनरावृत्ति एल्गोरिथ्म अभिसरण हो जाता है, तो इष्टतम नीति को उस कार्रवाई का चयन करके प्राप्त किया जा सकता है जो तत्काल इनाम के योग और प्रत्येक राज्य के लिए अपेक्षित भविष्य की उपयोगिता को अधिकतम करता है।

मान पुनरावृत्ति एल्गोरिथ्म कुछ शर्तों के तहत इष्टतम मूल्य फ़ंक्शन के अभिसरण की गारंटी देता है, जैसे कि एक परिमित स्थिति और कार्रवाई स्थान होना और भविष्य के पुरस्कारों को छूट देना। यह बेलमैन के समीकरण की अवधारणा का लाभ उठाता है, जो एक राज्य के मूल्य और उसके उत्तराधिकारी राज्यों के मूल्यों के बीच संबंध को व्यक्त करता है।

प्रत्येक पुनरावृत्ति में बेलमैन अपडेट को बार-बार लागू करके, मूल्य पुनरावृत्ति प्रभावी रूप से पूरे एमडीपी में अपेक्षित उपयोगिता जानकारी का प्रचार करती है, इष्टतम मूल्य फ़ंक्शन में अभिसरण करती है। परिणामी मूल्य प्रत्येक राज्य की वांछनीयता या अच्छाई का एक उपाय प्रदान करते हैं, इष्टतम निर्णय लेने की सुविधा प्रदान करते हैं।

एआई अनुप्रयोगों में मूल्य पुनरावृत्ति का व्यापक रूप से उपयोग किया जाता है जिसमें अनिश्चितता के तहत अनुक्रमिक निर्णय लेना शामिल होता है, जैसे रोबोटिक्स, गेम खेलना और संसाधन आवंटन। यह एमडीपी को हल करने और दीर्घकालिक उपयोगिता या पुरस्कारों को अधिकतम करने के लिए इष्टतम नीति निर्धारित करने के लिए एक कम्प्यूटेशनल रूप से कुशल दृष्टिकोण प्रदान करता है।

6.6 नीति पुनरावृत्ति: मार्कोव निर्णय प्रक्रियाओं (एमडीपी) को हल करने के लिए नीति पुनरावृत्ति आमतौर पर आर्टिफिशियल इंटेलिजेंस (एआई) में उपयोग किया जाने वाला एक और एल्गोरिदम है। मूल्य पुनरावृत्ति की तरह, नीति पुनरावृत्ति का उद्देश्य इष्टतम नीति खोजना है - कार्यों का अनुक्रम जो एमडीपी में अपेक्षित उपयोगिता या पुरस्कार को अधिकतम करता है।

नीति पुनरावृत्ति में दो मुख्य चरण होते हैं: नीति मूल्यांकन और नीति सुधार। इन चरणों को तब तक दोहराया जाता है जब तक कि नीति इष्टतम नीति में परिवर्तित नहीं हो जाती।

यहां नीति पुनरावृत्ति एल्गोरिथ्म का अवलोकन किया गया है:

1. नीति को प्रारंभ करें: एक मनमानी प्रारंभिक नीति से शुरू करें, जो एमडीपी में प्रत्येक राज्य को एक कार्रवाई प्रदान करता है।
2. नीति मूल्यांकन:
    - अभिसरण तक पुनरावृत्ति करें:
        - एमडीपी में प्रत्येक राज्य के लिए:
            - वर्तमान नीति के तहत अपेक्षित तत्काल इनाम और परिणामी राज्यों की अपेक्षित भविष्य की उपयोगिता पर विचार करके राज्य के मूल्य की गणना करें।
        - अभिसरण की जाँच करें: सभी राज्यों के लिए अद्यतन मानों और पिछले मानों के बीच अधिकतम अंतर की गणना करें। यदि यह अंतर पूर्व निर्धारित सीमा से नीचे आता है, तो नीति मूल्यांकन चरण अभिसरण हो गया है।
3. नीति में सुधार:
    - एमडीपी में प्रत्येक राज्य के लिए:
        - उस कार्रवाई का चयन करके नीति को अपडेट करें जो तत्काल इनाम के योग और वर्तमान स्थिति के लिए अपेक्षित भविष्य की उपयोगिता को अधिकतम करता है।
    - यदि नीति पिछले पुनरावृत्ति से नहीं बदली है, तो एल्गोरिथ्म इष्टतम नीति में परिवर्तित हो गया है।
4. चरण 2 और 3 को तब तक दोहराएँ जब तक कि नीति अभिसरण न हो जाए.

नीति पुनरावृत्ति कुछ शर्तों के तहत इष्टतम नीति के अभिसरण की गारंटी देती है, जैसे कि एक परिमित स्थिति और कार्रवाई स्थान होना। यह नीति के मूल्य समारोह की अवधारणा का लाभ उठाता है, जो एक विशिष्ट नीति का पालन करने की अपेक्षित उपयोगिता का प्रतिनिधित्व करता है।

नीति मूल्यांकन के दौरान, एल्गोरिथ्म प्रत्येक राज्य की अपेक्षित उपयोगिता का अनुमान लगाकर वर्तमान नीति के लिए मूल्य फ़ंक्शन की गणना करता है। इस कदम में वर्तमान नीति के तहत उत्तराधिकारी राज्यों की तत्काल इनाम और अपेक्षित भविष्य की उपयोगिता पर विचार करना शामिल है।

नीति सुधार चरण में, एल्गोरिथ्म उन कार्यों का चयन करके नीति को अपडेट करता है जो तत्काल इनाम के योग और प्रत्येक राज्य के लिए अपेक्षित भविष्य की उपयोगिता को अधिकतम करते हैं। यह प्रक्रिया धीरे-धीरे लालचपूर्वक उन कार्यों का चयन करके नीति में सुधार करती है जो उच्च अपेक्षित उपयोगिता की ओर ले जाती हैं।

नीति मूल्यांकन और नीति सुधार को दोहराने से, नीति पुनरावृत्ति इष्टतम नीति में परिवर्तित हो जाती है। परिणामी नीति एमडीपी में अपेक्षित उपयोगिता या पुरस्कारों को अधिकतम करने के लिए प्रत्येक राज्य में की जाने वाली सर्वोत्तम कार्रवाई का प्रतिनिधित्व करती है।

नीति पुनरावृत्ति एमडीपी को हल करने के लिए एआई में एक मौलिक एल्गोरिथ्म है और विशेष रूप से उपयोगी है जब पर्यावरण की गतिशीलता को अच्छी तरह से समझा जाता है और इसे सटीक रूप से मॉडलिंग किया जा सकता

है। यह आमतौर पर विभिन्न एआई अनुप्रयोगों में नियोजित किया जाता है, जैसे रोबोटिक्स, गेम खेलना और संसाधन आवंटन, जहां अनिश्चितता के तहत निर्णय लेने की आवश्यकता होती है।

6.7 आंशिक रूप से अवलोकन योग्य एमडीपी (पीओएमडीपी): आंशिक रूप से अवलोकन योग्य मार्कोव निर्णय प्रक्रिया (पीओएमडीपी) मार्कोव निर्णय प्रक्रियाओं (एमडीपी) का एक विस्तार है जो उन स्थितियों में निर्णय लेने की समस्याओं से निपटते हैं जहां पर्यावरण के एजेंट के अवलोकन अधूरे या शोर हैं। पीओएमडीपी का उपयोग आमतौर पर आर्टिफिशियल इंटेलिजेंस (एआई) में वास्तविक दुनिया के परिदृश्यों को मॉडल करने के लिए किया जाता है जिसमें अनिश्चितता और सीमित अवधारणात्मक जानकारी शामिल होती है।

पीओएमडीपी में, एजेंट के पास पर्यावरण की वास्तविक अंतर्निहित स्थिति तक सीधी पहुंच नहीं होती है, बल्कि इसके बजाय अवलोकन प्राप्त होता है जो अंतर्निहित स्थिति के साथ सहसंबद्ध होते हैं। पूर्ण अस्पष्टता की यह कमी निर्णय लेने में अतिरिक्त जटिलता का परिचय देती है, क्योंकि एजेंट को उपलब्ध टिप्पणियों और पिछले कार्यों के आधार पर संभावित राज्यों पर अपने विश्वास या संभाव्यता वितरण के बारे में तर्क देना चाहिए।

एआई के संदर्भ में पीओएमडीपी के प्रमुख घटकों और अवधारणाओं में शामिल हैं:

1. विश्वास की स्थिति: पीओएमडीपी में विश्वास की स्थिति पर्यावरण की वास्तविक अंतर्निहित स्थिति के बारे में एजेंट के वर्तमान व्यक्तिपरक विश्वास का प्रतिनिधित्व करती है। यह कार्यों और टिप्पणियों के इतिहास पर आधारित सभी संभावित राज्यों पर एक संभाव्यता वितरण है।
2. अवलोकन मॉडल: अवलोकन मॉडल पर्यावरण की अंतर्निहित स्थिति को देखते हुए कुछ टिप्पणियों को देखने की संभावना का वर्णन करता है। यह एजेंट की टिप्पणियों से जुड़ी अनिश्चितता और शोर को पकड़ता है।
3. संक्रमण मॉडल: संक्रमण मॉडल एजेंट के कार्यों के आधार पर एक राज्य से दूसरे राज्य में संक्रमण की संभावनाओं को परिभाषित करता है। यह पर्यावरण की गतिशीलता का प्रतिनिधित्व करता है।
4. इनाम मॉडल: इनाम मॉडल पीओएमडीपी में विभिन्न कार्यों और राज्यों से जुड़े तत्काल पुरस्कारों को निर्दिष्ट करता है। एजेंट का उद्देश्य समय के साथ अपेक्षित संचयी इनाम को अधिकतम करना है।
5. नीति: पीओएमडीपी में एक नीति विश्वास राज्यों से कार्यों तक एक मानचित्रण है। यह वर्तमान विश्वास स्थिति के आधार पर कार्यों का चयन करके एजेंट की निर्णय लेने की प्रक्रिया का मार्गदर्शन करता है।

अनिश्चितता और आंशिक अस्पष्टता के कारण पीओएमडीपी को हल करना चुनौतीपूर्ण है। पीओएमडीपी को संबोधित करने के लिए कई एल्गोरिदम और तकनीकें विकसित की गई हैं, जैसे:

1. मूल्य पुनरावृत्ति एल्गोरिदम: एमडीपी के समान, मूल्य पुनरावृत्ति एल्गोरिदम को पीओएमडीपी के लिए अनुकूलित किया जा सकता है। ये एल्गोरिदम एक इष्टतम नीति खोजने के लिए मूल्य फ़ंक्शन या क्यू-फ़ंक्शन को पुन: अपडेट करते हैं। हालांकि, पीओएमडीपी मूल्य पुनरावृत्ति को विश्वास राज्यों की निरंतर प्रकृति के कारण अनुमान या अनुमान की आवश्यकता होती है।
2. विश्वास अंतरिक्ष खोज एल्गोरिदम: ये एल्गोरिदम एक इष्टतम नीति की खोज करके संभावित विश्वास राज्यों के स्थान का पता लगाते हैं। उदाहरणों में मोंटे कार्लो ट्री सर्च (एमसीटीएस) और कण फ़िल्टरिंग दृष्टिकोण शामिल हैं।

3. बिंदु-आधारित विधियां: ये विधियां प्रतिनिधि विश्वास बिंदुओं के एक सेट को बनाए रखने और इन बिंदुओं के आधार पर नीति को अनुकूलित करके विश्वास स्थान का अनुमान लगाती हैं। उदाहरणों में बिंदु-आधारित मूल्य पुनरावृत्ति (पीबीवीआई) और आंशिक रूप से अवलोकन योग्य मोंटे कार्लो प्लानिंग (पीओएमसीपी) शामिल हैं।

पीओएमडीपी रोबोटिक्स, स्वायत्त प्रणालियों, स्वास्थ्य देखभाल और प्राकृतिक भाषा प्रसंस्करण सहित विभिन्न डोमेन में अनुप्रयोग पाते हैं, जहां निर्णय अनिश्चित और आंशिक रूप से अवलोकन योग्य वातावरण में होता है। अनिश्चितता को स्पष्ट रूप से मॉडलिंग करके और विश्वास राज्यों को शामिल करके, पीओएमडीपी एआई सिस्टम को अधूरे और शोर पूर्ण टिप्पणियों के साथ भी सूचित निर्णय लेने में सक्षम बनाता है।

सारांश:

मार्कोव निर्णय प्रक्रिया (एमडीपी) गणितीय मॉडल हैं जो आर्टिफिशियल इंटेलिजेंस (एआई) में निर्णय लेने की नींव बनाते हैं। एमडीपी अनुक्रमिक निर्णय समस्याओं का प्रतिनिधित्व करने के लिए एक रूपरेखा प्रदान करते हैं जहां परिणाम कार्यों और स्टोकेस्टिक घटनाओं दोनों से प्रभावित होते हैं।

एक एमडीपी में, एक एजेंट असतत समय चरणों की एक श्रृंखला में एक पर्यावरण के साथ बातचीत करता है। प्रत्येक चरण में, एजेंट अपनी वर्तमान स्थिति के आधार पर एक कार्रवाई चुनता है। पर्यावरण तब चुनी हुई कार्रवाई और एक संभाव्य संक्रमण मॉडल के आधार पर एक नई स्थिति में बदल जाता है। एजेंट को चुनी गई कार्रवाई और नई स्थिति के आधार पर तत्काल इनाम मिलता है। एजेंट का लक्ष्य एक इष्टतम नीति खोजना है जो इसके दीर्घकालिक अपेक्षित पुरस्कार या उपयोगिता को अधिकतम करता है।

एमडीपी कई प्रमुख घटकों पर भरोसा करते हैं। इनमें राज्य स्थान शामिल है, जो पर्यावरण के सभी संभावित राज्यों का प्रतिनिधित्व करता है, एक्शन स्पेस, जो एजेंट के लिए उपलब्ध कार्यों के सेट को परिभाषित करता है, और संक्रमण मॉडल, जो चुनी हुई कार्रवाई के आधार पर एक राज्य से दूसरे राज्य में संक्रमण की संभावनाओं का वर्णन करता है।

एमडीपी ढांचे में एक इनाम मॉडल भी शामिल है जो प्रत्येक राज्य या राज्य-कार्रवाई जोड़ी को एक संख्यात्मक मूल्य प्रदान करता है, जो उस राज्य या कार्रवाई की तत्काल वांछनीयता का प्रतिनिधित्व करता है। एजेंट का उद्देश्य एक ऐसी नीति सीखना है जो समय के साथ अपेक्षित संचयी पुरस्कारों को अधिकतम करता है।

एमडीपी को हल करने के लिए, विभिन्न एल्गोरिदम नियोजित किए जाते हैं, जैसे मूल्य पुनरावृत्ति और नीति पुनरावृत्ति। मान पुनरावृत्ति पुनरावर्ती मूल्य फ़ंक्शन को अपडेट करता है, जो प्रत्येक राज्य के लिए अपेक्षित दीर्घकालिक उपयोगिता या पुरस्कारों का प्रतिनिधित्व करता है, जब तक कि यह इष्टतम मूल्यों पर अभिसरण न हो। नीति पुनरावृत्ति नीति मूल्यांकन के बीच वैकल्पिक होती है, जो किसी दिए गए नीति के तहत मूल्य फ़ंक्शन का अनुमान लगाती है, और नीति सुधार, जो वर्तमान मूल्य फ़ंक्शन के आधार पर नीति को अपडेट करता है।

एमडीपी में रोबोटिक्स, गेम प्लेइंग, संसाधन आवंटन और अनुकूलन समस्याओं सहित एआई में व्यापक अनुप्रयोग हैं। वे अनिश्चितता के तहत निर्णय लेने के लिए एक औपचारिक ढांचा प्रदान करते हैं और एआई

एजेंटों को दीर्घकालिक परिणामों पर विचार करके और भविष्य के परिणामों के साथ तत्काल पुरस्कारों को संतुलित करके इष्टतम विकल्प बनाने में सक्षम बनाते हैं।

सारांश में, एमडीपी एआई में निर्णय लेने की समस्याओं को मॉडलिंग और हल करने के लिए एक मौलिक ढांचे के रूप में काम करते हैं। राज्यों, कार्यों, पुरस्कारों और संक्रमण संभावनाओं को शामिल करके, एमडीपी एआई एजेंटों को जटिल वातावरण को नेविगेट करने, इष्टतम नीतियों को सीखने और अनिश्चित और गतिशील परिदृश्यों में इंटेलिजेंसमान निर्णय लेने में सक्षम बनाता है।

प्रमुख शब्द:

- मार्कोव निर्णय प्रक्रिया
- जैविक न्यूरॉन
- आर्टिफीसियल न्यूरॉन
- अक्षतंतु
- राशि
- अंतरग्रथनअ
- सक्रियण
- मैककुलोच-पिट्स न्यूरॉन
- परसेप्ट्रॉन
- वजन
- वर्गीकरण
- प्रतिगमन
- क्लस्टरिंग
- रैखिक रूप से वियोज्य।
- शुरूआत
- सिंगल लेयर फीड फॉरवर्ड एएनएन
- मल्टीलेयर फीड फॉरवर्ड एएनएन
- प्रतिस्पर्धी ANN
- आवर्तक ANN
- छिपी हुई परतें
- सक्रियण फ़ंक्शन
- पहचान समारोह
- मूल चरण फ़ंक्शन
- थ्रेशोल्ड फ़ंक्शन
- बाइपोलर स्टेप फंक्शन
- बाइपोलर थ्रेशोल्ड फंक्शन
- सिग्मोइड फ़ंक्शन
- बाइपोलर सिग्मोइड फंक्शन
- हाइपरबॉलिक स्पर्शरेखा फ़ंक्शन
- पर्यवेक्षित शिक्षा
- असुरक्षित शिक्षा।
- हेब सीखने का नियम

- परसेप्ट्रॉन लर्निंग रूल
- विड्रो-हॉफ सीखने का नियम
- विजेता सभी सीखने के नियम लेता है

अभ्यास

1. बहुत संक्षिप्त प्रकार के प्रश्न:[2 अंक]

a) आर्टिफीसियल इंटेलिजेंस में मार्कोव निर्णय प्रक्रिया क्या है?
b) मार्कोव निर्णय प्रक्रिया की भूमिका क्या है?
c) मार्कोव निर्णय प्रक्रिया के तीन तत्व क्या हैं?
d) आर्टिफीसियल इंटेलिजेंस में हम्म क्या है?
e) मार्कोव निर्णय प्रक्रिया का एक उदाहरण क्या है?
f) मार्कोव निर्णय प्रक्रिया का सूत्र क्या है?

2. लघु प्रकार के प्रश्न: [5 अंक]

a) मार्कोव की निर्णय प्रक्रिया क्या है? उदाहरण दीजिए।
b) उदाहरण के साथ छिपे हुए मार्कोव मॉडल की व्याख्या करें।
c) मार्कोव चेन क्या है? उपयुक्त उदाहरण के साथ समझाइए।
d) मूल्य पुनरावृत्ति का मुख्य लक्ष्य क्या है?
e) मार्कोव निर्णय प्रक्रिया में मूल्य पुनरावृत्ति कैसे काम करती है?

3. दीर्घ प्रकार के प्रश्न: [10 अंक]

a) हिडन मार्कोव मॉडल क्या है और इसका उपयोग किस तरह से किया जाता है?।
b) मार्कोव मॉडल में संदर्भी स्थितियों का सेट कैसे प्रदर्शित किया जाता है?
c) विभिन्न स्थितियों और परिणामों के बीच संबंध को निर्धारित करने के लिए कौन-कौन से मान्यताएं और साधारणीकरणें प्रयोग किए जाते हैं?
d) मार्कोव मॉडल का उपयोग किस क्षेत्रों में किया जाता है?
e) प्रवाहन प्राकृतिकी और प्रवर्तक प्राकृतिकी क्या होती है और इनका क्या महत्व होता है?

ग्रंथ-सूची

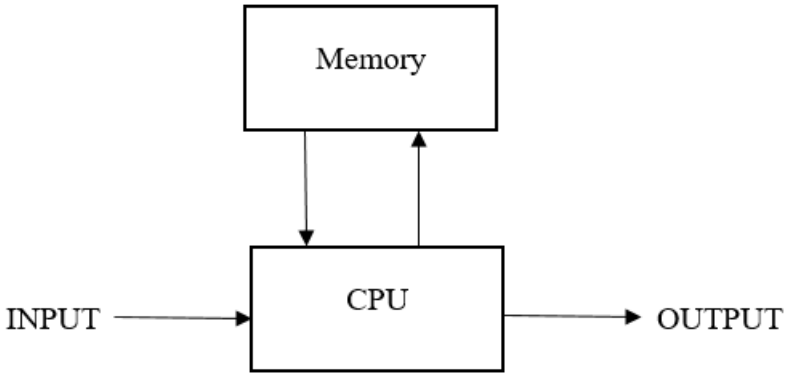
- एंडरसन, जे ए और रोसेनफील्ड, ई। (1988). *न्यूरोकंप्यूटिंग: अनुसंधान की नींव।*. कैम्ब्रिज, एमए, एमआईटी प्रेस।
- वॉन न्यूमैन, जे (1958)। *कंप्यूटर और मस्तिष्क*. न्यू हेवन: येल यूनिवर्सिटी प्रेस।
- रोसेनब्लैट, एफ (1959)। परसेप्ट्रॉन में सांख्यिकीय अलगाव के दो प्रमेय। *विचार प्रक्रिया का मशीनीकरण: राष्ट्रीय भौतिक प्रयोगशाला में संगोष्ठी की कार्यवाही* लंदन, पीपी - 421-456।
- रोसेनब्लैट, एफ (1962)। *न्यूरोडायनामिक्स के सिद्धांत*. स्पार्टन बुक्स।
- मिंस्की, एम एंड पेपर, एस (1969)। *परसेप्ट्रोन*. एमआईटी प्रेस।
- - मैककुलोच, डब्ल्यू एंड पिट्स, डब्ल्यू (1943)। तंत्रिका गतिविधि में निहित विचारों का एक तार्किक कैलकुलेशन। *गणितीय बायोफिज़िक्स का बुलेटिन*, 7, 99. 115-133.
- ब्लॉक, एचडी (1962)। परसेप्ट्रॉन: मस्तिष्क के कामकाज के लिए एक मॉडल। *आधुनिक भौतिकी के लिए समीक्षाएं*. खंड 34, पीपी 123-135।
- - कोहेन, एमए और ग्रॉसबर्ग, एस (1983)। प्रतिस्पर्धी तंत्रिका नेटवर्क द्वारा वैश्विक पैटर्न गठन और समानांतर स्मृति भंडारण की पूर्ण स्थिरता। *सिस्टम, मैन और साइबरनेटिक्स पर आईईईई लेनदेन*, एसएमसी -13, पीपी 815-826।
- ग्रॉसबर्ग, एस (1973)। *मन और मस्तिष्क का अध्ययन*. बोस्टन, रिडेल।
- हेब, डी ओ (1949)। *व्यवहार का संगठन*. न्यूयॉर्क, जॉन विली एंड संस।
- रॉय, एस एंड चक्रवर्ती, यू (2013)। *सॉफ्ट कंप्यूटिंग का परिचय: न्यूरो-फजी और जेनेटिक एल्गोरिदम*. पियर्सन।
- पाढ़ी, एनपी और साइमन, एसपी (2015)। *MATLAB प्रोग्रामिंग के साथ सॉफ्ट कंप्यूटिंग*. ऑक्सफोर्ड यूनिवर्सिटी प्रेस।

अध्याय 7

न्यूरल नेटवर्क का उपयोग कर मशीन लर्निंग : मूल बातें

7.1 परिचय

पारंपरिक कंप्यूटर आम तौर पर निर्देशों के अनुक्रम के माध्यम से काम करते हैं जो वांछित आउटपुट का उत्पादन करने के लिए इनपुट डेटा की संपत्ति पर गणना संचालन करते हैं। अभिकलन संक्रियाओं को निष्पादित करने वाले घटक को आमतौर पर वॉन न्यूमैन कंप्यूटर आर्किटेक्चर द्वारा परिभाषित किया जाता है जिसमें एक अलग प्रोसेसर, मेमोरी, इनपुट और आउटपुट घटक होते हैं जैसा कि चित्र 1 में दिखाया गया है। मेमोरी का उपयोग निर्देशों और डेटा दोनों को संग्रहीत करने के लिए किया जाता है और सेंट्रल प्रोसेसिंग यूनिट (सीपीयू) मेमोरी में संग्रहीत डेटा पर गणना करता है।



चित्र 7.1 वॉन न्यूमैन कंप्यूटर आर्किटेक्चर

पारंपरिक वॉन न्यूमैन आर्किटेक्चर आधारित कंप्यूटर बहुत कम समय में बड़ी गणितीय गणना करने के लिए काफी प्रभावी रहे हैं, इस प्रकार उन्हें वाणिज्य, स्वास्थ्य सेवा, संचार, शासन, अनुसंधान आदि जैसे हर क्षेत्र में विभिन्न संख्या क्रंचिंग नौकरियों में मनुष्यों की जगह लेने के लिए अग्रणी बनाया गया है।

फिर भी, कई दिन-प्रतिदिन की गतिविधियाँ हैं जो एक मानव कुछ ही छड़ों में कर सकता है, जबकि पारंपरिक कंप्यूटरों द्वारा हल किया जाना बहुत कठिन या असंभव होगा। इस तरह की गतिविधियां चेहरे की पहचान, वस्तु पहचान, प्राकृतिक भाषा प्रसंस्करण, भाषा समझ, भाषा अनुवाद, छवि वर्गीकरण, छवि क्लस्टरिंग, भविष्यवाणी और पूर्वानुमान आदि हो सकती हैं। ऊपर सूचीबद्ध ऐसे कार्य पारंपरिक कंप्यूटिंग एल्गोरिदम द्वारा किए जाने के लिए बहुत मुश्किल या लगभग असंभव हैं।

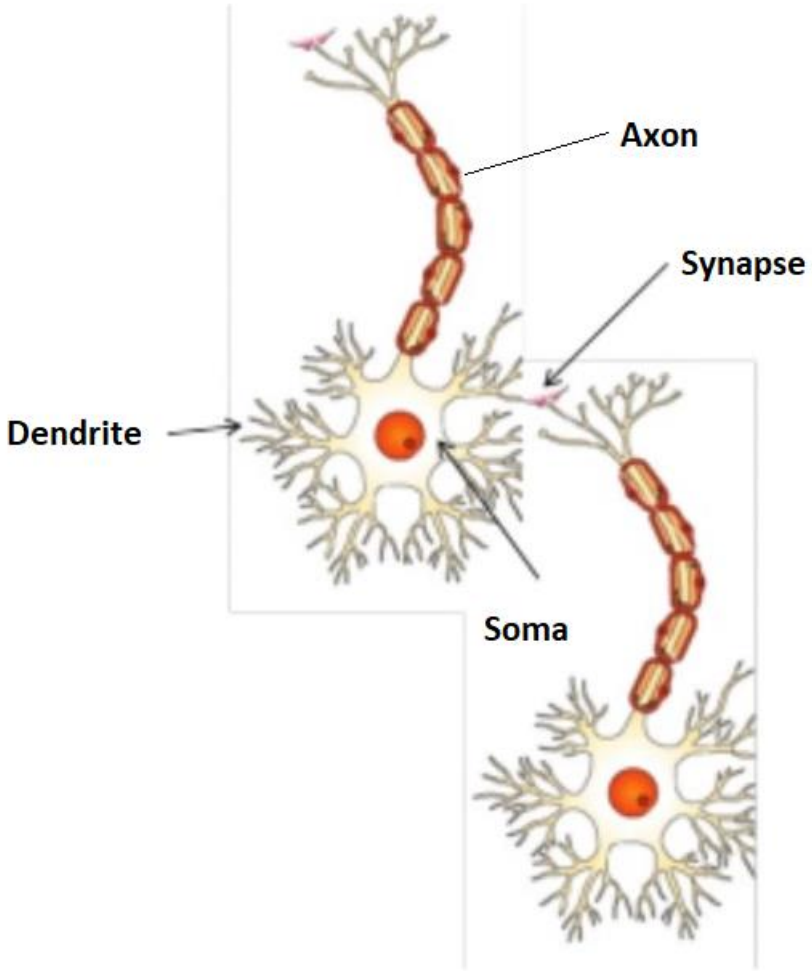
इस प्रकार, प्रसंस्करण संख्याओं और अंकगणितीय गणनाओं की तुलना में पारंपरिक कंप्यूटरों के संबंध में मानव मस्तिष्क का एक अलग नुकसान है, लेकिन जहां मानव मस्तिष्क कंप्यूटिंग के पारंपरिक रूप से ऊपर स्कोर करता है, पैटर्न के संदर्भ में काम करने के क्षेत्र में है। मानव मस्तिष्क जिसमें 100 बिलियन इंटरकनेक्टेड न्यूरॉन्स शामिल हैं, पैटर्न के संदर्भ में डेटा को संसाधित कर सकते हैं और पैटर्न पहचान कार्यों को बहुत कुशलता से कर सकते हैं।

आर्टिफिशियल न्यूरल नेटवर्क मानव मस्तिष्क के काम और संरचना से प्रेरित एक कंप्यूटिंग प्रकार है और इसका उद्देश्य आमतौर पर मानव मस्तिष्क द्वारा किए गए कार्यों को करने के लिए इसकी समानांतर और वितरित संरचना का उपयोग करना है।

यह अध्याय एक आर्टिफीसियल न्यूरॉन और एक आर्टिफीसियल न्यूरल नेटवर्क के विभिन्न आर्किटेक्चर की अवधारणा का परिचय देता है। यह न्यूरॉन्स के कई सक्रियण कार्यों का परिचय और आर्टिफीसियल तंत्रिका नेटवर्क द्वारा नियोजित कुछ लोकप्रिय सीखने की रणनीतियों का संक्षिप्त परिचय भी प्रदान करता है।

7.1.1 जैविक न्यूरॉन

मानव मस्तिष्क का मूल तत्व जैविक न्यूरॉन है। जैविक न्यूरॉन के मुख्य घटकों को चित्र 7.2.में दर्शाया गया है।



चित्र 7.2 जैविक न्यूरॉन की संरचना

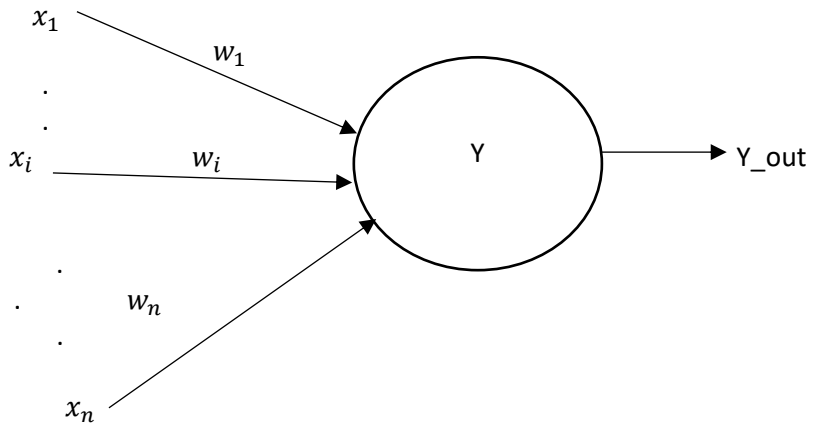
जैविक न्यूरॉन में तीन प्रमुख भाग होते हैं, अर्थात्, डेंड्राइट, सोमा और एक्सॉन। डेंड्राइट पड़ोसी न्यूरॉन्स से उत्तेजना इकट्ठा करते हैं और इसे सोमा या सेल बॉडी पर पारित करते हैं। सोमा डेंड्राइट से प्राप्त

उत्तेजनाओं को एकत्र करता है। यह 'फायर' करता है जब पर्याप्त उत्तेजना एकत्र होती है। जब एक न्यूरॉन फायर करता है तो यह अपनी उत्तेजना को स्थानांतरित करता है जो एक्सॉन के माध्यम से एक विद्युत आवेग के रूप में आगे जाती है। अंततः, यह उत्तेजना न्यूरोट्रांसमीटर नामक इलेक्ट्रोकेमिकल को नियोजित करके सिनैप्स के माध्यम से पड़ोसी न्यूरॉन्स को पारित की जाती है।

सिनैप्स का मानव मस्तिष्क के काम में प्रदर्शन करने के लिए एक महत्वपूर्ण हिस्सा है। यह एक वजन द्वारा एक न्यूरॉन के इनपुट सिग्नल को स्केल करता है। यदि कोई इनपुट है x और वजन है w, तो सेल बॉडी तक पहुंचने वाला कुल इनपुट उत्पाद है। $x \times w$. वजन का महत्व w सिनैप्स द्वारा प्रदान किया गया यह वजन इस तथ्य में निहित है कि अन्य सिनैप्टिक वजन के साथ यह वजन पारंपरिक वॉन-न्यूमैन आर्किटेक्चर आधारित डिजिटल कंप्यूटर के विपरीत तंत्रिका नेटवर्क में संग्रहीत ज्ञान का प्रतिनिधित्व करता है जहां ज्ञान को मेमोरी में एक प्रोग्राम के रूप में संग्रहीत किया जाता है। सिनैप्स द्वारा प्रदान किए गए वजन द्वारा दर्शाया गया यह ज्ञान समय और अनुभव के साथ संशोधित होता है जिससे अभ्यास के माध्यम से विभिन्न कौशल सीखने में मदद मिलती है।

7.1.2 आर्टिफीसियल न्यूरॉन

एक आर्टिफीसियल न्यूरॉन एक जैविक न्यूरॉन की संरचना और कार्यक्षमता के आधार पर एक कम्प्यूटेशनल मॉडल है। इसमें एक प्रसंस्करण तत्व, कई इनपुट और भारित किनारे होते हैं जो प्रत्येक इनपुट को प्रसंस्करण तत्व से जोड़ते हैं जैसा कि चित्र 7.3 में दिखाया गया है।



चित्र 7.3 आर्टिफीसियल न्यूरॉन की संरचना

एक आर्टिफीसियल न्यूरॉन में निम्नलिखित शामिल हैं: n इनपुट इकाइयों की संख्या $x_1, x_2, \ldots, x_n$, n वजन की संख्या $w_1, w_2, \ldots, w_n$ इनपुट इकाइयों, एक प्रसंस्करण इकाई के साथ जुड़ा हुआ है Y और एक एकल आउटपुट। Y_out.
कुल इनपुट Y_in प्रसंस्करण तत्व के लिए। Y इस प्रकार प्राप्त किया जाता है:

$$Y_in = x_1 w_1 + x_1 w_1 + \cdots + x_n w_n = \sum_{i=1}^{n} x_i w_i$$

(7.1)

वजन w_i इनपुट के साथ संबद्ध x_i यह सकारात्मक या नकारात्मक हो सकता है। एक सकारात्मक वजन का मतलब है कि संबंधित इनपुट का प्रसंस्करण इकाई Y पर उत्तेजक प्रभाव पड़ता है। यदि अन्यथा वजन नकारात्मक है, तो संबंधित इनपुट पर निरोधात्मक प्रभाव पड़ता है Y। प्रसंस्करण इकाई का उत्पादन Y नेट इनपुट का एक फ़ंक्शन है Y_in इस प्रकार वर्णित:

$$Y_out = f(Y_in) \tag{7.2}$$

फ़ंक्शन का सबसे सरल रूप $f()$ एक द्विआधारी चरण फ़ंक्शन है जिसे इस प्रकार परिभाषित किया गया है:

$$Y_out = f(Y_in) = \begin{cases} 1, & if\ Y_in > 0, \\ 0, & if\ Y_in \leq 0 \end{cases} \tag{7.3}$$

समीकरण 7.1 लेने पर हमें मिलता है:

$$Y_out = f(Y_in) = \begin{cases} 1, & if \sum_{i=1}^{n} x_i w_i > 0, \\ 0, & if \sum_{i=1}^{n} x_i w_i \leq 0 \end{cases} \tag{7.4}$$

जब एक गैर-शून्य सीमा θ समीकरण 7.3 और 7.4 का उपयोग किया जाता है:

$$Y_out = f(Y_in) = \begin{cases} 1, & if\ Y_in > \theta, \\ 0, & if\ Y_in \leq \theta \end{cases} \tag{7.5}$$

$$Y_out = f(Y_in) = \begin{cases} 1, & if \sum_{i=1}^{n} x_i w_i > \theta, \\ 0, & if \sum_{i=1}^{n} x_i w_i \leq \theta \end{cases} \tag{7.6}$$

फ़ंक्शन $f()$ आर्टिफीसियल न्यूरॉन और आउटपुट का सक्रियण फ़ंक्शन कहा जाता है Y_out प्रसंस्करण इकाई के सक्रियण के रूप में जाना जाता है Y।

7.2 मैककुलोच - पिट्स न्यूरल मॉडल

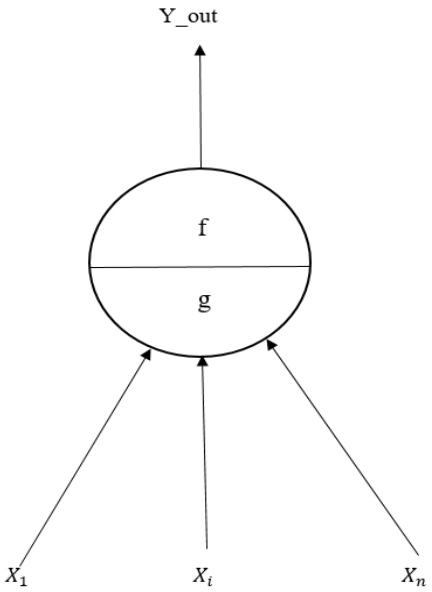
मैककुलोच-पिट्स न्यूरल मॉडल या एमपी न्यूरल मॉडल 1943 में वॉरेन मैककुलोच और वाल्टर पिट्स द्वारा सुझाया गया सबसे पहला आर्टिफीसियल न्यूरॉन मॉडल था। इसे बाइनरी थ्रेशोल्ड गेट के रूप में भी जाना जाता है। यह कम्प्यूटेशनल विज्ञान के क्षेत्र में मस्तिष्क में जैविक प्रक्रियाओं का पहला अनुप्रयोग था। चित्र 7.4 एक एमपी न्यूरॉन मॉडल की मूल संरचना को दर्शाता है। यह विभिन्न इनपुट से बना है $X_1, X_2, \ldots, X_n$ एकल प्रसंस्करण इकाई से कनेक्ट डे Y। इनपुट दो प्रकार के होते हैं, उत्तेजक या निरोधात्मक। एमपी न्यूरॉन का आउटपुट 0 होगा यदि इनपुट निरोधात्मक है अन्यथा यह 1 होगा।

प्रसंस्करण इकाई Y आगे दो कार्यों में विभाजित है। फ़ंक्शन $g()$ सभी इनपुट एकत्र करता है $X_1, X_2, \ldots, X_n$ और कार्य। फ़ंक्शन $f()$ इस एकत्रीकरण के आधार पर निर्णय लेता है।प्रसंस्करण इकाई का उत्पादन Y, $Y_out = 0$ यदि कोई इनपुट है X_i निरोधात्मक है, अन्यथा प्रसंस्करण तत्व के लिए कुल इनपुट Y, Y_in द्वारा दिया गया है:

$$Y_in = g(X_1, X_2, \ldots, X_n) = g(X) = \sum_{i=1}^{n} X_i \qquad (7.7)$$

एमपी न्यूरॉन एक सरल बाइनरी थ्रेशोल्ड सक्रियण फ़ंक्शन को नियोजित करता है। अगर θ थ्रेशोल्ड है, फिर प्रसंस्करण इकाई का उत्पादन Y इस प्रकार प्राप्त किया जाता है:

$$Y_out = f\big(g(X)\big) = \begin{cases} 1, & if\ g(X) \geq \theta, \\ 0, & otherwise. \end{cases} \qquad (7.8)$$

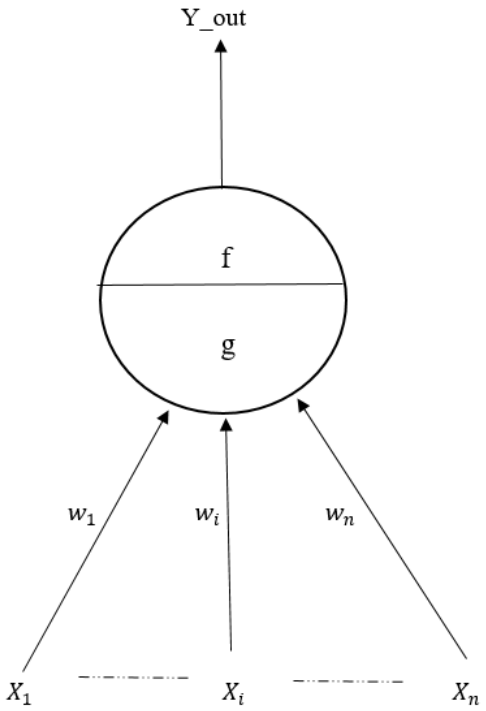


चित्र 7.4 मैककुलोच-पिट्स न्यूरॉन की संरचना

एमपी न्यूरॉन में एक एकल पैरामीटर होता है θ, जिसका अर्थ है कि इसमें जटिल गैर-रैखिक कार्यों को करने की सीमित क्षमता है। यह एमपी न्यूरल मॉडल को सरल रैखिक कार्यों के लिए उपयुक्त बनाता है जहां अंतर्निहित डेटा रैखिक रूप से वियोज्य है। जिन कार्यों को यह कर सकता है वे बूलियन वर्गीकरण की श्रेणी में आते हैं जिसमें इनपुट डेटा को आउटपुट वर्गों में से दो से संबंधित वर्गीकृत किया जाता है। इसमें एक दोष है कि यह केवल बूलियन मूल्यों को अपने प्रसंस्करण तत्व के इनपुट के रूप में स्वीकार कर सकता है। एमपी न्यूरॉन भी इसे कार्य करने के लिए किसी भी हेरिस्टिक इंटेलिजेंसमान सीखने के एल्गोरिदम को नियोजित नहीं करता है, इसके बजाय यह उचित थ्रेशोल्ड मान खोजने के लिए एक ब्रूट फोर्स सर्च करता है θ बूलियन वर्गीकरण कार्य करने के लिए। इसमें एक एकल पैरामीटर और कुछ संभावित इंटरसेप्ट मानों के कारण निश्चित ढलान का नुकसान भी है, इस प्रकार जटिल वर्गीकरण कार्यों को करने के लिए इसकी प्रयोज्यता को गंभीर रूप से सीमित किया गया है।

7.3 परसेप्ट्रॉन

परसेप्ट्रॉन तंत्रिका नेटवर्क के लिए शुरुआती मॉडलों में से एक था जिसमें 1962 में रोसेनब्लैट द्वारा प्रस्तावित इनपुट मूल्यों को कुछ महत्व प्रदान करने की विधि थी। यह संरचना में मैककुलोच-पिट्स न्यूरल मॉडल के समान है, सिवाय इसके कि प्रत्येक इनपुट मान में अब चित्र 7.5 द्वारा चित्रित एक संबद्ध वजन होता है।



चित्र 7.5 परसेप्ट्रॉन की संरचना

परसेप्ट्रॉन विभिन्न इनपुट से बना है। $X_1, X_2, \dots, X_n$ एकल प्रसंस्करण इकाई से कनेक्ट डे Y. इनपुट दो प्रकार के होते हैं, उत्तेजक या निरोधात्मक। इनपुट से प्रत्येक पथ X_i प्रसंस्करण तत्व के लिए। Y एक संबद्ध वजन है w_i इससे जुड़ा हुआ है। वजन w_i यदि इनपुट सकारात्मक है X_i उत्तेजक है और इनपुट होने पर इसका मूल्य नकारात्मक है X_i निरोधात्मक है। प्रसंस्करण तत्व के लिए कुल इनपुट Y द्वारा दिए गए भारित इनपुट का योग है:

$$Y_in = g(X_1w_1, X_2w_2, \dots, X_nw_n) = g(X.W) = \sum_{i=1}^{n} X_iw_i \tag{7.9}$$

समीकरण 7.7 को मैट्रिक्स रूप में इस प्रकार व्यक्त किया जा सकता है:

$$Y_in = \sum_{i=1}^{n} X_iw_i = [X_1\ X_2\ \dots\ X_n] \times \begin{bmatrix} w_1 \\ w_2 \\ . \\ . \\ . \\ w_n \end{bmatrix} \tag{7.10}$$

$$Y_in = X \times W^T \quad (7.11)$$

यहाँ X इनपुट वेक्टर है $X = [X_1\ X_2\ \dots\ X_n]$ और W वजन वेक्टर है $W = [w_1\ w_2\ \dots\ w_n]$.

परसेप्ट्रॉन भी मैककुलोच-पिट्स न्यूरल मॉडल के समान एक सरल बाइनरी थ्रेशोल्ड सक्रियण फ़ंक्शन को नियोजित करता है। अगर θ दहलीज है, फिर प्रसंस्करण इकाई का उत्पादन Y इस प्रकार प्राप्त किया जाता है:

$$Y_out = f\big(g(X)\big) = \begin{cases} 1, & if\ g(X) \geq\ \theta, \\ 0, & otherwise. \end{cases} \quad (7.12)$$

परसेप्ट्रॉन में दो पैरामीटर होते हैं, थ्रेशोल्ड। θ और वजन वेक्टर। $W = [w_1\ w_2\ \dots\ w_n]$, जिसका अर्थ है कि इसमें जटिल गैर-रैखिक कार्यों को करने की सीमित क्षमता भी है। यह परसेप्ट्रॉन मॉडल को सरल रैखिक कार्यों के लिए भी उपयुक्त बनाता है जहां अंतर्निहित डेटा रैखिक रूप से वियोज्य है। जिन कार्यों को यह कर सकता है वे बूलियन वर्गीकरण की श्रेणी में आते हैं जिसमें इनपुट डेटा को आउटपुट वर्गों में से दो से संबंधित वर्गीकृत किया जाता है। यह बूलियन मूल्यों के साथ-साथ वास्तविक मूल्यों को अपने प्रसंस्करण तत्व के इनपुट के रूप में स्वीकार कर सकता है। परसेप्ट्रॉन इसे कार्य करने के लिए एक हेरिस्टिक इंटेलिजेंसमान सीखने के एल्गोरिदम को नियोजित करता है जिसे परसेप्ट्रॉन लर्निंग एल्गोरिदम के रूप में जाना जाता है। इसमें दो मापदंडों और समायोज्य इंटरसेप्ट मूल्यों के कारण चर ढलान के फायदे भी हैं, इस प्रकार इसे एमपी न्यूरॉन मॉडल की तुलना में अधिक जटिल वर्गीकरण कार्यों के लिए अधिक शक्ति मिलती है, लेकिन यह अभी भी रैखिक अलगाववाद की समस्या से सीमित है और रैखिक समस्याओं को हल करने तक सीमित है।

7.4 न्यूरल नेटवर्क आर्किटेक्चर

एक आर्टिफीसियल न्यूरल **नेटवर्क में विभिन्न तरीकों से जुड़े कई आर्टिफीसियल न्यूरॉन्स होते हैं। कभी-कभी इन न्यूरॉन्स को परतों में परस्पर संबंध के साथ परतों में व्यवस्थित किया जाता है। नेटवर्क पूरी तरह से जुड़ा हो सकता है या नहीं भी हो सकता है और इन इंटरकनेक्शन की प्रकृति भी भिन्न हो सकती है। ये इंटरकनेक्शन या तो यूनिडायरेक्शनल या द्विदिश हो सकते हैं। यह खंड आर्टिफीसियल तंत्रिका नेटवर्क के प्रमुख आर्किटेक्चर का अवलोकन प्रस्तुत करता है।**

7.4.1 सिंगल लेयर फीड फॉरवर्ड एएनएन

सिंगल लेयर फीड फॉरवर्ड आर्टिफिशियल न्यूरल नेटवर्क सबसे सरल एएनएन आर्किटेक्चर है। इसमें आउटपुट न्यूरॉन्स के एक समूह से जुड़े इनपुट न्यूरॉन्स का एक समूह शामिल है। इनपुट न्यूरॉन्स किसी भी प्रसंस्करण शक्ति को नियोजित नहीं करते हैं, लेकिन वे केवल आउटपुट न्यूरॉन्स को इनपुट अग्रेषित करते हैं, इस प्रकार उन्हें एक परत का गठन नहीं माना जाता है। इसलिए, चित्र 7.6 में दिखाए गए सिंगल लेयर फीड फॉरवर्ड न्यूरल नेटवर्क में एकमात्र परत आउटपुट न्यूरॉन्स से बनी है। सिंगल लेयर फीड फॉरवर्ड एएनएन किससे बना है? m जानकारी $X_1, X_2, \dots, X_m$ और n आउटपुट $Y_1, Y_2, \dots, Y_n$. प्रत्येक इनपुट X_i प्रत्येक आउटपुट से जुड़ा हुआ है Y_j. बीच के पथ से जुड़ा वजन X_i और Y_j इसे इस रूप में निरूपित किया जाता है w_{ij}. कनेक्शन पथ यूनिडायरेक्शनल हैं और इनपुट इकाइयों से आउटपुट परत तक निर्देशित होते हैं।

कुल इनपुट y_in_j आउटपुट इकाई के लिए। Y_j है:

$$y_in_j = x_1 w_{1j} + \cdots + x_i w_{ij} + \cdots + x_m w_{mj} = \sum_{i=1}^{m} x_i w_{ij} \tag{7.13}$$

समीकरण 7.14 को वेक्टर रूप में इस प्रकार लिखा जा सकता है:

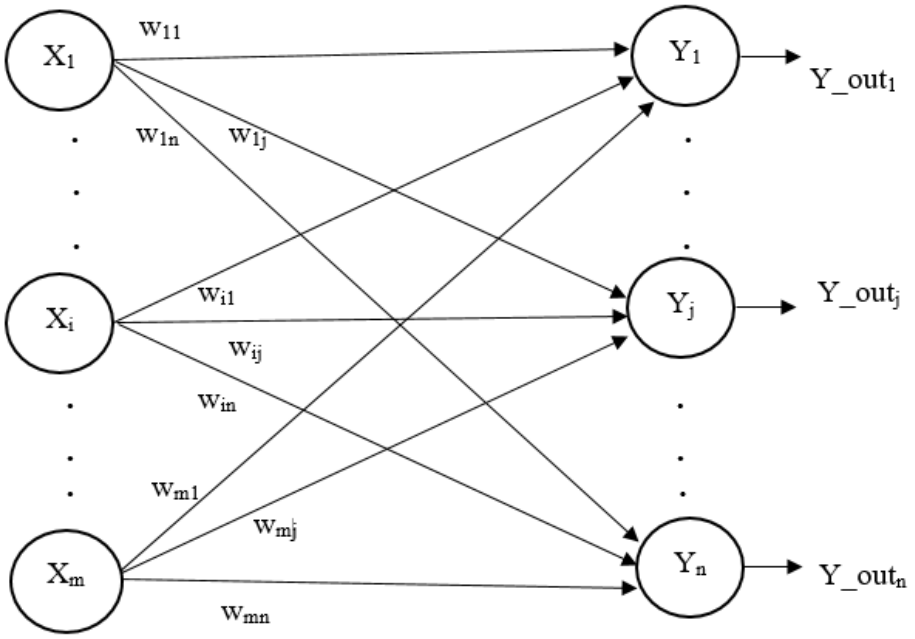
$$y_in_j = [x_1 \ldots x_m] \times \begin{bmatrix} w_{1j} \\ \cdot \\ \cdot \\ \cdot \\ w_{mj} \end{bmatrix} = X \times W_j \tag{7.14}$$

कहां $X = [x_1 \ldots x_m]$ इनपुट वेक्टर है और W_j यह है j^{th} वजन मैट्रिक्स का स्तंभ

$$W = \begin{bmatrix} w_{11} & w_{12} \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} \cdots & w_{mn} \end{bmatrix} \tag{7.15}$$

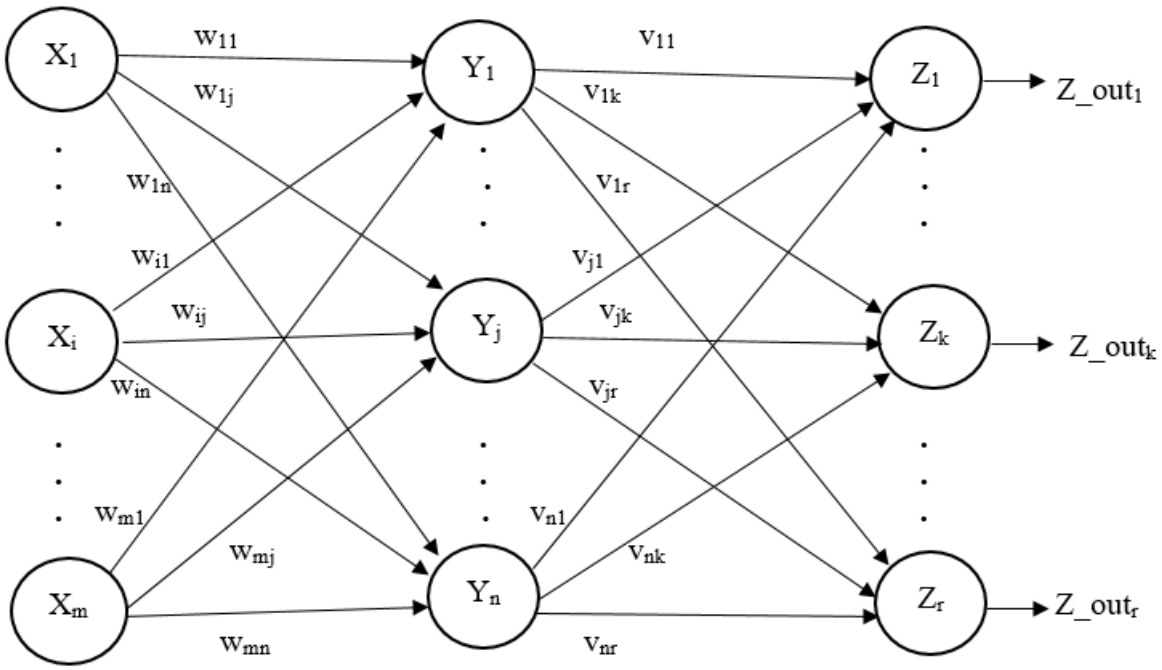
अगर Y_in आउटपुट इकाइयों को शुद्ध इनपुट के लिए वेक्टर को दर्शाता है:

$$Y_in = X \times W \tag{7.16}$$



चित्र 7.6 सिंगल लेयर फीड फॉरवर्ड आर्टिफिशियल न्यूरल नेटवर्क

7.4.2 मल्टीलेयर फीड फॉरवर्ड एएनएन



चित्र 7.7 एक छिपी हुई परत के साथ मल्टीलेयर फीड फॉरवर्ड आर्टिफिशियल न्यूरल नेटवर्क
एक मल्टीलेयर फीड फॉरवर्ड आर्टिफिशियल न्यूरल नेटवर्क सिंगल लेयर फीड फॉरवर्ड नेटवर्क के समान है, सिवाय इसके कि इनपुट इकाइयों और आउटपुट परतों के बीच प्रसंस्करण इकाइयों की अतिरिक्त परतें हैं। इन अतिरिक्त परतों को आर्टिफीसियल तंत्रिका नेटवर्क की छिपी परतों के रूप में जाना जाता है। चित्र 7.7 एक छिपी हुई परत के साथ एक बहुस्तरीय फ़ीड फॉरवर्ड न्यूरल नेटवर्क दिखाता है।

प्रत्येक इनपुट X_i प्रत्येक छिपी हुई परत इकाई से जुड़ा हुआ है Y_j. बीच के पथ से जुड़ा वजन X_i और Y_j इसे इस रूप में निरूपित किया जाता है w_{ij}. इसके अलावा, प्रत्येक छिपी हुई परत इकाई। Y_j प्रत्येक आउटपुट इकाई से जुड़ा हुआ है Z_k. बीच के पथ से जुड़ा वजन Y_j और Z_k इसे इस रूप में निरूपित किया जाता है v_{jk}. कनेक्शन पथ यूनिडायरेक्शनल हैं और इनपुट इकाइयों से छिपी परत इकाइयों से आउटपुट परत तक निर्देशित होते हैं।

कुल इनपुट Y_in छिपी हुई परत इकाइयों के लिए है:

$$Y_in = X \times W \tag{7.17}$$

कुल इनपुट Z_in आउटपुट परत इकाइयों के लिए है:

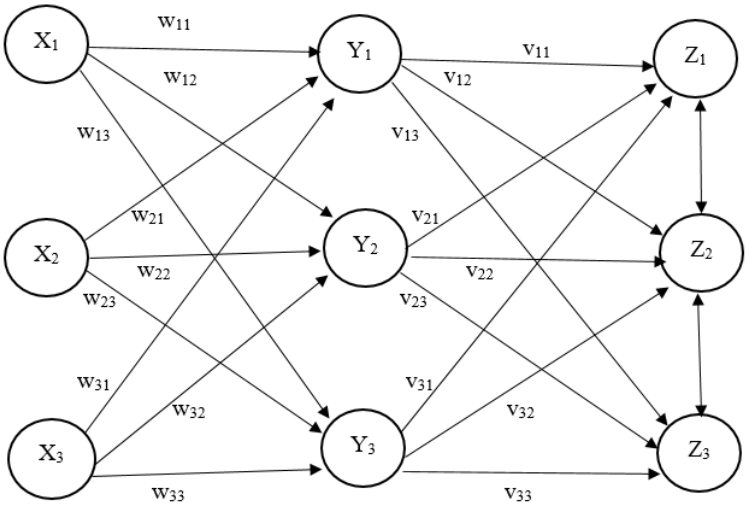
$$Z_in = Y_out \times V \tag{7.18}$$

कहां X इनपुट वेक्टर है $[X_1, X_2, \dots, X_m]$, Y_in छिपी हुई परत के लिए शुद्ध इनपुट है, Y_out छिपी हुई परत से आउटपुट वेक्टर है, Z_in आउटपुट परत के लिए इनपुट वेक्टर है और W और V इनपुट इकाइयों के बीच वजन के लिए वजन मैट्रिक्स हैं - छिपी हुई परत और छिपी हुई परत-आउटपुट परत क्रमशः।

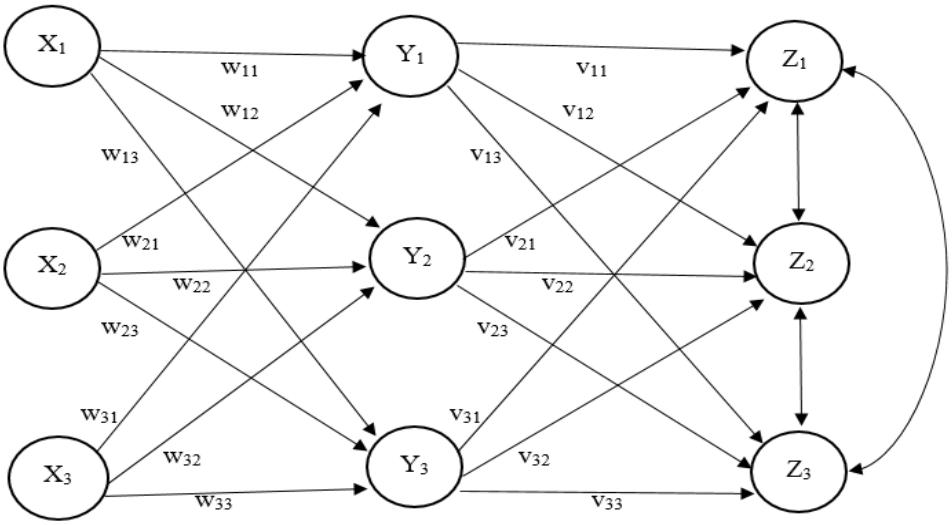
7.4.3 प्रतिस्पर्धी एएनएन

प्रतिस्पर्धी तंत्रिका नेटवर्क संरचना में मल्टीलेयर फीड फॉरवर्ड न्यूरल नेटवर्क के समान हैं, जिस तरह से इसकी आउटपुट परत इकाइयां जुड़ी हुई हैं। प्रतिस्पर्धी तंत्रिका नेटवर्क में आउटपुट परत की इकाइयाँ आमतौर पर आपस में जुड़ी होती हैं।

चित्र 7.8 आसन्न कनेक्टेड आउटपुट इकाइयों के साथ प्रतिस्पर्धी आर्टिफीसियल तंत्रिका नेटवर्क



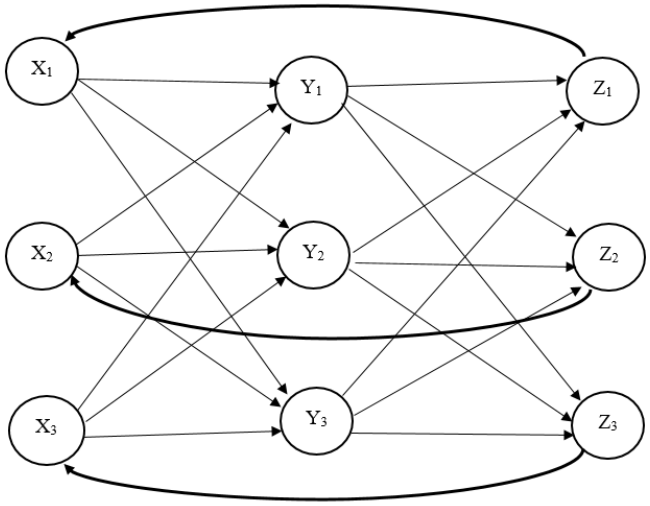
यह इसलिए किया जाता है ताकि आउटपुट इकाइयों के बीच प्रतिस्पर्धा को सक्षम किया जा सके ताकि इनपुट को सबसे सही ढंग से दर्शाया जा सके। चित्र 7.8 पहले प्रकार, आसन्न प्रतिस्पर्धी तंत्रिका नेटवर्क को दर्शाता है जिसमें आउटपुट इकाइयां केवल अपने संबंधित पड़ोसियों से जुड़ी होती हैं और चित्र 7.9 दूसरे प्रकार, पूरी तरह से प्रतिस्पर्धी तंत्रिका नेटवर्क को दर्शाता है जिसमें आउटपुट इकाइयां पूरी तरह से जुड़ी होती हैं।



चित्र 7.9 पूरी तरह से जुड़े आउटपुट इकाइयों के साथ प्रतिस्पर्धी आर्टिफीसियल तंत्रिका नेटवर्क

7.4.4 आवर्तक एएनएन

पिछले उल्लिखित फीड फॉरवर्ड न्यूरल नेटवर्क आर्किटेक्चर में सूचना का प्रवाह छिपी परतों के माध्यम से इनपुट इकाइयों से अंतिम आउटपुट परत तक यूनिडायरेक्शनल है। इस प्रकार के फीड फॉरवर्ड न्यूरल नेटवर्क आर्किटेक्चर में पिछली परतों के लिए कोई मार्ग नहीं होता है। आवर्तक तंत्रिका नेटवर्क आमतौर पर आर्किटेक्चर में एक फीडबैक लूप की अनुमति देते हैं ताकि अगले फॉरवर्ड पास के लिए इनपुट के रूप में कार्य करने के लिए न्यूरल नेटवर्क में आउटपुट के प्रवाह को सक्षम किया जा सके। चित्र 7.10 एक विशिष्ट आवर्तक तंत्रिका नेटवर्क आर्किटेक्चर दिखाता है।



चित्र 7.10 आउटपुट परत से इनपुट इकाइयों तक फीडबैक लूप के साथ आवर्तक आर्टिफीसियल तंत्रिका नेटवर्क

7.5 सक्रियण फ़ंक्शन

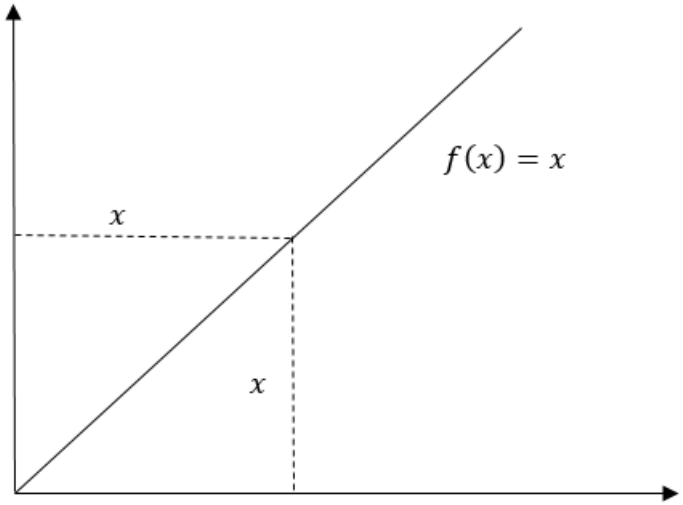
एक आर्टिफीसियल न्यूरॉन से आउटपुट को इसकी सक्रियता कहा जाता है। आर्टिफीसियल न्यूरॉन का सक्रियण आर्टिफीसियल न्यूरॉन के कुल इनपुट का एक कार्य है। यह फ़ंक्शन जो एक आर्टिफीसियल न्यूरॉन के कुल इनपुट को उसके आउटपुट में मैप करता है, आर्टिफीसियल न्यूरॉन के सक्रियण फ़ंक्शन के रूप में जाना जाता है। सबसे अधिक उपयोग किए जाने वाले सक्रियण फ़ंक्शन इस खंड में वर्णित हैं।

7.5.1 **आइडेंटिटी** फ़ंक्शन

आइडेंटिटी सक्रियण फ़ंक्शन होने पर सबसे सरल सक्रियण फ़ंक्शन सबसे अधिक उपयोग किया जाता आइडेंटिटी सक्रियण फ़ंक्शन आने वाले इनपुट मान को इसे बदले बिना आउटगोइंग आउटपुट मान के रूप में पास करता है। इस प्रकार पहचान सक्रियण फ़ंक्शन को इस प्रकार परिभाषित किया गया है:

$$f(x) = x \qquad (7.19)$$

चित्र 7.11 आइडेंटिटी सक्रियण फलन के आरेखीय रूप को दर्शाता है। आमतौर पर, इनपुट परत के न्यूरॉन्स आइडेंटिटी सक्रियण फ़ंक्शन का उपयोग करते हैं क्योंकि आर्टिफीसियल तंत्रिका नेटवर्क में इनपुट न्यूरॉन्स आने वाले इनपुट को अगली परत में पारित करते हैं क्योंकि यह उनके भारित पथों के माध्यम से होता है।



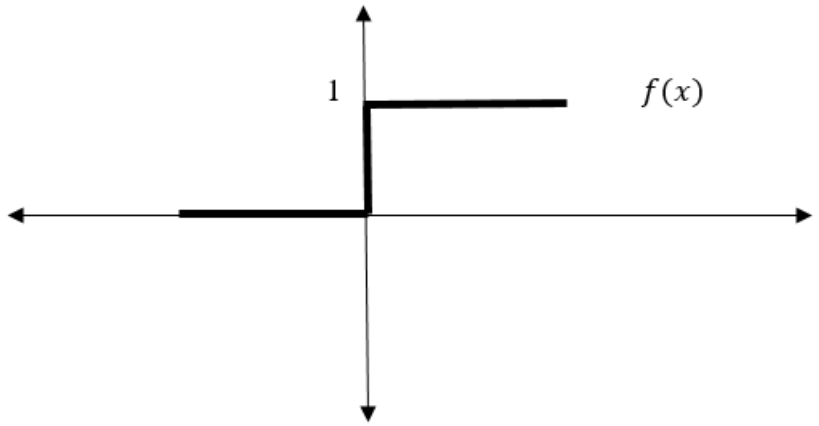
चित्र 7.11 आइडेंटिटी सक्रियण फ़ंक्शन

7.5.2 **स्टेप** फ़ंक्शन

दूसरा आमतौर पर इस्तेमाल किया जाने वाला सक्रियण फ़ंक्शन स्टेप सक्रियण फ़ंक्शन है। स्टेप सक्रियण फ़ंक्शन का मूल रूप 1 या 0 का उत्पादन करता है जो इस बात पर निर्भर करता है कि कुल इनपुट 0 से अधिक है या अन्यथा। गणितीय रूप से मूल चरण सक्रियण फ़ंक्शन को इस प्रकार परिभाषित किया गया है:

$$f(x) = \begin{cases} 1, & if\ x > 0, \\ 0, & otherwise. \end{cases} \tag{7.20}$$

चित्र 7.12 मूल स्टेप सक्रियण फलन का चित्रमय चित्रण प्रदान करता है।

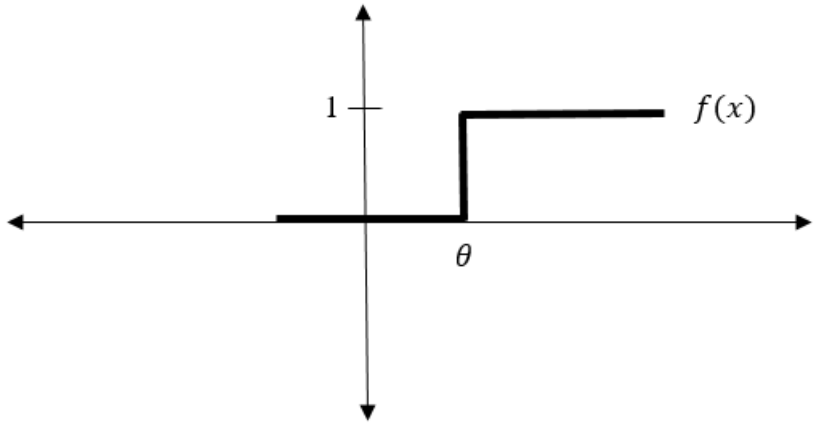


चित्र 7.12 मूल स्टेप सक्रियण फलन

कभी-कभी, 0 के बजाय एक गैर-शून्य थ्रेशोल्ड मान θ इसका भी उपयोग किया जाता है। गैर-शून्य थ्रेशोल्ड मान को नियोजित करने वाला इस प्रकार का स्टेप सक्रियण फ़ंक्शन θ थ्रेशोल्ड सक्रियण फ़ंक्शन के रूप में जाना जाता है और इसे इस प्रकार परिभाषित किया गया है:

$$f(x) = \begin{cases} 1, & if\ x > \theta, \\ 0, & otherwise. \end{cases} \tag{7.21}$$

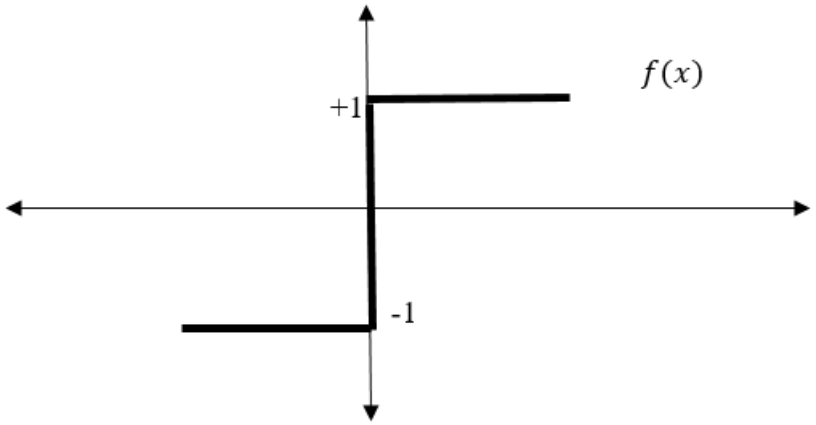
थ्रेशोल्ड सक्रियण फ़ंक्शन को चित्र 7.13 में रेखांकन द्वारा दर्शाया गया है।



चित्र 7.13 थ्रेशोल्ड सक्रियण फ़ंक्शन

उपरोक्त दोनों प्रकार के चरण सक्रियण फ़ंक्शन प्रकृति में द्विआधारी हैं क्योंकि वे हमेशा 0 या 1 का आउटपुट मान देते हैं। एक अन्य श्रेणी चरण सक्रियण फ़ंक्शन द्विध्रुवी चरण सक्रियण फ़ंक्शन हैं जो +1 या -1 का मूल्यांकन करते हैं। इस प्रकार बेसिक स्टेप एक्टिवेशन फ़ंक्शन और थ्रेशोल्ड एक्टिवेशन फ़ंक्शंस के अपने द्विध्रुवीय समकक्ष हैं, अर्थात् क्रमशः बाइपोलर स्टेप एक्टिवेशन फ़ंक्शन और बाइपोलर थ्रेशोल्ड एक्टिवेशन फ़ंक्शन। बाइपोलर स्टेप एक्टिवेशन फ़ंक्शन को चित्र 7.14 में दर्शाया गया है और गणितीय रूप से वर्णित किया गया है:

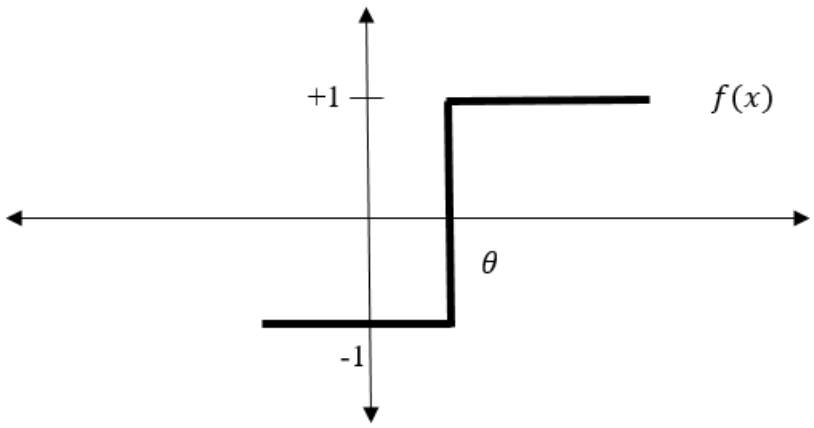
$$f(x) = \begin{cases} +1, & if\ x > 0, \\ -1, & otherwise. \end{cases} \tag{7.22}$$



चित्र 7.14 द्विध्रुवीय चरण सक्रियण फलन

बाइपोलर थ्रेशोल्ड एक्टिवेशन फ़ंक्शन को चित्र 7.15 में दर्शाया गया है और गणितीय रूप से वर्णित किया गया है:

$$f(x) = \begin{cases} +1, & if\ x > \theta, \\ -1, & otherwise. \end{cases} \tag{7.23}$$



चित्र 7.15 बाइपोलर थ्रेशोल्ड एक्टिवेशन फंक्शन

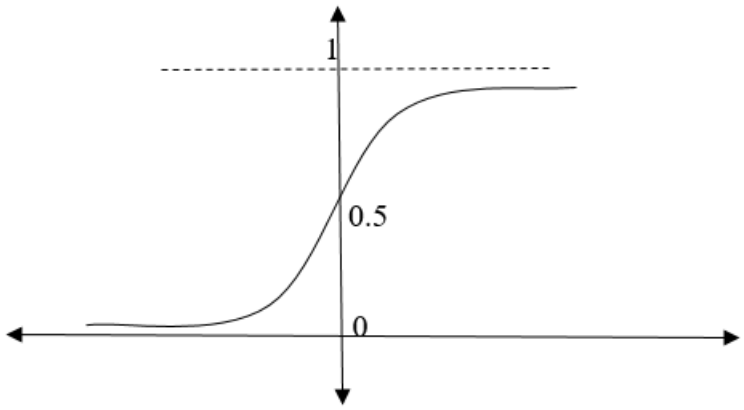
7.5.3 सिग्मोइड फ़ंक्शन

बेसिक स्टेप एक्टिवेशन और थ्रेशोल्ड एक्टिवेशन फ़ंक्शंस में एक महत्वपूर्ण दोष है कि वे अलग-अलग नहीं हैं और इस प्रकार वास्तविक दुनिया के उपयोग के मामलों में डेटा में आत्मसात वास्तविक जटिलता को कैप्चर नहीं कर सकते हैं। इस प्रकार सक्रियण कार्यों के एक नए परिवार की आवश्यकता होती है जिसे डेटा की अंतर्निहित जटिलता और गैर-रैखिकता से निपटने के लिए नियोजित किया जा सकता है

जो आमतौर पर आर्टिफीसियल तंत्रिका नेटवर्क का उपयोग करके जटिल वास्तविक दुनिया की समस्याओं को हल करते समय पाया जाता है। इस उद्देश्य के लिए नियोजित सक्रियण कार्यों का सबसे लोकप्रिय परिवार सिग्मोइड फ़ंक्शन है क्योंकि यह अपनी भिन्नता को छोड़े बिना डेटा में गैर-रैखिक संबंधों का अनुमान लगा सकता है। बाइनरी सिग्मोइड सक्रियण फ़ंक्शन, जिसे लॉजिस्टिक सिग्मोइड सक्रियण फ़ंक्शन के रूप में भी जाना जाता है, सिग्मोइड फ़ंक्शन परिवार में सबसे सरल और सबसे व्यापक रूप से उपयोग किया जाने वाला सक्रियण फ़ंक्शन है और इसे गणितीय रूप से परिभाषित किया गया है:

$$f(x) = \frac{1}{1+e^{-zx}} \tag{7.24}$$

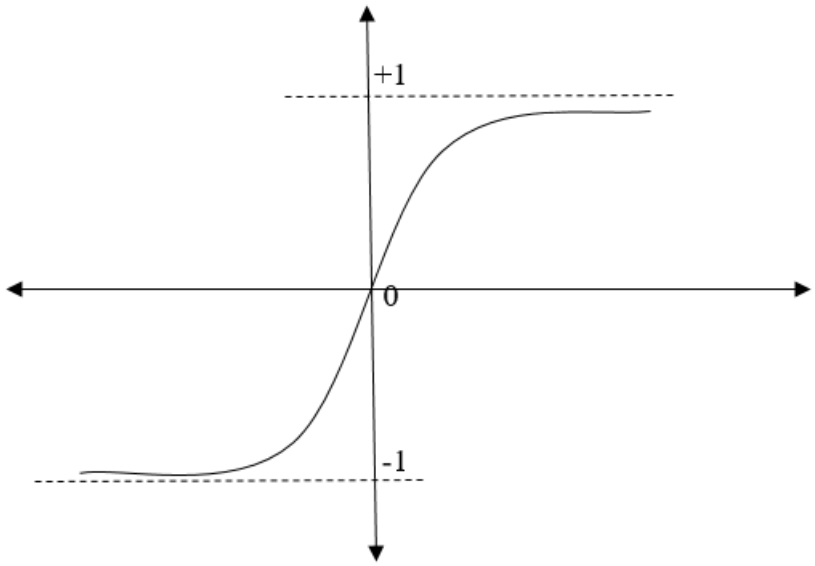
पैरामीटर z समीकरण 7.25 में इसे कठोरता पैरामीटर कहा जाता है। बाइनरी सिग्मोइड सक्रियण फलन को चित्र 7.16 में रेखांकन द्वारा दर्शाया गया है। फ़ंक्शन के मान को बदलकर इसे इच्छानुसार तीव्र बनाया जा सकता है z.



चित्र 7.16 द्विआधारी सिग्मोइड सक्रियण फ़ंक्शन

बाइनरी सिग्मोइड सक्रियण फ़ंक्शन को मानों की किसी भी श्रेणी में बढ़ाया जा सकता है ताकि अंतर्निहित डेटा का सटीक प्रतिनिधित्व किया जा सके। सबसे व्यापक रूप से नियोजित सीमा -1 से +1 तक है और संबंधित सिग्मोइड सक्रियण फ़ंक्शन को बाइपोलर सिग्मोइड सक्रियण फ़ंक्शन के रूप में जाना जाता है। बाइपोलर सिग्मोइड सक्रियण फलन का गणितीय निरूपण नीचे दिया गया है और इसका आरेखीय निरूपण चित्र 7.17 में दर्शाया गया है।

$$f(x) = \frac{1-e^{-zx}}{1+e^{-zx}} \tag{7.25}$$



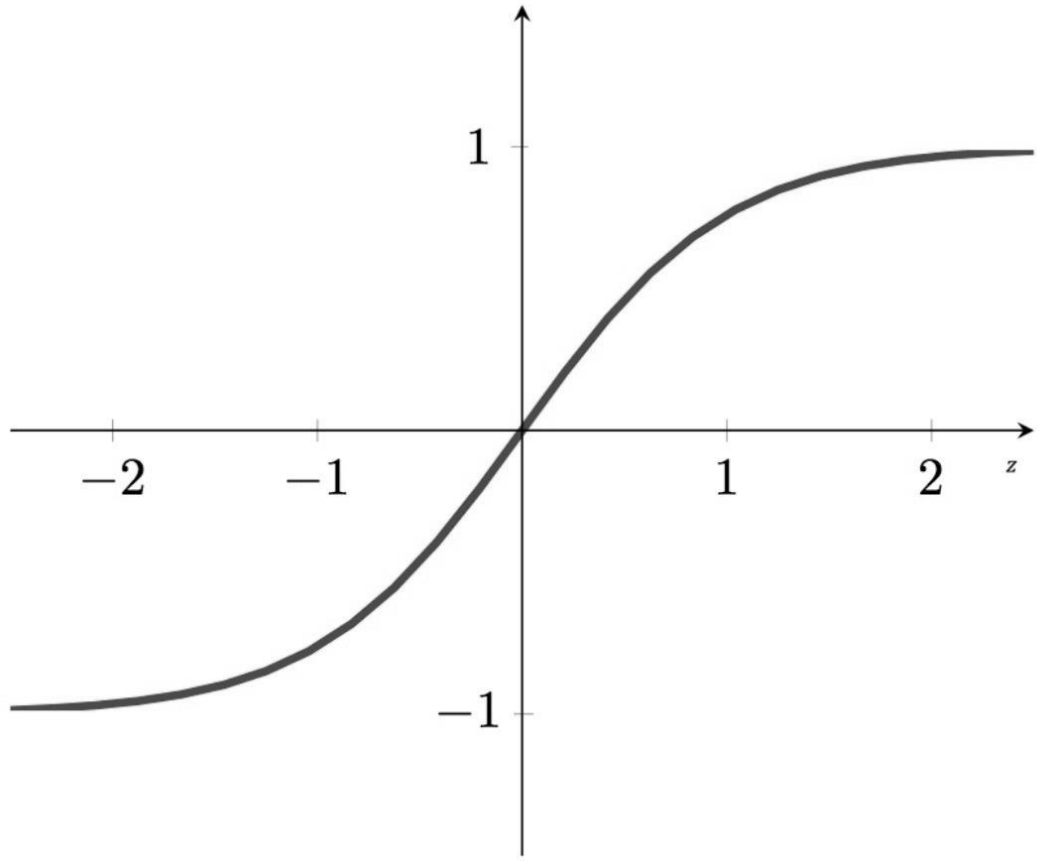
चित्र 7.17 बाइपोलर सिग्मोइड सक्रियण फलन।

7.5.4 हाइपरबॉलिक स्पर्शरेखा फ़ंक्शन

सक्रियण कार्यों के सिग्मोइड परिवार का एक महत्वपूर्ण नुकसान यह है कि वे z बहुत बड़ा या बहुत छोटा है और वे भी बहुत संवेदनशील हैं जब मूल्य z शून्य के पड़ोस में है। इस कमी को दूर करने के लिए और फिर भी भिन्नता की शक्ति का उपयोग करने के लिए द्विध्रुवी सिग्मोइड सक्रियण फ़ंक्शन से निकटता से संबंधित एक और समान सक्रियण फ़ंक्शन का उपयोग किया जाता है जिसे हाइपरबॉलिक स्पर्शरेखा सक्रियण फ़ंक्शन कहा जाता है। यह ज्यादातर तब नियोजित होता है जब अंतर्निहित डेटा प्रकृति में द्विआधारी होता है और लगातार 0 से 1 तक की सीमा में मूल्यवान नहीं होता है। इस मामले में, डेटा को पहले आमतौर पर अपने द्विध्रुवी रूप में परिवर्तित किया जाता है और फिर हाइपरबॉलिक स्पर्शरेखा सक्रियण फ़ंक्शन पर पारित किया जाता है। हाइपरबॉलिक स्पर्शरेखा सक्रियण फ़ंक्शन का गणितीय रूप है:

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{7.26}$$

हाइपरबॉलिक स्पर्शरेखा सक्रियण फलन का चित्रमय चित्रण चित्र 7.18 में दर्शाया गया है।



चित्र 7.18 हाइपरबॉलिक स्पर्शरेखा सक्रियण फलन

7.6 आर्टिफीसियल न्यूरल नेटवर्क लर्निंग

आर्टिफिशियल न्यूरल नेटवर्क लर्निंग वजन और मापदंडों के उपयुक्त सेट को खोजने की प्रक्रिया है जो नुकसान को कम करते हुए इनपुट मूल्यों और आउटपुट के बीच अंतर्निहित संबंध को सबसे सटीक रूप से अनुमानित करता है ताकि आर्टिफीसियल तंत्रिका नेटवर्क उस कार्य को करने के लिए सबसे इष्टतम तरीके से कार्य करने की क्षमता प्राप्त कर सके जिसके लिए इसे डिज़ाइन किया गया है। इस प्रक्रिया को आर्टिफीसियल तंत्रिका नेटवर्क के प्रशिक्षण के रूप में भी जाना जाता है।

आर्टिफिशियल न्यूरल नेटवर्क लर्निंग का मूल सिद्धांत बहुत सरल है। यह यादृच्छिक रूप से शुरू किए गए वजन के प्रारंभिक वितरण से शुरू होता है। यह तब तक वजन को समायोजित करता है जब तक कि कुछ रोक मानदंड तक नहीं पहुंच जाता है या मॉडल अभिसरण नहीं होता है। इस प्रकार, यदि w_t एक निश्चित पथ का भार क्या है? i^{th} पुनरावृत्ति फिर वजन पर $(i+1)^{th}$ पुनरावृत्ति, अर्थात। w_{t+1} सामान्य नियम द्वारा प्राप्त किया जाता है:

$$w_{i+1} = w_i + \Delta w_i \tag{7.27}$$

कहां Δw_i यह है i^{th} वजन में समायोजन w. किसी भी आर्टिफीसियल तंत्रिका नेटवर्क सीखने के एल्गोरिथ्म का मुख्य उद्देश्य गणना करने के लिए की जाने वाली विधि को परिभाषित करना है। Δw_i.

आर्टिफीसियल तंत्रिका नेटवर्क सीखने के मोटे तौर पर दो वर्ग हैं, पर्यवेक्षित और असुरक्षित शिक्षा। यह खंड नीचे विस्तार से इन दोनों सीखने की रणनीतियों का वर्णन करता है।

7.6.1 पर्यवेक्षित शिक्षा

एक आर्टिफीसियल तंत्रिका नेटवर्क प्रशिक्षण वैक्टर के एक सेट की मदद से प्रशिक्षित किया जाता है। इन वैक्टर के आउटपुट पहले से ज्ञात हो सकते हैं या नहीं भी हो सकते हैं। जब आर्टिफीसियल तंत्रिका नेटवर्क को इनपुट-आउटपुट प्रशिक्षण जोड़े का उपयोग करके प्रशिक्षित किया जाता है तो प्रशिक्षण को पर्यवेक्षित प्रशिक्षण कहा जाता है। पर्यवेक्षित प्रशिक्षण को नियोजित करके किए गए विशिष्ट कार्य वर्गीकरण और प्रतिगमन हैं। कुछ सबसे आम पर्यवेक्षित प्रशिक्षण विधियों को संक्षेप में नीचे पेश किया गया है:

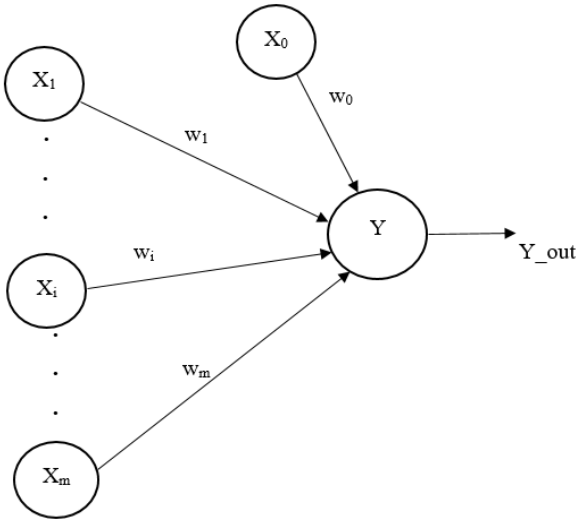
(A) हेब सीखने का नियम

हेब लर्निंग नियम आर्टिफीसियल तंत्रिका नेटवर्क के प्रशिक्षण के लिए सबसे प्राथमिक नियमों में से एक है। इस नियम के अनुसार वजन समायोजन इस प्रकार पूरा किया जाता है:

$$\Delta w_i = x_i \times t \tag{7.28}$$

कहां Δw_i यह है i^{th} वजन में समायोजन w, x_i यह है i^{th} इनपुट वेक्टर की पुनरावृत्ति और t लक्ष्य सक्रियण मान है.

एक एकल परत फ़ीड फॉरवर्ड नेटवर्क जिसे हेब लर्निंग नियम का उपयोग करके प्रशिक्षित किया जाता है, को हेब नेट के रूप में जाना जाता है और हेब नेट की वास्तुकला को चित्र 7.19 में दिखाया गया है।



चित्र 7.19 हेब नेट आर्किटेक्चर

उपरोक्त चित्र में इनपुट इकाई X_0 और इसका वजन w_0 पूर्वाग्रह की भूमिका निभाते हैं। का मूल्य X_0 इसे हमेशा 1 के रूप में रखा जाता है। इस प्रकार, वजन के अपडेशन के लिए समीकरण 7.29 w_0 **बन जाता है**:

$$w_{0\,new} = w_{0\,old} + 1 \times Y_out = w_{0\,old} + Y_out \quad (7.29)$$

हेब लर्निंग नियम को लागू करने के लिए विस्तृत एल्गोरिथ्म नीचे एल्गोरिथ्म 7.1 में प्रस्तुत किया गया है।

एल्गोरिथ्म **7.1:** हेब लर्निंग

चरण **1: सभी वजनों को** 0 **के मान पर प्रारंभ करें.**

$$w_i = 0, \quad for\ all\ i = 0\ to\ m.$$

चरण **2: प्रत्येक प्रशिक्षण इनपुट वेक्टर के लिए - लक्ष्य आउटपुट जोड़ी,** $I{:}T$, **चरण** 3 - 5 **का पालन करें।**

चरण **3: इनपुट वेक्टर को इनपुट परत में असाइन करें**:

$$X_0 = 1\ and\ X_i = I_i\ for\ all\ i = 1\ to\ m.$$

चरण **4: लक्ष्य आउटपुट के साथ आउटपुट की गणना करें** $Y_out = T$.

चरण **5: वजन को इस प्रकार समायोजित करें**:

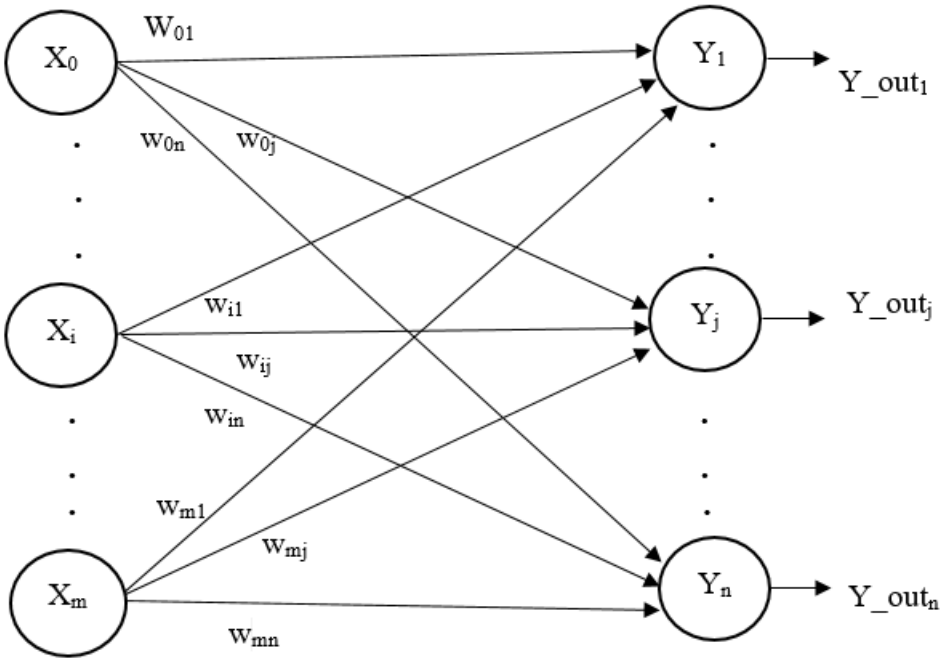
$$w_{i\,new} = w_{i\,old} + X_i \times Y_out$$

चरण **6: रुको।**

(बी) परसेप्ट्रॉन लर्निंग रूल

एक साधारण परसेप्ट्रॉन पर विचार करें। Y **चित्र** 7.20 **में दर्शाया गया है** $m + 1$ **इनपुट इकाइयाँ** $X_0, X_1, \ldots, X_m$ **और संबंधित वजन।** $w_0, w_1, \ldots, w_m$. **पूर्वाग्रह को वजन के रूप में चित्रित किया गया है।** w_0 **इनपुट से कनेक्ट डे** X_0 **जिसका मान** 1 **पर तय होता है। आउटपुट न्यूरॉन्स के लिए, द्विध्रुवी सक्रियण फ़ंक्शन का उपयोग किया जाता है**:

$$Y_out = \begin{cases} 1, & if\ Y_in > 0 \\ 0, & if\ Y_in = 0 \\ -1, & if\ Y_in < 1 \end{cases}$$



चित्र 7.20 **एकल परसेप्ट्रॉन वाई के साथ एएनएन की संरचना।**

छोड़ देना $X = [X_0, X_1, \dots, X_m]$ **प्रशिक्षण वेक्टर हो जिसके लिए परसेप्ट्रॉन का उत्पादन होने की उम्मीद है** t **कहां** $t = 1, 0\ or - 1.$ **वजन वेक्टर द्वारा वजन को दर्शाया गया है।** $W = [w_0, w_1, \dots, w_m].$ **यदि परसेप्ट्रॉन वांछित आउटपुट उत्पन्न करता है तो वजन को अपडेट करने की आवश्यकता नहीं है। लेकिन अगर परसेप्ट्रॉन गलत वर्गीकरण करता है** X **नकारात्मक रूप से यानी यह सही आउटपुट +1 के बजाय -1 का उत्पादन करता है तो वजन को उपयुक्त रूप से बढ़ाया जाना चाहिए। इसके विपरीत यदि भार परसेप्ट्रॉन गलत वर्गीकृत होता है।** X **सकारात्मक रूप से यानी यह -1 के बजाय +1 का उत्पादन करता है तो वजन को उपयुक्त रूप से कम किया जाना चाहिए। इस प्रकार परसेप्ट्रॉन प्रशिक्षण नियम बताता है कि:** ***यदि आउटपुट गलत वर्गीकृत है तो वजन को अपडेट करें अन्यथा वजन अपडेट नहीं किया जाता है।***

इस प्रकार परसेप्ट्रॉन लर्निंग नियम को गणितीय रूप से वर्णित किया जा सकता है:

$$\Delta w_i = \eta \times (t - Y_out) \times X_i \ \ for\ i = 0,1, \dots, m \qquad (7.30)$$

यहाँ η **यह एक स्थिरांक है जिसे आमतौर पर सीखने की दर के रूप में जाना जाता है। मैट्रिक्स नोटेशन का उपयोग करके, परसेप्ट्रॉन लर्निंग नियम को इस प्रकार लिखा जा सकता है**:

$$\Delta W = \eta \times (t - Y_out) \times X \qquad (7.31)$$

कहां ΔW और X भार और इनपुट का प्रतिनिधित्व करने वाले वैक्टर हैं।

परसेप्ट्रॉन लर्निंग नियम को लागू करने के लिए विस्तृत एल्गोरिथ्म नीचे एल्गोरिदम 7.2 में प्रस्तुत किया गया है।

एल्गोरिथ्म **7.2:** परसेप्ट्रॉन लर्निंग

चरण **1:** **सभी वजन ों को प्रारंभ करें $W_0, W_1, \ldots, W_m$ अनियमित।**

चरण **2:** **सीखने की दर सेट करें η जैसे कि $0 < \eta \leq 1$ और दहलीज। θ.**

चरण **3:** **प्रत्येक प्रशिक्षण जोड़ी के लिए $I:T$ चरण 4 - 8 निष्पादित करें।**

चरण **4:** **इनपुट वेक्टर को इनपुट परत में असाइन करें:**

$$X_i = I_i \; for \; all \; i = 0 \; to \; m.$$

चरण **5:** **आउटपुट परत के लिए कुल इनपुट की गणना करें:**

$$Y_in = \sum_{i=1}^{m} w_i X_i$$

चरण **6:** **का उपयोग करके आउटपुट परत इकाइयों के सक्रियण की गणना करें द्विध्रुवी सक्रियण समारोह के रूप में:**

$$Y_out = \begin{cases} 1, & if \; Y_in > \theta \\ 0, & if -\theta < Y_in \leq \theta \\ -1, & if \; Y_in < -\theta \end{cases}$$

चरण **7:** **यदि कोई त्रुटि है, अर्थात** Y_out ≠ **टी, फिर वह वजन अपडेट करता है। निम्नानुसार:**

$$w_{i\,new} = w_{i\,old} + \eta \times t \times X_i$$

यदि कोई त्रुटि नहीं है तो वजन अपडेट नहीं किया जाता है।

चरण **8:** **यदि कोई त्रुटि नहीं थी, अर्थात** Y_out = T, **के पूरे सेट के लिए इनपुट-आउटपुट जोड़े प्रशिक्षण फिर बंद करो। अन्यथा, चरण 3 पर जाएँ।**

(सी) विड्रो-हॉफ लर्निंग नियम

विड्रो-हॉफ लर्निंग नियम जिसे कम से कम मीन स्क्वायर (एलएमएस) या डेल्टा नियम के रूप में भी जाना जाता है, आर्टिफीसियल तंत्रिका नेटवर्क सीखने के लिए सबसे महत्वपूर्ण सीखने के नियम में से एक है। विड्रो-हॉफ लर्निंग नियम का उद्देश्य लक्ष्य मूल्य और आर्टिफीसियल तंत्रिका नेटवर्क द्वारा अनुमानित मूल्य के बीच औसत वर्ग त्रुटि को कम

करना है। इस नियम में, भार को निम्नलिखित गणितीय सूत्र के अनुसार अद्यतन किया जाता है:

$$\Delta w_i = \eta \times (t - Y_in) \times X_i \quad (7.32)$$

विड्रो-हॉफ लर्निंग नियम को लागू करने के लिए विस्तृत एल्गोरिदम नीचे एल्गोरिदम 7.3 **में प्रस्तुत किया गया है।**

एल्गोरिथ्म **7.3:** विड्रो-हॉफ लर्निंग

चरण **1: सभी वजन ों को प्रारंभ करें** $W_0, W_1, \ldots, W_m$ **अनियमित।**

चरण **2: सीखने की दर सेट करें** η **जैसे कि** $0 < \eta \leq 1$ **और दहलीज।** θ.

चरण **3: प्रत्येक प्रशिक्षण जोड़ी के लिए** $I{:}T$ **चरण** 4 - 5 **का पालन करें जब तक कि रोकने का मानदंड पूरा नहीं हुआ है।**

चरण **4: आउटपुट परत के लिए कुल इनपुट की गणना इस प्रकार करें**:

$$Y_in = \sum_{i=1}^{m} w_i X_i$$

चरण **5: सूत्र का उपयोग करके वजन समायोजित करें**:

$$w_{i\,new} = w_{i\,old} + \eta \times (t - Y_in) \times X_i \; for \; i = 0, \ldots, m$$

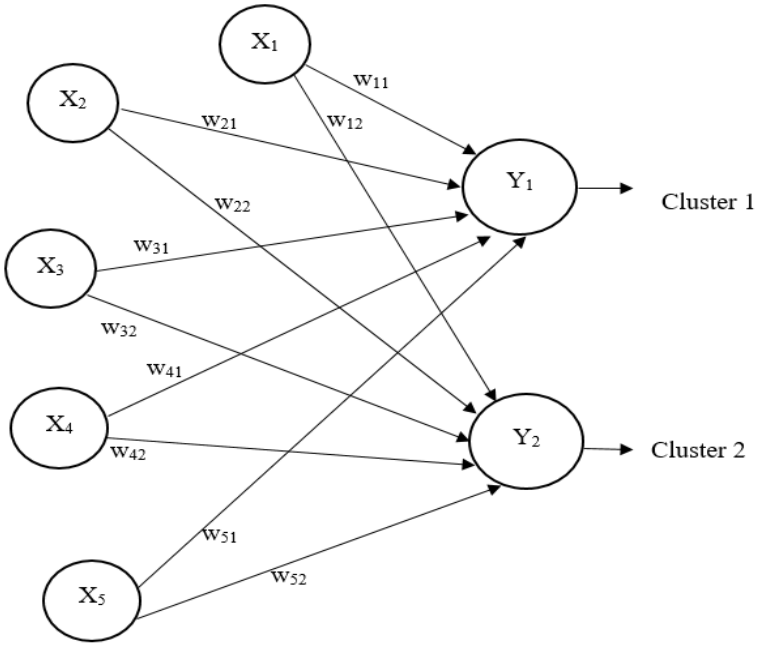
7.6.2 असुरक्षित शिक्षा

जब आर्टिफीसियल तंत्रिका नेटवर्क को केवल इनपुट प्रशिक्षण वैक्टर का उपयोग करके प्रशिक्षित किया जाता है और लक्ष्य आउटपुट वैक्टर उपलब्ध नहीं होते हैं तो प्रशिक्षण को असुरक्षित प्रशिक्षण कहा जाता है। असुरक्षित प्रशिक्षण को नियोजित करके पूरा किए गए विशिष्ट कार्य क्लस्टरिंग है। क्लस्टरिंग दिए गए इनपुट को समूहों में इस तरह से विभाजित करने की प्रक्रिया है कि समान इनपुट एक ही क्लस्टर से संबंधित हैं और असमान इनपुट अलग-अलग क्लस्टर से संबंधित हैं। यह असुरक्षित प्रशिक्षण का मामला है क्योंकि इसमें कोई लक्ष्य आउटपुट नहीं है। आर्टिफीसियल तंत्रिका नेटवर्क को अंतर्निहित डेटा में

अंतर्निहित पैटर्न को स्वयं समझना होगा और फिर शुरुआत में केवल समूहों की संख्या को देखते हुए इनपुट को उपयुक्त समूहों को असाइन करना होगा। इस प्रकार यह एक ऐसी स्थिति की ओर जाता है जहां किसी दिए गए डेटा इनपुट के लिए आउटपुट न्यूरॉन्स में से बिल्कुल एक में आग लग जाती है जो अंतिम क्लस्टर का संकेत देती है जिससे इनपुट डेटा संबंधित है। ऐसा करने के लिए आर्टिफीसियल तंत्रिका नेटवर्क को फिर से डिज़ाइन किया जाना चाहिए ताकि नेटवर्क यह तय करने के लिए बाध्य हो कि दिए गए इनपुट डेटा के लिए कौन सा आउटपुट न्यूरॉन फायर करना चाहिए। यह एक रणनीति के माध्यम से प्राप्त किया जाता है जिसे प्रतियोगिता के रूप में जाना जाता है। सबसे व्यापक रूप से इस्तेमाल की जाने वाली प्रतिस्पर्धी पद्धति विजेता टेक ऑल लर्निंग है।

चित्र 7.21 में दर्शाए अनुसार पांच इनपुट इकाइयों और दो आउटपुट न्यूरॉन्स से बने एक साधारण आर्टिफीसियल तंत्रिका नेटवर्क पर विचार करें। चूंकि प्रत्येक इनपुट इकाई इनपुट वेक्टर के एक घटक को स्वीकार करती है, इस प्रकार आर्टिफीसियल तंत्रिका नेटवर्क फॉर्म के इनपुट वैक्टर को स्वीकार करता है। $X = [X_1, X_2, X_3, X_4, X_5]$. नेटवर्क का प्रत्येक आउटपुट न्यूरॉन एक क्लस्टर का प्रतिनिधित्व करता है। इस प्रकार वर्तमान नेटवर्क इनपुट वैक्टर को दो समूहों में से एक को असाइन करेगा।

विजेता टेक ऑल लर्निंग रणनीति में आर्टिफीसियल तंत्रिका नेटवर्क आउटपुट न्यूरॉन पाता है जो दिए गए इनपुट वेक्टर के लिए सबसे अच्छा फिट बैठता है और इसे विजेता बनाता है। विजेता न्यूरॉन के लिए



चित्र 7.21 ए पांच इनपुट दो आउटपुट क्लस्टरिंग आर्टिफीसियल तंत्रिका नेटवर्क।

वजन वेक्टर तब एल्गोरिदम के अनुसार अपडेट किया जाता है। सबसे अच्छा फिट और इस प्रकार विजेता आउटपुट न्यूरॉन खोजने का सबसे आम तरीका इनपुट वेक्टर और वजन वेक्टर के बीच यूक्लिडियन दूरी माप के वर्ग का उपयोग करना है। आउटपुट न्यूरॉन जिसमें इसके वजन वेक्टर और इनपुट वेक्टर के बीच सबसे छोटा यूक्लिडियन दूरी माप होता है, विजेता के रूप में चुना जाता है। विजेता सभी एल्गोरिथ्म को नीचे एल्गोरिथ्म 7.4 में वर्णित किया गया है:

एल्गोरिथ्म **7.4:** विजेता सभी सीख लेता है

चरण 1: प्रत्येक आउटपुट न्यूरॉन के लिए $Y_j, j = 1\ to\ n$, वर्ग की गणना करें यूक्लिडियन दूरी के रूप में:

$$ED(j) = \sum_{i=1}^{m} \left(w_{ij} - x_i\right)^2$$

चरण 2: छोड़ देना Y_j सबसे छोटे वर्ग के साथ आउटपुट न्यूरॉन बनें यूक्लिडियन दूरी $ED(j)$.

चरण 3: पड़ोस के भीतर सभी आउटपुट न्यूरॉन्स के लिए Y_j आधुनिकीकरणअ वजन निम्नानुसार है:

$$w_{ij\ new} = w_{ij\ old} + \eta \times \left[x_i - w_{ij\ old}\right]$$

7.7 सारांश

आर्टिफीसियल तंत्रिका नेटवर्क मानव मस्तिष्क के जैविक कार्य की अवधारणा से प्रेरित हैं और यह वॉन न्यूमैन डिजिटल कंप्यूटर द्वारा निष्पादन की पारंपरिक विधि के बजाय पैटर्न के रूप में डेटा को कैसे संसाधित करता है। अलग-अलग मेमोरी में जानकारी संग्रहीत करने और इसे एक समर्पित प्रोसेसर में संसाधित करने के पारंपरिक दृष्टिकोण के बजाय, आर्टिफीसियल तंत्रिका नेटवर्क जानकारी को विभिन्न न्यूरॉन्स को जोड़ने वाले मार्गों से जुड़े वजन के रूप में संग्रहीत करता है, इस प्रकार वितरित शैली में जानकारी संग्रहीत करता है। आर्टिफीसियल तंत्रिका नेटवर्क वर्गीकरण, प्रतिगमन और क्लस्टरिंग के कार्यों से संबंधित समस्याओं को हल करने के लिए नियोजित किया जा सकता है।

पहला तंत्रिका मॉडल 1943 में वॉरेन मैककुलोच और वाल्टर पिट्स द्वारा प्रस्तावित किया गया था जो एक सरल थ्रेशोल्ड तर्क था। यह प्रसंस्करण इकाई को दिए गए कुल इनपुट को एकत्र करता है और बाइनरी आउटपुट का उत्पादन करने के लिए एक बाइनरी थ्रेशोल्ड ऑपरेशन करता है। यह अपने डिजाइन द्वारा बुलियन डेटा पर केवल रैखिक बुलियन वर्गीकरण कार्यों को करने के लिए गंभीर रूप से प्रतिबंधित था और कुछ संभावित इंटरसेप्ट के साथ केवल निश्चित ढलान प्रदान करता था, इस प्रकार जटिल वर्गीकरण कार्यों को करने के लिए इसकी प्रयोज्यता को सख्ती से सीमित करता था। परसेप्ट्रॉन मॉडल का सुझाव 1962 में रोसेनब्लाट द्वारा दिया गया था। यह संरचना में मैककुलोच-पिट्स न्यूरल मॉडल के समान है, सिवाय इसके कि प्रत्येक इनपुट मान में अब इसके लिए एक संबद्ध वजन होता है। परसेप्ट्रॉन में दो पैरामीटर होते हैं जिसका अर्थ है कि इसमें जटिल गैर-रैखिक कार्यों को करने की सीमित क्षमता भी है। यह परसेप्ट्रॉन मॉडल को सरल रैखिक कार्यों के लिए भी उपयुक्त बनाता है जहां अंतर्निहित डेटा रैखिक रूप से वियोज्य है। जिन कार्यों को यह कर सकता है वे बूलियन वर्गीकरण की श्रेणी में आते हैं जिसमें इनपुट डेटा को आउटपुट वर्गों में से दो से संबंधित वर्गीकृत किया जाता है। यह बूलियन मूल्यों के साथ-साथ वास्तविक मूल्यों को अपने प्रसंस्करण तत्व के इनपुट के रूप में स्वीकार कर सकता है। परसेप्ट्रॉन इसे कार्य करने के लिए एक हेरिस्टिक इंटेलिजेंसमान सीखने के एल्गोरिदम को नियोजित करता है जिसे परसेप्ट्रॉन लर्निंग एल्गोरिदम के रूप में जाना जाता है। आधुनिक आर्टिफीसियल तंत्रिका नेटवर्क का निर्माण सिंगल लेयर फीड फॉरवर्ड, मल्टी-लेयर फीड फॉरवर्ड, प्रतिस्पर्धी और आवर्तक तंत्रिका नेटवर्क जैसे विभिन्न आर्किटेक्चर का उपयोग करके किया जा सकता है जो जटिल गैर-रैखिक कार्यों को करने के लिए आर्टिफीसियल तंत्रिका नेटवर्क की पूर्ण क्षमताओं का दोहन करने की अनुमति देता है। यह उपयुक्त सक्रियण फ़ंक्शंस के उपयोग से संभव है। कई लोकप्रिय मानक सक्रियण कार्यों का व्यापक रूप से उपयोग किया जाता है जैसे कि पहचान फ़ंक्शन, स्टेप फ़ंक्शन, सिग्मोइड फ़ंक्शन और हाइपरबॉलिक स्पर्शरेखा फ़ंक्शन। जटिल कार्यों को करने के लिए, आर्टिफीसियल तंत्रिका नेटवर्क सीखने के तरीकों को नियोजित करते हैं। सीखने के दो प्रकार के तरीके हैं, अर्थात् पर्यवेक्षित और असुरक्षित। पर्यवेक्षित शिक्षण इनपुट-आउटपुट प्रशिक्षण जोड़े के सेट प्रदान करके किया जाता है। विशिष्ट पर्यवेक्षित शिक्षण कार्य वर्गीकरण और प्रतिगमन हैं। असुरक्षित शिक्षण केवल इनपुट प्रशिक्षण डेटा प्रदान करके पूरा किया जाता है और लक्ष्य आउटपुट डेटा उपलब्ध नहीं है। विशिष्ट असुरक्षित कार्य क्लस्टरिंग है। कई पर्यवेक्षित सीखने की रणनीतियाँ हेब नियम, परसेप्ट्रॉन नियम और विड्रो-हॉफ नियम हैं। एक सामान्य असुरक्षित सीखने की

रणनीति विजेता सभी रणनीति लेती है। प्रत्येक सीखने की रणनीति का उद्देश्य गणना करने के लिए की जाने वाली विधि को परिभाषित करके वजन को अपडेट करना है। Δw_i.

प्रमुख शब्द

- आर्टिफीसियल तंत्रिका नेटवर्क
- जैविक न्यूरॉन
- आर्टिफीसियल न्यूरॉन
- अक्षतंतु
- राशि
- अंतरग्रथनअ
- सक्रियण
- मैककुलोच-पिट्स न्यूरॉन
- परसेप्ट्रॉन
- वजन
- वर्गीकरण
- प्रतिगमन
- क्लस्टरिंग
- रैखिक रूप से वियोज्य।
- शुरूआत
- सिंगल लेयर फीड फॉरवर्ड एएनएन
- मल्टीलेयर फीड फॉरवर्ड एएनएन
- प्रतिस्पर्धी ANN
- आवर्तक ANN
- छिपी हुई परतें
- सक्रियण फ़ंक्शन
- पहचान समारोह
- मूल चरण फ़ंक्शन
- थ्रेशोल्ड फ़ंक्शन
- बाइपोलर स्टेप फंक्शन
- बाइपोलर थ्रेशोल्ड फंक्शन
- सिग्मोइड फ़ंक्शन
- बाइपोलर सिग्मोइड फंक्शन
- हाइपरबॉलिक स्पर्शरेखा फ़ंक्शन
- पर्यवेक्षित शिक्षा
- असुरक्षित शिक्षा।
- हेब सीखने का नियम
- परसेप्ट्रॉन लर्निंग रूल
- विड्रो-हॉफ सीखने का नियम
- विजेता सभी सीखने के नियम लेता है

अभ्यास

1. बहुत छोटे प्रकार के प्रश्न (**2** अंक):

a) सक्रियण फ़ंक्शन क्या है? न्यूरॉन मॉडल में इसके उपयोग की व्याख्या कीजिए?
b) एक एकल परत फ़ीड फॉरवर्ड नेटवर्क बनाएं और इसके कामकाजी कार्यों की व्याख्या करें।
c) आवर्तक नेटवर्क कैसे काम करता है? बहुस्तरीय तंत्रिका नेटवर्क के साथ तुलना करें।
d) सीखने की विभिन्न तकनीकों पर चर्चा कीजिए।
e) मल्टीलेयर परसेप्ट्रॉन मॉडल क्या है? इसे समझाइए .
f) एकल परत और बहुस्तरीय परसेप्ट्रॉन मॉडल की तुलना करें।
g) आर्टिफीसियल तंत्रिका नेटवर्क का उपयोग क्यों करें? क्या हैं इसके फायदे
h) आर्टिफिशियल न्यूरल नेटवर्क सामान्य कंप्यूटर से कैसे अलग हैं?
i) एक साधारण आर्टिफीसियल न्यूरॉन क्या है?
j) आर्टिफीसियल तंत्रिका नेटवर्क जैविक नेटवर्क से कैसे संबंधित हैं?
k) विभिन्न सक्रियण फ़ंक्शंस सूचीबद्ध करें। .
l) मल्टीलेयर फीड फॉरवर्ड नेटवर्क में छिपी परतों की भूमिका सूचीबद्ध करें?
m) न्यूरो-कंप्यूटिंग और पारंपरिक कंप्यूटर कंप्यूटिंग के बीच अंतर को स्पष्ट करें।
n) पर्यवेक्षित और गैर-पर्यवेक्षित सीखने में अंतर करें।
o) सीखने की दर पैरामीटर क्या है?
p) सिग्मोइड कार्यों की विशेषताएं क्या हैं?
q) मैट्रिक्स गुणन विधि का उपयोग करके शुद्ध इनपुट की गणना कैसे की जाती है?
r) एक जैविक न्यूरॉन की संरचना खींचें?

2. लघु प्रकार के प्रश्न (**5** अंक):

a) सिग्मोइड फ़ंक्शन के महत्व का उदाहरण दें?
b) एएनएन में सीखना क्या है? एएनएन में महत्वपूर्ण सीखने की रणनीतियों को सूचीबद्ध
c) आर्टिफीसियल तंत्रिका नेटवर्क की शब्दावली दीजिए।
d) एकल और बहुस्तरीय नेटवर्क में उपयोग किए जाने वाले कुछ सक्रियण कार्यों को बताएं आउटपुट की गणना करें।
e) एएनएन का उपयोग किए गए वजन का क्या महत्व है?
f) जैविक तंत्रिका नेटवर्क और आर्टिफीसियल तंत्रिका नेटवर्क के बीच अंतर
g) रैखिक रूप से वियोज्य समस्या का उदाहरण दें?
h) मैककुलोच पिट के आर्टिफीसियल न्यूरॉन मॉडल पर संक्षेप में चर्चा करें। अपनी सीमाएं बताइए।
i) एएनएन में महत्वपूर्ण शिक्षण कानूनों को सूचीबद्ध करें। संक्षेप में हेब्बियन सीखने पर चर्चा करें।
j) एएनएन में प्रयुक्त विभिन्न सक्रियण कार्यों की व्याख्या कीजिए।
k) मैककुलोच-पिट्स न्यूरॉन मॉडल का उपयोग करके फ़ंक्शन का एहसास न करें।
l) मैककुलोच-पिट्स न्यूरॉन मॉडल का उपयोग करके एक्स-ओआर फ़ंक्शन का एहसास करें।
m) आवर्तक तंत्रिका नेटवर्क क्या है.

3. लंबे प्रकार के प्रश्न (**10** अंक):

a) विजेता-टेक-ऑल लर्निंग लॉ बनाएं।
b) एएनएन के विभिन्न आर्किटेक्चर पर चर्चा करें।
c) हेब लर्निंग नियम को विस्तार से समझाएं।
d) परसेप्ट्रॉन लर्निंग रूल को विस्तार से समझाइए?
e) विड्रो-हॉफ लर्निंग नियम को विस्तार से समझाएं।

ग्रंथ-सूची

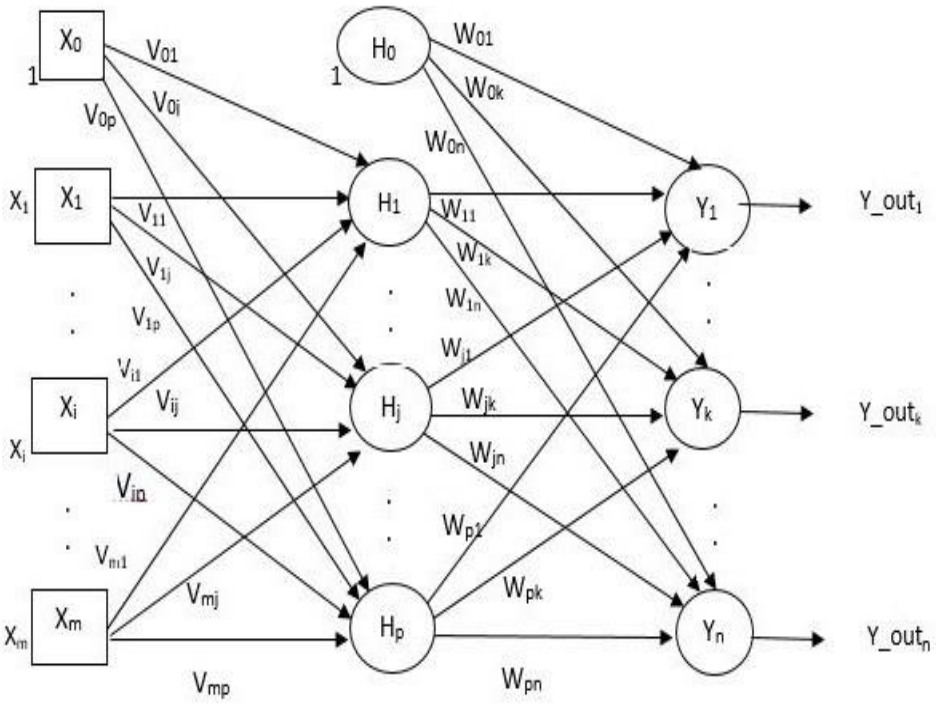
- एंडरसन, जे ए और रोसेनफील्ड, ई। (1988). *न्यूरोकंप्यूटिंग: अनुसंधान की नींव।*. कैम्ब्रिज, एमए, एमआईटी प्रेस।
- वॉन न्यूमैन, जे (1958)। *कंप्यूटर और मस्तिष्क.* न्यू हेवन: येल यूनिवर्सिटी प्रेस।
- रोसेनब्लैट, एफ (1959)। परसेप्ट्रॉन में सांख्यिकीय अलगाव के दो प्रमेय। *विचार प्रक्रिया का मशीनीकरण: राष्ट्रीय भौतिक प्रयोगशाला में संगोष्ठी की कार्यवाही।*लंदन, पीपी - 421-456।
- रोसेनब्लैट, एफ (1962)। *न्यूरोडायनामिक्स के सिद्धांत.* स्पार्टन बुक्स।
- मिंस्की, एम एंड पेपर, एस (1969)। *परसेप्ट्रोन.* एमआईटी प्रेस।
- - मैककुलोच, डब्ल्यू एंड पिट्स, डब्ल्यू (1943)। तंत्रिका गतिविधि में निहित विचारों का एक तार्किक कैलकुलेशन। *गणितीय बायोफिज़िक्स का बुलेटिन,* 7, 99. 115-133.
- ब्लॉक, एचडी (1962)। परसेप्ट्रॉन: मस्तिष्क के कामकाज के लिए एक मॉडल। *आधुनिक भौतिकी के लिए समीक्षाएं.* खंड 34, पीपी 123-135।
- - कोहेन, एमए और ग्रॉसबर्ग, एस (1983)। प्रतिस्पर्धी तंत्रिका नेटवर्क द्वारा वैश्विक पैटर्न गठन और समानांतर स्मृति भंडारण की पूर्ण स्थिरता। *सिस्टम, मैन और साइबरनेटिक्स पर आईईईई लेनदेन,* एसएमसी -13, पीपी 815-826।
- ग्रॉसबर्ग, एस (1973)। *मन और मस्तिष्क का अध्ययन.* बोस्टन, रिडेल।
- हेब, डी ओ (1949)। *व्यवहार का संगठन.* न्यूयॉर्क, जॉन विली एंड संस।
- रॉय, एस एंड चक्रवर्ती, यू (2013)। *सॉफ्ट कंप्यूटिंग का परिचय: न्यूरो-फजी और जेनेटिक एल्गोरिदम.* पियर्सन।
- पाढ़ी, एनपी और साइमन, एसपी (2015)। *MATLAB प्रोग्रामिंग के साथ सॉफ्ट कंप्यूटिंग.* ऑक्सफोर्ड यूनिवर्सिटी प्रेस।

अध्याय 8

न्यूरल नेटवर्क का उपयोग कर मशीन लर्निंग : उन्नत

8.1 बैकप्रोपेगेशन

बैकप्रोपेगेशन एल्गोरिदम एक पर्यवेक्षित शिक्षण विधि है जहां न्यूरल नेटवर्क बार-बार प्रशिक्षण इनपुट के जवाब में लक्ष्य आउटपुट से त्रुटि या विचलन के आधार पर अपने इंटरकनेक्शन वजन को समायोजित करता है। प्रशिक्षण कई पुनरावृत्तियों या युगों के माध्यम से होता है और प्रत्येक युग के दौरान प्रशिक्षण इनपुट इनपुट को इनपुट परत को दिया जाता है और सिग्नल इनपुट से छिपी परतों के माध्यम से आउटपुट परत तक प्रवाहित होते हैं। त्रुटियों की गणना आउटपुट परत पर शुरू होती है और त्रुटियों को आउटपुट परत से इनपुट परत की ओर पीछे की ओर प्रचारित किया जाता है। मल्टीलेयर फीड फॉरवर्ड आर्टिफिशियल न्यूरल नेटवर्क के लिए बैकप्रोपेगेशन एल्गोरिथ्म जिसमें एक छिपी हुई परत होती है जैसा कि चित्र में दर्शाया गया है। 8.1 निम्नानुसार दिया गया है:



अंजीर। 8.1. एक छिपी हुई परत के साथ एक बहुस्तरीय फ़ीड फॉरवर्ड आर्टिफिशियल न्यूरल नेटवर्क।

एल्गोरिथ्म **8.1:** प्रशिक्षण जोड़े के एक सेट के साथ एकल छिपी परत के साथ एक मल्टीलेयर फीड फॉरवर्ड आर्टिफिशियल न्यूरल नेटवर्क को प्रशिक्षित करने के लिए बैकप्रोपेगेशन एल्गोरिदम:

$$(\boldsymbol{x}, \boldsymbol{t})\ \boldsymbol{where}\ \boldsymbol{x} = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_m}]\ \boldsymbol{and}\ \boldsymbol{t} = [\boldsymbol{t_1}, \ldots, \boldsymbol{t_n}]$$

चरण **1:** **वजन शुरू करें** v_{ij} **और** w_{jk} **छोटे के साथ बेतरतीब ढंग से मान।**

चरण **2:** **जबकि रोकने का मानदंड पूरा नहीं होता है, चरण** 3 **दोहराएं** - 10.

चरण **3:** **प्रत्येक प्रशिक्षण जोड़ी के लिए, चरण** 4 - 9 **दोहराएं।**

चरण **4:** **पैटर्न लागू करें** $x = [x_1, \dots, x_m]$ **इनपुट परत के लिए।** $[X_1, \dots, X_m]$ **और संबंधित आउटपुट भेजें** $x_1, \dots, x_m$ **छिपी हुई परत के लिए। इकाइयों।**

चरण **5:** **कुल इनपुट की गणना करें** h_{in_j} **प्रत्येक छिपी हुई परत इकाई के लिए** H_j **जैसा**:

$$h_{in_j} = \sum_{i=0}^{m} x_i \cdot v_{ij} = v_{0j} + \sum_{i=1}^{m} x_i \cdot v_{ij}, \ j = 1, \dots, p$$

प्रत्येक छिपी हुई परत इकाई के आउटपुट की गणना करें:

$$h_{out_j} = f_h(h_{in_j})$$

यह आउटपुट भेजें h_{out_j} प्रत्येक इकाई के लिए Y_k आउटपुट परत का।

चरण **6:** **कुल इनपुट की गणना करें** y_{in_k} **प्रत्येक इकाई के लिए** Y_k आउटपुट की मात्रा परत के रूप में:

$$y_{in_k} = \sum_{j=0}^{p} h_{out_j} \cdot w_{jk} = w_{0k} + \sum_{j=1}^{p} h_{out_j} \cdot w_{jk}, \qquad k = 1, \dots, n$$

प्रत्येक आउटपुट लेयर इकाई के आउटपुट की गणना करें:

$$y_{out_k} = f_o(y_{in_k})$$

चरण **7:** **आउटपुट परत पर त्रुटि की गणना करें**:

$$\delta w_k = (t_k - y_{out_k}) \cdot f_o{}'(y_{in_k}), \qquad k = 1, \dots, n$$

वजन के लिए वजन अपडेशन की गणना करें w_{jk} जैसा:

$$\Delta w_{jk} = \eta \cdot \delta w_k \cdot h_{out_j}$$

और

$$\Delta w_{0k} = \eta \cdot \delta w_k$$

कहां $j = 1, \dots, p$ और $k = 1, \dots, n$ और η **सीखने की दर है.** अब त्रुटि भेजें δw_k **छिपी हुई परत के पीछे की ओर।**

चरण **8:** **प्रत्येक छिपी हुई परत इकाई के लिए H_j, कुल त्रुटि की गणना करें आउटपुट परत से प्राप्त:**

$$\delta v_{in_j} = \sum_{k=1}^{n} \delta_k \cdot w_{jk}$$

छिपी हुई परत पर त्रुटि की गणना करें:

$$\delta v_j = \delta v_{in_j} \cdot f_h'\left(h_{in_j}\right), \quad j = 1, \dots, p$$

वजन अद्यतन करने के लिए वजन सुधार शर्तों की गणना करें v_{ij} जैसा:

$$\Delta v_{ij} = \eta \cdot \delta v_j \cdot x_i$$

और

$$\Delta v_{0j} = \eta \cdot \delta v_j$$

चरण **9:** **नए वजन को अपडेट करें:**

a. $w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}, \; for\, j = 0, \dots, p \; and \; k = 1, \dots, n$

b. $v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}, for\, i = 0, \dots, m \; and \; j = 1, \dots, p$

चरण **10:** **रोकने के मानदंड के लिए परीक्षण।**

8.2 एडालिन

1960 में विड्रो और हॉफ द्वारा प्रस्तावित, अनुकूली रैखिक न्यूरॉन (एडीएलाइन) को दो वर्गों में से किसी एक में इनपुट पैटर्न देने के द्विआधारी वर्गीकरण के लिए नियोजित किया गया था। वास्तुकला की दृष्टि से एक परसेप्ट्रॉन के समान, इसमें कई इनपुट इकाइयां और एक एकल आउटपुट इकाई शामिल थी। कई

इनपुट इकाइयों में से, इनपुट इकाई में से एक नेटवर्क में पूर्वाग्रह के रूप में कार्य करता है और स्थायी रूप से मान 1 पर तय किया जाता है। अन्य इनपुट इकाइयों की संख्या इनपुट पैटर्न की लंबाई के लिए तय की जाती है। यह द्विध्रुवी रूप में इनपुट लेता है और द्विध्रुवी य रूप में आउटपुट भी पैदा करता है। एडालाइन तंत्रिका नेटवर्क को अध्याय 7 के एल्गोरिथ्म 7.1 में वर्णित विड्रो-हॉफ सीखने के नियम का उपयोग करके प्रशिक्षित किया जाता है। **लक्ष्य मान और तंत्रिका नेटवर्क द्वारा अनुमानित मूल्य के बीच औसत वर्ग त्रुटि को कम करने का लक्ष्य है। यह प्रशिक्षण के दौरान आउटपुट इकाई पर पहचान सक्रियण फ़ंक्शन का उपयोग करता है। दूसरी ओर यह एप्लिकेशन के दौरान आउटपुट यूनिट के सक्रियण फ़ंक्शन के रूप में बाइपोलर स्टेप फ़ंक्शन का उपयोग करता है। सीखने की दर**, η ADALINE **का गठन निम्नलिखित असमानता के आधार पर किया जाता है**:

$$0.1 \leq n \times \eta \leq 1.0$$

कहाँ 'n' ADALINE नेटवर्क में इनपुट इकाइयों की कुल संख्या है।

एडालाइन तंत्रिका नेटवर्क सीखने के लिए विस्तृत एल्गोरिथ्म नीचे एल्गोरिथ्म 8.2 में प्रस्तुत किया गया है:

एल्गोरिथ्म **8.2:** एडालिन लर्निंग एल्गोरिदम

चरण **1: रेंज में वजन को यादृच्छिक करें।** $0 < w_i < 1.$

चरण **2: सेट करें सीख दर** η **असमानता के आधार पर** $0.1 \leq n \times \eta \leq 1.0$**, जहां** 'n' **इनपुट की कुल संख्या है** ADALINE **नेटवर्क में इकाइयाँ।**

चरण **3: चरण** 4 -7 **को तब तक दोहराएँ जब तक कि रोकने की स्थिति न हो हासिल।**

चरण **4: प्रत्येक द्विध्रुवी जोड़ी** s:t **के लिए, चरण** 5-7 **दोहराएँ**:

चरण **5: सेट करें इनपुट इकाइयों के लिए आउटपुट इस प्रकार है**:

$x_0 = 1,$ **(पूर्वाग्रह के स्थायी आउटपुट को** 1 **के रूप में सेट करें),**

$x_i = s_i,$ **के लिए** $i = 1,2, \dots, n$

चरण 6: **आउटपुट इकाई के लिए कुल इनपुट की गणना इस प्रकार करें**:

$$Y_{in} = \sum_{i=0}^{n} x_i . w_i$$

चरण 7: **विड्रो-हॉफ लर्निंग लागू करके वजन अपडेट करें।**
नियम के रूप में:

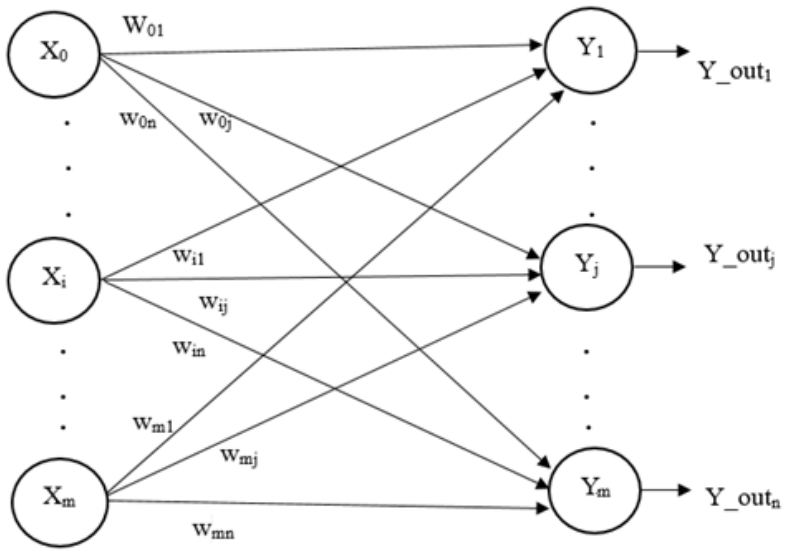
$$w_{i\,new} = w_{i\,old} + \eta \times (t - Y_{in}) \times X_i \; for \; i = 0, \ldots, n$$

8.3 ऑटो-एसोसिएटिव **नेट्स**

पैटर्न एसोसिएशन किसी दिए गए इनपुट पैटर्न को मशीन की मेमोरी में पहले से संग्रहीत पैटर्न से संबंधित करने की प्रक्रिया है। औपचारिक रूप से, पैटर्न एसोसिएशन को इनपुट पैटर्न से संबंधित कार्य के रूप में कहा जा सकता है। s_i इसके संबंधित संग्रहीत पैटर्न के लिए। t_i इनपुट-आउटपुट पैटर्न जोड़े को देखते हुए $s_1 : t_1, s_2 : t_2, \ldots, s_n : t_n$ कहां $i = 1, 2, \ldots, n.$

एसोसिएटिव मेमोरी न्यूरल नेटवर्क तंत्रिका नेटवर्क का एक उप-समूह है जिसका उपयोग वजन के रूप में उनकी मेमोरी में पैटर्न संघों के सेट को स्टोर करने के लिए किया जाता है। एक एसोसिएटिव मेमोरी न्यूरल नेटवर्क या तो एक फीडफॉरवर्ड न्यूरल नेटवर्क या आवर्तक तंत्रिका नेटवर्क हो सकता है। यह अध्याय तीन प्रकार के सहयोगी तंत्रिका नेटवर्क का वर्णन करता है, अर्थात् ऑटो-एसोसिएटिव, हॉपफील्ड नेटवर्क और द्विदिश एसोसिएटिव मेमोरी। ऑटो-एसोसिएटिव में एक सरल फीडफॉरवर्ड न्यूरल नेटवर्क आर्किटेक्चर होता है जबकि हॉपफील्ड नेटवर्क और द्विदिश एसोसिएटिव मेमोरी एक आवर्तक तंत्रिका नेटवर्क आर्किटेक्चर का दावा करते हैं।

ऑटो-एसोसिएटिव न्यूरल नेटवर्क में, इनपुट और आउटपुट पैटर्न समान हैं। नेटवर्क एक इनपुट पैटर्न दिए



गए एक ही पैटर्न को लौटाता है यदि यह संग्रहीत पैटर्न में से एक से मेल खाता है। वही

अंजीर। 8.2. एक ऑटो-एसोसिएटिव न्यूरल नेटवर्क।

एक ऑटो-एसोसिएटिव न्यूरल नेटवर्क की वास्तुकला में एक फीडफॉरवर्ड न्यूरल नेटवर्क होता है जिसमें इनपुट और आउटपुट इकाइयों की समान संख्या होती है जैसा कि चित्र 8.2. में दर्शाया गया है।

अध्याय 7 के एल्गोरिथ्म 7.1 में वर्णित हेब लर्निंग नियम को नियोजित करके एक ऑटो-एसोसिएटिव न्यूरल नेटवर्क को प्रशिक्षित किया जाता है। एक पैटर्न पर विचार करें $s = [s_1, s_2, \dots, s_m]$ जो एक ऑटो-एसोसिएटिव न्यूरल नेटवर्क पर संग्रहीत होता है। फिर वजन मैट्रिक्स W तंत्रिका नेटवर्क का है:

$$W = s^T \times s$$

$$W = \begin{pmatrix} s_1 \\ \cdot \\ \cdot \\ s_i \\ \cdot \\ \cdot \\ s_m \end{pmatrix} \times \begin{pmatrix} s_1 \dots s_j \dots s_m \end{pmatrix}$$

$$W = \begin{pmatrix} s_1 \times s_1 & \cdots & s_1 \times s_m \\ \vdots & \ddots & \vdots \\ s_m \times s_1 & \cdots & s_m \times s_m \end{pmatrix}$$

वजन मैट्रिक्स की गणना के बाद W, ऑटो-एसोसिएटिव न्यूरल नेटवर्क किसी भी इनपुट पैटर्न को पहचानने के लिए तैयार है। $x = [x_1, x_2, \dots, x_m]$ लंबाई की संख्या m यदि दिया गया इनपुट पैटर्न निम्नलिखित चरणों का पालन करके ऑटो-एसोसिएटिव न्यूरल नेटवर्क में संग्रहीत पैटर्न से मेल खाता है: –

1. प्रत्येक आउटपुट इकाई के लिए $Y_i, i = 1, 2, \dots, m,$ कुल इनपुट की गणना करें Y_{in} आउटपुट इकाई के लिए। Y_i जैसा:

   $$Y_{in} = x \times W$$

   कहां $Y_{in} = [y_{in_1}, y_{in_2}, \dots, y_{in_i}], x = [x_1, x_2, \dots, x_m]$ और W वजन मैट्रिक्स क्या है?
   ऊपर गणना की गई।

2. प्रत्येक आउटपुट इकाई के लिए Y_i, द्विध्रुवी स्टेप सक्रियण फ़ंक्शन का उपयोग करके नेटवर्क आउटपुट निम्नानुसार ढूंढें:

   $$Y_{out} = \begin{cases} 1, & if\ Y_{in} > 0, \\ -1, & otherwise \end{cases}$$

यदि आउटपुट Y_{out} संग्रहीत पैटर्न के समान मूल्य है तो इनपुट पैटर्न को मान्यता दी गई है, अन्यथा नहीं।

उदाहरण

एक ऑटो-एसोसिएटिव न्यूरल नेटवर्क पर विचार करें जिसे पैटर्न को स्टोर और पहचानना है। $s = [1, -1, 1, -1]$. फिर उपरोक्त ऑटो-एसोसिएटिव न्यूरल नेटवर्क के लिए वजन मैट्रिक्स द्वारा दिया गया है:

$$W = s^T \times s$$

$$W = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \times (1 \quad -1 \quad 1 \quad -1)$$

$$W = \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix}$$

अब, दिए गए इनपुट पैटर्न को पहचानने के लिए $x = [1, -1, 1, -1]$, उपरोक्त वर्णित दो चरणों का पालन किया जाता है:

1. कुल इनपुट की गणना करें Y_{in} आउटपुट इकाई के रूप में:

   $$Y_{in} = x \times W$$

$$Y_{in} = [1 \ \ -1 \ \ 1 \ \ -1] \times \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix}$$

$$Y_{in} = [4 \ \ -4 \ \ 4 \ \ -4]$$

2. द्विध्रुवी चरण सक्रियण फ़ंक्शन का उपयोग करके आउटपुट की गणना करें:

$$Y_{out} = [Y_{out_1}, Y_{out_2}, Y_{out_3}, Y_{out_4}]$$

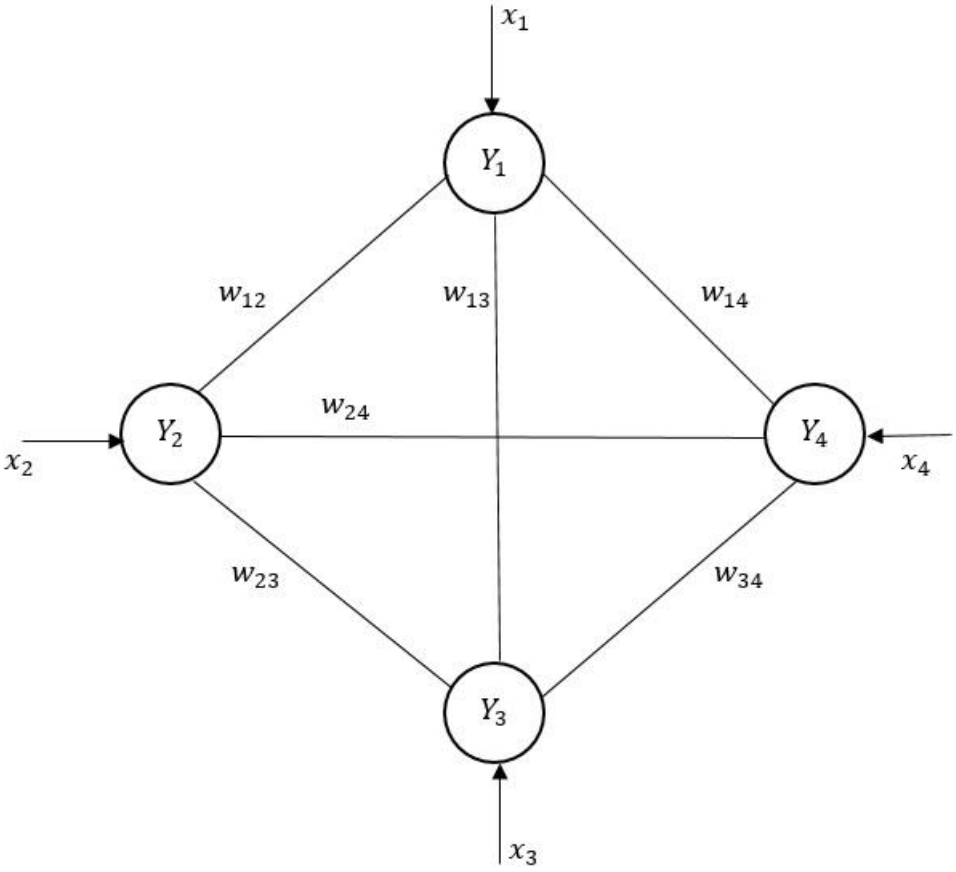
$$Y_{out} = [1, -1, 1, -1]$$

इस प्रकार ऑटो-एसोसिएटिव न्यूरल नेटवर्क ने संग्रहीत पैटर्न से दिए गए इनपुट पैटर्न को मान्यता दी है।

8.4 हॉपफील्ड नेटवर्क

जॉन हॉपफील्ड के नाम पर हॉपफील्ड तंत्रिका नेटवर्क आवर्तक तंत्रिका नेटवर्क वास्तुकला का उपयोग करने वाले ऑटो-एसोसिएटिव न्यूरल नेटवर्क का सबसे पहला कार्यान्वयन है।

एक हॉपफील्ड तंत्रिका नेटवर्क में, प्रत्येक नोड पूरी तरह से सममित भार के साथ हर दूसरे नोड से जुड़ा होता है और इसमें कोई स्व-लूप नहीं होता है। इस प्रकार, प्रत्येक नोड को दिए गए इनपुट पैटर्न के संबंधित घटक के साथ तंत्रिका नेटवर्क में अन्य सभी नोड्स से अपना इनपुट सक्रियण मिलता है। एक पूरी तरह से जुड़े 4 इकाई हॉपफील्ड तंत्रिका नेटवर्क की संरचना को चित्र 8.3 में दर्शाया गया है।



चित्र 8.3. हॉपफील्ड न्यूरल नेटवर्क आर्किटेक्चर।

इस प्रकार, एक के लिए 'nयूनिट हॉपफील्ड न्यूरल नेटवर्क, किसी दिए गए यूनिट के लिए शुद्ध इनपुट Y_i इस प्रकार दिया गया है:

$$Y_{in_i} = x_i + \sum_{\substack{p=1 \\ p \neq q}}^{n} y_p \cdot w_{pq}$$

कहां x_i यह है i^{th} इनपुट पैटर्न का तत्व $x = [x_1, x_2, \dots, x_n]$ और y_p इकाई का वर्तमान आउटपुट क्या है? Y_p.

अध्याय 7 के एल्गोरिथ्म 7.1 में वर्णित हेब लर्निंग नियम को नियोजित करके एक हॉपफील्ड तंत्रिका नेटवर्क को भी प्रशिक्षित किया जाता है। एक इनपुट पैटर्न पर विचार करें। $s = [s_1, s_2, \dots, s_m]$ जो हॉपफील्ड न्यूरल नेटवर्क में संग्रहीत है। फिर वजन मैट्रिक्स।W तंत्रिका नेटवर्क का दिया गया है:

$$W = s^T \times s$$

$$W = \begin{pmatrix} s_1 \\ . \\ . \\ s_i \\ . \\ . \\ s_m \end{pmatrix} \times (s_1 \dots s_j \dots s_m)$$

$$W = \begin{pmatrix} s_1 \times s_1 & \cdots & s_1 \times s_m \\ \vdots & \ddots & \vdots \\ s_m \times s_1 & \cdots & s_m \times s_m \end{pmatrix}$$

वजन मैट्रिक्स की गणना के बाद W, हॉपफील्ड तंत्रिका नेटवर्क किसी भी इनपुट पैटर्न को पहचानने के लिए तैयार है $x = [x_1, x_2, \dots, x_m]$ लंबाई की संख्या m. हॉपफील्ड तंत्रिका नेटवर्क को लागू करने के लिए विस्तृत एल्गोरिथ्म नीचे एल्गोरिथ्म 8.3 में प्रस्तुत किया गया है:

एल्गोरिथ्म **8.3:** हॉपफील्ड तंत्रिका नेटवर्क लर्निंग एल्गोरिदम

चरण **1:** **इनपुट मानों के साथ इकाइयों के आउटपुट को प्रारंभ करें।**

$$Y_{out_i} = x_i \quad for\ i = 1, 2, \dots, m$$

चरण **2:** **इकाइयों के आउटपुट को यादृच्छिक तरीके से अपडेट करें**
प्रत्येक इकाई के लिए चरण 3-4 **दोहराना** Y_j **नेटवर्क में।**

चरण **3:** **कुल इनपुट के लिए** Y_{in_j} इकाई के लिए Y_j as

$$Y_{in_j} = x_j + \sum_{i \neq j}^{n} y_{out_i} \times w_{ij}$$

चरण **4:** **इकाई का आउटपुट ज्ञात कीजिये** Y_j **जैसा**:

$$Y_{out_j} = 1, if\ Y_{in_j} > 0,$$
$$Y_{out_j} = Y_{out_j}, if\ Y_{in_j} = 0,$$

$$Y_{out_j} = 0, if\ Y_{in_j} < 0,$$

जब इकाई का उत्पादन Y_j गणना की जाती है, इसे भेजा जाता है अपने आउटपुट के अद्यतन के लिए नेटवर्क में प्रत्येक इकाई।

चरण 5: **यदि इकाइयों के आउटपुट में कोई और बदलाव नहीं होता है नेटवर्क में तो अभिसरण हासिल किया गया है और सीखना बंद हो सकता है।**

उदाहरण

मान लीजिए कि हम एक पैटर्न स्टोर करना चाहते हैं। $s = [1, 0, 1, 0]$ एक हॉपफील्ड तंत्रिका नेटवर्क में। दिए गए इनपुट पैटर्न को द्विध्रुवीय रूप में परिवर्तित करना हमें प्राप्त होता है $s = [1, -1, 1, -1]$. फिर हॉपफील्ड तंत्रिका नेटवर्क के लिए वजन मैट्रिक्स द्वारा दिया गया है:

$$W = s^T \times s$$

$$W = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \times (1 \quad -1 \quad 1 \quad -1)$$

$$W = \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix}$$

चूंकि, हॉपफील्ड तंत्रिका नेटवर्क में कोई स्व-लूप की अनुमति नहीं है, इस प्रकार वजन मैट्रिक्स में विकर्ण तत्व शून्य पर सेट होते हैं। इस प्रकार अंतिम वजन मैट्रिक्स इस प्रकार दिया गया है:

$$W = \begin{pmatrix} 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ -1 & 1 & -1 & 0 \end{pmatrix}$$

अब, दिए गए इनपुट पैटर्न को पहचानने के लिए $x = [1, 0, 1, 0]$, ऊपर वर्णित शिक्षण एल्गोरिथ्म का पालन किया जाता है:

1. इनपुट होने दें $x = [1, 0, 1, 0]$.
2. इकाइयों के आउटपुट को इस प्रकार प्रारंभ करें:

$$Y = [Y_{out_1}, Y_{out_2}, Y_{out_3}, Y_{out_4}]$$

3. पहले पुनरावृत्ति में, यादृच्छिक क्रम में इकाइयों के आउटपुट को अपडेट करें। Y_1, Y_3, Y_2, Y_4.
   आउटपुट अद्यतन करेंY_{out_1} का Y_1:

$$Y_{out_1} = x_1 + \sum_{i=1}^{4} Y_{out_i} \times w_{i1}$$

$$Y_{out_1} = 1 + (1 \times 0 + 0 \times (-1) + 1 \times 1 + 0 \times (-1))$$

$$Y_{out_1} = 2$$

$$\therefore Y_{out_1} = 1$$

आउटपुट अद्यतन करेंY_{out_3} का Y_3:

$$Y_{out_3} = x_3 + \sum_{i=1}^{4} Y_{out_i} \times w_{i3}$$

$$Y_{out_3} = 1 + (1 \times 1 + 0 \times (-1) + 1 \times 0 + 0 \times (-1))$$

$$Y_{out_3} = 2$$

$$\therefore Y_{out_3} = 1$$

आउटपुट अद्यतन करेंY_{out_2} का Y_2:

$$Y_{out_2} = x_2 + \sum_{i=1}^{4} Y_{out_i} \times w_{i2}$$

$$Y_{out_2} = 0 + (1 \times (-1) + 0 \times 0 + 1 \times (-1) + 0 \times 1)$$

$$Y_{out_2} = -2$$

$$\therefore Y_{out_2} = 0$$

आउटपुट अद्यतन करेंY_{out_4} का Y_4:

$$Y_{out_4} = x_4 + \sum_{i=1}^{4} Y_{out_i} \times w_{i4}$$

$$Y_{out_4} = 0 + (1 \times (-1) + 0 \times 1 + 1 \times (-1) + 0 \times 0)$$

$$Y_{out_4} = -2$$

$$\therefore Y_{out_4} = 0$$

इस प्रकार $Y = [Y_{out_1}, Y_{out_2}, Y_{out_3}, Y_{out_4}] = [1, 0, 1, 0]$

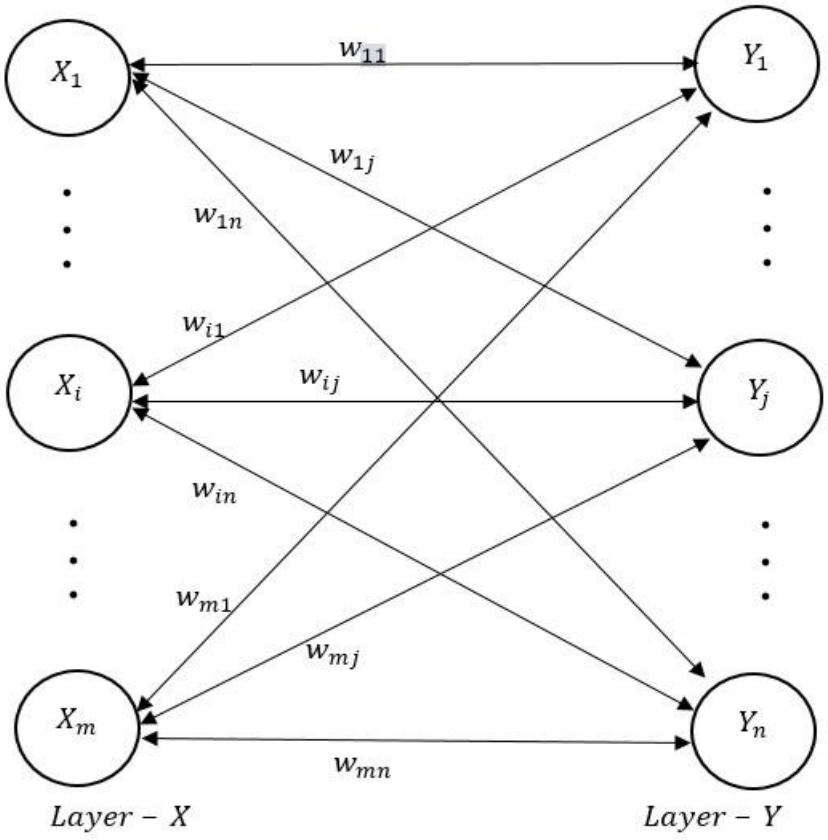
4. हॉपफील्ड तंत्रिका नेटवर्क की इकाइयों को दिए गए पैटर्न को पहचानने के लिए अपनी इकाइयों के आउटपुट को अपडेट करने की आवश्यकता नहीं है। इसलिए हॉपफील्ड तंत्रिका नेटवर्क अभिसरण हुआ है।

इसलिए, हॉपफील्ड तंत्रिका नेटवर्क पैटर्न को पहचान सकता है। $[1, 0, 1, 0]$.

8.5 द्विदिश एसोसिएटिव मेमोरी

1988 में कोस्को द्वारा विकसित एक द्विदिश एसोसिएटिव मेमोरी (बीएएम), एक हेटेरो एसोसिएटिव न्यूरल नेटवर्क है जो इकाइयों की दो परतों से बना है जो दिशात्मक लिंक से जुड़े होते हैं जो बीएएम को एक पैटर्न जोड़ी को स्टोर करने में मदद करता है। एक बीएएम तंत्रिका नेटवर्क को एक पैटर्न जोड़ी को स्टोर करने के लिए नियोजित किया जा सकता है। $s:t$ जैसे कि जब भी इसे इनपुट के रूप में किसी भी पैटर्न के साथ प्रदान किया जाता है तो यह आउटपुट के रूप में दूसरे को प्रदान करता है।

द्विदिश साहचर्य स्मृति (बीएएम) की वास्तुकला, जैसा कि चित्र 8.4 न्यूरल नेटवर्क में दर्शाया गया है, में दो परतें होती हैं जिनमें परत के रूप में लेबल की गई प्रसंस्करण इकाइयाँ होती हैं-X और परत-Y क्रमशः।



अंजीर। 8.4 एक द्विदिश सहयोगी मेमोरी तंत्रिका नेटवर्क

बीएएम तंत्रिका नेटवर्क में दोनों परतों के बीच कोई समझदार भेदभाव नहीं है क्योंकि दोनों को नेटवर्क की इनपुट या आउटपुट परत माना जा सकता है। यदि परत को इनपुट पैटर्न प्रदान किया जाता है-X

फिर यह परत के लिए इनपुट परत के रूप में कार्य करता है-Y जो आउटपुट परत के रूप में कार्य करता है और आउटपुट के रूप में मान्यता प्राप्त पैटर्न प्रदान करता है और इसके विपरीत।

बीएएम तंत्रिका नेटवर्क पैटर्न के जोड़े को संग्रहीत करता है। $s:t$ कहां $s = [s_1, s_2, \dots, s_m]$ और $t = [t_1, t_2, \dots, t_n]$ क्रमशः इस प्रकार कि m परत में इकाइयाँ-X और n परत में इकाइयाँ-Y ताकि पैटर्न प्रदान किया जा सके। s परत पर-X और पैटर्न। t परत पर-Y इनपुट के रूप में। वजन मैट्रिक्स W_{XY} परत को प्रदान किए गए इनपुट के लिए नेटवर्क के लिए-X और परत द्वारा उत्पादित आउटपुट-Y इसकी गणना इस प्रकार की जाती है:

$$W_{XY} = s^T \times t$$

$$W_{XY} = \begin{pmatrix} s_1 \\ . \\ . \\ s_i \\ . \\ . \\ s_m \end{pmatrix} \times (t_1 \dots t_j \dots t_n)$$

$$W_{XY} = \begin{pmatrix} s_1 \times t_1 & \cdots & s_1 \times t_n \\ \vdots & \ddots & \vdots \\ s_m \times t_1 & \cdots & s_m \times t_n \end{pmatrix}$$

वजन मैट्रिक्स W_{YX} परत को प्रदान किए गए इनपुट के लिए नेटवर्क के लिए-Y और परत द्वारा उत्पादित आउटपुट-X किसके प्रतिस्थापन की गणना की जाती है? W_{XY} अर्थात। ${W_{XY}}^T$.

बीएएम तंत्रिका नेटवर्क को लागू करने के लिए विस्तृत एल्गोरिथ्म नीचे एल्गोरिथ्म 8.4 में प्रस्तुत किया गया है:

एल्गोरिथ्म **8.4:** बीएएम न्यूरल नेटवर्क लर्निंग एल्गोरिदम

चरण **1:** **सभी इकाइयों के आउटपुट को प्रारंभ करें** 0.

$$X_{out_i} = 0 \quad for\ i = 1, 2, \dots, m$$
$$Y_{out_j} = 0 \quad for\ i = 1, 2, \dots, n$$

चरण **2:** **छोड़ देना** $x:y$ **बीएएम में खिलाए जाने वाले पैटर्न की जोड़ी बनें जहां**

$x = [x_1, x_2, \dots, x_m]$ और $y = [y_1, y_2, \dots, y_m]$. के आउटपुट सेट करें

परत-X तक x और **परत-**Y तक y. यदि इनपुट केवल इसके लिए दिया गया है

एक परत फिर दूसरी परत के आउटपुट 0 एस रहते हैं।

चरण 3: **परत के आउटपुट को अपडेट करें-**Y आउटपुट का उपयोग करने वाली इकाइयाँ परत-X. इकाई के लिए कुल इनपुट Y_j का **परत-**Y इसकी गणना इस प्रकार की जाती है:

$$Y_{in_j} = \sum_{i=1}^{m} x_i \times w_{ij} \qquad for\ j = 1, \dots, n$$

परत के आउटपुट की गणना करें-Y इकाइयाँ निम्नानुसार हैं:

$$Y_{out_j} = 1, if\ Y_{in_j} > 0,$$
$$Y_{out_j} = Y_{out_j}, if\ Y_{in_j} = 0,$$
$$Y_{out_j} = -1, if\ Y_{in_j} < 0$$

चरण 4: **परत के आउटपुट को अपडेट करें-**X आउटपुट का उपयोग करने वाली इकाइयाँ परत-Y. इकाई के लिए कुल इनपुट X_i का **परत-**X इसकी गणना इस प्रकार की जाती है:

$$X_{in_i} = \sum_{j=1}^{n} y_j \times w_{ij} \qquad for\ i = 1, \dots, m$$

परत के आउटपुट की गणना करें-Y इकाइयाँ निम्नानुसार हैं:

$$X_{out_i} = 1, if\ X_{in_i} > 0,$$
$$X_{out_i} = X_{out_i}, if\ X_{in_i} = 0,$$
$$X_{out_i} = -1, if\ X_{in_i} < 0$$

चरण 5: **अभिसरण प्राप्त होने पर सीखना बंद हो सकता है और बीएएम तंत्रिका नेटवर्क स्थिर हो गया है, अन्यथा चरण** 3**। और** 4.

उदाहरण

मान लीजिए कि हम एक पैटर्न जोड़ी स्टोर करना चाहते हैं। $s: t = [1, -1, 1, -1]: [1, -1]$ एक बीएएम तंत्रिका नेटवर्क में। फिर वजन मैट्रिक्स। W_{XY} परत से बीएएम तंत्रिका नेटवर्क के लिए-X परत पर-Y द्वारा दिया गया है:

$$W_{XY} = s^T \times t$$

$$W_{XY} = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \times (1 \quad -1 \quad)$$

$$W_{XY} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix}$$

इस प्रकार, वजन मैट्रिक्स W_{YX} परत से बीएएम तंत्रिका नेटवर्क के लिए-Y परत पर-X इसकी गणना किसके द्वारा की जाती है? W_{XY} जैसा:

$$W_{YX} = {W_{XY}}^T = \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix}$$

अब, दो मामले हैं। पहले मामले में हम पैटर्न प्रदान करते हैं। $[1, -1, 1, -1]$ परत के लिए-X इनपुट के रूप में और परत से आउटपुट की गणना करें-Y और फिर दूसरे मामले में हम पैटर्न प्रदान करते हैं। $[1, -1]$ परत के लिए-Y और परत से आउटपुट की गणना करें-X.

केस 1: परत पर इनपुट-X, $x = [1, -1, 1, -1]$.

i. सभी इकाइयों के सभी आउटपुट को 0 पर सेट करें।

$$x_{out_i} = 0 \quad i = 1, \dots, 4$$

$$y_{out_j} = 0 \quad j = 1,2$$

ii. पैटर्न लागू करें $x = [1, -1, 1, -1]$ परत की इकाइयों के उत्पादन के लिए-X, जैसे कि

$$x_{out_1} = 1, x_{out_2} = -1, x_{out_3} = 1, x_{out_4} = -1$$

iii. परत के लिए कुल इनपुट की गणना करें-Y और इसका आउटपुट इस प्रकार है:

$$Y_{in} = X_{out} \times W_{XY}$$

$$Y_{in} = [1, -1, 1, -1] \times \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$Y_{in} = [4, -4]$$

$$\therefore\ Y_{out} = [1, -1]$$

iv. परत के लिए कुल इनपुट की गणना करें-X परत से आउटपुट का उपयोग करना-Y जैसा:

$$X_{in} = Y_{out} \times W_{YX}$$

$$X_{in} = [1, -1] \times \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix}$$

$$X_{in} = [2, -2, 2, -2]$$

$$\therefore\ X_{out} = [1, -1, 1, -1]$$

इस प्रकार, बीएएम तंत्रिका नेटवर्क अभिसरण तक पहुंच गया है इसलिए हम रुक जाते हैं। जैसा $Y_{out} = [1, -1] = t$, हम इस निष्कर्ष पर आ सकते हैं कि बीएएम तंत्रिका नेटवर्क इनपुट पैटर्न को जोड़ सकता है। $s = [1, -1, 1, -1]$ तक $t = [1, -1]$.

केस 1: परत पर इनपुट-X, $x = [1, -1, 1, -1]$.

i. सभी इकाइयों के सभी आउटपुट को 0 पर सेट करें।

$$x_{out_i} = 0 \quad i = 1, \ldots, 4$$

$$y_{out_j} = 0 \quad j = 1,2$$

ii. पैटर्न लागू करें $x = [1, -1]$ परत की इकाइयों के उत्पादन के लिए-Y, जैसे कि

$$y_{out_1} = 1, y_{out_2} = -1$$

iii. परत के लिए कुल इनपुट की गणना करें-Y और इसका आउटपुट इस प्रकार है:

$$Y_{in} = X_{out} \times W_{XY}$$

$$Y_{in} = [0, 0, 0, 0] \times \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$Y_{in} = [0, 0]$$

जैसे, परत की इकाइयों के लिए कुल इनपुट-$Y0$ है, इस प्रकार उनके आउटपुट अपरिवर्तित रहते हैं।

$$\therefore\ Y_{out} = [1, -1]$$

iv. परत के लिए कुल इनपुट की गणना करें-X परत से आउटपुट का उपयोग करना-Y जैसा:

$$X_{in} = Y_{out} \times W_{YX}$$

$$X_{in} = [1, -1] \times \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix}$$

$$X_{in} = [2, -2, 2, -2]$$

$$\therefore X_{out} = [1, -1, 1, -1]$$

v. परत के आउटपुट की गणना करें-Y नई परत के साथ-X आउटपुट के रूप में:

$$Y_{in} = X_{out} \times W_{XY}$$

$$Y_{in} = [1, -1, 1, -1] \times \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$Y_{in} = [4, -4]$$

$$\therefore Y_{out} = [1, -1]$$

इस प्रकार, बीएएम तंत्रिका नेटवर्क अभिसरण तक पहुंच गया है इसलिए हम रुक जाते हैं। जैसा $Y_{out} = [1, -1] = t$, हम इस निष्कर्ष पर आ सकते हैं कि बीएएम तंत्रिका नेटवर्क इनपुट पैटर्न को जोड़ सकता है। $s = [1, -1, 1, -1]$ तक $t = [1, -1]$.

vi. परत के आउटपुट की गणना करें-X उपरोक्त नई गणना परत के साथ-Y आउटपुट के रूप में:

$$X_{in} = Y_{out} \times W_{YX}$$

$$X_{in} = [1, -1] \times \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix}$$

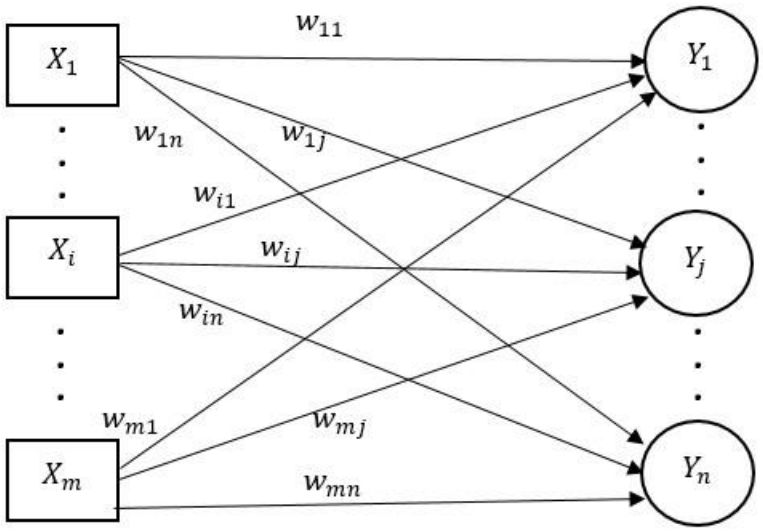
$$X_{in} = [2, -2, 2, -2]$$

$$\therefore X_{out} = [1, -1, 1, -1]$$

इस प्रकार, बीएएम तंत्रिका नेटवर्क अभिसरण तक पहुंच गया है इसलिए हम रुक जाते हैं और हम इस निष्कर्ष पर पहुंच सकते हैं कि बीएएम तंत्रिका नेटवर्क इनपुट पैटर्न को जोड़ सकता है। $s = [1, -1, 1, -1]$ तक $t = [1, -1]$.

8.6 सेल्फ ऑर्गनाइजिंग मैप

सेल्फ ऑर्गनाइजिंग मैप (एसओएम) तंत्रिका नेटवर्क का एक उप-समूह है जिसका उपयोग क्लस्टरिंग पैटर्न के उद्देश्य से किया जाता है, जिसमें विजेता की असुरक्षित सीखने की रणनीति को नियोजित किया जाता है, जो अध्याय 7 के एल्गोरिथ्म 7.4 में वर्णित है। एसओएम इनपुट पैटर्न को व्यवस्थित करता है जो एक ही क्लस्टर समूह में शारीरिक रूप से एक साथ करीब होते हैं। एक एसओएम नेटवर्क जिसका उद्देश्य 'एम' के पैटर्न को 'एन' क्लस्टर संख्या में क्लस्टर करना है, उसमें इनपुट इकाइयों की 'एम' संख्या और आउटपुट इकाइयों की 'एन' संख्या होनी चाहिए। इस प्रकार, एक एसओएम में आउटपुट इकाइयों की संख्या उन समूहों की कुल संख्या को इंगित करती है जिनमें इनपुट पैटर्स को समूहीकृत किया जाएगा। एसओएम द्वारा उपयोग की जाने वाली सीखने की रणनीति असुरक्षित सीखने की रणनीति है क्योंकि शुरू में केवल दिए गए इनपुट पैटर्न और आउटपुट क्लस्टर की कुल संख्या ज्ञात है और संबंध का कोई ज्ञान नहीं है।



चित्र 8.5 एक 'एम' इनपुट 'एन' आउटपुट सेल्फ ऑर्गनाइजिंग मशीन आर्किटेक्चर।

इनपुट पैटर्न और आउटपुट क्लस्टर के बीच जिसमें ये पैटर्न संबंधित हैं। सीखने की प्रक्रिया के दौरान, एसओएम खुद को इस तरह से व्यवस्थित करता है कि जो पैटर्न काफी करीब हैं, उन्हें एक ही क्लस्टर में वर्गीकृत किया जाता है। अंजीर। 8.5 एक विशिष्ट एम-इनपुट एन-आउटपुट एसओएम तंत्रिका नेटवर्क को दर्शाता है। इसमें शामिल हैं 'm' इनपुट इकाइयां $X_1, X_2, \dots, X_m$ और 'n' आउटपुट इकाइयाँ $Y_1, Y_2, \dots, Y_n$ वजन के साथ एक किनारे के साथ w_{ij} प्रत्येक इनपुट इकाई को कनेक्ट करना X_i प्रत्येक आउटपुट इकाई के साथ Y_j. छोड़ देना $s_1, s_{2,\dots,} s_p$ 'का एक सेट बनो'p' फॉर्म के पैटर्न की संख्या $s_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ जिन्हें 'में बांटा जाना है'n' क्लस्टर। $C_1, C_2, \dots, C_n$. पड़ोस में'Rएक समूह का '। Y_j इसे उन सभी इनपुट

पैटर्न के सेट के रूप में परिभाषित किया गया है जो ' के दायरे में स्थित हैंR' क्लस्टर के केंद्र से इकाइयाँ Y_j. स्व आयोजन मानचित्र को लागू करने के लिए विस्तृत एल्गोरिथ्म नीचे एल्गोरिथ्म 8.5 में प्रस्तुत किया गया है:

एल्गोरिथ्म **8.5** सेल्फ ऑर्गनाइजिंग मैप मशीन लर्निंग एल्गोरिदम

चरण **1:** **वजन मैट्रिक्स (**w_{ij} **सभी** I **और** j **के लिए,**
पड़ोस की दूरी R **और सीखने की दर।** η.

चरण **2:** **चरण** 3 - 9 **को तब तक दोहराएं जब तक कि रोकने का मानदंड न हो।**
के साथ।

चरण **3:** **प्रत्येक इनपुट पैटर्न के लिए** s_i **चरण** 4 - 6 **का पालन करें।**

चरण **4:** **प्रत्येक वजन वेक्टर के लिए** w_{*j}**, यूक्लिडियन दूरी ज्ञात कीजिये** $D(j)$
के बीच s_i **और** w_{*j} **निम्नानुसार**:

$$D(j) = \sqrt{\sum_i (x_i - w_{ij})^2}$$

चरण **5:** **अनुक्रमणिका ज्ञात कीजिये**'J' **ताकि** $D(J)$ **न्यूनतम है।**

चरण **6:** **निर्दिष्ट के भीतर सभी इकाइयों के वजन को अपडेट करें**
'के पड़ोस में'J' **इस प्रकार है**:

सभी के लिए J**, जैसे कि** $J - R \leq j \leq J + R$ **का**
सभी के लिए $i = 1 \, to \, m$ **का**

$$w_{ij\,new} = w_{ij\,old} + \eta \times [x_i - w_{ij\,old}]$$

चरण **7:** **सीखने की दर को अपडेट करें** η **निम्नानुसार**:

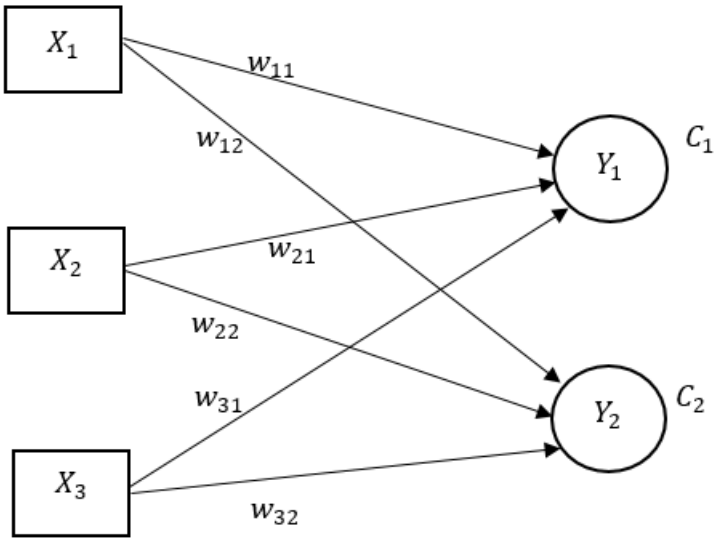
$$\eta = k \times \eta, \; 0 < k < 1$$

चरण **8:** **यदि आवश्यक हो तो** R **को कम करें।**

चरण **9:** **जांचें कि रोकने का मानदंड पूरा हुआ है या नहीं।**

उदाहरण

मान लीजिए कि चार इनपुट पैटर्न हैं। $s_1 = [1, 0, 1], s_2 = [0, 1, 0], s_3 = [1, 1, 1]$ **और** $s_4 = [1, 0, 0]$ **जिन्हें स्व-आयोजन मानचित्र नेटवर्क को दो समूहों में नियोजित करके समूहीकृत किया जाना है जैसा कि चित्र** 8.6. **में दिखाया गया है।**



चित्र 8.6 **उदाहरण एसओएम नेटवर्क**

भार मैट्रिक्स जिसमें समूहों के लिए वजन वैक्टर होते हैं C_1 **और** C_2 **इस प्रकार दर्शाया गया है**:

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix}$$

इस प्रकार $W_{*1} = [w_{11}, w_{21}, w_{31}]$ **और** $W_{*2} = [w_{12}, w_{22}, w_{32}]$ **समूहों के लिए वजन वैक्टर हैं।** C_1 **और** C_2 **क्रमशः। उपरोक्त समस्या का समाधान एसओएम लर्निंग एल्गोरिदम के आवेदन द्वारा नीचे वर्णित है**:

चरण **1: प्रारंभ**

(**ए) वजन मैट्रिक्स।** W **यादृच्छिक रूप से इस प्रकार**:

$$\begin{bmatrix} 0.4 & 0.6 \\ 0.7 & 0.3 \\ 0.5 & 0.8 \end{bmatrix}$$

(b) **पड़ोस की त्रिज्या** R 0 **पर प्रारंभ किया गया है।**

(c) **सीखने की दर** η 0.8 **तक प्रारंभ किया गया है और इसे एक कारक के साथ घटाया जाएगा**

प्रत्येक प्रशिक्षण पुनरावृत्ति के बाद ज्यामितीय रूप से 0.5 **की वृद्धि।**

चरण **2:** **चरण** 3 - 9 **को तब तक दोहराएं जब तक कि रोकने का मानदंड न हो। के साथ।**

जब वजन समायोजन $\Delta w_{ij} < 0.01$ **फिर एसओएम नेटवर्क मान लिया जाएगा। अभिसरण किया जाना था।**

चरण **3:** **प्रत्येक इनपुट पैटर्न के लिए** s_i **चरण** 4 - 6 **का पालन करें।**

पहला इनपुट पैटर्न क्या है? $s_1 = [1, 0, 1].$

चरण **4:** **प्रत्येक वजन वेक्टर के लिए** w_{*j}**, यूक्लिडियन दूरी ज्ञात कीजिये** $D(j)$ **के बीच** s_i **और** w_{*j} **निम्नानुसार**:

$$D(j) = \sqrt{\sum_i (x_i - w_{ij})^2}$$

$$D(1) = distance\ between\ s_1\ and\ W_{*1}$$

$$= \sqrt{(1-0.4)^2 + (0-0.7)^2 + (1-0.5)^2}$$

$$= \sqrt{0.36 + 0.49 + 0.25}$$

$$= \sqrt{1.1}$$

$$= 1.048$$

$$D(2) = distance\ between\ s_1\ and\ W_{*2}$$

$$= \sqrt{(1-0.6)^2 + (0-0.3)^2 + (1-0.8)^2}$$

$$= \sqrt{0.16 + 0.09 + 0.04}$$

$$= \sqrt{0.29}$$

$$= 0.538$$

चरण **5: अनुक्रमणिका ज्ञात कीजिये** 'J' **ताकि** $D(J)$ **न्यूनतम है।**

क्योंकि $D(2) < D(1), s_1$ के करीब है C_2 से C_1. इस प्रकार केवल वजन वेक्टर W_{*2} **अद्यतन किया जाएगा.**

चरण **6: निर्दिष्ट के भीतर सभी इकाइयों के वजन को अपडेट करें** '**के पड़ोस में**' J ' **इस प्रकार है**:

सभी के लिए J**, जैसे कि** $J - R \leq j \leq J + R$ **का**

सभी के लिए $i = 1\ to\ m$ **का**

$$w_{ij\ new} = w_{ij\ old} + \eta \times [x_i - w_{ij\ old}]$$

$$\begin{aligned} w_{12\ new} &= w_{12\ old} + \eta \times [x_1 - w_{12\ old}] \\ &= 0.6 + 0.8 \times [1 - 0.6] \\ &= 0.6 + 0.8 \times 0.4 \\ &= 0.92 \end{aligned}$$

$$\begin{aligned} w_{22\ new} &= w_{22\ old} + \eta \times [x_1 - w_{22\ old}] \\ &= 0.3 + 0.8 \times [0 - 0.3] \\ &= 0.3 + 0.8 \times -0.3 \\ &= 0.3 - 0.24 \\ &= 0.06 \end{aligned}$$

$$\begin{aligned} w_{32\ new} &= w_{32\ old} + \eta \times [x_1 - w_{32\ old}] \\ &= 0.8 + 0.8 \times [1 - 0.8] \\ &= 0.8 + 0.8 \times 0.2 \\ &= 0.96 \end{aligned}$$

इस प्रकार, पहले इनपुट पैटर्न के साथ प्रशिक्षण के बाद अद्यतन वजन मैट्रिक्स पुनरावृत्ति है:

$$\begin{bmatrix} 0.4 & 0.92 \\ 0.7 & 0.06 \\ 0.5 & 0.96 \end{bmatrix}$$

अब इनपुट पैटर्न के लिए उपरोक्त चरण 2 – 6 का प्रदर्शन करें $s_2 = [0, 1, 0]$:

चरण **2:** **चरण** 3 - 9 **को तब तक दोहराएं जब तक कि रोकने का मानदंड न हो। के साथ।**

चरण **3:** **प्रत्येक इनपुट पैटर्न के लिए** s_i **चरण** 4 - 6 **का पालन करें।**
दूसरा इनपुट पैटर्न क्या है? $s_2 = [0, 1, 0]$.

चरण **4:** **प्रत्येक वजन वेक्टर के लिए** w_{*j}**, यूक्लिडियन दूरी ज्ञात कीजिये** $D(j)$ **के बीच** s_i **और** w_{*j} **निम्नानुसार**:

$$D(j) = \sqrt{\sum_i (x_i - w_{ij})^2}$$

$$D(1) = distance\ between\ s_1\ and\ W_{*1}$$
$$= \sqrt{(0-0.4)^2 + (1-0.7)^2 + (0-0.5)^2}$$
$$= \sqrt{0.16 + 0.09 + 0.25}$$
$$= \sqrt{0.50}$$
$$= 0.707$$

$$D(2) = distance\ between\ s_1\ and\ W_{*2}$$
$$= \sqrt{(0-0.92)^2 + (1-0.06)^2 + (0-0.96)^2}$$
$$= \sqrt{0.84 + 0.88 + 0.92}$$
$$= \sqrt{2.64}$$
$$= 1.62$$

चरण **5:** **अनुक्रमणिका ज्ञात कीजिये** $'J'$ **ताकि** $D(J)$ **न्यूनतम है।**

क्योंकि $D(1) < D(2)$, s_2 के करीब है C_1 से C_2. इस प्रकार केवल वजन वेक्टर W_{*1} **अद्यतन किया जाएगा.**

चरण **6: निर्दिष्ट के भीतर सभी इकाइयों के वजन को अपडेट करें 'के पड़ोस में'J' इस प्रकार है:**

सभी के लिएJ, जैसे कि $J-R \leq j \leq J+R$ का

सभी के लिए $i = 1\ to\ m$ का

$$w_{ij\ new} = w_{ij\ old} + \eta \times [x_i - w_{ij\ old}]$$

$$w_{11\ new} = w_{11\ old} + \eta \times [x_1 - w_{11\ old}]$$
$$= 0.4 + 0.8 \times [0 - 0.4]$$
$$= 0.4 + 0.8 \times -0.4$$
$$= 0.4 - 0.32$$
$$= 0.08$$

$$w_{21\ new} = w_{21\ old} + \eta \times [x_1 - w_{21\ old}]$$
$$= 0.7 + 0.8 \times [1 - 0.7]$$
$$= 0.7 + 0.8 \times 0.3$$
$$= 0.94$$

$$w_{31\ new} = w_{31\ old} + \eta \times [x_1 - w_{31\ old}]$$
$$= 0.5 + 0.8 \times [0 - 0.5]$$
$$= 0.5 + 0.8 \times 0.2$$
$$= 0.5 - 0.40$$
$$= 0.10$$

इस प्रकार, पहले पुनरावृत्ति में दूसरे इनपुट पैटर्न के साथ प्रशिक्षण के बाद अद्यतन वजन मैट्रिक्स है:

$$\begin{bmatrix} 0.08 & 0.92 \\ 0.94 & 0.06 \\ 0.10 & 0.96 \end{bmatrix}$$

इसी तरह, बाकी पैटर्न के लिए सीखना। $s_3 = [1,1,1]$ **और** $s_4 = [1,0,0]$ **चरण** $2-6$ **को दोहराकर निष्पादित किया जाता है जिसे तालिका** 8.1 **में दर्शाया गया है।**

तालिका 8.1 **पैटर्न के लिए सीखने की गणना** s_1, s_2, s_3 **और** s_4

इनपुट पैटर्न	यूक्लिडियन दूरी		विजेता	अद्यतन वजन	
	D(1)	D(2)		C_1	C_2
$s_1 = [1,0,1]$	1.048	0.538	C_2	-------	(0.92, 0.06, 0.96)
$s_2 = [0,1,0]$	0.707	1.620	C_1	(0.08, 0.94, 0.10)	-------

$s_3 = [1, 1, 1]$	1.288	0.944	C_2	-------	(0.98, 0.81, 0.99)
$s_3 = [1, 0, 0]$	1.318	1.279	C_2	-------	(0.99, 0.16, 0.19)

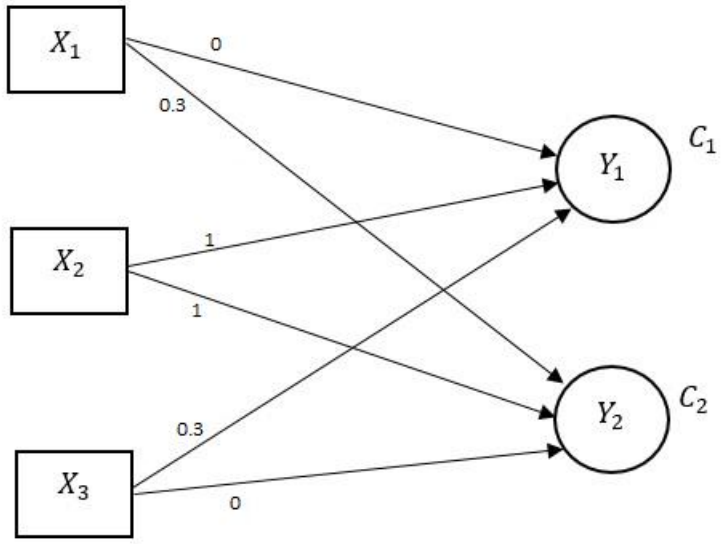
इस प्रकार, पहली पुनरावृत्ति के बाद अद्यतन वजन मैट्रिक्स है:

$$\begin{bmatrix} 0.08 & 0.99 \\ 0.94 & 0.16 \\ 0.10 & 0.19 \end{bmatrix}$$

सीखने का एल्गोरिदम 21 **पुनरावृत्तियों के बाद अभिसरण होता है जब वजन मैट्रिक्स बन जाता है**:

$$\begin{bmatrix} 0 & 0.3 \\ 1 & 1 \\ 0.3 & 0 \end{bmatrix}$$

अंजीर। 8.6 **दी गई समस्या के लिए स्व-आयोजन मानचित्र को दर्शाता है**:



चित्र 8.6. **अंतिम एसओएम नेटवर्क**

8.7 अनुकूली न्यूरो-फजी अनुमान प्रणाली (एएनएफआईएस)

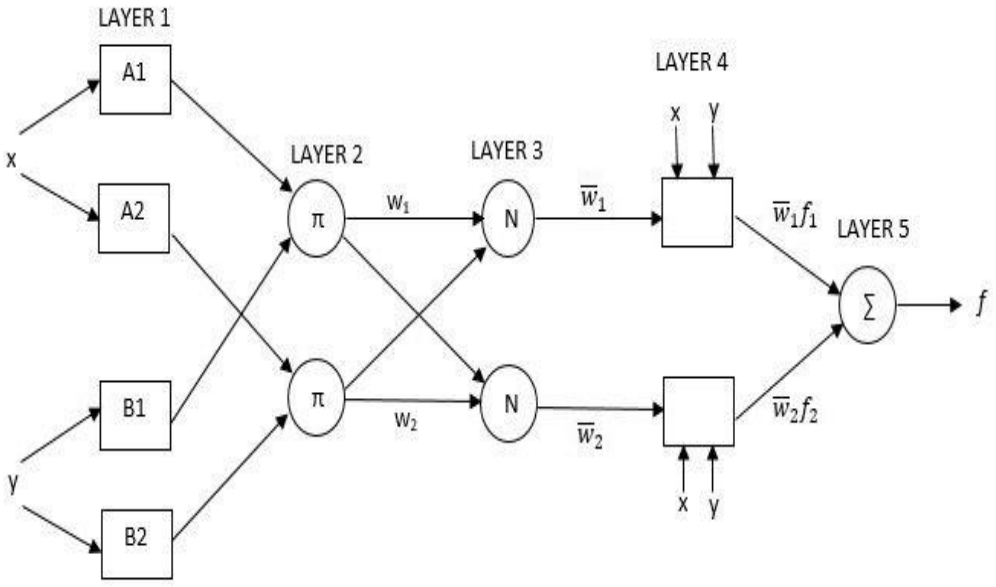
एडेप्टिव न्यूरो-फजी इनफेरेंस सिस्टम (एएनएफआईएस) हाइब्रिड एडेप्टिव न्यूरल नेटवर्क की एक श्रेणी है जो फजी इनफेरेंस सिस्टम के समान ही कार्य करता है। एडेप्टिव न्यूरो फजी इनफेरेंस सिस्टम (एएनएफआईएस) एक ऐसी तकनीक है जो सटीकता के किसी भी ग्रेड के लिए किसी भी वास्तविक कार्य का अनुमान लगाने में सक्षम है। एक फजी अनुमान प्रणाली एक नियम आधारित प्रणाली है जो तार्किक अनुमान प्रणाली के माध्यम से क्रिस्प मूल्यों के फ्यूज़िफिकेशन और डिफ्यूज़िफिकेशन को नियोजित करती है, इस प्रकार जटिल गणितीय संबंधों के उपयोग के बिना अनिश्चितता का प्रतिनिधित्व करने की क्षमता

रखती है। आर्टिफिशियल न्यूरल नेटवर्क एक प्रक्रिया में शामिल मापदंडों के बीच जटिल गैर-रेखीय संबंधों को सीखने में सक्षम हैं, लेकिन वास्तविक दुनिया की स्थितियों में काम करने में सक्षम नहीं हैं, जिसमें अनिश्चितता की एक निश्चित डिग्री शामिल है और पैरामीटर प्रकृति में अस्पष्ट हैं। इस प्रकार, फजी अनुमान प्रणाली और आर्टिफीसियल तंत्रिका नेटवर्क के संयोजन पर, एक नया हाइब्रिड दृष्टिकोण बनता है जिसे एएनएफआईएस कहा जाता है। साहित्य में दो प्रकार की फजी प्रणालियां हैं: सुजेनो ताकागी अनुमान प्रणाली और ममदानी अनुमान प्रणाली। इस पेपर में, मॉडलिंग के लिए पहले क्रम के सुजेनो ताकागी फजी मॉडल का उपयोग किया जाता है जो निम्नानुसार विस्तृत है। दो इनपुट x और y और एक आउटपुट z के लिए पहले क्रम के Sugeno Takagi फजी मॉडल को इस प्रकार व्यक्त किया जा सकता है:

$$Rule\ 1: If\ x\ is\ A_1\ and\ y\ is\ B_1\ then\ z = f_1(x,y) = p_1x + q_1y + r_1$$

$$Rule\ 2: If\ x\ is\ A_2\ and\ y\ is\ B_2\ then\ z = f_2(x,y) = p_2x + q_2y + r_2$$

कहां p_1, q_1, r_1 और p_2, q_2, r_2 पहले क्रम के सुगेनो ताकागी फजी मॉडल के परिणामी भाग में पैरामीटर हैं। ANFIS मॉडल की संरचना चित्र 8.7 में दिखाई गई है जो दो नियमों के साथ दो इनपुट प्रथम-क्रम सुजेनो ताकागी फजी मॉडल लेता है। एएनएफआईएस मॉडल में पांच परतें होती हैं, जिनमें से प्रत्येक परत को एक स्पष्ट जिम्मेदारी सौंपी जाती है। इन परतों को नीचे वर्णित किया गया है:



चित्र 8.7 दो इनपुट प्रथम-क्रम सुजेनो ताकागी फजी मॉडल के लिए दो नियमों के साथ एएनएफआईएस आर्किटेक्चर।

परत 1: प्रत्येक नोड इस परत में एक अनुकूली नोड है जिसमें एक नोड फ़ंक्शन है जैसा कि नीचे वर्णित है:

$$O_{1,i} = \mu_{A_i}(x), \quad for\ i = 1,2$$
$$O_{1,i} = \mu_{B_{i-2}}(y), \quad for\ i = 3,4$$

यहाँ x और y नोड में इनपुट हैं i और A_i और B_{i-2} इस नोड के साथ जुड़े भाषाई लेबल हैं। इसलिए $O_{1,i}$ एक फजी सेट की सदस्यता ग्रेड है A_1, A_2, B_1 नहीं तो B_2 गॉसियन, बेल, त्रिकोणीय या ट्रेपोज़ॉइडल जैसे सदस्यता समारोह के आकार की विशेषता।

परत 2: इस परत में कई नोड्स होते हैं, जिनमें से प्रत्येक को प्रोड लेबल किया जाता है और इसमें आने वाले सभी इनपुट के उत्पाद को इसके आउटपुट के रूप में उत्पादित किया जाता है।

$$O_{2,i} = \mu_{A_i}(x) \,.\, \mu_{B_i}(y), \quad for\ i = 1,2$$

इनमें से प्रत्येक नोड्स से आउटपुट संबंधित नियम की फायरिंग ताकत का प्रतिनिधित्व करता है।

परत 3: इस परत में नोड्स निश्चित नोड्स हैं जिन्हें नॉर्म और i^{th} इस परत का नोड किसके बीच के अनुपात की गणना करता है? i^{th} नियम की फायरिंग ताकत और सभी नियमों की फायरिंग ताकत का योग।

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \quad for\ i = 1,2$$

इन नोड्स के आउटपुट को सामान्यीकृत फायरिंग ताकत के रूप में जाना जाता है।

परत 4: नोड फ़ंक्शन का i^{th} नोड जो इस परत में मॉडल आउटपुट में आईटीएच नियम के योगदान की गणना करता है:

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}\,(p_i x + q_i y + r_i)$$

कहां w_i परत 3 और { का सामान्यीकृत आउटपुट है।$p_i, q_i, r_i\}$ को परिणामी पैरामीटर कहा जाता है, वे किसके पैरामीटर हैं? i^{th} गांठ।

परत 5: यह एकल नोड परत आने वाले इनपुट के योग द्वारा नेटवर्क के आउटपुट की गणना करती है और इसे इस प्रकार व्यक्त किया जाता है:

$$O_{5,i} = \sum \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

एएनएफआईएस को कम से कम वर्गों और ढाल वंश विधियों से बना एक हाइब्रिड लर्निंग एल्गोरिदम का उपयोग करके प्रशिक्षित किया जाता है। फॉरवर्ड पास के दौरान परत 4 में परिणामी मापदंडों को समझने के लिए न्यूनतम वर्ग विधि का उपयोग किया जाता है। बैकवर्ड पास के दौरान त्रुटियों को पीछे की ओर प्रचारित किया जाता है और ग्रेडिएंट डिसेंट विधि का उपयोग करके आधार मापदंडों को अपडेट किया जाता है।

8.8 सारांश

बैकप्रोपेगेशन एल्गोरिदम एक पर्यवेक्षित शिक्षण विधि है जो एक बहु स्तरीय फीड फॉरवर्ड न्यूरल नेटवर्क को प्रशिक्षित करने के लिए नियोजित है। यह एल्गोरिथ्म बार-बार प्रशिक्षण इनपुट के जवाब में लक्ष्य आउटपुट से त्रुटि या विचलन के आधार पर तंत्रिका नेटवर्क परतों के इंटरकनेक्शन भार को समायोजित करता है। प्रशिक्षण कई पुनरावृत्तियों या युगों के माध्यम से होता है और प्रत्येक युग के दौरान प्रशिक्षण इनपुट इनपुट को इनपुट परत को दिया जाता है और सिग्नल इनपुट से छिपी परतों के माध्यम से आउटपुट परत तक प्रवाहित होते हैं। त्रुटियों की गणना आउटपुट परत पर शुरू होती है और त्रुटियों को आउटपुट परत से इनपुट परत की ओर पीछे की ओर प्रचारित किया जाता है।

एडेप्टिव लीनियर न्यूरॉन (एडीएलाइन) एक एकल आउटपुट यूनिट न्यूरल नेटवर्क है जिसमें कई इनपुट यूनिट होती हैं जिसमें एक इनपुट यूनिट पूर्वाग्रह के रूप में कार्य करती है और इसका मूल्य स्थायी रूप से 1 पर तय होता है। यह दो वर्गों में से किसी एक में इनपुट पैटर्न देने के द्विआधारी वर्गीकरण के लिए नियोजित है। एडालाइन तंत्रिका नेटवर्क को विड्रो-हॉफ सीखने के नियम का उपयोग करके प्रशिक्षित किया जाता है जिसका उद्देश्य लक्ष्य मूल्य और तंत्रिका नेटवर्क द्वारा अनुमानित मूल्य के बीच औसत वर्ग त्रुटि को कम करना है। यह प्रशिक्षण के दौरान आउटपुट इकाई पर पहचान सक्रियण फ़ंक्शन का उपयोग करता है। दूसरी ओर यह एप्लिकेशन के दौरान आउटपुट यूनिट के सक्रियण फ़ंक्शन के रूप में बाइपोलर स्टेप फ़ंक्शन का उपयोग करता है।

एसोसिएटिव मेमोरी न्यूरल नेटवर्क तंत्रिका नेटवर्क का एक उप-समूह है जिसका उपयोग वजन के रूप में उनकी मेमोरी में पैटर्न संघों के सेट को स्टोर करने के लिए किया जाता है। एक एसोसिएटिव मेमोरी न्यूरल नेटवर्क या तो एक फीडफॉरवर्ड न्यूरल नेटवर्क या आवर्तक तंत्रिका नेटवर्क हो सकता है। इस अध्याय में तीन प्रकार के सहयोगी तंत्रिका नेटवर्क का वर्णन किया गया है, अर्थात् ऑटो-एसोसिएटिव, हॉपफील्ड नेटवर्क और द्विदिश एसोसिएटिव मेमोरी। ऑटो-एसोसिएटिव में एक सरल फीडफॉरवर्ड न्यूरल नेटवर्क आर्किटेक्चर होता है जबकि हॉपफील्ड नेटवर्क और द्विदिश एसोसिएटिव मेमोरी एक आवर्तक तंत्रिका नेटवर्क आर्किटेक्चर का दावा करते हैं। ऑटो-एसोसिएटिव न्यूरल नेटवर्क में, इनपुट और आउटपुट पैटर्न समान हैं। नेटवर्क एक इनपुट पैटर्न दिए गए एक ही पैटर्न को लौटाता है यदि यह संग्रहीत पैटर्न में से एक से मेल खाता है। एक ऑटो-एसोसिएटिव न्यूरल नेटवर्क को हेब लर्निंग नियम को नियोजित करके प्रशिक्षित किया जाता है। जॉन हॉपफील्ड के नाम पर हॉपफील्ड तंत्रिका नेटवर्क आवर्तक तंत्रिका

नेटवर्क वास्तुकला का उपयोग करने वाले ऑटो-एसोसिएटिव न्यूरल नेटवर्क का सबसे पहला कार्यान्वयन है। एक हॉपफील्ड तंत्रिका नेटवर्क में, प्रत्येक नोड पूरी तरह से सममित भार के साथ हर दूसरे नोड से जुड़ा होता है और इसमें कोई स्व-लूप नहीं होता है। इस प्रकार, प्रत्येक नोड को दिए गए इनपुट पैटर्न के संबंधित घटक के साथ तंत्रिका नेटवर्क में अन्य सभी नोड्स से अपना इनपुट सक्रियण मिलता है। हेब लर्निंग नियम को नियोजित करके एक हॉपफील्ड तंत्रिका नेटवर्क को भी प्रशिक्षित किया जाता है। एक द्विदिश एसोसिएटिव मेमोरी (बीएएम) एक हेटेरो एसोसिएटिव न्यूरल नेटवर्क है जो इकाइयों की दो परतों से बना होता है जो दिशात्मक लिंक से जुड़े होते हैं जो बीएएम को एक पैटर्न जोड़ी को स्टोर करने में मदद करता है। एक बीएएम तंत्रिका नेटवर्क को एक पैटर्न जोड़ी को स्टोर करने के लिए नियोजित किया जा सकता है जैसे कि जब भी इसे इनपुट के रूप में किसी भी पैटर्न के साथ प्रदान किया जाता है तो यह दूसरे को आउटपुट के रूप में प्रदान करता है।

सेल्फ ऑर्गनाइजिंग मैप (एसओएम) तंत्रिका नेटवर्क का एक उप-समूह है जिसका उपयोग क्लस्टरिंग पैटर्न के उद्देश्य के लिए विजेता की असुरक्षित सीखने की रणनीति को नियोजित करके किया जाता है। एसओएम इनपुट पैटर्न को व्यवस्थित करता है जो एक ही क्लस्टर समूह में शारीरिक रूप से एक साथ करीब होते हैं। एक एसओएम नेटवर्क जिसका उद्देश्य 'एम' के पैटर्न को 'एन' क्लस्टर संख्या में क्लस्टर करना है, उसमें इनपुट इकाइयों की 'एम' संख्या और आउटपुट इकाइयों की 'एन' संख्या होनी चाहिए। सीखने की प्रक्रिया के दौरान, एसओएम खुद को इस तरह से व्यवस्थित करता है कि जो पैटर्न काफी करीब हैं, उन्हें एक ही क्लस्टर में वर्गीकृत किया जाता है।

एडेप्टिव न्यूरो-फजी इनफेरेंस सिस्टम (एएनएफआईएस) हाइब्रिड एडेप्टिव न्यूरल नेटवर्क का एक वर्ग है जो फजी इनफेरेंस सिस्टम के समान कार्य कर सकता है। पहले क्रम के सुगेनो ताकागी फजी मॉडल पर तैयार किए गए एएनएफआईएस आर्किटेक्चर में पांच परतें होती हैं, जिनमें प्रत्येक परत की एक विशिष्ट जिम्मेदारी होती है।

प्रमुख शब्द

- आर्टिफीसियल तंत्रिका नेटवर्क
- बहुस्तरीय फ़ीड फॉरवर्ड न्यूरल नेटवर्क
- बैकप्रोपेगेशन एल्गोरिथ्म
- ढाल वंश
- वजन
- वजन मैट्रिक्स
- वर्गीकरण
- छिपी हुई परतें
- सक्रियण फ़ंक्शन
- पहचान समारोह
- पर्यवेक्षित शिक्षा

- असुरक्षित शिक्षा।
- हेब सीखने का नियम
- विड्रो-हॉफ सीखने का नियम
- विजेता सभी सीखने के नियम लेता है
- अनुकूली रैखिक न्यूरॉन (एडालिन)
- एसोसिएटिव मेमोरी न्यूरल नेटवर्क
- पैटर्न एसोसिएशन
- पैटर्न वर्गीकरण
- आवर्तक तंत्रिका नेटवर्क
- ऑटो-एसोसिएटिव न्यूरल नेटवर्क
- हॉपफील्ड नेटवर्क
- द्विदिश साहचर्य स्मृति (बीएएम)
- हेटेरो एसोसिएटिव न्यूरल नेटवर्क
- क्लस्टरिंग
- स्व-आयोजन मानचित्र (एसओएम)
- अनुकूली न्यूरो-फजी अनुमान प्रणाली (एएनएफआईएस)
- हाइब्रिड अनुकूली न्यूरल नेटवर्क

अभ्यास

1. बहुत छोटे प्रकार के प्रश्न (**2** अंक)**:**

a) हेब्बियन सीखने के कानून को बताएं?
b) कैसे एडालिन न्यूरॉन मॉडल मैककुलोच पिट के न्यूरॉन मॉडल से अलग है।
c) बैक प्रोपेगेशन नेटवर्क के किन्हीं दो अनुप्रयोगों को सूचीबद्ध करें।
d) बैक प्रोपेगेशन नेटवर्क के अवगुणों का उल्लेख कीजिए।
e) बैक प्रोपेगेशन नेटवर्क के गुणों का उल्लेख करें। .
f) एसोसिएटिव मेमोरी क्या है?
g) एसोसिएटिव मेमोरी के दो प्रकारों के नाम बताइए।
h) द्विदिश साहचर्य स्मृति के दो प्रकारों के नाम बताइए?
i) पैटर्न एसोसिएशन समस्या क्या है?
j) हॉपफील्ड मेमोरी क्या है?
k) पैटर्न एसोसिएशन नेटवर्क के लिए विकसित दो एल्गोरिदम का नाम बताइए।
l) प्रतिस्पर्धी तंत्रिका नेटवर्क क्या हैं?
m) स्व-संगठन क्या है?
n) Kohonen के SOFM की संरचना दिखाएँ।
o) एसओएफएम एल्गोरिथ्म का सारांश प्रदान करें?
p) Kohonen के SOFM का फ़्लोचार्ट लिखिए
q) अनुकूली अनुनाद सिद्धांत आधारित तंत्रिका नेटवर्क का लाभ दें।
r) अनुकूली अनुनाद सिद्धांत आधारित तंत्रिका नेटवर्क के अनुप्रयोगों का उल्लेख करें?

s) निम्नलिखित कथन सत्य है या असत्य? "ज्ञान एक सीखने की प्रक्रिया के माध्यम से अपने पर्यावरण से एक तंत्रिका नेटवर्क द्वारा प्राप्त किया जाता है, और यह ज्ञान प्रसंस्करण इकाइयों (वजन) के बीच कनेक्शन ताकत (न्यूरॉन्स) में संग्रहीत होता है। [

t) डेल्टा सीखने के नियम को परिभाषित करें?

2. लघु प्रकार के प्रश्न (**5** अंक)**:**

a) एआरटी नेटवर्क पर संक्षिप्त नोट लिखें?

b) कोहोनेन के स्व-आयोजन अधिगम एल्गोरिथ्म की विभिन्न विशेषताएं लिखिए।

c) हॉपफील्ड नेटवर्क की वास्तुकला का वर्णन करें?

d) ऑटो एसोसिएटिव मेमोरी और हेटेरो एसोसिएटिव मेमोरी के बीच अंतर क्या है?

e) द्वि-दिशात्मक मेमोरी (बीएएम) क्या है? इसकी वास्तुकला को संक्षेप में समझाइए?

f) हॉपफील्ड मेमोरी के लिए प्रशिक्षण एल्गोरिथ्म प्रदान करें।

g) हेब नियम का उपयोग करके बुनियादी पैटर्न एसोसिएशन समस्या के लिए चरण-दर-चरण प्रशिक्षण एल्गोरिथ्म दिखाएं?

h) एआरटी नेटवर्क के प्रशिक्षण एल्गोरिथ्म की व्याख्या करें।

i) एक साफ स्केच के साथ कोहोनेन के स्व-आयोजन सुविधा मानचित्र (एसओएम) एल्गोरिदम के संचालन की व्याख्या करें और बताएं कि यह किस प्रकार की समस्याओं के लिए सबसे उपयुक्त है।

3. लंबे प्रकार के प्रश्न (**10** अंक)**:**

a) मल्टीलेयर फीड फॉरवर्ड न्यूरल नेटवर्क के लिए एक बैक प्रोपेगेशन एल्गोरिदम विकसित करें जिसमें पहले सिद्धांतों से एक इनपुट परत, एक छिपी हुई परत और आउटपुट परत शामिल है।

b) हेब नियम का उपयोग करके बुनियादी पैटर्न एसोसिएशन समस्या के लिए चरण-दर-चरण प्रशिक्षण एल्गोरिथ्म दिखाएँ .

c) असतत बीएएम की चरण-दर-चरण प्रशिक्षण और परीक्षण प्रक्रिया दें।

d) एक हेटेरो एसोसिएटिव नेटवर्क दिया गया है। वजन मैट्रिक्स का पता लगाएं और प्रशिक्षण इनपुट वैक्टर के साथ नेटवर्क का परीक्षण करें। S1 = (1,1,0,0), S2 = (0,1,0,0), S3 = (0,0,1,1), S4 = (0,0,1,0) t1 = (1,0), t2 = (1,0), t3 = (0,1), t4 = (0,1)

e) एक हेटेरो एसोसिएटिव नेटवर्क को इनपुट वेक्टर एस = [x1, x2, x3, x4] के लिए हेब बाहरी उत्पाद नियम द्वारा प्रशिक्षित किया जाता है ताकि आउटपुट पंक्ति वैक्टर t = [t1,t2] हो सके। वजन मैट्रिक्स ज्ञात कीजिये: S1 = (1,1,0,0), S2 = (1,1,1,0), S3 = (0,0,1,1), S4 = (0,1,0,0)t1 = (1,0), t2 = (0.1), t3 = (1,0), t4 = (1,0)

f) हॉपफील्ड मेमोरी के लिए प्रशिक्षण एल्गोरिदम प्रदान करें?

g) एडलिन लर्निंग एल्गोरिथ्म को विस्तार से समझाएं?

h) एक हॉपफील्ड तंत्रिका नेटवर्क में एक पैटर्न एस = [1,0,1,1] को स्टोर और पहचानें?

i) लोकप्रिय स्व-आयोजन मानचित्रों के आर्किटेक्चर की व्याख्या करें। Kohonen नेटवर्क के प्रशिक्षण एल्गोरिथ्म प्राप्त करें। यह भी बताएं कि डेटा संपीड़न के लिए एसओएम का उपयोग कैसे किया जा सकता है।

j) दो क्लस्टर इकाइयों और पांच इनपुट इकाइयों के साथ एक कोहोनेन नेटवर्क पर विचार करें। क्लस्टर इकाइयों के लिए वजन वैक्टर w1 = [0.1, 0.3, 0.5, 0.7, 0.9] और w2 = [0.9, 0.7, 0.5, 0.3, 0.1] इनपुट पैटर्न के लिए विजेता क्लस्टर इकाई खोजने के लिए यूक्लिडियन दूरी के वर्ग का उपयोग करें।

ग्रंथ-सूची

- चेरकास्की और एन वासिलास, "एसोसिएटिव डेटाबेस पुनर्प्राप्ति के लिए बैक-प्रोपेगेशन नेटवर्क का प्रदर्शन"। न्यूरल नेटवर्क पर अंतर्राष्ट्रीय संयुक्त सम्मेलन की कार्यवाही, वाशिंगटन, डीसी, पीपी आई-77-84, 1989।
- हेच-निक्लसन, "बैकप्रोपेगेशन न्यूरल नेटवर्क का सिद्धांत"। न्यूरल नेटवर्क पर अंतर्राष्ट्रीय संयुक्त सम्मेलन, वाशिंगटन, डीसी, पीपी ।-593-605, 1989।
- रूमेलहार्ट, जीई हिंटन और आरजे विलियम्स, "बैक-प्रचार त्रुटि द्वारा सीखने का प्रतिनिधित्व", प्रकृति, वॉल्यूम 323, 533-536, 1986।
- विड्रो और एसडी स्टर्न्स, "एडेप्टिव सिग्नल प्रोसेसिंग", एंगलवुड क्लिफ्स, एनजे: प्रेंटिस-हॉल, 1985।
- हेब, "व्यवहार का संगठन", न्यूयॉर्क: जॉन विली एंड संस, 1949।
- होपफील्ड, "उभरती सामूहिक कम्प्यूटेशनल क्षमताओं के साथ तंत्रिका नेटवर्क और भौतिक प्रणालियां", संयुक्त राज्य अमेरिका के नेशनल एकेडमी ऑफ साइंसेज की कार्यवाही, वॉल्यूम 79, नंबर 8, पीपी 2554-2558, 1982।
- कोहोनेन, "स्व-संगठन और सहयोगी स्मृति", बर्लिन" स्प्रिंगर-वर्लग, 1989।
- कोस्को, "द्विदिश साहचर्य यादें", न्यूरल नेटवर्क पर आईईईई लेनदेन, वॉल्यूम 1, नंबर 1, पीपी 44-57, 1988।
- कोहोनेन, "स्व-आयोजन मानचित्र"। स्प्रिंगर, बर्लिन, 1997।
- एस. आर. जंग, सी.टी. मिजुतानी, "न्यूरो-फजी और सॉफ्ट कंप्यूटिंग; सीखने और मशीन इंटेलिजेंस के लिए एक कम्प्यूटेशनल दृष्टिकोण", 1997।
- रॉय, एस एंड चक्रवर्ती, यू (2013)। *सॉफ्ट कंप्यूटिंग का परिचय: न्यूरो-फजी और जेनेटिक एल्गोरिदम.* पियर्सन।
- पाढ़ी, एनपी और साइमन, एसपी (2015)। *MATLAB प्रोग्रामिंग के साथ सॉफ्ट कंप्यूटिंग.* ऑक्सफोर्ड यूनिवर्सिटी प्रेस।

एआईसीटीई पाठ्यक्रम

आर्टिफीसियल इंटेलिजेंस

उद्देश्य:आर्टिफिशियल इंटेलिजेंस की अवधारणाओं और आर्टिफिशियल इंटेलिजेंस का उपयोग करके समस्याओं को हल करने के तरीकों को सीखने के लिए।	
इकाई	विषय
I	परिचय:एआई की अवधारणा, इतिहास, वर्तमान स्थिति, दायरा, एजेंट, वातावरण, समस्या सूत्रीकरण, पेड़ और ग्राफ संरचनाओं की समीक्षा, राज्य अंतरिक्ष प्रतिनिधित्व, खोज ग्राफ और खोज पेड़।
II	खोज एल्गोरिदम: यादृच्छिक खोज, बंद और खुली सूची के साथ खोज, गहराई पहली और चौड़ाई पहली खोज, हेरिस्टिक खोज, सर्वश्रेष्ठ पहली खोज, ए * एल्गोरिदम, गेम खोज।
III	संभाव्य तर्क:संभाव्यता, सशर्त संभाव्यता, बेयस नियम, बायेसियन नेटवर्क-प्रतिनिधित्व, निर्माण और अनुमान, लौकिक मॉडल, छिपे हुए मार्कोव मॉडल।
IV	मार्कोव निर्णय प्रक्रिया:एमडीपी सूत्रीकरण, उपयोगिता सिद्धांत, उपयोगिता कार्य, मूल्य पुनरावृत्ति, नीति पुनरावृत्ति और आंशिक रूप से अवलोकन योग्य एमडीपी।
V	सुदृढीकरण सीखना:निष्क्रिय सुदृढीकरण सीखने, प्रत्यक्ष उपयोगिता आकलन, अनुकूली गतिशील प्रोग्रामिंग, अस्थायी अंतर सीखने, सक्रिय सुदृढीकरण सीखने- क्यू सीखने।

सुझाई गई पुस्तकों की सूची

1. स्टुअर्ट रसेल और पीटर नॉर्विग, "आर्टिफिशियल इंटेलिजेंस: एक आधुनिक दृष्टिकोण", तीसरा संस्करण, प्रेंटिस हॉल
2. ऐलेन रिच और केविन नाइट, "आर्टिफिशियल इंटेलिजेंस", टाटा मैकग्रा हिल
3. त्रिवेदी, एम.सी., "ए क्लासिकल अप्रोच टू आर्टिफिकल इंटेलिजेंस", खन्ना पब्लिशिंग हाउस, दिल्ली।
4. सरोज कौशिक, "आर्टिफिशियल इंटेलिजेंस", चेंज लर्निंग इंडिया, 2011
5. डेविड पूल और एलन मैकवर्थ, "आर्टिफिशियल इंटेलिजेंस: कम्प्यूटेशनल एजेंटों के लिए नींव", कैम्ब्रिज यूनिवर्सिटी प्रेस 2010।

अब्दुल कलाम टेक्निकल यूनिवर्सिटी (एकेटीयू), लखनऊ
सिलेबस

आर्टिफिशियल इंटेलिजेंस **(KCS071)**		
पाठ्यक्रम परिणाम **(CO)**		ब्लूम का ज्ञान स्तर (केएल)
पाठ्यक्रम के अंत में, छात्र समझने में सक्षम हो जाएगा।		
सीओ 1	आर्टिफिशियल इंटेलिजेंस के सिद्धांत और अभ्यास के मूल सिद्धांतों को एक अनुशासन के रूप में और इंटेलिजेंसमान एजेंटों के बारे में समझें।	K_2
सीओ 2	खोज तकनीक और गेमिंग सिद्धांत को समझें।	K_2, K_3
सीओ 3	छात्र ज्ञान प्रतिनिधित्व तकनीकों और समस्या को हल करने के लिए लागू करना सीखेंगे। सामान्य एआई अनुप्रयोगों के लिए रणनीतियाँ।	K_3 , K_4
सीओ 4	छात्र को वर्गीकरण और क्लस्टरिंग के लिए उपयोग की जाने वाली तकनीकों के बारे में पता होना चाहिए।	K_2 , K_3
सीओ 5	छात्र को पैटर्न मान्यता की मूल बातें और इसके लिए आवश्यक कदमों के बारे में पता होना चाहिए।	K_2 , K_4
विस्तृत पाठ्यक्रम		**3-0-0**
इकाई	विषय	प्रस्तावित पढ़ना
मैं	परिचय: परिचय - परिभाषा - आर्टिफिशियल इंटेलिजेंस का भविष्य - इंटेलिजेंसमान एजेंटों की विशेषताएं - विशिष्ट इंटेलिजेंसमान एजेंट - विशिष्ट एआई समस्याओं के लिए समस्या समाधान दृष्टिकोण।	**08**
द्वितीय	समस्या को हल करने के तरीके: समस्या सुलझाने के तरीके - खोज रणनीतियाँ - अनसूचित - सूचित - हेरिस्टिक्स - स्थानीय खोज एल्गोरिदम और अनुकूलन समस्याएं - आंशिक टिप्पणियों के साथ खोज - बाधा संतुष्टि समस्याएं - बाधा प्रसार - बैकट्रैकिंग खोज - गेम खेलना - खेलों में इष्टतम निर्णय - अल्फा - बीटा छंटाई - स्टोकेस्टिक गेम	**08**
III	ज्ञान प्रतिनिधित्व: पहला क्रम विधेय तर्क - प्रोलॉग प्रोग्रामिंग - एकीकरण - फॉरवर्ड चेनिंग-बैकवर्ड चेनिंग - रिज़ॉल्यूशन - ज्ञान प्रतिनिधित्व - ऑन्कोलॉजिकल इंजीनियरिंग-श्रेणियां और वस्तुएं - घटनाएं - मानसिक घटनाएं और मानसिक वस्तुएं - श्रेणियों के लिए तर्क प्रणाली - डिफ़ॉल्ट जानकारी के साथ तर्क	**08**
IV	सॉफ्टवेयर एजेंट: इंटेलिजेंसमान एजेंटों के लिए वास्तुकला - एजेंट संचार - बातचीत और सौदेबाजी - एजेंटों के बीच तर्क - मल्टी-एजेंट सिस्टम में विश्वास और प्रतिष्ठा।	**08**
में	अनुप्रयोग: एआई अनुप्रयोग - भाषा मॉडल - सूचना पुनर्प्राप्ति - सूचना निष्कर्षण - प्राकृतिक भाषा प्रसंस्करण - मशीन अनुवाद - भाषण पहचान - रोबोट - हार्डवेयर - धारणा - योजना - आगे बढ़ना	**08**

पाठ्य पुस्तकें:

1. एस. रसेल विज्ञापन पी.नामिंग, "आर्टिफिशिया।

2. ब्रैटको, "प्रोलॉग: प्रोग्रामिंग फॉर आर्टिफिशियल इंटेलिजेंस", चौथा संस्करण, एडिसन-वेस्ले एजुकेशनल पब्लिशर्स इंक., 2011।

3. एम. टी.एम. जोन्स, -आर्टिफिशियल इन ऑर्डरजी एंड एनथैट: ए सिस्टम्स ए पी पीआर या एन या एच (सी या एन या एमपी बी एच आर एन दैट)॥, जे एट एंड ए एम जर्क बार्टल एंड टी पब्लिश एच लीडर्स इंक. प्रथम संस्करण, 2008

4. ज़ीरोस जे. निल्सन, "टी दैट क्यू इन इट टीएफ इफ नॉट आर्टिफिशियल इन एनटी एली जी एंड एनसी एंड", सी ए सीएमबीआर इंडगे अन इन एंड आर वाई प्रेस, 2009।

5. इलिया;एमएफ.व्हीक्सिन एड क्राइस्ट नॉट पीएच एंडआरएस.मेलिश, ॥पीआर नॉट जीआर एएमएम आई का एन प्रोलैंड: वी आई काथेआईएसओएसटी ए टीके एआरडी ॥, फिफ्टी श एंड डी नॉट यू एन, स्प्रिंगर, 2003।

6. डेनार्ड वीस, —एमएलटी इंग्मेंट सिस्टम एंडएमएस॥, सेकंड एनडी और डी इट्स टू.एन, एम प्रेस, 2013।

7. डेव और एल.पोल्गिनकलन के.मैक,—आर्टिफिशियल ऑर्डर गैंडदैट:फॉन्डेशन्सऑफकनॉटकम्प्यूटेशनअलोनगैंडेंट्स॥, एम्ब्रिज यूनिवर्सिटी प्रेस, 2010।

लखनऊ विश्वविद्यालय का पाठ्यक्रम

आर्टिफीसियल इंटेलिजेंस (CS-703) L T P 3 1 0

यूनिट-1

परिचय: आर्टिफीसियल इंटेलिजेंस का परिचय, आर्टिफीसियल इंटेलिजेंस की नींव और इतिहास, आर्टिफीसियल इंटेलिजेंस के अनुप्रयोग, इंटेलिजेंसमान एजेंट, इंटेलिजेंसमान एजेंटों की संरचना, कंप्यूटर दृष्टि और प्राकृतिक भाषा रखने वाले।

यूनिट-2

खोज का परिचय: समाधान, वर्दीधारी खोज रणनीतियों, सूचित खोज रणनीतियों, स्थानीय खोज एल्गोरिदम और आशावादी समस्याओं, प्रतिकूल खोज, गेम की खोज और अल्फा - बीटा छंटाई की खोज।

यूनिट-3

ज्ञान प्रतिनिधित्व और तर्क: प्रस्तावात्मक तर्क, प्रथम क्रम तर्क का सिद्धांत, पहले क्रम तर्क में अनुमान, आगे और पीछे की श्रृंखला, संकल्प, संभाव्य तर्क, उपयोगिता सिद्धांत, छिपे हुए मार्कोव मॉडल (एचएमएम) और बायेसियन नेटवर्क।

यूनिट-4

मशीन लर्निंग: पर्यवेक्षित और असुरक्षित सीखने, निर्णय पेड़, सांख्यिकीय सीखने के मॉडल, पूर्ण डेटा के साथ सीखना - भोले बेयस मॉडल और छिपे हुए डेटा के साथ सीखना - ईएम एल्गोरिदम और सुदृढीकरण सीखना।

यूनिट-5

पैटर्न मान्यता: परिचय, पैटर्न मान्यता प्रणाली के डिजाइन सिद्धांत, सांख्यिकीय पैटर्न मान्यता, पैरामीटर आकलन विधियां - प्रमुख घटक विश्लेषण (पीसीए) और रैखिक भेदभाव विश्लेषण (एलडीए); वर्गीकरण तकनीक - निकटतम पड़ोसी (एनएन) नियम, बेयस क्लासिफायर, सपोर्ट वेक्टर मशीन (एसवीएम) और के - साधन और क्लस्टरिंग।

पाठ्य पुस्तकें:

1. स्टुअर्ट रसेल, पीटरनॉरविग, "आर्टिफिशियल इंटेलिजेंस-ए मॉडर्न अप्रोच", पियर्सन।
2. ऐलेन रिच और केविन नाइट, "आर्टिफिशियल इंटेलिजेंस", मैकग्रा-हिल।
3. ई चार्नियाक और डी मैकडरमॉट, "आर्टिफिशियल इंटेलिजेंस का परिचय", पियर्सन।

संदर्भ पुस्तकें:

1. पैटरसन, "आर्टिफिशियल इंटेलिजेंस एंड एक्सपर्ट सिस्टम", प्रेंटिस हॉल।
2. सरोज कौशिक, "आर्टिफिशियल इंटेलिजेंस", चेंज लर्निंग।
3. फिलिप सी जैक्सन"आर्टिफिशियल इंटेलिजेंस का परिचय", डोवर प्रकाशन।

डॉ शकुंतला मिश्रा राष्ट्रीय पुनर्वास विश्वविद्यालय पाठ्यक्रम

आर्टिफीसियल इंटेलिजेंस

(TCS501)

उद्देश्य:आर्टिफिशियल इंटेलिजेंस एक बड़ा कदम है कि कंप्यूटर सिस्टम कैसे अनुकूलन, विकसित और सीखता है। इसका लगभग हर उद्योग में व्यापक अनुप्रयोग है और इसे एक बड़ा तकनीकी बदलाव माना जाता है, जो औद्योगिक क्रांति, कंप्यूटर युग और स्मार्ट फोन क्रांति जैसी पिछली घटनाओं के समान है।	
इकाई	विषय
I	***परिचय***: एआई की अवधारणा, इतिहास, वर्तमान स्थिति, दायरा, एजेंट, वातावरण, समस्या सूत्रीकरण, पेड़ और ग्राफ संरचनाओं की समीक्षा, राज्य अंतरिक्ष प्रतिनिधित्व, खोज ग्राफ और खोज पेड़।
II	***खोज एल्गोरिदम:*** यादृच्छिक खोज, बंद और खुली सूची के साथ खोजें, गहराई पहले और चौड़ाई पहली खोज, हेरिस्टिक खोज, सर्वश्रेष्ठ पहली खोज, ए * एल्गोरिदम, गेम खोज।
III	***संभाव्य तर्क:*** संभाव्यता, सशर्त संभाव्यता, बेयस नियम, बायेसियन नेटवर्क- प्रतिनिधित्व, निर्माण और अनुमान, लौकिक मॉडल, छिपे हुए मार्कोव मॉडल।
IV	***मार्कोव निर्णय:*** एमडीपी सूत्रीकरण, उपयोगिता सिद्धांत, उपयोगिता कार्य, मूल्य पुनरावृत्ति, नीति पुनरावृत्ति और आंशिक रूप से अवलोकन योग्य एमडीपी।
V	***सुदृढीकरण सीखना:*** निष्क्रिय सुदृढीकरण सीखने, प्रत्यक्ष उपयोगिता आकलन, अनुकूली गतिशील प्रोग्रामिंग, अस्थायी अंतर सीखने, सक्रिय सुदृढीकरण सीखने- क्यू सीखने।

संदर्भ:

1. शिंग रोजर जांग, चुएन: साई सन, इजी मिजुतानी, न्यूरो: फजी और सॉफ्ट कंप्यूटिंग, पीएचआई

2. क्लिर और बो युआन, फजी सेट और फजी लॉजिक: सिद्धांत और अनुप्रयोग, पीएचआई

3. जॉर्ज एफ लुगर, आर्टिफिशियल इंटेलिजेंस। जटिल समस्या के लिए संरचनाएं और रणनीतियाँ सॉल्विंग, एडिसन-वेस्ले

4. बेन कॉपियर, आर्टिफिशियल इंटेलिजेंस रोमिनेटेड, नरोसा

ख्वाजा मोइनुद्दीन चिश्ती भाषा विश्वविद्यालय पाठ्यक्रम

आर्टिफीसियल इंटेलिजेंस

(CS603)

उद्देश्य:आर्टिफिशियल इंटेलिजेंस की अवधारणाओं और आर्टिफिशियल इंटेलिजेंस का उपयोग करके समस्याओं को हल करने के तरीकों को सीखने के लिए।	
इकाई	विषय
मैं	परिचय:आर्टिफिशियल इंटेलिजेंस का परिचय, नींव और इतिहास आर्टिफिशियल इंटेलिजेंस, आर्टिफिशियल इंटेलिजेंस का अनुप्रयोग संचार - एजेंटों के बीच संचार प्राकृतिक भाषा प्रसंस्करण औपचारिक व्याकरण
द्वितीय	खोज का परिचय: समाधान की खोज, वर्दीधारी खोज रणनीतियाँ, सूचित खोज रणनीतियाँ, स्थानीय खोज एल्गोरिदम और आशावादी समस्याएं, प्रतिकूल खोज, गेम की खोज, अल्फा - बीटा छंटाई।
III	ज्ञान प्रतिनिधित्व और तर्क: प्रस्तावात्मक तर्क, प्रथम क्रम तर्क का सिद्धांत, प्रथम क्रम तर्क में अनुमान, आगे और पीछे की श्रृंखला, संकल्प, संभाव्य तर्क, उपयोगिता सिद्धांत, छिपे हुए मार्कोव मॉडल (एचएमएम), बायेसियन नेटवर्क।
IV	निर्णय लेने- उपयोगिता सिद्धांत, उपयोगिता कार्य, और निर्णय सैद्धांतिक विशेषज्ञ सिस्टम। डिफ़ॉल्ट तर्क, फजी सेट और फजी लॉजिक; एआई भाषाएं और उपकरण
मैं	पैटर्न पहचान: परिचय, पैटर्न मान्यता प्रणाली के डिजाइन सिद्धांत, सांख्यिकीय पैटर्न मान्यता, पैरामीटर आकलन विधियां - प्रमुख घटक विश्लेषण (पीसीए) और रैखिक भेदभाव विश्लेषण (एलडीए)

संदर्भ:

1. केविन नाइट और ऐलेन रिच, नायर बी., "आर्टिफिशियल इंटेलिजेंस (एसआईई)", मैकग्रा हिल, 2008.

पैटरसन, "एआई और ईएस का परिचय", पियर्सन एजुकेशन, 2007।

3. पीटर जैक्सन, "विशेषज्ञ प्रणालियों का परिचय", पियर्सन शिक्षा, 2011।

सुमन कुमार मिश्रा

डॉ सुमन कुमार मिश्रा वर्तमान में लखनऊ में ख्वाजा मोइनुद्दीन चिश्ती भाषा विश्वविद्यालय के इंजीनियरिंग और प्रौद्योगिकी संकाय में कंप्यूटर विज्ञान और इंजीनियरिंग विभाग में सहायक प्रोफेसर के रूप में कार्यरत हैं। शिक्षण और उद्योग दोनों में 17 से अधिक वर्षों के अनुभव के साथ, उनकी विशेषज्ञता का क्षेत्र आर्टिफिशियल इंटेलिजेंस (एआई), क्लाउड कंप्यूटिंग, डेटा माइनिंग और वेयरहाउसिंग, और क्लस्टर कंप्यूटिंग है। वह प्रसिद्ध अंतरराष्ट्रीय और सम्मेलन पत्रिकाओं में प्रकाशित 20 से अधिक शोध पत्रों के लेखक रहे हैं, साथ ही कई आमंत्रित वार्ताएं दे रहे हैं और कई संदर्भ पुस्तकों में 4 किताबें और अध्याय प्रकाशित कर रहे हैं। उनकी शोध गतिविधियों का मुख्य फोकस सॉफ्ट कंप्यूटिंग, मशीन लर्निंग और आर्टिफिशियल इंटेलिजेंस पर रहा है।

शोभित शुक्ला

श्री शोभित शुक्ला वर्तमान में डॉ. शकुंतला मिश्रा राष्ट्रीय पुनर्वास विश्वविद्यालय में सूचना प्रौद्योगिकी विभाग में सहायक प्रोफेसर के रूप में कार्यरत हैं। उनके पास 8 साल का कुल शिक्षण अनुभव है। उन्होंने वर्ष 2022 में लखनऊ विश्वविद्यालय, भारत से पीएचडी की डिग्री और वर्ष 2013 में मणिपाल इंस्टीट्यूट ऑफ टेक्नोलॉजी, भारत से मास्टर ऑफ टेक्नोलॉजी की उपाधि प्राप्त की है। उन्होंने प्रतिष्ठित अंतरराष्ट्रीय और राष्ट्रीय पत्रिकाओं में 12 से अधिक शोध पत्र प्रकाशित किए हैं और अंतर्राष्ट्रीय सम्मेलनों में पेपर प्रस्तुत किए हैं। उन्होंने विभिन्न आमंत्रित वार्ताएं भी दी हैं और 3 पुस्तक अध्याय प्रकाशित किए हैं। उन्होंने 2 किताबें प्रकाशित की हैं और उनके शोध के 2 कॉपीराइट हैं। उनका मुख्य शोध कार्य सॉफ्ट कंप्यूटिंग, मशीन लर्निंग और आर्टिफिशियल इंटेलिजेंस पर केंद्रित है।

अमित मिश्रा

डॉ. अमित मिश्रा वर्तमान में वाणिज्य विभाग, लखनऊ विश्वविद्यालय, लखनऊ में सहायक प्रोफेसर के रूप में कार्यरत हैं। डॉ मिश्रा को शिक्षण और उद्योग दोनों में 12 से अधिक वर्षों का अनुभव है। उनकी विशेषज्ञता का क्षेत्र आर्टिफिशियल इंटेलिजेंस, बिजनेस एनालिटिक्स और फाइनेंस है। उन्होंने प्रसिद्ध

अंतरराष्ट्रीय पत्रिकाओं, सम्मेलनों और पत्रिकाओं में 15 से अधिक शोध पत्र प्रकाशित किए हैं। इसके अलावा, डॉ मिश्रा ने विभिन्न आमंत्रित वार्ताएं दी हैं। उन्होंने प्रतिष्ठित अंतरराष्ट्रीय प्रकाशनों में 3 पुस्तकें और अध्याय भी प्रकाशित किए हैं और अपने शोध कार्य के लिए 3 पेटेंट प्राप्त किए हैं। डॉ मिश्रा के मुख्य शोध क्षेत्र आर्टिफिशियल इंटेलिजेंस, बिजनेस एनालिटिक्स, फाइनेंस और कैपिटल मार्केट हैं।

No Objection Certificate

(i) I, ***(Dr. Suman Kumar Mishra: author / Dr. Shobhit Shukla: co-author / Dr. Amit Mishra: co-author)*** represent and warrant that I am the sole owner of all copyright, trademark, and other intellectual property and proprietary rights in relation to the book/material.

(ii) I, ***(Dr. Suman Kumar Mishra: author / Dr. Shobhit Shukla: co-author / Dr. Amit Mishra: co-author)*** undertake that the Book/Material is not subject to any contract or arrangement which would conflict with my permission herein.

(iii) This is to certify that the undersigned hereby gives permission to UGC and also authorizes UGC to get it translated or published and uploaded on e-kumbh portal आर्टिफिशियल इंटेलिजेंस की अवधारणाएं और व्यावहारिक दृष्टिकोण. The book authored by the undersigned and translated using anuvadini tool has been properly vetted by me and it can be sent for publishing on e-kumbh portal.

(iv) UGC will have the full right to publish the book/text and is authorized to do any modifications, republication, or any other assistance related to the text if required.

(v) No legal action will be taken by the author in this regard.

Dr. Suman Kumar Mishra,
Assistant Professor,
Department of Computer Science and Engineering,
Khwaja Moinuddin Chishti Language University,
Lucknow.

Dr. Shobhit Shukla,
Assistant Professor,
Department of Information Technology,
Dr. Shakuntala Misra National Rehabilitation University, Lucknow.

Dr. Amit Mishra,
Assistant Professor,
Department of Commerce,
University of Lucknow,
Lucknow.

Date: 17/07/2023

Place: Lucknow