

**T.C.
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

**YAPAY SİNİR AĞLARI
(BİTİRME ÖDEVİ)**

**Süha TOZKAN
99220056**

**YÖNETEN
Yrd. Doç. Dr. Hasan H. BALIK**

**ELAZIĞ
2004**

**T.C.
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

**YAPAY SİNİR AĞLARI
(BİTİRME ÖDEVİ)**

**Süha TOZKAN
99220056**

**ELAZIĞ
2004**

TEŞEKKÜR

Proje çalışmalarım sırasında yardımlarını benden esirgemeyen ve beni yönlendiren sayın hocam Yrd. Doç. Dr. Hasan H. BALIK 'a yardımlarından dolayı teşekkürü bir borç bilirim.

Süha TOZKAN

Bu çalışma Fırat Üniversitesi Mühendislik Fakültesi Elektrik Elektronik Mühendisliği Bölümüne Bitirme Ödevi Yükümlülüğünü yerine getirmek amacıyla sunulmuştur./...../2004

Süha TOZKAN

Proje Yöneticisi

Yrd. Doç. Dr. Hasan H. BALIK

Verilen Not:.....

Jüri Üyeleri :

Proje Yöneticisi

Yrd. Doç. Dr. Hasan H. BALIK

Üye

Üye

Bölüm Başkanı

Prof. Dr. Mustafa POYRAZ

1.YAPAY ZEKA

1.1.Giriş

Literatürde “Artificial Intelligence”olarak adlandırılan yapay zeka ilk bakışta herkese farklı bir şeyin çağrışımını yaptırmaktadır. Kimilerine göre, yapay zeka kavramı, insanoğlunun yerini alan elektromekanik bir robotu çağrıştırmaktadır. Fakat bu alanla ilgili olan herkes, insanoğlu ile makinalar arasında kesin bir farklılığın olduğu bilincindedir. Bilgisayarlar hiçbir zaman insanoğlunun yaratıcılık, duygu ve mizacının benzeşimini aktarabilme becerisine sahip olamayacaktır. Bununla beraber, bilgisayarların belirli insan davranışlarını (nesneleri alma ve bunları belirli yerlere yerleştirme gibi) yapan makinalara yön vermesi ve belirli bir uzmanlık alanı ile ilgili (veri hesaplaması, tıbbi teşhis gibi) beşeri düşünme sürecinin benzetimini yapan (simüle eden) sistemlere beyin olma becerisine sahip olması mümkündür.

Yapay Zeka terimi resmen ilk defa 1950’lerin ortasında kullanılmıştır. Yapay Zeka, başka türlü otomatikleştirilemeyecek zeka süreçlerini modellemek ve makinalaştırmak için yöntemler sağlamaya gayret eder. Yapay Zeka alanında çalışmak; geleneksel bilgisayar yaklaşımlarına ilaveten; çoğunlukla sembolik, algoritmik olmayan problem çözme metodları ve bilgisayarların, sayılarla hesap yapmaktan ziyade kavramlarla muhakeme yapmak için kullanımına dayanır.

Yapay Zeka bir bilgisayar bilimi alanıdır. Görüşleri; psikoloji, bilmeye ve kavramaya ait bilim, hesaplamalı dilbilim, veri işleme, karar destek sistemleri ve hesaplamalı modelleme gibi çeşitli diğer disiplinlere yakından bağlıdır ve tüm bu disiplinleri kaynak olarak kullanır.

1.2.Tanım

Yapay Zeka Ansiklopedisi'ne bakıldığında, Yapay Zeka'nın şu tanımıyla karşılaşılır:

Yapay Zeka, bilgisayar biliminin akıllı, yani dili kullanabilme, öğrenme, akıl yürütme, problem çözme gibi niteliklere sahip bilgisayar sistemleri tasarımıyla uğraşan koludur.

Benzer bir tanımla başka kaynaklarda, örneğin Patrick Winston'ın, bu alanın klasik kaynakları arasında sayılan *Yapay Zeka* başlıklı kitabında, veya Yapay Zeka'nın tarihçesinin 1940'lı yıllarda başladığını yazan *Ana Britanica* 'da da bulmak mümkündür.

Yapay zeka, insanın düşünme yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmak olarak da tanımlanabilir. Yani programlanmış bir bilgisayarın düşünme girişimidir. Daha geniş bir tanıma göre ise, yapay zeka, bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekasına özgü kapasitelerle donatılmış bilgisayarlardır.

Aşağıda çeşitli yapay zeka tanımları sunulmuştur:

Yapay Zeka, insanlar tarafından yapıldığında zeka gerektiren şeyleri gerçekleştiren makineler yapma bilimidir (**Minsky, 1968**).

Yapay Zeka, zeki davranışları taklit eden bilgisayar programlarının yapımı boyunca insan zekasının doğasını anlamayı amaç edinen disiplindir (**Bonnet, 1885**) .

1.3.Gelişme Süreci

Yapay zeka konusundaki ilk çalışma McCulloch ve Pitts tarafından yapılmıştır. Bu araştırmacıların önerdiği, yapay sinir hücrelerini kullanan hesaplama modeli, önermeler mantığı, fizyoloji ve Turing'in hesaplama kuramına dayanıyordu. Herhangi bir fonksiyonun sinir hücrelerinden oluşan ağlarla hesaplanabileceğini ve mantıksal “ve” ve “veya” işlemlerinin gerçekleştirilebileceğini gösterdiler. Bu ağ yapılarının uygun şekilde tanımlanmaları halinde öğrenme becerisi kazanabileceğini de ileri sürdüler.

Hebb, sinir hücreleri arasındaki bağlantıların şiddetlerini değiştirmek için basit bir kural önerince, öğrenebilen yapay sinir ağlarını gerçekleştirmek de mümkün hale gelmiştir.

1950’lerde Shannon ve Turing bilgisayarlar için satranç programları yazıyorlardı. İlk yapay sinir ağı temelli bilgisayar SNARC, MIT’de Minsky ve Edmonds tarafından 1951’de yapıldı. Çalışmalarını Princeton Üniversitesi’nde sürdüren Mc Carthy, Minsky, Shannon ve Rochester’le birlikte 1956 yılında Dartmouth’da iki aylık bir toplantı düzenledi. Toplantının en önemli özelliği Mc Carthy tarafından önerilen “Yapay Zeka” adının konmasıdır. İlk kuram ispatlayan programlardan Logic Theorist (Mantık Kuramcısı) burada Newell ve Simon tarafından tanıtılmıştır.

Daha sonra Newell ve Simon, “insan gibi düşünme” yaklaşımına göre üretilmiş ilk program olan General Problem Solver (Genel Sorun Çözücü)’ı geliştirmişlerdir. Simon, daha sonra fiziksel simge varsayımını ortaya atmış ve bu kuram, insandan bağımsız zeki sistemler yapma çalışmalarıyla uğraşanların hareket noktasını oluşturmuştur.

Bundan sonraki yıllarda mantık temelli çalışmalar egemen olmuş ve programların performanslarını sergilemek için bir takım yapay sorunlar ve dünyalar kullanılmıştır. Daha sonraları bu sorunlar gerçek yaşamı hiçbir şekilde temsil etmeyen “yapay dünyalar” olmakla suçlanmış ve yapay zekanın yalnızca bu alanlarda başarılı olabileceği ve gerçek yaşamdaki sorunların çözümünde kullanılamayacağı ileri sürülmüştür.

Geliştirilen programların gerçek sorunlarla karşılaştığında çok kötü bir performans göstermesinin ardındaki temel neden, bu programların yalnızca yapay bir şekilde çalışıp konu ile ilgili bilgileri kullanmamasıydı. Bu dönemin en ünlü programlarından Weizenbaum tarafından geliştirilen Eliza, karşısındaki ile sohbet edebiliyor gibi görünmesine karşın, yalnızca karşısındaki insanın cümleleri üzerinde bazı işlemler yapıyordu. İlk makine çevirisi çalışmaları sırasında benzeri yaklaşımlar kullanılıp çok gülünç çevirilerle karşılaşıncı bu çalışmaların desteklenmesi durdurulmuştur.

Her sorunu çözecek genel amaçlı program yerine belirli bir uzmanlık alanındaki bilgiyle donatılmış programlar kullanma fikri, yapay zeka alanında yeniden bir canlanmaya yol açtı. Kısa sürede “Uzman Sistemler” adı verilen bir metodoloji gelişti. Fakat burada çok sık rastlanan tipik bir durum, bir otomobilin tamiri için önerilerde bulunan uzman sistem programının otomobilin ne işe yaradığından haberi olmamasıydı.

İnsanların iletişimde kullandıkları Türkçe, İngilizce gibi doğal dilleri anlayan bilgisayarlar konusundaki çalışmalar bu sıralarda hızlanmaya başladı. Doğal dil anlayan programların dünya hakkında genel bilgiye sahip olması ve bu bilgiyi kullanabilmek için genel bir metodolojisi olması gerektiği belirtilmekteydi.

Bütün bu çalışmaların sonunda yapay zeka araştırmacıları iki gruba ayrıldılar. Bir grup, insan gibi düşünen sistemler yapmak için çalışırken, diğer grup ise rasyonel karar verebilen sistemler üretmeyi amaçlamaktaydı.

Aşağıda bu yaklaşımları kısaca inceleyeceğiz:

1.3.1. İnsan Gibi Düşünen Sistemler

İnsan gibi düşünen bir program üretmek için insanların nasıl düşündüğünü saptamak gerekir. Bu da psikolojik deneylerle yapılabilir. Yeterli sayıda deney yapıldıktan sonra elde edilen bilgilerle bir kuram oluşturulabilir. Daha sonra bu kurama dayanarak bilgisayar programı üretilebilir. Eğer programın giriş/çıkış ve zamanlama davranışı insanlardaki ile aynı ise programın düzeneklerinden bazılarının insan beyninde de mevcut olabileceği söylenebilir.

İnsan gibi düşünen sistemler üretmek bilişsel bilimin araştırma alanına girmektedir. Bu çalışmalarda asıl amaç genellikle insanın düşünme süreçlerini çözümlemeye bilgisayar modellerini bir araç olarak kullanmaktır.

1.3.2.İnsan Gibi Davranan Sistemler

Yapay zeka araştırmacılarının baştan beri ulaşmak istediği ideal, insan gibi davranan sistemler üretmektir. Turing zeki davranışı, bir sorgulayıcıyı kandıracak kadar bütün bilişsel görevlerde insan düzeyinde performans göstermek olarak tanımlamıştır. Bunu ölçmek için de Turing testi olarak bilinen bir test önermiştir. Turing testinde denek, sorgulayıcıyla bir terminal aracılığıyla haberleşir. Eğer sorgulayıcı, denneğin insan mı yoksa bir bilgisayar mı olduğunu anlayamazsa denek Turing testini geçmiş sayılır.

Turing, testini tanımlarken zeka için bir insanın fiziksel benzetiminin gereksiz olduğunu düşündüğü için sorgulayıcıyla bilgisayar arasında doğrudan fiziksel temastan söz etmekten kaçınmıştır. Burada vurgulanması gereken nokta, bilgisayarda zeki davranışı üreten sürecin insan beynindeki süreçlerin modellenmesiyle elde edilebileceği gibi tamamen başka prensiplerden de hareket edilerek üretilmesinin mümkün olmasıdır.

1.3.3.Rasyonel Düşünen Sistemler

Bu sistemlerin temelinde mantık yer alır. Burada amaç çözülmesi istenen sorunu mantıksal bir gösterimle betimledikten sonra çıkarım kurallarını kullanarak çözümünü bulmaktır. Yapay zekada çok önemli bir yer tutan mantıkçı gelenek zeki sistemler üretmek için bu çeşit programlar üretmeyi amaçlamaktadır.

Bu yaklaşımı kullanarak gerçek sorunları çözmeye çalışınca iki önemli engel karşımıza çıkmaktadır. Mantık, formel bir dil kullanır. Gündelik yaşamdan kaynaklanan, çoğu kez de belirsizlik içeren bilgileri mantığın işleyebileceği bu dille göstermek, hiç de kolay değildir. Bir başka güçlük de daha karmaşık sorunları çözerken kullanılması gereken bilgisayar kaynaklarının üstel olarak artmasıdır.

1.3.4.Rasyonel Davranan Sistemler

Bu yaklaşımda yapay zeka, rasyonel davranışların incelenmesi ve oluşturulması olarak tanımlanmaktadır. Rasyonel davranabilmek için gerekli koşullardan biri de doğru çıkarımlar yapabilmek ve bu çıkarımların sonuçlarına göre harekete geçmektir. Ancak,

yalnızca doğru çıkarım yapabilmek yeterli değildir. Çünkü bazı durumlarda doğruluğu ispatlanmış bir çözüm olmadığı halde yine de bir şey yapmak gerekebilir. Bunun yanında çıkarımdan kaynaklanmayan bazı rasyonel davranışlar da vardır. Örneğin, sıcak bir şeye dokunduğunda insanın elini çekmesi bir refleks harekettir ve uzun düşünce süreçlerine girmeden yapılır.

Bu yüzden yapay zekayı rasyonel davranış tasarımı olarak gören araştırmacılar, iki avantaj öne sürerler. Birincisi “düşünce yasaları” yaklaşımından daha genel olması, ikincisi ise bilimsel geliştirme yöntemlerinin kullanılmasına daha uygun olmasıdır.

1.4.Yapay Zeka Türleri

Yapay zeka konusundaki araştırmalar şu gruplar altında toplanabilir.

- Bilgi Tabanlı Yapay Zeka ve Uzman Sistemler
- Doğal Diller (Bilgisayar ile Doğrudan İletişim)
- Beşeri Algılama Yeteneklerinin Simülasyonu (Görme, Konuşma, İşitme, Koklama vs.)
- Robotikler (Rutin, Kirli ve Tehlikeli İşler İçin Kullanılan Robotikler)
- Yapay Sinir Ağları
- Bulanık Mantık
- Sanal Gerçeklik

1.4.1.Bilgi Tabanlı Yapay Zeka ve Uzman Sistemler

Bilgi tabanlı yapay zeka sistemi, belli bir uygulama alanına (bilgisayar onarımı gibi) ilişkin pratik çözüm veya yordamlama bilgilerinden (sezgi, yargı ve çıkarımlar) oluşmuş bir bilgi tabanına dayalı olarak çalışır. İnsanların kendilerine ait bilgi tabanı sistemindeki **EĞER- O ZAMAN (IF-THEN)** kurallarını kullanarak belirli sorunları çözme kabiliyeti, bu yapay zeka türüne ilham kaynağı olmuştur. Bilgi tabanlı sistemlerin en gelişmiş örneği uzman sistemlerdir. Belli bir soruna ilişkin uzmanlık bilgileri bir uzman sistemin bilgi tabanına yerleştirildikten sonra kullanıcıların bu bilgiden yararlanmak amacıyla uzman sistemle kurduğu iletişim bir uzman şahısla

kurulan iletişimin bir benzeri olacaktır. Sorun çözülene kadar kullanıcı ile bilgisayar tabanlı uzman sistem arasında karşılıklı soru-cevap türünde bir iletişim oluşur.

Yıllık vergi iadesi formunun hazırlanmasında bireylere yardımcı olmak üzere hazırlanan “DAN” isimli yazılım, bilgi tabanlı sisteme güzel bir örnektir. Sistem kullanıcının veri girmesine yol gösterici olacak bir vergi iadesi formunu içermektedir. Girilen verilere bağlı olarak gerekli olan hesaplamalar sistem tarafından otomatikman yapılmaktadır. Elde edilen çıktı doğrudan resmi makamlara sunulabilecek formatta olduğundan, herhangi bir uzmanın yardımına gereksinim duyulmadan vergi iadesi formu bireylerce hazırlanabilmektedir. DAN vergi yasalarındaki değişikliklere bağlı olarak her yıl yeniden gözden geçirilmekte ve gerekli değişiklikler yapılmaktadır.

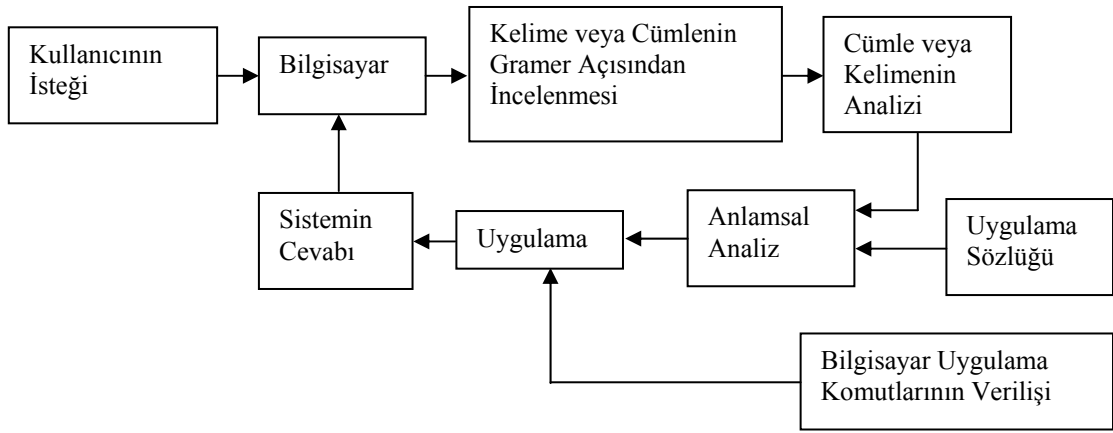
Normal yazılımlardan DAN’ın farkı, DAN’da içerilen “Ask DAN (DAN’a sor)”ve “Checklist (Kontrol listesi)” sistemleridir. “Ask DAN”, bir vergi uzmanının bilgisayarlaştırılmış bir biçimi olup, kullanıcı ile soru-cevap şeklinde bir iletişim kurarak ona yardımcı olan bir sistemdir. Dan isimli bir vergi uzmanından bilgi tabanının yaratılmasında istifade edildiğinden dolayı bu sisteme alan uzmanın adı olan “DAN” ismi verilmiştir. Kontrol listesi sistemi ise kullanıcıya bazı hususları belirleme açısından sorular sorar. Örneğin, ne kadar gelir beyan edilmeli, hangi muafiyetler kullanılmalı vb. gibi vergi iadesi formunun hazırlanması açısından gerekli olan cevaplar bu yazılımla belirlenmektedir.

Askeri alanda da bilgi tabanlı sistemlerden yararlanmak mümkündür. Örneğin karar verme durumunun karmaşık; buna karşın karar verme mantığının belli bir kural hiyerarşisine dönüştürülebileceği uzmanlık alanlarında eğer bilgi tabanlı sistemden yararlanmak ekonomik ise böylesi sistemler ihtiyaca bağlı olarak yaratılabilir.

1.4.2.Doğal Diller

Doğal diller nihai kullanıcının doğal dili ile (ingilizce gibi) bilgisayarla iletişim kurmasını sağlayan yazılımlar için kullanılan isimdir. Doğal dil yazılımlarındaki nihai amaç, geleneksel program dillerinde kullanılan komutlara olan gereksinimi ortadan

kaldırmaktır. Fakat halen uygulamada gelinen nokta tatmin edici düzeyde değildir. Şu anda piyasada kullanılan doğal dillerin çoğu kullanıcının bir uzman sistem ya da veri tabanı ile iletişimini sağlamaktan öte bir fonksiyon görememektedir. Yine de, bilgi işleminin sınırlı olduğu bazı alanlarda doğal dil uygulamasının oldukça başarılı olduğu gözlenmektedir. Örneğin insan kaynakları ve satın alma ile ilgili araştırma ve rapor hazırlama faaliyetlerinde doğal dil uygulaması oldukça gelişmiş düzeyde olup kullanıcı normal İngilizce konuşur gibi bilgisayar ile iletişim kurabilmektedir. Doğal dilin nasıl çalıştığı Şekil 1.1’de gösterilmektedir.



Şekil 1.1

Örneğin aşağıdaki sorunun kullanıcı tarafından sorulduğunu varsayalım:

“Pazarlama bölümünde görev itibariyle ortalama maaş nedir?”

Doğal dil yazılımı yukarıdaki soru cümlesini bizim gramer (dil bilgisi) çalışmasında yaptığımız gibi analiz eder. Cümle kelime kelime parçalara ayrılarak yukarıdaki cümle bilgisayarın anlayacağı uygulama komutlarına dönüştürülür. Anlam analizi aşamasında, cümlenin öğeleri genellikle yüklemlerle başlayarak uygulama sözlüğündeki anahtar kelimeler ile karşılaştırılır. Uygulama sözlüğünde yer alan kelimeler günlük dilde kullanılan kelimelerdir. Bu örnekte sistem soru amacıyla kullanılan kelimelerle karşılaştırma yapacaktır.

Yukarıdaki örnekte “nedir ?” kelimesi uygulama komutu olan”DISPLAY”e (Göster) doğal dil yazılımı vasıtasıyla dönüştürülecektir. Diğer kelimeler de benzeri şekilde doğal dil yazılımıyla yorumlanarak uygulama komutlarına dönüştürülür. Eğer kullanıcının isteği sistem tarafından anlaşılamazsa, doğal dil yazılımı muğlak olan noktaları sorarak isteği anlama çabasını devam ettirir. Örneğin sistem “Nedir?” ifadesini anlamamışsa, “Nedir?” i anlayamadım”der ve bunun anlamının ne olduğunu uygulama sözlüğünde yer alan benzeri komutları ya da ifadeleri sıralayarak kullanıcıya bunlardan hangisinin “ Nedir?” ile aynı anlamda olduğunu sorar. Gelen cevaba göre süreç devam eder.

1.4.3.İnsan Algılama Yeteneklerinin Benzetimi

Bu yapay zeka türü, insani yeteneklerin benzetimi ile ilgili olup bilgisayar sistemlerini görme, işitme, konuşma ve hissetme (dokunma) yetenekleri ile donatma çabasıdır. Bu yapay zeka yeteneklerini bugünün teknolojisini kullanarak belirli ölçüde gerçekleştirmek mümkün gözükmektedir.

İnsan algılama yeteneklerine sahip bilgisayarlar tıpkı insanlar gibi çevre ile iletişim kurma becerisine sahip olabilmektedir. Aşağıda buna bazı örnekler verilmektedir.

Konuşma ; Ses Cevap Üniteleri

Şayet uluslararası veya şehirlerarası bir telefon numarasını aramışsanız, “Aradığınız numara kullanımda değildir” veya tam otomatik bir arabaya binmişseniz “Emniyet kemerinizi takınız” gibi sözel mesajlara şahit olmuşsunuzdur. Bu mesajlar konuşma makinelerinin ses cevap ünitelerinden gelen mesajlardır. İki ayrı ses cevap ünitesi türü vardır; birinci tür bir insan sesinin kaydını kullanırken diğeri bir konuşma synthesizer’ından yararlanır.

Birinci tür, kullanıcı tarafından kasete önceden kaydedilmiş kelime, cümle, müzik, alarm gibi kayıtlardan çıktıyı seçer. Bu ses cevap ünitelerinde, sesin gerçek analog boyutları digital verilere çevirerek bir hafıza yongasına sürekli kullanılacak biçimde yüklenir. Çıktı alınırken ise, seçilen ses tekrar analog hale dönüştürülür. Bu tür yongalar

belirli kullanım alanları için seri üretim teknolojisiyle üretilmektedir. örneğin mikrofonlar, yangın alarm cihazları, asansörler, alarm saatleri, otomobil uyarı sistemleri, video oyunları gibi cihaz ve arabalarda kullanılan ses cevap üniteleri bu gruba girer.

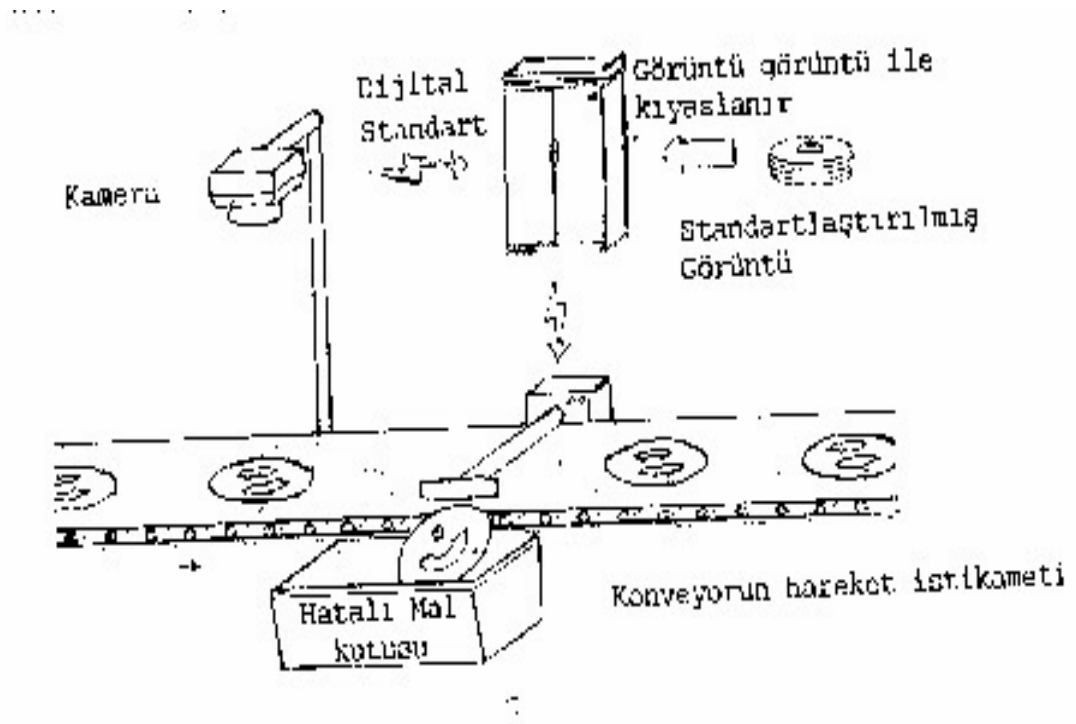
Konuşma Synthesizer'ları ise ham veriyi elektronik olarak üretilmiş konuşmalara dönüştürür. Bunun için de, bu cihazlar konuşmayı oluşturan temel ses birimlerine benzer sesleri bir arada kullanmaya çalışır. Bir konuşma Synthesizer'ı en az 64 temel sesi üretebilme kapasitesine sahiptir. Günümüz teknolojisi ile bunu sınırlı sayıda cümle için yapabilmek söz konusu ise de bu teknolojinin kullanım alanı gittikçe gelişmektedir. Örneğin, bugün okuyucu bir kitabı tarayarak ham verileri elde etmekte ve daha sonra konuşma synthesizer'ı bu ham verileri görme özürlü insanların istifadesine sunmak için konuşma haline getirmektedir. Diğer bir uygulama ise, konuşma özürlü çocuklar için geliştirilmiş sistemdir. Bu cihaz vasıtası ile bu çocuklar çevreleri ile konuşma imkanı elde edebilmektedir. Bu uygulama alanlarının daha da gelişeceği açıktır. Beklenildiğinin aksine, bu tür cihazlar nispeten ucuz olup, bu açıdan yakın gelecekte kişisel bilgisayarlarda da kullanım alanı bulacağı tahmin edilebilir.

İşitme ; Ses Tanımı

Bilgisayarlar büyük konuşmacıdırlar ama iyi bir dinleyici değildirler. Bilgisayarların çok doğal olan sesleri yanlış algılaması olağan bir şeydir. Bununla birlikte, ses tanımanın birtakım uygulamaları da mevcuttur. örneğin, satış elemanlarının telefonla bilgisayarı arayıp, müşteri ve sipariş numaraları ile sipariş miktarlarını bilgisayara girmesi bugünün teknolojisi ile mümkündür. Havaalanlarında bagaj yükleme biriminde çalışan elemanların üç harften oluşan varış adresini (örneğin, Los Angeles International için "L-A-X" harfleri) sözel olarak ifade etmesi ve bagajın sistem tarafından bu sese göre uygun konveyöre gönderilmesi bugünün uygulamalarından bir tanesidir. Bu sistem şu şekilde çalışmaktadır. Kişi mikrofona konuştuğunda her ses parçalara ayrıştırılır ve frekansları bulunur. Her frekanstaki ses dijital hale getirilerek bilgisayarın elektronik sözlüğünde yer alan formatla karşılaştırılır. Dijital format bilgisayarın 1 ve 0 olarak yorumladığı ve depoladığı bir formattır. Ses tanımda, veri tabanının yaratılması sürecine eğitim denir. Çoğu ses tanıma sistemleri konuşmacıya bağımlıdır, yani, ancak

belli konuşmacıların sesi sistem tarafından tanınabilmektedir. Bu nedenle, sistemi kullanan her kişi için ayrı bir kelime veri tabanının yaratılması gerekmektedir. Bu veri tabanını oluşturma sürecinde, sistemi kullanan kişinin sistemin her kelimeyi doğru anlamasını sağlamak açısından her kelimeyi en azından 20 defa tekrarlama gerekmektedir. Yani bir anlamda, bilgisayarı eğitmek gerekmektedir. Bu eğitim gerçekten de zorunludur. Çünkü nadiren bir kelimeyi farklı zamanlarda aynı şekilde ifade ederiz. Dolayısıyla bilgisayarın bu farklı ifadelerle alıştırılması gerekmektedir. Konuşmacıdan bağımsız sistemler ise, “evet”, “hayır” ve 10 haneli rakamlardan oluşmuş çok sınırlı bir sözcüğe sahip bulunmaktadır. Sözlüğü kısıtlı olmasına karşın bu sistemlerin eğitim gerektirmemesi, sistemin herkes tarafından kullanılmasını mümkün kılmaktadır.

Görme; Görsel Sistemler



Şekil 1.2

Görme, simülasyonu en zor olan insan algılama becerisidir. Örneğin, bir bilgisayarın bir insan gibi bir nesneyi görmesi ve onu yorumlaması mümkün değildir. Bilgisayara görüş kazandırmak için bir kameradan yararlanılır. Kamera aracılığıyla veri tabanı yaratmak

için gerekli olan girdiler elde edilir. Bir görsel sistem kamera desteğiyle yorumlanması istenen nesnenin standart halini dijital hale getirir ve bu dijital hale getirilmiş nesnelerin görüntüsü veri tabanına yüklenir. Daha sonra dijital sistem çalışırken, kamera görüntüyü dijital çeviriciye gönderir. Dijital hale getirilmiş bu görüntü bilgisayarın veri tabanındaki önceden kaydedilmiş dijital görüntüler ile karşılaştırılır. Bu kıyaslama neticesinde sistem nesneyi tanır. Aşağıdaki şekilde benzeri bir prensiple çalışan bir sistemin yapısı gösterilmektedir.

Şekil 1.2’den de görüldüğü gibi, parçalar dijital görsel sistem tarafından hatalı olanları saptamak için incelenmektedir. Eğer parçanın dijital görüntüsü standart dijital görüntü ile uyuşmuyorsa, parçanın hatalı olduğuna karar verilir ve hatalı parça sistem tarafından ayıklanır.

Görsel sistemleri ancak birkaç görüntünün (imajın) yer alabildiği özel durumlar için kullanmak mümkündür. Bu durumların ortak özelliği, basit ve monoton olmasıdır. Örneğin kalite kontrol durumu basit ve monoton bir olaydır.

1.4.4.Robotikler

Robotikler bilgisayarlar ile endüstriyel robotların uyumlu bir bütünleşmesidir. Endüstriyel robotlara bilgisayarlar yardımıyla herhangi bir rutin hareketin nasıl yapılacağını öğretmek mümkündür. Örneğin; araba boyama, vida sıkma, malzeme taşıma ve hatta kusurlu parçaları tespit etme gibi daha karmaşık davranışları yapan robotikleri günümüzde görmek mümkündür.

Yapay zekanın en büyük ticari başarıyı elde ettiği alan robotik alanıdır. Genel inancın aksine, robotikler bilim-kurgu filmlerinde görülen robotlardan gerek görünüm gerekse işlev açısından oldukça farklıdır. Endüstriyel robotların, günümüzde en fazla kullanılanı ise bir bilgisayar tarafından kontrol edilen bir mekanik koldur. Manipulatör olarak da adlandırılan bu kol bir insan kolunun yapabileceği çoğu hareketi yapabilme becerisine sahiptir.

Endüstriyel robotlar daha ziyade tekdüze işler için uygundur. Örneğin, ağır yüklerin taşınmasında ve tehlikeli işlerin yapılmasında endüstriyel robotlardan yararlanmak iyi bir strateji olabilir. Bu tür tehlikeli ve ağır tekdüze işler hemen her iş alanında mevcuttur. Günümüzde otomotiv sektörü robotlardan en fazla yararlanan sektördür. Bu sektörde robotlar daha çok boyama ve montaj işlemlerinde kullanılmaktadır. Elektronik sektörü bu konuda ikinciliği tutmaktadır. Elektronik devrelerin testi ve yongaların yerleştirilmesi işlemlerinde robotlardan istifade edilmektedir. Bugün, artık cerrahi de bile robotlar kullanılabilmektedir. Örneğin, bir beyin cerrahına yardımcı olan robotları hastanelerde görmek mümkündür. Robotlar büyük bir doğruluk yüzdesi ile biyopsi yapabilmekte ve böylece ameliyatın daha hızlı, daha doğru ve daha güvenli yapılmasını sağlamaktadır.

Robotlara işin nasıl yapılacağı bilgisayar tarafından öğretilir. Bir bilgisayar programı ile robotları kontrol etmek mümkündür. Bu program robota hareketin zamanı, yönü, mesafesi gibi konularda komut veren bir programdır. Bir kere programlandıktan sonra, robotların hareketlerini kontrol etmeye fazla ihtiyaç yoktur. O artık işini büyük bir titizlikle herhangi bir şey talep etmeden (yeme, içme gibi) yapmaya devam edecektir.

Robotlar konusunda görülen bir diğer gelişme ise, robotlara bazı beşeri algılama becerisini yerleştirmektir. Daha önce açıklanan robotlar beşeri algılama becerilerine sahip olmadıklarından dolayı ancak tekdüze işleri yapma becerisine sahiptirler ve bu yüzden de bu tür robotlara “seç ve yerleştir” robotları denmektedir. Eğer bu robotlara görme, işitme, konuşma gibi beşeri algılama becerileri kazandırılırsa, bu robotların insan gibi davranması ve böylece bu robotlara akıllı denmesi de mümkün olabilecektir. Bugünkü teknoloji ile bir robotu görsel bir alt sistemle teçhiz edip, robotun belli standarttaki bir nesneden farklı nesneleri ayırt etmesi sağlanabilir. Doğal olarak, görsel sistem teknolojisindeki gelişmeler devam ettiği müddetçe, robotların tıpkı bir insan gibi işyerinde dolaşması da mümkün olabilecektir.

Sonuç olarak, robot teknolojisindeki gelişmeler bilim-kurgu filmlerinde gördüğümüz kimi sahneleri gerçek hale getirecek gibi görünmektedir. Bu noktada, robotların işçi çıkarımına yol açacağı ve işsizlik sorununu yaratacağı konusunda kuşklar da bulunmaktadır. Fakat iktisadi hayatta yaşanan gerçekler bu kuşkunun ne denli yersiz

olduğunu ortaya çıkarmıştır. Çünkü robot teknolojisinin gelişmesiyle yeni iş alanları doğmuştur. Robotların tasarımı, üretimini, satışını, montajını, programını, tamirini ve bakımını yapan insangücüne ihtiyaç doğmuştur. Ayrıca robotlar maliyetlerde düşüş yaratarak, bazı işletmeleri iflas etmekten de kurtarmıştır. ABD’de bazı sendikaların işletmenin iflas etmesini ve işçilerin işini kaybetmesini önlemek için işverenin robot teknolojisini kullanmasına sıcak baktığı şeklindeki bazı haberler basında yer almaktadır. Dolayısıyla robot teknolojisi verimliliği arttırdığı müddetçe kullanılmaya devam edecek ve bu verimlilik artışı toplumun genelinde de bir refah artışı meydana getirecektir.

1.4.5.Yapay Sinir Ağları (YSA)

1.4.5.1.Giriş

Sinirsel ağlar çeşitli yollarla birbirine bağlı birimlerden oluşmuş topluluklardır. Her birim iyice basitleştirilmiş bir nöronun niteliklerini taşır. Nöron ağları, sinir sisteminin parçalarında olup biteni taklit etmekte, işe yarar ticari cihazlar yapmakta ve beynin işleyişine ilişkin genel kuramları denemekte kullanılır. Sinirsel ağ içindeki birimler, her birinin belli işlevi olan katmanlar şeklinde örgütlenmiştir ve bu yapıya “yapay sinir ağı mimarisi” denir.

Günümüzde sinirsel ağ uygulamaları ya geleneksel bilgisayarlar üzerinde yazılım simülatörleri kullanılarak, veya özel donanım içeren bilgisayarlar kullanarak gerçekleştirilmektedir. Kredi risk değerlemesinden imza kontrolü, mevduat tahmini ve imalat kalite kontrolüne kadar uzanan uygulamalar yazılım paketlerinden faydalanılarak yapılmaktadır.

Başka bir deyişle Yapay Sinir Ağları (YSA), beynin fizyolojisinden yararlanılarak oluşturulan bilgi işleme modelleridir. Literatürde 100’den fazla yapay sinir ağı modeli vardır. Bazı bilim adamları, beynimizin güçlü düşünme, hatırlama ve problem çözme yeteneklerini bilgisayara aktarmaya çalışmışlardır. Bazı araştırmacılar ise, beynin fonksiyonlarını kısmen yerine getiren bir çok modelleri oluşturmaya çalışmışlardır.

YSA' nın öğrenme özelliği, araştırmacıların dikkatini çeken en önemli özelliklerden birisidir. Çünkü herhangi bir olay hakkında girdi ve çıktılar arasındaki ilişkiyi, doğrusal olsun veya olmasın, elde bulunan mevcut örneklerden öğrenerek daha önce hiç görülmemiş olayları, önceki örneklerden çağrışım yaparak ilgili olaya çözümler üretebilme özelliği YSA' lardaki zeki davranışın da temelini teşkil eder.

1943 yılında bir nörobiyolojist olan Warren Mc Culloch ve bir istatistikçi olan Walter Pitts, "*Sinir Aktivitesindeki Düşüncelere Ait Bir Mantıksal Hesap*" başlıklı bir makale ile ilk dijital bilgisayarlara ışık tutmuştur. John Von Neumann bu makaleyi, "elektronik beyinler" için bir kopya olarak görmüştür. Yapay zeka alanındaki araştırmacılar içerisinde istisnai bir yeri olan Marvin Minsky, bu makaleden aldığı ilhamla makroskobik zeka fikrini ortaya atmış ve uzman sistemlerin doğmasına neden olmuştur. Bronx Yüksek Bilim Okulu'ndan Frank Rosenblatt, gözün hesaplamaları ile ilgilenmiştir. Bu bilim adamları, öğrenmenin ve zekanın herhangi bir özelliğinin simülasyonunda bilgisayarların aktif olarak nasıl kullanılabileceğini, 1956 yılında düzenlemiş oldukları ilk yapay zeka konferansında tartışmışlardır.

1959'da, Stanford üniversitesinden Bernard Widrow, basit nöron benzeri elemanlara dayanan ve "adaline" (Addaptive Linear Neuron) olarak adlandırılan bir adaptif lineer elemanı geliştirmiştir. Adaline ve iki tabakalı biçimi olan "madaline" (Multiple Adaline); ses tanıma, karakter tanıma, hava tahmini ve adaptif kontrol gibi çok çeşitli uygulamalar için kullanılmıştır. Daha sonraları adaline, ayrık bir çıkış yerine sürekli bir çıkış üretmek için geliştirilmiştir. Widrow, telefon hatları üzerindeki ekoları elimine etmeye yarayan adaptif filtreleri geliştirmede, adaptif lineer eleman algoritmasını kullanmıştır. Bununla ilk defa YSA' ları gerçek bir probleme uygulanmıştır.

Helsinki Teknik Üniversitesi'nden Teuvo Kohonen, 1970'lerin ilk yıllarında adaptif öğrenme ve birleşik hafızalar üzerine temel çalışmalar yapmış ve bu olduğu çalışmaları ile danışmansız öğrenme metotlarının gelişmesine ışık tutmuştur.

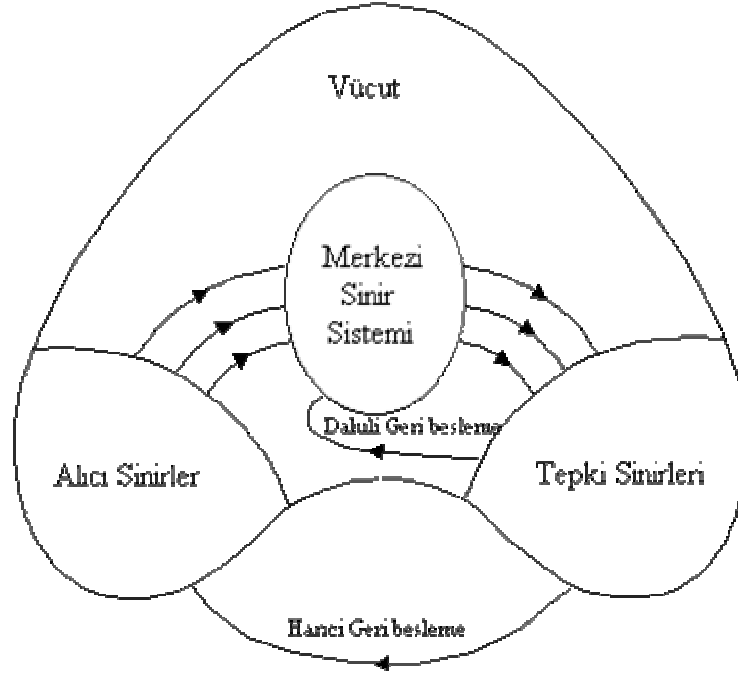
Minsky ve Papert' in Perseptron isimli kitaplarında, YSA' nın temel olarak ilgi çekici konular olmadığını belirtmeleri bir çok araştırmacının bu alanda çalışmaktan vazgeçmelerine sebebiyet vermiştir. YSA konusunda çalışmaya devam eden Grossberg, YSA modellerini yapılandırmak için nörolojik verinin kullanılması, algı ve hafıza için YSA tabanlı mekanizmaların önerilmesi, belirgin eşitliklerle bütünleşen bir sinaptik model için bir ilişkilendirici kural üzerinde çalışmıştır.

1982 yılında ilgi çeken bir başka gelişme, moleküller biyolojiden beyin kuramcılığına geçiş yapan bir model Caltech fizikçisi Hopfield tarafından sunulmuştur. Kendi adıyla anılan bir ağ yapısı mevcuttur ve bir çok alana uygulanmıştır.

1987 yılında yapılan ilk YSA sempozyumundan sonra YSA uygulamaları yaygınlaşmıştır. Günümüzde, YSA ile ilgili araştırmalar yapan çok sayıda bilim adamı ve araştırma grupları vardır. Farklı bilim ve ilgi alanlarında çalışan birçok araştırmacı, birçok yeni gelişmeyi sunmaya devam edeceklerdir.

1.4.5.2.Biyolojik Nöron Yapısı

Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin (merkezi sinir ağı) bulunduğu 3 katmanlı bir sistem olarak açıklanmaktadır. Alıcı sinirler (receptor), organizma içerisinden ya da dış ortamlardan algıladıkları uyarıları, elektriksel sinyallere dönüştürerek beyne iletirler. Tepki sinirleri (effector) ise, beynin ürettiği elektriksel sinyalleri organizma çıktısı olarak uygun tepkilere dönüştürür. Şekil 1.3'de biyolojik sinir sisteminin blok şeması görülmektedir.



Şekil 1.3: Biyolojik sinir sisteminin blok gösterimi.

Sinir hücreleri nöron olarak bilinir. Nöron, özellikle beyin olmak üzere sinir sisteminin temel birimidir ve başlıca üç kısımdan oluşur. Bunlar:

- Gövde (cell body)
- Gövdeye gire sinyal alıcı lifler (dendrit),
- Gövdeden çıkan sinyal iletici lifler (axon).

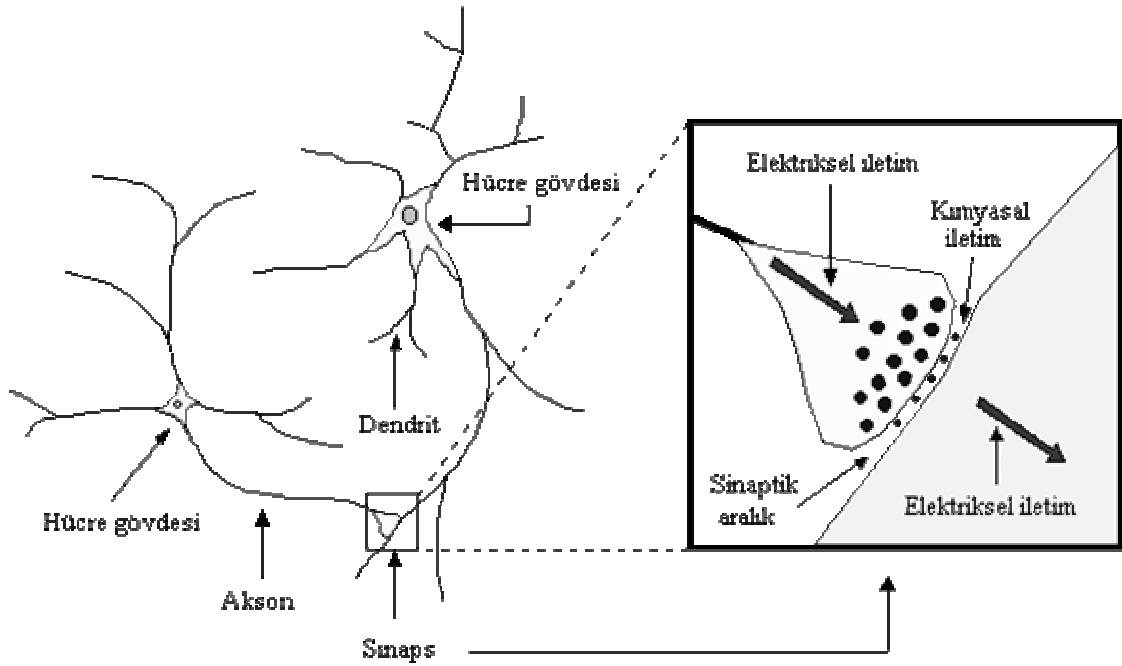
YSA' lar, insan beyninin çalışma prensibi örnek alınarak geliştirilmeye çalışılmıştır ve aralarında yapısal olarak bazı benzerlikler vardır. Bu benzerlikler Tablo 1.1'de verilmiştir.

Tablo 1.1: Sinir sistemi ile YSA' nın benzerlikleri

SİNİR SİSTEMİ	YSA SİSTEMİ
Nöron	İşlem elemanı
Dendrit	Toplama fonksiyonu
Hücre gövdesi	Transfer fonksiyonu
Aksonlar	Eleman çıkışı
Sinapslar	Ağırlıklar

Şekil 1.4'de bir nöron hücresinin yapısı verilmiştir. Bir nörondan yüzlerce, bazen de binlerce dendrit çıkabilir. Bunların uzunluğu genellikle bir milimetreden daha kısadır. Bazıları ise birkaç milimetre uzunluğa ulaşabilir. Dendritler çevre hücrelerden gelen sinyalleri (impulse) alıp, gövdeye ulaştırırlar. Her nöron, dendritler vasıtasıyla diğer bir çok nöronlardan gelen işaretleri alan ve birleştiren basit bir mikroişlem birimidir.

Beyin, sıkışık olarak ara bağlaşımlı milyarlarca (yaklaşık olarak 1011) nörondan oluşmaktadır. Her eleman kendi aralarında oldukça çok sayıda nörona (eleman başına yaklaşık olarak 104 bağlantı) bağlanmıştır. Bir nöronun aksonu (çıkış yolu) ayrıştırılmıştır ve bir sinaps olarak adlandırılan bir jonksiyon vasıtasıyla diğer nöronların dendritlerine bağlanmıştır. Bu jonksiyon uçlarındaki iletim doğal olarak kimyasaldır ve işaretin miktarı, akson tarafından serbest bırakılan kimyasalların büyüklüğüne bağlı olarak transfer edilir ve dendritler vasıtasıyla alınır. Bu sinaptik büyüklük, beyin öğrenirken neyin modifiye edildiğini belirtir. Bu sinaps, beynin temel hafıza mekanizmasına dayanarak nöron içerisindeki bilginin işlenmesi ile birleştirilir.

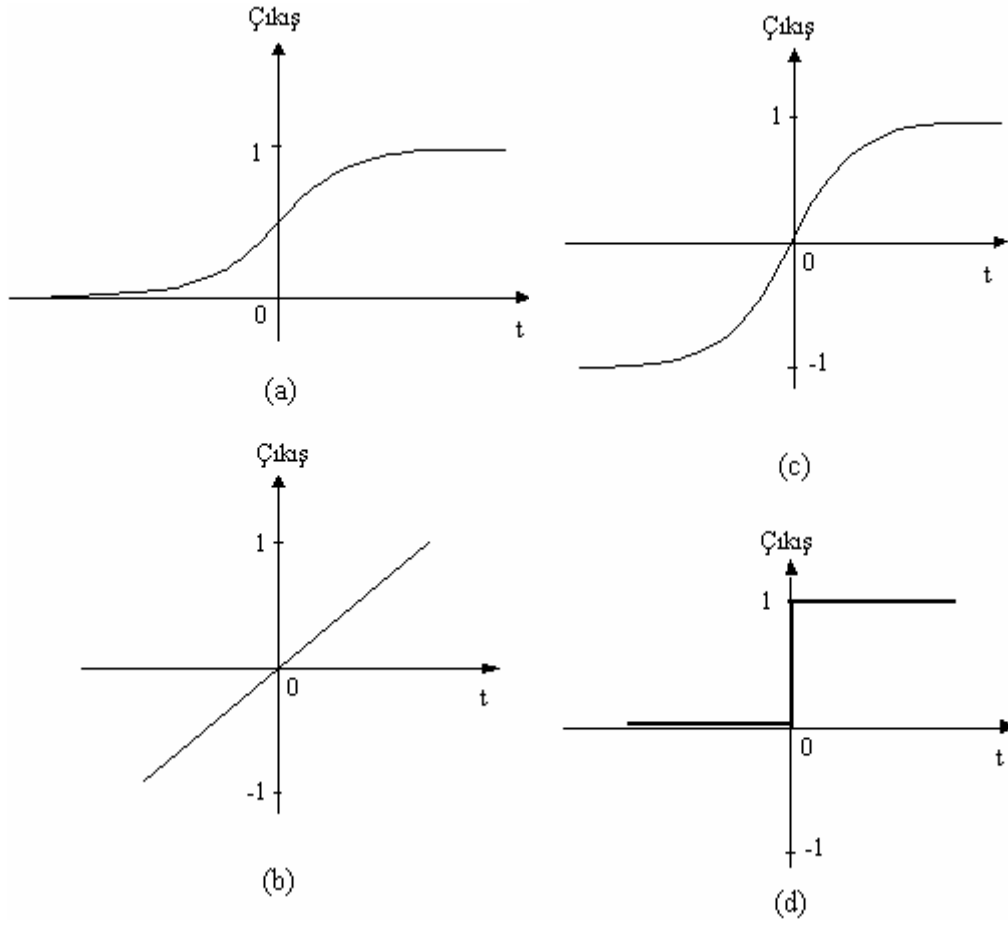


Şekil 1.4: Nöron

1.4.5.3. Aktivasyon Fonksiyonları

Şekil 1.5’da, bir hücrenin girdisine göre hücre çıkışını tanımlayan aktivasyon fonksiyonlarından 4 tanesi verilmiştir. Kullanılacak olan diğer fonksiyonların türevi alınabilir ve süreklilik arz etmesi gerekmektedir. YSA’ nın kullanım amacına uygun olarak tek veya çift yönlü aktivasyon fonksiyonları da kullanılabilir.

YSA’ da en çok kullanılan aktivasyon fonksiyonu Şekil 1.5a’da verilmiştir. Doğrusal YSA’ da ve katmanlı YSA’ nın çıkış katmanında kullanılan doğrusal fonksiyon Şekil 1.5b’de verilmiştir ve işlemci elemanın girişini doğrudan işlemci elemanın çıkışı olarak verir. Bipolar özelliğe sahip olan tanjant hiperbolik fonksiyonu Şekil 1.5c’de gösterilmiştir. Giriş uzayının genişletilmesinde etkili bir aktivasyon fonksiyonudur. Şekil 1.5d’de görülen keskin sınırlayıcı aktivasyon fonksiyonu, mantıksal çıkış verir ve sınıflandırıcı ağlarda tercih edilir.



Şekil 1.5: YSA’ da en çok tercih edilen aktivasyon fonksiyonları

a) sigmoid, b) doğrusal, c) hiperbolik tanjant ve d) keskin sınırlayıcı

1.4.5.4.İşlemci eleman (Yapay Nöron)

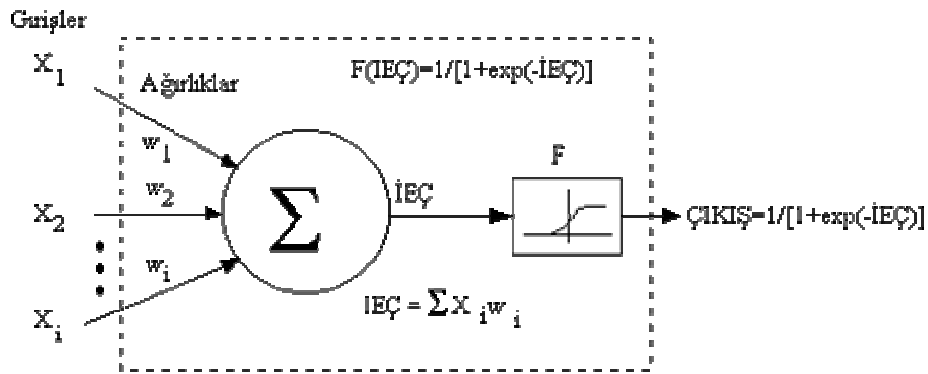
Bir YSA modelinin temel birimi, Şekil 1.6’da gösterilen işlem elemanıdır. Burada girişler dış kaynaklardan veya diğer işlem elemanlarından gelen işaretlerdir. Bu işaretler, kaynağına göre kuvvetli veya zayıf olabileceğinden ağırlıkları da farklıdır. YSA’ da girilen giriş değerlerine önce toplama fonksiyonları uygulanır ve her bir işlem elemanının çıkış (İEÇ) değeri :

$$IEÇ = \sum_{i=1}^n X_i W_{ij} - \theta_j \quad (1.1)$$

olarak bulunur. Burada X_i i' inci girişi, W_{ij} j' inci elemandan i' inci elemana bağlantı ağırlığını ve θ_i eşik (threshold) değerini göstermektedir. Daha sonra bu çıkış değerleri sigmoidal aktivasyon fonksiyonuna yani öğrenme eğrisine uygulanır. Sonuçta çıkış değeri aşağıdaki şekilde bulunur.

$$ÇIKIŞ = \frac{1}{1 + e^{-IEÇ}} \quad (1.2)$$

Transfer fonksiyonları olarak çoğunlukla, Şekil 1.5' de verilen fonksiyonlardan birisi kullanılabilir. Uygulamalarda, en çok *hiperbolik tanjant* veya *sigmoid* fonksiyonu kullanılmaktadır. Şekil 1.6' de işlemci eleman çıkışında kullanılan sigmoid fonksiyona göre çıkış değerinin hesaplanması gösterilmiştir. Bu işlemci elemanın çıkış değeri diğer işlemci elemanlarına giriş veya ağırlık çıkış değeri olabilir.



Şekil 1.6: Bir işlemci elemanı (yapay nöron).

1.4.5.5.YSA ile Hesaplamanın Özellikleri

YSA' nın hesaplama özelliklerini, paralel dağılmış yapısından, öğrenbilme ve genelleme yapma yeteneğinden aldığı söylenebilir. Genelleme, eğitim ya da öğrenme süresince kullanılmayan girişler için de YSA' nın uygun tepkileri üretmesi olarak tanımlanır. Bu özellikleri ile YSA' lar karmaşık ve çözülmesi güç problemleri de çözebilme yeteneğine sahiptir. Nesne tanıma , işaret işleme, sistem tanımlama ve

denetimi gibi birçok mühendislik alanında YSA' lar, aşağıda belirtilen özellikleri nedeniyle başarılı olmuşlardır.

Doğrusal Olmama

YSA' nın temel işlem elemanı olan hücre doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir ve bu özellik tüm ağa yayılmış durumdadır. Bu özelliği ile YSA, doğrusal olmayan karmaşık problemlere çözüm getirmektedir.

Öğrenme

Aslında öğrenmeden kasıt, ilgili problemdeki girdi-çıkı ilişkisini en güzel tanımlayacak optimum ağırlıkların bulunmasıdır. Problemden alınan örneklerden faydalanılarak ilgili problemi kendisine uygulanan örneklerden öğrenmeye çalışır. Probleme farklı bir çözüm sağlar.

Genelleme

YSA, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de belirtilen tepkiyi üretme kabiliyetine sahiptir. Örneğin, karakter tanıma amacıyla eğitilmiş bir YSA, bozuk karakter girişlerinde de doğru karakteri verirler. Nöral hesaplamada hafızalar birleşiktir. Yani eğitilmiş ağa girişin sadece bir kısmı verilse bile, ağ hafızadan bu girişe en yakını seçerek tam bir giriş verisi alıyormuş gibi kabul eder ve buna uygun bir çıkış değeri üretir. Veri YSA' ya, eksik , bozuk veya daha önce hiç karşılaşmadığı şekilde verilse bile, ağ kabul edilebilir en uygun çıkışı üretecektir. Bu özellik ağın genelleştirme özelliğidir.

Uyarlanabilirlik

YSA ağırlıkları, uygulanan probleme göre değiştirilir. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere göre tekrar eğitilebilir. Değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir. Bu özelliği ile YSA, uyarlamalı örnek tanıma, işaret işleme, sistem tanımlama ve denetim gibi alanlarda etkin olarak kullanılır.

Dağıtılmış Birleşik Hafıza

YSA' ların en önemli özelliklerinden biri de bilgiyi depolamalarıdır. Nöral hesaplamalarda bilgi ağırlıklar üzerine dağıtılmıştır. Bağlantıların ağırlıkları nöral ağın hafıza birimidir. Bu ağırlıklar ağın o andaki sahip olduğu bilgiyi veya uygulanan örneklerden öğrenmiş olduğu davranışı verir. Bu bilgiler, ağdaki birçok ağırlıklar üzerine (hafıza birimine) dağıtılır.

Eğitilmiş ağa, eğitimde kullanılmamış herhangi farklı bir giriş uygulanırsa ağ daha önceki girişlerden öğrenmiş olduğu davranış doğrultusunda beklenen çıkışa uygun bir çıkış değeri üretebilecektir. YSA' ya uygulanan veri eksik, gürültülü veya daha önce hiç karşılaşmamış olsa bile, ağ kabul edilebilir en uygun çıkışı üretecektir. Bu özelliğe, genelleştirme özelliği denir.

Hata Toleransı

YSA' lar, çok sayıda işlemci elemanların bağlantısı paralel dağılmış bir yapıya sahiptir ve ağın sahip olduğu bilgi, ağdaki tüm bağlantılara dağılmıştır. Giriş data setinde bulunabilecek herhangi bir gürültü, bütün ağırlıklar üzerine dağıtıldığından dolayı, gürültü etkisi tolere edilebilir. Geleneksel yöntemlere göre hatayı tolere etme yetenekleri daha fazladır.

Paralel İşlem Yapma

YSA' lar, paralel yapısı nedeniyle büyük ölçekli entegre devre (VLSI) teknolojisi ile gerçekleştirilebilir. Bu özellik, YSA' nın hızlı bilgi işleme yeteneğini ve örnek tanıma,

işaret işleme, sistem kimliklendirme ve denetim gibi gerçek zaman uygulamalarında kullanımı artırır.

1.4.5.6.Yapay Sinir Ağlarının Sınıflandırılması

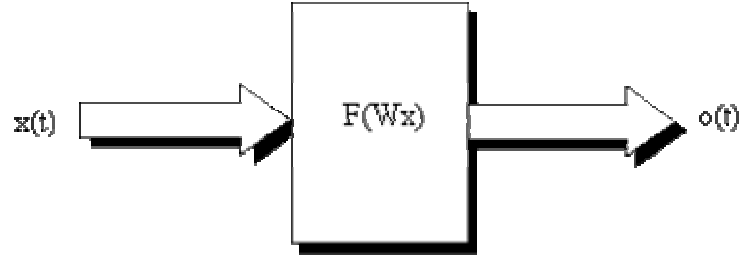
YSA' lar, genel olarak birbirleri ile bağlantılı işlemci birimlerden veya diğer bir ifade ile işlemci elemanlardan (neurons) oluşurlar. Her bir sinir hücresi arasındaki bağlantıların yapısı ağı yapısını belirler. İstenilen hedefe ulaşmak için bağlantıların nasıl değiştirileceği öğrenme algoritması tarafından belirlenir. Kullanılan bir öğrenme kuralına göre, hatayı sıfıra indirecek şekilde, ağı ağırlıkları değiştirilir. YSA' lar yapılarına ve öğrenme algoritmalarına göre sınıflandırılırlar.

1.4.5.6.1.YSA' ların Yapılarına Göre Sınıflandırılması

Yapay sinir ağları, yapılarına göre, ileri beslemeli (feed forward) ve geri beslemeli (feed back) ağlar olmak üzere iki şekilde sınıflandırılırlar.

İleri Beslemeli Ağlar

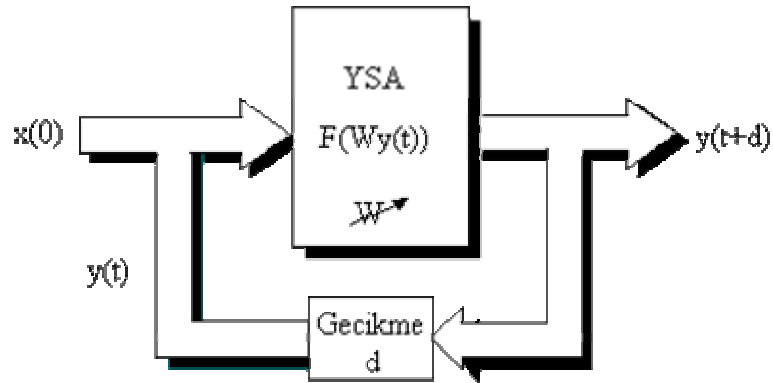
İleri beslemeli bir ağda işlemci elemanlar (İE) genellikle katmanlara ayrılmışlardır. İşaretler, giriş katmanından çıkış katmanına doğru tek yönlü bağlantılarla iletilir. İE' ler bir katmandan diğer bir katmana bağlantı kurarlarken, aynı katman içerisinde bağlantıları bulunmaz. Şekil 1.7'de ileri beslemeli ağ için blok diyagram gösterilmiştir. İleri beslemeli ağlara örnek olarak çok katmanlı perseptron (Multi Layer Perseptron-MLP) ve LVQ (Learning Vector Quantization) ağları verilebilir.



Şekil 1.7: İleri beslemeli ağ için blok diyagram

Geri Beslemeli Ağlar

Bir geri beslemeli sinir ağı, çıkış ve ara katlardaki çıkışların, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır. Böylece, girişler hem ileri yönde hem de geri yönde aktarılmış olur. Şekil 1.8’de bir geri beslemeli ağ görülmektedir. Bu çeşit sinir ağlarının dinamik hafızaları vardır ve bir andaki çıkış hem o andaki hem de önceki girişleri yansıtır. Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundur. Bu ağlar çeşitli tipteki zaman-serilerinin tahmininde oldukça başarı sağlamışlardır. Bu ağlara örnek olarak Hopfield, SOM (Self Organizing Map), Elman ve Jordan ağları verilebilir.



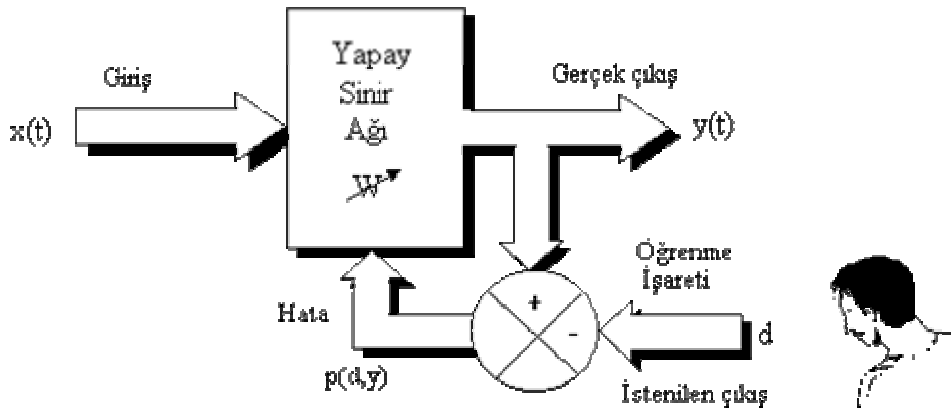
Şekil 1.8: Geri beslemeli ağ için blok diyagram

1.4.5.6.2.YSA' ların Öğrenme Algoritmalarına Göre Sınıflandırılması

Öğrenme; gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanmaktadır. O halde, birtakım metot ve kurallar, gözlem ve eğitime göre ağıdaki ağırlıkların değiştirilmesi sağlanmalıdır. Bunun için genel olarak üç öğrenme metodundan ve bunların uygulandığı değişik öğrenme kurallarından söz edilebilir. Bu öğrenme kuralları aşağıda açıklanmaktadır.

Danışmanlı Öğrenme (Supervised Learning)

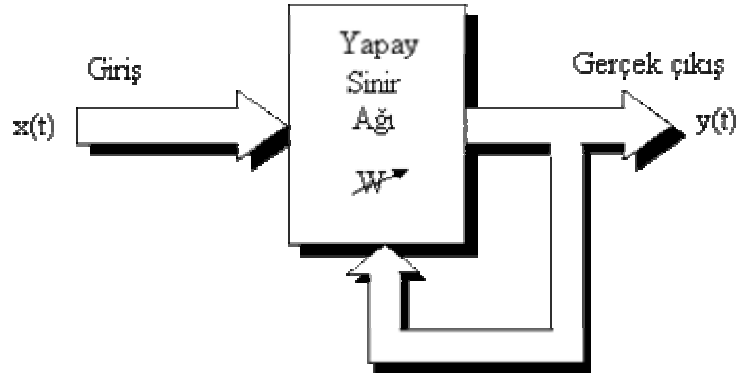
Bu tip öğrenmede, YSA' ya örnek olarak bir doğru çıkış verilir. İstenilen ve gerçek çıktı arasındaki farka (hataya) göre İE' ler arası bağlantıların ağırlığını en uygun çıkışı elde etmek için sonradan düzenlenebilir. Bu sebeple danışmanlı öğrenme algoritmasının bir “öğretmene” veya “danışmana” ihtiyacı vardır. Şekil 1.9’da danışmanlı öğrenme yapısı gösterilmiştir. Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geri besleme (back propagation) algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir.



Şekil 1.9: Danışmanlı öğrenme yapısı.

Danışmansız Öğrenme (Unsupervised Learning)

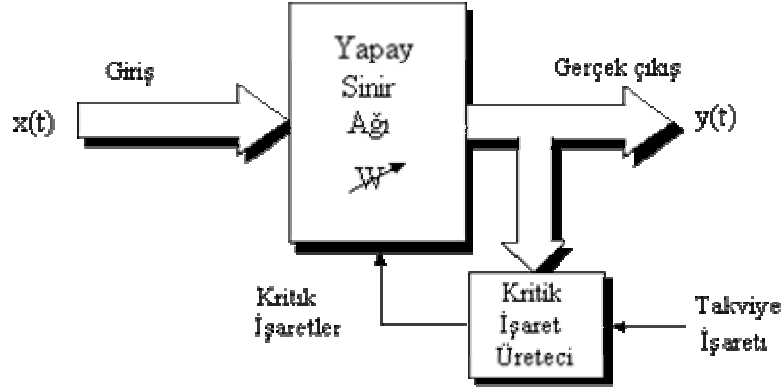
Girişe verilen örnekten elde edilen çıkış bilgisine göre ağ sınıflandırma kurallarını kendi kendine geliştirmektedir. Bu öğrenme algoritmalarında, istenilen çıkış değerinin bilinmesine gerek yoktur. Öğrenme süresince sadece giriş bilgileri verilir. Ağ daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren desenler (patterns) oluşturmak üzere ayarlar. Şekil 1.10’da danışmansız öğrenme yapısı gösterilmiştir. Grossberg tarafından geliştirilen ART (Adaptive Resonance Theory) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı danışmansız öğrenmeye örnek olarak verilebilir.



Şekil 1.10: Danışmansız öğrenme yapısı.

Takviyeli öğrenme (Reinforcement Learning)

Bu öğrenme kuralı danışmanlı öğrenmeye yakın bir metoddur. Denetimsiz öğrenme algoritması, istenilen çıkışın bilinmesine gerek duymaz. Hedef çıktıyı vermek için bir “öğretmen” yerine, burada YSA’ ya bir çıkış verilmemekte fakat elde edilen çıkışın verilen girişe karşılık iyiliğini değerlendiren bir kriter kullanılmaktadır. Şekil 1.11’de takviyeli öğrenme yapısı gösterilmiştir. Optimizasyon problemlerini çözmek için Hinton ve Sejnowski’ nin geliştirdiği Boltzmann kuralı veya GA takviyeli öğrenmeye örnek olarak verilebilirler.



Şekil 1.11: Takviyeli öğrenme yapısı.

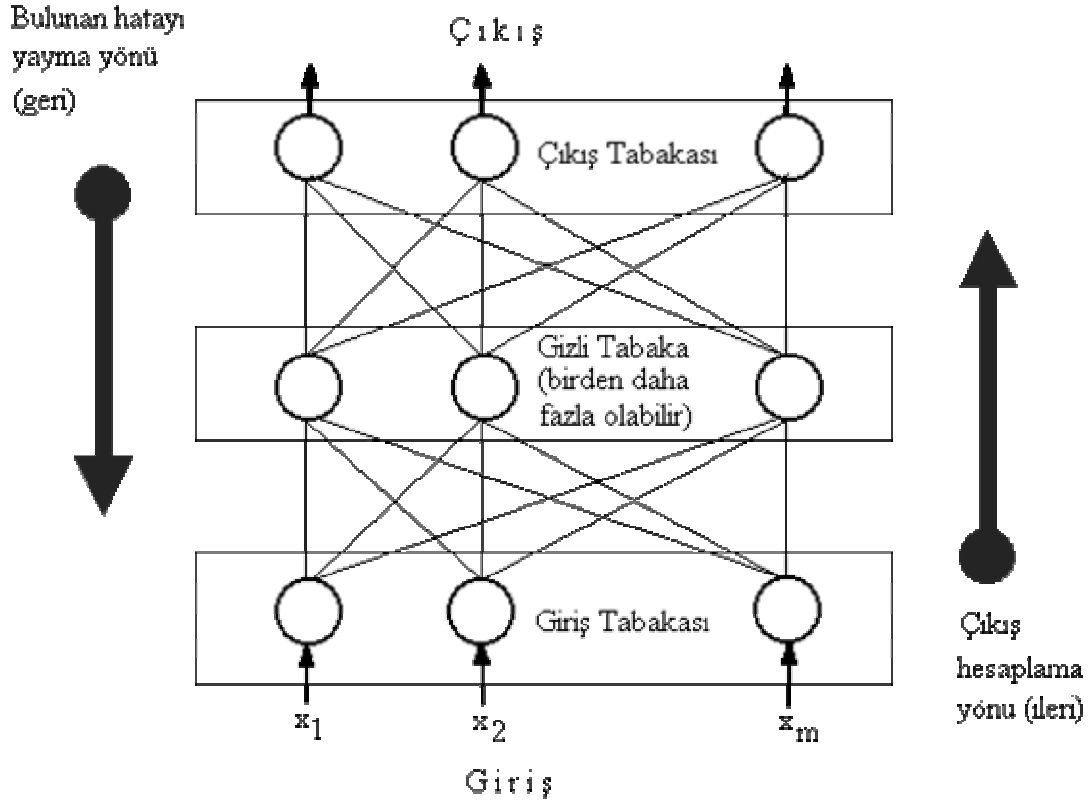
1.4.5.7.Çok Katmanlı Perseptronlar ve Öğrenme Algoritmaları

MLP' ler bir çok öğretme algoritması kullanılarak eğitilebilirler. Bu tez çalışmasında, danışmanlı öğrenme algoritmaları olan geri-yayılım (BP:back-propagation), hızlı yayılım (QP-Quick Prop), delta-bar-delta (DBD), genişletilmiş delta-bar-delta (EDBD-extended delta-bar-delta) ve takviyeli öğrenme metodu olan genetik algoritma (GA) mikroşerit antenlerin (MSA) parametrelerinin hesaplanmasında ve tasarımında kullanılan nöral modelleri eğitmede kullanılmıştır. Bu algoritmalar, bu bölümde detaylı olarak açıklanacaktır.

1.4.5.7.1.Çok Katmanlı Perseptronlar

Bir çok katmanlı perseptron sinir ağı modeli, Şekil 1.12'de gösterilmiştir. Bu ağ modeli özellikle mühendislik uygulamalarında en çok kullanılan sinir ağı modeli olmuştur. Bir çok öğretme algoritmasının bu ağı eğitmede kullanılabilir olması, bu modelin yaygın kullanılmasının sebebidir. Bir MLP modeli, bir giriş, bir veya daha fazla ara ve bir de çıkış katmanından oluşur. Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. Bilgi akışı ileri doğru olup geri besleme yoktur. Bunun için ileri beslemeli sinir ağı modeli olarak adlandırılır. Giriş katmanında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı tamamen uygulanan problemler giriş sayısına bağlıdır. Ara katman sayısı ve ara katmanlardaki işlem

elemanı sayısı ise, deneme-yanılma yolu ile bulunur. Çıkış katmanındaki eleman sayısı ise yine uygulanan probleme dayanılarak belirlenir.



Şekil 1.12: Geri yayılım MLP yapısı.

MLP ağlarında, ağa bir örnek gösterilir ve örnek neticesinde nasıl bir sonuç üreteceği de bildirilir (danışmanlı öğrenme). Örnekler giriş katmanına uygulanır, ara katmanlarda işlenir ve çıkış katmanından da çıkışlar elde edilir. Kullanılan eğitme algoritmasına göre, ağın çıkışı ile arzu edilen çıkış arasındaki hata tekrar geriye doğru yayılarak hata minimuma düşünceye kadar ağın ağırlıkları değiştirilir. İleri beslemeli ağlar, en genel anlamıyla giriş uzayıyla çıkış uzayı arasında statik haritalama yapar. Bir andaki çıkış, sadece o andaki girişin bir fonksiyonudur.

1.4.5.7.2.Geri Yayılım Algoritması

Bir çok uygulamalarda kullanılmış en yaygın öğretim algoritmasıdır. Anlaşılması kolay ve matematiksel olarak ispatlanabilir olmasından dolayı en çok tercih edilen öğretim algoritmasıdır. Bu algoritma, hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır.

Tipik çok katlı geri yayılım ağı, daima; bir giriş tabakası, bir çıkış tabakası ve en az bir gizli tabakaya sahiptir. Gizli tabakaların sayısında teorik olarak bir sınırlama yoktur. Fakat genel olarak bir veya iki tane bulunur. Desen sınıflandırma problemlerini çözmek için maksimum dört tabakaya (üç gizli ve bir çıkış tabakası) gereksinim duyulduğuna işaret eden bazı çalışmalar yapılmıştır. Geri yayılım algoritması, gradyan azalan ve MLP' leri eğitmede en çok kullanılan temel bir algoritmadır. Bu algoritmanın akış şeması Şekil 1.13'de verilmiştir. Eğitim işlemi ve eğitimden sonraki test işlemi bu akışa göre yapılır. Bu algoritma ile, i ve j kat işlem elemanları arasındaki ağırlıklardaki $\Delta w_{ji}(t)$ değişikliği hesaplanır. Bu ifade,

$$\Delta w_{ji}(t) = \eta \delta_j x_i + \alpha \Delta w_{ji}(t-1) \quad (1.3)$$

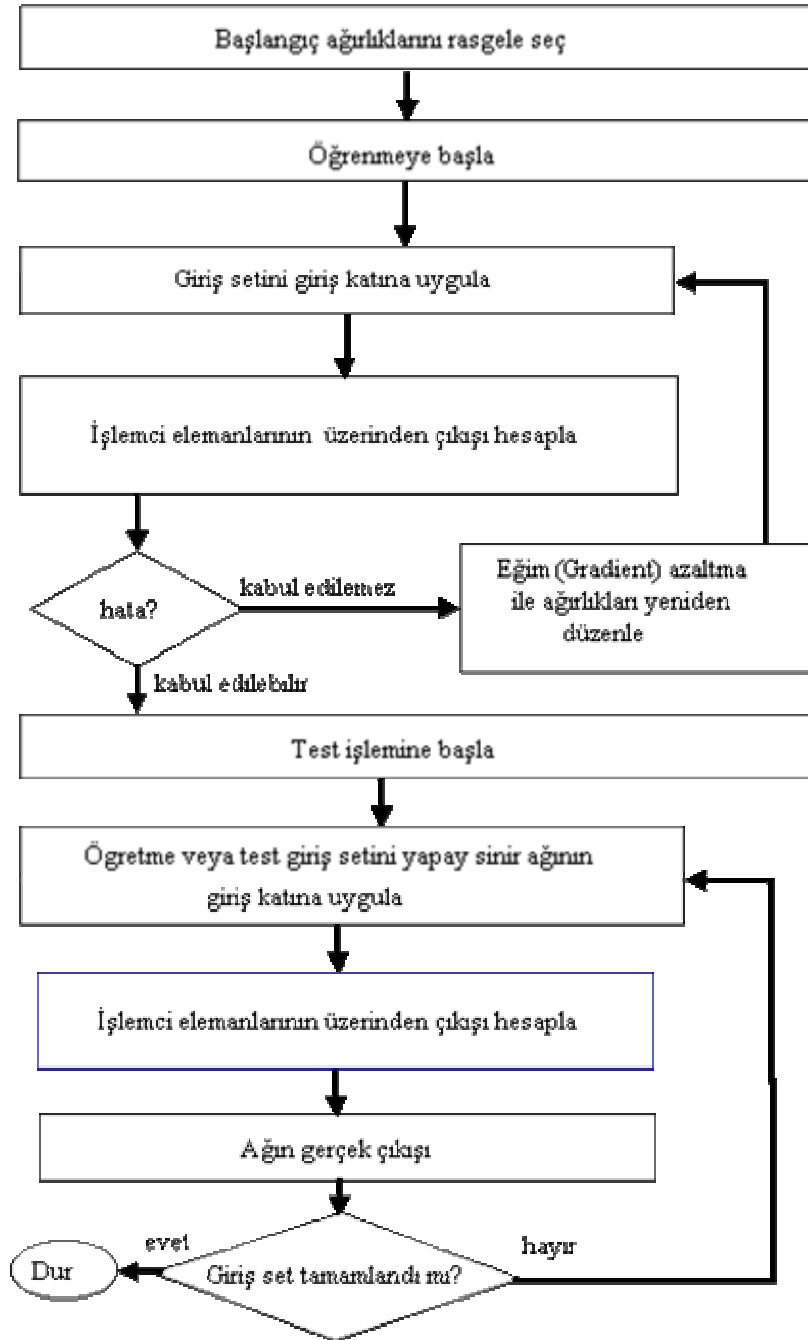
olarak verilir. Eşitlik (1.3)'de η öğrenme katsayısı, α momentum katsayısı ve d_j ara veya çıkış katındaki herhangi bir j nöronuna ait bir faktördür. Çıkış katı için bu faktör aşağıdaki şekilde verilir.

$$\delta_j = \frac{\partial}{\partial net_j} (y_j^{(t)} - y_j) \quad (1.4)$$

Burada, $net_j = \sum x_j w_{ji}$ ve $y_j(t)$ ise j işlemci elemanın hedef çıkışıdır. Ara katlardaki İE' ler için ise bu faktör,

$$\delta_j = \left(\frac{\partial f}{\partial net_j} \right) \sum w_{qj} \delta_q \quad (1.5)$$

olarak verilir. Ara katlardaki İE' ler için herhangi bir hedef çıkış olmadığından, eşitlik (1.4) yerine eşitlik (1.5) kullanılır. Bu duruma bağlı olarak çıkış katından başlayarak d_j faktörü, bütün katlardaki İE' ler için hesaplanır. Daha sonra eşitlik (1.3)'teki formüle bağlı olarak, bütün bağlantılar için ağırlıkların güncelleştirilmesi gerçekleştirilir.



Şekil 1.13: Çok katmanlı bir perseptron geri yayılım akış şeması

1.4.5.7.3.Delta-Bar-Delta (DBD)

Delta-Bar-Delta (DBD) çok katmanlı perseptronlarda bağlantı ağırlıklarının yakınsama hızını arttırmak için kullanılan bir sezgisel yaklaşımdır. Deneysel çalışmalar, ağırlık uzayının her boyutunun tüm hata yüzeyi açısından tamamen farklı olabileceğini

göstermiştir. Hata yüzeyindeki değişimleri açıklamak için, özellikle, ağın her bağlantısı kendi öğrenme katsayısına sahip olmalıdır. Bu düşünce, tek ağırlık boyutu için uygun adım büyüklüğü, tüm ağırlık boyutları için uygun olmayabilir. Bununla birlikte, her bir bağlantıya bir öğrenme oranı atanarak ve bu öğrenme oranının zamanla değişmesine izin verirse, yakınsama zamanını azaltmak için daha çok serbestlik derecesi sağlanmış olur.

İleri beslemeli YSA yapıları çoğu zaman karmaşıktırlar. Ağdaki her bağlantı için en uygun öğrenme katsayıları kümesini belirlemek oldukça zaman alıcı olabilir. Eğimin geçmişteki değerlerini kullanarak, yerel hata yüzeyinin eğriliğini çıkarmak için sezgisellik uygulanabilir. Bir bağlantı için ağırlık değişimlerinin işareti, birkaç ardışık zaman adımları sırayla değiştiği zaman, bağlantı için öğrenme oranı azalmalıdır. Bağlantı ağırlık boyutu hata yüzeyi ile ilgili büyük bir eğriliğe sahiptir. Bağlantı ağırlık değişimleri bir kaç ardışık zaman adımları için aynı işarete sahip olduğundan bağlantı için öğrenme oranı arttırılmalıdır. Ağırlık boyutu hata yüzeyi ile ilgili küçük bir eğriliğe sahiptir ve yüzey önemli bir mesafe için aynı doğrultudaki eğime göre devam eder.

Standart geri yayılım algoritmasında eğim bileşeni aşağıdaki şekilde verilir:

$$\delta(k) = \frac{\partial E(k)}{\partial w(k)} \quad (1.6)$$

Burada $E(k)$, k anındaki hata değerini, $w(k)$ bağlantı ağırlığını ve $d(k)$ ise ağırlık değişiminin eğim bileşenini göstermektedir. Standart geri yayılım algoritmasında bağlantı ağırlığı,

$$w(k+1) = w(k) + \alpha \delta(k) \quad (1.7)$$

olarak güncelleştirilir. Burada α sabit bir öğrenme oranıdır. DBD öğrenme kuralında, her bağlantı için değişken öğrenme oranı $\alpha(k)$ atanır ve bağlantı ağırlığının güncelleşmesi

$$w(k+1) = w(k) + \alpha(k) \delta(k) \quad (1.8)$$

şeklinde yapılır. Jacobs, her bağlantıda öğrenme hızının azalması ve artması için sezgiselliğin gerçekleştirilmesinde, $d(k)$ eğim bileşeninin ağırlıklı ortalamasını

kullanılmıştır. Bu ağırlıklı ortalama $\bar{\delta}(k)$ 'dır. Özellikle, ağırlıklı ortalama aşağıdaki şekilde gösterilmiştir.

$$\bar{\delta}(k) = (1 - \theta)\delta(k) + \theta\bar{\delta}(k-1) \quad (1.9)$$

Burada, θ konveks ağırlık faktörüdür. Sezgisel açıdan, önceki eğim bileşeninin üstel artması ve şu anki eğim bileşeni aynı işaretli ise, öğrenme oranı η sabiti ile artan ağırlıkla birleştirilir. Mevcut eğim bileşeni üstel ortalamadan farklı işaretli ise, öğrenme oranı mevcut değeri ile orantılı olarak azalır. Bu öğrenme oranının güncelleştirilmesi aşağıdaki eşitlikle tanımlanmıştır.

$$\Delta\alpha(k) = \begin{cases} \eta & \bar{\delta}(k-1)\delta(k) > 0 \\ -\eta\alpha(k) & \bar{\delta}(k-1)\delta(k) < 0 \\ 0 & \text{diğer} \end{cases} \quad (1.10)$$

Burada, η öğrenme katsayısı artma faktörü, η öğrenme katsayısı azaltma faktörü, $\alpha(k)$ k anındaki öğrenme oranını göstermektedir. Eşitlik (1.10)'dan görülebileceği gibi algoritma, öğrenme katsayılarını lineer olarak artırmakta, fakat geometrik olarak azaltmaktadır.

1.4.5.7.4. Genişletilmiş Delta-Bar-Delta (Extended DBD)

Genişletilmiş Delta-Bar-Delta (EDBD) öğretim algoritması, eğitimde belirli bir anda epoch genel hatayı hesaplayarak hata düzeltme özelliğine sahiptir. Belirli bir andaki (ya da herhangi bir anda) ölçülen hata, bir önceki anda bulunan hatadan büyük ise ağırlıklar bir önceki ağırlık değerlerini alır. Daha iyi sonuçlar elde etmek için hata düzeltme özelliğine dayanıklılık (patience) faktörü eklenmiştir. Her periyotdaki (epoch) hatayı test etmek yerine daha önce iyileştirilme işlemi yapıldığından hata, n periyotda test edilir.

EDBD' nin DBD' den farkı aşağıda özetlenmiştir.

- Sezgisel momentum kullanılır.
- Ağırlık uzayında büyük atlamalar ortadan kaldırılmıştır.
- Geometrik azaltma büyük (amacı aşan) atlamaları engellemede yeterince hızlıdır.
- EDBD' de, her bir bağlantıdaki momentum hızı $m(k)$ ve öğrenme hızı $a(k)$ zamanla değişmektedir. DBD ile karşılaştırmak için standart delta kuralı ile verilen momentum için bağlantı ağırlıklarının değiştirilmesi

$$\Delta w(k+1) = \alpha \delta(k) + \mu \Delta w(k) \quad (1.11)$$

değerinin ağırlıklara katkısıyla

$$w(k+1) = w(k) + \Delta w(k+1) \quad (1.12)$$

elde edilebilir. Burada a ve m sabittir. Bunun yanında EDBD için değişken öğrenme ve değişken momentum hızları aşağıdaki gibi elde edilebilir. Ağırlıklar,

$$\Delta w(k+1) = \alpha(k) \delta(k) + \mu(k) \Delta w(k) \quad (1.13)$$

değerinin bir önceki ağırlığa eklenmesiyle

$$w(k+1) = w(k) + \Delta w(k+1) \quad (1.14)$$

bulunur. Burada, $\mu(k)$ k zamanındaki bağlantı momentum hızıdır. Momentum hızı $\mu(k)$ ve öğrenme hızı $\alpha(k)$ aşağıdaki kurallara göre benzer şekilde ayarlanır. İlk önce, $\bar{\epsilon}$ nin hesaplanması DBD algoritmasına göre belirlenir.

$$\bar{\epsilon}(k) = (1 - \theta)(\delta(k) + \theta \bar{\epsilon}(k-1)) \quad (1.15)$$

Burada q konveks ağırlık faktörüdür. EDBD için öğrenme hızı değişimi:

$$\Delta\alpha(k) = \begin{cases} \kappa_{\alpha} \exp\{-\gamma_{\alpha}|\bar{\delta}(k)|\} & \text{eğer } \bar{\delta}(k-1)\delta(k) > 0 \\ -\varphi_{\alpha}\alpha(k) & \text{eğer } \bar{\delta}(k-1)\delta(k) < 0 \\ 0 & \text{diğer durumda} \end{cases} \quad (1.16)$$

formülünden elde edilir. Burada $\exp$, üstel fonksiyonunu, κ_{α} , sabit öğrenme hızı skala faktörünü φ_{α} , sabit öğrenme hızı azaltma faktörü ve γ_{α} , sabit öğrenme hızı üstel faktörünü göstermektedir. Benzer şekilde, k zamanındaki momentum hızı değişimi,

$$\Delta\mu(k) = \begin{cases} \kappa_{\mu} \exp\{-\gamma_{\mu}|\bar{\delta}(k)|\} & \text{eğer } \bar{\delta}(k-1)\delta(k) > 0 \\ -\varphi_{\mu}\mu(k) & \text{eğer } \bar{\delta}(k-1)\delta(k) < 0 \\ 0 & \text{diğer durumda} \end{cases} \quad (1.17)$$

ifadesinden bulunabilir. Burada, κ_{μ} , sabit momentum hızı skala faktörünü φ_{μ} , sabit momentum hızı azaltma faktörü ve γ_{μ} , sabit momentum hızı üstel faktörünü göstermektedir.

- Şuna da dikkat edilmelidir ki öğrenme ve momentum hızları, onların azalması ve artmasını kontrol eden farklı sabitlere sahiptir. $\bar{\delta}(k)\delta(k)$,³ nin işareti sezgisel olarak artma mı yoksa azalma mı olduğunu tespit etmek için kullanılır. Azalmanın ayarlanması DBD' nin kurallarına göre belirlenir. Ancak öğrenme ve momentum oranlarının artırılması $|\bar{\delta}(k)|$,³ nin üstel azalan fonksiyonu olarak değiştirilir. Böylece, küçük eğimli bölgelerde, büyük eğimli bölgelere nazaran daha büyük artırımlar yapılabilecektir. Bu, atlama problemi için kısmi bir çözümdür.

- Ağırlık uzayında, osilasyonları ve aşırı atlamaları engellemek için, üst sınırlar her bir bağlantı öğrenme hızları ve momentum hızları üzerinde belirlenir. Matematiksel ifadesi, tüm bağlantılar için aşağıdaki gibidir.

$$\alpha(k) \leq \alpha_{\max} \quad (1.18)$$

$$\mu(k) \leq \mu_{\max} \quad (1.19)$$

Burada, $\alpha_{\max}$ öğrenme katsayısının üst sınırı ve $\mu_{\max}$ ise momentum katsayısının üst sınırıdır.

Sonuç olarak, EDBD algoritmasına düzeltme özellikli bir hafıza yerleştirilmiştir. Kullanıldığı zaman, eğitme datalarının sunumu her iterasyondan sonra, hata değerlendirilir. Hata, $E(k)$, bir önceki minimum hatadan küçük ise ağırlıkları o anki en iyiler olarak hafızada saklanır. Düzeltme tolerans parametresi λ , düzeltme sürecini kontrol eder. O andaki hata önceki minimum hatayı aşarsa, yani

$$E(k) > E_{\min} \lambda \quad (1.20)$$

olursa, tüm bağlantı ağırlıkları, hafızada saklı olan en iyi değerlerle değiştirilir. Ayrıca, öğrenme ve momentum hızları, düzeltmeyi başlatmak için azaltılır.

1.4.5.7.5.Hızlı Yayılım (Quick Prop)

Hızlı yayılım (Quick Prop) algoritması, Scott Fahlman tarafından Carbegie Mellon Üniversitesi'nde geliştirilen ve Newton metoduna dayanan, çok katlı perseptronların eğitilmesi için kullanılan bir sezgisel öğretim algoritmasıdır.

Sezgisel algoritmalar, tecrübeye ve eğitilmiş tahmine dayalı kurallar içermektedir. Bu algoritmalar optimum çözümü garanti edemeyip sadece ona yakın çözümü bulmayı garanti etmektedir. Sezgisel metotları kullanmanın sağladığı bazı avantajlar şöyle özetlenebilir:

1-Sezgisel metotlar karar verici mekanizma için sadeleştirici olabilir.

2-Sezgisel metot, herhangi bir tam yöntemin parçası olarak öğrenme amacıyla kullanılabilir.

3-Gerçek dünya problemleri için her zaman matematiksel formülasyon kurmak çok kolay olmayabilir. Bu basitleştirme sonucu oluşan hata, bir sezgisel metodun sağladığı optimale yakın çözümün sahip olduğu hatadan daha büyük olabilir.

Problem uzayında mümkün olan en kısa sürede, bir çözüm bulmak için ağırlık uzayında eğimi (gradient) son derece küçük azaltmak gerekebilir. Bunun yerine çözümden fazla uzaklaşmaksızın mümkün olan en büyük adımlarla kısa sürede çözüme ulaşmak arzu edilir. Hata fonksiyonunun eğim derecesi biliniyorsa, güvenli bir şekilde ağırlık uzayında daha büyük adımla çözüme ulaşılabilir. Bu algorithmada iki geleneksel yaklaşım birleştirilerek çözülmüştür. Bunlar;

- Hesaplamanın geçmişteki durumu hakkında bazı sezgilere dayanan (genel veya her bir ağırlık için ayrı) dinamik olarak ağırlıkların ayarlanması.
- Her bir ağırlığa göre hatanın ikinci türevinin belirgin kullanımı
- Hızlı yayılım algoritması bir ikinci dereceli metoddan daha sezgiseldir ve çok az da olsa Newton-metoduna dayanan bu methodda iki kabul kullanılır.
- Her bir ağırlık için, ağırlık hata eğrisi kolları yukarı doğru açık olan bir parabol ile yaklaştırılabilir.
- Hata eğrisinin eğilimindeki değişim, diğer tüm ağırlıkların aynı andaki değişiminden etkilenmez.

Yukarıda bahsettiğimiz parabol, birbirinden bağımsız her bir ağırlık için şimdiki ve önceki hata eğimleri ve bu eğimlerin ölçüldüğü noktalar arasındaki değişim kullanılarak belirlenir. Daha sonra, algoritma doğrudan bu parabolün minimum noktasına atlar. Genellikle hızlı yayılım algoritmasının performans testleri diğer tekniklerle

karşılaştırıldığında oldukça iyidir. Bu algoritma, özellikle gürültü seviyesi az olan dataya sahip problemlerde, EDBD kadar iyi sonuç vermektedir. Hızlı yayılım algoritması gürültü seviyesi fazla olan datalı problemlerde çok iyi çalışır ancak gürültülü dataların bulunduğu problemlerde EDBD kullanılmalıdır.

Bu algoritmada ağırlık hızlandırma ve kırılması ihmal edilerek $t-1$ den t ' ye kadar ağırlıklardaki değişim,

$$\Delta w(t) = \epsilon L(t) + \alpha Q(t) \quad (1.21)$$

formülü ile verilir. Burada ϵ öğrenme katsayısı ve α ise momentum katsayısıdır.

$$L(t) = \begin{cases} h(t) & \text{eğer } h(t)h(t-1) \geq 0 \\ 0 & \text{diğer durumlarda} \end{cases} \quad (1.22)$$

ve

$$Q(t) = \begin{cases} \mu \Delta w(t-1) & \text{eğer } h(t) \left(h(t) - \left(\frac{\mu}{\mu+1} \right) h(t-1) \right) \geq 0 \\ \Delta q(t) & \text{diğer durumlarda} \end{cases} \quad (1.23)$$

eşitlikleri ile verilir. Bu ifadelerde m momentum büyüme faktörü,

$$h(t) = \partial E / \partial w(t) \quad \text{eğimi ve} \quad \Delta q(t) = \frac{\Delta w(t)h(t)}{h(t-1) - h(t)}$$

minimum adım miktarını göstermektedir. Buradan ağırlık fonksiyonunu güncelleştirmek için, delta ağırlık fonksiyonu ve ağırlık hızlandırma katsayısı işleme katılır.

$$w(t) = (1 - \delta)w(t-1) + \Delta w(t) \quad (1.24)$$

Burada δ hızlandırma katsayısıdır. Son olarak ağırlık çok küçük ise, 0 alınarak kırılır.

$$|w(t)| < \kappa \text{ ise } w(t) = 0 \text{ alınır.} \quad (1.25)$$

burada, κ ağırlık kırılma faktörüdür.

1.4.5.7.6.Genetik Algoritma

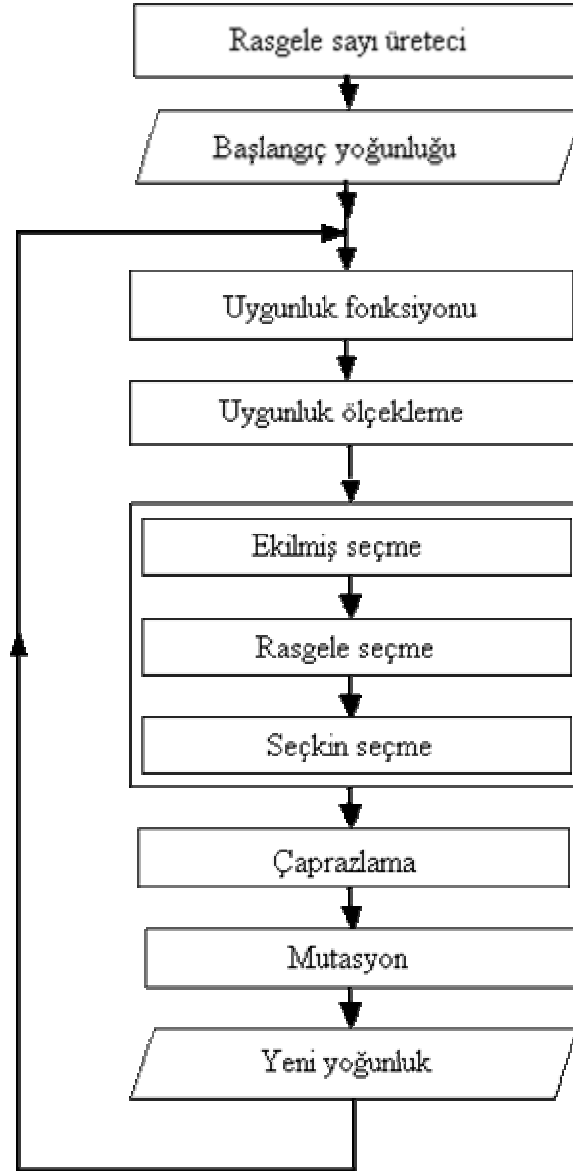
Genetik algoritmalar (GA), araştırma ve optimizasyon algoritmaları olup, canlılardaki doğal gelişim prensibine dayanmaktadırlar. Probabilistik karakterleri ve çoklu mümkün çözümleri araştırma gibi önemli özelliklere sahip olmaları ve amaç (objective) fonksiyonunun gradyentinin bilinmesine ihtiyaç duymamaları en önemli avantajlarıdır.

Genler ve kromozomlar doğal olarak türler ile ilgili bilgileri saklarlar. Bir GA' da potansiyel çözümler ile ilgili bilgiler, diziler (string) şeklindeki sayılarla ifade edilirler. Bu çalışmada kullanılan algoritmanın akış şeması, Şekil 1.14'de gösterilmiştir. Şekil 1.14'den de açıkça görülebileceği gibi, optimizasyonun başlangıcında, çoğunlukla rasgele olarak, bir sayı üretme jeneratörü tarafından bir grup çözüm üretilir. Başlangıçta iyi bir yaklaşık çözüm için, herhangi bir bilgiye ihtiyaç duyulmaması GA' nın diğer bir avantajıdır.

Başlangıç yoğunluğunun üretilmesini takiben, her çözümün uygunluğu veya iyiliği, seçilen bir uygunluk fonksiyonu kullanılarak değerlendirilir. Uygunluk fonksiyonu, eldeki veriler dikkate alınarak belirlendiği için, bunun seçimi probleme bağlıdır. Uygunluk ölçekleme işlemi, bazen daha iyiyi seçmek için elde edilen iyi çözümlerden daha iyi çözümlere kadar devam ettirilen bir işlemdir. Seçme, çaprazlama ve mutasyon gibi genetik operatörler, elde edilen iyi çözümlerden denenmiş yeni çözümler üretmek için kullanılırlar. Bu iyileştirme işlemi, daha önceden belirlenen bir jenerasyon sayısına veya tatmin edici bir sonuca ulaşınca kadar devam ettirilir.

Şekil 1.14' de, bu çalışmada kullanılan 3 farklı seçme operatörü verilmiştir. Bu operatörlerden ekilmiş (seeded) seçme, daha iyi çözümlere, daha fazla yaşama şansı veren yeni bir yoğunluk için çözüm anındaki yoğunluktan seçme işlemini gerçekleştirir. Rasgele (random) seçme, o andaki elde edilen çözümlerden tamamen rasgele olarak yeni bir yoğunluk oluşturur. Seçkin (elite) operatörü, her zaman en iyi çözümü bir

kuşaktan diğerine kopyalar. Çaprazlama operatörü, var olan iki çözümün uzantılarının belirli bir noktadan itibaren yer değiştirmesi ile denenmemiş yeni iki çözüm üretir. Tablo 1.2a ve 1.2b’de çaprazlama ve mutasyon işlemlerine ait bir örnek verilmiştir. Mutasyon operatörü ise, var olan bir çözümün herhangi bir bitinin terslendirilmesi ile yeni bir çözüm üretir.



Şekil 1.14: Genetik algoritmanın akış şeması

GA, ağ çıkışının hesaplanması için gerekli olan optimum ağırlıkların elde edilmesinde kullanılmıştır. Bu her bir ağırlık, W_{ij} ' nin bir bit dizisi şeklinde sıralanması ve sıraların tamamen bit dizileri şeklinde ifade edilmesiyle kolayca uygulanabilir. Sıraların veya bit dizilerinin uygunluğu toplam hataların karelerinin $E(w)$ toplamının tersi olarak ifade edilir. Eşiklikte $E(w)$ amaç hata fonksiyonu olup m başlangıç hata terimleri $e_i^2(w)$ aşağıdaki gibidir.

$$E(w) = \sum_{i=1}^m e_i^2(w) \quad (1.26)$$

Burada $e_i^2(w) = (y_i^{(t)} - y_i)^2$ dir.

Tablo 1.2: (a) Çaprazlama ve (b) Mutasyon operatörleri

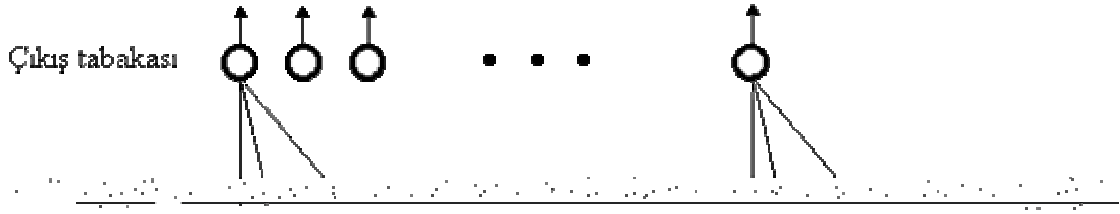
Çaprazlamadan Önce	Çaprazlamadan Sonra
Çözüm 1 : $\hat{e}$ 1 0 1 1 1 1 1 0 0 1 1 0 1 1 1 1	Yeni Çözüm 1: 1 0 1 1 1 1 1 0 1 1 1 1 1 0 0 0
Çözüm 2: $\hat{e}$ 1 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0	Yeni Çözüm 2: 1 1 0 0 0 1 1 0 0 1 1 0 1 1 1 1
(a)	
Mutasyondan Önce	Mutasyondan Sonra
$\hat{e}$ 1 0 1 1 1 1 1 0 0 1 1 0 1 1 1 0	1 0 1 1 1 1 0 0 0 1 1 0 1 1 1 0
(b)	

1.4.5.8. Diğer Yapay Sinir Ağları

Bu kısımda, literatürde adı en çok geçen YSA' lar kısaca açıklanacaktır.

1.4.5.8.1.LVQ (Learning Vector Quantization)

Şekil 1.15, üç kattan oluşan LVQ ağını göstermektedir. Bu tabakalar, giriş, gizli tabaka ve çıkış tabakasıdır. LVQ ağı, giriş ve gizli tabaka arasında tamamen, gizli ve çıkış tabakası arasında da kısmen bağlıdır. Her çıkış işlemci elemanı farklı bir gizli işlemci eleman kümesine bağlıdır. Gizli ve çıkış işlemci elemanları arasındaki ağırlıklar 1'e sabitlenmiştir. Giriş-gizli işlemci eleman bağlantılarının ağırlıkları "referans" vektörlerinin elemanlarını oluşturur (her gizli işlemci elemana bir referans vektör atanmıştır). Ağı öğretilmesi esnasında bunlar yeniden değerler alırlar. Hem gizli işlemci elemanlar (bunlar Kohonen işlemci elemanları olarak da bilinir) hem de çıkış işlemci elemanları ikili (binary) çıkışa sahiptir. Ağa bir giriş deseni verildiğinde referans vektörü giriş desenine en yakın olan gizli işlemci eleman kümesi "1", diğerleri "0" üretir. "1" üreten çıkış işlemci elemanı giriş işaretini sınıflar ve her işlemci eleman ayrı bir sınıfa atanmıştır.



Şekil 1.15: LVQ ağı

En basit LVQ öğretme prosedürü şu sırayla açıklanabilir.

- Referans vektörlerinin ağırlıklarının başlangıç değerlerini belirle,
- Ağa giriş deseni uygula,
- Giriş deseni ile her referans vektörü arasındaki “Öklit (Euclidian) uzaklığı”nı hesapla,
- Giriş desenine en yakın referans vektörünün (-ki bu vektör kazanan gizli işlemci elemanının referans vektörüdür) ağırlıklarını yeniden düzenle. Eğer bu gizli işlemci eleman, o çıkış işlemci elemanına bağlı olduğu gizli işlemci eleman kümesine bağlı ise referans vektörünü giriş desenine daha yakın hale getir. Aksi takdirde referans vektörünü uzaklaştır,
- İkinci adıma yeni bir giriş deseni ile dön ve işlemlere bütün giriş desenleri sınıflandırılncaya kadar devam et.

1.4.5.8.2.Hopfield Ağı

Şekil 1.16’de Hopfield ağının bir şekli gösterilmiştir. Bu ağ genellikle ikili (0 veya 1) ve bipolar (+1 veya -1) girişler kabul eder. Tek tabaka işlemci elemanları vardır ve her işlemci eleman bir diğerine bağlanmıştır. Bu da daha önceden belirtilen geri beslemeli yapıdır. Hopfield ağının eğitilmesi sadece bir adım alır ve w_{ij} ağırlıkları aşağıdaki gibi hesaplanır:

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{c=1}^P x_i^c x_j^c, & i \neq j \\ 0, & i = j \end{cases} \quad (1.27)$$

Burada w_{ij} , i işlemci elemanından j işlemci elemanına olan bağlantının ağırlığını, x_i^c c sınıfı için eğitim giriş deseninin i ’inci elemanını, P sınıf sayısını ve N de işlemci

eleman sayısını gösterir. Denklem (1.27)'de $w_{ij} = w_{ji}$ ve $w_{ii} = 0$ durumları ağıın kararlılığını garantiler. Bilinmeyen bir bilgi ağı girildiğinde, ağıın çıkışları bilinmeyen bir desen elemanlarına eşitlenir, yani;

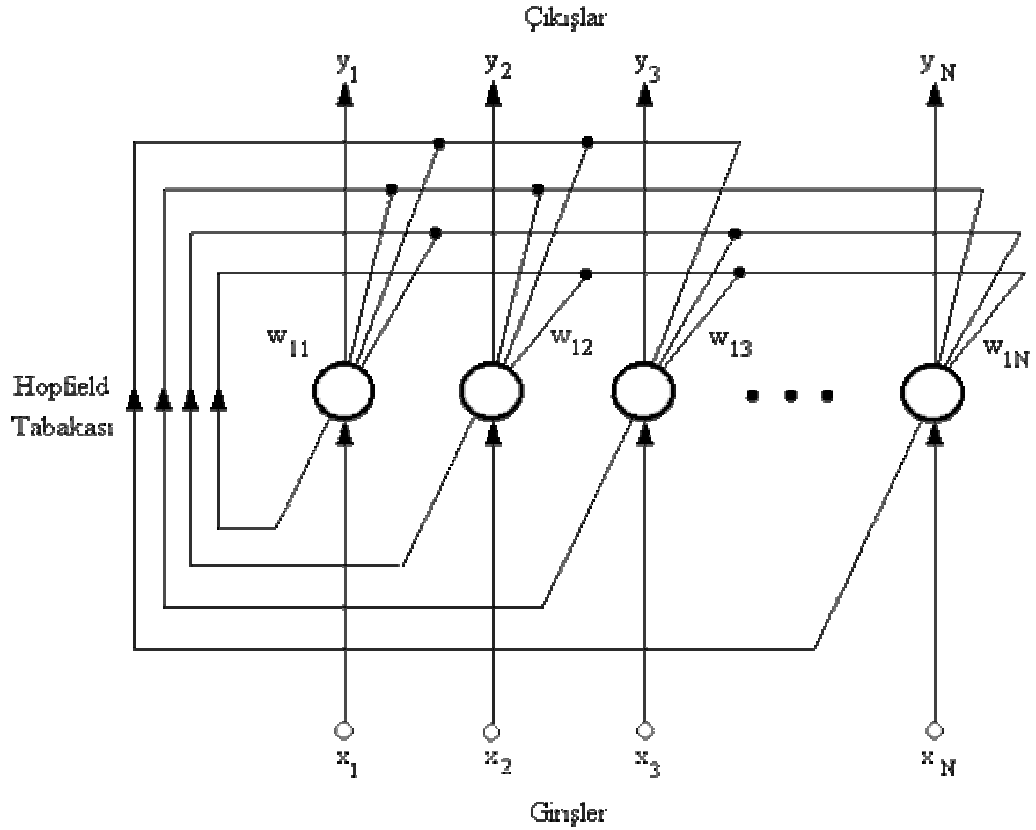
$$y_i(0) = x_i \quad 1 \leq i \leq N \quad (1.28)$$

Bu başlangıç değerleri ile başlayarak Hopfield ağı bir sonraki denklemi kullanarak minimum enerji durumuna geçmek için döngüye girer.

$$y_i(k+1) = f \left[\sum_{j=1}^N w_{ij} y_j(k) \right] \quad 1 \leq i \leq N \quad (1.29)$$

burada f aşağıdaki gibi tanımlanan keskin sınırlama (hard limiting) fonksiyonudur.

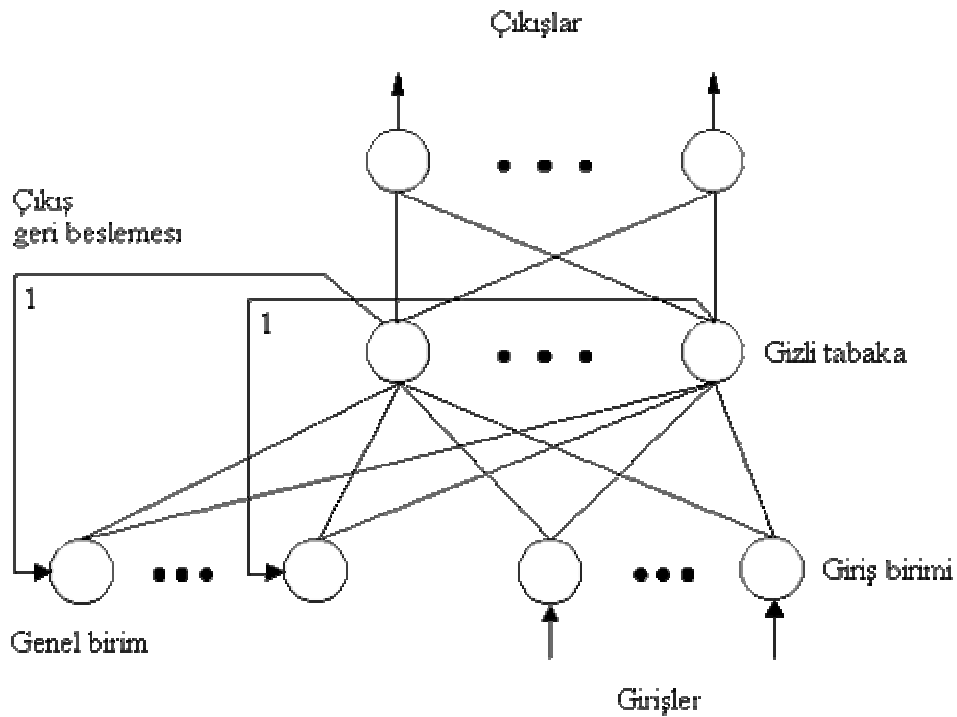
$$f(x) = \begin{cases} -1 & x < 0 \\ 1 & x > 0 \end{cases} \quad (1.30)$$



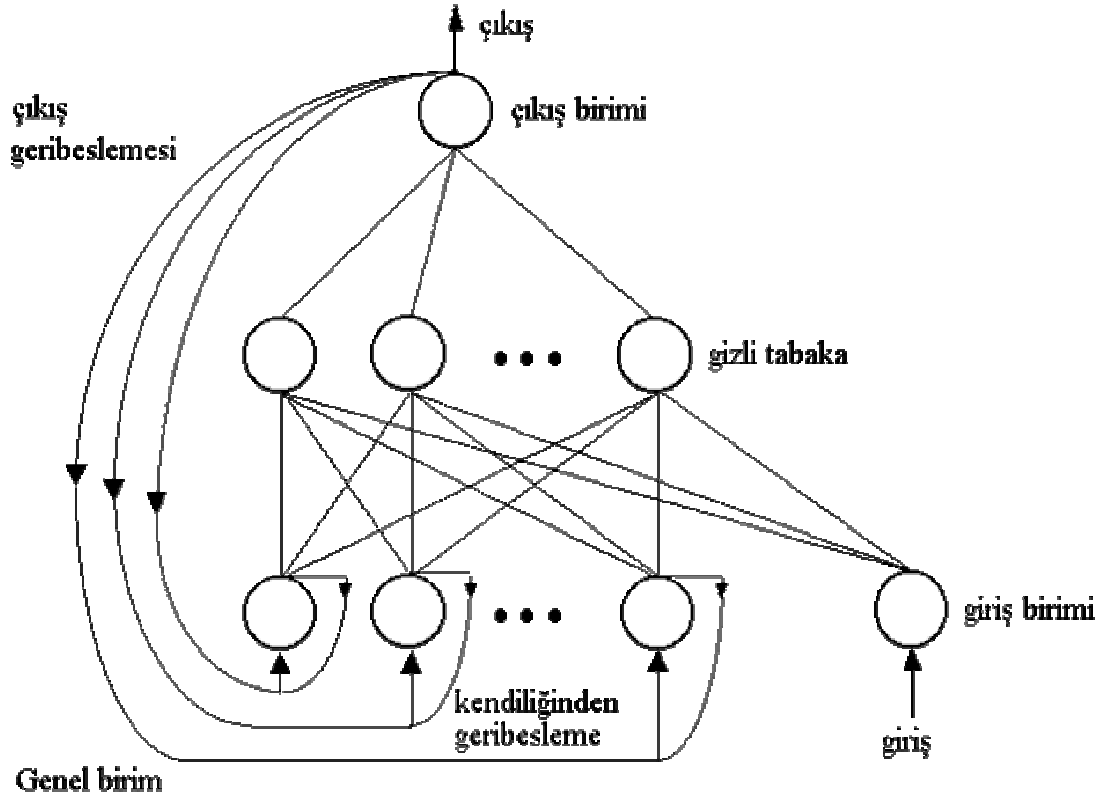
Şekil 1.16: Hopfield ağı

1.4.5.8.3.Elman ve Jordan Ağları

Şekil 1.17a ve b Elman ve Jordan ağlarını göstermektedir. Bu ağlar, MLP' lere benzer bir yapıdadır ve çok katlıdır. Her iki ağda da gizli tabakaya ek olarak diğer bir "durum" tabakası denilen özel bir gizli tabaka daha vardır. Bu tabaka gizli tabakadan veya çıkış tabakasından geri besleme işaretleri alır. Jordan ağının aynı zamanda durum tabakasındaki her işlemci elemandan kendisine bağlantıları vardır. Her iki ağda da durum tabakasındaki işlemci elemanların çıkışları ileriye doğru gizli tabakaya verilmektedir. Eğer sadece ileri doğru bağlantılar göz önüne alınır ve geri besleme bağlantılarına sabit değerler verilirse, bu ağlar sıradan ileri beslemeli ağlar haline gelirler.



(a)



(b)

Şekil 1.17: a) Elman ağı, b) Jordan ağı

1.4.5.8.4.Kohonen Ağı

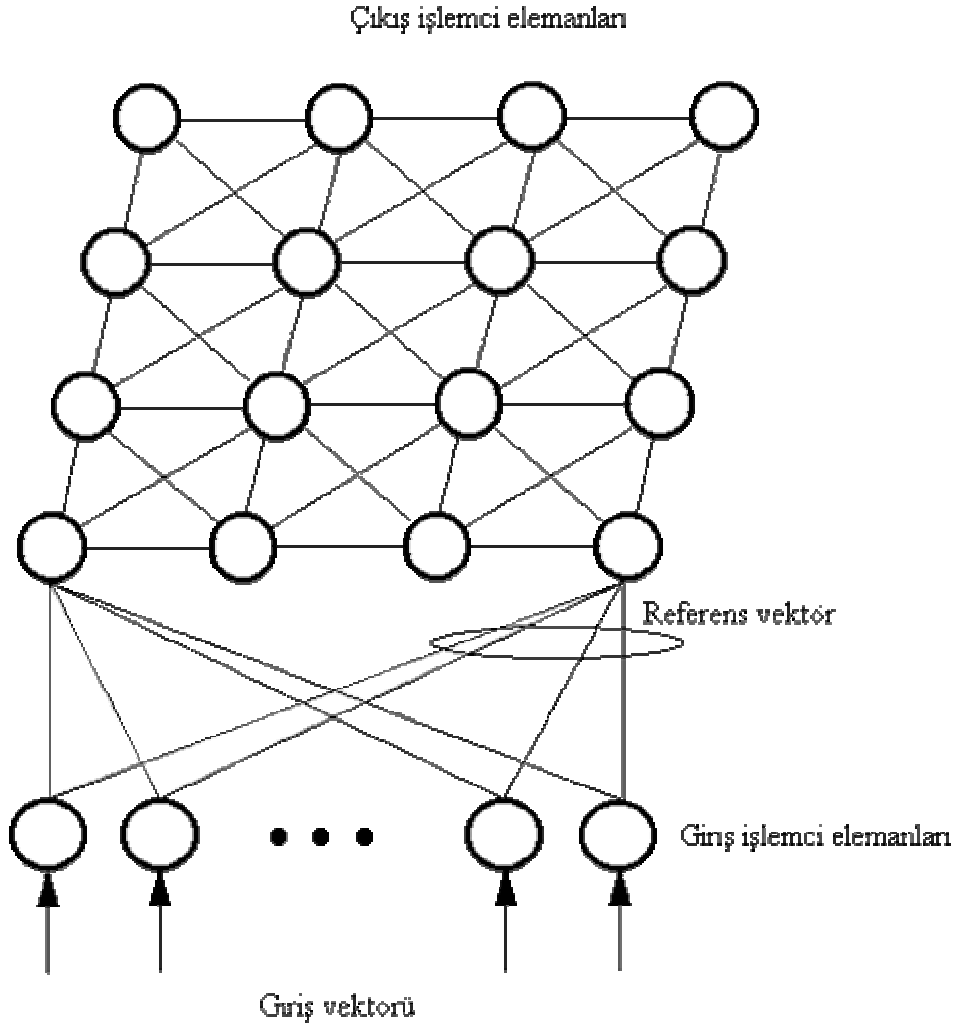
Kohonen ağı, bir giriş tabakası ve bir de çıkış tabakası olmak üzere iki tabakadan oluşur. Bu ağ Şekil 1.18’de gösterilmiştir. Çıkış tabakasındaki işlemci elemanlar genellikle düzenli iki boyutlu aralıklar olarak düzenlenir. Çıkıştaki her İE, bütün giriş işlemci elemanlarına bağlıdır. Bağlantıların ağırlıkları verilen çıkış işlemci elemanı ile ilgili olan referans vektörünün elemanlarını oluşturur. Kohonen ağının öğrenme adımları aşağıdaki gibidir:

- Çıkış İE’ lerin bütün referans vektörlerini küçük rasgele değerlere çek,
- Bir giriş desenini al,
- Kazanan çıkış işlemci elemanını belirle (Bu giriş desenine en yakın referans vektörüne sahip işlemci elemandır. Referans vektörü ile giriş vektörü arasındaki Öklit uzaklığı genellikle uzaklık ölçüsü olarak alınır.),

- Kazanan işlemci elemanın ve onun komşularının referans vektörünü güncelleştir.

Bu referans vektörleri giriş vektörüne yaklaştırılır. Bu yaklaştırma (ayarlama) kazanan işlemci eleman için en fazla ve bu işlemci elemandan uzaklaştıkça daha azdır. Öğrenme ilerledikçe komşuların sayısı azalmakta ve öğrenme sonunda sadece kazanan işlemci elemanın referans vektörü ayarlanmaktadır.

İyi öğrenmiş bir Kohonen ağında birbirine yakın çıkış işlemci elemanlarının referans vektörleri vardır. Öğrenmeden sonra bir etiketleme işlemine başlanır. Bilinen sınıfların giriş desenleri ağına giriş olarak verilir ve bu giriş desenleri tarafından aktif hale gelen çıkış işlemci elemanlarına bu sınıf etiketleri verilir. LVQ ağında ise bir çıkış işlemci elemanı eğer yarışmayı diğer çıkış işlemci elemanlarına karşı kazanırsa bir giriş deseni tarafından aktif hale getirildiği hatırlatılır.



Şekil 1.18: Kohonen ağı

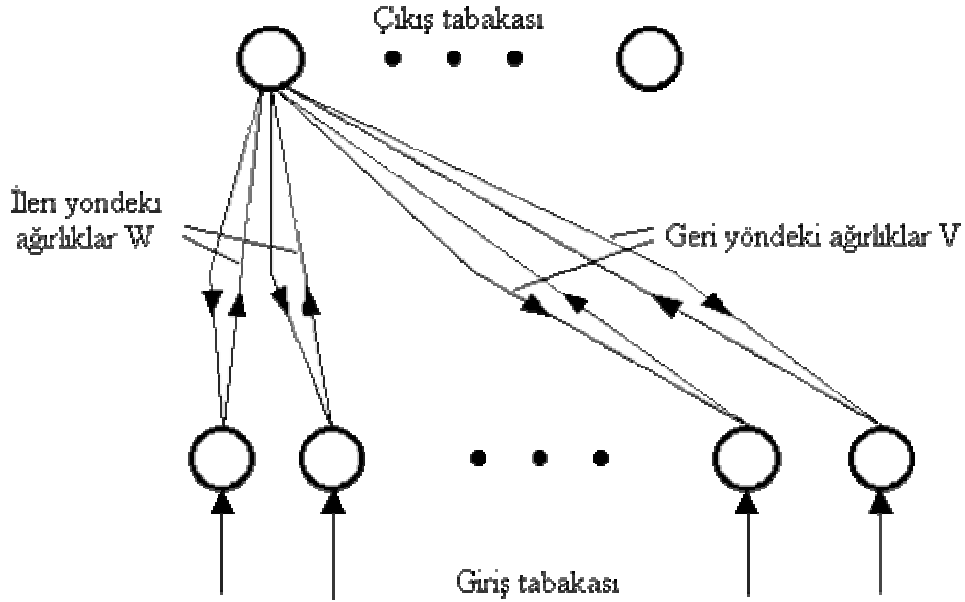
1.4.5.8.5.ART (Adaptive Resonance Theory) Ağı

ART ağının, ART-1, ART-2, Fuzzy-ART, MART gibi değişik çeşitleri vardır. Şekil 1.19’da ikili girişler kabul eden ART-1 gösterilmiştir. ART-2 gerçek değerler kabul etmektedir. Şekil 1.19’de görüldüğü gibi, ART-1 ağının giriş ve çıkış olmak üzere iki tabakası vardır. İki tabaka birbirleriyle tamamen bağlantılıdır ve bağlantılar ileri ve geri yöndedir. i ’inci çıkış işlemci elemanlarının ileri yöndeki bağlantılarının ağırlıklarının oluşturduğu W_i vektörü, temsil ettiği sınıfın bir örneğini oluşturur. İleri bağlantılarının W_i vektörlerinin toplamı ART ağının uzun-dönem (long-term) hafızasını oluşturur. Bu vektörler, giriş desenine benzerlikleri ölçüsünde, çıkış işlemci elemanlarını belirlerler.

Çıkış işlemci elemanlarının geri bağlantılarının W_i vektörleri ise, giriş deseninin, hafızadaki desenine yeterince benzeyip benzemediğini (o sınıfa ait olup olmadığını) kesin olarak belirlemek için bir çeşit test “vigilance” amacıyla kullanılır. Bu test vektörleri diyebileceğimiz V_i vektörleri ART ağının kısa-dönem (short-term) hafızasını oluştururlar. W_i, V_i ’nin normalize edilmiş halidir.

$$W_i = \frac{V_i}{\epsilon + \sum V_{ji}} \quad (1.31)$$

burada ϵ küçük bir sabittir ve V_{ji} , V_i ’nin j ’inci elemanıdır (yani i ’inci İE’den j ’inci giriş İE’ye bağlantı ağırlığıdır).



Şekil 1.19: ART-1 ağı

1.4.5.9.YSA' nın Mühendislik Uygulamaları

YSA' lar birçok önemli mühendislik problemlerin çözümünde kullanılmışlardır. Aşağıda, kimya mühendisliği, inşaat ve yapı mühendisliği, elektrik ve elektronik mühendisliği, imalat ve makine mühendisliği, sistem ve kontrol mühendisliği alanlarında bulunan uygulamalarından bazıları aşağıda.

Kimya Mühendisliği; kimyasal reaktör seçimi, dinamik işlemlerde hata belirlenmesi, endüstriyel polimerisasyonda eritme akış indisi tahmini, endüstriyel mayalama işleminin modellenmesi, biyokimyasal işlemlerde mikrobik konsantrasyonların tahmininde uygulanmıştır.

İnşaat ve Yapı Mühendisliği; konstrüksiyon projelerinde kaynak seviyelerini belirleme, bir rezervuardan çıkışı kontrol, biyolojik bilgiler yardımıyla nehir suyu kalitesinin sınıflandırılması, nehirlerin akışının tahmin edilmesi, sonlu-eleman-temelli yapısal analiz işleminin modellenmesi, yapı malzemelerinin iç yapılarındaki çatlakların tespit edilmesi, depreme maruz betonarme çerçevelerde emniyetli yatay taşıyıcı tahmininde uygulanmıştır.

Elektrik ve Elektronik Mühendisliği; hastaların alarmla kontrolü, gürültülü resimlerin kalitesini artırma, görüntülerin sıkıştırılması, gürültü filtreleme ve resimlerdeki kenar bilgisinin çıkarılması, güç sisteminde harmoniklerin tahmini, gezgin haberleşme sisteminde kanal dağıtımı, ultrasonik müziklerden objelerin sınıflandırılması, optik okuyucu sistemler için resimlerin ön işleme alanlarında kullanılmıştır.

İmalat ve Makine Mühendisliği; metal kesme tezgahının kontrolü, akustik salınım ve iş parçası kuvvetiyle iş parçası yatağının kontrolü ve güç tüketimi ve iş parçası ivmesi, hücreli imalat için grup teknolojisi parça gruplarının tasarımı, hareket eden nesneler için engelsiz yol planlaması, makine parametrelerinin optimizasyonu, makine arızalarının sınıflandırılması, malzemelerin ısı transferinin belirlenmesi, uçak kanat kutularının yapısının tasarımında uygulanmıştır.

Sistem ve Kontrol Mühendisliği; esnek kollu robotun kontrolü, bir model helikopterin havada kontrolü, çok değişkenli (mafsallı) robotun yörünge koordinasyonu, iki sıvı tank

sisteminin akış seviye kontrolü, anestezi derinliği kontrolü ve ölçülmesi, bir robot için optimal yolun bulunması, banyo suyu sıcaklığının kontrolü, endüstriyel robot kontrolü, sistem kimliklendirme gibi birçok alanda uygulanmıştır.

Anten ve Uygulamaları; yapay sinir ağlarında bir dinamik öğrenme algoritması (DÖA) kullanılarak anten dizi elemanlarından elde edilen işaretler arasındaki faz farklılıklarının karşılaştırılmasıyla radar izleme gerçekleştirilmiştir. Mikrodalga parlaklığına bağlı jeofiziksel parametrelerin tayini, DÖA sinir ağı kullanılarak gerçekleştirilmiştir. Uzaktan kontrollü görüntü sınıflama işlemi, danışmanlı DÖA ve danışmansız (ART2) öğrenme algoritmaları kullanılarak yapılmıştır. YSA' ları kullanılarak anten dizi analizi yapılmıştır. Aynı zamanda, anten dizi tasarımında, genetik algoritma ve tabu araştırma algoritması kullanılmıştır. Osilatör YSA ile elektromagnetik işaretlerin etkileşim modelleri oluşturulmuştur.

1.4.6.Bulanık Mantık

1.4.6.1.Tarihçe

Mantıksal paradokslar ve Heisenberg' in belirsizlik ilkesi, 1920'ler ve 1930'larda çok değerli mantık sistemlerinin gelişmesine yol açtı. Kuantum teorisyenleri, iki değerli mantık sistemlerinin 'doğru' ve 'yanlış'tan oluşan değer kümesine, bir üçüncü veya orta doğruluk değeri ekleyerek 'belirlenemezlik' in ifade edilebilmesine imkan sağladılar. Bundan sonraki aşamada, 'doğru' ve 'yanlış', 'belirlenemezlik' tayfının sınır koşulları olarak görülüp belirlenemezlik derecelendirildi.

Heisenberg' in belirsizlik ilkesi, 'belirlenemezlik' inin sürekliliğiyle, bilimi çok değerliliğe zorladı. Pek az batılı filozof çok değerliliği benimsemesine rağmen, Lukasiewicz, Gödel, ve Black, ilk çok-değerli ya da bulanık mantık ve küme sistemlerini geliştirdiler.

1930'ların başlarında Polonyalı mantıkçı Jan Łukasiewicz ilk üç-değerli mantık sistemini geliştirdi. Łukasiewicz, daha sonra doğruluk değerlerinin kümesini tüm sayılara genelleştirdi.

1930'larda kuantum filozofu Max Black, sürekli değerlere sahip mantığı, eleman düzeyinde kümelerle uyguladı. Black, bulanık-küme üyelik fonksiyonlarından bahseden ilk kişi oldu. Black, ifade etmeye çalıştığı yapılardaki belirsizliği 'müphemlik' olarak adlandırdı. Zadeh'in bulanık-küme teorisinin aksine, Black' in çok değerli kümelerindeki her bir eleman, sürekli değerlere sahip bir mantık çerçevesinde ele alınan bir cümleyle eş-değerdi.

1965'te Azeri kökenli sistem bilimci Prof. Lotfi A. Zadeh, bir çok değerli küme teorisi geliştirdi, ve 'bulanık' kelimesini teknik terimlere dahil etti. California Berkeley Üniversitesinden olan Prof. Lotfi A.Zadeh bu konu üzerinde ilk makalelerinin yayımlanmasıyla tanınmıştır. Bulanık mantık, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için kurulmuş katı bir matematiksel düzen olarak tanımlanabilir. Bilindiği gibi istatistikte ve olasılık kuramında, belirsizliklerle değil kesinliklerle çalışılır ama insanın yaşadığı ortam daha çok belirsizliklerle doludur. Bu yüzden insanoğlunun sonuç çıkarabilme yeteneğini anlayabilmek için belirsizliklerle çalışmak gereklidir.

1.4.6.2.Bulanık Mantık Nedir?

"Fuzzy" kuramının merkez kavramı "fuzzy" kümeleridir. Küme kavramı kulağa biraz matematiksel gelebilir ama anlaşılması kolaydır. Örneğin "orta yaş" kavramını inceleyecek olursak, bu kavramın sınırlarının kişiden kişiye değişiklik gösterdiğini görürüz. Kesin sınırlar söz konusu olmadığı için kavramı matematiksel olarak da kolayca formüle edemeyiz. Ama genel olarak 35 ile 55 yaşları orta yaşlılık sınırları olarak düşünülebilir.

Bulanık mantık ile matematik arasındaki temel fark bilinen anlamda matematiğin sadece aşırı uç değerlerine izin vermesidir. Klasik matematiksel yöntemlerle karmaşık sistemleri modellemek ve kontrol etmek işte bu yüzden zordur, çünkü veriler tam olmalıdır. Bulanık mantık kişiyi bu zorunluluktan kurtarır ve daha

niteliksel bir tanımlama olanağı sağlar. Bir kişi için 38,5 yaşında demektense sadece orta yaşlı demek bir çok uygulama için yeterli bir veridir. Böylece dikkate değer ölçüde bir bilgi indirgenmesi söz konusu olacak ve matematiksel bir tanımlama yerine daha kolay anlaşılabilen niteliksel bir tanımlama yapılabilecektir.

Bulanık mantıkta fuzzy kümeleri kadar önemli bir diğer kavramda linguistik değişken kavramıdır. Linguistik değişken “sıcak” veya “soğuk” gibi kelimeler ve ifadelerle tanımlanabilen değişkenlerdir. Bir linguistik değişkenin değerleri fuzzy kümeleri ile ifade edilir. Örneğin oda sıcaklığı linguistik değişken için “sıcak”, “soğuk” ve “çok sıcak” ifadelerini alabilir. Bu üç ifadenin her biri ayrı ayrı fuzzy kümeleri ile modellenir.

Bulanık mantığın uygulama alanları çok geniştir. Sağladığı en büyük fayda ise “insana özgü tecrübe ile öğrenme” olayının kolayca modellenebilmesi ve belirsiz kavramların bile matematiksel olarak ifade edilebilmesine olanak tanınmasıdır. Bu nedenle lineer olmayan sistemlere yaklaşım yapabilmek için özellikle uygundur.

Bilgisayar, bilgiyi yalnızca "doğru" ya da "yanlış" olarak algılayıp, bir ve sıfırdan oluşan dizgeler halinde işlem yapar. İnsansa, bilgisayarın tersine, kısmi doğrular ya da yanlışlar üzerinden, duyularını ve tecrübelerini kullanarak işlemi gerçekleştirir. Bilgisayar için sıcak ya da soğuk vardır ama insan için soğuk, serin, normal, ılık ya da sıcak olabilir. Bilgisayarın sıcak ve soğuk içinde serini de tanımlayabilmesi için bulanık mantık (fuzzy logic) kullanılır.

Bulanık mantık konusunda yapılan araştırmalar Japonya’da oldukça fazladır. Özellikle fuzzy process controller olarak isimlendirilen özel amaçlı bulanık mantık mikroişlemcinin üretilmesine çalışılmaktadır. Bu teknoloji fotoğraf makineleri, çamaşır makineleri, klimalar ve otomatik iletim hatları gibi uygulamalarda kullanılmaktadır. Bundan başka uzay araştırmaları ve havacılık endüstrisinde de kullanılmaktadır. TAI’ de araştırma gelişme kısmında bulanık mantık konusunda çalışmalar yapılmaktadır. Yine bir başka uygulama olarak otomatik civatalamaların değerlendirilmesinde bulanık mantık kullanılmaktadır. Bulanık mantık yardımıyla civatalama kalitesi belirlenmekte,

civatalama tekniği alanında bilgili olmayan kişiler açısından konu şeffaf hale getirilmektedir. Burada bir uzmanın değerlendirme sınırlarına erişilmekte ve hatta geçilmektedir.

Bulanık mantık, adından anlaşılabilceği gibi mantık kurallarının esnek ve bulanık bir şekilde uygulanmasıdır. Klasik (boolean) mantıkta bildiğiniz gibi, "doğru" ve "yanlış" yada "1" ve "0"lar vardır, oysa bulanık mantıkta, ikisinin arasında bir yerde olan önermeler ve ifadeler izin verilebilir ki, gerçek hayata baktığımızda hemen hemen hiçbir şey kesinlikle doğru veya kesinlikle yanlış değildir. Gerçek hayatta önermeler genelde kısmen doğru veya belli bir olasılıkla doğru şeklinde değerlendirilir. Bulanık mantığa da zaten klasik mantığın gerçek dünya problemleri için yeterli olmadığı durumlar dolayısıyla ihtiyaç duyulmuştur.

Bulanık mantığın sistemi şu şekildedir. Bir ifade tamamen yanlış ise klasik mantıkta olduğu gibi 0 değerindedir, yok eğer tamamen doğru ise 1 değerindedir. (Ancak bulanık mantık uygulamalarının çoğu bir ifadenin 0 veya 1 değerini almasına izin vermezler, veya sadece çok özel durumlarda izin verirler.) Bunların dışında tüm ifadeler 0 dan büyük 1 den küçük reel değerler alırlar. Yani değeri 0.32 olan bir ifadenin anlamı %32 doğru %68 yanlış demektir.

Bulanık mantığın da klasik mantıkta olduğu gibi işleçleri (operatör) vardır, örneğin and, or, not ... ancak bunlar kendine has işlemlerdir mesela -başka yaklaşımlarda olmasına rağmen and işlemi- genelde çarpma olarak ifade edilir veya not işlemi de birden çıkarma şeklinde ifade edilir. Bunlar;

$$\text{AND: } A=0.2 \text{ } B=0.8 \Rightarrow A \text{ and } B = (A) * (B) = 0.2 * 0.8 = 0.4$$

$$\text{NOT: } A=0.4 \Rightarrow \text{not } A = 1-(A) = 1 - 0.4 = 0.6$$

şeklinde örneklenebilir. Ancak bunlar en basit yaklaşımlardır.YSA ve Fuzzy Logic tekniklerinin beraber kullanımı ile daha etkili sistemler dizayn etmek mümkündür, ancak bu işlem ortaya çıkan sistemi çok yavaşlatmaktadır ve henüz bu tekniklerin birleştirilmesi yöntemi geliştirme ve test aşamalarında, aslında YSA algoritmaları da her gün hızla güncellenmektedir. Yani bu konuların (mesela özyineleme ya da search gibi) tam olarak oturdukları söylenemez, fakat başarılı uygulamaları da mevcuttur.

Bulanık mantık, bilgisayarın insan gibi davranmasını ve "akıllı" olmasını sağlar. BULANIK MANTIK, bir tür makine zekasıdır. Bilgisayar bu zeka sayesinde tam olarak tanımlanmamış sistemleri sağduyulu bir yaklaşımla anlayıp kontrol eder. Bilgisayar, çevresindeki olayları sensörler kullanarak algılar. Bu bilgileri zekasıyla yorumlayıp bir sonuca ulaşır ve kontrol hareketi üretir. "Zeki" olmayan bir çamaşır makinesi, içindeki çamaşırın ne kadar kirli olduğuna bakmasızın yıkama işlemini daha önce programlandığı gibi sabit bir şekilde yapar. Öte yandan akıllı bir çamaşır makinesiye, yıkama suyunu inceleyerek çamaşırların temizlenme durumuna bakar. Makine, "Eğer çamaşır çok kirliyse daha uzun yıka, az çamaşır varsa az su kullan" gibi önerme cümlelerine göre kontrol yapar. Elde ettiği sonuçlara göre yıkama programını değiştirir. Böylece, her zaman aynı şekilde yıkama yapan "zeki olamayan" bir makine, bir insan gibi düşünüp yıkama şeklini çamaşıra göre belirleyen zeki bir makineye dönüşmüştür. Bu zeka, bulanık mantığın, çamaşır makinesini kontrol eden birimde kullanılmasıyla elde edilmiştir. Bulanık mantığın temelinde belirsizlik (vagueness) yatar. Bertrant Russell, yalnızca bir ve sıfırdan oluşan iki değerli mantığa karşı belirsizlik içeren ve $[0,1]$ aralığında tanımlı olan mantık üzerine araştırma yapmıştır. Russell, bunun için eski çağlardan gelen yalancı paradoksunu kullanır:

Giritli bir adam, tüm Giritlilerin yalancı olduğunu söylemektedir. Eğer Giritli yalan söylüyorsa doğru, doğru söylüyorsa yalan söylüyordur. İki değerli mantığa göre bu olanaksızdır oysa ki belirsizlik mantığına göre Giritli %50 doğru, %50 yalan söylüyordur. Belirsizlik mantığı, bu paradoksu "yarım doğru" olarak kabul eder. Bu $[0,1]$ kapalı aralığındaki orta noktadır. $[0,1]$ kapalı aralığında tanımlanmış problem artık bulanık mantık problemi olarak adlandırılır. "Giritli paradoksu" tek boyutlu bir bulanık mantık problemidir.

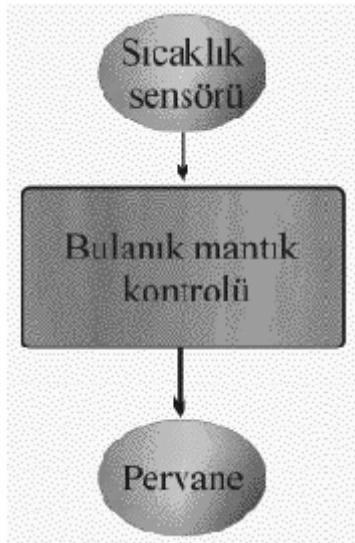
Bulanık mantık problemleri birçok boyuta sahip olabilir. İki boyutlu bulanık mantık problemi için Sokrates ve Plato'nun yalancı paradoksu kullanılır:

Sokrates: Plato'nun söyleyeceği şey doğrudur.

Plato: Sokrates yalan söylüyor!

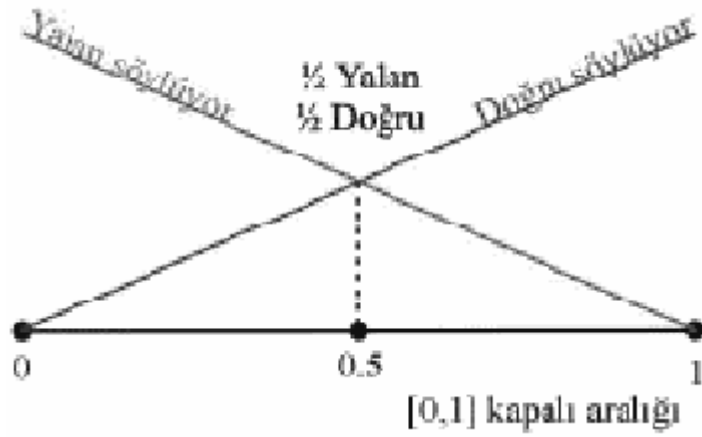
Paradoksta, Sokrates'in doğru söylemesi, Plato'nun yalancı olmasını bu da Sokrates'in yalancı olmasını gerektirir. Aynı şekilde Plato'nun doğru söylemesi, Sokrates'in doğru,

Plato'nun yalan söylemesidir. Paradoks, bulanık mantık kullanılarak iki boyutlu bir probleme dönüştürülür. Problem, ayrıtları bir birim olan kare düzlem içinde tanımlıdır. Karenin sola alt köşesi hem Sokrates'in hem de Plato'nun yalan söylediğini gösteren (0,0) noktasını, sağ üst köşesiye her ikisinin de doğru söylediğini gösterir. Problemin paradoks halindeki biçimi (0.5, 0.5) noktasında dengede bulunmaktadır. Bu nokta, en "bulanık" noktadır. Nokta, "hem Plato hem de Sokrates %50 doğrudur" önermesini yapar. Karenin köşe noktalarıysa problemin "netlik" noktalarıdır. Aynı şekilde, kare düzlemi üzerindeki (0.25,0.33) noktası, Sokrates'in %25 doğru, Plato'nun %33 doğru söylediğini gösterir. Paradoks eğer iki değerli mantık üzerinde tanımlansaydı, paradoksta yalnızca (0,0), (1,0), (0,1) ve (1,1) noktaları olacak ve paradoksun bir denge noktası bulunmayacaktı. Oysa ki bulanık mantık sayesinde paradoks bir birim düzlem üzerinde tanımlanmış ve orta noktasında dengede duran probleme dönüşmüştür. Bulanık mantık problemlerin tanım boyutu arttıkça, problem 1x1 birimlik kareye 1x1x1 birimlik küpe, ya da daha üst boyutlu şekillere dönüşür.



Bulanık mantık kullanan sistemler sensörlerden gelen bilgileri işleyerek kontrol hareketlerini yaparlar. Yandaki şekil, oda sıcaklığını kontrol eden bulanık mantık birimidir. Oda sıcaklığı, soğutucu motorunun pervane hızıyla kontrol edilmektedir.

Şekil 1.20

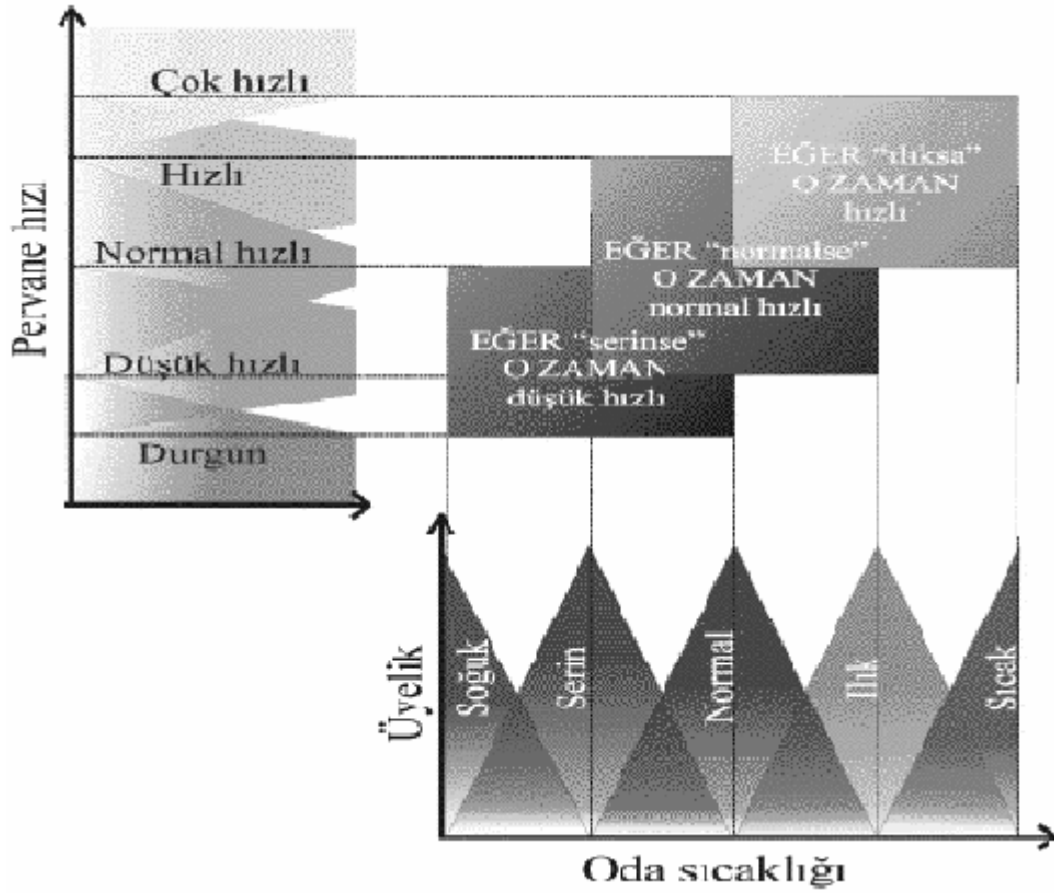


Şekil 1.21

Eğer Giritli yalan söylüyorsa doğru, doğru söylüyorsa yalan söylüyordur. İki değerli mantığa göre bu olanaksızdır oysa ki belirsizlik mantığına göre Giritli %50 doğru, %50 yalan söylüyordur. Belirsizlik mantığı, bu paradoksu "yarım doğru" olarak kabul eder. Bu $[0,1]$ kapalı aralığındaki orta noktadır. Üstteki şekilde $[0,1]$ aralığı ve Giritli'nin herhangi bir noktada ne kadar doğru ve yanlış olduğu gösterilmiştir.

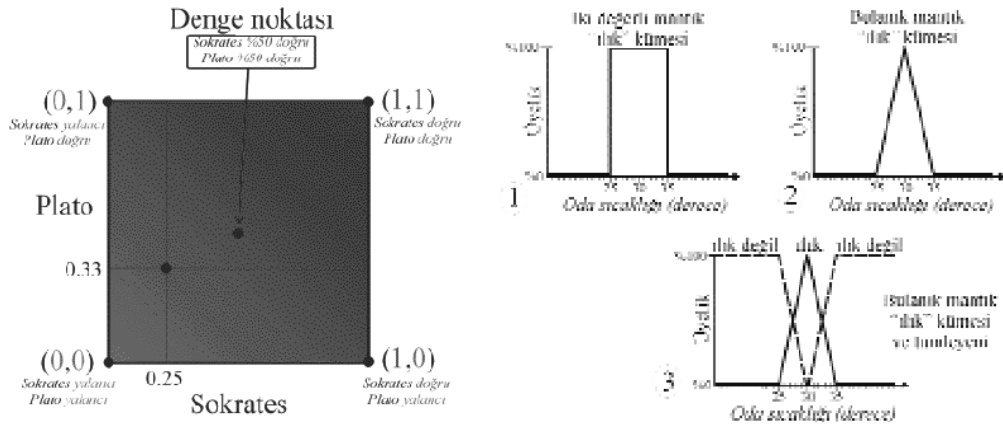
Bulanık mantığın kontrol sistemlerine uygulanmasında Giritlinin, Plato ve Sokrates'in yarattıkları paradokslar yerine tecrübeye dayanan sözel kurallar kullanılır. "Eğer oda ılıksa pervaneyi hızlı döndür" kuralında, odanın ılık olması $[0,1]$ kapalı aralığında tanımlıdır. Odanın %60 ılık olması, %40 ılık olmaması anlamına gelir. Eğer yalnızca iki değerli mantık kullanılsaydı, oda ya ılık olacak ya da ılık olmayacaktı. Ancak bulanık mantık sayesinde %100 ılık olan oda aynı zamanda %0 ılık değildir. Tecrübeyeyle oluşturulan sözel kuralların her biri bulanık mantık kümelerini oluşturur. Ilık kümesi, oda sıcaklığına göre yüzde cinsinden odanın ne kadar ılık ve ne kadar ılık olmadığını gösterir. Bu kümelerin oda sıcaklığına göre değişimleri üyelik işlevi (membership function) olarak tanımlanır. Üyelik işlevleri soğuk, serin, normal, ılık ve sıcakın birbirlerine göre durumlarını grafiksel olarak gösterir. Buna göre %30 serin olan oda %70 normal sıcaklıktadır. Üyelik işlevleriyle belirlenmiş giriş kuralları çıkıştaki kontrol hareketlerini belirler. Bu kurallar EĞER ...O ZAMAN ... kalıbında yazılır. Buna göre oda sıcaklığını kontrol edecek bir sistemin kuralları şu şekilde belirlenir :

EĞER oda soğuksa O ZAMAN pervaneyi durdur. EĞER oda serinse O ZAMAN pervaneyi düşük hızla döndür.



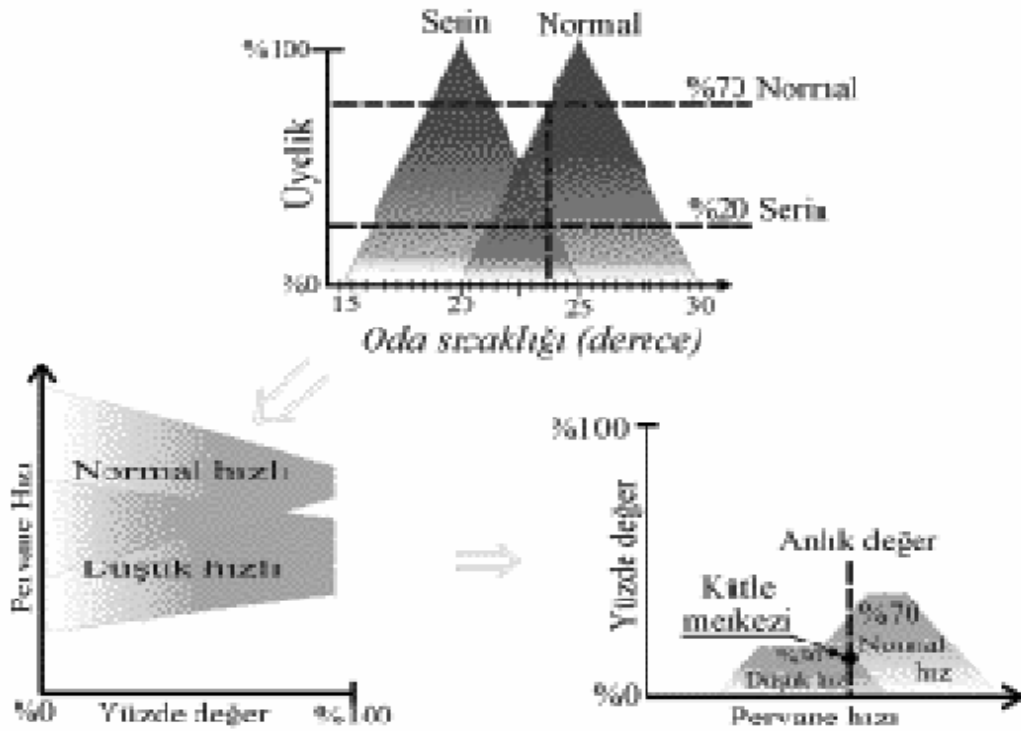
Şekil 1.22

Tecrübeyle oluşturulan sözel kuralların her biri bulanık mantık kümelerini oluşturur. "Ilık" kümesi, oda sıcaklığına göre yüzde cinsinden odanın ne kadar ılık ne kadar normal ya da sıcak olduğunu gösterir. Bu kümelerin oda sıcaklığına göre değişimleri üyelik işlevi olarak tanımlanır. Üyelik işlevleri soğuk, serin, normal, ılık ve sıcakın birbirlerine göre durumlarını grafiksel olarak gösterir. Üyelik işlevleri, kontrol hareketlerini de belirler. Pervanenin hızı, ilgili üyelik kümesine göre belirlenir. Üstteki şekilde üyelik işlevleri ve çıkış hareketleri gösterilmiştir.



Şekil 1.23

Plato ve Sokrates'in yarattığı paradoks, iki boyutlu belirsizlik düzlemi üzerinde tanımlanır. Üstteki şekilde paradoksun denge noktası gösterilmiştir. Kırmızı renk yalanı, maviyse doğruyu belirtir. 1. şekilde iki değerli mantığa göre "ılık" tanımı yapılmıştır. Buna göre oda ya ılık ya da ılık değildir. Eğer bu durum bulanık mantığa göre incelenirse, 2. şekil elde edilir. Bulanık mantık kümelerinin tümlenmesiyle 3. şekilde gösterilmiştir.



Şekil 1.24

Eğer anlık oda sıcaklığı 20°C olursa, bu durum %30 serin, %70 normaldir. Bu iki kümenin o sıcaklık noktasındaki yüzde değeri, motor hızının hesaplanmasında kullanılır. Motor hızını belirleyen bulanık mantık kümeleri elde edilen yüzdelerle çarpılır. Motor, %30 düşük hızlı, %70 normal hızda döndürülmelidir. Motorun hızını tanımlayan bu iki şeklin kütle merkezi motorun hızını belirler.

EĞER oda normalse O ZAMAN pervaneyi normal hızda döndür. EĞER oda ılıksa O ZAMAN pervaneyi hızlı döndür. EĞER oda sıcaksa O ZAMAN pervaneyi çok hızlı döndür. Deneyimler doğrultusunda normal oda sıcaklığı 25°C olarak belirlensin. Eğer anlık oda sıcaklığı 20° olursa, bu durum %30 serin, %70 normaldir. Bu iki kümenin o sıcaklık noktasındaki yüzde değeri, motor hızının hesaplanmasında kullanılır. Motor hızını belirleyen bulanık mantık kümeleri elde edilen yüzdelerle çarpılır. Motor, %30 düşük hızlı, %70 normal hızda döndürülmelidir. Motorun hızını tanımlayan bu iki şeklin kütle merkezi motorun hızını belirler. Motor hızının hesaplanmasına "netleştirme" (defuzzing) denir. Bulanık mantığın temelini, Bertrand Russell atmıştır. Öte yandan Polonyalı Jan Lukasiewicz, belirsizlik mantığının temel işlemlerini tanımlamıştır. 1930'lu yıllarda kuantum fizikçisi Max Planck tarafından uygulanan belirsizlik mantığı, 1958 yılında Kaplan ve Schott'un yaptığı eklemelerle gelişmiştir. 1965 yılında, Berkeley'de öğretim üyesi olan Lotfi Zadeh, tüm bu süreç içinde bir devrim yaratmış ve belirsizlik mantığını bulanık mantık olarak adlandırıp yeniden şekillendirmiştir. Bulanık mantık, bir buhar makinesini kontrol edebilmek amacıyla 1970 yılında İbrahim Mamdani tarafından kullanılmış ve bulanık mantık mühendisliğinin tanımlanmasını sağlamıştır. Daha sonra Japon firmalarca geliştirilen bulanık mantık sistemleri, ilk kez Japonya'nın Sendai şehri metrosunun kontrolünde başarıyla kullanılmıştır. Bulanık mantığı temel alan kontrol birimlerinin kullanılması, içinde bulanık mantık komutları bulunan mikro işlemcilerin üretilmesiyle her geçen yıl artmaktadır. Alışıl gelmiş kontrol sistemlerinde, sistemin matematiksel modelinin oluşturulması gerekir. Özellikle doğrusal olmayan sistemlerde bu işlem oldukça zor ve zaman alıcıdır. Arkasında römorku bulunan bir taşıyıcı robotun geri manevra yaparak deponun istenilen bölgesine girebilmesini sağlayan bir kontrol biriminin yapımı, klasik kontrol yöntemleriyle olanaksızdır. Burada, sistemin modelinin çıkartılması oldukça güç, hatta değişik başlangıç noktaları yüzünden olanaksızdır. Bulanık mantık temelli kontrolde, sistemin modelinin bilinmesine gerek yoktur. Sistemin yapılması deneyimlere ve sözel kurallara bağlıdır. Bu sayede taşıyıcı robotun römorkuyla geri manevra yapabilmesi, deneyimli bir sürücünün deneyimlerinden yola çıkarak elde edilen sözel kurallara göre kolayca yapılabilir. Günümüzde bulanık mantık sistemleri çamaşır makinesinden video kameralara, görüntü işlemeden ülkelerin ekonomik modellerinin çıkartılmasına kadar birçok alanda kullanılır. Bulanık mantık

sistemlerinde kullanılan sözel deneyimler kimi sistemlerde yetersiz kalır. Öte yandan, kontrol sisteminin ince ayarının üyelik işlevlerinin katsayılarının ayarlanmasıyla yapılması gerekir. Bu işlem deneme yanılma yoluyla yapılabilir ama bu yöntem karmaşık sistemlerde oldukça zaman alan bir iş haline dönüşür. Son yıllarda yapay sinir ağları kullanarak kendi kendine öğrenen bulanık mantık sistemleri kullanılmaya başlanmıştır.

1.4.7.Sanal Gerçeklik

Sanal gerçeklik bilgisayar ortamında oluşturulan bir gerçekliktir ve “cyberspace” olarak da bilinir. Yapay zekanın bu alanında doğal gerçekliğe uygun, insan/bilgisayar arabirimlerinin kullanıldığı bir ortam oluşturulur. Sanal gerçeklik, gözlük ve stereo kulaklıktan oluşan başlık seti, vücut hareketlerini algılayan özel bir giysi veya eldivenden oluşan, çok algılayıcı giriş-çıkış cihazlarına bağlı olarak oluşturulmaktadır. Böylelikle üç boyutlu sanal dünyayı görebilir ve dokunabilirsiniz. Sanal gerçeklik sizin bilgisayar benzetimli nesneler ve varlıklar ile etkileşim içine girebilmenize olanak sağlamaktadır.

Sanal gerçeklik uygulamaları geniş bir alana yayılmıştır. Bilgisayar destekli tasarımda (CAD), tıbbi teşhis ve tedavide, fiziksel ve biyoloji bilimlerindeki bilimsel deneyimlerde, pilot ve astronotların eğitimi için uçuş simülatörlerinde ve eğlence olarak üç boyutlu video oyunlarında kullanılmaktadır. CAD en geniş şekliyle endüstriyel sanal gerçeklik uygulamalarında kullanılmaktadır. Mimarlar ve tasarımcılar, ürünlerin ve yapıların üç boyutlu modelleri üzerinde test ve tasarım işlemleri yapmakta kullanırlar. Bu teknoloji ayrıca ecza ve biyoteknoloji firmaları tarafından yeni ilaçların bilgisayarda tanımlanmış davranışlarını geliştirmek ve gözlemlemek için kullanılmaktadır. Ayrıca doktorlar hasta vücudunun sanal bir modelinin oluşturulup sorgulanmasında faydalanmaktadır.

Sanal gerçeklik uygulaması ile ilgili aşağıda iki örnek verilmiştir:

1998 yılında kullanıma açılacak olan Paris yakınlarındaki stadyum, IBM Fransa tarafından yapımından önce sanal olarak inşa edilmiştir. Amaç tasarım aşamasında

insan akınlarını ve onların davranışlarını analiz etmektir. Ayrıca sağlık ve güvenlik kuruluşlarını ihtiyaç duyulan yerlere yerleştirmek ve ziyaretçilere mümkün olduğunca konfor ve hareket serbestliği sağlayabilmektir. Bunların yanı sıra müdahale olanaklarını ve etkilerini daha iyi tahmin etme imkanı olmaktadır. Gelecekte bu simülasyonun, havaalanları, resmi binalar ve alışveriş merkezlerinin tasarımında kullanılacağı belirtilmektedir.

Almanya Fraunhofer Enstitüsünde, yolcuların uçuş korkusunu yenebilecekleri, yolculara yönelik ilk uçuş simülatörü gerçekleştirilmiştir. Bu proje, sanal gerçeklikle psikolojinin, fobilerin tedavisi için ilişkilendirilmesi fikrinden doğmuştur. Sanal ortama, yürüyen bir bant üzerindeymişcesine pencerelerin yanından geçip hafif eğimli olan kapıya vararak giriyorsunuz. Uçağa biniyor, doğru yeri buluyor ve oturuyorsunuz. Klima çalışıyor ve hoparlörlerden müzik sesi geliyor. Hafif bir sarsıntıyla uçak kapıdan ayrılıyor ve piste doğru yol alıyor. Makinelere uğultulu bir ses geliyor, ivme sizi koltuğa bastırıyor ve Take-off. Yolcu, uçuşu, sanal gerçeklik kaskı ve kulaklık vasıtasıyla yaşıyor, gerçek uçuş duygusunu ise podestinin altındaki performansı yüksek elektromotorlar sağlıyor. Uçuş deneyine katılan yolcuların sorgulanması ile birlikte yaklaşık kırk dakika kadar sürüyor.

2.UZMAN SİSTEMLER

2.1.Giriş

Günümüzün bilgisayarları muazzam işlem yapma kapasitesine sahip olmasına karşın, ne yazık ki öğrenme becerisine sahip değildir. Yapay zeka araştırmalarının bir kolu olan bilgi tabanlı sistemler bu olguyu değiştirmeye çalışmaktadır. YZ araştırmacıları bilgi tabanlı sistemlere iki temel beceriyi kazandırmayı amaçlamaktadır:

- İnsan muhakeme sistemini taklit edebilme,
- Öğrenebilme.

Uygulamada, "uzman sistemler" ve "bilgi-tabanlı sistemler" aynı anlamda kullanılan terimlerdir. Teknik açıdan bakıldığında ise, uzman sistem bir bilgi-tabanlı sistemin en gelişmiş biçimidir. Bir uzman sistem sorulara cevap veren, açıklık getirmek için soru soran, tavsiyelerde bulunan ve karar verme sürecine yardımcı olan diyaloga açık bir sistemdir. Daha az gelişmiş bilgi-tabanlı sistemlere ise yardımcı sistemler denilmektedir. Yardımcı sistem, kullanıcının göreceli olarak basit nitelikteki kararları vermesine yardımcı olan bir sistemdir. Yardımcı sistemler nihai kullanıcının belirli bir sorunu çözmekten ziyade muhakeme sürecinde yapabileceği bir hata olasılığını azaltma amacını gütmektedir.

Uzman sistemleri, yardımcı sistemleri ve bunların arasındaki herhangi bir sistemi geliştirmek için ihtiyaç duyulan teknoloji aynı teknolojidir. Bu yüzden yukarıda bahsedilen kavram kargaşası ortaya çıkmaktadır.

Uzman sistemler insan düşünce sürecini taklit etmeye çalışır, muhakeme edebilir, çıkarımda ve yargıda bulunabilir. Günümüzde uzman sistemler değişik bilim dallarında karar vermeye yardımcı olarak kullanılmaktadır. Örneğin, tıbbi teşhiste, petrol araştırmasında, finansal planlamada, vergi hesaplamada, kimyasal analizde, cerrahide, lokomotif onarımında, hava tahmininde, bilgisayar tamiratında, uydu onarımında, bilgisayar sistemlerinin tasarımında, nükleer santrallerin işletilmesinde, devlet yasalarını yorumlamada ve daha nice alanlarda etkin bir biçimde kullanılmaktadır.

2.2.Tanım

Bir uzman sistem, uzmanlık ve deneyim kullanımına ihtiyaç duyan ve zor problemleri yüksek bir yeterlilik seviyesinde çözebilen, yalnızca belli bir alanla ilgili geniş tabanlı bir bilgiye sahip ileri bir bilgisayar programıdır. Bunu; teknik bilgiler, bilgi, sezgisel yöntemler ve bu tür problemleri çözmek için uzmanlar tarafından kullanılan problem çözme süreçlerini kullanarak başarır. Uzman sistemler, böylece; insan bilgisi, uzmanlığı ve deneyiminin bilgisayarlarda depolanmasını sağlar.

Arkada uzman sistemlerle ilgili çeşitli tanımlar verilmiştir:

Bir uzman sistem belirli bir uzmanlık alanındaki önemli problemleri çözmek için düşünen uzmanı taklit eden bilgi tabanlı bir sistemdir (Jackson, 1986).

Uzman Sistem, çözümleri için önemli ölçüde insan uzmanlığı gerektiren karmaşık problemleri çözmek için bilgi, mantıksal çıkarım prosedürleri kullanan akıllı bir bilgisayar programıdır (Feigenbaum, 1982).

2.3.Uzman Sistemlerin Gelişim Süreci

Uzman sistem alanındaki öncü proje DENDRAL'dır. Bu proje 1965'te E.Feigenbaum ve meslektaşları tarafından Birleşik Devletler Standford Üniversitesinde bir kimyagere, organik bir bileşiğin yapısını, kitle spektrogramının ve ham kimyasal formülünün verileriyle bulması için, yardımcı olmak üzere başlatılmıştır. Fizik-kimya alanına özgü bilgiler muhakeme mekanizmalarına sıkı sıkıya bağlıdır. Bilgi tabanlı sistemlerin ve uzman sistemlerin asıl temeli zamanla, verili bir problemi çözmek için bir bilgiler ve olgular bütününe kullanan çıkarım mekanizması kavramıyla birlikte doğmuştur. Daha sonra tıp alanında yeni programlar geliştirilmiştir.

1976 yılında Standford üniversitesinde Edward Feingbaum başkanlığında bir grup uzman hekim tarafından MYCIN olarak adlandırılan uzman sistem geliştirilmiştir. Bakteriyolojik ve Menenjitik hastalıkların tedavisine yönelik bir sistemdir.

Sistem girdi olarak aşağıdaki bilgileri almaktadır:

- Hastanın geçmiş bilgileri (hasta kayıt dosyasından),
- Laboratuvar sonuçları,
- Semptomların sorgulanması.

Bilgilerin derlenmesi ile sonuç olarak ;

- Teşhis koyma,
- Reçete yazımı,
- Tedavi süreçlerinin belirlenmesi, sistemden çıktı olarak alınmaktadır.

MYCIN'e veri girme ve diğer işlemler sırasında, niçin ve nasıl soruları sistem tarafından cevaplanmakta, sistem kullanıcı ile etkileşimli olarak çalışmaktadır.

2.4.Uzman Sistemlerin Genel Özellikleri

Başarılı bir uzman sistemin dört temel özelliği vardır. Bunlar:

- Önemli Performans Yeteneği
- Makul Cevap Süresi
- Değişebilirlik
- Aktif Kullanıcı Arayüzü

Önemli Performans Yeteneği

Uzman sistemlerin başarılı bir şekilde gerçekleştirilmesi ve kullanılması için en az, ilgili konudaki uzman kadar performans göstermesi gerekir. Aksi halde sistemi meydana getirmek gereksizdir. Hatta sistemin uzmandan “daha iyi” performans göstermesi istenecektir. Ancak uzman “kadar” performans gösteren sistemin geniş bir kullanıcı kitlesi arasında yaygınlaşmasında elde edilecek çeşitli faydalar bulunmaktadır.

Makul Cevap Zamanı

Bu özellik bir önceki ile yakından ilgilidir. Sistem, sorularla cevapları sağlama sırasında makul olmayan bir zaman harcarsa kullanılmaz ve son kullanıcılar uzman tavsiyesi için uzmanın kendisine danışma uygulamasına dönerler. Standart bir bilgisayar sistemi dizayn edildiğinde cevap süreleri için kriter elde edilmelidir.

Değişebilirlik

Uzman her zaman değişir. Yeni gerçekler, teknikler ve metotlar keşfedilmekte, yeni gerçekler ve kurallar gün ışığına çıkmaktadır. Bir uzman sistem, uzmanların bilgilerini değiştirebilmeleri- ile benzer ölçüde esnekliği kullanıcılara sağlamak zorundadırlar.

Aktif Kullanıcı Arayüzü

Çok iyi bir bilgi tabanına sahip; ancak zayıf bir kullanıcı arayüzü ile sunulan bir sistem kullanılmayacaktır. Bununla birlikte küçük bir bilgi tabanı ve iyi bir arayüzden oluşan sistem çok faydalı olabilir.

2.5. Uzman Sistemlerin Geleneksel Bilgisayar Programlarından Farkları

Uzman sistemler, kendilerini geleneksel bilgisayar programlarından ayıran çeşitli kabiliyetlere ve özelliklere sahiptirler. Bir uzman sistemin temel hedefi, uzmanların sezgisel uzmanlıklarına sahip olmak ve bunu yaygınlaştırmaktır; geleneksel bir programın hedefi ise bir algoritma setini tamamlamaktır.

Uzman sistemler, geleneksel yaklaşımlarla ele alınmasının çok zor veya imkansız olduğu düşünülen durumlarda kullanılabilir. Ayrıca uzman sistemler, ihtiyaç duyulan bilginin eksik veya ulaşılabilen bilginin tutarsız olduğu durumları ele alma kabiliyetine sahiptir. Uzman sistem teknolojisindeki teknikler belirsiz durumları ve belirsiz verileri tahlil ederler ve bazı uzman sistemler yeni bilgilere ulaşıldığında yargılarını değiştirebilirler. Bunlara ilaveten, çoğu kez bir uzman sistem aldığı kararlara veya neden belirli bir operasyonu gerçekleştirdiğine ya da belirli bir bilgiyi sorduğuna açıklama da getirebilir.

Uzman sistemler öncelikle, sayısal hesaplamalardan çok kavramlar hakkında sembolik muhakemeye dayanırlar. Bu sistemler çoğunlukla işleme ait yaklaşımlardan çok açıklayıcı yaklaşımlar kullanılarak programlanırlar ve programlama teknikleri, program kontrolünün alan bilgisinden geniş ölçüde ayrılmasına müsaade eder. Program kontrolünden ayrılmış açıklayıcı bilginin kullanımı, çoğu kez uzman sistemlerin, geleneksel programlara göre daha esnek ve tekrar gözden geçirilip düzeltilme ve güncellenmesinin daha kolay olmasını sağlamaktadır.

Geleneksel bir programın bu yeteneklere sahip olması çok zordur. Şekil 2.1 uzman sistemlerle geleneksel bilgisayar programlarını karşılaştırmaktadır.

Tablo 2.1 Uzman sistemler ve geleneksel bilgisayar programları arasındaki genel farklar

Uzman Sistem	Geleneksel Program
Kararlar alır	Sonuçlar hesaplar
Sezgisel yöntemlere dayanır	Algoritmalara dayanır
Daha esnek	Daha az esnek
Belirsizliği ele alabilir	Belirsizliği ele alamaz
Kısmi bilgi, tutarsızlıklar ve kısmi kanaatlerle çalışabilir	Komple bilgiye ihtiyaç duyar
Sonuçların açıklamalarını sağlayabilir	Sonuçları açıklamaz
Sembolik muhakeme	Sayısal hesaplamalar
Öncelikle açıklayıcı	Öncelikle işleme ait
Kontrol ve bilgi ayrılmış	Kontrol ve bilgi iç içe

2.6. Uzman Sistemlerin Avantajları ve Dezavantajları

2.6.1 Uzman Sistemlerin Avantajları

- **Elde Edilebilirlik:** Eğer uzmanın bilgisi bir dosyada saklanırsa bu bilgiyi organizasyon içerisinde geniş bir şekilde dağıtmak mümkündür. İhtiyaç duyan her bölüm, bu dosyanın bir kopyasını alabilir ve bu kopyaya her zaman ulaşılabilir. İnsan uzmanların aksine bilgisayar programları, sinirlenmez, yorulmaz, tatile çıkmaz, hasta olmaz ve ölmezler. Bilgiler daima kullanıma hazırdır.
- **Tutarlılık :** En iyi uzman bile yanlışlar yapabilir ya da önemli bir noktayı unutabilir. İyi bir uzman sistemle yanlışlar daha az meydana gelecektir. Bir bilgisayar programı tutarlıdır. Doğru olarak programlanması durumunda daima doğru sonuçlar verecektir.

- **Geniş Kapsamlılık:** Birden fazla uzmanın ortak fikrini almak çok zordur ve bir grup uzmanı bir konuda tartışıp, ortak bir yargıya varmak üzere bir araya getirmek neredeyse imkânsızdır. Bir uzman yalnızca kendi bilgi ve deneyimlerini kullanabilir. Bir bilgisayar sistemi ile, bir uzman sistem birden fazla uzmanın bilgisini içerebilir. Böylece uzman sistem tarafından verilen karar, en az diğer katılımcıların vereceği karar kadar iyi olacaktır. Ayrıca uzman sistemler birbirleriyle görüş alışverişinde bulunabilirler ve çeşitli seçenekler sunabilirler.

2.6.2 Uzman Sistemlerin Dezavantajları

- **Alan Seçimi:** Uzman sistem uygulamalarının bazıları iyi çalışırken diğerleri çalışmaz.

Bu sebeple belirli bir alanda belirli bir teknoloji kullanma ya da kullanmama kararı bilgisayar tabanlı sistemlerde önemlidir. Bazı problemler uzman sistemler için fazla karmaşık olabilir. Eğer uzmanlar hemfikir değilse veya o alandaki uzman müsait değilse bu durumda o alan uygun değildir. Benzer şekilde çözmesi çok uzun zaman alan, bir çok etkileşimin olduğu ya da uzaysal (spatial) ilişkilere ve prosedürlere çok fazla bağımlılığın bulunduğu problemler de uzman sistemler için uygun değildir.

- **Test Etme:** Uzman sistemleri test etmek, özellikle, basitçe tavsiye vermekten çok karar almaya yardımcı olmak için kullanılan sistemlerde kritik bir ihtiyaçtır. Bir uzman sistemin test edilmesinde pek çok sorunla karşılaşılabilir. Programı geliştirenler, sistemin nasıl davranması gerektiği konusunda her zaman emin değildirlere ve bu yüzden sistemi tam anlamıyla test edemezler. Biz uzman sistem programının izleyeceği yolu tanımlamak kolay değildir. Bu durum çok fazla risk içeren uygulamalarda ciddi bir problemdir (örneğin tıbbi uygulamalar). Sistemin büyüklüğü arttıkça kontrol ve bakım işlemlerinin zorluğu da artar. Kullanıcı, sistem tarafından önerilen tavsiyenin en uygun tavsiye olduğundan emin olmak ve olasılık dahilindeki tüm yan etkileri hesaba katmak zorundadır. Sistemin doğruluğunun ve tamlığının test edilmesi çok zordur. Bununla birlikte,

nihayet bu bir sorumluluk sorunudur: Kim sorumlu? Uzman sistemi kullanan, yazan, bilgisi uzman sistemde kullanılan ya da bu sistemin kullanım hakkına sahip olan kişi mi?

- **Belirsizlik:** Gerçek dünyada sadece doğru ve yanlış gerçeklerle ilgilenmeyiz. Genellikle sadece belli bir derece emin oluruz (kağıtları evde bıraktığımızda %80 emin oluruz gibi). İdeal olarak bir uzman sistem, belirsizliğin üstesinden gelebilmelidir. Fakat, uzman sistemin sunduğu mantıksal muhakemenin kullanımı için geçerli istatistiksel kurallar bulmak büyük bir teorik sorundur.
- **Sınırlamalar:** Uzman sınırlarını bilir; bilgisayar tabanlı bir sistem eğer özellikle programlanmamışsa bilemez. Bu bakımdan, uzman sistemler pek o kadar iyi sonuç vermezler. Her zaman bir cevap üretmeye yöneliktirler. Bu durum problem olabilir ve uzman sistemlerin uzmanlara ikame etmek yerine yardımcı bir araç olarak kullanılmalarının daha uygun olacağının altı çizilmelidir.
- **Kabul Edilebilirlik:** Uzman sistem teknolojisine giriş, firma içerisinde, firmanın organizasyonel yapısı için önemli sonuçlar doğurur. Herkes bir bilgisayara güvenmek ya da kullanmak istemeyebilir. Bazı insanlar bilgisayar kullanımına direnç gösterirler ve uzmanlarla çalışmayı tercih ederler. Uzmanlar bile bazen uzman sistemler ile ilgili şüphecidir. Hatta sistemler iyi çalıştığı ve uzmanlarla hemfikir olduğu zamanlarda bile kendileriyle aynı tip muhakeme yöntemini kullanmadıklarını düşündükleri için uzman sistemlere güvenmezler. Bu faktörler uzman sistem kullanımı düşünüldüğünde göz önünde bulundurulmalıdır. Aksi halde sistem, firma içinde aktif bir muhalefet ile karşılaşabilir.
- **Bilgi Edinme:** Uzman sistem geliştirmek için şimdiye kadar karşılaşılan en ciddi sorun, bilgi edinmenin yol açtığı “dar boğaz (bottleneck)”dır. Uzmanın bilgisini saklama ve ortaya çıkarma prosesi çoğunlukla uzun ve yavaş bir prosestir.
- **Güncelleme:** Bilgilerin sık sık değiştiği alanlar uzman sistemlerin geliştirilmesine pek uygun değildir. Uzman sistemin uzmanlığını kaybetmemesi

için bilgi tabanını sürekli güncellemesi gerekir. Bilgi tabanının güncellenmesi için gerekli koşullar tesis edilmelidir.

- **Davranış :** Uzman sistemlerin amacı, insan uzmanları taklit etmek olmasına rağmen bunu yapabilen çok az sistem vardır. Sistemle kullanıcı arasındaki diyaloglar genellikle program tarafından yönlendirilir ve genellikle tanımlamaları anlamak zor olabilir. Konsültasyonlar kullanıcının değil bilgisayar programının güdümünde olma eğilimindedir ve kullanıcı cevabı elde edebilmek için sık sık gereksiz açıklamalarla uğraşmak zorunda kalmaktadır.

2.7.Uzman Sistemlerin Kullanımına Uygun Alanlar

Uzman sistemler ancak gerekli olduğu zamanlarda kullanılmalıdır. Bazı işletmelerin bulunduğu koşullar, uzman sistemin kullanılmasını gerek maliyet, gerekse sağlanan faydanın önemsiz olması yüzünden haklı kılmayabilir. Uzman sistemlerden aşağıdaki koşullar altında yararlanılması tavsiye edilmektedir:

- İşin tekdüzeliğine bağlı olarak uzman sistemin sık sık kullanılmasına gereksinim duyuluyorsa ve kullanıcı sayısı uzman sistemin kullanılmasını maliyet boyutunda ekonomik kılacak kadar fazla sayıda ise,
- Karar verme durumu karmaşıksa (basit durumlar için basit bir bilgisayar programından da yararlanılabilir.)
- Karar verme mantığı bir kural hiyerarşisine dönüştürülebiliyorsa
- Gözden geçirmenin çok uzun zaman alacağı, çok sayıda mümkün kombinasyonun olduğu problemlerde,
- Çok miktarda dikkate değer veriyi anlamlandırmanın ya da bilgilerin geri çağırılması (retrieval) işlemlerinin yapılmasının gerektiği durumlarda.
- Uygulama öneri, sınıflama, teşhis, yorum, açıklama, çözüm yolu seçme, durumu değerlendirme ve tahmin etme üzerinde yoğunlaşıyorsa.

2.8.Uzman Sistemlerin Rollerı

Bir uzman sistem, ařağıda verilen olası uzman rollerinin bir veya birden fazlasını gerekleřtirebilir.

- **Otonom Uzman:** Bir uzman sistem bir durumu analiz eder, bir karar alır ve daha sonra doęrudan doęruya ona gre davranır veya bir insan faaliyetini kumanda eder. Tm bu ařamalarda bir insan katılımı sz konusu deęildir. Nadir veya olaęandışı durumlara uygun bir řekilde tepki gsterir. rneęin, otonom bir uzmanın rolnde, bir uzman sistem; alıcı aletleri okuyarak, sıcaklık ve basın gibi kontrol edilebilir deęiřkenlerde gerekli dzeltmeleri yapar ve doęrudan doęruya bir kimyasal sreci bařlatabilir. Veya bir teknisyeni, sistem tarafından hatalı olduęu belirlenmiř bir makine parasını deęiřtirmeye yneltebilir.
- **İnsan Katılımlı Otonom Uzman:** Uzman sistem, otonom bir uzman gibi davranır fakat aldıęı kararları, bu kararları dikkate almayacak veya deęiřtirebilecek olan bir uzmana sunar. Bu rolde, bir uzman sistem, uygulamadan nce bir tasarımcı tarafından gzden geirilecek olan, kompleks bir tasarımı formle edebilir.
- **Uzman Danıřman:** Bir uzman sistem, bir grevi yerine getiren kiři iin, uzman danıřmanların yaptıęı gibi, zellikle zor veya olaęandışı durumlar iin tavsiyelerde bulunan, uzman seviyede bir danıřmanlık saęlar.
- **Meslektař:** Uzman sistem, kendi sahasındaki insanlara nerilerde bulunur. Her ne kadar insandan daha yksek ustalık seviyesine sahip olamasa da, insanın yapmaya zaman bulamayacaęı analizleri yapabilir veya insan, faaliyetin en muhtemel seeneęini elde etmeye alıřırken; uzman sistem, ok sayıda karmařık alternatifi arařtırabilir. rneęin, aksaklıkları saptayıp zmleyen bir uzman sistem, hataların muhtemel olmayan sebeplerini inceleyebilir. Oysa, aksaklıkları saptayıp zmleyen insanlar, bu muhtemel olmayan sebeplerden birinin gerekleřmesi durumunda, analizler iin gerekli olan zamanın zararı tazmin edilemedięinden bunlarla uęrařmazlar.
- **Zeki Yardımcı:** Uzman sistem, uygulayıcının zeki bir yardımcısı gibidir. Belli bir durum meydana geldięinde, bir dizi makul neride bulunabilir veya dikkate alınması

gereken bir dizi madde belirleyebilir. Uygulayıcının ilgileneceği şekilde değişkenlerin belli kombinasyonlarını araştırarak verileri gözleyebilir. Örneğin, zeki bir yardımcı rolünde, bir uzman sistem, bir fabrikada alıcı aletleri gözleyebilir ve ölçülen değerlerin bazı kombinasyonlarının problem yaratacağını saptadığında bir gösterge paneli aracılığıyla işaret verebilir.

- **Düşük – Seviye Zeki Yardımcı:** Uzman sistem, uygulayıcının vaktini işin daha zor kısımlarına ayırabilmesini sağlamak amacıyla nispeten düşük seviyede ve zaman alan görevleri yerine getirir. Örnek olarak, bir uzman sistem, büyük miktarlardaki ham veriyi, uzmanın incelemesini kolaylaştıracak şekilde daha küçük miktarlarda kısmi işlenmiş veriye dönüştürerek karmaşık bir uzman analizinin ilk adımını gerçekleştirebilir.

2.9.Uzman Sistemlerin Uygulama Alanları

Teknolojideki ilgi yayıldıkça, uzman sistemlerin uygulanmış olduğu ve uygulandığı problem tiplerinin sayısı sürekli artmaktadır. Uzman sistemler şu amaçlarla geliştirilebilir:

- teşhis
- çizelgeleme
- planlama
- izleme
- süreç kontrol
- tasarım
- tahmin
- sinyal yorumlama
- konfigürasyon
- eğitim

Uzman sistemlerin kullanıldığı endüstrilerin ve disiplinlerin sayısı da ayrıca artmaktadır.

Tablo 2.2:Uzman Sistemlerin Geliştirilmiş Olduğu Çeşitli Alanları Göstermektedir.

Havacılık ve Uzay	Mühendislik	Tıp
Tarım	Mühendislik tasarımı	Askeri bilimler
Kimya mühendisliği	Çevre bilimleri	Nükleer mühendisliği
Kimya	Finans	Basın-yayın
Bilgisayar destekli tasarım	Jeoloji	Kaynak Yönetimi
İnşaat	Devlet	Telekomünikasyon
Elektrik mühendisliği ve ve elektronik	Sigorta	Eğitim
Elektrik kurumları	İmalat	Taşıma

2.10.Uzman Sistemlerin Faydaları

Uzman sistem teknolojisi; uzmanlığın korunmasının veya uzmanlığa ulaşım ve uzmanlığın dağılım alanının genişletilmesinin değerli olduğu durumlar için düşünülebilir. Uzman sistemler aşağıdaki durumlarda uygun (ve çoğu durumda başarılı olabilecek tek teknoloji) olabilir:

- Uzmanlığın nadir olduğu durumlarda,
- Uzmanlığı elde etmenin pahalı olduğu durumlarda,
- Uzmanlara ulaşamayan zamanlarda,
- Uzmanlara ulaşamayan mekanlarda uzmanlık sağlamak.
- Bir uzman analizinin alacağı zamanı kısaltmak.
- Daha az deneyimli ve daha az usta personelin performans derecesini yükseltmek.
- Deneyimli ve usta personelin performans derecesini arttırmak, böylece en tepedeki uzmanın seviyesine daha yakın bir performans göstermelerini sağlamak.

- Uzmanlık analizlerini uzmanların zaman bulamadığı durumlara uygulamak (bir karar vermeden önce büyük miktarda veriyi analiz etmek veya herbir olasılığı –en olmayacak olanı bile- göz önünde bulundurmak gibi).
- Uzmanlığın; yeniden atama, emeklilik veya başka personel kaybı yoluyla kaybedilebileceği durumlarda bilgiyi ele geçirmek ve saklamak.
- Uzmanlar uzmanlıklarının en tepesindeyken, ileride aynı yoğunlukta çalışmadıklarında mevcut hünerlerinin bir kısmını kaybedebilecekleri ihtimalini düşünerek bilgiyi ele geçirmek ve saklamak.
- Uzmanlar hastalık veya tatil nedeniyle bulunmadıklarında,
- Uzmanlar düşük performans seviyelerinde olduklarında uzmanlığa ulaşım sağlamak.
- Sürekli personel değişiminin sözkonusu olduğu durumlarda, yeni personelin sürekli olarak eğitilmesi ihtiyacını,
- En iyi uzmanların bile zaman zaman yapabilecekleri, imla hataları ve diğer küçük hataları en aza indirmek veya ortadan kaldırmak.
- Otomatikleşmiş uzmanlığı, diğer bağlı görevlerin otomatikleştirilmesi için bir temel olarak kullanarak faydalı kılmak, böylece geniş bir görevin komple otomasyonuna izin vermek.
- Uzmanlığın, her seferinde; aynı tarzda, tarafsız ve devamlı olarak uygulanmasını sağlamak, böylece, örneğin şirket politikasını kararlı hale getirmek.
- En iyi uzmanların zamanını en zor problemler veya diğer önemli görevler için boşaltmak.
- Bir ürün veya servis olarak (veya bir ürün veya servisin parçası olarak) pazarlanabilecek bir şekilde uzmanlığı otomatikleştirmek.

Böylece uzman sistemler; performans ve kaliteyi yükseltmek, verimliliği arttırmak, kararlılık sağlamak, önemli bilgiyi korumak, insan kaynaklarının daha iyi kullanılmasını temin etmek ve yeni veya daha iyi ürünlerin pazarlanmasına imkan vermek için uzmanlık bilgisi sağlarlar.

2.11.Uzman Sistemlerin Uygulanmasındaki Kısıtlar

Her ne kadar uzman sistemler, bazı durumlarda en iyi uzmanlardan daha iyi performans gösterebilir ve her ne kadar uzman sistem teknolojisi, geleneksel programlama teknikleriyle çözümü aşırı derecede zor olan (eğer tamamen imkansız değilse) problemlerin çözümünde kullanılabilirse de, bu sistemler her derde deva olarak düşünülmemelidir. Hiçbir teknoloji bütün problemleri çözemez ve bu yüzden uzman sistemlerin kısıtları bilinmelidir.

Herhangi bir alanda, bir uzman sistemin önde gelen bir uzmanın komple uzmanlığına ve deneyimine sahip olması ve insan bilgisine dayanıyor olması nedeniyle, bu sistemlerin, uzmanların çözemediği problemleri çözmesi mümkün değildir (vaktin yetmediği ve bilgisayarın sahip olduğu kaynaklara ulaşamama nedeniyle çözülemeyen problemler hariç).

Bir uzman sistem geliştirmek sadece dar ve sınırları iyi belirlenmiş bir alanda mümkündür. Mevcut uzman sistemler çoğunlukla sınırlarında zayıf performans gösterirler ve bu sistemlerin çoğu bir durumun kendi faaliyet alanlarına girip girmediğini belirlemede güçlük çekerler. Mevcut ticari uzman sistemler önemli öğrenme yeteneklerine sahip değildir ve sadece birkaçı, sezgisel yöntemler işe yaramadığında faaliyet alanının temel prensiplerinin kullanımına imkan tanır.

2.12.Uzman Sistemlerin Geliştirilmesi

Uzman sistem geliştirmede sabit kurallar yoktur, birden fazla yaklaşımın kullanılabileceği birçok durum vardır ve her durumda işe yarayan birkaç standart teknik vardır. Bu yüzden burada, bir uzman sistemin geliştirilmesinde takip edilecek adımlar tanımlanırken, her adımda, uygulamada başarılı olmuş, ispatlanmış teknikler üzerinde durulacaktır.

Uzman sistem geliştirmenin öncelikli hedefi, seçilen alandaki bir uzmandan, alan içindeki bir görev deneyiminden kazanmış olduğu bilgiyi elde etmek ve daha sonra bu bilgiyi bir uzman sistem programında kullanmaktır.

En temel haliyle, uzman sistem geliştirme dört temel adımdan oluşur:

- 1) Alan Seçimi
- 2) Uzmanların Seçimi
- 3) Bilginin Elde Edilmesi
- 4) Program Geliştirilmesi

2.12.1.Alan Seçimi

Uzman sistem için bir uygulama alanının seçimidir. Bir uzman sistem için uygun alanın seçiminde, takip edilebilecek birçok teknik ve teknik olmayan düşünce vardır.

2.12.2.Uzmanların Seçimi

Seçilen alanla ilgili bir veya birden fazla uzmanın seçilmesidir. Uzman sistemler tasarlanırken, planlayıcı mutlaka, uzmanların derin tecrübelerinden ve tekrarlanan gözlemlerinden elde edilen uzmanlık bilgileri üzerine odaklanmalıdır. Bir uzman sistem geliştirme projesi için bir alan uzmanı belirlerken hesaba katılması gereken birçok faktör vardır. Burada anahtar nokta, belirlenen amaçlar ve problemler için doğru tip bilgiye sahip bir uzman bulmaktır.

Uzman; bir görevi, eğitimini ve deneyimini kullanarak, ele geçirmek ve dağıtmanın yararlı olacağı seviyede bir ustalıkla gerçekleştiren kişidir. Uzman, Nobel Ödülü kazanmış bir fizikçi veya işinin ehli bir ofis elemanı olabilir. Her ne kadar genellikle en üst seviyedeki kişinin bilgisini ele geçirmeye çalışsak da, zaman zaman orta seviyedeki bir karar vericinin muhakemelerini bile elde etmek ve otomatikleştirmek yararlı olabilir. Bu durumda, bu kişi bizim amaçlarımız için bir uzman olarak düşünülebilir.

Çok sayıda tanımlanmış alt problemleri ile birlikte geniş bir bilgi sistemi inşa ederken, en alttan tepe yönetime kadar, organizasyonun her aşamasında, sık sık bilgi toplamak uygundur. Burada fikir, problemin nasıl büyüdüğünün, organizasyona nasıl etki ettiğini ve çözümüne kimlerin katılması gerektiğinin analiz edilmesidir. Örnek olarak, özel bir tip film şirketinin kullanacağı, farklı test prosedürlerinin belgelendirileceği bir uzman sistem yaratan mühendislik firmasındaki proje ele alınırsa; burada bilgi araştırmasına en

alt sorumluluk düzeyinden, çeşitli film stoklarının üzerine ışık yayan karanlık oda teknikerlerinden, başlanır. Bir sonraki görüşme, farklı test yerlerinden filmleri taşıyarak, yükleyerek ve açığa çıkararak yarar sağlayan kamera operatörleri ile olacaktır. Bir başka seviyede, test prosedürlerinin çeşitli örneklerinin teknik talepleri ile ilgili olarak tasarım mühendislerinden bilgi alınır. Satın alma yöneticisi ise maliyet ve kazançlarla ilgili verileri sağlar.

Ayrıca, bilgi tabanına ek sağlamak için diğer firmalarla ilgili bilgilere sahip dış tecrübeli uzmanlar tercih edilebilir. Yukarıdaki örnekte, iç personel sadece ticari yayınları okurken dış video danışmanı yeni film ve prosesler ile ilgili bilgi sağlayabilir. İç uzmanlar konu hakkında sınırlı veya yüzeysel bilgiye sahipken, dış uzmanlar başlıca bilgi kaynağı haline gelir.

Sınırlı bir problem sahasında küçük bir bilgi tabanı inşa ederken, tekil bir kaynak tercih edilir. Bir uzmanlık alanındaki çeşitli uzmanların bilgilerini entegre etmek oldukça zordur ve girdiler gerçekten çelişkili olabilir.

İstekli bir uzman seçilirse, bilgi toplama prosesi eksiksiz olarak gerçekleştirilir.

2.12.3.Bilginin Elde Edilmesi

Uzman sistem geliştirme sırasında alan bilgisinin –gerçekler, kurallar, sezgisel yöntemler, prosedürler- uzmanlardan elde edilme süreci, **bilginin elde edilmesi** olarak adlandırılır. Bir uzman sistem geliştirilirken, bilgi mühendisleri, dokümanlardan ve doğrudan doğruya uzmanlardan yola çıkarak, uzman sistemin çözmeye çalıştığı problemi uzmanların çözme biçimlerini belirlemeye çalışır. Bilginin elde edilmesinin sonucu, uzman sistemin bilgisinin ayrıntılarıyla tanımlanmasıdır.

Bilginin elde edilmesi, çoğunlukla, bilgi mühendisleri ve alan uzmanları arasında yapılan ve bilgi mühendislerinin uzmanların bilgisini temin etmeye çalıştığı toplantılar yoluyla başlar. Bilgi mühendisleri için bir alan uzmanıyla görüşmenin genel bir yaklaşımı, uzmanın kullandığı prosedür ve sezgisel yöntemleri belirleyebilmek için, seçilen alana ait görevi nasıl yerine getirdiği ile ilgili sorular sormaktır. Bilgi

mühendisleri uzmana belli durumlar (örneğin, gerçek bir alan probleminin verileri) sunar ve o durumda karar vermek için kullanılan sezgisel yöntemleri temin etmeye çalışır. Bu tipik bir döngüdür. Bilgi mühendisleri ve uzman, temin edilmiş olan alan bilgisini uzmanın performansı ile karşılaştırmak için test problemlerini kullanırlar. Sistemin bilgisindeki zayıflıklar bulunur, değiştirilir ve yeniden uzman değerlendirmesine sunulur. Bu döngü, uzman sistemin bilgisi istenilen noktaya gelinceye kadar devam eder.

Bilginin elde edilmesi, çoğunlukla zordur. Çoğu uzman uzmanlıklarını veya karar verme şekillerini kolayca tanımlayamazlar. Bir çoğu, arzu edileceği şekilde düşüncelerini rahatça ifade edebilen ve konuşkan kişiler değillerdir. Bundan başka, uzmanlar sıklıkla, kullandıkları süreçleri, sezgisel yöntemleri ve uzmanlıklarını nasıl kullanılır hale getirdiklerinin tam olarak farkına varmadan görevlerini yaparlar. Bir karar aldıklarında, gerçekte göz önünde bulundurmuş oldukları safhaların farkında olmayabilirler. Bunlar gibi faktörlerden dolayı, bilginin elde edilmesi sürecinde, mümkün olabildiğince tam ve doğru sezgisel uzman yöntemlerinin belirlenebilmesi için, çoğunlukla bilgi mühendislerinin uzun dönemler boyunca alan uzmanlarıyla etkileşim halinde olmaları gerekir.

Örnek 2.1 bilginin elde edilmesi sürecinin basit bir gösterimidir. Ayrıca, bilgi elde etme sırasında meydana gelebilecek çeşitli zorluklardan bazılarını ortaya koymaktadır.

Örnek 2.1: Bilginin Elde Edilmesi İle İlgili Basit Bir Örnek

Yaygın bir uzmanlık alanıyla ilgili olan aşağıdaki kuramsal örnek (bir bilgi mühendisi ve otomobil kullanma konusunda uzman bir kişi arasında geçebilecek bir diyalog), tipik bir bilginin elde edilmesi süreci örneği (ve bazı zorlukların işaret edilmesi) olarak düşünülebilir.

Bilgi mühendisi: Araba kullanırken kullandığınız teknikleri bana anlatır mısınız?

Uzman: . . . kırmızı ışıktaki durmuşsam ve yeşil ışık yanarsa, gaza basarak arabayı ileri doğru hareket ettiririm. Ve daha sonra . . .

Bilgi mühendisi: Burada durun. Bu önemli bir sezgisel yöntem benziyor. Hadi bunu kaydedelim. Tekrar söyleyebilir misiniz?

Uzman: Bu çok genel bir durum. Yeşil ışık yanarsa, gaza basarak arabayı ileri doğru hareket ettiririm. Bu kadar basit.

Bilgi mühendisi: Tamam. Hadi bunu sezgisel bir kural şeklinde resmileştirelim.

YEŞİL-IŞIK-YANDIĞINDA-İLERİ-DOĞRU-HAREKET-ET Kuralı

Eğer araba kırmızı ışıktaki durmuş vaziyette ise,

ve yeşil ışık yanarsa,

O zaman gaza basarak arabayı ileri doğru hareket ettir.

[Uzmanların sezgisel yöntemlerinin çoğunlukla *if-then* kuralı şeklinde yazıldığına dikkat edilmelidir.]

Uzman: Bu kadar. Çok basit kurallar kullandığımı söylemiştim. Çok yakında bitirmiş olacağız.

Bilginin elde edilmesinde daha sonraki bir aşama:

Bilgi mühendisi: Şimdi şu senaryoyu düşünün. Güney Sokağı'na doğru gidiyorsunuz ve kırmızı ışıktaki durmuşsunuz. Yeşil ışık yandığında ne yaparsınız?

Uzman: Gaza basar ve arabayı ileri doğru hareket ettiririm.

Bilgi mühendisi: Şu ana kadar topladığımız kuralları kullanarak, sistemimiz aynı kararı verecek. Bu noktaya gelindiğinde, YEŞİL-IŞIK-YANDIĞINDA-İLERİ-DOĞRU-HAREKET-ET Kuralı'na müracaat edilecek. Şu ana kadar çok güzel.

Sezgisel yöntem doğru gibi gözükebilir. Bununla birlikte, bilgi mühendisi daha sonra sorabilir:

Bilgi mühendisi: Şimdi şu durumu düşünün: Kırmızı bir ışıktaki durmuşsunuz ve tam karşınızda 'Girilmez' işareti var. Yeşil ışık yanarsa ne yaparsınız?

Uzman: İleri gidemem öyleyse. Sağa ya da sola dönmek zorundayım.

Bilgi mühendisi: Fakat bu, daha önce bulduğumuz, YEŞİL-IŞIK-YANDIĞINDA-İLERİ-DOĞRU-HAREKET-ET kuralıyla tezat teşkil ediyor.

Uzman: Bu doğru. Özür dilerim. Sadece ‘Girilmez’ işaretinin olmadığı durumda ileri doğru hareket edebileceğimi söylemeyi unutmuşum.

Bilgi mühendisi: Tamam. YEŞİL-IŞIK-YANDIĞINDA-İLERİ-DOĞRU-HAREKET-ET kuralını aşağıdaki şekilde genişletirsek daha iyi olacak:

YEŞİL-IŞIK-YANDIĞINDA-İLERİ-DOĞRU-HAREKET-ET Kuralı

Eğer araba kırmızı ışıktaki durmuş vaziyette ise,
ve yeşil ışık yanarsa,
ve tam karşıda ‘Girilmez’ işareti yoksa,
O zaman gaza basarak arabayı ileri doğru hareket ettir.

Daha sonra, devam ederken:

Bilgi mühendisi: Eğer ışığa rağmen biri karşıdan karşıya geçiyorsa ne yaparsın?

Uzman: Daha önce bundan bahsetmedim mi? Tabii ki ileri doğru hareket etmeye başlamadan önce arabanın önünden birinin geçip geçmediğini kontrol ederim.

Bilgi mühendisi: Anlıyorum. Hadi YEŞİL-IŞIK-YANDIĞINDA-İLERİ-DOĞRU-HAREKET-ET Kuralı’nı tekrar genişletelim.

YEŞİL-IŞIK-YANDIĞINDA-İLERİ-DOĞRU-HAREKET-ET Kuralı

Eğer araba kırmızı ışıktaki durmuş vaziyette ise,
ve yeşil ışık yanarsa,
ve tam karşıda ‘Girilmez’ işareti yoksa,
ve arabanın önünden kimse geçmiyorsa,
O zaman gaza basarak arabayı ileri doğru hareket ettir.

Bilgi mühendisi: (sonraki çeşitli zamanlarda): Eğer arabanın önünde büyük ve derin bir çukur varsa ne yaparsınız? [veya] Yan sokaktan yaklaşıarak gelen bir siren sesi duyarsanız ne yaparsınız?

Ve böyle devam edebilir. Önemli bir nokta şu; uzmanlar gerçekte ‘Girilmez’ işaretleri, sirenler ve daha birçok şeyi kontrol ederler, ancak bu kontroller onlar için öyle doğaldır ve otomatikleşmiştir ki, sorulduğunda, ne yaptıklarını ifade edemezler. Görüleceği gibi basit sezgisel yöntemler genelde düşünüldüğünden çok daha karışık olabilir. Bilgi mühendisinin yeteneği herbir sezgisel yöntemi –ve tabii komple sezgisel yöntemi- uzmandan temin etmekte saklıdır.

2.12.4.Program Geliştirilmesi

Belirlenen teknikleri, bilgiyi ve sezgisel yöntemleri bir araya toplayan bir bilgisayar programının tasarımıdır.

Bilginin elde edilmesi sürecinde sağlanan uzman bilgisini düzenlemek için bir uzman sistem programının geliştirilmesinde beş temel eleman vardır:

- i. Yazılım aracının seçilmesi
- ii. Donanımın seçilmesi
- iii. Bilginin test edilmesi
- iv. Bilginin dönüştürülmesi
- v. Programın test edilmesi ve yeniden değerlendirilmesi

Uzman sistemlerin çoğu bir **yazılım aracı** veya **kabuğu (shell)** kullanılarak geliştirilirler. Yazılım aracı veya kabuğu, uzman sistem geliştirilmesine yardım imkanı sağlayan bir yazılım paketidir. Uzman sistem geliştiricileri böyle bir paketin kullanılıp kullanılmayacağına (doğrudan doğruya bir bilgisayar dili kullanılarak programlamak yerine) ve eğer kullanılacaksa hangisinin kullanılacağına karar vermelidirler. Ayrıca uzman sistemin hangi bilgisayar donanımı üzerinde geliştirileceğini de belirlemelidirler. Yazılım aracının ve donanımın seçilmesi işlemlerinin ikisi birlikte geliştirme çevresi olarak adlandırılabilir.

Bilginin temsil edilmesi, bilginin elde edilmesi sürecinde sağlanan alan bilgisini alma ve bu bilgiyi uzman sistem programında temsil etmek için kullanılacak olan yaklaşımı tarif etme işlemidir. Uzmanlık bilgisi, bilgisayarda, bir programlama dili kullanılarak temsil edilebilir, fakat genellikle; üretim kuralları, çatılar veya nesneye yönelik programlama gibi bir veya daha fazla **Yapay Zeka Paradigmaları**'nın kullanılır hale getirilmesiyle temsil edilir. Bu paradigmalarla (çoğunlukla yazılım aracı tarafından sağlanırlar), uzman sistem geliştiricileri uzman bilgisi için bir temsil planı belirlerler.

Bilginin dönüştürülmesi, bilginin elde edilme sürecinde sağlanan bilgiyi alma ve bilginin temsil edilmesini kapsayan yapıları ve paradigmaları kullanarak, bilgiyi bir operasyonel uzman sistem programına dönüştürme işlemidir.

Geliştirme sürecinde ve ayrıca son transfer veya yayılmadan önce, uzman sistem programı test edilmeli ve yeniden değerlendirilmelidir. Uzman sistem programlarının bazı safhaları test etme ve yeniden değerlendirme işlemleri, geleneksel bilgisayar programlarının test edilmesi ve yeniden değerlendirilmesinden biraz farklı yapar.

2.13.Uzman Sistemlerin Yapısı

2.13.1.Uzman Sistemlerin Temel Elemanları

Bir uzman sistemin yapısı aşağıdaki şekilde görüldüğü gibidir

Şekil 2.1’den de görüldüğü gibi, bir uzman sistem bilgi elde etme ünitesi, bilgi tabanı, çıkarım sistemi ve kullanıcı ile iletişim ünitesinden oluşur.

2.13.1.1.Bilgi Elde Etme Ünitesi

Bu ünite bilgi tabanını oluşturan birimdir. Bilgi tabanı bir bilgi mühendisi ile bir veya birden fazla konu ile ilgili uzmanın koordineli çalışması sonucunda yaratılır. Bilgi mühendisi, uzman sistemin çalışma esası ve mülakat teknikleri konusunda eğitim görmüş bir insandır. Başlangıç görüşmelerinde alan uzmanı belirli bir sorunun nasıl çözülmesi konusunda bildiği herşeyi bilgi mühendisine anlatır. Başlangıçta yapılan mülakatların sonucunda elde edilen bilgiler genellikle elastik ve systemsizdir. İkinci aşamada, bilgi mühendisi alan uzmanını iş yerinde gözlemler ve bu esnada da daha fazla bilgi almak ve mevcut kuralların yeterli ve doğru olduğunu teyit etmek için alan uzmanına sürekli soru sorar. Bu aşama genelde bir yıl kadar uzun sürer.

Bilgi mühendisi alan uzmanının konu ile ilgili tesbit ve kurallar setini uzman sisteme bilgi tabanı olarak aktarmaya çalışır. Bilgi tabanındaki bilgiler EĞER-0 ZAMAN (IF-THAN) kuralları şeklinde yer alır.

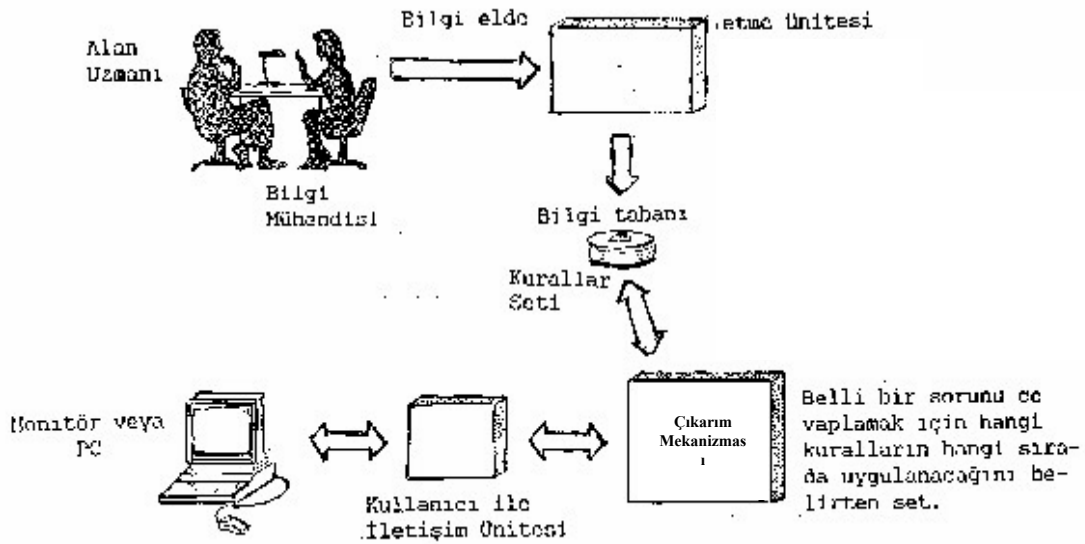
- “Eğer” bölümü durumu açıklar,
- “0 zaman” bölümü sonuç veya amacı izah eder.

Bilgi tabanında yer alan bir kurala şöyle bir örnek verebiliriz: “Eğer bir müşterinin aylık geliri bankaya yapılan aylık ödemenin 3 katından daha az ise, o zaman müşterinin kredi geçmişini incele”. Sonuçta, elde edilen bilgi tabanı öyle bir kural seti olacaktır ki, bunu doğrudan illetmenin veri tabanından veya uygulamada kullanılan kurallardan müşahade etmek mümkün değildir.

2.13.1.2.Bilgi Tabanı

Bilgi tabanı veri tabanından farklı bir kavramdır. Klasik veri tabanının konusu öğeler arasındaki durağan ilişkiler ile ilgili verilerdir. Örneğin, bir iş gören kaydı ile iş görenin işi ve ücreti alanları arasında sabit bir ilişki vardır. Öte yandan bilgi tabanındaki bilgiler şu tür bilgilerden oluşur:

- Çözülecek problem ya da problemlerin belirlenmesi,
- Problem ya da problemlere çözüm yolları,
- Problemden çözüme doğru nasıl ilerlemeli (tespit ve kurallar seti aracılığıyla)



Şekil 2.3

Bilgi tabanının içerdiği bilgiler zamanla yaşanan tecrübelerle bağlı olarak artış gösterir. Bazı kurallar atılır yerine yenileri eklenir.

2.13.1.3.Çıkarım Mekanizması

Çıkarım mekanizması, bir uzman sistemin çekirdeğidir. Bilgi tabanında yer alan tesbit ve kuralların belli bir soruna tatbik edilmesini sağlayan araçtır. Bu sistemde uzman sisteme muhakeme yeteneği kazandırılır. Bu muhakeme gücü kullanıcıya bir mantık silsilesinin sunulması ile sağlanır ve böylece çözüme ulaşılır.

Bir çıkarım sisteminin muhakeme becerisi ileri zincir ya da geri zincir çıkarım süresinin birlikte ya da tek başına kullanılması esasına dayanır. İleri zincirde, uzman sistem nihai kullanıcıdan bilgiler alır ve çözüme ulaşınca kadar bilgi tabanından duruma uygun kuralları sırası ile takip eder. Bu süreç esnasında sürekli olarak kullanıcı ile uzman sistem arasında iletişim vardır ve bu iletişim önceden yerleştirilmiş kurallar setinin oluşturduğu mantık silsilesine göre yürütülür.

Geri zincir çıkarım sürecinde ise ileri zincirin tam zıttı bir yaklaşım kullanılır. Sistem nihai kullanıcıya istediği hedef ya da sonucu sorar ve daha sonra “Eğer-o zaman” mantık silsilesine geri dönerek uzatılmak istenen hedef ya da sonucun doğru olup olmadığını araştırır. Eğer bilgi tabanındaki “Eğer-o zaman” kurallar seti hedef ya da sonuç ile uyuyorsa, kullanıcı tarafından saptanan hedef ya da sonuç, sorunun çözümü demektir.

2.13.1.4.Kullanıcı ile İletişim Ünitesi

Yordamlama süreci biçimsel değildir, yani, bir sorunu çözmek için geliştirilmiş yazılı bir algoritma yoktur. Bir uzman sistem bir strateji ise yarattığı süreci kullanmaya devam eder. Sistemde yeni bir stratejiye dönüş seçeneği her zaman mevcuttur. Bu yordamlama süreci bu yüzden her zaman kullanıcı ile iletişime gereksinim duyar. Kullanıcı ile iletişim sayesinde nihai kullanıcı uzman sisteme sorunu ya da hedefi belirtir. Kullanıcı ile iletişim ünitesi aracılığıyla kullanıcı ile uzman sistem arasında iletişim kurulur ve böylece çözüme ulaşılmaya çalışılır.

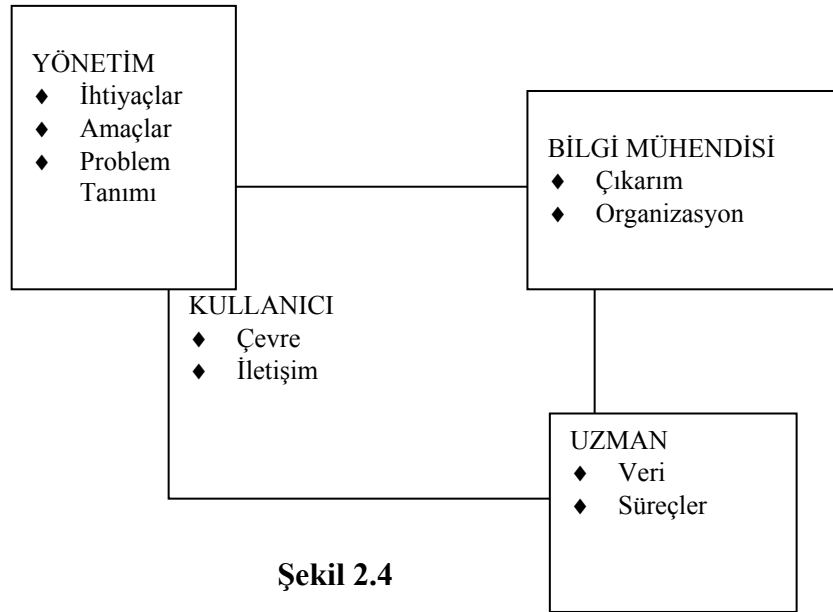
2.13.2.Uzman Sistemlerin İnsan ve Makine Yönü

2.13.2.1.İnsan Yönü

Uzman sistemin insan yönü dört gruptan oluşur. Birinci grup, yönetimdir. Yönetim, uzman sistemin ihtiyacını belirlemek zorundadır ve bunu yaparken organizasyonun faydalarının sistemden türediğini farzetmelidir. İkinci grup, bilgi mühendisidir. Bilgi mühendisi, bilgi tabanındaki bilgileri uygun veri haline getirmeli ve bilgiyi sezgisel olarak organize etmelidir. Üçüncü grup kullanıcıdır. Kullanıcı, sistemin nasıl kullanılacağı, ne tip problemlerin çözüleceği ve ne tarz programların insan operatörüyle iletişim kuracağını hazırlar. Dördüncü grup ise, uzmandır. Sistem uzmana ihtiyaç duyar ya da uzman grupları gerçeklere dayalı bilgi formunda ve alan içindeki problemleri çözmede kullanılan analitik metodlarla herhangi bir konu için bilgi tedarik ederler. Şekil 2.4 uzman sistemlerin insan yönünü özetlemektedir.

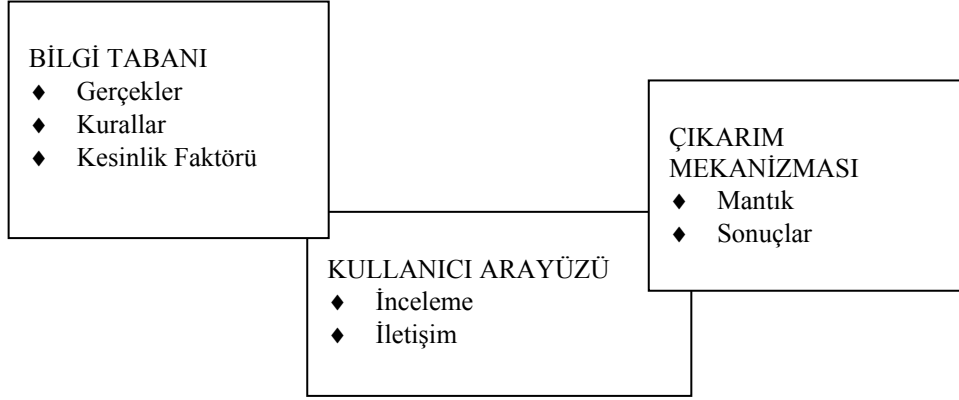
2.13.2.2.Makine Yönü

Makine yönünde, gerçeklerin sembolik olarak temsil edildiği bilgi tabanının yazılım araçları vardır. Bütün uzman sistemlerde; bilgi tabanı, kullanıcı arayüzü ve çıkarım



Şekil 2.4

mekanizması bulunmaktadır. Bilgi tabanı gerçeklerin ifadesinden oluşur. Gerçekler



Şekil 2.5

arasındaki bağlantı sezgisel kurallarla belirlenir. Her kural ‘if’ önermesi ve ‘then’ sonucuna sahiptir.

Önceki ve sonraki rastlantı sonucu olan olaylar tanımlanır. Gerçek iddiaların kesinlik derecesi her zaman mutlak değildir. Nicelik kesinliğinin faktörleri uzman sistemlerinin sonuçlarının doğruluğunu artırır. Bu ilgili ifadelerin güvenilirliği istatistik, olasılık veya tamamıyla kişisel fikirlere dayanır. Çıkarım mekanizması, sonuç çizimleri ve sonuçlar testinin kuralları arasında mantıksal ilişki kurar. Kullanıcı arayüzü, sistemi inceler ve yöneticiyle iletişim kurar. Şekil 2.5 Uzman Sistemlerin makine yönünü özetlemektedir.

Kaynaklar

- 1.Kosko, Bart, "Fuzzy Engineering", Prentice Hall, 1997
- 2.Satoru, I., Kosko, B. "Fuzzy Logic", Scientific American, Temmuz 1993
- 3.Driankov, D., Hellendoorn, H., "An Introduction to Fuzzy Control", Springer-Verlag Berlin Heidelberg, 1993
- 4.Klir, G., Yuan, B., "Fuzzy Sets and Fuzzy Logic", Prentice Hall, 1995
5. Haykin S., Neural Networks, Macmillan, New York 1994.
6. Carrie, A., Simulation of Manufacturing Systems, John Wiley&Sons, New York 1988.