

YAPAY SİNİR AĞLARI GELİŞİMİ VE YAPILARININ İNCELENMESİ

Doç.Dr. Haldun AKPINAR
İ.Ü. İşletme Fakültesi

1. GİRİŞ

Yatırım yönetimi dalında faaliyet gösteren ve 130 milyon Amerikan Doları tutarında portföye sahip bulunan LBS Capital Management isimli bir A.B.D. İşletmesi, 5 yıllık süre içerisinde bir uzman sistemden^{1a} faydalanarak, Standard & Poor 500 hisse senedi indeksinde % 72 oranında başarılı tahminde bulunabildiklerini, bir yapay sinir ağı modelinin sisteme eklenmesi ile, tahminlerindeki başarı oranlarının % 92'ye yükseldiğini ifade etmektedir. [SCH, 69]

St. John's Üniversitesi'nden Patrick Lyons "*Integrating Neural Nets with Expert Systems for Mergers and Acquisitions Decisions*" (İşletmelerin Birleşme ve Satın Alınma Kararlarında, Uzman Sistemlerle Yapay Sinir Ağlarının Entegrasyonu) isimli bildirisinde, basit olarak hesaplanabilen finansal oranları ve yapay sinir ağlarını kullanarak, bir firmanın finansal yapısının çok iyi, iyi, orta veya zayıf olarak sınıflandırılabilirdiğini göstermektedir. [SCH, 70]

Pace Üniversitesi'nden Michael Gargano, Larry von Kleeck ve Hofstra Üniversitesi'nden Robert Marose tarafından hazırlanan "*An Application of Artificial Neural Networks and Genetic Algorithms to Personnel Selection in the*

^{1a} Uzman Sistemler tasarım, planlama, teşhis, yorumlama, denetim, danışmanlık gibi uzman kişilerin yapabildiği faaliyetleri gerçekleştiren bilgisayar programlarıdır. [AKP-6, 14]

Financial Industry" (Finans Endüstrisinde Personel Seçimi için, Yapay Sinir Ağları ve Genetik Algoritmaların^{1b} Bir Uygulaması) isimli makalede, yapay sinir ağları ve genetik algoritmaların kullanımı ile gerçekleştirilen etkin bir personel seçim sistemi tanıtılmaktadır. Kurulan sistem yapılan testlerin sonucunda işe alınan adayların gelecekteki başarı durumlarını % 82 ihtimalle doğru olarak tahmin etmektedir. [LIN, 37]

Hoptroff tarafından yapay sinir ağları kullanılarak gerçekleştirilen bir başka çalışma, ekonominin dönüş noktalarını tahmin etmektedir.

MasterCard, MasterCom adını verdiği sistemde görüntü işlem teknolojisi ve yapay sinir ağlarını kullanarak, satış noktasında el yazısıyla hazırlanan kredi kartı belgelerini otomatik olarak işleyebilmektedir. [STE, 205]

Kredi kararlarının verilmesi, *underwriting* işlemleri, işletmelerin finansal durumlarının derecelendirilmesi (*rating*), ülkelerin finansal ve politik durumlarının derecelendirilmesi gibi çeşitli uygulamalarda, yapay sinir ağı modelleri etkin bir başarı sağlamaktadır.

Bütün bu örnekler yapay zeka disiplini altında gelişen, yapay sinir ağı modellerinin işletme uygulamalarındaki başarısını göstermektedir. Başlangıçta bilgisayarların konuşulanı algılaması, yazılan karakterleri tanınması, robotların çeşitli birimlerinin kontrolü gibi uygulamalarda kullanılan yapay sinir ağı modelleri, artık işletmecilik uygulamalarında da büyük bir önem taşımaya başlamıştır.

Bu çalışma içerisinde ilk olarak yapay zeka çalışmalarına kısaca değinilmekte ve yapay sinir ağı modellerinin, yapay zeka disiplini içerisinde bulunduğu konum tanıtılmaktadır. Bu tanıtımı yapay sinir ağı modellerinin temel çıkış noktası olan insan sinir fizyolojisinin, ana hatları ile incelenmesi takip etmekte, bu temel bilgilerin ardından, yapay sinir ağı kavramına, yapay sinir ağı modellerinin gelişimine, kullanıldıkları sahalara ve genel özelliklerine yer verilmektedir.

^{1b} Genetik Algoritmalar, belirli bir sahada yapılan gözlemleri temsil edecek uygun bir fonksiyonun bulunmasını amaç edinen bir optimizasyon yöntemidir. John. H. Holland tarafından geliştirilen bu yöntem, canlı varlıkların genetik yapısından esinlendiği için bu isimle anılmaktadır. [HOL, 66-72]

2.1. YAPAY ZEKA

Başta Zeus olmak üzere tüm tanrıların öfkesini çeken **Pronietheus**, sadece ateşi çalmakla kalmamış, Olympus'a ait olan zekayı, medeniyetini geliştirebilmesi için insanoğluna hediye etmiştir. [MEY,339] Efsaneye göre zekasını böyle kazanan insanoğlu, günümüzde zekasını kullanarak, yapay zeka (*Artificial Intelligence*) adı altında toplanan çeşitli araştırma çalışmaları ile, makinaları zeki yapma uğraşı içindedir.

Latince'de zeka kelimesinin karşılığında kullanılan *Intellectus*, algılama, bilme, anlayış ve tanıma anlamlarına sahiptir. Yapılan çeşitli tanımlamalara göre zeka; algılama, bellek, öğrenme, düşünme, soyutlama ve yeni durumlara uyma gibi bir çok zihinsel yeteneğin (*Mental Faculties*) bileşimidir. Diğer bir tanıma göre,

"zeka, kişinin yeni durum, engel ve sorunlar karşısında deneyimlerinden ve öğrendiklerinden yararlanarak o an için gerekeni yapması, uyum sağlayabilmesi, yeni çözümler bulabilmesi yeteneğidir." [KÖK, 51-56]

Tüm araştırmacıların üzerinde anlaştığı bir tanım olmamakla birlikte, **Yapay Zeka**, insana özgü zeka davranışların otomasyonunu araştıran, bilgilerin saklanması ve işlenmesinde veri yapıları, algoritmalar, programlama dilleri ve teknikleri gibi bilgi işlem yöntemlerini kullanan, bilgisayar biliminin bir alt dalıdır. [LUG,1; CHA,1; WIL,1-6; LEV,3]

İnsanlığın en büyük hayallerinden olan düşünen ve öğrenen bir sistem kurma konusunda yüzyıllardır çeşitli çalışmalar yapılmış, kurgu bilim dalları başta olmak üzere konu çeşitli sanat eserlerinde işlenmiştir. Yapay zeka konusunun iyi anlaşılabilmesi, felsefeden bilgisayar bilimine, psikolojiden beynin anatomik yapısına kadar geniş bir yelpaze içinde çeşitli bilim dallarında yapılan çalışmaların izlenmesi ile mümkündür.

Düşüncenin temel mantık kuralları ile gösteriminin araştırıldığı **Aristoteles'in** (M.Ö. 384-322) çalışmalarından, **Descartes'a** (1596-1650) kadar çeşitli düşünürlerin incelemeleri, iki temel sonuca varılmasını sağlamıştır.

Descartes ve onu izleyen düşünürlerin oluşturduğu fikir yapısına göre varlık, ruh ve madde (düşünce ve uzam) olmak üzere iki cevherden oluşmuştur. Bu yaklaşım maddi cevherle ilgili elemanların tamamen maddi,

hatta mekanik bir yoldan açıklanabileceği düşüncesini ortaya koymaktadır. [MEY,3/591] Zihni işlemlerin kendine özgü bir varlığı ve kendi kanunlarının geçerli olduğunu belirten bu düşünce, bilgi kuramı (*Epistemology*), psikoloji gibi dallarla birlikte yapay zekanın da başlangıç kuramını oluşturmuştur. [LUG,6]

Ruh ve maddenin birbirinden ayrılması fikri sonucunda, düşünürler insanın oluşumu için gerekli bu iki varlığın nasıl birlikte hareket ettiğinin açıklanması ile ilgilenmiştir. Bu konuda ileri sürülen varsayımlar içerisinde, İskoç düşünürü David Hume'un "*Cognition is computation*" ifadesi ile açıkladığı, zihni işlemlerin de fiziksel işlemler gibi matematik yoluyla ifade edilebileceği fikri, geniş kabul görmüştür. [LUG,6-7] Düşünmenin bir hesaplama işlemi olduğu kabul edildikten sonra, bu işlemin nasıl ifade edilebileceği ve çözülebileceği konusundaki çalışmalar önem kazanmıştır.

1956 yılında John McCarthy tarafından düzenlenen ve Minsky, Newell, Simon, Shannon başta olmak üzere on bilim adamının, iki ay süre ile *Dartmouth College*'de (New Hampshire, Hannover, A.B.D.) yaptıkları çalışmaların sonucunda, John McCarthy'nin önerisi ile "*Artificial Intelligence*" ismi ilk kez kullanılmış ve yapay zeka bir araştırma disiplini olarak benimsenmiştir. [CHA,11]

Oyun programları, otomatik çıkarım ve teorem ispatlama, konuşulan dilin anlaşılması ve modellenmesi, insan davranışlarının modelinin kurulması, yapay zeka için gerekli programlama dil ve ortamlarının geliştirilmesi, makina öğrenimi, robotik, görüntü algılama ve uzman sistemler yapay zeka araştırmaları içerisinde başlıca alt araştırma dalları olmuştur.

Yapay zeka konusunda yapılan ilk araştırmalarda kuramsal olarak, Turing Makinası temel alınmıştır. Yapay zeka dalının kurucularından sayılan Newell ve Simon, yapay zekanın temel varsayımını,

"Genel zeka faaliyetleri için, Fiziksel Simge Sistemleri (*Physical Symbol Systems*) gerekli ve yeterlidir"

tezi ile ifade etmektedir. [NEW,1] Bu görüşe göre simgeleri işleyebilen, bilgisayar gibi her hangi bir sistemin zeka unsurlarına sahip olabileceği ifade edilmektedir. Ancak bu alanda çalışanlar, halen hangi fiziksel simge sistemlerinin zeka için gerekli ve yeterli olduğu sorusunu, tam olarak

cevaplayamamaktadır. Ayrıca Searle, Winograd, Flores gibi araştırmacılar, zekanın simgelerle ifade edilemeyecek biyolojik bir varlık olduğu görüşü ile bu sistemi reddetmektedir. [LUG,28]

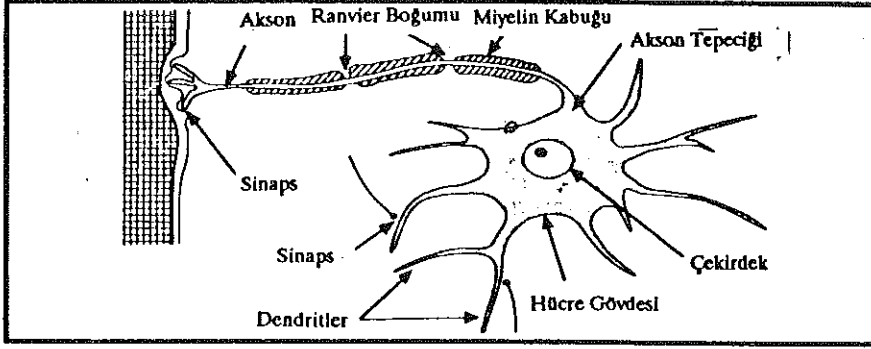
1950'li yılların ortalarından başlayarak yapay zeka araştırmaları, simgesel (*symbolic*) ve simgesel olmayan (*nonsymbolic*) iki temel yaklaşıma ayrılmıştır. Bu çalışma içerisinde simgesel olmayan yaklaşım ele alınmakta ve bu yaklaşımın temsilcisi olan yapay sinir ağı modelleri işletmecilik uygulamaları açısından incelenmektedir.

2.2. TEMEL SİNİR FİZYOLOJİSİ

Günümüzde yapay sinir ağı modelleri, biyolojik ağların karmaşıklığını yansıtmaktan henüz çok uzaktır. Ancak bir çok yapay sinir ağı modelinin tasarlanmasında, kısıtlı da olsa biyolojik sinir ağlarının temel alındığı unutulmamalıdır. Bu nedenle biyolojik sinir hücresi ağlarının, çok basit düzeyde de olsa tanıtılmasında yarar bulunmaktadır.

Sinir sisteminin bir parçası olan ve ortalama 1.5 kilogram ağırlığındaki insan beyinde, tahminen 10^{11} sayısında sinir hücresi bulunmaktadır. Öğrenme, hatırlama, düşünme algılama gibi tüm bilişsel davranışları da içeren, her türlü İnsan davranışının temelinde nöron hücreleri bulunmaktadır. İşlevleri henüz tam olarak açıklığa kavuşmayan, ancak nöronların çalışmasını destekleyici ve onları besleyici bir işlevi olduğu düşünülen ikinci bir tip sinir hücresi gliall hücreleridir. [KAN.18]

Tüm sinir hücrelerine doğuştan sahip olan bir insanın, sinir hücreleri yaşamı içerisinde yenilenmemektedir. Beynin gelişmesi ve ağırlık kazanması, sinir hücrelerinin büyümesi ve aralarında yeni bağlantıların kurulmasından kaynaklanmaktadır. Sinir sistemi içerisinde farklı işlevlere sahip olan beş ayrı sinir hücresi olmakla birlikte, her sinir hücresi şekil 1.1. de gösterildiği gibi hücre gövdesi, dendritler (*Dendrite*) ve aksondan (*Axon*) meydana gelir. Sinir sisteminde gördüğü işleve göre bir milimetreden daha küçük veya bir metreden daha büyük olan bilgi taşıyıcısı aksonlar, sinir akımının daha süratli olarak akson üzerinden aktarılmasını sağlayan *Myelin Sheath* isimli bir kabukla kaplıdır. Bu kabuk yer yer Ranvier Boğumu (*Ranvier Node*) adı verilen noktalarda kesintiye uğramaktadır. Sinir hücrelerinin birbirleriyle iletişim kurması *Synapse* adı verilen birleşme yerlerinde gerçekleşmektedir. [CÜC.54-56]



Şekil 1.1 : Sinir Hücresini Oluşturan Elemanlar [KAN,19]

Sinir hücreleri arasındaki iletişim, elektro-kimyasal bir süreç olan sinir akımları ile sağlanmaktadır. Belirli iyonlara karşı geçirgen olan sinir hücresi zarı, bu iyonların bulunduğu hücre içi ve dışı sıvılar arasında, potansiyel bir farkı sodyum-potasyum pompasına benzer bir yapıda korumayı amaçlar. Bu mekanizma içerisinde sinir hücresi zarı sodyum iyonlarının hücre dışına çıkmasını, potasyum iyonlarının ise hücre içine girmesini sağlar. Sıvı içerisinde yer alan klorid gibi diğer tüm iyon tipleri hücre zarından geçebilir. Ancak hücre zarının geçirgenliğine göre çok büyük olan organik iyonlar hücre dışına yayılmadığı için, negatif etkileri ile hücre içine yayılan klorid iyonlarını etkisiz hale getirirler. Bunun sonucunda hücre dışı sıvıda klorid iyonlarının daha yüksek bir konsantrasyonu oluşur. Hücre zarı potasyum iyonlarını, sodyum iyonlarına göre daha kolay geçirmektedir. Potasyumun kimyasal derecesi potasyum iyonlarının hücre dışına yayılmasına neden olurken, negatif organik iyonlar potasyumu hücre içerisinde tutmak için kuvvetli bir faaliyet gösterir. Bu karşıt güçlerin savaşımlı sonucunda, daha fazla sodyum ve klorid iyonları hücre dışında kalırken, daha fazla potasyum ve organik iyonlar hücre içinde kalmakta, bu da bir denge durumuna ulaşılmasını sağlamaktadır. Oluşan bu denge ile hücre içi sıvının daha negatif olması sonucunda, hücre zarında 70-100 milivolt düzeyinde bir potansiyel enerji kalmaktadır.

Bu şekilde polarize duruma geçen sinir hücresi, kendisine yakın diğer hücreler tarafından uyarıldığında, hücre zarının geçirgenlik derecesi değişerek, sodyum iyonlarının süratle hücre içerisine dolmasını ve zarın iç kısmının, dışına göre daha pozitif yüklü duruma gelmesini sağlamaktadır. Bu duruma ise

hücre zarının depolarize olması denilmektedir. Akson tepeciğinden (*Axon Hillock*) başlayarak, zarın bir noktasında meydana gelen depolarizasyon, bu noktaya bitişik diğer noktaların da geçirgenlik derecesini değiştirmekte, depolarizasyon durumunun aksonun sonuna kadar zincirleme bir reaksiyon şeklinde devam etmesini, sinirsel akımın hücre gövdesinden, aksonun en ucunda yer alan ve diğer sinir hücrelerinin dendritleri ile ilişki kurabilen sinaptik uçlara kadar aktarılmasını sağlamaktadır.

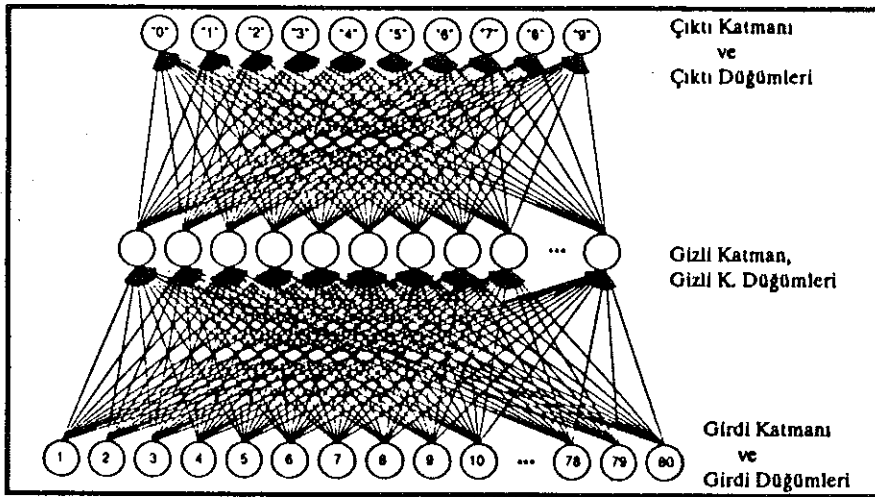
Sinaptik uçlara gelen sinirsel akım, kalsiyum iyonlarının etkisiyle zarın geçirgenliğini değiştirmekte ve sinaptik keseciklerdeki (*Synaptic Vesicles*) biokimyasal maddelerden oluşan sinirsel aktarıcıları (*Neurotransmitters*) etkilemektedir. Etkilenen sinirsel aktarıcılar, 1/2.000.000 santimetre genişliğindeki sinaptik aralıktan (*Synaptic Gap*) geçerek, diğer sinir hücresinin dendritlerine ulaşmaktadır. Sinaptik aktarıcılar bu tür sinaptik birleşmeler sırasında, diğer sinir hücresinde uyarıcı (*Excitatory*) veya engelleyici (*Inhibitory*) olmak üzere iki farklı durumun gerçekleşmesine neden olur. Alıcı durumunda bulunan sinir hücresindeki kimyasal faaliyet, bazı iyon örneklerinin daha kolay geçmesini sağlayacak şekilde zarın geçirgenliğini değiştirir. Uyarıcı durumda, hücre zarının geçirgenliğinin değişmesi sonucunda hücre içerisine daha kolay akan pozitif iyonlar, sinir hücresinin depolarize olmasını sağlayıp, bu hücrede de sinirsel bir akımın başlamasına yol açar. Engelleyici durumda ise, negatif iyonlar sinir hücresini daha da polarize ederek (*Hyperpolarization*), sinirsel akımın durmasını sağlamaktadır.

Ancak sinir sistemindeki sinir hücrelerinin her birisi, sadece yukarıda belirtilen şekilde tek bir sinir hücresi tarafından uyarılmamaktadır. Bir sinir hücresine, birden fazla sinir hücresi tarafından aynı anda yapılan etkiler, hem uyarıcı hem de engelleyici olabilecektir. Alıcı sinir hücresi, uyarıcı ve engelleyici nitelikteki bu etkileri akson tepeciğinde toplar. Elde edilen bu toplam, belirli bir eşik değerinden (*Threshold*) büyükse, sinirsel akımın bu sinir hücresinde de devamı sağlanır.

Yüzbinlerce sinir hücresinin birbirleriyle ilişki içerisinde bulunması sonucunda, sonsuz denilebilecek sayıda sinaptik birleşme söz konusu olacaktır. Bu sayıdaki sinaptik birleşimi gerçekleştirecek bir bilgisayarın, dünyadan daha büyük bir hacmi kaplayacağı hesaplanmaktadır. Ortalama 1.5 kilogram olan insan beyninin karmaşıklığı, bu örnekle olanca yalınlığı ile ifade edilebilmektedir. [CÜC,57]

2.3. BİYOLOJİK SİNİR HÜCRELERİNDEN YAPAY SİNİR AĞLARINA

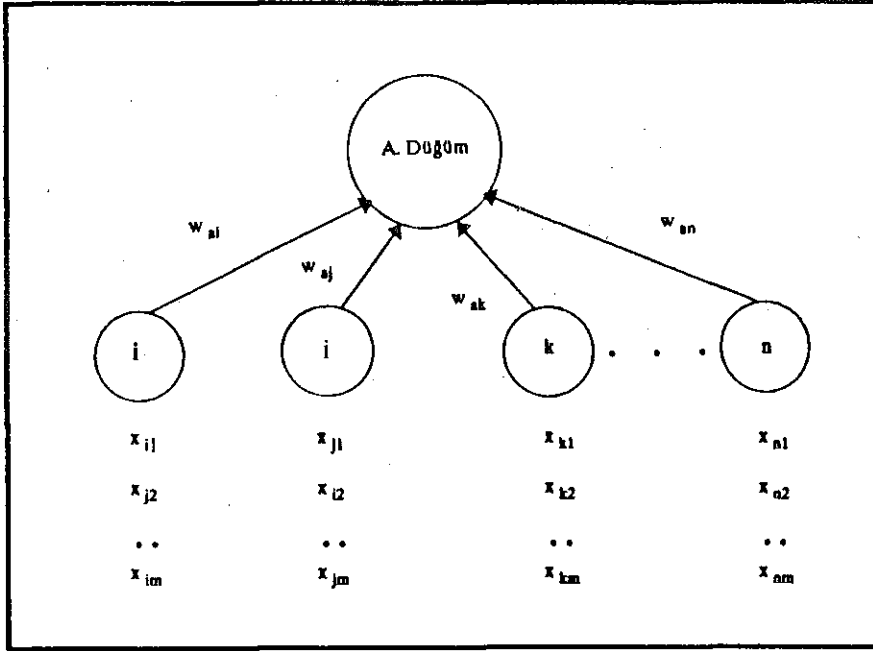
Yapay Sinir Ağı² modelleri (*Artificial Neural Net Models*), *Connectionist Modeller*, *Paralel Dağıtılmış İşleme Modelleri (Parallel Distributed Processing Models)* veya *Neuronorphic Sistem* gibi değişik şekillerde isimlendirilmektedir. [LIP,5] Ancak ismi ne olursa olsun tüm bu modeller, biyolojik sinir sisteminin bilinen yapısını göz önüne alarak, yüksek bir performansın elde edilmesini sağlayacak şekilde basit hesaplama elemanlarının yoğun bağlantılarından meydana gelmiştir. YSA modelleri, paralel ve yüksek hesaplama hızlarının gerekli olduğu ve mevcut en iyi bilgisayar sistemlerinin dahi gerçekleştirmekten oldukça uzak olduğu, özellikle konuşma ve görüntü algılama başta olmak üzere çeşitli sahalarda büyük bir potansiyele sahiptir. Von Neumann bilgisayarlarında olduğu gibi program komutlarının sırasal olarak icra edilmesi yerine, YSA modelleri bir çok hesaplama elemanının değişken ağırlıklarla birbirlerine bağlanmasını ve yoğun paralel işlemlerin yapılmasını öngören çeşitli varsayımlara dayandırılmaktadır.



Şekil 1.2 : Genel Bir YSA Modeli

² Çalışma içerisinde Yapay Sinir Ağı ifadesi YSA kısaltması ile temsil edilmektedir.

Genel olarak bir YSA modeli şekil 1.2 de görüldüğü gibi, n adet Katman (Layer), her kalmada biyolojik sinir hücrelerine benzer işlevi yerine getiren ve değişik sayılarda olabilen hesaplama elemanları, katmanlar boyunca bu hesaplama elemanları arasındaki yoğun bağlantılardan meydana gelmektedir. Çeşitli YSA modellerinde kullanılan hesaplama elemanları, Yapay Sinir Hücresi (Artificial Neuron), Düğüm (Node), Birim (Unit) veya İşlem Elemanı (Processing Element) olarak isimlendirilmektedir.



Şekil 1.3 : Bir YSA Düğümünde Gerçekleşen Hesaplama İşlemi

Biyolojik bir sinir hücresine benzer şekilde faaliyet gösteren bir düğümün birden fazla girdisi, fakat ağda yer alan diğer düğümlere gönderebileceği tek bir çıktısı bulunmaktadır. Şekil 1.3 de görüldüğü gibi ağ içerisinde A düğümünden önce işlemde bulunan, i,j,k,...,n düğümleri, x_{i1} i. düğümünden, x_{j1} j. düğümünden gönderilen 1. örüntüyü³ (Pattern) göstermek üzere, kendi çıktılarını A düğümüne girdi olacak şekilde göndermektedir.

³ Almanca'da *Muster* veya *Flachmuster*, Fransızca'da *Image*, *Figure*, *Mosaïque* kelimelerinin [WIT, P63] kullandığı örüntü kelimesi, belirli, ancak bilinmeyen bir sınıfta yer alan nesne veya olayları tanımlamaktadır. [SCH, 365] Bir görüntünün daha sonra bilgisayar tarafından algılanabilmesi için, bu görüntü sınıfına ait olan görüntü örnekleri (Örneğin A karakterinin değişik yazılışları), ses örnekleri **örüntü** olarak isimlendirilmektedir.

Önceki düğümlerden gönderilen bu girdilerin her biri, A düğümünün bu girdiler için belirlediği ve w_i, w_j ile gösterilen ağırlık değerleri ile çarpılmakta, bu çarpımlar sonucunda elde edilen toplam, A düğümünün Net Girdi (*Net Input*) değerini oluşturmaktadır.

Düğümler arasında aktarılan bütün bu değerler standart YSA modellerinde analog olmakla birlikte, anlatılacak modeller içerisinde gerçek sayılarla ifade edilmektedir. Düğümler arasında oluşturulan engelleyici bağlantılar negatif ağırlıklarla, uyarıcı bağlantılar ise pozitif ağırlıklarla çarpılmaktadır. Engelleyici ve uyarıcı olan bu iki tip bağlantının dışında, çeşitli YSA modellerinde Kazandırıcı (*Gain*), Söndürücü (*Quenching*) ve Uyandırıcı (*Nonspecific Arousal*) terimleri ile ifade edilen özel bağlantı tipleri de bulunmaktadır.

Bu tip özel bağlantıların bulunmaması halinde, bir düğüme gelen girdi değerlerinin yukarıda belirtilen şekilde ağırlık değerleri ile çarpılması ve elde edilen sonuçların toplanması, bu düğüm için net girdi değerinin hesaplanmasını sağlayacaktır. Bu işlem, n uyarı gönderen düğüm sayısını, x_k k. düğümden gelen girdiyi, w_{ik} i. düğümün k. düğüme uyguladığı ağırlığı ve net_i i. düğümün net girdisini göstermek üzere,

$$\text{net}_i = \sum_{k=1}^n x_k * w_{ik}$$

eşitliği ile genellenebilir. Bir YSA içerisinde bu tip hesaplamalar yoğun sayıda yapıldığı için, bu hesaplamaların yapılma sürati doğrudan ağırlık bilgisayarı uygulamasındaki süratini belirleyecektir.

Net girdi değerinin belirlenmesini,

$$a_i(t) = F_i(a_i(t-1), \text{net}_i(t))$$

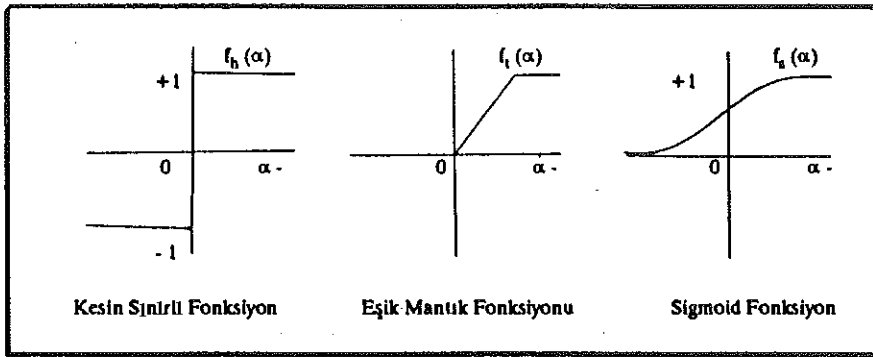
eşitliği ile ifade edilen, i. düğüm için Faaliyet Değerinin (*Activation Value*) hesaplanması izleyecektir. Eşitlikte $a_i(t)$, t. zamanda i. düğümün faaliyet değerini göstermektedir. Eşitlikte görüldüğü gibi, t. zamandaki faaliyet değeri, t-1 zamanındaki faaliyet değeri ve t. zamandaki net girdi değerinin bir fonksiyonu olarak hesaplanmaktadır. Ancak genel yapı bu olmakla birlikte,

bir çok YSA modelinde faaliyet ve net girdi değeri eşdeğer olarak kullanılmaktadır.

Faaliyet değerinin hesaplanmasını çıktı değerinin belirlenmesi izler. Bir çok YSA modelinde faaliyet değeri, net girdi değerine eşdeğer olarak kullanıldığı için, çıktı değerinin hesaplanmasında kullanılan eşitliği,

$$x_i = f_i (\text{net}_i)$$

şeklinde yazabilmek mümkündür. Kullanılan YSA modeline göre çıktı değerinin elde edilmesi için, net_i değerine uygulanan doğrusal olmayan fonksiyon şekil 1.4. de görüldüğü gibi, Kesin Sınırlı (*Hard Limiters*), Eşik Mantık Elemanı (*Threshold Logic Element*), Sigmoid düzende olabilmektedir. [LIP,6]



Şekil 1.4. : Çeşitli YSA Modellerinde Kullanılan Eşik Fonksiyonları

Kullanılan modele bağlı olarak bu şekilde çıktı değeri hesaplanan bir düğüm, ağıın içerisinde bağlantılı olduğu düğümlere çıktısını gönderecektir. Bir YS Ağıının öğrenim süreci, girdi katmanındaki düğümlere uygulanan girdi değerlerine göre, kabul edilebilir düzeydeki doğru çıktıyı hesaplayacak ağırlık matrisinin oluşturulmasıdır. Bağlantı ağırlıklarının nasıl düzenleneceği YSA araştırmalarının temel konusu olup, bu düzenleme Uyum Gösterme (*Adaptation*) veya Öğrenim olarak isimlendirilmektedir. Uyum gösterebilme yeteneği ve sürekli öğrenim, eğitim verilerinin kısıtlı olduğu ve yeni konuşmacılar, yeni kelimeler, yeni şiveler, yeni tümceler ve yeni çevrenin sürekli eklendiği bir sahada en önemli koşuldur. Uyum gösterme aynı zamanda işlem elemanlarının özelliklerinde küçük değişikliklerin giderilebilmesi için belirli bir düzeye kadar sisteme güç kazandırmaktadır.

Bir düğüm birden fazla düğüm tarafından etkilendiği için, etkileyici düğümlerdeki verilerin yetersiz veya bozuk olması, sistemin tüm performansını etkilemeyecektir. Bu durum YS Ağlarının klasik bilgi işlem uygulamalarına göre, yetersiz veya bozuk verilerle çalışma sırasında ortaya çıkabilecek hatalara karşı ne kadar esnek olduğunun bir göstergesidir.

3.YAPAY SİNİR AĞI ARAŞTIRMALARININ GELİŞİMİ

3.1. McCulloch-Pitts Modeli

Biyolojik sinir hücrelerinde sinirsel akımın nasıl aktarıldığının ve sinir hücrelerinin birbirlerini nasıl etkilediklerinin görülmesinden sonra, akla gelecek ilk soru, bu kadar basit bir mekanizmanın çeşitli şekillerde birleşerek, beyne üstün yeteneklerini nasıl kazandırdığı olacaktır.

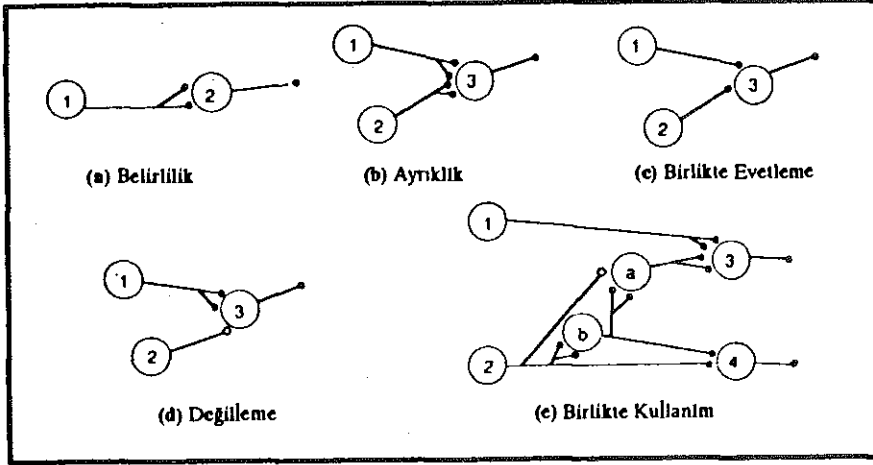
1943 yılında McCulloch ve Pitts tarafından gerçekleştirilen çalışmalar, bu soruya cevap bulmayı amaçlayan ilk önemli girişim sayılmaktadır. Bu çalışmada ikil (*Binary*) çıktı veren yapay sinir hücrelerinden oluşturulan bir ağın, hesaplama yeteneğine sahip olduğu gösterilmiştir.

McCulloch-Pitts Kuramı beş temel varsayım üzerine kurulmuştur.

1. Bir sinir hücresinin faaliyetinde, hep veya hiç ilkesi (*Ali or None Principle*) geçerlidir.
2. Bir sonraki sinir hücresinin uyarılması için, birden fazla sabit sayıdaki sinir hücresinin belirli bir zaman dilimi içerisinde uyarılması gerekmektedir.
3. Sinir sistemi içerisindeki tek belirgin gecikme sinaptik gecikmedir.
4. Her hangi bir engelleyici sinapsın faaliyeti, o anda sinir hücresinin uyarılmasını kesinlikle engeller.
5. Ağın iç bağlantılarının yapısı zamana bağlı olarak değişmez.

Birinci varsayım sinir hücresi faaliyetlerini hep veya hiç işlemi olarak belirttiği için, sinir hücreleri 0,1 veya -1,+1 değerleri ile gösterilebilecek ikil hesaplama düzenine sahiptir. Önergeler Mantığı (*Propositional Calculus*)

simgeleri kullanılarak $N_i(t)$ ile, i. sinir hücresinin t. zamanda uyarılacağı, $\neg N_i(t)$ ile ise uyarılmayacağı gösterildiğinde, şekil 1.5. de yer alan beş basit ağıın önermeler mantığı cümleleri ile ifade edilmesi mümkün olacaktır. Şekilde yer alan büyük daireler sinir hücresi gövdelerini, içi dolu küçük daireler uyarıcı bağlantıları, içi boş küçük daireler ise engelleyici uyarıları temsil etmektedir.



Şekil 1.5 : McCulloch-Pitts Modelinde Ağ Yapılarının Önermeler Mantığı Cümleleri ile İfade Edilmesi.

Şekil 1.5.e. de yer alan ağ, ilk dört ağın tanımlanmasında kullanılan basit önermeler mantığı cümlelerinin kombinasyonu ile ifade edilebilmektedir. Geri besleme bağlantılarına sahip olmayan her hangi bir ağın, **Belirlilik**, **Ayrıklık** (Veya İşlecisi), **Birlikte Evetleme** (Ve İşlecisi), **Değilleme** olarak isimlendirilen dört basit önermeler mantığı cümlesinin kombinasyonu ile tanımlanabilmesi, McCulloch-Pitts Kuramı'nın en güçlü yanlarından birisini oluşturmaktadır. Daha sonraki bulgulara göre McCulloch-Pitts Kuramı beyin faaliyetlerinin doğru bir modeli olmamakla birlikte, çalışmanın önemi her zaman vurgulanmış, kuram modern bilgisayar biliminin gelişmesinde etkileri olan çeşitli kişilerin düşüncelerine yansımıştır. Anderson ve Rosenfeld'in işaret ettiği diğer bir nokta, sinir hücreleri basit cihazlar olmakla birlikte, uygun bir şekilde birleştirildiğinde ve sinir sistemi içine gömüldüğünde büyük hesaplama gücünü gerçekleştirebilecekleri şeklindedir.

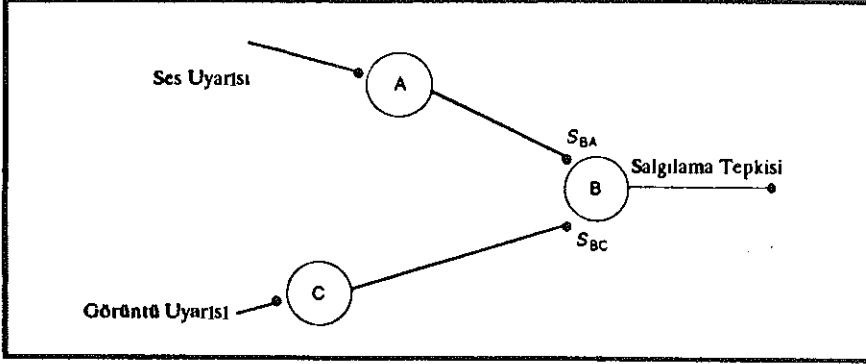
3.2. Hebbian Öğrenimi

Biyolojik sinir sistemlerinin tüm bilgi ve yetenekleri ile önceden programlanarak doğmadığı açıktır. Zaman içerisinde yaşanan öğrenme süreci sonucunda, sinir sistemi ağı kendisini sürekli düzenleyerek, yeni bilgileri bünyesine katmaktadır. Ancak bu öğrenme sürecinin nasıl gerçekleştiği, karşılaşılan en önemli sorudur. 1949 yılında **Donald Hebb** tarafından yayınlanan *Organization of Behaviour* (Davranışın Organizasyonu) isimli kitapta sunulan ve günümüzde çeşitli YSA mimarilerinde kullanılan bir model, öğrenme sürecinin nasıl gerçekleştiğine cevap arayan önerilerden birisidir. Bu modeldeki temel fikrin kolayca anlaşılabilmesi için, psikolojinin temel deneylerinden birisi olan Pavlov Deneyi'nin hatırlanmasında yarar bulunmaktadır.

Psikoloji kaynaklarında **Öğrenimde Klasik Koşullama** örneği olarak gösterilen bu deney, 1849-1936 yılları arasında yaşayan Rus fizyoloğu İvan Pavlov tarafından köpekler üzerinde gerçekleştirilmiştir. Doğal uyarıcı olan etin verilmesi, köpeğin doğal tepki sonucunda salgılamada bulunmasına neden olmaktadır. Deneyde ise, bir zil sesinden bir kaç saniye sonra köpeğe et verilmiştir. Başlangıçta zil sesine her hangi bir salgılamada bulunmayan köpek, deneyin defalarca tekrarı sonucunda zil sesine tepki göstererek salgılamada bulunmuş, Pavlov köpeğin zil sesine yaptığı salgılama davranışını **Koşullu Tepki** (*Conditioned Response*) olarak isimlendirmiştir. [CÜC,140-141]

Zil sesi ve etin verilmesi sonucunda uyarılan sinir hücrelerinin sırasıyla C ve A ile, salgılama tepkisi veren sinir hücresinin B ile, oluşan iki sinaptik bağlantının ise S_{BA} ve S_{BC} ile simgelandiği üç sinir hücresi ve aralarındaki bağlantı yapısı şekil 1.6. da görülmektedir. [FRE, 16] Doğal uyarıcı olan etin görülmesi sonucunda uyarılan C sinir hücresi, doğal tepki kuralı gereğince salgılama tepkisi veren B sinir hücresini uyaracaktır. Ancak başlangıçta zil sesi gibi A sinir hücresinin uyarılmasını sağlayan ilave bir uyarıcı bulunmadığı için, A sinir hücresinin B sinir hücresini uyarması söz konusu değildir. Buna karşılık, etin gösterilmesi ile uyarılan C sinir hücresinin B sinir hücresini uyarması sırasında, A sinir hücresi zil sesi ile uyarılacak olursa, B sinir hücresinin uyarılmasına A sinir hücresi de katılmış olacaktır. **Hebb**, bu durumun yeterli sayıda tekrarlanması halinde, A sinir hücresi ile B sinir hücresi arasında yeni bir düzenlemenin oluştuğunu, A sinir hücresinin B sinir

hücresi üzerindeki etkisini artırdığını ve et verilmediği durumda dahi A sinir hücresinin B sinir hücresini uyarabildiğini savunmuştur.



Şekil 1.6 : Hebbian Öğrenme Kuralının, Pavlov Deneyi İçin Grafik Gösterimi

1951 yılında Dean Edmonds ve Marvin Minsky bu fikirleri temel alan, elektro-mekanik bir öğrenme makinası geliştirmiştir. Bu çalışma yapay zeka araştırmalarında önemli bir yere sahip olan Minsky'nin, MIT'de hazırladığı *Neural Nets and Brain-Model Problem* (Yapay Sinir Ağları ve Beyin Modeli Problemi) isimli doktora tezine temel oluşturmuştur.

3.3. Perceptron

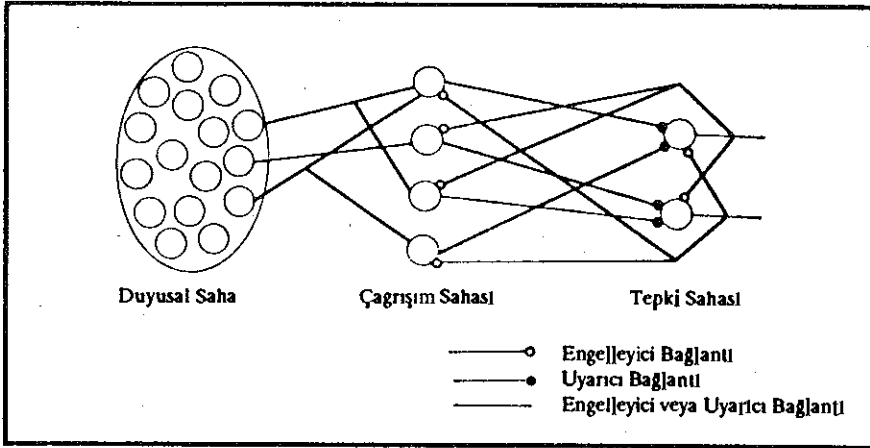
1950'li yılların sonlarında Cornell Üniversitesi'nden psikolog Frank Rosenblatt'ın *Perceptron*⁴ olarak bilinen çalışması, yapay sinir ağlarının gelişiminde ve aynı zamanda duraklamasında önemli bir yere sahip olmuştur.

Rosenblatt, biyolojik sinir ağlarındaki bağlantıların tesadüfi olarak oluştuğunu ileri sürmüş, bu düşüncenin bir sonucu olarak McCulloch-Pitts Modeli gibi sembolik mantığın kullanıldığı önceki analizlerin aksine, en uygun aracın Olasılık Teorisi (*Probability Theory*) olacağını belirtmiştir. Bu düşüncenin paralelinde herhangi bir şekilde tesadüfi olarak birbirine bağlanan ağların genel özelliklerinin tanımlanabilmesi için, İstatistiksel Ayrılabilirlik (*Statistical Separability*) ve Perceptron Öğrenme Kuramını (*Perceptron Learning Theorem*) geliştirmiştir.

⁴ *Perceptron* kelimesinin Türkçe karşılığı, algılayıcı olarak ifade edilebilir.

Şekil 1.7. de örneği görülen **Photoperceptron**, optik örüntülere tepki veren bir cihazdır. [FRE, 22] Bu cihazda ışık, retina yapısı üzerindeki **Duyusal Noktalara (Sensory Points)** vurmakta, her duyuşal nokta gelen ışığa var veya yok şeklinde tepki göstermektedir. Bir duyuşal noktadan gelen tepkiler **Çağrışım Katmanındaki (Association Layer) Çağrışım Düğümlerine⁵ (Associator Units)** aktarılmakla, her çağrışım düğümü, çağrışım düğümlerinin kaynak kümesi olan duyuşal noktaların tesadüfi bir kümesine bağlanmaktadır. Bu bağlantılar uyarıcı veya engelleyici bir işlev yerine getirmekte ve muhtemel değerleri +1, 0, -1 olabilmektedir. Bir Uyarım (*Stimulus*) örüntüsü retinada belirdiğinde, çağrışım düğümü girdilerinin belirli bir eşik değerini aşması halinde sırasıyla,

- Çağrışım birimi aktif hale getirilmekte,
- Bir çıktı üretilmekte,
- Elde edilen bu çıktı bir sonraki katmanda yer alan düğümlere aktarılmaktadır.



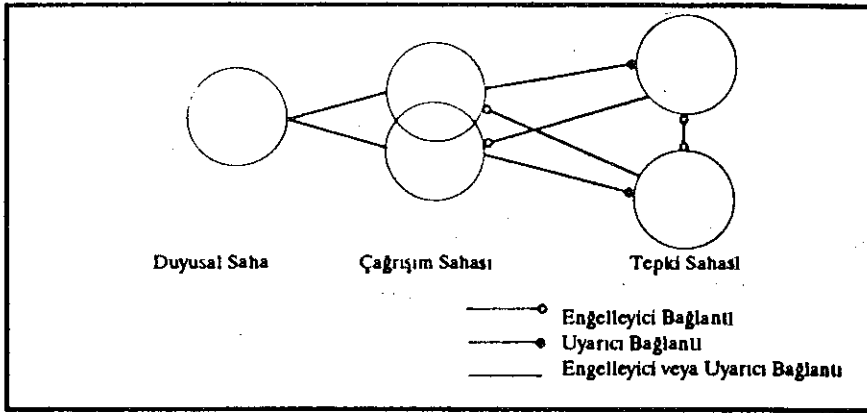
Şekil 1.7 : *Photoperceptron* Cihazının Grafik Gösterimi

Benzer şekilde, çağrışım düğümleri de Tepki Katmanındaki (*Response Layer*) **Tepki Düğümlerine (Response Unit)** bağlantı örüntüsü yine tesadüfi olacak şekilde bağlanmıştır. Ancak tepki katmanından çağrışım katmanına

⁵ **Unit** kelimesinin Türkçe karşılığı birim olmakla birlikte, çalışma içerisinde kavram bütünlüğünü sağlayabilmek amacıyla ile düğüm kelimesinin kullanımı tercih edilmiştir.

engelleyici geri besleme bağlantıları ilave edilmiş ve tepki birimleri arasında engelleyici bağlantılar oluşturulmuştur.

Şekil 1.8. de Venn diyagramları kullanılarak iki tepki düğümünden meydana gelen basit bir *perceptron*'un tüm bağlantı şeması görülmektedir. [FRE, 23] Bu grafik, her tepki düğümünün kendi kaynak kümesinin tamamlayıcısı olan çağrışım düğümlerini ve her tepki düğümünün diğerlerini engellediğini göstermektedir. Bu faktörler retinada beliren her uyarım örüntüsü için bir tek kazanan tepki düğümünün oluşmasına yardımcı olmaktadır. Tepki düğümleri, çağrışım düğümlerinde gerçekleştirilene benzeyen bir şekilde tepki göstermektedir. Tepki düğümü girdileri toplamının, belirli bir eşik değerini aşması halinde çıktı değeri +1, aksi halde -1 olmaktadır.



Şekil 1.8 : Perceptron Cihazının Venn Diyagramları ile Gösterimi

Perceptron ancak belirli bir eğitim sürecinden geçtikten sonra, farklı örüntüleri birbirinden ayırt edebilme yeteneğini kazandığı için öğrenen bir cihazdır. Verilen bir örüntü için *perceptron*'un doğru tepkilerine katılan veya katılmayan çağrışım düğümlerine bağlı olarak, çağrışım düğümlerinin çıktısı arttığı veya azaldığı için, öğrenme kuvvetlendirici bir işlemi gerçekleştirmektedir. Retinaya bir örüntü uygulanmakta ve bir tepki düğümü aktif hale gelinceye kadar, katmanlar arasında uyarım yayılmaktadır. Doğru tepki düğümünün aktif olması halinde, işleme katılan çağrışım düğümünün çıktı değeri artırılmakta, yanlış bir tepki düğümünün aktif olması halinde ise çıktı değeri azaltılmaktadır.

Rosenblatt'm bu çalışması Perceptron Yakınsama Kuramı (*Perceptron Convergence Theorem*) olarak bilinen önemli bir sonucun ispatını sağlamıştır. Kuram iki farklı sınıfın farklı örüntülerini öğrenen tek bir tepki düğümlü *perceptron* için ispatlanmıştır. Sınıflandırma *perceptron* tarafından öğrenilebiliyorsa, anlatılan süreç sonlu sayıdaki eğitim dönemlerinin sonucunda, *perceptron*'un öğrenebileceğini garanti etmektedir.

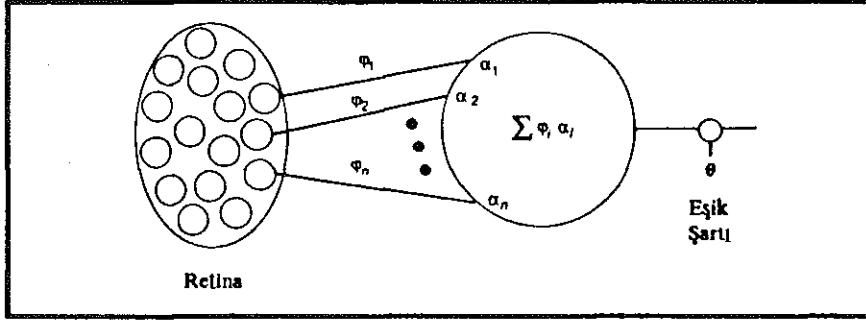
Sistemdeki tepki düğümlerinin sayısına göre, bu veya buna benzeyen bir sistem retinada beliren bir örüntüyü, yeterince birbirine benzeyen örüntüler aynı tepki düğümünü uyuracak şekilde sınıflandırmada kullanılabilir. Ancak farklı örüntü sınıflarını birbirinden başarıyla ayırt edebilecek bir *perceptron* cihazı, sadece 1.4.4. kısmında açıklanan belirli durumlar için söz konusu olabilmektedir.

3.4. Perceptron Kuramının Yıkılışı

1969 yılında Marvin Minsky ve Seymour Papert tarafından yayınlanan *Perceptrons : An Introduction to Computational Geometry* (Perceptron'lar : Hesapsal Geometriye Giriş) isimli kitap, konuyla ilgili bir çok araştırmacı tarafından YSA alanındaki çalışmaların duraklamasına neden olarak gösterilmektedir. Minsky ve Papert, Rosenblatt tarafından önerilen tek katmanlı ve doğrusal *perceptron*'un yetenekleri ve kısıtları üzerine, geçerliliğini günümüzde de koruyan yorumlar yanında, ayrıntılı analizler sunmuşlardır. Yüzlerce örnek üzerinde çalışarak *perceptron* benzeri cihazların öğrenme, uyum gösterme ve kendi kendini organize etme özelliklerinin basit problemler için geçerli, ancak problemler zorlaştığında geçersiz olduğunu matematik yöntemler kullanarak ispatlamışlardır.

Minsky ve Papert *perceptron*'un sadece Doğrusal Ayrılabilir (*Linearly Seperable*) örüntüleri sınıflandırabildiğini, buna karşılık bir çok problemin doğrusal ayrılabilir olmadığını ve bu durumun da *perceptron* için önemli sınırlamalara neden olduğunu özellikle vurgulamışlardır. Minsky ve Papert tarafından önerilen *perceptron* yapısı, Rosenblatt'm önerdiği olasılık yaklaşımını bir kenara iterek, Niceme Mantığı (*Predicate Calculus*) fikrini temel almıştır. Şekil 1.9. da görülen basit *perceptron* yapısı, şekil 1.1. de görülen genel işlem elemanının oldukça benzeridir. Burada sadece ilave olarak

çıkıtıda bir eşik şartı bulunmakta, net girdi değerinin eşik değerinden büyük olması halinde, cihazın çıkıtısı +1, aksi halde 0 olmaktadır. [FRE,25]



Şekil 1.9 : Mantıksal Bağlantıların Perceptron Yapısında Gösterimi

Her biri retinada bir noktayı temsil eden $\Phi = \{\phi_1, \phi_2, \dots, \phi_n\}$ kümesi içerisinde, retinada i. nokta var (*on*) durumunda ise $\phi_i = 1$, aksi halde $\phi_i = 0$ değerini almaktadır. Girdi elemanlarının her birine $\{\alpha\phi_1, \alpha\phi_2, \dots, \alpha\phi_n\}$ kümesinden bir sayı atandığında θ eşik değeri olmak üzere, $\sum_n \alpha\phi_n > \theta$ ise Ψ çıkıtısı 1 değerini almaktadır.

Önermeler mantığı içerisinde x_1 ve x_2 ile gösterilen iki basit önerme, \wedge (ve), \vee (veya), xor gibi mantık işlemcileri kullanılarak bileşik önerme haline getirilebilir. Mantıkta doğruluk tablosu olarak isimlendirilen ve bazı mantık işlemcileri için tablo 1.1. de görüldüğü gibi, mantık işlemcilerinin kullanımı ile elde edilen bileşik önermelerin doğru ya da yanlış olarak yorumlanması, kullanılan mantık işlemcisine bağlıdır. Doğru 1 ile yanlış 0 ile gösterildiğinde, örneğin $(x_1 \wedge x_2)$ bileşik önermesinin doğru olabilmesi için, her iki basit önermenin de doğru olması gerekmektedir.

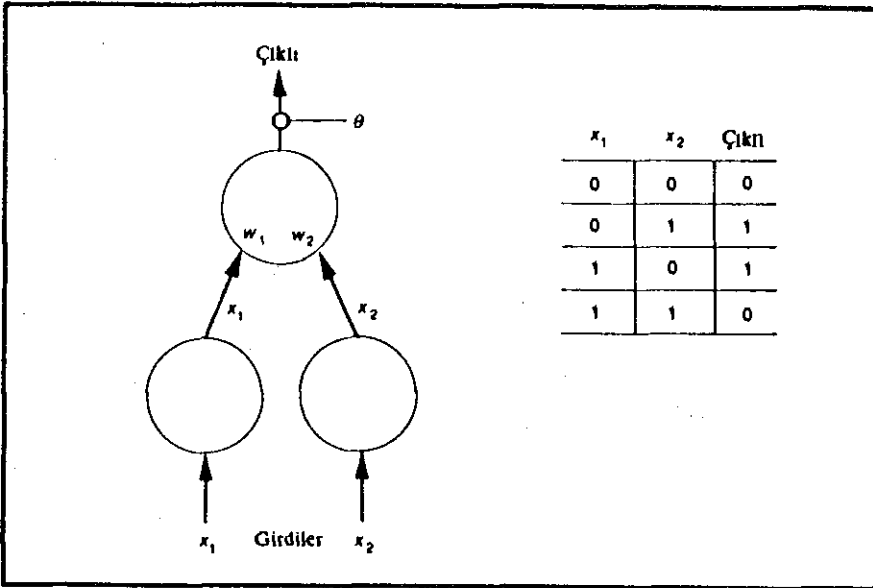
x_1	x_2	$x_1 \wedge x_2$	$x_1 \vee x_2$	$x_1 \text{ XOR } x_2$
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

Tablo 1.1. : Bazı Mantık İşlemcileri İçin Doğruluk Tablosu

Tanımlanan x_1 ve x_2 değerleri için aynı tepki düğümünü uyararak doğru sonucu verecek w_1 , w_2 ağırlık değerlerinin ne olacağı sorunun özünü oluşturmaktadır. *Perceptron* tarafından çözümlenemeyen problemler içerisinde en basitlerinden birisi, xor mantık işlemcisi ile yapılan işlemlerin sınıflandırılmamasıdır. Şekil 1.10. daki ağ yapısında çıktı düğümünün çıktı fonksiyonu, θ eşik değerini göstermek üzere,

$$f(\text{net}) = \begin{cases} \text{net} \geq \theta \Rightarrow 1 \\ \text{net} < \theta \Rightarrow 0 \end{cases}$$

değerlerini alan ve Doğrusal Eşik Düğümü (*Linear Threshold Unit*) olarak isimlendirilen bir eşik fonksiyonudur. [FRE,26]. Şekil 1.10. da 0 veya 1 değerlerini alabilen, x_1 ve x_2 gibi iki düğümüne sahip iki katmanlı ağ yer almaktadır. x_1 ve x_2 düğümleri girdilerinin her ikisinin de 0 veya 1 değerini alması halinde, xor işlemcisi kullanıldığında çıktı değerinin yanlış olması gerekmektedir.



Şekil 1.10 : XOR Mantık İşlemcisinin Perceptron Yapısında İfade Edilmesi

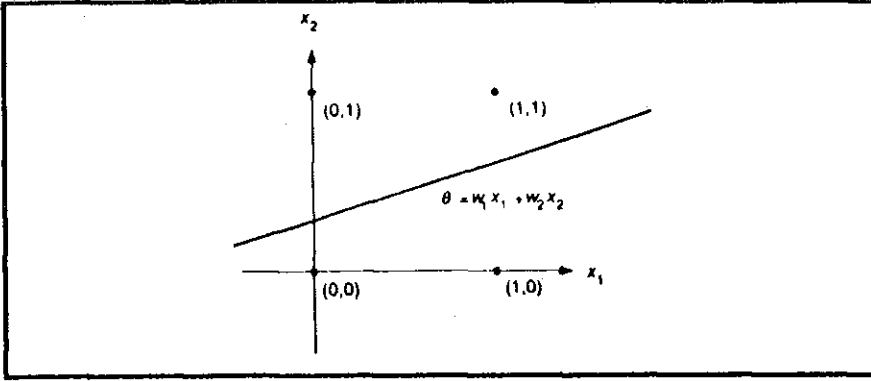
Çıktı düğümünün faaliyeti,

$$\text{net} = w_1x_1 + w_2x_2$$

eşitliğinden hesaplanmakta ve $f(\text{net})$ çıktı değeri

$$f(\text{net}) = \begin{cases} w_1x_1 + w_2x_2 \geq \theta & \Rightarrow 1 \\ w_1x_1 + w_2x_2 < \theta & \Rightarrow 0 \end{cases}$$

durumunda 0 değerini almaktadır. Problem her girdi değeri çifti için ilgili çıktı değerini bulabilecek ağırlık değerlerinin seçilmesidir. Ancak xor problemini aynı anda gerçekleyecek ağırlık değerlerinin bulunabilmesi mümkün değildir.



Şekil 1.11. : XOR Mantık İşlemcisinin x_1, x_2 Düzleminde İncelenmesi.

Şekil 1.11. de x_1, x_2 düzlemi için mümkün girdi değerleri $(0,0), (1,0), (0,1), (1,1)$ şeklinde 4 nokta ile, bu noktaları birbirinden ayırt edebilecek doğrusal eşik fonksiyonu,

$$\theta = w_1x_1 + w_2x_2$$

ile gösterilmiştir. [FRE,27] Şekilde görüldüğü gibi eşik fonksiyonu doğrusu, x_1, x_2 düzlemini iki parçaya ayırmaktadır. Bu durumda bir parçada yer alan mümkün girdi değerlerinin çıktı olarak 1 değerini, diğer parçada ise 0 değerini vermesi gerekmektedir. Şekilden de izlenebileceği gibi xor mantık işlemcisinin kullanılması halinde, x_1 ve x_2 girdi değerlerinin her ikisinin de doğru veya

yanlış olması durumunda, elde edilecek çıktı yanlış olmalıdır. Ancak aynı anda düzlemi iki parçaya bölerek tüm xor işlemlerini gerçekleyebilecek bir doğrunun çizilmesi ve dolayısıyla bu doğruyu sağlayacak w_1 , w_2 değerlerinin bulunabilmesi söz konusu değildir.⁶

3.5. Duraklama Dönemi

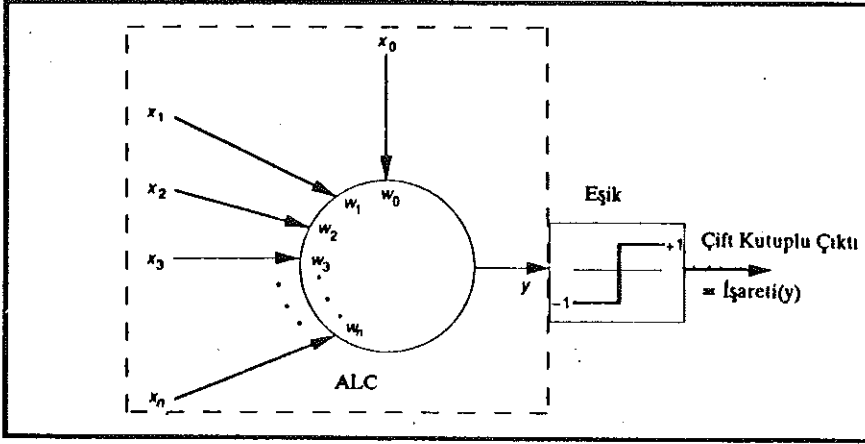
Perceptron 'un başarısız olduğunun Minsky ve Papert tarafından ispatlanması, yapay sinir ağları konusunda çalışan ve büyük hayalleri olan araştırmacıların çöşkusunu yıkararak, yapay sinir ağlarındaki araştırmaların 1970'li yıllara kadar duraklamasına neden olmuştur. Bir çok araştırmacı yapay sinir ağları konusundaki araştırmalarını bırakıp klasik sembolik yaklaşımlara dönerken, bir kaç araştırmacı, ilk *perceptron* çalışmalarının açtığı yolda araştırmalarını direnerek sürdürmüştür.

3.5.1. Adaline

Bu araştırmacıların içerisinde 1956 yılında yapay zekanın bir disiplin olarak doğduğu Dartmouth Konferansı'na da katılan, Stanford Üniversitesinden Bernard Widrow önemli bir yere sahiptir. Widrow ve Stanford Üniversitesi'ndeki öğrencileri, ilk olarak 1960 lı yılların başlarında *Adaline* (*Adaptive Linear Neuron Computer*) projesini gerçekleştirmişlerdir. Negatif geri besleme yapısının kullanıldığı *Adaline*, *perceptron*'un çözümleyemediği bir çok problemin üstesinden gelmiştir. Bu çalışmayı, konuşulan kelimeleri ve görüntü örüntülerini tanıma yeteneğine sahip olan *Madaline* (*Multiple-Neuron*) izlemiş, Widrow ve öğrencisi Hoff, yapay sinir ağlarının eğitiminde halen yaygın olarak kullanılan basit öğrenme kurallarını geliştirmişlerdir.

Adaline teknik olarak bir YSA modeli sayılmayan, fakat YSA modellerinin gelişimine önemli katkılarda bulunan ve tek bir düğümünden meydana gelen bir cihazdır. *Adaline*, *ADaptive LInear NEuron* (Uyum Gösteren Doğrusal Sinir Hücresi) ifadesinin kısaltması olmakla birlikte, YSA modellerinin önemini kaybettiği 1960'lı yılların sonlarında, *ADaptive LINear Element* (Uyum Gösteren Doğrusal Eleman) olarak anılmaya başlanmıştır.

⁶ Bu probleme günümüz YSA modellerinde hiperdüzlem (*hyperplane*) kullanılarak cevap bulunmaktadır.



Şekil 1.12 : Adaline

Şekil 1.12. de genel yapısı görülen Adaline, 1.2. kısımda anlatılan genel hesaplama elemanı ile önemli benzerlikler göstermekte, genel hesaplama elemanı ile Adaline arasında sadece,

- Eğim (*Bias*) değeri olarak ifade edilen w_0 ağırlıklı ve girdi değeri daima 1'e eşit olan ilave bir bağlantının bulunması,
- Çıktı değerinin belirlenmesinde *bipolar* bir şartın bulunması

şeklinde iki farklılık görülmektedir. Şekil 1.12. de tirelerle kutu içine alınan kısım Adaline cihazının, *Adaptive Linear Combiner* (ALC) olarak isimlendirilen bir bölümünü göstermektedir. ALC'nin çıktı değerinin pozitif olması durumunda, Adaline cihazının çıktı değeri +1, negatif olması durumunda ise -1 olmaktadır. ALC bölümünde genel hesaplama elemanının gerçekleştirdiği işlemlerin aynısı yapılmaktadır. ALC, girdi ve ağırlık vektörlerini kullanarak, çarpımların toplamını hesaplamakta ve tek bir çıktı değerinin elde edilebilmesi için bir çıktı fonksiyonunu hesaplanan değere uygulamaktadır. w_0 eğimin ağırlık değerini simgelemek üzere,

$$y = w_0 + \sum_{j=1} w_j x_j$$

eşitliği, yapılan bu hesaplamayı ifade etmektedir. $x_0 = 1$ olması halinde bu denklemi,

$$y = \sum_{j=0} w_j x_j$$

şeklinde veya vektör gösterimi ile

$$y = \mathbf{w}^t \mathbf{x}^7 \quad (1.1)$$

düzeninde ve çıktı fonksiyonu faaliyet fonksiyonuna benzer bir fonksiyonu gösterecek şekilde yazmak mümkün olacaktır.

3.5.2. LMS Öğrenme Kuralı

Girdi katmanında yer alan düğümlere \mathbf{x} girdi vektörü uygulandığında, y çıktı değerini sonuç olarak verecek, \mathbf{w} ağırlıklar kümesinin belirlenmesi daha önce belirtildiği şekilde öğrenme sürecini oluşturmaktadır. Sırasıyla girdi katmanında yer alan düğümlere $\{x_1, x_2, \dots, x_L\}$ girdi vektörleri uygulandığında, hesaplamalar sonucunda her vektörün d_k ($k=1,L$) şeklinde tek bir çıktı değeri olacaktır. Öğrenme sürecinin tamamlanabilmesi için aynı sınıfa ait tüm girdi vektörlerini, çıktı değerleri ile bağlantılı kılacak tek bir ağırlık vektörünün bulunması gerekmektedir. Adaline Cihazı'nda, ilgili ağırlık vektörünün bulunması için kullanılan yöntem En Küçük Kareler Öğrenme Kuralı (*Least Mean Square Learning Rule*) olarak isimlendirilmektedir. Bu yöntemde eğitim kümesinde bulunan girdi vektörleri ve bu vektörlere bağlı olarak elde edilmesi gereken çıktı değerleri, ALC doğru çıktı değerlerini verinceye kadar sisteme uygulanmaktadır. Eğitim olarak isimlendirilen bu uygulama sırasında, uygulanan girdi vektörü için doğru çıktı değerini verecek şekilde, ağırlık vektörü değerleri sürekli olarak yeniden düzenlenmektedir.

x_k girdi vektörlerini, d_k olması gereken çıktı değerlerini, \mathbf{w}^* bu ikisi arasındaki bağlantıyı sağlayacak ağırlık vektörünü temsil ettiğinde ve k . girdi vektörü için tesadüfi bir ağırlık vektörü kullanıldığında, fiili çıktı değeri ile (y_k), elde edilmesi gereken çıktı değeri (d_k) arasında,

$$\epsilon_k = d_k - y_k$$

⁷ Vektör veya matrisleri simgeleyen karakterler **koyu** yazılmıştır. t üssü ilgili elemanın transpozisini göstermektedir.

eşitliği ile ifade edilen bir hata değeri bulunacaktır. L eğitim kümesindeki girdi vektörlerinin sayısını göstermek üzere, hatanın beklenen değeri,

$$\langle \varepsilon_k^2 \rangle = 1/2 \sum_{k=1}^L \varepsilon_k^2 \quad (1.2)$$

olacaktır. (1.1) ve (1.2) eşitliklerinden yararlanarak,

$$\langle \varepsilon_k^2 \rangle = \langle (d_k - \mathbf{w}^t \mathbf{x}_k)^2 \rangle \quad (1.3)$$

eşitliği ve eğitim kümesinin istatistiksel olarak durağan (*Statistically Stationary*) olduğu varsayımı altında,

$$\langle \varepsilon_k^2 \rangle = \langle d_k^2 \rangle + \mathbf{w}^t \langle \mathbf{x}_k \mathbf{x}_k^t \rangle \mathbf{w} - 2 \langle d_k \mathbf{x}_k^t \rangle \mathbf{w} \quad (1.4)$$

eşitliği elde edilir. Girdi İlişki Matrisi (*Input Correlation Matrix*) olarak isimlendirilen \mathbf{R} matrisi ile $\langle \mathbf{x}_k, \mathbf{x}_k^t \rangle$, \mathbf{p} vektörü ile $\langle d_k \mathbf{x}_k^t \rangle$ ve ξ ile $\langle \varepsilon_k^2 \rangle$ ifade edilecek olursa, (1.4) eşitliğini,

$$\xi = \langle d_k^2 \rangle + \mathbf{w}^t \mathbf{R} \mathbf{w} - 2 \mathbf{p}^t \mathbf{w} \quad (1.5)$$

şeklinde yazmak mümkün olacaktır. (1.5) eşitliğinin türevi sıfıra eşitlenirse, beklenen hata değerini minimize edecek \mathbf{w}^* ağırlık vektörü bulunacaktır.

$$\frac{\partial \xi(\mathbf{w})}{\partial \mathbf{w}} = 2\mathbf{R}\mathbf{w} - 2\mathbf{p} \quad (1.6)$$

$$2\mathbf{R}\mathbf{w}^* - 2\mathbf{p} = 0$$

$$\mathbf{R}\mathbf{w}^* = \mathbf{p} \quad (1.7)$$

$$\mathbf{w}^* = \mathbf{R}^{-1} \mathbf{p} \quad (1.8)$$

ξ 'nin skalar bir değer, $\partial\xi(w) / \partial w$ işleminin bir vektör olduğuna dikkat edilmesi gerekmektedir. (1.6) eşitliği ξ değerinin, yine bir vektör olan $\nabla\xi$ gradyenini temsil etmektedir.

$$\nabla\xi = \left[\frac{\partial\xi}{\partial w_1}, \frac{\partial\xi}{\partial w_2}, \dots, \frac{\partial\xi}{\partial w_n} \right]^t \quad (1.9)$$

Örnek

Bütün yapılan işlemler $\xi(w)$ fonksiyonunun eğimini, sıfıra eşitleyecek minimum veya maksimum bir noktayı bulunmasını sağlamaktır. ALC elemanının sadece iki ağırlığa sahip olduğu aşağıda sunulan örnekte $\xi(w)$ fonksiyonunun grafiği paraboloid olacaktır. ξ değeri ağırlık değerlerinin tüm durumlarda pozitif olabilmesi için, beklenen hatanın konkav olması gerekmektedir. Bu sonuç genel bir çıkarım olup, ağırlık vektörünün boyutlarından etkilenmemektedir. Boyutların ikiden fazla olması durumunda ise paraboloid, yerini *hyperparaboloid*'e bırakacaktır.

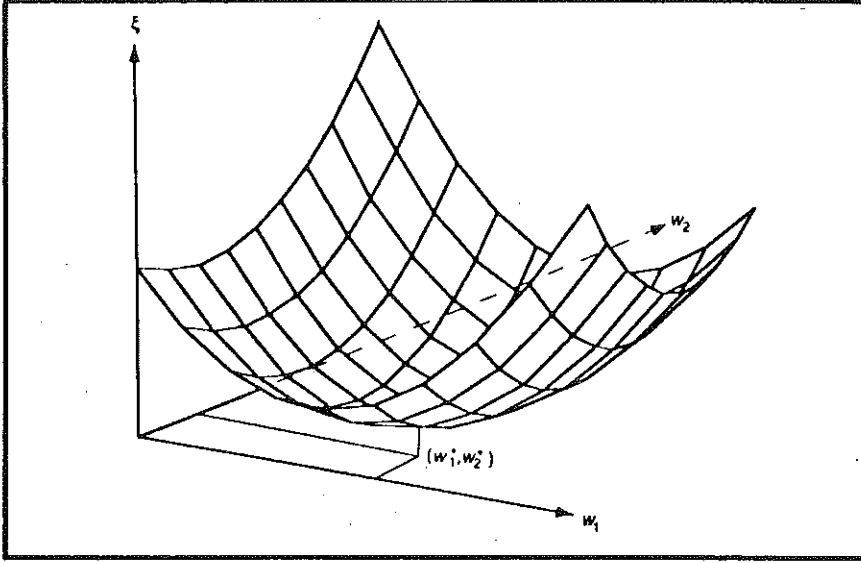
$$R = \begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix} \quad p = \begin{bmatrix} 4 \\ 5 \end{bmatrix} \quad \langle d_k^2 \rangle = 10$$

değerlerinin tanımlandığı 2 girdiye sahip bir ALC göz önüne alındığında, R matrisinin tersini hesaplamak yerine, (1.7) eşitliği kullanılarak optimum ağırlık vektörü bulunabilir.

$$\begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} w_1^* \\ w_2^* \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$$

$$\begin{aligned} 3w_1^* + w_2^* &= 4 \\ w_1^* + 4w_2^* &= 5 \end{aligned}$$

Bu işlemler sonucunda ağırlık vektörü $w^* = (1,1)^t$ şeklinde bulunacaktır. İki ağırlığın bir fonksiyonu olan ξ değerinin grafiği şekil 1.13. de görülmektedir.



Şekil 1.13 : İki Ağırlık Değerinin Bir Fonksiyonu Olan ξ Değerinin Grafik Gösterimi

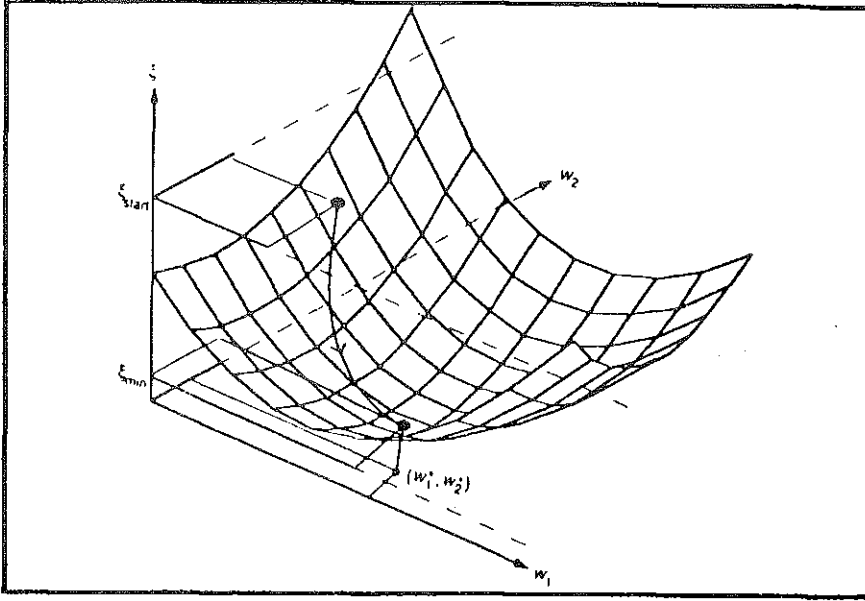
3.5.3. w^* değerinin *Steepest Descent* Yöntemi İle Bulunması

Yukarıda verilen örnekte görüldüğü gibi bir problem için en uygun ağırlık değerlerinin bulunması analitik hesaplamalarla oldukça güçtür. Matris işlemlerinin yapılmasındaki zorluğun yanı sıra, R ve p 'nin her elemanının farklı bir beklenen değerinin bulunması hesaplamaları daha da zorlaştırmaktadır.

En uygun ağırlık değerlerinin bulunmasında daha etkin bir yaklaşım, minimum nokta bulunacak şekilde ağırlık yüzeyinin ALC tarafından adım adım araştırılması olacaktır. Tamamen tesadüfi bir arama etkin olamayacağı için, bu arama sürecine zeka kazandıracak niteliklerin eklenmesi gerekmektedir.

Ağırlık yüzeyinde, ilk ağırlık değerlerinin keyfi olarak belirlendiği bir noktadan başlayarak eğimin (*steepest slope*) yönünü aşağı yönde tanımlayabilmek için, ağırlık değerleri değiştirilir. Şekil 1.14. de görüldüğü gibi ağırlık değerlerini düzenleme işlemi, ağırlık yüzeyinin baştan bilindiği

varsayımından hareket ederek minimum noktaya erişilinceye kadar sürdürülür.



Şekil 1.14 : Minimum Noktanın Araştırılması Süreci

Bu süreç içerisinde ağırlık vektörü sürekli değiştiği için, süreç t zaman adımlarının bir fonksiyonu olarak yazılabilir. Başlangıç ağırlık vektörü $w(0)$, t . zamandaki ağırlık vektörü $w(t)$, t . zaman adımında w vektöründeki değişim $\Delta w(t)$ ile simgelendiğinde, her adımda bir sonraki ağırlık vektörü,

$$w(t+1) = w(t) + \Delta w(t) \quad (1.10)$$

şeklinde yazılabilir. Yüzeydeki her noktada yarıdan iniş yönü araştırıldığı için, bu yönü verecek yüzeyin gradyeninin hesaplanması gerekmektedir. Gradyenin negatif değeri yarıdan iniş yönünde olacaktır. Değişimin büyüklük değeri ise, gradyenin uygun bir η sabiti ile çarpılması sonucunda elde edilecektir. Bu süreç,

$$w(t+1) = w(t) + \eta \nabla \xi(w(t)) \quad (1.11)$$

eşitliği ile ifade edilmektedir. Süreci tamamlamak için gerekli olan işlem, her başardı ardışık adımda önceki kısımda analitik yöntemlerle nasıl bulunulacağı gösterilen $\nabla \xi(w(t))$ değerinin, (1.6) ve (1.9) eşitliklerinden yararlanılarak bulunmasıdır. Ancak w^* değerinin analitik olarak belirlenmesi sırasında R ve p değerlerinin baştan bilinmesi zorunluluğu problemi burada da söz konusudur. Bu bilgi baştan ağırlık yüzeyinin nasıl görüneceğinin bilinmesi ile eşdeğerdedir. Bu problemi ortadan kaldırmak amacıyla, gradyen için aşağıda belirtilen şekilde bir yaklaşım kullanılır.

Sürecin gerçekleştirilmesi sırasında her adımda sırasıyla,

1. x_k girdi vektörü Adaline girdisi olarak uygulanır,
2. ε_k^2 hata değeri, ağırlık vektörünün mevcut değerinden,

$$\varepsilon_k^2(t) = (d_k - w^T(t) x_k)^2 \quad (1.12)$$

eşitliği kullanılarak hesaplanır.

3. $\nabla \xi(t)$ için bir yaklaşım hesaplanır ve bu $\langle \varepsilon_k^2 \rangle$ için bir yaklaşık değer olacak şekilde, $\varepsilon_k^2(t)$ değeri kullanılarak gerçekleştirilir.

$$\nabla \varepsilon_k^2(t) \approx \nabla \langle \varepsilon_k^2 \rangle \quad (1.13)$$

$$\nabla \varepsilon_k^2(t) = -2 \varepsilon_k(t) x_k \quad (1.14)$$

4. (1.11) eşitliğine göre, (1.14) eşitliği kullanılarak ağırlık vektörü yeniden düzenlenir.

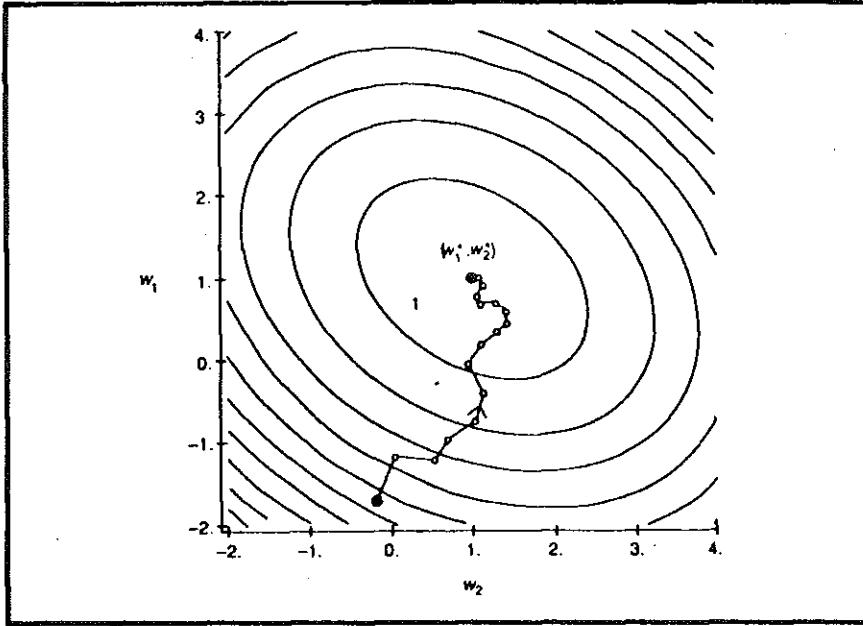
$$w(t+1) = w(t) + 2 \eta \varepsilon_k x_k \quad (1.15)$$

5. Hata kabul edilebilir bir değere varıncaya kadar, diğer girdi vektörleri için ilk 4 adım tekrarlanır.

(1.15) eşitliği LMS algoritmasını ifade etmekte, η parametresi ağırlık vektörünün minimum hala değerine doğru yaklaşımı için sürati ve dengeyi sağlamaktadır. Gradyenin yaklaşık değerinin bulunabilmesi için (1.15) eşitliği

kullanılmaktadır. Şekil 1.15. de (1.15) eşitliği ile LMS algoritması kullanılarak ilgili yolun nasıl bulunduğu görülmektedir.

Ağırlık vektöründeki değişimlerin çok büyük tutulması halinde ağırlık vektörü için minimum noktaların bulunması tesadüfi olacağı için, bu değişimlerin her aşamada η parametresinin katkısı ile küçük tutulması yerinde olacaktır.



Şekil 1.15 : LMS Algoritması ile Minimum Noktanın Araştırılması

3.6. Duraklama Döneminde Gerçekleştirilen Diğer Araştırmalar

Boston Üniversitesi'nden Stephen Grossberg, 1960'lı yıllardan başlayarak Çağrışımsal Öğrenmenin (*Associative Learning*) temellerini oluşturan matematik analizlerini yayınlamıştır. 1977 yılında James Anderson, bağlantı örüntüleri içerisinde pozitif geri besleme çevrimi kullanarak öğrenen *Brain State in a Box* isimli, basit bir Doğrusal Çağrışımsal Model (*Linear Associator Model*) geliştirmiştir.

Aynı dönemde bazı Japon araştırmacıları, yapay sinir ağları konusunda teorik ve uygulamalı çalışmalar gerçekleştirmişlerdir. Tokyo Üniversitesi'nden **Shun-ichi Amari**, ilk olarak Grossberg tarafından önerilen *cooperative/competitive* (ortaklaşa/rekabetli) öğrenme yaklaşımını araştırmış, NHK Laboratuvarları'ndan **Kuniko Fukushima** bulunulan konum ve eksikliklerden bağımsız olarak yazı örüntülerini ve konuşmayı tanıyan **neocognitron** isimli, çok katmanlı *perceptron* modelini geliştirmiştir. Finlandiya'da **Teuvo Kohonen**, Çağrışımsal Bellek (*Associative Memory*) konusunda önemli teorik çalışmalarda bulunmuş, bu çalışmalarını gramer analizi problemlerinin çözümünü sağlayacak yapay sinir ağları uygulamalarında sürdürmüştür.

3.7. Yapay Sinir Ağlarının Yeniden Doğuşu

1980'li yıllarda gerçekleştirilen çeşitli araştırmalar ve faaliyetler, YS Ağlarının yeniden önemli bir araştırma disiplini olarak doğmasını sağlamıştır. Bu gelişmelerin içerisinde en önemlilerini şu şekilde sıralamak mümkündür.

Fizikçi **John Hopfield** 1982 yılında yayınlanan *Neural Networks and Physical Systems with Emergent Collective Computational Abilities* (YS Ağları ve Gelişen Kollektif Hesapsal Yetenekli Fiziksel Sistemler) isimli kitabı ve bunu izleyen makalelerinde, yapay sinir ağlarının basit bir analog devre modelini sunmuş, bu modeli kullanarak analog-ikil dönüştürücü tasarımı, **Gezgin Satıcı** (*Traveling-Salesman*) gibi problemler için çözümler geliştirmiştir. Hopfield elementer özellikli YS Ağlarındaki kollektif hesaplama yeteneklerini göstererek, bu davranışın açıklanması için hesaplama enerjisinin termodinamik teorisini önermiştir.

Carnegie-Mellon Üniversitesi'nden **James McClelland** ve UCSD'den **David Rumelhart** tarafından kurulan, Nobel ödülü sahibi **Francis Crick**'in de katılımıyla güç kazanan, **PDP** (*Parallel Distributed Processing*) araştırma grubu ve bu grubun yayını olan üç bölümlük *Parallel Distributed Processing* (Paralel Dağıtılmış İşlem) isimli çalışma, YS Ağlarının gelişmesinde önemli rol oynamıştır.

The American Physical Society tarafından ilki 1985 yılında 60 katılımcıyla Santa Barbara'da gerçekleştirilen *Neural Networks for Computing* isimli konferans, yapay sinir ağı araştırmalarının, temelde fizik kanunları ve

Bilişim Psikolojisi (*Cognitive Psychology*) teorilerinde birleşmesini sağlamıştır.

1987 yılında San Diego'da 2000 araştırmacının katılımıyla IEEE tarafından düzenlenen, *IEEE First Annual International Conference on Neural Networks (ICNN)* isimli konferans, yapay sinir ağları disiplininin resmi başlangıcı olarak kabul edilmektedir. Bu konferansla başla **Hecht-Nielsen Neurocomputer Corporation, Nestor, Inc., ve Neuronics, Inc.** gibi çeşitli işletmeler tarafından sunulan, ticari yapay sinir ağı ürünleri büyük ilgi uyandırmıştır.

Bu yeni ilgi,

- Yeni YSA modellerinin ve algoritmalarının geliştirilmesi,
- Yeni analog VLSI (*Very Large Scale Integration - Çok Geniş Ölçekli Devre*) devrelerinin geliştirilmesi,
- Etkileyici gösteriler,
- İnsan beynine olan hayranlığın artması

gibi nedenlerle önemli boyutlara varmıştır.

3.8. YAPAY SİNİR AĞI MODELLERİNİN SINIFLANDIRILMASI

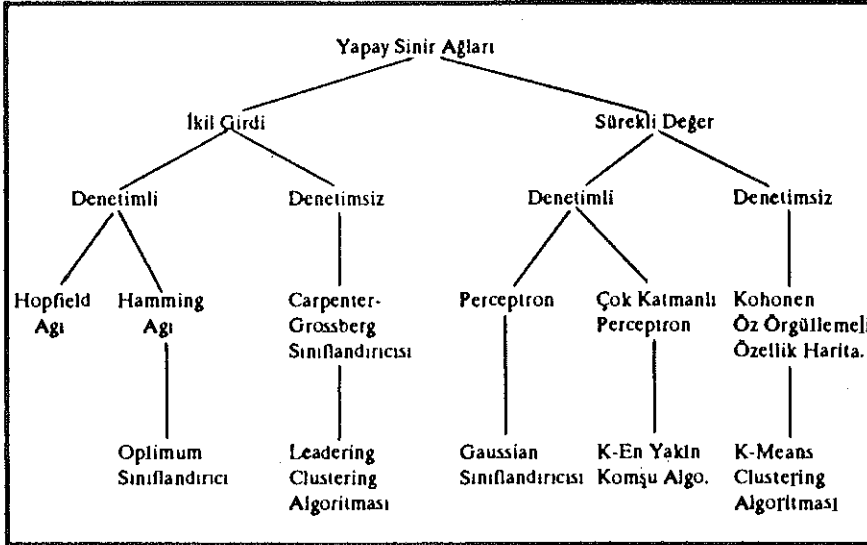
Genel olarak YSA modellerini,

- Ağm yapısına,
 - İleri beslemeli (*feed forward*)
 - Geri beslemeli (*feed back*)
 - Ağırlık matrislerinin simetrik veya asimetrik oluşuna
 - Ağırlık matrisi değerlerinin sabit veya değişken oluşuna,
- Ağda yer alan düğümlerin özelliklerine,
- Kullanılan eşik fonksiyonunun deterministik veya stokastik oluşuna,
- Düğüme sadece analog/ikil veya sürekli değerlerin uygulanabilmesine,
- Eğitim veya öğrenme kurallarına

göre sınıflandırmak mümkündür.

Zamandan bağımsız olan örneklemlerin sınıflandırılmasında kullanılabilen 6 önemli YSA Ağının aile ağacı şekil 1.16. da sunulmuştur. Bu ağaç ilk olarak ikil veya sürekli girdi değerleri ile çalışanlar olmak üzere ikiye ayrılmıştır. Bu ayırım kendi içerisinde eğitimin denetimli veya denetimsiz gerçekleştirilmesine göre tekrar ikiye ayrılmıştır.

Hopfield YSA ve perceptron cihazı gibi denetimli eğitilen ağlar, çağrışimli bellek veya sınıflandırıcı olarak kullanılmaktadır. Denetimli eğitimde her girdi vektörü için, çıktının ne olacağı baştan sisteme tanımlanmaktadır. Eğitim sırasında verilen çıktı değerlerine göre ağ, fiili çıktıyı olması gereken çıktı ile karşılaştırır ve aradaki farkı gidermek için ağırlıklar matrisi elemanlarının değerlerini değiştirir. Ancak bazı araştırmacılar, bir biyolojik sistem içerisinde uygulanan girdilere karşı gelen çıktının ne olacağı baştan bilinemeceği için, bu şekildeki denetimli eğitimin biyolojik kurallara uyumlu olmadığını savunmuş, kendi modellerini bilinen biyolojik kurallarla ters düşmeyecek şekilde kısıtlamışlardır.



Şekil 1.16 : Çeşitli YSA Modelleri

Şekil 1.16. da YSA modellerinin altında yer alan klasik algoritmalar, tanımlanan bu ağlarla benzer veya aynı işlevleri yerine getirmektedir. Örneğin Hamming Ağı, tesadüfi olarak verileri bozulmuş ikil örüntüler için, optimum sınıflandırıcı (*optimum classifier*) klasik algoritmasının YSA uyarlamasıdır. Aynı şekilde ağırlık değerlerinin ve kullanılan eşik fonksiyonlarının uygun olarak seçilmesi halinde, *perceptron Gaussian Sınıflandırıcı*'sının işlevlerini gerçekleştirebilmektedir. Diğer durumlarda ise YSA algoritmaları, klasik algoritmalarından farklılık göstermektedir. Örneğin perceptron yakınsama süreci ile eğitilen perceptronlar, Gaussian Sınıflandırıcısı'ndan farklı davranış göstermektedir. [LP,6]

Şimdiye kadar çeşitli uygulama alanlarında başarılı olan YS Ağlarının sayısı elliye aşmıştır. Tablo 1.2. de yeni ağların geliştirilmesine temel teşkil eden ve/veya günümüzde halen çeşitli uygulama alanlarında etkin bir şekilde kullanılan onüç YSA temel özellikleri ile sunulmuştur.

Klasik von Neumann bilgisayarlarında YSA modellerinin etkin bir şekilde işletimi mümkün olmadığı için, bir çok araştırmacı ve Nestor Inc., Neural Systems Inc., Intel Corp., Neuronics Inc., Texas Instruments, IBM, Hecht-Nielsen Inc. gibi firmalar tarafından bu bilgisayarları etkin hale getirecek çeşitli elektronik kartlar geliştirilmektedir.

Intel tarafından piyasaya sunulan *Intel Neural Network Training System* (Intel Yapay Sinir Ağı Eğitim Sistemi),

- El yazısı karakterlerinin tanınması,
- Radar ve sonar sınıflandırılması,
- Arızaların tanınması,
- Robot kontrolü,
- Tıbbi teşhis,
- Konuşma tanıma

gibi çeşitli uygulama programlarını içermektedir. Bu sistemi elektronik kart olarak destekleyen *ETANN Multi-Chip Board* birbirine bağlantılı olarak çalışan 8 adet Intel 80170NX yongasından (*chip*) meydana gelmekte, her düğüm 128 sinaptik ağırlık içermek üzere, 512 düğüme eşdeğer olarak çalışabilmekte ve saniyede 16 milyar bağlantıyı icra edebilmektedir. [SOL,20]

YSA Adı	Geliştirilenler	İlk Uygulamalar	Sınırları	Yorumlar
Adaptive Resonance Theory	Gail Carpenter Stephen Grossberg	Şekli tanıma, radar veya sonar çıkılının tanımlanması	Bozuk ve eksik verilere karşı duyarlı	Çok karmaşık
Avalanche	Stephen Grossberg	Süreklili söz tanıma. Robot kollarına motor komutları öğretiminde	Hareket dizilerinin aynen tekrarlanması, hızı değiştirmek veya yeni hareket eklemek için basit bir yöntem yok	YSA sınıfı, bütün bu işleri yapabilen tek bir ağ bulunmamakta
Backpropagation (Geriyeye Yayınım Ağı)	Paul Werbos, David Parker, David Rumelhart	Yazılı metinden söz sentezi, robot kollarının kontrolü, çeşitli işletme uygulamaları	Denetimli eğitim, girdi ve çıktı örnekleri çok sayıda olmalı	Günümüzde en yaygın kullanılan YSA, öğrenimi kolay, sonuçları etkin.
Bidirectional	Bart Kasko	İçeriği adreslenebilir bellek	Düşük saklama yoğunluğu, bilgilerin düzgün kodlanmasını gerektirir.	Öğrenimi en kolay ağ, eğitim için uygun.
Boltzmann Makinası Cauchy Makinası	Jeffrey Hinton, Terry Sejnowski, John Hopkins, Harold Szu	Sonar, radar ve görüntüler için şekil tanıma	Uzun eğitim süresi	Basit Ağ Modelleri
Brain State in a Box	James Anderson	Veri bankalarından bilgi çıkarımı	Bir adımda karar verme	Parçalanmış girdileri tanıma
Cerebellatron	Andreas Peltionez, David Mar, James Albus	Robot kolları motor hareketlerini kontrolü	Girdilerde karmaşık kontrollere gerek duymaktadır.	Avalanche Ağına benzer. İstenen türden hareketler elde etmek için, farklı ağırlıklar ile bir çok komut dizisi birlikte kullanılabilir.

YSA Adı	Geliştirilenler	İlk Uygulamalar	Sınırları	Yorumlar
Counter Propagation	Robert Hecht-Nielsen	Görüntü sıkıştırma, istatistik analiz	Çok sayıda Düğüm ve Bağlantı Gereklidir.	Geriyeye Yayınım Ağının Benzeri, daha basit, daha güçsüz
Hopfield	John Hopfield	Bozuk verilerden, bilgi çıkarımı, Karakter Tanıma	Öğrenme yok, ağırlıklar baştan belirlenir	Geniş ölçüde tümeleşik olarak gerçekleştirilebilir.
Madaline	Bernard Widrow	Birden fazla Adaline Düğümünden meydana gelir. Uzak telefon görüşmelerinde ekonun giderilmesi.	Öirdi ve çıktı değerleri arasında doğrusal ilişki varsayımına dayanır.	Öğrenme kuralları oldukça güçlüdür.
Neocognitron	Kunihiko Fukushima	El yazısı karakteri tanıma	Çok sayıda düğüm ve bağlantı gerekir.	Geliştirilen en karmaşık ağ, Kanji karakterlerini tanıyabilir.
Perceptron	Frank Rosenblatt	Karakter tanıma	Kanji gibi karmaşık karakterleri tanıyamaz.	Bilinen en eski YSA Günüümüzde kullanımını yok
Self Organizing Maps	Teuvo Kohonen		Çok fazla eğitime gerek duyar.	Sayısal aerodinamik akış hesaplamalarında bir çok algoritmadan daha üstün

Tablo 1.2. : Geliştirilen Yapay Sinir Ağı Modelleri ve Özellikleri [ÖZM, 28-30]

FAYDALANILAN KAYNAKLAR

- [AKP-4]* AKPINAR Haldun, *Turbo ve UCSD Pascal*, Kipaş Yayım, 1985
- [AKP-6] AKPINAR Haldun, *Yapay Zeka ve Uzman Sistemler*, Yayınlanmamış Araştırma, 1991
- [AKP-8] AKPINAR Haldun, *Daha Hızlı, Daha Güçlü, Daha Yüksek*, Bankalar Birliği Yayını, 1993, ISBN 975-7628-41-7
- [ALS] ALSPECTOR Joshua, *Neural-Style Microsystems that Learn*, IEEE Communications Magazine, Kasım 1989, s. 29-36
- [BRI]* O'BRIEN Stephen, *Turbo Pascal 5.5*, Borland-Osborne/McGraw-Hill, 1989, ISBN 0-07-881501-1
- [CAU] CAUDILL Maureen, *Neural Network Training Tips and Techniques*, AI Expert, Ocak 1991, s. 56 - 61
- [CHA] CHARNIAK Eugene, MCDERMOTT Drew, *Introduction to Artificial Intelligence* Addison Wesley Publishing Company, 1985, ISBN 0-201-11946-3
- [CÜC] CÜCELOĞLU Doğan, *İnsan ve Davranış*, Remzi Kitapevi, 1991, ISBN 975-14-0250-6
- [FIR]* FIREBAUGH Morris, *Artificial Intelligence, A Knowledge Based Approach*, PWS-Kent Publishing Company, 1989, ISBN 0-87835-9
- [FRE] FREEMAN James, SKAPURA David, *Neural Networks : Algorithms, Applications and Programming Techniques*, Addison Wesley Publishing Company, 1991
- [GEN] GENTRY James A., SHAW J. Michael, WHITFORD David T., *Predicting Loan Risk With Systems That Use Either Probit Or An Inductive Learning Approach*.
- [HOL] HOLLAND H. John, *Genetic Algorithms*, Scientific American, Temmuz 1992, s. 66 - 72
- [KAN] KANDEL Eric R., *Principles of Neural Science*, Elsevier Science Publishing Co., 1991
- [KLE] KLEIN Michel, METHLIE Leif B., *Expert Systems, A Decision Support Approach* Addison-Wesley Publishing Company, 1990, ISBN 0-201-17562-2
- [KÖK] KÖKNEK Özcan, *Kavımdan Muthluğa Kişilik*, Altın Kitaplar, 1984, 5. Baskı
- [LEV] LEVINE Robert, DRANG Diane, EDELSON Barry, *A Comprehensive Guide to AI and Expert Systems : Turbo Pascal Edition*, McGraw Hill, 1990, ISBN 0-07-037476-7
- [LIN] LIN Frank, LIN Mei, *Analysis of Financial Data Using Neural Nets*, AI Expert, Şubat 93, s. 37-41

- [LIP] LIPPMANN Richard P., *An Introduction to Computing with Neural Nets*, IEEE ASSP Magazine, Nisan 1987, s. 5 - 23
- [LUG] LUGER George, STUBBLEFIELD William, *Artificial Intelligence and the Design of Expert Systems*, The Benjamin/Cummings Publishing Company, 1989, ISBN 0-8053-0139-9
- [MEY] Meydan Larousse, 3. Cilt, Sayfa 690
- [MOC] MOCKLER Robert J., *Developing Knowledge-Based Systems Using an Expert System Shell*, Maxwell Macmilian International Editions, 1992, ISBN 0-02-946452-8
- [MOR]* MORGAN David P., SCOFIELD Christopher L., *Neural Networks and Speech Processing*, Kluwer Academic Publisher, 1991, ISBN 0-7923-9144-6.
- [MOS] MOSTAFA Yaser S., *Information Theory, Complexity, and Neural Networks*, IEEE Communications Magazine, Kasım 1989, s. 25 - 28.
- [ÖZM] ÖZMETELER Erkin, *Yapay Nöral Ağlar*, İ.T.Ü. Yayınlanmamış Yüksek Lisans Tezi.
- [SCH] SCHWARTZ Tom J., *AI Applications on Wall Street*, IEEE Expert, Şubat 1992, s. 69 - 70
- [SCN] SCHNEIDER Hans-Jochen, *Lexikon der Informatik und Datenverarbeitung*, Oldenbourg, 1983, ISBN 3-486-22661-4
- [SEV] SEVAL Belkıs, SEVAL Selim, *Bankacılıkta Kullanılan Formüller, Oranlar ve Analizler*, Pamıkbank Eğitim Yayınları, 1989
- [SHA] SHAW J. Michael, GENTRY James A., *Using An Expert System with Inductive Learning to Evaluate Business Loans*, Financial Management, Sonbahar 1988
- [SOL] ?, *Intel Develops Neural Network Add-On Board*, Microcomputer Solutions, Temmuz-Ağustos 1992, s. 20
- [STE] STEINMANN Heinrich, CHORAFAS N. Dimitris, *Expert Systems in Banking* New York University Press, 1990, ISBN 0-8147-1449-8
- [WIL] WILLIAMS Noel, *The Intelligent Micro*, McGraw Hill, 1989, ISBN 0-07-707154-9
- [WIT] WITTMANN Alfred, KLOS Joel, *Wörterbuch der Datenverarbeitung* Oldenbourg, 1984 ISBN 3-486-39064-3

* Sembolü bulunan kaynaklardan doğrudan alıntı yapılmamıştır.