

SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**YAPI ANALİZİ VE TASARIMINDA YAPAY
SİNİR AĞLARI**

Yunus DERE
YÜKSEK LİSANS TEZİ
İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI
Konya, 1997

67372

SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

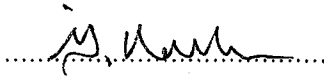
YAPI ANALİZİ VE TASARIMINDA YAPAY SINIR AĞLARI

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

Yunus DERE

YÜKSEK LİSANS TEZİ
İNŞAAT ANABİLİM DALI

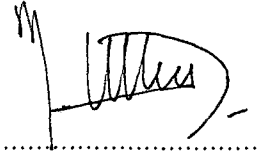
Bu tez 21-01-1997 tarihinde aşağıdaki jüri tarafından kabul edilmiştir.



Doç.Dr.M.Yaşar KALTAKCI
(Danışman)



Prof.Dr.Fuat ERBATUR
(Üye)



Prof.Dr.Mehmet UTKU
(Üye)

ÖZET

Yüksek Lisans Tezi

YAPI ANALİZİ VE TASARIMINDA YAPAY SİNİR AĞLARI

Yunus DERE

Selçuk Üniversitesi
Fen Bilimleri Enstitüsü
İnşaat Mühendisliği Anabilim Dalı

Danışman: Doç.Dr.M.Yaşar KALTAKCI
1997, 126 Sayfa

Jüri : Doç.Dr.M.Yaşar KALTAKCI
.....
.....

Bu çalışmada, temeli yapay zekaya dayanan ve yeni bir hesaplama modeli olan Yapay Sinir Ağları (YSA) ile bunun yapı analizi ve tasarımındaki uygulamaları incelenmiştir.

YSA'nın tanımı ve karakteristikleri özetlendikten sonra uzman sistemler ile karşılaştırılmıştır. YSA uygulamaları geliştirme safhaları ele alınmış ve yapı mekaniğindeki uygulama sahaları ortaya konulmuştur. Hasar tesbiti işlemlerinde, yapısal optimizasyon problemlerinde, malzeme davranışının modellenmesinde, yapıların aktif kontrolünde YSA kullanımının yeri ve önemi anlatılmıştır. Örnek olarak, yüksek kırılganlıkta kesme mukavemetinin belirlenmesi ve Karapınar volkanik agregasından imal edilen, hafif beton hazır döşeme plaklarının taşıma gücünün tesbiti problemlerinin YSA ile tasarımı ve çözümü yapılmıştır. Değişik YSA modelleri için sayısal çözüm sonuçları çizelgeler ve grafikler halinde verilmiş ve metodun etkinliği tartışılmıştır.

Ayrıca, çok katmanlı, ileri yayımlı YSA'nın bilgisayar modellemesini yapan bir bilgisayar programı Turbo Pascal programlama dili kullanılarak geliştirilmiştir.

Anahtar kelimeler: Sinir ağları, sinirsel hesaplama, yapay zeka, uzman sistemler, yapı analizi, paralel işleme, hasar tahmini, yüksek kırılganlık, hafif beton.

ABSTRACT

Master Thesis

NEURAL NETWORKS IN STRUCTURAL ANALYSIS AND DESIGN

Yunus DERE

Selçuk University
Graduate School of Natural and Applied Science
Department of Civil Engineering

Supervisor: Doç.Dr.M.Yaşar KALTAKCI
1997, Page:126

Jury : Doç.Dr.M.Yaşar KALTAKCI
.....
.....

In this work, Neural Networks (NN), a new method of computing which based on artificial intelligence and its applications in structural analysis and design has been investigated.

After summarising description of NN and its characteristics, it is compared with expert systems. Developing phases of NN applications are discussed and structural mechanics applications of NN are explained. Importance of NN applications in damage assessment process, structural optimisation problems, modelling of material behaviour and active control of structures is described. The problems of obtaining shear strength of deep beams and determination of the ultimate strength of lightweight precast slabs manufactured using Karapınar volcanic aggregates are designed and solved using NN as example applications. Numerical results for various NN models are given in tables and graphs and the efficiency of the method is discussed.

Moreover, a computer program for computer simulation of Multilayer, Feed-forward NN is developed in Turbo Pascal programming language.

Key words: Neural networks, neural computing, artificial intelligence, expert systems, structural analysis, parallel processing, damage assessment, deep beams, lightweight concrete.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	I
ABSTRACT.....	II
İÇİNDEKİLER	III
ÇİZELGELER LİSTESİ	V
ŞEKİLLER LİSTESİ	VI
1. GİRİŞ	1
1.1. Konunun Tanımı	1
1.2. Kaynak Araştırması	3
1.3. Çalışmanın Amacı ve Önemi.....	6
2. YAPAY SİNİR AĞLARI	7
2.1. Yapay Sinir Ağlarının Tarihsel Gelişimi	7
2.2. Biyolojik Sinir Ağları ve Nöronun Biyolojik Yapısı.....	11
2.3. YSA'nın Tanımı	12
2.4. Nöron Modeli	13
2.5. YSA'nın Yapısı ve İşlem Elemanı.....	14
2.6. Giriş İşareti Sınıfları	16
2.7. Bağlantı Geometrileri	18
2.8. Ağ Tipleri	19
2.9. Transfer Fonksiyonları	22
2.10. Ağırlık Uzayı.....	24
2.11. YSA'da Eğitim	26
2.11.1. Eğitim algoritmaları	26
2.11.2. Bellek	27
2.11.3. Hata toleransı.....	28
2.12. Öğrenme Kuralları.....	29
2.12.1. Perceptron	29
2.12.2. Çok katmanlı perceptron (Multi-Layer Perceptron).....	32
2.12.3. Hatanın geriye yayılması algoritması ve genelleştirilmiş delta kuralı.....	34
2.12.4. Öğrenme ve momentum katsayıları	40

2.13. Sinirsel Hesaplama	41
3. YAPI MÜHENDİSLİĞİNDE YSA KULLANIMI.....	43
3.1. YSA ve Uzman Sistemler	43
3.2. YSA Uygulamalarının Geliştirilmesi	47
3.2.1. Tasarım	48
3.2.2. Öğrenme	50
3.2.3. Test/Uygulama	53
3.3. Yapı Mekaniğinde YSA'nın Kullanılabileceği Alanlar	57
4. ÖZEL BAZI YSA UYGULAMALARI	59
4.1. Hasar Belirleme İşlemlerinde YSA ve Uzman Sistemler	59
4.1.1. Hasar tesbitine YSA tabanlı yaklaşım	63
4.2. Malzeme Davranışının YSA Kullanımı ile Bilgi Tabanlı Modellemesi.....	70
4.3. Optimizasyon Problemlerinin Çözümünde YSA	71
4.3.1. Yapısal optimizasyon	72
4.3.2. YSA ile optimizasyon	74
4.3.3. Optimizasyonda YSA'nın yeri.....	76
4.4. Bir Yapısal Kontrol Yöntemi Olarak Sinir Ağları.....	76
5. ÖRNEK SAYISAL YSA UYGULAMALARI.....	78
5.1. Yüksek Kirişlerde Kesme Mukavemetinin Belirlenmesi Problemi.....	78
5.1.1. Problemin YSA ile modellenmesi.....	79
5.1.2. Değişik YSA modeli seçeneklerinin eğitilmesi	81
5.1.3. YSA modelinin test edilmesi ve sayısal sonuçlar	87
5.2. Karapınar Volkanik Agregasından İmal Edilen Hazır Döşeme Plaklarının Taşıma Gücünün YSA İle Belirlenmesi	93
5.2.1. Deney numunesi plakların özellikleri	94
5.2.2. Deney düzeni.....	94
5.2.3. Problemin YSA modellemesi	98
5.2.4. YSA modelinin eğitilmesi	101
5.2.5. YSA modelinin test edilmesi ve sayısal sonuçlar	108
6. SONUÇ.....	112
7. EK-1. YSA BİLGİSAYAR PROGRAMI	114
8. KAYNAKLAR.....	125

ÇİZELGELER LİSTESİ

- Çizelge 3.1 Sinir ağları eğitme algoritmaları
- Çizelge 3.2 Yaygın olarak kullanılan sinirsel hesap mimarilerine ait genel bilgiler (Bailey and Thompson 1990)
- Çizelge 3.3 Yaygın olarak kullanılan sinirsel hesap mimarilerinin performans karakteristikleri (Bailey and Thompson 1990)
- Çizelge 4.1 EPADA adlı bir uzman sistemde yapıya ait global hasar düzeyinin puanlandırılması
- Çizelge 5.1 Eğitme işleminde kullanılan yüksek giriş verileri (Siao 1993)
- Çizelge 5.2 Eğitme sırasında kullanılan numunelere ait deneysel sonuçlar ve farklı gizli düğüm sayılarına sahip YSA modellerine ait sonuçlar
- Çizelge 5.3 Eğitme sırasında kullanılan numuneler için farklı gizli düğüm sayılarıyla elde edilmiş hata yüzdeleri (Sütunlardaki maksimum hata yüzdeleri koyu olarak gösterilmiştir)
- Çizelge 5.4 Test işleminde kullanılan yüksek giriş verileri (Siao 1993)
- Çizelge 5.5 Test numunelerine ait deneysel sonuçlar ve farklı gizli düğüm sayılarına sahip YSA modellerine ait sonuçlar
- Çizelge 5.6 Test numuneleri için farklı gizli düğüm sayılarına sahip YSA modellerine ait hata yüzdeleri
- Çizelge 5.7 YSA modeli (gizli katmanda 4 düğüm) ile Strut-and-tie modeline ait eğitme ve test numunesi sonuçlarının karşılaştırılması
- Çizelge 5.8 YSA'nın eğitilmesinde kullanılacak numunelerin özellikleri (Açıkel 1995)
- Çizelge 5.9 YSA'nın test edilmesinde kullanılacak numunelerin özellikleri (Açıkel 1995)
- Çizelge 5.10 Eğitme numuneleri için YSA modellerinden elde edilen sonuçlar ve hata yüzdeleri
- Çizelge 5.11 Eğitme numuneleri için, gizli katman düğüm sayısı 6 olan YSA modeline ait sonuçlar
- Çizelge 5.12 Test numuneleri için YSA modellerinden elde edilen sonuçlar
- Çizelge 5.13 Test numuneleri için YSA modellerinden elde edilen hata yüzdeleri

ŞEKİLLER LİSTESİ

- Şekil 2.1 Biyolojik nöronun/sinir hücresinin şematik yapısı
- Şekil 2.2 Perceptron modeli
- Şekil 2.3 İki katmanlı basit bir YSA örneği
- Şekil 2.4 Genel işlem elemanı yapısı
- Şekil 2.5 Komşu hücrelerin merkez hücreye etkisi
- Şekil 2.6 Tetikleyici ve yasaklanan girişlere sahip bir işlem elemanı
- Şekil 2.7 Giriş katmanında 4, çıkış katmanında 3 işlem elemanı bulunan bir düz ağ
- Şekil 2.8 Fonksiyonel bağlı ağ örnekleri
- Şekil 2.9 Sıkça kullanılan transfer fonksiyonları: (a)Lineer, (b)Rampa, (c)Basamak, (d)Sigmoid
- Şekil 2.10 Üç boyutlu uzayda ağırlık vektörünün değişimi
- Şekil 2.11 Çok katmanlı perceptronda bir işlem elemanının izole edilmiş hali
- Şekil 2.12 (a) Tek katmanlı perceptron yapısı, (b) Bir analog giriş vektörünü A ve B ile gösterilen iki sınıfa ayıran bir tek katmanlı perceptron
- Şekil 2.13 Lineer ayrılabilirliğin gösterimi
- Şekil 2.14 Tek katmanlı perceptron ile iki ve üç katmanlı bulunan çok katmanlı perceptronların sınırlayabileceği karar bölgeleri (Lippmann 1987)
- Şekil 2.15 (n:m:p) mimarisine sahip bir çok katmanlı perceptronun yapısı
- Şekil 2.16 Hatanın geriye yayılması algoritmasının blok diyagramı
- Şekil 2.17 Gizli katmanı olmayan ağırlık hata fonksiyonu
- Şekil 2.18 Gizli katmana ait ağırlık hata fonksiyonu
- Şekil 2.19 Bir tekil işlem elemanı
- Şekil 3.1 YSA ve uzman sistemlerin tarihsel gelişimi
- Şekil 3.2 Giriş veri desenlerinin YSA tarafından okunabilecek şekilde giriş vektörlerine dönüştürülmesi: (a) İkili, (b) Sürekli değer dönüşümü

Şekil 3.3 Ana/baba-evlat oluşumu

Şekil 3.4 YSA uygulamaları geliştirme safhaları: (a) Tasarım , (b) Öğrenme, (c) Test/uygulama

Şekil 4.1 YSA'nın eğitilmesinde kullanılacak olan yapısal tepki verilerinin elde edilmesi için hazırlanmış deney düzeneği (Wu et al. 1992)

Şekil 4.2 Deney düzeneğinden elde edilen ivme-zaman grafiğinin hızlı fourier dönüşümü (FFT, Fast Fourier Transformation) ile ivme şiddeti-frekans grafiğine dönüştürülerek ağa giriş için hazırlanması (Wu et al. 1992)

Şekil 4.3 YSA mimarisi (200:10:3) ve yapı elemanlarının hasar durumu (Wu et al. 1992).

Şekil 4.4 YSA ile optimizasyonun akış diyagramı

Şekil 5.1 Kirişlerde açıklık gerilmesi yayılımı

Şekil 5.2 Kiriş ve yükleme durumu (yalnız alttaki donatı gösterilmiştir.)

Şekil 5.3 Yüksek kiriş probleminin YSA modeli.

Şekil 5.4 Hatanın iterasyon sayısına bağlı değişiminin farklı gizli katman düğüm sayılarına bağlı değişimi

Şekil 5.5 YSA modeli ile Strut-and-Tie modellerinden elde edilen hata yüzdeleri (Test numuneleri için)

Şekil 5.6 Plaklarda donatının yerleştirilmesi

Şekil 5.7 Plaklarda deney düzeni

Şekil 5.8 Plakların moment ve kesme kuvveti diyagramı

Şekil 5.9 Problemin 7:6:1 düzenindeki YSA mimarisi

Şekil 5.10 Probleme ait bazı verilerin ikili düzende gösterilmesi

Şekil 5.11 İterasyon sayısı ile hata yüzdesi arasındaki ilişki

Şekil 5.12 Öğrenme katsayısının iterasyon sayısına bağlı değişimi

Şekil 5.13 Farklı gizli düğüm sayıları için iterasyon sayısına bağlı (a) Ortalama (b) Maksimum (c) Minimum hata grafikleri

Şekil 5.14 YSA modellerinin eğitim seti için bireysel performansları

Şekil 5.15 Test seti için farklı gizli düğüm sayılarının hata yüzdelere etkisi

Şekil 5.16 Farklı gizli düğüm sayılarının test numuneleri üzerindeki bireysel performansları

1. GİRİŞ

1.1. Konunun Tanımı

Bilindiği gibi birçok önemli buluşların temelinde doğanın taklit edilmesi gerçeği yatmaktadır ve bu yüzden insanlar sürekli doğayı taklit etmeye çalışmıştır. Bu taklit etme çabalarının en yeni ürünlerinden biri de Yapay Sinir Ağları yada kısaca YSA dır.

YSA modelleri, bağlı modeller (İng. connectionist models), paralel dağılımlı işlem modelleri (İng. parallel distributed processing models) ve nöromorfik sistemler gibi değişik isimlerle de bilinirler. İsmi ne olursa olsun tüm bu modeller, basit işlem elemanlarının birbirleri ile yoğun bir biçimde bağlanması yoluyla uygun ve yeterli performans elde etmeye çalışırlar. YSA, insan beyninin ve sinir sisteminin davranışlarının (öğrenme, sonuç çıkarma vb.) yapay olarak benzetimi/yapay zeka çalışmalarının bir sonucu olarak ortaya çıkan yeni bir bilgi işleme sistemidir. En genel anlamda bir YSA, insan beynindeki birçok nöronun, yada yapay olarak basit işlemcilerin birbirlerine değişik etki seviyeleri ile bağlanması ile oluşan karmaşık bir sistem olarak düşünülebilir. Yıllardır YSA modelleri konusunda yapılan çalışmaların başta gelen amaçları arasında ses ve görüntü tanıma yeteneklerinin bilgisayarlar yoluyla taklit edilmesi yer almaktadır.

YSA üzerindeki çalışmalar önceleri temel tıp bilimlerinde insan beynindeki nöronların matematiksel modelleme çabaları ile başlamasına rağmen, bugün fizik, matematik, elektrik ve bilgisayar mühendisliği gibi çok farklı bilim dallarında araştırma konusu haline gelmiştir. Son on yıl içerisinde YSA üzerindeki çalışmalar hızla artmaktadır. YSA'nın mühendislik başta olmak üzere birçok bilimsel alanda, karmaşık ve belirsiz veriler altında problemlere çözümler ürettikleri ispatlanmıştır. Bu sistemlerin başlıca belirgin özellikleri algoritmasız, tamamıyla paralel, uyarlanabilen (İng. adaptive), öğrenebilen ve paralel dağıtılmış bir hafızaya sahip olmalarıdır.

YSA lar, olayları öğrenerek karar verme prensibi üzerine kurulmuşlardır. Öğrenme, zeki sistemlerin bilgi yetenek ve tecrübelerini artırma olayı olarak düşünülebilir. Öğrenme “Sistemlerin aynı veya benzer işleri yaptıklarında, o işi veya işleri bir önceki yaptıkları şekilden daha verimli ve etkin olarak değişiklikleri oluşturma süreci”¹ olarak tanımlanabilir.

YSA nın pratik kullanımı genelde, çok farklı yapıda ve formlarda bulunabilen bilgi (İng. information) verilerini hızlı bir şekilde tanımlama ve algılama üzerinedir. Aslında mühendislik uygulamalarında YSA nın geniş çaplı kullanımının en önemli nedeni, klasik tekniklerle çözümü zor problemler (çok sayıda nonlineer değişkene sahip sistemler) için etkin bir alternatif oluşturmastır.

İnsan beyninin çalışma prensipleri tesbit edildikleri veya tahmin edildikleri kadarı ile taklit edilmeye çalışılmaktadır. İnsan beyninin ne olduğu ve nasıl çalıştığı henüz yeterince anlaşılabilmiş değildir. Ancak öğrenme gibi bazı özellikler bilgisayarlaştırılmaktadır. Günümüzde her ne kadar karmaşık matematiksel hesaplamalarını ve hafıza işlemlerini eldeki mevcut bilgisayarlar ile şaşılabilecek şekilde hızlı ve doğru olarak yapmak mümkün ise de aynı bilgisayarlar ile beynin birçok basit fonksiyonunu (görmek, koklamak, vb.) yerine getirmek ya mümkün olmamakta ya da oldukça zor olmaktadır. Aynı şekilde, biyolojik beyin, tecrübe ile öğrenme ve bilgiyi kendi kendine yorumlama, hatta eksik bilgilerden sonuç çıkarma kabiliyetine sahiptir. Bu daha çok biyolojik sistemlerin hücreler üzerinde dağıtılmış bilgiyi paralel olarak işleme özelliklerinden kaynaklanır. Hücreler birbirine bağlı ve paralel çalıştıklarından bazılarının işlevini yitirmesi ile diğerleri çalıştığı için sinir sistemi fonksiyonunu tamamen yitirmez. YSA, bu özellikleri bünyesinde toplayacak şekilde geliştirilmektedir. Bu konuda yapılan çalışmalardan yola çıkılarak insan beyninin fonksiyonlarını nasıl gerçekleştirdiği konusunda da bazı ipuçları elde edilmektedir. YSA üzerine yapılan başarılı benzetim modellerinin nörofizyolojistlerin ortaya attıkları

¹ Simon, H.A. 1983. Why should machine learn. *Machine Learning*, RS Michalski, JG Garbonell (ed.) Morgan Kaufmann Publishing.

varsayımların değerlendirilmesinde büyük faydalar sağladığı açıkça görülmüştür (Öztemel 1996).

1.2. Kaynak Araştırması

Hopfield ve Tank (1985), belirli problemlerin çözümü için ağ yapılandırma ile ilgili genel prensipleri ele almışlardır. YSA'nın hesaplama gücünü göstermek için, zor ama iyi tanımlanmış bir optimizasyon probleminin çözümü için tasarlanmış bir YSA'nın bilgisayar simülasyonlarının sonuçlarını vermişlerdir.

Lippmann (1987), YSA ile hesaplama bir giriş yaparak çeşitli ağ tiplerini ve algoritmaları incelemiş, bunların birbirlerine göre üstünlüklerini ve sakıncalarını tartışmıştır.

- * Karnin (1990), ele alınan bir özel problem için uygun boyutta bir YSA elde etmedeki güçlüklerden bahsederek, bu problemin çözümü için bazı metotlar önermiştir.

Blum ve Li (1991), bilinen türdeki ileri besleme ağlarla gerçek fonksiyonların yaklaşım konusunu incelemişlerdir.

Ghaboussi, Garret ve Wu (1991), kişisel gözlemlerden ve deney verilerinden yola çıkarak türetilen malzeme davranışının matematiksel modelleri ile yapılmakta olan malzeme modellemesi için alternatif bir metot olarak YSA kullanımını ele almışlardır. Çalışmada monotonik iki eksenli ve tek eksenli periyodik basınç yüklemesi altında, düzlem gerilme halindeki beton davranışını YSA ile modellemişlerdir. Malzeme modellemesi için YSA kullanımının önhazırlık sonuçlarını oldukça ümit verici bulmuşlardır.

Hajela ve Berke (1991), sinirsel hesaplama stratejilerinin yapı analizi ve tasarımındaki rolünü araştırmışlardır. Başlıca odak noktaları, statik yapı analizindeki kuvvet deplasman ilişkisini sinir ağları kullanarak ifade etmektir. Sinirsel hesaplama yaklaşımına genel bir bakıştan sonra, sayısal doğruluk ve etkinlik üzerine direk etkisi olan bu tür öğrenme stratejilerinin özellikleri, temsili yapısal optimizasyon problemleri

yoluyla incelenmiştir.

Hornik (1991), rastgele koşullandırılmış ve değişken aktivasyon fonksiyona sahip ve tek bir gizli katmanı olan standart çokkatmanlı ileribeslemeli ağların bile, $L^p(m)$ performans kriterine göre çok genel yaklaşım kabiliyetleri olduğunu göstermiştir. Yeteri kadar düzgün aktivasyon fonksiyonları olan ağların, bir fonksiyona ve onun türevlerine rastgele doğruluk düzeyinde yaklaşabilmelerini mümkün kılan çok genel şartlar vermiştir.

Moselhi, Hegazy ve Fazio (1991), YSA yı yapı endüstrisindeki mevcut otomasyon çabalarını ve uzman sistem uygulamalarını geliştirebilecek ümit verici bir araç olarak tanıtmışlardır. Temel YSA mimarileri ve bunların yapı mühendisliği ve yönetimindeki uygulamaları ele alınmıştır. Bir örnek olarak, optimum fiyat artışı tahmini için bir YSA uygulaması geliştirmişlerdir.

Ghaboussi, Garret ve Wu (1992) de yapı hasarının yeri ve sınırları genellikle gözlemler sonucu belirlenebilmekte ise de, bazı durumlarda gözlemlerin verimli olmadığı belirtilerek, yapı hasarını belirlemek için otomatik gözleme metotlarını amaçlayan bir çalışmanın ilk safhasını gerçekleştirmeyi amaçlamışlardır. Temel strateji, bir sinir ağını, hasarsız bir yapı ile mümkün olan çeşitli hasarlara uğramış bir yapının davranışını tanımak üzere eğitmektir. Yapısal tepki ölçümleri bu eğitilen ağa verildiğinde, mevcut hasarı belirlemek mümkün olmalıdır. Bu temel düşüncelerini örnek bir minyatür yapıda denemişler ve sonuçları ümit verici bulmuşlardır.

Berke, Patnaik ve Murthy (1993), yapay sinir ağlarının yapı tasarımı uzmanlığını elde etmek için kullanımını göstermişlerdir. Bir halka ve iki uçak kanadı için statik ve dinamik sınırlamalar altında optimizasyon kuralları kullanılarak bir optimum tasarımlar kümesi üretilmiştir. Yapay sinir ağları kullanımı ile optimum tasarımların tahmini, elde edilen tasarım parametrelerinin çoğu için yeterli bulunmuştur.

Erbatur (1993), yapısal optimizasyon alanındaki en son gelişmeleri ve yenilikleri ele almıştır. Yapısal optimizasyon araştırmalarının yeni metotlara açık olduğundan ve bu yeni metotlardan biri olarak YSA dan bahsetmiştir.

Siao (1993), yüksek kirişlerde ve kazık başlıklarında kesme mukavemeti davranışını kafes modeli (İng. Strut-and-tie Model) kullanımı ile incelemiştir.

Karlık (1994), YSA'nın tanımını, özelliklerini ve tiplerini ele aldıktan sonra, çok fonksiyonlu protezler (yapay kol, bacak vb.) için YSA kullanımıyla miyoelektrik kontrolü incelemiştir.

Ghaboussi ve Joghataie (1995), yeni bir YSA tabanlı kontrol algoritması geliştirerek, bu algoritmayı zemin hareketlerine maruz bırakılan üç katlı bir çerçeve sistemin aktif kontrolünü bilgisayar simülasyonunda test etmişlerdir. Bu başlangıç çalışması sonucunda, sinir ağı tabanlı kontrol algoritmalarının ileride güçlü uyarlanabilir kontrol sistemlerinin gelişimi için ümit vaat ettiği sonucuna varmışlardır.

Gülkan ve Yakut (1995), depremlerde meydana gelen hasarın belirlenmesinde uzman sistem ve sinirsel şebekelerin kullanılmasını incelemişlerdir. Ayrıca, betonarme binaların hasar tesbitine yönelik olarak hazırlanan ve puanlama sistemine dayanan EPEDA olarak adlandırılmış bir uzman sistemi tanıtmışlardır.

Goh (1995), betonarme yüksek kirişlerin kesme mukavemetinin belirlenmesinde, basit geri-yayılımlı YSA modelinin kullanımıyla modellenmesinin uygulanabilirliğini araştırmıştır. Çalışma sonucunda, YSA modeli, diğer bilinen metotlarla yapılan tahminlerden daha güvenilir bulunmuştur.

Adeli (1996), YSA ve genetik algoritmalar gibi yeni hesaplama teknolojilerinin inşaat mühendisliği uygulamalarını ele almıştır. Yazarın danışmanlığı altında sürdürülen ve son yıllarda yapılmış araştırma projeleri örnekleri kısaca özetlenmiştir.

Moselhi (1996), YSA'nın tanımını yaparak özelliklerini ele almış ve YSA'nın inşaat mühendisliği uygulamalarından bahsetmiştir. Ayrıca yapı analizinde YSA kullanımına bir örnek olması bakımından, tekil yükle yüklenmiş bir basit kirişteki momentlerin bulunması problemini incelemiştir.

Öztemel (1996), bilgisayarlarda öğrenme ve YSA'yı sistematik bir biçimde ele alarak, öğrenme türleri, YSA'nın temel özellikleri ve uygulama alanlarını incelemiştir.

Sanad ve Saka (1996), eksenel yüke maruz tekil temel problemi ile betonarme kiriş ve kolon kesitlerinin optimum boyutlandırılması problemlerini YSA kullanımıyla çözmüşlerdir. Sonuçların yeterli doğruluk değerine sahip olduğunu göstermişlerdir.

1.3. Çalışmanın Amacı ve Önemi

Yapay sinir ağları son bir kaç yıl içinde teorik gelişimini tamamlamış ve artık pratik uygulamalarda kullanılmaya başlanmış olan çok yeni bir hesap modelidir. Bu metodun yapı analizi ve tasarımında kullanılması çalışmaları ise, büyük bir hızla devam etmektedir. İnsan beynindeki sinir hücrelerinin öğrenme kabiliyetinin modellenmesi çalışmaları ile ortaya çıkan yapay sinir ağları, sayısal optimizasyon tasarımları için de etkin yetenekleri olan bir metottur.

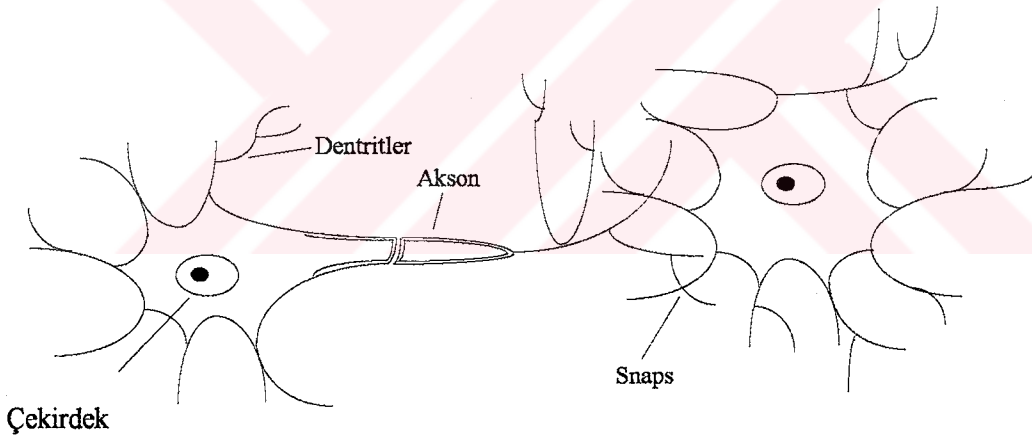
Yapı analizi ve tasarımında bilgisayar kullanımına yeni bir boyut katabilecek olan yapay sinir ağları modelleri, çözümü ve modellenmesi zor non-lineer problemler için etkin bir çözüm metodudur. Bu metotta, probleme ait eldeki veriler (test sonuçları veya önceden başka bir metot ile hesaplanmış değerler) ile YSA eğitilir. YSA öğrendiği verilerden hareketle problem girdileri ve çıktıları arasında bağıntı kurar. Daha sonra, yapay sinir ağına daha önce öğrenmediği bir problemin girdileri verildiğinde, öğrendiklerinden hareketle probleme çözüm bulur.

2. YAPAY SİNİR AĞLARI

2.1. Yapay Sinir Ağlarının Tarihsel Gelişimi

1940’larda Mc Culloch ve Pitts, bir biyolojik nöronun temel fonksiyonlarının basit bir eşik cihazı olarak modellenebileceğini göstermiştir². 1949’da Donald Hebb ise “The Organization of Behaviour” adlı kitabında hücresel seviyede beyinin öğrenme mekanizmasından bahsetmiştir³.

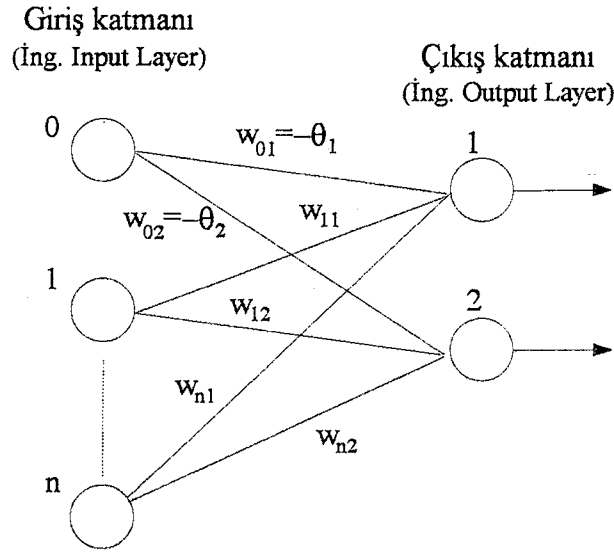
Hebb’in biyolojik öğrenme kuralına göre, bir nöronun dendrit (Şekil 2.1) yoluyla gelen ve bir akson yoluyla alınan giriş onun bir darbe üretmesine sebep olur. Sonraki aksonal girişlerin darbe üretmesi olasılığı artar. Böylelikle yapılan davranışın mükafatı ortaya çıkar.



Şekil 2.1 Biyolojik nöronun/sinir hücresinin şematik yapısı

² Mc Culloch, W.S., Pitts, W.H. 1943. A logical calculus of the ideas immanent in neural nets. *Bull.Math. Biophys.* 5:115-133.

³ Hebb, D.O. 1949. *The Organization of Behaviour*, John WileySons, New York.



Şekil 2.2 Perceptron modeli

Hızlı hesaplama yönelik ilk YSA çalışmaları 1950’li yıllarda başlamıştır. Basit nöron modellerine dayalı bir hesaplama modeli, 1950’lerde Frank Rosenblatt tarafından önerilmiş⁴ ve ardından perceptron diye bilinen tek katmanlı ilk YSA modeli (Şekil 2.2) ortaya çıkmıştır⁵. 1960’lı yıllarda Widrow ve Hoff, bu basit nöron modellerini kullanarak öğrenebilen ilk adaptif sistemler üzerinde çalışmış ve delta kuralı diye bilinen; gerçek çıkış ile istenen çıkış arasındaki farka eşit bir hata terimi kullanarak bağ ağırlıklarının değiştirildiği bir öğrenme kuralını ortaya koymuşlardır⁶. 1969’da Minsky ve Papert, basit perceptron modelinin, yalnızca örnek sınıflarının lineer bir sınırla ayrılabilirdiği örneklemeleri sınıflandırabildiğini ispatlamışlardır. Örneğin bu modelle klasik XOR (“ayrıcıklı veya”) problemine çözüm bulmak imkansızdır. Bu çalışmalarında YSA yardımı ile öğrenme ve hesaplamada aşılması zor engeller olduğunu iddia etmişler ve bu iddia YSA konusundaki çalışmaları önemli derecede yavaşlatmıştır⁷.

⁴ Rosenblatt, F. 1959. *Principles of Neuradynamics*, Spartan Books, New York.

⁵ Rosenblatt, F. 1962. *Principles of Neuradynamics: Perceptrons and The Theory of Brain Mechanisms*, Spartan Books, Washington D.C.

⁶ Widrow, B., Hoff, M. 1960. Adaptive switching circuits. *IRE WESCON Convention Record* 4:96-104.

⁷ Minsky, M. ve Papert, S. 1969. *Perceptrons: An introduction to Computational Geometry*, MIT Press, Cambridge, MA.

1969-1982 yılları arasındaki çalışmalarda ise teori artık oturmuş ve 1982’de J.J. Hopfield tarafından yayınlanan “Neural Networks and Physical Systems” adlı çalışma ile çağdaş YSA devri başlamıştır⁸. Bu çalışmada Hopfield, nöronların karşılıklı etkileşimlerine dayanan bir nöral hesaplama modeli önermiştir. Bu model, bir enerji fonksiyonunu, alabileceği en az değerine indiren 1. mertebe lineer olmayan diferansiyel denklemlerden oluşmuştur. Hopfield, ağ seviyesinde, tek tek nöron seviyesinde varolmayan hesaplama kapasitesinin bulunduğunu öne sürmüştür. Bu tür YSA ya, “Hopfield Ağı” denmektedir. Hopfield’in geri beslemeli YSA modelini ortaya atması⁹ ve bunun pratik optimizasyon problemlerinde kullanılabilirliğini göstermesi (Hopfield 1985), YSA konusundaki çalışmaları yeniden hızlandırmıştır.

1986’da Grossberg, ART yani Uyarlanabilir Rezonans Teorisi (İng. Adaptive Resonance Theory) adında bir YSA yapısını geliştirmiştir¹⁰. ART çok gelişmiş bir YSA dır ve henüz çok fazla probleme uygulanmamıştır. O sıralarda Kohonen’de “Kendi kendini düzenleyen nitelik haritası”nı (İng. self-organizing maps) geliştirmiştir¹¹. Bu YSA, nümerik aerodinamik akış hesaplamaları için çoğu algoritmik yöntemden daha etkili olmuştur.

1986’da Rumelhart ve arkadaşları “Parallel Distributed Processing” (paralel dağılımlı işleme) adlı kitaplarında, ileri beslemeli (İng. feed-forward) modellerde yeni öğrenme modeli olan hatanın geriye yayılması algoritmasını (İng. backpropagation algorithm) geliştirerek, bu konudaki daha önce Minsky ve arkadaşları tarafından iddia edilen aksaklıkların aşılabileceğini göstermişlerdir¹². Bugün endüstride birçok YSA uygulamasında bu öğrenme yöntemi ile bunun değişik varyasyonları kullanılmaktadır.

⁸ Hopfield, J.J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc.Nat.Acad.Sci* 79:2554-2558.

⁹ Hopfield, J.J 1987. Learning algorithms and probability distributions in feed-forward and feed-back networks. *Proc.Nat.Acad.Sci* 88:8429-8433.

¹⁰ Grossberg, S. 1986. *The Adaptive Brain I: Cognition, Learning, Reinforcement, and Rhythm, and The Adaptive Brain II: Visions, Speech, Language, and Motor Control*, Elsevier/North-Holland, Amsterdam.

¹¹ Kohonen, T. 1984. *Self-Organization and Associative Memory*, Springer-Verlag, Berlin.

¹² Rumelhart, D.E., McClelland, J.L., PDP Research Group 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, v. 1: Foundations*, MIT Press, Cambridge, MA.

Back-propagation algoritması, kullanımı çok yaygın olan ve öğrenilmesi kolay bir ağıdır. Bu tez çalışmasında da bu algoritma kullanılmıştır.

İnşaat mühendisliği alanındaki YSA çalışmaları ise şunlardır:

YSA'nın bölgesel bir dergide yayınlanan yapı mühendisliği alanındaki ilk uygulamasında, Adeli ve Yeh¹³, mühendislik tasarımında iç kontrol parametrelerinin kendini ayarlaması ve perceptron kavramına dayalı bir makine öğrenmesi (İng. machine learning) modelini ortaya koymuşlardır. Bu çalışmada, yapı tasarımı problemi, gizli katmanı bulunmayan bir perceptron ile ifade edilebilecek biçime dönüştürülmüştür. Hung ve Adeli¹⁴, mühendislik tasarımında iki katmanlı bir YSA ile perceptron öğrenme algoritması¹⁵ kavramına dayalı bir makine öğrenmesi modelini sunmuşlardır. PERHID diye adlandırdıkları bu modeli, perceptron ile tek katmanlı bir YSA'yı biraraya getirerek meydana getirmişlerdir. Adeli ve Park¹⁶ yapı mühendisliğinde counterpropagation YSA ların¹⁷ uygulamasını gerçekleştirmişlerdir. Backpropagation algoritması ile counterpropagation algoritmasını karşılaştırarak, counterpropagation algoritmasının daha yaklaşık değerler verdiğini ve işlem süresinin de oldukça azaldığını görmüşlerdir.

YSA araştırmalarının birçoğu makine öğrenmesi konusunda yapılmıştır¹⁸. Sinir ağı hesaplamaları optimizasyon için de kullanılabilir. Adeli ve Park, yapıların optimum tasarımı için bir sinirsel dinamik model sunmuşlardır¹⁹. Sinirsel dinamik

¹³ Adeli, H., Yeh, C. 1989. Perceptron learning in engineering design. *Microcomputers in Civil Engineering* 4:247-256.

¹⁴ Hung, S.L., Adeli, H. 1991. A model of perceptron learning with a hidden layer for engineering design. *Neurocomputing* 3:3-14.

¹⁵ Rosenblatt, F. 1962.

¹⁶ Adeli, H., Park, H.S. 1995. Counterpropagation neural networks in structural engineering. *Journal of Structural Engineering, ASCE* 8:1205-1212.

¹⁷ Hecht-Nielsen, R. 1987. Counterpropagation networks. *Applied Optics* 23:4979-4985.

¹⁸ Adeli, H., Hung, S.L. 1995. *Machine Learning - Neural Networks, Genetic Algorithms, and Fuzzy Systems*, John Wiley and Sons, New York.

¹⁹ Adeli, H., Park, H.S. 1995. A neural dynamics model for structural optimization-theory. *Computers and Structures* 3:383-390.

model, ilk defa, bir lineer optimizasyon problemi olan, az katlı düzlem çelik çerçevelerin plastik teoriye göre minimum ağırlık tasarımı problemine uygulanmıştır²⁰.

2.2. Biyolojik Sinir Ağları ve Nöronun Biyolojik Yapısı

Biyolojik sinir ağları başlangıçta sadece nörofizyolojistler tarafından araştırılmıştır. Araştırmaların konusu, genel olarak her bir nöronun (sinir hücresinin) yapısı ve birbirleri ile nasıl ilişkide bulundukları üzerine olmuştur. Yapay sinir ağlarının oluşturulmasında bu çalışmaların faydası inkar edilemez. Sinir sisteminin en basit yapısı olan bir nöronun analizi bu nedenle önem arzeder. Nöron, sinir ağlarının en temel elemanlarından birisi olup sinir sistemi içerisindeki fonksiyon ve görevlerine göre değişik şekil ve büyüklükte olabilir. Bütün nöronların ortak bazı özellikleri bulunmaktadır (Şekil 2.1). Nöron, soma adı verilen hücre gövdesi, bir ucunda bir grup liflere benzer “dentrit” adı verilen ve hücreye diğer hücrelerden veya dış dünyadan bilgiler (sinyaller) getiren bağlantı elemanları, diğer ucunda ise tek bir life benzer “akson” adı verilen ve hücrelerden diğerlerine veya dış dünyaya bilgiler taşıyan bağlantı elemanından müteşekkildir. Bu akson daha sonra diğer hücrelerle birleşme esnasında dallanarak dallara ayrılmaktadır. Bu iki uçtaki bağlantı noktalarının elektrofizyolojik olarak hücrelerdeki bilgileri işlemede önemli yeri vardır. Dentrit tarafından alınan sinyaller hücrede birleştirilerek bir çıkış darbesi üretilip üretilmeyeceğine karar verilir. Eğer bir iş yapılacaksa üretilen çıkış darbesi aksonlar tarafından taşınarak diğer nöronlarla olan bağlantılara veya terminal organlara iletilir. Nöronların birbirleri ile elektrik sinyalleri şeklinde birleştiği belirlenmiştir. Sinyaller bir nöronun aksonundan diğerinin dentritine gönderilir. Bir akson birden fazla dentrit ile ilişkiye girebilir. Bu bağlantının yapıldığı yere “snaps” (Şekil 2.1) denir. Nöronlar elektrik sinyalini hücre duvarındaki voltajı değiştirerek üretirler. Bu ise hücrenin içinde ve dışında dağılmış bulunan iyonlar vasıtasıyla olur. Bu iyonlar sodyum, potasyum, kalsiyum ve klor gibi iyonlardır. Potasyum yoğunluğu nöronun içinde sodyum

²⁰ Park, H.S., Adeli, H. 1995. A Neural Dynamics Model for Structural Optimization-Application to Plastic Design of Structures. *Computers and Structures* 53:391-399.

yoğunluğu nöronun dışındadır. Bir hücre diğer bir hücreye elektrik enerjisini bu kimyasal iyonlar vasıtasıyla transfer eder. Bazı iyonlar elektrik ve magnetik kutuplaşmaya sebep olurken bazıları kutuplaşmadan kurtulup hücre zarını açarak iyonların somaya geçmesini sağlar. Zaten sinyallerin bir hücreden diğerine akmasını sağlayan da bu kutuplaşmanın azalması olayıdır. Sinyaller hücrenin etkinliğini (dürtüsünü) belirler. Bir hücrenin etkinliği, hücreye gelen snaps sayısı, snapslardaki iyonların konsantrasyonu, snapsın sahip olduğu güç olmak üzere üç faktöre bağlıdır. Bir nöron sahip olduğu dürtü miktarınca diğer hücreleri etkiler. Bazı hücreler diğerlerinin dürtülerini pozitif yönde, bazı hücreler de negatif yönde etkiler. İnsan beyni bu şekilde çalışan milyonlarca nöronun bir araya gelmesinden oluşur. Beyinde korteks denilen bölgede her nöronun bir karşılığı vardır. Bir nöronun çıkışı ona bağlı olan bütün nöronlara iletilir. Fakat korteks, işin yapılabilmesi için hangi nöron harekete geçecekse sadece ona komut gönderir.

Biyolojik beynin en önemli özelliklerinden birisi de öğrenme olayıdır. İnsanlar ve hayvanlar sürekli olarak içlerinde bulunduğu çevre ile ilişkiler neticesinde bir öğrenme süreci içerisindeyler. Öğrenilen her yeni bilgi, hemen beynin fonksiyonlarını etkileyerek davranışlarda da kendini gösterir. YSA'nın geliştirilmesinde bu özellik esas teşkil eder.

2.3. YSA'nın Tanımı

YSA, biyolojik sinir ağlarından esinlenerek modellenmiş olup, onlardan çok daha basit bir yapıya sahiptir. O nedenle de anlaşılması daha kolaydır. Geliştirilen birçok YSA, biyolojik sinir ağlarının bilinen temel birkaç özelliğini (öğrenme kabiliyeti gibi) benzetmek (İng. simulation) üzere geliştirilmiştir. Birtakım özellikler ise nörofizyolojik yaklaşımlar yerine mühendislik yaklaşımı ile geliştirilmektedir. Bir YSA'nın yapısını belirleyen bazı faktörler vardır. Yapay nöronlar veya mühendislik deyişiyle işlem elemanları, sinir ağının yapısal dağılışı şekli, ağın sahip olduğu öğrenme kural ve stratejisi bunların başında gelir.

YSA lar sinir mimarilerinin çeşitliliğine rağmen, benzer kısımlardan oluşurlar. Bir sinir ağının genel formunda, nörona benzeyen birime “işlem elemanı” denilmektedir. Ağ, bu elemanların biraraya gelmeleriyle meydana gelmektedir. Bu elemanlar genellikle tamamen veya kısmen bağlı sıralı katmanlar şeklinde düzenlenmişlerdir.

YSA, paralel dağılmış bir bilgi işleme sistemidir. Bu sistem genellikle tek yönlü işaret kanalları (bağlantılar) ile birbirine bağlanan işlem elemanlarından oluşur. Çıkış işareti bir tane olup isteğe göre çoğaltılabilir. YSA yaklaşımının temel düşüncesiyle, insan beyninin fonksiyonları arasında benzerlik vardır. Bu yüzden YSA sistemine “insan beyninin modeli” denilebilir. YSA çevre şartlarına göre davranışlarını şekillileyebilir. Girişler ve istenen çıkışların sisteme verilmesi ile kendisini farklı cevaplar verebilecek şekilde ayarlayabilir. Ancak son derece karmaşık bir içyapısı vardır.

2.4. Nöron Modeli

Nöronlar çok sayıda dentritleri vasıtasıyla giriş uyarısını alır. Dentritlerce alınan bir giriş, harekete geçirici (tetikleyici) veya yasaklayıcı olabilir. Girişler toplanır ve nöron gövdesine yerleştirilir. Bu giriş, belirli bir eşik değerini aştığı zaman, hücre diğer hücrelere aksonu vasıtasıyla bir etki iletir. Nöron davranışı bütün yönleriyle anlaşılmaktan uzak olmasına ve bunun yanı sıra, yukarıda tanımlanan temel fonksiyona ait istisnalar mevcut olmasına rağmen, bu basit anlatım yapay nöron için bir model oluşturur.

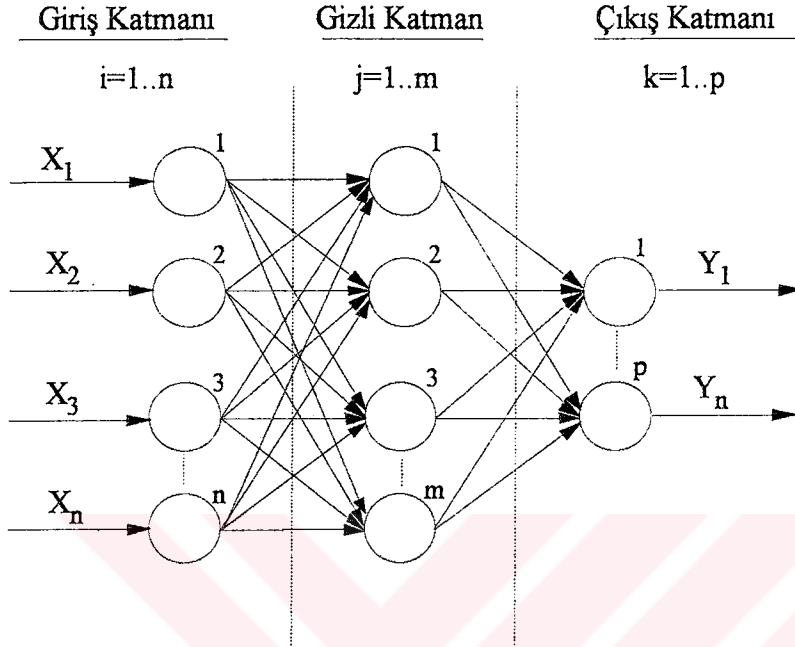
Bir nöron/işlem elemanı, kendine gelen girişleri toplayan ve sadece girişlerin toplamı iç eşik değerini aştığında bir çıkış üreten bir işlem elemanıdır. Bir eşik birimi olarak nöron snapslarındaki işaretleri alır ve hepsini toplar. Eğer toplanan işaret gücü, eşiği geçecek kadar güçlü ise, diğer nöronları ve dentritleri uyarmak üzere akson boyunca bir işaret gönderilir. Kesişen dentritlerde oluşan snapslarca tutulan bütün işaretleri soma toplar. Toplam işaret daha sonra nöronun iç eşik değeri ile karşılaştırılır ve eşik değeri aşmışsa aksona bir işaret yayar. YSA bu basit nöronların (düğümelerin yada ünitelerin) bağlanarak bir ağa dönüştürülmesiyle meydana getirilir.

2.5. YSA nın Yapısı ve İşlem Elemanı

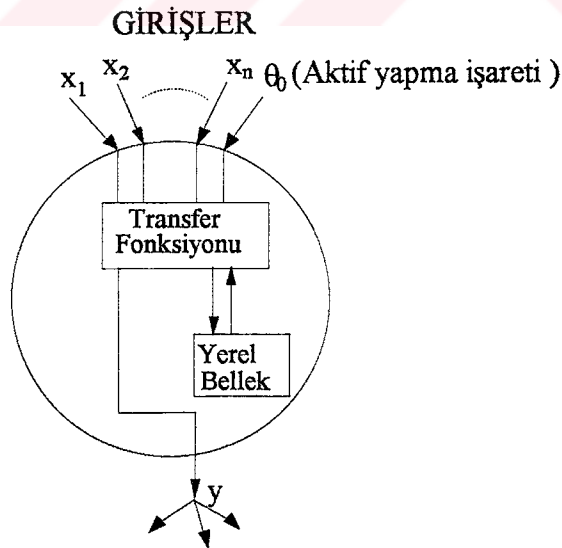
YSA birtakım alt kümelerle ayrılabilir. Transfer fonksiyonları aynı olan elemanlardan oluşan bu altkümelerle “katman/tabaka” (İng. layer) adı verilir. YSA, katmanların birbirine hiyerarşik bir şekilde bağlanmasından oluşmuştur. Bir YSA da verilerin ağa girildiği bir giriş katmanı ve verilen bir girdiye ağın cevabının bulunduğu bir çıkış katmanı vardır. Bazı kaynaklarda, ağa verilerin girildiği bölge olan giriş bölgesi, bir transfer fonksiyonuna sahip olmadığından dolayı bir katman olarak ele alınmamıştır (Moselhi et al. 1991). Giriş katmanı ve çıkış katmanı dışındaki katmanlara gizli katmanlar denilmektedir. Bir işlem elemanı (yapay nöron) kendine ait çıkış değerini belirlemek için toplama (Σ) ve geçiş fonksiyonu (F) gibi işlemler gerçekleştirir (giriş katmanında olanlar bunun dışındadır). Aslında, herbiri önceki katmandaki bir diğer nöronun çıkışını temsil eden girişler kümesi, işlem elemanı tarafından alınır. Herbir giriş kendi bağ ağırlığı (insandaki snaptik güce benzer) ile çarpılır ve bütün ağırlıklı girişler, daha sonra geçiş fonksiyonu ile değişime uğrayacak olan nöron “NET” veya diğer bir deyişle aktivasyon değerini üretmek üzere toplanırlar. Geçiş fonksiyonu, NET’in değeri belirli bir düzeye eriştiği zaman bilgiyi geçen/ileten ayrı bir eşik fonksiyonu olabildiği gibi, NET değerinin sürekli bir fonksiyonu da olabilir. Eğer geçiş fonksiyonu kullanılıyorsa $NET > T$ (T: Eşik değeri, İng. Treshold Value) olduğu zaman işlem elemanı çıkışı bir olur, aksi takdirde çıkış sıfırdır. NET değeri 0 ila 1.0 arasında reel bir sayıdır.

YSA temel olarak, basit yapıda ve yönlü bir şebeke/ağ biçimindedir (Şekil 2.3). Her bir düğüm, hücre denilen n. dereceden lineer olmayan bir devredir. Düğümler işlem elemanı olarak tanımlanır. Düğümler arasında bağlantılar vardır. Her bağlantı tek yönlü işaret iletim yolu olarak görev yapar. Her işlem elemanı istenildiği sayıda giriş bağlantısı ve tek bir çıkış bağlantısı alabilir. Fakat bu bağlantı kopya edilebilir. Yani bu tek çıkış, bir çok hücreyi besleyebilir. Ağdaki tek gecikme, çıkışları ileten bağlantı yollarındaki iletim gecikmeleridir. İşlem elemanının çıkışı isteğe bağlı olarak sürekli/reel veya iki tabanlı bir değeri olan herhangi bir matematiksel tipte olabilir. Kısmen sürekli çalışma konumunda “aktif” halde eleman bir çıkış işareti üretir. Giriş

işaretleri YSA ya bilgi taşır. Sonuç ise çıkış işaretlerinden alınabilir. Şekil 2.4'te genel bir işlem elemanı (nöron, düğüm) gösterilmiştir.



Şekil 2.3 İki katmanlı basit bir YSA örneği

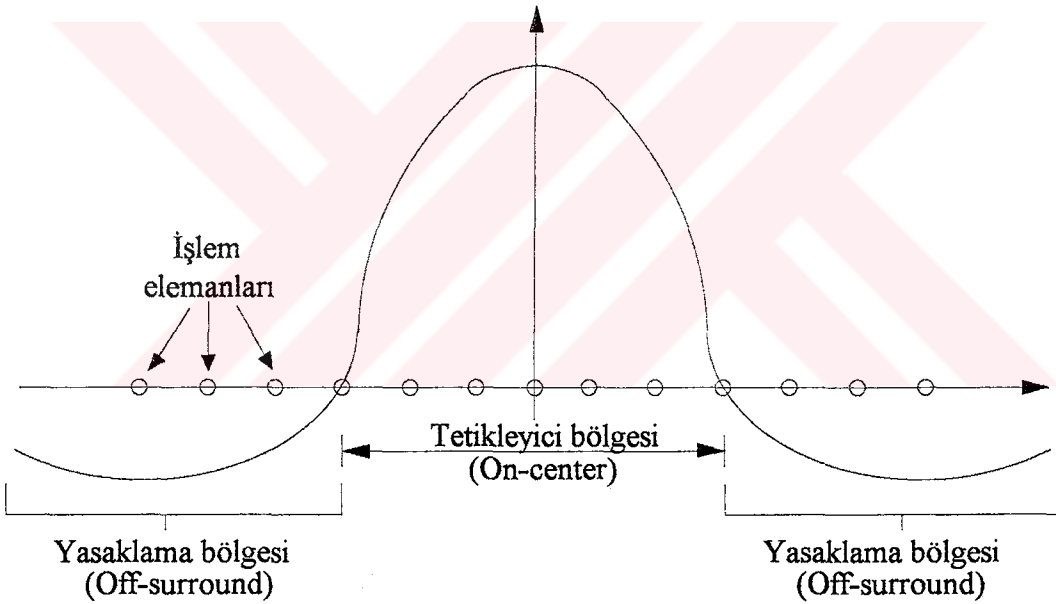


Şekil 2.4 Genel işlem elemanı yapısı

YSA transfer fonksiyonu ve yerel bellek elemanı, bir öğrenme kuralı ile giriş çıkış işareti arasındaki bağlantıya göre ayarlanır. Aktif yapma girişi (İng. bias value) için, bir zamanlama fonksiyonu tanımlanması gerekebilir.

2.6. Giriş İşareti Sınıfları

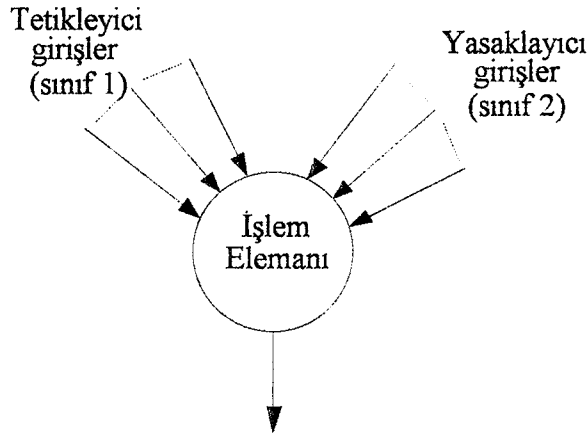
İşlem elemanının transfer fonksiyonu, gelen bütün giriş işaretleri için tanımlanır. Bazen değişik katman davranışlarının farklı olması tabiidir. İşaretlerin hangi bölgelerden geldiğinin bilinmesi gerekir. Değişik bölgelere göre işaretlerin sınıfları tanımlanabilir. Sıkça izlenen bir yapı ise merkezde evet/çevrede hayır (on center/of surround) yapısıdır. Şekil 2.5'te bu yapı gösterilmektedir. Meksika şapkasına benzer bağlantı tipindedir (Karlık, 1994).



Şekil 2.5 Komşu hücrelerin merkez hücreye etkisi

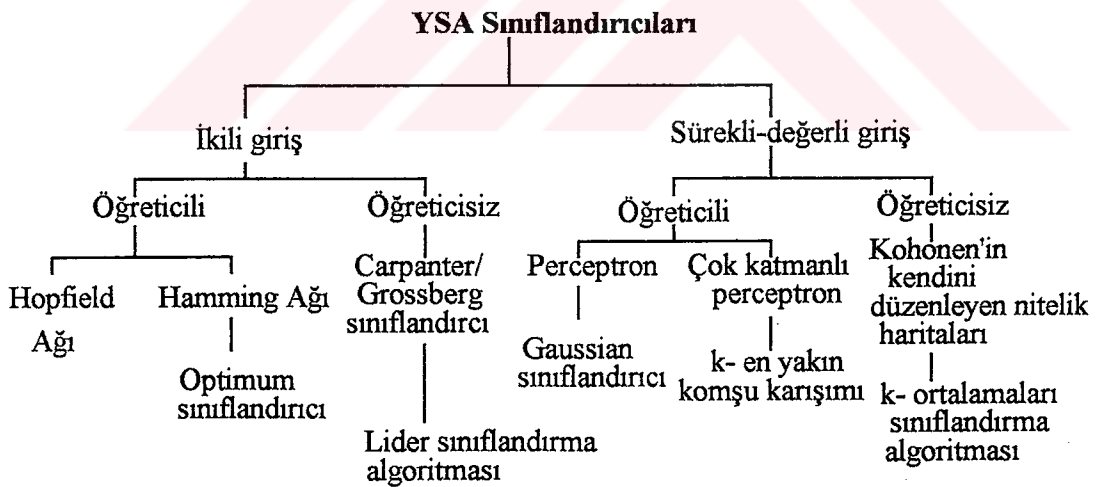
İşlem elemanı tetikleyici girişlerin kendine yakın komşu girişlerden, yasaklanan girişlerini daha uzaktan alır. Böylece işlem elemanına gelen girişler sınıflarına göre

değerlendirilmiş olur. Tetikleyici bölgeden gelen girişler yasaklanan sınıfı oluşturur. Şekil 2.6 böyle bir işlem elemanını göstermektedir.



Şekil 2.6 Tetikleyici ve yasaklanan girişlere sahip bir işlem elemanı

Bir işlem elemanına gelen girişler matematiksel tiplerine göre etiketlenilerek sınıflandırılır. YSA, giriş veri tiplerine göre ikili giriş(0,1) ve sürekli değerli giriş olmak üzere aşağıdaki gibi sınıflandırılır.



Lippmann (1987), statik desenlerin sınıflandırılmasında kullanılabilen altı önemli sinir ağına ait sınıflandırma verilmiştir. Sınıflandırma öncelikle giriş biçimlerine bağlı ikili ve sürekli değerli giriş olmak üzere ikiye ayrılmıştır. Bunun altında ise ağlar, öğreticili ve öğreticisiz eğitilmelerine bağlı olarak sınıflandırılmışlardır. Bu tezdeki örneklerde giriş

işaretleri sürekli-değerli/reel sayı olduğundan dolayı, sınıflandırıcı olarak öğreticili öğrenme ile eğitilen çok katmanlı perceptron kullanılmıştır.

2.7. Bağlantı Geometrileri

Bağlantılarda, taşınan işaret verisinin cinsi tanımlanmalıdır. Bağlantı geometrisi YSA için çok önemlidir, bağlantı işareti her cinsten olabilir. Bağlantının nerede başlayıp nerede bittiğinin bilinmesi gerekir. 1’den n’e kadar olan bir işlem elemanı kümesinin bağlantıları aşağıda tanımlandığı gibi $n \times n$ boyutlu matris biçiminde gösterilebilir.

$$[w_{ij}] = \begin{bmatrix} w_{11} & w_{12} & . & . & w_{1n} \\ w_{21} & w_{22} & . & . & w_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ w_{n1} & w_{n2} & . & . & w_{nn} \end{bmatrix} \quad (2.1)$$

$w_{ij}=w_{ji}=1 \iff i. \text{ işlem elemanı } j. \text{ işlem elemanına bağlı}$

$w_{ij}=w_{ji}=0 \iff \text{bağlı değil}$

En fazla n^2 bağlantı olur. Bağlantılar çeşitli geometrik bölgeler arasında demetler halinde düşünülebilir. Bu bağlantı demetlerinin uyması gereken kurallar şunlardır:

1. Bağlantı demetini oluşturan işlem elemanları aynı bölgeden çıkmalıdır.
2. Bağlantı demetinin işaretleri aynı matematiksel tipten olmalıdır.
3. Bağlantı demetinin işaretleri aynı sınıftan olmalıdır.
4. Bağlantı demetinin bir seçim fonksiyonu (σ) olmalıdır.

$\sigma : T \rightarrow 2^S$ T:Hedef bölgesi s:Kaynak bölgesi

Hedef bölgesindeki her işlem elemanı, kaynak bölgesindeki her elemene giderse “tam (İng. full) bağlıdır”. (Örn: çok katmanlı perceptron). Eğer hedef bölgesi elemanı, n adet kaynak bölgesi elemanına bağlı ise “düzgün dağılmış/üniform” denir. Ayrıca bir elemene, yine bir kaynak elemanı bağlı ise buna “bire-bir bağlı” denir.

2.8. Ağ Tipleri

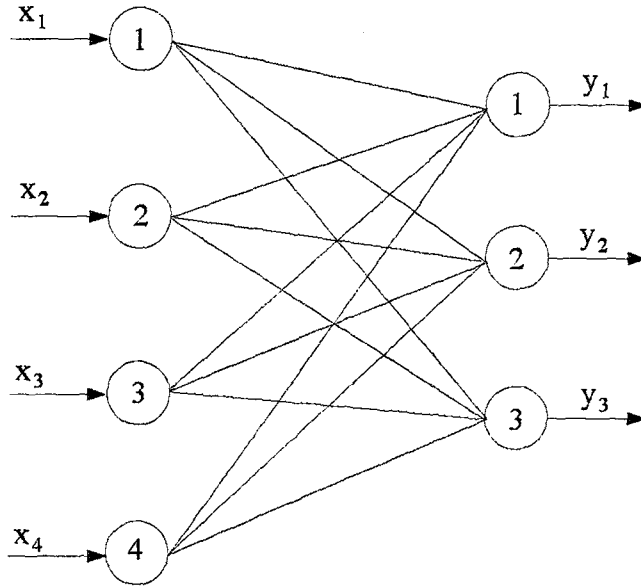
Üç çeşit ağ tipi vardır:

- (1)**İleri Beslemeli Ağ:** Herbir katmandaki hücreler sadece önceki katmanın hücrelerince beslenir ve dolayısı ile bilgi akışı sadece ileriye doğrudur (Şekil 2.3).
- (2)**Kaskat Bağlı Ağ:** Hücreler sadece önceki katmanlardaki hücrelerce beslenir.
- (3)**Geri Beslemeli Ağ:** En az bir hücre sonraki katmanlardaki hücrelerce beslenir. Bir katmandan diğerine veya kendisine doğru uzanan ağırlıklar yoluyla geri besleme bağlantılarına izin veren daha genel bir ağ yapısı gösterir. Bu çalışmada hem ileri, hem de geri beslemeli (hata geriye yayılımı sırasında) ağ tipi birlikte uygulanmaktadır.

Bu ağ tipleri uygulanırken değişik ağ mimarileri de seçilebilir. Bu ağ mimarilerinden bazıları kısaca özetlenecektir.

a) Düz Ağlar

Düz ağlar, girişlerin tüm çıkış birimlerini etkilediği tam bağlı ağlardır. Sadece bir giriş ve bir de çıkış katmanları mevcuttur. Bu tür ağlar giriş ve çıkış büyüklükleri arasındaki doğrusal/lineer ilişkileri temsil etmede iyi bir performans göstermelerine karşın, aktivasyon/geçiş fonksiyonları olmadığından doğrusal olmayan (İng. non-linear) ilişkilerin modellenmesinde yetersiz kalmaktadırlar. Şekil 2.7’de bir düz ağ örneği gösterilmiştir.



Şekil 2.7 Giriş katmanında 4, çıkış katmanında 3 işlem elemanı bulunan bir düz ağ

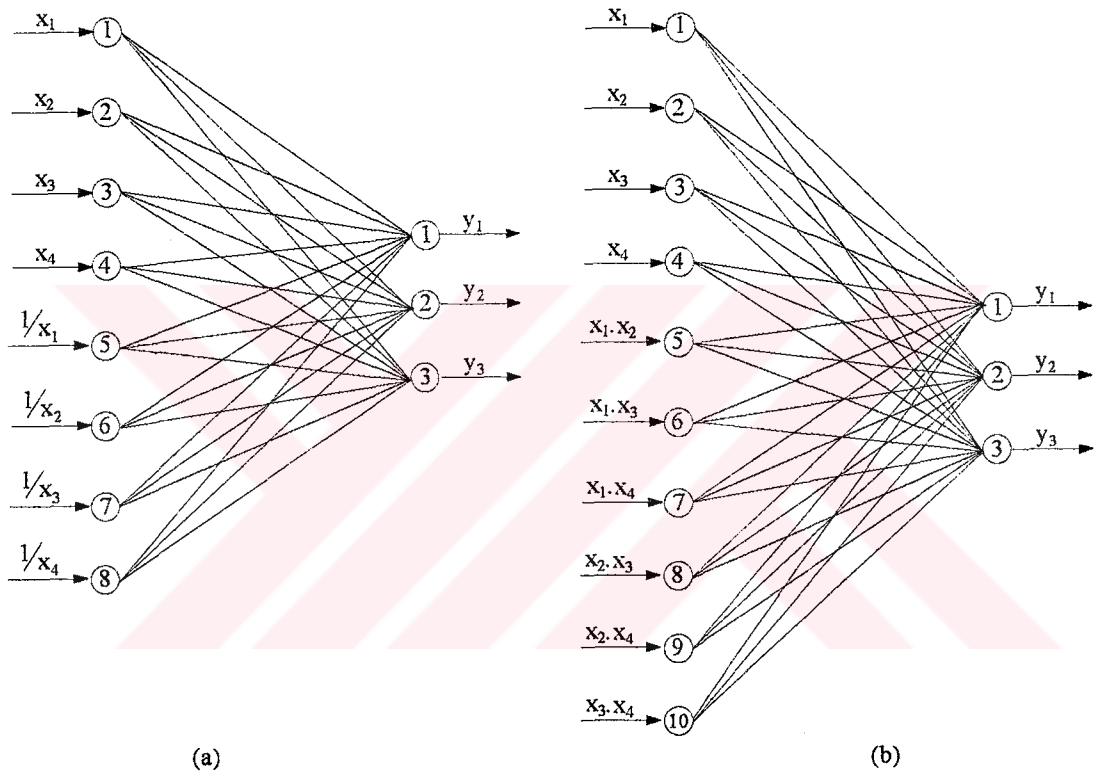
b) Gizli Katmanlı Ağlar

Düz ağlarda karşılaşılan modelleme sorunu, giriş ve çıkış katmanları arasında ilave katmanlar teşkili ile çözülebilir. Bu tür ağlara gizli katmanlı ağlar denilmektedir. Giriş katmanı dışarıdan uyarıyı alır ve çıkış katmanı bu girişe karşılık sinyal verir. Gizli katmanlar dış ortamla ilişkiye geçmez sadece sistem içinden bilgi kabul eder veya verir. Gizli katman sayısı ve gizli katmanda yer alan düğümlerin sayısı, probleme göre değişmektedir. Ancak birçok problemlerden elde edilen tecrübelerle dayanarak gizli katman sayısının minimum yapılması, hesapsal etkinlik açısından oldukça faydalı olmaktadır (Hajela et al. 1991). Doğrusal olmayan/non-linear aktivasyon fonksiyonu içermeyen bu tür ağların, eşdeğeri olan bir düz ağa indirgenmesi mümkündür. Bu tez çalışmasında da doğrusal olmayan ilişkilerin mevcut olmasından ötürü gizli katmanlı ağlar tercih edilmiştir.

c) Fonksiyonel Bağlı Ağlar

Bu tür ağlar, son zamanlarda önerilmiş olan yeni bir yaklaşımdır. Fonksiyonel bağlı ağlar, doğrusal ağlarda karşılaşılan sorunları aşmaya yönelik olarak geliştirilmişlerdir. Mimari, düz ağdakinin aynıdır; ancak, temel giriş büyüklüklerinin lineer olmayan

dönüşümlerinin yer aldığı ilave giriş düğümleri ağı eklenmiştir. Bu tür geliştirilmiş bir giriş setinin, giriş-çıkış verileri arasında düz ağırlarda varolmayan “bağlılık” özelliğini kazandırdığı matematiksel olarak gösterilebilir²¹. Ayrıca, gerçek uygulamalarda bu tür ağlar için öğrenme süresi çok katmanlı ağlara göre daha makul seviyededir. Şekil 2.8a ve b’de Şekil 2.7’de gösterilen düz ağı fonksiyonel bağlı bir ağa dönüştürülmesi gösterilmektedir.

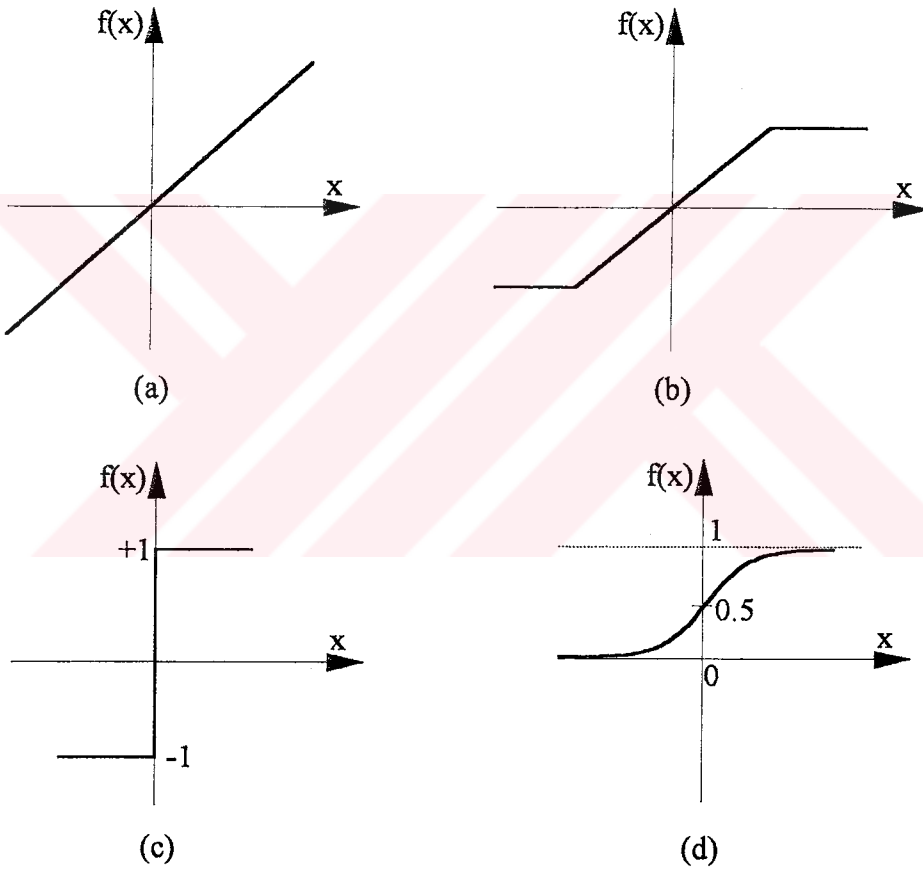


Şekil 2.8 Fonksiyonel bağlı ağ örnekleri

²¹ Sobajic, D.J. 1988. Artificial neural networks for transient stability assessment of electric power systems, PhD. dissertation, Case Western Reserve University, Cleveland, Ohio.

2.9. Transfer Fonksiyonları

Transfer veya işaret fonksiyonları olarak da adlandırılan eşik/aktivasyon fonksiyonları, muhtemel sonsuz giriş kümesine sahip işlem elemanlarından önceden belirlenmiş sınırlar içinde çıkışlar üretirler. Dört tane yaygın olarak kullanılan eşik fonksiyonu vardır. Bunlar rampa, basamak ve sigmoid fonksiyonlarıdır. Şekil 2.9'da bu fonksiyonlar gösterilmiştir.



Şekil 2.9 Sıkça kullanılan transfer fonksiyonları: (a)Lineer, (b)Rampa, (c)Basamak, (d)Sigmoid

Şekil 2.9a’da gösterilen lineer fonksiyonun denklemi aşağıdaki gibidir.

$$f(x)=\alpha .x \quad (2.2)$$

α işlem elemanının x aktivitesini ayarlayan reel değerli bir sabittir. Lineer fonksiyon $(-\tau, +\tau)$ sınırları arasında kısıtlandığında Şekil 2.9b’deki rampa eşik fonksiyonu olur ve denklemi;

$$f(x) = \begin{cases} +\tau & : \text{Eğer } x \geq \tau \text{ ise} \\ x & : \text{Eğer } |x| < \tau \text{ ise (yani } -\tau < x < \tau) \\ -\tau & : \text{Eğer } x \leq -\tau \text{ ise} \end{cases} \quad (2.3)$$

şeklini alır.

$+\tau$ / $-\tau$ işlem elemanının maksimumu/minimumu çoğu zaman doyma seviyesi olarak adlandırılan çıkış değeridir. Eğer eşik fonksiyonu bir giriş işaretine bağlı ise yaydığı $+\tau$ giriş toplamı pozitif, bağlı değilse eşik basamak fonksiyonu $|\delta|$ olarak adlandırılır. Şekil 2.9c, basamak eşik fonksiyonunu gösterir ve denklemi;

$$f(x) = \begin{cases} +\tau & : \text{Eğer } x > 0 \text{ ise} \\ -\delta & : \text{Diğer durumlar} \end{cases} \quad (2.4)$$

şeklindedir.

Bu tez çalışmasında da kullanılan, son ve en önemli eşik fonksiyonu ise, sigmoid fonksiyonudur. Şekil 2.9d’de gösterilen yatık S biçimindeki sigmoid fonksiyonu; seviyeli, doğrusal olmayan/non-lineer çıkış veren, sınırlı, monoton artan bir fonksiyondur ve denklemi;

$$f(x)=\frac{1}{1+e^{-x}} \quad (2.5)$$

biçimindedir.

Her işlem elemanı kendisine gelen bir yerel veriye göre, kendisini ayarlayarak bütün YSA'nın enformasyon bölgesini öğrenmesini sağlar. Yukarıdaki transfer fonksiyonlarını kullanabilmek için, giriş verilerinin gerçek değerlerinin “0” ile “1”

arasındaki bir reel sayıya dönüştürülmesi (İng. normalization) gerekir. (Normalizasyon işlemi: örneğin gerçek değeri 85 olan bir değeri 0.85 şeklinde ağa uygulamaktır).

2.10. Ağırlık Uzayı

Birçok YSA öğrenme işlemi, işlem elemanlarının bağ ağırlıkları değiştirilerek sağlanır. Böylece tanımlanan bu ağırlıklar değiştirilerek, öğrenmede iyi bir model kullanılması ve ağırlıkların bu modele göre değiştirilmesi esastır. Basit bir matematiksel model olarak herbir işlem elemanının “n” adet gerçek ağırlığı olduğu düşünülür ve N adet işlem elemanı gözönüne alınırsa;

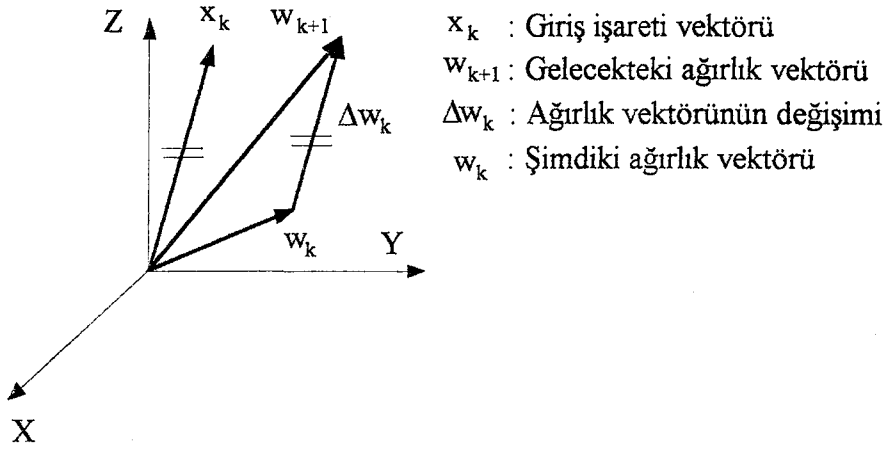
$$w=(w_{11},w_{12},...,w_{1n},w_{21},w_{22},...,w_{2n},w_{N1},w_{N2},...,w_{Nn})^T \quad (2.6)$$

$$w=(w_1^T, w_2^T, w_3^T, ..., w_N^T) \quad (2.7)$$

$w_1, w_2, ..., w_N$: İşlem elemanlarının ağırlık vektörleridir. $[]^T$, matris transpoze işlemini göstermektedir.

$$w_1 = \begin{bmatrix} w_{11} \\ w_{12} \\ \vdots \\ w_{1n} \end{bmatrix} \dots\dots w_N = \begin{bmatrix} w_{N1} \\ w_{N2} \\ \vdots \\ w_{Nn} \end{bmatrix} \quad (2.8)$$

YSA ağırlık vektörü $N \times n$ boyutlu uzayda yayılır. YSA'nın bilgi işleme performansı, ağırlık vektörünün belirli bir değeri ile bulunacaktır. Aşağıda Şekil 2.10'da ağırlıkların düzeltilmesi işleminin vektörel gösterimi yapılmıştır.



Şekil 2.10 Üç boyutlu uzayda ağırlık vektörünün değişimi

Şekilde görüldüğü gibi $\overline{\Delta w_k}$ vektörü, $\overline{x_k}$ vektörü ile aynı doğrultuda olduğunda istenen hata düzeltimini en küçük ağırlık değişimi ile elde etmek mümkündür. Böylece yeni bir giriş örneği uygulandığında, önceki eğitim örneklerinin cevabı en az bozulmuş olur. Hata değişimini inceleyen iki değişik kural vardır.

1- Hata düzeltme kuralları , 2- Gradyen kuralları

Hata düzeltme kuralları; herbir giriş örneğinde ağırlıkları yenileyerek çıktı hatasını en aza indirmeye çalışırlar. Gradyen kurallarında ise, ağırlıklar yeniden ayarlanarak ortalama karesel hata (İng. mean square error) en aza indirilmeye çalışılır.

Ağırlık vektörü ile çalışan YSA da önemli noktalardan birisi, bir öğrenme kuralı geliştirip, enformasyon bölgesi kullanarak, eşik fonksiyonu ile ağırlık vektörü “w” yi, istenilen YSA performansını verecek noktaya yönleltmektir. Genellikle öğrenme kuralı için bir performans yada maliyet fonksiyonu tanımlanır. Minimizasyon veya maksimizasyon ile “w” vektörü bulunur. Bir performans çeşidi olarak bilinen, ortalama karesel ortalama hata şu şekilde tanımlanır.

$$F(w) = \oint_A |f(x) - G(x, w)|^2 \rho(x) dv(x) \quad (2.9)$$

Amaç, $F(w)$ ‘yi küçültmeye çalışmaktır. Burada;

$y = G(w, x)$: Sistemin giriş fonksiyonu

y : Çıkış işareti vektörü

x : Giriş işareti vektörü

w : Ağırlık vektörü

$p(x)$: Olasılık yoğunluk fonksiyonu

2.11. YSA da Eğitim

2.11.1. Eğitim algoritmaları

Eğitim algoritmaları YSA'nın ayrılmaz bir parçasıdır. Eğitim algoritması, eldeki problemin özelliğine göre öğrenme kuralının YSA'ya nasıl adapte edeceğini belirtir. Kullanımı yaygın olan üç çeşit eğitim algoritması vardır.

1-Öğreticili eğitim (İng. Supervised training)

Öğreticili eğitimde, elde doğru örnekler vardır. Yani eğitim örneklerinin tamamı için $(x_1, x_2, \dots, x_n)$ şeklindeki giriş vektörünün, $(y_1, y_2, \dots, y_n)$ şeklindeki çıkış vektörü, tam ve doğru olarak bilinmektedir. Herbir (x_1, y_1) , (x_2, y_2) , $\dots$, (x_n, y_n) çifti için, ağ doğru sonuçları verecek şekilde, seçilen bir öğrenme kuralı yardımıyla eğitilir.

2-Skor ile eğitim (İng. Graded training)

Skor ile eğitimde giriş işaretlerine karşılık gelen çıkış işaretleri tam olarak bilinmemektedir. Çıkış işareti yerine skor verilir ve ağın değerlendirmesi yapılır. Özellikle kontrol uygulamaları için idealdir. Çeşitli maliyet (İng. cost) fonksiyonları kullanılır.

3-Kendini düzenleme ile eğitme (İng. Self-organization training)

Probleme ait veri girişlerine karşın çıkışlar mevcut değildir. Bu yüzden bu tür bir eğitme giriş verilerini gruplandırmak üzere eğitilir ve daha sonra verilen herhangi bir girişin eğitme sınıflarından hangi sınıfa ait olduğunu gösterebilir. Kendini düzenleyen ağ, giriş işaretine göre kendini düzenleyerek organize eder. Olasılık yoğunluk fonksiyonlarına, sınıflandırma ve şekil tanıma problemlerine uygulanabilir.

Ne tür eğitme yöntemi kullanılırsa kullanılsın, herhangi bir ağ için gerekli karakteristik özellik, ağırlıkların verilen eğitme örneğine nasıl ayarlanacağını belirtilerek öğrenme kuralının oluşturulmasıdır. Öğrenme kuralının oluşturulması için bir örneğin ağa defalarca tanıtılması gerekebilir. Öğrenme kuralı ile ilişkili parametreler (öğrenme katsayısı vb.) ağın zaman içinde gelişme kaydetmesiyle değişebilir. Hangi YSA algoritmasında ne tür bir eğitme kullanıldığı bu bölümün giriş işaretlerinin sınıflandırılması kısmında gösterilmiştir.

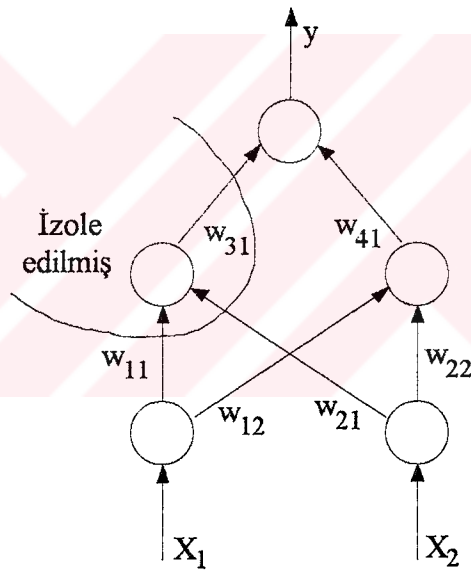
2.11.2. Bellek

YSA'nın önemli bir özelliği bilgiyi saklama şeklidir. YSA da bellek, birçok yerel bellekler (İng. local memory) oluşturularak dağıtılır, bunlar YSA'nın bağ ağırlıklarıdır. Ağırlıkların sahip olduğu değerler, ağın o andaki bilgi durumunu temsil eder. Mesela; bir giriş/istenen çıkış çiftine ait bilgi bloğu ağın içindeki birçok bellek birimine paralel olarak dağıtılmıştır. Bu yüzden eğitilen ağa bir giriş uygulandığında, ağ bu giriş için belleğindeki en yakın çıkışı seçer ve tam girişe bağlı çıkış ortaya çıkar. Eğer YSA oto-ilişkili ise, kısmi giriş vektörlerinin ağa verilmesi ve bu girişlerin tamamlanması ile sonuçlanır. YSA belleğinin yapısı; eksik, gürültülü ve tam seçilemeyen bir giriş uygulandığı zaman bile, mantıklı çıkış üretmeye uygundur. Bu kurala "genelleme" adı verilir. Bir genellemenin kalitesi ve anlamı, uygulama çeşidine, ağın tipine ve karmaşıklığına bağlıdır. Lineer olmayan çok katmanlı ağlar (özellikle geriye yayılım ağları) gizli katmandaki özelliklerden öğrenirler ve bunları çıkışlar üretmek üzere birleştirirler. Gizli katmandaki bilgi, yeni giriş örneklerine akılcı çözümler oluşturmak için kullanılabilir.

2.11.3. Hata toleransı

Klasik hesaplama sistemleri çok az bir zarardan bile etkilenirler. Verilerde eğer bir eksiklik söz konusu olursa, klasik yöntemler çalışmazlar. YSA için durum farklıdır. Bu farklılık YSA'nın hata toleranslı olmasıdır. İyi eğitilmiş ve genelleme kapasitesi yüksek bir sinir ağı kendisine takdim edilen veriler eksik bile olsa karar verme işlemine devam eder. Aynı şekilde YSA üzerinde birtakım problemler ve bozukluklar olabilir.

İşlem elemanlarının az da olsa zarar görmesi sistemin bütününe etkiler. YSA paralel dağılmış parametrelili bir sistem olduğundan her bir işlem elemanı izole edilmiş bir ada olarak düşünülebilir. Şekil 2.11'de çok katmanlı perceptron (MLP) için bu durum gösterilmiştir.



Şekil 2.11 Çok katmanlı perceptronda bir işlem elemanının izole edilmiş hali

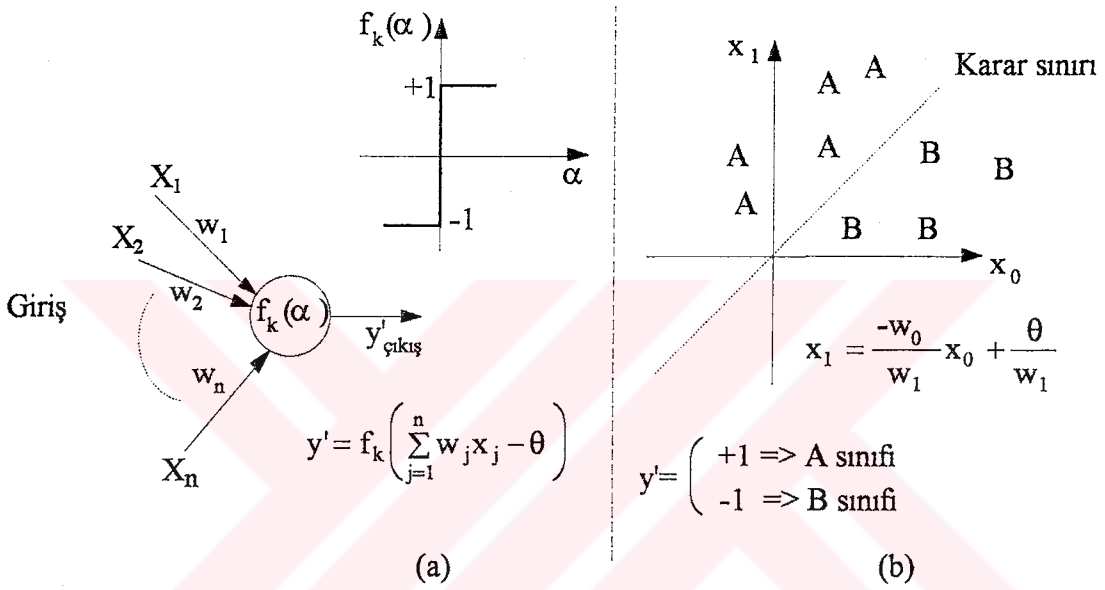
Verilerdeki eksiklik veya YSA'daki yapısal bozukluktan dolayı daha çok işlem elemanının zarar görmesi ile sistemin davranışı biraz daha değişir/bozulur. Performans düşer ama sistem hiçbir zaman durma noktasına gelmez. YSA sistemlerinin hata toleranslı olmasının nedeni bilginin tek bir yerde saklanmayıp, sisteme dağıtılmasıdır. Bu özellik sistemin durmasının önemli bir zarara neden olacağı uygulamalarda önem kazanır.

2.12. Öğrenme Kuralları

2.12.1. Perceptron

Perceptron ağı, ilk defa 1943 yılında Mc Culloch ve Pitts tarafından ortaya atılmıştır.

Şekil 2.12a'da tek katmanlı perceptronun yapısı görülmektedir.



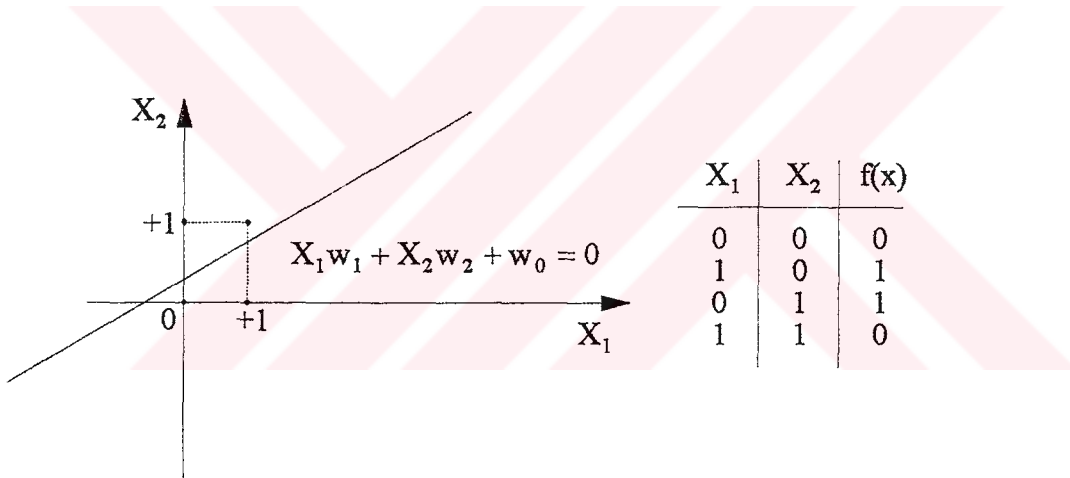
Şekil 2.12 (a) Tek katmanlı perceptron yapısı, (b) Bir analog giriş vektörünü A ve B ile gösterilen iki sınıfa ayıran bir tek katmanlı perceptron

$\sum_{j=1}^n w_j x_j - \theta = 0$; n boyutlu uzayda, n-1 boyutlu bir düzlem belirler.

Bu basit ağ, ilk defa geliştirildiğinde basit desenleri öğrenme kabiliyetlerinden ötürü oldukça ilgi toplamıştır. Şekil 2.12b'de bir girişin iki sınıftan (A veya B) hangisine ait olduğuna karar veren bir perceptron görülmektedir. Bir tek düğüm giriş elemanlarının ağırlıklı toplamını hesaplar, bir eşik θ 'yı çıkarır ve y' çıkışı +1 veya -1 olacak şekilde

sonucu bir süreksizlikten geçirir. Karar verme kuralı ise; çıkış +1 ise A sınıfı ile, -1 ise B sınıfı ile karşılık vermektir.

Yani bu ilk perceptron modeli, giriş bilgisinin mevcut iki sınıftan hangisine eşit olabileceğini bulacak şekilde eğitilen basit bir ağıdır. Daha sonra 1960 yıllarında F.Rosenblatt yukarıdaki ağ tipini biraz daha geliştirmiştir²². Fakat, Minsky ve Papert, bu tek katmanlı perceptronun XOR/"ayrıcılık" veya" işlemini gerçekleştiremediğini ispatlamışlardır. Şekil 2.13 ve 2.14'ten anlaşılacağı gibi 0'ların bir tarafta, 1'lerin bir tarafta ayrılacağı şekilde bir bölge oluşmamaktadır. XOR gibi 3 veya daha fazla sınıfa ihtiyaç duyulan problemleri çözmek için yapılması gereken işlem ise YSA ya yeni katmanlar eklemektir. Eşik bağlarıyla oluşturulan karar bölgesi şeklinin karmaşıklığı, sadece eklenmiş olan katmanların sayısı ile sınırlıdır. Şekil 2.14'te gösterildiği gibi içbükey ayrılabilir fonksiyonlar gerçekleştirilebilir²³.



Şekil 2.13 Lineer ayrılabilirliğin gösterimi

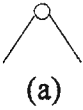
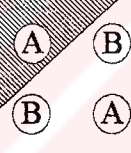
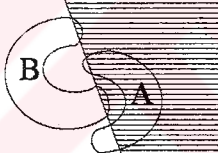
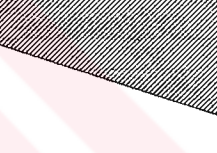
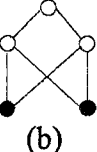
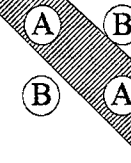
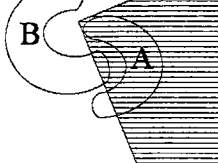
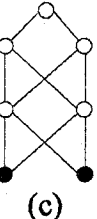
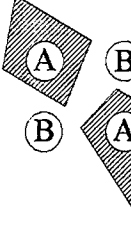
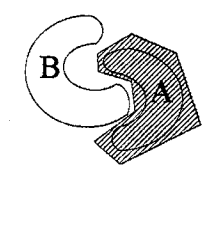
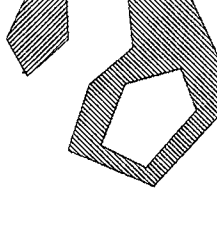
Şekil 2.14'te tek ve çok katmanlı perceptronların sınırlayabilecekleri/ayırabilecekleri karar bölgeleri görülmektedir. Taranmış bölgeler A sınıfına ait karar bölgelerini göstermektedir. Bir tek katmanlı perceptron bir düzlemi ikiye ayıran karar bölgeleri

²² Hebb, D.O. 1949.

²³ Pao, Y.H. 1989. *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing Co., Reading, Mass.

oluşturur. İki katmanlı bir perceptron, açık ve kapalı konveks bölgeler oluşturabilir. Üç katmanlı perceptron ise, keyfi derecelerde kompleks karar bölgelerini ayırabilir.

Buradan anlaşılmaktadır ki; perceptron gibi ileri yayımlı ağlarda, iki adet gizli katmandan fazlasına gerek yoktur. Üç katmanlı perceptronlarda (giriş katmanı dahil değil) kullanılacak düğüm sayılarının belirlenmesine gelince; karar bölgeleri arasında bağlantı yoksa veya bu bölgeler bir konveks alan ile oluşturulamıyorlarsa, ikinci katmandaki düğüm sayısı bir'den büyük olmalıdır. En kompleks durum için, ikinci katmandaki düğümlerin sayısı, giriş dağılımlarındaki ayrı bölgelerin sayısı kadar olmalıdır. Benzer şekilde birinci katmandaki düğüm sayıları da her bir ikinci katman düğümünce oluşturulan her bir konveks bölge için üç veya daha fazla kenar sağlanacak şekilde yeterli sayıda olmalıdır. Bu yüzden, birinci katmanda, ikinci katmandaki düğüm sayısının üç katından fazla düğüm bulunmalıdır. (Lippmann 1987)

Yapı	Karar bölgeleri tipi	XOR problemi	Bölgelere dayalı sınıflar	En iyi ayırdığı bölge şekilleri
Tek katman  (a)	Doğrusal ayrıştırabilir. XOR'ün Ayrıcılığı veya işlevini gerçekleyemez.			
İki katman  (b)	Konveks/içbükey açık veya kapalı bölgeler.			
Üç katman  (c)	Rastgele bölgeler (Komplekslik derecesi düğüm sayısına bağlı olarak sınırlıdır.)			

Şekil 2.14 Tek katmanlı perceptron ile iki ve üç katmanlı bulunan çok katmanlı perceptronların sınırlayabileceği karar bölgeleri (Lippmann 1987)

Bununla beraber, Gutierrez ve arkadaşları, değişik perceptron ağlarının ihtiyacı olan düğüm sayıları hakkında araştırma yapmışlar ve çok fazla düğümün de, çok az sayıda olduğu gibi zararlı etkisi olduğunu bulmuşlardır²⁴. Tek katmanlı perceptron uygulanan her eğitim seti modelinin en önemli özelliği, lineer biçimde dağılmak zorunda olmasıdır. Eğer bu doğruysa; Rosenblatt, perceptron ağının kararlı olacağını göstermiştir. (Örneğin ağırlıklar iterasyonla yakınsar).

2.12.2. Çok katmanlı perceptron (Multi-Layer Perceptron)

Çok katmanlı perceptron giriş ve çıkış katmanları arasında birden fazla katmanın kullanıldığı ileri yayımlı (İng. feed-forward) YSA sistemidir. Gizli katman (İng. hidden layer) olarak isimlendirilen bu ara katmanlarda, düğümleri giriş ve çıkış katmanlarına doğrudan/aracısız bağlı olmayan işlem elemanları/düğümmler vardır. Şekil 2.15'te çok katmanlı perceptronun genel yapısı verilmiştir.

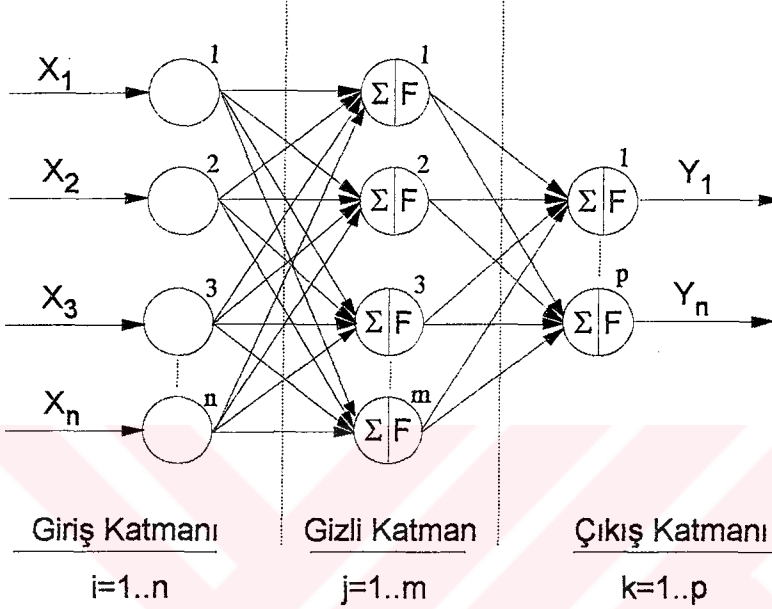
Çok katmanlı perceptronlar, tek katmanlı perceptronlara ait sınırlamaların birçoğunu ortadan kaldırmalarına rağmen, önceleri efektif eğitime algoritmalarının mevcut olmamasından ötürü uygulamaları yaygın değildi. Ancak yeni eğitime algoritmalarının geliştirilmesi ile bu durum değişmiştir²⁵.

Çok katmanlı ağlarda, veriler giriş katmanı tarafından kabul edilirler. Ağ içinde yapılan işlemler sonucunda çıkış katmanında oluşan sonuç değer, istenen cevap ile karşılaştırılır. Bulunan cevap ile istenen cevap arasında herhangi bir ayrılık varsa, ağırlıklar bu farkı azaltacak şekilde yeniden düzenlenir. Girişteki değer, ağırlıklar uygun noktaya ulaşana kadar değişmez. Hesaplanan çıkışlar, istenilen cevaplarla karşılaştırılarak sonuçta gerekirse hata belirtilir. Hata işareti gizli işlem elemanlarından/birimlerden çıkış birimine olan ağırlıkları değiştirmekte kullanılır. Ama

²⁴ Gutierrez, W. J., Grandin, R.D. 1989. Estimating hidden units for two layer perceptrons. *Proc. of the 1st Int. Conf. on Artificial Neural Networks*, London, U.K., October, 120-124.

²⁵ Rumelhart, D.E., Hinton, G.E., Williams, R.J. 1986. "Learning Internal Representations by Error Propagation", *Rumelhart Parallel Distributed Processing: Explorations in the Microstructure of Cognition, v.1: Foundations* (Edited by Rumelhart, D.E. & McClelland J.L.), MIT Press, Cambridge, MA.

bunu yaparken giriş katmanından gizli katmana gelenin değiştirilip değiştirilemediğini düşünmek gerekir. Gizli birimlerden ne tür bir çıkış istendiği bilinmeyeceği için gizli birimlerin çıkışında hata işareti verilmesi kolay bir şey değildir. Bunun yerine her birimin çıkış biriminin hatalarına olan etkisi bilinmelidir. Bu ise hatalı birim için gizli birime bağlı olan çıkış birimlerinin hata işaretlerinin toplamı alınarak yapılır.



F : Nöron çıkış değerini veren Transfer/Eşik Fonksiyonu
(Çıkış değeri komşu olan ilerdeki katmanın giriş değeri olacaktır.)
 Σ : Nöron NET değerini elde etmek üzere tüm (Giriş x Ağırlık) değerlerinin toplan

$\bigcirc$: Sadece veri girişinin sağlandığı işlem elemanı (yapay nöron)

$\bigcirc \Sigma F$: Σ & F hesaplamalarının gerçekleştirildiği işlem elemanı (yapay nöron)

Şekil 2.15 (n:m:p) mimarisine sahip bir çok katmanlı perceptronun yapısı

Çok sayıda gizli katmana sahip sistemlerde, her sistemin hata işaretleri, bir önceki katmanın düzeltilmiş işlemlerinden çıkartılarak işlem tekrarlanır. Sonuç olarak, ağırlık düzeltme işlemi çıkış seviyesine bağlı ağırlıklardan başlar ve işlem ters yönde, giriş seviyesine varana kadar devam eder. Sonuçta sistem hatalar yapar, ama bu hatalardan

birşeyler öğrenip isteneni bulana kadar işleme devam eder. Bu yöntem “hatanın geriye yayılması algoritması” (İng. back-propagation algorithm) denilir²⁶.

2.12.3. Hatanın geriye yayılması algoritması ve genelleştirilmiş delta kuralı

Hatanın geriye yayılması eğitime algoritması, çok katmanlı, ileri yayımlı bir perceptrondan elde edilen çıkışlar ile eldeki hedef çıkışlar arasındaki hataların karesinin ortalamasını minimum yapmak için geliştirilmiş iteratif bir gradyan algoritmadır ve eğitime işlemi için genelleştirilmiş delta kuralını (İng. Generalized Delta Rule) kullanır. Şekil 2.16’da mimarisi gösterilen algoritma, ana hatlarıyla şöyledir:

Ağ mimarisi tanımlamasını müteakiben ve ağırlıkların bazı rastgele küçük sayılar ile başlatılmasıyla, ağa ilk giriş sunulur. m-boyutlu giriş desenleri girildiğinde $x_i = [x_1, x_2, \dots, x_m]^T$ ‘dir. Benzer şekilde istenilen n-boyutlu çıkış desenlerini $d_k = [d_1, d_2, \dots, d_n]^T$ belirtir.

x_i değerleri i katmanındaki nöronların çıkış değerleri ise, j katmanındaki bir nörona gelecek olan toplam giriş,

$$\text{net}_j = \sum_{i=1}^m w_{ji} \cdot x_i \quad (\text{i. düğümden j. düğüme}) \quad (2.10)$$

olur. Gizli katmandaki j nöronunun /işlem elemanının çıkışı (transfer fonksiyonu çıkışı);

$$y_j = f_j(\text{net}_j), \quad j=1,2,\dots,J \quad (2.11)$$

dir ve burada f_j transfer fonksiyonudur.

Çıkış katmanındaki k nöronuna gelecek olan toplam giriş ise,

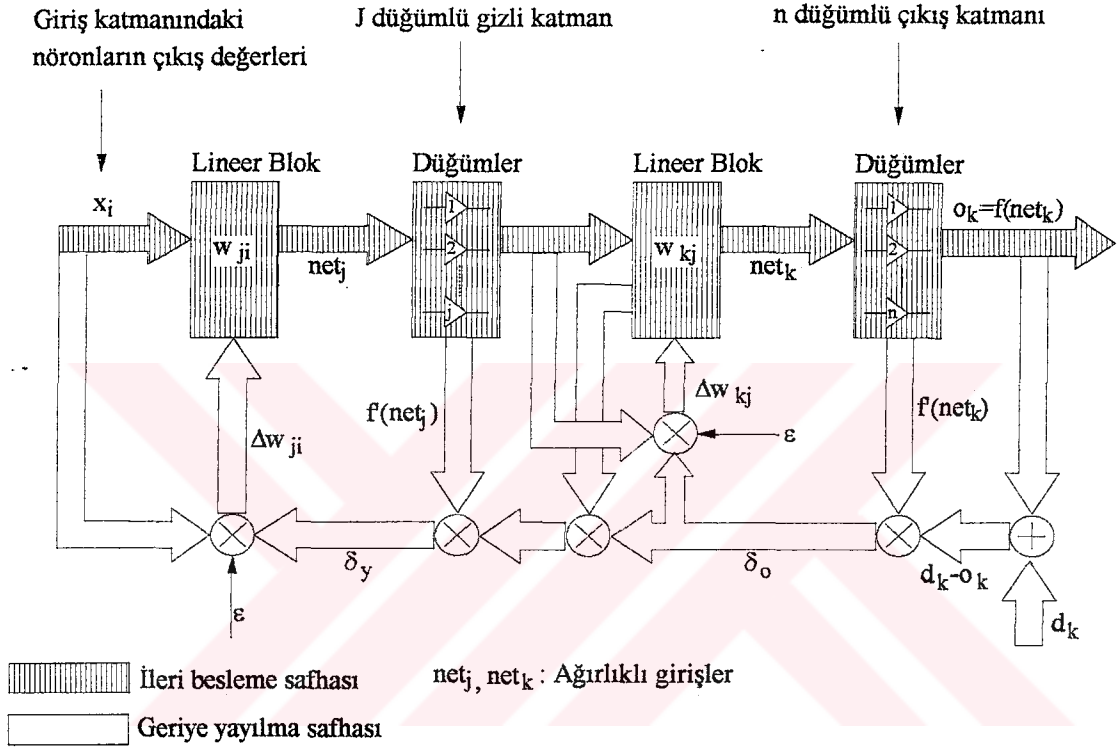
²⁶ Rumelhart, D.E., McClelland, J.L., PDP Research Group 1986.

$$\text{net}_k = \sum_{j=1}^J w_{kj} \cdot y_j \quad (2.12)$$

dir. Yine çıkış katmanındaki bir k nöronun lineer olmayan çıkışı;

$$o_k = f_k(\text{net}_k) \quad , \quad k=1,2,\dots,n \quad (2.13)$$

olur.



Şekil 2.16 Hatanın geriye yayılması algoritmasının blok diyagramı

Ağdan elde edilen çıkış, asıl çıkış ile karşılaştırılır ve e_k hatası aşağıdaki gibi hesaplanır;

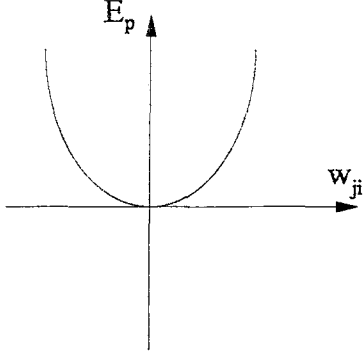
$$e_k = (d_k - o_k) \quad (2.14)$$

Burada d_k ve o_k sırasıyla çıkış katmanındaki herhangi bir k nöronunun hedef/istenen ve ağdan elde edilen/asıl çıkışlarıdır. Dikkat edilmesi gereken husus, böyle bir karşılaştırmanın sadece ağın çıkış katmanı için mümkün olmasıdır. Böylece çıkış katmanıyla olan bağlar için ağırlık ayarlaması öncelikle gözönüne alınır.

Herbir desen (İng. pattern) için toplam karesel hata;

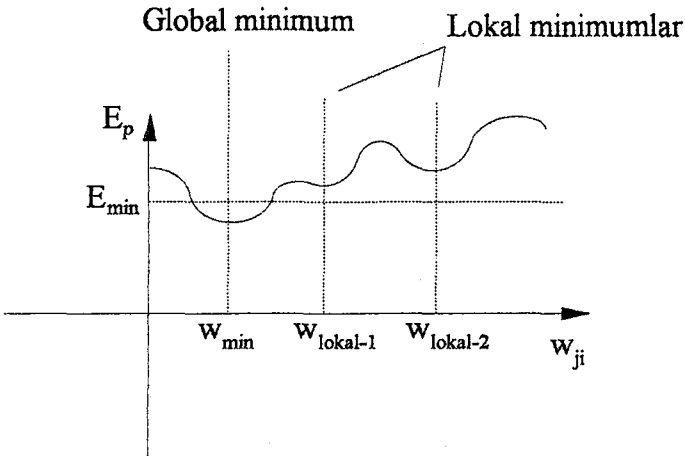
$$E = \frac{1}{2} \sum_k (d_k - o_k)^2 \quad (2.15)$$

olur. Hata fonksiyonu Şekil 2.17 ve 18’de gösterildiği gibidir.



Şekil 2.17 Gizli katmanı olmayan ağına hata fonksiyonu

Gizli katman olduğu zaman; hata düzeyi Şekil 2.17’de olduğu gibi sadece bir minimumdan oluşmaz. Şekil 2.18’deki gibi çeşitli minimumlar oluşur. Bu minimumlar lokal ve global olarak sınıflandırılırlar. Lokal minimumlar birden fazla olabilir. Global minimum, hatanın en küçük olduğu minimumdur. Öğrenme sırasında global minimuma ulaşmak amaçlanır.



Şekil 2.18 Gizli katmana ait ağına hata fonksiyonu

Ağırlıkların değişimi;

$$\Delta w_{kj} = -\varepsilon \frac{\partial E}{\partial w_{kj}} \quad (2.16)$$

Burada ε , öğrenme oranı katsayısı adı verilen küçük değerde pozitif bir sayıdır (çoğunlukla 0.01 ile 0.9 arasında seçilir). Bu öğrenme oranı, bir sayısal optimizasyon algoritmasındaki adım boyutu parametresine benzemektedir. (2.16)'da eşitliğin sağ tarafı açılırsa (Rumelhart et al. 1986), şu sonuç elde edilir:

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} \quad (2.17)$$

$$\frac{\partial net_k}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}} \sum_j w_{kj} \cdot y_j = \sum_j \frac{\partial w_{kj} \cdot y_j}{\partial w_{kj}} = y_j \quad (2.18)$$

Amaç, uygun w seçimiyle, $E = \sum_p E_p$ toplam hatayı yeterince küçük yapmaktır. Bu amacı gerçekleştirmek için, bir $p \in P$ deseni ard arda ve rastgele biçimde seçilir. k .nıncı nöronda oluşan ve “delta” adı verilen hata işareti,

$$\delta_o = -\frac{\partial E}{\partial net_k} \quad (2.19)$$

(2.18) ve (2.19)'u (2.17)'de yerine konulursa,

$$\frac{\partial E}{\partial w_{kj}} = -\delta_o \cdot y_j \quad (2.20)$$

ve, (2.20) ifadesi (2.16)'da yerine konulursa,

$$\Delta w_{kj} = \varepsilon \delta_o y_j \quad (2.21)$$

elde edilir. E_p değerini düşürmek demek, ağırlığı $\delta_o y_j$ 'ye bağlı olarak değiştirmek demektir. Buna “Delta kuralı” denir. Önceki gizli katmandaki tüm nöronlar ile çıkış katmanındaki k .nıncı nöron arasındaki bağ mukavemetleri, (2.21)'deki Δw_{kj} miktarı ile ayarlanır. (2.19) denklemi kısmi türevlerine ayrılırsa,

$$\delta_o = -\frac{\partial E}{\partial \text{net}_k} = -\frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial \text{net}_k} \quad (2.22)$$

$$\frac{\partial E}{\partial o_k} = -(d_k - o_k) \quad (2.23)$$

olup, k.nıncı nöron çıkışının lokal hatasını verir.

$$\frac{\partial o_k}{\partial \text{net}_k} = f'_k(\text{net}_k) \quad (2.24)$$

Son iki formül (2.22)'de yerine konulursa,

$$\delta_o = (d_k - o_k) f'_k(\text{net}_k) \quad (2.25)$$

bulunur. Bu son terim (2.21)'de yerine konulduğunda k nöronu için;

$$\Delta w_{kj} = \varepsilon (d_k - o_k) f'_k(\text{net}_k) y_j \quad (2.26)$$

olur. Eğer ağırlıklar çıkış nöronlarını direkt olarak etkilemiyorsa/arada gizli katman varsa, (2.21)'e benzer biçimde delta kuralı bu gizli katman için de şu şekilde uygulanır.

$$\begin{aligned} \Delta w_{ji} &= -\varepsilon \frac{\partial E}{\partial w_{ji}} \\ &= -\varepsilon \frac{\partial E}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial w_{ji}} = -\varepsilon \frac{\partial E}{\partial \text{net}_j} x_i \\ &= -\varepsilon \left(\frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial \text{net}_j} \right) = -\varepsilon \left(\frac{\partial E}{\partial y_j} \right) f'_j(\text{net}_j) x_i \end{aligned} \quad (2.27)$$

$$\Delta w_{ji} = \varepsilon \delta_y x_i \quad (2.28)$$

Bununla birlikte, $\partial E / \partial y_j$ faktörü direkt olarak geliştirilemez. Özel olarak çıkış katmanına etkisi ise,

$$-\frac{\partial E}{\partial y_j} = -\sum_k \frac{\partial E}{\partial \text{net}_k} \frac{\partial \text{net}_k}{\partial y_j} = \sum_k \left(-\frac{\partial E}{\partial \text{net}_k} \right) \frac{\partial}{\partial y_j} \sum_m w_{km} y_j = \sum_k \left(-\frac{\partial E}{\partial \text{net}_k} \right) w_{kj} = \sum_k \delta_o w_{kj}$$

şeklinde bulunur. Bu durumda;

$$\delta_y = f'_j(\text{net}_j) \sum_k \delta_o w_{kj} \quad (2.29)$$

olur. Kısaca özetlenecek olursa; (2.28) formülü şayet j. nöron, çıkış katmanı nöronuysa (2.25)'e benzer biçimde,

$$\delta_y = (d_y - o_y) f'_j(\text{net}_j) \quad (2.30)$$

olur. Eğer j nöronları gizli katmana ait nöronlar ise, o zaman (2.20) denklemi kullanılır. Transfer/eşik fonksiyonu olarak sigmoid fonksiyonu kullanıldığında,

$$f(\text{net}_j) = y_j = \frac{1}{1 + e^{-\text{net}_j}} \quad (2.31)$$

olur. Bu ifadenin türevi alınırsa,

$$f'(\text{net}_j) = \frac{1}{1 + e^{-\text{net}_j}} \frac{1 + e^{-\text{net}_j} - 1}{1 + e^{-\text{net}_j}} \quad (2.32)$$

ve gerekli sadeleştirme yapıldığında,

$$\frac{\partial y_j}{\partial \text{net}_j} = y_j(1 - y_j) \quad (2.33)$$

bulunur. Benzer işlem k nöronu için de yapılırsa;

$$\frac{\partial o_k}{\partial \text{net}_k} = f'_k(\text{net}_k) = o_k(1 - o_k) \quad (2.34)$$

elde edilir. (2.34) denklemi (2.25)'de, (2.33) denklemi de (2.29)'da yerine konursa, (2.25) ve (2.29)'daki delta ifadeleri aşağıdaki gibi olur. (δ_o çıkış katmanı, δ_y gizli katman elemanları içindir);

$$\delta_o = (d_k - o_k) o_k (1 - o_k) \quad (2.35)$$

$$\delta_y = y_j (1 - y_j) \sum_k \delta_o w_{kj} \quad (2.36)$$

w_{ji} yi öğrenme durumunda her eğitime deseni seti için Δw_{ij} yi hesaplamak gerekir. Öğrenme oranı ϵ , hızlı öğrenmeyi sağlar; fakat dalgalanmalara sebep olabilir. Rumelhart ve Mc Clelland (1986) (2.26) ve (2.28) ifadelerine bir tür momentum terimi dahil ederek bir değişim öngörmüşlerdir. Bu durumda;

$$\Delta w_{ji}(n+1) = \Delta w_{ji}(n) + \epsilon \delta_y x_i + \alpha [\Delta w_{ji}(n) - \Delta w_{ji}(n-1)] \quad (2.37)$$

$$\Delta w_{kj}(n+1) = \Delta w_{kj}(n) + \epsilon \delta_o y_j + \alpha [\Delta w_{kj}(n) - \Delta w_{kj}(n-1)] \quad (2.38)$$

olarak yazılabilir. Burada n , öğrenme döngülerinin/iterasyonlarının sayısını gösterir. Momentum terimi olan α , küçük değerde pozitif bir sayıdır ve genellikle 0.9 alınır.

2.12.4. Öğrenme ve momentum katsayıları

YSA ile ilgili bir başka sorun da, düzgün bir öğrenme katsayısının (ϵ) ayarlanmasıdır. Ağırlıkları çok yüksek tutmak davranışın bozulmasına sebep olabilir. Bu nedenle, öğrenme katsayısını böyle bir davranışı önlemek için küçük tutmak gereklidir. Öğrenme katsayısı, $0.01 < \epsilon < 10$ aralığında seçilen sabit bir sayıdır. Öte yandan, çok küçük bir öğrenme oranında, öğrenme işleminin yavaşlamasına yolaçar. Öğrenme katsayısı ϵ 'u başlangıçta büyük bir değerle başlatmak ve öğrenme sırasında azaltmak lokal minimum problemini ortadan kaldırmak için oldukça faydalıdır. Bu tez çalışmasında, ϵ 'nin başlangıç değeri için yaygın olarak kullanılan 0.7 değeri seçilmiş ve program içinde azaltılarak öğrenme hızlandırılmıştır.

Momentum (α) fikri bu noktadan hareketle ortaya atılmıştır. Momentum mevcut delta ağırlığı üzerinden önceki delta ağırlığının belirli bir kısmını besler. Böylece daha düşük öğrenme katsayısı ile daha hızlı öğrenme elde edilir. Momentum katsayısı genellikle $0 < \alpha < 1$ aralığında değişen bir sayıdır ve genellikle 0.9 alınır. Öğrenme sürecine bir bellek katmaya çalışan α teriminin dahil edilmesi, programın/yürütmenin

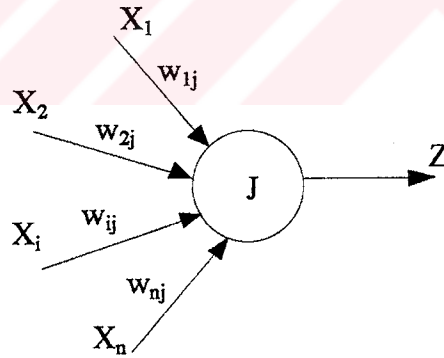
kararlılığını/stabilitesini artırır ve ayrıca bir lokal optimuma yaklaşmayı engellemede bir dereceye kadar yardım eder.

Hatanın geriye yayılması algoritması genellikle iyi bir performans göstermesine rağmen; bu algoritma, bir gradyent arama tekniği olduğu için, global minimum yerine, en küçük kareler (İng. Least Mean Square) fonksiyonunun bir lokal minimumunu bulabilmektedir. Performansı artırmak ve lokal minimum problemini azaltmak için;

- Gizli düğümler ilave edilmesi,
- Ağırlıkların ayarlanması için kullanılan öğrenme katsayısının azaltılması,
- Farklı rastgele ağırlık setlerinden başlayarak, birçok defa eğitme işleminin tekrarlanması, öneriler arasındadır (Lippmann 1987).

2.13. Sinirsel Hesaplama

Ağın davranışı hakkında daha iyi bir fikir edinmek için, bir işlem elemanı incelenecektir.



Şekil 2.19 Bir tekil işlem elemanı

Şekil 2.19’da gösterildiği gibi, işlem elemanı x_i , $i=1,2,...,n$ olan bir veri giriş grubu alır. Bu girişler, bir biyolojik modelde bir nöron tarafından alınan elektro-kimyasal sinyallere benzerdir. Bu giriş sinyalleri bağ ağırlıkları w_{ij} ile çarpılırlar ve elemana gelen etkin(efektif) giriş aşağıdaki gibi girişlerin ağırlıklı toplamıdır:

$$Z = \sum_{i=1}^n w_{ij} \cdot x_i \quad (2.39)$$

Bir biyolojik sistemde gelen sinyalin gücü ancak belirli bir düzeye eriştiğinde nöron bir çıkış sinyali üretir. Bu, sinir ağında her bir sinirsel eleman için bir eşik seviyesi vererek simüle/taklit edilebilir. Ancak, bir çıkış sinyali elde etmek için girişlerin ağırlıklı toplamını bir aktivasyon (harekete geçirme) fonksiyonu F ile işlemek en yaygın olanıdır. Yani;

$$Y = F(Z) \quad (2.40)$$

Değişik formlarda aktivasyon fonksiyonları önerilmiştir. En yaygın olarak kullanılan aktivasyon fonksiyonlarından olan ve aşağıdaki ifade ile verilen,

$$F(Z) = \frac{1}{1 + e^{-(Z+T)}} \quad (2.41)$$

sigmoid fonksiyonu, bu çalışmada kullanılmıştır. Burada, Z düğüme gelen ağırlıklı giriş, T ise, eleman çıkışını ayarlamak için kullanılan bir aktif yapma işareti parametresidir. Bu fonksiyonun başlıca avantajı, hem küçük, hem geniş/büyük sinyalleri kullanabilme yeteneğidir. Bu fonksiyonun eğimi, elde edilen kazancın bir göstergesidir. Giriş sinyalinin büyük pozitif ve negatif değerleri için kazanç yok denecek kadar küçüktür; giriş sinyalinin orta değerlerinde ise kazanç sonsuzdur. Böylece giriş sinyallerinin büyük bir kısmı için uygun bir kazanç seviyesi elde edilir. Aktivasyon fonksiyonundan elde edilen çıkış, ağdaki diğer nöronlar için bir giriş olarak ele alınabilir. Uygun ağırlık katsayılarının ve aktif yapma işareti parametrelerinin belirlenmesi, kendi başına bir hata minimizasyon problemi olan ağ öğrenme işleminde gerçekleşir.

3. YAPI MÜHENDİSLİĞİNDE YSA KULLANIMI

Mühendislik uygulamaları için YSA konusundaki araştırmalar henüz çok yenidir. YSA son yıllarda inşaat mühendisliği alanında birçok problemlere çözüm bulmak amacıyla kullanılmaya çalışılmıştır. Karayolu, malzeme, yapı analizi, şantiye mühendisliği ve yapı işletmesi dallarında çeşitli uygulama örnekleri mevcuttur.

Bu tez çalışmasının asıl amacı da, yapı mühendisliği alanında uygulamaları bulunan YSA teknolojisine bir giriş sunmaktır. YSA, genellikle problemin türüne göre belirli oranlarda hata ile sonuca ulaşabilmektedir. Ancak, bu hata düzeyi, taşıyıcı sistem boyutlandırması problemlerinin birçoğu için önemsiz mertebede kalmaktadır. Kullanılan malzemenin beton gibi heterojen ve daneli, non-lineer bir malzeme olduğu gerçeği ve projenin uygulanması sırasında yapılan hatalar gözönüne alınırsa; projenin çözümünde yapılan %5-10 mertebesindeki hataların birçok problemde önemli olmadığı ortaya çıkar. Kaldıkı betonarme elemanlarda meydana gelen yeniden uyum, bu mertebedeki hataları bertaraf ederek, yapı elemanının fonksiyonunu yerine getirmesine imkan sağlamaktadır (Arslan ve ark. 1995).

İnşaat mühendisliği problemlerinin bir diğer özelliği de, çok gerçekli karar verme durumudur. Yani, problemin çözümü için birden fazla kabul edilebilir metodun bulunmasıdır. Bu tür durumlarda, kesin veri isteyip ona göre kesin sonuç isteyen problemlerin çözümü zorlaşmaktadır. Ancak, YSA kullanarak bu tür problemlere daha gerçekçi yaklaşmak mümkündür.

3.1. YSA ve Uzman Sistemler

Geçen birkaç yıl içinde, hesaplama dayalı inşaat mühendisliği hakkındaki literatür, özellikle "Uzman Sistem" teknolojisi ile ilgili yapay zeka uygulamaları üzerinde odaklanmıştır. Bilgisayar biliminin bir konusu olan yapay zeka konusundaki araştırmalar, yapı endüstrisi için oldukça uygun çözümler sağlamaktadır. Yapay zekanın bir dalı olan uzman sistemler, endüstride çeşitli uygulamalarda kullanıla

gelmektedir. Ancak, son on yıldır bu sistemlerin performansı hedeflenen ideallerden uzak kalmıştır. Yapay zekanın YSA araştırmaları, klasik uzman sistemlerin yardımcısı veya uzantısı olarak çalışan daha güçlü sistemler oluşturmuşlardır. Uzman sistem benzerlerinin aksine, sınır ağları gözlemlerle elde edilmiş bilgiye dayanarak eğitilebilmektedir.

Uzman sistemler, robotikler ve YSA, mevcut yapay zeka çalışmalarının yapı endüstrisi ile ilgi araştırma sahaları arasındadırlar. Bir taraftan, uzman sistemler insan beyninin zekaya dayanan sebepleme ve problem çözme kabiliyetlerini modellemeye çalışmakta; diğer taraftan YSA, beynin öğrenme, düşünme, bilgi depolama, çağrıştırmaya dayalı tanıma ve bilgi elde edilmesini modellemeye çalışmaktadır. 1950'li yılların başından itibaren birlikte başlamalarına rağmen, başlangıçtaki şüpheli yaklaşımdan dolayı YSA'daki araştırmalar, uzman sistemlere göre çok daha yavaş ilerlemiştir. 1980'lerin ortalarına doğru ise, YSA araştırması gelişmeye başlamış ve çeşitli sahalarda çok sayıda uzman sistem ortaya çıkmıştır (Şekil 3.1). Yapı mühendisliği ve yönetimi birçok uzman sistem uygulamaları için verimli bir alan sayılmıştır²⁷. Bunun sebebi, uzman bilgi, yargı ve tecrübe, birçok yapı mühendisliği ve yönetimi işlemlerinin yerine getirilmesi sırasında ve yapı problemlerinin çözümü için anahtar konumundaki gereksinimler olmalarından kaynaklanmaktadır²⁸. Yapay zeka tabanlı sistemlerin aksine, klasik karar verme ve analiz teknikleri (ihtimal metodları, bulanık kümeler teorisi vb.) için genellikle ileri matematik gereklidir; bu yüzden bu tür teknikler yapı personeli için pek kullanışlı olmamaktadırlar²⁹.

Uzman sistemlerin çok azının, esasen iddia edildiği gibi, zekice çalışmakta olduğunu ortaya koymak için, yapı sahasında son zamanlarda birçok uzman sistem değerlendirilmiştir³⁰ ve uzman sistemlerin gelişmesinin önünde ana engel olarak,

²⁷ Mohan, S. 1990. Expert systems applications in construction management and engineering. *Journal of Construction Engineering and Management*, ASCE 1:87-89.

²⁸ Lewitt, R.E. 1987. *Expert Systems In Construction: State Of The Art, Expert Systems For Civil Engineers: Technology And Applications*, M.L. Maher, ed., ASCE, New York, N.Y., 85-112.

²⁹ Ahmad, I., Minkarah, I. 1990. Decision analysis and expert system technology: a construction industry perspective. *Proc.Int.Symp. on Building Economics and Construction Management*, CIB, Sydney, Australia, 363-372.

³⁰ Adeli, H. 1988. *Expert Systems in Construction and Structural Engineering*, Chapman and Hall, Ltd., London, England.

bilginin elde edilmesi gösterilmiştir. Brandon ve Fung Fai, klasik uzman sistem mimarilerinde ana problemlerin bilginin yapılandırılması ve kullanım sahasının sınırlandırılması olduğunu kaydetmişlerdir³¹. Uzman sistemlerin zekalarına dair basmakalıp beklentilere rağmen, bu sistemlerin sebepleme yeteneklerinin zekanın elde edilmesinde gerekli olan sadece bir bölümden ibaret olduğu kaydedilmiştir³². Uzman sistemler, kendi kendilerine öğrenme, çözümleri genelleme gibi konularda ve daha önceden karşılaşılmamış, eksik, gürültülü, aşırı ilişkili verilerle işlemlerde yetersiz kalmaktadırlar³³. Klasik uzman sistemlerin seri(ardışık) mimarileri uygulanabilirliklerine katı kısıtlamalar getirmektedir. Çünkü; yeni nesil paralel bilgi işleme mimarilerine karşı olmaları, karmaşık uzman sistem uygulamalarını oldukça yavaşlatır ve işletme maliyetini artırır. Uzman sistem uygulamaları, geniş yerleşik hafızanın (RAM) yanında yoğun yazılım ve desteğe ihtiyaç göstermektedirler. Bu problemler bir bütün olarak klasik uzman sistemlerin kullanımını, sebepleme için yeterli miktarda süreyi sağlayacak olan durumları olaylara bağlayan yeterli miktarda bilgi yapısının mevcut olduğu alanlarla sınırlayacaktır³⁴. Böylece, sonuç çıkarmayla ilgili problemler ve çabuk cevapların gerektiği sonuç çıkarma ve analoji problemlerinin pek azı için uygundurlar.

Uzman sistemler, beynin problem çözme fonksiyonlarını taklit eden belirli prosedürlere dayalı program sistemleri iken; sinir ağları hem yapı hem de fonksiyonalite açısından biyolojik sinir sistemlerine benzer donanım veya yazılım sistemleri olarak tarif edilirler.

³¹ Brandon, P.S. 1990. Expert Systems- After The Hype Is Over. *Proc.Int.Symp. on Building Economics and Construction Management*, CIB, Sydney, Australia, 314-347.

FUNG FAI, N.G. 1990. Knowledge Base Systems: Their Impact on Construction Industry. *Proc.Int.Symp. on Building Economics and Construction Management*, CIB, Sydney, Australia, 549-560.

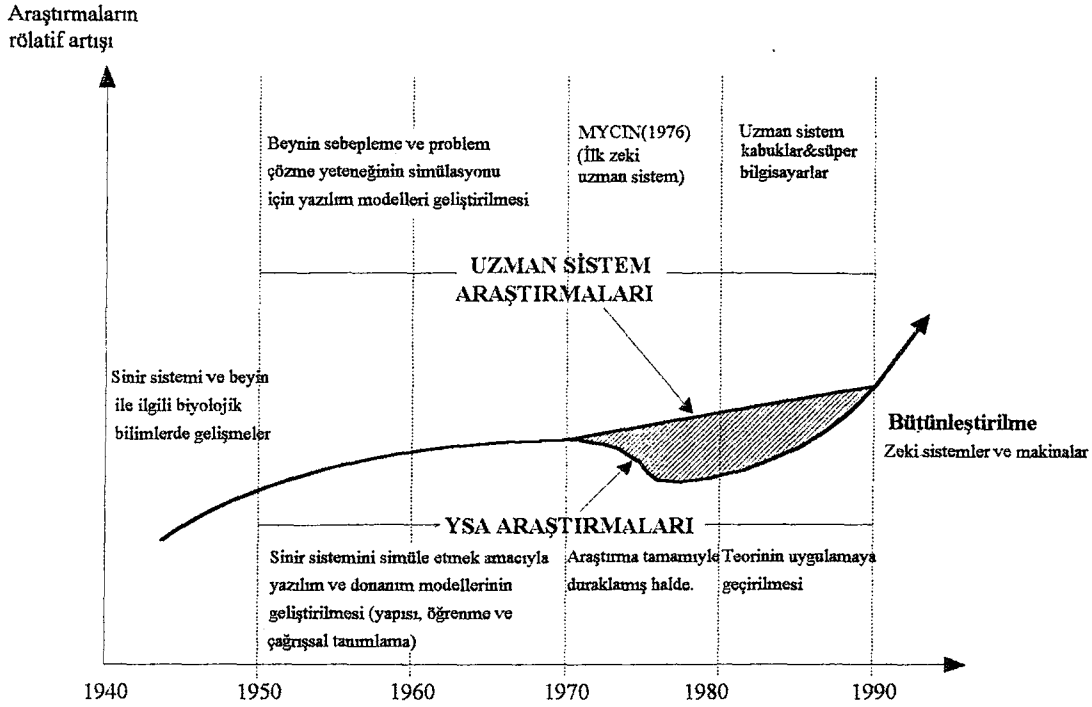
³² Gallant, S. 1998. Connectionist expert systems, *Commun. ACM*, n. 2:152-169.

Castelaz, P., Angus, J., Mahoney, J. 1987, Application of Neural Networks to Expert Systems and Command and Control Systems. *Proc. IEEE Western Conf. on Expert Systems*, 118-125.

³³ Wassermann, P.D 1989. *Neural Computing,: Theory and Practice*, Van Nostrand Reinhold, New York, N.Y.

Pao, Y.H. 1989.

³⁴ Jackson, P. 1986. *Introduction to Expert Systems*, Addison-Wesley, London, U.K.



Şekil 3.1 YSA ve uzman sistemlerin tarihsel gelişimi

YSA ları diğer yapay zeka uygulamalarından (uzman sistemler vb.) ayıran birçok karakteristikleri vardır³⁵. Bunlardan bazıları şöylece sıralanabilir:

- 1- Çok sayıda niteliğin paralel olarak gözönüne alınması gerektiği desen tanımlama işlemleri için oldukça uygundurlar.
- 2- Örnekler yoluyla öğrenirler. Uzman sistemlerin aksine, YSA lar bir çok örnek desen ve bunların hedeflenen sonuçları ile öğrenirler. Bu örnekler uzmanlardan neden ve nasıl bu sonuçlara vardıklarını sormaya gereksinim duymadan temin edilebilir. Bu yüzden bilgi elde edilmesine dair problemler ortadan kalkar.
- 3- Öğrenme için uzun bilgisayar sürelerine ihtiyaç duymalarına rağmen, hızlı sonuçlar üretirler. Bu, YSA nın paralel yapısından kaynaklanmaktadır.

³⁵ Wasserman, P.D. 1989; Pao, Y.H. 1989; Gallant, S., 1988; Castelez, P. 1987.

4- Öğreticisiz öğrenmede halinde, çok sayıda giriş örneğinden sınıflandırma (gruplandırma) karakteristiklerini çıkarabilirler.

5- Dağıtılmış bir hafızaya sahiptirler; işlem elemanlarını birbirine bağlayan bağ ağırlıkları YSA'nın hafıza birimleridir. Ağırlıkların değerleri, YSA'daki mevcut bilgi durumunu ifade eder. Örneğin bir giriş ve istenen çıkış çifti ile temsil edilen bir bilgi birimi, YSA'nın tüm ağırlıklı bağları boyunca dağıtılmıştır.

6- YSA, bozuk, eksik veya önceden görülmemiş verilere tamamlama veya ara değer bulma şeklinde sonuçlar üretebilir. Bir oto-ilişkili, yani girişin istenen çıkış değerine eşit olduğu YSA ya kısmi bir giriş verildiğinde tam bir çıkış üretebilir. Buna “genelleme” özelliği denir.

7- YSA'larda hata toleransı vardır. Hafıza dağınık bir şekilde bulunduğu için dolayı, bazı işlem elemanlarındaki hata, YSA'nın toplam davranışını az bir şekilde değiştirecektir. Diğer bir klasik hesaplama sisteminin herhangi bir küçük kesimindeki hata sistemin tamamen durmasına yolaçabilir. YSA'nın bu karakteristiği, özellikle emniyetli sistemlerin daha az emniyetli elemanlar ile oluşturulmasının gerektiği uygulamalar için oldukça faydalıdır.

8- Belirsizliği temsil edebilirler; ‘kanı’ ölçüsü, problem deseninin değiştirilmesi ile iki şekilde hesaba katılabilir: (1) Nitelikteki kanıyı temsil edebilecek giriş değerleri seçerek, (2) Kanıyı temsil edebilecek bir niteliği/nitelikleri giriş örneğine ilave ederek.

9- Daha düşük miktarda depolama hafızasına ihtiyaç duyarlar. Çünkü, çok büyük miktarda depolanmış veri desenlerini temsil edebilecek, sadece bir ağırlıkları seti mevcuttur.

3.2. YSA Uygulamalarının Geliştirilmesi

Bu tez çalışmasında, hata geri yayımlı (İng. back-propagation), ileri beslemeli (İng. feed-forward) YSA modeli kullanılacaktır. Bu tip YSA modelleri, problem çözme yeteneklerini insanın iş tecrübesi kazanmasına benzer bir biçimde, karşılaştığı örnek

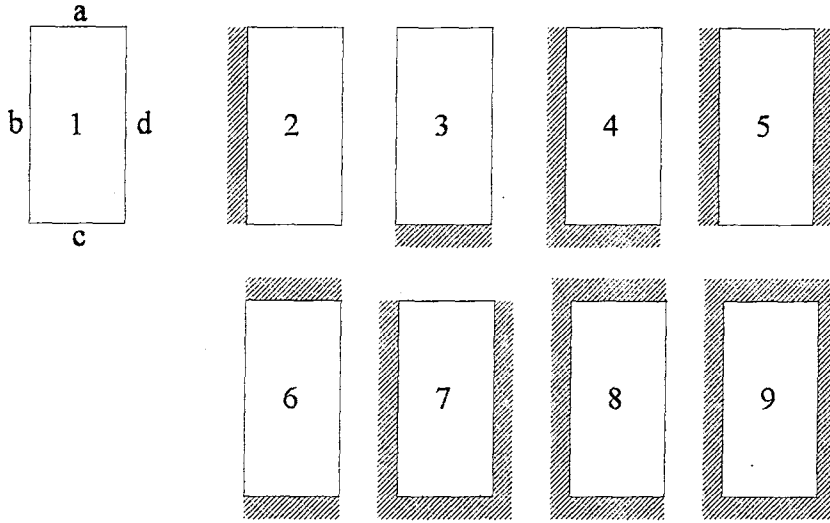
oluşumlardan alırlar. Herbir oluşum için, giriş parametrelerinin bir giriş verisi deseni, istenen çıkış parametrelerinin de bir çıkış deseni oluşturduğu bu örnek oluşumlara, “eğitme örnekleri” adı verilmektedir. Böylece, herbir eğitim örneği, ağı tecrübesini artırmaktadır. Eğitim sırasında, YSA eğitim örneklerindeki bilgiyi genelleyerek (İng. generalization) yeni problemlere çözümler üretme kabiliyetini kazanır.

Genel olarak, uygulamaların üç ana safhada geliştirilmesi önerilmektedir. Bunlar, tasarım, öğrenme ve test/kullanımdır. Bu üç safha aşağıda açıklanacaktır.

3.2.1. Tasarım

Tasarım; problemin veya geliştirilmekte olan uygulamanın, tamamiyle kavranmasının ve anlaşılmasının gerekli olduğu önemli bir safhadır. Bu arada, ele alınan problemin/uygulamanın YSA ile çözüme uygun olup olmadığı da başka bir önemli noktadır. Problemin çözümüne uygun bir YSA mimarisi seçilir. Birbirinden farklı yeteneklere sahip birçok sinir ağı mimarisi geliştirilmiştir. Çizelge 3.2 ve 3.3’te yaygın olarak kullanılan YSA mimarilerinin performans karakteristikleri, üstünlükleri ve sınırlamaları verilmiştir. YSA mimarisi belirlendikten sonra, problemin giriş ve çıkış katmanlarında yer alacak parametreler kesin olarak tanımlanmalıdır. Bu parametreler, kalitatif/nitelik bildiren veya kantitatif/miktar bildiren tiplerde olabilirler. Daha sonra, kullanılacak ağ mimarisinin gereklerine uygun olarak bu parametreler uygun desenlere/değerlere dönüştürülür. Bu işlem, verilerin ikili (İng. binary) veya sürekli (İng. continuous) değerlere dönüştürülmesiyle/normalizasyonuyla gerçekleştirilebilir. İkili değerlere dönüştürme sırasında, giriş ve çıkışlar 0 ve 1 lerden oluşan vektörlere dönüştürülürler. Öte yandan, sürekli değerlere dönüşümde ise, veriler reel sayılara dönüştürülür. Buna bir örnek olarak, plakların mesnetlenme durumlarının Şekil 3.3a’da ikili ve Şekil 3.3b’de sürekli değerli girişinin nasıl hazırlanabileceği gösterilmiştir. Görüldüyorki, ikili girişte mesnetlenme durumu 4 giriş düğümü ile, sürekli değerli girişte ise tek düğümle ifade edilebilmektedir. Her ne kadar ikili düzende verinin girişi daha çok düğüme dolayısı ile daha büyük bir matrise ve işlem süresine gerek duyarsa da, paralel dağılımlı algoritmaların yapısına uygun olan verinin paralel olarak daha geniş bir bölgeye dağıtılması olduğundan, bu çeşit bir veri girişi YSA’nın

Plakların çeşitli mesnetlenme durumları



Plak mesnetlenme durum numaraları (0..9)

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Veri giriş deseni :

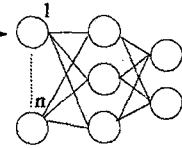
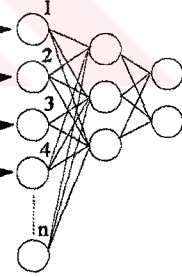
a								
b								
c								
d								

a) İkili düzendeki karşılığı :

a	0	0	0	0	0	1	0	1	1
b	0	1	0	1	1	0	1	1	1
c	0	0	1	1	0	1	1	1	1
d	0	0	0	0	1	0	1	0	1

b) Sürekli değer karşılığı :

(0.1) (0.2) (0.3) (0.4) (0.5) (0.6) (0.7) (0.8) (0.9)



Yapay Sinir Ağı

Şekil 3.2 Giriş veri desenlerinin YSA tarafından okunabilecek şekilde giriş vektörlerine dönüştürülmesi: (a) İkili, (b) Sürekli değer dönüşümü

performansını önemli derecede artıracaktır. Giriş verisine ait reel sayılar, ölçeklendirme, normalizasyon ve/veya fonksiyona tabi tutulma gibi önışlemlerden geçirilebilirler. Çıkış verisine ait reel sayılar ise genellikle 0 ~ 1.0 arasında değerler alacak şekilde ayarlanır.

Problemin büyüklüğüne bağılı olarak, yani giriş ve çıkış parametrelerinin sayısına bağılı olarak, bir tek YSA veya birçok alt YSA gruplarından oluşan bir süper/global YSA kullanılabilir. Heriki durumda da çıkış katmanındaki herbir eleman için, bir ağı kullanımı da düşünülebilir (Moselhi 1996). YSA yapısı hakkında bir karara vardıldıktan sonra, gizli katman sayısı ve herbir katmandaki düğümler/ışlem elemanı sayısı belirlenmelidir. YSA'nın en iyi performans gösterdiği, yani ağı hatasını minimum, öğrenme hızını maksimum yapan optimum veya optimuma yakın katman ve ışlem elemanı sayılarının belirlenebilmesi, ancak tatonmanla/deneme yanılmayla mümkündür. Böylece, artık YSA tasarlanmıştır ve eğitilmeye hazırdır.

3.2.2. Öğrenme

Öğrenme işlemine başlamadan önce; bilginin elde edilmesi, eğitme örneklerinin hazırlanması gibi işlemler önceden yapılmış olmalıdır. Öğrenme, veri giriş bölümünde veya isteğe bağılı olarak çıkış katmanında verilen belirli sayıda örneğe göre, YSA'daki bağı ağırlıklarının ayarlanması süreci olarak tanımlanabilir. Amaç, öğrenme sırasında kullanılan örnek desenlerinin tamamı ile bunların istenen çıkış desenlerini doğru bir biçimde ilişkilendirebilecek benzersiz bir ağırlıklar kümesine varmaktır. İki ana öğrenme tipi vardır: öğreticili ve öğreticisiz. Öğreticili öğrenme; bazı giriş örnekleri ile bunların çıkışlarının/istenen tepkilerinin ağı verildiğı tipteki öğrenmedir. Bu tür bir öğrenmede, istenen çıkışlar ağı doğru cevapları öğretme amacıyla kullanılırlar. Öte yandan, öğreticisiz öğrenmede, sadece giriş örnekleri ağı verilir ve ağı aldığı giriş uyarılarına dayanarak kendi kendini düzenlemek zorundadır (Moselhi et al. 1991).

Öğrenme işlemi için, genellikle, bir eğitme algoritması kullanılır ve bir öğrenme örneğine göre ağırlıkların nasıl düzenleneceğı bu algoritma tarafından belirlenir (Çizelge 3.1).

Çizelge 3.1 Sinir ağı eğitme algoritmaları

Eğitme algoritması (1)	Kaynak (2)	Öğrenme tipi (3)	Ağırlık Düzenlenmesi (4)
Hebbian	Hebb (1949)	Öğreticili	Gerçekçi/Deterministic
Perceptron	Rosenblatt (1959, 1961)	Öğreticili	Gerçekçi
Widrow-Hoff	Widrow and Hoff (1959)	Öğreticili	Gerçekçi
Grossberg	Grossberg (1974)	Öğreticili	Gerçekçi
Delta rule	Rumelhart et al. (1986)	Öğreticili	Gerçekçi
Boltzman	Hinton and Sejnowski (1986)	Öğreticili	İhtimalci/Probabilistic
Kohonen	Kohonen (1984)	Öğreticisiz	Gerçekçi

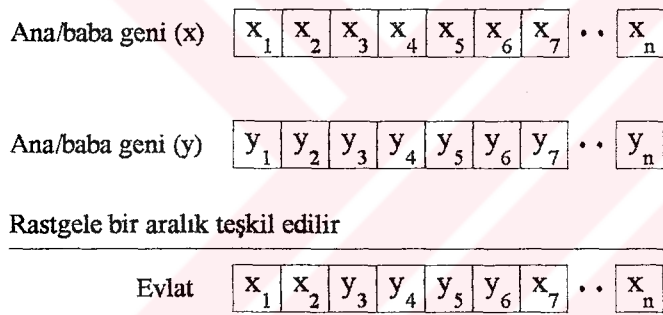
Tipik bir öğrenme oturumunda, öğrenme sürecini sona erdirmek için önceden belirlenmiş belirli bir kritere ulaşıncaya kadar öğrenme örnekleri ağı defalarca gösterilir. Böyle bir kriter, bütün öğrenme örneklerinde, ağ tarafından öngörülen çıkış ile istenen çıkış arasındaki hata belirlenen bir sınırdan az ise, ağı yeterince öğrendi kabul etmektir. Öğrenme işlemleri genelde çok uzun işlemci zamanı tüketirler. Genelde ağ eğitme hatasını kabul edilebilir bir seviyede en aza indirebilmek için, birkaç bin eğitme döngüsü yeterli olmaktadır (Moselhi 1996). Ana amaç, ağı genelleme kapasitesini ilerletmektir. Yani, YSA'nın eğitme sırasında kullanılmamış test örnekleriyle test edilmesi sırasında karşılaşılan hatanın minimum olmasıdır.

Bu safhada;

- 1- Öğrenme katsayısı başlangıçta 0.7 ye eşit alınıp daha sonra eğitme sırasında düşürülebilir. (Bu tez çalışmasında uygulanmıştır.)
- 2- Ağ ağırlıkları ayarlanmalı veya nadiren $\pm 10\%$ civarlarında tutulmalıdır.
- 3- Eğitme örnekleri sayısı artırılmalıdır.
- 4- Aynı zamanda, YSA'nın yapısı veya özellikleri gözden geçirilmelidir.

5- Genetik algoritmalar (GA) yararlanılabilir³⁶. Genetik algoritmalar, biyolojik ve sosyal sistemlerin değişim geçirmesinden esinlenerek geliştirilmiş olan optimizasyon teknikleridir.

Genetik algoritmaları uygulamak için tasarlanmış, eğitilmiş ve test edilmiş asıl YSA yı kullanarak ana/baba diye tabir edilen bir rastgele ağırlar popülasyonu teşkil edilir. Bu ana/babalar eğitilir, test edilir ve bir hedefi veya istenen performansı gerçekleştirme derecelerine göre kademelendirilirler. Optimize edilecek ağı parametrelerini içermek üzere eğitilmiş her bir ana/babayı temsil eden bir “gen” veya “kromozom” teşkil edilmelidir. Örneğin, eğer ağırlıklar optimize edilecekse, “gen” bu ağırlıkları içeren bir vektör olabilir. Optimizasyon süreci için esas olan da bu genin ağı karakteristiklerini içinde tutmasıdır. Özel “genetik operasyonlar”la ana/baba gen çiftlerinin evlendirilmeleriyle yeni kuşaklar veya evlatlar teşkil edilir. Her bir evlat eğitilir ve test edilir.



Şekil 3.3 Ana/baba-evlat oluşumu

Eğer evlat popülasyon içindeki en kötü performansa sahip ana/babayı bulup çıkarırsa, evlat ana/babanın yerine geçer, aksi takdirde kendisi atılır. Yeni evlatların teşkili ve test edilmesi süreci, bazı hedeflenen performanslara ulaşıncaya veya popülasyon içindeki tüm ağların performansı daha fazla ilerlemeyinceye kadar devam eder. Bunu yanında iyi eğitim setleri kullanmaya da gayret edilmelidir. Bu durum, özellikle eksik ve gürültülü verilerle karşılaşıldığında daha da önem kazanır.

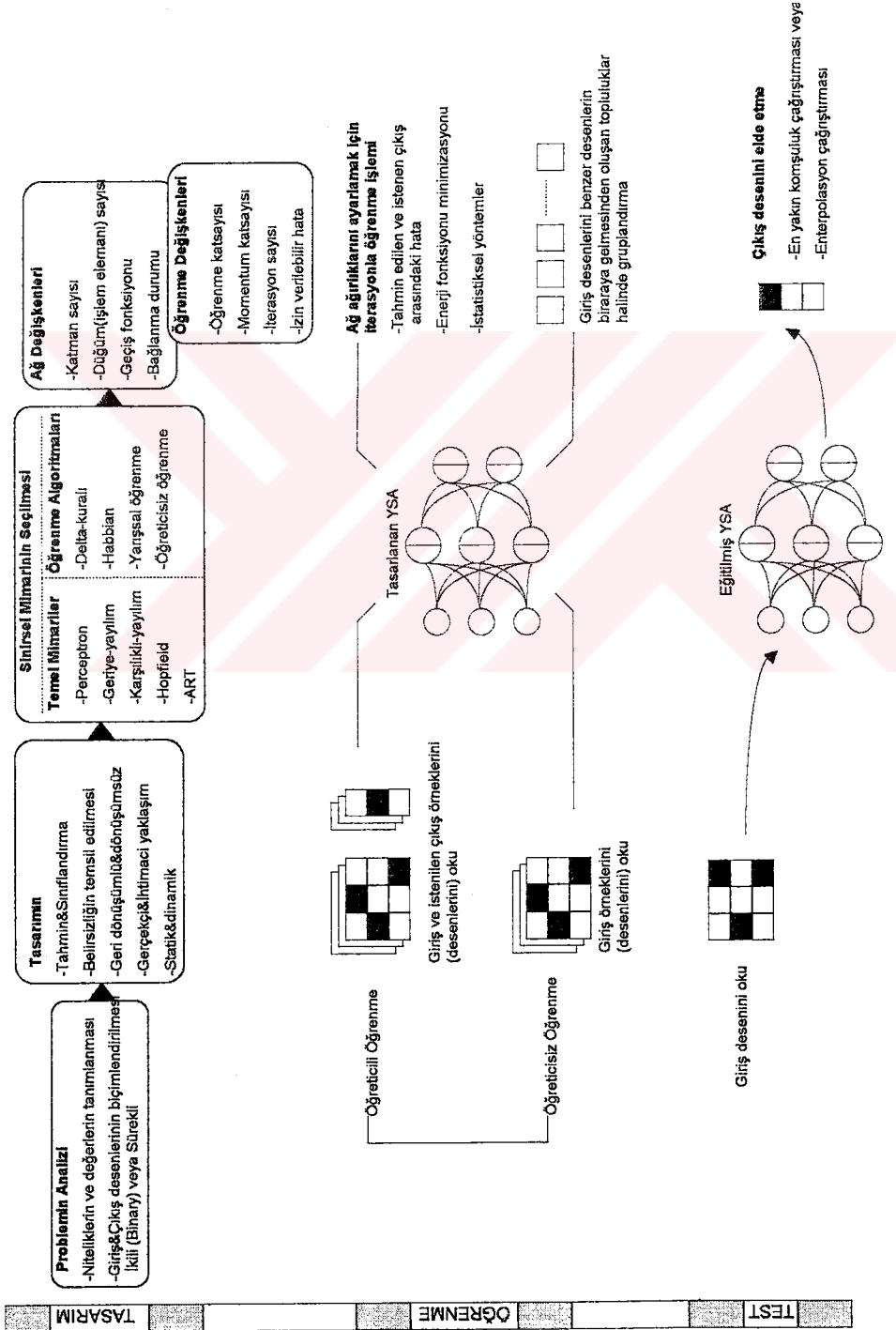
³⁶ Dodd, N. 1991. Optimization of Neural Network Structure using Genetic Techniques. *Proceedings of the 6th International Conference on Artificial Intelligence in Engineering*, 939-944, Oxford, UK.

3.2.3. Test/Uygulama

YSA başarılı bir şekilde eğitildikten sonra artık danışılmaya hazır hale gelmiştir. YSA aslında eğitime sırasında edindiği bilgileri, eğitime sırasında kullanılmamış (daha önce karışmadığı yada görmediği) durumlar için çözümler üreterek “genelleme” (İng. generalization) yapar.

Test işlemi sonunda, test veri seti için YSA dan elde edilen sonuçlar yeterli yaklaşıklık sağlıyorsa, artık YSA nın güvenilirliği onaylanmış olur. Eğer yeterli doğruluk elde edilemiyorsa, YSA nın mimarisinde değişiklikler yapılarak eğitime ve test işlemleri terarlanır. Ayrıca unutulmamalıdır ki; daha yaklaşık sonuçlar, eldeki eğitim setinin problemin çözüm aralıklarını daha iyi temsil edecek şekilde genişletilmesiyle/eğitim setlerinin artırılmasıyla elde edilebilir.

Şekil 3.4’de YSA uygulamalarının geliştirilmesi safhaları bir blok diyagram olarak verilmiştir.



Şekil 3.4 YSA uygulamaları geliştirme safhaları: (a) Tasarım , (b) Öğrenme, (c) Test/uygulama

Çizelge 3.2 Yaygın olarak kullanılan sinirsel hesap mimarilerine ait genel bilgiler (Bailey and Thompson 1990)

Mimari	Kabul edilen giriş tipi	Çıkış tipi	Transfer fonksiyonu	Katman sayısı	Gizli katmanların boyutu	Bağlılık	Öğrenme algoritmaları	Eğitime metodu	Ağ tipi
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Perceptron	Sürekli	Reel sayı, sınıf	Perceptron	1	N/A	Rastgele	Perceptron yaklaşımı, delta kuralı	Öğreticili	Geri dönüşümsüz
Back-Propagation	Sürekli	Desen, Reel sayı, sınıf	Sigmoid, hiperbolik tanjant	2 veya daha fazla	Küçük ile orta arası	Tam bağlı, rastgele	Genelleştirilmiş delta kuralı	Öğreticili	Geri dönüşümsüz
Counter-Propagation	Sürekli	Desen, Reel sayı, sınıf	Kohonen ve sigmoid	2 veya 4	Veri kategorisi sayısına eşit	Tam bağlı	Kohonen ve Grossberg	Öğreticili, Öğeticisiz	Geri dönüşümsüz
Boltzmann makinesi	İkili, Sürekli	Desen, Reel sayı, sınıf	Değişken	2 veya daha fazla	Küçük ile orta arası	Tam bağlı, rastgele	Boltzmann	Öğreticili	Geri dönüşümsüz
Hopfield ağı	İkili, bipolar ^a	Desen, sınıf	Sıkı sınırlayıcı	1	N/A	Tam bağlı	Hopfield	Öğreticili	Geri dönüşümlü
BAM	Bipolar	Desen, sınıf	Kenetli lineer	1	N/A	Tam bağlı	BAM	Öğreticili	Geri dönüşümlü
ART 2	Gri ölçek ^b	Desen, sınıf	Sigmoid	1	Veri kategorisine bağlı olarak artar	Tam bağlı	ART-2	Öğreticisiz	-

^a İkili sistemde (0 ve 1)'e karşılık gelen (-1 ve +1) değerleri.^b Doğru/Yanlış değerlerine karşılık gelen aktivasyon seviyesi

Çizelge 3.3 Yaygın olarak kullanılan sinirsel hesap mimarilerinin performans karakteristikleri (Bailey and Thompson 1990)

Ağ mimarisi	Geliştiricisi	Eğitime süresi	İşletme süresi	Bilgi içeriği	Avantajları	Dezavantajları (Sınırlamaları)	Faydalanma/ Kullanım
Perceptron	Rosenblatt (1961)	Orta	Hızlı	Düşük	*Çok kolay	* Sınırlı temsil kapasitesi	Düşük
Hatanın geriye yayılması (Back Propagation)	Rumelhart et al. (1986)	Yavaş	Hızlı	Yüksek	* (a) Güçlü ve doğru ilişkilendirme * (b) Statik ortam için uygun (Girişin zamana göre değişmediği hallerde)	* (1) Bölgesel minimum veya yetersizlik problemlerine takılabilir. * (2) Gerçek-zaman uygulamaları için uygun değildir. * (3) Kademeli öğrenme yoktur.	Yüksek
Karşılıklı yayılım (Counter propagation)	Hecht-Nielsen (1987, 1988)	Orta	Hızlı	Yüksek	* Hızlı modelleme için iyi * (b) * Bir fonksiyon ve onun tersini üretebilir. (Wassermann 1989) * Güçlü ihtimal kapasitesine sahiptir.	* Back-propagation kadar doğru değildir. * (1), (2), ve (3)	Yüksek
Boltzmann	Hinton and Sejnowski (1986)	Yavaş	Yavaş	Yüksek	* Bölgesel minimum problemini azaltır. * İhtimale dayalı eğitime metodu kullanır.	* Çok yavaş * (2) ve (3)	Yüksek
Hopfield ağı	Hopfield (1982, 1984)	Hızlı	Orta	Düşük	* (c) Dinamik (gerçek-zaman) performans sağlar. * (d) Gürültülü, eksik ve daha önce görülmemiş girişlere yeterince cevap verir. * (e) Optimizasyon için kullanılabilir.	* Stabilité garantisi değildir. * (4) Hafızaya depolanmış örnekler kullandığından hafıza sınırlaması vardır. * Eğitime metodunu formalize etmek güçtür. * (1) ve (3)	Yüksek
BAM	Kosko (1987)	Hızlı	Hızlı	Düşük	* (c), (d) ve (e) * Hopfield'den daha kolay ve sistematik eğitime metodolojisine sahiptir. * Stabilité problemi yoktur. * Uyarlanabilir performans sağlar.	* (1), (3) ve (4)	Düşük
ART-2	Carpenter and Grossberg (1987)	Hızlı	Orta	Yüksek	* Önceden edinilmiş tecrübelerle zarar vermeden yeni örnekleri öğrenmeyi kolaylaştırır. (Kademeli öğrenme)		Yüksek

3.3. Yapı Mekanğinde YSA'nın Kullanılabileceği Alanlar

Bilindiği gibi YSA, herhangi bir sisteme ait mevcut veriler arasındaki nonlinear ilişkileri modellemelerinden ve yine bu veriler arasındaki ilişkiyi öğrendikten sonra, daha önceden öğrenmedikleri veriler için hemen anında makul sonuçlar üretebilmelerinden dolayı, bu tür kabiliyetlere ihtiyaç duyulan her türlü problemde kullanılmaları mümkün olacaktır.

(a) Deneysel Parametrik Çalışmalarda YSA

Bilindiği gibi parametrik çalışmalarda, problem içinde bir veya daha fazla parametreye çeşitli değerler verilerek problem tekrar tekrar çözülür. Eğer problem deneysel türde ise, problemin YSA modellenmesi sayesinde, araştırma sırasında farklı özelliklere/boyutlara sahip numuneler üzerinde yapılacak deneylerin, sonuca ulaşmak için gerekenden daha az tekrarlanması yeterli olacaktır. Böylece hem zamandan, hem de ilave deney numuneleri için kullanılacak olan deney malzemesinden tasarruf edilmiş olacaktır. Ayrıca, betonarme gibi nonlinear özellik taşıyan ve bulunduğu ortamdan ve şartlardan oldukça etkilenen numuneler üzerinde yapılan çalışmaların YSA modellemesi sırasında, deney sonucuna dolaylı etkisi olabilecek, çevre şartları ve ortamın yeterince tarif edilmesi ve ilave parametreler olarak YSA'ya verilmesi, deney sonuçlarının daha sağlıklı bir şekilde değerlendirilebilmesini mümkün kılacaktır.

(b) Deneysel Olmayan Parametrik Çalışmalarda YSA

Eğer çözümü yapılacak olan problem deneysel değilse, yani teorik türde bir problemse, yine problemin YSA modeli avantajlar sağlayacaktır. Örnek olarak, bir plaklar sisteminin üç boyutlu olarak incelenmesi ve plaktaki moment, kesme kuvveti ve normal kuvvet dağılımlarının belirlenmesi istensin. Ayrıca sistemin değişik boyutlar ve yükleme durumları için tekrar tekrar çözümü gerekli olsun. Böyle bir problemin çözümü için değişik yöntemler önerilebilir. Problemi sonlu elemanlar yöntemine dayalı bir bilgisayar programı ile çözelim. Plaklar sistemi, yeterli sayıda sonlu kabuk (shell) elemanlara bölünecek ve programa verilecektir. Bu tür problemlerde bilgisayar

programı çok büyük boyutlara sahip matrislerle işlem yapacak ve bu da oldukça zaman alacaktır. Eğer, çözülecek sistem sayısı çok fazla ise, veya boyutlar üzerinde çok küçük adımlarla olacak çok sayıdaki değişim incelenecek ise, sistemin bir YSA modeli hazırlanır ve YSA'nın problemi öğrenmesi için yeterli sayıda sistem önce mevcut herhangi bir analiz programı vasıtasıyla çözülür ve sisteme ait giriş verileri ile elde edilen sonuçlar YSA'ya tanıtılır ve böylece YSA'nın öğrenmesi sağlanmış olur. Bundan sonra, YSA, değişik boyut ve yükleme durumları için sisteme ait çözümü anında verecektir.



4. ÖZEL BAZI YSA UYGULAMALARI

4.1. Hasar Belirleme İşlemlerinde YSA ve Uzman Sistemler

Yapıda meydana gelen hasarlar, normal kullanım, kazalar, çevresel etkiler ile deprem veya fırtınalar gibi sert doğal olaylar sonucu oluşabilir. Çoğu zaman hasarın yeri ve düzeyi gözlemsel araştırma ile belirlenebilir. Hasar tahmini için en yaygın metod gözlemsel incelemelerdir. Ancak, bazı durumlarda gözlemsel inceleme verimli olmayabilir. Büyük ve kompleks yapıların gözleme dayalı incelemesi zor olabilir ve ulaşım problemlerinden ötürü pahalı olabilir. Bazı durumlarda gözlemsel incelemenin emniyetsiz olduğu da söylenebilir. Böyle yapılara örnek olarak, petrol arama ve üretim platformları ile uçak gövdeleri gösterilebilir. Son yıllarda, yapı hasar tahmini için dolaylı teşhis metodolojisi birçok araştırmacının ilgisini çekmiştir. Etkin ve güvenilir bir hasar tahmin metodolojisi, yapısal elemanların hasarının anında belirlenmesi için oldukça önemlidir. Bu yüzden yapı hasarının otomatik bir gözlem metoduyla belirlenmesine çalışılacaktır. Yapı hasarı için yukarıda sıralanan bir dizi sebepten en önemlisi ve burada incelenecek olanı, yapıda depremden doğan hasarlardır.

Güçlü yer hareketlerine maruz kalan binalarda meydana gelen hasarın hangi düzeylerde gerçekleştiğinin belirlenmesi için, uzman sistemler ve YSA gibi otomatik gözlem metodları kullanılması incelenmiştir. Genellikle depremlerden hemen sonra değişik düzeylerde tecrübeye sahip arazi ekipleri tarafından saha verileri toplanır. Ele alınan bir binanın hasar düzeyine ait net kararlar verilebilmesi için, hiyerarşik bir biçimde hazırlanan sorular çerçevesinde kişisel kararlar büyük ölçüde ortadan kaldırılabilir.

Deprem sonrası hasar düzeyinin belirlenmesi konusu, özellikle sıkça depreme maruz kalan ülkelerde çözüm bekleyen önemli bir sorun olagelmıştır. Bu özelliği ile konu, bilim dünyasının hep gündeminde kalmış ve pek çok çalışmaya esas oluşturmıştır.

Hasar düzeyi belirleme yöntemleri üzerinde yürütülen çalışmalar, bilgisayar teknolojisindeki gelişmelerin de yardımıyla, son dönemlerde oldukça ilerlemiştir.

Park et al.³⁷ deprem hareketine maruz kalan betonarme yapıların uğrayacağı hasar düzeyini belirleyen bir metodoloji önermektedir. Bu yöntemde yapının hasar durumu, uğradığı maksimum deformasyon ve toplam histerik enerji cinsinden ifade edilmektedir. Önerilen metodoloji, geçmiş depremlerde hasara uğramış yapılar üzerinde denenmiş ve bu yolla kalibre edilmiştir. Kabul edilen eksikliklerine rağmen, bu çalışmalarda ileri sürülen hasar modeli, hem düktilite hem de histerik enerji yok edilmesini içine aldığı için halen en yaygın olarak kullanılan modeldir.

Pagnoni et al.³⁸ tarafından yapılan bir diğer çalışmada ise, deprem sonrası hasar tesbiti için görevlendirilen, ancak bu konuda yeterli bilgi ve deneyime sahip olmayan kimselere yardımcı olması amacıyla AMADEUS isimli bir uzman sistem oluşturulmuştur. AMADEUS, yığma yapıların incelenmesi ve hasarlarının belirlenmesi amacıyla geliştirilen bir rehber sistemdir. Gözlemlerden elde edilen bilgilerin depolanmasını sağlamak üzere, sistem, bir veri tabanı programı ile desteklenmiştir. Böyle bir sistemin düzenli, etkili ve gerçekçi bir hasar tesbiti sağlayacağı belirtilmiştir.

Shwe ve Adeli³⁹ tarafından yapılan bir çalışmada, deprem hareketi sırasında zorlanan veya hasara maruz kalan binaların dayanımını belirlemek amacıyla, bilgisayar grafikleri ve uzman sistemler geliştirmek yoluyla bir yöntem ortaya konulmuştur. Bu çerçevede, Kaliforniya depremlerinde hasar görmüş bir dizi binanın uğradığı hasarı belirlemek üzere, geçmiş depremlerden elde edinilen bilgileri içeren EXQUAKE2 diye adlandırdıkları bir uzman sistem geliştirilmiştir. Bu sistemde Exsys “shell”i kullanılmış olup, uzman sistemlerin yapısal hasara ait bilgileri toplamak ve kullanmak açısından çok gerekli olduğu sonucuna varılmıştır.

³⁷ Park, Y.J., Ang, A.H., Wen, Y.K. 1985. Seismic damage analysis of reinforced concrete buildings. *Journal of Structural Engineering* 4:740-757.

³⁸ Pagnoni, T., Tazir, Z.H., Gavarini, C. 1989. AMEDEUS: A KBS for the Assessment of Earthquake Damaged Buildings, *Report of the IABSE Colloquium on Expert Systems in Civil Engineering*, 141-150, Bergamo, Italy

³⁹ Shwe, T.T., Adeli, H. 1993. AI and CAD for earthquake damage evaluation. *Engineering Structures* 5:315-319.

Shepherd ve Haynes⁴⁰, Los Angeles yapı kodu çerçevesinde bir ve iki katlı yığma binaların hasarını belirleyen bir uzman sistemi tanıtmaktadır. Bu sistem binanın genel özelliklerinden hareketle, standarda uygun olarak duvarlarda ve çeşitli yapısal elemanlarda oluşan yatay kuvvetleri tahmin etmektedir. Bu uzman sistem üç prototip yapıya uygulanmıştır.

Wu et al. (1992), yapı hasarını belirlemek için otomatik gözleme metodlarını kullanmayı amaçlayan bir çalışmanın ilk safhasını gerçekleştirmişlerdir. Çalışmalarındaki temel strateji, bir YSA yı hasarsız bir yapıya ait verileri kullanarak mümkün olan çeşitli hasarlara uğramış bir yapının davranışını tanımak üzere eğitmektir. Yapısal tepki ölçümleri bu eğitilen ağa verildiğinde, mevcut hasarı belirlemek mümkün olmalıdır. Bu temel düşüncelerini örnek bir minyatür yapıda denemişler ve sonuçları ümit verici bulmuşlardır.

Laboratuvar ortamında gerçekleştirilen deneylerden elde edilen bilgiler yardımıyla tekil yapısal elemanlarda veya basit yapısal alt sistemlerde meydana gelen hasarı, niceliksel bir tarzda ifade etmek mümkündür. Buna karşılık, gerçek bir bina veya başka tür karmaşık yapı sisteminin uğradığı deprem etkileri sonrasında ne mertebede hasar gördüğünün aynı biçimde ortaya konulması, çok daha zor bir iştir ve zorluğun tek sebebi daha çok sayıda taşıyıcı eleman olması değildir. Laboratuvarda olduğu gibi, alınmış olan ölçüm ve gözlemlerin elde bulunmaması, elemanların geçirdiği zorlamaların ne mertebede olduğunun dolaylı biçimde tahmin edilmesini zorunlu kılar. Üstelik hasarın göstergeleri, farklı tür fonksiyon gösteren elemanlar için farklı olabilir, yada elemanların uğradığı bozulmaların birikimi farklı sistemler için farklı manalara gelebilir. Başka bir deyişle hasarın belirlenmesi işlemi, eski bilgi ve gözlemlerin “en iyi” bir tahminde bulunmak üzere yeniden kullanılması diye adlandırılabilir.

Uzman sistemler, elemanların bölgesel hasar düzeylerini genellikle hasarın şiddeti, elemanın nisbi önemi ve hasar gören eleman sayısına bağlı olarak belirlenmektedirler.

⁴⁰ Shepherd, R., Haynes, T.E 1993. Expert System and Assessment of Earthquake Hazard Reduction, *Proceedings of IABSE Colloquium on Knowledge-Based Systems in Civil Engineering*, Beijing, China, 351-360.

Hasara yolaçabilecek faktörler ve sistemin hasar görebilme özelliğinden doğan hasar puanları bu puanla toplanmaktadır. Toplam hasar puanı 0 ile 100 arasında değişen rakamsal değerler cinsinden ifade edilmektedir. Bu puanlar yardımıyla yapının hasar düzeyi yaklaşık olarak tahmin edilmektedir. Çizelge 5.1’de EPADA adındaki bir uzman sisteme ait puanlandırma görülmektedir. (Gülkan ve ark. 1995)

Çizelge 4.1 EPADA adlı bir uzman sistemde yapıya ait global hasar düzeyinin puanlandırılması

Toplam hasar puanı (THP)	Hasar düzeyi
$0 < \text{THP} < 5$	Hasarsız
$6 \leq \text{THP} \leq 14$	Az hasarlı
$15 \leq \text{THP} \leq 43$	Orta hasarlı
$\text{THP} > 43$	Ağır hasarlı

Geçmişte meydana gelen depremlerin öğrettiği dersler, oldukça iyi bir şekilde belgelendirilmiştir. Elde edilen bu bilginin daha geniş bir yarar sağlaması için ,kuvvetli yer hareketlerine maruz kalmış sistemlerdeki kategorize edilmiş hasar belirtilerinin bir YSA veya uzman sistem yardımıyla ileride olabilecek depremlerden etkilenen sistemlere uzatılabilmesi gereklidir. Böyle bir uygulama hasar tazminatı, sigorta vb. amaçlar için objektif bir karar verme ortamı meydana getireceği için, sadece teorik bir irdeleme niteliği taşımamaktadır. Bir hasar tesbit işlemi sonucu elde edilecek bilgi, ekonomik onarım programlarının geliştirilmesinde önemli bir rol oynayacaktır. Yurdumuzda geçerli olan “Afetler Kanunu”nda yeralan hükümler doğrultusunda konutu veya işyeri tabii afetlerden etkilenen “hak sahiplerine” bu yerlerin uğradığı hasarla orantılı tazminat verilir. Demek ki, hasarın doğru ve tartışmaya yer vermeyecek biçimde tayini, afet zararlarının giderilmesinde objektif bir zemin hazırlayacaktır. Bu işlem, aynı zamanda afet sonrası hasar tesbitlerinin hızlandırılması ve zaman zaman uzman olmayan kimselerce yapılan bu tesbitlerin de daha güvenilir hale getirilmesini sağlayacaktır. Bunun da ötesinde depremden hemen sonra daha artçı sarsıntıların devam ettiği süre içinde hangi binaların güvenli, hangilerinin ise can

güvenliği bakımından sakıncalı olduğunun ayırddedilmesi, bu nitelikte bir sistem ile sağlanabilir. Uzman sistemler yardımıyla hasar tesbiti veya nicelik olarak ifade edilmesi sırasında karşılaşılan bazı özel durumlar ve belirsizlikler bulunmaktadır. İlki, hasar parametrelerinin kesin yargı ifadeleri ile belirlenmesindeki imkansızlıktır. Başka bir deyişle hasar parametresi olarak kabul edilen değişkenler her durum altında sabit değildir. Kişisel yargıya bağlı olabildiği gibi, bir durumdan diğerine “taşınması” kolay olmamaktadır; dolayısı ile bilimsel ifadeyle bulanık bir kümenin elemanlarıyla çalışma mecburiyeti vardır. İkinci güçlük, uzman sistemlerin muhakeme yürütme ve öğrenme yeteneklerinin bulunmamasıdır. Parametreler aynı olsa dahi her durumun değerlendirilmesi sıfırdan başlayan yeni bir süreç meydana getirmektedir. Bu zorluğun aşılması da YSA'nın formülasyonda işin içine sokulmasını gerekli kılmaktadır.

4.1.1. Hasar tesbitine YSA tabanlı yaklaşım

Literatürde önerilen hasar tahmin metotlarının çoğu, az veya çok aynı yaklaşımı takibederler. Bu klasik hasar tahmin metotlarında birçok temel adımlar tanımlanabilir. İlk olarak, yapı için bir matematiksel model kurulur. Daha sonra, bu matematiksel model yardımıyla yapısal davranışın anlaşılması sağlanarak spesifik eleman hasar durumları ile yapısal tepkideki değişimler arasında korelasyonlar kurulur. Bu matematik modeller, lineer olarak sebeplerden sonuçlara ilerleyen direk işlem modellerinin bir parçasıdır. Ancak, hasarlı yapının tepkisinden eleman hasarının belirlenmesi tersine bir işlemdir, sebepler sonuçlardan çıkarılmalıdır. Teşhis adımı gözlenen yapısal davranışın nedenlerinin kavranılması için yapılan bir araştırmadır. Bu işlem çok hesap gerektirir ve iyi geliştirilmiş bir veri izleme sistemine ihtiyaç gösterir (Wu et al. 1992).

Bu çalışmada, YSA kullanımı ile hasarlı ve hasarsız yapının tepkisindeki desenlere ait bilginin depolanması ve kullanılması sağlanacaktır. Bu yüzden matematiksel modellerin, kavramaya yönelik tersine arama işlemine ihtiyaç kalmamaktadır. Hasar miktarının belirlenmesi için YSA'nın hesaplama yetenekleri kullanılacaktır.

Temel yaklaşım dinamik tepkideki ve/veya yapının dinamik karakteristiklerindeki değişikliklerin tesbit edilmesidir. Bu değişiklikler çeşitli hasar durumları ile dolaylı olarak ilişkilidir. Herhangi bir hasar tesbitinin konusu yapısal hasarın meydana gelip gelmediği ve eğer meydana geldiyse hasarın yeri ve derecesinin belirlenmesidir.

Araştırmacılar, hasardan önce ve sonra yapının ölçülmüş dinamik tepkilerinin analizine dayanan hasar tahmin çizelgeleri önermişlerdir. Önerilen çizelgelerden bazıları sadece yapının uygulanan yükler altında frekans tepkilerini veya çevresel vibrasyon testlerinin frekans içeriğini kullanmaktadır (Burke et al. 1980, Coppolino et al. 1980, Rubin et al. 1980, Shahrivar et al. 1984, Vandiver et al. 1975). Bu alandaki daha yeni çalışmalar, yapının mod şekillerine ait bilgiden de faydalanmaktadır (Shahrivar et al. 1989). Kompleks yapıların mevcut matematiksel modelleri analiz ve tasarım amaçları için yeteri kadar doğru olsalar da, bu modeller yapının dinamik tepkisindeki gözlenen değişimlere hangi elemanların sebep olduğunun teşhisinde kendi içlerinde yeterli değildirler. Sağlıklı bir hasar tahmin metodolojisi, bireysel eleman hasarının sebep olduğu yapı tepkisinde gözlenen desenleri tanıyabilmeli, eleman hasarının derecesini belirleyebilmelidir. YSA'nın desen karşılaştırma yetenekleri bu metodolojinin gereklerini karşılayabilir (Wu et al. 1992).

YSA'nın kendi kendini düzenleme (İng. self-organization) ve öğrenme kabiliyetlerinden yararlanarak yapı hasarı tahmin edilmeye çalışılacaktır. YSA, hasar görmemiş yapı davranışı ile muhtemel çeşitli hasarlara maruz kalmış yapı davranışını tanımak üzere eğitilir. Eğitilmiş YSA ya yapısal tepki ölçümleri verildiğinde, mevcut hasarın tesbiti mümkün olacaktır. Bu temel düşünce, basit bir yapı üzerinde deneysel olarak Wu et al (1990) tarafından gerçekleştirilmiştir.

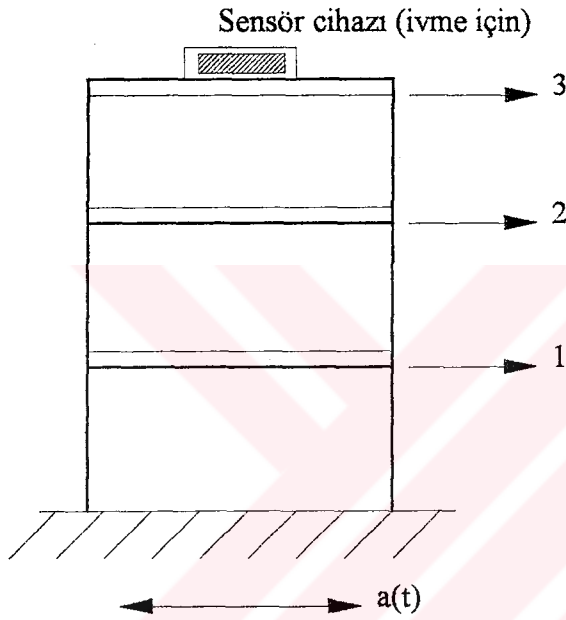
Bir yapısal sistemdeki hasar tahminine YSA ya dayalı yaklaşımın geliştirilmesindeki temel strateji bir geriye yayımlı YSA'yı yapıların ölçülmüş tepkilerinden yola çıkarak bireysel eleman hasarını tanıyacak şekilde eğitmektir. Bu strateji herhangi bir yapının ölçülmüş tepkisinin doğal frekansları ve titreşim mod şekilleri ile ilgili bilgiyi içermesi gerçeğine dayanmaktadır. Yapısal hasar, yapının bu dinamik karakteristiklerini etkilemektedir. Bu yüzden teorik olarak, hasarsız bir yapının tepkisiyle karşılaştırıldığında hasarlı bir yapının ölçülen tepkisi, hasarın yeri ve derecesi hakkında

bilgi içermektedir. Ancak, bu bilginin klasik metotlarla elde edilmesi son derece zor olabilir; sebep sonuç ilişkilerine dair kurallar açıkça ortaya konulmalı ve bu kuralları kullanmak için bir metodoloji geliştirilmelidir. YSA tabanlı yaklaşımda, uygun bir geriye yayımlı YSA'nın eğitilmesi ile sebep sonuç ilişkileri elde edilebilir ve ardından elde edilen bu ilişki YSA'nın bağ kuvvetlerinde depolanabilir. Bu yolla, YSA'nın kendi kendisini organize etme ve öğrenme yetenekleri sebep-sonuç ilişkisini elde etme ihtiyacını ortadan kaldırmak için kullanılmış olacaktır. Bu ilişkiler, YSA'nın bağ kuvvetlerinde yayılmış bir şekilde mevcuttur; fakat gözlemlenebilir veya kullanıcının incelemesine sunulabilir şekilde değildirler.

YSA ile yapılan hasar tahmin metodunun esası, sebep-sonuç ilişkileri ile ilgili bilgileri içeren uygun veriler ile bir YSA'nın eğitilmesi işlemidir. Böylece, ilk adım eğitimde kullanılabilecek veri seti ve yapının tepkisinden doğan hasar desenlerini tanımak için uygun bir YSA'nın üretilmesidir. İdealde, bu veri seti çeşitli hasar durumlarındaki yapıya ait tepkiler ile birlikte hasarsız yapının tepkilerini de içermelidir. Bu veriler, yapısal tepkinin ölçülmesi, model testi sonuçları veya sayısal simülasyon/benzetme yollarından biri veya bu üç tip verilerin bir kombinasyonu olarak üretilebilir. Wu et al. (1992), yaptıkları çalışmada sayısal simülasyon yöntemini kullanmışlardır.

Herhangi bir tipik hatanın geriye yayılması algoritmasını kullanan bir YSA uygulamasında, uygun YSA mimarisi belirlenmeli ve bir eğitim algoritması seçilmelidir. Daha sonra bu YSA, eğitim seti ile eğitilir. Eğitilen YSA, eğitildiği durumları hangi ölçüde öğrendiğini kontrol etmek üzere test edilir. Son adım, genellikle YSA'nın genelleme yeteneğinin test edilmesi işlemidir. Bu adımda, eğitilen YSA, eğitim veri setinde mevcut olmayan veriler ile test edilmektedir. Bu son adım aslında YSA'nın performans yeteneğinin testidir. Eğitilmiş bir YSA'nın ne kadar iyi bir genelleme yapabileceği seçilen YSA mimarisinin yeterliliğinden ve hasar tahmini için gerekli bilgiyi içeren eğitim veri setinin zenginliğinden büyük derecede etkilenir. Sık sık YSA mimarisinde değişiklik yapılmasına ve/veya eğitim setine ilavelere gerek duyulur. Böyle değişimler bütün eğitim ve test işleminin tekrarlanmasını gerekli kılmaktadır. Yeterli performans elde edilinceye kadar bu tür yarı iteratif ince ayarlamalar tekrarlanır.

Wu et al.(1992), yaptıkları deneysel çalışmada basit bir üç katlı çerçeve modeli kullanmışlardır (Şekil 4.1). Çerçeve depremsel sarsıntıya uğratılmış ve yapısal dinamik tepkiler çerçevenin en üst katına yerleştirilmiş bir sensör cihazı ile tesbit edilerek Newmark metodu yardımı ile belirli zaman periyotları için yapının ölçülmüş tepkisi olarak ivme zaman kayıtları hesaplanmıştır. Daha sonra bu ivme zaman kayıtları bir FFT (Fast Fourier Transformation-Hızlı Furier Dönüşümü) işleminden geçirerek elde edilen ivme-zaman kayıtlarının Fourier spektrumlarını YSA ya giriş verisi olarak kullanmışlardır (Şekil 4.2).



Şekil 4.1 YSA'nın eğitilmesinde kullanılacak olan yapısal tepki verilerinin elde edilmesi için hazırlanmış deney düzeneği (Wu et al. 1992)

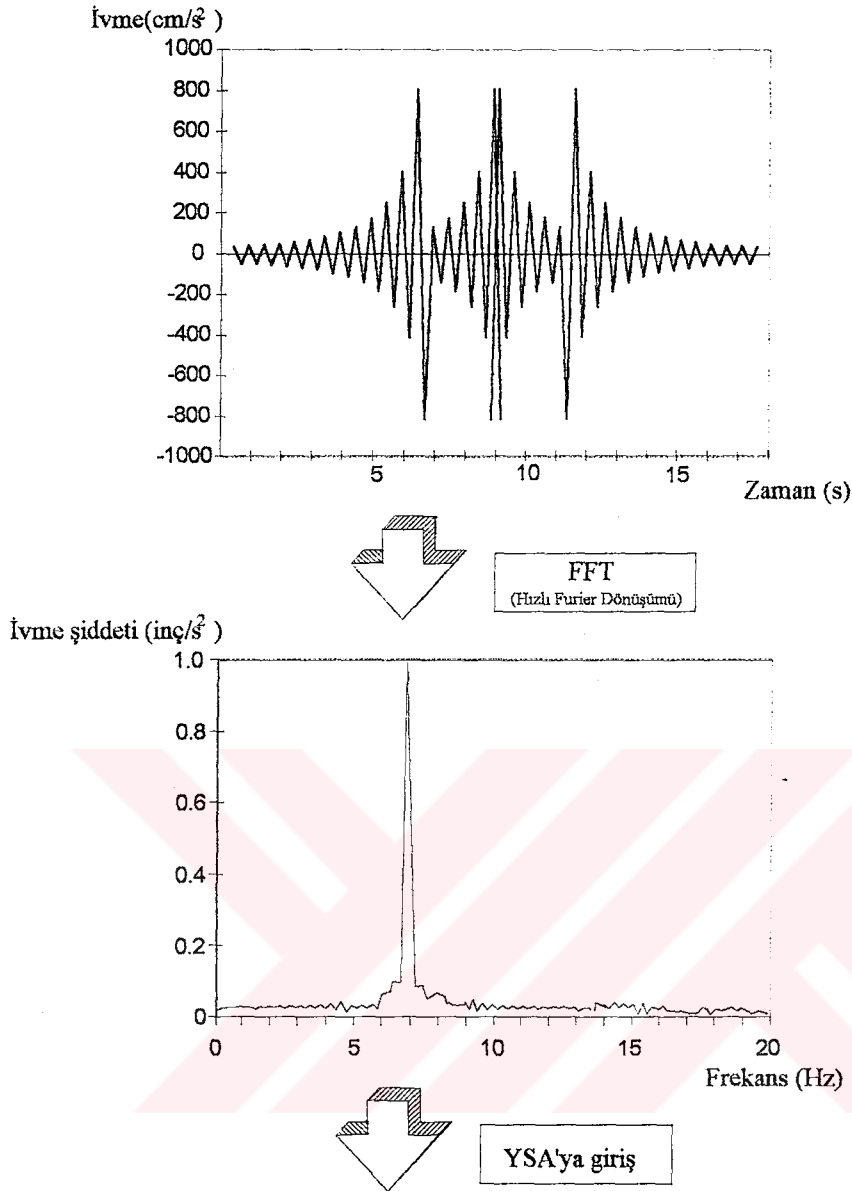
Eleman hasarı, eleman rijitliğindeki azalma olarak tanımlanmıştır. Değişik eleman hasar durumları için yapının üçüncü kattaki dinamik tepkileri hesaplanmıştır. Araştırma amacıyla, belirli bir zamanda yalnız bir eleman hasar görebilmiştir; yapısal elemanın hasarı, rijitliğinde belli bir yüzde oranındaki azalma ile gösterilmiş ve elemanın hasar durumu YSA çıkışından elde edilecek rijitliğin tutulma yüzdesi olarak tanımlanmıştır. Yani, eğer yapıda hasar yoksa, hasar durumu %100 olarak verilmiştir (Şekil 4.3). Yapıya ait rölatif ivme tepkisini zaman aralığından frekans aralığına dönüştürme amacıyla FFT kullanılmıştır (Şekil 4.2).

Çalışmada basit bir ağ şekli kullanılmıştır; her bir işlem elemanı ilerideki katmandaki tüm düğümlere bağlanmıştır (Şekil 4.3). YSA'nın kapasitesi eğitim verisindeki bilgi miktarına ve o verideki bilginin komplekslik derecesine bir şekilde bağlı olmaktadır. Halihazırda ileri-beslemeli çok katmanlı bir YSA için güvenilir kurallar mevcut değildir. 0~20Hz arasındaki Fourier spektrumu 0.1Hz'lik aralıklara bölünerek YSA giriş katmanındaki 200 düğüme girilmiş, böylece her bir düğüm bir frekans bölmesini temsil etmiştir (Şekil 4.3). Çok çeşitli sayıda gizli katman denenmiştir. Genellikle, bir YSA'nın kapasitesi gizli katman sayısının, her bir katmandaki düğüm sayısının ve katmanlar arasındaki bağlanma deseninin bir fonksiyonudur⁴¹. Çalışmada tek gizli katmanda 10 gizli düğüm yeterli görülmüştür. Çıkış katmanı eleman başına bir düğüm içermektedir ve bu düğümlerdeki aktivasyon değerleri, o elemandaki hasar düzeyini temsil etmektedir; aktivasyon değeri 1 ise hasarın olmadığı ve aktivasyon değeri 0 ise tamamen hasarlı olduğu anlamına gelmektedir.

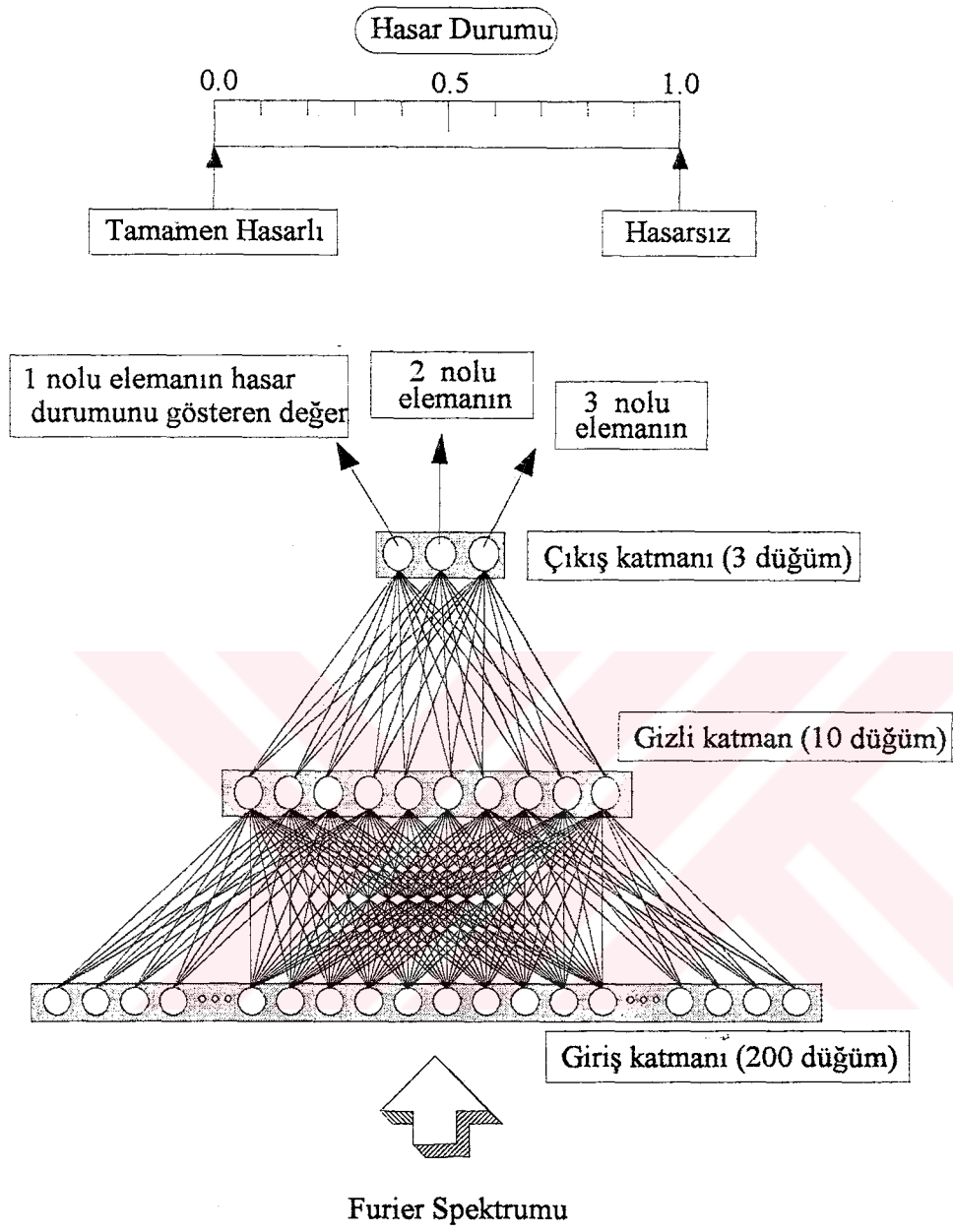
YSA modelinin eğitilmesinde 42 eğitim seti kullanılmış ve yeterli öğrenme elde edildikten sonra YSA'nın daha önce karşılaşmadığı hasar spektrumları ile test edilmesine geçilmiştir. YSA, 3. elemana ait hasarı yeterli derecede belirleyebilmesine rağmen, diğer iki eleman için yetersiz kalmıştır. Bunun nedeni olarak 1 ve 2 elemanlarına ait hasar spektrumlarının ağa yeterince öğretilmediği ve bu yüzden eğitim setinin genişletilmesi gerektiği sonucuna varılmıştır.

Wu et al. (1992), her ne kadar test sonuçlarında bahsedilen uygunsuz durumlarla karşılaşmış olsalar da, hasar tahmini işleminde YSA kullanımını oldukça ümit verici bulmuşlardır.

⁴¹ Hornik, K., Stinchcombe, M., White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:356-366.



Şekil 4.2 Deney düzeneğinden elde edilen ivme-zaman grafiğinin hızlı furier dönüşümü (FFT, Fast Fourier Transformation) ile ivme şiddeti-frekans grafiğine dönüştürülerek ağa giriş için hazırlanması (Wu et al. 1992)



Şekil 4.3 YSA mimarisi (200:10:3) ve yapı elemanlarının hasar durumu (Wu et al. 1992)

4.2. Malzeme Davranışının YSA Kullanımı ile Bilgi Tabanlı Modellemesi

Genel olarak, günümüze kadar malzeme modellemesi, deneysel verilerin gözlemlenmesi ve elde edilen sonuçlara dayalı matematiksel modellerin çıkarılmasını gerektiriyordu. Bu amaç için diğer bir alternatif yol ise, YSA denilen bir hesaplama ve bilgi temsil mimarisini kullanmaktır.

Gevrek kırılma, duktıl akma, mikro çatlamlar ve gerilme düşmeleri gibi kompleks davranışları inceleyen malzeme modellemesi konusundaki modern araştırmalar gerilme-şekil değiştirme ilişkisini tanımlayabilmek için matematiksel modeller geliştirmeyi amaçlamaktadır. Bu modeller, bu değişken ve kompleks davranışı açıklayabilen matematiksel kurallar ve ifadelerden oluşurlar. Hooke zamanından şimdiye kadar değişik davranışlar için bu malzeme modelleri az veya çok benzer yollarla geliştirilmişlerdir. Bunun için:

- (1) Bir malzeme test edilir ve davranışı gözlenir;
- (2) Gözlemlenen davranışı açıklamak için bir matematiksel model varsayılır ve malzeme parametreleri belirlenir;
- (3) Matematiksel model henüz test edilmemiş gerilme çizgilerini tahmin etmek amacıyla kullanılır ve mevcut veya yeni deneylerden elde edilen sonuçlarla kontrol edilir;
- (4) Daha sonra gözlemlenmiş fakat model tarafından açıklanmamış davranışları hesaba katmak amacıyla bu matematiksel model üzerinde değişiklikler yapılır.

YSA denilen yeni hesap mimarisi ise, malzeme davranışı ilişkilerinin modellenmesi için temelde yukarıdaki işlem basamaklarından tamamen farklı bir yaklaşım sağlamaktadır.

YSA alanındaki araştırmalar, malzeme davranışının modellenmesine ve çıkarılmasına ait tamamıyla farklı bir yaklaşımın geliştirilmesini sağlamıştır. Bu yeni yaklaşımla, malzeme davranışına ait bilgi deneysel verilerle eğitilmiş olan kendi kendini düzenleyen YSA'nın bağları içinde yakalanır. Daha fazla test verisi elde edildikçe, bu

yeni verileri de hesaba katıp daha yaklaşık modellemeler elde etmek üzere YSA'nın bağ ağırlıklarını değiştirmesi için eğitime işleminde kullanılabilir. YSA gerilme ve şekil değiştirmeler arasındaki ilişkileri düğümleri arasındaki bağ ağırlıkları vasıtasıyla elde etmesine rağmen, açık matematiksel kurallar veya formüller kullanılmamakta veya bu tip malzeme modeli içinde halihazırda gözlemlenememektedir. Ancak, YSA'nın bir hesaplama modeli olamasından ötürü hesaplamalı mekanikte kullanılması teknik olarak herhangi bir sorun çıkarmaz.

4.3. Optimizasyon Problemlerinin Çözümünde YSA

Yapısal optimizasyon, doğrudan tasarım için çok önemli bir hedef olan eldeki kaynaklardan yararlanarak mümkün olan maksimum faydanın(etkinlik) elde edilmesi işlemidir. Yapısal optimizasyon, araştırmacılar ve yapısal tasarımcılar arasında büyük ilgi toplamıştır. Daha ucuz ve hafif yapılara olan ihtiyaç, doğrudan tasarımın çekici yapısı ve doğal kaynakların gereksiz tüketiminden kaçınma arzusu, yapısal optimizasyonu popüler bir araştırma ve uygulama sahası haline getirmektedir. Birçok yayınlarda açıkça görülmektedir ki, yapısal optimizasyon teorisinde birçok araştırmalar yapılmış ve hala gelişme kaydetmektedir (Erbatur 1993).

Burada, YSA'nın sayısal optimizasyon uygulamalarında kullanımı tartışılacaktır. Eğitilmiş bir YSA, diğer sayısal hesaplama mimarilerine göre benzersiz avantajlar sağlamaktadır. Bunlardan en önemlisi verilen giriş ile istenen çıkış değerleri arasında uygun ilişkiyi kurarak yeniden tasarım (İng. redesign) sürecini çok hızlı ve etkin bir hale getirmektedir. Bu özellik, analiz içinde lineer olmayan özelliklerin bulunması halinde, analiz ve optimizasyon için yeniden tasarımın iteratif bir yaklaşım gerektirdiği durumlarda daha büyük bir önem arz etmektedir. YSA analitik veya deneysel verilerin kullanılması yoluyla eğitilebilir.

4.3.1. Yapısal optimizasyon

Bir yapısal optimizasyon probleminin bir genel formülasyonu şu şekilde gösterilebilir (Erbatur 1993). Optimizasyon,

$$f(x_i) \quad (4.1)$$

fonksiyonunu

$$g_j(x) > 0, \quad j=1,2,\dots,m \quad (4.2)$$

$$h_k(x) = 0, \quad k=1,2,\dots,m \quad (4.3)$$

ifadelerini sağlayacak şekilde bir minimum/maksimum yapan

$$x_i, \quad i= 1,2,\dots,n \quad (x_{ia} < x_i < x_{id})$$

değişkenlerinin bulunması problemidir.

(4.1) ifadesindeki,

- x_i , tasarım değişkenleridir. Bunlar, herhangi bir kafes sistemdeki çubukların alanları, kolon ve kiriş kesitinin genişlik ve yükseklikleri, kabuk ve levha kalınlıkları gibi parametreler olabilir.
- $f(x_i)$, x_i nin maksimum yada minimum yapılacağı amaç (İng. objective) fonksiyonudur. Amaç fonksiyonu ağırlık, hacim, maliyet veya burkulma yükü, bir deplasman bileşeni vb. gibi diğer problem parametreleri olabilir.
- (4.2) ve (4.3) ifadelerinde yer alan g_j ve h_k , sırasıyla tasarım probleminin eşitsizlik veya eşitlik sınırlamalarıdır. Yapısal optimizasyon problemleri için sınırlamalar, genellikle yapının tepkisine (davranış sınırları) ve/veya yapının geometrisi ve kesit boyutlarına (geometrik ve teknolojik sınırlamalara) bağlı olarak tanımlanır.

Yapısal tasarım uygulamalarına ait sınırlamalar genel olarak x değişkenlerinde non-lineerdir. (4.3)'te verildiği gibi eşitlik sınırlamaları da düşünülebilir, ancak yapısal

tasarım problemlerinde bu tür sınırlamalar genellikle yer almazlar. Dikkate alınan davranış parametreleri, çeşitli yükleme durumları altında gerilme, deplasman ve frekans sınırlamalarıdır. Herbir yükleme durumu için gerilme sınırlamaları,

$$g_j = \left| \frac{\sigma_j}{\sigma_{j0}} \right| - 1.0 \leq 0 \quad (4.4)$$

ifadesi ile belirlenir. Burada σ_j , j.ninci eleman için tasarım gerilmesidir ve σ_{j0} , j.ninci eleman için emniyet gerilmesidir. Herbir yükleme durumu için deplasman sınırlamaları,

$$g_{js+j} = \left| \frac{u_j}{u_{j0}} \right| - 1.0 \leq 0 \quad (4.5)$$

ifadesi ile verilir. Burada u_j , j.ninci deplasman bileşeni, u_{j0} j.ninci deplasman bileşeni için deplasman sınırı ve js toplam gerilme sınırlamaları sayısıdır. Frekanslardaki sınırlamalar

$$g_f = \left(\frac{f_{n0}}{f_n} \right)^2 - 1.0 \leq 0 \quad (4.6)$$

ifadesi ile elde edilir. Burada f_n yapının doğal frekanslarını ve f_{n0} bu frekanslardaki sınırlamaları temsil eder.

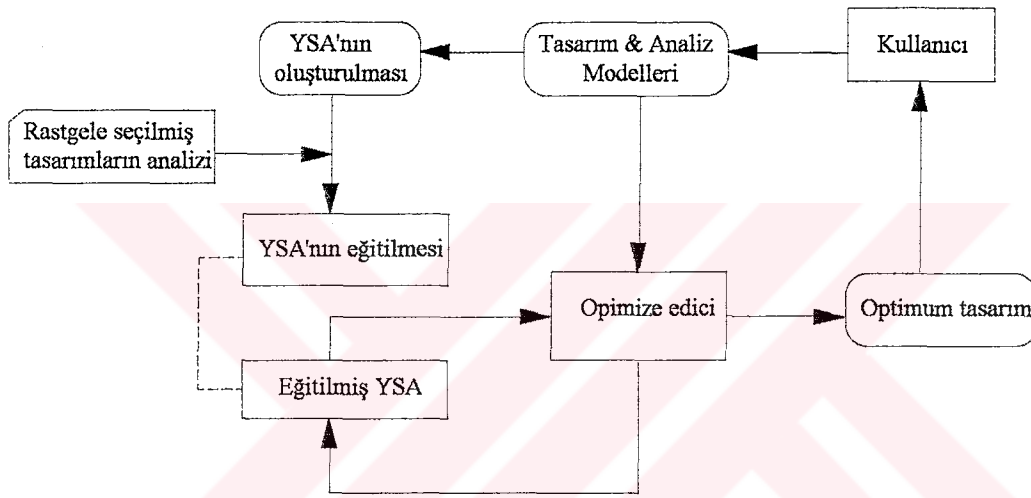
Matematiksel bir programlama tekniğinde, x_{opt} optimum tasarım noktasına bir x_0 tasarım başlangıcından başlanarak, diyelim K iterasyonda ulaşılır. Tasarım, herbir iterasyonda doğrultu vektörü $\bar{\phi}_k$ ve adım uzunluğu α nın hesaplanmasıyla yinelenir. Optimum tasarım işlemi şu şekilde formüle edilebilir (Berke et al. 1993):

$$x_{opt} = x_0 + \sum_{i=1}^K \alpha_k \bar{\phi}_k \quad (4.7)$$

Burada $\bar{\phi}_k$ k.ninci iterasyondaki doğrultu vektörü ve α_k da $\bar{\phi}_k$ doğrultu vektörü boyunca adım uzunluğudur. k.ninci iterasyonda, $\bar{\phi}_k$ doğrultu vektörü amaç fonksiyonunun gradyentlerinden/türevlerinden ve aktif sınırlamaları sağlayacak şekilde,

mevcut doğrultu üretme algoritmalarının birinin uygulanmasıyla oluşturulur. Doğrultu vektörü $\bar{\phi}_k$ boyunca, α_k nın elde edilmesi için yine birçok mevcut prosedürlerin birinden faydalanarak tekboyutlu bir araştırma gerçekleştirilir. Yinelenen tasarım, daha sonra bir veya daha çok durma kriteri ile kontrol edilir ve iterasyon işlemi yaklaşıklık kazanıncaya kadar tekrarlanır. Literatürde kapsamlı şekilde anlatılmakta olan lineer olmayan matematiksel programlama tekniklerine burada yerverilmeyecektir.

4.3.2. YSA ile optimizasyon



Şekil 4.4 YSA ile optimizasyonun akış diyagramı

Herhangi bir sayısal optimizasyon probleminin YSA ile çözümüne ait işlem basamakları Şekil 4.4'te gösterilmiştir. Öncelikle tasarım probleminin YSA modellemesi yapılmalıdır. Probleme ait amaç fonksiyonunda veya sınırlamalarda yer alan tasarım değişkenleri genellikle giriş katmanında ve sistemin yapısal analizi sonucu bulunacak olan büyüklükler ise çıkış katmanında yer almalıdırlar. Gizli katman ve gizli düğümlerin sayısı giriş ve çıkış katmanındaki düğüm sayılarına ve sistemdeki lineer olmayan büyüklüklere bağlı olarak çeşitli denemeler sonucunda belirlenebilir.

YSA nın eğitilmesinde kullanılacak olan eğitim verileri seti ise, YSA nın giriş düğümlerinde yer alan tasarım değişkenlerine rastgele değerler verilerek, sistemin sonlu elemanlar, matris-deplasman veya matris-kuvvet metodu gibi bilinen sayısal analiz

metotlarından birisiyle analizinin yapılması sonucu çıkış katmanındaki büyüklüklerin bulunması ve bu işlemin yeteri kadar tekrarlanmasıyla hazırlanır. YSA elde edilen bu eğitim seti ile eğitilerek giriş ve çıkış düğümleri arasındaki ilişkiyi en iyi bir şekilde öğrenmesi sağlanır. Bu arada Şekil 4.4'te yer almayan bir ara işlem olarak, YSA'nın test edilmesi işleminin gerçekleştirilmesi de uygun görülebilir. Bir test seti, aynen eğitim setinin hazırlanmasına benzer bir biçimde oluşturulur. YSA bu test seti ile test edildiğinde hata düzeyi kabul edilebilir sınırlar içinde kalıyorsa işleme devam edilir, aksi takdirde eğitme işlemine geri dönlür.

Eğitilmiş ve test edilmiş bir YSA, artık optimizasyon sürecindeki analiz işleminin yerini alabilecektir. Optimizasyona, probleme ait sınırlamaları sağlayan rastgele bir giriş değerleri seti ile başlanabilir. (4.7)'deki x_{opt} değerinin bulunması için yapılacak araştırma sırasında tesbit edilecek noktalarda, tasarım değişkenleri dışındaki bilinmeyenlerin bulunmasına yönelik olarak yapılacak olan yapısal analiz işlemi, yerini YSA'ya terkedecektir. Optimizasyon işlem basamaklarında bunun haricinde bir değişiklik yapılmayacaktır.

Optimizasyon sürecine YSA'nın katılmasındaki amaç, optimumun araştırılması sırasında sistemin tekrar tekrar çözülmesi ile kaybedilen zaman ve bunun getirdiği külfetlerden kaçınmaktır. YSA eğitim setinin hazırlanması sırasında ve eğitme sırasında kaybedilecek olan zaman, eğitim seti sayısına ve YSA'nın katmanlarındaki düğüm sayılarına bağlı olarak değişecektir. YSA ile optimizasyon işlemi, oldukça geniş bir eğitim setinin kullanılması halinde veya çok fazla gizli katman ve düğüm sayısına sahip YSA modellerinin söz konusu olduğu zamanlarda fizibilitesini/verimliliğini yitirebilmektedir. Böyle zamanlarda giriş-çıkış eğitim çiftlerinin gruplandırılması (İng. clustering of training pairs) işlemi eğitme sırasındaki zaman kaybını büyük ölçüde azaltabilmektedir (Hajela et al. 1991).

Çok sayıda elemana sahip kompleks sistemler ile lineer olmayan çok sayıda değişkenin yer aldığı problemlerin analizi, bilinen sayısal analiz metotları ile oldukça uzun sürelerde yapılabilmekte ve/veya sisteme ait lineer olmayan değişkenlerin tanımlanması sırasında güçlükler yaşanabilmektedir. Bu tür sistemlerin YSA ile optimizasyonu, büyük olasılıkla daha kısa sürelerde uygun optimumların bulunmasını

sağlayabilecektir. Çok basit sistemlerin optimizasyonu sırasında ise sayısal ve/veya klasik analiz işlemlerinin kolay olması, YSA ile optimizasyona gerek bırakmayabilir.

4.3.3. Optimizasyonda YSA'nın yeri

Yapısal optimizasyon araştırması yeni metotlara da açıktır. Erbatur (1993), özellikle YSA'nın yapı mekaniğinde ve yapıların optimum tasarımında kullanımından bahsetmiştir.

YSA ile yapısal optimizasyon konusu oldukça yeni bir araştırma sahası olduğundan, henüz literatürde yapılmış çalışma sayısı çok azdır. Yöntemin etkinliği ve sonuçların güvenilirliği tartışılmaktadır.

4.4. Bir Yapısal Kontrol Yöntemi Olarak Sinir Ağları

Yapısal kontrolün (İng. structural control) amacı yapının dış yükler altındaki hareketini belli bir düzeye indirmek fakat tamamıyla ortadan kaldırmamaktır. Son yıllarda yapısal kontrol alanında artan araştırma çalışmaları ile, birçok yapısal kontrol yöntemleri önerilmiştir. Bu yöntemlerin çoğu sistemin tanımlanmasına ve/veya analizine gereksinim duyarlar. Belli düzeylerde doğrusal olmayan ve doğrusal modeller yaygın olarak geliştirilebilmelerine karşın kontrol algoritmalarında problemin tanımlanmasının ve ele alınmasının çok güç birçok diğer yönleri vardır. Kontrol halkasında önceki döngülerden kalan gecikmelerden doğan problemlerin üstesinden gelinmesi de oldukça güçtür. Kısacası doğrusal olmayan davranışlara, belirsizliklere ve ölçme uygunsuzluklarına sebep olan birçok kaynak mevcuttur. Bunlar da zayıf matematiksel modeller ve etkin olmayan kontrol algoritmaları ile sonuçlanmaktadır. YSA tabanlı kontrol algoritmaları, bu problemlerin birçoğunu aşabilecek yeteneklere sahiptirler (Ghaboussi 1995).

YSA, kontrol problemlerinde etkin bir yöntem olarak tanınmaktadır⁴². Klasik kontrol yöntemleri ile sinirsel kontrol etme mekanizmaları (İng. neurocontrollers) arasındaki en önemli fark kontrol edici mekanizmanın yeteneklerini elde etme biçimidir. Klasik kontrol yöntemleri spesifik bir kontrol işlemi gerçekleştirmek üzere programlanırlar. YSA ile kontrolde ise kontrol işlemi öğrenilir. Öğrenme, öğreticili veya öğreticisiz (İng. supervised or unsupervised) olarak gerçekleştirilebilir. YSA lar aşırı karmaşık, doğrusal olmayan ilişkileri öğrenebilir. YSA lar öğrenme yeteneklerinden dolayı, aynı zamanda adapte edilebilir (İng. adaptive) kontrol mekanizmalarıdır. Yeniden öğrenme yoluyla sistemin değişen durumuna ve değişen kontrol gereksinimlerine göre adapte olabilirler.

Deprem etkisine karşı bir yapının kontrolü sırasında birçok doğrusal olmayan davranış ortaya çıkmaktadır. Bunlar, aşırı deformasyonlardan doğan ve geometrik olarak doğrusal olmayan davranış ile yapının çeşitli bölgelerinde meydana gelen hasarlardan kaynaklanan malzemenin doğrusal olmayan davranışdır. Ayrıca, yapıya yük tesiri sağlayacak olan cihazların sınırlı kapasitelerinden doğan doğrusal olmayan davranışlar da söz konusu olabilmektedir. YSA nın doğrusal olmayan davranışları modelleme kabiliyetlerinden ötürü, yapı kontrolü probleminde etkin olarak kullanımları mümkündür.

⁴² Miller, W.T., Sutton, R.S., Werbos, P.J. 1990. *Neural Networks For Control*, MIT Press, Cambridge, Mass.

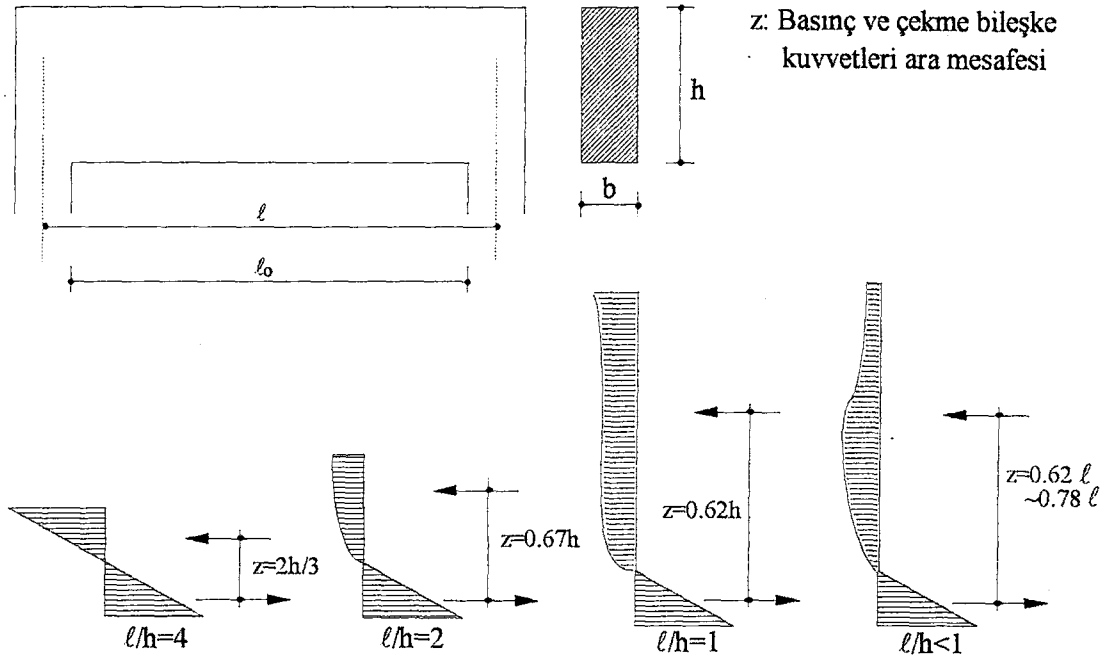
Warwick, K., Irwin, G.W., Hunt, K.J. 1992. *Neural networks for control and systems*, IEEE Control Engrg. Ser. 46, New York, N.Y.

5. ÖRNEK SAYISAL YSA UYGULAMALARI

5.1. Yüksek Kirişlerde Kesme Mukavemetinin Belirlenmesi Problemi

Yüksek kirişler ℓ/h oranının çok küçük olması nedeni ile klasik kiriş davranışı göstermeyen kirişlerdir. TS-500 deki tanıma göre, iki moment sıfır noktası arasındaki uzaklığın derinliğe oranı (ℓ/h) 2 den az olduğunda, söz konusu kiriş “Yüksek kiriş” olarak hesaplanmalıdır.

Yüksek kirişlerin davranışını incelemek üzere yapılan deneyler, çatlama oluştuktan sonra davranışın, elastisite teorisinden saptanan davranıştan oldukça farklı olduğunu göstermiş ve elastisite teorisinden elde edilen sonuçların emniyetsiz yönde olduğunu ispatlamıştır. Kirişlerde gerilme yayılımının belirlenmesi sırasında düzlem kesitlerin, şekil değiştirmeden sonra da düzlem kaldığı kabul edilir ve bunun sonucu olarak kiriş gövdesinde şekil değiştirmelerin doğrusal olarak değiştiği ortaya çıkar. Ancak yüksekliğin açıklığa oranla artması, gerilme yayılımını doğrusal olmaktan çıkarır ve yüksek kirişlerde klasik hesap yöntemlerinden ayrılmayı gerektirir. Şekil 5.1’de homojen malzemeden yapılmış bir kiriş için verilen gerilme diyagramlarından anlaşılacağı gibi, yüksekliğin artması ile basınç gerilmelerinin yayılımı düzgünleşir ve yüksekliğin açıklıktan fazla olması durumunda üst tarafta gerilmesi düşük olan bir bölge ortaya çıkar. Ayrıca yüksek kirişlerde, diğer kirişlerden farklı olarak kayma deformasyonları ihmal edilemeyecek kadar önemli olmaktadır.



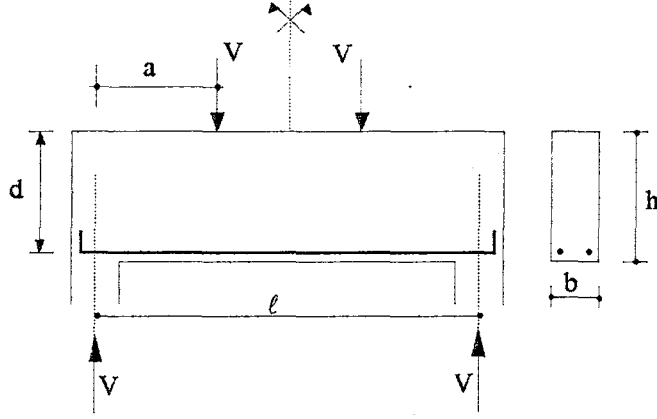
Şekil 5.1 Kirişlerde açıklık gerilmesi yayılımı

Çalışmanın bu bölümünde, yüksek kirişlerde yeterli gövde donatısının bulunmadığı durumlarda, asal çekme gerilmelerinden oluşan eğik çatlakların kırılmaya yol açtığı kırılma türü incelenecektir. Yüksek kirişlerde kesme mukavemetinin belirlenmesi konusunda çok sayıda deneysel ve analitik çalışmalar yapılmıştır. Burada ise, kesme mukavemetinin tahmini için YSA kullanılacaktır. YSA, beton mukavemeti, kiriş geometrisi ve kirişteki donatı gibi birçok doğrusal olmayan değişkeni olan problemler için oldukça yararlı bir metottur. Değişkenler arasında önceden tanımlanmış bir matematiksel bağıntı mevcut değildir. Bunun yerine YSA, mevcut deney verilerinden yola çıkarak öğrenir (Siao 1993). İşlemlerde, mesnetlerden eşit uzaklıkta simetrik olarak uygulanan iki tekil yük ile yüklenmiş bir yüksek kiriş modeli kullanılacaktır (Şekil 5.2).

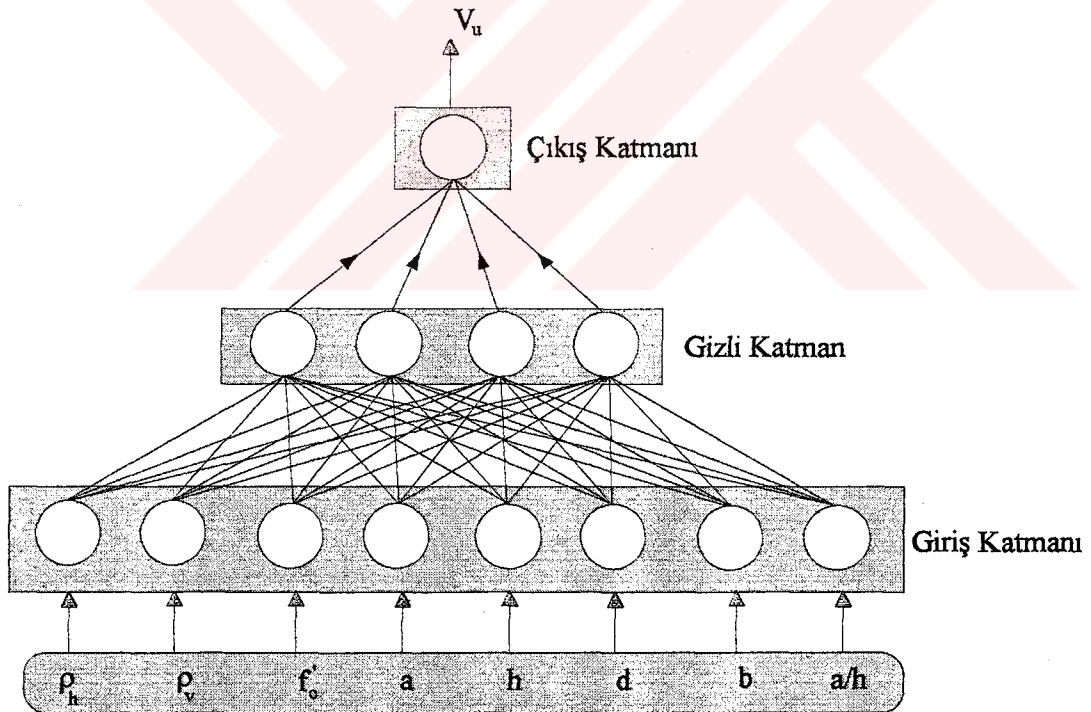
5.1.1. Problemin YSA ile modellenmesi

YSA modellemesi için giriş verileri olarak ρ_h , ρ_v , f'_c , d , b , a , h ve a/h değerleri kullanılması uygun görülmüştür. ρ_h ve ρ_v sırasıyla enine ve boyuna donatı oranıdır. f'_c ,

MPa cinsinden beton silindir mukavemeti, d , b , a ve h ise Şekil 5.2’de gösterilen kiriş geometrisine ait değişkenlerdir. Çıkış değişkeni ise V_u , kN cinsinden kiriş kayma mukavemetidir. V_u deneylerden elde edilen kayma mukavemeti dayanımıdır (Şekil 5.3).



Şekil 5.2 Kiriş ve yükleme durumu (yalnız alttaki donatı gösterilmiştir.)



Şekil 5.3 Yüksek kiriş probleminin YSA modeli.

5.1.2. Değişik YSA modeli seçeneklerinin eğitilmesi

YSA modelinin giriş ve çıkış katmanında yer alan değişkenler dolayısı ile giriş-çıkış katmanı düğüm sayıları belli olduğuna göre, geriye yalnızca gizli katman sayısı ile herbir gizli katmandaki düğüm sayısının belirlenmesi kalmaktadır. Bilindiği gibi, gizli katman sayısı doğrusal olmayan değişkenlerin derecesi ile ilgilidir; bu yüzden, bu problemde bir adet gizli katman yeterli görülmüştür. Gizli katmandaki düğüm sayısının belirlenmesi için, halihazırda bir metot mevcut değildir. Bu çalışmada da deneme-yanılma/tatonman (İng. trial-and-error) yapılarak gizli düğüm sayılarından uygun olanı belirlenmeye çalışılacaktır. Bu yüzden 1 den 8 e kadar gizli düğüm sayısı kullanılarak YSA modelleri oluşturulacaktır. Eğitim işlemi sonunda herbir modeldeki minimum ortalama hatadan veya minimum standart sapmadan yola çıkarak uygun model seçilecektir (Şekil 5.3).

YSA nın eğitilmesi için gerekli veriler, Siao (1993) tarafından yapılan çalışmadan alınmıştır. Ayrıca burada, referansta önerilen kafes giriş modelinden (Strut-and-Tie Model) elde edilen sonuçlarla, YSA sonuçları da karşılaştırılacaktır.

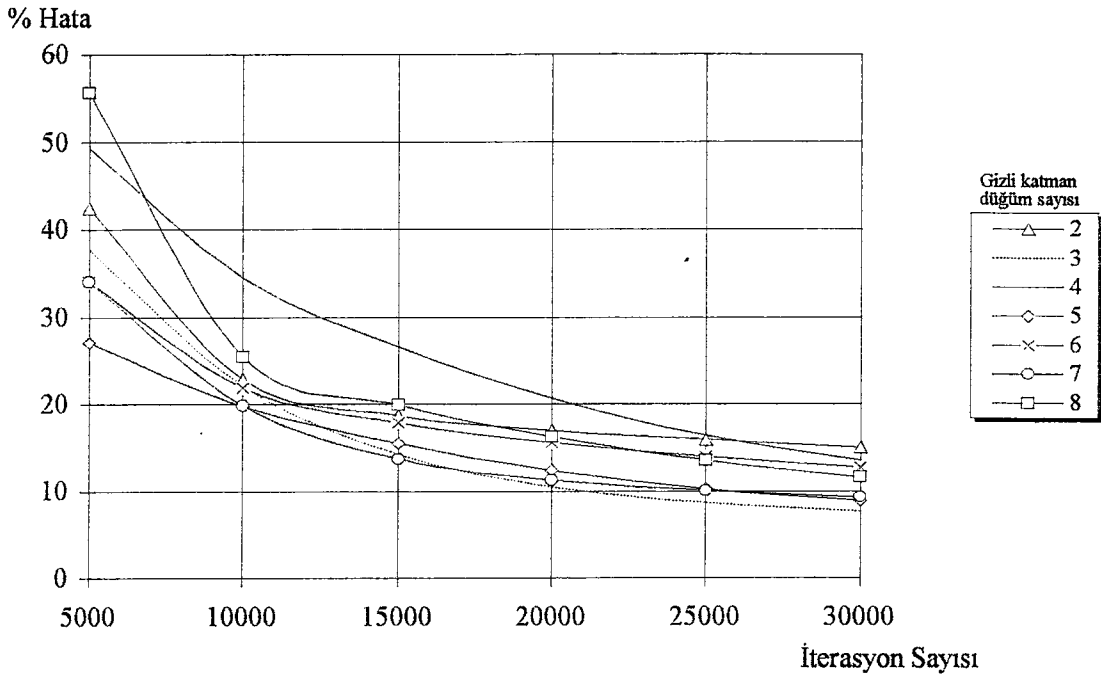
Kullanılan eğitim veri setleri Çizelge 5.1'de verilmiştir. Eğitim işlemi sırasında bu veriler ağa binlerce defa tanıtılacak ve böylece YSA nın, 8 adet giriş ve 1 adet çıkış düğümü arasındaki bağ ağırlıklarını ayarlayarak, giriş-çıkış verileri arasında bir ilişki kurması sağlanacaktır.

Eğitim işlemi için 30 adet numuneye ait veri giriş ve çıkış değerleri kullanılmıştır. YSA modellememizde transfer fonksiyonu olarak sigmoid fonksiyonunu kullandığımızdan dolayı, verilerin ağa girilmeden önce 0~1 arasında normalize edilmesi gereklidir. Bunun için her bir değişken, o değişkenin alabileceği maksimum değere veya maksimuma yakın bir değere bölünerek 0~1 arasında bir sayı elde edilebilir. YSA dan elde edilecek çıkışlar da, giriş sırasında daha önce kullanılmış bölen ile çarpılarak normalizasyon işlemi tersine çevrilir. Örneğin, bu problemde

$\rho_h / 1.1$, $\rho_v / 2.5$, $f'_c / 60$, $a / 900$, $h / 920$, $d / 900$, $b / 102$, $(a/h) / 1.1$, $V_u / 800$ şeklinde veri normalizasyonu yapılmıştır.

Burada öğrenme modeli olarak “hatanın-geriye yayılımı” (İng. Back-propagation) algoritması kullanılmıştır. Öğrenme algoritması, örnekleri iki aşamada işleme tabi tutmaktadır. İlk aşamada giriş katmanından çıkış katmanına ileriye doğru bir işaret akışı gerçekleşir. Her çıkış düğümü için hata, gözlenen (deneysel sonuç) kesme kuvveti ile YSA dan hesaplanan kesme kuvveti arasındaki farktır. İkinci aşama ise, hesaplanan ve istenen sonuçlar arasındaki farkı (hatayı) azaltmak üzere gizli katman ile çıkış katman arasındaki ağırlıkların yeniden ayarlanması işlemidir. Ağırlıkların değiştirilmesi işlemi “Genelleştirilmiş Delta Kuralı” ile gerçekleştirilmektedir.

Öğrenme momentum katsayısı $\alpha=0.9$ ve öğrenme katsayısı $\epsilon=0.7$ alınmıştır. Hazırlanan bilgisayar programı (EK-1) içinde öğrenme katsayısı azaltılarak hassasiyet artırılacaktır. Başlangıçta ϵ değerinin çok küçük alınması ise yetersiz öğrenmelere sebep olabilmektedir. Eğitim, tüm eğitim örnekleri için hataların karelerinin toplamının ortalamasını minimum yapana kadar tekrar tekrar gerçekleştirilir (iterasyon). Bu çalışma için yaklaşık olarak 30000 iterasyon sonucu sözü edilen minimizasyon gerçekleşmiştir. İterasyon sayısının artırılmasının, Şekil 5.4’ten de anlaşılacağı gibi hatanın azaltılmasına önemli bir etkisi olmayacaktır.



Şekil 5.4 Hatanın iterasyon sayısına bağlı değişiminin farklı gizli katman düğüm sayılarına bağlı değişimi

Çizelge 5.2’de eğitime sırasında kullanılan numunelere ait deneysel V_u kayma mukavemeti değerleri ile farklı gizli düğüm sayıları hallerinde elde edilen YSA sonuçları ayrı ayrı verilmiştir. 3 adet gizli düğüme sahip YSA modelinden elde edilen sonuçlar gerçeğe en yakın değerler almaktadır. Ayrıca eğitim setine ait hata dağılımının daha kolay incelenebilmesi için Çizelge 5.3’te farklı gizli düğüm sayıları için hata yüzdeleri ile ortalama hatalar ve hataların standart sapmaları verilmiştir. Ayrıca maksimum hata yüzdesi ile ortalama hataların minimumu ve hataların standart sapmalarının minimumu koyu karakterlerle gösterilmiştir. Hata yüzdeleri, deneysel sonuçlar ile YSA sonuçları arasındaki farkın mutlak değerinin deneysel sonuçlara bölünmesi ile elde edilmiştir. Uygun gizli düğüm sayısını tesbit ederken, hata ortalamalarına veya standart sapmalara bakmak yeterli olacaktır. Görülüyor ki, ortalama hata yüzdeleri içinde en düşük değer, 3 gizli düğüme sahip YSA modelinde meydana gelmiş olup, değeri %2.6 ve bu YSA modelinde hataların standart sapması ise %2.0 olmaktadır. Eğitime işlemi başarıyla tamamlanmıştır. Sırada eğitilmiş YSA’nın test edilmesi gelmektedir.

Çizelge 5.1 Eğitme işleminde kullanılan yüksek kiriş verileri (Siao 1993)

Sıra No	Numune No	ρ_h -	ρ_v -	f'_c (MPa)	a (mm)	h (mm)	d (mm)	b (mm)	a/h -	V_u kN
1	1A1-10	0.23	0.28	18.690	304.8	355.6	304.8	101.6	0.86	161.25
2	1A3-11	0.45	0.28	18.034	304.8	355.6	304.8	101.6	0.86	148.35
3	1A4-12	0.68	0.28	16.069	304.8	355.6	304.8	101.6	0.86	141.24
4	1A6-37	0.91	0.28	21.034	304.8	355.6	304.8	101.6	0.86	184.10
5	2A3-39	0.45	0.63	19.759	304.8	355.6	304.8	101.6	0.86	170.60
6	2A4-40	0.68	0.63	20.345	304.8	355.6	304.8	101.6	0.86	171.93
7	3A1-42	0.23	1.25	18.414	304.8	355.6	304.8	101.6	0.86	163.70
8	3A4-45	0.68	1.25	20.828	304.8	355.6	304.8	101.6	0.86	178.56
9	3A6-46	0.91	1.25	19.931	304.8	355.6	304.8	101.6	0.86	168.15
10	1B1-04	0.23	0.24	22.069	368.3	355.6	304.8	101.6	1.04	147.46
11	1B6-31	0.91	0.24	19.517	368.3	355.6	304.8	101.6	1.04	153.36
12	2B1-05	0.23	0.42	19.172	368.3	355.6	304.8	101.6	1.04	129.00
13	2B6-32	0.91	0.42	19.759	368.3	355.6	304.8	101.6	1.04	145.24
14	3B1-08	0.23	0.63	16.241	368.3	355.6	304.8	101.6	1.04	130.78
15	3B1-36	0.23	0.77	20.414	368.3	355.6	304.8	101.6	1.04	158.96
16	3B4-34	0.68	0.77	19.241	368.3	355.6	304.8	101.6	1.04	155.03
17	4B1-09	0.23	1.25	17.103	368.3	355.6	304.8	101.6	1.04	153.47
18	1-30	0.00	2.45	21.517	254.0	762.0	723.9	76.2	0.33	238.88
19	1-25	0.00	2.45	24.552	254.0	635.0	596.9	76.2	0.40	224.20
20	1-15	0.00	2.45	21.241	254.0	381.0	342.9	76.2	0.67	164.15
21	2-30	0.00	0.86	19.207	254.0	762.0	723.9	76.2	0.33	249.11
22	2-25	0.00	0.86	18.621	254.0	635.0	596.9	76.2	0.40	224.20
23	2-15	0.00	0.86	22.759	254.0	381.0	342.9	76.2	0.67	139.68
24	5-30	0.61	0.61	18.552	254.0	762.0	723.9	76.2	0.33	239.32
25	5-20	0.61	0.61	20.138	254.0	508.0	469.9	76.2	0.50	172.60
26	5-10	0.61	0.61	22.552	254.0	254.0	215.9	76.2	1.00	77.85
27	4G1	0.35	0.35	52.000	393.7	914.4	863.6	101.6	0.44	648.58
28	4G3	1.01	1.01	54.000	393.7	914.4	863.6	101.6	0.44	798.04
29	4G4	0.34	0.34	52.000	850.9	914.4	863.6	101.6	0.94	461.30
30	2A2	0.51	0.51	28.414	195.6	508.0	457.2	101.6	0.39	307.83

Çizelge 5.2 Eğitim sırasında kullanılan numunelere ait deneysel sonuçlar ve farklı gizli düğüm sayılarına sahip YSA modellerine ait sonuçlar

Sıra No	Numune No	Deneysel V_u (kN)	Gizli Düğüm Sayılarına Bağlı YSA Sonuçları							
			1	2	3	4	5	6	7	8
1	1A1-10	161.25	150.65	157.21	152.90	157.46	157.56	157.98	153.24	156.80
2	1A3-11	148.35	154.50	153.44	154.00	156.32	153.69	151.14	150.86	153.31
3	1A4-12	141.24	157.58	150.77	147.50	151.46	147.80	141.59	147.48	148.50
4	1A6-37	184.10	170.91	183.93	175.64	169.18	184.79	181.08	180.80	180.96
5	2A3-39	170.60	158.63	158.91	167.22	167.37	163.07	161.01	161.04	162.50
6	2A4-40	171.93	165.61	164.03	170.40	169.15	167.03	165.45	166.76	166.43
7	3A1-42	163.70	155.83	166.69	172.86	174.21	170.96	166.14	169.81	170.43
8	3A4-45	178.56	171.62	168.64	176.21	177.50	171.79	173.01	172.71	172.29
9	3A6-46	168.15	178.34	177.33	168.94	174.48	174.74	179.30	176.52	175.81
10	1B1-04	147.46	143.54	144.81	143.14	143.91	143.22	148.86	143.30	145.24
11	1B6-31	153.36	153.25	147.23	147.74	142.35	146.44	149.85	149.94	143.98
12	2B1-05	129.00	142.25	144.84	139.10	140.44	141.27	144.34	141.30	143.28
13	2B6-32	145.24	154.54	148.64	150.90	145.91	149.12	153.02	152.43	146.68
14	3B1-08	130.78	141.12	146.09	135.37	137.63	140.15	141.02	140.15	142.09
15	3B1-36	158.96	144.46	149.04	152.71	150.93	150.46	149.83	150.79	151.60
16	3B4-34	155.03	151.21	142.00	151.87	148.89	144.46	144.98	146.16	144.18
17	4B1-09	153.47	143.98	153.02	152.90	151.61	152.74	147.44	153.05	153.19
18	1-30	238.88	257.84	248.18	237.18	242.93	242.38	244.20	240.38	243.67
19	1-25	224.20	224.24	228.18	230.54	225.81	226.54	225.88	225.81	226.63
20	1-15	164.15	152.43	161.12	157.81	155.25	160.46	159.49	159.76	158.17
21	2-30	249.11	215.34	237.46	246.66	245.76	246.71	244.78	248.78	241.83
22	2-25	224.20	184.38	212.52	215.94	210.33	214.45	213.82	217.38	211.74
23	2-15	139.68	146.00	143.75	143.78	141.71	144.23	151.73	147.86	144.10
24	5-30	239.32	255.10	243.78	240.00	245.10	240.90	245.08	241.00	243.41
25	5-20	172.60	182.03	175.20	180.47	177.13	176.63	166.56	169.56	180.61
26	5-10	77.85	134.05	82.46	78.36	89.92	80.17	79.54	81.34	81.83
27	4G1	648.58	662.75	646.82	648.63	648.09	648.10	649.20	648.54	646.41
28	4G3	798.04	755.55	786.82	784.86	783.31	786.09	781.84	785.10	787.02
29	4G4	461.30	462.11	462.94	462.22	463.60	462.76	461.84	462.38	463.45
30	2A2	307.83	313.29	311.76	309.93	311.32	309.02	311.77	311.83	311.76

Çizelge 5.3 Eğitim sırasında kullanılan numuneler için farklı gizli düğüm sayılarıyla elde edilmiş hata yüzdeleri (Sütunlardaki maksimum hata yüzdeleri koyu olarak gösterilmiştir)

Sıra No	Numune No	Gizli Düğüm Sayısına Bağlı YSA Hataları							
		1	2	3	4	5	6	7	8
1	1A1-10	% 6.6	% 2.5	% 5.2	% 2.3	% 2.3	% 2.0	% 5.0	% 2.8
2	1A3-11	% 4.1	% 3.4	% 3.8	% 5.4	% 3.6	% 1.9	% 1.7	% 3.3
3	1A4-12	% 11.6	% 6.7	% 4.4	% 7.2	% 4.6	% 0.2	% 4.4	% 5.1
4	1A6-37	% 7.2	% 0.1	% 4.6	% 8.1	% 0.4	% 1.6	% 1.8	% 1.7
5	2A3-39	% 7.0	% 6.9	% 2.0	% 1.9	% 4.4	% 5.6	% 5.6	% 4.7
6	2A4-40	% 3.7	% 4.6	% 0.9	% 1.6	% 2.8	% 3.8	% 3.0	% 3.2
7	3A1-42	% 4.8	% 1.8	% 5.6	% 6.4	% 4.4	% 1.5	% 3.7	% 4.1
8	3A4-45	% 3.9	% 5.6	% 1.3	% 0.6	% 3.8	% 3.1	% 3.3	% 3.5
9	3A6-46	% 6.1	% 5.5	% 0.5	% 3.8	% 3.9	% 6.6	% 5.0	% 4.6
10	1B1-04	% 2.7	% 1.8	% 2.9	% 2.4	% 2.9	% 0.9	% 2.8	% 1.5
11	1B6-31	% 0.1	% 4.0	% 3.7	% 7.2	% 4.5	% 2.3	% 2.2	% 6.1
12	2B1-05	% 10.3	% 12.3	% 7.8	% 8.9	% 9.5	% 11.9	% 9.5	% 11.1
13	2B6-32	% 6.4	% 2.3	% 3.9	% 0.5	% 2.7	% 5.4	% 5.0	% 1.0
14	3B1-08	% 7.9	% 11.7	% 3.5	% 5.2	% 7.2	% 7.8	% 7.2	% 8.6
15	3B1-36	% 9.1	% 6.2	% 3.9	% 5.1	% 5.3	% 5.7	% 5.1	% 4.6
16	3B4-34	% 2.5	% 8.4	% 2.0	% 4.0	% 6.8	% 6.5	% 5.7	% 7.0
17	4B1-09	% 6.2	% 0.3	% 0.4	% 1.2	% 0.5	% 3.9	% 0.3	% 0.2
18	1-30	% 7.9	% 3.9	% 0.7	% 1.7	% 1.5	% 2.2	% 0.6	% 2.0
19	1-25	% 0.0	% 1.8	% 2.8	% 0.7	% 1.0	% 0.7	% 0.7	% 1.1
20	1-15	% 7.1	% 1.8	% 3.9	% 5.4	% 2.2	% 2.8	% 2.7	% 3.6
21	2-30	% 13.6	% 4.7	% 1.0	% 1.3	% 1.0	% 1.7	% 0.1	% 2.9
22	2-25	% 17.8	% 5.2	% 3.7	% 6.2	% 4.3	% 4.6	% 3.0	% 5.6
23	2-15	% 4.5	% 2.9	% 2.9	% 1.5	% 3.3	% 8.6	% 5.9	% 3.2
24	5-30	% 6.6	% 1.9	% 0.3	% 2.4	% 0.7	% 2.4	% 0.7	% 1.7
25	5-20	% 5.5	% 1.5	% 4.6	% 2.6	% 2.3	% 3.5	% 1.8	% 4.6
26	5-10	% 72.2	% 5.9	% 0.7	% 15.5	% 3.0	% 2.2	% 4.5	% 5.1
27	4G1	% 2.2	% 0.3	% 0.0	% 0.1	% 0.1	% 0.1	% 0.0	% 0.3
28	4G3	% 5.3	% 1.4	% 1.7	% 1.8	% 1.5	% 2.0	% 1.6	% 1.4
29	4G4	% 0.2	% 0.4	% 0.2	% 0.5	% 0.3	% 0.1	% 0.2	% 0.5
30	2A2	% 1.8	% 1.3	% 0.7	% 1.1	% 0.4	% 1.3	% 1.3	% 1.3
Ortalama Hata		% 8.2	% 3.9	% 2.6	% 3.8	% 3.0	% 3.4	% 3.1	% 3.6
Standart Sapma		% 12.7	% 3.1	% 2.0	% 3.4	% 2.3	% 2.8	% 2.4	% 2.5

5.1.3. YSA modelinin test edilmesi ve sayısal sonuçlar

Eğitme işlemini takiben, YSA'nın test edilmesine geçilecektir. Daha önce ağı eğitilmesinde kullanılmayan giriş verilerinden oluşan test, veri seti kullanılarak YSA'dan makul sonuçlar üretmesi beklenecektir.

Test işlemi için kullanılacak olan giriş ve çıkış değerleri Çizelge 5.4'te verilmiştir. Yine karşılaştırma amacıyla gizli katmanında 1 den 8 e kadar düğümün bulunması halleri için 21 adet numune üzerinde test işlemi gerçekleştirilecektir.

Test seti numunelerine ait deneysel V_u değerleri ile farklı gizli düğüm sayılarına sahip modellerden elde edilen test sonuçları Çizelge 5.5'te verilmiştir. Yine Çizelge 5.6'da ise farklı gizli düğüme sahip YSA modelleri için hata yüzdeleri verilmiştir. Farklı gizli düğüm sayılarına sahip her bir YSA modeli için hata yüzdeleri ile ortalama hata ve hataların standart sapması da ayrıca verilmiş; maksimum hatalar, ortalama hataların minimumu ve standart sapmaların minimumu koyu karakterlerle gösterilmiştir.

Test numuneleri için deneysel sonuçlara en yakın sonuçlar, gizli katmanında 4 düğümü bulunan model, yani diğer bir deyişle 8:4:1 mimarisine sahip YSA modelinden elde edilmiştir.

Görülüyor ki, eğitim setleri için 8:3:1 mimarisindeki YSA daha uygun sonuçlar verirken, test setleri için 8:4:1 mimarisindeki YSA daha uygun sonuçlar vermiştir. Anlaşılan, 8:3:1 mimarisi eğitim setlerini, 8:4:1 mimarisi ise test setlerini daha iyi temsil edebilmektedir.

8:4:1 mimarisine ait test sonuçlarının incelenmesinden görülmektedir ki, ilk 12 numuneye ait hata yüzdeleri oldukça kabul edilebilir sınırlar içerisinde kalmasına rağmen, diğer numunelerde ise % 25'e varan hatalar kaydedilmiştir.

Hatanın azaltılması işlemi için iterasyon sayısını artırmak Şekil 5.4'te de görüldüğü gibi önemli bir performans artışı sağlamayacaktır. Eğitim veri setinin genişletilmesi veya eğitim çiftlerinin gruplandırılması (İng. clustering training pairs) düşünülebilir.

Ancak, bu çalışma uygulamaya yönelik olmadığından dolayı problemin daha fazla genişletilmesine gidilmeyecektir.

Çizelge 5.7’de deneysel verilerin temin edildiği kaynakta tartışılan ve analitik bir metot olan kafes kiriş modellemesinin (İng. strut-and-tie model) kullanılmasıyla kayma mukavemetinin tahmininden elde edilen sonuçlar ile problemin YSA modellemesinden elde edilen sonuçlar ve hata yüzdeleri karşılaştırma amacıyla verilmiştir. Ayrıca Şekil 5.5’te test veri seti için problemin kafes kiriş modellemesi ile YSA modellemesine ait hata yüzdeleri karşılaştırma amacıyla çubuk grafik olarak gösterilmiştir. Sonuçların incelenmesinden görülmektedir ki;

Eğitme seti olarak kullanılmış olan 30 adet numune için YSA modellemesinden elde edilen hata yüzdeleri ortalaması %2.65 ve hataların standart sapması %1.95 olarak bulunmuştur. Aynı numuneler için Strut-and-tie modelinden elde edilen hata yüzdeleri ortalaması %9.04 ve hataların standart sapması %6.19 olarak bulunmuştur. Test numunelerine için ise YSA modellemesinden elde edilen hata yüzdeleri ortalaması %7.54 ve hataların standart sapması %6.93 olarak bulunmuştur. Test numuneleri için strut-and-tie modelinden elde edilen hata yüzdeleri ortalaması %9.36 ve hataların standart sapması %6.51 olarak bulunmuştur. Buradan, YSA’nın eğitim seti için oldukça yüksek bir performans gösterdiği, test seti için ise ortalama hata baz alındığında strut-and-tie modelinden daha iyi olmakla birlikte biraz düşük bir performans gösterdiği anlaşılmaktadır.

Çizelge 5.4 Test işleminde kullanılan yüksek kiriş verileri (Siao 1993)

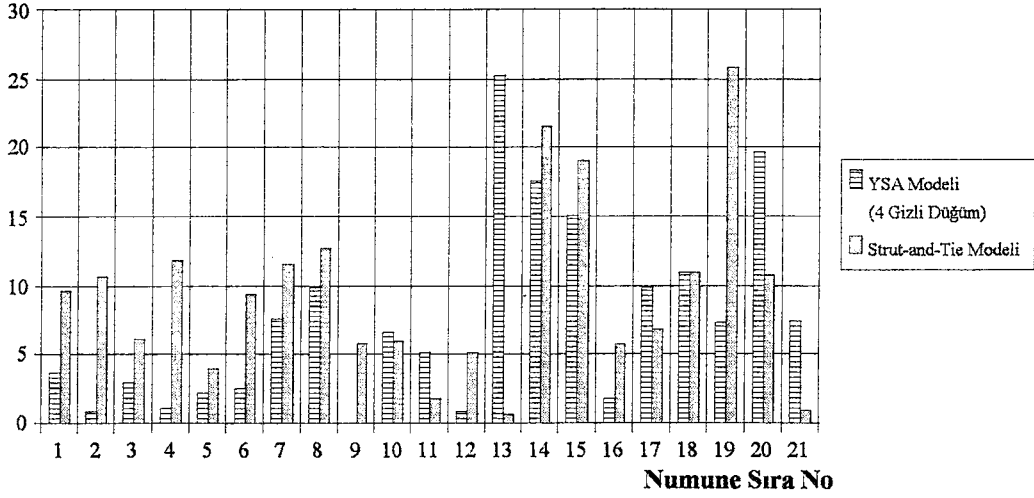
Sıra No	Numune No	ρ_h -	ρ_v -	f'_c (MPa)	a (mm)	h (mm)	d (mm)	b (mm)	a/h -	V_u kN
1	1A4-51	0.68	0.28	20.552	304.8	355.6	304.8	101.6	0.86	170.95
2	2A1-38	0.23	0.63	21.690	304.8	355.6	304.8	101.6	0.86	174.51
3	2A6-61	0.91	0.63	19.138	304.8	355.6	304.8	101.6	0.86	161.92
4	3A3-43	0.45	1.25	19.241	304.8	355.6	304.8	101.6	0.86	172.73
5	1B3-29	0.45	0.24	20.103	368.3	355.6	304.8	101.6	1.04	143.57
6	1B4-40	0.68	0.24	20.828	368.3	355.6	304.8	101.6	1.04	140.35
7	2B3-06	0.45	0.42	19.000	368.3	355.6	304.8	101.6	1.04	131.23
8	2B4-07	0.68	0.42	17.483	368.3	355.6	304.8	101.6	1.04	126.11
9	2B4-52	0.68	0.42	21.793	368.3	355.6	304.8	101.6	1.04	149.91
10	3B3-33	0.45	0.77	19.000	368.3	355.6	304.8	101.6	1.04	158.36
11	3B6-35	0.91	0.77	20.655	368.3	355.6	304.8	101.6	1.04	161.70
12	1-20	0.00	2.45	21.241	254.0	508.0	469.9	76.2	0.50	189.50
13	1-10	0.00	2.45	21.655	254.0	254.0	215.9	76.2	1.00	89.41
14	2-20	0.00	0.86	19.862	254.0	508.0	469.9	76.2	0.50	215.30
15	2-10	0.00	0.86	20.138	254.0	254.0	215.9	76.2	1.00	99.64
16	5-25	0.61	0.61	19.241	254.0	635.0	596.9	76.2	0.40	208.19
17	5-15	0.61	0.61	21.931	254.0	381.0	342.9	76.2	0.67	127.22
18	G33S-12	0.00	1.09	19.931	203.2	228.6	203.2	76.2	0.89	84.52
19	G33S-32	0.00	1.09	20.069	203.2	228.6	203.2	76.2	0.89	101.42
20	4G2	1.01	1.01	54.000	850.9	914.4	863.6	101.6	0.94	560.94
21	1C2	0.35	0.35	35.517	393.7	914.4	863.6	101.6	0.44	485.32

Çizelge 5.5 Test numunelerine ait deneysel sonuçlar ve farklı gizli düğüm sayılarına sahip YSA modellerine ait sonuçlar

Sıra No	Numune No	Deneysel V_u (kN)	Gizli Düğüm Sayılarına Bağlı YSA Sonuçları							
			1	2	3	4	5	6	7	8
1	1A4-51	170.95	163.20	163.17	167.45	164.61	165.47	162.81	164.13	163.94
2	2A1-38	174.51	155.65	163.23	172.34	173.03	170.19	168.85	166.95	168.77
3	2A6-61	161.92	170.99	172.78	165.61	166.65	171.48	169.51	171.87	170.82
4	3A3-43	172.73	162.31	163.37	173.36	174.67	167.77	165.06	166.98	167.98
5	1B3-29	143.57	145.41	139.49	142.51	140.44	138.22	141.43	138.97	139.22
6	1B4-40	140.35	149.95	142.06	149.68	143.92	142.75	146.23	145.25	142.07
7	2B3-06	131.23	145.33	139.84	142.94	141.29	138.83	140.25	139.32	139.84
8	2B4-07	126.11	147.90	136.32	139.57	138.55	134.62	134.30	136.69	135.14
9	2B4-52	149.91	151.80	145.57	157.04	149.85	148.66	152.64	150.96	147.52
10	3B3-33	158.36	146.85	143.06	150.54	147.94	144.70	143.93	144.90	145.22
11	3B6-35	161.70	157.70	153.10	158.21	153.47	155.97	161.55	158.68	153.22
12	1-20	189.50	179.58	198.21	192.71	187.96	192.77	192.50	192.45	192.82
13	1-10	89.41	132.83	112.65	102.30	111.96	113.12	116.49	114.84	110.62
14	2-20	215.30	163.14	182.03	183.90	177.43	182.05	184.35	185.11	180.49
15	2-10	99.64	129.68	96.50	71.05	84.63	83.82	100.09	91.90	86.92
16	5-25	208.19	214.58	212.62	212.78	211.85	210.66	205.64	206.25	214.71
17	5-15	127.22	155.62	132.54	140.99	139.77	137.53	128.33	130.78	139.84
18	G33S-12	84.52	131.71	101.24	77.95	93.81	93.18	106.05	97.77	94.57
19	G33S-32	101.42	131.75	101.23	78.29	94.04	93.36	106.23	97.94	94.75
20	4G2	560.94	638.04	741.78	724.92	671.37	732.79	728.26	705.29	736.71
21	1C2	485.32	521.69	386.15	420.04	449.61	419.48	433.04	433.86	396.84

Çizelge 5.6 Test numuneleri için farklı gizli düğüm sayılarına sahip YSA modellerine ait hata yüzdeleri

Sıra No	Numune No	Gizli Düğüm Sayısına Bağlı YSA Hataları							
		1	2	3	4	5	6	7	8
1	1A4-51	% 4.5	% 4.6	% 2.0	% 3.7	% 3.2	% 4.8	% 4.0	% 4.1
2	2A1-38	% 10.8	% 6.5	% 1.2	% 0.8	% 2.5	% 3.2	% 4.3	% 3.3
3	2A6-61	% 5.6	% 6.7	% 2.3	% 2.9	% 5.9	% 4.7	% 6.1	% 5.5
4	3A3-43	% 6.0	% 5.4	% 0.4	% 1.1	% 2.9	% 4.4	% 3.3	% 2.7
5	1B3-29	% 1.3	% 2.8	% 0.7	% 2.2	% 3.7	% 1.5	% 3.2	% 3.0
6	1B4-40	% 6.8	% 1.2	% 6.6	% 2.5	% 1.7	% 4.2	% 3.5	% 1.2
7	2B3-06	% 10.7	% 6.6	% 8.9	% 7.7	% 5.8	% 6.9	% 6.2	% 6.6
8	2B4-07	% 17.3	% 8.1	% 10.7	% 9.9	% 6.7	% 6.5	% 8.4	% 7.2
9	2B4-52	% 1.3	% 2.9	% 4.8	% 0.0	% 0.8	% 1.8	% 0.7	% 1.6
10	3B3-33	% 7.3	% 9.7	% 4.9	% 6.6	% 8.6	% 9.1	% 8.5	% 8.3
11	3B6-35	% 2.5	% 5.3	% 2.2	% 5.1	% 3.5	% 0.1	% 1.9	% 5.2
12	1-20	% 5.2	% 4.6	% 1.7	% 0.8	% 1.7	% 1.6	% 1.6	% 1.8
13	1-10	% 48.6	% 26.0	% 14.4	% 25.2	% 26.5	% 30.3	% 28.4	% 23.7
14	2-20	% 24.2	% 15.5	% 14.6	% 17.6	% 15.4	% 14.4	% 14.0	% 16.2
15	2-10	% 30.2	% 3.1	% 28.7	% 15.1	% 15.9	% 0.4	% 7.8	% 12.8
16	5-25	% 3.1	% 2.1	% 2.2	% 1.8	% 1.2	% 1.2	% 0.9	% 3.1
17	5-15	% 22.3	% 4.2	% 10.8	% 9.9	% 8.1	% 0.9	% 2.8	% 9.9
18	G33S-12	% 55.8	% 19.8	% 7.8	% 11.0	% 10.2	% 25.5	% 15.7	% 11.9
19	G33S-32	% 29.9	% 0.2	% 22.8	% 7.3	% 7.9	% 4.7	% 3.4	% 6.6
20	4G2	% 13.7	% 32.2	% 29.2	% 19.7	% 30.6	% 29.8	% 25.7	% 31.3
21	1C2	% 7.5	% 20.4	% 13.5	% 7.4	% 13.6	% 10.8	% 10.6	% 18.2
Ortalama Hata		% 15.0	% 8.9	% 9.1	% 7.5	% 8.4	% 7.9	% 7.7	% 8.8
Standart Sapma		% 15.3	% 8.7	% 8.8	% 6.9	% 8.1	% 9.4	% 7.6	% 7.9

% Hata

Şekil 5.5 YSA modeli ile Strut-and-Tie modellerinden elde edilen hata yüzdeleri (Test numuneleri için)

5.2. Karapınar Volkanik Agregasından İmal Edilen Hazır Döşeme Plaklarının Taşıma Gücünün YSA İle Belirlenmesi

Konya bölgesinde bol miktarda bulunan Karapınar Volkanik Agregası'ndan hazır döşeme plakları imal edilmesi konusu (Açıkel 1995) tarafından araştırılmış ve yazarın doktora tez konusunu oluşturmuştur. Yapılan çalışmada, Karapınar Volkanik Agregası kullanılarak imal edilen donatılı hafif betonarme plakların davranış ve taşıma gücünün incelenmesi için plak numuneler hazırlanmış ve bu numuneler üzerinde çeşitli deneyler yapılmıştır. Aşağıda, bu çalışmada deney numunesi plakların özellikleri ile deney düzeni hakkında kısaca bilgi verilecektir.

Bu tez çalışmasında ise, kompozit malzeme davranışının modellenmesine bir örnek olmak üzere bu plakların taşıma gücünün belirlenmesi için YSA kullanılacaktır. Böylece YSA'nın deneysel çalışmalar için etkinliği araştırılacaktır.

5.2.1. Deney numunesi plakların özellikleri

Plaklar eni 100cm, boyu 150, 250, 300, 350, 400cm; kalınlığı (h) 10, 12, 15cm olmak üzere muhtelif boyutlarda imal edilmiştir.

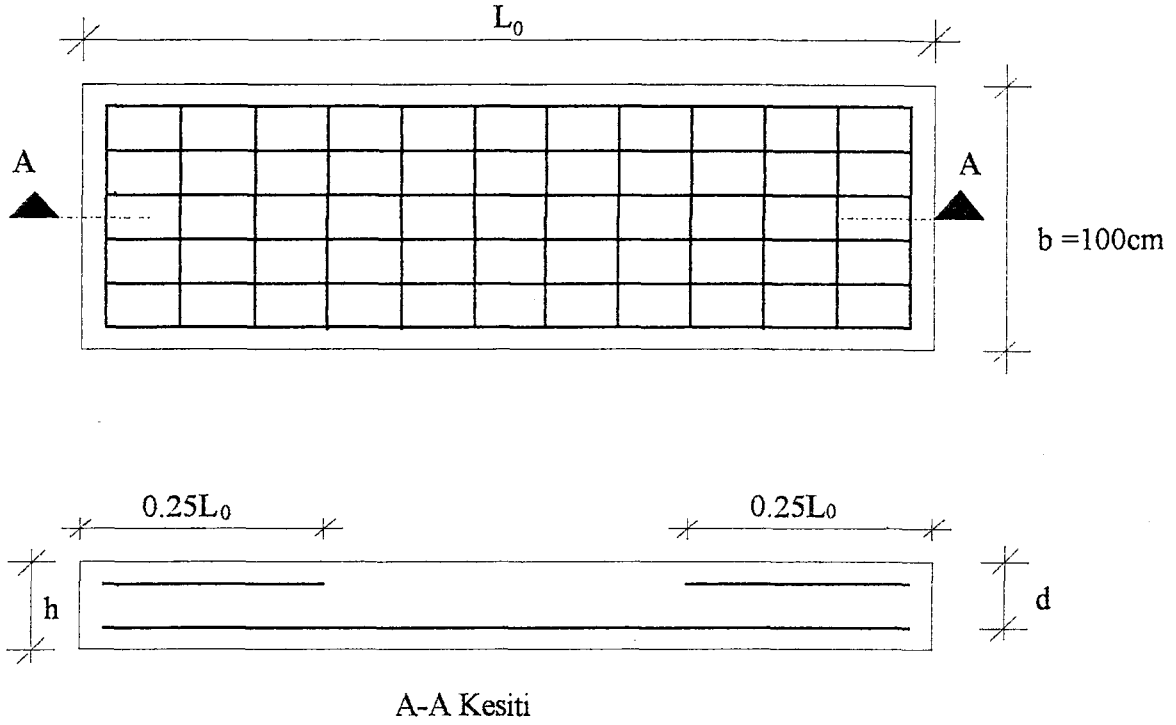
Beton karışımları için agrega granülometrisi uygun bir şekilde seçilerek sabit tutulmuştur. İki ayrı grup karışım, ilkinde 450 kg/m^3 dozajlı, $E/C=0.50$, $f_{ck,28}=200 \text{ kg/cm}^2$ olan 1730 kg/m^3 birim hacim ağırlığında olacak şekilde, ikincisinde ise 400 kg/m^3 dozajlı, $E/C=0.45$, $f_{ck,28}=168 \text{ kg/cm}^2$ olan 1650 kg/m^3 birim hacim ağırlığında olacak şekilde hazırlanmıştır.

Herbir plakta kullanılan beton ve çeliğin deneysel olarak bulunan karakteristik dayanımları (f_{ck} , f_{ctk} , f_{yk} , f_{su}); kullanılan donatı alanı (A_s), ve donatı oranı (ρ) vb. özellikler Çizelge 5.8 ve 5.9'da verilmiştir. Plak faydalı yükseklikleri (d), deney sonrası plak üzerinden ölçülerek alınmıştır.

Plaklara donatı Şekil 5.6'da görüldüğü gibi yerleştirilmiştir. Plakların sistemi basit kiriş olduğundan açıklık boyunca pozitif momente maruzdur. Ancak, elemanlar prefabrik olarak üretildiği için kaldırma esnasında üst donatıya ihtiyaç olacağından, plak elemanın heriki ucunda 0.25L mesafede üst donatı kullanılmıştır.

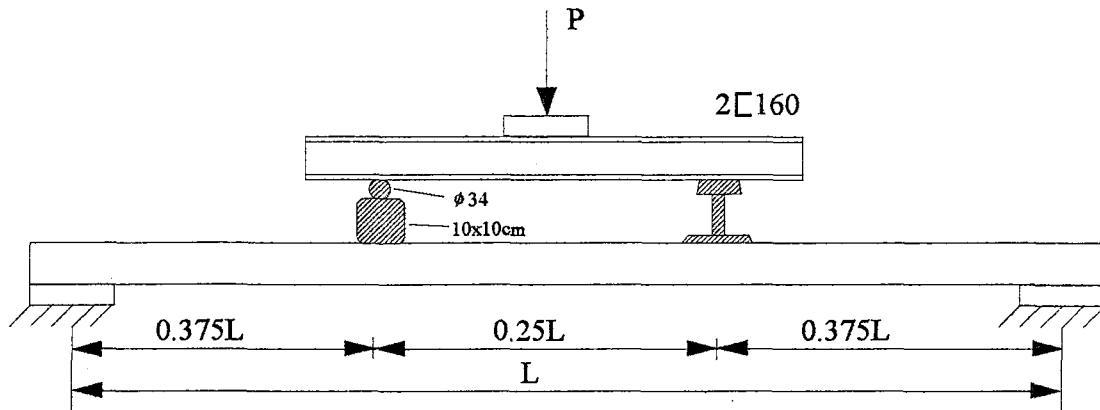
5.2.2. Deney düzeni

Açıkel (1995)'te üretilen plaklar S.Ü. Müh.-Mim.Fakültesi İnşaat Mühendisliği Bölümü Yapı Laboratuvarı'nda bulunan eğilme çerçevesinde deneye tabi tutulmuştur. Yükleme 20 ton kapasiteli deney aleti yardımıyla gerçekleştirilmiştir. Mesnet reaksiyonlarının plak enkesiti boyunca düzgün yayılışını sağlamak için çimento+alçı karışımıyla harç yapılmış, plaklar bunlar üzerine yerleştirilmiştir. Basit mesnetli kiriş davranışı için mesnetlerden biri sabit, diğeri ise bir adet $\phi 34$ çelik çubuk yardımıyla hareketli mesnet tarzında teşkil edilmiştir. Deney aletinin hidrolik presiyle test elemanına uygulanan P yükü, 0.25L_o mesafede simetrik olarak yerleştirilen ray



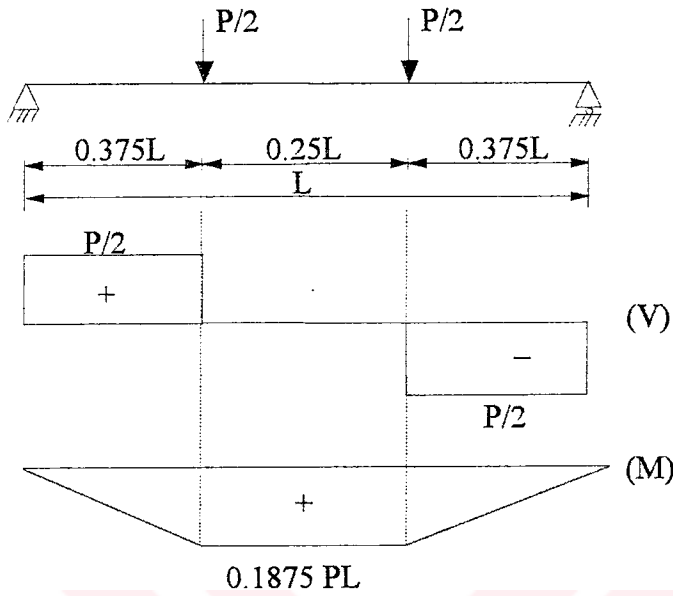
Şekil 5.6 Plaklarda donatının yerleştirilmesi

profili ve 10x10cm dolu çelik çubuk+ $\phi 34$ dairesel çelik çubuk üzerine mesnetlenen 2[160 profilinin, birbirine eşit mesnet reaksiyonları olarak verilmiştir. Plakların b genişliğince yerleştirilen bu ray profili, 10x10cm dolu çelik çubuk+ $\phi 34$ dairesel çelik çubuk sayesinde, yük test elemanlarına çizgisel olarak uygulanmıştır (Şekil 5.7).



Şekil 5.7 Plaklarda deney düzeni

Plaklara uygulanan P yükü, sıfırdan başlayarak artırılmış ve göçme meydana gelinceye kadar yüklemeye devam edilmiştir (Şekil 5.8).



Şekil 5.8 Plakların moment ve kesme kuvveti diyagramı

İki noktasından simetrik olarak yüklenmiş bir basit kirişte, eğilme momenti mesnetlerde sıfır olacağından, mesnetler ile yüklerin tesir ettiği noktalar arasında lineer bir artış göstererek, yükler arasındaki $0.25L$ lik bölgede ise sabit ve maksimum değerine ulaşacak ve $M_{\max}=0.1875PL$ olacaktır. Kesme kuvvetinin maksimum değeri ise $V_{\max}=P/2$ dir.

Çizelge 5.8 YSA'nın eğitilmesinde kullanılacak numunelerin özellikleri (Açıkel 1995)

Sıra No	Num. No	Ölçülen d	h	L _o	f _{ck}	f _{ctk}	f _{yk}	f _{su}	Donatı Alanı (A _s)	Donatı Oranı (ρ)
		cm	cm	cm	kg/cm ²	kg/cm ²	kg/cm ²	kg/cm ²	cm ²	A /b.d
1	2	8.00	10	250	202	16	6532	7185	1.58	0.001975
2	3	9.60	12	250	193	15	6706	7377	2.57	0.002677
3	4	9.50	12	250	195	15	6309	6940	3.77	0.003968
4	5	7.70	10	300	185	15	6954	7650	1.88	0.002442
5	6	10.00	12	300	210	16	6540	7194	3.52	0.003520
6	7	10.00	12	300	180	15	6954	7650	3.76	0.003760
7	8	10.00	12	350	185	15	6706	7377	2.57	0.002570
8	9	10.00	12	350	170	14	5893	6482	2.95	0.002950
9	10	12.60	15	350	190	15	6794	7473	4.42	0.003508
10	12	13.00	15	400	195	15	6794	7473	4.42	0.003400
11	13	12.00	15	400	188	15	5893	6482	5.90	0.004917
12	14	7.50	10	250	145	13	5624	6186	1.88	0.002507
13	15	10.00	12	250	135	13	6706	7377	2.57	0.002570
14	17	7.20	10	300	150	13	5624	6186	1.88	0.002611
15	19	10.00	12	300	165	14	5893	6482	2.95	0.002950
16	21	13.00	15	350	153	14	6954	7650	1.88	0.001446
17	23	10.00	12	400	162	14	6706	7377	2.57	0.002570
18	25	13.00	15	400	152	14	6309	6940	3.77	0.002900

Çizelge 5.9 YSA'nın test edilmesinde kullanılacak numunelerin özellikleri (Açıkel 1995)

Sıra No	Num. No	Ölçülen d	h	L_o	f_{ck}	f_{ctk}	f_{yk}	f_{su}	Donatı Alanı (A_s)	Donatı Oranı (ρ)
		cm	cm	cm	kg/cm ²	kg/cm ²	kg/cm ²	kg/cm ²	cm ²	A/b.d
1	3-II	9.83	12	250	185	15	6706	7377	2.57	0.002614
2	4-II	11.20	12	250	188	15	6309	6940	3.77	0.003366
3	6-II	10.00	12	300	210	16	6540	7194	3.52	0.003520
4	8-II	10.40	12	350	170	14	6706	7377	2.57	0.002471
5	15-II	9.85	12	250	139	13	6706	7377	2.57	0.002609
6	16	10.00	12	300	180	15	6954	6482	2.95	0.003760
7	18	10.00	12	250	165	14	5893	7377	2.57	0.002950
8	20	9.80	12	350	160	14	6706	7377	2.57	0.002622
9	20-II	10.10	12	350	186	15	6706	7377	2.57	0.002545
10	22	13.00	15	350	162	14	6309	6940	3.77	0.002900
11	24	10.00	12	400	165	14	6794	7473	2.21	0.002210

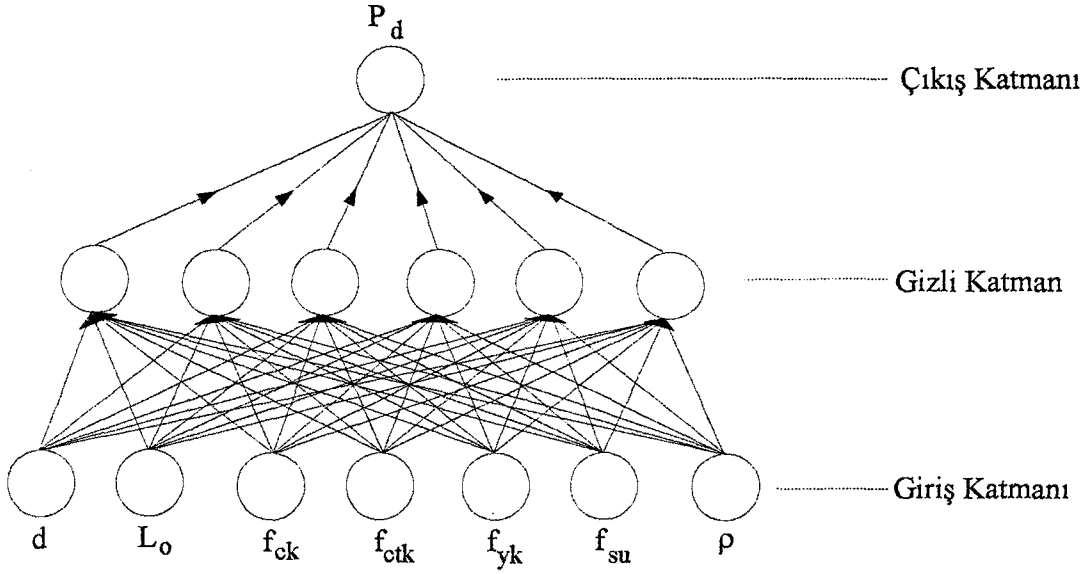
5.2.3. Problemin YSA modellemesi

Modelleme için öncelikle giriş ve çıkış katmanlarında hangi verilerin yeralacağını belirlenmesi gerekmektedir. Böylelikle, giriş ve çıkış katmanlarındaki hücre sayıları da belirlenmiş olacaktır. Problem, plakların taşıma gücünün belirlenmesi olduğuna göre, çıkış katmanında göçme yükü (P_d) nin yeralması kaçınılmaz olacaktır. Bunun yanında ilave bir çıkışa ihtiyacımız olmadığından, çıkış katmanı için bir adet düğüm yeterlidir. Giriş katmanındaki verilerin belirlenmesi ise oldukça önem taşımaktadır. Burada dikkat edilecek husus, giriş katmanında yeralan verilerin çıkış katmanındaki verilere direk/dolaylı etkisinin olmasıdır. Çıkış katmanını etkilemeyecek olan verilerin giriş

katmanında yerilmesi, YSA ile elde edilecek performansı olumsuz yönde etkileyecek ve sonuçların gerçek değerden oldukça uzaklaşmasına yolaçacaktır. Ayrıca, çıkış katmanını doğrudan etkileyecek olan bir verinin giriş katmanında yerilmemesi da yine aynı olumsuz etkinin oluşmasına yolaçacaktır. Plaktaki A_s donatı miktarına veya ρ donatı oranına ait herhangi bir bilginin giriş katmanında yerilmemesi böyle bir olumsuzluğa örnek olabilir. Giriş katmanında yerilmesine karar verilen veriler; plak faydalı yüksekliği d , plak genişliği L_o , plaktaki donatı oranı ρ , hazırlanan hafif beton karışımının karakteristik basınç dayanımı f_{ck} ve karakteristik çekme dayanımı f_{ctk} , kullanılan donatının basınçtaki akma dayanımı f_{yk} ve çekmedeki akma dayanımı f_{su} olarak belirlenmiştir.

Oluşturulacak YSA mimarisinde bir adet gizli katman yeterli görülmüştür. Böylece giriş katmanında 7 adet düğüm, çıkış katmanında ise 1 adet düğüm/işlem elemanı bulunacaktır. Gizli katmandaki düğüm sayısı için ise, giriş ve çıkış katmanındaki düğüm sayılarından büyük olanına yakın bir değer seçmenin iyi sonuçlar verdiği sanılmaktadır (Karlık 1994). Gizli katmanda fazla düğüm bulunması öğrenme süresini artıracığı gibi öğrenmeyi de sanıldığı gibi olumlu yönde etkilemeyebilecektir. Aynı şekilde gizli katman düğüm sayısının küçük seçilmesi de problemdeki doğrusal olmayan davranışların YSA tarafından hesaba katılmasını engelleyebilmektedir. Burada bahsedilen durumların daha iyi görülebilmesi için problem gizli katman düğüm sayısının 4, 5, 6, 7, 8 ve 9 olması halleri için çözülmüş ve sonuçlar çizelgeler ve grafikler halinde verilmiştir.

Toplam 29 adet deney numunesinden 18 tanesi YSA modellerinin eğitilmesi amacıyla kullanılacak, geriye kalan 11 tane numune ise modellerin test edilmesinde kullanılacaktır. YSA mimarileri belirlendikten sonra (Şekil 5.9), eğitme işlemi gerçekleştirilebilir.



Şekil 5.9 Problemin 7:6:1 düzenindeki YSA mimarisi

Eğitme ve test işlemleri sırasında verilerin ağı olduğu gibi yani hiçbir değişikliğe uğratılmadan verilemeyeceği unutulmamalıdır. Çünkü, eşik fonksiyonu olarak kullandığımız sigmoid fonksiyonu, ancak 0-1 arasındaki sayılar için işleyecektir. Bu yüzden verilerin normalize edilmesi, yani 0-1 arasındaki sayılara dönüştürülmesi gerekecektir. Her veri alabileceği maksimum değere veya maksimuma yakın bir değere bölünerek normalize edilmiş olur. Elbette, ağıdan elde edilecek sonuçlar için de aynı işlemin tersi söz konusu olacaktır. Yani, ağıdaki çıkış katmanındaki düğüme ait 0~1 arasında bir değere sahip sonucun da yine o veri bloğunun maksimum değeri ile çarpılarak eski haline getirilmesi gereklidir. Eğer bazı veriler ağı ikili düzende (İng. binary coding) girilmek istense idi, böyle bir dönüştürme işlemine o veriler için gerek kalmayacaktı (Şekil 5.10).

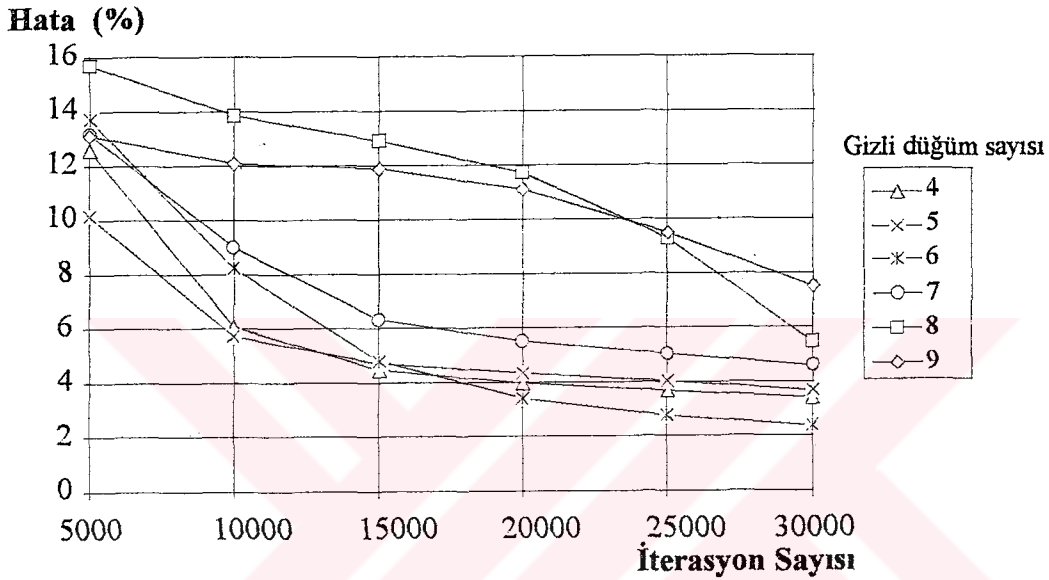
$h(\text{cm})$	Desen	İkili kodu
10		0 0
12		0 1
15		1 0

$L_o(\text{cm})$	Desen	İkili kodu
150		0 0 0
250		0 0 1
300		0 1 0
350		0 1 1
400		1 0 0

Şekil 5.10 Probleme ait bazı verilerin ikili düzende gösterilmesi

5.2.4. YSA modelinin eğitilmesi

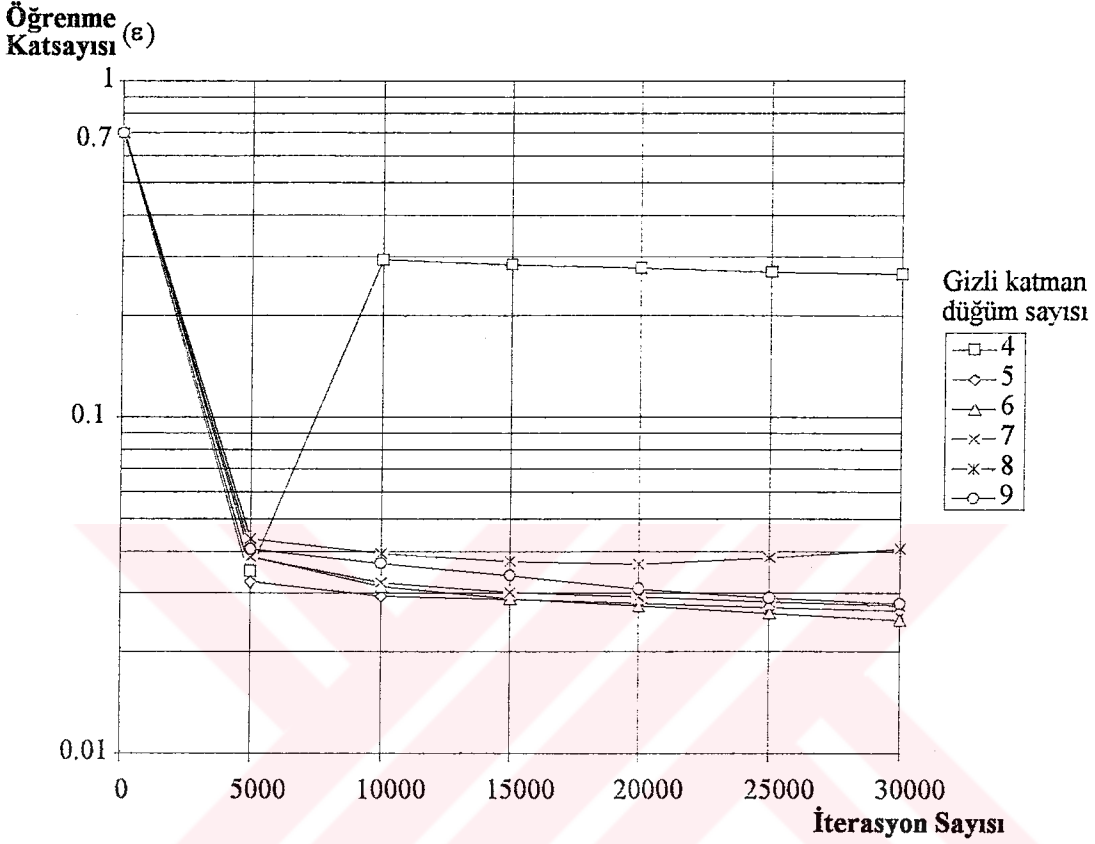
Eğitme işleminde 18 adet numuneye ait özellikler kullanılacaktır (Çizelge 5.8). Eğitim için 30000 iterasyon yapılmış ve her 5000 iterasyonda bir hata yüzdeleri kaydedilmiştir. Eğitim sırasında iterasyon sayısı ile hata yüzdesi arasındaki ilişki Şekil 5.11’de görülmektedir.



Şekil 5.11 İterasyon sayısı ile hata yüzdesi arasındaki ilişki

Görüldüğü gibi iterasyon sayısının artırılması hata yüzdesini gittikçe azaltmaktadır. Fakat bu azalma doğru orantılı olmamaktadır. İterasyon sayısını 30000 den daha büyük bir değere çıkarmak yalnız 8 ve 9 gizli düğümlü YSA modelleri için hata yüzdesinde önemli miktarda bir azalmaya sebep olacak, diğer modeller için ise dikkate değer bir azalma söz konusu olmayacaktır. 8 ve 9 gizli düğümlü YSA mimarilerinde eğitime süresi diğerlerine nazaran çok daha uzun olmasına rağmen, bu mimariler için 30000 iterasyon sonunda elde edilen hata yüzdesi en düşük hata yüzdesinin hemen hemen 3 katı olmaktadır. Bu nedenle bu problem için 30000 iterasyonun yeterli olduğu sonucuna varılmıştır.

Öğrenme momentum katsayısı $\alpha=0.9$ ve öğrenme katsayısının başlangıç değeri $\varepsilon=0.7$ alınmıştır. Öğrenme katsayısı (ε) bilgisayar programı içinde öğrenme hassasiyetini artırıcı yönde ayarlanmıştır (Şekil 5.12).



Şekil 5.12 Öğrenme katsayısının iterasyon sayısına bağlı değişimi

Çizelge 5.10'da eğitime işlemi için farklı gizli katman düğüm sayılarına sahip YSA modellerinden elde edilen sonuçlar ve hata yüzdeleri verilmiştir. Çizelgede ayrıca farklı düğüm sayısına sahip YSA modellerinin karşılaştırılması amacıyla, hata yüzdelerinin ortalama, maksimum, minimum ve standart sapma değerleri de verilmiştir. En düşük ortalama hata %4.31 değeri ile 6 gizli düğümlü modelde meydana gelmiştir. Genelde en düşük ortalama hatanın meydana geldiği model en uygun YSA modelidir, ortalama hataların eşitliği söz konusu olduğunda ise standart sapması küçük olan model uygun model olarak seçilebilir.

Çizelge 5.10 Eğitime numuneleri için YSA modellerinden elde edilen sonuçlar ve hata yüzdeleri

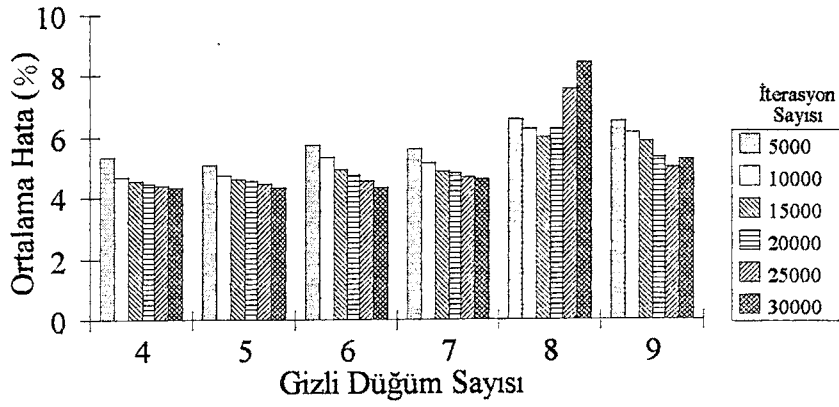
Sıra No	Numune No	Deneyisel P _a (kg)	Gizli Düğüm Sayısına Bağlı YSA Sonuçları (kg)						Gizli Düğüm Sayısına Bağlı YSA Hataları					
			4	5	6	7	8	9	4	5	6	7	8	9
1	2	1718	1551.2	1549.8	1548.0	1516.7	1232.8	1421.5	%10.75	%10.85	%10.98	%13.27	%39.36	%20.86
2	3	3322	3150.6	3120.2	3179.3	3129.3	3091.5	3106.7	%5.44	%6.47	%4.49	%6.16	%7.46	%6.93
3	4	4371	4108.4	4105.2	4091.0	4085.7	3912.2	4115.2	%6.39	%6.47	%6.84	%6.98	%11.73	%6.22
4	5	1565	1617.8	1625.8	1644.5	1599.5	1537.9	1590.1	%3.26	%3.74	%4.83	%2.16	%1.76	%1.58
5	6	3644	3702.1	3745.3	3745.0	3779.8	3706.2	3814.2	%1.57	%2.70	%2.70	%3.59	%1.68	%4.46
6	7	4222	4256.0	4231.8	4200.7	4242.1	4222.0	4181.9	%0.80	%0.23	%0.51	%0.47	%0.00	%0.96
7	8	2556	2569.5	2557.0	2566.9	2556.4	2652.8	2493.8	%0.53	%0.04	%0.42	%0.02	%3.65	%2.49
8	9	2572	2835.2	2818.6	2804.0	2830.9	2903.9	2812.9	%9.28	%8.75	%8.27	%9.15	%11.43	%8.56
9	10	4743	5026.8	5022.0	5007.6	4966.3	4763.3	4854.5	%5.65	%5.56	%5.28	%4.50	%0.43	%2.30
10	12	4885	4756.0	4760.5	4774.7	4791.3	4723.5	4772.3	%2.71	%2.62	%2.31	%1.96	%3.42	%2.36
11	13	4730	4710.0	4706.0	4728.7	4715.0	4872.1	4791.9	%0.42	%0.51	%0.03	%0.32	%2.92	%1.29
12	14	1573	1569.0	1586.3	1573.6	1583.7	1390.6	1542.5	%0.25	%0.84	%0.04	%0.68	%13.12	%1.98
13	15	3200	3218.9	3222.4	3257.1	3234.7	3337.5	3244.2	%0.59	%0.70	%1.75	%1.07	%4.12	%1.36
14	17	1428	1325.2	1345.6	1305.0	1318.5	1145.6	1338.5	%7.76	%6.12	%9.43	%8.30	%24.65	%6.69
15	19	2986	3215.6	3198.6	3193.5	3187.8	3164.1	3153.3	%7.14	%6.65	%6.50	%6.33	%5.63	%5.31
16	21	2610	2702.0	2709.6	2673.4	2736.0	2760.6	2822.4	%3.40	%3.68	%2.37	%4.61	%5.46	%7.53
17	23	2262	2183.1	2174.5	2190.1	2180.3	2290.4	2156.2	%3.61	%4.02	%3.28	%3.75	%1.24	%4.91
18	25	4848	4458.4	4467.7	4504.4	4429.2	4258.2	4448.4	%8.74	%8.51	%7.63	%9.46	%13.85	%8.98
			Ortalama:						%4.35	%4.36	%4.31	%4.60	%8.44	%5.27
			Minimum:						%0.25	%0.04	%0.03	%0.02	%0.00	%0.96
			Maksimum:						%10.75	%10.85	%10.98	%13.27	%39.36	%20.86
			Standart Sapma:						%3.41	%3.26	%3.38	%3.78	%9.95	%4.72

Çizelge 5.10'un incelenmesinden görülmektedir ki, YSA modellerinin maksimum hata yüzdeleri 2 nolu numune için bulunmuştur. YSA'nın 2 nolu numuneye ait giriş-çıkış ilişkisini daha iyi öğrenebilmesi için eğitim setinin genişletilmesi uygun olacaktır. Uygun model olarak seçilen 7:6:1 mimarisine sahip YSA modeli için her 5000 iterasyon sonunda elde edilen sonuçlar ve hata yüzdeleri Çizelge 5.11'de verilmiştir. Şekil 5.13'te iterasyon sayılarına ve farklı gizli katman düğüm sayılarına bağlı olarak ortalama, maksimum ve minimum hataların değişimi görülmektedir. Burada, 5 ve 6 gizli düğüm sayısına sahip YSA modellerinin en uygun sonuçları verdiği açıkça görülmektedir.

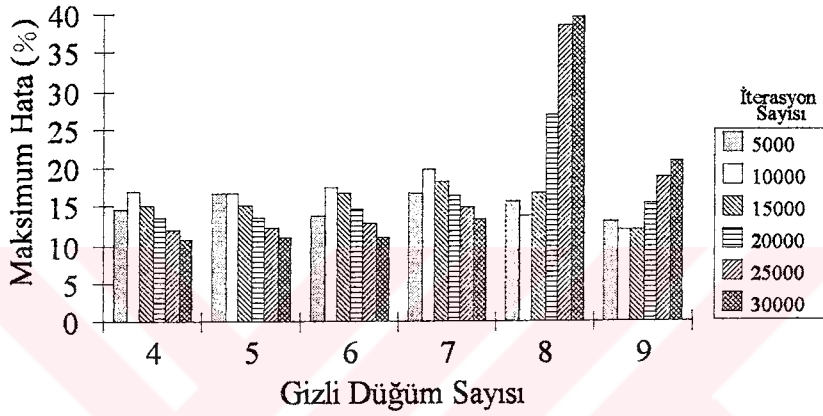
Uygun model olarak seçilen YSA modeli, her zaman, bütün numuneler için en uygun sonucu vermeyebilir. Bunun nedeni, bu çalışmada hatanın geriye yayılması algoritmasıyla YSA'nın hata minimizasyonu yapmasından kaynaklanmaktadır. Bilindiği gibi, bu algorithmada amaç, ortalama hataların karesinin minimum yapılmasıdır. Şekil 5.14'te ise farklı gizli katman düğüm sayısına sahip YSA modellerinin, numuneler üzerindeki bireysel performansları görülmektedir.

Çizelge 5.11 Eğitim numuneleri için, gizli katman düğüm sayısı 6 olan YSA modeline ait sonuçlar

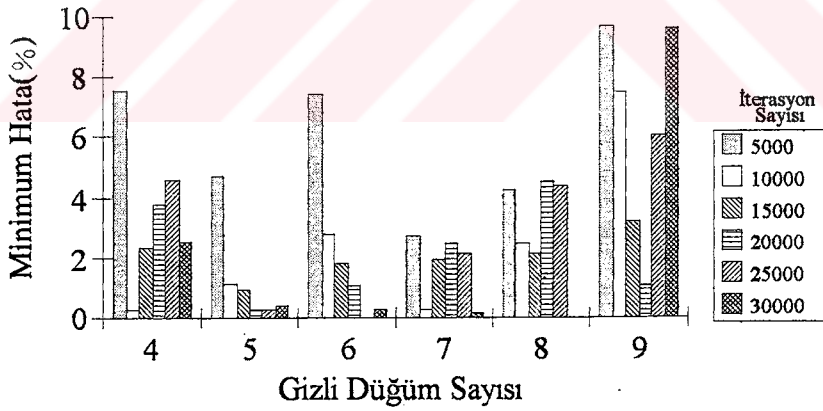
Sıra No	Num No	Deneyisel Pd (kg)	Belirli iterasyon aralıkları için YSA sonuçları (kg)						Belirli iterasyon aralıkları için hata yüzdeleri					
			5000	10000	15000	20000	25000	30000	5000	10000	15000	20000	25000	30000
1	2	1718	1536.4	1462.9	1472.7	1499.6	1525.0	1548.0	%11.8	%17.4	%16.7	%14.6	%12.7	%11.0
2	3	3322	3091.7	3135.9	3162.0	3172.6	3176.9	3179.3	%7.4	%5.9	%5.1	%4.7	%4.6	%4.5
3	4	4371	4172.8	4139.5	4094.1	4080.9	4083.2	4091.0	%4.7	%5.6	%6.8	%7.1	%7.0	%6.8
4	5	1565	1668.4	1617.8	1617.1	1625.9	1635.0	1644.5	%6.2	%3.3	%3.2	%3.7	%4.3	%4.8
5	6	3644	3764.6	3805.7	3792.0	3772.7	3757.2	3745.0	%3.2	%4.2	%3.9	%3.4	%3.0	%2.7
6	7	4222	4102.1	4142.4	4158.3	4171.3	4185.9	4200.7	%2.9	%1.9	%1.5	%1.2	%0.9	%0.5
7	8	2556	2537.2	2548.9	2560.5	2566.3	2568.1	2566.9	%0.7	%0.3	%0.2	%0.4	%0.5	%0.4
8	9	2572	2739.5	2807.6	2822.3	2819.1	2812.1	2804.0	%6.1	%8.4	%8.9	%8.8	%8.5	%8.3
9	10	4743	4838.4	4933.8	4985.3	5001.3	5006.4	5007.6	%2.0	%3.9	%4.9	%5.2	%5.3	%5.3
10	12	4885	4665.5	4769.5	4812.0	4808.6	4792.5	4774.7	%4.7	%2.4	%1.5	%1.6	%1.9	%2.3
11	13	4730	5116.3	4864.5	4758.6	4735.1	4730.1	4728.7	%7.6	%2.8	%0.6	%0.1	%0.0	%0.0
12	14	1573	1637.7	1610.6	1599.6	1591.6	1582.4	1573.6	%4.0	%2.3	%1.7	%1.2	%0.6	%0.0
13	15	3200	3236.4	3263.1	3274.2	3273.6	3266.6	3257.1	%1.1	%1.9	%2.3	%2.2	%2.0	%1.8
14	17	1428	1361.2	1339.0	1327.2	1318.6	1311.1	1305.0	%4.9	%6.6	%7.6	%8.3	%8.9	%9.4
15	19	2986	3063.8	3145.0	3176.0	3187.5	3192.3	3193.5	%2.5	%5.1	%6.0	%6.3	%6.5	%6.5
16	21	2610	3025.7	2844.6	2741.7	2702.0	2684.2	2673.4	%13.7	%8.2	%4.8	%3.4	%2.8	%2.4
17	23	2262	2240.0	2218.3	2203.5	2194.8	2191.2	2190.1	%1.0	%2.0	%2.7	%3.1	%3.2	%3.3
18	25	4848	4109.8	4272.6	4369.4	4421.6	4463.5	4504.4	%18.0	%13.5	%11.0	%9.6	%8.6	%7.6
Ortalama Hata:									%5.7	%5.3	%4.9	%4.7	%4.5	%4.3
Min. Hata:									%0.7	%0.3	%0.2	%0.1	%0.0	%0.0
Maks. Hata:									%13.7	%17.4	%16.7	%14.6	%12.7	%11.0



(a)

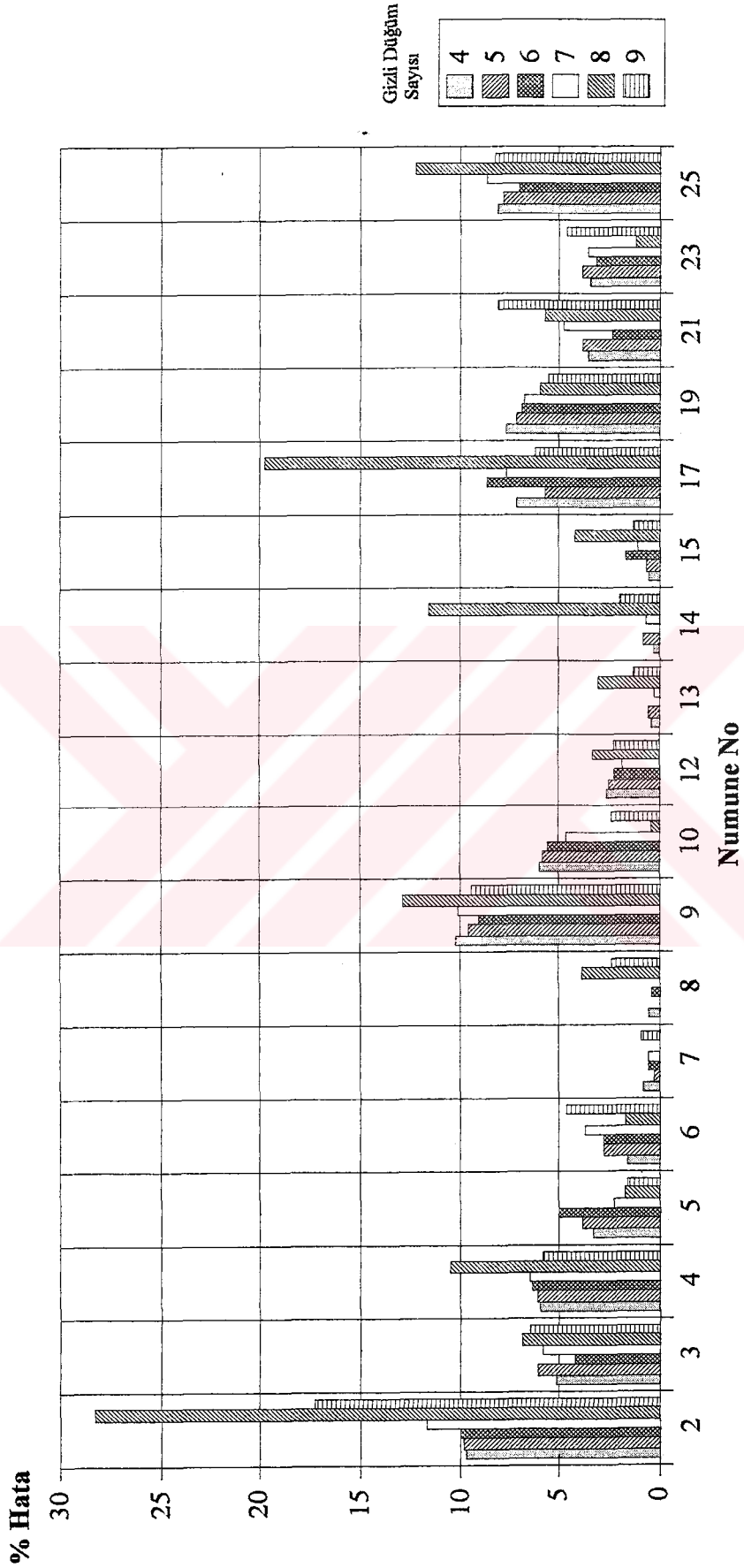


(b)



(c)

Şekil 5.13 Farklı gizli düğüm sayıları için iterasyon sayısına bağlı (a) Ortalama (b) Maksimum (c) Minimum hata grafikleri

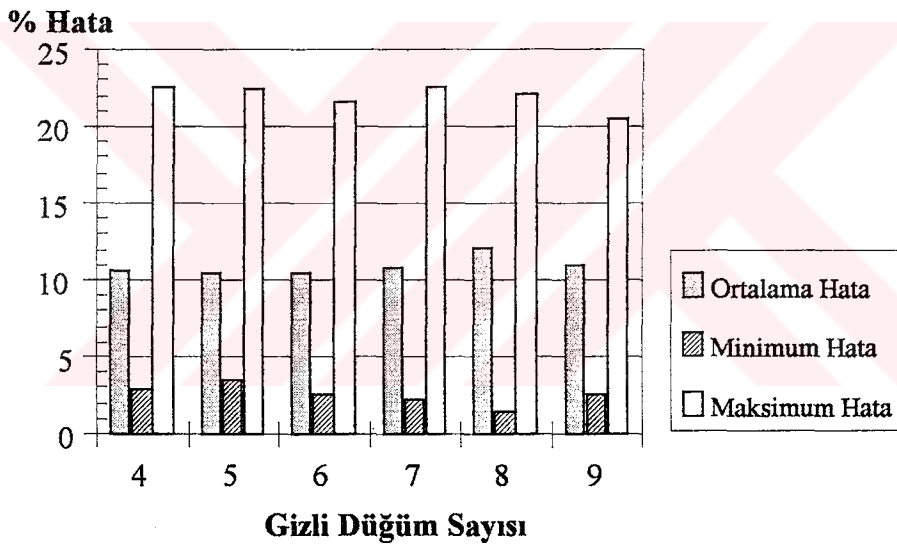


Şekil 5.14 YSA modellerinin eğitim seti için bireysel performansları

5.2.5. YSA modelinin test edilmesi ve sayısal sonuçlar

Eğitme işlemi tamamlandıktan sonra artık YSA'nın test edilmesine geçilebilir. YSA'nın öğrendiği giriş-çıkış verileri ilişkisini genelleme yeteneğinin sergileneceği bu safhanın başarısı, büyük ölçüde eğitime sırasında elde edilen başarıya ve eğitime numunelerinin oluşturduğu örnek kümesinin, giriş-çıkış verileri popülasyonunu iyi bir şekilde temsil edebilmesine bağlıdır.

Test işleminde 11 adet numune kullanılmıştır. Çizelge 5.9'da bu numunelerin özellikleri verilmiştir. Test işlemi sonunda farklı gizli katman düğüm sayıları için elde edilen ortalama, minimum ve maksimum hata yüzdelerinin dağılımı Şekil 5.15'te verilmektedir.



Şekil 5.15 Test seti için farklı gizli düğüm sayılarının hata yüzdelerine etkisi

Test numunelerine ait deneysel Pd yükleri ile farklı mimarilere sahip YSA'lardan elde edilen sonuçlar karşılaştırma amacıyla Çizelge 5.12'de verilmiştir. Çizelge 5.13'te ise test işlemine ait YSA sonuçlarıyla deneysel sonuçlar arasındaki farklardan kaynaklanan hata yüzdeleri verilmiştir. Bu hata yüzdelerine ait ortalama, maksimum, minimum ve standart sapma değerleri de ayrıca gösterilmiştir. Çizelgelerin incelenmesinden test işleminde en yüksek performansı gösteren YSA mimarilerinin,

eğitme sırasında da en yüksek performansa sahip olan 5 ve 6 gizli katman düğüm sayısına sahip modeller oldukları anlaşılmaktadır. Heriki YSA mimarisi için de ortalama hata %9.63 olarak bulunmuştur. 6 gizli düğümü olan YSA modelinin standart sapması, 5 gizli düğümü olan modelden daha azdır. Yani, 6 gizli düğümlü YSA modeline ait değerler ortalamaya daha yakındır. Görüldüğü gibi, minimum hata 8 gizli düğümlü YSA

Çizelge 5.12 Test numuneleri için YSA modellerinden elde edilen sonuçlar

Sıra No	Numune No	Deneysel P_d (kg)	Gizli Düğüm Sayısına Bağlı YSA Sonuçları (kg)					
			4	5	6	7	8	9
1	3-II	3342	3249.47	3226.93	3257.12	3210.81	3158.27	3139.17
2	4-II	4902	4724.79	4725.52	4679.45	4583.49	4245.90	4462.82
3	6-II	4416	3702.18	3745.33	3745.03	3779.81	3706.27	3814.22
4	8-II	2432	2611.99	2576.32	2628.95	2611.77	2783.14	2605.06
5	15-II	3492	3176.10	3166.52	3213.03	3187.12	3310.82	3206.87
6	16	3944	4114.24	4093.33	4086.13	4038.28	3884.79	4088.18
7	18	2850	3680.95	3673.18	3638.57	3684.27	3654.80	3582.72
8	20	2372	2476.96	2464.39	2495.23	2473.17	2605.25	2436.41
9	20-II	3002	2592.61	2577.63	2590.48	2580.18	2680.74	2518.83
10	22	5134	4793.07	4786.05	4803.74	4720.17	4481.33	4727.65
11	24	2107	1747.20	1738.44	1749.26	1762.09	1824.62	1763.14

Çizelge 5.13 Test numuneleri için YSA modellerinden elde edilen hata yüzdeleri

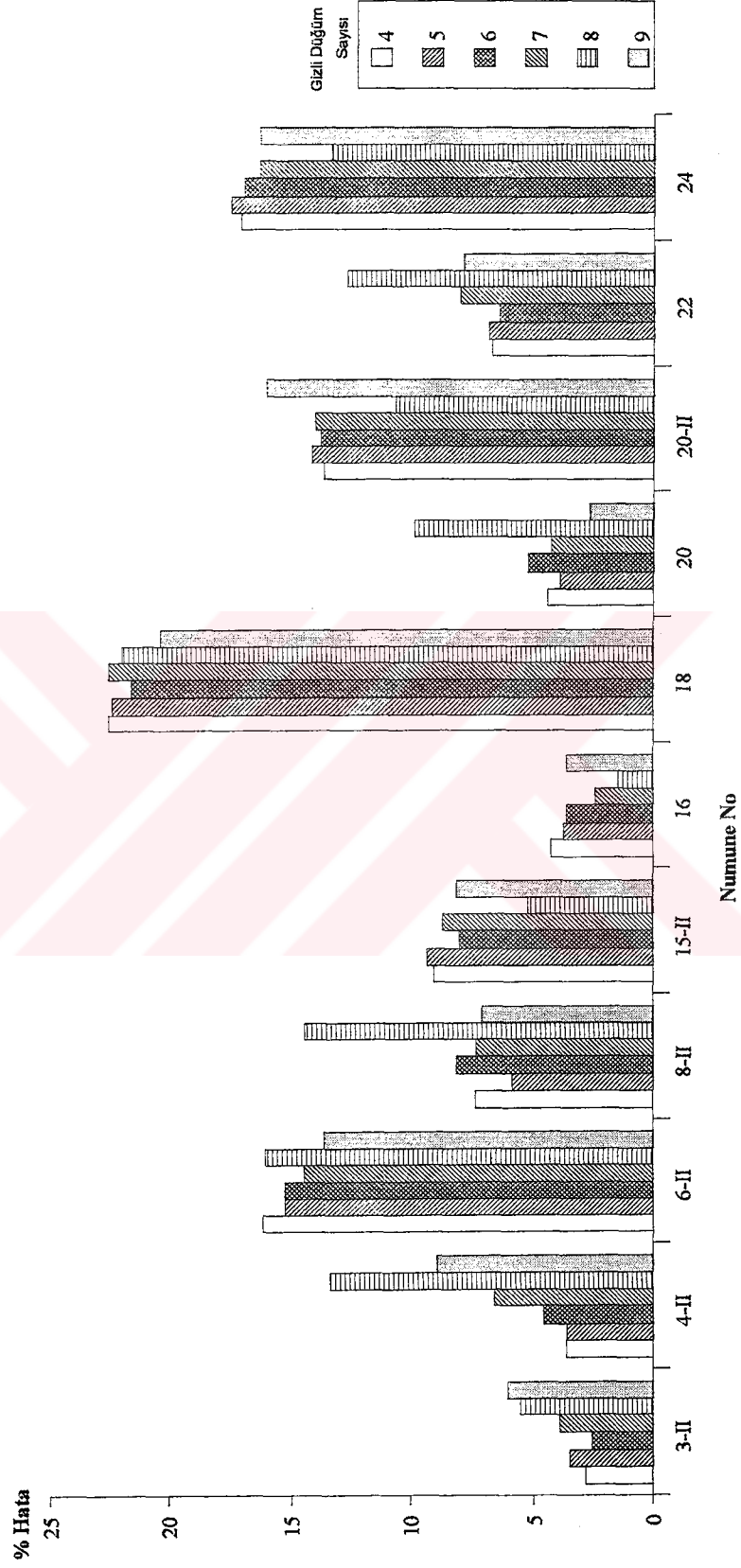
Sıra No	Numune No	Gizli Düğüm Sayısına Bağlı Hata Yüzdeleri					
		4	5	6	7	8	9
1	3-II	%2.77	%3.44	%2.54	%3.93	%5.50	%6.07
2	4-II	%3.62	%3.60	%4.54	%6.50	%13.38	%8.96
3	6-II	%16.16	%15.19	%15.19	%14.41	%16.07	%13.63
4	8-II	%7.40	%5.93	%8.10	%7.39	%14.44	%7.12
5	15-II	%9.05	%9.32	%7.99	%8.73	%5.19	%8.17
6	16	%4.32	%3.79	%3.60	%2.39	%1.50	%3.66
7	18	%22.57	%22.41	%21.67	%22.64	%22.02	%20.45
8	20	%4.42	%3.90	%5.20	%4.27	%9.83	%2.72
9	20-II	%13.64	%14.14	%13.71	%14.05	%10.70	%16.09
10	22	%6.64	%6.78	%6.43	%8.06	%12.71	%7.91
11	24	%17.08	%17.49	%16.98	%16.37	%13.40	%16.32
Ortalama:		%9.79	%9.63	%9.63	%9.88	%11.34	%10.10
Minimum:		%2.77	%3.44	%2.54	%2.39	%1.50	%2.72
Maksimum:		%22.57	%22.41	%21.67	%22.64	%22.02	%20.45
Standart Sapma:		%6.59	%6.63	%6.28	%6.23	%5.73	%5.70

modelinde 16 nolu numune için %1.5 olarak bulunmuştur. Ancak daha önce de belirtildiği gibi, bu bireysel bir performanstır ve uygunluk değerlendirmesi yaparken tek başına bağlayıcı bir unsur olamaz.

Herbir test numunesi için farklı mimarilere sahip modellerden elde edilen bireysel performanslar Şekil 5.16'da gösterilmiştir.

YSA'nın plakların taşıma gücünün bulunması probleminde elde ettiği başarı küçümsenemeyecek derecededir. Çünkü YSA modellemesi için, daha önce plağın davranışı için çıkarılmış olan herhangi bir matematiksel formül kullanılmamıştır. Test sırasında elde ettiği başarı tamamıyla YSA'nın öğrenme yeteneğinden kaynaklanmaktadır.





Şekil 5.16 Farklı gizli düğüm sayılarının test numuneleri üzerindeki bireysel performansları

6. SONUÇ

Bu tez çalışmasında, önce, yapay zekanın bir konusu olan yapay sinir ağları (YSA) nın tanımı yapılarak; temel birimleri, yapısı, önemli karakteristikleri, avantaj ve dezavantajları ile sınırlamaları ele alınmıştır. YSA ile yapılması düşünülen tasarımlar için işlem basamakları sunulmuştur.

Daha sonra YSA nın, yapı analizi ve tasarımındaki uygulamalarından bahsedilerek, yapay zeka uygulamaları arasında YSA dan daha eski bir geçmişe sahip bulunan uzman sistemler ile YSA lar karşılaştırmalı bir şekilde ele alınmış, birbirlerine göre üstünlükleri ve dezavantajları ortaya konulmuştur.

Bu çalışmada, kullanımının ve uygulamalarının yaygın olması sebebiyle geriye yayımlı çok katmanlı YSA modeli esas alınmış ve işlemler bu model üzerinde gerçekleştirilmiştir.

YSA nın yapı analizi ve tasarımındaki uygulamalarına örnek olmak üzere, iki ayrı problem çözülerek metodun etkinliği araştırılmıştır. Bu sayısal örneklerden birincisi, betonarme yüksek kirişlerin kesme mukavemetinin; ikincisi ise hafif betondan imal edilen plakların taşıma gücünün belirlenmesi problemidir. Turbo PASCAL programlama dilinde geliştirilen bir bilgisayar programı yardımıyla hazırlanan YSA modeli ile problemler çözülmüş ve elde edilen sayısal sonuçlar irdelenmiştir. Deneysel sonuçlar ile çeşitli YSA modellerinden elde edilen sonuçlar karşılaştırılarak, YSA nın giriş verileri ile çıkış verileri arasında gerçekten oldukça etkin bir ilişki kurabildiği görülmüştür. YSA ya, daha önce eğitime sırasında kullanılmayan değerler uygulandığında da, YSA bu etkinliğini göstermekte; ayrıca, girişler için çıkışların üretilmesi işlemini anında gerçekleştirmektedir.

Kompozit bir malzeme olan betonarmenin davranışının YSA ile modellenmesine yönelik çözülen iki sayısal örnekten elde edilen sonuçlar oldukça ümit verici bulunmuştur. Malzeme davranışının YSA ile modellenmesinin birçok kolaylıkları vardır. Böyle bir modelleme malzeme davranışı hakkında bir varsayım yapmaz,

davranışın tahmini için eğitim sırasında kullanılan deneysel veriyi kullanır. Ayrıca, sonradan elde edilen yeni deneysel verilerin eğitim setine ilâve edilmesi ile eğitim işlemi tekrarlanarak YSA'nın güncelleştirilmesi de mümkündür. Böylece YSA, malzeme davranışına ait bilgiler arttıkça malzeme modelinin gelişebildiği esnek bir model olma özelliği göstermektedir. Ancak YSA'nın diğer matematiksel modellerden bir farkı da, modelin adım adım uygulanan sembolik ifadelerden oluşmamasıdır. Model bilgisi YSA'nın bağ ağırlıklarında yerleşik olarak bulunur.

YSA modellemesi için geliştirilen herhangi bir bilgisayar programı, problem çeşidine göre değişiklik arz etmemektedir. Sadece program parametrelerinin eldeki probleme göre değiştirilmesi yeterli olmaktadır. Bu da YSA uygulamalarının geliştirilmesini oldukça kolaylaştırmaktadır.

YSA'yı diğer matematiksel modellerden ayıran en önemli özelliklerden birisi de, verilerdeki eksiklik veya belirli düzeylerdeki hatalardan dolayı, sistemin durma konumuna gelmemesidir. YSA; bozuk, eksik veya önceden görülmemiş verilere tamamlama veya ara değer bulma şeklinde sonuçlar da üretebilir.

YSA'nın her türlü hesaplama sahasında kullanımının uygun olduğu söylemek de yanlış olur. Ancak, giriş-çıkış ilişkilerinin tamamıyla açıklanamadığı bir çok problem için etkin bir çözüm yöntemi olduğu söylenebilir. YSA'nın birçok alanda kullanımının hızla artmasına rağmen, şunu da unutmamak gerekir ki, YSA her probleme çözüm getirebilen bir metot değildir ve kullanımı dikkatle tahkik edilmelidir. YSA ile elde edilen çözümlerin nasıl ve hangi temellere dayanarak elde edildiği konusunda açıklama yapılamamaktadır (Moselhi 1996).

Genelde, çözümü karmaşık algoritmik metotlara ihtiyaç gösteren veya algoritmik metotlarla çözülemeyen problemlerde; çok sayıda kalitatif ve kantitatif verinin anında tahminin/belirlenmesinin gerekli olduğu problemlerde; spesifik bir dalda tecrübeye dayanan gerçekçi çözümlere ihtiyaç duyulan problemlerde; çözüm için gerekli olan yeterli kesin bilgiyi elde etmenin mümkün olmadığı ve problemin önceden bilinen kesin verilerle karakterize edilebildiği yerlerde YSA kullanımı önerilmektedir.

7.EK-1. YSA BİLGİSAYAR PROGRAMI

PROGRAM Yapay_Sinir_Agi;
 { \$N+, E+ }
 Uses crt;

{Bu program "Turbo Pascal version 7.0" programlama dilinde geliştirilmiş ve bir "PC/AT" bilgisayarda "MS-DOS" ortamında çalıştırılmıştır. Program; Çok Katmalı, İleri Yayılımlı Yapay Sinir Ağı (İng. Multi Layer, Feed Forward Artificial Neural Network) bilgisayar modellemesini gerçekleştirmektedir. Giriş, Çıkış ve Gizli katman düğüm sayıları hafıza ile sınırlıdır. Eğitim seti sayısı ise hafıza ile sınırlandırılmak istenmediğinden, işlemlerde hafıza yerine, daha yavaş olmasına rağmen disk kullanılacaktır. Öğrenme katsayısı epsilon (ϵ) 0.7 değeri ile başlatılıp program içinde hataya göre ayarlanacaktır. Momentum katsayısı alfa (α) ise 0.9 değerinde sabit tutulmuştur.}

CONST

M_K = 3; { Maksimum katman sayısı }
 M_Dg_S = 6; { Herbir katmandaki maksimum düğüm sayısı }
 E_Seti_S = 42; { Eğitim seti sayısı/ Eğitim sırasında kullanılacak örnek grubu sayısı }
 G_K_Dg_S = 6; { Giriş katmanı düğüm sayısı }
 C_K_Dg_S = 1; { Çıkış katmanı düğüm sayısı }
 Gz_K_Dg_S = 6; { Gizli katman düğümleri sayısı }
 K_D_S: *array* [0..M_K-1] of integer = (G_K_Dg_S, Gz_K_Dg_S, C_K_Dg_S); { Katman düğüm sayıları }
 G_bolenleri: *Array*[0..G_K_Dg_S-1] of double = (1, 2.5, 25, 750, 105, 1.3);
 C_bolenleri: *Array*[0..C_K_Dg_S-1] of double = (250);
 İlk_Epsilon = 0.7; { Öğrenme katsayısının başlangıç değeri }
 Alfa = 0.9; { Momentum katsayısı }
 M_iter_S = 10; { Maksimum iterasyon/döngü sayısı }
 C_araligi = 5; { Çıktı alma aralığı }

TYPE

Formatli_Veri =Record
 G: *Array*[0..G_K_Dg_S-1] of double;
 C: *Array*[0..C_K_Dg_S-1] of double;
 End;
 T_Formatli_Veri =Record
 G: *Array*[0..G_K_Dg_S-1] of double;
 End;

VAR

i, ii, j, cevap: *integer*;
 Epsilon: *double*;
 Gecici_matris: *array* [0..*Trunc*(M_iter_S/C_araligi), 0..*E_Seti_S*-1,
 0..*(G_K_Dg_S+C_K_Dg_S-1)*] of *double*;
 G: *array* [0..*M_K*-1, 0..*M_Dg_S*-1] of *double*;
 Dt_hata: *array* [0..*M_K*-1, 0..*M_Dg_S*-1] of *double*;
 Teta: *array* [0..*M_K*-1, 0..*M_Dg_S*-1] of *double*;
 Dt_teta: *array* [0..*M_K*-1, 0..*M_Dg_S*-1] of *double*;
 Ag: *array* [0..*M_K*-1, 0..*M_Dg_S*-1, 0..*M_Dg_S*-1] of *double*;
 Dt_Ag: *array* [0..*M_K*-1, 0..*M_Dg_S*-1, 0..*M_Dg_S*-1] of *double*;
 Hedef_Deger: *array* [0..*E_Seti_S*-1, 0..*C_K_Dg_S*-1] of *double*;
 Veri: *Formatli_Veri*;
 T_veri: *T_Formatli_Veri*;
 G_Dosya, C_dosya, T_Dosya: *text*;
 Ag_Dosya: *File of double*;
 Formatli_G_Dosya: *File of Formatli_Veri*;
 Formatli_T_Dosya: *File of T_Formatli_Veri*;

Procedure Baslangic_Degerleri;

{ Programın başlangıcındaki ağırlıklar }

var

K_no, Dg_no, Onceki_k_dg_no: *integer*;

Begin

{ Rastgele (İng. *Random*) değerler }

For K_no:= 1 *to* M_K-1 *do*

Begin

For Dg_no:= 0 *to* K_D_S[K_no]-1 *do*

Begin

Teta[K_no, Dg_no]:= *Random*(30)/100;

Dt_teta[K_no, Dg_no]:= 0;

End;

End;

{ Rastgele ağırlıklar }

For K_no:= 1 *to* M_K-1 *do*

Begin

For Dg_no:= 0 *to* K_D_S[K_no]-1 *do*

For Onceki_k_dg_no:= 0 *to* K_D_S[K_no-1]-1 *do*

Begin

Ag[K_no, Dg_no, Onceki_k_dg_no]:= *Random*(100)/100;

Dt_Ag[K_no, Dg_no, Onceki_k_dg_no]:= 0;

End;

End;

End;

Procedure Girisleri_sifirla;

Var

K_no, Dg_no: *integer*;

Begin

For K_no:= 0 to M_K-1 do
 For Dg_no:= 0 to K_D_S[K_no]-1 do
 G[K_no, Dg_no]:= 0;

End;

Procedure Set_G_Degerlerini_Oku (E_Seti_no: integer);

{ Giriş örneklerini oku }

Begin

Seek(Formatli_G_Dosya, E_Seti_no);
 Read(Formatli_G_Dosya, Veri);
 For i:= 0 to G_K_Dg_S-1 do G[0, i]:= Veri.G[i];

End;

Function Sigmoid (x: double): double;

var

F: double;

Begin

If x>=1E6 then F:=1
 else If x<= -1E6 then F:=0
 else F:= 1/(1+exp(-x));

Sigmoid:= F;

End;

Procedure İleri_yayılım;

{ İleri yayılım (İng. Feed-Forward) }

var

K_no,

Dg_no, K_Dg_S,

Onceki_k_dg_no,

Onceki_K_Dg_S: integer;

x, th: double;

Begin

For K_no:= 1 to M_K-1 do

Begin

K_Dg_S:= K_D_S[K_no];

Onceki_K_Dg_S:= K_D_S[K_no-1];

For Dg_no:= 0 to K_Dg_S-1 do

Begin

x:= 0;

For Onceki_k_dg_no:= 0 to Onceki_K_Dg_S-1 do

x:= x + Ag[K_no, Dg_no, Onceki_k_dg_no] * G[K_no-1, Onceki_k_dg_no];

G[K_no, Dg_no]:= Sigmoid(x-Teta[K_no, Dg_no]);

End;

End;

End;

```

Function Hata_geri_yayilimi( E_Seti_no: Integer ): double;
{ Hatanın geriye yayılması algoritması (İng. Back-Propagation) }
var
K_no, Dg_no,
Onceki_k_dg_no: integer;
F_Hata, Hata, x, d, w, YSA_sonucu: double;
Begin
Hata:= 0;
For Dg_no:= 0 to C_K_Dg_S-1 do
  Begin
YSA_sonucu:= G[M_k-1, Dg_no];
F_Hata:= Hedef_Deger[E_Seti_no, Dg_no] - YSA_sonucu;
Dt_hata[M_k-1, Dg_no]:= F_Hata*YSA_sonucu*(1-YSA_sonucu);
Hata:= Hata+ F_Hata*F_Hata;
  End;
For K_no:= M_k-1 downto 1 do
  Begin
For Onceki_k_dg_no:= 0 to K_D_S[K_no]-1 do
  Begin
x:= 0;
YSA_sonucu:= G[K_no-1, Onceki_k_dg_no];
For Dg_no:= 0 to K_D_S[K_no]-1 do
  Begin
d:= Dt_hata[K_no, Dg_no];
w:= Ag[K_no, Dg_no, Onceki_k_dg_no];
x:= x + d*w;
  End;
Dt_hata[K_no-1, Onceki_k_dg_no]:= YSA_sonucu*(1-YSA_sonucu)*x;
  End;
  End;
Hata_geri_yayilimi:= 0.5*Hata;
End;

```

```

Procedure Ogrenme (Alfa, Epsilon: double );
var
K_no, Dg_no, Onceki_k_dg_no: integer;
di, yo, ew, et: double;
Begin
For K_no:= 1 to M_K-1 do
  Begin
For Dg_no:= 0 to K_D_S[K_no]-1 do
  Begin
et:= Dt_hata[K_no, Dg_no];
Dt_teta[K_no, Dg_no]:= -Epsilon*et + Alfa*Dt_teta[K_no, Dg_no];
Teta[K_no, Dg_no]:= Teta[K_no, Dg_no] + Dt_teta[K_no, Dg_no];
  End;
  End;
For K_no:= 1 to M_K-1 do

```

```

Begin
For Dg_no:= 0 to K_D_S[K_no]-1 do
  Begin
    di:= Dt_hata[K_no, Dg_no];
    For Onceki_k_dg_no:= 0 to K_D_S[K_no-1]-1 do
      Begin
        yo:= G[k_no-1, Onceki_k_dg_no];
        ew:= di*yo;
        Dt_Ag[K_no, Dg_no,
          Onceki_k_dg_no]:= Epsilon*ew + Alfa*Dt_Ag[K_no, Dg_no,
Onceki_k_dg_no];
        Ag[K_no, Dg_no, Onceki_k_dg_no]:= Ag[K_no, Dg_no,
Onceki_k_dg_no] +
Dt_Ag[K_no, Dg_no, Onceki_k_dg_no];
      End;
    End;
  End;
End;

```

```

Procedure Gecici_matrisi_kur( it, E_Seti_no: integer );
var
i, j: integer;
Begin
For i:= 0 to G_K_Dg_S-1 do
  Gecici_matris[it, E_Seti_no, i]:= G[0, i];
For i:= 0 to C_K_Dg_S-1 do
  Gecici_matris[it, E_Seti_no, G_K_Dg_S+i]:= G[M_K-1, i];
End;

```

```

Procedure Agirliklari_yaz;
var
K_no, dg_no, onceki_k_dg_no: integer;
Begin
Rewrite(Ag_Dosya);
For K_no:= 1 to M_K-1 do
  Begin
    For dg_no:= 0 to K_D_S[K_no]-1 do
      For onceki_k_dg_no:= 0 to K_D_S[K_no-1]-1 do
        Begin
          Write(Ag_Dosya, Ag[K_no, dg_no, onceki_k_dg_no]);
          Write(Ag_Dosya, Teta[K_no, dg_no]);
        End;
      End;
    End;
  End;
End;

```

```

Procedure Agirliklari_oku;
var
K_no, dg_no, onceki_k_dg_no: integer;

```

Begin

Reset(Ag_Dosya);

For K_no:= 1 *to* M_K-1 *do*

Begin

For dg_no:= 0 *to* K_D_S[K_no]-1 *do*

For onceki_k_dg_no:= 0 *to* K_D_S[K_no-1]-1 *do*

Begin

read(Ag_Dosya, Ag[K_no, dg_no, onceki_k_dg_no]);

read(Ag_Dosya, Teta[K_no, dg_no]);

End;

End;

End;

Procedure Kullanim_hatasi;

Begin

Writeln('Program parametreli kullanılacaksa: { Aksi takdirde Dosya adı:"Work" alınacaktır }');

Writeln('YSA <Dosya adı> { Not: Dosya adında uzantı yok }');

halt;

End;

Procedure T_degerleri;

var

Satir, Sayi: *String*;

Donusum_Hatasi: *Integer*;

Begin

Rewrite(Formatli_T_Dosya);

Reset(T_Dosya);

While not eof(T_Dosya) *do*

Begin

Readln(T_Dosya, Satir);

For i:= 0 *to* G_K_Dg_S-1 *do*

Begin

If pos(' ', Satir) <> 0 *then* Sayi:= Copy(Satir, 0, pos(' ', Satir)-1) *else*

Sayi:= Satir;

Donusum_Hatasi:= 0;

Val(Sayi, T_veri.G[i], Donusum_Hatasi);

If Donusum_Hatasi <> 0 *then Begin Writeln*('Test dosyasında hata var!');

halt; **End**;

T_veri.G[i]:= T_veri.G[i]/G_bolenleri[i];

Delete(Satir, 1, length(Sayi)+1);

End;

Seek(Formatli_T_Dosya, Filesize(Formatli_T_Dosya));

Write(Formatli_T_Dosya, T_veri);

End;

Close(T_Dosya);

Close(Formatli_T_Dosya);

End;

Procedure *Giris_degerleri*;

var

Satir, Sayi: *String*;

Donusum_Hatasi: *Integer*;

Begin

Rewrite(Formatli_G_Dosya);

Reset(G_Dosya);

While not eof(G_Dosya) *do*

Begin

Readln(G_Dosya, Satir);

For i:= 0 to G_K_Dg_S-1 *do*

Begin

Sayi:= Copy(Satir, 0, pos(' ', Satir)-1);

Donusum_Hatasi:= 0;

Val(Sayi, Veri.G[i], Donusum_Hatasi);

If Donusum_Hatasi<>0 then *Begin Writeln*('Giris Dosyasında hata var!');

halt; *End*;

Veri.G[i]:= Veri.G[i]/G_bolenleri[i];

Delete(Satir, 1, length(Sayi)+1);

End;

For i:= 0 to C_K_Dg_S-1 *do*

Begin

If pos(' ', Satir)<>0 then Sayi:= Copy(Satir, 0, pos(' ', Satir)-1)

else Sayi:= Satir;

Donusum_Hatasi:= 0;

Val(Sayi, Veri.C[i], Donusum_Hatasi);

If Donusum_Hatasi<>0 then *Begin Writeln*('Giris Dosyasında hata var!');

halt; *End*;

Veri.C[i]:= Veri.C[i]/C_bolenleri[i];

Delete(Satir, 1, length(Sayi)+1);

End;

Seek(Formatli_G_Dosya, Filesize(Formatli_G_Dosya));

Write(Formatli_G_Dosya, Veri);

End;

Close(G_Dosya);

Close(Formatli_G_Dosya);

End;

Procedure *Egitme*;

var

st1: *String*[7];

st2, st3, st4: *String*[9];

E_Seti_no, count: *Integer*;

Oran: *array* [0..E_Seti_S-1] *of double*;

B_hata, Top_hata, Hata, Hata_miktari, Top_oran: *Double*;

Begin

Rewrite(C_dosya);

```

Baslangic_Degerleri;
j:= 0;
While ( j < M_iter_S ) do
  Begin
    For count:= 0 to C_araligi-1 do
      Begin
        Hata:= 0;
        For E_Seti_no:= 0 to E_Seti_S-1 do
          Begin
            Girisleri_sifirla;
            Set_G_Degerlerini_Oku( E_Seti_no );
            Ileri_yayilim;
            Hata_miktari:= Hata_geri_yayilimi( E_Seti_no );
            Ogrenme( Alfa, Epsilon );
            Hata:= Hata+Hata_miktari;
          End;
        Hata:= Hata / C_K_Dg_S ;
        Epsilon:= 0.5 * Sqrt( Hata );
        Inc( j );
      End;
    { Ara test etme, çıktı sayısı kadar bu işlem tekrarlanır. }
    Top_hata:= 0;
    For E_Seti_no:= 0 to E_Seti_S-1 do
      Begin
        Set_G_Degerlerini_Oku( E_Seti_no );
        Ileri_yayilim;
        B_hata:= Hata_geri_yayilimi(E_Seti_no );
        Top_hata:= Top_hata + B_hata;
        Gecici_matrisi_kur( Trunc( j/C_araligi ) - 1, E_Seti_no );
      End;
    Write( C_dosya, 'SetNo');
    For i:= 1 to g_K_dg_S do Write(C_dosya, ' X', i: 2, ' ');
    For i:= 1 to C_K_dg_S do Write(C_dosya, ' Y', i: 2, ' ');
    For i:= 1 to C_K_dg_S do Write(C_dosya, ' Y"', i: 2, ' ');
    Writeln( C_dosya, '' ); Write(C_dosya, '—— ');
    For i:= 1 to g_K_dg_S do Write(C_dosya, '——— '); Write(C_dosya, ' ');
    For i:= 1 to c_K_dg_S*2 do Write(C_dosya, '——— '); Writeln( C_dosya, '' );
    For i:= 0 to E_Seti_S-1 do
      Begin
        str( i: 3, st1 );
        Write(C_dosya, st1+ ' ');
        For ii:= 0 to G_K_Dg_S-1 do
          Begin
            str(Gecici_matris[Trunc(j/C_araligi)-1, i, ii]: 7: 6, st1 );
            Write(C_dosya, st1+ ' ');
          End;
        Write(C_dosya, ' ');
        For ii:= 0 to C_K_Dg_S-1 do

```

```

    Begin
        str( Gecici_matris[Trunc(j/C_araligi)-1, i, G_K_Dg_S+ii]: 7: 6,
st1);
        Write( C_dosya, st1+' ');
        End;
        For ii:= 0 to C_K_Dg_S-1 do
            Begin
                str(Hedef_Deger[i, ii]: 7: 6, st1);
                Write( C_dosya, st1+' ');
                End;
            Writeln(C_dosya, ' ');
            End;
        Top_hata:= Top_hata/C_K_Dg_S;
        str ( j: 4, st1 ); str ( Epsilon: 9: 6, st2 );
        str ( Hata: 9: 6, st3 ); str ( Top_hata: 9: 6, st4 );
        Writeln( C_dosya, 'İterasyon= ', st1, ' Epsilon= ', st2, ' Bireysel Hata= ', st3, '
Toplam Hata= ', st4 );
        Writeln( C_dosya );
        Top_oran:= 0;
        For E_Seti_no:= 0 to C_K_Dg_S-1 do
            Begin
                Oran[E_Seti_no]:= Trunc(Gecici_matris[Trunc(j/C_araligi)-1, E_Seti_no,
E_Seti_no]);
                Top_oran:= Top_oran + Oran[E_Seti_no];
                End;
            Top_oran:= Top_oran/C_K_Dg_S; ;
            If ( j = M_iter_S ) then Writeln( C_dosya, 'Öğrenme ve test bitiş zamanı:');
            Agirliklari_yaz;
            End;
        Close( C_dosya );
        End;

```

Procedure Test;

var

T_seti_S, Set_no: *integer*;

st1: *String*[7];

Begin

Append(C_dosya); { T sonuçları çıktı dosyasına ilave edilecektir. }

Reset(Formatli_T_Dosya);

T_seti_S:= *Filesize*(Formatli_T_Dosya);

Writeln(C_dosya, ''); *Writeln*(C_dosya, '');

Writeln('Test');

Agirliklari_oku;

For Set_no:= 0 to T_seti_S-1 *do*

Begin

Seek(Formatli_T_Dosya, Set_no); { Test dosyasındaki test setleri okunuyor }

Read(Formatli_T_Dosya, T_veri);

For i:= 0 to G_K_Dg_S-1 *do* G[0, i]:= T_veri.G[i];

```

Ileri_yayilim;
Write(C_dosya, Set_no+1, ' . Test Seti Sonuçları: ');
For ii:= 0 to C_K_Dg_S-1 do
    Begin
        str(G[M_K-1, ii]*C_bolenleri[ii]: 4: 3, st1);
        Write(C_dosya, st1+' ');
    End;
Writeln(C_dosya, '');
End;
Close(C_dosya); { Çıkış dosyasını kapat }
Close(Formatli_T_Dosya);
End;

```

```

Procedure Hedef_degerler;
Begin
Reset(Formatli_G_Dosya);
For i:= 0 to E_Seti_S-1 do
    Begin
        Read(Formatli_G_Dosya, Veri);
        For j:= 0 to C_K_Dg_S-1 do
            Hedef_Deger[i, j]:= Veri.C[j];
        End;
    End;
Close(Formatli_G_Dosya);
End;

```

```

Procedure Calisma_dosyasi;
Begin
If paramcount<>0 then
If paramcount>1 then Kullanim_hatasi else
    Begin
        If pos(' ', Paramstr(1))>0 then Kullanim_hatasi;
        Assign(G_Dosya, Paramstr(1)+' .in');
        Assign(Formatli_G_Dosya, Paramstr(1)+' .fin');
        Assign(C_dosya, Paramstr(1)+' .out');
        Assign(Ag_Dosya, Paramstr(1)+' .wgt');
        Assign(T_Dosya, Paramstr(1)+' .tst');
        Assign(Formatli_T_Dosya, Paramstr(1)+' .fts');
    End else
    Begin
        Assign(G_Dosya, 'work.in');
        Assign(Formatli_G_Dosya, 'work.fin');
        Assign(C_dosya, 'work.out');
        Assign(Ag_Dosya, 'work.wgt');
        Assign(T_Dosya, 'work.tst');
        Assign(Formatli_T_Dosya, 'work.fts');
    End;
End;

```

Procedure Matris_sifirla;

var

E_S, C_S: *integer*; {E_S: Eğitim sayısı , C_S: Çıktı sayısı}

Begin

For C_S:= 0 *to* Trunc(M_iter_S/C_araligi)-1 *do*

For E_S:= 0 *to* E_Seti_S-1 *do*

For j:= 0 *to* (G_K_Dg_S+C_K_Dg_S-2) *do*

Gecici_matris[C_S, E_S, j]:= 0;

End;

BEGIN { Ana Program Buradan Başlıyor }

Calisma_dosyasi;

Giris_degerleri;

Hedef_degerler;

T_degerleri;

Epsilon:= Ilk_Epsilon;

Matris_sifirla;

Clrscr;

Write('Eğitme (0) veya T (1) ? ');

Readln(cevap);

Clrscr; *If* cevap= 0 *then* Egitime *else* Test;

END.

8. KAYNAKLAR

- Açıkel, H. 1995. Karapınar volkanik agregasından (TS 4047'ye uygun) hazır döşeme ve çatı plakları imali. Doktora Tezi, Fen Bilimleri Enstitüsü, Konya.
- Adeli, H. 1996. Innovative computing in civil engineering. *Proceedings of the 2nd International Conference in Civil Engineering on Computer Applications Research and Practice*, 1:1-5, Bahrain.
- Arslan, A., İNCE, R. 1994. The neural network based design of edge supported reinforced concrete slabs. *Artificial Intelligence and Object Oriented Approaches for Structural Engineering* 91-97.
- Arslan, A., İNCE, R. 1994. Yapı mühendisliğinde yapay sinir ağları ve uygulama potansiyeli. İnşaat Mühendisliğinde Bilgisayar Kullanımı IV. Sempozyumu 14-16 Haziran, 219-228.
- Arslan, A., İNCE, R. 1995. Geriye yayılma yapay sinir ağı kullanılarak betonarme kolonların tasarımı. *Journal of Engineering and Environmental Sciences* 19:127-135.
- Arslan, A., İNCE, R. 1996. The neural network approximation to the size effect in fracture of cementitious materials. *Engineering Fracture Mechanics* 2:249-261.
- Berke, L., Patnaik, S.N., MURTHY, P.L.N. 1993. Optimum design of aerospace structural components using neural networks. *Computers & Structures* 6:1001-1010.
- Blum, E. K., Li, L. K. 1991. Approximation theory and feedforward networks. *Neural Networks* 4:511-515.
- Erbatur, F. 1993. Advances in structural optimization. İnşaat Mühendisliğinde Gelişmeler 1. Teknik Kongre 2:763-770, 25-27 Ekim, Gazi Mağusa-KKTC.
- Ghaboussi, J., Garret, J.H. and Wu, X. 1991. Knowledge-based modelling of material behaviour with neural networks. *Journal of Engineering Mechanics* 1:132-153.
- Ghaboussi, J., Joghataie, A. 1995. Active control of structures using neural networks. *Journal of Engineering Mechanics* 4:555-567.
- Goh, A.T.C. 1995. Prediction of ultimate shear strength of deep beams using neural networks. *ACI Structural Journal* 1:28-32.

- Gülkan, P. ve Yakut, A. 1995. Uzman sistem ve sinirsel şebekelerin deprem hasarının belirlenmesinde kullanılması. Üçüncü Ulusal Deprem Mühendisliği Konferansı, 27-31 Mart, İstanbul 506-528.
- Hajela, P., Berke, L. 1991. Neurobiological computational models in structural analysis and design. *Computers & Structures* 4:657-667.
- Hopfield, J.J., Tank, D.W. 1985. "Neural" computation of decisions in optimization problems. *Biological Cybernetics* 52:141-152.
- Hornik, K. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Networks* 4:251-257.
- Karlık, B. 1994. Çok fonksiyonlu protezler için YSA kullanarak miyoelektrik kontrol. Doktora Tezi, Fen Bilimleri Enstitüsü, İstanbul.
- Lippmann, R.P. 1987. An introduction to computing with neural nets. *IEEE ASSP Magazine* April: 4-22.
- Moselhi, O. 1996. Neural networks' applications in civil engineering. *Proceedings of the 2nd International Conference in Civil Engineering on Computer Applications Research and Practice*, 1:7-14, Bahrain.
- Moselhi, O., Hegazy, T., Fazio, P. 1991. Neural networks as tools in construction. *Journal of Construction Engineering and Management* 4:606-625.
- Öztemel, E. 1996. Bilgisayarda öğrenme ve yapay sinir ağları. *Otomasyon* Mart, 134-140.
- Sanad, A., Saka, M.P. 1996. Design of reinforced concrete structures using neural networks. *Proceedings of the 2nd International Conference in Civil Engineering on Computer Applications Research and Practice*, 1:69-83, Bahrain.
- Siao, W.B. 1993. Strut-and-tie model for shear behaviour in deep beams and pile caps failing in diagonal splitting. *ACI Structural Journal* 4:356-363
- Vanluchene, R.D., Sun, R. 1990. Neural networks in structural engineering. *Microcomputers in Civil Engineering* 5:207-215.
- Wu, X., Ghaboussi, J., Garret, J.H. 1992. Use of neural networks in detection of structural damage. *Computers & Structures* 4:649-659.