**KOCAELİ ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

# YAPAY SİNİR AĞLARI YAKLAŞIMIYLA LASTİK KALIBI MALİYETLERİNİN TAHMİN EDİLMESİ

**AYNUR GÜRSOY**

**KOCAELİ 2012**

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

# YAPAY SİNİR AĞLARI YAKLAŞIMIYLA LASTİK KALIBI MALİYETLERİNİN TAHMİN EDİLMESİ

AYNUR GÜRSOY

| | |
|---|---|
| **Prof.Dr. Nilgün FIĞLALI**<br>**Danışman, Kocaeli Üniv.** | ..... |
| **Yrd.Doç.Dr. Pınar YILDIZ KUMRU**<br>**Jüri Üyesi, Kocaeli Üniv.** | ..... |
| **Yrd.Doç.Dr. Ümit TERZİ**<br>**Jüri Üyesi, Beykent Üniv.** | ..... |

**Tezin Savunulduğu Tarih: 27.12.2012**

## ÖNSÖZ ve TEŞEKKÜRLER

Araba lastik üretimi yapan firmalarda üretim maliyetlerinin hesaplanması çok değişken ürün tipleri olduğundan dolayı uzun ve zor bir süreçtir. Verilerin doğru ve uzun süreçleri kapsayacak şekilde tutulması gerekir.

Bu çalışmayla gerçeğe en yakın tahminleri yapabilen yapay sinir ağı yapısı oluşturularak her farklı kalıbın maliyetine etki edebilecek faktörler göz önünde bulundurulabilecek ve maliyetler kısa sürede hesaplanabilecektir. Günümüzde işletmelerin en çok önemsediği maliyetleri düşürme politikasına bununla birlikte başlanabilecek ve satış fiyatları da maliyetler doğrultusunda verilebilecektir.

Çalışmalarım sırasında bilgi, emek ve tecrübesiyle her türlü konuda bana destek olan ve yol gösteren danışmanım Sayın Prof. Dr. Nilgün Fığlalı'ya, bilgi ve fikirleriyle beni aydınlatan Prof. Dr. Alpaslan Fığlalı'ya, çalışmanın her aşamasında bilgileriyle beni aydınlatan ve yol gösteren değerli arkadaşım Ahmet Cihan'a, bana uygulama imkanı veren iş yeri sahiplerine teşekkür ederim. Ayrıca hayatım boyunca beni destekleyen aileme de sonsuz minnet duygularımı sunarım.

Aralık-2012 Aynur Gürsoy

## İÇİNDEKİLER

## ŞEKİLLER DİZİNİ

## TABLOLAR DİZİNİ

## SİMGELER DİZİNİ VE KISALTMALAR

$E_{eğitim}$ : Öğrenme hata oranı
$E_{test}$ : Test seti hata oranı
$E^p$ : Eğitim örneğinin gerçek çıktı değeri
g : Ağın gradyen fonksiyonu
H : Hessian matsi
I : Birim matris
J : Jakobien matrisi
w : Yapay sinir ağı için ağırlık vektör
x : Giriş vektörü
y : Yapay sinir ağı için çıkış vektörü
$\alpha$ : Momentum katsayısı
$\lambda$ : Öğrenme katsayısı
$\mu$ : Marguardt parametresi

### Kısaltmalar

ANN : Artificial Neural Network (Yapay Sinir Ağı)
ART : Adaptive Resonance Theory (Adaptif Rezonans Teorisi)
BFY : Birim Fiyat Yöntemi
CNC : Computer Numerical Control ( Bilgisayar Destekli Kontrol)
EMB : Eşik Mantık Birimi
LM : Levenberg- Marguardt Eğitim Algoritması
LMS : Least Mean Square Rule ( En Küçük Kareler Kuralı)
LVQ : Linear Vektor Quantization (Lineer Vektör Nicelemesi)
ME : Mean Error (Ortalama Mutlak Hata)
MLP : Multilayer Perceptron (Çok Katmanlı Algılayıcı)
MSE : Mean Square Error (Hata Kareleri Ortalaması)
PBNN : Probability Based Neural Networks (Olasılık Tabanlı Yapay Sinir Ağı)
RA : Regresyon Analizi
RBF : Radial Basias Function ( Radyal Tabanlı Fonksiyon Ağları)
RBN : Radial Basis Networks ( Radyal Tabanlı Yapay Sinir Ağı)
RMSE : Root Mean Square Errors (Ortalama Karesel Hata)
SOM : Self Organizing Maps (Kendi Kendine Organize Eden Model)
YSA : Yapay Sinir Ağı

## YAPAY SİNİR AĞLARI YAKLAŞIMIYLA LASTİK KALIBI MALİYETLERİNİN TAHMİN EDİLMESİ


## ÖZET

Bu çalışmada son yıllarda sıkça kullanılan bir tahmin yöntemi haline gelen Yapay Sinir Ağları (YSA) kullanılarak araba lastik kalıbı maliyetleri tahmin edilmeye çalışılmıştır. Karmaşık dinamiklerinden dolayı oldukça değişken ve etkileşimli bir yapıya sahip kalıp maliyetlerinin bulunabilmesi için doğru ve güvenilir tahminlere gereksinim vardır.

Kalıp maliyetlerini belirleyen alt ve üst yanak dış çapı, iç çapı, kalınlıkları, desen karmaşıklığı ve kasa türü ağ mimarisinde ana kriter olarak alınmıştır. Yapay sinir ağının ürettiği sonuçların gerçeği ne kadar yansıttığı istatistiksel olarak araştırılmıştır. Elde edilen sonuçlar, Regresyon Analizi (RA) ile yapılan maliyet hesaplamaları ile karşılaştırılmış ve yapay sinir ağları ile gerçeğe daha yakın maliyet tahminleri sağlanmıştır.



**Anahtar Kelimeler:** Lastik Kalıbı, Maliyet Tahmini, Regresyon Analizi, Yanak, Yapay Sinir Ağları.

**FORECASTING COSTS OF TIRE MOLD WITH ARTIFICIAL NEURAL NETWORKS APPROACH**


## ABSTRACT

This study has tried to forecast the future costs of automobile tire molds using Artificial Neural Networks (ANN) which became a very popular forecast technique in the recent years. For calculating the cost of a mold which has a considerably unsteady and interactive structure because of its complex dynamics, straight and confidential predictions are needed.

Top and bottom sidewall outer diameters, inner diameters, thicknesses, complexity of the pattern and case type were assumed as mean criteria of the cost of each tire mold. The accuracy of the results produced by the neural network was statistically evaluated. The results were compared with cost estimates made by Regression Analysis (RA) and it is comprehensible that the results of ANN are nearer than the results of Regression Analysis to the real costs of these tire molds.



**Keywords:** Tire Mold, Cost Forecasting, Regression Analysis, Sidewall, Artificial Neural Networks.

# GİRİŞ

Endüstride büyük bir yer edinmiş olan kalıpçılık artık tüm sanayi kollarında ihtiyaç duyulan bir meslek olmuştur. Teknolojinin gelişmesiyle hem imalat aşamasında hem de tasarım aşamasında büyük kolaylıklar sağlanmıştır. Tasarım aşamasında 3D modelleme ve benzeşim programları yardımıyla optimizasyon yapılmakta ve imalat aşamasında kalıbın imali CNC teknolojisinin gelişmesiyle hassas bir şekilde yapılmaktadır. Kalıptan çıkan ürünler verilen toleranslar dahilinde istenilen ölçülerde yapılabilmektedir [1].

Saç şekillendirme yöntemi; en eski imalat yöntemlerinden biri olmakla birlikte, gerek hızlı imalat özelliği ile seri imalat sağlaması ve gerekse ekonomikliği nedeniyle günümüzde, özellikle otomotiv ve beyaz eşya endüstrisi başta olmak üzere birçok alanda kullanılmaktadır.

Tüm imalat yöntemleri gibi, saç şekillendirme yöntemleri de zaman içerisinde gelişerek kendilerini yenilemişlerdir. Dünya çapında meydana gelen sanayi devrimi, teknolojinin hızla gelişmesi ve her geçen gün değişerek artan tüketici ihtiyaçları ve bu ihtiyaçların hızlı ve verimli bir şekilde karşılanma çabası, ülkelerin ticari kazançlarını ve dünyada ki imalat potansiyelini ellerinde bulundurma istekleri neticesinde imalat yöntemlerinde değişmeler ve gelişmeler olmuştur.

Günümüzde bilgisayarlar ve bilgisayar sistemleri insan hayatının vazgeçilmez bir parçası haline gelmiştir. Bilgisayarların zaman içerisinde, büyük miktarda verileri filtreleyerek düzenleyebilen ve mevcut verilerle olaylar hakkında yorum yapabilen sistemler haline geldikleri görülmektedir [2].

Yapay Sinir Ağları (YSA), beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanır. YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar şeklinde düzenlenir. Donanım olarak elektronik devrelerle ya da bilgisayarlarda yazılım olarak gerçeklenebilir. Beynin bilgi işleme yöntemine uygun olarak YSA, bir

öğrenme sürecinden sonra; bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemcidir. Öğrenme süreci, arzu edilen amaca ulaşmak için YSA ağırlıklarının yenilenmesini sağlayan öğrenme algoritmalarını içerir [3].

Bu çalışma, lastik kalıbı üretiminde maliyetlere etki eden faktörlerin kesin olarak belirlenmesi ve karmaşık dinamiklerinden dolayı oldukça değişken, etkileşimli bir yapıya sahip araba lastik kalıbı maliyetlerinin bulunması amacıyla yapılmıştır. Kalıp maliyetlerine etki eden faktörler belirlenmiş ve kayıtlı bilgilerden yararlanılarak veri seti oluşturulmuştur. Bu verileri kullanarak çalışan YSA yöntemiyle kalıp maliyetleri tahmin edilmeye çalışılmıştır.

Birinci bölümde, yapılan literatür araştırması sonucunda YSA yöntemi kullanılarak yapılmış çalışmalardan bahsedilmiştir.

İkinci bölümde, YSA'nın Tanımı, Tarihsel Gelişimi, Kullanım Alanları, Avantajları, Dezavantajları ve Çalışma Şekilleri, Öğrenme Stratejileri, Öğrenme Kuralları, Modelleri ve Tasarımı hakkında bilgi verilmiştir.

Üçüncü bölümde ise, lastik kalıbı üreten işletme ve kalıp üretimi iş akışı hakkında bilgi verilmiş, maliyetlere etki eden faktörler tanımlanmış, uygun YSA konfigürasyonundan ve Levenberg-Marquardt öğrenme algoritmasından bahsedilmiş, kalıp maliyetleri YSA ve RA yöntemleriyle tahmin edilmiş, sonuçlar karşılaştırılmış ve tahmin kalitesi istatistiksel olarak araştırılmıştır.

# 1. KURAMSAL TEMELLER VE KAYNAK ARAŞTIRMASI

Maliyet; Üretimde bir mal elde edilinceye değin harcanan değerlerin toplamı olarak tanımlanmaktadır. Tahmin; "yaklaşık olarak değerlendirme, oranlama", "akla, sezgiye veya bazı verilere dayanarak bir şeyi, olayı kestirme" ya da "önceden kestirilen, düşünülen şey" anlamına gelmektedir.

Yatırım kaynaklarının kısıtlı ve teorik olarak insan ihtiyaçlarının da sonsuz olduğu göz önüne alındığında, maliyetlerin çok titiz çalışmalarla belirlenmesi, programlanması ve kontrol edilmesi gerekliliği büyük önem arz etmektedir [4].

Tanımlayıcı modeller yaklaşımı; diğer alanlarda başarılı olan ve maliyet modeli araştırmalarında on beş yıldan fazla bir süredir kullanılır ve maliyetleri miktarlar yerine tasarımın tanımlayıcı özelliklerine bağlamak fikrini temel alır. Tanımlayıcı modeller regresyon analizleri kullanılarak geliştirilmiştir. Regresyon analizi, hesaplamaların bir ya da daha fazla değişkenin değerlerinin bilinmesinden yararlanılarak başka bir değişkenin değerinin, bulunduğu bir yöntemdir.

Gerçekçi modeller; Tahmin yolu ile yapılan maliyet hesaplamalarının doğru bir sonuç vermeyeceği düşüncesinden hareketle kurgulanmış olan modelleri tanımlamaktadır. Gerçekçi modellerin tahmin doğruluğunun miktarlara dayalı modellere göre daha azdır. Buna rağmen, miktarlara dayalı modeller gelişmelerini neredeyse tamamladıkları halde, gerçekçi modellerin gelişme potansiyellerinin çok yüksek olduğu ileri sürülmektedir.

Uzman sistemler; Gelişmiş maliyet modelleri adı altında da incelenen bu modeller ile maliyet tahmini yapabilmek için veri tabanı ve bilgisayardan yararlanılır. Maliyet tahmininde başarılı olabilmek için de bu sistemlerin mutlaka uzman kişiler tarafından yönlendirilmesi gerekir.

Yapay zeka; Latince "intellectus" kelimesinin karşılığı olan zekanın, kavramsal olarak birkaç tarifi şöyledir;

• Cevap vermede, muhtemel çözümleri inceden inceye aramadaki çabukluk ve bir problemin evreleri arasındaki yeni ilişkileri anlayabilme kapasitesidir,

• Yeni bir düzeneği veya kuralı keşfetme ya da bir tahmin yürütme ile ilgili faaliyettir,

• Beynin bilgiyi alıp, hızlı ve doğru olarak analiz etmesidir.

Biyologlar, zekayı çevreye uyum kabiliyeti olarak görürken, eğitimciler öğrenme, psikologlar ilişkileri anlama, bilgisayarcılar bilgiyi işleme kabiliyeti şeklinde değerlendirmektedirler. Şuur, bilinçaltı, ruh gibi açık uçlu bir kelime olduğu için zekanın evrensel bir tarifi yapılamamıştır.

Yapay zeka ise; bir bilgisayar bilim dalıdır, bilgi ve davranışa dayalı sistemler oluşturur ve zeki davranışlar üzerine araştırmalar yapar.

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

Yapay zeka teknikleri aşağıdaki gibi gruplandırılabilir;

• Yapay Sinir Ağları,

• Bulanık Mantık,

• Sezgisel Algoritmalar (Genetik Algoritmalar, Tabu Arama, Karınca Algoritması, Isıl İşlemler, Bağışıklık Sistemi vb.)

• Uzman Sistemler [4].

YSA'nın en önemli özelliklerinden birisi, gerçek hayattaki olası doğrusal olmayan yapıları da dikkate alabilmesidir. YSA, herhangi bir sürekli fonksiyona veya türevlerine yakınsama yeteneğine sahiptir ve bu yüzden Evrensel Fonksiyon Yakınsayıcı Yöntem olarak tanımlanmaktadırlar." Analiz konusunun içerdiği veri setinin doğrusal veya doğrusal olmayan yapı içeriyor olması, analiz sonuçlarını

etkileyecek önemli bir faktördür. Bu yüzden YSA'nın doğrusal olmayan yapıları da dikkate alabilmesi kendisinin önemli bir özelliğidir.

Esin kaynağı insan beyninin çalışma sistemi olan bu yöntem, eğitme veya başlangıç tecrübesi sayesinde, veriyi kullanarak öğrenme yeteneğine sahiptir. Bu özelliği sayesinde ise geleneksel teknikler için çok karmaşık kalan problemlere çözüm sağlayabilmektedirler. Ayrıca, insanların kolayca yapabildiği ama geleneksel metotların uygulanamadığı basit işlemler için de oldukça uygundurlar.

YSA'lar geleneksel işlemcilerden farklı şekilde işlem yapmaktadırlar. Geleneksel işlemcilerde, tek bir merkezi işlem elemanı her hareketi sırasıyla gerçekleştirir. YSA modelleri ise, her biri büyük bir problemin bir parçası ile ilgilenen çok sayıda basit işlem elemanlarından oluşma ve bağlantı ağırlıklarının ayarlanabilmesi gibi özelliklerinden dolayı önemli derecede esnek bir yapıya sahiptirler.

Toplam işlem yükünü paylaşan işlem elemanlarının birbirleri arasındaki yoğun bağlantı yapısı sinirsel hesaplamanın temel güç kaynağıdır. Bu yerel işlem yapısı sayesinde, YSA yöntemi en karmaşık problemlere bile uygulanabilmekte ve tatminkar çözümler sağlayabilmektedir.

YSA modelleri sınırsız sayıda değişken ve parametre ile çalışabilmektedir. Bu sayede mükemmel bir öngörü doğruluğu ile genel çözümler sağlanabilmektedir [4].

İşlem verilerinin birbirleri arasında yoğun bağlantı kurdukları, bilinen örnekleri kullanarak daha önce karşılaşmamış durumlarda genelleme yapabildikleri, hatalı veya kayıp veriler için çözüm üretebildikleri, karmaşık veya sorunlu verilerden bile anlam çıkarabildikleri ve çok değişken verilere sahip olan lastik kalıbı maliyetlerinin tahmini için bu çalışmada YSA yöntemi kullanılmıştır.

İşletmeler arasındaki artan rekabet nedeniyle, müşterinin istediği özelliklerde ürün üretmek (müşteri memnuniyeti), tam zamanında teslim ve maliyetlerin düşürülmesi tüm işletmeler için önem arz etmektedir [4].

Yapılan literatür taraması sonucunda kalıp maliyetleri, ham petrol fiyatlarının tahmini, ikinci el otomobil fiyatlarının tahmini, toplu konut inşaat maliyetleri ve yapı maliyetlerinin tahmini vb. YSA çalışmaları aşağıda sunulmuştur.

Vikaros ve Miko (2011) , "Artificial Neural Network Approach For Injection Mould Cost Estimation" adlı çalışmalarında; kalıp maliyetinin etkileyen on üç girdi parametresine dayalı olarak YSA yöntemiyle on dört fiyat tipi tahmin etmişlerdir. En düşük hata değeriyle YSA ile maliyet tahminin en doğru sonucu verdiğini görmüşlerdir [6].

Kaynar, Taştan ve Demirkoparan (2010) "Ham Petrol Fiyatlarının Yapay Sinir Ağları İle Tahmini" adlı çalışmalarında; ham petrol fiyatlarını tahmin etmek için klasik zaman serileri analizi yöntemlerinden ARIMA ile veri seti içerisindeki karmaşık ilişkileri modelleyebilen MLP (çok katmanlı ileri beslemeli YSA) ve RBF (radyal tabanlı fonksiyon ağları) yapay sinir ağlarını kullanmışlar ve uygun ağ yapısı ve yeterli sayıda veri kullanıldığında, zaman serilerinin tahmininde yapay sinir ağları istatistiksel yöntemlere alternatif bir yöntem olarak kullanılabildiği sonucuna varmışlardır [7].

"Estimating Resource Requirements at Conceptual Design Stage Using Neural Networks" (1997) adlı makalelerinde Elazouni ve arkadaşları; tahmin modellerinin proje maliyeti ve performansını optimize edebilmek için farklı alternatifleri değerlendirmek üzere tasarımcılara potansiyel yardımcı olabileceğini ifade etmişlerdir. Bu aşamada farklı alternatiflerin maliyetinin belirlenmesinin, yeniden tasarım maliyetinin engellenmesi ve bakım, operasyon ve değiştirme maliyetlerinin minimize edilmesi sayesinde maliyet tasarrufu sağlayacağını vurgulamışlardır. Buna ilave olarak, bu modellerin yüklenicinin son dakika teklif tahmini için çok yararlı olduğunu belirtmişlerdir. Kaynak ihtiyacı tahmini için, tahmin aracı olarak geriye yayılım ağlarının kullanılabilirliğini araştırdıkları bu çalışmada, 28 adet silo inşaatına ait değerler gruplanmıştır. Uygulanan YSA modeli ile elde edilen sonuçlar çoklu regresyon analizi sonuçlarıyla karşılaştırılmış ve YSA modelinin tahmin için oldukça kullanışlı olduğu tespit edilmiştir [8].

Asilkan ve Irmak (2009), "İkinci el otomobillerin gelecekteki fiyatlarının yapay sinir ağları ile tahmin edilmesi" adlı çalışmalarında; yapay sinir ağlarını kullanılarak ikinci el otomobillerin gelecekteki fiyatlarını tahmin etmeye çalışılmışlardır. Girdi olarak Avrupa kökenli çok sayıda ikinci el otomobil web sitesinden elde edilmiş olan ilan verileri kullanılmıştır. YSA uygulaması ile bulunan sonuçlarla karşılaştırılmıştır.

Elde edilen sonuçlar, yapay sinir ağlarının ikinci el otomobillerin gelecekteki fiyatlarını tahmin etmede başarıyla kullanılabileceğini ortaya koymuştur [7].

Kim ve arkadaşları (2004), 530 eski maliyet verisinin yardımı ile çoklu regresyon analizi, YSA ve vaka tabanlı sebeplendirme adlı üç yöntemin performanslarını karşılaştırmışlardır. Çalışmalarının sonucunda YSA esaslı yöntem, diğer iki maliyet değerleme yöntemine göre daha kusursuz sonuçlar vermiştir [9].

Demirel (2007), "Toplu konut inşaat maliyetlerinin yapay sinir ağları ile tahmini" adlı çalışmasında; toplu konut inşaat maliyetlerini YSA ile tahmin etmeyi amaçlamıştır. Bu amaçla betonarme taşıyıcı sisteme haiz ve benzer nitelikteki çok katlı konut projelerinin inşaat maliyetleri hesaplanmış ve mevcut verilerden yararlanılarak oluşturulan çok katmanlı, danışmanlı, geri beslemeli, danışmanlı öğrenme özelliklerinde yapılandırılan YSA'ya veri olarak girilmiştir. Bu yapıların projelerinden hesaplanan; tip kat alanları, yapı yükseklikleri ve toplam dış cephe alanları, ağ mimarisinde ana değerlendirme kriteri olarak alınmıştır. Ağa hesaplatılan maliyet tahminleri, Birim Fiyat Yöntemi (BFY) ve Regresyon Analizi ile yapılan maliyet hesaplamaları ile karşılaştırılmış ve uygulanan YSA yönteminin sağladığı performans değerlendirilmiştir. Oluşturulan YSA'dan sağlanan veriler, Regresyon Analizi verilerine göre BFY ile bulunan maliyetlere daha yakın ve uygulanabilir sonuçlar sağlanmıştır. Bu alandaki çalışmalarda hibrit yöntemlerin kullanılmasının daha verimli tahminler için avantaj sağlayacağı ve farklı yapı tipleri için benzer araştırmaların yapılmasının olumlu gelişmeler yaratacağı sonucuna varılmıştır [3].

Basheer ve Hajmeer (2000), "Artificial neural networks: fundamentals, computing, design and application" adlı çalışmalarında; YSA'nın, beyin hücrelerindeki görev dağılımının ve birbirleri arasındaki ilişkilerin benzerlerinin bilgisayar ortamında oluşturulması olduğunu ifade etmişlerdir. Matematiksel olarak modellenen çok sayıda nöron, birbirlerinden aldıkları bilgileri değerlendirerek bir çıktıya dönüştürmektedirler. Birçok nöronun eşzamanlı olarak çalışması, eğitilebilmesi, denetlenmesi ve çözüme ulaşma hızının kısa olması gibi özellikleri nedeni ile YSA'nın, bilimsel araştırmalarda giderek artan bir şekilde kullanılmakta olduğunu belirtmişlerdir [10].

Weiss ve Kulikowski'nin (1999), "Computer systems that learn"; Hinton'un "How neural networks learn from experience"; Ripley, Barndoff – Nielsen, Jensen ve Kendall'ın "Statistical aspects of neural networks, in networks and chaos–statistical and probabilistic aspects" ve Warren'ın "Neural networks and statistical models" adlı çalışmalarında; YSA'nın son yıllarda hem teorik ve hem de pratik uygulamalar bakımından geliştiği, kendisine olan ilgiyi daha da arttırdığı vurgulanarak YSA modelleri ile istatistik modellerin benzer olduğuna (bazılarının ise aynı olduğuna) dikkat çekilmiştir [11].

Stern'in (1996) "Neural networks in applied statistics", Ripley'in "Pattern recognation and neural Networks", Wang'ın (1999) "An adaptive approach to market development forecasting" ve Yasdi'nin (1999) "Prediction of road traffic using u neural network approach" isimli makalelerinde YSA modelleri ile istatistik modellerin benzer olmasının tesadüfi olmadığını, bu iki alanın sıkı ilişkili olduğunu göstermişlerdir. YSA ve istatistik metotların karşılaştırılması, bu modellerden birinin, uygun olan bir diğerinin geliştirilmesinde önemli olduğunu ortaya koymuşlardır. Birçok pratik problemde, her iki sınıf yöntemlerin kullanılarak hesaplama sonuçlarının karşılaştırılmasının daha iyi çözümün bulunması içinde bir araç olduğunu ifade etmişlerdir [12-14,28].

"An artificial neural network approach to assess project cost and time risks at the front of projects" (1998) adlı yüksek lisans tezinde Liu, projelerin önünde bulunan maliyet ve zamansal risklerin değerlendirilmesi konusunda YSA esaslı bir araştırma yapmış ve sonuçlar, petrol ile gaz endüstrisindeki projelerde, eski basit projelerden öğrenme yaparak genel temayı betimleyebilecek bir YSA oluşturmanın mümkün olabileceğini göstermiştir. Liu, YSA esaslı uygulamaların çoklu regresyon yöntemine göre daha üstün sonuçlar verdiğini belirtmiştir [15].

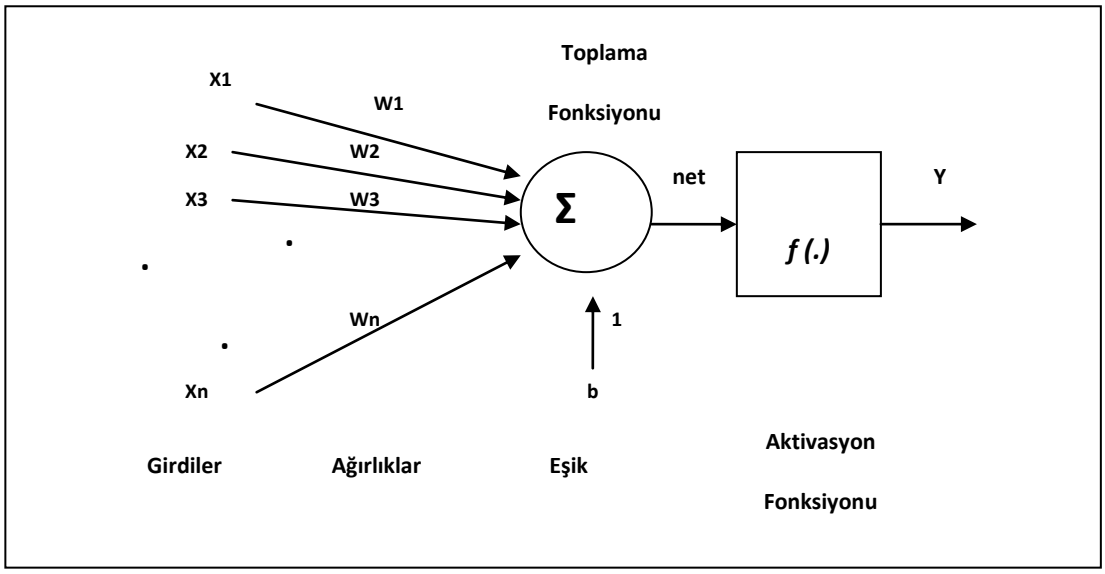
Uğur (2007)," Yapı maliyetlerinin yapay sinir ağı ile analizi" adlı çalışmasında; yapı maliyetlerinin YSA ile tahmin edilmesi amacıyla; betonarme taşıyıcı sistemli ve benzer nitelikteki çok katlı toplu konut projelerinin inşaat maliyetlerini hesaplanmış ve mevcut verilerden yararlanarak oluşturulan çok katmanlı, geri beslemeli, danışmanlı öğrenme özelliklerinde yapılandırılan YSA'na veri olarak girmiştir. Bu yapıların projelerinden hesaplanan; yapı yükseklikleri, tip katlardaki daire sayıları,

tip kat alanları, kat yükseklikleri, toplam katsayıları, kat yükseklikleri, cephe alanları, cephe boşluğu alanları ve ortalama daire alanları, ağ mimarisinde ana değerlendirme kriterleri olarak alınmıştır. Ağa hesaplatılan maliyet tahminleri, Birim Fiyat Yöntemi (BFY) ve RA ile yapılan maliyet hesaplamaları ile karşılaştırılmış ve uygulanan YSA yönteminin sağladığı performans değerlendirilmiştir [4].

Yapılan literatür araştırmasında lastik kalıbının maliyet tahminlenmesi için YSA kullanımına rastlanmamıştır. Araştırmanın uygulama sürecinde henüz imalatına başlanmamış olan yeni tasarımlar için gözlemler yapılarak kalıp maliyeti, yapay sinir ağı yaklaşımıyla tahmin edilmeye çalışmıştır.

## 2. YAPAY SİNİR AĞLARI

Yapay sinir ağları, biyolojik sinir ağlarından esinlenilerek ortaya çıkarılan ve biyolojik sinir ağlarına benzer bazı performans özellikleri içeren bir bilgi işleme sistemidir. Basit bir şekilde insan beyninin çalışma şeklini taklit eden YSA' lar veriden öğrenebilme, genelleme yapabilme, sınırsız sayıda değişkenle çalışabilme vb. birçok önemli özelliğe sahiptir. YSA' nın çalışmasına esas teşkil eden en küçük birimler yapay sinir hücresi ya da işlem elemanı olarak isimlendirilir. En basit yapay sinir hücresi Şekil 2.1'de de görüleceği üzere girdiler, ağırlıklar, birleştirme fonksiyonu, aktivasyon fonksiyonu ve çıkış olmak üzere 5 ana bileşenden oluşmaktadır.




Şekil 2.1.Yapay sinir hücresi [7]

Girdiler ($x_1$, $x_2 \ldots x_n$ ), diğer hücrelerden ya da dış ortamlardan hücreye giren bilgilerdir. Bunlar ağın öğrenmesi istenen örnekler tarafından belirlenir. Ağırlıklar ($w_1$, $w_2 \ldots w_n$), girdi kümesi veya kendinden önceki bir tabakadaki başka bir işlem elemanının bu işlem elemanı üzerindeki etkisini ifade eden değerlerdir. Her bir girdi, o girdiyi işlem elemanına bağlayan ağırlık değeriyle çarpılarak, toplam fonksiyonu aracılığıyla birleştirilir.

Toplam fonksiyonu sonucunda elde edilen değer doğrusal ya da doğrusal olmayan türevlenebilir bir transfer fonksiyonundan geçirilerek işlem elemanının çıktısı hesaplanır.

Yapay sinir ağlarında çok çeşitli ağ yapıları ve modelleri vardır. YSA, Şekil 2.1'de gösterilen bir dizi sinir hücresinin ileri sürümlü ve geri beslemeli bağlantı şekilleri ile birbirine bağlanmasından oluşur. Günümüzde, belirli amaçlarla ve değişik alanlarda kullanılmaya uygun birçok yapay sinir ağı modeli (MLP, RBF, LVQ, Hopfield, Recurren vb.) geliştirilmiştir. Bu ağ yapıları içerisinde en yaygın kullanım alanı bulan ve çalışmamızda da kullanılan çok katmanlı ileri beslemeli yapay sinir ağları (MLP, Multiple Layer Perceptron) ile radyal tabanlı fonksiyon (RBF, Radial Basis Function) ağlarıdır [7].

## 2.1. Tarihsel Gelişimi

İnsanoğlu tarih boyunca sürekli insan beyninin nasıl çalıştığını merak etmiştir. Bilgisayarın doğmasında aslında bu merakın bir neticesidir. İlk hesap makinelerinden günümüzdeki çok karmaşık bilgisayar sistemlerine geçişin temelinde bu merak ve arayışın rolünü unutmamak gerekmektedir. Gelişmelere bakarak gelecekte daha karmaşık sistemlerin çıkacağını da kestirmek mümkün değildir. Bilgisayarlar başlangıçta sadece aritmetik işlemler yapmak amacı ile geliştirilmiş iken, bugün olayları öğrenmeleri ve çevre şartlarına göre karar vermeleri istenmektedir. Gelecekte insanoğlunun gerçekleştirdiği çok yüksek oranda beyin gücü gerektiren işleri yapmalarının bekleneceğini kestirmek zor değildir. Yapay sinir ağları günümüzde bu gelişmeyi tetikleyen bilim dallarından birisidir. Gelecekte de yine en önemli bilim dallarından birisi olacaktır.

1890 yılında beynin fonksiyonları hakkında bilgi veren ilk eser yayınlanmıştır. 1940'dan önceki yıllarda bazı bilim adamlarının (Helmholtz, Pavlov, Poincare vb.) yapay sinir ağı kavramı üzerinde çalıştıkları bilinmektedir. 1940'lı yıllardan sonra Hebb, McCulloch ve Pitts gibi bilim adamları yapılan araştırmaları mühendislik alanlarına kaydırmaya ve günümüzdeki yapay sinir ağlarının temellerini oluşturmaya başladılar. İlk yapay sinir hücresinin yapısını oluşturdular.

1950'li yılların sonlarında, büyük ölçekli işlemcilerin geliştirilmesiyle, beynin yaptığı işlemleri yapabilecek sinir ağlarının oluşturulabilmesi mümkün hale gelmiştir [16].

YSA simülasyonları, nispi olarak yeni bir gelişme olarak görülmektedir. Bununla beraber, bu alan bilgisayarın çıkışından önce ortaya çıkmıştır ve bir bocalama devresi geçirdikten sonra yoluna devam etmiştir.

Bilgisayarların yaygın bir şekilde kullanılmaya başlanmasıyla birlikte, YSA alanında oldukça önemli gelişmeler olmuştur. Bu alandaki araştırmalar ve çalışmalar büyük bir ilgi ile başlamış fakat beklenen gelişmelerin gerçekleşmemesi sonucunda ilgi azalmış ve bir suskunluk dönemi yaşanmıştır. Profesyonel ve maddi katkının en az olduğu bu dönemde, sadece birkaç araştırmacı tarafından katkı sağlanmıştır. Bu araştırmacılar, Minsky ve Papert tarafından tanımlanan sınırlamaları etkisiz kılan bir teknoloji geliştirmişlerdir. Minsky ve Papert, 1969 yılında bir kitap yayınlamışlar ve bu kitapta, araştırmacılar arasında ön plana çıkan ve ekstra analiz yapılmadan kabul gören YSA'na karşı bazı olumsuzlukları toplamışlardır. Son yıllarda ise, YSA alanı ilgi ve katkı olarak yeniden canlanmaktadır. YSA tarihi, dönemler itibariyle incelenebilir [4].

## 2.2. Yapay Sinir Ağlarının Yapısı

### 2.2.1. Biyolojik sinir ağları

Biyolojik sinir ağları beynimizde bulunan bir çok sayıda sinir hücresinin bir koleksiyonudur. Bir sinir ağı milyarlarca sinir hücresinin bir araya gelmesi ile oluşmaktadır. Sinir hücreleri birbirleri ile bağlanarak fonksiyonlarını yerine getirirler. Beynimizde $10^{10}$ adet sinir hücresi ve bunların da $6X10^{13}$'ten fazla sayıda bağlantısının olduğu söylenmektedir. İnsan beyni, çok hızlı çalışabilen mükemmel bir bilgisayar gibi görülebilir. Bir grup insan resmi içinden tanıdık bir resmi 100-200 ms gibi kısa bir sürede fark edilebilir. Halbuki geleneksel bilgisayarların böyle bir tanıma işlemini yapması çok daha uzun zaman alabilir. Bugün insan beyninin kapasitesinin çok küçük bir oranında kapasiteye sahip ve çalışabilen bir makine yapılsa olağanüstü bilgi işleme ve kontrol edebilme mekanizmaları geliştirmek ve mükemmel sonuçlar elde etmek mümkün olabilir. Biyolojik sinir ağlarının

performansları küçümsenemeyecek kadar yüksek ve karmaşık olayları işleyebilecek yetenektedir. Yapay sinir ağları ile bu yeteneğin bilgisayara kazandırılması amaçlanmaktadır.

Biyolojik sinir ağları insan beyninin çalışmasını sağlayan en temel taşlardan birisidir. İnsanın bütün davranışlarını ve çevresini anlamasını sağlarlar. Biyolojik sinir ağları beş duyu organından gelen bilgiler ışığında geliştirdiği algılama ve anlama mekanizmalarını çalıştırarak olaylar arasındaki ilişkileri öğrenir [16].

Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin bulunduğu üç katmanlı bir sistem olarak açıklanmaktadır. Bu katmanlar; çevreden gelen girdileri elektriksel sinyallere dönüştürerek beyine ileten Alıcı Sinirler (Receptor), beynin ürettiği elektriksel sinyalleri çıktı olarak uygun tepkilere dönüştüren Tepki Sinirleri ile alıcı ve tepki sinirleri arasında ileri ve geri besleme yaparak uygun tepkiler üreten Merkezi Sinir Ağı olarak sıralanır [17].

### 2.2.2. Yapay sinir hücresi modeli

Biyolojik sinir ağlarının sinir hücreleri olduğu gibi yapay sinir ağlarının da yapay sinir hücreleri vardır. Yapay sinir hücreleri mühendislik biliminde proses elemanları olarak da adlandırılmaktadır.

Bir yapay sinir ağı, birbiriyle bağlantılı çok sayıda yapay sinir hücresinden meydana gelmektedir. Yapay sinir hücreleri biyolojik sinir hücrelerinin basit bir modelidir.

Tablo 2.1. Biyolojik sinir ağı ve yapay sinir ağının karşılaştırılması

| Sinir Hücresi (Nöron) | İşlemci Eleman (Yapay Sinir Hücresi, Düğüm) |
|---|---|
| Sinaps | İşlemci elemanlar arasındaki bağlantı ağırlıkları |
| Biyolojik Sinir Ağı | Yapay Sinir Ağı |
| Sinir Sistemi | Sinirsel Hesaplama Sistemi |
| Dendrit | Toplama Fonksiyonu |
| Hücre Gövdesi | Aktivasyon Fonksiyonu |
| Akson | İşlemci Eleman çıktısı |

Yapay sinir ağlarının içinde bulunan tüm sinir hücreleri bir veya birden fazla girdi alırlar ve tek bir çıktı verirler. Bu çıktı yapay sinir ağının dışına verilen bir çıktı olabileceği gibi başka bir yapay sinir hücresine girdi olarak da verilebilir. Bir yapay sinir hücresi genel olarak beş temel bileşenden oluşmaktadır.

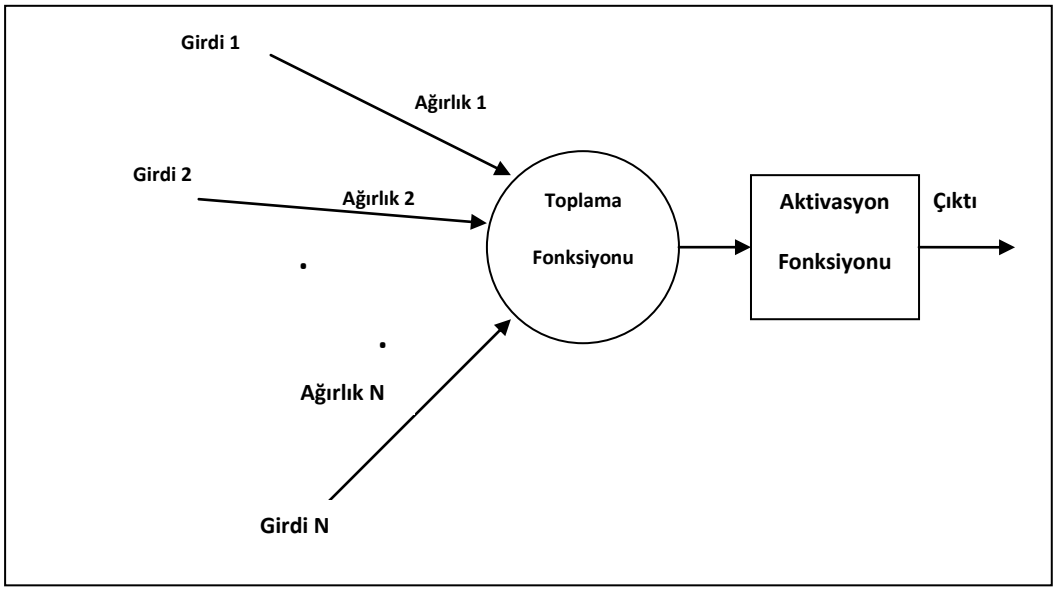
• Girdiler

• Ağırlıklar

• Toplama fonksiyonu

• Aktivasyon fonksiyonu

• Çıktı




Şekil 2.2. Yapay sinir hücresinin yapısı [18]

### 2.2.2.1. Girdiler

Girdiler, bir yapay sinir hücresine gelen bilgilerdir. Bu bilgiler dış ortamlardan ya da diğer sinir hücrelerinden gelebilir. Dış ortamlardan gelen bilgiler, ağın öğrenmesi istenen örnekler tarafından belirlenmektedir.

#### 2.2.2.2. Ağırlıklar

Ağırlıklar, gelen bilgilerin hücre üzerindeki etkisini belirleyen değerlerdir. Bilgiler, bağlantılar üzerindeki ağırlıklar üzerinden hücreye girmekte ve ağırlıklar yapay sinirde girdi olarak kullanılacak değerlerin göreceli kuvvetini (matematiksel katsayısını) göstermektedirler. YSA içinde girdilerin hücreler arasında iletimini sağlayan tüm bağlantıların farklı ağırlık değerleri bulunur. Böylelikle ağırlıklar her işlemci elemanın her girdisi üzerinde etki yapmış olur. Ağırlıklar değişken veya sabit değerler olabilirler.

#### 2.2.2.3. Toplama fonksiyonu

Toplama fonksiyonu, hücreye gelen net girdiyi hesaplayan fonksiyondur ve genellikle girişlerin kendi ağırlıklarıyla çarpımının toplamı Eşitlik (2.1)'deki gibi ifade edilir.

$$F_{net} = \sum x_i w_i \tag{2.1}$$

$X_i$ hücreye gelen girdileri, $W_i$ ise girdilerin kendi ağırlıklarını ifade etmektedir. Yapay sinir ağının yapısına göre toplama fonksiyonu, maksimum, minimum, çarpım veya çeşitli normalizasyon işlemlerinden birisi olarak da ifade edilebilir. Bir problem için en uygun toplama fonksiyonu çeşidini bulmak için herhangi bir formül yoktur. Toplama fonksiyonu genellikle deneme yanılma yoluyla bulunmaktadır. Ayrıca bir yapay sinir ağındaki bütün işlemci elemanların aynı toplama fonksiyonuna sahip olması gibi bir zorunluluk da yoktur. Bazen aynı yapay sinir ağı içindeki işlemci elemanların bazıları aynı toplama fonksiyonunu, diğerleri ise başka fonksiyonları kullanabilirler. Bu tamamen tasarımcının kendi kararına bağlıdır.

Tablo 2.2. Toplama fonksiyonları

| **Net Giriş** | **Açıklama** |
|---|---|
| Çarpım<br>Net Girdi= $\Pi G_i W_i$<br>i | Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleri ile çarpılarak NET girdi hesaplanır. |
| Maksimum<br>Net Girdi= Max ($G_i W_i$) , i=1…N | N tane girdi içinden ağırlıklar ile çarpıldıktan sonra en büyüğü yapay sinir hücresinin NET girdisi olarak kabul edilir. |

Tablo 2.2. (Devam)Toplama fonksiyonları

| **Net Giriş** | **Açıklama** |
|---|---|
| Minimum<br>Net Girdi= Min ($G_iW_i$) , i=1…N | N tane girdi içinden ağırlıklar ile çarpıldıktan sonra en küçüğü yapay sinir hücresinin NET girdisi olarak kabul edilir. |
| Çoğunluk<br>Net Girdi= $\Sigma_i$ sgn ($G_iW_i$) | N tane girdi içinden ağırlıklar ile çarpıldıktan sonra pozitif ve negatif olanların sayısı bulunur. Büyük olan sayı hücrenin NET girdisi olarak kabul edilir. |
| Kümülatif Toplam<br>Net Girdi= Net(eski) + $\Sigma_i$ ($G_iW_i$) | Hücreye gelen bilgiler ağırlıklı olarak toplanır ve daha önce gelen bilgilere eklenerek hücrenin net girdisi bulunur. |

### 2.2.2.4. Aktivasyon fonksiyonu

Aktivasyon fonksiyonu, toplama fonksiyonundan gelen girdiyi işleyerek yapay sinir hücresinin çıkışını belirler. Transfer fonksiyonu olarak da adlandırılan aktivasyon fonksiyonu çeşitli tiplerde ve genellikle doğrusal olmayan bir fonksiyondur. Doğrusal fonksiyonların tercih edilmemesinin nedeni, doğrusal fonksiyonlarda girdi ile çıktının doğru orantılı olmasıdır. Bu durum ilk yapay sinir ağları denemelerinin başarısızlıkla sonuçlanmasının temel nedenidir.

Uygun aktivasyon fonksiyonunun seçimi tasarımcının farklı fonksiyonları denemeleri sonucunda belirlenmektedir. Ancak çok katmanlı perceptron gibi bazı modeller aktivasyon fonksiyonunun, türevi alınabilir bir fonksiyon olmasını şart koşmaktadır. Ayrıca fonksiyonun seçimi, yapay sinir ağının verilerine ve neyi öğrenmesinin istendiğine de bağlıdır. Aktivasyon fonksiyonu olarak en çok kullanılanlar sigmoid fonksiyon ve hiperbolik tanjant fonksiyonlarıdır.

Aktivasyon fonksiyonu, toplama fonksiyonundan gelen girdiyi dönüştürerek istenilen değerler arasında sınırlandırmaktadır. Bu değerler kullanılan aktivasyon fonksiyonun tipine göre genellikle [0,1] veya [-1,1] arasındadır. Bu değer aktivasyonun fonksiyonunun, dolayısıyla yapay sinir hücresinin çıktı değeri olarak ya dış ortama ya da girdi olarak başka bir yapay sinir hücresine iletilmektedir.

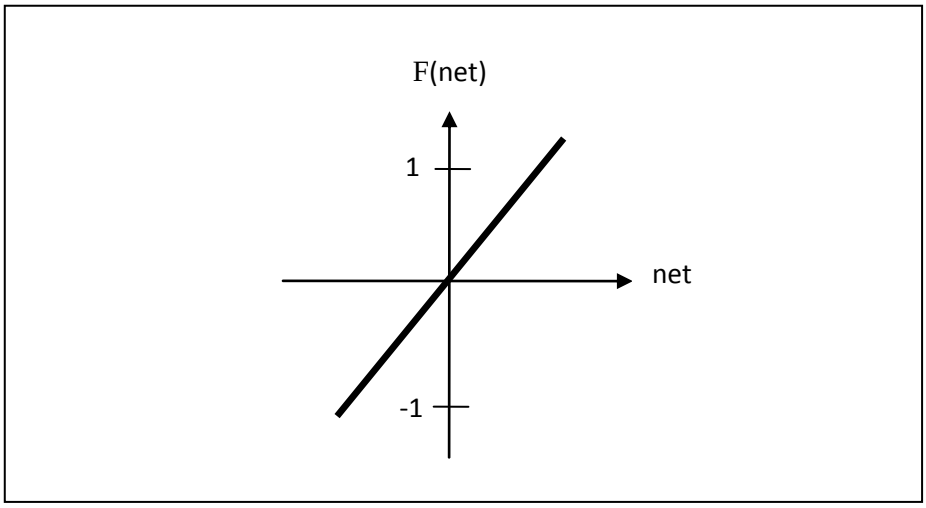
Aktivasyon fonksiyonu işlemi öncesinde, sisteme tekdüze (uniform) dağılmış bir rassal hata eklenebilmektedir. Bu rassal hatanın kaynağı ve büyüklüğü sistemin öğrenme sürecinde belirlenir ve sebebi, insan beyninin işlevinin, içinde bulunduğu ortamın koşullarından etkilenmesidir. Örneğin ortamın soğuk/sıcak olmasından insan beyni etkilenmektedir. Bu nedenle yapay sinir ağları literatüründe rassal hata ekleme işlemi "sıcaklık (temperature)" olarak da adlandırılmaktadır. Ancak günümüzde rassal hata işlevi tam olarak kullanılmamakta ve hala bir araştırma süreci içinde bulunmaktadır. Ayrıca bazı yapay sinir ağlarında, aktivasyon fonksiyonunun çıktısı üzerinde başka işlemler, ölçeklendirme ve sınırlandırma yapılabilmektedir [18].

Tablo 2.3. Aktivasyon fonksiyonları

| **Aktivasyon Fonksiyonu** | **Açıklama** |
|---|---|
| Doğrusal fonksiyon<br>F(NET )= NET | Gelen girdiler olduğu gibi hücrenin çıktısı olarak kabul edilir. |
| Step fonksiyonu<br>F(NET )= { 1 if NET > eşik değer ; 0 if NET ≤ eşik değer | Gelen NET girdi değerinin belirlenen bir eşik değerinin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır. |
| Sinüs fonksiyonu<br>F(NET )= Sin(NET ) | Öğrenilmesi düşünülen olayların sinüs fonksiyonuna uygun dağılım gösterdiği durumlarda kullanılır. |
| Hiperbolik tanjant fonksiyonu<br>F(NET )= $(e^{NET} + e^{-NET}) / (e^{NET} – e^{-NET})$ | Gelen NET girdi değerinin tanjant fonksiyonundan geçirilmesi ile hesaplanır. |

a) Doğrusal aktivasyon fonksiyonu

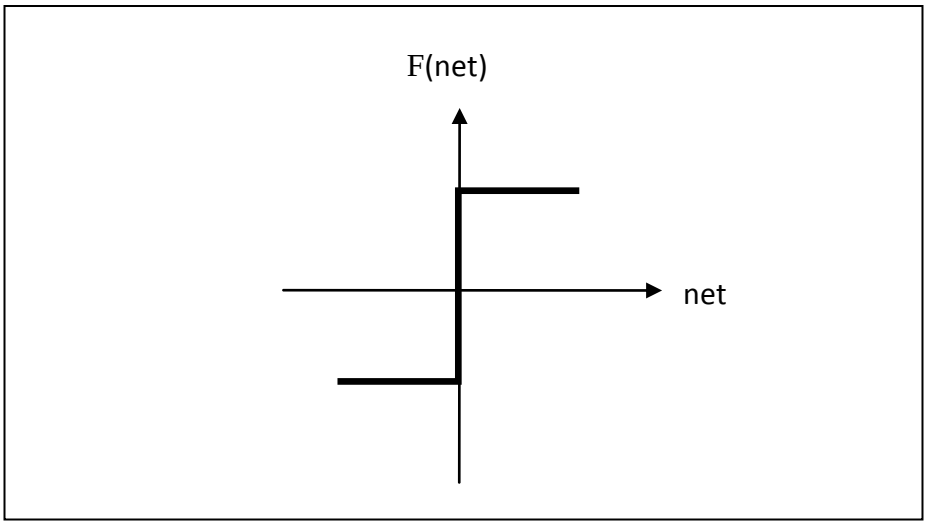
Doğrusal problemlerin çözümünde kullanılan bu fonksiyon, gelen net girdileri doğrudan hücre çıkışı olarak vermektedir. Matematiksel olarak F(net)= net şeklinde tanımlanmaktadır. Şekil 2.3'te gösterildiği gibi.




Şekil 2.3. Doğrusal aktivasyon fonksiyonunun şekilsel gösterimi [18]

b) Adım fonksiyonu

Gelen net girdi değerinin belirlenen bir eşik değerinin altında ya da üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerlerini almaktadır. Şekil 2.4'te gösterilmektedir.




Şekil 2.4.Adım fonksiyonunun şekilsel gösterimi [18]

c) Sigmoid aktivasyon fonksiyonu

En önemli eşik fonksiyonu Sigmoid fonksiyonudur. Bu fonksiyon, seviyeli lineer olmayan çıkış veren, sınırlı, monoton artan, türevi alınabilen bir fonksiyondur.

Eşitlik (2.2);

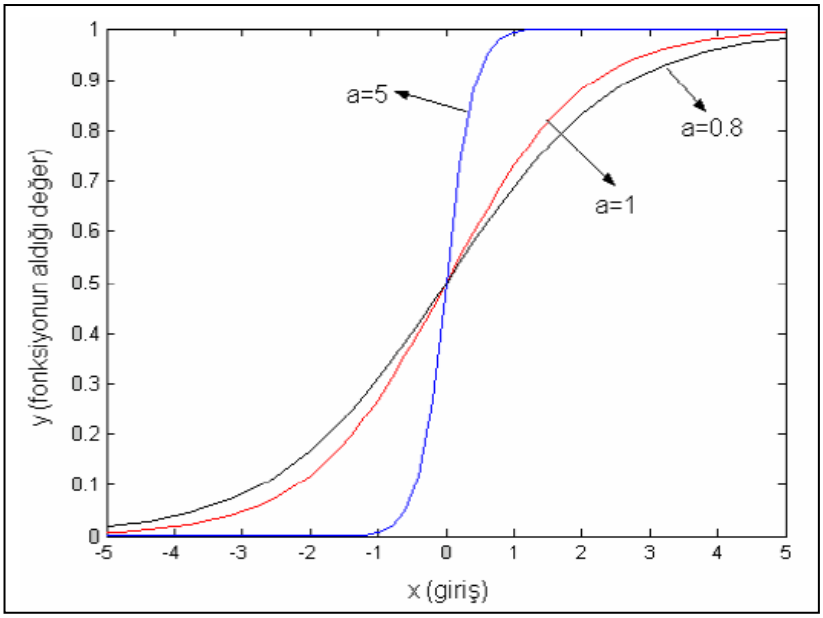
$$y(x)=\frac{1}{1+e^{-ax}} \tag{2.2}$$

şeklindedir. YSA çıktısı (2.3);

$$y(x)=\frac{1}{1+e^{-aynet}} \tag{2.3}$$

şeklinde olur.

Sigmoid fonksiyonuna a kazancı ilave edilmesiyle şekli değiştirilebilir. a eğim parametresidir. a'nın değişik değerleri için bulunan sigmoid fonksiyonları Şekil 2.5'te gösterilmiştir. Eğer a çok artırılırsa Sigmoid fonksiyonu basamak fonksiyonuna dönüşür. Her iki fonksiyonda [0,1] aralığında değişir fakat Sigmoid fonksiyonunun türevinin alınabilmesi YSA teorisi ve özellikle Geriye Yayılım (Back-Propagation) teorisinde çok önemlidir.



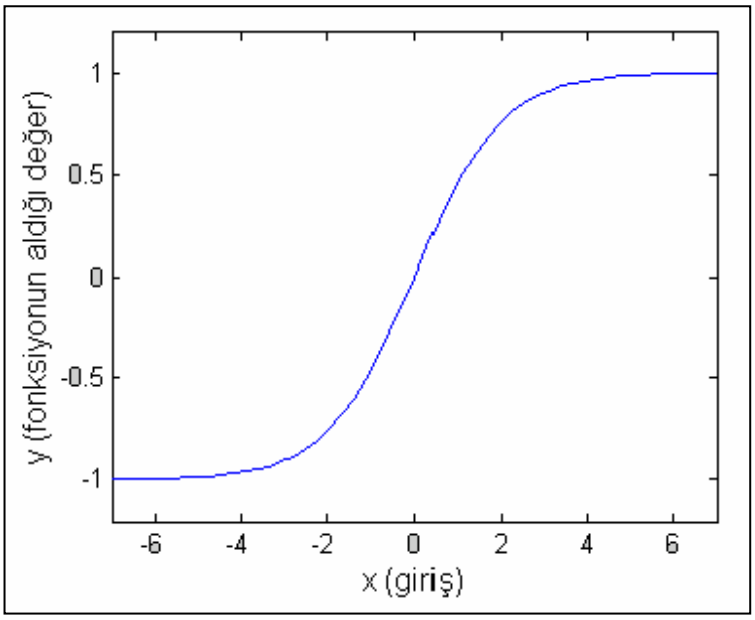

Şekil 2.5. Sigmoid fonksiyonu [4]

d) Hiperbolik tanjant fonksiyonu

Hiperbolik tanjant fonksiyonu, gelen net girdinin tanjant fonksiyonundan geçirilmesi ile hesaplanmaktadır ve sigmoid aktivasyon fonksiyonunun farklı bir çeşididir. Sigmoid aktivasyon fonksiyonunda çıktı 0 ile 1 arasında bir değer alırken, hiperbolik

tanjant fonksiyonunda çıktı -1 ile 1 arasındadır ve Eşitlik (2.4)' deki gibi hesaplanır [18].

$$y=F_{(s)}=\frac{e^{s}+e^{-s}}{e^{s}-e^{-s}} \tag{2.4}$$




Şekil 2.6. Hiperbolik tanjant fonksiyonu [4]

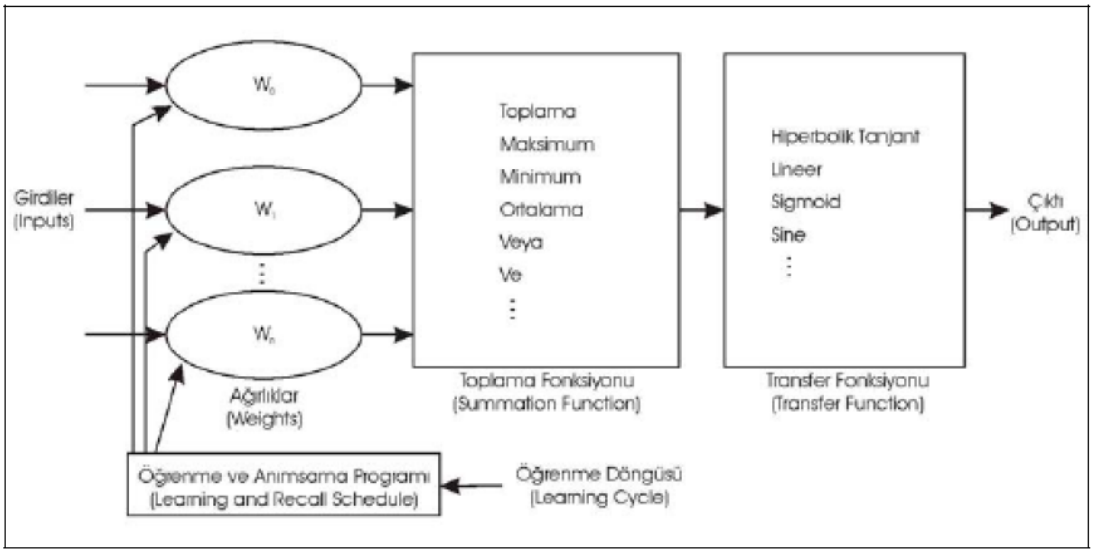
e) Hücrenin çıktısı

Aktivasyon fonksiyonu tarafından belirlenen çıktı değeridir. Bu değer ya başka bir yapay sinir hücresine girdi olarak ya da dış ortama gönderilmektedir. Bir işlemci elemanın birden fazla girdisi olmasına rağmen tek bir çıktısı olmaktadır [16].

Tüm YSA'lar, yukarıda temel elemanları anlatılan bu temel yapı taşlarından yani nöronlardan oluşturulurlar. Bu yapı taşlarının dizaynı, sinir ağı sanatının, başka bir deyişle mimarisinin oluşturulmasının ilk bölümüdür. Bu sanatın ikinci bölümü ise bu işlem elemanlarının kümelendirilmesi ve birbirleri arasındaki bağlantıların oluşturulmasını içerir. Beyinde kümelendirme, bilginin dinamik, etkileşimli ve kendiliğinden organize bir şekilde işlenmesini sağlayacak şekildedir. Biyolojik sinir ağları üç boyutlu uzayda mikroskobik elemanlarla oluşturulur. Bu nöronlar hemen hemen sınırsız sayıda bağlantılar içerirler. Bu, yapay sinirler için mümkün değildir.

Bugünkü teknoloji ile iki boyutlu ortamda ve belirli sayıda bağlantı içeren nöronlar oluşturulabilmektedir. Bu durum, YSA'nın yetenek ve çeşitlerini sınırlamaktadır [4].

Şekil 2.7'de, daha önce tanımlanan basit bir yapay nöron yapısının daha detaylı bir şeması gösterilmiştir. Şekilde, girdi değerleri işlem elemanına üst sol bölümden girmektedir. İşlemde ilk adım, bu girdi değerlerin her birinin ilgili ağırlıklarla w(i) ağırlıklandırılmalarıdır. Bir nöron genellikle, eş anlı olarak birçok sayıda girdi alır. Her girdinin kendi nispi ağırlığı vardır. Bu ağırlıklar, biyolojik nöronların değişen sinaptik etkililikleri ile aynı görevi üstlenirler. Her iki durumda da, bazı girdiler diğerlerine göre daha önemli hale gelir. Bu sayede, işlem elemanının bir sinirsel tepki üretmesi işleminde daha fazla etkili olurlar. Ayrıca, ağırlıklar girdi sinyalinin güçlülüğünü belirleyen adaptif katsayılardır. Yani, girdinin bağlantı gücünün bir ölçüsüdür. Bu bağlantı güçleri, çeşitli eğitme setlerine göre değiştirilebilirler [19].




Şekil 2.7. Yapay nöronun detaylı yapısı [19]

Ağırlıklandırmadan sonra, bu değiştirilmiş girdiler toplama fonksiyonuna gönderilirler. Toplama fonksiyonunda, adından da anlaşılacağı gibi, genelde toplama işlemi yapılmaktadır. Fakat birçok farklı işlem çeşidi toplama fonksiyonu için kullanılabilir. Toplama fonksiyonu, bu basit çarpımlar toplamına ek olarak, minimum, maksimum, mod, çarpım veya çeşitli normalizasyon işlemlerinden birisi olabilir. Girdileri birleştirecek olan algoritma genellikle seçilen ağ mimarisine de bağlıdır. Bu fonksiyonlar farklı şekilde değerler üretebilir ve sonra bu değerler ileri doğru gönderilir. Ek olarak, uygulamacı kendi fonksiyonunu oluşturup toplama

fonksiyonu olarak kullanabilir. Bazı toplama fonksiyonları, transfer fonksiyonuna iletmeden önce, sonuçlar üzerinde ilave işlemler yaparlar. Bu işlem aktivasyon fonksiyonu olarak adlandırılan işlemdir. Bir aktivasyon fonksiyonu kullanmanın amacı, toplama fonksiyonu çıktısının zamana bağlı olarak değişmesini sağlamaktır. Fakat, aktivasyon fonksiyonu literatürü henüz tam olarak gelişmemiştir. Bundan dolayı, çoğu yapay sinir ağında birim aktivasyon fonksiyonu kullanılmaktadır. Birim aktivasyon fonksiyonu ise bir aktivasyon fonksiyonu kullanılmaması ile aynı anlama gelmektedir. Ayrıca, aktivasyon fonksiyonu, her işlem birimi için ayrı ayrı kullanılan bir bileşenden ziyade ağın genel bir bileşenidir. Yani, oluşturulan bir ağ yapısında, tüm işlem elemanları aynı aktivasyon fonksiyonunu kullanırlar [19].

Sonraki aşamada toplama fonksiyonunun çıktısı transfer fonksiyonuna gönderilir. Bu fonksiyon, aldığı değeri bir algoritma ile gerçek bir çıktıya dönüştürür. Transfer fonksiyonu genellikle doğrusal olmayan bir fonksiyondur. Doğrusal fonksiyonlar genelde tercih edilmez, çünkü doğrusal fonksiyonlarda çıktı, girdi ile orantılıdır. Bu durum, ilk YSA denemelerinin başarısızlıkla sonuçlanmasının temel nedenidir [4].

Transfer fonksiyonu işlemi öncesinde, sisteme uniform dağılmış bir rassal hata eklenebilmektedir. Bu rassal hatanın kaynağı ve büyüklüğü, ağın öğrenme işlemi sürecinde belirlenir. Sisteme böyle bir hata teriminin eklenmesinin sebebi, insan beyninin işlevinin içinde bulunduğu ortamın şartlarından (örnek olarak sıcak/soğuk olmasından) etkileniyor olmasıdır. Bu yüzden, YSA literatüründe rassal hata ekleme işlemi sıcaklık olarak da adlandırılmaktadır. Günümüzde, rassal hata kullanımı fiilen tam olarak yerleşmemiştir ve halen bir araştırma süreci içerisindedir. Ayrıca, bazı ağlarda, transfer fonksiyonunun çıktısı üzerinde başka işlemler, ölçeklendirme ve sınırlandırma yapılabilmektedir.

Transfer fonksiyonundan çıkan değer işlem elemanının da çıktısıdır. Fakat, bazı durumlarda işlem elemanının bu çıktıyı, bir çıktı fonksiyonu ile dönüşüme uğratması gerekebilmektedir[19].

## 2.3.Yapay Sinir Ağlarının Kullanım Alanları

Yapay sinir ağlarının kullanılabileceği birçok alan vardır. Dolayısıyla yapay sinir ağlarının kullanıldığı ve başarılı sonuçlar elde edilen yüzlerce uygulama sayılabilir. Örneğin, 1997 yılında Caere firması tarafından üretilen optik karakter okuma sistemi yılda 3 milyon$'dan fazla gelir getirmiştir. Aynı yıl HNC firması tarafından pazarlanan ve kredi kartlarının haksız yere kullanılmasını ortaya çıkartan Falcon isimli yapay sinir ağı sistemi yılda 23 milyon$ kar sağlamıştır. 1998 yılında Sensory firması tarafından geliştirilen ses tanıma sistemindeki yonganın 5$'a mal olduğu ve bir milyondan fazla sattığı bilinmektedir.

Bu örneklerin çoğaltılması mümkündür. Fakat herhangi bir problemin çözümü için yeterli etkinlikte ve verimlilikte bir yöntem varsa, bu problemi yapay sinir ağı ile çözmek mantıklı olmayabilir. Başarılı uygulamalar incelendiğinde, yapay sinir ağlarının, doğrusal olmayan, çok boyutlu, gürültülü ve hata olasılığı yüksek sensör verilerinin bulunduğu, kesin olmayan, karmaşık, verilerin eksik olduğu, ortaya çıkan çözümde matematiksel modele ve algoritmaya ihtiyaç duyulmayan hallerde yaygın olarak kullanıldıkları görülmektedir [20].

Yapay sinir ağları çeşitli alanlarda kompleks problemlerin çözümünde yaygın olarak kullanılmakta olup örüntü tanıma, sınıflandırma ve kontrol sistemlerinde basarı ile uygulanmaktadırlar. Geleneksel bilgisayarlar ve insanlar için oldukça zor olan problemleri çözmek için eğitilebilmektedirler. Diğer taraftan istenilen bilgiyi verilerden doğrudan elde edebildikleri için geleneksel yaklaşımların sınırlamasının üstesinden gelebilirler.

Yapay sinir ağları, yaygın olarak kullanılan çok değerli, lineer olmayan sistemlerdir. YSA yazılım teknikleri özellikle gerçek zamanlı fonksiyonların kullanıldığı problemlerde (belirsizlik isleme, sensor tümleme gibi) uygulanabilir çözümler önerebilirler.

YSA esnekliği ve performansı nedeniyle mekanik sistemlerdeki yeni teknolojilerin etkisini analiz etmede yaygın olarak kullanılırlar ve analiz için gereken zamanı azaltırlar. Dizayn nesnelerinin optimizasyonunda önemli gelişimi sağladığı gibi, dizayn nesneleri arasında zıtlık olduğunda dahi hem tekli hem çoklu nesneler için

yakın gerçek zamandaki sistem çıktısının tahminini sağlar. YSA'nın kullanımı yakın gerçek zamanlı karmaşık sistemlerin tasarım analiz araçlarının kabul edilebilir toleranslarla simülasyonunu olanaklı kılar. YSA herhangi bir konuda çözüm bulmanın en iyi ikinci yoludur. En iyi yöntem, problemi tam olarak kavrayarak doğru formülü ve en iyi algoritmayı bulmaktır. Fakat bu her zaman mümkün olmamaktadır ve çoğu problem en iyi 2. yaklaşıma başvurularak çözülmektedir. YSA, doğrusal olmayan, çok boyutlu, karmaşık, kesin olmayan, eksik, kusurlu, hata olasılığı yüksek verilerin olması ve problemin çözümü için özellikle bir matematik modelin ve algoritmanın bulunmaması hallerinde yaygın olarak kullanılmaktadırlar. YSA, çok değişken ve karmaşık etkileşim içeren üretim prosesleri için mükemmel sistemlerdir [2].

a) Yapay sinir ağlarının kullanıldığı teorik uygulamalar

• Doğrusal olmayan sistem modelleme

• Akıllı kontrol

• Sinyal filtreleme ve doğrusal olmayan sinyal işleme

• Optimizasyon

• Probabilistik fonksiyon kestirimleri

• Sınıflandırma

• Örüntü tanıma, ilişkilendirme ve eşleştirme

b) Yapay sinir ağlarının kullanıldığı pratik uygulamalar

• Kimyasal proseslerin modellenmesi

• Akıllı araçlar ve robotlar için optimum rota belirleme

• Robotlarda görme sistemlerinin ve hareket mekanizmalarının kontrol edilmesi

• Makina, sistem ve süreçler için arıza tespiti

• İletişim kanallarındaki ekoların filtrelenmesi, anahtarlama ve yoğunluğun kontrolü

• Hedef tanıma ve takip sistemleri

• Radar ve sonar sinyallerinin sınıflandırılması

• Radar ve görüntü sinyalleri işleme

• Güvenlik sistemlerinde konuşma ve parmak izi tanıma

• Mekanik parçaların ömürlerinin ve kırılmalarının tahmin edilmesi

• Endüstriyel ürünlerin görsel kalite kontrolü ve imalatta meydana gelen hataların tespiti

• Kredi kartı hilelerinin tespiti

• Döviz kuru tahminleri, risk analizleri

### 2.4. Yapay Sinir Ağlarının Avantajları

Gerçek dünyada olaylar birçok parametreye bağlı olabilir. Ayrıca bu parametrelerin birbirleri üzerinde, açıkça görülemeyen ilişkileri ve etkileri olabilir. Geleneksel yöntemler kullanıldığında bu ilişkileri belirlemek, eğer belirlenemiyorsa yok saymak gerekmektedir. Oysaki YSA kullanıldığında, kullanıcının bu ilişkileri bilmesi ve ağa söylemesi beklenmemektedir. Örneklerin dışında herhangi bir ön bilgiye ihtiyaç yoktur. YSA, bu ilişkileri, örnekler yardımıyla kendisi öğrenir.

Olayların bağlı olduğu parametrelerin tümü bilinse bile, parametreler arasındaki ilişkiler doğrusal olmayabilir. Geleneksel yöntemler kullanıldığında, varsayımlarla çalışma noktaları civarında lineerleştirmeler yapılır. Bu durum, oluşturulan modelin, gerçek sisteme olan uygunluğunu azaltmakta ve gerçek sistemin kontrolünü zorlaştırmaktadır. YSA'da ise ilişkilerin doğrusal olup olmaması problem teşkil etmez [20].

YSA' nın temel işlem elemanı olan hücre doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özelliği ile YSA, doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmuştur.

Alışılmış bilgi işlem yöntemlerinin çoğu seri işlemlerden oluşmaktadır. Bu da hız ve güvenilirlik sorunlarını beraberinde getirmektedir. Seri bir işlem gerçeklenirken herhangi bir birimin yavaş oluşu tüm sistemi doğruca yavaşlatırken, paralel bir sistemde yavaş bir birimin etkisi çok azdır. Nitekim seri bir bilgisayarın bir işlem elemanı beyine göre binlerce kez daha hızlı işlemesine rağmen, beynin toplam işlem hızı seri çalışan bir bilgisayara göre kıyaslanamayacak kadar yüksektir [17].

YSA, yeni bilgilerin ortaya çıkması ve sistemde bazı değişikliklerin olması durumunda yeniden eğitilebilirler, hatta bazı ağların eğitilmesine bile gerek yoktur.

Bilgilerin eksik olduğu durumlarda, YSA etkin çözümler üretebilmektedir. Ayrıca YSA'nın belirsiz bilgileri işleyebilme yetenekleri vardır.

YSA hata toleransına sahiptir. Ağın bazı hücrelerinin bozulması ve çalışamaz duruma gelmesi halinde bile ağ çalışmaya devam eder. Fakat ağın bozuk olan hücrelerinin önemine göre performansında düşmeler görülebilir.

YSA paralel çalışabilmeleri, gerçek zamanlı kullanımlarını kolaylaştırmaktadır [20].

### 2.5. Yapay Sinir Ağlarının Dezavantajları

Yapay sinir ağlarının donanım bağımlı çalışmaları önemli bir sorun olarak görülebilir. Ağların temel varoluş nedenlerinden birisi de paralel işlemciler üzerinde çalışabilmeleridir. Ağların özellikle, gerçek zamanlı bilgi işleyebilmeleri paralel çalışabilen işlemcilerin varlığına bağlıdır günümüzdeki makinelerin çoğu seri şekilde çalışabilmekte ve aynı zamanda sadece tek bir bilgiyi işleyebilmektedir. Paralel işlemleri seri makinelerde yapmak ise zaman kaybına yol açmaktadır. Bunun yanı sıra bir ağın nasıl oluşturulması gerektiğini belirleyecek kuralların olmaması da başka bir dezavantajdır. Her problem farklı sayıda işlemci gerektirebilir. Bazı problemleri çözebilmek için gerekli olan paralel olan işlemcilerin tamamını bir arada (paralel olarak) çalıştırmak mümkün olmayabilir.

Probleme uygun ağ yapısının belirlenmesi genellikle deneme yanılma yoluyla yapılmaktadır bu ise önemli bir problemdir. Çünkü eğer problem için uygun bir ağ oluşturulmazsa çözümü olan bir problemin çözülememesi veya performansı düşük çözümlerin elde edilmesi söz konusu olabilir. Bu aynı zamanda bulunan çözümün en

iyi çözüm olduğunu da garanti etmez. Yani yapay sinir ağları kabul edilebilir çözümler üretebilir. Optimum (en iyi) çözümü garanti etmez.

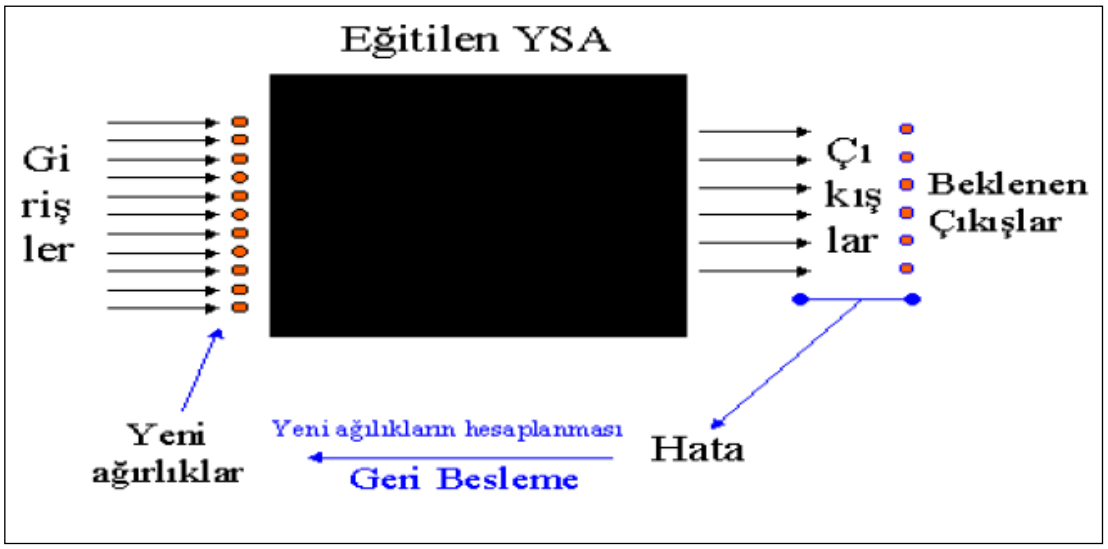
Ağın öğreneceği problemin ağa gösterimi de çok önemli bir problemdir. YSA, sadece nümerik bilgiler ile çalışmaktadırlar. Problemin nümerik gösterime dönüştürülmesi gerekmektedir. Bu ise kullanıcının becerisine bağlıdır. Uygun bir gösterim mekanizmasının kurulamamış olması problemin çözümünü engelleyebilir veya düşük performanslı bir öğrenme (çözüm) elde edilebilir. Problemin nümerik gösterimi mümkün olsa bile bunun ağa gösteriliş şekli problemin başarılı bir şekilde çözülmesini yakından etkiler.

Ağın eğitiminin ne zaman bitirileceğine karar vermek içinde geliştirilmiş bir yöntem yoktur. Ağın örnekler üzerindeki hatasının belirli bir değerin altına indirilmesi eğitimin tamamlanması için yeterli görülmektedir. Fakat neticede optimum öğrenmenin gerçekleştiği söylenememektedir. Sadece iyi çözümler üretebilen bir ağ oluştu denilmektedir. Optimum neticeleri veren bir mekanizma henüz geliştirilmemiştir.

Bir diğer sorun ise, ağın davranışlarının açıklanamamasıdır. Bir probleme çözüm üretildiği zaman nasıl ve neden üretildiği konusunda bir bilgi bulmak mümkün değildir. Bu ise ağın sonucuna olan güveni azaltmaktadır [16].

## 2.6. Yapay Sinir Ağlarının Çalışma Şekli

YSA'ların iki türlü çalışma şekli vardır. Biri eğitme diğeri kullanma aşamasıdır. YSA'lar kullanma aşamasında eğitme aşamasına göre daha hızlı çalışırlar. Bir YSA'nın kullanılabilmesi için önce eğitilmesi gerekir. Bu durum Şekil 2.8'de görülmektedir.




Şekil 2.8. YSA'nın eğitilmesi [5]

Eğitme aşamasında YSA'nın içindeki düğümlerin (node) birbirine bağlantı yüzdelerini gösteren ve ağırlık (weight) diye tabir edilen değerler hesaplanır. Bu aşamada genellikle kullanılan algoritma geriye yayılma ya da geriye yansıma (Back Propagation) algoritmasıdır. Geriye yayılma algoritmasının özü; ilerleme sonucunda ortaya çıkan hatanın, geriye doğru yansıtılarak, ağırlıkların daha doğru sonuçlar verecek şekilde değiştirilip yeniden hesaplanarak düzeltilmesidir.

Eğitme aşamasında ağırlıkların hesaplanabilmesi için YSA'ya girişler ve karşılık gelen çıkışlar verilir. YSA için öğrenme bu giriş ve çıkış verileri arasında bir çeşit bağlantı kurmak diye de tanımlanabilir. Eğitme aşamasında hesaplanan bu ağırlık değerleri daha sonra sadece girişlerin verilip çıkışların hesaplanmasının istenildiği kullanma aşamasında işe yararlar. Eğitme aşamasının bir basamağı hem ilerleme hem de geri yayılma safhalarını içerirken, kullanma aşamasında sadece ilerleme işlemi uygulanır. Zaten gerçek sonuçlar bilinmediğinden hatanın hesaplanıp geri yansıtılması mümkün değildir.

Kullanma aşamasındaki algoritma eğitme aşamasına göre daha basittir. Dolayısıyla YSA'lar kullanma aşamasında eğitme aşamasına göre daha hızlı çalışırlar. Yani bir kere tam manasıyla eğitilmiş olan bir YSA eğitme aşamasında güçlükler çıkarmış olsa da, kullanma aşamasında özellikle hız açısından o kadar sorun çıkarmaz. Eğitmede de, kullanma aşamasında girişlere karşılığı istenen değerler konulur.

Eğitme asmasında hesaplanan ağırlık değerleri bu aşamada, sadece giriş değerlerinin verilip çıkışların YSA tarafından hesaplanmasında kullanılır.

Eğitme yöntemleri YSA'nın çok önemli bir aşamasıdır. YSA'nın öğrenmesi, hücreler arasındaki ağırlıkların, uygun değerlere ayarlanması anlamındadır. Eğitme ve öğrenme farklı kavramlardır. Eğitme, ağın öğrenmesi için gerçeklenen adımlardır, öğrenme ise eğitim işleminin sonucudur. Eğitme yöntemi ilgilenilen problemin özelliğine göre öğrenme kuralını YSA'ya nasıl uyarlayabileceğini belirtir [5].

### 2.6.1. Ağın eğitilmesi

Ağın kendisine gösterilen girdi örneği için beklenen çıktıyı üretmesini sağlayacak ağırlık değerleri bulunmaktadır. Başlangıçta bu değerler rastgele atanmakta ve ağa örnekler gösterildikçe ağın ağırlıkları değiştirilerek zaman içerisinde istenen değerlere ulaşması sağlanmaktadır. Ağın en az hatanın olduğu ağırlık vektörüne ulaşması istenmektedir. O nedenle her iterasyonda ağırlık değerleri değiştirilerek hatanın düşmesi sağlanır. Bazen ağ farklı bir çözüme takılabilmekte ve performansı daha da iyileştirmek mümkün olmamaktadır. Böyle durumlarda kullanıcı tarafından belli bir tolerans değeri kadar hata kabul edilebilmektedir. Bazı durumlarda ağın takıldığı lokal sonuç kabul edilebilir hata düzeyinin üstünde kalabilir. Çok katmanlı ağların yerel sonuçlara takılıp kalmaması için momentum katsayısı geliştirilmiştir. Bu katsayının kullanılmasıyla yerel çözümler kabul edilebilir hata düzeyinin altına çekilebilmektedir. Ağın performansını ölçmek için, ağın eğitildiği problem üzerinden hem eğitimde kullanılacak hem de test esnasında kullanılacak örnekler seçilir. Eğitim sırasında ağa sadece eğitim setindeki örnekler gösterilir. Ağ bunları öğrenince ağa hiç görmediği test setindeki örnekler verilir. Ağın performansı bu görmediği örnekler karsısında ürettiği doğru cevaplar oranı ile ölçülür.

$$P=\frac{D}{T}x100 \tag{2.5}$$

Eşitlik (2.5)' de D test setinden doğru olarak cevaplandırılan örnek sayısını, T test setinde bulunan toplam örnek sayısını, P ise performans oranını göstermektedir [7].

YSA'ında işlemci elemanlar arasındaki bağlantıların ağırlık değerlerinin değiştirilmesi işlemine "ağın eğitilmesi" denilmektedir. Başlangıçta rastgele atanan

bu ağırlık değerleri, ağa gösterilen örneklerle değiştirilmektedir. Amaç, ağa gösterilen örnekler için doğru çıktıları üretecek ağırlık değerlerinin belirlenmesidir. Yapay sinir ağının eğitilmesinde" kullanılan girdi ve çıktı çiftlerinden oluşan verilerin tümüne "eğitim seti" adı verilmektedir.

YSA'nın eğitim süreci, belli kurallar çerçevesinde olmaktadır. Bu kurallara öğrenme kuralları adı verilmektedir. Ağırlıkların değiştirilmesi öğrenme kurallarına göre yapılır. Yapay sinir ağında ağırlıkların doğru değerlere ulaşması, örneklerin temsil ettiği problem konusunda ağın genellemeler yapabilme yeteneğine kavuşması demektir. Genelleme, yapay sinir ağının eğitiminde kullanılmamış, ancak aynı evrenden gelen girdi-çıktı örneklerini doğru sınıflandırabilme yeteneği olarak tanımlanır. Ağın bu genelleştirme özelliğine kavuşması işlemine "ağın öğrenmesi" denilir [18].

Makine öğrenmesi: "Bilgisayarların bir olay ile ilgili bilgileri ve tecrübeleri öğrenerek, gelecekte oluşacak benzeri olaylar hakkında kararlar verebilmesi ve çözümler üretebilmesidir" [16].

Ağın ağırlıklarının değiştirilmesinde iki tür strateji uygulanır.

1. Hata bilgileri toplanarak her bir iterasyon sonrasında ağırlıklar değiştirilir.

2. Ağırlıklar veri setindeki her bir veriden sonra değiştirilir.

YSA'nın eğitiminin tamamlanmasının ardından öğrenip öğrenmediğini (performansını) ölçmek için yapılan denemelere ise ağın test edilmesi denmektedir. Bazı paket programlar ağ eğitilirken bir yandan da Doğrulama Seti (Validation Set) ile ağın her iterasyonda ne kadar öğrendiğini test ederler ve öğrenim kriteri olarak doğrulama setinin hata değerlerini kullanırlar.

Eğitim setindeki verilerle istenilen başarıya ulaşılmışsa ağ, daha önce görmediği verilerle test edilir. Test işleminin yapıldığı veri setine "Test Seti" denir. Test işleminde ağın ağırlık değerleri değiştirilmemektedir. Örnekler ağa gösterilmekte ve ağ, eğitim sonucunda belirlenen ağırlık değerlerini kullanarak daha önce görmediği bu örnekler için çıktılar üretmektedir. Elde edilen çıktıların doğruluk dereceleri ağın öğrenmesi hakkında bilgi vermektedir. Sonuç ne kadar iyi olursa eğitimin

performansı da o kadar iyi demektir. Eğer test seti ile istenilen sonuçlara ulaşılırsa ağın eğitimi biter.

Bazı ağlarda ise eğitim gerçek verilerle sürekli olarak devam eder. Bu noktada ağlar öğrenme türlerine göre ikiye ayrılabilirler.

Çevrimdışı Öğrenme: Çevrimdışı (Offline) öğrenme kuralına dayalı ağlar eğitildikten sonra gerçek hayatta kullanıma alındığında artık öğrenme olmamaktadır. Delta Öğrenme Kuralı bu tür öğrenmeye örnek olarak verilebilir.

Çevrimiçi Öğrenme: Çevrimiçi (Online) öğrenme kuralına göre öğrenen ağlar, gerçek zamanda çalışırken bir taraftan fonksiyonlarını yerine getirmekte, bir taraftan da öğrenmeye devam etmektedirler. ART ve Kohonen's SOM ağlarında kullanılan Kohonen Öğrenme Kuralı bu öğrenme kuralına örnek olarak verilebilir [21].

## 2.7. Yapay Sinir Ağlarında Öğrenme Stratejileri

YSA'da öğrenme kuralı Hebbian öğrenme kuralı denilen basit bir modele dayanır. Hebbian öğrenme kuralı temel olarak "Eğer iki düğüm arasında aynı zamanda etkin ise aralarındaki bağ gücü artar" kuramına dayanmaktadır. Öğrenme ağın içinde bulunduğu ortam tarafından, devam eden bir süreçte bağımsız ağ parametrelerinin ayarlanması işlemidir. Yani kısaca herhangi bir sistemi modellemek amacıyla tasarlanan bir ağda bağlantı ağırlıklarının ve hücre eşiklerinin istenilen giriş-çıkış eşleştirmesini sağlayacak şekilde belirleme işlemine öğrenme denir.

YSA'nın arzu edilen davranışı gösterebilmesi için amaca uygun olarak tasarlanması gerekir. Bu durum, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. YSA'nın karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Genellikle ağırlıklar, rastgele ya da sabit bir değerde seçilir. YSA'nın, istenen davranışı gösterecek şekilde ilgilendiği problemden aldığı eğitim örneklerini kullanarak problemi öğrenmelidir. Belli bir hata kriterine ve öğrenme algoritmasına göre, ağırlıkların yenilenerek, artık değişmediği durumda öğrenmenin gerçekleştiği söylenebilir.

Öğrenme yöntemi veya ağ mimarisi, hangi parametrelerin değiştirileceğine bağlıdır. Öğrenme işlemini gerçekleştirmek üzere tanımlanmış kuralların tümüne öğrenme algoritması denilir. Bu güne kadar çeşitli öğrenme algoritmaların her birinin kendine has üstünlükleri ve eksiklikleri vardır.

Danışmanlı ve danışmasız olmak üzere iki tip öğrenme türü vardır. Danışmanlı öğrenmede bir öğretmene ihtiyaç vardır. Öğretmen, bir veri alıştırma kümesi veya ağ sonuçlarının performansını derecelendiren bir gözlemci olabilir. Danışmanlı öğrenmede eğitilmiş sinirlere öğretme işaretini göndererek sinirler eğitilir. Bu işaretin bağlantısındaki ağırlıkları ayarlamakta kullanılır.

Bütün YSA; sinirler, bağlantılar ve aktarım işlevlerine bağlı olduğu için, farklı mimariler, yapılar ya da sinir ağları arasında bir benzerlik bulunmaktadır. Çeşitliliğin çoğunluğu farklı öğrenme kurallarından ve bu kuralların bir ağın yapısını nasıl değiştirdiğinden kaynaklanmaktadır. Öğrenme yöntemlerine göre ağ yapıları Tablo 2.4'te görülmektedir [5].

Tablo 2.4. Öğrenme yöntemlerine göre ağ yapıları

| Ağ Yapıları | |
|---|---|
| Danışmanlı | Danışmansız |
| Perceptron | Hopfield Ağı |
| Çok Katmanlı Perceptron | Olasılıksal Sinir Ağı |
| Geri yayılım Ağı | Uyarlanır Rezonans Ağı |
| Daha Yüksek Düzeyli Sinir Ağı | Öz Örgütlemeli Harita Ağı |
| İşlevsel Bağ Ağı | Boltzman Makinesi |
| | Hamming Ağı |
| | Geri Yayılma İçine Özörgütlemeli Harita Ağı |
| | İki Yönlü Çağrışım Belleği |
| | Yığın Ağı |
| | Karşı Yayılma Ağı |
| | Öğrenme Vektör Nicelendirmesi |

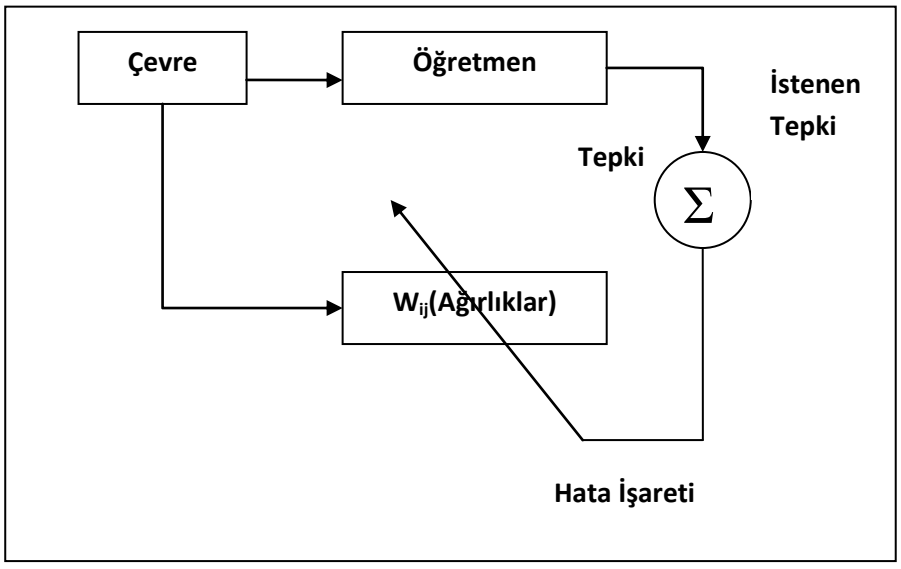
### 2.7.1. Danışmanlı öğrenme

YSA'da gerçek bir çıkış, istenen çıkışla kıyaslanır. Rastgele değişen ağırlıklar ağ tarafından öyle ayarlanır ki, bir sonraki döngüde gerçek çıkış ile istenen çıkış arasında daha yakın karsılaştırma üretebilsin. Öğrenme yöntemi, bütün işleme

elemanlarının anlık hatalarını en aza indirmeye çalışır. Bu hata azaltma işlemi, kabul edilebilir doğruluğa ulaşana kadar ağırlıklar devamlı olarak derlenir.

Danışmanlı öğrenmede, YSA kullanılmadan önce eğitilmelidir. Eğitme işlemi, sinir ağına giriş ve çıkış bilgileri sunmaktan oluşur. Bu bilgiler genellikle eğitme kümesi olarak tanımlanır. Yani, her bir giriş kümesi için uygun çıkış kümesi ağa sağlanmalıdır [5].

Danışmanlı öğrenmede, girdi ve çıktı değerlerinin her ikisi de ağa gösterilir. Girdi değerleri ağ tarafından işlenerek istenilen çıktı değerleri ile ağın çıktı değerlerini karşılaştırır. Aradaki fark "hata" olarak ele alınır. Performans fonksiyonu ile hesaplanan bu hata değerini minimize etmek için hesaplanan değer sisteme geri verilir. Ağ, kendi çıktı değerini istenilen çıktı değerine yaklaştırmak için hücre bağlantılarının ağırlıklarını değiştirir. Bu sayede girdilerle çıktılar arasındaki ilişkiler öğrenilmektedir [21]. Hata değeri istenen değerin altına düştüğünde tüm ağırlıklar sabitlenerek eğitim işlemi sonlandırılır. Eğitim işlemi sırasında her bir eğitim bilgisi çifti için oluşan hata değerine göre ağırlıkların değiştirilmesine 'örüntü kipi' öğrenme, tüm eğitim kümesi için hataların toplanarak toplam hata değerine göre ağırlıkların değiştirilmesine ise 'küme kipi' öğrenme denilmektedir. Danışmanlı öğrenme Şekil 2.9'da gösterilmiştir [5].




Şekil 2.9. Danışmanlı öğrenme [5]

Öğrenme süreci, hata değerleri istatistiksel olarak kabul edilebilir seviyeye ininceye kadar devam eder. Çok Katmanlı Algılayıcı (Multilayer Perceptron) ağları bu tür öğrenme metodunu kullanan ağlara örnektir [21].

Eğer sistemin önemli olan özelikleri ve ilişkileri öğrenmesi gerekiyorsa, o zaman eğitme kümesi, bütün ihtiyaç duyulan bilgileri içermesi gerekir. Eğer ağ sadece bir örnekle eğitilirse, bir olay için çok hassas olan bütün ağırlıklar kümesi, bir sonraki olayda yeterli çözüm vermez. Yeni şeyler öğrenme safhasında eski olaylar unutulabilir. Sonuç olarak, sistem gerekli bilgilerle birlikte öğrenmek zorundadır.

Giriş ve çıkış bilgilerinin nasıl sunulacağı veya nasıl kodlanacağı, bir ağı başarılı bir şekilde yönlendirmek için önemli bir unsurdur. YSA sadece sayısal giriş bilgileriyle çalışırlar. Bu nedenle ham bilgiler genellikle ölçeklendirilmelidirler [5].

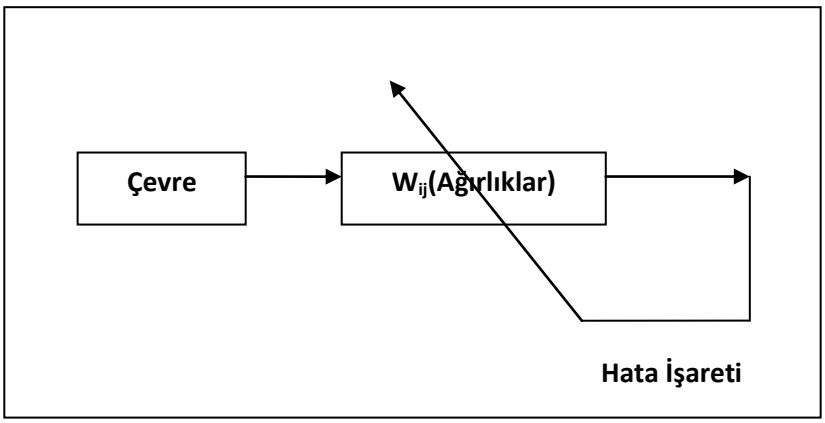
### 2.7.2. Danışmansız öğrenme

Danışmansız öğrenmede YSA' ya sadece girdiler verilmekte, ulaşılması gereken beklenen çıktılar verilmemektedir. Girişte verilen örnek değerlere bakarak yapay sinir ağı, parametreler arasındaki ilişkileri kendi kendine öğrenir. YSA daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren örüntüler (patterns) oluşturmak üzere ayarlar. Danışmansız öğrenme genellikle sınıflandırma problemlerinin çözümünde kullanılmaktadır. ART (Adaptive Resonance Theory) danışmansız öğrenmeye örnek olarak verilebilir [17].

Danışmansız öğrenme metodu, yapay sinir ağlarında sürekli araştırılan ve gelişen bir öğrenme metodudur. Bu metod, gelecekte bilgisayarların insan yardımı olmadan öğrenebileceklerinin göstergesidir. Ancak günümüzde sınırlı kullanım alanları bulan ve hala yoğun araştırma konusu olan bir öğrenme metodudur [28].

Danışmansız öğrenmeye, Hebbian öğrenme, Grossberg öğrenme, Kohonen'in öz örgütlemeli harita ağı örnek olarak verilebilir. Kohonen tarafından geliştirilen danışmansız öğrenme yönteminin kullanıldığı öz örgütlemeli harita ağ da biyolojik sistemlerdeki öğrenmeden esinlenmiştir. Bu yöntemde sinirler öğrenmek için elverişli durum ya da ölçülerini güncellemek için yarışırlar. En büyük çıkış ile

islenen sinir, kazananı belirler ve komsularına bağlantı boyutlarını güncellemeleri için izin verir. Şekil 2.10'da danışmansız öğrenme ana hatlarıyla gösterilmiştir.




Şekil 2.10. Danışmansız öğrenme [5]

Danışmansız öğrenmede ağ istenen dış verilerle değil, girilen bilgilerle çalışır. Bu tür öğrenmede gizli sinirler dışarıdan yardım almaksızın kendilerini örgütlemek için bir yol bulmalıdırlar. Bu yaklaşımda, verilen giriş vektörleri için önceden bilinebilen performansını ölçebilecek ağ için hiç bir çıkış örneği sağlanamaz. Yani ağ yaparak öğrenmektedir [5].

Ayrıca hem danışmanlı hem de danışmansız öğrenmeyi birlikte kullanan YSA da bulunmaktadır. Bu ağlarda ağırlıkların bir kısmı danışmanlı öğrenmeyle bir kısmı da danışmansız öğrenmeyle ayarlanır. Radyal tabanlı yapay sinir ağları (Radial Basis Networks - RBN) ve olasılık tabanlı yapay sinir ağları (Probability Based Neural Networks - PBNN) bunlara örnek olarak verilebilir [18].

### 2.7.3. Destekleyici öğrenme

Destekleyici öğrenme, danışmanlı öğrenmenin özel bir biçimidir. Bu öğrenmede stratejisinde, beklenen çıktı değeri tam olarak bilinmemektedir. Sadece üretilen çıktının doğru olup olmadığına bakılır. Sistemin verilen girdi değerlerine karşılık çıktı üretmesi beklenir. Girdi-çıktı uygunluğu öğretmen tarafından kontrol edilerek sisteme doğru veya yanlış şeklinde sinyal gönderilir. LVQ ağı bu stratejiyi kullanmaktadır [16].

Bu metottan bazı kaynaklarda danışmanlı bazı kaynaklarda danışmansız öğrenmenin bir alt türü olarak, bazı kaynaklarda ise kendi başına bir öğrenme metodu olarak bahsedilmektedir. Bu metoda göre, YSA' ya sadece girdiler verilmekte, bu girdilere karşılık çıktıları üretmesi beklenmekte ve bu çıktıların ne derece doğru olduğunu belirten bir skor veya derece bildirilmektedir [22].

### 2.7.4. Karma öğrenme

Yukarıdaki 3 stratejiden birkaçını birlikte kullanarak öğrenme gerçekleştiren ağlarda vardır. Radyal tabanlı yapay sinir ağları (RBN) ve olasılık tabanlı ağlar (PBNN) bunlara örnek verilebilir [16].

## 2.8. Yapay Sinir Ağlarında Öğrenme Kuralları

YSA gibi öğrenen sistemlerde öğrenme, yukarıda anlatılan stratejilerden hangisi uygulanırsa uygulansın bazı kurallara göre gerçekleştirilmektedir. Bu kurallardan bazıları çevrimiçi (on-line) bazıları ise çevrimdışı (off-line) çalışmaktadır.

### 2.8.1. Çevrimiçi (on-line) öğrenme kuralları

Bu kurallar gerçek zamanlı çalışabilmektedir. Bir taraftan fonksiyonlarını yerine getirmekte diğer taraftan ise öğrenmeye devam etmektedirler. Sistem üzerinde bir öğrenme algoritması ve bu yapıyı destekleyen donanım ve yazılımlar mevcuttur. Bu yaklaşım daha çok farklı uygulamaların sıkça kullanıldığı ve öğretilecek sistemin devamlı olarak farklı davranışlar sergilediği uygulamalarda sıkça kullanılırlar. ART ağının öğrenme kuralı ve Kohonen öğrenme kuralı bu sınıfta bulunan öğrenme kurallarına örnek verilebilir.

### 2.8.2. Çevrimdışı (off-line) öğrenme kuralları

Ağların çoğu bu çalışma modunu kullanır. Çevrimdışı öğrenme kuralına dayanan sistemler kullanıma alınmadan önce örnekler üzerinde eğitilirler. Bu kuralları kullanan sistemler eğitildikten sonra gerçek hayatta kullanıma alındığında artık öğrenme olmamaktadır. Sistemin öğrenmesi gereken yeni bilgiler söz konusu olduğunda sistem kullanımdan çıkarılmakta ve çevrimdışı olarak yeniden eğitilmektedir [2].

Eğitim tamamlanınca sistem tekrar kullanıma alınmaktadır. YSA'da yaygın olarak kullanılan 'Delta Öğrenme Kuralı' bu tur öğrenmeye örnek olarak verilebilir [16].

## 2.9. Temel Öğrenme Kuralları

YSA'nın eğitim süreci, belli kurallar çerçevesinde olmaktadır. Bu kurallara öğrenme kuralları adı verilmektedir. YSA'nın eğitimi için birçok öğrenme kuralı kullanılmaktadır. Öğrenme kuralları, kullanılan yapay sinir ağlarının amacı ve ağın topolojisi ile doğrudan ilişkilidir. Ağırlıkların değiştirilmesi bu kurallara göre yapılmaktadır.

Birçok öğrenme kuralı en eski öğrenme kuralı olan Hebb öğrenme kuralının varyasyonudur. Bunun dışında, araştırmacılar sürekli olarak yeni öğrenme kuralları geliştirmekte ve insanın öğrenmesine benzeyen çeşitli öğrenme kuralları geliştirmektedirler. Fakat makinelerin öğrenme işlemleri insanın öğrenme işlemiyle karşılaştırıldığında çok sınırlıdır. Öğrenme gerçekte, var olan öğrenme kuralları ile basitçe ifade edilemeyecek kadar karmaşık bir yapıya sahiptir. Bazı önemli öğrenme kuralları şunlardır [21].

### 2.9.1. Hebb kuralı

Bilinen en eski öğrenme kuralıdır. Diğer öğrenme kurallarının temelini oluşturmaktadır. 1949 yılında geliştirilen bu kurala göre, bir hücre (YSA elemanı) diğer bir hücreden bilgi alırsa ve her iki hücrede aktif ise (matematik olarak aynı işareti taşıyorsa) her iki hücrenin arasındaki bağlantı kuvvetlendirilmelidir. Diğer bir deyişle bu kural şu şekilde özetlenebilir. Bir hücre kendisi aktif ise pasif yapmaya çalışmaktadır. Diğer öğrenme kurallarının çoğu bu felsefeyi baz alarak geliştirilmiştir [16].

### 2.9.2. Hopfield kuralı

Bu kural, kuvvetlendirme veya zayıflatmanın genliğini belirleyebilmesi istisnası haricinde Hebb kuralıyla benzerdir. Buna göre, "Eğer istenilen çıkış ve girişin her ikisi de aktif veya her ikisi de durgun ise, bağlantı boyutlarını öğrenme oranı kadar arttır, aksi halde boyutu öğrenme oranı kadar azalt". (Öğrenme fonksiyonlarının çoğunun öğrenme oranı veya öğrenme sabiti için bazı koşulları vardır).

Ayrıca Boltzman Makinesi ve Hamming Ağı öğrenme kuralları Hopfield Kuralının değişik açılardan geliştirilmiş halleridir. Boltzman Makinesi Hopfield ağına ilave olarak özgün modelleme tekniğinde benzer işlev ve işlemleri kullanırlar. Boltzman Makinesi model seviyelerini araştırıp durum uzayında kavramları benzeterek birleştirmiştir. Hamming Ağı ise giriş vektörleri için en az ikili sayı hatasının temel sınıflandırılması yerine getirmektedir. Burada Hamming tarafından hata aralığı tanımlanmaktadır [5].

### 2.9.3. Delta kuralı

Widrow ve Hoff tarafından geliştirilen bu kural Hebb Kuralının gelişmiş şeklidir. En çok kullanılan kurallardan biri olan Delta Kuralı, yapay sinir hücresinin gerçek çıktısı ile beklenen çıktısı arasındaki farkı azaltmak için YSA'nın işlemci elemanları arasındaki bağlantı ağırlık değerlerinin sürekli değiştirilmesi ilkesine dayanır. Bu kuralla, gerçek çıktı ile beklenen çıktı arasındaki hatanın karesi en aza indirilmeye çalışılmaktadır. Bu nedenle En Küçük Kareler Kuralı (Least Mean Square Rule-LMS) olarak da adlandırılır. Ayrıca bazı kaynaklarda Widrow-Hoff Kuralı olarak da geçer [18].

Bu öğrenme kuralı zaman içinde geliştirilerek yeni öğrenme kuralları oluşturulmuştur. Bunlardan bazıları söyle sıralanabilir: Delta Bar Delta, Genişletilmiş Delta Bar Delta, İşlevsel-Bağ Ağı (Daha Yüksek Düzeyli Sinir Ağı). Bu kuralların değişik durumlarda birbirlerine göre üstün ve zayıf yönleri vardır. Bu kurallar aynı temel mantığa sahip olmalarının rağmen öğrenme oranları, moment terimi, hata farkı değişkeninin ağ üzerindeki etkisi vb. noktalar üzerinde bir takım gelişmeler ve değişiklikler içermektedirler [5].

### 2.9.4. Kohonen kuralı

Teuvo Kohonen (1982) tarafından, biyolojik sistemlerdeki öğrenmeden esinlenilerek geliştirilen bu öğrenme kuralına, Yarışmacı Öğrenme Kuralı (Competitive) da denmektedir.

Bu kuralda işlemci elemanlar, ağırlıklarının ayarlanması için yarışmaktadırlar. Hebb kuralından farklı olarak bir seferde yalnız bir işlemci elemanın, yani yalnızca

kazanan nöronun bağlantı ağırlıkları değiştirilmektedir. En uygun çıktıya sahip işlemci elemanın kazandığı bu kuralda, kazanan işlemci eleman, kendisine komşu diğer işlemci elemanların ağırlıklarının değiştirilmesine de izin vermektedir. Sadece kazanan elemanın çıktı üretmesine ve komşu hücreleri ile birlikte ağırlıklarının değiştirilmesine izin verilir.

Komşu sayısı eğitim süresince değişiklik gösterir. Eğitim süreci boyunca en geniş komşu tanımından en dar komşu tanımına inilir. Kazanan eleman girdi desenini en iyi ifade eden eleman olarak tanımlanır [21].

### 2.9.5. Eğimli iniş kuralı

Bu kural Delta kuralına benzer çünkü transfer fonksiyonunun türevi bağlantı ağırlıklarına uygulamadan önce, Delta hatasını düzeltmek için kullanılır. Bu kural durağan bir noktaya çok bir şekilde yaklaşmasına rağmen sıkça kullanılır. Bir ağın farkı katmanların için öğrenme oranları, öğrenme işleminin daha hızlı olmasına yardımcıdır. Giriş verilerinin güçlü bir modelden çıkarılmadığı uygulamalarda, bu işlem özellikle önemlidir [5].

## 2.10. Yapay Sinir Ağı Modelleri

Bir YSA'da, işlemci elemanların bağlanması sonucu oluşan topoloji, işlemci elemanların sahip oldukları toplama ve aktivasyon fonksiyonları, kullanılan öğrenme metodu ve öğrenme kuralı ağın modelini belirlemektedir. YSA uygulamasının başarısı, modelin oluşturulması aşamasının en doğru şekilde yürütülmesi ile yakından ilgilidir. Bunun için YSA tasarımcısının, ağın yapısına ve işleyişine ilişkin şu kararları vermesi gerekmektedir [18]:

- Ağ mimarisinin seçimi ve yapısal özelliklerinin belirlenmesi (katman sayısı ve katmandaki işlemci eleman sayısı gibi),
- İşlemci elemanların kullandığı fonksiyonların karakteristik özelliklerinin belirlenmesi,
- Öğrenme algoritması ve parametrelerinin belirlenmesi,
- Eğitim ve test setinin oluşturulması.

Bu kararların doğru verilememesi durumunda, YSA'ları sistem karmaşıklığı artacaktır. Sitem karmaşıklığı yapısal ve toplam hesaplama karmaşıklığının bir fonksiyonudur. Toplam hesaplama karmaşıklığı ise, genellikle yapısal karmaşıklığın bir fonksiyonu olarak ortaya çıkar ve bu hesaplamanın en aza indirilmesi amaçlanır. Bu hesaplama karmaşıklığının ölçülmesinde de genellikle YSA sisteminin toplam tepki süresi veya sisteme ait bir işlemci elemanın tepki süresi değeri temel alınır. Bunun yanında kapladığı hafıza ve zaman karmaşıklığı bazı uygulamalarda hesaplanmaktadır [17].

Bir problemin çözümü için uygulanacak olan YSA modeli öncelikle problemin türüne bağlı olmaktadır. Hangi ağ modelinin hangi problemin çözümü için daha uygun olduğunun bilinmesi oldukça önemlidir. Kullanım amacı ve o alanda başarılı olan ağ modelleri aşağıda görülmektedir [18].

YSA mimarisi ve öğrenme algoritmasının seçimi birbirleriyle yakın ilişki içerisindedir. Ağda kullanılacak öğrenme algoritması seçildiğinde, ağın mimarisi de zorunlu olarak seçilmiş olmaktadır veya tam tersi ağın mimarisi seçildiğinde kullanılabilecek öğrenme algoritması da büyük ölçüde belirlenmiş olmaktadır.

YSA modelinin tasarlamasındaki bir diğer adım ise, çok katmanlı bir ağ modelinin seçilmesi durumunda, ağdaki ara katman sayısına karar vermektir. İşlemci elemanların aynı doğrultu üzerinde bir araya gelmeleriyle katmanlar oluşmaktadır. Çoğu problem için 2 veya 3 katmanlı bir ağ tatmin edici sonuçlar üretebilmektedir. Katman sayısını belirlemenin en iyi yolu, birkaç deneme yapılarak ağın performansına bağlı olarak en uygun katman sayısına karar vermektir.

Ağın yapısal özelliklerinden bir diğeri de her katmandaki işlemci eleman sayısının belirlenmesidir. İşlemci elemanların sayısına da genellikle deneme yanılma yolu ile karar verilir. Bir YSA'da işlemci eleman sayısının, olması gerekenden daha az olması ağın öğrenme yeteneğini azaltırken, gereğinden çok olması genelleme yeteneğini azaltmaktadır.

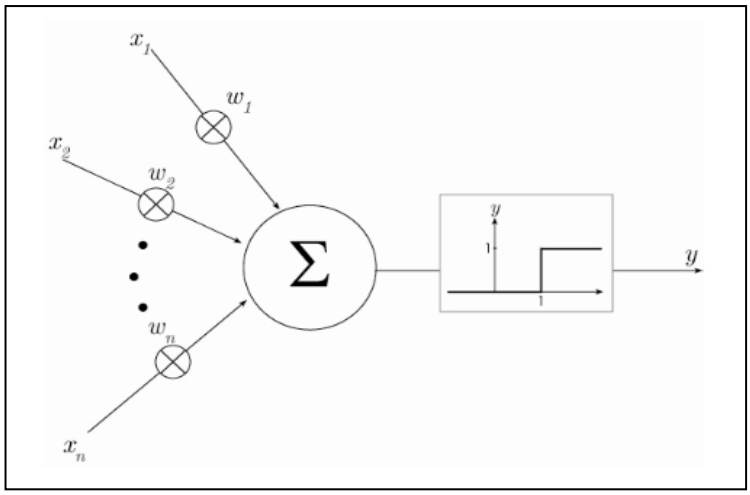
İşlemci elemanların karakteristik özelliklerinin belirlenmesi de YSA'nın tasarımında önemli kararlardan biridir. Toplama ve aktivasyon fonksiyonlarının belirlenmesi, büyük ölçüde verilerin özelliklerine ve ağın neyi öğrenmesinin istendiğine bağlıdır.

Aktivasyon fonksiyonları içinde en çok kullanılanlar, sigmoid ve hiperbolik tanjant fonksiyonlarıdır. Eğer ağın, bir modelin ortalama davranışını öğrenmesi isteniyorsa sigmoid, ortalamadan sapmasını öğrenmesi isteniyorsa hiperbolik tanjant fonksiyonlarının kullanımı önerilir [18].

### 2.10.1. Tek katmanlı algılayıcılar

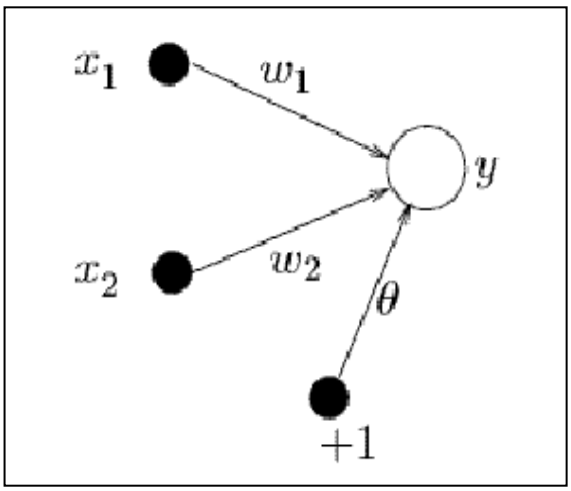
Sinir ağları üzerindeki en etkili çalışma 60'lı yıllarda 'Perceptrons' terimi baslığı altında Frank Rosenblatt tarafından kazandırılmıştır. Algılayıcıya bazı ilave, sabitleme, on işlemeler ile MCP modeli (neurone with weighted inputs/girdileri ağırlıklandırılmış nöron) meydana gelir. 1969 'da Minsky ve Papert'in bir kitap yazarak tek katmanlı algılayıcıların sınırlarını ortaya koymalarıyla birçok araştırmacının araştırmaları durdu. 80 'li yıllara kadar yapay sinir ağları üzerindeki araştırmalar birkaç araştırmacı dışında neredeyse durmuştu. Daha sonraları uygun bir eğitim ile çok katmanlı algılayıcıların bu problemleri çözebileceği kanıtlanmıştır [2].

Tek katmanlı, ileri beslemeli ağlar, bir veya birden fazla nörona sahip bir girdi ve bir çıktı katmanından oluşurlar. Girdi katmanındaki her nöron, çıktı katmanındaki tüm nöronlarla bağlantılıdır. Nöronların girdi değerleri bağlantıların ağırlıkları ile çarpılıp toplanır. Eğer bu toplam değeri belirli bir eşik değerini aşarsa nöron bir çıktı değeri üretir. Bu en basit yapay nörona Eşik Mantık Birimi (EMB) denir. McCulloch ve Pitts tarafından önerilen en basit EMB yapısı Şekil 2.11 'de gösterilmiştir.




Şekil 2.11. McCulloch ve Pitts tarafından önerilen en basit EBM modeli [21]

n adet girdi ($x_1$, $x_2$,....,$x_n$) ve bağlantıların ağırlıkları ($w_1$,$w_2$,....,$w_n$) ile ifade edilir. En basit haliyle, tek işlemci elemanlı bir yapay sinir ağı Şekil 2.12'de gösterilmiştir.




Şekil 2.12. Tek işlemci elemana sahip yapay sinir ağı [21]

Çıktı katmanındaki nöronun girdi değeri, girdi katmanındaki nöronların ağırlıklı toplamı ile Yanlı (Bias-θ) teriminin toplamından oluşur ve Eşitlik (2.6)'daki gibi ifade edilir.

$$y=F\left(\sum_{i=1}^{n}(w_i+x_i)+\theta\right) \tag{2.6}$$

Tek katmanlı ağlarda, girişlerin toplamının ve dolayısıyla çıkışın sıfır olmasını önleyecek bir eşik değer (φ ) kullanılabilir. Kullanılmasını zorunlu kılan durumların dışında, eşik değerin olması ya da olmaması keyfidir. Eşik değer iterasyonlar sırasında, ağırlık değerleri gibi değiştirilebilir. Eşik değerin girişi her zaman 1'dir [20].

Ağın çıktısı, çıktı katmanındaki nöronun aktivasyon fonksiyonunun çıktısıdır (y). $X_i$ değerleri girdi katmanındaki nöronların değerlerini, $w_i$ değerleri bağlantıların ağırlıklarını, n değeri girdi katmanındaki toplam nöron sayısını, θ değeri yanlı terimini (eşik değeri), F fonksiyonu ise aktivasyon fonksiyonunu ifade eder. Aktivasyon fonksiyonu; doğrusal, sigmoid, eşik, adım(işaret) fonksiyonu gibi bir fonksiyon olabilir. Eğer eşik fonksiyonu seçilmişse ağın çıktısı şu fonksiyona göre

Eşitlik (2.7)'deki gibi, eğer adım fonksiyonu seçilmişse Eşitlik (2.8)'deki gibi hesaplanmaktadır [21].

$$y = (F) = \begin{cases} 1 & s > \theta \\ 0 & s \leq \theta \end{cases} \tag{2.7}$$

$$y = (F) = \begin{cases} 1 & s > 0 \\ -1 & s \leq 0 \end{cases} \tag{2.8}$$

Tek katmanlı algılayıcılarda, giriş değerleri iki farklı sınıfa ayrılarak, kümelendirilmeye çalışılır. Dolayısıyla problem, iki sınıfı birbirinden ayıran bir yüzeyin bulunmasıdır. Bu yüzey, uzayı iki farklı bölgeye ayırır ve farklı çıkış değerlerini alan giriş parametreleri, bu yüzey tarafından kümelendirilmiş olur [11].

Tek katmanlı ağın (Perceptron) eşik fonksiyonu ile kullanımı Doğrusal Ayrım Fonksiyonunu(Linear Discriminant Function) ifade eder. Şekil 2.13'deki örnekte doğrusal ayrım fonksiyonunu kullanan EMB, girdi desenlerini iki grup şeklinde sınıflandırmaktadır, yani girdi uzayını iki parçaya bölmektedir. Bu durumda ağ, sınıflandırma görevi görecektir, yani ağ, girdi desenlerinin belirlenen iki sınıftan birine ait olduğuna karar verecektir. Ağ, eğer toplam girdiler pozitif ise girdi deseninin çıktısını +1 yapacak ve deseni +1 sınıfına atayacak, eğer toplam girdiler negatif ise girdi desenini aynı şekilde -1 olarak atayacaktır. İki sınıf arasındaki ayrım, bu durumda şu denklemle ifade edilir.

$$(w_1+x_1)+(w_2+x_2)+\theta=0 \tag{2.9}$$

EMB'nin çalışma şekli geometrik olarak gösterilebilir. Her çıktı deseni $x_1$ ve $x_2$ şeklinde iki bileşene sahiptir. Girdi değerlerinin oluşturduğu uzaya desen uzayı (Pattern Hyberspace) denir. Her desen uzayda kendi bileşenini oluşturarak bir nokta belirtir. n girdi olması durumunda uzay n boyutlu olacaktır. Açıktır ki, üçten fazla girdi olduğu takdirde desen uzayının çizimi imkansız hale gelecektir.

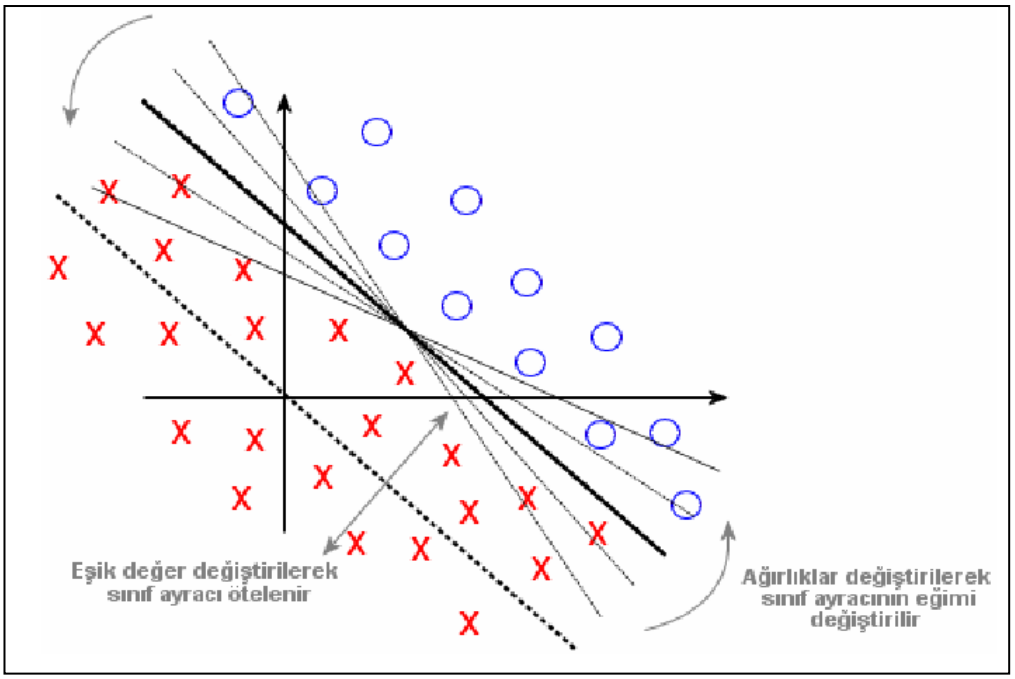
Eşitlik (2.9), şu şekilde tekrar yazılır (2.10):

$$x_i = -\frac{w_1}{x_1} x_i - \frac{\theta}{w_2} \tag{2.10}$$

Bu eşitlik iki boyuttaki doğru denklemidir:

$$x_2 = ax_1 + b \tag{2.11}$$

Bu doğruya Karar Doğrusu (Decision Line) denir. Eşitlik (2.10)'da ağırlıkların oranının doğrunun eğimini, yanlı teriminin de doğrunun koordinat ekseninin başlangıç noktasından ne kadar uzaklaştığını belirlediği görülmektedir. Doğrusal Eşik Sinir Ağı'nın Düzlem (Hyperplane) üzerindeki geometrik şekli Şekil 2.13'te gösterilmiştir. Dikkat edilirse ağırlıklar girdi uzayında vektör olarak ifade edilirler ve ağırlık vektörü ayrım fonksiyonuna her zaman diktir. İkiden fazla boyutlarda Karar Düzlemi (Decision Hyperplane) haline gelir [21].




Şekil 2.13. Doğrusal ayrım fonksiyonunun geometrik gösterimi [20]

Eşik değerin değiştirilmesi, sınıf ayracını ötelerken, ağırlıkların değiştirilmesi eğimini etkiler. Eğitim sırasında eşik değer ve ağırlıklar değiştirilerek, doğrunun, sınıfları ayıracak şekilde konumlandırılması sağlanır [20].

Tek katmanlı yapay sinir ağlarında öğrenmeden kastedilen, sınıfları birbirinden ayıran en uygun doğrunun bulunmasıdır. Bunun için ağırlık değerleri değiştirilmektedir. Ağırlıkların değiştirilmesi doğrunun eğiminin değiştirilmesi anlamına gelir. t zaman biriminde ağırlık $\Delta_w$ değerleri kadar değiştirilir ise;

$$w_i(t+1)=w_i(t)+\Delta w_i(t) \quad (2.12)$$

olmaktadır. Öğrenme sırasında bu değişim her iterasyonda gerçekleştirilerek doğrunun en uygun eğimi bulunmaya çalışılır. Ancak bu işlem yeterli olmayabilir. Bu nedenle eşik değerinin de değiştirilmesi gerekir. Bu işlem, doğrunun sınıflar arasında kaymasına yardımcı olmaktadır. Böylece aktivasyon fonksiyonunun konumu belirlenmiş olur. Bu durumda t anında eşik değeri;

$$\theta(t+1)=\theta(t)+\Delta\theta(t) \quad (2.13)$$

şeklinde değiştirilmektedir. Öğrenme sırasında ağırlıklarda olduğu gibi eşik değeri de her iterasyonda $\Delta\theta$ kadar değiştirilmektedir.

Tek katmanlı yapay sinir ağlarında önemli iki modelden bahsedilebilir. Bu modeller perceptron ve Adaline/Madaline'dir [18].

### 2.10.2. Çok katmanlı algılayıcılar

Rosenblatt'ın orijinal perceptron (algılayıcı) tanımlamasından sonra diğer perceptron modelleri de gelişmiştir. Çok katmanlı algılayıcı ağları birçok tek katmanlı yapının girdi ve çıktı katmanları arasında yer alan bir veya daha fazla gizli katman bulunduran hiyerarşik olarak ileri beslemeli topolojik yapısıdır. Gizli katman sayıları ve her katmandaki nöron sayısı uygulamaya göre değişkenlik gösterir [2].

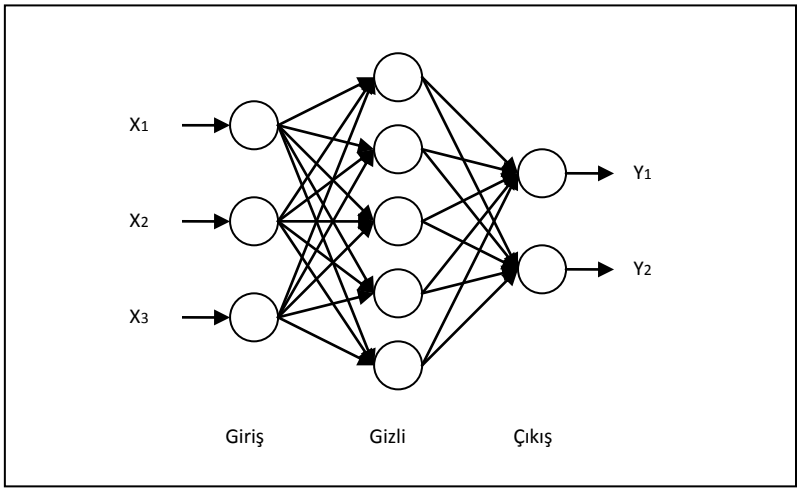
Çok katmanlı perceptron (multilayered perceptron-MLP), tek katmanlı perceptronların aksine doğrusal olmayan problemlere çözüm üretmeleri nedeniyle günümüzde geniş kullanım alanları bulan en popüler YSA'dır. Ayrıca kullandığı öğrenme algoritması nedeniyle geri yayılım ağı olarak da anılmaktadır [18].

YSA, sinir hücresi olarak adlandırılan birbirleriyle bağlanmış işlem ünitelerinin katmanlar halinde düzenlenmeleriyle oluşur. Üniteler arasındaki her bağlantı, beyindeki biyolojik sinapsların etkisine benzeyen gerçek ağırlık değerlerine sahiptir.

Öğrenme sürecinde ağ, ağırlık değerlerini ayarlar böylelikle ağa verilen girdi setine karşılık gelen doğru şekilde tahmin edilmiş veya sınıflandırılmış çıktı değerleri elde edilir [23].

Tek katmanlı algılayıcılar basit ve hızlı olmakla birlikte, sadece karmaşık olmayan lineer problemleri çözebildiklerinden, mühendislik uygulamalarında kullanılabilecek yeterlilikte değillerdir. Bu durum eğitilebilecek tek bir ağırlık matrisinin olmasından kaynaklanmaktadır. Bu yüzden karmaşık ve lineer olmayan problemlerin çözülmesinde çok katmanlı YSA'ya ihtiyaç duyulmaktadır.

Giriş ve çıkış katmanlarındaki hücre sayıları, uygulamanın niteliğine bağlıdır. Örneğin 3-girişli 2-çıkışlı bir sistem için tasarlanacak yapay sinir ağında, giriş katmanında 3 ve çıkış katmanında 2 tane hücre bulunacaktır. Gizli katman sayısı ve gizli katmanlarda bulunacak gizli hücre sayılarının belirlenmesinde ise bir kısıtlama yoktur. Fakat gizli katman ve gizli hücre sayılarının uygun şekilde seçilmemesi, YSA'nın performansını olumsuz şekilde etkileyecektir. Bu sayıların belirlenmesi bir optimizasyon problemidir. Şekil 2.14'de 3-girişli 2-çıkışlı bir sistem için tasarlanan çok katmanlı YSA gösterilmektedir. Bu tasarımda, tek bir gizli katman ve bu gizli katmanda 5 tane gizli hücre kullanılmıştır.



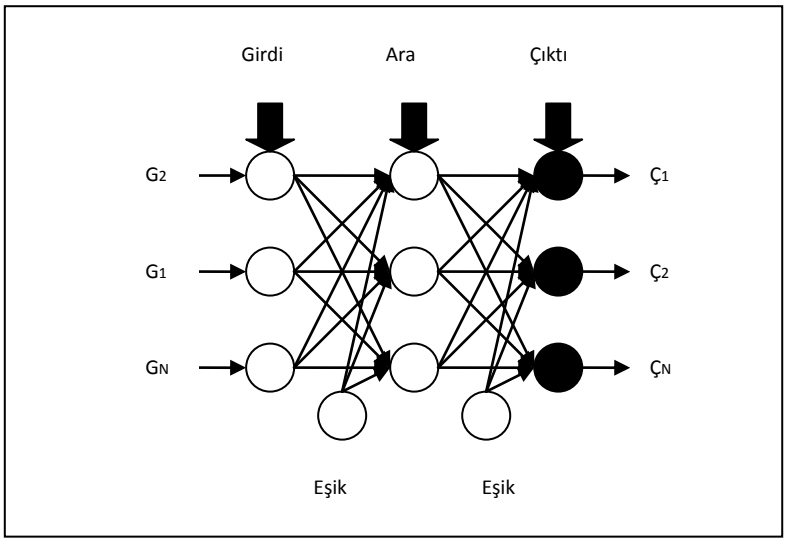

Şekil 2.14. İleri beslemeli çok katmanlı yapay sinir ağı [21]

Şekil 2.14'deki gibi ileri beslemeli ağ yapılarında, her bir hücre, sonraki katmanda bulunan tüm hücrelere bağlıdır. Bu yapısından ötürü tam-olarak-bağlanmış "fullyconnected" terimi kullanılır. Fakat bu durum hücrelerin birden fazla çıkışı

olduğu anlamına gelmez. Her hücrenin yalnız ve yalnız bir çıkışı vardır. Diğer bir ifade ile hücrelerin ürettikleri çıkış değeri tekdir. Fakat bu çıkış değeri, hücrelere aktarılırken ilgili ağırlıklar ile çarpılır ve böylelikle sonraki hücrelere kuvvetlendirilerek ya da zayıflatılarak iletilir [20].

Girdi katmanı: Dış dünyadan gelen girdileri ($G_1$,$G_2$,…,$G_N$) alarak ara katmana gönderir. Bilgi işleme olmadan gelen her bilgi ileri doğru iletilir. Her proses elemanının sadece bir girdisi ve bir çıktısı vardır. Bu çıktı bir sonraki katmanda bulunan bütün proses elemanlarına gönderilir [2]. Bir katmandaki bir işlemci eleman, bir sonraki katmandaki tüm işlemci elemanlara bağlanır. Ancak çok katmanlı perceptronlarda bütün katmanlardaki işlemci elemanlar yalnızca bir sonraki işlemci elemanlarla bağlantılıdır, kendi içlerinde bağlantıları bulunmaz. Girdi katmanındaki işlemci eleman sayısı uygulanan problemin giriş sayısına bağlıdır [18].

Ara katmanlar: Ara katmanlar girdi katmanından gelen bilgileri isleyerek bir sonraki katmana gönderir. Ara katmandaki her proses elemanı bir sonraki katmandaki bütün proses elemanlarına bağlıdır. Ara katman sayısı ve ara katmanlardaki proses elemanı sayısı deneme yanılma yolu ile bulunur [2].




Şekil 2.15. Çok katmanlı ağ modeli [2]

Çıktı katmanı: Ara katmandan gelen bilgileri isleyerek ağa girdi katmanından verilen girdilere karşılık ağın ürettiği çıktıları ($Ç_1$,$Ç_2$,…,$Ç_N$) belirleyerek dış dünyaya gönderir. Bir çıktı katmanında birden fazla proses elemanı olabilir. Bu sayı

uygulanan probleme bağlıdır. Her proses elemanının sadece bir tane çıktısı vardır [21].

### 2.10.3. Öğrenme Algoritması

Kaynaklarda kullanılan birçok öğrenme algoritması mevcuttur. Bu öğrenme algoritmalarının çoğunluğu matematik tabanlı olup ağırlıkların güncelleştirilmesi için kullanılırlar.

#### 2.10.3.1. Geriye Yayılım Algoritması

Geriye yayılım algoritması, YSA'nın parametrelerinin güncellenmesi için en çok kullanılan öğrenme algoritmasıdır. Günümüzde ses tanıma problemlerinden doğrusal olmayan sistem problemlerine kadar YSA ile çözüm üretilen birçok alanda başarı ile kullanılmaktadır. Ağ içerisinde hatayı geri yönde azaltmaya çalışmasından dolayı algoritmaya geriye yayılım algoritması adı verilir. Günümüzde geriye yayılım algoritmasının gelişmiş birçok versiyonu türetilmiştir. Fakat geriye yayılım algoritması genellikle genelleştirilmiş delta öğrenme algoritması ile ifade edilir.

Geriye yayılım algoritmalarında hesaplama iki bölümden oluşur. Bunlar:

• İleri hesaplama

• Geri hesaplama [24].

a. İleri Hesaplama

Bilgi işleme eğitim setindeki bir örneğin girdi katmanından ağa gösterilmesi ile başlar. Girdi katmanındaki k. proses elemanının çıktısı Eşitlik (2.14)'deki gibi belirlenir. Ara katmandaki her proses elemanı girdi katmanındaki bütün proses elemanlarının gelen bilgileri bağlantı ağırlıklarının ($A_1$, $A_2$…) etkisi ile alır. Ara katmandan proses elemanlarına gelen net girdi Eşitlik (2.14)'de verilmiştir [20].

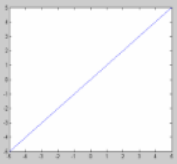
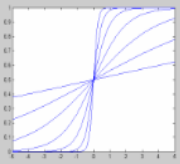
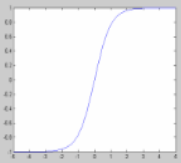
$$Ç_k^i = G_k$$

$$NET_j^a = \sum_{k=1}^{n} A_{kj} Ç_k^i \qquad (2.14)$$

Burada kj A k. girdi elemanını j. ara katman elemanına bağlayan bağlantının ağırlık değerini göstermektedir. j. ara katman elemanının çıktısı ise bu net girdinin aktivasyon fonksiyonundan geçirilmesiyle hesaplanır. Uygulamada genellikle sigmoid fonksiyonu kullanılır fakat kullanılması zorunlu değildir. Önemli olan türevi alınabilir bir fonksiyonun kullanılmasıdır. Sigmoid fonksiyonunun kullanılması halinde çıktı Eşitlik (2.15)'deki gibidir.

$$Ç_j^a = \frac{1}{1+e^{-(NET_j^a + B_j^a)}} \qquad (2.15)$$

Burada $\beta_j$, ara katmanda bulunan j. elemana bağlanan eşik değer elemanının ağırlığını göstermektedir. Bu eşik değer ünitesinin çıktısı sabit olup 1'e eşittir. Ağırlık değeri ise sigmoid fonksiyonunun oryantasyonunu belirtmek üzere konulmuştur. Eğitim sırasında ağ bu değeri kendisi belirlemektedir [6].

| Fonksyonun Adı | Fonksyonun Şekli | Matematiksel İfadesi | Türevi | Açıklama |
|---|---|---|---|---|
| Lineer Fonksyon |  | $f(NET) = NET$ | $F' = 1$ | Hesaplanan net giriş değeri, hücrenin çıkışı olarak kabul edilir |
| Sigmoid Fonksyonu |  | $f(NET) = \frac{1}{1+e^{-\beta.NET}}$ | $F' = \beta.F.(1-F)$ | $\beta$ değeri keyfi şekilde değiştirilerek farklı eğimlere sahip eğriler elde edilebilir |
| Hiperbolik Tanjant Fonksyonu |  | $f(NET) = \frac{e^{NET} + e^{-NET}}{e^{NET} - e^{-NET}}$ | $F' = (1-F^2)$ | Hesaplanan net giriş değeri, tanjant fonksyonuna uygulanır |

Şekil 2.16. Yapay sinir ağlarında kullanılan aktivasyon fonksiyonları [10]

b. Geri Hesaplama

Ağa sunulan girdi için ağın ürettiği çıktı ağın beklenen çıktıları ile karşılaştırılır. Bunların arasındaki fark hata olarak kabul edilir ve minimum değere indirilmeye çalışılır. Bunun için geriye hesaplamada bu hata ağın ağırlık değerlerine dağıtılarak

bir sonraki iterasyonda hatanın azaltılması sağlanır. Çıktı katmanındaki m proses elemanı için oluşan hata Eşitlik (2.16)'daki gibidir.

$$E_m=B_m-Ç_m \tag{2.16}$$

Bu bir proses elemanı için oluşan hatadır. Çıktı katmanı için oluşan toplam hatayı (TH) bulmak için bütün hataların toplanması gerekir. Bazı hata değerleri negatif olacağından toplamın '0' olmasını önlemek amacı ile ağırlıkların kareleri hesaplanarak sonucun kare kökü alınır.

$$TH=\frac{1}{2}\sum_{m} E_m^2 \tag{2.17}$$

Toplam hatayı en azlamak için bu hatanın kendisine neden olan proses elemanlarına dağıtılması gerekmektedir. Bu ise proses elemanlarının ağırlıklarını değiştirmek demektir. Ağın ağırlıklarını değiştirmek için iki durum söz konusudur.

- Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi
- Ara katmanlar arası veya ara katman girdi katmanı arasındaki ağırlıkların değiştirilmesi [16].

### 2.10.3.2. Levenberg–Marquardt Eğitim Algoritması (LM)

Levenberg–Marquardt (LM) eğitim algoritmasında Hessian matrislerinin hesaplanmadığı bir optimizasyon tekniği kullanılmıştır. LM eğitim algoritması, hata fonksiyonunun hata değerlerinin kareleri toplamı şeklinde ifade edilmesi durumunda Hessian matrislerini Eşitlik (2.18)' de belirtilen ifade yardımı ile hesaplanmaktadır.

$$H=J^TJ \tag{2.18}$$

Bu eşitlikte yer alan J, Jacobian matrisidir. Jacobian matrisi ağ yapısında oluşan hata değerlerinin ağırlık ve bias değişkenlerine göre 1. dereceden kısmi türevlerinden oluşur. LM eğitim algoritmasında hata fonksiyonunun gradientinin hesaplanabilmesi için de Eşitlik (2.19) kullanılır.

$$g=J^Te \tag{2.19}$$

Burada; e, ağ yapısında meydana gelen hata değerlerinin oluşturduğu bir hata vektörüdür. Hessian matrislerinin hesaplanmasına göre daha basit bir şekilde bulunabilen Jacobian matrisleri standart geri yayılım kuralına göre hesaplanabilmektedir. LM eğitim algoritması, Newton yönteminde kullanılan değişim ifadesine benzer bir yaklaşımı Hessian matrislerine uygulamaktadır. Bu yaklaşım Eşitlik (2.20)'de gösterilmiştir.

$$x_{k+1}=x_k-\left[J^TJ+\mu I\right]^{-1}J^Te \tag{2.20}$$

Skaler μ ifadesinin sıfır olduğu durumda ifade hemen hemen Newton yöntemi ile benzer olmaktadır. μ değerinin büyümesi durumunda ise eşitlik küçük bir adım boyunun kullanıldığı bir Gradient Descent ifadesine dönüşmektedir. LM eğitim algoritması uygun ölçülerde boyutlandırılmış ileri beslemeli bir ağ yapısı için en hızlı şekilde yakınsayan bir yöntem olarak görünmektedir [25].

## 2.11. Yapay Sinir Ağı Tasarımı

Bir sinir ağı modeli oluşturmak için nöronların bağlanış şekli (topoloji), işlemci elemanların kullandıkları toplama ve aktivasyon fonksiyonları, öğrenme metodu, öğrenme kuralı ve algoritması belirlenmelidir. Eldeki veriye ve ağdan yapmasını istediğiniz uygulamanın şekline göre model tasarlanır. Kurulan modelin başarısı modelin mimarisinin doğru oluşturulması ile direkt ilgilidir. Bunun için YSA tasarımcısının, ağın yapısına ve işleyişine ilişkin şu kararları vermesi gerekmektedir.

• Ağ mimarisinin seçimi ve yapısal özelliklerinin belirlenmesi (katman sayısı ve katmandaki işlemci eleman sayısı gibi),

• İşlemci elemanların kullandığı fonksiyonların karakteristik özelliklerinin belirlenmesi,

• Öğrenme algoritması ve parametrelerinin belirlenmesi,

• Eğitim ve test setinin oluşturulması.

Bu kararlar doğru verilmediği takdirde sistem karmaşıklığı artacak veya kararlı ve istikrarlı sonuçlar alınamayacaktır. YSA'nın toplam tepki süresi, eğitim süresi,

sistem karmaşıklığı ile artar. Sistem tasarlanırken uygulamanın ne kadar süreceği, hafızada ne kadar yer kaplayacağı gibi bilgiler hesaplanmalıdır.

### 2.11.1. Öğrenme algoritması ve yapay sinir ağı topolojisinin seçimi

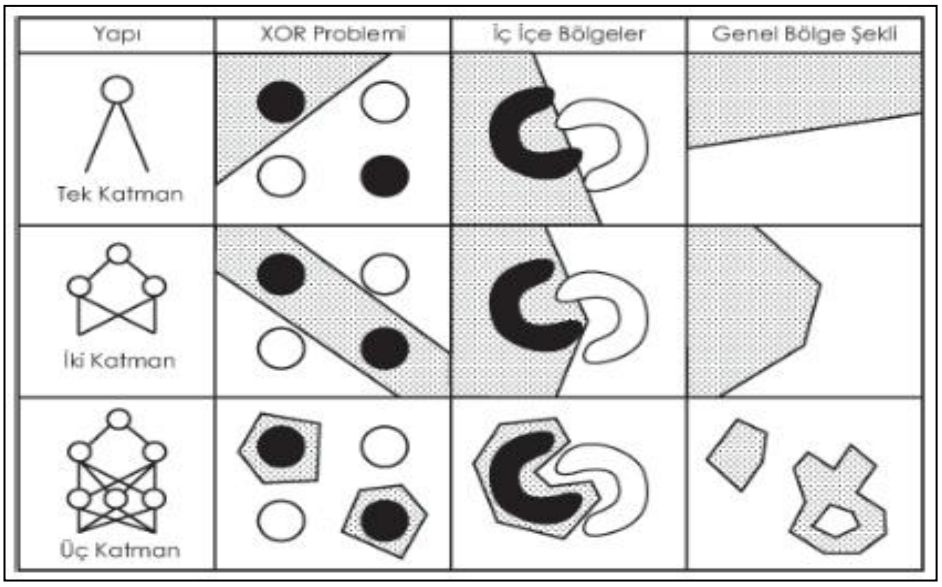
Bir problemin çözümü için uygulanacak olan YSA topolojisi öncelikle problemin türüne bağlı olmaktadır. Hangi ağ modelinin hangi problemin çözümü için daha uygun olduğunun bilinmesi oldukça önemlidir.

Probleme uygun YSA topolojisinin seçimi ağda kullanılması düşünülen öğrenme algoritmasına bağlıdır. Birçok YSA yapısı tek bir öğrenme algoritması ile kullanılabildiğinden öğrenme algoritmasının seçimi ile kullanılacak YSA topolojisi de zorunlu olarak seçilmiş olacaktır.

### 2.11.2. Katman ve nöron sayısının seçimi

Ağ yapısında kullanılan katman sayısı ve her katmanda bulunan nöron sayısı YSA'nın karmaşıklığını belirler. Katman sayısı ve katmanlardaki nöron sayısı arttıkça YSA'nın işlem ve öğrenme yeteneği artarken yakınsama süresi de artmakta, ağın genelleme kabiliyeti düşmekte ve ağın ezberlemesine (memorization) neden olmaktadır. Gereğinden az kullanımı ise verilerdeki desenin yeteri kadar öğrenilememesine yol açar. Çoğu problem için iki veya üç katmanlı bir ağ tatmin edici sonuçlar üretebilmektedir.

İşlemci eleman sayısı ve katman sayısını belirlemenin en iyi yolu, birkaç deneme yapılarak ağın performansına bağlı olarak en uygun katman sayısına karar vermektir. Bunun için izlenecek yol, başlangıçtaki nöron sayısını istenilen performansa ulaşıncaya kadar arttırmak veya tersi şekilde istenen performansın altına inmeden azaltmaktır. Şekil 2.17'de katman sayısının iki boyutlu desenlerin öğrenilmesinde yarattığı farklar geometrik olarak gösterilmiştir.




Şekil 2.17. Sınıflandırma işleminde, gizli katman sayısının iki boyutlu örnek uzayındaki rolünün geometrik gösterimi [22]

Çok katmanlı ileri beslemeli ağlar, bir gizli katman yardımı ile düzlemi konveks bölgelere ayırabilme yeteneğine sahiptir. Gizli katman sayesinde karar bölgeleri doğrusallıktan çıkarılmış, farklı türlerde konveks geometrik şekillere kavuşulmuş olunur. En azından her bir karar bölgesi için bir gizli işlemci eleman gerekmektedir. Eğer karar bölgeleri fazla ayrık veya üst üste iseler ikinci bir gizli eleman gerekmektedir.

### 2.11.3. Fonksiyonların seçimi

İşlemci elemanların karakteristik özelliklerinin belirlenmesi de yapay sinir ağının tasarımında alınacak önemli kararlardan biridir. Toplama fonksiyonunun ve aktivasyon fonksiyonunun seçimi büyük ölçüde verilerin özelliklerine ve ağdan öğrenilmesi istenen verinin türüne ve yapısına bağlı olmaktadır. Aktivasyon fonksiyonları içinde en çok kullanılanlar doğrusal, işaret, eşik, sigmoid, hiperbolik tanjant ve lojistik fonksiyonlarıdır. Doğrusal olmayan problemlerde doğrusal olmayan aktivasyon fonksiyonları kullanılmaktadır.

### 2.11.4. Normalizasyon

Veriler ağa sunulmadan önce normalizasyon işlemine tabi tutulurlar. Verideki aşırı salınımları engellemek ve sistem performansını arttırmak için kullanılırlar. Bunun

için logaritmik fonksiyonlar kullanıldığı gibi, genellikle verilerin [0,1] veya [-1,+1] aralıklarından birine ölçeklendirilmesi önerilmektedir. Ölçekleme verilerin geçerli eksen sisteminde sıkıştırılması anlamı taşıdığından veri kalitesi aşırı salınımlar içeren problemlerin YSA modellerini olumsuz yönde etkileyebilir [17].

### 2.11.5. Performans fonksiyonunun seçilmesi

Performans fonksiyonları, istenilen çıktı değerleri ile ağın ürettiği değerler arasındaki farkların kümülatif değerleri hesaplamaktadır. Hesaplanan bu değerler sayesinde ağın, eğitim setinin gösterdiği desene ne kadar yaklaştığı gözlenmekte ve bağlantıların ağırlık değerleri bu bilgiler kullanılarak değiştirilmektedir. Bu nedenle, performans fonksiyonları, öğrenme performansını etkileyen önemli hususlardan biridir. İleri beslemeli ağlarda kullanılan tipik performans fonksiyonu Hata Kareleri Ortalaması'dır (Mean Square Error, MSE).

$$MSE=\frac{1}{n}\sum_{i=1}^{n}[e(t)]^2 \tag{2.21}$$

Ortalama Hata(Mean Error, ME) Eşitlik (2.22)'de, Hata Kareleri Ortalamasının Karekökü (Root Mean Square) Eşitlik (2.23)'de ve Ortalama Mutlak Hata (Mean Absolute Error, MAE) Eşitlik (2.24)'de gösterilen fonksiyonlar, diğer performans fonksiyonlarından bazılarıdır.

$$ME=\frac{1}{n}\sum_{i=1}^{n}e(t) \tag{2.22}$$

$$RMSE=\sqrt{MSE=\frac{1}{n}\sum_{i=1}^{n}[e(t)]^2} \tag{2.23}$$

$$MAE=\frac{1}{n}\sum_{i=1}^{n}|e(t)| \tag{2.24}$$

### 2.11.6. Öğrenme katsayısı ve momentumun seçilmesi

Öğrenme sürecinde ağırlıklardaki değişim öğrenme oranı/katsayısı ($\lambda$) ile orantılıdır. Genellikle bu oran salınıma (oscillation) yol açmayacak kadar büyük alınmaya

çalışılır. Öğrenme oranı, ağırlıkların bir sonraki düzeltmede hangi oranda değiştirileceğini belirlemektedir. Ağ üzerinde önemli bir etkisi bulunan öğrenme oranının değeri uygulanan probleme göre değişmekle birlikte büyük bir değer seçilmesi salınıma yol açmakta ve ağın herhangi bir minimum değerine ulaşması zorlaşmaktadır. Küçük bir değer seçilmesi ise öğrenme süresinin uzamasına ve ağın yerel çözümlere takılmasına neden olmaktadır.

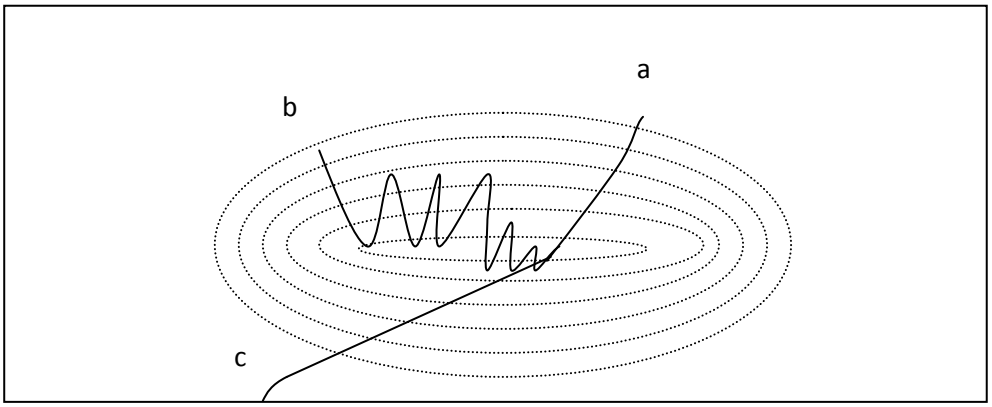
Yerel çözüm şu şekilde açıklanabilmektedir; bir problemin çözümü için en az hatayı veren ağırlık vektörünü pratikte her zaman yakalamak mümkün olmayabilmektedir. Bu çözüm, ağın sahip olabileceği en iyi çözümdür (Mutlak Minimum). Fakat bu çözüme nasıl ulaşılacağı tam olarak bilinmemektedir. Ağ, eğitim sırasında bu çözüme ulaşmaya çalışmaktadır. Ancak bazen ağ farklı bir çözüme takılabilmekte ve performansı daha fazla iyileştirmek mümkün olmamaktadır.

Karmaşık bir ağda hata yüzeyi fazlasıyla inişli çıkışlıdır. Bu durum, dereceli azaltmadan dolayı ağın yerel minimum tuzağına düşmesiyle sonuçlanır. Olasılık yöntemleri bu tuzaktan kurtulmaya yardımcı olsa da oldukça yavaş kalmaktadır. Bir başka öneri, gizli katman veya gizli katmandaki nöron sayısının arttırılmasıdır. Bu yöntem işe yarasa da hata yüzeyinin boyut sayısını da arttırmaktadır. Gizli nöron sayısının bir üst limiti her zaman olacaktır, çünkü araştırmalar gizli nöron sayısının fazla olduğu durumlarda ağın ezberlediğini ortaya koymuşlardır.

Büyük öğrenme oranlarında salınıma imkan vermemenin bir yolu Momentum Katsayısının ($\alpha$) eklenmesidir.

Momentumun oynadığı rol Şekil 2.18'de gösterilmiştir. Eğer momentum terimi kullanılmazsa ağın düşük öğrenme katsayısı ile yerel minimuma ulaşması çok uzun sürmekte, büyük öğrenme katsayısında ise salınımdan dolayı yerel minimuma asla ulaşamamaktadır. Momentum kullanıldığında ise yerel minimuma hızlı bir şekilde ulaşılabilmektedir.

Özetle momentum, özellikle yerel çözümlere takılan ağların bir sıçrama ile daha iyi sonuçlar bulmasını sağlamak amacıyla kullanılmaktadır. Bu değerin küçük olması yerel çözümlerden kurtulmayı, büyük bir değer olması ise tek bir çözüme ulaşmayı zorlaştırabilmektedir.




Şekil 2.18. Ağırlık uzayındaki düşme: a) düşük öğrenme oranı, b) büyük öğrenme oranı: salınım, c) momentum terimi eklenmiş büyük öğrenme oranı [21]

### 2.11.7. Performans faktörleri

Performansın arttırılması şu faktörlere bağlıdır:

• Öğrenme algoritması ve iterasyon (epoch) sayısı: Hatanın minimize edilmesi için gereklidir. Örneklerin ağa tekrar tekrar gösterilmesi ile ağ hata oranını minimum yapmaya çalışmaktadır.

• Eğitim setindeki örnek sayısı: Örneklerin gerçek fonksiyonu temsil etmesi için gereklidir.

• Gizli işlemci eleman sayısı: Ağın gücüdür. Yumuşak dalgalı fonksiyonlarda oldukça az gizli elemana ihtiyaç duyulurken, sert inişli çıkışlı fonksiyonlarda daha fazla gizli eleman kullanılmalıdır.

İlk olarak doğru hata ölçütü tanımlanmalıdır. Tüm öğrenme algoritmaları, eğitim boyunca eğitim setinin hatalarını minimize etmeye çalışır. Eğitim setindeki her bir örnek için ortalama hata Öğrenme Hata Oranı olarak ifade edilir ve Eşitlik (2.25)'deki gibi formülüze edilir;

$$E_{eğitim}=\frac{1}{P_{eğitim}}\sum_{p=1}^{P_{eğitim}} E^{p} \qquad (2.25)$$

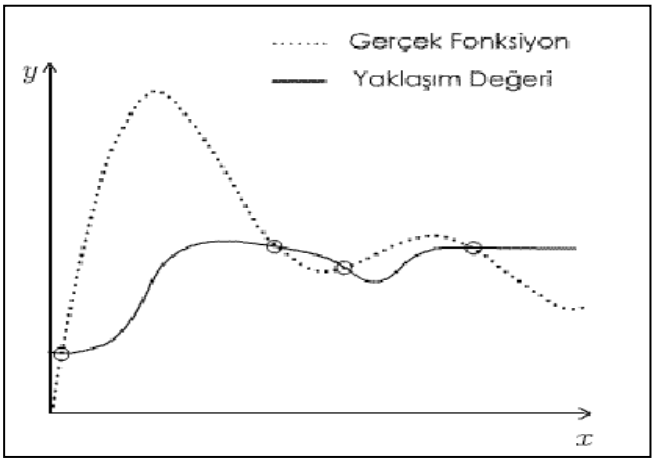
Daha önceki denklemlerde de ifade edildiği gibi, $E^p$ eğitim örneğinin gerçek çıktı değeri ile ağın çıktı değeri arasındaki farktır (Eşitlik (2.26)):

$$E^p=\frac{1}{2}\sum_{o=1}^{N_o}\left(d_o^p-y_o^p\right)^2 \tag{2.26}$$

Bu hata, eğitim süresince ölçülebilen bir hatadır ve açıkça görüldüğü gibi eğitim setindeki her bir örnek için farklı değerler almaktadır. Bu hata sistemin performansının doğru bir şekilde ölçülmesi açısından tüm örnekler için genişletilmeli ve sistem için tek bir hata ölçütü ortaya konmalıdır. Sonrasında test verilerinin de ortalama hata oranlarının hesaplanması gerekmektedir (Eşitlik (2.27)):

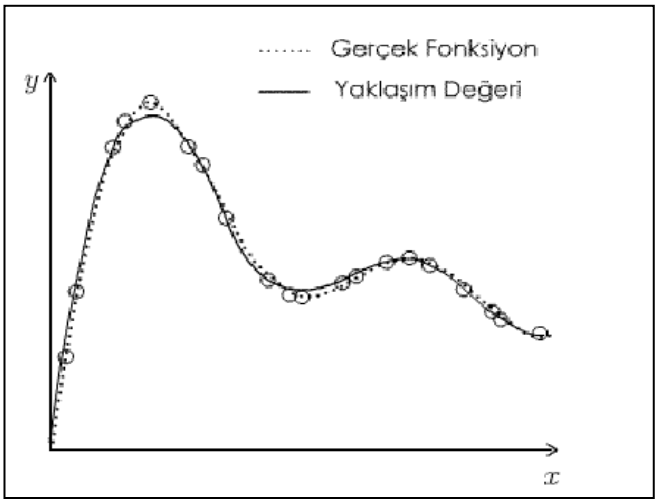
$$E_{Test}=\frac{1}{P_{Test}}\sum_{p=1}^{P_{Test}}E^p \tag{2.27}$$

Örnek Sayısının Performansa Etkisi: Eğitim setindeki örnek sayısı ne kadar fazla olursa, ağ, fonksiyon üzerindeki bilinen nokta sayısı fazla olacağından, fonksiyon hakkında daha yakın bir kestirim yapma imkanı bulacaktır. Bir *y = f(x)* fonksiyonunun ileri beslemeli bir ağ ile yaklaştırılacağı (function approximation) varsayılsın. Ağın, bir girdi elemanına, sigmoid aktivasyon fonksiyonlu 5 gizli elemana ve doğrusal aktivasyon fonksiyonlu bir çıktı birimine sahip olduğu ve az sayıda eğitim örneği ile eğitildiği varsayılsın. Şekil 2.19; 4 örnekle eğitilen böyle bir ağın, eğitim sonuç grafiğini göstermektedir.



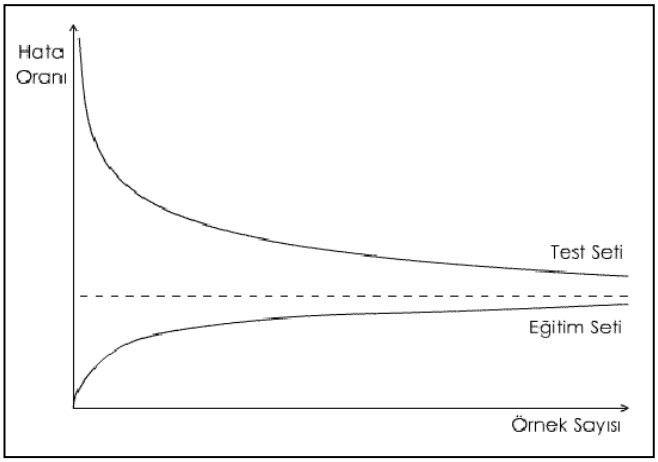

Şekil 2.19. Örnek sayısının fonksiyon yaklaştırma üzerindeki etkisi (4 örnek) [21]

Şekil 2.19'da gerçek fonksiyon, kesikli çizgi ile gösterilirken, eğitim sonrası yapay sinir ağının yaklaşımı düz çizgi ile gösterilmiştir. İki eğrinin kesim noktaları ağa öğrenme sürecinde verilen örneklerdir. Aynı ağa 4 yerine 20 örnek gösterildiğinde Şekil 2.20'deki grafik ortaya çıkacaktır.




Şekil 2.20. Örnek sayısının fonksiyon yaklaştırma üzerindeki etkisi (20 örnek) [21]

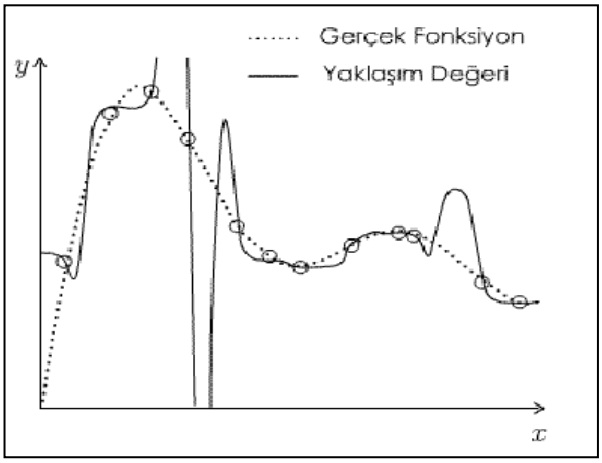
İlk durumda az örnekle $E_{Eğitim}$ oldukça az olacak ama $E_{Test}$ oldukça yüksek çıkacaktır. İkincisinde ise $E_{Eğitim}$ nispeten daha fazla çıksa da çok düşük $E_{Test}$ değeri ile karşılaşılacaktır. Aynı ağ, farklı örnek sayıları ile eğitildiğinde ve her örnek seti ağa onar defa gösterildiğinde, Şekil 2.21'deki ortalama öğrenme ve test hatası oranı grafiği ile karşılaşılacaktır.




Şekil 2.21. Örnek sayısı ile hata oranları arasındaki ilişki [21]

Görüldüğü gibi, eğitim hatası, örnek sayısı arttıkça artarken, test hatası düşmektedir. Eğitim hatasının düşük olması, ağın performansının iyi olduğunu her zaman göstermez. Eğitim setinin örnek sayısı arttırıldıkça her iki hata oranı da aynı değere yakınsamaktadır. Bu değer, ağın temsil gücü ile ilişkilidir. Bu değer aynı zamanda gizli eleman sayısına ve kullanılan aktivasyon fonksiyonuna bağlıdır. Eğer bu iki hata oranı aynı değere yakınsamazlarsa, eğitim süreci için mutlak minimum bulunması imkansızlaşır.

Gizli Nöron Sayısının Performansa Etkisi: Aynı fonksiyonun yaklaşımı için, aynı ağın kullanıldığını, ağın bu kez farklı gizli işlemci eleman sayılarıyla eğitildiği varsayılsın. Şekil 2.22; 20 gizli nörona sahip olması durumunda, eğitim sonrasındaki davranışını göstermektedir.

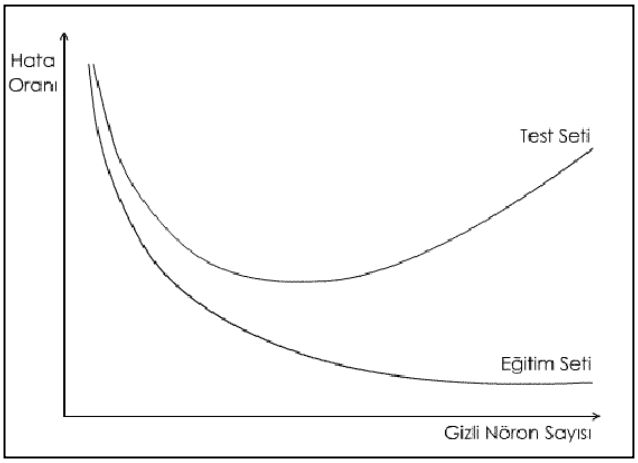



Şekil 2.22. Gizli nöron sayısının fonksiyon yaklaştırma üzerindeki etkisi (20 gizli nöron) [21]

Şekil 2.22'te ağ, örneklere tam olarak uymaktadır fakat fazla gizli eleman kullanıldığından, yaklaşım fonksiyonu, gerçek fonksiyondan çok daha sert iniş çıkışlara sahip olmaktadır. Gerçek hayatta, eğitim örnekleri gürültü (Noise) içermekte ve örnekte kullanılan fonksiyon gibi düzgün bir grafik çizmemektedir. B nedenle gizli katman sayısının fazlalığı gerçek dünyanın verileri için daha gerçekçi, daha doğru yaklaşımlar sağlayabilmektedir.

Bu örnekte de görüldüğü gibi gizli eleman sayısının fazlalığı eğitim hata oranının düşmesini sağlarken, test hata oranını arttıracaktır. Fazladan gizli eleman eklemek $E_{Eğitim}$'i her zaman düşürecektir. Fakat, $E_{Test}$ ilk başlarda azalacak, sonrasında

artacaktır. Aynı ağ, aynı eğitim seti fakat farklı gizli işlemci eleman ile test edildiğinde ortalama eğitim ve test hatası oranlarının grafiği Şekil 2.23'deki gibi olmaktadır [21].




Şekil 2.23. Gizli nöron sayısı ile hata oranları arasındaki ilişki [21]

### 2.11.8. Tasarımda karşılaşılabilecek problemler

Yapay sinir ağının ileriye doğru hesaplama kısmında, ilk iterasyonun yapılabilmesi için ağırlıklara değer atamak gerekmektedir. Ağırlıklara atanacak başlangıç değerleri ağın performansını yakından ilgilendirmektedir. İlgili literatürde bu değerlerin, gaussian dağılım gösteren, büyük değerler almayan rastgele sayılar olması gerektiği söylenmektedir. Genellikle $[-0.1, 0.1]$ aralığı kullanılmaktadır [16].

Öğrenme katsayısının belirlenmesi, istenilen hata değerine ulaşılması için yapılması gerekli iterasyon sayısını etkilemektedir. Öğrenme katsayısı [0,1] aralığında seçilebilir. Bu değer sıfıra yakın seçildiğinde istenilen hata değerine ulaşmak için daha fazla iterasyon yapılması gerekir. Diğer bir ifade ile yakınsama yavaş olur. Öğrenme katsayısının değeri arttıkça yapılması gerekli iterasyon sayısı azalır. Fakat büyük değerler YSA'nın öğrenmesi yerine ezberlemesine neden olmaktadır. "Yapay sinir ağı öğreniyor mu yoksa ezberliyor mu?" sorusunun cevabını bir örnek ile açıklamak yararlı olacaktır. Eğitim verileri için yapılacak $10^{-4}$ mertebesindeki bir hatanın uygulama için yeterli olacağını düşünelim. Bu durumda test verileri için yapılacak $[10^{-3},10^{-5}]$ aralığındaki hatalar seçilen öğrenme katsayısının uygun olduğunu gösterir. Fakat test verileri için $10^{-1}$ mertebesinde bir hata yapılıyorsa bu

durumda yapay sinir ağının öğrenmediğini, bunun yerine ezberlediğini söyleyebiliriz. Bu durumda öğrenme katsayısının küçültülmesi gerekir. Görüldüğü gibi öğrenme katsayısı küçük seçildiğinde yakınsama hızı düşmekte, büyük seçildiğinde ise öğrenme problemi doğmaktadır. Bu sorunu aşmak için öğrenme katsayısı adaptif olarak değiştirilebilir. Çözüm noktasından uzakta öğrenme katsayısı büyük seçilerek yakınsama hızlandırılabilir. Çözüm noktası civarında ise katsayı azaltılarak öğrenme problemi aşılabilir.

Hatanın değişiminde büyük salınımlar görülüyorsa, öğrenme katsayısının değeri küçültülmelidir.

Momentum katsayısı, yerel minimum noktalarında takılmayı önlemektedir. Ağırlıklardaki değişimin bir önceki değerleri momentum katsayısı ile çarpılarak yerel minimumlarda sıçrama sağlanır. Momentum katsayısı [0,1] aralığında seçilebilir. Bu katsayı küçük seçildiğinde daha iyi sonuçlar alındığı gözlenmiştir.

Yapay sinir ağlarında kullanılan aktivasyon fonksiyonları sebebi ile giriş ve çıkış değerlerinin ölçeklendirilmesi gerekebilir. Örneğin giriş kümesinde bulunan 10, 100 ve 1000 değerleri için tanjant hiperbolik aktivasyon fonksiyonu 1 değerlerini üretecektir. Aktivasyon fonksiyonlarının doymasını “saturation” önlemek için bu değerlerin ölçeklendirilmesi gerekmektedir. Çıkışta ise ölçeklendirme işlemi tersine çevrilmelidir. Lineer aktivasyon fonksiyonunu kullanan ağlar için ölçeklendirme işlemine gerek yoktur [20].

# 3. UYGULAMA: KALIP İMALATI YAPAN BİR İŞLETMEDE KALIP MALİYETLERİNİN YAPAY SİNİR AĞI YAKLAŞIMIYLA TAHMİN EDİLMESİ

Bu tez çalışması kalıp üretimi yapan bir firmada yapılmıştır. Çalışma, lastik kalıbı üretiminde maliyetlere etki eden faktörlerin kesin olarak belirlenmesi ve karmaşık dinamiklerinden dolayı oldukça değişken, etkileşimli bir yapıya sahip kalıp maliyetlerinin bulunması amacıyla yapılmıştır.

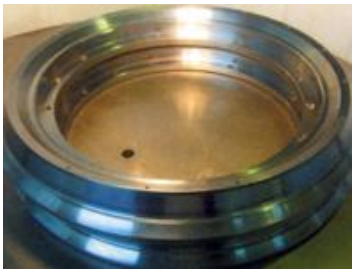
## 3.1. İşletme Tanıtımı

Firma makine imalatı ve lastik kalıbı sanayine mühendislik ve tamir bakım hizmeti veren bir işletmedir. Lastik sektörünün ihtiyacı olan lastik kalıpları, kalıp taşıyıcı konteynırları, lastik pişirme presleri, pişirme torbası kalıpları, iç lastik kalıpları, topuk ringleri, lastik imal tamburları, lastik pişirme preslerine ait hidrolik ve pnömatik sistemleri üretmekte, yurtiçi ve yurtdışı lastik üretimi yapan firmalara pazarlamaktadır. Kalıp üretiminin yanı sıra tamir bakım ve modifikasyon hizmetleri de vermektedir.

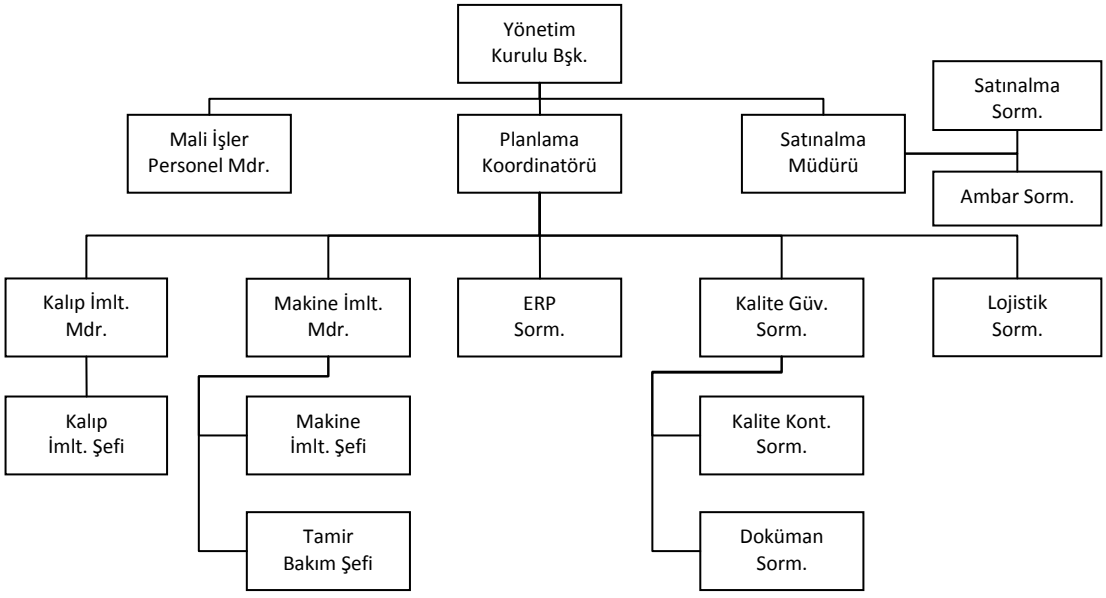
Tablo 3.1. İşletmede üretilen kalıp çeşitleri

| | |
|---|---|
| Yanak Lastik Kalıbı |  |
| Desen Lastik Kalıbı |  |

Tablo 3.1. (Devam) İşletmede üretilen kalıp çeşitleri

| | |
|---|---|
| Torba Kalıbı |  |
| Topuk Halkası |  |

Bu çalışmanın yapıldığı işletmenin organizasyon şeması Şekil 3.1'de verilmiştir.

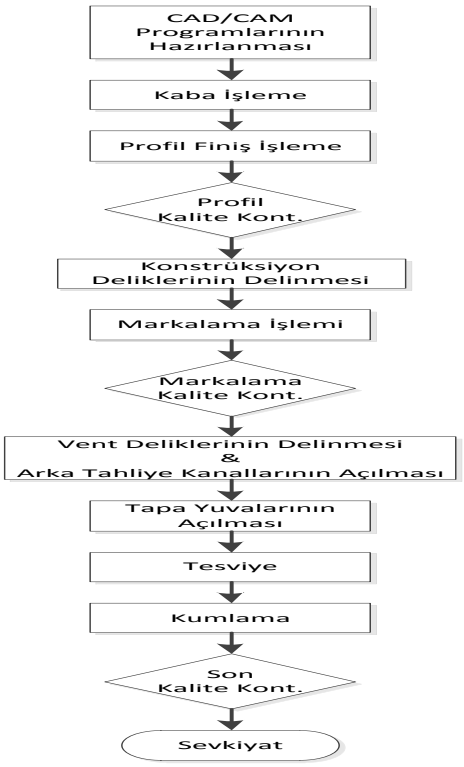



Şekil 3.1. İşletme organizasyon şeması

### 3.2. Uygulamaya Giriş

İşletmede yanak ve desen olmak üzere iki çeşit lastik kalıbı, iç lastik kalıpları, pişirme torbası kalıpları vb. kalıplar üretilmesine rağmen bu çalışmada sadece yanak lastik kalıbı maliyetleri tahmin edilmeye çalışılmıştır. Maliyetlere etki eden faktörlerin belirlenmesi için ilk önce; her kalıbın teknik resimlerine göre çapları ve kalınlıkları, paketlemede kullanılan kasa türleri ve sırasıyla üretim aşamaları incelenmiştir. Her kalıbın ebat ve desenlerine göre üretim süreleri incelenmiş ve üretim sürelerine göre desenlerin karmaşıklığı sınıflandırılmıştır. Veriler Excel dosyası olarak tablolaştırılmıştır. Ek-A'da YSA'nın eğitiminde kullanılan veriler, Ek-B'de ise YSA'nın deneysel veriler ile sınanmasında kullanılan veriler gösterilmektedir. Matlab programında girişler; alt yanak ve üst yanak çapları milimetrik değerlerle girilirken desen karmaşıklığı; Basit desen=1, Az karmaşık desen=2, Karmaşık desen=3, Çok karmaşık desen=4 olarak kasa türü ise; Ağaç kasa=1, H300 demir kasa=2, H360 demir kasa=3, H280 demir kasa=4, H368 demir kasa=5 olarak sayısallaştırılmıştır.

Şekil 3.2'de mevcut duruma ait yanak kalıbı imalat şeması gösterilmiştir.




Şekil 3.2. Lastik kalıbı iş akış şeması

Kaba işlem prosesinde CNC torna tezgahında, tedarikçiden alınan malzemenin çapları, profil finiş prosesinde ise yüzeyi işlenerek malzeme teknik resim ölçülerine getirilir. Yanak lastik kalıbı, başka elemanlarla birlikte lastik pişirme kazanına yerleştirildiğinde bunlarla birleşmesini sağlamak için torna tezgahında kalıp üzerine konstrüksiyon delikleri açılır. Kaba işlemleri tamamlandıktan sonra yazı tezgahlarına alınarak markalama işlemi yapılır. Yazı işlemleri biten kalıbın daha sonra, lastik pişirme esnasında hava tahliyesinin sağlanması için gerekli olan hava tahliye kanalları açılır. Kalıpların farklı ebatlar için kullanım çeşitliliğini sağlayan ve tapa olarak adlandırılan parçalar için kalıp üzerinde yuvalar açılıp, tesviye sırasında yuvalara yerleşmeleri sağlanır. Kumlama prosesinde ise özel bir malzeme olan cam küre adındaki kum ile üretimi biten kalıp üzerinde kalan çapaklar giderilir. Prosesler arasında ve son kalite kontrolü yapılan kalıp sevkiyata hazırlanarak imalat süreci tamamlanmış olur.

Lastik kalıbı maliyetlerinin tahmini için; son yıllarda sıkça kullanılan bir tahmin yöntemi haline gelen YSA kullanılmıştır.

Gerçeğe en yakın tahmin sonuçlarının alınabilmesi için YSA topolojisinin en iyisinin seçilmesi gerekmektedir.

## 3.3. Çalışma Metodolojisi

### 3.3.1. Kalıp maliyetlerine etki eden faktörlerin belirlenmesi

Lastik kalıplarının ebatları, desenleri ve işleme süreleri çok değişken olduğundan, bunlara bağlı olarak da kalıp maliyetlerinin tahmin edilmesi zorlaşmaktadır. Bu tahminleri yapabilmek için her şeyden önce uzun süreli ve nitelikli bir veri toplama işlemi gerçekleştirilmelidir.

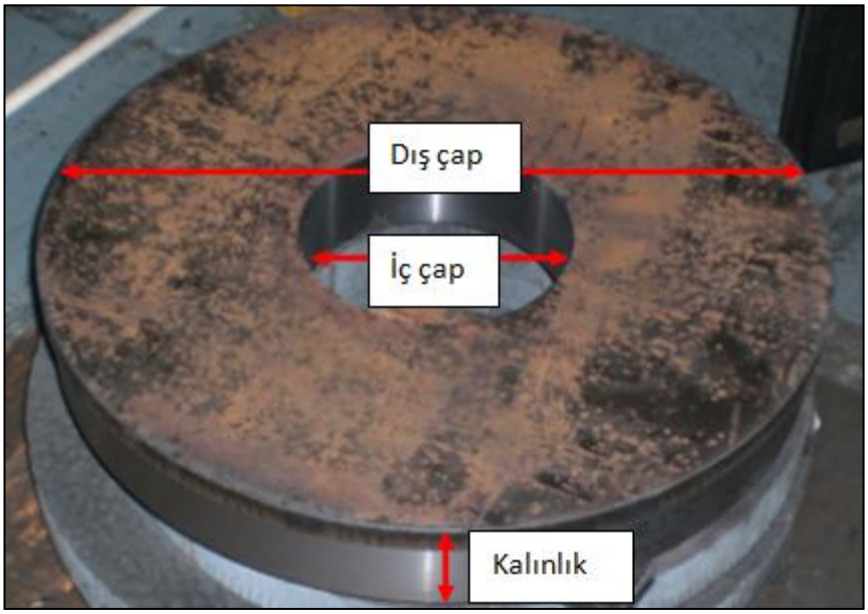
YSA'daki girdiler için kullanılacak her yanak kalıbının tüm geçmişi incelenerek, kalıp maliyetlerine etki eden uzman kişilerin de yardımıyla belirlenmiştir ve her bir kalıp için sekiz ana faktör üzerinde uzlaşılmıştır. Bu faktörler; alt yanak dış çapı, iç çapı, kalınlığı, üst yanak dış çapı, iç çapı, kalınlığı, desen karmaşıklığı ve kasa türüdür.

### 3.3.2. Veri toplama

YSA'nın hem eğitimi hem de performansının ölçülmesi için işletme verilerinden yararlanılmış, teknik resimlere göre ölçüler belirlenmiş, işlem süreleri ölçülmüş ve maliyetlere etki eden faktör sayısı yapay sinir ağının giriş hücre (neuron) sayısını oluşturmaktadır.

### 3.3.3. Veri setinin oluşturulması

Her kalıp için hesaplanan; alt yanak dış çapı (mm), iç çapı (mm), kalınlığı (mm); üst yanak dış çapı (mm), iç çapı (mm), kalınlığı (mm), desen karmaşıklığı ve kasa türü değerleri oluşturulan çok katmanlı, geri beslemeli, danışmanlı öğrenme özelliklerinde yapılandırılan YSA'na veri olarak girilmiştir (girdi vektörü). Şekil 3.3'de yanak kalıbı gösterilmektedir.




Şekil 3.3.Yanak kalıbı

Her bir kalıbın maliyeti (YTL), firmadan alınan verilere göre hesaplanmış ve oluşturulan ağa çıktı vektörü olarak tanıtılmıştır. Elde edilen veri setinin gösterimi Tablo 3.2'de verilmiştir.

Tablo 3.2. Analizde kullanılan yanak kalıbı parametreleri

| Parametreler | Kalıplar | | | | | |
|---|---|---|---|---|---|---|
| | Kalıp 1 | Kalıp 2 | Kalıp 3 | …… | Kalıp 79 | Kalıp 80 |
| Üst yanak dış çapı (mm) | 615 | 600 | 605 | …… | 645 | 645 |
| Üst yanak iç çapı (mm) | 285 | 285 | 285 | …… | 300 | 300 |
| Üst yanak kalınlığı (mm) | 70 | 70 | 65 | …… | 80 | 80 |
| Alt yanak dış çapı (mm) | 685 | 685 | 685 | …… | 645 | 645 |
| Alt yanak iç çapı (mm) | 330 | 340 | 340 | …… | 305 | 300 |
| Alt yanak kalınlığı (mm) | 70 | 70 | 65 | …… | 65 | 80 |
| Desen karmaşıklığı | 4 | 4 | 4 | …… | 4 | 2 |
| Kasa türü | 1 | 1 | 1 | …… | 2 | 2 |

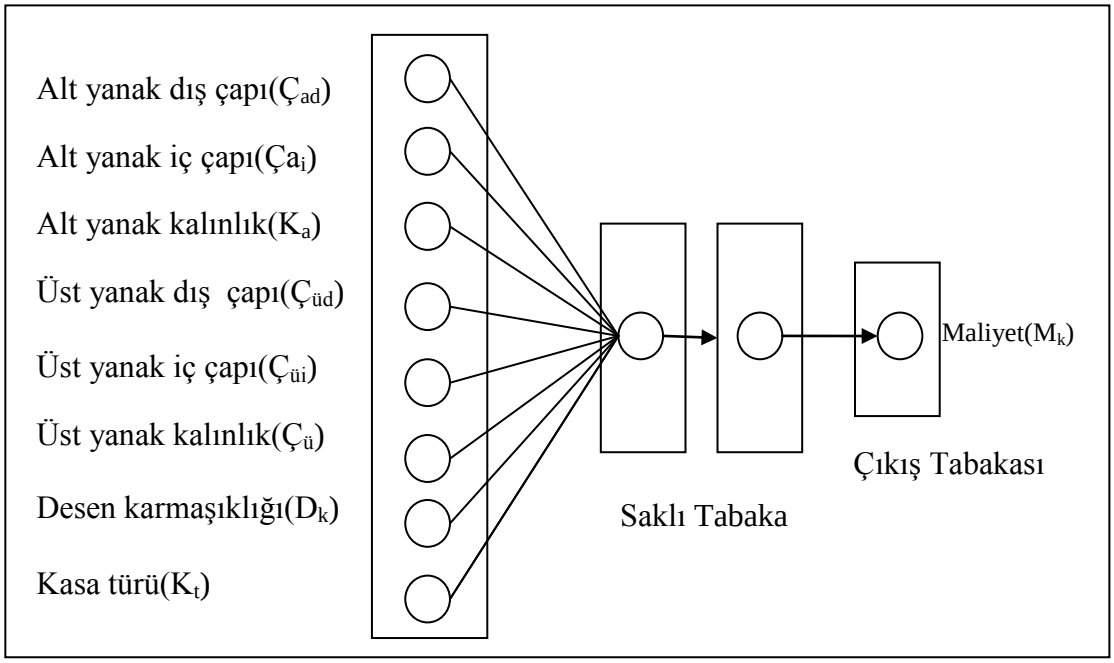
Yapılan öğretme ve test etme işlemlerinin ardından aynı normlardaki farklı kalıplara ait verilerin ağa girilmesi ile bu kalıplara ait maliyet tahminleri yaptırılmıştır. Bu tahminler RA yöntemiyle yapılan maliyet hesaplamaları ile karşılaştırılmış ve uygulanan YSA yönteminin sağladığı performans değerlendirilmiştir.

Verilerin kullanımı ile maliyetlerin hesaplanmasında MS Excel, ağın oluşturulmasında ise Matlab (7.0) yazılımından yararlanılmıştır. Regresyon analizleri için SPSS (17.0) programı kullanılmıştır.

## 3.4. Levenberg-Marquardt Öğrenme Algoritması Kullanılarak Yapay Sinir Ağının Modellenmesi

Kalıp maliyetlerinin tahmini için geliştirilen YSA modeli; giriş, çıkış ve 2 gizli katman olmak üzere dört katmandan oluşmaktadır. Her bir gizli katmandaki yapay sinir hücre sayısı birdir. Şekil 3.4'te ağın giriş parametreleri olarak gösterilen 8 adet giriş parametresi sırasıyla şöyledir; alt yanak dış çapı ($Ç_{ad}$), alt yanak iç çapı ($Ç_{ai}$), alt yanak kalınlığı ($K_a$), üst yanak dış çapı ($Ç_{üd}$), üst yanak iç çapı ($Ç_{üi}$), üst yanak kalınlığı ($K_ü$), desen karmaşıklığı ($D_k$) ve kasa türüdür ($K_t$). Giriş verilerine karşılık YSA'nın hesaplaması istenilen çıktı ise kalıp maliyet ($M_k$) değeridir.

Çalışmada kullanılan toplam 95 verinin 80 tanesi ağın eğitilmesi için, 15 tanesi ise ağın sınanması için ayrılmıştır. YSA'nın eğitilmesinde kullanılan eğitim verilerinin seçimi, ağın performansı ve sistem davranışının öğrenilebilmesi için çok önemlidir.




Şekil 3.4. Yapay sinir ağında kullanılan ağ topolojisi

## 3.5. Uygun YSA Konfigürasyonunun Belirlenmesi

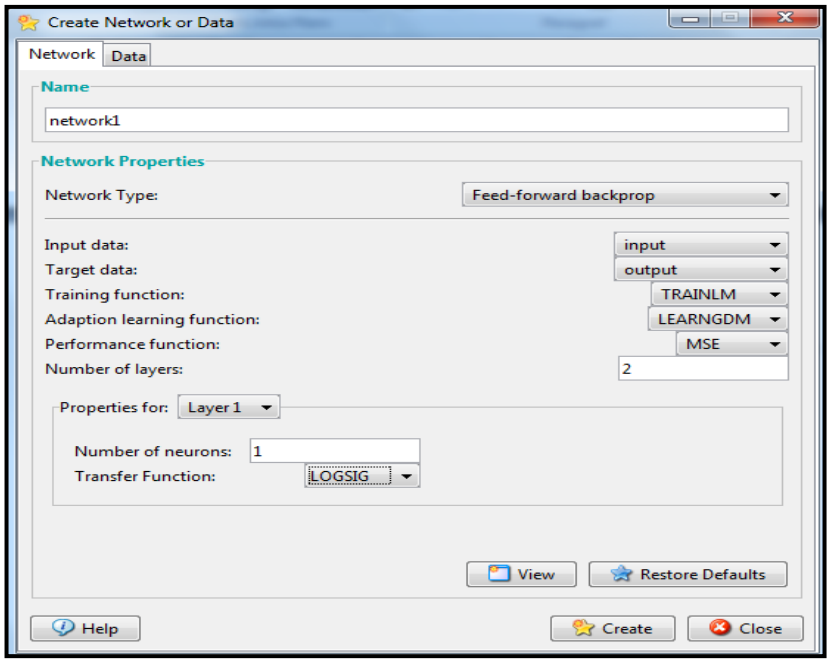
Çalışma için Matlab (7.0) programında çok fazla olasılık denemesi yapılmış ve en iyi sonuç veren yapay sinir hücreleri için kullanılacak aktivasyon fonksiyonunun belirlenmesinde kesin bir kural bulunmamaktadır. Bu nedenle, kalıp maliyeti tahmin sisteminin davranışına bağlı olarak gizli katman ve çıkış katmanındaki yapay sinir

hücrelerinde kullanılan aktivasyon fonksiyonları, YSA'nın eğitiminde elde edilen en az hata değerine göre belirlenmiştir. Uygulamada tercih ettiğimiz eğitim algoritması ise nümerik optimizasyon tekniklerini kullanan TrainLM (Levenberg-Marguardt) algoritmasıdır.

'Levenberg-Marguardt' öğrenme algoritması ortalama boyutta bir ağ için en hızlı öğrenme algoritması olarak belirlenmiştir. Eğitim seti genişlediğinde ise hafıza eksiltme özelliği vardır. Bu algoritma çok fazla hafıza kullanmakla beraber daha kısa sürede daha az epok (devir) ile sonuca ulaşmaktadır [2].

Levenberg-Marguardt metodunun adımları 'Matlab-Neural Network Toolbox-NNTool' programına girdi ve hedeflerin girilmesi, ağ tipi, girdi aralığı, öğrenme fonksiyonu, performans fonksiyonu, ara katman nöron sayılarının belirlenmesi ve eğitim - test verilerinin tanıtılarak ağırlıklandırma işleminin yapılmasıyla program tarafından otomatik olarak hesaplanacaktır. $\eta$ ; öğrenme oranı, $\alpha$ ; momentum katsayısı ve diğer parametreler hazır olarak programın öngördüğü şekilde alınmıştır. Öğrenme katsayısı ağırlıkların değişim miktarını, momentum katsayısı ise ağın öğrenmesi esnasında yerel bir optimum noktaya takılıp kalmaması için ağırlık değerinin belirli bir oranda bir sonraki değişime eklenmesini sağlar. Transfer (aktivasyon) fonksiyonu olarak doğrusal olmayan sigmoid transfer fonksiyonu kullanılmıştır.

Şekil 3.5'te Matlab (7.0) programında; eğitim, öğrenme ve aktivasyon fonksiyonlarının seçilmesi, gizli katman ve hücre sayısının veri girişleri gösterilmiştir.




Şekil 3.5. Ağ yapısının ve katmanların belirlenmesi

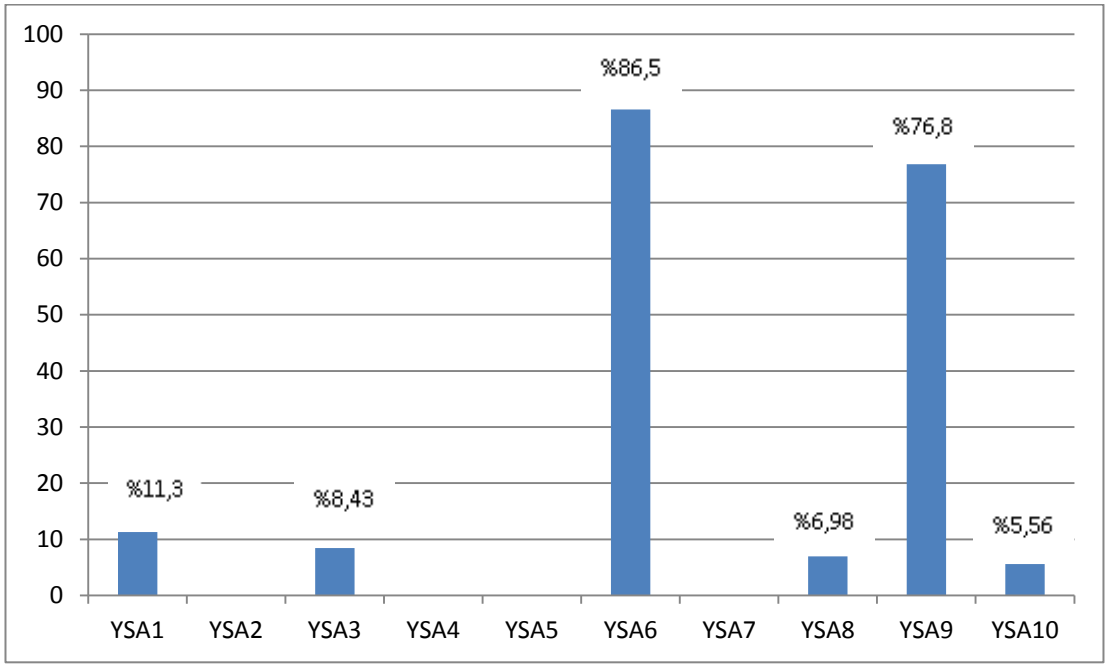
Eğitim fonksiyonu olarak Trainlm ve öğrenme fonksiyonu olarak Learngdm ile birlikte denemeler yapılmış ve bu denemelerden bazı örnekler Tablo 3.3'de gösterilmiştir.

Tablo 3.3. Farklı ağ yapılarının karakteristikleri

| **Örnek No** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| Gizli Katman Sayısı | 2 | 2 | 2 | 3 | 3 | 2 | 1 | 1 | 3 | 2 |
| Hücre (Neuron) Sayısı | 1 | 3 | 10 | 1 | 1 | 3 | 200 | 2 | 100 | 1 |
| Öğrenme Katsayısı | 0,5 | 0,5 | 0,4 | 0,4 | 0,4 | 0,4 | Prg | Prg | 0,6 | Prg |
| Momentum Katsayısı | 0,5 | 0,5 | 0,6 | 0,8 | 0,6 | 0,4 | Prg | Prg | 0,6 | Prg |
| Öğrenme Algorit. | GB | GB | GB | GB | GB | GB | GB | GB | GB | GB |
| Eğitim Fonk. | TGDM | TGDM | TGDM | TGDM | TGDM | TGDM | TLM | TLM | TGDM | TLM |
| Öğrenme Fonk. | LGDM | LGDM | LGDM | LGDM | LGDM | LGDM | LGDM | LGDM | LGDM | LGDM |
| Aktiv.Fonk. | LSIG | LSIG | LSIG | LSIG | LSIG | TSIG | LSIG | LSIG | TSIG | LSIG |
| % Sapma | 11,30 | - | 8,43 | - | - | 86,55 | - | 6,98 | 76,83 | 5,56 |

GB: Geri Beslemeli, LGDM: Leargdm, TRL: Trainlm, LSIG: Logsigmoid, TSIG: Tansigmoid, Prg: öğrenme ve momentum katsayıları manuel girilmemiş programın öngördüğü şekilde alınarak hesaplanmıştır.

Tablo 3.3'de ayrıca on farklı deneme ağının test girdilerine göre hesaplanmış sonuçlarının gerçek verilerden ortalama % sapma değerleri de verilmiş ve Şekil 3.6'da da % sapma değerler gösterilmiştir.




Şekil 3.6. Farklı topoloji ve özelliklerdeki ağların gerçek verilerinden ortalama % sapma oranları

Denemeler arasından Tablo 3.3'de gösterilmiş olan on ağdan gerçek verilerden ortalama % sapma oranı 5,56 ile en küçük değere sahip olan YSA10 adlı ağ deneyimizde kullanacağımız sonuç ağımız olmuştur.

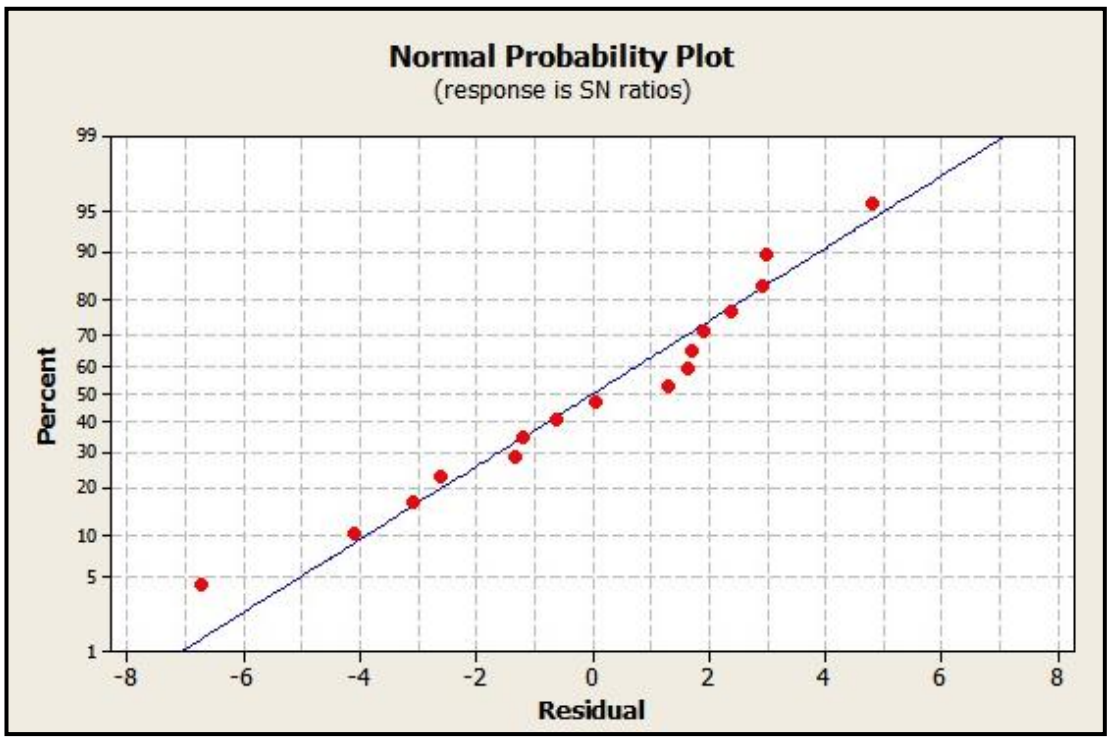
### 3.6. Yapay Sinir Ağının Performansının Ölçülmesi

Öğrenme performansını etkileyen önemli faktörlerden birisi de hata fonksiyonudur. Hata fonksiyonunun kullanımı ağın daha küçük ağırlık ve performans değerlerine sahip olmasına neden olur. Bu da ağın daha yumuşak davranmasına neden olarak, ağı aşırı öğrenme sürecinden kurtarır. Bu çalışmada en iyi sonuç ortalama hata yüzdesi ile alınmıştır.

### 3.7. Yapay Sinir Ağının Kullanımı ve Sonuçların Analizi

İşletme verilerinden alınan ve maliyet muhasebesiyle hesaplanmış 95 adet yanak lastik kalıbı maliyetinin 80 adeti YSA için eğitim verisi olarak kullanılmış ve 15

adeti de test verisi olarak kullanılmıştır. Matlab (7.0) da oluşturulan yapay sinir ağıyla elde edilen tahminlerinin gerçek değerlerin etrafındaki dağılımı gösteren normal dağılım grafiği Şekil 3.7'de gösterilmiştir. Yapay sinir ağı eğitim verileriyle birlikte olaylar arasındaki ilişkileri öğrenmiştir.




Şekil 3.7. Normal dağılım grafiği

YSA modelinin eğitimi için geriye yayılım algoritmasının Levenberg-Marquardt öğrenme algoritması kullanılmıştır.

Elde edilen yapay sinir ağının 80 adet eğitim verisi için üretmiş olduğu çıktılar ve gerçek değerlerle tahmini değerler arasındaki fark Tablo 3.4'de verilmiştir.

Tablo 3.4. Kalıp maliyetleri için elde edilen en iyi eğitim sonuçları

| No | Eğitim Verisi Çıktı | Eğitim Verisi Çıktı (YSA) | Sapma | Mutlak Sapma | % Sapma |
|---|---|---|---|---|---|
| **1** | 2755 | 2857 | -102 | 102 | 3,70 |
| **2** | 2734 | 2860 | -126 | 126 | 4,61 |
| **3** | 2771 | 2818 | -47 | 47 | 1,70 |
| **4** | 2382 | 2651 | -269 | 269 | 11,29 |
| **5** | 2763 | 2846 | -83 | 83 | 3,00 |

Tablo 3.4. (Devam) Kalıp maliyetleri için elde edilen en iyi eğitim sonuçları

| No | Eğitim Verisi Çıktı | Eğitim Verisi Çıktı (YSA) | Sapma | Mutlak Sapma | % Sapma |
|---|---|---|---|---|---|
| **6** | 2904 | 3019 | -115 | 115 | 3,96 |
| **7** | 2536 | 2298 | 238 | 238 | 9,38 |
| **8** | 2483 | 2370 | 113 | 113 | 4,55 |
| **9** | 2020 | 2078 | -58 | 58 | 2,87 |
| **10** | 2608 | 2846 | -238 | 238 | 9,13 |
| **11** | 2927 | 2948 | -21 | 21 | 0,72 |
| **12** | 2505 | 2576 | -71 | 71 | 2,83 |
| **13** | 2527 | 2869 | -342 | 342 | 13,53 |
| **14** | 2663 | 2873 | -210 | 210 | 7,89 |
| **15** | 2492 | 2393 | 99 | 99 | 3,97 |
| **16** | 2388 | 2312 | 76 | 76 | 3,18 |
| **17** | 2183 | 2259 | -76 | 76 | 3,48 |
| **18** | 2639 | 2768 | -129 | 129 | 4,89 |
| **19** | 2654 | 2798 | -144 | 144 | 5,43 |
| **20** | 3107 | 3113 | -6 | 6 | 0,19 |
| **21** | 2774 | 2806 | -32 | 32 | 1,15 |
| **22** | 2541 | 2835 | -294 | 294 | 11571 |
| **23** | 2569 | 2889 | -320 | 320 | 12,46 |
| **24** | 2427 | 2656 | -229 | 229 | 9,44 |
| **25** | 2646 | 2676 | -30 | 30 | 1,13 |
| **26** | 2980 | 2824 | 156 | 156 | 5,23 |
| **27** | 2951 | 2810 | 141 | 141 | 4,78 |
| **28** | 2044 | 2165 | -121 | 121 | 5,92 |
| **29** | 2737 | 2775 | -38 | 38 | 1,39 |
| **30** | 2600 | 2706 | -106 | 106 | 4,08 |
| **31** | 3285 | 3367 | -82 | 82 | 2,50 |
| **32** | 2238 | 2309 | -71 | 71 | 3,17 |
| **33** | 2664 | 2878 | -214 | 214 | 8,03 |
| **34** | 2569 | 2889 | -320 | 320 | 12,46 |
| **35** | 2453 | 2590 | -137 | 137 | 5,58 |
| **36** | 2279 | 2390 | -111 | 111 | 4,87 |
| **37** | 3040 | 2798 | 242 | 242 | 7,96 |
| **38** | 3660 | 2798 | 862 | 862 | 23,55 |
| **39** | 2654 | 2798 | -144 | 144 | 5,43 |
| **40** | 2515 | 2485 | 30 | 30 | 1,19 |
| **41** | 3041 | 2798 | 243 | 243 | 7,99 |
| **42** | 3297 | 2903 | 394 | 394 | 11,95 |
| **43** | 2852 | 2690 | 162 | 162 | 5,68 |
| **44** | 2414 | 2393 | 21 | 21 | 0,87 |
| **45** | 2182 | 2113 | 69 | 69 | 3,16 |

Tablo 3.4. (Devam) Kalıp maliyetleri için elde edilen en iyi eğitim sonuçları

| No | Eğitim Verisi Çıktı | Eğitim Verisi Çıktı (YSA) | Sapma | Mutlak Sapma | % Sapma |
|---|---|---|---|---|---|
| **46** | 2180 | 2042 | 138 | 138 | 6,33 |
| **47** | 2427 | 2656 | -229 | 229 | 9,44 |
| **48** | 2435 | 2379 | 56 | 56 | 2,30 |
| **49** | 2304 | 2315 | -11 | 11 | 0,48 |
| **50** | 2492 | 2393 | 99 | 99 | 3,97 |
| **51** | 2346 | 2297 | 49 | 49 | 2,09 |
| **52** | 2411 | 2341 | 70 | 70 | 2,90 |
| **53** | 2025 | 2088 | -63 | 63 | 3,11 |
| **54** | 2115 | 2110 | 5 | 5 | 0,24 |
| **55** | 2058 | 2136 | -78 | 78 | 3,79 |
| **56** | 3018 | 2771 | 247 | 247 | 8,18 |
| **57** | 2918 | 2846 | 72 | 72 | 2,47 |
| **58** | 2863 | 2792 | 71 | 71 | 2,48 |
| **59** | 4184 | 4116 | 68 | 68 | 1,63 |
| **60** | 2913 | 2851 | 62 | 62 | 2,13 |
| **61** | 2454 | 3541 | -1088 | 1088 | 44,34 |
| **62** | 2057 | 1922 | 135 | 135 | 6,56 |
| **63** | 1882 | 2488 | -606 | 606 | 32,20 |
| **64** | 1986 | 2069 | -83 | 83 | 4,18 |
| **65** | 2030 | 2121 | -91 | 91 | 4,48 |
| **66** | 2061 | 2134 | -73 | 73 | 3,54 |
| **67** | 2134 | 2052 | 78 | 78 | 3,66 |
| **68** | 2160 | 2261 | -101 | 101 | 4,68 |
| **69** | 1824 | 1960 | -16 | 136 | 7,46 |
| **70** | 2042 | 2127 | -85 | 85 | 4,16 |
| **71** | 2658 | 2503 | 155 | 155 | 5,83 |
| **72** | 1996 | 2147 | -151 | 151 | 7,57 |
| **73** | 1916 | 2161 | -245 | 245 | 12,79 |
| **74** | 2000 | 2061 | -61 | 61 | 3,05 |
| **75** | 2120 | 2089 | 31 | 31 | 1,46 |
| **76** | 2170 | 2061 | 109 | 109 | 5,02 |
| **77** | 1992 | 2053 | -61 | 61 | 3,06 |
| **78** | 2950 | 3248 | -298 | 298 | 10,10 |
| **79** | 2557 | 2745 | -188 | 188 | 7,35 |
| **80** | 2290 | 2093 | 197 | 197 | 8,60 |
| **Toplam** | | | | | 493,5 |

80 tane eğitim verisi için ortalama % sapma 6,1'dir.

Öğrenme verileriyle olaylar arasındaki ilişkileri öğrenen yapay sinir ağı daha sonra öğrendiği bilgileri kullanarak hiç görmediği test verileri için tahminlerde bulunmuştur.

Tablo 3.5. Kalıp maliyetleri için elde edilen en iyi test sonuçları

| No | Test Verisi Çıktı | Test Verisi Çıktı (YSA) | Sapma | Mutlak Sapma | % Sapma | No | Test Verisi Çıktı | Test Verisi Çıktı (YSA) | Sapma | Mutlak Sapma | % Sapma |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1972 | 2046 | -74 | 74 | 3,75 | **9** | 1939 | 2095 | -156 | 156 | 8,05 |
| **2** | 2130 | 2138 | -8 | 8 | 0,38 | **10** | 2540 | 2520 | 20 | 20 | 0,79 |
| **3** | 2201 | 2048 | 153 | 153 | 6,95 | **11** | 1885 | 1962 | -77 | 77 | 4,08 |
| **4** | 2128 | 1933 | 195 | 195 | 9,12 | **12** | 1884 | 2067 | -183 | 183 | 9,71 |
| **5** | 2414 | 2153 | 261 | 99 | 4,10 | **13** | 2574 | 2590 | -16 | 16 | 0,62 |
| **6** | 2232 | 2039 | 193 | 193 | 8,65 | **14** | 2022 | 2192 | -170 | 170 | 8,36 |
| **7** | 2250 | 2058 | 192 | 192 | 8,58 | **15** | 2083 | 1997 | 86 | 86 | 4,13 |
| **8** | 2015 | 2139 | -124 | 124 | 6,15 | **Toplam** | | | | | 83,42 |

YSA'nın hiç görmediği test verilerine karşı ürettiği sonuçlar ve gerçek veriler arasındaki toplam % sapma Tablo 3.5'te görüldüğü gibi 83,42 ve 15 adet test verisi için ortalama % sapma 5,56'dır.

### 3.8. Regresyon Analizi İle Maliyet Hesabı Yapılması

YSA yönteminin sonuçlarının test edilmesi amacı ile aynı verilere uygulanan RA'ya ait SPSS (17.0) programı kullanılarak elde edilen regresyon denklemi verileri Tablo 3.6'daki gibi hesaplanmıştır.

Tablo 3.6. Regresyon denklemi verileri

| | |
|---|---|
| Sabit | -1034,605 |
| Alt Yanak Dış Çap (a) | 1,914 |
| Alt Yanak İç Çap (b) | -1,083 |
| Alt Yanak Kalınlık (c) | 3,76 |
| Üst Yanak Dış Çap (d) | 1,55 |
| Üst Yanak İç Çap (e) | 0,481 |
| Üst Yanak Kalınlık (f) | 3,865 |
| Desen Karmaşıklığı (g) | 322,353 |
| Kasa Türü (h) | -33,19 |

SPSS Paket Programı ile yapılan RA çıktıları, Ek-D'de verilmiştir. Bu verilerin kullanılması ile Regresyon Denklemi, Eşitlik (3.1) de verilmiştir.

$$\Sigma\text{Maliyet} = (-1034{,}605)+(1{,}914a)+(-1{,}083b)+(3{,}76c)+(1{,}55d)+(0{,}481e)+(3{,}865f)$$
$$+(322{,}353g)+(-33{,}19h) \qquad (3.1)$$

Tablo 3.7. Regresyon analizi ile elde edilen test sonuçları

| No | Test Verisi Çıktı | RA Sonucu | Sapma | Mutlak Sapma | % Sapma | No | Test Verisi Çıktı | RA Sonucu | Sapma | Mutlak Sapma | % Sapma |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1972 | 2130 | -131 | 131 | 6,2 | **9** | 1939 | 2196 | -257 | 257 | 11,70 |
| **2** | 2130 | 2239 | -109 | 109 | 4,87 | **10** | 2540 | 1886 | 654 | 654 | 34,67 |
| **3** | 2201 | 2189 | 12 | 12 | 0,55 | **11** | 1885 | 2278 | -393 | 393 | 17,25 |
| **4** | 2128 | 1881 | 247 | 247 | 13,13 | **12** | 1884 | 2134 | -250 | 250 | 11,72 |
| **5** | 2414 | 2316 | 98 | 98 | 4,23 | **13** | 2574 | 3051 | -477 | 477 | 15,63 |
| **6** | 2232 | 2457 | -225 | 225 | 9,16 | **14** | 2022 | 2245 | -223 | 223 | 9,93 |
| **7** | 2250 | 2223 | 27 | 27 | 1,21 | **15** | 2083 | 2274 | -191 | 191 | 8,4 |
| **8** | 2015 | 2318 | -303 | 303 | 13,07 | **Toplam** | | | | | 161,72 |

Regresyon denkleminin test verilerine karşı ürettiği sonuçlar ve gerçek veriler arasındaki mutlak sapma 161,72'dir, 15 adet test verisi için ortalama % sapma 10,78'dir ve Tablo 3.7'de verilmiştir.

### 3.9. Maliyet Tahminine Yönelik Bulguların Karşılaştırılması

YSA ve RA ile yapılan hesaplamalar sonucu belirlenen maliyet tahminlerinin hata oranları incelenmiş ve sonuç olarak YSA'ların RA'ya göre daha az hata ile maliyet tahmini yaptığı görülmüştür. 15 adet test verisine karşı YSA'nın ürettiği maliyet tahminlerinde % sapma 5,56 iken RA ile elde edilen maliyet tahminlerinde % sapma 10,78 dir. Tablo 3.8'de her iki yöntemin de test verileri için "% Sapma" değerleri ve "% sapma değerlerinin ortalamaları" gösterilmiştir.

Tablo 3.8. RA ve YSA ile yapılan tahminlerin % sapma değerleri ve % sapma değerlerinin ortalamaları

Tablo 3.8. RA ve YSA ile yapılan tahminlerin % sapma değerleri ve % sapma değerlerinin ortalamaları

| Test Verileri | YSA Sapma(%) | Regresyon Sapma(%) |
|---|---|---|
| **1** | 3,75 | 6,2 |
| **2** | 0,38 | 4,87 |
| **3** | 6,95 | 0,55 |
| **4** | 9,12 | 13,1 |
| **5** | 4,1 | 4,23 |
| **6** | 8,65 | 9,16 |
| **7** | 8,58 | 1,21 |
| **8** | 6,15 | 13,1 |
| **9** | 8,05 | 11,7 |
| **10** | 0,79 | 34,67 |
| **11** | 4,08 | 17,25 |
| **12** | 9,71 | 11,72 |
| **13** | 0,62 | 15,63 |
| **14** | 8,36 | 9,93 |
| **15** | 4,13 | 8,4 |
| **Ortalama** | 5,56 | 10,78 |

YSA ile yapılan hesaplamalar sonucu belirlenen maliyet tahminlerinin sayısal değerleri birbirleriyle ve beklenen değerlerle karşılaştırılmıştır ve Tablo 3.9'da gösterilmiştir.

Tablo 3.9. Gerçekleşen maliyetlerle YSA ve beklenen değerlerin tahmin değerleri

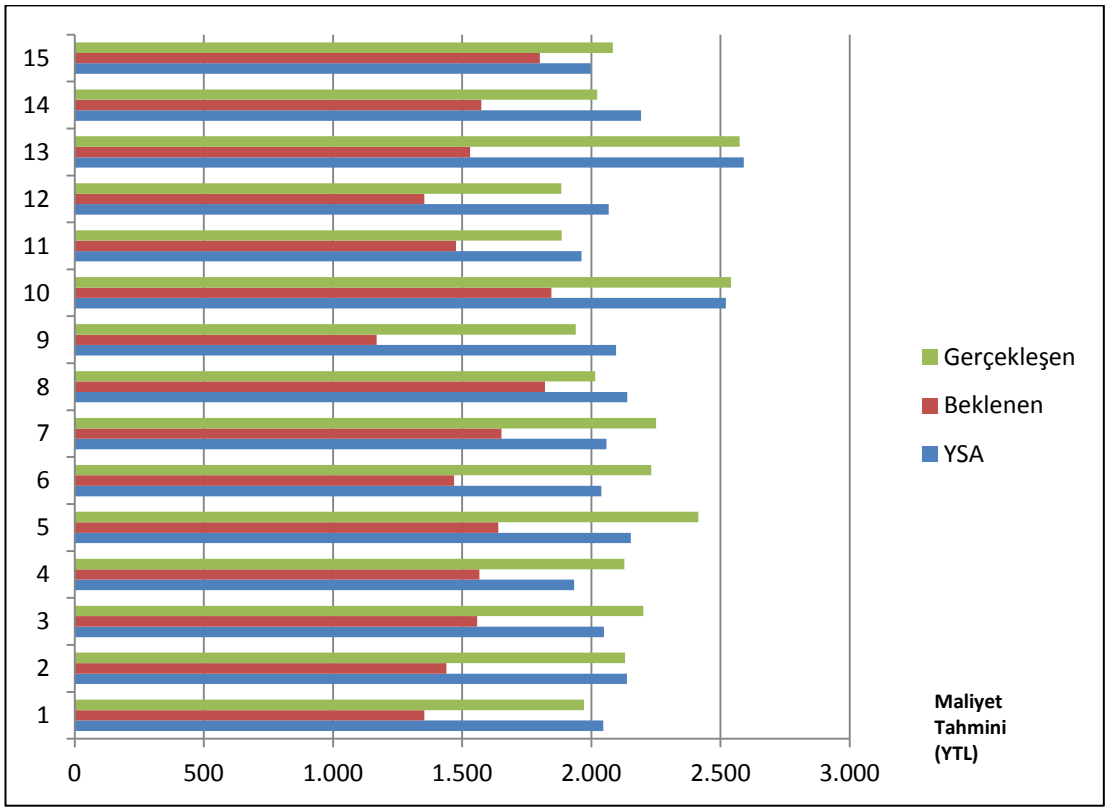
| Test Grup No | YSA | Beklenen | Gerçekleşen Maliyet |
|---|---|---|---|
| **1** | 2046 | 1353 | 1972 |
| **2** | 2138 | 1439 | 2130 |
| **3** | 2048 | 1558 | 2201 |
| **4** | 1933 | 1566 | 2128 |
| **5** | 2153 | 1640 | 2414 |
| **6** | 2039 | 1468 | 2232 |
| **7** | 2058 | 1652 | 2250 |
| **8** | 2139 | 1820 | 2015 |
| **9** | 2095 | 1169 | 1939 |
| **10** | 2520 | 1845 | 2540 |
| **11** | 1962 | 1476 | 1885 |
| **12** | 2067 | 1353 | 1884 |
| **13** | 2590 | 1531 | 2574 |

Tablo 3.9. (Devam)Gerçekleşen maliyetlerle YSA ve beklenen değerlerin tahmin değerleri

| Test Grup No | YSA | Beklenen | Gerçekleşen Maliyet |
|---|---|---|---|
| **14** | 2192 | 1574 | 2022 |
| **15** | 1997 | 1800 | 2083 |
| **G.M. Ort. % Sapma** | 5,56 | 39,5 | |

YSA ile elde edilen sonuçların şirket tahminlerine göre beklenen değerlerle gerçekleşen maliyet değerlerinin karşılaştırılması Tablo 3.9'da gösterilmiştir. Ortalama % sapma değerlerinden de görüldüğü gibi beklenen tahminlerle gerçekleşen maliyet değerlerinden ortalama %39,5 sapma olmuştur. YSA'nın ürettiği tahminler ise ortalama %5,56 gibi küçük bir sapma değerine sahiptir.

Her iki yöntemin verilerinin gerçekleşen maliyet değerleriyle karşılaştırılması Şekil 3.8'da grafiksel olarak gösterilmiştir.




Şekil 3.8. Her iki yöntemin verilerinin gerçekleşen maliyet değerleriyle karşılaştırılması

## SONUÇLAR VE ÖNERİLER

Maliyetlerin doğru ve müşteriye fiyat teklifi verilmeden önce verilmesi ve maliyetlerin düşürülmesi işletmeler için çok önemlidir. İşletmeler üretimi gerçekleştirmeden önce üretim maliyetlerinin bilinmesi gerekmektedir.

Bu çalışmada YSA yönteminin maliyet tahminlerinde de kullanılabileceği gösterilmektedir. YSA; insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirilmek amacı ile geliştirilen bilgisayar sistemleridir.

İnsan davranışlarının modellenmesi fikri üzerine kurulu olan yapay zeka bilimi, bilgisayar programlarına olaylar hakkında bilgi toplama, olaylar arasındaki ilişkileri öğrenebilme ve karar verebilme yeteneği kazandırmıştır. Çözümü aranan problemlerin matematik modeli olmasa dahi, bilgisayar programları tarafından sezgisel yaklaşımlar kullanılarak bu problemlere çözüm getirilebilmektedir [3].

Uygulama, lastik kalıbı üreten bir firmada kalıp maliyetlerinin tahmini için yapılmıştır. Bu yaklaşım kullanılması ile çok fazla çeşide sahip olan kalıp üretiminde, kısa sürede bütçelendirilmesi gereken projelerde, gerçeğe daha yakın ve daha az hesaplamayla maliyet tahmini yapılması mümkün olmuştur. İşletmede seri üretim olmadığı ve her siparişte farklı bir kalıp üretimi gerçekleştiğinden maliyetlerin önceden tahmin edilmesi oldukça zor ve zaman alıcıdır. Ayrıca çalışma öncesinde maliyetlere etki eden faktörlerin çoğu göz ardı edilmiş maliyet tahminleri tamamen sezgisel olarak yapılmaktadır.

Çalışmada; uzman kişilerle birlikte bir ürünün üretimine etki eden faktörler belirlenmiş ve çok fazla değişkenliğe sahip olan ürünler kendi aralarında sınıflandırılmış ve bir yıl boyunca bu faktörlerin kaydı tutulmuş ve bu bilgiler doğrultusunda muhasebe tarafından maliyet muhasebesi yöntemiyle gerçekleşen

maliyetler bulunmuştur. Buradan alınan değerler yapay sinir ağının eğitim verilerini oluşturmaktadır.

Çalışmanın metot kısmında yapay sinir ağı uygulamalarında en çok kullanılan geriye yayılım algoritmasının Levenberg-Marquardt algoritması kullanılarak, kalıp maliyet faktörlerine karşılık yeni bir kalıp üretiminde maliyet değerlerinin tahminini yapabilecek yapay sinir ağı modeli oluşturulmuştur. Modelin gösterdiği başarı, ağın hiç görmediği test verileriyle sınanmıştır.

Yapay sinir ağının yanı sıra var olan veriler için regresyon analizi de yapılmış ve yapay sinir ağının tahminleriyle regresyon denkleminin ürettiği sonuçlar karşılaştırılmıştır. YSA'nın, RA'ya göre daha az hata ile maliyet tahmini yapabildiği anlaşılmıştır (% 5,56 < % 10,8).

Yapılan literatür araştırmalarının sonucu; maliyet tahminlerinin yapıldığı çalışmalarda %5'lik bir hatanın uygun olduğu görülmüştür.

Elde edilen YSA çıktıları ile gerçek veriler ve beklenen değerlerle gerçek veriler arasındaki sapmalar hesaplanmış ve YSA'nın beklenen değerlere göre daha az hata ile maliyet tahmininde bulunduğu görülmüştür (%5,56 < %39,5).

Bu yöntem ile her farklı kalıbın maliyetine etki edebilecek tüm faktörler göz önüne alınarak maliyetler kısa sürede ve gerçeğe en yakın şekilde tahmin edilebilmektedir. Çalışma sonucunda, mevcut durumda hiçbir veri ve yöntem kullanılmaksızın yapılan maliyetlendirme çalışmaları yerine kullanılabilecek bir yöntem önerilmekte ve fiyat aşamasında maliyetlendirme işlemi yapılabilmektedir. Ayrıca maliyetlerin artışına sebep olan faktörlerin belirlenmesi ve iyileştirilmesi çalışmalarının da yapılabilmesi için işletmeye yol gösterilmektedir.

## KAYNAKLAR


[1] Baba F., Saç metal kalıpçılığında kesme, çekme ve birleşik kalıplarının maliyet analizi, Yüksek Lisans Tezi, Zonguldak Karaelmas Üniversitesi, Fen Bilimleri Enstitüsü, Zonguldak, 2005, 198214.

[2] Helvacı Ö., Santrifüj pompalarda yapay sinir ağı uygulamaları, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2007, 178963.

[3] Demirel Y., Toplu konut inşaat maliyetlerinin yapay sinir ağları ile tahmini, *Selçuk Üniversitesi Müh.-Mim. Fakültesi Dergisi,* 2007, **22**, 4.

[4] Uğur L. O., Yapı maliyetlerinin yapay sinir ağları ile analizi, Doktora Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2007, 201095.

[5] Özcan B., Yapay sinir ağı yaklaşımıyla, peçete makinesi imalatı yapan işletmede makine işleme sürelerinin tahmin edilmesi, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2007, 232730.

[6] Miko B., Viharos J., Artificial neural network approach for injection mould cost estimation, *Hungarian Academy of Science*, Budapest, Hungary, 2011.

[7] Demirkoparan F., Kaynar O., Taştan S., Ham petrol fiyatlarının yapay sinir ağları ile tahmini, *Ege Akademik Bakış*, 2010, **2**, 559-573.

[8] Elazouni A. M., Mohieldin Y. A., Nosair I. A., Estimating resource requirements at conseptual design stage neural Networks, *Journal of Computing in Civil Engineering*, 1997, **4**, 217-223.

[9] Kim G., An S., Kang K., *Building and environment,* 10th ed., Elsevier, Korea, 2004.

[10] Basheer I. A., Hajmeer, M., Artifical neural networks: fundamentals, computing, design and application, *Journal of Microbiological Methods*, 2000, **43**, 3-31.

[11] Weiss S. M., Kulikowski C. A., *Computer Systems that learn*, Morgan Kaufmann Publisher Inc., San Francisco, 1991.

[12] Stern H. S., *Neural networks in applied statistics. Technometrics*, 3th ed., Taylor&Francis Group, USA, 205-214, 1996.

[13] Wang S., An adaptive approach to market development forecasting, *Neural Computing& Applications*, 1999, **8**, 3-8.

[14] Yasdi R., Prediction of road trafic using a neural network approach, *Neural Computing & Application*,1999, **8**, 135-142.

[15] Liu X., An artifical neural network approach to assess Project cost and time risks at th front of projects, Master of Science Thesis, Department of Civil Engineering, The University of Calgary, Calgary,1998.

[16] Öztemel E., *Yapay sinir ağları*, Papatya Yayıncılık, İstanbul, 2003.

[17] Saraç T., Yapay sinir ağları, Seminer Projesi, Gazi Üniversitesi, Endüstri Mühendisliği Ana Bilim Dalı, Ankara, 2004.

[18] Baş N., Yapay sinir ağları yaklaşımı ve bir uygulama, Yüksek Lisans Tezi, Mimar Sinan Güzel Sanatlar Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2006, 184183.

[19] Yurtoğlu H., Yapay sinir ağları metodolojisi ile öngörü modellemesi: bazı makroekonomik değişkenler için Türkiye örneği, Uzmanlık Tezi, Ekonomi Modeller ve Stratejik Araştırmalar Genel Müdürlüğü, Ankara, 2005, DPT:2683.

[20] Öğücü O., Yapay sinir ağları ile sistem tanıma, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2006, 223555.

[21] Bayır F., Yapay sinir ağları ve tahmin modellemesi üzerine bir uygulama, Yüksek Lisans Tezi, İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul, 2006, 215542.

[22] Jain A. K., Mao J., Mohuddin K. M., Artifical neural networks: a tutorial, *Computer*,1996, **29**, 31-44.

[23] Vural B. B., Yapay sinir ağları ile finansal tahmin, Yüksek Lisans Tezi, Ankara Üniversitesi, Sosyal Bilimler Enstitüsü, Ankara, 2007, 208211.

[24] Özveren U., Pem yakıt hücrelerinin yapay sinir ağları ile modellenmesi, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2006, 182655.

[25] Ünal B., Bileşik kesitli akarsu yataklarında taşıma kapasitesinin tayini, Doktora Tezi, Çukurova Üniversitesi, Fen Bilimleri Enstitüsü, Adana, 2011, 275553.

[26] Asilkan Ö., Irmak S., İkinci el otomobillerin gelecekteki fiyatlarının yapay sinir ağları ile tahmin edilmesi, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi,* 2009, **14**, 375-391.

[27] Ceylan M., Kompleks değerli yapay sinir ağı ile algoritma geliştirilmesi ve uygulamaları, Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya, 2004, 153740.

[28] Anderson D., McNeill G., Artificial Neural Networks Technology , Kaman Science Corporation, http://ebookbrowse.com/anderson-d-mcneill-g-artificial-neural-networks-technology-1992-en-82s-pdf-d328389481 (Ziyaret tarihi:10 Mart 2012).

[29] Ripley B. D., Barnndoff-Nielsen O. E., Jensen J. L., Kendall W. S., Statistical Aspects of Neural Networks, in Networks and Chaos, Statistical and Probabilistic Aspects, http://www.google.com.tr/books?hl=tr&lr=&id=m12UR8QmLqoC&oi=fnd&pg=PR9&dq=Pattern+recognition+and+neural+networks&ots=aMMmlHYH_i&sig=8jwEhOGJrshcpcCQgIo84085zrM&redir_esc=y#v=onepage&q=Pattern%20recognition%20and%20neural%20networks&f=false (Ziyaret tarihi:10 Mart 2012).

[30] Kevin G., An Introduction to Neural Networks, Middx, http://ebookbrowse.com/kevin-gurney-an-introduction-to-neural-networks-pdf-d303236861 (Ziyaret tarihi: 15 Mart 2012)

[31] The MathWorks Inc., http://www.mathworks.com/help/nnet/ref/trainlm.html (Ziyaret tarihi:4 Nisan 2012)

[32] Uğuz S., Yapay Sinir Ağları-Matlab Uygulaması, http://ybssoftware.files.wordpress.com/2011/03/ysa_uygulama.pdf (Ziyaret tarihi: 15 Nisan 2012)

[33] Yu H., Wilamowski B.M., Levenberg-Marquardt Training, Intelligence Systems, http://www.eng.auburn.edu/~wilambm/pap/2011/K10149_C012.pdf (Ziyaret tarihi: 5 Mayıs 2012)

[34] Ranganathan A., The Levenberg-Marquartdt Algorithm, http://www.ananth.in/Notes_files/lmtut.pdf (Ziyaret tarihi: 5 Mayıs 2012)

# EKLER

**Ek-A**

## YSA'NIN EĞİTİMİNDE KULLANILAN VERİLER

| Veri No | Girişler | | | | | | | | Çıkış |
|---|---|---|---|---|---|---|---|---|---|
| | $Ç_{ad}$ | $Ç_{ai}$ | $Ç_{ka}$ | $Ç_{üd}$ | $Ç_{üi}$ | $K_ü$ | $D_k$ | $K_t$ | $M_k$ |
| 1 | 685 | 330 | 70 | 615 | 285 | 70 | 4 | 1 | 2755 |
| 2 | 685 | 340 | 70 | 600 | 285 | 70 | 4 | 1 | 2734 |
| 3 | 685 | 340 | 65 | 605 | 285 | 65 | 4 | 1 | 2771 |
| 4 | 675 | 330 | 55 | 675 | 285 | 55 | 4 | 1 | 2382 |
| 5 | 685 | 330 | 65 | 590 | 285 | 65 | 4 | 1 | 2763 |
| 6 | 700 | 250 | 60 | 650 | 300 | 60 | 4 | 1 | 2904 |
| 7 | 675 | 330 | 60 | 675 | 285 | 65 | 3 | 1 | 2536 |
| 8 | 645 | 330 | 65 | 645 | 285 | 85 | 3 | 1 | 2483 |
| 9 | 685 | 330 | 85 | 590 | 285 | 65 | 2 | 1 | 2020 |
| 10 | 685 | 330 | 65 | 590 | 285 | 65 | 4 | 1 | 2608 |
| 11 | 840 | 330 | 65 | 740 | 285 | 85 | 3 | 1 | 2927 |
| 12 | 640 | 330 | 85 | 575 | 285 | 60 | 4 | 1 | 2505 |
| 13 | 685 | 330 | 60 | 585 | 285 | 65 | 4 | 1 | 2527 |
| 14 | 685 | 330 | 70 | 600 | 285 | 70 | 4 | 1 | 2663 |
| 15 | 685 | 330 | 70 | 585 | 285 | 70 | 3 | 1 | 2492 |
| 16 | 640 | 330 | 70 | 540 | 285 | 70 | 3 | 1 | 2388 |
| 17 | 640 | 330 | 60 | 560 | 285 | 60 | 3 | 1 | 2183 |
| 18 | 590 | 285 | 65 | 685 | 330 | 65 | 4 | 1 | 2639 |
| 19 | 685 | 330 | 60 | 600 | 285 | 60 | 4 | 1 | 2654 |
| 20 | 600 | 350 | 110 | 780 | 425 | 60 | 4 | 1 | 3107 |
| 21 | 600 | 285 | 65 | 685 | 330 | 65 | 4 | 1 | 2774 |
| 22 | 685 | 330 | 65 | 600 | 285 | 65 | 4 | 1 | 2541 |
| 23 | 685 | 330 | 70 | 585 | 285 | 70 | 4 | 1 | 2569 |
| 24 | 640 | 330 | 60 | 575 | 285 | 60 | 4 | 1 | 2427 |
| 25 | 550 | 260 | 60 | 550 | 320 | 60 | 4 | 1 | 2646 |
| 26 | 645 | 240 | 95 | 670 | 270 | 95 | 4 | 1 | 2980 |
| 27 | 590 | 285 | 70 | 680 | 330 | 70 | 4 | 1 | 2951 |
| 28 | 645 | 285 | 65 | 640 | 330 | 65 | 2 | 1 | 2044 |
| 29 | 695 | 330 | 65 | 695 | 285 | 65 | 4 | 1 | 2737 |
| 30 | 670 | 330 | 65 | 670 | 285 | 65 | 4 | 1 | 2600 |
| 31 | 790 | 330 | 90 | 675 | 285 | 90 | 4 | 1 | 3285 |
| 32 | 640 | 330 | 70 | 545 | 285 | 70 | 3 | 1 | 2238 |
| 33 | 685 | 330 | 75 | 630 | 285 | 75 | 4 | 1 | 2664 |
| 34 | 685 | 330 | 70 | 585 | 285 | 70 | 4 | 1 | 2569 |
| 35 | 625 | 330 | 65 | 625 | 285 | 65 | 4 | 1 | 2453 |
| 36 | 685 | 330 | 70 | 590 | 285 | 70 | 3 | 1 | 2279 |
| 37 | 685 | 330 | 60 | 600 | 285 | 60 | 4 | 1 | 3040 |
| 38 | 685 | 330 | 60 | 600 | 285 | 60 | 4 | 1 | 3660 |
| 39 | 685 | 330 | 60 | 600 | 285 | 60 | 4 | 1 | 2654 |
| 40 | 535 | 285 | 50 | 680 | 330 | 50 | 4 | 1 | 2515 |
| 41 | 685 | 330 | 60 | 600 | 285 | 60 | 4 | 1 | 3041 |
| 42 | 615 | 285 | 70 | 685 | 330 | 70 | 4 | 1 | 3297 |

| Veri No | Girişler | | | | | | | | Çıkış |
|---|---|---|---|---|---|---|---|---|---|
| | $Ç_{ad}$ | $Ç_{ai}$ | $Ç_{ka}$ | $Ç_{üd}$ | $Ç_{üi}$ | $K_{ü}$ | $D_k$ | $K_t$ | $M_k$ |
| 43 | 640 | 330 | 65 | 575 | 285 | 65 | 4 | 1 | 2852 |
| 44 | 685 | 330 | 70 | 585 | 285 | 70 | 3 | 1 | 2414 |
| 45 | 685 | 330 | 70 | 585 | 285 | 70 | 2 | 1 | 2182 |
| 46 | 620 | 330 | 75 | 620 | 285 | 75 | 2 | 1 | 2180 |
| 47 | 640 | 330 | 60 | 575 | 285 | 60 | 4 | 1 | 2427 |
| 48 | 685 | 330 | 70 | 605 | 285 | 70 | 3 | 1 | 2435 |
| 49 | 640 | 330 | 70 | 535 | 285 | 70 | 3 | 1 | 2304 |
| 50 | 685 | 330 | 70 | 585 | 285 | 70 | 3 | 1 | 2492 |
| 51 | 640 | 330 | 65 | 530 | 285 | 65 | 3 | 1 | 2346 |
| 52 | 640 | 330 | 75 | 530 | 285 | 75 | 3 | 1 | 2411 |
| 53 | 640 | 330 | 75 | 530 | 285 | 75 | 2 | 1 | 2025 |
| 54 | 685 | 330 | 70 | 595 | 285 | 70 | 2 | 1 | 2115 |
| 55 | 685 | 330 | 80 | 595 | 285 | 80 | 2 | 1 | 2058 |
| 56 | 580 | 285 | 70 | 680 | 330 | 70 | 4 | 1 | 3018 |
| 57 | 685 | 330 | 65 | 590 | 285 | 65 | 4 | 1 | 2918 |
| 58 | 675 | 330 | 75 | 675 | 285 | 75 | 4 | 1 | 2863 |
| 59 | 1080 | 490 | 60 | 975 | 465 | 60 | 4 | 1 | 4184 |
| 60 | 685 | 330 | 65 | 585 | 285 | 65 | 4 | 1 | 2913 |
| 61 | 710 | 370 | 50 | 660 | 405 | 50 | 4 | 2 | 2454 |
| 62 | 690 | 380 | 75 | 690 | 245 | 75 | 1 | 3 | 2057 |
| 63 | 710 | 365 | 55 | 660 | 405 | 55 | 2 | 2 | 1882 |
| 64 | 660 | 295 | 45 | 660 | 295 | 60 | 2 | 2 | 1986 |
| 65 | 660 | 310 | 55 | 660 | 310 | 70 | 2 | 2 | 2030 |
| 66 | 660 | 310 | 60 | 660 | 310 | 75 | 2 | 2 | 2061 |
| 67 | 645 | 310 | 70 | 645 | 310 | 70 | 2 | 4 | 2130 |
| 68 | 660 | 350 | 65 | 660 | 350 | 80 | 2 | 2 | 2160 |
| 69 | 645 | 310 | 55 | 645 | 310 | 70 | 1 | 4 | 1824 |
| 70 | 660 | 310 | 65 | 660 | 310 | 75 | 2 | 2 | 2042 |
| 71 | 715 | 335 | 80 | 715 | 335 | 80 | 3 | 5 | 2658 |
| 72 | 645 | 350 | 55 | 645 | 350 | 70 | 2 | 4 | 1996 |
| 73 | 645 | 350 | 60 | 645 | 350 | 75 | 2 | 4 | 1916 |
| 74 | 645 | 310 | 60 | 645 | 310 | 70 | 2 | 4 | 2000 |
| 75 | 680 | 310 | 75 | 680 | 310 | 75 | 2 | 4 | 2120 |
| 76 | 645 | 310 | 60 | 645 | 310 | 70 | 2 | 4 | 2170 |
| 77 | 685 | 335 | 75 | 645 | 335 | 95 | 1 | 4 | 1992 |
| 78 | 685 | 335 | 90 | 685 | 335 | 90 | 4 | 2 | 2950 |
| 79 | 645 | 305 | 65 | 645 | 305 | 65 | 4 | 2 | 2557 |
| 80 | 645 | 300 | 80 | 645 | 300 | 80 | 2 | 2 | 2290 |

**Ek-B**

## YSA'NIN DENEYSEL VERİLER İLE SINANMASINDA KULLANILAN VERİLER

| Veri No | Girişler | | | | | | | | Çıkış |
|---|---|---|---|---|---|---|---|---|---|
| | $Ç_{ad}$ | $Ç_{ai}$ | $Ç_{ka}$ | $Ç_{üd}$ | $Ç_{üi}$ | $K_ü$ | $D_k$ | $K_t$ | $M_k$ |
| 1 | 645 | 300 | 75 | 645 | 300 | 75 | 2 | 4 | 1972 |
| 2 | 700 | 335 | 75 | 700 | 335 | 75 | 2 | 5 | 2130 |
| 3 | 645 | 290 | 85 | 645 | 290 | 85 | 2 | 4 | 2201 |
| 4 | 645 | 260 | 85 | 645 | 260 | 85 | 1 | 4 | 2128 |
| 5 | 645 | 310 | 95 | 645 | 310 | 95 | 2 | 2 | 2414 |
| 6 | 645 | 290 | 80 | 645 | 290 | 80 | 2 | 4 | 2232 |
| 7 | 645 | 290 | 90 | 645 | 290 | 90 | 2 | 4 | 2250 |
| 8 | 725 | 355 | 95 | 665 | 310 | 95 | 2 | 6 | 2015 |
| 9 | 690 | 315 | 70 | 645 | 290 | 70 | 2 | 2 | 1939 |
| 10 | 725 | 355 | 95 | 660 | 310 | 95 | 3 | 5 | 2540 |
| 11 | 710 | 300 | 75 | 630 | 210 | 75 | 2 | 5 | 1885 |
| 12 | 690 | 320 | 75 | 645 | 300 | 75 | 2 | 5 | 1884 |
| 13 | 675 | 330 | 55 | 675 | 285 | 55 | 4 | 2 | 2574 |
| 14 | 685 | 330 | 60 | 600 | 285 | 60 | 3 | 5 | 2022 |
| 15 | 670 | 270 | 95 | 645 | 240 | 95 | 2 | 5 | 2083 |

$D_k$=Desen karmaşıklığının Matlab'a sayısal girişleri şu şekildedir;

1= Basit desen

2= Az karmaşık desen

3= Karmaşık desen

4= Çok karmaşık desen

$K_t$=Kasa türünün Matlab'a sayısal girişleri şu şekildedir;

1= Ağaç kasa

2= H300 demir kasa

3= H360 demir kasa

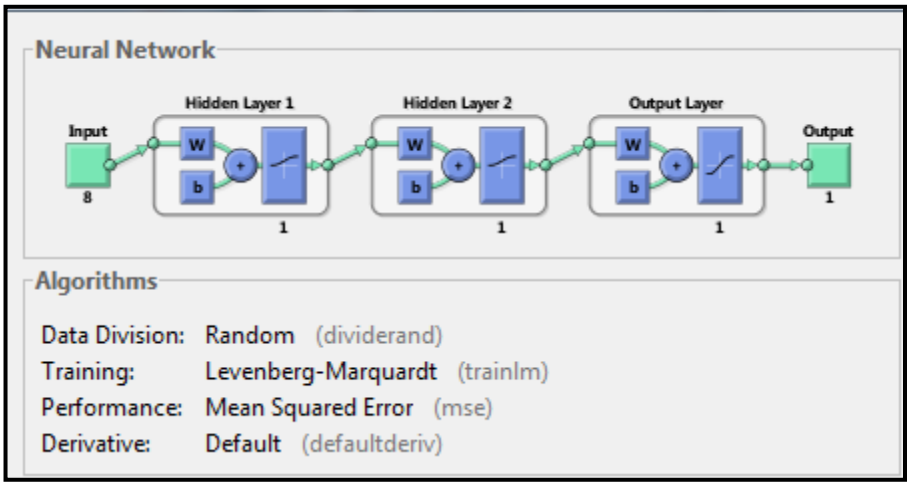
4= H280 demir kasa

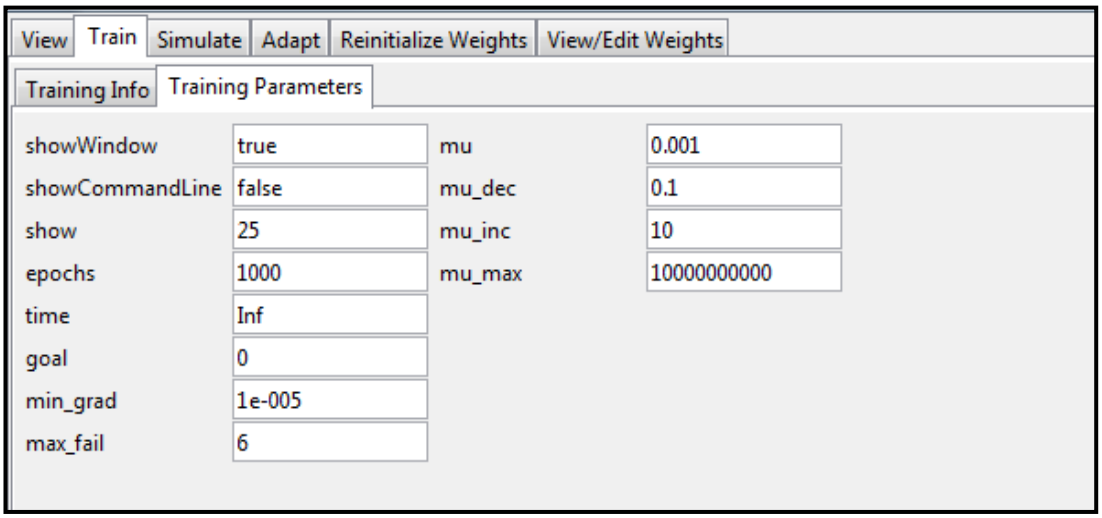
5= H368 demir kasa

6= H500 demir kasa

**Ek-C**

# KALIP MALİYET TAHMİNLERİ İÇİN KULLANILAN LEVENBERG MARGUARDT ALGORİTMASININ KULLANILDIĞI MATLAB PROGRAMI



**REGRESYON**

## Regression

**Variables Entered/Removed**

| Model | Variables Entered | Variables Removed | Method |
|---|---|---|---|
| 1 | kasaturu, adıscap, akalınlık, uiccap, desenkarmasıklı gı, aiccap, ukalınlık, udıscap[a] | . | Enter |

a. All requested variables entered.

**Model Summary[b]**

| Model | R | R Square | Adjusted R Square | Std. Error of the Estimate |
|---|---|---|---|---|
| 1 | ,896[a] | ,802 | ,780 | 197,19600 |

a. Predictors: (Constant), kasaturu, adıscap, akalınlık, uiccap, desenkarmasıklıgı, aiccap, ukalınlık, udıscap

b. Dependent Variable: toplammaliyet

**Coefficients[a]**

| Model | | Unstandardized Coefficients | | Standardized Coefficients | t | Sig. |
|---|---|---|---|---|---|---|
| | | B | Std. Error | Beta | | |
| 1 | (Constant) | -1034,605 | 394,170 | | -2,625 | ,011 |
| | adıscap | 1,914 | ,617 | ,287 | 3,100 | ,003 |
| | aiccap | -1,083 | 1,188 | -,077 | -,911 | ,365 |
| | akalınlık | 3,760 | 2,932 | ,091 | 1,282 | ,204 |
| | udıscap | 1,550 | ,563 | ,236 | 2,755 | ,007 |
| | uiccap | ,481 | ,972 | ,040 | ,495 | ,622 |
| | ukalınlık | 3,865 | 3,955 | ,081 | ,977 | ,332 |
| | desenkarmasıklı gı | 322,353 | 31,995 | ,727 | 10,075 | ,000 |
| | kasaturu | -33,190 | 30,726 | -,081 | -1,080 | ,284 |

a. Dependent Variable: toplammaliyet

**Residuals Statistics[a]**

| | Minimum | Maximum | Mean | Std. Deviation | N |
|---|---|---|---|---|---|
| Predicted Value | 1679,9790 | 3950,4448 | 2517,7125 | 376,14995 | 80 |
| Residual | -376,86948 | 960,11157 | ,00000 | 186,94494 | 80 |
| Std. Predicted Value | -2,227 | 3,809 | ,000 | 1,000 | 80 |
| Std. Residual | -1,911 | 4,869 | ,000 | ,948 | 80 |

a. Dependent Variable: toplammaliyet

## KİŞİSEL YAYIN VE ESERLER

[1] Baynal K., Gülkaç H., **Gürsoy A.,** Aktel A., Okuma hızını etkileyen önemli faktörlerin deney tasarımı ile optimizasyonu, *XI. Üretim Araştırmaları Sempozyumu,* Üretim Araştırmaları Derneği, İstanbul, 23-24 Haziran 2011.

## ÖZGEÇMİŞ

1987 yılında Balıkesir'de doğdu. İlk, orta ve lise öğrenimini Balıkesir'de tamamladı. 2005 yılında girdiği Süleyman Demirel Üniversitesi Mühendislik Mimarlık Fakültesi Tekstil Mühendisliği Bölümü'nden 2009 yılında mezun oldu. Eylül 2009'da Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı'nda Yüksek Lisans eğitimine başladı.