

**BAŐKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BENZER SÜREÇLERDE ÜRETİLEN ÜRÜNLER İÇİN
YAPAY ZEKA İLE ZAMAN TAHMİNİ**

SONER ŐÜKRÜ ALTIN

**YÜKSEK LİSANS TEZİ
2011**

**BENZER SÜREÇLERDE ÜRETİLEN ÜRÜNLER İÇİN
YAPAY ZEKA İLE ZAMAN TAHMİNİ**

**TIME FORECASTING WITH ARTIFICIAL INTELLIGENCE
FOR PRODUCTS WHICH ARE MANUFACTURED WITH
SIMILAR PROCESSES**

SONER ŞÜKRÜ ALTIN

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
ENDÜSTRİ Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

20011

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma, jürimiz tarafından **ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI 'nda**
YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Başkan : Doç. Dr. Hasan OĞUL

Üye (Danışman) : Doç. Dr. Ergün ERASLAN

Üye : Yrd. Doç. Dr. Y. Tansel İÇ

ONAY

Bu tez/...../..... tarihinde, yukarıdaki jüri üyeleri tarafından kabul edilmiştir.

/ / 2011

Prof.Dr. Emin AKATA
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRÜ

TEŐEKKÖR

Bugüne kadar her an benden desteęini esirgemeyen aileme ve yüksek lisans hayatım boyunca her türlü destek ve bilgisini paylaşan Sayın Doç. Dr. Ergün ERASLAN'a sonsuz teşekkür ederim.

ÖZ

Benzer Süreçlerde Üretilen Ürünler için Yapay Zeka İle Zaman Tahmini

Soner Şükrü ALTIN

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Standart süre, firmaların üretim ve kapasite planlarını yapmalarının temel şart olmakla birlikte ölçülmesi oldukça maliyetli, zaman alıcı ve ekip çalışması gerektiren bir çalışmadır. Bu durumda günümüzde alternatif ölçüm yöntemlerinin geliştirilmesi gerekliliği ortaya çıkmaktadır.

Standart zamanın yanlış tahmini ve bunun üzerinden yapılan iş planlaması, firmaları zamanından önce işi bitirmelerine ve makinelerin boшта kalmalarına ya da zamanında yetiştirmemeleri sonuçlarını ortaya çıkarabilir. İki durumda da firma maddi açıdan zarara uğrayabilir.

Yapay sinir ağları, günümüzde bilgisayar ve otomasyonun olduğu her alanda kullanılmakta olup, hızlı ve doğru sonuçlar vermektedir. Özellikle robot araştırmalarında ve akıllı sistemlerde kullanılan yapay sinir ağları, geçmiş bilgilere dayanarak herhangi bir olayı etkin tahmin etmek içinde kullanılmaktadır.

Bu çalışmada, zaman etüdü ile veri toplayan bir firmada ölçülemeyen diğer ürün grupları için standart zaman tahmini yapan alternatif bir iş ölçümü metodu geliştirilmiştir. Problem çözümünde geri beslemeli yapay sinir ağları kullanılmıştır ve kullanıcının kendi modelini tasarlayabileceği bir arayüz geliştirilmiştir. Bu yazılım kullanıcıya sonuçları detaylı bir şekilde vermektedir. Ayrıca sonuçları kaydetme ve grafiklerle göstermeye olanak sağlamaktadır.

ANAHTAR SÖZCÜKLER: yapay sinir ağları, zaman tahmini, geri beslemeli ağlar, çok katmanlı algılayıcı, iş etüdü, zaman etüdü.

Danışman: **Doç.Dr. Ergün ERASLAN, Başkent Üniversitesi, Endüstri Mühendisliği Bölümü.**

ABSTRACT

TIME FORECASTING WITH ARTIFICIAL INTELLIGENCE FOR PRODUCTS WHICH ARE FABRICATED WITH SIMILAR PROCESSES

Soner Şükrü ALTIN

Baskent University, Institute of Science

The Department of Industrial Engineering

Standart time is the basic requirement in production and capacity plans of the companies, although the measurement is very cost-effective, time consuming and requires a worw including teamwork. In this case, it is necessary today, the development of alternative methods of measurement.

If standart time predicted incorrectly and making future plans with using this estimation, company can finish work before the estimated and this cause to idle machines or company can not finish work until the deadline. Company loose money in both cases.

Recently neural networks are used in every fields where computer and otomation is used and neural networks gives accurate and fast results. Especially, neural networks is used in robotic researches and smart systems and it also is used for time forecasting with using past data.

In this study, an alternative time prediction method is developed for a company which collects time with time study. Backpropagation is used during solving this problem and an application which lets user to design his own neural network model. In addition application export results and graph desired model results with the user interface.

KEY WORDS: artificial intelligence, time forecasting, backpropagation, multi layer network, work study, time study.

Danışman: **Associate Professor Ergün ERASLAN, Baskent University, Department of Industrial Engineering.**

İÇİNDEKİLER LİSTESİ

| | <u>Sayfa</u> |
|---|--------------|
| ÖZ | i |
| ABSTRACT | ii |
| İÇİNDEKİLER LİSTESİ | iii |
| ŞEKİLLER LİSTESİ | v |
| ÇİZELGELER LİSTESİ | vi |
| SİMGELER ve KISALTMALAR | vii |
| 1 GİRİŞ..... | 1 |
| 2 İŞ ETÜDÜ..... | 3 |
| 2.1 İş Etüdü | 3 |
| 2.1.1 İş etüdü özellikleri | 3 |
| 2.1.2 İş etüdünün bölümleri..... | 4 |
| 2.1.3 İş etüdünün aşamaları | 4 |
| 2.2 Yapay Sinir Ağları..... | 5 |
| 2.2.1 Biyolojik sinir ağları | 5 |
| 2.2.2 Yapay sinir ağları | 7 |
| 2.2.3 Yapay sinir ağlarında öğrenme | 9 |
| 2.2.4 Örneklerden öğrenme | 9 |
| 2.2.5 Yapay sinir hücresi..... | 10 |
| 2.2.6 Çok katmanlı yapay sinir ağının yapısı | 10 |
| 2.2.7 ÇKA ileri doğru hesaplama | 12 |
| 2.2.8 ÇKA geriye doğru hesaplama | 13 |
| 2.2.8.1 Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi..... | 13 |
| 2.2.8.2 Ara katmanlar arası veya ara katman - girdi katmanı arasındaki ağırlıkların değiştirilmesi | 15 |
| 2.2.9 Aktivasyon fonksiyonları | 16 |
| 3 YAPAY SİNİR AĞLARI İLE ZAMAN TAHMİNİ | 18 |
| 3.1 Mevcut Durum ve Problemin Tanımı..... | 18 |

| | | |
|---------|--|-----------|
| 3.2 | Eđitim Seti | 19 |
| 3.2.1 | Girdiler ve ıktıların belirlenmesi | 19 |
| 3.2.2 | Verilerin toplanması ve verinin zellikleri | 20 |
| 3.2.3 | Girdiler ve ıktıların leklendirilmesi..... | 20 |
| 3.3 | Geri Yayılmalı Yapay Sinir Ađları Modelleri | 21 |
| 3.3.1 | ok katmanlı geri yayılım ađının alıřma prosedr | 23 |
| 3.4 | Yapay Sinir Ađları ile Zaman Tahmini Yapan Yazılımın Oluřturulması | 25 |
| 3.4.1 | Tasarlanan ađ yapısı | 25 |
| 3.4.2 | Geliřtirilen yazılım..... | 27 |
| 3.4.2.1 | Yazılım alıřma stratejileri..... | 28 |
| 3.4.2.2 | Yazılımın alıřtırılması | 30 |
| 3.4.2.3 | Yazılımın sonularının gsterilmesi..... | 31 |
| 3.4.3 | Yapay sinir ađları model sonuları | 37 |
| 4 | SONU VE NERİLER | 43 |
| | KAYNAKLAR LİSTESİ | 46 |
| | EKLER LİSTESİ | 48 |

ŞEKİLLER LİSTESİ

| | <u>Sayfa</u> |
|--|--------------|
| Şekil 2.1 Biyolojik sinir sisteminin blok gösterimi | 6 |
| Şekil 2.2 Biyolojik Sinir Hücresi ve Bileşenleri | 7 |
| Şekil 2.3 Çok katmanlı yapay sinir ağı | 11 |
| Şekil 3.1 Geri yayılım algoritması çalışma prosedürü | 25 |
| Şekil 3.2 Tasarlanan ağ yapısı | 26 |
| Şekil 3.3 Geliştirilen kullanıcı ara yüzü | 27 |
| Şekil 3.4 Yazılım çalışma stratejileri | 28 |
| Şekil 3.5 Verilen değerlerle ağ programlama | 28 |
| Şekil 3.6 Ara katman sayısına göre programlama | 29 |
| Şekil 3.7 Aktivasyon fonksiyonlarına göre programlama | 29 |
| Şekil 3.8 Tüm olasılıkları deneyen programlama | 30 |
| Şekil 3.9 Veri yüklendikten sonraki görünüm | 30 |
| Şekil 3.10 Veri yüklendikten sonra yazılımın görüntüsü | 31 |
| Şekil 3.11 Ana Sonuç Ekranı | 32 |
| Şekil 3.12 Toplam başarı grafiği | 34 |
| Şekil 3.13 En hızlı ve en az özyineleme modelleri yayılım grafiği..... | 35 |
| Şekil 3.14 Diğer örnek fonksiyonların yayılım grafikleri | 36 |
| Şekil 3.15 Sonuçları excel e kaydetme diyalogu..... | 37 |
| Şekil 3.16 Ara katman sayısının zamana etkisi | 41 |
| Şekil 3.17 Ara katman sayısının özyineleme sayısına etkisi..... | 41 |
| Şekil 3.18 Yayılım fonksiyonları | 42 |

ÇİZELGELER LİSTESİ

| | <u>Sayfa</u> |
|---|--------------|
| Çizelge 2.1 Aktivasyon fonksiyonları | 17 |
| Çizelge 3.1 Yazılım çalışma başlangıç değerleri | 38 |
| Çizelge 3.2 Örnek çözüm sonuçları | 39 |

SİMGELER VE KISALTMALAR

| | |
|-----------------------------|--|
| $x_1, x_2, x_3, \dots, x_n$ | n elemanlı girdi vektörü |
| y_m | m. çıktı |
| ζ_k^i | k. proses elemanının çıktısı |
| G_k | k. girdi katmanı örneği |
| NET_j^a | Proses elemanlarından gelen net girdi |
| A_{kj} | k. girdi katmanının elemanının j. ara katman elemanına bağlayan bağlantının ağırlık değeri |
| ζ_j^a | j. ara katman için hesaplanan çıktı |
| E_m | m. proses elemanı için oluşan hata |
| B_m | m. proses elemanı için beklenen çıktı |
| ζ_m | m. proses elemanı için hesaplanan çıktı |
| TH | Toplam Hata |
| ΔA^a | m. proses elemanına bağlayan bağlantının ağırlığındaki değişim miktarı |
| λ | Öğrenme katsayısı |
| α | Momentum katsayısı |
| δ_m | m. çıktı biriminin hatası |
| f' | Aktivasyon fonksiyonunun türevi |
| β^c | Proses elemanlarının eşik değer aralıkları |
| δ^a | Hata terimi |
| β^a | Ara katman eşik değer ağırlıkları |
| ÇKA | Çok Katmanlı Algılayıcı |
| YSA | Yapay Sinir Ağları |

1 GİRİŞ

Günümüzde bir işletmenin faaliyetlerini etkin bir şekilde yürütebilmesi için standart zamanlara büyük ihtiyaç vardır. Üretim plan ve programlarının hazırlanması, kısa ve uzun vadeli tahminler, maliyet kontrolü, ücretlendirme konuları başta olmak üzere işletme içinde yürütülen tüm faaliyetlerde standart zamanlara dayanmadan yapılacak işlerin tutarlı ve yararlı olması mümkün değildir. Buna karşılık, yapılan yoğun araştırmalara rağmen işleri düşük maliyetle ölçebilecek yöntem ve araçların henüz geliştirilemediği bir gerçektir. İş ölçümünün, uygulamasında karşılaşılan güçlüklerin pek çoğunun bu gerçeğe ilgili olduğu söylenebilir.

Standart sürelerin tespiti aşamasında ortaya çıkan zorluklar, zaman etüdü gibi doğrudan ölçüm yöntemlerine ek olarak alternatif zaman belirleme yöntemlerini de beraberinde getirmiştir. Ancak bu yöntemleri bütün işletmelerde her ürün ya da yarı ürünün standart süresinin tespitinde kullanılabilecek yöntemler olmadığı ve çeşitli yetersizlikleri bulunduğu bir gerçektir.

Yukarıda bahsedilen nedenlerden dolayı zamanların yapay zeka (makine öğrenmesi) ile tahmininde büyük kazançlar sağlanabilecektir. Makine öğrenme yöntemlerinden en çok kullanılan yöntemlerden biride yapay sinir ağlarıdır. Bu yöntem günümüzde pek çok sektörde uygulama alanı bulmuştur. Özellikle benzer süreçleri olan ve ürün yelpazesi çok geniş olan firmalarda, makine öğrenmesi yöntemi farklı algoritmalar ile denenerek tahmin doğruluğu en yüksek olan algoritma uygulanabilir. Böylece uzun zaman alıcı ve maliyetli iş ölçüm teknikleri yerine geliştirilen yazılımla zaman tahminleri çok yüksek bir doğrulukla elde edilebilir.

Zaman etüdünde amaç nitelikli bir işçinin, belli bir işi, belli bir çalışma hızıyla yapması için gereken zamanı saptamaktır. Tez kapsamında incelenen firma, zaman etüdü yapmak için ciddi kaynak ayırmış olmasına rağmen istediği verimliliğe ulaşamamıştır.

Yeni ürün için gelen talepler için sunulacak tekliflerde ise eski ürünler için tutulan zamanlar değerlendirilerek bir zaman tahmini ve maliyet çıkarılmaya çalışılmaktadır. Geçmişe yönelik insan deneyimden faydalanarak yapılan bu analizler ise her zaman doğru ve güvenilir bilgi vermediğinden ve bu analizlerin yapılması zaman aldığından dolayı, firma istediği kadar hızlı ve doğru tekliflere cevap verememektedir.

Çalışmada fabrika parça üretim biriminden alınan ölçümler ile yapay sinir ağları eğitime ile firmanın üretmesi gereken yeni parçalar için zaman hesabı çıkarabilecek bir ağ tasarlanmış ve çözülmüştür. Ek kabiliyet olarak ağın en kısa öğrenme yolu bulunmuş ve elde edilen diğer sonuçlar tartışılmıştır. Yapılan çalışmada kullanıcı ara yüzü tasarlanarak yapay sinir ağlarında değişik şekillerde ağ eğitimi için kullanıcıya izin verilmiştir.

Yapay sinir ağları çözülmesi zor problemlere hızlı ve hata oranı düşük sonuçlar vermektedir. Literatürde yapay sinir ağları kullanarak bilgisayar yardımıyla hızlı bir zaman tahmini yapılabileceğini ve bunun zaman etüdüne alternatif bir yöntem olduğunu gösteren çalışmalar yoktur.

Bu çalışmanın amacı zaman etüdü ile süre tahmini yapan firmalar için alternatif bir iş ölçüm yöntemi geliştirilmiştir.

Çalışma içeriğinin ikinci bölümünde, iş etüdü ve yapay sinir ağları ile ilgili bilgiler, ve bu konuyla ilgili çalışmalar ele alınmaktadır.

Üçüncü bölümünde ise ele alınan problemin çözümü için kullanılan yapay sinir ağları ile ilgili bilgiler, önerilen çözüm yöntemi, en hızlı çözüm için geliştirilen yöntem ve geliştirilen yazılımdan bahsedilmiştir.

Son olarak, sonuç ve öneriler bölümünde, çalışmadan elde edilen sonuçlar değerlendirilmektedir.

2 İŞ ETÜDÜ

Bu bölümde, çalışmada veri toplama aşamasında kullanılan iş etüdü yöntemi ve problem çözmede kullanılan yapay sinir ağları hakkında bilgiler verilmektedir.

2.1 İş Etüdü

2.1.1 İş etüdü özellikleri

İş Etüdü, gelişme olanağı yaratabilmek amacıyla, belirli bir olayı ya da etkinliği ekonomiklik ve etkenlik yönünden etkileyen tüm kaynakları ve etmenleri dizgesel olarak araştırmaya yönelik ve insan çalışmasını geniş kapsamda inceleyen bir teknik olup özellikle metot(yöntem) etüdü ve iş ölçümü teknikleri için kullanılan genel bir terimdir.

İş etüdünün kullanılmasının nihai amacı, verimliliği artırmaktır. Bu sonuca ulaşılması için birtakım alt amaçların gerçekleştirilmesi gereklidir. Bu amaçlar gerçekleştirilmeye çalışılırken uyulması gereken temel varsayımlardan biri de işlerin tanımlı fonksiyonlarından herhangi bir kayba izin verilmemesi gereğidir [1] [12]. Böylece, tanımlanmış bir iş ile ilgili faaliyetler, bu işin yapılmasından beklenen fonksiyonlar dikkate alınarak değerlendirilirler. Söz konusu amaçlar bu görüşler ışığında aşağıdaki gibi ele alınabilirler:

- Gereksiz faaliyetlerden kurtulmak
- Gerekli faaliyetleri mümkün olan en ekonomik şekilde düzenlemek:
- Uygun çalışma yöntemlerini standartlaştırmak
- İş ile ilgili doğru zaman standartlarını saptamak
- Üretimde kullanılan faktörlerden yararlanma oranını artırmak
- İşgücünü eğitmek
- Mevcut çalışma koşullarından daha iyi çalışma koşullarına geçme

İş etüdü çalışması ağırlıklı olarak emek faktörü ve onunla ilgili faaliyetleri kapsar. Tanımlı bir işin yada faaliyetin gerçekleştirilmesinde nitelik, süre, miktar

ve benzeri nedenlere göre belirleyici rol oynayan faktöre, o iş yada faaliyet için baskın faktör denir. Buna göre iş etüdü, yukarıdaki amaçları gerçekleştirmek üzere emek baskın faaliyetlerdeki analizlerde kullanılır. Ancak işin yapılmasında emekle birlikte bir üretken birim kombinasyonu oluşturan diğer üretim faktörleri(malzeme, araç-gereç, tezgah ve yerleşim gibi) üzerinde de dolaylı analizler gerçekleştirilebilir. Bu tür analizlerde yapılacak değişikliklerin boyutu, alacağı süre ve maliyeti önemli bir sınırlama ile karşı karşıyadır. Tek başına iş etüdü çalışmasıyla, emek dışı faktörlerde radikal değişiklikler yaratacak politika ve uygulamalardan kaçınılmalıdır. Zira iş etüdü ile örneğin iş yerinin yeniden düzenlenmesi, tezgah veya makine yenilenmesi yada teknolojik değişiklik yaratması gibi maliyeti yüksek, uzun sürede gerçekleştirilebilecek kararların alınması doğru değildir.

2.1.2 İş etüdünün bölümleri

İş etüdü konusunda yapılan tanımlar incelendiğinde görülecektir ki; etüt iki ana bölümden oluşmaktadır. Bunların ilki metot etüdü, ikincisi ise iş ölçümü olarak adlandırılır. Teoride bu bölümler, art arda gerçekleştirilmesi gereken ana aşamalar olarak ifade edilmesine rağmen uygulamada, önceden tanımlanmış işler için gerektiğinde doğrudan iş ölçümü yapılabilmektedir. Zira metot etüdüne göre daha kolay ve sonuçları itibariyle daha analitik bir yapısı vardır. Halbuki, üzerinde metot etüdü çalışması yapılan işlerin, verimliliği ne kadar değiştirdiğini ölçebilmek için yeni bir iş ölçümü çalışması yapmak kaçınılmazdır. Dolayısıyla belirli bir iş yada sürecin, iş etüdü tekniği ile incelenmesinden beklenen amaçların özelliğine göre bu bölümlerden; farklı sıralarda, farklı defalarda ve farklı oranlarda yararlanılabilir. Bu projede, projenin amacına da en uygun olan metot etüdünün ana bölümleri ile iş ölçümünün doğrudan ölçme bölümündeki zaman etüdü tekniği üzerinde ağırlıklı olarak durulacaktır.

2.1.3 İş etüdünün aşamaları

İş etüdü Taylorizm olarak da bilinen bilimsel yönetim prensiplerini uygulayarak verimliliği artırmaya yönelik problemleri çözmeye çalışır. Doğal olarak tekniğin kendine özgü yapısı gereği bu adımların sayısı değişecek fakat temel felsefe aynen korunacaktır. Metot etüdü ve iş ölçümü bölümlerini kapsayan tam bir iş

etüdü çalışması, sekiz ana asamadan oluşur [2]. Bunlar:

- Etüdü yapılacak işin yada sürecin seçimi.
- Seçilen işin yada süreçle ilgili bilgilerin toplanması ve uygun kayıt ortamlarına kayıt edilmesi.
- Kaydedilen olayların eleştirel bir gözle incelenmesi ve irdelenmesi.
- Analiz sonucunda en ekonomik yöntemin geliştirilmesi.
- Geliştirilen yöntemin kapsadığı iş miktarının ölçülmesi ve standart zamanın hesaplanması.
- Yeni yöntemin ve buna bağlı olarak standart zamanın tanımlanması.
- Yeni yöntemin ve sürenin onaylanarak, standart uygulama olarak sisteme yerleştirilmesi.
- Yeni standartların iyi bir denetimle sürdürülmesi.

olarak haddelenebilir.

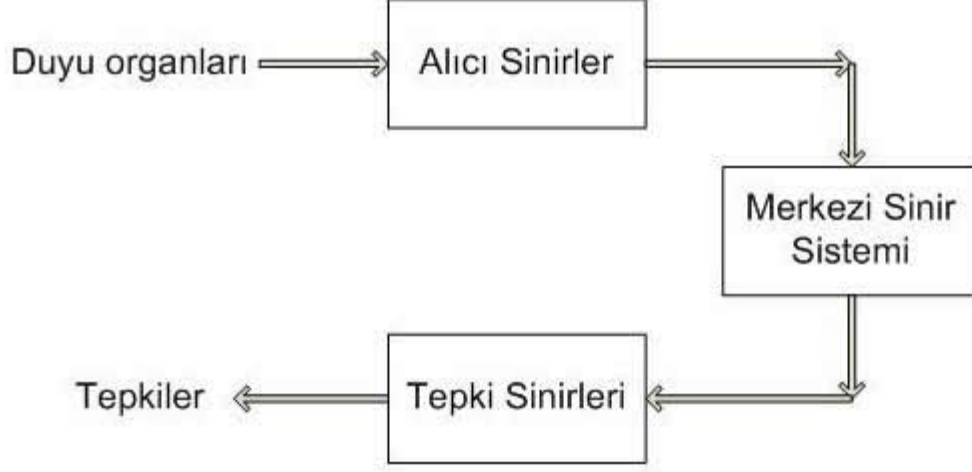
2.2 Yapay Sinir Ağları

2.2.1 Biyolojik sinir ağları

Canlı hücrelerin en önemli özelliği, kimyevi potansiyel enerjiyi kendi organize yapılarını korumak için gerekli diğer enerji şekillerine çevirebilmeleridir. Her hücre entropisinin artmasına yani dağılmasına mani olmak için enerji sarf etmek zorundadır. Çevredeki değişiklikleri, bunların kendi üzerine tesir derecesini, yani, düzenini, dengesini, ne dereceye kadar bozabileceğini bilmesi, gerekli ayarlamaları yapması, bu ayarlamaların da ne dereceye kadar hedefe uygun olduğunu ölçmesi ve gereken düzeltmelerde bulunması, kısaca çevreye uyumunu sağlaması, canlılığını devam ettirebilmesi için bir “Haber Alma- Karar Verme- İcra” sistemine ihtiyaç vardır. Sinir sistemi denilen fevkalade farklılaşmış, canlı doku bu önemli vazifeyi yerine getirmektedir [3].

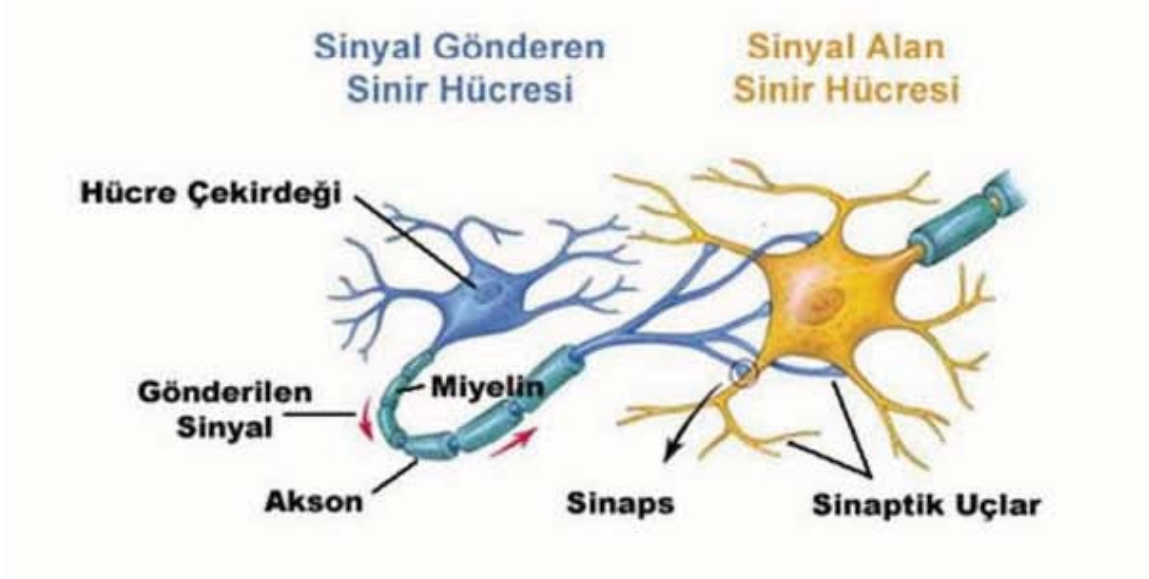
Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin (merkezi sinir ağı) bulunduğu 3 katmanlı bir sistem olarak açıklanır. Alıcı sinirler organizma içerisinden ya da dış ortamlardan algıladıkları uyarıları, beyne bilgi ileten elektriksel sinyallere

dönüştürür. Tepki sinirleri ise, beyinin ürettiği elektriksel darbeleri organizma çıktısı olarak uygun tepkilere dönüştürür Şekil 2.1'de bir sinir sisteminin blok gösterimi verilmiştir.



Şekil 2.1 Biyolojik sinir sisteminin blok gösterimi

Merkezi sinir ağında bilgiler, alıcı ve tepki sinirleri arasında ileri ve geri besleme yönünde değerlendirilerek uygun tepkiler üretilir. Bu yönüyle biyolojik sinir sistemi, kapalı çevrim denetim sisteminin karakteristiklerini taşır. Merkezi sinir sisteminin temel işlem elemanı, sinir hücresidir (nöron) ve insan beyinde yaklaşık 10 milyar sinir hücresi olduğu tahmin edilmektedir. Sinir hücresi; hücre gövdesi, dendritler ve aksonlar olmak üzere 3 bileşenden meydana gelir. Dendritler, diğer hücrelerden aldığı bilgileri hücre gövdesine bir ağaç yapısı şeklinde ince yollarla iletir. Aksonlar ise elektriksel darbeler şeklindeki bilgiyi hücreden dışarı taşıyan daha uzun bir yoldur. Aksonların bitimi, ince yollara ayrılabilir ve bu yollar, diğer hücreler için dendritleri oluşturur. Şekil 2.2'de görüldüğü gibi akson - dendrit bağlantı elemanı sinaps olarak söylenir.



Şekil 2.2 Biyolojik Sinir Hücresi ve Bileşenleri

Sinapstan gelen ve dendritler tarafından alınan bilgiler genellikle elektriksel darbelerdir ancak, sinapsdaki kimyasal ileticilerden etkilenir. Belirli bir sürede bir hücreye gelen girişlerin değeri, belirli bir eşik değerine ulaştığında hücre bir tepki üretir. Hücrenin tepkisini artırıcı yöndeki girişler uyarıcı, azaltıcı yöndeki girişler ise önleyici girişler olarak söylenir ve bu etkiyi sinaps belirler.

İnsan beyninin 10 milyar sinir hücresinden ve 60 trilyon sinaps bağlantısından oluştuğu düşünülürse son derece karmaşık ve etkin bir yapı olduğu anlaşılır. Diğer taraftan bir sinir hücresinin tepki hızı, günümüz bilgisayarlarına göre oldukça yavaş olmakla birlikte duyuşsal bilgileri son derecede hızlı değerlendirebilmektedir. Bu nedenle insan beyni; öğrenme, birleştime, uyarılama ve genelleştime yeteneği nedeniyle son derece karmaşık, doğrusal olmayan ve paralel dağılmış bir bilgi işleme sistemi olarak tanımlanabilir.

2.2.2 Yapay sinir ağları

Yapay sinir ağları, insan beyninin temel birimi olan nöronlara benzer olarak teşkil edilen yapay nöronların farklı topoloji ve ağ modelleriyle birbirine bağlanmasıyla oluşan karmaşık sistemlerdir. Bir yapay sinir ağı, bir biriyle etkileşim içindeki pek çok yapay nöronun paralel bağlı bir hiyerarşik organizasyonudur [4]. Yapay sinir ağında hesaplama algoritmik programlamaya

bir seçenek oluşturan, temel olarak yeni ve farklı bir bilgi işleme tekniğidir. Programda adım adım yürütülen bir yöntemin verilmesi yerine sinirsel ağ ilişkilendirmeyi yapan iç kurallarını kendi üretir ve bu kuralları, sonuçları örneklerle karşılaştırarak düzenler [5]. En genel anlamda yapay sinir ağları ileri beslemeli ve geri beslemeli ağlar şeklinde iki ana grupta düşünülebilir. İleri beslemeli ağlarda nöronlar; girdi, saklı ve çıktı olarak adlandırılan katmanlar vasıtasıyla organize edilir. Her bir katmandaki nöronlar; bir sonraki katman nöronları ile bağlantı ağırlıkları vasıtasıyla ilişkilidir. Ancak katmanların kendi aralarında herhangi bir bağlantı yoktur. Bilgi, girdi katmanından çıktı katmanına doğru ilerler. Buna aktivasyon yönü de denilir. Bu tür yapay sinir ağına örnek olarak tek ve çok katmanlı ağ verilebilir. Bu tür ağlar denetimli öğrenme teknikleriyle eğitilir.

Geri beslemeli ağların en belirgin özelliği; katmanlar arasındaki nöronlar biri biriyle bağlantılı olup ayrıca bir dinamik hafızaya sahiplerdir. Bu ağlara örnek olarak kendi kendini düzenleyen öz örgütlenme haritası özelliğine sahip Kohonen ağı [6] ve ağ iç enerjisinin en küçüklemesine dayanan Hopfield ağı [7] verilebilir. Bu tür ağların eğitilmesi takviyesiz öğrenmeye bir örnek oluşturur. Hopfield ağları daha çok bir içerikli adreslenebilir bellek olarak veya optimizasyon tipi problemlerde başarılıdır.

Yapay sinir ağlarındaki işleme elemanları biyolojik olarak insan beynindeki nöronlara karşılık gelmektedir, Dendrit olarak adlandırılan yapı, diğer hücrelerden bilgiyi alan nöron girişleri olarak görev yapar. Diğer hücrelere bilgiyi transfer eden eleman aksonlardır. Dolayısıyla aksonlar nöron çıkışları olarak görev yaparlar. Akson ile dendrit arasındaki bağlantı ise sinapslar tarafından gerçekleştirilir.

Yapay sinir ağlarının işleyişi de buna benzer olarak gelişmektedir. 1940 yılında McCulloch ve Pitts nöronun, mantık sistemlerinde basit eş değer yapısıyla modellenebileceğini ortaya atmışlardır [8].

YSA kullanım alanları Őu Őekilde sıralanabilir:

- Veri madenciliđi
- Optik karakter tanıma
- Banka kredi müracaat deđerlendirme
- Zeki araçlar
- Rota belirleme
- Süre tahmini
- Kur, borsa tahmini

2.2.3 Yapay sinir ađlarında öğrenme

Simon[13] öğrenmeyi “zaman içinde yeni bilgilerin keşfedilmesi yoluyla davranışların iyileştirme süreci” olarak tanımlamaktadır. Yapay sinir ađları öğrenmesi ise bu öğrenme işinin bilgisayar tarafından gerçekleştirilmesinin sağlanmasıdır. Burada zaman içerisinde iyileşme kavramına dikkat çekmek gerekmektedir. Bilgisayarın da insan gibi zaman içerisinde tecrübe kazanması istenmektedir. Diđer bir deyişle makine öğrenmesi “bilgisayarın bir olay ile ilgili bilgileri ve tecrübeleri öğrenerek gelecekte oluşacak benzeri olaylar hakkında karar verebilmesi ve problemlere çözümler üretebilmesidir” denilebilir. Bilgisayarın öğrenebilmesi ve tecrübe sahibi olabilmesi bilgisayarın ilgili olay hakkında bilgiler ile donatılmasına bađlıdır. YSA yolu ile öğrenen bilgisayarların bilgiler ile donatılması, örnekler yolu ile sağlanmaktadır.

2.2.4 Örneklerden öğrenme

Bir olay hakkındaki gerçekleşmiş örnekleri kullanarak olayın girdi ve çıktıları arasındaki ilişkileri öğrenmek ve bu ilişkilere göre daha sonra oluşacak olan yeni örneklerin çıktılarını belirlemektir. Burada bir olay ile ilgili örneklerin girdi ve çıktıları arasındaki ilişkinin olayın genelini temsil edecek bilgiler içerdiği kabul edilmektedir. Deđişik örneklerin olayı deđişik açılardan temsil ettiği varsayılmaktadır. Farklı örnekler kullanarak böylece olay deđişik açılardan öğrenilmektedir. Burada bilgisayara sadece örnekler gösterilmektedir. Bunlardan başka herhangi bir ön bilgi verilmemektedir. Öğrenmeyi gerçekleştirecek sistem aradaki ilişkiyi kendi paradigması ile belirlemektedir.

Girdi vektörü X ve çıktı vektörü Y ile gösterilirse, X vektörünün n adet elemanı olacaktır. Bunlar $x_1, x_2, x_3, \dots, x_n$ şeklinde gösterilebilmektedir. Benzer şekilde çıktı vektörünün m adet boyutu olabilir. Ve $y_1, y_2, y_3, \dots, y_m$ şeklinde gösterilebilir.

2.2.5 Yapay sinir hücresi

Biyolojik sinir ağlarının sinir hücreleri olduğu gibi yapay sinir yapılarının da yapay sinir hücreleri vardır. Yapay sinir hücreleri mühendislik biliminde proses elemanı olarak da adlandırılmaktadır. Her proses elemanının 5 temel elemanı vardır:

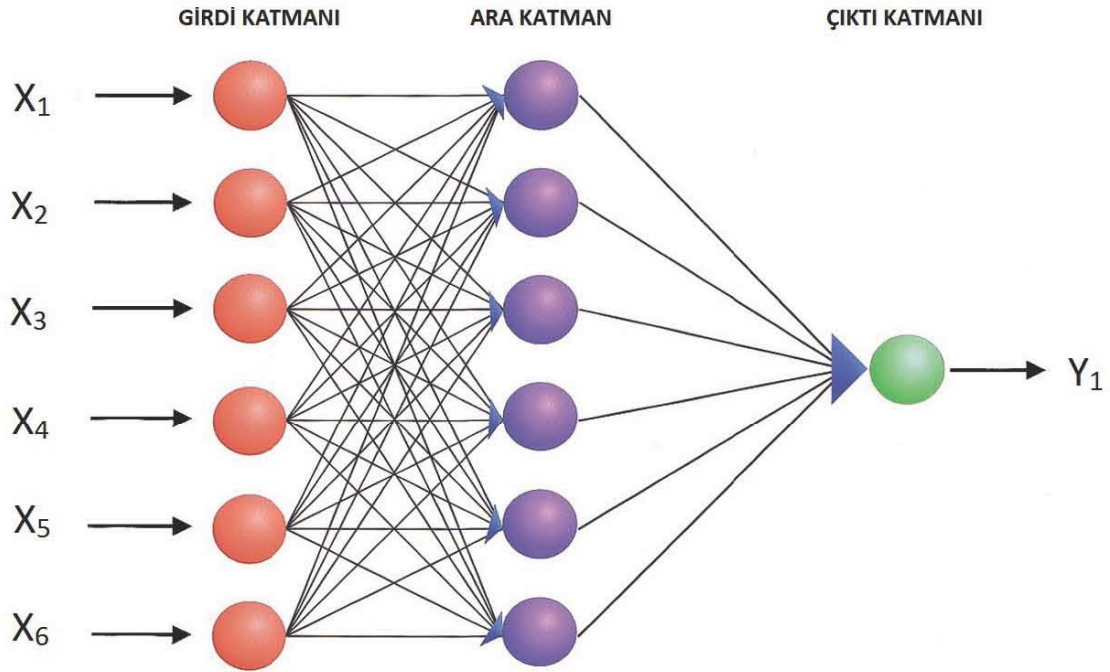
- **Girdiler:** Bir yapay sinir hücresine dış dünyadan bilgilerdir. Bunlar ağın öğrenmesi istenen örnekler tarafından belirlenir.
- **Ağırlıklar:** Bir yapay hücreye gelen bilginin önemini ve hücre üzerindeki etkisini gösterir. Ağırlıkların büyük ya da küçük olması önemli veya önemsiz olduğu anlamına gelmez.
- **Toplama Fonksiyonu:** Bu fonksiyon bir hücreye gelen net girdiyi hesaplar. Bunun için değişik fonksiyonlar kullanılmaktadır. En yaygın olanı ağırlıklı toplamı bulmaktır. Burada her gelen girdi değeri kendi ağırlığı ile çarpılarak toplanır.
- **Aktivasyon Fonksiyonu:** Bu fonksiyon hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler. Toplama fonksiyonunda olduğu gibi aktivasyon fonksiyonu olarakta çıktıyı hesaplamak içinde değişik formüller kullanılmaktadır. Günümüzde en yaygın olarak kullanılan çok katmanlı algılayıcı modelinde genel olarak aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmaktadır.
- **Hücre Çıktısı:** Aktivasyon fonksiyonu tarafından belirlenen çıktı değeridir. Üretilen çıktı dış dünyaya veya başka bir hücreye gönderilir. Hücre kendi çıktısını kendine girdi olarakta gönderebilir.

2.2.6 Çok katmanlı yapay sinir ağının yapısı

Yapay sinir hücreleri birleşerek yapay sinir ağını oluşturur. Sinir hücrelerinin bir araya gelmesi rastgele olmaz. Genel olarak hücreler 3 katman halinde ve her

katman kendi içinde paralel olarak bir araya gelerek ağı oluşturur. Şekil 2.3 de örnek bir yapay sinir ağı yapısı verilmiştir.

- **Girdi Katmanı:** Dış dünyadan gelen girdileri alarak ara katmana gönderir. Bu katmanda bilgi işleme olmaz. Gelen her bilgi geldiği gibi bir sonraki katmana gider. Her proses elemanın sadece bir tane girdisi ve bir tane çıktısı vardır. Yani, girdi katmanındaki her proses elemanı bir sonraki katmanda bulunan proses elemanlarının hepsine bağlıdır.
- **Ara Katman:** Ara katmanlar girdi katmanından gelen bilgileri işleyerek bir sonraki katmana gönderir. Çok katmanlı bir ağda birden fazla ara katman ve her katmanda birden fazla proses elemanı bulunabilir. Bir ağ içinde birden fazla ara katman olabilir
- **Çıktı Katmanı:** Ara katmandan gelen bilgileri işleyerek ağa girdi katmanından verilen girdilere karşılık ağın ürettiği çıktıları belirleyerek dış dünyaya gönderir. Bir çıktı katmanında birden fazla proses elemanı olabilir. Her proses elemanı bir önceki katmanda bulunan bütün proses elemanlarına bağlıdır. Her proses elemanının bir çıktısı vardır.



Şekil 2.3 Çok katmanlı yapay sinir ağı

2.2.7 ÇKA ileri doğru hesaplama

Bu safhada bilgi işleme eğitim setindeki bir örneği girdi katmanından (G_1, G_2, \dots) ağa gösterilmesi ile başlar. Girdi katmanında herhangi bir bilgi işleme olmaz. Gelen girdiler hiçbir değişiklik olmadan ara katmana gönderilir. Girdi katmanındaki k . proses elemanının çıktısı (C_k^i) (2.1) formülü kullanılarak hesaplanır [13].

$$C_k^i = G_k \quad (2.1)$$

Ara katmandaki her proses elemanı girdi katmanındaki bütün proses elemanlarından gelen bilgileri bağlantı ağırlıklarının (A_1, A_2, \dots) etkisi ile alır. Önce ara katmandaki proses elemanlarına gelen net girdi (NET_j^a) (2.2) formülü kullanılarak hesaplanır.

$$NET_j^a = \sum_{k=1}^n A_{kj} C_k^i \quad (2.2)$$

Burada A_{kj} k . girdi katmanının elemanının j . ara katman elemanına bağlayan bağlantının ağırlık değerini göstermektedir. j . ara katman elemanının çıktısı ise bu net girdinin aktivasyon fonksiyonundan (Genellikle sigmoid fonksiyonundan) geçirilmesi ile hesaplanır. Uygulamada genellikle bu fonksiyon kullanılmakla beraber, kullanılması zorunlu değildir. Önemli olan burada türevi alınabilir bir fonksiyon kullanmaktır. Aktivasyon fonksiyonlarından herhangi birisini burada kullanmak mümkündür. Yalnız geriye doğru hesaplamada burada kullanılan fonksiyonun türevinin alınacağını unutmamak gerekir. Sigmoid fonksiyonu kullanılması halinde çıktı (2.3) şeklinde olacaktır.

$$C_j^a = \frac{1}{1 + e^{-(NET_j^a + \beta_j^a)}} \quad (2.3)$$

Burada β_j , ara katmanda bulunan j . elemana bağlanan eşik değer elemanın ağırlığını göstermektedir. Bu eşik değeri ünitesinin çıktısı sabit olup bire eşittir. Ağırlık değeri ise sigmoid fonksiyonunun oryantasyonunu belirlemek üzere konulmuştur. Eğitim esnasında ağ bu değeri kendisi belirlemektedir.

Ara katmanın bütün proses elemanları ve çıktı katmanının proses elemanlarının çıktıları aynı şekilde kendilerine gelen NET girdinin hesaplanması ve sigmoid fonksiyonundan geçirilmesi sonucu belirlenirler. Çıktı katmanından çıkan değerler, yani çıktıları, $(\zeta_1, \zeta_2, \dots)$ bulununca ağın ileri hesaplama işlemi tamamlanmış olur.

2.2.8 ÇKA geriye doğru hesaplama

Ağa sunulan girdi için ağın ürettiği çıktı, ağın beklenen çıktıları (B_1, B_2, \dots) ile karşılaştırılır. Bunların arasındaki fark hata olarak kabul edilir. Amaç bu hatanın azaltılmasıdır. Bu nedenle geriye hesaplamada bu hata ağın ağırlık değerlerine dağıtılarak bir sonraki özyinelemede hatanın azaltılması sağlanır. Çıktı katmanındaki m . proses elemanı için oluşan hata (E_m) (2.4) formülü ile hesaplanır.

$$E_m = B_m - \zeta_m \quad (2.4)$$

Bu bir proses elemanı için oluşan hatadır. Çıktı katmanı için oluşan toplam hatayı (TH) bulmak için bütün hataların toplanması gerekir. Bazı hata değerleri negatif olacağı için toplamın sıfır olmasını önlemek amacıyla ağırlıkların kareleri hesaplanarak sonucun karekökü alınır. ÇKA ağının eğitilmesindeki amaç bu hatayı en aza indirmektir. TH (2.5) formülü ile bulunur.

$$TH = \frac{1}{2} \sum_m E_m^2 \quad (2.5)$$

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

2.2.8.1 Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi

Ara katmanındaki j . proses elemanını çıktı katmanındaki m . proses elemanına bağlayan bağlantının ağırlığındaki değişim miktarına ΔA^a denilirse; herhangi bir

t zamanında (t. özyinelemede) ağırlığın değişim miktarı (2.6) formülü ile hesaplanır [14].

$$\Delta A_{jm}^a(t) = \lambda \delta_m \zeta_j^a + \alpha \Delta A_{jm}^a(t-1) \quad (2.6)$$

Burada λ , öğrenme katsayısını; α momentum katsayısını göstermektedir. öğrenme katsayısı ağırlıkların değişim miktarını, momentum katsayısı ise ÇKA ağırlığın öğrenmesi esnasında yerel bir optimum noktaya takılıp kalmaması için ağırlık değişim değerinin belirli bir oranda bir sonraki değişime eklenmesini sağlarlar. Eşitlikteki δ_m ise m. çıktı biriminin hatasını göstermektedir. (2.7) ile hesaplanır.

$$\delta_m = f'(NET).E_m \quad (2.7)$$

Buradaki $f'(NET)$ aktivasyon fonksiyonunun türevidir. Sigmoid fonksiyonunun kullanılması durumunda (2.8) şeklinde olacaktır.

$$\delta_m = \zeta_m(1 - \zeta_m).E_m \quad (2.8)$$

Değişim miktarı hesaplandıktan sonra ağırlıkların t. özyinelemedeki yeni değerleri (2.9) formülündeki gibi olacaktır.

$$A_{jm}^a(t) = A_{jm}^a(t-1) + \Delta A_{jm}^a(t) \quad (2.9)$$

Benzer şekilde eşik değer biriminin de ağırlıklarını değiştirmek gerekmektedir. Bunun için öncelikle değişim miktarını hesaplamak gerekir. Eğer çıktı katmanında bulunan proses elemanlarının eşik değer aralıkları β^c ile gösterilirse; bu birimin çıktısının sabit ve bir olması nedeniyle değişim miktarı (2.10) formülündeki gibi olacaktır.

$$\Delta \beta_m^c(t) = \lambda \delta_m + \alpha \Delta \beta_m^c(t-1) \quad (2.10)$$

Eşik değerlerin t. özyinelemedeki ağırlığının yeni değeri ise (2.11) formülü ile hesaplanacaktır.

$$\beta_m^c(t) = \beta_m^c(t-1) + \Delta \beta_m^c(t) \quad (2.11)$$

2.2.8.2 Ara katmanlar arası veya ara katman - girdi katmanı arasındaki ağırlıkların değiştirilmesi

Dikkatli incelenirse, ara katman ile çıktı katmanı arasındaki ağırlıkların değişiminde her ağırlık için sadece çıktı katmanındaki bir proses elemanının hatası dikkate alınmıştır. Bu hataların oluşmasında girdi katmanı ve ara katman arasındaki ağırlıkların (varsa iki katman arasındaki ağırlıkların) payı vardır [15]. Çünkü, en son ara katmana gelen bütün bilgiler girdi katmanı veya önceki ara katmandan gelmektedir. Bu nedenle girdi katmanı ile ara katman arasındaki (veya iki ara katman arasındaki) ağırlıkların değiştirilmesinde çıktı katmanındaki proses elemanlarının hepsinin hatasından payını alması gerekir. Bu ağırlıklardaki değişimi (mesela girdi katmanı ile ara katman arasındaki ağırlıkların değişimi) ΔA^i ile gösterilirse değişim miktarı (2.12) formülündeki gibi olacaktır.

$$\Delta A_{kj}^i(t) = \lambda \delta_j^a C_k^i + \alpha \Delta A_{kj}^i(t-1) \quad (2.12)$$

buradaki hata terimi δ^a ise (2.13) formülü ile hesaplanacaktır.

$$\delta_j^a = f'(\text{NET}) \sum_m \delta_m A_{jm}^a \quad (2.13)$$

Aktivasyon fonksiyonu olarak sigmoid fonksiyonu düşünülürse bu hata değeri (2.14) formülü ile hesaplanacaktır.

$$\delta_j^a = C_j^a (1 - C_j^a) \sum_m \delta_m A_{jm}^a \quad (2.14)$$

hata değeri hesaplandıktan sonra yukarıda verilen eşitlikle değişim miktarını bulmak mümkün olur. Ağırlıkların yeni değerleri ise (2.15) formülü ile hesaplanacaktır.

$$A_{kj}^i(t) = A_{kj}^i(t-1) + \Delta A_{kj}^i(t) \quad (2.15)$$

Benzeri şekilde eşik değer biriminin yeni ağırlıkları da yukarıdaki gibi hesaplanır. Ara katman eşik değer ağırlıkları β^a ile gösterilirse değişim miktarı (2.16) formülündeki gibi olacaktır.

$$\Delta\beta_j^a(t) = \lambda\delta_j^a + \alpha\Delta\beta_j^a(t-1) \quad (2.16)$$

Ağırlıkların yeni değerleri ise t. özyinelemede (2.17) formülü ile hesaplanacaktır.

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta\beta_j^a(t) \quad (2.17)$$

Böylece ağırlıklarının hepsi değiştirilmiş olacaktır. Bir özyineleme hem ileri hem de geriye hesaplamaları yapılarak tamamlanmış olacaktır. İkinci bir örnek vererek sonraki özyinelemeye başlanır ve aynı işlemler öğrenme tamamlanıncaya dek yinelenir.

2.2.9 Aktivasyon fonksiyonları

Çalışmada kullanılan aktivasyon fonksiyonları şu şekilde sıralanabilir.

Çizelge 2.1 Aktivasyon fonksiyonları

| | |
|--------------------|---|
| Gaus Fonksiyon | $f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$ |
| Doğrusal Fonksiyon | $f(x) = mx + b$ |
| Rampa Fonksiyon | $R(x) := \begin{cases} x, & x \geq 0; \\ 0, & x < 0 \end{cases}$ |
| Sgn Fonksiyon | $\text{sgn}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}$ |
| Sigmoid Fonksiyon | $P(t) = \frac{1}{1 + e^{-t}}$ |

Çizlege 2.1 Aktivasyon fonksiyonları devamı

| | |
|----------------------|---|
| Adım Fonksiyon | $f(x) = \sum_{i=0}^n \alpha_i x A_i(x)$ |
| Hiperbolik Fonksiyon | $\tanh x = \frac{e^{2x} - 1}{e^{2x} + 1}$ |

3 YAPAY SİNİR AĞLARI İLE ZAMAN TAHMİNİ

3.1 Mevcut Durum ve Problemin Tanımı

Çalışmada ele alınan firma 1953 den bu yana otomotiv sektöründe her türlü ihtiyacı karşılayacak şekilde sektöre parça üreten bir firmadır. Firma bünyesinde 24 saat üretim yapılmaktadır. Firma sektörde yapılan her türlü değişikliğe hızlı bir şekilde cevap vermek zorundadır ve bundan dolayı yeni üretilecek her ürün için yapılacak üretim makine zaman tahmini firma için çok önemlidir. Yapılan analizlerde ortaya çıkan tahminler eğer fazla olursa, siparişi zamanında yetiştirememesi sorunu ile karşı karşıya kalacaktır. Analizler sonucunda zamanı az tahmin ederler ise firma yapacağı iş planlaması sonucunda boşta kalan ve çalışmayan makineler ortaya çıkacaktır. Bu nedenlerden dolayı yapılacak zaman tahmini firmayı maddi anlamda zarara uğratabilecek en öncelikli parametre haline gelmektedir.

Çalışmanın yapıldığı firma, sektöründe öncü bir döküm ve hammadde işleme firmasıdır. Piyasaya binlerce farklı amaçlarla kullanılan gri ve sfero pik parçalarda ham, işlenmiş ve birim üretimi yapmaktadır. Ana ürünleri motor blokları, silindir kafaları, güç aktarım organları grubu, yağ karterleri, volan, volan muhafazaları, aks kovanları, adaptörler sıralanabilir. Piyasada öncü olan firma, yerli ve yabancı müşterisine hizmet vermektedir.

Firmada süre ölçümleri zaman etüdü ile yapılmaktadır. Dökümden gelen hammaddeler, Mazak CNC tezgahlarında işleme, temizleme ve taşıma zamanları ölçülerek standart zamanlar kayıt altına alınmıştır. Fakat bu zamanlar firmada bir veri tabanında tutulmamaktadır ve aynı ürünü üreten CNC tezgahları arasında farklı zamanlar gözlenebilmektedir. Firmanın yaptığı tahminler ile gerçekleşen tahminler arasında ciddi farklılıklar vardı ve firma bu durumdan şikayetçi idi. Bu durumun iyileştirilmesi firma için önemli bir adımdı.

Çalışmada yapay sinir ağlarının kullanılmasının temelinde hızlı ve esnek bir araç olarak kullanılabilmesi yatmaktadır.

CNC tezgahında üretilecek her ürün için Bölüm 3.2.1 de sıralanan parametreler standart zamanı etkileyen faktörler olarak belirlenmiştir.

Bu girdilere karşılık hesaplanacak parametre standart zaman olarak belirlenmiştir. Fabrikada yapılan çalışmalarda yapay sinir ağlarında kullanılacak 114 örnek veri toplanmıştır ve bu veriler ağ eğitimi için kullanılmıştır.

Fabrikadan veri toplanma süreci zahmetli ve zor olmuştur. Verilerin sayısal ortama aktarılmaması ve zaman etüt çizelgelerinin iyi korunmamasından dolayı ve fabrikanın bu verileri paylaşma konusundaki isteksizliğinden dolayı, veri toplama süreci yaklaşık bir yıl sürmüştür. Verilerin sayısal ortama aktarılmaması ise zaman tahmininin, personel tarafından yapılmasının ne kadar zor ve hataya açık olduğunu göstermektedir.

3.2 Eğitim Seti

3.2.1 Girdiler ve çıktıların belirlenmesi

Çalışmada standart zamanı en çok etkileyen veriler firmadan alınan bilgilere göre belirlenmiştir. Girdiler aşağıdaki parametrelerden oluşmaktadır:

Tezde kullanılan veriler şu şekilde gösterilebilir

Çıktı:

y_1 : Standart Zaman: CNC makinenin bir ürünü üretmesi için geçen zaman.

Girdiler:

x_1 : Kesme Uzunluğu: Hammaddeden ürünü üretmek için kesilecek toplam uzunluk.

x_2 : Kesme Hızı: Makine kesme hızı.

x_3 : Delme Uzunluğu: Hammaddeden ürünü üretmek için delinecek toplam uzunluk.

x_4 : Delme Hızı: Makine delme hızı.

x_5 : Kılavuz Uzunluğu: Hammaddeden ürünü üretmek için kılavuz çekilecek uzunluk.

x_6 : Kılavuz Hızı: Makine kılavuz hızı.

x_7 : Çap Uzunluğu: Hammaddeden ürünü üretmek için çap işleme uzunluğu.

x_8 : Çap Hızı: Makine çap işleme hızı.

x_9 : Makine Kodu: Kullanılan makinenin kodu

Ürünler arasında geometrik farklılıklar çok fazla olduğundan dolayı parametreler arasında belirli bir önem sırası yoktur.

Çalışmada çıktı olarak ise sadece standart zaman örneklendirilmiştir.

Çalışmada kullanılan örnek veriler EK 1’de verilmiştir.

3.2.2 Verilerin toplanması ve verinin özellikleri

Çıktı verileri daha önce belirtildiği gibi zaman etüdü kullanılarak ölçülmüştür. Girdi verileri ise kullanılan tezgahların programlarından alınmıştır. Girdilerin programlardan alınmasına rağmen standart makine zamanları zaman etüdü ile ölçülmüştür. Çıktı verilerinde insan hatalarından kaynaklanan hatalar oluşmuş olabilir. Bu hatalar tolerans olarak fabrika tarafından belirlenmiş fakat paylaşılmamıştır. Eğer bu veriler olsaydı, girdi parametresi olarak ağa verilebilirdi.

Standart zamana etki eden temizleme, yerleştirme ve taşıma zaman değerlerinde ağa girdi olarak verilebilirdi fakat bu değerler çıktıya etki etmeyecekti. Çünkü bu değerler makina zamanı ile toplanarak tüm işlemin toplam standart zamanını oluşturuyordu. Direk girdi olarak eklenebilecek bu değerlerin ağ için anlamlı bir değer olmadığı belirlenmiştir. Çalışmada makina standart zamanı çıktı olarak kabul edilmiştir.

Toplam 114 örnek, 9 girdi ve 1 çıktı toplanarak , çalışmada kullanılacak örnek uzay oluşturulmuştur.

3.2.3 Girdiler ve çıktıların ölçeklendirilmesi

Problem çözümünde değişik transfer fonksiyonları kullanıldığından tüm girdi ve çıktılar normalize edilmiştir. Sigmoid fonksiyonu sadece 0 – 1 arasında sonuç

vereceği için, ağ çıktıları 0 – 1 arasında hesaplanacaktır. Bu yüzden girdi ve çıktılar normalize edilmiştir.

Girdi ve çıktı değerleri arasında çok büyük farklar olduğu için normalizasyon işlemi için (3.1) formülü ile ölçeklendirme yapılmıştır.

$$x_{nn} = \frac{X_{\max} - X_{\min}}{X_{\max}} x_n \quad (3.1)$$

x_{nn} hesaplanan n. normalize değeri ifade etmektedir. x_n ise n. girdiyi ifade etmektedir. Formül (3.1) e göre tüm girdi ve çıktılar normalize edilmiş ve kurulan ağ yapısında kullanılmıştır.

Normalize edilen değerler Ek 2'de verilmiştir.

3.3 Geri Yayılmalı Yapay Sinir Ağları Modelleri

Geri yayılma algoritması, basitliği ve uygulamadaki görüş açısı gibi başarılarından dolayı ağ eğitimi için en popüler algoritmalarından biridir [9]. Bu algoritma; hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Geri yayılmalı öğrenme kuralı ağ çıkışındaki mevcut hata düzeyine göre her bir tabakadaki ağırlıkları yeniden hesaplamak için kullanılmaktadır. Bir geri yayılımlı ağ modelinde giriş, gizli ve çıkış olmak üzere 3 katman bulunmakla birlikte, problemin özelliklerine göre gizli katman sayısını artırabilmek mümkündür.

Geri yayılım çok katmanlı ağlarda kullanılan delta kuralı için geliştirilmiştir bir algoritmadır. Bu algoritma çok katlı ağlarda hesap işlerini öğrenmede kullanılabilir. Geri yayılım ağında hatalar, ileri besleme aktarım işlevinin türevi tarafından, ileri besleme mekanizması içinde kullanılan aynı bağlantılar aracılığıyla, geriye doğru yayılmaktadır. Öğrenme işlemi, bu ağda basit çift yönlü hafıza birleştirmeye dayanmaktadır [10].

Geri beslemeli YSA' da, en az bir hücrenin çıkışı kendisine ya da diğer hücrelere giriş olarak verilir ve genellikle geri besleme bir geciktirme elemanı üzerinden yapılır. Geri besleme, bir katmandaki hücreler arasında olduğu gibi

katmanlar arasındaki hücreler arasında da olabilir. Bu yapısı ile geri beslemeli YSA , doğrusal olmayan dinamik bir davranış gösterir. Karmaşık verilerin sınıflandırılmasında kullanılan etkin YSA modellerinden birisidir. İlk olarak Werbos tarafından düzenlenen daha sonra Parker, Rummelhart ve McClelland tarafından geliştirilen geri yayılım ağıdır. İlk uygulamaları yazılı metinden söz sentezi, robot kollarının kontrolüdür. Sınırları ise denetimli eğitim giriş ve çıkış örneklerinin çok sayıda olmasıdır. Geriye yayılma günümüzde en yaygın kullanılan öğrenimi kolay sonuçları etkin bir YSA'dır [11].

Yayınma ve uyum gösterme yayılma ve adaptasyon olmak üzere iki aşamada işlemleri gerçekleştiren GYA, katmanlar arasında tam bir bağlantının bulunduğu çok katmanlı, ileri beslemeli ve denetimli olarak eğitilen bir YSA modelidir. Geri yayılım algoritması anlaşılması kolay ve tercih edilen öğretme algoritmasıdır. Bu algoritma; hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Geri yayımlı öğrenme kuralı ağ çıkışındaki mevcut hata düzeyine göre her bir tabakadaki ağırlıkları yeniden hesaplamak için kullanılmaktadır.

Bir giriş verisinin ağı ilk katmanında yer alan düğümlere uygulandığında en üst katman olan çıkış katmanına erişinceye kadar, bu veri üzerinde çeşitli işlemler gerçekleştirilir. Bu işlemlerin sonucunda elde edilen fiili çıktı, olması gereken çıktı ile karşılaştırılır. Fiili ve olması gereken değerler arasındaki fark, her çıktı düğümü için bir hata sinyali olarak hesaplanır. Hesaplanan hata sinyalleri, her çıktı düğümüne karşı gelen ara katmandaki düğümlere aktarılır. Böylece ara katmandaki düğümlerin her biri toplam hatanın sadece hesaplanan bir kısmını içerir. Bu süreç her katmandaki düğümler toplam hatanın belirli bir kısmını içerecek şekilde giriş katmanına kadar tekrarlanır. Elde edilen hata sinyalleri temel alınarak, bağlantı ağırlıkları her düğümde yeniden düzenlenir. Bu düzenleme tüm verilerin kodlanabileceği bir duruma ağı yakınsamasını sağlar.

İleri besleme safhasında, giriş tabakasındaki nöronlar veri değerlerini doğrudan gizli katmana iletirler. Gizli katmandaki her bir nöron kendi giriş değerlerini ağırlandırarak toplam değer hesap ederler ve bunları bir taşıma fonksiyonu ile

işleyerek bir ileriki tabakaya veya doğrudan çıkış katmanına iletirler. Katmanlar arasındaki ağırlıklar başlangıçta rastgele küçük rakamlardan seçilir.

Çıkış katmanındaki, her bir nöron ağırlıklandırılmış değeri hesaplandıktan sonra, bu değer yine taşıma fonksiyonu ile karşılaştırılarak mevcut hata minimize edilmeye çalışılır. Hata değeri belli bir seviyeye ininceye kadar öz-yineleme işlemine devam edilir ve böylece ağıın eğitim aşaması tamamlanmış olur. Katmanlar arasındaki bağlantılardaki ağırlık değerleri eğitimi tamamlamış ağdan alınarak deneme safhasında kullanılmak üzere saklanır.

Geri yayımlı çok katmanlı algılayıcılar öğretmenli öğrenme stratejisine göre çalışırlar. Yani; bu ağlara eğitim sırasında hem girdiler hem de o girdilere karşılık üretilmesi gereken çıktılar gösterilir. Ön görevi her girdi için o girdiye karşılık gelen çıktıyı üretmektir. ÇKA ağının öğrenme kuralı en küçük kareler yöntemine dayalı Delta Öğrenme Kuralının genelleştirilmiş halidir. Bu nedenle öğrenme kuralına Genelleştirilmiş Delta Kuralı da denmektedir. Ağın öğrenebilmesi için eğitim seti adı verilen ve örneklerden oluşan bir sete ihtiyaç vardır. Bu set içinde her örnek için ağın hem girdiler hem de o girdiler için ağın üretmesi gereken çıktılar belirlenmiştir. Genelleştirilmiş Delta Kuralı iki safhadan oluşur.

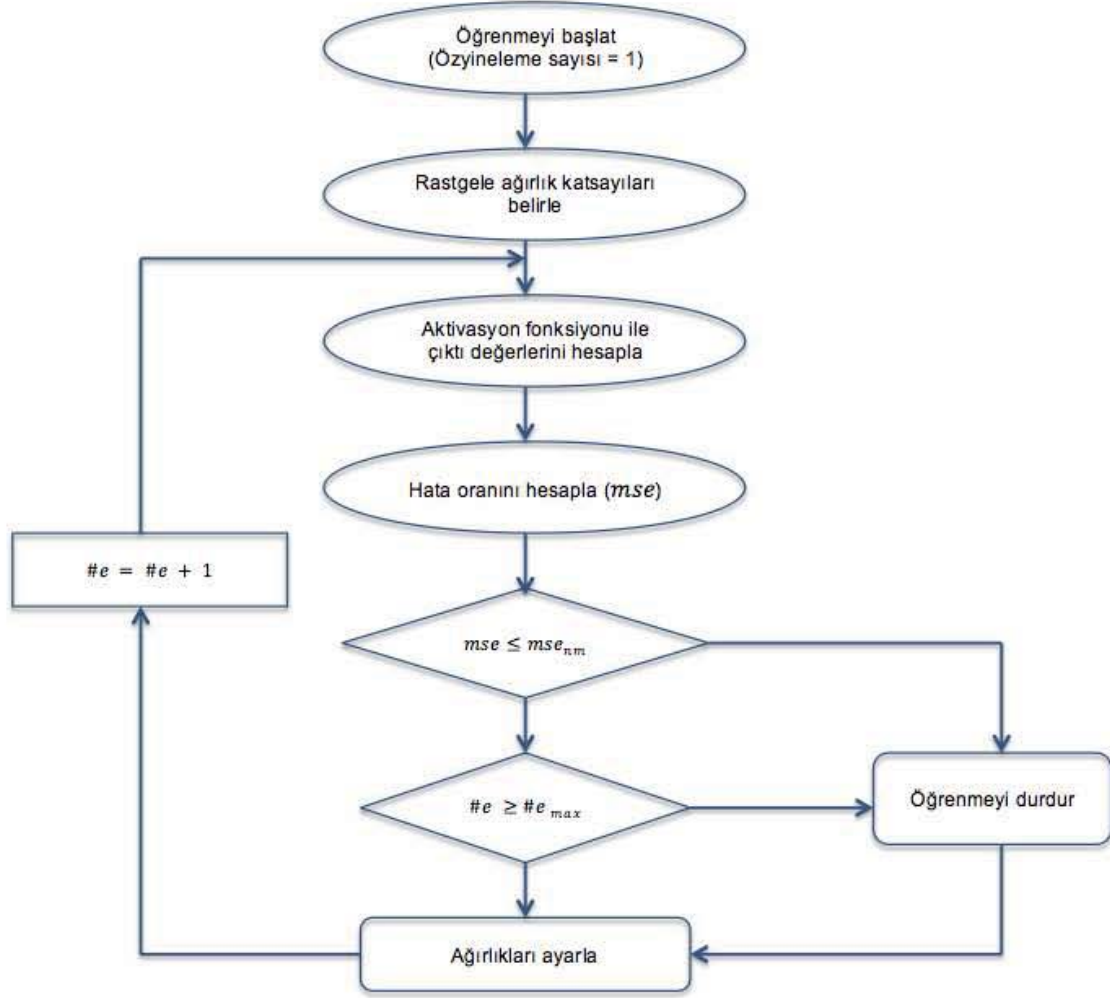
3.3.1 Çok katmanlı geri yayılım ağının çalışma prosedürü

Çalışmamızda geliştirilen ve izlenen prosedür aşağıdadır. **Error! Reference source not found.** bu prosedürün akış şemasıdır [12].

- Örneklerin toplanması: Ağın çözmesi istenen olay için daha önce gerçekleşmiş örneklerin bulunması adımıdır. Ağın eğitilmesi için örnekler toplandığı gibi (eğitim seti) ağın test edilmesi için de örneklerin (test seti) toplanması gerekmektedir. Ağın eğitilmesi sırasında test seti ağa hiç gösterilmez. Eğitim setindeki örnekler tek tek gösterilerek ağın olayı öğrenmesi sağlanır. Ağ olayı öğrendikten sonra test setindeki örnekler gösterilerek ağın performansı ölçülür. Hiç görmediği örnekler karşısındaki başarısı ağın iyi öğrenip öğrenmediğini ortaya koymaktadır.

- Ağın topolojik yapısının belirlenmesi: Öğrenilmesi istenen olay için oluşturulacak olan ağın topolojik yapısı belirlenir. Kaç tane girdi birimi, kaç tane ara katman, her ara katmanda kaç tane proses elemanı ve kaç tane çıktı elemanı olması gerektiği bu adımda belirlenmektedir.
- Öğrenme parametrelerinin belirlenmesi: Ağın öğrenme katsayısı, proses elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.
- Ağırlıkların başlangıç değerlerinin atanması: Proses elemanlarını birbirlerine bağlayan ağırlık değerlerinin ve eşik değer ünitesinin ağırlıklarının başlangıç değerlerinin atanması yapılır. Başlangıçta genellikler rastgele değerler atanır. Daha sonra ağ uygun değerleri öğrenme sırasında kendisi belirler.
- Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi: Ağın öğrenmeye başlaması ve öğrenme kuralına uygun olarak ağırlıkları değiştirmesi için ağa örnekler (girdi/çıkı değerleri) belirli bir düzeneğe göre gösterilir.
- Öğrenme sırasında ileri hesaplamaların yapılması: Sunulan girdiler için ağın çıktı değerleri hesaplanır.
- Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması: Ağın ürettiği hata değerleri(mse) bu adımda hesaplanır.
- Öğrenme tamamlanma kriterlerinin kontrol edilmesi: Bu adımda belirlenen hata değerine veya en fazla özyineleme sayısına ulaşıp ulaşılmadığı kontrol edilir. Eğer istenilen hata değerine ulaşılmış ise öğrenme tamamlanmış demektir. Eğer en fazla özyineleme sayısına ulaşıldıysa istenilen aralıkta öğrenmenin gerçekleştirilmediği anlamına gelmektedir. Eğer iki koşulda gerçekleşmemiş ise öğrenme aşağıdaki adımla devam eder.
- Ağırlıkların değiştirilmesi: Geri hesaplama yöntemi uygulanarak üretilen hatanın azalması için ağırlıkların değiştirilmesi yapılır.

Çalışma prosedürü Şekil 3.1'de verilmiştir. Bu prosedür uygulanarak algoritma programda kullanılmıştır [13].



Şekil 3.1 Geri yayılım algoritması çalışma prosedürü

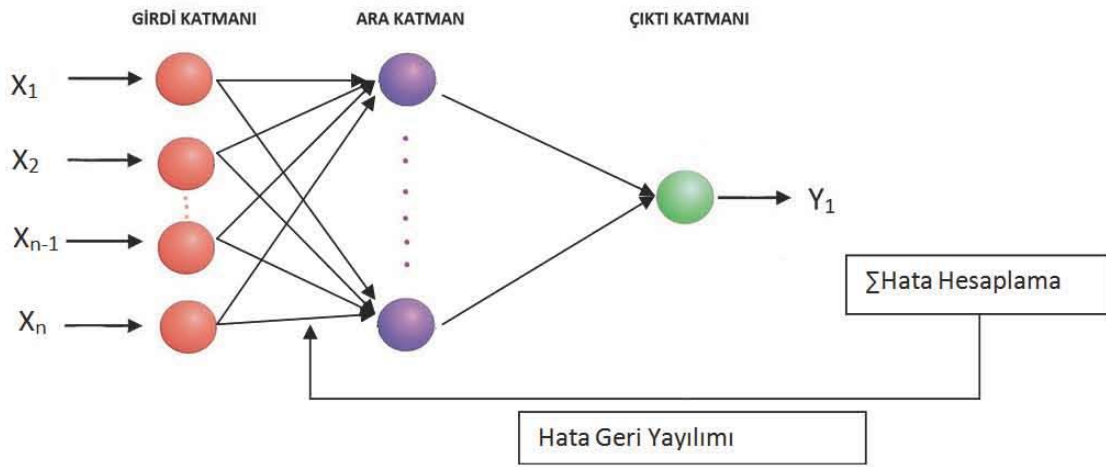
3.4 Yapay Sinir Ağları ile Zaman Tahmini Yapan Yazılımın Oluşturulması

3.4.1 Tasarlanan ağ yapısı

Yapay sinir ağlarında kullanılan ara katman ve transfer fonksiyonları ağın eğitilmesinde önemli birer kriterdir. Ağın öğrenmesi için geçen süre, kullanılacak ara katman sayısı ve transfer fonksiyonu ile değişmektedir. Örneğin ara katman sayısı 1 iken bir ağın öğrenmesi 2 milyon özyinelemede gerçekleşirken, 5 ara katman kullanıldığı zaman özyineleme sayısı bine inebilir. Fakat sistematik bir kural yoktur. Genellikle kullanılacak ara katman sayısı ve transfer fonksiyonu tecrübelerle dayanarak seçilirler.

Yapılan çalışmada ara katman sayısı ve aktivasyon fonksiyonlarını kullanıcının kendi seçmesine olanak tanınmıştır. Yazılım, problemi çözmesi yanında, kullanıcıya öğrenmenin en kısa hangi parametrelerle gerçekleştiğini de sunar. Hangi parametreler ile en kısa sürede ağ öğrenmesinin ve en az öz yineleme sayısının gerçekleştiği sorularına cevap verilir.

Tasarlanan yazılım ile geri yayılım algoritması dinamik bir şekilde değiştirilebilir. Bu ağ yapısı Şekil 3.2'de gösterilmiştir.



Şekil 3.2 Tasarlanan ağ yapısı

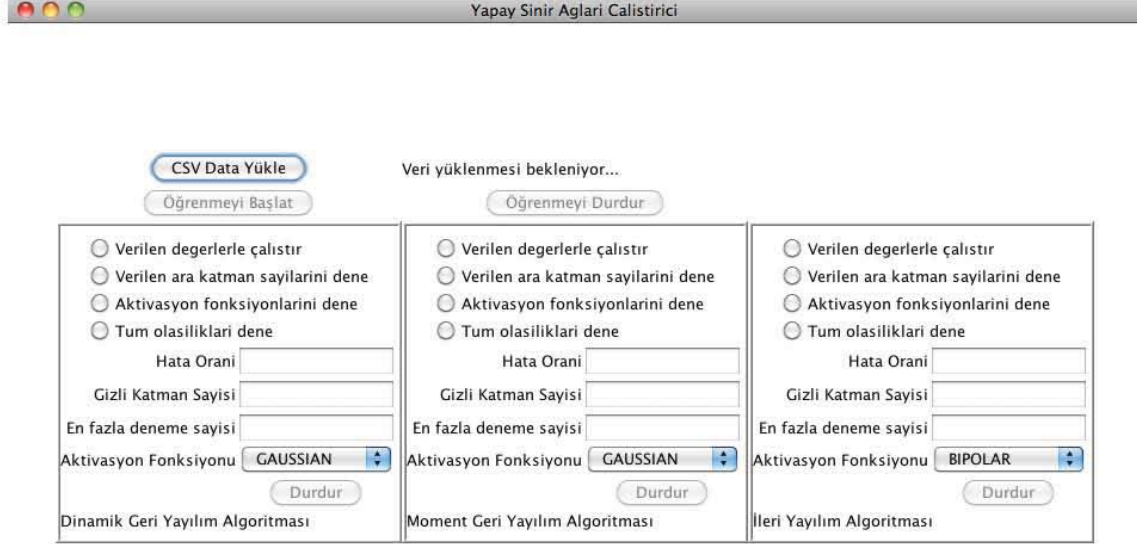
Yazılım Şekil 3.4'teki seçeneklerden birisi ile çalıştırabilir. Yazılımda ağın öğrenmeyi durdurma kriterleri olarak, hata oranı ve en fazla deneme sayısı kullanıcı tarafından yazılıma girilir.

Yazılım üç farklı algoritmayı çözecek şekilde tasarlanmıştır. Bu algoritmalar, ileri yayımlı öğrenme, dinamik geri yayımlı öğrenme ve moment geri yayımlı öğrenme algoritmalarıdır. Bu algoritmalar geri yayılım algoritmasından türemişlerdir [15] [16] [17]. Bölüm 0 de anlatıldığı üzere algoritmalar JAVA ile kodlanmıştır.

Bölüm 2.2.7 ve 0 de verilen algoritmalar, EK 6, EK 7 ve EK 8 da kodlarıyla beraber verilmiştir.

3.4.2 Geliştirilen yazılım

Yapay sinir ağları ile belirlenen problemin çözümü için de gösterilen kullanıcı ara yüzü geliştirilmiştir.



Şekil 3.3 Geliştirilen kullanıcı ara yüzü

Yazılım kullanıcının kolayca kullanması için sade ve basit bir ara yüze sahip olacak şekilde tasarlanmıştır. kullanıcı ara yüzü ile ağı kendi tasarlayabilmektedir. Bunun yanında ağ kendi başına öğrenmeyi de sağlayabilmektedir. Yazılım ağı tanımak için aşağıda sıralanan değerleri bilmelidir.

- Hata oranı: Öğrenmenin duracağı toplam hata değeri
- Gizli katman sayısı: Ağın öğrenme sırasında kullanacağı hata sayısı
- En fazla deneme sayısı: Öğrenmenin duracağı en fazla özyineleme sayısı
- Aktivasyon fonksiyonu: Ara katmandan çıktı katmanına geçmek için kullanılan aktivasyon fonksiyonu

3.4.2.1 Yazılım çalışma stratejileri

Çalışma kapsamında geliştirilen yazılım, dört farklı seçenekte çalıştırılabilir. Bu seçenekler farklı ihtiyaçlara ve farklı sorulara yanıt verecek şekilde tasarlanmıştır. Bu çalışma seçenekleri Şekil 3.4'de gösterilmiştir. Aşağıda bu çalışma stratejileri açıklanmıştır.

- Verilen degerlerle çalıştır
- Verilen ara katman sayilarini dene
- Aktivasyon fonksiyonlarini dene
- Tum olasiliklari dene

Şekil 3.4 Yazılım çalışma stratejileri

Yazılımın çalışma seçenekleri şöyledir:

- Verilen değerlerle çalışma seçeneği
Bu seçeneğin amacı kullanıcının belirleyeceği değerlerle öğrenmeyi sağlamaktır. Kullanıcı ağı istediği sayıda ara katman ve belirleyeceği aktivasyon fonksiyonu ile eğitmeye programlar.

Örneğin ara katman sayısı 6, aktivasyon fonksiyonu sigmoid, hata oranı 0.01 ve en fazla deneme sayısı 30000 ise yazılım Şekil 3.5'de gösterildiği gibi programlanmalıdır.

| | |
|------------------------|--------------------------------------|
| Hata Oranı | <input type="text" value="0.01"/> |
| Gizli Katman Sayısı | <input type="text" value="6"/> |
| En fazla deneme sayısı | <input type="text" value="30000"/> |
| Aktivasyon Fonksiyonu | <input type="text" value="SIGMOID"/> |

Şekil 3.5 Verilen değerlerle ağı programlama

- Verilen ara katman sayılarını dene seçeneği:

Bu seçenekte amaç, belirli bir aktivasyon fonksiyonunun ara katman sayılarını değiştirerek, ara katman sayısının, ağ öğrenimine etkisini göstermektir.

Örneğin 6 - 16 arasında en hızlı ara katman hangisi sorusuna yanıt arayan bir kullanıcı belirlediği aktivasyon fonksiyonu ve durdurma kriterleri ile bu soruya cevap bulabilir. Hata oranı 0.01, en fazla deneme sayısı 30000 ve aktivasyon fonksiyonu sigmoid ise yazılım Şekil 3.6'de gösterildiği gibi programlanmalıdır.

| | |
|------------------------|--|
| Hata Oranı | <input type="text" value="0.01"/> |
| Gizli Katman Sayısı | <input type="text" value="6"/> - <input type="text" value="16"/> |
| En fazla deneme sayısı | <input type="text" value="30000"/> |
| Aktivasyon Fonksiyonu | <input type="text" value="SIGMOID"/> |

Şekil 3.6 Ara katman sayısına göre programlama

- Aktivasyon fonksiyonlarını dene seçeneği:

Bu seçenekte amaç, belirli bir ara katman sayısı ile, aktivasyon fonksiyonlarını değiştirerek, aktivasyon fonksiyonlarının ağ öğrenimine etkisini göstermektir.

Örneğin 6 ara katman ile hangi aktivasyon fonksiyonunun en hızlı olacağı sorusuna yanıt bulunabilir. Hata oranı 0.01, en fazla deneme sayısı 30000 ve ara katman sayısı 6 iken yazılım Şekil 3.7'da gösterildiği gibi programlanmalıdır.

| | |
|------------------------|------------------------------------|
| Hata Oranı | <input type="text" value="0.01"/> |
| Gizli Katman Sayısı | <input type="text" value="6"/> |
| En fazla deneme sayısı | <input type="text" value="30000"/> |

Şekil 3.7 Aktivasyon fonksiyonlarına göre programlama

- Tüm olasılıkları dene seçeneği:

Bu seçenekte amaç, kullanıcının seçtiği aralıkta ara katman sayısında tüm aktivasyon fonksiyonlarını denemektir. Böylelikle ağ en kısa hangi ara katman sayısında ve hangi aktivasyon fonksiyonu ile eğitilir sorusuna yanıt bulunabilecektir.

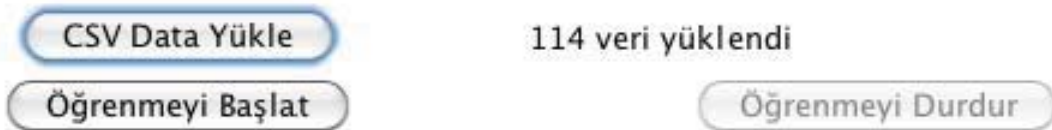
6 – 16 ara katman sayısı arasında, hata oranı 0.01 ve en fazla deneme sayısı 30000 iken en hızlı öğrenme ve en az özyineleme hangi parametrelerle gerçekleşir sorularına cevap vermek için yazılım Şekil 3.8'de gösterildiği gibi programlanmalıdır.

| | |
|------------------------|--|
| Hata Oranı | <input type="text" value="0.01"/> |
| Gizli Katman Sayısı | <input type="text" value="6"/> - <input type="text" value="16"/> |
| En fazla deneme sayısı | <input type="text" value="30000"/> |

Şekil 3.8 Tüm olasılıkları deneyen programlama

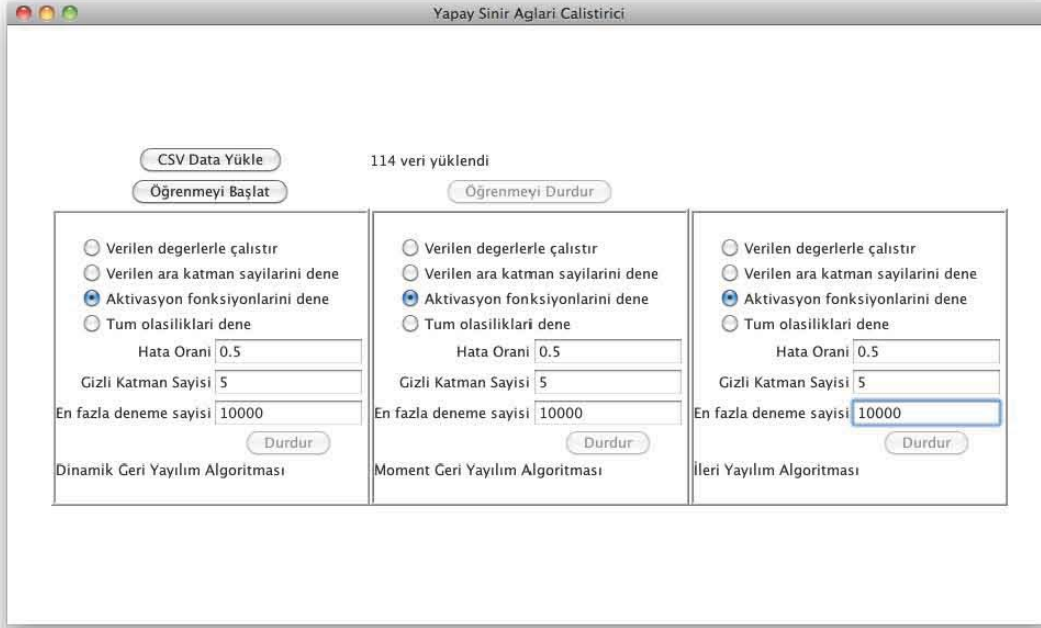
3.4.2.2 Yazılımın çalıştırılması

Ağ öğrenmesinin başlanması için öncelikle CSV formatında bir dosyadan veri setinin yazılıma verilmesi gerekiyor. Şekil 3.3'de gösterilen, CSV Data Yükle tuşu ile, dosya sistemindeki herhangi bir CSV formatındaki dosyadan yazılıma veri yüklenebilir. Öğrenmeye Başlat tuşu veri yüklenmeden aktif hale gelmez. Ek 1'deki veriler CSV dosyası olarak kaydedildikten sonra, yazılımın görüntüsü Şekil 3.9'da verilmiştir.



Şekil 3.9 Veri yüklendikten sonraki görünüm

Error! Reference source not found.'de yazılıma veri yüklendikten ve başlangıç parametreleri girilmiş ekran çıktısı verilmiştir.

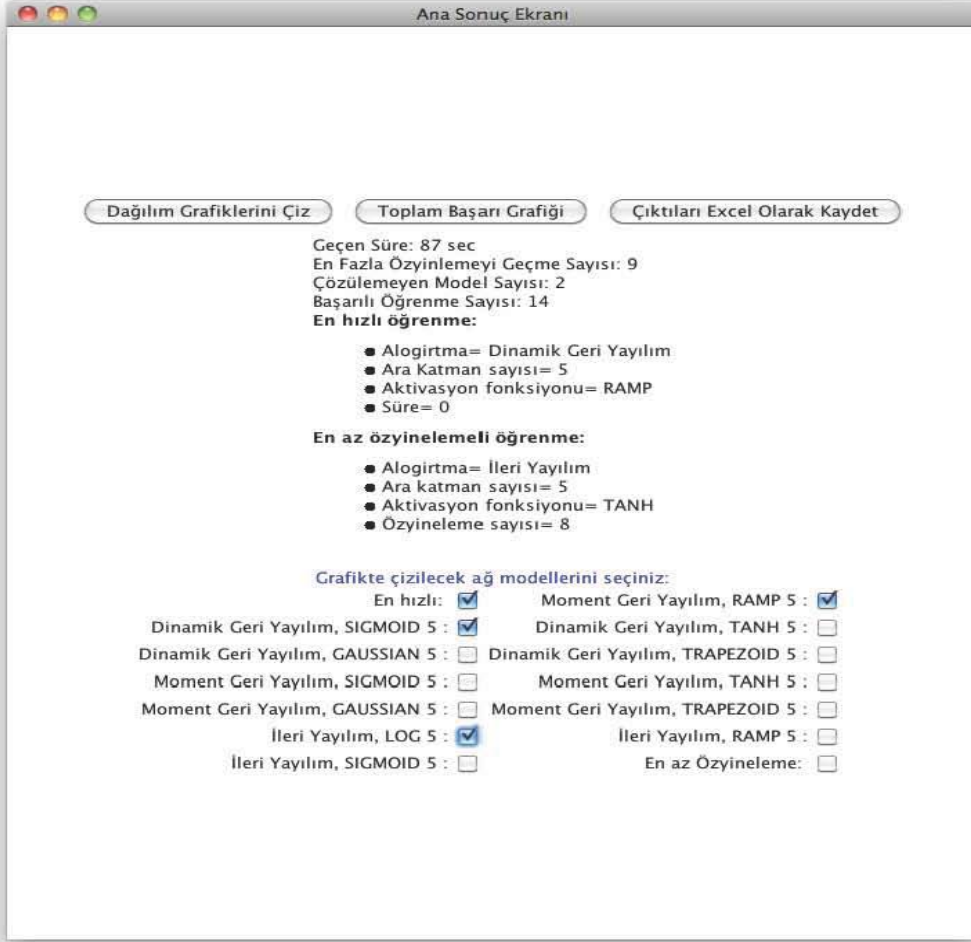


Şekil 3.10 Veri yüklendikten sonra yazılımın görüntüsü

Veriler yüklendikten sonra, 0 da anlatılan herhangi bir strateji seçilir. Seçilen stratejiler için zorunlu alanlar girildikten sonra, Öğrenmeyi Başlat tuşuna basarak öğrenme başlatılır.

3.4.2.3 Yazılımın sonuçlarının gösterilmesi

Çalışmada hazırlanan yazılımda çıktıları kullanıcı grafiksel olarak ve excel dosya formatında kaydedebilecek şekilde ayarlanmıştır. Ana sonuç ekranından Şekil 3.11'de gösterildiği gibi ana sonuçlar görülebilmektedir. Bu ekranın amacı, detaylı sonuçların gösterileceği panellere geçişleri sağlamaktır.



Şekil 3.11 Ana Sonuç Ekranı

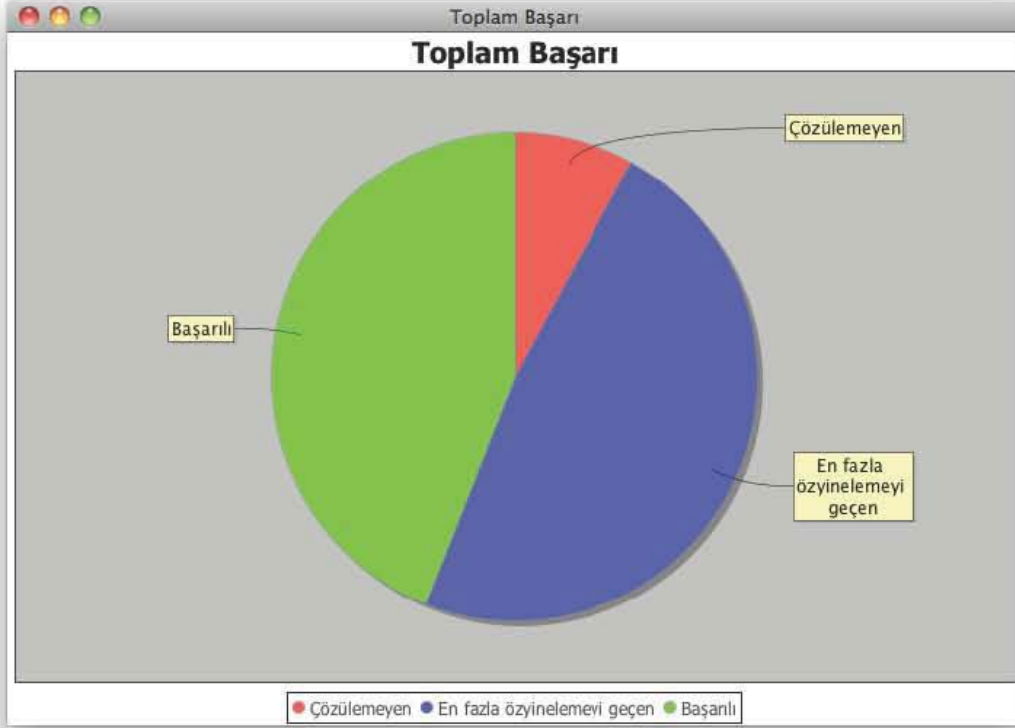
Şekil 3.11’de gösterildiği üzere Ana Sonuç Ekranında şu bilgiler vardır:

- Geçen Süre: Tüm işlemlerin ne kadar sürdüğünü gösterir.
- En Fazla Özyinelemeyi Geçme Sayısı: Dinamik modellenen ağlardan kaç tanesi en fazla özyineleme sayısına ulaşmış, öğrenmeyi tamamlayamamış bilgisini verir.
- Çözülmeyen Model Sayısı: Dinamik modellenen ağlardan kaç tanesi öğrenmeyi tamamlayamamıştır bilgisini verir.
- Başarılı Öğrenme Sayısı: Dinamik modellenen ağlardan kaç tanesi öğrenmeyi tamamlamıştır bilgisini verir.
- En Hızlı Öğrenme: Dinamik modellenen ağlardan hangisi en kısa sürede öğrenmeyi tamamlamıştır bilgisini verir. En kısa sürede öğrenmeyi tamamlayan

ağın, hangi algoritma, kaç ara katman sayısı, hangi aktivasyon fonksiyonu ile gerçekleştiğini ve bu gerçekleşmenin saniye cinsinden süresini verir.

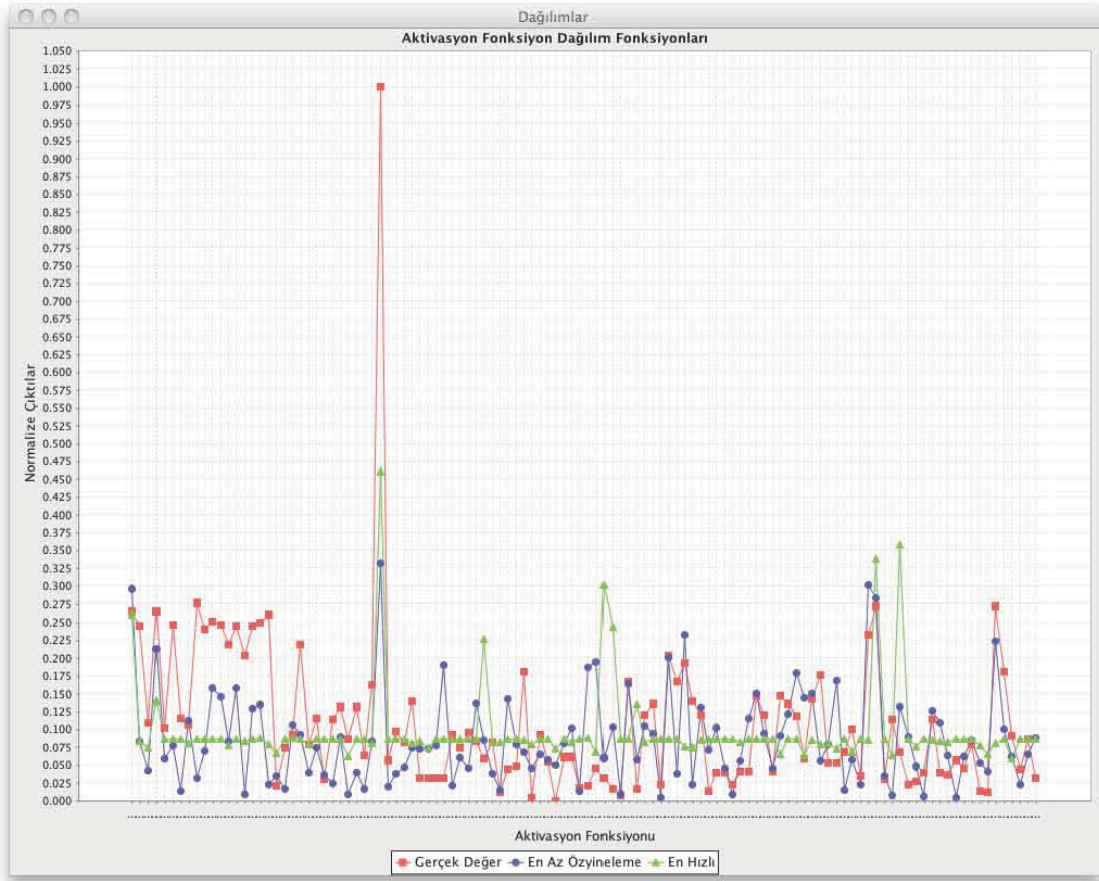
- En Az Özyinelemeli Öğrenme: Dinamik modellenen ağlardan hangisi en az sayıda özyineleme sonrasında öğrenmeyi tamamlamıştır bilgisini verir. En az özyinelemeli öğrenmeyi tamamlayan ağın, hangi algoritma, kaç ara katman sayısı, hangi aktivasyon fonksiyonu ile gerçekleştiğini ve bu gerçekleşmenin kaç özyinelemede olduğu bilgisini verir.
- Grafikte çizilecek ağ modelleri: Bu panelde, ağ öğrenmesini tamamlayan tüm modeller sıralanır. İsimlendirme düzeni algoritma ismi, aktivasyon fonksiyonu ve ara katman sayısının yan yana yazılmasıyla ortaya çıkarılmıştır. Örneğin Moment Geri Yayılım, SIGMOID, 5. Bu isimlendirme haricinde, diğer modellere göre, kullanıcının görüntülemek isteyeceği en hızlı yayılım ve en az özyinelemeli modeller ayrı kendi isimleriyle isimlendirilmiştir. Buradaki seçme kutuları, hangi sonuçların grafikte yer alacağını gösteren kutucuklardır. Şekil 3.13'de gösterilen örnek grafik bu kutuların seçilmesiyle yaratılmıştır.

Şekil 3.12'de gösterilen Toplam Başarı Grafiği tuşuna basılarak, modellerin kaçının öğrenmeyi tamamladığını gösterir. Örnek Şekil 3.12'de görülen yeşil alan öğrenmeyi başaran modelleri, mavi alan en fazla özyinelemeyi geçen modelleri ve kırmızı alan ise çözülemeyen modelleri gösterir.



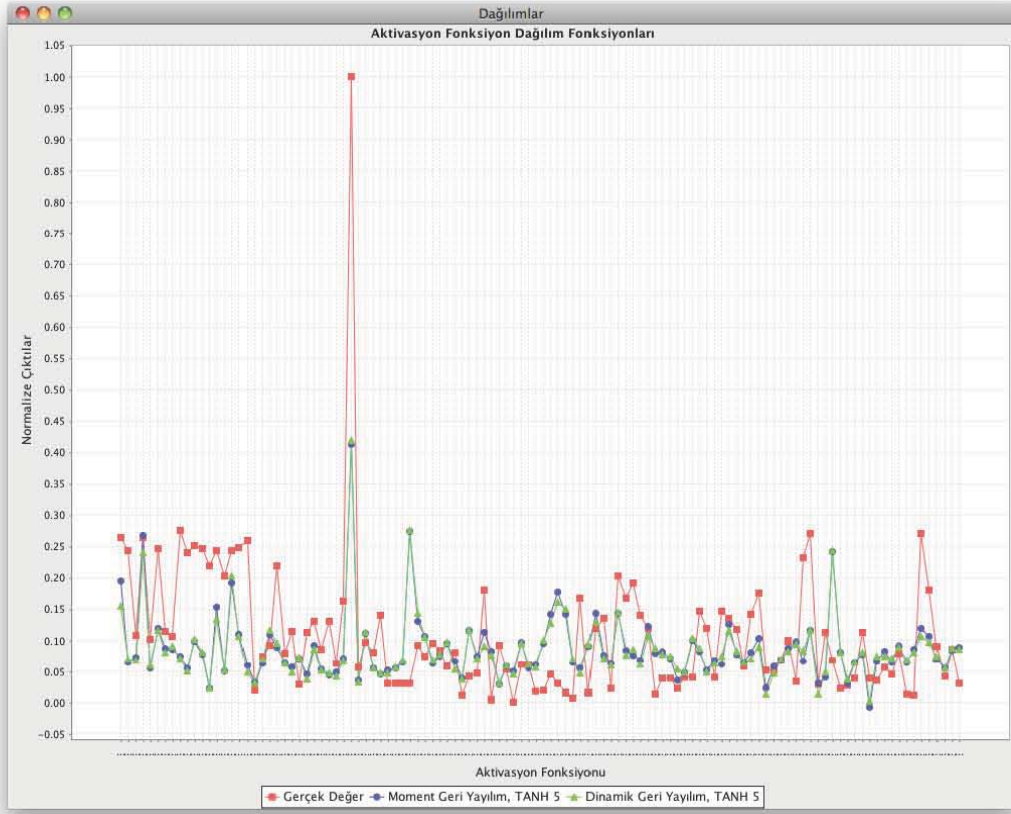
Şekil 3.12 Toplam başarı grafiği

Şekil 3.11'de gösterilen Dağılım Grafiklerini Çiz tuşuna basılarak, seçilen başarılı modellerin, sonuçları grafik olarak gösterilir. Yazılım bu grafiklerin yanına, veri setindeki gerçek çıktı grafiğini de ekler. Böylelikle, beklenen değer grafiği ile diğer grafikleri karşılaştırarak, beklene sonuçlara en yakın dağılım gösteren modelleri gözlemleyebiliriz.



Şekil 3.13 En hızlı ve en az özyineleme modelleri yayılım grafiği

Şekil 3.13'de en hızlı öğrenme ve en az özyineleme sayısına sahip modellerin çıktılarının gerçek değer ile birlikte grafiği verilmektedir. Grafik bize, en hızlı öğrenmenin, gerçek değerlerle çok iyi bir şekilde örtüşmediğini göstermektedir. En az özyineleme sayısına ulaşan model, en hızlı öğrenmeye nazaran daha iyi örtüştüğünü göstermektedir.



Şekil 3.14 Diğer örnek fonksiyonların yayılım grafikleri

Şekil 3.14'de ise, moment geri yayılım ve dinamik geri dağılım algoritmasını TANH aktivasyon fonksiyonunun kullanan 5 ara katmanlı modellerin yayılım grafikleri verilmiştir. Bu modellerde Şekil 3.13'de verilen modellerden daha iyi bir yayılıma sahiptir.



Şekil 3.15 Sonuçları excel e kaydetme diyalogu

Sonuçları excel dosya biçiminde kaydetmek için Şekil 3.11'de gösterilen Kaydet tuşuna basmak gerekmektedir. Şekil 3.15'de gösterilen diyalogdan kullanıcı çıktı olarak zamanları veya tüm çıktıları ayrı ayrı kaydetme şansına sahiptir. Şekil 3.15'de gösterilen Kaydet tuşuna basıldığı zaman sonuçlar excel formatında, yazılımın çalıştığı dizine kaydeder. Kaydedilen dosyalar "zaman.xls" ve "tüm çıktılar.xls" adları ile kaydedilir.

"zaman.xls" dosyasında tüm başarılı öğrenmelerinin hangi öğrenme yöntemi, aktivasyon fonksiyonu, kaç tane ara katmanla gerçekleştiği ve bu gerçekleşmelerin zaman süreleri verilir.

"tüm çıktılar.xls" de ise her bir öğrenen modelin ürettikleri çıktılar ve gerçek çıktı değerleri yer alır.

3.4.3 Yapay sinir ağıları model sonuçları

Yazılım Çizelge 3.1'de verilen değerlerle çalıştırılmıştır. Amaç geniş bir ara katman sayısında, tüm aktivasyon fonksiyonlarının nasıl çalıştığını göstermektir.

Çizelge 3.1 Yazılım çalışma başlangıç değerleri

| Algoritma | Strateji | Hata oranı | Ara Katman Sayısı Aralığı | En Fazla Deneme Sayısı |
|----------------------|-----------------------|------------|---------------------------|------------------------|
| İleri yayılım | Tüm olasılıkları dene | 0.01 | 5 – 15 | 100000 |
| Dinamik geri besleme | Tüm olasılıkları dene | 0.01 | 5 – 15 | 100000 |
| Moment geri besleme | Tüm olasılıkları dene | 0.01 | 5 – 15 | 100000 |

Deneme sonuçları şu şekildedir:

- Tüm modellerin çalışması 12 saat sürmüştür. Çizelge 3.2'de öğrenmeyi başaran fonksiyonların performans değerleri verilmiştir.
- Toplam 275 farklı model denenmiştir.
- İleri beslemeli öğrenme ile öğrenme gerçekleşmemiştir. 9 girdi, 1 çıktı ve 114 örnek olan ve doğrusal olmayan bu sistemde ileri beslemeli öğrenme ile sonuç hesaplanamamıştır.
- Öğrenmeyi tamamlayan öğrenme algoritmaları dinamik ve moment geri yayılım algoritmalarıdır.
- 100000 deneme için öğrenme ara katman sayısı 5 den sonra başlamıştır.
- Sadece gaus, sigmoid ve tanh aktivasyon fonksiyonları öğrenmeyi tamamlamıştır.
- Sigmoid fonksiyonu 15 ara katmanla beraber öğrenmeye başlamıştır.
- Gaus fonksiyonu 8 ara katmanla beraber öğrenmeye başlarken, tanh fonksiyonu 11 ara katman sayısından sonra öğrenmeye başlamıştır.
- En hızlı öğrenme ve en az özyinelemeli öğrenmeye; dinamik geri yayılım algoritmasını kullanan, gaus aktivasyon fonksiyonlu ve 14 ara katmanlı model ulaşmıştır. Öğrenme 3 saniye ve 1690 özyineleme ile gerçekleşmiştir.
- Şekil 3.16'da gösterildiği üzere, ağın eğitilmesi için geçen zaman ile ara katman sayısı arasındaki ilişki doğrusal değildir. Problem için uygulanan çözümde, öğrenme zamanı 11, 12, 13 ve 14 ara katman kullanıldığı zaman en düşük seviyelerdedir. Ara katman sayısı 5, 6 ve 7 iken ağın öğrenme zamanı, 8 ara katmanlıya göre fazladır. Sonuç olarak, öğrenme zamanı ara katman sayısı artarken belli bir noktaya kadar düşer (14 ara katman), sonra tekrar artmaya başlar.

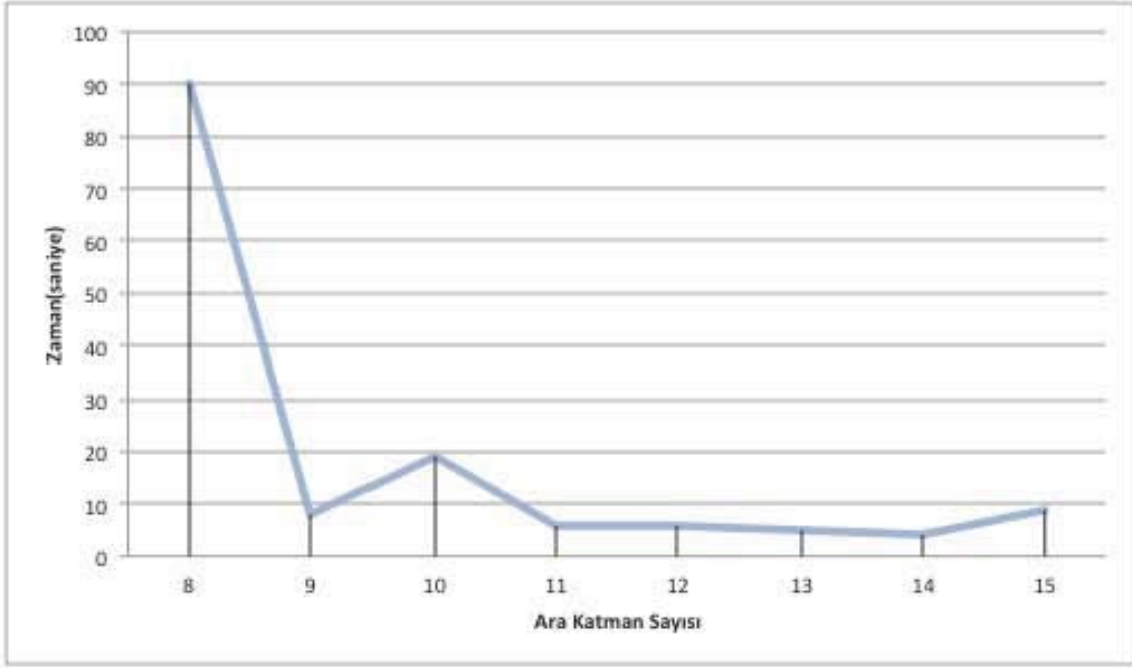
- Şekil 3.17’de gösterildiği üzere, ağın eğitilmesi için yapılması gereken özyineleme sayısı da zaman ile benzer bir grafik sergiler. Öğrenme için geçen özyineleme sayısı ara katman sayısı artarken belli bir noktaya kadar düşer (14 ara katman), sonra tekrar artmaya başlar.
- Şekil 3.18’de görüleceği üzere, ağ öğrenme çıktıları ile gerçek çıktı değerleri arasındaki farklar çok azdır. Grafikten görüleceği gibi en hızlı öğrenen model, gerçek çıktıları çok benzer çıktılar üretmiştir. Bu olabilecek en iyi durumdur; çünkü hem kısa sürede öğrenme tamamlanmakta hem de gerçek çıktılara çok yakın sonuçlar öğrenmiştir.
- Şekil 3.18’de gösterilen grafiğe ait veriler EK 3’de verilmiştir. Ayrıca beklenen çıktılar ile, hesaplanan değerler arasındaki farklar hem EK 3 de değer olarak verilmiş hem de EK 4 te grafik olarak sunulmuştur. Şekil 3.18 büyük hali EK 5 te daha büyük olarak verilmiştir.
- Diğer aktivasyon fonksiyonlarıyla öğrenme tamamlanması için daha fazla özyineleme yapılması gerekmektedir. Bu da daha fazla işlem gücü gerektirmektedir.
- Geri yayılımda en çok kullanılan aktivasyon fonksiyonu sigmoid fonksiyonudur. Geliştirilen çözümde sigmoid fonksiyonun ağ öğrenmesi için çok kullanışlı olduğu gösterilmiştir.

Çizelge 3.2 Örnek çözüm sonuçları

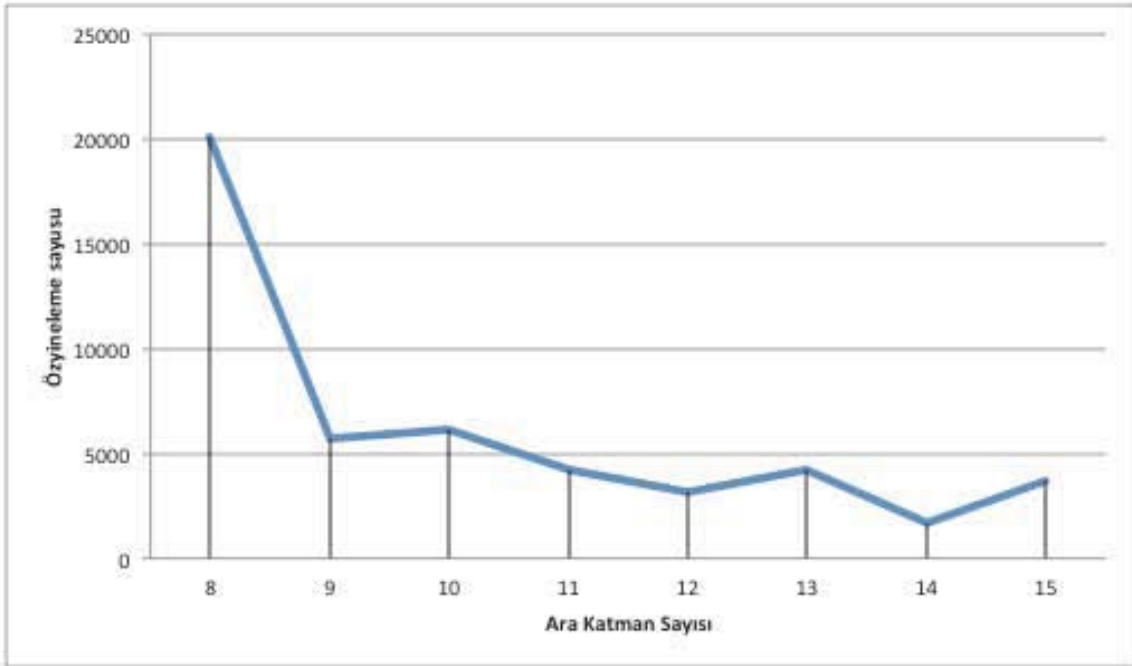
| Öğrenme Algoritması | Aktivasyon | Ara Katman | Zaman(saniye) | Özyineleme |
|----------------------|------------|------------|---------------|------------|
| Dinamik Geri Yayılım | GAUSSIAN | 8 | 24 | 20127 |
| Moment Geri Yayılım | GAUSSIAN | 8 | 90 | 71940 |
| Dinamik Geri Yayılım | GAUSSIAN | 9 | 7 | 5732 |
| Moment Geri Yayılım | GAUSSIAN | 9 | 8 | 5931 |
| Dinamik Geri Yayılım | GAUSSIAN | 10 | 9 | 6141 |
| Moment Geri Yayılım | GAUSSIAN | 10 | 19 | 12157 |

Çizelge 3.2 Örnek çözüm sonuçları devamı

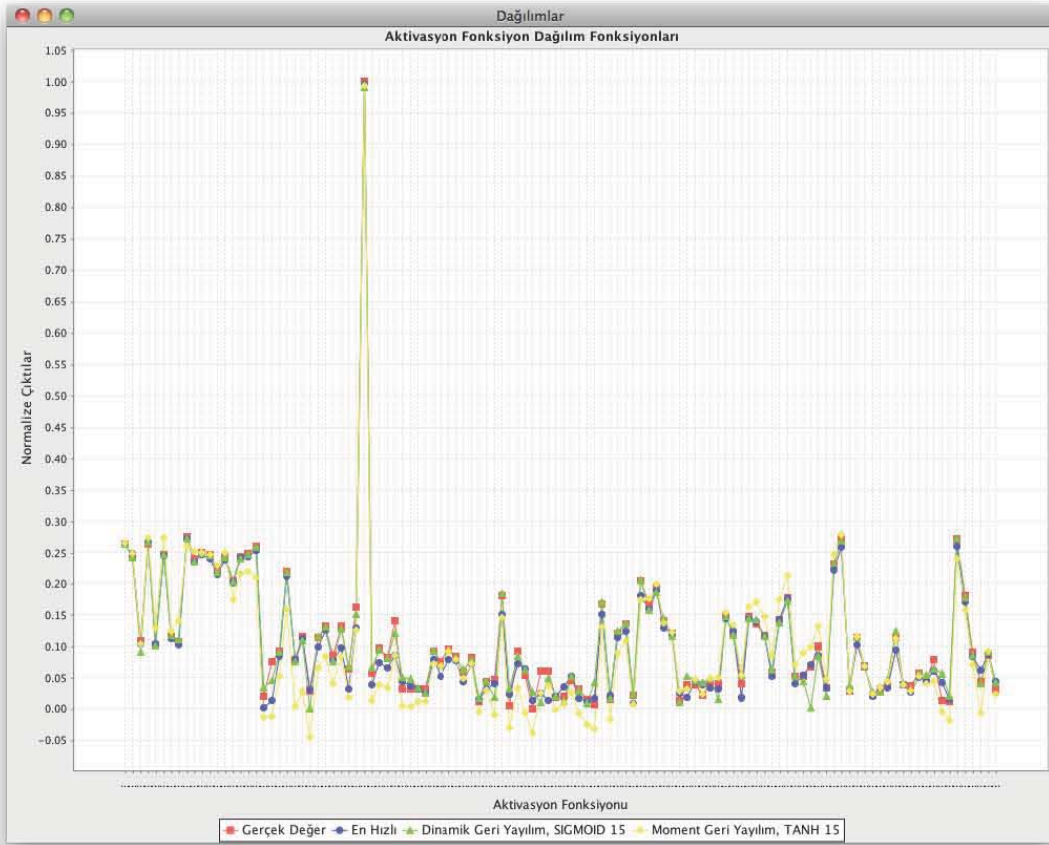
| | | | | |
|----------------------|----------|----|-----|-------|
| Dinamik Geri Yayılım | TANH | 11 | 115 | 66583 |
| Dinamik Geri Yayılım | GAUSSIAN | 11 | 7 | 4175 |
| Moment Geri Yayılım | TANH | 11 | 82 | 47525 |
| Moment Geri Yayılım | GAUSSIAN | 11 | 6 | 3408 |
| Dinamik Geri Yayılım | TANH | 12 | 133 | 70658 |
| Dinamik Geri Yayılım | GAUSSIAN | 12 | 5 | 3157 |
| Moment Geri Yayılım | GAUSSIAN | 12 | 6 | 3041 |
| Dinamik Geri Yayılım | TANH | 13 | 86 | 34394 |
| Dinamik Geri Yayılım | GAUSSIAN | 13 | 10 | 4223 |
| Moment Geri Yayılım | TANH | 13 | 96 | 43362 |
| Moment Geri Yayılım | GAUSSIAN | 13 | 5 | 2529 |
| Dinamik Geri Yayılım | TANH | 14 | 112 | 43509 |
| Dinamik Geri Yayılım | GAUSSIAN | 14 | 3 | 1690 |
| Moment Geri Yayılım | TANH | 14 | 46 | 18881 |
| Moment Geri Yayılım | GAUSSIAN | 14 | 4 | 1928 |
| Dinamik Geri Yayılım | SIGMOID | 15 | 256 | 88837 |
| Dinamik Geri Yayılım | TANH | 15 | 51 | 20828 |
| Dinamik Geri Yayılım | GAUSSIAN | 15 | 9 | 3692 |
| Moment Geri Yayılım | TANH | 15 | 65 | 27001 |
| Moment Geri Yayılım | GAUSSIAN | 15 | 9 | 3637 |



Şekil 3.16 Ara katman sayısının zamana etkisi



Şekil 3.17 Ara katman sayısının özyineleme sayısına etkisi



Şekil 3.18 Yayılım fonksiyonları

4. SONUÇ VE ÖNERİLER

Günümüz piyasa koşullarında, taleplere hızlı ve doğru bir şekilde cevap verebilmek çok önemlidir. Kısıtlı kaynakları en verimli şekilde kullanabilmek için gelen yeni taleplere doğru biçimde zaman tahmini yapmak herhangi bir firma için çok önemlidir. Tahminlerdeki hatalar milyon lira seviyelerinde zarara yol açabilmektedir. Yapay sinir ağları birçok sektörde, geçmişten gelen verileri işleyip; bu veriler için geleceğe yönelik sonuç üretmede çok başarılı bir yöntemdir. Hemen hemen her akıllı cihaz yapay sinir ağları ile yapılan analizlere göre işlevlerini yerine getirmektedir.

Bu çalışmada ele alınan problem çözümü için yapay sinir ağları kullanılmış ve yapay sinir ağları modellemesi dinamik bir hale getirilip kullanıcının isteyebileceği soruya cevap verecek bir yazılım geliştirilmiştir. Yazılım üç farklı öğrenme algoritmasını modellemiştir, bunlar ileri yayımlı öğrenme, dinamik geri yayımlı öğren ve moment geri yayımlı öğrenmedir.

Çalışmada ileri doğru yayımlı öğrenme modeli ile problem çözümü bulunamamıştır. İleri doğru yayımlı modellerde geriye doğru bilgi aktarılmadığı için fazla sayıda girdi ve fazla sayıda örnek sayısı olan problem çözümsüz kalmıştır. Dinamik geri yayılım ve moment geri yayılım algoritmaları ile probleme çözüm bulunmuştur.

Yapılan çalışmanın diğer çalışmalara göre farkları, her kombinasyonun tek bir seçenek ile denenmesi, excel dosya formatında çıktı kaydetmesi ve grafikleri kullanıcının istediği şekilde ortaya çıkarmasıdır. Diğer bir farkı ise, veri setlerini kolay bir şekilde basit bir CSV dosyasından okuyabilmesidir.

Yazılım ayrıca en kısa öğrenme ve en az özyineleme sayısı ile öğrenmenin hangi parametrelerle gerçekleştiğini görüp, sadece bu parametrelerle problem çözümüne kolayca ulaşabilmektedir.

En hızlı öğrenme için yapılan tüm denemelerde gaus fonksiyonu en hızlı öğrenmeyi sağlamıştır. Daha sonra tanh fonksiyonu ve sigmoid sıralaması elde edilmiştir. EK 5 grafiğine bakılacak olunursa beklenen değerler ve hesaplanan değerler arasında çok fark olmadığı görülebilir. Ara katman sayısı arttıkça hesaplama hem daha az zaman almaktadır.

Çalışmada öğrenme tamamlayan tüm modeller için hata oranı 0.01 olmuştur. Fakat modeller öğrenmeyi tamamlarken geçen süre ve özyineleme sayıları farklılık göstermiştir.

Her kullanıcıya hitap edebilen kullanıcı dostu ara yüz ile firmaya özel zaman tahmini için bir karar destek sistemi ortaya koyulmuştur. Bu çalışma ile aşağıdaki faydalar sağlanmıştır.

- Zaman tahmini konusunda tecrübesi olmayan bir çalışan için etkin zaman tahmini yapabilecek bir altyapı oluşturulmuştur
- Zaman etüdünün, zaman alıcı, efor gerektiren bir çalışma olmaktan çıkartıp, ucuz ve hızlı bir hale getirmiştir
- Zamanı etkileyecek tüm faktörler ve bunların etkileri araştırmaya dahil edilmiştir
- Standart zamanda artışa neden olabilecek faktörleri izleme olanağını ortaya çıkarmıştır
- Firmaya ekonomik yönüyle fayda sağlamıştır

Bu çalışmanın diğer tahmin yöntemlerine göre üstünlükleri şu şekilde sıralanabilir.

- Ucuz olması
- Hızlı olması
- Tecrübe gerektirmemesi
- Tahminlerde istenilen hata oranını yakalayabilmesi.

Firma bu çalışmanın sonucunda hızlı ve ucuz bir tahmin programına sahip olmuştur. Çalışma sonucunda geliştirilen program herhangi başka bir firmada kullanılabilir. Ekstra bir değişiklik yapılmasına gerek yoktur.

Bundan sonra yapılacak çalışmalarda, gerçek çıktılarla en çok yayılımı benzeyen modellerin belirtilmesi sağlanabilir. Böylelikle kullanıcı bu modeli problemin çözümünde gerçek değerlere en çok benzeyen model olarak kabul edip bu model katsayılarını saklayıp hızlı bir şekilde, en az hatalı ve en gerçek sonuçlara kolayca ulaşabilir. Ayrıca grafik gösterimine yeni gösterimler eklenebilir.

Bu çalışmada problem, ÇKA yapay sinir ağı ile çözülmüştür. Problem çözülmesinde JAVA kullanılmıştır, böylece geliştirilen yazılım, tüm işletim sistemlerinde ve tüm versiyonlarda çalışabilecek olgunlukta sunulmuştur.

KAYNAKLAR LİSTESİ

- [1] Akal, Z., İş Etüdü, ILO, MPM Yayınları Yayın No: 29, Ankara, 1997.
- [2] İlhan, R., Burdurlu E., İş Etüdü, Yardımcı Ders Kitabı, Hacettepe Üniversitesi, A. Endüstri Mühendisliği Bölümü, Beytepe, 1989.
- [3] Civalek, Ö., Dairesel Plakların Nöro-Fuzzy Tekniği ile Analizi, DEÜ Mühendislik Fakültesi Fen ve Mühendislik Dergisi, Cilt. 1, Sayı 2, 13-31, 1999.
- [4] Aleksander, I., Morton, I., An introduction to neural computing, International Thomson Computer Press., 1995.
- [5] Fausett, L., Fundamentals of neural networks, architectures, algorithms, and applications., Prentice-Hall, Inc., New-Jersey, 1994.
- [6] Kohonen, T., Content addressable memories, Springer-Verlag, New-York, 1980.
- [7] Hopfield, JJ., Neural networks and physical systems with emergent collective computational abilities., In Proc. of National Academy of Sciences, 1982;79: 2554- 58.
- [8] McCulloch, WS., and Pitts, W., A logical calculus of ideas imminent in nervous activity., Bull. Math. Biophysics, 1943;5: 115-33.
- [9] Kamiyama N., Iijama N., Taguchi A., Mitsui H., Yoshida Y., Sone M., Tuning of learning rate and momentum on backpropagation, Japan, 1992.
- [10] Elmas Ç., Yapay Sinir Ağları, Seçkin Yayıncılık, Ankara, 2003.
- [11] Keleşoğlu Ö., Fırat A., Tuğla Duvardaki ve Tesisattaki Isı Kaybının Yapay Sinir Ağları İle Belirlenmesi, Elazığ 2005.

- [12] Öztürk H., Konfeksiyonda MTM Yöntemiyle Zaman ve Metod Etüdü Çalışmaları, Ankara, 1991.
- [13] Öztemel E., Yapay Sinir Ağları, İstanbul, 2006.
- [14] Xiao-Hu Yu*and Guo-An Chen, Efficient Estimation of Dynamically Optimal Learning Rate and Momentum for Backpropagation Learning, China, 2005.
- [15] Liang J., Madan M., Gupta, Stable Dynamic Backpropagation Learning in Recurrent Neural Networks, Fellow, IEEE, 1999.
- [16] Cheng Yu C., Bin-Da L., A Backpropagation Algorithm with Adaptive Learning and Momemntum Coefficient, Taiwan, 2002.
- [17] De Jesus O., Hagan M. T., Backpropagation Algorithms for a Broad Class of Dynamic Networks, IEEE Transaction on Neural Networks, Vol. 18, No. 1, JANUARY 2007

EKLER LİSTESİ

| | <u>Sayfa</u> |
|--|--------------|
| EK 1: Fabrikadan Alınan Normalize Olmayan Veri Seti | 49 |
| EK 2: Normalize Veri Seti | 53 |
| EK 3: Öğrenme Sonunda Hesaplanan Çıktı Değerleri | 57 |
| EK 4: Beklenen Değerler ile Çıktılar Arasındaki Fark Grafik | 62 |
| EK 5: Yayılım Grafiği Büyük | 63 |
| EK 6: İleri Yayılmalı Öğrenme JAVA Kodu | 64 |
| EK 7: Dinamik Geri Yayılım Öğrenme JAVA Kodu | 69 |
| EK 8: Moment Geri Yayılım Öğrenme JAVA Kodu | 72 |