

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NEURAL NETWORK YAKLAŞIMI İLE ÜRETİM SİSTEM
PARAMETRELERİNİN TAHMİNİ:
SERAMİK FİRMASINDA UYGULAMA

YÜKSEK LİSANS TEZİ

End. Müh. Seher ARSLANKAYA

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

OCAK 2001

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

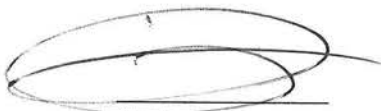
NEURAL NETWORK YAKLAŞIMI İLE ÜRETİM
SİSTEM PARAMETRELERİNİN TAHMİNİ:
SERAMİK FİRMASINDA UYGULAMA

YÜKSEK LİSANS TEZİ

End. Müh. Seher ARSLANKAYA

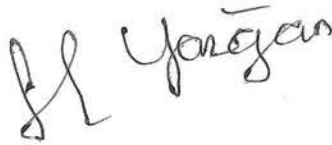
Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez/....../2001 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.



Jüri Başkanı

Yrd. Doç. Dr. İ. H. Şahin
C. S. DİM ÖZEL



Jüri Üyesi

Yrd. Doç. Dr.
Harun Reşit YAZGAN



Jüri Üyesi

Yrd. Doç. Dr. Cemil ÖZ

TEŞEKKÜR

Bu çalışmada yardımlarından dolayı sayın hocam Yrd.Doç.Dr. Harun Reşit YAZGAN'a, Yrd. Doç. Dr. Cemil ÖZ'e, Yrd. Doç .Dr. İsmail Hakkı CEDİMOĞLU'na, Arş. Gör. Nihal SÜTÜTEMİZ'e, ve benden desteğini esirgemeyen eşim Zeki ARSLANKAYA' ya teşekkür ederim.

Seher ARSLANKAYA

İÇİNDEKİLER

KISALTMALAR.....	vi
ŞEKİLLER LİSTESİ.....	vii
TABLolar LİSTESİ.....	viii
ÖZET.....	ix
SUMMARY.....	x

BÖLÜM 1

GİRİŞ	1
-------------	---

BÖLÜM 2.

SİMÜLASYON

2.1.Simülasyon nedir?.....	3
2.2.Simülasyona Ne zaman Başvurulmalıdır?	5
2.3.Simülasyonun Kullanma Alanları	5
2.3.1.Üretim Sistemlerinin Simülasyonu	6
2.3.2. Servis Sistemlerinin Simülasyonu	6
2.4.Simülasyonun Avantajları.....	6
2.5.Simülasyonun Dezavantajları	7
2.6. Simülasyon Süreci.....	7
2.7.Simülasyon Dilleri	8

2.7.1 ProModel paket programı.....	8
2.7.2. Promodel 'da model oluřturma ařamaları (Temel elemanlar)	10
2.7.2.1.General information (Genel bilgiler)	10
2.7.2.2.Locations (Yerleřimler)	10
2.7.2.3.Pathnetwork (Ađ řebekesi)	11
2.7.2.4.Resources (Kaynaklar)	11
2.7.2.5.Entities (Girdiler)	12
2.7.2.6.Processing (Prosesleme)	13
2.7.2.8.Arrival (Geliřler)	14
2.8. Modelin alıřtırılması.....	16

BÖLÜM 3

YAPAY SİNİR AĞLARI

3.1.Giriř.....	17
3.2. YSA'nın Tanımı.....	17
3.3. YSA'nın Tarihesi.....	18
3.4. Yapay Sinir Ađlarının Özellikleri.....	19
3.5. Biyolojik Sinir Hücresinin Yapısı ve Nöron Modeli.....	20
3.6. Yapay Sinir Modeli.....	22
3.7. YSA'ların özellikleri.....	23
3.7.1. Paralellik	23
3.7.2. Hata toleransı	24
3.7.3. Gerekleme kolaylıđı	24
3.7.4. Öđrenebilirlik ve adaptasyon	24
3.8. Yapay Sinir Ađlarının Temel Bileřenleri.....	25

3.9. YSA'nın Bağlantı Elemanları	28
3.9.1. Bağlantı modelleri.....	28
3.10. Yapay Sinir Ağlarında Bilgi İşleme, Öğrenme ve Zeki Davranış.....	31
3.10.1. Eğitici öğrenme	32
3.10.2. Takviyeli öğrenme.....	32
3.10.3. Eğitici öğrenme.....	33
3.11. Yapay Sinir Ağlarında Ağırlık Uzayı.....	33
3.12. Hata Minimizasyonu ve Gradient Descent Tanımı.....	34
3.13. Temel Bazı YSA Modelleri.....	36
3.13.1. Hopfield YSA modeli	36
3.13.2. Adaline ve Madeline.....	38
3.13.3. Çok katmanlı algılayıcı (ÇKA) ve backpropagation YSA.....	39
3.13.3.1. Geriye yayılım algoritması.....	41
3.14. YSA' da Kullanılan Önemli Öğrenme Algoritmaları.....	46

BÖLÜM 4.

UYGULAMA: NEURAL NETWORK YAKLAŞIMI İLE ÜRETİM PARAMETRELERİNİN BELİRLENMESİ: SERAMİK FİRMASINDA UYGULAMA

4.1. Giriş.....	49
4.2. Üretim Prosesleri.....	49
4.2.1. Çamurhanedeki Üretim Aşamaları	50
4.2.2. Dökümhanedeki üretim aşamaları.....	50
4.2.3. Kurutma odasındaki üretim aşamaları.....	52
4.2.4. Rötuş bölümündeki üretim aşamaları	53
4.2.5. Sırlama bölümündeki üretim aşamaları.....	53
4.2.6. Fırınlama bölümündeki üretim aşamaları.....	53
4.2.7. Kalite Kontrol.....	53

4.2.8. Depo.....	54
4.3 Üretim Parametrelerinin Neural Network İle Tahmin Edilmesi.....	54
4.3.1. Çelvit A.Ş'nin ProModel paket programı ile modellenmesi.....	54
4.3.1.1. Model ile ilgili genel bilgiler (General information).....	55
4.3.1.2. Modelin yerleşimlerinin girilmesi (Locations).....	55
4.3.1.3. Girdilerin belirlenmesi (Entities).....	56
4.3.1.4. Kaynakların belirlenmesi (Resources).....	57
4.3.1.5. Modelin ağlarının ve yollarının belirlenmesi (Path networks)....	58
4.3.1.6. Modelin proseslerinin belirlenmesi (Processing).....	59
4.3.1.7. Varışların tanımlanması (Arrivals)	63
4.3.2. Farklı üretim parametreleri ile simülasyon sonuçlarının elde edilmesi....	63
4.3.3. Neural network modelinin oluşturulması.....	66
4.3.3.1. Önerilen ANN modeli.....	67
4.3.4. Neural network kullanılarak üretim parametrelerini tahmin edilmesi.....	74
4.4. İstatistiksel Analiz.....	74
4.4.1. İki Modelin ortalamaları arasındaki farkın anlamlılık testi.....	75
 BÖLÜM 5.	
SONUÇLAR VE ÖNERİLER.....	85
 KAYNAKLAR.....	97
 ÖZGEÇMİŞ.....	102

KISALTMALAR LİSTESİ

YSA	: Yapay Sinir Ağları
YZ	: Yapay Zeka
ÇKA	: Çok Katmanlı Algılayıcı
GPS	: Genel Problem Çözücü
HYSA	: Hücresel Yapay Sinir Ağı
BM	: Boltzmann Machine
SCADA	: Supervisory Control and Data Acquisition
LTU	: Linear Thresholded Unit
W	: Ağırlık
ANN	: An Neural Network

ŞEKİLLER LİSTESİ

Şekil 2.1.Promodel'de genel yapısı.....	9
Şekil 3.1. Tipik olarak sinir ağının fiziksel bağlantı şekli.....	20
Şekil 3.2. Bir nöronun şematik yapısı.....	21
Şekil 3.3. Yapay sinir modeli	22
Şekil 3.4. Ağırlık vektörünün değişimi.....	33
Şekil 3.5. Hata minimizasyonu ve "Gradient Descent" geometrik yorumu.....	35
Şekil 3.6. Üç katmanlı yapay neural ağ.....	36
Şekil 3.7. Basit Adaline.....	38
Şekil 3.8. Çok katmanlı algılayıcı (ÇKA) YSA' nın yapısı.....	40
Şekil 3.9. Geriye yayılım ağ yapısı.....	42
Şekil 4.1. Üretim aşamaları.....	54
Şekil 4.2. Mevcut durumdaki proseslerin belirlenmesi.....	60
Şekil 4.3. Geliştirilen modeldeki proseslerin belirlenmesi.....	60
Şekil 4.4. Yeni bir ANN kurulması ve simülasyon sonuçlarıyla bütünleştirilmesi.....	66
Şekil 4.5. ANN'in Eğitimi.....	68
Şekil 4.6. Eğitilen ANN'in test edilmesi.....	69

TABLÖLER LİSTESİ

Tablo 3.1. En çok kullanılan toplama fonksiyonları.....	26
Tablo 3.2. Aktivasyon fonksiyonu türleri.....	27
Tablo 3.3. YSA' da çeşitli bağlantı modelleri.....	30
Tablo 4.1. Modeldeki yerleşimler ve özellikleri.....	56
Tablo 4.2. Modelde kullanılan girdiler ve özellikleri.....	57
Tablo 4.3. Modelde kullanılan kaynak isimleri ve özellikleri.....	58
Tablo 4.4. Modeldeki model ağırları ve yolları.....	59
Tablo 4.5. Modeldeki işlem adımları ve süreleri.....	61
Tablo 4.5. (Devam) Modeldeki işlem adımları ve süreleri.....	62
Tablo 4.6. Modele gelen entity geliş zamanları, frekansları ve miktarları.....	63
Tablo 4.7. Girdi ve çıktı parametreleri.....	64
Tablo 4.7. (Devam) Girdi ve çıktı parametreleri.....	65
Tablo 4.8. ANN katmanları ve değişkenleri.....	67
Tablo 4.9. Neural network eğitim seti.....	70
Tablo 4.10. Neural network eğitim setinin normalize edilmesi.....	71
Tablo 4.11. Neural network tahmin seti.....	72
Tablo 4.12. Neural network tahmin setinin normalize edilmesi.....	73
Tablo 4.13. z ve z_k değerlerinin karşılaştırılması.....	77

ÖZET

Anahtar kelimeler: ProModel simülasyon paket programı, yapay sinir ağıları, backpropagation algoritması

Bu çalışmada yeni bir Neural Network yaklaşımı (ANN) geliştirildi. Türkiye’de bir seramik fabrikasında yapılan bu çalışmada üretimin çıktıları kullanılarak sistem üretim parametreleri tahmin edilmiştir. Çalışma gerçekleştirilirken üç ürün üzerinde duruldu, tuvalet taşı, klozet ve lavabo. Neural Network ile üretim parametrelerini tahmin ederken dört adım takip edilmiştir. Üretim sistem parametrelerini çözmek için aşağıda anlatılan yollar takip edilmiştir.

Birincisi, işletmenin üretim prosesinin ProModel simülasyon paket programı ile modellenmesidir.

İkincisi, farklı üretim parametreleri ile simülasyon sonuçlarının elde edilmesidir.

Üçüncüsü, simülasyon sonuçlarından oluşturulan eğitim datasıyla neural network geliştirilmesidir.

Son olarak ta neural network kullanılarak ürünlerin talep miktarlarıyla üretim parametrelerinin tahmin edilmesidir.

Neural Network’un eğitimi sırasında girdi değişkenleri sistem parametresi ve çıktı değişkenleri üretim miktarları olarak hesap edilir. Geliştirilen ANN’ le, simülasyon sürecindeki üretim miktarlarının, girdi değişkenlerinin tahmini hem daha kısa sürede ve hem de daha kesin olarak bulunmaktadır. Bu yaklaşım firma tarafından test edilmiş ve başarılı bulunmuştur.

Son olarakta istatistiksel analiz yapılmıştır.

SUMMARY

AN ESTIMATION OF PRODUCTION SYSTEM DESIGN PARAMETER WITH A NEW NETWORK APPROACH: A CASE STUDY IN THE CERAMIC FIRM

Key words: ProModel Simulation Packed Programming, Artificial Neural Network, Backpropagation Algorithm,

A new artificial neural network (ANN) approach has been developed in this study. The problem is to determine production system parameters with considering production output at a ceramic factory in Turkey. Three products are considered as an example such as, toilet marble, washbasin and closet. The following stages are used in order to solve the production system parameters.

At the first stage, the production system of three products is modeled.

At the second stage, the system is simulated and generated system performance values under different system parameters.

At third stage, an ANN is developed and trained with using the simulation results.

At the last stage, the trained neural network estimates the system parameters with considering production quantities of the products.

During teaching process of the neural network, the system parameters as input variables and production quantities as output variables are considered. The results show that the new proposed ANN estimates much faster and finds out more accurate variables than a simulation approach in terms of determining system parameters with considering production demands of the products. The approach has been tested for the company and results were found very encouraging.

At the last end, statistical analysis made.

BÖLÜM 1

GİRİŞ

Artan uluslar arası rekabet ile sürekli gelişen bilgisayar teknolojisi, işletmelerin bütün alanlarında önemli bir role sahiptir. Son yıllarda özellikle bilgisayarların imalat içindeki kullanımı ise klasik üretim sistemlerinin yerini, otomasyona dayalı üretim sistemlerine bırakmıştır.

Bu çalışmanın ikinci bölümünde simülasyon hakkında bilgi verilmiştir. Simülasyonun tanımı yapılmış, avantajları, dezavantajları, kullanım alanları, kolaylıkları vb. konulardan bahsedilmiştir. Simülasyon doğru karar verme amacına hizmet için geliştirilmiş, sisteminizin içindeki aksaklıkları problem olmadan evvel ortaya çıkaran, kararları vermeden etkilerini inceleyen bir karar verme mekanizması ve yönetim bakış açısıdır.

Üçüncü bölümde Yapay sinir ağlarından bahsedilmiştir. YSA'nın çekiciliği, mükemmel bir yapı olan insan beyninin işleyişinin kopyalanması gibi bir ayrıcalıktan ileri gelmektedir. İnsanlık her zaman için kendi düşünce ve fikir kabiliyeti üzerinde durmuş ve bunun mantığını anlamaya ve gerçek hayatta uygulamaya çalışmıştır. Bugün geline aşamada ise beyin bağlantıları üzerinde oldukça yol alınmış olmasına rağmen işleyişi üzerinde elde edilen bilgi henüz yeterli düzeyde değildir. Burada tezde kullanılan Backprobagation Algoritması da anlatılmaktadır.

Dördüncü bölüm uygulamanın yapıldığı bölümdür. Önce bir işletmedeki üretim planlama sorunları simülasyon programı olan ProModel paketi ile çözülmeye çalışılmıştır. Bunu için önce mevcut işletme modellenmiş daha sonra aksaklıklar incelenerek alternatif bir çözüm geliştirilmiştir. Bu sayede işletmede üretim

miktarları artırılmıştır. Fakat tek dezavantajı simülasyon süresinin 30 dk. ile 5 saat arasında değişmesidir. Buda üretim parametrelerinin tahmininde büyük zaman kayıplarına yol açmaktadır. Bunu engellemek için bu model Neural Network' öğretilmiştir. Bunun için Backprobagation algoritması kullanılmıştır. Neural Network ile üretim parametrelerini tahmin ederken dört adım takip edilmektedir. Birincisi, işletmenin üretim prosesinin ProModel simülasyon paket programı ile modellenmesidir. İkincisi, farklı üretim parametreleri ile simülasyon sonuçlarını elde edilmesidir. Üçüncüsü, simülasyon sonuçlarından oluşturulan eğitim datasıyla Neural Network geliştirilmesidir. Son olarakta Neural Network kullanılarak ürünlerin talep miktarlarıyla üretim parametrelerinin tahmin edilmesidir. Eğitim gerçekleştirildikten sonra test edilmiş sonuçlar kontrol edilmiştir. Sonuçların yakın çıktığı gözlenmiştir. Bu varsayımı kuvvetlendirmek için istatistiksel analiz yapılmıştır. t testi uygulanmıştır.

Beşinci bölümde yapılan bu çalışmadaki sonuçlar üzerinde durulmuş ve gerekli tavsiyeler yapılmıştır.

BÖLÜM 2: SİMÜLASYON

2.1.Simülasyon Nedir?

Bir sistemin simülasyonu, bu sistemi temsil edebilecek bir model oluşturma işlemidir. Bu model temsil ettiği sistem üzerinde yapılması çok pahalı olan veya mümkün olmayan işlemlerin yapılmasına imkan verir.

Simülasyon doğru karar verme amacına hizmet için geliştirilmiş, sisteminizin içindeki aksaklıkları problem olmadan evvel ortaya çıkaran, kararları vermeden etkilerini inceleyen bir karar verme mekanizması ve yönetim bakış açısıdır. İlk bakışta medyumluk gibi bir anlam çıksa da simülasyon metodu onu kullananın teknik bilgisine ve yaratıcılık sanatına da bağlıdır. Dünya standardında üretim yapan, tepede kalmak isteyen veya kendini geliştirmek isteyen firmaların 90'lı yıllarda bilinçli olarak kullandıkları bir teknolojik üstünlük olan simülasyon en basit anlamı ile “gerçeğin taklit edilmesi” dir. (Erkut (1992))

Simülasyon (Benzetim), bir sistemin modeli kurulurken yapılan çalışmaların, kazanılan deneyimlerin yönlendirilmesi, birleştirilmesi işlemleri anlamında kullanılır. Bu model, ya direk olarak sistemin kendisiyle ilgili elde edilen bir bilgi, bir deneyim ya da sistemle ilgili bir problemin analitik çözümüdür.

Simülasyon sistemi, birini biliyorsak diğeri içinde bilinen bu olayın karakteristiklerini kullanmaktır. Kıyaslamada bu olay tamamen aynı kabul edilmeyip birbirine oranlanmaktadır. İki olayı birbirine benzeterek, benzedikleri kanısına

vardıktan sonra olaylardan birisi için bulunan karakteristikleri öteki için kullanma metodudur.

Bir simülasyon çalışmasının temel amaçları şöyle sıralanabilir :

- Bir gerçek hayat sistemini girdi ve çıktılarıyla matematiksel olarak ifade etmek,
- Gerçek sistemi kurulan model üzerinden tanıyıp, araştırmak, değişik kararları ve seçenekler gerçek sistemde hiç bir değişiklik yapmadan deneyebilmek,
- Elde edilen ve bilgiler ışığında sistemle ilgili ön görüşlerde bulunabilme ve uygulamaya esas olan kararları belirleme

Simülasyon, özellikle analitik ve nümerik çözüm metotlarının uygulanmadığı veya uygulanmasının pratik olmadığı durumlarda kullanılır. Varsayımlarımızı yapabiliyorsak, sistemimiz de müsaitse ilk olarak analitik modelleri denememiz gerekmektedir. Simülasyon modelleriyle analitik modeller arasında karma modeller vardır. Analitik modeller cevap vermezse karma modeller, onlardan da sonuç alınamazsa simülasyon modeline gidilir. Birçok kompleks sistem için analitik metotlar ve konusu geçen karma metotlar, sisteme uygun olmayan, çok pahalı, zaman alıcı metotlardır.

Bu tanımlara göre simülasyon süreci hem modelin kurulması hem de problemin incelenmesi için modelin analitik olarak kullanılmasını içermektedir. Simülasyon ile modelleme :

- 1) Sistemin davranışını tanımlama,
- 2) Teori veya hipotez kurma,
- 3) Kurulan teoriyi sistemin gelecekteki davranışlarını tahmin etmek için kullanmak, şeklinde olan deneme ve uygulama metotları içerir.

Simülasyon metotlarıyla, sistemin zaman içerisindeki davranışları ele alınır. Metodun amacı, değişik koşullar altında sistem elemanlarının davranışlarını incelemek, elemanlar arasındaki ilişkiyi anlamak, ortaya çıkarmaktır. Matematik anlamda bir çözüm aranmaz. Ama simülasyon çalışmalarının da bir amacı vardır ve ulaşılan amaca da çözüm denilebilir. Bir modelin amacına ulaşabilmesi için iki önemli özelliği içermesi gerekir; bir yandan gerçek sistemin bütün önemli özelliklerini gösteren iyi bir yaklaşım, bir yandan da anlaşılabilir ve çözülebilir olmalıdır.

2.2.Simülasyona Ne Zaman Başvurulmalıdır?

- Problemin tam bir matematik formülasyonu mevcut değildir veya matematik formülasyonu analitik yöntemlerle çözümü henüz bulunamamıştır. (Kuyruk modellerinde olduğu gibi)
- Analitik yöntemler çözüm için elverişli ama çok karmaşıktır,
- Analitik çözümler vardır ve kullanılabilir, fakat problem üzerinde çalışanlarda bu bilgiler yoktur,
- Belirli parametrelerin tahmin edilmesi için de simülasyona başvurulabilir,
- Deneme için simülasyon tek yol olabilir,
- Sistemlerin davranış karakteristiklerini ortaya koymak zaman gerektirebilir. Bu durumlarda simülasyon modelini bilgisayara adapte ederek çalıştırmak ve sistem karakteristiğini kısa zamanda öğrenmek mümkündür.

2.3.Simülasyonun Kullanma Alanları

Simülasyon, üretim sistemlerinde, servis sistemlerinde, iletişim sistemlerinde, bir bilgisayar sisteminin donanım ve analizinin incelenmesinde, finansal ve ekonomik sistemlerin analizinde, dağıtım kanallarının tasarımında kullanılmaktadır. Bunları daha ayrıntılı olarak incelersek:

- 1) Üretim sistemlerinin simülasyonu,
- 2) Servis sistemlerinin simülasyonu.

2.3.1.Üretim sistemlerinin simülasyonu

Üretim sistemlerinin simülasyonu, birbirlerini etkileyen çok fazla alt birimin oluşturduğu ve matematiksel olarak çözülmeyen üretim sistemlerinin, bilgisayar üzerinde kurulup sistem dinamiğinin araştırıldığı ve değişikliklerin yaratacağı riskleri önlemeye yarayan bir sistem mühendisliği aracıdır. Üretim sistemlerinin belirsizliklerle bezenmiş olan yapıları onların kesin formüllerle incelenmesini önler.

2.3.2. Servis sistemlerinin simülasyonu

Servis sistemlerinde uygulanan simülasyonla müşterilerin bekleme zamanı en aza indirgenmeye çalışılmakta ve üretim sistemlerinde uygulanan alanların hepsi servis sistemlerinde de gerçekleştirilmektedir.

2.4.Simülasyonun Avantajları

- Simülasyon esnek bir çözüm yöntemidir,
- Klasik çözüm yöntemlerinin kullanılmadığı büyük ve karmaşık problemlerin çözümünde oldukça etkilidir,
- Bir başka yöntemle ele alınması olanaklı olmayan koşullar ve kısıtlar simülasyonla rahatça modellenebilir,
- Sonuçları ancak aylar yıllar sonra alınabilecek kararlar simülasyonla çok kısa sürede analiz edilebilir,
- What-if türü senaryoların modellenmesi ile yöneticilere değişik seçenekleri değerlendirme olanağı tanır,

- Simülasyon modellenen sistemi değiştirmeden yeni fikir ve politikaların model üzerinde rahatça uygulanmasına olanak verir,
- Mevcut sistemde önerilen veya mevcut olmayan sistemler analiz edilebilir,
- Pek çok faktörün etkisi aynı anda ve etkileşimli olarak denetlenebilir,
- Deney koşulları üzerinde kullanıcı tam bir kontrole sahiptir ,simülasyonu istenen zamanda başlanıp durdurulabilir.

2.5.Simülasyonun Dezavantajları

Şüphesiz yerinde doğru ve bilinçsiz kullanıldığı zaman simülasyon modellerinin bazı sakıncalarından da bahsedilebilir. Bunların bazıları şöyle sıralanabilir:

- İyi bir simülasyon modelini geliştirmek vakit alıcı ve pahalı olabilir,
- Optimum sonuçlar üretmez. Bir çeşit deneme yanılma yöntemidir. Yöneticiler denemek istedikleri alternatifler için tüm koşullar ve kısıtlar ortaya koymalıdır. Yoksa simülasyon kendi başına çözüm üretmez,
- Her simülasyon modeli kendine özgüdür. Genelde bir başka problemin çözümünde aynen kullanılamaz,
- Stokastik yapısı çok sayıda deneyi gerektiren istatistiksel analiz öngörür,
- Modellemede ve bulguların analizinde yapılacak hatalar yanlış anlamalara sebep olabilir

2.6.Simülasyon Süreci

Simülasyon çalışmalarının adımlarını aşağıdaki gibi ele alınabilir:

- Sistem tanımı ve problemin formülasyonu,
- Gerçek verilerin toplanması ve işlenmesi,

- Matematik modelin formülasyonu,
- Gerçek verilerden yararlanarak parametrelerin tahmini,
- Modelin test edilmesi (modelin geçerliliğinin araştırılması),
- Bilgisayar modelinin kurulması,
- Bilgisayar modelinin geçerliliği,
- Deneysel tasarım,
- Simülasyon verilerinin analizi.

Eğer bilgisayar simülasyonu, başka bir teknik uygulandığında oluşacak maliyetten daha düşük fiyata mal oluyorsa ve anlaşılır, gösterimi kolay çözümler elde ediliyorsa bu teknik seçilmelidir. Bu şartlar sağlanamıyorsa daha başka alternatifler aranmalıdır.

2.7.Simülasyon Dilleri

Piyasada birçok simülasyon yazılımı bulunmaktadır. Bunlardan hangisinin seçileceği, uygulama alanına göre değişecektir. Üretim alanındaki yazılımların hem grafik hem de büyük sistemleri modelleyebilme kapasiteleri oldukça büyüktür.

Bu tezde promodel paket programı kullanılmıştır.

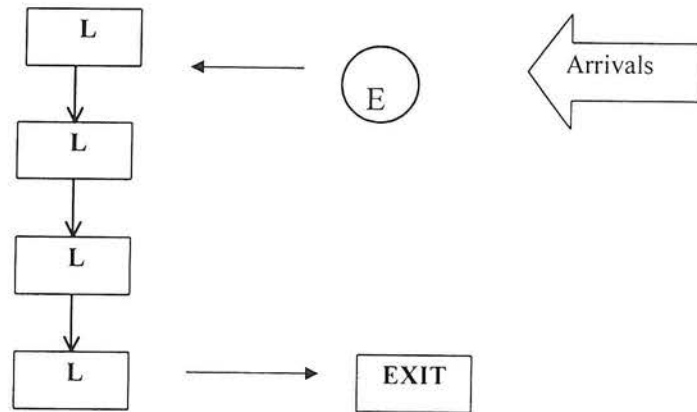
2.7.1 ProModel paket programı

Günümüzde ilerleyen teknolojiye bağlı olarak, simülasyon çalışmalarının gerçekleştirilmesinde bilgisayar kullanılmaya başlanmış ve ilerleyen zamana bağlı olarak da bu işi gerçekleştirecek paket programlar oluşturulmuştur. ProModel, imalat tabanlı üretim istemlerinin dizaynı için yazılmış bir paket programdır.

Promodel'in en önemli menüsü build menüsüdür. Modellenen bir programda olması gereken temel bileşenler şunlardır:

- 1) **Locations:** Parça veya servis elemanlarının gideceği ve işlem göreceği yeri gösterir. Bunlar imalat sistemlerinde, tezgahlar olarak tanımlanırken, servis sistemlerinde kaynaklar (bir hastane için yataklar, bir hava yolu için uçaklar gibi) olabilmektedir.
- 2) **Entities:** Sisteme giren varlıklardır. İmalat sistemlerinde işlem gören parçalar olarak tanımlanabilir.
- 3) **Processing:** Kısaca işlemler denilebilir. Hangi parçanın hangi tezgahta işleneceği, ne olarak çıkacağı ve nereye gideceği burada tanımlanır.
- 4) **Arrivals:** Gelişler demektir. Burada ise, sisteme dışarıdan gelen parçaların ne olacağı, nereye gideceği, frekansının ne olacağı ve simülasyonun kaç parça için yapılacağı tanımlanır.

Bu dört ana bölümü şekil üzerinde şu şöyle şematize edebiliriz:



Şekil 2.1.Promodel'de genel yapısı

2.7.2. Promodel 'da model oluřturma ařamaları (Temel elemanlar)

2.7.2.1.General information (Genel bilgiler)

Modeli kurmaya ilk bařladıđımızda General Information' da modelimizin adı, uygulama t r , kullandığımız zaman  l  s  ve kullanılan grafik t r  belirtilir.

2.7.2.2.Locations (Yerleřimler)

Modelin inřasında ilk adım sistemdeki Locationsların yerleřtirilmesidir. Bu locationslar tezgahlar, stok alanları,  alıřma istasyonları, depolar vb yerleřimlerdir.

Locations edit r     pevcere i erir ve bunlar mouse yardımıyla ulařılır:

- Location Edit Table,
- Location Graphic Window,
- Layout Table.
- **Location edit table (Yerleřim giriř tablosu)**

Yerleřim grafik penceresinden bizim modelimize uygun olan grafikler belirlenir. Klavye yardımıyla yerleřimin adını, kapasitesini, h cre tipini,  retim  rneklerinde her bir h cre i in birim yerleřim sayısını, yerleřim veya kaynaklar i in bozulma frekansını, s resini, ilk defa ne zaman olacađını, belirlenen yerleřime giriřlerin hangi

kurala göre gerçekleşeceğini buradan belirleriz.

- **Locations graphic windows (Yerleşim grafik penceresi)**

Grafik olarak yerleşimini oluşturduğumuz programın animasyon olarak görüntülendiğindeki grafikler bu pencereden mouse yardımıyla seçilir. Bu penceredeki EDİT tuşu yardımıyla grafiklerin ebatları rengi, açıları değiştirilebilir.

- **Layout Table**

Bu adımda ise Grafik Penceresinden seçilen grafikler mouse yardımı ile animasyonda nasıl görüntülenmesi isteniyorsa o alana yerleştirilir.

2.7.2.3.Pathnetwork (Ağ şebekesi)

Modelimizde hareket edecek olan parçanın yerleşim alanındaki tezgahlara gidiş geliş rotasının çizilmesi, bir başlangıç düğümü ve bitiş düğümleri vasıtasıyla layout üzerinde işaretlenir. Klavyeden ise bu şebekenin adı, düğümlerdeki kuyruk varlığı, hangi düğüme ait olduğu belirlenir.

2.7.2.4.Resources (Kaynaklar)

Oluşturulacak olan programdaki teçhizatın taşınması veya her şeyden önce girdilerin taşınması kişi veya makinelerle olabilir. Kaynaklar hareketli olabildikleri gibi sabitte olabilirler. Kaynaklar editörü iki pencere içerir:

- Resource Edit Table,
- Resource Graphic Window.
- **Resources edit table (Kaynaklar giriş tablosu)**

Grafik pencerelerinden seçilen kaynak grafikleri burada şekil olarak belirlenir. Klavye vasıtasıyla kaynağın ismi, kaynağın temsil ettiği birim sayısı, bozulmayla ilgili ayrıntılar, kaynağın proses esnasında izleyeceği yolu çalışma ve durma yollarını belirler.

- **Resources graphic window (Kaynaklar grafik penceresi)**

Animasyon esnasında görülecek kaynak grafikleri bu menüden seçilir.

2.7.2.5.Entities (Girdiler)

Modeldeki girdilerden kasıt ürünler ve insanlar gibi sistemdeki işlenen parçalardır. Girdiler tek bir girdi olarak gruplanabilir, iki veya daha fazla girdiye ayrılabilir. Bu editör iki pencereye sahiptir:

- Entities,
- Entities Graphics .
- **Entities (Hareketli parçalar)**

Modelde kullanılacak olan mouse ile seçilecek olan nesne bu editörde

şekillenecektir. Klavye yardımıyla adı, ağ şebekesi boyunca hareketinin belli bir zamana bağlı hızı, istatistiki bilgileri buradan girilir.

- **Entity graphics (Girdi grafikleri)**

Girdinin animasyonda nasıl görünmesi isteniyorsa uygun bir şekil buradan mouse yardımı ile seçilecektir. Girdiyle ilgili olarak ayrıca renk, büyüklük, görüntü düzenlemeleri de burada yapılır.

2.7.2.6.Processing (Prosesleme)

Modelin bu kısmında entitieslerin hangi tezgahta işlem göreceği, tezgaha girmeden önceki şekli, girdikten sonraki şekli, ne kadar zaman işlem göreceği, bu tezgahdan çıktıktan sonra hangi tezgaha gideceği gibi bilgilerin yer aldığı bölümdür. Prosesleme mantığı sistemdeki her yerleşime gelen her girdi için yolları ve operasyonları belirlemektir. Girdilerin sisteme ilk gelişleri gelişler editöründen belirlenir. Prosesleme editörü dört pencereden oluşur:

- Process Edit Table,
 - Routing Edit Table,
 - Tools Windows,
 - Layout Windows.
- **Process edit table (İşlem giriş tablosu)**

Process edit table sistemdeki her yerleşime gelen her girdi için operasyon mantığını belirler. İşlemlerin düzeni, artan operasyonlar için özellikle bir yerleşimde bir girdinin bulunmasını istediğimizde önemlidir. Bu girdinin hangi yerleşime gideceği ve bu yerleşimde göreceği operasyonun süresi ve mantığı buradan belirlenir.

- **Routing edit table (Yol giriş tablosu)**

Bu tablodan işlem giriş tablosunda belirlenen her operasyon kaydının çıktısı belirlenir. Şunu anlamak önemlidir ki, tüm proses kayıtları uygun bir proses yoluna sahip olma gereksinimi duymaz. Bir girdi aynı yerleşimde ard arda birden fazla işlem gördüğünde her işlem için farklı proses yolu belirlemeye gerek yoktur. Ayrıca proses yolunu takip ederken uyulacak kural ve proses yolu belirleme işlemleri buradan yapılır.

- **Tools windows (Araçlar penceresi)**

Bu pencere yardımı ile belirtilecek prosesleme ve prosesleme yolu belirleme işlemleri grafiksel olarak yapılabilir. Proseslenecek olan girdi bu pencereden, prosesleme de kullanılan yerleşimler layout üzerinden, mouse yardımıyla seçilir.

2.7.2.8.Arrival (Gelişler)

Bu kısım modeldeki girdinin sisteme gelişini ifade eder. Her geliş başına miktar, gelişlerin frekansı, hangi aralıkta geleceği ve hangi yerleşime geliş olacağı gibi bilgiler belirlenir.

Bu editör üç pencereye sahiptir:

- Arrivals Edit Table,
- Tools Windows,
- Layout Window.

- **Arrivals edit table (Gelişler giriş tablosu)**

Bu tablodan girdinin ve yerleşimin adı, her girdi başına geliş miktarları, ilk geliş zamanı, sistem süresince kaç gelişin olacağı, frekans ve geliş mantığı belirlenir.

- **Tools windows (Araçlar penceresi)**

Gelişler bu tablodan grafiksel olarak belirlenir. Bu tabloda beliren girdiler arasından mouse yardımıyla seçim yapılır. Seçilen girdi gelişler Giriş tablosunda belirir. Diğer bilgiler yine aynı tablodan klavye ile girilir.

Yukarıda anlatılan menüler yeni yapılan bir programda mutlaka olması gereken modüllerdir. Ancak istenirse daha fazla bilgileri girmek için diğer modüllerden de yararlanılabilir. İstenirse modelin diğer ayrıntıları şu seçeneklerden girilebilir:

- Variables (Değişkenler),
- Attributes (Davranışlar),
- Macros (Geniş Ölçekli Programlama),
- Files (Dosyalar),
- Streams (Süreklilik),

- Tables (Tablolar),
- Background Grafics (Arka Plan Grafikleri),
- Layout Options (Yerleşim Seçenekleri).

Model oluşturulurken bazı sınırlamalarla karşılaşabiliriz. Bunları şu şekilde sınırlayabiliriz:

2.8.Modelin Çalıştırılması

Yapacağımız model bu aşamalardan geçtikten sonra çalıştırılması oldukça kolaydır. Simülasyon menüsünden run komutu mouse yardımıyla seçilerek program çalıştırılabilir. Model çalışırken eğer model hata içeriyorsa, nerede hata yapıldığını ve hata tipini detaylı bir mesajla verecektir. Modelin tam olarak çalışıp çalışmadığını animasyon vasıtası ile hemen görülebilir

Simülasyon seçenekleri penceresinden ise çalışma süresi, hazırlık zamanı, tekrarlanma sayısı, operasyonun istatistiki sonuçlarının detaylı olup olmayacağı, zamanın hassaslığı belirlenir.

Animasyon devam ederken layout penceresi büyütülüp küçültülebilir. İstenirse sadece istenen belli bir alanı görülebilir ve durdurulabilir. Animasyon durdurulduktan sonra isteğe bağlı olarak programın istatistiki sonuçları alınabilir. Bu istatistiki sonuçlar; işlem zamanları, kaynak kullanımları, işlenen parça sayısı, atılan parça sayısı, her prodesteki işlem zamanları vb bilgileri kapsamaktadır.

BÖLÜM 3.YAPAY SİNİR AĞLARI

3.1.Giriş

Yapay sinir ağları teknolojisi bilgisayar dünyasında insan beynini ve sinir sisteminin davranışlarını taklit etme esası üzerine kurulmuş yeni bir bilgi işleme yaklaşımıdır. YSA bugün düşündüğümüz anlamda ne bir bilgisayardır ne de bir bilgisayar programıdır. Çok basit ve çok sayıda içten birbirine bağlı nöronlar diye adlandırılan işlemcilerden oluşmaktadır. Önceleri temel tıp bilimlerinde insan beynindeki nöronların matematiksel modelleme çabaları ile başlayan çalışmalar bugün fizik, matematik, elektrik ve bilgisayar mühendisliği gibi çok farklı bilim dallarında araştırma konusu haline gelmiştir. İnsan zekasına has görünen bazı kavramların sayısal olarak ifade edilebileceği ve böylece makinelerin insan zekasına şaşırtıcı derecede benzer yollarla öğrenme ve hatırlama işlerini yapabileceği görülmüştür.

3.2. YSA'nın Tanımı

YSA son derece paralellığe sahip, algoritmik ve sayısal olmayan bir bilgi işleme sistemidir, karmaşık ve belirsizlikler altında problemleri çözümler (Öz, (1998)). Bu teknoloji bilgisayar dünyasına insan beynini ve sinir sisteminin davranışlarını taklit etme esası üzerine kurulmuş yeni bir bilgi işleme yaklaşımı gelişmiştir (Kaya,(1998)).

Yapay sinir ağları yada kısaca YSA ; insan beyninin çalışma sisteminin yapay olarak benzetimi çabalarının bir sonucu olarak ortaya çıkmıştır. En genel anlamda bir YSA insan beynindeki birçok nöronun yada basit işlemcilerin birbirlerine değişik etki seviyeleri ile bağlanmasıyla oluşan karmaşık bir sistemdir.

YSA öğrenerek karar verme prensibi üzerine kurulmuştur. Öğrenme, zeki sistemlerin bilgi, yetenek ve tecrübelerini arttırma olayı olarak düşünülebilir. Farklı tanımlamalar yapılmakla beraber en genel şekliyle öğrenme “sistemlerin aynı veya benzeri işleri bir önceki yapıldıkları şeklinden daha verimli etkili olarak geliştirecek değişiklikleri oluşturma süreci” olarak tanımlanır.

3.3. YSA'nın Tarihçesi

YSA konusu, ilk olarak 1943 yılında, McCulloch ve Pitts tarafından ele alınmıştır. 1943 yılı YSA sistemleri için bir doğuş yılı olarak görülmüştür. McCulloch ve Pitts yaptıkları ilk YSA sistemi (1943), vakum tüplerinden ve çeşitli aletlerden oluşuyordu. Bu sistem genel olarak basit mantıksal işlemleri çözmede kullanıldı. Fakat sistem çok alan kaplıyor ve sistem yavaş çalışıyordu. Yine de bu iki araştırmacı YSA araştırmalarında ilk fonksiyonel nöronu yaparak yeni bir devir açtılar (Khanna, (1990)).

Bu tarihten sonra kısaca gelişmeler şu şekildeydi: bu on yılın içerisinde(1949), Donal Hebb, bilginin sinirler arası bağlantılarda saklanabileceğini keşfetti.1948'de Kohonen eğitici olmadan öğrenen ve 'Self Organizing Maps' ismini verdiği özel bir ağ geliştirmiştir. Daha sonra Rosenblatt (1958) nöron benzeri bir kavram ortaya attı. Bu kavram perceptron idi ve bu küçük sanal makine desenleri sınıflandırmayı, bağlantı ağırlıklarını değiştirerek öğrenebiliyordu. Bu gelişmeden sonra desen tanıma üzerine çalışmalar gittikçe hızlandı.

1960 yıllarının başında ADALINE (ADaptive LINEar combiner) denilen bir sistem icat edildi. Bu yeni icat Widrow-Hoff adında güçlü öğrenme kurallarına sahip idi ve Bernard Widrow-Hoff (1960-1962) tarafından bulunmuştu. Bu kural kısaca şöyle idi: Desen tanıma sırasında oluşan hata payları karesel bir orantı ile hesaplanıp sisteme geri gönderiliyordu. Böylece sistem tanıma işlemini daha doğru bir şekilde yeniden yapıyordu. Bu sistem daha sonraları geliştirilerek MADALINE (Many ADALINES) yapıldı. Bu alet, desen tanıma, hava durumu tahmininde ve uyumluluk kontrol sistemlerinde kullanıldı.

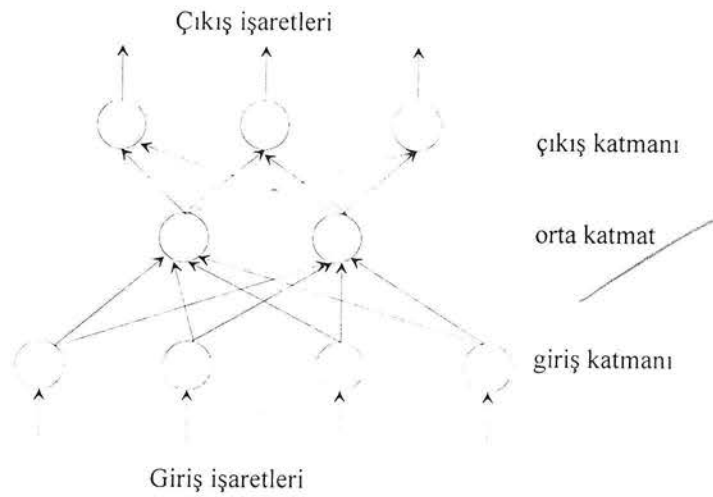
Bundan sonra arařtırmalar daha çok olayın formülize edilmesi ve yeni matematiksel formüller üretme çerçevesinde yapıldı. Amari (1972-1977), Fukushima (1980), Kohonen (1977-1982-1984-1988), Anderson (1977), ve Hopfield (1982-1984)YSA üzerine çalışmalar yaptılar. John Hopfield (1982-1984) hafıza ilişkilerini anlamak üzere bir ağ mimarisi geliřtirdi (Beale, (1990)). Bu geliřme YSA çalışmalarına hız verdi. Bařka bir geliřmede 1986 yılında McClelland ve Rumelhart tarafından yapıldı (Freeman,(1985)). Bu kural 1960-1980 arası çözülemeyen problemleri ortadan kaldırdı. 1990'larda artık YSA için özel makineler, sistemler, mikro-çipler, yazılımlar geliřtirildi, arařtırma grupları arttı, kullanım alanı son derece geniřledi.

3.4. Yapay Sinir Ağlarının Özellikleri

Yapay sinir ağları, insan sinir sisteminin biyolojik yapısından esinlenmiř bir yapıya sahiptir. Yani, yapay sinir ağları, insan sinir sistemindeki nöronların (Sinir hücresi) fonksiyonunu gören temel elemanlardan oluřmaktadır. Bu elemanlar insan beyninin anatomisine benzer řekilde organize edilmiřlerdir. Bu büyük benzemenin yanı sıra, yapay sinir ağları insan beyninin řaşırtıcı derecede bir çok özelliklerini taşırlar. Örneğın tecrübe ile öğrenirler, daha önce öğrenilen bilgileri genele indirgeyerek yeni sonuçlar çıkarırlar, yani bir bilgiden gereksiz kısımlarını çıkararak önemli olan özü alırlar. Fakat bütün bu fonksiyonel benzemelere rağmen, yapay sinir ağlarının yakın bir gelecekte insan beyninin fonksiyonlarını aynen yerine getirebileceğı söylenemez. Bugün en geliřmiř ve karmařık yapay sinir ağlarının gösterdiğı zeka düzeyi bir bağırsak kurdunun zeka düzeyini geçemez (Philip, (1990)). Fakat buna rağmen bazı yapay sinir ağlarının beyine yakın derecede gösterdiğı performans da göz ardı edilemez.

Yapay sinir ağlarında kullanılan nöronlar, beyinde bulunan analog biyolojik sinir hücrelerini modeli olarak düşünölmüřtür. Beyinde yaklaşık 10^{10} tane sinir hücresi vardır. Yine hücre başına bağlantı sayısı 10^4 mertebesindedir. Her nöron tipik olarak kendi bağlantılarından çeřitli řaretler alır. Bu řaretlerin bazıları diğır nöronlardan olduđu gibi bazıları da dıř dünyadan olabilir. Nöronlar birden fazla çıkıř řareti

üretirler. Bu işaretler nöron çok küçük bağlantı parçalarına ayrılan farklı uzaklıklarda son bulan kollardan oluşan çıkış bağlantısı üzerinden gönderilirler. Çıkış bağlantısının her kolu aynı işareti gönderir. İşaret kollar arasında bölünmez. Dışarı çıkan kolların büyük çoğunluğu ağdaki diğer nöronlardan gelen bazı kollarda son bulmaktadır. Diğerleri ise ağın dışında son bulmakta ve kontrol işareti veya cevap işareti üretmektedirler. Şekil 3.1 tipik olarak sinir ağının fiziksel bağlantı şeklini göstermektedir (Caudil, (1992)).



Şekil 3.1. Tipik olarak sinir ağının fiziksel bağlantı şekli

İnsan beyninin mimarisinden esinlenildiği için, bu bağlantı kuralları direkt olarak yapay sinir ağı bağlantılarında kullanılmaktadır. Sonuç olarak yapay sinir ağı nöronları biyolojik nöronların kabataslak benzerleri olduğundan biyolojik nöronların işlevini tam olarak göremezler.

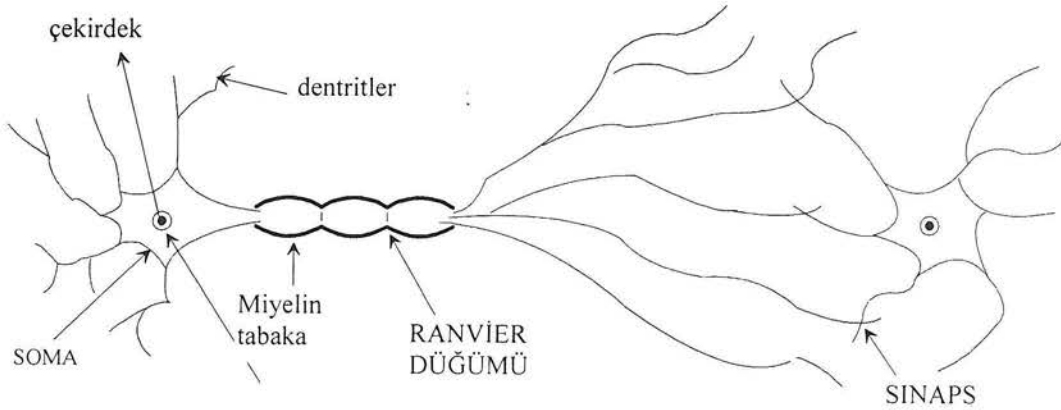
3.5. Biyolojik Sinir Hücresinin Yapısı ve Nöron Modeli

Yapay sinir ağları biyolojik nöronlardan esinlenerek düşünülmüştür. Araştırmacılar ağ şeklini ve algoritmasını düşünürken beyin organizasyonunu incelemişlerdir. Ancak beyin çalışma sistemi ile ilgili bilgiler çok sınırlıdır ve bu yöndeki çalışmalara yol gösterme olanağı çok kısıtlıdır. Bu bakımdan ağ tasarlayıcıları mevcut biyolojik

bilgilerden daha ileriye gitmeye, yararlı fonksiyon bulmaya yardımcı olacak kavramlar aramaya mecbur kalmışlardır.

Bu inceliğe rağmen, biyolojik olarak çok benzemese de , yapay sinir ağları ile beyin arasında yapılan karşılaştırmalarda benzerlikler ortaya konulabilmektedir. İnsan beyni üzerinde detaylı olarak çalışmalar yapılmış olmasına rağmen beynin çalışması hakkındaki bilgiler hala yetersizdir. Yapay sinir ağı fonksiyonları insanların algılamasını anımsatır. İnsan sinir sistemi nöron adı verilen hücrelerden oluşur. Beyinde olduğu tahmin edilen 100 milyar nöronun yaklaşık 1 katrilyon bağlayıcısı vardır.

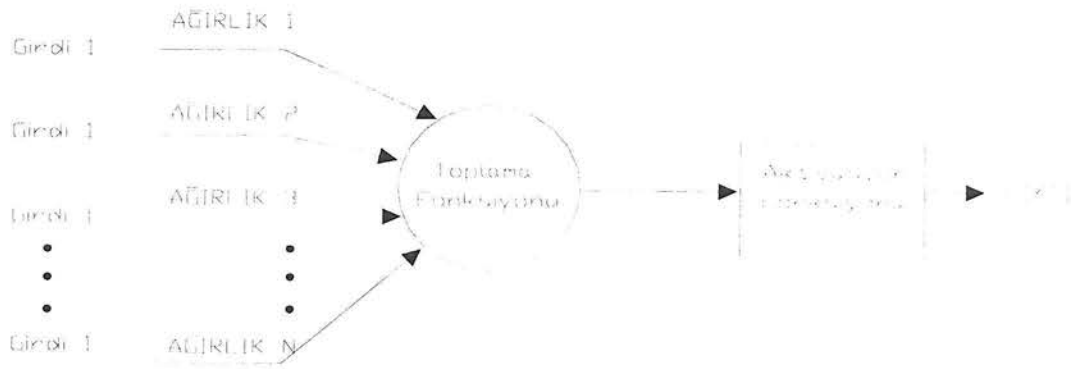
Bir biyolojik nöronun şematik yapısı Şekil 3.2 'de gösterilmiştir. Dentritler diğer hücrelerden elektriksel bilgi alan nöronun girişleridir. Aksonlar ise diğer hücrelere karşı darbelerin üretildiği elektriksel olarak aktif nöron çıkışlarıdır. Bir hücrenin aksonu ile diğer bir hücrenin dentritinin bağlantı noktasına sinaps denir.



Şekil 3.2. Bir nöronun şematik yapısı

Her akson diğer nöronlarla sinaps'lar vasıtasıyla bağlantı kurar. Sinaps'lar nöronun girişine diğer nöronlardan gelen işaretleri ağırlık katsayılarıyla çarpar daha sonra da toplamalarını alır. Eğer gelen işaretler toplamı yeterli seviyede ise (eşik seviyesini aşmış ise) aksonları üzerinden diğer nöronlara işaretler gönderir. Nöron bir t anında yalnızca bir işaret gönderebilir.

3.6. Yapay Sinir Modeli



Şekil 3.3. Yapay sinir modeli

Yapay nöron önceki bölümlerde anlatılan gerçek biyolojik nöronun birinci dereceden karakteristiklerini göstermek üzere tasarlanmıştır. Bu nörona her biri diğer nöronların çıkışlarını göstermek üzere bir girişler kümesi uygulanmıştır. Her bir giriş ise biyolojik nörondaki sinaptik dayanımla benzer olarak ağırlıklarla çarpılmıştır. Bütün bu ağırlıklı girişler ise nöronun aktivasyon düzeyini belirlemek için toparlanmıştır (Konar, (1996)). Bu fonksiyonel açıklamayı gerçekleyen model şekil 3.3.'de görülmektedir.

$$Y_{NET} = W_1 X_1 + W_2 X_2 + + W_n X_n \quad (3.1)$$

$$Y_{NET} = \sum_{j=1}^n W_j X_j$$

Ağ yapılarının farklı olmasına rağmen YSA'ların tamamına yakını yukarıdaki nöron tanımıyla işlemektedir. Burada $x_1, x_2, \dots, x_n$ nöron girişleridir. Herbir nöron girişi toplama işlemine girmeden önce $w_1, w_2, \dots, w_n$ ağırlıkları ile çarpılmaktadır. Toplam olarak girişler X vektörü ile ağırlıkları ise W vektörü ile gösterilebilir. Bu durumda 3.1. denklemin vektör notasyonu ile

$$Y_{NET} = W * X \quad (3.2)$$

şeklinde gösterilebilir. Daha sonra ok katmanlı almaç ve back propagation (geri yayılım) modeli incelendiğinde W vektör yerine bir matris olarak tanımlanacaktır (Konar, (1996), Khanna, (1990), Beale, Jackson, (1990)).

3.7. YSA'ların Özellikleri

YSA öğrenerek karar verme prensibi üzerine kurulmuştur. Öğrenme, sistemlerin aynı veya benzeri işlemleri yaptıklarında, o işleri bir önceki yapıldıkları şekilden daha verimli ve etkin olarak gerçekleştirerek değişiklikleri oluşturma süreci olarak tanımlanabilir. İnsan beynini taklit etme esası üzerine kurulan YSA, bir bilgi işleme yöntemi olup, birbirine paralel olarak bağlanmış işlem birimlerinde ve bu birimlerin hiyerarşik organizasyonundan oluşurlar (Freeman, (1992)).

YSA lineer olmayan sistemlerin kontrolü ve modellemesinde büyük kolaylıklar sağlamaktadır. Mühendislik problemlerinde ve birçok pratik uygulamalarda verimli bir kullanım için YSA'nın şu özelliklerini bilmek gerekmektedir.

3.7.1. Paralellik

Bilgi işlem yöntemlerinin çoğu ardışıl seri işlemlerden oluşmaktadır. Bu durum da beraberinde özellikle hız problemlerini getirmektedir. Bilgisayarlar insan beynine göre çok hızlı çalışmasına rağmen beynin toplam hızı bilgisayarlara göre kıyaslanamayacak kadar yüksektir. YSA paralel bir yapıya sahip olduğundan ağa gelen girişler, diğer katmanlar boyunca paralel olarak işleme sokulur. Aynı

katmanlardaki hücreler arasında zaman bağımlılığı yoktur. Böylece diğer yöntemlere göre işlem hızı arttığından daha kısa zamanda işlem sonucuna ulaşılır (Lippman, (1987)).

3.7.2. Hata toleransı

Bilgisayarda herhangi bir işlem elemanının yerinden çıkarılması veya değiştirilmesiyle bilgisayar iş yapamaz hale gelir. YSA birkaç elemanın olmaması toplam ağ performansında önemli bir azalma meydana getirmez. Çünkü YSA yapısında paralel çalışma bulunması ile sistemde bulunan elemanlar eksik elemanların işlerini yapabilecek hale gelir .

3.7.3. Gerçekleme kolaylığı

Yapay Sinir Ağları, karışık fonksiyonlar yerine basit işlemler içerdiği için gerçekleştirme işlemi diğerlerine göre daha kolaydır. YSA'nda her bir işlem birimi çözülecek problemin tümüyle ilgilenmek yerine yalnızca bir kısmıyla ilgilenmektedir. Hücrelerin çok basit işlemleri yapmalarına rağmen sağlanan görev paylaşımı nedeniyle çok karmaşık ve çok zor problemler daha kolay çözülebilmektedir (Khanna, (1990)).

3.7.4. Öğrenebilirlik ve adaptasyon

Veri işleme yöntemlerinin çoğu programlama yoluyla hesaplamaya dayanır. Bu yöntemlerle herhangi bir problemin çözümü için uygun bir algoritmanın geliştirilmesi zorunluluğu vardır ve ispatı mümkün olmayan durumların çözülmesinde kullanılamaz. Yapay sinir ağları nicel ve nitel verilerle işlem yapabilir ve bir sistemin çalışmasından alınan verilerle eğitilir. Bu eğitim sonucunda ağ sistemin çalışmasını öğrenir. Uygun bir şekilde eğitilen ağ, daha sonra kendisine sunulan ve eğitim setinde bulunmayan bir girişe de uygun cevabı verecektir. YSA'da problemler verilen örneklerle göre çözülür. Öğrenme, yapay sinir ağlarında ağırlıklarla olmaktadır. YSA bir çok girişi işleme sokarak bu girişlere uygun çıkışlar

üretebilme özelliğinden dolayı çok değişkenli sistemlere kolayca uygulanabilmektedir.

3.8. Yapay Sinir Ağlarının Temel Bileşenleri

Bir YSA modeli birbirinden bağımsız ve paralel olarak çalışabilen proses elemanlarının (yapay sinir hücrelerinin, nöronların) hiyerarşik bir şekilde organizasyonundan oluşur. YSA' yı oluşturan proses elemanlarının her biri beş temel parçadan oluşur (Gürgün, (1993), Khanna, (1990)).

- 1) Girdi elemanları,
- 2) Ağırlıklar,
- 3) Toplama fonksiyonu,
- 4) Aktivasyon fonksiyonu,
- 5) Çıktı elemanı.

Bu beş elemanı tek tek şu şekilde ele alabiliriz.

1) Girdi Elemanları : YSA' da girdi (input) bir dış kaynaktan gelebileceği gibi diğer bir sinirden de gelebilir. Bir nörona birden fazla girdi gelebilir. Nöral hesaplama sadece sayıları işler. Eğer problem sayılardan oluşmuyor ise problemin girdileri sayısallaştırılır. 0 ve 1 arasında temsil edilirler. Örneğin problem onaylama yada onaylamama olarak ifade ediliyorsa onaylama1 olarak onaylamama 0 olarak ifade edilir. Bunun gibi YSA' da girdi karakterlerinin veya grafiklerinin piksel değerleri de 0-1 arasında ağa verilir. Örneğin 5*10 piksellik bir karakter 50 bit vektör girişi ile ağa girdi olarak verilebilir.

2) Ağırlıklar : YSA' nın en önemli unsurlarındandır. Sinir hücresinde girdilerin etkisini tespit eder. Ağırlıklar bir proses elemanının her bir girdisinin önemini gösterir. Ağırlıklar değişken veya sabit olabilir. Ağırlık değerinin (-) olması etkinin ters yönde olduğunu gösterir. Ağırlıklar bir yerde YSA' nın hafızaları olarak ifade

edilebilir. Problemlerde ağırlıklarla oynayarak en uygun ağırlıklar bulunur. Uygun ağırlıkların bulunması problemin çözüldüğünü gösterir.

3) Toplama Fonksiyonu : Sinir hücresine ait net girdiyi açıklar. Toplama fonksiyonu her bir ağırlığı girdi ile çarpar, daha sonra bunları tek bir proses elemanı olacak şekilde toplar. Kısacası bir proses elemanından gelen bilgileri birleştirme işlevini yürütür. Toplama fonksiyonunda girdiler tek veya çok miktarda olabilirler. Bu yüzden tek bit terim halinde temsil edilme ihtiyacı duyarlar. Bunun için bazı fonksiyonlar kullanılır.

En çok kullanılan toplama fonksiyonu tipleri maksimum, minimum, çoğunluk, çarpım, toplam ve kümülatif toplam fonksiyonlarıdır. Tablo3.3.' de bu fonksiyonlar görülmektedir.

Tablo 3.1. En çok kullanılan toplama fonksiyonları

Toplam : $Net_i = \sum w_{ij} I_j$	Çarpım : $Net_i = \prod_J w_{ij} I_j$
Maksimum : $Net_i = \text{Max} (w_{ij} I_j)$	Minimum : $Net_i = \text{Min} (w_{ij} I_j)$
Çoğunluk : $Net_i = \sum_J \text{Sgn} (w_{ij} I_j)$	Küm. Top. : $Net_{\text{yeni}} = Net_{\text{eski}} + \sum_j w_{ij} I_j$
Burada i, j : Proses elemanı I_j : Proses elemanının çıktısı W_{ij} : i ve j proses elemanları arasındaki bağlantının ağırlığını gösterir Net_i : Proses elemanlarına giren net girdi	

4) Aktivasyon Fonksiyonu : Aktivasyon fonksiyonu toplama fonksiyonunun sonucunu alır, ilgili fonksiyonla bunu işler, proses elemanının nihai çıktısını üretir. Burada elde edilen çıktılar diğer proses elemanların yada dış dünyaya iletilir. Aktivasyon fonksiyonları her bir proses elemanının ürettiği çıktının (0, 1) aralığında olmasını sağlar. Bunun sebebi yapılan işlemlerde aşırı değerlerin üretilmesinin önlenmesi ve belli sınırlar içerisinde çalışmayı gerçekleştirmektir. Problemin yapısına göre aktivasyon ve toplama fonksiyonları tercih edilir.

5) Çıktı Fonksiyonu: Çıktılar ağların sonuçlarıdır. Her bir proses elemanı sadece bir çıktı üretebilir ama birden fazla girdiye sahip olabilir.

Tablo 3.2' de bazı aktivasyon fonksiyonu türleri verilmiştir.

Tablo3.2.Aktivasyon fonksiyonu türleri

Lineer Fonksiyon $f(x) = x$	Sinüsoidal Fonksiyon $f(x) = \sin(x)$
Sigmoidal Fonksiyon $f(x) = 1 / (1 + e^{-x})$	Hiperbolik Tanjant Fonksiyonu $F(x) = (e^x - e^{-x}) / (e^x + e^{-x})$
Adım Fonksiyonu $f(x) = \begin{cases} 1 & \text{eğer } x > \text{eşik değeri} \\ 0 & \text{diğer durumlarda} \end{cases}$	Eşit Mantıksal Fonksiyonu $f(x) = \begin{cases} 0 & \text{eğer } x \leq \text{eşik değeri} \\ x & \text{eğer } 0 < x < 1 \\ 1 & \text{eğer } x \geq 1 \end{cases}$

3.9. YSA'nın Bağlantı Elemanları

Sinir ağlarına ait hesaplamanın temel prensibi, kompleks davranışların daha basit parçaların kollektif davranışından meydana gelmiştir. YSA birimlerinin basit aktivasyon fonksiyonları, YSA' nın girdi ve çıktıları arasındaki kompleks ilişkinin basit parçalarıdır. YSA modellerindeki iç dinamiklerin asıl kaynağı, büyük bir değişime sebebiyet5 veren bağlantı elemanları ve nispeten küçük bir takım aktivasyon fonksiyonlarının varlığıdır (Beale, Jackson (.1990), Khanna, (1990)).

3.9.1.Bağlantı modelleri

YSA bağlayıcı modelinin en önemli safhası, dinamik davranışlara öncülük eden ve ağ içindeki aktivasyonları devam ettirebilen dairevi bağlantılara sahip olup olmadığıdır. YSA' da dairevi bağlayıcıların üç tipi vardır :

- Çift yöne ait bağlayıcı (örneğin BM' ler de ve Hopfield ağlarda)
- Otomatik ilmiçler (kendi kendine bağlantılar BM' de)
- Geri besleme bağlayıcısı (örneğin Jordan tarafından kullanılan süreli ağlar)

Dairevi bağlayıcı YSA' lar “geri beslemeli ağlar” olarak adlandırılır. Analiz edilmeleri zor olsa bile, girdi-çıktı ilişkilerinin türleri zengindir. Geçici modellerin oluşturulması için özellikle uygundur. Geri beslemeli ağların bir çok uygulaması bulunmaktadır. Bu ağlar konuşulan dili tanıma, sembolik sıraların tahmininde, doğal dil işleme de ve planlamada kullanılır.

Bir diğer önemli bağlayıcı model rekabet modelidir. Bir tabakadaki her bir birimden öteki tabakadaki her bir diğer birime negatif ağırlıklı bağlantılar ihtiva eder. Bir birimin harekete geçirilmesi diğer birimlerin aktivasyonunu engeller. Bütün

engelleyici bağlantılar aynı öneme sahip olduğu zaman, maksimum girdiyi alan birime “yarışı (rekabeti) kazanan” denilir.

Bazen, özellikler arsındaki ilişkilere bir öncelik tanınır. Örneğin, iki veya daha fazla girdinin ortak tasnifi önemli olduğu zaman,girdilerin değerleri yüksek derecede bağlantılar kullanarak birleştirilirler. Yüksek derecede bağlantılara sahip bir birim, girdilerin artan toplamından ziyade, sonuçların artan toplamına cevap verir: böyle birimleri kullanan YSA’ lar “ Yüksek Dereceli Sinir Ağları-(HONNs)” olarak adlandırılır.

Tablo 3.3’de bu bölümde tartışılan çeşitli bağlayıcı örnekleri gösterir. Çift yöne ait bağlantılar belirsiz uçlarla (yönü bilinmeyen uçlar) gösterilir.

Tablo 3.3. YSA' da çeşitli bağlantı modelleri

Bağlantı Modelleri	Grafiksel Gösterim
İleri Beslemeli Bağlantılar (Feedforward connections)	
İki Yönlü Bağlantılar (Bidirectional connections)	
Kendinden Bağlantılar (Self connections)	
Yinelemeli Bağlantılar (Recurrent connections)	
Yüksek Seviyede Bağlantılar (High-order connections)	

Tablo 3.3’de “yüksek dereceli bağlantılar”a) birkaç ucun bir araya gelmesi, girdiler üzerinde bir sonuç işlemini gösterir. Tablodaki ilk ve son çeşit hariç tüm ağlar geri besleme ağlarını gösterir. Tamamen birleştirilmiş birimlerin destek bir grup ihtiva eden YSA’ lar, optimizasyon ve birleşik bellek (associative memory) için sık sık kullanılır. Böyle YSA’ ların çift yönlü bağlantıları dinamik hareketle sonuçlanır (Freemann, Skapura, (1992)).

3.10.Yapay Sinir Ağlarında Bilgi İşleme, Öğrenme ve Zeki Davranış

YSA sahip olduğu bilgi proses elemanları arasındaki bağlantı hatları üzerinde saklanır ve ağırlıklar vasıtasıyla gösterilir. Ağ olaylar hakkında girdi ve çıktılar arasındaki ilişkiyi, elde bulunan mevcut örneklerden genellemeler yaparak öğrenir ve bu genellemeler ile yeni oluşan veya ortaya çıkan daha önce hiç görülmemiş olaylar hakkında karar verir. Yani ağa örnek bir model gösterildiğinde, girdi katmanından alınıp ara katmanlarda işletilerek ağın, o olay hakkında ürettiği sonuç çıktı katmanından sunulur. Bu bilgi işleme ağın sahip olduğu tecrübeye göre bilginin ara katmanlarda çağrıştırılması ile gerçekleşir. Bu çağrıştırma olayı modelden modele değişmektedir. Mesela ara katmanlardaki proses elemanları sahip oldukları bağlantı ağırlıkları ile kendi kararlarını üretir ve çıktı katmanındaki proses elemanlarına gönderir. Çıktı katmanındaki proses elemanları da yine ilgili ağırlıkları kullanarak ağın en son kararını oluştururlar. Bu ağırlıkları tıpkı ilgili olayın belirli özelliklerini hafızada saklayan elemanlar gibi düşünülebilir. Bilgi işleme ise bir olay gösterildiğinde hafızadan ilgili özellikleri çağırmak ve bunlar ile ilgili girdileri birlikte analiz ederek karar vermek şeklinde yorumlanabilir.

Ağın zeki davranış (öğrenme) gösterebilmesi için sahip olduğu bütün ağırlıkların ilgili problemde öğrenilmesi istenen özellikleri genelleştirecek şekilde optimum değerlere sahip olması gerekir. Bu doğruluk ne kadar artarsa ağın genelleme kabiliyeti dolayısıyla zeki davranış o kadar artar. Optimum ağırlık değerleri, bir öğrenme kuralına göre tespit edilir. Öğrenme kuralları ile proses elemanının örnekleri gördükçe kazandığı tecrübeye göre ilgili bağlantı ağırlıklarını nasıl değiştireceğini belirleyen algoritmalarıdır.

YSA' da öğrenme üç ana başlık altında incelenebilir (Will, (1988)):

- Eğitici öğrenme,
- Takviyeli öğrenme,
- Eğitici öğrenme.

3.10.1. Eğitici öğrenme

Eğitici öğrenme yönteminde hedeflenen sonuçları bilen bir eğitici, ağın ürettiği yanıtın doğru olup olmadığını kontrol eder. Ağa bir girdi-çıkı kümesi verilir ve istenen çıktılar alınana kadar ağa müdahalelerle sonuca ulaşılır. Ağırlıklar bir öğrenme kuralı ile ayarlanarak, girdi-çıkı arasındaki hata en aza, mümkünse sıfıra indirgenmeye çalışır. Bağlantı ağırlık katsayılarındaki değişim, geri besleme olarak gelen hatayı azaltacak ve hatta mümkünse sıfırlayacak şekilde ayarlanır. Bunun için çeşitli öğrenme kuralları kullanılır.

Eğitici öğrenme basit anlamda öğrenciler ile öğretmenin bir sınıftaki dersine benzetilebilir. Öğrenciler ve öğretmen ders boyunca karşılıklı etkileşim içerisindeyler. Öğrencilerin sorulan sorulara verdikleri cevaplar öğretmen tarafından denetlenmektedir ve gerekirse doğru cevap öğretmen tarafından öğrencilere iletilmektedir.

3.10.2. Takviyeli öğrenme

Takviyeli öğrenme yönteminde de YSA, bir öğreticiye gereksinim duyar. Bununla birlikte, bu stratejinin farklılığı çıktının istenilen çıktı olup olmadığı hususunda ağa öğretici tarafından müdahalede bulunulmasıdır. Bu tür öğrenme, girdi modellerine karşılık istenen çıktı modellerinin oluşturulmasının zor olduğu durumlarda etkilidir.

3.10.3. Eğiticişiz öğrenme

Eğiticişiz öğrenme yada kendi kendine öğrenmede, herhangi bir eğiticiden ağın ürettiği yanıtın doğruluğuna dair herhangi bir geri besleme gelmemektedir. YSA' nın bağlantı ağırlık katsayıları, oluşan hatayı göz önüne almaksızın daha önce belirtilen bir amaç ölçütü uyarınca, sadece girişlere bağlı olarak değişmektedirler. Bu tür bir öğrenme her YSA için yoktur ve her probleme uygulanamaz.

3.11.Yapay Sinir Ağlarında Ağırlık Uzayı

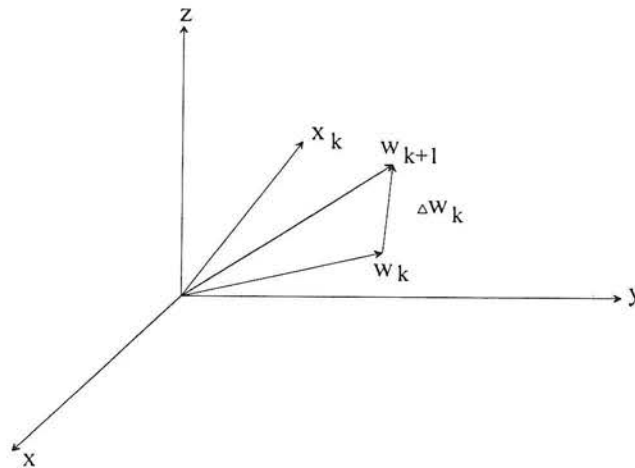
Her bir işlem elemanının "n" adet gerçek ağırlığı ve "N" adet işlem elemanı (nöronu) olduğu göz önüne alınırsa;

$$W = (W_{11}, W_{12}, \dots, W_{1n}, W_{21}, \dots, W_{2n}, \dots, W_{N1}, W_{N2}, \dots, W_{Nn})^T$$

$$W = (W_1^T, W_2^T, \dots, W_n^T)$$

Burada; $W_1, W_2, \dots, W_N$: İşlem elemanlarının (nöronların) ağırlık vektörleridir.

Şekil 3.4.'da eğitim süresince ağırlıkların düzeltiliminin vektörel çizimi verilmiştir.



Şekil 3.4. Ağırlık vektörünün değişimi

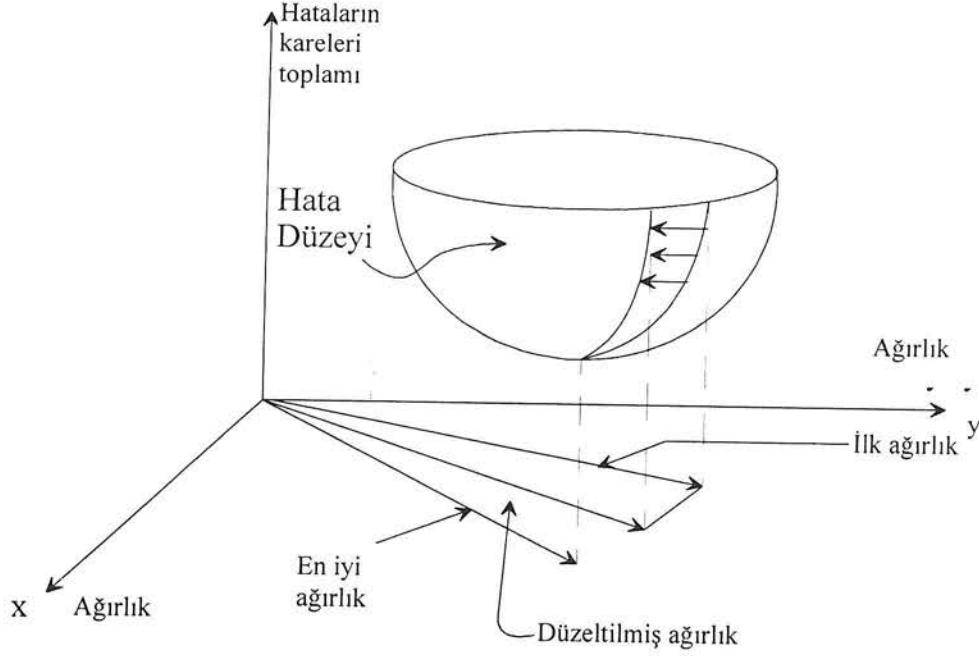
- X_k : Giriş işareti vektörü
 W_{k+1} : Gelecekteki ağırlık vektörü
 ΔW_k : Ağırlık vektörünün değişimi
 W_k : Şimdiki ağırlık vektörüdür.

Ağırlık vektörü ile çalışan yapay nöral ağ 'da önemli noktalardan birisi, bir öğrenme kuralı geliştirip, ağırlık vektörü "W" yı istenilen ağ performansı verecek noktaya yöneltmektir. Genellikle öğrenme kuralı için bir performans veya maliyet fonksiyonu tanımlanır. Ağırlık değiştirme denklemi, ağ 'daki hataların karelerinin toplamını minimize edecek şekilde düzenlenir.

3.12. Hata Minimizasyonu Ve Gradient Descent Tanımı

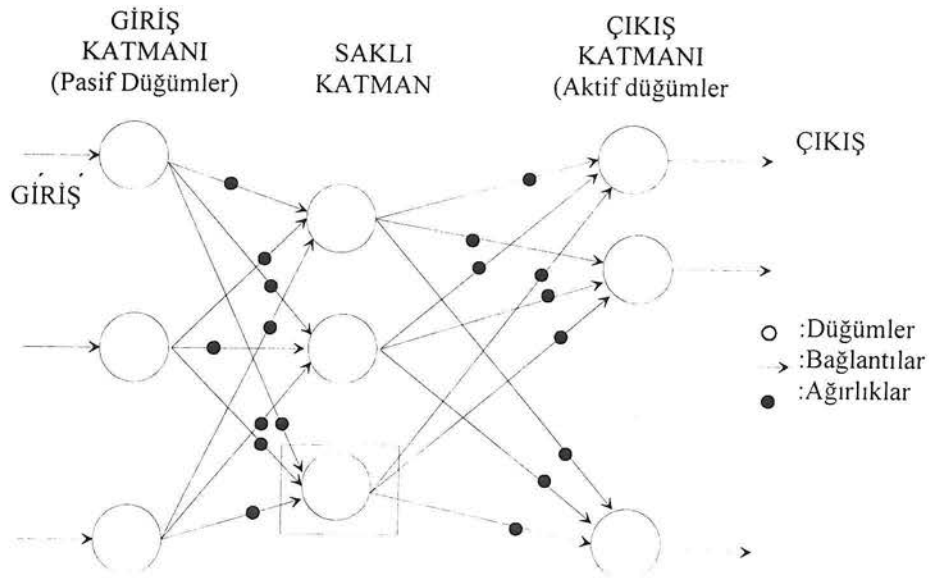
Hata minimizasyonun geometrik yorumunu yapmak mümkündür. Bunu görebilmek için, ağırlıkları mümkün olan tüm değerleri hataların kareleri toplamına karşılık gelecek şekilde (xyz) koordinat sistemine çizilir. Bu çizim sonunda hata yüzeyi Şekil 3.6 'de görüldüğü gibi küresel bir topa benzer. Bu yüzeyi bir tasa benzetmekte mümkündür. Tasın en alt kısmı hataların kareleri toplamının en küçüklerine karşı gelmektedir. Eğitim sırasında amaç, ağırlıklar kümesinin en iyisini olan en alt kısmını bulmaktır.

İleride ele alınacak olan yapay nöral ağ modellerinden Back Propagation (geri yayılma) algortiması, o anki ağırlıklar yerine yüzey hatasının eğimini hesaplayarak amacına ulaşır.



Şekil 3.5. Hata minimizasyonu ve "Gradient Descent" geometrik yorumu

değiştirir. İşte bu artımsal olarak tasın üst kısmından alt kısmına doğru ilerleme işlemine, Gradient Descent denir. Bir fonksiyonun Gradient 'ı (burada, hata fonksiyonudur ve değişkenler ise ağ ağırlıklarıdır) fonksiyonun en hızlı arttığı doğrultuyu verir. Gradient eksi ise fonksiyonun en hızlı azalma yönünü verir.



Şekil 3.6. Üç katmanlı yapay nöral ağı

Yapay nöral ağı 'lar paralel dağılmış parametrelili sistemler olduğundan her bir işleme elemanı(kare içine alınan) izole edilmiş ada gibi düşünülebilir. Şekildeki ağı 'da olduğu gibi herhangi bir işlem elemanının zarar görmesi sistem performansını düşürür. Fakat hiçbir zaman sistemi durma noktasına getirmez. Nedeni ise daha öncede belirtildiğı gibi bilginin tek bir yerde saklanmayıp, yerel belleklere dağıtılmasıdır. Bunun aksine klasik hesap sistemlerinde, az bir zarar dahi sistemi durma noktasına getirir.

3.13. Temel Bazı YSA Modelleri

Bu bölümde bir çok uygulamaya temel teşkil eden önemli YSA modelleri hakkında bilgi verilmiştir. Modellerin kullanım alanları belirtilmiştir.

3.13.1. Hopfield YSA modeli

Hopfield YSA modelinin tasarımcısı California Teknoloji Enstitüsü'nde John Hopfield' dir. Hopfield YSA'nı basit yapısı ve matematiksel tutarlığı nedeniyle yapay sinir sistemleri çalışmalarında önemli yeri olan ve tercih edilen bir modeldir.

Her probleme uygulanmamasına rağmen, daha karmaşık modelleri anlamak için mükemmel bir giriş bilgisi sağlar (Lippman, 1987).

Hopfield ağı tek katmanlı ve geri beslemeli bir YSA' dır. Genellikle giriş değerlerinin ikili düzeyde (0 ve 1) temsil edebileceği uygulamalarda kullanılır. Örnek olarak siyah-beyaz görüntülerin işlenmesi, 8 bitlik ASCII karakterleri ile temsil edilen metinlerin işlenmesi verilebilir.

Sürekli Hopfield Ağı modeli optimizasyon problemlerinin çözümünde başarılı uygulamalar verir. Örnek olarak gezin-satıcı problemi (Travelling Salesman Problem) verilebilir. Hopfield modeli YSA' nı pek çok optimizasyon problemine uygulamak mümkündür. Yapısında bir enerji fonksiyonunu minimize etmek özelliği bulunduğu için gerçek dünyadaki pek çok probleme uygunluk göstermektedir.

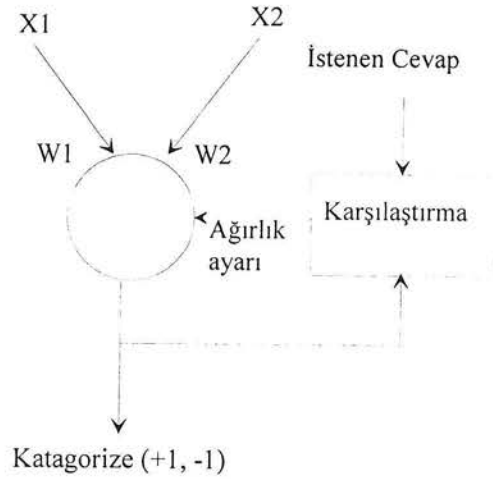
Hopfield modeli, yoğun bağlantılı, geri beslemeli ve bunun sonucu olarak dinamik bir yapıya sahip olduğu için gerçek biyolojik sistemlere en çok uyan modellerden biridir. Hopfield modeli, gezgin satıcı gibi zor optimizasyon problemlerinin çözümünde bile kullanılabilecek bir yapay sinir ağıdır. Hopfield modelinde donanım tavlama yönteminin kullanılması ile global minimum bulunması sağlanabilir. Hopfield modelini çeşitli uygulamalarda kullanmada en önemli unsur, devre parametrelerinin belirlenmesidir.

Problemin enerji fonksiyonu olarak iyi ifadesi alınır ve parametreler uygun seçilirse devre problemi için geçerli ve iyi çözümler bulur. Hopfield modelini herhangi bir probleme uygulamada kullanabilmek için problemin çözümü olacak şekilde devre parametrelerinin belirlenmesini sağlayacak bir koşul bulunması üzerinde çalışılması gerekmektedir

3.13.2. Adaline ve madeline

ADALINE ve MADALINE (Multiple – ADALİNE) Bernard WIDROW tarafından Stanford Üniversitesi'nde geliştirilmiş birer özel filtredir. Analog filtreler iletişim sinyallerinden sesi gidermek için tasarlanırlar. ADLINE YSA tek bir işlem elemanından oluşmuştur. Adını önceleri Adaptive Linear Neuron daha sonra ise Adaptive Linear Element ifadelerinden alınmıştır. ADALINE, yapı olarak Perceptrona çok benzer (Zurada, 1991).

Adaline basit iki kutup çıkışlıdır. Eğer girişin net ağırlıklı toplamı 0'dan büyük ise çıkış +1 ve eğer net ağırlıklı giriş toplamı 0 'dan küçükse çıkış -1 dir. Tipik olarak +1 çıkış A ve -1 çıkış B olarak kategorize edilebilir. Aşağıdaki şekilde iki giriş vektörü olan basit bir Adaline gösterilmiştir.



Şekil 3.7. Basit adaline

Şekil 3.7. de görüldüğü üzere, Adaline gerçek çıkış ile istenen değeri karşılaştırmaktadır. Bu bilgi ile ağırlık ayarları yapılır. Adaline hatası,

$$\text{hata} = < \text{istenen değer} > - < \text{gerçek çıkış} >$$

dır. Hata bulunduktan sonra ağırlık ayarları delta kuralı öğrenme yöntemi ile yapılır. Delta kuralı, ağırlıklardaki değişimleri:

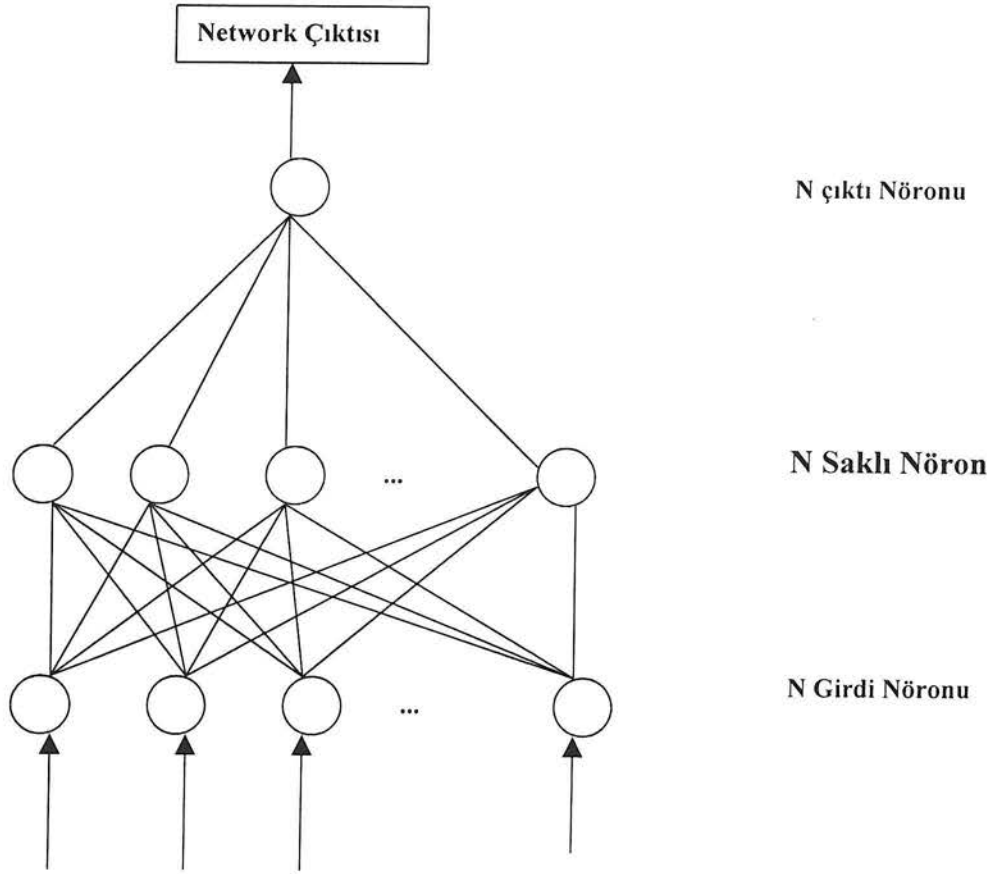
$$W = W^{\text{eski}} + \frac{\beta Ex}{|x|^2}$$

ifadesi ile hesaplar.

Bu ifade de β 0 ve 1 arasında sabit bir öğrenme katsayısıdır. E hesap edilen hata değeri, x ise (X_1 ve X_2) bileşenlerinden oluşan giriş vektörü, W 'da (W_1 , W_2) bileşenlerinden oluşan ağırlık vektörleridir (Freeman, Skapura, 1992), Lippman, 1987).

3.13.3. Çok katmanlı algılayıcı (ÇKA) ve backpropagation YSA

Çok katmanlı algılayıcı giriş ve çıkış düğümleri arasında bir veya birden fazla katman olan ileri beslemeli bir YSA' dır. Ara katmanlardaki saklı işlem elemanları giriş ve çıkış katmanlarındaki işlem elemanlarına doğrudan bağlantılı değildir. Çok katmanlı Algılayıcı, ilk geliştirildiği yıllarda etkin bir eğitime algoritması olarak yoğun bir şekilde çok kullanılmıştır. En iyi eğitime algoritmalarından biri adı geçen Backpropagation (geri yayma)' dır. Çok katmanlı algılayıcı yapısındaki bir ağa bu algoritmanın uygulanması ile Backpropagation YSA geliştirilmiştir. Backpropagation YSA karmaşık modellerin teşhisinde başarı ile kullanılmaktadır. Karmaşık bir modeli teşhis edebilmek için modeldeki tüm noktaların paralel olarak incelenmesi gerekir. Böyle bir incelemeyi yapabilecek sistem kolayca programlanamaz. Eğer sisteme birden fazla örnek model tanıtılıp bunlar arasındaki ilişkiyi öğrenmek için kendini uyarlayabilmesi sağlanırsa aynı ilişkiyi yeni modeller uygulayacak olan sistem bu zor problemi çözümleyebilir. İşte Backpropagation YSA böyle bir sistemidir. Geriye yayılım algoritmalarında saklı katman sayısının artırılması öğrenme oranını genelde yükseltir.



Şekil 3.8. Çok katmanlı algılayıcı (ÇKA) YSA' nın yapısı

Geriye yayılım algoritmasında YSA' nın eğitilmesindeki amaç, giriş değerleri ile, ağın üreteceği cevabın istenilen çıkış vektörünün aynısını veya istenen hassasiyetteki yakın değerini sağlayacak bir ağırlıklar kümesi bulmaktır. İstenen ağırlıkların bulunması problemin çözümü manasına gelir (Lippman, 1987).

Backpropagation YSA ileri beslemeli topolojisine sahiptir ve eğitimli öğrenme biçimini kullanır. İlk defa P.Werbos, daha sonra da D.B.Parker ve Rumelhart ile McClelland tarafından resmen kullanılmıştır. ÇKA YSA modeli kullanılarak daha önce özellikle kontrol kartlarının tanınması ve böylece prosesin otomatik olarak

denetiminin sağlanması için bir takım çalışmalar yapılmıştır (Lippman, 1987), (Lisboa, 1992).

Bu çalışmada ise bütün ağ modelleri değil, sadece tezde kullanılan "geriye yayılım" (backpropagation) modeli incelenecektir.

Geriye yayılım ağ 'ı, Adaline öğrenme yönteminin bir varyasyonudur. Adaline, en küçük kareler toplamı veya delta kuralı olarak adlandırılan öğrenme kuralını kullanır. Geriye yayılım ağ 'ı, delta kuralını çok katlı ağ 'a göre uyarlar. Bu uyarlamaya genelleştirilmiş delta kuralı denir.

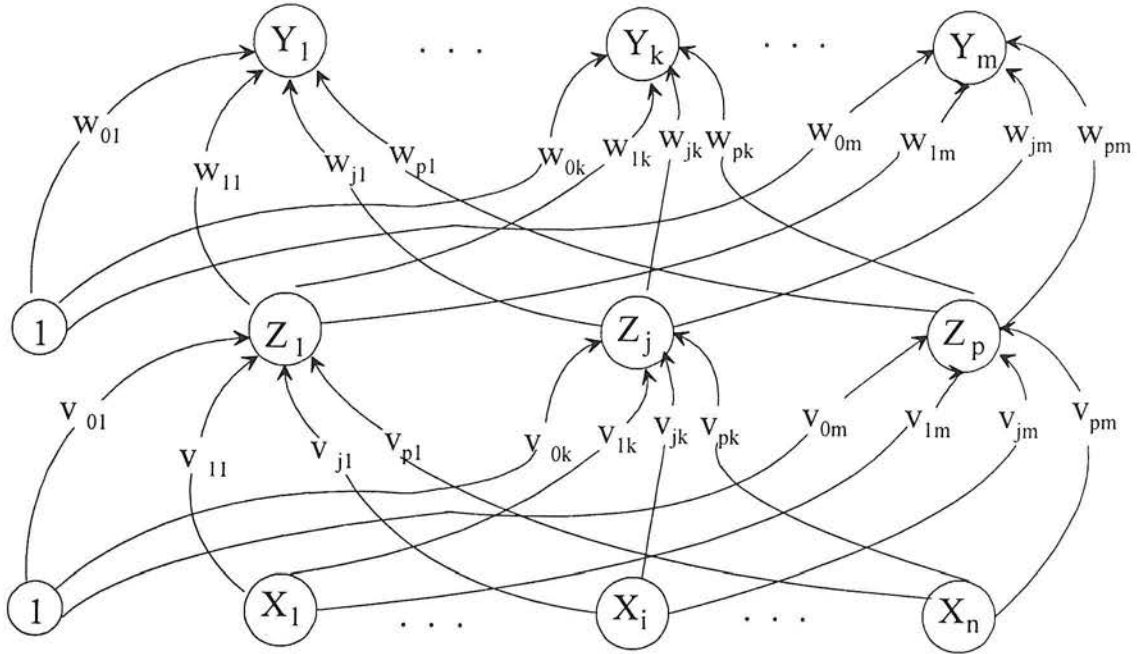
3.13.3.1. Geriye yayılım algoritması

Geriye Yayılım Algoritmasının (GYA) (Backpropagation-BP) bulunması YSA'ya olan ilgiyi tekrar canlandırmıştır. GYA çok katmanlı YSA'lar için sistematik bir metottur ve çok güçlü bir matematik esasa sahiptir. Bunun yanında her türlü kısıta rağmen GYA algoritması YSA'ların uygulanabildiği problem aralığının genişlemesine yol açmıştır ve çok sayıda başarılı ve güçlü uygulamaların üretilmesine imkan vermiştir.

GYA ilginç bir tarihsel gelişimi vardır. Rumelhart, Hinton ve Williams 1986 yılında GYA açık ve net bir biçimde ortaya koymuşlardır. Bu araştırmaya ilk başlayanlardan birisi olarak Parker görülmektedir. Fakat bu metodu ilk olarak ortaya atanlardan birisi ise Werbos' du ve 1974 yılında metodu ortaya koymuştu. Rumelhart ve Parker, Werbos'un bu çalışmasını genişletmek için çaba harcamışlar ve böylece metodu ortaya koymuşlardır. YSA'lar disiplinler arası olması, her yere kolayca adapte edilebilmesi sebebiyle çok çabuk gelişmiş ve konuyla ilgili kayda değer çalışmalarda bulunulmuştur bu ise konunun gelişimini otomatik olarak hızlandırmıştır.

Diğer ağ 'larda olduğu gibi Geriye Yayılım Ağ 'ını nöronlar arası bağlantılar, nöronların kullandığı transfer fonksiyonları ve ağ 'ın eğitimi sağlayan ağırlıkları

değişim kuralları belirler. Geriye Yayınım Ağ 'ı çok katmanlı ileri yön bağlamalı bir ağıdır. Katmanlardaki nöronlar arasında ve bir katmandan önceki katmana geriye doğru bağlantı yoktur. Geriye yayınım ağ 'ı, karesi alınmış hata fonksiyonunu minimize eden kodlu bir algoritma olup ve genelleştirilmiş delta kuralını eğitmek için kullanılır. Geriye yayınım ağ yapısı şekil 3.9 'de verilmiştir.



Şekil 3.9 Geriye yayınım ağ yapısı

Veri girişleri, giriş düğümleri $X_1 \dots X_i \dots X_n$ 'den yapılır. Giriş düğümleri pasiftir ve hesaplanabilir değildir. Her nöron için ayrıca bir eşik değer girişi mevcuttur. Bu girişler nöronun Bias'ı ya da referans seviyesi olarak adlandırılır. Bütün saklı nöronlar tüm veri girişlerini alırlar. Fakat her biri farklı ağırlık kümesine sahip oldukları için, değerler kümeside farklıdır. Her bir nöron kendisine gelen girişleri işleyerek sonucu çıkış nöronlarına iletirler. Çıkışlar da farklı ağırlık kümesine sahiptir.

Geriye yayınım ağ 'ında eğitim üç safhada gerçekleşir. Giriş eğitim örneklerinin ileri yayılması, hesaplama ve ilgili hataların geriye yayını ve ağırlıkların

ayarlanmasıdır. Eğitimden sonra, yalnızca ileri yayılım fazı kullanılır. Eğitim çok yavaş olsa bile, eğitilmiş ağ çok hızlı çıkış üretebilir. Birden fazla gizli katman bazı uygulamalarda faydalı olabilir, fakat bir gizli katman yeterlidir.

Geriye yayılım ağ 'ında çıkış katmanını eğitmek bağıl olarak basittir. Sorun hiç bir hedef vektörünün olmadığı ara katman ağırlıklarını eğitirken ortaya çıkar. Bunun çözümü, hata işaretinin çıkıştan daha iç katmanlara doğru yayılımı sağlamakla bulunur. Aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır.

Geriye yayılım ağ 'ının eğitme algoritması ana hatlarıyla şöyledir.

Adım 0 : Tüm ağırlıkların ilk değerlerini küçük rastlantısal değerlere ayarla.

Adım 1 : Döngü şartları sağlanmıyorsa adım 2 - 9 tekrar et.

Adım 2 : Her eğitim çiftleri için Adım 3 - 8 tekrar et.

İleri besleme:

Adım 3 : Her giriş birimi (X_i , $i = 1, \dots, n$) giriş işaretleri X_i 'leri al ve bir üst katman birimlerine ilet (Gizli katman 'a)

Adım 4 : Her gizli birim (Z_j ; $j = 1, \dots, p$) ağırlıklı girişlerini topla.

$$Z_{-inj} = V_{oj} + \sum_{i=1}^n X_i V_{ij}$$

z_j çıkış işaretini hesap etmek için aktivasyon fonksiyonuna uygula,

$$z_j = f(Z_{-inj})$$

ve bu işaretleri bir üst katmandaki çıkış birimlerine gönder.

Adım 5 : Her çıkış birimi (Y_k , $k = 1, \dots, m$) ağırlıklı girişlerini topla.

$$Y_{-ink} = w_{ok} + \sum_{j=1}^p Z_j W_{jk}$$

Y_k çıkış işaretlerini hesaplamak için aktivasyon fonksiyonuna uygula,

$$y_k = f(y_{-ink})$$

Hatanın geriye yayılımı:

Adım 6 : Her çıkış birimi için (Y_k , $k = 1, \dots, m$) kendi çıkışı y_k ile istenen çıkış t_k 'dan hata bilgisi

$$\delta_k = (t_k - y_k) f'(y_{-ink})$$

hesaplanır. Daha sonra W_{jk} 'lar güncelleştirilmek üzere ağırlık düzeltme katsayısı

$$\Delta W_{jk}(t+1) = \alpha \delta_k z_j + \mu \Delta W_{jk}(t), \mu, \text{pozitif küçük bir sayıdır.}$$

ve bias düzeltme katsayısı

$$\Delta W_{ok} = \alpha \delta_k$$

hesaplanır ve ilk alt katmana gönderilir

Adım 7 : Her gizli birim (Z_j , $j=1,\dots,p$) bir üst katman birimlerinden delta girişlerini toplar,

$$\delta_{-inj} = \sum_{k=1}^m \delta_k W_{jk}$$

hata bilgisinin hesabı için aktivasyon fonksiyonunun türeviyle çarpılır.

$$\delta_j = \delta_{-nj} f'(Z_{-inj})$$

Daha sonra V_{ij} 'ler güncelleştirilmek üzere ağırlık düzeltme katsayısı

$$\Delta V_{ij}(t+1) = \alpha \delta_j X_i + \mu \Delta V_{ij}(t) \quad \mu, \text{ pozitif küçük bir sayıdır.}$$

ve bias düzeltme katsayıları

$$\Delta V_{0j} = \alpha \delta_j \quad \text{hesaplanır.}$$

Ağırlık ve Bias 'ların güncelleştirilmesi

Adım 8 : $W_{jk}(t+1) = W_{jk}(t) + \Delta W_{jk}(t+1)$

$$V_{jj}(t+1) = V_{jj}(t) + \Delta V_{jj}(t+1)$$

Adım 9 : Stop durumunu test et.

Geriye yayılım öğrenme algoritmasının en büyük özelliği görüldüğü gibi, hatalara karşı verdiği cevapta, ağırlıklarının değerini değiştirebilmesidir. Ağ çıkış hatalarının türevi çıkış katmanından geriye doğru saklı katmanlara iletilir. Geriye yayılım algoritmasına ismini veren de hataların geriye doğru yayılması özelliği olmuştur.

Geriye yayılım algoritmasının matematiksel temeli "Gradient descent" olarak bilinen optimizasyon tekniğidir. Bu sebepten geriye yayılım algoritması hataların kareleri toplamını minimize eden ağırlıkları bulmada oldukça etkilidir. Geriye yayılım ağ 'ında eğitim yavaştır. Eğer eğitim bir seferlik bir ürün geliştirme problemi olarak görülürse, eğitim süresi ciddi bir problem olmaz. Geriye yayılım ağ 'ı yeni bilgiler öğretildiğinde eski bilgileri unutma eğilimi gösterir. Böyle bir davranış, ağ 'ın sürekli olarak değişen bir çevreye uyum sağlamasının beklendiği yerlerde uygun olmaz. Eğer eğitime esnasında eğitime kümesi tamamen biliniyorsa ve çevreye sürekli

adaptasyon gereksiz ise bu sorun gözardı edilebilir. Bu tez çalışmamızda, klasik kontrolle gerçekleştirilen sistemde, kontrolör ve lineer olmayan denklem çözüm blokları bir seferlik geliştirileceğinden ve eğitime esnasında eğitime kümeleri tamamen bilindiğinden ve çevreye sürekli bir adaptasyon gerekmediğinden yukarıda sözü edilen sakıncalar önem teşkil etmemektedir.

Bir çok araştırmacı temel geriye yayılım algoritmasını genişletmeyi ve ıslah etmeyi düşünmüştür. Literatür bu konuda oldukça geniştir. Parker 1987' de geriye yayılım algoritmasının yaklaşım hızını iyileştirmek için bir metot belirlemiştir, Bu metot literatürde Second-order Backpropagation olarak geçmektedir. Bu teknik doğru ağırlık değişimlerinin daha mükemmel bir tahminini üretmek için ikinci türevi kullanır. İkinci türevin dışında alınacak türevler tahmini iyileştiremeyecektir. Stornette ve Huberman (1987) geriye yayılım ağının karakteristik çalışmasını iyileştiren basit bir metot keşfetmişlerdir. Bunlar konvansiyonel olarak girdilerin 0-1 dinamik aralığının ve gizli nöronun çıktıların optimum olmadığını belirlemişlerdir. Bunun için aktivasyon fonksiyonu önerdiler

Bu aktivasyon fonksiyonu yaklaşım süresini %30 ile % 50 oranında azaltmıştır. Bu algoritmanın performansı pratik olarak yapılan düzenlemelere verilen bir örnektir. Pineda (1988) ve Almeida (1987) yapay sinirsel ağın çıktıklarını aynı ağa girdi olarak uygulamışlardır. Sonuç olarak öğrenme işleminin bu gibi sistemlerde oldukça hızlı ve kararlılık kriterinin oldukça tatmin edici olduğunu görmüşlerdir.

3.14. YSA' da Kullanılan Önemli Öğrenme Algoritmaları

Burada eğitici ve eğitici olmayan öğrenmeye yönelik yedi önemli öğrenme algoritması verilecektir (Zurda, 1991).

- a) **Hebb öğrenme algoritması** : Hebb' in öğrenme algoritmasında temel fikir, hücrenin darbe üretmesine sebep olan bağlantıların ağırlık katsayılarını büyüterek etkilerini artırmaktadır. Dışarıdan istenilen bir çıkış uygulandığından eğitici olmayan bir öğrenme algoritmasıdır.

- b) Algılayıcı (perceptron) öğrenme algoritması :** Rosenblatt (1958) tarafından önerilen algılayıcı öğrenme algoritmasında ağırlık (w) değişimi çıkış ile istenen çıkışın farkı ile orantılıdır, dolayısıyla eğitici bir öğrenme algoritmasıdır.
- Devrenin çıkışı iki kutuplu ve w 'nin oluşturduğu bir hiper yüzey tarafından belirlenmektedir. Ağırlık ürettiği çıkış istenen çıkıştan farklı ise, hiperyüzey hatayı azaltacak yönde hareket etmektedir. Hata sıfır olduğunda w değişimi durmaktadır.
- c) Eğim-düşme öğrenme algoritması :** Eğim-düşme (Gradient Descent) öğrenme algoritması veya “Delta Kuralı” öğrenme algoritması sadece üretilebilir fonksiyonlara sahip olan ağırlıklara uygulanabilmektedir. Eğitici bir öğrenme algoritmasıdır. w değişimi hatanın eğiminin ters yönündedir. Böylece hata fonksiyonunun oluşturduğu çanağın dibine, yani hatanın minimum olduğu yere doğru hareket edilmektedir.
- d) Widrow-Hof öğrenme algoritması :** Widrow-Hof öğrenme algoritması eğitici bir algoritmadır. w değişimi ağırlığın çıkışından bağımsızdır. Aslında bu algoritma eğim-düşme algoritmasının özel bir halidir.
- e) İlinti öğrenme algoritması :** Hebb öğrenme algoritmasının eğitici uyarlamasıdır.
- f) Kazanan her şeyi alır öğrenme algoritması :** Yarışma türü öğrenme algoritmalarına örnek teşkil etmekte olan bu algorithmada genel kural, giriş değerine en yakın olan bağlantı ağırlık katsayılarını bulmaktır. Bu w ' lere ilişkin hücre kazanan hücre olarak adlandırılmaktadır. Sadece kazanan hücre bir çıkış üretmekte ve bu hücreye ilişkin w ' ler değişime uğramaktadır. Eğitici öğrenmenin bir örneği olan bu algoritma sonuçta ağa gelen girişleri sınıflandırmaktadır.

- g) Outstar öğrenme algoritması :** Eğiticili öğrenme algoritma türü olan bu öğrenme algoritmasında amaç, w' leri istenen çıkışa benzemektir (Freeman, Skapura, (1992)).

BÖLÜM 4. UYGULAMA: NEURAL NETWORK YAKLAŞIMI İLE ÜRETİM PARAMETRELERİNİN BELİRLENMESİ: SERAMİK FİRMASINDA BİR ÇALIŞMA

4.1.Giriş

Çelvit A.Ş seramik ürünleri üreten bir firmadır. İnşaat malzemeleri sektöründe satıcı iken 1996 yılında inşaat malzemeleri üretimine geçtiler. E-5 karayolu üzerinde Büyük Esence Mevki'nde faaliyetini sürdürmektedir. 1996 yılında kırk metre boyundaki tünel fırınlarla üretime başladı. 1996 Eylül 'ünde yeni yanma sistemleri getirildi. İşletmede tuvalet taşı, şenks klozet, aksesuar, takım ayak, tuvalet taşı, jet tuvalet taşı, oval 40x50 lavabo, takım lavabo, 40x50 lavabo, rezervuar, 28x45 lavabo, takım klozet, pisuvar, klozet üretilmektedir.

ÇELVİT A.Ş. üç kişilik idari personeli, beş kişilik teknik elemanı ve yetmiş kişilik işçisiyle çalışmalarını sürdürmektedir. Bu işletmede iki tanede danışman görev yapmakta. Bu danışmanlardan biri fırınlar konusunda diğeri ise tüm seramik konularında uzman kişiler. Firmada kışın belli bir stok seviyesinde, yazın ise talep ağırlıklı çalışılıyor. Ürün satışlarında yüz yüze sıcak satışlar vardır.

4.2.Üretim Prosesleri

Vitrifiye üretimi yapan Çelvit A.Ş'de ürünler yedi işleminden geçer.

1. Çamurhane,
2. Dökümhane,

3. Kurutma,
4. Rötüslama,
5. Sırlama,
6. Fırınlama,
7. Kalite kontrol.

4.2.1.Çamurhanedeki üretim aşamaları

Üretimde kullanılan hammaddeleri özsüz hammaddeler, özlü hammaddeler diye iki kısma ayırmıştık. Özsüz hammaddelerin üretimde kullanılabilir hale gelmesi için öncelikle işlenmesi gerekir. Bunun için bu hammaddeler değirmenlerde işlenerek kullanılabilir hale gelir. Çamurda feldspat, kaolen, silis kumu kil, silikat, baryum CO₃ ve su karıştırılır. Üretimde kullanılan özsüz hammaddeler değirmen içinde 25 saat su ile birlikte döndürülür (18-19 devir/dk). Bu arada özlü hammaddeler de kil açıcıda su ile karıştırılır. Sonra özlü ve özsüz hammaddelerin kil açıcıda 2.5-3 saat karıştırılır ve stok havuzlarına alınır. Bir diğer değirmen sırlamada kullanılan çamuru yapmak için kullanılan değirmendir. Sır çamurunda albit, kuars, zirkon, çinko oksit, mermer, kaolen, soda ve su kullanılır

Çamurhanede üç işçi, iki teknik eleman görev yapmaktadır. Ayrıca hammaddelerin değirmene ve kil açıcıya taşınması iki işçi tarafından el arabalarıyla yapılıyor.

4.2.2.Dökümhanedeki üretim aşamaları :

Tuvalet taşı döküm aşamaları :

1-) Tuvalet taşının dökümü gece gerçekleştirilmekte. Gece 01.00'de çamurhaneden çekilen çamur kalıplara dökülmekte.

- 2-) Çamurun kalıpta kalma süresi bir saat elli dakika. Yani 01.00 - 02.50 arası .
- 3-) Kalıbın sökülebilmesi için kalıba yarım saat boyunca havanın verilmesi işlemi . 02.50 - 03.20 arası.
- 4-) Kalıptaki çekirdeklerin (yardımcı elemanların) çıkarılması.09.00 - 10.00 arası.
- 5-) Çekirdekteki çapakların temizlenmesi. 10.00 - 11.00 arası.
- 6-) Mamuldeki deliklerin tıkanması. 11.00 - 12.00 arası.
- 7-) Mamullerin kalıplardan alınıp raflara dizilmesi. 12.00 - 12.45 arası.
- 8-) Raftaki malların çapaklarının (fazlalıkların) alınması. 12.45 - 13.45 arası.
- 9-)Kalıpların bir sonraki döküme hazır hale getirilmesi. 13.45 - 16.45 arası.
- 10-) Raflara alınan malların genel rötuşunun yapılması. 16.45 - 18.45 arası.
- 11-) Raflardaki malların 20-22 saat sonra kurutma odasına alınması.

Lavabo taşı üretme aşamaları :

- 1-) Kalıbın kontrol edilmesi . 08.45 - 09.00 arası.
- 2-) Çamurhaneden alınan çamurun kalıplara dökülmesi. 09.00 - 10.10 arası.
- 3-) Çamurun boşaltılması. 10.10 - 10.25 arası.
- 4-) Mamulün kalıptan sökülebilmesi için kalıba hava verilmesi. 10.25 - 11.25 arası.
- 5-) Kalıbın açılmasından sonra mamuldeki deliklerin açılması. 11.25 - 12.00 arası.
- 6-) Kalıptan çıkarılan mamulün regallere (raflara) alınması. 12.00 - 13.00 arası.
- 7-) Bir sonraki döküm için kalıpların hazırlanması. 13.00 - 15.00 arası.

8-) Raflara alınan mamullerin genel rötuşunun yapılması. 15.00 - 17.00 arası.

9-) 22 - 24 saat raflarda bekledikten sonra kurutma odalarına alınması.

Klozet taşının dökülmesi :

1-) Kalıbın kontrol edilmesi. 08.45 - 09.00 arası.

2-) Çamurhaneden çekilen çamurun kalıplara dökülmesi. 09.00 - 09.40 arası.

3-) Çamurun kalıpta şekil alması için bekletilmesi. 09.40 - 11.20 arası.

4-) Kalıbın ters çevrilmesi. 11.20 - 12.45 arası.

5-) Kalıpların açılmasından sonra deliklerin delinmesi ve iç haznenin temizlenmesi. 13.30 - 15.45 arası.

6-) Mamulün raflara (regellere) alınması. 15.45 - 17.00 arası

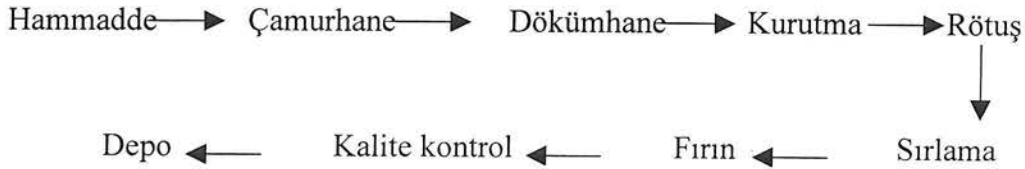
7-) Mamullerin kurutma odasına alınmadan önce 22 saat bekletilmesi.

4.2.3. Kurutma odasındaki üretim aşamaları

Dökümhaneden sonra mamullerin uğradığı bölüm kurutma odalarıdır. Buraya gelen ürünler 22 saat kalıyorlar. Kurutma odalarının kapasitesi 700 - 800 ürün arasında değişiyor. Mamuller kurutma odalarına iki veya üç katlı arabalarla alınıyor. Kurutma odalarına yirmi beş araba girebiliyor. İki katlı arabalar yirmi altı tane tuvalet taşı ve on tane klozet alabiliyor. Üç katlı arabalar ise on beş tane klozet taşı, otuz lavabo, iki yüz elli aksesuar, on beş tane pisuvar, otuz tane ayak alabilmekte. Kurutma odalarında mamuller belli bir ısıda tutularak daha iyi kurumaları sağlanıyor. Yirmi iki saatin sonunda mamuller dışarı alınmakta.

4.2.8. Depo:

Kalite kontrolü yapılmış sağlam ürünler depoya alınır.



Şekil 4. 1. Üretim aşamaları

4.3. Üretim Parametrelerinin Neural Network İle Tahmin Edilmesi

Neural network ile üretim parametrelerini tahmin ederken dört adım takip edilmiştir. Birincisi, işletmenin üretim prosesinin ProModel simülasyon paket programı ile modellenmesidir. İkincisi, farklı üretim parametreleri ile simülasyon sonuçlarının elde edilmesidir. Üçüncüsü, simülasyon sonuçlarından oluşturulan eğitim datasıyla neural network geliştirilmesidir. Son olarakta neural network kullanılarak ürünlerin talep miktarlarıyla üretim parametrelerinin tahmin edilmesidir.

4.3.1. Çelvit A.Ş'nin ProModel paket programı ile modellenmesi

Bu çalışmamız ÇELVİT AŞ. de ham olarak gelen malzemenin ürün olarak çıkışı için gerekli işlemlerin modellenmesini içermektedir. Bu ürünler tuvalet taşı, lavabo, klozet ve aksesuarlardır

Bu çalışmada önce Çelvit A.Ş'nin mevcut durumu modellenmiş ve istatistik sonuçları da incelenerek üretimdeki tıkanıklıklar, aksaklıklar, beklemler tespit edilmiştir. Daha sonra sorunları çözecek şekilde yeni bir model geliştirilmiştir. İşletmede mevcut durumda bir fırın vardı, geliştirilen modelde ise fırın sayısı ikiye

çıkarıldı. Kurutma odasının kapasitesi 750 idi, geliştirilen modelde 1000 adet yapıldı. Bu sayede üretim miktarları , tezgahların verimlilikleri, kaynaklardan yararlanma oranı arttı.

Bir modeli kurabilmek için iki ana menüyü bilmek gerekmektedir. Bu menüler build ve options menüsüdür. Öncelikle kuracağımız modelde build menüsü modüllerin nasıl kurulacağını ve neler yapılacağı açıklanacaktır.

4.3.1.1.Model ile ilgili genel bilgiler (General information)

Modelin oluşturulmadan önce model ile ilgili genel bazı bilgilerin girilmesi gerekmektedir. Bu bilgiler zaman birimi (saniye/dakika/saat), mesafe birimi (feet/metre), kullanılacak grafik dosyasıdır. Bu modelde zaman birimi dakika, mesafe birimi metredir

4.3.1.2. Modelin yerleşimlerinin girilmesi (Locations)

Modeldeki yerleşimlerin layout üzerinde gösterilmesi gerekmektedir. Build menüsünden locations tıklanır. Grafik dosyasındaki sembollerden istenenler sıra ile mouse kullanılarak layout üzerinde istenen yere gelinip tıklanır. Yerleşim ebatları küçültülüp büyütülebilir. Sistemimizde yirmi dokuz tane yerleşim kullanılmıştır ve bunlar tablo 4.1 deki gibi girilmiştir. Bozulma zamanının (downtime) none olması belirlenen yerleşimin hiç bozulmayacağını göstermektedir. İstatistikler kısmında bilgilerin sadece ana bilgiler olacağı belirlenmiştir. Rules da ise yerleşimler de girdilerin hangi kurala göre işlem göreceği belirlenmiştir.

Tablo 4.1. Modeldeki yerleşimler ve özellikleri

Name	Capacity	Units	Dawntimes	Statistic	Rules
Değirmen1	300	1	none	Time series	oldest
Değirmen2	300	1	none	Time series	oldest
Stok havuzu1	300	1	none	Time series	oldest
Stok havuzu2	300	1	none	Time series	oldest
Stok havuzu3	400	1	none	Time series	oldest
Ttaşı kalıpları	300	1	none	Time series	oldest
Lavabo kalıpları	300	1	none	Time series	oldest
klozet kalıpları	300	1	none	Time series	oldest
Kurutma stoğu	5000	1	none	Time series	oldest
Kurutma	650	1	none	Time series	oldest
Rotus kuyruğu	Infinite	1	none	Time seriesoldest,FIFO	
Rotus1	2	1	none	Time series	oldest
Fırın1	300	1	none	Time series	oldest
Kalite kontrol alanı	1	1	none	Time series	oldest
Hammadde stok	4000	1	none	Time series	oldest
Kil açıcı	2000	1	none	Time series	oldest
Bekleme rafları	2000	1	none	Time series	oldest
Loc1	3000	1	none	Time series	oldest
Rotus2	2	1	none	Time series	oldest
Rotus3	2	1	none	Time series	oldest
Sırlama2	2	1	none	Time series	oldest
Sırlama3	2	1	none	Time series	oldest
Loc2	5000	1	none	Time series	oldest
Desk	1	1	none	Time series	oldest
Loc3	5000	1	none	Time series	oldest
Desk2	1	1	none	Time series	oldest

4.3.1.3. Girdilerin belirlenmesi (Entities)

Modelimizde beş tane girdi vardır. Bunlar aşağıdaki tabloda gösterilmiştir.

Tablo 4.2. Modelde kullanılan girdiler ve özellikleri.

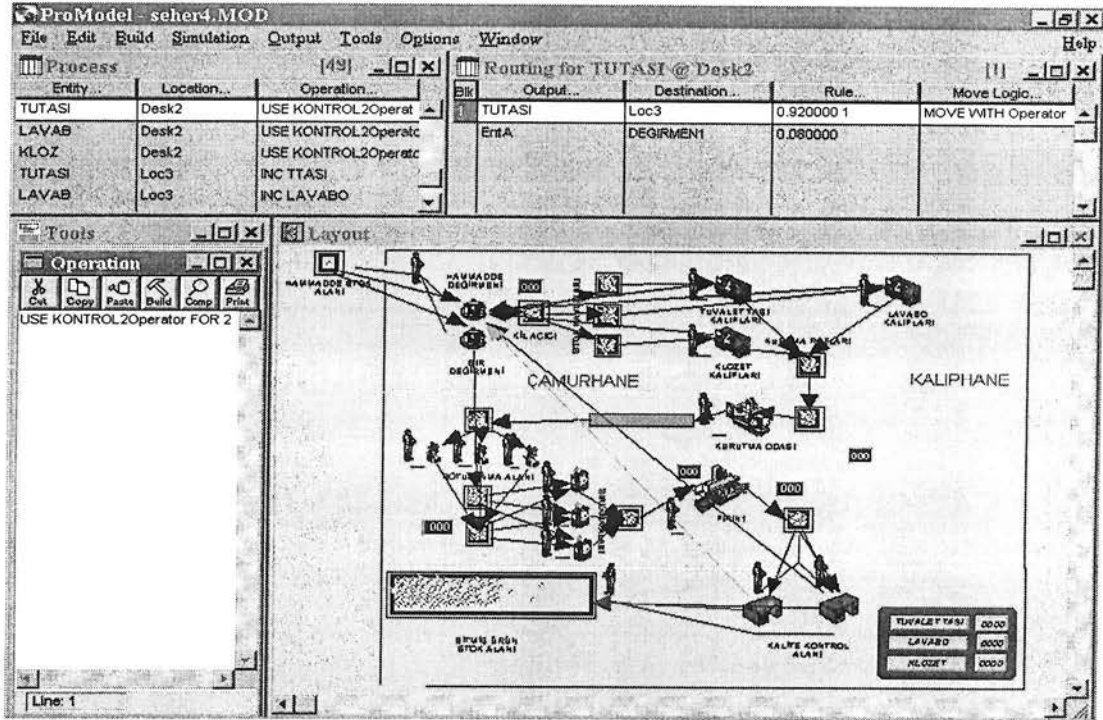
Name	Speed	Stats
Hammadde	50	Time series
Tutası	50	Time series
Lavab	50	Time series
Kloz	50	Time series
Sır çamuru	50	Time series

Burada kullanılan girdiler sistemde işlenecek olan mamulün işlemler esnasında geçirdiği aşamalarındaki durumlarını ifade etmek için kullanılan terimlerdir. Değirmen1 de işlenmek üzere hammadde gelmektedir. Daha sonra kalıplarda işlenerek tutası, lavab ve kloza dönüşecektir. Değirmen2 ye gelen hammadde işlenerek sır çamuru olarak çıkmaktadır. Speed kısmında, girdilerin bir yol boyunca herhangi bir hareketi için belirlenen bir zaman dilimi içinde yine belirlenen bir uzaklık ölçüsünde ne kadar bir hızla yol alacağını göstermektedir.

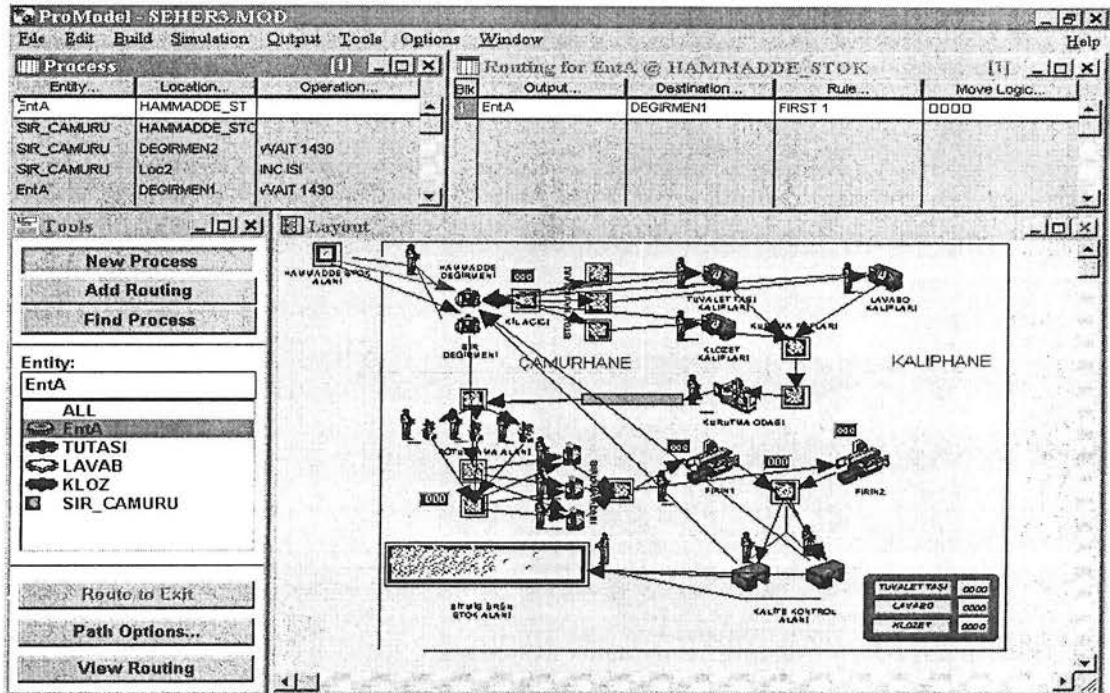
4.3.1.4. Kaynakların belirlenmesi (Resources)

Oluşturduğumuz modelde 15 tane operatör kullanılmıştır. Name adı altında operatörlerin isimleri verilmiştir. DTs bozulmaları göstermektedir ki bizim kullanmış olduğumuz kaynak bozulmamaktadır. Stats kısmı simülasyon sonunda kaynaklar hakkındaki istatistiki bilgilerin birim şeklinde olduğunu göstermektedir. Specs o kaynağın hangi yolu kullanacağı ve girdileri taşımadığı süre içinde kaynağın hangi düğümde duracağını göstermektedir. Örneğin operatör 2 Net15 yolunu izleyecek ve boş kaldığı anlarda N1 düğümünde çağırılmayı bekleyecektir.

varış yerleşimi, proses kuralı ve hareket eşleminin nasıl olacağı girilir.



Şekil 4.2. Mevcut durumdaki proseslerin belirlenmesi



Şekil 4.3. Geliştirilen modeldeki proseslerin belirlenmesi

Tablo 4.5. Modeldeki işlem adımları ve süreleri

Entity	Locatio	Operation	Output	Destina.	Rule	Move logic
EntA	Ham. stok		EntA	Değirmen1	First1	
Sır çamuru	Ham.stot		Sır çamuru	Değirmen2	First1	
Sır çamuru	Değirmen2	WAIT 1430	Sır çamuru	loc2	First1	
Sır çamuru	loc2	INC ISI	Sır çamuru	Sırlama 1	Join 1	Move for 1 dec ısı
EntA	Değirmen1	WAIT 1430	Enta	Kil açıcı	First1	
EntA	Kil açıcı	INC PL wait 180	EntA	stok havuzu1	0.30	DEC PL
EntA	St.havuzu1		EntA	Ttası kalıpları	First1	
EntA	st.havuzu2	WAIT 265	EntA	lvb.kalıpları	First1	
EntA	St.havuzu3	WAIT 355	EntA	Klz.kalıpları	First1	
EntA	Ttası kalıbı	WAIT 715	Tutası	Bek.rafları	0.94	
EntA	Lvb.kalıbı	WAIT 450	Lavab	Bek.rafları	0.94	
EntA	Klz.kalıbı	WAIT 360	Kloz	Bek.rafları	0.94	
Tutası	Bek.rafları	WAIT 1320	Tutası	Kurutma stoğu	First1	
Lavab	Bek.rafları	WAIT 1320	Lavab	Kurutma stoğu	First1	
Kloz	Bek.rafları	WAIT 1320	Kloz	Kurutma stoğu	First1	
Tutası	Kur.stoğu		Tutası	Kurutma	First1	
Lavab	Kur.stoğu		Lavab	Kurutma		
Kloz	Kur.stoğu		Kloz	Kurutma		
Tutası	kurutma	WAIT 1440	Tutası	Rotus kuyruğu	Random1	
Lavab	kurutma	WAIT 1440	Lavab	Rotus kuyruğu	Random1	
Kloz	kurutma	WAIT 1440	Kloz	Rotus kuyruğu	Random1	
Tutası	Rotus kuyru.		Tutası	Loc1	First1	
Lavab	Rotus kuyru.		Lavab	Loc1	First1	
Kloz	Rotus kuyru.		Kloz	Loc1	First1	
Tutası	Loc1		Tutası	Rotus1	First1	
Lavab	Loc1		Lavab	Rotus2	First1	
Kloz	Loc1		Kloz	Rotus3	First1	
Tutası	Rotus1	WAIT 2	Tutası	Sır kuyruğu	First1	
Tutası	Sır kuyruğu		Tutası	Sırlama1	First1	
Lavab	Rotus2	WAIT 3.06	Tutası	Sır kuyruğu	First1	
Lavab	Sır kuyruğu		Tutası	Sırlama2	First1	
Kloz	Rotus3	WAIT 7	Tutası	Sır kuyruğu	First1	
Kloz	Sır kuyruğu		Tutası	Sırlama3	First1	
Tutası	Sırlama1	Join1 sır çamuru WAIT 5	Tutası	Fırın stok alanı	First1	
Lavab	Sırlama1	Join1 sır çamuru WAIT 6	Lavab	Fırın stok alanı	First1	
Kloz	Sırlama1	Join1 sır çamuru WAIT 7	Kloz	Fırın stok alanı	First1	
Tutası	fır.st.alanı		Tutası	Fırın 1	Random1	
Lavab	fır.st.alanı		Lavab	Fırın 1	Random1	
Kloz	fır.st.alanı		Kloz	Fırın 1	Random1	
Tutası	Fırın1		Tutası	Kkontrol alanı	First1	Dec var1

Tablo 4.5.(Devam) Modeldeki işlem adımları ve süreleri

Lavab	Fırın1	Inc var1 WAIT 960	Lavab	Kkontrol alanı	First1	Dec var1
Kloz	Fırın1	Inc var1 WAIT 960	Kloz	Kkontrol alanı	First1	Dec var1
Tutasi	Kkontrol	Inc Fir	Tutasi	Desk	Random1	Move with kopepator for .1 then free
Lavab	Kkontrol	Inc Fir	Lavab	Desk2	Random1	Move with kopepator for .1 then free
Kloz	Kkontrol	Inc Fir	Kloz	Desk	Random1	Move with kopepator for .1 then free
Tutasi	Desk	use kontrol1 operator for 2	Tutasi	Loc3	0.92	Move with operator2 for .1 then free
			EntA	Değirmen1	0.08	
Lavab	Desk	use kontrol1 operator for 2	Lavab	Loc3	0.92	Move with operator2 for .1 then free
			EntA	Değirmen1	0.08	
Kloz	Desk	use kontrol1 operator for 2	Kloz	Loc3	0.92	Move with operator2 for .1 then free
			EntA	Değirmen1	0.08	
Tutasi	Desk	use kontrol2 operator for 2	Tutasi	Loc3	0.92	Move with operator2 for .1 then free
			EntA	Değirmen1	0.08	
Lavab	Desk	use kontrol2 operator for 2	Tutasi	Loc3	0.92	Move with operator2 for .1 then free
			EntA	Değirmen1	0.08	
Kloz	Desk	use kontrol2 operator for 2	Tutasi	Loc3	0.92	Move with operator2 for .1 then free
			EntA	Değirmen1	0.08	
Tutasi	Loc3	Inc Ttasi WAIT 3000	Tutasi	EXIT	First1	
Lavab	Loc3	Inc Ttasi WAIT 3000	Tutasi	EXIT	First1	
Kloz	Loc3	Inc Ttasi WAIT 3000	Tutasi	EXIT	First1	

4.3.1.7. Varışların tanımlanması (Arrivals)

Arrivals bölümünde sisteme ulaşacak girdilerin belirtilmesi işlemi yapılır. Bu modelde üç tane giriş olmaktadır. EntA hammadde stok alanına ve kil açıcıya gelmekte, sır çamuru ise hammadde stok alanına gelmektedir. EntA'nın hammadde stok alanına gelişi ile simülasyon başlamaktadır. Bu durum First time (ilk zaman) alanından sıfır olarak belirlenmiştir. Gelişler arasındaki zaman ise frekans kısmından belirlenir. Buradaki gelişler her 1430 dakikada bir olmaktadır.

Tablo 4.6. Modele gelen entity geliş zamanları, frekansları ve miktarları

Entity	Location	Qty Eac	First ti	Occur.	Frequenc.	Logic	Disable
EntA	Ham.Stok	680	0	2000	1430		No
EntA	Kil açıcı	320	1430	2000	1430		No
Sır çamuru	Ham.stok	1000	3657	2000.	1430		No

4.3.2. Farklı sistem parametreleri ile üretim miktarlarının belirlenmesi

Bu bölümde oluşturulan simülasyon modelinde farklı parametreler verilerek farklı üretim değerleri elde edilmiştir. Tuvalet taşı, lavabo ve klozet taşının kalıp kapasiteleri değiştirilerek üretimin değişik miktarları elde edilmiştir. Oluşturulan bu farklı değerlerler Neural Network de kullanılmıştır.

Çıktı değişkenleri:

O1: Değirmen kapasitesi
 O2: Klozet taşı kapasitesi
 O3: Tuvalet taşı kapasitesi
 O4: Lavabo kapasitesi
 O5: Kurutma Odası kapasitesi
 O6: Sırlama alanı 1

O7: Sırlama alanı 2
 O8: Sırlama alanı 3
 O9: Rötuşlama alanı 1
 O10: Rötuşlama alanı 2
 O11: Rötuşlama alanı 3
 O12: Fırın sayısı

Girdi değişkenleri:

I1: klozet üretim miktarı
 I2: Tuvalet taşı üretim miktarı
 I3: lavabo üretim miktarı

Tablo 4.7.Girdi ve çıktı parametreleri

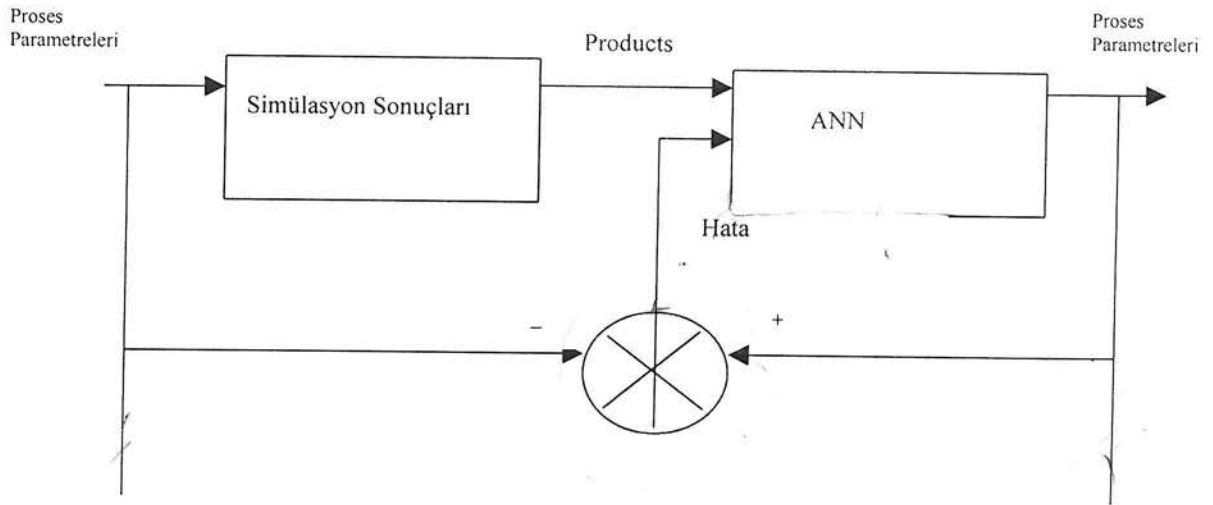
OUTPUT VARIABLES												ÇIKTI DEĞİŞKENLERİ		
O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	I1	I2	I3
1500	200	200	300	650	1	2	2	1	2	2	1	2529	2585	3401
1500	200	200	500	1000	2	2	2	2	2	2	2	4063	4081	4743
1500	200	300	300	650	2	2	2	2	2	2	2	3685	3671	4859
1500	200	300	500	1000	2	2	2	2	2	2	2	3993	4125	4775
1500	200	400	300	650	2	2	2	2	2	2	2	3680	3689	4862
2000	200	400	500	1200	3	3	2	3	3	2	2	4150	4195	4556
2000	200	500	300	1000	3	2	2	3	2	2	2	4026	4179	4710
2000	200	500	500	1200	3	3	2	3	3	2	2	4150	4195	4556
1500	300	200	300	650	2	2	2	2	2	2	2	3742	3633	4813
2000	300	200	500	1000	3	2	2	3	2	2	2	4151	4044	4716
2000	300	300	400	1000	3	2	2	3	2	2	2	4118	4088	4691
1500	300	400	200	1000	3	2	2	3	2	2	2	4103	4085	4736
1500	300	500	200	1000	3	2	2	3	2	2	2	4103	4085	4736
2000	300	500	400	1200	3	3	2	3	3	2	2	4175	4150	4559
1500	400	200	200	650	2	2	2	2	2	2	2	3771	3632	4826
1500	400	200	400	1000	3	2	2	3	2	2	2	4143	4057	4726
1500	400	300	200	650	2	2	2	2	2	2	2	3816	3643	4745
2000	400	300	400	1000	3	2	2	3	2	2	2	4183	3996	4714
2000	400	400	200	1000	3	2	2	3	2	2	2	4140	4080	4699
2000	400	400	400	1200	3	3	2	3	3	2	2	4205	4134	4557
2000	400	500	200	1200	3	3	2	3	3	2	2	4182	4211	4520
2500	400	500	400	1200	3	3	2	3	3	2	3	4164	4096	4652
1500	500	200	200	650	2	2	2	2	2	2	2	3767	3645	4834
2000	500	200	400	1200	3	3	2	3	3	2	2	4143	4187	4588
1500	500	300	200	1000	2	3	2	2	3	2	2	4124	4071	4719
2000	500	300	400	1200	3	3	2	3	3	2	2	4173	4172	4568
2000	500	400	200	1200	3	2	2	3	2	2	2	4192	4126	4601
2500	500	400	400	1200	3	3	2	3	3	2	3	4164	4096	4652
2500	500	500	200	1000	3	3	2	3	3	2	3	4148	4032	4725
2000	300	300	300	650	2	2	2	2	2	2	1	2623	2574	3374
1500	200	200	300	650	1	2	2	1	2	2	1	2529	2585	3401

Tablo 4.7. (Devam) Girdi ve çıktı parametreleri

ÇIKTI DEĞİŞKENLERİ												GİRDİ DEĞİŞKENLERİ		
O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	I1	I2	I3
1500	200	200	200	650	1	2	2	1	2	2	1	2529	2254	3413
1500	200	200	400	650	2	2	2	2	2	2	2	3654	3623	4955
1500	200	300	200	650	2	2	2	2	2	2	1	2602	2587	3345
1500	200	300	400	650	2	2	2	2	2	2	2	2565	2574	3369
1500	200	400	200	650	2	2	2	2	2	2	2	3688	3650	4880
2000	200	400	400	1000	3	2	2	3	2	2	2	4066	4106	4719
1500	200	500	200	650	2	2	2	2	2	2	2	3696	3634	4857
2000	200	500	400	1200	3	3	2	3	3	2	2	4157	4222	4601
1500	300	200	200	650	2	2	2	2	2	2	2	3721	3629	4801
1500	300	200	400	1000	2	2	2	2	2	2	2	4095	4074	4720
1500	300	300	200	650	2	2	2	2	2	2	2	3756	3741	4730
2000	300	300	500	1200	3	3	2	3	3	2	2	4176	4151	4565
2000	300	400	400	1000	3	2	2	3	2	2	2	4138	4041	4718
2000	300	500	300	1200	3	3	2	3	3	2	2	4127	4187	4583
2500	300	500	500	1200	3	3	2	3	3	2	3	4133	4114	4658
1500	400	200	300	1000	2	2	2	2	2	2	2	4135	4040	4739
2000	400	200	500	1000	2	2	2	2	2	2	2	4127	4048	4755
1500	400	300	300	650	2	2	2	2	2	2	2	2570	2566	3388
2000	400	300	500	1200	3	3	2	3	3	2	2	4182	4107	4596
2000	400	400	300	1000	3	2	2	3	2	2	2	4088	4130	4711
2500	400	400	500	1200	3	3	2	3	3	2	3	4147	4093	4679
2000	400	500	300	1200	3	3	2	3	3	2	3	4111	4124	4698
2500	400	500	500	1400	3	3	3	3	3	3	3	4058	4066	4781
1500	500	200	300	1000	3	2	2	3	2	2	2	4168	4054	4667
2000	500	200	500	1200	3	3	2	3	3	2	2	4127	4192	4583
2000	500	300	300	1200	3	3	2	3	3	2	2	4160	4189	4583
2500	500	300	500	1200	3	3	2	3	3	2	3	4125	4124	4669
2500	500	400	300	1200	3	3	2	3	3	2	2	4117	4098	4688
2500	500	400	500	1200	3	3	2	3	3	2	3	4177	4092	4679
2500	500	500	500	1500	3	3	3	3	3	3	3	4058	4066	4781
2000	300	300	300	1000	2	2	2	2	2	2	2	4119	4071	4740

4.3.3. Neural network modelinin oluşturulması

Üçüncü adım olarak ANN modeli geliştirilmiştir. ProModel ile üretim miktarları bulunuyordu. Fakat ProModel ile modeli simüle etmenin çok fazla zaman aldığı görüldü. Bir sonuca ulaşabilmek için simülasyon süreleri otuz dakika ile beş saat arasında değişmektedir. En önemli dezavantajı ise istenilen üretim miktarını bulmak için sistemi bir kere simüle etmenin yetersiz olmasıdır. İstenilen sonuca ulaşmak için çok fazla deneme yapmak gerekmektedir. Buda çok fazla zaman anlamına gelmektedir. Buradaki zaman kaybını engellemek amacıyla ANN modeli oluşturulmuştur.



Şekil 4.4. Yeni bir ANN kurulması ve simülasyon sonuçlarıyla bütünleştirilmesi

Karmaşık bir modelde modelin tüm ayrıntılarının paralel olarak işlenmesi çözümü kolaylaştırır. ANN paralel dağıtımlı proses sistemidir. Tek yönlü sinyal akış kanalları aracılığıyla birbirine bağladığı operatörleri birleştirir. Bu sayede lineer olamayan bir eğitim biçimlenir. Girdiler ve çıktılar arasında girdi, çıktı katmanları ve gizli katmanlar içerir. ANN yaklaşımının ana fikri insan beyni fonksiyonlarına benzemektir. Bu nedenle ANN çabuk yanıt verir ve sıralı dijital bilgisayarlara göre performansı daha yüksektir. Farklı girdilere mantıklı cevaplar vererek ortama kendi kendine adapte olabilir. Bununla birlikte karışık bir iç yapısı vardır. ANN verilerini

kavramak, nöronların temel biyolojik fonksiyonlarını taklit eden sistemler olduğundan karışıktır (Abulafya, 1995), (Narendra, 1992). Çalışmada backpropagation algoritması kullanılmıştır. Backpropagation algoritması bölüm 3'te anlatılan adımları içerir.

Burada $f(x)$ aktivasyon fonksiyonu olarak gösterilmektedir.

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

4.3.3.1.Önerilen ANN modeli

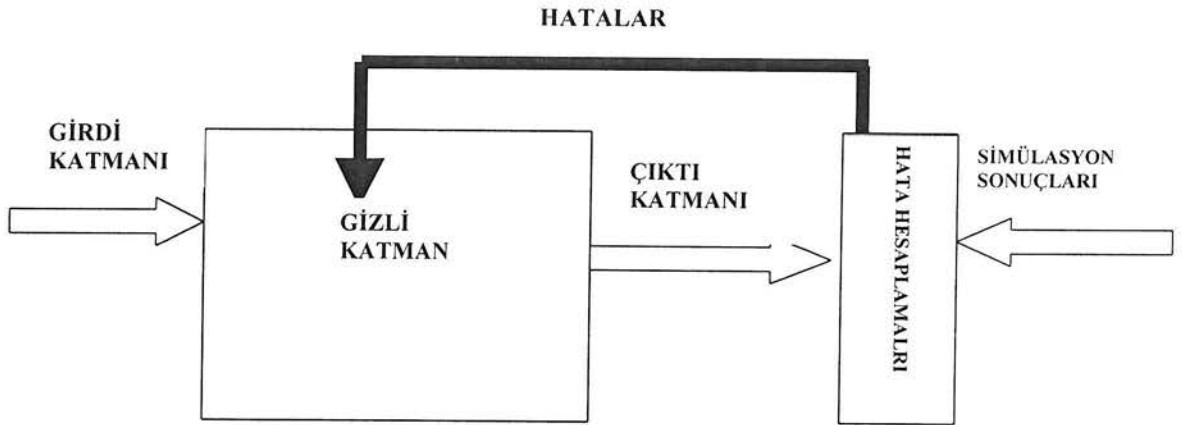
Bu çalışmada yani bir ANN tasarlandı. ANN'de bir girdi katmanı, bir gizli katman ve bir de çıktı katmanı vardır. Katman, girdi ve çıktı değişkenleri tablo 4.8'de görülmektedir. Eğitim ve test adımları ise şekil 4.5 ve şekil 4.6'da görülmektedir.

Tablo 4.8. ANN katmanları ve değişkenleri

GİRDİ KATMANI	GİZLİ KATMAN	ÇIKTI KATMANI
3 NÖRON • Klozet Üretim Miktarı • Tuvalet Taşı Üretim Miktarı • Lavabo Üretim Miktarı	22 NÖRON	12 NÖRON • Değirmen kapasitesi • Kozet taşı kapasitesi • Tuvalet taşı kapasitesi • Lavabo kapasitesi • Kurutma odası kapasitesi • Rötuşlama alanı 1 • Rötuşlama alanı 2 • Rötuşlama alanı 3 • Sırlama alanı 1 • Sırlama alanı 2 • Sırlama alanı 3 • Fırın kapasitesi

- **Eğitim aşaması**

ANN oluşturulurken ProModel sonuçlarında faydalanılmıştır. Bunun için sistem farklı parametrelerle 61 kere simüle edilmiştir. Bu denemeler tablo 4.7 de gösterilmiştir. Bu 61 denemeden rasgele seçilen 31 tanesi eğitim seti olarak kullanılmıştır. Bunlar tablo 4.9'da gösterilmiştir. Geri kalan 30 deneme ise test aşaması için kullanılmıştır. Neurel Network'de aktivasyon (sigmoid) fonksiyonu 0 ve 1 aralığındadır. Fakat modelin simülasyon sonuçları 0 ve 5000 aralığındadır. Bu nedenle simülasyon sonuçları 0 ve 1'e indirgenmiştir. Eğitim setinin normalize edilmiş hali tablo 4.10 verilmiştir. Test aşaması için kullanılan simülasyon sonuçları tablo 4.11 deki gibidir. Bu değerlerin normalize edilmiş şekli tablo 4.12'de verilmiştir.



Şekil 4.5 ANN'in eğitimi

Simülasyon sonuçlarını normalize için aşağıdaki formül kullanılmıştır.

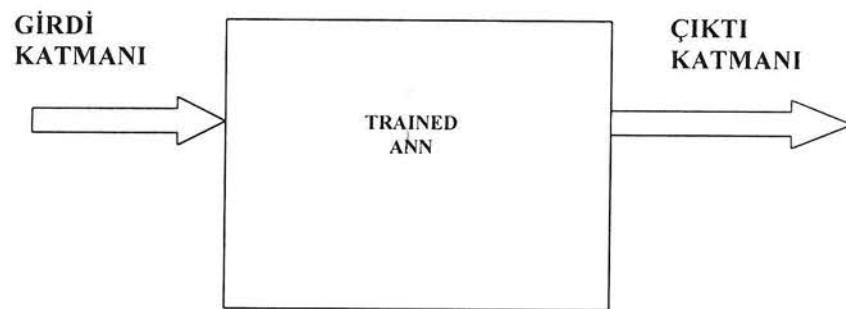
$$X_{nor} = \frac{X_{ger} - X_{min}}{X_{max} - X_{min}}$$

Kullanılan algoritmada üretim miktarları giriş değerleridir. Aktivasyon fonksiyonu olarak yukarıda bahsedilen sigmoid fonksiyonu kullanılmış ve çıkış değerleri de değirmen kapasitesi, tuvalet taşı kalıpları, lavabo kalıpları, klozet taşı kalıpları, kurutma odası kapasitesi, sırlama ve rötuş bölümündeki elemanların sayısı ve fırın adedidir.

Neural Network’de en iyi sonuca ulaşmak için momentum katsayısı 0.8 ve öğrenme oranı olarak 0.2 alınmıştır. Ortalama hata 0.02’den küçük oluncaya kadar Neural Network eğitim prosesinde iterasyonlara devam edilmiştir. Eğitim prosesi bir milyon adımda tamamlanmıştır. Neural Network eğitimi tamamlandıktan sonra değerlerin bazıları tahmin edildi. Tahmin değerleri gerçek simülasyon değerleriyle karşılaştırıldı. Neural Network sonuçları gösteriyor ki ağ % 90 başarılı olmuştur. Hata değerleri minimuma yaklaştığında ANN parametreleri saklandı ve sistem parametrelerinin tahmini için kullanıldı.

- **Test aşaması**

ProModel’den elde edilen atmış bir denemenin otuz tanesi test amaçlı kullanılmıştır. ANN eğitim parametreleri bulunduğu girdi parametreleriyle bulunan çıktı parametreleri test edilir. ANN ile belirlenen herhangi bir örneğin parametreleri ProModel sonuçları ile karşılaştırılır. Bu çalışmada hatalar kabul edilebilir aralıkta oluncaya kadar eğitime devam edilir. Şirket üretim miktarını belirlediğinde ANN ile arzu edilen ürünlerin üretim parametreleri kolayca bulunabilir.



Şekil 4.6.Eğitilen ANN’in test edilmesi

Geliştirilen ANN modeli üç ürünün üretim parametrelerini simülasyon paketinden daha çabuk ve iyi tahmin edebilmektedir.

Tablo 4.9. Neural network eğitim seti

ÇIKTI DEĞİŞKENLERİ												GİRDİ DEĞİŞKENLERİ		
O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	I1	I2	I3
1500	200	200	300	650	1	2	2	1	2	2	1	2529	2585	3401
1500	200	200	500	1000	2	2	2	2	2	2	2	4063	4081	4743
1500	200	300	300	650	2	2	2	2	2	2	2	3685	3671	4859
1500	200	300	500	1000	2	2	2	2	2	2	2	3993	4125	4775
1500	200	400	300	650	2	2	2	2	2	2	2	3680	3689	4862
2000	200	400	500	1200	3	3	2	3	3	2	2	4150	4195	4556
2000	200	500	300	1000	3	2	2	3	2	2	2	4026	4179	4710
2000	200	500	500	1200	3	3	2	3	3	2	2	4150	4195	4556
1500	300	200	300	650	2	2	2	2	2	2	2	3742	3633	4813
2000	300	200	500	1000	3	2	2	3	2	2	2	4151	4044	4716
2000	300	300	400	1000	3	2	2	3	2	2	2	4118	4088	4691
1500	300	400	200	1000	3	2	2	3	2	2	2	4103	4085	4736
1500	300	500	200	1000	3	2	2	3	2	2	2	4103	4085	4736
2000	300	500	400	1200	3	3	2	3	3	2	2	4175	4150	4559
1500	400	200	200	650	2	2	2	2	2	2	2	3771	3632	4826
1500	400	200	400	1000	3	2	2	3	2	2	2	4143	4057	4726
1500	400	300	200	650	2	2	2	2	2	2	2	3816	3643	4745
2000	400	300	400	1000	3	2	2	3	2	2	2	4183	3996	4714
2000	400	400	200	1000	3	2	2	3	2	2	2	4140	4080	4699
2000	400	400	400	1200	3	3	2	3	3	2	2	4205	4134	4557
2000	400	500	200	1200	3	3	2	3	3	2	2	4182	4211	4520
2500	400	500	400	1200	3	3	2	3	3	2	3	4164	4096	4652
1500	500	200	200	650	2	2	2	2	2	2	2	3767	3645	4834
2000	500	200	400	1200	3	3	2	3	3	2	2	4143	4187	4588
1500	500	300	200	1000	2	3	2	2	3	2	2	4124	4071	4719
2000	500	300	400	1200	3	3	2	3	3	2	2	4173	4172	4568
2000	500	400	200	1200	3	2	2	3	2	2	2	4192	4126	4601
2500	500	400	400	1200	3	3	2	3	3	2	3	4164	4096	4652
2500	500	500	200	1000	3	3	2	3	3	2	3	4148	4032	4725
2000	300	300	300	650	2	2	2	2	2	2	1	2623	2574	3374
1500	200	200	300	650	1	2	2	1	2	2	1	2529	2585	3401

Tablo 4.10. Neural network eğitim setinin normalize edilmesi

OUTPUT VARIABLES											INPUT VARIABLES				
O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	I1	I2	I3	
0,3	0,04	0,04	0,04	0,13	0,0002	0,0004	0,0004	0,0002	0,0004	0,0004	0,0002	0,5058	0,517	0,6802	
0,3	0,04	0,04	0,08	0,13	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,8126	0,8162	0,9486	
0,3	0,04	0,06	0,04	0,13	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0002	0,737	0,7342	0,9718	
0,3	0,04	0,06	0,08	0,13	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,7986	0,825	0,955	
0,3	0,04	0,08	0,04	0,13	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,736	0,7378	0,9724	
0,4	0,04	0,08	0,08	0,2	0,0006	0,0004	0,0004	0,0006	0,0004	0,0004	0,0004	0,83	0,839	0,9112	
0,3	0,04	0,1	0,04	0,13	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,8052	0,8358	0,942	
0,4	0,04	0,1	0,08	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0004	0,83	0,839	0,9112	
0,3	0,06	0,04	0,04	0,13	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,7484	0,7266	0,9626	
0,3	0,06	0,04	0,08	0,2	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,8302	0,8088	0,9432	
0,3	0,06	0,06	0,04	0,13	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,8236	0,8176	0,9382	
0,4	0,06	0,06	0,1	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0004	0,8206	0,817	0,9472	
0,4	0,06	0,08	0,08	0,2	0,0006	0,0004	0,0004	0,0006	0,0004	0,0004	0,0004	0,8206	0,817	0,9472	
0,4	0,06	0,1	0,06	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0004	0,835	0,83	0,9118	
0,5	0,06	0,1	0,1	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0006	0,7542	0,7264	0,9652	
0,3	0,08	0,04	0,06	0,2	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,8286	0,8114	0,9452	
0,4	0,08	0,04	0,1	0,2	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,7632	0,7286	0,949	
0,3	0,08	0,06	0,06	0,13	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,8366	0,7992	0,9428	
0,4	0,08	0,06	0,1	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0004	0,828	0,816	0,9398	
0,4	0,08	0,08	0,06	0,2	0,0006	0,0004	0,0004	0,0006	0,0004	0,0004	0,0004	0,841	0,8268	0,9114	
0,5	0,08	0,08	0,1	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0006	0,8364	0,8422	0,904	
0,4	0,08	0,1	0,06	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0006	0,8328	0,8192	0,9304	
0,5	0,08	0,1	0,1	0,28	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,7534	0,729	0,9668	
0,3	0,1	0,04	0,06	0,2	0,0006	0,0004	0,0004	0,0006	0,0004	0,0004	0,0004	0,8286	0,8374	0,9176	
0,4	0,1	0,04	0,1	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0004	0,8248	0,8142	0,9438	
0,4	0,1	0,06	0,06	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0004	0,8346	0,8344	0,9136	
0,5	0,1	0,06	0,1	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0006	0,8384	0,8252	0,9202	
0,5	0,1	0,08	0,06	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0004	0,8328	0,8192	0,9304	
0,5	0,1	0,08	0,1	0,24	0,0006	0,0006	0,0004	0,0006	0,0006	0,0004	0,0006	0,8296	0,8064	0,945	
0,5	0,1	0,1	0,1	0,3	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,5246	0,5148	0,6748	
0,4	0,06	0,04	0,06	0,2	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004				

Tablo 4.11. Neural network tahmin seti

ÇIKTI DEĞİŞKENLERİ												GİRDİ DEĞİŞKENLERİ		
O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	I1	I2	I3
1500	200	200	200	650	1	2	2	1	2	2	1	2529	2254	3413
1500	200	200	400	650	2	2	2	2	2	2	2	3654	3623	4955
1500	200	300	200	650	2	2	2	2	2	2	1	2602	2587	3345
1500	200	300	400	650	2	2	2	2	2	2	2	2565	2574	3369
1500	200	400	200	650	2	2	2	2	2	2	2	3688	3650	4880
2000	200	400	400	1000	3	2	2	3	2	2	2	4066	4106	4719
1500	200	500	200	650	2	2	2	2	2	2	2	3696	3634	4857
2000	200	500	400	1200	3	3	2	3	3	2	2	4157	4222	4601
1500	300	200	200	650	2	2	2	2	2	2	2	3721	3629	4801
1500	300	200	400	1000	2	2	2	2	2	2	2	4095	4074	4720
1500	300	300	200	650	2	2	2	2	2	2	2	3756	3741	4730
2000	300	300	500	1200	3	3	2	3	3	2	2	4176	4151	4565
2000	300	400	400	1000	3	2	2	3	2	2	2	4138	4041	4718
2000	300	500	300	1200	3	3	2	3	3	2	2	4127	4187	4583
2500	300	500	500	1200	3	3	2	3	3	2	3	4133	4114	4658
1500	400	200	300	1000	2	2	2	2	2	2	2	4135	4040	4739
2000	400	200	500	1000	2	2	2	2	2	2	2	4127	4048	4755
1500	400	300	300	650	2	2	2	2	2	2	2	2570	2566	3388
2000	400	300	500	1200	3	3	2	3	3	2	2	4182	4107	4596
2000	400	400	300	1000	3	2	2	3	2	2	2	4088	4130	4711
2500	400	400	500	1200	3	3	2	3	3	2	3	4147	4093	4679
2000	400	500	300	1200	3	3	2	3	3	2	3	4111	4124	4698
2500	400	500	500	1400	3	3	3	3	3	3	3	4058	4066	4781
1500	500	200	300	1000	3	2	2	3	2	2	2	4168	4054	4667
2000	500	200	500	1200	3	3	2	3	3	2	2	4127	4192	4583
2000	500	300	300	1200	3	3	2	3	3	2	2	4160	4189	4583
2500	500	300	500	1200	3	3	2	3	3	2	3	4125	4124	4669
2500	500	400	300	1200	3	3	2	3	3	2	2	4117	4098	4688
2500	500	400	500	1200	3	3	2	3	3	2	3	4177	4092	4679
2500	500	500	500	1500	3	3	3	3	3	3	3	4058	4066	4781
2000	300	300	300	1000	2	2	2	2	2	2	2	4119	4071	4740

4.3.4. Neural network kullanarak üretim parametrelerin tahmin edilmesi

Geliştirilen ANN modeli ile işletme çok kısa bir zamanda istediği üretime ulaşmak için ne kadar girdi kullanması gerektiğine karar verebilecektir. Neural Network modeline yapılan çok sayıda deneylerle parametreler öğretildikten sonra firma yöneticisi ne kadar ürün istediğini söyleyince değerler programa girilerek o dönemde ne kadar üretim girdisi kullanması gerektiği birkaç saniye içinde söylenebilecektir. Ürettiği üç ana ürün için gerekli kalıp sayılarına, değirmen kapasitesine, kurutma odası kapasitesine, sırlama bölümünde ne kadar eleman kullanacağına, rötuşlama bölümünde ne kadar eleman kullanacağına ve fırın sayısına hemen karar verebilecektir. Böylece işletmedeki deneme yanılmaya dayalı yada deneyimlere dayalı hammadde ve zaman kaybı indirgenmiş olacaktır. Bu parametreleri tahmin etmek işletmeye hem zaman hem de maliyet açısından büyük faydalar sağlayacaktır.

4.4. İstatistiksel Analiz

Her iki modeldeki üretim süreleri 590 saat, bir aylık iş gücüdür. Sistem bu süreden sonra dengeye ulaşmakta ve kararlı bir yapıya kavuşmaktadır. Modelin dengeye gelmesindeki koşum süresi yaklaşık 20 dakika ile 5 saat arasında sürmektedir. Farklı parametreler seçilerek 30 farklı benzetim deneyi sonuçları alınmıştır. Bu deney seti ise, birinci değirmenin kapasitesi, kalıp sayıları, kurutma odası kapasitesi, birinci ve ikinci fırın kapasiteleri için farklı değerler seçilerek oluşturulmuştur. Oluşturulan ANN modeli ile simülasyon sonuçlarının birbiriyle benzer olup olmadığını anlamak için istatistiksel analiz yapılmıştır.

4.4.1. İki modelin ortalamaları arasındaki farkın anlamlılık testi

Bu bölümde ANN modeli ile ProModel ile elde edilen deney sonuçlarının ortalamaları arasındaki istatistiki olarak anlamlı bir farkın olup olmadığının test edilmesidir. Bu testin amacı, iki modelin birbiriyle yaklaşık olarak aynı sonuçları verip vermediğini yani eşit ortalamalara sahip olup olmadığını araştırmaktır.

Bu testte ileri sürülecek olan hipotez iki modelin ortalamaları arasında bir fark olmadığı yani ortalamaların eşit olduğudur. Bunu H_0 hipotezi ile gösterebiliriz.

$$H_0 : \mu_1 = \mu_2$$

Alternatif hipotez ise, H_0 hipotezini reddi durumunda kabul edilecek olan ve ortalamaların eşit olmadığını gösteren H_1 hipotezidir.

$$H_1 : \mu_1 \neq \mu_2$$

H_1 karşıt hipotezinde ileri sürülen iddianın kabul edilip edilmeyeceğinin ortaya çıkartılması, hipotez testlerinin amacını oluşturmaktadır. n deney sayısını göstermek üzere $n \geq 30$ durumu söz konusu olduğundan uygulanacak test istatistiği

$$z = \frac{|\bar{X}_1 - \bar{X}_2|}{S_{\bar{X}_1 - \bar{X}_2}}$$

formülün paydasında yer alan, ortalamalar arasındaki farkın standart hatası aşağıdaki şekilde hesaplanır.

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

Burada;

$\bar{X}_1 = 1$. Modelin ortalamasını,

$\bar{X}_2 = 1$. Modelin ortalamasını,

$S_1 = 1$. Modelin standart sapmasını,

$S_2 = 2$. Modelin standart sapmasını,

$n_1 = 1$. Modelin deney sayısını,

$n_2 = 2$. Modelin deney sayısını

ifade etmektedir.

Burada, red bölgesi: $z > z_k$ şeklinde tanımlanır. $\alpha = 0.05$ anlamlılık düzeyine göre $z_k = 1.96$ 'dır. $z > z_k$ olduğunda H_0 hipotezi reddedilir. $z \leq z_k$ olduğunda ise reddedilemez. H_0 hipotezinin reddedilememesi, iki model ortalamaları arasındaki farkın, istatistiki olarak sıfırdan anlamlı bir farklılığa sahip olmadığı anlamına gelir.

Bu bilgiler ışığında, Neural ile ProModel sonuçları SPSS programında değerlendirilmiş ve her bir değişken için, iki model ortalaması arasında fark olup olmadığı test edilmiştir. Her bir değişken için hesaplanan test istatistikleri aşağıda toplu olarak verilmiştir.

Testte kullanmak için 30 neural network, 30 da ProModel programlarında denemeler yapılmıştır. Neural Network'de otuz deneyde programa istenilen üretim miktarları girilmiştir. Program üretim girdi parametreleri tahmin edilmiştir. Bulunan üretim girdi parametrelere değerleri yani Neural Network'un çıktıları ProModel'da girdi olarak kullanılmıştır ve otuz tane deney sonucunda üretim miktarları elde edilmiştir. SPSS programında bu iki modelde deneyler sonucunda bulunan değerlerinin birbirleriyle benzer olup olmadığı kontrol edilmiştir. H_0 hipotezinde iki modelin benzer olduğu kabul edilmiştir. H_1 hipotezinde ise iki modelin birbirinden farklı olduğu kabul edilmiştir.

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

$$\alpha = 0.05$$

Red bölgesi : $z > 1.96$

Tablo 4.13. z ve z_k değerlerinin karşılaştırılması

Değişkenler	z test istatistiği	z_k
Tuvalet taşı üretim miktarı	0.30	1.96
Lavabo üretim miktarı	0.15	1.96
Klozet üretim miktarı	0.27	1.96
Değirmen kapasitesi	0.09	1.96
Tuvalet taşı kalıp sayısı	0.02	1.96
Klozet kalıp sayısı	0.02	1.96
Lavabo kalıp sayısı	0.38	1.96
Kurutma odası kapasitesi	0.30	1.96

Görüldüğü gibi tüm değişkenler için hesaplana z istatistikleri z_k değerinden küçüktür. Bu nedenle, her bir değişken için H_0 hipotezi kabul edilir. Yani 1. model ile 2. model ortalamaları birbirinden farklı değildir. Bunun anlamı, örneğin; tuvalet taşı üretim miktarı 1. modelden elde edilen sonuç 2. modelden elde edilen sonuç arasında istatistiksel olarak sıfırdan anlamlı bir farklılık yoktur. Yani ortalamaları eşittir. Bütün değişkenler için durum aynıdır.

Tuvalet taşı üretim miktarı için istatistiki hesaplamalar:

Ttaşı2 için	Ortalama	= 3788.2
	Varyans	= 327325.2
	Standart Error	= 104.455

Ttaşı1 için Ortalama = 3832.0
 Standart Error = 104.557

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{104.455^2}{30} + \frac{104.557^2}{30}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = 26.983$$

$$t^* = \frac{|\bar{X}_1 - \bar{X}_2|}{S_{\bar{X}_1 - \bar{X}_2}} = \frac{43.8}{26.983} = 1.623$$

Varyansların eşitliği için Levene Testte : F =0.002 P=0.963

	t value	2.tail sig
Equal=	0.30	0.768

Klozet üretim miktarı için istatistiki hesaplamalar:

Klzl için Ortalama = 4494.6333
 Standart Error = 91.938

Klz2 için Ortalama = 4454.3667
 Standart Error = 120.277

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{91.938^2}{30} + \frac{120.277^2}{30}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = 27.639$$

$$t^* = \frac{|\bar{X}_1 - \bar{X}_2|}{S_{\bar{X}_1 - \bar{X}_2}} = \frac{40.2666}{27.639} = 1.457$$

Varyansların eşitliği için Levene Testte : F =2.030 P=0.160

	t value	2.tail sig
Equal=	0.27	0.791

Lavabo üretim miktarı için istatistiki hesaplamalar:

Lav1 için	Ortalama	= 3799.2333
	Standart Error	= 103.325

Kl2 için	Ortalama	= 3820.6
	Standart Error	= 95.008

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{103.325^2}{30} + \frac{95.008^2}{30}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = 25.627$$

$$t^* = \frac{|\bar{X}_1 - \bar{X}_2|}{S_{\bar{X}_1 - \bar{X}_2}} = \frac{-21.3667}{25.627} = 0.834$$

Varyansların eşitliği için Levene Testte : F =2.030 P=0.160

	t value	2.tail sig
Equal=	-015	0.880

Değirmen için istatistiki hesaplar:

Deg1 için	Ortalama	= 1943.5967
	Standart Error	= 49.748

Deg2 için	Ortalama	= 1937.3333
	Standart Error	= 52.178

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{49.5967^2}{30} + \frac{52.178^2}{30}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = 13.165$$

$$t^* = \frac{|\bar{X}_1 - \bar{X}_2|}{S_{\bar{X}_1 - \bar{X}_2}} = \frac{6.2634}{13.165} = 0.478$$

Varyansların eşitliği için Levene Testte : F =0.124 P=0.726

t value	2.tail sig
---------	------------

Equal= 0.09 0.931

Tuvalet taşı kalıp sayıları için istatistiki hesaplamalar:

TKalıp1 için Ortalama = 348.1067
Standart Error = 11.120

TKalıp2 için Ortalama = 348.1667
Standart Error = 11.139

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{11.120^2}{30} + \frac{11.139^2}{30}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = 2.874$$

$$t^* = \frac{|\bar{X}_1 - \bar{X}_2|}{S_{\bar{X}_1 - \bar{X}_2}} = \frac{|-0.06|}{2.874} = 0.021$$

Varyansların eşitliği için Levene Testte : F =0.000 P=0.986

t value 2.tail sig
Equal= 0.00 0.997

Lavabo kalıp sayıları için istatistiki hesaplamalar:

LKalıp1 için Ortalama = 337.7133
 Standart Error = 8.528

LKalıp2 için Ortalama = 341.1
 Standart Error = 7.963

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = \sqrt{\frac{8.528^2}{30} + \frac{7.963^2}{30}}$$

$$S_{\bar{X}_1 - \bar{X}_2} = 1.614$$

$$t^* = \frac{|\bar{X}_1 - \bar{X}_2|}{S_{\bar{X}_1 - \bar{X}_2}} = \frac{|-3.3867|}{1.614} = 1.59$$

Varyansların eşitliği için Levene Testte : F =0.539 P=0.466

t value 2.tail sig

Equal= -0.38 0.703

Klozet kalıp sayıları için istatistiki hesaplamalar:

KKalıp1 için Ortalama = 367.62
 Standart Error = 11.746

$$S_{\bar{X}_1 - \bar{X}_2} = 10.671$$

$$t^* = \frac{|\bar{X}_1 - \bar{X}_2|}{S_{\bar{X}_1 - \bar{X}_2}} = \frac{|-17.4566|}{10.671} = 1.636$$

Varyansların eşitliği için Levene Testte : F =0.179 P=0.673

	t value	2.tail sig
Equal=	-0.30	0.766

BÖLÜM 5. SONUÇLARIN YORUMLANMASI

Üretimde kullanılan makine, araç, gereç, yöntem ve üretim tekniklerinin tümü birden üretim teknolojisini oluşturur. Buharlı makinelerin kullanılmaya başlandığı XIX. yüzyılın başlangıcından bu yana üretim sürecinde köklü bir değişim ve hızlı bir gelişim sağlanmıştır. Teknolojinin gelişmesiyle birlikte daha önceleri insan emeği ile yapılan birçok ürün artık makineler tarafından daha kolay, daha sağlam, daha basit ve ergonomik bir şekilde üretilmeye başlanmıştır.

Neural Network ile üretim parametrelerini tahmin ederken dört adım takip edilmiştir. Birincisi, işletmenin üretim prosesinin ProModel simülasyon paket programı ile modellenmesidir. İkincisi, farklı üretim parametreleri ile simülasyon sonuçlarının elde edilmesidir. Üçüncüsü, simülasyon sonuçlarından oluşturulan eğitim datasıyla Neural Network geliştirilmesidir. Son olarakta Neural Network kullanılarak ürünlerin talep miktarlarıyla üretim parametrelerinin tahmin edilmesidir.

Son zamanlarda ülkemizde ProModel paket programının kullanımı giderek artmaktadır. ProModel paket programı ile modellenen herhangi bir problem için yerleşimlere ait çıktı miktarları, bekleme zamanları, boş zamanlar, mevcut kapasitenin yüzde kaçından yararlanıldığı, operasyon zamanları gibi sonuçlara bakılarak değerlendirilebilir ve istenen sonuçlara ulaşabilmek için modelleme elemanlarına ait veriler değiştirilebilir ve her istenen anda yapılan düzenleme ile ilgili sonuçlar ekranda görülebilir, çıktı alınabilir. Bir üretim sisteminde saatler, haftalar hatta aylar kadar sürebilecek bir modelin simülasyonu ProModel paket programı yardımıyla fazla bir maliyete katlanmadan kolayca kurulabilmekte ve sonuçları görülebilmektedir.

Location bölümünde yerleşim içindeki makinelerimiz ve kapasiteleri ve kullanım miktarları görülebilir. Çıktı ile makinelerin ne kadar süre çalıştığı ve ne kadar süre boş kaldıkları görülür. Hazırlık ve bozulma ihtimalleri verilmediğinden bu değerler sürekli sıfır çıkmaktadır.

Seramik alanında üretim yapan Çelvit AŞ'nin şu andaki mevcut durumu ProModel paket programı ile simüle edilmiş ve istatistikleri incelenmiştir. Modelin bir aylık simülasyonu sonucunda işletmenin şu andaki üretim miktarları elde edilmiştir. İstatistik sonuçları incelendikten sonra nerelerde dar boğaz olduğu, beklemlerin nerelerde olduğu, toplam çıktı miktarları, hatalı ürün oranları gibi istatistikleri görebiliyoruz. Şu anda işletmede kurutma odası önünde ve fırının önünde beklemler meydana gelmekte. Bunun en büyük sebeplerinden biri kapasitelerin yetersiz olmasıdır. Günlük üretimi tamamen içeriye alabilecek kurutma kapasitesi olmaması beklemleri artırmaktadır. Geliştirilen alternatif çözüm ile kurutmanın kapasitesi artırılmış ve günlük üretimi tamamen içeri alarak beklemler engellenmeye çalışılmıştır. Kurutma odasından çıkan parça sayısı artırılmış ve bu sayede rötuşa ulaşan parça sayısı artmıştır. Birinci durumda aynı süreli bir yanma ile altı yüz elli parça içeri alınabilirken alternatif çözüm ile içeriye bin parça alınmaktadır. Bu da doğal olarak üretimde artışlara neden olmaktadır. Üstelik kurutma odasının kapasitesini artırmak işletmeye fazla bir maliyet getirmemektedir.

İkinci olarak fırın önünde meydana gelen beklemleri azaltmak için yapılan çalışmadır. İlk durumda fırının kapasitesi üç yüz birim ve günlük çıkarılan üretimi pişirecek kapasitede değildi. Bu nedenle fırın iki günde üç kere yanarak ürünleri içeri almaktadır. Fırın sürekli üretimde olmakta ama yine de önünde beklemler meydana gelmektedir. Alternatif çözüm olarak fırın kapasitesi artırma yoluna gidilmiştir. Bunun için ikinci bir fırın eklenmiştir. Bu yeni fırının kapasitesi iki yüz elli olarak belirlenmiştir. Bu sayede birinci fırının yoğunluğu azalmıştır. Bu iki fırın ile beklemlerde azalmalar olmuştur. Bir aylık simülasyon sonunda üretim miktarları artırılmıştır.

Kaynaklar bölümünde kullandığımız operatörlerin ne kadar süreyle kullanıldığı, ortalama kullanım oranı ve operatörden yararlanma miktarımız karşımıza gelmiştir. Geliştirilen alternatif çözüm ile operatörlerden daha fazla yararlanılmaktadır. Operatörlerin aylak zamanlarında azalmalar olmuştur. Yine aynı şekilde operatörlerin beklemelelerinde de azalmalar olmuştur.

Modelde hiç fire olmamıştır, bunun sebebi ise bozuk olan ürünlerin tekrar değirmene alınarak işlenip tekrar üretime gönderilmesidir.

Mevcut durum ile aylık üretim kapasitesi sekiz bin ile on bin arasında değişirken, geliştirilen durum ile aylık üretim kapasitesi on iki bin ile on dört bin arasında olmuştur. Alternatif olarak geliştirilen bu durumdan başka alternatifler de olabilir.

İstatistik sonuçlarında nerede dar boğazlar olduğu görülebilir. Daha sonraki işlem ise bu dar boğazların nasıl giderileceğine çalışmaktır. Ele alınan modelde dar boğazların kurutma odasında ve fırında olduğunu gördük ve kurutma odasının kapasitesini artırıp, yeni bir fırın ekleyerek bu dar boğazları gidermeye çalıştık. Modeldeki kalıp tezgahlarının verimleri farklı olarak çıkmaktadır. Bunun sebebi ise kapasitelerinin farklı olmasıdır.

Bir üretim sisteminde saatler, haftalar hatta aylar kadar sürebilecek bir modelin simülasyonu ProModel paket programı yardımıyla fazla bir maliyete katlanmadan kolayca kurulabilmekte ve sonuçları görülebilmektedir.

Önerilen alternatif model firma yönetimi tarafından uygun görülerek gerekli değişiklikler yapılmış ve benzetim senaryoları ile bulunan üretim sonuçları fiili üretim değerleri ile gerçekleştirilmiştir.

Fakat ProModel’da yaptığımız bu programda bazı durumlarda üretimi simüle ederken beş saat sürebiliyordu. Firma yöneticisi “bu ay şu kadar ürün istiyorum ne kadar hammadde, ne kadar eleman, ne kadar kalıp kullanmam gerekiyor” dediğinde o üretimi veren girdilere ulaşana kadar çok fazla sayıda simülasyon çalışması

yapılması gerekiyordu. Bu deneme çalışmaları da otuz dakika ile beş saat arasında değişen simülasyon zamanları istiyordu. Bu da bizim için çok fazla zaman kaybıydı. En önemli dezavantajı ise istenilen üretim miktarını bulmak için sistemi bir kere simüle etmenin yetersiz olmasıdır. İstenilen sonuca ulaşmak için çok fazla deneme yapmak gerekmektedir. Buda çok fazla zaman anlamına gelmektedir. Buradaki zaman kaybını engellemek amacıyla ANN modeli oluşturulmuştur. ANN yaklaşımının ana fikri insan beyni fonksiyonlarına benzemektir. Bu nedenle ANN çabuk yanıt verir ve sıralı dijital bilgisayarlara göre performansı daha yüksektir. Neural Network'e programı öğretmek zaman problemini ortadan kaldırabilirdi. Bu sebeple modelin sonuçlarını girdi olarak kullanıp asıl girdileri çıktı olarak verecek bir programla Neural Network'e öğretildi. Bu sayede çok kısa zamanda sonuca ulaşılabilmekteydi.

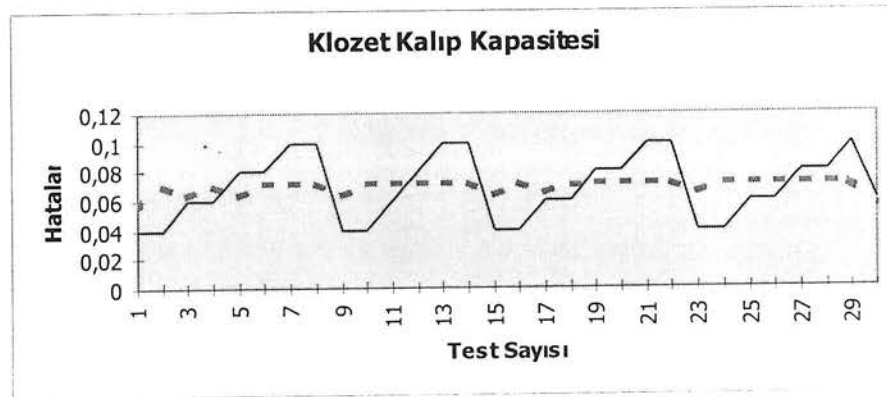
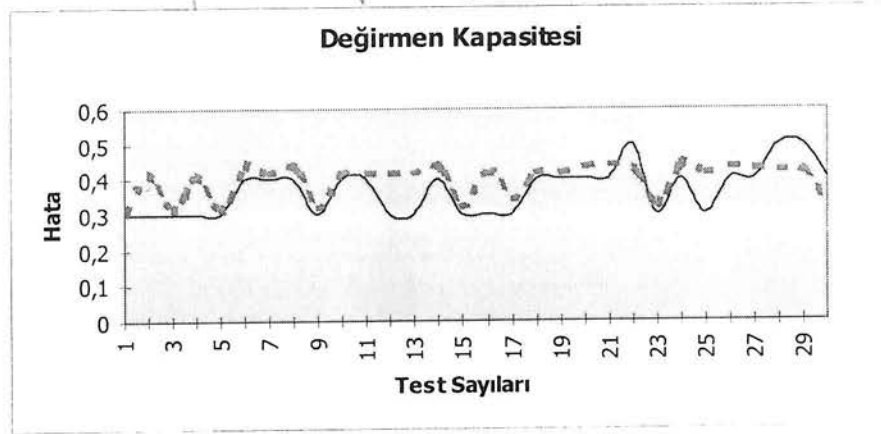
ANN oluşturulurken ProModel sonuçlarında faydalanılmıştır. Bunun için sistem farklı parametrelerle 61 kere simüle edilmiştir. Bu denemeler tablo 4.7 de gösterilmiştir. Bu 61 denemeden rasgele seçilen 31 tanesi eğitim seti olarak kullanılmıştır. Bunlar tablo 4.9'da gösterilmiştir. Geri kalan 30 deneme ise test aşaması için kullanılmıştır. Neural Network'de aktivasyon (sigmoid) fonksiyonu 0 ve 1 aralığındadır. Fakat modelin simülasyon sonuçları 0 ve 5000 aralığındadır. Bu nedenle simülasyon sonuçları 0 ve 1'e indirgenmiştir. Eğitim setinin normalize edilmiş şekli tablo 4.10 verilmiştir. Test aşaması için kullanılan simülasyon sonuçları tablo 4.11 deki gibidir. Bu değerlerin normalize edilmiş şekli tablo 4.12'de verilmiştir.

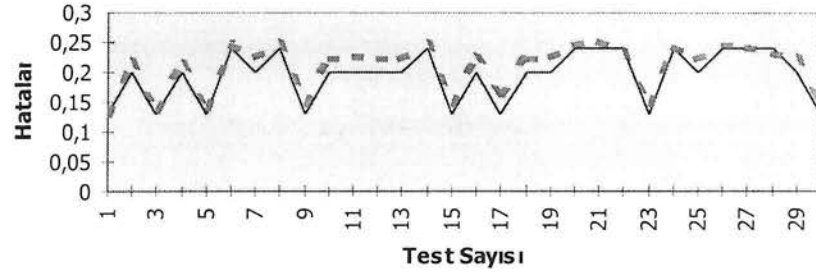
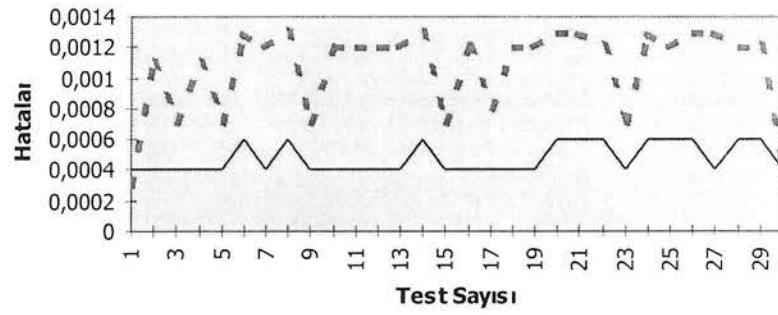
Kullanılan algoritmada üretim miktarları giriş değerleridir. Aktivasyon fonksiyonu olarak yukarıda bahsedilen sigmoid fonksiyonu kullanılmış ve çıkış değerleri de değirmen kapasitesi, tuvalet taşı kalıpları, lavabo kalıpları, klozet taşı kalıpları, kurutma odası kapasitesi, sırlama ve rötuş bölümündeki elemanların sayısı ve fırın adedidir.

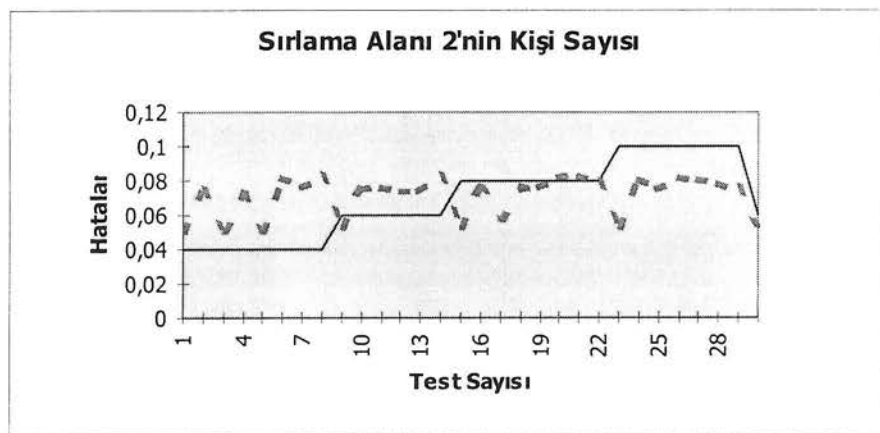
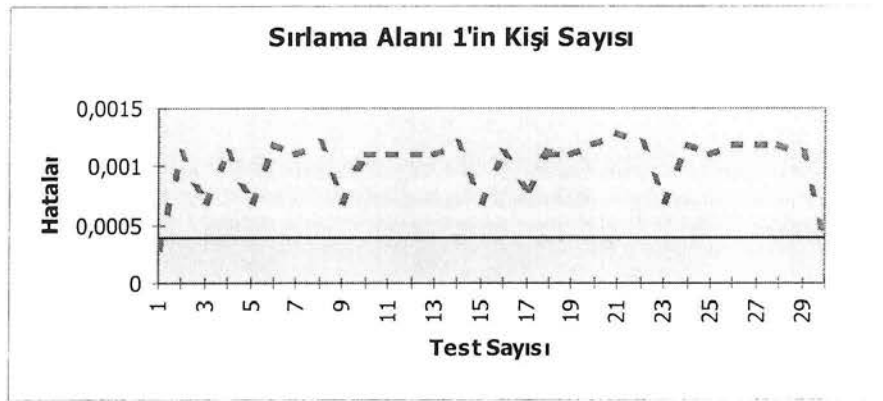
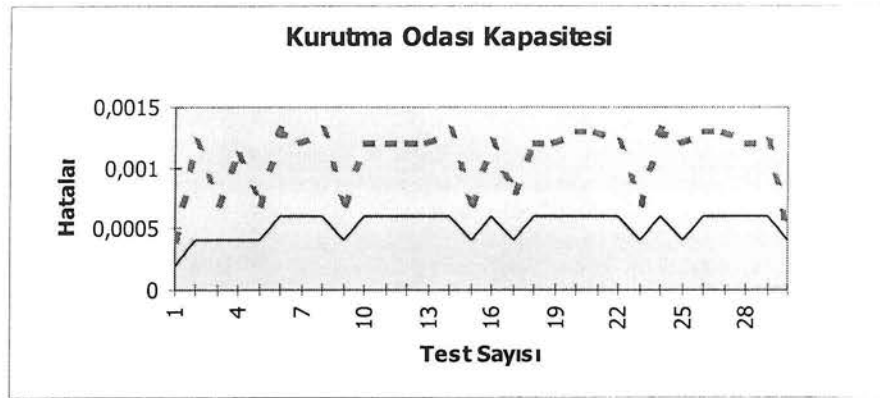
Neural Network'de en iyi sonuca ulaşmak için momentum katsayısı 0.8 ve öğrenme oranı olarak 0.2 alınmıştır. Ortalama hata 0.02'den küçük oluncaya kadar Neural

Network eğitim prosesinde iterasyonlara devam edilmiştir. Eğitim prosesi bir milyon adımda tamamlanmıştır. Neural Network eğitimi tamamlandıktan sonra değerlerin bazıları tahmin edildi. Tahmin değerleri gerçek simülasyon değerleriyle karşılaştırıldı. Neural Network sonuçları gösteriyor ki ağ % 90 başarılı olmuştur. Hata değerleri minimuma yaklaştığında ANN parametreleri saklandı ve sistem parametrelerinin tahmini için kullanıldı.

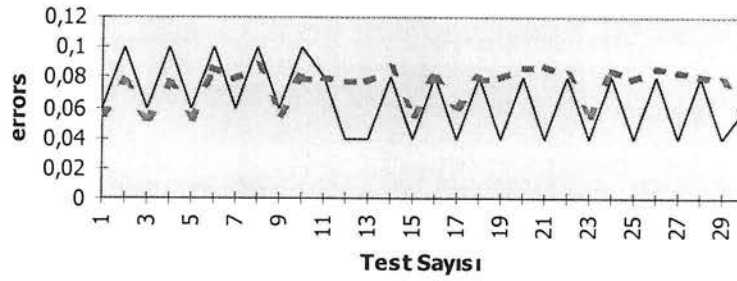
ProModel ve Neural Network'ten elde edilen sonuçlar grafik olarak şu şekilde gösterilebilir.



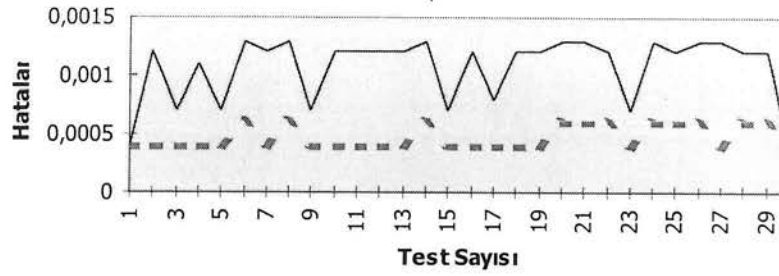
Tuvalet Taşı Kapasitesi**Lavabo Kapasitesi**



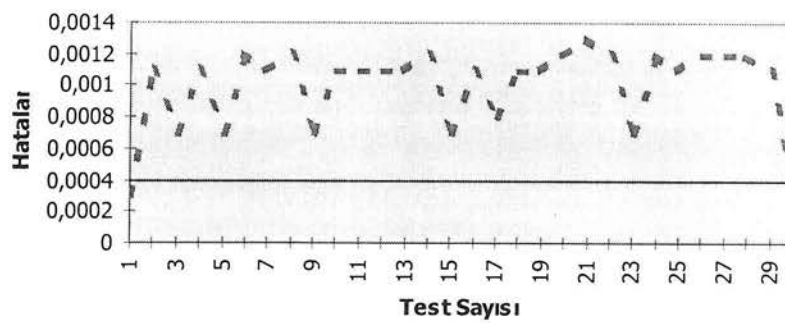
Sırlama Alanı 3'ün Kişi Sayısı

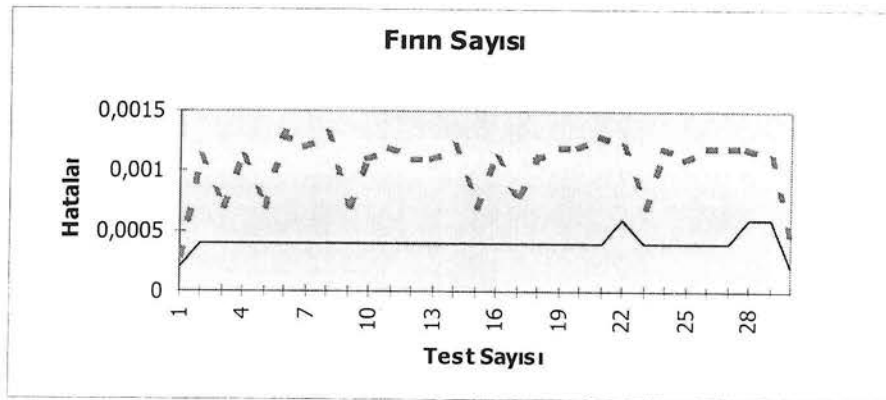
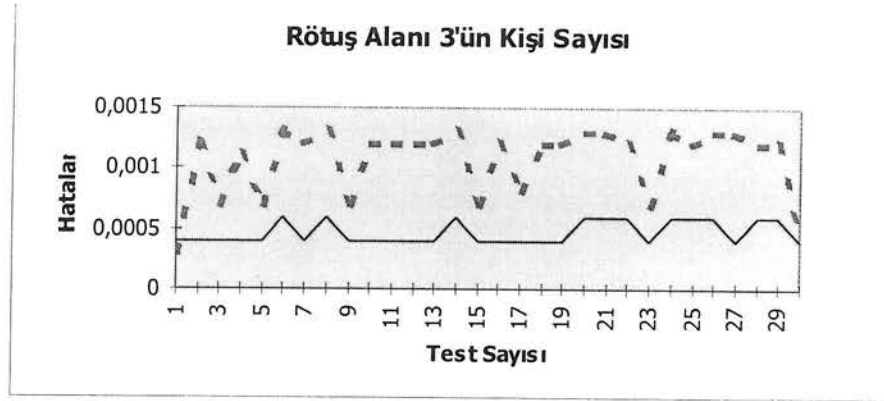


Rötuş Alanı 1'in Kişi Sayısı

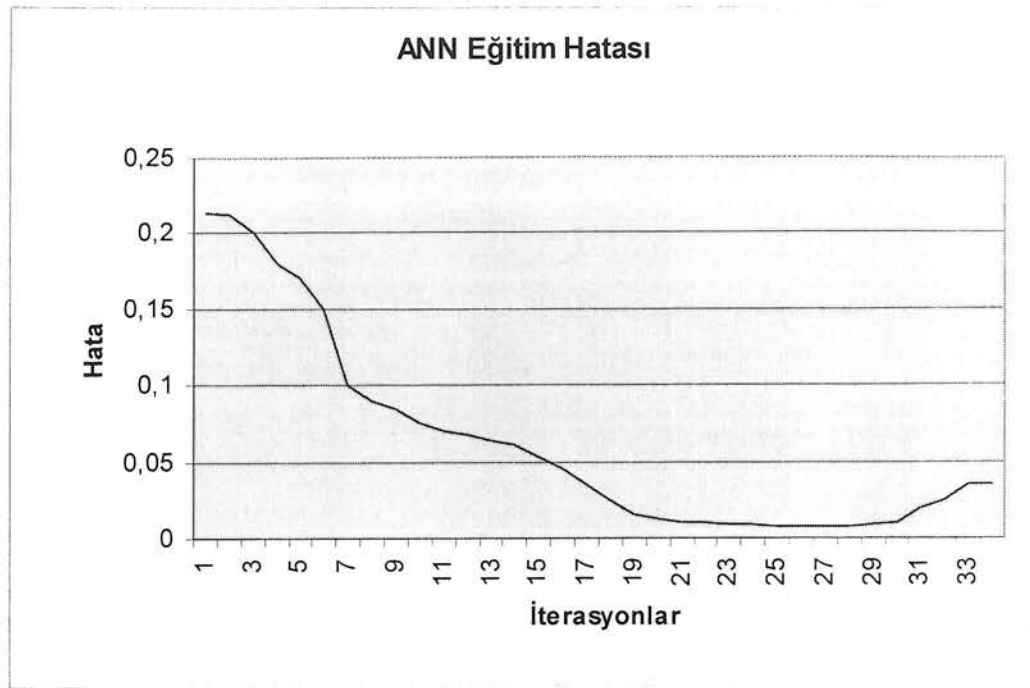


Rötuş Alanı 2'nin Kişi Sayısı





Grafik 5.1. Çıktı parametrelerini tahmini ile simülasyon değerlerinin hata grafikleri



Grafik 5.2. ANN eğitiminin hata grafiği

Oluşturulan ANN modeli ile simülasyon sonuçlarının birbirleriyle benzer olup olmadığını anlamak için istatistiksel analiz yapılmıştır.

Bu bölümde ANN modeli ile ProModel ile elde edilen deney sonuçlarının ortalamaları arasındaki istatistiki olarak anlamlı bir farkın olup olmadığının test edilmiştir. Bu testin amacı, iki modelin birbirleriyle yaklaşık olarak aynı sonuçları verip vermediğini yani eşit ortalamalara sahip olup olmadığını araştırmaktır.

Bu testte ileri sürülecek olan hipotez iki modelin ortalamaları arasında bir fark olmadığı yani ortalamaların eşit olduğudur. Bunu H_0 hipotezi ile gösterebiliriz.

$$H_0 : \mu_1 = \mu_2$$

Alternatif hipotez ise, H_0 hipotezini reddi durumunda kabul edilecek olan ve ortalamaların eşit olmadığını gösteren H_1 hipotezidir.

$$H_1 : \mu_1 \neq \mu_2$$

Burada, red bölgesi: $z > z_k$ şeklinde tanımlanır. $\alpha = 0.05$ anlamlılık düzeyine göre $z_k = 1.96$ 'dır. $z > z_k$ olduğunda H_0 hipotezi reddedilir. $z \leq z_k$ olduğunda ise reddedilemez. H_0 hipotezinin reddedilememesi, iki model ortalamaları arasındaki farkın, istatistiki olarak sıfırdan anlamlı bir farklılığa sahip olmadığı anlamına gelir.

Bu bilgiler ışığında, Neural ile ProModel sonuçları SPSS programında değerlendirilmiş ve her bir değişken için, iki model ortalaması arasında fark olup olmadığı test edilmiştir.

Testte kullanmak için 30 neural network, 30 da ProModel programlarında denemeler yapılmıştır. Neural Network'de otuz deneyde programa istenilen üretim miktarları girilmiştir. Program üretim girdi parametreleri tahmin edilmiştir. Bulunan üretim girdi parametrelere değerleri yani Neural Network'un çıktıları ProModel'da girdi olarak kullanılmıştır ve otuz tane deney sonucunda üretim miktarları elde edilmiştir. SPSS programında bu iki modelde deneyler sonucunda bulunan değerlerinin birbirleriyle benzer olup olmadığı kontrol edilmiştir. H_0 hipotezinde iki modelin benzer olduğu kabul edilmiştir. H_1 hipotezinde ise iki modelin birbirinden farklı olduğu kabul edilmiştir.

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

$$\alpha = 0.05$$

Red bölgesi : $z > 1.96$

Değişkenler	z test istatistiği	z_k
Tuvalet taşı üretim miktarı	0.30	1.96
Lavabo üretim miktarı	0.15	1.96
Klozet üretim miktarı	0.27	1.96
Değirmen kapasitesi	0.09	1.96
Tuvalet taşı kalıp sayısı	0.02	1.96
Klozet kalıp sayısı	0.02	1.96
Lavabo kalıp sayısı	0.38	1.96
Kurutma odası kapasitesi	0.30	1.96

Görüldüğü gibi tüm değişkenler için hesaplanan z istatistikleri z_k değerinden küçüktür. Bu nedenle, her bir değişken için H_0 hipotezi kabul edilir. Yani 1. model ile 2. model ortalamaları birbirinden farklı değildir. Bunun anlamı, örneğin; tuvalet taşı üretim miktarı 1. modelden elde edilen sonuç 2. modelden elde edilen sonuç arasında istatistiksel olarak sıfırdan anlamlı bir farklılık yoktur. Yani ortalamaları eşittir. Bütün değişkenler için durum aynıdır.

İki modelin sonuçları birbirleriyle aynıdır, neural network modeli bize zamandan kazanç sağlamıştır.

KAYNAKLAR

- [1] ABULAFYA N., "Neural Networks For System Identification and Control" MSc Thesis University of London 1995.
- [2] ARER, M., M., ÖZDEMİREL N.E., Simulation of Capacity Expansion and Sequencing Alternatives for a Make-to-order Sheet Metal producer.11 The European Simulation Multiconferance, June 1-4,1997, Türkiye.
- [3] ASKIN Ronald G., STANDRIGE Charles R., "Modelling and Analysis of Manufacturing Systems", John Wiley and Sons, 1993
- [4] AYDIN, M.E., ÖZKAN, Y., "Dağılım Türünün Belirlenmesinde YSA'nın Kullanılması", 1. Ulusal ZİS Sempozyumu Bildiriler Kitabı , s. 176-184, Mayıs 1996
- [5] BAYRAKTAR, E., Karmaşık Endüstriyel Sistemlerin Tasarım ve Analizi İçin Simülasyon, Makine Magazin, 1997.
- [6] BEALE R. and JACKSON T., " Neural Computing : An Introduction", Adam Hilger, Bristol, Philadelphia and New York, 1990
- [7] CAUDİLL Mauren and BUTLER Charles, Understanding Neural Networks Computer Explorations Vol. 1, The MIT Press, 1992
- [8] CHECKLAND P.B., System Thinking, System Science,John Willey &Sons, New York, 1990

- [9] DAN, W. F., " Introduction to Artificial Intelligence and Expert System ", Prentice Hall, pp 1-9, New Jersey, 1990
- [10] DAİM,T.,Simülasyon Proje Yönetimi, CAD İşletmesi, Şubat 1994.
- [11] DOMENİCO, F, FELİCE, F., 1997, Planning of Automated Assembly Line by Means of Simulation Techniques in an Electromechanical Company. 11 The European Simulation Multiconference, June 1-4,1997, Türkiye
- [12] ERKUT, H., Yönetimde Simülasyon Yaklaşımı, İrfan Yayıncılık ve Tic., 1992.
- [13] ELLACOTT Stephan W., NASON John C., and ANDERSON Lain J., "Mathematics of Neural Networks", Kluwer Academic Publishers, 1997
- [14] ERDEM H. İbrahim, ÖNÜT Semih, DEMİREL Tufan, GÜNAY Güney, "Bilgisayar Destekli Mühendislik Sistemlerinin Yapısal Analizi, Planlanması ve Geliştirilmesi", Verimlilik dergisi, 45-57
- [15] ELZAS; Simulation and Model-Based Methodologies: An Interagtive View,pp.3-40,Sprinter-Verlag, New York.Interactive,
- [16] FREEMAN James A., SKAPURA David M. "Neural Networks Algorithms, Applications and Programming Techniques", Addison-Wesley Publishing Company,1992
- [17] FİSHWICK, P, A., Simulation Model Design and Execution, Prentice Hall, Inc. Englewood Cliff, New Jersey, 1995
- [18] GRANİNO, A.Korn, Interactive Dynamis System Simulation,McGraw-Hill, New York, 1998.

- [19] GEVARTEL, W., B., " Intelligent Machines ", New Jersey Prentice Hall, 1985
- [20] GUPTA Yash P. And UDO J. Godwin. "Application of Neural Networks in Manufacturing Management Systems", Production Planning and Control, Vol:5, No:3, pp 258 – 270, 1994
- [21] GÜLSEÇEN S., "Yapay Sinir Ağları, İşletme Alanında uygulaması ve Bir örnek Çalışma", Doktora Tezi i.ü. İşletme Fakültesi, İstanbul 1993
- [22] GÜRGÜN Burçkan, "Scheduling With Artificial Neural Networks", A Thesis Submitted to the Department of Industrial Engineering and the Institute of Engineering and Sciences of BILKENT UNIVERSITY in Partial Fulfilment of the Requirements for the Degree of Master of Science, June 1993,
- [23] HALAÇ, O., İşletmelerde Simülasyon Teknikleri, İ.Ü. İşletme Fakültesi Yayını, İstanbul, 1992.
- [24] Introduction to Simulation with SIMAN, System Modeling Corp., USA, 1986.
- [25] JAIKUMAR Ramchandran, "Post-industrial Manufacturing", Harward Business Review, November – December, pp 69-76, 1986
- [26] KAYA S. "Yapay Sinir Ağları ile Ünite Planlaması ve Paylaşımı", Y.Lisans, Sakarya Üniversitesi, Kasım 1997
- [27] KHANNA Tarun, " Foundations of Neural Networks", Addison – Wesley Publishing Company, 1990
- [28] KINEMIA J. G. and GERSHWIN S.B., "Flow Optimisation in FMS", International Journal of Production Research, Vol 23, pp 81-96, 1985

- [29] KONAR, Ferit, "Neural Network Applications", Ders Notları, Sakarya, 1996
- [30] LAW, A., KELTON, W.D., 1991, Simulation Modeling and Analysis. McGraw-Hill International Editions.
- [31] LIPPMAN Richard P., " An Introduction to Computing With Neural Networks", IEEE ASSP Magazine, April , pp 4-20, 1987
- [32] LISBOA, P., G., J., Neural Networks Current Applications, Chagoman and Hall, 1992
- [33] LIVOPMAN Richard P., An Introduction to Computing with Neural Networks, IEEE ASSP Magazine April 1987
- [34] NAHMIAS Steven, " Production and Operations Analysis", Richard D. Irwin, inc. 1989 and 1993
- [35] NARENDRA, K.S. "Adaptive Control Using Neural Networks" in Neural Networks for Control (W.T. Miller, R.S. Sutton, and P.J. Werbos, edds). 3 rd printing 1992 MIT.
- [36] Neural networks Algorithms, Applications and Programming Techniques, James A. Freeman, David M. Skopura, Addison-wesley Publishing Company, 1991
- [37] NAYLOR, T.H., BALINTFY, J.L., BURDICK, D.S., CHU, K., Computer Simulation Tecniques, John Wiley and Sons, 1966.
- [38] ÖREN T,I, Model-Based Activites:A Paradigm Shift. Editors, T,I, Ören,B.P. Zeigler and M.S,1984

- [39] PEDGEN, C.D., SHANNON, R.E., SADOWSKI, R. P, Introduction to Simulation Using SIMAN, McGraw-Hill, New York, 1990
- [40] PEDGEN, C.D., Introduction to SIMAN with Version Enhancements, Systems Modeling Corp., Pennsylvania, 1986.
- [41] PEDGEN, C.D., SHANNON, R. E., SADOWSKI, R. P., Introduction to Simulation Using SIMAN, McGraw-Hill, New York, 1990.
- [42] PHILIP, D.W., Neural Computing Theory and Practice, New York, 1990
- [43] PIPLANI R. And TALAVAGE , ' Launching and Dispatching strategies multi-criteria control of closed manufacturing systems with flexible using capability', Int. J. Prod. Res., Vol 33, No: 8, pp 2181 – 2196, 1995
- [44] Promodel Training Work book, Introductory Course notes and modeling exercises, 1995.
- [45] Promodel Ders Notları, 1997-1998
- [46] RICHARD, P. Lippman, "An Introduction to Computing with Neural Nets" IEEE ASSP, pp.4-22, 1987
- [47] RONALD G. Askin and CHARLES R. Standridge Modelling and Analysis of Manufacturing Systems, John Wiley and Sons, 1993
- [48] SABUNCUOĞLU İhsan, "Scheduling with Neural Networks : a review of the literature and new research directions Production Planning and Control, Vol 9, No:1, pp 2-12, 1998

- [49] SEVİNÇ S., Learning events from simulation Model runs: From Procedurel Models to Relational Abstractions, International Journal of General Systems,19(32), s337-355, 1991.
- [50] TÜRKER, A. K.,Atelye Tipi Üretimde Maliyet Optimizasyonunun Simülasyon Yoluyla Sağlanması, Y. Lisans Tezi (Yayınlanmamıştır), Yıldız Teknik Üniversitesi, 1991.
- [51] TURBAN, E., " Decision Support and Expert Systems Management Support System ", Prentice Hall, 1992
- [52] VONDERSPEK Peter G., "Planning for Factory Automation", McGraw – Hill, 1993
- [53] WINSTANLEY, Graham, " Artificial Intelligence in Engineering ", Courier Int. Ltd., England, 1991

ÖZGEÇMİŞ

1975 yılında Kayseri’de doğdu. İlk, orta, lise eğitimini Kayseri’de çeşitli okullarda tamamladı. 1993 yılında Erciyes Üniversitesi Bilgisayar Programcılığı Bölümünü kazandı. 1995 yılında mezun olduktan sonra, dikey geçişle Sakarya Üniversitesi Endüstri Mühendisliği bölümünde okumaya hak kazandı. 1998 yılında bölüm birincisi olarak mezun oldu. 1998-1999 eğitim yılında Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümünde yüksek lisansa başladı.