

**T.C**  
**SAKARYA ÜNİVERSİTESİ**  
**SOSYAL BİLİMLER ENSTİTÜSÜ**

**YAPAY SİNİR AĞLARI İLE AFET YÖNETİMİNDE**  
**SOSYAL ZARAR GÖREBİLİRLİK RİSKİNİN**  
**BELİRLENMESİ**

**YÜKSEK LİSANS TEZİ**

**Dilek SÜRMEİ**

**Enstitü Anabilim Dalı : İşletme**  
**Enstitü Bilim Dalı : Üretim Yönetimi ve Pazarlama**

**Tez Danışmanı: Prof. Dr. Erman COŞKUN**

**TEMMUZ - 2011**

T.C  
SAKARYA ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ

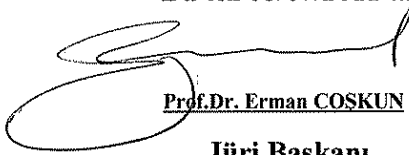
YAPAY SİNİR AĞLARI İLE AFET YÖNETİMİNDE  
SOSYAL ZARAR GÖREBİLİRLİK RİSKİNİN  
BELİRLENMESİ

YÜKSEK LİSANS TEZİ

Dilek SÜRMEİ

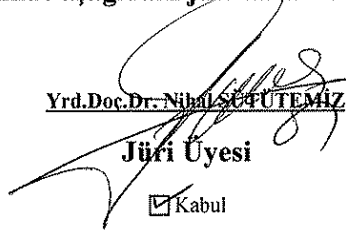
Enstitü Anabilim Dalı : İşletme  
Enstitü Bilim Dalı : Üretim Yönetimi ve Pazarlama

Bu tez 08/07/2011 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

  
Prof. Dr. Erman COSKUN

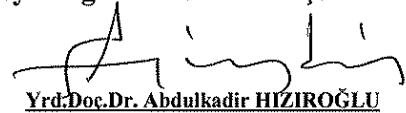
Jüri Başkanı

- Kabul  
 Red  
 Düzeltme

  
Yrd. Doç. Dr. Nihal SÖĞÜTMEZ

Jüri Üyesi

- Kabul  
 Red  
 Düzeltme

  
Yrd. Doç. Dr. Abdulkadir HIZIROĞLU

Jüri Üyesi

- Kabul  
 Red  
 Düzeltme

## **BEYAN**

Bu tezin yazılmasında bilimsel ahlak kurallarına uyulduđunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduđunu, kullanılan verilerde herhangi bir tahrifat yapılmadıđını, tezin herhangi bir kısmının bu üniversite veya başka bir üniversite başka bir tez çalışması olarak sunulmadıđını beyan ederim.

**Dilek SÜRMEİİ**  
**08/07/2011**

## ÖNSÖZ

“Yapay Sinir Ağları İle Afet Yönetiminde Sosyal Zarar Görebilirlik Riskinin Belirlenmesi” konusu üzerine yapılan bu çalışmada, Yapay Zeka'nın altında Yapay Sinir Ağları ilkelerine yer verilmiştir. Uygulamada ise bu ilkeler kullanılarak SPSS Clementine ve MATLAB programı aracılığıyla sosyal faktörler ve sosyal sınıflar arasındaki ilişki incelenmiştir.

Bu çalışmanın hazırlanmasında, değerli görüşlerinden yararlandığım tez danışmanım Prof. Dr. Erman COŞKUN'a içtenlikle teşekkür ederim.

Desteğini esirgemeyip yol almamdaki yardımları için Yrd. Doç. Dr. Nihal SÜTÜTEMİZ'e, Yrd. Doç. Dr. Abdulkadir HIZIROĞLU'na, Arş. Gör. Ahmet KÜÇÜKER'e, Arş. Gör. Ömer Faruk SEYMEN'e ve Arş. Gör. Ulaş YURTSEVER'e teşekkürlerimi bir borç bilirim.

Bugünlere gelmemde büyük katkıları olan, bana olan inancını daima hissettiren ve kararlarımın her aşamasında desteğini esirgemeyen değerli hocam Fulya TEKELİOĞLU'na teşekkürlerimi sunarım.

Ayrıca tezin hazırlanması aşamasında maddi ve manevi desteğini her zaman yanımda hissettiğim Yrd. Doç. Dr. Nevran KARACA'ya, aileme ve arkadaşlarıma teşekkür ederim.

**Dilek SÜRMELİ**  
**08/07/2011**

## İÇİNDEKİLER

<b>KISALTMALAR</b> .....	<b>iii</b>
<b>TABLO LİSTESİ</b> .....	<b>iv</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>v</b>
<b>ÖZET</b> .....	<b>vii</b>
<b>SUMMARY</b> .....	<b>viii</b>
<b>GİRİŞ</b> .....	<b>1</b>
<b>BÖLÜM 1: YAPAY ZEKA</b> .....	<b>5</b>
1.1. Yapay Zeka Kavramı .....	5
1.2. Yapay Zeka'nın Tarihçesi .....	6
1.3. Yapay Zeka'nın Avantajları ve Dezavantajları .....	7
1.4. Yapay Zeka Teknikleri.....	8
1.4.1. Genetik Algoritmalar .....	9
1.4.2. Tabu Araştırma Algoritması .....	9
1.4.3. Isıl İşlem Algoritması .....	10
1.4.4. Karınca Koloni Algoritması.....	10
1.4.5. Bağışıklık Sistemi Algoritması .....	11
1.4.6. Uzman Sistemler .....	11
1.4.7. Bulanık Mantık .....	12
1.4.8. Yapay Sinir Ağları .....	12
<b>BÖLÜM 2: YAPAY SİNİR AĞLARI</b> .....	<b>13</b>
2.1. Yapay Sinir Ağları Kavramı .....	13
2.2. Yapay Sinir Ağları'nın Tarihsel Gelişimi .....	14
2.3. Yapay Sinir Ağları'nın Yapısı ve Temel Elemanları .....	16
2.4. Yapay Sinir Ağı Çeşitleri .....	19
2.5. Yapay Sinir Ağları'nın Avantajları .....	26
2.6. Yapay Sinir Ağları'nın Dezavantajları.....	27
2.7. Yapay Sinir Ağları Öğrenme Algoritmaları.....	29
2.7.1. Danışmanlı Öğrenme (Supervised Learning) .....	29

2.7.2. Danışmansız Öğrenme (Unsupervised Learning).....	30
2.7.3. Takviyeli Öğrenme (Reinforcement Learning).....	30
2.8. Birleşik Yapay Sinir Ağları.....	31
2.9. Yapay Sinir Ağları'nın Uygulama Alanları .....	32
2.10. Yapay Sinir Ağları ve Bulanık Mantık .....	33
<b>BÖLÜM 3: AFET YÖNETİMİ VE AFETLERDEN ZARAR GÖREBİLİRLİK .</b>	<b>39</b>
3.1. Afet Yönetimi ve Aşamaları .....	39
3.2. Zarar Görebilirlik .....	41
3.2.1. Fiziksel Zarar Görebilirlik .....	42
3.2.2. Ekonomik Zarar Görebilirlik .....	42
3.2.3. Sosyal Zarar Görebilirlik .....	43
<b>BÖLÜM 4: YSA KULLANILARAK TÜRKİYE'DEKİ İLLERİN ZARAR</b>	
<b>GÖREBİLİRLİĞİNİN HESAPLANMASINA İLİŞKİN YÖNTEM</b>	
<b>VE UYGULAMA .....</b>	<b>46</b>
4.1. Araştırmanın Yöntemi.....	46
4.1.1. Uygulamada Kullanılan Programlar .....	47
4.2. Uygulama .....	51
4.2.1. SOM (Kendini Düzenleyen Haritalar) Kullanılarak Sosyal Zarar	
Görebilirlik Sınıflarının Oluşturulması.....	51
4.2.2. MATLAB'ta Yapay Sinir Ağının Oluşturulması ve Eğitim Aşaması .....	55
4.2.3. Test Sonuçları ve Yorumları .....	65
<b>SONUÇ VE ÖNERİLER.....</b>	<b>67</b>
<b>KAYNAKÇA .....</b>	<b>70</b>
<b>EKLER.....</b>	<b>76</b>
<b>ÖZGEÇMİŞ.....</b>	<b>91</b>

## KISALTMALAR LİSTESİ

- ABD** : Amerika Birleşik Devletleri  
**ART** : Adaptive Resonance Theory  
**ART-1**: Adaptive Resonance Theory-1  
**ART-2**: Adaptive Resonance Theory-2  
**BAM** : Bidirectional Associative Memories  
**Diğ.** : Diğerleri  
**DLVQ**: Dynamic Learning Vector Quantization  
**ECG** : Elektrokardiyografi  
**EEG** : Elektroenseflografi  
**FCM** : Fuzzy Cognitive Maps  
**LVQ** : Learning Vector Quantization  
**LVQ1** : Learning Vector Quantization-1  
**LVQ2** : Learning Vector Quantization-2  
**PC** : Personal Computer  
**Sn.** : Saniye  
**SOM** : Self Organizing Map  
**TÜİK** : Türkiye İstatistik Kurumu  
**YSA** : Yapay Sinir Ağları

## TABLO LİSTESİ

<b>Tablo 1.</b> Bilgisayar ve Beynin Karşılaştırılması.....	16
<b>Tablo 2.</b> Biyolojik Sinir Sistemi Bileşenlerinin YSA'daki Karşılıkları .....	18
<b>Tablo 3.</b> MATLAB Veri İşleme Fonksiyonları.....	50
<b>Tablo 4.</b> MATLAB Öğrenme Algoritmaları .....	50
<b>Tablo 5.</b> Bazı Kümeleme Yaklaşımlarının Karşılaştırılması.....	51
<b>Tablo 6.</b> Normalizasyon Aşaması.....	53
<b>Tablo 7.</b> Sosyal Faktörlerden Elde Edilen Sınıflar .....	54
<b>Tablo 8.</b> Sosyal Sınıflar ve Sınıflara Ait İller.....	55
<b>Tablo 9.</b> MATLAB Eğitim Seti.....	56
<b>Tablo 10.</b> MATLAB Eğitim Seti Çıkış Verileri.....	57
<b>Tablo 11.</b> İllere Göre Test Sonuçları .....	64



## ŞEKİL LİSTESİ

Şekil 1. Genetik Algoritmaların Yapısı .....	9
Şekil 2. Uzman Sistemlerin Geliştirilmesi.....	11
Şekil 3. Biyolojik Sinir Hücresinin Yapısı .....	17
Şekil 4. Yapay Nöronun Yapısı.....	18
Şekil 5. Sık Kullanılan Transfer Fonksiyonları .....	19
Şekil 6. Yapay Sinir Ağları'nın Sınıflandırılması .....	20
Şekil 7. İleri Beslemeli ve Geri Beslemeli Ağ Yapıları.....	21
Şekil 8. Bir Ağda Geri Yayılımlı Bir Öğrenme Algoritmasının Uygulanması .....	22
Şekil 9. İleri Beslemeli Geri Yayılma Ağların Genel Yapısı.....	23
Şekil 10. Öğrenmenin Hata Uzayındaki Gösterimi .....	25
Şekil 11. Çok Boyutlu Hata Uzayı.....	25
Şekil 12. Danışmanlı Öğrenme Yapısı .....	29
Şekil 13. Danışmansız Öğrenme Yapısı .....	30
Şekil 14. Takviyeli Öğrenme Yapısı.....	31
Şekil 15. Birleşik Sinir Ağlarının Elemanları .....	31
Şekil 16. Klasik ve Bulanık Mantık Arasındaki Fark .....	34
Şekil 17. Üyelik Fonksiyonları Çeşitleri.....	35
Şekil 18. Sinirsel Bulanık Sistemlerin Sinir Ağları ve Bulanık Mantık İle İlişkisi .....	36
Şekil 19. Sinirsel Bulanık Sistemlerin Birinci Yapısı.....	37
Şekil 20. Sinirsel Bulanık Sistemlerin İkinci Yapısı .....	37
Şekil 21. Afet Yönetimi Modeli .....	39
Şekil 22. Wisconsin Üniversitesi Afet Yönetimi Modeli .....	40
Şekil 23. Tehlike, Zarar Görebilirlik, Risk ve Afet Arasındaki İlişki.....	42
Şekil 24. SPSS Clementine Programı Arayüzü .....	47
Şekil 25. MATLAB Genel Yapı .....	49
Şekil 26. Verilerinin SPSS Clementine 11.1 Programına Aktarılması.....	52
Şekil 27. Kohonen Ağları İle Kümeleme Analizi Sonuçları .....	54
Şekil 28. Eğitim Seti Giriş Verilerinin MATLAB Programına Aktarılması .....	57
Şekil 29. Eğitim Seti Çıkış Verilerinin MATLAB Programına Aktarılması.....	58
Şekil 30. Neural Networks Toolbox Modülü.....	59

<b>Şekil 31.</b> Import to Network / Data Manager Arayüzü.....	60
<b>Şekil 32.</b> Eğitilecek Ağın Tanımlanması .....	60
<b>Şekil 33.</b> Veriler Aktarıldıktan Sonra Network/Data Manager Arayüzü.....	61
<b>Şekil 34.</b> MATLAB'ta Ağın Eğitilmesi.....	61
<b>Şekil 35.</b> Neural Network Training Arayüzü.....	62
<b>Şekil 36.</b> Denenen İterasyonların Regresyon Sonuçları .....	62
<b>Şekil 37.</b> Eğitimin Tamamlanmasından Sonra Elde Edilen Regresyon Sonucu.....	63

**Tezin Başlığı:** Yapay Sinir Ağları İle Afet Yönetiminde Sosyal Zarar Görebilirlik Riskinin Belirlenmesi

**Tezin Yazarı:** Dilek SÜRMEİ

**Danışman:** Prof. Dr. Erman COŞKUN

**Kabul Tarihi:** 08/07/2011

**Sayfa Sayısı:** viii(ön kısım)+76(tez)+15(ekler)

**Anabilim dalı :** İşletme

**Bilim dalı:** Üretim Yönetimi ve Pazarlama

Son yıllarda insan düşünüş ve yargısını bilgisayarlar aracılığı ile gerçekleştirmeyi hedefleyen Yapay Zeka konusu önem kazanmış ve teknoloji ve yazılım alanındaki gelişmeler ile Yapay Zeka'nın farklı alanlarda uygulamaları gerçekleştirilmiştir. Bu tezin amacı Yapay Zeka teknikleri olan Yapay Sinir Ağları ve Bulanık Mantık'ın detaylı incelenmesi ve depremden sosyal zarar görebilirliğin tahmininde bu tekniklerin yardımcı olabileceğinin gösterilmesidir.

Ülkemizde sıkça rastlanan doğal afet türlerinden biri olan depremin meydana getirebileceği zararlar fiziksel, ekonomik ve sosyal olmak üzere üç kategoride incelenmektedir. Fiziksel zarar görebilirlik literatürde detaylı olarak çalışılmıştır. Ancak özellikle sosyal zarar görebilirlik ülkemizde çok fazla çalışmanın yapılmadığı bir alandır. Hızlı nüfus artışı, iç ve dış göçler, bilgi ve eğitim eksikliği gibi faktörler sosyal zarar görebilirlik kavramını oluştururken bu faktörlerin iyileştirilememesi zaman içerisinde doğal afetin etkilerinin artmasına neden olmaktadır. Fiziksel zarar görebilirliğin yanı sıra sosyal zarar görebilirliğinde çalışılması gerekmektedir ki yaşanan afetlerin yarattığı kayıpların minimum düzeye indirilmesi mümkün olabilsin.

Tezde bu temele dayanılarak Yapay Zeka tekniklerinden Yapay Sinir Ağları kullanılarak sosyal faktörlere göre Türkiye'deki iller sınıflara ayrılmıştır. Oluşturulan sınıflarla sosyal faktörler arasında Yapay Sinir Ağları ile ilişki kurulabildiğinin gösterilmesi amaçlanmıştır.

Bu amaçla çalışmada öncelikle Yapay Sinir Ağları kapsamlı bir literatür taraması yapılarak anlatılmış, sonrasında SPSS Clementine programı kullanılarak kümeleme analiziyle illerin sosyal sınıfları belirlenmiştir. Daha sonra ise MATLAB programının Yapay Sinir Ağı modülü kullanılarak analizler tamamlanmıştır. Analiz sürecinde kullanılan 81 ile ait tablolar Türkiye İstatistik Kurumu'ndan alınan ikincil veriler doğrultusunda hazırlanmıştır.

Çalışma sonunda ise Yapay Sinir Ağları kullanılarak sosyal faktörlerle, oluşturulan sosyal sınıfların ilişkilendirilebildiği yapılan analiz ve testlerin sonuçlarıyla desteklenmiştir.

**Anahtar Kelimeler:** Yapay Zeka, Yapay Sinir Ağları, Sosyal Zarar Görebilirlik.

**Title of Thesis:** Utilization of Artificial Neural Networks for Calculation of Social Vulnerability in Disaster Management

**Author:** Dilek SÜRMEĪ

**Supervisor:** Prof. Dr. Erman COŐKUN

**Date:** 08/07/2011

**Nu. of Pages:** viii(pre text)+76(main body)+15(app.)

**Department:** Business

**Subfield:** Manufacturing Management and Marketing

In recent years, artificial intelligence, which tries to mimic human thinking and decision making process in computerized environments, has been applied in different areas. New developments in technology and software areas allowed to broaden these different applications. In this thesis, the goal is to research artificial intelligence and its techniques and to apply them for calculation of social vulnerability in earthquakes. Especially Neural Networks and Fuzzy Logic are main focus areas.

Earthquake damage, one of the common types of natural disasters in our country, can be scrutinized in three categories: physical, economic and social. Among these three physical vulnerability has been studied widely. However when it comes to social vulnerability, it is a new area and there has been limited number of studies conducted in this area. Rapid population growth, internal and external migration, factors such as lack of information and training constitute social vulnerability factors and if they are not improved over time, the risks of natural disaster increases. Therefore, social vulnerability is also important and it must be also researched in detail to avoid consequences.

In order to reach this goal, first cities in Turkey are classified into groups based on social factors by using neural networks. Then, the relationship between created groups and social factors is established by using neural networks.

For this purpose, first a comprehensive literature review about Artificial Intelligence, Artificial Neural Networks and Fuzzy Logic was conducted. Then, in order to classify cities into social groups SPSS Clementine was used. Finally artificial neural networks module of MATLAB has been used to complete the analysis. The data for social indicators of 81 cities were gathered from Turkish Statistics Institute.

The result of this study provides a support that social factors can be used in order to measure social vulnerability level of cities and Artificial Neural Networks can be utilized for this purpose.

**Key Words:** Artificial Intelligence, Artificial Neural Networks, Social Vulnerability.

## GİRİŞ

Yapay zeka, insanın düşünme yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmak olarak tanımlanır. Yani programlanmış bir bilgisayarın düşünme girişimidir. Daha geniş bir tanıma göre ise, yapay zeka; bilgisayarların, bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekasına özgü kapasitelerle donatılması bilimidir (Sağirođlu ve diđ., 2003:4).

Bilgisayarların, hızlarının ve kapasitelerinin artması, yeni programlama tekniklerinin geliştirilmesi sayesinde bilimin her alanında kullanımları yaygınlaşmaktadır. Yapay Zeka uygulamalarından; Yapay Sinir Ağları (YSA), Genetik Algoritmalar ve Bulanık Mantık teknikleri son yıllarda sıkça kullanılan programlama metotlarıdır. Günümüzde yaygın olarak kullanılan başlıca Yapay Zeka yöntemleri; Uzman Sistemler, Bulanık Mantık, Genetik Algoritma ve Yapay Sinir Ağları'dır.

Yapay Zeka'nın bir alt kolu olan Yapay Sinir Ağları (YSA) genel anlamda, beynin bir işlevi yerine getirme yöntemini bilgisayar ortamında modellemek için tasarlanan bir sistem olarak tanımlanabilir. YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar şeklinde düzenlenir. Donanım olarak elektronik devrelerle ya da bilgisayarlarda yazılım olarak gerçekleştirilebilir. Beynin bilgi işleme yöntemine uygun olarak YSA, bir öğrenme sürecinden sonra; bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip, paralel dağılmış bir işlemcidir. Öğrenme süreci, arzu edilen amaca ulaşmak için YSA ağırlıklarının yenilenmesini sağlayan öğrenme algoritmalarını içerir.

Dođal afetlerden ve özellikle de depremden kaçışın mümkün olmadığı, depremle birlikte yaşamının öğrenilmesi gerektiđi ve büyük bölümü deprem kuşađı üzerinde bulunan ülkemizde depremin büyüklüğünün, nerede ve ne zaman olacağına önemi azalırken, bilim ve teknolojiye dayalı Afet Yönetimi'nin önemi giderek artmaktadır. Çünkü hızlı nüfus artışı, iç ve dış göçler, çarpık kentleşme, bilgi ve eğitim eksikliği, denetim yetersizliği ve cezaların caydırıcılığının yitilmesi zaman içerisinde dođal afet tehlikesi ve riskini daha da arttırmaktadır. Yaşanan afetlerin yarattığı can ve mal kayıpları yanında ekonomik kayıpların da minimize edilmesi, ancak etkili bir Afet Yönetimi'nin varlığı ile sağlanabilir (Uzunçubuk, 2005:3).

## **Çalışmanın Konusu ve Önemi**

Matematiksel modeli verilen bir sistemin analizi yapılabilir veya değişik girişlere karşılık çıkışları tahmin edilebilir. Giriş ve çıkış eğitim çiftleri verilen bir fonksiyonu yapay sinir ağı öğrenebilmektedir. Öğrenme işlemi, ağı içerisindeki ağırlıkların belirlenmesiyle gerçekleşmekte ve verilen fonksiyona optimal yaklaşım sağlanmaktadır.

Çalışma, yapay zeka tekniklerinden yapay sinir ağları kullanılarak depremden sonra ortaya çıkabilecek hasarların tespitinde sosyal faktörlerin etkisine göre sosyal sınıflandırılmanın yapılabileceğini ve bu sınıflarla faktörlerin ilişkilendirilebileceğini ortaya koymaktadır. Çalışma bu yönüyle daha önce yalnız fiziksel faktörlere göre belirlenen deprem risk derecelerinin sosyal ve ekonomik faktörlerin de dahil edilerek yeniden hesaplanabilmesine zemin hazırlayarak literatüre katkıda bulunmaktadır.

## **Çalışmanın Amacı**

Yapay Sinir Ağları birçok amaç için kullanılan tahmin tekniklerinden biridir. Bu bağlamda çalışmanın amacı, Yapay Sinir Ağları tekniğinden faydalanarak iller bazında sosyal sınıflar oluşturmak ve oluşturulan sınıfların faktörlerle ilişkilendirilmesi sağlanarak afet yönetimi üzerine bir uygulama ile desteklemektir.

Çalışmada bu amaçla 81 ilin verileri derlenmiş ve bu verilerle Yapay Sinir Ağlarının özellikle doğrusal olmayan ve karmaşık ilişkileri öğrenebilme özelliğinin deprem risk derecelerinin sosyal faktörler açısından tahmin edilmesinde kullanılabilirliğinin araştırılması amaçlanmıştır.

## **Çalışmanın Yöntemi**

Çalışmada keşifsel araştırma yaklaşımı benimsenmiştir. Bunun nedeni; bu çalışmada ölçülmesi fiziksel faktörlere göre daha zor olan sosyal ve ekonomik faktörlerin Yapay Sinir Ağları ile ölçülebileceği ortaya konulmaktadır. Bu bağlamda illere ait deprem risk derecelerinin hesaplanmasında sosyal faktörlerin de kullanılabileceğinin mümkün olduğuna inanılmaktadır. Çalışma bu ekseninde yürütülmüştür.

Uygulama bölümünde illere göre deprem sonrasında ortaya çıkabilecek hasarların en aza indirgenmesinde sosyal ve ekonomik faktörlerin de kullanılması gerektiğinin tespiti için illerin kümeleme analizi ile sınıflandırılmasında SPSS Clementine programı ve elde

edilen sınıfların faktörlerle ilişkilendirilebildiğinin gösterilmesi için de MATLAB programının Yapay Sinir Ağları modülü kullanılmıştır. Türkiye İstatistik Kurumu'ndan alınan veriler doğrultusunda eğitimde kullanılacak giriş ve çıkış setleri oluşturulmuştur. Bu setler yapay sinir ağına aktarılarak ağın eğitilmesi sağlanmıştır.

Uygulamada kullanılan veri seti Türkiye'de bulunan 81 ilin sosyal zarar görebilirliğe etki eden faktörlerden oluşmaktadır. Deprem risk derecelerinin analizinde kullanılacak veriler Türkiye İstatistik Kurumu'ndan (TÜİK) alınarak illere göre kategorilere ayrılmıştır. Bu kategoriler; "Eğitim Durumu", "Konut Sayısı", "Doğurganlık Hızı", "Kişi Başına Gelir", "Net Göç", "İşsizlik Oranı" ve "Toplam Nüfus" olmak üzere 7 başlık altında toplanmaktadır. "Eğitim Durumu" faktörü ise "Okuma Yazma Bilmeyen", "İlköğretim ve Ortaöğretim Mezunu" ve "Yükseköğretim Mezunu" olarak ayrılarak veriler toplamda 9 faktöre ayrılmaktadır.

Analizde kullanılan anakütlenin sonlu yapıda olması ve kümeleme analizi sonucunda bazı sınıflarda yeterli sayıda ilin bulunmaması nedeniyle, analizlerde daha verimli sonuçlar elde etmek için eğitim ve test aşamalarında tamsayım yapılmıştır.

### **Çalışmanın Kapsamı**

Tezin ilk bölümünde, Yapay Zeka'nın tanımı, gelişim süreci, uygulama alanları, Yapay Zeka tekniklerinden "Genel Algoritmalar", "Tabu Araştırma", "Isıl İşlemler", "Karınca Algoritması", "Bağımsızlık Sistem Algoritması", "Yapay Sinir Ağları", "Bulanık Mantık" ve "Uzman Sistemler" kısaca anlatılmıştır.

İkinci bölümde; Yapay Sinir Ağları'nın tarihsel gelişimi incelenerek günümüze kadar sağlanan ilerlemeler sunulmuş, Yapay Sinir Ağlarının yapıları, çeşitleri, öğrenme algoritmaları, Yapay Sinir Ağları uygulamalarının avantajları ve dezavantajları belirtilmiştir. Ayrıca belirsizlikleri çözümlemede katı bir matematiksel model yerine insan düşünme sistemine daha yakın çözümler getirmesi açısından Yapay Sinir Ağları ile birleştirilerek Sinirsel Bulanık Sistemlerin oluşturulmasını sağlayan Bulanık Mantık yaklaşımına ve Sinirsel Bulanık Sistemlerin işleyişine de değinilmiştir.

Üçüncü bölümde uygulamanın konusunu oluşturan Afet Yönetimi ve afetlerden zarar görebilirlik kavramlarına değinilmiştir. Daha sonra zarar görebilirlik kavramı genişletilerek fiziksel, ekonomik ve sosyal zarar görebilirlik kavramları ile içerdikleri

faktörler daha geniş olarak anlatılmıştır.

Dördüncü bölümde ise, yapılan uygulamada izlenen yöntem ve kullanılan programlar hakkında bilgi verilmiştir. Analiz süreçleri anlatılarak kümeleme analizindeki, ağın eğitilmesi ve test edilmesindeki adımlara değinilmiştir. Çalışmanın sonunda kullanılan programlardan elde edilen sonuçlara ve yorumlarına yer verilmiştir.



## **BÖLÜM 1: YAPAY ZEKA**

Bu bölümde, Yapay Sinir Ağları ve Bulanık Mantık tekniklerine temel oluşturan Yapay Zeka konusunun ayrıntılı olarak incelenmesi amaçlanmıştır. Bu amaçla bölümde Yapay Zeka kavramı, Yapay Zeka'nın tarihçesi ve Yapay Zeka teknikleri anlatılmıştır.

### **1.1. Yapay Zeka Kavramı**

Yapay Zeka; insanlarda, hayvanlarda ve makinelerde zeki davranışın ne olduğunu inceleyen ve insan yapımı aygıtların nasıl bu tip davranışlar sergileyebileceğini bulmaya çalışan bir bilim dalıdır (Whitby, 2005:17).

Yapay Zeka, zeka ve düşünme gerektiren işlemlerin bilgisayarlar tarafından yapılmasını sağlayacak araştırmaların yapılması ve yeni yöntemlerin geliştirilmesi hususunda çalışılan bir bilim dalıdır (Sağiroğlu ve diğ., 2003:4). Yapay Zeka aynı zamanda, bilgi ve davranışa dayalı sistemler oluşturan ve zeki davranışlar üzerinde araştırma yapan bir bilgisayar bilim dalı olarak da tanımlanabilir. Bu tanıma göre Yapay Zeka, bilgisayar bilimlerinin, zeki bilgisayar sistemleri tasarlama ile ilgili olan dalıdır. Buradaki zekilik kavramından, dili anlama, öğrenme, cevaplama, problem çözme gibi insan davranışlarındaki zeka kastedilmektedir (Barr ve diğ., 1989:3).

Yapay Zeka, çoğunlukla insanın düşünme yeteneğini ve beynin çalışma özelliğini modellemeye çalışan yöntemlerden oluşmaktadır. Yapay Zeka'nın asıl amacı, bilgisayarları daha yararlı hale getirmektir (Winston, 1984:1). Başka bir deyişle Yapay Zeka sistemleri, insanın zekasını bilgisayar aracılığı ile taklit etmek ve bu anlamda belli bir ölçüde bilgisayarlara öğrenme yeteneği kazandırabilmeyi amaçlamaktadır (Serhatlıoğlu ve Hardalaç, 2009:1).

Belirli durum veya problem karşısında veriye dayalı bilgiler toplayabilen ve bu bilgiler ışığında çözüm üretebilen zeki sistemlerinin geliştirilebilmesini mümkün kılan en önemli faktörlerden biri Yapay Zeka araştırmalarıdır. Yapay Zeka sistemleri yeni kavram ve işleri öğrenebilmekte; dünya ile ilgili faydalı sonuçlar çıkarıp muhakeme edebilmekte; tabii dilleri anlayabilmekte veya görülebilir bir tabloyu sezebilmektedir. Özetle Yapay Zeka sistemleri, insan zekasına benzer akıllı davranışlar gösterip, bu özellikleri sayesinde daha farklı ve olağanüstü özelliklerde performans gösterebilmektedir (İncetürkmen, 1999:19).

## 1.2. Yapay Zeka'nın Tarihçesi

Yapay Zeka konusunda ilk çalışma McCulloch ve Pitts tarafından 1940'lı yıllarda yapılmıştır. 1943'te Alan Turing, İkinci Dünya Savaşı sırasında Kripto Analizine yönelik olarak üretilen Elektro-Mekanik cihazlar sayesinde "Bilgisayar Bilimi" ve "Yapay Zeka" kavramlarını ortaya çıkarmıştır.

Yapay Zeka terimi ise ilk defa 1956 yılında ABD'de "Makine Zekası" konferansında John McCarthy tarafından ortaya konmuştur. Kaynaklarda "Yapay Us" veya "Suni Zeka" olarak da bilinen Yapay Zekanın (Artificial Intelligence) çağdaş bir bilim dalı olarak gelişmesi, 1956 yılında C. Shannon, M. Minsky ve J. McCarthy'nin katkıları ile başlamıştır (Allahverdi, 2002:1).

1958 yılında John McCarty yapay zeka araştırmalarında kullanılan LISP programlama dilini geliştirmiştir. 1965 yılında, Zadeh tarafından önerilen bulanık mantık teorisi ise Yapay Zeka'ya yeni bir bakış açısı kazandırmıştır. 1975'te Holland canlılarda doğal gelişim prensibine dayanan genetik algoritmayı önermiştir. Daha sonra 1976 yılında Sejnowski ise "Boltzman Makinesi"ni geliştirmiş ve buna geri yayılım algoritmasını uygulamıştır. Sutton ve Barto ise 1978 yılında takviyeli öğrenme modelini geliştirmişlerdir.

1985 yılında Mead, elektronik bir devreye hayvan sinir sistemini yerleştirmeye çalışmıştır. Yine aynı yıl Kosko, bulanık mantık ile yapay sinir ağları yaklaşımlarını bir araya getirerek BAM (Bidirectional Associative Memories) ve FCM (Fuzzy Cognitive Maps) yapılarını oluşturmuştur.

1990'lardan sonra günümüze kadar geçen sürede bilgisayar ve yapay zeka teknolojilerinde yaşanan gelişmeler şu şekilde özetlenebilir (Kayabaş, 2010:27):

- 1993: MIT'de Cog adlı insan biçimli bir robotun yapımına başlanmıştır.
- 1997: IBM firması tarafından geliştirilen ve Yapay Zeka teknikleri kullanılarak tasarlanan Deep Blue Satranç Programı dünya satranç şampiyonu Gary Kasparov'u yenmeyi başarmıştır.
- 1998: Tiger Electronics firması evlere girmeyi başaran ilk yapay zeka oyuncuğu olan Furby'yi piyasaya sürmüştür.

- 1999: İlk robot köpek AIBO ilk robot evcil hayvan tanıtımıyla piyasaya sürülmüştür.
- 2000: Cynthia Breazeal, karşısındaki kişiyle konuşurken mimikler ve yüz ifadeleri kullanabilen “Kismet” adında bir robot geliştirmiştir.
- 2005: Honda firması o güne kadar yapılmış en becerikli insansı robot olan Asimo'yu tüm dünyaya tanıtmıştır.

Günümüzde A.B.D.’de Yapay Zeka çalışmalarına katkı sağlamak amacıyla Alan Turing’in adıyla anılan Turing Testi, Makine Zekasına sahip yazılımların üzerinde uygulanarak, başarılı yazılımlar ödüllendirilmektedir. Turing Testi’nin amacı bir makinenin düşündüğünü söylemenin mantıksal olarak mümkün olup olmadığıdır. Yapılan Turing Testi’nde, birbirini tanımayan birkaç insandan oluşan bir denek grubu birbirleri ile ve bir Yapay Zeka diyalog sistemi ile geçerli bir süre sohbet etmektedirler. Birbirlerini yüz yüze görmeden yazışma yolu ile yapılan bu sohbet sonunda deneklere sorulan sorular ile hangi deneğin insan hangisinin Makine Zekası olduğunu saptamaları istenir. Şimdiye kadar yapılan testlerin bir kısmında Makine Zekası insan zannedilirken gerçek insanlar Makine zannedilmiştir ([www.yapay-zeka.org](http://www.yapay-zeka.org)).

### **1.3. Yapay Zeka’nın Avantajları ve Dezavantajları**

Yapay Zeka yazılımı, bilgisayarı daha kullanışlı hale getirmektedir. Yapay Zeka yazılımlarıyla kullanıcılar bilgisayarlarla, gizli komutların veya bilgisayar dillerinin yerine kendi anadilleriyle konuşma imkanına sahip olmaktadır. Böylece Yapay Zeka ile hiçbir eğitim almamış kullanıcılar bile bilgisayara hemen adapte olabilecek ve faydalı işler yapabileceklerdir.

Yapay Zeka, verinin yetersiz olduğu ya da bilinmediği ya da bilinen algoritmaların olmadığı karmaşık problemler için de çözüm üretebilmektedir. Bu teknikler, kullanıcının tecrübesiyle birleştirildiğinde, performansı ve üretkenliği arttırmaktadır.

Yapay Zeka, aşırı bilgi yüklenmesinin hafiflemesine de yardımcı olmaktadır. Gereksinim duyulan bilgiyi bulma ve ona erişmek için yeni yollar sağlamaktadır. Buna ek olarak, doğal dil arayüzleri bilgisayar veri tabanlarının kullanımını daha kolay hale getirmektedir.

Yapay Zeka'nın en önemli avantajlarından biri de bilginin kullanılabilir bilgiye dönüştürülmesinde kolaylık sağlamasıdır. Süreçte bilgiyi analiz etmek, organize etmek, elemek ve önemli olanlarını seçmek gerekmektedir. Böylece bilgi kullanılabilir bilgiye dönüşmekte ve bir problemi çözmek için ya da bir karar vermek için kullanılabilir hale gelmektedir.

Yapay Zeka'nın avantajlarının yanı sıra dezavantajları da bulunmaktadır. Uygulamalar geliştirildikçe Yapay Zeka avantajları artmaktadır ancak bu gelişmeler maliyetli olmaktadır. Bilgisayarlar çok güçlü ve faydalı araçlar haline gelirken artan maliyetler dezavantaj olarak ortaya çıkmaktadır. Orta ve büyük ölçekli Yapay Zeka uygulamaları genellikle çok hızlı merkezi işlem birimi ve çok belleği olan güçlü bilgisayarlar gerektirmektedir.

Yapay Zeka'nın diğer bir dezavantajı da yazılımının geliştirilmesinin zorluğudur. Yapay Zeka programları genel olarak kompleks ve doğası gereği geleneksel programlamadan farklı anlayışla programlama olduğundan karmaşık bir yapıya sahiptir. Bunun sonucunda, yazılımların geliştirilmesi için çok zaman harcanmaktadır. Bunun sonucunda ise yazılımlar daha pahalı hale gelmektedir.

Geliştirilmiş Yapay Zeka programlama dilleri ve uzman sistem geliştirme ortamları gibi yazılım geliştirme araçları yazılım geliştirmeyi daha hızlı ve daha basit hale getirmektedir. Ancak bu geliştirme araçları genellikle pahalıdır ve bunları kullanmak için yetenekli, eğitilmiş insanlar gerekmektedir.

#### **1.4. Yapay Zeka Teknikleri**

Yapay Zeka çalışmalarının amacı, insan gibi düşünen ve davranan sistemler tasarlamaktır. Genel olarak Yapay Zeka çalışmaları incelendiğinde insanın sadece zekası ve beyni ile sınırlı kalmayıp diğer canlıların da çalışmalara konu edildiği görülmektedir.

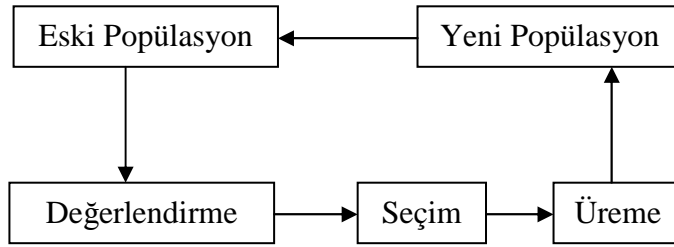
Yapay Zeka teknikleri; Yapay Sinir Ağları, Bulanık Mantık, Genetik Algoritmalar, Tabu Araştırma Algoritması, Isıl İşlemler Algoritması, Karınca Algoritması, Bağışıklık Sistemi Algoritması ve Uzman Sistemler olarak sayılabilir. Bu tekniklerin her biri izleyen kısımlarda detaylı olarak incelenmiştir.

### 1.4.1. Genetik Algoritmalar

Genetik Algoritmalar, karmaşık düzenli problemlerin çözümünü gerçekleştirmek amacıyla, kromozomların yeni diziler üretme esasını temel alan, sezgisel bir araştırma yöntemidir (Özçakar, 1998:69). Bu algoritmalar biyoloji bilimindeki doğal evrim ve doğal üreme teorisinin bilgisayarda modellenmesine dayanmaktadır (Cebesoy, 1995:185). Popülasyon temelli olarak belirli bir uygunluk fonksiyonunun optimize edilmesi esasına dayalı çalışan Genetik Algoritmalar ilk defa Holland tarafından 1970'lerde ortaya atılmış ve daha sonraları Goldberg tarafından geliştirilmiştir.

Genetik Algoritmalarda, araştırma uzayında bulunan bazı çözümlerin oluşturduğu bir başlangıç yoğunluğu kullanılmaktadır. Bu başlangıç yoğunluğu her kuşakta, doğal seçme ve tekrar üreme işlemleri vasıtası ile sırasıyla geliştirilmektedir. Son kuşağın en uygun bireyi, problem için optimal çözüm olarak kabul edilmektedir (Sağiroğlu ve diğ., 2003:12). Temel bir genetik algoritmanın yapısı Şekil 1'de gösterilmiştir.

**Şekil 1. Genetik algoritmaların yapısı**



**Kaynak:** Civalek (2005:1330)

Optimizasyon işlemine ihtiyaç duyulan çok çeşitli alanlarda kullanılabilen Genetik Algoritmalar çok sayıda, sürekli ve ayrık değişkenlerle çalışabilmektedir. Paralel bilgisayarlarla çalışmaya uygundur ve son derece karmaşık sistemlere ait değişkenleri bölgesel en iyi aralıklara takılmadan optimize edebilmektedir. Sayısal olarak üretilmiş veya deneysel verilerle, analitik fonksiyonlarla çalışabilmekte ayrıca tek bir çözüm yerine optimum değişkenler kümesi sunabilmektedir.

### 1.4.2. Tabu Araştırma Algoritması

Tabu Araştırma Algoritması, 1989 yılında Glover tarafından ortaya konulmuştur. Tabu Araştırma Algoritması, kombinatorial eniyileme problemlerini çözmek üzere

geliştirilmiş ve tesis yerleşimi, çizelgeleme, ulaştırma, parti büyüklüğü gibi birçok kombinatoriyal eniyileme problemlerine başarıyla uygulanmıştır (Alabaş, 1999:36).

Tabu Araştırma Algoritması, bir başlangıç çözümü ile aramaya başlar. Algoritmanın her iterasyonunda tabu olmayan bir hareket ile mevcut çözümün komşuları içerisinde bir tanesi seçilerek değerlendirilir. Tabu Araştırma Algoritması, bir veya birden fazla durdurma koşulunu sağlayıncaya kadar aramasını sürdürmektedir. Bu koşullardan bazıları (Güden ve diğ., 2005:8);

- Seçilen bir komşu çözümün başka bir komşusunun olmaması,
- Belirlenen iterasyon sayısına ulaşılması,
- Belirlenen çözüm değerine ulaşılması,
- Algoritmanın bir yerde tıkanması veya daha iyi sonuç üretememesi,

olarak ifade edilmektedir. Eğer amaç fonksiyonunun değerinde bir iyileştirme sağlanmışsa komşu çözüm, mevcut çözüm olarak dikkate alınmaktadır.

#### **1.4.3. Isıl İşlem Algoritması**

“Tavlama Benzetimi Yöntemi” olarak da bilinen Isıl İşlem Algoritması, ilk karşılaşılan daha iyi çözümü kabul ederek en iyiyi arama metodlarının farklı bir uygulama biçimidir. Her adımda daha iyi bir noktaya gitmeyi hedefleyen gradyan ile arama yönteminden farklı olarak, bu algoritmanın en önemli dezavantajı, süreç içinde giderek azalan bir olasılıkla daha kötü bir çözüm bulmaya neden olabilir (Coşkun, 2007:144).

#### **1.4.4. Karınca Koloni Algoritması**

Karınca Koloni Algoritması 1996’da Marco Dorigo tarafından önerilmiş en yeni sezgisel algoritmalarından biridir. Karınca Koloni Algoritması karıncaların doğal hayatta yiyecek bulmak için aralarında kurduğu sistemi modellemektedir. Bu sistemde bir karınca geçtiği yol üzerinde bir salgı bırakır. Karıncalar gidecekleri yolu bu salgının yoğunluğuna bağlı olarak seçmektedirler. Belirli bir süre sonra tüm karıncalar yiyeceklerin yerlerini aralarındaki bu dolaylı iletişim ile öğrenirler. Bu süreç optimizasyon problemlerinde kullanılmaktadır. Ayrıca Karınca Koloni Algoritması, Genetik Algoritmalar gibi popülasyon tabanlı yaklaşıma sahiptir.

### 1.4.5. Bağışıklık Sistemi Algoritması

Yapay Zeka tekniklerinden biri olan Bağışıklık Sistemi Algoritması, insan vücudundaki bağışıklık sistemi örnek alınarak tasarlanmıştır. Bağışıklık sistemi yabancı antijenlerle, kendi vücut antijenlerinin ayrılmasını sağlayan ve yabancı maddeleri yok eden karmaşık bir mekanizmadır. Yapay bağışıklık sistemleri öğrenme yeteneğine sahip sistemlerdir. Bu nedenle sürekli deęişen ortamlarda bilinmeyen keşfetme problemlerinin çözümü için kullanılmaktadır.

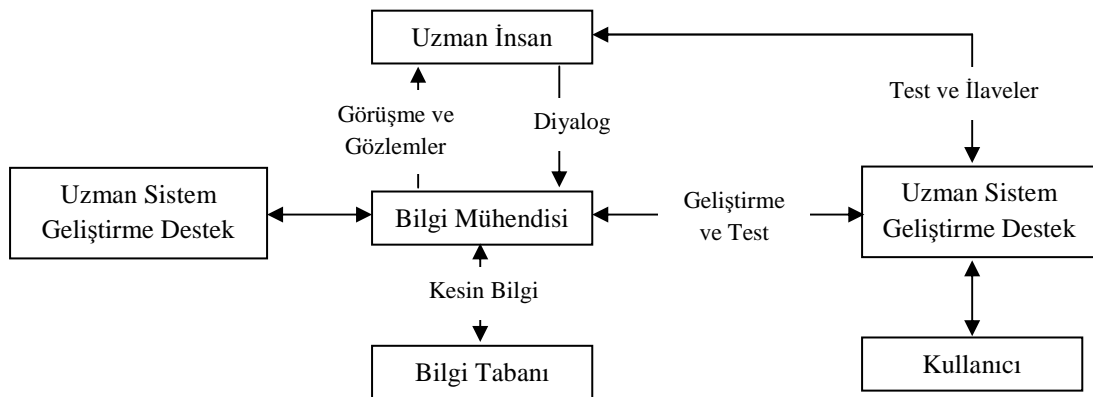
### 1.4.6. Uzman Sistemler

Uzman Sistemler 1960'larda, J. LEDERBERG'in spektrograf verilerini bilgisayarla yorumlama çabaları sonucu ortaya çıkmış, Yapay Zeka'nın problem çözme alanı dışına taşarak yeni bir dal haline gelmiştir. Uzman Sistemler belirli bir alanda sisteme girilen verileri yorumlayarak karar verme işlemlerini modelleyen programlardır.

Uzman Sistemler; öneride bulunan, problemi analiz edebilen, sınıflandırabilen, iletişim kurabilen, dizayn yapabilen, tanımlayabilen, inceleyebilen, tahmin yürütebilen, belirleyen, yargılayabilen, öğrenebilen, kontrol edebilen, programlayabilen ve öğreten yazılımlardır. Yani uzmanların düşünce biçimini taklit ederek, özelleşmiş bir alanda önemli problemleri çözmek için geliştirilmektedir.

Uzman Sistemler gerek kullanım biçimi gerekse yapı açısından geleneksel yazılımlardan önemli farklılıklar gösterir. Geleneksel yazılımlar data kullanırlar, Uzman Sistemler ise; bilgi kullanırlar ve sonradan veri girişi mümkün olup esnektir. Bir Uzman Sistem'in geliştirilmesinde takip edilecek genel aşamalar Şekil 2'de gösterilmektedir.

### Şekil 2. Uzman sistemlerin geliştirilmesi



**Kaynak:** Civalek (2005:1332)

#### **1.4.7. Bulanık Mantık**

Klasik Mantık temelinde mevcut olan “dođru-yanlıř”, “var-yok” ya da “1-0” terimleri yer alır. Bulanık Mantık ise bu deđerler arasında esneklik sađlamaktadır. Çünkü gerçek hayatta olasılıklar her zaman bu kadar kesin olmayabilmektedir. Dolayısıyla Bulanık Mantık kuralların esnek ya da bulanık bir řekilde uygulanabilmesini sađlamaktadır. Örneđin klasik mantıkta az ya da çok olarak belirtilen ifadeler, Bulanık Mantık’ta, az, biraz, çok, çok fazla, daha fazla gibi ifadelerle anlatılabilmektedir. Karmařık problemlerin çözümlerinde de Bulanık Mantık’ın bu esnekliđinden faydalanılmaktadır.

#### **1.4.8. Yapay Sinir Ađları**

Yapay Sinir Ađları genel olarak, beynin bir iřlevini yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanmaktadır. Yapay Sinir Ađları’ndaki iřleme elemanları biyolojik olarak insan beynindeki nöronlara karřılık gelmektedir. Nöronlar, beynin haberleřme sistemini oluřturarak sinyal alma, iřlem yapma, ve sinyallerin sinir ađları içinde iletimini sađlarlar (Türkođlu, 2008:49). Beynin iřleyiř kuralları birçok Yapay Sinir Ađı modelinin geliřtirilmesinde kullanılmaktadır. Yapay Sinir Ađları’nın öđrenme özelliđi ise bu sistemin geliřtirilmesinde önemli bir etken olmaktadır. Uygulamada kullanılan Yapay Sinir Ađları ikinci bölümde ayrıntılı olarak inceleneceđinden burada kısaca deđinilmekle yetinilmiřtir.



## **BÖLÜM 2: YAPAY SİNİR AĞLARI**

Bu bölümde, insan beyninin sinir hücresi yapısını örnek alan ve öğrenme kabiliyetine sahip olan Yapay Sinir Ağları'nın ayrıntılı olarak incelenmesi amaçlanmıştır. Bu amaçla bölümde Yapay Sinir Ağları kavramı, tarihsel gelişimi, yapısı ve temel elemanları, çeşitleri, avantajları ve dezavantajları anlatılmıştır.

### **2.1. Yapay Sinir Ağları Kavramı**

Yapay Sinir Ağları (YSA); beynin fizyolojik yapısı, düşünme, hatırlama ve problem çözme yeteneklerini bilgisayara aktarma mantığına dayalı bir yapay zeka teknolojisidir. Yapay Sinir Ağları, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle tabakalar şeklinde düzenlenir (Altan, 2008:143).

Bir başka tanıma göre ise Yapay Sinir Ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir (Öztemel, 2003:29).

Kohonen (1982), yapay sinir ağını, “Yapay Sinir Ağları, paralel olarak bağlantılı ve çok sayıda basit elemanın gerçek dünyanın nesnelileriyle biyolojik sinir sisteminin benzeri yolla etkileşim kuran, hiyerarşik bir organizasyonudur” şeklinde tanımlamıştır. Haykin (1999) ise, “bir sinir ağı, basit işlem birimlerinden oluşan, deneyimsel bilgileri biriktirmeye yönelik doğal bir eğilimi olan ve bunların kullanılmasını sağlayan yoğun bir şekilde paralel dağıtılmış bir işlem” olarak tanımlamıştır.

Yapay Sinir Ağları, yukarıdaki tanımlardan da anlaşılacağı üzere biyolojik sinir sisteminden esinlenerek geliştirilmiştir. Biyolojik sinir ağlarının sinir hücrelerinden oluştuğu gibi Yapay Sinir Ağları da yapay sinir hücrelerinden oluşmaktadır. Yapay Sinir Ağları ve biyolojik sinir ağları arasında hem mimarileri hem de yetenekleri yönünden büyük benzerlikler vardır (Zurada, 1992).

Yapay Sinir Ağları beynin çalışma ilkelerinin sayısal bilgisayarlar üzerinde taklit edilmesi fikri ile ortaya çıkmış ve ilk çalışmalar beyni oluşturan biyolojik hücrelerin matematiksel olarak modellenmesi üzerinde yoğunlaşmıştır (Demir ve Ülengin, 2008:95).

Yapay Zeka alanının bir alt dalını oluşturan YSA teknolojisi öğrenebilen sistemlerin temelini oluşturmaktadır (Sattari ve diğ., 2007:338). YSA, karmaşık ilişkileri öğrenebilmekte, genelleymekte ve böylece daha önce hiç karşılaşmadığı sorulara kabul edilebilir bir hata düzeyiyle cevap bulabilmektedir (Özalp ve Anagün, 2003:30). Yapay Sinir Ağları, matematiksel bir model oluşturmaktadır. Bu nedenle teknik genel bir fonksiyon yaklaşımcı olarak bilinmektedir (Zhang, 1998:35). Yapay Sinir Ağları verilere tamamen bağlı olmayıp, eksik, düzensiz, kısmen hatalı veya gürültülü (erratic) veriyi başarıyla değerlendirebilmektedir. Oysa ki verilerin eksik ve/veya aşırı sapma göstermesi durumlarında klasik yöntemlerle yapılan tahminler, hatalı ve tutarsız sonuçlar verebilmektedir.

YSA'nın öğrenme özelliği, araştırmacıların ilgisini çeken en önemli özelliklerden birisidir (Tortum ve diğ., 2007:58 ). Çünkü teknik, herhangi bir olayın girdi ve çıktıları arasındaki ilişkiyi, doğrusal olsun veya olmasın mevcut örneklerden öğrenerek çözümler üretebilmektedir. Bu nedenle YSA modelleri genellikle doğrusal değildir. Verileri trend veya yapıyı (desen/pattern) iyi tanımlayan bir yöntem olmaları dolayısıyla doğrusal modellere göre tahmin işlemleri için oldukça uygun olduğu söylenebilir (Yurtoğlu, 2005:9). Bu nedenle doğrusal olmayan bir yapıda YSA modelinin kullanımı önem kazanmaktadır (Aslanargün ve diğ., 2007:29).

## **2.2. Yapay Sinir Ağları'nın Tarihsel Gelişimi**

Yapay Sinir Ağları'nın gelişim sürecine bakıldığında ilgili çalışmaların 1970'ten önce başladığı ancak sonrasında yeni yaklaşımlarla daha farklı boyutlarda ilerlediği görülmektedir.

Yapay Zeka konusundaki ilk çalışmalardan biri McCulloch ve Pitts tarafından önerilen, yapay sinir hücrelerini kullanan hesaplama modelidir. Bu model, önermeler mantığı, fizyoloji ve Turing'in hesaplama kuramına dayanmaktadır. Her hangi bir hesaplanabilir fonksiyonun sinir hücrelerinden oluşan ağlarla hesaplanabileceğini ve mantıksal işlemlerinin gerçekleştirilebileceğini göstermektedirler. Bu ağ yapılarının uygun şekilde tanımlanmaları halinde öğrenme becerisi kazanabileceğini de ileri sürmektedirler. Hebb'in, sinir hücreleri arasındaki bağlantıların şiddetlerini değiştirmek için basit bir kural önermesinin ardından, öğrenebilen Yapay Sinir Ağları'nı gerçekleştirme çalışmalarının da başladığı görülmektedir.

1950'lerde Shannon ve Turing bilgisayarlar için satranç programları yazmaktaydılar. İlk Yapay Sinir Ağı temelli bilgisayar SNARC, 1951'de Minsky ve Edmonds tarafından yapılmıştır. 1956 yılında Dartmouth'da düzenlenen toplantıda birçok çalışmanın temelleri atılmakla birlikte, ilk kuram ispatlayan programlardan Logic Theorist (Mantık kuramcısı) Newell ve Simon tarafından tanıtılmıştır. Sonrasında ise Newell ve Simon, insan gibi düşünme yaklaşımına göre üretilmiş ilk program olan Genel Sorun Çözücü'yü (General Problem Solver) geliştirmişlerdir.

1957 yılında Rosenblatt, McCulloch-Pitts'in modeline öğrenme ve hatırlamayı ekleyerek genelleştirmiş ve iki katlı bir perseptronun iki farklı sınıfı ayırt edebileceğini ispatlamıştır. Bu çalışmada, nöronlar arasındaki değişim için basit prensipler kullanılarak, sınıflandırılmış modellere öğrenme yeteneği kazandıran bir ağ yapısı geliştirilmiştir (Sağiroğlu ve diğ., 2003:8).

Simon, fiziksel simge varsayımını ortaya atmış ve bu kuram, insandan bağımsız zeki sistemler yapma çalışmalarıyla uğraşanların hareket noktasını oluşturmuştur.

Bütün bu gelişmelerin ve süreçlerin sonunda bir grup Yapay Zeka araştırmacısı, insan gibi düşünebilen sistemleri araştırmaya devam ederken, diğer bir grup ise ticari değeri olan rasyonel karar alan sistemler (Uzman Sistemler) üzerinde yoğunlaşmıştır.

1959'da Widrow ve Hoff, Adaline olarak isimlendirilen perseptrona benzeyen bir Yapay Sinir Sistemi geliştirmiştir. 1960'larda Grossberg, Adaptif Sistemler Merkezi'nde kendi kendine öğrenebilen, kararlılığını sağlayabilen ve verileri belirli bir aralığa otomatik olarak çekebilen Yapay Sinir Ağları yapıları üzerinde çalışmalar yapmış ve ART, ART-1 ve ART-2 olarak bilinen yapıları geliştirmiştir. 1961'de Steinbuch, Yapay Sinir Ağları'nı bilgi kodlama alanına ilk defa uygulamıştır.

Amari 1967'de biyolojik sinir sistemi aktiviteleri ile ileri düzey matematik bilgisini çalışmalarında birleştirmiş ve ilişkilendirilmiş hafızanın matematiksel analizini ve yarışmacı öğrenme yaklaşımını sunmuştur. 1968 yılında ise Anderson, bir nöronun ağırlıklarının aktivasyon ilişkileri üzerine oluşan bir hafıza modeli önermiştir.

1969'da Fukushima, görüntü işleme için çok katlı bir yapay sinir ağları yapısı sunmuş ve Cognitron ile Neocognitron'u geliştirilmiştir. O yıllarda zeka içeren sistem tasarlama çalışmalarında kullanılan temel yapılarıdaki bazı önemli yetersizliklerin ortaya

çıkmasıyla birçok araştırmacı çalışmalarını durdurmuştur. Bunun en büyük sebeplerinden birisi, Minsky ve Papert'in 1969'da yayınladıkları Perseptronlar isimli kitaplarında, Tek Katmanlı Yapay Sinir Ağları'nın bazı basit problemleri bile çözemeyeceğini göstermeleri ve aynı problemin Çok Katmanlı Yapay Sinir Ağları'nda da beklenilmesi gerektiğini belirtmeleridir.

Bu duraksamalara rağmen 1971 yılında Kohonen, ilişkili hafızalar olan LVQ'yu geliştirmiş daha sonra da yeni versiyonları olan LVQ1, LVQ2, DLVQ gibi yaklaşımları ortaya çıkarmıştır. Hecht-Nielsen, ilk modern elektronik nöro-bilgisayar olan TRW MARK III ile PC tabanlı nöro bilgisayar olan ANZA'yı 1982'de tasarlamışlardır.

Günümüzde ise Yapay Sinir Ağları teorik çalışmaların yanı sıra günlük hayatta da kullanılan sistemler haline gelmektedir. Birçok araştırmacı yapılan sempozyum ve konferanslarda yeni öğrenme tekniklerini ve yeni modelleri tartışmaya sunmaktadır. Yapay Sinir Ağları bu öğrenebilme ve genelleme özellikleri nedeniyle günümüzde birçok bilim alanında geniş uygulama olanağı bulmakta ve karmaşık problemleri başarı ile çözebilme yeteneğini ortaya koymaktadır (Ergezer ve diğ., 2003:14).

### 2.3. Yapay Sinir Ağları'nın Yapısı ve Temel Elemanları

Yapay Sinir Ağı, biyolojik sinir ağına benzer şekilde, belirli bir performans karakteristiğine sahip bilgi işleme sistemi olarak ifade edilmektedir. Yapay Sinir Ağları, sinir biyolojisinin matematik modelleri genelleştirilerek geliştirilmektedir. YSA'lar insan beyninin çalışma prensibini örnek almaktadır. Beyin ve YSA'ların yapısına bakıldığında görülen benzerlikler Tablo 1'de gösterilmektedir.

**Tablo 1. Bilgisayar ve beyin karşılaştırılması**

İşlemler	Beyin	Bilgisayar
İşlem sırası	Paralel ve dağıtılmış	Sıralı
Anahtarlama hızı	1.000 pals/sn	Yaklaşık $10^{-10}$ sn'den az (İşlemci hızıyla artmaktadır)
Hesaplama	Düşük	Çok yüksek (İşlemci hızıyla artmaktadır)
Tanımlama	Desen tanımda mükemmel	Karmaşık hesaplama gerekir
Veri işleme	Karışık ve hatalı veriyi işler ve dağıtır	Tam ve doğru veri kabul eder
Hata toleransı	Yüksek	Hata artışıyla performans düşer
Öğrenme	Var	Programlanabilir

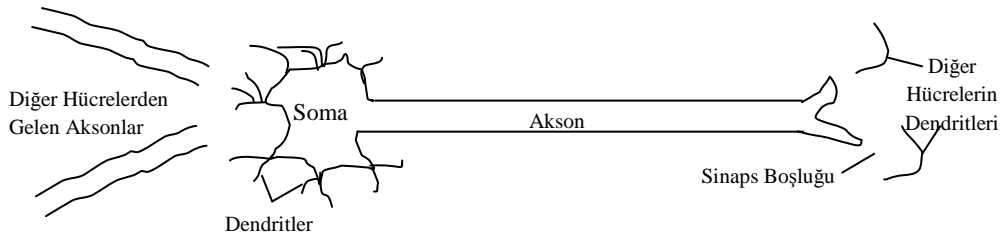
**Kaynak:** Sağiroğlu ve diğ. (2003: 31)

Yapay Sinir Ağları, tıpkı biyolojik sinir sisteminde olduğu gibi, hata tolerans özelliğine sahiptir. Ayrıca, sinir sisteminin kendisinde meydana gelen zararın tolere edilmesini de sağlamaktadır. İnsan doğduğunda yüz milyarlarca beyin hücresine sahiptir. Zamanla hücreler öldüğünde yenisi ile değiştirilememektedir. Hücre ölümü sürekli olmasına rağmen insanın öğrenmesi devam etmektedir. Böyle durumlarda, kaybedilen hücrelerin fonksiyonunu diğer hücreler üstlenmektedir. Aynı şekilde, Yapay Sinir Ağları da küçük zararları karşılayabilecek şekilde tasarlanabilmekte veya büyük zararlar karşısında Yapay Sinir Ağı tekrar eğitilebilmektedir (Chen, 1992:320).

YSA'nın ürettiği çıktılar, ağ içerisinde birbirine paralel bağlantılar aracılığıyla dağıtılmaktadır. Ancak ağın ürettiği çıktıların değerleri çok yüksek değerler olmakta ve ağın eğitilebilmesini engellemektedir. Bu nedenle transfer (aktivasyon) fonksiyonları aracılığıyla bu değerler belirli bir aralıkta normalleştirilerek ağın eğitiminin yapılabilmesi sağlanmaktadır (Bayramoğlu, 2007:103).

Beyin veya sinir hücresi ile işlem birimi (yapay hücre) yapıları arasında çok yakın bir benzetim bulunmaktadır. İşlem birimi yapısının anlaşılabilmesi açısından, biyolojik hücre dendritler, çekirdek (soma), akson ve sinaps olmak üzere dört temel bileşene ayrılmaktadır.

### Şekil 3. Biyolojik sinir hücresinin yapısı



**Kaynak:** Durmuş (2005:17)

Şekil 3'te biyolojik sinir hücresinin yapısı gösterilmektedir. Biyolojik sinir hücresinin işleyişi ise; dendritler diğer hücrelerden gelen sinyalleri alır daha sonra çekirdek, dendritlerden gelen sinyalleri toplar. Yeterli giriş alındığında ise hücre etkinleşir ve böylelikle akson üzerinden diğer hücelere sinyal gönderilir şeklinde ifade edilmektedir.

Biyolojik sinir sistemi ile yapay sinir sistemi bileşenleri arasında birçok benzerlik bulunmaktadır. Biyolojik sinir sistemi ve yapay sinir sistemi bileşenleri arasındaki benzerlikler Tablo 2'de gösterilmektedir.

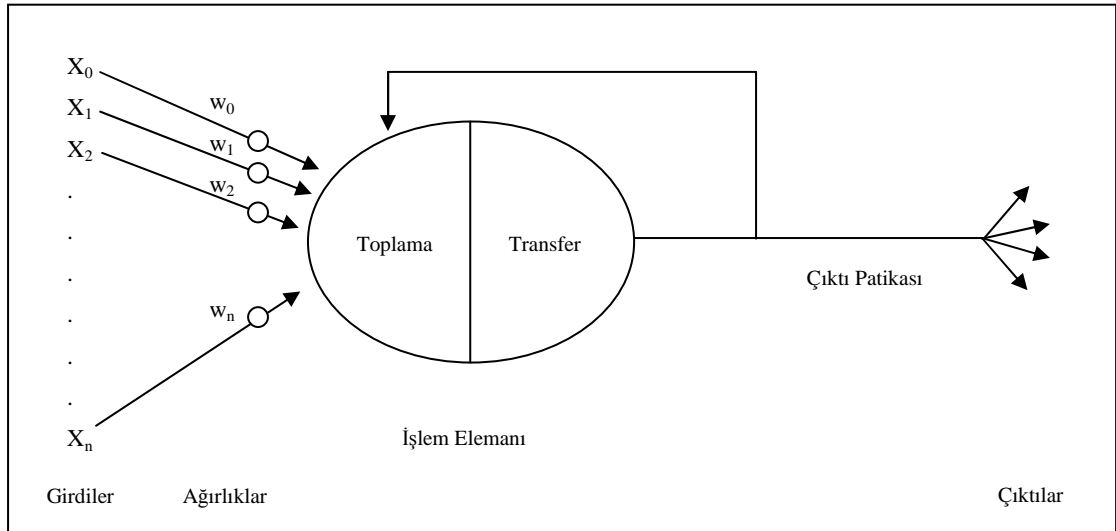
**Tablo 2. Biyolojik sinir sistemi bileşenlerinin YSA'daki karşılıkları**

Biyolojik Sinir Sistemi	Yapay Sinir Ağları
Nöron	İşlemci Eleman
Dendrit	Toplama Fonksiyonu
Hücre Gövdesi	Transfer Fonksiyonu
Aksonlar	Yapay Nöron Çıkışı
Sinapslar	Ağırlıklar

**Kaynak:** Sağiroğlu ve diğ. (2003: 33)

Yapay Sinir Ağları'nın temel işlem elemanı olan yapay nöronlar, biyolojik sinir hücresinin dört temel fonksiyonunu simüle etmektedir. Yapay nöronun temel yapısı Şekil 4'te gösterilmektedir.

**Şekil 4. Yapay nöronun yapısı**



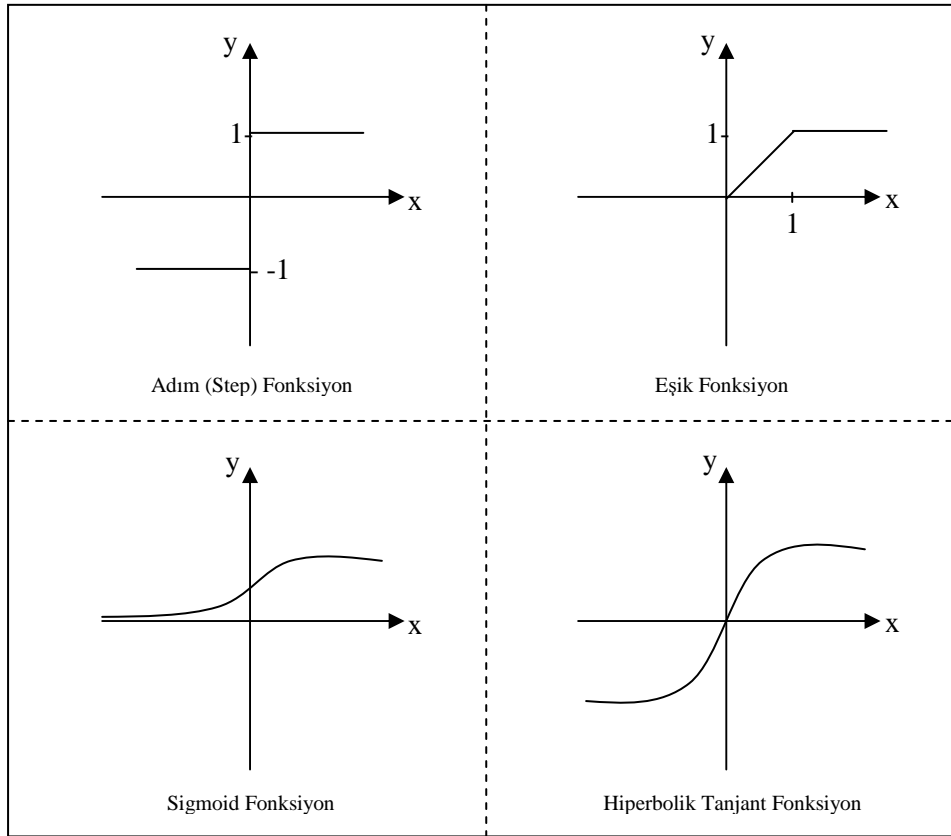
**Kaynak:** Yurtoğlu (2005: 14)

Bir yapay nöron temel olarak girişler, ağırlıklar, toplama fonksiyonu, transfer fonksiyonu ve çıkış olmak üzere beş kısımdan oluşmaktadır. Yapay nöronun işlevleri ise aşağıdaki adımlardan oluşmaktadır:

- Yapay nöron eşzamanlı olarak birçok sayıda girdi almaktadır ve bu girdiler kendi nispi ağırlıklarıyla ağırlıklandırılmaktadır. Böylelikle bazı girdiler diğerlerine göre daha önemli hale gelerek işlem elemanının sinirsel tepki üretmesinde daha etkili olmaktadır.

- Ağırlıklandırmadan sonra değiştirilmiş girdiler toplama fonksiyonuna iletilmektedirler.
- Toplama fonksiyonunun çıktısı transfer fonksiyonuna iletilmektedir. Transfer fonksiyonu aldığı değeri bir algoritma ile gerçek bir çıktıya dönüştürmektedir. Şekil 5’te gösterilen bu fonksiyonlar genellikle doğrusal olmayan eşik, sigmoid, hiperbolik tanjant vb. fonksiyonlardır.
- Transfer fonksiyonundan alınan çıktı dış ortama ya da başka bir nörona girdi olarak gönderilmektedir.

**Şekil 5. Sık kullanılan transfer fonksiyonları**

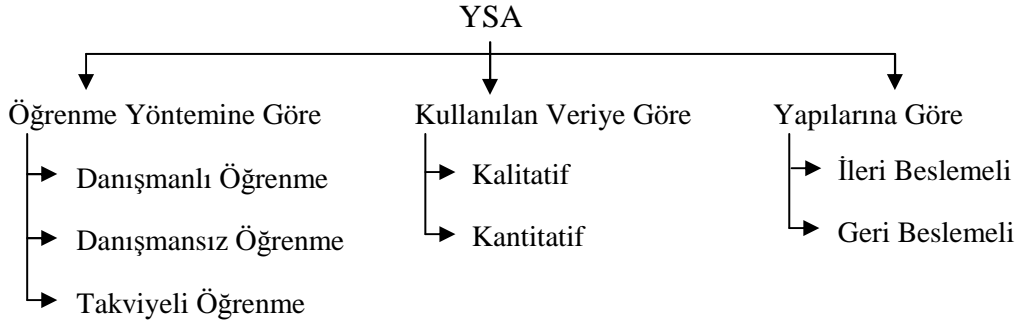


**Kaynak:** Yurtoğlu (2005:18)

#### 2.4. Yapay Sinir Ağı Çeşitleri

YSA’ların çok sayıda çeşitleri vardır. Bu farklılıkların kaynağı mimarisi, öğrenme yöntemi, bağlantı yapısı vb. olabilmektedir. Şekil 6’da Yapay Sinir Ağları’nın sınıflandırılması özet olarak gösterilmektedir.

## Şekil 6. Yapay Sinir Ağları'nın sınıflandırılması



Genel olarak, YSA'lar üç ana kritere göre sınıflandırılmaktadır. Bu kriterlerden biri öğrenme yöntemidir. Temel olarak üç çeşit öğrenme algoritması vardır. Bunlar danışmanlı öğrenme, danışmansız öğrenme ve takviyeli öğrenmedir. Her yöntemin kullandığı öğrenme kuralı değişebilmekteyse de, YSA'lar bu üç algoritmaya göre sınıflandırılırlar (Yurtoğlu, 2005:27).

İkinci bir sınıflandırma ise, ağıın kullandığı veriye göre yapılmaktadır. Temel olarak, kalitatif ve kantitatif olmak üzere iki tür veri vardır. Kalitatif verilerle çalışan ağlar, ister danışmanlı, ister danışmansız öğrenme kullansın, sınıflandırma ağları olarak bilinir. Kantitatif veriler kullanan danışmanlı eğitime ise regresyon olarak adlandırılmaktadır (Yurtoğlu, 2005:28).

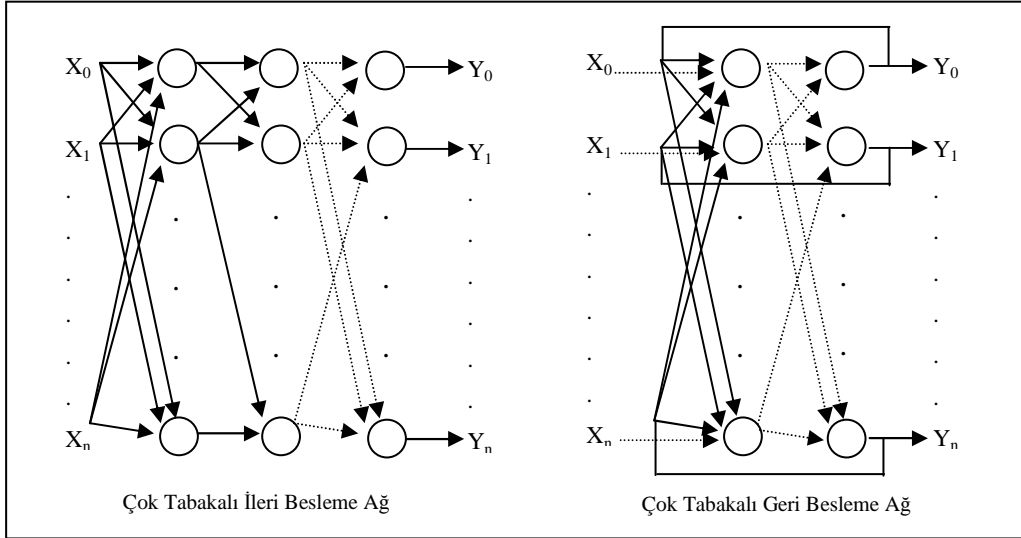
Diğer sınıflandırma kriteri ise ileri beslemeli sinir ağları ve geri beslemeli sinir ağları olmak üzere ağıın yapısına göre adlandırılmaktadır. Ortaya çıkışından sonra, hem etkili hem de çok kullanışlı olmasından dolayı büyük bir popülerite kazanmış ve hala en çok kullanılan ağ türü olarak bilinmektedir (Yurtoğlu, 2005). İleri beslemeli bir ağda işlemci elemanlar genelde katmanlara ayrılmışlardır. İşaretler, giriş katmanından çıkış katmanına doğru tek yönlü bağlantılarla iletilir (Sağiroğlu ve diğ., 2003:43). Bu ağlar girdi veriye genellikle hızlı bir şekilde karşılık üretirler. İleri beslemeli sinir ağlarının en önemli özelliği doğrusal olmayan yapı içeren problemlerde de etkili olabilmesidir (Sattari ve diğ., 2007:340).

Geri beslemeli sinir ağı, çıkış ve ara katman çıkışların, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır (Sağiroğlu ve diğ., 2003:43). Geri beslemeli ağlarda, bağlantılar döngü içerirler ve hatta her seferinde yeni veri kullanabilmektedirler. Bu ağlar, döngü sebebiyle girdinin karşılığını yavaş bir şekilde oluştururlar. Bu yüzden, bu tür ağların eğitime süreci daha uzun olmaktadır. Ayrıca, hem



ileri besleme hem de geri yayılma olarak tanımlanabilecek ağ yapıları da mevcuttur (Yurtoğlu, 2005:28). Şekil 7’de ileri beslemeli ve geri beslemeli ağ yapıları gösterilmektedir.

**Şekil 7. İleri beslemeli ve geri beslemeli ağ yapıları**



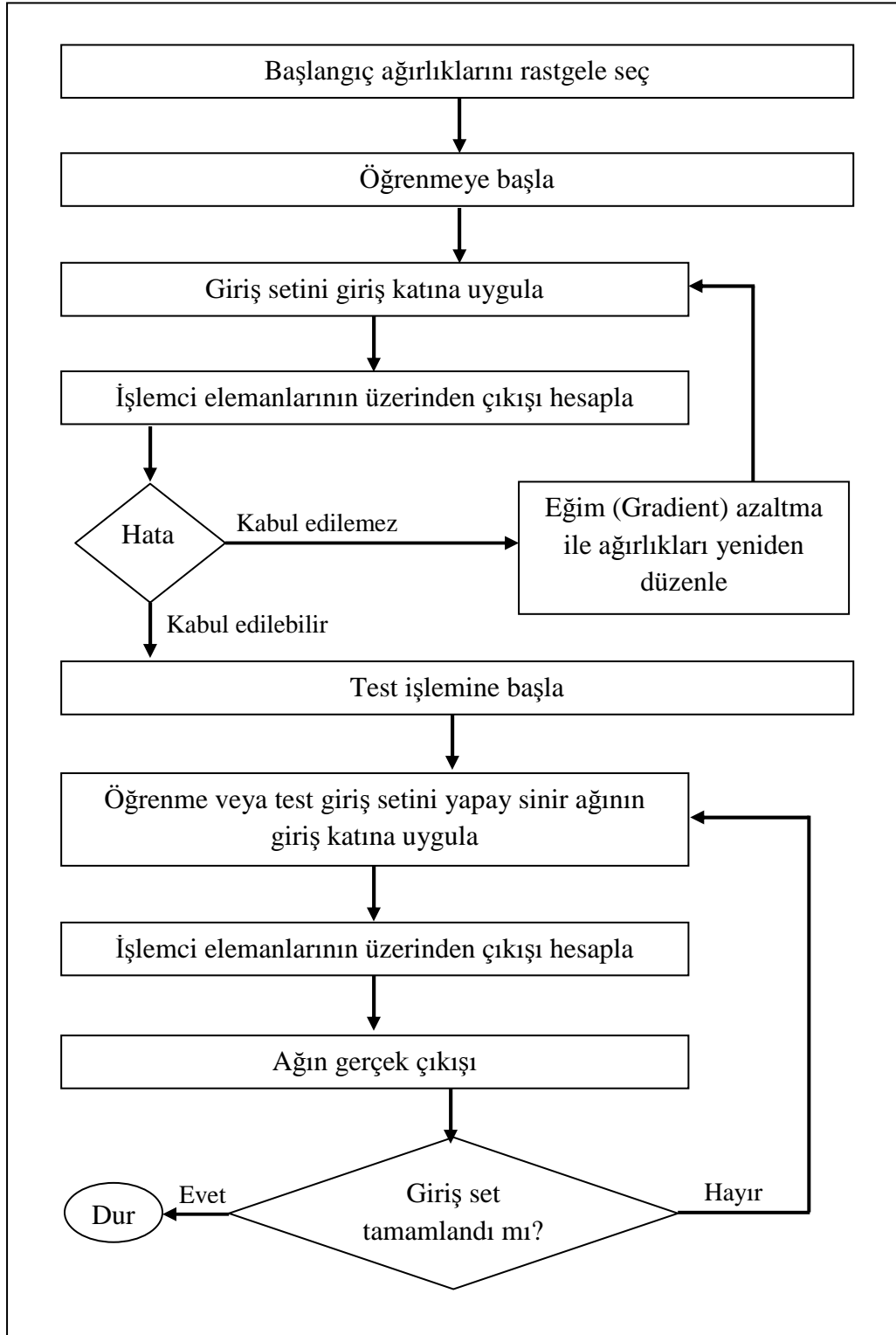
**Kaynak:** Yurtoğlu (2005: 28)

YSA çeşitleri arasında en çok bilinen ve kullanılan ağlar arasında hata algoritması genellikle geri yayılma ile eğitilen çok tabakalı geri yayımlı ağ, radyal tabanlı fonksiyon, Hopfield ve Kohonen olarak ifade edilmektedir. Şekil 8’de bir ağda geri yayımlı bir öğrenme algoritmasının uygulamaları aşamaları gösterilmektedir.

İleri beslemeli geri yayılma mimarisi 1970’li yıllarda geliştirilmiştir. Daha sonra Rumelhart, Hinton ve Williams (1986) tarafından geliştirilmiştir. Ortaya çıkışından sonra, hem etkili hem de çok kullanışlı olmasından dolayı büyük bir popülerite kazanmıştır ve hala en çok kullanılan ağ türü olarak bilinmektedir. Çok sayıda farklı uygulama alanında kullanılmaktadır ve en büyük özelliği doğrusal olmayan yapı içeren problemlerde de etkili olabilmesidir (Yurtoğlu, 2005:29).

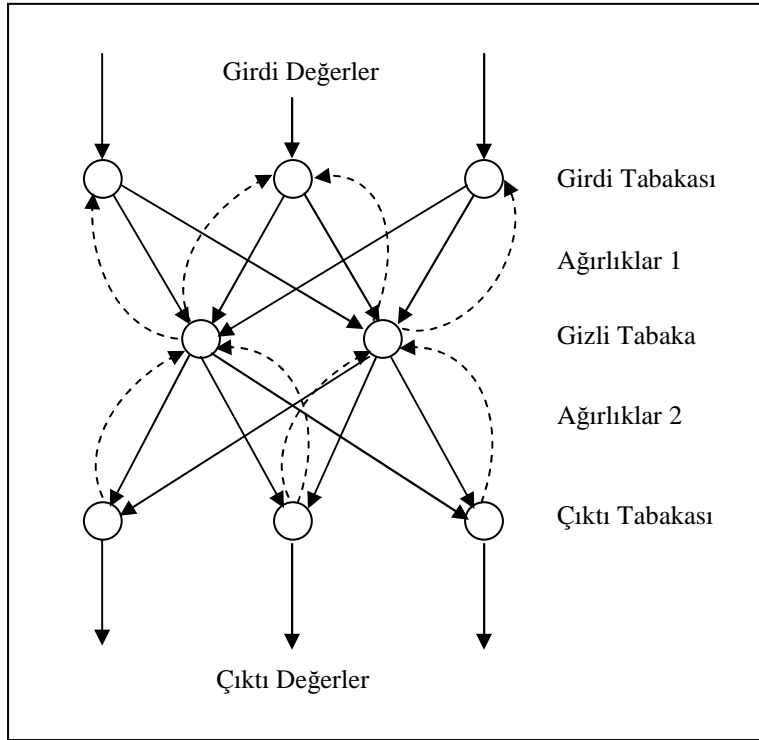
Şekil 9’da bir girdi tabakası, bir gizli tabaka ve bir çıktı tabakasını içeren bir geri yayılma ağ yapısı gösterilmektedir. Tabakalar halinde düzenlenmiş daireler, işlem elemanlarını yani nöronları temsil etmektedir. Girdi tabakasında üç nöron, gizli tabakada iki nöron, çıktı tabakasında ise üç nöron bulunmaktadır ve ağdan üç değişken olarak çıktı alınmaktadır. Geri yayılma sadece eğitime sürecinde kullanılır.

Şekil 8. Bir ağda geri yayımlı bir öğrenme algoritmasının uygulanması



Kaynak: Sağiroğlu ve diğ. (2003: 80)

**Şekil 9. İleri beslemeli geri yayılma ağların genel yapısı**



**Kaynak:** Yurtoğlu (2005: 30)

Ağın genel yapısında, tabaka sayısı ve tabakaların içerdiği işlem elemanı sayısı ağın performansı açısından önem arz etmektedir. Uygulamalar için net bir seçim kriteri bulunmamaktadır. Uygulamalar sonucunda ortaya çıkmış ve araştırmacılar tarafından benimsenmiş bazı kurallar bulunmaktadır. Bu kurallar;

- Girdi ve çıktı verileri arasındaki ilişkinin karmaşıklık derecesi arttıkça, tabakaların içerdiği işlem elemanı sayısı da artmalıdır.
- Modellenen konu değişik safhalara ayrılabilirse, tabaka sayısının artırılması gerekebilir.
- Eldeki eğitim verisinin genişliği, gizli tabakalardaki toplam nöron sayısı için bir üst limit kriteri oluşturur.

olarak ifade edilmektedir (Yurtoğlu, 2005:32).

Geri yayılma algoritmasında, genelde Delta Kuralı kullanılmaktadır. Delta Kuralı, temel olarak, ilgili bağlantı ağırlığının ayarlanması için gerekli olan düzeltme miktarını

formüller. Delta Kuralı ile bağlantıların ayarlanmasının matematiksel gösteriminde nöron(i) ve nöron(j) arasındaki bağlantı için düzeltme miktarı aşağıdaki şekilde hesaplanmaktadır;

$$\begin{array}{l} \text{Ağırlık} \\ \text{Düzeltilme} \\ \text{Miktarı} \end{array} = \begin{array}{l} \text{Öğrenme} \\ \text{Oranı} \\ \text{Parametresi} \end{array} \times \begin{array}{l} \text{Yerel} \\ \text{Değişim} \\ \text{(Gradient)} \end{array} \times \begin{array}{l} \text{Nöron(j)} \\ \text{İçin Girdi} \\ \text{Sinyali} \end{array}$$

veya matematiksel formül olarak;

$$\Delta w_{ji}(n) = \eta * \delta_j(n) * Y_i(n) \text{ şeklindedir.}$$

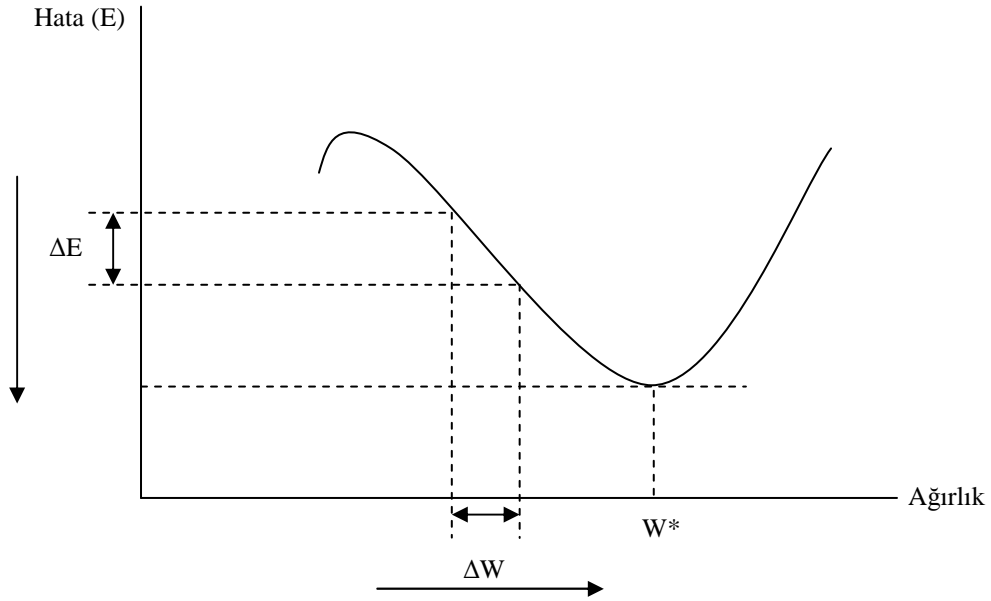
Burada dikkat edilmesi gereken nokta, yerel değişimin hesaplanma şeklinin nöron(j)'nin çıktı veya gizli nöron olmasına göre değişiklik gösterdiğidir. Buna göre:

- Nöron(j) bir çıktı nöronu ise, yerel değişim, nöron(j)'ye ait hata sinyali ve fonksiyon sinyalinin türevinden hesaplanmaktadır.
- Nöron(j) bir gizli nöron ise, yerel değişim, fonksiyon sinyalinin türevi ve bir sonraki tabakadaki nöronlara ait değişimlerin ağırlıklı toplamı kullanılarak hesaplanır (Yurtoğlu, 2005:33).

Çok Katmanlı Algılayıcı Ağları'nın kendisine gösterilen girdi örneği için beklenen çıktıyı üretmesini sağlayacak ağırlık değerleri bulunmaktadır. Başlangıçta bu değerler rastgele atanmakta ve ağa örnekleri gösterildikçe ağın ağırlıkları değiştirilerek zaman içinde istenen değerlere ulaşması sağlanmaktadır. İstenen ağırlık değerlerinin ne olduğu bilinmemektedir. Bu nedenle YSA'nın davranışlarını yorumlamak ve açıklamak mümkün olamamaktadır. Bunun temel nedeni, bilginin ağ üzerinde dağılmış olması ve ağırlık değerlerinin kendi başlarına herhangi bir anlam ifade etmemesidir. Ağ ile ilgili bilinen konu, problem uzayında en az hata verebilecek ağırlık değerlerinin bulunmasıdır.

Ağın  $W^*$  değerine ulaşması istenmektedir. Bu ağırlık değeri problem için hatanın en az olduğu noktadır. Bu nedenle her iterasyonda  $\Delta W$  kadar değişim yaparak hata düzeyinde  $\Delta E$  kadar bir hatanın düşmesi sağlanmaktadır. Buradaki  $W^*$  en az hatanın olduğu aralık vektörünü göstermektedir (Öztemel, 2003:82). Hatanın en az değerinin grafiksel ifadesi Şekil 10'da, daha karmaşık bir hata düzeyi ise Şekil 11'de gösterilmektedir.

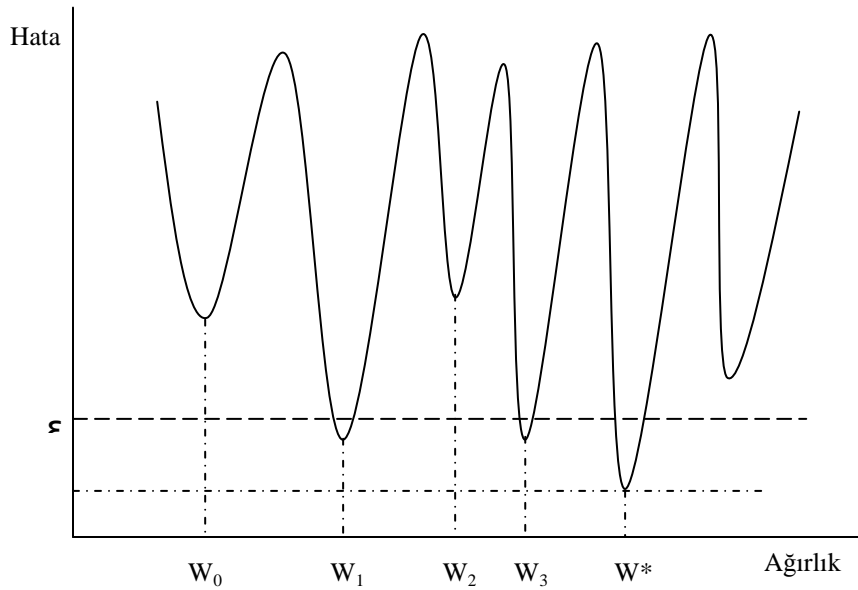
### Şekil 10. Öğrenmenin hata uzayındaki gösterimi



**Kaynak:** Öztemel (2003:82)

Görüldüğü gibi, problemin çözümü için en az hatayı veren ağırlık vektörü  $W^*$  olmasına rağmen, pratikte bu hata değerini yakalamak çoğu zaman mümkün olmamaktadır. Ağın en iyi çözümüne nasıl ulaşılacağı konusunda kaynaklarda bilgi bulunmamaktadır. En iyi çözüm eğitim sırasında bulunmaya çalışılmaktadır. Problemlere üretilen çözümlerde belirli bir hata payı olduğu kabul edilmektedir.

### Şekil 11. Çok boyutlu hata uzayı



**Kaynak:** Öztemel (2003:83)

Şekil 11’de çok boyutlu hata uzayında yerel çözümler gösterilmektedir. Bu çözümlerden  $W_0$  ve  $W_2$  çözümlerinin hataları, kabul edilebilir hata düzeyinin üzerinde olduğundan dolayı çözüm olarak kabul edilmemektedir.  $W_1$  ve  $W_3$  çözümlerinin hataları ise kabul edilebilir hata düzeyinin altında olduğundan kabul edilebilir çözümler olarak ifade edilmektedir.

YSA’lar her zaman en iyi çözümü üretmeseler de problemde birden fazla çözüm üretilebilmektedirler. Ayrıca üretilen çözüm en iyi çözüm olsa bile bunun bilinmesi mümkün olamamaktadır. En iyi sonucun bulunamamasının nedenleri arasında (Öztemel, 2003:83);

- Problem eğitilirken bulunan örneklerin problem uzayını tamamen temsil edememesi,
- Oluşturulan ağ için doğru parametreler seçilememiş olması,
- Ağın ağırlıklarının başlangıçta tam istenilen şekilde belirlenmemiş olması ve
- Ağın topolojisinin yetersiz seçilmiş olması,

söylenmektedir.

Yapay sinir ağlarında ağın öğrenme performansını başlangıç değerleri, öğrenme ve momentum katsayılarının belirlenmesi etkilemektedir. Geri yayılma ağlarının bazı kısıtlamalara da sahip olduğu belirtilmektedir. Geri yayılma mekanizması oldukça geniş bir girdi-çıkı veri seti ile geniş çaplı bir yönlendirmeli eğitime ihtiyaç duymaktadır.

## **2.5. Yapay Sinir Ağları’nın Avantajları**

YSA metodolojisi, özellikleri ve yapabildikleri sayesinde önemli avantajlar sağlamaktadır. Bu özelliklere bakıldığında; doğrusal olmama, öğrenme, genelleme, esneklik, uygulanabilirlik, hata toleransı, analiz ve tasarım kolaylığı ve hafıza olarak karşımıza çıkmaktadır. YSA, her türlü bilgiyi işlemek ya da analiz etmek amacıyla kullanılmaktadırlar. İş hayatı, finans, endüstri, eğitim ve karışık problemler bilim alanlarında, bulanık veya mevcut yöntemlerle çözülemeyen problemlerin çözümünde, doğrusal olmayan sistemlerde başarıyla uygulanmaktadır (Elmas, 2003b:26).

İşlem hücrelerinin birleşmesinden meydana gelen YSA’ların yapısına bakıldığında

hücrelerin doğrusal olmadığı ve dolayısıyla da bu yapının tüm ağa yansıdığı görülmektedir. Diğer taraftan sisteme gerekli bağlantılar ya da ağırlıklar önceden tasarlanarak verilememektedir. YSA'lar istenen çözüme ulaşabilmek için problemden aldığı eğitim örneklerini kullanarak öğrenmektedir. YSA'lar bu özellikleri sayesinde doğrusal olmayan karmaşık problemlerin çözümünde kullanılabilir. Kolayca yapılabilen ama geleneksel metotların uygulanamadığı basit işlemler için de çözüm avantajı sağlamaktadır.

YSA'lar geleneksel işlemcilerden farklı şekilde işlem yapmaktadırlar. Geleneksel işlemcilerde, tek bir merkezi işlem elemanı her hareketi sırasıyla gerçekleştirir. YSA modelleri ise, her biri büyük bir problemin bir parçası ile ilgilenen çok sayıda basit işlem elemanlarından oluşma ve bağlantı ağırlıklarının ayarlanabilmesi gibi özelliklerinden dolayı önemli derecede esnek bir yapıya sahiptirler. Bu esnek yapı sayesinde ağın bir kısmının zarar görmesi modelde sadece performans düşüklüğü yaratır. Modelin işlevini tamamen yitirmesi söz konusu olmaz. Ayrıca, toplam işlem yükünü paylaşan işlem elemanlarının birbirleri arasındaki yoğun bağlantı yapısı sinirsel hesaplamaların temel güç kaynağıdır. Bu yerel işlem yapısı sayesinde, YSA yöntemi en karmaşık problemlere bile uygulanabilmekte ve tatminkar çözümler sağlayabilmektedir (Yurtoğlu, 2005:35).

Belirli bir problemi, programlama yerine direkt olarak mevcut örnekler üzerinden eğitilerek öğrenirler (Baylar ve diğ., 1999:3). YSA'lar öğrenme yetenekleri sayesinde, bilinen örnekleri kullanarak daha önce karşılaşılmamış durumlarda genelleme yapabilmektedir. Bu özellikleri ile hatalı veya kayıp veriler için çözüm üretebilmektedirler. YSA modelleri sınırsız sayıda değişken ve parametre ile çalışabilmektedir. Bu sayede mükemmel bir öngörü doğruluğu ile genel çözümler sağlanabilmektedir.

## **2.6. Yapay Sinir Ağları'nın Dezavantajları**

YSA'nın oluşturulmasında, model seçilmesinde, ağın topolojisinin belirlenmesinde bir kurallar seti yoktur. Bu kurallar, kullanıcının tecrübesine dayalı olarak düzenlenmektedir. Ağın, davranışlarının ve işleyişlerinin açıklanamaması, ağa duyulacak güveni azaltmaktadır. Hayati önem taşıyan bazı problemlerde bu nedenden dolayı kullanılamamaktadır.

Eđitim iřlemlerinin gerekleřtirilmesi uzun zaman alabilmektedir. Ayrıca problem özümlelerinde optimum sonuçlar garanti edilememektedir. Örneklerin bulunmasının gü olduđu durumlarda ve problemi dođru temsil eden örnek bulunmaması durumunda; sađlıklı özümler üretmek mümkün olamamaktadır. Bunların yanısıra YSA'larda uygulama başarısı, uygun parametrelerin seçimine bađlıdır. Uygun parametrelerin seçimi ise kolaylıkla yapılamamaktadır. Uygun parametrelerin seçiminde karşılaşılan birçok gülük bulunmaktadır. Bu gülükler (Sađırođlu ve diđ., 2003:102);

- Probleme uygun olan YSA yapısı ve mimarisi seçimi,
- Problemin kabul edilebilir özümü için YSA giriş ve ıkıř sayılarının en uygun veya en az sayıda seçimi,
- Ara katman nöron sayılarının en uygun sayıda belirlenmesi,
- Ara katman sayısının seçimi,
- Kullanılacak öğrenme algoritmasının YSA yapısına uygun olması,
- En uygun öğrenme algoritması parametrelerinin seçimi,
- Seçilen veri kodlama yapısı,
- Veri normalizasyon yaklaşımı,
- Seçilen transfer fonksiyonunun yapısı,
- Toplama fonksiyonu tipi,
- Uygun performans fonksiyonu seçimi,
- Uygun iterasyon veya epok sayısı seçimi,
- Ön işleme ve son işleme işlemlerine, uygun veri tipinin ve sayısının belirlenmesi,
- Ađ yapılandırması ve aynı ađ yapısı içerisinde farklı aktivasyon fonksiyonu seçimi,

olarak ifade edilmektedir.



## 2.7. Yapay Sinir Ağları Öğrenme Algoritmaları

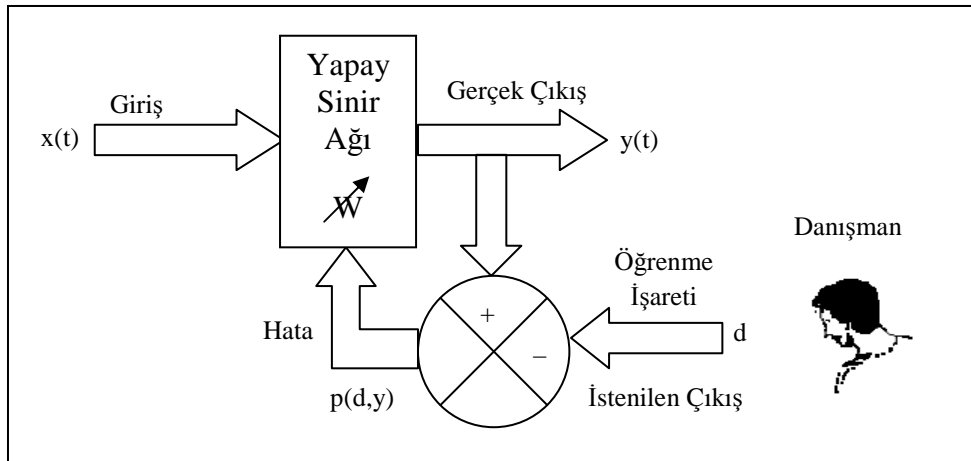
Öğrenme; gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanmaktadır. Öğrenme sürecine bakıldığında kurallar, gözlem ve eğitime göre ağıdaki ağırlıkların değiştirilmesi sağlanmaktadır. Bunun için genel olarak üç öğrenme metodundan ve bunların uygulandığı değişik öğrenme kurallarından söz edilmektedir. Bu öğrenme kuralları danışmanlı öğrenme, danışmansız öğrenme ve takviyeli öğrenme olmak üzere üç başlık altında incelenmektedir.

### 2.7.1. Danışmanlı Öğrenme (Supervised Learning)

Danışmanlı öğrenmede YSA'ya ne öğrenmesi gerektiği örnek bir çıkış kullanılarak verilmektedir. En uygun çıkışı elde etmek için hata oranına göre nöronlar arası bağlantı ağırlığının düzenlenmesinde öğrenme algoritmasına ve böylelikle bir öğretmene ya da danışmana ihtiyaç duyulmaktadır.

Danışmanlı öğrenme, giriş vektörlerinin data setini ve ağı eğitmek için çıkış vektörlerinin cevabını kullanır. Ağırlık matrisi, toplam ağ hatasının kabul edilebilir hatadan daha büyük olduğu sürece güncelleştirilir (Fırat ve Güngör, 2004:3271). Şekil 12'de danışmanlı öğrenme yapısı gösterilmektedir.

#### Şekil 12. Danışmanlı öğrenme yapısı



**Kaynak:** Sağiroğlu ve diğ. (2003:81)

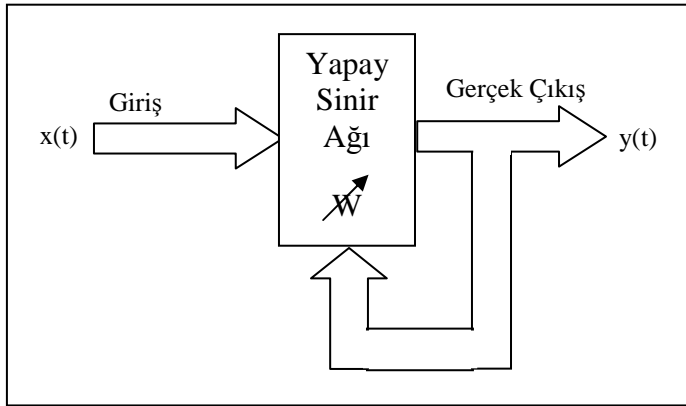
Danışmanlı öğrenme algoritmalarına Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geri besleme algoritması örnek oluşturmaktadır.

### 2.7.2. Danışmansız Öğrenme (Unsupervised Learning)

Danışmansız öğrenme türünde girişe verilen örnekten elde edilen çıkış bilgisine göre ağ, sınıflandırma kurallarını kendi kendine geliştirmektedir. Öğrenme süresince sadece giriş bilgileri verilmektedir. Bu öğrenme algoritmalarında, istenilen çıkış değerinin bilinmesine gerek duyulmadığından danışmanın da olması gerekmemektedir. Danışmansız olarak eğitilebilen ağlar, istenen ya da hedef çıkış olmadan giriş bilgilerinin özelliklerine göre ağırlık değerlerini ayarlar (Elmas, 2003b:96).

Grossberg tarafından geliştirilen ART (Adaptive Resonance Theory) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı danışmansız öğrenmeye örnek olarak verilmektedir. Şekil 13'te danışmansız öğrenme yapısı gösterilmektedir.

**Şekil 13. Danışmansız öğrenme yapısı**



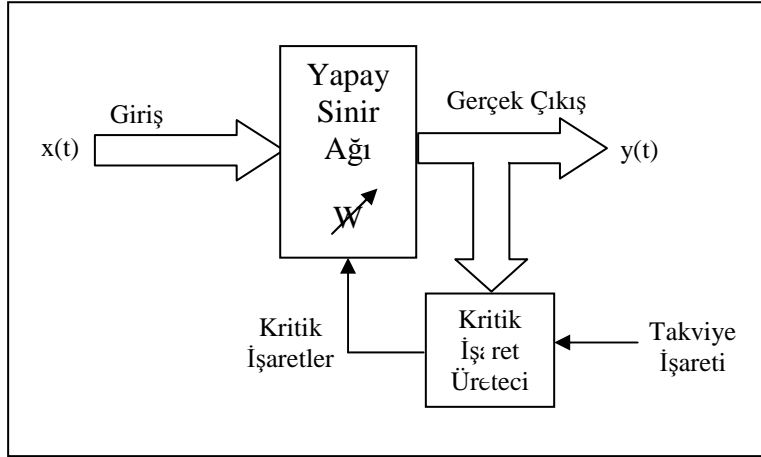
**Kaynak:** Sağıroğlu ve diğ. (2003:81)

### 2.7.3. Takviyeli Öğrenme (Reinforcement Learning)

Takviyeli öğrenme kuralı danışmanlı öğrenme kuralına benzemektedir. İstenilen çıkış değerlerinin bilinmesine gerek duyulmamaktadır. YSA'ya elde edilen çıkışın verilen girişe karşılık iyiliğini değerlendiren bir kriter kullanılmaktadır.

Optimizasyon problemlerini çözmek için Hinton ve Sejnowski'nin geliştirdiği Boltzmann kuralı veya Genetik Algoritma takviyeli öğrenmeye örnek olarak verilmektedir. Şekil 14'te takviyeli öğrenme yapısı gösterilmektedir.

**Şekil 14. Takviyeli öğrenme yapısı**

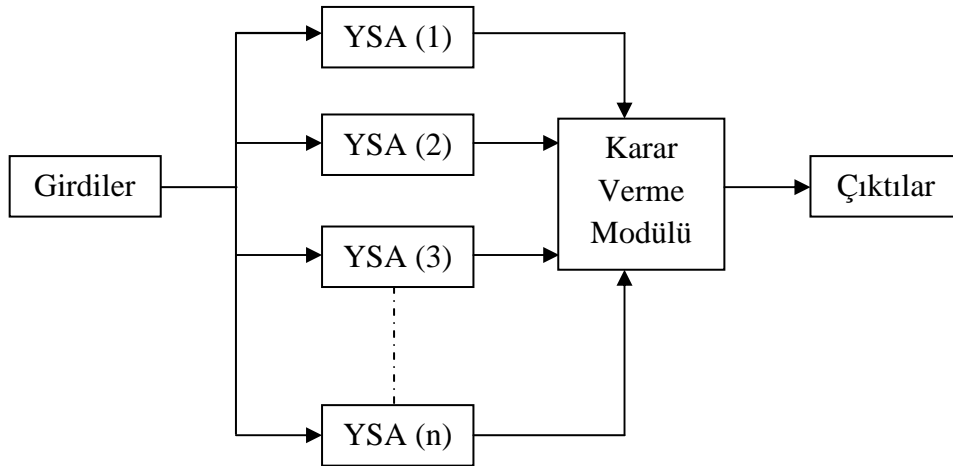


**Kaynak:** Sağıroğlu ve diğ. (2003:81)

## 2.8. Birleşik Yapay Sinir Ağları

Karmaşık problemler daha iyi sonuç üretebilmek için birden fazla ağın eğitilerek birlikte kullanılması gerekmektedir. Birden fazla ağın aynı problemde çözüm üretmesini sağlayan entegre sisteme “Birleşik Ağlar” denilmektedir. Birleşik Ağlar’ın yapısına bakıldığında birden fazla ağın aynı anda eğitildiği ve ortak bir karara varıldığı görülmektedir. Şekil 15’te Birleşik Sinir Ağları’nın genel yapısı gösterilmektedir.

**Şekil 15. Birleşik sinir ağlarının elemanları**



**Kaynak:** Öztemel (2003:188)

Birleşik Ağ oluşturmak için aynı ya da farklı en az iki yapay sinir ağının bir araya gelmesi yeterlidir. Verilen kararların daha rahat ortak bir karara dönüştürülmesi için 3 adet YSA ağının kullanılması önerilmekle beraber böyle bir zorunluluk

bulunmamaktadır. Bazı durumlarda ağa sunulan bir örnek, bir ağ tarafında tanınmaz iken diğer ağ tarafından tanınmaktadır. Üçüncü bir ağın kararı bu durumda önemli olmaktadır. Bu ağın kararı hangi ağa yakın ise o zaman birleşik ağın kararı o yönde olacaktır (Öztemel, 2003:188). Birleşik Ağların eğitilmesinde ve test edilmesinde herhangi özel bir algoritmaya gerek duyulmamaktadır. Birleşik ağı oluşturan her ağ kendi öğrenme kurallarına göre eğitilmektedir.

Problem çözümünde kullanılan girdiler birleşik ağı oluşturan her ağa birbirlerinden bağımsız olarak verilmekte ve bağımsız çıktılar alınmaktadır. Sonrasında bu çıktılar bir araya getirilerek karar verilmektedir. Verilen kararda oy birliği anlayışı uygulanmaktadır. Süreçteki en önemli adım ise alınan farklı kararları tek bir sonuca indirgeyebilmektir.

## **2.9. Yapay Sinir Ağları'nın Uygulama Alanları**

YSA'lar, çözümü güç ve karmaşık olan ya da ekonomik olmayan çok farklı alanlardaki problemlerin çözümüne uygulanarak başarılı sonuçlar ortaya çıkarabilmektedir. Genel olarak bakıldığında YSA'ların arıza analizi ve tespitinde, tıp alanında, savunma sanayinde, haberleşmede, üretimde, otomasyon ve kontrol gibi birçok alanda uygulandığı görülmektedir.

Entegre devrelerin arıza analizinde YSA, cihazın ya da elemanın düzenli (doğru) çalışma şeklini öğrenmektedir ve bu sayede sistemlerde meydana gelebilecek arızaların tanımlanmasında kullanılabilir. Tıp alanında ise EEG ve ECG gibi tıbbi sinyallerin analizi, kanserli hücrelerin analizi, protez tasarımı, transplantasyon zamanlarının optimizasyonu ve hastanelerde giderlerin optimizasyonu v.s gibi uygulama yeri bulmuştur (Subaşı, 2010:24).

Ayrıca, silahların otomasyonu ve hedef izleme, nesnelere/görüntüleri ayırma ve tanıma, yeni algılayıcı tasarımı ve gürültü önleme, görüntü ve veri sıkıştırma, otomatik bilgi sunma servisleri, üretim sistemlerinin optimizasyonu, ürün analizi ve tasarımı, ürünlerin kalite analizi ve kontrolü, planlama ve yönetim analizi, uçaklarda otomatik pilot sistemi otomasyonu, ulaşım araçlarında otomatik yol bulma/gösterme, robot sistemlerin kontrolü, doğrusal olmayan sistem modelleme ve kontrolü gibi karmaşık problemlerin çözümünde de kullanılmaktadır.

## 2.10. Yapay Sinir Ağları ve Bulanık Mantık

Karmaşık olayların belirli matematiksel ifadelerle tanımlanması ve kesin bir şekilde kontrol altında tutulması mümkün değildir. Bütün teori ve denklemler gerçek dünyayı yaklaşık bir biçimde ifade ederler. İncelenen bir konunun tam ve kesinlikle bilinmemesi bulanıklılığı ifade etmektedir (Erdoğan, 2003:20). Bu belirsizliklerin sözel ifadeler kullanılarak daha belirgin hale gelmesi için geliştirilen mantığa da “Bulanık Mantık” denilmektedir. Bulanık Mantık yaklaşımı, makinelere insanların özel verilerini işleyebilme ve onların deneyimlerinden ve öngörülerinden yararlanarak çalışabilme yeteneği verir (Elmas, 2003a:24). Bu yeteneği kazandırırken sayısal ifadeler yerine sembolik ifadeler kullanır.

Bulanık Mantık, 1965 yılında Lotfi Zadeh’in California Berkeley Üniversitesi’nde “Bulanık Kümeler” adlı makalesini “Information and Control” adlı dergide yayınlamasıyla başlamıştır. Zadeh bu çalışmasında insan beyninin büyük bir bölümünün bulanık olduğunu belirtmiştir. Araştırmalar Bulanık Mantık denetimi ile elde edilen sonuç performansının klasik yöntemlerle elde edilenlere göre daha iyi olduğunu göstermiştir. Zadeh daha sonraları bulanık mantık ve düşünüşün temelini oluşturan Bulanık Algoritması’nı ileri sürmüştür. 1972 yılında Michio Sugeno bulanık ölçüm ve integral kavramlarıyla bulanık konusuna yeni bakış açıları getirmiştir. 1974 yılında, Ebrahim Mamdani Bulanık Mantık’ı ilk kez bir buhar makinesinin kontrol aşamasında kullanmıştır (Jinglu, 1999:86).

Klasik Mantık’ın birçok alanda yetersiz hale gelmesi, insan zekasının işleyişine uygun olmaması sonucu Bulanık Mantık kavramı popüler hale gelmiştir. Yapay Zeka’nın yönlendirici bir unsuru olan Bulanık Sistemler (Fuzzy Systems), klasik küme üyeliğine ve mantığına karşı oluşturulmuş bir seçenektir (Murat ve Uludağ, 2008:4367).

Klasik kümelerde bir nesne 1 değerini alırsa kümenin elemanı, 0 değerini alırsa kümenin elemanı değildir. 0 ve 1 değerlerini alan kesin kümelere karşılık olarak bulanık mantık kümelerinde 0 ve 1 arasında değişebilen değerler vererek üyelik işlevlerini ortaya koymuştur. Bulanık Mantık’ta belirsizlik durumları, bu durumu temsil eden küme elemanlarına üyelik fonksiyonlarının verilmesi ile tanımlanmaktadır. En büyük önem derecesine sahip olan öğelere 1 değeri atanırsa, diğerleri 0 ile 1 arasında değişim göstermektedir. Bu şekilde 0 ile 1 arasındaki değişimin her bir öğe için değerine üyelik

derecesi, ve bunun bir alt küme içindeki değişimine de üyelik fonksiyonu denilmektedir (Elmas, 2003a:33). Şekil 16’da Klasik Mantık ve Bulanık Mantık farkı şematik olarak gösterilmiştir. Görüldüğü gibi Klasik Mantık’ta ara değer yoktur oysa Bulanık Mantık’ta ara değerler söz konusudur.

### Şekil 16. Klasik ve bulanık mantık arasındaki fark



**Kaynak:** Durmuş (2005:34)

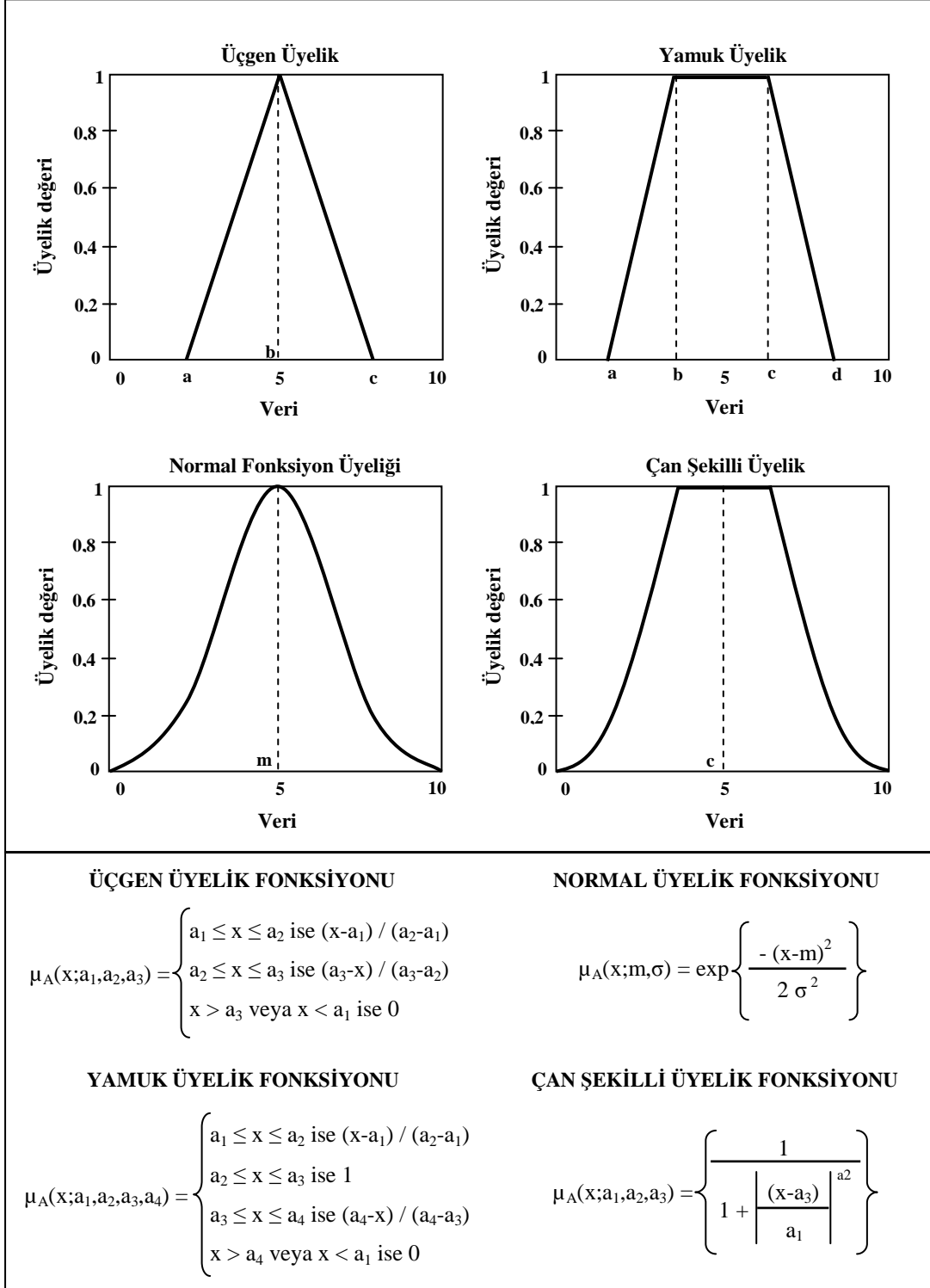
Bulanık Mantık ilkeleri belirsizliği açıklama kabiliyeti açısından üstünlüğü ile öne çıkmaktadır. Teori, matematiksel işlemleri ve programlamayı bulanık alanda uygulamaya da elverişlidir (Sofyalıoğlu, 2009:8). Bulanık küme kuramı, kesin olmamaya yol açan bazı problemler için iyi bir çözüm olarak görüldüğü için; ekonomi, işletme, kontrol teorisi, karar ve bilgi sistemleri, mantık, insan durumu, yapay zeka, uzman sistemler, sosyal bilimler, yöneylem araştırması vb. konularda uygulanabilmektedir. Bulanık Mantık kullanılarak modelleme yapılırken konu hakkında tecrübeye sahip bir bilirkişi bulunmalıdır.

Bulanık Mantık, bir bulanık küme mantığına dayanır. Bulanık küme, kümeye aitlik derecesi üyelik değeri ile tanımlanmış olan kümeyi ifade etmektedir. (Ergülen ve Deran, 2009:229; Karanfil, 1993:35). Kuralları sisteme girmek için ihtiyaç duyduğumuz üyelik fonksiyonu, her girişin katılımının büyüklüğünün grafik temsilidir. Üyelik fonksiyonu, gerekli her giriş ile ağırlığı birleştirmekte, girişler arasındaki fonksiyonel hataları belirlemektedir ve son olarak bir çıkış cevabı oluşturmaktadır. Kurallar, son çıkış kararının bulanık çıktıdaki etkisini belirlemek için, faktörleri ağırlıklandırarak, giriş üyelik değerlerini kullanmaktadır (Jinglu, 1999:88). Üyelik fonksiyonlarının belirlenmesinde sezgi, çıkarım, mertebelenme, açılı bulanık kümeler, yapay sinir ağları, genetik algoritmalar gibi yaklaşımlar kullanılmaktadır (Şenol, 2000:13).

Bunun yanı sıra üyelik derecelerinin belirlenmesinde pek çok formülasyon ve teknik kullanılmaktadır. Bunlar lineer ve lineer olmayan fonksiyonlar olabileceği gibi üçgen ve çan eğrisi olarak adlandırılacak bilinen şekillerde de olabilmektedir (Gönül ve Çelebi, 2003:112).

Şekil 17’de sık kullanılan üyelik fonksiyonları ve fonksiyonlara ait formüller gösterilmektedir.

**Şekil 17. Üyelik fonksiyonları çeşitleri**

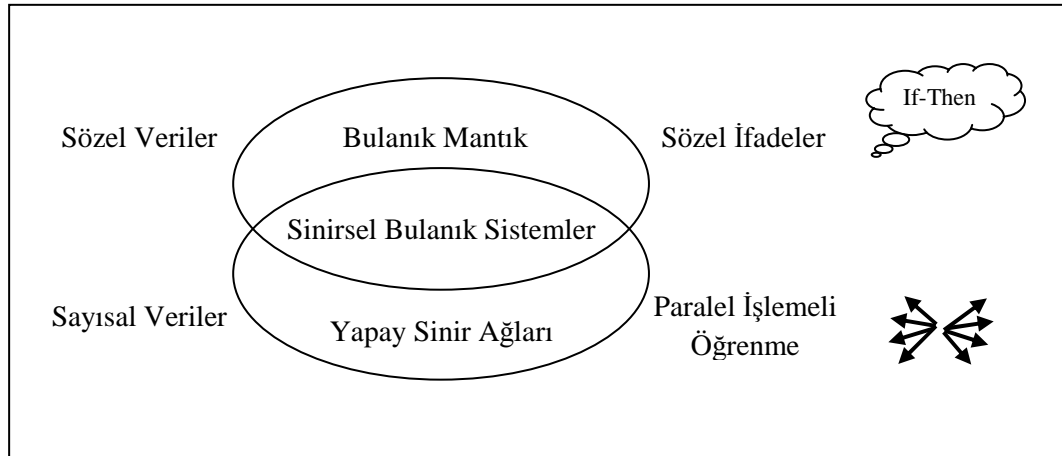


**Kaynak:** Baykal ve Beyan (2004:79)

Bulanık Mantık yaklaşımının ve Yapay Sinir Ağları'nın birçok alanda birlikte kullanımları Sinirsel Bulanık Sistemleri ortaya çıkarmıştır. Sinirsel Bulanık Mantık yaklaşımı, Yapay Sinir Ağları'nın öğrenme yeteneği, en uygunu bulma ve bağlantılı yapılar gibi, bulanık mantığın insan gibi karar verme ve uzman bilgisi sağlama kolaylığı gibi üstünlüklerinin birleştirilmesi fikrine dayanmaktadır. Bu yolla, Bulanık Denetim Sistemleri'ne, sinir ağlarının öğrenme ve hesaplama gücü verilebilirken, sinir ağlarına da bulanık denetimin insan gibi karar verme ve uzman bilgisi sağlama yeteneği kazandırılmaktadır (Elmas, 2003a:165).

Sinirsel Bulanık Sistemler, sayısal ve sözel verilerin birleştirilmesine olanak vermektedir. Aynı zamanda sayısal verilerden bulanık bilginin çıkarılmasını sağlamaktadır. Sinirsel Bulanık Sistemler, her iki yaklaşımın da faydalarını toplayıp, birleştirmektedir. Sinirsel Bulanık Sistemler'in, Bulanık Mantık ve Yapay Sinir Ağları arasındaki ilişkisi ise Şekil 18'de gösterilmektedir.

**Şekil 18. Sinirsel bulanık sistemlerin sinir ağları ve bulanık mantık ile ilişkisi**

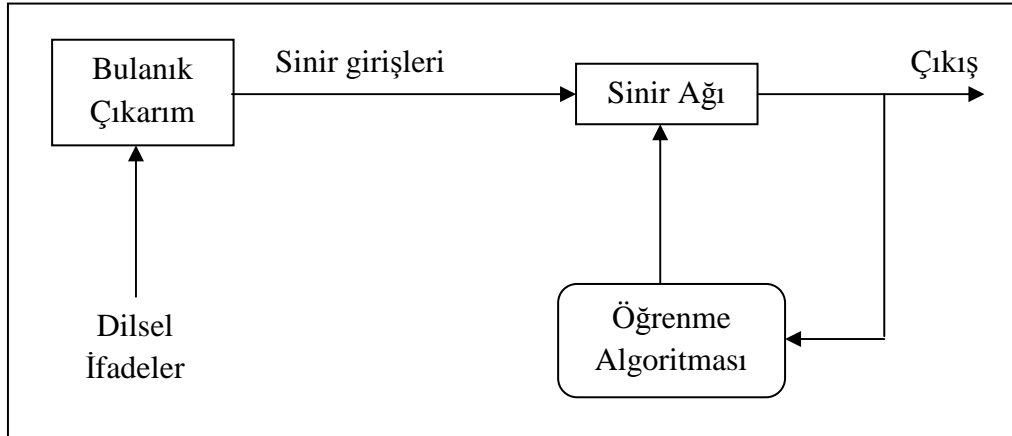


**Kaynak:** Tortum ve diğ. (2007:146)

Sinirsel Bulanık Sistem temel olarak iki yapıdan oluşmaktadır. Şekil 19'daki birinci yapıya bakıldığında, bulanık çıkarımın dilsel ifadelerle oluşturduğu çıktıların çok katmanlı sinir ağına giriş vektörü olarak verildiği görülmektedir. Şekil 20'deki ikinci yapıda ise çok katmanlı sinir ağının çıktıları bulanık çıkarım mekanizmasını sürmektedir (Elmas, 2003a:167).

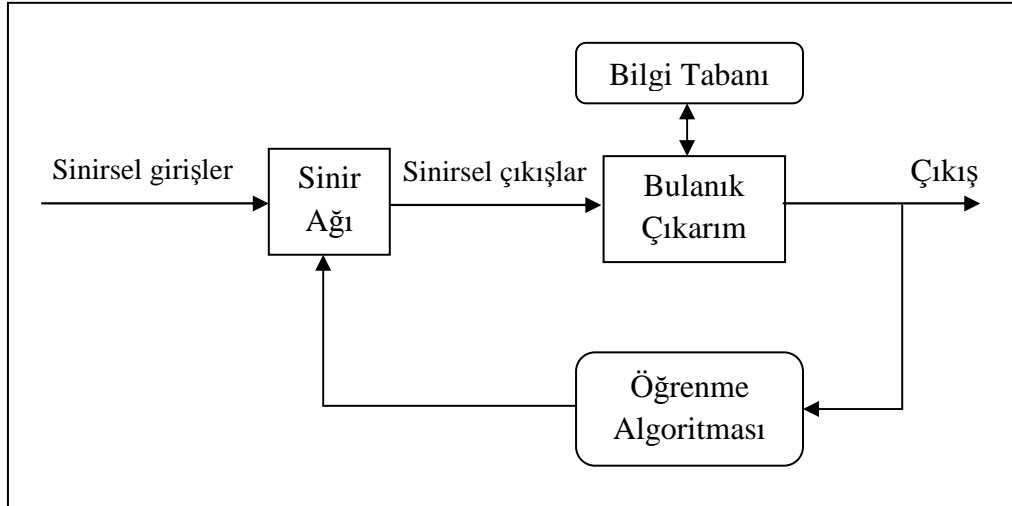


**Şekil 19. Sinirsel bulanık sistemlerin birinci yapısı**



**Kaynak:** Elmas (2003a:167)

**Şekil 20. Sinirsel bulanık sistemlerin ikinci yapısı**



**Kaynak:** Elmas (2003a:167)

Sinirsel Bulanık Sistem birleşimi yıllardır kontrol, veri analizi, karar destek gibi çeşitli amaçlarla kullanılmaktadır. Bulanık sınıflayıcıların veriden öğrenmesi, Sinirsel Bulanık Sistem yaklaşımı ile basit bir şekilde sağlanmaktadır.

Sinirsel Bulanık Sistemleri açıklamak ve diğer sistemlerden ayırmak için şu özellikleri sıralamak mümkündür (Uygunoğlu ve Yurtçu, 2008:64):

- Sinirsel Bulanık Sistem, bir bulanık sistemdir ve öğrenme algoritması için sinir ağı teorisi kullanılarak eğitilmektedir.
- Bir Sinirsel Bulanık Sistem, özel bir 3 katmanlı ileri beslemeli sinir ağı olarak

görülebilmektedir. Bu ağın üniteleri, sinir ağındaki aktivasyon potansiyeli yerine t-norms veya t-conorms kullanılmaktadır. İlk katman giriş değişkenlerini, orta katman bulanık kuralları ve üçüncü katman ise çıkış değişkenlerini içermektedir.

- Bir Sinirsel Bulanık Sistem genellikle bir bulanık kurallar sistemi gibi düşünülmektedir.
- Bir Sinirsel Bulanık Sistem, n boyutlu bir fonksiyon çıkarımı yapmaktadır. Bu da eğitim verisi ile verilmektedir. Bir Sinirsel Bulanık Sistem, bulanık uzman sistem gibi görülmemelidir. Sinirsel Bulanık Sistem, veriden bir bulanık sistem oluşturma tekniği veya örneklerden öğrenerek bunu geliştiren bir teknik olarak düşünülmektedir.

Sinirsel Bulanık Sistemler, Bulanık Mantık yaklaşımının veya Yapay Sinir Ağları'nın tek başına kullanımından daha avantajlıdır. Çünkü bu birleşim onlara hem Bulanık Mantık'ın hem de Yapay Sinir Ağları'nın avantajlarının da birleşimini sağlamıştır. Bu avantajlar aşağıda sıralanmıştır (Tunalı, 2007:7):

- **Öğrenme ve dilsel yetenek:** Yapay Sinir Ağları'nın eğitilmesi ile sağlanan öğrenme yeteneği bu birleşimlerde mevcuttur. Aynı zamanda bu birleşimlerde Bulanık Mantık'ın insanın düşünce yapısına uygun dilsel ifade yeteneği de mevcuttur.
- **Sayısal dilbilimsel veya mantıksal bilgi işleme:** Yapay Sinir Ağları'nın sayısal bilgi işleme ve Bulanık Mantık'ın mantıksal bilgi işleme yeteneğinin her ikisini de kullanabilir.
- **Belirsiz davranış veya belirli bilgi:** Yapay Sinir Ağları doğrusal olmayan problemlerin çözümünde başarılı olabilirken kesin sonuçlar vermesi bazen sınırlı kalabilmektedir. Bu kesin olmayan sonuçlar Bulanık Mantık ile ifade edilir.
- Bulanık sistemin esneklik, hız ve uyarlanırlılık gibi özellikleri sinir ağları ile artırılmıştır.

Bu avantajlar sayesinde Sinirsel Bulanık Sistemler; öğrenebilen, hızlı, performanslı, belirsizliğe karşı daha toleranslı, düşük maliyetli, uygulanabilirliği ve bilgi işleme yeteneği yüksek sistemler haline gelmektedir.

## BÖLÜM 3: AFET YÖNETİMİ VE AFETLERDEN ZARAR GÖREBİLİRLİK

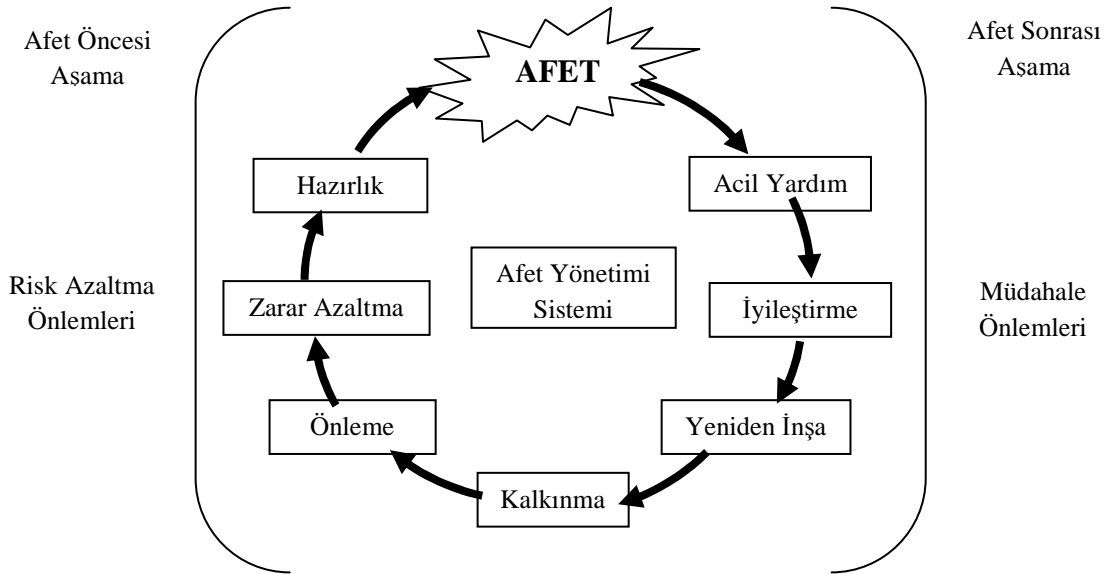
Bu bölümde, Afet Yönetimi ve Zarar Görebilirlik kavramlarının incelenmesi amaçlanmıştır. Bu amaçla bölümde Afet Yönetimi ve aşamaları, Zarar Görebilirlik kavramı, Zarar Görebilirlik çeşitleri, Sosyal Zarar Görebilirlik kavramı ve uygulamada yer alan Zarar Görebilirlik faktörleri anlatılmıştır.

### 3.1. Afet Yönetimi ve Aşamaları

Afet bir olay veya tehlikenin yol açtığı kayıplardır. Ortaya çıkabilecek bu olumsuz sonuçlar; fiziksel, sosyal veya ekonomik düzeyde olabilmektedir. Afet yönetiminde ise afetlerin önlenmesi ve sonrasında ortaya çıkabilecek zararların azaltılması amaçlanmaktadır. Bu nedenle afete yol açacak nedenlerin iyi bilinmesi ve sonrasında oluşabilecek zararların azaltılması yönünde çalışmalar yapılması gerekmektedir.

Afet yönetimi; afet öncesi, afet sırası ve afet sonrası olmak üzere farklı açılardan afet olayına yaklaşmaktadır. Afet sonrası yapılacak müdahaleler kadar, afet zararını azaltmada afet öncesi hazırlıkların önemi kavranmış ve bu konsept geliştirilmiştir (Akdağ, 2002:8). Afet Yönetimi Modeline ait aşamalar Şekil 21’de gösterilmektedir.

#### Şekil 21. Afet yönetimi modeli

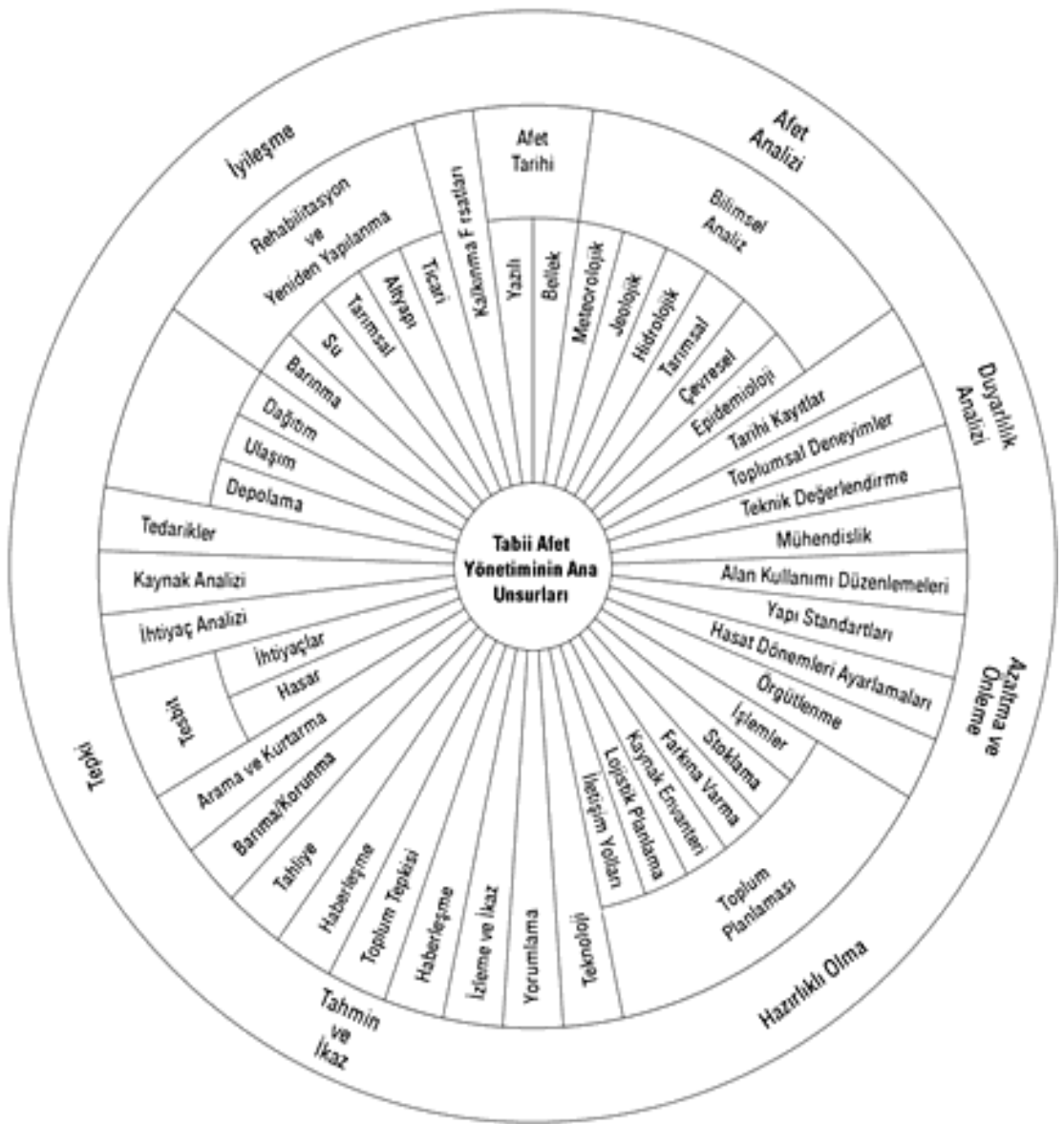


**Kaynak:** Ergünay (2008:98)

Afetin büyüklüğü genel olarak, olayın neden olduğu can kayıpları, yaralanmalar, yapısal hasarlar, sosyal, ekonomik ve çevresel kayıpların büyüklükleri ile değerlendirilmektedir. Kökenleri ve gelişim hızları ne olursa olsun, tüm afet olayları ile ilgili faaliyetler, zarar azaltma, hazırlık, olaya müdahale, iyileştirme olmak üzere dört ana aşamaya ayrılmaktadır (Ergünay, 2008:97).

Başka bir Afet Yönetimi Modeli ise Şekil 22’de Wisconsin Üniversitesi, Afet Yönetim Merkezi’nden uyarlanan dairesel diyagramda gösterilmektedir.

**Şekil 22. Wisconsin Üniversitesi afet yönetimi modeli**



**Kaynak:** www.dmc.metu.edu.tr

Afet öncesi aşamada yer alan risk azaltma önlemlerinde eğitim ve diğer sosyal faktörler önemli bir paya sahiptir. Bu konuda yapılan eğitim ve sosyal çalışmalarının kapsamı; afet bilincinin toplumun her kesimine yaygınlaştırılması, zarar azaltma, hazırlıklı olma, müdahale ve iyileştirme aşamaları için gerekli eğitimin verilmesi, becerilerin geliştirilmesiyle toplumun afetlerle mücadele kapasitesinin artırılması ve sivil toplumun örgütlenmelerinin sağlanması hususları olmalıdır (Uzunçubuk, 2005:360).

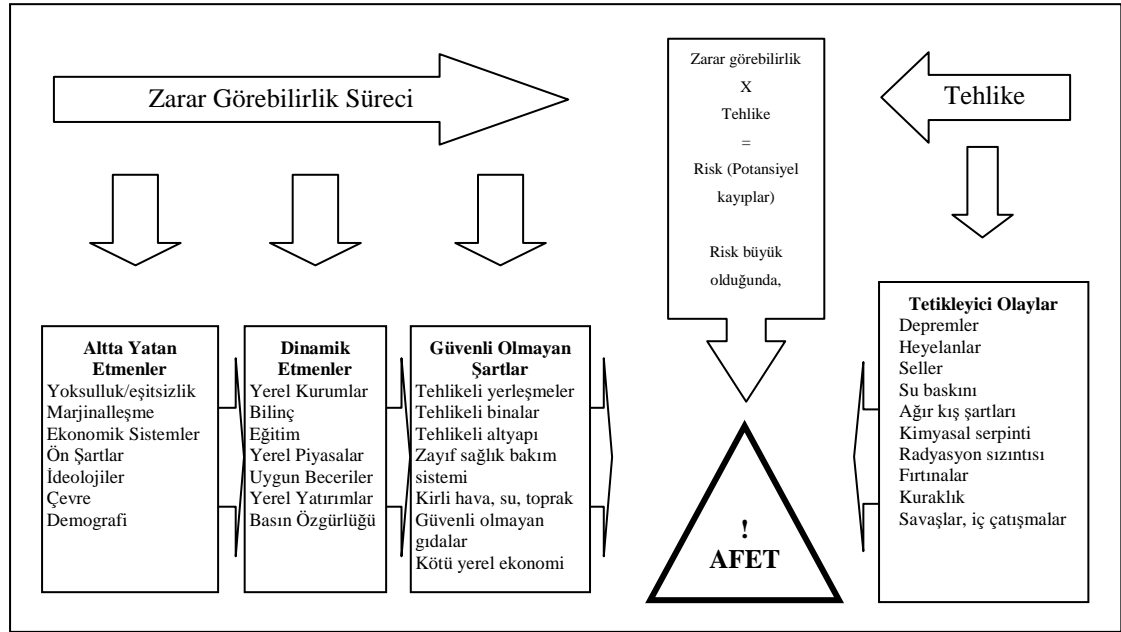
Afet sırası ve sonrası aşamada ise; mümkün olan en fazla sayıdaki insanı kurtarmak ve sağlıklarına kavuşmalarını sağlamak, afetlerin doğurabileceği ek tehlike ve risklerinden insan canını ve malını korumak, afetten etkilenen toplulukların hayati ihtiyaçlarını mümkün olan en kısa zamanda karşılamak ve hayatın bir an önce normal hale gelmesini sağlamak, afetin yol açtığı fiziksel, ekonomik, sosyal, psikolojik ve çevresel kayıpların en düşük düzeyde kalmasını veya yaraların bir an önce sarılmasını sağlamak, afetten etkilenen topluluklar için riskleri azaltılmış, daha güvenli ve gelişmiş yeni bir yaşam çevresi oluşturmak gibi temel esasları dikkate alınmalıdır (Ergünay, 2009a:5).

### **3.2. Zarar Görebilirlik**

Bir toplumun, bir yapının veya bir hizmetin, afet tehlikesi oluştuğunda görebileceği hasar veya zararın olası ölçüsünü etkileyen faktörler zarar görebilirlik olarak tanımlanmaktadır. Aynı zamanda; herhangi bir tehlikenin etkilerini tahmin etme, vereceği zararları azaltma, sonuçlarına direnç gösterme ve başa çıkma, normal yaşama dönebilme konularında, bir insanın ya da sosyal grubun imkan ve kapasitelere sahip olma düzeyini ifade etmektedir.

Halkın ve karar vericilerin riski algılamaları ve önleyici ve zarar azaltıcı politika ve eylemleri uygulamaya başlaması kolay değildir. Kaynakların ve kapasitelerin sınırlı, günlük risklerinse çok büyük olduğu durumlarda, gelecekteki bir riski azaltmak amacıyla zaman ve kaynak ayırmak oldukça zordur. Bu durumda bilim adamları ve araştırmacılar, meslek odaları ve yardım kuruluşlarına çok büyük görevler düşmektedir. Bu kuruluşlar, özellikle gelişmekte olan ülkelerde yoğun günlük risklerle iç içe yaşayan insanlara, gelecekte karşılaşılabilecekleri riskleri açıklamak, risklerin büyük bir kısmının önlenebilir veya zararlarının azaltılabilir olduğu konusunda onları bilgilendirmek ve bilinçlendirmek zorundadırlar (Ergünay, 2002:18). Şekil 23'te tehlike, zarar görebilirlik, etki eden faktörler, risk ve afet arasındaki ilişki gösterilmektedir.

**Şekil 23. Tehlike, zarar görebilirlik, risk ve afet arasındaki ilişki**



**Kaynak:** Ergünay (2002:19)

Zarar görebilirliği, fiziksel, sosyal ve ekonomik zarar görebilirlik olarak üçe ayırarak incelemek mümkündür.

### 3.2.1. Fiziksel Zarar Görebilirlik

Fiziksel zararlar; insan eliyle oluşturulmuş yapılar, altyapılar, çevre, tarım, sanayi, üretim vb. gibi unsurlar ile toplumların fiziksel kapasiteleriyle ilgili olası afet etkilerini içermektedir (Özkul ve Karaman, 2007:257). Ölçülebilme veya sayısal hale getirilmesi mümkündür (Ergünay, 2002:12).

### 3.2.2. Ekonomik Zarar Görebilirlik

Ekonomik zararlar, ekonomik yaşam biçimi ve ilişkilerin nasıl düzenlendiği, yürütüldüğü, üretim ve geçim sağlamayla ilgili imkan ve kapasitelerin olası deprem felaketinden etkilenmelerini kapsamaktadır (Özkul ve Karaman, 2007:257). Geçmişte yaşanan afetlerde, fakirlikle fiziksel zarar görebilirlik arasında doğrudan bir ilişki olduğu görülmüş ve toplulukların gelir düzeyi düşük kesimlerinin ekonomik nedenlerle doğal ve teknolojik tehlikelere daha çok maruz olan dere yatakları, heyelana müsait yamaçlar, depremlere daha dayanaksız konutlarda yaşadıkları ve bu nedenle de afetlerden daha çok etkilendikleri sonucuna varılmıştır (Ergünay, 2002:12).

### 3.2.3. Sosyal Zarar Görebilirlik

Sosyal zarar görebilirlik ise toplumun nüfusu, nüfus yoğunluğu, yaş ve cinsiyet oranları, eğitim ve kültür düzeyleri ile diğer ilgili sosyal faktörleriyle, olası afet etkilerini içermektedir (Güler, 2008:16). Bu faktörlerin ölçülebilmesi ve sayılaştırılması güç olmaktadır. Ancak fiziksel zarar görebilirlik düzeyinin ve başa çıkma kapasitelerinin sosyal gruplar arasında farklılık göstermesi ve geçmişte yaşanan afet olaylarının analizinde, yaşlıların, çocukların, özürlerin olaylardan daha çok etkilendiklerinin ve başa çıkma kapasitelerinin daha az olduğunun görülmesi, bu tür bir zarar görebilirlik tanımına ihtiyaç göstermiştir.

Bir tehlikenin afet sonucunu doğurması, tehlikenin büyüklüğünden veya oluş sıklığından çok, zarar görebilirliğin büyüklüğüne bağlıdır. Bu nedenle olay öncesinde önlenmesi mümkün olmayan doğal ve teknolojik afetler için zarar azaltma strateji planları, sektörel planlar ve uygulama programları, genellikle zarar görebilirliğin azaltılmasına yöneliktir. Zarar görebilirliğin azaltılabilmesi içinde, insan yerleşmelerini zarara açık ve kolaylıkla etkilenebilir hale getiren ana faktörlerin neler olduğunun iyi tanımlanmış olması gerekmektedir (Ergünay, 2009b:14).

İnsan toplulukları ve yerleşmelerini zarar görebilir hale getiren faktörlerden bazılarını (Ergünay, 2009b:15);

- Yoksulluk, az gelişmişlik, işsizlik,
- Hızlı nüfus artışı,
- Hızlı ve denetimsiz kentleşme, yapılaşma ve sanayileşme,
- Bilgisizlik, bilinçsizlik ve eğitim eksikliği,
- Yaşam tarzında meydana gelen büyük değişimler,

şeklinde tasnif etmek mümkündür.

Faktörleri daha detaylı olarak incelediğimizde; yapılan çalışmalarda yoksulluk, işsizlik ve gelir dağılımındaki dengesizliklerin zarar görebilirliği önemli ölçüde arttırdığı, bu tür toplumların afetlerden daha çok etkilendiği ve bir afetle karşılaştıklarında baş edebilme kapasitelerinin çok sınırlı olduğu görülmektedir.

Hızlı nüfus artışları diğer parametreler aynı da kalsa afetin büyüklüğünü arttıracaktır. Bunun yanı sıra hızlı nüfus artışları birçok insanın tehlikeye maruz alanlarda yerleşmesine, kaçak yapılaşmalara ve sınırlı kaynakların daha çok kişi tarafından yetersiz şekilde kullanılmasına da yol açmaktadır. Hızlı nüfus artışıyla ortaya çıkan hızlı, plansız ve denetimsiz kentleşme ve sanayileşme de zarar görebilirliği ve dolayısıyla afetin boyutlarını artırmaktadır (Ergünay, 2009b:15).

Zarar görebilirliğin ve afetin büyük boyutlara ulaşmasının temel nedenlerinden birisi de toplumun her kesimindeki bilgi, bilinç ve eğitim eksikliğidir. Zarar azaltma çalışmalarının büyük bir yoğunluk ve ağırlıkla, her düzeydeki halkı ve her düzeydeki yetkilileri bilgilendirme ve bilinçlendirmeyi amaçlayan eğitim faaliyetlerine yönlendirilmesi zorunlu olmaktadır (Ergünay, 2009b:15). Afet zararlarının azaltılması, her şeyden önce, birey ve toplumun afet zararlarının azaltılması konusunda etkili ve sürekli eğitim faaliyetleri ile bilgilendirilmesi ve bilinçlendirilmesinden geçtiğini hiç unutmamak gerekmektedir. Afet zararları yalnızca uzmanlar, bilim adamları ve araştırmacılar, kamu kurum ve kuruluşlarının bu konudaki çaba ve faaliyetleri ile azaltılamaz. Toplumun tümünün bu faaliyetlere katılması ve toplumda bir “zarar görebilirliği azaltma” kültürü oluşturulması, en önce gelen bir faaliyet olmalıdır (Ergünay, 2002:16).

Bunun yanı sıra dört yıllık lisans eğitimi ile inşaat mühendisi ve mimar yetiştiren yaklaşık 37 üniversitenin büyük bir çoğunluğunda “Depreme Dayanıklı Yapı Tasarımı” dersi zorunlu olarak okutulmamaktadır. Bazı üniversitelerimizde bu konudaki dersler sadece seçmelidir. Dolayısı ile mimar ve mühendislerimiz “Depreme Dayanıklı Yapı Tasarımı” konusunda yeterli bilgi ve beceri sahibi olmadan mezun olmaktadır. Bu eksikliklerin, meslek içi kurslarla telafi edebilmek için yoğun çaba sarf edilmelidir. Aynı zamanda “Depreme Dayanıklı Yapı Tasarımı” dersinin içeriği ülkemizdeki tüm inşaat mühendisliği ve mimarlık bölümlerinden bilim adamlarının ortak görüşleri doğrultusunda belirlenmeli, bundan önceki depremlerde oluşan hasarların sebepleri irdelenmeli, yanlışlar ve hatalar öğrencilere aktarılmalıdır (Özkul ve Karaman, 2007:258).

Sosyal faktörler ve risk dereceleri arasında doğrudan ve dolaylı ilişkiler söz konusudur. Ancak risk dereceleri belirlenirken bu faktörler göz ardı edilmektedir. Risk dereceleri



jeolojik, jeofizik, jeomorfolojik kriterlere göre hazırlanmaktadır. Daha güvenilir sonuçlar için sosyal faktörler de risk dereceleri belirlenirken kullanılmalıdır.

Sosyal faktörlerin bu tür modellerde yer alamamasının nedenlerinin başında, faktörlere bağlı ortaya çıkabilecek hasarların matematiksel modellerle ölçümünün zor olması gelmektedir. Günümüzde ise Bulanık Mantık yaklaşımının belirsizlikleri çözebilmesi ve Yapay Sinir Ağları'nın da insan beynini taklit eder bir yapıda olması bu ağırlıklandırmaların ve ölçümlerin yapılabilmesine işaret etmektedir.

Çalışmanın uygulama bölümünde bu bağlamda, Yapay Sinir Ağları'ndan yararlanılarak sosyal faktörlerin risk dereceleriyle ilişkilendirilebildiği gösterilmiştir. Bu süreçte sosyal faktörler belirlenmiş, 81 ile ait veriler derlenerek "SPSS Clementine" programında SOM (Kendini Düzenleyen Haritalar) Yapay Sinir Ağları kullanılarak sınıflandırılma yapılmıştır. Daha sonra MATLAB programının Neural Networks Toolbox modülünde sosyal faktörler ve sınıflandırma sonuçlarıyla belirlenen risk derecelerinin ilişkilendirilmesine yönelik test sonuçları alınmıştır.

## **BÖLÜM 4: YSA KULLANILARAK TÜRKİYE’DEKİ İLLERİN ZARAR GÖREBİLİRLİĞİNİN HESAPLANMASINA İLİŞKİN YÖNTEM VE UYGULAMA**

### **4.1. Araştırmanın Yöntemi**

Bu bölümde sosyal zarar görebilirliğe etki eden faktörler ve bu faktörlere bağlı sosyal sınıflar Yapay Sinir Ağları ile ilişkilendirilerek iller bazında incelenmiştir. Eğitim ve analizlerde kullanılan veri seti Türkiye’de bulunan 81 ille ait verilerden derlenerek hazırlanmıştır. Analizlerin sonuçlarına değinilmeden önce çalışmada kullanılan SPSS Clementine ve MATLAB programının genel yapısı ve işleyişi anlatılmış, son olarak uygulama süreci ve sonuçları açıklanarak yorumlanmıştır.

Uygulamada, Yapay Sinir Ağları kullanılarak Afet Yönetimi’nde sosyal zarar görebilirliğe etki eden faktörlerin, kümeleme analizi ile oluşturulan sosyal sınıflarla ilişkilendirilebildiği gösterilmektedir. Afet Yönetimi’nin daha verimli hale gelebilmesi için tüm faktörlerin etkin bir şekilde değerlendirilmesi gerekmektedir. Literatürde ise deprem risk derecelerinin yalnızca fiziksel faktörlere göre hesaplandığı görülmektedir. Ölçülmesi fiziksel faktörlere göre daha zor olan sosyal ve ekonomik faktörlerin Yapay Sinir Ağları ile ölçülebileceği ortaya konularak, illere ait deprem risk derecelerinin hesaplanmasında daha verimli sonuçların alınabilmesi için sonraki çalışmalarda diğer faktörlerin de göz önünde bulundurulması gerektiğine değinilmiştir. Çalışma bu açıdan keşifsel araştırma özelliği göstermektedir.

Uygulamada kullanılan veri seti Türkiye’de bulunan 81 ilin sosyal zarar görebilirliğe etki eden faktörlerden oluşmaktadır. Deprem risk derecelerinin analizinde kullanılacak veriler Türkiye İstatistik Kurumu’ndan (TÜİK) alınarak illere göre kategorilere ayrılmıştır. Bu kategoriler; “Eğitim Durumu”, “Konut Sayısı”, “Doğurganlık Hızı”, “Kişi Başına Gelir”, “Net Göç”, “İşsizlik Oranı” ve “Toplam Nüfus” olmak üzere 7 başlık altında toplanmaktadır. “Eğitim Durumu” faktörü ise “Okuma Yazma Bilmeyen”, “İlköğretim ve Ortaöğretim Mezunu” ve “Yükseköğretim Mezunu” olarak ayrılarak veriler toplamda 9 faktöre ayrılmaktadır.

Analizde kullanılan anakütlenin sonlu yapıda olması ve analizlerde daha verimli sonuçlar elde etmek için tamsayım yapılmıştır. Veriler matris tablolar halinde

düzenlenirken Microsoft Office Excel programından yararlanılmıştır. Daha sonra ise kümeleme analizinde SPSS Clementine 11.1 programı kullanılmıştır. Yapay Sinir Ağının oluşturulmasında, eğitim ve test aşamalarında ise MATLAB programı kullanılmıştır.

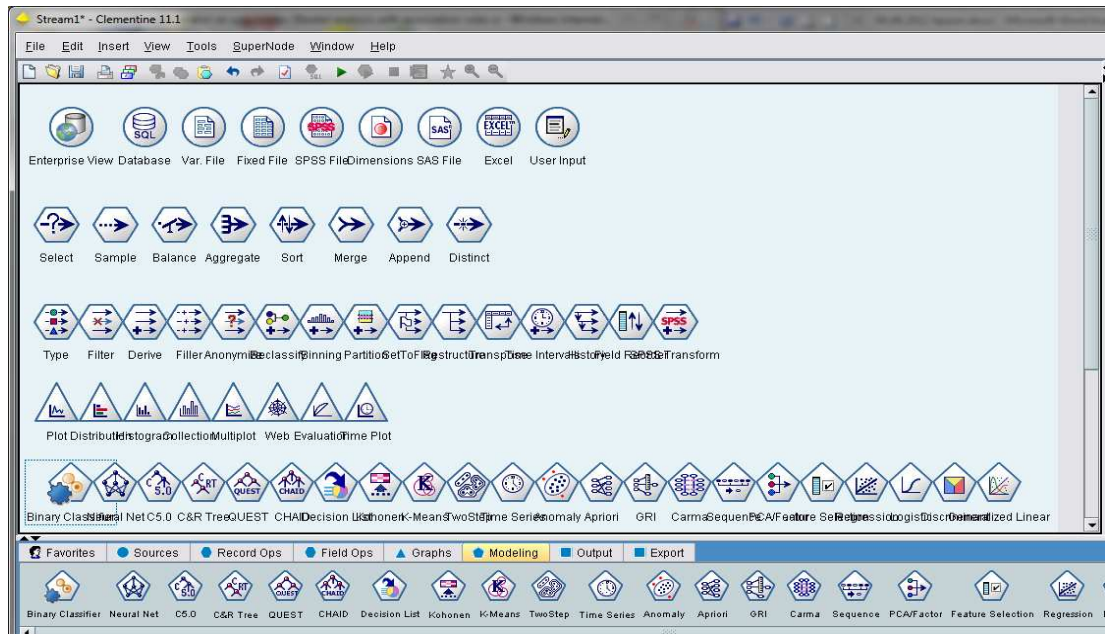
#### 4.1.1. Uygulamada Kullanılan Programlar

Yapay Sinir Ağları tekniği için çalışma kapsamında incelenebilecek, piyasada bulunabilecek ya da çeşitli araştırma çalışmaları bünyesinde geliştirilmiş olan, farklı özellik ve işlevlere sahip, birçok uygulama yazılımı bulunmaktadır. Bu yazılımlar “Neuro Solutions”, “Statistica-Neural Networks”, “Mathematica-Neural Networks”, Netmaker, “JOONE”, “Fuzzy Cope”, “SPSS Clementine” ve “MATLAB” olarak isimlendirilmektedir (Köse, 2010:35). Bu çalışmada üniversite bünyesinde bulunan “SPSS Clementine” ve “MATLAB” programına yer verilmektedir.

Uygulamada kullanılan SPSS Clementine 11.1 programı İngiliz menşeli bir firma olan ISL (Integral Solutions Limited) tarafından geliştirilmiş ve piyasaya sunulmuştur. Program veri girişi yapılmasına, farklı veri setlerini birleştirmeye, modellemeye ve analiz etmeye olanak sağlamaktadır.

Şekil 24’te SPSS Clementine 11.1 programına ait arayüz görülmektedir.

#### Şekil 24. SPSS Clementine programı arayüzü



SPSS Clementine uygulamacıya, kolay kullanabileceği bir arayüz ve kısa sürede sonuca ulaşabileceği görsel bir çalışma ortamı sunmaktadır. Araştırmacının kullanmak isteyeceği tüm fonksiyonlar arayüzün alt tarafında bulunan bir araç çubuğunda ikonlar halinde yer almaktadır. Yapılması gereken, gerekli ikonların analiz sayfasına taşınarak birbirlerine bağlanması ve çalıştırılmasıdır (Gürgen, 2008: 61).

Yapay Sinir Ağının oluşturulması ve eğitilmesi aşamasında ise MATLAB programı kullanılmaktadır. MATLAB, MATrix LABoratory kelimelerinden türetilmiş, daha çok matematiksel işlemler yaptırmaya yönelik tasarlanmış bir bilgisayar programıdır.

MATLAB ile birlikte gelen Neural Networks Toolbox uygulama yazılımı, MATLAB yazılımının da etkisi ve desteği ile birlikte, Yapay Sinir Ağları üzerine oldukça geniş bir uygulama alanını destekleyen, teknik anlamda etkili kullanım özellikleri ve işlevleri sunan, popüler bir platform olarak kullanıcılara sunulmaktadır. Yazılım bünyesinde yer alan görsel arayüz sayesinde, Yapay Sinir Ağları sistemleri hızlı bir şekilde tasarlanıp, uygulamaya dönüştürülebilmekte, yine görsel işlevlerle birlikte, gerçekleştirilen çalışmalarla alakalı veriler etkili bir şekilde gözlemlenebilmektedir ([www.mathworks.com](http://www.mathworks.com)).

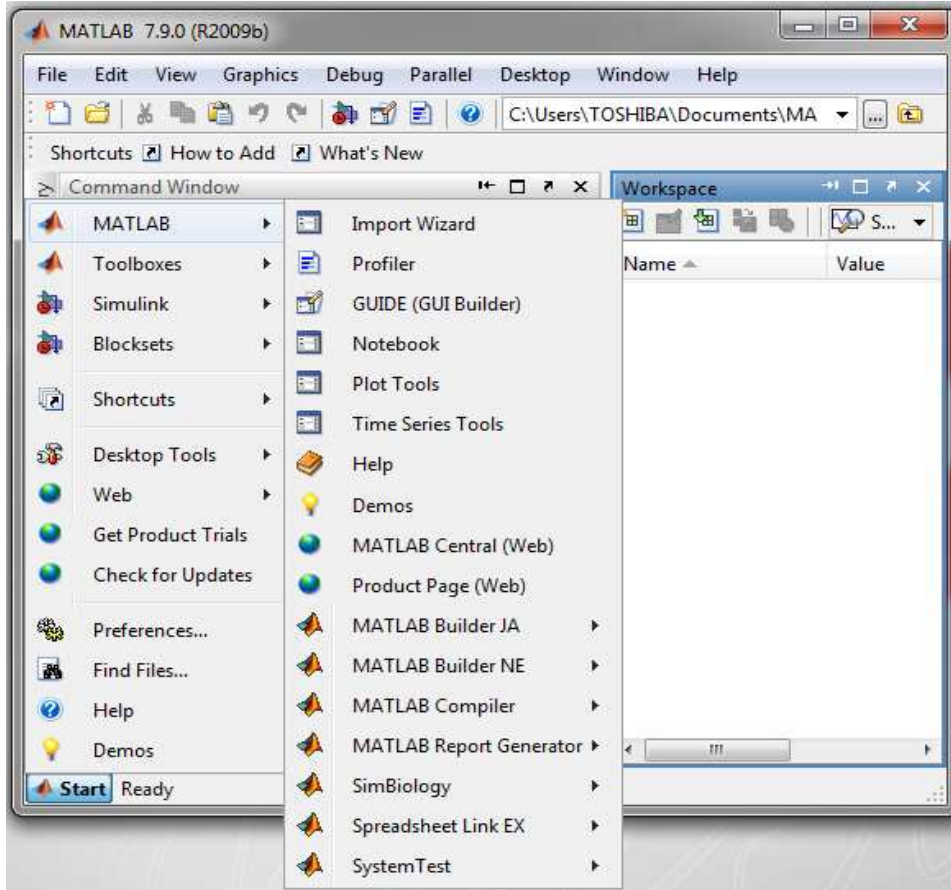
Neural Networks Toolbox yazılımı yaygın bir şekilde kullanılan, çok katmanlı ağ ve Kohonen (kendi kendine organize) ağları gibi yaygın kullanım alanı bulan mimariler ile birlikte birçok farklı Yapay Sinir Ağları mimarisini ve ilgili öğrenme kurallarını desteklemektedir ([www.mathworks.com](http://www.mathworks.com)).

Yazılım bünyesinde sunulan kontroller yardımıyla, üzerinde çalışılan sistemin kaynakları elverdiği ölçüde Yapay Sinir Ağı yapıları tasarlanıp, düzenlenebilmektedir. Geliştirilen bir uygulama ile ilgili performans ve eğitim süreçleri, etkili görsel işlevlerle kullanıcılara sunulmaktadır.

Yazılım ile birlikte gelen görsel grafik seçenekleri, kullanılan verilerin ve elde edilen sonuçların kolay bir şekilde gözlemlenmesine imkan sağlamaktadır. Yazılımın en önemli avantajı ise, Bulanık Mantık'ta olduğu gibi, Yapay Sinir Ağları için de Simulink desteğinin var olmasıdır. Bu şekilde çok çeşitli kontrol problemleri ve uygulamaları, Yapay Sinir Ağları tabanında, kolay ve hızlı bir şekilde geliştirilebilmektedir. Simulink ortamında kullanılan etkileşimli, görsel "blok" yapıları, ilgili kullanıcılara basit bir

kullanım tecrübesi vaat etmektedir. Bu avantajların yanında, ilgili yazılım için değinilebilecek tek dezavantaj, sunduğu teknik anlamdaki bazı özellik ve işlevlerin, daha çok ileri düzeyde kullanıcıları hedef alması sayılabilmektedir (Köse, 2010:35). Şekil 25'te MATLAB programının genel görünümü gösterilmektedir.

**Şekil 25. MATLAB genel yapı**



MATLAB programında Yapay Sinir Ağları'nın eğitilmesi için ağırlıklarının ve değerlerinin kurulması gerekmektedir. Kurulum gerçekleşikten sonra ağ eğitime hazır hale gelmektedir. Eğitim verilirken deneme yanılma yoluyla en iyi sonuca ulaşılmaktadır. İterasyon sayısı, gizli katman sayısı, eğitim setinin verileri ve diğer tüm elemanlar deneme yanılma yönteminin bir parçasını oluşturmaktadır.

Programa genel olarak bakıldığında birçok öğrenme algoritması görülmektedir. Her birinin hafıza ihtiyacı ve hesap yükü farklılıklar göstermektedir. Tablo 3'te kullanılan bazı veri işleme fonksiyonları ve Tablo 4'te ise MATLAB programında yer alan öğrenme algoritmaları gösterilmektedir.

**Tablo 3. MATLAB veri işleme fonksiyonları**

<b>premnmx</b>	Veriyi $\pm 1$ arasına çeker.
<b>postmnmx</b>	“premnmx” komutunun tersidir.
<b>trammx</b>	Önceden hesaplanan min ve max değerlerini kullanarak ağa yeni uygulanacak veriyi normalize eder.
<b>prestd</b>	Veriyi normal dağılıma uyacak şekilde normalize eder.
<b>poststd</b>	“prestd” komutunun tersidir.
<b>trastd</b>	Önceden hesaplanan standart sapma ve ortalama değere göre ağa yeni uygulanacak veriyi normalize eder.
<b>prepca</b>	Temel bileşen analizini gerçekleştirir. Giriş eleman sayısını azaltır.
<b>trapca</b>	Önceden hesaplanmış olan temel bileşen analiz matrisini kullanarak ağa yeni uygulanacak veriyi uygun forma dönüştürür.
<b>postreg</b>	Ağın çıkışları ile hedef çıkışları arasında lineer bir regresyon analizi yapar. Ağın genellemesinin uygunluğunun analizi için kullanılır.

**Kaynak:** Sağıroğlu ve diğ. (2003: 369)

**Tablo 4. MATLAB öğrenme algoritmaları**

<b>traingd</b>	Basit gradiyent azaltımına dayanır. Yavaş cevap verir. Arttırmalı öğrenmede kullanılabilir.
<b>traingdm</b>	Momentum kullanan gradiyent düşürmeli öğrenme yöntemidir. Genelde traingd’den hızlıdır. Arttırmalı öğrenmede de kullanılabilir.
<b>trainrp</b>	Esnek backpropagation yapısıdır. Basit grup öğrenme algoritmasıdır. Hızla sonuca yakınsar ve hafıza ihtiyacı azdır.
<b>traingcf</b>	Fletcher-Reeves eşleştirmeli gradiyent algoritmasıdır. Hafıza ihtiyacı azdır.
<b>traingcp</b>	Polak-Ribiere eşleştirmeli gradiyent algoritmasıdır. Hafıza ihtiyacı çok fazladır. Bazı problemler için yakınsama hızı etkileyicidir.
<b>traingqb</b>	Powell-Beale eşleştirmeli gradiyent algoritmasıdır. Hafıza ihtiyacı oldukça fazladır. Sonuca hızla yakınsar.
<b>traingcg</b>	Ölçeklendirilmiş eşleştirmeli gradiyent algoritmasıdır. Eşlemeli gradiyent algoritmaları arasında doğru aramaya gecek duymayan tek algoritmadır.
<b>trainbfg</b>	Quasi-Newton öğrenme algoritmasıdır. Hafıza ihtiyacı fazladır. Fakat sonuca çok hızlı yakınsar.
<b>trainoss</b>	Tek adımlı sekant metodudur. Eşleştirmeli gradiyent metotları ile Quasi-Newton metodunun birleşimi ile oluşturulmuştur.
<b>trainlm</b>	Levenberg-Marquardt algoritmasıdır. Öğrenme kümesi çok büyük ise hafıza ihtiyacını azaltan bir parametreye de sahiptir. Modern öğrenme algoritmalarını en hızlısıdır.
<b>trainbr</b>	Bayesian düzenleyici öğrenme metodudur. Levenberg-Marguart algoritmasının, genelleme yeteneğini iyileştirmek için geliştirilmiş bir halidir. Optimum ağ boyutunun ne olması gerektiği problemini kısmen azaltmaktadır.

**Kaynak:** Sağıroğlu ve diğ. (2003: 368)

## 4.2. Uygulama

### 4.2.1. SOM (Kendini Düzenleyen Haritalar) Kullanılarak Sosyal Zarar Görebilirlik Sınıflarının Oluşturulması

Kümeleme modellerinde amaç küme üyelerinin birbirlerine çok benzediği, ancak özellikleri birbirlerinden çok farklı olan kümelerin elde edilmesi ve verilerin bu farklı kümelere bölünmesidir. SOM veya diğer adıyla Kohonen Ağlar 1980’lerde geliştirilmiştir. Kohonen ağları eğitilirken, kullanılan öğrenme algoritması denetimsizdir. Yani, ağ eğitilirken bağımlı değişken yani çıktılar kullanılmaz. Ağ yalnızca girdi modelini öğrenir. Katmanlar arasındaki ağırlıkların ayarlanması ağ tarafından kendiliğinden gerçekleştirilir. Denetimsiz öğrenmede sadece girdi katmanındaki değerler kullanılmaktadır.

Bu tekniğin diğer kümeleme algoritmalarına göre bir avantajı, çok boyutlu verileri iki boyutlu bir düzlem kullanarak görselleştirmesidir. Kendini düzenleyen haritalar tekniğinde, her girdi düğümü bir boyuta karşılık gelirken, her çıktı düğümü de iki boyutlu düzlemde bir düğüme karşılık gelir (Taşkın ve Emel, 2010:401). Tablo 5’te bazı kümeleme yaklaşımlarının bir karşılaştırması verilmektedir.

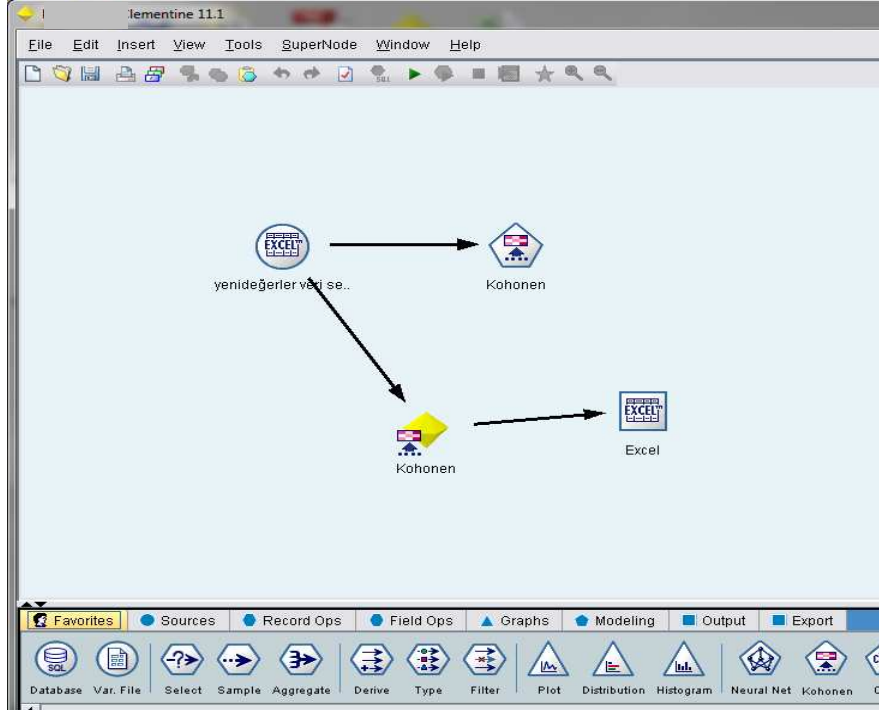
**Tablo 5. Bazı kümeleme yaklaşımlarının karşılaştırılması**

Kümeleme Yaklaşımları	Avantajlar	Dezavantajlar
<b>Yapay Sinir Ağları (Kohonen)</b>	<ul style="list-style-type: none"><li>•Doğal başlangıç</li><li>•Büyük hacimli veriler üzerinde çalışabilme</li><li>•Küme sayısını belirleyebilmesi</li></ul>	<ul style="list-style-type: none"><li>•Uzun hesaplama zamanları</li><li>•Eğitim parametrelerinin ayarlanmasının zor olması ve farklı parametrelerin farklı sonuçlar vermesi</li></ul>
<b>Hiyerarşik Yöntemler (Ward’ın Minimum Varyans Yöntemi)</b>	<ul style="list-style-type: none"><li>•Küme sayısını belirleyebilmesi</li></ul>	<ul style="list-style-type: none"><li>•Büyük hacimli veriler üzerinde çalışma zorluğu</li><li>•Aykırı değerlerden (outlier) kolayca etkilenmesi</li><li>•Geri alınamama</li></ul>
<b>Hiyerarşik Olmayan Yöntemler (K-means Yöntemi)</b>	<ul style="list-style-type: none"><li>•Başlangıç noktası ve küme sayısı verildiğinde daha yüksek doğruluk</li><li>•Büyük hacimli veriler üzerinde çalışabilme</li></ul>	<ul style="list-style-type: none"><li>•Küme sayısını belirleyememe</li><li>•Başlangıç noktasının ve küme sayısının rastgele olarak seçilmesi</li><li>•Aynı gruba ait iki merkezin seçilebilmesi</li></ul>

**Kaynak:** Taşkın ve Emel (2010:402)

SPSS Clementine 11.1 programına aktarılan 81 ile ait sosyal faktörlerden oluşan girdi verileri Kohonen Ağları kullanılarak sınıflandırılmıştır. Böylelikle illere göre sosyal faktör sınıfları oluşturulmuştur. Şekil 26’da kümeleme analizinde kullanılan verilerin programa aktarılması gösterilmektedir.

**Şekil 26. Verilerin SPSS Clementine 11.1 programına aktarılması**



Veriler programa Excel’de düzenlenen 81x9 boyutunda matris olarak aktarılmıştır. Veriler düzenlenirken kümeleme analizinde sonuçların elde edilebilmesi açısından normalizasyon yapılmıştır. Normalizasyon işlemi için her bir sütundaki minimum ve maksimum değerler bulunmuştur. Daha sonra her bir veri kendi sütunundaki minimum ve maksimum değerler yardımı ile hesaplanarak 0-1 aralığında değerler olarak atanmakta ve tüm veriler bu şekilde normalize edilmektedir.

Bu yöntem için kullanılan formül ise aşağıdaki gibidir:

$$X_n = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Tablo 6’da “Nüfus” faktörünün verilerine ilişkin normalizasyon aşamasına örnek gösterilmektedir. Normalizasyon sonucu elde edilen tüm veriler ise Ek’te verilmiştir.



**Tablo 6. Normalizasyon aşaması**

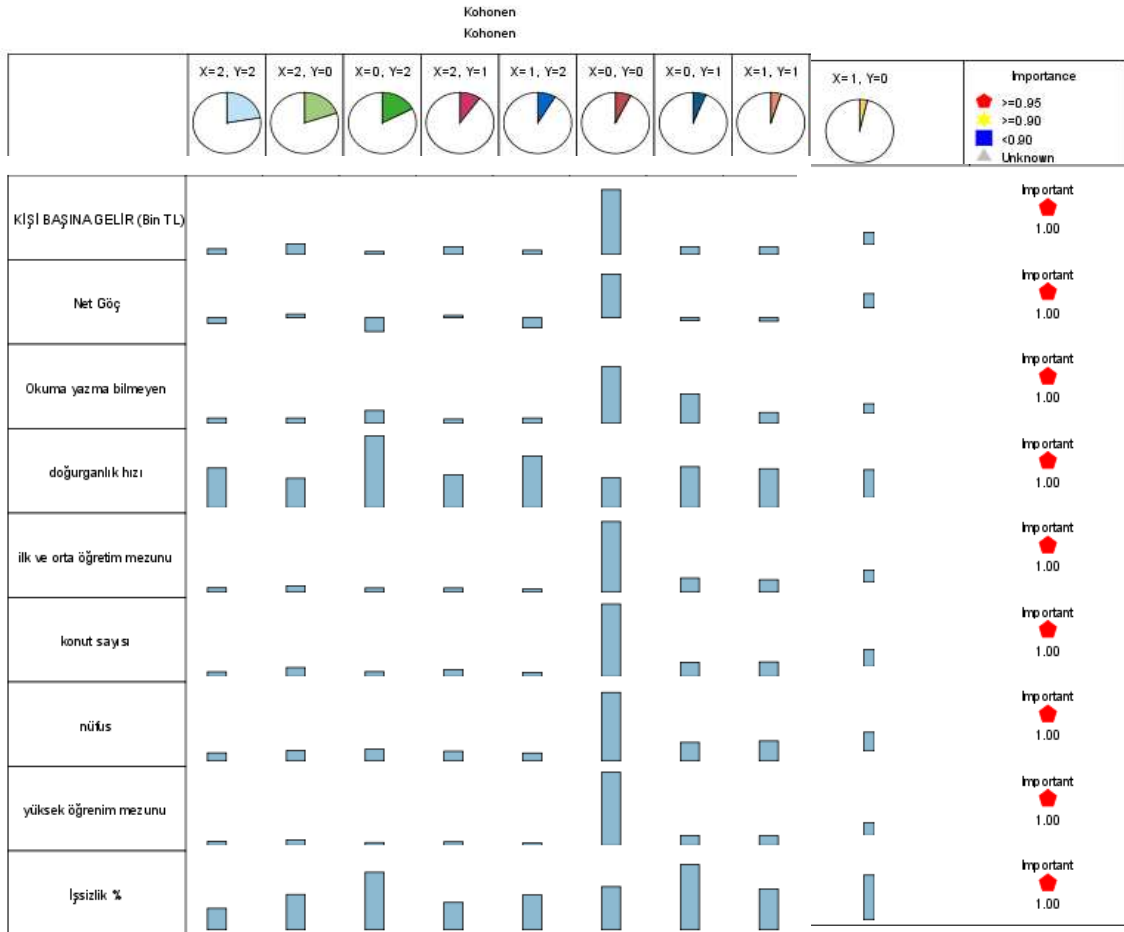
İller	Nüfus	Hesaplama	Normalizasyon Sonucu
Ardahan	133756	$(133756-93584) / (10018735-93584)$	0,0040
Yalova	168593	$(168593-93584) / (10018735-93584)$	0,0076
Karabük	225102	$(225102-93584) / (10018735-93584)$	0,0133
Kayseri	1060432	$(1060432-93584) / (10018735-93584)$	0,0974
Konya	2192166	$(2192166-93584) / (10018735-93584)$	0,2114
Malatya	853658	$(853658-93584) / (10018735-93584)$	0,0766
Tunceli (MİN)	93584	$(93584-93584) / (10018735-93584)$	0,0000
Adıyaman	623811	$(623811-93584) / (10018735-93584)$	0,0534
Ağrı	528744	$(528744-93584) / (10018735-93584)$	0,0438
Bitlis	388678	$(388678-93584) / (10018735-93584)$	0,0297
Antalya	1719751	$(1719751-93584) / (10018735-93584)$	0,1638
Aydın	950757	$(950757-93584) / (10018735-93584)$	0,0864
Bursa	2125140	$(2125140-93584) / (10018735-93584)$	0,2047
Edirne	402606	$(402606-93584) / (10018735-93584)$	0,0311
İçel	1651400	$(1651400-93584) / (10018735-93584)$	0,1570
İstanbul (MAX)	10018735	$(10018735-93584) / (10018735-93584)$	1,0000
İzmir	3370866	$(3370866-93584) / (10018735-93584)$	0,3302
Kocaeli	1206085	$(1206085-93584) / (10018735-93584)$	0,1121

Normalizasyonu tamamlanan veriler SPSS Clementine programında kümeleme analizinde kullanılmaktadır. Kohonen Ağlar kullanılarak elde edilen analiz sonuçları ise yine Excel'e çıktı olarak aktarılmaktadır. Aktarılan veriler ve çıktı tablosu arasında gerekli bağlantılar yapıldıktan sonra ağ çalıştırıldığında Şekil 27'de gösterilen grafiğe ulaşılmaktadır. Bu grafik yardımıyla sınıfların koordinatları belirlenmekte ve sonrasında Excel'de yer alan çıktılar doğrultusunda sınıflara ait iller ve verileri belirtilebilmektedir. Kohonen Ağlar kullanılarak yapılan analiz sonucunda 4 farklı sınıf elde edilmiştir. Bu sınıflar ve "Kohonen X", "Kohonen Y" koordinatları Tablo 7'de gösterilmektedir. Şekil 27'de ise Kohonen Ağları ile SPSS Clementine programında yapılan kümeleme analizinin çıktısı gösterilmektedir.

**Tablo 7. Sosyal faktörlerden elde edilen sınıflar**

Sosyal Sınıf	İl Sayısı	Kohonen X	Kohonen Y
1	42	2	0
		2	1
		2	2
2	9	0	1
		1	1
3	21	0	2
		1	2
4	9	0	0
		1	0

**Şekil 27. Kohonen Ağları ile kümeleme analizi sonuçları**



Analiz sonucunda Excel çıktısı olarak elde edilen sınıflara ait iller ve sosyal faktörler ait veriler ise Ek'te gösterilmektedir. Sosyal sınıflar ve sınıflara ait iller ise özet halinde Tablo 8'de gösterilmektedir.

**Tablo 8. Sosyal sınıflar ve sınıflara ait iller**

İl Adı	Sosyal Sınıf	İl Adı	Sosyal Sınıf	İl Adı	Sosyal Sınıf
Afyon	1	Samsun	1	Bitlis	3
Amasya	1	Sinop	1	Diyarbakır	3
Artvin	1	Tekirdağ	1	Hakkari	3
Balıkesir	1	Tokat	1	Kahramanmaraş	3
Bilecik	1	Trabzon	1	Mardin	3
Bolu	1	Uşak	1	Muş	3
Burdur	1	Zonguldak	1	Niğde	3
Çanakkale	1	Bayburt	1	Siirt	3
Çankırı	1	Karaman	1	Sivas	3
Çorum	1	Kırıkkale	1	Şanlıurfa	3
Denizli	1	Bartın	1	Van	3
Erzincan	1	Ardahan	1	Yozgat	3
Erzurum	1	Yalova	1	Aksaray	3
Eskişehir	1	Karabük	1	Batman	3
Giresun	1	Düzce	1	Şırnak	3
Gümüşhane	1	Adana	2	Iğdır	3
Isparta	1	Elazığ	2	Kilis	3
Kars	1	Gaziantep	2	Osmaniye	3
Kastamonu	1	Hatay	2	Ankara	4
Kırklareli	1	Kayseri	2	Antalya	4
Kütahya	1	Kırşehir	2	Aydın	4
Manisa	1	Konya	2	Bursa	4
Muğla	1	Malatya	2	Edirne	4
Nevşehir	1	Tunceli	2	İçel	4
Ordu	1	Adıyaman	3	İstanbul	4
Rize	1	Ağrı	3	İzmir	4
Sakarya	1	Bingöl	3	Kocaeli	4

Uygulamanın bir sonraki aşamasında elde edilen sosyal sınıflara ait illerin tamamı MATLAB programına aktararak oluşturulan Yapay Sinir Ağının faktörler ve sınıflar arasındaki ilişkiyi öğrenebilmesi test edilmektedir. Bu bağlamda 81 ile ait 9 faktörün verileri girdi olarak, SPSS Clementine programında yapılan kümeleme analizi ile oluşturulan sınıflar ise çıktı olarak sisteme aktarılmaktadır.

#### 4.2.2. MATLAB'ta Yapay Sinir Ağının Oluşturulması ve Eğitim Aşaması

MATLAB programının Neural Networks Toolbox yazılımında verilerin işlenebilmesi için öncelikle bir eğitim seti hazırlanmalıdır. Eğitim seti verileri sisteme girildikten sonra ağın eğitilmesi aşamasına geçilebilir.

Çalışmanın kısıtları aşağıdaki gibidir:

- İllere göre hazırlanan tablolarda yer alan sosyal faktörlere ait veriler güncel verilere ulaşamadığından geçmiş dönem verileri kullanılmıştır.
- Kümeleme analizi sonucunda elde edilen sosyal sınıflar 4 gruba ayrılmaktadır. Ancak sosyal sınıfların önem dereceleri kümeleme analizi sonucunda belirtilememektedir. Bu nedenle mevcut deprem risk dereceleri ile kıyaslanması mümkün olamamaktadır.
- Ağın eğitilmesi aşamasında kullanılan veri setine bakıldığında çıktı olarak kullanılan 2. ve 4. sosyal sınıfa ait illerin az sayıda olduğu görülmektedir. İl sayısının daha fazla artırılmamasından dolayı ağın test edilmesi aşamasında hatalı sonuçların olabilmesi açısından bu eksiklik çalışmanın önemli bir kısıtı olarak değerlendirilmektedir.
- Yapay Sinir Ağları'nın eğitilmesi uzman görüşü gerektirmektedir. Ancak bu konuya ilişkin bir çalışma bulunmadığından eğitim aşamasında kullanılacak il sayısı, iterasyon sayısı, hedeflenen hata oranı, eğitim algoritmaları ve gizli katman sayısı gibi unsurlar için deneme yanılma yoluyla sonuçlara ulaşılmıştır.

İllere göre derlenen sosyal faktörlere ait veriler eğitim setinin girdi verilerini oluşturmaktadır. Eğitim setinin çıktı verileri ise kümeleme analizi ile daha önce belirlenen sosyal sınıflara ilişkin verilerdir. Tablo 9'da ağın eğitilmesi aşamasında kullanılan girdi verilerinden bazıları gösterilmektedir.

**Tablo 9. MATLAB eğitim seti**

İller	Afyon	Sakarya	Kayseri	Sivas	İstanbul
Okuma Yazma Bilmeyen	16910	5,00	6,00	81,00	6,00
İlk ve Orta Öğretim Mezunu	210444	73,00	67,00	1001,00	110,00
Yüksek Öğrenim Mezunu	28999	868,00	603,00	551,00	609,00
Konut Sayısı	171512	48647,00	65022,00	640197,00	112877,00
Doğurganlık Hızı	2,82	1,93	2,12	1,98	1,68
Kişi Başına Gelir (Bin TL)	533,8	1009,50	639,60	1628,50	934,00
Net Göç	-2033	2684,00	-515,00	10247,00	610,00
İşsizlik %	7,7	11,50	6,40	10,80	7,00
Nüfus	812416	270654,00	256803,00	2125140,00	464975,00

MATLAB programına eğitim seti olarak verilen girdilere bakıldığında her sütun bir ili temsil etmektedir. Çıkış verileri olan risk dereceleri de sisteme yine bir matris tablo aracılığıyla hazırlanıp aktarılmaktadır. Hazırlanan matris tablo ikili sistemde olmalıdır. 9x81 boyutundaki eğitim setine karşılık 4x81 boyutunda çıkış verileri hazırlanmıştır. Örnek girdi verilerine göre hazırlanmış çıkış setine ait veriler Tablo 10'da gösterilmektedir.

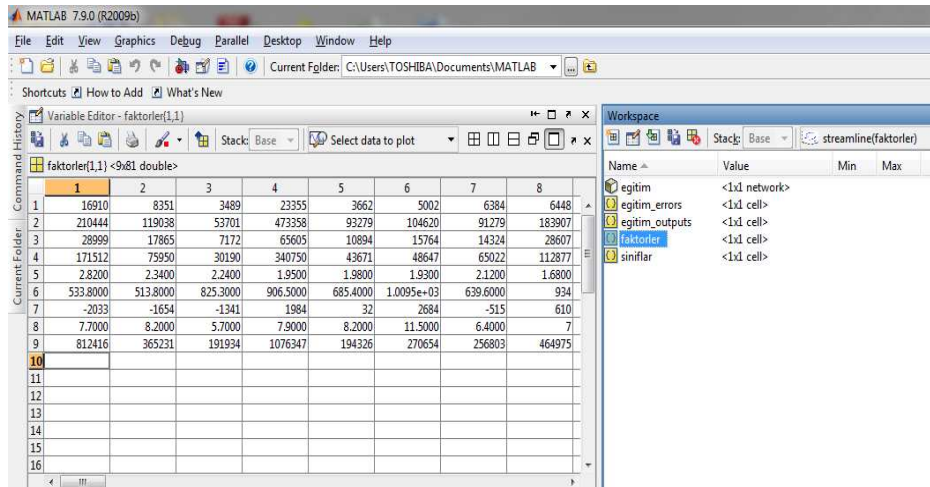
**Tablo 10. MATLAB eğitim seti çıkış verileri**

Sosyal Sınıflar	İller				
	Afyon	Sakarya	Kayseri	Sivas	İstanbul
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	0	0	1

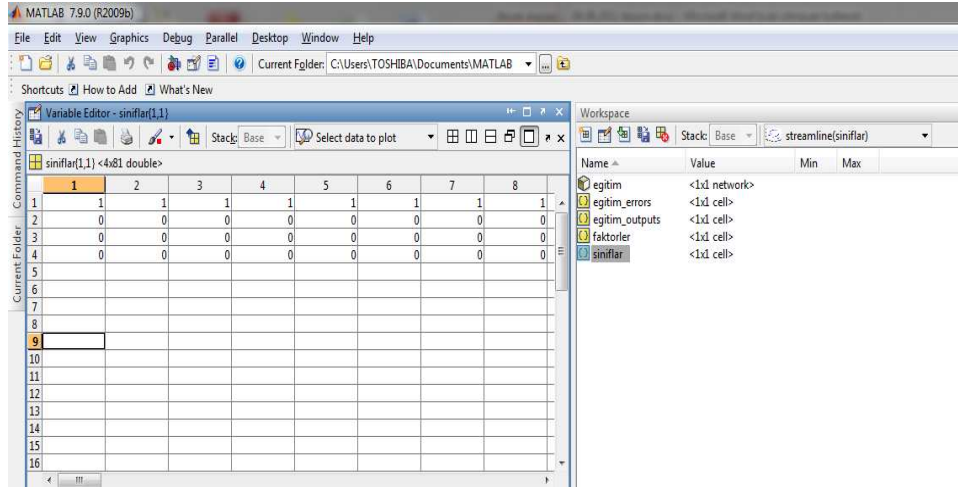
Tablo 10'da gösterilen ikili düzende hazırlanmış verilere bakıldığında, satırlar sosyal sınıfları ve sütunlar da illeri temsil etmektedir. Örneğin, matrisin 1. sütununa ait veriler (1,0,0,0), eğitim setinde yer alan Afyon ilinin 1. sosyal sınıfa ait olduğunu gösterir. Bu durumda Afyon ve Sakarya 1. sınıfa aitken, Kayseri 2. sınıfta, Sivas 3. sınıfta ve İstanbul ise 4. sınıfta yer almaktadır.

MATLAB'da eğitim setine ait girdi verileri "faktorler" adı altında 9x81 boyutunda matris olarak, eğitim setine ait çıktı verileri ise "siniflar" adı altında 5x81 boyutunda matris olarak aktarılmaktadır. Şekil 28 ve 29'da mevcut durum gösterilmektedir.

**Şekil 28. Eğitim seti giriş verilerinin MATLAB programına aktarılması**



## Şekil 29. Eğitim seti çıkış verilerinin MATLAB programına aktarılması



Ağın eğitilmesi aşamasında eğitilecek veriler sisteme aktarılırken iki fark yol izlenebilmektedir. Birincisi arayüz kullanılarak aktarılması, diğeri ise formül kullanılarak aktarılması olarak ifade edilmektedir. Yapar Sınır Ağının eğitilmesi aşamasında kullanılan formül aşağıdaki gibi gösterilmektedir.

### Ağın eğitilmesi aşamaları

```
egitim=newff(minmax(faktorler),[10,20,1],{'tansig','tansig','purelin'},'trainlm');
```

```
egitim.trainParam.perf='mse';
```

```
egitim.trainParam.epochs=1000;
```

```
egitim.trainParam.goal=1e-10;
```

```
egitim =train(egitim, faktorler,siniflar);
```

### Eğitim verisinin simülasyonu

```
sim_egitim=sim(egitim,faktorler);
```

```
sim_siniflar=mapminmax('reverse',sim_siniflar,ets);
```

### Test verisinin simülasyonu

```
sim_il adı=sim(egitim,faktorler);
```

```
sim_il adı=mapminmax('reverse',sim_il adı,tts);
```

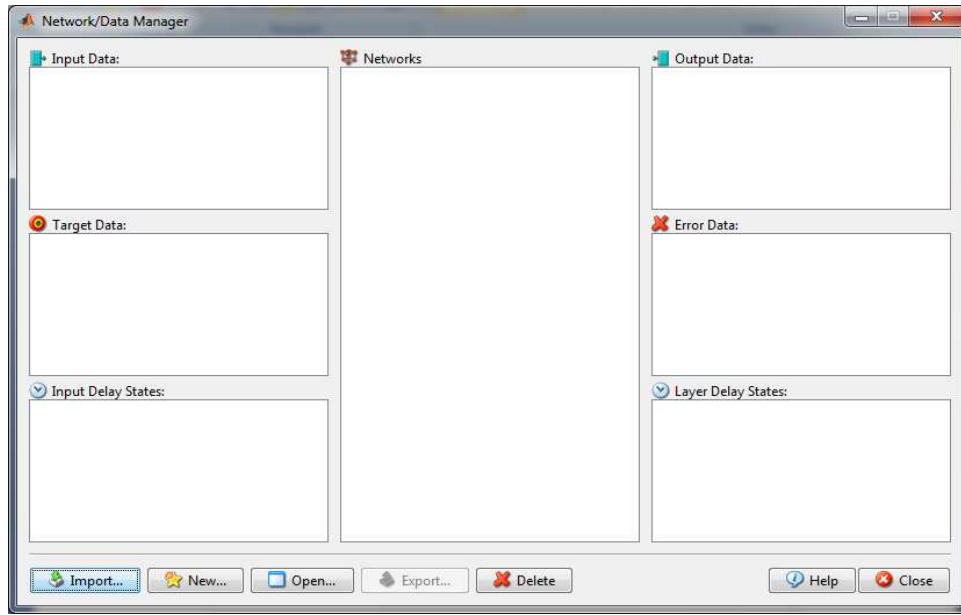
Burada, “epoch” parametresi iterasyon sayısını, “goal” parametresi hedeflenen hata

değerini, belirtmektedir. Yani “epoch” veya “goal” parametrelerinden herhangi biri sağlandığında eğitim işlemi durdurulacaktır.

Ağın eğitilmesi aşamasında iki gizli katman tanımlanmaktadır. Bu gizli katmanların ilkinde 10 ve ikincisinde de 20 nöron bulunduğu görülmektedir. Ayrıca iterasyon sayısı 1000 ve hedeflenen hata oranı da  $1.10^{-10}$  olarak ifade edilmektedir. Test aşamasında ise test edilecek ilin verileri eğitim setinden veya test setinden ağa aktarılarak çıktılar alınmaktadır.

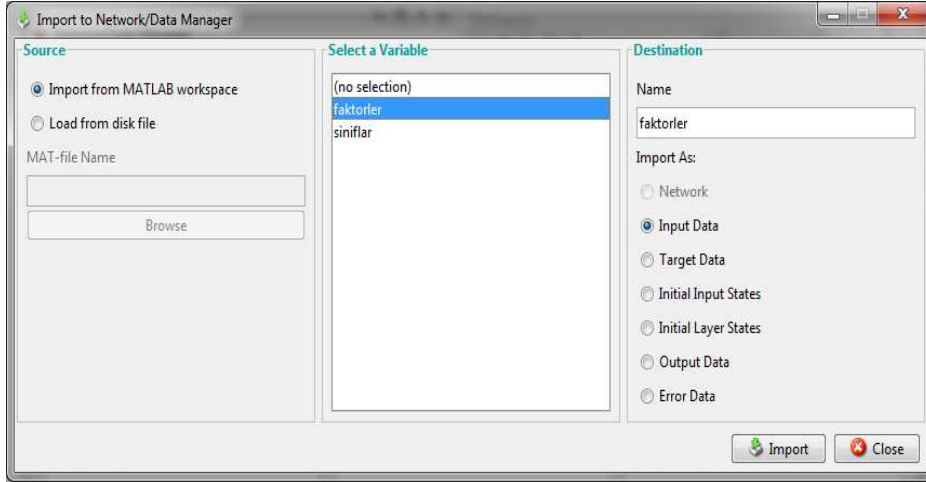
Eğitilecek verilerin ağa aktarılmasında kullanılan diğer yolda ise istenilen parametreler arayüz ekranından seçilerek tanımlanmaktadır. Şekil 30’da “Neural Networks Toolbox” modülünün arayüzü gösterilmektedir.

**Şekil 30. Neural Networks Toolbox modülü**



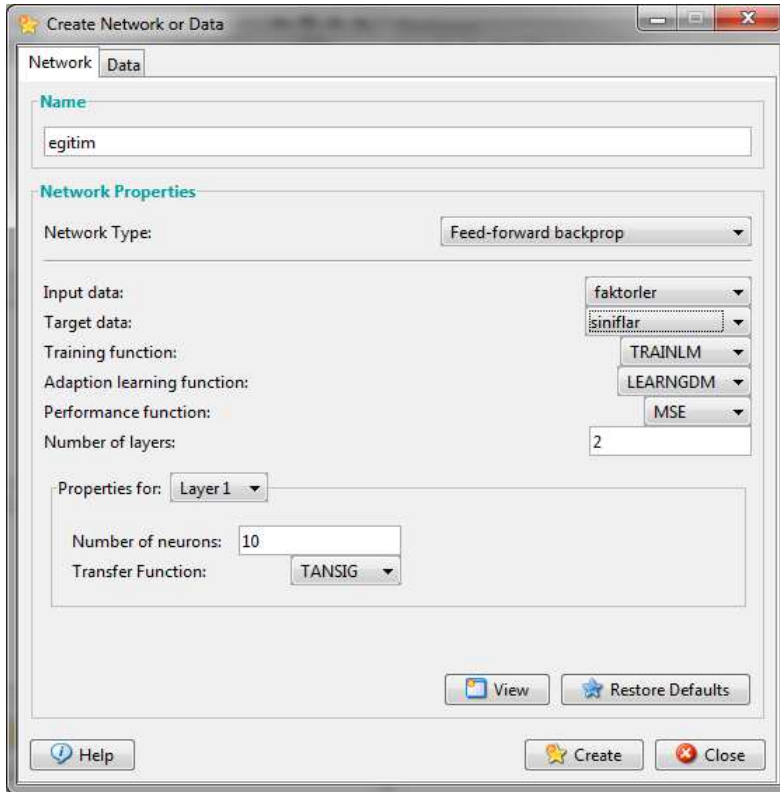
Arayüze bakıldığında giriş verisinin (input data), çıkış verisinin (target data), eğitilecek ağın (networks), eğitim sonrasında elde edilecek çıktının (output data) ve hata verilerinin (error data) gösterildiği ekran görülmektedir. İçeri aktarım (import), bölümüne girildiğinde ise ağın eğitilmesi aşamasında kullanılacak verilerin sisteme aktarılması sağlanmaktadır. Şekil 31’de “Import to Network / Data Manager” arayüzü gösterilmektedir.

**Şekil 31. Import to Network / Data Manager arayüzü**



Aktarım işlemi bittikten sonra sıra sisteme yeni bir ağ tanımlamaya gelir. Bunun için de ilk ekranda görülen “New” bölümüne girilerek mevcut verilerden sistemde eğitilecek olan ağ tanımlanır. Şekil 32’te ağın tanımlanması gösterilmiştir.

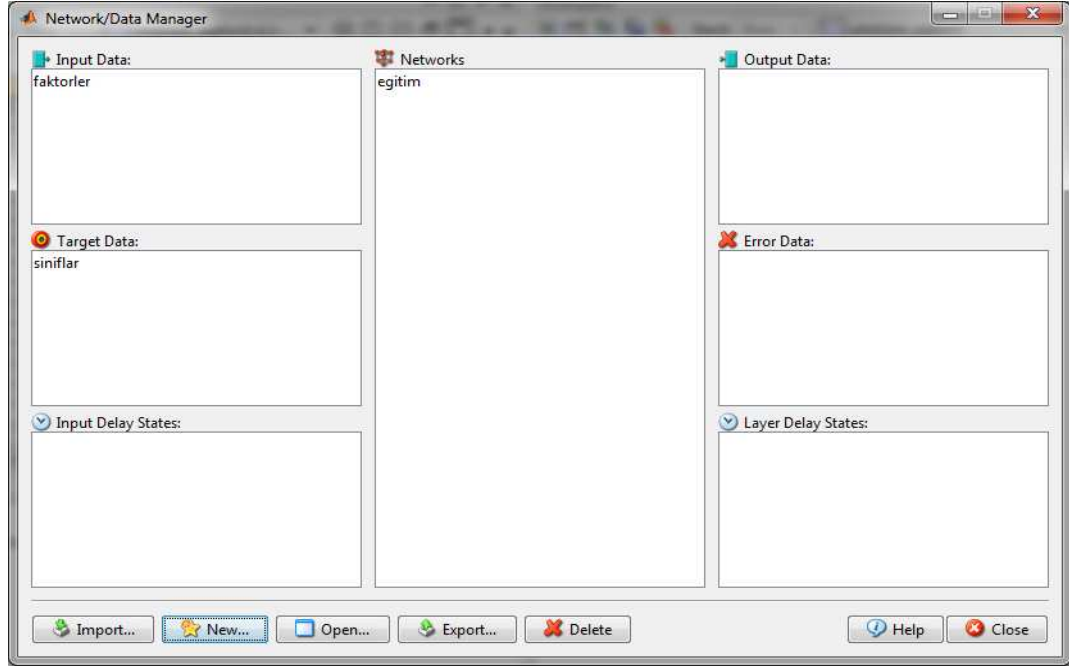
**Şekil 32. Eğitilecek ağın tanımlanması**



Arayüze bakıldığında daha önce formülle belirtilen eğitim parametrelerinin burada da seçilebildiği görülmektedir. Verilerin sisteme aktarılmasından sonra “Neural Networks Toolbox” arayüzündeki değişiklikler ise Şekil 33’te gösterilmiştir.

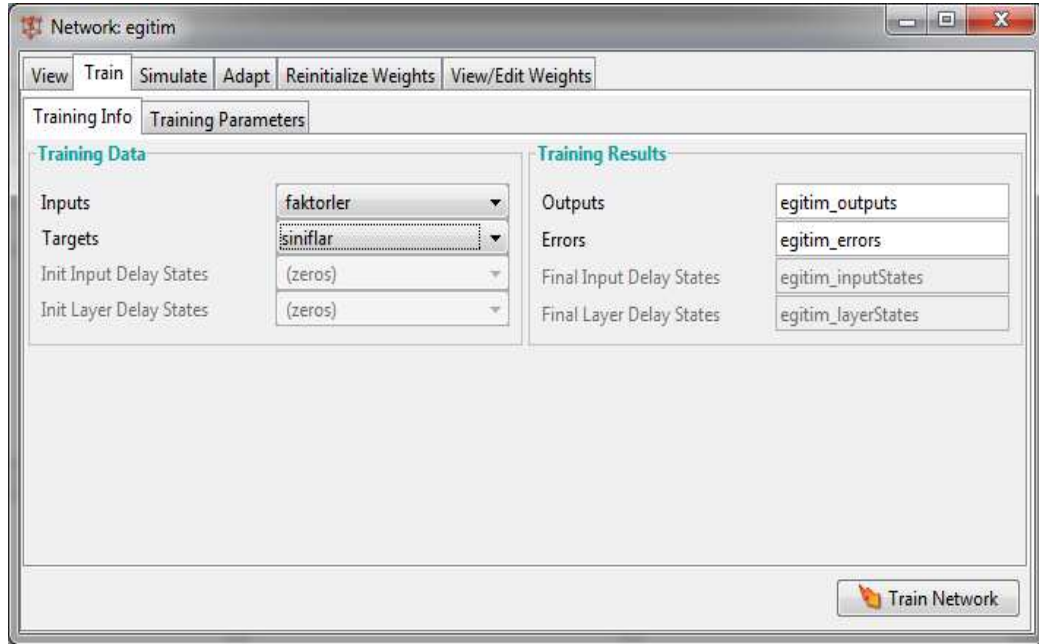


**Şekil 33. Veriler aktarıldıktan sonra Network/Data Manager arayüzü**



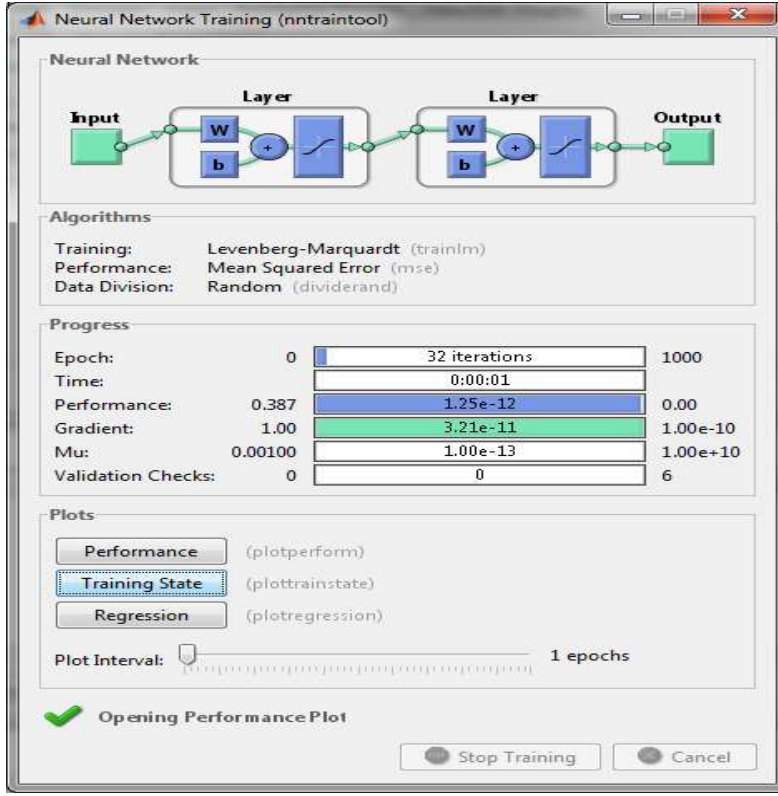
Verilerin sisteme aktarılmasından sonraki ilk adımda ağı eğitilmesi gerekmektedir. Bu eğitim, “egitim” adı ile tanımlanan ağ ile yapılmaktadır. Verilerin eğitildiği modül ve sisteme aktarılan verilerin seçilmesi Şekil 34’te gösterilmektedir.

**Şekil 34. MATLAB’ta ağı eğitilmesi**



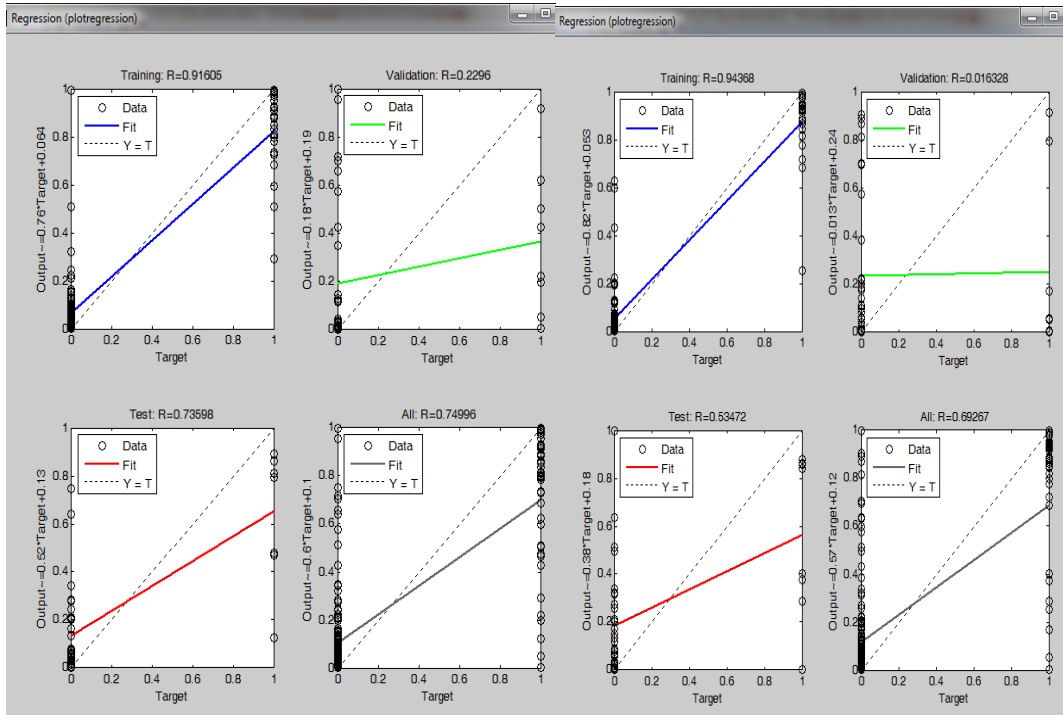
Veriler aktarıldıktan sonra eğitimin başlatılacağı arayüz Şekil 35’te gösterilmektedir.

Şekil 35. Neural Network Training arayüzü



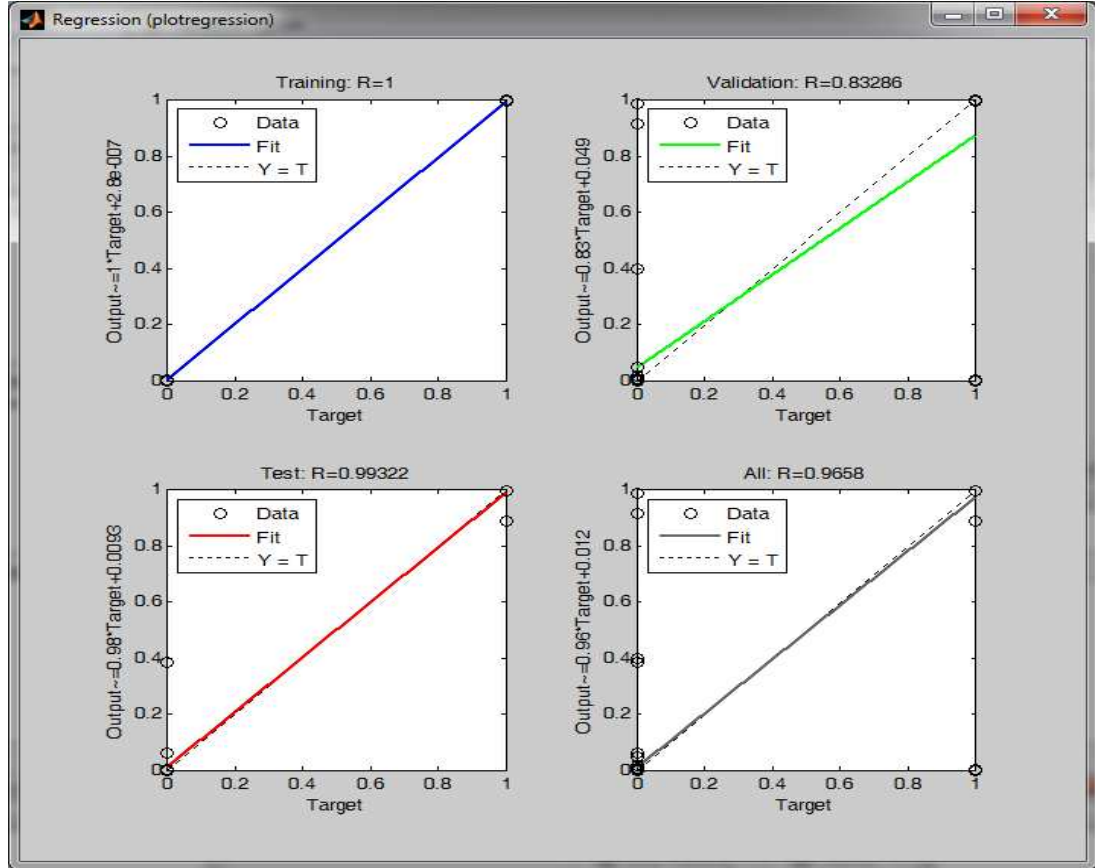
Deneme yanılma yoluyla yapılan çalışma sonucunda eğitimden alınan sonuçların bazıları Şekil 36'da gösterilmektedir.

Şekil 36. Denenen iterasyonların regresyon sonuçları



Ağın eğitilmesi istenilen sonuca ulaştığında ise regresyon katsayısının 1 olduğu görülmektedir. Şekil 37’de gösterilen sonuç denenen iterasyonların en iyisidir ve ağın eğitilmesi tam olarak sağlanmıştır. Eğitim aşaması tamamlanan ağ, test verilerinin simülasyonu için hazır hale getirilmiştir.

**Şekil 37. Eğitimin tamamlanmasından sonra elde edilen regresyon sonucu**



Çalışmada yapılan teste tüm illerinin değerleri test aşamasında kullanılmakta ve simüle edilmektedir. Kullanılan formül ise  $sim\_“il\ adı” = sim(egitim, “il\ adı”)$  şeklindedir. Formüle göre test girdisi “il adı” olarak ifade edilen ilgili ille ait verilerden oluşmaktadır. Daha önce eğitimi tamamlanan “egitim” ağı ile karşılaştırma yapılarak, test çıktısı  $sim\_“il\ adı”$  adı altında sisteme yüklenmektedir. Çıktıda elde edilen sonuçlar 4x1 boyutlu bir sütun matrisi olarak karşımıza çıkmaktadır. Test sonucunda elde edilen matrisin her satırı ilgili ilin, hangi sosyal sınıfa ne düzeyde ait olduğunu göstermektedir. 1 ve 0 olarak girilen veriler doğrultusunda test sonuçlarından elde edilen verilerde her sütunda 1’e en yakın değer bulunduğu satır, ilgili ilin dahil olduğu sosyal sınıfı belirtmektedir. Tablo 11’de illere göre test sonuçları gösterilmektedir.

**Tablo 11. İllere göre test sonuçları**

İller	Afyon	Amasya	Artvin	Bahkesir	Bilecik	Bolu	Burdur	Çanakkale	Çankırı
Sosyal Sınıf	1	1	1	1	1	1	1	1	1
Test Sonucu	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
İller	Çorum	Denizli	Erzincan	Erzurum	Eskişehir	Giresun	Gümüşhane	Isparta	Kars
Sosyal Sınıf	1	1	1	1	1	1	1	1	1
Test Sonucu	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
İller	Kastamonu	Kırklareli	Kütahya	Manisa	Muğla	Nevşehir	Ordu	Rize	Sakarya
Sosyal Sınıf	1	1	1	1	1	1	1	1	1
Test Sonucu	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
İller	Samsun	Sinop	Tekirdağ	Tokat	Trabzon	Uşak	Zonguldak	Bayburt	Karama
Sosyal Sınıf	1	1	1	1	1	1	1	1	1
Test Sonucu	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
İller	Kırıkkale	Bartın	Ardahan	Yalova	Karabük	Düzce	Adana	Elazığ	Gaziantep
Sosyal Sınıf	1	1	1	1	1	1	2	2	2
Test Sonucu	0,8886	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	1,0000	1,0000
	0,3848	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0016	0,0000	0,0000

Tablo 11'in devamı

<b>İller</b>	<b>Hatay</b>	<b>Kayseri</b>	<b>Kırşehir</b>	<b>Konya</b>	<b>Malatya</b>	<b>Tunceli</b>	<b>Adıyaman</b>	<b>Ağrı</b>	<b>Bingöl</b>
<b>Sosyal Sınıf</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>3</b>
<b>Test Sonucu</b>	0,0000	0,0466	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	1,0000	0,0000	1,0000	0,0000	1,0000	1,0000	0,0000	0,0000	0,0000
	0,0000	0,0169	0,0000	0,9886	0,0000	0,0000	1,0000	1,0000	1,0000
	0,0000	0,0005	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
<b>İller</b>	<b>Bitlis</b>	<b>Diyarbakır</b>	<b>Hakkari</b>	<b>Kahramanmaraş</b>	<b>Mardin</b>	<b>Muş</b>	<b>Niğde</b>	<b>Siirt</b>	<b>Sivas</b>
<b>Sosyal Sınıf</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	
<b>Test Sonucu</b>	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0001	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
<b>İller</b>	<b>Şanlıurfa</b>	<b>Van</b>	<b>Yozgat</b>	<b>Aksaray</b>	<b>Batman</b>	<b>Şırnak</b>	<b>Iğdır</b>	<b>Kilis</b>	<b>Osmaniye</b>
<b>Sosyal Sınıf</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	
<b>Test Sonucu</b>	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
<b>İller</b>	<b>Ankara</b>	<b>Antalya</b>	<b>Aydın</b>	<b>Bursa</b>	<b>Edirne</b>	<b>İçel</b>	<b>İstanbul</b>	<b>İzmir</b>	<b>Kocaeli</b>
<b>Sosyal Sınıf</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>	
<b>Test Sonucu</b>	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	0,0589	0,0000	0,0000	0,0000	0,0001	0,9171	0,3976	0,0080	0,0000
	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	1,0000	1,0000	1,0000	1,0000	0,9998	1,0000	1,0000	1,0000	1,0000

#### 4.2.3. Test Sonuçları ve Yorumları

Test sonuçlarına göre 81 ilin ait olduğu sosyal sınıflar ve bu sınıflara ait olma oranları aşağıdaki gibidir;

- Test sonucuna bakıldığında, Edirne, Kayseri, Kırıkkale ve Konya illeri hariç 77 ilin, kümeleme analizi sonucunda elde edilen sınıflara % 100 oranla atandığı görülmektedir.

- Sosyal sınıf numarası 2 olan Kayseri iline ait test sonucuna bakıldığında % 4,66 oranla 1. sınıfa atandığı görülmektedir.
- 2. grupta yer alan Konya iline ait test sonucuna bakıldığında % 98,86 oranla 3. gruba atandığı görülmektedir.
- Edirne (% 99,98) ve Kırıkkale (% 88,86) illerinin de 1'e yakın oranlarla ait oldukları sosyal sınıflara atandıkları görülmektedir.

2. sosyal sınıfa ait Kayseri ve Konya illerinin test sonucunda başka sınıflara atandığı görülmektedir. Ağın eğitilmesi aşamasında 2. sınıfa ait yalnızca 9 ilin bulunması bu tür bir yanılmaya neden olabilir. Daha iyi sonuçların elde edilmesi için ağ eğitilirken daha fazla veri kullanılmalıdır.

Karşılaştırma sonuçlarına göre test edilen 81 ilin çıktı değerlerinden 79'unun mevcut sosyal sınıflarla aynı olduğu görülmektedir. Bu da test sonuçlarının % 97,5 oranında başarılı olduğunu göstermektedir. Başarı oranının artırılması için daha fazla veriyle çalışılmalıdır.

## SONUÇ VE ÖNERİLER

Yapay Sinir Ağları çok değişkenli ve değişkenler arasındaki karmaşık, karşılıklı etkileşimin bulunduğu veya tek bir çözüm kümesinin bulunmadığı durumlarda başarılı sonuçlar üreten bir Yapay Zeka tekniğidir. Yapay Sinir Ağları, özellikle günümüze kadar çözümü güç ve karmaşık olan ya da ekonomik olmayan çok farklı alanlardaki problemlerin çözümüne uygulanmış ve genellikle başarılı sonuçlar elde edilerek, kendisine geniş bir uygulama alanı bulmuştur.

Çağımızda Afet Yönetimi'nin önemli olduğu ve bu konuda sıklıkla çalışmalar yapıldığı gözlenmiştir. Afet Yönetimi'nde; afetlerin önlenmesi ve sonrasında ortaya çıkabilecek fiziksel, sosyal veya ekonomik zararların azaltılması amaçlanmaktadır. Doğal afetlerin nerede ve ne zaman olacağı kesin olarak belirtilemezken, bilim ve teknolojiye dayalı Afet Yönetimi'nin önemi her geçen gün biraz daha artmaktadır.

Hızlı nüfus artışı, iç ve dış göçler, çarpık kentleşme, bilgi ve eğitim eksikliği gibi sosyal faktörlere yönelik iyileştirmelerin yapılabilmesi için Yapay Sinir Ağları gibi insan düşünüş ve yargısına daha yakın teknikler kullanılmalı ve Afet Yönetimi'ne ait çalışmalar geliştirilmelidir. Ayrıca bu tekniklerin klasik yöntemlere nazaran sosyal faktörlerle ilgili daha güvenilir sonuçların elde edilmesine katkı sağlayacağı ifade edilebilir.

Bu bağlamda çalışmanın amacı Yapay Sinir Ağları kullanılması suretiyle yalnızca fiziksel yapıya göre hazırlanan deprem risk derecelerinin sosyal faktörlerle de ilişkilendirilmesini sağlamaktır. Bu amaç doğrultusunda Türkiye'de bulunan 81 ilin sosyal faktörlere ait verileri Türkiye İstatistik Kurumu'ndan alınarak uygulamada kullanılacak veri setleri oluşturulmuş ve bu veri setleri SPSS Clementine programında sınıflandırılarak, Yapay Sinir Ağları modülünün bulunduğu MATLAB programına matris tablolar halinde aktarılmıştır.

Çalışmanın uygulama kısmında kullanılan veri seti için Türkiye İstatistik Kurumu verilerinden yararlanılmıştır. Ancak her veri güncel tarihli olmadığından değişikliklerin izlenememesi, sosyal faktörlerden yalnızca 9 kategorinin incelenmesi ve kesin sonuçlara deneme yanılma yoluyla ulaşılması zorunluluğu çalışmanın kısıtlarını oluşturursa da çalışma sonunda ağırlık eğitilmesi tamamlanmış ve sonuçlar analiz edilmiştir.

Eđitimi tamamlanan ađa herhangi bir ilin verileri gnderildiđinde hangi sosyal sınıfa ait olduđunun sistem tarafından yakınsama olarak ifade edildiđi grlmstr. Bylelikle illere gre sınıflandırılan sosyal sınıfların ile ve diđer yerleřim birimleri bazında da geliřtirilmesine zemin hazırlanmıřtır.

alıřmada elde edilen sonular incelendiđinde; sosyal faktrlerle kmeleme analizi sonucunda elde edilen sosyal sınıflar arasında iliřki kurulabildiđi saptanmıřtır. Sınıflara ait illerde afetlerle ortaya ıkabilecek olası kayıpların minimum seviyeye indirilebilmesi iin sosyal faktrlerin hangi dzeyde iyileřtirilmesi gerektiđinin saptaması Yapay Sinir Ađlarının kullanılmasıyla mmkn hale gelmiřtir.

Sonu olarak denilebilir ki; uzmanlar tarafından yapılacak Afet Ynetimi alıřmaları iin bu alıřma sonucunda elde edilen bulgular gznnde bulundurularak deprem risk derecelerinin belirlenmesinde disiplinlerarası alıřmalar yapılmalıdır.

Gelecek alıřmalar İin neriler:

- Bu alıřmada illere ait sosyal sınıflar oluřturulurken sosyal faktrlerden yararlanılmıřtır. Daha sonra yapılacak alıřmalarda fiziksel, sosyal ve ekonomik faktrlerin tamamından yararlanılarak deprem risk derecelerinin belirlenmesinde daha sađlıklı sonular retilir.
- Sosyal zarar grebilirliđin tespiti konusunda ile bazında analizler yapılarak alıřmalar daha detaylı hale getirilebilir.
- Bulanık mantıđın belirsiz bilgileri iřleme yeteneđinden ve yapay sinir ađının đrenme yeteneđinden yararlanabilmek iin, bu iki teknoloji, deđiřik yntemlerle birleřtirilebilmektedir. Aynı zamanda, bulanık mantık ve yapay sinir ađının birleřtirilmesi, birbirlerinin dezavantajlarını da rtmektedir.

Uygulamalarda Yapay Sinir Ađları ve Bulanık Mantık yaklařımlarının birlikte kullanılması da birbirlerinin eksik ynlerini tamamlamaları ve daha bařarılı sonular elde edilmesi aısından nem tařımaktadır. Bu bađlamda daha sonraki alıřmalarda iki tekniđin entegre kullanımıyla elde edilen Sinirsel Bulanık Mantık yaklařımı ile fiziksel, sosyal ve ekonomik faktrlerin tamamı ele alınarak deprem risk dereceleri hesaplanabilir.



- İllere ait güncel verilerin bulunmaması mevcut durumun tahmininde kısıt oluşturmaktadır. Sağlıklı sonuçların alınabilmesi için veriler güncellenmeli, hatta daha detaylı analizlerde kullanılmak üzere ilçe ve diğer yerleşim birimleri bazında da veriler hazırlanmalıdır.

## KAYNAKÇA

- AKDAĞ, S. Emre (2002), “Mali Yapı ve Denetim Boyutuyla Afet Yönetimi”, *Sayıştay Yayın İşleri Müdürlüğü, Araştırma/İnceleme/Çeviri Dizisi: 20*, Araştırma Raporu, Birinci Basım, Mart.
- ALABAŞ, Çiğdem (1999), *Tabu Arama ve Tavlama Benzetimi Algoritmalarıyla Bilgisayar Şebekelerinin Topolojik Optimizasyonu*, Yayınlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü.
- ALLAHVERDİ, Novruz (2002), *Uzman Sistemler - Bir Yapay Zeka Uygulaması*, Atlas Yayın Dağıtım.
- ALTAN, Şenol (2008), “Döviz Kuru Öngörü Performansı İçin Alternatif Bir Yaklaşım: Yapay Sinir Ağı”, *Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, Cilt 10, Sayı 2, s. 141-160.
- ASLANARGÜN, Atilla., Mammadagha Mammadov, Berna Yazıcı ve Şenay Yolaçan (2007), “Comparison of ARIMA, Neural Networks and Hybrid Models In Time Series: Tourist Arrival Forecasting”, *Journal of Statistical Computation and Simulation*, Vol. 77, No. 1, s.29-53.
- BARR, Avron, Paul R. Cohen ve Edward Feigenbaum (1989), *The Handbook Of Artificial Intelligence*, Vol. 4, William Kaufmann Inc.
- BAYKAL, Nazife ve Timur Beyan (2004), *Bulanık Mantık, Uzman Sistemler ve Denetleyiciler*, Bıçaklar Kitabevi, Ankara.
- BAYLAR, Ahmet, M. E. Emiroğlu ve A. Arslan (1999), “ Geriye Yayılma Yapay Sinir Ağı Kullanılarak Yanal Su Alma Yapısına Yönelecek Olan Sürüntü Maddesi Oranının Bulunması”, *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, Cilt 1, Sayı 2, s. 1-11.
- BAYRAMOĞLU, Mehmet Fatih (2007), *Finansal Endekslerin Öngörüsünde Yapay Sinir Ağı Modellerinin Kullanılması: İMKB Ulusal 100 Endeksinin Gün İçi En Yüksek Ve En Düşük Değerlerinin Öngörüsü Üzerine Bir Uygulama*, Yayınlanmamış Yüksek Lisans Tezi, Zonguldak Karaelmas Üniversitesi SBE.

- CEBESoy, T. (1995), “Çok Kriterli Bulanık Dominant Algoritmasına Dayalı Açık İşletme Ekipman Seçimi”, *Türkiye IS.Madencilik Kongresi*, ISBN 975-395-216-3
- CHEN, S. (1992), “Billings, S. A., “Neural Networks for Nonlinear Dynamic System Modeling and Identification”, *International J. Control*, Vol. 2, s. 319-346.
- CİVALEK, Ömer (2005), “Optimum Yapı Dizaynında Yapay Zeka Yaklaşımlar”, *Deprem Sempozyumu*, Kocaeli, s. 1325-1334.
- COŞKUN, Aysun (2007), “Yapay Zeka Optimizasyon Teknikleri: Literatür Değerlendirmesi”, *Firat Üniversitesi Doğu Anadolu Bölgesi Araştırma ve Uygulama Merkezi Dergisi*, Cilt 5, Sayı 2, s.142-146.
- DEMİR, R. ve F. Ülengin (2008), “Yapay Sinir Ağları Yardımı İle Şirket Birleşmelerinin Kestirimi”, *İTÜ Dergisi/d*, Cilt 7, Sayı 6, s. 92–101.
- DURMUŞ, Ali (2005), *Yapay ve Bulanık Sinir Ağları İle Sistemlerin Modellenmesi*, Yayınlanmamış Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü.
- ELMAS, Çetin (2003a), *Bulanık Mantık Denetleyiciler (Kuram, Uygulama, Sinirsel Bulanık Mantık)*, Seçkin Yayıncılık, 1. Baskı, Ankara.
- ELMAS, Çetin (2003b), *Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama)*, Seçkin Yayıncılık, 1. Baskı, Ankara.
- ERDOĞAN, T. Y. (2003), *Beton*, ODTÜ Geliştirme Vakfı Yayıncılık ve İletişim Şti, Ankara, s. 140-160
- ERGEZER, H., M. Dikmen ve E. Özdemir (2003), “Yapay Sinir Ağları ve Tanıma Sistemleri”, *Pivolka*, Cilt 2, Sayı 6, s. 14-17.
- ERGÜLEN, Ahmet ve Ali DERAN (2009), “Taşıma Maliyetlerinin Bulanık Mantık Yaklaşımı İle Yönetilmesi Ve Finansal Performans Üzerindeki Etkisinin İncelenmesi”, *MUFAD Dergisi*, Temmuz.
- ERGÜNAY, Oktay (2002), “Afete Hazırlık ve Afet Yönetimi”, *Türkiye Kızılay Derneği Genel Müdürlüğü Afet Operasyon Merkezi (AFOM)*, Ankara, s. 1-44.

ERGÜNAY, Oktay (2008), “Afet Yönetiminde Kurumsal Yapılanma ve Mevzuat Nedir? Nasıl Olmalıdır?”, *İstanbul Depremini Beklerken*, *Sorunlar ve Çözümler Bildiriler Kitabı, 20 Eylül 2008 CHP İstanbul Deprem Sempozyumu*, s. 97-108.

ERGÜNAY, Oktay (2009a), “Doğal Afetler Ve Sürdürülebilir Kalkınma”, *Abant İzzet Baysal Üniversitesi 11-12 Kasım Deprem Sempozyumu*, s. 1-11.

ERGÜNAY, Oktay (2009b), “Afet Yönetimi: Genel İlkeler, Tanımlar, Kavramlar”, *Afet İşleri Eski Genel Müdürlüğü*, Ankara, s. 1-49.

FIRAT, Mahmut ve Mahmud Güngör (2004), “Askı Madde Konsantrasyonu ve Miktarının Yapay Sinir Ağları ile Belirlenmesi”, *İMO Teknik Dergi*, Yazı 219, s. 3267-3282.

GLOVER, F. (1989), “Tabu Search-Part I”, *ORSA Journal of Computing*, Vol. 1, No. 3.

GÖNÜL, İ. A. ve G. Çelebi (2003), “Binalarda Zeminden Kaynaklanan Nemlenmeyi Önleme Yöntemleri”, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, Cilt 18, Sayı 4, s. 109-122.

GÜDEN Hüseyin, Barış Vakvak, Barış E. Özkan, Fulya Altıparmak ve Berna Dengiz (2005), “Genel Amaçlı Arama Algoritmaları İle Benzetim Eniyilemesi: En İyi Kanban Sayısının Bulunması”, *Makine Mühendisleri Odası Endüstri Mühendisliği Dergisi*, Cilt 16, Sayı 1, s. 2-15.

GÜLER, H. Hüseyin (2008), *Afet Yönetimine Giriş*, Ankara Üniversitesi, Mühendislik Fakültesi, Jeofizik Mühendisliği Bölümü, Ders Notu, s. 1-24.

GÜRGEN, Güneş (2008), *Birliktelik Kuralları İle Sepet Analizi ve Uygulaması*, Yayınlanmamış Yüksek Lisans Tezi, Marmara Üniversitesi Sosyal Bilimleri Enstitüsü.

<http://www.dmc.metu.edu.tr/tr/>, 20.06.2011

[http://www.mathworks.com/access/helpdesk/help/pdf\\_doc/nnet/nnet.pdf](http://www.mathworks.com/access/helpdesk/help/pdf_doc/nnet/nnet.pdf), 10.03.2011

<http://www.mathworks.com/products/neuralnet/description3.html>, 22.04.2011

<http://www.yapay-zeka.org/kullanim-alanlari.htm>, 12.05.2011

- İNÇETÜRKMEN, Berrin (1999), *Bir Esnek İmalat Sisteminde Yapay Sinir Ağları Kullanılarak Üretim Hatlarının Belirlenmesi*, Yayınlanmamış Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü.
- JINGLU, H. (1999), “Fuzzy Modelling and Identification”, *Katholieke Universiteit Leuven*.
- KARANFİL, S. (1993), *Bulanık Kümeler ve Bulanık Mantığın Temelleri*, Yayınlanmamış Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.
- KAYABAŞ, İlker (2010), *Yapay Zeka Sohbet Ajanlarının Uzaktan Eğitimde Öğrenci Destek Sistemi Olarak Kullanılabilirliği*, Yayınlanmamış Yüksek Lisans Tezi, Anadolu Üniversitesi Sosyal Bilimler Enstitüsü.
- KOHONEN, T. (1982), “Self-Organised Formation of Topologically Correct Feature Maps”, *Biological Cybernetics*, Vol. 43, s. 59-69.
- KÖSE, Utku (2010), *Bulanık Mantık Ve Yapay Sinir Ağları İçin Eğitim Yazılımı Geliştirilmesi*, Yayınlanmamış Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü.
- MURAT, Y. Şazi ve Nurcan Uludağ (2008), “Bulanık Mantık ve Lojistik Regresyon Yöntemleri ile Ulaşım Ağlarında Geçki Seçim Davranışının Modellenmesi”, *İMO Teknik Dergi*, s. 4363-4379.
- NAUCK, D. ve R. Kruse (1994), “Choosing Appropriate Neuro-Fuzzy Models”, *Proc. Third European Congress on Intelligent Techniques and Soft Computing*, Aachen.
- ÖZALP, Alperen ve A. S. Anagün (2003), “Sektörel Hisse Senedi Fiyat Tahmininde Yapay Sinir Ağı Yaklaşımı ve Klasik Tahminleme Yöntemleriyle Karşılaştırılması”, *Endüstri Mühendisliği Dergisi*, Cilt 12, Sayı 3-4, s. 2-17.
- ÖZÇAKAR, N. (1998), “Genetik Algoritmalar”, *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, Cilt: 27, Sayı: 1, s. 69-82.
- ÖZKUL Barış ve Erkan Karaman (2007), “Doğal Afetler İçin Risk Yönetimi”, *TMMOB Afet Sempozyumu*, s.251-260.

- ÖZTEMEL, Ercan (2003), *Yapay Sinir Ağları*, Papatya Yayıncılık, I. Baskı, İstanbul.
- SAĞIROĞLU, Ş., E. Beşdok ve E. Erler (2003), *Mühendislikte Yapay Zeka Uygulamaları - I: Yapay Sinir Ağları*, Ufuk Yayıncılık, Kayseri.
- SATTARI, Mohammad T., A. Fahker Fard, Mohammad Docherkhesaz ve Fazlı Öztürk (2007), “Yapay Sinir Ağları Yönetimi İle Savalan Sulama Rezervuarının Simülasyonu”, *Ankara Üniversitesi Ziraat Fakültesi Tarım Bilimleri Dergisi*, Cilt 13, Sayı 4, s. 337-345.
- SERHATLIOĞLU, Selami ve Fırat Hardalaç (2009), “Yapay Zeka Teknikleri ve Radyolojiye Uygulanması”, *Fırat Tıp Dergisi*, Cilt 14, Sayı 1, s. 1-6.
- SOFYALIOĞLU, Çiğdem (2009), “Bulanık Analitik Hiyerarşi Süreci İle Uygun Altı Sigma Metodolojisinin Seçimi”, *Yönetim ve Ekonomi Dergisi*, Cilt 16, Sayı 2, s. 1-17.
- SUBAŞI, Handan (2010), *Yapay Sinir Ağı İle Atıksu Arıtma Performansının Modellenmesi*, Yayınlanmamış Yüksek Lisans Tezi, Çukurova Üniversitesi Fen Bilimleri Enstitüsü.
- ŞENOL, Fikriye (2000), *Bulanık Mantık Kontrolcüsü*, Yayınlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi.
- TAŞKIN, Çağatan ve Gül Gökay Emel (2010), “Veri Madenciliğinde Kümeleme Yaklaşımları Ve Kohonen Ağları İle Perakendecilik Sektöründe Bir Uygulama”, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, Cilt: 15, Sayı: 3, s. 395-409.
- TORTUM, Ahmet, Nadir YAYLA ve Mahir GÖKDAĞ (2007), “Yapay Sinir Ağları Ve Birleştirilmiş Sinirsel Bulanık Sistemler İle Şehirlerarası Yük Taşınması Tür Seçiminin Modellenmesi”, <http://www.e-kutuphane.imo.org.tr/pdf/3186.pdf>, 25.04.2011.
- TUNALI, Ayfer (2007), *Bulanık Sinir Ağları ve Sinirsel Bulanık Ağları İle Sistem Çözümlemesi ve Simülasyonunun Gerçekleştirilmesi*, Yayınlanmamış Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

- TÜRKOĞLU, A. (2008), *Temel Dişli Mekanizmalarının Yapay Sinir Ağları İle Web Tabanlı Kinematik Tasarımı*, Yayınlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü.
- UYGUNOĞLU, Tayfun ve Şaban Yurtçu (2006), “Yapay Zeka Tekniklerinin İnşaat Mühendisliği Problemlerinde Kullanımı”, *Yapı Teknolojileri Elektronik Dergisi*, www.teknolojikarastirmalar.com, Sayı: 1, s.61-70.
- UZUNÇIBUK, Levent (2005), *Yerleşim Yerlerinde Afet ve Risk Yönetimi*, Yayınlanmamış Doktora Tezi, Ankara Üniversitesi Sosyal Bilimleri Enstitüsü.
- WHITBY, Blay (2005), *A Beginner’s Guide: Artificial Intelligence*, Çev., Çiğdem Karabağlı, İletişim Yayıncılık, 1. Baskı, İstanbul.
- WINSTON, “Patrick H. (1984), *Artificial Intelligence*, 2<sup>rd</sup> Edition, Addison-Wesley Publishing Company.
- YURTOĞLU, Hasan (2005), *Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi: Bazı Makroekonomik Değişkenler İçin Türkiye Örneği*, Ekonomik Modeller ve Stratejik Araştırmalar Genel Müdürlüğü, Yayın No: DPT 2683.
- ZHANG, G., B. E. Patuwo ve M. Y. Hu (1998), “Forecasting with Artificial Neural Networks: The State of the Art”, *International Journal of Forecasting*, Vol. 14, s. 35-62.
- ZURADA, J. M. (1992), *Introduction of Artificial Neural Systems*, St. Paul: West Publishing.

## EKLER

### EK 1: İllere Göre Eğitim Durumu

İl Adı	Okuma Yazma Bilmeyen	İlk ve Orta Öğretim Mezunu	Yüksek Öğrenim Mezunu	İl Adı	Okuma Yazma Bilmeyen	İlk ve Orta Öğretim Mezunu	Yüksek Öğrenim Mezunu
Amasya	8351	119038	17865	Diyarbakır	131556	395788	43574
Aydın	25175	327393	55090	Elazığ	33466	197119	29520
Balıkesir	23355	473358	65605	Erzurum	31010	243361	36595
Bilecik	3662	93279	10894	Eskişehir	17307	413860	69798
Bingöl	14276	62910	6575	Kars	11422	112766	15456
Bitlis	17537	68975	7877	Kütahya	30708	242867	37176
Bolu	5002	104620	15764	Samsun	40203	452868	65244
Burdur	6384	91279	14324	Tekirdağ	23313	192551	26609
Bursa	80651	1317914	164895	Tunceli	103988	263570	22116
Çanakkale	6448	183907	28607	Uşak	11223	131088	18017
Çankırı	5345	65292	7944	Van	61338	190636	19815
Denizli	24018	364829	53691	Zonguldak	11284	167122	24690
Erzincan	5883	65796	10481	Batman	41403	124605	12023
Hakkari	16304	54390	6066	Şırnak	36614	80400	7823
Hatay	44630	360774	45947	Ardahan	2158	18398	2714
Isparta	8347	166561	30790	Iğdır	10052	37839	4856
İstanbul	440039	7030705	1077409	Artvin	3489	53701	7172
İzmir	129652	2061759	351402	Gaziantep	99935	637811	54399
Kastamonu	44872	548357	73263	Gümüşhane	1962	32736	4517
Kırşehir	50628	807296	101395	İçel	73413	670457	92982
Kocaeli	62317	816796	104546	Kayseri	6724	136751	18554
Malatya	46167	509979	60816	Mardin	50853	155619	13671
Manisa	40109	299759	34450	Nevşehir	6970	86734	11644
Kahramanmaraş	8821	62391	9165	Ordu	29135	217528	26436
Muğla	8154	197451	38949	Sivas	23030	226702	32475
Muş	14692	50803	4938	Şanlıurfa	16664	319078	39969
Sakarya	23522	370542	41623	Yozgat	13383	144571	14958
Siirt	22471	65757	7523	Bayburt	1823	20426	2690
Tokat	18864	225914	40566	Kilis	6510	40351	4165
Kırkkale	12028	135813	15551	Ankara	133714	2471157	593193
Bartın	1975	36744	6919	Edirne	11841	160394	25583
Yalova	4087	75943	10482	Giresun	15040	135029	20214
Karabük	8244	101003	13401	Kırklareli	7974	88956	12536
Osmaniye	22790	171033	20175	Konya	14221	225377	28282
Düzce	7744	108384	12928	Niğde	6574	88062	14168
Adana	118567	922105	118540	Rize	9136	110202	14352
Adıyaman	29908	160751	17013	Sinop	4606	61012	9378
Afyon	16910	210444	28999	Trabzon	3393	30874	4908
Ağrı	27600	82152	7910	Aksaray	12869	117449	13161
Antalya	40303	745900	124834	Karaman	5782	91747	10359
Çorum	20321	196472	22769				

Kaynak: (www.tuik.gov.tr)



**EK 2: İllere Göre Konut Sayısı**

İl Adı	Konut Sayısı	İl Adı	Konut Sayısı	İl Adı	Konut Sayısı
Amasya	75950	Siirt	30244	Şırnak	38184
Aydın	274260	Tokat	133690	Ardahan	8079
Balıkesir	340750	Kırıkkale	83177	Iğdır	20691
Bilecik	43671	Bartın	29927	Artvin	30190
Bingöl	28073	Yalova	80284	Gaziantep	279617
Bitlis	36149	Karabük	56725	Gümüşhane	25977
Bolu	48647	Osmaniye	85733	İçel	440184
Burdur	65022	Düzce	37197	Kayseri	273620
Bursa	640197	Adana	469189	Mardin	87668
Çanakkale	112877	Adıyaman	75690	Nevşehir	75838
Çankırı	43616	Afyon	171512	Ordu	163529
Denizli	234168	Ağrı	41151	Sivas	117149
Erzincan	51796	Antalya	456371	Şanlıurfa	158645
Hakkari	20392	Çorum	114547	Yozgat	111431
Hatay	273294	Diyarbakır	200351	Bayburt	11369
Isparta	122422	Elazığ	109729	Kilis	20397
İstanbul	3393077	Erzurum	117810	Ankara	1128625
İzmir	1140731	Eskişehir	207717	Edirne	94979
Kastamonu	69899	Kars	29557	Giresun	95659
Kırşehir	55573	Kütahya	154313	Kırklareli	83150
Kocaeli	352079	Samsun	255042	Konya	469894
Malatya	154466	Tekirdağ	222641	Niğde	79757
Manisa	304817	Tunceli	12930	Rize	75972
Kahramanmaraş	166693	Uşak	82656	Sinop	40363
Muğla	194620	Van	80101	Trabzon	195111
Muş	36019	Zonguldak	142825	Aksaray	81540
Sakarya	156386	Batman	55577	Karaman	55882

Kaynak: (www.tuik.gov.tr)

**EK 3: İllere Göre Doğurganlık Hızı**

İl Adı	Doğurganlık Hızı	İl Adı	Doğurganlık Hızı	İl Adı	Doğurganlık Hızı
Amasya	2,34	Siirt	6,05	Şırnak	7,06
Aydın	2,12	Tokat	3,06	Ardahan	2,95
Balıkesir	1,95	Kırıkkale	2,39	Iğdır	4,17
Bilecik	1,98	Bartın	2,11	Artvin	2,24
Bingöl	3,56	Yalova	1,93	Gaziantep	3,83
Bitlis	5,03	Karabük	1,99	Gümüşhane	2,92
Bolu	1,93	Osmaniye	2,95	İçel	2,38
Burdur	2,12	Düzce	2,18	Kayseri	2,62
Bursa	1,98	Adana	2,68	Mardin	4,98
Çanakkale	1,68	Adıyaman	3,66	Nevşehir	2,55
Çankırı	2,27	Afyon	2,82	Ordu	2,81
Denizli	2,19	Ağrı	5,49	Sivas	2,76
Erzincan	2,54	Antalya	1,93	Şanlıurfa	4,83
Hakkari	6,69	Çorum	2,66	Yozgat	2,84
Hatay	2,97	Diyarbakır	4,51	Bayburt	3,29
Isparta	2,04	Elazığ	2,52	Kilis	3,54
İstanbul	1,97	Erzurum	3,51	Ankara	1,90
İzmir	1,75	Eskişehir	1,74	Edirne	1,66
Kastamonu	2,18	Kars	3,76	Giresun	2,31
Kırşehir	2,40	Kütahya	2,19	Kırklareli	1,70
Kocaeli	2,13	Samsun	2,55	Konya	3,00
Malatya	2,56	Tekirdağ	1,83	Niğde	2,98
Manisa	2,14	Tunceli	1,90	Rize	2,01
Kahramanmaraş	3,54	Uşak	2,18	Sinop	2,48
Muğla	1,94	Van	6,00	Trabzon	2,10
Muş	4,18	Zonguldak	1,93	Aksaray	2,85
Sakarya	2,23	Batman	5,27	Karaman	2,77

Kaynak: (www.tuik.gov.tr)

**EK 4: İllere Göre Kişi Başına Gelir**

İl Adı	Kişi Başına Gelir (Bin TL)	İl Adı	Kişi Başına Gelir (Bin TL)	İl Adı	Kişi Başına Gelir (Bin TL)
Amasya	513,80	Siirt	289,80	Şırnak	243,80
Aydın	678,20	Tokat	406,10	Ardahan	277,00
Bahkesir	906,50	Kırıkkale	1233,60	İğdır	265,40
Bilecik	685,40	Bartın	555,70	Artvin	825,30
Bingöl	266,60	Yalova	950,00	Gaziantep	531,20
Bitlis	241,90	Karabük	817,00	Gümüşhane	353,10
Bolu	1009,50	Osmaniye	354,80	İçel	2245,50
Burdur	639,60	Düzce	600,20	Kayseri	980,60
Bursa	1628,50	Adana	852,4	Mardin	215,00
Çanakkale	934,00	Adıyaman	289,90	Nevşehir	463,40
Çankırı	469,20	Afyon	533,80	Ordu	479,30
Denizli	792,60	Ağrı	220,00	Sivas	650,70
Erzincan	524,60	Antalya	1496,20	Şanlıurfa	284,60
Hakkari	95,60	Çorum	464,00	Yozgat	356,80
Hatay	1180,80	Diyarbakır	388,10	Bayburt	346,20
Isparta	738,70	Elazığ	597,10	Kilis	298,00
İstanbul	6376,50	Erzurum	540,70	Ankara	6247,70
İzmir	4904,50	Eskişehir	1481,00	Edirne	1199,70
Kastamonu	633,40	Kars	323,30	Giresun	476,10
Kırşehir	548,30	Kütahya	800,30	Kırklareli	1398,20
Kocaeli	15593,10	Samsun	863,20	Konya	743,30
Malatya	509,20	Tekirdağ	2191,90	Niğde	396,90
Manisa	778,20	Tunceli	509,80	Rize	1049,00
Kahramanmaraş	395,90	Uşak	571,90	Sinop	516,70
Muğla	1410,70	Van	321,80	Trabzon	939,90
Muş	190,10	Zonguldak	1200,30	Aksaray	532,80
Sakarya	523,10	Batman	340,70	Karaman	410,90

Kaynak: (www.tuik.gov.tr).

**EK 5: İllere Göre Net Göç**

İl Adı	Net Göç	İl Adı	Net Göç	İl Adı	Net Göç
Amasya	-1654	Siirt	-3322	Şırnak	-4669
Aydın	884	Tokat	-1558	Ardahan	-3258
Balıkesir	1984	Kırıkkale	296	İğdır	-2807
Bilecik	32	Bartın	462	Artvin	-1341
Bingöl	-3740	Yalova	2036	Gaziantep	1950
Bitlis	-4891	Karabük	-731	Gümüşhane	-1763
Bolu	2684	Osmaniye	-1152	İçel	-832
Burdur	-515	Düzce	2706	Kayseri	2244
Bursa	10247	Adana	-424	Mardin	-22012
Çanakkale	610	Adıyaman	-6204	Nevşehir	-1702
Çankırı	8335	Afyon	-2033	Ordu	-961
Denizli	-1587	Ağrı	-10498	Sivas	-5368
Erzincan	777	Antalya	17064	Şanlıurfa	-7964
Hakkari	-3635	Çorum	-9027	Yozgat	-7841
Hatay	-3749	Diyarbakır	-11534	Bayburt	-1319
Isparta	2811	Elazığ	-2101	Kilis	-1015
İstanbul	39481	Erzurum	-8851	Ankara	37079
İzmir	26873	Eskişehir	9.121	Edirne	2273
Kastamonu	-1523	Kars	-6632	Giresun	-2597
Kırşehir	-1239	Kütahya	-2919	Kırklareli	-883
Kocaeli	12033	Samsun	-707	Konya	-4964
Malatya	-2402	Tekirdağ	8589	Niğde	-3190
Manisa	-5481	Tunceli	-2105	Rize	-2147
Kahramanmaraş	-8413	Uşak	-951	Sinop	4
Muğla	1777	Van	-4309	Trabzon	10394
Muş	-10024	Zonguldak	-4443	Aksaray	-2370
Sakarya	3711	Batman	1471	Karaman	-571

Kaynak: (www.tuik.gov.tr)

**EK 6: İllere Göre İşsizlik Oranı**

İl Adı	İşsizlik Oranı %	İl Adı	İşsizlik Oranı %	İl Adı	İşsizlik Oranı %
Amasya	8,20	Siirt	17,90	Şırnak	22,10
Aydın	12,40	Tokat	5,90	Ardahan	3,70
Balıkesir	7,90	Kırıkkale	11,10	Iğdır	6,70
Bilecik	8,20	Bartın	5,20	Artvin	5,70
Bingöl	14,80	Yalova	12,10	Gaziantep	16,80
Bitlis	14,20	Karabük	7,50	Gümüşhane	5,00
Bolu	11,50	Osmaniye	16,30	İçel	13,10
Burdur	6,40	Düzce	10,20	Kayseri	11,10
Bursa	10,80	Adana	20,50	Mardin	17,00
Çanakkale	7,00	Adıyaman	16,50	Nevşehir	9,00
Çankırı	7,70	Afyon	7,70	Ordu	4,90
Denizli	9,40	Ağrı	6,50	Sivas	10,70
Erzincan	6,10	Antalya	9,70	Şanlıurfa	12,80
Hakkari	18,30	Çorum	7,50	Yozgat	12,20
Hatay	17,70	Diyarbakır	15,70	Bayburt	4,70
Isparta	8,10	Elazığ	15,50	Kilis	10,90
İstanbul	11,20	Erzurum	6,20	Ankara	11,80
İzmir	11,80	Eskişehir	9,90	Edirne	14,30
Kastamonu	5,30	Kars	4,10	Giresun	6,30
Kırşehir	11,10	Kütahya	6,50	Kırklareli	11,50
Kocaeli	11,60	Samsun	7,80	Konya	10,70
Malatya	11,50	Tekirdağ	9,10	Niğde	9,30
Manisa	9,80	Tunceli	17,90	Rize	5,20
Kahramanmaraş	12,80	Uşak	9,20	Sinop	6,90
Muğla	10,10	Van	13,00	Trabzon	6,10
Muş	13,20	Zonguldak	6,90	Aksaray	10,00
Sakarya	8,50	Batman	14,30	Karaman	7,70

Kaynak: (www.tuik.gov.tr)

**EK 7: İllere Göre Toplam Nüfus**

<b>İl Adı</b>	<b>Toplam Nüfus</b>	<b>İl Adı</b>	<b>Toplam Nüfus</b>	<b>İl Adı</b>	<b>Toplam Nüfus</b>
Amasya	365231	Siirt	263676	Şırnak	353197
Aydın	950757	Tokat	828027	Ardahan	133756
Balıkesir	1076347	Kırıkkale	383508	İğdır	168634
Bilecik	194326	Bartın	184178	Artvin	191934
Bingöl	253739	Yalova	168593	Gaziantep	1285249
Bitlis	388678	Karabük	225102	Gümüşhane	186953
Bolu	270654	Osmaniye	458782	İçel	1651400
Burdur	256803	Düzce	314266	Kayseri	1060432
Bursa	2125140	Adana	1849478	Mardin	705098
Çanakkale	464975	Adıyaman	623811	Nevşehir	309914
Çankırı	270355	Afyon	812416	Ordu	887765
Denizli	850029	Ağrı	528744	Sivas	755091
Erzincan	316841	Antalya	1719751	Şanlıurfa	1443422
Hakkari	236581	Çorum	597065	Yozgat	682919
Hatay	1253726	Diyarbakır	1362708	Bayburt	97358
Isparta	513681	Elazığ	569616	Kilis	114724
İstanbul	10018735	Erzurum	937389	Ankara	4007860
İzmir	3370866	Eskişehir	706009	Edirne	402606
Kastamonu	375476	Kars	325016	Giresun	523819
Kırşehir	253239	Kütahya	656903	Kırklareli	328461
Kocaeli	1206085	Samsun	1209137	Konya	2192166
Malatya	853658	Tekirdağ	623591	Niğde	348081
Manisa	1260169	Tunceli	93584	Rize	365938
Kahramanmaraş	1002384	Uşak	322313	Sinop	225574
Muğla	715328	Van	877524	Trabzon	975137
Muş	453654	Zonguldak	615599	Aksaray	396084
Sakarya	756168	Batman	456734	Karaman	243210

Kaynak: (www.tuik.gov.tr)

## EK 1: İllere Göre Eğitim Durumu Normalizasyon Verileri

İl Adı	Okuma Yazma Bilmeyen	İlk ve Orta Öğretim Mezunu	Yüksek Öğrenim Mezunu	İl Adı	Okuma Yazma Bilmeyen	İlk ve Orta Öğretim Mezunu	Yüksek Öğrenim Mezunu
Adana	0,2664074	0,1288744	0,1077956	Kocaeli	0,1380461	0,1138567	0,0947745
Adıyaman	0,0640894	0,0203005	0,0133272	Konya	0,028292	0,0295165	0,0238127
Afyon	0,0344282	0,027387	0,0244799	Kütahya	0,065915	0,0320107	0,0320884
Ağrı	0,0588226	0,0090917	0,0048571	Malatya	0,1011921	0,0701026	0,0540848
Amasya	0,0148968	0,0143519	0,01412	Manisa	0,0873679	0,0401239	0,0295519
Ankara	0,3009726	0,3497792	0,5494487	Mardin	0,1118855	0,0195686	0,0102176
Antalya	0,0878106	0,1037465	0,113652	Muğla	0,0144472	0,0255341	0,0337381
Artvin	0,0038018	0,0050344	0,0041704	Muş	0,0293668	0,0046212	0,0020917
Aydın	0,0532888	0,0440647	0,0487569	Nevşehir	0,0117453	0,0097452	0,0083315
Balıkesir	0,0491356	0,0648802	0,0585409	Niğde	0,0108417	0,0099345	0,01068
Bilecik	0,0041966	0,0106785	0,0076336	Ordu	0,0623254	0,0283972	0,0220951
Bingöl	0,0284175	0,0063477	0,0036149	Rize	0,0166881	0,0130918	0,0108512
Bitlis	0,035859	0,0072126	0,0048264	Sakarya	0,0495167	0,050218	0,0362262
Bolu	0,0072544	0,0122958	0,012165	Samsun	0,0875824	0,0619582	0,058205
Burdur	0,0104081	0,0103933	0,0108252	Siirt	0,0471183	0,0067537	0,004497
Bursa	0,1798839	0,1853193	0,1509278	Sinop	0,0063507	0,006077	0,006223
Çanakkale	0,0105542	0,0236026	0,0241151	Sivas	0,0483939	0,0297055	0,0277142
Çankırı	0,0080371	0,0066874	0,0048887	Şanlıurfa	0,0338669	0,0428789	0,0346872
Çorum	0,0422121	0,0253945	0,018683	Tekirdağ	0,0490397	0,0248353	0,0222561
Denizli	0,0506485	0,0494033	0,0474552	Tokat	0,0388872	0,0295931	0,0352427
Diyarbakır	0,2960481	0,0538182	0,0380416	Trabzon	0,0035827	0,0017792	0,0020638
Edirne	0,0228609	0,0202495	0,0213014	Tunceli	0,2331385	0,0349631	0,0180754
Elazığ	0,0722087	0,0254868	0,0249647	Uşak	0,0214506	0,0160703	0,0142614
Erzincan	0,0092648	0,0067593	0,0072493	Van	0,135812	0,0245622	0,0159344
Erzurum	0,0666041	0,0320812	0,0315478	Yozgat	0,0263797	0,0179931	0,0114151
Eskişehir	0,0353342	0,0563954	0,0624424	Zonguldak	0,0215898	0,021209	0,0204705
Gaziantep	0,2238896	0,0883323	0,048114	Aksaray	0,0252067	0,0141253	0,009743
Giresun	0,0301609	0,0166323	0,0163057	Bayburt	0	0,0002892	0
Gümüşhane	0,0003172	0,0020447	0,0017	Karaman	0,0090344	0,01046	0,0071358
Hakkari	0,0330453	0,0051327	0,0031413	Kırkkale	0,0232876	0,0167441	0,0119668
Hatay	0,0976847	0,048825	0,0402496	Batman	0,0903208	0,0151458	0,0086841
Isparta	0,0148876	0,021129	0,0261464	Şırnak	0,0793924	0,0088419	0,0047761
İçel	0,1633669	0,0929878	0,0840145	Bartın	0,0003469	0,0026163	0,003935
İstanbul	1	1	1	Ardahan	0,0007645	0	2,233E-05
İzmir	0,2917032	0,2913964	0,3244681	İğdır	0,0187784	0,0027724	0,0020154
Kahramanmaraş	0,0159693	0,0062737	0,0060248	Yalova	0,0051664	0,0082063	0,0072503
Kars	0,0219047	0,0134575	0,0118785	Karabük	0,0146526	0,01178	0,0099663
Kastamonu	0,0982369	0,0755756	0,0656665	Kilis	0,0106956	0,0031306	0,0013725
Kayseri	0,011184	0,0168779	0,0147611	Osmaniye	0,0478463	0,0217667	0,0162694
Kırklareli	0,0140365	0,010062	0,0091615	Düzce	0,0135116	0,0128326	0,0095262
Kırşehir	0,111372	0,1125019	0,0918426				

**EK 2: İllere Göre Konut Sayısı Normalizasyon Verileri**

<b>İl Adı</b>	<b>Konut Sayısı</b>	<b>İl Adı</b>	<b>Konut Sayısı</b>	<b>İl Adı</b>	<b>Konut Sayısı</b>
Adana	0,1362216	Giresun	0,025873	Samsun	0,0729581
Adıyaman	0,0199737	Gümüşhane	0,0052874	Siirt	0,006548
Afyon	0,0482816	Hakkari	0,0036375	Sinop	0,0095374
Ağrı	0,0097702	Hatay	0,0783501	Sivas	0,0322216
Amasya	0,0200505	Isparta	0,0337793	Şanlıurfa	0,0444804
Ankara	0,331033	İçel	0,127653	Tekirdağ	0,0633862
Antalya	0,1324349	İstanbul	1	Tokat	0,0371081
Artvin	0,0065321	İzmir	0,3346094	Trabzon	0,0552532
Aydın	0,0786355	Kahramanmaraş	0,0468579	Tunceli	0,0014331
Bahkesir	0,098278	Kars	0,0063451	Uşak	0,0220316
Bilecik	0,0105146	Kastamonu	0,0182629	Van	0,0212768
Bingöl	0,0059067	Kayseri	0,0784464	Yozgat	0,0305324
Bitlis	0,0082925	Kırklareli	0,0221776	Zonguldak	0,0398068
Bolu	0,0119846	Kırşehir	0,0140307	Aksaray	0,0217019
Burdur	0,0168222	Kocaeli	0,1016249	Bayburt	0,0009719
Bursa	0,186741	Konya	0,1364299	Karaman	0,014122
Çanakkale	0,0309595	Kütahya	0,0432006	Kırıkkale	0,0221855
Çankırı	0,0104984	Malatya	0,0432458	Batman	0,0140319
Çorum	0,0314529	Manisa	0,0876627	Şırnak	0,0088937
Denizli	0,0667915	Mardin	0,0235123	Bartın	0,0064544
Diyarbakır	0,0568012	Muğla	0,0551082	Ardahan	0
Edirne	0,0256721	Muş	0,0082541	Iğdır	0,0037259
Elazığ	0,0300296	Nevşehir	0,0200174	Yalova	0,0213309
Erzincan	0,0129149	Niğde	0,0211752	Karabük	0,0143711
Erzurum	0,0324169	Ordu	0,0459232	Kilis	0,003639
Eskişehir	0,0589773	Rize	0,020057	Osmaniye	0,0229406
Gaziantep	0,0802181	Sakarya	0,043813	Düzce	0,0086021



**EK 3: İllere Göre Doğurganlık Hızı Normalizasyon Verileri**

<b>İl Adı</b>	<b>Doğurganlık Hızı</b>	<b>İl Adı</b>	<b>Doğurganlık Hızı</b>	<b>İl Adı</b>	<b>Doğurganlık Hızı</b>
Adana	0,1888889	Giresun	0,1203704	Samsun	0,1648148
Adıyaman	0,3703704	Gümüşhane	0,2333333	Siirt	0,812963
Afyon	0,2148148	Hakkari	0,9314815	Sinop	0,1518519
Ağrı	0,7092593	Hatay	0,2425926	Sivas	0,2037037
Amasya	0,1259259	Isparta	0,0703704	Şanlıurfa	0,587037
Ankara	0,0444444	İçel	0,1333333	Tekirdağ	0,0314815
Antalya	0,05	İstanbul	0,0574074	Tokat	0,2592593
Artvin	0,1074074	İzmir	0,0166667	Trabzon	0,0814815
Aydın	0,0851852	Kahramanmaraş	0,3481481	Tunceli	0,0444444
Balıkesir	0,0537037	Kars	0,3888889	Uşak	0,0962963
Bilecik	0,0592593	Kastamonu	0,0962963	Van	0,8037037
Bingöl	0,3518519	Kayseri	0,1777778	Yozgat	0,2185185
Bitlis	0,6240741	Kırklareli	0,0074074	Zonguldak	0,05
Bolu	0,05	Kırşehir	0,137037	Aksaray	0,2203704
Burdur	0,0851852	Kocaeli	0,087037	Bayburt	0,3018519
Bursa	0,0592593	Konya	0,2481481	Karaman	0,2055556
Çanakkale	0,0037037	Kütahya	0,0981481	Kırıkkale	0,1351852
Çankırı	0,112963	Malatya	0,1666667	Batman	0,6685185
Çorum	0,1851852	Manisa	0,0888889	Şırnak	1
Denizli	0,0981481	Mardin	0,6148148	Bartın	0,0833333
Diyarbakır	0,5277778	Muğla	0,0518519	Ardahan	0,2394444
Edirne	0	Muş	0,4666667	Iğdır	0,4653704
Elazığ	0,1592593	Nevşehir	0,1648148	Yalova	0,0507407
Erzincan	0,162963	Niğde	0,2444444	Karabük	0,0614815
Erzurum	0,3425926	Ordu	0,212963	Kilis	0,3474074
Eskişehir	0,0148148	Rize	0,0648148	Osmaniye	0,2388889
Gaziantep	0,4018519	Sakarya	0,1055556	Düzce	0,0955556

**EK 4: İllere Göre Kişi Başına Gelir Normalizasyon Verileri**

İl Adı	Kişi Başına Gelir (Bin TL)	İl Adı	Kişi Başına Gelir (Bin TL)	İl Adı	Kişi Başına Gelir (Bin TL)
Adana	0,0488337	Giresun	0,0245523	Samsun	0,0495306
Adıyaman	0,0125375	Gümüşhane	0,0166156	Siirt	0,0125311
Afyon	0,0282755	Hakkari	0	Sinop	0,0271721
Ağrı	0,0080271	Hatay	0,0700242	Sivas	0,0358187
Amasya	0,026985	Isparta	0,041497	Şanlıurfa	0,0121955
Ankara	0,3969737	İçel	0,1387256	Tekirdağ	0,135267
Antalya	0,0903759	İstanbul	0,4052847	Tokat	0,0200355
Artvin	0,047085	İzmir	0,3103017	Trabzon	0,0544798
Aydın	0,0375932	Kahramanmaraş	0,0193773	Tunceli	0,0267269
Balıkesir	0,0523246	Kars	0,0146927	Uşak	0,030734
Bilecik	0,0380578	Kastamonu	0,0347024	Van	0,0145959
Bingöl	0,011034	Kayseri	0,057106	Yozgat	0,0168543
Bitlis	0,0094402	Kırklareli	0,0840523	Zonguldak	0,0712825
Bolu	0,0589708	Kırşehir	0,0292112	Aksaray	0,028211
Burdur	0,0351024	Kocaeli	1	Bayburt	0,0161704
Bursa	0,0989127	Konya	0,0417938	Karaman	0,0203452
Çanakkale	0,054099	Kütahya	0,0454719	Kırıkkale	0,0734312
Çankırı	0,0241071	Malatya	0,0266882	Batman	0,0158155
Çorum	0,0237716	Manisa	0,0440458	Şırnak	0,0095628
Denizli	0,044975	Mardin	0,0077045	Bartın	0,0296887
Diyarbakır	0,018874	Muğla	0,0848588	Ardahan	0,0117051
Edirne	0,0712437	Muş	0,0060978	Iğdır	0,0109566
Elazığ	0,0323601	Nevşehir	0,0237329	Yalova	0,0551315
Erzincan	0,0276819	Niğde	0,0194418	Karabük	0,0465494
Erzurum	0,0287208	Ordu	0,0247588	Kilis	0,0130602
Eskişehir	0,0893951	Rize	0,0615196	Osmaniye	0,0167253
Gaziantep	0,0281078	Sakarya	0,0275851	Düzce	0,0325601

**EK 5: İllere Göre Net Göç Normalizasyon Verileri**

<b>İl Adı</b>	<b>Net Göç</b>	<b>İl Adı</b>	<b>Net Göç</b>	<b>İl Adı</b>	<b>Net Göç</b>
Adana	0,3510643	Giresun	0,315727	Samsun	0,3464622
Adıyaman	0,2570699	Gümüşhane	0,3292895	Siirt	0,303937
Afyon	0,3248988	Hakkari	0,298847	Sinop	0,3580245
Ağrı	0,1872408	Hatay	0,2969932	Sivas	0,270665
Amasya	0,3310621	Isparta	0,403672	Şanlıurfa	0,2284488
Ankara	0,9609386	İçel	0,3444294	Tekirdağ	0,4976339
Antalya	0,6354544	İstanbul	1	Tokat	0,3326232
Artvin	0,3361521	İzmir	0,7949685	Trabzon	0,5269868
Aydın	0,3723351	Kahramanmaraş	0,2211471	Tunceli	0,3237279
Bahkesir	0,3902233	Kars	0,2501098	Uşak	0,3424943
Bilecik	0,3584798	Kastamonu	0,3331924	Van	0,2878864
Bingöl	0,2971395	Kayseri	0,3944514	Yozgat	0,230449
Bitlis	0,2784219	Kırklareli	0,3436001	Zonguldak	0,2857073
Bolu	0,4016067	Kırşehir	0,3378108	Aksaray	0,3194185
Burdur	0,3495845	Kocaeli	0,5536403	Bayburt	0,3365098
Bursa	0,5245963	Konya	0,2772348	Karaman	0,3486738
Çanakkale	0,3678793	Kütahya	0,3104906	Kırıkkale	0,362773
Çankırı	0,4935033	Malatya	0,3188981	Batman	0,3818809
Çorum	0,2111622	Manisa	0,2688273	Şırnak	0,2820321
Denizli	0,3321516	Mardin	0	Bartın	0,3654725
Diyarbakır	0,1703934	Muğla	0,386857	Ardahan	0,3049778
Edirne	0,394923	Muş	0,194949	Iğdır	0,312312
Elazığ	0,323793	Nevşehir	0,3302815	Yalova	0,3910689
Erzincan	0,370595	Niğde	0,3060836	Karabük	0,3460719
Erzurum	0,2140244	Ordu	0,3423316	Kilis	0,3414535
Eskişehir	0,5062853	Rize	0,3230449	Osmaniye	0,3392256
Gaziantep	0,3896704	Sakarya	0,4183078	Düzce	0,4019645

**EK 6: İllere Göre İşsizlik Oranı Normalizasyon Verileri**

İl Adı	İşsizlik Oranı %	İl Adı	İşsizlik Oranı %	İl Adı	İşsizlik Oranı %
Adana	0,9130435	Giresun	0,1413043	Samsun	0,2228261
Adıyaman	0,6956522	Gümüşhane	0,0706522	Siirt	0,7717391
Afyon	0,2173913	Hakkari	0,7934783	Sinop	0,173913
Ağrı	0,1521739	Hatay	0,7608696	Sivas	0,3804348
Amasya	0,2445652	Isparta	0,2391304	Şanlıurfa	0,4945652
Ankara	0,4402174	İçel	0,5108696	Tekirdağ	0,2934783
Antalya	0,326087	İstanbul	0,4076087	Tokat	0,1195652
Artvin	0,1086957	İzmir	0,4402174	Trabzon	0,1304348
Aydın	0,4728261	Kahramanmaraş	0,4945652	Tunceli	0,7717391
Balıkesir	0,2282609	Kars	0,0217391	Uşak	0,298913
Bilecik	0,2445652	Kastamonu	0,0869565	Van	0,5054348
Bingöl	0,6032609	Kayseri	0,4021739	Yozgat	0,4619565
Bitlis	0,5706522	Kırklareli	0,423913	Zonguldak	0,173913
Bolu	0,423913	Kırşehir	0,4021739	Aksaray	0,3423913
Burdur	0,1467391	Kocaeli	0,4293478	Bayburt	0,0543478
Bursa	0,3858696	Konya	0,3804348	Karaman	0,2173913
Çanakkale	0,1793478	Kütahya	0,1521739	Kırıkkale	0,4021739
Çankırı	0,2173913	Malatya	0,423913	Batman	0,576087
Çorum	0,2065217	Manisa	0,3315217	Şırnak	1
Denizli	0,3097826	Mardin	0,7228261	Bartın	0,0815217
Diyarbakır	0,6521739	Muğla	0,3478261	Ardahan	0
Edirne	0,576087	Muş	0,5163043	Iğdır	0,1630435
Elazığ	0,6413043	Nevşehir	0,2880435	Yalova	0,4565217
Erzincan	0,1304348	Niğde	0,3043478	Karabük	0,2065217
Erzurum	0,1358696	Ordu	0,0652174	Kilis	0,3913043
Eskişehir	0,3369565	Rize	0,0815217	Osmaniye	0,6847826
Gaziantep	0,7119565	Sakarya	0,2608696	Düzce	0,3532609

**EK 7: İllere Göre Toplam Nüfus Normalizasyon Verileri**

<b>İl Adı</b>	<b>Toplam Nüfus</b>	<b>İl Adı</b>	<b>Toplam Nüfus</b>	<b>İl Adı</b>	<b>Toplam Nüfus</b>
Adana	0,1769136	Giresun	0,043348	Samsun	0,1123966
Adıyaman	0,0534226	Gümüşhane	0,0094073	Siirt	0,0171375
Afyon	0,0724253	Hakkari	0,0144075	Sinop	0,0132985
Ağrı	0,0438442	Hatay	0,1168891	Sivas	0,0666496
Amasya	0,0273696	Isparta	0,0423265	Şanlıurfa	0,1360018
Ankara	0,3943795	İçel	0,1569564	Tekirdağ	0,0534004
Antalya	0,163843	İstanbul	1	Tokat	0,0739982
Artvin	0,0099092	İzmir	0,3301997	Trabzon	0,0888201
Aydın	0,0863637	Kahramanmaraş	0,0915654	Tunceli	0
Bahkesir	0,0990174	Kars	0,0233177	Uşak	0,0230454
Bilecik	0,0101502	Kastamonu	0,0284018	Van	0,0789852
Bingöl	0,0161363	Kayseri	0,0974139	Yozgat	0,0593779
Bitlis	0,0297319	Kırklareli	0,0236648	Zonguldak	0,0525952
Bolu	0,0178405	Kırşehir	0,0160859	Aksaray	0,0304781
Burdur	0,016445	Kocaeli	0,1120891	Bayburt	0,0003802
Bursa	0,2046877	Konya	0,2114408	Karaman	0,0150754
Çanakkale	0,0374192	Kütahya	0,0567567	Kırıkkale	0,029211
Çankırı	0,0178104	Malatya	0,0765806	Batman	0,0365889
Çorum	0,0507278	Manisa	0,1175383	Şırnak	0,0261571
Denizli	0,076215	Mardin	0,0616126	Bartın	0,0091277
Diyarbakır	0,1278695	Muğla	0,0626433	Ardahan	0,0040475
Edirne	0,0311352	Muş	0,0362785	İğdır	0,0075616
Elazığ	0,0479622	Nevşehir	0,0217961	Yalova	0,0075575
Erzincan	0,0224941	Niğde	0,0256416	Karabük	0,013251
Erzurum	0,0850168	Ordu	0,080017	Kilis	0,0021299
Eskişehir	0,0617044	Rize	0,0274408	Osmaniye	0,0367952
Gaziantep	0,1200652	Sakarya	0,0667581	Düzce	0,0222346

**EK 8: İllere Göre Sosyal Sınıflar**

İl Adı	Sosyal Sınıf	İl Adı	Sosyal Sınıf	İl Adı	Sosyal Sınıf
Afyon	1	Samsun	1	Bitlis	3
Amasya	1	Sinop	1	Diyarbakır	3
Artvin	1	Tekirdağ	1	Hakkari	3
Balıkesir	1	Tokat	1	Kahramanmaraş	3
Bilecik	1	Trabzon	1	Mardin	3
Bolu	1	Uşak	1	Muş	3
Burdur	1	Zonguldak	1	Niğde	3
Çanakkale	1	Bayburt	1	Siirt	3
Çankırı	1	Karaman	1	Sivas	3
Çorum	1	Kırıkkale	1	Şanlıurfa	3
Denizli	1	Bartın	1	Van	3
Erzincan	1	Ardahan	1	Yozgat	3
Erzurum	1	Yalova	1	Aksaray	3
Eskişehir	1	Karabük	1	Batman	3
Giresun	1	Düzce	1	Şırnak	3
Gümüşhane	1	Adana	2	Iğdır	3
Isparta	1	Elazığ	2	Kilis	3
Kars	1	Gaziantep	2	Osmaniye	3
Kastamonu	1	Hatay	2	Ankara	4
Kırklareli	1	Kayseri	2	Antalya	4
Kütahya	1	Kırşehir	2	Aydın	4
Manisa	1	Konya	2	Bursa	4
Muğla	1	Malatya	2	Edirne	4
Nevşehir	1	Tunceli	2	İçel	4
Ordu	1	Adıyaman	3	İstanbul	4
Rize	1	Ağrı	3	İzmir	4
Sakarya	1	Bingöl	3	Kocaeli	4

## ÖZGEÇMİŞ

Dilek SÜRMEĖĖ, 26.12.1985 yılında İstanbul'da doğdu. İlk öğrenimini, orta öğrenimini ve lise öğrenimini İstanbul'da tamamladı. 2009 yılında Sakarya Üniversitesi İktisadi ve İdari Bilimler Fakültesi İşletme Bölümünden mezun oldu. Aynı yıl içerisinde Sakarya Üniversitesi Sosyal Bilimler Enstitüsü Üretim Yönetimi ve Pazarlama yüksek lisans programında eğitim almaya hak kazandı. Halen aynı programda yüksek lisans eğitimine devam etmektedir.