

T.C.  
MUNZUR ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ



MUNZUR  
ÜNİVERSİTESİ  
2008

**BÖLGESEL DEPREM TAHMİNİ İÇİN TEKRARLI YAPAY SİNİR  
AĞI MODELİNİN GELİŞTİRİLMESİ**

**YÜKSEK LİSANS TEZİ**  
**Şeriban BALKASOĞLU**

**Anabilim Dalı: Elektrik ve Elektronik Mühendisliği**

**DANIŞMAN**  
**Dr. Öğr. Üyesi Özal YILDIRIM**  
**TUNCELİ – 2018**

**T.C.**  
**MUNZUR ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**BÖLGESEL DEPREM TAHMİNİ İÇİN TEKRARLI YAPAY SİNİR  
AĞI MODELİNİN GELİŞTİRİLMESİ**

**YÜKSEK LİSANS TEZİ**  
**Şeriban BALKASOĞLU**  
**(12877049)**

**Anabilim Dalı: Elektrik ve Elektronik Mühendisliği**

**DANIŞMAN**  
**Dr. Öğr. Üyesi Özal YILDIRIM**  
**TUNCELİ – 2018**

**T.C.**  
**MUNZUR ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**BÖLGESEL DEPREM TAHMİNİ İÇİN TEKRARLI YAPAY SİNİR AĞI  
MODELİNİN GELİŞTİRİLMESİ**

**Şeriban BALKASOĞLU**  
**YÜKSEK LİSANS TEZİ**  
**Elektrik ve Elektronik MÜHENDİSLİĞİ ANABİLİM DALI**

Bu tez / / 2018 tarihinde aşağıdaki jüri üyeleri tarafından **oybirliği/oyçokluğu** ile kabul edilmiştir.

**İmza:.....**

Dr. Öğr. Üyesi Özal  
YILDIRIM  
(Munzur Üniversitesi)

**DANIŞMAN**

**İmza:.....**

Doç. Dr. Ayşegül UÇAR  
(Fırat Üniversitesi)

**ÜYE**

**İmza:.....**

Dr. Öğr. Üyesi Bilgin  
ZENGİN  
(Munzur Üniversitesi)

**ÜYE**

Bu tez, Enstitümüz Elektrik ve Elektronik Mühendisliği Anabilim Dalı'nda hazırlanmıştır.

Doç. Dr. Numan YILDIRIM  
Enstitü Müdürü  
İmza ve Mühür

## ÖZET

Bu tez çalışmasında, Tekrarlı Yapay Sinir Ağlarının (TYSA) deprem tahmininde kullanımına yönelik bir uygulamanın gerçekleştirilmesi amaçlanmıştır. Bu amaçla, TYSA'ların en yaygın versiyonu olan Uzun-Kısa Vadeli Hafıza (UKVH) ağlarının bölgesel deprem tahmininde kullanılması sağlanmıştır. UKVH ağlarının ardışık veri kümeleri üzerinde elde ettikleri başarımlar dikkate alınarak bu ağların derin bir versiyonunun deprem verileri üzerinde uygulaması gerçekleştirilmiştir. Bu amaçla, öncelikle beş katmanlı derin bir UKVH ağ modeli tasarlanmıştır. Deneysel çalışmalar için Türkiye bölgesinde Bingöl ili merkez alınarak bu merkez etrafındaki belirlenen yarıçaplı dairesel deprem tahmin alanları oluşturulmuştur. Bu dairesel alan içerisinde zamana bağlı olarak ardışık bir şekilde meydana gelen depremler UKVH ağının eğitiminde kullanılmıştır. Eğitimi tamamlanan ağın aynı bölge üzerinde gelecekte meydana gelecek depremleri tahmin etmesi sağlanmıştır.

Deprem verilerinin elde edilmesinde B.Ü Kandilli Rasathanesi BDTİM Deprem Sorgulama Sisteminden faydalanılmıştır. Merkez noktası etrafında belirlenen yarıçaplar ve yıl aralıkları arasındaki depremler sorgulanarak bir veri tabanı oluşturulmuştur. Dairesel olarak 50km, 100km ve 200km yarıçapında tahmin bölgeleri kullanılmıştır. Bu yarıçaplar arasında meydana gelen 2 ve 9 şiddetleri arasındaki hareketlilikler kullanılmıştır. Elde edilen tahmin sonuçları değerlendirildiğinde, tez kapsamında önerilen uygulamanın deprem tahminlerini kayda değer bir doğrulukta tahmin ettiği gözlemlenmiştir.

**Anahtar Kelimeler:** Makine Öğrenmesi, Yapay Sinir Ağları, Tekrarlı Yapay Sinir Ağları, Uzun-Kısa Vadeli Hafıza Modeli

## **ABSTRACT**

### **Development of Recurrent Neural Network Model for Regional Earthquake Estimation**

In this thesis, it is aimed to implement an application for use of Recurrent Neural Networks (RNN) in earthquake prediction. For this purpose, the most common version of RNN, Long-Short Term Memory (LSTM), has been provided for regional earthquake prediction. The implementation of a deep version of these networks on earthquake data, taking into account the success of the LSTM networks on sequential data sets. For this purpose, a five-layer deep UKVH network model was designed. For experimental studies in Turkey on Bingöl center radius set around that center circular areas were formed in earthquake prediction. Earthquakes that occur in a sequential manner in this circular area depending on time have been used in the training of the LSTM network. It is ensured that the trained network can estimate future earthquakes on the same area.

Earthquake data were obtained from B.U. Kandilli Observatory and Earthquake Research Institute. A database was created by querying the earthquakes between the radius and year intervals determined around the center point. Circular 50km, 100km and 200km radius are used in the estimation regions. The activities between the magnitude of 2 and 9, which took place between these radius, was used. When the obtained estimation results are evaluated, it is observed that the application proposed in the thesis predicts earthquake estimations with a reasonable accuracy.

**Key words:** Machine Learning, Artificial Neural Networks, Recurrent Neural Networks, Long-Short Term Memory

## TEŞEKKÜRLER

Bu çalışmanın gerçekleşmesinde, iki yıl boyunca değerli bilgilerini paylaşan ve zamanını bana ayıran her çıkmaza girdiğimde “hadi gel, bir yolunu buluruz yeter ki çalış ve emek harca” diyen sonsuz minnetlerimi ve saygımı sunduğum değerli danışmanım Dr. Öğr. Üyesi Özal YILDIRIM’a teşekkürü bir borç bilirim.

Tam altı yıl boyunca arkadaşlık ettiğim ve bu tez sürecinde de bir an bile yanımdan ayrılmayan her zaman desteğini esirgemeyen canım arkadaşım Özlem İLHAN’a sonsuz teşekkürlerimi sunarım. Aynı zamanda bu süreçte maddi manevi yanımda olan, “sen yaparsın kızım” diye beni destekleyen aileme, her zorluğumda yanıma koşan ve bana verdiği tavsiyelerle beni aydınlatan Elektrik ve Elektronik Müh. Özcan CÖMERTOĞLU’na ve tez aşamasında programlama dilini öğrenmemde sabırla bana yardım eden arkadaşım Servet TARTAR’a sonsuz teşekkürlerimi sunarım.

**Şeriban Balkasoğlu**  
**TUNCELİ – 2018**

## İÇİNDEKİLER

<b>ÖZET</b> .....	<b>I</b>
<b>ABSTRACT</b> .....	<b>II</b>
<b>TEŞEKKÜRLER</b> .....	<b>III</b>
<b>İÇİNDEKİLER</b> .....	<b>IV</b>
<b>ŞEKİLLER LİSTESİ</b> .....	<b>VI</b>
<b>TABLolar LİSTESİ</b> .....	<b>VIII</b>
<b>KISALTMALAR LİSTESİ</b> .....	<b>IX</b>
<b>1.GİRİŞ</b> .....	<b>1</b>
1.1.Genel.....	1
2.1.Tekrarlı Yapay Sinir Ağlarının Uygulamaları.....	4
<b>2.MATERYAL ve METOTLAR</b> .....	<b>6</b>
2.1.Makine Öğrenmesi .....	6
2.1.1.Makine Öğrenmesinin Gelişimi.....	6
2.1.1.1.Denetimli Öğrenme .....	7
2.1.1.2.Denetimsiz Öğrenme .....	7
2.1.1.3.Yarı Denetimli Öğrenme .....	8
2.1.1.4.Öğrenmek İçin Öğrenme .....	8
2.1.2.Makine Öğrenmesinin Uygulamaları .....	8
2.1.2.1.Bilim ve Mühendislik Uygulamaları .....	8
2.1.2.2.Sağlık Uygulamaları .....	9
2.1.2.3.Bankacılık ve Finansal Uygulamaları.....	9
2.1.2.4.Eğitim Uygulamaları .....	9
2.1.2.5.Web Uygulamaları .....	9
2.1.2.6.Ses ve Görüntü Tanıma Uygulamaları .....	10
2.1.3. Makine Öğrenmesi Teknikleri.....	10
2.1.3.1. Sınıflandırma .....	10
2.1.3.2. Kara Ağacı .....	11
2.1.3.3. Destek Vektör Makinesi .....	12
2.1.3.4. Kümeleme.....	12
2.2. Yapay Sinir Ağları.....	13
2.2.1. Yapay Sinir Ağlarının Yapısı .....	13
2.2.2. Yapay Sinir Ağların Üstünlükleri.....	15
2.2.3. Yapay Sinir Ağlarının Dezavantajları .....	15
2.2.4. Yapay Sinir Ağlarda Sınıflandırma .....	16
2.2.4.1.Yapay Sinir Ağların Yapılarına Göre Sınıflandırılması .....	16
2.2.4.1.1. İleri Beslemeli Ağlar .....	16
2.2.4.1.2. Geri Beslemeli Ağlar .....	17
2.2.4.2. Yapay Sinir Ağlarında Temel Öğrenme .....	19
2.2.4.2.1. Hebb Kurallı .....	19
2.2.4.2.2. Hopfield Kurallı.....	19
2.2.4.2.3. Delta Kurallı .....	19
2.2.4.2.4. Kohonen Kurallı .....	20
2.2.4.3.Yapay Sinir Ağların Öğrenme Algoritmalarına Göre Sınıflandırılması .....	20
2.2.4.3.1. Danışmanlı Öğrenme .....	20
2.2.4.3.2. Danışmansız Öğrenme.....	20
2.2.4.3.3. Pekiştirmeli Öğrenme .....	21
2.2.5. Geri Yayılım Algoritması.....	21

2.2.6. Yapay Sinir Ağların Yapıları .....	22
2.2.6.1. Adaline .....	22
2.2.6.2. Çok Katmanlı Ağlar.....	23
2.2.6.3. Vektör Kuantalamalı Öğrenme.....	23
2.2.6.4. ART Ağları .....	23
2.2.7. Eğitim İnişi .....	24
2.2.7.1. Eğitim İnişi Prosedürü .....	25
2.2.7.2. Makine Öğrenimi için Stokastik Eğitim İnişi.....	26
2.2.8. Boltzman Makineleri .....	26
2.2.9. Yapay Sinir Ağlarının Uygulamaları.....	27
2.3. Tekrarlı Yapay Sinir Ağları.....	28
2.3.1. Giriş .....	28
2.3.2. Tekrarlı Yapay Sinir Ağları .....	28
2.3.3. İlk Tekrar Eden Ağların Tasarımları .....	30
2.3.4. Tekrarlı Ağların Eğitimi .....	32
2.3.5. Uzun Kısa Vadeli Hafıza Ağları.....	35
2.3.5.1. Uzun ve Vadeli Bağımlılık Sorunu .....	36
2.3.5.2. Uzun Kısa Vadeli Hafıza Ağların Yapısı .....	39
2.3.5.3. Uzun Kısa Vadeli Hafıza Ağlarda Adımlar .....	40
2.3.6. Çift Yönlü Tekrarlı Yapay Sinir Ağları.....	43
<b>3. BULGULAR .....</b>	<b>45</b>
3.1. Derin Uzun-Kısa Vadeli Hafıza Ağ Modelini Kullanarak Deprem Tahmini .....	45
3.1.1. Deprem Verileri .....	45
3.1.2. Önerilen Derin Uzun-Kısa Vadeli Hafıza ağ Modeli .....	47
3.1.3. Deneysel Sonuçlar .....	49
3.1.3.1 Deneysel Çalışma 1: 50km Yarıçap .....	49
3.1.3.2. Deneysel Çalışma 2: 100km Yarıçap .....	56
3.1.3.3. Deneysel Çalışma 3: 200km Yarıçap .....	61
<b>4. TARTIŞMA.....</b>	<b>67</b>
<b>5. SONUÇLAR.....</b>	<b>68</b>
<b>KAYNAKLAR .....</b>	<b>69</b>
<b>ÖZGEÇMİŞ.....</b>	<b>74</b>

## SEKİLLER LİSTESİ

### Sayfa No

Şekil 2.1. Sınıflandırma Süreci .....	11
Şekil 2.2 Basit bir sinir ağ yapısı. ....	14
Şekil 2.3. Çok katmanlı ileri beslemeli ağ yapısı .....	17
Şekil 2.4. Çok katmanlı geri beslemeli ağ yapısı.....	18
Şekil 2.5. ART ağının yapısı.....	24
Şekil 2.6. Basit bir TYSA ağ modeli .....	29
Şekil 2.7. Şekil 2.6'nın TYSA'ların zaman kademeleri boyunca açılımı.....	30
Şekil 2.8. Jordan tarafından önerilen TYSA modeli.....	31
Şekil 2.9. Elman tarafından önerilen TYSA modeli.....	33
Şekil 2.10. Bir giriş, çıkış ve gizli katmana sahip TYSA modeli .....	34
Şekil 2.11. Şekil 2.10'da gösterilen ağı kullanarak Graves'den uyarlanan bir TYSA modeli .....	34
Şekil 2.12. TYSA yapısı .....	36
Şekil 2.13. TYSA modelinin zamana bağlı olarak açılımı .....	36
Şekil 2.14. Kısa mesafelerde tahmin yapan TYSA'nın açılımı .....	37
Şekil 2.15. Uzun mesafelerde tahmin yapan TYSA'nın açılımı .....	37
Şekil 2.16. Standart bir TYSA'ları tek bir katman içerir.....	38
Şekil 2.17. Dört katmanlı UKVH ağ modeli .....	38
Şekil 2.18. UKVH ağ hücresi .....	39
Şekil 2.19. UKVH ağ modelinin kapı gösterimi.....	40
Şekil 2.20. UKVH ağ hücresi .....	41
Şekil 2.21 UKVH ağ hücresi .....	41
Şekil 2.22. UKVH ağ hücresi .....	42
Şekil 2.23. UKVH ağ hücresi .....	43
Şekil 3.1. Türkiye deprem haritası.....	46
Şekil 3.2. B.Ü. Kandilli Rasathanesi BDTİM Deprem Sorgulama Sistemi .....	46
Şekil 3.3. 1950 ve 1970 yılları arası 50km yarıçap için meydana gelen deprem haritası.....	47
Şekil 3.4. 1950 ve 1970 yılları arası 50km yarıçap için meydana gelen deprem	

listesi .....	47
Şekil 3.5. Deprem tahminleri için kullanılan derin UKVH ağ modeli .....	48
Şekil 3.6. 2000 ve 2010 arası 50km yarıçap için sorgu sonucu.....	50
Şekil 3.7. 2000 ve 2017 yılları arası 50km yarıçap için farklı SL olan deprem tahminleri.....	52
Şekil 3.8. 2000 ve 2017 yılları arası 50km yarıçap için SL=50 olan grafik .....	52
Şekil 3.9. 1950 ve 2017 yılları arası 50km yarıçap için farklı SL olan deprem tahminleri.....	54
Şekil 3.10. 1950 ve 2017 yılları arası 50km yarıçap için SL=5 olan grafik .....	55
Şekil 3.11. 2000 ve 2010 arası 100km yarıçap için sorgu sonucu.....	56
Şekil 3.12. 2000 ve 2017 yılları arası 100km yarıçap için farklı SL olan deprem tahminleri.....	58
Şekil 3.13. 2000 ve 2017 yılları arası 100km yarıçap için SL=5 olan grafik .....	58
Şekil 3.14. 1950 ve 2017 yılları arası 100km yarıçap için farklı SL olan deprem tahminleri.....	60
Şekil 3.15. 1950 ve 2017 yılları arası 100km yarıçap için SL=5 olan grafik .....	60
Şekil 3.16. 2000 ve 2005 arası 200km yarıçap için sorgu sonucu.....	62
Şekil 3.17. 2000 ve 2017 yılları arası 200km yarıçap için farklı SL olan deprem tahminleri.....	64
Şekil 3.18. 2000 ve 2017 yılları arası 200km yarıçap için SL=50 olan grafik .....	64
Şekil 3.19. 1950 ve 2017 yılları arası 200km yarıçap için farklı SL olan deprem tahminleri.....	66
Şekil 3.20. 1950 ve 2017 yılları arası 200km yarıçap için SL=25 olan grafik .....	66

## TABLolar LİSTESİ

### Sayfa No

Tablo 3.1. 50km yarıçap için yıllara göre veri sayıları.....	50
Tablo 3.2. 2000-2017 yılları arası 50km yarıçap için her SL için kayıplar.....	51
Tablo 3.3. 2000-2017 yılları arasında SL=50 için gerçek ve tahmin edilen ilk 30 deprem .....	53
Tablo 3.4 1950-2017 yılları arası 50km yarıçap için her SL için kayıplar.....	53
Tablo 3.5. 1950-2017 yılları arasında SL=5 için gerçek ve tahmin edilen ilk 30 deprem .....	55
Tablo 3.6. 100km yarıçap için yıllara göre veri sayıları.....	57
Tablo 3.7. 2000-2017 yılları arası 100km yarıçap için her SL için kayıplar.....	57
Tablo 3.8. 2000-2017 yılları arasında SL=5 için gerçek ve tahmin edilen ilk 30 deprem .....	59
Tablo 3.9 1950-2017 yılları arası 100km yarıçap için her SL için kayıplar.....	59
Tablo 3.10. 1950-2017 yılları arasında SL=5 için gerçek ve tahmin edilen ilk 30 deprem .....	61
Tablo 3.11. 200km yarıçap için yıllara göre veri sayıları.....	62
Tablo 3.12. 2000-2017 yılları arası 200km yarıçap için her SL için kayıplar.....	63
Tablo 3.13. 1950-2017 yılları arası 200km yarıçap için her SL için kayıplar.....	65

## KISALTMALAR

<b>YSA</b>	: Yapay Sinir Ağları
<b>TYSA</b>	: Tekrarlı Yapay Sinir Ağları
<b>LVQ</b>	: Vektör Kuantamalı Öğrenme
<b>DVM</b>	: Destek Vektör Makinesi
<b>MLP</b>	: Çok Katmanlı Ağlar
<b>ART</b>	: Adaptif Rezonans Theory
<b>UDH</b>	: Uzun Dönemli Hafıza
<b>KDH</b>	: Kısa Dönemli Hafıza
<b>TBPTT</b>	: Kesintisiz Geri Yayılım Süresi
<b>GPU</b>	: Grafik İşlem Birimi
<b>KAF</b>	: Kuzey Anadolu Fay Hattı
<b>DAF</b>	: Doğu Anadolu Fay Hattı
<b>GAF</b>	: Güney Anadolu Fay Hattı
<b>SL</b>	: Sequence Length
<b>SGD</b>	: Stochastic Gradient Descent
<b>ÇYTSA</b>	: Çift Yönlü Tekrarlı Sinir Ağları
<b>BPTT</b>	: Gerilim Yayılım Süresi
<b>TBPTT</b>	: Kesintisiz Gerilim Yayılım Süresi
<b>SOM</b>	: Self Organizing Map
<b>KDH</b>	: Kısa Dönemli Hafıza
<b>UKVH</b>	: Uzun Kısa Vadeli Hafıza

# 1. GİRİŞ

## 1.1. Genel

Türkiye’de Kuzey Anadolu Fay Hattı (KAF), Doğu Anadolu Fay Hattı (DAF) ve Güney Anadolu Fay Hattı (GAF) olmak üzere üç deprem fay hattına sahiptir. Bu fay hatları üzerinde birçok şiddetli deprem hareketliliği meydana gelmiştir. Bu depremler sonucunda binlerce insanın hayatı olumsuz yönde etkilenmiştir. Bunlar ölümler, yaralanmalar ayrıca binlerce binanın yıkımı ve hasarı şeklinde olmuştur. Bu büyük depremlerden birkaçı aşağıda belirtilmiştir:

1 Mayıs 2003’te yaşanan Bingöl depreminin yüzey dalga büyüklüğünün (Mc) 6,4 olduğu tespit edilmiştir. Bu deprem, 176 kişinin ölümüne, 521 kişinin yaralanmasına, 570 konuttun yıkımına ve 6000 konutun hasar görmesine neden olmuştur. Aynı zamanda bu ilimizde 22 Mayıs 1971’de de gerçekleşen depremin dalga büyüklüğü 6,8 olarak ölçülmüştür. Bu büyüklüğün doğurdu sonuçlara bakacak olursak; 878 kişinin hayatını kaybetmiş, 700 kişi hafif ve ağır yaralanmış ve 9111 tane bina yıkılmış veya hasar görmüştür.

17 Ağustos 1999’da Gölcük’te meydana gelen depremin büyüklüğü 7,8 olarak ölçülmüştür. Felakette resmi rakamlara göre 17 bin 480 kişi yaşamını yitirmiştir. On binlerce kişinin yaralandığı bu depremde 73 bin 342 konut hasar görülmüştür.

23 Ekim 2011’de Van’da meydana gelen 7,2 büyüklüğündeki deprem Kuzey Irak’tan bile hissedilmiştir. Bu deprem sonucunda, 644 kişinin hayatını kaybettiği, 1966 kişinin yaralandığı tespit edilmiştir (URL1, 2017).

Yakın tarihte gerçekleşen bu doğal afetlerin ne tür kayıplar doğurduğuna ve doğuracağına tanıklık etmişizdir. Bu doğal afetler daha önceden bir uyarı vermediklerinden yaşanan kayıplar daha da artmaktadır. Bu nedenle deprem tahmini konusu her zaman ilgi çekici bir konu olmuştur. Bilim adamları bu konu üzerine birçok çalışma yapmıştır.

Negarestani ve ark. (2002), Yapay Sinir Ağları (YSA) tabanlı bir model kullanarak radon konsantrasyonunun çevresel parametreler ile ilişkisini incelemiştir. Analizler sonucunda bu yaklaşımın radondaki anormal değişimlerin deprem gibi etkilere neden olduğu görülmüştür.

Asencio-Cortés ve ark. (2015) çalışmalarında bir denetimli öğrenme sınıflandırıcısı ile temel bileşen analizini (Principal Component Analysis-PCA) birlikte kullanarak bir deprem tahmin yöntemi önermişlerdir.

Lin ve ark. (2009), çalışmasında Tayvan depreminin 1999 Chi-Chi`den öncesi ve sonrası Alishan, Tayvan bölgesinde otoyol yamaçlarının başarısızlık özelliklerin ve potansiyelini değerlendirmek için ampirik bir model oluşturmuştur. Çalışmanın temelinde Alishan bölgesindeki dört otoyoldan 955 eğim kaydından oluşan bir veri tabanı oluşturulmuştur. Bu veri tabanından öğrenmek için YSA kullanılmıştır. Geliştirilen YSA modeli daha sonra Chi-Chi depreminin Alishan bölgesindeki eğim arıza karakteristikleri üzerindeki etkisini incelemek için kullanılmıştır. Birkaç faktörün (değişkenler) etki derecelerinde önemli değişiklikler saptanmış ve bu değişikliklerin muhtemel sebepleri tartışılmıştır. Geliştirilen YSA modelleri, Chi-Chi depreminden önceki ve sonraki eğim arıza özelliklerini araştırmak için bir araç olarak kullanılmıştır.

Sikder ve ark. (2009), deprem öncesi faktörlerin karakterize edilmesi için bir makine öğrenme yaklaşımı sunmuştur. Sismik olayların karakteristik asimetrik dağılımı ve örnekleme sınırlamaları geleneksel istatistiksel öngörme tekniklerini uygulamayı zorlaştırmıştır. Kaba küme teorisi ve karar ağacı (C4.5 algoritması) gibi induktif makine öğrenme teknikleri, radon konsantrasyonunun ve zaman çevresel değişkenlerin uzay-zaman dağılımı gibi ön tanımlı tanımlayıcıları içeren anlamlı karar kurallarına göre sismik aktivitenin bilgi sunum yapısını geliştirmeye izin vermiştir. Her iki teknik de, bilgi teorik önlemlerini, örneğin, indirgemelerde entropi ve oluşma frekansı kullanarak, ön-öncesi değişkenleri ve sıralamayı belirlemiştir.

Yıldırım (2010), “Veri Madenciliği Yöntemiyle Deprem Analizi” çalışmasında deprem verileri üzerinden bazı veri madenciliği tekniklerini uygulanarak ve bu verilere ait birtakım istatistiksel çalışmaları gerçekleştirilmiştir. Deprem verileri üzerine uygulanan bu yöntemler yardımı ile bölgelerin deprem karakteristiklerini tespit edilmesi ve gelecek deprem hareketlerini hakkında çeşitli tahminler yapılmıştır.

Nasser (2012), “MATLAB Ortamında Kesme Dalgaların Ayrılmanması Ölçümleri Kullanarak Deprem Tahmini” bu tez çalışmasında MATLAB ortamında kesme dalgası ayrılmanması parametrelerini kullanarak deprem tahmini yapılmıştır. Bu çalışmada, kesme dalgalarının ayrılmanması ölçümleri kullanılarak deprem tahminleri için iki program yazılmıştır. Birinci program, kesme dalgalarının ayrılmanmasının parametrelerini ölçmek

için kullanılmıştır. İkinci program ise önceki programın bulduğu ayrımlanma parametrelerine dayanarak, deprem tahmin etmek için kullanılmıştır.

Bilen ve ark. (2015), “Sismik Darbelerin Sınıflandırarak Deprem Tehlikesinin Tahmin Edilmesi” adlı makalesinde Polonya maden ocaklarından elde edilen sismik darbe verileri KNN (K En Yakın Komşular), DVM (Destek Vektör Makinesi), YSA algoritmaları kullanılarak sınıflandırılmıştır. Sınıflandırma hızının ve doğruluğunun artırılması için PCA (Principal Component Analysis: Temel Bileşen Analizi) gibi öznelik seçme algoritmalarından yararlanılarak verinin boyutu azaltılmıştır. Sınıflandırma sonucunda, deprem tehlikesi %94,12’ye kadar çıkan bir başarı oranı ile doğru tahmin edilmiştir ve bulgular paylaşılmıştır.

Depremleri tahmin etmeye yönelik birçok çalışmalar yapılmıştır. Bu tez çalışmasında, makine öğrenmesi alanında son zamanlarda popülerleşen ve kullanımı giderek artan Tekrarlı Yapay Sinir Ağlarının (TYSA) bir uygulamasının gerçekleştirilmesi amaçlanmıştır.

Tekrarlayan yapay sinir ağları (TYSA), birimler arasındaki bağlantıların yönlendirilmiş bir döngü oluşturduğu yapay sinir ağı sınıfıdır. Doğrusal olmayan aktivasyonlara sahip sonlu TYSA’lar neredeyse keyfi hesaplama yapabilecek kapsamlı bir model ailesidir. Bilenen bir sonuç, sigmoid aktivasyon işlevlerine sahip nihai boyutlu tekrarlayan bir sinir ağı, evrensel bir Turing makinesini taklit edebilir. TYSA’lar keyfi hesaplama yapabilecek kabiliyeti ve ifade güçlerine sahiptirler. TYSA; ağırlık kaybı (weight loss), terk etme(dropout) ve bağımsızlık(dependencies) derecelerinin sınırlandırılması gibi aşırı uyumu (overfitting) önlemeye yardımcı olan standart parametrelerle düzenlenebilir. TYSA’lar, bitişik zaman adımları boyunca uzanan kenarların eklenmesiyle güçlendirilmiş ileri beslemeli sinir ağlarıdır ve bu da modele zaman kavramı getirir. TYSA’nın gelişimi ile birçok problemler çözüm bulmuştur. Örneğin C programında seçilen bir kayıp fonksiyonunu en aza indirmek için bir C programın gradyanını hesaplamak için genel bir yol yoktur. Dahası, herhangi bir sonlu veri kümesi göz önüne alındığında, veri kümesini aşan, istenilen eğitim çıktısını üreten, ancak örnek test etmek için başarısız olan sayısız program mevcuttur (Lipton ve ark., 2015).

Bu tez çalışmasında, TYSA’ların ardışık veri kümeleri üzerindeki başarımı dikkate alınarak deprem verileri üzerinde kullanılması hedeflenmektedir. Bu amaçla, TYSA yapısı ile belli bölgelerdeki deprem karakteristiklerini ve gelecekteki hareketlerini tahmin etmesi sağlanacaktır. Tekrarlı sinir ağları bitişik zaman adımları boyunca uzanan kenarların

eklenmesiyle güçlendirilmiş ileri beslemeli sinir ağıdır bu da normal yapay sinir ağlarına zaman kavramı getirir. Böylece deprem veri tabanına tekrarlı sinir ağlarıyla zaman kavramını uygulayıp bir sonraki hareketliliğin tahmin edilebilmesi amaçlanmaktadır. Bu çalışma, belirlenecek olan bir bölgeye ait deprem verileri üzerinde bir tahmin algoritması tasarlanmayı hedeflemektedir. Bu bölgede oluşan geçmiş deprem hareketlilikleri üzerinde eğitilecek algoritmanın gelecekte olması muhtemel deprem hareketlerini tahmin edebilecek yapıya kavuşturulması amaçlanmıştır.

## 1.2. TYSA Uygulamaları

TYSA'lar günümüzde en popüler makine öğrenme konularından birisidir. Bu nedenle birçok uygulama üzerine kurulmaya çalışılmıştır. Bu alandaki bazı uygulamalar şu şekildedir;

Connor ve ark. (1994), TYSA'lar ile zaman serileri tahmini çalışmasını yapmıştır. Bu çalışmada gürbüz bir öğrenme algoritması önerilmiş ve tekrar eden sinir ağlarına uygulanmıştır. Bu algoritma, verilerden aykırı değerlerin filtrelenmesine ve daha sonra filtrelenmiş verilerden parametrelerin tahmin edilmesine dayanmaktadır. Filtreleme hem hedef fonksiyondan hem de sinir ağı girişlerinden aykırı değerleri kaldırmıştır. Filtreleme bazı dışa vurucuların ne tamamen reddedilmiş ne de kabul edilmiş olması bakımından yumuşaktır. Gürbüz yinelenen ağlara duyulan gereksinimi göstermek için, yapay verideki ve Puget Güç Elektrik Talebi zaman serileri üzerindeki en düşük kareler tahmini ile yinelenen ağların öngörücü kabiliyetini karşılaştırmışlardır.

Funahashi ve Nakamura 'nın (1992), sürekli zaman tekrarlayan sinir ağları ile dinamik sistemlerin yaklaştırılması adında bir çalışma yapmıştır. Bu çalışmasında başlangıç koşulu verilen bir n-boyutlu dinamik sistemin herhangi bir sonlu yörüngesinin, n çıkış birimi, bazı gizli birimler ve uygun olan sürekli bir zaman tekrarlayan sinir ağının çıkış birimlerinin iç durumu tarafından yaklaşık olarak gerçekleştirilebileceğini kanıtlamıştır. Kanıtın esas fikri, n-boyutlu dinamik sistemi tekrar eden bir sinir ağını tanımlayan daha yüksek boyutlu bir düzeye yerleştirmektir. Sonuç olarak, sürekli bir eğrisinin tekrarlayan bir sinir ağı çıktısıyla yaklaştırılabileceğini de gösterilmiştir.

Mikolov ve ark. (2010), TYSA dil modelinin uzantıları çalışmalarını yapmışlardır. Çalışmalarında, özgün TYSA çalışma dili modelinin birkaç değişikliğini sunmuşlardır. Bu

modelin doğruluk açısından birçok rekabetçi dil modelleme tekniğinden belirgin şekilde daha iyi performans gösterdiği raporlanmıştır.

Saha ve Raghava (2006), TYSA kullanarak bir antijendeki sürekli B-hücreli epitoplarnın tahmini çalışmasını yapmışlardır. Bu çalışmada, standart ilerleme ve TYSA antijenik bir dizide B-hücreli epitoplarnın öngörülmesi için kullanılmıştır.

Kamijo ve Tanigawa (1990), TYSA yaklaşımı ile hisse senedi fiyat örüntü tanıma üzerine bir çalışma yapmışlardır. Bu çalışmada, stok modellerinin tanımlanmasına TYSA modeli uygulanmış ve hisselerin değerlendirilmesi için bir yöntem sunulmuştur.

Barbounis ve ark. (2006), yerel TYSA modellerini kullanarak uzun vadeli rüzgâr hızı ve güç tahmini adında çalışmasını yapmışlardır. Yapmış oldukları çalışmada meteorolojik bilgilere dayanan uzun vadeli rüzgâr hızı ve güç tahmini problemi ile ilgilidir. Yunan adası Girit'te bir rüzgâr parkı için 72 saat öncesine kadar saatlik tahminler üretilmiştir.

Cho ve ark. (2014), istatistiksel makine çevirisi için TYSA kodlayıcı-kod çözücü kullanan öbek tabakalarını öğrenme adında bir çalışma yapmışlardır. Bu çalışmada, iki TYSA ağından oluşan kodlayıcı ve kod çözücü olarak adlandırılan yeni bir sinir ağı modellenmiştir. Bir TYSA, bir dizi sembolünü sabit uzunlukta bir vektör gösterimine kodlarken, diğerinde sembollerin başka bir dizi dizilimini çözmüştür. Önerilen modelin kodlayıcı ve kod çözücüsü, bir kaynak sırası ile verilen bir hedef dizinin koşullu olasılığını en yükseğe çıkarmak için birlikte eğitilmiştir. İstatistiksel bir makine çevirisi sisteminin performansı, mevcut logaritmik-doğrusal modelde ilave bir özellik olarak TYSA kodlayıcı-kod çözücü tarafından hesaplanan cümle çiftlerinin koşullu olasılıklarını kullanarak ampirik olarak iyileştiğini bulmuştur. Nitel olarak, önerilen modelin dilsel cümleciklerin semantik dizimsel olarak anlamlı bir temsilini öğrendiğini ortaya koymuşlardır.

Graves (2014), TYSA'lar ile seriler oluşturma konusunda bir çalışma yapmışlardır. Bu çalışmasında bir defada tek bir veri noktasını tahmin ederek, uzun vadeli hafızalı tekrarlayan sinir ağlarının uzun menzilli karmaşık dizileri üretmek için nasıl kullanılabileceği gösterilmiştir.

## 2. MATERYAL VE METOTLAR

### 2.1. Makine Öğrenmesi

Yapay zekânın evrelerinden olan makine öğrenmesi, bilgisayarda bir problemi özel olarak tasarlamak yerine, bu sorunun çözümünü kendi başına öğrenebilen bir sistemdir. Makine öğrenmesi, mazide yapılan örnekleri değerlendirerek ve verileri kullanarak bir olayı önceden kestirebilir ve yeni bir model ortaya koyabilir. Makine öğrenmesi, bilgisayar kullanıcılarının geri beslemeli öğrenme biçiminin taklit edilmiş halidir.

Makine öğrenmesi, veri setlerini kullanarak uygulamaları ve algoritmaları eğitirler. Eğitim verileri kullanarak anlaşılır biçimde neticeler elde edilmektedir. Verilerde bulunabilecek küçük ayrıntılara erişmek için makine öğrenmesi uygun bir yol olmaktadır. Örneğin sinema severler tarafından bir filmin beğenilmesine ya da beğenilmemesine sebep olan durumlar tam olarak belirlenemeyebilir. Bu sebeple büyük bir içerik sağlayıcının kullanıcıların verdiği cevapların bir örnekleme dayanarak, hazırlanan bir film derecelendirme tahmini algoritması çalışmasında makine öğrenmesi teknikleri oldukça başarılı olmuştur. Başka bir çalışmada ise bir film yapım şirketi tarafından hazırlanan web sayfası üzerinden yaptığı özgeçmiş taraması ile makine uygulamasında içerik değerlendirme için el ile yapılacak düzenlemeler kaçınılmazdır. Makine öğrenmesi üç aşamada gerçekleşir (Kaya, 2015).

- Verilerin hazırlanması,
- Öğrenme metotlarının uygulanması,
- Öğrenme performansının değerlendirmesi (Leskovec ve ark., 2014).

#### 2.1.1. Makine Öğrenmesinin Gelişimi

Makine öğrenmesi, veri setinden yola çıkarak veri madenciliği yöntemlerinin değerlendirilip tatbik edilecek makine öğrenmesi tekniğine elverişli organizasyonlar oluşturulmasıdır. Makine öğrenmesi teknik istatistiksel yaklaşımlardan türetilmiştir ve benzer yapı üzerinden meydana gelmiştir. Yeni tasarlanan algoritmalar geliştirdiğinde başarımları değerlendirilir ve uygulanan teknikler ile başarımları karşılaştırılır. Makine öğrenmesinin varoluşundan beri yeni tekniklerin geliştirilmesi devam etmektedir.

Bu tekniklerin algoritma yapılarına göre sınıflandırılması aşağıdaki gibi yapılmıştır (Kaya, 2015).

- Denetimli Öğrenme,
- Denetimsiz Öğrenme,
- Yarı-Denetimli Öğrenme,
- Takviyeli Öğrenme,
- Uyum Sağlama ile Öğrenme,
- Öğrenme ile Öğrenme.

#### **2.1.1.1. Denetimli Öğrenme**

Makine öğrenmesi tekniği olarak bilinen denetimli öğrenme, eğitim verilerini kullanarak bir yaklaşım sunmaktadır. Pozitif ve negatif örneklerden meydana gelen bir sınıf vasıtasıyla algoritmanın girdileri ve çıktıları oluşturulur. Tahmin işlemi yapılırken, eğri uydurma veya sınıflama metotları kullanılmaktadır. Denetimli öğrenme,

- Eğitim verisinin çeşitlerini belirleme,
- Eğitim verisinin elde edilmesi,
- Öğrenme fonksiyonunun açıklanması,
- Öğrenme algoritmasının tasarlanması,
- Öğrenme algoritmasının başarımlarının ölçülmesi için belli aşamalardan oluşur.

Örneğin; bir denetimli öğrenme çalışmasında, sahte kredi kartı kullanımı tanımlayan bir model, bilinen sahte ve uygulanan ödemelerin veri noktalarıyla etiketlenmiş bir veri kümesinden eğitilir (Kaya, 2015).

#### **2.1.1.2. Denetimsiz Öğrenme**

Denetimli öğrenmede doğru sonuçlar bir danışman aracılığıyla öğretilmektedir. Denetimsiz öğrenmedeyse herhangi bir danışman yoktur. Yalnızca girdi veri setlerini modele uygulayarak, çıktı veri setini tasarlanan modele göstermeden öğrenme işlemi gerçekleştirilmektedir. Hedef girdi verilerinin organize edilmesi sağlanmaktadır. Makine öğrenmesi tekniği olan denetimsiz öğrenme, gözlemlere dayalı bir öğrenme şeklidir. Denetimsiz öğrenme tekniği, öncelikle olasılık tabanlı kümeleme işlemlerinde kullanılır.

Kümelemeyle desenlerin yoğunluğu irdelenerek, çeşitli gruplardaki benzer desenler sınıflandırılmaktadır (Alpaydın, 2004).

### **2.1.1.3. Yarı-Denetimli Öğrenme**

Yarı-Denetimli öğrenme, uygun sınıflama ya da sınıflama fonksiyonu sağlamak için sınıflandırılmış ve sınıflandırılmamış veriler kullanılmaktadır. Sınıflandırılmış örneklemin büyüklüğü, sınıflandırılmamış örnekleme göre daha azdır. Bütün etiketlenmemiş verileri de etiketlemek sıkıntılıdır. Bu nedenden dolayı yarı-denetimli öğrenme elverişli bir metottur (Chapelle ve Schölkopf, 2006).

### **2.1.1.4. Öğrenmek için Öğrenme**

Geçmiş deneyimlerden yola çıkarak, kendi tüme-varımsal eğilimleriyle öğrenen bir algoritma türüdür (Alpaydın, 2004).

## **2.1.2. Makine Öğrenmesi Uygulamaları**

Geçmişten günümüze veri depolarının ve bilgiye olan açlığın giderek artması insanları bu konuda çalışmaya yönlendirmiştir. Ticaret, bilim ve mühendislik, eğitim, metin ve dil işleme, oyun, bankacılık ve finans, web, sağlık, güvenlik, bilgisayar görmesi gibi yapılan diğer çalışmalarda makine öğrenmesi uygulamaları yapılmaktadır. Makine öğrenmesinin uygulama alanlarına bazı örnekler aşağıda belirtilmiştir (Kaya, 2015).

### **2.1.2.1. Bilim ve Mühendislik Uygulamaları**

Bilimsel ve mühendislik alanında yapılan çalışmalarda seri olarak veri üretimi meydana gelmektedir. Üretimde idare performansı, kaliteli denetim süreci, deprem olasılıkları gibi pek çok alanda uygulamaları gösterilmiştir (Kaya, 2015).

### **2.1.2.2. Sağlık Uygulamaları**

Tarama testleri, verileri hastalıkların ön teşhisi, kalp krizi riskinin belirlenmesi, acil servis hastalarının vaziyetlerine bakılarak risk durumunu ve önceliklerin belirlenmesi gibi bu alanda birçok uygulama bulunmaktadır (Kaya, 2015).

### **2.1.2.3. Bankacılık ve Finansal Uygulamaları**

Bankacılık bölümünde kredi ve kredi kartı risk seviyesinin oranını, sahtekarlık metotlarının belirlenmesi, kar analizi gibi alanlarda uygulamalara rastlanmaktadır. Finans kuruluşları stok fiyat saptamalarını, portföy yönetimi gibi benzer alanlarda birçok uygulama ortaya çıkmıştır (Kaya, 2015).

### **2.1.2.4. Eğitim uygulamaları**

Öğrencilerin başarı durumlarıyla ilgili çeşitli uygulamalar tasarlanmıştır. Bunlar öğrencilerin öğrenim durumlarını belirleyerek elde edilen verileri ışığında başarı ve başarısızlık nedenleri, başarıyı artırıcı faktörlerin belirlenmesi, meslek seçiminde doğru oryantasyonun sağlanabilmesi, akademik başarı ile öğrenci verilerinin ilişki orantılarını bulunması gibi çeşitli yöntemler ile eğitim kalitesini ve başarımlarını artırıcı çalışmalar yapılmıştır (Kaya, 2015).

### **2.1.2.5. Web Uygulamaları**

Web üzerindeki hizmetlerin çeşitlenmesi ve artmasıyla web verileri kapasite ve karmaşıklık açısından hızla artmaktadır. Web madenciliği özetle internetten fonksiyonel veri keşfi olarak tanımlanmıştır. Web sitelerinde ürün satan firmalar makine öğrenmesi uygulamalarını kullanarak müşterilerin baktıkları ürünlerin benzerlerini göz önüne sererek satın almaları yönünde tavsiyede bulunabilmektedir (Kaya, 2015).

### **2.1.2.6. Ses ve Görüntü Tanıma Uygulamaları**

Makineler tarafından sayısal ortamda kayıt altına alınmış ses ve görüntü verileri üzerinden tanıma işlemi yapılabilmektedir. Birçok alanda da görülen bu uygulama örnekleri; kimlik tespiti, yüz tanıma, el yazısı tanıma, karakter tespiti, konuşmaların tanınması, plaka okuma, üretim hatalarının tespiti, tıbbi görüntülerin tanınması, robot algılaması, savunma sistemleri, uydu görüntüleri tanıma gibi pek çok uygulamaları görülmektedir (Baykasoğlu, 2005).

### **2.1.3. Makine Öğrenme Teknikleri**

Makine öğrenmesi, makinelerin geçmiş deneyimlerinden yararlanarak kendi kendine öğrenmesi neticesinde belli bir çıkış elde etme varsayımdır. Öğrenme sürecinde eğitim veri tabanını, modelin başarımını ölçmek için ise test veri tabanını kullanırlar. Eğitim veri tabanının akla uygun ve herkesçe kabul edilen verilerden meydana gelmesi gerekir. Böylece en iyi ve başarılı sonuca varılabilir. En uygun model belirlendikten sonra sistem tahminde bulunmaktadır (Kaya, 2015).

#### **2.1.3.1. Sınıflandırma**

Sınıflandırma, daha önceki deneyimlerden yararlanarak yeni gelen verilerin sınıflarını tahmin etme işlemidir. Aynı zamanda paradigmalardan çıkarım diye tanımlanır. İstenilen çıktı, modellemeyle önceden açıklanmamış paradigmalardan değerlendirilerek en güçlü başarımları gösterecek sınıflandırıcıyı belirtmektir. Veri setinin, nitelikleri nicel veya simgesel değerleri olabilir. Sınıfın niteliklerinde kesin değerler kullanılır, sınıflandırıcı nitelik iki değer alır, ikiden fazla değer kullanılması halinde çoklu sınıflandırma modeline geçmelidir. Kaynaklarda, çeşitli sınıflandırma algoritmalarını değerlendiren metotlar keşfedilmesine rağmen çalışma olanakları birbirine yakındır. Sınıflandırma metotları ile işlev ve fonksiyon yapılandıran kurallar öğrenilebilir (İkizler ve ark., 2002).

Sınıflandırma teknikleri ile kredi kartı portföyü, satın alma hareketi, tıp, kredi derecelendirmesi ve stratejik idare gibi farklı bölümlerde birçok uygulama tasarlanmıştır. Bir sınıflandırma işleminde, eğitim veri tabanları gözlemlerden meydana gelir. Aynı zamanda veri setinin her gözlem verisi  $x$ , gözlem sınıfının etiketi  $y$  olarak varsayılırsa veri seti  $(x, y)$  vektörü şeklinde tanımlanmaktadır. Sınıflandırma daha önceden tanımlanmış

sınıfın bir  $y$  etiketinin, özneliklerinin vektörü ile eşleşen bir  $x$  fonksiyonun öğrenme sürecidir (Kaya, 2015).



Şekil 2.1. Sınıflandırma süreci (Kaya, 2015)

Sınıflandırma modeli ile eğitim veri setinin içinde bulunan verilerinin hangi sınıfa ait olduğu öngörülebilir. Sınıflandırma modelinde sınıflandırma aşamaları iki etapta oluşturulmuştur. İlk etapta eğitim veri setine elverişli modelleri seçmemiz gerekir. Bunun için en elverişli algoritma tasarlanıp eğitim aşaması tamamlanır. İkinci aşamada test veri setini değerlendirerek yapılan modelin başarımı ölçülür, bu aşama test aşaması diye tanımlanabilir. Sınıflandırma yöntemleri olarak karar ağaçları, yapay sinir ağları, destek vektör makineleri ve istatistiksel sınıflandırma yöntemleri kullanılmaktadır (Durak, 2011).

### 2.1.3.2. Karar Ağacı

Karar ağacı, karar ağaç düğümleri ve sınıf etiketlerini açıklayan yapraklardan meydana gelir. Karar ağacı ile ortaya çıkan ağaç modeli ters dönmüş bir ağaç biçimindedir. Hiyerarşik bir metot olan karar ağacı diğer algoritmalara göre değerlendirilmesi, hatalara ulaşması ve algoritma neticesinde uygulamanın tasarlanması oldukça kolaydır. Sınıflandırma işleminde ilk aşamada önemli özellikler belirleyip kök düğümü belirler. Karar düğümü sayısına, düğümler arasındaki eşik değerlerine ve elde edilen çıktıda verinin ait olduğu sınıfı gösteren sonuç yapraklarına bakılarak, en iyi sonucu veren karar ağacı meydana

gelir. Her dal yeni bir karar düğümüne bağlanırken, yeni özellikler test edilir ve bu düğümlerden de yeni dallar devam etmektedir (Wu ve ark, 2006).

### **2.1.3.3. Destek Vektör Makinesi**

Destek vektör makinesi (DVM), denetimli öğrenme kategorisinde yer alır. Başarımının güçlü olması sebebiyle sıkça tercih edilen bir yöntemdir. Sınıfları farklılaştıran marjı en büyük, doğrusal bir ayırıcı işlev olmasını amaçlar. Doğrusal olarak ayırıştırma yapılamayan veri setleri için, doğrusal olarak ayırıştırma yapılabilen uygun ortama çekilerek sınıflandırma yeni ortamda biçimlendirilir. Bu dönüşüm işleminde DVM'nin çekirdek olarak tanımlanan işlevleri kullanılır (Kavzoğlu ve Çölkesen, 2010).

### **2.1.3.4. Kümeleme**

Kümeleme metotlarıyla gruplandırma işlemi için veri setindeki gözlemlerin benzeşimlerine bakılıp uygun gruplar seçilir. Kümelemenin sınıflandırma metotlarından farkı kümelenen verilerin sınıflarının önceden belli olmamasıdır. Küme oluşturma metotları ve değerlendirilen verilere göre bir algoritma modeli tasarlanır.

Kümeleme yönteminde, veriler iki ayrı yaklaşıma göre değerlendirilir. Bunlar mesafeye göre ve model tabanlı yaklaşım teknikleridir. Mesafeye göre yaklaşımda, veriler arasında mesafe veya benzerlik ölçüsü değerlendirilir. Model-tabanlı yaklaşımda, veri sonlu karışım modeli tarafından yapılandırılan modelin parametreleri ile tahminlerde bulunur. Her iki yaklaşımda da bölümlenmeli ya da hiyerarşik kümeleme metotları kullanılır. Bu yaklaşımların kapsamında üç ayrı algoritma çeşidi meydana gelir.

- Bölümlenmeli algoritmalar,
- Yoğunluğa dayalı algoritmalar,
- Grid temelli algoritma.

Yapay sinir ağları, istatistiksel yöntemler, birliktelik kuralları ve regresyon gibi kullandığımız makine öğrenme teknikleri de bulunmaktadır (Kaya, 2015).

## 2.2. Yapay Sinir Ağları

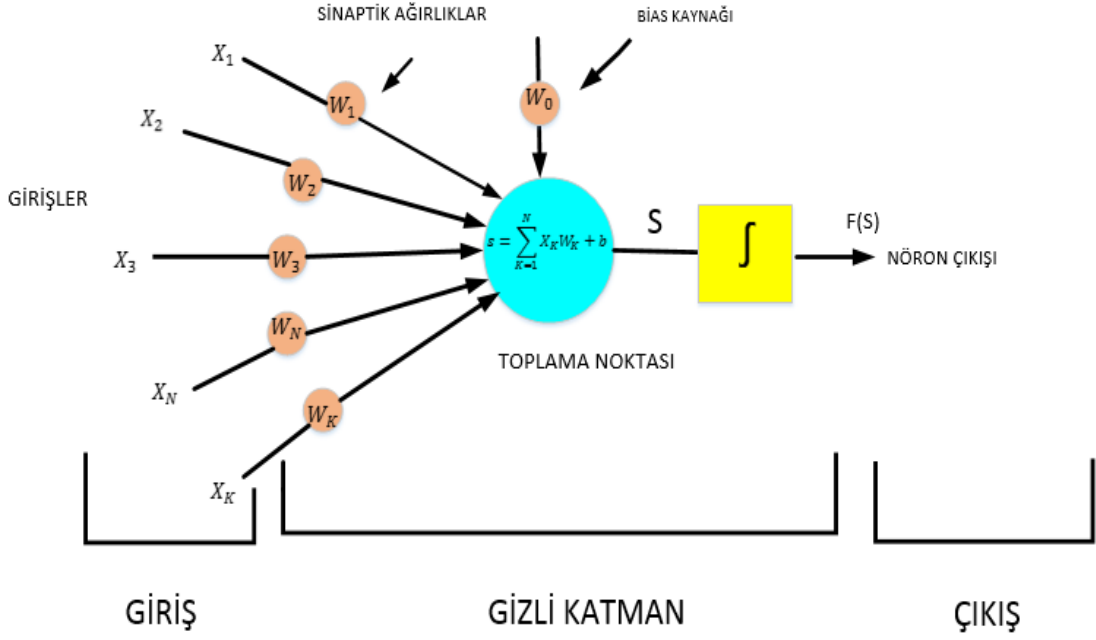
Yapay sinir ağları, insan aklından esinlenerek geliştirilmiş, ağırlıklı bağlantılar vasıtasıyla birbirine bağlanan ve her biri kendi hafızasına sahip fonksiyon öğelerinden meydana gelen paralel ve dağıtılmış bilgi işlem yapılarıdır. Farklı bir deyişle YSA, biyolojik sinir ağlarını taklit eden bilgisayar programlarıdır. YSA zaman zaman bağlantıcılık, paralel dağıtılmış işlem, sinirsel-işlem, doğal zekâ sistemleri ve makine öğrenme algoritmaları gibi adlarla tanımlanmaktadır.

YSA bir programcının geleneksel beceri gerektirmeyen, kendi kendine öğrenme mekanizmasına sahiptir. YSA'lar aynı zamanda, ezberleme ve bilgiler arasında bağlantıyı yapılandırma kabiliyetine sahiptirler.

YSA insan beyninin birkaç organizasyon öğelerini andıran özellikleri kullanmaktadır. YSA bilgi işleme yöntemleri yeni kuşağın sembolüdür. Geniş anlamda YSA'nın; model seçimi ve sınıflandırması, fonksiyon tahmine uygun çıktı değeri ortaya çıkarmak ve veri sınıflandırması gibi işlerdeki başarımı oldukça iyidir. Geleneksel bilgisayarlar ise özellikle model seçme işinde elverişsizdir ve yalnız algoritmaya dair hesaplama işlemleriyle salt aritmetik fonksiyonlarda hızlıdır. Belirsiz, gürültülü ve eksik bilgilerin işlenmesinde yapay sinir ağları başarıyla kullanılmaktadır (Elmas, 2011).

### 2.2.1. Yapay Sinir Ağların Yapısı

Yapay sinir ağları, insan beyninin veri işleme yapısından esinlenerek geliştirilmiş bilgi işlem teknolojisidir. Bir YSA ile basit biyolojik sinir sisteminin çalışma şekli benzetilir. Benzetimi yapılan sinir hücreleri nöronlar içerirler ve bu nöronlar çeşitli şekillerde birbirine bağlanarak tüm ağı oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi oluşturma kapasitesine sahiptirler.



Şekil 2.2. Basit bir sinir ağı yapısı (Aşkın, 2011)

Şekil 2.2’de basit bir yapay sinir ağı yapısı gösterilmiştir. Katmanların değişik şekilde birbirleriyle bağlanmaları değişik ağı mimarileri doğurur. YSA’lar temel olarak üç katmandan oluşur. Giriş katmanında giriş sinyalleri bulunur. Giriş sinyalleri  $X_1, X_2, X_3, \dots, X_N$  normalde sürekli değişkenlerdir fakat ayrık sinyallerde olabilir. Gizli katmanda her giriş sinyali bağlantı ağırlıklarıyla çarpılır. Ağırlıklar elektriksel sinyalin akışına göre pozitif (hızlandırıcı) veya negatif (yavaşlatıcı) olabilir. Toplama noktasında bütün ağırlıklı giriş sinyalleri ve bias sinyali (b) toplanır ve son olarak toplam çıkışından elde edilen sonucu doğrusal olmayan bir aktivasyon fonksiyonundan geçirilerek çıkış katmanına girilir (Bose, 2002).

Çıkış katmanındaki her nöron değiştirilebilir ağırlıklarla bağlantılı gizli katmandaki nöronlar aracılığıyla bütün girişlerden sinyal alır. Uygun öğrenme yöntemini kullanarak bağlantı ağırlıkları ayarlanır (Foo ve Ghosh, 2002).

Matematiksel olarak ifade edilirse;

$$Y = F(S) = F[\sum_{K=1}^N X_K W_K + b] \quad (2.1)$$

### 2.2.2. Yapay Sinir Ağların Üstünlükleri

YSA doğrusal olmayan özelliğiyle karmaşık sistemlerde tahmin yapabilme kabiliyetlerinden dolayı istatistik hesaplamalara göre daha kolay ve kullanışlı sonuçlar elde edilir. Genel olarak; bilişim, sağlık, uzay-havacılık, işletme ve finans gibi birçok alanda kullanılırlar. YSA'lar paralel çalışmaya uygun bir yapıya sahiptir. Bundan dolayı performansı oldukça hızlıdır ve bununla birlikte hatayı tolere etme açısından oldukça iyidir. Çünkü seri bilgi işlem yapan sistemler birbirine bağlı olduklarından en ufak bir hata tüm sistemin bozulmasına neden olmaktadır. YSA'nın öğrenme yeteneği de kazanılmıştır. Böylece bir problemin çözüm yöntemi için herhangi bir bilgi verilmeksizin sadece örnek girdiler için sonuçlar üretilmektedir. YSA çalıştığı probleme göre eğitildikten sonra bu eğitim sırasında daha önce karşılaşmadığı bir olaya da yanıt verebilir. YSA çalıştığı probleme göre kendini düzenleyerek ağırlıklarını belirler. Bir problemi çözmek için eğitilen YSA herhangi bir başka problemde de kolaylıkla kullanılabilir. Bunun için gereken tek şey yeni problemlerin girdi ve çıktılarıyla ağına tekrar eğitilmesidir. Hız ve donanım ihtiyacı duyan YSA'lar paralel yapısında dolayı büyük ölçekli entegre devre teknolojisiyle gerçekleştirilir.

### 2.2.3. Yapay Sinir Ağların Dezavantajları

Yapay sinir ağlarının pek çok avantajının yanında bazı dezavantajları da vardır. Bunların en başında donanıma bağımlı çalışmaları sayılabilir. Gerçek zamanlı bilgi işleyebilmeleri için paralel çalışan işlemcilerin varlığına ihtiyaç vardır.

Bir ağına nasıl oluşturulması gerektiğini belirleyecek kuralların olmaması da başka bir dezavantajdır. Probleme uygun ağ yapısının belirlenmesi genellikle deneme yanılma yolu ile yapılmaktadır. Bu ise önemli bir problemdir. Çünkü eğer problem için uygun bir ağ oluşturulamaz ise çözümlü olan bir problemin çözülmemesi veya performansı düşük çözümlerin elde edilmesi söz konusu olabilir. Bazı ağlarda ağına parametre değerlerinin (mesela öğrenme katsayısı, nöron sayısı, katman sayısı vb.) belirlenmesinde de bir kural olmaması bir dezavantajdır.

Ağına eğitilmesinin ne zaman bitirileceğine karar vermek için geliştirilmiş optimum bir yöntem de yoktur. Ağına örnekler üzerindeki hatasının belirli bir değerin altına indirilmesi eğitimin tamamlanması için yeterli görülmektedir. Fakat neticede optimum öğrenmenin

gerçekleştığı söylenememektedir. Sadece iyi çözümler üretebilen bir ağ oluşturulabilir (Öztemel, 2013).

#### **2.2.4. Yapay Sinir Ağlarda Sınıflandırma**

Yapay sinir ağları birbirleri bağlantılı sinir hücrelerinden oluşmaktadır. Her bir sinir hücresinin aralarındaki ilişki yapısı ağın yapısını belirler. Asıl amaca ulaşmak için aralarındaki ilişkiyi nasıl değiştirileceği öğrenme algoritması tarafından belirlenir. Yapay sinir ağları ağ yapılarına ve öğrenme algoritmalarına göre sınıflandırılmıştır.

##### **2.2.4.1. Yapay Sinir Ağların Yapılarına Göre Sınıflandırılması**

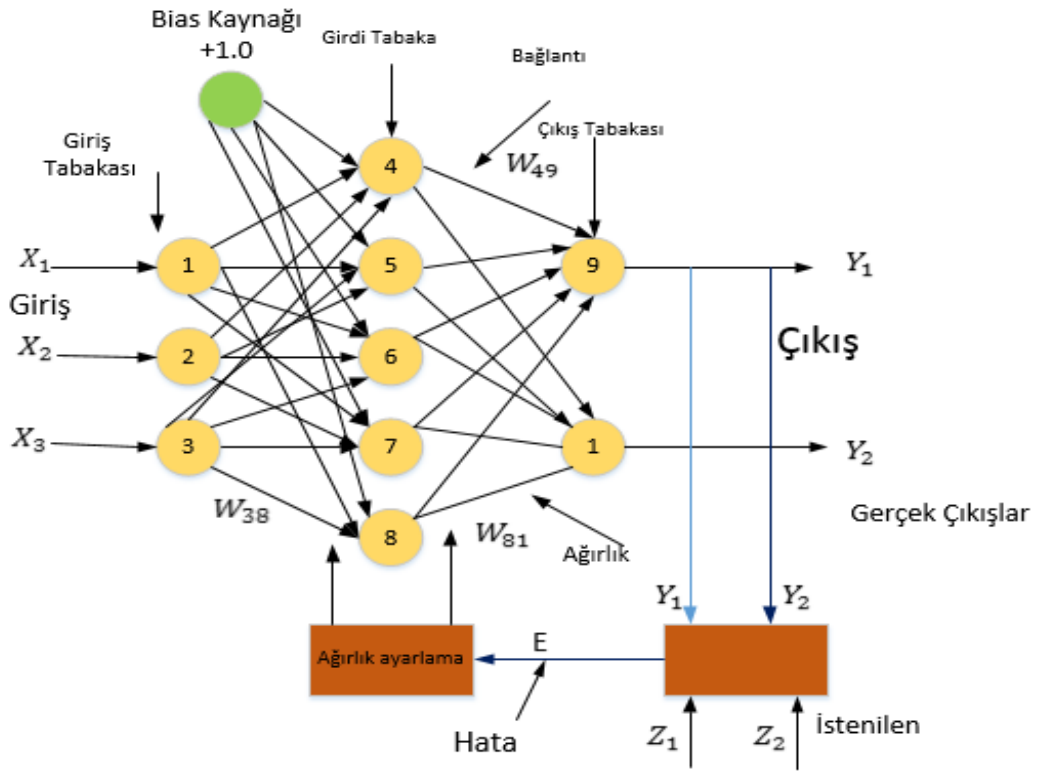
Sinir ağlarının işaretin akış yönüne bağlı olarak, ileri beslemeli (feedforward) ve geri beslemeli (feedback veya recurrent) ağlar olmak üzere iki çeşittir.

###### **2.2.4.1.1. İleri Beslemeli Ağlar**

İleri beslemeli ağları, YSA'ların en basit ve ilk yapısıdır. Bu ağda bilgi sadece ileri yönde gizli veya çıkış katmanına hareket eder. Ağda herhangi bir geri bildirim yoktur (Nguyen, 2007).

Şekil 2.3'te çok katmanlı ileri beslemeli ağ yapısını görülmektedir. 3 girişli ( $X_1, X_2, X_3$ ), 2 çıkışlı ( $Y_1, Y_2$ ) ve gizli katmanda 5 tane nöronu olan bir ağ yapısıdır. Sinyaller daima ileri yönde taşınmaktadır.

Çok katmanlı ileri beslemeli ağlar doğrusal olmayan sistemlerin belirtilmesi ve kontrolü, finansal piyasa analizleri, sinyal modelleme, güç yük tahminlerinde danışmanlık öğrenme metodunu kullanarak bazı zor problemlerin çözümünde başarılıdır (He ve ark., 2000).



Şekil 2.3. Çok katmanlı ileri beslemeli ağ yapısı (Aşkın, 2011)

#### 2.2.4.1.2. Geri Beslemeli Ağlar

Bir geri beslemeli sinir ağı, çıkış ve ara katmanlardaki çıkışlarının, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır. Bu çeşit sinir ağlarının dinamik hafızaları vardır. Bu yapıda nöronların çıkışı sadece o anki giriş değerlerine bağlı değildir ayrıca önceki giriş değerlerine de bağlıdır. Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundur. Bu ağlar çeşitli tipteki zaman serilerinin tahmininde oldukça başarı sağlamıştır (Mackay, 1992).



Gelecek adım olarak,  $j$ . çıkış nöronu aktivasyon fonksiyonu  $F$  den geçirilerek aşağıdaki eşitlik bulunur.

$$Y_j(k+1)=F(Q_i(k)) \quad (2.5)$$

#### **2.2.4.2. Yapay Sinir Ağlarda Temel Öğrenme**

Yapay sinir ağlarında öğrenme kurallarının çoğu Hebb kuralına dayanmaktadır. Bu öğrenme algoritmalarının çoğunluğu matematik tabanlı olup ağırlıkların güncelleştirilmesi için kullanılırlar.

##### **2.2.4.2.1. Hebb Kurallı**

1949 yılında geliştirilen bu kurala göre, bir hücre diğer bir hücreden bilgi alışverişi yaparken her iki hücre aktif ise aralarında ki ağırlıkları kuvvetlendirirler. Eğer hücre aktif ise bağlı olduğu hücreyi aktifleştirir, pasif ise pasifleştirir.

##### **2.2.4.2.2. Hopfield Kurallı**

YSA elemanlarının aralarındaki bağlantılarının ne kadar kuvvetlendirilmesi veya zayıflatılması gerektiği dışında Hebb kuralına benzemektedir. Eğer girdi ve çıktılar ikisi de aktif/pasif ise öğrenme katsayısı kadar ağırlık değerlerini kuvvetlendir/zayıflat denmektedir. Öğrenme katsayısı genellikle 0-1 arasında değerler almaktadır (Öztemel, 2003).

##### **2.2.4.2.3. Delta Kurallı**

Delta kurallı Hebb kuralından daha gelişmiş olup aynı zamanda en çok kullanılan algoritmalarından biri haline gelmiştir. Bu kuralda istenilen çıktı ile ortaya çıkan çıktı arasındaki hata değerini indirmek için YSA elemanlarının arasındaki ağırlık değerlerini sürekli değiştirme prensibine dayanmaktadır. Hatayı indirmek için çıkış katmanından giriş katmanına doğru geri yayılarak azaltılır.

#### **2.2.4.2.4. Kohonen Kuralları**

Öncelikle bu algoritma danışmansız öğrenmeye bir örnektir. Çünkü bu kuralda istenilen çıkışa gereksinim duymaz. Bunun nedeni ise bu algoritmada ağırlık elemanları ağırlıklarını değiştirmek için birbiri ile yarışır. En büyük çıkışa sahip nöron kazanır ve bağlantı ağırlıkları değiştirilmektedir. Kazanan nöron komşularını uyarma ve yasaklama yetkisine sahiptir (Aşkın, 2011).

#### **2.2.4.3. Yapay Sinir Ağların Öğrenme Algoritmalarına Göre Sınıflandırılması**

YSA'larda bilgiler, elemanlar arasındaki bağlantı ağırlıklarında depolanmaktadır. YSA'larda öğrenme, nöronlar arasındaki ağırlık değerlerinin değişimi sonucu oluşur. Bir takım öğrenme algoritmaları kullanılarak nöronlar arasındaki ağırlık değerlerini dinamik olarak değiştirilebilir. Temelde bu öğrenme algoritmaları danışmanlı (supervised), danışmansız (unsupervised) ve destekleyici (reinforcement) öğrenme olmak üzere üç gruba ayrılır.

##### **2.2.4.3.1. Danışmanlı Öğrenme**

Bu öğrenme algoritması sistemin olaya hakim olması için bir öğretmene dayalı olarak çalışmaktadır. Anlatılmak istenen öğretmene sisteme öğrenilmesi gereken olay ile ilgili girdi/çıkış setini verir. Yani her örnek için hem girdiler hem de o girdiler karşılığında oluşturulması gereken çıktılar sisteme gösterilir. Sistemin görevi girdileri öğretmenin belirlediği çıktılara haritalamaktır. Bu sayede girdiler ile çıktılar arasındaki ilişki öğrenilmektedir (Öztemel, 2003).

Delta kuralı ve geri-yayılım (backpropagation) algoritması danışmanlı öğrenme algoritmalarına örnek verilebilir.

##### **2.2.4.3.2. Danışmansız Öğrenme**

Danışmansız öğrenmede öğrenme süreci herhangi bir öğreticiye bağlı değildir. Öğrenme sistemin hedeflenen bir çıkışı yoktur verilen veri setine göre kendi kendine örneklemede bulunurlar. Özellikle sınıflandırmada kullanılan öğrenme algoritmasıdır.

Danışmansız olarak eğitilen ağlar, giriş bilgilerinin özelliklerine göre ağırlık değerlerini ayarlar. Danışmansız öğrenmeye; Grossberg tarafından geliştirilen ART(Adaptive Resonance Theory) veya Kohonen tarafından geliştirilen SOM(Self Organizing Map) örnek gösterilebilir.

#### 2.2.4.3.3. Pekiştirmeli Öğrenme

Bu yaklaşımda da öğrenen sistem bir uzman tarafından desteklenir ancak uzman, girdi seti için üretilmesi gereken çıktıyı vermez ama çıktının doğru veya yanlış olduğunu gösteren bir sinyal üretir. Sistem uzmandan gelen bu sinyali dikkate alarak öğrenme sürecini devam ettirir. Vektör kuantalamalı öğrenme (LVQ) bu öğrenme stratejisini kullanan ağ yapısına örnektir (Aşkın, 2011).

#### 2.2.5. Geri Yayılım Algoritması

Geri yayılma algoritması, sahip olduğu kolaylık ve uygulamadaki kullanılabilirlik gibi başarılarından dolayı ağ eğitimi için en popüler algoritmalarından biridir (Aktaş ve Okumuş, 2003). Bu algoritma; hataları çıkıştan girişe geriye doğru indirgemeye çalışmasından dolayı geri yayılım adını almıştır. Geri yayımlı öğrenme kuralı, ağ çıkışındaki mevcut hata seviyesine göre her bir katmandaki ağırlık değerlerini tekrar düzenlemek için kullanılmaktadır. Bir geri yayımlı ağ modelinde; giriş katmanı, gizli katman ve çıkış katmanı olmak üzere üç katman bulunmakla birlikte, problemin özelliklerine göre gizli katman sayısını artırmak mümkündür. Geri yayılım, çok katmanlı ağlarda kullanılan delta kuralı için geliştirilmiştir bir algoritmadır. Bu algoritma çok katlı ağlarda hesaplama işlemlerini öğrenmede kullanılabilir. Bu ağdaki hatalar, ileri besleme aktarım fonksiyonunun türevi tarafından, ileri besleme mekanizması dahilinde kullanılan aynı bağlantılar aracılığıyla, geriye doğru yayılmaktadır. Öğrenme işlemi, bu ağda basit çift yönlü hafıza birleştirme işlemine dayanmaktadır (Elmas, 2003).

Q katmanlı ileri beslemeli bir ağ için geri yayılım algoritması;

$q=1,2,3,\dots,Q$  katman numarası,

$H_i^p$ :  $q$ 'inci katmandaki  $i$  birimi girdisi,

$y_i^q$ :  $q$ 'ninci katmandaki  $i$  biriminin çıktısı,

$w_{ij}^q$ :  $(q-1)$ 'inci katmandaki  $i$  birimini,  $q$ 'ncü

katmandaki  $j$  birimine bağlayan ağırlık olmak üzere:

1. *Adım*:  $w$ 'ye reel değerleri küçük rastlantısal sayılar başlangıç değeri olarak atanır.

2. *Adım*: Rastgele bir (gidiş-hedef) çalışma modeli seçilir ve  $q$  katmandaki her bir  $j$  birimi için ileri yönde "çıkış" değerleri hesaplanır. Böylece çıkış,

$$y_i^q = f(\sum_i y_i^{q-1} w_{ij}^q) \text{ olur.} \quad (2.6)$$

3. *Adım*: Çıkış birimleri için hata terimleri hesaplanır.

$$\delta_i^0 = (y_i^0 - y_i^p) f'(H_i^0) \quad (2.7)$$

4. *Adım*:  $q=Q, Q-1, \dots, 2$  katmanlarındaki tüm  $i$  birimleri için geriye yayılımla delta ları yani gizli katman birimleri için hata terimleri hesaplanır.

$$\delta_i^{q-1} = f'(H_i^{q-1}) \sum_i \delta_i^q w_{ij}^q \quad (2.8)$$

5. *Adım*: Bütün bu ağırlıklar  $w_{ij}$ 'leri kullanılarak güncellenir.

$$w_{ij}^{yeni} = w_{ij}^{eski} + \Delta w_{ij}^q \quad (2.9)$$

$$\Delta w_{ij}^q = \eta \delta_i^q y_i^{q-1} \quad (2.10)$$

6. *Adım*: İkinci adıma dönüp, toplam hata kabul edilebilir bir düzeye gelene kadar hep bir  $p$  modeli için işlemler tekrarlanır.

Geri yayılım algoritmasının amacı uygunluk fonksiyonunu minimum yapmaktır. Uygunluk fonksiyonu YSA'nın ağırlık değerlerine bağlı olduğundan, algoritma YSA ağırlıklarının en uygun biçimde değiştirilmesi işlemlerden oluşmaktadır (Lin ve Lee, 1996).

## 2.2.6. Yapay Sinir Ağların Yapıları

### 2.2.6.1. Adaline

Adaline, YSA'nın en basit versiyonudur. Widrow tarafından önerilen adaptif doğrusal eleman, her adımda ortalama karesel hatayı azaltarak ağırlık değerlerini ayarlayan ve sınıflandırmayı sağlayan basit bir perceptrondur. Ayrıca öğrenme algoritması için danışmanlı öğrenmeyi kullanır. Adaline birçok uygulama için oldukça iyi çalışmasına rağmen doğrusal problem uzayıyla sınırlıdır. Doğrusal transfer fonksiyonu kullanırlar. Bu ağ yapısı ses sinyalleri üzerinde bulunan eko gürültülerinin giderilmesi yanında görüntü

sinyalleri üzerinde bulunan gürültülerin giderilmesi içinde aktif olarak kullanılır (Sağırođlu ve ark., 2003).

#### **2.2.6.2. Çok katmanlı Ağlar**

Adaline girdi ve çıktı arasındaki bağlantının doğrusal olmadığı sürece herhangi bir problemi çözmektedir. Minks (Hopfield ve Tank, 1985) YSA'ların basit algılayıcı modellerinin doğrusal olmayan problemlere çözüm getiremeyeceğini iddia etmiştir. Daha sonra XOR problemini çözerek çok katmanlı algılayıcı modelini geliştirmişlerdir. Yani; XOR, girdi ve çıktı arasında doğrusal olmayan problemlere çözüm bulmaktadır. Rumelhart ve arkadaşları tarafından geliştirilen bu model hata yayma veya geriye yayma olarak kullanılmaktadır.

Bütün işlem elemanları kendi katmanından bir üst katmana bağlı olup bilgi akışını ileriye doğru besler. Ayrıca geri besleme olmaz. Her giriş sayısı kadar nöron vardır. Ara katmanlarda bulunan nöron sayısı ve katman sayıları deneme ve yanılma yoluyla bulunmaktadır.

#### **2.2.6.3. Vektör Kuantalamalı Öğrenme**

Vektör Kuantalamalı Öğrenme (LVQ) ağı Kohonen tarafından geliştirilmiş olup aynı zamanda Kohonen ağı olarak da adlandırılır. Bazı ağı modellerinde eğitim aşamasında sisteme girdi ve oluşacak çıktı da verilir fakat bazı problemlerde gerçek çıktının ne olduğu bilinmediğinden sadece oluşan çıktının yanlış ya da doğru olup olmadığına dair bir sinyal üretilmiştir.

#### **2.2.6.4. ART Ağları**

ART ağları kendi kararlarını oluşturma yetilerine sahiptir. Ayrıca danışmansız öğrenme algoritmasına sahip olup gerçek zamanlı olarak oldukça hızlı ve kararlı bir şekilde öğrenim görürler. ART ağları gerçek zamanlı olarak oldukça hızlı ve kararlı bir şekilde öğrenme yeteneklerine sahiptirler. ART ağı modellerine daha önce farklı ve değişik nitelikteki bir örnek gösterildiğinde, ağı buna karşın kendi kendilerine kararlı bir yapı oluşturabilirler. Ağı yeni gelen bir girdi girdiği zaman daha önce kullandıkları sınıfların

kodlarına ulaşabilecek şekilde değişiklikler yapılır ya da uygun bir kodlama tasarlanır. ART ağları girdi değerlerini otomatik olarak normalize ederler. Çok fazla ve oldukça düşük orandaki gürültülerin girdi işaretindeki etkilerini ortadan kaldırırlar.

ART ağları iki katmandan oluşmaktadır. Bu katmanlar F1 ve F2 olarak adlandırılmıştır. F1 katmanı girdinin özelliklerini gösterirken, F2 katmanı kategorileri göstermektedir. Bu iki katman birbirine uzun dönemli hafıza (UDH) ile bağlanmaktadır. Girdi bilgileri F1 katmanından alınır ve sınıflandırma işlemi ise F2 katmanında yapılır.



Şekil 2.5. ART ağının genel yapısı (Aşkın, 2011)

ART ağlarında girdiler doğrudan sınıflandırılmazlar. İlk olarak girdilerin özellikleri incelenerek F1 katmanının aktivasyonu belirlenir. UDH'deki bağlantı değerleri ile gelen bilgiler kategorilere ayrılarak F2 katmanına gönderilir. F2 katmanındaki sınıflandırma ile F1 katmanından gelen sınıflandırma birbiri ile eşleştirilerek, eğer örnek belirlenmiş bir sınıfa uyuyorsa o kategoride gösterilir. Aksi takdirde ya yeni bir sınıf oluşturulur veya girdinin sınıflandırılması yapılmaz (Öztemel, 2003).

### 2.2.7. Eğim İnişi

Optimizasyon, makine öğrenmesinin büyük bir parçasıdır. Neredeyse her makine öğrenme algoritması, çekirdeğinde bir optimizasyon algoritmasına sahiptir.

Eğim inişi, bir maliyet fonksiyonunu (maliyet) en aza indirgeyen bir fonksiyonun ( $f$ ) parametrelerinin (katsayıları) değerlerini bulmak için kullanılan bir optimizasyon algoritmasıdır. Eğim inişi, parametreler analitik olarak hesaplanamadığında (örneğin doğrusal cebir kullanılarak) ve bir optimizasyon algoritması tarafından aranması gerektiğinde kullanılabilir en iyi yoldur.

### 2.2.7.1. Eğim İniş Prosedürü

İşlem, katsayı veya fonksiyon için katsayılar için başlangıç değerleriyle başlar. Bunlar 0.0 veya rastgele küçük bir değer olabilir.

$$\text{Katsayısı} = 0.0$$

Katsayıların maliyeti, fonksiyona bakılarak ve maliyetin hesaplanmasıyla değerlendirilir.

$$\text{Maliyet} = f(\text{katsayı})$$

Veya

$$\text{Maliyet} = \text{değerlendir } f(\text{katsayı})$$

Maliyetin türevi hesaplanır. Türev, hesaptan gelen bir kavramdır ve belirli bir noktadaki fonksiyonun eğimine atıfta bulunmaktadır. Bir sonraki iterasyonda daha düşük bir maliyet elde etmek için katsayı değerlerini hareket ettirme yönünü (işareti) bildiğimizden eğimi bilmemiz gerekir.

$$\text{Delta} = \text{türev}(\text{maliyet})$$

Türevden hangi yönde yokuş aşağı gittiğini bildiğimize göre katsayı değerlerini şimdi güncelleyebiliriz. Her güncelleme sırasında katsayıların ne kadarının değişebileceğini kontrol eden bir öğrenme hızı parametresi (alfa) belirtilmelidir.

$$\text{Katsayısı} = \text{katsayısı} - (\text{alfa} * \text{delta})$$

Bu işlem, katsayıların maliyeti 0.0 ya da yeterince iyi olması için sifıra yakın olana kadar tekrarlanır. Basit gradyan inişinin ne olduğunu görebilirsiniz. Maliyet fonksiyonunuzun gradyanını veya optimize ettiğiniz fonksiyonu bilmenizi ister fakat bunun yanında çok basittir.

### 2.2.7.2. Makine Öğrenimi İçin Stokastik Eğim İnişi

Eğim inişi çok büyük veri kümelerinde yavaş çalışabilir. Degrade indirgeme algoritmasının bir yinelemesi, eğitim veri setindeki her bir örnek için bir tahmin gerektirdiğinden, milyonlarca örneğe sahip olduğunuzda uzun zaman alabilir. Büyük miktarda veriye sahip olduğunuz durumlarda, stokastik düşme iniş denilen bir degrade kökenli bir varyasyon kullanabilirsiniz.

Bu varyasyonda, yukarıda açıklanan gradyan iniş prosedürü çalıştırılır ancak katsayılara güncelleme, örneklerin toplu halinin sonundan ziyade, her eğitim örneği için gerçekleştirilir. Prosedürün ilk adımı, eğitim veri setinin sırasının rastgele seçilmesini gerektirir. Bu, güncellemelerin katsayılara yapıldığı sırayı karıştırmak içindir. Katsayılar, her eğitim örneğinden sonra güncellendiğinden, güncellemeler bir yere gürültülü atlama olacak ve buna karşılık gelen maliyet fonksiyonu olacaktır. Katsayılara güncellemelerin sırasını karıştırarak, bu rasgele yürüyüşü kullanıyor ve dağılmış veya sıkışmamasını önüyor.

Katsayılar için güncelleme prosedürü, yukarıdaki ile aynıdır, ancak maliyet tüm eğitim modelleri üzerinde toplanmaz, bunun yerine bir eğitim modeli için hesaplanır. Öğrenme, çok büyük eğitim veri setleri için stokastik düşme inişiyle çok daha hızlı olabilir ve genellikle, yeterince iyi veya yeterince iyi bir katsayı grubuna ulaşmak için yalnızca veri setinde az sayıda geçişe ihtiyaç duyarsınız (Brownlee, 2016).

### 2.2.8. Boltzman Makineleri

Bir Boltzmann makinesi, simetrik olarak bağlı, sinir ağına benzer olan açık veya kapalı olmasına ilişkin stokastik kararlar veren bir ağıdır. Boltzmann makineleri, eğitim verisinde karmaşık düzenlilikleri temsil eden ilginç özellikleri keşfetmelerini sağlayan basit bir öğrenme algoritmasına sahiptir (Hinton ve Sejnowski, 1983). Öğrenme algoritması, özellik algılayıcılarının birçok katmanına sahip ağlarda çok yavaş ancak özellik algılayıcılarının tek bir katmanına sahip "kısıtlı Boltzmann makineleri" de hızlıdır. Birçok gizli katman, birinin özellik aktivasyonlarını bir sonraki eğitim verisi olarak kullanarak, kısıtlanmış Boltzmann makineleri oluşturarak verimli bir şekilde öğrenilebilir.

Boltzmann makineleri iki farklı hesaplama problemini çözmek için kullanılırlar. Bir arama problemi için, bağlantılar üzerindeki ağırlıklar sabitlenir ve bir maliyet fonksiyonunu temsil etmek için kullanılır. Bir Boltzmann makinesinin stokastik

dinamikleri daha sonra maliyet fonksiyonunun düşük değerlerine sahip olan ikili durum vektörlerini örneklemesine izin verir.

Bir öğrenme problemi için, Boltzmann makinesi bir dizi ikili veri vektörleri gösterilir ve bu olasılıklı vektörleri üretmeyi öğrenmek zorundadır. Bunu yapmak için bağlantılarda ağırlık bulunması gerekir, böylece diğer olası ikili vektörlere göre veri vektörlerinin maliyet fonksiyonunun düşük değerleri olur. Bir öğrenme problemini çözmek için Boltzmann makineleri ağırlıklarından çok az güncelleme yapar ve her güncelleme birçok farklı arama problemini çözmesini gerektirir.

### **2.2.9. Yapay Sinir Ağlarının Uygulamaları**

Bu kısımda yapay sinir Ağları ile ilgili yapılmış birkaç çalışma verilecektir. Örneğin; ErKaymaz ve Yaşar (2011), YSA 'yı kullanarak hava sıcaklığının tahminini yapmıştır. İleri beslemeli YSA modeli kullanarak ortam sıcaklığı en düşük hata ile Matlab ortamında geliştirilen uygulama ile tespit edilmeye çalışılmıştır. Ayrıca sistemin girişinden çıkışa doğru ileri beslemeli YSA'nın simulink modelleri yapılmıştır.

Yetkin (2014), Tanker şamandıra bağlama sistemlerinin yapay sinir ağları tekniğiyle optimizasyonu çalışmasında OrcaFlex programı ile 4 noktalı tanker-şamandıra bağlama sistemlerini modellendikten sonra değişik çevre şartları ve farklı bağlama şekilleri ile simülasyonlar yapmıştır. Bu simülasyonlardan elde edilen sonuçlar yapay sinir ağlarını eğitmede kullanılmış ve bir model elde edilmiştir.

Deveci (2012), Yapay sinir ağları ve bekleme süresinin tahmininde kullanılması adlı çalışmasında, yapay sinir ağları ile hastaların bekleme süresinin tahminini yapmıştır. Çeşitli istatistiksel normalizasyon tekniklerini öğrenme yöntemleri ile birlikte kullanmıştır.

Çırak ve Çokluk (2012), Yükseköğretimde öğrenci başarılarının sınıflandırılmasında Yapay Sinir Ağları ve lojistik regresyon yöntemlerinin kullanılması çalışmasında üniversite öğrencilerinin akademik başarılarını etkileyen değişkenler anketi kullanarak elde edilen bilgilerle öğrencilerin başarı durumlarına göre sınıflandırılmasında yapay sinir ağları ve lojistik regresyon yöntemlerini toplam doğru sınıflandırma oranlarından faydalanarak karşılaştırmıştır.

## 2.3. Tekrarlı Yapay Sinir Ağları

### 2.3.1. Giriş

Doğrusal olmayan aktivasyonlara sahip sonlu boyutlu TYSA, bitişik zaman adımları boyunca uzanan kenarların eklenmesiyle güçlendirilmiş ileri beslemeli sinir ağlarıdır ve bu da normal sinir ağ modeline zaman kavramını getirir. TYSA'lar neredeyse keyfi hesaplama yapabilecek zengin bir model ailesidir. Sigmoidal aktivasyon işlevlerine sahip sonlu boyutlu tekrarlayan bir sinir ağı, evrensel bir Turing makinesini taklit edebilir (Siegelmann ve Sontag, 1991). TYSA'ların keyfi hesaplama yapabilme kabiliyetleri onların ifade güçlerini gösterir, ancak C programlama dilinin keyfi programları eşit derecede kabiliyetli olduğu iddia edebilir. Ayrıca, C'nin icadının makine öğrenimi için her derde deva olduğunu iddia eden hiçbir bildiri yoktur. Bunun temel bir sebebi, C programlarının alanını verimli bir şekilde keşfetmenin basit bir yolu yoktur. Özellikle, seçilen bir kayıp fonksiyonunu en aza indirmek için keyfi bir C programının gradyanını hesaplamak için genel bir yol yoktur. Dahası, herhangi bir sonlu veri kümesi göz önüne alındığında, veri kümesini aşan, istenilen eğitim çıktısını üreten, ancak örnekleri test etmek için başarısız olan sayısız program mevcuttur. O halde neden benzer sorunlardan TYSA'lar daha az uğraşırlar. İlk olarak, sabit mimari (düğümün, kenarların ve aktivasyon işlevlerinin seti) göz önüne alındığında, bu mimari ile tekrar eden sinir ağları uçtan uca değiştirilebilir. Kayıp fonksiyonunun türevi, modelin her bir parametresine (ağırlık) göre hesaplanabilir. Dolayısıyla, TYSA'lar azalmaya dayalı eğitim için uygundur. İkincisi, TYSA'ların Turing eksiksizliği belirgin bir mimari ile sabit büyüklükte bir TYSA verildiğinde etkileyici bir özellik olsa da herhangi bir keyfi programı üretmek aslında mümkün değildir. Ayrıca, C'de oluşturulan bir programın aksine, bir tekrarlayan sinir ağı, ağırlık kaybı, terk etme ve bağımsızlık derecelerinin sınırlandırılması gibi aşırı uyumu önlemeye yardımcı olan standart parametrelerle düzenlenebilir.

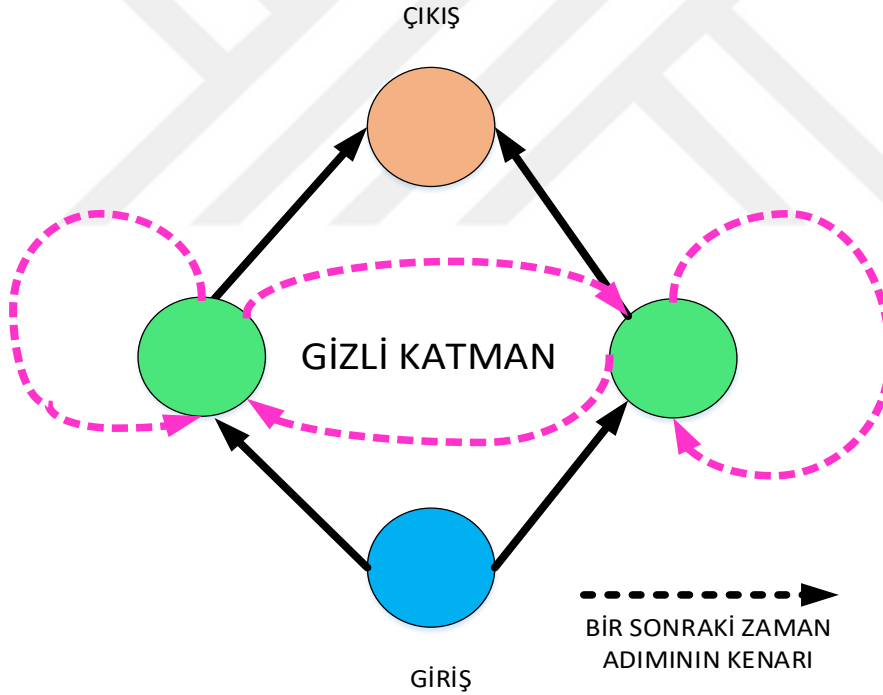
### 2.3.2. Tekrarlı Yapay Sinir Ağları

Tekrarlı yapay sinir ağları, birimler arasındaki bağlantıların yönlendirilmiş bir döngü oluşturduğu yapay sinir ağı sınıfıdır. TYSA, bitişik zaman adımları boyunca uzanan kenarların eklenmesiyle güçlendirilmiş ileri beslemeli sinir ağlarıdır ve bu da normal sinir

ağ modeline zaman kavramı getirir. İleri beslemeli ağlar gibi, TYSA'lar geleneksel kenarlar arasında döngülere sahip olmayabilir. Bununla birlikte, tekrar eden kenarlar olarak adlandırılan bitişik zaman adımlarını birbirine bağlayan kenarlar, bir düğümden kendi kendine bağlantı olan uzunluklu döngüler de dahil olmak üzere döngüler oluşturabilir. Zaman  $t$ 'de tekrarlayan kenarlı düğümler, geçerli veri noktası  $x^{(t)}$  ve ağın önceki durumundaki gizli düğüm değerleri  $h^{(t-1)}$  'den alınır. Her  $t$  zamanındaki çıktı  $y^{(t)}$ ,  $t$  zamanında gizli düğüm değerlerini  $h^{(t)}$  vererek hesaplanır.  $t-1$  zamanındaki  $x^{(t-1)}$  girişi,  $t$  zamanındaki çıktıyı  $y^{(t)}$  ve daha sonra tekrarlayan bağlantılar yoluyla etkileyebilir.

İki denklem, Şekil 2.6'daki gibi basit tekrar eden bir sinir ağında ileri geçişte her zaman adımında gerekli tüm hesaplamaları belirtmektedir:

$$h^{(t)} = \sigma(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h) \quad (2.11)$$



Şekil 2.6. Basit bir TYSA ağ modeli (Liptom ve ark., 2015)

$$y^{(t)} = \text{softmax}(W^{yh}h^{(t)} + b_y) \quad (2.12)$$

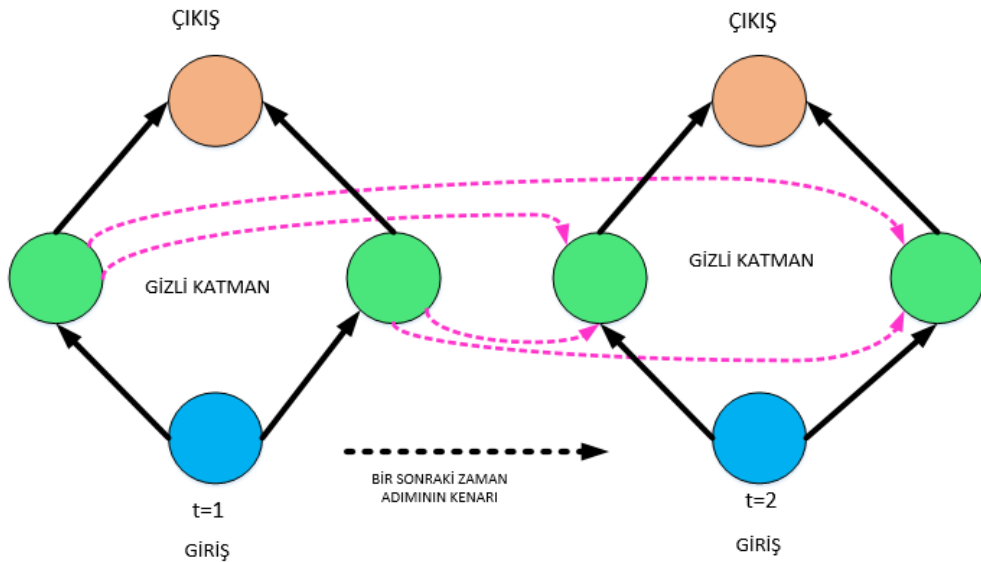
Burada,  $W_{hx}$ , giriş ile gizli katman arasındaki geleneksel ağırlıkların matrisi ve  $W_{hh}$ , bitişik zaman adımlarındaki gizli katman ile kendisi arasındaki tekrar eden ağırlıkların

matrisidir.  $b_h$  ve  $b_y$  vektörleri, her bir düğümün bir ofset öğrenmesine izin veren bias parametreleridir.

Şekil 2.6'da tasvir edilen ağın zaman adımlarındaki dinamikleri Şekil 2.7'deki gibi açılarak genişletilebilir. Bu tablo zaman adımında ağırlıklar göz önüne alındığında, ağ çevrimsel olarak değil, zaman basamağında bir katmana sahip derin bir ağ olarak yorumlanabilir. Ardından, açılmış ağın, geri yayılımı kullanarak birçok zaman kademesinde eğitilebileceği açıktır. Geri yayılım zamanı (BPTT) adı verilen bu algoritma Werbos (1990) tarafından tanıtılmıştır. Ortak geçerli kullanımdaki tekrarlayan ağlar bunu uygular.

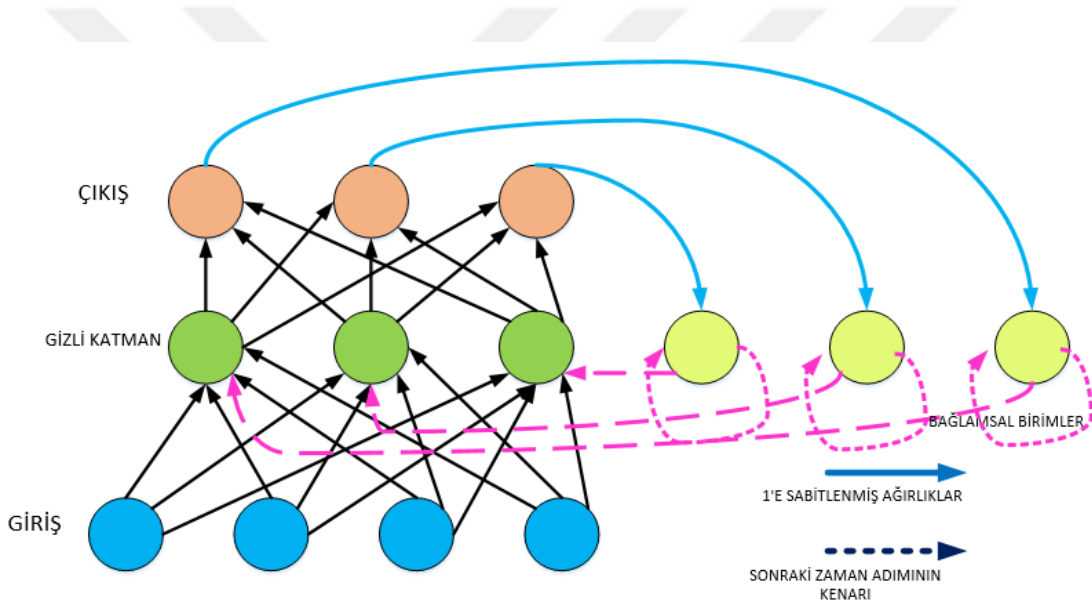
### 2.3.3. İlk Tekrar Eden Ağların Tasarımları

Tekrarlayan ağlar üzerine yapılan temel araştırmalar 1980'lerde gerçekleşti. 1982'de Hopfield, kalıp tanıma yeteneklerine sahip tekrarlayan sinir ağı ailesini tanıtmıştır. Bunlar düğümler arasındaki ağırlıkların değerleri ile tanımlanır ve bağlantı fonksiyonları sıfırdaki basit eşik değerlerdir. Bu ağlarda, düğümlerin değerlerini ayarlayarak ağa bir desen yerleştirilir. Ağ sonra güncelleme kurallarına göre bir süre çalışır ve sonunda başka bir desen okunur. Hafif alan ağları, bozulmuş bir versiyondan saklanan bir kalıbı kurtarmak için kullanışlıdır ve Boltzmann makineleri ve otomatik kodlayıcıların öncüleridir.



Şekil 2.7. Şekil 2.6'nın TYSA'larının zaman kademeleri boyunca açılması (Lipton ve ark., 2015)

Diziler üzerinde denetlenen öğrenme için bir mimari, Jordan (1986) tarafından tanıtıldı. Böyle bir ağ (Şekil 2.8), özel ünitelerle genişletilmiş tek gizli katmana sahip ileri beslemeli bir ağıdır. Çıkış düğümü değerleri, özel birimler tarafından beslenir ve bu değerler aşağıdaki zaman adımında bu değerleri gizli düğümlere gönderir. Çıktı değerleri işlemleri ise ağın önceki zaman adımlarında atılan işlemleri hatırlamasına izin verir. Birçok modern mimari, çıktı düğümlerinden ilgili doğrudan aktarım biçimini kullanır (Sutskever ve ark., 2014), doğal diller arasındaki cümleleri çevirir ve bir metin dizisi üretirken, her zaman adımında seçilen kelime, aşağıdaki zaman adımında girdi olarak ağa beslenir. Buna ek olarak, bir Jordan ağı özel birimleri kendine bağlıdır. Sezgisel olarak, bu kenarlar, her ara zaman adımında çıktıyı bozmadan birden fazla zaman adımında bilgi göndermeye izin verir.



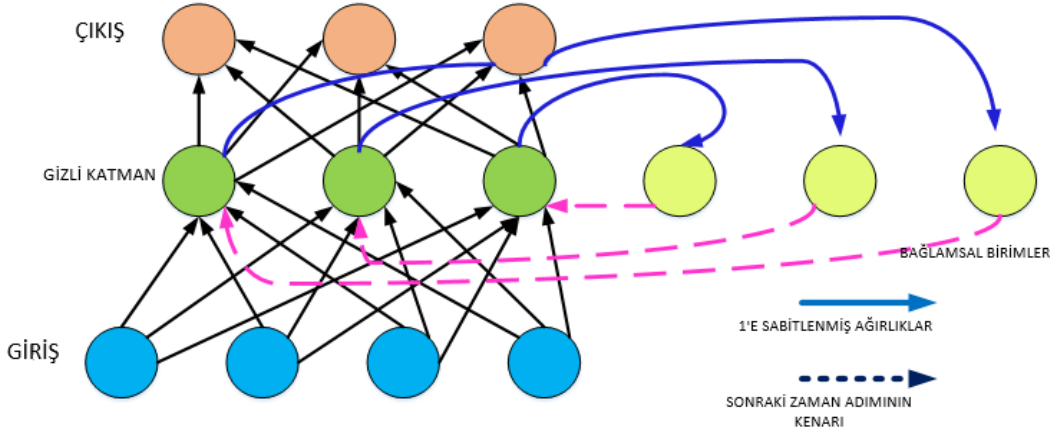
Şekil 2.8. Jordan tarafından önerilen TYSA modeli (Lipton ve ark., 2015)

Elman (1990) tarafından ortaya atılan mimari, önceki Jordan mimarisinden daha basittir. Gizli katmandaki her birimle ilişkili olan bir bağlamsal birimdir. Her bir birim  $j'$ , sabit bir ağırlığın bir kenarı boyunca önceki zaman adımına karşılık gelen ilgili gizli düğümün  $j$  durumunu girdi  $w_{j,j} = 1$  olarak alır. Bu değer daha sonra standart bir kenar boyunca aynı gizli düğüm  $j'$ 'ye geri beslenir. Bu mimari, her bir gizli düğümünün tek bir kendiliğinden bağlı yinelenen kenarına sahip olduğu basit bir TSA'ya eşdeğerdir. Gizli düğümleri kendi kendine bağlayan sabit ağırlıklı yinelenen kenarlar fikri, UKVH ağlarında yapılan daha sonraki çalışmalarda temel taşır (Hochreiter ve Schmidhuber, 1997).

Elman (1990), geri iletme yöntemini kullanarak ağa eğitim verir ve ağın zaman bağımlılıklarını öğrenebileceğini gösterir. Çalışma iki deney kümesine sahiptir. İlk önce mantıksal işlemi, üç belirteç dizisini birbirine ekleyerek zaman alanına özel-veya (XOR) genişletir. Her bir segment için, ör. "011", ilk iki belirteç ("01") rasgele seçilir ve üçüncü ("1") ilk ikisinde XOR işlemiyle belirlenir. Rastgele tahmin, %50 oranında doğruluk elde etmelidir. Mükemmel bir sistem, ilk iki token için aynı şeyi yapmalı, ancak üçüncü mütalaasını mükemmel bir şekilde tahmin ederek %66,7'lik bir doğruluk elde etmelidir. Elman'ın basit ağı (1990) gerçekte ulaşılabilir bu maksimum puana yaklaşmaktadır.

#### 2.3.4. Tekrarlı Ağları Eğitme

Yinelenen ağlar ile öğrenmenin uzun zamandan beri yapıldığı düşünülmektedir. Standart ileri-beslemeli ağlar için bile, optimizasyon görevi NP-complete (Blum ve Rivest, 1993) 'dır. Ancak tekrarlayan ağlarla öğrenmek, Bengio ve diğerleri tarafından tarif edildiği gibi, uzun menzilli bağımlılıkları öğrenmenin başarısızlığı nedeniyle özellikle zor olabilir (Hochreiter ve ark., 2001). Gradientlerin kaybolması ve patlaması problemleri, birçok zaman adımında hataları tersine döndürdüğünde ortaya çıkar. Basit bir örnek olarak, tek bir giriş düğümlü, tek bir çıkış düğümlü ve tek bir tekrar eden gizli düğümlü bir ağ düşünün (Şekil 2.9). Şimdi  $\tau$  zamanında  $\tau$  ağında geçirilen bir girdi ve ara zaman adımlarında sıfır girdisi varsayılarak  $t$  zamanında hesaplanan bir hata düşünün. Zaman adımları arasındaki ağırlıkların birleştirilmesi, gizli düğümdeki tekrar eden kenarın daima aynı ağırlığa sahip olduğu anlamına gelir. Bu nedenle, zaman  $\tau$ 'daki girdinin  $t$  zamandaki çıktıya katkısı,  $t-\tau$  büyüdükçe patlayabilir veya üssel olarak hızla sıfıra yaklaşır. Dolayısıyla, hatanın girdiye göre türevi patlayacaktır veya kaybolacaktır.



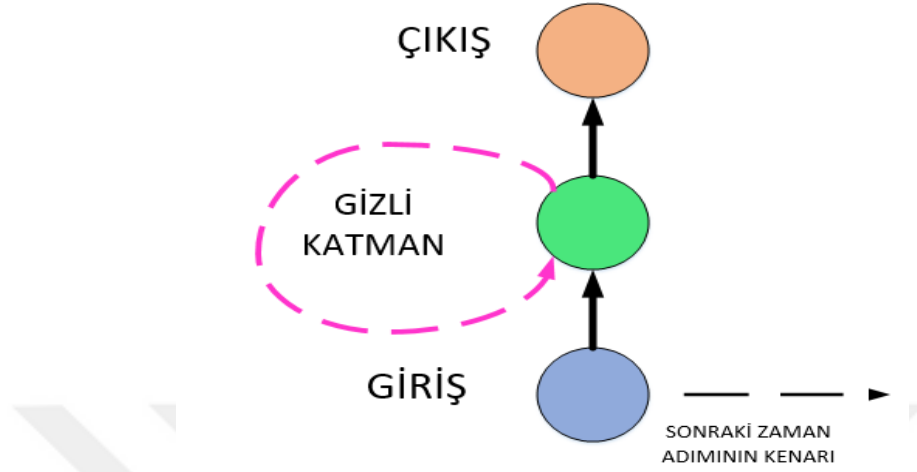
Şekil 2.9. Elman (1990) tarafından tarif edilen TYSA modeli (Lipton ve ark., 2015)

İki olaydan hangisinin gerçekleştiğini, gizli düğümdeki aktivasyon fonksiyonunun üzerinden ve  $|w_{jj}| > 1$  veya  $|w_{jj}| < 1$  olup olmadığı tekrarlayan kenarın ağırlığına bağlıdır (Şekil 2.11). Bir sigmoid aktivasyon fonksiyonu göz önüne alındığında, kaybolan gradient problemi daha çok basmaktadır, fakat doğru doğrusal bir birim  $\max(0, x)$  ile patlama gradyanını hayal etmek daha kolaydır. Pascanu ve ark. (2012), bu problemlerin meydana gelebileceği kesin koşulları karakterize eden kaybolan ve patlayan gradient problemlerinin kapsamlı bir matematiksel çözümlerini verir. Bu koşullar göz önüne alındığında, ağırlıkları, ortadan kaybolan veya patlayan değerlere zorlayan normalleştirme terimi yoluyla eğitime bir yaklaşım önerilmektedir.

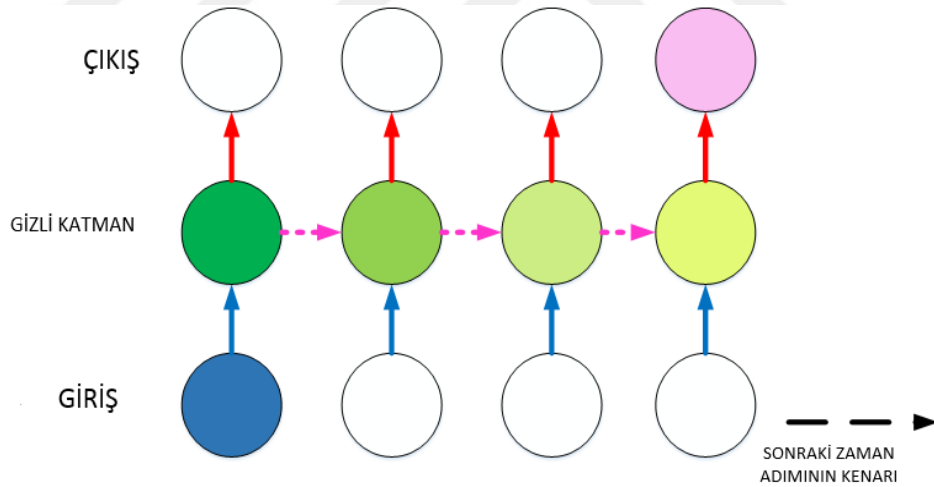
Kesintisiz geri yayılım süresi (TBPTT), kesintisiz çalışan ağlar için patlama eğilimi sorununa bir çözümdür (Williams ve Zipser, 1989). TBPTT ile, hatanın yayılabileceği bazı maksimum zaman adımları sayısı ayarlanır. Patlayan gradyan problemini hafifletmek için küçük bir kesme ile TBPTT kullanılabilir ancak, uzun menzilli bağımlılıkları öğrenme yeteneğini saklı tutmak gerekir. Aşağıda açıklanan UKVH mimarisi, kaybolan gradient sorununun bir çözümü olarak sabit birim ağırlığına sahip tekrar eden kenarlı dikkatle tasarlanmış düğümler kullanmaktadır.

Yerel optima meselesi, etkili bir eğitimin, ağ mimarisini değiştirerek basitçe ele alınamayacak bir engeli teşkil eder. Gizli katmanlı bir ileri beslemeli ağın bile en iyi duruma getirilmesi NP ile komple bir sorundur (Blum ve Rivest, 1993). Bununla birlikte, son zamanlardaki deneysel ve teorik çalışmalar, pratikte konunun bir kez düşündüğü kadar önemli olmayabileceğini önermektedir. Dauphin ve ark. (2014), büyük sinir ağlarının hata yüzeylerinde birçok kritik nokta mevcut olmakla birlikte, eyer noktalarının gerçek yerel

minimum oranının şebeke büyüklüğüyle katlanarak arttığını noktalarından kaçacak şekilde tasarlanabileceğini göstermektedir.



Şekil 2.10. Bir giriş, çıkış ve gizli katmana sahip basit bir TYSA modeli (Lipton ve ark., 2015)



Şekil 2.11. Şekil 2.10 'da gösterilen ağı kullanarak Graves (2012)'den uyarlanan bir TYSA modeli (Lipton ve ark., 2015)

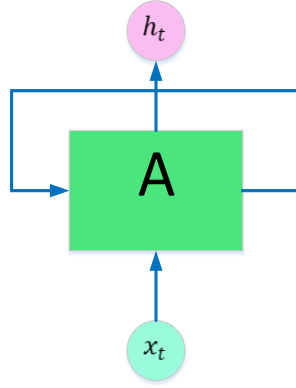
Genel olarak, aşağıda açıklanan gelişmiş mimarilerle birlikte, hızlı uygulamalar ve daha iyi gradyan takip eden buluşsal yöntemler TYSA eğitimini uygulanabilir hale getirmiştir. Theano (Bergstra ve ark., 2010) ve Torch (Collobert ve ark., 2011) paketleri gibi GPU'ları kullanarak ileriye ve geriye doğru yayılım uygulamaları hızlı eğitim algoritmalarını uygulamaya koymayı kolaylaştırmıştır. 1996'da UKVH'nın piyasaya sürülmesinden önce, uzun zaman boşluklarını kapatmak için tekrarlayan ağları kurma girişimleri, rastgele

tahminlerden daha iyi sonuç vermemektedir (Hochreiter ve Schmidhuber, 1996). Bununla birlikte, TYSA'lar sık sık başarılı bir şekilde eğitilmektedir.

Bazı görevler için, serbestçe temin edilebilen yazılımlar tek bir GPU üzerinde çalıştırılabilir ve saatler içerisinde ilgi çekici sonuçlar üretebilir (Karpathy, 2015). Martens ve Sutskever (2011), Hessian içermeyen, kesilmiş Newton yaklaşımı ile tekrarlayan sinir ağları performansında başarılı olduklarını bildirdiler ve metodu (Sutskever ve ark., 2011)'de bir kerede bir karakter üretmeyi öğrenen bir ağa uygulamışlardır. Yazarlar, sinir ağlarının hata yüzeylerinde eyer noktalarının bolluğunu açıklayan makalede (Dauphin ve ark., 2014), Newton'un yönteminin eyersiz bir halini sunmuşlardır. Eyer noktaları da dahil olmak üzere kritik noktalara çekilen Newton yönteminin aksine, bu varyant, onlardan kaçmak için özel olarak tasarlanmıştır. Deneysel sonuçlar, yinelenen ağlarda iyileştirilmiş bir performans sergiliyor. Newton'un yöntemi, büyük ağlar için pahalı olan Hessian'ı, parametrelerin sayısı ile kare olarak ölçeklendirerek hesaplamayı gerektirir. Algoritmaları yalnızca Hessian'a yaklaşırken, SGD'ye kıyasla hala hesaplama açısından pahalıdır. Böylece yazarlar, eyersiz modeli olan Newton metodunun sadece SGD'nin sıkışıp kaldıkları yerlerde uygulandığı karma bir yaklaşımı açıklarlar.

### **2.3.5. Uzun Kısa Vadeli Hafıza Ağları**

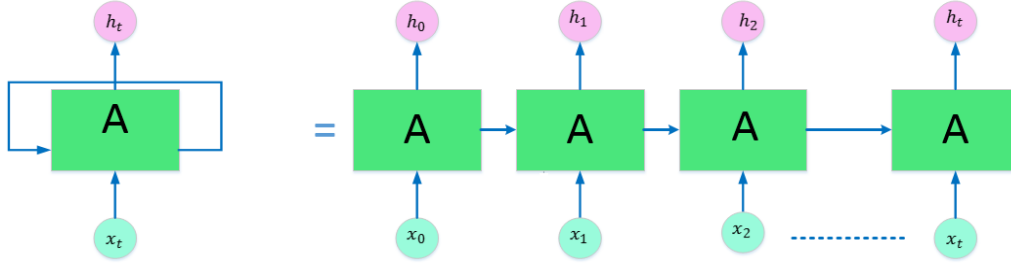
YSA konusunda da bahsedildiği gibi tüm bilim adamları insan beyninden esinlenerek akıllı sistemler tasarlamaya çalışmıştır. Burada anlatılmak istenen olay insanlar düşüncelerini sürekli olarak sıfırlanıp yeniden başlatmazlar. Bundan dolayı sürekli olarak sözcükleri birbirine bağlayıp yeni kavramlar meydana getiriler. Bilim adamları insan beyin hücrelerinden esinlendikleri için genel olarak insanlar üzerinden örneklemelerde bulunmuştur. TYSA'ları insan beyninden etkilenerek zamana yönelik tasarımlar yapmışlardır. Bu işlemleri geleneksel YSA'lar yapamadıkları için çok büyük bir eksiklik olarak görülmektedir. Mesela; bir filmdeki her noktada ne tür bir olayın meydana geldiğini sınıflandırmak istediğinde geleneksel bir YSA, filmdeki önceki olaylarla daha sonra gelişen olaylar arasındaki bağlantıyı mantıklı bir şekilde kuramazlar. Fakat TYSA bu problemi nasıl ele alacağını bilmektedir. TYSA'lar zamana bağlı döngüleri olduğundan bilginin devam etmesine izin veren ağlardır. Şekil 2.12'de gösterilen şemada TYSA'nın döngülerinin var olduğu gösterilmektedir.



Şekil 2.12. TYSA yapısı (Olah, 2015)

Yukarıdaki şemada, A sinir ağının bir kısmı, bazı girdilere  $x_t$  bakar ve bir  $h_t$  değeri çıkarır. Bir döngü, ağın bir adımından bir sonraki adıma geçilmesine izin verir. Bu döngüler tekrar eden sinir ağlarının gizemli gibi görünmesini sağlamaktadır.

Tekrarlayan bir yapay sinir ağı, aynı ağın birden fazla kopyası olarak düşünülür ve her bir ağ diğer bir ağa mesaj gönderir.



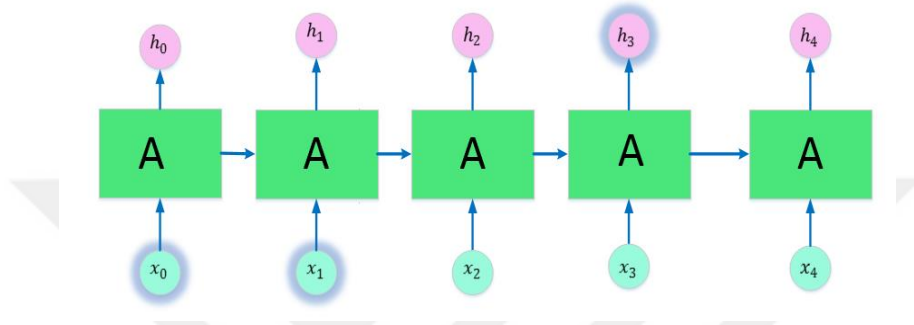
Şekil 2.13. TYSA modelinin zamana bağlı olarak açılımı (Olah, 2015)

Bu zincir benzeri yapı TYSA'larının diziler ve listelerle yakından alakalı olduğunu ortaya koymaktadır.

### 2.3.5.1 Uzun Vadeli Bağımlılık Sorunu

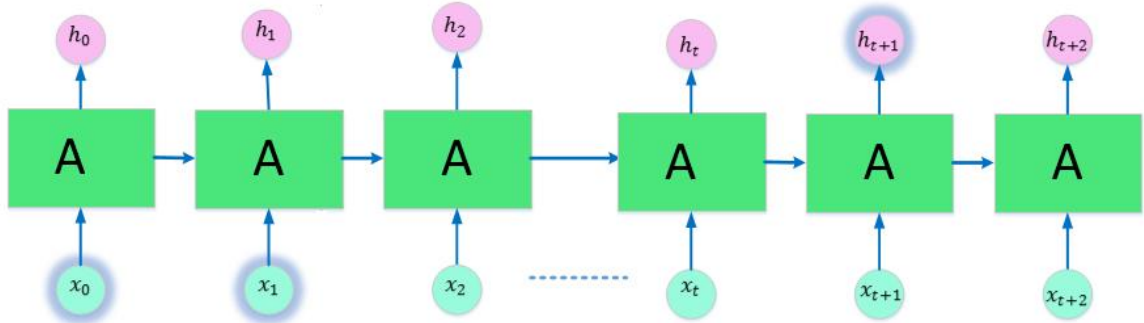
TYSA'lar geçmiş bilgileri kullanıp gelecekle ilgili tahminlerde bulunurlar. Fakat TYSA'ların en önemli sıkıntılardan birisi ileriki örneklerde belirtildiği gibi uzun vadeli tahminlerde bulunamamalarıdır.

Örneğin, herhangi bir kelimeyi temel alarak bir sonraki kelimeyi tahmin etmeye çalışan bir dil modeli tasarlanmıştır. "Bulutlar gökyüzünde" ifadesinin son kelimesi tahmin edildiğinde, daha fazla bağlama ihtiyacı duyulmaksızın, bir sonraki kelimenin gökyüzü olacağı açıkça bilinmektedir. Bu gibi durumlar, konu ile ilgili bilgi ve ihtiyaç duyulan yer arasındaki mesafenin az olduğundan, TYSA'lar geçmiş bilgileri kullanmayı öğrenmişlerdir. Şekil 2.14'te gösterilen şemada kısa mesafedeki TYSA modelinin dağılımı gösterilmektedir.



Şekil 2.14. Kısa mesafelerde tahmin yapan TYSA'nın açılımı

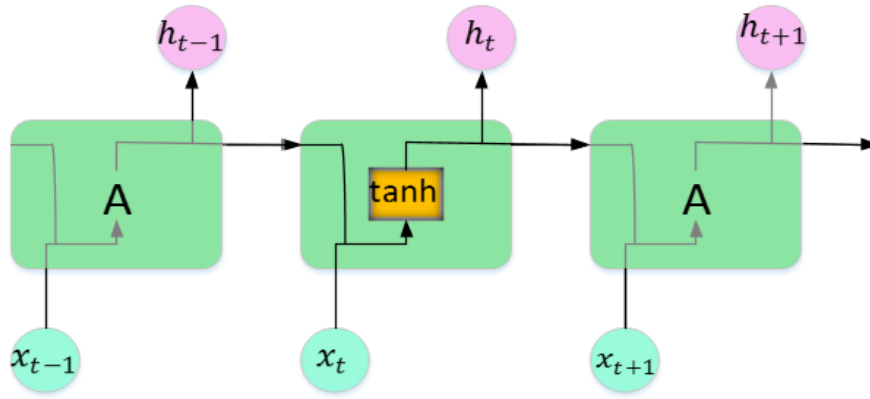
Fakat daha fazla bağlantıya ihtiyaç duyulduğu durumlar da vardır. "Fransa'da büyüdüm ... akıcı Fransızca konuşuyorum" metninde geçen kelime önceden tahmin edilmeye çalışıldığında, yeni bilgilerin bir sonraki kelimenin tahmini olarak bir dilin adı olduğu öngörülmüştür. Daha sonra eğer o dili daraltmak istendiğinde arka arkaya Fransa bağlamına ihtiyaç duyulmaktadır. Konu ile ilgili bilgi ve oluşan yeni bilginin arasındaki mesafe arttıkça, TYSA'ların bilgiyi bağlamayı öğrenmeleri oldukça zordur. Şekil 2.15'te gösterilen TYSA modelinin uzun mesafedeki ağ açılımı verilmiştir.



Şekil 2.15. Uzun mesafelerde tahmin yapan TYSA'nın açılımı

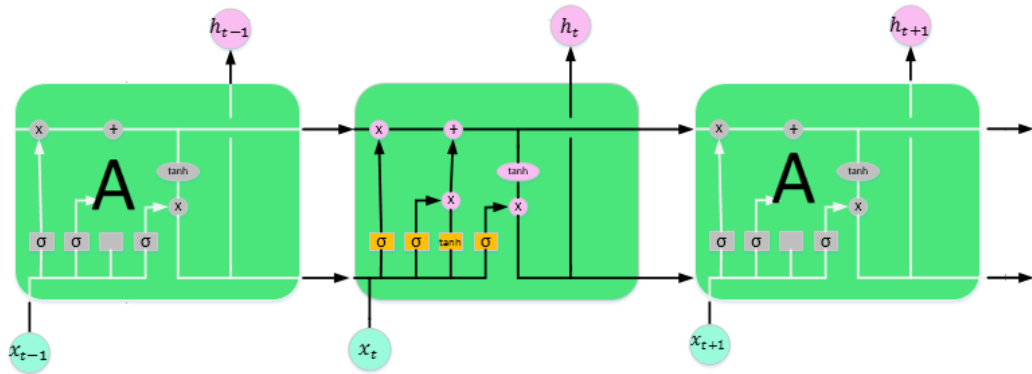
Uzun Kısa Vadeli Hafıza (UKVH) Ağları genellikle literatürde "UKVH" olarak adlandırılır. Uzun vadeli bağımlılıkları öğrenebilen özel bir TYSA türüdür. Bunlar ilk olarak

Hochreiter ve Schmidhuber (1997) tarafından tanıtılmıştır. Çok çeşitli problemler üzerinden muazzam derecede çalışırlar ve yaygın olarak kullanılmaktadırlar. UKVH'lar, uzun vadeli bağımlılık sorununun önüne geçmek için tasarlanmıştır. Uzun süre bilgi hatırlamak pratikte varsayılan davranışlardır, öğrenmek için mücadele ettikleri bir şey değildir. Tüm tekrarlı sinir ağları, modül zinciri biçimindedir. Şekil 2.16'da de gösterildiği gibi standart TYSA'larda bu yinelenen modül, tek bir *tanh* katmanı gibi çok basit bir yapıya sahiptirler.

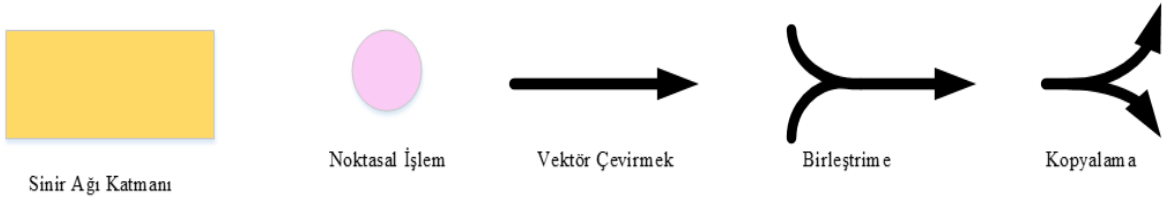


Şekil 2.16. Standart bir TYSA'ları tek bir katman içerir (Olah, 2015)

UKVH'ların da bu zincir benzeri yapıları vardır ancak yinelenen modül farklı bir yapıya sahiptir. Buda Şekil 2.17'de gösterildiği gibi tek bir sinir ağı katmanı yerine, çok özel bir şekilde etkileşen dört katman vardır.



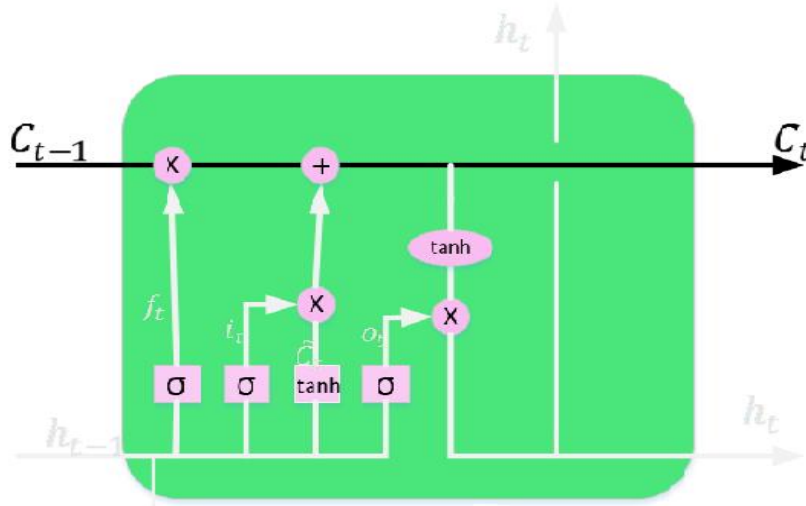
Şekil 2.17. Dört katmanlı UKVH ağı modeli (Olah, 2015)



Yukardaki diyagramda, her satır, bir düğümün çıktısından başkalarının girişlerine kadar tüm vektörü taşır. Sarı kutular sinir ağı katmanları ve pembe daireler vektör eklenmesi gibi noktasal işlemleri temsil eder. Birleştirilen satırlar birleştirme hattı, çizgi çatal çizmesi, içeriğinin kopyalanması ve kopyaların farklı yerlere gideceği anlamına gelmektedir.

### 2.3.5.2. Uzun Kısa Vadeli Hafıza Ağların Yapısı

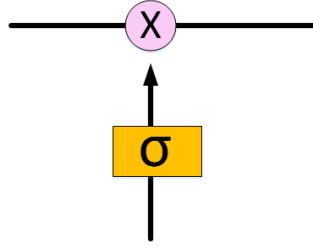
UKVH'ların anahtarı hücre durumudur, Şekil 2.18'de gösterildiği gibi diyagramın üstünden geçen yatay çizgidir. Hücre hali bir çeşit konveyör kemeri gibidir. Yalnızca bazı küçük doğrusal etkileşimlerle, zincirin tamamı boyunca düzleşir. Bilginin değişmeden akması çok kolaydır.



Şekil 2.18. UKVH ağ hücresi (Olah, 2015)

UKVH, kapılar denilen yapılarlar özenle düzenlenmiş hücre durumuna bilgi ekleme veya kaldırma kabiliyetine sahiptir.

Kapılar, isteğe bağlı olarak bilginin geçmesine izin vermek için bir yoldur. Şekil 2.19'da gösterilen kapı, sigmoid sinir ağı katmanı ve noktasal çarpma işleminden oluşur.



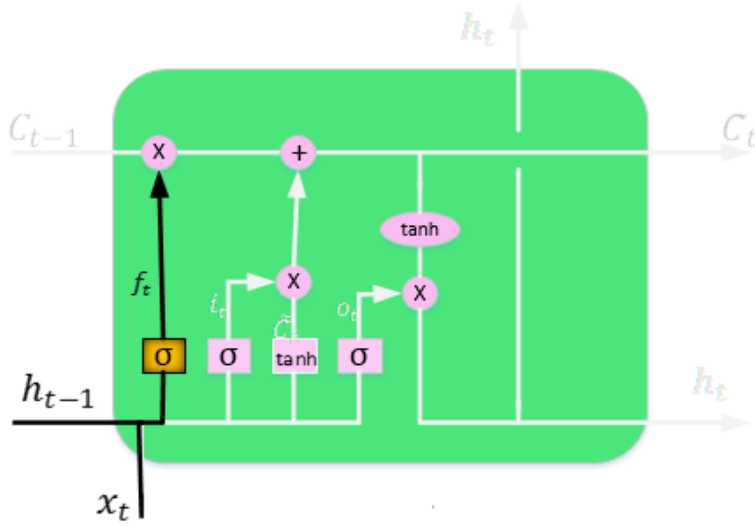
Şekil 2.19. UKVH ağ modelinin kapı gösterimi (Olah, 2015)

Sigmoid katman, sıfırdan bire kadar sayıları çıkartır ve her bileşenin ne kadarına izin verileceğini açıklar. Sıfır değeri “hiçbir şeyin geçmesine izin vermeyin” anlamına gelir; buna karşın “1” değeri “her şeye izin ver” anlamına gelmektedir. UKVH, hücre durumunu korumak ve kontrol etmek için bu kapılardan üçüne sahiptir.

### 2.3.5.3. Uzun Kısa Vadeli Hafıza Ağlarda Adımlar

UKVH’ın ilk adımı, hangi bilginin hücre durumundan atılacağına karar vermektir. Bu karar, "Unut kapısı " olarak adlandırılan bir sigmoid tabaka tarafından yapılmıştır. Şekil 2.20’de gösterilen diyagramda da  $h_{t-1}$  ve  $x_t$ 'ye bakar ve hücre durumu  $C_{t-1}$ 'deki her sayı için 0 ile 1 arasında bir sayı çıkarmaktadır. 1, "tamamen koru" ifadesini temsil ederken 0, "tamamen bundan kurtul" anlamına gelmektedir.

Bir sonraki kelimeyi önceki tüm kelimelere bağlı olarak tahmin etmeye çalışan bir dil probleminde, hücre hali, doğru zamirlerin kullanılabilmesi için mevcut öznenin cinsini içerebilmiştir. Yeni bir konu gördüğünde, eski konunun cinsini unutmak istemektedir.

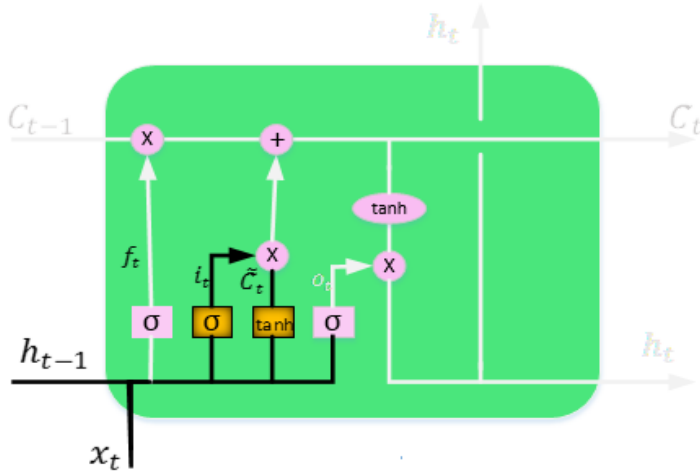


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.13)$$

Şekil 2.20. UKVH ağ hücresi (Olah, 2015)

Bir sonraki adım, Şekil 2.21’de da gösterildiği gibi hücre durumuna hangi yeni bilgiyi depolayacağımıza karar vermektedir. Bu iki bölümden oluşmaktadır. İlk olarak, "giriş kapısı" olarak adlandırılan bir sigmoid katman hangi değerlerin güncelleneceğine karar vermektedir. Daha sonra, bir *tanh* katmanı, mevcut duruma eklenebilecek yeni aday değerler  $\tilde{C}_t$  vektörü oluşmaktadır. Bir sonraki adımda ise, hücre durumunu güncelleştirmek için bu iki katmanı birleştirir.

Dil modelimiz örneğinde, yeni konunun cinsini belirlemek için hücre durumundaki unutulmuş eski bilgilerin yerine yeni gelen bilgilerin eklenmesi gerekmektedir.



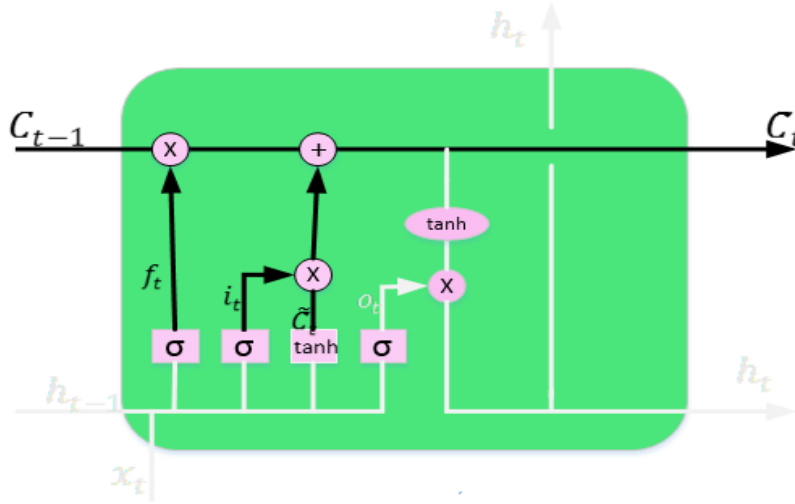
Şekil 2.21. UKVH ağ hücresi (Olah, 2015)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.14)$$

$$C_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.15)$$

Şekil 2.22’de gösterildiği gibi eski hücre durumunu  $C_{t-1}$ ’i yeni hücre durumu  $C_t$ ’ye güncelleme yapılmıştır. Daha önceki adımların hangi işlemlerden geçebileceği belirlenmiştir. Daha önce unutmaya dair verdiği kararları uygulayıp eski durumu,  $f_t$  ile çarpmıştır. Sonra  $i_t * \tilde{C}_t$  eklenmiştir. Bu, her durum değerini ne kadar güncellemeye karar verdiğinde ağırlık ölçtüğü değerlerin yeni aday değerleri ile değiştirmesidir.

Dil modeli öğreneğinden yola çıktığımızda, bu hücrede eski konunun cinsi hakkındaki bilgileri bırakıp önceki adımlarda kararlaştırıldığı gibi yeni bilgileri eklediği yerde gerçekleşmektedir.



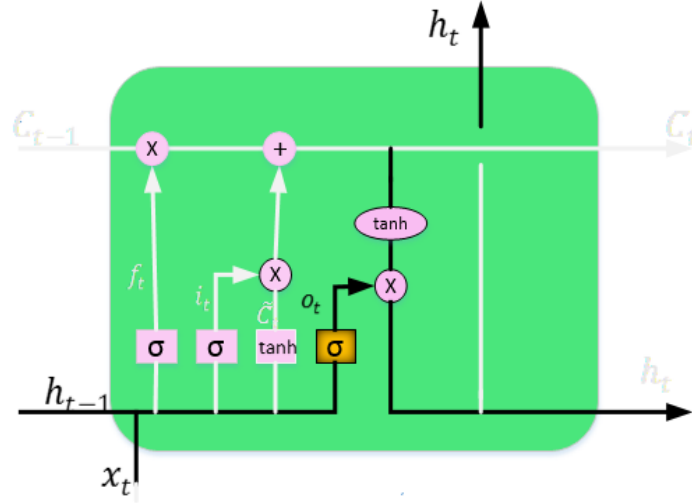
Şekil 2.22. UKVH ağ hücresi (Olah, 2015)

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.16)$$

Son olarak, Şekil 2.23’te gösterildiği gibi ağ, nasıl sonuç vereceğine bu katmanda karar vermektedir. Bu çıktı hücre duruma göre olmuştur, ancak filtrelenmiş bir sürüm olarak devreye girilmiştir. İlk olarak, hücre durumunun hangi bölümlerine çıktı çıkarıldığına karar veren bir sigmoid katmanı çalıştırılmıştır. Daha sonra hücrenin durumunu  $\tanh$  ile (-1 ile 1 arasında olacak şekilde) işleme koyduktan sonra sigmoid kapısının çıktısıyla çarpılmıştır ve kararlaştırdığı parçaları çıktı olarak vermiştir.

Dil modeli örneği için, sadece bir konu gördüğü için, bir sonraki fiilin gelmesi durumunda, fiil ile ilgili bilgileri çıkarmak isteyebilir. Örneğin, öznenin tekil mi yoksa çoğul

mu olduğunu, yani bir fiilin nasıl bir formun konması gerektiğini bilinmektedir. (Olah, 2015).



Şekil 2.23. UKVH ağ hücresi (Olah, 2015)

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.17)$$

$$h_t = o_t * \tanh(C_t) \quad (2.18)$$

### 2.3.6. Çift Yönlü Tekrarlı Yapay Sinir Ağlar

En çok kullanılan TYSA mimarilerinden biri olan UKVH'nın yanında, Schuster ve Paliwal (1997) tarafından tanımlanan çift yönlü tekrarlayan sinir ağıdır (ÇYTSA). Bu mimaride iki gizli düğüm katmanı vardır. Her iki katman da girdi ve çıktıya bağlıdır. İki gizli katman, diğeri geçmiş zaman adımlarından birincil bağlantılara sahipken diğeri bağlantıların yinelenen yönü çevrilerek sırayla birlikte geriye doğru aktive olması şeklinde tanımlanır. Bir girdi dizisi ve bir hedef dizisi göz önüne alındığında, ÇYTSA zamana karşı açıldıktan sonra sıradan geri yayılımla eğitilebilir. Aşağıdaki üç denklem bir ÇYTSA'larını tanımlamaktadır.

$$h^{(t)} = \sigma(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h) \quad (2.19)$$

$$z^{(t)} = \sigma(W^{hx}x^{(t)} + W^{zz}z^{(t+1)} + b_z) \quad (2.20)$$

$$y^{(t)} = \text{softmax}(W^{yh}h^{(t)} + W^{yz}z^{(t)} + b_y) \quad (2.21)$$

Burada  $h^{(t)}$  ve  $z^{(t)}$  sırasıyla ileri ve geriye doğru gizli katmanların değerleridir.

ÇYTSA'nın bir sınırlaması hem gelecekte hem de geçmişte sabit bir son nokta gerektirdiğinden sürekli olarak çalışamaz. Ayrıca, çevrimiçi ayar için uygun bir makine

öğrenme algoritması değildir, çünkü gelecekte bilgi almak, yani gözlemlenmeyen dizi öğelerini bilmek imkânsızdır. Fakat sabit uzunluk dizisi üzerinde tahmin yapmak için, geçmiş ve gelecek sıra unsurlarını hesaba katmak mantıklı olmuştur. Konuşma parçasının etiketlenmesinin doğal dili görevini göz önünde bulundurulur. Cümledeki herhangi bir kelimeyi verilirse, önündeki kelimelere ve kelimenin konuşma bölümünü tahmin etmede kullanışlı olan kelimelere ilişkin bilgiler bulunur (Olah, 2015).



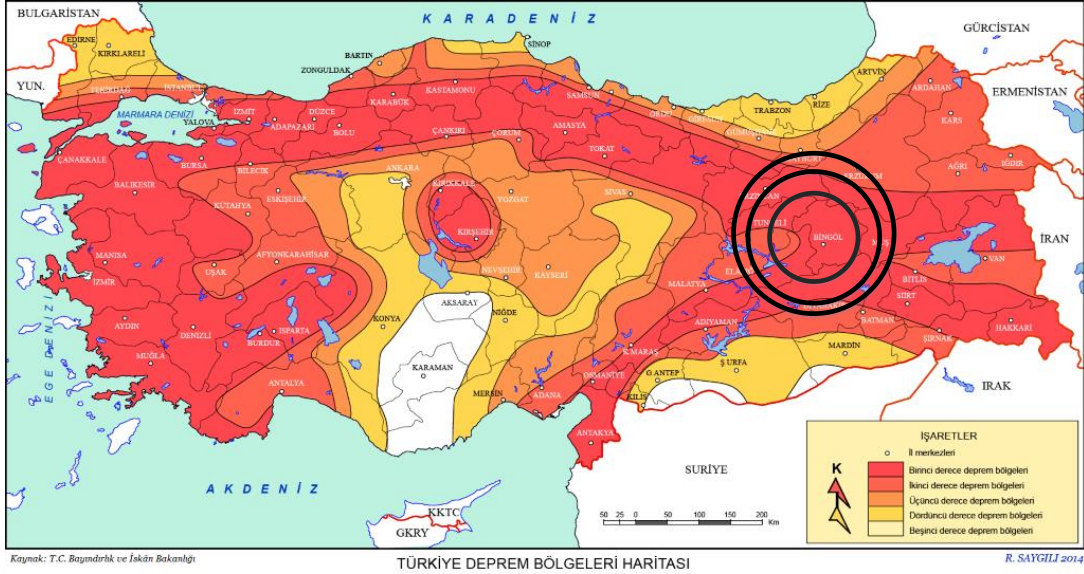
### **3. BULGULAR**

#### **3.1. Derin Uzun-Kısa Vadeli Hafıza Ağ Modelini Kullanarak Deprem Tahmini**

Bu bölümde, derin UKVH ağları kullanarak deprem tahmini ile ilgili çalışmanın detayları anlatılmıştır. İlk olarak Türkiye deprem haritası üzerinde çalışılan bölge ile ilgili bilgilere yer verilmiştir. Sonraki bölümlerde ise bölge üzerinde oluşan deprem verilerinin toplanması ve organizasyonu hakkında bilgiler sunulmuştur. Elde edilen verilerin yıllara, alan çaplarına göre ve depremin şiddetine göre sorgulanması ile bu verilerin düzenlenmesi hakkında bilgiler verilmiştir. Daha sonraki aşamalarda ise deprem tahmini için geliştirilen derin UKVH ağ modelinden bahsedilmiştir. Son olarak tahmin sonuçları grafikler ve verilerle sunularak tartışılmıştır.

##### **3.1.1. Deprem Verileri**

Türkiye’de deprem bölgelerinin; KAF, DAF ve BAF olarak ayrıldığından giriş bölümünde de bahsedilmişti. Şekil 3.1’de, Türkiye deprem haritası verilmiştir. Bu harita üzerindeki kırmızı bölgeler birinci dereceden deprem bölgelerini temsil etmektedir. Bu çalışmada, deprem fay hatlarından ziyade aktif bir deprem bölgesi olan Bingöl ili ve çevresi baz alınarak tahminlerin yapılması amaçlanmıştır. Bu bölgenin seçilme nedeni yoğun bir deprem hareketliliğine sahip olmasıdır. Giriş kısmında da bahsedildiği gibi bu bölge geçmişte birçok büyük depreme maruz kalmıştır. Tez çalışması için Bingöl ili merkez nokta olarak belirlenmiş ve farklı yarıçap içerisinde kalan deprem verileri sorgulanmıştır. Verilerin elde edilmesinde, B.Ü. Kandilli Rasathanesi BDTİM deprem sorgulama sisteminden yararlanılmıştır (URL2, 2018). Bu sistem üzerinde dairesel arama seçilerek sorgulanmak istenen bölgenin enlem ve boylam değerleri girildikten sonra hangi zaman aralığında oluşan depremlerin sorgulanacağı bilgisi girilmektedir. Sistem üzerindeki bir diğer sorgu parametresi ise, yarıçap bilgisidir. Belirtilen yarıçap (km cinsinden) parametresi ile sorgulanmak istenen bölgenin sınırları belirtilmektedir. Ayrıca deprem sorgulamasındaki en önemli parametrelerden birisi deprem şiddet aralığıdır. Böylece istenilen bölge, zaman aralığı ve şiddet aralığındaki depremler kolayca sorgulanabilmektedir.



Şekil 3.1. Türkiye deprem haritası (URL3, 2018)

Şekil 3.2’de, deprem sorgulama sistemi üzerinde sorgu parametrelerinin belirtildiği bir ekran görüntüsü verilmiştir. Bu ekran görüntüsünde, 1950 ile 1970 yılları arasında şiddet değeri 2.0 ile 9.0 arasındaki depremler belirtilen bir merkezin etrafında 50km yarıçapında dairesel bölge içerisindeki depremleri sorgulayacak şekildedir.

B.Ü. Kandilli Rasathanesi BDTİM Deprem Sorgulama Sistemi

Koordinata göre **Dairesel arama** Sarsıntı türü **Hepsi** EN

Başlama **1950** **01** **01** Bitiş **1970** **01** **01** Tarih Aralığı

Enlem **39,4** Boylam **40,7696** Yarıçap **50**

Büyüklik **2** ≤ M ≤ **9.0**

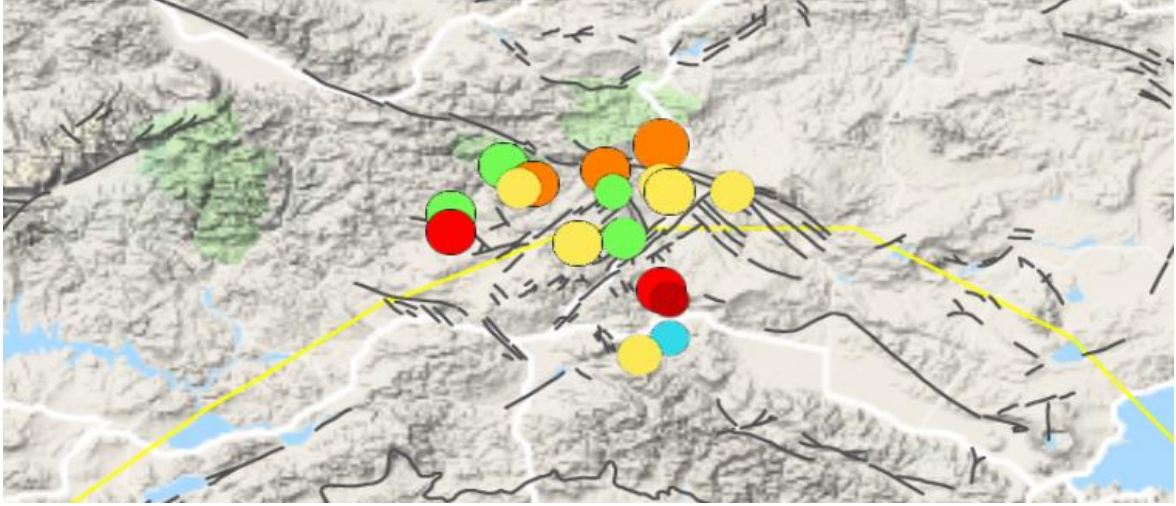
Derinlik **0** ≤ d ≤ **500**

**B.Ü. KRDAE BÖLGESEL DEPREM-TSUNAMİ İZLEME VE DEĞERLENDİRME MERKEZİ**

**Yeni arama** **Harita**

Şekil 3.2. B. Ü Kandilli Rasathanesi BDTİM Deprem Sorgulama Sistemi

İkinci aşama olarak, sorgu sonucunda çıkan veriler Şekil 3.3’teki gibi harita üzerinde veya Şekil 3.4’teki gibi liste halinde elde edilebilmektedir. Daha sonraki aşamalarda ise elde edilen ham veriler belirlenen formatlarda düzenlenerek giriş verisi haline getirilmiştir.



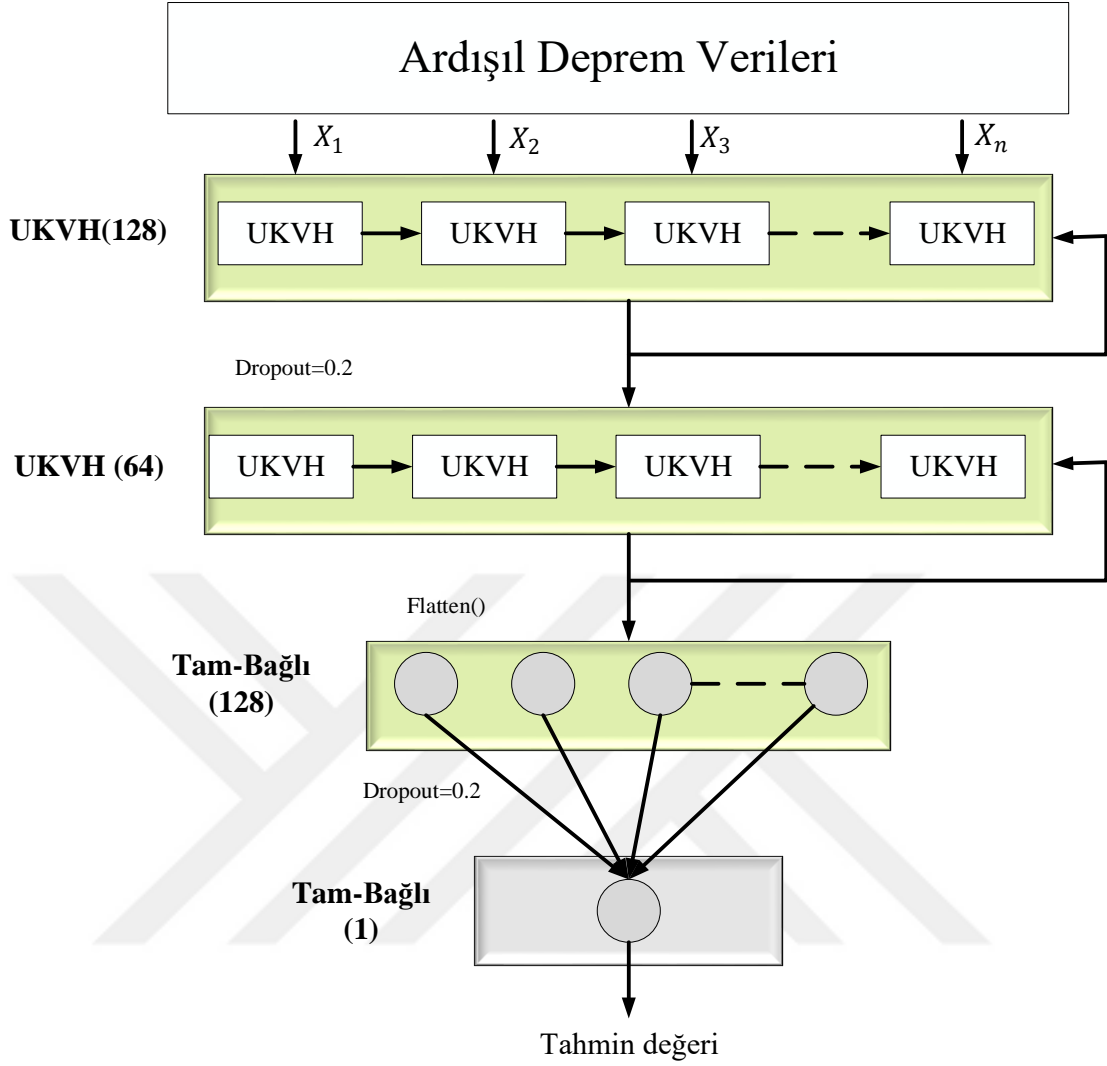
Şekil 3.3. 1950 ve 1970 yılları arası 50km yarıçap için meydana gelen deprem haritası

No	Deprem Kodu	Olus tarihi	Olus zamani	Enlem	Boylam	Der(km)	xM	MD	ML	Mw	Ms	Mb	Tip	Yer
000001	19691001203337	1969.10.01	20:33:37.00	39.3200	40.5600	017.0	5.0	4.7	4.7	5.0	4.7	4.7	Ke	CATMAOLUK-ADAKLI (BINGOL) [North West 5.3 km]
000002	19690723025411	1969.07.23	02:54:11.00	38.9000	41.0000	169.0	4.4	4.2	4.2	4.4	4.0	4.3	Ke	MURATKOY-SOLHAN (BINGOL) [North East 2.5 km]
000003	19680925205215	1968.09.25	20:52:15.80	39.2400	40.2900	041.0	5.3	5.0	5.0	5.3	5.1	5.0	Ke	DALLICA-KIGI (BINGOL) [South West 1.5 km]
000004	19680924041953	1968.09.24	04:19:53.00	39.1900	40.2900	008.0	5.3	5.0	5.0	5.3	5.1	5.0	Ke	DARKOPRU-KIGI (BINGOL) [West 0.4 km]
000005	19670211041901	1967.02.11	04:19:01.00	39.0000	41.0000	005.0	4.0	4.0	0.0	0.0	0.0	0.0	Ke	YIGITHARMANI-SOLHAN (BINGOL) [East 2.7 km]
000006	19661226042101	1966.12.26	04:21:01.00	38.8500	40.9000	028.0	4.4	4.4	0.0	0.0	0.0	0.0	Ke	GELINTEPE-SOLHAN (BINGOL) [West 1.9 km]
000007	19661226042001	1966.12.26	04:20:01.00	38.8500	40.9000	028.0	4.8	4.6	4.6	4.8	4.5	4.7	Ke	GELINTEPE-SOLHAN (BINGOL) [West 1.9 km]
000008	19660913202351	1966.09.13	20:23:51.00	39.1700	40.8500	046.0	4.8	4.7	4.6	4.8	4.6	4.5	Ke	DEVECİK-KARLIOVA (BINGOL) [North West 1.0 km]
000009	19660823013545	1966.08.23	01:35:45.00	39.3200	40.9700	030.0	4.7	4.5	4.5	4.7	4.4	4.6	Ke	KARLIOVA (BINGOL) [North West 4.5 km]
000010	19660820175408	1966.08.20	17:54:08.60	39.3000	40.8200	070.0	4.3	4.1	4.1	4.3	3.9	4.2	Ke	KARLICA-KARLIOVA (BINGOL) [North East 9.7 km]
000011	19660820151734	1966.08.20	15:17:34.70	39.3100	40.5100	034.0	4.8	4.7	4.6	4.8	4.6	4.5	Ke	AYSAKLI-ADAKLI (BINGOL) [East 4.8 km]
000012	19660820120143	1966.08.20	12:01:43.70	39.1600	40.7000	033.0	5.7	5.4	5.4	5.7	5.6	5.4	Ke	KIRKPINAR-ADAKLI (BINGOL) [South West 3.7 km]
000013	19660820115909	1966.08.20	11:59:09.00	39.4200	40.9800	014.0	6.0	5.8	5.7	5.6	6.0	5.3	Ke	KASIKCI-KARLIOVA (BINGOL) [North West 2.3 km]
000014	19650831072946	1965.08.31	07:29:46.90	39.3600	40.7900	011.0	5.5	5.4	5.3	5.4	5.5	5.1	Ke	KAYNARPINAR-KARLIOVA (BINGOL) [South East 3.1 km]
000015	19650831055703	1965.08.31	05:57:03.80	39.3000	41.2000	033.0	4.8	4.6	4.6	4.8	4.5	4.5	Ke	KARTALDERE-VARTO (MUS) [North West 1.1 km]
000016	19570707055858	1957.07.07	05:58:58.20	39.3700	40.4600	060.0	5.3	5.0	5.0	5.3	5.1	5.1	Ke	CEVIZLI-ADAKLI (BINGOL) [North East 6.2 km]
000017	19540328044752	1954.03.28	04:47:52.30	39.0300	40.9700	010.0	5.5	5.3	5.3	5.5	5.4	5.3	Ke	YIGITHARMANI-SOLHAN (BINGOL) [North 3.4 km]
000018	19500102211500	1950.01.02	21:15:00.00	39.3000	41.0000	030.0	5.2	4.9	4.9	5.2	4.9	4.9	Ke	KARLIOVA (BINGOL) [North West 1.3 km]

Şekil 3.4. 1950 ve 1970 yılları arası 50km yarıçap için meydana gelen deprem listesi

### 3.1.2. Önerilen Derin Uzun Kısa Vadeli Hafıza Modeli

Bu çalışmada, bölgesel deprem tahminlerini elde etmek için bir derin UKVH ağ modeli tasarlanmıştır. Şekil 3.5'te bu derin UKVH ağına ait blok diyagram gösterimi yapılmıştır.



Şekil 3.5. Deprem tahminleri için kullanılan derin UKVH ağ modeli

Ağ modeline 128'lik ve 64'lük iki adet UKVH katmanı, 128 birime sahip bir tam-bağlı katmanı ve çıkışı oluşturan bir tam-bağlı katmanı yer almaktadır. Derin sinir ağlarındaki en önemli zorluklardan biri de aşırı öğrenme problemidir. Aşırı öğrenme problemini önlemek için son zamanlardaki etkili çözümlerden biri dropout tekniğidir. Dropout tekniğinde, eğitim süreci sırasında birimler, belirtilen oranlarda rastgele ayrılırlar. Bu yöntem, birimlerin birbiriyle çok uyumlu olmasını önleyerek aşırı öğrenmeyi önemli ölçüde azaltır. Bu nedenle, tasarlanan ağlarda dropout fonksiyonu iki farklı katmana yerleştirilmiştir. Derin ağlarda katman parametrelerinin optimizasyonu uzun ve zaman alıcı bir süreçtir. Bu çalışmada, iki parametre dizisinin farklı kombinasyonları incelenmiş ve en yüksek kararlılığı elde etmek için doğrulama setinde uygulanmıştır. Bu diziler tam-bağlı katman birim sayısı={32, 64, 128,256}, UKVH katman sayısı={16, 32, 64, 128} ve

dropout parametreleri dizisi={0.1, 0.2, ..., 1} şeklindedir. Bu ağda, UKVH katmanı her giriş zaman adımı için gizli durum çıktısını geri döndürür. Bu nedenle, UKVH ve tam-bağlı katmanları arasına bir düzleştirici (flatten) katmanı yerleştirilmiştir. Bu katman çok boyutlu dizi yapısını vektör şekline dönüştürerek bir sonraki katmanın girişine uygun hale getirir. Doğrusal olmayan aktivasyon fonksiyonu, karmaşık problemlerin çözümünde önemli avantajlar sağlayan sinir ağlarının diğer önemli bileşenleridir. Aktivasyon fonksiyonlarda ReLU, softmax, tanh, Lineer vb. fonksiyonlar kullanılır. Bu modelde en iyi sonucu doğrusal fonksiyonda elde edildiği için aktivasyon fonksiyonu olarak lineer kullanılmıştır. Diğer aktivasyon fonksiyonlarının performansları belki giriş verileri normalize edilerek denenebilir.

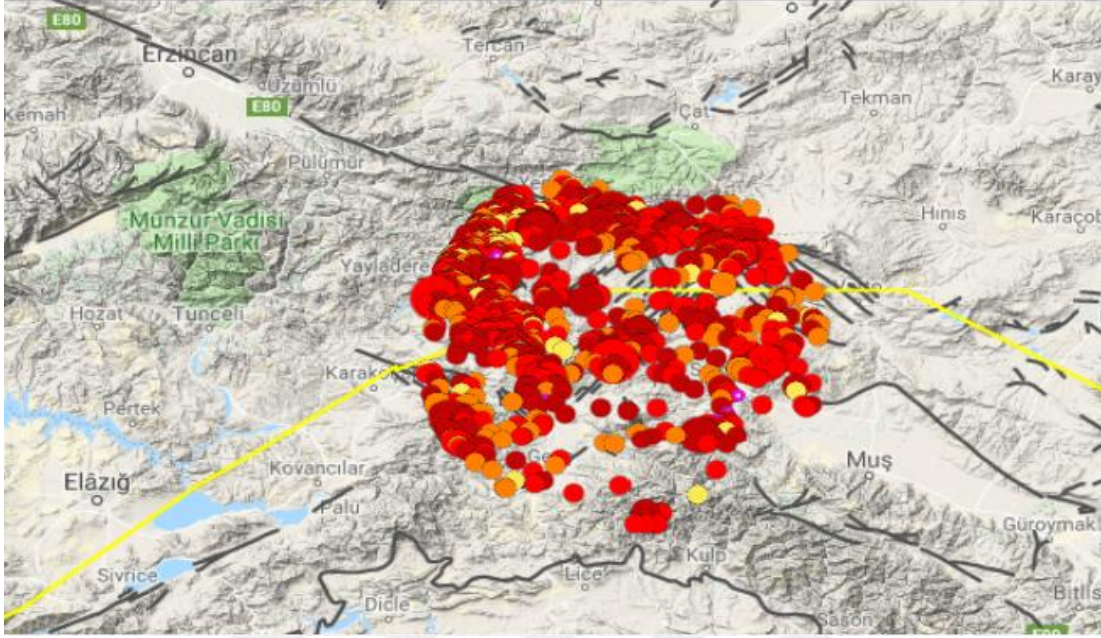
### **3.1.3. Deneysel Sonuçlar**

Deprem tahminleri için tez çalışmasında farklı senaryolar kullanılmıştır. Bu amaçla, belirlenen deprem bölgesi üzerinden sırasıyla 50km, 100km ve 200 km yarıçaplı alanlar için sorgular oluşturulmuştur. Belirlenen her yarıçap için belli yıl aralıklarında deprem verileri elde edilmiştir. Ayrıca tahmin için derin UKHV model parametrelerinden olan ardışıl boyutu (Sequence Length-SL) tahmin edilecek deprem için ne kadarlık zaman dilimini kullanacağını belirler. Örneğin SL parametresi 5 seçilirse her 5 deprem hareketliliğinden sonra oluşabilecek sonraki depremin şiddeti tahmin edilir. Kayan pencere mantığı ile bu süreç veriler üzerinde birer örnek kaydırılarak devam eder. Bu SL değerleri sırasıyla 5, 10, 25 ve 50 şeklinde belirlenip her bir veri seti üzerinde uygulanarak sonuçlar verilmiştir. Yapılan deneysel çalışmalarda, eğitim (train) aşamasında verilerin %90'ı kullanılmış ve geri kalan %10'u üzerinde ağın performansı ölçülmüştür. Eğitim verisi içerisindeki verilerin %30'u doğrulama (validation) aşaması için kullanılmıştır.

#### **3.1.3.1. Deneysel Çalışma 1: 50km Yarıçap**

Deneysel çalışmaların birincisinde, Bingöl il merkezinin dairesel olarak 50km yarıçapı içerisinde kalan bölge üzerinde tahmin çalışmaları yapılmıştır. Tahmin çalışmaları için iki farklı zaman aralığı belirlenmiştir. Bu zaman aralıkları, 2000- 2017 ve 1950 – 2017 olarak belirlenmiştir. Şekil 3.6'da 50 km yarıçaplı bölge için 2000-2010 yılları arasında

deprem şiddeti (Mc) 2.0 ve 9.0 arasındaki depremlerin harita üzerinde bir gösterimi sunulmuştur.



Şekil 3.6. 2000 ve 2010 arası 50km yarıçap için sorgu sonucu

50km yarıçap için belli yıllar arasında oluşan deprem hareketleri elde edilmiştir. Elde edilen hareketlilikler sıralı olarak düzenlenip derin UKVH ağ modeline uygulanmıştır. Bu model de tekrar adımı (epoch) 300 alınır, yığın boyutu (batch size) ise 16 olarak seçilir. Aynı zamanda her yıl aralığı için 5, 10, 25 ve 50 SL değerleri kullanılmıştır. Aşağıdaki Tablo 3.1.'de her yıl için kullandığımız toplam veri setleri, eğitim için kullanılan veri sayısı ve test için kullanılan veri sayısı verilmiştir.

Tablo 3.1.50 km yarıçap için yıllara göre veri sayıları

Yıllar	Toplam Veri Sayısı	Eğitim Veri Sayısı	Test Veri Sayısı
2000-2017	4178	3760	418
1950-2017	4266	3840	426

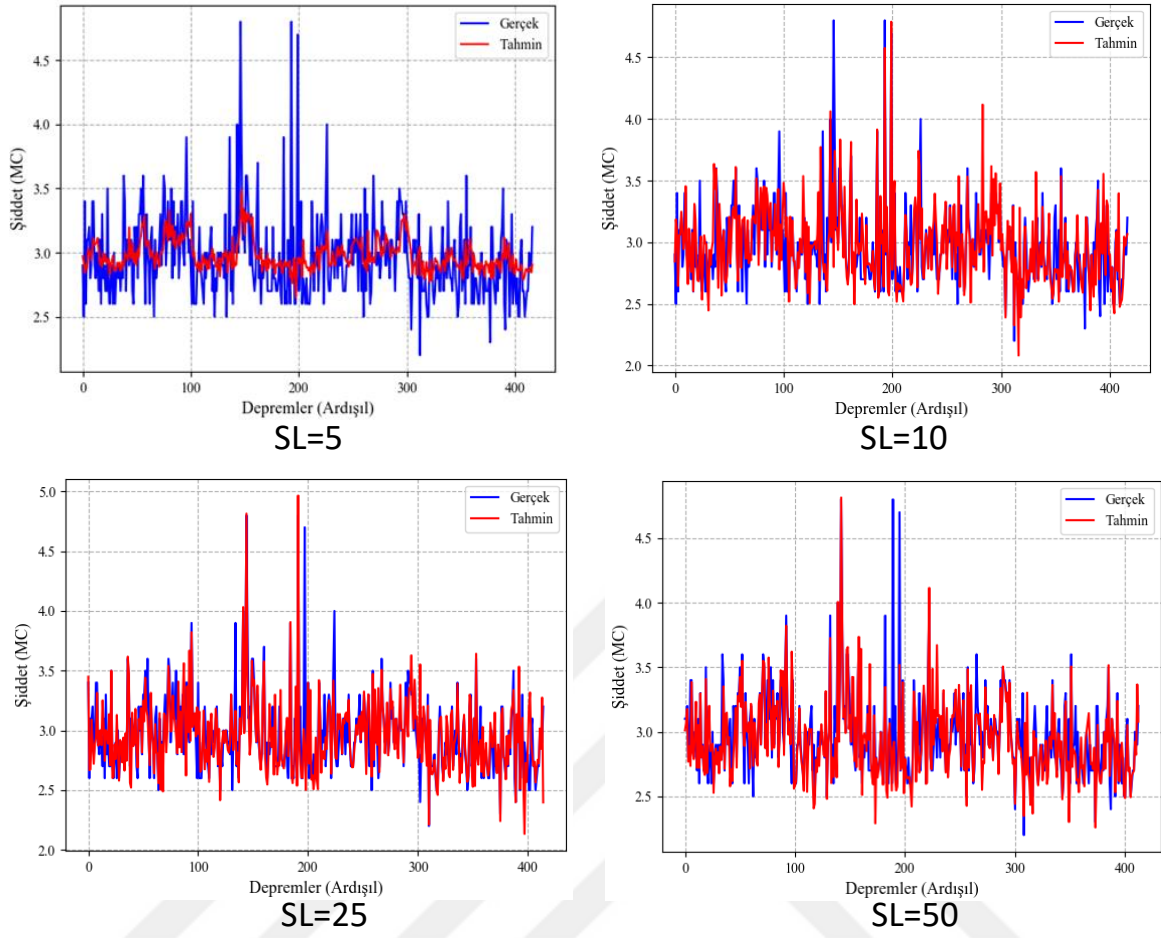
2000-2017 yılları arasındaki 50km yarıçap için yukarıdaki Tablo 3.1.'de belirtildiği gibi toplam verisayısının 4178, eğitim için kullanılan verisayısının 3760, test için kullanılan

verisayının 418 olduğu gösterilmiştir. Yukarıda da bahsedildiği gibi derin UKVH ağ modelinde her bir SL için 300 aşamada kullanılmıştır. Bu 300 aşama süresince eğitilmiş her bir SL için UKVH modeline ait kayıp (loss) fonksiyonun son beş değeri Tablo 3.2’de verilmiştir.

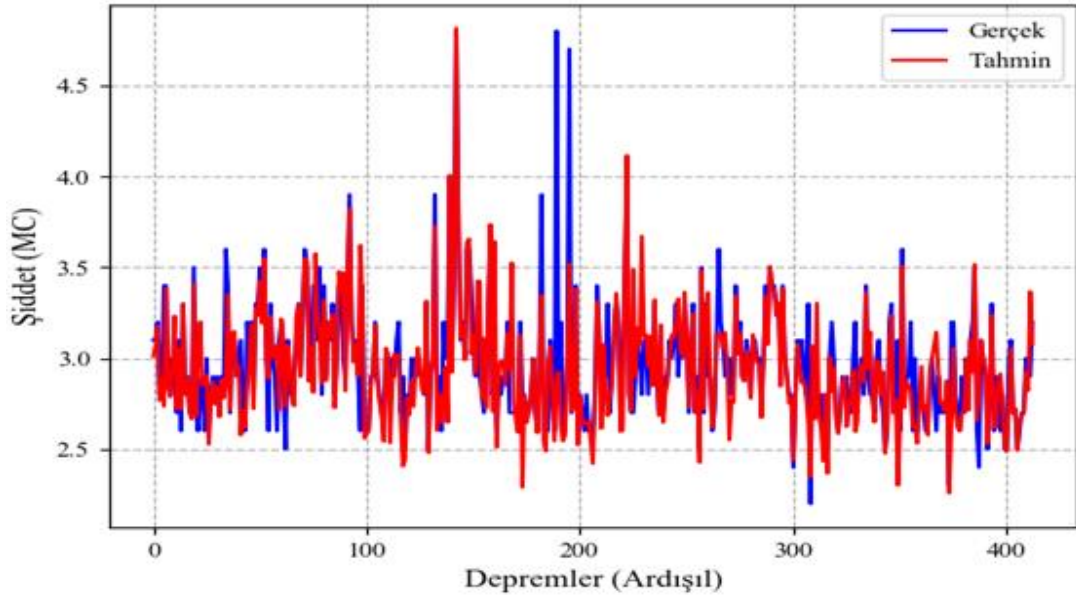
**Tablo 3.2.** 2000-2017 yılları arası 50km yarıçap için her SL için kayıplar

Kayıplar(Loss)	SL=5	SL=10	SL=25	SL=50
<b>Kayıp 1</b>	0.1653	0.1640	0.2001	0.1467
<b>Kayıp 2</b>	0.1757	0.1727	0.2016	0.1501
<b>Kayıp 3</b>	0.1685	0.1715	0.2056	0.1454
<b>Kayıp 4</b>	0.1696	0.1713	0.2003	0.1369
<b>Kayıp 5</b>	0.1607	0.1716	0.1989	0.1489

Aşağıdaki Şekil 3.7’deki grafikler 2000-2017 arasındaki 50km yarıçap için gerçek deprem değerleri ile UKVH ağ modelinin tahmin ettiği değerler verilmiştir. UKVH ağ modelinde ki her bir SL değeri için bir grafik oluşmuştur. Bunlar sırasıyla 5, 10, 25 ve 50 olarak belirlenmiştir. Gösterildiği gibi her SL için farklı sonuçlar elde edilmiştir. Bu grafiklerdeki depremlerin iniş ve çıkışlarının başarılı bir şekilde tahmin edildiği gözlemlenmiştir. Şekil 3.8’de SL değerinin 50 olarak belirlendiği derin UKVH ağının sonuçlarının detaylı gösterimine yer verilmiştir. Grafikte görüleceği gibi anlamlı bir tahmin eğrisi elde edilmiştir. Fakat yüksek şiddetli depremlerin tahminini pek başarılı değildir. Diğer grafiklerde anlık yüksek şiddetli depremler daha iyi bir şekilde tahmin edilmiştir.



Şekil 3.7. 2000-2017 yılları arası 50km yarıçap için farklı SL olan deprem tahminleri



Şekil 3.8. 2000-2017 yılları arası 50km yarıçap için SL=50 olan grafik

Tablo 3.3'te, SL 50 değeri için 2000 ile 2017 yılları arasında 50km yarıçap için gerçek deprem verileri ile derin UKVH modelinin tahmin ettiği deprem verilerinden ilk 30'unun değerleri gösterilmiştir. Tahmin edilen ve gerçek depremler arasındaki düşük farklılık dikkat çekmektedir.

**Tablo 3.3.** 2000-2017 yılları arasında SL=50 için gerçek ve tahmin edilen ilk 30 deprem şiddeti.

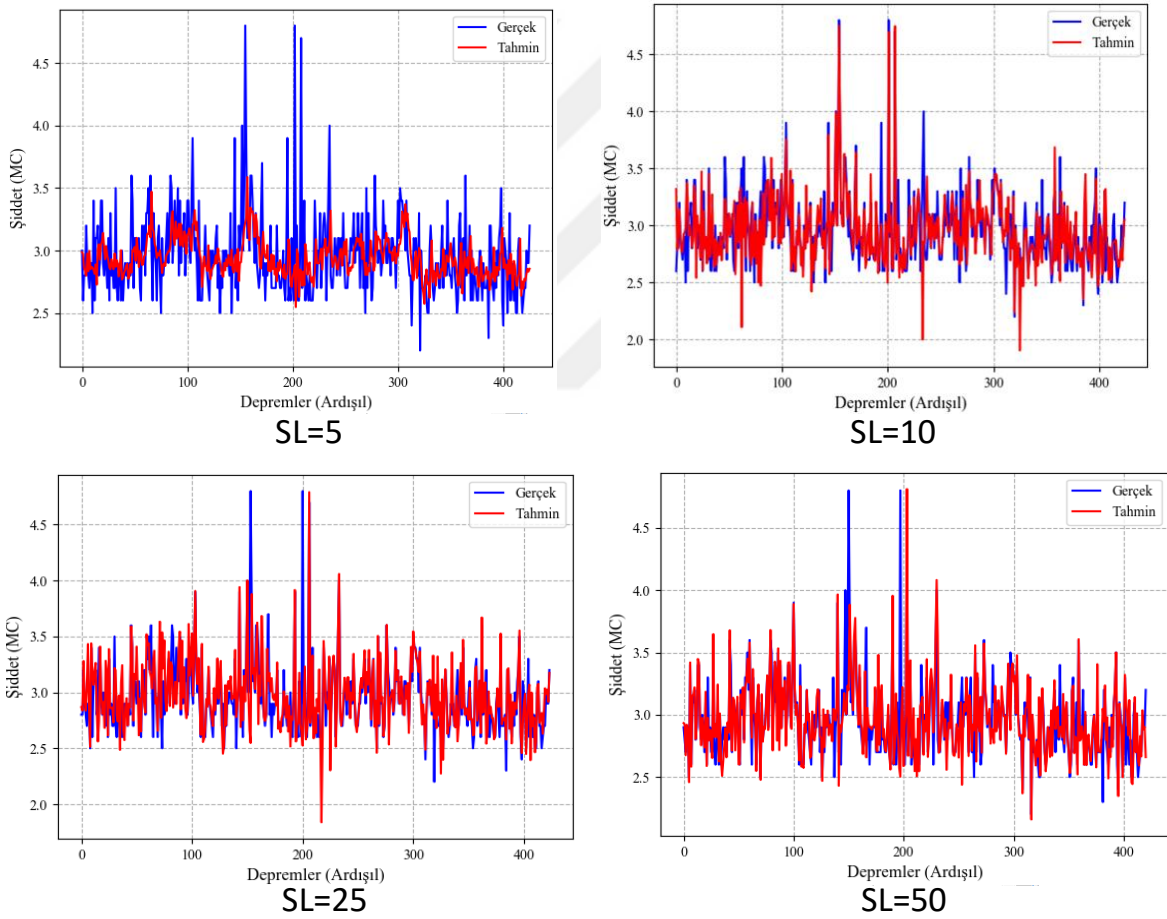
No	Gerçek	Tahmin	No	Gerçek	Tahmin	No	Gerçek	Tahmin
1	3.1	3.010	11	2.9	3.232	21	2.7	2.769
2	3.1	3.064	12	2.7	2.720	22	2.6	3.038
3	3.2	3.176	13	3.1	3.051	23	2.8	3.201
4	2.8	2.768	14	2.6	2.720	24	2.9	2.849
5	2.9	2.888	15	3.3	3.297	25	2.6	2.863
6	3.4	2.735	16	2.8	2.801	26	3	2.907
7	3.4	3.384	17	2.9	2.887	27	2.6	2.527
8	2.9	2.840	18	2.7	2.702	28	2.8	2.794
9	2.8	2.788	19	2.7	2.666	29	2.9	2.869
10	3	2.991	20	3.5	3.408	30	2.8	2.749

1950-2017 yılları arasındaki 50km yarıçap için yukarıdaki Tablo 3.1'de belirtildiği gibi toplam verisayısının 4266, eğitim için kullanılan verisayının 3840, test için kullanılan verisayının 426 olduğu gösterilmiştir. Yukarıda da bahsedildiği gibi derin UKVH ağ modelinde her bir SL için 300 aşama kullanılmıştır. Bu 300 aşama süresince eğitilmiş her bir SL için UKVH modeline ait kayıp(loss) fonksiyonunun son beş değeri Tablo 3.4'de verilmiştir.

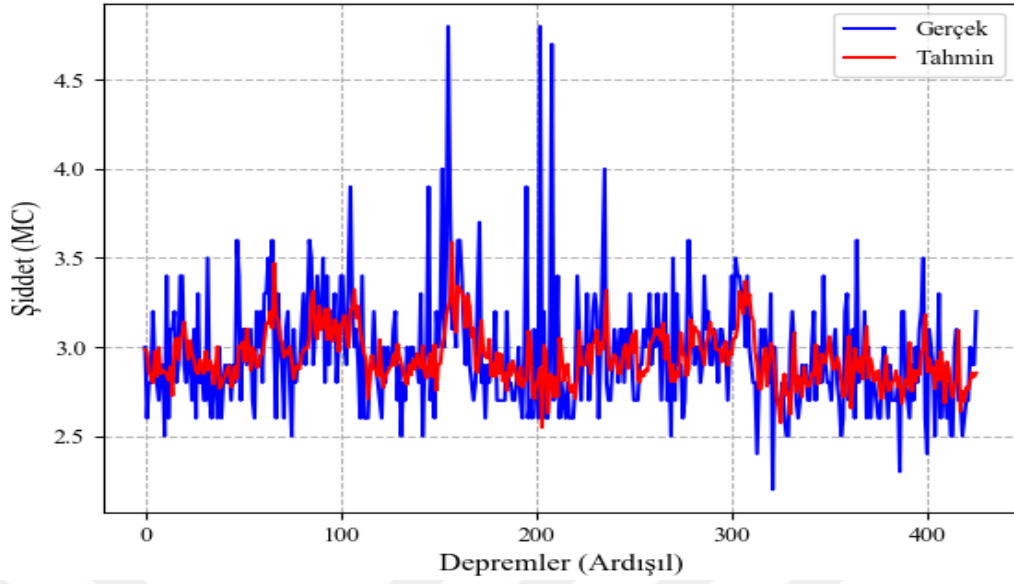
**Tablo 3.4.** 1950-2017 yılları arası 50km yarıçap için her SL için kayıplar

Kayıplar(Loss)	SL=5	SL=10	SL=25	SL=50
<b>Kayıp 1</b>	0.1543	0.1908	0.1651	0.1671
<b>Kayıp 2</b>	0.1607	0.1847	0.1603	0.1664
<b>Kayıp 3</b>	0.1583	0.1831	0.1616	0.1706
<b>Kayıp 4</b>	0.1577	0.1792	0.1625	0.1668
<b>Kayıp 5</b>	0.1616	0.1815	0.1629	0.1706

Aşağıdaki Şekil 3.9'daki grafikler 1950-2017 arasındaki 50km yarıçap için gerçek deprem değerleri ile UKVH ağ modelinin tahmini değerler verilmiştir. UKVH ağ modelinde ki her bir SL değeri için bir grafik oluşturulmuştur. Bunlar sırasıyla SL değeri 5, 10, 25 ve 50 olarak belirlenmiştir. Gösterildiği gibi her SL için farklı sonuçlar elde edilmiştir. Bu grafiklerdeki depremlerin iniş ve çıkışlarının başarılı bir şekilde tahmin ettiği gösterilmiştir. Şekil 3.10'da SL değerinin 5 olarak belirlendiği derin UKVH ağının sonuçlarının detaylı gösterimine yer verilmiştir. Grafikte görüleceği gibi anlamlı bir tahmin eğrisi elde edilmiştir. Fakat yüksek şiddetli depremlerin tahminini pek başarılı değildir. Diğer grafiklerde anlık yüksek şiddetli depremler daha iyi bir şekilde tahmin edilmiştir.



Şekil 3.9. 1950-2017 yılları arası 50km yarıçap için farklı SL olan deprem tahminleri



Şekil 3.10. 1950-2017 yılları arası 50km yarıçap için SL=5 olan grafik

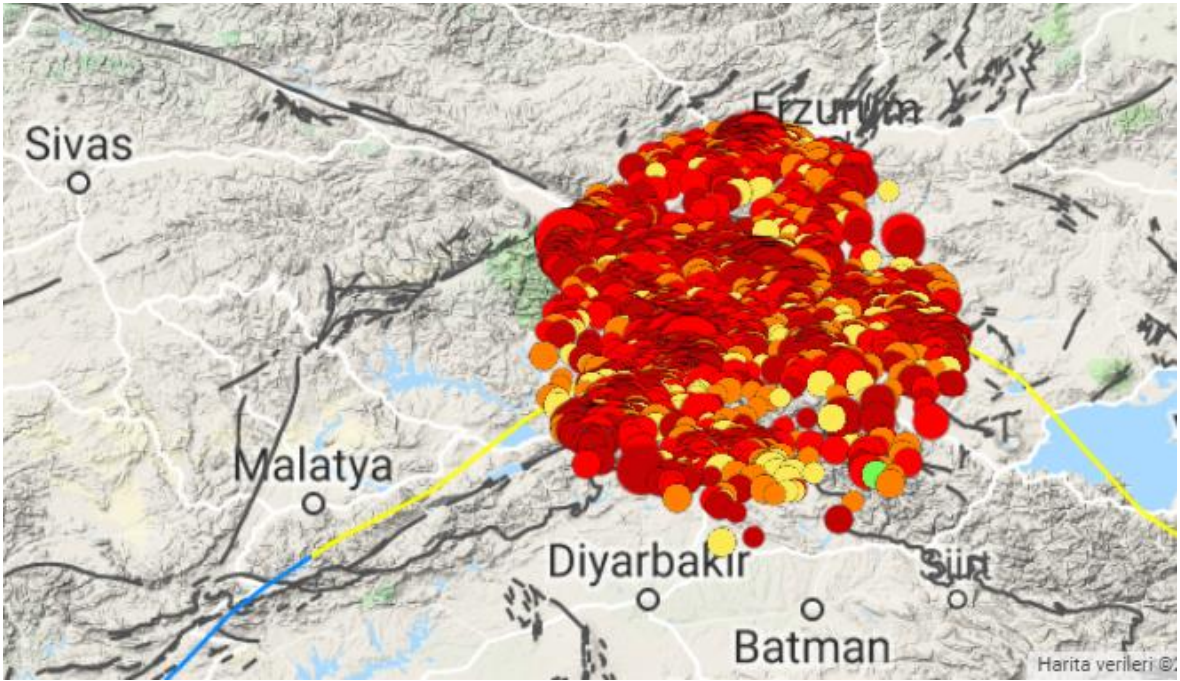
Tablo 3.5’ te, SL değeri 5 için 1950 ile 2017 yılları arasında 50km yarıçap için gerçek deprem verileri ile UKVH modelinin tahmin ettiği deprem verilerinden ilk 30’unun değerleri gösterilmiştir.

Tablo 3.5. 1950-2017 yılları arasında SL=5 için gerçek ve tahmin edilen ilk 30 deprem

No	Gerçek	Tahmin	No	Gerçek	Tahmin	No	Gerçek	Tahmin
1	3	2.986	11	2.5	2.864	21	2.9	3.034
2	2.6	2.974	12	3.4	2.845	22	2.8	2.930
3	2.8	2.860	13	2.6	2.813	23	3	2.877
4	2.8	2.834	14	3.1	2.888	24	2.9	2.850
5	3.2	2.798	15	3.1	2.726	25	2.7	2.977
6	2.9	2.975	16	3.2	3.050	26	3.1	2.870
7	2.8	2.834	17	2.6	2.927	27	2.6	2.923
8	2.7	2.998	18	3.3	3.048	28	3.3	2.895
9	2.9	2.870	19	2.8	3.070	29	2.8	2.792
10	2.9	2.991	20	2.9	3.408	30	2.9	2.872

### 3.1.3.2. Deneysel Çalışma 2: 100km Yarıçap

Deneysel çalışmaların ikincisinde ise, aynı şekilde Bingöl il merkezinin dairesel olarak 100km yarıçapı içerisinde kalan bölge üzerinde tahmin çalışmaları yapılmıştır. Tahmin çalışmaları için iki farklı zaman aralığı belirlenmiştir. Bu zaman aralıkları, 2000-2017 ve 1950 – 2017 olarak belirlenmiştir. Şekil 3.11’de, 100 km yarıçaplı bölge için 2000-2010 yılları arasında deprem şiddeti (Mc) 2.0 ve 9.0 arasındaki depremlerin harita üzerinde bir gösterimi sunulmuştur.



Şekil 3.11. 2000 ve 2010 arası 100km yarıçap için bir harita

100km yarıçap için belli yıllar arasında oluşan deprem hareketlerini elde edilmiştir. Elde edilen veri tabanlarını derin UKVH ağ modeline uygulanmıştır. Bu model de tekrar adımı olarak 300, yığın boyutu (batch size) 16 alınan aynı zamanda her yıl aralığı için 5, 10, 25 ve 50 SL değerleri alınmıştır. Aşağıdaki Tablo 3.6’da her yıl için kullandığımız toplam veri setleri, eğitim için kullanılan veri sayısı ve test için kullanılan veri sayısı verilmiştir.

**Tablo 3.6.** 100 km yarıçap için yıllara göre veri sayıları

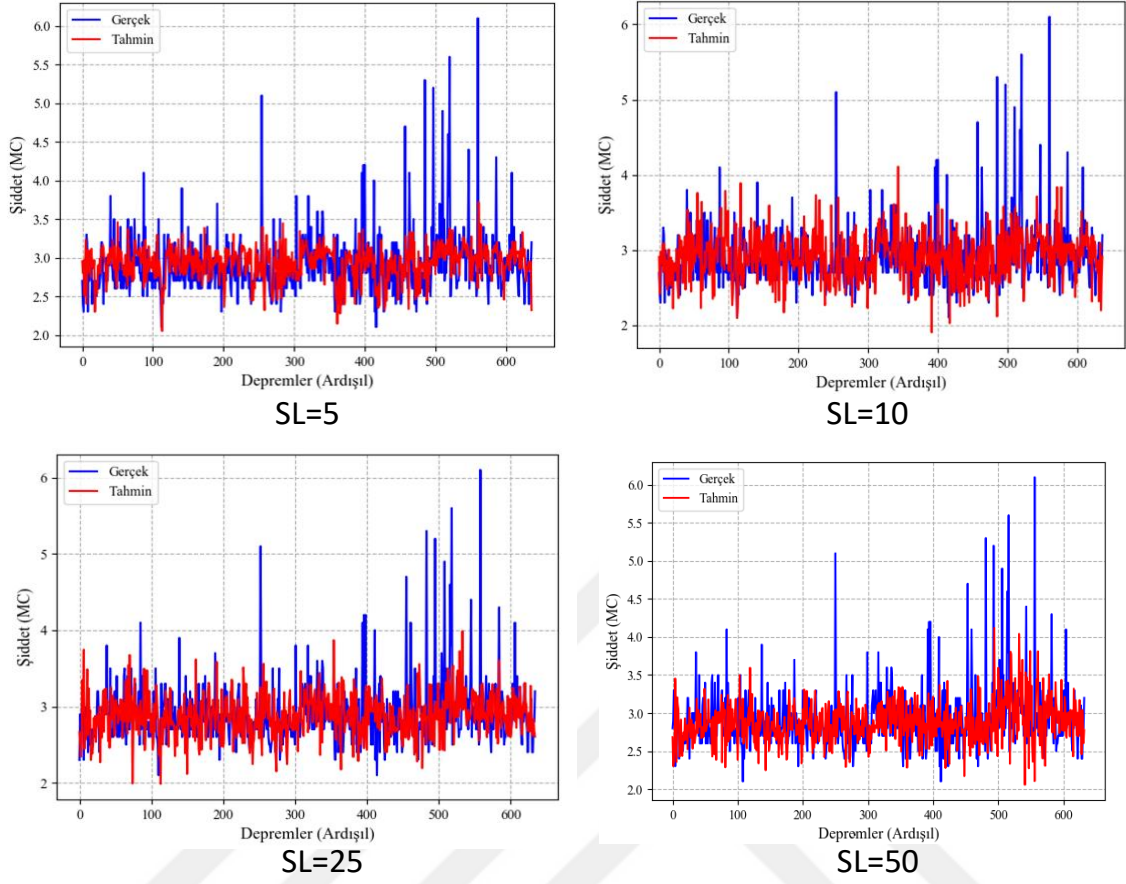
Yıllar	Toplam Veri Sayısı	Eğitim Veri Sayısı	Test Veri Sayısı
2000-2017	6378	5741	637
1950-2017	6690	6021	669

2000-2017 yılları arasındaki 100km yarıçap için yukarıdaki Tablo 3.6.'da belirtildiği gibi toplam verisayısının 6378, eğitim için kullanılan veri sayısının 5741, test için kullanılan verisayısının 637 olduğu gösterilmiştir. Yukarıda da bahsedildiği gibi derin UKVH ağ modelinde her bir SL için 300 aşama kullanılmıştır. Bu 300 aşama süresince eğitilmiş her bir SL için UKVH modeline ait kayıp(loss) fonksiyonunun son beş değeri Tablo 3.7'de verilmiştir.

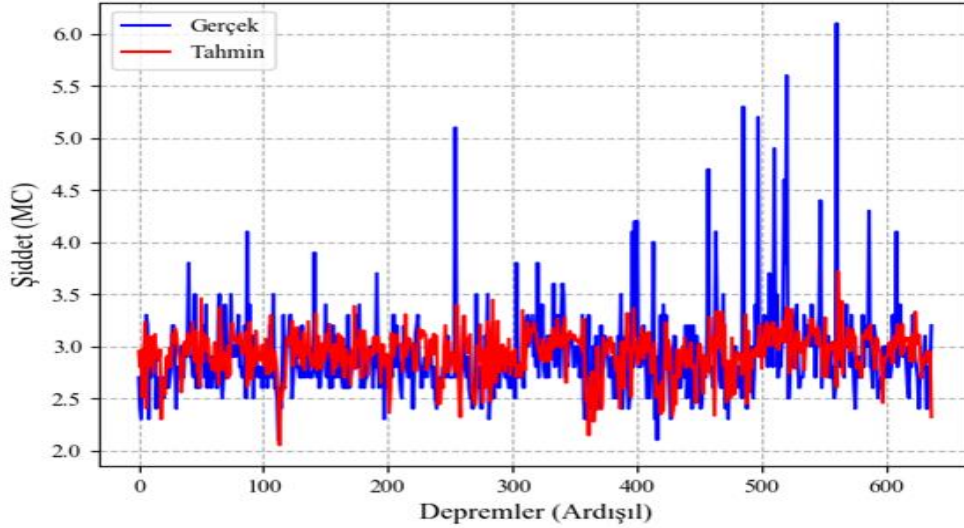
**Tablo 3.7.** 2000-2017 yılları arası 100km yarıçap için her SL için kayıplar

Kayıplar(Loss)	SL=5	SL=10	SL=25	SL=50
<b>Kayıp 1</b>	0.2072	0.2450	0.2204	0.2381
<b>Kayıp 2</b>	0.2063	0.2349	0.2177	0.2318
<b>Kayıp 3</b>	0.1976	0.2404	0.2186	0.2339
<b>Kayıp 4</b>	0.2008	0.2376	0.2232	0.2294
<b>Kayıp 5</b>	0.2041	0.2389	0.2210	0.2338

Aşağıdaki Şekil 3.12'deki grafikler 2000-2017 arasındaki 100km yarıçap için gerçek deprem değerleri ile UKVH ağ modelinin tahmini değerler verilmiştir. UKVH ağ modelinde ki her bir SL değeri için bir grafik oluşmuştur. Bunlar sırasıyla 5, 10, 25 ve 50 olarak belirlenmiştir. Gösterildiği gibi her SL için farklı sonuçlar elde edilmiştir. Bu grafiklerdeki depremlerin iniş ve çıkışlarının başarılı bir şekilde tahmin ettiği gösterilmiştir. Şekil 3.13'te SL değerinin 5 olarak belirlendiği derin UKVH ağının sonuçlarının detaylı gösterimine yer verilmiştir. Grafikte görüleceği gibi anlamlı bir tahmin eğrisi elde edilmiştir. Fakat yüksek şiddetli depremlerin tahminini pek başarılı değildir. Diğer grafiklerde anlık yüksek şiddetli depremler daha iyi bir şekilde tahmin edilmiştir.



Şekil 3.12. 2000-2017 yılları arası 100km yarıçap için farklı SL olan deprem tahminleri



Şekil 3.13. 2000-2017 yılları arası 100km yarıçap için SL=5 olan grafik

Tablo 3.8’de, SL değeri 5 için 2000 ile 2017 yılları arasında 100km yarıçap için gerçek deprem verileri ile UKVH modelinin tahmin ettiği deprem verilerinden ilk 30’unun değerleri gösterilmiştir.

**Tablo 3.8.** 2000-2017 yılları arasında SL=5 için gerçek ve tahmin edilen ilk 30 deprem

No	Gerçek	Tahmin	No	Gerçek	Tahmin	No	Gerçek	Tahmin
1	2.7	2.952	11	2.7	2.818	21	2.5	2.563
2	2.4	2.800	12	2.8	3.089	22	2.7	2.536
3	2.3	2.966	13	2.9	2.743	23	2.6	2.887
4	2.9	2.510	14	3	3.115	24	2.8	2.906
5	2.8	2.924	15	2.4	2.924	25	2.6	2.797
6	2.9	3.229	16	2.7	2.914	26	2.7	2.985
7	3.3	2.404	17	2.5	2.865	27	2.8	2.296
8	3.2	2.862	18	2.5	2.969	28	3.2	3.116
9	2.3	3.100	19	2.6	2.298	29	3.2	3.035
10	3	2.809	20	2.7	2.674	30	2.9	3.170

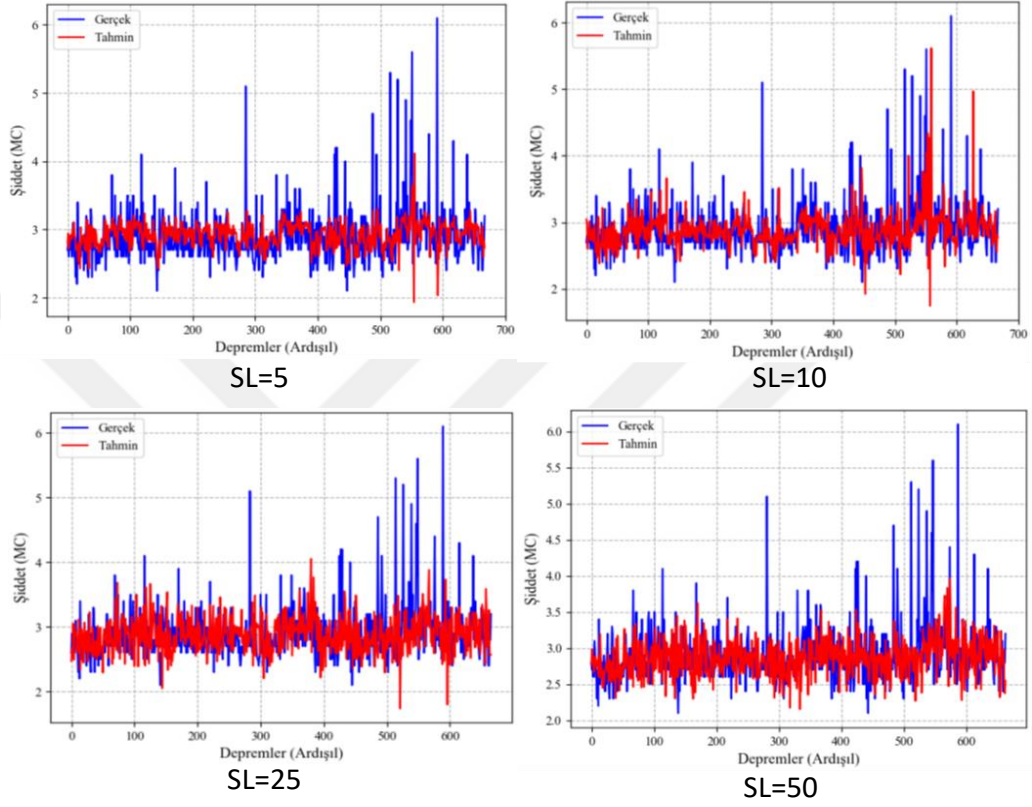
1950-2017 yılları arasındaki 100km yarıçap için yukarıdaki Tablo 3.6’da belirtildiği gibi toplam verisayısının 6690, eğitim için kullanılan verisayısının 6021, test için kullanılan verisayısının 669 olduğu gösterilmiştir. Yukarıda da bahsedildiği gibi derin UKVH ağ modelinde her bir SL için 300 aşama kullanılmıştır. Bu 300 aşama süresince eğitilmiş her bir SL için UKVH modeline ait kayıp(loss) fonksiyonunun son beş değeri Tablo 3.9’da verilmiştir.

**Tablo 3.9.** 1950-2017 yılları arası 100km yarıçap için her SL için kayıplar

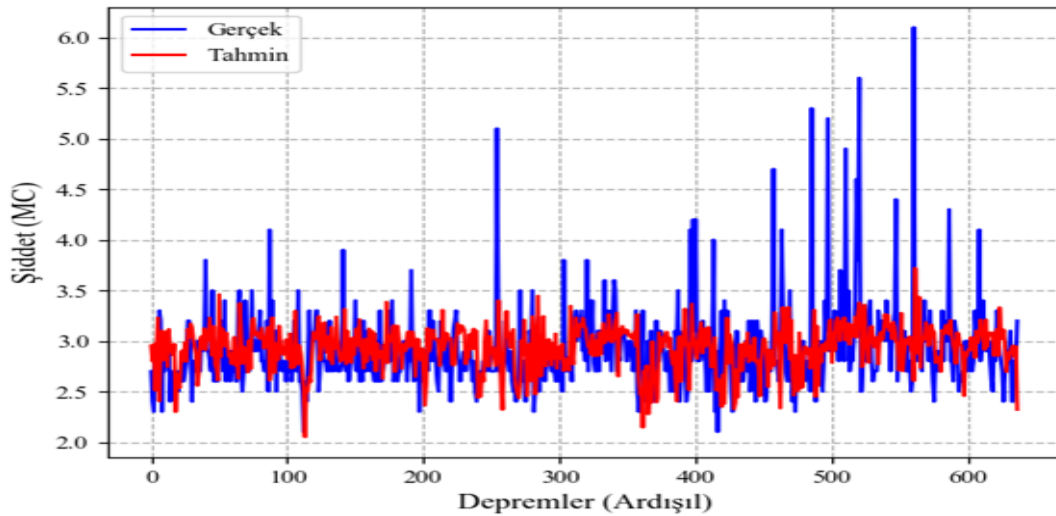
Kayıplar(Loss)	SL=5	SL=10	SL=25	SL=50
<b>Kayıp 1</b>	0.1827	0.2032	0.2159	0.2244
<b>Kayıp 2</b>	0.1814	0.1978	0.2310	0.2172
<b>Kayıp 3</b>	0.1815	0.1843	0.2205	0.2146
<b>Kayıp 4</b>	0.1823	0.1997	0.2180	0.2167
<b>Kayıp 5</b>	0.1808	0.1962	0.2208	0.2113

Aşağıdaki Şekil 3.14’teki grafikler 1950-2017 arasındaki 100km yarıçap için gerçek deprem değerleri ile UKVH ağ modelinin tahmini değerler verilmiştir. UKVH ağ modelinde ki her bir SL değeri için bir grafik oluşmuştur. Bunlar sırasıyla SL=5, 10, 25 ve 50 olarak belirlenmiştir. Gösterildiği gibi her SL için farklı sonuçlar elde edilmiştir. Bu grafiklerdeki depremlerin iniş ve çıkışlarının başarılı bir şekilde tahmin ettiği gösterilmiştir. Şekil 3.15’te

SL değerinin 5 olarak belirlendiği derin UKVH ağının sonuçlarının detaylı gösterimine yer verilmiştir. Grafikte görüleceği gibi anlamlı bir tahmin eğrisi elde edilmiştir. Fakat yüksek şiddetli depremlerin tahminini pek başarılı değildir. Diğer grafiklerde anlık yüksek şiddetli depremler daha iyi bir şekilde tahmin edilmiştir.



Şekil 3.14. 1950-2017 yılları arası 100km yarıçap için farklı SL olan deprem tahminleri



Şekil 3.15. 1950-2017 yılları arası 100km yarıçap için SL=5 olan grafik

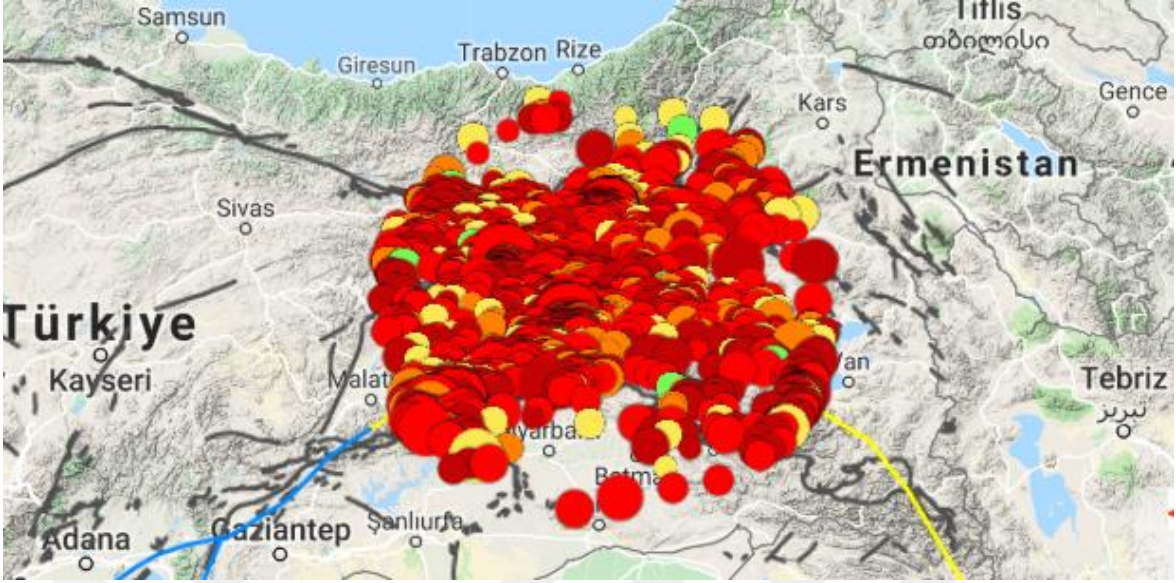
Tablo 3.10’da SL değeri 5 için 1950 ile 2017 yılları arasında 100km yarıçap için gerçek deprem verileri ile UKVH modelinin tahmin ettiği deprem verilerinden ilk 30’unun değerleri gösterilmiştir.

**Tablo 3.10.** 1950-2017 yılları arasında SL=5 için gerçek ve tahmin edilen ilk 30 deprem

No	Gerçek	Tahmin	No	Gerçek	Tahmin	No	Gerçek	Tahmin
1	2.7	2.853	11	2.9	2.749	21	2.7	2.925
2	2.8	2.916	12	2.6	2.526	22	2.6	2.846
3	2.6	3.003	13	2.6	2.804	23	2.8	2.701
4	2.9	2.904	14	2.3	2.611	24	2.7	2.864
5	2.8	3.106	15	2.7	2.426	25	2.9	2.683
6	2.7	2.930	16	2.2	3.033	26	2.8	3.148
7	2.9	2.879	17	3.4	2.793	27	2.4	2.905
8	2.6	2.786	18	2.9	2.780	28	2.7	2.587
9	2.9	2.695	19	2.5	2.863	29	2.5	2.423
10	3.1	2.719	20	2.7	2.894	30	2.9	3.063

### 3.1.3.3. Deneysel Çalışma 3: 200km Yarıçap

Deneysel çalışmaların üçüncüsünde ise, aynı şekilde Bingöl il merkezinin 200km yarıçapı içerisinde kalan bölge üzerinde tahmin çalışmaları yapılmıştır. Tahmin çalışmaları için iki farklı zaman aralığı belirlenmiştir. Bu zaman aralıkları, 2000- 2017 ve 1950 – 2017 olarak belirlenmiştir. Şekil 3.16’da 200 km yarıçaplı bölge için 2000-2010 yılları arasında deprem şiddeti (Mc) 2.0 ve 9.0 arasındaki depremlerin harita üzerinde bir gösterimi sunulmuştur.



Şekil 3.16. 2000-2005 arası 200km yarıçap için sorgu sonucu

200km yarıçap için belli yıllar arasında oluşan deprem hareketlerini elde edilmiştir. Elde edilen veri tabanlarını derin UKVH ağ modeline uygulanmıştır. Bu modelde aşama sayısı=300, yığın boyutu (batch size)=16 alınmış ve her yıl aralığı için SL=5, 10, 25 ve 50 değerleri kullanılmıştır. Aşağıdaki Tablo 3.11’de her yıl için kullandığımız toplam veri setleri, eğitim için kullanılan veri sayısı ve test için kullanılan veri sayısı verilmiştir.

Tablo 3.11. 200 km yarıçap için yıllara göre veri sayıları.

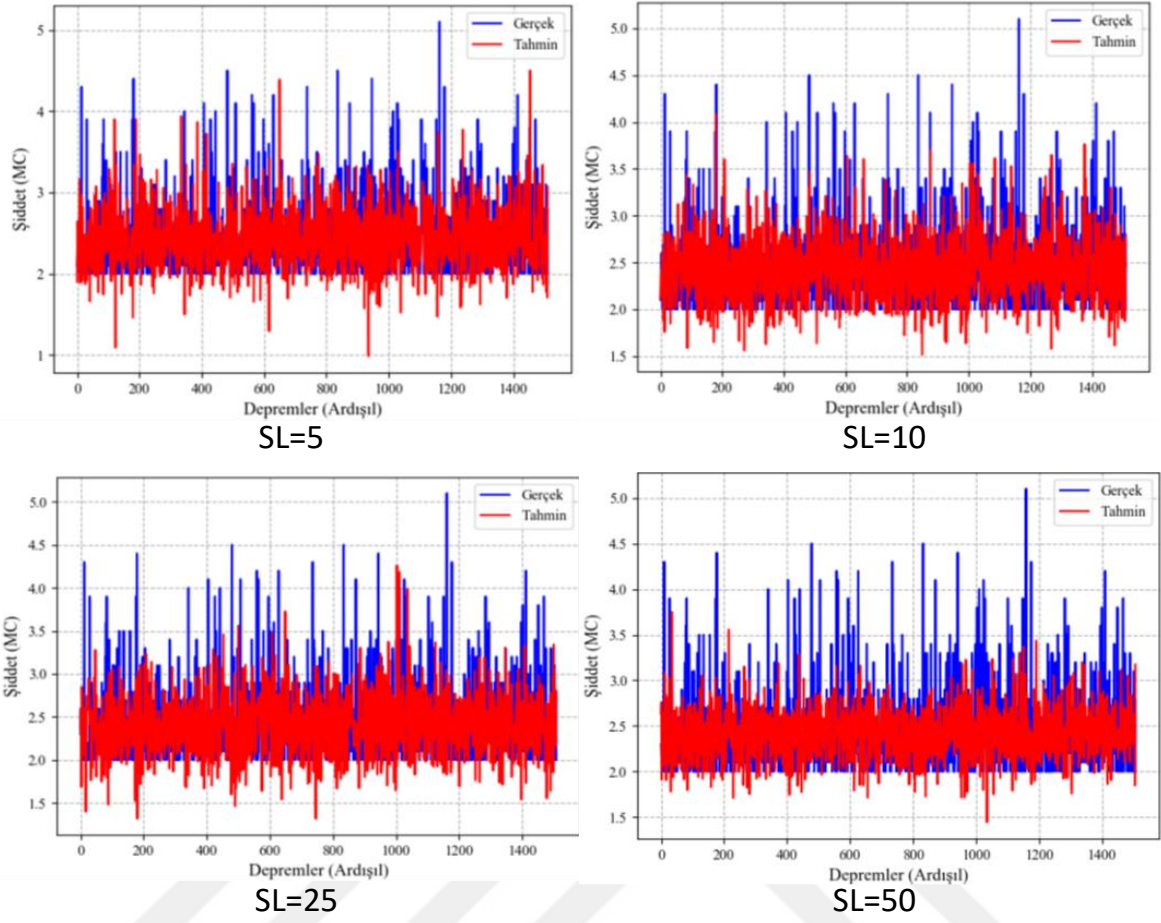
Yıllar	Toplam verisayısı	Eğitim Veri Sayısı	Test Veri Sayısı
2000-2017	15121	13610	1511
1950-2017	15936	14343	1593

2000-2017 yılları arasındaki 200km yarıçap için yukarıdaki Tablo 3.11’de belirtildiği gibi toplam verisayısının 15121, eğitim için kullanılan verisayısının 13610, test için kullanılan verisayısının 1511 olduğu gösterilmiştir. Yukarıda da bahsedildiği gibi derin UKVH ağ modelinde her bir SL için 300 aşama kullanılmıştır. Bu 300 aşama süresince eğitilmiş her bir SL için UKVH modeline ait kayıp(loss) fonksiyonun son beş değeri Tablo 3.12’de verilmiştir.

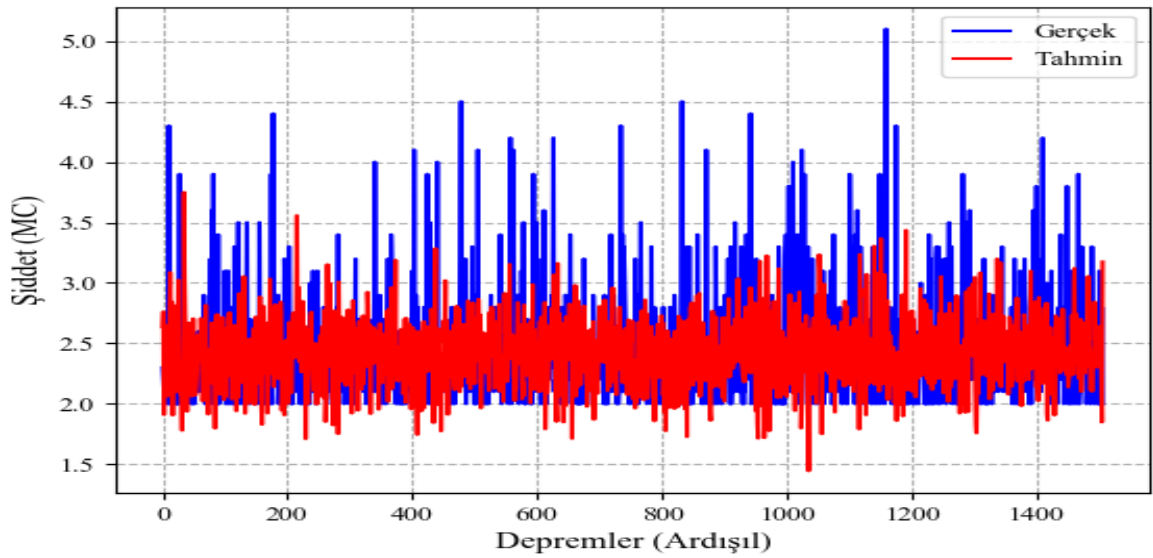
**Tablo 3.12.** 2000-2017 yılları arası 200km yarıçap için her SL için kayıplar.

Kayıplar(Loss)	SL=5	SL=10	SL=25	SL=50
<b>Kayıp 1</b>	0.2312	0.2324	0.2182	0.2111
<b>Kayıp 2</b>	0.2226	0.2321	0.2216	0.2156
<b>Kayıp 3</b>	0.2210	0.2412	0.2206	0.2154
<b>Kayıp 4</b>	0.2183	0.2314	0.2166	0.2113
<b>Kayıp 5</b>	0.2269	0.2348	0.2180	0.2167

Aşağıdaki Şekil 3.17'deki grafikler 2000-2017 arasındaki 200km yarıçap için gerçek deprem değerleri ile UKVH ağ modelinin tahmini değerler verilmiştir. UKVH ağ modelinde ki her bir SL değeri için bir grafik oluşmuştur. Bunlar sırasıyla 5, 10, 25 ve 50 olarak belirlenmiştir. Gösterildiği gibi her SL için farklı sonuçlar elde edilmiştir. Bu grafiklerdeki depremlerin iniş ve çıkışlarının başarılı bir şekilde tahmin ettiği gösterilmiştir. Şekil 3.18'de SL değerinin 50 olarak belirlendiği derin UKVH ağının sonuçlarının detaylı gösterimine yer verilmiştir. Grafikte görüleceği gibi anlamlı bir tahmin eğrisi elde edilmiştir. Fakat yüksek şiddetli depremlerin tahminini pek başarılı değildir. Diğer grafiklerde anlık yüksek şiddetli depremler daha iyi bir şekilde tahmin edilmiştir.



Şekil 3.17. 2000-2017 yılları arası 200km yarıçap için farklı SL olan deprem tahminleri



Şekil 3.18. 2000-2017 yılları arası 200km yarıçap için SL=50 olan grafik

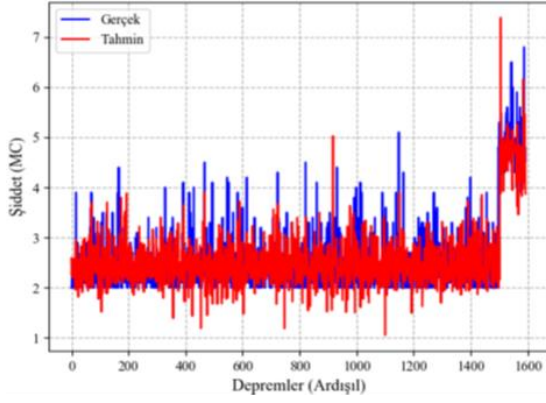
1950-2017 yılları arasındaki 200km yarıçap için yukarıdaki Tablo 3.11’de belirtildiği gibi toplam verisayısının 15936, eğitim için kullanılan verisayısının 14343, test için kullanılan

verisayının 1593 olduğu gösterilmiştir. Yukarıda da bahsedildiği gibi derin UKVH ağ modelinde her bir SL için 300 aşama kullanılmıştır. Bu 300 aşama süresince eğitilmiş her bir SL için UKVH modeline ait kayıp(loss) fonksiyonunun son beş değeri Tablo 3.13'te verilmiştir.

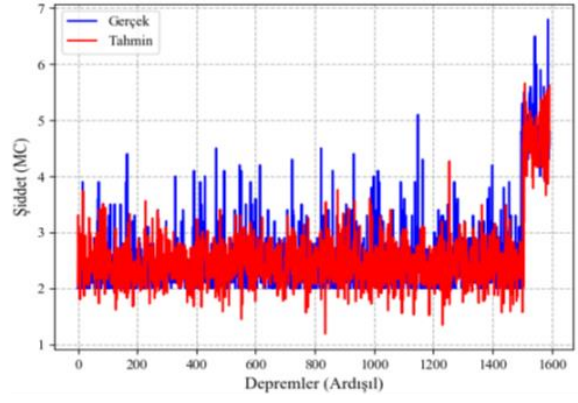
**Tablo 3.13.** 1950-2017 yılları arası 200km yarıçap için her SL için kayıplar

Kayıplar(Loss)	SL=5	SL=10	SL=25	S=50
<b>Kayıp 1</b>	0.2357	0.2656	0.2066	0.2108
<b>Kayıp 2</b>	0.2475	0.2506	0.2016	0.2080
<b>Kayıp 3</b>	0.2431	0.2561	0.2070	0.2133
<b>Kayıp 4</b>	0.2457	0.2465	0.2089	0.2065
<b>Kayıp 5</b>	0.2467	0.2529	0.2057	0.2093

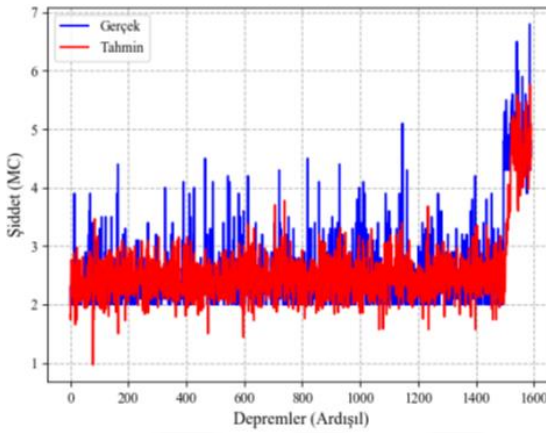
Aşağıdaki Şekil 3.19'daki grafikler 1950-2017 arasındaki 200km yarıçap için gerçek deprem değerleri ile UKVH ağ modelinin tahmini değerler verilmiştir. UKVH ağ modelinde ki her bir SL değeri için bir grafik oluşmuştur. Bunlar sırasıyla 5, 10, 25 ve 50 olarak belirlenmiştir. Gösterildiği gibi her SL için farklı sonuçlar elde edilmiştir. Bu grafiklerdeki depremlerin iniş ve çıkışlarının başarılı bir şekilde tahmin ettiği gösterilmiştir. Şekil 3.20'de SL değerinin 25 olarak belirlendiği derin UKVH ağının sonuçlarının detaylı gösterimine yer verilmiştir. Grafikte görüleceği gibi anlamlı bir tahmin eğrisi elde edilmiştir. Fakat yüksek şiddetli depremlerin tahminini pek başarılı değildir. Diğer grafiklerde anlık yüksek şiddetli depremler daha iyi bir şekilde tahmin edilmiştir. Şekil 3.20'de, SL 25 değeri için modelin tahmin grafiği detaylı olarak gösterilmiştir.



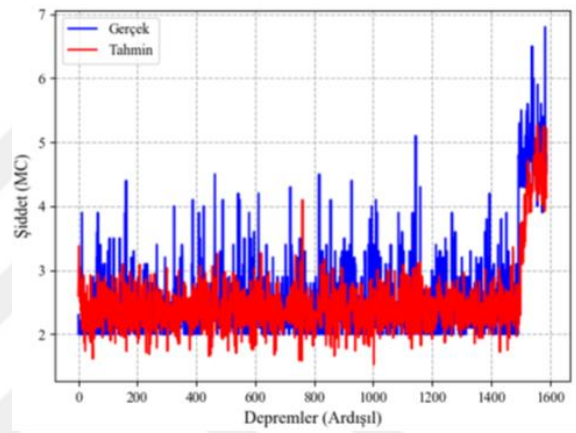
SL=5



SL=10

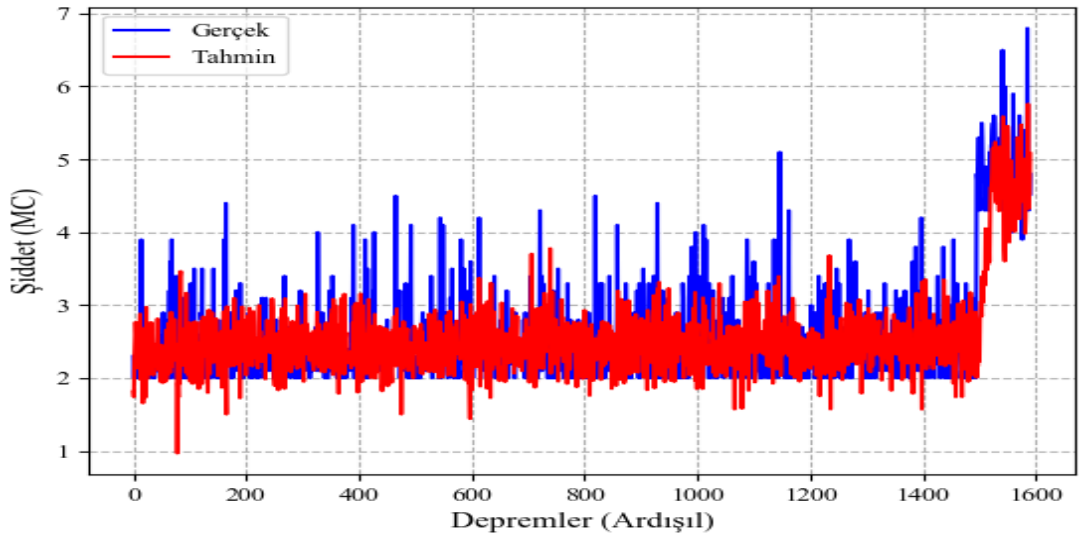


SL=25



SL=50

Şekil 3.19. 1950-2017 yılları arası 200km yarıçap için farklı SL olan deprem tahminleri



Şekil 3.20. 1950-2017 yılları arası 200km yarıçap için SL=25 olan grafik

#### 4. TARTIŞMA

Bu tez çalışmasının sonucunda TYSA'ların en sık kullanılan versiyonu olan UKVH ağ modeli kullanılarak Bingöl ili ve çevresini ele alınmıştır. Alınan bölgeyi dairesel olarak 50km, 100km ve 200km yarıçaplar alarak tahminler elde edilmiştir. Elde edilen tahminlerle doğrultusunda yüksek oranda başarı elde edildiği gözlemlenmiştir. Literatürde UKVH ağ modelini kullanarak deprem tahmini yapan mevcut çalışmaların az olması nedeniyle çok fazla çalışma olmayıp benzer konuda Wang ve ark. (2017) UKVH ağ modelini kullanarak mekânsal ve zamansal perspektiften yeni bir deprem tahmini sistemi önerilmiştir. Elde edilen sonuçların zamansal ve mekansal tahmin parçacıklarıyla doğru tahminler yapabileceğini göstermişlerdir. Yaptığımız çalışmanın tahmini sonuçları Wang ve ark. elde ettiği tahmini sonuçların destekler niteliktedir. UKVH ağ modelini kullanarak farklı çalışmalar yapılmıştır. Bunların birkaçı aşağıda verilmiştir.

Sak ve ark. (2014), Uzun-Kısa Vadeli Hafıza (UKVH) kullanarak büyük ölçekli akustik modelleme için çalışma yapmışlardır. UKVH, zamansal dizileri ve uzun menzilli bağımlılıklarını klasik TYSA'lardan daha doğru modellemek üzere tasarlanmış spesifik tekrarlayan bir sinir ağı mimarisidir. Bu çalışmalarında, konuşma tanımada büyük ölçekli akustik modelleme için UKVH TYSA mimarilerini önermişlerdir.

Sutskever ve ark. (2014), sinir ağlarıyla ardışıl öğrenme dizisi konulu bir çalışma yapmışlardır. Bu çalışmada, dizi yapısı üzerinde minimal varsayımlar yapan dizi öğrenmeye genel bir uçtan-uca yaklaşım sunmuşlardır. Yöntem olarak, giriş sırasını sabit bir boyutluluk vektörüne eşleştirmek için çok katlı UKVH kullanılmış ve ardından vektörden hedef sekansı çözmek için bir başka derin UKVH ağı kullanmışlardır.

## 5. SONUÇLAR

Bu tez çalışmasında öncelikle makine öğrenmesine değinilmiştir. Makine öğrenmesinin öğrenme gelişimiyle ilgili parametreleri incelenmiştir ve makine öğrenmesinin öğrenme teknikleri ele alınmıştır. Bunların yanı sıra yapay sinir ağları detaylıca incelenmiştir. Günümüzde, yapay sinir ağları pek çok alanda uygulama ve birçok araştırma yapılmıştır. YSA araştırmaları; sınıflandırma, optimizasyon, tahmin, şekil tanıma, modelleme ve öğrenme gibi birçok alanda kullanıldığı gösterilmiştir. Yapay sinir ağların yapısı, öğrenme şekilleri, algoritmalarına yer verilmiştir. Bunlara ek olarak YSA'ların üstünlükleri ve dezavantajları ele alınmıştır. Yapay sinir ağlarının diğer bir sınıfı olan "Tekrarlayan Yapay Sinir Ağları" son yıllarda popüler hale gelmiştir. Bu çalışmada Tekrarlı Sinir Ağlarının Yapısı, tasarımı ve eğitimi detaylıca incelenmiştir. Tekrarlayan Yapay Sinir Ağları el yazısı tanıma, görüntü tanıma, makine tercüme, dil modellenmesi, konuşma tanıma bunun gibi birçok alanda başarı göstermiştir.

Bu tez çalışmasında, TYSA'ların en sık kullanılan versiyonu olan UKVH ağ modeli kullanılarak bölgesel deprem tahminleri sağlanmıştır. Bölge olarak Türkiye bölgesindeki Bingöl ili ve çevresi ele alınmıştır. Belirlenen bölgelerin deprem hareketlilik verileri B.Ü Kandilli Rasathanesi BDTİM Deprem Sorgulama Sisteminden elde edilmiştir. Daha sonra belirlediğimiz ilin belli yıl aralıkları arasında depremler tahminler üretilmiştir. Her yıl aralığı için deprem sorgulama sisteminde seçilen bölgenin belirlenen yarıçaptaki dairesel alanı taranmıştır. Dairesel olarak 50km, 100km ve 200km yarıçapındaki bölgeler üzerinde tahminler üretilmiştir. Sorgulanan depremlerin şiddeti ise 2.0 ve 9.0 derinliği ise 0 ve 500 km arasında belirlenmiştir. Daha sonra yıllara göre elde edilen veriler organize edildikten sonra bir derin UKVH ağ modeline uygulanmıştır. UKVH modelinden elde edilen sonuçlar tahminlerin başarılı olduğunu göstermiştir. Elde edilen deprem tahminleri gerçek ve tahmin değerleri ile grafiksel olarak sunulmuştur.

Elde edilen sonuçlara göre önerilen sistemin ardışık deprem tahmininde kullanılabilecek etkili bir yapı olduğu söylenebilir. Fay hatlı bazlı çalışarak ve verileri arasındaki zaman farklılıkları modellenerek daha iyi sonuçlar elde edilebilir. Bu çalışmalar gelecekte üzerinde durulacak başlıca konular olacaktır.

## KAYNAKLAR

- Aktaş, M., Okumuş, H. İ.,** 2003. Doğrudan Moment Kontrollü Asenkron Motorun Stator Direncinin Yapay Sinir Ağı ile Kestirimi, International XII. In Turkish Symposium on Artificial Intelligence and Neural Networks.
- Alpaydin, E.,** 2004. Introduction to machine learning. MIT press.
- Asencio-Cortés, G., Martínez-Álvarez, F., Morales-Esteban, A., Reyes, J., & Troncoso, A.,** 2015. Improving earthquake prediction with principal component analysis: application to Chile. In International Conference on Hybrid Artificial Intelligence Systems, 393-404, Springer, Cham.
- Aşkın, D.,** 2011. Kuru Tip Transformatörlerin Sargı Sıcaklık Davranışının Yapay Sinir Ağ Kullanarak Modellenmesi, Yüksek lisans tezi, Gazi Üniversitesi, Ankara.
- Baose, B.,** 2002. "Neural Network Principles and Applications", Prentice Hall PTR, United States of America, 625-657.
- Barbounis, T. G., Theocharis, J. B., Alexiadis, M. C., & Dokopoulos, P. S.,** 2006. Long-term wind speed and power forecasting using local recurrent neural network models. IEEE Transactions on Energy Conversion, 21(1), 273-284.
- Baykasoğlu, A.,** 2005. Veri Madenciliği ve Çimento Sektöründe Bir Uygulama. Akademik Bilişim, 2-4.
- Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., ... & Bengio, Y.,** 2010. Theano: A CPU and GPU math compiler in Python. In Proc. 9th Python in Science Conf, 1-7.
- Bilen, M., Işık, A. H., & Yiğit, T.,** 2015. Sismik Darbelerini Sınıflandırarak Deprem Tehlikesinin Tahmin Edilmesi. Bilgisayar Mühendisliği, Burdur.
- Blum, A. L., & Rivest, R. L.,** 1993. Training a 3-node neural network is NP-complete. In Machine learning: From theory to applications, 9-28, Springer Berlin Heidelberg.
- Bozo, M.,** 2014. Makine Öğrenmesi ODTÜ bilgisayar topluluğu elektronik dergisi, e-bergi. <http://e-bergi.com/yy/makine-grenimi>. (Erişim tarihi: 01.02.2018).
- Brownlee, J.,** 2016. Gradient descent for machine learning. <http://machinelearningmastery.com/gradient-descent-for-machine-learning>. (Erişim tarihi: 01.02.2018).
- Chapelle, O. and Schölkopf, B.,** 2006. Semi-supervised learning. Cambridge: MIT press, 1-4.

- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y.,** 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
- Connor, J. T., Martin, R. D., & Atlas, L. E.,** 1994. Recurrent neural networks and robust time series prediction. IEEE transactions on neural networks, 5(2), 240-254.
- Çırak, G., & Çokluk, Ö.,** 2012. Yükseköğretimde öğrenci başarılarının sınıflandırılmasında yapay sinir ağları ve lojistik regresyon yöntemlerinin kullanılması. Yüksek Lisans Tezi, Ankara Üniversitesi, Eğitim Bilimleri Enstitüsü, Ölçme ve Değerlendirme Anabilim Dalı, Ankara.
- Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y.,** 2014. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Advances in neural information processing systems, 2933-2941.
- Deveci, M.,** 2012. Yapay sinir ağları ve bekleme süresinin tahmininde kullanılması. Yüksek Lisans tezi, Gazi Üniversitesi, Ankara.
- Durak, B.,** 2011. A classification Algorithm Using mahalanobis Distance Clustering of Data With Applications on Biomedical Data Sets, Doctoral dissertation, Middle East Technical University.
- Elman, J. L.,** 1990. Finding structure in time. Cognitive science, 14(2), 179-211.
- Elmas, Ç.,** 2011. Yapay zekâ uygulamaları. Seçkin yayıncılık, Ankara, 23.
- Elmas, Ç.,** 2003. Yapay Sinir Ağları, Seçkin Yayıncılık, Ankara.
- Foo, J.S., Ghosh, P.S.,** 2002. "Artificial Neural Network Modelling of Partial Discharge Parameter For Transformer Oil Diagnosis", Annual Report Conference on Electrical Insulation and Dielectric Phenomena, Malaysia, 470-473.
- Funahashi, K. I., & Nakamura, Y.,** 1993. Approximation of dynamical systems by continuous time recurrent neural networks. Neural networks, 6(6), 801-806.
- Graves, A.,** 2013. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.
- He, Q., Si, J., & Tylavsky, D. J.,** 2000. Prediction of top-oil temperature for transformers using neural networks. IEEE Transactions on Power Delivery, 15(4), 1205-1211.
- Hinton, G. E.,** 1986. Learning distributed representations of concepts. In Proceedings of the eighth annual conference of the cognitive science society, 12(1).
- Hinton, G. E., & Sejnowski, T. J.,** 1983. Optimal perceptual inference. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 448-453, IEEE New York.

- Hochreiter, S., & Schmidhuber, J.,** 1997. Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J.,** 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.
- Hopfield, J. J.,** 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554-2558.
- Hopfield, J. J., & Tank, D. W.,** 1985. "Neural" computation of decisions in optimization problems. *Biological cybernetics*, 52(3), 141-152.
- İkizler, N.,** 2002. Benefit maximizing classification using feature intervals, Doctoral dissertation, Bilkent University, Ankara.
- Jordan, M. I.,** 1986. Serial order: a parallel distributed processing approach. Technical report, June 1985-March 1986, No. AD-A-173989/5/XAB; ICS-8604. California Univ., San Diego, La Jolla, USA, Inst. For Cognitive Science.
- Kamijo, K. I., & Tanigawa, T.,** 1990. Stock price pattern recognition-a recurrent neural network approach. In *Neural Networks, 1990.*, 1990 IJCNN International Joint Conference on, 215-221.
- Karpathy, A.,** 2015. The unreasonable effectiveness of recurrent neural networks. Andrej Karpathy blog.
- Kavzoğlu, T., & Çölkesen, İ.,** 2010. Destek vektör makineleri ile uydu görüntülerinin sınıflandırılmasında kernel fonksiyonlarının etkilerinin incelenmesi. *Harita Dergisi*, 144(7), 73-82.
- Kaya, T.,** 2015. Makine öğrenme yöntemleri ile trafik kazaları için risk tahmini yapabilen web tabanlı bir yazılım, Yüksek Lisans Tezi, Gazi Üniversitesi, Bilişim Enstitüsü, Ankara.
- Leskovec, J., Rajaraman, A., & Ullman, J. D.,** 2014. Mining of massive datasets. Cambridge university press.
- Lin, C. T., & Lee, C. G.,** 1996. Neural fuzzy systems. PTR Prentice Hall.
- Lin, H. M., Chang, S. K., Wu, J. H., & Juang, C. H.,** 2009. Neural network-based model for assessing failure potential of highway slopes in the Alishan, Taiwan Area: Pre- and post-earthquake investigation. *Engineering Geology*, 104(3), 280-289
- Lipton, Z. C., Berkowitz, J., & Elkan, C.,** 2015. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.
- MacKay, D. J.,** 1992. Bayesian interpolation. *Neural computation*, 4(3), 415-447.

- Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S.,** 2010. Recurrent neural network based language model, In Interspeech, 2(3).
- Nasser, A.R.,** 2012. Matlab Ortamında Kesme Dalgalarının Ayrırılması Ölçümleri Kullanarak Deprem Tahmini. Yüksek Lisans Tezi, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Negarestani, A., Setayeshi, S., Ghannadi-Maragheh, M., & Akashe, B.,** 2002. Layered neural networks based analysis of radon concentration and environmental parameters in earthquake prediction. Journal of environmental radioactivity, 62(3), 225-233.
- Nguyen, H. H.,** 2007. A neural fuzzy approach to modeling the thermal behavior of power transformers, Doctoral dissertation, Victoria University.
- Olah, C.,** 2015. Understanding LSTM Neural Network. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. ( Erişim tarihi: 02.02.2018).
- Öztemel, E.,** 2003. Yapay Sinir Ağları. Papatya Yayıncılık, İstanbul, 29-34.
- Pascanu, R., Mikolov, T., & Bengio, Y.,** 2013. On the difficulty of training recurrent neural networks. ICML (3), 28, 1310-1318.
- Sağiroğlu, Ş., Beşdok, E., Erler, M.,** 2003. Yapay Zekâ Uygulamaları-1, Ufuk Yayıncılık, 43.
- Saha, S., & Raghava, G. P. S.,** 2006. Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. Proteins: Structure, Function, and Bioinformatics, 65(1), 40-48.
- Sak, H., Senior, A., & Beaufays, F.,** 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth Annual Conference of the International Speech Communication Association.
- Schuster, M., and Paliwa, K. K.,** 1997. Bidirectional Recurrent Neural Networks, IEEE Transactions on signal processing, 45(11).
- Siegelmann, H. T., & Sontag, E. D.,** 1991. Turing computability with neural nets. Applied Mathematics Letters, 4(6), 77-80.
- Sikder, I. U., & Munakata, T.,** 2009. Application of rough set and decision tree for characterization of premonitory factors of low seismic activity. Expert Systems with Applications, 36(1), 102-110.
- Sutskever, I., Vinyals, O., & Le, Q. V.,** 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, 3104-3112.
- URL-1,** 2017. <https://www.frmartuklu.org/konu/ulkemizde-meydana-gelen-buyuk-depremler-ve-tarihleri-nelerdir.245649/>. (Erişim tarihi: 30.12.2017).

- URL-2**, 2018. B. Ü. K. R. D. Sistemi, “Boğaziçi Üniversitesi Kandilli Rasathanesi ve Deprem Araştırma Enstitüsü (KRDAE) internet sitesi: <http://www.koeri.boun.edu.tr/sismo/zeqdb>”.( Erişim tarihi:10.06 2018).
- URL-3**, 2018. <https://images.search.yahoo.com>. (Erişim tarihi: 10.06.2018).
- Wang, Q., Guo, Y., Yu, L., & Li, P.**, 2017. Earthquake prediction based on spatio-temporal data mining: an LSTM network approach. *IEEE Transactions on Emerging Topics in Computing*.
- Werbos, P. J.**, 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10), 1550-1560.
- Williams, R. J., & Zipser, D.**, 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2), 270-280.
- Wu, R. C., Chen, R. S., & Chian, S. S.**, 2006. Design of a product quality control system based on the use of data mining techniques. *IIE Transactions*, 38(1), 39-51.
- Yetkin, M.**, 2014. Tanker şamandıra bağlama sistemlerinin yapay sinir ağları tekniğiyle optimizasyonu. Yüksek lisans tezi, İstanbul Teknik Üniversitesi, İstanbul.
- Yıldırım, Ö.**, 2010. Veri Madenciliği Yöntemleriyle Deprem Analizi. Yüksek Lisan Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elâzığ.

## ÖZGEÇMİŞ

08.09.1994 tarihinde Bingöl'de doğdu. İlkokullu eğitimini İMKB Kaleönü İlkokulu'nda, lise öğrenimini İMKB Bingöl Lisesi'nde tamamladı. 2012 yılında Munzur Üniversitesi Mühendislik Fakültesi Elektrik ve Elektronik mühendisliğine başlayıp 2016 yılında mezun oldu. Halen Munzur Üniversitesi Fen Bilimleri Enstitüsü Elektrik ve Elektronik Mühendisliğinde Yüksek Lisans eğitimine devam etmektedir.

