

BİR VE İKİ GİZLİ KATMANLI YAPAY SİNİR

AĞLARININ MATEMATİĞİ

Tayfun ÜNAL

Matematik Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Ünver ÇİFTÇİ

2019

T.C
TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

BİR VE İKİ GİZLİ KATMANLI YAPAY SİNİR AĞLARININ
MATEMATİĞİ

Tayfun ÜNAL

MATEMATİK ANABİLİM DALI

DANIŞMAN: Doç. Dr. Ünver ÇİFTÇİ

TEKİRDAĞ-2019

Her hakkı saklıdır.

ÖZET

Yüksek Lisans Tezi

BİR VE İKİ GİZLİ KATMANLI YAPAY SİNİR AĞLARININ MATEMATİĞİ

Tayfun ÜNAL

Tekirdağ Namık Kemal Üniversitesi
Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı

Danışman: Doç.Dr. Ünver ÇİFTÇİ

Bu tezde lojistik regresyon ve yapay sinir ağlarından bahsedilmiş, özellikle bir ve iki gizli katmanlı yapay sinir ağlarının mimarileri incelenmiştir. Bu mimariler XOR veri kümesi üzerinde kullanılmıştır. Lojistik regresyonun sınıflandıramadığı XOR veri kümesi bir ve iki gizli katmanlı yapay sinir ağı mimarileriyle sınıflandırılmıştır. Daha sonra matematiksel temellerinin araştırılması için baskı maliyet fonksiyonu tanımlanmıştır. Bu fonksiyon, bir gizli katmanlı yapay sinir ağının gizli katmanındaki sonuncu nöronunun parametrelerini sıfır yapacak şekilde tasarlanmıştır. Yani sonuncu nöron baskılanmıştır. Böylece r ideal nöron sayısı ve ϵ istenilen maliyet değeri olduğu varsayılırsa r için hesaplanan maliyet değeri, r için hesaplanan baskı maliyet değeri arasındaki fark çok küçük ve aralarındaki ilişki $J(W, b) \leq J_{baski}(W, b) \leq \epsilon$ olduğu gözlenmiş ve ispatlanmıştır. Tanımlanan baskı maliyet fonksiyonu kullanılarak bir deney yapılmış ve bu deneyin sonucunda XOR veri kümesi için bir gizli katmanlı yapay sinir ağının ideal nöron sayısı tespit edilmiştir. Daha sonra iki gizli katmanlı yapay sinir ağları için baskı maliyet fonksiyonu tanımlanmıştır. Bu baskı maliyet fonksiyonu birinci ve ikinci gizli katmandaki nöron sayılarını eşit tutup bu mimarinin birinci ve ikinci gizli katmanındaki son nöronlar baskılanmıştır. Bu fonksiyon kullanılarak bir gizli katmanlı yapay sinir ağları için yapılan deney, iki gizli katmanlı yapay sinir ağları için de yapılmıştır. Bu deneyle XOR veri kümesi için iki gizli katmanlı yapay sinir ağının ideal nöron sayısı tespit edilmiştir. Sonuç bölümünde elde edilen yöntemin geliştirilmesi incelenmiştir.

Anahtar kelimeler: Derin öğrenme, yapay sinir ağları, gizli katman sayısı, XOR problemi

2019, 96 sayfa

ABSTRACT

MSc. Thesis

MATHEMATICS OF NEURAL NETWORKS WITH ONE AND TWO HIDDEN LAYERS

Tayfun ÜNAL

Tekirdağ Namık Kemal University
Graduate School of Natural and Applied Sciences
Department of Mathematics

Supervisor: Doç. Dr. Ünver ÇİFTÇİ

In this study, logistic regression and artificial neural networks were mentioned. Especially, architectures of one and two hidden layer artificial neural networks were analyzed. These architectures were used for XOR problem. XOR problem which logistic regression couldn't classify was classified with one and two hidden layer architectures. Then, in order to investigate the mathematical foundations, the suppressed cost function was defined. This function makes last neuron's parameters in the hidden layer of one neural networks zero. So, suppose that r is the ideal number of neurons and ε is the desired value for cost function, then the connection of cost function, suppressed cost function and ε is $J(W, b) \leq J_{baski}(W, b) \leq \varepsilon$. By using this function, an experiment have been done and the ideal number of neuron was detected for XOR dataset with this experiment. Then, suppressed cost function was defined for the two hidden layer artificial neural Networks. For this suppressed cost function's first and second hidden layers were kept equal. So, the experiment which is done for one hidden layer artificial neural network was done for two hidden layer artificial neural networks. With this experiment, the ideal number of neuron was detected for XOR dataset. In the conclusion section, the generalization of obtained method is observed.

Keywords: Deep learning, artificial neural networks, the number of hidden layers, XOR problem.

2019, 96 pages

İÇİNDEKİLER	<u>Sayfa</u>
ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİL DİZİNİ	vi
TABLO DİZİNİ	viii
Sayfa	viii
ÖNSÖZ	ix
1. GİRİŞ	1
1.1. Makine Öğrenmesi	1
1.2. Yapay Zeka	1
1.3. Derin Öğrenme.....	2
1.4. Problem	2
1.5. Literatür Taraması	3
2. LOJİSTİK REGRESYON	4
2.1. İkili Sınıflandırma	4
2.1.1. İkili Sınıflandırma Hipotezi	5
2.1.2. Sigmoid Fonksiyonu	5
2.1.3. Kayıp(Loss) Fonksiyonu	6
2.1.4. Maliyet(Cost) Fonksiyonu	8
2.1.5. Meyilli Azalım(Gradient Descent).....	9
2.1.6. İleri Yayılım	12
2.1.7. Geri Yayılım.....	13
2.1.8. Lojistik Regresyon Algoritmasında Öğrenme	15
2.2. İkili Sınıflandırma İçin Bir Örnek.....	15
2.3. XOR Problemi.....	16
2.4. Lojistik Regresyon İle XOR Probleminin Sınıflandırılması.....	18
3. YAPAY SINIR AĞLARI	20
3.1. Bir Gizli Katmanlı Yapay Sinir Ağları	22
3.1.1. Vektörleştirme.....	22
3.1.2. Aktivasyon Fonksiyonları	24
3.1.3. Relu(Rectified Linear Unit)	25
3.1.4. Bir Gizli Katmanlı Yapay Sinir Ağları İçin Hipotez Tespiti	26

3.1.5. Maliyet Fonksiyonu	26
3.1.6. İleri Yayılım	27
3.1.7. Geri Yayılım.....	28
3.1.8. Meyilli Azalım	32
3.1.9. Bir gizli Katmanlı Yapay Sinir Ağlarında Öğrenme.....	32
3.2. İki Gizli Katmanlı Yapay Sinir Ağları	33
3.2.1. İki Gizli Katmanlı Yapay Sinir Ağları İçin Hipotez Tespiti	33
4. XOR PROBLEMİNİN BİR GİZLİ KATMANLI YAPAY SİNİR AĞLARI İLE SINIFLANDIRILMASI	34
4.1. Gizli Katmanındaki Nöron Sayısı Bir Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	34
4.2. Gizli Katmanındaki Nöron Sayısı İki Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	36
4.3. Gizli Katmanındaki Nöron Sayısı Üç Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	38
4.4. Gizli Katmanındaki Nöron Sayısı Dört Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	40
4.5. Gizli Katmanındaki Nöron Sayısı Beş Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	42
4.6. Gizli Katmanındaki Nöron Sayısı Altı Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	44
4.7. Gizli Katmanındaki Nöron Sayısı Yedi Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	46
4.8. Gizli Katmanındaki Nöron Sayısı Sekiz Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	48
4.9. Gizli Katmanındaki Nöron Sayısı Dokuz Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	50
4.10. Gizli Katmanındaki Nöron Sayısı On Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması	54
5. BASKI MALİYET FONKSİYONU	58
5.1. BASKI MALİYET FONKSİYONUNUN UYGULANMASI	59
5.1.1. $r = 1$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	59
5.1.2. $r = 2$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	60
5.1.3. $r = 3$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	61
5.1.4. $r = 4$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	62
5.1.5. $r = 5$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	63

5.1.6. $r = 6$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	64
5.1.7. $r = 7$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	65
5.1.8. $r = 8$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	66
5.1.9. $r = 9$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	67
5.1.10. $r = 10$ İçin Baskı Maliyet Fonksiyonunun Uygulanması	69
6. DENEY	71
7. XOR PROBLEMİNİN İKİ GİZLİ KATMANLI YAPAY SİNİR AĞLARI İLE SINIFLANDIRILMASI	74
7.1. Birinci Gizli Katmanında $r1 = 1$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	74
7.2. Birinci Gizli Katmanında $r1 = 2$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	75
7.3. Birinci Gizli Katmanında $r1 = 3$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	77
7.4. Birinci Gizli Katmanında $r1 = 4$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	78
7.5. Birinci Gizli Katmanında $r1 = 5$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	80
7.6. Birinci Gizli Katmanında $r1 = 6$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	81
7.7. Birinci Gizli Katmanında $r1 = 7$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	83
7.8. Birinci Gizli Katmanında $r1 = 8$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	84
7.9. Birinci Gizli Katmanında $r1 = 9$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	86
7.10. Birinci Gizli Katmanında $r1 = 10$ ve İkinci Gizli Katmanında Sırasıyla $r2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması.	87
8. İKİ GİZLİ KATMANLI YAPAY SİNİR AĞLARI İÇİN BASKI MALİYET FONKSİYONU	90
9. İKİ GİZLİ KATMANLI YAPAY SİNİR AĞLARI İÇİN BİR DENEY	92
10. SONUÇ	94
11. KAYNAKÇA	95

ŞEKİL DİZİNİ

Sayfa

Şekil 2.1. İkili Sınıflandırma Algoritması İçin Bir Gösterim	4
Şekil 2.2. Sigmoid Fonksiyonu	5
Şekil 2.3. $y=0$ için Kayıp Fonksiyonunun Grafiği	7
Şekil 2.4. $y=1$ için Kayıp Fonksiyonunun Grafiği	8
Şekil 2.5. Maliyet Fonksiyonunun Temsili Gösterimi	9
Şekil 2.6. Bir parametrelili meyilli azalım.....	11
Şekil 2.7. İleri Yayılım Hesap Diyagramı	12
Şekil 2.8. Geri Yayılım Hesap Diyagramı.....	13
Şekil 2.9. Tensorflow Playground Gauss Veri Kümesi.....	15
Şekil 2.10. Gauss Verisi İçin Lojistik Regresyon.....	16
Şekil 2.11. XOR Problemi.....	17
Şekil 2.12. XOR Probleminin Genişletilmesi.....	17
Şekil 2.13. Uygulama İçin XOR Problemi Veri Kümesi Grafiği.....	18
Şekil 2.14. XOR Veri Kümesi İçin Lojistik Regresyon	19
Şekil 3.1. Yapay Sinir Ağlarının Genel Bir Gösterimi.....	21
Şekil 3.2. Bir Gizli Katmanlı Yapay Sinir Ağı.....	22
Şekil 3.3. Relu Fonksiyonu	25
Şekil 3.4. İleri Yayılım Hesap Diyagramı	27
Şekil 3.5. Geri Yayılım Hesap Diyagramı.....	28
Şekil 3.6. İki Gizli Katmanlı Yapay Sinir Ağı Mimarisi.....	33
Şekil 4.1. XOR Eğitim Kümesi Görseli	34
Şekil 4.2. Bir Gizli Katmanlı ve Bir gizli Nöronlu Mimari.....	34
Şekil 4.3. $r = 1$ için XOR'un Sınıflandırılması	35
Şekil 4.4. Bir Gizli Katmanlı ve İki gizli Nöronlu Mimari	36
Şekil 4.5. $r = 2$ İçin XOR'un Sınıflandırılması	37
Şekil 4.6. Bir Gizli Katmanlı ve Üç gizli Nöronlu Mimari	38
Şekil 4.7. $r = 3$ için XOR'un Sınıflandırılması	39
Şekil 4.8. Bir Gizli Katmanlı ve Dört gizli Nöronlu Mimari	40
Şekil 4.9. $r = 4$ için XOR'un Sınıflandırılması	41
Şekil 4.10. Bir Gizli Katmanlı ve Beş Gizli Nöronlu Mimari.....	42
Şekil 4.11. $r = 5$ için XOR'un Sınıflandırılması	43
Şekil 4.12. Bir Gizli Katmanlı ve Altı Gizli Nöronlu Mimari.....	44
Şekil 4.13. $r = 6$ için XOR'un Sınıflandırılması	45

Şekil 4.14. Bir Gizli Katmanlı ve Yedi Gizli Nöronlu Mimari	46
Şekil 4.15. $r=7$ için XOR'un Sınıflandırılması	47
Şekil 4.16. Bir Gizli Katmanlı ve Sekiz Gizli Nöronlu Mimari	48
Şekil 4.17. $r=8$ için XOR'un Sınıflandırılması	49
Şekil 4.18. Bir Gizli Katmanlı ve Dokuz Gizli Nöronlu Mimari	51
Şekil 4.19. $r=9$ için XOR'un Sınıflandırılması	53
Şekil 4.20. Bir Gizli Katmanlı ve On Gizli Nöronlu Mimari	54
Şekil 4.21. $r=10$ için XOR'un Sınıflandırılması	55
Şekil 4.22. Nöron Sayısı – Hata Değeri Grafiği	57
Şekil 6.1. Deney Sonucu	72
Şekil 7.1. $r_1 = 1$ ve $r_2 = 2$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	75
Şekil 7.2. $r_1 = 2$ ve $r_2 = 6$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	76
Şekil 7.3. $r_1 = 3$ ve $r_2 = 9$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	78
Şekil 7.4. $r_1 = 4$ ve $r_2 = 10$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	79
Şekil 7.5. $r_1 = 5$ ve $r_2 = 8$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	81
Şekil 7.6. $r_1 = 6$ ve $r_2 = 9$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	82
Şekil 7.7. $r_1 = 7$ ve $r_2 = 9$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	84
Şekil 7.8. $r_1 = 8$ ve $r_2 = 10$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	85
Şekil 7.9. $r_1 = 9$ ve $r_2 = 9$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	87
Şekil 7.10. $r_1 = 10$ ve $r_2 = 8$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma	88
Şekil 7.11. r_1 ve r_2 Nöronları İçin Hata Değeri Grafiği	89
Şekil 9.1. Deney Sonucu	92

TABLO DİZİNİ**Sayfa**

Tablo 6.1. Deneysel Sonuçları	72
Tablo 7.1. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	74
Tablo 7.2. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	76
Tablo 7.3. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	77
Tablo 7.4. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	78
Tablo 7.5. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	80
Tablo 7.6. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	81
Tablo 7.7. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	83
Tablo 7.8. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	84
Tablo 7.9. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	86
Tablo 7.10. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları	87
Tablo 9.1. Maliyet ve Baskı Maliyet Değerleri	93

ÖNSÖZ

Tez çalışması süresince kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan değerli danışman hocam Doç. Dr. Ünver ÇİFTÇİ'ye teşekkürü bir borç bilirim.

Çalışmalarım boyunca manevi desteğiyle beni hiçbir zaman yalnız bırakmayan eşim, annem, babam ve kardeşime sonsuz teşekkür ederim.

1. GİRİŞ

Bu bölümde makine öğrenmesi, yapay zeka ve derin öğrenme kavramlarından bahsedilmiştir. Bunun yanında problem açıklanmış ve literatür taraması yapılmıştır.

1.1. Makine Öğrenmesi

Makine Öğrenmesi (Machine Learning), matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarımlar yapan, bu çıkarımlarla bilinmeyene dair tahminlerde bulunan yöntemlere verilen isimdir.

Kökleri 20. yüzyılın ortalarına uzanan makine öğrenimi, 1950’li yıllarda İngiliz matematikçi Alan Turing tarafından ortaya çıkarılmıştır.

İki temel öğrenme türü vardır:

- **Gözetimli Öğrenme**

Bu öğrenme tekniği etiketli (labelled) yani girdi ve çıktı değerlerine sahip veri içeren veri kümelerine uygulanır. Bu teknikte giriş değerleri ile istenen çıkış değerleri arasında eşleme yapan bir fonksiyon oluşturulur.

- **Gözetimsiz Öğrenme**

Bu öğrenme tekniği işaretlenmemiş (unlabelled) yani etiketsiz veri üzerinden bilinmeyen bir yapıyı tahmin etmek için kullanılan bir makine öğrenmesi tekniğidir. Burada giriş verisinin hangi sınıfa ait olduğu belirsizdir.

Bu tezde kullanılan veri kümeleri etiketli veriye sahip olduğundan gözetimli öğrenme tekniği kullanılacaktır.

1.2. Yapay Zeka

Makine öğrenmesi yapay zekanın bir alanıdır. Yapay zeka, insan zekâsına özgü olan algılama, öğrenme, düşünme, fikir yürütme, sorun çözme, iletişim kurma, çıkarım yapma ve karar verme gibi yüksek bilişsel fonksiyonları veya otonom davranışları sergilemesi beklenen yapay bir işletim sistemidir. Bu sistem aynı zamanda düşüncelerinden tepkiler üretebilmeli ve bu tepkileri fiziksel olarak dışa vurabilmelidir.

Günümüzde hızla gelişen grafik işlemci teknolojisi sayesinde, derin öğrenme algoritmaları grafik işlemciler üzerinde genel amaçlı işlemcilere kıyasla 10 ile 20 kat arasında

daha hızlı bir şekilde eğitilebilmektedir. Bu teknoloji sayesinde 2000'lerin başında Derin Öğrenme, yapay zekâ alanında çığır açmış ve yaygınlaşmaya başlamıştır.

1.3. Derin Öğrenme

Derin öğrenme yapay sinir ağları üzerinde gerçekleştirilen öğrenme teknikleri felsefesidir. Makine öğrenmesinden farklı olarak derin öğrenmede çok katmanlı yapay sinir ağları kullanılmaktadır. Derin öğrenmeyi başarılı kılan da bu çok katmanlı yapay sinir ağları teknikleridir.

Günümüzün büyük yazılım firmalarından Google, Microsoft, Facebook, Amazon, gibi yazılım devleri ar-ge çalışmalarının büyük bir kısmını derin öğrenme üzerine yapmaktadır. Buradaki amaç, yazılımların insanlar gibi öğrenmesini sağlamaktır. Bu bağlamda büyük firmalar her geçen gün yeni ürünlerini piyasaya sürmektedir. Facebook, videolara ve resimlere derin öğrenme ile resim filtreleri uygulayarak gerçek zamanlı video ve resim dönüşümleri yapmaktadır. Bunun yanı sıra derin öğrenme ile teknolojinin ulaştığı nokta sürücüsüz araçlar, röntgenleri yorumlayıp kanser teşhisi, fotoğraf tanıma gibi birçok çalışma alanı mevcuttur.

1.4. Problem

Derin öğrenme mimarisi ile tanınan sistemlerin performansı son zamanlarda müthiş bir artış gösterdi (Goodfellow ve ark. 2016). Ancak bu mimarinin başarısının matematiksel sonucu tasvir edilememektedir (Goodfellow ve ark. 2016). Bu yüzden ideal mimariyi bulmak için birçok deneme yapılmalıdır. Bir yapay sinir ağı modeline yalnız bir tane sinir hücresinin eklenmesi veya çıkarılmasıyla parametreler, hata oranı ve hatta sonuç bile değişebilmektedir. Bir yapay sinir ağına bunun gibi birçok sinir hücresi eklenebilir veya çıkarılabilir. Ancak en iyi tahmini yapabilmek, hatayı minimum seviyeye indiren modeli bulmak oldukça zordur. Bu problemin çözülmesiyle bu anlamda büyük bir aşama kaydedilecek ve birçok model denemek zorunda kalınmayacaktır. Çünkü sayısız yapay sinir ağı modeli oluşturulabilir.

Bu çalışmanın amacı bir ve iki gizli katmanlı yapay sinir ağı için ideal nöron sayısını bulmak için matematiksel bir inceleme yapmaktır.

1.5. Literatür Taraması

Derin öğrenmede yapay sinir ağı mimarisi, regularizasyon ve optimizasyon algoritması olmak üzere üç ana faktör vardır(Vidal ve ark. 2017). Bu faktörlerin hepsinin birleştiği nokta maliyet fonksiyonunun küçültülmesidir. Bir yapay sinir ağı mimarisinin tasarımının en önemli özelliği, verilen girdilere keyfi fonksiyonlarla yaklaşmaktır(Vidal ve ark. 2017). Ancak bu özellik yapay sinir ağı mimarisinin parametrelerine bağlıdır(Vidal ve ark. 2017). Yani ne kadar katman ve bu katmanlarda ne kadar nöron olduğuna bağlıdır.

Yapay sinir ağlarının matematiksel incelenmesi ile ilgili kaynaklar sınırlıdır. Problem henüz açık bir araştırma konusudur. Çalışmaların bir çoğu deneyseldir. İncelenilen bazı kaynaklar şunlardır; Benjamin D. Haefele ve Ren'e Vidal "Global Optimality in Neural Network Training" makalesiyle konveks olmayan yapay sinir ağları problemi üzerinde çalışmışlardır. Bazı koşullar altında yeterince geniş ağlar için herhangi bir lokal minimum noktasının bazı girdileri(entries) sıfır ise bu noktanın aynı zamanda global minimum olduğunu göstermişlerdir(Haefele ve Vidal 2017). Kenji Kawaguchi ve Leslie Pack Kaelbling "Elimination of All Bad Local Minima in Deep Learning" makalesiyle nöron ekleyerek lokal minimumların elenebileceği ispatlanmıştır(Kawaguchi ve Kaelbling 2019). Jeff Heaton "Introduction to Neural Networks for C#" kitabında katmanlardaki gizli nöron sayısını belirlerken birçok metot olduğunu belirterek bunlardan üçüne değinmiştir(Heaton 2008). Birincisi, gizli nöron sayısının girdi ve çıktı boyutu arasında olduğu, ikinci metot olarak gizli nöron sayısının girdi katmanıyla çıktı katmanının toplamının $\frac{2}{3}$ 'ü kadar olacağı ve son olarak girdi boyutunun iki katından küçük olması gerektiğinden bahsetmiştir(Heaton 2008). Ariel Gordon, Elad Eban, Ofir Nachum, Bo Chen, Hao Wu, Tien-Ju Yang ve Edward Choi "MorphNet: Fast & Simple Resource-Constrained Structure Learning of Deep Networks" makalesiyle performansı arttırmak adına yapay sinir ağlarının mimarisini otomatik olarak tasarlama ile ilgilenmişlerdir.

Yapay sinir ağları ile lojistik regresyon arasında bir benzerlikten söz edilebilir. Lojistik regresyonda kullanılan çoğu fonksiyon yapay sinir ağlarında da kullanılmaktadır. Bu yüzden yapay sinir ağlarına giriş yapmadan lojistik regresyondan bahsedilmiştir.

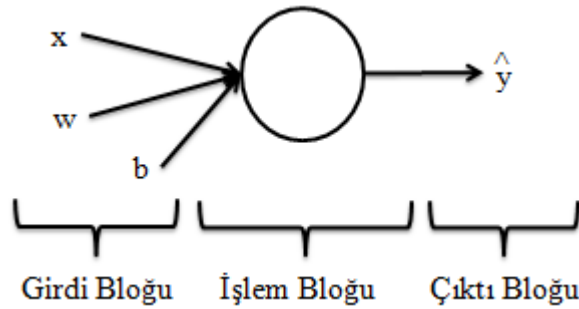
2. LOJİSTİK REGRESYON

Lojistik regresyon bir sınıflandırma algoritmasıdır(Andrew 2017). Görüntü kümesi ayrık olan bir veri kümelerini kategorize etmek için kullanılır. İkili sınıflandırma ve çoklu sınıflandırma olmak üzere ikiye ayırabiliriz. Bu tezde ikili sınıflandırma ele alınacaktır.

2.1. İkili Sınıflandırma

Görüntü kümesi ayrık ve iki elemanı olan bir veri kümesini kategorize etmek için kullanılan bir algoritmadır. İkili denilmesinin sebebi görüntü kümesinin iki elemanlı olmasındandır. Genellikle pozitif sınıflar için “1”, negatif sınıflar için “0” değeri kullanılır(Andrew 2017). Örneğin, bir tümörün iyi huylu olup olmadığı, gelen bir e-postanın spam olup olmadığı gibi problemler ikili sınıflandırma algoritmasıyla öğrenilir(Andrew 2017).

x girdi, \hat{y} tahmin değeri ve w , b parametre olmak üzere Şekil 2.1’de ikili sınıflandırma algoritmasının bir gösterimi verilmiştir.



Şekil 2.1. İkili Sınıflandırma Algoritması İçin Bir Gösterim

Kısaca lojistik regresyon algoritması şöyle işlemektedir;

İlk önce verilen veri kümesine göre başlangıç parametreleri atanır. Daha sonra verilerin girdileri teker teker işlem bloğuna girerek her bir veri için bir tahmin yapılır ve bu tahminler verilerin çıktıları ile karşılaştırılarak ne kadar hata yapıldığı bulunur. Hata değeri büyük ise parametreler güncellenip tekrar tahmin yapılır. Bu işlem hata değeri yeterince küçük oluncaya kadar devam eder. En son bulunan parametreler aranan parametrelerdir. Böylece algoritma verilen veriyi öğrenmiş olur. Veri kümesinde olmayan bir veri bu algoritmaya girdiğinde, bulunan parametreler ile hangi kategoriye ait olduğuna dair bir tahmin yapılabilir. Sonuç olarak hangi kategoriye ait olduğu bilinmeyen bir veri kategorize edilmiş olur.

İkili sınıflandırma algoritmasının yakından incelenilmesi adına öncelikle ikili sınıflandırma hipotezinden bahsedilmiştir.

2.1.1. İkili Sınıflandırma Hipotezi

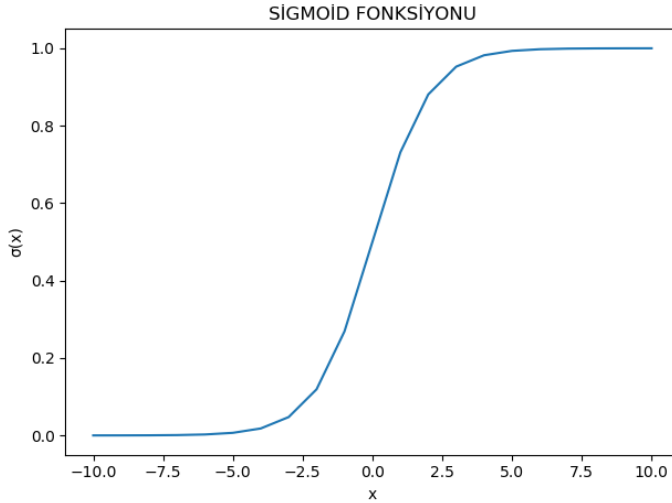
İkili sınıflandırmada hipotez; girdilerin işlem bloğuna girdiğinde çıktı yaklaşık olarak “0” veya “1” değerlerini vermesidir. Bu hipotez için sigmoid fonksiyonu(2.1.2) kullanılır. Sigmoid fonksiyonunun kullanılmasının sebebi hipotezi anlamlı kılmasıdır. Öyle ki sigmoid fonksiyonun görüntü kümesi (0,1) aralığındadır. Bu aralıkta aldığı değerler de olasılık olarak yorumlanır.

2.1.2. Sigmoid Fonksiyonu

$$\sigma: \mathbb{R} \rightarrow (0,1)$$

$$x \rightarrow \sigma(x) = \frac{1}{1 + e^{-x}}$$

fonksiyonuna sigmoid fonksiyonu denir. σ sembolü ile gösterilir. Grafiği Şekil 2.2’de verilmiştir.



Şekil 2.2. Sigmoid Fonksiyonu

Sigmoid Fonksiyonunun Türevi:

$$\begin{aligned} \frac{d\sigma(x)}{dx} &= \frac{d[(1 + e^{-x})^{-1}]}{dx} = (-1)(1 + e^{-x})^{-2}(-e^{-x}) = \frac{e^{-x}}{(1 + e^{-x})^2} \\ &= \left(\frac{1}{1 + e^{-x}}\right) \left(\frac{e^{-x}}{1 + e^{-x}}\right) = \sigma(x) \left(1 - \frac{1}{1 + e^{-x}}\right) = \sigma(x)(1 - \sigma(x)) \end{aligned}$$

Sigmoid Fonksiyonunun Olasılık Olarak Değerlendirilmesi:

Sigmoid fonksiyonu olasılık olarak

$$\sigma: \mathbb{R} \rightarrow (0,1)$$

$$(wx + b) \rightarrow \sigma(wx + b) = p(y = 1 | x; w)$$

şeklinde değerlendirilir. Örneğin $\sigma(x) = 0.7$ hesaplınsın. O halde x girdisinin $y = 1$ olma olasılığı %70 olur denilir. Olasılık değerlerinin toplamı “1” olacağından,

$$p(y = 0 | x; w) + p(y = 1 | x; w) = 1 , \quad (1)$$

olur. (1)'den dolayı $y = 0$ olma olasılığı

$$p(y = 0 | x; w) = 1 - p(y = 1 | x; w)$$

olur. Yukarıda verilen örneğe göre $p(y = 0 | x; w) = 0.3$ olur. O halde x girdisinin $y = 0$ olma olasılığı %30 olduğu sonucuna varılır.

Sigmoid fonksiyonu ile yapılan tahminleri gerçek çıktılar ile karşılaştırıp yapılan hatanın hesaplanması için kayıp fonksiyonu(2.1.3.) kullanılır.

2.1.3. Kayıp(Loss) Fonksiyonu

Verilen bir veri kümesindeki her bir veri için yapılan hatayı hesaplayan fonksiyondur. Kayıp fonksiyonunun elde edilişi:

$$\hat{y} = \sigma(wx + b) = p(y = 1 | x; w) \text{ olmak üzere,}$$

(1) Eğer $y = 1$ ise $p(y = 1 | x; w) = \hat{y}$ olacaktır.

(2) Eğer $y = 0$ ise $p(y = 0 | x; w) = 1 - \hat{y}$ olacaktır. Çünkü olasılıkların toplamı “1”e eşit olması gerekir.

$$(3) \quad p(y|x) = \hat{y}^y \cdot (1 - \hat{y})^{1-y}$$

(3) denkleminde görüldüğü gibi (1) ve (2) satırları bir satırda yazılabilir. Çünkü $y = 1$ olduğunda (1) denklemi $p(y = 1 | x; w) = \hat{y}$ ve $y = 0$ olduğunda (2) denklemi $p(y = 0 | x; w) = 1 - \hat{y}$ elde edilmektedir.

(3) denklemin $y = 1$ olduğunda $p(y = 1 | x; w) = \hat{y}$ olduğundan hatayı aza indirmek için tahmin değerimizi maksimize etmemiz gerekmektedir. Aynı şekilde (3) denkleminden

$y = 0$ olduğunda $p(y = 0 | x; w) = 1 - \hat{y}$ olur ve verinin çıktısı 0 olduğundan hatayı aza indirmek için tahmin değerimizi yine maksimize etmemiz gerekmektedir ki olasılık “0” çıksın ya da sıfıra yaklaşsın. O halde hatanın minimum olabilmesi için (3) denklemi maksimize edilmelidir.

Problemin optimizasyon problemine dönüştürülmesi adına (3) denkleminin her iki eşitliğinin logaritması alınırsa

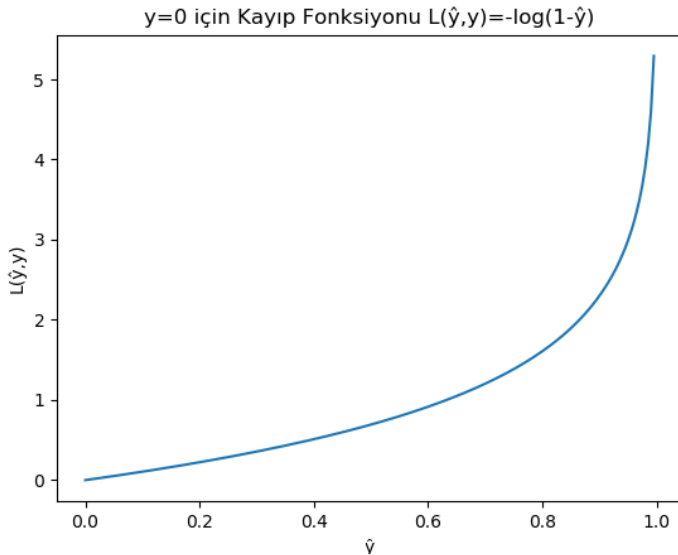
$$\begin{aligned}\log p(y|x) &= \log(\hat{y}^y \cdot (1 - \hat{y})^{1-y}) \\ &= \log \hat{y}^y + \log(1 - \hat{y})^{1-y} = y \log \hat{y} + (1 - y) \log(1 - \hat{y})\end{aligned}$$

denklemi elde edilir. Logaritma fonksiyonunun monoton artan özelliğinden bu denklem maksimize edildiğinde (3) denklemi ile aynı sonucu verir. Eğer eşitliğin sağ tarafını “-” ile çarpılır ve eşitliğin sağ tarafı minimize edildiğinde (3) denklemi maksimize edilmiş olacaktır. O halde kayıp fonksiyonu,

$$L(\hat{y}, y) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$$

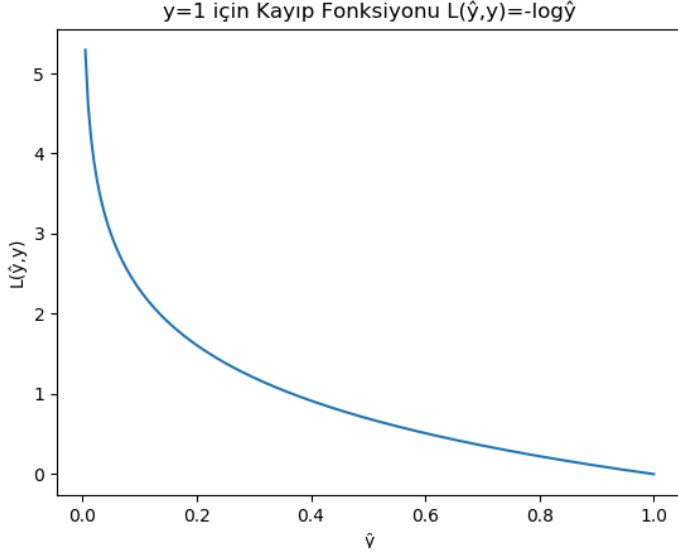
olarak tanımlanır ve L harfi ile gösterilir[2].

Eğer $y = 0$ ise $L(\hat{y}, y) = -\log(1 - \hat{y})$ olur ve Şekil 2.3’ de grafiği verilmiştir. Burada \hat{y} olabildiğince küçültülmelidir ki L fonksiyonu yani kayıp fonksiyonu küçük çıksın.



Şekil 2.3.y=0 için Kayıp Fonksiyonunun Grafiği

Eğer $y = 1$ ise $L(\hat{y}, y) = -\log \hat{y}$ olur ve Şekil 2.4’de grafiği verilmiştir. Burada da \hat{y} olabildiğince büyütülmelidir ki kayıp fonksiyonu küçük çıksın.



Şekil 2.4. $y=1$ için Kayıp Fonksiyonunun Grafiği

Şekil 2.3 ve Şekil 2.4’den görülmektedir ki kayıp fonksiyonu konvektir.

Hata değeri her bir örnek için ayrı olacaktır. O halde verilen $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$ veri kümesi için kayıp fonksiyonu,

$$L(\hat{y}^{(i)}, y^{(i)}) = -[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})], \quad 1 \leq i \leq m,$$

şeklinde hesaplanacaktır(Andrew 2017).

Veri kümesindeki her bir veri için bir hata değeri bulunur ve bu hata değerlerinin ortalaması algoritmanın maliyetini verir. Bu maliyet aşağıda maliyet fonksiyonu ile hesaplanmaktadır.

2.1.4. Maliyet(Cost) Fonksiyonu

Verilen veri kümesindeki her bir veri için yapılan hataların ortalamasını veren fonksiyondur. Kayıpların tüm veri kümesine yaymak ve ortalama bir hata oluşturmak için maliyet fonksiyonu,

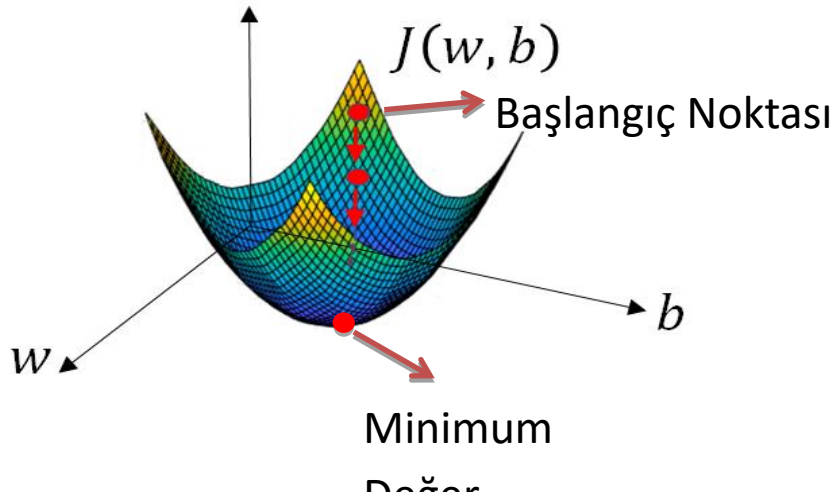
$$J(w, b) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$$

olarak tanımlanır(Andrew 2017).

Maliyet fonksiyonu ile verilen veriler için bir maliyet değeri bulunur. Maliyet fonksiyonu w ve b parametrelerine bağlı olduğundan öyle parametreler bulunmalıdır ki maliyet çok küçük olsun ve algoritma veri setini öğrenebilsin. Bu parametreler meyilli azalım yöntemi ile bulunur.

2.1.5. Meyilli Azalım(GradientDescent)

Meyilli azalım (gradientdescent) maliyet fonksiyonunu minimize etmek için kullanılan bir yöntemdir. Şekil 2.5’de maliyet fonksiyonunun temsili bir gösterimi verilmiştir.



Şekil 2.5.Maliyet Fonksiyonunun Temsili Gösterimi

Maliyet fonksiyonu konveks seçilmiştir. Bunun sebebi maliyet fonksiyonunun tek yerel minimumunun olmasıdır. Böylece nerede bir başlangıç noktası seçilirse seçilsin aynı minimum noktası elde edilecektir. Yukarıdaki kırmızı nokta başlangıç parametresi olarak seçilirse, meyilli azalım yöntemi maliyet fonksiyonuna uygulanarak maliyet fonksiyonunun minimum değeri alması için uygun parametreler hesaplanır.

Meyilli azalım yöntemini daha ayrıntılı incelemek adına maliyet fonksiyonu bir boyutlu olsun. Yani b parametresi devre dışı olsun. O halde fonksiyon $J(w)$ olacaktır. Meyilli azalım yöntemi, w başlangıç noktası ve α (yeterince küçük pozitif bir sayı) adım büyüklüğü olmak üzere w 'nin güncelleme,

$$w := w - \alpha \frac{dJ(w)}{dw}$$

ifadesinin tekrar etmesiyle gerçekleşir. $\frac{dJ(w)}{dw}$ ifadesi $J(w)$ 'nin w 'ye göre türevidir. $1 \leq i \leq m$ için $\hat{y}^{(i)} = \sigma(wx^{(i)})$ olduğundan ve maliyet fonksiyonunun tanımından

$$\frac{dJ(w)}{dw} = -\frac{1}{m} \sum_{i=1}^m \frac{dL(\hat{y}^{(i)}, y^{(i)})}{dw}$$

olur.

$J(w)$ fonksiyonu konveks olduğundan grafiği Şekil 2.6'daki gibi olacaktır. Başlangıç noktası w olmak üzere eğer başlangıç noktasını sağdan seçilirse, eğim sağa yatık olduğundan pozitif çıkacaktır. w noktasındaki eğim aynı zamanda türev olacağından $\frac{dJ(w)}{dw}$ ifadesi de pozitif olacaktır.

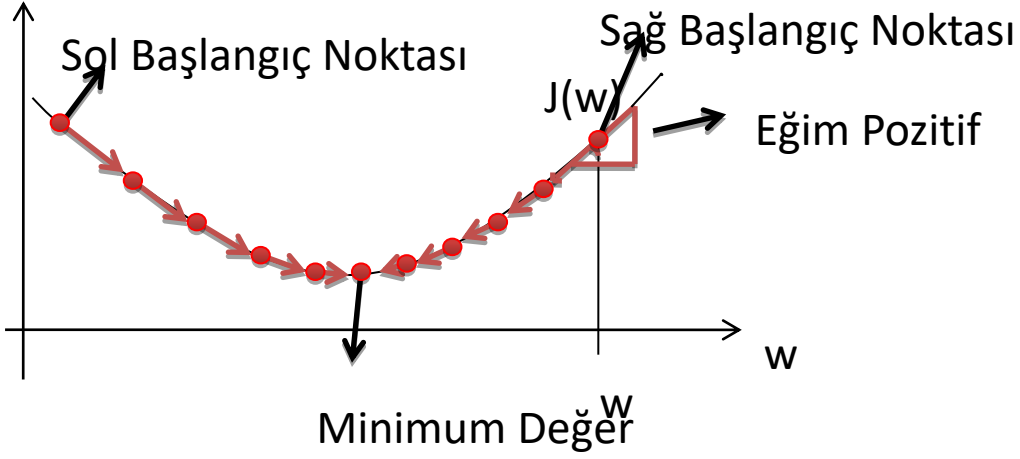
$$w := w - \alpha \frac{dJ(w)}{dw} \rightarrow w := w - \alpha(\text{pozitif sayı}) \rightarrow w \text{ azalır.}$$

Bu işlem tekrar ettirildiği takdirde $J(w)$ minimum değere yaklaşılabilecektir. Şekil 2.6'de gösterilmiştir.

Eğer başlangıç noktasını soldan seçilirse, eğim sola yatık dolayısıyla eğim negatif olacaktır.

$$w := w - \alpha \frac{dJ(w)}{dw} \rightarrow w := w - \alpha(\text{negatif sayı}) \rightarrow w \text{ artar.}$$

Bu işlem tekrar ettiği takdirde her adımda minimum değere biraz daha yaklaşmış olur. Böylece $J(w)$ değeri için minimum değer bulunmuş olacaktır. Şekil 2.6'de gösterilmiştir.



Şekil 2.6. Bir parametrelili meyilli azalım

Sonuç olarak $J(w)$ değerini minimum yapan w parametresi bulunmuş olur. Veri kümesi dışından bir girdi algoritmaya girdiğinde bulunan parametre kullanılarak tahmin yapılır ve bu veri kategorize edilir.

Meyilli azalım yöntemi en genel haliyle aşağıdaki gibidir:

$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}) \in \mathbb{R}^n$, $y^{(i)} \in \{0,1\}$, $i = 1,2, \dots, m$, olmak üzere $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, veri kümesi verilsin. O halde en genel meyilli azalım algoritması

$$b := b - \alpha \frac{\partial J(w_1, w_2, \dots, w_n, b)}{\partial b}$$

$$w_1 := w_1 - \alpha \frac{\partial J(w_1, w_2, \dots, w_n, b)}{\partial w_1}$$

$$w_2 := w_2 - \alpha \frac{\partial J(w_1, w_2, \dots, w_n, b)}{\partial w_2}$$

⋮

$$w_n := w_n - \alpha \frac{\partial J(w_1, w_2, \dots, w_n, b)}{\partial w_n}$$

ifadeleri ile gösterilir.

Meyilli azalım yöntemini kullanabilmek için

$$\frac{\partial J(w, b)}{\partial w}, \frac{\partial J(w, b)}{\partial b},$$

türevlerinin bulunması gerekir. Bu türevlerin bulunması için zincir kuralı(Goodfellow ve ark. 2016) kullanılacaktır. Geri yayılım(2.1.7) algoritması ile bu değerler bulunacaktır. Ancak geri yayılımdan önce verilen veri kümesi için bir tahmin yapılması gerekir. Bu tahmin ileri yayılım algoritması ile yapılır.

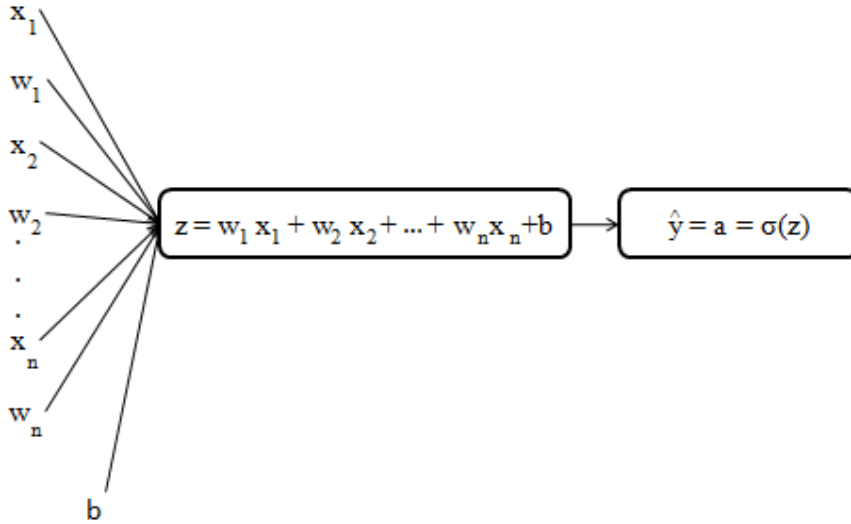
2.1.6. İleri Yayılım

İleri yayılım, girdinin algoritmaya girmesinden çıktının elde edilmesine kadar yapılan işlemlere denir. O halde ileri yayılım, n boyutlu herhangi bir $x = (x_1, x_2, \dots, x_n)$ girdisi ve $w = (w_1, w_2, \dots, w_n), b$ parametreleri için

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n + b,$$

$$\hat{y} = a = \sigma(z),$$

şeklinde tanımlanır.



Şekil 2.7.İleri Yayılım Hesap Diyagramı

İleri yayılım Şekil 2.7’de olduğu gibi görselleştirilebilir. Bu görsele hesap diyagramı denir(Andrew 2017). Yapılan işlemlerin adımlarını görebilmeyi ve hesap yapmayı kolaylaştırır. Hesap diyagramına göre ileri yayılım soldan sağa doğru olan süreçtir.

İleri yayılım ile tahmin yapıldıktan sonra maliyet fonksiyonu ile hata hesaplanır ve meyilli azalım ile maliyet değeri minimize edilir. Ancak meyilli azalım algoritmasının

kullanılması için maliyet fonksiyonunun parametrelere göre türevleri bulunmalıdır. Bu işlemi geri yayılım algoritması yapmaktadır.

2.1.7. Geri Yayılım

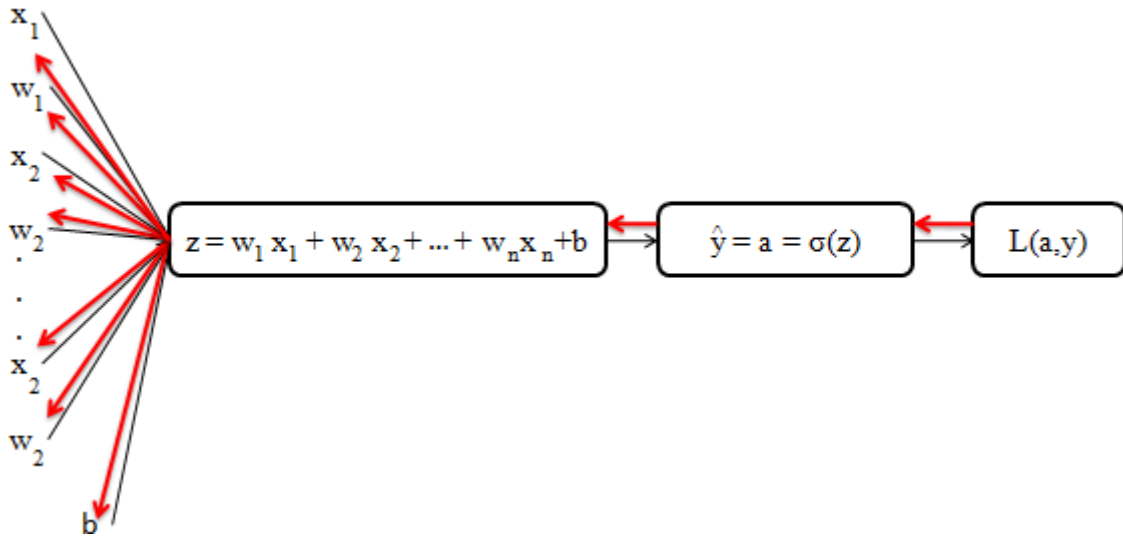
Geri yayılım zincir kuralı(Goodfellow ve ark. 2016) kullanılarak maliyet fonksiyonunun her bir parametreye göre türevinin alınmasıdır. Boyutu n olan herhangi bir $x = (x_1, x_2, \dots, x_n)$ girdisi ve $w = (w_1, w_2, \dots, w_n), b$ parametreleri için

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n + b,$$

$$\hat{y} = a = \sigma(z),$$

$$L(a, y) = -(y \log(a) + (1 - y) \log(1 - a)),$$

olsun.



Şekil 2.8.Geri Yayılım Hesap Diyagramı

Geri yayılım Şekil 2.8’de kırmızı oklar ile gösterilebilir. Bu şekil de geri yayılım için hesap diyagramıdır(Andrew 2017). Yapılan işlemlerin adımlarını görebilmeyi ve hesap yapmayı kolaylaştırır. Bu hesap diyagramına göre geri yayılım sağdan sola doğru kırmızı oklar ile gösterilmiş olan süreçtir. Geri yayılım, parametreleri güncellememiz için çok önemlidir. Burada “Zincir Kuralı” kullanılır(Goodfellow ve ark. 2016).

Zincir kuralı yöntemi ile $a = \hat{y} = \sigma(z)$ olmak üzere

$$\frac{\partial L(a, y)}{\partial w_1} = \frac{\partial L(a, y)}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_1} = \frac{a - y}{a(1 - a)} \cdot a(1 - a) \cdot x_1 = x_1(a - y),$$

$$\frac{\partial L(a, y)}{\partial w_2} = \frac{\partial L(a, y)}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_2} = \frac{a - y}{a(1 - a)} \cdot a(1 - a) \cdot x_2 = x_2(a - y),$$

⋮

$$\frac{\partial L(a, y)}{\partial w_n} = \frac{\partial L(a, y)}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_n} = \frac{a - y}{a(1 - a)} \cdot a(1 - a) \cdot x_n = x_n(a - y),$$

$$\frac{\partial L(a, y)}{\partial b} = \frac{\partial L(a, y)}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial b} = \frac{a - y}{a(1 - a)} \cdot a(1 - a) \cdot 1 = (a - y),$$

ifadeleri aşağıdaki türevler kullanılarak elde edilir.

$$\frac{\partial L(a, y)}{\partial a} = \frac{\partial [-(y \log(a) + (1 - y) \log(1 - a))] }{\partial a} = -\left(\frac{y}{a} - \frac{1 - y}{1 - a}\right) = \frac{a - y}{a(1 - a)},$$

$$\begin{aligned} \frac{\partial a}{\partial z} &= \frac{\partial \sigma(z)}{\partial z} = \frac{\partial \left(\frac{1}{1 + e^{-z}}\right)}{\partial z} = \frac{\partial [(1 + e^{-z})^{-1}]}{\partial z} = \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{1}{(1 + e^{-z})} \cdot \frac{e^{-z}}{(1 + e^{-z})} \\ &= \frac{1}{(1 + e^{-z})} \cdot \left(1 - \frac{1}{1 + e^{-z}}\right) = a(1 - a), \end{aligned}$$

$$\frac{\partial z}{\partial w_1} = \frac{\partial (w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b)}{\partial w_1} = x_1,$$

$$\frac{\partial z}{\partial w_2} = \frac{\partial (w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b)}{\partial w_2} = x_2,$$

⋮

$$\frac{\partial z}{\partial w_n} = \frac{\partial (w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b)}{\partial w_n} = x_n,$$

$$\frac{\partial z}{\partial b} = \frac{\partial (w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b)}{\partial b} = 1.$$

Çalışmanın sonucu olarak

$$\frac{\partial J(w, b)}{\partial w}, \frac{\partial J(w, b)}{\partial b},$$

türevleri elde edilmiştir. O halde artık meyilli azalım kullanılarak “w”, “b” parametreleri bulunabilir ve girdiyle ilgili bir tahminde bulunulabilir.

2.1.8. Lojistik Regresyon Algoritmasında Öğrenme

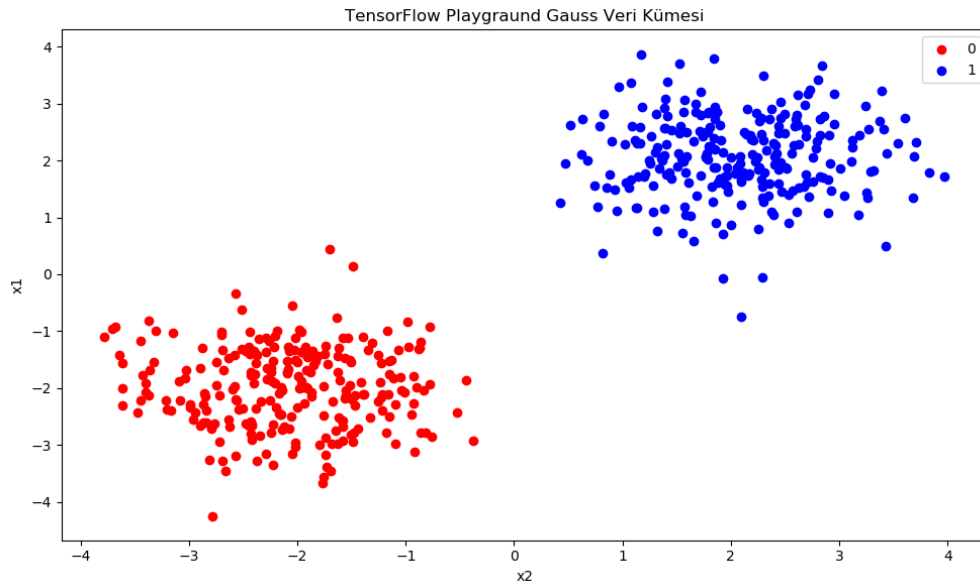
Bir veri kümesi verilsin. Bu veri kümesi için başlangıç parametreleri atanır. Daha sonra ileri yayılım ile tahmin yapılır. Yapılan tahminler için maliyet fonksiyonu kullanılarak hata hesaplanır. Hata değeri yeterince küçük değil ise veri kümesinin öğrenimi için aşağıdaki adımlar takip edilir;

- Geri yayılım,
- Meyilli azalım ile parametrelerin güncellenmesi,
- İleri yayılım,
- Maliyet fonksiyonu.

Yukarıdaki adımlar sırasıyla uygulanır. Maliyet fonksiyonu yeterince küçük oluncaya kadar adımlar başa dönülerek tekrar tekrar uygulanır. Bu tekrarların her birine **epoch** denir. Öğrenim hata değeri yeterince küçük olunca kadar devam eder. Öğrenim tamamlandığında elde edilen parametreler istenilen parametrelerdir. Böylece algoritma verilen veriyi öğrenmiş olur.

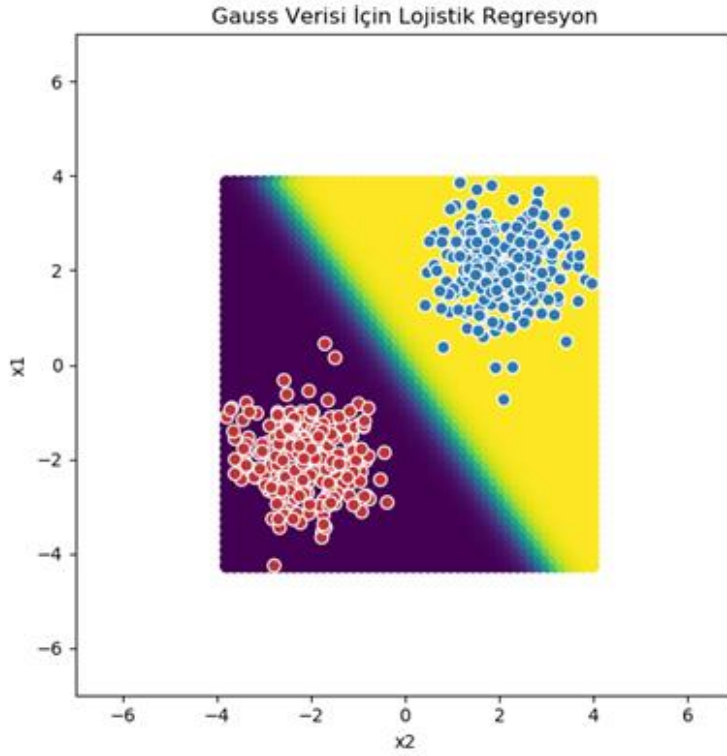
2.2. İkili Sınıflandırma İçin Bir Örnek

İki girdiye sahip 500 tane veri kümesi alınsın(Smilkov ve Carter 2018). Verinin grafiği Şekil 2.9'daki gibidir.



Şekil 2.9.TensorflowPlayground Gauss Veri Kümesi

Verilen veri kümesi lojistik regresyon algoritması ile sınıflandırıldığında Şekil2.10'daki gibi sınıflandırılacaktır.



Şekil 2.10.Gauss Verisi İçin Lojistik Regresyon

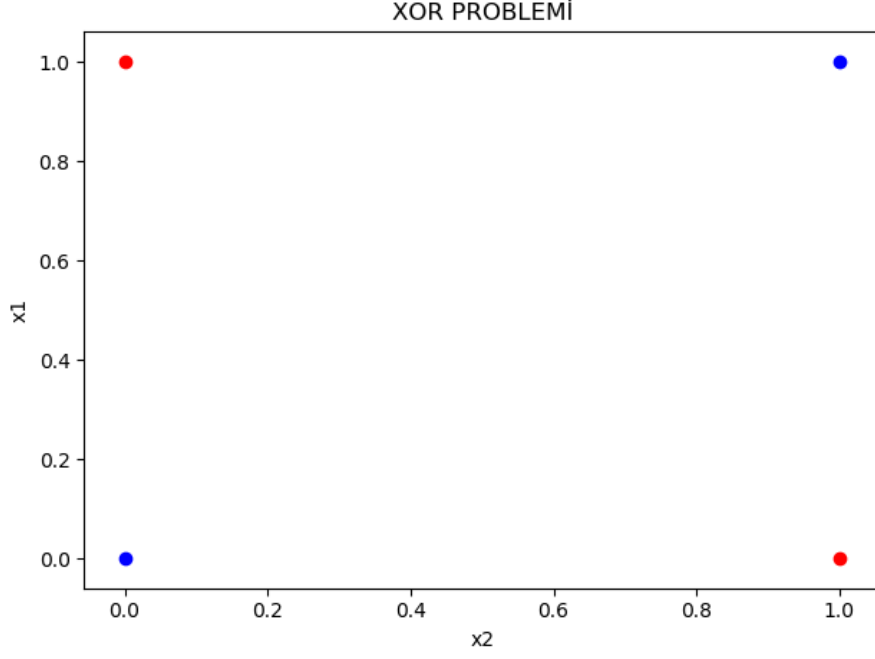
Şekil 2.10’da görüldüğü gibi kolaylıkla verilen veri sınıflandırılmıştır.

2.3. XOR Problemi

XOR problemi yapay sinir ağları arařtırmalarında kullanılan klasik bir sınıflandırma problemidir. Verilen iki girdi ile XOR probleminin çıktıları yapay sinir ağları kullanılarak tahmin edilir. XOR probleminin doğruluk çizelgesi aşağıda verilmiştir.

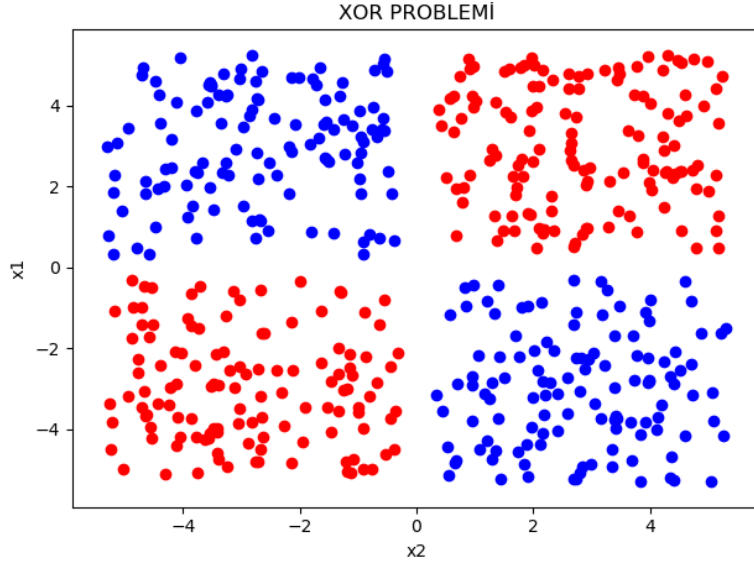
x_1	x_2	y
0	0	0
0	1	1
1	1	0
1	0	1

Şekil 2.11’de XOR probleminin doğruluk çizelgesindeki değerlerinin grafiğı verilmiştir. Bu grafikten görülmektedir ki bir doğruyla kırmızı ve mavi değerler birbirinden ayrılamaz.



Şekil 2.11.XOR Problemi

XOR problemi düzlem üzerine genişletilerek $x^{(i)} = (x_1^{(i)}, x_2^{(i)}) \in \mathbb{R}^2$, $y^{(i)} \in \{0,1\}$, $i = 1, 2, \dots, m$, olmak üzere $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$ m tane veri içeren bir eğitim kümesi verilsin. Bu eğitim kümesinin bir örneği Şekil2.12’de verilmiştir.

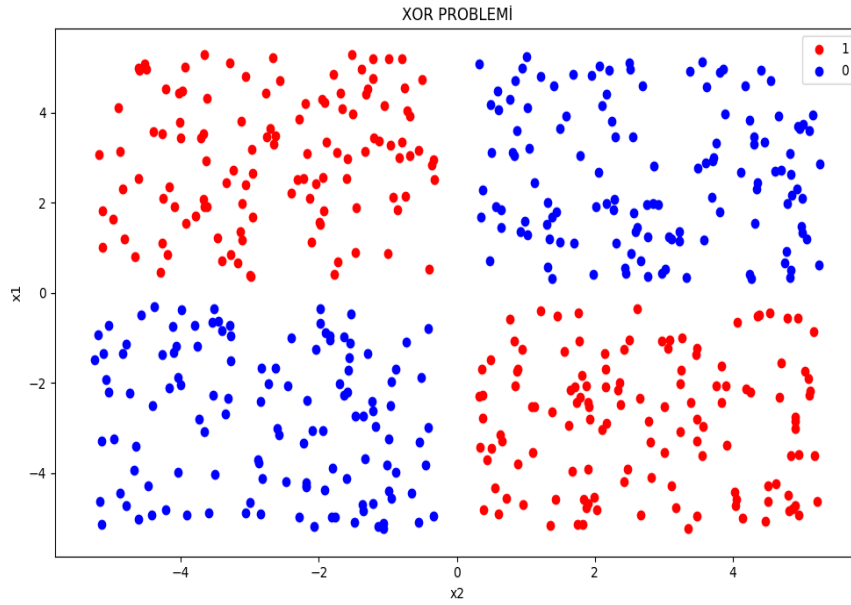


Şekil 2.12.XOR Probleminin Genişletilmesi

Şekil 2.12’de girdilere karşılık gelen kırmızı noktalar “1” çıktısını, mavi noktalar ise “0” çıktısını temsil etmektedir. Yapay sinir ağları bu grafikte görülen kırmızı ve mavi renkteki verileri birbirinden ayırarak sınıflandırabilmektedir.

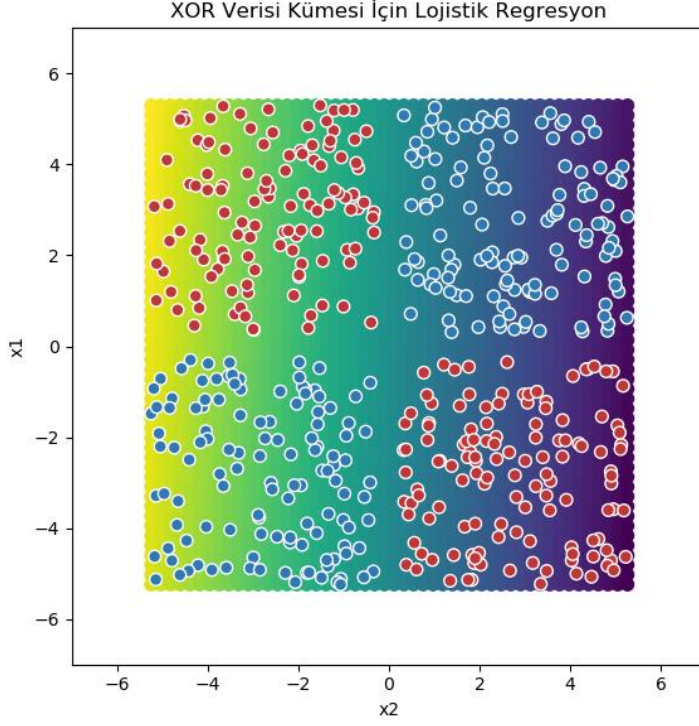
2.4. Lojistik Regresyon İle XOR Probleminin Sınıflandırılması

XOR problemi için tensorflow’un playground uygulamasından bir veri alınmıştır(Smilkov ve Carter 2018). Veri kümesi 500 veri içermektedir. Alınan veri kümesinin grafiği Şekil 2.13’deki gibidir.



Şekil 2.13.Uygulama İçin XOR Problemi Veri Kümesi Grafiği

Verilen veri kümesi lojistik regresyon algoritması ile sınıflandırıldığında Şekil 2.14’de sonuç elde edilmiştir.



Şekil 2.14.XOR Veri Kümesi İçin Lojistik Regresyon

Şekil 2.14’de görüldüğü gibi lojistik regresyon XOR problemini sınıflandıramamaktadır.

Bu bölümde lojistik regresyon hakkında bilgiler verilmiş ve lojistik regresyonun sınıflandırdığı gauss verisi ve sınıflandıramadığı XOR verisi örnekleri ve sonuçları verilmiştir. Lojistik regresyon yapay sinir ağları için çok önemlidir. Yapay sinir ağlarında da benzer yöntemler kullanılacaktır.

3. YAPAY SİNİR AĞLARI

Yapay sinir ağıları, insan beynine has bir özellik olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri, herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir[6]. Yapay sinir ağıları; insan beyninden esinlenerek, öğrenme sürecinin matematiksel olarak modellenmesi uğraşı sonucu ortaya çıkmıştır(Öztemel 2016).

Yapay sinir ağıları üç ana bölümden oluşur;

- **Girdi Katmanı:**

Yapay sinir ağına dış dünyadan girdilerin geldiği katmandır. Bu katmanda, girdi sayısı kadar hücre bulunmaktadır ve girdiler herhangi bir işleme uğramadan gizli katmana iletilirler.

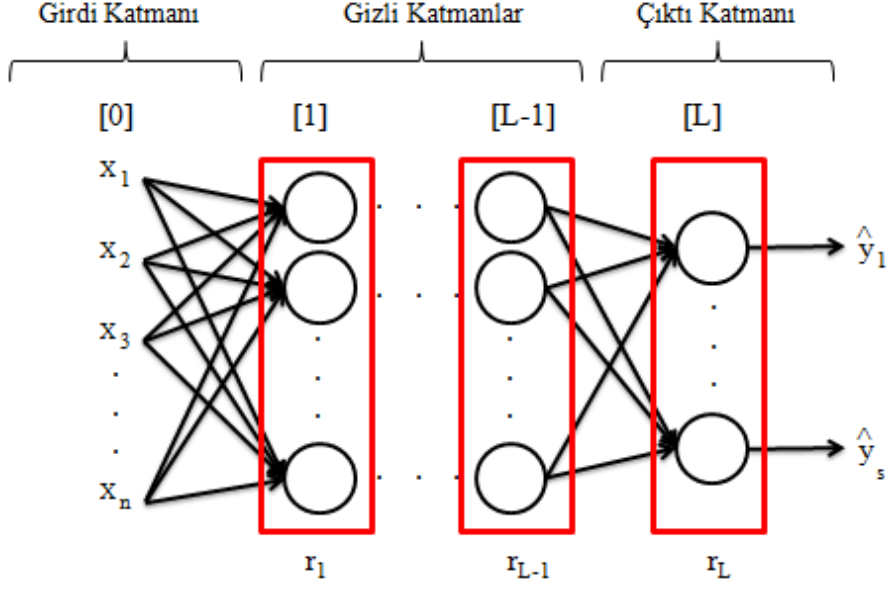
- **Gizli Katmanlar:**

Girdi katmanından aldığı bilgiyi işleyerek bir sonraki katmana iletir. Gizli katman sayısı ve gizli katmandaki hücre sayısı ağdan ağa değişebilir. Gizli katmanlardaki hücre sayıları, girdi ve çıktı sayılarından bağımsızdır.

- **Çıktı Katmanı:**

Gizli katmandan gelen bilgiyi işler ve girdi katmanına gelen girdiye uygun olarak üretilen çıktıyı dış dünyaya gönderir. Çıktı katmanındaki hücre sayısı birden büyük olabilir. Her bir çıktı hücresinin bir adet çıktısı vardır. Her bir hücre bir önceki katmandaki bütün hücrelere bağlıdır.

Yapay sinir ağıları en genel gösterimi Şekil 3.1'de verilmiştir.



Şekil 3.1.Yapay Sinir Ağlarının Genel Bir Gösterimi

Şekil 3.1’deki kırmızı bloklara **katman** denir. Katman numaraları köşeli parantezler içinde tanımlanır. Şekildeki yapay sinir ağı L tane katmana sahip olduğundan L katmanlı yapay sinir ağı veya $L-1$ gizli katmanlı yapay sinir ağı olarak adlandırılır.

NOT: Girdi katmanı isimlendirmede sayılmaz(Andrew 2017).

Katmanlardaki yuvarlar ile gösterilenlere **sinir hücresi** veya **nöron** denir. Nöronların iç yapısında ise lojistik regresyona benzer işlemler olmaktadır(Andrew 2017). Nöron sayısı, katman sayısından bağımsızdır. r_L ile “L.” katmandaki nöron sayısı gösterilmektedir.

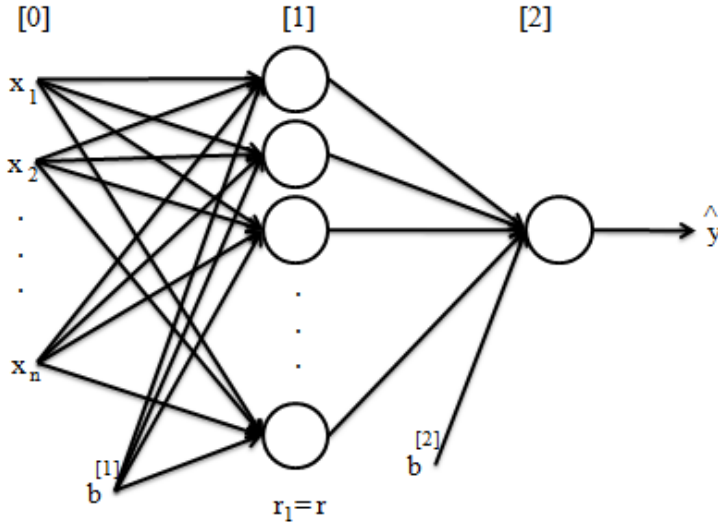
Yapay sinir ağlarında verilen problemin çözümü için kaç katman ve bu katmanlarda kaç tane nöron olacağı konusu şu an için karmaşık bir problemdir. Bu bölümde bir gizli katmanlı ve iki gizli katmanlı yapay sinir ağları incelenecektir. Böylece katman sayısı sabit tutularak yapay sinir ağının gizli katmanındaki ideal nöron sayısını bulmak amaçlanmaktadır.

Veri kümesi olarak XOR problemi ele alınacaktır. XOR problemi seçilmesinin sebebi lojistik regresyonun çözemediği ancak yapay sinir ağlarının çözdüğü bir problem olmasıdır(Andrew 2017).

3.1. Bir Gizli Katmanlı Yapay Sinir Ağları

$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}) \in \mathbb{R}^n$ ve $y^{(i)} \in \{0,1\}$, $i = 1, 2, \dots, m$, olmak üzere $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, veri kümesi verilsin. Bu kümeye aynı zamanda **eğitim kümesi** de denir (Andrew 2017).

$L = 2$ ve $r_1 = r$ nöron sayısı olmak üzere Şekil 3.2’de ikili sınıflandırma problemi için bir gizli katmanlı yapay sinir ağının en genel gösterimi verilmiştir.



Şekil 3.2. Bir Gizli Katmanlı Yapay Sinir Ağı

Bir gizli katmanlı yapay sinir ağı için başlangıç parametreleri $w_r^{[1]} = (w_{r,1}^{[1]}, w_{r,2}^{[1]}, \dots, w_{r,n}^{[1]})$, $b^{[1]} = (b_{1,1}^{[1]}, b_{2,1}^{[1]}, \dots, b_{r,1}^{[1]})$, $w_1^{[2]} = (w_{1,1}^{[2]}, w_{1,2}^{[2]}, \dots, w_{1,r}^{[2]})$, $b^{[2]} = (b_{1,1}^{[2]})$ olur. Buradaki parametreler ile her bir veri işleme koyularak bir tahmin yapılır. Bu işlemler her bir veri için tekrar tekrar yapıldığından elle yapılması çok zaman alır. Bu yüzden bir programlama dili kullanılır. Bu tezde de python programlama dili kullanılmıştır. İşlemler python da vektörler yardımıyla yapılır. Bu yüzden verilen veriye atanan parametreler vektörler ile temsil edilir. Bölüm 3.1.1’ de veri kümesinin ve parametrelerin nasıl vektörleştirildiğinden bahsedilmiştir.

3.1.1. Vektörleştirme

Bu bölümde bir gizli katmanlı yapay sinir ağının vektörleştirilmesi verilecektir. Vektörleştirmenin nedeni, yapay sinir ağı algoritmalarının programlama dillerinde daha hızlı çalışmasını sağladığıdır (Andrew ve ark. 2014).

Eğitim Kümesinin Vektörleştirilmesi:

$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}) \in \mathbb{R}^n$ ve $y^{(i)} \in \{0,1\}$, $i = 1, 2, \dots, m$, olmak üzere $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, eğitim kümesi verilsin.

Eğitim kümesindeki her bir girdi sütun halinde yazılır. O halde i . veri

$$x^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n \times 1}, i = 1, 2, \dots, m,$$

olur. Bütün bu veriler bir matrisin sütunları olacak şekilde yazılır. Bu matris X olarak adlandırılır ve

$$X = \begin{bmatrix} | & | & | & | & | \\ x^{(1)} & x^{(2)} & \dots & x^{(m)} & \\ | & | & | & | & | \end{bmatrix} \in \mathbb{R}^{n \times m}, \quad (2)$$

olur ve X matrisi ile girdiler vektörleştirilmiştir.

Eğitim kümesindeki her bir çıktı sırasıyla bir satır matrisi olarak yazılır ve Y ile gösterilir. O halde,

$$Y = [y^{(1)} \quad y^{(2)} \dots \quad y^{(m)}] \in \mathbb{R}^{1 \times m}, \quad (3)$$

olur ve çıktılar da vektörleştirilmiş olur.

Sonuç olarak (2) ve (3) de görüldüğü gibi eğitim kümesi vektörleştirilmiş olur.

Parametrelerin Vektörleştirilmesi:

Nöron sayısı r olmak üzere birinci katman için her bir nörona giden w parametreleri ayrı, ayrı sütun halinde yazılır. Böylece r tane sütun elde edilir.

$$w_1^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} \\ w_{1,2}^{[1]} \\ \vdots \\ w_{1,n}^{[1]} \end{bmatrix}, w_2^{[1]} = \begin{bmatrix} w_{2,1}^{[1]} \\ w_{2,2}^{[1]} \\ \vdots \\ w_{2,n}^{[1]} \end{bmatrix}, \dots, w_r^{[1]} = \begin{bmatrix} w_{r,1}^{[1]} \\ w_{r,2}^{[1]} \\ \vdots \\ w_{r,n}^{[1]} \end{bmatrix}$$

Daha sonra bu sütunların devriği alınarak bir matrisin satırları olacak şekilde sırasıyla yazılır ve bu matris $W^{[1]}$ olarak adlandırılır. O halde elde edilen matris,

$$W^{[1]} = \begin{bmatrix} - & w_1^{[1]T} & - \\ - & w_2^{[1]T} & - \\ - & \vdots & - \\ - & w_r^{[1]T} & - \end{bmatrix} \in \mathbb{R}^{r \times n}, \quad (4)$$

olur. Birinci katmandaki b parametresi için r tane nörona giden b değerleri sırasıyla sütun vektörü halinde yazılır. $b^{[1]}$ ile gösterilir. O halde elde edilen vektör,

$$b^{[1]} = \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ \vdots \\ b_r^{[1]} \end{bmatrix} \in \mathbb{R}^{rx1}, \quad (5)$$

olur. Böylece birinci katmanın parametreleri $W^{[1]}$ ve $b^{[1]}$ olarak vektörleştirilmiştir.

İkinci katmanın parametrelerinin vektörleştirilmesi ise çıkış katmanı $r_2 = 1$ nörona sahip olduğundan w parametresi bu nörona gelen parametrelerin sütun halinde yazılması ile elde edilir.

$$w_1^{[2]} = \begin{bmatrix} w_{1,1}^{[2]} \\ w_{1,2}^{[1]} \\ \vdots \\ w_{1,r}^{[1]} \end{bmatrix}$$

Daha sonra bu vektörün devriği alınır ve $W^{[2]}$ olarak adlandırılır. O halde,

$$W^{[2]} = \begin{bmatrix} - & w_1^{[2]T} & - \end{bmatrix} \in \mathbb{R}^{1xr}, \quad (6)$$

olur.

İkinci katmandaki b parametresi ise çıktı katmanında bir tane nöron olduğundan bir elemanlıdır. $b^{[2]}$ ile gösterilir. O halde,

$$b^{[2]} = \begin{bmatrix} b_{1,1}^{[2]} \end{bmatrix} \in \mathbb{R}^{1x1}, \quad (7)$$

olur.

Sonuç olarak (4) ve (5) ile birinci katmanın parametreleri, (6) ve (7) ile ikinci katmanın parametreleri vektörleştirilmiş olur.

Verilen veri ve parametreler vektörleştirdikten sonra nöronlarda çeşitli fonksiyonlar kullanılarak işleme koyulur. Bu fonksiyonlardan en çok kullanılanlar bölüm 3.1.2'de verilmiştir.

3.1.2. Aktivasyon Fonksiyonları

Nöronlarda kullanılan fonksiyonlardır. Relu(rectifiedlinearunit) fonksiyonu, sigmoid fonksiyonu(2.1.2.), hiperbolik tanjant fonksiyonu, sızıntı(leaky) relu fonksiyonu, softmax fonksiyonu bir aktivasyon fonksiyonudur(Goodfellow ve ark. 2016).Modern yapay sinir

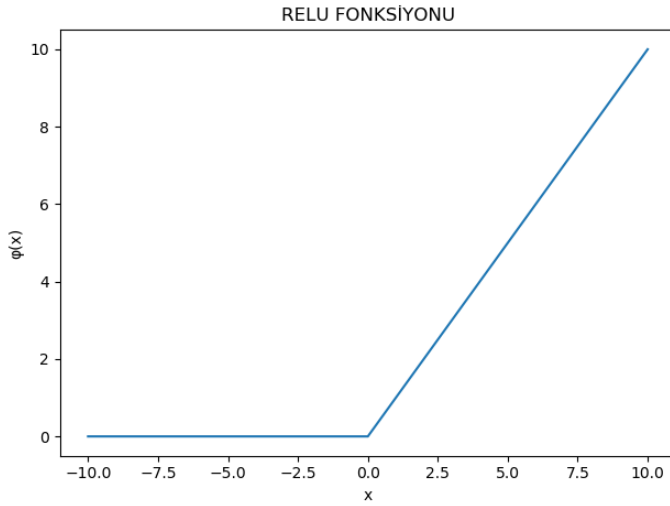
ağlarında varsayılan öneri relu aktivasyon fonksiyonunun kullanılmasıdır(Goodfellow ve ark. 2016).Bu tezde de gizli katmandaki nöronlar için relu kullanılmıştır ve tezde belirlenen problem ikili sınıflandırma problemi olduğundan sigmoid fonksiyonu da çıktı nöronunda kullanılmıştır.

3.1.3. Relu(RectifiedLinearUnit)

$$\varphi: \mathbb{R} \rightarrow [0, \infty)$$

$$x \rightarrow \varphi(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Fonksiyonuna relu(rectified linear unit) aktivasyon fonksiyonu denir. Grafiği Şekil 3.3'deki gibidir.



Şekil 3.3.Relu Fonksiyonu

Relu Fonksiyonunun Türevi:

$$\frac{d\varphi(x)}{dx} = \varphi'(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Relu fonksiyonu gizli katmanda kullanacağından r tane nöron için,

$$a_j^{[1]} = \varphi \left(w_{j,1}^{[1]}x_1 + w_{j,2}^{[1]}x_2 + \dots + w_{j,n}^{[1]}x_n + b_{j,1}^{[1]} \right), \quad 1 \leq j \leq r$$

şeklinde de gösterilir. Üst indis katman sayısını ve alt indis hangi nöron olduğunu göstermektedir.

3.1.4. Bir Gizli Katmanlı Yapay Sinir Ağları İçin Hipotez Tespiti

İkili sınıflandırma için hipotez 2.1.1.'de verilmişti. Burada da problem ikili sınıflandırma olduğundan hipotez aynıdır. Girdiler ile çıktılar arasında bir bağlantı kurmaktır. Hipotezin sağlanması için burada da sigmoid fonksiyonu kullanılacaktır. Bir gizli katmanlı yapay sinir ağlarında hipotez $\sigma(x) = a_1^{[2]}$ ile gösterilmiştir. Üst indis hangi katmanda olduğunu, alt indis ise o katmana ait nöron sayısını vermektedir. İkili sınıflandırma için çıktı katmanı bir nöronlu olduğundan alt indis "1" alınmıştır. O halde,

$$\sigma: \mathbb{R} \rightarrow (0,1)$$

$$x \rightarrow \sigma(x) = p(y = 1 | x; w)$$

olur.

Bir gizli katmanlı yapay sinir ağı için n boyutlu girdiye sahip olan ikili sınıflandırma probleminin hipotezi, $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)})$ ve r nöron sayısı olmak üzere,

$$a_1^{[2](i)} = \sigma \left(\sum_{j=1}^r w_{1,j}^{[2]} \varphi \left(w_{j,1}^{[1]} x_1^{(i)} + w_{j,2}^{[1]} x_2^{(i)} + \dots + w_{j,n}^{[1]} x_n^{(i)} + b_{j,1}^{[1]} \right) + b_{1,1}^{[2]} \right), \quad 1 \leq i \leq m,$$

şeklinde tanımlanır ve m tane veri için ayrı ayrı hesaplanır.

Hipotez için vektörleştirilmiş girdi ve parametreler kullanılacak olursa hipotez fonksiyonu

$$A^{[2]} = \left[a_1^{[2](1)} \ a_1^{2} \ \dots \ a_1^{[2](m)} \right] = \sigma(W^{[2]} \varphi(W^{[1]}X + b^{[1]}) + b^{[2]})$$

olarak tanımlanır. $A^{[2]} \in \mathbb{R}^{1 \times m}$ olur. $A^{[2]}$ ile elde edilen m tane değer veri kümesindeki m tane girdinin çıktısı için bir tahmindir. Bu tahminleri gerçek çıktılar ile karşılaştırıp yapılan hatayı hesaplamak için maliyet fonksiyonu kullanılır. Eğer hesaplanan hata yeterince küçük değil ise bu hata düşürülmeye çalışılır.

3.1.5. Maliyet Fonksiyonu

Bir gizli katmanlı yapay sinir ağı için maliyet fonksiyonu, m tane veri içeren, n boyutlu her hangi bir girdi, $x = (x_1, x_2, \dots, x_n)$, olmak üzere ve $1 \leq j \leq r$ için iki katmanlı

yapay sinir ağının parametreleri $w_r^{[1]} = (w_{r,1}^{[1]}, w_{r,2}^{[1]}, \dots, w_{r,n}^{[1]})$, $b^{[1]} =$

$(b_{1,1}^{[1]}, b_{2,1}^{[1]}, \dots, b_{r,1}^{[1]})$, $w_1^{[2]} = (w_{1,1}^{[2]}, w_{1,2}^{[2]}, \dots, w_{1,r}^{[2]})$, $b^{[2]} = (b_{1,1}^{[2]})$ olur. O halde, $1 \leq i \leq m$,

$$\hat{y}^{(i)} = a_1^{[2](i)} = \sigma \left(\sum_{j=1}^r w_{1,j}^{[2]} \varphi \left(w_{j,1}^{[1]} x_1^{(i)} + w_{j,2}^{[1]} x_2^{(i)} + \dots + w_{j,n}^{[1]} x_n^{(i)} + b_{j,1}^{[1]} \right) + b_{1,1}^{[2]} \right),$$

hipotezi oluşturulur ve

$$L(y^{(i)}, \hat{y}^{(i)}) = -[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})],$$

kayıp fonksiyonu olmak üzere maliyet fonksiyonu

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m L(y^{(i)}, \hat{y}^{(i)}) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})],$$

olur.

Maliyet fonksiyonu ile yapılan hata değerini düşürmek için ileri yayılım ve geri yayılım algoritmalarından yararlanır.

3.1.6. İleri Yayılım

Bir eğitim kümesi verilsin. Bu eğitim kümesindeki girdilerin belirlenen parametreler ile yapay sinir ağında soldan sağa doğru ilerleyen işlemlere denir. Yani girdinin yapay sinir ağı modeline girmesinden tahminin yapılmasına kadar olan süreçtir.

m veri sayısı, n girdi boyutu ve r nöron sayısı olmak üzere ileri yayılım sırasıyla,

$$z_j^{[1(i)]} = w_{j,1}^{[1]} x_1^{(i)} + w_{j,2}^{[1]} x_2^{(i)} + \dots + w_{j,n}^{[1]} x_n^{(i)} + b_{j,1}^{[1]}, \quad 1 \leq j \leq r,$$

$$a_j^{[1(i)]} = \varphi(z_j^{[1(i)]}), \quad 1 \leq j \leq r$$

$$z_1^{[2(i)]} = w_{1,1}^{[2]} a_1^{[1]} + w_{1,2}^{[2]} a_2^{[1]} + \dots + w_{1,r}^{[2]} a_r^{[1]} + b_{1,1}^{[2]}$$

$$a_1^{[2(i)]} = \sigma(z_1^{[2(i)]}),$$

adımları her bir örneğe uygulanarak belirlenen parametrelere göre tahmin yapılmış olur.

İleri yayılımın vektörleştirilmiş hali ise $X \in \mathbb{R}^{n \times m}$, $W^{[1]} \in \mathbb{R}^{r \times n}$, $b^{[1]} \in \mathbb{R}^{r \times 1}$, $W^{[2]} \in \mathbb{R}^{1 \times r}$ ve $b^{[2]} \in \mathbb{R}$ olmak üzere

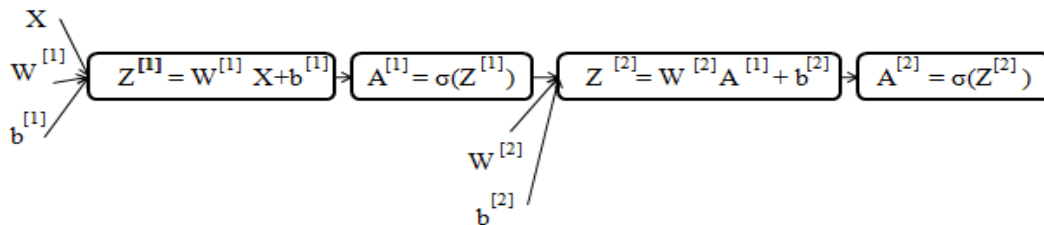
$$Z^{[1]} = W^{[1]}X + b^{[1]},$$

$$A^{[1]} = \varphi(Z^{[1]}),$$

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]},$$

$$A^{[2]} = \sigma(Z^{[2]}),$$

Şekil 3.4'den yararlanılarak tanımlanır. Burada $Z^{[1]} \in \mathbb{R}^{r \times m}$, $A^{[1]} \in \mathbb{R}^{r \times m}$, $Z^{[2]} \in \mathbb{R}^{1 \times m}$, $A^{[2]} \in \mathbb{R}^{1 \times m}$ olur.



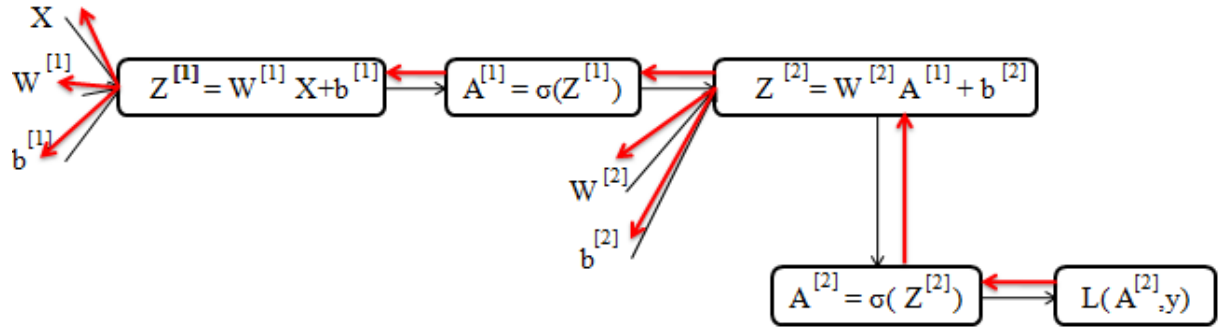
Şekil 3.4. İleri Yayılım Hesap Diyagramı

İleri yayılımda yapılan tahminlerin hata değeri maliyet fonksiyonu yardımı ile bulunur. Eğer maliyet fonksiyonunun değeri yeterince küçük değil ise maliyet fonksiyonunun minimize edilmesi için geri yayılım algoritması kullanılarak maliyet fonksiyonunun tüm parametrelere göre türevlerinin bulunması sağlanır.

3.1.7. Geri Yayılım

Geri yayılım, zincir kuralı kullanılarak maliyet fonksiyonunun tüm parametrelere göre türevlerini veren algoritmadır.

Maliyet fonksiyonunun parametrelere göre türevinin alınması için önce kayıp fonksiyonunun parametrelere göre türevi bulunur. Bunun için geriye yayılım hesap diyagramı kullanılır. Şekil 3.5’de geri yayılım hesap diyagramı verilmiştir.



Şekil 3.5. Geri Yayılım Hesap Diyagramı

Şekil 3.5’de kırmızı ile gösterilen oklar geri yayılım algoritmasını gösterir. Bu diyagramdan yararlanılarak, sağdan sola doğru ok yönünde zincir kuralı ile kayıp fonksiyonunun parametrelere göre türevleri bulunur. O halde kayıp fonksiyonunun parametrelere göre türevleri,

$$\frac{\partial L(y, a_1^{[2]})}{\partial w_{1,j}^{[2]}} = \left(\frac{\partial L(y, a_1^{[2]})}{\partial a_1^{[2]}} \right) \left(\frac{da_1^{[2]}}{dz_1^{[2]}} \right) \left(\frac{dz_1^{[2]}}{dw_{1,j}^{[2]}} \right), \quad (8)$$

$$\frac{\partial L(y, a_1^{[2]})}{\partial b_{1,1}^{[2]}} = \left(\frac{\partial L(y, a_1^{[2]})}{\partial a_1^{[2]}} \right) \left(\frac{da_1^{[2]}}{dz_1^{[2]}} \right) \left(\frac{dz_1^{[2]}}{db_{1,1}^{[2]}} \right), \quad (9)$$

$$\frac{\partial L(y, a_1^{[2]})}{\partial w_{j,n}^{[1]}} = \left(\frac{\partial L(y, a_1^{[2]})}{\partial a_1^{[2]}} \right) \left(\frac{da_1^{[2]}}{dz_1^{[2]}} \right) \left(\frac{dz_1^{[2]}}{da_j^{[1]}} \right) \left(\frac{da_j^{[1]}}{dz_j^{[1]}} \right) \left(\frac{dz_j^{[1]}}{dw_{j,n}^{[1]}} \right), \quad (10)$$

$$\frac{\partial L(y, a_1^{[2]})}{\partial b_{j,1}^{[1]}} = \left(\frac{\partial L(y, a_1^{[2]})}{\partial a_1^{[2]}} \right) \left(\frac{da_1^{[2]}}{dz_1^{[2]}} \right) \left(\frac{dz_1^{[2]}}{da_j^{[1]}} \right) \left(\frac{da_j^{[1]}}{dz_j^{[1]}} \right) \left(\frac{dz_j^{[1]}}{db_{j,1}^{[1]}} \right), \quad (11)$$

Olarak elde edilir. (8) ve (9) de kayıp fonksiyonunun ikinci katmanının parametrelerine göre türevleri, (10) ve (11) ise kayıp fonksiyonunun birinci katmanının parametrelerine göre türevleri elde edilmiştir. Bu türevlerin değerleri ayrı ayrı m tane veri için toplanır ve $\frac{1}{m}$ ile çarpıldığında maliyet fonksiyonunun parametrelere göre türevleri bulunur. Eşitliğin sağ tarafındaki türevler ise

$$\begin{aligned} \frac{\partial L(y, a_1^{[2]})}{\partial a_1^{[2]}} &= \frac{\partial \left[-\left(y \log a_1^{[2]} + (1-y) \log (1 - a_1^{[2]}) \right) \right]}{\partial a_1^{[2]}} = - \left[\frac{y}{a_1^{[2]}} - \frac{(1-y)}{(1 - a_1^{[2]})} \right] \\ &= -\frac{y}{a_1^{[2]}} + \frac{1-y}{1 - a_1^{[2]}} \end{aligned}$$

$$\begin{aligned} \frac{da_1^{[2]}}{dz_1^{[2]}} &= \frac{d\sigma(z_1^{[2]})}{dz_1^{[2]}} = -\left(1 + e^{-z_1^{[2]}}\right)^{-2} \cdot \left(-e^{-z_1^{[2]}}\right) = \frac{e^{-z_1^{[2]}}}{\left(1 + e^{-z_1^{[2]}}\right)^2} \\ &= \left(\frac{1}{1 + e^{-z_1^{[2]}}} \right) \cdot \left(\frac{e^{-z_1^{[2]}}}{1 + e^{-z_1^{[2]}}} \right) = \left(a_1^{[2]} \right) \cdot \left(1 - \frac{1}{1 + e^{-z_1^{[2]}}} \right) \\ &= \left(a_1^{[2]} \right) \cdot \left(1 - a_1^{[2]} \right) \end{aligned}$$

$$\frac{dz_1^{[2]}}{dw_{1,j}^{[2]}} = \frac{d \left[w_{1,1}^{[2]} a_1^{[1]} + w_{1,2}^{[2]} a_2^{[1]} + \dots + w_{1,r}^{[2]} a_r^{[1]} + b_{1,1}^{[2]} \right]}{dw_{1,j}^{[2]}} = a_j^{[1]}, \quad 1 \leq j \leq r,$$

$$\frac{dz_1^{[2]}}{db_{1,1}^{[2]}} = \frac{d \left[w_{1,1}^{[2]} a_1^{[1]} + w_{1,2}^{[2]} a_2^{[1]} + \dots + w_{1,r}^{[2]} a_r^{[1]} + b_{1,1}^{[2]} \right]}{db_{1,1}^{[2]}} = 1$$

$$\frac{dz_1^{[2]}}{da_j^{[1]}} = \frac{d \left[w_{1,1}^{[2]} a_1^{[1]} + w_{1,2}^{[2]} a_2^{[1]} + \dots + w_{1,r}^{[2]} a_r^{[1]} + b_{1,1}^{[2]} \right]}{da_j^{[1]}} = w_{1,j}^{[2]}, \quad 1 \leq j \leq r,$$

$$\frac{da_j^{[1]}}{dz_j^{[1]}} = \frac{d\varphi(z_j^{[1]})}{dz_j^{[1]}} = \begin{cases} 1 & , z_j^{[1]} \geq 0 \\ 0 & , z_j^{[1]} < 0 \end{cases}, \quad 1 \leq j \leq r,$$

$$\frac{dz_j^{[1]}}{dw_{j,n}^{[1]}} = \frac{d \left[w_{j,1}^{[1]} x_1 + w_{j,2}^{[1]} x_2 + \dots + w_{j,n}^{[1]} x_n + b_{j,1}^{[1]} \right]}{dw_{j,n}^{[1]}} = x_n, \quad 1 \leq j \leq r,$$

$$\frac{dz_j^{[1]}}{db_{j,1}^{[1]}} = \frac{d \left[w_{j,1}^{[1]} x_1 + w_{j,2}^{[1]} x_2 + \dots + w_{j,n}^{[1]} x_n + b_{j,1}^{[1]} \right]}{db_{j,1}^{[1]}} = 1, \quad 1 \leq j \leq r,$$

olur. O halde bu sonuçlar (8),(9),(10) ve (11)'de yerine koyularak, m tane veri için ayrı ayrı hesaplanıp, $\frac{1}{m}$ çarpılırsa maliyet fonksiyonunun parametrelere göre türevleri bulunmuş olur.

Buradan yola çıkarak

$$dw_{1,j}^{[2]} = \frac{\partial J(w, b)}{\partial w_{1,j}^{[2]}} = \frac{1}{m} \sum_{i=1}^m \left(\frac{\partial L(y^{(i)}, a_1^{[2](i)})}{\partial w_{1,j}^{[2]}} \right),$$

$$db_{1,1}^{[2]} = \frac{\partial J(w, b)}{\partial b_{1,1}^{[2]}} = \frac{1}{m} \sum_{i=1}^m \left(\frac{\partial L(y^{(i)}, a_1^{[2](i)})}{\partial b_{1,1}^{[2]}} \right),$$

$$dw_{j,n}^{[1]} = \frac{\partial J(w, b)}{\partial w_{j,n}^{[1]}} = \frac{1}{m} \sum_{i=1}^m \left(\frac{\partial L(y^{(i)}, a_1^{[2](i)})}{\partial w_{j,n}^{[1]}} \right),$$

$$db_{j,1}^{[1]} = \frac{\partial J(w, b)}{\partial b_{j,1}^{[1]}} = \frac{1}{m} \sum_{i=1}^m \left(\frac{\partial L(y^{(i)}, a_1^{[2](i)})}{\partial b_{j,1}^{[1]}} \right),$$

ifadeleri maliyet fonksiyonunun parametrelere göre türevlerini göstermek üzere

$$\begin{aligned} dz_1^{[2]} &= \left(\frac{dL(y, a_1^{[2]})}{da_1^{[2]}} \right) \left(\frac{da_1^{[2]}}{dz_1^{[2]}} \right) = \left(-\frac{y}{a_1^{[2]}} + \frac{1-y}{1-a_1^{[2]}} \right) \cdot (a_1^{[2]}) \cdot (1-a_1^{[2]}) \\ &= -y(1-a_1^{[2]}) + (1-y)(a_1^{[2]}) = -y + ya_1^{[2]} + a_1^{[2]} - ya_1^{[2]} \\ &= a_1^{[2]} - y, \end{aligned}$$

$$dw_{1,j}^{[2]} = \frac{1}{m} (dz_1^{[2]}) \left(\frac{dz_1^{[2]}}{dw_{1,j}^{[2]}} \right) = \left(\frac{1}{m} \right) (a_1^{[2]} - y) (a_j^{[1]}), \quad (12)$$

$$db_{1,1}^{[2]} = \frac{1}{m} (dz_1^{[2]}) \left(\frac{dz_1^{[2]}}{db_{1,1}^{[2]}} \right) = \frac{1}{m} (a_1^{[2]} - y) (1) = \frac{1}{m} (a_1^{[2]} - y), \quad (13)$$

$$dz_j^{[1]} = (dz_1^{[2]}) \left(\frac{dz_1^{[2]}}{da_j^{[1]}} \right) \left(\frac{da_j^{[1]}}{dz_j^{[1]}} \right) = (a_1^{[2]} - y) (w_{1,j}^{[2]}) (\varphi'(z_j^{[1]})),$$

$$dw_{j,n}^{[1]} = \frac{1}{m} (dz_j^{[1]}) \left(\frac{dz_j^{[1]}}{dw_{j,n}^{[1]}} \right) = \frac{1}{m} (a_1^{[2]} - y) (w_{1,j}^{[2]}) (\varphi'(z_j^{[1]})) (x_n), \quad (14)$$

$$\begin{aligned} db_{j,1}^{[1]} &= \frac{1}{m} (dz_j^{[1]}) \left(\frac{dz_j^{[1]}}{db_{j,1}^{[1]}} \right) = \frac{1}{m} (a_1^{[2]} - y) (w_{1,j}^{[2]}) (\varphi'(z_j^{[1]})) (1) \\ &= \frac{1}{m} (a_1^{[2]} - y) (w_{1,j}^{[2]}) (\varphi'(z_j^{[1]})), \end{aligned} \quad (15)$$

bulunur.

Sonuç olarak (12), (13), (14) ve (15)'de maliyet fonksiyonunun tüm parametrelere türevleri bulunmuş olur.

Geri yayılım, vektörleştirilmiş eğitim kümesi ve vektörleştirilmiş parametreler kullanılarak hesaplanırsa;

$X \in \mathbb{R}^{n \times m}$ girdi matrisi, $Y \in \mathbb{R}^{1 \times m}$ çıktı vektörü, $W^{[1]} \in \mathbb{R}^{r \times n}$, $b^{[1]} \in \mathbb{R}^{r \times 1}$, $W^{[2]} \in \mathbb{R}^{1 \times r}$ ve $b^{[2]} \in \mathbb{R}$ başlangıç parametreleri ve $Z^{[1]} \in \mathbb{R}^{r \times m}$, $A^{[1]} \in \mathbb{R}^{r \times m}$, $Z^{[2]} \in \mathbb{R}^{1 \times m}$, $A^{[2]} \in \mathbb{R}^{1 \times m}$ olmak üzere Şekil 3.5'deki geri yayılım hesap diyagramı göz önüne alınarak

$$dZ^{[2]} = (A^{[2]} - Y) \in \mathbb{R}^{1 \times m},$$

$$dW^{[2]} = \frac{\partial J(W, b)}{\partial W^{[2]}} = \frac{1}{m} \cdot (dZ^{[2]})(A^{[1]T}) \in \mathbb{R}^{1 \times r}, \quad (16)$$

$$db^{[2]} = \frac{1}{m} \cdot \left(\sum_{i=1}^m dZ_{1,i}^{[2]} \right) \in \mathbb{R}, \quad (17)$$

$$dZ^{[1]} = \frac{1}{m} \cdot (W^{[2]T} dZ^{[2]}). \varphi'(Z^{[1]}) \in \mathbb{R}^{r \times m},$$

$$dW^{[1]} = \frac{1}{m} \cdot (dZ^{[1]})(X^T) \in \mathbb{R}^{r \times n}, \quad (18)$$

$$db_j^{[1]} = \left(\sum_{i=1}^m dZ_{j,i}^{[2]} \right), \quad 1 \leq j \leq r,$$

$$db^{[1]} = \frac{1}{m} \cdot \begin{pmatrix} db_1^{[1]} \\ db_2^{[1]} \\ \vdots \\ db_r^{[1]} \end{pmatrix} \in \mathbb{R}^{r \times 1}, \quad (19)$$

elde edilir.

Sonuç olarak vektörleştirilmiş eğitim kümesi ve parametreler kullanılarak tüm eğitim kümesi için geri yayılım (16), (17), (18) ve (19) matrisleriyle elde edilmiş olur.

Geri yayılım algoritması ile bulunan türevler meyilli azalım algoritmasında kullanılır. Meyilli azalım algoritması ise maliyet fonksiyonunu minimize etmek için kullanılan bir algoritmadır.

3.1.8. Meyilli Azalım

Meyilli azalım algoritması 2.1.5'te lojistik regresyon için tanımlanmıştı. Bu algoritma bir gizli katmanlı yapay sinir ağları içinde hemen hemen aynıdır. Öyle ki geri yayılımda bulduğumuz $dW^{[2]}$, $db^{[2]}$, $dW^{[1]}$, $db^{[1]}$ matrisleri kullanılarak bütün parametreler güncellenecektir. $\alpha < 1$ adım büyüklüğü olmak üzere güncellenen parametreler

$$W^{[2]} := W^{[2]} - \alpha dW^{[2]},$$

$$b^{[2]} := b^{[2]} - \alpha db^{[2]},$$

$$W^{[1]} := W^{[1]} - \alpha dW^{[1]},$$

$$b^{[1]} := b^{[1]} - \alpha db^{[1]},$$

olur. Sol tarafta elde edilen güncellenen parametrelerdir. Güncellenen parametreler kullanılarak maliyet fonksiyonunu minimize edilir.

Buraya kadar yapılan işlemler bölüm 3.1.9'da adım adım anlatılmıştır.

3.1.9. Bir gizli Katmanlı Yapay Sinir Ağlarında Öğrenme

Bir veri kümesi verilsin. Bu veri kümesi için başlangıç parametreleri atanır. Daha sonra girdiler ve birinci karmana ait başlangıç parametreleri birlikte gizli katmana gelir. Gizli katmandaki nöronlarda gerekli işlemler yapıldıktan sonra buradaki nöronlar ve ikinci katmana ait başlangıç parametreleri çıktı katmanına iletilir. Çıktı katmanında verilen girdilerin tahminleri yapılır. Yapılan tahminler ve verilerin gerçek çıktıları, maliyet fonksiyonu kullanılarak yapılan hata hesaplanır. Hata değeri yeterince küçük değil ise veri kümesinin öğrenilmesi için aşağıdaki adımlar takip edilir;

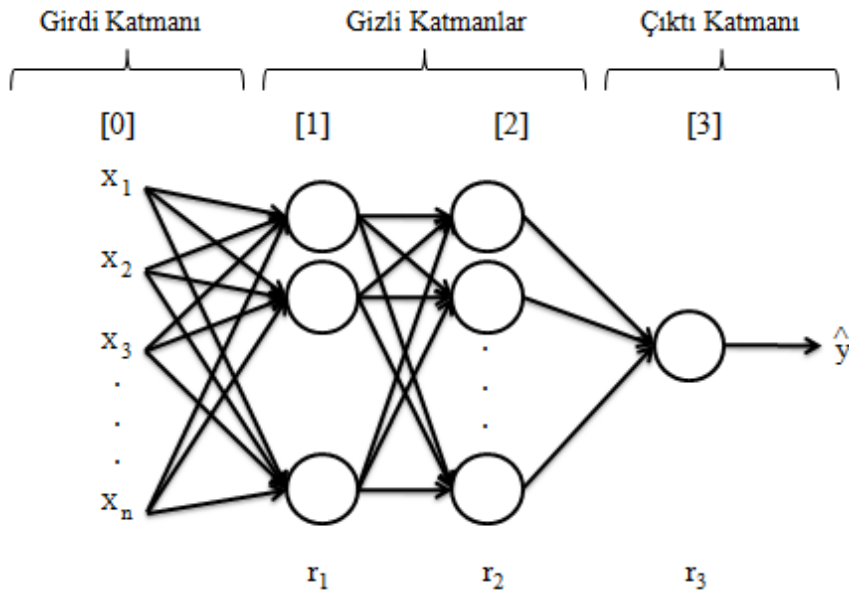
- Geri yayılım,
- Meyilli azalım ile parametrelerin güncellenmesi,
- İleri yayılım,
- Maliyet fonksiyonu.

Yukarıdaki adımlar sırasıyla uygulanır. Maliyet fonksiyonu yeterince küçük oluncaya kadar adımlar başa dönülerek tekrar tekrar uygulanır. Öğrenme hata değeri yeterince küçük oluncaya kadar devam eder. Öğrenme tamamlandığında elde edilen parametreler istenilen parametrelerdir. Böylece bu parametreler ile algoritma veri kümesini öğrenmiş olur.

3.2. İki Gizli Katmanlı Yapay Sinir Ağları

$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}) \in \mathbb{R}^n$ ve $y^{(i)} \in \{0,1\}$, $i = 1, 2, \dots, m$, olmak üzere $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, veri kümesi verilsin.

$L = 3$, r_1 birinci katmana ait nöron sayısı, ikinci katmana ait nöron sayısı r_2 ve $r_3 = 1$ olmak üzere Şekil 3.6'daki gizli katmanlı ikili sınıflandırma problemi için yapay sinir ağının en genel gösterimi verilmiştir.



Şekil 3.6.İki Gizli Katmanlı Yapay Sinir Ağı Mimarisi

3.2.1. İki Gizli Katmanlı Yapay Sinir Ağları İçin Hipotez Tespiti

Vektörleştirilmiş veri ve parametreler kullanılarak iki gizli katmanlı ikili sınıflandırma problemi için yapay sinir ağı hipotezi

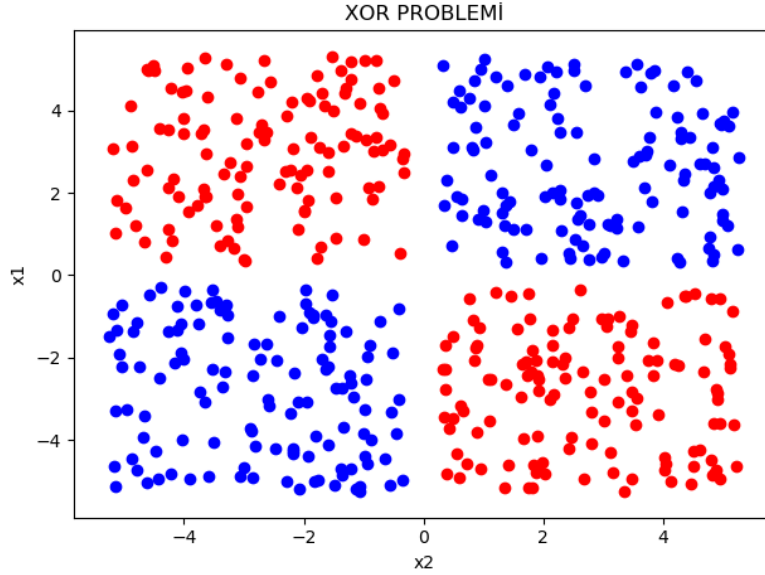
$$\hat{y} = a_1^{[3]} = \sigma(W^{[3]}\varphi(W^{[2]}\varphi(W^{[1]}X + b^{[1]}) + b^{[2]}) + b^{[3]})$$

olur.

Bir gizli katmanlı yapay sinir ağları içi kullandığımız kayıp fonksiyonu, maliyet fonksiyonu, ileri yayılım, geri yayılım ve meyilli azalım algoritmaları aynı şekilde iki gizli katmanlı yapay sinir ağları içinde kullanılmaktadır.

4. XOR PROBLEMİNİN BİR GİZLİ KATMANLI YAPAY SİNİR AĞLARI İLE SINIFLANDIRILMASI

XOR problemi için rastgele seçilmiş bir veri alınır (Smilkov ve Carter 2018). Veri kümesi 500 veri içermektedir. Alınan eğitim kümesinin görseli Şekil 4.1’de verilmiştir.

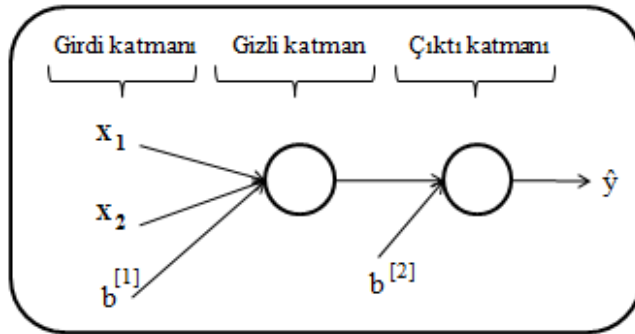


Şekil 4.1. XOR Eğitim Kümesi Görseli

Burada yapay sinir ağı başlangıç parametreleri, adım büyüklüğü ($\alpha = 0.1$) ve epoch sayısı (1000) sabit tutularak XOR probleminin bir gizli katmanlı yapay sinir ağı mimarisi için ideal nöron sayısının bulunması amaçlanmıştır. İstenilen hata değeri olarak $\varepsilon = 0.016$ atanmıştır.

4.1. Gizli Katmanındaki Nöron Sayısı Bir Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 1$ olmak üzere yapay sinir ağı mimarisi Şekil 4.2’de verilmiştir.



Şekil 4.2. Bir Gizli Katmanlı ve Bir gizli Nöronlu Mimari

Bu mimari için başlangıç parametreleri

$$W^{[1]} = [w_{1,1}^{[1]} \quad w_{1,2}^{[1]}] = [-0.00416758 \quad -0.00056267],$$

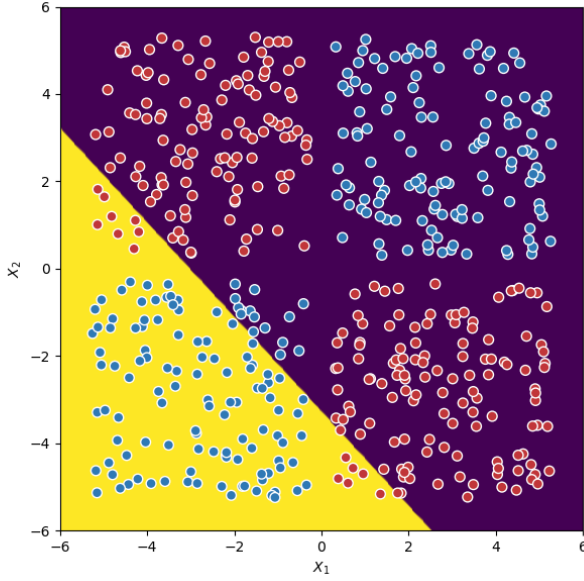
$$b^{[1]} = [b_{1,1}^{[1]}] = [0],$$

$$W^{[2]} = [w_{1,1}^{[2]}] = [0.01788628],$$

$$b^{[2]} = [b_{1,1}^{[2]}] = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari aşağıdaki gibi sınıflandırmıştır.



Şekil 4.3. $r=1$ için XOR'un Sınıflandırılması

$r = 1$ için meyilli azalım kullanılarak bulunan parametreler

$$W^{[1]} = [-0.6466238 \quad -0.59732197],$$

$$b^{[1]} = [-1.55920259],$$

$$W^{[2]} = [1.79013557],$$

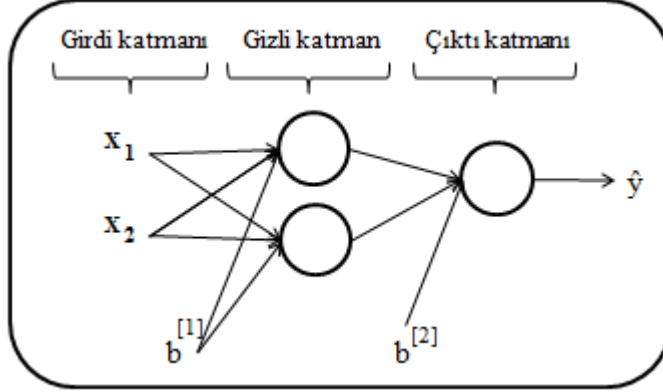
$$b^{[2]} = [-0.52373678],$$

olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.558232$ dur. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %68'tir.

$r = 1$ olduğunda görülmektedir ki XOR probleminin çözümü için gizli katmanda bir nöron yeterli değildir. $J(w, b)$ değeride oldukça büyüktür.

4.2. Gizli Katmanındaki Nöron Sayısı İki Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 2$ olmak üzere yapay sinir ağı mimarisi Şekil 4.4'te verilmiştir.



Şekil 4.4. Bir Gizli Katmanlı ve İki gizli Nöronlu Mimari

Bu mimari için birinci nöronun başlangıç parametreleri sabit tutularak başlangıç parametreleri

$$W^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} \end{bmatrix} = \begin{bmatrix} -0.00416758 & -0.00056267 \\ -0.02136196 & 0.01640271 \end{bmatrix},$$

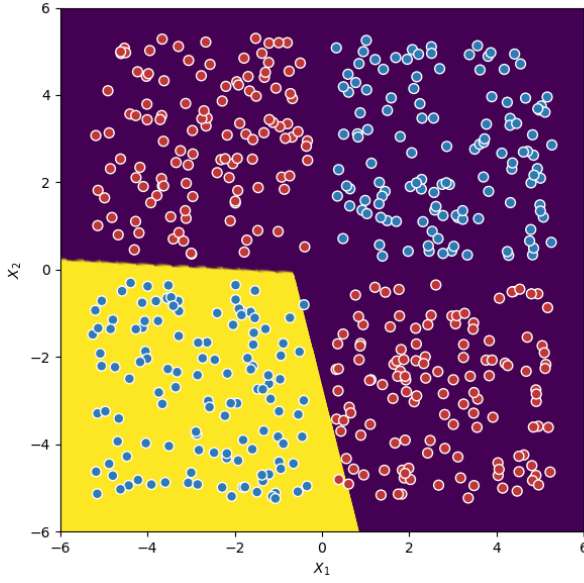
$$b^{[1]} = \begin{bmatrix} b_{1,1}^{[1]} \\ b_{2,1}^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$W^{[2]} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} \end{bmatrix} = [0.01788628 \quad 0.0043651],$$

$$b^{[2]} = \begin{bmatrix} b_{1,1}^{[2]} \end{bmatrix} = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari aşağıdaki gibi sınıflandırmıştır.



Şekil 4.5. $r=2$ için XOR'un Sınıflandırılması

$r = 2$ için meyilli azalım kullanılarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.82200408 & -0.46187805 \\ -1.79698039 & 0.49026884 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.97045222 \\ -0.93350411 \end{bmatrix},$$

$$W^{[2]} = [2.11519597 \quad -2.08308646],$$

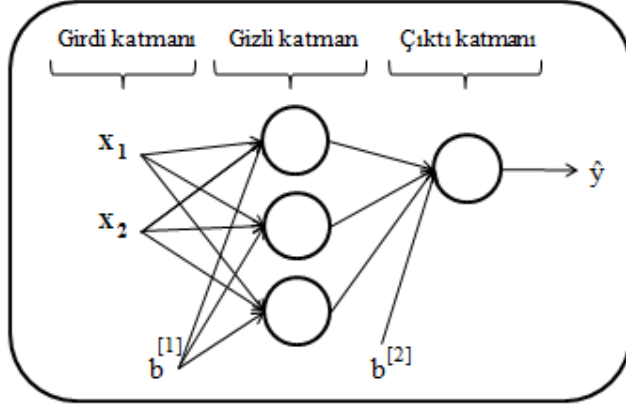
$$b^{[2]} = [-0.01868901],$$

olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.387435$ dur. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %74'tür.

$r = 2$ olduğunda görülmektedir ki XOR probleminin çözümü için gizli katmanda iki nöron da yeterli değildir. $J(w, b)$ değeri de halen oldukça büyüktür.

4.3. Gizli Katmanındaki Nöron Sayısı Üç Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 3$ olmak üzere yapay sinir ağı mimarisi Şekil 4.6'da verilmiştir.



Şekil 4.6. Bir Gizli Katmanlı ve Üç gizli Nöronlu Mimarisi

Bu mimari için 1. ve 2. nöronun başlangıç parametreleri sabit tutularak başlangıç parametreleri

$$W^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} \end{bmatrix} = \begin{bmatrix} -0.00416758 & -0.00056267 \\ -0.02136196 & 0.01640271 \\ -0.01793436 & -0.00841747 \end{bmatrix},$$

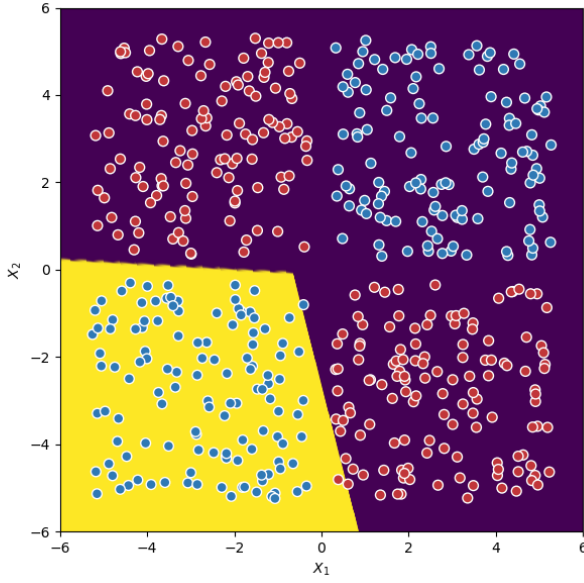
$$b^{[1]} = \begin{bmatrix} b_{1,1}^{[1]} \\ b_{2,1}^{[1]} \\ b_{3,1}^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$W^{[2]} = [w_{1,1}^{[2]} \ w_{1,2}^{[2]} \ w_{1,3}^{[2]}] = [0.01788628 \ 0.0043651 \ 0.00096497],$$

$$b^{[2]} = [b_{1,1}^{[2]}] = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari aşağıdaki gibi sınıflandırmıştır.



Şekil 4.7. $r=3$ için XOR'un Sınıflandırılması

$r = 3$ için meyilli azalım kullanılarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.37297974 & -0.34979449 \\ -1.79834739 & 0.48925823 \\ -1.19891595 & -0.30548031 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.73644521 \\ -0.93355689 \\ -0.64328836 \end{bmatrix},$$

$$W^{[2]} = [1.59670474 \quad -2.08405655 \quad 1.39418183],$$

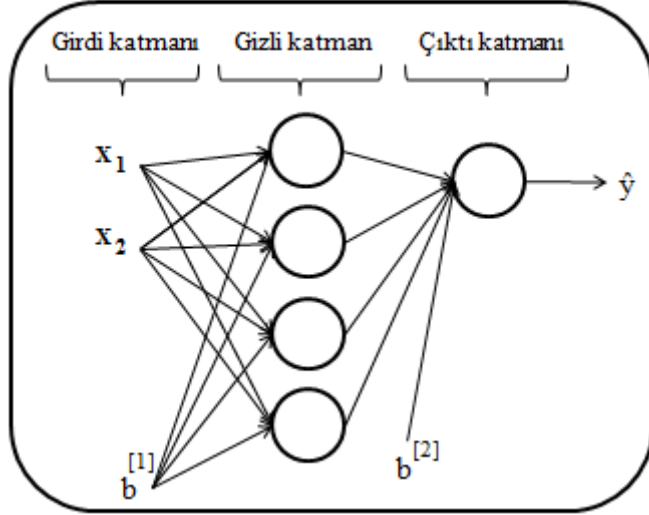
$$b^{[2]} = [-0.01842142],$$

olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.387346$ dır. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %74'tür.

$r = 3$ olduğunda görülmektedir ki XOR probleminin çözümü için gizli katmanda üç nöron da yeterli değildir. $J(w, b)$ değeri de halen oldukça büyüktür.

4.4. Gizli Katmanındaki Nöron Sayısı Dört Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 4$ olmak üzere yapay sinir ağı mimarisi Şekil 4.8'de verilmiştir.



Şekil 4.8. Bir Gizli Katmanlı ve Dört gizli Nöronlu Mimari

Bu mimari için 1., 2. ve 3. nöronun başlangıç parametreleri sabit tutularak başlangıç parametreleri

$$W^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} \end{bmatrix} = \begin{bmatrix} -0.00416758 & -0.00056267 \\ -0.02136196 & 0.01640271 \\ -0.01793436 & -0.00841747 \\ 0.00502881 & -0.01245288 \end{bmatrix},$$

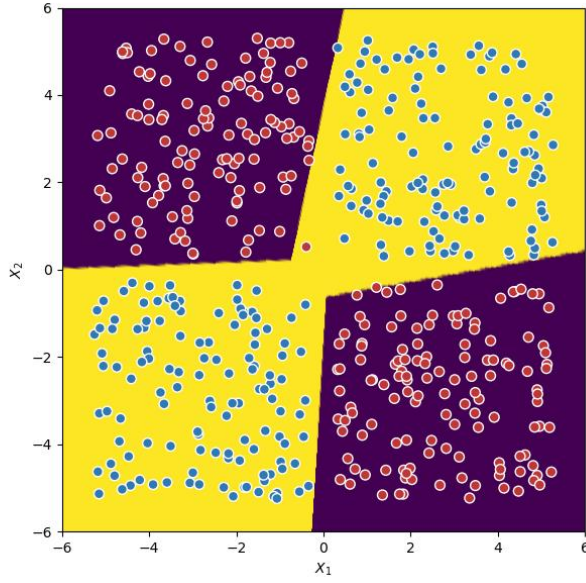
$$b^{[1]} = \begin{bmatrix} b_{1,1}^{[1]} \\ b_{2,1}^{[1]} \\ b_{3,1}^{[1]} \\ b_{4,1}^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$W^{[2]} = [w_{1,1}^{[2]} w_{1,2}^{[2]} w_{1,3}^{[2]} w_{1,4}^{[2]}] = [0.01788628 \quad 0.0043651 \quad 0.00096497 \quad -0.01863493],$$

$$b^{[2]} = [b_{1,1}^{[2]}] = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari aşağıdaki gibi sınıflandırmıştır.



Şekil 4.9. r=4 için XOR'un Sınıflandırılması

r = 4 için meyilli azalım kullanılarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.30730919 & -1.25253798 \\ -2.09783362 & 0.4427557 \\ -1.14131966 & -1.09352531 \\ 0.37280674 & -2.09555605 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.49404586 \\ -0.19895629 \\ -0.43151103 \\ -0.08775851 \end{bmatrix}$$

$$W^{[2]} = [1.87673505 \quad -2.1529249 \quad 1.6383433 \quad -2.13013238],$$

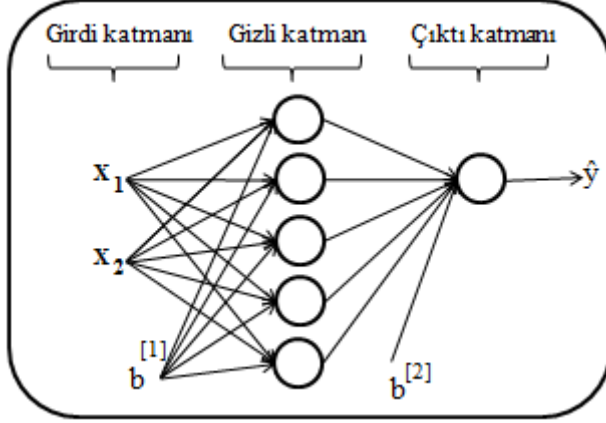
$$b^{[2]} = [2.61344466],$$

olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.045593$ dur. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %99'dur.

r = 4 olduğunda görülmektedir ki XOR probleminin çözümü için gizli katmanda dört nöron yeterli olmaktadır. $J(w, b)$ değerinde büyük bir azalma olmuştur. Ancak istenilen hata değerinden büyük olduğundan ideal olduğu konusunda bir şey söylenemez.

4.5. Gizli Katmanındaki Nöron Sayısı Beş Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 5$ olmak üzere yapay sinir ağı mimarisi Şekil 4.10'da verilmiştir.



Şekil 4.10. Bir Gizli Katmanlı ve Beş Gizli Nöronlu Mimari

Bu mimari için 1., 2., 3. ve 4. nöronların başlangıç parametreleri sabit tutularak başlangıç parametreleri

$$W^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} \\ w_{5,1}^{[1]} & w_{5,2}^{[1]} \end{bmatrix} = \begin{bmatrix} -0.00416758 & -0.00056267 \\ -0.02136196 & 0.01640271 \\ -0.01793436 & -0.00841747 \\ 0.00502881 & -0.01245288 \\ -0.01057952 & -0.00909008 \end{bmatrix},$$

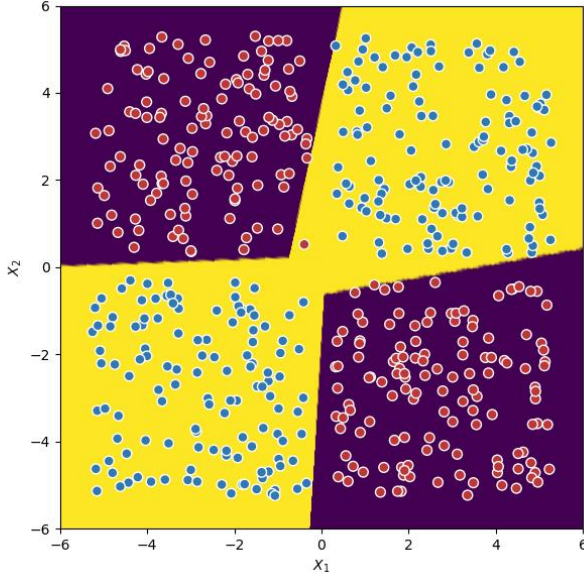
$$b^{[1]} = \begin{bmatrix} b_{1,1}^{[1]} \\ b_{2,1}^{[1]} \\ b_{3,1}^{[1]} \\ b_{4,1}^{[1]} \\ b_{5,1}^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$W^{[2]} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]} & w_{1,4}^{[2]} & w_{1,5}^{[2]} \end{bmatrix} \\ = [0.01788628 \quad 0.0043651 \quad 0.00096497 \quad -0.01863493 \quad -0.00277388],$$

$$b^{[2]} = \begin{bmatrix} b_{1,1}^{[2]} \end{bmatrix} = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari aşağıdaki gibi sınıflandırmıştır.



Şekil 4.11. $r=5$ için XOR'un Sınıflandırılması

$r = 5$ için meyilli azalım kullanılarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.21838385 & -1.16753869 \\ -2.09792947 & 0.44282065 \\ -1.06376824 & -1.0192641 \\ 0.37273312 & -2.09576196 \\ -0.62941133 & -0.60309232 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.46160799 \\ -0.1991115 \\ -0.40313222 \\ -0.08767399 \\ -0.23851318 \end{bmatrix},$$

$$W^{[2]} = [1.74952872 \quad -2.15304895 \quad 1.52728159 \quad -2.13032173 \quad 0.90364517],$$

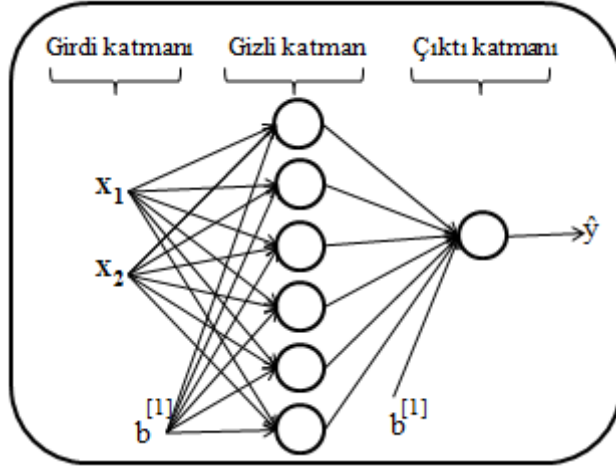
$$b^{[2]} = [2.61352887],$$

olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.045572$ dur ve $r = 4$ 'e göre biraz daha azalmıştır. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %99'tir.

$r = 5$ olduğunda görülmektedir ki XOR probleminin çözümü için gizli katmanda beş nöron da yeterli olmaktadır. Ancak hata değeri istenilen hata değerinden ($\epsilon = 0.016$) büyüktür.

4.6. Gizli Katmanındaki Nöron Sayısı Altı Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 6$ olmak üzere yapay sinir ağı mimarisi Şekil 4.12'de verilmiştir.



Şekil 4.12. Bir Gizli Katmanlı ve Altı Gizli Nöronlu Mimarisi

Bu mimari 1., 2., 3., 4. ve 5. nöronların başlangıç parametreleri sabit tutularak başlangıç parametreleri

$$W^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} \\ w_{5,1}^{[1]} & w_{5,2}^{[1]} \\ w_{6,1}^{[1]} & w_{6,2}^{[1]} \end{bmatrix} = \begin{bmatrix} -0.00416758 & -0.00056267 \\ -0.02136196 & 0.01640271 \\ -0.01793436 & -0.00841747 \\ 0.00502881 & -0.01245288 \\ -0.01057952 & -0.00909008 \\ 0.00551454 & 0.02292208 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} b_{1,1}^{[1]} \\ b_{2,1}^{[1]} \\ b_{3,1}^{[1]} \\ b_{4,1}^{[1]} \\ b_{5,1}^{[1]} \\ b_{6,1}^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

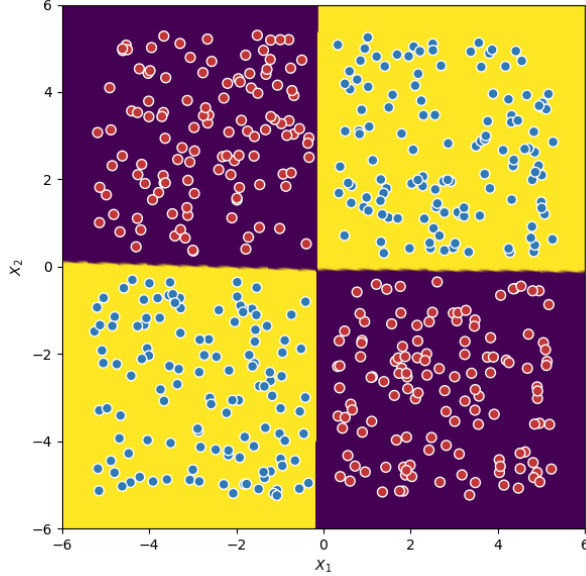
$$W^{[2]} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]} & w_{1,4}^{[2]} & w_{1,5}^{[2]} & w_{1,6}^{[2]} \\ w_{2,1}^{[2]} & w_{2,2}^{[2]} & w_{2,3}^{[2]} & w_{2,4}^{[2]} & w_{2,5}^{[2]} & w_{2,6}^{[2]} \end{bmatrix}$$

$$= \begin{bmatrix} 0.01788628 & 0.0043651 & 0.00096497 & -0.01863493 & -0.00277388 \\ -0.00354759 & & & & & \end{bmatrix},$$

$$b^{[2]} = [b_{1,1}^{[2]}] = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari aşağıdaki gibi sınıflandırmıştır.



Şekil 4.13. $r=6$ için XOR'un Sınıflandırılması

$r = 6$ için meyilli azalım kullanılarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.01372638 & -0.99861548 \\ -1.45087942 & 1.23610499 \\ -0.88501907 & -0.87184679 \\ 1.26884377 & -1.48866592 \\ -0.52365019 & -0.51587034 \\ 1.36103523 & 1.26147237 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09512789 \\ 0.03635731 \\ -0.0833593 \\ 0.01420144 \\ -0.04932769 \\ 0.05984022 \end{bmatrix},$$

$W^{[2]}$

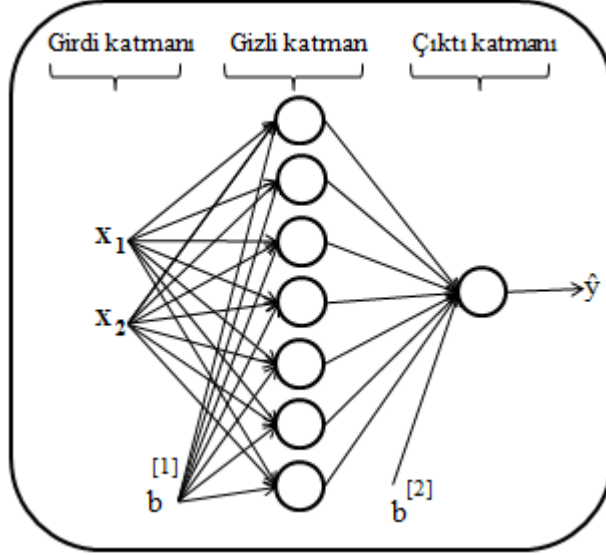
$$= [1.42624428 \quad -1.90620662 \quad 1.24497431 \quad -1.95609683 \quad 0.73661043 \quad 1.85655843],$$

$$b^{[2]} = [0.09367112],$$

olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.015309$ dur ve $r = 5$ 'e göre biraz daha azalmıştır. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %100'dür. $r = 6$ olduğunda XOR probleminin sınıflandırdığı görülmektedir. Üstelik hata değeri istenilen hata değerinden küçüktür. O halde atanan başlangıç parametreleri için $r = 6$ 'nın ideal nöron sayısı olduğu söylenebilir.

4.7. Gizli Katmanındaki Nöron Sayısı Yedi Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 7$ olmak üzere yapay sinir ağı mimarisi Şekil 4.14'te verilmiştir.



Şekil 4.14. Bir Gizli Katmanlı ve Yedi Gizli Nöronlu Mimari

Bu mimari için 1., 2., 3., 4., 5. ve 6. nöronların başlangıç parametreleri sabit tutularak başlangıç parametreleri

$$W^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} \\ w_{5,1}^{[1]} & w_{5,2}^{[1]} \\ w_{6,1}^{[1]} & w_{6,2}^{[1]} \\ w_{7,1}^{[1]} & w_{7,2}^{[1]} \end{bmatrix} = \begin{bmatrix} -0.00416758 & -0.00056267 \\ -0.02136196 & 0.01640271 \\ -0.01793436 & -0.00841747 \\ 0.00502881 & -0.01245288 \\ -0.01057952 & -0.00909008 \\ 0.00551454 & 0.02292208 \\ 0.00041539 & -0.01117925 \end{bmatrix},$$

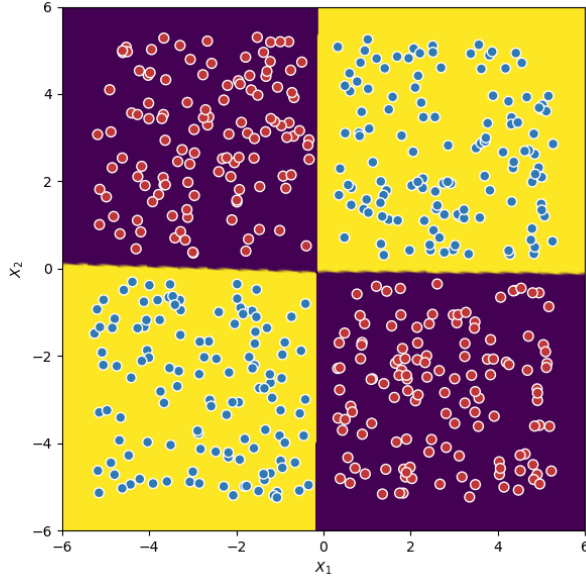
$$b^{[1]} = \begin{bmatrix} b_{1,1}^{[1]} \\ b_{2,1}^{[1]} \\ b_{3,1}^{[1]} \\ b_{4,1}^{[1]} \\ b_{5,1}^{[1]} \\ b_{6,1}^{[1]} \\ b_{7,1}^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\begin{aligned}
W^{[2]} &= [w_{1,1}^{[2]} \quad w_{1,2}^{[2]} \quad w_{1,3}^{[2]} w_{1,4}^{[2]} \quad w_{1,5}^{[2]} w_{1,6}^{[2]} \quad w_{1,7}^{[2]}] \\
&= [0.01788628 \quad 0.0043651 \quad 0.00096497 - 0.01863493 \quad -0.00277388 \\
&\quad - 0.00354759 \quad - 0.00082741],
\end{aligned}$$

$$b^{[2]} = [b_{1,1}^{[2]}] = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari aşağıdaki gibi sınıflandırmıştır.



Şekil 4.15. $r=7$ için XOR'un Sınıflandırılması

$r = 7$ için meylli azalım kullanılarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.01362538 & -0.99897917 \\ -1.45112772 & 1.23552996 \\ -0.88493887 & -0.87215601 \\ 1.24433294 & -1.46066948 \\ -0.52360923 & -0.51604714 \\ 1.36115628 & 1.26091703 \\ 0.24797248 & -0.29115411 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09533964 \\ 0.03697203 \\ -0.08348818 \\ 0.01324979 \\ -0.0494038 \\ 0.05952028 \\ 0.00290295 \end{bmatrix},$$

$$W^{[2]}$$

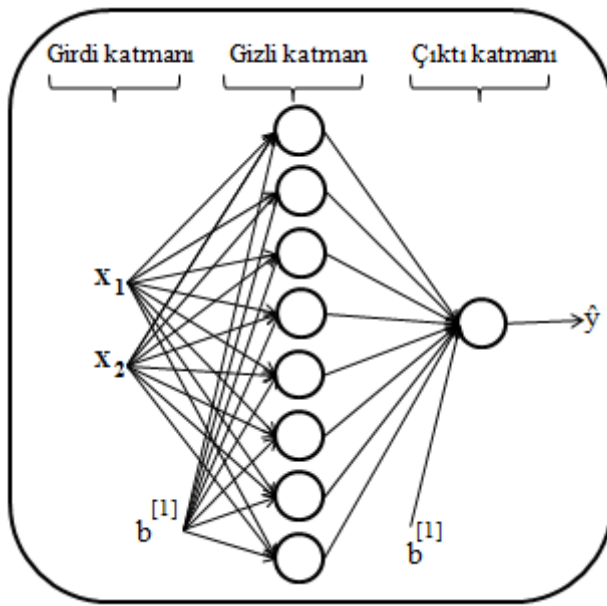
$$\begin{aligned}
&= [1.4264412 \quad -1.90603459 \quad 1.24514249 - 1.91888498 \quad 0.73671022 \quad 1.8562596 \\
&\quad - 0.38229393],
\end{aligned}$$

$$b^{[2]} = [0.09572404],$$

olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.015307$ dur ve $r = 6$ 'ya göre biraz daha azalmıştır. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %100'dür. $r = 7$ olduğunda XOR probleminin sınıflandırdığı görülmektedir. Üstelik hata değeri de istenilen hata değerinden küçüktür. Ancak $r = 7$ nöron işlem yükünü arttırdığından $r = 6$ ideal nöron olarak tercih edilir.

4.8. Gizli Katmanındaki Nöron Sayısı Sekiz Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 8$ olmak üzere yapay sinir ağı mimarisi Şekil 4.16'da verilmiştir.



Şekil 4.16. Bir Gizli Katmanlı ve Sekiz Gizli Nöronlu Mimari

Bu mimari için 1., 2., 3., 4., 5., 6. ve 7. nöronların başlangıç parametreleri sabit tutularak başlangıç parametreleri

$$W^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} \\ w_{5,1}^{[1]} & w_{5,2}^{[1]} \\ w_{6,1}^{[1]} & w_{6,2}^{[1]} \\ w_{7,1}^{[1]} & w_{7,2}^{[1]} \\ w_{8,1}^{[1]} & w_{8,2}^{[1]} \end{bmatrix} = \begin{bmatrix} -0.00416758 & -0.00056267 \\ -0.02136196 & 0.01640271 \\ -0.01793436 & -0.00841747 \\ 0.00502881 & -0.01245288 \\ -0.01057952 & -0.00909008 \\ 0.00551454 & 0.02292208 \\ 0.00041539 & -0.01117925 \\ 0.00539058 & -0.0059616 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} b_{1,1}^{[1]} \\ b_{2,1}^{[1]} \\ b_{3,1}^{[1]} \\ b_{4,1}^{[1]} \\ b_{5,1}^{[1]} \\ b_{6,1}^{[1]} \\ b_{7,1}^{[1]} \\ b_{8,1}^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

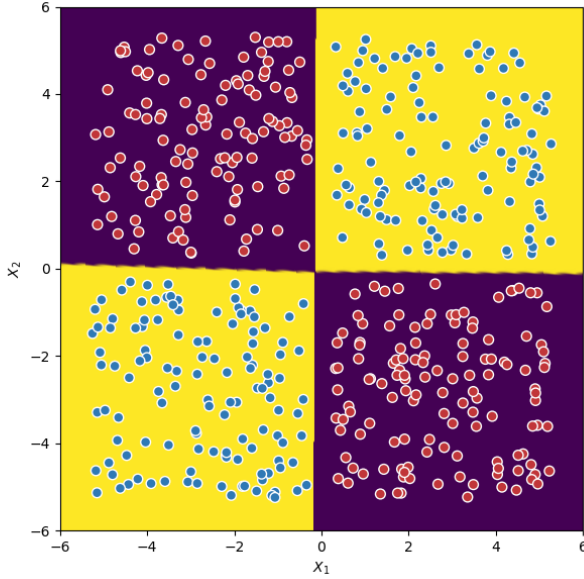
$$W^{[2]} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]} & w_{1,4}^{[2]} & w_{1,5}^{[2]} & w_{1,6}^{[2]} & w_{1,7}^{[2]} & w_{1,8}^{[2]} \end{bmatrix}$$

$$= \begin{bmatrix} 0.01788628 & 0.0043651 & 0.00096497 & -0.01863493 & -0.00277388 \\ -0.00354759 & -0.00082741 & -0.00627001 & & & & & \end{bmatrix},$$

$$b^{[2]} = \begin{bmatrix} b_{1,1}^{[2]} \end{bmatrix} = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari aşağıdaki gibi sınıflandırmıştır.



Şekil 4.17. $r=8$ için XOR'un Sınıflandırılması

$r = 8$ için meyilli azalım kullanılarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.01288805 & -1.00072345 \\ -1.45187317 & 1.23302499 \\ -0.88431755 & -0.8736174 \\ 1.13515935 & -1.3351207 \\ -0.523234 & -0.51692195 \\ 1.36218319 & 1.25822406 \\ 0.22614088 & -0.26608367 \\ 0.52029578 & -0.61194884 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09623848 \\ 0.03969303 \\ -0.08425485 \\ 0.00940269 \\ -0.04984092 \\ 0.05771055 \\ 0.00213176 \\ 0.00427202 \end{bmatrix},$$

$W^{[2]}$

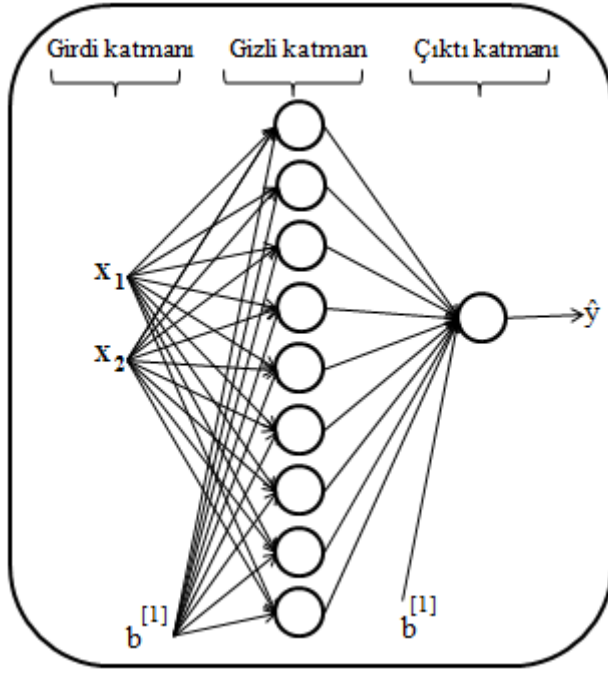
$$= [1.4271997 \quad -1.90503426 \quad 1.24577662 \quad -1.75250057 \quad 0.73708604 \quad 1.85512807 \\ -0.34903288 \quad -0.80321937],$$

$$b^{[2]} = [0.10544392],$$

olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.015296$ dur ve $r = 7$ 'ye göre biraz daha azalmıştır. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %100'dür. $r = 8$ olduğunda da XOR probleminin sınıflandırdığı görülmektedir. Üstelik hata değeri de istenilen hata değerinden küçüktür. Ancak $r = 8$ nöron işlem yükünü arttırdığından $r = 6$ ideal nöron olarak tercih edilir.

4.9. Gizli Katmanındaki Nöron Sayısı Dokuz Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 9$ olmak üzere yapay sinir ağı mimarisi Şekil 4.18'de verilmiştir.



Şekil 4.18. Bir Gizli Katmanlı ve Dokuz Gizli Nöronlu Mimari

Bu mimari için 1., 2., 3., 4., 5., 6., 7. ve 8. nöronların başlangıç parametreleri sabit tutularak başlangıç parametreleri

$$W^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} \\ w_{5,1}^{[1]} & w_{5,2}^{[1]} \\ w_{6,1}^{[1]} & w_{6,2}^{[1]} \\ w_{7,1}^{[1]} & w_{7,2}^{[1]} \\ w_{8,1}^{[1]} & w_{8,2}^{[1]} \\ w_{9,1}^{[1]} & w_{9,2}^{[1]} \end{bmatrix} = \begin{bmatrix} -0.00416758 & -0.00056267 \\ -0.02136196 & 0.01640271 \\ -0.01793436 & -0.00841747 \\ 0.00502881 & -0.01245288 \\ -0.01057952 & -0.00909008 \\ 0.00551454 & 0.02292208 \\ 0.00041539 & -0.01117925 \\ 0.00539058 & -0.0059616 \\ -0.0001913 & 0.01175001 \end{bmatrix} ,$$

$$b^{[1]} = \begin{bmatrix} b_{1,1}^{[1]} \\ b_{2,1}^{[1]} \\ b_{3,1}^{[1]} \\ b_{4,1}^{[1]} \\ b_{5,1}^{[1]} \\ b_{6,1}^{[1]} \\ b_{7,1}^{[1]} \\ b_{8,1}^{[1]} \\ b_{9,1}^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$W^{[2]} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]} & w_{1,4}^{[2]} & w_{1,5}^{[2]} & w_{1,6}^{[2]} & w_{1,7}^{[2]} & w_{1,8}^{[2]} & w_{1,9}^{[2]} \end{bmatrix}$$

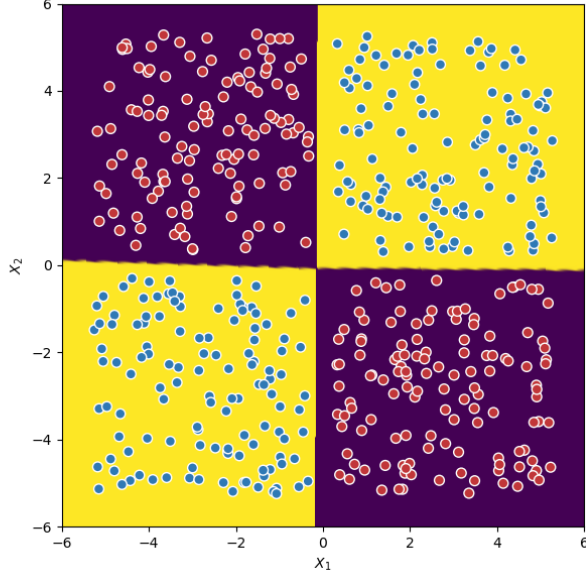
$$= [0.01788628 \quad 0.0043651 \quad 0.00096497 \quad -0.01863493 \quad -0.00277388$$

$$- 0.00354759 \quad - 0.00082741 \quad - 0.00627001 \quad - 0.00043818],$$

$$b^{[2]} = [b_{1,1}^{[2]}] = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari şekil 4.19'daki gibi sınıflandırmıştır.



Şekil 4.19. $r=9$ için XOR'un Sınıflandırılması

$r = 9$ için meyilli azalım kullanılarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.0136868 & -1.00015171 \\ -1.41406102 & 1.20043858 \\ -0.88495952 & -0.87315561 \\ 1.13434909 & -1.33506879 \\ -0.52362907 & -0.51664252 \\ 1.36094442 & 1.25903768 \\ 0.22597893 & -0.26607988 \\ 0.51996412 & -0.61190068 \\ -0.33406061 & 0.28369982 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09626295 \\ 0.03813728 \\ -0.08431385 \\ 0.01043617 \\ -0.04988762 \\ 0.05770886 \\ 0.00233415 \\ 0.00475978 \\ 0.00873227 \end{bmatrix},$$

$W^{[2]}$

$$= \begin{bmatrix} 1.42736792 & -1.85509327 & 1.24591292 & -1.75194222 & 0.73717397 & 1.85477074 \\ -0.34892638 & -0.80297071 & -0.43820768 \end{bmatrix},$$

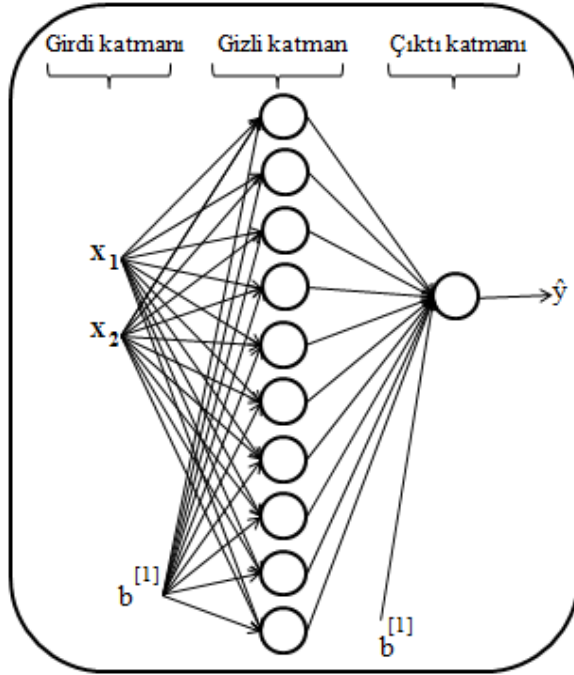
$$b^{[2]} = [0.10862671],$$

olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.015294$ dur ve $r = 8$ 'e göre biraz daha azalmıştır. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %100'dür. $r = 9$ olduğunda da XOR probleminin sınıflandırdığı görülmektedir. Üstelik hata değeri de istenilen

hata değerinden küçüktür. Ancak $r = 9$ nöron işlem yükünü arttırdığından $r = 6$ ideal nöron olarak tercih edilir.

4.10. Gizli Katmanındaki Nöron Sayısı On Olan Yapay Sinir Ağı İçin XOR Probleminin Sınıflandırılması

$m = 500, n = 2, L = 2, r_1 = r = 10$ olmak üzere yapay sinir ağı mimarisi Şekil 4.20’de verilmiştir.



Şekil 4.20. Bir Gizli Katmanlı ve On Gizli Nöronlu Mimarisi

Bu mimari için 1., 2., 3., 4., 5., 6., 7., 8. ve 9. nöronların başlangıç parametreleri sabit tutularak başlangıç parametreleri

$$W^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} \\ w_{5,1}^{[1]} & w_{5,2}^{[1]} \\ w_{6,1}^{[1]} & w_{6,2}^{[1]} \\ w_{7,1}^{[1]} & w_{7,2}^{[1]} \\ w_{8,1}^{[1]} & w_{8,2}^{[1]} \\ w_{9,1}^{[1]} & w_{9,2}^{[1]} \\ w_{10,1}^{[1]} & w_{10,2}^{[1]} \end{bmatrix} = \begin{bmatrix} -0.00416758 & -0.00056267 \\ -0.02136196 & 0.01640271 \\ -0.01793436 & -0.00841747 \\ 0.00502881 & -0.01245288 \\ -0.01057952 & -0.00909008 \\ 0.00551454 & 0.02292208 \\ 0.00041539 & -0.01117925 \\ 0.00539058 & -0.0059616 \\ -0.0001913 & 0.01175001 \\ -0.00747871 & 0.00009025 \end{bmatrix},$$

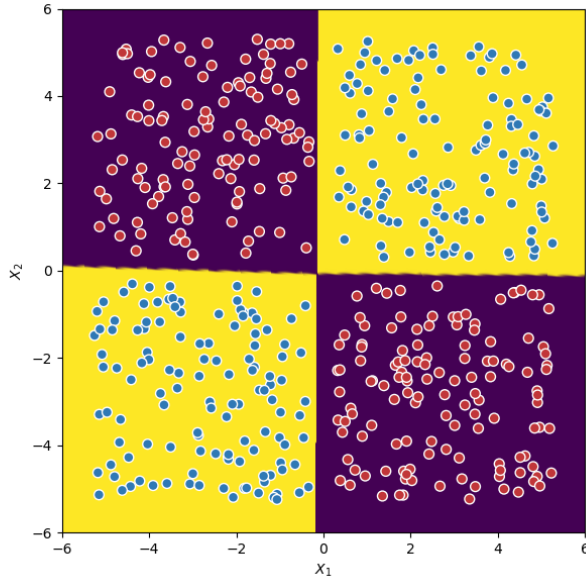
$$b^{[1]} = \begin{bmatrix} b_{1,1}^{[1]} \\ b_{2,1}^{[1]} \\ b_{3,1}^{[1]} \\ b_{4,1}^{[1]} \\ b_{5,1}^{[1]} \\ b_{6,1}^{[1]} \\ b_{7,1}^{[1]} \\ b_{8,1}^{[1]} \\ b_{9,1}^{[1]} \\ b_{10,1}^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$W^{[2]} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]}w_{1,4}^{[2]} & w_{1,5}^{[2]}w_{1,6}^{[2]} & w_{1,7}^{[2]}w_{1,8}^{[2]}w_{1,9}^{[2]}w_{1,10}^{[2]} \\ 0.01788628 & 0.0043651 & 0.00096497 & -0.01863493 & -0.00277388 \\ -0.00354759 & -0.00082741 & -0.00627001 & -0.00043818 & -0.00477218 \end{bmatrix},$$

$$b^{[2]} = [b_{1,1}^{[2]}] = [0],$$

seçilsin ve $\alpha = 0.1$ adım büyüklüğü, 1000 epoch ile bu mimari eğitilsin.

Öğrenmenin sonucunda XOR problemini bu mimari aşağıdaki gibi sınıflandırmıştır.



Şekil 4.21. $r=10$ için XOR'un Sınıflandırılması

$r = 10$ için meyilli azalım kullanılarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.01621998 & -0.99826774 \\ -1.30586294 & 1.10699855 \\ -0.88711832 & -0.87159754 \\ 1.13179119 & -1.33490905 \\ -0.52489964 & -0.51571555 \\ 1.35704233 & 1.26162255 \\ 0.22545358 & -0.26604621 \\ 0.51877754 & -0.6118527 \\ -0.30849259 & 0.2616301 \\ -0.56685727 & 0.48051931 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09568177 \\ 0.03364858 \\ -0.08384424 \\ 0.013574 \\ -0.04961149 \\ 0.05754452 \\ 0.00295515 \\ 0.00617719 \\ 0.00769581 \\ 0.01463178 \end{bmatrix},$$

$W^{[2]}$

$$= [1.42781193 \quad -1.71206169 \quad 1.24632614 \quad -1.75018704 \quad 0.73741024 \quad 1.85366249 \\ -0.34856542 \quad -0.80217601 \quad -0.4044067 \quad -0.74323274],$$

$$b^{[2]} = [0.11803367],$$

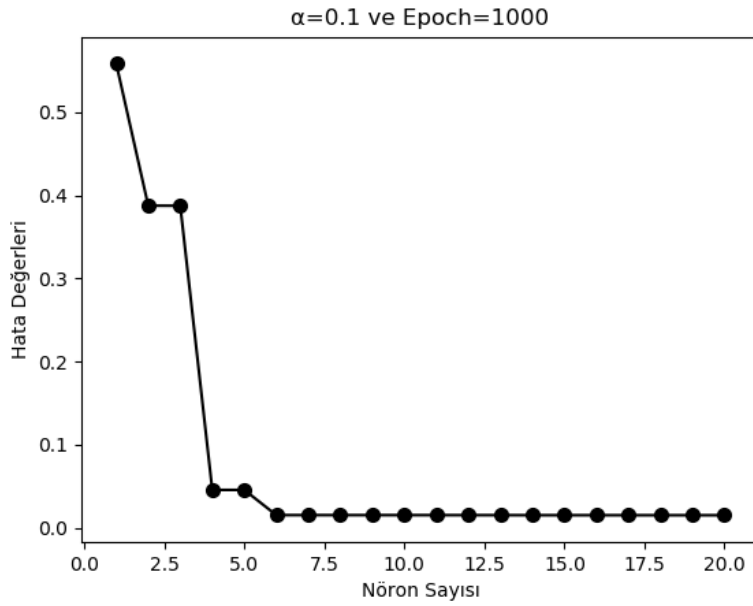
olur. Maliyet fonksiyonu için hesaplanan değer $J(w, b) = 0.015287$ dur ve $r = 9$ 'a göre biraz daha azalmıştır. Bu mimarinin eğitim kümesi üzerindeki başarısı ise %100'dür. $r = 10$ olduğunda da XOR probleminin sınıflandırdığı görülmektedir. Üstelik hata değeri de istenilen hata değerinden küçüktür. Ancak $r = 10$ nöron işlem yükünü arttırdığından $r = 6$ ideal nöron olarak tercih edilir.

Sonuç olarak 1 nörondan 10 nöronlu mimari sonuçları bulunmuştur. Böyle devam edilerek 1'den 20 ye kadar elde edilen nöron sayısına karşılık gelen hata değerleri

$$0.558232, 0.387435, 0.387346, 0.045593, 0.045572, 0.015309, 0.015307, \\ 0.015296, 0.015294, 0.015287, 0.015269, 0.015256, 0.015243, 0.015196, 0.015193, \\ 0.015179, 0.015170, 0.015159, 0.015156, 0.015151$$

bulunmuştur.

Nöron sayısına karşılık gelen hata değerleri grafiği Şekil 4.22’de verilmiştir.



Şekil 4.22. Nöron Sayısı – Hata Değeri Grafiği

Şekil 4.22’ye göre 6 nöronun sonrasındaki hata değerinde çok küçük miktarda değişim olmuştur. Üstelik $r = 6$ ’da hesaplanan hata değeri de istenilen hata değerinden küçük olduğundan ve 7,8,9,...,20 nöronlu mimariler işlem yükünü arttıracığından atanan başlangıç parametrelerine göre XOR eğitim kümesi için ideal nöron sayısının 6 olduğu söylenebilir.

İdeal nöron sayısını tespit etmek için 5. bölümde baskı maliyet fonksiyonu tanımlanmıştır. Baskı maliyet fonksiyonu ile r ve $r+1$ nörona sahip olan mimariler karşılaştırılarak ideal nöron bulunur.

5. BASKI MALİYET FONKSİYONU

Bir gizli katmanlı yapay sinir ağının gizli katmanındaki nöronlara bağlı parametrelerin değerleri sıfıra yakın olan nöronlar gereksizdir. Çünkü bunların algoritmaya bir katkısı olmayacaktır. Gizli katmandaki nöronlar arttırdıkça gereksiz olan nöronlara bağlı parametrelerin sıfıra gideceği hipotezi ortaya atılmıştır. Buradan yola çıkılarak amacı sonuncu nöronun parametrelerini sıfıra götüren bir maliyet fonksiyonu tanımlanmıştır ve **baskı maliyet fonksiyonu** olarak adlandırılmıştır.

Bir gizli katmanlı yapay sinir ağının sonuncu nöronu r^s ve bu nörona bağlı olan parametreler w^s olsun. Bu sonuncu nörondaki parametrelerin karelerinin toplamı olacak şekilde

$$\Omega(w^s) = \left[\sum_{i=1}^n (w_{r^s,i}^{[1]})^2 \right] + (b_{r^s,1}^{[1]})^2 + (w_{1,r^s}^{[2]})^2$$

fonksiyonu tanımlansın. O halde yeni maliyet fonksiyonu bir $\gamma \geq 0$ için

$$J_{baski}(w, b) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log a^{[2]}(x^{(i)}) + (1 - y^{(i)}) \log(1 - a^{[2]}(x^{(i)}))] + \gamma \Omega(w^s)$$

şeklinde tanımlanır.

Burada r ideal nöron sayısı ve ε istenilen maliyet değeri olduğu varsayılırsa r için hesaplanılan maliyet değeri ile r için hesaplanılan baskı maliyet değeri arasındaki ilişki

$$J(W, b) \leq J_{baski}(W, b) \leq \varepsilon$$

olduğunu gösterelim. Yukarıda $J_{baski}(W, b)$ fonksiyonunun tanımından ve $\gamma \geq 0, \Omega(w^s) \geq 0$ olduğundan sonuç görülür.

Baskı maliyet fonksiyonu $J_{baski}(W, b)$ tanımındaki γ parametresinin rolü $\Omega(w^s)$ değerini ve buna özdeş olarak $w_{r^s,i}^{[1]}, b_{r^s,1}^{[1]}, w_{1,r^s}^{[2]}$ parametrelerini kontrol etmektir. Bu parametreleri sıfıra götürmek için uygun bir γ değeri seçilebilir. Bu sayede r nöronlu ve $r+1$ nöronlu yapay sinir ağlarının maliyet fonksiyonları ve optimal parametreleri karşılaştırılabilir.

Yaptığımız gözlemler bunu doğrular niteliktedir. Dolayısıyla r nöronlu yapay sinir ağının global minimumu $r+1$ nöronlu yapay sinir ağının ya global ya da yerel minimumu olmaktadır. Sonuç olarak eğer r nöronlu yapay sinir ağının global minimumu $r+1$ nöronlu

yapay sinir ağının global minimumuna çok yakınsa r 'nin ideal nöron sayısı olduğu söylenebilir.

Bölüm 5.1'de bölüm 4'de bulunan ideal nöron sayısı baskı maliyet fonksiyonu ile doğrulanmıştır.

5.1. Baskı Maliyet Fonksiyonunun Uygulanması

Bu bölümde bölüm 4'de bulunan ideal nöron sayısı baskı maliyet fonksiyonu kullanılarak teyit edilecektir. XOR veri kümesi alınarak sırasıyla 1,2,3, ... ,10 nöronlu yapay sinir ağı mimarilerine baskı maliyet fonksiyonu uygulanacaktır. Burada nöronların başlangıç parametreleri sabit tutulup adım büyüklüğü $\alpha = 0.1$, $epoch = 1000$, $\gamma = 5$ ve istenen hata değeri $\varepsilon = 0.016$ alınmıştır.

5.1.1. $r = 1$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 1$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.1)'de bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.558232 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %68'dir.

Bu mimari için

$$W^{[1]} = [-0.6466238 \quad -0.59732197]$$

$$b^{[1]} = [-1.55920259]$$

$$W^{[2]} = [1.79013557]$$

$$b^{[2]} = [-0.52373678]$$

$W^{[1]}$, $b^{[1]}$, $W^{[2]}$, $b^{[2]}$ parametreleri elde edilmiştir.

Baskı maliyet değeri ise 0.693139 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %50'dir. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = [0 \quad 0]$$

$$b^{[1]} = [0]$$

$$W^{[2]} = [0]$$

$$b^{[2]} = [0.00800004]$$

olmuştur. Görüldüğü gibi $r = 1$ olduğundan bu nörona bağlı olan parametreler sıfırlanmıştır. Ancak maliyet değeri istenilen maliyet değerinden çok büyüktür.

5.1.2. $r = 2$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 2$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.2)'de bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.387435 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %74'dür. Bu mimari için

$$W^{[1]} = \begin{bmatrix} -1.82200408 & -0.46187805 \\ -1.79698039 & 0.49026884 \end{bmatrix}$$

$$b^{[1]} = \begin{bmatrix} -0.97045222 \\ -0.93350411 \end{bmatrix}$$

$$W^{[2]} = [2.11519597 \quad -2.08308646]$$

$$b^{[2]} = [-0.01868901]$$

$W^{[1]}$, $b^{[1]}$, $W^{[2]}$, $b^{[2]}$ parametreleri elde edilmiştir.

$r = 2$ için baskı maliyet değeri ise 0.558232 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %68'dir. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -0.64662381 & -0.59732197 \\ 0 & 0 \end{bmatrix}$$

$$b^{[1]} = \begin{bmatrix} -1.55920266 \\ 0 \end{bmatrix}$$

$$W^{[2]} = [1.79013564 \quad 0]$$

$$b^{[2]} = [-0.52373678]$$

olmuştur. Görüldüğü gibi $r = 2$ olduğundan 2. nörona bağlı olan parametreler sıfırlanmıştır. Bu parametreler sıfırlandığında mimari $r = 1$ gibi davrandığı gözlenmektedir. Parametreler neredeyse birbirlerine eşittir. Ancak hem maliyet fonksiyonu ile hem de baskı maliyet fonksiyonu ile bulunan hata değerleri istenilen maliyet değerinden oldukça büyüktür.

5.1.3. $r = 3$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 3$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.3)'de bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.387346 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %74'dür.

Bu mimari için

$$W^{[1]} = \begin{bmatrix} -1.37297974 & -0.34979449 \\ -1.79834739 & 0.48925823 \\ -1.19891595 & -0.30548031 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.73644521 \\ -0.93355689 \\ -0.64328836 \end{bmatrix}$$

$$W^{[2]} = [1.59670474 \quad -2.08405655 \quad 1.39418183],$$

$$b^{[2]} = [-0.01842142],$$

bulunmuştur.

$r = 3$ için baskı maliyet değeri ise 0.387435 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %74'dir. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.82200401 & -0.461878047 \\ -1.79698031 & 0.49026886 \\ 0 & 0 \end{bmatrix}$$

$$b^{[1]} = \begin{bmatrix} -0.97045213 \\ -0.93350412 \\ 0 \end{bmatrix}$$

$$W^{[2]} = [2.11519587 \quad -2.0830864 \quad 0]$$

$$b^{[2]} = [-0.01868901]$$

olmuştur. Görüldüğü gibi $r = 3$ olduğundan 2. nörona bağlı olan parametreler sıfırlanmıştır. Bu parametreler sıfırlandığında mimari $r = 2$ gibi davrandığı gözlenmektedir. Parametreler neredeyse birbirlerine eşittir. Ancak hem maliyet fonksiyonu ile hem de baskı maliyet fonksiyonu ile bulunan hata değerleri istenilen hata değerinden büyüktür.

5.1.4. $r = 4$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 4$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.4)'de bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.045593 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %99'dür.

Bu mimari için

$$W^{[1]} = \begin{bmatrix} -1.30730919 & -1.25253798 \\ -2.09783362 & 0.4427557 \\ -1.14131966 & -1.09352531 \\ 0.37280674 & -2.09555605 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.49404586 \\ -0.19895629 \\ -0.43151103 \\ -0.08775851 \end{bmatrix},$$

$$W^{[2]} = [1.87673505 \quad -2.1529249 \quad 1.6383433 \quad -2.13013238],$$

$$b^{[2]} = [2.61344466],$$

bulunmuştur.

$r = 4$ için baskı maliyet değeri ise 0.387345 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %74'dir. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.3729853 & -0.34976278 \\ -1.79837735 & 0.48924833 \\ -1.1989545 & -0.30549996 \\ 0 & 0 \end{bmatrix}$$

$$b^{[1]} = \begin{bmatrix} -0.73651135 \\ -0.9335747 \\ -0.64324862 \\ 0 \end{bmatrix}$$

$$W^{[2]} = [1.59673308 \quad -2.08408805 \quad 1.39420103 \quad 0]$$

$$b^{[2]} = [-0.01842211]$$

olmuştur. Görüldüğü gibi $r = 4$ olduğunda 4. nörona bağlı olan parametreler sıfırlanmıştır. Bu parametreler sıfırlandığında mimari $r = 3$ gibi davrandığı gözlenmektedir. Baskı maliyet fonksiyonunun değeri istenilen maliyet değerinden büyüktür. Üstelik baskı maliyet ile maliyet fonksiyonu arasındaki fark da çok büyüktür.

5.1.5. $r = 5$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 5$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.5)'de bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.045572 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %99'dur.

Bu mimari için

$$W^{[1]} = \begin{bmatrix} -1.21838385 & -1.16753869 \\ -2.09792947 & 0.44282065 \\ -1.06376824 & -1.0192641 \\ 0.37273312 & -2.09576196 \\ -0.62941133 & -0.60309232 \end{bmatrix},$$
$$b^{[1]} = \begin{bmatrix} -0.46160799 \\ -0.1991115 \\ -0.40313222 \\ -0.08767399 \\ -0.23851318 \end{bmatrix},$$

$$W^{[2]} = [1.74952872 \quad -2.15304895 \quad 1.52728159 \quad -2.13032173 \quad 0.90364517],$$
$$b^{[2]} = [2.61352887],$$

$r = 5$ için baskı maliyet değeri ise 0.045593 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %99'dur. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.30730866 & -1.2525375 \\ -2.09783367 & 0.44275568 \\ -1.14132031 & -1.09352605 \\ 0.37280674 & -2.09555616 \\ 0 & 0 \end{bmatrix},$$
$$b^{[1]} = \begin{bmatrix} -0.49404572 \\ -0.19895626 \\ -0.43151136 \\ -0.08775858 \\ 0 \end{bmatrix},$$

$$W^{[2]} = [1.87673432 \quad -2.15292494 \quad 1.63834433 \quad -2.13013248 \quad 0],$$

$$b^{[2]} = [2.61344464],$$

olmuştur. Görüldüğü gibi $r = 5$ olduğunda 5. nörona bağlı olan parametreler sıfırlanmıştır. Bu parametreler sıfırlandığında mimari $r = 4$ gibi davrandığı gözlenmektedir. Parametreler de neredeyse birbirlerine eşittir. Ancak hala maliyet değeri yeterince küçük değildir.

5.1.6. $r = 6$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 6$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.6)'de bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.015309 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %100'dür.

Bu mimari için

$$W^{[1]} = \begin{bmatrix} -1.01372638 & -0.99861548 \\ -1.45087942 & 1.23610499 \\ -0.88501907 & -0.87184679 \\ 1.26884377 & -1.48866592 \\ -0.52365019 & -0.51587034 \\ 1.36103523 & 1.26147237 \end{bmatrix},$$
$$b^{[1]} = \begin{bmatrix} -0.09512789 \\ 0.03635731 \\ -0.0833593 \\ 0.01420144 \\ -0.04932769 \\ 0.05984022 \end{bmatrix},$$

$W^{[2]}$

$$= [1.42624428 \quad -1.90620662 \quad 1.24497431 \quad -1.95609683 \quad 0.73661043 \quad 1.85655843],$$
$$b^{[2]} = [0.09367112],$$

$r = 6$ için baskı maliyet değeri ise 0.045572 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %99'dir. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.21838289 & -1.16753789 \\ -2.09792931 & 0.44282061 \\ -1.06376897 & -1.01926501 \\ 0.37273317 & -2.09576199 \\ -0.62941143 & -0.60309254 \\ 0 & 0 \end{bmatrix},$$
$$b^{[1]} = \begin{bmatrix} -0.46160763 \\ -0.19911122 \\ -0.40313253 \\ -0.08767416 \\ -0.23851324 \\ 0 \end{bmatrix},$$

$$W^{[2]} = [1.74952743 \quad -2.15304876 \quad 1.52728279 \quad -2.13032178 \quad 0.9036454 \quad 0],$$

$$b^{[2]} = [2.61352892],$$

olmuştur. Görüldüğü gibi $r = 6$ olduğunda 6. nörona bağlı olan parametreler sıfırlanmıştır. Bu parametreler sıfırlandığında mimari $r = 5$ gibi davrandığı gözlenmektedir. Parametreler de neredeyse birbirlerine eşittir. Ancak maliyet ve baskı maliyet değerleri arasındaki değer büyüktür.

5.1.7. $r = 7$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 7$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.7)'de bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.015307 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %100'dür.

Bu mimari için

$$W^{[1]} = \begin{bmatrix} -1.01362538 & -0.99897917 \\ -1.45112772 & 1.23552996 \\ -0.88493887 & -0.87215601 \\ 1.24433294 & -1.46066948 \\ -0.52360923 & -0.51604714 \\ 1.36115628 & 1.26091703 \\ 0.24797248 & -0.29115411 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09533964 \\ 0.03697203 \\ -0.08348818 \\ 0.01324979 \\ -0.0494038 \\ 0.05952028 \\ 0.00290295 \end{bmatrix},$$

$W^{[2]}$

$$= [1.4264412 \quad -1.90603459 \quad 1.24514249 \quad -1.91888498 \quad 0.73671022 \quad 1.8562596 \\ -0.38229393],$$

$$b^{[2]} = [0.09572404],$$

$r = 7$ için baskı maliyet değeri ise 0.015309 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %100'dir. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.01372648 & -0.99861543 \\ -1.45087969 & 1.23610484 \\ -0.88501923 & -0.87184682 \\ 1.26884348 & -1.48866607 \\ -0.5236503 & -0.51587037 \\ 1.36103499 & 1.26147236 \\ 0 & 0 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09512794 \\ 0.03635727 \\ -0.08335936 \\ 0.01420147 \\ -0.04932773 \\ 0.05984025 \\ 0 \end{bmatrix},$$

$$W^{[2]} = [1.42624431 \quad -1.90620672 \quad 1.24497446 \\ -1.95609676 \quad 0.73661053 \quad 1.85655825 \quad 0],$$

$$b^{[2]} = [0.09367126],$$

olmuştur. Görüldüğü gibi $r = 7$ olduğunda 7. nörona bağlı olan parametreler sıfırlanmıştır. Bu parametreler sıfırlandığında mimari $r = 6$ gibi davrandığı gözlenmektedir. Parametreler de neredeyse birbirlerine eşittir. Burada iki maliyet değerleri arasındaki fark azalmıştır. Üstelik hata değerleri de istenilen hata değerinden küçüktür. O halde 6 nöronlu yapay sinir ağının ideal nöron olduğu söylenebilir.

5.1.8. $r = 8$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 8$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.8)'de bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.015296 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %100'dür.

Bu mimari için

$$W^{[1]} = \begin{bmatrix} -1.01288805 & -1.00072345 \\ -1.45187317 & 1.23302499 \\ -0.88431755 & -0.8736174 \\ 1.13515935 & -1.3351207 \\ -0.523234 & -0.51692195 \\ 1.36218319 & 1.25822406 \\ 0.22614088 & -0.26608367 \\ 0.52029578 & -0.61194884 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09623848 \\ 0.03969303 \\ -0.08425485 \\ 0.00940269 \\ -0.04984092 \\ 0.05771055 \\ 0.00213176 \\ 0.00427202 \end{bmatrix},$$

$W^{[2]}$

$$= [1.4271997 \quad -1.90503426 \quad 1.24577662 \quad -1.75250057 \quad 0.73708604 \quad 1.85512807 \\ -0.34903288 \quad -0.80321937],$$

$$b^{[2]} = [0.10544392],$$

$r = 8$ için baskı maliyet değeri ise 0.015307 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %100'dir. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.01363128 & -0.99897181 \\ -1.45113134 & 1.23553726 \\ -0.88494441 & -0.8721492 \\ 1.24432775 & -1.46066505 \\ -0.52361267 & -0.51604326 \\ 1.36114607 & 1.26093123 \\ 0.24796765 & -0.2911491 \\ 0 & 0 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09534104 \\ 0.03696949 \\ -0.08348932 \\ 0.01324742 \\ -0.0494045 \\ 0.05950518 \\ 0.00290243 \\ 0 \end{bmatrix},$$

$$W^{[2]} = [1.42644034 \quad -1.90604203 \quad 1.24514173 \\ -1.91887823 \quad 0.73671 \quad 1.85626128 \quad -0.38228698 \quad 0],$$

$$b^{[2]} = [0.09572821],$$

olmuştur. Görüldüğü gibi $r = 8$ olduğunda 8. nörona bağlı olan parametreler sıfırlanmıştır. Bu parametreler sıfırlandığında mimari $r = 7$ gibi davrandığı gözlenmektedir. Parametreler de neredeyse birbirlerine eşittir. Burada iki maliyet değerleri arasındaki fark azalmıştır. Aynı zamanda hata değerleri istenilen hata değerinden de küçüktür. Ancak $r = 6$ koşulları sağladığından $r = 7$ işlem yükünü arttıracığından 6 daha idealdir.

5.1.9. $r = 9$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 9$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.9)'de bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.015294 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %100'dür. Bu mimari için

$$W^{[1]} = \begin{bmatrix} -1.0136868 & -1.00015171 \\ -1.41406102 & 1.20043858 \\ -0.88495952 & -0.87315561 \\ 1.13434909 & -1.33506879 \\ -0.52362907 & -0.51664252 \\ 1.36094442 & 1.25903768 \\ 0.22597893 & -0.26607988 \\ 0.51996412 & -0.61190068 \\ -0.33406061 & 0.28369982 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09626295 \\ 0.03813728 \\ -0.08431385 \\ 0.01043617 \\ -0.04988762 \\ 0.05770886 \\ 0.00233415 \\ 0.00475978 \\ 0.00873227 \end{bmatrix},$$

$W^{[2]}$

$$= [1.42736792 \quad -1.85509327 \quad 1.24591292 \quad -1.75194222 \quad 0.73717397 \quad 1.85477074 \\ -0.34892638 \quad -0.80297071 \quad -0.43820768],$$

$$b^{[2]} = [0.10862671],$$

$r = 9$ için baskı maliyet değeri ise 0.015296 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %100'dir. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.01288764 & -1.00072326 \\ -1.45187234 & 1.23302567 \\ -0.88431727 & -0.87361733 \\ 1.13516015 & -1.33512012 \\ -0.52323384 & -0.51692191 \\ 1.36218383 & 1.25822434 \\ 0.22614107 & -0.26608359 \\ 0.52029613 & -0.61194855 \\ 0 & 0 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09623828 \\ 0.03969311 \\ -0.08425468 \\ 0.00940272 \\ -0.04984082 \\ 0.05771046 \\ 0.00213177 \\ 0.00427203 \\ 0 \end{bmatrix},$$

$$W^{[2]} = [1.42719925 \quad -1.90503407 \quad 1.24577636 \\ -1.75250065 \quad 0.73708588 \quad 1.85512872 \quad -0.34903295 \\ -0.80321938 \quad 0],$$

$$b^{[2]} = [0.10544327],$$

olmuştur. Görüldüğü gibi $r = 9$ olduğunda 9. nörona bağlı olan parametreler sıfırlanmıştır. Bu parametreler sıfırlandığında mimari $r = 8$ gibi davrandığı gözlenmektedir. Parametreler de neredeyse birbirlerine eşittir. Burada iki maliyet değerleri arasındaki fark azalmıştır. Aynı zamanda hata değerleri istenilen hata değerinden de küçüktür. Ancak $r = 6$ koşulları sağladığından $r = 8$ işlem yükünü arttıracığından 6 daha idealdir.

5.1.10. $r = 10$ İçin Baskı Maliyet Fonksiyonunun Uygulanması

$r = 10$ için maliyet değeri, parametreler ve veri kümesi üzerindeki başarısı (4.10)'da bulunmuştur. Buradan alınan sonuçlar şöyledir;

Maliyet değeri 0.015287 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %100'dür.

Bu mimari için

$$W^{[1]} = \begin{bmatrix} -1.01621998 & -0.99826774 \\ -1.30586294 & 1.10699855 \\ -0.88711832 & -0.87159754 \\ 1.13179119 & -1.33490905 \\ -0.52489964 & -0.51571555 \\ 1.35704233 & 1.26162255 \\ 0.22545358 & -0.26604621 \\ 0.51877754 & -0.6118527 \\ -0.30849259 & 0.2616301 \\ -0.56685727 & 0.48051931 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09568177 \\ 0.03364858 \\ -0.08384424 \\ 0.013574 \\ -0.04961149 \\ 0.05754452 \\ 0.00295515 \\ 0.00617719 \\ 0.00769581 \\ 0.01463178 \end{bmatrix},$$

$W^{[2]}$

$$= [1.42781193 \quad -1.71206169 \quad 1.24632614 \quad -1.75018704 \quad 0.73741024 \quad 1.85366249 \\ -0.34856542 \quad -0.80217601 \quad -0.4044067 \quad -0.74323274],$$

$$b^{[2]} = [0.11803367],$$

$r = 10$ için baskı maliyet değeri ise 0.015294 olarak hesaplanmıştır ve veri kümesi üzerindeki başarısı %100'dir. Baskı maliyet fonksiyonu kullanarak bulunan parametreler

$$W^{[1]} = \begin{bmatrix} -1.01368563 & -1.00015118 \\ -1.41406032 & 1.20043931 \\ -0.88495969 & -0.87315645 \\ 1.13435 & -1.33506895 \\ -0.5236279 & -0.51664364 \\ 1.36094559 & 1.25903744 \\ 0.22597836 & -0.26607867 \\ 0.51996503 & -0.61190035 \\ -0.33405825 & 0.28369829 \\ 0 & 0 \end{bmatrix},$$

$$b^{[1]} = \begin{bmatrix} -0.09626264 \\ 0.03813751 \\ -0.08431371 \\ 0.01043585 \\ -0.04988772 \\ 0.05770876 \\ 0.00233414 \\ 0.00475978 \\ 0.00873227 \\ 0 \end{bmatrix},$$

$$W^{[2]} = \begin{bmatrix} 1.4273667 & -1.85509321 & 1.24591362 \\ -1.75194293 & 0.73717393 & 1.85477144 & -0.34892509 \\ -0.80297105 & -0.43820488 & 0 \end{bmatrix},$$

$$b^{[2]} = [0.1086252],$$

olmuştur. Görüldüğü gibi $r = 10$ olduğunda 10. nörona bağlı olan parametreler sıfırlanmıştır. Bu parametreler sıfırlandığında mimari $r = 9$ gibi davrandığı gözlenmektedir. Parametreler de neredeyse birbirlerine eşittir. Burada iki maliyet değerleri arasındaki fark azalmıştır. Aynı zamanda hata değerleri istenilen hata değerinden de küçüktür. Ancak $r = 6$ koşulları sağladığından $r = 9$ işlem yükünü arttıracığından 6 daha idealdir.

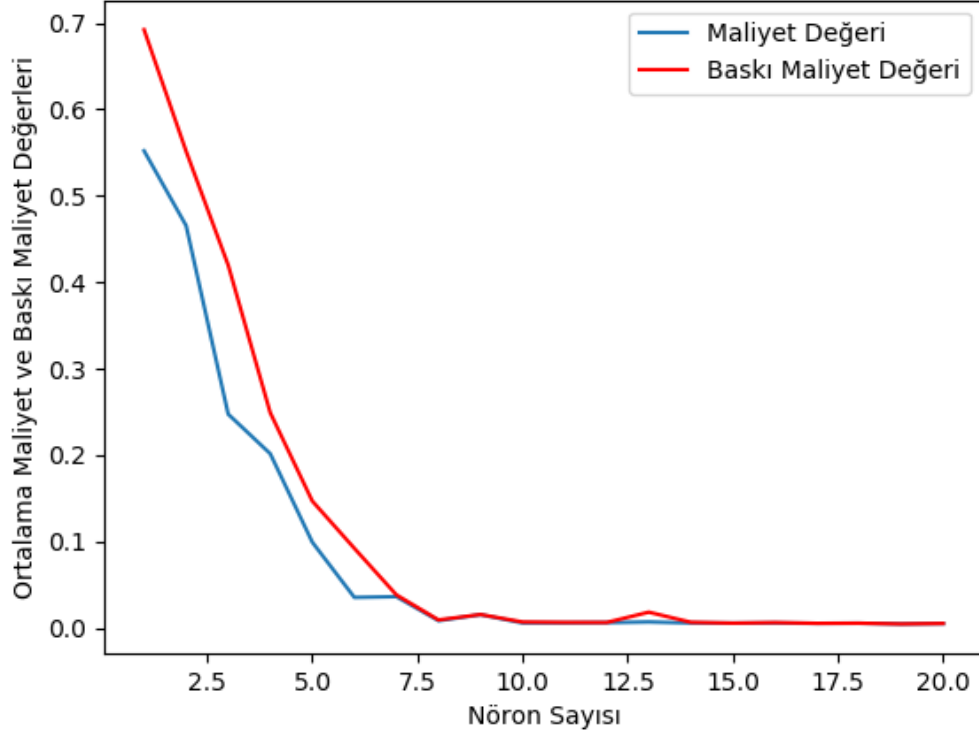
Sonuç olarak 4. Bölümde bulunan $r = 6$ nöron sayısı XOR için ideal nöron olduğu bulunmuştu. Burada da baskı maliyet fonksiyonu ile doğrulanmıştır. Yani 6 ideal nöron sayısı olduğundan 7 için hesaplanan maliyet değeri, 7 için hesaplanan baskı maliyet değeri arasındaki fark çok küçük ve aralarındaki ilişki $\text{maliyet değeri} \leq \text{baskı maliyet değeri} \leq \epsilon$ olmuştur.

Yapay sinir ağları non-konveks bir problem olduğundan farklı başlangıç parametreleri ile algoritma çalıştırıldığında ideal nöron sayısı değişmektedir. Üstelik farklı bir XOR eğitim kümesi kullanıldığında bile değişebilmektedir. Başlangıç parametreleri ve eğitim kümesine bağlılığı azaltmak adına bölüm 6'da bir deney yapılmıştır.

6. DENEY

XOR eğitim kümesi bir gizli katmanlı yapay sinir ağı modellerine uygulanarak gizli katmandaki en uygun nöron sayısını bulmak amaçlanmıştır. Bölüm 4’de bir eğitim kümesi için başlangıç parametreleri sabit tutulup, $\alpha = 0.1$, $epoch = 1000$ olmak üzere sonuçlar elde edildiğinde 6 nöron ideal olduğu gözlenmiştir. Ancak tecrübelerden edinildiği üzere farklı eğitim kümesi ve farklı başlangıç noktaları verildiğinde 6 nöronun bazen XOR problemini sınıflandıramadığı tespit edilmiştir. Buradan yola çıkarak bu deneyde her hangi bir XOR eğitim kümesi verildiğinde hangi başlangıç noktasından başlanılırsa başlansın XOR problemini sınıflandırabilen bir gizli katmanlı yapay sinir ağı mimarisini bulmak hedeflenmiştir.

Deney için 500 tane veri içeren 100 tane eğitim kümesi kullanılmıştır. Deneyde epoch sayısı 2500 ve adım büyüklüğü $\alpha = 0.1$ olarak atanmıştır. Bir eğitim kümesi sırasıyla $r = 1, 2, \dots, 20$ nöronlu modeller ile her bir nöron 20 kez farklı başlangıç parametreleriyle çalıştırılarak $r = 1, 2, \dots, 20$ nöron için ayrı ayrı ortalama maliyet değerleri hesaplanmıştır. Yani $r = 1$ için ortalama bir maliyet değeri, $r = 2$ için ayrı ve bu şekilde $r = 20$ ’ ye kadar ortalama maliyet değerleri hesaplanmıştır. Daha sonra 100 farklı eğitim kümesi için bu maliyet değerleri hesaplanmış ve bu değerlerinde nöron sayıları göz önüne alınarak 1 nöronlu maliyet değerlerinin ortalaması, 2 nöronlu maliyet değerlerinin ortalaması ve böyle devam ederek 20 nöronlu maliyet değerlerinin ortalaması hesaplanmıştır. Sonuç olarak $r = 1, 2, \dots, 20$ nöron için bir maliyet değeri elde edilmiştir. Aynı deney için baskı maliyet fonksiyonu kullanılarak bulunan ideal nöron sayısı doğrulanmıştır. Şekil 6.1’de nöron sayılarına karşılık gelen ortalama maliyet değerlerini ve baskı maliyet değerini veren grafik verilmiştir.



Şekil 6.1. Deney Sonucu

Eğer istenilen hata değeri $\varepsilon = 0.006$ seçilirse tablo 6.1'den de anlaşıldığı gibi 15 nöronlu mimaride maliyet değeri ile baskı maliyet değeri arasındaki fark çok düşük ve üstelik istenilen hata değerinden de bu değerler küçük olduğundan ideal nöron sayısı 15 denilebilir. Diğer mimarilerde de hemen hemen aynı hata değeri olduğundan bunlarda ideale yakındır ancak işlem yükünü arttıracığından 15 ideal olarak alınabilir. Bulunan maliyet ve baskı maliyet değerleri ve bunlar arasındaki fark tablo 6.1'de verilmiştir.

Tablo 6.1. Deney Sonuçları

Nöron Sayısı	Maliyet Değeri	Baskı Maliyet Değeri	Fark
1	0.5520670020329068	0.6921783806026041	0.1401113785696972
2	0.46547245095623907	0.5515601466379715	0.0860876956817324
3	0.24743623593806016	0.41986481301183903	0.1724285770737788
4	0.20167036392363083	0.24908169514017092	0.0474113312165400
5	0.09931327273111411	0.14687251754237113	0.0475592448112570
6	0.035437656557939214	0.09221384388901188	0.0567761873310726
7	0.03626082199908032	0.03831273109639588	0.0020519090973155
8	0.008476381189197984	0.009384463403277246	0.0009080822140792

9	0.015430106402450383	0.015617226230962403	0.0001871198285120
10	0.005852566048284839	0.006964557257608486	0.0011119912093236
11	0.005929366978113975	0.006512395718898259	0.0005830287407842
12	0.006250282805678232	0.006439205599969103	0.0001889227942908
13	0.007052614467811636	0.018355013817960543	0.0113023993501489
14	0.005827314741028232	0.006664315198096034	0.0008370004570678
15	0.005657146686394957	0.005659121325297955	0.0000019746389029
16	0.005812342801592794	0.006311873854130176	0.0004995310525373
17	0.00541479443821729	0.005418560261672685	0.0000037658234553
18	0.005574566110075312	0.005576205677846589	0.0000016395677712
19	0.004700078690093847	0.004701800525528006	0.0000017218354341
20	0.005124062683954621	0.005262674083405796	0.0001386113994511

Deneyde eğitim kümesinin birden fazla seçilmesi ile yapay sinir ağı modelinin veriye bağımlılığının azaltılması ve aynı zamanda her nöron farklı başlangıç parametreleri ile 20 kez çalıştırılarak da başlangıç parametrelerine olan bağımlılığın azaltılması amaçlanmıştır. Sonuç olarak eğitim kümesinden ve başlangıç parametrelerinden bağımsız olarak XOR problemi için $r = 15$ ideal nöron sayısı olduğu tespit edilmiştir.

7. XOR PROBLEMİNİN İKİ GİZLİ KATMANLI YAPAY SİNİR AĞLARI İLE SINIFLANDIRILMASI

XOR eğitim kümesi olarak 4. Bölüm de alınan eğitim kümesi alınacaktır. Alınan eğitim kümesi görseli Şekil 4.1’de verilmiştir. Başlangıç parametreleri olarak 4. Bölümde alınan başlangıç parametreleri kullanılmıştır. Burada bir katman fazla olduğundan oraya yeni başlangıç parametreleri atanmış ve bu parametreler her başlangıçta sabit tutulmuştur. Uygun adım büyüklüğü $\alpha = 0.3$ ve epoch = 5000 olarak atanmıştır.

Bu bölümde iki gizli katmanlı yapay sinir ağları ile XOR eğitim kümesini sınıflandıran ideal nöron sayısı r_1, r_2 değerleri incelenmiştir.

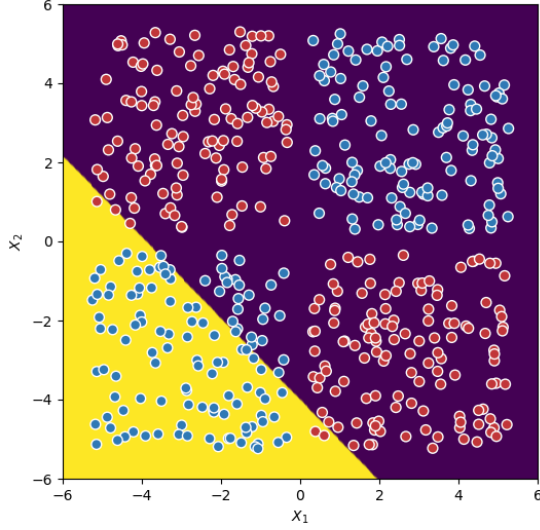
7.1. Birinci Gizli Katmanında $r_1 = 1$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

Birinci gizli katmanında $r_1 = 1$ ve ikinci gizli katmanında sırasıyla $r_2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.1’de verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.1. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
1	1	%68	0.549463	X
1	2	%69	0.549452	X
1	3	%68	0.549462	X
1	4	%68	0.549462	X
1	5	%68	0.549462	X
1	6	%68	0.549462	X
1	7	%68	0.549462	X
1	8	%68	0.549462	X
1	9	%68	0.549462	X
1	10	%68	0.549462	X

Tablo 7.1’de görüldüğü gibi verilen mimariler lokal minimuma takılmıştır. Bu mimariler için XOR eğitim kümesi sınıflandırılmamıştır. Tablo 7.1 için en az hatayı $r_1 = 1$ ve $r_2 = 2$ nöronu olan mimari vermiştir. Verilen eğitim kümesini bu mimari Şekil 7.1’deki gibi sınıflandırmıştır. Buradan görülmektedir ki eğitim kümesini sınıflandırma konusunda bu mimari yetersiz kalmıştır.



Şekil 7.1. $r_1 = 1$ ve $r_2 = 2$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Sonuç olarak burada $r_1 = 1$ ve sırasıyla $r_2 = 1, 2, \dots, 10$ nöron içeren iki gizli katmanlı yapay sinir ağları verilen eğitim kümesini sınıflandıramamıştır. O halde buradaki mimariler ideal nöron sayılarını vermemektedir.

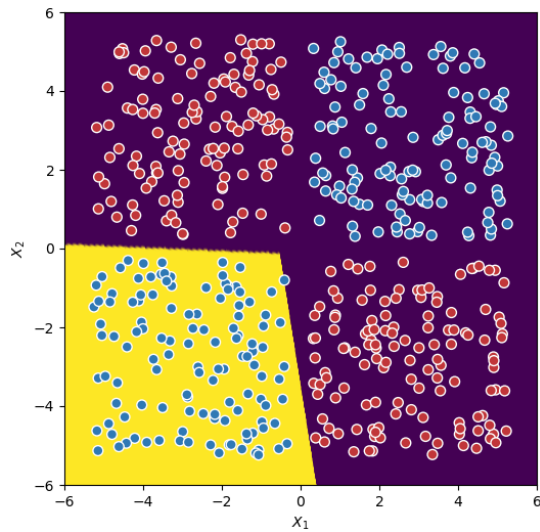
7.2. Birinci Gizli Katmanında $r_1 = 2$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

Birinci gizli katmanında $r_1 = 2$ ve ikinci gizli katmanında sırasıyla $r_2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.2’de verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.2. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
2	1	%74	0.369392	X
2	2	%74	0.369383	X
2	3	%74	0.369383	X
2	4	%74	0.369383	X
2	5	%74	0.369383	X
2	6	%74	0.356701	X
2	7	%74	0.359366	X
2	8	%74	0.356709	X
2	9	%74	0.356709	X
2	10	%74	0.359389	X

Tablo 7.2’de görüldüğü gibi verilen mimariler lokal minimuma takılmıştır. Bu mimariler için XOR eğitim kümesi sınıflandırılmamıştır. Tablo 7.2 için en az hatayı $r_1 = 2$ ve $r_2 = 6$ nöronu olan mimari vermiştir. Verilen eğitim kümesini bu mimari Şekil 7.2’deki gibi sınıflandırmıştır. Buradan görülmektedir ki eğitim kümesini sınıflandırma konusunda bu mimari yetersiz kalmıştır.



Şekil 7.2. $r_1 = 2$ ve $r_2 = 6$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Sonuç olarak burada $r_1 = 2$ ve sırasıyla $r_2 = 1, 2, \dots, 10$ nöron içeren iki gizli katmanlı yapay sinir ağları verilen eğitim kümesini sınıflandıramamıştır. Bu yüzden buradaki mimariler ideal nöron sayılarını vermemektedir.

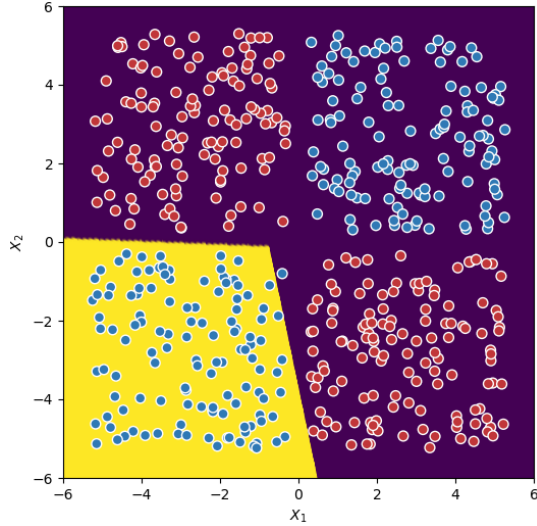
7.3. Birinci Gizli Katmanında $r_1 = 3$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

Birinci gizli katmanında $r_1 = 3$ ve ikinci gizli katmanında sırasıyla $r_2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.3’de verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.3. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
3	1	%74	0.369388	X
3	2	%74	0.369388	X
3	3	%74	0.369388	X
3	4	%74	0.369346	X
3	5	%74	0.359365	X
3	6	%74	0.359365	X
3	7	%74	0.359360	X
3	8	%74	0.359344	X
3	9	%74	0.359329	X
3	10	%74	0.359329	X

Tablo 7.3’de görüldüğü gibi verilen mimariler lokal minimuma takılmıştır. Bu mimariler için XOR eğitim kümesi sınıflandırılmamıştır. Tablo 7.3 için en az hatayı $r_1 = 3$ ve sırasıyla $r_2 = 9, 10$ nöronu olan mimariler vermiştir. Verilen eğitim kümesini $r_1 = 3$ ve $r_2 = 9$ nöronu olan mimari Şekil 7.3’deki gibi sınıflandırmıştır. Buradan görülmektedir ki eğitim kümesini sınıflandırma konusunda bu mimariler de yetersiz kalmıştır.



Şekil 7.3. $r_1 = 3$ ve $r_2 = 9$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Sonuç olarak burada $r_1 = 3$ ve sırasıyla $r_2 = 1, 2, \dots, 10$ nöron içeren iki gizli katmanlı yapay sinir ağı verilen eğitim kümesini sınıflandıramamıştır. Buradaki mimariler ideal nöron sayılarını vermemektedir.

7.4. Birinci Gizli Katmanında $r_1 = 4$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

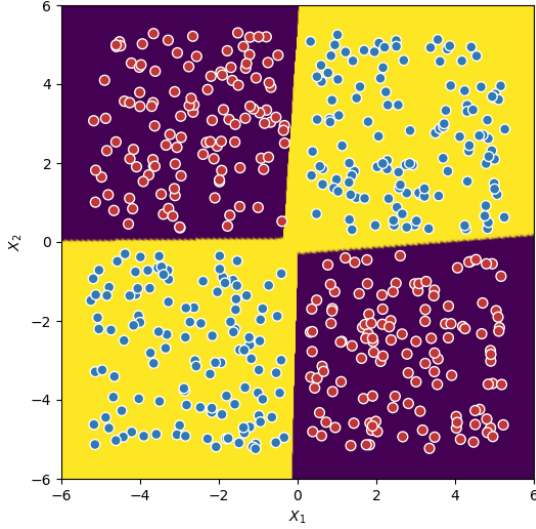
Birinci gizli katmanında $r_1 = 4$ ve ikinci gizli katmanında sırasıyla $r_2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.4’de verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.4. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
4	1	%50	0.693139	X
4	2	%50	0.693139	X
4	3	%50	0.693139	X
4	4	%100	0.000501	√
4	5	%100	0.000992	√
4	6	%100	0.000987	√

4	7	%100	0.000975	√
4	8	%100	0.000190	√
4	9	%100	0.000190	√
4	10	%100	0.000189	√

Tablo 7.4’de görüldüğü gibi $r_1 = 4$ ve sırasıyla $r_2 = 1,2,3$ nöronu olan mimariler lokal minimuma takılmıştır. Bu mimariler için XOR eğitim kümesi sınıflandırılmamıştır. Ancak $r_1 = 4$ ve sırasıyla $r_2 = 4,5,6,7,8,9,10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Tablo 7.4 için en az hatayı $r_1 = 4$ ve $r_2 = 10$ nöronu olan mimari vermiştir. Verilen eğitim kümesini bu mimari Şekil 7.4’deki gibi sınıflandırmıştır.



Şekil 7.4. $r_1 = 4$ ve $r_2 = 10$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Sonuç olarak burada $r_1 = 4$ ve sırasıyla $r_2 = 1,2,3$ nöronu olan iki gizli katmanlı yapay sinir ağları eğitim kümesini sınıflandıramazken, $r_1 = 4$ ve sırasıyla $r_2 = 4,5,6,7,8,9,10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Üstelik bölüm 7.1, 7.2 ve 7.3’deki mimarilere göre hata değeri oldukça azalarak 0.000189 değerini almıştır. Buradaki ideal nöron sayıları $r_1 = 4$ ve $r_2 = 10$ ’dur denilebilir.

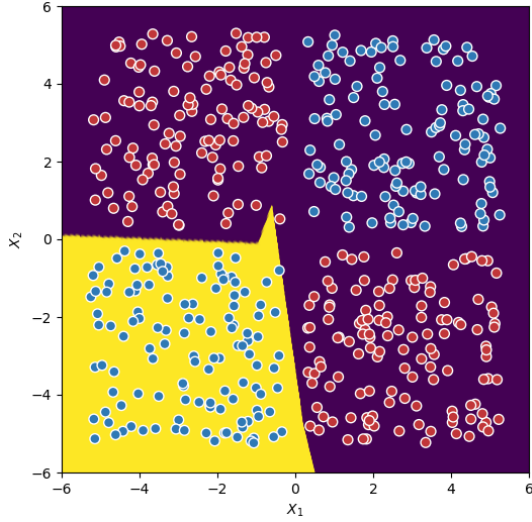
7.5. Birinci Gizli Katmanında $r_1 = 5$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

Birinci gizli katmanında $r_1 = 5$ ve ikinci gizli katmanında sırasıyla $r_2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.5’de verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.5. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
5	1	%50	0.693139	X
5	2	%50	0.693139	X
5	3	%74	0.360178	X
5	4	%74	0.360193	X
5	5	%74	0.369357	X
5	6	%74	0.369360	X
5	7	%74	0.369370	X
5	8	%74	0.358947	X
5	9	%74	0.360155	X
5	10	%74	0.360155	X

Tablo 7.5’de görüldüğü gibi $r_1 = 5$ ve sırasıyla $r_2 = 1, 2$ nöronu olan mimariler lokal minimuma takılmış ve $r_1 = 5$ ve sırasıyla $r_2 = 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimarilerde başka bir lokal minimuma takılmışlardır. Bu mimarilerin hiç birisi XOR eğitim kümesini sınıflandıramamıştır. Tablo 7.5 için en az hatayı $r_1 = 5$ ve $r_2 = 8$ nöronu olan mimari vermiştir. Verilen eğitim kümesini bu mimari Şekil 7.5’deki gibi sınıflandırmıştır. Buradan görülmektedir ki eğitim kümesini sınıflandırma konusunda bu mimariler de yetersiz kalmıştır.



Şekil 7.5. $r_1 = 5$ ve $r_2 = 8$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Sonuç olarak burada $r_1 = 5$ ve sırasıyla $r_2 = 1, 2, 3, 6, 7, 8, 9, 10$ nöronu olan iki gizli katmanlı yapay sinir ağları eğitim kümesini sınıflandıramamıştır. Buradaki mimariler ideal nöron sayılarını vermemektedir.

7.6. Birinci Gizli Katmanında $r_1 = 6$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

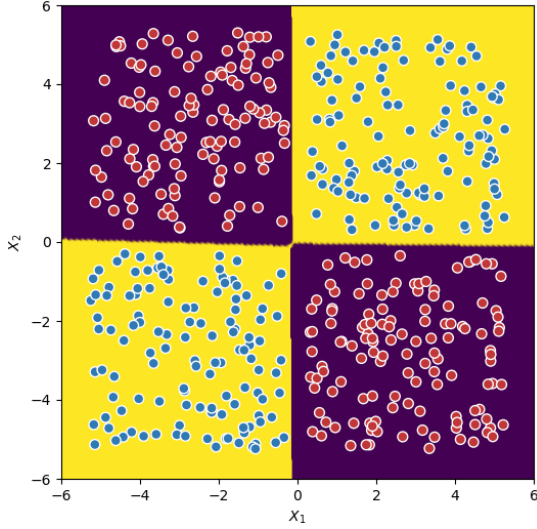
Birinci gizli katmanında $r_1 = 6$ ve ikinci gizli katmanında sırasıyla $r_2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.6’da verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.6. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
6	1	%50	0.693139	X
6	2	%100	0.000928	√
6	3	%100	0.000929	√
6	4	%100	0.000873	√
6	5	%100	0.000873	√
6	6	%100	0.000115	√

6	7	%100	0.000111	√
6	8	%100	0.000111	√
6	9	%100	0.000110	√
6	10	%100	0.000110	√

Tablo 7.6’da görüldüğü gibi $r_1 = 6$ ve $r_2 = 1$ nöronu olan mimari lokal minimuma takılmış ve $r_1 = 6$ ve sırasıyla $r_2 = 2,3,4,5,6,7,8,9,10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Tablo 7.6 için en az hatayı $r_1 = 6$ ve sırasıyla $r_2 = 9,10$ nöronu olan mimariler vermiştir. Verilen eğitim kümesini $r_1 = 6$ ve $r_2 = 9$ nöronu olan mimari Şekil 7.6’daki gibi sınıflandırmıştır.



Şekil 7.6. $r_1 = 6$ ve $r_2 = 9$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Sonuç olarak burada $r_1 = 6$ ve $r_2 = 1$ nöronu olan iki gizli katmanlı yapay sinir ağı eğitim kümesini sınıflandıramazken, $r_1 = 6$ ve sırasıyla $r_2 = 2,3,4,5,6,7,8,9,10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Üstelik bölüm 7.4’de bulunan hata değeri 0.000079 azalarak 0.000110 değerini almıştır. O halde ideal nöron sayıları $r_1 = 6$ ve $r_2 = 9$ ’dur denilebilir.

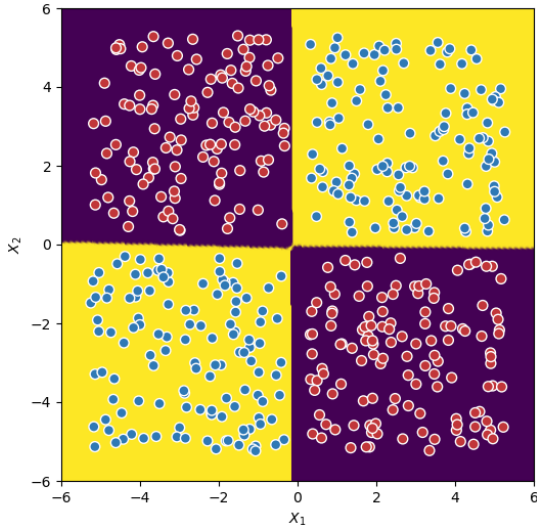
7.7. Birinci Gizli Katmanında $r_1 = 7$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

Birinci gizli katmanında $r_1 = 7$ ve ikinci gizli katmanında sırasıyla $r_2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.7’de verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.7. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
7	1	%50	0.693139	X
7	2	%100	0.000877	√
7	3	%100	0.000877	√
7	4	%100	0.000876	√
7	5	%100	0.000118	√
7	6	%100	0.000118	√
7	7	%100	0.000114	√
7	8	%100	0.000114	√
7	9	%100	0.000111	√
7	10	%100	0.000113	√

Tablo 7.7’de görüldüğü gibi $r_1 = 7$ ve $r_2 = 1$ nöronu olan mimari lokal minimuma takılmış ve $r_1 = 7$ ve sırasıyla $r_2 = 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Tablo 7.7 için en az hatayı $r_1 = 7$ ve $r_2 = 9$ nöronu olan mimari vermiştir. Verilen eğitim kümesini $r_1 = 7$ ve $r_2 = 9$ nöronu olan mimari Şekil 7.7’deki gibi sınıflandırmıştır.



Şekil 7.7. $r_1 = 7$ ve $r_2 = 9$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Sonuç olarak burada $r_1 = 7$ ve $r_2 = 1$ nöronu olan iki gizli katmanlı yapay sinir ağı eğitim kümesini sınıflandıramazken, $r_1 = 7$ ve sırasıyla $r_2 = 2,3,4,5,6,7,8,9,10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Burada bölüm 7.6’da bulunan hata değeri 0.000001 artarak 0.000111 değerini almıştır. O halde bölüm 7.6’daki hata değeri az olduğundan ideal nöron sayıları bölüm 7.6’daki nöron sayılarıdır.

7.8. Birinci Gizli Katmanında $r_1 = 8$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

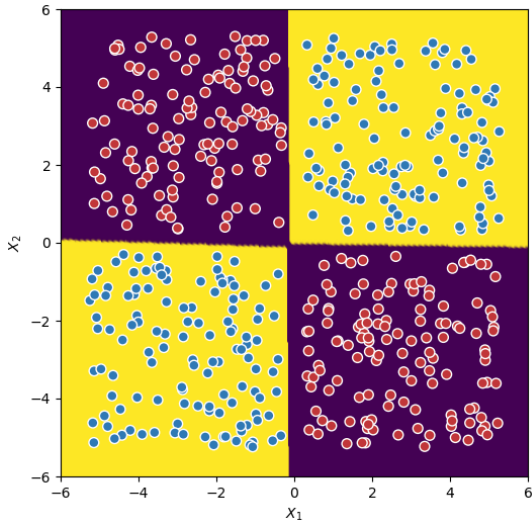
Birinci gizli katmanında $r_1 = 8$ ve ikinci gizli katmanında sırasıyla $r_2 = 1,2,3,4,5,6,7,8,9,10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.8’de verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.8. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
8	1	%50	0.693139	X
8	2	%100	0.000910	√
8	3	%100	0.000866	√
8	4	%100	0.000865	√
8	5	%100	0.000865	√

8	6	%100	0.000865	√
8	7	%100	0.000841	√
8	8	%100	0.000840	√
8	9	%100	0.000838	√
8	10	%100	0.000100	√

Tablo 7.8’de görüldüğü gibi $r_1 = 8$ ve $r_2 = 1$ nöronu olan mimari lokal minimuma takılmış ve $r_1 = 8$ ve sırasıyla $r_2 = 2,3,4,5,6,7,8,9,10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Tablo 7.8 için en az hatayı $r_1 = 8$ ve sırasıyla $r_2 = 10$ nöronu olan mimari vermiştir. Verilen eğitim kümesini $r_1 = 8$ ve $r_2 = 10$ nöronu olan mimari Şekil 7.8’deki gibi sınıflandırmıştır.



Şekil 7.8. $r_1 = 8$ ve $r_2 = 10$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Sonuç olarak burada $r_1 = 8$ ve $r_2 = 1$ nöronu olan iki gizli katmanlı yapay sinir ağı eğitim kümesini sınıflandıramazken, $r_1 = 7$ ve sırasıyla $r_2 = 2,3,4,5,6,7,8,9,10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Burada bölüm 7.6’da bulunan hata değeri 0.000010 azalarak 0.000100 değerini almıştır. O halde ideal nöron sayıları $r_1 = 8$ ve $r_2 = 10$ ’dur denilebilir.

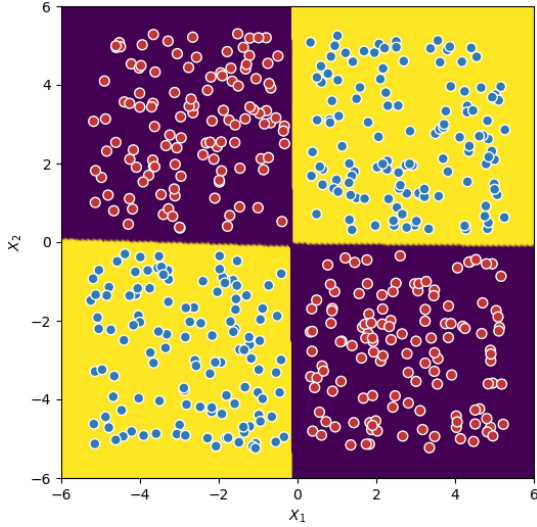
7.9. Birinci Gizli Katmanında $r_1 = 9$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

Birinci gizli katmanında $r_1 = 9$ ve ikinci gizli katmanında sırasıyla $r_2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.9’da verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.9. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
9	1	%50	0.693139	X
9	2	%100	0.000874	√
9	3	%100	0.000854	√
9	4	%100	0.000847	√
9	5	%100	0.000842	√
9	6	%100	0.000843	√
9	7	%100	0.000843	√
9	8	%100	0.000843	√
9	9	%100	0.000106	√
9	10	%100	0.000107	√

Tablo 7.9’da görüldüğü gibi $r_1 = 9$ ve $r_2 = 1$ nöronu olan mimari lokal minimuma takılmış ve $r_1 = 9$ ve sırasıyla $r_2 = 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Tablo 7.9 için en az hatayı $r_1 = 9$ ve sırasıyla $r_2 = 9$ nöronu olan mimari vermiştir. Verilen eğitim kümesini $r_1 = 9$ ve $r_2 = 9$ nöronu olan mimari Şekil 7.9’daki gibi sınıflandırmıştır.



Şekil 7.9. $r_1 = 9$ ve $r_2 = 9$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Sonuç olarak burada $r_1 = 9$ ve $r_2 = 1$ nöronu olan iki gizli katmanlı yapay sinir ağı eğitim kümesini sınıflandıramazken, $r_1 = 7$ ve sırasıyla $r_2 = 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler eğitim kümesini sınıflandırmıştır. Burada bölüm 7.8’de bulunan hata değeri 0.000007 artarak 0.000107 değerini almıştır. O halde bu bölümdeki nöron sayıları yerine bölüm 7.8’deki ideal nöron sayıları tercih edilir.

7.10. Birinci Gizli Katmanında $r_1 = 10$ ve İkinci Gizli Katmanında Sırasıyla $r_2 = 1, 2, \dots, 10$ Nöron Olan Yapay Sinir Ağı Mimarileri İle XOR Probleminin Sınıflandırılması

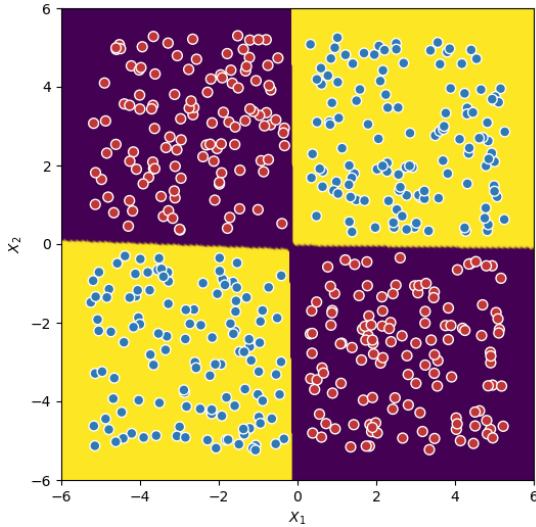
Birinci gizli katmanında $r_1 = 10$ ve ikinci gizli katmanında sırasıyla $r_2 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ nöronu olan mimariler için başarı, hata değerleri ve sınıflandırmaya dayalı bulunan sonuçlar Tablo 7.10’da verilmiştir. Sınıflandırıldığında “√” sembolü, sınıflandırılmadığında “X” sembolü kullanılmıştır.

Tablo 7.10. Verilen Mimariler İçin İki Gizli Katmanlı Yapay Sinir Ağı Sonuçları

r_1	r_2	Başarı	Hata	Sınıflandırma
10	1	%50	0.693139	X
10	2	%50	0.693139	X
10	3	%100	0.001011	√
10	4	%100	0.001011	√

10	5	%100	0.001011	√
10	6	%100	0.000986	√
10	7	%100	0.000954	√
10	8	%100	0.000103	√
10	9	%100	0.000105	√
10	10	%100	0.000105	√

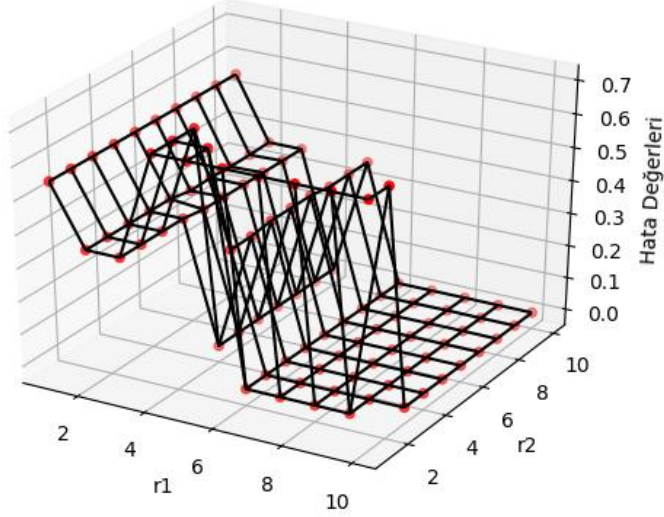
Tablo 7.10’da görüldüğü gibi $r_1 = 10$ ve $r_2 = 1,2$ nöronu olan mimariler lokal minimuma takılmış ve $r_1 = 10$ ve sırasıyla $r_2 = 3,4,5,6,7,8,9,10$ nöronu olan mimariler eğitim kümesini sınıflandırmışlardır. Tablo 7.10 için en az hatayı $r_1 = 10$ ve sırasıyla $r_2 = 8$ nöronu olan mimari vermiştir. Verilen eğitim kümesini $r_1 = 10$ ve $r_2 = 8$ nöronu olan mimari Şekil 7.10’daki gibi sınıflandırmıştır.



Şekil 7.10. $r_1 = 10$ ve $r_2 = 8$ Nöronu Olan Mimarinin Yaptığı Sınıflandırma

Burada $r_1 = 10$ ve $r_2 = 1,2$ nöronu olan iki gizli katmanlı yapay sinir ağları eğitim kümesini sınıflandıramazken, $r_1 = 10$ ve sırasıyla $r_2 = 3,4,5,6,7,8,9,10$ nöronu olan mimariler eğitim kümesini sınıflandırmışlardır. Burada bölüm 7.8’de bulunan hata değeri 0.000003 artarak 0.000103 değerini almıştır. O halde bu bölümdeki nöron sayıları yerine bölüm 7.8’deki ideal nöron sayıları tercih edilir.

Bölüm 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9 ve 7.10'da elde edilen sonuçlar Şekil 7.11'de verilmiştir. Burada r_1 ve r_2 nöron sayıları olmak üzere bunlara karşılık gelen hata değerleri verilmiştir.



Şekil 7.11. r_1 ve r_2 Nöronları İçin Hata Değeri Grafiği

XOR problemini sınıflandıran iki gizli katmalı yapay sinir ağının en az hatayı veren mimariye göre ideal nöron sayısı yapılan gözlemler sonucunda $r_1 = 8$ ve $r_2 = 10$ 'dur denilebilir.

8. İKİ GİZLİ KATMANLI YAPAY SİNİR AĞLARI İÇİN BASKI MALİYET FONKSİYONU

İki gizli katmanlı yapay sinir ağının gizli katmanındaki nöronlara bağlı parametrelerin değerleri sıfıra yakın olan nöronlar gereksizdir. Çünkü bunların algoritmaya bir katkısı olmayacaktır. Gizli katmandaki nöronlar arttırdıkça gereksiz olan nöronlara bağlı parametrelerin sıfıra gideceği hipotezi ortaya atılmıştır. Buradan yola çıkılarak amacı sonuncu nöronun parametrelerini sıfıra götüren bir maliyet fonksiyonu tanımlanmıştır ve **baskı maliyet fonksiyonu** olarak adlandırılmıştır.

Baskı maliyet fonksiyonu iki gizli katmanlı yapay sinir ağlarında gizli katmanlardaki nöron sayıları eşit ($r_1 = r_2$) olan mimariler için tanımlanmıştır. Burada baskı maliyet fonksiyonu katmanlardaki nöron sayısı eşit olduğundan birinci gizli katmanın sonuncu nörona bağlı olan parametreleri ve ikinci gizli katmanın sonuncu nöronuna bağlı olan parametreleri sıfıra götürmektedir. İki gizli katmanlı yapay sinir ağının nöron sayıları $r^s = r_1^s = r_2^s$ ve son nöronlara bağlı olan parametreler w^s olsun. Bu sonuncu nöronlardaki parametrelerin karelerinin toplamı olacak şekilde

$$\Omega(w^s) = \left[\sum_{i=1}^n (w_{r^s,i}^{[1]})^2 \right] + (b_{r^s,1}^{[1]})^2 + \left[\sum_{i=1}^{r^s} (w_{r^s,i}^{[2]})^2 \right] + (b_{r^s,1}^{[2]})^2 + (w_{1,r^s}^{[3]})^2$$

fonksiyonu tanımlansın. O halde yeni maliyet fonksiyonu bir $\gamma \geq 0$ için

$$J_{baski}(w, b) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log a^{[2]}(x^{(i)}) + (1 - y^{(i)}) \log(1 - a^{[2]}(x^{(i)}))] + \gamma \Omega(w^s)$$

şeklinde tanımlanır.

Burada $r_1 = r_2$ ideal nöron sayısı ve ε istenilen maliyet değeri olduğu varsayılırsa $r_1 = r_2$ için hesaplanılan maliyet değeri ile $r_1 = r_2$ için hesaplanılan baskı maliyet değeri arasındaki ilişki

$$J(W, b) \leq J_{baski}(W, b) \leq \varepsilon$$

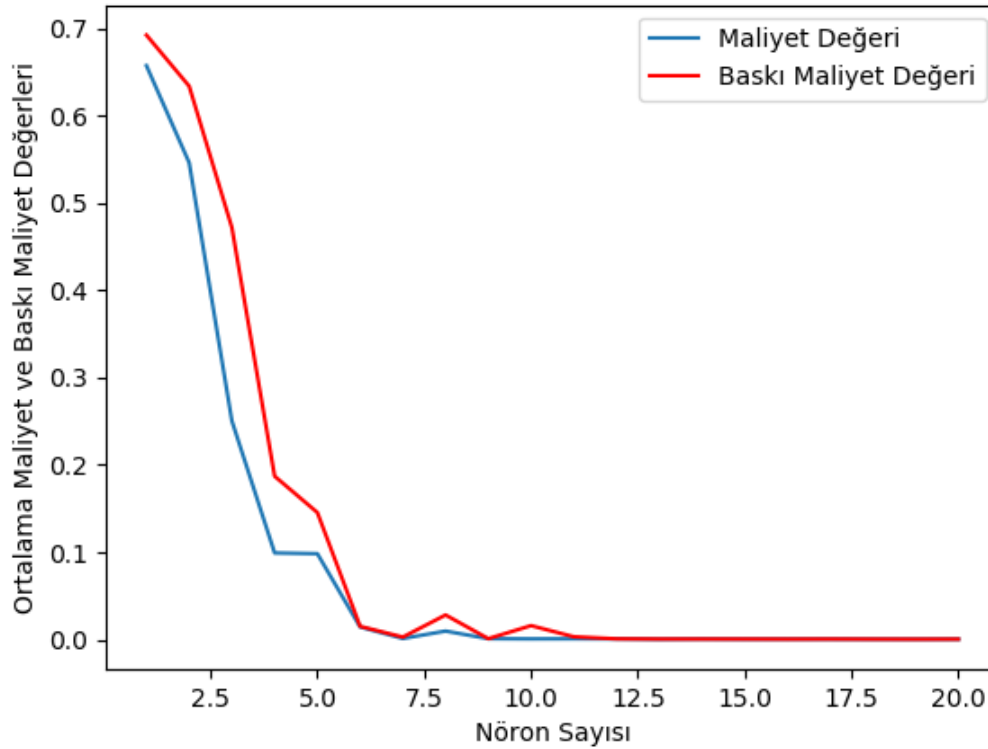
olduğunu gösterelim. Yukarıda $J_{baski}(W, b)$ fonksiyonunun tanımından ve $\gamma \geq 0$, $\Omega(w^s) \geq 0$ olduğundan sonuç görülür.

Baskı maliyet fonksiyonu $J_{baski}(W, b)$ tanımındaki γ parametresinin rolü $\Omega(w^s)$ değerini ve buna özdeş olarak $w_{r^s,i}^{[1]}$, $b_{r^s,1}^{[1]}$, $w_{r^s,i}^{[2]}$, $b_{r^s,1}^{[2]}$, $w_{1,r^s}^{[3]}$ parametrelerini kontrol etmektir.

Bu parametreleri sıfıra götürmek için uygun bir γ değeri seçilebilir. Ancak bir gizli katmanda tanımlanan baskı maliyetteki gibi r ile $r+1$ nöronlarının karşılaştırılması zordur. Bu durum tablo 9.1’de de görülmektedir.

9. İKİ GİZLİ KATMANLI YAPAY SİNİR AĞLARI İÇİN BİR DENEY

Deney için 500 tane veri içeren 100 tane eğitim kümesi kullanılmıştır. Deneyde epoch sayısı 2500 ve adım büyüklüğü $\alpha = 0.3$ olarak atanmıştır. Bir eğitim kümesi sırasıyla $r_1 = r_2 = 1, r_1 = r_2 = 2, \dots, r_1 = r_2 = 20$ nöronlu modeller ile her bir model 20 kez farklı başlangıç parametreleriyle çalıştırılarak $r_1 = r_2 = 1, r_1 = r_2 = 2, \dots, r_1 = r_2 = 20$ nöron için ayrı ayrı ortalama maliyet değerleri hesaplanmıştır. Yani $r_1 = r_2 = 1$ için ortalama bir maliyet değeri, $r_1 = r_2 = 2$ için ayrı ve bu şekilde $r_1 = r_2 = 20$ ' ye kadar ortalama maliyet değerleri hesaplanmıştır. Daha sonra 100 farklı eğitim kümesi için bu maliyet değerleri hesaplanmış ve bu değerlerinde nöron sayıları göz önüne alınarak $r_1 = r_2 = 1$ nöronlu maliyet değerlerinin ortalaması, $r_1 = r_2 = 2$ nöronlu maliyet değerlerinin ortalaması ve böyle devam ederek $r_1 = r_2 = 20$ nöronlu maliyet değerlerinin ortalaması hesaplanmıştır. Sonuç olarak $r_1 = r_2 = 1, r_1 = r_2 = 2, \dots, r_1 = r_2 = 20$ nöron için bir maliyet değeri elde edilmiştir. Aynı deney için baskı maliyet fonksiyonu kullanılarak bulunan ideal nöron sayısı doğrulanmıştır. Şekil 8.1'de nöron sayılarına karşılık gelen ortalama maliyet değerlerini ve baskı maliyet değerini veren grafik verilmiştir.



Şekil 9.1.Deney Sonucu

Şekil 8.1’den görülmektedir ki $r_1 = r_2 = 11$ nörondan sonra maliyet değerinde büyük bir değişim olmamıştır. O halde $r_1 = r_2 = 11$ nöron iki gizli katmanlı yapay sinir ağı için ideal nöron denilebilir.

Tablo 9.1. Maliyet ve Baskı Maliyet Değerleri

Nöron Sayısı	Maliyet Değeri	Baskı Maliyet Değeri	Fark
$r_1 = r_2 = 1$	0.6571437520254799	0.6921783806048412	0.0350346285793
$r_1 = r_2 = 2$	0.5460772888842271	0.6336688514250568	0.0875915625408
$r_1 = r_2 = 3$	0.2503264900397174	0.4717417671655557	0.2214152771258
$r_1 = r_2 = 4$	0.09916726445384653	0.18703520595199669	0.0878679414981
$r_1 = r_2 = 5$	0.09832263716626803	0.14544962705860756	0.0471269898923
$r_1 = r_2 = 6$	0.014416900404004476	0.015114180909969589	0.0006972805059
$r_1 = r_2 = 7$	0.0011796841230689076	0.0028083275185933703	0.0016286433955
$r_1 = r_2 = 8$	0.009704281459293957	0.028141407249532867	0.0184371257902
$r_1 = r_2 = 9$	0.000982338968991349	0.0010119994551743663	0.0000296604861
$r_1 = r_2 = 10$	0.0007675677261132074	0.01592326993233654	0.0151557022062
$r_1 = r_2 = 11$	0.0008465513095678639	0.003186536064339478	0.0023399847547
$r_1 = r_2 = 12$	0.0007611227458294063	0.0007921595153843804	0.0000310367695
$r_1 = r_2 = 13$	0.00036778492132222225	0.0005610165270301889	0.0001932316057
$r_1 = r_2 = 14$	0.0005740966079886477	0.00054053897844377	-0.000033557629
$r_1 = r_2 = 15$	0.0004211911145163297	0.0005289419621051684	0.0001077508475
$r_1 = r_2 = 16$	0.0004906880680247903	0.0004442489468152607	-0.000046439121
$r_1 = r_2 = 17$	0.00043887048627899245	0.0005235974396048367	0.0000847269533
$r_1 = r_2 = 18$	0.0003749666453235878	0.00042492508254378885	0.0000499584372
$r_1 = r_2 = 19$	0.0003588777790087233	0.00036152570210513935	0.0000026479230
$r_1 = r_2 = 20$	0.0003808417647036046	0.00043811992914027096	0.0000572781644

Sonuç olarak en az hatayı veren mimariye göre ideal nöron sayısı $r_1 = r_2 = 19$ olarak belirlenebilir.

10. SONUÇ

Bu tez çalışmasında bir ve iki gizli katmanlı yapay sinir ağları için optimal nöron sayısı araştırıldı. Bu algoritmalar için XOR veri kümesi kullanıldı. Bu veri kümesi için başlangıç parametreleri sabit tutulup, adım büyüklüğü $\alpha = 0,1$ ve epoch = 1000 olmak üzere bir ve iki gizli katmanlı sinir ağının gizli katmanındaki ideal nöron sayısını bulmak adına gözlemler yapıldı. Ardından bir ve iki gizli katmanlı yapay sinir ağları için yeni maliyet fonksiyonları tanımlandı. Bu fonksiyonlar, gizli katmanlardaki sonuncu nöronların parametrelerini sıfır yapacak şekilde tasarlanmıştır. Başka bir deyişle gizli katmandaki sonuncu nöronlar baskılanmıştır. Böylece r ideal nöron sayısı ve ϵ istenilen maliyet değeri olduğu varsayılırsa r için hesaplanan maliyet değeri, r için hesaplanan baskı maliyet değeri arasındaki fark çok küçük ve aralarındaki ilişki $J(W, b) \leq J_{baski}(W, b) \leq \epsilon$ olduğu ispatlanmıştır. Buradan yola çıkarak bir ve iki gizli katmanlı yapay sinir ağları için bir deney yapılmıştır. Bu deney için veri kümesine ve başlangıç parametrelerine olan bağımlılığını azaltmak adına 100 tane her biri 500 nokta içeren XOR veri kümesi alınmış ve her veri kümesi 1, 2, ..., 20 nöron için 20 kez farklı başlangıç parametreleriyle ile $\alpha = 0,1$ ve epoch= 1000 olmak üzere çalıştırılarak ortalama maliyet değerleri hesaplanmıştır. Aynı deney için ortalama baskı maliyet değerleri hesaplanmıştır. Burada bulunan ortalama maliyet değerleri ve ortalama baskı maliyet değerleri göz önünde bulundurularak XOR veri kümesi için bir ve iki gizli katmanlı yapay sinir ağlarının ideal nöron sayıları tespit edilmiştir. Yapay sinir ağları birçok katmandan oluştuğu gibi farklı mimarilerde vardır(CNN, RNN v.b). Benzer bir çalışma kompleks mimarilere de uygulanabilir. Bunun bir örneği olarak Ariel ve ark. (2018) tarafından yürütülen çalışmada katmanların parametrelendiği bir inceleme yapılmıştır. Bizim sonuçlarımızla bahsedilen çalışmanın sonuçları birleştirildiğinde yapay sinir ağlarının ideal yapısı için önemli sonuçlar elde edileceği umulmaktadır.

11. KAYNAKÇA

- Andrew NG, Jiquan N, Chuan Y, Yifan M, Caroline S, Adam C, Andrew M, Awni H, Brody H, Tao W, Sameep T (2014). Stanford Unsupervised Deep Learning Tutorial, <http://ufldl.stanford.edu/tutorial/> (erişim tarihi, 30.05.2019).
- Andrew NG (2011). Andrew NG's notes. Stanford Machine Learning, <http://www.holehouse.org/mlclass/index.html> (erişim tarihi, 30.05.2019).
- Andrew NG (2013). Stanford Üniversitesi CS229 - Machine Learning Course. Stanford Üniversitesi İnternet sitesi, <https://see.stanford.edu/Course/CS229/54> (erişim tarihi, 30.05.2019).
- Andrew NG (2017). Neural Networks and Deep Learning Course. Coursera, <https://www.coursera.org/learn/neural-networks-deep-learning> (erişim tarihi, 30.05.2019).
- Anonim (2018). Deep Learning Türkiye. <https://medium.com/deep-learning-turkiye/şu-kara-kutuyu-açalım-yapay-sinir-ağları-7b65c6a5264a>(erişim tarihi, 30.05.2019).
- Apaydın E (2017). Introduction To Machine Learning, Third Education. Boğaziçi Üniversitesi Yayınevi, 486s, İstanbul.
- Ariel G, Elad E, Ofir N, Bo, HaoW, Tien-JuY, Edward C (2018). MorphNet: Fast& Simple Resource-Constrained Structure Learning of Deep Networks. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). DOI :10.1109/cvpr.2018.00171 . 1586-1595.
- Ari M, David B, Neil R, Matthew B (2018). Understanding deeplearning through neuron deletion DeepMind. Neural Information Processing Systems (NeurIPS), <https://arxiv.org/pdf/1806.05759.pdf> (erişim tarihi, 30.05.2019).
- Catherine H, Desmond H (2018). Deep Learning: An Introduction for Applied Mathematicians. Arxiv, <https://arxiv.org/pdf/1801.05894.pdf>(erişim tarihi, 30.05.2019).
- Cybenko G (1989). Approximation by superpositions of a sigmoidal function. Mathematics of Control Signals and Systems, <https://pdfs.semanticscholar.org/05ce/b32839c26c8d2cb38d5529cf7720a68c3fab.pdf> (erişim tarihi, 30.05.2019).
- Elmas Ç (2003). Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama). Seçkin Yayıncılık, 192s, Ankara .
- Elmas Ç (2011). Yapay Zekâ Uygulamaları. Seçkin yayıncılık, 424s, Ankara.
- Goodfellow I, Bengio Y, Courville A(2016). Deep Learning. The MIT Press, 800s, United States.
- Haefele B and Ren'eVidal (2017). Global Optimality in Neural Network Training. IEEE Conference on Computer Vision and Pattern, http://openaccess.thecvf.com/content_cvpr_2017/papers/Haefele_Global_Optimality_in_CVPR_2017_paper.pdf(erişim tarihi, 30.05.2019).
- Heaton J (2008). IntroductiontoNeural Networks for C#, Second Edition. Heaton Research, 428s, United States.
- Johnson J (2018). Deep, SkinnyNeural Networks are not Universal Approximators. Arxiv, <https://arxiv.org/pdf/1810.00393.pdf>(erişim tarihi, 30.05.2019).

- Kawaguchi K, Kaelbling L(2019). Elimination of All Bad Local Minima in Deep Learning. Arxiv, <https://arxiv.org/pdf/1901.00279.pdf> (erişim tarihi, 30.05.2019).
- Kurt F (2018). Evrişimli Sinir Ağlarında Hiper Parametrelerin Etkisinin İncelenmesi. Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Malach E, Shalev-Shwartz S (2018). A Provably Correct Algorithm for Deep Learning that Actually Work. Arxiv, <https://arxiv.org/pdf/1803.09522.pdf>(erişim tarihi, 30.05.2019).
- Öztemel, E. (2016). Yapay Sinir Ağları. Papatya Yayıncılık Eğitim, 232s, İstanbul.
- Smilkov D, Carter S (2018).Tensorflow Playground. TensorFlow, <http://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle®Dataset=reg-plane&learningRate=0.03®ularizationRate=0&noise=0&networkShape=4,2&seed=0.89799&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false>(erişim tarihi, 30.05.2019)
- Şeker S.(2008). <http://bilgisayarkavramlari.sadievrenseker.com/2008/11/03/gizli-katmandaki-noron-sayisi/>(erişim tarihi, 30.05.2019).
- Vidal R, Joan B, Raja G, Stefano S (2017). Mathematics of Deep Learning. Arxiv, <https://arxiv.org/pdf/1712.04741.pdf> (erişim tarihi, 30.05.2019).