



T.C.
HATAY MUSTAFA KEMAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK KATMANLI YAPAY SİNİR AĞLARINDA KULLANILAN
MATEMATİKSEL MODELLER ÜZERİNE BİR ARAŞTIRMA

Enis İGDE

ENFORMATİK ANABİLİM DALI

YÜKSEK LİSANS TEZİ

HATAY
OCAK - 2023



T.C.
HATAY MUSTAFA KEMAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK KATMANLI YAPAY SİNİR AĞLARINDA KULLANILAN
MATEMATİKSEL MODELLER ÜZERİNE BİR ARAŞTIRMA

Enis İGDE
ORCID: 0000-0001-9562-7014

ENFORMATİK ANABİLİM DALI

YÜKSEK LİSANS TEZİ

Danışman
Dr. Öğretim Üyesi Oğuz KILIÇOĞLU
ORCID: 0000-0001-7358-8034

HATAY
OCAK - 2023

T.C.
HATAY MUSTAFA KEMAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ÇOK KATMANLI YAPAY SİNİR AĞLARINDA KULLANILAN
MATEMATİKSEL MODELLER ÜZERİNE BİR ARAŞTIRMA**

Enis İGDE

ENFORMATİK ANABİLİM DALI

YÜKSEK LİSANS TEZİ

Dr. Öğretim Üyesi Oğuz KILIÇOĞLU danışmanlığında hazırlanan bu tez 20/01/2023 tarihinde aşağıdaki jüri üyeleri tarafından **OYBİRLİĞİ** ile kabul edilmiştir.

Dr. Öğretim Üyesi Oğuz KILIÇOĞLU
Başkan

Dr. Öğretim Üyesi Mücahid GÜNAY
Üye

Dr. Öğretim Üyesi Fatih BALAMAN
Üye

Kod No:

Doç. Dr. Cengiz KARACA
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

20/01/2023

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını ve tez üzerinde Yükseköğretim Kurulu tarafından hiçbir değişiklik yapılamayacağı için tezin bilgisayar ekranında görüntülendiğinde asıl nüsha ile aynı olması sorumluluğunun tarafıma ait olduğunu beyan ederim.

İmzası

Enis İGDE

ÖZET

ÇOK KATMANLI YAPAY SİNİR AĞLARINDA KULLANILAN MATEMATİKSEL MODELLER ÜZERİNE BİR ARAŞTIRMA

Çok katmanlı yapay sinir ağları sınıflandırma ve regresyon problemleri için kullanılan ve günümüzde yaygın olarak kabul görmüş bir yapay zeka teknolojisidir. Bu ağlar genel itibariyle başarılı sonuçlar vermekle birlikte derin öğrenme uygulamalarının temelini oluşturmaktadır. Bu tez Yapay Sinir Ağlarında çok katmanlı sinir ağlarının matematiksel anlamda incelenmesi ve çok katmanlı modellerde aktivasyon modellerinin incelenmesi hakkındadır. Bu tez çalışmasında Yapay sinir ağları ve tipleri hakkında literatür taraması yapıp derin öğrenme ve makine öğrenmesi hakkında bilgi verilmiştir. Çalışmanın yöntem ve materyal bölümünde en sık kullanılan aktivasyon fonksiyonlarından Sigmoid ve ReLU fonksiyonları Python programı Pycharm IDE'sinde modeller kullanılarak karşılaştırılmıştır. Her iki yöntem birbiriyle kıyaslanmış olup performansları, avantajları ve dezavantajları ile ilgili problemi üzerinde değerlendirilmiştir. Kullanılan veri seti, Kaggle web sitesinde açık kaynak olarak bulunmaktadır. Massachusetts, Framingham kasabasında yaşayan bireylerde devam eden bir kardiyovasküler çalışma bilgilerini içermektedir. Veri seti içerisinde toplam 4238 kişinin 15 farklı özellik bilgileri yer almaktadır. Sonuç olarak; yapay sinir ağı ara katmanlarda önce Sigmoid, daha sonra ReLU fonksiyonu kullanılarak aynı veri seti ile eğitilmiştir. Eğitim sonucunda elde edilen tahmin sonuçlarına göre ReLU fonksiyonu daha başarılı bulunmuş olup, tez kapsamında yapılan araştırma ve çalışmaların sonuçları paylaşılırken önerilerde bulunulmuştur.

2023, 49 sayfa

Anahtar Kelimeler: Çok Katmanlı Yapay Sinir Ağı, Sigmoid, ReLU, Matematiksel Model, Aktivasyon Fonksiyonları

ABSTRACT

A RESEARCH on MATHEMATICAL MODELS USED in MULTILAYER ARTIFICIAL NEURAL NETWORKS

Multi-layered artificial neural networks are widely accepted artificial intelligence technology used for classification and regression problems. These networks generally provide successful results and form the basis of deep learning applications. This thesis is focused on the mathematical analysis of multi-layered neural networks and the examination of activation models in multi-layered models. In this thesis, a literature review was conducted on artificial neural networks and their types, and information was provided on deep learning and machine learning. The method and material section of the study compared the Sigmoid and ReLU functions, which are the most commonly used activation functions, using models in the Python program Pycharm IDE. Both methods were compared and evaluated in terms of their performance, advantages, and disadvantages. The dataset used in the study is available as an open source on the Kaggle website and includes information from a cardiovascular study of individuals living in the town of Framingham, Massachusetts, with a total of 4238 people and 15 different characteristic features. As a results, artificial plexus has been trained by using first Sigmoid in middile layers, and after that ReLU function with the same set of data. According to the results of the guesses acquired at the and of the training, ReLU function was found more successful, and whilst sharing the results of the studies and research conducted in the scope of this thesis, recommendations were made.

2023, 49 pages

Key Words: Multi-layered Artificial Neural Network, Sigmoid, ReLU, Mathematical Model, Activation Functions.

TEŞEKKÜR

Çalışmalarım boyunca farklı bakış açıları ve bilimsel katkılarıyla beni aydınlatan, yakın ilgi ve yardımlarını esirgemeyen ve bu günlere gelmemde en büyük katkı sahibi sayın hocam Dr. Öğr. Üyesi Oğuz KILIÇOĞLU'na teşekkürü bir borç bilirim. Tez çalışmam sırasında karşılaştığım zorlukları aşmamda yardımlarından ve kıymetli desteklerinden dolayı değerli arkadaşım Kaan ÇUKUR'a ve Yüksek Mühendis Mehmet Mükerrerem RENÇBER'e teşekkür ederim. Ayrıca; çalışmalarım süresince sabır göstererek beni daima destekleyen babam Naim İGDE'ye, annem Naciye İGDE'ye, ablam Hülya SÖNMEZ ve son olarak abim Ali İGDE'ye en içten teşekkürlerimi sunarım.



İÇİNDEKİLER

ÖZET	I
ABSTRACT	II
TEŞEKKÜR	III
İÇİNDEKİLER	IV
ÇİZELGELER DİZİNİ	VI
ŞEKİLLER DİZİNİ	VII
SİMGELER ve KISALTMALAR DİZİNİ	VIII
1. GİRİŞ	1
1.1. Makine Öğrenme ve Derin Öğrenmeye Genel Bir Bakış	1
1.1.1. Denetimli Öğrenme	3
1.1.2. Denetimsiz Öğrenme	3
1.1.3. Destekleyici Öğrenme	3
1.2. Yapay Sinir Ağları	4
1.2.1. Yapılarına Göre Yapay Sinir Ağları	4
1.2.1.1. İleri Beslemeli Yapay Sinir Ağları	5
1.2.1.1.1. Tek Katmanlı İleri Beslemeli YSA (SLFFNN)	5
1.2.1.1.2. Çok Katmanlı İleri Beslemeli YSA (MLFFNN)	6
1.2.1.1.3. Tekrarlayan Yapay Sinir Ağları (RNN)	7
1.2.1.2. Geri Beslemeli Yapay Sinir Ağları	9
1.2.1.2.1. Denetimli Yapay Sinir Ağları	10
1.2.1.2.2. Denetimsiz Geri Beslemeli YSA	10
1.2.1.2.3. Destekleyici Geri Beslemeli YSA	11
1.3. Çok Katmanlı Yapay Sinir Ağlarında Matematiksel Modeller	11
1.3.1. Hebb	11
1.3.2. Hopfield	11
1.3.3. Delta	12
1.3.4. Dereceli Azaltma	12
1.3.5. Kohonen	13
1.4. Aktivasyon Kodları	14
1.4.1. Aktivasyon Fonksiyonları	14
1.4.1.1. Basamak (Step) Fonksiyonu	15
1.4.1.2. Doğrusal (Linear) Fonksiyon	15
1.4.1.3. Sigmoid Fonksiyonu	16
1.4.1.4. Softmax Fonksiyonu	18
1.4.1.5. Hiperbolik Tanjant Fonksiyonu	18
1.4.1.6. ReLU (Rectified Linear Unit) Fonksiyonu	19
1.4.1.7. Sızıntı (Leaky) ReLU Fonksiyonu	20
1.4.1.8. Swish (A Self-Gated/Kendinden Geçitli) Fonksiyonu	20
1.4.1.9. Swish Fonksiyonu	21
2. ÖNCEKİ ÇALIŞMALAR	26
3. MATERYAL ve YÖNTEM	28
4. ARAŞTIRMA BULGULARI ve TARTIŞMA	31
4.1. Sigmoid Fonksiyonu Üzerine Bir Uygulama	31
4.2. ReLU Fonksiyonu Üzerine Bir Uygulama	36
5. SONUÇ ve ÖNERİLER	43

KAYNAKLAR	45
ÖZGEÇMİŞ	49



ÇİZELGELER DİZİNİ

Çizelge 1.1. RNN'nin Temel Avantaj ve Dezavantajları.....	9
Çizelge 1.2. Aktivasyon Fonksiyonlarının Matematiksel İfadeleri	22
Çizelge 3.1. MLFFNN Avantaj ve Dezavantajları	29



ŞEKİLLER DİZİNİ

Şekil 1.1. Tek Katmanlı YSA Modeli (Öztemel, 2006).....	5
Şekil 1.2. Çok Katmanlı Yapay Sinir Ağı Modeli (Kızrak, 2018).....	7
Şekil 1.3. Gizli nöronlarla Tekrarlayan Ağ Yapısı (RNN) (Haykin, 1998).....	8
Şekil 1.4. İleri ve Geri Beslemeli Ağ Yapısı (Karakaya,2012).....	10
Şekil 1.5. Hata Değişim Eğrisi (Akça, 2020).....	13
Şekil 1.6. Basamak (Step) Fonksiyonu (Kızrak, 2019).....	15
Şekil 1.7. Doğrusal (Linear) Fonksiyon (Kızrak, 2019)	16
Şekil 1.8. Sigmoid Fonksiyonu (Kızrak,2019).....	17
Şekil 1.9. Hiperbolik Tanjant Fonksiyonu (Kızrak, 2019).....	18
Şekil 1.10. ReLU (Rectified Linear Unit) Fonksiyonu (Kızrak, 2019)	19
Şekil 1.11. Sızıntı (Leaky) ReLU Fonksiyonu (Kızrak, 2019)	20
Şekil 1.12. Swish (A Self-Gated/Kendinden Geçitli) Fonksiyonu (Kızrak, 2019).....	21
Şekil 1.13. Örnek sınıflar için temsil verileri (GitHub, 2013)	23
Şekil 1.14. Örnek yitim hesabı gösterimi (GitHub, 2013)	25
Şekil 4.1. Üç Katmanlı YSA Modeli Sigmoid Yitim Grafiği.....	32
Şekil 4.2. Üç Katmanlı YSA Modeli Sigmoid ROC/AUC Grafiği	32
Şekil 4.3. Beş Katmanlı YSA Modeli Sigmoid Yitim Grafiği.....	33
Şekil 4.4. Beş Katmanlı YSA Modeli Sigmoid ROC/AUC Grafiği	34
Şekil 4.5. Yedi Katmanlı YSA Modeli Sigmoid Yitim Grafiği.....	35
Şekil 4.6. Yedi Katmanlı YSA Modeli Sigmoid ROC/AUC Grafiği	36
Şekil 4.7. Üç Katmanlı YSA Modeli ReLU Yitim Grafiği.....	37
Şekil 4.8. Üç Katmanlı YSA Modeli ReLU ROC/AUC Grafiği	38
Şekil 4.9. Beş Katmanlı YSA Modeli ReLU Yitim Grafiği	39
Şekil 4.10. Beş Katmanlı YSA Modeli ReLU ROC/AUC Grafiği.....	40
Şekil 4.11. Yedi Katmanlı YSA Modeli ReLU Yitim Grafiği.....	41
Şekil 4.12. Yedi Katmanlı YSA Modeli ReLU ROC/AUC Grafiği	42

SİMGELER ve KISALTMALAR DİZİNİ

KISALTMALAR

YSA	: Yapay Sinir Ağları
SLFFNN	: Tek Katmanlı Yapay Sinir Ağı
MLFFNN	: Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı
RNN	: Tekrarlayan Ağ Yapısı
XOR	: eXclusive OR Kapısı
SVM	: Destek Vektör Makineleri
ReLU	: Düzeltilmiş Doğrusal Birim
RBF	: Radyal Temelli Fonksiyon



1. GİRİŞ

Bu bölümde yapa zekâ, makine öğrenmesi ve derin öğrenme gibi kavramlardan bahsedilmiştir. Bunun yanı sıra bu öğrenme çeşitlerinin sınıflandırılmasından bahsedilip literatürdeki tanımları açıklanmıştır.

1.1. Makine Öğrenme ve Derin Öğrenmeye Genel Bir Bakış

Derin öğrenme, çoklu soyutlama düzeyine sahip verilerin temsillerini öğrenmek için birden çok işlem katmanından oluşan yöntemleri, hesaplama modelleri ile sağlar. Bu yöntemler, konuşma tanıma, görsel nesne tanımlama ve belirleme, ilaç keşfi ve genom bilimi gibi diğer birçok alanlarda yüksek teknolojiyi önemli ölçüde geliştirmiştir (Yann LeCun, 2015).

Derin öğrenme, büyük veri kümelerindeki karmaşık yapıları açıklayabilmek için geri yayılım algoritmasını kullanmakta olup bu algoritma bir makinenin önceki katmandaki temsilden her bir katmandaki temsili hesaplamak için kullanılan dâhili parametrelerini nasıl değiştirmesi gerektiğini keşfeder. Derin erişimli ağlar; görüntü, video, konuşma ve sesin işlenmesinde çığır açmasıyla birlikte yinelenen ağlar, metin ve konuşma gibi sıralı verilere de ışık tutmaktadır (Tompson, 2014).

Makine öğrenimi teknolojisi, web aramalarından sosyal ağlara, içerik filtrelemeden, e-ticarete, birçok alana yön vermekle kalmayıp kameralar ve akıllı telefonlar gibi tüketici ürünlerinde giderek daha fazla yer almaktadır. Geleneksel makine öğrenimi teknikleri, doğal verileri ham hallerinde işlemesi bakımından oldukça sınırlıdır. Ayrıca bir örüntü tanıma veya makine öğrenimi sistemi oluşturmak veya bir özellik çıkarıcı tasarlamak için, iyi bir mühendislik ve önemli alan uzmanlığı gerektirmektedir (Farabet, 2013).

Temsil öğrenimi, bir makinenin ham verilerle beslenmesine ve algılama veya sınıflandırma için gerekli temsilleri otomatik olarak keşfetmesine izin veren bir dizi yöntemdir.

Derin öğrenme yöntemleri; her biri bir düzeydeki (ham girdiden başlayarak) temsili daha yüksek, biraz daha soyut bir düzeydeki bir temsile dönüştüren, basit ancak doğrusal olmayan modüller oluşturarak elde edilen, birden çok temsil düzeyine sahip temsilleri

öğrenme yöntemleridir. Bu tür dönüşümlerin bileşimi ile çok karmaşık fonksiyonlar bile öğrenilebilir. Sınıflandırma görevleri için, daha yüksek temsil katmanları, girdinin ayırım için önemli olan yönlerini genişletir ve ilgisiz varyasyonları eler. Örneğin bir görüntü, bir piksel değerleri dizisi biçiminde gelir ve birinci temsil katmanındaki öğrenilen özellikler tipik olarak görüntüdeki belirli yönlerde ve konumlarda kenarların varlığını veya yokluğunu temsil eder. İkinci katman, kenar konumlarındaki küçük değişikliklerden bağımsız olarak, kenarların belirli düzenlemelerini tespit ederek motifleri tipik olarak tespit eder. Üçüncü katman, motifleri, tanıdık nesnelerin parçalarına karşılık gelen daha büyük kombinasyonlar halinde bir araya getirebilir ve sonraki katmanlar, nesneleri bu parçaların kombinasyonları olarak algılayabilir. Derin öğrenmenin buradaki kilit yönü ise, katmanlarının bu özelliklerinin mühendisler tarafından tasarlanmaması olup genel amaçlı bir öğrenme prosedürü kullanılarak verilerden kendi kendine öğrenmesidir (Krizhevsky, 2012).

Derin öğrenme, uzun yıllardır yapay zekâ topluluğunun en iyi girişimlerinde bile karşısına çıkan problemlerin çözümünde büyük ilerlemeler kaydetmeye devam etmektedir. Derin öğrenme yüksek boyutlu verilerdeki karmaşık yapıları keşfetmede oldukça iyi olduğu için bilim, iş dünyası ve finans ve kamusal birçok alanda uygulanabilir durumdadır.

Derin öğrenme, görüntü tanıma ve konuşma tanıma konularında çığır açmasına ek olarak, potansiyel ilaç moleküllerinin aktivitesini tahmin etme, parçacık hızlandırıcı verilerini analiz etme beyin devrelerini yeniden yapılandırma ve kodlayıcı olmayan DNA'daki mutasyonların gen ekspresyonu ve hastalık üzerindeki etkilerinin tahmin edilmesi gibi alanlarda makine öğrenmesine üstünlük sağlamıştır. Ayrıca derin öğrenme, özellikle konu sınıflandırması, duygu analizi, soru yanıtlama ve dil çevirisi olmak üzere, doğal dili anlama alanındaki çeşitli görevler için son derece önemli sonuçlar üretmiştir.

Derin öğrenme mühendisliğin manuel olarak yaptığı şeyleri çok aza indirgediği için yakın gelecekte çok daha fazla başarıya sahip olacağını düşünülmektedir. Bu sebeple mevcut hesaplama ve veri miktarındaki artışlardan kolayca faydalanılabilecektir.

Günümüzde, derin sinir ağları için geliştirilmekte olan yeni öğrenme algoritmaları ve mimarileri bu ilerlemeyi hızlandıracaktır. (Yann LeCun, 2015)

1.1.1. Denetimli Öğrenme

Derin öğrenme olsun ya da olmasın, makine öğreniminin en yaygın biçimi denetimli öğrenmedir. Görüntüleri ev, araba, insan veya evcil hayvan olarak sınıflandırabilen bir sistem oluşturmak istediğimizi varsayalım. Öncelikle, her biri kendi kategorisine göre etiketlenmiş evlerin, arabaların, insanların ve evcil hayvanların görüntülerinden oluşan geniş bir veri seti toplarız. Eğitim sırasında, makineye bir görüntü gösterilir ve makine her kategori için bir çıktı üretir. YSA eğitildikten sonra makineye gösterilen görüntünün uygun kategoriye göre sınıflandırılması beklenir. Gerçek sonuç ile YSA tahmini arasındaki hatayı ölçen bir amaç fonksiyonu hesaplanır. (Glorot, 2011).

1.1.2. Denetimsiz Öğrenme

Denetimsiz öğrenmede sistem çıktıyı tam olarak bilmez ve girdiye göre kendini örnekler. Ağ, ağırlık değerlerini girdi bilgilerinin özelliklerine göre veya çıktı değerleri ile keyfi olarak ayarlar. Bu tür bir öğrenmede, ara yazılım, dışarıdan yardım almadan kendini organize etmenin bir yolunu bulmalıdır. Kohonen tarafından geliştirilen harita ağlarında da biyolojik sistemlerden yararlanıldığı söylenebilir. Bu yöntemde nöronlar, öğrenme için mevcut durumu veya adımları güncellemek için yarışır. En büyük çıktıya sahip işlenmiş nöron kazananı belirler ve komşularının bağlantı boyutlarını güncellemelerine izin verir (E. Öztemel, 2003).

1.1.3. Destekleyici Öğrenme

Bu stratejide, bir öğretmen öğrenme sürecine yardımcı olur. Ancak sisteme her bir girdi seti için gerekli çıktı setini göstermek yerine, öğretmen sistemin kendisine sunulan girdilere karşılık olarak çıktılar üretmesini bekler ve çıktıların olup olmadığını belirtir. Üretilen doğru veya yanlıştır. Öğretmenden bu sinyali alan sistem, öğrenme sürecini sürdürür (E. Öztemel, 2012).

Yani sonuçlar hakkında sayısal bir bilgi olmasa dahi, bazen sözlü olarak bilgi alınabilmektedir. Örneğin iyi-kötü, güzel-çirkin, evet-hayır, doğru-yanlış gibi bilgiler danışmanın ağı kısmen kontrol etmesine olanak tanır (Sevim, 2012).

Tez kapsamında bölüm 1.2 de yapay sinir ağları irdelenmiş ve yapılarına göre sınıflandırılmış olup detaylıca açıklanmıştır.

1.2. Yapay Sinir Ağları

Yapay sinir ağlarının (YSA) literatürde çeşitli tanımları olup deneyimsel bilgiyi depolamak ve onu kullanıma hazır hale getirmek için doğal bir eğilime sahip basit işlem birimlerinden oluşan büyük ölçüde paralel dağıtılmış işlemci olarak tanımlanır. YSA, doğrudan insan sinir sistemini referans alarak geliştiren bir teknoloji olduğu için, beynin öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri, dışarıdan herhangi bir müdahale olmadan otomatik bir şekilde gerçekleştirebilmek amacı ile tasarlanan bilgisayar sistemleridir şeklinde de tanımlanmıştır (Krizhevsky, 2012).

Yapay sinir ağının genelleme yeteneği ağın topolojinin doğru seçilmesi ile birebir ilişkilidir. Ağ için en uygun mimari, problemi öğrenmek için yeterince büyük, genelleme yapabilmek için ise bir o kadar küçük olmalıdır. En uygun mimariden daha küçük bir ağ problemi iyi öğrenemez diğer taraftan daha büyük bir ağ ise eğitim verisini aşırı öğrenir ki bu da ezberlemesine neden olduğundan genelleme yeteneği zayıf kalır. Ağın yapısının belirlenmesinde temel olarak büyüyen/yapıcı ve budama/yıkıcı olmak üzere iki açgözlü yaklaşım bulunmaktadır. Ağın yapısı küçük seçilip öğrenme sürecinde büyüyor ise büyüyen/yapıcı bir yaklaşım, aksine büyük seçilip öğrenme süresince küçülüyor ise budama/yıkıcı bir yaklaşım izlenmiştir (O. Aran, 2009) Yapay sinir ağları yapılarına katman sayılarına öğrenme metotlarına bağlı olarak çeşitli şekillerde sınıflandırılmışlardır.

Aşağıda bunlar detaylıca incelenmiş olup bu tez çalışmasında çok katmanlı yapay sinir ağı sistemi kullanılacaktır.

1.2.1. Yapılarına Göre Yapay Sinir Ağları

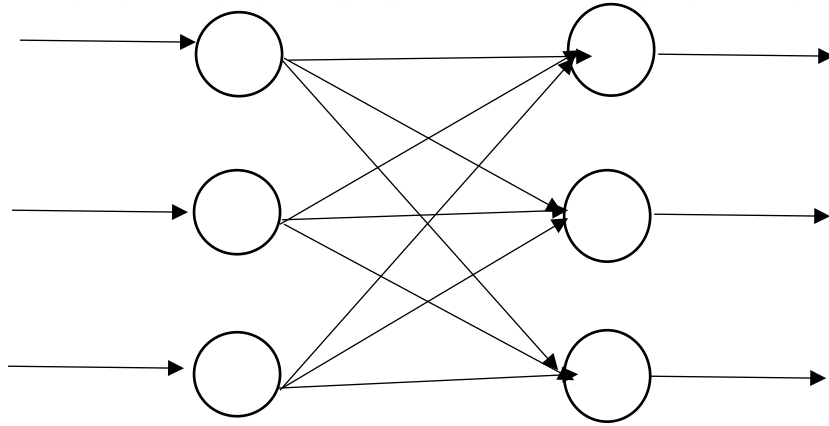
Yapay sinir ağları yapılarına göre ileri ve geri beslemeli olmak üzere ikiye ayrılırlar. Burada temel parametre nöronların birbirine bağlantı şekilleri olup yapılarına göre YSA'lar bu tez kapsamında aşağıda incelenmiştir.

1.2.1.1. İleri Beslemeli Yapay Sinir Ağları

İleri beslemeli YSA'lar, birimler arasındaki bağlantıların bir döngü oluşturmadığı YSA'lardır. İleri beslemeli YSA'ları, icat edilen ilk yapay YSA türüydü. Ayrıca, muadilleri olan RNN'e (tekrarlayan yapay sinir ağları) göre daha basittirler. İleri Besleme olarak adlandırılırlar, çünkü bilgi yalnızca ağda (döngü olmadan) önce giriş düğümleri, sonra gizli düğümler (varsa) ve son olarak da çıkış düğümleri aracılığıyla ileri doğru hareket eder. İleri beslemeli yapay sinir ağları, aşağıdaki gibi iki ana türe ayrılır. (E. Öztemel, 2012).

1.2.1.1.1. Tek Katmanlı İleri Beslemeli YSA (SLFFNN)

Tek katmanlı yapay sinir ağları doğrusal problemlerin çözümünde kullanılıp sadece girdi ve çıktı katmanından oluşmaktadır. Katmanların bir veya daha fazla nöronu bulunabilmektedir. Şekil 1.1 'de basit bir tek katmanlı algılayıcı modeli gösterilmiştir. Tek katman, hesaplama düğümlerinin (nöronlar) çıktı katmanını ifade eder. Kaynak nöronları giriş katmanı sayılmaz, çünkü burada herhangi bir hesaplama yapılmaz.



Şekil 1.1. Tek Katmanlı YSA Modeli (Öztemel, 2006)

Eşik girdisi, bu tip ağlarda nöron elemanlarının değerlerinin ve de ağın çıktısının 0 olmasını önler. Değeri daima 1'dir. Ağın çıktısı Denklem (1.1)'de verilen denklemin

gösterildiği gibi ağırlıklandırılmış girdi değerlerinin eşik değeri ile toplanması sonucu elde edilir.

$$\zeta = f(\sum_{i=1}^n x_i w_i + \phi) \quad (1.1)$$

Denklem (1.1)'de x_i , $i = 1, 2, \dots, n$ ile ağırlık girdileri, w_i , $i = 1, 2, \dots, n$ ile bu girdilere karşılık gelen ağırlık değerleri, ϕ ile eşik değeri gösterilmiştir. Tek katmanlı algılayıcıda çıktı fonksiyonu doğrusaldır (Bayır, 2006). Böylelikle, ağa gösterilen örnekler eşik değeri fonksiyonu ile iki sınıf arasında paylaştırılarak iki sınıfı birbirinden ayıran doğru bulunmaya çalışılır.

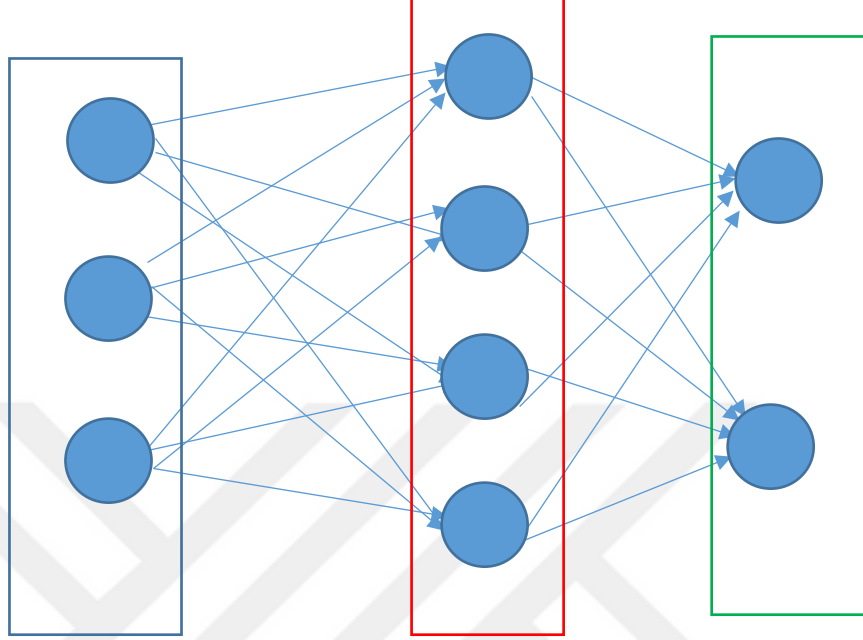
1.2.1.1.2. Çok Katmanlı İleri Beslemeli YSA (MLFFNN)

Tek katmanlı algılayıcıların doğrusal olmayan problemlerin çözümünde başarısız olmasının üzerine geliştirilmiş olup bu yapay sinir ağının mimarisi bir veya daha fazla gizli katmana sahiptir. Şekil 1.2. bir gizli katman ve bir çıkış katmanı ile tamamen bağlantılı ileri beslemeli ağı göstermektedir.

GİRİŞ KATMANI
INPUT LAYER

GİZLİ KATMAN
HIDDEN LAYER

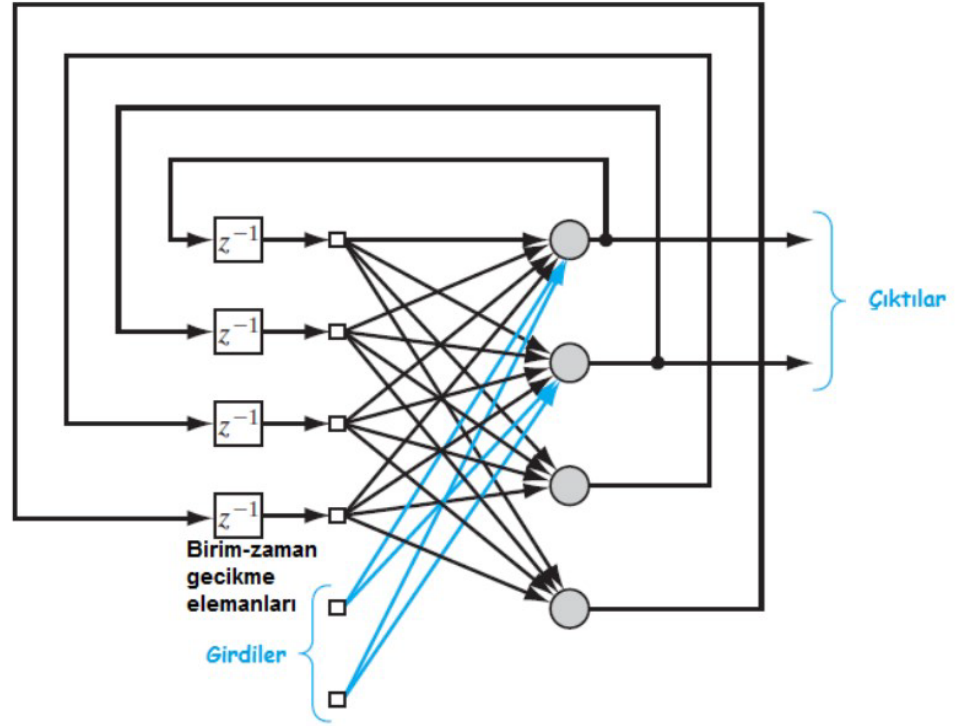
ÇIKIŞ KATMANI
OUTPUT LAYER



Şekil 1.2. Çok Katmanlı Yapay Sinir Ağı Modeli (Kızrak, 2018)

1.2.1.1.3. Tekrarlayan Yapay Sinir Ağları (RNN)

Tekrarlayan Yapay Sinir Ağları, karar fonksiyonlarında bazı bağlamları hesaba katabilen sinirsel modellerdir. RNN, Şekil 1.3'te giriş katmanları girdiler olarak ve çıkış katmanları ise çıktılar şeklinde belirtilerek gösterildiği gibi en az bir geri besleme döngüsüne sahip olmasıyla kendisini ileri beslemeli sinir ağından (MLFFNN) ayırır.



Şekil 1.3. Gizli nöronlarla Tekrarlayan Ağ Yapısı (RNN) (Haykin, 1998)

RNN, her bilgiyi zaman içinde hatırlar. Önceki girdileri de hatırlama özelliği sebebiyle zaman serisi tahmininde kullanışlıdır. Buna Uzun Kısa Süreli Bellek denir. Ayrıntılı olarak, RNN'nin avantajları ve dezavantajları Çizelge 1.1'de sunulmaktadır.

Çizelge 1.1. RNN'nin Temel Avantaj ve Dezavantajları

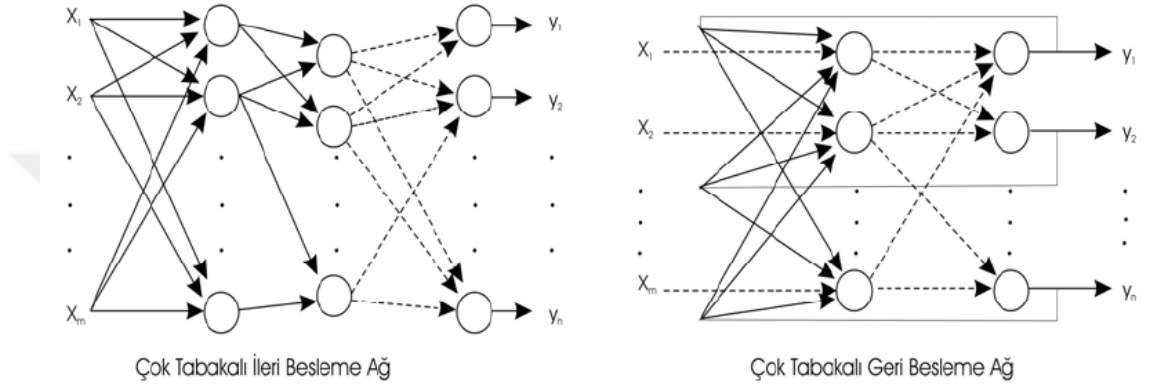
Yapay Sinir Ağı Tipi	Avantajları	Dezavantajları
TEKRARLAYAN YAPAY SİNİR AĞLARI(RNN)	• RNN, dinamik sinir ağıdır ve hesaplama açısından oldukça güçlüdür. Ayrıca birçok zamansal işleme modelinde ve uygulamarda kullanılabilir (Maass W,2007) • RNN'ler, evrensel yaklaşım özelliğine sahiptir, yani, giriş dizilerinden çıkış dizilerine kadar karmaşık eşlemeler gerçekleştirerek, rastgele lineer olmayan dinamik sistemlerin çözümünde oldukça etkilidir. (Torkar C, 2019). • RNN ileri beslemeli çok katmanlı diğer sinir ağlarına göre oldukça kompakttır. Yani boyut olarak daha küçüktür. • Tekrarlayan sinir ağı, insan-robot etkileşimi (HRI) görevlerinde yaygın olarak kullanılması bakımından oldukça önemlidir.	• RNN basit ve güçlü bir model olmasına rağmen, pratikte düzgün bir şekilde eğitilmesi zordur. • Özellikle Gradyan problemlerinden ötürü hantal bir modeldir. • Birim aktivasyon çıkış özelliklerinin doğrusal olmayan doğası ve ağırlık ayarlama stratejileri nedeniyle, ağ kararlılığının tespit edilmesi genellikle zordur. • Aktivasyon fonksiyonu olarak Tanh veya ReLU kullanılıyorsa çok uzun dizileri işleyemez.

Çizelge 1.1'den de görüleceği üzere Tekrarlayan sinir ağları (RNN) dinamik bir yapay sinir ağı olup birçok zamansal işleme modeli uygulamalarında kullanılabilmektedir. Boyutunun küçük ve kompakt olması en önemli avantajlarından birisi olmakla birlikte özellikle bu tez kapsamında da incelenen RELU aktivasyon fonksiyonunda uzun dizeleri işleyemediği görülmüştür. RNN Çizelge 1'de görüldüğü gibi, MLFFNN ile karşılaştırıldığında daha karmaşık bir yapıya sahiptir. RNN'nin matematiksel hesaplaması da MLFFNN ile karşılaştırıldığında daha karmaşık olduğu sonucuna varılmıştır (Kabalcı, 2015).

1.2.1.2. Geri Beslemeli Yapay Sinir Ağları

Geri beslemeli YSA modellerinde bağlantılar döngüsel biçimde yer alırlar. Geri beslemeli ağların ileri beslemeli ağlardan farkı, bir nöronun çıktısının sadece kendinden

sonra gelen nöron katmanına girdi olarak verilmesiyle birlikte ayrıca kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir nörona da girdi olarak tanıtılabilmesidir. Geri beslemeli YSA, sahip oldukları döngü sebebiyle girdinin karşılığını yavaş bir şekilde oluştururlar (Öztemel, 2006). Bu yüzden, bu tür ağların eğitime süreci daha uzun olmaktadır. Aşağıdaki şekilde yapay sinir ağlarında ileri ve geri beslemeli ağ yapılarının akış şeması gösterilmektedir (Kabalcı, 2015).



Şekil 1.4. İleri ve Geri Beslemeli Ağ Yapısı (Karakaya,2012)

1.2.1.2.1. Denetimli Yapay Sinir Ağları

Yapay sinir ağlarında denetimli öğrenmede sisteme tanıtılan giriş değerlerinden elde edilmesi hedeflenen çıkış verileri de sisteme tanıtılır. Ağ tanıtılan bu giriş verileri için istenen sonucu bulabilmek amacıyla kendi ağırlık atamalarını günceller. Ağ'dan elde edilen çıktılar ile bulunması istenilen değerler arasındaki hata payı hesaplanarak, ağın bu hata payına göre yeni ağırlık ataması gerçekleşir. Hata payı hesaplaması ise, sistemden elde edilen tüm çıktılar ile bulunulması beklenen çıktılar arasındaki fark hesaplanıp, bu farka göre tüm nöronlara düşen hata payı bulunur. Bu işlem ardından her nöron kendisine gelen ağırlıkları güncelleyerek işleme devam eder. Bu işlem, istenilen verilere en yakın değer elde edilene kadar devam eder (E. Öztemel, 2012).

1.2.1.2.2. Denetimsiz Geri Beslemeli YSA

Yapay sinir ağlarında denetimsiz öğrenme, sisteme sadece örnek verilerin tanıtılması ile gerçekleştirilebilir. Sisteme herhangi bir çıktı bilgisi verilmediğinden

sistem girişte tanıtılan veriler doğrultusunda her bir veriyi kendi arasında sınıflandırma işlemi yaparak kendi kurallarını oluşturur. YSA son olarak ağ bağlantı ağırlıklarını aynı özellikte olan dokuları ayırabileceği şekilde düzenleyerek öğrenme işlemi sonlandırır (Öztemel, 2006).

1.2.1.2.3. Destekleyici Geri Beslemeli YSA

Destekleyici öğrenme yaklaşımında, sistem ağı her bir iterasyon sonucunda elde ettiği verileri iyi ya da kötü olarak kategorize etmesiyle gerçekleşir. Bu veriler doğrultusunda ağ kendini yeniden düzenler ve bu şekilde ağ kendini elde ettiği bu bilgiler ile hem kendisini eğiterek hem de işlemlerden çıkardığı sonuçlar ile döngüyü devam ettirerek sürekli bir işleme tabi tutar (Öztemel, 2012).

1.3. Çok Katmanlı Yapay Sinir Ağlarında Matematiksel Modeller

1.3.1. Hebb

1949 yılında Donald Hebb tarafından geliştirilen biyolojik öğrenmeye dayalı en eski eğitim yasasıdır. Diğer öğrenme yasalarının da temeli olan Hebb yasasına göre, eğer bir yapay nöron başka bir hücreden bilgi alıyorsa ve her iki hücre de aktifse, iki hücre arasındaki bağlantının güçlendirilmesi gerekir (Anderson ve McNeill, 1992). Yani bu kuralı şu şekilde özetleyebiliriz: Bir hücre aktif ise aktif hücreye bağlanmaya çalışır, aktif değilse devre dışı bırakmaya çalışır. Kuralın çalışma şeklini bir örnekle açıklamak gerekirse; A hücresi, B hücreyi uyaracak kadar yaklaşırsa ve sürekli olarak B hücrenin aktivasyonuna dahil olursa, A hücrenin ağırlığı değişecek veya her iki A hücrenin aktivitesi artacaktır (Hebb, 1949).

1.3.2. Hopfield

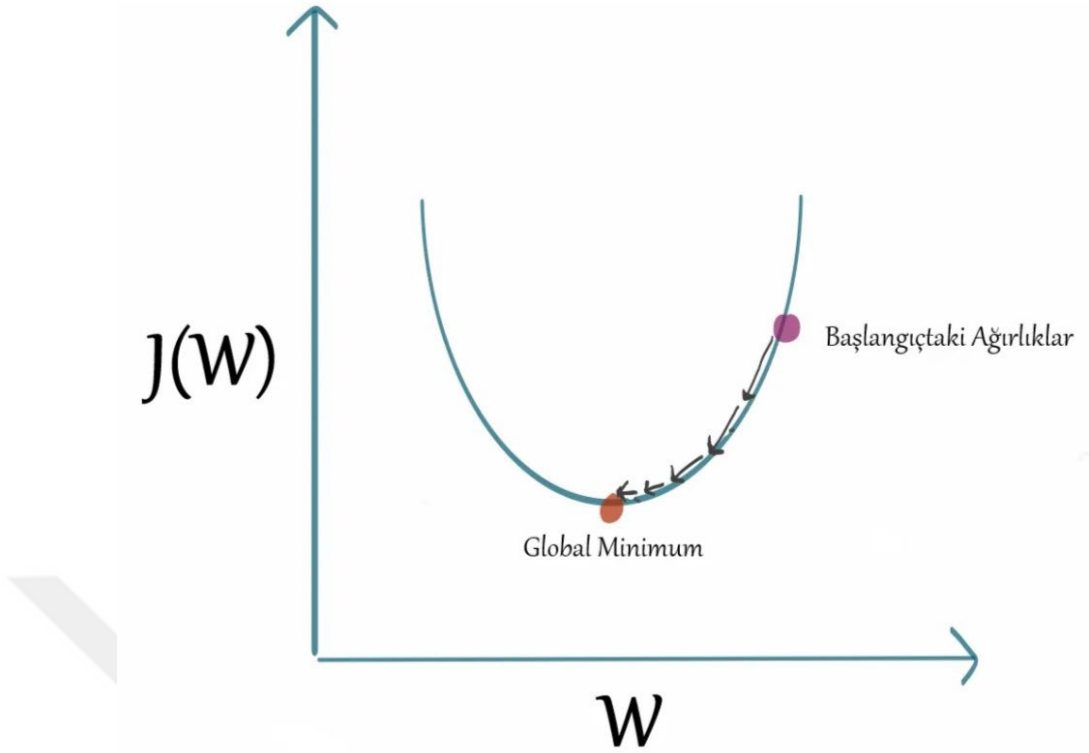
Hopfield yasası genel olarak Hebb yasasına benzer Yapay sinir ağı bileşenlerinin bağlantılarının ne kadar güçlü veya zayıf olması gerektiğini belirlemek için kullanılır. Yani ağırlıkların güçlü veya zayıf olması öğrenme kalitesi tarafından belirlenir. Öte yandan öğrenme katsayısı, genellikle 0 ile 1 arasında olan ve kullanıcı tarafından atanan sabit ve pozitif bir değerdir (Öztemel, 2012).

1.3.3. Delta

Bu, Hebb kuralının biraz daha gelişmiş bir versiyonudur. Bu kuralın prensibi, beklenen çıktı ile gerçek çıktı arasındaki farkı azaltmak için sinir ağı bileşenlerinin ağırlık değerlerini sürekli güncellemektir. Amaç ağı ürettiği çıktı ile üretilecek çıktı arasındaki ortalama karesel hatayı en aza indirmeyi amaçlar. Hata sıfır ise ağırlıklar değiştirilmeden öğrenme süreci durur. Hata pozitif ise ağırlıklar kabul edilene kadar değiştirilir. Delta kuralının mekanizması şu şekildedir: Çıkış katmanındaki delta hatası, transfer fonksiyonunun türevi kullanılarak dönüştürülür ve bir önceki katmana gönderilir. Bu değişikliğin bir sonucu olarak, önceki seviyeye giriş bağlantılarının kalitesi değişir, böylece hata her seferinde bir önceki seviyeye döner. Girdi kümesindeki verilerin delta kuralı kullanılarak rastgele dağıtıldığına dikkat edilmelidir. Eğitim örneğindeki veriler doğru dağıtılmazsa istenilen doğruluğun elde edilmesi zorlaşacak ve ağı eğitimi engellenecektir (Bayır, 2006).

1.3.4. Dereceli Azaltma

Gizli katmanı olmayan bir ağda, hata yüzeyinde yalnızca bir optimal nokta vardır. Gizli katmanda, optimum noktayı bulmak zordur çünkü birden fazla optimal nokta vardır. Bu durumda, ağırlıklar eğim inişi kullanılarak değiştirilir. Yöntem en iyi değer bulunana kadar Eğim iniş yönteminde, ağı çıkış değeri her bir ağırlık değeri ile kısmen elde edilir. Bu yöntemde elde edilen değer her zaman bir önceki değerden küçük olacaktır. Aşağıda verilen grafikte yatay eksen W ağırlık girdilerini ve dikey eksen ise bu ağırlıkların çıktı değerlerini $J(W)$ olarak ifade etmektedir.



Şekil 1.5. Hata Değişim Eğrisi (Akça, 2020)

Öğrenme sürecindeki hataları düzeltirken her döngü bu parabolün üzerine iner. Gizli bir seviye varsa yerel minimum değere iner. Bunu görselleştirmek için ağırlıkların olası tüm değerleri x, y ve z koordinat düzleminde çizilir. Hatanın karesine karşılık gelir. Bu çizim bize hata yüzeyini verir. Şekil 1.5'te gösterildiği gibi, bu yüzey bir çanak şeklindedir. Kutunun altı, kareleri alınmış hataların en küçük toplamına karşılık gelir. Ağı eğitirken, amaç en iyi ağırlıklar kümesidir, geri yayılım algoritması, mevcut ağırlıklar yerine hata yüzeyine olan eğimi hesaplayarak amacına ulaşır. Daha sonra bu ağırlıkları kabın dibine doğru hareket ettirir. Kâsenin üstünden kâsenin dibine doğru artan bu ilerlemeye eğme denir (Bayır, 2006).

1.3.5. Kohonen

Teuvo Kohonen (1982), bu öğrenme ilkesini biyolojik sinir sistemlerinde meydana gelen öğrenmeden esinlenmiştir. Bu kuralda ve düğümü rekabetçi bir öğrenme kuralı olarak görmek, işlemci bileşenleri ağırlıkları belirlemek için rekabet eder. Bu özellik nedeniyle bu kural, kazanan kural olarak da bilinir (Yavuz, 2012).

Tez kapsamında verilen matematiksel modellerin başarılı sonuçlar vermesi için seçilen matematiksel modelin içerisinde hatanın minimize edilebilmesi açısından uygun aktivasyon kodunun kullanılması önemlidir.

1.4. Aktivasyon Kodları

Aktivasyon kodları YSA için çok önemli bir parametre olup nöronların öğrenmesini, süresini ve mühendislik açısından sağladığı avantaj ve dezavantajları etkilemesi bakımından oldukça kritiktir. Bu tez kapsamında uygun aktivasyon kodunu belirlemek için YSA’da kullanılan aktivasyon fonksiyonları irdelenmiştir.

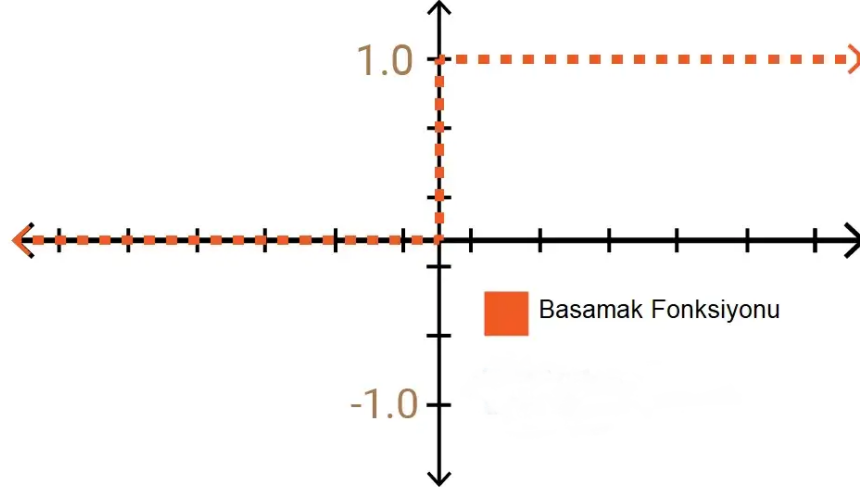
Nöral ağ süreci üzerine eğitim, çoklu gizli katmana sahip olduğunda karmaşık ve zorlu hale gelir. Zikzaklı ağırlık, yok olan gradyan problemi, çok karmaşık formül veya aktivasyon fonksiyonunun sinir ağında doygunluk problemi bu zorluklardan bazılarıdır. Bu da uzun zaman alabilen sürekli bir öğrenme sürecine yol açmaktadır Byrd ve ark. (2016), yaptıkları bir araştırma makalesinde yapılan farklı aktivasyon fonksiyonlarının karşılaştırmasını hem pratik hem de teorik olarak tartışmıştır. Bu etkinleştirme işlevleri arasında softplus, tanh, swish, linear, Maxout, sigmoid, Leaky ReLU ve ReLU bulunur. Her işlevin analizi bir tanım, kısa bir açıklama ve eksilerini ve artılarını içerir. Bu, her durum için en iyi aktivasyon işlevini seçmek için yönergeler formüle etmemizi sağlar (Szandala, 2021).

1.4.1. Aktivasyon Fonksiyonları

Aktivasyon fonksiyonu hücreye gelen net girdinin ağırlıklı toplamı hesaplandıktan sonra bu toplama sabit bir hata değeri (bias) eklenerek elde sonucu işleyerek bu girdiye karşılık gelen çıktıyı hesaplar. Aktivasyon fonksiyonları doğrusal değildir. Bu durum karmaşık yapılarda öğrenmenin gerçekleşmesi açısından önemlidir, çünkü geri beslemeli ağlarda hatanın minimize edilmesi açısından geriye yayılım algoritmasından ağırlıklar güncellenmesi işleminde türev alınır. Bu sebepten hesaplamaların da yavaşlamaması için türevi kolay hesaplanabilen ve genellikle de öğrenmenin gerçekleşmesi için her noktada türevi farklı değerlere karşılık gelen fonksiyonlar tercih edilmektedir (Kızrak, 2020).

1.4.1.1. Basamak (Step) Fonksiyonu

$$(f(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases}) \quad (3.1)$$

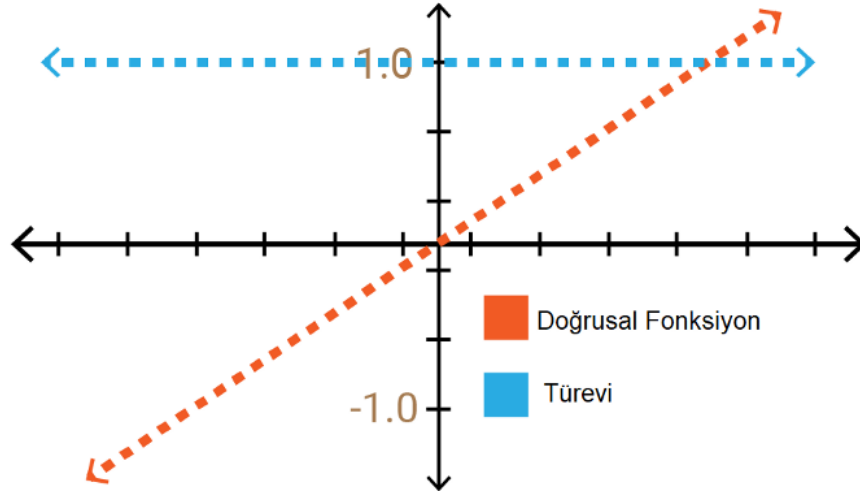


Şekil 1.6. Basamak (Step) Fonksiyonu (Kızrak, 2019)

Temel olarak eşik tabanlı bir sınıflandırıcıdır. İkili değer alan bir fonksiyon olduğundan dolayı genellikle çıkış katmanlarında tercih edilir. Türevinin 0 değerine eşit olması öğrenmenin gerçekleşmediği anlamına gelir. Başka bir ifadeyle fonksiyonun gradyanı sıfır olduğu için ağırlıklar ve eğilimler güncellenemez. Denklem (3.1) ve Şekil 1.6. incelendiğinde basamak fonksiyonunun çıkış katmanlarında kullanılması uygun olabilir. (Kumawat, 2019).

1.4.1.2. Doğrusal (Linear) Fonksiyon

$$(f(x) = x) \quad (3.2)$$

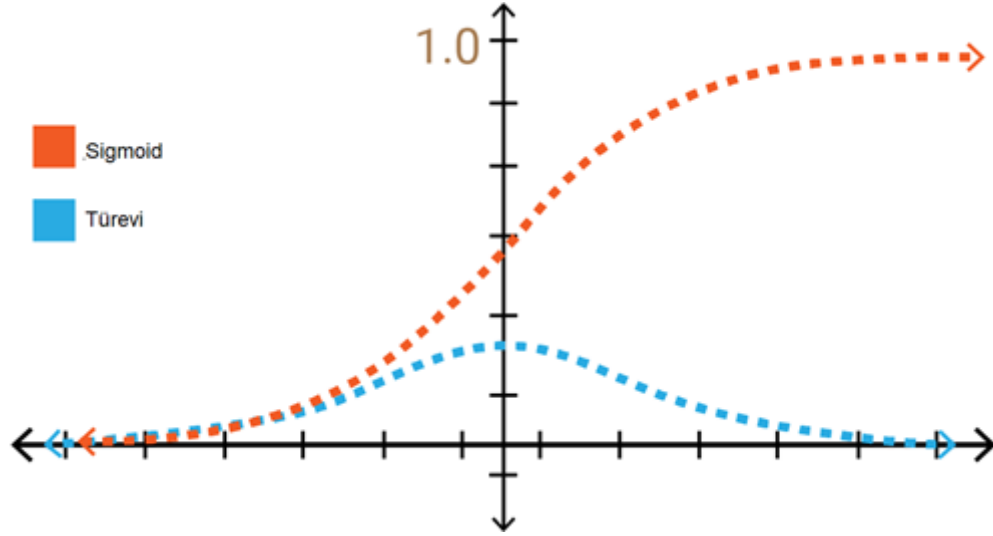


Şekil 1.7. Doğrusal (Linear) Fonksiyon (Kızrak, 2019)

Basamak fonksiyonundan farklı olarak ikili değerle sınır olmayan bir dizi aktivasyon değeri üretir. Ancak türevinin sabit bir değer alması öğrenmenin sürekli olarak gerçekleşmediği anlamına gelir. Bununla birlikte bütün katmanlarda doğrusal fonksiyon kullanılırsa giriş ve çıkış katmanlarında hep aynı doğrusal sonuca ulaşarak ara katmanlar işlevsiz kalabilir. Denklem (3.2) ve Şekil 1.7. incelendiğinde YSA içerisinde bulunan ara katmanları görmezsek yalnızca giriş katmanının doğrusal bir fonksiyonu olan bir çıktı elde edilebilir. (Kumawat, 2019)

1.4.1.3. Sigmoid Fonksiyonu

$$\left(f(x) = \frac{1}{1+e^{-x}} \right) \quad (3.3)$$



Şekil 1.8. Sigmoid Fonksiyonu (Kızrak,2019)

Doğadaki problemler genellikle doğrusal değildir. Bu sebepten sigmoid fonksiyonunun kombinasyonlarının da doğrusal olmaması önceki fonksiyonlara göre avantajlı olmasını sağlar.

Sigmoid fonksiyonunun türevlenebilir olması öğrenmenin gerçekleşebileceği anlamına gelir. Grafikte belli noktalarda y değerleri hızlı bir değişim göstermektedir. Bu x'te yapılacak küçük değişimlerin y de büyük sonuçlara karşılık geleceği anlamına gelir. Bu sigmoid fonksiyonun iyi bir sınıflayıcı olduğu anlamına gelir. Bunlarla beraber fonksiyonun diğer bir artısı da $(-\infty, \infty)$ tanım aralığında her zaman $(0,1)$ Aralığında değerler üretmesidir.

Fonksiyonun grafiğinde uçlara doğru grafik dikkatlice incelendiğinde y değerleri x'teki değerlere çok az tepki verdiği için bu tanım aralığındaki değerlerde türev değerleri küçülerek sıfıra yakınsar ve bu da öğrenmenin bu noktalarda gerçekleşmediği anlamına gelir. Öğrenme olayı yavaş olduğunda hatayı minimize eden optimizasyon algoritması yerel(lokal) minimum değerlere takılabilir ve bu durum maksimum performansın elde edilmesine engel olabilir.

Sonuç olarak sigmoid hesaplama açısından maliyetli olup gradyanların yok olması problemine neden olur. Sıfır merkezli değildir. Bu yöntem genellikle ikili sınıflandırma problemlerinde kullanılabilir (Jain, 2019).

1.4.1.4. Softmax Fonksiyonu

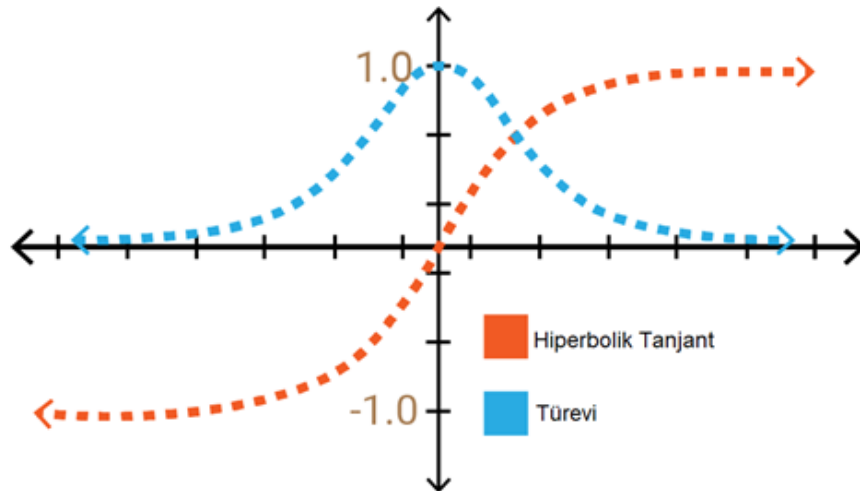
$$\left(\overrightarrow{f(z)}\right)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \forall i \in 1 \dots N \quad (3.4)$$

Sigmoid fonksiyonu ile benzer yapıdadır. Sigmoid fonksiyonu gibi sınıflayıcı olarak kullanılırsa iyi performans sergileyebilir. Asıl farkı Sigmoid fonksiyonu gibi ikiden fazla sınıflama gerektiren durumlar için genellikle derin öğrenme modellerinin çıkış katmanlarında kullanılabilir. Girdinin belirli bir sınıfa ait olma olasılığını (0,1) aralığında değerler üreterek sağlayabilir. Bu durumda olasılıksal bir yorumlama gerçekleştirir. Çoklu sınıflandırma gerektiren problemler için softmax fonksiyonu her bir sınıfa ait olasılıkları kolayca yorumlanabilen değerleri atamayı kolaylaştırdığı için tercih edilmektedir.

Sonuç olarak softmax, sigmoidin genelleştirilmiş bir şeklidir. Çok sınıflı sınıflandırma problemlerinde kullanılabilir. Sigmoide benzer şekilde (0,1) aralığında değerler ürettiğinden dolayı sınıflandırma modellerinde son katman olarak tercih edilebilir (Jain, 2019).

1.4.1.5. Hiperbolik Tanjant Fonksiyonu

$$\left(\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}\right) \quad (3.5)$$

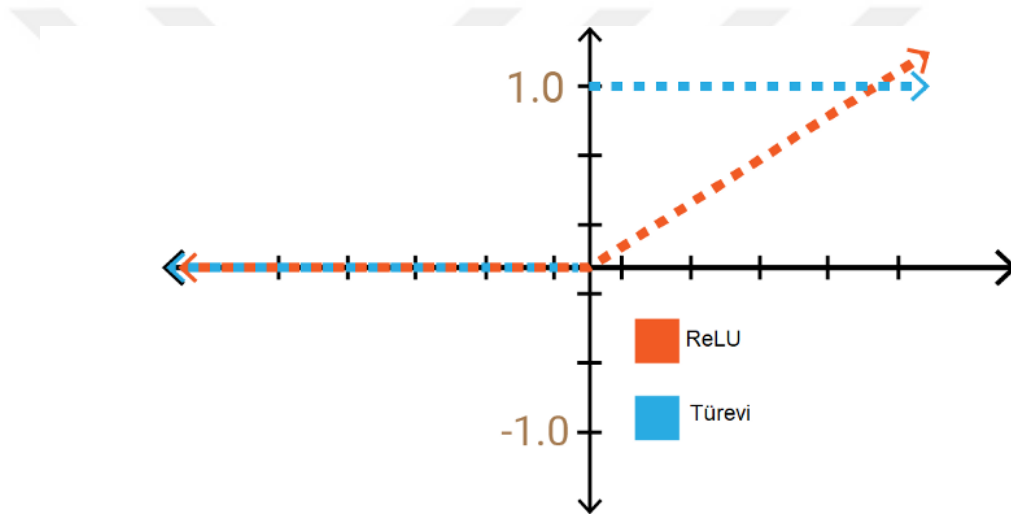


Şekil 1.9. Hiperbolik Tanjant Fonksiyonu (Kızrak, 2019)

Sigmoid fonksiyonuna benzerdir. Fakat fonksiyonun aralığı $(-1,+1)$ olarak tanımlanmaktadır. Sigmoid fonksiyonuna kıyasla sağladığı avantajı türevinin daha çok değer alabilmesidir. Böylece daha hızlı öğrenme ve sınıflama işlemi için daha geniş aralığa sahip olmasıyla YSA'nın eğitimi daha verimli olabilir. Ama grafik incelenerek yine fonksiyonun uçlarında gradyanların yok olması problemi görülebilir (Jain, 2019).

1.4.1.6. ReLU (Rectified Linear Unit) Fonksiyonu

$$f(x) = \begin{cases} 0 & \text{ için } x < 0 \\ x & \text{ için } x \geq 0 \end{cases} \quad (3.6)$$



Şekil 1.10. ReLU (Rectified Linear Unit) Fonksiyonu (Kızrak, 2019)

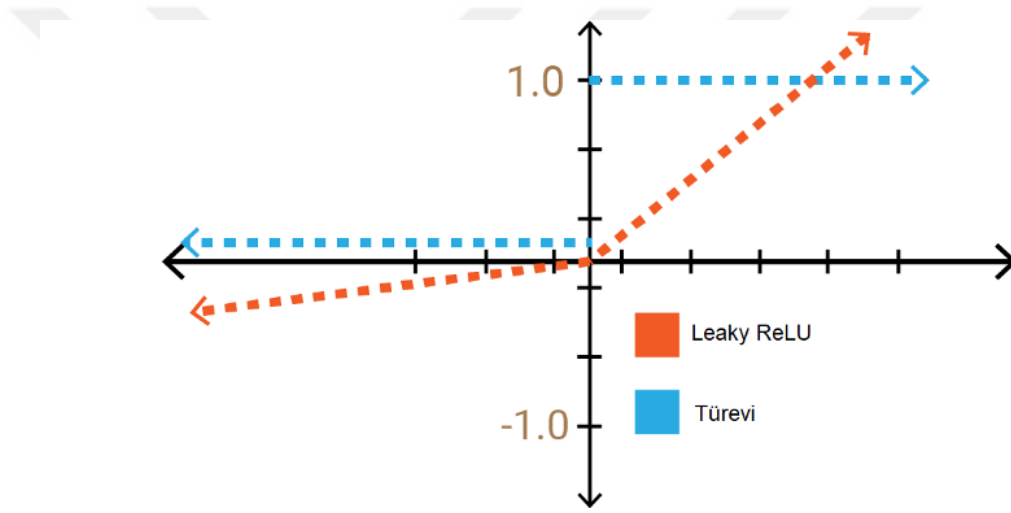
ReLU, düzeltilmiş doğrusal birim anlamına gelir ve bir tür aktivasyon işlevidir. Matematiksel olarak $y = \max(0, x)$ şeklinde tanımlanır. Grafik incelendiğinde pozitif ekseninde doğrusal fonksiyon ile aynı özelliklere sahip gibi görünmesine rağmen ReLU doğrusal bir fonksiyon değildir. ReLU fonksiyonunun iyi bir tahmin edici olduğu söylenebilir. ReLU'nun kombinasyonları ile herhangi başka bir fonksiyona da yakınsamak mümkün olabilir.

ReLU $[0, +\infty)$ aralığında değerler almaktadır. Ancak çok fazla nöronu olan bir sinir ağında Sigmoid ve Hiperbolik Tanjant neredeyse tüm nöronların aynı şekilde aktive olmasına sebep olmasından dolayı aktivasyon çok işlem gerektiriyordu. Ağdaki bazı nöronların aktif olup, aktivasyon seyrek, yani verimli bir hesaplama yükü olması ağı daha hızlı ve başarılı bir şekilde eğitilmesini sağlayabilir. ReLU ile bu açıdan oldukça iyi

sonuçlar verebilir. Hesaplama yükü açısından sigmoid ve hiperbolik tanjant fonksiyonlarıyla kıyaslandığında hesaplama yükünün az olmasından dolayı çok katmanlı ağlarda daha çok tercih edilmesine sebep olduğu söylenebilir. İşlem hızı kazandıran bu sıfır değer bölgesinin türevinin de sıfır olması öğrenmenin o bölgede gerçekleşmiyor olması anlamına gelmesi ReLU fonksiyonun da bir noktada yetersiz kalacağı anlamına gelebilir (Liu, 2017).

1.4.1.7. Sızıntı (Leaky) ReLU Fonksiyonu

$$\left(f(x) = \begin{cases} 0,01 & \text{için } x < 0 \\ x & \text{için } x \geq 0 \end{cases} \right) \quad (3.7)$$

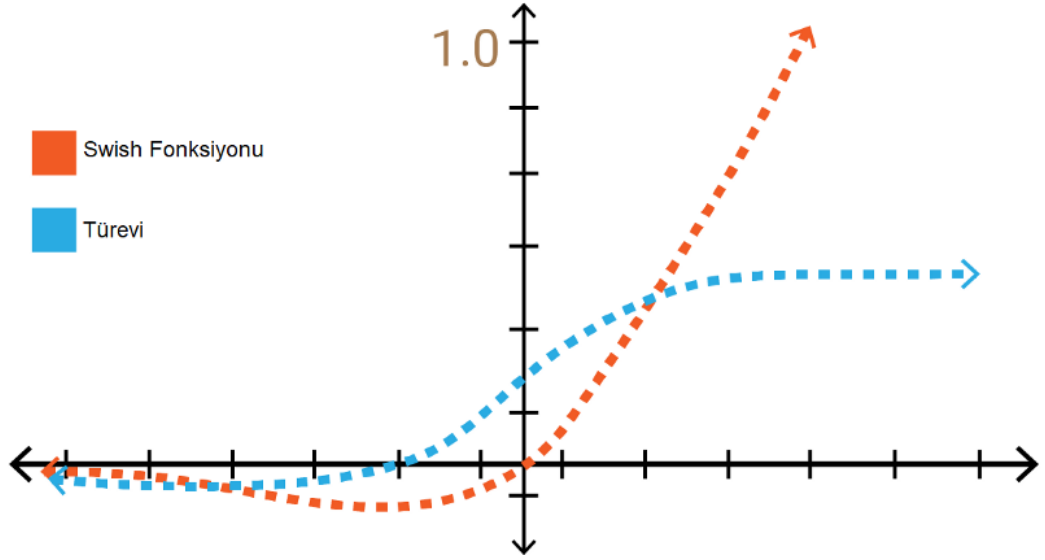


Şekil 1.11. Sızıntı (Leaky) ReLU Fonksiyonu (Kızrak, 2019)

Sızıntı değeri 0,01 olarak verilir. Sıfıra yakın farklı bir değer verilirse fonksiyonun adı rastgele Leaky ReLU olarak değişmektedir. Sızdırılan ReLU'nun tanım aralığı eksi sonsuza doğru devam eder. Bu 0'a yakın ama 0 olmayan değerler sayesinde ReLU'daki yok olan gradyanları var ederek öğrenmenin negatif bölgedeki değerler için de sağlandığı söylenebilir. (Liu, 2017).

1.4.1.8. Swish (A Self-Gated/Kendinden Geçitli) Fonksiyonu

$$\left(f(x) = 2x\sigma(\beta x) = \begin{cases} \beta = 0 & \text{için } f(x) = x \\ \beta \rightarrow \infty & \text{için } f(x) = 2\max(0, x) \end{cases} \right) \quad (3.8)$$



Şekil 1.12. Swish (A Self-Gated/Kendinden Geçitli) Fonksiyonu (Kızırak, 2019)

Denklem (3.8) ve Şekil 12 incelendiğinde Swish Fonksiyonu $\beta = 0$ olduğunda doğrusal (linear) fonksiyona dönüşür. ReLU’dan farkı negatif bölgede değer alabiliyor olmasıdır. Leaky ReLU’dan farkı ise Swish’in negatif bölgedeki değerlerinin doğrusal olmamasıdır. Diğer tüm aktivasyon fonksiyonları monotondur. Swish fonksiyonun çıktısı girdi arttığında bile düşebiliyor. Bu swish’e özgü bir özelliktir (Kızırak, 2020).

1.4.1.9. Swish Fonksiyonu

$$f(x) = \frac{2 \cdot x}{1 + e^{-\beta \cdot x}} \quad (3.9)$$

Swish fonksiyonu tanımında bulunan beta’nın öğrenilebilir bir parametre olması Swish’in basit bir versiyonu olup beta değeri 0 olduğunda Sigmoid kısmı her zaman 0,5 değerini alır ve $f(x)$ doğrusal olur. Diğer yandan eğer beta çok büyük bir değer seçilirse Sigmoid ikili basamak fonksiyonuna yakın bir fonksiyona dönüşür. Böylece $f(x)$ ReLU fonksiyonuna yakınsar. Bu nedenle standart Swish fonksiyonunda beta değeri 1 olarak seçilebilir. Böylece yumuşak bir enterpolasyon sağlanabilir. Bu sayede gradyanların yok olması problemi giderilerek öğrenme üst düzeyde gerçekleşebilir. (Kızırak, 2020)

Yukarıda verilen aktivasyon fonksiyonlarının matematiksel ifadeleri ve aldıkları değer aralığını veya diğer bir ifadeyle çıktı aralığını ifade eden Çizelge 1.2’de verilmiştir.

Çizelge 1.2. Aktivasyon Fonksiyonlarının Matematiksel İfadeleri

AKTİVASYON FONKSİYONU	DENKLEM	ARALIK
Doğrusal Fonksiyon	$f(x) = x$	$(-\infty, \infty)$
Basamak Fonksiyonu	$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases}$	$\{0,1\}$
Sigmoid Fonksiyonu	$f(x) = \frac{1}{1 + e^{-x}}$	$(0,1)$
Hiperbolik Tanjant Fonksiyonu	$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1,1)$
ReLU Fonksiyonu	$f(x) = \begin{cases} 0 \text{ için } x < 0 \\ x \text{ için } x \geq 0 \end{cases}$	$[0, \infty)$
Leaky (Sızıntı) ReLU Fonksiyonu	$f(x) = \begin{cases} 0,01 \text{ için } x < 0 \\ x \text{ için } x \geq 0 \end{cases}$	$(-\infty, \infty)$
Swish (A Self-Gated/Kendinden Geçitli) Fonksiyonu	$f(x) = 2x\sigma(\beta x)$ $= \begin{cases} \beta = 0 \text{ için } f(x) = x \\ \beta \rightarrow \infty \text{ için } f(x) = 2\max(0, x) \end{cases}$	$(-\infty, \infty)$
Swish Fonksiyonu	$f(x) = \frac{2 \cdot x}{1 + e^{-\beta \cdot x}}$	$(-\infty, \infty)$

Verilen aktivasyon fonksiyonları çıktının üretilmesini sağlayan doğrusal kombinasyonun doğrusal olmayan dönüşümü ortaya koymasını sağlayabilir.

Yapay sinir ağının eğitiminde başlangıçta W ağırlık vektörünün değerleri rastgele atanabileceği gibi geçmişte eğitilmiş bir modelin ağırlıkları giriş başlangıç ağırlıkları olarak tanımlanabilir. Ancak bu değerler atanırken (+) ya da (-) değerlerden rastgele

dağılımlı olarak belirlenebilir. Çünkü '0' değeri verildiği durumlarda hesap katmanlarda sürekli aynı çıkacağından öğrenme gerçekleşmeyebilir.

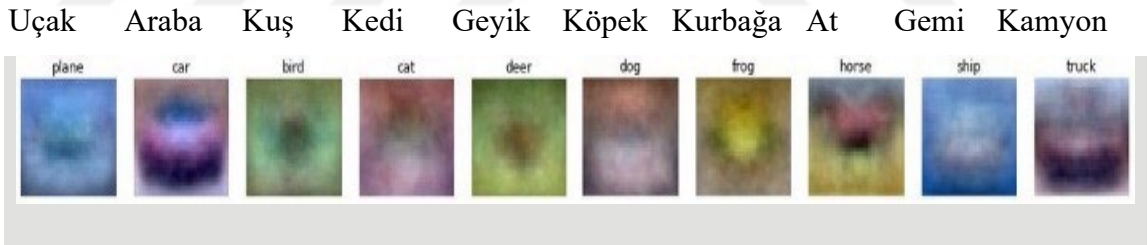
XOR problemine benzer tek katmanlı modelle çözilemeyen problemler için optimizasyon fonksiyonları tasarlanarak çözümü kolaylaştırmak mümkün olabilir.

Başarılı şekilde tasarlanan bir ağ modelinin başarımını belirleyen Kayıp/Yitim fonksiyonu (Loss function) değeri çeşitli optimizasyon teknikleriyle '0'a yaklaşması istenen bir fonksiyondur. Burada eğitim ve test gruplarının karşılaştırılmasıyla hesaplanan bir değerden söz edilmektedir. Ancak bu iki farklı kısıta sebep olabilir.

Modeldeki tüm bilgileri kullanıldığında büyük boyutlu bir parametre bilgisi elde edilir.

Test işlemini yaparken eğitim setindeki tüm verilerle karşılaştırılma yapıldığında bu işlem yükünü artırır ve tespit süresini uzatılabilir.

Bu iki durumun verimli bir ağ modelinde olmaması daha iyi olabilir. Basit bir çözümle üstesinden gelinebilir. Test verileri eğitim verilerinden oluşan bir temsil grubuyla karşılaştırılırsa işlem yükü azaltılarak sistemin hız bakımından performans kazanması sağlanabilir. (Kızrak, 2020).



Şekil 1.13. Örnek sınıflar için temsil verileri (GitHub, 2013)

Eğitim verisinin içeriğindeki tüm bilgileri temsil eden bir veri elde etmek gerekebilir. Bunun için öğrenme işleminden sonra ilgili verinin ağırlık matrisi (ortalaması) temsil verisi olarak alınabilir.

Loss (yitim) değerine eklenen iyileştirme yani Regularization (R) değeri sebebiyle loss hiçbir zaman 0'a eşit olmayabilir. Bu durumun W ağırlık vektörünün 0 olduğu durumlarda geçerli olduğu söylenebilir.

Regularization (iyileştirme veya regülasyon) teknikleri genellikle tasarlanmış bir modelde overfitting (aşırı öğrenme veya ezberleme) olmasını önleyerek başarıyı artırmak için kullanılmaktadır. Regülasyon sayısı kullanılması zorunlu olmayan bir

hiperparametre olup öğrenme başarısına pozitif yönde katkısından dolayı kullanılması tavsiye edilebilir. Loss hesabı için çeşitli optimizasyon teknikleri uygulanabilir. İki sınıflı basit bir problem için sigmoid ve daha karmaşık sınıflandırma problemlerinde ise SVM’de $s(f(W,X))$ skor fonksiyonu model ile üretilen skor değerleri baz alınarak olasılıksal yitim hesabı yapar. Burada olabilirlik değeri maksimize ve negatif olabilirlik değerini minimize etmek mantıklı olabilir. Bu işlemler yapılırken olasılık değerlerini en büyük olabilirlik yöntemiyle (maksimum olasılık yöntemi) dönüştürme işlemi yapmak gerekir. Yitim fonksiyonunun parçalı ve doğrusal bir matematiksel yapıda olduğu söylenebilir (Kızırak, 2020).

$$R(W) = \sum_k \sum_l W_{k,l}^2 \quad (3.10)$$

$$L = \frac{1}{N} \sum_i L_i + \lambda R(W) \quad (3.11)$$

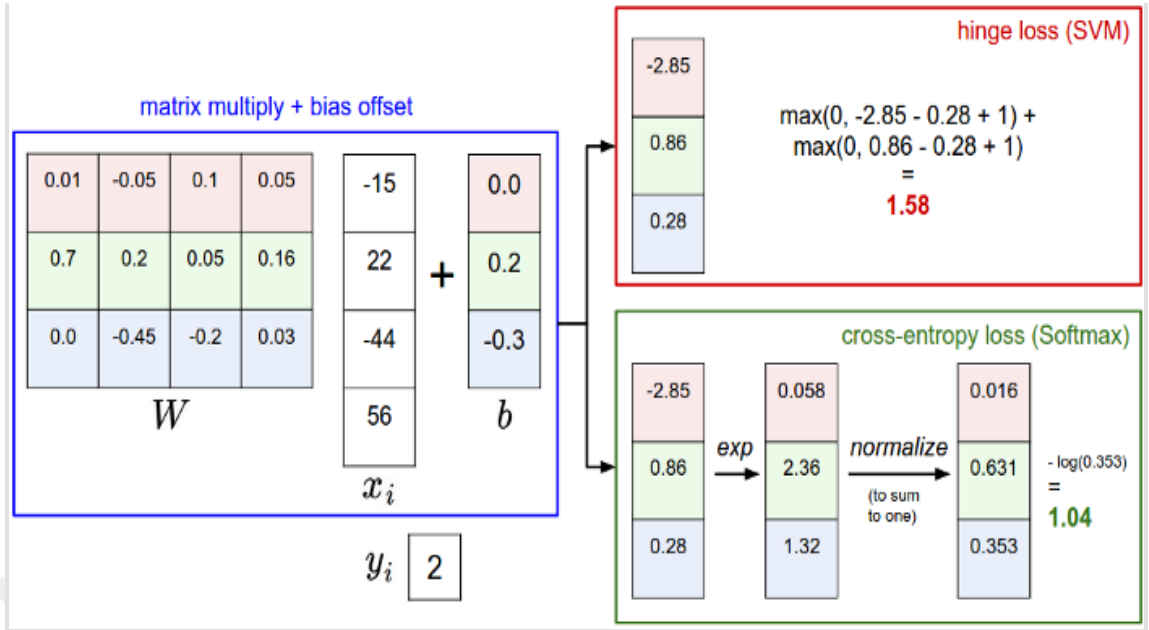
veri kaybı regularizasyon kaybı

$$L = \frac{1}{N} \sum_i \sum_{j \neq y_i} [\max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + \Delta)] + \Delta \sum_k \sum_l W_{k,l}^2 \quad (3.12)$$

Yukarıda görülen Denklem (3.11)’de bulunan λ , oluşabilecek bir bilgi kaybının önüne geçmek için kullanılan katsayıdır. Denklem (3.12)’de Δ değeri ise tahmin değeri ile gerçek değer arasındaki farkı göstermektedir. Denklemler incelendiğinde regulasyon aslında yitim fonksiyonuna eklenen bağımsız parametre olarak değerlendirilebilir.

SVM ya da Softmax kullanılacağına karar verebilmek için iki fonksiyonu da kullanacağımız uygulamada deneyip değerlendirdikten sonra karar verilmesi daha iyi olabilir.

SVM (Destek Vektör Makineleri), benzerlik skoru üretirken Softmax benzerlikle ilgili olasılık değerleri üretir ve sonuç üzerinde normalizasyon yaparak olasılıksal hale dönüştürebilir. Bu sayede sınıflar arasındaki fark daha kolay yorumlanabilir. Sınıflandırma problemlerinde Softmax fonksiyonunun tercih edilmesi tavsiye edilebilir. Sistemin performansı açısından karşılaştırıldığında fark azdır. Aşağıda SVM ve Softmax için yitim hesabı gösterilmiştir (Kızırak, 2020).



Şekil 1.14. Örnek yitim hesabı gösterimi (GitHub, 2013)

2. ÖNCEKİ ÇALIŞMALAR

Yapay sinir ağları robotik alanlarda (Sharkawy ve ark.,2020), konuşma tanımda (Passricha ve ark., 2018), insan yüzü tanıma (Zhao ve ark., 2004), tıbbi uygulamalarda (Moreno-Escobar, 2018), imalat endüstrisinde (Abdelhameed, 2002), ve ekonomi gibi değişik alanlarda birçok araştırmacı tarafından kullanılmaktadır (Badea LM, 2018).

Robotik uygulamalarda, Sharkow (2018) ve arkadaşları tarafından Çok Katmanlı İleri Beslemeli Sinir Ağı (MLFFNN) önerilmiş olup insan operatör ve robot arasındaki iş birliği sırasında meydana gelen çarpışmaları tespit etmek için kullanılmıştır. Ito M ve ark. (2006), ise parametrik önyargılı tekrarlayan veya devirli yapay sinir ağı (RNN) ile küçük bir insansı robot tarafından nesne taşıma davranışlarının öğrenilmesi üzerine deneysel çalışmalar yapmıştır. Yine Sharkow (2020) tarafından MLFFNN'nin çevrimiçi eğitimine dayanarak insan-robot iş birliğinde değişken bir giriş kontrolü kullanılmasını önermiştir. Bu giriş kontrolöründe, değişken sönümleme ayarlanmıştır.

Konuşma tanıma alanında keza YSA kullanılması tavsiye edilmiştir. Passricha ve ark. (2018), yaptıkları çalışmalarda ham konuşma sinyali için Evrimsel Sinir Ağları (CNN) tabanlı akustik modeli tartışmışlardır. Bu çalışmada, ham konuşma sinyali ile telefonlar arasında veri odaklı bir ilişki kurulmuştur. Palaz D ve ark. (2015), geniş kelime dağarcığı konuşma tanıma görevi için evrimsel YSA (CNN) tabanlı yaklaşım önermiştir. Bu yaklaşım, Wall Street Journal dergisinde klasik YSA tabanlı yaklaşımla karşılaştırılmıştır.

İnsan yüzü tanıma ile ilgili literatürde YSA kullanılan birçok çalışma mevcuttur. Örneğin Fadzil ve Bakar (2004), bir çalışmada insan yüzü tanıma sisteminin geliştirilmesi için MLFFNN tabanının daha uygun olacağını belirtmiştir. Bu çalışmada sinir ağı, öğrenilen duruma gelene kadar bir dizi yüz görüntüsü ile eğitilmiştir. Çalışma sonucunda Ağ, yüz girdisini kendi sınıfına sınıflandırabilir. Zhao (2004) tarafından sinir ağları komite makinesi kullanılarak farklı özelliklere dayalı yeni bir yüz tanıma yöntemi geliştirilmiştir. Yapılan çalışmada kullanılan Komite, farklı özellik alanlarındaki orijinal görüntülerin farklı görüntü blokları ile eğitilen birkaç bağımsız sinir ağından oluşturulmuştur.

Yapay sinir ağları, tıbbi uygulamalar ve hastalıklara tanı konulması gibi konularda yaygın olarak kullanılmaktadır Fukuka (2002) ve Moreno(2007), tıbbi görüntü işleme uygulamalarında (YSA) modellerinden olan Radyal Tabanlı Fonksiyon (RBF) ağlarını

kullanılmışlardır. Er O ve ark. (2010), çok katmanlı, olasılıksal, öğrenen vektör kuantizasyonunu ve genelleştirilmiş regresyon nöral ağları kullanılarak karşılaştırmalı bir göğüs hastalıkları teşhisi gerçekleştirilmiştir. Göğüs hastalıkları veri seti, bir göğüs hastalıkları hastanesinin veri tabanından alınan hasta epikriz raporları kullanılarak hazırlanmıştır.

Ekonomi uygulamalarında da yapay yapay sinir ağları son derece kullanılan bir yöntemdir. Falat L ve Pancikova (2015), finansal verileri tahmin etme süreci için bir yaklaşım olan RBF'yi kullanmıştır geleneksel anketlerden elde edilen bilgilere dayalı olarak yapay sinir ağları kullanılarak tüketici davranışı tanımlanmıştır. Sonuçlar, sinir ağlarının iyi bir ayırt edici güce sahip olduğunu göstermiştir.

Matematik alanında ise YSA'ların matematiksel incelenmesi ile ilgili literatür taraması yapıldığında yapılan çalışmaların son derece sınırlı olduğu ve birçok çalışmanın deneysel olduğu gözlemlenmiştir (Unal, 2019). Benjamin D. Haefele ve Ren'e Vidal 'Yapay Sinir Ağı Eğitiminde Global Optimallik' makalesiyle konveks olmayan yapay sinir ağları problemi üzerinde çalışmışlardır. Bazı koşullar altında yeterince geniş ağlar için herhangi bir lokal minimum noktasının bazı girdileri(inputs) sıfır ise bu noktanın aynı zamanda global minimum olduğunu göstermişlerdir (Haefele ve Vidal 2017). Kenji Kawaguchi ve arkadaşları derin öğrenmede tüm kötü yerel minimaların ortadan kaldırılması konusunda çalışma yapmışlar ve sinir ağlarına nöron ekleyerek lokal minimumların elendiğini ispatlamışlardır (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 'ü kadar olacağı ve son olarak girdi boyutunun iki katından küçük olması gerektiğinden bahsetmiştir (Heaton, 2008). Ariel Gordon (2018) derin ağların hızlı ve basit kaynak kısıtlı yapısı öğrenimi üzerine çalışmış ve performansı arttırmak adına yapay sinir ağlarının mimarisini otomatik olarak tasarlanması ile ilgili katkıda bulunmuşlardır.

3. MATERYAL ve YÖNTEM

Bu tez kapsamında Yapay Sinir Ağlarında(YSA) çok katmanlı sinir ağları, matematiksel anlamda incelenmiş ve kullanılan fonksiyonlar karşılaştırılmıştır. Bu tezde, yapılan literatür taramaları sonucunda sinir ağlarının en popüler iki tipi ele alınmıştır. Bunlar çok katmanlı ileri beslemeli sinir ağı (MLFFNN) ve tekrarlayan sinir ağı (RNN) olup tez materyalinde bu türlerin avantajları ve dezavantajları ile eğitim süreci yer almaktadır. Araştırmacıların çalışmalarında uygun YSA türünü seçmelerine ve belirlemelerine yardımcı olmak için iki YSA türünün avantajları ve dezavantajları bu tez kapsamında sunulmuştur. Ayrıca yöntem olarak aktivasyon fonksiyonları olan Sigmoid ve ReLu fonksiyonları Pycharm üzerinden Python kodları ile derin öğrenme teknikleri kullanılıp geliştirilmiştir. PyCharm, JetBrains tarafından geliştirilmiştir. PyCharm, kullanıcıların Python kodlarını yazmaları, düzenlemeleri, test etmeleri gibi seçenekler de sunmaktadır. PyCharm uygulaması üzerinden yapılan çalışmaların bir örneği ve kullanılan Python kodları kaynaklar bölümünde Kaggle üzerinden bir Url adresi şeklinde açık kaynak olarak paylaşılmıştır. Yapılacak olan uygulamada kullanılan YSA yapısı MLFFNN'lerin temel avantaj ve dezavantajları Çizelge 3.1'te özetlenmiştir.

Çizelge 3.1. MLFFNN Avantaj ve Dezavantajları

Yapay	Avantajları	Dezavantajları
ÇOK KATMANLI İLERİ BESLEMELİ YAPAY SINIR AĞI	• MLFFNN, birçok YSA uygulaması için kullanılan popüler YSA'lardan biridir (Fung, 2005) • MLFFNN birçok durumda iyi performans verir (Jeatrakul P, 2009). • Fonksiyon yaklaşım problemlerinde, düzgün tepe noktaları ve vadisi olmayan yüzeyler için genel bir model olarak tercih edilirler • MFFNN, çeşitli problem alanlarında kolaylıkla ve başarılı bir şekilde uygulanabilen sağlam bir sinir ağıdır	• Bu ağ türü, eğitim süreci için çok sayıda girdi ve hedef çifti gerektirir (Du K, 2014)

Çizelge 3.1’de görüldüğü üzere Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı (MLFFNN) birçok durumda iyi performans veren bir YSA tipidir. MLFFNN’nin en büyük dezavantajının ise çok sayıda giriş (input) verisi istemesi olduğu sonucuna varılmıştır.

Dünya Sağlık Örgütü, kalp hastalıkları nedeniyle dünya çapında her yıl 12(on iki) milyon ölüm olduğunu tahmin etmektedir. Amerika Birleşik Devletleri ve diğer gelişmiş ülkelerdeki ölümlerin büyük çoğunluğu kalp ve damar hastalıklarından kaynaklanmaktadır. Kardiyovasküler hastalıkların yaşam tarzını değiştirmeleri ve hastalığın komplikasyonlarını azaltıcı etkenleri bilmeleri gerekmektedir.

Bu araştırma, çok katmanlı yapay sinir ağları kullanarak genel riski tahmin etmeyi ve amaçlamaktadır. Kullanılan veri seti, Kaggle web sitesinde açık kaynaktır ve Massachusetts, Framingham kasabasında yaşayan bireylerde devam eden bir kardiyovasküler çalışmadan alınmıştır. Sınıflandırma hedefi, hastanın devam eden yıllarda kardiyovasküler hastalık riskini taşıyıp taşımadığını tahmin etmektir. Veri seti içerisinde toplam 4238 kişinin 15 farklı özellik bilgileri yer almaktadır (Kaggle, 2019).

Bu çalışmada derin öğrenme modelleri için bilinen en popüler kütüphane olan tensorflow kütüphanesinden Sequential modeli, Dense, Dropout, BatchNormalization katmanları ve Adam optimizer çalışma ortamının içine aktarılmıştır. Sigmoid aktivasyon fonksiyonu Dense katmanı içerisine argüman olarak kullanılmıştır.

Veri seti incelendiğinde 645 adet boş değer olduğu gözlemlenmiş olup boş değere sahip kişiler veri seti içerisinden çıkarılmıştır. Veri setinin içerisinde boş değerler çıkarıldıktan sonra 3593 gözlem kalmıştır. Gözlemlerin hedef değişken olan on yıllık kardiyovasküler hastalık riski taşıyıp taşımadığı bilgisini 0 (sıfır) ve 1 (bir) olarak bize göstermektedir. Bu noktada veri seti içerisindeki gözlemlerin yaklaşık %84,8 'i 10 yıl için kardiyovasküler hastalığı riski taşımamaktadır.

Makine ve derin öğrenme modellerinin başarı oranının yüksek olması için veri seti dağılımının eşit olması tavsiye edilmektedir. Çalışma esnasında SMOTE örnekleme tekniği kullanılarak YSA modelleri incelenmiş olup modellerin aşırı öğrenme sorunu (overfitting) ile karşılaşılması sonucunda bu çalışmada hiçbir örnekleme tekniği kullanılmamıştır. Çalışma içerisinde sequential fonksiyonu kullanılarak model nesnesi tanımlanmakta olup model.add fonksiyonu ile birlikte birkaç katman eklenmektedir. Dense katmanları açık katmanlar olmayıp girdilerin önceki katmandaki bütün nöronlarla tam bağlantılı olduğu katmanlardır. Dropuot katmanları overfittingi önlemek için kullanılmaktadır ve batchNormalization katmanları ise öğrenme hızını artırmak, stabilizasyon sağlamak için kullanılmıştır. Model.compile fonksiyonu model ile derlendikten sonra model.fit fonksiyonu ile eğitilmiştir. Adam optimizer kullanılırken binary_crossentropy kaybı ve doğruluk metriği kullanılmıştır. Modelin eğitimi sırasında kullanılan veri örneğinin boyutunu belirten batch değeri 32 olarak ayarlanmış olup verinin her bir eğitim döngüsünü belirten epoch değeri 100 olarak ayarlanmıştır.

Bu çalışmada Sigmoid ve ReLU aktivasyon fonksiyonlarının her birisi için toplam 3, 5 ve 7 katmanlı modeller oluşturulmuştur. Sağlıklı bir karşılaştırma yapılabilmesi için her bir katmandaki sinir hücre sayısı aynı tutulmuş olup her bir katman arasına batch normalization işlemi ve %20 dropout kullanılarak model eğitilmiştir. Loss (yitim) fonksiyonu grafikleri ile modelin overfittliği kontrol edilmiş ve ROC eğrisinin altında kalan alan ile model doğrulukları yorumlanmıştır.

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

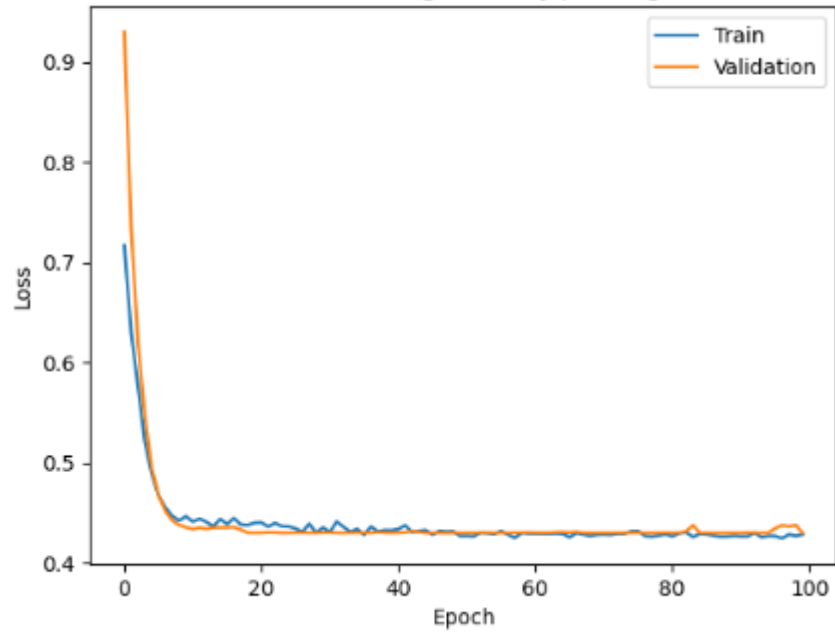
Bu bölümde çok katmanlı YSA'nın en çok kullanılan tiplerinden Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı (MLFFNN) tipi ele alınmıştır. Ayrıca tezin bu bölümünde Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı (MLFFNN) Sigmoid ve ReLU aktivasyon fonksiyonları ile birlikte uygulanmıştır.

Bölüm 1.2.1.1'de detayları ile anlatılan Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı (MLFFNN) icat edilen ilk YSA türüdür. Ek olarak, benzerleri olan Tekrarlayan Sinir Ağları (RNN)'lerden daha basittir. İleri Besleme olarak adlandırılmalarının sebebi bilginin ağda döngü olmadan önce giriş düğümleri, sonra gizli düğümler ve son olarak çıkış düğümleri aracılığıyla ileri doğru hareket etmesidir. MLFFNN genel olarak Levenberg-Marquardt (LM) algoritmasını kullanmaktadır. Levenberg-Marquardt güçlü bir teorik temeli olan ve önemli ölçüde hızlı yakınsama sağlayan ikinci dereceden optimizasyon tekniklerinin bir türüdür ve Newton Yöntemine bir yaklaşım olarak kabul edilir (Du K, 2014). LM öğrenimi, klasik Newton yönteminin hızlı öğrenme hızı ile gradyan inişinin garantili yakınsaması arasında değiş tokuş yapabildiği için kullanılır (Singh A, 2017). LM algoritması her zaman büyük veri setlerine uygundur ve diğer eğitim algoritmalarına göre daha az iterasyonda ve daha kısa sürede yakınsamaktadır. MLFFNN'lerin temel avantajı, zor ve karmaşık problemler için kullanılabilmesidir ancak bazen uzun eğitim sürelerine ihtiyaç duyulabilir.

4.1. Sigmoid Fonksiyonu Üzerine Bir Uygulama

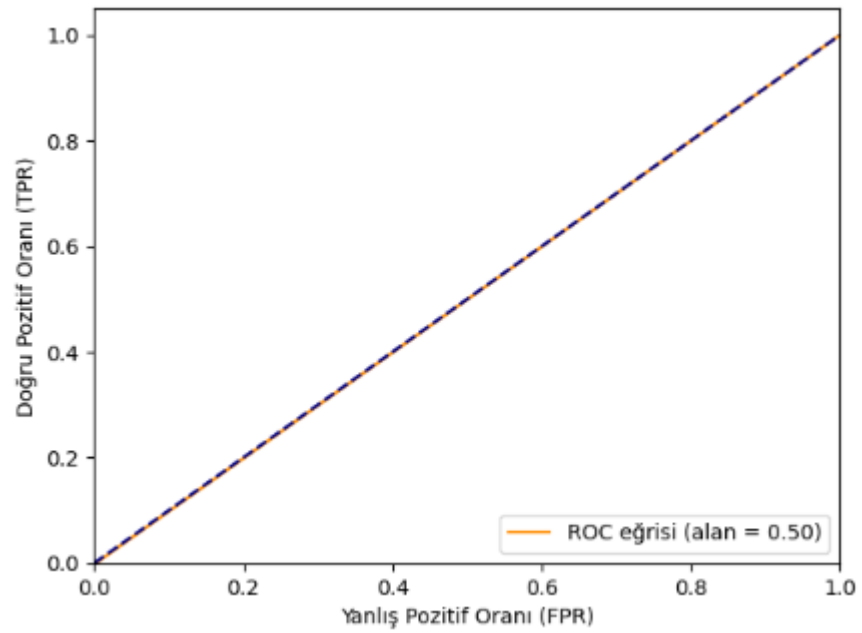
Yöntem bölümünde bahsedildiği üzere Sigmoid aktivasyon fonksiyonu için 3, 5 ve 7 katmanlı modeller oluşturulmuştur. Bu modeller bir giriş katmanı, bir çıkış katmanı ve arasında kalan gizli katmanların olduğu modeller tasarlanmıştır.

İlk modelin giriş, çıkış ve ara katmanları Sigmoid aktivasyon fonksiyonu kullanılarak ve nöron sayısı giriş katmanı için 4, gizli katman için 2 ve çıkış katmanı için 1 olarak düzenlenmiştir. Bunun sonucunda model %84,5 test doğruluğuna sahip olmakla birlikte yitim grafiği aşağıda gösterilip yorumlanmıştır.



Şekil 4.1. Üç Katmanlı YSA Modeli Sigmoid Yitim Grafiği

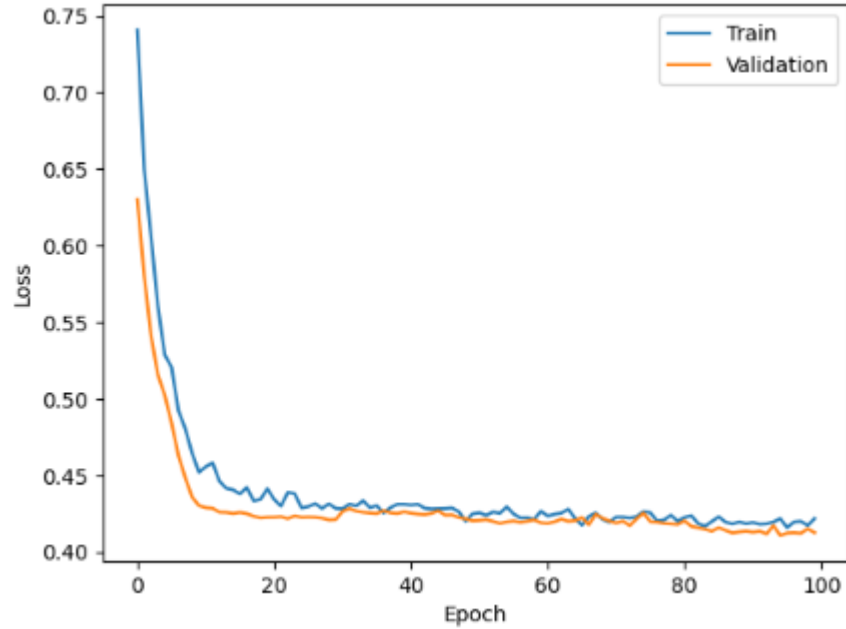
Modelin loss yani yitim değeri 0,4 ile 0,5 değerleri arasındadır. Bu aralık modelin tahmin değerleri ve gerçek değerler arasındaki farkı göstermektedir. Test veri setinin yitim oranının eğitim veri setinin yitim oranı düşerken yükselmesi overfitting işareti olarak yorumlanabilmektedir. Modelin doğruluk oranı Şekil 4.2’de incelenmiştir.



Şekil 4.2. Üç Katmanlı YSA Modeli Sigmoid ROC/AUC Grafiği

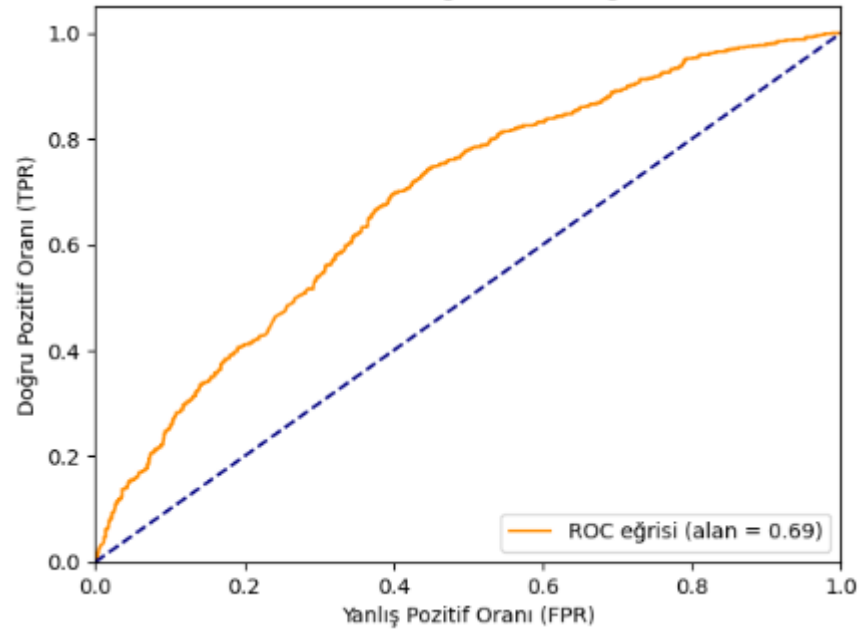
ROC eğrisinin sol üst köşeye ne kadar yakın olursa modelin doğruluğunun o kadar yüksek olduğu varsayılır. ROC eğrisinin altında kalan alan ise Area Under the Curve (Eğri altındaki alan) AUC olarak adlandırılır. AUC değerinin 0,5 olması modelin tamamen rastgele bir sınıflandırma yaptığının göstergesidir.

İkinci modelde giriş, çıkış ve 3 ara katmanı Sigmoid aktivasyon fonksiyonu kullanılarak ve nöron sayısı giriş katmanı için 16, birinci gizli katman için 8, ikinci gizli katman için 4, üçüncü gizli katman için 2 ve çıkış katmanı için 1 olarak düzenlenmiştir. Bunun sonucunda model %84,5 test doğruluğuna sahip olmakla birlikte yitim grafiği aşağıda gösterilip yorumlanmıştır.



Şekil 4.3. Beş Katmanlı YSA Modeli Sigmoid Yitim Grafiği

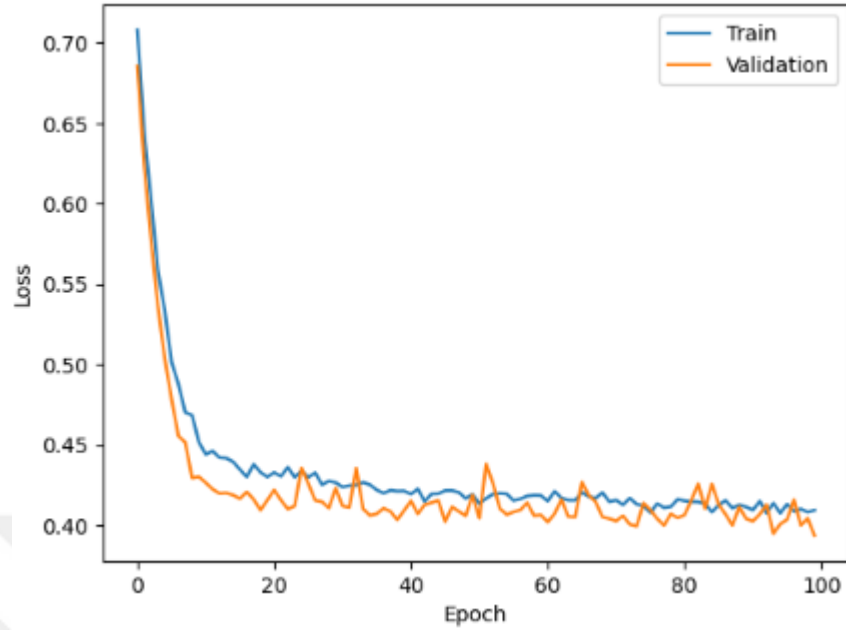
Şekil 4.3'te test doğruluğunun yitim değerinin eğitim veri seti yitim değerinden düşük olması modelin beklenmeyen değişimlere karşı başarılı olduğunu gösterme ihtimali taşırken değerlerin 0,40-0,45 aralığında olması ve stabil olmaması modelin yüksek başarı oranlarına ulaşmadığının bir göstergesi olabilmektedir. Fikri desteklemek adına modelin ROC/AUC grafiği aşağıda incelenmiştir.



Şekil 4.4. Beş Katmanlı YSA Modeli Sigmoid ROC/AUC Grafiği

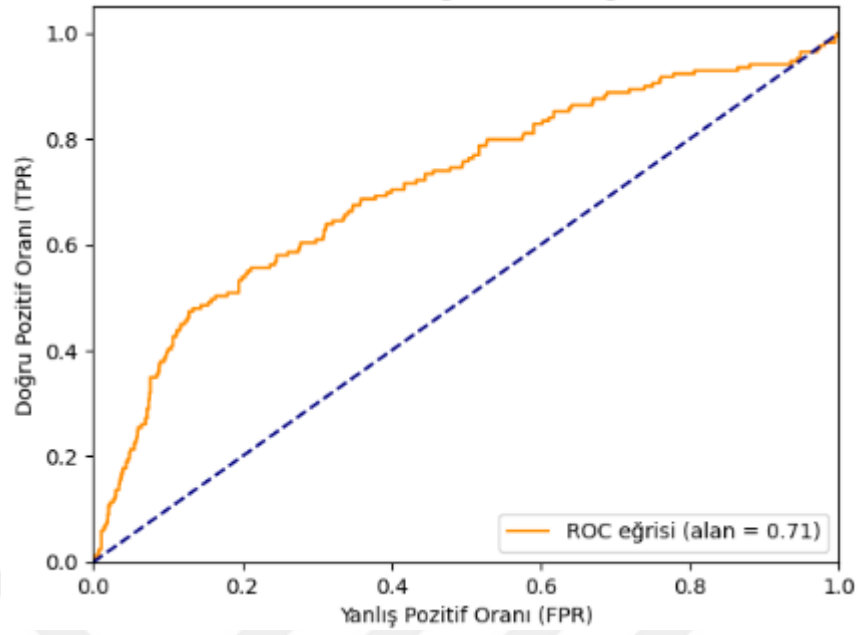
5 katmanlı modelin 3 katmanlı modele kıyasla sol üst köşeye yaklaşması ve ROC eğrisi altında kalan alanın 0,69 olması, katman sayısı arttıkça modelin doğruluğunun arttığının bir göstergesidir.

Üçüncü modelde giriş-çıkış ve 5 ara katmanı Sigmoid aktivasyon fonksiyonu kullanılarak ve nöron sayısı giriş katmanı için 64, birinci gizli katman için 32, ikinci gizli katman için 16, üçüncü gizli katman için 8, dördüncü gizli katman için 4, beşinci gizli katman için 2 ve çıkış katmanı için 1 olarak düzenlenmiştir. Bunun sonucunda model %84,5 test doğruluğuna sahip olmakla birlikte loss (yitim) grafiği aşağıda gösterilip yorumlanmıştır.



Şekil 4.5. Yedi Katmanlı YSA Modeli Sigmoid Yitim Grafiği

Şekil 4.5'te test doğruluğunda yitim değerinin eğitim veri seti yitiminden düşük olması modelin beklenmeyen değişimlere karşı başarılı olduğunu göstermekle birlikte bazı noktalarda test veri setinin kaybından daha yukarı çıkması modelin test veri seti için tamamen istikrarlı olmadığını göstermektedir. Değerlerin 0,40-0,45 aralığında olması ve stabil olmaması modelin yüksek başarı oranlarına ulaşmadığının bir göstergesi olabilmektedir. Fikri desteklemek adına modelin ROC/AUC grafiği aşağıda incelenmiştir.



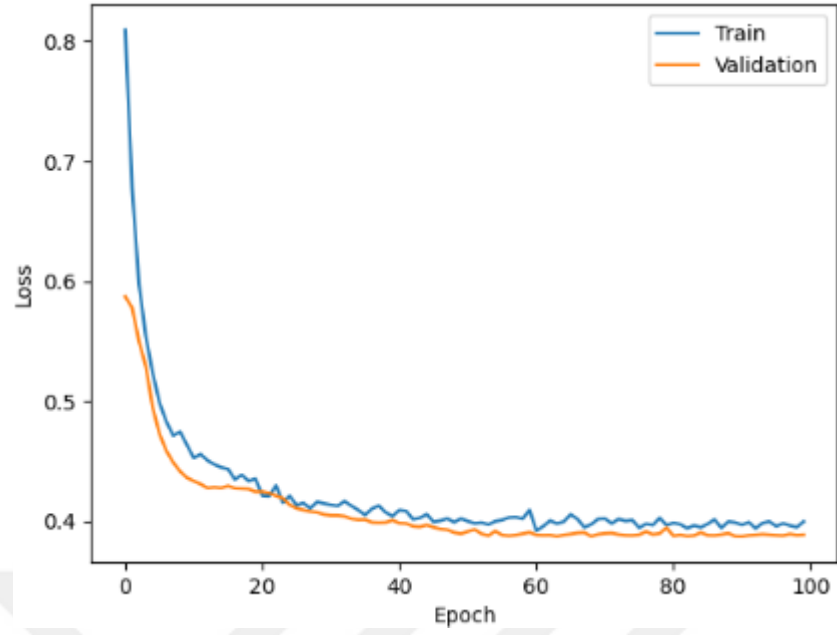
Şekil 4.6. Yedi Katmanlı YSA Modeli Sigmoid ROC/AUC Grafiği

Modelin loss (yitim) grafiğinin stabil olmamasına karşın ROC eğrisi altında kalan alan artış göstermiştir. 0,71 değeri ile birlikte Sigmoid aktivasyon fonksiyonu için katman sayısı arttıkça modelin doğruluk oranının arttığı desteklenmiştir.

4.2. ReLU Fonksiyonu Üzerine Bir Uygulama

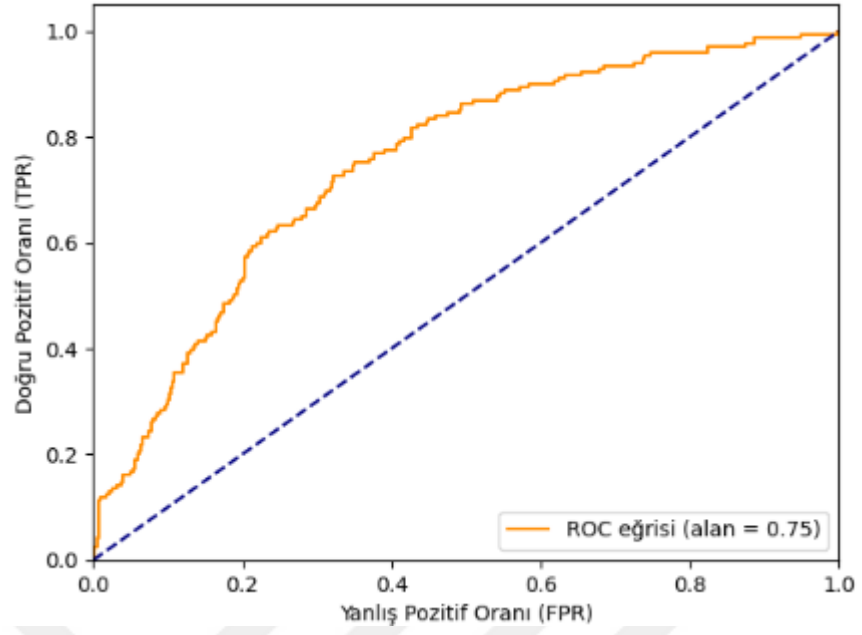
Yöntem bölümünde bahsedildiği üzere ReLU aktivasyon fonksiyonu için 3, 5 ve 7 katmanlı modeller oluşturulmuştur. Bu modeller bir giriş katmanı, bir çıkış katmanı ve arasında kalan gizli katmanların olduğu modeller tasarlanmıştır.

İlk modelin giriş ve ara katmanları ReLU aktivasyon fonksiyonu, çıkış katmanı ise ikili sınıflandırma olduğu için Sigmoid aktivasyon fonksiyonu kullanılmıştır. Nöron sayısı giriş katmanı için 4, gizli katman için 2 ve çıkış katmanı için 1 olarak düzenlenmiştir. Bunun sonucunda model %84,5 test doğruluğuna sahip olmakla birlikte yitim grafiği aşağıda gösterilip yorumlanmıştır.



Şekil 4.7. Üç Katmanlı YSA Modeli ReLU Yitim Grafiği

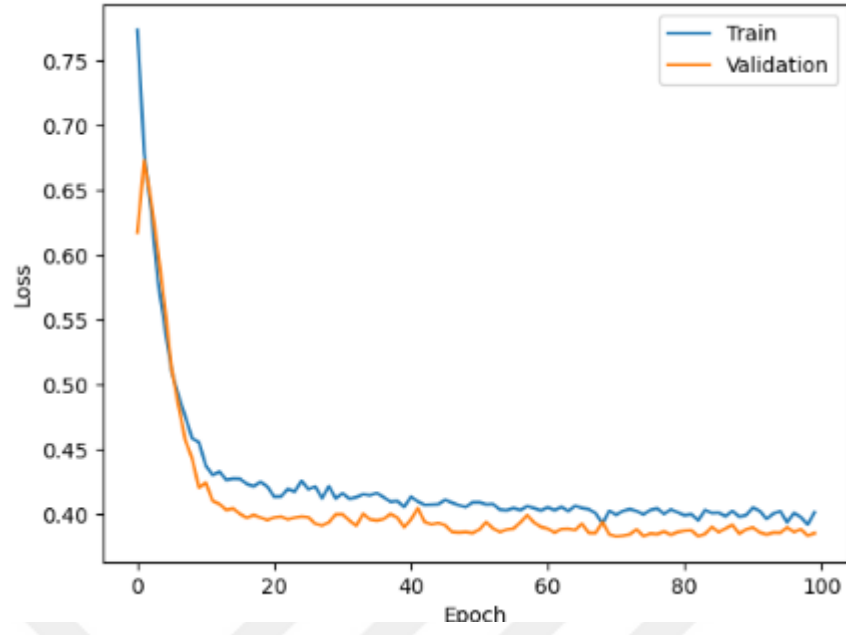
Şekil 4.7’de loss (yitim) grafiği incelendiğinde yitim değerlerinin aralıklarının 0,4’e yaklaştığı görülmekle birlikte 3 katmanlı Sigmoid aktivasyon fonksiyonu kullanılan modele kıyasla daha düşük bir değer aralığındadır. Bu aralık modelin tahmin değerleri ve gerçek değerler arasındaki farkı göstermektedir. Test veri setinin yitim oranının eğitim veri setinin yitim oranının düzenli olarak altında olması ve eğitim kayıpları değişkenlik gösterirken test kayıplarının stabil kalması modelin farklı verilere hazırlıklı olduğunun işareti olarak yorumlanabilmektedir. Modelin doğruluk oranı Şekil 4.8’de incelenmiştir.



Şekil 4.8. Üç Katmanlı YSA Modeli ReLU ROC/AUC Grafiği

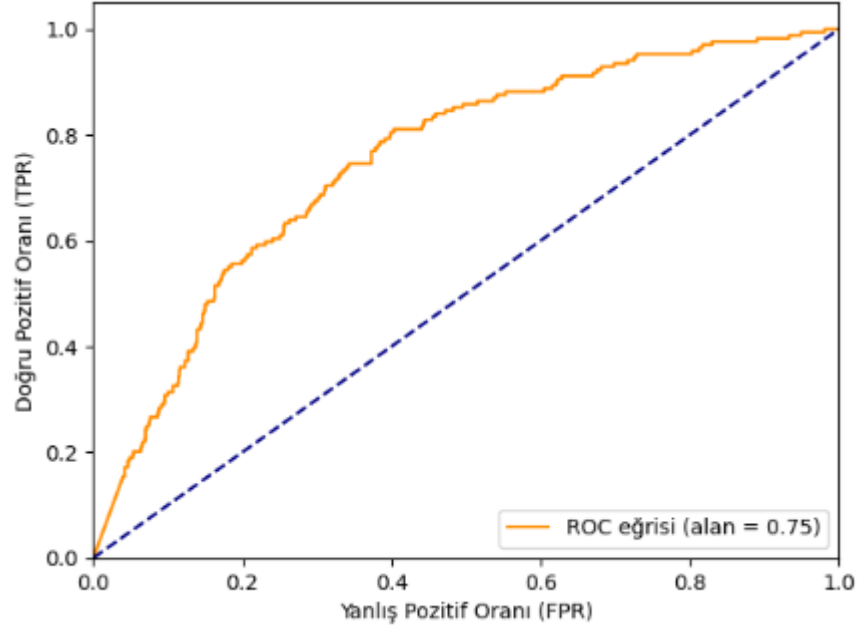
ROC eğrisi Sigmoid aktivasyon fonksiyonu kullanılan modellere kıyasla sol üst köşeye daha yakın olması ve altında kalan alanın 0,75 ile şimdiye kadar olan en yüksek oran olması ReLU aktivasyon fonksiyonunun sadece üç katman ile Sigmoid aktivasyon fonksiyonundan daha başarılı ve yüksek olduğu gösterilmektedir.

İkinci modelde giriş ve 3 ara katmanı ReLU aktivasyon fonksiyonu, çıkış katmanı Sigmoid fonksiyonu kullanılarak ve nöron sayısı giriş katmanı için 16, birinci gizli katman için 8, ikinci gizli katman için 4, üçüncü gizli katman için 2 ve çıkış katmanı için 1 olarak düzenlenmiştir. Bunun sonucunda model %84,5 test doğruluğuna sahip olmakla birlikte kayıp grafiği aşağıda gösterilip yorumlanmıştır.



Şekil 4.9. Beş Katmanlı YSA Modeli ReLU Yitim Grafiği

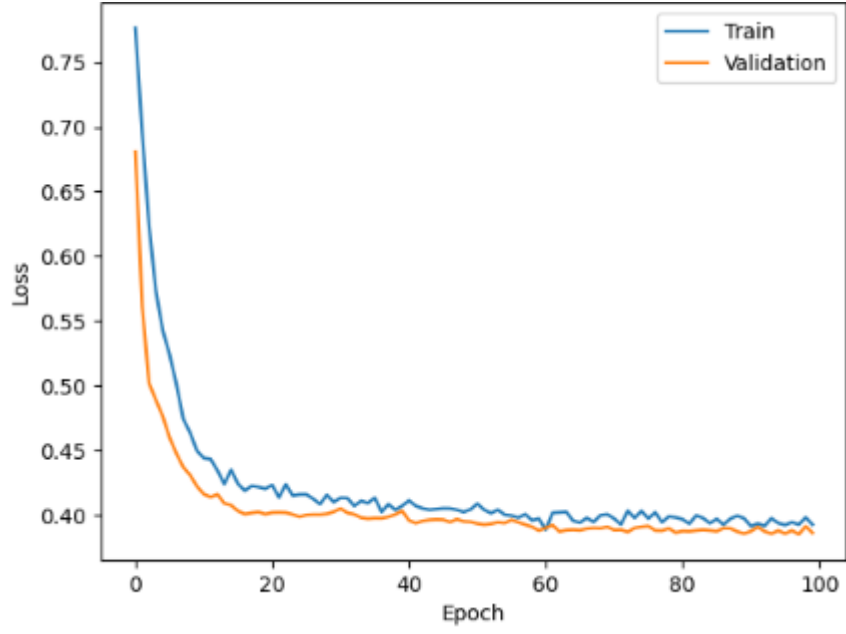
Şekil 4.9’da test doğrulundaki yitimin eğitim veri seti yitiminden düşük olması modelin beklenmeyen değişimlere karşı başarılı olduğunu gösterme ihtimali taşıırken test yitim değerlerinin 0,4 ve altında olması ve stabil olmaması modelin önceki modellere göre daha başarılı olması ancak yine de yüksek başarı oranlarına ulaşmadığının bir göstergesidir. Fikri desteklemek adına modelin ROC/AUC grafiği aşağıda incelenmiştir.



Şekil 4.10. Beş Katmanlı YSA Modeli ReLU ROC/AUC Grafiği

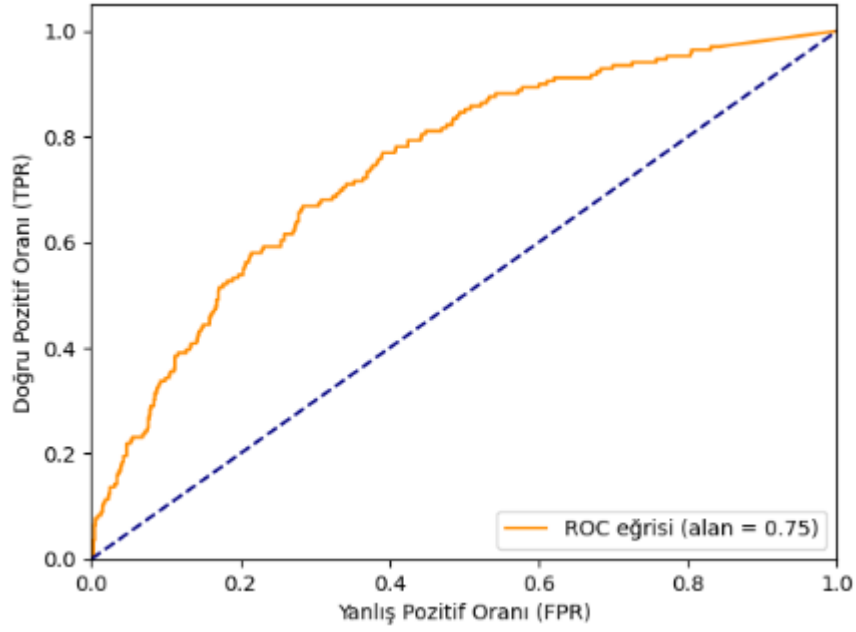
5 katmanlı modelin 3 katmanlı model ile aynı alan değerine sahip olması modelin başarısının korunduğunu göstermektedir. ReLU ve Sigmoid fonksiyonlarının sağlıklı bir şekilde karşılaştırılabilmesi için katmanlardaki nöron sayısı optimize edilmediği için ROC değerinin artmadığı düşünülebilir. Optimizasyon yapılması halinde alan yani doğruluk oranının artması öngörülebilir. 0,75 değeri ile Sigmoid aktivasyon fonksiyonuna nazaran ReLU aktivasyon fonksiyonu aynı katman sayısı için Sigmoid aktivasyon fonksiyonundan daha başarılı olduğu gözlemlenmiştir.

Üçüncü modelde giriş ve 5 ara katmanda ReLU, çıkış kısmında Sigmoid aktivasyon fonksiyonu kullanılarak ve nöron sayısı giriş katmanı için 64, birinci gizli katman için 32, ikinci gizli katman için 16, üçüncü gizli katman için 8, dördüncü gizli katman için 4, beşinci gizli katman için 2 ve çıkış katmanı için 1 olarak düzenlenmiştir. Bunun sonucunda model %84,5 test doğruluğuna sahip olmakla birlikte loss (yitim) grafiği aşağıda gösterilerek yorumlanmıştır.



Şekil 4.11. Yedi Katmanlı YSA Modeli ReLU Yitim Grafiği

Şekil 4.11’de test doğruluğundaki yitimin eğitim veri yitiminin altında olması ve son döngülerde eğitim kaybındaki beklenmeyen değişiklikleri desteklemesi çalışmanın şimdiye kadar çalışılan tüm modellere kıyasla daha başarılı olduğu söylenebilir. Değerlerin 0,4 civarında olması ve stabil olmaması modelin yüksek başarı oranlarına ulaşmadığının bir göstergesi olabilmektedir. Fikri desteklemek adına modelin ROC/AUC grafiği aşağıda incelenmiştir.



Şekil 4.12. Yedi Katmanlı YSA Modeli ReLU ROC/AUC Grafiği

Modelin loss (yitim) grafiğinin stabil olmamasına karşın ROC eğrisi altında kalan alan artış göstermiştir. 0,75 değeri ile Sigmoid aktivasyon fonksiyonuna nazaran ReLU aktivasyon fonksiyonu aynı katman sayısı için Sigmoid aktivasyon fonksiyonundan daha başarılı olduğu gözlemlenmiştir.

5. SONUÇ ve ÖNERİLER

Tez kapsamında matematiksel modellerin topolojisi incelendiğinde yapay sinir ağının iyi eğitilebilmesi için kullanılan matematiksel modellerin doğru seçilmesi gerektiği sonucuna ulaşılmıştır. Bu çalışmada yapılan literatür taraması sonucunda YSA'da kullanılan matematiksel modellerin genel olarak Hebb yasasına ve onun genişletilmiş modeline dayandığı görülmüştür.

Aktivasyon fonksiyonlarının incelendiğinde ise bu inceleme sonucunda en yaygın kullanılan aktivasyon fonksiyonlarının Sigmoid, ReLU, Softmax fonksiyonları olduğu gözlemlenmiştir. İkili çıkış sonucunun elde edildiği durumlarda Sigmoid fonksiyonu kullanılırken çoklu sınıflandırma gerektiren problemlerde Softmax fonksiyonunun her bir sınıfa ait olasılıkları kolayca yorumlanabilen değerleri atamayı kolaylaştırdığı için tercih edilebilir.

Bu çalışmada türevi daha çok değer alan fonksiyonların çok katmanlı sinir ağlarının eğitilmesi ve öğrenmenin gerçekleşmesi açısından daha başarılı sonuçlar verebileceği düşünülmektedir.

Bu bağlamda gradyanların yok olması problemine ilk çözüm Leaky ReLU olabilir. Derin öğrenme modelleri için ReLU ile denemelere başlanması tavsiye edilebilir. Çıkış katmanlarında genellikle sınıf sayısına göre Sigmoid veya Softmax kullanılabilir.

Pycharm uygulaması üzerinden yapılan çalışmalarda Sigmoid aktivasyon fonksiyonu ile ReLU aktivasyon fonksiyonu kullanılarak geliştirilen modeller karşılaştırıldığında, Sigmoid aktivasyon fonksiyonu için katman sayısı arttıkça modelin eğitimi, öğrenmesi ve başarısının arttığını loss (yitim) ve ROC/AUC grafik değerleri baz alınarak gözlemlenmiştir. ReLU aktivasyon fonksiyonun ise Sigmoid aktivasyon fonksiyonunun başarı oranının arttığı model yapılarıyla çalıştırıldığında, herhangi bir parametre optimizasyonu dahi yapılmadan daha fit grafikler ve daha yüksek başarı oranına ulaşıldığı sonucuna varılabilir.

Bu tezin konusu olan Çok Katmanlı YSA'ların matematiksel modellemeleri yapılırken katman sayısının ve aktivasyon fonksiyonunun öğrenme ve başarı üzerinde etkili olduğu gözlemlenmiştir. Ancak katman sayısının artması bir noktaya kadar modelin başarısını arttırdığı belli bir katman sayısından sonra ise modelin aşırı öğrenmeye yatkınlığını arttırdığı görülmüştür. Bu durum kullanılan veri setinin özellik sayısı, gözlem sayısı, karmaşıklık düzeyi ve hedef özellik sınıf sayısına göre değişkenlik gösterebilir.

Kullanılacak veri setine bağılı olarak katman ve nöron sayısı üzerinde optimizasyon yapılması gerektiğı sonucuna varılabilir.



KAYNAKLAR

- Ariel, G. and Elad, E., 2018. Fast and Simple Resource-Constrained Structure Learning of Deep Networks. **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. DOI:10.1109/cvpr.2018.00171.
- Badea, LM., 2014. Predicting Consumer Behavior with Artificial Neural Networks. 15(14): 238-246.
- Bayır, F., 2006. Yapay Sinir Ağları ve Tahmin Modellemesi Üzerine Bir Uygulama., **İstanbul Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi**.
- Cho, J.S.H., 2005. Comparing the Performance of Different Neural Networks Architectures for the Prediction of Mineral Prospectivity. **International Conference on Machine Learning and Cybernetics**, 394-398.
- Ciodaro, T., Deva, D., de Seixas, J., 2012. Online particle detection with neural networks based on topological calorimetry information. **J. Phys. Conf. Series** 368, 012030
- Collobert, R., 2011. Natural language processing (almost) from scratch. **J. Mach. Learn. Res.** 12, 2493–2537
- Du K-L, Swamy MNS., 2013. **Neural Networks and Statistical Learning**. Springer,
- Er, O., Yumusak, N., Temurtas F., 2010. Chest diseases diagnosis using artificial neural networks. **Expert Syst Appl.** 37(12): 7648-7655.
- Fadzil, MHA, Bakar HA., 1994. Human face recognition using neural networks. in **Proceedings of 1st International Conference on Image Processing**.
- Falat, L., Pancikova L., 2015. Quantitative Modelling in Economics with Advanced Artificial Neural Networks. **Procedia Econ Financ.** 34: 194-201.
- Fukuoka, Y., 2002. Artificial Neural Networks in Medical Diagnosis. in **Computational Intelligence Processing in Medical Diagnosis Studies in Fuzziness and Soft Computing**, S. M., T. HN., J. A., J. A., J. S., and J. L.C., Eds. 197-228.
- Fung, CC., Iyer, V., Brown, W., Wong, KW., and Almeida, M.R.H., 2003. Study of the influence of the polyalcohol sorbitol on the electrodeposition of copper–zinc films from a non-cyanide bath. **Journal of Electroanalytical Chemistry**, 562, 153–159.
- Farabet, C., Couprie, C., Najman, L., LeCun., 2013. Learning hierarchical features for scene labeling. **IEEE Trans. Pattern Anal. Mach. Intell.** 35, 1915–1929
- Glorot, X., Bordes, A. and Bengio, Y., 2011. Deep sparse rectifier neural networks. In Proc. 14th **International Conference on Artificial Intelligence and Statistics** 315–323
- Byrd., H, Hansen., S. L, Nocedal, J. and Singer., Y., 2016. “A stochastic quasi-newton method for large-scale optimization,” **SIAM Journal on Optimization**, vol. 26, no. 2, pp. 1008–1031,
- Haeffele, B. D., and Vidal, R., 2017. Global optimality in neural network training. In Proceedings of the **IEEE Conference on Computer Vision and Pattern Recognition** (pp. 7331-7339).
- Haykin, S., 2009. Neural Networks and Learning Machines, **Third Edit. Pearson**, 1351-1358,
- Haykin, S., 1998. Neural Networks: **A Comprehensive Foundation, Second Ed., Prentice hall, New Jersey**, s. 23
- Heaton, J., 2008. Introduction to Neural Networks for C#, **Second Edition**. Heaton

- Hebb, D., 2011. The Organization of Behavior. NewYork: Wiley. Collobert, R., et al. Natural language processing (almost) from scratch. **J. Mach. Learn. Res.** 12, 2493–2537
- Helmstaedter, M., 2013. Connectomic reconstruction of the inner plexiform layer in the mouse retina. **Nature** 500, 168–174
- Hinton, G., 2017. Deep neural networks for acoustic modeling in speech recognition. **IEEE Signal Processing Magazine** 29, 82–97
- Ito, M., Noda K, Hoshino Y, Tani J., 2006. Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. **Neural Networks** 19(3): 323-337,
- Öztemel, E., 2003. Yapay Sinir Ağları, **Papatya Yayınları**. İstanbul.
- Öztemel, E., 2006. Yapay Sinir Ağları, **Papatya Yayıncılık**. İstanbul.
- Öztemel, E., 2012. Yapay Sinir Ağları. İstanbul: **Papatya Yayıncılık**.
- Jain, V., 2019. Everything You Need to Know About “Activation functions” in Deep Learning Models. **Medium**.
- Jean, S., Cho, K., Memisevic, R. and Bengio, Y., 2015. On using very large target vocabulary for neural machine translation. In **Proc. ACL-IJCNLP** <http://arxiv.org/abs/1412.2007>
- Jeatrakul, P., Wong KW., 2009. Comparing the performance of different neural networks for binary classification problems. in 2009 8th **International Symposium on Natural Language Processing**, 111-115.
- Simonyan K. and Zisserman, A., 2014. “**Very deep convolutional networks for large-scale image recognition**,” CoRR, vol. abs/1409.1556, 2014.
- Kaggle. **Higgs boson machine learning challenge**. Kaggle <https://www.kaggle.com/c/higgs-boson> (2014).
- Kaggle. 2019. Kalp hastalığını tahmin etmek için veri kümesi hakkında bilgi Kaggle: <https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression> adresinden alındı.
- Kaggle. 2023. Kodlar için Kaggle: <https://www.kaggle.com/code/enisgde/relusigmoid>
- Karakaya, K., 2012. Yapay Zeka Modelleriyle Tam Ölçekli Çamur Çürütme Reaktörlerinden Elde Edilen Biyogaz Üretiminin Tahmini: Yapay Sinir Ağları ve Fuzzy Logic Uygulamaları., **Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi**.
- Kızrak, A. 2020, Ocak 7. Derin Öğrenme için Aktivasyon Fonksiyonlarının Karşılaştırılması. **Medium**.
- Kızrak, A. 2020, Ocak 7. Şekil.1.13. Örnek Yitim Hesabı Gösterimi. <http://cs231n.github.io/linear-classify/> **Medium**.
- Kızrak, A. 2020, Ocak 7. Şekil 1.12. Örnek sınıflar için temsil verileri <http://cs231n.github.io/linear-classify/> **Medium**.
- Kumawat, D., 2019. **7 types of activation functions in neural network**. Analytics Steps | Learn AI, Machine Learning, Business Analytics, Startups & other Technologies.
- Krizhevsky, I. Sutskever, and G. E. Hinton., 2012. “ImageNet classification with deep convolutional neural networks,” in *Proceedings of the 25th international conference on neural information processing systems - volume 1*, pp. 1097–1105.

- Yann, LC., Bengio, Y. and Hinton., G., 2015. "**Deep learning.**" **nature** 521.7553 436-444.
- Leung, M. K., Xiong, H. Y., Lee, L. J. and Frey, B. J., 2014. Deep learning of the tissue-regulated splicing code. **Bioinformatics** 30, i121–i129
- Liu, D., 2017. A Practical Guide to ReLU. **Medium**.
- Lu, S., Chung, JH., Velinsky, SA., 2005. Human-Robot Collision Detection and Identification Based on Wrist and Base Force / Torque Sensors. **IEEE International Conference on Robotics and Automation**, no. 3796-3801.
- Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E. and Svetnik, V., 2015. Deep neural nets as a method for quantitative structure-activity relationships. **J. Chem. Inf. Model.** **55**, 263–274
- Maass, W, Joshi P, Sontag ED. Computational aspects of feedback in neural circuits. **PLoS Comput Biol.** 3(1): 0015-0034
- Moreno-Escobar, JA., Gallegos-Funes FJ, Ponomaryov., 2007. VI. Rank M-type radial basis functions network for medical image processing applications. in **Image Processing: Algorithms and Systems V**, 6497: 1-12.
- Aran, O, Yildiz, O. T. and Alpaydin, E., 2009. "**an Incremental Framework Based on Cross-Validation for Estimating the Architecture**"
- Palaz, D., Magimai-Doss, M., Collobert, R., 2015. Convolutional Neural Networks-Based Continuous Speech Recognition Using Raw Speech Signal. in **2015 IEEE International**
- Passricha V, Aggarwal RK. Convolutional Neural Networks for Raw Speech Recognition. in **From Natural to Artificial Intelligence - Algorithms and Applications**, 2018; 21-40
- Research, 428s, United States. Kawaguchi, K., Kaelbling, L., 2020. Elimination of all bad local minima in deep learning. In **International Conference on Artificial Intelligence and Statistics** (pp. 853-863). PMLR.
- Sainath, T., 2013. Mohamed, A.-R., Kingsbury, B. & Ramabhadran, B. Deep convolutional neural networks for LVCSR. In **Proc. Acoustics, Speech and Signal Processing** 8614–8618
- Sevim, C., 2012. Öncü Göstergeler Yaklaşımına Göre Finansal Krizler ve Türkiye Örneği, **BDDK Kitapları**. Ankara
- Sharkawy, A-N, Aspragathos N., 2018. Human-Robot Collision Detection Based on Neural Networks. **Int J Mech Eng Robot Res**, 7(2): 150-157.
- Sharkawy, A-N, Koustoumpardis PN, Aspragathos, N., 2020. Neural Network based Approach for Variable Admittance Control in Human- Robot Cooperation: Online Adjustment of the Virtual Inertia. **Intell Serv Robot** 1-37:
- Sharkawy, A-N, Koustoumpardis, PN., Aspragathos N., 2018. Manipulator Collision Detection and Collided Link Identification based on Neural Networks. in **Advances in Service and Industrial Robotics RAAD 2018 Mechanisms and Machine Science**, pp. 3-12.
- Siegelmann, HT., Sontag, ED., 1991. Turing computability with neural nets. **Appl Math Lett.** 4(6): 77-80.

- Sutskever, I., Vinyals, O., and Le, Q. V., 2014. Sequence to sequence learning with neural networks. In Proc. **Advances in Neural Information Processing Systems** 27 3104–3112
- Szandała, T., 2021. Review and comparison of commonly used activation functions for deep neural networks. In Bio-inspired neurocomputing (pp. 203-224). Springer, Singapore. Haykin S., **Neural Networks and Learning Machines, Third Edit. Pearson**, 1351-1358, 2009.
- Tompson, J., Jain, A., LeCun, Y. and Bregler, C., 2014. Joint training of a convolutional network and a graphical model for human pose estimation. In Proc. **Advances in Neural Information Processing Systems** 27 1799–1807
- Yavuz, M., 2012. Yapay Sinir Ağları ile Portföy Optimizasyonu. **Balıkesir Üniversitesi Fen bilimleri Enstitüsü Yüksek Lisans Tezi**
- Zhao, ZQ., Huang, DS., Sun, BY., 2004. Human face recognition based on multi-features using neural networks committee. **Pattern Recognit Lett.** 25(12): 1351-1358
- Akça, F., 2020. <https://tinyurl.com/4wzajuaw> adresinden alındı.
- Kızrak, A., 2019. <https://tinyurl.com/26375f7c> adresinden alındı.
- GitHub, 2013. <https://cs231n.github.io/linear-classify> adresinden alındı.

ÖZGEÇMİŞ

İlkokul, ortaokul ve lise öğrenimini Antakya’da tamamladıktan sonra Hatay’ın çeşitli ilçelerinde ve Nevşehir’de matematik öğretmeni olarak görev yaptım. 2009-2011 yılları arasında Cumhuriyet Üniversitesi Fen Fakültesi’nde Matematik bölümü okudum. Daha sonra 2011 yılında Mustafa Kemal Üniversitesi Fen Edebiyat Fakültesi Matematik Bölümü’ne geçiş yaparak 2013 yılında mezun oldum. 2014 yılından bu yana aktif olarak ortaöğretim ve lise kurumlarında matematik öğretmenliği yapıyorum. 2020 yılının eylül ayında Kumlu Abdurrahman Gülizar Karadeniz Çok Programlı Anadolu Lisesi’ne Lise Matematik öğretmeni olarak atandım. Hala atandığım kurumda matematik öğretmeni olarak görev yapmaktayım.

