

**İSTATİSTİKSEL VE MAKİNE ÖĞRENME İLE DERİN SİNİR
AĞLARINDA HİPER-PARAMETRE SEÇİMİ İÇİN MELEZ
YAKLAŞIM**

**HYBRID APPROACH FOR HYPER-PARAMETER
SELECTION IN DEEP NEURAL NETWORKS WITH
STATISTICAL AND MACHINE LEARNING**

CANSU DOĞAN

PROF. DR. ÇAĞDAŞ HAKAN ALADAĞ

Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
İstatistik Anabilim Dalı için Öngördüğü
YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

Ankara, 2021

*Sevgili annem Nuray Dođan
ve babam Alpaslan Dođan'a ithafen,*

ÖZET

İSTATİSTİKSEL VE MAKİNE ÖĞRENME İLE DERİN SİNİR AĞLARINDA HİPER-PARAMETRE SEÇİMİ İÇİN MELEZ YAKLAŞIM

Cansu DOĞAN

Yüksek Lisans, İstatistik Bölümü

Tez Danışmanı: Prof. Dr. Çağdaş Hakan Aladağ

2021, 54 sayfa

Literatürde, derin sinir ağlarında çok önemli bir problem hiper-parametre seçimidir. Veri analizinde kullanılacak hiper-parametrelerin seçimi için denenebilecek çok fazla kombinasyon vardır. Özellikle büyük veri söz konusu olduğunda, olası tüm kombinasyonların denenmesi mümkün değildir. Literatürde belirtilen seçimin yapılabilmesi için genel kurallar bulunmamasına rağmen, önerilen ve kullanılan bazı yaklaşımlar mevcuttur. Buna karşın, YSA veri odaklı bir yöntem olduğu için belli bir veride kullanılan bir yaklaşımı başka bir veri için kullanmak çoğu zaman mümkün ve uygulanabilir olmamaktadır.

Aladağ 2019 yılındaki çalışmasında YSA'nın en önemli problemlerinden biri olan mimari seçim probleminin çözümü için istatistiksel ve makine öğrenmesine dayanan bir melez yaklaşım önermiştir. Belirtilen yaklaşımda, YSA'nın mimarileri ile öngörü performansı arasındaki ilişkinin yapısı basit doğrusal regresyon modeli kullanılarak analiz edilebilmektedir. Bu sayede, mimari seçimi istatistiksel olarak gerçekleştirilmekte ve mimaride kullanılan nöron sayıları ve öngörü performansı arasındaki ilişki yapısı da ortaya konulabilmektedir. Aladağ tarafından önerilen yaklaşım, çok önemli iki avantaj sağlamaktadır. Öncelikle analiz edilen veriye özgü olarak mimari seçimi istatistiksel ve sistematik olarak yapılabilmektedir. İkinci olarak, mimari modeli ile performans arasındaki ilişki yapısı modeli elde edildiğinden, birçok mimarinin performansı çözümlene yapılmadan bile bilenebilmektedir. Gerçekleştirilen tez

alışmasında da, Aladağ 2019 tarafından YSA mimari seçimi için önerilen yaklaşım, derin sinir ağlarında hiper-parametre seçimi için genişletilmiştir ve ayrıca hem doğrusal regresyon analizi hem de doğrusal olmayan regresyon analizi uygulanarak kullanılmıştır. Belirtilen melez yaklaşım diyabet hastalığı verisine uygulanmıştır. Diyabet verisi uygulaması sonucunda elde edilen bulgulara göre genişletilen melez yaklaşımın oldukça iyi sonuçlar ürettiği gözlemlenmiştir.

Anahtar Kelimeler: derin sinir ağları, diyabet verisi, doğrusal regresyon analizi, doğrusal olmayan regresyon analizi, hiper-parametre seçimi, istatistiksel öğrenme, makine öğrenme, sınıflandırma.

Danışman: Prof. Dr. Çağdaş Hakan Aladağ, Hacettepe Üniversitesi, İstatistik Bölümü, Yöneylem Araştırması Anabilim Dalı.

ABSTRACT

HYBRID APPROACH FOR HYPER-PARAMETER SELECTION IN DEEP NEURAL NETWORKS WITH STATISTICAL AND MACHINE LEARNING

Cansu DOĞAN

Master's Thesis, Department of Statistics

Supervisor: Prof. Dr. Çağdaş Hakan Aladağ

2021, 54 pages

In the literature, a very important problem in deep neural networks is the determination of hyper-parameters. There are many combinations that can be tried for the selection of hyper-parameters to be used in data analysis. It is not possible to try all possible combinations, especially when it comes to big data. Although there are no general rules for making the selection specified in the literature, there are some suggested and used approaches. On the other hand, since ANN is a data-driven method, it is often not possible or applicable to use an approach, which is introduced for a specific data, for another data.

In 2019, Aladag proposed a hybrid approach based on statistical and machine learning for the solution of the architectural selection problem in ANN. This issue is one of the most important problems of ANN. In his hybrid approach, the structure of the relationship between ANN architectures and predictive performance can be analyzed using a simple linear regression model. In this way, the determination of architecture is performed statistically and the relationship structure between the number of neurons used in the architecture and the prediction performance can be revealed. The approach proposed by Aladag provides two very important advantages. First of all, architecture selection can be done statistically and systematically according to the analyzed data. Secondly, the performance of many architectures can be statistically predicted even without using these architectures since the relationship structure model between the architectures and the prediction performance is obtained. In the thesis study,

the hybrid approach proposed by Aladag 2019 for ANN architecture selection was used by applying both linear and nonlinear regression analysis to the hyper-parameter selection in deep neural networks. The hybrid approach was applied to the diabetes (diabetes.csv) data. As a result of the application, it has been observed that the hybrid approach gives very good results.

Keywords: hyper-parameter selection, linear regression analysis, non-linear regression, artificial neural network, deep neural network, classification, diabetes data

Advisor: Prof. Dr. Çağdaş Hakan Aladağ, Hacettepe University, Department of Statistics, Operation Research Section.

TEŐEKKÜR

Danışmanım Prof. Dr. Çağdaş Hakan Aladağ'a, önerileri, tez çalışmamda yol gösterdiği ve yönlendirdiği için, her zaman bana inandığı için ve her karşılaştığım zorluklarda bana verdiği güç için,

İstatistik bölümünde bulunan bütün hocalarıma benim bugüne gelmemde buldukları katkıları ve sağladıkları manevi katkıları için,

Ve benim en büyük servetim, sevgili ve değerli ailem, ablam Seda Doğan, abim İsmail Doğan, yeğenim Ece Doğan, babam Alpaslan Doğan, annem Nuray Doğan ve sevgili Güven Aycı'ya sağladıkları eşsiz destekleri ve her zaman yanımda oldukları için, her zorlukta bana umut ve güç verdikleri için ve buraya gelmemi sağlayan çok değerli emekleri için,

tüm içtenliğimle teşekkürlerimi sunarım.

İçindekiler Tablosu

1. GİRİŞ	1
2. YAPAY ZEKÂ, İSTATİSTİKSEL ÖĞRENME VE MAKİNE ÖĞRENME	4
3. YAPAY SİNİR AĞLARI (YSA)	6
3.1. YSA'nın Yapısı ve Temel Özellikleri	10
3.2. YSA'da Öğrenme	19
3.3. YSA ile Sınıflandırma	22
3.4. YSA Çeşitleri	23
3.5. Parametre ve Hiper-parametreler (Parameter and Hyperparameters)	31
4. DOĞRUSAL VE DOĞRUSAL OLMAYAN REGRESYON ANALİZİ	34
5. MİMARİ SEÇİMİ İÇİN ALADAĞ (2019) TARAFINDAN ÖNERİLEN MELEZ YAKLAŞIM	36
6. UYGULAMA	38
7. SONUÇ	49
8. KAYNAKÇA	51

Şekiller Dizini

Şekil1. 1. Biyolojik Sinir Hücresinin Yapısı.....	7
Şekil1. 2. YSA'nın Genel İşleyişi.....	11
Şekil1. 3. Tek Çıktı Nörona Sahip İleri Beslemeli YSA.....	12
Şekil1. 4. Adımsal Fonksiyon Grafiği.....	15
Şekil1. 5.Doğrusal Fonksiyon Grafiği.....	15
Şekil1. 6.Sigmoid Fonksiyon Grafiği.....	16
Şekil1. 7. Hiperbolik Tanjant Fonksiyonu Grafiği.....	17
Şekil1. 8.Düzeltilmiş Doğrusal Birim Grafiği (ReLU).....	18
Şekil1. 9.Sızıntı ReLU Fonksiyon Grafiği.....	18
Şekil1. 10. Dereceli Alçalma Optimal Arama Görseli.....	21
Şekil1. 11. Dereceli Alçalma Eğim ile İfadesi.....	21
Şekil1. 12.Perceptron Modeli.....	24
Şekil1. 13.İleri Beslemeli Sinir Ağı.....	25
Şekil1. 14.Geri Beslemeli Sinir Ağı.....	26
Şekil1. 15. Çok Katmanlı Algılayıcı.....	28
Şekil1. 16.Evrişimsel Sinir Ağı.....	29
Şekil1. 17.Yinelemeli Sinir Ağı Modeli.....	29
Şekil1. 18.Uzun-Kısa Terimli Hafıza Modeli.....	30
Şekil1. 19.Oto Kodlayıcı.....	31
Şekil1. 20. Tanımlayıcı İstatistikler.....	40
Şekil1. 21. Korelasyon Analizi.....	41

Tablolar Dizini

Çizelge1. 1. İstatistik Terminolojisi ve Yapay Sinir Ağları Terminolojisi Karşılaştırılması.....	7
Çizelge1. 2. Kayıp Gözlemlerin İncelenmesi.....	40
Çizelge1. 3. OLS Doğrusal Regresyon Model Anlamlılık Sonuçları.....	44
Çizelge1. 4.OLS Doğrusal Olmayan Regresyon Modeli Anlamlılık Sonuçları.....	44
Çizelge1. 5.OLS Doğrusal Regresyon Analizi Katsayı Anlamlılıkları.....	45
Çizelge1. 6. Örnek Dışı Derin Sinir Ağı-Doğrusal Regresyon Analizi Hiper-Parametre Seçimleri.....	46
Çizelge1. 7.Doğrusal Olmayan Regresyon Analizi Katsayıların Anlamlılık Sonuçları.....	47
Çizelge1. 8.Örnek-Dışı Derin Sinir Ağı-Doğrusal Olmayan Regresyon Analizi Hiper-Parametre Seçimleri.....	49

1. GİRİŞ

Yapay zekâ çalışma alanı makinelerle insan yeteneklerini, düşünce ve davranış biçimleri gibi fonksiyonları kazandırma işlemlerinden oluşmaktadır. Örüntüleri tanıma, tecrübelerden öğrenme, sonuç çıkarma ve tahminlerde bulunma gibi süreçleri içermektedir. Turing, yapay zekâyı, bir insan ve makineye aynı sorular sorulduğunda, verilen cevabın ayırt edilme durumuna göre “taklit oyunu” adını verdiği bir oyun ile tanımlamıştır. Eğer cevapların makineden veya insandan geldiği ayırt edilemiyorsa makinenin akıllı olduğu kabul edilecektir (Gürsakal, 2021). Yapay zekâ, günümüze kadar birçok duraklama dönemlerine girmiştir. Fakat bu fikirden vazgeçilmemiştir. Yapay zekâ, ilk dönemlerinde zayıf bilgisayar gücü nedeniyle hayata geçirilmede zorlanılmıştır. Ve bu sebeple yükselme dönemi bugünlere kadar uzamıştır. Günümüzde artan teknoloji ve bilgisayar gücü ile yapay zekâ, büyük veri kavramı önemli derecede ilgi odağı olmuştur. Disiplinlerarası bir kavram olması yapay zekâyı olan önemin ve ilginin artmasını sağlamıştır. Bugüne kadar, yapay zekânın ilerlemesi hakkında iyimser ve kötümser farklı yaklaşımlarda bulunulmuştur. Örnek olarak, Bill Gates ve Mark Zuckerberg yapay zekâ için iyimser bir görüşte olmasına karşın; ünlü fizikçi Stephen Hawking, yapay zekâ için “uygarlık tarihinin en kötü olayı” olarak bahsetmiştir, Tesla ve SpaceX CEO’su Elon Musk, yapay zekânın üçüncü dünya savaşına neden olabileceğini dile getirmiştir. Kötümser görüşler olmasına rağmen, yapay zekâdan vazgeçilememiştir.

Yapay zekâ, büyük veriye dayanmaktadır. Artan işlem hızı ve veriler, bilgisayarların hız ve verim açısından optimal kullanılması gerekliliğini ortaya koymuştur. Bilgisayarlar insan zekâsına oldukça benzemektedir. Bu sebeple insan zekâsının çalışma prensibi ne kadar iyi anlaşılırsa, yapay zekâ o kadar ilerleyecektir denilebilir (Gürsakal, 2021).

Yapay zekânın bir alt dalı olan makine öğrenmesi, matematiksel ve istatistiksel yöntemlerle veriden öğrenen, çıkarımlar yapan ve bu çıkarımlarla tahminleme yapan yöntemler paradigmasıdır. Makine öğrenmesi, denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning), yarı-denetimli öğrenme (semi-supervised learning) ve takviyeli öğrenme (reinforcement learning) olarak dört temel algoritmadan oluşmaktadır. Denetimli öğrenme, önceden etiketlenen veriler tarafından oluşturulan öğrenme algoritmasıdır. Denetimsiz öğrenme, etiket olmayan verilerden kendisi öğrenen öğrenme algoritmasıdır (Gürsakal,2021). Yarı-denetimli öğrenme, veri kümesi etiketlenmiş ve etiketlenmemiş örnekleri aynı anda tutar. Böylece hem etiketli hem de etiketsiz verilerle çalışmaktadır.

Takviyeli öğrenme ise insan davranışlarını taklit ederek öğrenen öğrenme algoritmasıdır. Diğer bir ifadeyle, insanların öğrenme şeklini taklit eden algoritmalarıdır. Hinton ve Sejnowski'nin geliştirdiği Boltzman Kuralı veya Genetik Algoritmalar takviyeli öğrenmeye örnek olarak verilebilir (Güzel Y.,2018). Makine öğrenmesinin en temel yapılarından birisi de yapay sinir ağları olarak gösterilmektedir.

Yapay sinir ağları, insan beyninin öğrenme yolundan faydalanarak elde ettiği bilgi edinme, yorumlama ve geliştirme özelliklerini taklit eden biyolojik sinir ağlarını matematiksel forma getiren algoritmalarıdır.

Yapay sinir ağlarının çok katmanlı kullanımı derin sinir ağları olarak adlandırılmaktadır. Derin öğrenme büyük miktarda veri ile eğitilir ve karar verir. Geleneksel yapay sinir ağı öğrenmesinde, veri eklendikçe performans artış göstermektedir. Fakat belirli bir süre sonra bu performans artışı sabit kalır. Derin öğrenmede veri arttıkça modelin performansı artacaktır. Tabii ki zaman ve maliyet açısından optimum seviyede veri artışı gerekmektedir. Veri arttıkça modelin eğitilebilmesi daha uzun zaman alacaktır. Diğer yandan, derin sinir ağlarının kompleks problem çözme yeteneğini artıracaktır. Günümüzde derin öğrenme görüntü işleme, ses ve video işleme, metin analizi, duygu analizi ve benzeri problemlerde gösterdiği başarılı performans sebebiyle tercih edilmektedir. Youtube, Amazon gibi ileri teknoloji ile çalışan öneri sistemleri de, derin öğrenmeye dayanan sistemlerle çalışmaktadır (Gürsaka, 2021).

Derin sinir ağları birçok parametre ve hiper-parametreler içermektedir. Modele dahil olan ve değeri veriden tahmin edilebilen değişkenlere parametre denir. Hiper-parametreler ise, model parametrelerini kontrol eden model dışı parametrelerdir. Hiper-parametreler kontrol parametreleridir. Derin sinir ağlarının parametreleri, ağırlık modelini oluşturan ağırlıklar ve yan değerleridir. Derin sinir ağlarının bazı hiper-parametreleri ise öğrenme oranı, eğitimdeki iterasyon sayısı, tabaka sayısı, tabakalardaki nöron sayısı, aktivasyon fonksiyonu türü, bırakma oranı, eğitim tur sayısı, parti boyutu, ağırlık başlatma, momentum, mini-parti boyutu, düzeltme parametreleri şeklinde verilebilir. Hiper-parametre değerleri seçilirken, çoğu zaman kullanımı tercih edilen hiper-parametre değerleri alınır veya tecrübeye dayalı seçimler yapılabilir. Hiper-parametreler için yapılabilecek seçim sayısı sonsuzdur. Bu sebeple hiper-parametre seçimi için literatürde birçok çalışma yapılmıştır. Literatürde kullanılan bazı hiper-parametre seçim yöntemleri Izgara Arama (Grid Search) (Çarkacı N., 2018), Rastgele Arama (Random Search) (Çarkacı N., 2018) ve Bayeşçi Yaklaşım (Bayesian Approach) (Çarkacı N., 2018) olarak verilebilir.

Yapay sinir ağlarında en iyi performansı veren ağ mimarisinin belirlenmesi problemine mimari seçimi problemi adı verilir. Bu önemli problemin çözümünde, en iyi performansın elde edilmesi için kaç tabaka kullanılacağı, hangi tabakada kaç nöron kullanılacağı gibi soruların cevapları araştırılır. Aladağ 2009 yılındaki çalışmasında mimari seçimi problemi çözümü için makine ve istatistiksel öğrenmesini birlikte kullanan melez bir yaklaşım önermiştir. Aladağ (2019) tarafından önerilen yaklaşımda, yapay sinir ağı girdi ve gizli tabakalarındaki nöron sayıları ile ağın zaman serilerindeki öngörü performansı arasındaki ilişki yapısı basit doğrusal regresyon modeli ile ortaya konulmuştur. Bulunan regresyon modeline göre, en iyi yapay sinir ağı mimarisi istatistiksel olarak belirlenebilmektedir. Bu şekilde, tabakalardaki nöron sayılarının ağın öngörü performansı üzerindeki etkisinin anlamlılık düzeyi istatistiksel olarak yorumlanabilmektedir. Böylece, çözümlenen veriye göre sinir ağı mimarisi istatistiksel olarak sistematik bir biçimde belirlenebilmektedir. Aynı zamanda, kurulan regresyon modeli ile hiç çözümlenme yapılmadan, nöron sayılarına göre ağın öngörü performansı tahmin edilebilmektedir. Özellikle büyük veri olması durumunda bu çok önemli bir avantaj sağlamaktadır (Aladağ, 2019).

Gerçekleştirilen tez çalışmasında, Aladağ (2019) çalışmasında önerdiği istatistiksel ve makine öğrenmesine dayalı melez yaklaşım, derin sinir ağlarının hiper-parametre seçimi probleminin çözümü için genişletilerek uygulanmıştır. İki gizli tabakalı ileri beslemeli derin sinir ağı için çözümlenme düşünülmüştür. Derin sinir ağının performans ölçütü olarak *ikili çapraz belirsizlik kaybı (log-loss)* değeri kullanılmıştır. Makine öğrenmesi, matematiksel ve istatistiksel yöntemlerle veriden öğrenen, çıkarımlar yapan ve bu çıkarımlarla tahmin üretebilen yöntemler toplamıdır. İstatistiksel öğrenme ise özellikler arasındaki ilişkiyi ortaya koyabilen, tahminler üretebilen ve varsayımlara dayanan kural tabanlı bir yaklaşımdır. Makine öğrenme çeşidi olan yapay sinir ağlarının hiper-parametreleri, istatistiksel öğrenme yöntemi olan regresyon modeli ile belirlenmektedir. Buna göre, elde edilen sonuçlara istatistiksel hipotez testleri uygulamak ve böylece istatistiksel olarak değerlendirmeler yapılarak en iyi hiper-parametre seçimi gerçekleştirilmektedir. En iyi hiper-parametre seçimi, istatistiksel bir modelle tanımlanabildiğinden, incelenmemiş mimarilerin sonuçlarını da tahmin etmek mümkün olmaktadır. Böylece, özellikle büyük veri kavramı söz konusu olduğunda, işlem zamanından çok önemli ölçüde kazanım sağlamak mümkün olmaktadır. Aynı zamanda, hiper-parametrelerin performansa olan etkileri istatistiksel olarak açıklanabilir ve yorumlanabilir olmaktadır.

Tezin ikinci bölümünde yapay zekâ, makine öğrenmesi, istatistiksel öğrenme hakkında genel bilgiler verilmektedir. Üçüncü bölümde yapay sinir ağları ve derin sinir ağları hakkında genel bilgiler ve parametre ve hiper-parametre kavramları verilmektedir. Dördüncü bölümde doğrusal regresyon ve doğrusal olmayan regresyon analizleri hakkında genel bilgiler verilmektedir. Beşinci bölümde Aladağ (2019) tarafından önerilen melez yaklaşımdan bahsedilmiştir. Altıncı bölümde veri tanıtımı yapılmış ve veri analizi adımları verilmiştir. Son olarak ise yedinci bölüm sonuç bölümünde ise uygulamadan elde edilen bilgiler yorumlanmıştır.

2. YAPAY ZEKÂ, İSTATİSTİKSEL ÖĞRENME VE MAKİNE ÖĞRENME

Yapay zekâ, makinelere insan yetenekleri, düşünce ve davranış biçimleri gibi fonksiyonlarını kazandırma işlemleridir. Örüntüleri tanıma, tecrübelerden öğrenme, sonuç çıkarma ve tahminlerde bulunma gibi özelliklere sahiptir. Yapay zeka fikir babası İngiliz matematikçi ve bilgisayar bilimci Alan Mathison Turing olarak görülmektedir. Alan Turing ikinci dünya savaşı sırasında Arthur Sherbius Enigmasını kıran kişi olarak tanınmaktadır. 1950 yılında yayınlanan “Makinelerin İşleyişi ve Zekâ” adlı makalesi ile bilgisayarın düşünüp düşünemeyeceği testini ortaya koyan ve “Taklit Oyunu” olarak isimlendirdiği çalışmasını ortaya atmıştır. Bu test, makine ile konuşulduğunda makine olup olmadığını anlama durumuna göre değerlendirilmektedir. Sonuçları halen tartışma söz konusu olsa da testin geçmesi 60 yılı almıştır (Gürsakal, 2021).

Yapay zekâ, teknikleri bazıları aşağıda verilmektedir (Yılmaz, 2020):

- Uzman sistemler (Expert systems)
- Bulanık Mantık (Fuzzy Logic)
- Yapay Sinir Ağları (Artificial Neural Network)
- Genetik Algoritmalar (Genetic Algorithms)
- Tavlama Benzetimi (Simulated Annealing)
- Bilgisayarlı Görme (Computer Vision)
- Robotik

Alpaydın iç içe kavramlar olan istatistiksel ve makine öğrenme hakkında aşağıdaki gibi bir tanımlama yapmaktadır (Keskin, 2018):

“Makine öğrenmede öğrenme işi istatistik ile yapılır, çünkü amaç örnek veri üzerinden genellenebilir yapılar ortaya çıkarmaktır. Bilgisayar bilimleri ise istatistiksel öğrenme alanında yer alan öğrenme yaklaşımlarının optimizasyonu ve performansı ile ilgilidir.”

İstatistiksel öğrenme, özellikler arasındaki ilişkiyi ortaya koyabilen, çıkarsama yapabilen ve varsayımlara dayanan kural tabanlı bir yaklaşımdır. İstatistiksel öğrenme “model” temelli bir yaklaşımdır. Makine öğrenmesinin temelini oluşturan yaklaşımlardan biri olan Regresyon Analizi istatistiksel bir öğrenme yöntemidir.

Makine öğrenmesi, matematiksel ve istatistiksel yöntemlerle veriden öğrenen, çıkarsama yapabilen yöntemler bütünü olarak tanımlanabilir. Makine öğrenmesinin en temel yapılarından birisi de YSA olarak gösterilmektedir.

Makine öğrenmesi, denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning), yarı-denetimli öğrenme (semi-supervised learning) ve takviyeli öğrenme (reinforced learning) olarak dört temel algoritmadan oluşmaktadır.

Denetimli Öğrenme (Supervised Learning)

Denetimli öğrenme girdi değişkenlerini (X), çıktı değişkenlerine (Y) eşleme işlevini öğrenmek için önceden etiketlenen eğitim verileriyle ağırlıklarının ve yan değerlerinin güncellenmesiyle gerçekleşmektedir. Sistem bu örneklerden yola çıkarak bir genelleme yapar ve bir çözüm uzayı verir. En çok kullanılan denetimli öğrenme algoritmaları (Géron A., 2019):

- K-En Yakın Komşu (k-nearest Neighbors)
- Doğrusal Regresyon (Linear Regression)
- Lojistik Regresyon (Logistic Regression)
- Destek Vektör Makineleri (Support Vector Systems-SVM)
- Karar Ağaçları ve Rastgele Ormanlar (Decision Trees and Random Forests)
- Sinir Ağları (Nerual Networks)

Denetimsiz Öğrenme (Unsupervised Learning)

Denetimsiz öğrenme denetimsiz öğrenme problemleri önceden etiketlenen veriler olmadan kendisi öğrenerek çalışmaktadır. Girdi değişkenlerine (X) sahipken, çıktı değişkenleri (Y) yoktur. Başka bir ifadeyle herhangi bir öğretmen veya denetmen yoktur (Yılmaz, 2020). Çok sayıda araştırmacı, denetimsiz öğrenmeyi yapay zekânın insan beyni düzeyinde geliştirmenin anahtarı olarak görmektedir (Gürsakal, 2021). En çok kullanılan denetimsiz öğrenme algoritmaları aşağıda verilmiştir (Géron A., 2019).

- Kümeleme (Clustering)
 - K-Ortalamalar (K-Means)Yoğunluğa Dayalı Uzamsal Kümeleme (Density-Based Spatial Clustering-DBSCAN)
 - Hiyerarşik Kümeleme Analizi (Hierarchical Cluster Analysis)
- Anomali ve Yenilik Tespiti (Anomaly detection and novelty detection)
 - Bir Sınıflı SVM (One Class SVM)
 - İzolasyon Ormanları (Isolation Forest)
- Görselleme ve Boyut İndirgeme (Visualization and dimensionality reduction)
 - Temel Bileşen Analizleri (Principal Component Analysis-PCA)
 - Çekirdek Temel Bileşen Analizleri (Kernel PCA)
 - Yerel Doğrusal Gömme (Locally Linear Embedding-LLE)
 - t-dağılımlı Stokastik Komşu Gömme (t-distributed Stochastic Neighbor Embedding)
- İlişkilendirme Kuralı Öğrenimi (Association rule learning)
 - Apriori
 - Eclat

Yarı-Denetimli Öğrenme

Veri kümesi etiketlenmiş ve etiketlenmemiş örnekleri aynı anda tutar. Genellikle etiketlenmemiş örneklerin sayısı, etiketlenmiş örneklerden fazladır. Etiketlenmemiş örnekler eklendiğinde, etiketlenmiş verilerin olasılık dağılımını daha iyi yansıtmaktadır (Okatan A. Vd., 2021)

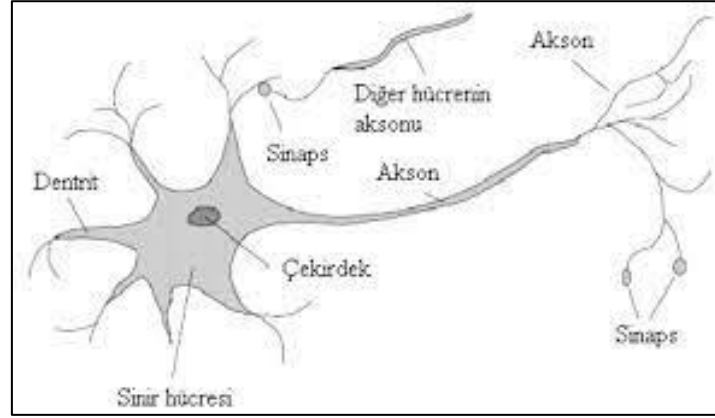
Takviyeli Öğrenme (Reinforcement Learning)

Takviyeli öğrenme ise insan davranışlarını taklit ederek öğrenen öğrenme algoritmasıdır. Denetimsiz öğrenmeye benzer olarak sadece girdi değişkenleri sisteme verilirken herhangi bir çıktı değeri bulunmamaktadır. Fakat sistemin bir denetmeni bulunur ve bu denetmen sisteme çıktı değerinin doğru veya yanlış olduğunu göstermektedir. Dolayısıyla çıktı değişkenleri verilmediği halde çıktının doğruluğu için denetmen ile takviye edilmiş olur. (Yılmaz, 2020)

3. YAPAY SİNİR AĞLARI (YSA)

YSA, insan beyninin öğrenme yolundan faydalanarak elde ettiği bilgi edinme, yorumlama ve geliştirme özelliklerini taklit eden biyolojik sinir ağlarını matematiksel forma getiren algoritmalarıdır. Diğer bir deyişle, YSA biyolojik sinir ağlarını taklit eden sentetik ağlardır

(Egrioglu vd, 2009). Her ne kadar YSA gerçek sinir hücrelerinden esinlenmiş olsa da hala biyolojik sinir ağlarının nasıl bir işleyişi olduğu hakkında tam bir bilgiye sahip olunamamıştır. Fakat genel bir işleyiş hakkında sahip olunan bilgiyle YSA kurulmuştur. Hiçbir model insan beyni performansı ile aynı performans göstermede başarılı olamamıştır (Gürsaka, 2021).



Şekil1. 1. Biyolojik Sinir Hücresinin Yapısı

Şekil1.1’de görüldüğü üzere biyolojik sinir hücrelerinin uçlarında dendrit yapısı bulunmaktadır. Dendritler, duyu organları ve diğer nöronlardan gelen sinyalleri alır ve çekirdeğe iletir. Çekirdeğe iletilen bu sinyaller, çekirdekte toplanır ve aksona iletilir. Bu sinyaller aksonda işlenir ve diğer nöronların bağlantılarına iletilir. Bu görevi ise sinapsler üstlenmektedir.

Yapay sinir ağları, biyolojik sinir ağlarındaki sinapsler ve dendritler arasındaki bilgi akışının ağırlık katsayılarıyla güncellenmesi düşüncesi ile ortaya çıkmıştır. Sinapsler ile öğrenmenin nasıl olması gerektiği fikri ortaya çıkmıştır.

YSA modelleri istatistiksel modellere benzer olmasına rağmen, YSA ile istatistik literatürleri arasındaki farklılıklar vardır (Aladağ, 2009). Literatürde YSA ile bazı istatistiksel modeller arasındaki ilişkiyi inceleyen çalışmalar yapılmıştır (Sarle, 1994). Çizelge1.1.’de YSA ve istatistik terminolojilerinin karşılaştırılması verilmektedir (Nelles, 2001, Günay vd, 2007):

Çizelge1. 1. İstatistik Terminolojisi ve Yapay Sinir Ağları Terminolojisi Karşılaştırılması

İstatistik Terminolojisi	YSA Terminolojisi
Model	YSA
Parametre	Ağırlık
Bağımsız değişken	Girdi
Tahmin değeri	Çıktı
Bağımlı değişken	Hedef
Artık	Hata

Güven aralığı	Hata çizgisi
Temel Fonksiyon	Nöron
Bağımsız değişkenler kümesi	Girdi tabakası
Temel fonksiyonlar kümesi	Gizli tabaka
Tahmin değerleri kümesi	Çıktı tabakası
Gözlem	Örüntü
Kestirim ya da optimizasyon	Eğitim, Öğrenme veya Adaptasyon
Örnekleme adaptasyonu	Çevrimiçi öğrenme
Grup adaptasyonu	Çevrimdışı öğrenme
Kestirim ölçütü	Hata, maliyet veya Lyapunov fonksiyonu
Dönüşüm	Fonksiyonel bağlantı
Diskriminant analizi	Sınıflama
Regresyon	Eşleme, Yaklaşım veya Denetimli öğrenme
Veri İndirgeme	Denetimsiz öğrenme veya şifreleme
Genelleştirme	İnterpolasyon veya Ekstrapolasyon

YSA'ların güçlü olduğu durumlar genel olarak aşağıdaki gibi özetlenebilmektedir (Yılmaz, 2020):

- YSA doğrusal olan ya da olmayan ya da her iki yapıyı da barındıran modellerde kullanılabilir. Bu özellik ile kompleks model görevlerinde daha esnek ve güçlü bir yaklaşım sağlamaktadır.
- Yapay sinir ağları parametrik olmayan bir yöntemdir. Herhangi bir varsayıma dayanmaz. Bu nedenle gerçek hayat verilerine uygulanması daha kolaydır.
- Yapay sinir ağları, yeterince büyük bir ağ verildiğinde herhangi bir karmaşık işlevi keyfi belli bir doğruluk ile yaklaştırabilen işlevsel yaklaşımlardır.

YSA'ların gerçek hayat uygulamaları üzerinde sağladıkları avantajların yanında yöntemin performansını etkileyen bazı etkenler de söz konusudur. Bu etkenler aşağıdaki gibi özetlenebilir (Atınc Y., 2020):

- YSA oluşturulurken öğrenme oranı, mimari seçimi, katman sayısı, katmanlardaki nöron sayısı gibi ağın yapısını belirleyen bazı bileşenler mevcuttur. Bu bileşenlerin belirlenmesi için genel bir kural yoktur. Bileşenlerin en iyi şekilde belirlenmesi gerekir.

- YSA'da, ağın davranışları açıklanamamaktadır. Bir probleme çözüm üretildiğinde nasıl çözüm ürettiğine dair bilgi vermez.
- YSA'da ağın eğitim süresi çözümlenecek veriye ve ağın seçilen bileşenlerine bağlı olarak büyük değişimler gösterebilir.
- YSA, donanım bağımlı bir algoritmadır. Paralel işlem gücü olan işlemcilere bağlı olarak performansı etkilenebilir.
- YSA sayısal değerlerle çalıştığında örneklerin sayısal değerlere nasıl çevrildiği performansı doğrudan etkileyebilmektedir.

YSA yönteminin geçmişi önemli çalışmalarla aşağıda verildiği gibi özetlenebilir (Yılmaz A, 2020).

- Warren McCulloch- Walter Pitts tarafından 1943 yılında YSA'nın temeli olan beyin hesaplama yeteneğini taklit ederek basit bir YSA oluşturulmuştur.
- Hebb 1949 yılında öğrenebilen ve adapte olabilen YSA için öğrenme algoritmasının temeli olan Öğrenme Kuralı ortaya koymuştur. Bu kural ile YSA bağlantılarının değiştirilmesi ile öğrenme işlevi ortaya çıkacağını belirtmiştir.
- Frank Rosenblatt 1957 yılında bilinen en basit model olan "Perceptron" modeli geliştirmiştir. Perceptron modeli, tek katmanlı öğrenme yeteneğine sahip, tek bir giriş ve çıkışa sahip bir YSA modelidir. Perceptron modeli basit doğrusal örüntü sınıflandırma problemlerinde kullanılmıştır.
- Bernard Widrow – Marcian Hoff ADALINE 1959 yılında öğrenme algoritmasını geliştirmişlerdir. Eğim düşümlü tabanlı öğrenim kuralı uygulanmıştır.
- 1965 yılında ilk makine öğrenmesi kitabı yayınlanmıştır.
- 1967-1969 yılları arasında Grosberg gibi bazı öğrenme algoritmaları ortaya konmuştur.
- Hopfield 1980 yılında kompleks bir yapı için matematiksel bir temel kurmuştur. Hopfield ağı bu temel üzerine kurulmuştur.
- Hopfield 1984 yılında Kohonen danışmasız öğrenme mantığını ortaya koymuştur.
- 1984 yılında Boltzman sabiti geliştirilmiştir.
- Rumelhart – McClelland 1986 yılında çok katmanlı ağ yapılarında geri yayımlı öğrenme algoritmasını ortaya koymuştur.
- 1987 yılın Elektrik Elektronik Mühendisliği Enstitüsü tarafından YSA'yı konu alan ilk konferans gerçekleştirilmiştir.

YSA, günümüzde birçok alanda gerçek hayat problemlerinin çözümünde kullanılmaktadır. Bu alanlardan bazıları aşağıda verilmiştir;

- Sistem modelleme
- Ses tanıma
- Parmak izi Tanıma
- Otomatik araç denetimi
- Fizyolojik işaretleri tanıma
- Sahtekarlık (Fraud) analizi
- Finans, endüstri, eğitim
- Basit yöntemler çözölemeyen problemler
- Doğrusal olmayan sistemler de kullanım alanları örnek olarak verilebilir.

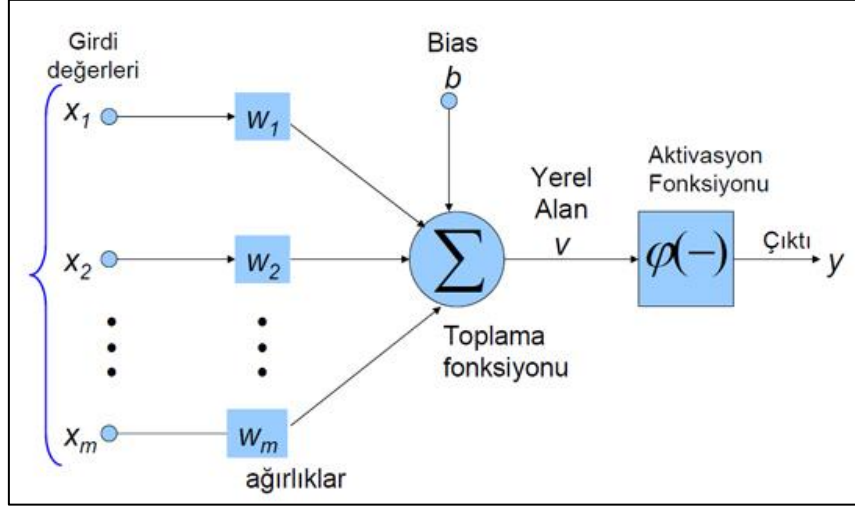
Son yıllarda YSA yaklaşımının kullanım amaçları aşağıdaki gibi verilebilir (Elmas, 2018).

- Öngörü
- Sınıflandırma
- Veri birleştirme
- Veri kavramlaştırma
- Veri süzülmesi
- Resim veya görüntü işleme

3.1. YSA'nın Yapısı ve Temel Özellikleri

YSA yöntemi bazı temel bileşenlerden oluşmaktadır. Belirtilen bileşenlerin doğru seçilerek YSA modelinin belirlenmesi yöntemin performansı üzerinde doğrudan etkilidir (Aladağ vd. 2013). YSA 3 temel bileşenden oluşmaktadır.

- Mimari yapı
- Aktivasyon Fonksiyonu
- Öğrenme Algoritması



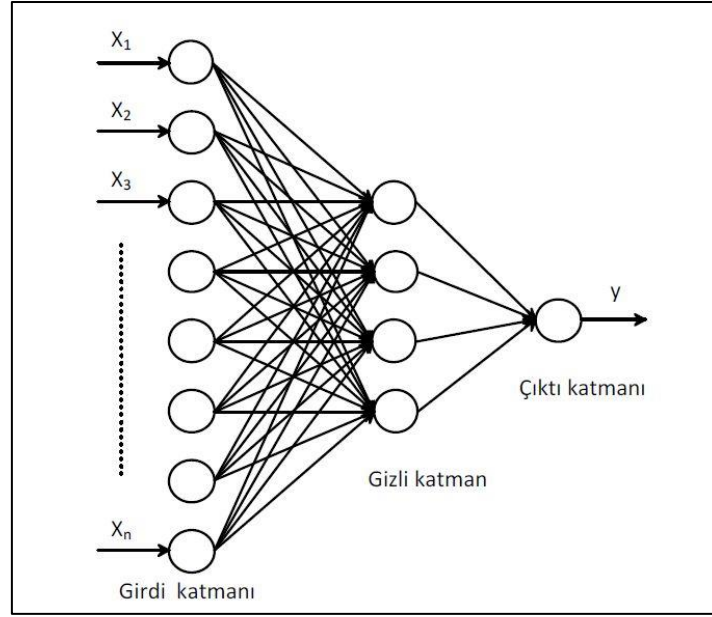
Şekil1. 2. YSA'nın Genel İşleyişi

YSA'da genel süreç Şekil-2 de gösterildiği üzere girdi değerleri ile alınan gözlemler ağırlıkları ile çarpılır (w_i) ve yan (bias) değerleri eklenerek bir toplama fonksiyonu içine girer. Toplama fonksiyonu, aktivasyon fonksiyonu ile işlenerek bir çıktı değeri elde edilir. Modellenen bu YSA'nın öğrenme algoritması, bu ağırlıkların ve yanların güncellenmesi ile oluşmaktadır. Model, öğrenme algoritması ile farklılaşmakta ve iyileşmektedir.

YSA'nın sahip olduğu temel bileşenler aşağıda açıklanmaktadır (Egrioglu 2009).

Mimari Yapı (Architecture Structure)

YSA'da yöntemin bileşenlerinin doğru seçilmesi modelin performansını doğrudan etkilemektedir. Doğru bileşenler seçilmediği takdirde, YSA yöntemi yanıltıcı sonuçlar verebilmektedir (Aladağ vd., 2007). Literatürde en iyi mimari yapıyı tanımlamak için genel bir kural yoktur. Eldeki veriye göre en iyi olarak belirlenen bir bileşen başka bir veri söz konusu olduğunda çok kötü sonuçlar verebilir (Bodyanskiy ve Popov, 2006). YSA'nın üç temel bileşenleri arasında en çok seçenek sunan mimari yapıdır (Aladağ, 2009). YSA girdi tabakası, gizli tabakaları ve çıktı tabakasından oluşmaktadır. Mimari yapı her tabakada bulunan nöron sayısı olarak tanımlanmaktadır. Bu nöronların her biri diğer nöronlara ağırlıklar ve yanlarla bağlanmaktadır. Modeli oluşturan tabaka sayıları, her tabakada yer alan nöron sayıları yüzlerce seçenek ortaya koyacaktır ve bu seçenekler arasında en doğru mimari yapıyı bulmak büyük bir önem arz etmektedir.



Şekil1. 3. Tek Çıktı Nörona Sahip İleri Beslemeli YSA

Şekil-3'te tek çıktı nörona sahip çok katmanlı ileri beslemeli sinir ağı gösterilmektedir. Görüldüğü üzere ileri beslemeli YSA mimarisi, girdi, gizli ve çıktı tabakaları olmak üzere üç tabakadan oluşmaktadır. Her bir tabakada nöronlar bulunmaktadır. Mimari yapının belirlenmesi her tabakadaki nöron sayılarının belirlenmesi sürecidir. Belirtilen nöronlar birbirine ağırlıklarla bağlanmaktadır. İleri beslemeli bir YSA olduğundan dolayı aynı tabakadaki nöronlar arasında bağlantı yoktur (Aladağ, 2011).

YSA yönteminde en iyi mimarinin belirlenmesi için literatürde önerilen yaklaşımlar, gözlemlere/tecrübelere dayanan ve sistematik olan yaklaşımlar olarak iki başlıkta toplanabilir (Aladağ, 2009). Literatürde kullanılan bazı sistematik yaklaşımlar aşağıda verilmiştir (Aladağ, 2009).

- Yapıcı ve budama algoritması (Constructive and pruning)(Siestema & Dow, 1988)
- Polinom zamanı algoritması (Polynomial Times Algorithm) (Roy, Kim & Mukhopadhyay, 1993)
- Ağ bilgi kriteri (Network Information Criterion)(Murata, Yoshizawa, & Amari, 1994)
- Yinelemeli inşa algoritması (Iterative construction algorithm)(Rathbun, Rogers, DeSimo, & Oxley, 1997)
- Box-Jenkins analizine dayalı yöntem (Method based on Box–Jenkins analysis)(Buhamra, Smaoui, & Gabr, 2003)
- Entropi bilgisine dayalı yöntem (Method based on information entropy)(Yuan, Xiong, & Huai, 2003)
- Genetik algoritma (Genetic algorithms)(Dam & Saraf, 2006)

- Temel bileşenler analizleri (Principle component analysis)(Zeng, Guo, & Hu, 2007)
- Ağırlıklı bilgi kriteri (Weighted information criterion)(Egrioglu et al., 2008)
- Silme /ikame/ekleme algoritması (Deletion/substitution/addition algorithm) (Durbin, Dudoit, & Van der Laan, 2008)
- Otoregresif mevsimsel zaman serileri için bir mimari seçim stratejisi (An architecture selection strategy for autoregressive seasonal time series) (Aladag, Egrioglu, & Gunay, 2008)
- Deney tasarımı (Design of Experiments) (Balestrassi, Popova, Paiva, & Marangon Lima, 2009)

Belirtilen mimari seçim yöntemleri farklı problemlerde farklı performans göstermektedirler. YSA yönteminin performansını doğrudan etkileyen mimari seçimi problemi için literatürde genel bir yaklaşım yoktur.

Öğrenme Algoritması (Learning Algorithm)

YSA yönteminde model parametreleri, ağırlıklar ve yan değerlerdir. Öğrenme işlemi ağırlıkların ve yan değerlerinin güncellenmesi ve modeli temsil edebilecek en iyi değerlerin bulunması olarak tanımlanır. En iyi ağırlık ve yan değerleri bulunduğu anda ağ eğitilmiştir. YSA modelinin eğitilmesinde kullanılan optimizasyon algoritmasına öğrenme algoritması adı verilir.

Literatürde YSA'nın eğitimi için çeşitli öğrenme algoritmaları bulunmaktadır. En çok kullanılan algoritmalarından bir tanesi de Geri Yayılım Öğrenme Algoritması'dır (Back Propagation Learning Algorithm). Bu öğrenme algoritması, gözlenen değer ve çıktı değer arasındaki farka dayanarak ağırlıkları güncellemektedir. Belirtilen algoritma Dereceli Alçalma (Gradient Descent) tabanlı bir optimizasyon algoritmasıdır. Bu algoritmanın dezavantajları aşağıda verilmiştir.

1. "Gradient" tabanlı olması sebebiyle yerel minimuma (local-minima) takılı kalabilir, optimal sonuca ulaşamayabilir.
2. Eğitim performansı, başlangıç ağırlık değerlerine ve algoritma parametrelerinin seçimine çok duyarlıdır.

Bu zayıflıkları düzeltmek için literatürde çeşitli çalışmalar yapılmıştır (Ye, 2007). Özellikle yerel (local) tuzaklardan kaçınmak için sezgisel algoritmalar (heuristic algorithms) yararlanan çalışmalar da bulunmaktadır (Aladağ, 2009). Sezgisel algoritmalara örnek olarak;

genetik algoritmalar (genetic algorithms), benzetimli tavlama algoritması (stimulated annealing), ve tabu arama algoritması (tabu search algorithm) verilebilir. Verilen yöntemler yerel optimuma takılmaktan kaçınmakta başarılı algoritmalar (Li, Su, & Chiang, 2003; Sexton, 1998 ; Sexton, Dorsey & Johnson, 1999).

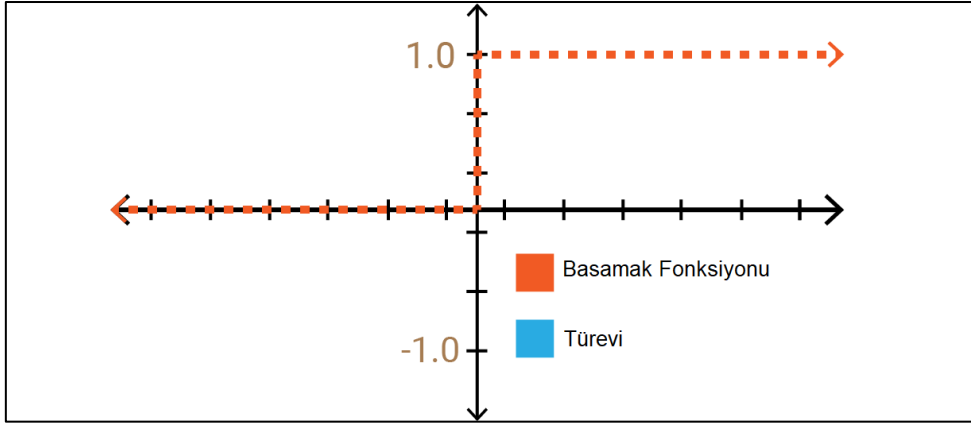
Aktivasyon Fonksiyonu (Activation Function)

Aktivasyon fonksiyonu girdi ve çıktı arasında doğrusal veya doğrusal olmayan ilişkiyi veren fonksiyondur. Ağın performansının aktivasyon fonksiyonu seçiminden etkilendiği bilinmektedir (Aladağ vd., 2014). Aktivasyon fonksiyonu girdi olarak alınan veri kümesiyle çıktı arasındaki ilişki yapısını modellemede kullanılmaktadır. YSA modelinde nöronlarda aynı veya farklı aktivasyon fonksiyonları kullanılabilir. Tek kutuplu, çift kutuplu veya doğrusal aktivasyon fonksiyonları kullanılabilir. Tek kutuplu aktivasyon fonksiyonlarının çıktı değerleri 0 ile 1 aralığında değer alırlarken, çift kutuplu aktivasyon fonksiyonlarının çıktı değerleri -1 ile 1 arasında değer almaktadır. Aktivasyon fonksiyonlarının en çok kullanılan bazı türleri aşağıda verilmektedir.

Adımsal Fonksiyon (Step Function)

- İkili değer alan fonksiyonlar olduğundan ikili sınıflandırma problemlerinde kullanımı tercih edilir.
- Genellikle çıktı katmanlarında kullanılmaktadır. Şekil1.4'te görüldüğü üzere gizli katmanlarda kullanımı türevi modelin öğrenmesini temsil edememesi sebebiyle kullanımı tercih edilmemektedir.
- Çok güçlü bir fonksiyon küçük değişikliklerden etkilenmez.
- {0,1} değerlerini almaktadır.

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

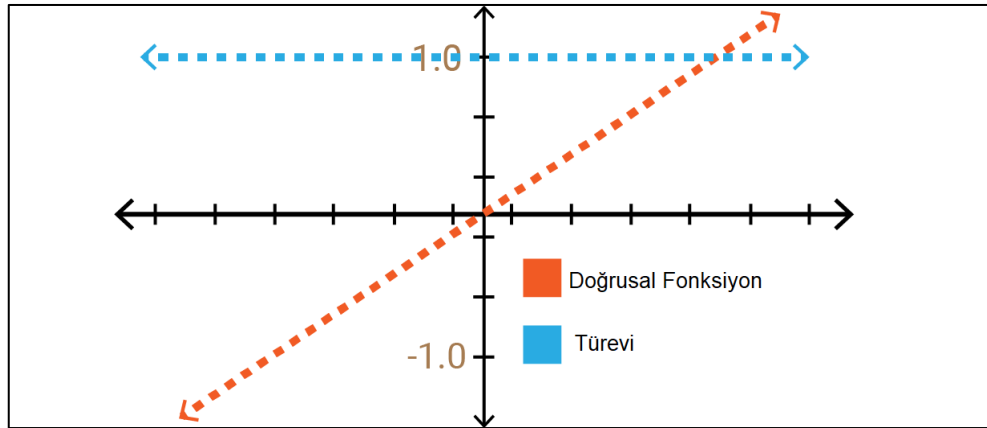


Şekil1. 4. Adımsal Fonksiyon Grafiği

Doğrusal Fonksiyon (Linear Function)

- Şekil-1.5'te görüldüğü üzere türev sabit olmasından dolayı öğrenme işlemi gerçekleştirilememektedir.
- Tüm katmanlarda doğrusal fonksiyon kullanılması girdi ile çıktı katmanları arasında aynı doğrusal ilişkiyi verecektir. Ki bu da gizli katmanların bir anlam ifade etmemesi anlamına gelmektedir. Yani kompleks yapılarda kullanımı tavsiye edilmemektedir.
- $(-\infty, \infty)$ aralığında değer almaktadır.

$$f(x)=x \quad (-\infty, \infty)$$



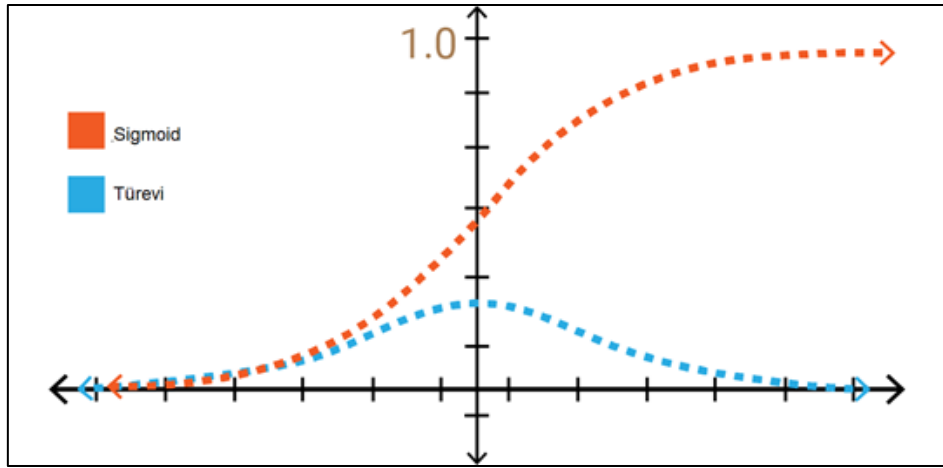
Şekil1. 5. Doğrusal Fonksiyon Grafiği

Sigmoid Fonksiyon (Sigmoid Function)

- Şekil-1.6'de görüldüğü üzere x değerinde meydana gelen değişiklikler y değerinde daha hızlı bir değişim meydana getirecektir. Ki bu özellik bu fonksiyonun küçük değişimlere karşı hassasiyetini göstermektedir. Dolayısıyla sınıflandırma problemlerinde kullanımı tercih edilmektedir.
- Sigmoid Değer = $\frac{\text{Küçük Değişimler}}{\text{Sınıflandırma}}$
- $[0,1]$ aralığında değer almaktadır.

- Sigmoid fonksiyonun dezavantajı bulunmaktadır. Şekil-1.6'de görüldüğü üzere bu fonksiyonun uç noktalarında x değerinde meydana gelen değişim y değerinde daha yavaş bir bir değişim meydana getirmektedir. Ki bu da bu bölgelerde türev değerlerinin çok küçük değer alması ve 0'a yakınsaması anlamına gelmektedir. Bu olay ise gradyanların kaybolmasına (vanishing gradient) neden olur. Gradyanların kaybolması öğrenme olayını azaltmakta ve minimuma indirmektedir. Yavaş bir öğrenme optimizasyonu yerel (lokal) minimuma takılmasına sebep olabilmektedir.

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

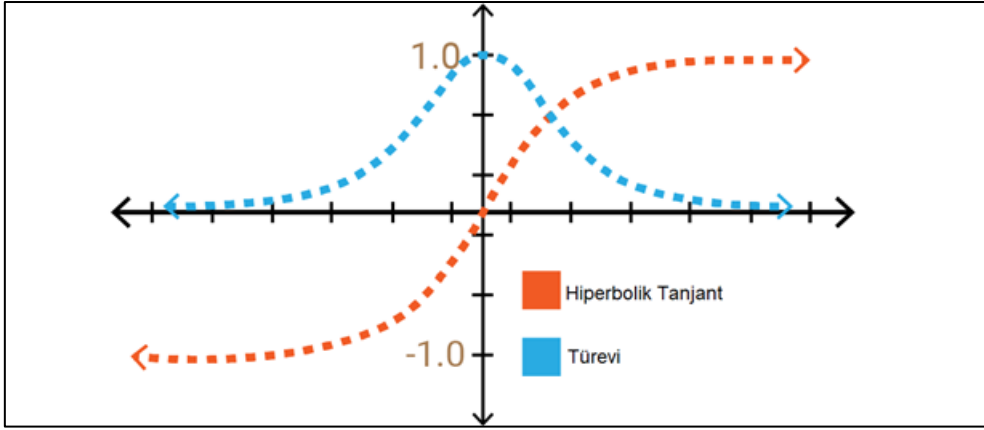


Şekil1. 6.Sigmoid Fonksiyon Grafiği

Hiperbolik Tanjant (Hyperbolic Tangent) ($\tanh(z)$)

- Sigmoid fonksiyonuna benzer bir fonksiyondur. Fakat bu fonksiyon $[-1,+1]$ aralığında yer almaktadır.
- Sigmoid fonksiyonuna göre avantajı Şekil-1.7'de görüldüğü üzere fonksiyonun daha dik olmasıdır. Bu durum hızlı öğrenme için ve sınıflama işlemlerinde daha geniş aralığa sahip olabilmesine sebep olmaktadır.
- Gradyan kaybolması (gradient vanishing) durumu devam etmektedir.
- Hassas noktalarda kullanılmaktadır.
- $[-1,1]$ aralığında değer almaktadır

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

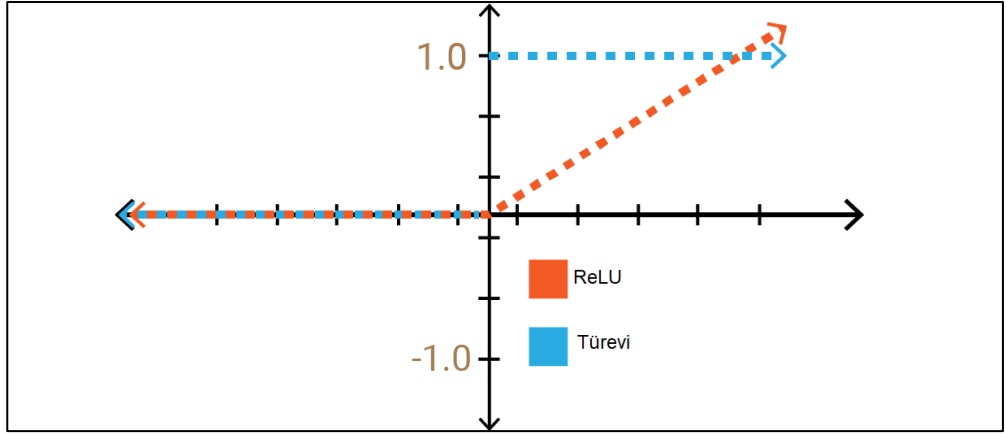


Şekil1. 7. Hiperbolik Tanjant Fonksiyonu Grafiği

Düzeltilmiş Doğrusal Birim (Rectified Linear Unit) (ReLU)

- ReLU çok iyi performansa sahip olmasından dolayı en çok tercih edilen yöntemdir.
- ReLU iyi bir tahmin edicidir.
- Özellikle 'gradyan kaybolması' olduğu durumlarda kullanılmaktadır.
- Şekil-1.8'de görüldüğü gibi negatif ekseninde 0 değeri alması ve daha az işlem yükü vermesi sebebiyle ağı hızlı çalışmasını sağlamaktadır.
- ReLU'nun işlem yükünü azaltarak hızlı çalışmasını sağlarken bu avantaj, ReLU fonksiyonunun bir zayıf noktasının ortaya çıkmasına da sebep olmaktadır. Bu zayıflık ReLU fonksiyonunun o bölgede öğrenme işleminin gerçekleşmemesi anlamına gelmektedir.
- $\max(0,z)$ aralığında değer almaktadır.

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

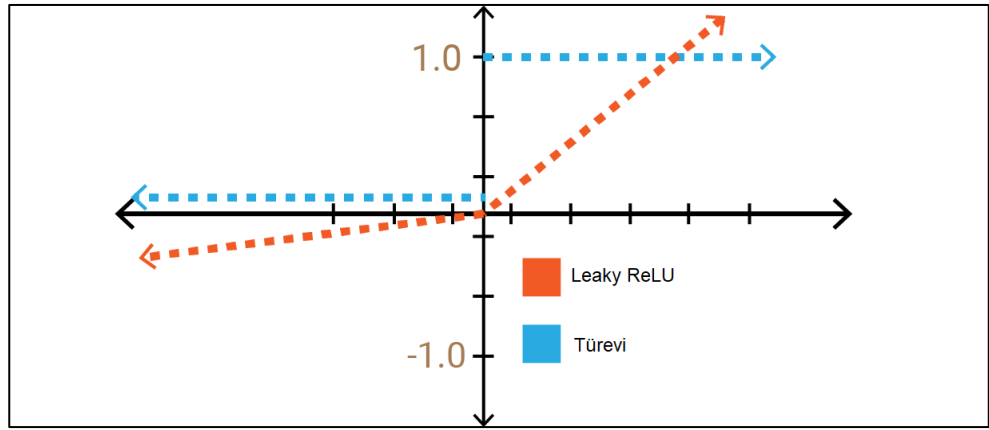


Şekil1. 8.Düzeltilmiş Doğrusal Birim Grafiği (ReLU)

Sızıntı ReLU Fonksiyonu (Leaky ReLU Function)

- Bu sızıntı değerine Şekil-1.9'da görüldüğü üzere 0.01 değeri veya sıfıra yakın herhangi değer verilirse fonksiyon Sızıntılı ReLU fonksiyonu olarak adlandırılmaktadır.
- Sızıntı ReLU fonksiyonunun amacı gradyan kaybolmasını önlemektir.
- $(-\infty, \infty)$ aralığında değer almaktadır.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0.01x, & x < 0 \end{cases}$$



Şekil1. 9.Sızıntı ReLU Fonksiyon Grafiği

Softmax Fonksiyonu (Softmax Function)

- Sınıflandırma problemlerinde iyi bir performans sergilemektedir.
- Sigmoid fonksiyonundan en önemli farkı çoklu sınıflandırma problemlerinde kullanılmaktadır. Girdilerin belirli bir sınıfa olan aitlik olasılığının 0-1 aralığında değerler alarak sınıflandırma yapmaktadır. Olasılıksal sınıflandırma gerçekleşmektedir.

3.2. YSA'da Öğrenme

İleri beslemeli bir YSA modeli için öğrenme, girdi verilerinin ağa girişiyle başlar. Eğitim için alınan girdi verileri ağırlıkları ile çarpılır ve yan değerlerinin eklenmesiyle elde edilen sonuç aktivasyon fonksiyonuna ulaşır ve çıktı değeri üretilir. Sistemden girdi verilerinin girmesiyle belirli işlemlerden geçen ve en sonunda ele alınan bu çıktı değeri, gerçek değer ile karşılaştırılır ve hata değeri bulunur. Bu hata değeri istenilen noktaya ulaşana kadar (hata değeri minimum olana kadar) ağırlıklar ve yan değerleri güncellenir. Hedef değeri ulaşıldıktan sonra ağırlık ve yan değerleri tutulur. Ağırlıkların güncellenerek optimum sonuca ulaşılana kadar geçen bu süre öğrenme olarak adlandırılmaktadır. Daha sonra eğitime girmeyen veriler test kümesinde test edilir. YSA test setindeki yeni karşılaştığı veriler ile iyi sonuçlar veriyorsa, YSA öğrenmiş anlamına gelir. Veri öğrenemediği problemlerde ise ağırlıklara atanan başlangıç değerlerinde, eğitim seti büyüklüğünde veya parametre / hiper-parametre seçimlerinde değişiklikler yapılarak sinir ağı tekrar eğitilebilir (Yılmaz, 2020).

En çok uygulanan öğrenme kuralları; Hebb Kuralı, Hopfield Kuralı, Delta kuralı, Dereceli alçalma (Gradient descent) kuralı, Kohonen öğrenme kuralları örnek verilebilir (Yılmaz, 2020).

YSA'da öğrenme verilen girdilere göre çıktı bulmaya dayanmaktadır (Yılmaz, 2020).. Öğrenme algoritmaları, makine öğrenmesi algoritmaları olan temel dört algoritmaya dayanmaktadır. Denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve takviyeli öğrenmedir. Makine öğrenmesi başlığında belirtilen öğrenme türlerinden kısaca bahsedilmiştir.

Öğrenme algoritması iki adımda gerçekleşmektedir. İlk adım ileriye doğru yayılım ve ikinci adım ise geriye doğru yayılım olarak adlandırılmaktadır. Aşağıda bu adımlardan kısaca bahsedilmektedir.

İleriye Doğru Yayılım (Forward Propagation)

İleriye doğru yayılım süreci girdi değerlerinin ağa sunulmasıyla, rastgele ağırlık değerlerinin atanmasıyla başlayan hesaplamaların ara katmanlara ve en son olarak çıktı katmanına gelerek hata değerlerinin hesaplanmasıyla biten süreci ifade etmektedir. Basit bir ileri doğru sinir ağında hesaplamalar aşağıdaki gibidir.

$$z = w^{[T]}x + b \quad (\text{Formül-1})$$

$g(z)$: aktivasyon fonksiyonu

şeklinde gerçekleştirilir.

Sinir ağında ise vektörleştirme yapılarak hesaplamalara Formül-2’de gösterilmektedir.

$$A^{[0]}=X \text{ (girdi değerleri)}$$

$$Z^{[l]}= W^{[l]T} A^{[l-1]}+ b^{[l]} \quad (\text{Formül-2})$$

$$A^{[l]}=g^{[l]}(Z^{[l]})$$

$A^{[1]}$: 1. tabakadaki aktivasyon fonksiyonunun çıktısını vermektedir. $A^{[0]}=X$ Girdi değişkenini vermektedir.

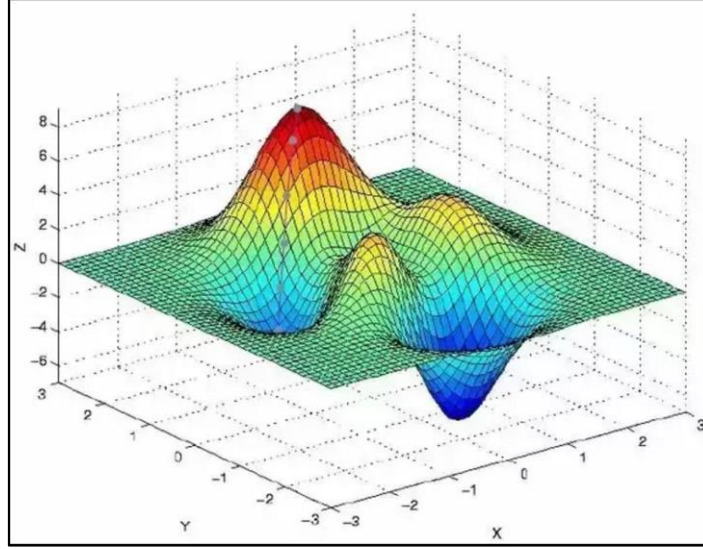
$Z^{[1]}$: 1. tabakadaki aktivasyon fonksiyonunun girdisini vermektedir. Ağırlıklar ile önceki çıktının çarpılması ve yan değerlerinin eklenmesiyle oluşan bir fonksiyondur.

l : İşlemin yapıldığı tabakayı göstermektedir.

Çıktı tabasında son tabakasında hesaplanan çıktı ile gerçek değer arasındaki fark kayıp fonksiyon (loss function) değerini vermektedir.

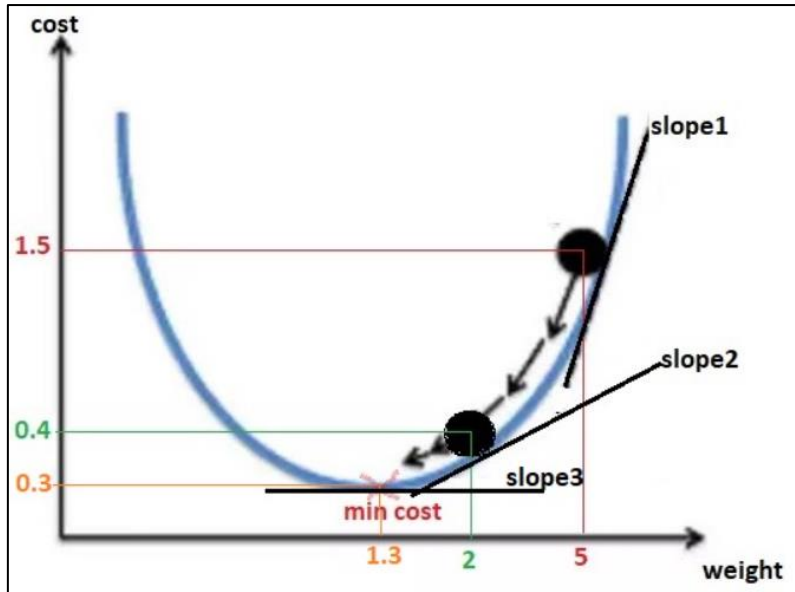
Geriye Doğru Yayılım (Backpropagation)

Ağı eğitmek için ağırlık ve yan değerlerinin en doğru değerlerinin tanımlanması gerekiyor. Bu sebeple maliyet fonksiyonu (cost function) tanımlanmalıdır. Her bir eğitim örneklerinden bir kayıp değeri (loss value) elde edilmektedir. Kayıp değeri, tahmin edilen değeri ve gerçek değerlerin farkının bir ölçüsü olarak ifade edilmektedir. Kayıp değerlerinin toplamı ise maliyet fonksiyonunu göstermektedir. Ağı geriye doğru yayılımında, eğitim ile hata değeri çıktı katmanında geriye doğru katmanlara aktarılarak ağırlıkların ve yan değerlerinin güncellenmesidir. Ağırlık ve yan değerleri, maliyet fonksiyonu minimum olana kadar güncellenir. Bu güncelleme ise dereceli alçalma (gradient descent) ile gerçekleştirilmektedir.



Şekil1. 10. Dereceli Alçalma Optimal Arama Görseli

Dereceli alçalma türev ile ifade edilmektedir. Matematiksel bir boyuttur. Maliyet fonksiyonu dışbükey bir fonksiyondur. Şekil-1.10'da dereceli alçalma uygulanan alan gösterilmektedir. Geri yayılım algoritması türev tabanlı bir algoritmadır (Portilla J., 2020). Fonksiyon bu alan üzerinde optimal adımlarla hedef noktaya ulaşmayı amaçlamaktadır. Bu arama işlemi türev ile gerçekleşmektedir. Türev ise eğimden gelmektedir. Eğim, bir doğrunun iki noktası arasındaki dikey değişimin yatay değişime oranı olarak bilinmektedir. Bu oran, yatay değişimde meydana gelen birim değişimin, dikey değişime etkisinin göstermektedir. Şekil-1.11'de eğim ile dereceli alçalmanın nasıl gerçekleştiği gösterilmektedir. Öğrenme oranı, optimal noktaya ulaşmadaki adım genişliği olarak adlandırılmaktadır. Ayarlanan adım genişlikleri ile optimal noktaya ulaşmaya çalışılmaktadır.



Şekil1. 11. Dereceli Alçalma Eğim ile İfadesi

Formül-3'de geri yayılım algoritması formülleri aşağıda verilmektedir.

$$dZ^{[l]} = dA^{[l]*} g^{[l]'}(Z^{[l]})$$

$$dW^{[l]} = \frac{1}{m} dZ^{[l]} A^{[l-1]T}$$

$$db^{[l]} = \frac{1}{n} np.sum(dZ^{[l]}, axis=1, keepdim=True) \quad (\text{Formül-3})$$

$$dA^{[l-1]} = W^{[l]T} \cdot dZ^{[l]}$$

3.3. YSA ile Sınıflandırma

YSA'da sınıflandırma sürecinde oluşturulacak modelin belirlenmesi için (Aladağ C.H., 2019):

- YSA'daki mimari yapıyı oluşturan tabaka sayılarının ve tabakalardaki birim sayılarının belirlenmesi
- Tabakalarda kullanılacak aktivasyon fonksiyonu çeşidinin belirlenmesi
- Öğrenme algoritmasının ve parametrelerinin belirlenmesi
- Verinin eğitim ve test kümelerinin büyüklerinin belirlenmesi
- En iyi modelin belirlenmesinde kullanılacak performans ölçütünün belirlenmesi

modelin performansı açısından büyük önem taşımaktadır. Bunlar arasında çok farklı seçenekler sunan mimari yapının belirlenmesi en zor seçenek olarak görülmektedir. Bunun nedeni, en fazla seçim olasılığı mimari seçimi probleminde ortaya çıkmaktadır (Aladağ, 2009). Önceki kısımlarda YSA'nın üç temel bileşenlerinden bahsedilmişti. Bu nedenle, burada YSA yönteminin performansını etkileyen diğer iki etken olan eğitim ve test kümesi büyüklükleri ve performans ölçütlerinin belirlenmesinden kısaca bahsedilecektir.

Eğitim ve Test Kümesi Büyüklüğü

YSA yaklaşımı modelleme için kullanıldığında, genellikle danişmalı öğrenme kullanımı tercih edilmektedir. YSA bu süreçte veri ya eğitim ve test kümesi olarak ikiye veya eğitim, doğrulama ve test kümesi olarak üçe ayrılmaktadır.

- *Eğitim Seti (Training Set)*: Model parametreleri olan ağırlıkların en iyi değerlerinin hesaplanması için kullanılır. Ağırlıkların en iyi değerlerinin bulunması sürecine ağırlık eğitimi adı verilir.
- *Doğrulama Seti (Validation Set)*: Düzeltme için kullanılmaktadır. İsteğe bağlı olarak kullanımı tercih edilmektedir. Eğitim setini test eder. Her döngüden sonra veri hızlıca test edilir. Eğer bir problem varsa geriye dönüp düzeltilebilmektedir. Aşırı öğrenme (overfitting) problemi hakkında bilgi verir ve test seti çalıştırılmadan

düzeltilmektedir. Makine Öğrenmesinin hassas noktasıdır. YSA yöntemi veri temelli olduğundan doğrulama seti kullanılması bazı çalışmalarda tercih edilmektedir.

- *Test Seti (Test Set)*: Test seti, eğitim setine girmeyen gözlemlerden oluşur. Eğitimde hiç kullanılmayan yani, modelin hiç görmediği test kümesi gözlemleri kullanılarak YSA modelinin tahmin performansı değerlendirilir. Bir başka ifadeyle, YSA modelinin genelleştirme yeteneğinin ölçülmesinde kullanılır. Literatürde belirtilen kümelerin büyüklüklerinin belirlenmesinde genel kurallar bulunmamaktadır. Analiz edilecek veriye göre en uygun küme büyüklüklerinin belirlenmesi YSA yönteminin performansını doğrudan etkileyecektir (Günay vd., 2007).

Performans Ölçütü

YSA modelinin analiz edilen verinin örüntüsünü ne kadar iyi öğrendiğini ölçmek amacıyla çeşitli performans ölçütleri kullanılmaktadır (Aladağ, 2009). Bu çalışmada kullanılan problem ikili sınıflandırma problemidir. Bu sebeple performans ölçütü olarak çapraz entropi kayıp değeri (binary_cross_entropy loss) değerleri veya doğruluk (accuracy) değerleri alınmaktadır. Çapraz entropi değeri veya log loss değeri, tahminin gerçek değerden ne kadar uzaklaştığını gösteren bir belirsizlik ölçütüdür. Doğruluk değeri ise tahmin edilen değerlerin gerçek değere eşit olduğu tahmin sayısı olarak bilinir. Doğruluk değeri ikili sınıflandırmada yorumlamada daha az tercih edilmektedir. Çapraz belirsizlik kayıp hesabı daha iyi yorumlanabilmektedir.

$$\text{Doğruluk} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Log-loss} = y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$

Yapılan bu çalışmada kullanılan performans ölçütü, ikili çapraz doğrulama kayıp değeri veya log loss değeri olarak alınmıştır. En iyi YSA modelinin belirlenmesinde kullanılacak performans ölçütü yine doğrudan rol oynayan önemli bir etkidir.

3.4. YSA Çeşitleri

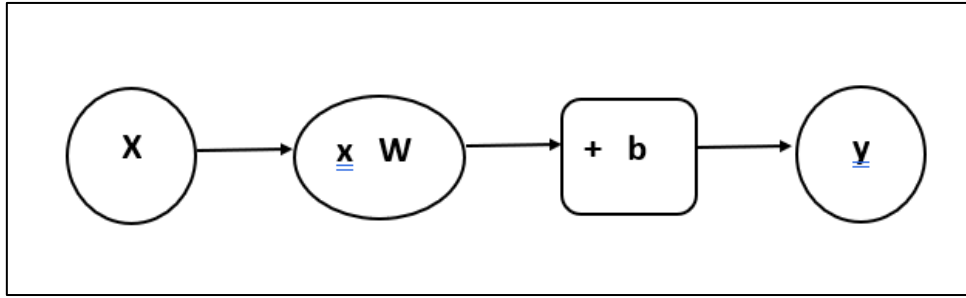
YSA modelleri, kullanılan mimari yapısına, nöronlar arasındaki bağlantılara ve ağ içindeki bilginin akış yönüne göre farklılık göstermektedir. En temel sınıflandırmada, ağ mimarilerinin yapılarına göre temel olarak, ileri beslemeli (feed forward) (Yolcu vd., 2015) ve geri beslemeli (recurrent) (Eğrioğlu vd., 2015) sinir ağları olmak üzere iki farklı sinir ağı vardır.

Literatürde kullanılan tek katmanlı algılayıcı sinir ağları (perceptron) (Yılmaz A., 2020), çok katmanlı algılayıcı sinir ağları (multi-perceptron neural network), ileri beslemeli sinir ağları, geri beslemeli sinir ağları, uzun-kısa terimli hafıza (long-short term memory/LSTM) sinir ağları

kohonen sinir ağı, otomatik kodlayıcı (autoencoder) sinir ağı (Atınç Y, Kaya U., 2020), evrişimsel sinir ağı (convolutional neural network) (Atınç Y, Kaya U., 2020), Dayanıklı YSA (robust artificial neural network) (Aladağ,2014) vb. birçok sinir ağı çeşidi de örnek olarak verilebilir. Belirtilen YSA çeşitleri, temel özellikleriyle kısaca aşağıda verilmiştir.

Tek Katmanlı Algılayıcı (Perceptron)

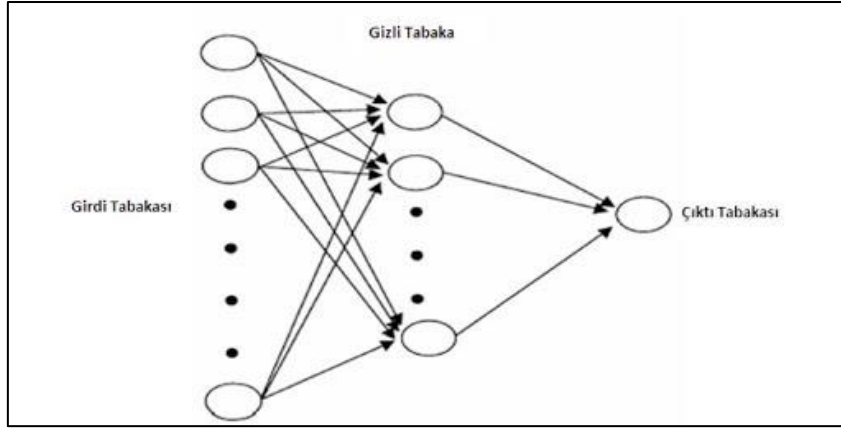
- En basit sinir ağı modelidir. Şekil-1.12’de gösterilmiştir.
- Temel olarak eğitilebilecek tek bir sinir ağından oluşmaktadır.
- Sadece girdi ve çıktı katmanlarından oluşmaktadır.
- Yeterince açıklayıcı bir model olmaması sebebiyle kompleks problemlerde kullanılmamaktadır.
-



Şekil1. 12.Perceptron Modeli

İleri Beslemeli Sinir Ağları (Feed Forward Neural Network):

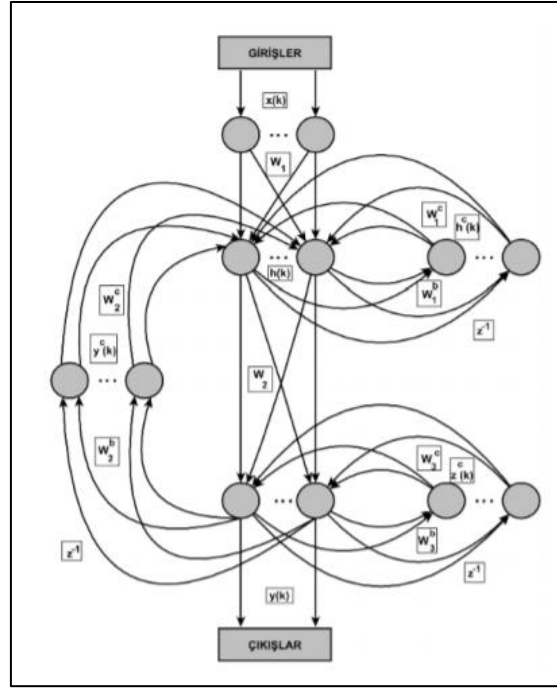
- Şekil-1.13’de gösterildiği üzere bağlantıların sadece ileri yönde beslendiği sinir ağlarıdır.
- Girdi değerleri bir ağdan geçirilerek bir çıktı değeri elde edilir. Elde edilen çıktı değeri gerçek değerler ile karşılaştırılarak hata değeri elde edilir. Ağ değerleri hataya bağlı olarak değiştirilir ve bu şekilde en doğru sonucu veren model seçilir.



Şekil1. 13. İleri Beslemeli Sinir Ağı

Geri Beslemeli Sinir Ağları (Feedback Neural Network)

- Geri beslemeli ağlarda, geri besleme işlemi bulunmaktadır. Çıkış ve ara katmanlardaki çıkışların bir önceki katmanlara geri beslenmesiyle oluşan ağ yapısıdır (Yılmaz,2020).
- Geri Beslemeli yapılarda sonuç sadece o andaki girdiye değil diğer girdilere de bağlı olarak çalışmaktadır
- Şekil-1.14’de görüldüğü üzere t anındaki girdi verilerinin yanında, t-1 anından gelen gizli katman sonuçları da gizli katmanın t anındaki girdisidir.
- Geri beslemeli sinir ağları bir belleğe sahiptir.



Şekil1. 14.Geri Beslemeli Sinir Ağı

Derin Sinir Ağları (Deep Neural Network)(DSA)

YSA'nın ikiden fazla anlamlı gizli tabakaya sahip veri modeli "Derin Sinir Ağı" (DSA) olarak adlandırılmaktadır. "Derin" ifadesi, sinir ağlarının içerdiği katman sayısını ifade etmektedir. Elimizdeki veri basit yapıdan karmaşık yapıya doğru ilerledikçe sinir ağımızdaki tabaka sayısı artacaktır ve karmaşıklığı da artıracaktır. Derinlik, kompleks problemleri çözme becerisini arttıracaktır.

Öğrenme algoritmaları son yıllarda büyük veri ve yüksek performanslı bilgisayarlarla birlikte yüksek doğruluk elde etmektedir. Daha az varsayım ve insan gücüyle veriden otomatik olarak öğrenmektedir. DSA'nın altında yatan en güçlü özelliği budur. Bu sayede karmaşık işlemleri çözebilmektedir. Büyük verinin karmaşık işlemlerini çözerek verideki örüntüyü veya düzeni anlayabilmektedir. DSA'da her katman önceki girdideki önemli özellikleri ayırt etmektedir. Ve bu özellikleri belirli ağırlıklar ile diğer katmana aktarmaktadır (Alpaydın, 2020).

Derin sinir ağlarının kısa tarihi aşağıdaki gibi özetlenmektedir (Elmas, 2018),(Yılmaz,2020):

- 1986'da R. Dechter bir makalede Derin Öğrenme terimini ilk kez kullanmıştır.
- 1980'de Kunihiko Fukushima ilk Evrişimli sinir ağının olarak bilinen Neocogitron'u tanıttı. Fakat geri yayılım algoritması eksik olması sebebiyle bu yıllarda kullanılmadı.
- 1985'de Hilton ve Sejnowski Boltzman Makinesi geliştirmişlerdir.
- *Rumelhart – McClelland 1986 yılında çok katmanlı ağ yapılarında geri yayımlı öğrenme algoritmasını ortaya koymuştur.*

- 1986'da Smolensky Harmonium'u tanıttı ve ayrıca Jordan Yinelemeli (Recurrent) Sinir Ağını tanıtmıştır.
- 1990'da Yann LeCun vd. LeNet ağını tanıtarak derin sinir ağlarının ilk uygulamasını yapmışlardır.
- 1997'de Schuster ve Paliwal Çift Yönlü Yinelemeli Sinir Ağını tanıtmışlardır.
- 2009'dan sonra Deng ve Hinton tarafından düzenlenen NIPS Çalıştayları ve 2010'dan sonra düzenlenen ImageNet yarışmaları düzenlenmiştir. Ve DSA kullanımının yaygınlaşmasına katkı sağlamıştır.

DSA kullanım alanları aşağıdaki gibi özetlenebilmektedir (Elmas, 2018)

- *Görüntü işleme*
- *Video işleme*
- *Örüntü tanıma*
- *Nesne tanıma*
- *Ses tanıma*
- *Doğal Dil İşleme*
- *Sinyal tanıma*
- *Otomatik araç denetimi*
- *Fizyolojik işaretleri izleme, tanıma ve yorumlama*

DSA Çeşitleri ve Uygulama Alanları

- ***Evrişimli Sinir Ağları:*** *Örüntü tanıma, video analizi, doğal dil işleme*
- ***Yinelemeli Sinir Ağları:*** *Konuşma tanıma, el yazısı tanıma*
- ***Uzun Kısa Süreli Hafıza:*** *Doğal dil işleme, el yazısı tanıma, konuşma tanıma, hareket tanıma, görüntü altyazı*
- ***Çok Katmanlı Algılayıcılar:*** *Tahminleme, öngörü, sınıflandırma, optimizasyon*

Geleneksel Öğrenmede (Traditional Learning)'te; veri eklendikçe performans artış gösterir fakat belli bir süre sonra sabit kalır. Bu durum verinin nasıl kullanılacağını bilmiyor gibi bir davranış gösterir. Fakat derin sinir ağları ile veriler eklendikçe, eğitildikçe daha iyi öğrenecektir ve yüksek performans gösterecektir. Belirli bir noktadan sonra performans artışı daha az olacaktır. DSA çeşitleri aşağıda verilmektedir.

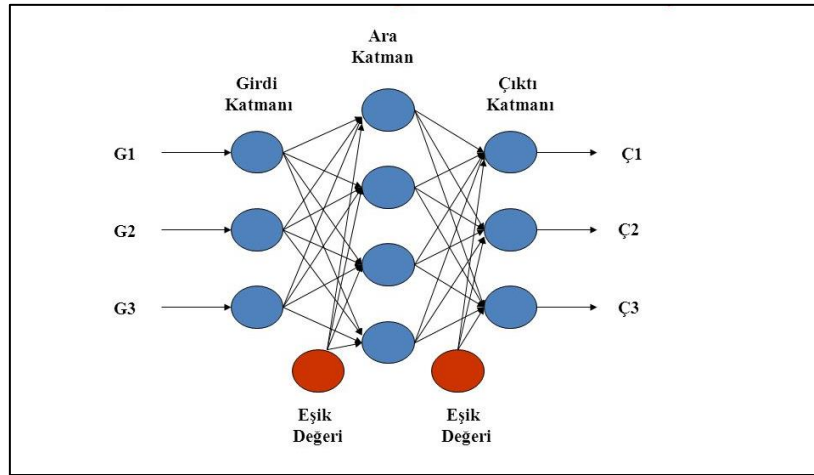
- Çok katmanlı Algılayıcı (Multilayer Perceptron)
- Evrişimsel Sinir Ağları (Convolutional Neural Network)

- Yinelemeli Sinir Ağları (Recurrent Neural Network)
- Uzun-Kısa Terimli Hafıza Sinir Ağları (Long-Short Term Memory Neural Network)

Çok Tabakalı Algılayıcı (Multilayer Perceptron)

Tek tabakalı algılayıcılar, doğrusal problemleri çözebilme yeteneğine sahipken, doğrusal olmayan ilişkileri çözememektedir. Günlük hayatta karşılaştığımız çoğu problem doğrusal değildir. Bu sebeple doğrusal olmayan XOR problemleri çözmek amacıyla yapılan çalışmalar sonucunda çok tabakalı algılayıcılar geliştirilmiştir. David Rumelhart, Geoffrey Hinton and Ronald Williams tarafından 1986’da yayınlanan makale ile bu model duyurulmuştur. Bu model geriye yayılım (backpropagation) temeline dayandırılmıştır (Yılmaz, 2020). Bu model Delta Kuralı olarak bilinen, hatayı minimize etmek için beklenen değer ve gerçek değer arasındaki farka dayanan, öğrenme kuralı tabanlıdır.

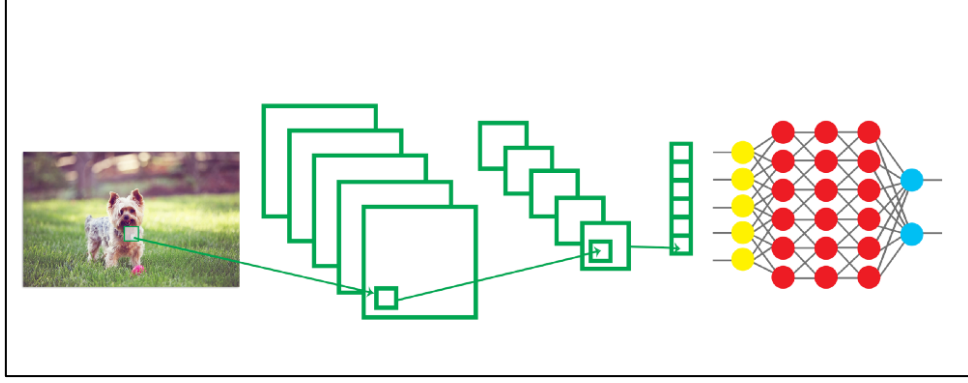
Çok katmanlı algılayıcılar, girdi tabakası, gizli tabakalar ve çıktı tabakalarından oluşmaktadır. Tek yönlü akış mevcuttur. Sinyal girdi katmanından, çıktı katmanına doğru ilerlemektedir. Bu sebeple bu mimari İleri Beslemeli Sinir ağı modeline örnektir. Öğrenme algoritması, 3.2. bölümde bahsedilen YSA’da Öğrenme bölümünde anlatılmaktadır. Şekil-1.15’de çok katmanlı sinir ağı mimarisi verilmiştir.



Şekil1. 15. Çok Katmanlı Algılayıcı

Evrışimli Sinir Ağları (Convolutional Neural Network /CNN)

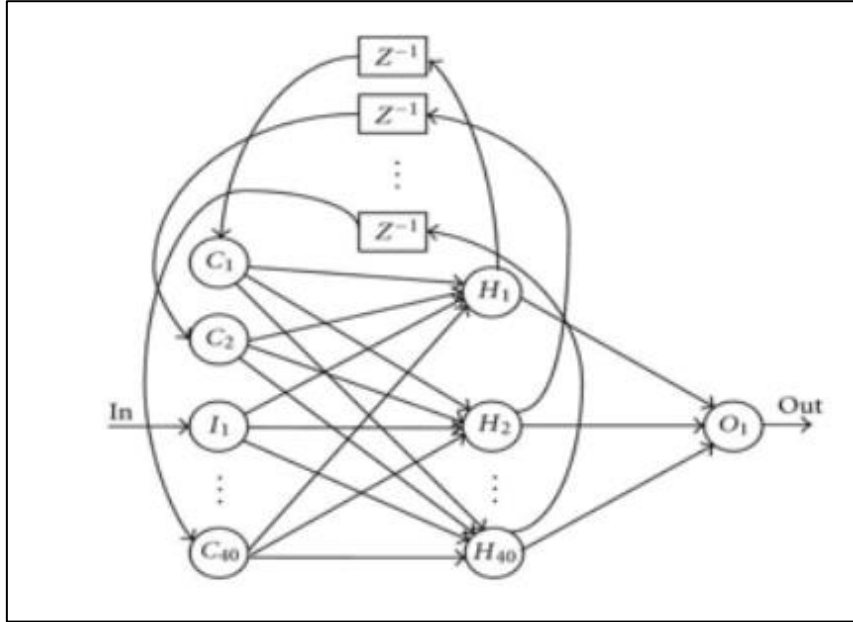
Evrışimli sinir ağları, derin öğrenmenin bir alt dalıdır. Görüntü analizinde yüksek başarılarından dolayı genellikle görsel bilginin analiz edilmesinde kullanılmaktadır. Yaygın kullanım alanları resim ve video tanıma, öneri sistemleri resim sınıflandırma, tıbbi görüntü analizi ve doğal dil işleme olarak sıralanabilir. Şekil-1.16’da nasıl işlendiği hakkında genel bir gösterim verilmiştir.



Şekil1. 16.Evrişimsel Sinir Ağı

Yinelemeli Sinir Ağları (Recurrent Neural Network / RNN)

Jeff Elman “Basit Tekrarlayan Ağ” (Simple Recurrent Network) adıyla 1990 yılında yinelemeli sinir ağını ilk olarak önermiştir. Yinelemeli sinir ağları, derin öğrenme mimarilerindedir. Yinelemeli sinir ağlarında, sonuç sadece o andaki girdiye değil diğer girdilere de bağlıdır. t-1 anındaki veriyi de girdi olarak almaktadır. Bir hafızaya sahiptir. Dizileri analiz etmek ve sınıflandırmak amacıyla kullanılmaktadır. En çok Doğal Dil İşleme süreçlerinde kullanılmaktadır. Görüntü yazısı, duygu sınıflandırma, video sınıflandırma örnek verilebilir (Elmas, 2018). Şekil-1.17’de yinelemeli sinir ağı modeli verilmiştir.



Şekil1. 17.Yinelemeli Sinir Ağı Modeli

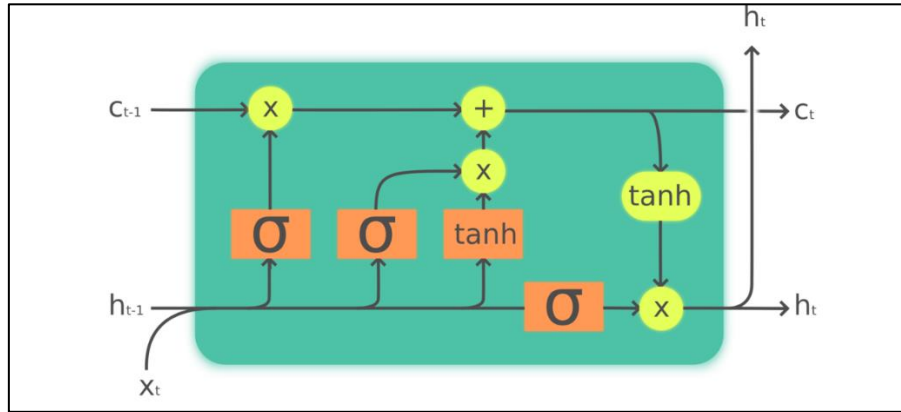
Uzun-Kısa Süreli Hafıza (Long – Short Term Memory / LSTM)

Uzun Kısa Süreli Hafıza, derin öğrenme alanında kullanılan Yinelemeli Sinir ağı mimarisinin bir çeşididir. Burada “hafıza, bellek” kavramı önem kazanmıştır. İleride kullanabilmek adına

hafızasında verileri saklamaktadır. Yinelemeli sinir ağından farklı olarak uzun kısa süreli hafıza modeli, veriyi ne kadar uzun süre saklaması gerektiğine ne zaman çıktısı alınmalı veya ne zaman unutulması gerektiği gibi bilgileri kontrol eden özel bir birime sahiptir.

Şekil-1.18'da görüldüğü üzere LSTM, standart ileri beslemeli sinir ağlarında farklılık göstermektedir. Geri bildirimleri bulunmaktadır.

Konuşma ve video işleme alanlarında kullanımı tercih edilmektedir.

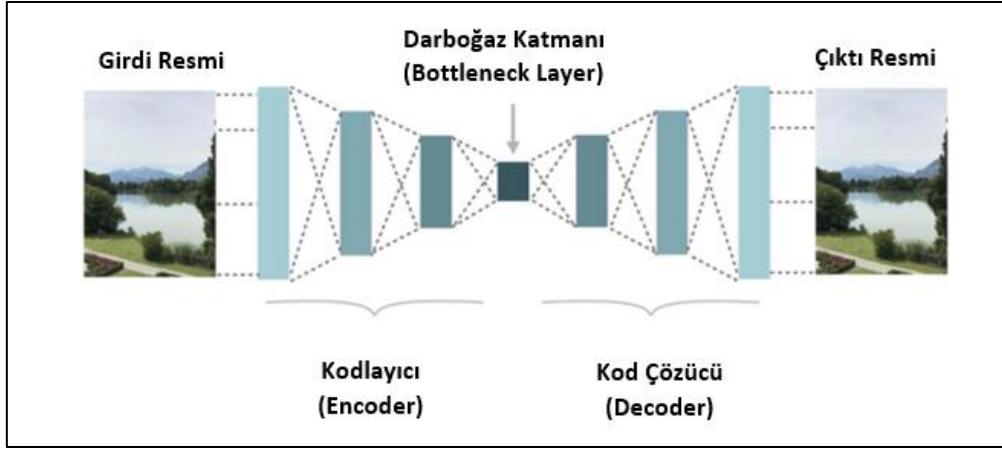


Şekil1. 18.Uzun-Kısa Terimli Hafıza Modeli

Oto Kodlayıcı (Autoencoder)

Oto kodlayıcı isminde de anlaşıldığı üzere herhangi bir veriyi koda dönüştürmeyi otomatik olarak öğrenmeyi amaçlamaktadır. Kodlayıcı (encoder) ve kod çözücü (decoder) olmak üzere iki kısımdan oluşmaktadır. Oto kodlayıcı (autoencoder) modelinin amacı, verilen veriyi sıkıştırıp olabildiğince az kayıpla tekrar üretmektir. Üretilen kod ise orta tabakanın çıktısı alınarak elde edilmektedir. Bu katmana darboğaz tabaka (bottleneck layer) denir. Darboğaz tabaka elde edilecek kodun büyüklüğünü belirlemektedir. Bu katman ne kadar fazla sıkıştırma yapılacağını belirlemektedir. Şekil-1.19'de bu sürecin genel gösterimi verilmiştir.

Öznitelik çıkarma (feature extraction), boyut indirgeme (dimension reduction), gürültü giderme (denoising), resim renklendirme, yeni veri üretme alanlarında kullanılmaktadır.



Şekil1. 19.Oto Kodlayıcı

3.5. Parametre ve Hiper-parametreler (Parameter and Hyperparameters)

Modele dahil olan ve değeri veriden tahmin edilebilen değişkenlere parametre denir. Hiper-parametreler ise, model parametrelerini kontrol eden model dışı parametrelerdir. Hiper-parametreler kontrol parametreleridir. Doğru hiper-parametreleri seçmek modelin performansını doğrudan etkilemektedir. Hiper-parametre seçimleri el ile ayarlanabilir parametrelerdir. Uzman kişilerin deneyimlerine dayanarak, deneme-yanılma yöntemleriyle seçilmesi tercih edilmiştir. Fakat doğru hiper-parametreleri seçilebilmesi amacıyla bir algoritma ortaya konulması bir ihtiyaç haline gelmiştir. Hiper-parametre seçimi bir optimizasyon problemi olarak görülmektedir.

DSA'yı geliştirmek için gerekenler aşağıdaki gibi özetlenebilir:

1. Parametreleri organize etmek
2. Hiperparametreleri organize etmek

DSA'nın parametreleri, tabakalardaki ağırlıkların değerleri (weights)($w^{[l]}$) ve tabakalardaki yan değerleri (bias)($b^{[l]}$) olarak gösterilmektedir. Hiper—parametreler ise aşağıdaki gibi gösterilmektedir.

- *Öğrenme oranı (learning rate)*: Optimizasyon problemlerinde, optimal noktaya ulaşmak için gereken adım boyutudur. Başka bir ifadeyle, model geri yayılım algoritması ile eğitilirken, minimum kayıp değerine ulaşmak için gerekli adım genişliğidir. Önemli bir hiper-parametredir. Adım genişliği büyük tutulması halinde optimal nokta kaçırılabilir, aksine adım sayısı küçük tutulması halinde ise optimal noktaya ulaşmak zaman açısından maliyet gerektirebilir. Bu sebeple öğrenme oranının seçimi model performansını doğrudan etkileyecektir.
- *İterasyon sayısı (Number of iterations)* : *Geçiş sayısı olarak adlandırılmaktadır.*

- *Gizli Tabaka sayısı (Number of layers)*: Modelde bulunması gereken gizli tabaka sayısını göstermektedir.
- *Gizli Tabakalardaki birim/nöron sayısı (number of units)*: Her gizli tabakada bulunması gereken nöron sayısını ifade etmektedir.
- *Aktivasyon fonksiyonu seçimi (Number of activation function)*: Modelin sahip olduğu aktivasyon fonksiyonu çeşidini ifade etmektedir.
- *Bırakma Oranı (Dropout Rate)*: Tam bağlı tabakalarda, önemsiz bilgileri düşürmek, model performansını etkilediği gözlemlenmiştir. Başka bir ifadeyle, belirli bir eşik altındaki değere sahip olan nöronun bırakılması başarıyı arttırdığı gözlemlenmektedir. Bu sebeple Bırakma Oranı kullanılmaktadır. Genel olarak 0.5 olarak alınmaktadır fakat veri setine göre değişiklik göstermektedir.
- *Eğitim Tur Sayısı (Number of Epochs)*: Tüm eğitim örneklerinde bir kez ileri ve geri geçişi eğitim tur sayısı olarak adlandırılmaktadır.
- *Parti Boyutu (Batch Size)*: İleri ve geri geçişte eğitimde alınan örnek sayısı “parti boyutu” olarak adlandırılmaktadır.
- *Ağırlık Başlatma (Weight Initialization)*: Sinir ağı tanımlanırken başlangıçta rastgele ağırlık (ve yan değerleri) atanmaktadır. Bu çalışmada “normal dağılımdan” alınmış rastgele başlangıç ağırlık değerleri alınmıştır.
- *Momentum*: Momentum, dereceli alçalmada iniş hızını ve salınımını ayarlamaktadır. Float hiper-parametresi ≥ 0 ilgili yönde dereceli inişi hızlandırır ve salınımları azaltır.
- *Mini-parti Boyutu (Mini Batch Size)*: Veri setinde bulunan tüm verileri aynı anda işlemek zaman ve bellek açısından zor olmaktadır. Çünkü öğrenmenin her iterasyonda geriye yayılım algoritması uygulanacaktır ve ağırlıklar güncellenecektir. Böylece veri sayısı arttıkça hesaplama da artacaktır. Bu sebeple mini-parti boyutu ile veri seti küçük gruplara ayrılarak öğrenme işlemleri bu küçük gruplar üzerinden yapılmaktadır. Modelin aynı anda kaç veriyi işleyeceğini göstermektedir.
- *Düzeltilme parametreleri (Regularization Parameters)(L1,L2)*: Aşırı öğrenmeyi (overfitting) engellemek amacıyla kullanılan bir tekniktir.

Literatürde bazı hiper-parametre optimize yöntemleri geliştirilmiştir. Bu yöntemlerden bazıları aşağıda verilmektedir:

Izgara Arama (Grid Search)

Verilen bir model için optimal değerleri tanımlamak amacıyla hiper-parametre ayarı (hyperparameter tuning) için geleneksel ve yaygın bir şekilde kullanılan bir yöntemdir. Izgara arama mümkün olan tüm kombinasyonları deneyerek çalışır. Bu durum ise daha fazla zaman alacağı anlamına gelir. Fakat mümkün olan tüm kombinasyonları deneyeceği için daha kesin sonuç verecektir.

Rastgele Arama (Random Search)

İnşa edilen model için en iyi çözümü bulan hiper-parametre değerlerini rastgele kombinasyonları ile arar. Rastgele aramanın dezavantajı, bazen arama alanındaki önemli noktaları gözden kaçırabilmesidir. Fakat Izgara arama yöntemi kadar zaman almaz.

Bayesçi Yaklaşım (Bayesian Approach)

Bayesçi yaklaşım, objektif fonksiyona bağlı olarak en iyi sonuçları veren hiper-parametreleri seçerek gerçek amaç fonksiyonunu değerlendirmek için bir sonraki hiper-parametre setini bularak çalışır. Geçmiş denemelere dayalı olarak değerlendirmek için en iyi hiperparametre kümesini düşününen bir yaklaşım olmasından dolayı bayes modeline dayalı yöntemlerin daha kısa sürede daha iyi hiper-parametre seçimi yapabilmektedir.

Alternatif Hiper-parametre Optimizasyon Yöntemleri

Hyperopt

Hyperopt, James Bergstra tarafından geliştirilen hiper-parametre optimizasyonu için güçlü bir Python kütüphanesidir. Belirli bir model için en iyi parametreleri elde edilmesine izin veren parametre ayarı için bir Bayes optimizasyonu biçimi kullanır. Bir modeli büyük ölçekte yüzlerce parametre ile optimize edebilir.

Scikit Optimize

Scikit-optimize, hiper-parametre optimizasyonu için başka bir açık kaynaklı Python kütüphanesidir. Sıralı model tabanlı optimizasyon için çeşitli yöntemler uygular. Kütüphane kullanım açısından kolaydır. Bayes optimizasyonu ile hiper-parametre ayarı için kullanılacak genel bir araç takımı sağlar. Ayrıca scikit-learn kütüphanesi tarafından sunulan makine öğrenimi algoritmalarının hiper-parametrelerin ayarlanması için destek sağlar.

Optuna

Optuna, hiperparametrelerin arama alanını otomatikleştirmek amacıyla Bayeşçi yaklaşım kullanarak hiperparametre optimizasyonu için başka bir açık kaynaklı Python kütüphanesidir. Bu kütüphane, Preferred Networks adlı bir Japon AI şirketi tarafından geliştirilmiştir.

4. DOĞRUSAL VE DOĞRUSAL OLMAYAN REGRESYON ANALİZİ

Regresyon analizi, iki veya daha fazla değişkenler arasındaki ilişkinin modellenmesi amacıyla kullanılan ve bu ilişkinin modellenmesi ile tahmin yapabilen istatistiksel bir yöntemdir (Montgomery D.C., Peck E., Vining G.G.,2013). Burada ilişki kavramı ile ilişkinin fonksiyonel şeklini, ilişki yönünü ve derecesini vurgulamaktadır. İlişkinin fonksiyonel şekli doğrusal, doğrusal olmayan ve eğrisellik tanımının belirlenmesidir. İlişki yönü, ilişkili değişkenlerin birlikte aynı yönde mi farklı yönde mi artış gösterdiğini belirtmektedir. İlişki derecesi ise değişkenler arasındaki ilişkinin kuvvetini belirtmektedir (Erilli, N.A., 2018).

İlişkinin fonksiyonel şekli regresyon analizinin yanı sıra korelasyon analizi ile de çözümlenebilmektedir. Ayrıca regresyon analizine benzer olarak korelasyon analizi, ilişkinin yönü ve kuvveti hakkında da bilgi vermektedir. Fakat, regresyon analizinde değişkenler arasında sebep-sonuç ilişkileri hakkında bilgi verirken, korelasyon analizi değişkenler arasındaki sebep sonuç ilişkisini hakkında bilgi veremez. Bu çalışmada regresyon analizinin fonksiyonel işlevlerinden doğrusal regresyon analizi ve doğrusal olmayan regresyon analizinden bahsedilecektir.

Doğrusal Regresyon Analizi

Bağımlı değişken ile bağımsız değişkenler arasında doğrusal ilişki söz konusu ise bu regresyon analizi doğrusal regresyon analizi olarak adlandırılmaktadır.

Basit Doğrusal Regresyon Modeli

Yanıt değişkeni y ile doğrusal ilişkiye sahip tek x bağımsız değişkenin bulunduğu regresyon modeline basit doğrusal regresyon modeli denilmektedir.

$$y = \beta_0 + \beta_1 x + \varepsilon$$

β_0 ve β_1 regresyon katsayıları olarak adlandırılmaktadır. β_0 , sabit değer ve β_1 , x değişkeninin katsayısını ifade etmektedir. ε ise rastgele hata bileşeni olarak adlandırılmaktadır. β_1 , x

üzerindeki 1 birimlik deęişiklik y’de ne kadarlık bir deęişim yarattığını göstermektedir. β_0 ise $x=0$ olduğunda y’nin ortalama deęerini vermektedir.

Çoklu Doğrusal Regresyon Modeli

Yanıt deęişkeni y ile doğrusal ilişkiye sahip iki veya daha çok bağımsız deęişkenin bulunduğu regresyon modeline çoklu doğrusal regresyon modeli denilmektedir.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon$$

$\beta_0, \beta_1 \dots \beta_n$ regresyon katsayıları olarak adlandırılmaktadır. β_1, x_1 üzerindeki 1 birimlik deęişiklik y’de ne kadarlık bir deęişim yarattığı, β_n, x_n üzerindeki 1 birimlik deęişikliğin y’de meydana getirdiği deęişimi göstermektedir. β_0 ise $x=0$ olduğunda y’nin ortalama deęerini vermektedir.

Parametre Kestirimleri

En Küçük Kareler Yöntemi

En küçük kareler kestiricileri, β_0 ve β_1 model parametrelerinin yansız kestiricileridir. En küçük kareler kestiricileri en iyi doğrusal yansız kestiricilerdir.

En Çok Olabilirlik Kestirimi

En küçük kareler yöntemi bir doğrusal regresyon modelindeki parametrelerin ε hatalarının dağılımından bağımsız olarak kestirilmede kullanılmaktadır. Hipotez test etme ve güven aralığı oluşturma gibi dięer istatistiksel işlemlerde hataların normal dağıldığı varsayılmaktadır. Eđer hataların dağılım biçimi biliniyorsa parametre kestirimi için alternatif bir yöntem de en çok olabilirlik yöntemidir.

Doğrusal Olmayan Regresyon Analizi

Doğrusal regresyon analizi oldukça fazla kullanılan, esnek ve birçok yönden güçlü bir istatistiksel analizdir. Bununla birlikte, doğrusal regresyon modelleri bazı durumlarda kullanımı tercih edilmemektedir. Açıklayıcı deęişkenler ile açıklanan deęişkenler arasında doğrusal olmayan ilişkiler olan problemler ile karşılaşılabilir. Bu modellere en küçük kareler yöntemi uygulandığında oluşan denklemler doğrusal değildir ve çözümü zordur (Montgomery vd., 2013). Fakat doğrusal olmayan modellerde deęişkenler bazı dönüşümlerle doğrusallaştırılarak model oluşturulabilmektedir. Bu yaygın modeller:

- *Logaritmik Modeller*

- *Tam Logaritmik Dönüşümler:* Her iki tarafın logaritması alınarak tanımlanır. Açıklanan değişken ve açıklayıcı değişken(lerin) logaritması alınır. $\log x$, $\log y$ olarak tanımlanır.
- *Yarı Logaritmik Dönüşümler :*
 - *Log-Lin Dönüşümü :* Bu dönüşüm modeli, bağımsız değişkenlerdeki 1 birimlik değişimin bağımlı değişkendeki yüzde büyümeyi göstermektedir. Diğer bir ifadeyle, bağımsız değişkendeki 1 birimlik artışın, bağımlı değişkendeki yüzde değişimini göstermektedir. Sadece bağımlı değişkenin logaritması alınmaktadır. $\ln y$ olarak tanımlanabilir.
 - *Lin-Log Dönüşümü:* Bu dönüşüm modeli, açıklayıcı değişken %1 arttığında açıklanan değişkendeki değişimi göstermektedir. Sadece bağımsız değişkenlerin logaritması alınmaktadır. $\ln x$ olarak tanımlanabilir.
- *Ters Modeller :* Bağımlı değişken ile bağımsız değişkenler arasında ters ilişki olduğu durumda kullanılan bir modeldir. Ters ilişkili olan değişkene x değişken olarak ifade edersek, $1/x$ dönüşümü ile model kurulabilmektedir.
- *Polinom Regresyon Modelleri :* Doğrusal olmayan ilişkileri yakalamak için kullanılan bir yöntemdir. Hipotezini kurduğumuz fonksiyon verilere uymuyorsa doğrusal regresyon analizi doğru bir seçenek değildir. Hipotez fonksiyonunun eğrisi doğrusal değil ise kübik, kuadratik, karekök fonksiyon vs. ile değiştirilebilir. Veriler arasında doğrusal olmayan bir ilişki söz konusu ise polinom regresyon da önemli yöntemlerden bir tanesidir. Genellikle, bağımsız değişkenlerin n . dereceden şeklindedir.

Bu çalışmada, hiper-parametre seçimleri için hem doğrusal hem de doğrusal olmayan yaklaşım kullanılarak log-loss değerinin hiper-parametreler ile açıklanabilirliği değerlendirilmiştir.

5. MİMARİ SEÇİMİ İÇİN ALADAĞ (2019) TARAFINDAN ÖNERİLEN MELEZ YAKLAŞIM

YSA'nın en büyük problemlerinden bir tanesi mimari seçim problemidir. Literatürde, mimari seçim problemleri için deneme yanılma yöntemlerinden sezgisel algoritmalara kadar birçok farklı yöntem önerilmiştir. Fakat, mimari seçim problemleri için deneme yanılma yönteminin kullanımı daha çok tercih edildiği görülmektedir. YSA'nın birçok bileşeni mevcuttur. Bu bileşenlerin her bir seçeneklerini deneme yanılma yöntemiyle denemek zaman açısından büyük bir kayıp olacaktır. Aladağ, 2019 yılında yayınladığı "Architecture Selection in Neural

Networks by Statistical and Machine Learning” adlı makalesinde YSA’nın en büyük sorunlarından biri olan mimari seçim problemi için istatistiksel öğrenme ve makine öğrenmesinin birlikte kullanılması ile melez bir yaklaşım önermiştir. Bu yaklaşım tabakalarda bulunan nöron sayılarının seçiminde sistematik ve istatistiksel bir yaklaşım sağlamaktır. Aladağ’ın önerdiği bu yaklaşım, istatistiksel ve makine öğrenmesinin birlikte kullanımıyla modelde yüksek performansla ulaşmak amaçlanmaktadır. Bu yeni bakış açısı ile istatistiksel hipotez testlerini uygulamak ve istatistiksel olarak değerlendirmek mümkün olmaktadır. En iyi mimari seçimi istatistiksel olarak tanımlanacaktır. Açıklanan değişken olan model performansı ile açıklayıcı değişkenler olan tabakalardaki nöron sayıları arasındaki korelasyonun tanımlanabileceğini göstermiştir. Regresyon analizinin uygulanması ve yorumlanması kolaydır. Ayrıca, YSA en baştan çalıştırmaya gerek olmadan istenilen mimarinin performans değeri için yorumlanmasını mümkün kılmaktadır. Makalede, önerilen bu yaklaşım üç gerçek veri seti üzerinde çalıştırılmış ve sonuçlar elde edilmiştir. Sonuçlara göre önerilen yaklaşım ile daha iyi sonuçlar elde edildiği gözlemlenmiştir. Deneme yanılma yöntemiyle elde edilen sonuçlar sistematik ve güvenilir değilken, önerilen bu yaklaşım ile güvenilir sonuçlara ulaşılmaktadır. Aladağ’ın önerdiği yöntemin adımları aşağıda verilmektedir:

Adım1: Çalıştırılacak mimari sayısı belirlenir.

Regresyon analizinin anlamlı sonuçlar vermesi için en az 50 mimari çalıştırılması önerilmektedir. Aladağ bir gizli tabakalı YSA çalıştırmıştır. Girdi tabaka ve gizli tabakadaki nöron sayıları 1 ve 12 arasında değiştirilmiştir. Böylece 144 farklı mimari oluşturulmuştur.

Adım2: Seçilen mimari için başlangıç ağırlık değerleri rastgele oluşturulmuştur.

Öğrenme algoritmasında sağlanan sonuçlar ağırlıkların başlangıç değerlerine bağlı olduğu bilinmektedir. Başlangıç ağırlık değerleri değişirse, sağlanan sonuçlarda değişecektir.

Adım3: Önceki adımda belirlenen mimarinin performansları değerlendirilir. Her bir mimari için performans ölçütü vardır. Performans ölçütü olarak HKOK değeri alınmıştır. Böylece 144 farklı mimari için, 144 farklı HKOK performans değeri elde edilmiştir.

Adım4: Doğrusal regresyon analizi oluşturulur.

Açıklanan değişken HKOK performans değeri, açıklayıcı değişkenler ise girdi ve gizli tabakalardaki nöron sayıları olarak alınmıştır.

$$y_{\text{performans}} = \beta_0 + \beta_1 x_{\text{girdi}} + \beta_2 x_{\text{gizli}}$$

Adım5: Önceki adımda oluşturulan regresyon modelinin istatistiksel olarak anlamlılığı test edilir.

Oluşturulan modelin istatistiksel anlamlılık testi için F-test istatistiği kullanılır. Bu model HKOK değeri ile girdi ve gizli tabakalardaki nöron sayıları arasındaki ilişkiyi açıklar. Eğer model istatistiksel olarak anlamlı değilse Adım2'ye geri dönülür. Aksi takdirde diğer adıma geçilir.

Adım6: Regresyon modelinin katsayılarının anlamlılığı kontrol edilir.

Oluşturulan regresyon modelinin istatistiksel olarak anlamlı olabilmesi için en az bir katsayının istatistiksel olarak anlamlı olması gerekir. Tüm katsayılar, t-test istatistiği kullanılarak test edilir. Örnek olarak, β_1 istatistiksel olarak anlamlı ise, girdi tabakasındaki nöron sayısı HKOK değerini etkilemektedir, denir. Sonuç olarak bu algoritma; bir regresyon modeli ve ilişkili test istatistiklerini vermektedir. Regresyon modelinde sağlanan bilgiye göre, uygun mimari kolay bir şekilde tanımlanır.

Yapılan çalışmada, Aladağ'ın 2019 yılında önermiş olduğu yaklaşım Python üzerinde DSA'da hiper-parametre seçimleri için genişletilmiştir. İki gizli tabakaya sahip DSA için, gizli tabakalarda bulunan nöron sayıları, eğitim setinde tüm örneklerin kaç kez eğitim setinden geçtiğini gösteren eğitim tur sayısı değeri ve eğitimde alınan örnek sayısını gösteren parti boyutu (batch_size) değeri, öğrenme oranı ve momentum ile DSA'nın hiper-parametrelerinin seçimi regresyon analizi kullanılarak istatistiksel bir yaklaşımla belirlenmiştir. Birinci gizli tabakada bulunan nöron sayıları, ikinci gizli tabakada bulunan nöron sayıları, eğitim tur sayısı değeri, parti boyutu, öğrenme oranı ve momentum açıklayıcı değişkenler olmak üzere, DSA'nın performans değeri ise HKOK açıklanan değişken alınarak regresyon modeli oluşturulmuştur. Elde edilen sonuçlar istatistiksel olarak değerlendirilmiş ve yorumlanmıştır.

6. UYGULAMA

Çalışmanın uygulama bölümünde kullanılan gerçek hayat verisi www.kaggle.com üzerinden alınan diyabet (diabetes.csv) verisidir. Kullanılan veri seti Ulusal Diyabet ve Sindirim Böbrek Hastalıkları Enstitüsü'nden alınmıştır. Bu veri setinde amaç, belirli tanısal ölçümlere dayalı olarak bir hastada diyabet olup olmadığının tahmin edilmesidir. Veri setindeki tüm hastalar Pima Kızılderili mirasına sahip 21 yaşındaki kadınlardır. 768 gözlem, 9 değişkenden oluşmaktadır.

Pregnancies: Hamile olma sayısı

Glucose: Plazma glikoz konsantrasyonu, bir oral glikoz tolerans testinde 2 saat

DiastolicBloodPressure(mm/Hg): Diyastolik kan basıncı

SkinThickness (mm): Triceps Deri kıvrım kalınlığı

İnsülin: Saatlik serum insülin miktarı

BMI: Vücut Kitle İndeksi (ağırlık/boy)

DiabetesPedigree: Diyabet Soyağacı İşlevi

Age: Yaş

Outcome: Çıktı sonucu (0 ve 1 olmak üzere iki sınıftan oluşmaktadır. 268 tanesi 1 500 tanesi 0 sınıfında bulunmaktadır.)

Aladağ (2019) Tarafından Önerilen Melez Yaklaşımın Diyabet Verisine Uygulanması

Yapılan çalışmada diyabet verisi DSA yaklaşımı ile analiz edilmiştir. En iyi DSA modelinin belirlenmesinde, Aladağ (2019) tarafından önerilen melez yaklaşım kullanılmıştır. Diyabet veri setinin belirtilen şekilde çözümlenmesinde, Python programlama dilinden yararlanılmıştır. DSA yöntemiyle modellemede Aladağ'ın 6 adımda izlediği yol, bu çalışma için 10 adımda aşağıda özetlenmiştir. Hiper-parametre seçimleri önerilen yaklaşımda Aladağ'dan farklı olarak hem doğrusal regresyon analizi hem de doğrusal olmayan regresyon analizi kullanılmıştır. Doğrusal olmayan regresyon analizinde yarı logaritmik dönüşümlerden Doğrusal-Logaritmik Dönüşümü ile doğrusallaştırılarak uygulanmıştır.

Adım1: Gerekli kütüphaneler ve veri yüklenir.

- Python'da gerekli kütüphaneler yüklenir ve veri okutulur.

Adım2: Keşifsel veri analizi (Exploratory Data Analysis) yapılır.

Tanımlayıcı İstatistiklerin İncelenmesi

Index	regnancie	Glucose	oodPressu	inThickne	Insulin	BMI	ssPedigreeF	Age
count	768	768	768	768	768	768	768	768
mean	3.84505	120.895	69.1055	20.5365	79.7995	31.9926	0.471876	33.2409
std	3.36958	31.9726	19.3558	15.9522	115.244	7.88416	0.331329	11.7602
min	0	0	0	0	0	0	0.078	21
25%	1	99	62	0	0	27.3	0.24375	24
50%	3	117	72	23	30.5	32	0.3725	29
75%	6	140.25	80	32	127.25	36.6	0.62625	41
max	17	199	122	99	846	67.1	2.42	81

Şekil1. 20. Tanımlayıcı İstatistikler

Ortalama, çeyreklik değerleri, standart hataları, minimum-maksimum değerleri ile verinin temel istatistikleri incelenmiştir. Şekil 1.20’de tanımlayıcı istatistikler verilmektedir.

Kayıp Gözlemlerin incelenmesi

“df.info()” ile kayıp değer analizi yapılmıştır. Ayrıca bu fonksiyon, değişkenlerin veri tipini de göstermektedir. İnteger; tamsayı, float; kesirli sayı, string; karakter olarak veri tipini göstermektedir. Çizelge 1.2’de kayıp gözlemler gösterilmektedir.

Çizelge1. 2. Kayıp Gözlemlerin İncelenmesi

	Boş Olmayan Sayısı (Non-Null Counts)	Veri Tipi (Dtype)
Pregnancies	768(Boş-olmayan)	int(64)
Glucose	768(Boş-olmayan)	int(64)
BloodPressure	768(Boş-olmayan)	int(64)
SkinThickness	768(Boş-olmayan)	int(64)
Insulin	768(Boş-olmayan)	int(64)
BMI	768(Boş-olmayan)	float(64)
DiabetesPedigreeFunction	768(Boş-olmayan)	float(64)
Age	768(Boş-olmayan)	int(64)
Outcome	768(Boş-olmayan)	int(64)

Tekrarlayan Verilerin incelenmesi

Tekrarlayan deęer (dublicate) olup olmadıęı kontrol edilir. Eęer varsa bu tekrarlayan deęerler ıkartılır. ünkü tekrarlayan deęerler analizi yanıtacaktır. Bu analizde tekrarlayan deęer bulunmamaktadır.

Korelasyon Analizi

Index	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	tesPedigreeFu	Age	Outcome
Pregnancies	1	0.129459	0.141282	-0.0816718	-0.0735346	0.0176831	-0.0335227	0.544341	0.221898
Glucose	0.129459	1	0.15259	0.0573279	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.15259	1	0.207371	0.0889334	0.281805	0.0412649	0.239528	0.0650684
SkinThickness	-0.0816718	0.0573279	0.207371	1	0.436783	0.392573	0.183928	-0.11397	0.0747522
Insulin	-0.0735346	0.331357	0.0889334	0.436783	1	0.197859	0.185071	-0.042163	0.130548
BMI	0.0176831	0.221071	0.281805	0.392573	0.197859	1	0.140647	0.0362419	0.292695
DiabetesPedigreeFunction	-0.0335227	0.137337	0.0412649	0.183928	0.185071	0.140647	1	0.0335613	0.173844
Age	0.544341	0.263514	0.239528	-0.11397	-0.042163	0.0362419	0.0335613	1	0.238356
Outcome	0.221898	0.466581	0.0650684	0.0747522	0.130548	0.292695	0.173844	0.238356	1

Şekil1. 21. Korelasyon Analizi

Şekil1.21’de deęişkenler arasında korelasyon matrisi gösterilmektedir. Korelasyon matrisinde amaç oklu-baęlantı (Multicollinearity) sorunu incelenmesidir. oklu-baęlantı sorunu, açıklayıcı deęişkenler arasında yüksek ilişki olup olmadıęını incelemektedir. Eęer açıklayıcı deęişkenler arasında yüksek ilişki varsa bu durum analizi yanlış yönlendirmektedir ve yanlış yorumlamaya sebep olmaktadır. oklu-baęlantı sorununa özüm için literatürde birçok yöntem vardır. Yapılan analizde yüksek korelasyon görülmemektedir.

Deęişkenler arasındaki ilişkiler; hamile olma sayısı ile yaş arasında %54’lük ilişki olduęu, BMI’nin diyabet olma durumu arasında yaklaşık olarak %30 ilişki olduęu gibi yorumlar ile açıklanabilmektedir.

Aykırı Deęer Analizi

Aykırı deęer (outlier) analizi yapılır. Yapılan analizde 80 tane aykırı deęer bulunmaktadır. Bu aykırı deęerler veriden çıkartılmıştır. Verimiz 768 gözlemken, aykırı deęer analizi ile 688 gözlem ile analize devam edilmiştir. Aykırı deęer analizinde $\pm 3\sigma$ aralığı dışında kalanlar aykırı gözlem olarak alınmıştır.

Adım3: Eğitim / doğrulama / test kümelerinin belirlenmesi

- Veri %60-%20-%20 sırasıyla eğitim, doğrulama ve test kümesi olarak ayrılmıştır.

Adım4: Performans ölçütü belirleme

- Yapılan bu çalışmada kullanılan performans ölçütü, ikili çapraz doğrulama kayıp deęeri veya log loss deęeri olarak alınmıştır.

Adım5: Model oluşturma

Bu adımda (Aladag, 2019);

- Girdi birim sayısı, gizli tabaka sayısı, gizli tabakalarda nöron sayıları ve çıktı tabakasındaki nöron sayısı,
- Birimlerde kullanılacak aktivasyon fonksiyonu,
- Parti_boyutu (Batch_size) ve eğitim tur sayısı (epochs) sayısı,
- Öğrenme oranı, momentum deęerleri
- Test kümesinde kullanılacak performans ölçütü

seçimleri yapılır ve sinir ağı modeli oluşturulur.

Adım7: Sınıflandırma test edilmesi

Bu son adımda, elde edilen sinir ağı modeli test setinde çalıştırılarak model performans deęerleri elde edilir. Başka bir ifadeyle, oluşturulan bu model, yeni özellikler ile çalıştırıldığında sınıflandırma sonuçları elde edilir.

Adım8: DSA'da elde edilen tahmin deęerleri DSA'ların sahip olduęu mimari yapı ve dięer hiper-parametrelerin seçiminden etkilenecektir. Yapılan bu çalışmada, önerilen yaklaşım (Aladag, 2019) ile iki gizli tabakalı ileri beslemeli derin sinir ağıları kullanıldığında, gizli tabakalarda bulunan nöron sayıları, modelin sahip olduęu parti_boyutu (batch_size) ve eğitim tur sayısı (epochs), öğrenme oranı ve momentum deęerlerinin belirlenmesi problemi olarak geliştirilmiştir. Bu problemi çözmek amacıyla doğrusal ve doğrusal olmayan regresyon analizleri kullanılmıştır. Regresyon analizi istatistiksel olarak yorumlama yapabilmesi ve

özellikler arasında ilişkiyi ortaya koyabilmesi yönüyle hiper-parametre seçimi için kullanılması tercih edilmektedir. Makine öğrenmesi veri temelli bir yaklaşımdır. Veriden öğrenir. Hiper-parametreler sonsuz değerlere sahip olacağından, daha iyi sonuca ulaşabilmek amacıyla DSA modeli sonsuz kez çalıştırılabilir. Böylece, zaman açısından maliyet gerektiren bir durumda olacaktır. İstatistiksel öğrenme yöntemlerinden olan “Regresyon Analizi” yöntemi ile hiper-parametreler arasındaki ilişkilerden ve bu hiper-parametrelerin modelin performansına olan etkisini yorumlanabilir duruma getirmesi sebebiyle tercih edilmiştir.

Bu çalışmada aşağıdaki 6 hiper-parametreler ile çalışılmıştır.

- Parti_boyutu= [64,128]
- Eğitim tur sayısı = [25,50]
- nöron1=np.arange(1,15)
- nöron2=np.arange(1,15)
- Öğrenme Oranı=[0.001,0.1]
- Momentum=[0.8,0.9]

hiper-parametreleri alınır.

$2*2*14*14*2*2=3136$ farklı kombinasyon mevcuttur. Her bir kombinasyon için 3136 farklı performans değerleri elde edilmiştir. Verilen altı hiper-parametre için doğrusal regresyon modeli kurulmuştur. Aşağıda bu doğrusal regresyon modeli verilmiştir.

$$y_{performans} = \beta_0 + \beta_1 * x_{nöron1} + \beta_2 * x_{nöron2} + \beta_3 * x_{parti_boyutu} + \beta_4 x_{eğitim_tur_sayısı} + \beta_5 * x_{öğrenme_oranı} + \beta_6 * x_{momentum}$$

3136 farklı kombinasyon, verilen bu altı parametre için etkileşimler de göz önüne alınarak doğrusal olmayan regresyon modeli kurulmuştur. Burada amaç, birlikte etkileşimlerinin log-loss performans değerine etkisinin incelenmesidir. Etkileşimler modele katıldığı için model doğrusal olmayacaktır. Bunun için yarı-logaritmik dönüşümler uygulanarak model doğrusallaştırılmıştır. Doğrusal-logaritmik dönüşümü uygulanmıştır. Her iki uygulanan regresyon modelleri için sonuçlar Adım9 ve Adım10’da verilmektedir.

Adım9: Önceki adımda elde edilen regresyon modelinin anlamlılığı test edilir.

Elde edilen regresyon modelinin istatistiksel olarak anlamlılığı test edilir. Bunun için F-test istatistiğinden yararlanılmaktadır. Eğer model anlamlı değilse, Adım6’daki model başlangıç ağırlık değerleri yeniden çalıştırılarak devam edilir. Aksi takdirde Adım10’a geçilir. Çizelge1.3

ve Çizelge1.4 doğrusal ve doğrusal olmayan regresyon modelleri için model anlamlılık sonuçları gösterilmektedir.

Çizelge1. 3. OLS Doğrusal Regresyon Model Anlamlılık Sonuçları

R-Kare (R-Squared)	0.849
Düzeltilmiş R-Kare (Adjusted R-Squared)	0.849
F-İstatistik Olasılığı (p)	0.000
Omnibus Olasılığı	0.000
Durbin Watson	1.763

Çizelge1. 4.OLS Doğrusal Olmayan Regresyon Modeli Anlamlılık Sonuçları

R-Kare (R-Squared)	0.865
Düzeltilmiş R-Kare (Adjusted R-Squared)	0.865
F-İstatistik Olasılığı (p)	0.000
Omnibus Olasılığı	0.000
Durbin Watson	1.971

- Model anlamlılık testi için aşağıdaki hipotez kurulur:

H_0 : Model anlamlı değildir.

H_1 : Model anlamlıdır.

Hem doğrusal regresyon hem de doğrusal olmayan regresyon analizinin F-İstatistik Olasılık değeri $p = 0.00 < \alpha = 0.05$ olduğundan H_0 hipotezi reddedilir. Böylece, kurulan regresyon modelleri %95 güven düzeyinde istatistiksel olarak anlamlı olduğu görülmektedir.

- Doğrusal regresyon analizinde, log-loss değeri açıklanan değişkeni, birinci tabakadaki nöron sayısı, ikinci tabakadaki nöron sayısı, eğitim tur sayısı değeri, parti boyutu, öğrenme oranı ve momentum açıklayıcı değişkenleri ile %85 (R-Kare=0.849) oranında açıklandığı görülmektedir. Doğrusal olmayan regresyon analizinde ise verilen bu altı hiper-parametre ve etkileşimleri log-loss performans değerini %87 oranında açıkladığı görülmektedir.

- Otokorelasyon, farklı zamanlarda hata terimleri arasında ilişkinin var olduğunu söylemektedir. Regresyon analizinin en önemli varsayımlarından birisi hatalar arasında otokorelasyonun olmamasıdır. Durbin-Watson test istatistiği otokorelasyonun varlığını göstermektedir. Hata terimlerinin korelasyon halinde olup olmadığını kontrol eder. Durbin Watson test istatistiği 0 ile 4 arasında değer almaktadır. DW=2 olduğunda korelasyon olmadığını göstermektedir. DW=1.763≈2 olması otokorelasyonun olmadığını göstermektedir.

Adım10: Regresyon modelinin katsayılarının anlamlılıkları kontrol edilir.

Regresyon modeli analizinde en az bir katsayı anlamlı ise model de anlamlıdır. Tüm katsayıların anlamlılıkları t-testi kullanılarak test edilir. Önemli katsayılar tanımlanır.

Doğrusal Regresyon Analizi

Çizelge1. 5.OLS Doğrusal Regresyon Analizi Katsayı Anlamlılıkları

	Katsayı	Stand. Hata	t	P> t	0.025	0.975
Sabit	0.9148	0.038	24.082	0.000	0.840	0.989
Nöron1	-0.0062	0.001	-11.610	0.000	-0.007	-0.005
Nöron2	-0.0060	0.001	-11.278	0.000	-0.007	-0.005
Parti_boyutu	0.0003	6.68e-05	4.414	0.000	0.000	0.000
Eğitim Tur Sayısı	-0.0011	0.000	-6.485	0.000	-0.001	-0.001
Öğrenme Oranı	-5.6753	0.043	-131.453	0.000	-5.760	-5.591
Momentum	-0.2261	0.0043	-5.290	0.000	-0.310	-0.142

Çizelge 1.5’de doğrusal regresyon analizi için regresyon modeli katsayıları anlamlılık sonuçları verilmektedir.

Elde edilen regresyon modeli:

$$Y_{performans} = 0.9148 + (-0.0062) * x_{nöron1} + (-0.0060) * x_{nöron2} + (0.0003) * x_{parti_boyutu} + (-0.0011) * x_{eğitim_tur_s} + (-5.6753) * x_{öğrenme_oranı} + (-0.2261) * x_{momentum}$$

Çizelge1.5.’de açıklayıcı değişkenlerin katsayılarının anlamlı olduğu görülmektedir.

$$-0.007 < \beta_{nöron1} < -0.005$$

$$-0.007 < \beta_{nöron2} < -0.005$$

0.00≈0’a çok yakın pozitif sayı β_{parti_boyutu} <math><0.00≈0</math>’a çok yakın pozitif sayı

$$\approx -0.001 < \beta_{\text{eğitim_tur}} < \approx -0.001$$

$$-5.760 < \beta_{\text{öğrenme_oranı}} < -5.591$$

$$-0.310 < \beta_{\text{momentum}} < -0.142$$

$\beta_{\text{nöron1}}$, -0.007 ile -0.005 değerleri arasında olduğundan 0 değeri içermediği görülmektedir. Böylece, $\beta_{\text{nöron1}}$ katsayısı %95 güven düzeyinde istatistiksel olarak anlamlı olduğu görülmektedir. Birinci gizli tabakadaki nöron sayısının 1 birimlik artması, log-loss değerini 0.0062 azaltacaktır. Diğer katsayılar için de aynı şekilde yorum yapılmaktadır. Sonuç olarak, log-loss değerini birinci gizli tabakadaki nöron sayısı, ikinci gizli tabakadaki nöron sayısı, parti boyutu sayısı, eğitim tur sayısı ve öğrenme oranının log-loss performans değerini %95 güven düzeyinde istatistiksel olarak etkilediği gözlemlenmektedir.

Çizelge-1.6’de örneklem dışı gözlemlerin DSA’nın hiper-parametrelerinin doğrusal regresyon ile seçimiyle oluşturulan model için tahmini log-loss değerleri hesaplanmıştır. Ayrıca bu gözlemlerin DSA’daki log-loss değerleri de hesaplanarak karşılaştırma yapılmıştır. Log-loss değerlerinin yakın değerler aldığı gözlemlenmiştir. Bir başka ifadeyle, eğitimde hiç kullanılmamış hiper-parametreler için DSA modellerinin nasıl sonuçlar üreteceği kurulan doğrusal regresyon modelleri ile iyi bir biçimde tahmin edilebilmektedir.

Çizelge1. 6. Örnek Dışı Derin Sinir Ağı-Doğrusal Regresyon Analizi Hiper-Parametre Seçimleri

Nöron1	Nöron2	Parti-Boyutu	Eğitim Tur Sayısı	Öğrenme Oranı	Momentum	Regresyon log-loss	DSA_log-loss
15	15	32	10	0.01	0.6	0.5437	0.5402
15	16	32	10	0.01	0.7	0.5178	0.5154
25	23	32	5	0.01	0.6	0.4519	0.5488
19	16	646	5	0.01	0.6	0.5329	0.6187
18	18	64	25	0.01	0.5	0.5259	0.5026
18	18	128	5	0.001	0.5	0.6159	0.6976
19	21	64	5	0.01	0.7	0.4823	0.5875
17	15	64	15	0.01	0.7	0.5193	0.5605
22	21	64	10	0.01	0.8	0.4437	0.5205
19	15	32	10	0.002	0.9	0.5119	0.4307

Doğrusal Olmayan Regresyon Analizi

Verilen altı hiper-parametrelerin etkileşimlerinin de log-loss performans değerine olan etkisi incelenmiştir. Bu sebeple doğrusal olmayan regresyon analizi kurulmuştur. Doğrusal olmayan regresyon analizi yarı logaritmik model olan doğrusal-logaritmik dönüşüm uygulanarak doğrusallaştırılmış ve en küçük kareler yöntemi ile regresyon modeli kurulmuştur. Burada bağımsız değişkenlerin logaritması alınarak etkileşimlerinin yüksek artış göstermesi sebebiyle

oluşan eğriselliği engellemek amacıyla bağımsız değişkenleri logaritması alınarak doğrusallaştırılmıştır. Bu regresyon analizi sonucu, bağımsız değişkenlerce, bağımlı değişkenin %87 oranında açıklanabildiği görülmektedir. Çizelge 1.7’de doğrusal olmayan regresyon analizi sonucunda katsayıların anlamlılık testi sonuçları verilmektedir. Burada nöron1 değişkeni için $p=0.019 < \alpha=0.05$ olduğundan nöron1 katsayısı %95 güven düzeyinde anlamlıdır denir. Nöron1 katsayısı %1 arttığında log-loss performans değeri 0.0009 birim artacaktır, yorumu yapılmaktadır. Aynı şekilde diğer değişkenler için de yorumlar yapılabilir.

Çizelge1. 7.Doğrusal Olmayan Regresyon Analizi Katsayıların Anlamlılık Sonuçları

	Katsayı Değerleri	Std Hata	T	P> t	[0.025	0.975
sabit	-0.1227	0.035	-3.470	0.001	-0.192	-0.053
Nöron1	0.0009	0.000	2.349	0.019	0.000	0.002
Nöron2	0.0020	0.001	4.022	0.000	0.001	0.003
Parti Boyutu	0.0054	0.000	11.388	0.000	0.004	0.006
Eğitim Tur Sayısı	0.0023	0.001	3.630	0.000	0.001	0.004
Öğrenme Oranı	-0.0074	0.000	-45.972	0.000	-0.008	-0.007
Momentum	-0.0101	0.002	-4.453	0.000	-0.015	-0.006
Nöron1_nöron2	0.0029	0.001	3.520	0.000	0.001	0.005
Nöron1_partiboyutu	0.0063	0.001	8.985	0.000	0.005	0.008
Nöron1_eğitimgur	0.0032	0.001	3.404	0.001	0.001	0.005
Nöron1_öğrenmeoranı	-0.0065	0.000	-19.926	0.000	-0.007	-0.006
Nöron1_momentum	-0.0092	0.002	-4.715	0.000	-0.013	-0.005
nöron2_parti_boyutu	0.0074	0.001	8.614	0.000	0.006	0.009
nöron2_eğitimgur	0.0044	0.001	4.209	0.000	0.002	0.006
nöron2_öğrenmeoranı	-0.0054	0.000	-12.490	0.000	-0.006	-0.005
nöron2_momentum	-0.0081	0.002	-4.451	0.000	-0.012	-0.005
partiboyutu_eğitimgur	0.0077	0.001	8.714	0.000	0.006	0.009
partiboyutu_öğrenmeoranı	-0.0021	0.000	-4.969	0.000	-0.003	-0.001
partiboyutu_momentum	-0.0048	0.002	-2.359	0.018	-0.009	-0.001
eğitimgur_öğrenme oranı	-0.0051	0.001	-9.664	0.000	-0.006	-0.004
eğitimgur_momentum	-0.0078	0.002	-4.343	0.000	-0.011	-0.004
nöron1_nöron2_partiboyutu	0.0083	0.001	7.299	0.000	0.006	0.011
nöron1_nöron2_eğitimgur	0.0053	0.001	3.864	0.000	0.003	0.008
nöron1_nöron2_öğrenmeoranı	-0.0045	0.001	-6.033	0.000	-0.006	-0.003
nöron1_nöron2_momentum	-0.0072	0.002	-4.790	0.000	-0.010	-0.004
nöron1_parti_boyutu_eğitimgur	0.0086	0.001	7.301	0.000	0.006	0.011
nöron1_partiboyutu_öğrenmeoranı	-0.0012	0.001	-1.914	0.056	-0.002	2.85e-05
nöron1_partiboyutu_momentum	-0.0039	0.002	-2.288	0.022	-0.007	-0.001
nöron1_eğitimgur_partiboyutu	0.0097	0.001	7.457	0.000	0.007	0.012
nöron1_eğitimgur_öğrenmeoranı	-0.0031	0.001	-3.348	0.001	-0.005	-0.001

nöron1_eğitimgur_momentum	-0.0058	0.001	-4.339	0.000	-0.008	-	0.003
nöron2_partiboyutu_eğitimgur	0.0097	0.001	7.457	0.000	0.007	0.012	
nöron2_partiboyutu_öğrenmeoranı	-4.376e-05	0.001	-	0.955	-0.002	0.001	
Nöron2_partiboyutu_momentum	-0.0027	0.002	-1.740	0.082	-0.006	0.000	
Nöron2_eğitimgur_öğrenmeoranı	-0.0031	0.001	-3.348	0.001	-0.005	-	0.001
Nöronn2_eğitimgur_momentum	-0.0058	0.001	-4.339	0.000	-0.008	-	0.003
Nöron2_öğrenmeoranı_momentum	-0.0155	0.002	-8.022	0.000	-0.019	-	0.012
Partiboyutu_eğitimgur_öğrenmeoranı	0.0003	0.001	0.350	0.726	-0.001	0.002	
Partiboyutu_eğitimgur_momentum	-0.0024	0.002	-1.614	0.107	-0.005	0.001	
Eğitimgur_öğrenmeoranı_momentum	-0.0152	0.002	-8.037	0.000	-0.019	-	0.012
Nöron1_nöron2_partiboyutu_eğitimgur	0.0106	0.002	6.593	0.000	0.007	0.014	
Nöron1_nöron2_partiboyutu_öğrenmeoranı	0.0009	0.001	0.822	0.411	-0.001	0.003	
Nöron1_nöron2_partiboyutu_momentum	-0.0018	0.001	-1.482	0.139	-0.004	0.001	
Nöron1_nöron2_eğitimgur_öğrenmeoranı	-0.0022	0.001	-1.745	0.081	-0.005	0.000	
Nöron1_nöron2_eğitimgur_momentum	-0.0049	0.001	-4.736	0.000	-0.007	-	0.003
Nöron1_nöron2_öğrenmeoranı_momentum	-0.0146	0.002	-9.022	0.000	-0.018	-	0.011
Nöron1_partiboyutu_eğitimgur_öğrenmeoranı	0.0012	0.001	1.104	0.270	-0.001	0.003	
Nöron1_partiboyutu_eğitimgur_momentum	-0.0015	0.001	-1.301	0.193	-0.004	0.001	
Nöron2_partiboyutu_eğitimgur_öğrenmeoranı	0.0023	0.001	1.934	0.053	-3.1e-05	0.005	
Nöron2_partiboyutu_eğitimgur_momentum	-0.0004	0.001	-0.397	0.691	-0.002	0.002	
Partiboyutu_eğitimgur_öğrenmeoranı_momentum	-0.0099	0.002	-6.156	0.000	-0.013	-	0.007
Nöron1_nöron2_partiboyutu_eğitimgur_öğrenmeoranı	0.0032	0.001	2.135	0.033	0.000	0.006	
Nöron1_nöron2_partiboyutu_eğitimgur_momentum	0.0005	0.001	0.692	0.489	-0.001	0.002	
Nöron1_nöron2_eğitimgur_öğrenmeoranı_momentum	-0.0123	0.001	-	0.000	-0.015	-	0.010
Nöron1_partiboyutu_eğitimgur_öğrenmeoranı_momentum	-0.0090	0.001	-7.046	0.000	-0.011	-	0.006
Nöron2_partiboyutu_eğitimgur_öğrenmeoranı_momentum	-0.0078	0.001	-6.919	0.000	-0.010	-	0.006
Nöron1_nöron2_partiboyutu_eğitimgur_öğrenmeoranı_momentum	-0.0069	0.001	-8.782	0.000	-0.008	-	0.005

Çizelge1.7’de verilen doğrusal olmayan regresyon analizi sonucunda;

Nöron1_partiboyutu_öğrenmeoranı ,

Nöron2_partiboyutu_öğrenmeoranı,

Nöron2_partiboyutu_momentum,

partiboyutu_eğitimgur_öğrenmeoranı,

partiboyutu_eğitimgur_momentum ,

Nöron1_nöron2_partiboyutu_öğrenmeoranı,

Nöron1_nöron2_partiboyutu_momentum,

Nöron1_nöron2_eğitimgur_öğrenmeoranı,

Nöron1_partiboyutu_eğitimgur_öğrenmeoranı,

Nöron1_partiboyutu_eğitimgur_momentum,

Nöron2_partiboyutu_eğitimgur_momentum,

Nöron1_nöron2_partiboyutu_eğitimgur_momentum, etkileşimlerinin $p > \alpha=0.05$ olması nedeniyle kurulan sıfır hipotezi reddedilemez ve bu değişkenlerin log-loss performans değerine %95 güven düzeyinde istatistiksel olarak anlamlı bir etkisi olmadığı söylenmektedir.

Çizelge-1.8'de örneklem dışı gözlemlerin DSA'nın hiper-parametrelerinin doğrusal olmayan regresyon analizi ile seçilen model için tahmini log-loss değerleri hesaplanmıştır. Ayrıca bu gözlemlerin DSA'daki log-loss değerleri hesaplanarak karşılaştırma yapılmıştır. Log-loss performans değerlerinin yakın değerler aldığı gözlemlenmiştir. Bir başka ifadeyle, eğitimde hiç kullanılmamış hiper-parametreler için DSA modellerinin nasıl sonuçlar üreteceği kurulan doğrusal olmayan regresyon modelleri ile iyi bir biçimde tahmin edilebilmektedir.

Çizelge1. 8. Örnek-Dışı Derin Sinir Ağı-Doğrusal Olmayan Regresyon Analizi Hiper-Parametre Seçimleri

Nöron1	Nöron2	Parti-Boyutu	Eğitim Tur Sayısı	Öğrenme Oranı	Momentum	Regresyon log-loss	DSA_log-loss
15	15	64	15	0.001	0.5	0.6574	0.6802
21	21	64	15	0.001	0.7	0.5744	0.6778
16	15	64	25	0.001	0.6	0.6071	0.6753
18	18	32	25	0.001	0.7	0.5501	0.6144
19	17	32	35	0.001	0.7	0.5393	0.5729
15	15	32	35	0.001	0.6	0.5788	0.6460
16	15	128	50	0.001	0.5	0.6328	0.6694
18	18	128	50	0.01	0.5	0.3346	0.2799
18	18	64	35	0.002	0.6	0.4972	0.5463
16	18	128	10	0.002	0.5	0.5937	0.6768

7. SONUÇ

DSA'da en önemli problemlerden birisi hiper-parametre seçimleridir. DSA ile veri analizinde kullanılacak birçok kombinasyon vardır. Tüm bu olası kombinasyonları denemek mümkün değildir. Bu sebeple literatürde geliştirilen bazı yöntemler bulunmaktadır.

Aladağ 2019 yılındaki çalışmasında, YSA yaklaşımının en önemli noktalarından biri olan mimari seçim problemi için melez bir yaklaşım önermiştir. Bu yaklaşımda tabakalarda bulunan nöron sayılarının seçiminde sistematik ve istatistiksel bir yaklaşım sağlayan regresyon modeli kullanımı önerilmiştir. Böylece tabakalardaki nöron sayılarının seçimi istatistiksel hipotez

testleri kullanılarak güvenilir ve açıklanabilir şekilde gerçekleştirilebilmektedir. Gerçekleştirilen tez çalışmasında ise Aladağ'ın 2019'da önermiş olduğu çalışma baz alınarak DSA'da hiper-parametre seçimlerinde kullanılmıştır. Verilen bu hiper-parametrelerin seçimleri herhangi bir yönteme dayanmamaktadır. Zaman ve elde edilen sonuçlar değerlendirilerek bu hiper-parametrelerin seçilmesinde karar kılınmıştır. Ayrıca Aladağ'ın kurmuş olduğu regresyon modelinin yanı sıra, hiper-parametrelerin etkileşimlerinin de eklenmesi ile doğrusal olmayan regresyon modeli de uygulanmıştır. Bu çalışma kaggle.com üzerinden alınan diyabet hastalığı (diabetes.csv) gerçek veri seti üzerinde uygulanmıştır. Kurulan doğrusal regresyon modeli sonucunda altı hiper-parametrenin log-loss performans değerini %85 oranında, doğrusal olmayan regresyon modelinin ise log-loss performans değerini %87 oranında açıkladığı görülmüştür. Alınan örneklem dışı gözlemlerin hem DSA performans değeri hem de kurulan regresyon modellerinde tahmin sonuçları karşılaştırıldığında, kurulan regresyon modellerinin DSA modelleri performanslarını çok iyi tahmin ettiği görülmüştür.

Sonuç olarak DSA'nın hiper-parametrelerinin Aladağ (2019) yöntemi kullanılarak seçimi ile Diyabet verisinin başarılı bir biçimde çözümlendiği görülmüştür. Ayrıca doğrusal olmayan regresyon modelinin de kullanılmasıyla hiper-parametrelerin etkileşimlerinin de DSA yaklaşımı performansı üzerindeki etkisi yapılan tez çalışmasında ortaya konmuştur. Hiper-parametre seçimi ile performans arasındaki ilişki yapısı hem doğrusal hem doğrusal olmayan yapıda incelenmiş ve istatistiksel olarak anlamlı modellere ulaşılmıştır. Buna göre, birçok hiper-parametrenin kombinasyonunun nasıl bir DSA performansı üreteceği çözümlene yapılmadan, istatistiksel olarak anlamlı biçimde tahmin edilebilmektedir. Belirtilen bu durum özellikle büyük veri söz konusu olduğunda büyük avantaj sağlayacaktır.

8. KAYNAKÇA

- Elmas Ç. (2018, Haziran) Yapay Zeka Uygulamaları (4.Baskı, Ankara), Seçkin Yayıncılık, ISBN:978-975-02-4897-9 .
- Yılmaz A. (2020, Mart) Yapay Zeka (7. Baskı, İstanbul), İnkılâp Kitabevi Yayın.San.Tic.A.Ş. ISBN:978-605-9118-80-4, syf:1-89.
- Yılmaz A. , Kaya U. (2020, Nisan) Derin Öğrenme, (2.Baskı, İstanbul) İnkılâp Kitabevi Yayın San Tic. A.Ş., ISBN:978-605-2118-39-9 syf:1-126.
- Arslan İ. (2019, Mart) Python ile Veri Bilimi (1.Baskı, İstanbul), Pusula 20 Teknoloji ve Yayıncılık A.Ş., ISBN:978-605-2359-64-8, syf: 287-381.
- Burkov A, The Hundred-Page Machine Learning Book” kitabından çeviri, Çeviren: Okatan A., Karatekin T., Okatan K., (2021, Şubat) ,100 Sayfada Makine Öğrenmesi Kitabı (1.Baskı, İstanbul), Papatya Yayıncılık Eğitim, ISBN:978-975-6797-94-5, syf:1-69.
- Montgomery D.C., Peck E., Vining G.G., Introduction to Linear Regression Analysis, 5. Basımdan çeviri Çeviren: Erar M.A. Doğrusal Regresyon Analizine Giriş (2013, Eylül), Nobel Akademik Yayıncılık Eğitim Danışmanlık Tic. LTD. ŞTİ., ISBN:978-605-133-618-3, syf:389-396
- Alpaydın E. (2020, Aralık), Yapay Öğrenme : Yeni Yapay Zeka, (1.Baskı, İstanbul), Tellekt, ISBN:978-625-7118-09-5, syf:11-131.
- Gürsakal N., (2021), Yapay Zeka Fareli Köyün Kavalcısı, (1. Baskı, Bursa), Dora Basım-Yayın Dağıtım Ltd.Şti., ISBN:978-605-247-272-9, syf:125-157.
- Aladag, C.H., Editor, (2017) Advances in Time Series Forecasting, Volume 2, Bentham Science Publishers Ltd., eISBN: 978-1-68108-528-9, ISBN: 978-1-68108-529-6.
- Aladag, C.H., Egrioglu, E., Yolcu, U. (2014), Robust multilayer neural network based on median neuron model, Neural Computing & Applications, 24, 945-956.
- Aladag, C.H. (2011a), A New Candidate List Strategy for Architecture Selection in Artificial Neural Networks, New Developments in Artificial Neural Networks Research, (Editor: Robert W. Nelson), pp. 139-150, Nova Publisher. ISBN: 978-1-61324-286-5.
- Aladag, C.H. (2011), A Hybrid Intelligent Technique Combines Neural Networks and Tabu Search Methods for Forecasting, Computer Search Algorithms, (Editor: Elisabeth C. Salander), pp. 89-100, Nova Publisher. ISBN: 978-1-61209-043-6.
- Aladag, C.H., Kayabasi, A., Gokceoglu, C. (2013), Estimation of pressuremeter modulus and limit pressure of clayey soils by various artificial neural network models, Neural Computing & Applications, Volume 23, Issue 2, 333-339.
- Aladag, C.H. (2011c), A new architecture selection method based on tabu search for artificial neural networks, Expert Systems with Applications, 38, 3287–3293.

- Aladag, C.H., Yolcu, U., Egrioglu, E. (2013), A new multiplicative seasonal neural network model based on particle swarm optimization, Neural Processing Letters, Volume 37, Issue 3, 251-262.
- Yolcu, U., Bas, E., Egrioglu, E., Aladag, C.H. (2015), A new multilayer feedforward network model based on trimmed mean neuron model, Neural Network World, Volume 25 (2015), pp 587-602.
- Egrioglu E., Aladag, C.H., Yolcu, U., Bas, E. (2015) Recurrent Multiplicative Neuron Model Artificial Neural Network for Non-Linear Time Series Forecasting, Neural Processing Letters, 41 (2), 249-258.
- Unal, B., Aladag, C.H., Stock Exchange Prediction via Long Short-Term Memory Networks, International Conference on Data Science, Machine Learning and Statistics (DMS 2019) Proceeding Book, pp. 246–248, Van Yüzüncü Yıl University, Van, Turkey, June 26-29, 2019.
- Aladag,C.H. (2019), Architecture Selection in Neural Networks by Statistical and Machine Learning, Oriental Journal of Computer Science and Technology 12 (3), DOI : <http://dx.doi.org/10.13005/ojst12.03.02>
- Aladağ C.H., Yapay Sinir Ağlarının Mimari Seçimi için Tabu Arama Algoritması, 92s., Hacettepe Üniversitesi, Doktora Tezi, (2009), <https://www.ulusaltezmerkezi.net/yapay-sinir-aglarinin-mimari-secimi-icin-tabu-arama-algoritmasi/>
- Bodyanskiy Y., Popov S., (2006) Neural network approach to forecasting of quasiperiodic financial time series, European Journal of Operational Research 175 (2006) 1357–1366
- Gèron A., (2019), Second Edition, Hands-On Machine Learning with Scikit-Learn, Keras&Tensorflow Concepts, Tools, and Techniques to Build Intelligent Systems
- Kızrak A. (2019, Şubat 4). *Derin öğrenme için aktivasyon fonksiyonlarının karşılaştırılması*. Medium. <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd> (Son Ulaşma Tarihi: 05.06.2021)
- Doğan. Ö. (2020, Ağustos 24), *Derin öğrenme nedir? Yapay sinir ağları ne işe yarar?* Teknoloji.org https://teknoloji.org/derin-ogrenme-nedir-yapay-sinir-aglari-ne-ise-yarar/#Yapay_Sinir_Aglari_Cesitleri (Son Ulaşma Tarihi: 05.06.2021)
- Yapay sinir ağı eğitimi (2018 Temmuz 5) – Çok katmanlı perceptron (Multi layer perceptron)*. Devhunter.<https://devhunteryz.wordpress.com/2018/07/05/yapay-sinir-agi-egitimi-cok-katmanli-perceptronmulti-layer-perceptron/> (Son Ulaşma Tarihi: 05.06.2021)
- Gupta N. (2020, Mayıs 17). *Difference between model parameters vs hyperparameters*. GeeksforGeeks. <https://www.geeksforgeeks.org/difference-between-model-parameters-vs-hyperparameters/> (Son Ulaşma Tarihi: 05.06.2021)

- Uslu M. (2016, Mayıs 3). *Yapay sinir ağları (YSA) nedir- Bölüm1?* Veri Bilimci. <http://veribilimci.org/yapay-sinir-aglari-ysa-nedir-bolum-1/> (Son Ulaşma Tarihi: 05.06.2021)
- Dayvdova O. (2017, Eylül 26). *7 types of artificial neural networks for natural language processing.* Medium. <https://medium.com/@datamonsters/artificial-neural-networks-for-natural-language-processing-part-1-64ca9ebfa3b2> (Son Ulaşma Tarihi: 05.06.2021)
- Shukla P., Iriundo R. (2021, Mart 14). *Main types of neural networks and its applications-Tutorial.* <https://medium.com/towards-artificial-intelligence/main-types-of-neural-networks-and-its-applications-tutorial-734480d7ec8e> (Son Ulaşma Tarihi: 05.06.2021)
- Ulgen E.K. (2017, Ekim 24). *Makine öğrenimi Bölüm3| (Yapay sinir ağları).* Medium. <https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-3-4b160df1f4c8> (Son Ulaşma Tarihi: 05.06.2021)
- Ergüder H. (2018, Haziran 21). *Recurrent neural network nedir?* Medium. <https://medium.com/@hamzaerguder/recurrent-neural-network-nedir-bdd3d0839120> (Son Ulaşma Tarihi: 05.06.2021)
- Çelikten A. (2020, Nisan3). *RNN (Recurrent Neural Network) nedir ve nasıl çalışır?* Medium. <https://medium.com/@azercelikten/rnn-recurrent-neural-network-nedir-ve-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-d1246b1a61fb> (Son Ulaşma Tarihi: 05.06.2021)
- Evrişimli sinir ağları. (2021, Mart 14). Wikipedi. https://tr.wikipedia.org/wiki/Evri%C5%9Fimsel_Sinir_A%C4%9Flar%C4%B1 (Son Ulaşma Tarihi: 05.06.2021)
- Şimşek N.Y. (2019, Şubat 24). *Derin öğrenme (deep learning) nedir ve nasıl çalışır?* Medium. <https://nyilmazsimsek.medium.com/derin-%C3%B6%C4%9Frenme-deep-learning-nedir-ve-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-2d7f5850782> (Son Ulaşma Tarihi: 05.06.2021)
- Cerebrov (2018, Şubat 13). *Yeni başlayanlar için makine öğrenmesi algoritmaları.* Medium. <https://medium.com/t%C3%BCrkiye/yeni-ba%C5%9Flayanlar-i%C3%A7in-makine-%C3%B6%C4%9Frenmesi-algoritmalar%C4%B1-ae22f794af2f> (Son Ulaşma Tarihi: 05.06.2021)
- Jordan J. (2017, Kasım 2). *Hyperparameter tuning for machine learning models.* <https://www.jeremyjordan.me/hyperparameter-tuning/> (Son Ulaşma Tarihi: 05.06.2021)
- David D. (2020, Ekim 12). *Hyperparameter optimization techniques to improve your machine learning model's performance.* Freecodecamp. <https://www.freecodecamp.org/news/hyperparameter-optimization-techniques-machine-learning/> (Son Ulaşma Tarihi: 05.06.2021)
- Brownlee J. (2019, Ağustos 6). *Machine Learning Mastery. A gentle introduction to dropout for regularizing deep neural networks.* <https://machinelearningmastery.com/dropout->

for-regularizing-deep-neural-networks/#:~:text=The%20default%20interpretation%20of%20the.rate%2C%20such%20as%20of%200.8. (Son Ulaşma Tarihi: 05.06.2021)

Güzel Y. (2018, Mart 23). *Yapay zeka ders notları 04| Yapay sinir ağları yapısı*. Medium. <https://medium.com/@yasinguzel/yapay-zeka-ders-notlar%C4%B1-04-yapay-sinir-a%C4%9Flar%C4%B1n%C4%B1n-yap%C4%B1s%C4%B1-6b9bfc3004bf> (Son Ulaşma Tarihi: 05.06.2021)

QA Stack. <https://qastack.info.tr/stats/222584/difference-between-feedback-rnn-and-lstm-gru> (Son Ulaşma Tarihi: 05.06.2021)

<https://stackoverflow.com/questions/4752626/epoch-vs-iteration-when-training-neural-networks>(Son Ulaşma Tarihi: 05.06.2021)

Akademi 4.0 Ekibi (2019, Aralık 6). *Makine öğrenimi ve derin öğrenme tarihi*. <https://akademi40.org/makine-ogrenimi-tarihi> (Son Ulaşma Tarihi: 05.06.2021)

Sutskever vd. (2013).Keras. <https://keras.io/api/optimizers/sgd/> (Son Ulaşma Tarihi: 05.06.2021)

Çarkacı N. (2018, Ocak 22). *Derin öğrenme uygulamalarında en sık kullanılan hiper-parametreler*. Medium. <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4> (Son Ulaşma Tarihi: 05.06.2021)

Koptur M. (2017, Mayıs 31). *Düzenleştirme (Regularization)*. *Makine Öğrenimi*. <https://makineogrenimi.wordpress.com/2017/05/31/duzenlilestirme-regularization/> (Son Ulaşma Tarihi: 05.06.2021)

Keskin M.V. (2018, Eylül 30) *Makine öğrenmesi ile istatistiksel öğrenme arasında ne fark var?* VBO. <https://www.veribilimiokulu.com/makine-ogrenmesi-ile-istatistiksel-ogrenme-arasinda-ne-fark-var/> (Son Ulaşma Tarihi: 05.06.2021)

Güzel Y. (2018, Mart 23). *Yapay zeka ders notları 04| Yapay sinir ağlarının yapısı*. Medium. <https://medium.com/@yasinguzel/yapay-zeka-ders-notlar%C4%B1-04-yapay-sinir-a%C4%9Flar%C4%B1n%C4%B1n-yap%C4%B1s%C4%B1-6b9bfc3004bf> (Son Ulaşma Tarihi: 05.06.2021)

Çarkacı N., (2018, Şubat 12), *Derin Öğrenme Uygulamalarında Hiper Parametre Seçim Yöntemleri*, Medium. <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-model-dogrulama-ve-hiper-parametre-secim-yontemleri-823812d95f3> (Son Ulaşma Tarihi: 06.06.2021)

Ng A., (2020, Kasım). *Neural Network and Deep Learning*, <https://www.coursera.org/learn/neural-networks-deep-learning> (Son Ulaşma Tarihi: 06.06.2021)

Portilla J., *Complete Tensorflow2 and Keras Deep Learning Bootcamp*, <https://www.udemy.com/course/complete-tensorflow-2-and-keras-deep-learning-bootcamp/learn/lecture/16844518#announcements> (Son Ulaşma Tarihi: 06.06.2021)