T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÜMÜŞ/ONS PARİTESİ ÜZERİNE DERİN ÖĞRENME İLE FİNANSAL TAHMİN MODELLEMESİ GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Adem ÜNTEZ

Enstitü Anabilim Dalı : BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

Tez Danışmanı : Dr. Öğr. Üyesi Mümtaz İPEK

Ocak 2022

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÜMÜŞ/ONS PARİTESİ ÜZERİNE DERİN ÖĞRENME İLE FİNANSAL TAHMİN MODELLEMESİ GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Adem ÜNTEZ

Enstitü Anabilim Dalı : BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

Bu tez …./…./2022 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

Jüri Başkanı **Üye** **Üye**

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Adem Üntez

TEŞEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Dr. Öğr. Üyesi Mümtaz İpek'e teşekkürlerimi sunarım.

Ayrıca çalışmamın kodlama bölümlerinde büyük desteğini gördüğüm, kardeşim Sinan Üntez'e, yüksek lisans eğitimim boyunca maddi ve manevi desteklerini esirgemeyen sevgili aileme teşekkür ederim.

İÇİNDEKİLER

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Derin öğrenme, finansal tahmin, makine öğrenimi, yapay zeka, gümüş

Tüm dünyada kabul görmüş altın, gümüş, platin gibi madenler küresel olarak fiyatlarında dalgalanmalar yaşamakta ve yatırımcılar bu ürünleri değerlendirerek kazanç elde etmek istemektedirler. Bu ürünlerin grafik yön tahmini için teknik ve temel analiz yöntemleri kullanılmaktadır. Ayrıca derin öğrenme yöntemlerinin yaygınlaşması ile beraber grafik yön tahmini için daha farklı metotlar uygulamaya konulmuştur. Bu metotlardan en yaygınlarından bazıları ise derin öğrenme algoritmalarıdır. Bu çalışmada derin öğrenme algoritmalarından biri olan LSTM mimarisi ve bir zaman serisi metodu olan ARIMA mimarisi kullanılmıştır. Bu mimariler ile gümüş/ons paritesi üzerinden finansal tahmin modelleri geliştirilmiştir. Eğitim ve test verileri kurulan algoritmalara yüklenerek sistemin öğrenmesi sağlanmış ve gelecek 10 gün için gümüş/ons paritesi tahmin değerleri üretilmiştir. Algoritmaların başarısının doğruluk oranını artırmak için algoritmalar 10 defa çalıştırılmış ve çıkan 10 günlük tahmin verilerinin ortalaması alınmıştır. Bu yöntemin seçilmesinin sebebi algoritmalar her çalıştırıldığında farklı tahminler ve grafikler üretmesidir. Araştırmada elde edilen sonuçlara göre ARIMA mimarisi LSTM mimarisinden daha iyi değerler üretmiştir.

DEVELOPING FINANCIAL FORECASTING MODELING WITH DEEP LEARNİNG ON SILVER / OUNCE PARITY

SUMMARY

Keywords: Deep learning, financial forecasting, machine learning, artificial intelligence, silver

Metals such as gold, silver and platinum, which are accepted all over the world, experience fluctuations in their prices globally, and investors want to earn profits by evaluating these products. Technical and fundamental analysis methods are used for graphical direction estimation of these products. In addition, with the widespread use of deep learning methods, different methods have been put into practice for graphical direction estimation. Some of the most common of these methods are deep learning algorithms. In this study, LSTM architecture, which is one of the deep learning algorithms, and ARIMA architecture, which is a time series method, were used. With these architectures, financial forecasting models have been developed over silver/ounce parity. The training and test data were loaded into the established algorithms, allowing the system to learn, and silver/ounce parity prediction values for the next 10 days were produced. In order to increase the accuracy of the success of the algorithms, the algorithms were run 10 times and the average of the 10-day forecast data was taken. The reason for choosing this method is that it produces different predictions and graphs each time the algorithms are run. According to the results obtained in the research, ARIMA architecture produced better values than LSTM architecture.

BÖLÜM 1. GİRİŞ

Bilgisayarların günlük hayata girmesi ile birlikte pek çok görevi insan gücünden makine gücüne devretmiş bulunmaktadır. En basit hesap makinesinden, robotların yönettiği montaj hattı üretim sistemlerine kadar makineler ve bilgisayarlar hayatı kolaylaştırmaktadır. İkinci Dünya Savaşı sırasında Alan Turing tarafından yapay zekanın temellerinin atılması ile bilgisayarlar çok daha işlevsel hale gelmiş ve zeki canlılara benzer şekilde bazı faaliyetleri yerine getirmeye başlamıştır. Yapay zekanın etki ettiği pek çok alan bulunmaktadır. Önerici sistemler; kullanıcıların geçmiş davranışlarına göre yeni içerik önerileri sunan, sosyal medya ve online sitelerde kullanılan sistemlerdir. Makine çevirisi olarak ifade edilen bir dilden başka bir dile çeviri yapmayı sağlayan sistemler mevcuttur. Ses ve görüntü işleme ve anlamlandırma gibi görevleri yerine getiren sinyal işleme alanında yapay zeka kullanılmaktadır. Son olarak regresyon analizi; geçmiş verilerin değerlendirilerek geleceğe yönelik tahmin modellerinin oluşturulmasını sağlamaktadır. Bu konuda ekonomik, finansal veya başka herhangi bir alanda sayısal verilerin tutarlı ve yeterli çoklukta olması analiz güvenilirliğini artırmaktadır.

Şekil 1.1. Yapay zeka, makine öğrenimi ve derin öğrenmenin kronolojisi (Kaynak: Nvidia)

Makine öğrenmesi ve derin öğrenme ise yapay zeka fikrinden daha sonra ortaya çıkmıştır. Makine öğrenimi programlanmadığı sonuçları bile çıktı olarak verebilen bir tür yapay zeka olarak kabul edilmektedir. 1959 yılında Arthur Samuel makine öğrenimini: "makinelerin bilhassa programlanmadığı sonuçları öğrenebilme kabiliyeti" olarak tanımlamıştır. Makine öğrenmesi tek katmanda işlem yaparken derin öğrenme çok sayıda katmanda işlem gerçekleştirir. Makine öğrenmesinde bir girdinin tanımlanabilmesi için parametreleri siz belirlersiniz. Derin öğrenme ise parametreleri kendi belirleyerek kendi kurallarını oluşturur. Bu durumda girilen verilerde insan duyuları ile algılanamayan farkları derin öğrenme algoritmaları tanımlayabilir.

Derin öğrenme, 2010 yılında Toronto Üniversitesi'nden Alex Krizhevsky, Ilya Sutskever ve Geoffrey E. Hinton'ın "AlexNet" ismini verdikleri bir yapay sinir ağı modeli geliştirmesiyle beraber adından söz ettirmeye başlamıştır. Derin öğrenme, basitçe tanımlamak gerekirse, çok katmanlı yapay sinir ağlarının (Multi Layer Artificial Neural Networks) geri yayılım (backpropagation) isimli bir algoritma ile eğitilmesine verilen isimdir (Krizhevsky, 2012). Derin öğrenme modelleri eğitildikleri veri ile sınıflandırma, regresyon analizi ve zaman serilerinde tahmin gibi uygulamalarda büyük başarılar göstermektedir. Bu modelleri kullanarak borsa, kıymetli maden ve kripto piyasası gibi pek çok alanda geleceğe yönelik finansal yön tahminleri yapmak mümkündür. Bu amaçla kullanılan pek çok derin öğrenme modeli olup her birinin başarı oranları kullanılan veri setine, zaman aralığına veya uygulanan teknik gibi pek çok değişkene göre farklılık göstermektedir.

Günümüz ekonomik dünyasında insanlar farklı yatırım alanlarını değerlendirerek mevcut sermayelerini artırmak istemektedir. Tüm dünyada kabul görmüş altın, gümüş, platin gibi madenler küresel olarak fiyatlarında dalgalanmalar yaşamakta ve yatırımcılar bu ürünleri değerlendirerek kazanç elde etmek istemektedirler. Bu ürünlerin grafik yön tahmini için teknik ve temel analiz yöntemleri kullanılmaktadır. Ayrıca derin öğrenme yöntemlerinin yaygınlaşması ile beraber grafik yön tahmini için daha farklı metotlar uygulamaya konulmuştur.

Bu çalışmada öncelikle derin öğrenme ve finansal tahmin modelleri hakkında geniş bir literatür taraması yapılacaktır. Sonrasında gümüş/ons paritesinde kullanmak istediğimiz veriler belirli bir zaman aralığı için elde edilecek. LSTM ve ARIMA algoritmalarına bu veriler öğretilmiş ve geleceğe yönelik tahminler üretilmiştir. Daha sonra iki algoritma tarafından üretilen tahmin değerlerinin performans ve doğrulukları incelenmiştir. Sonuç olarak ARIMA algoritması LSTM algoritmasından daha iyi performans göstermiştir.

BÖLÜM 2. YAPAY ZEKA

Önceki çağlarda düşünebilen makineler üzerine felsefe ve mitler konuşulurken 19. yüzyılda bilgisayarların keşfi ile insan beyni gibi öğrenen, düşünebilen ve karar verebilen programlar geliştirme konusunda ilk adım atılmış olmuştur. Nöroloji alanında yapılan çalışmalar göstermiştir ki insan beynindeki nöronlar ya hep ya hiç prensibi ile çalışmaktadır. Yani insan beyni elektriksel bir nöron ağı şeklinde çalışmaktadır. Norbert Wiener'in sibernetik tanımı, Claude Shannon'ın bilgi kuramı ve Alan Turing'in hesaplama teorisi arasındaki yakın ilişki, elektronik bir beyin inşa etmenin mümkün olabileceğini göstermiştir.

Yapay zekanın ilk modern tanımı 1950 yılında Alan Turing tarafından yapılmıştır. Alan Turing yazdığı makalede "Eğer bir makine karşısındaki insanla, makine olduğu ayırt edilemeyecek şekilde bir konuşma gerçekleştirebiliyorsa, düşünebilen bir makinedir" ifadesi ile ilk yapay zeka tanımını yapmıştır. 70'li yıllara kadar yapay zeka üzerine pek çok teorem geliştirilmiş, konferanslar verilmiş ve büyük yatırımlar yapılmıştır. Ancak 70'li yıllarda bazı zorluklardan dolayı "ilk yapay zeka kışı" denilen bir duraksama dönemi gerçekleşmiştir. Bu engellerden en önde geleni yeteri kadar güçlü bilgisayarların olmamasıydı. Yeterli ram ve işlemci gücüne sahip olmayan zamanın bilgisayarları yapay zeka destekli sistemleri çalıştırmakta yetersiz kalmıştır. Ayrıca uzun işlem süresi ve yeterli veri olmaması çalışmaların verimliliğini önemli ölçüde düşürmüştür. Bu problemlerden kaynaklı olarak yapay zeka çalışmaları başarısız olarak değerlendirilmiş ve dünya çapında yapılan fonlamaların pek çoğu kesilmiştir. 1980 sonrası uzman sistemlerin geliştirilmesi ile yapay zeka yeniden gündeme gelmiştir. Özellikle Japon firmaları "beşinci nesil bilgisayar projesi" ile uzman sistemlere büyük yatırımlar yapmışlardır. Şirketler tasarruf etmek için uzman sistemler geliştirmeye ilgi duymaya başlamıştır. Aynı dönemde geliştirilen yapay zeka programları satranç ustası insanları yenmiştir. Yeni bir yöntem olarak, bilgiyi öğrenip

işleyen yapay sinir ağlarının da (Artificial neural network – ANN) temeli bu dönemde atılmıştır. 1987-1993 yılları arasında yine benzer sebeplerden ve ekonomik sıkıntılardan kaynaklı ikinci bir yapay zeka kışı dönemi yaşanmıştır. Ancak sonraki yıllarda yapay zeka yükselişini adım adım sürdürmüştür. 1997 yılında IBM'in Deep Blue programı dünya satranç şampiyonu Gary Kasparov'u yenerek büyük bir başarıya imza atmış ve adından söz ettirmiştir. Bu başarının temelinde yıllar içinde gelişmiş bilgisayarlar yatmaktadır. Moore'un yasası denilen bir ölçüte göre bilgisayarların hız ve hafıza kapasiteleri, entegre devrelerde transistor sayısının artışına göre her iki yılda iki katına çıkmaktadır. Bu durumda yapay zeka için gerekli donanım gücü her geçen yıl artmaktadır. Ray Kurzweil, Moore yasasını göz önüne alarak yaptığı öngörüde 2029 yılı itibariyle insan düzeyinde zekaya sahip yapay zekalı makinelerin var olabileceğini söylemiştir.

2000'li yılların başında itibaren büyük miktarda veriye erişimin mümkün olması, daha ucuz ve hızlı bilgisayarlar ve ileri makine öğrenimi teknikleri pek çok problemin çözümünde kullanılmaya başlamıştır. Günümüzde yapay zeka, makine öğrenimi, derin öğrenme, big data ve AGI (artificial general intelligence) gibi alt başlıklara bölünmüştür (Buchanan,2005).

2.1. Makine Öğrenimi ve Derin Öğrenme

Makine öğrenimi yapay zekanın bir alt kümesi olarak değerlendirilebilir. Makine öğrenimi "training data" denilen örnek veri kümesini kullanarak bir görevi gerçekleştirmek üzere tam olarak programlanmamasına rağmen karar verebilen veya tahmin yapabilen bir model şeklinde tanımlanabilir. Zaman içinde kullanılan datanın güncellenmesi ve devamının sağlanması ile makine öğrenimi algoritmaları daha tutarlı tahminler oluşturabilir ve daha doğru kararlar verebilirler. Makine öğrenimi algoritmaları bir görevi gerçekleştirmek üzere programlanmazlar ancak verilen veri kümesini kullanarak tahmin ve kararları gerçekleştirmek üzere bir model inşa ederek çalışırlar. Makine öğrenmesi algoritmaları sanayide, endüstride, e-marketing sektöründe ve daha pek çok alanda kullanılmaktadır. Başlıca kullanım amaçları arasında e-mail filtreleme (spam detection), fraud tespiti (fraud recognition), doğal dil

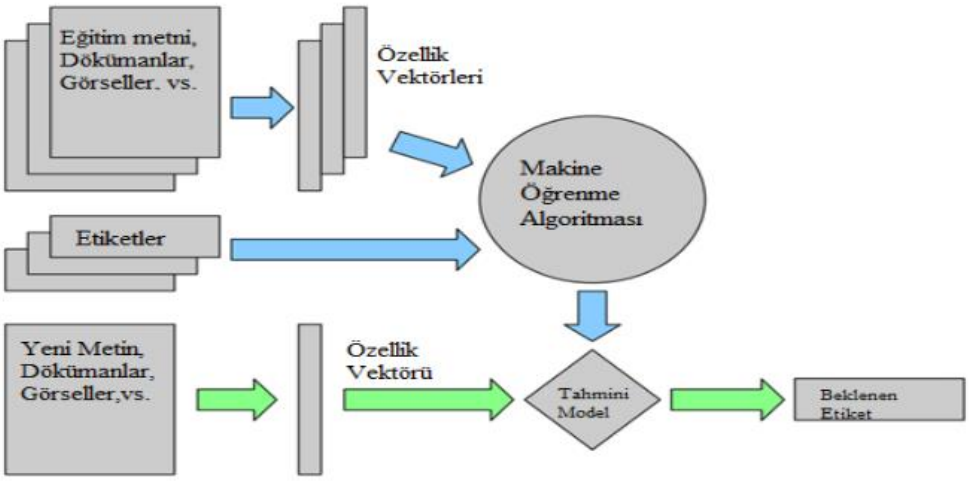
işleme (natural language processing), yüz tanıma (face recognition), online alışveriş sitelerinde ürün tavsiye etme (product recommendation) ve medikal teşhisler örnek olarak verilebilir.

Şekil 2.1. Yapay zeka, makine öğrenmesi ve derin öğrenme ilişkisi

Derin öğrenme ise makine öğrenmesinin alt kümesi olarak değerlendirilebilir. Birbirinden çok farklı olmamakla birlikte çalışma alanları da aynıdır ancak derin öğrenmede kullanılan yapay sinir ağları (artificial neural networks) biyolojik bir sinir hücresini taklit eden bir yapıya sahiptir. Bir sinir hücresi almış olduğu bilgiyi daha önceki bilgilerle karşılaştırarak bir çıkarımda bulunmaktadır. Derin öğrenme algoritmaları, bir insan beyni gibi öğeleri etiketleyerek ve çeşitli kategorilere atayarak bilgileri çözümlemektedir (Jakhar ve Kaur, 2009). Buradaki öğrenme kavramı denetimli, yarı denetimli veya denetimsiz olabilir.

2.1.1. Denetimli ve denetimsiz öğrenme

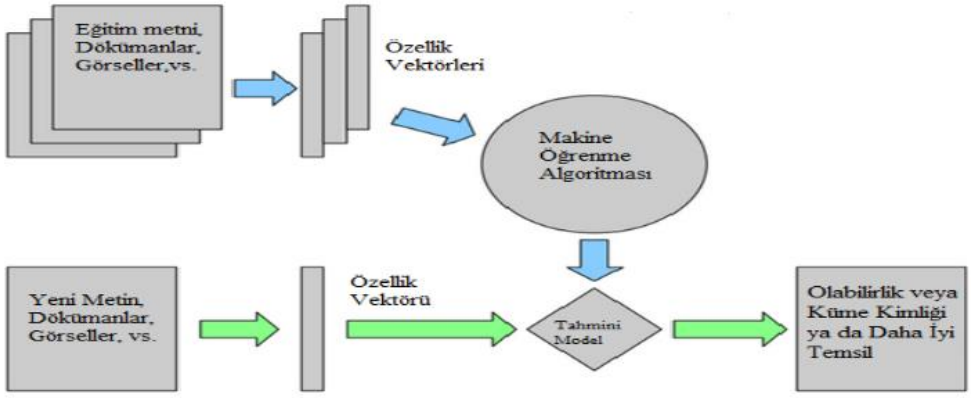
Örnek girdi-çıktı çiftlerini esas alarak bir girdiyi bir çıktıya eşleyen makine öğrenmesi algoritmalarıdır. Yapılan eşleşmeler arasındaki bağlantıları kullanarak algoritma eğitilmiş olur. Denetimli öğrenme yapılması için girdilerin ve çıktıların olduğu bir eğitim seti bir gözetmen eşliğinde algoritmaya yedirilerek anlamlı sonuçlar çıkarması sağlanır. Makine öğrenmiş olduğu ilişkilerden yola çıkarak hiç tanıtılmamış bir örneklem için varsayımlarda bulunabilir. Denetimli öğrenme, eğitim setinden elde edilen çıktılar ile gerçek sonuçlar arasındaki farkı en aza indirgemeye çalışır. Maillerin spam olarak tespit edilmesi denetimli öğrenme uygulamasına örnek verilebilir. Denetimli öğrenme algoritmaları sınıflandırma ve regresyon amaçlı algoritmalar için tercih edilir (Mohri ve diğ, 2012). Destek vektör makineleri, lineer regresyon, lojistik regresyon, naive bayes sınıflandırıcıları, karar ağaçları, k-en yakın komşu algoritmaları ve yapay sinir ağları sıklıkla kullanılan denetimli öğrenme algoritmalarına örnek verilebilir. Bu algoritmaların içinden tercih yapılabilmesi için dikkat edilmesi gereken bazı faktörler mevcuttur. Bunlar verinin homojenliği, verinin temizliği ve girdi-çıktı etkileşimlerinin tutarlılığıdır (Russell ve Norvig, 2010).



Şekil 2.2. Denetimli Öğrenme Modeli (Kaynak: Tiken, 2015)

Denetimsiz öğrenme sisteme eğitim seti olarak sadece girdilerin verildiği, çıktı ve etiketleme bilgisinin verilmediği algoritmalardır. Sisteme eğitim sırasında bir

gözetmen bulunması da gerekmemektedir. Verileri kullanan sistem girdi-çıktı bağıntılarını kendisi inceleyerek sınıflandırma, kümeleme gibi işlemleri yapmaya çalışır. Denetimsiz öğrenme algoritmaları veri içerisinde bağıntıları bulmaya çalışır ancak veri setinde sınıflandırma veya çıktı bilgisi olmadığı için sistem performansının ölçülmesi zordur (Mohri ve diğ, 2012). Verilerin sürekli işlenerek bağıntıların sistem tarafından öğrenilmesi sağlanmaktadır. Bu bakımdan veri ne kadar fazla ise o kadar tutarlı sonuçlar elde edilmesi muhtemeldir. Bu yönüyle denetimsiz öğrenme algoritmaları veri madenciliğine benzetilmektedir (Bell, 2014).



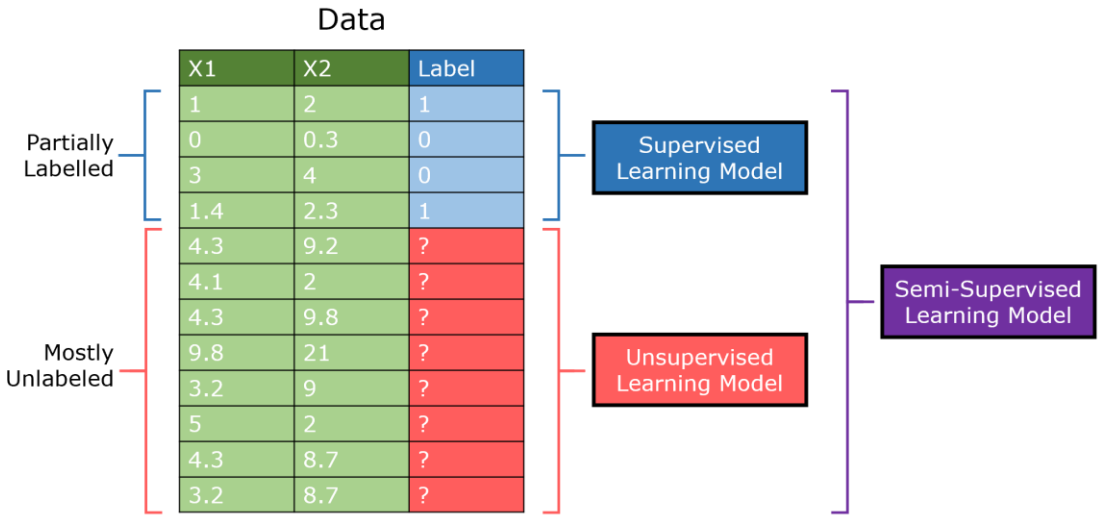
Şekil 2.3. Denetimsiz Öğrenme Modeli (Kaynak: Tiken, 2015)

Denetimsiz öğrenme algoritmaları kümeleme ve anomali tespiti için uygulanmaktadır. Bu algoritmalardan bazıları şunlardır:

- Tekil değer ayrıştırması
- Korelasyon analizi
- Temel bileşen analizi
- Hiyerarşik kümeleme
- Faktör analizi
- K-ortalama kümeleme

2.1.2. Yarı denetimli öğrenme (semi-supervised learning)

Bu yöntemde kullanılan eğitim setindeki verilerde hem denetimli hem denetimsiz öğrenme yöntemlerinde olduğu gibi etiketli ve etiketsiz veriler mevcuttur. Tüm verilerde etiketleme işleminin zor ve maliyetli olduğu durumlarda bu yönteme başvurulur. Yarı denetimli öğrenme yöntemi regresyon, sınıflandırma, sıralama gibi problemleri çözmek için kullanılabilir (Mohri ve diğ, 2012).



Şekil 2.4. Yarı Denetimli Öğrenme Yöntemi (Andre Ye)

2.2. Derin Öğrenme Algoritmaları

Derin öğrenme algoritmaları yapay zeka ve makine öğrenmesi terimlerinin alt dalı olarak değerlendirilebilir. İnsan beyin yapısından ve sinir hücrelerinin çalışma prensibinden ilham alarak geliştirilen derin öğrenme bir tür makine öğrenmesidir. Biyolojik sinir hücrelerinde olduğu gibi yapay nöronlar giriş sinyallerini alır, toplar ve işleyerek çıkışlara iletirler (Şişmanoğlu ve diğ., 2019). Derin öğrenme, özellik çıkarma ve dönüştürme için birçok doğrusal olmayan işlem birimi katmanını kullanır.

Her ardışık katman bir önceki katmandaki çıktıyı girdi olarak kabul eder (Şeker ve diğ., 2017).

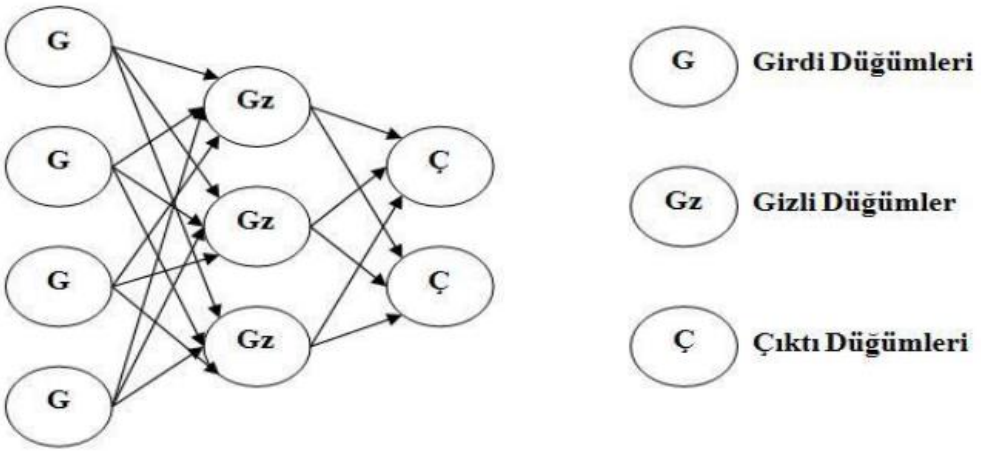
İnternet her geçen gün gelişmekte ve dijital ortamda oluşturulan veriler çok büyük boyutlara ulaşmaktadır. Bu büyük verilerle klasik makine öğrenmesi algoritmaları başa çıkmakta zorlanmakta ve yeterli performansı göstermemektedir. Derin öğrenme algoritmaları ise büyük verilerle yapılan işlemlerde daha verimli sonuçlar üretmektedir. Derin öğrenme sınıflandırma özellik ve görevlerini doğrudan verileri kullanarak gerçekleştirmeyi öğrenir. Bu algoritmalar, insan düzeyinde öğrenmeyi başarmak hatta bu seviyeyi aşmak için çok sayıda sinir ağı mimarisi kullanılarak eğitilebilir (Şişmanoğlu ve diğ., 2019). Bu kapsamda derin öğrenme algoritmalarını tanıtmadan önce yapay sinir ağlarını ve çalışma prensiplerini anlamak gerekmektedir.

2.2.1. Yapay sinir ağları (YSA)

Yapay sinir ağları (YSA) bağlamında ilk kez "Derin Öğrenme" (deep learning) ifadesi 2000 yılında Igor Aizenberg ve arkadaşları tarafından tanıtılmıştır. 2006 yılında Geoffrey Hinton tarafından yayınlanan makalede, çok katmanlı ileri beslemeli bir sinir ağının her iterasyonda bir katmanı etkili şekilde nasıl eğitebildiğini göstermiştir (Şeker ve diğ., 2017).

YSA, yeni bilgiler üretebilme, keşfedebilme, karar verme gibi insan beynine ait özellikleri taklit edebilen bilgisayar temelli sistemlerdir. Bu özellikleri bilinen programlama dilleriyle oluşturmak imkansıza yakın olduğundan YSA bu işi mümkün kılmaktadır.

YSA birbirine bağlı ve paralel bir düzen içinde çalışan düğümlerden oluşmaktadır. Bu düğümler aşağıdaki şekilde göründüğü gibi bir ağ yapısı oluşturmaktadır.



Şekil 2.5. Yapay Sinir Ağlarının Temel Gösterimi

Düğümleri birbirine bağlayan bağlantıların farklı değerleri vardır ve her bir farklı değer ağın yapısı sayesinde farklı düğümlere iletilerek öğrenme sağlanır. Bu yapının vazifesi girdilere karşılık çıktı üretebilmektir.

2.2.1.1. Yapay sinir ağlarının özellikleri

- Doğrusal Olmama: YSA'nın temel işlem birimi olan sinir hücresi doğrusal olmadığından sinir hücrelerinin bir araya gelmesiyle oluşan ağ yapısı da doğrusal değildir.

- Öğrenme: Sisteme giren ve çıkan veriler arasında eşleştirme yapmaktadır. Bu eşleştirme bir tür haritalandırma olarak görülebilir. YSA'nın istenilen davranışı gösterebilmesi için buna uygun şekilde ayarlanması gerekmektedir. Bu da sinir hücreleri arasında doğru bağlantı kurulması ve uygun ağırlıklandırma yapılması ile mümkündür.

- Adaptasyon: Değişen çevre şartlarına bağlı olarak bağlantı ağırlıklarını ayarlayabilme yeteneğine sahiptir.

- Genelleme: Her bir nöron diğer nöronlarda meydana gelen değişikliklerden etkilenebilir ve karşılaşmadığı örneklere karşı bir genelleme yaparak istenen tepkiyi verebilir.

- Hata Toleransı: Yapay sinir ağlarının mantıksal ve fiziksel hata toleransı yüksek olduğundan küçük hatalardan etkilenmez. Bazı bağlantıların koparak işlevselliğini yitirmesi YSA'nın çıktılarını çok fazla etkilemez. Veriler tüm ağa yayılmış halde bulunduğundan sistemin hata toleransı yüksektir.

- Uygulanabilirlik: YSA kullanılarak büyük boyutlarda bütünleşik devre uygulamaları yapılabilir.

- Analiz ve Tasarım Birliği: Yapay sinir ağlarının temel birimi olan hücrenin yapısal özellikleri bütün YSA yapılarında büyük oranda aynıdır. Bu durumda farklı uygulama alanlarına sahip YSA modelleri birbirine benzerlik gösteren teori ve algoritmaları uygulayabilirler. Bu özellik sayesinde YSA kullanılarak çözülen problemlerin çözümünde büyük kolaylık sağlanmış olur (Doğan, 2003).

2.2.1.2. Yapay sinir ağlarının avantajları

- Daha önce öğrenmediği bir veri ile karşılaştığında sistemdeki mevcut bağlantıları değerlendirerek bir çıktı üretebilme kapasitesine sahiptir.

- YSA doğrusal bir sisteme sahip olmadığından çok karmaşık problemleri doğrusal çözüm tekniklerinden daha doğru bir şekilde çözebilirler. Doğrusal olmayan problemlerin matematiksel olarak çözümü zordur.

- Ağ üzerindeki ağırlık katsayıları değiştiğinde YSA kendini bu duruma adapte ederek problemin çözümünü sağlayabilir.

- Geleneksel yöntemler küçük hatalara karşı hassastır ve bu durum çıktılarda büyük farklara sebep olabilmektedir. Buna rağmen yapay sinir ağlarında

durum farklıdır çünkü sistemdeki bir veya birkaç nöron zarar görse bile sistem geleneksel yöntemlere göre oldukça az hasar görür.

- YSA verilerden faydalanarak gizli ilişkileri ortaya çıkarmakta büyük başarı göstermektedir.

- YSA'nın matematiksel olarak ifade edilmesi gerekmez.

- YSA bir defa eğitildikten sonra yeni gördüğü veri kümesinden doğrudan çıktı ve bağıntı oluşturabilir.

2.2.1.3. Yapay sinir ağlarının dezavantajları

- YSA'nın farklı sistemlere uyarlanması kolay değildir.

- Sistemin eğitilmesi uzun zaman alabilmektedir. Bu sebepten maliyeti de yüksek olabilmektedir.

- Problemlerin çözümünde hata yapılıp istenilen çözüme kavuşulamayabilir. Bunun nedeni sistem eğitilirken herhangi bir fonksiyon kullanılmamasıdır. Sistem fonksiyon bulsa bile yeterli veri olmaması da yanlış çözüme götürebilmektedir.

- Ağın verdiği sonuçları değerlendirmek sistemin içinde ne olduğu bilinmediği durumlarda mümkün olmamaktadır.

- Sistemdeki düğümlerin sayısındaki artış çalışma zamanının artmasına neden olmaktadır. Ağın kalite ve kapasitesi uygulama esnasındaki hızı ile doğru orantılıdır (Tolon ve Tosunoğlu, 2008).

2.2.1.4. Yapay sinir ağlarının uygulama alanları

Tahmin, modelleme ve sınıflandırma gibi pek çok alanda YSA kullanılabilmektedir. Yapılan çalışmalar incelendiğinde daha çok matematiksel çözümün mümkün olmadığı ve herhangi bir algoritmanın bulunmadığı karmaşık, kesin olmayan, gürültülü, çok boyutlu ve sadece örneklerin mevcut olduğu problemlerin çözümünde kullanılmaktadır. Yapay sinir ağlarının gerçekleştirdiği fonksiyonlar örüntü tanıma, örüntü işleme, optimizasyon, veri sıkıştırma, sınıflandırma, ilişkilendirme, sinyal filtreleme, doğrusal olmayan sistem modelleme ve sinyal işleme olarak sıralanabilir. YSA, eğlence, elektronik, bankacılık, finans, sigortacılık, uzay, robotik, dil, sağlık, güvenlik, petrokimya, telekomünikasyon, üretim, savunma, otomotiv gibi sektörlerde uygulanmaktadır (Çayıroğlu, 2013).

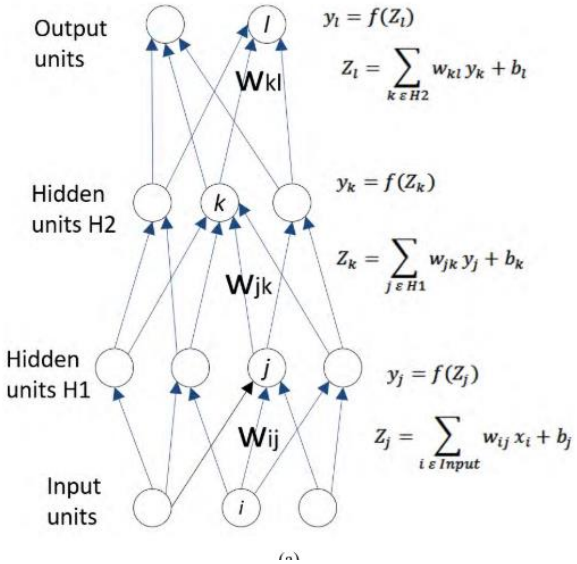
2.2.2. Yapay sinir ağlarının sınıflandırılması

YSA aşağıda gösterilen beş farklı şekilde sınıflandırılmaktadır.

1. İleri beslemeli sinir ağları (Feedforward Neural Network)
2. Tekrarlayan sinir ağları (Recurrent Neural Network (RNN))
3. Radyal temel işlev ağı (Radial Basis Function Neural Network)
4. Kendi kendini düzenleyen ağlar (Kohonen Self Organizing Neural Network)
5. Modüler sinir ağı (Modular Neural Network)

2.2.2.1. İleri beslemeli sinir ağları

İleri beslemeli ağlarda bilgi sadece tek yönde, girdi katmanlarından çıktı katmanlarına doğru gizli düğümler üzerinden (eğer varsa) aktarılmaktadır. Veriler sistem içinde daire oluşturmamakta ve geri besleme yapmamaktadır. Şekil 2.2. ileri beslemeli çok katmanlı bir yapay sinir ağındaki fonksiyonları ve değerleri göstermektedir.



Şekil 2.6. İleri beslemeli çok katmanlı ağ uygulaması

Buradaki Z değeri girdilerin ağırlıklı toplamını, y ise her katmanda Z'nin doğrusal olmayan aktivasyon fonksiyonunu f'yi temsil eder. W, alt simge harfleriyle gösterilen bitişik katmanlardaki iki birim arasındaki ağırlıkları temsil eder ve b ise birimin önyargı değerini temsil eder.

2.2.2.2. Tekrarlayan sinir ağları

İleri beslemeli sinir ağlarından farklı olarak, işlem birimleri Tekrarlayan sinir ağları (RNN)'de bir döngü oluşturur. Bir katmanın çıktısı, ağdaki tek katman olan bir sonraki katmanın girdisi haline gelir, böylece katmanın çıktısı, geri bildirim döngüsü oluşturan kendisine bir girdi haline gelir. Bu yapı ağın önceki durumlar hakkında belleğe sahip olmasını ve bunu mevcut çıktıyı etkilemek için kullanmasını sağlar. İleri beslemeli sinir ağlarından önemli bir fark olarak RNN bir dizi girdi alarak bir dizi çıktı üretebilir. Bu durum, dil işleme gibi zaman aşamalı girdi verilerinin işlenmesini gerektiren uygulamalar için çok faydalı olmaktadır.

2.2.2.3. Radyal temel işlev ağı

Radyal temel işlev ağları sınıflandırma, işlev yaklaşımı ve zaman serisi tahmin problemleri gibi işlemler için kullanılır. Girdi ve çıktı katmanı ile gizli katmanlardan oluşur. Gizli katmanlar bir radyal temel işlevi (gauss işlevi olarak uygulanır) içerir ve her düğüm bir küme merkezini temsil eder. Ağ, girdiyi bir merkeze atamayı öğrenir ve çıktı katmanı, sınıflandırma veya çıkarım gerçekleştirmek için radyal temel işlevinin çıktılarını ve ağırlık parametrelerini birleştirir.

2.2.2.4. Kendi kendini düzenleyen ağlar

Kohonen kendi kendini organize eden sinir ağı, denetimsiz öğrenme yaklaşımı ile girdi verilerini kullanarak ağ modelini kendisi oluşturur. Girdi ve çıktı katmanları olmak üzere tamamen birbirine bağlı iki katmandan oluşur. Çıktı katmanı iki boyutlu bir sistem olarak düzenlenmiştir. Aktivasyon fonksiyonu yoktur ve ağırlıklar çıktı katmanı düğümünün niteliklerini (pozisyonunu) temsil eder. Ağırlıklara göre giriş verileri ile her çıkış katmanı düğümü arasındaki Öklid mesafesi hesaplanır. Giriş verilerinden en yakın düğüm ve komşularının ağırlıkları, aşağıdaki formülle onları giriş verilerine yaklaştırmak için güncellenir.

$$wi\ (t + 1) = wi\ (t) + \alpha(t)\eta j * i(x(t) - wi\ (t)) \quad (2.1)$$

Burada x (t), t anında giriş verisidir, wi (t) t anında i'inci ağırlıktır ve ηj ∗ i, i'nci ve j'nci düğümler arasındaki komşuluk fonksiyonudur.

2.2.2.5. Modüler sinir ağı

Modüler sinir ağı büyük ağı daha küçük ve bağımsız sinir ağı modüllerine ayırır. Bu küçük ağlar, daha sonra tüm ağın tek bir çıktısının parçası olarak birleştirilen belirli bir görevi yerine getirir (Shrestha and Mahmood, 2019).

2.2.3. Derin sinir ağları

Derin sinir ağları adı verilen birkaç gizli katmana sahip yapay sinir ağları, çeşitli makine öğrenimi görevlerindeki benzeri görülmemiş başarıları nedeniyle popüler hale gelmiştir. Derin sinir ağları tek katmanlı gizli birimlere sahip sığ ağlara göre daha fazla tercih edilmekte ve beşten fazla katmanla uygulanmaktadır. Bu modellerin ara katmanları, girdilerinin birkaç parçasını aynı çıktıya eşleyebilir. Bu şekilde hesaplanan fonksiyonların katman bazlı bileşimi, katman sayısı arttıkça üssel olarak düşük seviyeli hesaplamaları yeniden kullanır. Bu anahtar özellik, derin ağların oldukça karmaşık ve yapılandırılmış işlevleri hesaplamasını sağlar (Montufar ve diğ., 2014).

YSA'da daha derin mimariler kullanan derin sinir ağları bir katmandaki birim sayısının artırılması ile daha karmaşık işlemleri yerine getirmek için kullanılmaktadır. Yeterince etiketli eğitim veri kümeleri ve uygun modeller verildiğinde, derin öğrenme yaklaşımları, insanların operasyon kolaylığı için haritalama işlevleri oluşturmasına yardımcı olabilir. Derin sinir ağları mimarilerinden bazıları şunlardır:

- Sınırlandırılmış Boltzmann Makineleri
- Derin inanç ağı
- Autoencode
- Evrişimli sinir ağları (Liu ve diğ., 2017)

Ayrıca bu çalışmada bahsedilen diğer derin öğrenme mimarileri aşağıdaki gibidir:

- Tekrarlayan sinir ağları (Recurrent neural network - RNN)
- Uzun Kısa Vadeli Hafıza Ağları (Long Short Term Memory - LSTM)
- Derin ileri beslemeli ağlar (Deep feed-forward networks)

2.3. Derin Öğrenme Mimarileri

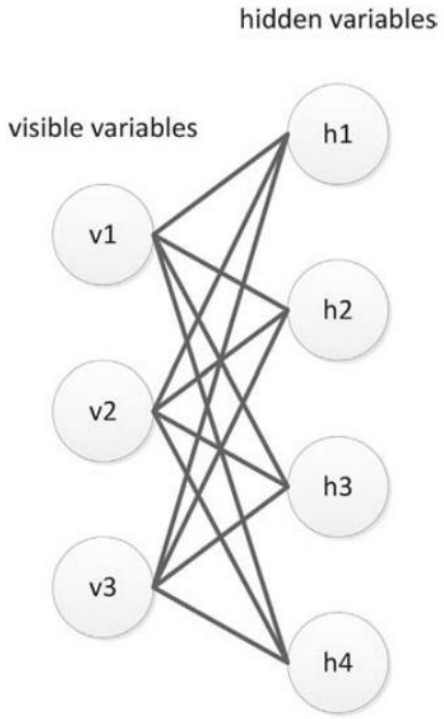
Bu başlık altında, Sınırlandırılmış boltzmann makineleri, derin inanç ağları, otomatik kodlayıcı, evrişimli sinir ağları, tekrarlayan sinir ağları, uzun kısa vadeli hafıza ağları, derin ileri beslemeli ağlar detaylı şekilde açıklanmıştır.

2.3.1. Sınırlandırılmış boltzmann makineleri (RBM'ler)

RBM'ler, tarihsel önemi ve göreceli basitlikleri nedeniyle derin öğrenme ağlarında yaygın olarak kullanılmaktadır. RBM ilk olarak Smolensky tarafından bir konsept olarak önerilmiş ve 2006 yılında Hinton çalışmasını yayınladıktan sonra öne çıkmıştır. RBM'ler, girdilerine göre olasılık dağılımını öğrenebilen YSA'nın stokastik modellerini oluşturmak için kullanılmıştır. RBM'ler Boltzmann makinelerinin (BM'ler) bir çeşidinden oluşur. BM'ler, çift yönlü olarak bağlanmış stokastik işleme birimleri olan yapay ağlar olarak yorumlanabilir. Bilinmeyen bir olasılık dağılımının yönlerini öğrenmek zor olduğundan, RBM'ler ağın topolojisini basitleştirmek ve modelin verimliliğini artırmak için önerilmiştir. Bir RBM'nin, bir katmanda stokastik görünür birimleri ve diğer katmanda stokastik gözlemlenebilir birimleri olan özel bir Markov rasgele alanları türü olduğu iyi bilinmektedir.

Şekil 2.7'de görüldüğü gibi nöronlar, bir RBM'de iki parçalı bir grafik oluşturmak için sınırlandırılmıştır. Görünen birimler ile gizli olanlar arasında tam bir bağlantı olduğu, aynı katmandaki birimler arasında bağlantı olmadığı görülebilir. Bir RBM'yi eğitmek için Gibbs örnekleyici benimsenmiştir. Bir katmanda rastgele bir durumla başlayıp Gibbs örneklemesi gerçekleştirerek, RBM'den veri üretilebilir. Bir katmandaki birimlerin durumları verildiğinde, diğer katmanlardaki tüm birimler güncellenecektir. Bu güncelleme işlemi, denge dağılımına ulaşılana kadar devam edecektir. Daha sonra, bir RBM içindeki ağırlıklar, bu RBM'nin olasılığını maksimize ederek elde edilir. Spesifik olarak, eğitim verilerinin log-olasılığının gradyanını alarak, ağırlıklar güncellenebilir:

Bu formülde ωij, görünür birim i ile gizli birim j arasındaki ağırlığı temsil eder. Eşitliğin sağındaki ifadeler ise görünür ve gizli birimlerin sırasıyla en alt katmanda ve en yüksek katmanda olduğu korelasyonlardır. Gradyan bazlı kontrastlı diverjans (CD) algoritması kullanılırken eğitim sürecinin daha verimli olacağı unutulmamalıdır. RBM eğitimi için CD algoritması Hinton (2002) tarafından geliştirilmiştir.



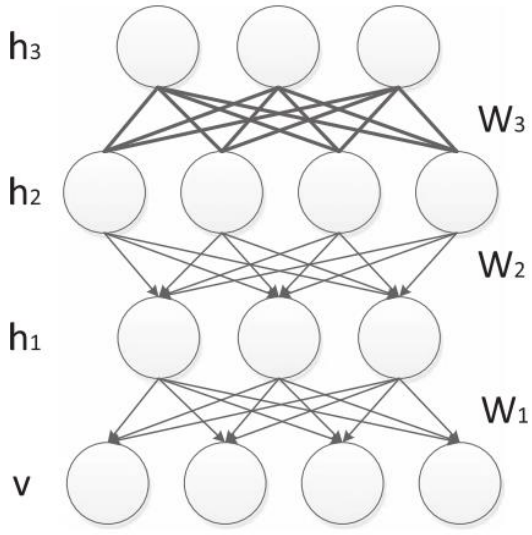
Şekil 2.7. RBM'in şematik diyagramı (Liu ve diğ., 2017)

Model ve hedef dağılım arasındaki farkın büyük olmadığını varsayarsak, Gibbs zinciri tarafından oluşturulan örnekleri negatif gradyanı yaklaşık olarak tahmin etmek için kullanılabilir. İdeal olarak, zincirin uzunluğu arttıkça, olasılığa katkısı azalır ve sıfıra iner. Bununla birlikte gradyan tahmininin gradyanın kendisini temsil edemeyeceği söylenebilir. Ayrıca, çoğu CD bileşeni ve karşılık gelen log-olabilirlik gradyanı eşit işaretlere sahiptir. Bu nedenle, kalıcı kontrastlı sapma adı verilen daha pratik bir algoritma önerilmiştir. Bu yaklaşımda Liu ve diğ. (2017), Gibbs Markov zincirinin başlangıç değerini belirli bir veri vektöründe aramak yerine kalıcı zincirlerin durumlarını izlemeyi önermiştir. Kalıcı zincirdeki gizli ve görünür birimlerin durumları her ağırlığın güncellenmesini takiben yenilenir. Bu şekilde, küçük bir öğrenme oranı bile daha doğru tahminler sağlarken güncellemeler ile kalıcı zincir durumları arasında çok fazla fark yaratmayacaktır (Liu ve diğ., 2017).

2.3.2. Derin inanç ağları (DBN)

DBN’lerde gizli ve görünür değişkenler karşılıklı olarak bağımsız değildir. Spesifik olarak, DBN'ler çoklu stokastik ve gizli değişken katmanlarından oluşur ve Bayesci olasılıksal üretken modelin özel bir formu olarak kabul edilebilir. YSA'larla karşılaştırıldığında, DBN'ler özellikle etiketlenmemiş verilerle ilgili sorunlara uygulandıklarında daha etkilidir.

Derin sinir ağlarında her iki bitişik katman bir RBM oluşturur. Her bir RBM'nin görünür katmanı, önceki RBM'nin gizli katmanına bağlanır ve üstteki iki katman yönsüzdür. Yukarıdaki katman ile alt katman arasındaki yönlendirilmiş bağlantı yukarıdan aşağı bir şekildedir. Bir DBN'deki farklı RBM katmanları sıralı olarak eğitilir: önce düşük RBM'ler, sonra daha yüksek olanlar eğitilir. Özellikler üst RBM tarafından çıkarıldıktan sonra, alt katmanlara geri yayılacaktır. Tek bir RBM ile karşılaştırıldığında, yığılmış model, daha güçlü öğrenme yetenekleri anlamına gelen günlük olasılığının üst sınırını artıracaktır.



Şekil 2.8. DBN’nin şematik diyagramı

Bir DBN'nin eğitim süreci iki aşamaya ayrılabilir: ön eğitim aşaması ve ince ayar aşamasıdır. Ön eğitim aşamasında, özellik çıkarma için aşağı-yukarı yönde denetimsiz

öğrenmeye dayalı bir eğitim gerçekleştirilir; ince ayar aşamasında iken, ağ parametrelerinin daha fazla ayarlanması için denetimli öğrenme tabanlı bir yukarı-aşağı algoritması gerçekleştirilir. DBN'lerin geliştirilmiş performansının büyük ölçüde, ağın başlangıç ağırlıklarının girdi verilerinin yapısından öğrenildiği eğitim öncesi aşamaya atfedilebileceği bilinmektedir. Rastgele başlatılanlarla karşılaştırıldığında, bu ağırlıklar global optimaya daha yakındır ve bu nedenle daha iyi performans sağlayabilir.

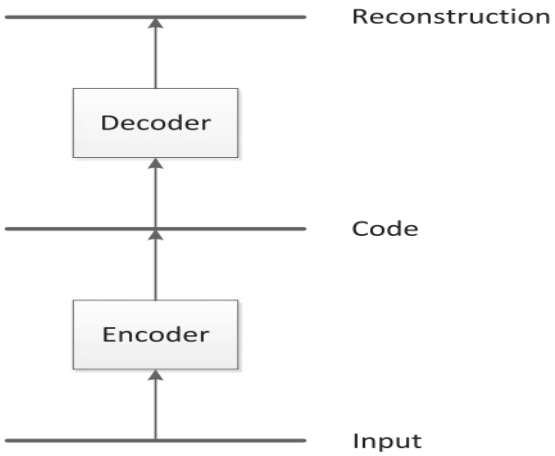
Önceki bölümde sunulan CD algoritması, bir DBN'yi önceden eğitmek için kullanılabilir. Bununla birlikte performans, özellikle giriş verileri sıkıştırıldığında genellikle tatmin edici değildir. Bu sorunun üstesinden gelmek için, ağın boyutuna ve derinliğine doğrusal zaman karmaşıklığında bir DBN'nin ağırlıklarını optimize eden bir katman katman öğrenme algoritması tanıtılmıştır. Katman katman öğrenme algoritmasında, DBN'yi oluşturan RBM'ler sıralı olarak eğitilir. Özellikle en düşük RBM'nin görünür katmanı, ilk olarak girdi olarak h (0) ile eğitilir. Görünür katmandaki değerler daha sonra gizli değişkenlerin aktivasyon olasılıklarının hesaplandığı gizli katmanlara aktarılır. Önceki RBM'de elde edilen gösterim, sonraki RBM için eğitim verileri olarak kullanılacak ve bu eğitim süreci tüm katmanlar geçilene kadar devam edecektir. Bu algoritmada, olasılık fonksiyonunun yaklaştırılması yalnızca bir adımda gerekli olduğundan, eğitim süresi önemli ölçüde azaltılmıştır. Genellikle derin ağlarda ortaya çıkan yetersiz uyum sorunu eğitim öncesi süreçte de aşılabilir. Bu ön eğitim algoritmasına, açgözlü katman katman denetimsiz eğitim algoritması da denilmektedir (Liu ve diğ., 2017).

2.3.3. Otomatik kodlayıcı (AE)

Başka bir YSA türü olan bir otomatik kodlayıcı (AE) otomatik ilişkilendirici olarak ta adlandırılır. Veri setini boyutsallık azaltma amacıyla verimli bir şekilde kodlamak için kullanılan denetimsiz bir öğrenme algoritmasıdır. Son zamanlarda, AE'ler üretken veri modellerini öğrenmek için kullanılmıştır. Giriş verileri önce soyut bir temsile ve daha sonra kodlayıcı işlevi tarafından orijinal formata dönüştürülür. Daha spesifik olarak, girdinin bu gösterimden yeniden yapılandırılabilmesi için girdiyi bazı temsillere

kodlamak üzere eğitilmiştir. Esasen, AE bu süreçte kimlik işlevine yaklaşmaya çalışır. AE'nin önemli bir avantajı, bu modelin yayılma sırasında sürekli olarak yararlı özellikleri çıkarabilmesi ve gereksiz bilgileri filtreleyebilmesidir. Ayrıca, kodlama sürecinde girdi vektörü daha düşük boyutlu bir temsile dönüştürüldüğü için, öğrenme sürecinin verimliliği artırılabilir.

AE'ler MLP'ye benzer tek gizli katmanlı ileri beslemeli bir sinir ağıdır. 1988'de Bourlard ve Kamp tarafından bulunan çok katmanlı bir algılayıcı (MLP) otomatik ilişkilendirme modundaki MLP, bilgi işleme gibi alanlarda veri sıkıştırma ve boyutsallık azalması sağlayabileceğini bulmuştur. Bir MLP ve bir AE arasındaki fark, AE'nin amacının girdiyi yeniden yapılandırmak iken MLP'nin amacının belirli girdilerle hedef değerleri tahmin etmesidir. Giriş katmanındaki ve çıktı katmanındaki düğümlerin sayısı aynıdır. Kodlama sürecinde, AE ilk olarak x girdi vektörünü bir ağırlık matrisi ω kullanarak gizli bir temsile (h) dönüştürür; daha sonra deşifreleme sürecinde, AE, başka bir ağırlık matrisi ω′ ile x´ elde etmek için h'yi orijinal formata geri eşler. Teorik olarak, ω ′, ω'nin transpozu olmalıdır. X ve x´ arasındaki ortalama yeniden yapılandırma hatasını en aza indirmek için parametre optimizasyonu benimsenmiştir. Ortalama kare hataları, girdi özelliklerinin varsayılan dağılımına göre yeniden yapılandırma doğruluğunu ölçmek için kullanılır.



Şekil 2.9. Autoencoderların şematik gösterimi (Liu ve diğ., 2017)

DBN'lere benzer şekilde, bir AE için eğitim süreci de iki aşamaya ayrılabilir: ilk aşama, denetimsiz öğrenmeyi kullanarak özellikleri öğrenmektir ve ikincisi, denetimli

öğrenmeyi kullanarak ağın ince ayarını yapmaktır. Özellikle ilk aşamada, x´ çıktı değerini elde etmek için ilk olarak her girdi için ileri besleme yayılımı gerçekleştirilir. Daha sonra x´ değerinin girdi değerinden sapmasını ölçmek için karesel hatalar kullanılır. Son olarak, ağırlıkları güncellemek için hata ağ üzerinden yeniden yayımlanacaktır. İnce ayar aşamasında, her katmanda uygun özelliklere sahip ağ ile, her katmandaki parametreleri ayarlamak için standart denetimli öğrenme yöntemi ve gradyan iniş algoritması benimsenebilir.

2.3.4. Evrişimli sinir ağları (CNN)

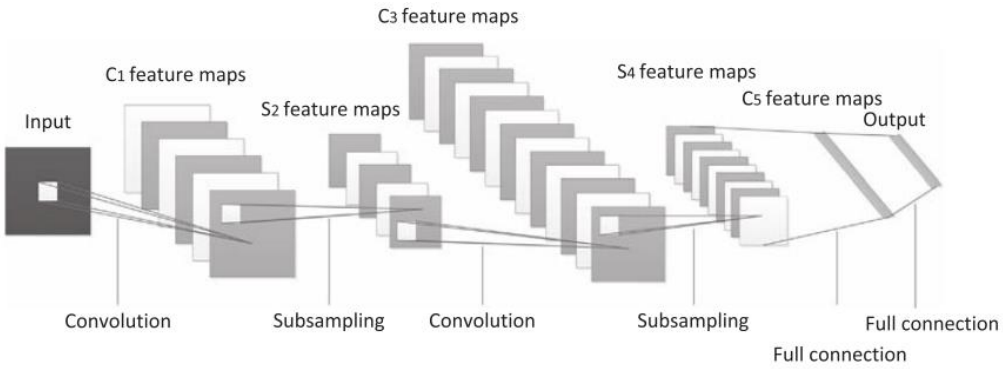
CNN'ler, ayırt edici derin mimarinin bir alt tipidir ve iki boyutlu verileri, görüntüler ve videolar gibi ızgara benzeri topolojiyle işlemede tatmin edici performans göstermişlerdir. CNN'lerin mimarisi, hayvan görsel korteks organizasyonundan esinlenmiştir. 1960'larda alıcı alanlar adı verilen bir kavram önerilmiştir. Hücrelerin karmaşık düzenlemelerinin, görsel alanın üst üste binen ve küçük alt bölgelerinde ışık tespitinden sorumlu hayvan görsel korteksinde yer aldığı bulunmuştur. Ayrıca, bir hesaplamalı model olan Neocognitron, hiyerarşik olarak organize edilmiş görüntü dönüşümleriyle birlikte tanıtılmıştır. Ancak Neocognitron, paylaşılan bir ağırlık gerektirmemesi bakımından CNN'lerden farklıdır.

CNN kavramı, zaman gecikmeli sinir ağlarından (TDNN) esinlenmiştir. Bir TDNN'de ağırlıklar zamansal bir boyutta paylaşılır ve bu da hesaplamada azalmaya yol açar. CNN'lerde evrişim, standart NN'lerde genel matris çarpımının yerini almıştır. Bu şekilde, ağırlıkların sayısı azaltılır ve ağın karmaşıklığı azaltılmış olur. Ayrıca görüntüler, ham girdiler olarak doğrudan ağa aktarılabilir ve böylece standart öğrenme algoritmalarındaki özellik çıkarma prosedüründen kaçınılır. Hiyerarşik katmanların başarılı bir şekilde eğitilmesinden dolayı CNN'lerin gerçekten başarılı ilk derin öğrenme mimarisi olduğu unutulmamalıdır. CNN modelinin bir diğer avantajı da minimum ön işlem gerektirmesidir.

Hesaplama tekniklerinin hızla gelişmesiyle birlikte, GPU hızlandırılmış hesaplama teknikleri, CNN'leri daha verimli bir şekilde eğitmek için kullanılmaktadır.

Günümüzde CNN'ler el yazısı tanıma, yüz algılama, davranış tanıma, konuşma tanıma, tavsiye sistemleri, görüntü sınıflandırması ve doğal dil işlemeye başarıyla uygulanmıştır.

Bir CNN'in öğrenme sürecinde seyrek etkileşim, parametre paylaşımı ve eşdeğer temsil olmak üzere üç faktör kilit rol oynar. Girdi ve çıktı birimleri arasındaki ilişkinin matris çarpımıyla türetildiği geleneksel NN'lerden farklı olarak, CNN'ler, çekirdeklerin girdilerden daha küçük yapıldığı ve tüm görüntü için kullanıldığı seyrek etkileşimle hesaplama yükünü azaltır. Parametre paylaşımının temel fikri, her konumda ayrı bir parametre seti öğrenmek yerine, bunlardan yalnızca bir setini öğrenmeyi gerektirir. Bu da CNN'nin daha iyi bir performans gösterdiği anlamına gelir. Parametre paylaşımı, CNN'ye eşdeğerlik adı verilen bir özellik kazandırmıştır. Bu özellik sayesinde girdiler değiştiğinde, çıktı aynı şekilde değişir. Sonuç olarak, diğer geleneksel NN algoritmalarına kıyasla CNN için daha az parametre gerekir, bu da bellekte azalmaya ve verimlilikte iyileşmeye yol açar.



Şekil 2.10. Standart bir CNN katmanının bileşenleri (Liu ve diğ., 2017)

Bir CNN, evrişim katmanlarından (c-katmanları) ve alt örnekleme katmanlarından (s-katmanları) oluşan çok katmanlı bir sinir ağıdır. C katmanları ve s katmanları dönüşümlü olarak bağlanır ve ağın orta bölümünü oluşturur. Giriş görüntüsü şekilde göründüğü gibi birinci c-katmanında (C1) özellik haritaları oluşturmak için tüm olası ofsetlerde eğitilebilir filtrelerle birleştirilir. Her filtreye bir bağlantı ağırlıkları katmanı dahildir. Normalde, özellik haritasındaki dört piksel bir grup oluşturur. Bir sigmoid

işlevinden geçen bu pikseller, ilk s-katmanında ek özellik haritaları üretir. Bu prosedür devam eder ve böylece sonraki c katmanlarında ve s katmanlarında özellik haritalarını elde edebiliriz. Son olarak, bu piksellerin değerleri rasterleştirilir ve ağın girdisi olarak tek bir vektörde görüntülenir.

Genel olarak c katmanları, her bir nöronun girişi, önceki katmanın yerel alıcı alanına bağlandığında özellikleri çıkarmak için kullanılır. Tüm yerel özellikler çıkarıldıktan sonra, aralarındaki konum ilişkisi çözülebilir. Bir s-katmanı, esasen özellik haritalama katmanıdır. Bu özellik haritalama katmanları ağırlıkları paylaşır ve bir düzlem oluşturur. Ek olarak, ölçek değişmezliği elde etmek için sigmoid işlevi, işlev çekirdeği üzerindeki hafif etkisinden dolayı etkinleştirme işlevi olarak seçilir. Ayrıca, bu modeldeki filtrelerin bir dizi örtüşen alıcı alanı bağlamak ve 2 boyutlu görüntü toplu girişini çıktıdaki tek bir birime dönüştürmek için kullanıldığı da unutulmamalıdır.

CNN için eğitim prosedürü, geri yayılım kullanan standart bir yapay sinir ağına benzer. İlk aşamada bilgi, farklı katmanlar aracılığıyla ileri besleme yönünde yayılır. Her katmana dijital filtreler uygulanarak belirgin özellikler elde edilir. Çıkış değerleri daha sonra hesaplanır. İkinci aşamada, çıktının beklenen ve gerçek değerleri arasındaki hata hesaplanır. Bu hatayı en aza indirmek için, ağırlık matrisine yeniden ayarlama yapılır ve böylece ağın ince ayarı yapılmış olur. Görüntü sınıflandırmadaki diğer standart algoritmalardan farklı olarak, ön işleme CNN'lerde genellikle gerçekleştirilmemektedir. Geleneksel NN'lerde olduğu gibi parametreleri ayarlamak yerine, CNN'lerde filtreleri eğitmemiz gerekir. Dahası, özellik çıkarımında, CNN'ler önceki bilgilerden ve insan müdahalesinden bağımsızdır.

1998'de, LeNets'te alt örnekleme için maksimum havuzlama yöntemi önerilmiştir. Yakındaki çıktıların istatistiklerini özetleyerek, belirli bir konumdaki ağın çıktısını değiştirmek için bir havuzlama işlevi kullanılır. Max-pooling yöntemini kullanarak, dikdörtgen bir mahallede maksimum çıktı elde edebilir. Max-pooling yöntemini kullanarak, dikdörtgen bir komşuluk için maksimum çıktı elde edebilir. Havuzlama prosedürü, gösterimi girdinin çevirilerine göre değişmez hale getirebilir. Sonuç olarak

evrişimli katmanlar arasına maksimum bir havuz katmanı ekleyerek, özellik soyutluğunun artmasıyla uzamsal soyutluk artar.

Havuzlama, görüntü dönüşümlerinde değişmezlik elde etmek için kullanılır. Bu süreç, gürültüye karşı daha iyi bir sağlamlık sağlayacaktır. Çeşitli havuzlama yöntemlerinin performansının, düşük seviyeli özelliklerin çıkarıldığı çözünürlük ve örnek kardinaliteler arasındaki bağlantılar gibi çeşitli faktörlere bağlı olduğuna işaret edilmektedir. Özellikler birbirine çok benzemese bile, konumları yakın olduğu sürece bunları bir araya toplamak mümkündür. Ayrıca, havuzlama aşamasından önce kümeleme gerçekleştirerek daha iyi performansın sağlanabileceği bulunmuştur. Alıcı alanları daha uyarlamalı öğrenerek daha iyi havuzlama performansına ulaşılabileceği gösterilmiştir. Özellikle, aşırı tamlık kavramını kullanarak, artan özellik seçimine dayalı eğitim sürecini hızlandırmak için verimli bir öğrenme algoritması önerilmektedir (Liu ve diğ., 2017).

2.3.5. Tekrarlayan sinir ağları (RNN)

RNN, 1986 yılında David Rumelhart'ın çalışmalarına dayanan bir derin öğrenme sınıfıdır. RNN'ler, sıralı verileri işleme ve bunlardan içgörü elde etme yetenekleriyle ön plana çıkmaktadır. Bu nedenle, video analizi, görüntü altyazıları, doğal dil işleme ve müzik analizinin tümü, tekrarlayan sinir ağlarının yeteneklerine önemli ölçüde bağlıdır. Veri noktaları arasında bağımsızlık sağlayan standart sinir ağlarının aksine, RNN'ler veriler arasındaki sıra ve zaman ilişkilerini etkin bir şekilde yakalamaktadır.

RNN'lerle ilgili en tanımlayıcı özelliklerden biri parametre paylaşımıdır. Parametre paylaşımı olmadan, bir model, bir dizideki her veri noktasını temsil etmek için benzersiz parametreler tahsis eder ve bu nedenle, değişken uzunluk dizileri hakkında çıkarımlar yapamamaktadır. Bu sınırlamanın etkisi, doğal dil işlemede tam olarak gözlemlenebilir. İdeal bir model, farklı iki cümlede yer alan kelimelerin konumuna bakmaksızın, her iki cümlede de tartışılan aynı kelimeler olduğunu kabul eder. Böyle bir senaryodaki geleneksel çok katmanlı bir ağ, cümledeki her konum (kelime) için belirlenen benzersiz ağırlıklara göre dilin bir yorumunu yapacağından başarısız

olacaktır. Bununla birlikte, RNN'ler, zaman adımlarında ağırlıkları paylaştıkları için (yani cümledeki kelimeler) görev için daha uygun olacaktır. Bu şekilde RNN'ler daha doğru cümle kavrayışı sağlar.

RNN'ler genellikle, bitişik düğümleri veya zaman adımlarını bağlayan döngülerin eklenmesi ile geleneksel çok katmanlı ağ mimarisini güçlendirir. Bu döngüler, yakın geçmişten veri noktalarına göre eldeki mevcut veri noktasının özelliklerini değerlendirmek için kullanılan ağın dahili belleğini oluşturur. Ayrıca, çoğu geleneksel ileri beslemeli sinir ağlarının, girdi ve çıktı için bire bir eşlemeyle sınırlı olduğuna dikkat etmek gerekmektedir. Bununla birlikte, RNN'ler, birden çoğa, çoktan çoğa (örneğin, konuşmayı çevirme) ve çoktan bire (örneğin, sesi tanımlama) eşleştirmeleri gerçekleştirebilir. Girişler ile çıkışlar ve kayıp arasındaki eşlemeleri göstermek için bir hesaplama grafiği kullanılır. Grafiğin bir olaylar zinciri halinde açılması, ağ içindeki parametre paylaşımının net bir şekilde görünmesini sağlar.

RNN mimarilerinin önceki sürümleri büyük umut ve çok yönlülük barındırmaktadır. Ancak bazı önemli kusurlarla ilişkilendirilmektedir. Teoride RNN yapıları, bilgileri uzun süreler boyunca hatırlayabilir, ancak pratikte bu her zaman böyle değildir. Vanilya RNN'leri olarak da bilinen geleneksel RNN ağları, her ikisi de birçok zaman adımında biriken yayılma hatalarından kaynaklanan fenomen olan, özellikle kaybolan bir gradyana ve patlayan bir gradyana eğilimlidir. Referanslar arasındaki boşluk küçük kalırsa, RNN bilgi bitlerine atıfta bulunmak için iyi çalışır. RNN'nin zarar görmeye başladığı nokta, referans verilen veriler arasındaki boşluğun artması ve RNN'nin bu veriler arasında her zaman bağlantı kuramamasıdır. Uzun Kısa Süreli Bellek (LSTM) ve Zaman İçinde Kesilmiş Geri Yayılım (TBPTT), bu sorunları gidermek için önerilen geleneksel RNN mimarisinin varyantlarıdır. LSTM mimarisi, kaybolan gradyana karşı koymak için sabit birim ağırlıklara sahip tekrarlayan kenarları kullanmaktadır. TBPTT mimarisi, patlayan gradyanı düzeltmek için hatanın yayılabileceği adım sayısı için bir sınır belirlemektedir.

2.3.6. Uzun kısa vadeli hafıza ağları (LSTM)

LSTM, değerleri rastgele aralıklarla hatırlayan en yaygın RNN mimarisidir. İlk olarak 1997 yılında Hochreiter ve Schmidhuber tarafından tanıtılmıştır. Geleneksel veya vanilya RNN'lerin yeterli olmadığı uzun vadeli bağımlılık sorununu çözerek, zaman serisi verilerine dayalı tahminlerde çok iyi sonuçlar vermektedir. LSTM ayrıca sınıflandırma ve işleme görevleri için çok uygundur. Bu sebeple Google Translate, Apple Siri ve Amazon Alexa uygulamalarında kullanılmaktadır.

Daha önce bahsedildiği gibi, RNN, kaybolan gradyan problemi olarak bilinen fenomene atfedilebilen bir bağlam problemine sahiptir. Kaybolan gradyan sorunu, gradyan inişi, geri yayılımla birlikte bir optimizasyon algoritması olarak kullanıldığında ortaya çıkar. Bağımlılıklar arasındaki boşluk boyutları arttıkça, hata gradyanları katlanarak kaybolur ve bir ağın eğitiminin çok yavaş olmasına veya hatta öğrenememesine neden olabilir.

Stokastik gradyan inişinde gradyan, zincir kuralı ve geri yayılım kullanılarak ağırlıklara göre kayıp fonksiyonunun kısmi türevine göre hesaplanır. Stokastik gradyan inişi, gradyan inişinin optimize edilmiş bir şeklidir. Gradyan inişi, ağdaki tüm eğitim örneklerinin kaybını optimize eder. Her eğitim örneği için kaybı optimize ettiği için yoğun bir işlemdir. Bir milyon eğitim örneği varsa, gradyan bir milyon kez hesaplanır. Stokastik gradyan inişini kullanarak, ağın parametrelerini optimize etmek için yalnızca bir eğitim örneği kullanılır ve ağı eğitme süresini önemli ölçüde azaltır. Ayrıca, başka bir yöntem olan mini parti gradyan inişi de maliyeti optimize etmek için kullanılabilir. Minibatch gradyan inişi, ağdaki parametreleri güncellemek için n sayıda örnek kullanır. Stokastik gradyan inişi, gradyan inişine kıyasla maksimum optimizasyona ulaşmayacak olsa da genel olarak doğruluk optimizasyona yeterince yakındır ve büyük bir veri setine sahip bir ağı eğitirken büyük ölçüde faydalıdır.

Ağdaki parametrelerde yapılan güncellemeler zincir kuralı kullanılarak uygulanır. Zincir kuralıyla, gradyanlar, zinciri destekleyecek şekilde ağın önüne doğru yayılırken,

her düğümden alınan ağırlığa göre maliyet fonksiyonunun türevinin toplamı olarak hesaplanır. Gradyan daha sonra fonksiyonların ağırlıklarını önceki düğümlerden güncellemek için kullanılır. Katmanlar arasındaki zaman bağımlılığı arttıkça, ağırlığa ilişkin "gözden kaybolan" küçük düzeltmeler nedeniyle ağırlıklar yalnızca marjinal olarak güncellenir.

Birden küçük bir değere sahip hesaplanan gradyanı düşünüldüğünde; geri yayılımla ağırlıklar geriye doğru ayarlanır ve gradyanlar birden küçük çok miktarda sayı içerdiğinde, gradyan, ağın gerisine doğru ilerledikçe üssel olarak küçük hale gelir. Sonuç olarak öğrenme oranları ile çarpıldıklarında daha da küçülür. Ağırlıklar, eğitim için bir ağ kurarken başlangıçta rastgele bir sayıya ayarlandığından, başlangıçta daha büyük kayıplara sahip olma eğilimindedirler ve ağırlıklar yalnızca marjinal olarak ayarlandığından, kaybolan gradyan problemi sorununu birleştirir. LSTM daha sonra hücre yapıları içinde farklı kapılar kullanarak bu sorunu çözer.

Klasik ağlardan farklı olarak, LSTM yalnızca mevcut durumdan bilgi türetmekle kalmaz, aynı zamanda önceki durumlardan da bilgi alabilir. LSTM'nin kritik bileşenleri bellek hücresi ve kapılarıdır. LSTM'nin farklı varyasyonları vardır, ancak hepsi ağırlıklı olarak unutma kapısı, giriş kapısı ve çıkış kapısı olarak bilinen üç kapı içerir. Bellek hücresinin içeriği giriş ve unutma kapıları tarafından modüle edilir. Bu kapıların her ikisinin de kapalı olduğunu varsayarsak, hafıza hücresinin içeriği bir zaman adımı ve sonraki arasında değişmeden kalacaktır. Geçiş yapısı, bilginin birçok zaman adımında tutulmasına izin verir ve sonuç olarak gradyanların birçok zaman adımı boyunca akmasına izin verir. Bu, LSTM modelinin, çoğu Tekrarlayan Sinir Ağı modelinde ortaya çıkan kaybolan gradyan probleminin üstesinden gelmesine izin verir. Bir LSTM ağının katlanmamış grafiği, bir katmandan diğerine geçen verilerin, doğrusal etkileşimler kullanılarak giriş ve unutma kapıları üzerinden her katmandan geçerken hafifçe değiştirildiği bir konveyör bandı olarak düşünülebilir.

Unutma kapısı, bilgilerin hücre durumundan çıkarılmasından sorumludur ve amacı, hangi bilgilerin artık yararlı olmadığını ve unutulabileceğini belirlemektir. Önceki hafıza hücresinden aldığı “gizli durum” ve belirli zaman adımındaki mevcut hücre

durumu olarak bilinen “güncel girdi” olmak üzere iki girdi almaktadır. Girişler ağırlık matrisleriyle çarpılır ve bir sapma eklenir. Bundan sonra bir sigmoid işlevi uygulanır; sigmoid işlevi, hangi değerlerin korunacağına ve hangilerinin atılacağına karar vermekten sorumludur. Fonksiyon, 0 ile 1 arasında değerlere sahip bir vektör çıkarır. 0, unutma geçidinin bilgiyi tamamen unutmak istediğini belirtirken, 1, unutma kapısının tüm bilgi parçasını hatırlamak istediğini gösterir.

Giriş kapısı 2 adımlı bir süreci içerir ve hücre durumuna hangi yeni bilgilerin ekleneceğine karar vermekten sorumludur. Unutma kapısına benzer şekilde, h (t − 1) ve x (t) 'ye bir sigmoid işlevi uygulanır. Bir hiperbolik tanjant işlevi, -1 ile 1 arasında değişen tüm olası değerlerin bir vektörünü oluşturur. Bu vektör, hücre durumuna eklenebilecek aday değerleri gösterir.

Çıkış kapısı, 3 adımlı bir işlemde çıktı olarak hücre durumundan yararlı bilgileri seçer. İlk adımda, hücre durumuna hiperbolik bir teğet işlevi uygulanır ve -1'den 1'e ölçeklenmiş değerlere sahip bir vektör oluşturulur. İkinci adımda sigmoid işlevini ve önceki gizli durumu kullanarak bir düzenleyici filtre oluşturmak için girdi olarak, h(t−1) ve x (t) kullanılır. Son adımda, adım 2'deki düzenleyici filtre adım 1'deki vektör ile çarpılarak bir sonraki hücreye bir çıktı ve gizli durum üretilir. LSTM mimarisini kullanarak, ağ herhangi bir uzun vadeli bağımlılığı en aza indirebilmek ve 1000 adımı aşan veri referanslarındaki boşlukları kapatabilmek mümkündür (Witold Pedrycz, Shyi-Ming Chen).

2.3.7. Derin ileri beslemeli ağlar

Derin ileri beslemeli sinir ağı, yalnızca ileriye doğru hareket eden düğümlerin birbirine bağlı olduğu en temel derin öğrenme mimarisidir. Temel olarak, çok katmanlı bir sinir ağı birden fazla sayıda gizli katman içerdiğinde, buna derin sinir ağı denmektedir. Birden çok gizli katman, karmaşık doğrusal olmayan ilişkiyi sığ mimariye kıyasla daha verimli bir şekilde modellemeye yardımcı olur. Mimarinin basitliği ve bu modeldeki eğitimin kolaylığı nedeniyle, mühendisliğin hemen hemen tüm alanlarında araştırmacılar ve uygulayıcılar arasında her zaman popüler bir mimaridir. Gradyan iniş

kullanan geri yayılım, bu modeli eğitmek için kullanılan en yaygın öğrenme algoritmasıdır. Algoritma önce ağırlıkları rasgele başlatır ve ardından ağırlıklar, gradyan inişi kullanarak hatayı en aza indirecek şekilde ayarlanır. Öğrenme prosedürü, arka arkaya birden çok ileri ve geri geçişi içerir. İleri geçişte, girdiyi, doğrusal olmayan çoklu gizli katmanlardan çıktıya doğru iletilir ve sonuçta hesaplanan çıktıyı karşılık gelen girdinin gerçek çıktısı ile karşılaştırılır. Geri geçişte, ağ parametrelerine göre hata türevleri, çıktıdaki hatayı en aza indirmek için ağırlıkları ayarlamak üzere geri yayılır. Süreç, model tahmininde istenen bir iyileştirmeyi elde edene kadar birçok kez devam eder.

Derin bir sinir ağı basit bir şekilde eğitilirse, aşırı uyum, yerel minimumda tuzağa düşme ve kaybolan gradyan sorunu gibi birçok sorun ortaya çıkabilir. Bu tür rahatsız edici sorunlar, 1990'ların sonlarında sinir ağları üzerine yapılan araştırmaların yavaşlamasına katkıda bulunmuştur. Bununla birlikte, on yıl sonra derin sinir ağlarında denetimsiz ön eğitim yaklaşımlarının ortaya çıkmasıyla, sinir ağı araştırması görüntü işleme ve konuşma gibi karmaşık görevler için kullanılmak üzere yeniden canlandırılmıştır. Son zamanlarda L1 ve L2 düzenlenmesi, bırakma, toplu normalleştirme, iyi bir ağırlık başlatma teknikleri koleksiyonu ve iyi bir dizi etkinleştirme işlevi gibi birçok teknik, derin sinir ağlarının eğitiminde uzun süredir devam eden sorunlarla mücadele etmek için çeşitli derecelerde başarı ile tanıtılmıştır (Sengupta ve diğ., 2020).

2.3.8. ARIMA (Autoregressive Integrated Moving Average Model)

Otoregresif Entegre Hareketli Ortalama Modeli (ARIMA), Otoregresif (AR) süreci ve Hareketli Ortalama (MA) süreçlerini birleştiren ve zaman serilerinin bileşik bir modelini oluşturan genelleştirilmiş bir Otoregresif Hareketli Ortalama (ARMA) modelidir. Model şu şekilde açıklanabilir:

1. AR: Otomatik regresyon. Bir gözlem ile birkaç gecikmeli gözlem arasındaki bağımlılıkları kullanan bir regresyon modelidir.

2. I: Entegre. Gözlemlerin farklı zamanlardaki farklılıklarını ölçerek zaman serilerini durağan hale getirmektir.
3. MA: Hareketli Ortalama. Gecikmiş gözlemler için hareketli bir ortalama model kullanıldığında, gözlemler ve kalan hata terimleri arasındaki bağımlılığı hesaba katan bir yaklaşımdır.

Box ve Jenkins tahmini olarak da bilinen ARIMA tahmini, "bütünleştirme" adımı nedeniyle durağan olmayan zaman serisi verileriyle başa çıkma yeteneğine sahiptir. Aslında, "entegre etme" bileşeni, durağan olmayan bir zaman serisini durağan hale getirmek için zaman serilerinin farklılaştırılmasını içerir.

Mevsimsel zaman serisi verileriyle, kısa vadeli mevsimsel olmayan bileşenlerin modele katkıda bulunması muhtemeldir. Bu nedenle, çarpımsal bir modelde hem mevsimsel olmayan hem de mevsimsel faktörleri içeren mevsimsel ARIMA modelini tahmin etmek gerekir. Mevsimsel bir ARIMA modelinin genel formu, ARIMA (p, d, q) × (P, D, Q) S olarak belirtilir, burada p mevsimsel olmayan AR sırasıdır, d mevsimsel olmayan farktır, q ise Mevsimsel olmayan MA siparişi, P mevsimsel AR sırası, D mevsimsel farklılaşmadır, Q mevsimsel MA düzenidir ve S sırasıyla tekrarlayan mevsimsel modelin zaman aralığıdır.

Mevsimsel ARIMA modelini tahmin etmenin en önemli adımı (p, d, q) ve (P, D, Q) değerlerini belirlemektir. Verinin zaman grafiğine bağlı olarak, varyansın zamanla büyüdüğü düşünülürse, varyans dengeleyici dönüşümler ve farklılaşmayı kullanmalıdır. Daha sonra, bir gecikme p ile ayrılmış bir zaman serisindeki gözlemler arasındaki doğrusal bağımlılık miktarını ölçmek için otokorelasyon fonksiyonunu kullanmak ve kaç otoregresif terim q'nun gerekli olduğunu belirlemek için kısmi otokorelasyon fonksiyonu ve ters otokorelasyon fonksiyonu, otoregresif sıra p'nin ön değerlerini, d farklılaşma sırasını, hareketli ortalama düzeni q ve bunlara karşılık gelen mevsimsel parametreleri P, D ve Q'yu belirleyebilir. D parametresi, durağan olmayan zaman serilerinden durağan zaman serilerine değişen frekans farkı sırasıdır (Namini ve diğ., 2018).

2.4. İlgili Çalışmalar

Yapay zeka teknikleri kullanılarak altın fiyatlarının tahminlerini yapan çalışmalarla sıklıkla karşılaşılmaktadır. Bu çalışmalar şu şekilde sıralanabilir; Benli ve Yıldız (2014); Chamzini, (2012); Yüksel ve Akkoç, (2016); Kocatepe ve Yıldız; (2016); Lemin, (2016); Çelik ve Başarır, (2017) ve Zhu ve Wang, (2017). Görüldüğü üzere son zamanlarda yapılan çalışmaların çoğu, diğer değerli metallerin tahminlerini göz ardı ederek altın fiyatlarının tahminine odaklanmaktadır. Bu nedenle bu çalışma, gümüş fiyatlarının LSTM ve ARIMA yöntemleri kullanılarak tahmin edilmesini içermektedir. Gümüş fiyatlarını kullanan modelin tahmin gücü, portföyünü geliştirmeyi ve daha karlı yatırımlar yapmayı hedefleyen yatırımcılar için daha iyi sonuçlar vermek üzere yükselmektedir. Bu çalışmada altın fiyatlarını tahmin etmeden ziyade gümüş fiyatlarının yapay zeka algoritamlarıyla tahmini yapılmıştır.

Çalışmamızda öncelikle 01.01.2018 ile 12.04.2021 tarihleri arasında gümüş/ons parite verileri elde edilmiştir. Bu veriler test için kullanıldı. Elde edilen veriler normalleştirildi. Bu veriler uygulamada kullanılan LSTM ve ARIMA algoritmalarına öğretildi. Daha sonra 13.04.2021 ve 21.04.2021 tarihleri için iki algoritma ile tahmin değerleri üretilmiştir. Daha sonra bu tahminlerin performansı MAE ve RMSE performans kriterleri kullanılarak değerlendirilmiştir.

BÖLÜM 3. MATERYAL VE YÖNTEM

Bu çalışmada gümüş/ons paritesinin belirli zaman aralığı için günlük açılış, kapanış, en yüksek, en düşük ve değer gibi verileri elde edilmiştir. Öncelikle veri dizisi seçilir ve sisteme yüklenir ve sistemin hem öğrenmesi hem de test etmesi sağlanır.

1.1.2013 ile 21.04.2021 tarihleri arasındaki gümüş/ons paritesinin verileri elde edilmiştir. 01.01.2015 ile 11.04.2021 tarihleri arası kapanış değerleri algoritmalara eğitim verisi olarak verilmiştir. 12.04.2021 ile 21.04.2021 tarihleri arası kapanış değerleri test verisi olarak kullanılmıştır. Veri setinin tamamına ulaşmak için aşağıdaki linki kullanılabilir:

https://raw.githubusercontent.com/SnnUntz/Data/main/XAGUSD_D1.csv

Tarih	Açılış	Yüksek	Düşük	Kapanış	Hacim
1.01.2015	15.678	15.898	15.654	15.805	60
2.01.2015	15.805	16.057	15.514	15.723	1373
4.01.2015	15.767	15.780	15.623	15.733	61
5.01.2015	15.732	16.249	15.712	16.141	1441
6.01.2015	16.140	16.696	16.099	16.439	1433
6.04.2021	24.802	25.288	24.753	25.155	1598
7.04.2021	25.154	25.266	24.843	25.145	1596
8.04.2021	25.142	25.608	24.951	25.389	1641
9.04.2021	25.394	25.492	24.971	25.191	1453

Tablo 3.1. Kullanılan Gümüş/Ons verileri örneği

Tüm verilerin sadece 9 günlük kısmı Tablo 3.1.'de gösterilmektedir.

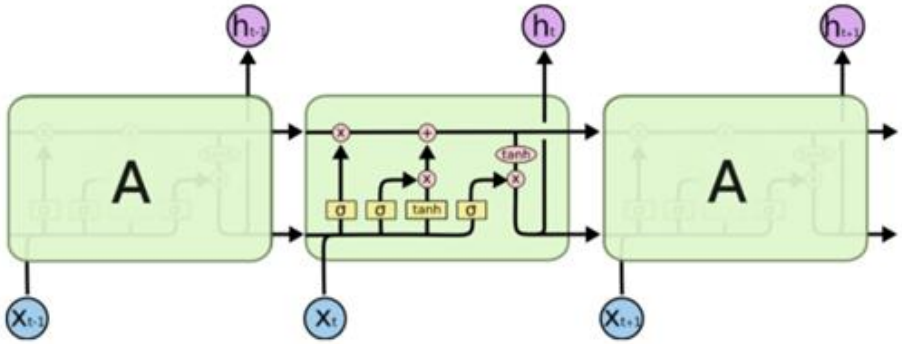
Veriler indirildikten sonra LSTM ve ARIMA algoritmaları ile derin öğrenmede kullanıma hazırlanır. Hazırlık aşamasında veriler 0 ile 1 arasında olacak şekilde normalize edilmiştir. Ön işlemlerden sonra tahmin için kullanılacak algoritmalar seçilmiştir.

Hem uzun vadeli hem de kısa vadeli değerleri hatırlama yeteneği ile LSTM modelleri, zaman serilerinin işlenmesi için finansal olarak faydalı olduğunu kanıtlamış ve böylece zaman serisi analizi için tercih edilen Derin Öğrenme aracı haline gelmiştir (Sadefo Kamdem ve diğerleri, 2020). Bu nedenle çalışmada LSTM algoritması kullanılmıştır.

Otoregresif Entegre Hareketli Ortalama (ARIMA) ve varyantları, literatürde hisse senedi fiyat serilerini tahmin etmek için en çok kullanılmıştır (Aamir ve Shabri, 2018). Bunlar, ARIMA modelinin kısa vadede güçlü bir tahmin potansiyeline sahip olduğunu ve mevcut hisse senedi fiyat tahmin araçlarıyla olumlu bir şekilde rekabet etme yeteneğine sahip olduğunu göstermiştir (Ariyo ve diğerleri, 2014). Bu nedenle çalışmamızda kullanılan ikinci algoritma ARIMA olarak seçilmiştir.

3.1. LSTM Mimarisi Kullanılarak Geliştirilen Tahmin Modeli

Uzun ve Kısa Süreli Bellek, RNN'nin bir varyasyonudur ve uzun menzilli zamansal bağımlılıklarla ilgili problemleri öğrendiği bilinmektedir, bu nedenle RNN'ler bazen MT ağlarında LSTM'lerle değiştirilir (Hochreiter ve Schmidhuber, 1997). LSTM'ler de bu zincir benzeri yapıya sahiptir, ancak yinelenen modülün yapısı RNN'den farklıdır. Bir modülde tek bir sinir ağı katmanı yerine dört katman vardır. Bu katmanlar, aynı modüller içinde ve diğer öğrenme modülleri ile etkileşime girer (Saini ve Sahula, 2018). LSTM modülünün tipik bir yapısı Şekil 3.1.'de gösterilmektedir.



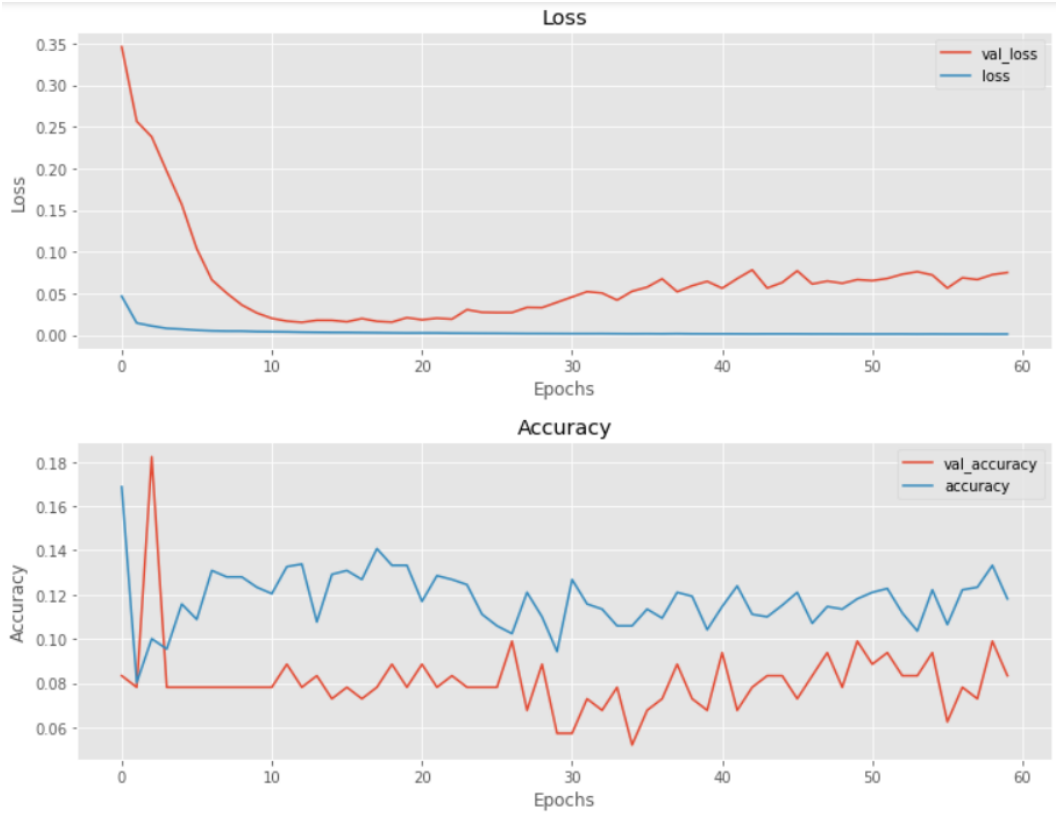
Şekil 3.1. Bir LSTM'deki yinelenen modül, dört etkileşimli katman içerir (Saini ve Sahula, 2018)

Phyton programı Keras kütüphanesi kullanılarak LSTM algoritması uygulanmıştır. Kapanış değerleri esas alınmış ve 1.1.2015 ile 11.4.2021 tarihleri arası veriler yüklenmiştir. Algoritma 10 defa çalıştırılmış ve elde edilen günlük kapanış değerlerinin ortalaması alınmıştır. Bu şekilde hangi algoritmanın ortalamada daha tutarlı sonuçlar verdiği tespit edilmiştir.

Normalleştirilmiş grafik aşağıdaki gibidir. Verilerin 2015 tarihinden itibaren alınmasının sebebi hem programın hızlı çalışmasını sağlamak hem de günlük kapanış değerleri için altı yıllık verinin yeterli görülmesidir. Veri aralığı daha fazla alınabileceği gibi daha az alan çalışmalar da görülmüştür. Örneğin Hiransha M. ve arkadaşlarının yapmış olduğu çalışmada veri aralığı 2011 ocak – 2016 aralık olarak alınmıştır. (Hiransha M. ve ark.)

Şekil 3.2. LSTM Mimarisi İçin Normalleştirilmiş Gümüş/Ons Grafiği

Eğitim verisi algoritmanın eğitilmesi için kullanılmış ve geleceğe yönelik 10 günlük tahmin yapması için programlanmıştır. Algoritmada loss değerlerinin birbirine ne kadar yakın olursa o kadar iyi sonuç vereceği belirtilmiştir. Uygulanan modelde loss ve accuracy değerleri aşağıdaki şekilde belirtilmiştir.

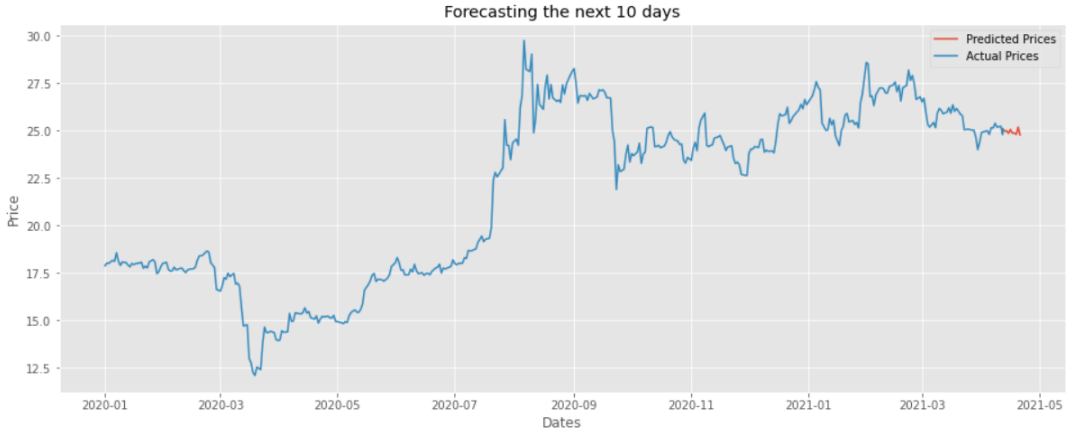


Şekil 3.3. LSTM algoritması için loss ve accuracy değerleri

Verilerde 12.04.2021 tarihinden itibaren tahmin değerleri üretilmiştir. Algoritma 10 defa çalıştırılarak elde edilen sonuçların ortalaması alınmıştır. Bu şekilde her gün için ortalama günlük kapanış değerleri elde edilmiştir. Algoritmanın ürettiği günlük kapanış için ortalama tahmin değerleri Şekil 3.2.'teki gibidir.

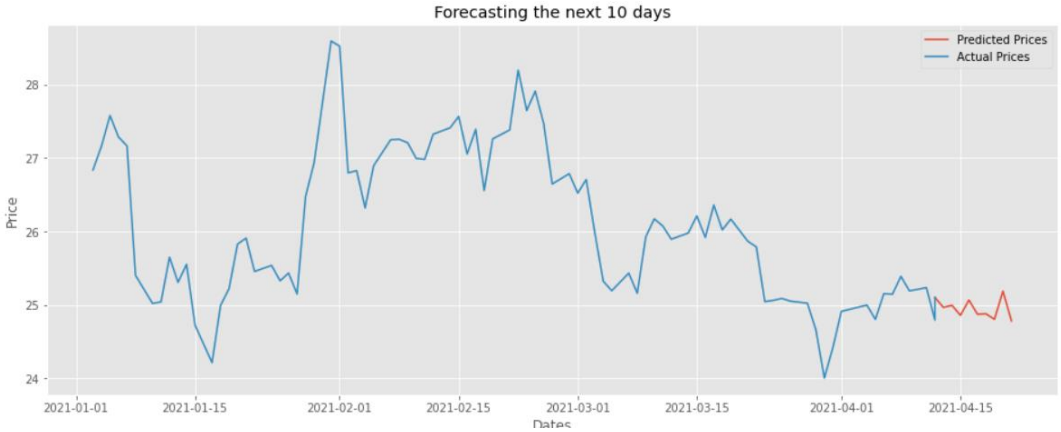
Tarih	Kapanış değeri
2021-04-12	26.128
2021-04-13	26.398
2021-04-14	26.436
2021-04-15	26.458
2021-04-16	26.506
2021-04-17	26.594
2021-04-18	26.408
2021-04-19	26.433
2021-04-20	26.290
2021-04-21	26.474

Tablo 3.2. LSTM algoritması ile yapılan tahminde ortalama günlük kapanış değerleri



Şekil 3.4. LSTM algoritması ile 10 günlük tahmin modeli grafiği örneği

Elde edilen grafik uzun bir zaman dilimini ifade ettiği için Şekil 3.5.'da daha yakından değerler gösterilmiştir.



Şekil 3.5. LSTM algoritması detay grafiği örneği

3.2. ARIMA Mimarisi Kullanılarak Geliştirilen Tahmin Modeli

Box-Jenkins yöntemini kullanan zaman serisi modeli, Box ve Jenkins (1970) tarafından önerilmiştir. Bu yaklaşım, basitliği ve iyi sonuçları nedeniyle literatürde yaygın olarak kullanılmaktadır. Genellikle bu yönteme ARIMA yöntemi denir. ARIMA yöntemi, tahmin serilerinin açıklayıcı bağımsız değişkenler içermemesi nedeniyle diğer yöntemlerden oldukça farklıdır. Bir ARIMA modeli oluştururken,

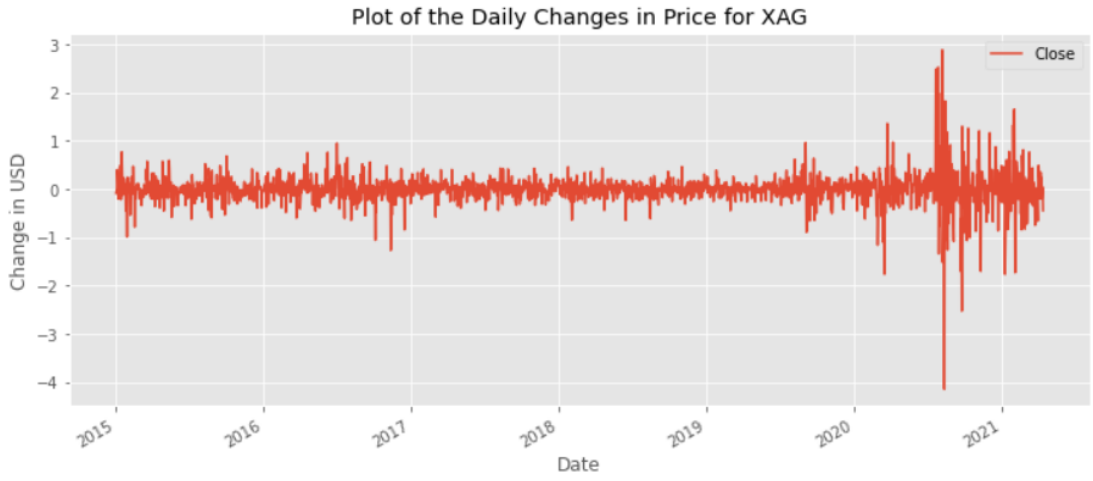
mevcut seri durağan olmalıdır. Seri durağan değilse yöntem kullanılmadan önce serilerin farkı alınmalıdır (Gahirwal, 2013).

Döviz kuru zaman serileri için en iyi ARIMA modelini oluşturabilmek için etkin bir model için otoregresif (p) ve hareketli ortalama (q) parametrelerinin tanımlanması gerekmektedir. Tümleşik terimi (d) 1. sıra olarak koruyarak çeşitli otoregresif (p) ve hareketli ortalama (q) terimleri için Bayesian Information Criterion'a (BIC) dayalı en iyi modeli belirlemeye karar verdik (Babu ve Reddy, 2015). Bu çalışmada ARIMA parametreleri algoritma tarafından belirlenmiştir. Tüm mümkün p,q,d değerleri için sistem çalıştırılmış ve en düşük AIC elde eden kombinasyon seçilmiştir. AIC (Akaike ölçütü) belirli bir veri kümesi için kaliteli bir istatistiksel göreceli model ölçüsüdür. Yani, veri modelleri koleksiyonu verildiğinde, AIC her model kalitesini, diğer modellerin her birini göreceli olarak tahmin eder.

LSTM mimarisi için kullanılmış veriler ARIMA mimarisi için de kullanılmıştır. ARIMA algoritması Phyton programının Keras kütüphanesi kullanılarak uygulanmıştır. 01.01.2015 tarihinden itibaren olmak üzere 12.04.2021 tarihine kadar olan veriler kullanılmıştır. Şekil 3.6.'de kapanış değerlerine göre oluşturulan gümüş/ons grafiği yer almıştır.

Şekil 3.6. ARIMA mimarisi için gümüş/ons grafiği

Gümüş/ons grafiğinde fiyattaki günlük değişimi Şekil 3.7.'deki gibidir.



Şekil 3.7. ARIMA mimarisi için gümüş/ons günlük fiyat değişim grafiği

Algoritma eğitildikten sonra elde edilen test değerleri üretilmiştir. ARIMA algoritması ile elde edilen 10 günlük tahmin değerleri Tablo 3.3.'teki gibidir. Tablo, 10 defa çalıştırılan algoritmanın sonuçlarından günlük ortalama değerler hesaplanarak oluşturulmuştur.

Tarih	Kapanış değeri
2021-04-12	25.121
2021-04-13	25.829
2021-04-14	25.815
2021-04-15	25.840
2021-04-16	25.834
2021-04-17	25.852
2021-04-18	25.851
2021-04-19	25.866
2021-04-20	25.867
2021-04-21	25.880

Tablo 3.3. ARIMA algoritması ile yapılan tahminde ortalama günlük kapanış değerleri

BÖLÜM 4. ARAŞTIRMA BULGULARI

Finansal tahmin çalışmaları için pek çok mimari ve algoritma kullanılmakta ve günden güne gelişmektedir. Yapılan literatür araştırmalarına istinaden günümüzde en iyi sonuç veren mimarilerden ikisi olan LSTM ve ARIMA mimarileri bu çalışmamızın ana konusu olmuş ve gümüş/ons grafiği üzerine eğitim ve test verilerini işleyerek 10 günlük tahmin yapması istenmiştir. Bu kapsamda 01.01.2015 yılında itibaren gümüş/ons grafiği için günlük açılış, kapanış, düşük, yüksek ve hacim değerleri elde edilerek her iki mimari için yazılan algoritmalara yedirilmiş ve istenilen grafikler oluşturulmuştur. Burada belirtilmelidir ki hem finansal piyasaların manipülasyona açık olması hem de hiçbir yapay zeka mimarisinin kesin doğru sonuçlar verememesi sebebiyle yapılan çalışma sonuçları oluşan gerçek kapanış değerleri ile aynı olmayabilir. Ancak yazılan algoritmalarda en tutarlı sonuçların 10 günlük gelecek tahmini için olduğu, daha fazla gün için tahmin yapılmak istendiğinde tutarlılığın düştüğü bilinmektedir.

12.04.2021 - 21.04.2021 tarihleri için LSTM ve ARIMA algoritmaları ile tahmin değerleri üretilmiştir. Üretilen tahmini değerler ile gerçek değerler arasındaki ilişki incelenmiştir. Gerçek değerlere yakın değerler üreten algoritmanın daha iyi performans gösterdiği bilinmektedir. Performans kriteri olarak birçok kriter kullanılabilir.

4.1. Yöntemlerin Performans Ölçütlerine Göre Karşılaştırılması

Bilimsel anlamda tahmin ve tahmine dayalı analitik, bir olasılıkla karakterize edilen ve istatistiksel veri analizine ve mevcut gelişmeye dayanan bir kişinin davranışını veya eğilimi tahmin etmek anlamına gelmektedir. Buna karşılık, tahmin, bir (toplu) tarihsel veri analizine, mevcut duruma ve bazen tahmine dayalı analitiklere dayanan değer veya belirli bir zaman noktasında bir olayın meydana gelmesi demektir.

İsim	Kısaltması
Ortalama Kare Hata	MSE
Kök Ortalama Kare Hata	RMSE
Ortalama Tahmin Hatası	MFE
Ortalama Mutlak Sapma	MAD
Ortalama Mutlak Yüzde Hatası	MAPE
Ağırlıklı Ortalama Mutlak Yüzde Hatası	wMAPE

Tablo 4.1. Tahmin doğruluğu ölçütleri

Ortalama günlük MAE, maksimum günlük MAPE ve günlük RMSE değerleri tüm yapılar için normalizeli olarak karşılaştırılmıştır.

Tahmin hataları MAPE olarak hesaplanmıştır. Oluşturulan her bir yapı için ortalama günlük MAPE ve günlük maksimum MAPE hesaplanmış ve karşılaştırılmıştır.

4.1.1. MSE (Mean Squared Error - Ortalama Kare Hata)

Basitçe, ortalama kare hata bir regresyon eğrisinin bir dizi noktaya ne kadar yakın olduğunu söyler. MSE, bir makine öğrenmesi modelinin, tahminleyicinin performansını ölçer, her zaman pozitif değerlidir ve MSE değeri sıfıra yakın olan tahminleyicilerin daha iyi bir performans gösterdiği söylenebilir.

$$\text{MSE} = \frac{1}{n} \sum_{j=1}^{n} e_j^2 \tag{4.1}$$

$$e_j^2 = (Gerçek\ değer_j - Tahmin\ değer_j)^2 \tag{4.2}$$

4.1.2. RMSE (Root Mean Square Error - Kök Ortalama Kare Hata)

Bir makine öğrenmesi modelinin, tahminleyicinin tahmin ettiği değerler ile gerçek değerleri arasındaki uzaklığın bulunmasında sıklıkla kullanılan, hatanın büyüklğünü ölçen kuadratik bir metriktir. RMSE tahmin hatalarının (kalıntıların) standart sapmasıdır. Yani, kalıntılar, regresyon hattının veri noktalarından ne kadar uzakta olduğunun bir ölçüsüdür; RMSE ise bu kalıntıların ne kadar yayıldığının bir ölçüsüdür.

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{n} e_j^2}{n}} \tag{4.3}$$

	MAE	RMSE
LSTM	797	922
ARIMA	283	388

Tablo 4.2. LSTM ve ARIMA algoritmaları için MAE ve RMSE değerleri

LSTM algoritmasının MAE hata değeri 792 ve RSME hata değeri 922 olarak belirlenmiştir. ARIMA algoritmassının MAE hata değeri 283 ve RSME hata değeri 388 olarak tespit edilmiştir. MAE ve RMSE performans kriterlerinden de anlaşılacağı gibi ARIMA, LSTM algoritmasından daha iyi performans göstermektedir. ARIMA algoritması, RMSE ve MAE açısından LSTM algoritmasından sırasıyla %137 ve %64 daha iyidir.

Gün	**ARIMA**	**MAE**	**RMSE**	**LSTM**	**MAE**	**RMSE**	**GERÇEK**
1	25.132	337	113569	26.128	1.333	1776889	24.795
2	25.356	11	121	26.398	1.053	1108809	25.345
3	25.386	25	625	26.436	1.025	1050625	25.411
4	25.455	380	144400	26.458	623	388129	25.835
5	25.623	321	103041	26.506	562	315844	25.944
6	25.758	892	795664	26.594	1.728	2985984	24.866
7	25.780	134	17956	26.408	494	244036	25.914
8	25.876	41	1681	26.433	598	357604	25.835
9	25.894	128	16384	26.290	524	274576	25.766
10	25.940	562	315844	26.474	28	784	26.502

Tablo 4.3. 10 günlük ortalama tahmin değerleri ile gerçekleşen gümüş/ons paritesi değerleri

BÖLÜM 5. TARTIŞMA VE SONUÇ

Günümüzde hem finansal piyasalar hem de yapay zeka günden güne önem kazanmakta ve dünyanın ekonomik durumundan ve teknolojik gelişmelerden her ikisi de etkilenerek bir ortak bir noktada buluşmaktadır. Derin öğrenme yöntemleri kullanılarak finansal grafikler için tahmin modelleri oluşturulmakta ve hatta al-sat botları pek çok indikatöre göre programlanarak kullanıcılarına büyük kazançlar sağlamaktadır. Hem bireysel olarak hem de ülke geleceği adına finansal olarak kar edebilmek için teknoloji ve yapay zeka aktif olarak kullanılmalıdır. Yapmış olduğumuz çalışmada ise altın ile beraber en çok rağbet gören emtia olan gümüş üzerinde bir tahmin modeli oluşturarak grafiğin hangi yöne gideceği tahmin edilmek istenmiştir. Bu sebeple yapılan literatür araştırması sonucunda pek çok derin öğrenme algoritması olduğu görülmüştür. Ancak bazı algoritmalar özellikle bazı problemler için daha uygun olarak değerlendirilmiştir.

Biz de bu çalışmamızda sürekli zaman serileri için kullanılan ve en iyi sonuçlar verdiğini gördüğümüz iki derin öğrenme ve zaman serisi mimarisini kullanarak gümüş/ons grafiği üzerinden bir tahmin modeli oluşturduk. ARIMA ve LSTM mimarilerini kullandığımız çalışmada her iki mimari de farklı sonuçlar verdi. Yazılan kodlamada değişiklikler yapılarak farklı sonuçlar da elde etmek mümkündür. Derin öğrenme katmanlarını değiştirerek veya farklı zaman dilimleri için tahmin üretmek istediğimizde yine sonuçlar değişecektir. Hiçbir yapay zeka veya derin öğrenme algoritması geleceği yüzde yüz bilemeyeceği için yüksek tutarlılıkta tahminler yapmaya çalışmak derin öğrenme ile finansal tahmin çalışmalarının esas konusu olmuştur.

Çalışmamızda LSTM ve ARIMA algoritmalarından elde edilen tahmin sonuçlarını karşılaştırmak için MAE ve RMSE performans kriterleri kullanılmıştır. Performans

kriterlerine göre ARIMA, LSTM algoritmasından daha iyi performans göstermiştir. ARIMA algoritması, RMSE ve MAE açısından LSTM algoritmasından daha iyidir.

Yapmış olduğumuz çalışmada ARIMA mimarisi yön ve aralık olarak doğru sonuçlar vermiştir. LSTM mimarisi ile yaptığımız tahminde gelecek 10 günlük pattern benzemesine rağmen yükseliş yönünde bir tahmin oluşmadığı için daha az doğruluk payına sahiptir. Ancak çalışmamız ana fikir olarak, uygun derin öğrenme mimarileri ile geleceğe yönelik finansal tahminler yapılmasının mümkün olduğunu göstermektedir. İlerleyen dönemde daha gelişmiş programlama dilleri veya algoritmalar kullanılarak daha yüksek tutarlılıkta finansal tahminler yapmak mümkün olacaktır.

KAYNAKLAR

Aamir M., Shabri A. (2018) Improving crude oil price forecasting accuracy via decomposition and ensemble model by reconstructing the stochastic and deterministic influences. Adv Sci Lett. 2018;24(6):4337–4342.

Ariyo A.A., Adewumi A.O., Ayo C.K. (2014) Stock price prediction using the ARIMA model; UKSim-AMSS 16th international conference on computer modelling and simulation. IEEE; 2014. pp. 106–112.

Babu AS, Reddy SK (2015) Exchange Rate Forecasting using ARIMA, Neural Network and Fuzzy Neuron. Journal of Stock Forex Trading 4, 1-5 155.

Bell, J. 2014. Machine Learning : Hands-On for Developers and Technical Professionals. Somerset, UNITED STATES, John Wiley & Sons, Incorporated.

Buchanan, Bruce G. Winter 2005. "A (Very) Brief History of Artificial Intelligence"

Benli, Y. K., & Yıldız, A. (2014). Altın Fiyatının Zaman Serisi Yöntemleri ve Yapay Sinir Ağları ile Öngörüsü. Dumlupınar Üniversitesi Sosyal Bilimler Dergisi, 42, 213-224.

Chamzini, A., Yakhchali, H. S. Y., Volungevičienė, D., & Zavadskas, Z. (2012). Forecasting Gold Price Changes by Using Adaptive Network Fuzzy Inference System, Journal of Business Economics and Management, 13(5): 994–1010.

Çayıroğlu, İ., 2013. İleri Algoritma Analizi, http://www.ibrahimcayiroglu.com/ Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi 10. Hafta, Parcacik Suru Algoritmasi.pdf. Erişim Tarihi: 03.04.2021.

Çelik, U., & Başarır, Ç. (2017). Yapay Sinir Ağları İle Kıymetli Maden Fiyatlarının RapidMiner İle Tahmin Edilmesi. Alphanumeric Journal, 5(1), 45-54.

Doğan, M. (2010). Neurological Diagnosis System Based On Neural Networks.

Jakhar D, Kaur I. Artificial intelligence, machine learning and deep learning: definitions and differences. Clin Exp Dermatol. 2020 Jan;45(1):131-132.

Gahirwal, M. (2013). Inter Time series sales forecasting. arXiv preprint

Hiransha, M., Gopalakrishnan, E.A., Menon, V.K., & Soman, K.P. (2018). NSE Stock Market Prediction Using Deep-Learning Models. Procedia Computer Science, 132, 1351-1362.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8):1735–1780.

Kocatepe, C. İ., & Yıldız, O. (2016). Ekonomik Endeksler Kullanılarak Türkiye'deki Altın Fiyatındaki Değişim Yönünün Yapay Sinir Ağları ile Tahmini. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 4(3), 926-934.

Krizhevsky, A., ve diğ. 2012. Imagenet Classification With Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25 (NIPS 2012), sf: 1097-1105.

Lemin, G. (2016) Analysis of GM Grey Model and the Theory. Computer Engineering and Applications, 52(6), 58-63.

Liu, Weibo & Wang, Zidong & Liu, Xiaohui & Zeng, Nianyin & Liu, Yurong & Alsaadi, Fuad. (2016). A survey of deep neural network architectures and their applications. Neurocomputing. 234.

Mehryar, Mohri, A. R. 2012. Foundations of Machine Learning. Cambridge, UNITED STATES, MIT Press.

Montufar, Guido & Pascanu, Razvan & Cho, Kyunghyun & Bengio, Y.. (2014). On the Number of Linear Regions of Deep Neural Networks. NIPS 2014.

Pedrycz Witold, Shyi-Ming Chen, Deep Learning: Concepts and Architectures sayfa:18-21

Sadefo Kamdem, J., Bandolo Essomba, R., & Njong Berinyuy, J. (2020). Deep learning models for forecasting and analyzing the implications of COVID-19 spread on some commodities markets volatilities. Chaos, solitons, and fractals, 140, 110215.

Saini, Sandeep & Sahula, Vineet. (2018). Neural Machine Translation for English to Hindi. 1-6. 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP).

Sengupta, S., Basak, S., Saikia, P., Paul, S., Tsalavoutis, V., Atiah, F.D., Ravi, V., & Peters, R.A. (2019). A Review of Deep Learning with Special Emphasis on Architectures, Applications and Recent Trends. ArXiv.

Shrestha, Ajay & Mahmood, Ausif. (2019). Review of Deep Learning Algorithms and Architectures. IEEE Access. PP. 1-1.

Siami Namini, Sima & Tavakoli, Neda & Siami Namin, Akbar. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. 1394-1401.

Stuart J. Russell, Peter Norvig. 2010. Artificial Intelligence: A Modern Approach

Şeker, A. , Diri, B. & Balık, H. H. (2017). Derin Öğrenme Yöntemleri Ve Uygulamaları Hakkında Bir İnceleme . Gazi Mühendislik Bilimleri Dergisi , 3 (3) , 47-64 .

Şişmanoğlu, G. , Koçer, F. , Önde, M. A. & Sahingoz, O. K. (2020). Derin Öğrenme Yöntemleri ile Borsada Fiyat Tahmini . Bitlis Eren Üniversitesi Fen Bilimleri Dergisi , 9 (1) , 434-445 .

Tolon, M. and Tosunoğlu, N.G., 2008, Tüketici tatmini verilerinin analizi: Yapay sinir ağları ve regresyon analizi karşılaştırması, Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 10 (2), 247-259.

Yüksel, R., & Akkoç, S. (2016). Altın Fiyatlarının YapaySinir Ağları İle Tahmini ve Bir Uygulama. Doğuş Üniversitesi Dergisi, 17(1), 39-50.

Zhu, H. C., & Dong, W. (2017). Predict the price of gold Based on Machine Learning Techniques. DEStech Transactions on Computer Science and Engineering, (mmsta).

ÖZGEÇMİŞ

Adı Soyadı **:** Adem Üntez

ÖĞRENİM DURUMU

Derece	**Eğitim Birimi**	**Mezuniyet Yılı**
Yüksek Lisans	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü /Bilişim Sistemleri Mühendisliği	2022
Lisans	Gazi Üniversitesi / Mühendislik Fakültesi / Endüstri Mühendisliği	2016
Lise	Ahmet Eren Anadolu Lisesi	2010

İŞ DENEYİMİ

Yıl	**Yer**	**Görev**
2021-Halen	Ushaş Uluslararası Sağlık Hizmetleri A.Ş.	İş Analisti
2020-2021	Payolog	İş Analisti
2017-2020	Kuveyt Türk Katılım Bankası	Bilgi Güvenliği Uzmanı

YABANCI DİL

İngilizce