T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI

DERİN PEKİŞTİRMELİ ÖĞRENME İLE KRİPTO PARA FİYAT TAHMİNİ

DOKTORA TEZİ

Serbest ZİYANAK
Danışman: Prof. Dr. H. Eray ÇELİK

VAN – 2023

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI

DERİN PEKİŞTİRMELİ ÖĞRENME İLE KRİPTO PARA FİYAT TAHMİNİ

DOKTORA TEZİ

Serbest ZİYANAK

VAN – 2023

KABUL VE ONAY SAYFASI

Van Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü, İstatistik Anabilim Dalı'nda Prof. Dr. H. Eray ÇELİK danışmanlığında, Serbest ZİYANAK tarafından sunulan "Derin Pekiştirmeli Öğrenme ile Kripto Para Fiyat Tahmini" başlıklı bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince 12/05/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile başarılı bulunmuş ve Doktora tezi olarak kabul edilmiştir.

Başkan: Prof. Dr. Sinan ÇALIK İmza:

Üye: Prof. Dr. H. Eray ÇELİK İmza:

Üye: Doç. Dr. Hamit MİRTAGİOĞLU İmza:

Üye: Doç. Dr. Recep ÖZDAĞ İmza:

Üye: Doç. Dr. M. S. Talha ARSLAN İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …. /.... /….. tarih ve ..... sayılı kararı ile onaylanmıştır.

İmza

……

Enstitü Müdürü

ETİK BEYAN

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

İmza
Serbest ZİYANAK

ÖZET

DERİN PEKİŞTİRMELİ ÖĞRENME İLE KRİPTO PARA FİYAT TAHMİNİ

ZİYANAK, Serbest
Doktora Tezi, İstatistik Anabilim Dalı
Danışman: Prof. Dr. H. Eray ÇELİK
Mayıs 2023, 63 sayfa

Kripto para piyasaları son yıllarda yüksek volatilite, karmaşık dinamikler ve önemli finansal kazanç potansiyeli ile karşımıza çıkmaktadır. Bu piyasadaki yüksek kazanç potansiyeli, kripto para fiyat hareketlerinin tahmini için yapay zekâ modellerinin geliştirilmesi ihtiyacını doğurmaktadır. Mevcut derin öğrenme modellerinin, bu piyasadaki fiyat hareketlerini tahmin etmekte yetersiz kaldığı görülmektedir.

Bu tez kapsamında, derin pekiştirmeli öğrenme algoritmaları kullanılarak kripto para fiyat tahmini yapan modeller geliştirilmiştir. Eğitilen modeller karşılaştırılarak sonuçlar ele alınmıştır. Model eğitimi için Bitcoin fiyat verileri kullanılmıştır. Karşılaştırılan modeller arasından en iyi sonuç veren model kullanılarak kendi kendine alım-satım yapabilen ve kar etmeyi hedefleyen bir uygulama geliştirilmiştir.

Anahtar kelimeler: Bitcoin, Derin pekiştirmeli öğrenme, Kripto para, Yapay Zeka

ABSTRACT

CRYPTOCURRENCY PRICE PREDICTION WITH DEEP REINFORCED LEARNING

ZİYANAK, Serbest
Ph.D. Thesis, Department of Statistics
Supervisor: Prof. Dr. H. Eray ÇELİK
May 2023, 63 pages

Cryptocurrency markets have emerged with high volatility, complex dynamics and significant financial gain potential in recent years. The high earning potential in this market necessitates the development of artificial intelligence models for prediction of cryptocurrency price movements. Existing deep learning models seem to be insufficient to predict price movements in this market.

Within the scope of this thesis, models that make cryptocurrency price predictions have been developed using deep reinforcement learning algorithms. The results were discussed by comparing the trained models. Bitcoin price data is used for model training. By using the model that gives the best results among the compared models, an application that can trade by itself and aims to make a profit has been developed.

Keywords: Artificial Intelligence, Bitcoin, Cryptocurrency, Deep reinforcement rearning

TEŞEKKÜR

Bu tez çalışmasında, her türlü ilgi ve yardımlarını esirgemeyen danışmanım, rehberim, ağabeyim Sayın Prof. Dr. H. Eray ÇELİK'e teşekkür ederim. Ayrıca her türlü yardım ve desteğini benden esirgemeyen bilgi ve birikimlerinden sıkça faydalandığım arkadaşım, dostum Fatih ULUDAĞ'a, her zaman yanımda olan çocukluk arkadaşım kardeşim Fırat KAPAR'a, orijinal sözleri ve nasihatlarıyla bizi her zaman iyiliğe ve başarıya teşvik eden arkadaşım, ağabeyim M. S. Talha ARSLAN'a teşekkürlerimi sunarım. Uzun yıllar birlikte çalıştığım ve bilgisinden faydalandığım hocam Doç. Dr. Recep ÖZDAĞ teşekkürlerimi sunarım.

Bugünleri görmesini çok istediğim canım babama,

Koşulsuz sevgisiyle motivasyon kaynağım annem ve kardeşlerime,

Sürecin her parçasında benden daha fazla heyecanlanan, daha fazla korkan, daha fazla sevinen, en büyük yardımcım, hayat arkadaşım, yoldaşım sevgili eşim Güneş'e,

Biricik evlatlarım, canlarım Kazaz ve Kerem'e

2023

Serbest ZİYANAK

İÇİNDEKİLER

Sayfa

ÖZET i
ABSTRACT iii
TEŞEKKÜR v
İÇİNDEKİLER vii
ÇİZELGELER LİSTESİ ix
ŞEKİLLER LİSTESİ xi
SİMGELER VE KISALTMALAR xiii
1. GİRİŞ 1
2. KAYNAK BİLDİRİŞLERİ 3
3. MATERYAL VE YÖNTEM 7
3.1 Yapay Zekâ 7
3.2 Makine Öğrenmesi 8
3.3 Derin Öğrenme 10
3.3.1 Derin Öğrenme Modelleri ve Yöntemleri 11
3.3.1.1 Yapay Sinir Ağları (Artificial Neural Networks) 11
3.3.1.2 Evrişimli Sinir Ağları (Convolutional Neural Networks) 11
3.3.1.3 Tekrarlayan Sinir Ağları (Recurrent Neural Networks) 12
3.3.1.4 Çekişmeli Üretici Ağlar (Generative Adversarial Networks) 12
3.3.1.5 Derin Pekiştirmeli Öğrenme (Deep Reinforcement Learning) 13
3.3.2 Derin Öğrenmenin Uygulama Alanları 14
3.3.2.1 Görüntü Tanıma 14
3.3.2.2 Doğal Dil İşleme 14
3.3.2.3 Tavsiye Sistemleri 15
3.3.2.4 Sağlık Hizmetleri 15
3.3.2.5 Otonom Sürüş 15
3.3.3 Derin Öğrenme Sorunları ve Zorlukları 15
3.3.3.1 Veri İhtiyacı 15
3.3.3.2 Model Yorumlanabilirliği 16
3.3.3.3 Hesaplama Kaynakları 16
3.3.3.4 Aşırı Öğrenme (Overfitting) 16
3.4 Pekiştirmeli Öğrenme 16
3.4.1 Pekiştirmeli Öğrenmede Kullanılan Unsurlar 18
3.5 Derin Pekiştirmeli Öğrenme 21
3.5.1 Markov Karar Süreçleri 22
3.5.1.1 Markov Süreçleri 22

3.5.1.2 Markov Ödül Süreci ... 23
3.5.1.3 Değer Fonksiyonu ... 23
3.5.2 Bellman Denklemi ... 24
3.5.2.1 Markov Ödül Süreçlerinde Bellman Denklemi ... 24
3.5.2.2 Markov Karar Süreci ... 25
3.5.2.3 Politikalar ... 25
3.5.2.4 Eylem – Değer Fonksiyonu ... 26
3.5.2.5 Optimal Politika ... 28
3.5.2.6 Bellman Optimallik Denklemi ... 29
3.5.3 Derin Q-Öğrenme ... 29
3.5.3.1 Eylem – Değer Fonksiyonu ... 29
3.5.3.2 Zamansal Fark (Temporal Difference) Öğrenme Algoritması ... 30
3.5.3.3 Sarsa ... 31
3.5.3.4 Q-Öğrenme ... 32
3.5.3.5 Derin Q-Öğrenme ... 35
3.6 Blok Zinciri (Blockchain) Teknolojisi ... 38
3.7 Kripto Para ve Bitcoin ... 39
4. UYGULAMA ... 43
5. TARTIŞMA VE SONUÇ ... 55
KAYNAKLAR ... 57
ÖZ GEÇMİŞ ... 63

ÇİZELGELER LİSTESİ

Çizelge 4.1 Çok işlemcili eğitim sonuçları..... 51
Çizelge 4.2 Çok işlemcili test sonuçları 52
Çizelge 5.1 BTC, ETH, ADA, LTC ve XRP test sonuçları 55

ŞEKİLLER LİSTESİ

Şekil 3.1 Yapay zekâ, makine öğrenimi, derin öğrenme, pekiştirmeli öğrenme ve derin pekiştirmeli öğrenme arasındaki ilişki 8
Şekil 3.2 Makine öğrenmesi alt dalları 9
Şekil 3.3 Yapay sinir ağı örneği 11
Şekil 3.4 Klasik bir CNN mimarisi 12
Şekil 3.5 Tekrarlayan sinir ağı örneği 12
Şekil 3.6 Çekişmeli üretici ağlar 13
Şekil 3.7 Görüntü tanıma aşamaları 14
Şekil 3.8 Pekiştirmeli öğrenme döngüsü 17
Şekil 3.9 Reinforcement learning durum, aksiyon, ödül örneği 18
Şekil 3.10 Pekiştirmeli öğrenme temel unsurları 19
Şekil 3.11 Pekiştirmeli öğrenmenin Markov Karar Süreci ile gösterimi 20
Şekil 3.12 Derin pekiştirmeli öğrenme şeması 21
Şekil 3.13 s durumundan s' durumuna stokastik geçiş 24
Şekil 3.14 v(s) ve q(s,a) arasındaki ilişki 26
Şekil 3.15 q(s,a) ile v(s') arasında olası durumlar 27
Şekil 3.16 q(s,a) ile q(s',a') arasındaki özyinelemeli ilişkiyi gösteren ağaç yapısı 28
Şekil 3.17 Örnek bir S durumunda farklı a eylemleri ve Q değerleri 30
Şekil 3.18 Q-öğrenme algoritması akış şeması 34
Şekil 3.19 Q-öğrenme ve Derin Q-öğrenme teknikleri arasındaki fark 36
Şekil 3.20 Q-Ağ (Q-Network) ve Hedef Ağ (Target Network)'ın Derin Q- Öğrenme döngüsündeki yeri ve Derin Q-Ağındaki(DQN) Kayıp Fonksiyon 37
Şekil 4.1 Uygulamada kullanılan Bitcoin verileri 44
Şekil 4.2 Ortam için gerekli örnek arabirim 45
Şekil 4.3 Pekiştirmeli öğrenme ile rastgele alışveriş yapan ajanın elde ettiği gelir 46
Şekil 4.4 Aktör model 47
Şekil 4.5 Eleştirmen model 47
Şekil 4.6 Aktör-eleştirmen model mimarisi ile eğitim süreci grafikleri 48
Şekil 4.7 Paylaşılan katmana sahip aktör-eleştirmen model mimarisi 49
Şekil 4.8 Dense ve CNN modeli Tensorboard grafiği 50

Şekil 4.9 Bitcoin kapanış değerleri 53
Şekil 4.10 Fark alma yönteminden sonra Bitcoin kapanış değerleri 53
Şekil 4.11 Logaritmik fark alma işleminden sonra Bitcoin kapanış değerleri 54
Şekil 4.12 0 ile 1 arasına alınarak normalizasyon uygulanmış Bitcoin kapanış değerleri 54

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamalarıyla aşağıda sunulmuştur.

Kısaltmalar	**Açıklama**
CNN	Convolutional Neural Network
DNN	Deep Neural Network
GAN	Generative Adversarial Networks
LSTM	Long Short-Term Memory
MDP	Markov Decision Process
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
ReLU	Rectified Linear Unit
TD	Temporal Difference
YSA	Yapay Sinir Ağları

1. GİRİŞ

Kripto paralar, son on yılın en önemli teknolojik ve ekonomik gelişmelerinden biri olarak ortaya çıkmıştır. Satoshi Nakamoto'nun 2008 yılında yayımladığı "Bitcoin: A peer-to-peer electronic cash system" İsimli makalesi ile başlayan bu süreç, şu anda dünya genelinde milyonlarca insanın günlük hayatında ve ticari işlemlerde kullanmış olduğu bir teknoloji haline gelmiştir (Nakamoto, 2008). Kripto paralar, merkezi olmayan, şifreleme algoritmalarına dayalı, dağıtık defter teknolojisine (Blockchain) dayanan dijital varlıklardır (Swan, 2015). Kripto paraların değeri, geçen süre zarfında önemli ölçüde artış veya düşüş gösterse de bu dijital varlıkların gelecekte finansal sistemlerde önemli bir rol oynayacağına dair genel bir kanaat bulunmaktadır (Swan, 2015).

Kripto para piyasaları, finansal piyasaların bir alt dalı olarak hızla büyümekte ve giderek daha fazla yatırımcının ilgisini çekmektedir (Hileman ve Rauchs, 2017). Bu yeni ve dinamik piyasalar, yüksek volatilite ve belirsizlik nedeniyle diğer finansal piyasalardan farklıdır (Corbet vd., 2018). Bu nedenle, kripto para fiyat tahminleri için geleneksel finansal modellerin etkinliği sorgulanabilmektedir (Gkillas vd., 2019). Bu bağlamda, derin pekiştirmeli öğrenme yöntemlerinin kripto para piyasalarındaki fiyat hareketlerini tahmin etme ve bu hareketlerden faydalanma becerisi araştırmacılar ve yatırımcılar için büyük bir öneme sahiptir.

Kripto para piyasası, özellikle yüksek volatilite ve belirsizlik nedeniyle finansal tahminin zor olduğu bir alandır (Bouri vd., 2017). Geleneksel istatistiksel yöntemlerin yanı sıra, daha önce yapılan çalışmalar makine öğrenimi ve derin öğrenme yöntemlerini kullanarak kripto para fiyatlarını tahmin etmeye çalışmıştır (McNally vd., 2018; Jiang ve Liang, 2017).

Yapay zeka ve makine öğrenimi alanlarında son yıllarda kaydedilen önemli ilerlemeler, insanların düşünme ve öğrenme yeteneklerini taklit edebilen bilgisayar sistemlerinin geliştirilmesine olanak sağlamıştır (LeCun vd., 2015). Derin pekiştirmeli öğrenme, bu alanlardan biri olup, insanların öğrenme süreçlerini modelleyen bir yapay sinir ağı (YSA) mimarisi kullanmaktadır (Sutton ve Barto, 2018). Derin pekiştirmeli öğrenme algoritmaları, öğrenme sürecinde ödül ve ceza mekanizmalarını kullanarak karmaşık ve belirsiz ortamlarda etkili kararlar alabilmektedir (Mnih vd., 2015).

Bu tez, derin pekiştirmeli öğrenme yöntemlerinin, kripto para fiyat tahmini yapabilmek için uygun olup olmadığını incelemeyi amaçlamaktadır. Derin pekiştirmeli öğrenme, yapay sinir ağları ve pekiştirmeli öğrenme prensipleri üzerine kurulu olan bir yapay zekâ alt dalıdır (Mnih vd., 2015). Bu yöntem, özellikle atari oyunlarından AlphaGo'ya kadar birçok alanda başarılı sonuçlar vermiştir (Silver vd., 2016). Derin pekiştirmeli öğrenme, öğrenme sürecinde insanlar gibi deneme yanılma yoluyla hareket ederek ve alınan geri bildirimlerle modelin performansını optimize etmeyi amaçlamaktadır (Sutton ve Barto, 2018).

Bu tez, kripto paraların yüksek volatilite ve düşük likidite gibi özellikleri göz önünde bulundurarak, derin pekiştirmeli öğrenme yöntemlerini kullanarak bu alanda nasıl başarılı tahminler yapabileceğimize odaklanmaktadır. Yapılan literatür taraması ve uygulamalar sonucunda, elde edilen bulguların, kripto para tahminlerinde kullanılabilecek etkili araçlar geliştirmeye katkı sağlaması ve bu alandaki bilimsel çalışmalara yol gösterici olması hedeflenmektedir. Bu sayede, kripto paraların gelecekteki finansal sistemlerdeki rolünün daha iyi anlaşılması ve bu dijital varlıkların fiyatlarının daha doğru ve güvenilir bir şekilde tahmin edilmesi amaçlanmaktadır.

Ayrıca, tezde kullanılan yöntem ve elde edilen bulguların, kripto paralarla ilgili politika ve düzenlemelerin oluşturulmasına ve uygulanmasına katkı sağlayabilecek bilgiler sunması beklenmektedir.

Sonuç olarak, bu tez, derin pekiştirmeli öğrenme yöntemlerinin kripto para tahminlerinde kullanılabilirliğini ve başarısını değerlendirmeyi amaçlayan kapsamlı bir çalışmadır. Elde edilen bulgular, kripto paraların gelecekteki finansal sistemlerdeki önemini anlamamıza ve bu alandaki bilimsel çalışmalara katkı sağlamaya yönelik önemli bir adım olarak değerlendirilebilir. Bu tezin sonunda, derin pekiştirmeli öğrenme yöntemlerinin kripto para tahminlerindeki başarısını artırmak ve bu alanda daha etkili ve güvenilir tahminler yapmak için geliştirilebilecek potansiyel araştırma yönleri ve uygulama alanları hakkında öneriler sunulacaktır.

2. KAYNAK BİLDİRİŞLERİ

Derin öğrenme, yapay sinir ağları temelli bir makine öğrenimi yaklaşımıdır ve gözetimli, gözetimsiz ve pekiştirmeli öğrenme tekniklerini kullanır (LeCun, Bengio ve Hinton, 2015). Derin öğrenme yöntemleri, çok katmanlı yapay sinir ağları kullanarak karmaşık veri yapılarının özelliklerini modelleyebilir ve öğrenebilir. Derin öğrenme modellerinin başarısı, büyük veri kümeleri ve yüksek performanslı bilgisayar donanımı sayesinde artmıştır (Goodfellow vd., 2016).

Pekiştirmeli öğrenme, öğrenme sürecinde deneyim kazanarak ve alınan geri bildirimlerle iyileşen bir makine öğrenimi yaklaşımıdır (Sutton ve Barto, 2018). Pekiştirmeli öğrenmede, ajan, belirli bir durumda yapılacak eylemi seçerek ve geri bildirim (ödül) alarak öğrenir. Ajanın amacı, toplam ödülü en üst düzeye çıkaran eylem politikasını öğrenmektir.

Kripto para tahminlerinde derin öğrenme yöntemleri kullanılarak başarılı sonuçlar elde edilmiştir. Özellikle Long short-term memory (LSTM) ve Convolutional Neural Network (CNN) gibi derin öğrenme modelleri, kripto paraların zaman serisi verilerini modellemek için kullanılmıştır (Sirignano ve Cont, 2018; Stanciu, 2019).

Jiang ve Liang (2017) tarafından yapılan bir başka çalışmada, CNN kullanarak kripto para fiyat hareketlerini tahmin etme konusunda başarılı sonuçlar elde edilmiştir. CNN modelleri, kripto para fiyat hareketlerinin özelliklerini öğrenerek ve modelleyerek başarılı tahminler yapmıştır.

McNally vd. (2018) çalışmalarında Bitcoin fiyatlarını tahmin etmek için makine öğrenimi yöntemlerini kullanmışlardır. Çalışmada LSTM ve diğer geleneksel yöntemlerin performansını karşılaştırarak, kripto para tahminlerinde derin öğrenme yöntemlerinin etkinliğini değerlendirilmiştir. Sonuçlar, LSTM modelinin, kripto para fiyatlarını tahmin etmede geleneksel yöntemlerden daha başarılı olduğunu göstermektedir.

Kim vd. (2016), kullanıcı yorumları ve yanıtlarını temel alarak kripto para işlem dalgalanmalarını tahmin etmeyi amaçlamıştır. Kim vd. (2016), sosyal medya verilerini analiz ederek kripto para fiyat hareketlerinin tahminine katkıda bulunan faktörleri belirlemeyi hedeflemişlerdir. Bu amaçla, derin öğrenme yöntemleri kullanılarak sosyal medya verilerinden elde edilen bilgiler analiz edilip kripto para piyasasındaki

dalgalanmalar tahmin etmeye çalışılmıştır. Kim vd. (2016) yaptıkları çalışmada, derin öğrenme yöntemleri kullanarak sosyal medya verilerinden elde edilen bilgilerin kripto para piyasasındaki dalgalanmaları tahmin etmede başarılı olduğunu göstermektedir. Bu sonuç, sosyal medya analizinin ve derin öğrenme yöntemlerinin kripto para tahminlerinde etkili olduğunu ortaya koymaktadır.

Bao vd. (2017) yaptıkları çalışmada finansal zaman serilerini tahmin etmek için derin öğrenme tabanlı bir yöntem kullanmaktadırlar. Araştırmacılar, derin otomatik kodlayıcılar ve LSTM mimarilerini kullanarak hisse senedi fiyatları ve döviz kurları üzerinde tahminlerde bulunmaktadırlar. Çalışma, bu derin öğrenme yaklaşımının, finansal zaman serilerini tahmin etmede geleneksel yöntemlerden ve bazı diğer derin öğrenme modellerinden daha iyi performans gösterdiğini ortaya koymaktadır.

Fischer ve Krauss (2018)'nin yapmış olduğu çalışma, derin öğrenme ve LSTM ağlarının finansal piyasaları tahmin etmedeki etkinliğini incelemektedir. Çalışma, hisse senedi fiyatları, döviz kurları ve diğer finansal zaman serileri üzerinde deneyler yaparak, LSTM ağlarının performansını değerlendirmektedir. Çalışmanın sonuçları, LSTM tabanlı derin öğrenme modellerinin, finansal piyasa tahminlerinde geleneksel yöntemlere kıyasla daha iyi sonuçlar elde ettiğini göstermektedir.

Sezer ve Ozbayoglu (2018), yapmış oldukları çalışmada, algoritmik finansal işlemlerde CNN kullanarak zaman serisi verilerini görüntüye dönüştüren bir yaklaşımı incelemişlerdir. Araştırmacılar, CNN kullanarak, hisse senedi fiyatları ve diğer finansal zaman serileri için tahminlerde bulunmaktadır. Çalışma, zaman serisi verilerini görüntüye dönüştürerek CNN'lerin kullanılmasının, finansal piyasalardaki fiyat hareketlerini tahmin etmede başarılı olduğunu göstermektedir. Bu yaklaşım, finansal işlemlerde derin öğrenme yöntemlerinin potansiyelini ortaya koymaktadır.

Ding vd. (2015), olaya dayalı hisse senedi tahminleri için derin öğrenme yöntemlerini kullanmışlardır. Araştırmacılar, finansal olaylar ve haberlerle ilgili metin verilerini analiz ederek, hisse senedi fiyatlarına yönelik tahminlerde bulunan bir model geliştirmişlerdir. Çalışmada LSTM ve derin dil modelleri gibi derin öğrenme mimarileri kullanılmış ve tarihsel fiyat verileri ve finansal haber metinleri temel alınmıştır. Çalışma, derin öğrenme yöntemlerinin, finansal olaylara dayalı hisse senedi fiyat tahminlerinde etkili olduğunu göstermektedir. Bu sonuç, finansal piyasalarda olay

analizinin ve derin öğrenme yöntemlerinin önemli bir rol oynadığını ortaya koymaktadır.

Mevcut literatür, derin öğrenme ve pekiştirmeli öğrenme yöntemlerinin kripto para tahminleri ve alım-satım stratejilerinde başarılı sonuçlar elde ettiğini göstermektedir. Bununla birlikte, bu yöntemlerin performansının ve başarısının daha fazla araştırılması ve iyileştirilmesi gerektiği söylenebilir.

Özellikle, derin pekiştirmeli öğrenme yöntemlerinin farklı kripto paralar, zaman aralıkları ve piyasa koşulları için daha fazla test edilmesi ve değerlendirilmesi önemlidir. Ayrıca, mevcut algoritmaların ve tekniklerin daha fazla optimize edilmesi ve yeni algoritmaların geliştirilmesi, derin pekiştirmeli öğrenme yöntemlerinin kripto para tahminlerindeki başarısını daha da artırabilir. Ayrıca, pekiştirmeli öğrenme algoritmalarının hiperparametre optimizasyonu ve yapılandırması da daha fazla araştırma gerektiren bir konudur.

Bunun yanı sıra, kripto para piyasalarındaki duyarlılık analizi ve sosyal medya verilerinin kullanımı, kripto para tahminlerinin doğruluğunu ve güvenilirliğini artırmaya yönelik önemli bir potansiyel sunmaktadır (Liew vd., 2019). Duyarlılık analizi ve sosyal medya verilerinin derin pekiştirmeli öğrenme modellerine entegrasyonu, gelecek araştırmalar için önemli bir alan olabilir.

Son olarak, kripto para tahminleri ve alım satım stratejilerinde kullanılacak yeni ve gelişmiş derin pekiştirmeli öğrenme algoritmalarının geliştirilmesi, bu alandaki araştırmaların gelecekteki yönlerini belirleyecektir.

3. MATERYAL VE YÖNTEM

3.1 Yapay Zekâ

Bilgisayarlar ilk icat edildiğinden beri, bilim adamları makineleri daha akıllı hale getirmeye çalışmaktadırlar (McCorduck, 2004). Ancak, zekanın tanımı bugüne kadar hala tartışılmaktadır. Bu nedenle, Sir Alan Turing, zekânın ne olduğunu tanımlamadan, Turing Testini ilk olarak 1950'de Manchester Üniversitesi'ndeki "Bilgisayar Makineleri ve Zekâ" başlıklı makalesinde tanıtmıştır (Turing, 1950).

Turing testi, bir makinenin akıllı insan davranışını taklit edebilme yeteneğini ölçer. Turing testine göre, bir sorgulayıcı farklı odalarda bulunan bir bilgisayar ve bir insana bir dizi soru sorar. Sorgulayıcı aldığı cevaplara göre hangisinin bilgisayar hangisinin insan olduğunu saptamaya çalışır. Bilgisayar sorgulayıcıyı kandırabilirse testin başarılı olduğu anlamına gelmektedir (Turing, 1950).

Yapay Zekâ, 1955'de ünlü Dartmouth konferansında John McCarthy tarafından tanıtılmıştır (McCarthy vd., 1955). Bu konferans, Yapay zekâ'nın bilgisayar biliminin bir alt disiplini olmasının başlangıç noktası olarak görülmektedir (McCorduck, 2004). Yapay Zekâ algoritmaları 1950'lerden günümüze, temel olarak matematiksel kurallar ve mantık kuralları ile formüle edilebilen problemleri çözmek için tasarlanmıştır (Russell ve Norvig, 2016).

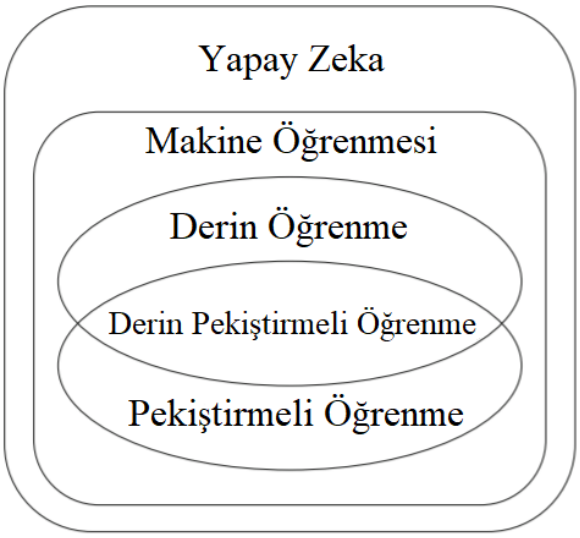
Bilgi işlem gücündeki ve veri hacmindeki hızlı gelişmelerden maksimum düzeyde faydalanan yapay zekâ, bir zamanlar satranç ve Go gibi masa oyunları, video oyunları, hastalık teşhisi gibi makineler için zorlu kabul edilen birçok görevde artık insanlardan daha iyi performans göstermektir (Silver vd., 2016).

Yapay zekâ teknolojisi günümüzde ilaç keşfi, hava durumu tahmini, gelişmiş malzemeler, öneri sistemleri, robotik algılama ve kontrol, otonom sürüş, yüz tanıma, konuşma tanıma ve diyalog gibi diğer uygulamalara da yaygın olarak kullanılmaktadır (Goodfellow vd., 2016).

Son on yılda Türkiye, Çin, İngiltere, ABD, Japonya ve Almanya gibi ülkeler yapay zekanın gelişimini desteklemek için somut yapay zekâ politika ve stratejilerini yürürlüğe koymuşlardır. Google, Facebook, MicroSoft, Apple, Baidu, Huawei ve Tencent gibi teknoloji devleri de yapay zekâ araştırmalarına milyarlar harcamaktadırlar

(Knight, 2017). Otonom araçlar, yüz tanıma ve sohbet robotları gibi yapay zekâ uygulamaları günlük hayatımızın neredeyse her yerinde kullanılmaktadır (Mnih vd., 2015; Taigman vd., 2014).

Yapay zekâ, bilgisayarların ve diğer makinelerin insan benzeri öğrenme, düşünme ve problem çözme yeteneklerini simüle etme süreci olarak da tanımlanmaktadır (Russell ve Norvig, 2016). Yapay zekâ birçok alt dala ayrılır ve bu alt dalların her biri, yapay zekâ sistemlerinin farklı yetenekler ve uygulamalar geliştirmesine yardımcı olur (Goodfellow vd., 2016). Makine öğrenmesi de bu alt dallardan bir tanesidir (Alpaydin, 2014). Şekil 3.1'de Yapay zekâ ve makine öğrenmesinin bazı alt dalları arasındaki ilişki gösterilmektedir.



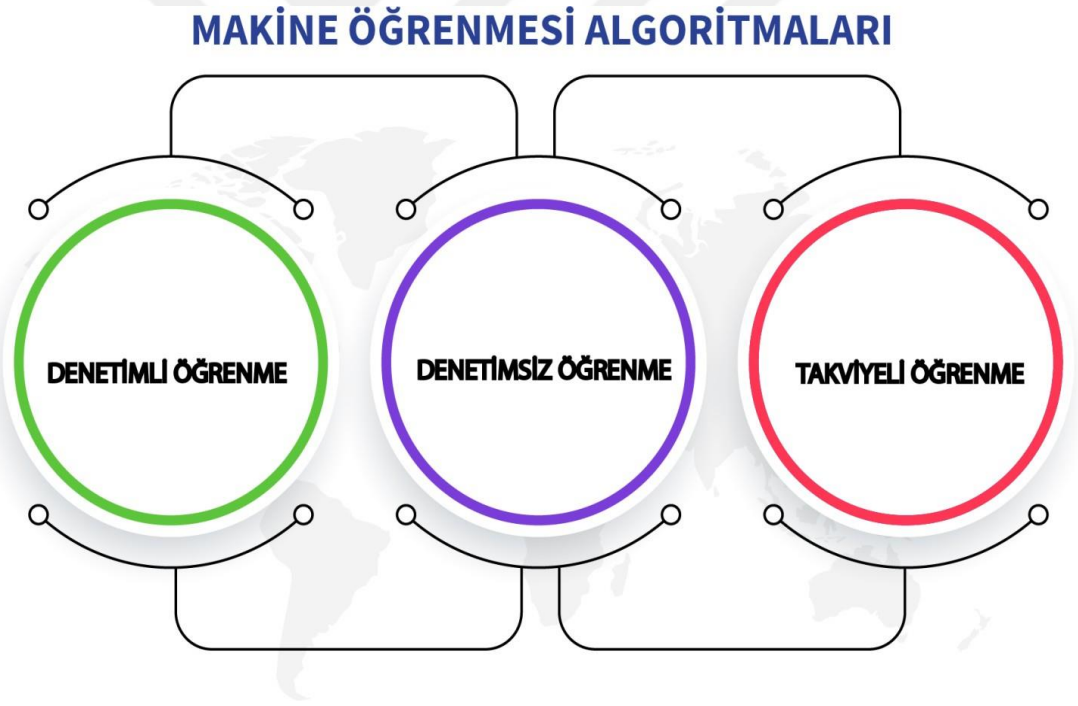
Şekil 3.1 Yapay zekâ, makine öğrenimi, derin öğrenme, pekiştirmeli öğrenme ve derin pekiştirmeli öğrenme arasındaki ilişki

3.2 Makine Öğrenmesi

Makine öğrenmesi veya makine öğrenimi, yapay zekâ biliminin bir alt dalıdır. Makine öğrenmesi, algoritmalar kullanarak verilerden öğrenme ve öğrenilen bilgilerle doğru tahminlerde bulunarak karar verme süreçlerini simüle etme amacı taşımaktadır (Goodfellow vd., 2016). İnsan benzeri öğrenme yetenekleri sergileyen bu modeller, büyük miktarda veriyi analiz ederek, düşünce süreçlerini ve akıllı karar verme mekanizmalarını taklit etmeye çalışır (Mitchell, 1997). Makine öğrenimi, veri

madenciliği, öğrenme teorisi ve optimizasyon gibi disiplinlerin kesiştiği bir alan olup, bu alanlardan edindiği bilgi ve yöntemlerle çeşitli sektörlerde kullanılmaktadır (Jordan ve Mitchell, 2015).

Makine öğrenimi, genellikle üç ana kategoriye ayrılır: denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning) ve pekiştirmeli öğrenme (reinforcement learning) (Kelleher vd., 2015). Denetimli öğrenmede, algoritma önceden etiketlenmiş verilerle eğitilir ve girdi-veri çiftlerini kullanarak doğru tahminlerde bulunmayı öğrenir (Russell ve Norvig, 2016). Denetimsiz öğrenmede ise, algoritma etiketsiz verileri kullanarak veri içindeki yapıları veya kalıpları keşfetmeye çalışır (Hinton vd., 2012). Pekiştirmeli öğrenme, algoritmanın eylemlerle ve geri bildirimlerle öğrenmesine dayanan bir yaklaşımdır. Algoritma, geri bildirimlere göre performansını en üst düzeye çıkarmayı amaçlar (Sutton ve Barto, 2018).



Şekil 3.2 Makine öğrenmesi alt dalları

Makine öğrenmesi, birçok endüstride etkili bir şekilde kullanılmaktadır. Tıbbi teşhislerden, otomobil endüstrisine, finans sektöründen, e-ticaret platformlarına kadar pek çok alanda makine öğrenmesi, öngörücülüğü keşfedici, müşteri segmentasyonu ve veri sınıflandırması gibi işlemlerde önemli bir rol oynamaktadır (Bishop, 2006).

Özellikle büyük veri çağında, makine öğrenmesi algoritmaları veri yığınlarını analiz ederek; süreçleri daha verimli ve etkili hale getirmekte büyük bir potansiyele sahiptir (Dhar, 2013).

Makine öğrenmesinin geleceği, algoritmaların daha hızlı, doğru ve ölçeklenebilir hale gelmesi yönünde ilerlemektedir (Halevy vd., 2009). Ayrıca, denetimsiz ve pekiştirmeli öğrenme gibi daha karmaşık öğrenme yöntemlerinin yaygınlaşması, yapay zekâ sistemlerinin insan beynine daha çok benzeyen öğrenme ve düşünme süreçlerine sahip olmalarına olanak tanıyacaktır (LeCun vd., 2015). Öğrenme süreçlerini ve veri işleme hızını geliştiren donanım ve yazılım teknolojileri de makine öğreniminin uygulama alanlarını ve etkisini artırmaya yardımcı olacaktır (Dean vd., 2012).

Sonuç olarak, makine öğrenmesi, yapay zekâ alanında hızla gelişen ve giderek daha geniş bir etkiye sahip olan bir alt daldır. Bu alandaki ilerlemeler, çeşitli sektörlerde ve günlük yaşamda karşılaştığımız problemlerin çözümünde büyük bir potansiyele sahiptir. Bunların yanında etik ve güvenlik endişelerini göz önünde bulundurarak, makine öğrenimi uygulamalarının dikkatli bir şekilde geliştirilmesi ve kullanılması önem taşımaktadır.

3.3 Derin Öğrenme

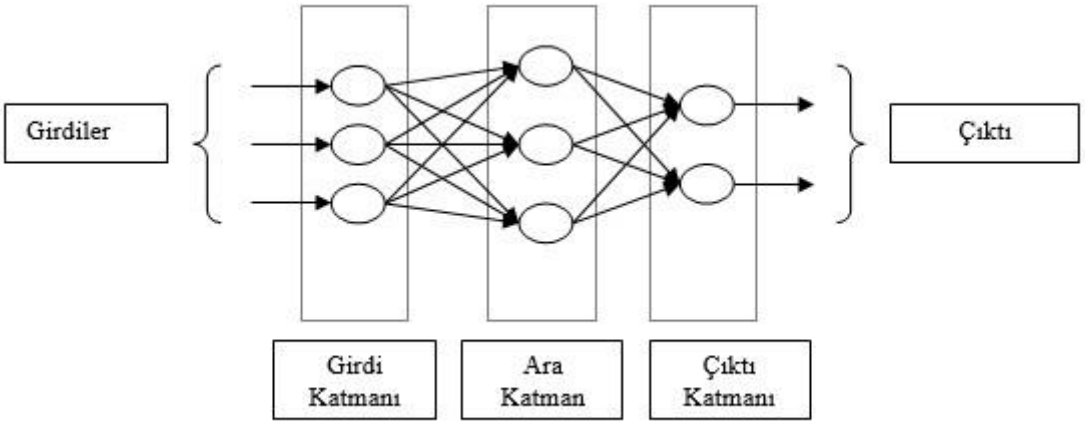
Derin öğrenme, genellikle büyük miktarda veriyi işlemek ve özellik çıkarımı yapmak için çok katmanlı yapay sinir ağları kullanan makine öğrenmesinin bir alt dalıdır. Yapay sinir ağları, öğrenme sürecinde veriden özellikleri otomatik olarak çıkarabilmektedirler (LeCun vd., 2015). Derin öğrenme modelleri, resim ve ses tanıma, doğal dil işleme gibi karmaşık problemlerde, insan beyninin öğrenme biçimini taklit etmeye çalışmaktadır.

Derin öğrenme, makine öğrenmesinin daha özelleşmiş bir formudur. Genellikle büyük veri kümelerinde karmaşık özellikler öğrenmek için kullanılmaktadır. Derin öğrenme, veri içindeki hiyerarşik özellik seviyelerini modellemek için işlem katmanları kullanmaktadırlar. (Goodfellow vd., 2016).

3.3.1 Derin Öğrenme Modelleri ve Yöntemleri

3.3.1.1 Yapay Sinir Ağları (Artificial Neural Networks)

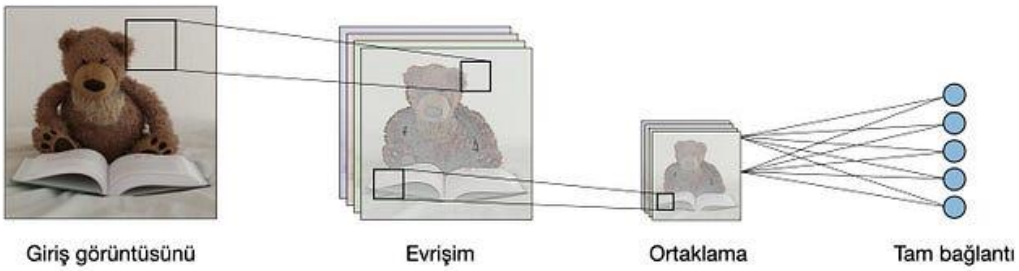
Yapay Sinir Ağları, insan beyninin çalışma mekanizmasını taklit etmeye çalışan bir hesaplama modeli olarak tanımlanabilmektedir. Yapay sinir ağları, nöronlar arasında karmaşık bağlantılar oluşturan ve bu nöronların birbirleriyle nasıl iletişim kurduğunu kontrol eden katmanlardan oluşmaktadır (McCulloch ve Pitts, 1943). Şekil 3.3'de basit bir yapay sinir ağı örneği gösterilmektedir.



Şekil 3.3 Yapay sinir ağı örneği

3.3.1.2 Evrişimli Sinir Ağları (Convolutional Neural Networks)

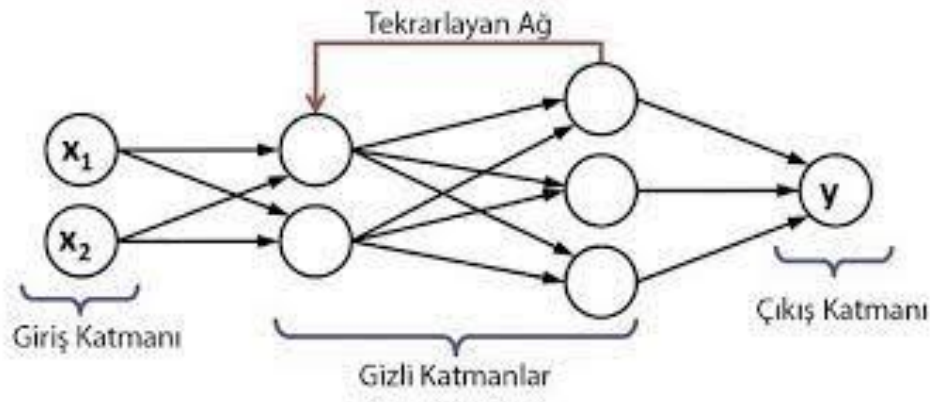
Evrişimli Sinir Ağları (Convolutional Neural Networks - CNN), bir görüntünün farklı bölgelerini analiz ederek özelliklerini öğrenen ve genellikle görüntü sınıflandırma görevlerinde kullanılan özel bir yapay sinir ağı türüdür. CNN'ler, yerel ve global bilgiyi birleştirme yeteneği sayesinde, genellikle doğal görüntülerdeki karmaşık özellikleri öğrenmek için kullanılır (LeCun vd., 1998).



Şekil 3.4 Klasik bir CNN mimarisi

3.3.1.3 Tekrarlayan Sinir Ağları (Recurrent Neural Networks)

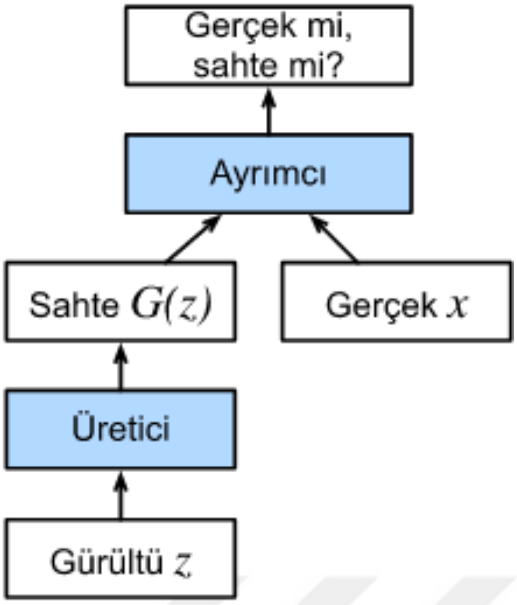
Tekrarlayan Sinir Ağları (Recurrent Neural Networks - RNN), zaman serisi gibi sıralı verileri daha iyi işleyebilen ve önceki girişlerin hafızasını koruyabilen bir yapay sinir ağı türüdür. Bu özelliği onları, zaman serisi tahmini, konuşma tanıma ve doğal dil işleme gibi görevler için ideal hale getirmektedir (Hochreiter ve Schmidhuber, 1997).



Şekil 3.5 Tekrarlayan sinir ağı örneği

3.3.1.4 Çekişmeli Üretici Ağlar (Generative Adversarial Networks)

Çekişmeli Üretici Ağlar (Generative Adversarial Networks - GAN), iki ayrı ağın birbirine karşı eğitildiği bir yapıdan oluşmaktadır: biri gerçekçi veri örnekleri oluştururken (üretici ağ), diğeri oluşturucunun ürettiği sahte verinin gerçek veriden ayırt edilmesini öğrenmektedir (ayrımcı ağ). Bu durum GAN'ların, kalitesi yüksek sahte veriler oluşturma yeteneğini öne çıkarmaktadır (Goodfellow vd., 2014).



Şekil 3.6 Çekişmeli üretici ağlar

Şekil 3.6'da GAN mimarisi basit bir şekilde gösterilmiştir. GAN mimarisinde iki ayrı ağ olduğu görülmektedir bunlar üretici ve ayrımcı ağlardır. Üretici ağ gerçeğe benzer sahte veriler üreterek ayrımcı ağı kandırmaya çalışır. Bir noktada ayrımcı ağ yeni sahte verileri daha iyi tanımaya başlar. Bu bilgi de üretici ağın iyileşmesini sağlamaktadır. Bu şekilde her iki ağ çekişmeli bir şekilde birbirine rekabet etmektedir. Bu durum ağın daha fazla gelişmesini sağlamaktadır.

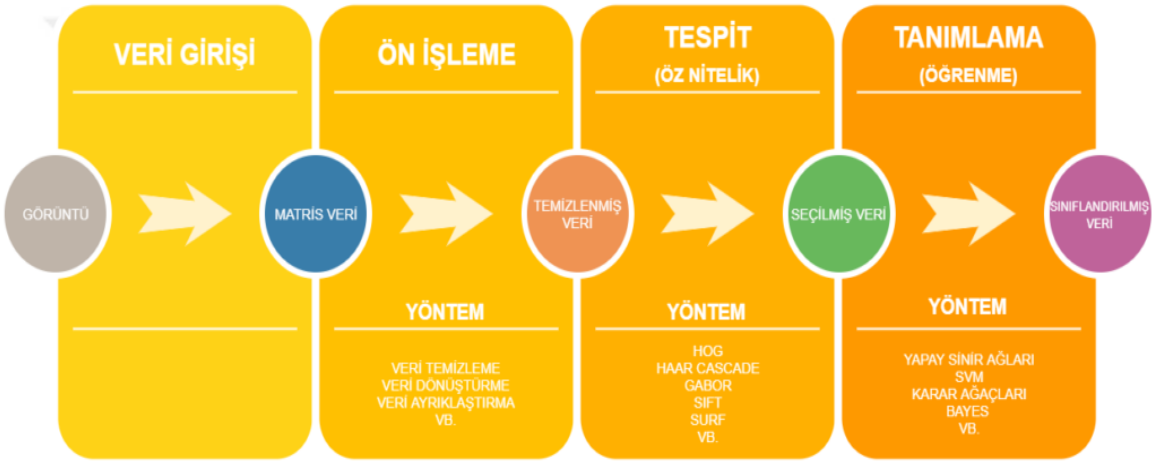
3.3.1.5 Derin Pekiştirmeli Öğrenme (Deep Reinforcement Learning)

Derin pekiştirmeli öğrenme, bir ajanın bir çevreyle etkileşim kurarak ve kararlarının sonuçlarına dayanarak öğrenmesini sağlayan bir derin öğrenme alt dalıdır. Ajan, belirli bir görevi en iyi şekilde tamamlamak için hangi eylemleri gerçekleştirmesi gerektiğini öğrenir. Bu, oyun oynama (örneğin, AlphaGo), otomatik sürüş ve robot kontrolü gibi karmaşık görevlerde başarı elde etmek için kullanılabilir (Mnih vd., 2015). Derin pekiştirmeli öğrenme bu tezde kullanılan temel derin öğrenme algoritmasıdır. Bu nedenle bu konu tezin ileriki kısımlarında daha detaylı anlatılmaktadır.

3.3.2 Derin Öğrenmenin Uygulama Alanları

3.3.2.1 Görüntü Tanıma

Görüntü tanıma, derin öğrenmenin en yaygın uygulamalarından biridir. CNN'ler, görüntü sınıflandırma, nesne tespiti ve yüz tanıma gibi görevlerde genellikle başarı sağlamaktadırlar. Bu teknoloji, sağlık hizmetleri, güvenlik ve otomatik sürüş gibi bir dizi endüstride sıkça kullanılmaktadır (Krizhevsky vd., 2012). Şekil 3.7'de görüntü tanıma aşamaları gösterilmektedir.



Şekil 3.7 Görüntü tanıma aşamaları

3.3.2.2 Doğal Dil İşleme

Doğal Dil İşleme (NLP), insan dili verilerini anlama ve işleme yeteneğine sahip en önemli derin öğrenme uygulama alanlarından biridir. Derin öğrenme, metin sınıflandırma, duygu analizi, otomatik özetleme ve makine çevirisi gibi NLP görevlerinde önemli gelişmeler sağlamıştır. Bu teknoloji, müşteri hizmetleri, pazarlama ve sosyal medya analitiği gibi alanlarda yaygın olarak kullanılmaktadır (Devlin vd., 2019).

3.3.2.3 Tavsiye Sistemleri

Tavsiye sistemleri, kullanıcıların ilgi alanlarını ve tercihlerini anlamak için derin öğrenme tekniklerini kullanan uygulama alanlarından bir tanesidir. Bu sistemler, müşterilere ürün veya hizmet önerileri sunarak, e-ticaret, müzik ve film önerileri ve kişiselleştirilmiş reklamlar gibi uygulamalarda kullanılmaktadırlar. (Zhang vd., 2019).

3.3.2.4 Sağlık Hizmetleri

Derin öğrenme, sağlık hizmetlerinde de önemli bir rol oynamaktadır. Tıbbi görüntü analizi, hastalık teşhisi ve tedavi önerileri gibi birçok uygulama alanı bulunmaktadır. Derin öğrenme algoritmaları, doktorların hastalıkları daha hızlı ve doğru bir şekilde teşhis etmelerine yardımcı olabilmektedirler (Litjens vd., 2017).

3.3.2.5 Otonom Sürüş

Otonom sürüş, derin öğrenmenin önemli uygulama alanlarından biridir. Otonom araçlar, çevrelerini algılamak, anlamak ve uygun şekilde tepki vermek için derin öğrenme algoritmalarına büyük ölçüde bağımlıdır. Bu teknoloji, yol tespiti, nesne tespiti ve çarpışma önleme gibi görevlerde kullanılmaktadır (Bojarski vd., 2016).

Derin öğrenme, çeşitli sektörlerde ve uygulama alanlarında önemli ve heyecan verici sonuçlar sağlamıştır. Bu teknolojinin ilerlemesi ve gelişimi, geniş bir uygulama yelpazesi sunarak, birçok endüstrinin operasyonel verimliliğini, hizmet kalitesini ve müşteri deneyimini geliştirmesine yardımcı olmaktadır.

3.3.3 Derin Öğrenme Sorunları ve Zorlukları

3.3.3.1 Veri İhtiyacı

Derin öğrenme modellerinin eğitilmesinde, genellikle büyük miktarda etiketli verilere ihtiyaç duyulmaktadır. Bu, özellikle veri etiketleme sürecinin zaman alıcı veya

pahalı olduğu durumlarda bir sorun olarak karşımıza çıkabilmektedir. Bu durum, derin öğrenme projelerinin başarısı için veri toplama ve etiketleme işlemlerinin ne kadar önemli olduğunu ortaya koymaktadır (Sun vd., 2017).

3.3.3.2 Model Yorumlanabilirliği

Derin öğrenme modelleri genellikle "kara kutu" olarak kabul edilmektedir. Çünkü bu modeller, tahminlerini yaparken hangi özelliklere dayanarak yaptıklarını açıklayamamaktadırlar. Bu durum, özellikle modelin kararlarının açıkça anlaşılması gereken durumlarda, örneğin tıbbi teşhislerde, ciddi bir sorun olabilmektedir (Samek vd., 2017).

3.3.3.3 Hesaplama Kaynakları

Derin öğrenme modellerinin eğitimi genellikle önemli miktarda hesaplama gücü gerektirmektedir. Bu durum hem donanım maliyetlerini artırabilmekte hem de modelin eğitim süresini uzatabilmektedir. Bu nedenle, derin öğrenme projelerinin başarısı, genellikle uygun donanım kaynaklarının ve verimli hesaplama tekniklerinin kullanılmasına bağlıdır (Sze vd., 2017).

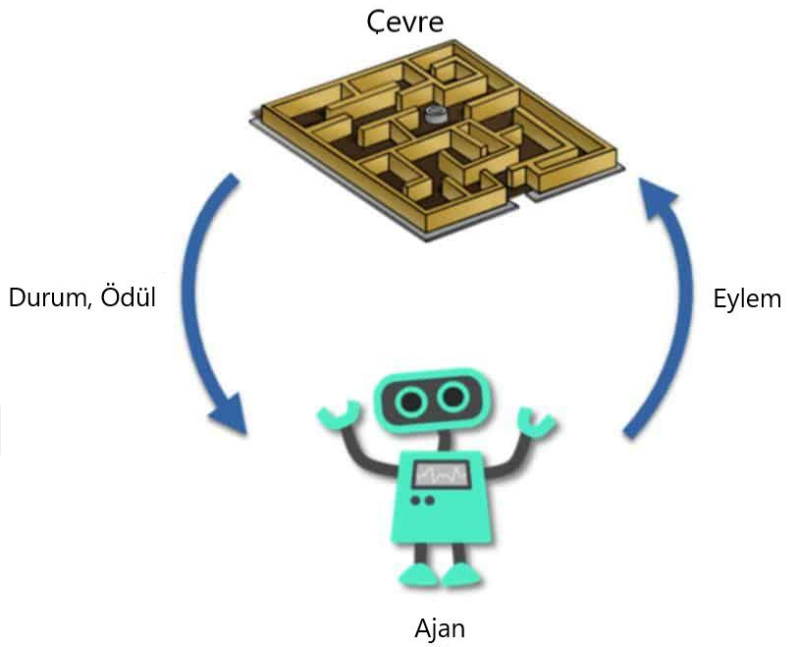
3.3.3.4 Aşırı Öğrenme (Overfitting)

Aşırı öğrenme, modelin eğitim verilerini çok iyi öğrenmesi ancak yeni verilere genelleme yapmakta zorlanması durumuna verilen addır. Bu, modelin gerçek dünyadaki performansını olumsuz etkileyebilmektedir. Aşırı öğrenmeyi önlemek için veri artırma ve düzenlileştirme gibi bazı teknikler kullanılabilmektedir. (Srivastava vd., 2014).

3.4 Pekiştirmeli Öğrenme

Denetimli öğrenme ve denetimsiz öğrenme konularına literatürde sıkça rastlanmaktadır. Fakat pekiştirmeli öğrenme ile yapılan çalışmalar daha yakın tarihe

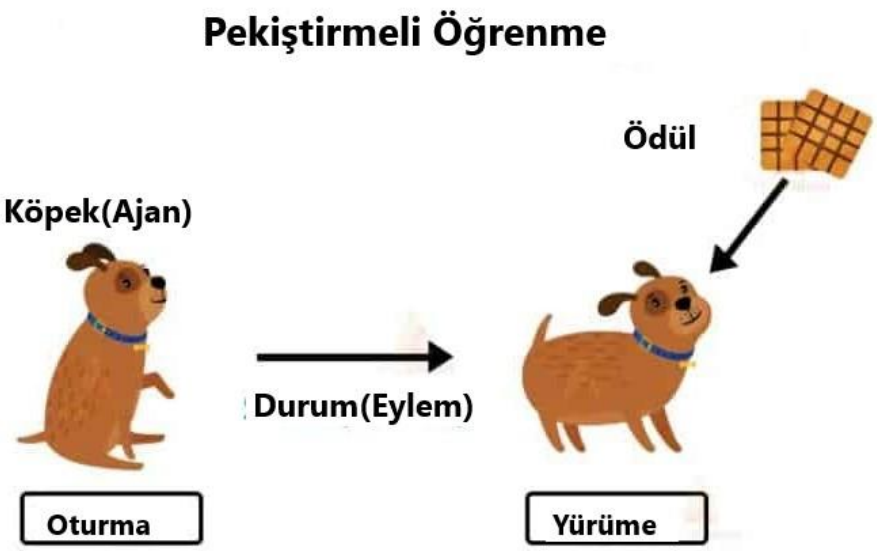
dayanmaktadır. Aslında pekiştirmeli öğrenme 1989 yılında keşfedilmiştir fakat uygulama alanı ve yapılan çalışmalar sınırlı olduğundan daha az bilinmektedir.



Şekil 3.8 Pekiştirmeli öğrenme döngüsü

Pekiştirmeli öğrenme, bir ajanın aksiyonlarından ve deneyimlerinden elde ettiği geri bildirimlerden yararlanarak daha çok deneme yanılma yoluyla algoritmanın verdiği en yüksek ödülü kazanmayı hedefleyen bir makine öğrenmesi tekniğidir. Şekil 3.8'de pekiştirmeli öğrenme döngüsü gösterilmektedir.

Artık günlük hayatımızda sıklıkla rastladığımız robot süpürgelerin ortamı tanıma yöntemi ve temizlik yaparken uyguladığı politika pekiştirmeli öğrenme ile geliştirilmiştir. Robot süpürge bir ortamda ilk defa çalıştırıldığında önce bulunduğu ortamda rastgele dolaşarak üzerinde bulunan sensörler yardımıyla da ortamı tanır ve bir harita çıkarır. Robot süpürge tekrar tekrar çalıştıkça ortamı her seferinde daha iyi tanıyarak hareket kabiliyeti artar ve istenilen bölgeye hızlıca erişir ve süpürür. Bu örnek pekiştirmeli öğrenmeye günlük hayatta verilecek en güzel örnektir.



Şekil 3.9 Reinforcement learning durum, aksiyon, ödül örneği

Şekil 3.9'da gösterildiği gibi bir köpek oturma durumunda iken komut verilerek yürümesi sağlanıyor. Köpek yürüdükten sonra bir yiyecekle ödüllendirilir. Köpek artık komut geldiğinde yürüme sonrasında elde edeceği ödülü bildiği için bu durumu öğrenmiş olacaktır.

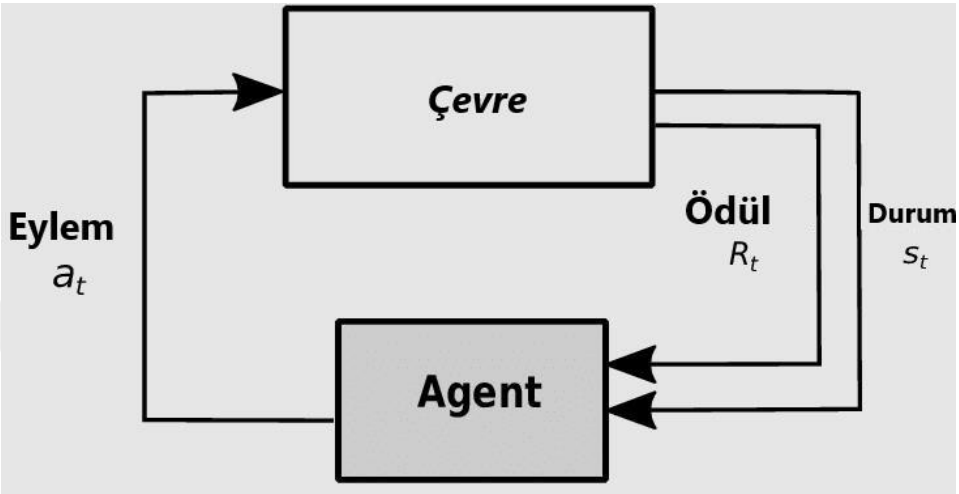
Pekiştirmeli öğrenme içerisinde birçok unsur barındıran bir yöntemdir. Bu unsurlar pekiştirmeli öğrenmeyi anlamamızı ve uygulamamızı kolaylaştıran unsurlardır. Bu unsurlardan en önemlisi ortam (environment) ve ajan (agent) olarak adlandıracağımız iki unsurdur. Robot süpürge örneğinde robot süpürgenin temizlediği ev ortam, robotun kendisi ajan olarak adlandırılır.

Aşağıda pekiştirmeli öğrenmede kullanılan unsurlar kısaca açıklanmıştır.

3.4.1 Pekiştirmeli Öğrenmede Kullanılan Unsurlar

- **Ajan (agent):** Ortam içerisinde aksiyonları gerçekleştiren ve öğrenen yapıya ajan diyoruz. Bu bazen bir robot bazen de bir algoritma olabilir.
- **Ortam (environment):** Ajanın içerisinde öğrendiği ortamdır.
- **Aksiyon (action):** Ajanın ortam içerisinde öğrenirken gerçekleştirdiği her türlü harekettir.

- **Ödül (reward):** Ajanın aksiyonlar sonucunda elde ettiği her türlü ödüldür. Bu ödül bazen başarı bazen de başarısızlık sonucunda verilebilir.
- **Durum (state):** Ajanın ortam içerisindeki anlık konumuna veya durumuna denir.



Şekil 3.10 Pekiştirmeli öğrenme temel unsurları

Şekil 3.10'da pekiştirmeli öğrenme unsurları, pekiştirmeli öğrenme döngüsüyle gösterilmektedir.

Pekiştirmeli öğrenme unsurlarıyla birlikte pekiştirmeli öğrenme çalışılırken bilinmesi gereken bazı kavramlar vardır. Bu kavramlar tam olarak anlaşılmadan pekiştirmeli öğrenme de anlaşılmayacaktır. Aşağıda pekiştirmeli öğrenme için kullanılan bazı kavramlar kısaca açıklanmıştır.

Politika (Policy): Ajanın birden çok aksiyon seçerken izlediği strateji olarak tanımlanabilir.

Değer Fonksiyonu (Value Function): Ajanın mevcut durum karşısında yürüttüğü politika sonucunda elde ettiği ödül değerleridir. Bu değerler ajanın öğrenmesinde büyük rol oynamaktadır.

Fonksiyon Yaklaşımcısı (Function Approximator): Eğitim kümesinden bir fonksiyon elde eden problemi ifade eder. Bu yaklaşımlardan bazıları karar ağaçları, sinir ağları ve en yakın komşu metodudur.

Ayrık Eylem Uzayı (Discrete Action Space): Sonlu bir aksiyon kümesinden hangi aksiyonun farklı gerçekleşeceğine karar verilirken ayrık eylem uzayı ile karar verilir.

Devamlı Eylem Uzayı (Continuos Action Space): Aksiyonların ifade edildiği tek değerli vektördür.

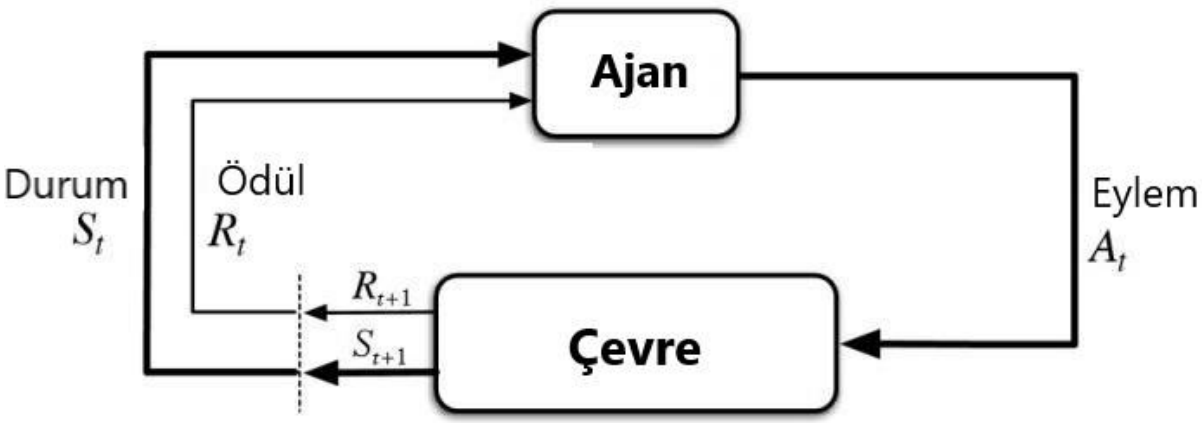
Model: Ajanın ortamdaki bakış açısı olarak değerlendirilebilir. Durum-aksiyon çiftlerinin olasılık dağılımlarını haritalar.

Dinamik Programlama (Dynamic Programming): Bir çözüm metodu sınıfıdır. Sıralı karar problemleri için kullanılır.

Monte Carlo Yöntemleri (Monte Carlo Methods): Mevcut durumdan başlayarak birçok deneme sonucu tahminde bulunan ve elde edilen ödüllerin ortalamasıyla değer işlevlerini öğrenmek için kullanılan bir yöntem sınıfıdır.

Geçici Fark Algoritmaları (Temporal Difference (TD) algorithms): Tahminleri geçici olarak karşılaştırıp bir sonuç üreten bir öğrenme yöntemi sınıfıdır.

Markov Karar Süreçleri (Markov Decision Process (MDP): Markov Karar Süreci, ayrık zamanlı bir stokastik kontrol süreci olarak tanımlanır. Bir ajanın seçim yapmasına yardımcı olan bir çerçevedir. Şekil 3.11'de gösterilen pekiştirmeli öğrenme döngüsünü markov karar süreci ile ifade etmek mümkündür.



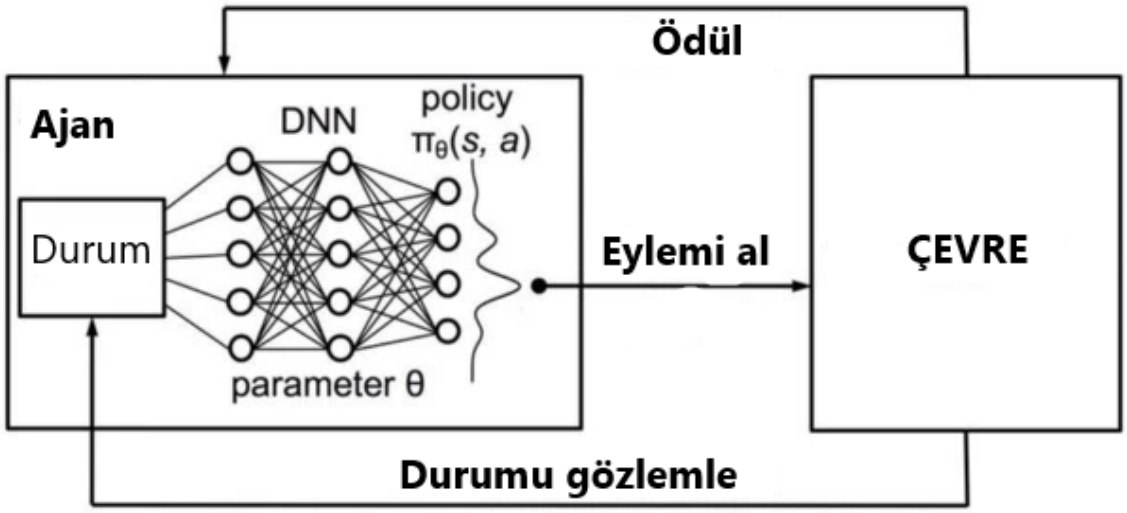
Şekil 3.11 Pekiştirmeli öğrenmenin Markov Karar Süreci ile gösterimi

Son yıllarda derin öğrenmenin bir alt dalı olan Derin Pekiştirmeli Öğrenme alanı araştırmacılar ve kitle iletişim araçları tarafından en çok konuşulan yapay zekâ

alanlarından biri olmuştur. En büyük kazanımlar derin pekiştirmeli öğrenme sayesinde gerçekleştirilmiştir. Google tarafından geliştirilen Alpha Go'nun, Go oyununda imkânsız olarak düşünülen dünya rekortmenini yenmesi derin pekiştirmeli öğrenmenin yapabileceklerini ortaya koymuştur.

3.5 Derin Pekiştirmeli Öğrenme

Derin pekiştirmeli öğrenme, ortam ile doğrudan etkileşime girerek kendi kendine öğrenen bir algoritma olarak tanımlanabilir. Ortam, bir bilgisayar oyunu, bir simülasyon, stranç gibi bir masa oynuu veya gerçek dünya olabilir. Ajan, ortam ile etkileşime girerek yapılan eylemlerin sonuçlarını öğrenir.



Şekil 3.12 Derin pekiştirmeli öğrenme şeması

Derin pekiştirmeli öğrenmede Ajan bir yapay sinir ağı ile temsil edilir. Yapay sinir ağı ortam ile doğrudan etkileşime girer. Ajan ortamın mevcut durumunu gözlemler. Ajan mevcut durum ve geçmiş deneyimlerle birlikte hangi eylemin gerçekleştirileceğine karar verir. Alınan eyleme göre ajan bir ödül alır. Ödülün miktarı, ajanın çözmesi gereken sorunla ilgili yapılan eylemin başarısını belirler. Ajanın amacı, biriken ödülleri zaman içerisinde maksimize edecek eylemleri almayı öğrenmektir.

3.5.1 Markov Karar Süreçleri

Markov karar süreci, ayrık zamanlı bir stokastik kontrol süreci olarak tanımlanır. Markov karar süreci, bir ajanının karmaşık bir ortamı modellemek için kullanılan şimdiye kadar sahip olduğumuz en iyi yaklaşımdır. Ajanın çözmeye çalıştığı her sorun, *S1, S2, S3,...* durumların bir sırası olarak düşünülebilir . Bir durum ajanın ortam içerisindeki anlık konumudur. Örneğin bir satranç tahtasının o anki konfigürasyonu olabilir. Ajan eylemleri alarak bir durumdan diğerine geçerek öğrenme sürecine devam eder.

3.5.1.1 Markov Süreçleri

Bir Markov süreci, mevcut durumun yalnızca önceki duruma bağlı olduğu olası durumlar dizisini tanımlayan bir stokastik modeldir. Buna Markov mülkiyeti denir. Eşitlik 3.1'de görüldüğü gibi pekiştirmeli öğrenmede ajanının bir sonraki durumu, sadece önceki bir duruma değil önceki tüm durumlara bağlıdır.

$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, S_2, S_3, \dots, S_t] \tag{3.1}$$

Markov süreci stokastik bir süreçtir. Eşitlik 3.2'de *s* mevcut durumu *s'* ise bir sonraki durumu temsil eder. Mevcut durumdan bir sonraki duruma geçiş sadece belirli bir olasılık ile gerçekleşebilir.

$$\mathcal{P}_{SS'} = \mathbb{P}[S_{t+1} = s'|S_t = s] \tag{3.2}$$

P Eşitlik 3.3'de gösterildiği gibi, tüm durumların geçiş olasılıklarının yer aldığı geçiş olasılık matrisidir.

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{P}_{n1} & \cdots & \mathcal{P}_{nn} \end{bmatrix} \tag{3.3}$$

3.5.1.2 Markov Ödül Süreci

Markov Ödül Süreci, <S, P, R> kümesinden oluşur. Eşitlik 3.4'de görüldüğü gibi R_s, ajanın s durumunda almasını beklediği ödüldür.

$$\mathcal{R}_S = \mathbb{E}[\mathcal{R}_{t+1}|S_t = s] \tag{3.4}$$

Eşitlik 3.5'de görüldüğü gibi her adımda alınan ödüller "indirim faktörü (discount factor)" olarak adlandırılan γ ile ağırlıklandırılır. Bu değer 0 ile 1 arasında olmalıdır. Bu değer gelecekte kazanılacak muhtemel ödüllerin önemini belirlememizde yardımcı olmaktadır. Eğer bu değer 0 seçilirse o anda alınan ödül önemli olur gelecekteki ödüllerin önemsiz olduğu anlamına gelir. Değer 1'e yaklaştıkça gelecekteki muhtemel ödüllerin toplamının mevcut adımdaki önemi artmış olur. Eşitlik 3.5'de toplam beklenen ödül gösterilmektedir.

$$\mathcal{G}_t = \mathcal{R}_{t+1} + \gamma\mathcal{R}_{t+2} + \gamma^2\mathcal{R}_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k \mathcal{R}_{t+k+1} \tag{3.5}$$

3.5.1.3 Değer Fonksiyonu

Derin pekiştirmeli öğrenmede diğer önemli kavramlardan biri de değer fonksiyonudur. V(s) olarak gösterilir. Değer fonksiyonu her durum için bir değer oluşturur. Bir s durumundaki değer, ajanın s durumuna ilerlemesi durumunda alacağı toplam ödülü gösterir.

$$v(s) = \mathbb{E}[G_t|S_t = s] \tag{3.6}$$

Eşitlik 3.6, s durumunda değer fonksiyonu.

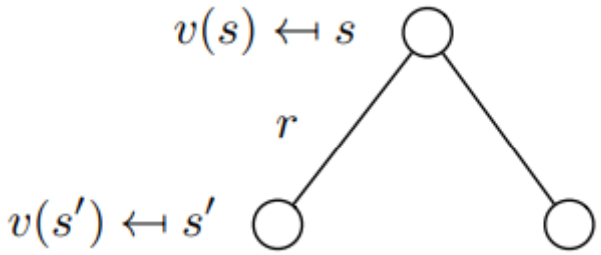
Eşitlik 3.5'deki G_t değeri Eşitlik 3.6'da yerine yazılırsa Eşitlik 3.7 elde edilir.

$$
\begin{aligned}
v(s) = \ & \mathbb{E}[G_t | S_t = s] \\
& = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s] \\
& = \mathbb{E}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \cdots) | S_t = s] \\
& = \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s] \\
& = \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) | S_t = s]
\end{aligned}
\tag{3.7}
$$

3.5.2 Bellman Denklemi

3.5.2.1 Markov Ödül Süreçlerinde Bellman Denklemi

Eşitlik 3.7'de elde edilen v(s) eşitliği aynı zamanda markov ödül işlemlerinde Bellman Denklemi olarak da adlandırılır.



Şekil 3.13 s durumundan s' durumuna stokastik geçiş

Bu eşitlik Şekil 3.13'de gösterildiği gibi s durumundan s' durumuna geçiş halinde kullanılır.

$$
v(s) = \ \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) | S_t = s] \tag{3.8}
$$

Eşitlik 3.8'de belirtilen S_{t+1} durumu birden fazla farklı s' durumlarını içerebilir. Örneğin ajan hem sola hem sağa hem de yukarı hareket edebilir durumda olabilir. Bu durumda her s' durumu için belirlenen Pss' olasılıkları ile v(s') değerinin çarpılıp toplanarak ödüle eklendiğinde Eşitlik 3.9'daki gibi bir eşitlik elde edilir.

$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in S} \mathcal{P}_{ss'} v(s') \tag{3.9}$$

Eşitlik 3.9, Bellman Eşitliği

3.5.2.2 Markov Karar Süreci

Markov karar süreci kararları olan bir Markov ödül sürecidir. Markov karar süreci ajanın bir durumda gerçekleştirebileceği muhtemel eylemlerin sınırlı olduğu <S, A, P, R> dizisinden oluşmaktadır. Bir s durumundaki ödülü almak ajanın bu durumda gerçekleştireceği eyleme de bağlıdır. Eşitlik 3.10'da bir durumdaki eyleme bağlı olarak beklenen ödülü göstermektedir.

$$\mathcal{R}_s^a = \mathbb{E}[\mathcal{R}_{t+1} | S_t = s, A_t = a] \tag{3.10}$$

3.5.2.3 Politikalar

Ajanın belirli bir durumda hangi eylemi gerçekleştirmesi gerektiği bir politika (π) tarafından belirlenir. Politikalar bir ajanın s durumundan bir sonraki durumuna geçebilmesi için yapacağı a eylemini belirler. Eşitlik 3.11'de s durumunda ajanın gerçekleştireceği a eyleminin politikası gösterilmektedir.

$$\pi(a|s) = \mathbb{P}[A_t = a | S_t = s] \tag{3.11}$$

Politika bir ajanın mevcut durumda belirli eylemleri seçmek için uygulanan strateji olarak tanımlanabilir. Belirli bir politika uygulandığında, v(s) durum değer fonksiyonu Eşitlik 3.12'deki gibi tanımlanır.

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] \tag{3.12}$$

3.5.2.4 Eylem – Değer Fonksiyonu

Durum – değer fonksiyonu v(s) kadar önemli olan diğer bir fonksiyon eylem-değer fonksiyonu olarak adlandırılan q(s,a) fonksiyonudur. Eylem – değer fonksiyonu bir politikayı takip ederek s durumunda a eylemini gerçekleştirerek elde edeceğimiz beklenen ödül olarak tanımlanabilir. Bir ajanın bir durum içerisinde yapabileceği birden fazla eylem olabilir. Derin pekiştirmeli öğrenmede Q(s,a) değeri bir yapay sinir ağı tarafından hesaplanır. Yapay sinir ağı girdi olarak verilen durum sonucunda her bir eylem için bir Q(s,a) değeri hesaplar. Elde edilen en yüksek değer hangi eylem için bulunmuşsa o eylem gerçekleştirilir. Eşitlik 3.13'de Eylem-değer fonksiyonunu görmekteyiz.

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] \tag{3.13}$$

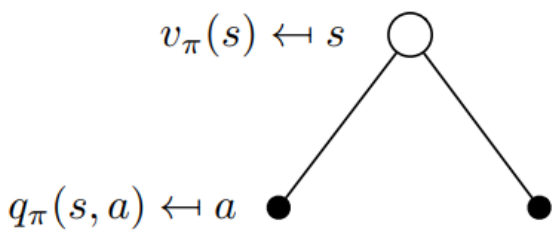
Belirli bir politikaya bağlı durum-değer fonksiyonu Eşitlik 3.14'te görüldüğü formda yazılabilir.

$$v_\pi(s) = \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s] \tag{3.14}$$

Aynı işlem eylem-değer fonksiyonu için uygulandığında Eşitlik 3.15 elde edilir.

$$q_\pi(s,a) = \mathbb{E}[\mathcal{R}_{t+1} + \gamma q_\pi(s_{t+1}, a_{t+1})|S_t = s, A_t = a] \tag{3.15}$$

Şekil 3.14'de v(s) ile q(s,a) arasındaki ilişki gösterilmektedir.

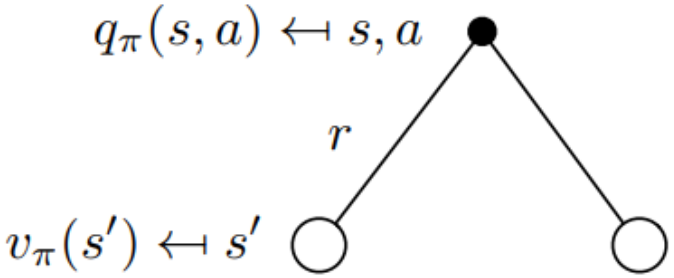


Şekil 3.14 v(s) ve q(s,a) arasındaki ilişki

Şekil 3.14'te belirli bir s durumunda olmanın olası iki eylemi gerçekleştirme ihtimali olduğu gösterilmektedir. Belirli bir durumda belirli bir eylemi gerçekleştirme sonucunda eylem değeri olan q(s,a) elde edilir. O halde durum değer fonksiyonu v(s) politika dışında olmayan s durumunda a eyleminin gerçekleşme olasılığının ağırlığına bağlı olan muhtemel q(s,a) değerlerinin toplamı olarak yazılabilir. Eşitlik 3.16'da gösterilmektedir.

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) q_\pi(s,a) \tag{3.16}$$

Şimdi Şekil 3.14'teki tam tersi durumu düşünelim. İkili ağacın kökünde belirli bir a eylemini seçtiğimiz durum bulunmaktadır. Markov süreçlerinin stokastik olduğu unutulmamalıdır. Bir eylemde bulunmak olmak istediğimiz yere yüzde yüz varacağımız anlamını vermez. Eyleme geçtikten sonra diğer durumlardaki ihtimaller de değerlendirilmelidir. Şekil 3.15'de verilen örnekte iki farklı sonraki durum(s') olduğu görülmektedir.



Şekil 3.15 q(s,a) ile v(s') arasında olası durumlar

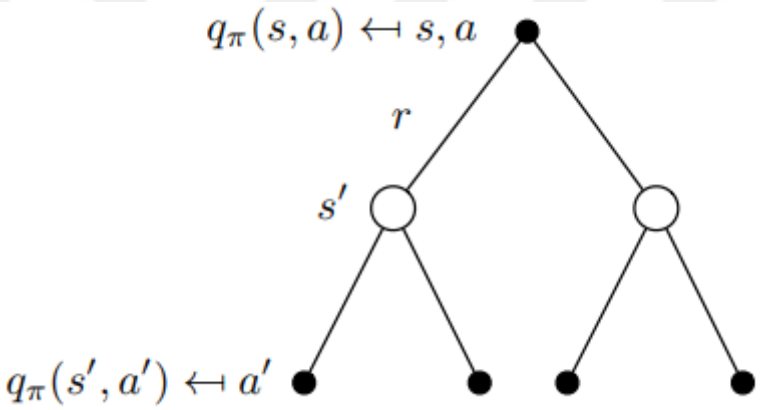
Eylem-durum değerini elde etmek için olası tüm durumlara ulaşma olasılığının durum değerinin çarpımlarının toplanıp discount faktörüyle birlikte ödüle eklenmelidir. Eşitlik 3.17'de gösterilmiştir.

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{ss'} v_\pi(s') \tag{3.17}$$

Eşitlik 3.17'deki v(s') değeri yerine Eşitlik 3.16'da belirtilen açılımı yazıldığında Eşitlik 3.18 elde edilir. Bu eşitlikten anlaşılacağı üzere q(s,a) ile q(s',a') arasında özyinelemeli bir ilişki olduğu görülür.

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{ss'} \sum_{a' \in A} \pi(a'|s') q_\pi(s',a') \tag{3.18}$$

Bu özyinelemeli ilişki Şekil 3.16'da bir ağaç yapısı üzerinde gösterilmiştir. Q(s,a) ile başlayan ağacımız bir sonraki s' durumuna belirli Pss' olasılıklarıyla a' eylemini gerçekleştirip q(s',a') durumunda son buluyoruz. Q(s,a) değerini elde etmek için ağaçtan yukarı doğru çıkıp tüm olasılıkları birleştirmemiz gerekmektedir.



Şekil 3.16 q(s,a) ile q(s',a') arasındaki özyinelemeli ilişkiyi gösteren ağaç yapısı

3.5.2.5 Optimal Politika

Derin pekiştirmeli öğrenmede en önemli konu optimal eylem-değer fonksiyonu $q_*(s,a)$'yı bulmaktır. $q_*(s,a)$'nın bulunması ajanın herhangi bir durumda eylemlerin kalitesini bildiği anlamına gelir. Böylelikle ajan hangi eylemin yapılması gerektiğine karar verebilir. Eşitlik 3.19'da görüldüğü gibi q_*, eylem-değer fonksiyonunu maksimize eden politikayı izleyen fonksiyondur.

$$q_*(s,a) = \max_\pi q_\pi(s,a) \tag{3.19}$$

Mümkün olan en iyi politikayı bulmak için a eylemini tüm olası eylemlerden q(s, a) değerinin en yüksek olduğu durumda seçmemiz gerekir. Bu da optimal politika için Eşitlik 3.20'yi verir.

$$\pi_*(s,a) = \begin{cases} 1, & a = arg\max_a q_*(s,a) \\ 0, & değilse \end{cases} \tag{3.20}$$

3.5.2.6 Bellman Optimallik Denklemi

Optimal politikanin koşullu eşitliği Eşitlik 3.18'e eklendiğinde Bellman Optimallik Denklemi elde edilir. Eşitlik 3.21'de gösterilmiştir.

$$q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{ss'} \max_{a'} q_*(s',a') \tag{3.21}$$

Bellman optimallik denklemiyle birlikte artık ajan herhangi bir durumda olası eylemlerin kalitesini bilecek ve uygun olan eylemi uygulayabilecektir.

3.5.3 Derin Q-Öğrenme

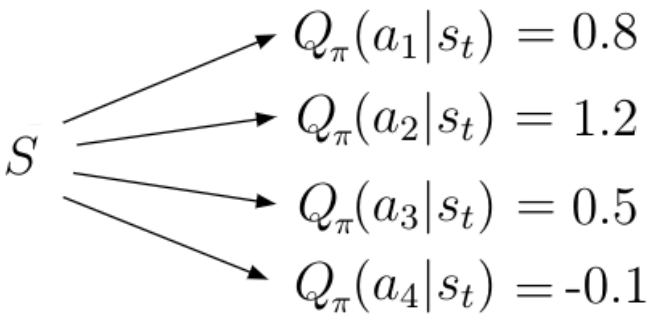
3.5.3.1 Eylem – Değer Fonksiyonu

Daha önce de anlattığımız gibi eylem – değer fonksiyonu bir ajanın s durumundan başlayarak π politikasını izleyerek gerçekleştireceği a eyleminden elde edeceği beklenen ödül olarak tanımlanır.

$$\begin{aligned} Q_\pi(s,a) &= \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \\ &= \mathbb{E}_\pi[r + \gamma Q_\pi(s_{t+1}, a_{t+1}) | S_t = s, A_t = a] \end{aligned} \tag{3.22}$$

Eşitlik 3.22, Eylem-değer fonksiyonu.

Q(s, a), bir s durumunda muhtemel eylemlerin kalitesini verir. En yüksek değerli a eylemi en uygun eylem demektir.



Şekil 3.17 Örnek bir S durumunda farklı a eylemleri ve Q değerleri

Beklenen değerini Eşitlik içerisinde yerine koyarsak olasılıkları da göz önünde bulundurduğumuz eylem değer fonksiyonunun yeni bir formunu elde ederiz (Eşitlik 3.23). Burada Pss' ortam içerisinde belirlenen s durumundan bir sonraki s' durumuna geçiş olasılığını göstermektedir. π (a '| s') politikayı göstermektedir.

$$Q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{ss'} \sum_{a' \in A} \pi(a'|s') Q_\pi(s',a') \tag{3.23}$$

3.5.3.2 Zamansal Fark (Temporal Difference) Öğrenme Algoritması

Derin Q Öğrenme'deki amacımız eylem-değer fonksiyonunu çözmektir. Çünkü ajan bu değeri bilirse herhangi bir durum için hangi eylemi gerçekleştireceğini bilir ve problemimiz çözülmüş olur. Eşitlik 3.18 bize Q(s, a) için özyinelemeli bir çözüm sunmaktadır. Fakat özyinelemeli bir çözüm çok da pratik bir çözüm değildir. Bunun yerine Zamansal Fark Öğrenme algoritması kullanılabilir.

Bu algoritmada bir s durumunda a eylemi gerçekleştiğinde $Q(s_t, a_t)$ değeri, TD-Hedef olarak adlandırılan $R_{t+1} + \gamma Q(s_{t+1}, a_{t+1})$ tahmini getiri ile $Q(s_t, a_t)$ arasındaki fark α olarak tanımlanan öğrenme oranı(learning rate) parametresi ile ağırlıklandırılıp $Q(s_t, a_t)$ ile toplanıp güncellenir. Bu işlem her durum-eylem çifti için uygulanır. Bu algoritma SARSA algoritması (State-Action-Reward-State-Action) olarakta bilinir.

Öğrenme oranı, discount rate(γ) gibi bir hiper parametredir. Bu değer zamansal fark ile yapılan güncelleme adımını belirler. α değeri 0 olarak seçilirse $Q(s_t, a_t)$ değerimizin güncellenmediği anlamına gelir. α değeri 1' yaklaştıkça $Q(s_t, a_t)$ değerinin güncelleme miktarı da artmış olacaktır.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, a_{t+1}) - Q(s_t, a_t)) \quad (3.24)$$

Eşitlik 3.24, Q(s, a) güncelleme kuralı.

3.5.3.3 Sarsa

Q(s, a)'ya uygulanan Zamansal Fark Algoritması genellikle SARSA (State-Action-Reward-State-Action) algoritması olarak bilinir. SARSA izlenen politikanın değerini tahmin eden bir politika içi pekiştirmeli öğrenme algoritmasıdır.

Bu algoritmada, ajan en uygun politikayı kavrar ve eylem seçmek için kullanır. Güncelleme için kullanılan politika ile eylemi gerçekleştirmek için kullanılan politika, Q-learning'in aksine aynıdır. Bu, politikaya dayalı öğrenmeye bir örnektir.

SARSA'daki bir deneyim ⟨S,A,R,S', A' ⟩ şeklindedir, yani

S: Mevcut durum,
A: Geçerli eylem,
R: Ödül,
S': Yeni durum,
A': Gelecekteki eylem.

SARSA, Eşitlik 3.25'deki Q(s, a) güncellemesi için yeni bir deneyim sağlar.

$$Q(s, a_1) = R + \gamma Q(s', a') \quad (3.25)$$

3.5.3.4 Q-Öğrenme

Pekiştirmeli öğrenme, bir ortamdaki bir aracının ödülleri en üst düzeye çıkarmak için gerçekleştirdiği eylemle ilgilenen makine öğreniminin bir parçasıdır. Pekiştirmeli Öğrenme, denetimli bir giriş/çıkış çiftine ihtiyaç duymadığından denetimli öğrenme ve denetimsiz öğrenmeden farklıdır. Ve eylemi daha etkili hale getirmek için herhangi bir durumda yüksek miktarda düzeltme gerektirmez.

Tüm pekiştirmeli öğrenme konularında, Markov Karar Süreci (MDP) çok büyük bir rol oynar; Dikkat edilmesi gereken önemli bir nokta, ortamda sunulan her durumun, önceki durumundan kaynaklandığı ve bu durumun da önceki durumunun bir sonucu olduğudur. Böylece herhangi bir ortamın mevcut durumu, önceki durumlardan toplanan bilgilerin bileşiminden kaynaklanır.

Bir ajanın amacı, gerçekleştireceği eylem ile ortam tarafından verilen ödülü artırmaktır. Markov Karar Süreci, bir ajanın, ödülü en üst düzeye çıkarmak için belirli bir durumda en uygun eylemi seçmeye karar vermesini sağlar. Bir durumdan belirli bir zamanda eylem seçme olasılığına politika denir. Dolayısıyla, Markov Karar Sürecinin amacı en uygun politikayı bulmaktır.

Q-Öğrenme'de aslında iki politika izlenmektedir. Bunlar μ (a | s) davarnış politikası ve π (a | s) hedef politikasıdır. Davranış politikası $Q(s_t, a_t)$'nin hesaplamasında kullanılırken, hedef politika sadece zamansal fark başlığı altında anlatılan TD-Hedef değerinin hesaplanması için eylemleri belirler. Aslında Q-Öğrenme algoritması hedef politikanın (π (a | s)) açgözlü olduğu özel bir durumdan ibarettir. Yani stratejimiz Q1nun en yüksek değerini oluşturan eylemlerde bulunmaktır. Bu durum aşağıdaki eşitlikte yer alan politikayı ortaya çıkarmaktadır.

$$\pi(s,a) = \begin{cases} 1, & a = argmax_a Q(s,a) \\ 0, & dd \end{cases} \tag{3.26}$$

Hedef politika bu durumdan dolayı Açgözlü Politika olarak adlandırılır. Açgözlü politika en yüksek Q(s, a) değerini elde edecek eylemlerin seçildiği anlamına gelir. Açgözlü politika daha önce rastgele uygulanan politikanın kullanıldığı eşitlikte yerine yazılırsa Eşitlik 3.27 elde edilir.

$$
\begin{aligned}
Q_\pi(s,a) &= R_s^a + \gamma \sum_{s' \in S} P_{ss'} \sum_{a' \in A} \pi(a'|s') Q_\pi(s',a') \\
&= \mathbb{E}_\pi[r + \gamma Q_\pi(s_{t+1}, a_{t+1}) | S_t = s, A_t = a] \\
&\downarrow \pi(s_t) = arg\max_{a'} Q(s_{t+1}, a') \\
Q_*(s,a) &= R_s^a + \gamma \sum_{s' \in S} P_{ss'} \max_{a'} Q_*(s',a') \\
&= \mathbb{E}_\pi[r + \gamma \max_{a'} Q_*(s_{t+1}, a') | S_t = s, A_t = a]
\end{aligned}
\tag{3.27}
$$

Açgözlü politika bize optimal Q∗(s, a) değerini sağlar. Bu politika Q∗(s, a) tanımına göre en yüksek Q(s, a) değerini verir(Eşitlik 3.28).

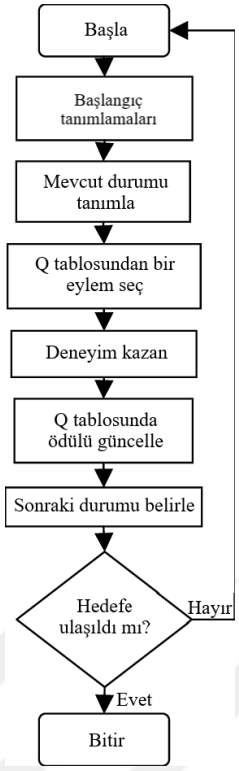
$$
Q_*(s,a) = \max_\pi Q_\pi(s,a) \tag{3.28}
$$

Eşitlik 3.27'de son satırda elde ettiğimiz eşitliğin Bellman optimallik denklemi olduğunu daha önceki konularda ele almıştık. Bu eşitlik optimal eylem-değer fonksiyonunu (Q∗(s, a)) tahmin etmek için özyinelemeli bir güncelleme kullanır.

Zamansal fark öğrenmesi hala Q∗(s, a)'yı elde etmek için en iyi yoldur. Açgözlü hedef politikası ile zamansal fark öğrenmesi güncellemesiyle birlikte aşağıdaki gibi yazılabilir.

$$
Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a_{t+1}) - Q(s_t, a_t)) \tag{3.29}
$$

Şekil 3.18'de, Q-Öğrenme algoritmasının dört ana süreç ve iki alt süreç izlediğini görebiliriz. Bu süreçler şu şekildedir:



Şekil 3.18 Q-öğrenme algoritması akış şeması

1) **Başlangıç tanımlamaları**: Bu adımda model, ajanın çalışacağı ortam, zaman ve bu ortamdaki durum ve eylemleri tanımlanır. Dolayısıyla Q tablo bu adımda tanımlanır.
2) **Mevcut durumu tanımla**: Bir ajanın, maksimum ödülleri kazanmak üzere en iyi şekilde hareket etmesi için önceki kayıtları saklaması gerekir. Mevcut bir durumda hareket etmek için, durumun tanımlanması ve bir dizi eylem gerçekleştirmesi gerekir. Mevcut durumun tanımlanması bu adımda gerçekleşir.
3) **Bir eylem seçin ve deneyim kazanın**: Başlangıç tanımlamaları adımında, eylemler ve durumların kombinasyonu hakkında bilgilerin yer aldığı bir Q-tablosu oluşturulmuştu. Bu tablo üzerinden geçmiş deneyimlere bakılır ve ağırlığı karşılaştırılır. Eğer yeni bir durumsa, Q-tablo bir sonraki adım için bunu güncelleyecektir.
4) **Ödülü Q-tablosunda güncelleyin ve bir sonraki durumu belirleyin**: Ajanlar, deneyim kazandıktan sonra ortamdan bir ödül alırlar. Bu ödülün

değeri, Q-tablosuna deneyim verileri olarak kaydedilir. Bu kayıtlar bir sonraki adımdaki eylemleri tahmin etmede yardımcı olacaktır.

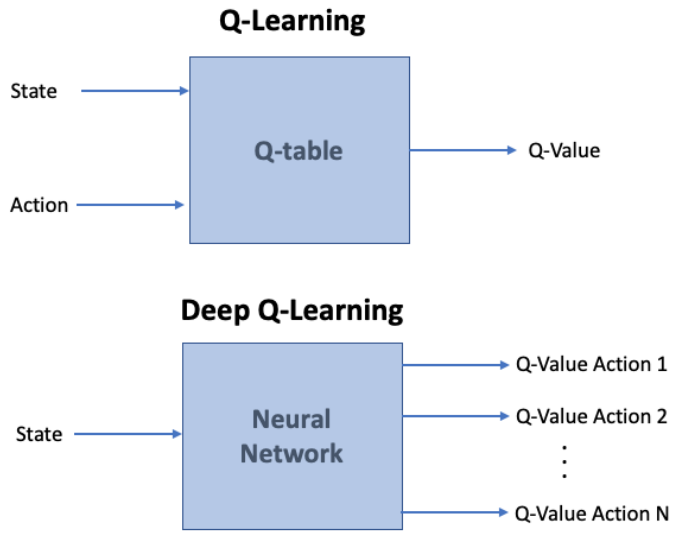
Q-Öğrenme'de, bir durum için muhtemel tüm eylemlerin durum-eylem değeri şeklinde Q(s, a) değerlerini öğreniyoruz. Bir ajan rastgele bir durum için diyelim ki üç eylem gerçekleştirebiliyor. Bu eylemlerin her biri için üç farklı değer hesaplanacak ve her değer Q tablosunda güncellenecektir. Eşitlik 3.30'de de görüleceği gibi Q tablosunun her bir zaman damgası için söz konusu eylem yapıldığında elde edilecek ödüllere göre güncellendiği görülmektedir.

$$Q(s,a) = r(s,a) + \gamma \max_{a} Q(s',a) \tag{3.30}$$

Q öğrenme konusu anlatılırken verilen örnekler genellikle kısıtlı durum ve az sayıda eyleme sahiptirler. Fakat gerçek hayatta durumlar 10.000'den fazla eylemlerin sayısı da 1000000 olabilir. Bu durumda, Q-tablosunun boyutu çok büyük olacaktır ve bu durumda modelin öğrenmesi çok zaman alacaktır. Böyle bir durumda Derin Q Öğrenme devreye girecektir.

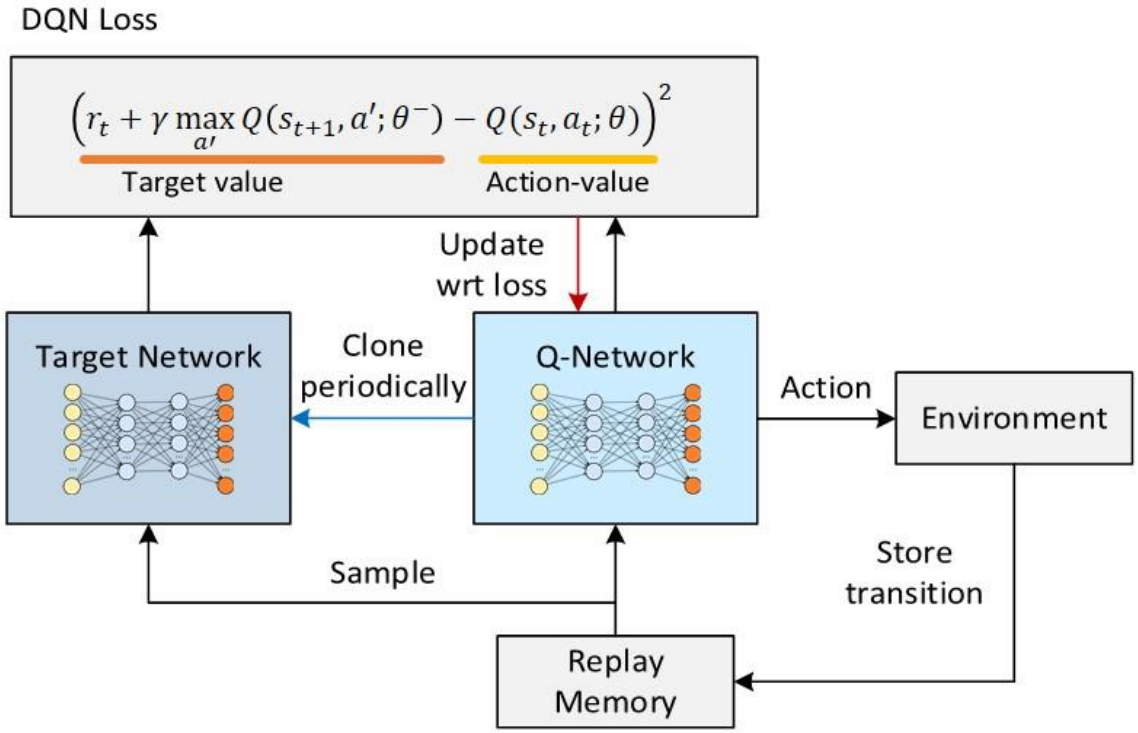
3.5.3.5 Derin Q-Öğrenme

Derin Q-Öğrenme'de, en uygun Q-değerini bulmak için Q-Öğrenme'yi bir sinir ağıyla birleştirilmektedir. Derin Q-Öğrenme algoritmasında, durumları girdi olarak verilmekte, tüm olası eylemlerin optimal Q-değerleri de çıktı olarak verilmektedir. Q-öğrenme ve Derin Q-öğrenme teknikleri arasındaki fark Şekil 3.19'da gösterilmektedir.



Şekil 3.19 Q-öğrenme ve Derin Q-öğrenme teknikleri arasındaki fark

Derin Q-Öğrenme'de kullanıcı, tüm geçmiş deneyimleri ve Q-ağ'ın çıktısı tarafından tanımlanan gelecekteki eylemi hafızasında saklar. Böylece Q-ağ, s_t durumundaki Q-değerini kazanır. Aynı zamanda hedef ağ (Neural Network), S_{t+1} durumu (sonraki durum) için Q-değerini hesaplayarak eğitimi stabilize eder ve Q-değeri sayısındaki ani artışları, Q-değerini Q-ağının yinelenen her bir Q-değerine eğitim verisi olarak kopyalayarak, engeller.



Şekil 3.20 Q-Ağ (Q-Network) ve Hedef Ağ (Target Network)'ın Derin Q- Öğrenme döngüsündeki yeri ve Derin Q-Ağındaki(DQN) Kayıp Fonksiyon

Daha önce de belirtildiği gibi, temsilcinin önceki deneyimleri toplaması ve depolaması gerekir. Derin Q-Öğrenme'de, sinir ağı ve Q-Öğrenme birlikte kullanılır. Bu sayede sinir ağı, deneyimleri <Durum, Sonraki durum, Eylem, Ödül> biçiminde bellekte saklamaktadır. Önceki verilerden rastgele bir grup seçildiğinde sinir ağı eğitiminin kararlılığının arttığı kanıtlanmıştı. Derin Q-Öğrenme ajanın performans düzeyini artırmak için Deneyim Tekrarı (Experience Replay) olarak adlandırılan bir kavram daha kullanır. Bu kavram önceki deneyimleri depolayıp kullanarak ajanın performansını artırmak için kullanılır.

Hedef ağ, eğitim için deneyim tekrarını kullanır ve Q-ağ, bunu Q-değerinin hesaplanması için kullanır. Kayıp fonksiyonu, hedeflenen Q-Değerinin ve tahmin edilen Q-Değerinin karesi alınmış farkıyla hesaplanır (Eşitlik 3.31).

$$Loss = (r + \gamma max_{a'} Q(s', a') - Q(s, a))^2 \quad (3.31)$$

Bu işlem, yalnızca parametrelerin hedef ağa kopyalanmasından hemen önce Q-Network eğitimi için gerçekleştirilir.

Derin Q Öğrenme ve Deneyim tekrarı algoritma detayları Algoritma 1'de gösterilmiştir.

Algoritma 1 Derin Q Öğrenme ve Deneyim Tekrarı

N kapasitesinde D tekrar belleğini başlat
Eylem-değer fonksiyonunu Q rastgele ağırlıklarla başlat
for iterasyon = 1 to M **do**
    Durumu st ve ön işleme sırasını başlat $\phi 1 = \phi\{s1\}$
    for iterasyon = 1 to T **do**
        ε olasılığına göre rastgele bir eylem at seç
        değilse at = maxaQ∗ (ϕ(st), a; θ)
        Ortamda eylemi at yürütün ardından rt ve st yi gözlemleyin
        Değişimi (st , at , rt , st+1)D de saklayın
        D den bir mini toplu geçiş örneği (sj , aj , rj , sj+1) alın
        $y_i = \begin{cases} r_j, & \text{terminal durumu için} \\ r_j + \gamma max_{a'} Q(s_{j+1}, a'\,;\,\theta), & \text{terminal olmama durumu için } s_{j+1} \end{cases}$
        θ yı göz önünde bulundurarak (yj − Q(sj , aj ; θ))2 üzerinde bir gradyan adımı gerçekleştirin
    end for
end for

3.6 Blok Zinciri (Blockchain) Teknolojisi

Blok zinciri, dijital bilgilerin güvenli ve şeffaf bir şekilde saklanması ve aktarılması için kullanılan dağıtık bir veri tabanı teknolojisidir (Nakamoto, 2008). Temel olarak, birbiriyle bağlantılı bloklardan oluşan bir veri yapısıdır. Her blok, işlemler, zaman damgası ve önceki bloğa bir referans içeren bilgileri saklar (Merkle, 1987). Bu yapı, verilerin değiştirilmesi veya hile yapılmasını zorlaştırarak güvenliği sağlar.

Blok zinciri, aşağıdaki prensipler üzerine kuruludur:

Dağıtık sistem: Blok zinciri, birden fazla bilgisayar veya düğüm tarafından yönetilen ve paylaşılan bir ağ üzerinde çalışır (Swan, 2015). Bu, merkezi bir otoriteye

gerek olmaması ve herhangi bir düğümün başarısız olması durumunda bile sistemin devam etmesi anlamına gelir.

Konsensus algoritması: Blok zinciri ağındaki düğümler, bir işlemi doğrulamak ve kabul etmek için belirli konsensus algoritmaları kullanır (Bonneau vd., 2015). Bu algoritmalar, ağdaki düğümlerin güvenli ve uyumlu bir şekilde çalışmasını sağlar. En popüler konsensus algoritmalarından biri Proof of Work'tür, ancak Proof of Stake, Delegated Proof of Stake ve diğerleri gibi alternatifler de vardır.

Kriptografik güvenlik: Blok zinciri, verilerin güvenliğini sağlamak için kriptografi kullanır (Diffie ve Hellman, 1976). Her blok, önceki bloğun kriptografik özeti (hash) ile bağlantılıdır. Bu, eğer biri geçmişteki bir işlemi değiştirmeye çalışırsa, bu değişiklik tüm sonraki blokları geçersiz kılacağından hemen tespit edilir.

Şeffaflık ve denetlenebilirlik: Blok zincirindeki işlemler genellikle herkes tarafından görülebilir ve takip edilebilir (Zohar, 2015). Bu, şeffaflık sağlar ve kötü niyetli faaliyetleri önlemeye yardımcı olur. Bununla birlikte, bazı blockchain uygulamaları özel veya izinli olabilir ve sadece belirli katılımcıların erişimine açık olabilir (Kosba vd., 2016).

Blok zinciri teknolojisi, kripto paralar (Bitcoin, Ethereum vb.) gibi finansal uygulamaların yanı sıra, akıllı sözleşmeler, tedarik zinciri yönetimi, kimlik doğrulama ve daha pek çok alanda kullanılabilir (Tapscott ve Tapscott, 2016). Bu teknoloji, güvenli, şeffaf ve merkezi olmayan bir çözüm sunarak, işlemleri ve veri paylaşımını daha güvenilir ve etkili hale getirir (Mougayar, 2016).

3.7 Kripto Para ve Bitcoin

Kripto paralar, teknolojinin finans dünyasında yarattığı büyük değişimin bir simgesidir (Nakamoto, 2008). Bu dijital varlıklar, son yıllarda hızla büyüyen ve hâlâ gelişmekte olan bir pazardır (Tapscott ve Tapscott, 2016). Kripto paralar, merkezi olmayan ve genellikle blockchain teknolojisiyle desteklenen dijital varlıklardır (Swan,

2015). Bu varlıklar, halka açık ve özel anahtarlar kullanarak şifrelenmiş işlemlerle gönderilip alınabilir (Maurer vd., 2013). Bitcoin, 2009'da anonim bir geliştirici olan Satoshi Nakamoto tarafından ortaya çıkan ilk kripto paradır (Nakamoto, 2008).

Bitcoin, merkezi olmayan bir para birimi olarak tasarlanmıştır (Nakamoto, 2008). İşlemler, blok zinciri adı verilen dağıtık bir veritabanında saklanır (Swan, 2015). Bu veritabanı, ağdaki düğümler (bilgisayarlar) arasında paylaşılır ve güncellenir (Maurer vd., 2013). Blok zinciri, işlemlerin geri alınamayacağı ve tüm katılımcıların işlem geçmişine erişebileceği güvenli bir kayıt sağlar (Zohar, 2015).

Bitcoin ve diğer kripto paraların üretimi, madencilik adı verilen bir süreçle gerçekleşir (Nakamoto, 2008). Madenciler, yeni bloklar oluşturarak ve işlem doğrulama ile ödüllendirilir (Eyal ve Sirer, 2014). Madencilik, karmaşık matematiksel problemleri çözmeyi gerektirir ve enerji yoğun bir süreçtir (Gervais vd., 2014). Konsensus mekanizması sayesinde, ağdaki katılımcılar, geçerli blok zincirinin hangi versiyonunun doğru olduğuna karar verirler (Bonneau vd., 2015).

Kripto paraların temel özellikleri şunlardır:

- **Merkezi olmayan yapı:** Kripto paraların değeri ve işlemleri, merkezi bir otorite tarafından kontrol edilmez.
- **Şeffaflık**: İşlem geçmişi herkese açıktır ve herhangi bir kişi veya kuruluş tarafından incelenebilir.
- **Sınırlı arz**: Bitcoin gibi bazı kripto paraların toplam arzı sınırlıdır. Bu durum, enflasyon riskini azaltır ve değerin korunmasına yardımcı olur (Nakamoto, 2008).
- **Düşük işlem ücretleri:** Kripto paraların işlem ücretleri, geleneksel bankacılık sistemine göre daha düşüktür (Tapscott ve Tapscott, 2016).
- **Hızlı ve küresel işlemler:** Kripto paralar, sınırlar ve zaman dilimleri arasında hızlı ve kolay işlemler sağlar (Swan, 2015).

Kripto paralar, çeşitli alanlarda kullanılabilir. Bunlar:

- **Ödeme aracı:** Giderek daha fazla işletme, Bitcoin ve diğer kripto paraları ödeme olarak kabul etmektedir (Hughes ve Middlebrook, 2015).

- **Uluslararası para transferi:** Kripto paralar, hızlı ve düşük maliyetli uluslararası para transferlerine olanak tanır (Maurer vd., 2013).
- **Yatırım ve spekülasyon:** Kripto paraların değeri zaman içinde artabileceği için, yatırımcılar ve spekülatörler bu dijital varlıklara yatırım yapmaktadır (Yermack, 2015).
- **Akıllı sözleşmeler:** Ethereum gibi platformlar, kripto paraların temel aldığı teknolojiyi kullanarak, otomatik ve güvenilir işlemler için akıllı sözleşmeler oluşturabilir (Buterin, 2014).

Kripto paraların popülaritesi ve yaygın kullanımı arttıkça, bazı riskler ve eleştiriler de ortaya çıkmıştır. Kripto paraların bazı dezavantajları şunlardır:

- Volatilite: Kripto paraların değeri, kısa sürede büyük dalgalanmalar yaşayabilir. Bu, yatırımcılar ve kullanıcılar için belirsizlik yaratır (Yermack, 2015).
- Güvenlik: Kripto paraların merkezi olmayan yapısı, güvenlik açıklarına ve siber saldırılara karşı daha savunmasız olmalarına neden olabilir (Moore ve Christin, 2013).
- Enerji tüketimi: Madencilik, enerji yoğun bir süreçtir ve ortam üzerinde olumsuz etkileri olabilir (Krause ve Tolaymat, 2018).
- Yasadışı kullanım: Kripto paraların anonim ve merkezi olmayan yapısı, yasadışı işlemler ve kara para aklama için kullanılabilir (Foley vd., 2019).

Kripto paralar ve Bitcoin, dijital varlık devriminin önemli bir parçasıdır (Tapscott ve Tapscott, 2016). Merkezi olmayan yapıları, şeffaflıkları ve hızlı işlem süreleri sayesinde, finansal sistemlerin geleceğini şekillendirecek gibi görünüyorlar (Swan, 2015). Ancak, kripto paraların yaygın kabul görmesi ve güvenilirliğini artırması için, güvenlik, enerji tüketimi ve yasal düzenlemeler gibi alanlarda ilerleme kaydetmesi gerekmektedir (Casey ve Vigna, 2018).

4. UYGULAMA

Çalışmanın bu bölümünde ortaya konulan algoritmaların kripto para fiyat tahmini için uygulanabilirliği ve uygulama sonuçlarının değerlendirilmesi hakkında bilgiler verilecektir.

Çalışma kapsamında otomatik olarak kripto para alım satımı yapan bir uygulama geliştirilip, yapılan alım satım işlemlerinin simülasyonu sağlanmıştır. Uygulama herhangi bir kripto para uygulanabilecek şekilde tasarlanmıştır. Fakat uygulama geliştirilirken ve test edilirken Bitcoin verileri ile çalışılmıştır.

Uygulamanın amacı geçmiş Bitcoin verileriyle bir model geliştirip bu modeli yine gerçek Bitcoin verilerine uygulayıp alım satım veya tutma işlemlerini gerçekleştirerek otomatik bir ticaret yapmak ve bu ticaret sonucunda en yüksek karı elde etmektir.

Uygulama geliştirilirken öncelikle Pekiştirmeli Öğrenme algoritmasıyla denenmiştir. Yapay sinir ağları eklenerek uygulama derin pekiştirmeli öğrenme yöntemleriyle geliştirilmeye devam edilmiştir. Daha sonra sonuçlar üzerinde değerlendirmeler yapılarak adım adım geliştirmeler yapılmış ve her adım sonucunda daha iyi sonuçlar elde edilmesi hedeflenmiştir. Uygulama geliştirilirken karşılaşılan problemler, hangi düzeltmelerin yapıldığı ve her adımdaki sonuçlar aşağıda detaylı bir şekilde anlatılmaktadır.

Uygulamaya başlamadan önce kripto para alım-satımı yapan uzman bir kişinin ticaret sırasında nelere dikkat ettiğini ve neye göre karar verdiğini düşünmek gerekir. Kripto para alım-satımı yapan bir kişi muhtemelen bazı teknik göstergelere bakarak alım satım yapacağı kripto paranın geçmiş fiyat hareketlerine bakarak hangi yönde hareket edeceğine bilinçli bir şekilde karar verir. Bu uygulama da gerçek uzman bir kişiyi taklit ederek geçmiş verilerden beslenip en uygun hamleleri yaparak kar elde etmeye çalışmaktadır.

Uygulama geliştirilirken python programlama dili kullanılmış olup Python 3.9.13 versiyonu kullanılmıştır. Kullanılan bilgisayar Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz 2.90 GHz işlemcisi, 16 GB Ram ve Windows 10 pro ile çalışmaktadır. Model eğitilirken geçen süre hesaplamaları bu bilgisayar için hesaplanmıştır.

Veriler https://bitcoincharts.com/ web adresinden istenilen tarih aralıkları belirtilerek istenilen kripto para verileri indirilmiştir. Bu uygulama için başlangıç olarak 2020-04-23 00:00 - 2020-10-19 07:00 aralığında saatlik değerleri içeren yaklaşık 6 aylık Bitcoin verileri kullanılmıştır. Veriler csv formatında kullanılmıştır. Veri dosyasında Bitcoin için saatlik olacak şekilde Tarih, Açılış Değeri, En yüksek değer, En düşük değer, Kapanış değeri ve Hacim değerleri yer almaktadır. Aşağıda verilerin nasıl göründüğü gösterilmektedir.

Date	Open	High	Low	Close	Volume
2020-04-23 00:00	7138.29	7158.7	7108	7122.21	81.59
2020-04-23 01:00	7126.3	7192.82	7118.33	7186.36	175.45
2020-04-23 02:00	7186.72	7193.13	7129.19	7130	131.63
2020-04-23 03:00	7130.92	7163.41	7117.97	7118.83	227.12
2020-04-23 04:00	7115.9	7137.92	7102.62	7132.38	60.48
2020-04-23 05:00	7135.17	7142.48	7098.86	7104.3	90.99
2020-04-23 06:00	7103.31	7103.31	7031.07	7046.02	452.96
2020-04-23 07:00	7038.45	7081.85	7038.45	7065.96	87.46
2020-04-23 08:00	7065.05	7108.19	7055	7091.22	161.56
2020-04-23 09:00	7087.35	7111.38	7056.42	7111.38	203.53
2020-04-23 10:00	7106.26	7113.61	7068.72	7074.57	124.44
2020-04-23 11:00	7075.06	7155.63	7068.48	7136.87	251.9
2020-04-23 12:00	7137.8	7199	7118.61	7186.21	1500.24
2020-04-23 13:00	7186.84	7274.37	7159.81	7251.1	1605.26
2020-04-23 14:00	7257.68	7756.69	7257.68	7546	3734.63
2020-04-23 15:00	7549.56	7566.55	7473	7538.64	1554.19

Şekil 4.1 Uygulamada kullanılan Bitcoin verileri

Uygulama öncelikle pekiştirmeli öğrenme algoritmasıyla test edilmiştir. Bunun için öncelikle ortamı (environment) tanımlamamız gerekmektedir. Ortam bir ajanın çalışması ve öğrenmesi için gerekli tüm işlevleri içermektedir. Aşağıda pekiştirmeli öğrenme'de her ortamın sahip olması gereken arabirim gösterilmektedir.

```
class CustomEnv:
    def __init__(self, arg1, arg2, ...):
        # Define action space and state size

        # Example when using discrete actions of 0,1,2:
        self.action_space = np.array([0, 1, 2])

        # Example for using image as input for custom environment:
        self.state_size = np.empty((HEIGHT, WIDTH, CHANNELS)), dtype=np.uint8)

    # Execute one time step within the environment
    def step(self, action):
        ...

    # Reset the state of the environment to an initial state
    def reset(self):
        ...
    # render environment visualization
    def render(self):
        ...
```

Şekil 4.2 Ortam için gerekli örnek arabirim

Şekil 4.2'deki değişkenlerden de anlaşılacağı üzere ortam tanımlanırken olması gereken tüm bilgilerin tanımlanması gerekmektedir. Bölüm 3'de anlatılan pekiştirmeli öğrenme unsurlarının burada tanımlanması gerekmektedir. Bu uygulamada ajanın ortam içerisinde gerçekleştireceği eylemler (action) al, sat ve tut şeklindedir. Yani Ajan sadece üç eylemde bulunabilir. Durumlar (State) ajan tarafından gözlemlenecek bütün durumları ifade etmektedir.

Uygulamada ajanın başlangıçta 1000 dolar parası bulunmaktadır. Ajan pekiştirmeli öğrenme algoritmasına göre rastgele alım satım yaparak eğitilmeden gözlemleniyor. Buradaki ilk amacımız ajanı eğitmek değil bütün parametrelerin ve algoritmamızın doğru çalışıp çalışmadığını kontrol etmek. Sonraki kısımlarda eğitilerek hareket eden ajan ile rastgele hareket eden ajanı karşılaştırma imkânı da doğmuş olacaktır. Ajana elindeki 1000 dolar ile rastgele Bitcoin alım satımı yaptırılmış ve aynı işlem 10 bölüm boyunca tekrarlanmıştır. Şekil 4.3'de gösterildiği gibia ajan 10 bölüm sonucunda 981 dolarlık ortalama bir gelir elde etmiştir.

```
Step: 1771, Net Worth: 973.1386787365324
Step: 1772, Net Worth: 973.1386787365324
Step: 1773, Net Worth: 973.1386787365324
Step: 1774, Net Worth: 973.1386787365324
Step: 1775, Net Worth: 973.1386787365324
Step: 1776, Net Worth: 973.1386787365324
Step: 1777, Net Worth: 973.1386787365324
net_worth: 973.1386787365324
average_net_worth: 981.7271660925813
>>>
```

Şekil 4.3 Pekiştirmeli öğrenme ile rastgele alışveriş yapan ajanın elde ettiği gelir

Bu sonuca göre ajanımız 19 dolarlık bir zarara uğramıştır. Fakat burada eylemler rastgele seçildiği için işlem her tekrarlandığında daha farklı bir sonuç elde edebilir. Yani ajan kar da edebilir zarar da. Bu aşamada bunun bizim için bir önemi bulunmamaktadır.

Algoritmamızın çalıştığını test ettikten sonra artık ajanımızın gerçekten bir yapay zekâ ile hareket ettirmemiz için uygulamamızı biraz değiştirip geliştirmemiz gerekmektedir. Derin pekiştirmeli öğrenme'de yapay sinir ağlarının kullanıldığı, daha önceki bölümlerde anlatılmıştır. Bu uygulamada Proksimal Politika Optimizasyon (PPO) algoritması kullanılacaktır. Bölüm 3'de anlatıldığı gibi PPO algoritmasında iki model kullanılır. Bunlar aktör model (actor model) ve eleştirmen model (critic model) olarak adlandırılır. Aktör model ortamda belirli bir durum içerisinde hangi eylemi gerçekleştirmesi gerektiğini öğrenen modeldir. Eleştirmen model ise aktör model tarafından tahmin edilen edilen eylemin ortamı daha iyi bir duruma getirip getirmediğini değerlendirmeyi öğrenir.

Aktör model aşağıdaki gibi tanımlanmıştır.

```
class Actor_Model:
    def __init__(self, input_shape, action_space, lr, optimizer):
        X_input = Input(input_shape)
        self.action_space = action_space

        X = Flatten(input_shape=input_shape)(X_input)
        X = Dense(512, activation="relu")(X)
        X = Dense(256, activation="relu")(X)
        X = Dense(64, activation="relu")(X)
        output = Dense(self.action_space, activation="softmax")(X)

        self.Actor = Model(inputs = X_input, outputs = output)
        self.Actor.compile(loss=self.ppo_loss, optimizer=optimizer(lr=lr))
```

Şekil 4.4 Aktör model

Eleştirmen model aşağıdaki gibi tanımlanmıştır.

```
class Critic_Model:
    def __init__(self, input_shape, action_space, lr, optimizer):
        X_input = Input(input_shape)

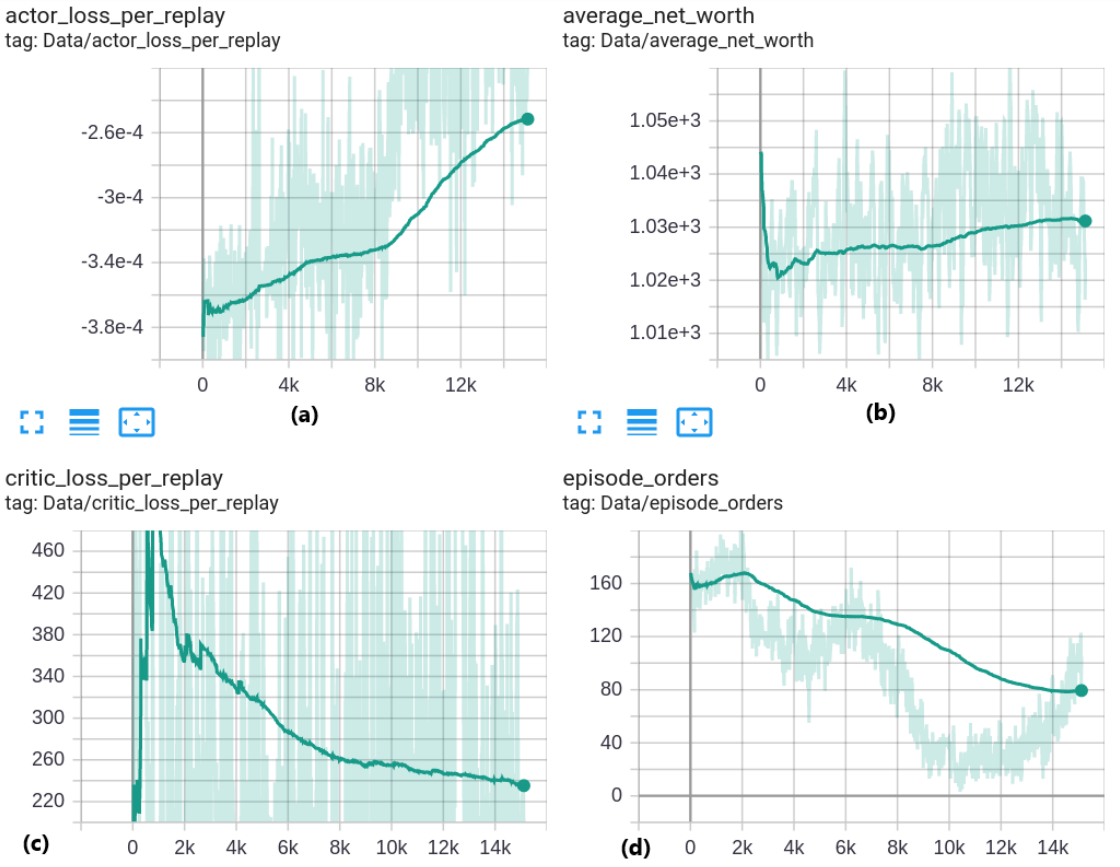
        V = Flatten(input_shape=input_shape)(X_input)
        V = Dense(512, activation="relu")(V)
        V = Dense(256, activation="relu")(V)
        V = Dense(64, activation="relu")(V)
        value = Dense(1, activation=None)(V)

        self.Critic = Model(inputs=X_input, outputs = value)
        self.Critic.compile(loss=self.critic_PPO2_loss, optimizer=optimizer(lr=lr))
```

Şekil 4.5 Eleştirmen model

Derin pekiştirmeli öğrenme algoritması ve PPO algoritmasının birleştirilmesiyle elde edilen algoritma için, başlangıç değer olarak 1000 dolarlık bir varlık, öğrenme oranı 0.00001, indirim faktörü (discount factor) 0.99, optimizasyon algoritması olarak Adam (Adaptive Moment Estimation), window size değeri 50, eğitim periyot sayısı(epochs) yani verilerin modele girme sayısı 1 olarak seçilmiştir. Ödül, eylem sonucunda elde edilen net gelir ile bir önceki durumda elde edilen net gelirin farkı olarak hesaplanmıştır. Uygulama model eğitimi için 20000 bölüm olarak seçilip çalıştırılmıştır. Bölümler sırasında elde edilen net gelir değerlerine göre modeller kayıt edilmiştir. Eğer bölüm sonunda bir öncekinden daha yüksek gelir elde edildiyse eski model silinip yerine yeni model kaydedilmiştir. Eğitim sürecinin performansını

görebilmek için python içerisinde bulunan Tensorbord kütüphanesi kullanılmıştır. Tensorboard üzerinde aşağıdaki grafikler elde edilmiştir.

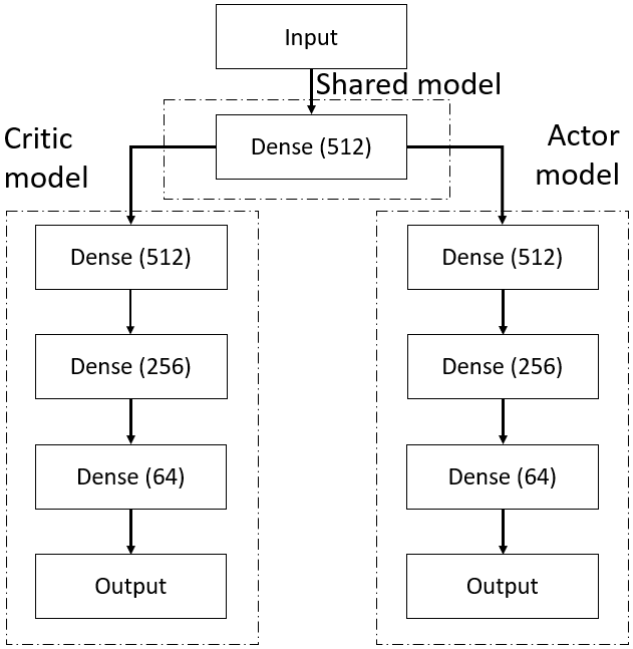


Şekil 4.6 Aktör-eleştirmen model mimarisi ile eğitim süreci grafikleri

Grafikler incelendiğinde Şekil4.6(b) grafiği, ortalama net gelirin az miktarda da olsa arttığını görmekteyiz. Bu da kesinlikle bir kar elde ettiğini göstermektedir. Bu durum ajanımızın bir şeyler öğrenebildiğini göstermektedir. Şekil4.6(d) grafiği, her bir bölümde alım satım işleminin azaldığını göstermektedir. Yani ajan alım satım yapmaktansa mevcut elinde tutma stratejisinin daha iyi olacağını öğrenmiş. Bu aslında istenilen bir durum değildir. Bu durum bir sonraki güncellemelerle düzeltilmeye çalışılacaktır.

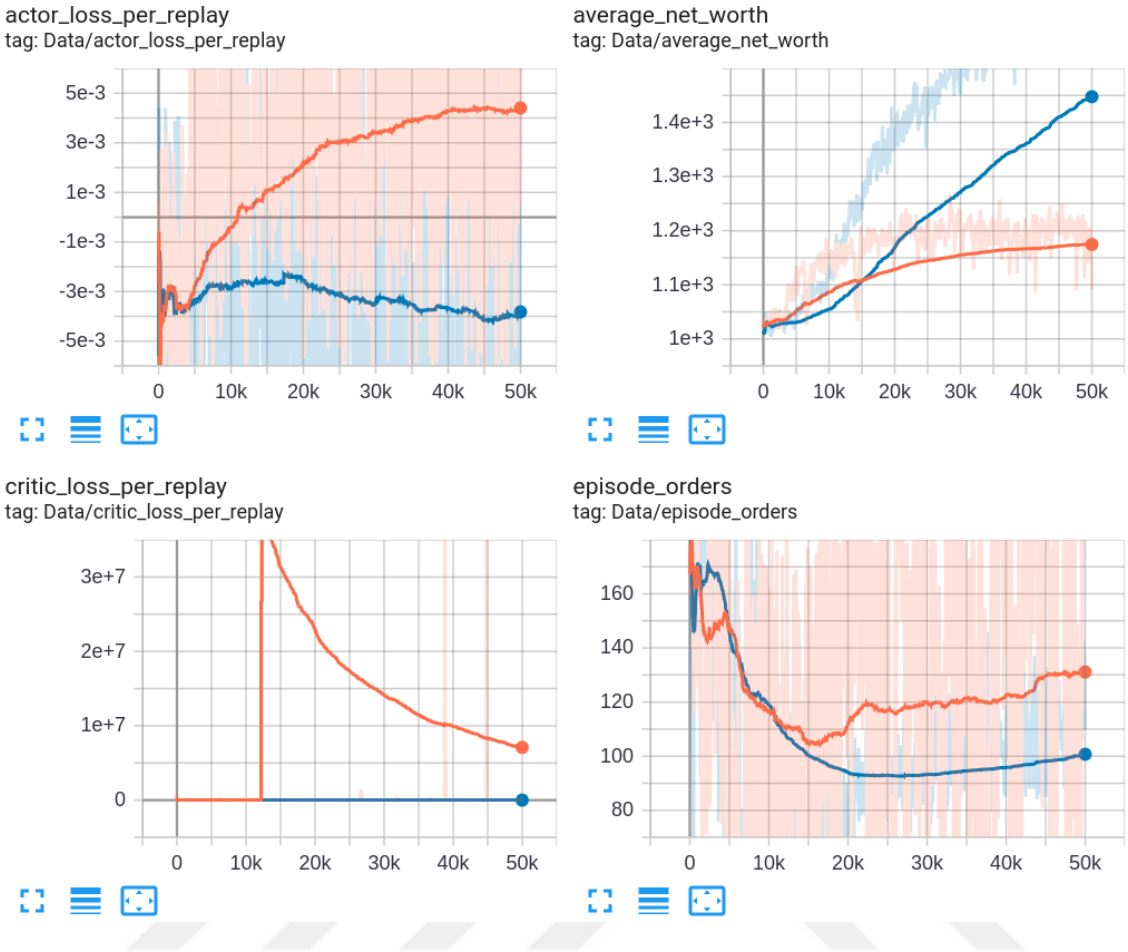
Eğitilen modellerle 30 günlük test verisi 1000 bölümlük bir süreçte test edildiğinde ortalama net gelir olarak 1043 dolar elde etmiştir. Bu da %4.3'lük bir kar demektir. Bu sonuç çok iyi bir sonuç olmasa da ajanımızın öğrenme konusunda başarısının arttığını göstermektedir.

Uygulamanın başarısını artırmak için model mimarisi üzerinde değişikliklerin yapılması sonuçları değiştirebilir. Aktör ve eleştirmen modeller arasına "paylaşılan model (Shared Model)" adında yeni bir model ekleyerek modellerimizi tekrar test etmek başarıyı artırabilir.



Şekil 4.7 Paylaşılan katmana sahip aktör-eleştirmen model mimarisi

Paylaşılan katman iki farklı model ile test edilip karşılaştırma yapılmıştır. Bu modellerden ilki zaten kullanılan "Dense" Ağı ikincisi ise CNN olarak kısaltılan Evrişimli Sinir Ağları (Convolution Neural Networks) modelidir. Ayrıca modellerin kaydedilmesinde de bir değişiklik yapılmıştır. İki farklı Paylaşılan katman için 50000 bölümden oluşan bir eğitim süreci gerçekleştirilmiştir. Eğitim için başlangıç değer olarak 1000 dolarlık bir varlık, öğrenme oranı 0.00005, indirim faktörü (discount factor) 0.95, optimizasyon algoritması olarak Adam (Adaptive Moment Estimation), window size değeri 50, eğitim periyot sayısı(epochs) yani verilerin modele girme sayısı 5 olarak seçilmiştir. Paylaşılan katman iki farklı model ile eğitim süreci gerçekleştirildiğinde Tensorboard grafikleri Şekil 4.8'deki gibi olmuştur.



Şekil 4.8 Dense ve CNN modeli Tensorboard grafiği

Grafikte turuncu renk Dense modelini mavi renk ise CNN modelini temsil etmektedir. Ortalama net gelir grafiği (average_net_worth) incelendiğinde CNN modelinin zamanla daha yüksek ödüller almayı öğrendiği görülmektedir.

Dense ile eğitilen modellerle 30 günlük test verisi 1000 bölümlük bir süreçte test edildiğinde ortalama net gelir olarak yaklaşık 1054 dolar elde etmiştir. Bu da %5.4'lük bir kar elde edildiğini gösterir.

CNN ile eğitilen modellerle 30 günlük test verisi 1000 bölümlük bir süreçte test edildiğinde ortalama net gelir olarak yaklaşık 1034 dolar elde etmiştir. Bu da %3.4'lük bir kar elde edildiğini gösterir.

Sonuçlar değerlendirildiğinde eğitim sırasında CNN daha başarılı görünmesine rağmen daha kötü sonuçlar çıktığı görülmüştür. Burda bir tutarsızlık olduğu anlaşılmakta ve modelin geliştirilmesi için başka güncellemeler yapılması gerektiği görülmektedir.

Yukarıdaki sonuçların çok tatmin edici sonuçlar olmadığı görülmektedir. Bunun sebeplerinden biri eğitim verisinin az olması, diğer bir sebebi de eğitim bölüm sayısının yetersiz kalması olabileceği düşünülmüştür. Fakat şimdiye kadar bunların artırılmasının sebebi eğitim sürecinin çok uzun sürmesiydi. Bir 20000 bölümlük bir eğitim yaklaşık 12 saat sürebilmektedir. Örneğin bölüm sayısını 200000 yapmak istersek bu günler sürecek bir eğitim demektir. Teknik koşullar sebebiyle eğitimin durması günlerce süren bir çabanın heba olması anlamına gelmekteydi. Fakat araştırmalar sonucunda buna bir çözüm bulunmuştur: Çok işlemcili eğitim (Multiprocessing training).

Çok işlemcili eğitim işlemi, yapılacak işi worker olarak adlandırılan birden fazla işçiye bölerek CPU üzerinde aynı anda birden fazla işlem yapma olanağı tanımaktadır. Bunun için yeni bir ortam tanımlaması yapılması gerekmektedir. Yeni ortam üzerinde yapılan değişikliklerden sonra artık eğitim için daha fazla veri ve daha fazla bölüm çalıştırılabilir.

Daha önce 6 aylık Bitcoin saatlik değişen değerleri içeren veri yerine 2018-2020 yılları arasındaki tüm Bitcoin saatlik verileri indirilmiş ve eğitim için kullanılmıştır. Bu da yaklaşık 3 yıllık veri demektir. Aynı zamanda eğitim bölüm sayısı 20000'den 200000'e çıkarılmıştır. Çizelge 4.1'de eğitim sonuçları detaylı bir şekilde verilmiştir.

Çizelge 4.1 Çok işlemcili eğitim sonuçları

Model adı	En iyi skor	Model	Window size	Eğitim bölüm sayısı	Öğrenme oranı	Batch size	Eğitim süresi	Başlangıç varlık
Model 1	1984.93	Dense	50	200bin	0.00005	32	20 saat	1000($)
Model 2	2961.57	CNN	50	200bin	0.00005	32	22 saat	1000($)
Model 3	1914.81	Dense	100	200bin	0.00005	32	24 saat	1000($)
Model 4	3142.13	CNN	100	200bin	0.00005	32	30 saat	1000($)

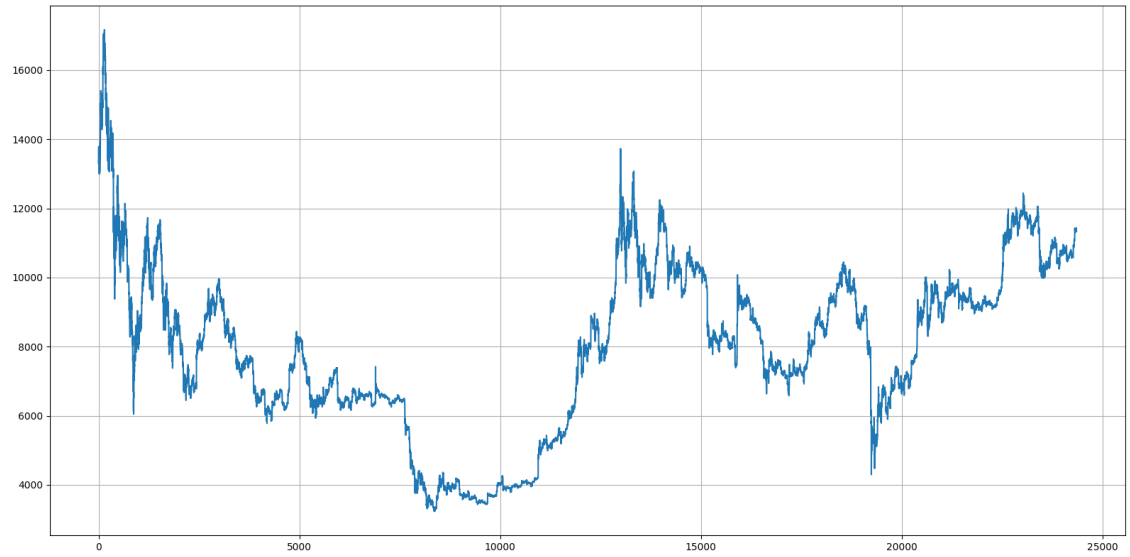
Yukarıdaki sonuçlardan sonra kullanılacak en iyi modelin CNN kullanılan Model 4 olduğu görülmektedir. 3 aylık test verisi 1000 bölümlük test süreciyle tüm modeller ile test edildiğinde Çizelge 4.2'deki sonuçlar elde edilmiştir.

Çizelge 4.2 Çok işlemcili test sonuçları

Model adı	En iyi skor	Model	Window size	Test bölüm sayısı	Başlangıç varlık	Zarar eden bölüm sayısı	Ortalama net gelir
Model 1	1984.93	Dense	50	1000	1000($)	1	1262
Model 2	2961.57	CNN	50	1000	1000($)	31	1155
Model 3	1914.81	Dense	100	1000	1000($)	0	1312
Model 4	3142.13	CNN	100	1000	1000($)	0	1332

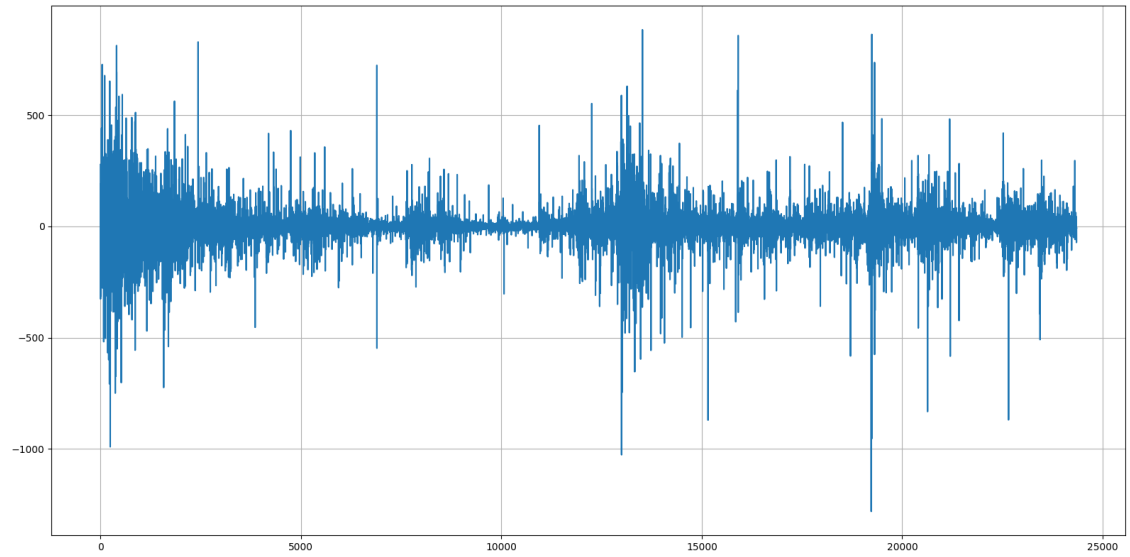
Yukarıdaki çizelgeden anlayacağımız gibi CNN modeli en yüksek getiri ile en iyi model olmuştur. O halde eğitim verilerini artırmak ve bölüm sayısını arttırmak, modelin daha iyi eğitilmesini ve %33 gibi çok iyi sayılabilecek bir kar oranı elde edilmesini sağlamıştır.

CNN modeli her ne kadar başarı oranımızı artırmış olsa da daha sonra farklı veriler üzerinde test edildiğinde yine başarılı fakat daha dengesiz sonuçlar verdiği görülmüştür. Bu da zaman serisi verilerinin durağan olmayan veriler olmasından kaynaklanabileceğini akla getirmektedir. Bu sebeple zaman serisi verilerinde eğitim sırasında yükseliş trendi verilerinden öğrenilen bir makine öğrenmesi modelinin, düşüş trendini tahmin etmesi zor olacaktır. Bu sorunu çözmek için verilerin normalizasyon işlemlerinden geçmesi gerekmektedir. Verileri daha normal bir dağılım biçimine dönüştürmek için fark alma ve dönüştürme teknikleri kullanılmıştır. Bitcoin verilerimizin kapanış değerleri bir grafiğe döküldüğünde Şekil 4.9'daki gibi bir sonuç elde edilmektedir.



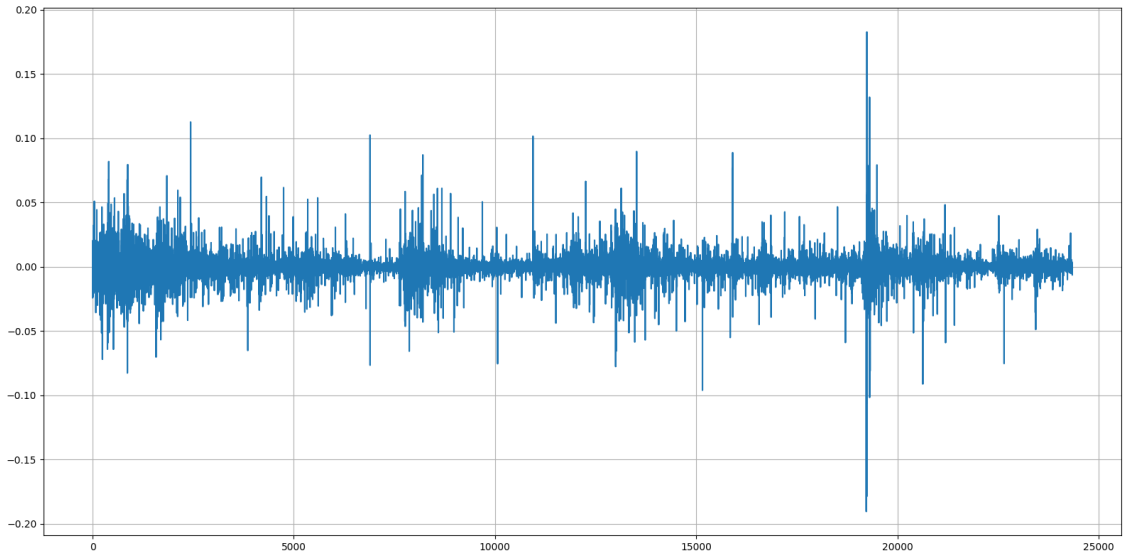
Şekil 4.9 Bitcoin kapanış değerleri

İlk olarak fark alma işlemini gerçekleştirmeliyiz. Fark alma işlemi her t anındaki değeri t-1 anındaki değerden çıkarma işlemidir. Bu işlemden sonra grafik Şekil 4.10'daki gibi olacaktır.



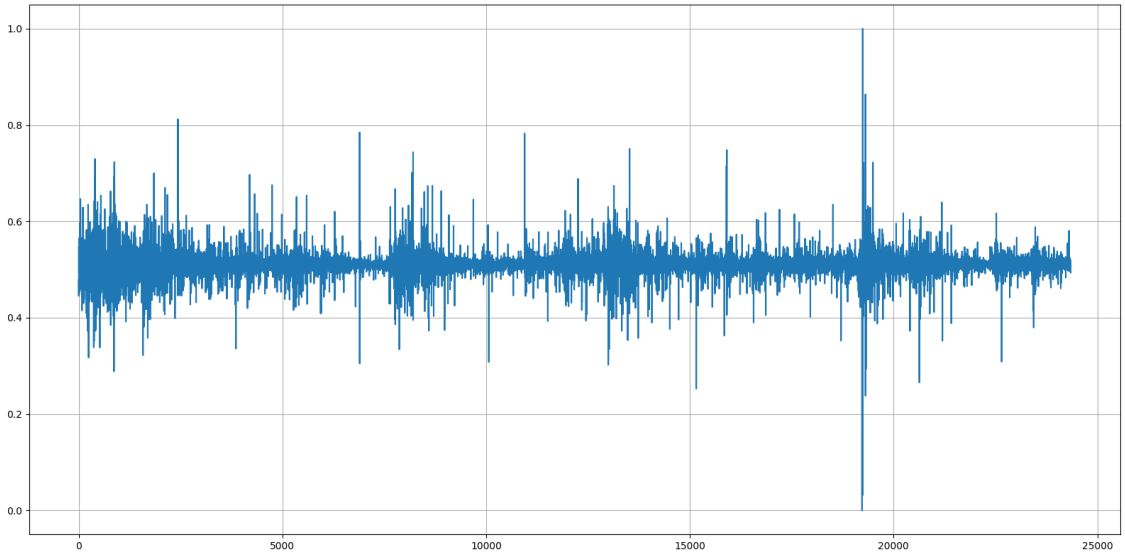
Şekil 4.10 Fark alma yönteminden sonra Bitcoin kapanış değerleri

Veri bir önceki andaki veriden çıkarılmadan önce logaritması alınarak yapılmalıdır böylelikle Şekil 4.11'deki gibi bir grafik elde edilecektir.



Şekil 4.11 Logaritmik fark alma işleminden sonra Bitcoin kapanış değerleri

Logaritmik fark alma işleminden sonra bu verilerin 0 ile 1 arasına indirgenmesi gerekmektedir. Bu da o anki kapanış değerinden en düşük kapanış değeri çıkarılıp en yüksek kapanış değeri ile en düşük kapanış değeri farkına bölerek yapılmaktadır. Artık Bitcoin kapanış değerlerimiz Şekil 4.12'deki gibi görülmektedir.



Şekil 4.12 0 ile 1 arasına alınarak normalizasyon uygulanmış Bitcoin kapanış değerleri

Bu işlemlerden sonra verimize normalizasyon uygulamış oluyoruz. Artık verilerimiz durağan olduğundan modelimiz daha iyi ve kararlı sonuçlar vermektedir.

5. TARTIŞMA VE SONUÇ

Çalışmanın amacının derin pekiştirmeli öğrenme ile kripto paraların fiyatlarının tahmin edilmesi olduğu daha önce anlatılmıştır. Bu çalışma kapsamında kripto para birimi olarak Bitcoin kullanılmıştır. Hem modellerin eğitilmesi için hem de test için Bitcoin verileri kullanılmıştır. Fakat test için kullanılacak veriler eğitim sürecine alınmamıştır. Test tamamen yeni verilerle yapılmıştır.

Tez kapsamında Bitcoin kapanış verileri ile kendi kendine öğrenen ve alım satım işlemleri yapan bir uygulama tasarlanmıştır. Uygulama derin pekiştirmeli öğrenme ve proksimal politika optimizasyonu algoritmaları kullanılarak aktör-eleştirmen mimarisi temelinde Dense Katmanları ve Evrişimsel Sinir Ağları (CNN) ile modeller eğitilmiştir. CNN ile eğitilen model en başarılı model olarak bulunmuştur. Sürekli güncellenip test edilerek ve sonuçlar karşılaştırılarak geliştirilen uygulama için birçok yöntem ve algoritma denenmiştir. Yapılan tüm işlemler bölüm 4'de detaylarıyla anlatılmış ve sonuçlar paylaşılmıştır.

Elde edilen en iyi model test edildiğinde başlangıçta 1000 dolar parası olan ajan, 1000 defa tekrar eden denemeler sonucunda, 3 aylık Bitcoin ticareti ile ortalama 1332 dolarlık net gelir elde etmiştir. Bu da %33'lük bir kar demektir. Bu sonuç kendi kendine öğrenen ve ticaret yapan bir yapay zekâ ajanı için oldukça başarılı bir sonuçtur.

Eğitilen modelin başarısı BTC, ETH, ADA, LTC ve XRP kripto paraları için 01.01.2023 – 24.04.2023 tarihleri arasındaki saatlik kapanış verileri ile test edildiğinde aşağıdaki sonuçlar elde edilmiştir.

Çizelge 5.1 BTC, ETH, ADA, LTC ve XRP test sonuçları

Model adı	Kripto para	Model	Window size	Test bölüm sayısı	Başlangıç varlık	Zarar eden bölüm sayısı	Ortalama net gelir
Model 4	BTC	CNN	100	100	1000($)	0	1292.69($)
Model 4	ETH	CNN	100	100	1000($)	0	1521.44($)
Model 4	ADA	CNN	100	100	1000($)	20	1140.37($)
Model 4	LTC	CNN	100	100	1000($)	70	953.56($)
Model 4	XRP	CNN	100	100	1000($)	5	1239.62($)

Yukarıdaki çizelgeden anlaşılacağı üzere modelimiz BTC üzerinde %29'luk kar etmiş ve hiçbir bölümde zarar etmemiştir. En iyi sonuç ETH üzerinde elde edilmiştir. %52'lik bir kar oranı elde edilmiş ve hiçbir bölümde zarar edilmemiştir. ADA kripto parası için %14 oranında kar edilmiş fakat 100 bölümün 80'inde kar elde ettiği görülmektedir. LTC kripto parası test edilen kripto paralardan en kötü performans gösteren kripto para olmuştur. %4.7'lik bir zarar ve 100 bölümün 30'unda kar elde edildiği görülmektedir. XRP için %23'lük bir kar elde edilmiş ve 100 bölümün 95'inde kar elde edilmiştir.

Sonuçlar incelendiğinde eğitilen modelin büyük oranda başarılı olduğu görülmektedir. Fakat yine de gerçek ortamda uygulanması risk taşımaktadır.

Model eğitilirken sadece bitcoin'in 3 yıllık saatlik verisi ile eğitilmiştir. Modelin geliştirilmesi için birden fazla kripto para verilerinin yer aldığı ve daha büyük veri setleri kullanılabilir. Bu durumda algoritma ve mimari de değiştirilip model tekrar eğitilebilir. Farklı kripto para verileriyle birbiri arasında alım-satım işlemi gerçekleştirilerek eğitilen modelin daha başarılı sonuçlar vermesi beklenebilir.

KAYNAKLAR

Alpaydin, E. (2014). ***Introduction to Machine Learning***. MIT Press: Cambridge, UK.

Bao, W., Yue, J., ve Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7), e0180944.

Bishop, C. M. (2006). ***Pattern Recognition and Machine Learning***. Springer: New York, USA.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., Zieba, K. (2016). End to end learning for self-driving cars. *arXiv preprint* arXiv:1604.07316.

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., Felten, E. W. (2015). SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. **IEEE Symposium on Security and Privacy**, California, USA.

Bouri, E., Molnár, P., Azzi, G., Roubaud, D., Hagfors, L. I. (2017). On the hedge and safe haven properties of Bitcoin: Is it really more than a diversifier? *Finance Research Letters*, 20, 192-198.

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *white paper*, *3*(37), 2-1.

Cao Z, Simon Z, Wei SE, Sheikh SE (2017) Realtime multi-person 2D pose estimation using part affinity fields. **IEEE conference on computer vision and pattern recognition, CVPR,** College Park, USA.

Casey, M. J., Vigna, P. (2018). ***The Truth Machine: The Blockchain and the Future of Everything.*** St. Martin's Press: NewYork, USA.

Corbet, S., Lucey, B., Yarovaya, L., ve Cummins, M. (2018). The inefficiency of Bitcoin revisited: a dynamic approach. *Economics Letters*, 171, 1-6.

Crawford, K., ve Calo, R. (2016). There is a blind spot in AI research. *Nature*, 538(7625), 311-313.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., ... ve Ng, A. Y. (2012). Large scale distributed deep networks. *Advances in Neural Information Processing Systems*, 1223-1231.

Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. **Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (long and short papers):** Minneapolis, USA.

Dhar, V. (2013). Data science and prediction. *Communications of the ACM*, 56(12), 64-73.

Diffie, W., Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 644-654.

Ding, X., Zhang, Y., Liu, T., Duan, J. (2015). Deep learning for event-driven stock prediction. **Twenty-fourth international joint conference on artificial intelligence**: California, USA.

Eyal, I., Sirer, E. G. (2014). Majority is not Enough: Bitcoin Mining is Vulnerable. **International Conference on Financial Cryptography and Data Security**: Christ Church, Barbados

Fischer, T., Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*,

270(2), 654-669.
Foley, S., Karlsen, J. R., Putniņš, T. J. (2019). Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? *The Review of Financial Studies*, 32(5), 1798-1853.
Gervais, A., Karame, G. O., Capkun, S., Capkun, V. (2014). Is Bitcoin a Decentralized Currency? *IEEE Security ve Privacy*, 12(3), 54-60.
Gkillas, K., Katsiampa, P., Longin, F. (2019). The cryptocurrency market: an exploratory study on the impact of Bitcoin on other digital currencies. *The European Journal of Finance*, 25(17), 1672-1690.
Glaser, F., Zimmermann, K., Haferkorn, M., Weber, M. C., Siering, M. (2014). Bitcoin - Asset or Currency? Revealing Users' Hidden Intentions. *ECIS 2014 Proceedings.*
Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: *Proceedings of the international conference on artificial intelligence and statistics (AISTATS)*, pp 315–323
Goodfellow I, Bengio Y, Courville A (2016) ***Deep learning***. MIT Press: Cambridge, UK.
Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. *Proceedings of the neural information processing systems conference: Advances in neural information processing systems 27.*
Halevy, A., Norvig, P., Pereira, F. (2009). The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 24(2), 8-12.
He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. *Proceedings of the IEEE international conference on computer vision*, 1026–1034.
He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.
Hileman, G., Rauchs, M. (2017). ***Global cryptocurrency benchmarking study***. Cambridge Centre for Alternative Finance: Cambridge, UK.
Hinton, G. E., Osindero, S., Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527-1554.
Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Netw*, 2(5):359–366
Hughes, S. J., Middlebrook, S. T. (2015). Advancing a Framework for Regulating Cryptocurrency Payments Intermediaries. *Yale Journal on Regulation*, 32, 495-559.
Jiang, Z., Liang, J. (2017). Cryptocurrency portfolio management with deep reinforcement learning. *2017 IEEE International Conference on Big Data*, 2270-2275.
Jordan, M. I., Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.
Kelleher, J. D., Namee, B. M., D'Arcy, A. (2015). ***Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies***. MIT Press: Cambridge, UK

Kim, Y. B., Kim, J. G., Kim, W., Im, J. H., Kim, T. H., Kang, S. J., Kim, C. H. (2016). Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PloS one*, *11*(8), e0161197.

Knight, W. (2017). The Dark Secret at the Heart of AI. *MIT Technology Review*, 120(3), 54-65.

Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. *In 2016 IEEE Symposium on Security and Privacy (SP)*, 839-858.

Krause, M. J., Tolaymat, T. (2018). Quantification of energy and carbon costs for mining cryptocurrencies. *Nature Sustainability*, 1(11), 711-718.

Krizhevsky A, Hinton G. (2009) *Learning multiple layers of features from tiny images. Technical Report*. Citeseer.

Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097-1105.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W (2017) Photo-realistic single image super-resolution using a generative adversarial network. **IEEE conference on computer vision and pattern recognition (CVPR),** College Park, USA.

Liao B, Zhang J, Cai M, Tang S, Gao Y, Wu C, Yang S, Zhu W, Guo Y, Wu F (2018a) DestResNet: a deep spatiotemporal residual network for hotspot traffic speed prediction. **2018 ACM multimedia conference on multimedia conference**, New York, USA.

Liao B, Zhang J, Wu C, McIlwraith D, Chen T, Yang S, Guo Y, Wu F (2018b) Deep sequence learning with auxiliary information for traffic prediction. In: Proceedings of the **24th ACM SIGKDD international conference on knowledge discovery ve data mining,** New York, USA.

Liew, J. K. S., Zeng, Y., Yao, Y. (2019). Predicting Cryptocurrency Movements with Sentiment Analysis and Deep Learning. **2019 IEEE International Conference on Big Data**, LA, CA, USA.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60-88.

Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis. **49th annual meeting of the association for computational linguistics: human language technologies, HLT '11, vol 1. Association for Computational Linguistics**, Stroudsburg, Pennsylvania.

Maurer, B., Nelms, T. C., Swartz, L. (2013). "When perhaps the real problem is money itself!": The practical materiality of Bitcoin. *Social Semiotics*, 23(2), 261-277.

McCarthy, J., Minsky, M. L., Rochester, N., Shannon, C. E. (1955). A proposal for the Dartmouth Summer Research Project on Artificial Intelligence. **The Dartmouth Summer Research Project on Artificial Intelligence**, Hannover, Deutschland.

McCorduck, P. (2004). ***Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence***. CRC Press: Boca Raton, FL.

McCulloch, W. S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous

activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115-133.
McNally, S., Roche, J., Caton, S. (2018). Predicting the price of Bitcoin using machine learning. *26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, 339-343.
Merkle, R. C. (1987). A digital signature based on a conventional encryption function. *Advances in Cryptology—CRYPTO'87: Proceedings 7*, 369-378.
Mitchell, T. M. (1997). ***Machine Learning***. McGraw-Hill: New York, USA.
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533.
Moore, T., Christin, N. (2013). Beware the middleman: Empirical analysis of Bitcoin-exchange risk. **Financial Cryptography and Data Security: 17th International Conference, FC 2013**, Okinawa, Japan.
Mougayar, W. (2016). ***The business blockchain: promise, practice, and application of the next internet technology***. John Wiley ve Sons: New York, USA.
Nakamoto, S., (2008). A peer-to-peer electronic cash system. Erişim tarihi: 20 Ocak 2023 Erişim adresi: https://bitcoin. org/bitcoin.pdf.
Ng AY, Jordan MI (2002) On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. *Proceedings of the neural information processing systems. Advances in neural information processing systems. Conference*, 841–848.
Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. *Proceedings of the international conference on computer vision (ICCV)*, 1520–1528.
Rish I (2001) An empirical study of the naive Bayes classifier. *International joint conference on artificial intelligence 2001 workshop on empirical methods in artificial intelligence. vol 3*, 41–46.
Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386
Ruck DW, Rogers SK, Kabrisky M, Oxley ME, Suter BW (1990) The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Trans Neural Netw* 1(4):296– 298
Russell, S. J., Norvig, P. (2016). ***Artificial Intelligence: A Modern Approach***. Pearson: London, UK
Samek, W., Wiegand, T., Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint* arXiv:1708.08296.
Samuel A (1959) Some studies in machine learning using the game of checkers. *IBM J Res Dev 3*:210–219
Schwartz, P. M. (2013). Data protection law and the ethical use of analytics. *Communications of the ACM*, 56(8), 38-40.
Sezer, O. A., Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70, 525-538.
Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... ve Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.
Sirignano, J., Cont, R. (2018). Universal features of price formation in financial

markets: perspectives from Deep Learning. *Quantitative Finance*, 18(10), 1635-1663.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.

Stanciu, A. (2019). A neural network based approach to forecast the value of the Bitcoin cryptocurrency. *Informatica Economica*, 23(4), 5-16.

Sun, C., Shrivastava, A., Singh, S., Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. *Proceedings of the IEEE international conference on computer vision*, 843-852.

Sutton, R. S., Barto, A. G. (2018). ***Reinforcement Learning: An Introduction***. MIT Press, Cambridge, UK.

Swan, M. (2015). ***Blockchain: blueprint for a new economy***. O'Reilly Media, Inc, California, USA.

Sze, V., Chen, Y. H., Yang, T. J., Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12), 2295-2329.

Taigman, Y., Yang, M., Ranzato, M., Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701-1708.

Tapscott, D., Tapscott, A. (2016). ***Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World***. Penguin: London, UK.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433-460.

Xu, B., Wang, N., Chen, T., Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Yang G, Yu S, Dong H, Slaubaugh, GG, Dragotti PL, Ye X, Liu F, Arridge SR, Keegan J, Guo Y, Firmin DN (2018) DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. *IEEE Trans Med Imaging* 37(6):1310–1321

Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) XLNet: generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 5754–5764.

Yermack, D. (2015). Is Bitcoin a Real Currency? An Economic Appraisal. ***Handbook of Digital Currency***. Elsevier: Amsterdam, Holland.

Zarsky, T. Z. (2013). Transparent predictions. *University of Illinois Law Review*, 1503.

Zhang J, Lertvittayakumjorn P, Guo Y (2019a) Integrating semantic knowledge to tackle zero-shot text classification. *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (long and short papers)*, 1031–1040.

Zhang J, Zhao Y, Saleh M, Liu PJ (2019b) PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. *Preprint. arXiv:191208777*

Zhang, S., Yao, L., Sun, A., Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1), 1-38.

Zohar, A. (2015). Bitcoin: under the hood. *Communications of the ACM*, 58(9), 104-113.

ÖZ GEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Serbest ZİYANAK

Eğitim Bilgileri

Lisans

Üniversite : Mersin Üniversitesi
Fakülte : Mühendislik Fakültesi
Bölüm : Bilgisayar Mühendisliği
Mezuniyet Yılı : 2010

Yüksek Lisans

Üniversite : Van Yüzüncü Yıl Üniversitesi
Enstitü : Fen Bilimleri Enstitüsü
Anabilim Dalı : İstatistik
Mezuniyet Yılı : 2017

Akademik Yayınlar

VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
LİSANSÜSTÜ TEZ ORİJİNALLİK RAPORU

Tarih 12/05/2023

Tez Başlığı: Derin Pekiştirmeli Öğrenme ile Kripto Para Fiyat Tahmini

Yukarıda başlığı belirtilen tez çalışmamın, kapak sayfası, giriş, ana bölümler ve sonuç bölümlerinden oluşan toplam altmış üç (63) sayfalık kısmına ilişkin, 12/05/2023 tarihinde şahsım/tez danışmanım tarafından turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre tezimin benzerlik oranı % 2 (yüzde iki) dir.

Uygulanan filtreler aşağıda verilmiştir:

- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
- Tezden çıkan yayınlar hariç,
- 7 kelimeden daha az örtüşme içeren metin kısımları hariç (Limit match size to 7 words)

Van Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi inceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.

Tarih ve İmza
12.05.2023

Adı Soyadı: Serbest ZİYANAK
Öğrenci No: 17910002105
Anabilim Dalı: İstatistik
Programı: Uygulamalı İstatistik
Statüsü: ( ) Yüksek lisans (x) Doktora

DANIŞMAN	**ENSTİTÜ ONAYI**
UYGUNDUR	UYGUNDUR