T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ÇEVRİM İÇİ ORTAMLARDA YAPILAN SAHTE KULLANICI YORUMLARININ TESPİTİNDE DERİN ÖĞRENME KULLANIMI

KENAN TAŞAĞAL

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Dr. Öğr. Üyesi Özlem UÇAR

EDİRNE-2019

KENAN TAŞAĞAL 'ın hazırladığı "Çevirim İçi Ortamlarda Yapılan Sahte Kullanıcı Yorumlarının Tespitinde Derin Öğrenme Kullanımı" başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisliği Anabilim Dalında bir Yüksek lisans tezi olarak kabul edilmiştir.

Jüri Üyeleri (Ünvan, Ad, Soyad): İmza

Doç. Dr. İlhan UMUT .....

Dr.Öğr.Üyesi Özlem UÇAR .....

Dr.Öğr.Üyesi Edip Serdar GÜNER .....

Tez Savunma Tarihi:09/07/2019

Bu tezin Yüksek Lisans tezi olarak gerekli şartları sağladığını onaylarım.

İmza

Dr. Öğr. Üyesi Özlem UÇAR .....

Tez Danışmanı

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

Prof. Dr. Murat YURTCAN

Fen Bilimleri Enstitüsü Müdürü

T.Ü. FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI YÜKSEK LİSANS PROGRAMI

DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

09 / 07 /2019

Kenan TAŞAĞAL

İmza

Yüksek Lisans Tezi

Çevrim İçi Ortamlarda Yapılan Sahte Kullanıcı Yorumlarının Tespitinde Derin Öğrenme Kullanımı

T.Ü. Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

İnternet teknolojilerinin günlük hayatımızın her alanında aktif bir şekilde yer alması internet kullanıcılarının ürün ya da hizmet satın alımı davranışlarında etkili olmaktadır. Kullanıcılar e-ticaret sitelerinde birçok alternatifi bulunan ve ihtiyaçlarına en uygun şekilde cevap verebilecek olan ürün ya da hizmet satın alımı konusunda zorlanmaktadırlar. Bu nedenle kullanıcılar satın almak istedikleri ürün hakkında daha önceki deneyimlerini çevrim içi ortamda paylaşan kullanıcıların yorumlarından sıklıkla faydalanmaktadırlar. Bu yorumlar kullanıcıların satın alma davranışlarında etkili olmaları, kar veya tanıtım amacı ile kullanıcıları yanıltmaya yönelik sahte ürün yorumların artmasına sebep olmuştur. İnternet kullanıcılarını bu tip yorumların etkisinden kurtarmak ve kullanıcılara daha sağlıklı bilgi sunabilmek için sahte yorumların tespiti son zamanlarda akademik çevrelerde ilgi uyandırmaktadır.

Bu tez çalışmasında derin öğrenme modellerinden yararlanılarak sahte kullanıcı yorumlarının tespitinde bu modellerin etkinliği test edilmiştir. Sahte kullanıcı yorumlarının tespitinde daha önce yapılan çalışmalar incelenerek Ott ve arkadaşlarının "Gold Standart" olarak tanımlanan 20 Chicago oteli hakkındaki doğru ve aldatıcı yorumları toplayarak oluşturdukları veri seti kullanılmıştır. Çalışmada veri seti üzerinde metin sınıflandırma yöntemlerinden yararlanarak oluşturduğumuz eğitim kümesi ile LSTM, BiLSTM ve CNN+LSTM modelleri kullanarak sahte kullanıcı yorumlarının tespitinde yüksek başarım elde etmeyi amaçlamaktayız.

Yıl : 2019

Sayfa Sayısı : 76

Anahtar Kelimeler :Aldatıcı yorum tespiti, Makine Öğrenmesi, Derin Öğrenme, LSTM

Master Thesis
Use of Deep Learning to Detect Fake User Comments Made Online
Trakya University Institute of Natural Sciences
Department of Computer Engineering

ABSTRACT

The fact that internet technologies play an active role in every aspect of our daily lives is effective on the behaviors of internet users to purchase products or services. Users have difficulty in purchasing products or services that have many alternatives on their e-commerce sites which can respond to their needs in the most appropriate way. Therefore, users often benefit from the comments of users who have shared their previous experiences about the product they want to buy online. These comments have led to an increase in fake product reviews to mislead users for profit or promotional purposes, as well as being effective in users' buying behavior. The detection of fake comments has recently aroused interest from the academic community in order to free internet users from the impact of such comments and to provide users with more accurate information.

In this thesis, the effectiveness of these models in the detection of fake user comments has been tested by using deep learning models. In the detection of fake user reviews, the previous studies were examined and the data set of Ott and colleagues gathering the correct and deceptive comments about 20 Chicago hotels defined as “Gold Standard” was used. In this study, we aim to achieve high performance in detecting fake user comments by using LSTM, BiLSTM and CNN + LSTM models with the training set we created using text classification methods on the data set.

Year: 2019

Number of Pages: 76

Key Words: Deceptive interpretation detection, Machine Learning, Deep Learning, LSTM

ÖNSÖZ

Bu tez çalışması boyunca sağladığı yardım ve destek için danışmanım Sayın Dr. Öğr. Üyesi Özlem UÇAR'a teşekkürlerimi sunarım.

Tez çalışmasının her aşamasında yanımda olan ve her zaman desteğini hissettiren sevgili eşim Burcu Taşağal'a ve biricik kızım Bilge'ye çok teşekkür ederim.

Ayrıca yaşamımın tüm safhalarında desteklerini benden esirgemeyen anneme, babama ve ablalarıma ayrıca minnet ve şükranlarımı sunarım.

Kenan TAŞAĞAL

EDİRNE-2019

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

ŞEKİLLER DİZİNİ ..... **ix**

ÇİZELGE DİZİNİ ..... **xi**

KISALTMALAR DİZİNİ ..... **xii**

BÖLÜM 1 ..... **1**

GİRİŞ **1**

BÖLÜM 2 ..... **3**

LİTERATÜR ARAŞTIRMASI **3**

BÖLÜM 3 ..... **8**

MATERYAL VE METODLAR **8**

3.1. Makine Öğrenmesi 8

3.2. Yapay Sinir Ağları ve Derin Öğrenme 9

3.2.1. Algılayıcılar 10

3.2.2. Aktivasyon Fonksiyonları 13

3.2.2.1. Sigmoid 13

3.2.2.2. Tanh 14

3.2.2.3. ReLU 15

3.2.3. İleri Besleme Ağları 15

3.2.3.1. Tek Katmanlı Algılayıcı 16

3.2.3.2. Çok Katmanlı Algılayıcılar: 17

3.2.3.3. Geri Beslemeli Ağlar 18
3.3. Derin Öğrenme(Deep Learning) 19
3.3.1. Makine Öğrenmesinin Zorlukları 21
3.3.2. Konvolüsyonel Sinir Ağları(CNN) 23
3.3.3. Tekrarlayan Sinir Ağları(RNN) 25
3.3.4. Uzun Kısa Sürekli Bellek Ağları(LSTM) 27
3.3.5. Uzun Kısa Süreli Belleğe Sahip Çift Yönlü Tekrarlayan Sinir Ağları (BiLSTM) 31
3.3.6. Derin Öğrenme Geliştirme Ortamları 33
3.3.6.1. Anaconda 33
3.3.6.2. TensorFlow 34
3.3.6.3. Keras 36
BÖLÜM 4 38
DENEYLER VE SONUÇLAR 38
4.1. Veri Seti 38
4.2. Veri Seti Ön İşlemi 41
4.3. Değerlendirme Ölçütleri 44
4.4. Deney Sonuçları 46
4.4.1. LSTM Modeli Eğitimi ve Test Sonuçları 46
4.4.2. BiLSTM Modeli Eğitimi ve Test Sonuçları 50
4.4.3. CNN+LSTM Modeli Eğitimi ve Test Sonuçları 52
BÖLÜM 5 55
SONUÇ 55
KAYNAKLAR 57
ÖZGEÇMİŞ 63
TEZ ÖĞRENCİSİNE AİT TEZ İLGİLİ BİLİMSEL FAALİYETLER 64

ŞEKİLLER DİZİNİ

Şekil 3. 1. Doğrusal Eşik Kapısının sembolik çizimi. 11
Şekil 3. 2. Doğrusal eşik fonksiyonu 11
Şekil 3. 3. Algılayıcı modeli 12
Şekil 3. 4. Sigmoid fonksiyonun grafiği 14
Şekil 3. 5. Tanh fonksiyonun grafiği 14
Şekil 3. 6. ReLu fonksiyonun grafiği 15
Şekil 3. 7. Tek katmanlı algılayıcı sinir ağı gösterimi 17
Şekil 3. 8. Çok katmanlı algılayıcı sinir ağı gösterimi 18
Şekil 3. 9. Geri beslemeli yapay sinir ağları 19
Şekil 3. 10. Derin Öğrenme ve Makine Öğrenmesi öğrenim şeması 20
Şekil 3. 11. Derin öğrenme modeli gösterimi 21
Şekil 3. 12. Verideki ilgili boyutların sayısı arttıkça(soldan sağa), ilgilendiğimiz düzlemlerin sayısı da katlanarak artar.(Boyut laneti) 22
Şekil 3. 13. Örnek bir konvolüsyon işlemi 24
Şekil 3. 14. Basit bir CNN Mimarisi 25
Şekil 3. 15. RNN modeli basit gösterimi (McGonagle, Williams, & Khim, 2019) 27
Şekil 3. 16. LSTM mimarisi 28
Şekil 3. 17. LSTM hücresi durumunun yapısı (Wang , Xia, Liu, Li, & Li, 2017) 29
Şekil 3. 18. Çift yönlü RNN'in kontrolsüz yapısı. 32
Şekil 3. 19. Arka arkaya 3 adımda katlanmış BiLSTM mimarisi (Cui, Ke, & Wang, 2017) 33
Şekil 3. 20. Anaconda Ortamı 34
Şekil 3. 21. Tensor boyut gösterimi 35
Şekil 3. 22. Derin öğrenme için framework kullanım oranları (2018 yılı verileri) 35

Şekil 4. 1. Veri setinde bulunan tüm yorumların kelime dağılımları 43
Şekil 4. 2. (a) Gerçek yorumlarda kelime dağılımı **(b)** Sahte yorumlarda kelime dağılımı 43

Şekil 4. 3. LSTM model mimarisi 48
Şekil 4. 4. LSTM modeli eğitim(Train) ve doğrulama(Test) sonuçları 49
Şekil 4. 5. BiLSTM model mimarisi 50
Şekil 4. 6. BiLSTM modeli eğitim(Train) ve doğrulama(Test) sonuçları 51
Şekil 4. 7. CNN+LSTM model mimarisi 52
Şekil 4. 8. CNN+LSTM modeli eğitim(Train) ve doğrulama(Test) sonuçları 53

ÇİZELGE DİZİNİ

Çizelge 2.1. Üç tür yinelenen spam incelemesi 4

Çizelge 2.2. Aldatıcı inceleme spam tespit tekniklerinin karşılaştırılması 7

Çizelge 4.1. Ott ve arkadaşlarının veri kümesindeki yorum grupları ve adetleri 39

Çizelge 4.2. Veri seti gösterimi 42

Çizelge 4.3. Veri setinden gürültülerin temizlenmesi 42

Çizelge 4.4. Karşıtlık Matrisi 44

Çizelge 4.5. LSTM Model test sonuçları 49

Çizelge 4.6. BiLSTM Model test sonuçları 51

Çizelge 4.7. CNN+LSTM Model test sonuçları 54

KISALTMALAR DİZİNİ

CNN : Evrişimsel Sinir Ağları(Convolutional Neural Network)

LSTM : Uzun Kısa Süreli Bellek Ağı(Long Short-Term Memory)

BiLSTM : İki Yönlü LSTM(Bidirectioanal LSTM)

RNN : Tekrarlayan Sinir Ağı(Recurrent Neural Network)

CV : Çapraz Doğrulama(Cross Validation)

DVM : Destek Vektör Makineleri

YSA : Yapay Sinir Ağı

NLP : Doğal Dil İşleme(Natural Language Processing)

AMI : Amazon Makine Görüntüsü(Amazon Machine Images)

NTLK : Doğal Dil Aracı(Natural Language Toolkit)

CPU : Merkezi İşlem Birimi(Central Processing Unit)

GPU : Grafik İşlem Birimi(Graphics Processing Unit)

ReLu : Doğrultulmuş Lineer Birim(Rectified Linear Unit)

BÖLÜM 1

GİRİŞ

Gelişen teknoloji ile birlikte gündelik hayatımızın içerisinde yer alan çevrim içi ortamlar, tüketicilerin satın alma alışkanlıklarında köklü değişikliklere sebep olmuştur. Çevrim içi ortamların sağlamış olduğu hizmetlerden faydalanmak isteyen kullanıcılar, farklı kullanıcıların deneyimlerini ve görüşlerini belirttiği yorumlardan yararlanarak almak istedikleri ürün veya hizmetlerin kalitesi hakkında fikir sahibi olmaktadırlar. Ürün tanıtımları ve reklamlarından farklı olarak bu kullanıcı yorumları, tüketicilerin kararlarını ve alışveriş davranışlarını fazlasıyla etkilemektedir. Luca tarafından yapılan çalışmada, bir ürün veya işletmenin çevrim içi ortamlarda kullanıcılar tarafından yapılan derecelendirmesinin +1 puan arttığında, ürün ya da işletmenin gelirinin %5 ile %9 arasında arttığı gösterilmektedir. (Luca, 2011) Bu gelir artışını göz önünde bulunduran işletmeler ve üreticiler, tüketicilerin satın alma kararlarını etkilemek için sahte çevrim içi kullanıcı yorumları kullandıkları bilinmektedir. Günümüzde kullanıcıların satın alma kararlarını etkilemek amaçlı sahte kullanıcı yorumları yazan şirketler de bulunmaktadır. Tüketiciyi yanıltma amaçlı yapılan sahte kullanıcı yorumlarının yaygın bir şekilde kullanılması sahte yorumların tespitini zorunlu hale getirmektedir.

İnternet pazarındaki hızlı gelişim, e-ticaret alanında rekabetin boyutlarının artmasına ve pazarlama taktiklerinin değişmesine neden oldu. Firmalar, markalarını veya ürünlerini ön plana çıkarabilmek için çevrim içi tüketicilerin kararlarını doğrudan etkileyebilen sahte yorumlardan faydalanmaktadır. Tüketicilerin satın alma kararlarını olumlu ya da olumsuz yönde etkileyebilmek için yapılan bu yorumların kullanılmasının

önüne geçebilmek amacıyla yapılan yasal düzenlemeler kontrol mekanizmasının zayıf olmasından dolayı etkili olamamaktadır. (Crawford, Khoshgoftaar, Prusa, Richter, & Najada , 2015) Günümüzde Amazon, TripAdvisor gibi kullanıcı yorumlarının sıklıkla kullanıldığı e-ticaret siteleri, kullanıcılarına daha iyi hizmet sunabilmek için bu tip yorumların tespitini yapabilmeye odaklı uygulamalar kullanmakta ve geliştirmektedirler. İstatistiklere göre, Orbitz, Priceline, Expedia ve TripAdvisor gibi seyahat sitelerinde sahte kullanıcı yorumlarının, %2 ile %6 arasında bir paya sahip olduğunu görülmektedir. Bu sitelerden farklı olarak restaurant, otel, kafe tarzı işletmeler hakkında üyelerinden yorum alan ve bu yorumları yayınlayan yelp.com sitesinde sahte kullanıcı yorumlarının %14 ile %20 oranında bulunduğu gözlenmektedir. (Ren & Ji, 2019)

Büyük miktarda sahte yorum, kullanıcıların çevrim içi yorumlara duyduğu güveni azaltmaktadır. Sahte kullanıcı yorumlarını otomatik olarak tanımlayan etkili modeller tasarlamak kullanıcıların güvenini kazanmak açısından önemlidir. Geçmişte bu konu üzerine makine öğrenmesi tekniklerine dayalı bir dizi çalışma yapılmıştır. (Heydari, Tavakoli, Salim, & Heydari, 2015) Yapılan bu çalışmaların en önemli yönü veri setleri olmuştur. Sahte kullanıcı yorumlarının tanımlanması ve tanımlanan bu yorumların gerçekliğinin tespit edilmesi maliyetli ve zor bir çalışmadır.

Çevrim içi sahte kullanıcı yorumları genel olarak içerik bakımından çeşitli türlerden ve kısa metinlerden oluşmaktadır. Mevcut yaklaşımların bu tür kısa metinlere uyarlanmasında zorluk çekilmekte ve sahte yorumların tespitinde etkili olamamaktadır. Bu sebepten kullanıcı yorumlarının gerçekliğinin tespiti üzerine yapılan "Derin Öğrenme" çalışmaları son yıllarda önem kazanmıştır.

Biz bu çalışmamızda derin öğrenme modelleri kullanarak çevrim içi sahte kullanıcı yorumlarının tespitinde şimdiye kadar yapılan çalışmalardan daha etkili bir sonuç elde etmeyi amaçlıyoruz. Çalışmamızda derin öğrenme modellerinden LSTM, BiLSTM ve CNN+LSTM modellerini kullanarak çevrim içi sahte yorumların tespitindeki doğruluk oranlarını karşılaştırdık ve bu 3 modelden en yüksek doğruluk oranına sahip olan modeli tespit ettik.

BÖLÜM 2

LİTERATÜR ARAŞTIRMASI

Günümüzde tüketiciler almak istedikleri hizmet veya ürünler hakkında fikir sahibi olabilmek için çevrim içi derecelendirmeler, incelemeler ve yorumlardan giderek daha fazla yararlanmaktadırlar. Bu nedenle tüketici incelemeleri içeren web siteleri, sahte ve yanıltıcı yorumların hedefi olmaktadır. Hedef olan bu web sitelerinde tüketiciyi yanıltma amaçlı yapılan sahte kullanıcı yorumlarının artması sahte yorumların tespitini zorunluluk haline getirmektedir. 2007 yılında Jindal ve Liu yayınladıkları "Eleştiri Spam Algılama" çalışmaları alandaki ilk çalışmadır. Çalışmalarında farklı kullanıcı hesaplarının bir ürün ile ilgili ya da farklı ürünler ile ilgili aynı yorumların yapıldığını Shingle metodunu (Broder, June 1997) kullanarak tespit etmişlerdir. Tespit ettikleri yorumları aldatıcı yorum olup olmama durumuna göre sınıflandırmışlardır. (Jindal & Liu, 2007)

N. Jindal ve B. Liu bir sonraki makalelerinde Amazon.com web sitesinden aldıkları kullanıcı yorumlarını inceleyerek, sahte yorumları genel olarak 3 tipe ayırmışlardır. (Jindal & Liu, 2008)

1) Kullanıcıyı olumlu veya olumsuz etkilemeye çalışan gerçek olmayan yorumlar
2) Marka veya ürünü ön plana çıkartmak amaçlı yorumlar
3) Yorum olmayan tüketiciyi yanıltıcı metinler

Jindal ve Liu bu yaptıkları sınıflandırma ile ikinci ve üçüncü tip yorumların tespitinin kolaylıkla yapılabildiğini ama birinci tip yorumların tespitinin çok zor olduğunu

belirlemişlerdir. Çizelge 2.1'de Jindal ve Liu'nun yaptıkları çalışmada elde ettikleri sonuçlar verilmiştir. (Jindal & Liu, 2008)

Çizelge 2.1. Üç tür yinelenen spam incelemesi

	Spam İnceleme Türü	**İncelenen Yorum Sayısı (İncelenen ürün Sayısı)**
1	Aynı üründeki farklı kullanıcılar	3067 (104)
2	Farklı ürünlerde aynı kullanıcı kimliği	50869 (4270)
3	Farklı ürünlerde farklı kullanıcılar	1383 (114)
	Toplam	55319 (4488)

,Jindal ve arkadaşları spam niteliğindeki yorumlara göre çeşitli yaklaşımlar önermiştir. Önerdikleri yaklaşımları kategorize etmek istediğimizde;

- İstenmeyen kullanıcıların dil kalıplarını analiz eden dilbilimsel yaklaşımlar
- Kullanıcıların gözden geçirme davranışlarını kullanan davranışsal yaklaşımlar
- Kullanıcılar, incelemeler ve ürünler arasındaki ilişkisel bağları analiz eden grafik tabanlı yöntemler.

olarak kategorize edilebilir.

Mevcut yaklaşımlar ayrıca sahte incelemeleri, spam kullanıcılarını veya spam kullanıcı gruplarını tespit edenler olarak iki şekilde gruplandırılabilir. (Rayana & Akoglu, 2015)

Aldatıcı fikir spamı, gerçek olmayan ve kullanıcıları aldatmaya yönelik gerçek kimliğe sahip kişiler tarafından kasıtlı bir şekilde hayali görüş ve değerlendirmelerini belirttikleri bir yorum türüdür. (Jindal & Liu, 2008) Çevrim içi ortamlarda yapılan tüm

kullanıcı yorum ve incelemelerinin üçte birinin aldatıcı fikir spamı olduğu tahmin edilirken (D'Onfro, 2013) temel zorluk öğrenilebilecek doğru verilerin elde edilmesi ve bu verilerin doğru bir şekilde etiketlenebilmesidir.

Bu zorluğu aşmak amacıyla Ott ve arkadaşlarının oluşturdukları aldatıcı fikir spamı alanındaki ilk halka açık veri setinden (Ott, Choi, Cardie, & Hancock, 2011) birer sahte ve gerçek kullanıcı yorumu aşağıda verilmiştir:

Birinci Yorum:

"Hem iş hem de eğlence için seyahatlerimde birçok otelde kaldım ve dürüstçe James'in kaldığım diğer otellerden daha iyi olduğunu söyleyebilirim. Otelde servis birinci sınıftır. Odalar modern ve çok rahat. Konumu mükemmel, harika turistik yerlere ve restoranlara yürüme mesafesinde. Hem iş seyahatinde olanlar hem de çiftler için önerilmektedir."

İkinci Yorum:

"Kocam ve ben yıldönümümüz için James Chicago Hotel'de kaldı. Burası harika! Geldiğimizde doğru seçimi yaptığımızı biliyorduk! Odalar güzel ve çok özenli ve personel harika! Otelin alanı harika, alışveriş yapmayı sevdiğim için daha fazla bilgiye gerek duymadım! Kesinlikle Şikago'ya döneceğiz ve kesinlikle James Şikago'ya döneceğiz."

Her iki yorumda gerçek kimlikli kullanıcılar tarafında yapılmıştır. Birinci inceleme doğru ve gerçek bir yorumdur. İkinci inceleme ise aldatıcı fikir spamıdır. Bu iki inceleme ile sahte kullanıcı yorumlarının tespitini açıklamak oldukça zordur. Ott ve arkadaşları manuel olarak yapıkları çalışmalarında sadece %60 doğruluk oranına ulaşabilmişlerdir. (Ott, Choi, Cardie, & Hancock, 2011)

Bu tez çalışmasında kullanılan veri setini oluşturan Ott ve arkadaşları, sahte yorumların tespiti için yaptıkları ilk çalışmada "Gold Standart" (Ott, 2019) adını verdikleri veri kümesini oluşturdular ve bu veri kümesi üzerinde 5-katlı (5-fold) çapraz doğrulama(CV) yaparak Naive Bayes (NB) ve Destek Vektör Makineleri (DVM) yöntemini test etmişlerdir. Sonuç olarak %86 doğruluk oranıyla olumsuz sahte yorumların tespitine ulaşmışlardır. (Ott, Cardie, & Hancock, 2013)

Qian ve arkadaşları oluşturdukları farklı bir veri kümesi üzerinde lojistik regresyon, doğrusal ayırıcı analiz, çok terimli Naif Bayes, DVM ve sinir ağlarını kullanarak yaptıkları çalışmalarında, sinir ağlarını kullanarak oluşturdukları öğrenme algoritması % 81.92'lik doğruluk oranı ile sahte yorumların tespitinde en iyi performansı göstermiştir. (Wang, Zhang, & Qian, 2017)

2015'in başında sahte kullanıcı yorumlarının tespiti için kullanılan teknikleri özetleyen iki çalışma yapılmıştır. (Crawford,2015; Heydari,2015) Ancak bu çalışmalar, sinir ağları ile ilgili teknikleri ve özellikle son yıllarda hızlı bir gelişme sağlayan derin öğrenme tekniklerini içermemektedir. (Ren & Ji, 2019) Yapılan bu çalışmalar günümüz şartlarına göre eksik kalmaktadır.

Ren ve Zhang, aldatıcı görüş spamlerini tespit etmek için bir sinir ağı modelini deneysel olarak araştırmıştır. (Ren & Zhang, 2016) Çalışmalarında önerdikleri sinir ağı modeli geleneksel modellerden daha iyi performans sağlamaktadır.

Wang ve arkadaşları yaptıkları çalışmalarında spam gönderenleri tespit etmek için derin öğrenme modeli olana LSTM üzerine yoğunlaşmışlardır. (Wang, Day, Chen, & Liou, 2018) Yaptıkları deneylerde Tayvan'da gerçekleşen sahte yorumları kullandılar ve mevcut çalışmanın analitik sonuçlarını önceki literatür sonuçlarıyla karşılaştırdılar. LSTM'nin sahte incelemeleri tespit etmede DVM'den daha etkili olduğunu buldular.

Çizelge 2.2'de sahte kullanıcı yorumu tespiti üzerine etiketli veri setlerine dayalı olarak yapılan çalışmaların yöntemleri ve doğruluk oranları verilmektedir.

Çizelge 2.2 Sahte kullanıcı yorumları üzerine etiketli verilere dayalı olarak yapılan çalışmaların karşılaştırılması

Yazarlar	Ana kavram	Özellikler	Yöntem	Sonuç
Jindal ve Liu	Metin çoğaltma	İnceleme, inceleme ve ürün odaklı	Lojistik regresyon	% 78 (doğruluk)
Lai ve diğ.	Metin benzerliği	Metni incele	DVM	% 81 (hassasiyet)
Algur ve diğ.	Ürün özelliği benzerliği	Ürün Özellikleri	Kosinüs benzerliği	% 43,6 (hassasiyet)
Ott ve diğ.	İçerik benzerliği	LIWC + Bigram	DVM	89.6 (doğruluk)
Ott ve diğ.	İçerik incelemesi (Yalnızca olumsuz yorumlar almak)	*n*- gram	DVM	% 86 (doğruluk)
Mukherjee ve diğ.	İçerik benzerliği	Davranış + Bigram	DVM	% 86.1 (doğruluk)
Shojaee ve diğ.	Stylometric	Sözcüksel ve sözdizimsel	DVM	% 84 (F-puanı)
Long ve diğ.	Ontoloji	Ontolojik özellikler	Koşullu filtreleme	% 75 (hassasiyet)

BÖLÜM 3

MATERYAL VE METODLAR

3.1.Makine Öğrenmesi

Günümüzde bilgisayar teknolojilerinin ve internetin insanlığın her alanında kullanılıyor olması büyük miktarda verinin oluşmasına neden olmaktadır. Örneğin tıp alanında kanser hastaları üzerinde yapılan testlerden elde edilen veriler sayesinde kanser hastalarına erken teşhis koyma ve müdahale etme şansı artmıştır. Ancak bu veri kümelerinin büyüklüğü yüzünden klasik veri tabanı sistemleri ile işlenmesinin ve analizinin gerçekleştirilmesi mümkün değildir. Bu sorunu çözmek ve elde edilen büyük veri kümelerinden daha etkin şekilde yararlanabilmek için günümüzdeki çalışmalar geçmiş deneyimlere ve gözlemlere dayanarak geleceğe yönelik kararları alabilecek sistemlerin nasıl gerçekleşebileceğine odaklanmaktadır. (LeCun, Bengio , & Hinton, Deep learning, 2015)

Makine öğrenmesi, geçmiş deneyimlerin veya örnek veri kümelerinin kullanılarak bir performans kriterinin optimize edilmesi için bilgisayarların programlamasıdır. (Alpaydın, 2010) Basit bir tanımla makine öğrenmesi veriden üretilen yazılımlardır. (Tynan, 2017)

Bir problemin çözümü matematiksel bir denklemin sonucu olabildiği durumlarda klasik yazılımlar yardımı ile bu problemin çözümü çok basit olmaktadır. Çünkü matematiksel modellemesi yapılabilen bir denklemin algoritmasını oluşturmak mümkündür. Ancak cevap aranan problemin matematiksel bir modellemesi

yapılamıyorsa klasik yazılımlar problemin çözümünde zorlanmaktadır. Böyle durumlarda makine öğrenmesi algoritmalarından yararlanılmaktadır.

Makine öğrenmesindeki temel amaç, elde edilen geçmiş deneyimleri kullanarak gelecek hakkında karar vermektir. (Nasrabadi, 2007) Makine öğrenmesi algoritmaları eldeki verilerden öğrenebilen algoritmalardır. Mitchell (Mitchell, 1997), öğrenebilme kavramını şu şekilde tanımlamıştır;

"Bir performans ölçütü P ve T görevleri ile belirlenmiş E deneyimleri üzerinde tanımlanan bir bilgisayar programı düşünelim. Eğer bu bilgisayar programının P performansı, E deneyimi ile artıyorsa, program öğreniyor demektir."

Örneğin, spam mail filtrelemelerini yapabilen bir bilgisayar programı geçmişte spam mail olarak bilinen mailleri gözlemleyerek spam mail tespitini öğrenir. Bu problemde, gelen mailler içerisinde spam maillerin tespiti görev T'dir. Bilgisayar programı gelen maillerin arasından spam ve spam olmayan mail şeklinde sınıflandırması P ölçütüdür. T görevinde P ölçütü ile sınıflandırılmış maillerin bilgisayar programı ile spam ya da spam olmadığını öğrenmesi E deneyimidir. Bilgisayar programı E deneyimi sonrasında doğruluk yüzdelerini hesaplayarak gelecek maillerde spam mail tespiti artıyorsa bu programın öğrenmiş olduğunu göstermektedir.

Makine öğrenmesi günümüzde e-ticaret, spam yakalama, tıbbi tanı, biyometrik kimlik tanıma, doğal dil işleme, bilgisayarlı görme, reklam gibi daha birçok alanda kullanılmaktadır.

Makine öğrenmesi algoritmaları çok çeşitli ve önemli problemlerde iyi performans göstermektedir. Ancak bu algoritmalar konuşma tanıma ve nesne tanıma gibi birçok yapay zeka merkezli problemlerde başarılı olamamıştır.

3.2. Yapay Sinir Ağları ve Derin Öğrenme

Beyin evrendeki en karmaşık organdır. Nörobiyoloklara göre, beyin bir girdiye milisaniye içinde cevap veren 10^{10} işlemciye sahip paralel çalışan bir bilgisayara benzer. (McCollum, 2019) Makine öğrenmesi ve derin öğrenmenin ilham kaynağı insan beynidir.

Beyin sürekli öğrenen ve değişen esnek bir yapıya sahiptir. Beynin yetenekleri, bilgisayarın yeteneklerinin çok ama çok ötesindedir. Ortalama insan beyni 100 milyardan fazla nörondan oluşmaktadır. Nöronlar, birbirleri ile ileşitim kurabilmek için sinaps adı verilen yapısal ve fonksiyonel bağlantıları kullanırlar. Her bir nöron 10 bine yakın nöron ile bağlantılıdır. Bu durumda sinaps sayısı 100 trilyon ile 1000 trilyon arasında olabilir. Beyin yeni bir uyarı ile karşı karşıya kaldığında nöronlar arasındaki karşılıklı bağlantılar değişir. Bu biçimde yeni bağlantıların oluşması varolan bağlantıların güçlenmesi ve kullanılmayan bağlantıların zayıflayarak ortadan kaldırılması söz konusudur.

Yapay nöronların yapısı biyolojik nöronların yapısına benzemektedir. Yapay nöronlarda bulunan işlem birimi hücre gövdesi ya da çekirdeğe, girdiler dentritlere ve çıktılar aksonlara benzer şekilde çalışmaktadır. Yapay nöronlar da tıpkı biyolojik nöronlar gibi birbirlerine sinapslara benzer yapıda bağlanarak yapay sinir ağını oluşturmaktadırlar. Cajal, nöronlar arasındaki sinaptik bağlantıların değişebilir olduğunu ve öğrenme yolu ile bağlantıların güncellenebileceği öne sürmüştür. Daha sonraki çalışmalarında Cajal, sinaps sayısındaki artışın öğrenme ve hafıza mekanizmalarına etkisini kanıtlamıştır. (Mayford, Siegelbaum, & Kandel, 2012)

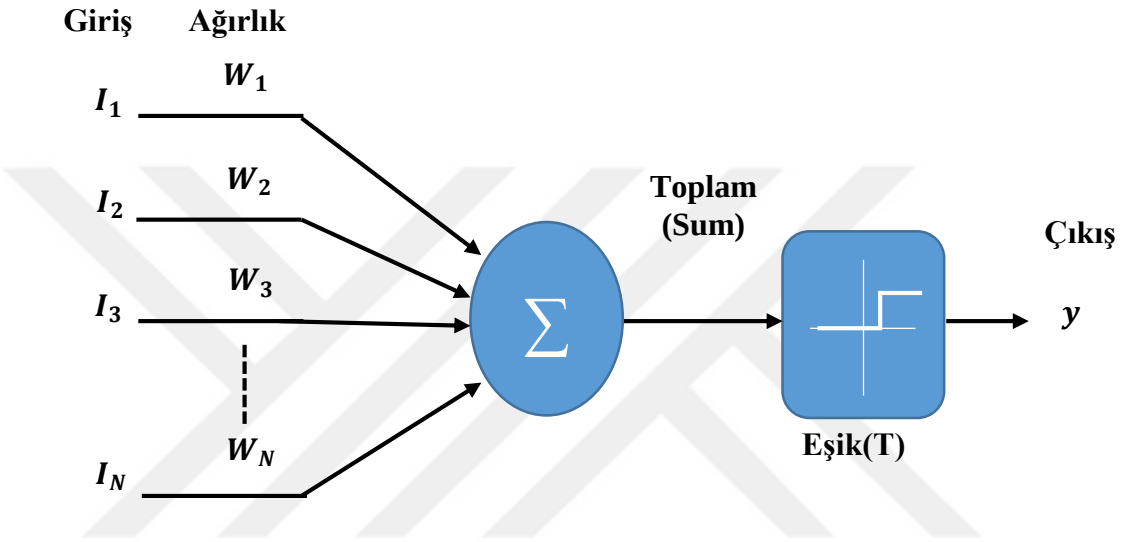
Yapay sinir ağlarının başlangıç noktası Waren McCulloch ve Walter Pitts'in biyolojik bir nöronu yapay olarak modelledikleri çalışmalarıdır. (McCulloch & Pitts, 1943) 1957 yılında Frank Rosenblatt algılayıcı olarak adlandırılan sinir ağlarının ilk yayınlanan algoritmasını keşfetmiştir. (Rosenblatt, 1958)

3.2.1.Algılayıcılar

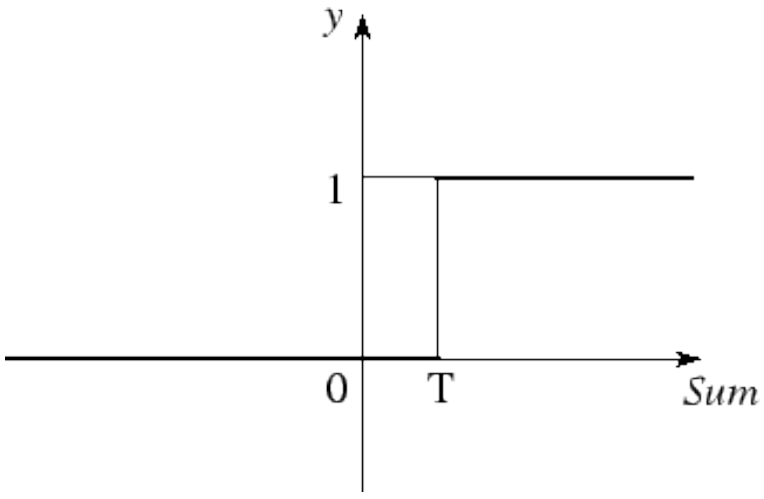
1943 yılında Warren McCulloch ve Walter Pitts doğrusal eşik kapısı (linear threshold model) olarak da bilinen ilk yapay nöron modelini tanıttı. McCulloch-Pitts modelinde kullanılan doğrusal eşik geçidi, girdileri iki farklı değerle sınıflandırır ve sonuç olarak ikili bir çıktı üretir. Matematiksel olarak bu fonksiyon Denklem 3.1'de görüldüğü gibi ifade edilir:

$$Sum = \sum_{i=1}^{N} I_i W_i \ , \ y = f(Sum) \qquad (3.1)$$

Şekil 3.1'de McCulloch-Pitts modelinin sembolik gösterimi ve Şekil 3.2'de T eşik değerindeki doğrusal fonksiyonu verilmektedir.



Şekil 3. 1. Doğrusal Eşik Kapısının sembolik çizimi.



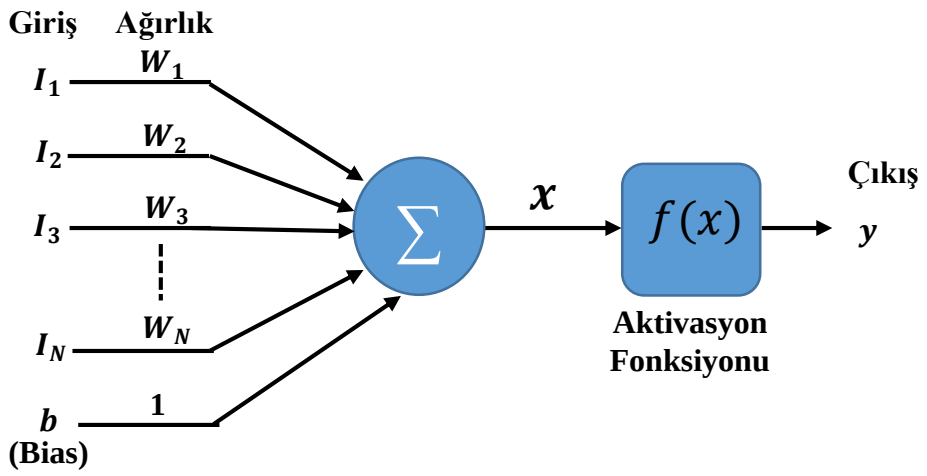
Şekil 3. 2. Doğrusal eşik fonksiyonu

McCulloch-Pitts modeli matematiksel olarak basit ve kesin bir şekilde tanımlanır. Aynı zamanda model önemli bir hesaplama potansiyeline sahiptir. Fakat modelin farklı özelliklere sahip nöral hesaplamalarda uygulanabilirliği zordur.

1957 yılında Frank Rosenblatt, McCulloch-Pitts modeli ile Hebbian'nın ağırlık ayarlama kuralını birleştirerek yeni bir yapay sinir ağ tasarladı. Rosenblatt'ın algılayıcı(perceptron) olarak adlandırdığı bu model, McCulloch-Pitts modelinin değiştirilmiş bir türevidir. Algılayıcıların girdileri, McCulloch-Pitts modelinden farklı olarak değişken ağırlık değerleri ve bias adı verilen ek bir ön yargı değeridir. Değiştirilmiş denklem aşağıdaki gibidir.

$$Sum = \sum_{i=1}^{N} I_i W_i + b \tag{3.2}$$

Şekil 3.3'de gösterilen algılayıcı modelinde giriş dizisinin her değeri, normalde 0 ile 1 arasında olan ağırlık değeriyle ilişkilendirilir. Ayrıca toplama işlevi, bir nöron eşiğini veya bias değerini temsil etmek için genellikle ağırlık değeri 1 olan ekstra bir giriş değeri alır.



Şekil 3. 3. Algılayıcı modeli

3.2.2.Aktivasyon Fonksiyonları

Aktivasyon fonksiyonlarıysa YSA'da daha iyi performans alabilmek için kilit rol oynar. Aktivasyon fonksiyonlarına, doğrusal olmayan gerçek dünya özelliklerini yapay sinir ağlarına tanımlamak için ihtiyaç duyulur.

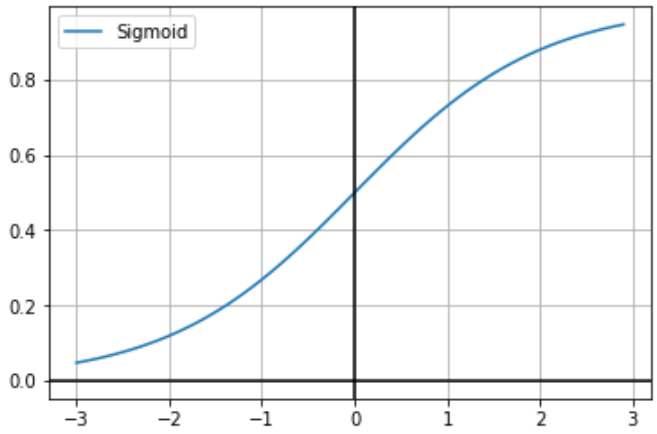
Aktivasyon fonksiyonlarının kullanılmadığı bir sinir ağında görüntü, video, yazı ve ses gibi karmaşık dünya bilgilerinin öğrenilmesi zordur. Çünkü sinir ağı sınırlı öğrenme gücüne sahip olacaktır. Yapay sinir ağlarında asıl istenen doğrusal olmayan durumlarda da öğrenimin gerçekleşmesidir.

YSA'da kullanılan çeşitli aktivasyon fonksiyonları bulunmaktadır. Nöronlar seçilen aktivasyon fonksiyonuna göre aktive olurlar. Bir nöron aktivasyon fonksiyonunun eşik değeri üstünde olursa aktive edilir eğer nöron eşik değerinin altında kalırsa aktive edilmez ve aktive edilmeyen nöronlar kullanılmaz. Yaygın olarak tercih edilen aktivasyon fonksiyonları ReLU, Sigmoid ve Tanh fonksiyonlarıdır.

3.2.2.1. Sigmoid

Sigmoid fonksiyonu, (0,1) aralığında değer üretmektedir. YSA'da iyi bir sınıflandırıcı olarak sıklıkla tercih edilir. Sigmoid fonksiyonunda girdi değerleri Denklem 3.3'de verilen işlem sonrasında 0 ve 1 aralığında ifade edilir.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3.3}$$

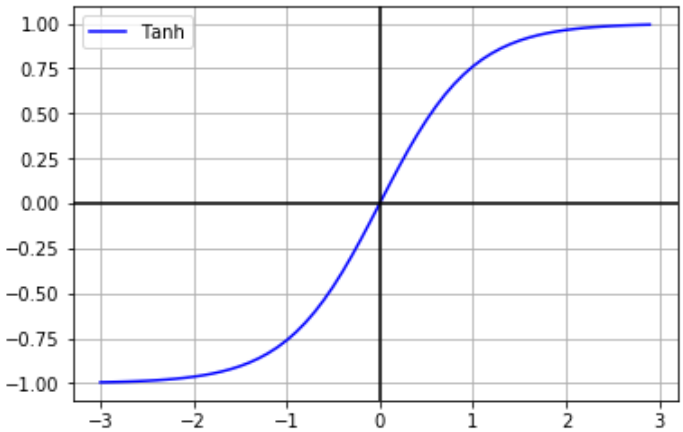


Şekil 3. 4. Sigmoid fonksiyonun grafiği

3.2.2.2.Tanh

Sigmoid fonksiyonuna çok benzer bir yapıya sahiptir. Fonksiyon aralığı $(-1, +1)$ olarak tanımlanır. Sigmoid fonksiyonuna göre avantajı daha çok değer alabilmesidir. Bu, daha hızlı öğrenme ve sınıflama işlemi anlamına gelmektedir.

$$tanh(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})} \tag{3.4}$$

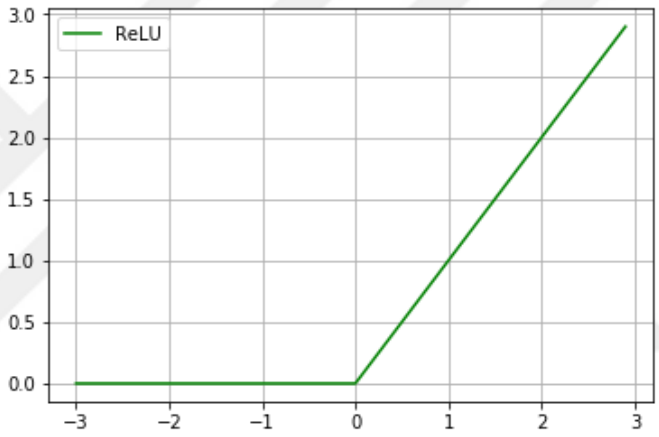


Şekil 3. 5. Tanh fonksiyonun grafiği

3.2.2.3.ReLU

ReLu fonksiyonu, aktivasyon işlemini diğer fonksiyonlara göre daha hızlı bir şekilde gerçekleştirmektedir. ReLU $[0, +\infty)$ aralığında değer alır. Fonksiyon negatif değerli girdileri 0, pozitif değerli girdilerin kendi değerlerini değiştirmeden Denklem 3.5'te tanımlandığı gibi eşikleme işlemini yapar.

$$f(x) = \max(0, x) \tag{3.5}$$



Şekil 3. 6. ReLu fonksiyonun grafiği

3.2.3.İleri Besleme Ağları

Sinir biliminden esinlenerek oluşturulan bu modellere ileri besleme denmesinin sebebi bilgi akışının yönünden kaynaklanmaktadır. Ağ içinde bulunan nöronlara düğüm denir. Aynı katmanda bulunan düğümler birbirlerine bağlı değildir. Düğümler bir sonraki katmanda bulunan düğümlere ağırlık değerini içeren bağlantılar ile bağlıdır.

Ağın ilk katmanı giriş katmanı ve son katmanı çıkış katmanı olarak adlandırılır. Bu iki katman arasında kalan katmanlara gizli katman denir. Gizli katmanın boyutları

modelin genişliğini belirler. Giriş katmanında bulunun düğüm sayısı kadar bilgi hesaplanan fonksiyon üzerinden akar, ara hesaplamalar ile çıktı katmanına ulaşır.

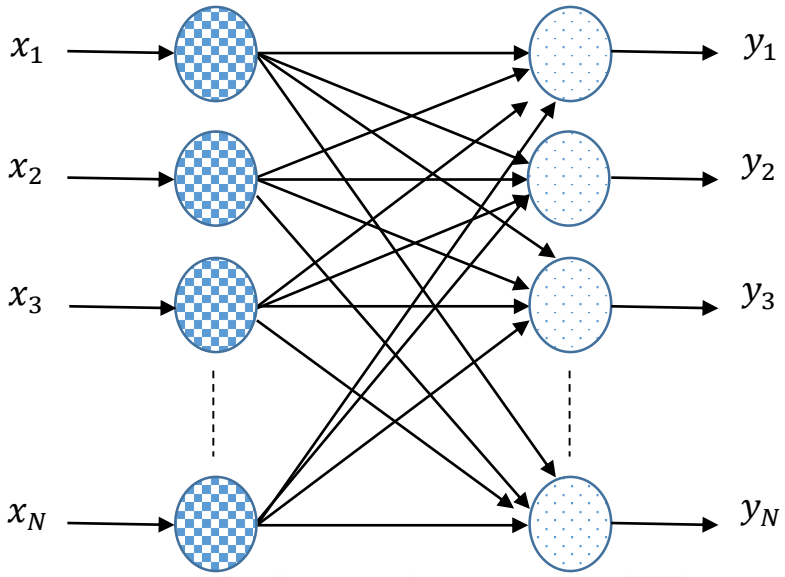
Birçok fonksiyonun bir arada kullanılmasıyla oluşturulan ileri besleme ağlarında fonksiyonlar arasındaki ilişki yönlü düz graflarla (Directed acyclic graph) belirtilen bir çizelge yardımı ile belirlenir. Örneğin; oluşturulan ağ modelinde $f^{(1)}, f^{(2)}, f^{(3)}$ fonksiyonları ard arda sıralanmış bir şeklinde birbirlerine bağlı olabilir ve üç fonksiyon arasındaki ilişki ile bir araya gelip $f(x) = f^{(3)}(f^{(2)}\left(f^{(1)}(x)\right))$ fonksiyonunu oluşturabilir. Bu durumda her bir fonksiyon ağın bir katmanını temsil eder ve $f^{(1)}$ ağın birinci katmanı, $f^{(2)}$ ağın ikinci katmanı, $f^{(3)}$ ağın üçüncü katmanı olarak adlandırılır. Bu tip zincir yapıların tamamının uzunluğu modelin derinliğini verir. ”Derin Öğrenme” adı bu terminolojiden doğmaktadır. (Goodfellow, Bengio, & Courville, 2016)

İleri besleme ağları görüntülerden nesne tanımada, sinyal işlemede ve doğal dil işlemede yaygın olarak kullanılmaktadır.

3.2.3.1.Tek Katmanlı Algılayıcı

Tek bir algılayıcı önemli düzeyde hesaplamalar yapabilmektedir. Sinir ağlarının gerçek hesaplama gücü, birden fazla algılayıcının birbirine bağlanarak bir ağ oluşturduklarında görülmektedir. Algılayıcılar ile bir ağ oluşturmanın en yaygın yapısı katmanlar oluşturularak birbirlerine bağlanmasıdır.

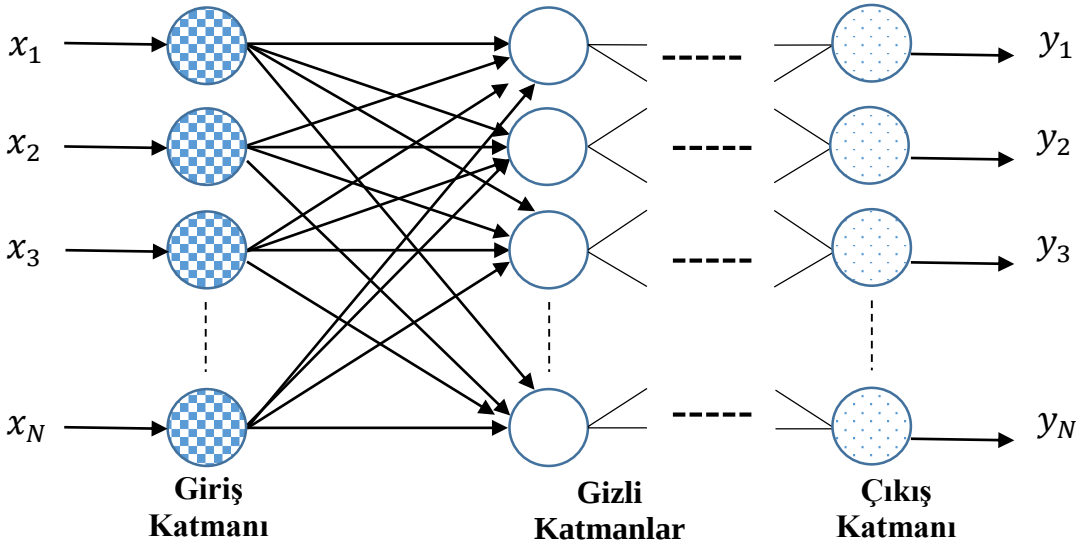
Şekil 3.7’de gösterimi yapılan sinir ağında soldaki düğümler, giriş katmanıdır. Giriş katmanında bulunana nöronlar yalnızca girdileri dağıtır ve hesaplama yapmazlar. Tek gerçek nöron katmanı sağdakidir. Girişlerin her biri, çıkış katmanındaki her yapay nörona bağlantı ağırlığı üzerinden bağlanır. Her çıkış, bağlantı ağırlıklarına göre değişir. (Kawaguchi, 2000) Tek katmalı algılayıcılar düşük düzeyde hesaplamalarda kullanılırlar.



Şekil 3. 7. Tek katmanlı algılayıcı sinir ağı gösterimi

3.2.3.2.Çok Katmanlı Algılayıcılar:

Doğrusal olmayan ve daha yüksek düzeyde hesaplamalar yapabilmek için daha karmaşık bir yapay sinir ağı yapısı gerekmektedir. Tek katmanlı ağlardan farklı olarak bir veya daha fazla gizli katmana sahip olan yapay sinir ağlarına Çok Katmanlı Algılayıcılar denmektedir. İlk katmanda girdiler genellikle işlenmeden gizli katmanlara aktarılır. Gizli katmanlar kendinden önceki katmanın ürettiği çıktıları işler. (Kawaguchi, 2000) Şekil 3.8'de çok katmanlı algılayıcı sinir ağı modelinin gösterimi verilmektedir.

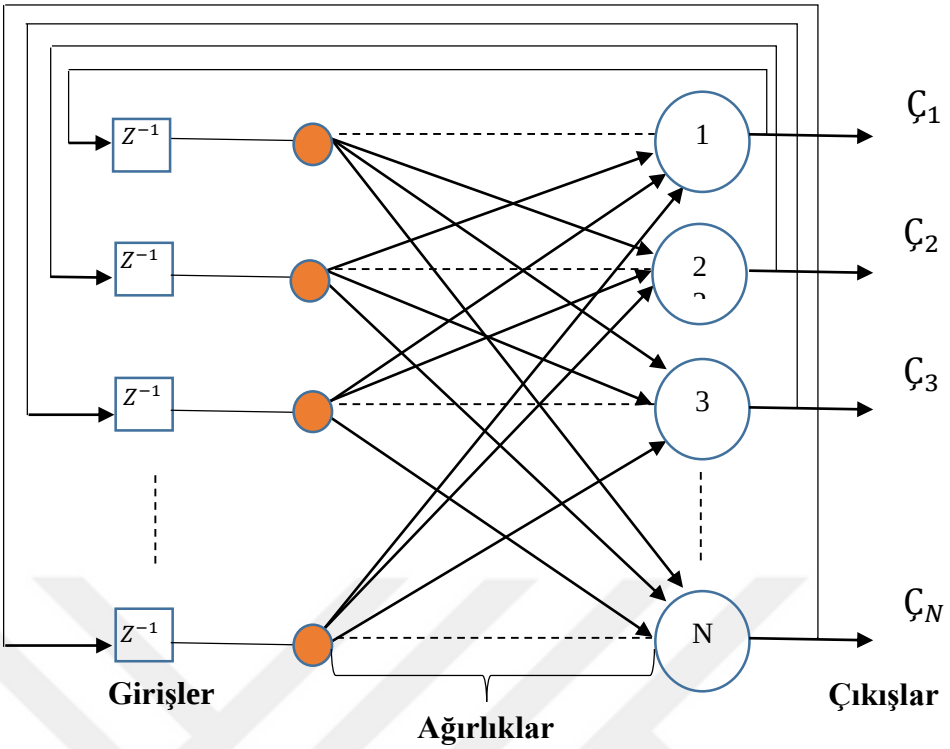


Şekil 3. 8. Çok katmanlı algılayıcı sinir ağı gösterimi

3.2.3.3.Geri Beslemeli Ağlar

Geri beslemeli yapay sinir ağlarında katmanlar arasındaki nöronlar, ileri beslemeli yapay sinir ağları katmanları arasındaki nöronlar gibi çıktılarını sadece bir sonraki katmanda bulunan nöronlara girdi olarak vermezler. Geri beslemeli sinir ağı, ara katmanlarda ve çıkışında elde ettiği çıktıları giriş birimlerine ya da kendinden önceki ara katmanlarda bulunan nöronlara geri besleme yapabilen ağ yapısına sahiptir.

Geri beslemeli ağ yapısında bir nöron bulunduğu katmandaki başka bir nörona ya da bulunduğu katmandan önceki bir katmana girdi olarak bağlanabilmektedir. Ağın bu özelliği sayesinde girişler, ileri ve geri olarak iki yönde de aktarılabilir. Şekil 3.9'da gösterildiği gibi hem ileri hem de geri yönde birbirlerine bağlanabilen nöronlar sayesinde oluşan ağ, doğrusal olmayan bir davranış göstermektedir.



Şekil 3. 9. Geri beslemeli yapay sinir ağları

3.3.Derin Öğrenme(Deep Learning)

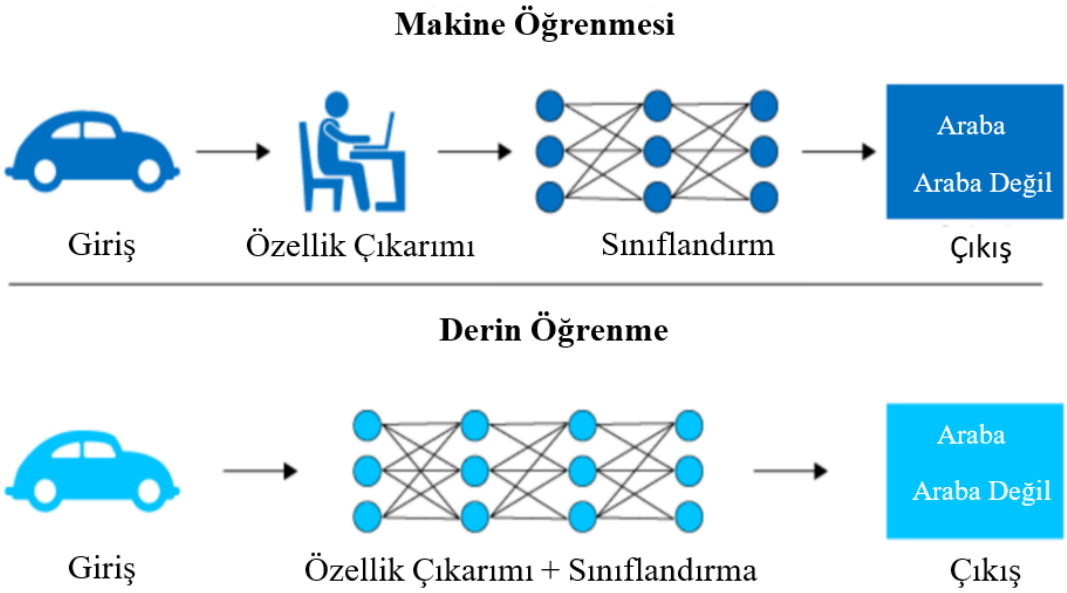
Tarih boyunca mucitlerin en büyük hayallerinden birisi düşünebilen makineler yapabilmekti. Günümüzde bilgisayar teknolojilerinin gelmiş olduğu nokta, mucitlerin hayalini kurduğu düşünebilen makinelerin gerçekleştirilebilir olmasına imkan tanımaktadır. Hayal olmaktan çıkan bu düşünce, günümüzde üzerine birçok uygulama geliştirilen aynı zamanda aktif olarak araştırılan, literatürde "Yapay Zeka" olarak tanımlanan bir bilim dalı haline gelmiştir. Bu gelişimler sayesinde otomasyon sistemlerinden görüntü tanımaya, konuşma anlamadan doğal dil işlemeye hatta tıbbı tanılar yapmaya kadar birçok alanda akıllı yazılımlardan yararlanılmaktadır. (Clark, 2015)

Yapay zekanın ilk dönemlerinde, insanlar için çözümü zor olan ama biçimsel veya matematiksel kurallarla tanımlanabilen problemler kolayca çözebilmekteydi. Ancak insanların günlük hayatında sürekli karşılaştıkları yüz tanıma, ses tanıma gibi biçimsel ya

da matematiksel olarak ifade edilemeyen daha sezgisel problemlerin çözümünde yapay zekanın sorunlar yaşadığı ortaya çıktı.

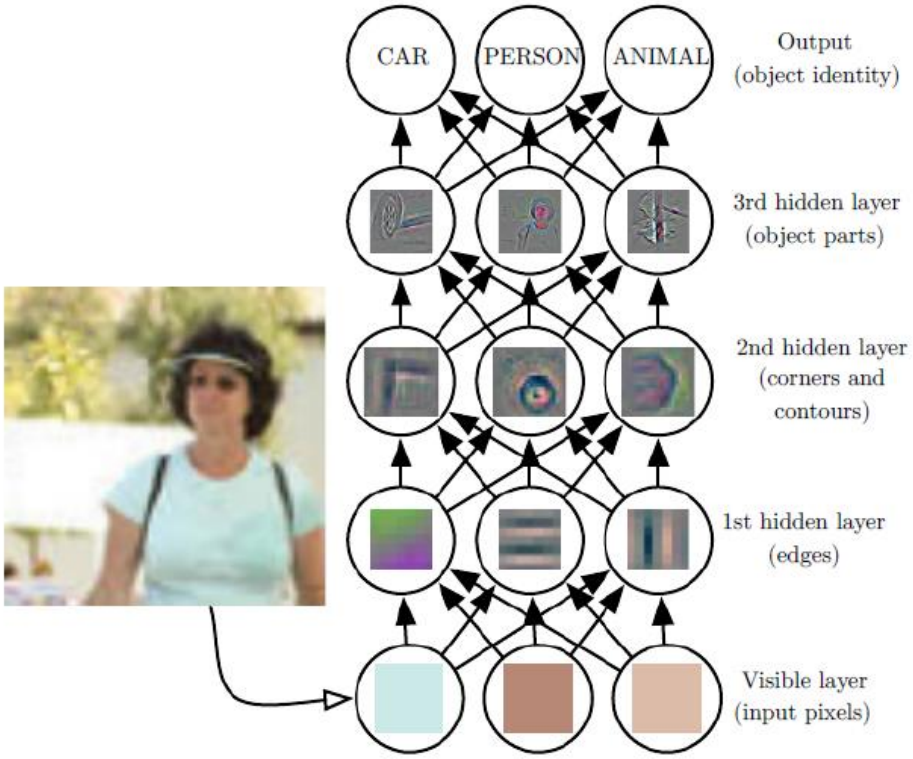
Bu tip sezgisel problemlerin çözümü için birçok yaklaşım üzerinde çalışmalar yapıldı. Bizim de tezimizde kullandığımız "Derin Öğrenme" yaklaşımı, sezgisel ve matematiksel olarak ifade edilmesi zor olan problemlerin çözümünde en çok kullanılan makine öğrenmesi algoritmasıdır.

Derin öğrenme, bilgisayarların gereksinim duyduğu biçimsel veya matematiksel kuralların insan eliyle girilmesine ihtiyaç kalmadan her kavramı daha temel kavramlarla ilişkilendirir. İlişkilendirme sonrasında oluşturduğu kavram hiyerarşisi öğrenmesini sağlar. Bilgisayar kavram hiyerarşisi yardımı ile daha basit kavramlardan yola çıkarak temel karmaşık kavramları oluşturur. (Goodfellow, Bengio, & Courville, 2016) Oluşturulan kavramlar hiyerarşisi, katmanların birbirleri üzerine gelerek oluşmasından dolayı çok katmanlı bir yapı haline gelir ve bir derinlik kazanır. Bu derinlikten dolayı "Derin Öğrenme" denmektedir.



Şekil 3. 10. Derin Öğrenme ve Makine Öğrenmesi öğrenim şeması

Derin öğrenme, basit kavramlardan daha karmaşık kavramlar kurulmasına olanak sağlamaktadır. Şekil 3.11'de görüldüğü gibi Derin Öğrenme bir görüntünün tanımındaki karmaşayı en basit piksel girdilerini kullanarak iç içe geçmiş daha basit eşleşmelere ayırarak çözer.



Şekil 3. 11. Derin öğrenme modeli gösterimi

3.3.1.Makine Öğrenmesinin Zorlukları

Makine öğrenmesi algoritmaları çok çeşitli ve önemli problemlerde iyi şekilde çalışmaktadır. Ancak bu algoritmalar konuşma tanıma ve nesne tanıma gibi birçok merkezi yapay zeka problemlerinde başarılı olamamıştır.

Klasik makine öğrenmesi algoritmaları, yüksek boyutlu uzaylarda karmaşık fonksiyonları öğrenmede yetersiz kalmaktadır. Bu ve benzeri problemleri ortadan kaldırmak için derin öğrenme tasarlanmıştır.

Şekil 3. 12. Verideki ilgili boyutların sayısı arttıkça(soldan sağa), ilgilendiğimiz düzlemlerin sayısı da katlanarak artar.(Boyut laneti)

Makine öğrenmesinde önceden gözlemlenmemiş girdilerin iyi çalışma yeteneğine genelleştirme denir. Makine öğrenmesinin en temel problemlerinden birisi, sadece eğitim setinde değil yeni girdilerde de iyi performans verecek bir algoritmanın geliştirilememesidir. Makine öğrenmesinde kullanılan birçok strateji, test hatasını doğrudan azaltmayı amaçlar ancak bunu yaparken genelde eğitim hatası yükselir. Eğitim hatasını değiştirmeden test hatalarını düşürmek için yapılan değişikliklere düzenlileştirme denir. Makine öğrenmesinin aksine derin öğrenme uygulayıcılarının kullanımına açık birçok düzenlileştirme formu mevcuttur. (Goodfellow, Bengio, & Courville, 2016)

Makine öğrenmesinde algoritmaların doğru çalışması, ham verilerden özelliklerin ve veri temsillerinin doğru bir şekilde oluşturulmasına bağlıdır. (Najafabadi, ve diğerleri, 2014) Algoritma, ham verilerin özelliklerini öğrenim sürecini otomatik bir şekilde gerçekleştirmiş olursa problemlerin çözümü daha kolay hale gelebilir. Derin Öğrenme, Şekil 3.10'da gösterildiği gibi öğrenme sürecinde ham veri üzerinde otomatik öğrenme sağlayarak bu zorluğu aşmıştır.

Derin öğrenmede algoritmaya girdi olarak verilen öğelerin belirli kriterler ile tanımlanmasına ihtiyaç yoktur. Çünkü algoritma büyük miktarda veriyi işleyerek detayları dışarıdan müdahale olmaksızın tanımlayabilir. Bu sebepten derin öğrenme ağlarının eğitilmesi için büyük veri kümelerine ihtiyaç duyulmaktadır. Böylelikle verileri

tanımlayan özelliklerinin ve kriterlerinin çıkarımı ile öğrenme yerine, sisteme sağlanan büyük veri kümeleri ile milyonlarca veri üzerindeki bilgilere ulaşılarak sorunsuz bir şekilde öğrenme sağlanır.

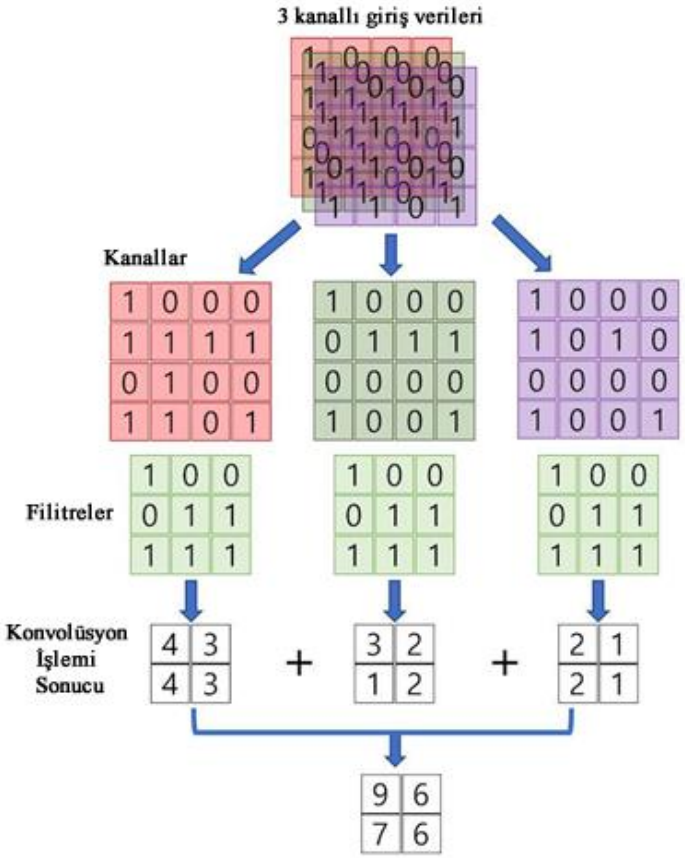
3.3.2.Konvolüsyonel Sinir Ağları(CNN)

Evrişimsel Sinir Ağı veya Konvolüsyonel Sinir Ağları (CNN), günümüzde doğal dil işleme ve bilgisayar görmesi problemlerinin çözümünde sıkça kullanılan bir Derin Öğrenme modelidir. Izgara benzeri bir yapıya sahip veriyi işlemek için kullanılan özel bir sinir ağıdır. (LeCun, Bottou, Bengio, & Haffner, 1998)

Konvolüsyon matematiksel olarak iki sinyalin birleştirilip, bir üçüncü sinyal üretilmesidir. Matematiksel olarak gösterimi Denklem 3.6'da gösterildiği gibidir.

$$[f * g](t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau \tag{3.6}$$

Sinir ağlarında konvolüsyon ise bir filtreleme operasyonu olarak tanımlanabilir. Amaç girdi olarak verilen ham verilerden (metin, görüntü vb.) özelliklerin ya da bilgilerin çıkarılmasıdır. Her bir veri değerler matrisi olarak düşünülmektedir. Konvolüsyonel sinir ağlarında bir filtre, tartılardan oluşan bir vektör ile gösterilir. Filtre bir girdi parçasının bir özelliğe ne kadar benzediğinin ölçülmesini sağlar.

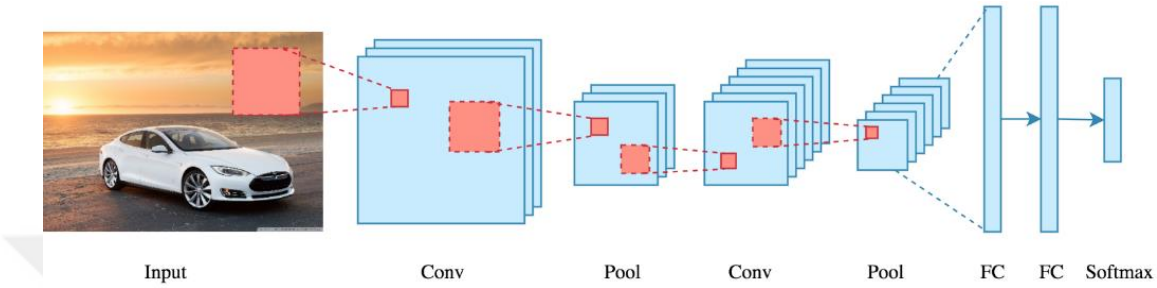


Şekil 3. 13. Örnek bir konvolüsyon işlemi

Konvolüsyonel Sinir Ağları, normal sinir ağlarına çok benzer. Bunlar ayrıca öğrenilebilir ağırlıkları ve önyargıları olan nöronlardan oluşur. En önemli fark katman sayısıdır. CNN, sonuçlara uygulanan doğrusal olmayan aktivasyon işlevlerine sahip sadece birkaç katlanma katmanıdır. Geleneksel bir yapay sinir ağında her giriş nöronu, bir sonraki katmandaki her çıkış nöronuna bağlanır. Buna tamamen bağlı bir katman denir. CNN'lerde bunun yerine, çıktıyı hesaplamak için girdi katmanı üzerinde konvolüsyonlar kullanılır. Bu, girişin her bölgesinin çıkıştaki bir nörona bağlandığı yerel bağlantılarla sonuçlanır. Her katman, genellikle yüzlerce veya binlerce farklı filtreler uygular ve sonuçlarını birleştirir. (Lopez & Kalita, 2017)

Konvolüsyonel Sinir Ağlarının ana özelliği, tipik olarak konvolüsyon tabakalarından sonra uygulanan havuzlama tabakalarının kullanılmasıdır. Havuz

katmanları girdilerini örneklemektedir. Havuzlamanın bir özelliği, genellikle sınıflandırma için gerekli olan sabit boyutlu bir çıktı matrisi sağlamasıdır. Havuzlama aynı zamanda en belirgin bilgileri korurken çıktı boyutluluğunu da azaltır. Her filtrenin belirli bir özellik algıladığını düşünebiliriz.



Şekil 3. 14. Basit bir CNN Mimarisi

Eğitim aşamasında, bir CNN gerçekleştirilecek göreve dayalı olarak filtrelerinin değerlerini otomatik olarak öğrenir. Örneğin, Şekil 3.14'de gösterilen resmin sınıflandırmasında bir CNN, birinci katmandaki ham piksellerden kenarları algılamayı öğrenebilir, ardından ikinci katmandaki basit şekilleri algılamak için kenarları kullanabilir ve daha sonra araba şekilleri gibi daha üst düzey özellikleri algılamak için bu şekilleri kullanabilir. (LeCun, Kavukcuoglu, & Farabet, 2010)

3.3.3.Tekrarlayan Sinir Ağları(RNN)

Tekrarlayan Sinir Ağları (Recurrent Neural Network) sıralı verileri işlemek için özelleştirilmiş sinir ağları ailesindendir. (Rumelhart, Hinton, & Williams , 1986) RNN'ler, insan belleğinin bir özelliği olan geçmiş ve gelecek bilgi arasında ilişki kurabilme özelliğini modelleye bilen bir yapıya sahiptir. Geleneksel sinir ağlarında, bütün girdiler ve çıktılar birbirinden bağımsızdır. Örneğin; bir cümlenin bir sonraki kelimesini tahmin etmek istediğimizde önceki kelimeleri hatırlama zorunludur.

Bilginin katmandan katmana bir yönde aktığı İleri Beslemeli sinir ağlarının aksine Tekrarlayan Sinir Ağlarında bilgi, durumun önceki durumlarından etkilenebilmesi için katmandan katmana döngüler halinde ilerler. (Britz, 2015) Ayrıca RNN'ler modelin geçmiş hesaplamaları hakkında bilgi depolamasına izin veren bir belleğe sahiptir. Bu, tekrarlayan sinir ağlarının dinamik geçici davranış sergilemesini sağlar.

Tekrarlayan sinir ağları, girdi-çıktı çiftlerinin zamansal dizilerini modelleye bildiklerinden doğal dil işleme (NLP) uygulamalarında büyük bir başarı yakalamışlardır.

En basit ifadeyle, aşağıdaki Denklem 3.7a ve 3.7b bir RNN'nin zaman içinde nasıl geliştiğini tanımlar:

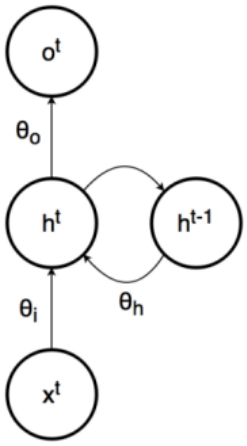
$$o^t = f(h^t; \theta) \tag{3.7a}$$

$$h^t = g(h^{t-1}, x^t; \theta) \tag{3.7b}$$

Formül 3.7a ve 3.7b'de o^t, t zamanında RNN'nin çıktısı olduğunda, x^t, t zamanında h^t, RNN'ye girdidir ve t zamanında gizli katmanın veya katmanların durumudur.

Birinci denklem, θ parametreleri (ağ için ağırlıkları ve önyargıları kapsıyor) verilen verilere göre, t zamanındaki çıktının, yalnızca t beslemesindeki bir sinir ağı gibi, sadece t zamanındaki gizli katmanın durumuna bağlı olduğunu söyler.

İkinci denklem, aynı parametreler göz önüne alındığında, t zamanındaki gizli katmanın $t-1$ zamanındaki gizli katmana ve t zamanındaki girişe bağlı olduğunu söylüyor. Bu ikinci denklem, RNN'nin geçmiş hesaplamaları h^{t-1}'in mevcut hesaplamaları h^t'yi etkilemesine izin vererek geçmişini hatırlayabildiğini göstermektedir. Şekil 3.15'de, bir RNN'nin hesaplama grafiğinde bu üç değişken arasındaki ilişkiyi göstermek için basit bir grafik modeli göstermektedir.



Şekil 3. 15. RNN modeli basit gösterimi (McGonagle, Williams, & Khim, 2019)

3.3.4.Uzun Kısa Sürekli Bellek Ağları(LSTM)

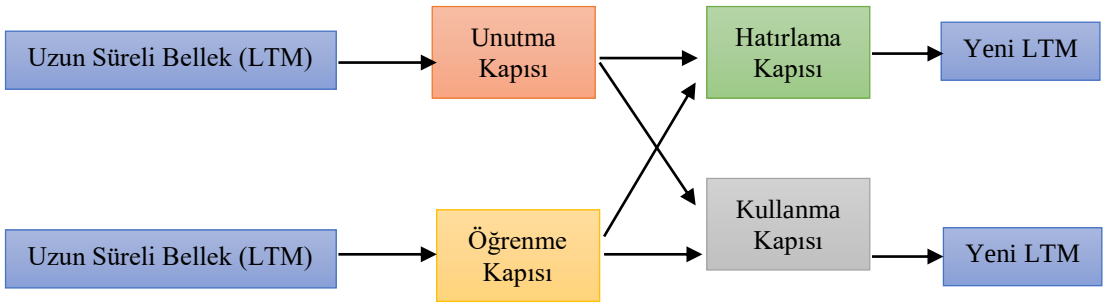
Uzun Kısa Süreli Bellek ağları kısacası LSTM'ler, tekrarlayan sinir ağlarında yaşanan uzun süreli bağımlılık probleminin önüne geçebilmek amacıyla geliştirilmiş bir RNN türüdür. (Hochreiter & Schmidhuber, 1997) RNN'ler kısa süreli belleklerin modellenmesinde çok iyi bir şekilde çalışmaktadır. Ancak uzun süreli bağımlılıklarda bu modeller etkili olamamaktadır. Bunun nedeni belirli bir zaman adımından sonra modelin gradyan hesaplamalarının sonsuza gitmesi ya da sıfırlanmasıdır. (Karakuş, Talo, Hallaç, & Aydin, 2018)

RNN'ler, gerçek değerli girdi sıralarını gerçek değerli çıktı sıralarına eşlemek için dağıtılmış dahili hafızaları kullanır. Bununla birlikte RNN'ler potansiyel olarak gürültüye dirençli algoritmaların çok genel ve sürekli bir alanında gradyan inişini problemi yaşmaktadır. Gürültü öğrenmeyi zorlaştıran bir etkendir. (Maass & Orponen, 1998) Uzun kısa süreli hafıza (LSTM) yöntemi bu problemden etkilenmemektedir. (Hochreiter & Schmidhuber, 1997)

RNN modelleri ile oluşturulan ağların önemli bir eksiği sadece var olan girdiler üzerinden işlem yapmasıdır. Derin öğrenmede uzun metinlerde cümlenin içeriğinin öğrenilebilmesi için uzun süreli bağımlıklara gerek duyulmaktadır. Örneğin "Umut Türkçe konuşuyor" cümlesinde uzun süreli bağımlılığa ihtiyaç duymadan RNN'ler cümlenin son kelimesini basitçe tahmin edebilir. Fakat daha uzun içeriğe sahip bir

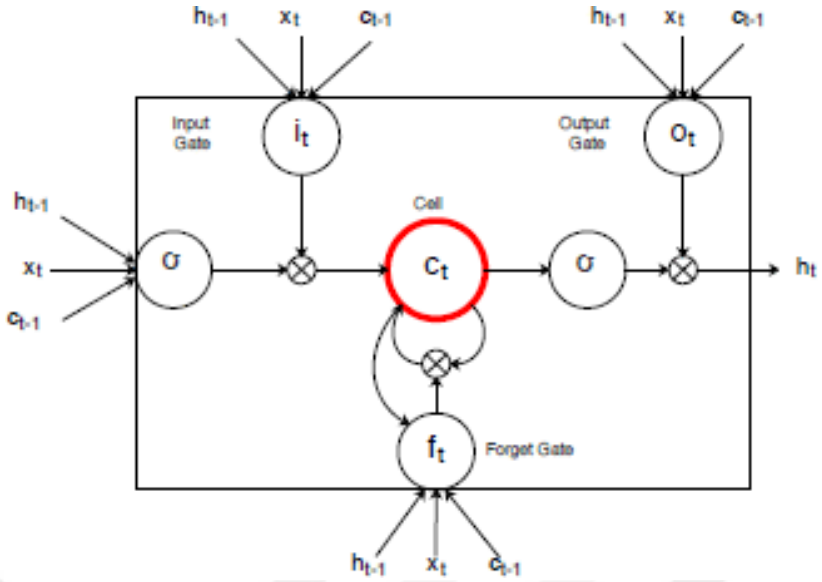
cümlede örneğin “Umut Almanya’nın Berlin şehrinde doğdu, ilkokulu İspanya’da, liseyi İngiltere’de ve üniversiteyi Türkiye’de okudu. Umut’un bildiği diller …” gibi cümlelerde RNN kullanılarak oluşturulan ağlar, cümlenin son kelimesini tahmin etmek için önceki kelimeleri hatırlamakta zorlanacaktır. Bu durum RNN’ler için uzun süreli bağımlık problemine neden olmaktadır. Problemin giderilmesi için uzun bir zaman aralığında verilen girdileri geriye dönük olarak hatırlayabilen ve girdilerin önemine göre hangi girdinin ne kadarını hatırlaması gerektiğine karar verebilen modellere ihtiyaç duyulmaktadır. (Karakuş, Talo, Hallaç, & Aydin, 2018) LSTM ağları uzun süreli bağımlılıklarda bu ihtiyaca cevap vermektedir. (Gers & Schmidhuber, 2001)

LSTM mimarisi uzun süreli bağımlılıklara ihtiyaç duyulan durumlarda hafıza hücreleri (memory cell) olarak adlandırılan gizli birimleri kullanmaktadır. Geniş bir zaman aralığında hatırlama ihtiyacı duyduğu girdileri bu hafıza birimlerinde saklar. Kaydettiği bu bilgilerin önemli olup olmadığına mimarisinde barındırdığı kapılar sayesinde karar verir. Şekil 3.16’da gösterildiği gibi LSTM mimarisi 4 farklı birimden oluşmaktadır. Bunlar unutma(forget), öğrenme(lear), hatırlama(remember) ve kullanma(use) kapılarıdır. (Udacity Deep Learning Team, 2019)



Şekil 3. 16. LSTM mimarisi

Standart bir RNN’deki yinelenen her parça, tanh katmanı gibi basit bir katman içerir. Bununla birlikte, LSTM farklı bir yinelenen modül yapısına sahiptir ve bağlam bilgisini öğrenmek için hücre durumunu kullanır. Ayrıca, hücre durumu önceki bağlam bilgisinin nasıl değiştirileceğini belirlemek için üç kapıya sahiptir. (Yao & Huang, 2016) Şekil 3.17’de LSTM hücresinin durumunun yapısı gösterilmektedir.



Şekil 3. 17. LSTM hücresi durumunun yapısı (Wang , Xia, Liu, Li, & Li, 2017)

İlk olarak, unutma kapısı, hangi bağlam bilgisinin hücre durumundan çıkarılması gerektiğini belirlemektedir (Cui, Ke, & Wang, 2017). Unutma kapısının mekanizmasını uygulamak için Denklem 3.8 kullanıyoruz:

$$f_t = \sigma\left(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f\right) \tag{3.8}$$

$\sigma(\cdot)$, lojistik sigmoid fonksiyonudur. W_f ve b_f , ağırlık matrisini ve unut kapısının önyargı vektörünü, x_t ise geçerli giriş vektörünü gösterir, h_{t-1} önceki çıkış vektörünü gösterir ve C_{t-1} önceki hücre durumu gösterir.

Bir sonraki adım, yeni bağlam bilgisinin hücre durumunda saklanıp saklanmayacağına karar vermek için giriş kapısını kullanmaktır. Hangi bilgilerin güncellenmesi gerektiğine karar vermek için bir sigmoid katman ve yeni aday bilgilerinin bir vektörünü oluşturmak için bir $tanh$ katman kullanılır. İki fonksiyon Denklem 3.9 ve 3.10 aşağıdaki gibidir:

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i) \quad (3.9)$$

$$\overline{C_i} = tanh(W_c \cdot [C_{t-1}, h_{t-1}, x_t] + b_c), \quad (3.10)$$

W_i ve W_c sırasıyla giriş kapısı ve aday hücrenin ağırlık matrisini belirtirken, b_i ve b_c sırasıyla giriş kapısı ve aday hücrenin bias vektörünü belirtir ve C_t yeni aday hücre bilgisidir. Önceki hücre durumu C_{t-1}''i yeni hücre durumu C_t'ye güncellemek için aşağıdaki işlevi Denklem 3.11 kullanılır:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t, \quad (3.11)$$

Son olarak, çıkış kapısı, çıkış bilgisini belirlemek için aşağıdaki fonksiyonu 3.12 ve 3.13 denklemlerini kullanır:

$$o_t = \sigma(W_o \cdot [C_{t-1}, h_{t-1}, x_t] + b_o), \quad (3.12)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (3.13)$$

W_o çıkış geçidinin ağırlık matrisini belirtirken, b_o çıkış geçidinin bias vektörünü belirtir ve h_t çıkış bilgisidir.

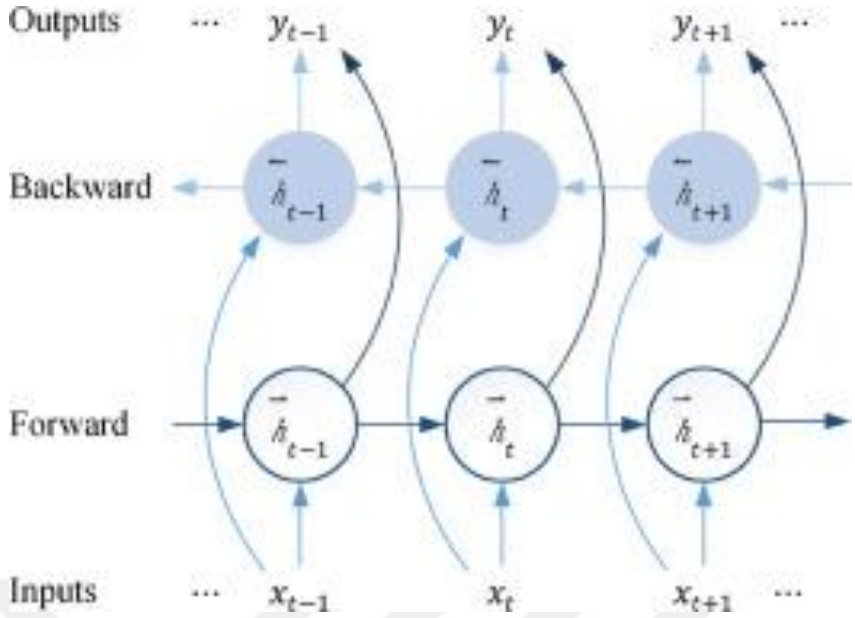
Kısacası LSTM mimarisi dört kapı ve iki bellekten oluşmaktadır. Unutma kapısı önemini yitiren veya önemli görülmeyen girdileri kaydedildikleri uzun süreli belleklerden alır ve girdinin bulunduğu bellek parçasını unutur. Yeni bilginin öğrenilmesi için kullanılan öğrenme kapısı, geçmiş zaman adımlarında kısa süreli bellekten öğrenilen bilgiler ile birlikte yeni girdileri alır ve yeni bilgiyi öğrenir. Hatırlama kapısı, öğrenilen

yeni bilgileri uzun süreli belleklere kaydederek uzun süreli hafızayı günceller. Kullanma kapısı, yeni tahminler yapmak için kısa süreli bellekleri günceller ve öğrenilen bilgiler ile birlikte uzun süreli bellekleri kullanır. LSTM'ler geri yayılım sürecinde sigmoid (σ), hiperbolik tanjant($\tan h$), çarpım(x) ve toplam($+$) fonksiyonlarını kullanarak ağırlıkları güncellemektedir.

LSTM mimarisinin en önemli özellikleri, bilgi kaybını engellemek amacıyla girdilerin unutulmadan saklanabilmesi ve sıfırlanan gradyan problemine çözüm olmasıdır. LSTM biriminde çıktısı 0 ve 1 arasında değer alan üç sigmoid fonksiyon kullanılmaktadır. Sigmoid fonksiyonları bir x girdisi için $\sigma(x) \approx 1$ sonucunu ürettiğinde tüm x girdisinin geçmesi, $\sigma(x) \approx 0$ sonucunda ise x girdisinin geçmemesi anlamına gelmektedir. Bu sayede LSTM, sigmoid fonksiyonlarını kullanarak hangi girdinin hücreden geçeceğine, hangi girdinin hücrede kalacağına ve hangisinin çıktıya ulaşacağına karar verebilmektedir.

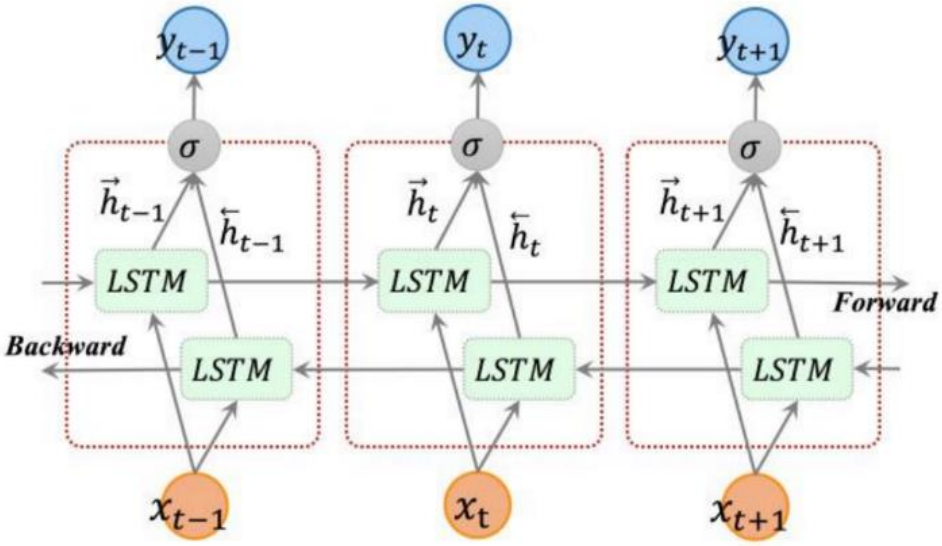
3.3.5.Uzun Kısa Süreli Belleğe Sahip Çift Yönlü Tekrarlayan Sinir Ağları (BiLSTM)

BiLSTM, çift yönlü RNN'nin LSTM mimarisi ile modifiye edilmiş halidir. Çift yönlü RNN, sırasıyla giriş sinyallerini ileri ve geri işleyen iki bağımsız RNN katmanından oluşur. Şekil 3.18'de görüldüğü gibi bu iki tabakanın çıktılarının her ikisi de nihai çıktıları vermek için kullanılır.



Şekil 3. 18. Çift yönlü RNN'in kontrolsüz yapısı.

Basit bir yapay sinir ağı yapısına sahip olan RNN'ler de genellikle kaybolan gradyan probleminin neden olduğu düşük performanslar görülmektedir. (Bengio, Simard, & Paolo , 1994) Bu problemin giderilmesi için özel bir RNN hücresi olan LSTM önerilmektedir. (Hochreiter & Schmidhuber, 1997) Tek yönlü bir ileri LSTM'deki gizli durum yalnızca önceki özellikleri yakalar ve geleceği dikkate almaz. Çift Yönlü RNN'in LSTM ile birleştirilmesi, önceki ve gelecekteki özellikleri etkili bir şekilde kullanılmasını sağlar. (Gravesa, 2005, Yao, 2016) LSTM ağından farklı olarak, BiLSTM ağının her iki yayılım yönünde de iki paralel katmanı vardır. Şekil 3.19'da gösterildiği gibi önceki özellikler ileri bir LSTM katmanı ile çıkarılır ve gelecekteki özellikler geriye doğru bir LSTM katmanı tarafından yakalanır.



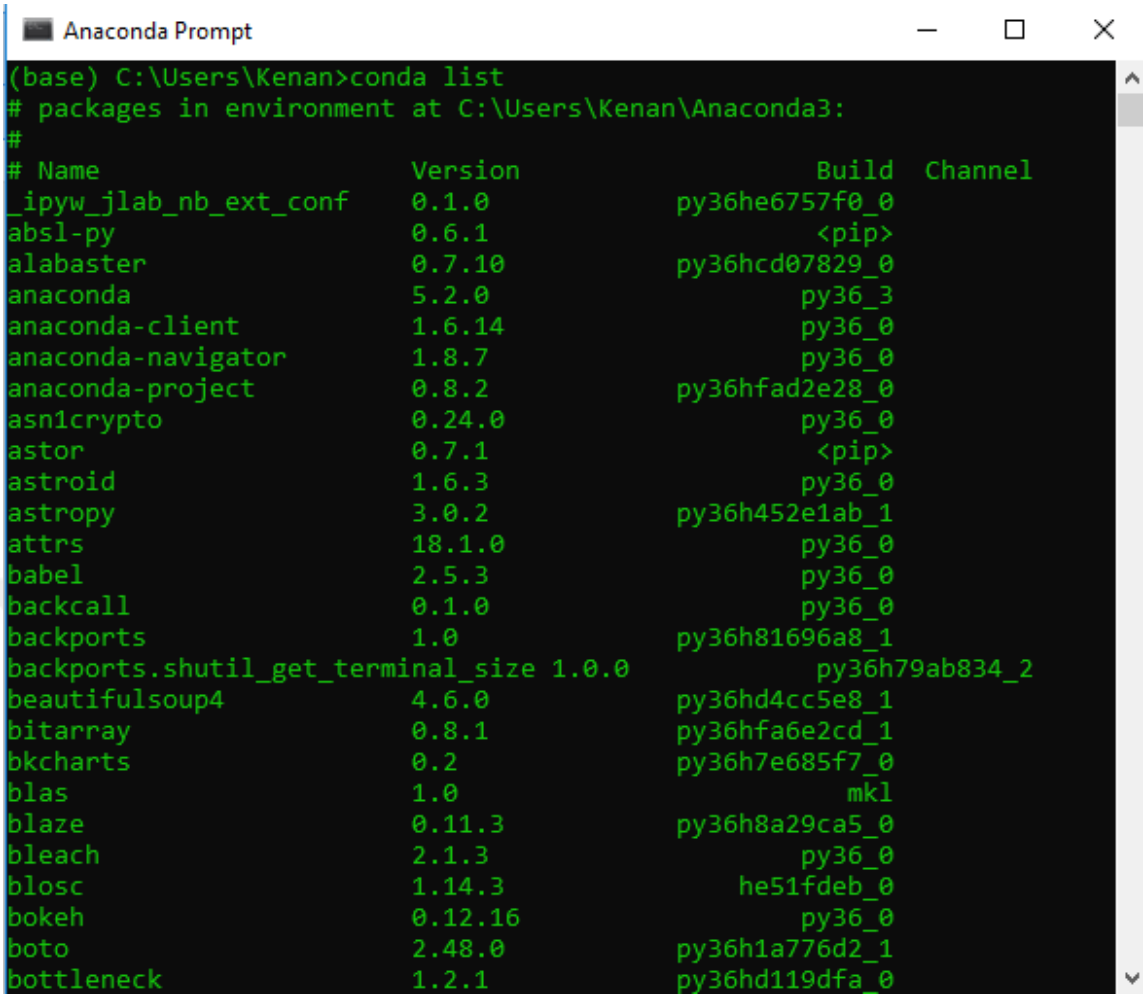
Şekil 3. 19. Arka arkaya 3 adımda katlanmış BiLSTM mimarisi (Cui, Ke, & Wang, 2017)

BiLSTM'lerde temel düşünce, sırasıyla geçmiş ve gelecekteki bilgileri yakalamak için ileri ve geri iki ayrı gizli katman sunmaktır. (Gravesa & Schmidhuber, 2005) Ardından, iki gizli durum final çıktısını oluşturmak için birleştirilir.

3.3.6.Derin Öğrenme Geliştirme Ortamları

3.3.6.1.Anaconda

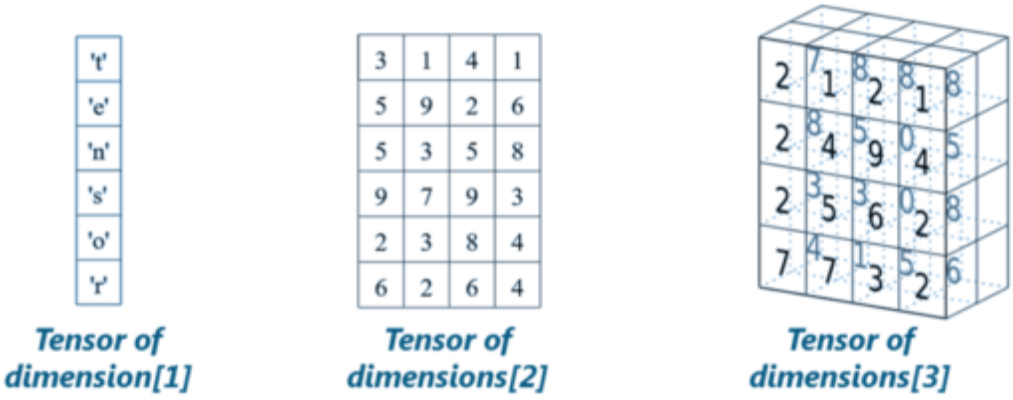
Anaconda, makine öğrenmesi ve derin öğrenmeye olan ilginin son yıllarda artması ile birlikte veri bilimciler ve bilgisayar programcılarının yaygın olarak kullandıkları bir veri bilimi platformudur. Anaconda, Python ve R programlama dillerinin dağıtıcısıdır. Aynı zamanda açık kaynaklı bir paket yöneticisidir. Makine öğrenmesi ve veri bilimi için tasarlanmış olup, genellikle büyük ölçekli veri işleme, bilimsel bilgi işlem ve tahmine dayalı analitik için kullanılır. 1.000'in üzerinde veri bilimi paketi koleksiyonu sunar. (Anaconda, 2019)



Şekil 3. 20. Anaconda Ortamı

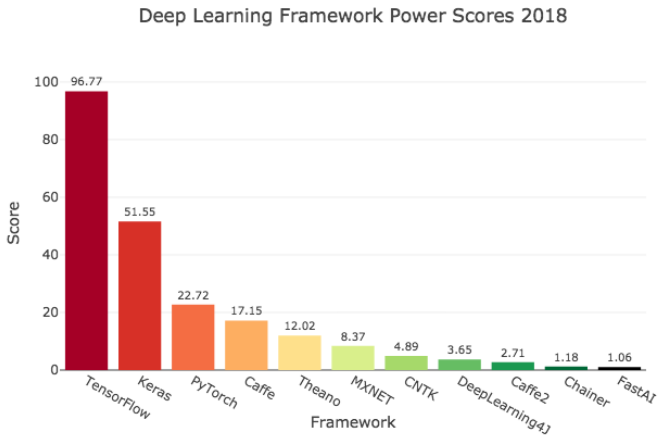
3.3.6.2.TensorFlow

2015 yılında Google tarafından geliştirilen, veri akış grafikleri konseptine dayanan açık kaynak kodlu bir makine öğrenmesi kütüphanesidir. Sistem özelinde verilen verilerin kodlaması ve ayırt edilmelerini sağlamaktadır. Tensorflow, derin öğrenme verilerini temsil etmek için tensor adı verilen nesneler kullanır. Tensorler, yüksek boyutlara sahip verileri temsil etmemize izin veren çok boyutlu veri dizileridir. (TensorFlow, 2019)



Şekil 3. 21. Tensor boyut gösterimi

Platform fark etmeksizin, bir veya birden fazla CPU, GPU kullanarak çalışabilme özelliğine sahiptir sağlar. Python kullanılarak geliştirilen bu kütüphane, C++, Java, C#, Javascript, R, Python gibi birçok dili desteklemektedir. (TensorFlow, 2019) Kütüphane bu özellikleri ile derin öğrenme ve yapay zeka alanında en çok kullanılan kütüphanelerdendir. (Clark, 2015;Sutskever, Vinyals, & Le, 2014;Szegedy, ve diğerleri, 2015;Vinyals, Fortunato, & Jaitly, 2015;Ramsundar, ve diğerleri, 2015) TensorFlow kullanılarak yapılan bir hesapla, telefonlar ve tabletler gibi mobil cihazlarda dahil olmak üzere çok çeşitli heterojen sistemlerde çalışabilmektedir. (Abadi, Agarwal, ve diğerleri, 2016)



Şekil 3. 22. Derin öğrenme için framework kullanım oranları (2018 yılı verileri)

3.3.6.3.Keras

Keras, TensorFlow, CNTK, Theano ve MXNet üstünde çalıştırabilen açık kaynaklı bir üst düzey sinir ağı kütüphanesidir. Standart sinir ağlarına ek olarak, Keras evrişimli ve tekrarlayan sinir ağlarını da destekler. Bırakma, toplu normalleştirme ve havuzlama gibi diğer genel yardımcı katmanları destekler. Keras, basit ve sezgisel bir sözdizimini destekleyen, kullanımı kolay bir kütüphanedir. (Keras, 2019)

Keras, dizili katmanlardan oluşan bir sinir ağı modeli için Sequential sarmanlayıcı sınıfını kullanmaktadır. Sinir ağı oluşturulurken yeni bir katman eklemek için add() komutunu, oluşturulan ağı eğitmek ve test etmek için compile(), fit() ve evaluate() komutlarını kullanır.

```
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Flatten, LSTM, Dropout
from keras.layers.embeddings import Embedding

# random veri seti üretimi
X_egitim = np.random.random((1000, 20))
Y_egitim = np.random.random(2, size=(1000, 1))
X_test = np.random.random((100, 20))
Y_test = np.random.random(2, size=(100, 1))

#yapay sinir ağının oluşturulması
model_lstm = Sequential()
model_lstm.add(Embedding(1000, 64, input_length=100))
model_lstm.add(LSTM(32, dropout=0.2, recurrent_dropout=0.2))
model_lstm.add(Dense(1, activation='sigmoid'))

#modelin derlenmesi
model_lstm.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])

#modelin eğitimi
model_lstm.fit(X_egitim,Y_egitim, epochs=5,
                   validation_data=(X_test, Y_test),shuffle=True)
```

```
#modelin test edilmesi
score= model_conv.evaluate(X_test, Y_test, batch_size = 128)
```

Yukarıda Keras'ta oluşturulan basit bir model örneği verilmiştir. Modelin derlenmesinde kullanılan compile() metodu için;

- optimizer(), sinir ağının parametrelerinin güncellenmesi için kullanılır. Bunu "loss" fonksiyonuna ve veriye bakarak yapar.
- loss(), optimizasyon sırasında modelin ne kadar doğru olduğunu ölçmede kullanılır.
- metrics(), eğitim ve test adımlarında tüm veri seti üzerinden çıkarılan bir metriktir. "Accuracy" örneklerin ne kadar doğru sınıflandırıldığını ölçer.

Model mimarisinin düğümlerini ve katmanlarını model.summary() komutu ile görülebilmektedir. Modelin eğitimi için fit() metoduna epoch sayısı ve test için üretilen eğitim seti belirtilir. Eğitim sonrası değerlendirme için evaluate() metodu kullanılır.

BÖLÜM 4

DENEYLER VE SONUÇLAR

4.1.Veri Seti

Bu tez çalışmasında; Ott ve arkadaşlarının (Ott, Cardie, & Hancock, 2013) olumsuz sahte yorumları veri kümesi üzerinde ortaya koydukları modelin performansını iyileştirmek üzere farklı analizler gerçekleştirdik. Bu nedenle veri kümesi olarak Ott ve arkadaşlarının kullandıkları veri kümesini kullandık. (Ott, Cardie, & Hancock, 2013) Veri setinin hazırlanması ile ilgili bilgi, yorum özelliklerinin başlıkları altında açıklanmıştır.

Ott ve arkadaşları veri kümesini takip edilen şekilde hazırlamıştır. Çizelge 4-1'de görüldüğü gibi veri kümesi toplamda 1600 adet yorum içermektedir. Veri kümesi, olumlu (pozitif) yorumlar ve olumsuz (negatif) yorumlar olmak üzere 2 klasörden oluşmaktadır ve her biri de kendi içinde 400 adet geçek yorum ve 400 adet sahte yorum olmak üzere toplam 800 adet yorum içermektedir. Pozitif yorum kümesi "Gold Standard" veri seti olarak kabul edilmiştir. (Ott, Choi, Cardie, & Hancock, 2011) Fakat negatif veri kümesi yeterli sayıda yorum içinden seçilmediği için "Gold Standard" olarak kabul görmemiştir. (Ott, Cardie, & Hancock , 2012)

Çizelge 4.1. Ott ve arkadaşlarının veri kümesindeki yorum grupları ve adetleri

Pozitif Yorumlar (800)		Negatif Yorumlar (800)	
Gerçek Yorumlar (400)	Sahte Yorumlar (400)	Gerçek Yorumlar (400)	Sahte Yorumlar (400)

Pozitif Gerçek Yorumların Elde Edilmesi

Gerçek yorumlar dünya genelinde kullanılmakta olan TripAdvisor.com sitesinden Şikago bölgesinde yer alan oteller hakkında kullanıcıların yazdığı yorumlar belli kısıtlamalara göre filtrelenerek alınmıştır. (Ott, Choi, Cardie, & Hancock, 2011)

Bu kısıtlamalar aşağıda açıklanmıştır:

- En çok yorum alan ilk 20 otele ait yorumlar alınmıştır.
- 1-5 ölçeğinde 5 yıldızlı olmayan yorumlar çıkarılmıştır.
- 150 kelimeden az olan yorumlar çıkarılmıştır. Bunun sebebi ise sahte yorum kümesinin en az 150 kelime içeriyor olmasıdır.
- İlk yorumunu yapmış olan kullanıcıların yorumları, sahte olma olasılığı yüksek olduğundan dolayı çıkarılmıştır.

Pozitif Sahte Yorumların Elde Edilmesi

Sahte yorumlar kalabalık bir çevrim içi iş gücü servisi sağlayan AMT (Amazon Mechanical Turk) web sitesinden yararlanılarak; toplam 400 sahte pozitif yorum oluşturulması için bir iş açılmıştır. Yorum yapacak olan AMT çalışanlarına, bazı kısıtlamalar ile yorum yapmaları zorlanmıştır. (Ott, Choi, Cardie, & Hancock, 2011)

Bu kısıtlamalar aşağıda listelenmiştir:

- Gerçek yorumların yapıldığı 20 otel için pozitif sahte yorum yapılması istenmiştir.
- 400 yorumun da her biri farklı bir çalışan tarafından yapılması sağlanmıştır.
- Sadece ABD'de yaşayan kullanıcıların yorum yapmaları sağlanmıştır. 30 dk içinde yorumlarını bitirmeleri istenmiştir.

Negatif Gerçek Yorumların Elde Edilmesi

Gerçek negatif (olumsuz) yorumlar popüler 6 web sitesinden (Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor ve Yelp), pozitif yorumların elde edildiği benzer kısıtlamalar kullanılarak elde edilmiştir. (Ott, Cardie, & Hancock, 2013)

Bu kısıtlamalar aşağıda açıklanmıştır:

- En çok yorum alan ilk 20 otele ait yorumlar alınmıştır.
- 1-5 ölçeğinde 1 veya 2 yıldızlı yorumlar alınmıştır.
- Gerçek negatif yorumlarda kelime sayısı sahte yorumlardakine göre fazla olduğu için, yorumlardaki 150 kelimeden sonrası log-normal dağılıma göre çıkarılmıştır.
- İlk yorumunu yapmış olan kullanıcıların yorumları, sahte olma olasılığı yüksek olduğundan dolayı çıkarılmıştır.

Negatif Sahte Yorumların Elde Edilmesi

Negatif (olumsuz) sahte yorumlar ise pozitif sahte yorumlardaki gibi aynı şekilde kalabalık bir çevrim içi iş gücü servisi sağlayan AMT (Amazon Mechanical Turk) web sitesinden yararlanılarak, toplam 400 sahte negatif yorum yapılması için bir iş açılmıştır.

Yorum yapacak olan AMT çalışanlarına bazı kısıtlamalar ile yorum yapmaları zorlanmıştır. (Ott, Cardie, & Hancock, 2013)

Bu kısıtlamalar aşağıda açıklanmıştır:

- Gerçek yorumların yapıldığı 20 otel için negatif yorum yapılması istenmiştir.
- 400 yorumun da her biri farklı bir çalışan tarafından yapılması sağlanmıştır.
- ABD'de yaşayan kullanıcıların yorum yapmaları sağlanmıştır.
- 30 dk içinde yorumlarını bitirmeleri istenmiştir.

4.2.Veri Seti Ön İşlemi

Kullanılan veri seti 1600 yorumdan oluşmaktadır. Bu yorumlar Ott ve arkadaşları tarafından negatif ve pozitif yorumlar olarak işaretlenmiştir. Bu tez çalışması için negatif ya da pozitif yorumların incelenmesinden daha çok her iki durumda da yapılan sahte yorumların tespiti üzerinde durmaktayız. Veri setinde yaptığımız düzenleme ile Çizelge 4.1'de gösterilen sahte ve gerçek yorumların etiketlendiği bir cvs dosyası oluşturuldu. Sahte yorumları 0 ve gerçek yorumları 1 ile etiketlendi.

Genel olarak, inceleme metni "is", "the"," and", "a" gibi gereksiz kelimeler içermektedir. Bu kelimeler spam incelemelerini tespit etmede yardımcı olmaz, bu nedenle gürültü ve gereksiz belirteçleri önlemek için tokenize etmeden önce bunları çıkarmak daha iyidir. Örneğin, "This is a very good car" durdurma sözcüklerini ve noktalama işaretlerini kaldırdıktan sonra, inceleme "good car" olarak görünür. Çizelge 4.2'de verilen örnek yorumlar Çizelge 4.3'de NTLK kütüphanesi kullanılarak gereksiz kelimeler ve belirteçler yorumlardan temizlenerek tüm harfler küçük harflere dönüştürülmüştür

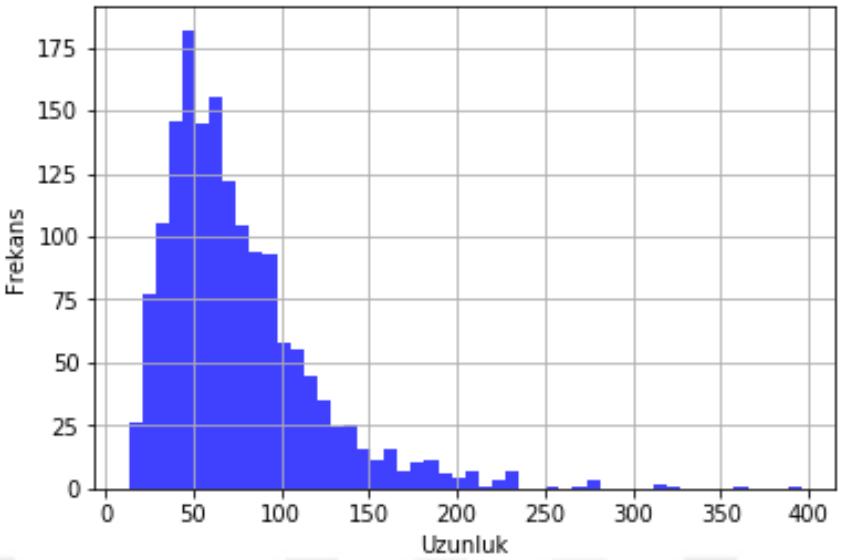
Çizelge 4.2. Veri seti gösterimi

No	**Durum (Gerçek=1) (Sahte =0)**	**Yorum**
1222	1	The James hotel is located close to everything...
144	0	I recently stayed in a room at the trendy Talb...
94	0	Extremely disappointed in our stay. This hotel...
1197	0	We really enjoyed our stay at the Palmer House...
600	1	no bell boys there when you need them sully s...
833	0	The James Chicago Hotel is located right in th...
154	0	I was going on a 3 day personal trip to Chicag...
296	0	I was really expecting a lot more from a quali...
220	0	I recently stayed at the Hyatt Regency Chicago...
544	1	Avoid The Talbot especially for group functio...

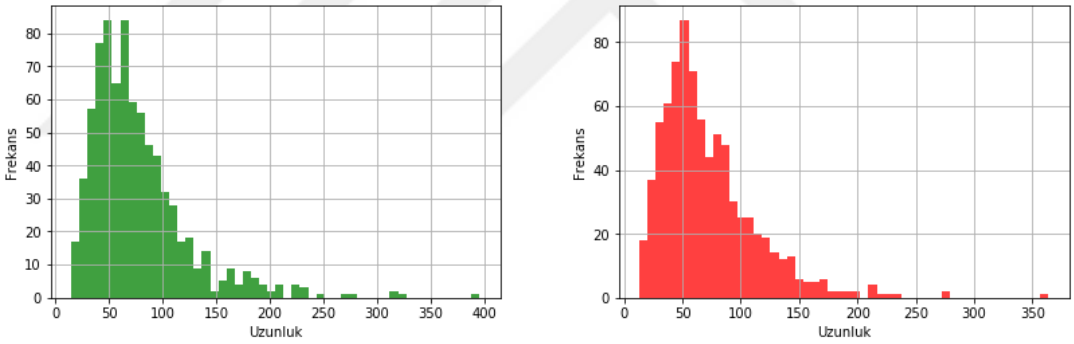
Çizelge 4.3. Veri setinden gürültülerin temizlenmesi

No	Durum (Gerçek=1) (Sahte =0)	Yorum
1222	1	james hotel located close everything...
144	0	recently stayed room trendy Talb...
94	0	extremely disappointed our stay this hotel...
1197	0	really enjoyed our stay palmer house...
600	1	bell boys there when you need them sully s...
833	0	james chicago hotel located right ...
154	0	going day personal trip chicag...
296	0	really expecting lot more from quali...
220	0	recently stayed hyatt regency chicago...
544	1	avoid talbot especially for group functio...

Veri setine uygulanan ön işlem sonrasında yorumlarda kullanılan tüm kelimelerin frekans dağılımları Şekil 4.1'de, sahte ve gerçek kullanıcı yorumlarını kelime dağılımları Şekil 4.2'de Matplotlib Python 2D çizim kütüphanesi kullanılarak grafiklerin gösterimi gerçekleştirilmiştir.



Şekil 4. 1. Veri setinde bulunan tüm yorumların kelime dağılımları



(a) **(b)**

Şekil 4. 2. (a) Gerçek yorumlarda kelime dağılımı **(b)** Sahte yorumlarda kelime dağılımı

Keras, derin öğrenme için metin belgeleri hazırlamak için Tokenizer sınıfını sağlar. Tokinezer sınıfı, eğitim verisini oluşturmak için kelime sıklığına dayalı kelime endeksini oluşturur. Tokinezer işlemi sonrası eğitim setinin ekran çıktısı şu şekildedir:

Çıktı:

```
array([[  0,   0,   0, ...,  50, 102,   1],
       [  0,   0,   0, ...,   3, 102,   4],
       [  0,   0,   0, ...,   4, 118,   1],
       ...,
       [  0,   0,   0, ...,   1,  25,  90],
       [  0,   0,   0, ..., 147,   1,  46],
       [  0,   0,   0, ...,   5,   3,  72]])
```

4.3. Değerlendirme Ölçütleri

Çizelge 4.4. Karşıtlık Matrisi

	Öngörülen Sınıf	
Gerçek Sınıf	Gerçek Pozitif (TP)	Sahte Negatif (FN)
	Sahte Pozitif (FP)	Gerçek Negatif (TN)

Doğruluk (Accuracy): Toplam kayıt sayısına göre doğru sınıflandırılmış kayıtların yüzdesi olarak tanımlanır.

$$Doğruluk(Accuracy) = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.1)$$

Kesinlik (P): Gerçek pozitif (TP) ve yanlış pozitif (FP) sınıflandırılmış kayıtların toplamına bölünen gerçek pozitif (TP) kayıt sayısının % oranı olarak tanımlanır.

$$Keskinlik(Precision) = \frac{TP}{TP+FP} \quad (4.2)$$

Hassasiyet (Recall): Gerçek pozitiflerin ve gerçek negatiflerin (FN) sınıflandırılmış kayıtların toplamına bölünmesiyle elde edilen gerçek pozitif kayıt sayısının % oranı olarak tanımlanır.

$$Hassasiyet(Recall) = \frac{TP}{TP+FN} \quad (4.3)$$

F-Ölçü (F-measure, F1-score): Keskinlik ve hassasiyet değerlerinin harmonik ortalamasıdır. Test edilen verilerin doğrulunun ölçütüdür. Genellikle ikili sınıflandırma problemlerinde kullanılır. Sonuç 1 değerine yaklaştıkça kullanılan yöntemin iyi çalıştığını göstermektedir.

$$F = \frac{2\times(Keskinlik\times Hassasiyet)}{(Keskinlik+Hassasiyet)} \quad (4.4)$$

Doğruluk, doğru olarak tahmin edilen sonuçları ifade etmektedir. Kesinlik, doğruluk hesaplamasında tahmin edilen doğru sonuçlar içinde hangilerinin kesin doğru sınıf içinde olduklarını gösterir. Dolayısıyla kesinlik değeri doğruluk değerinden küçük veya eşit olabilir. (Powers, 2011)

Bu ölçütlerin hesaplamasında kullanılan TP, TN, FP ve FN terimleri şunlardır:

- **Doğru tahmin edilen gerçek yorum sayısı(True Positive-TP):** Model, girdi olarak verilen gerçek kullanıcı yorumunu gerçek olarak tahmin eder ve yorumun durum etiketi gerçek yani 1 ise TP kabul edilir. TP sayısı gerçek tahmin edilmiş, durum etiketi gerçek olarak işaretlenmiş yorumların sayısını göstermektedir.

- **Doğru tahmin edilen sahte yorum sayısı(True Negative-TN):** Model, girdi olarak verilen sahte kullanıcı yorumunu sahte olarak tahmin eder ve

yorumun durum etiketi sahte yani 0 ise TN kabul edilir. TN sayısı sahte tahmin edilmiş, durum etiketi sahte olarak işaretlenmiş yorumların sayısını göstermektedir.

- **Yanlış tahmin edilen gerçek yorum sayısı(False Positive-FP):** Model, girdi olarak verilen gerçek kullanıcı yorumunu sahte olarak tahmin eder ve yorumun durum etiketi gerçek yani 1 ise FP kabul edilir. FP sayısı sahte tahmin edilmiş, durum etiketi gerçek olarak işaretlenmiş yorumların sayısını göstermektedir.

- **Yanlış tahmin edilen sahte yorum sayısı(False Negative-FN):** Model, girdi olarak verilen sahte kullanıcı yorumunu gerçek olarak tahmin eder ve yorumun durum etiketi sahte yani 0 ise FN kabul edilir. FN sayısı gerçek tahmin edilmiş, durum etiketi sahte olarak işaretlenmiş yorumların sayısını göstermektedir.

4.4.Deney Sonuçları

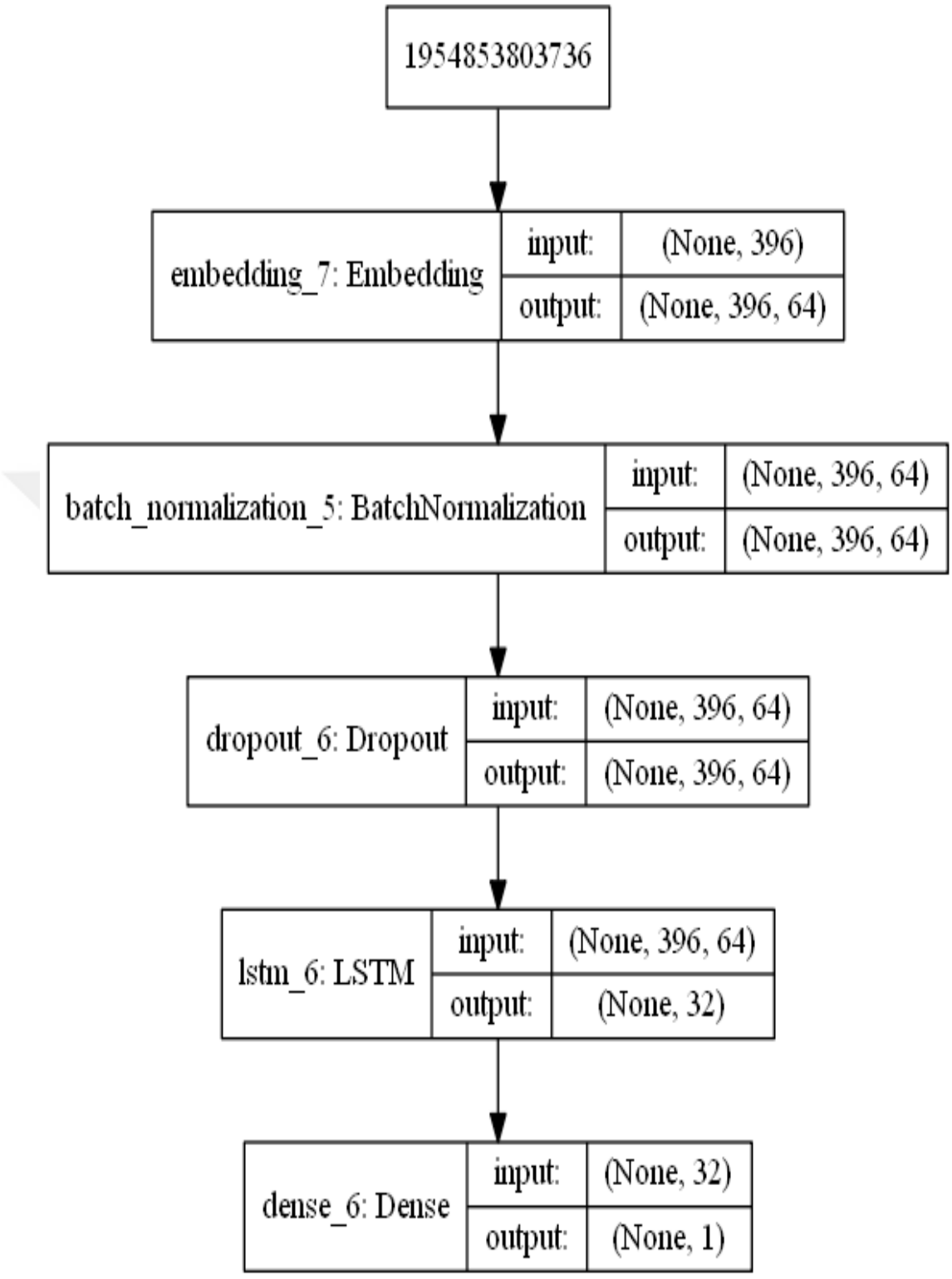
Bu bölümde LSTM, BiLSTM ve CNN-LSTM modellerinin sahte kullanıcı yorumlarını tespit etmedeki performansları tartışılmıştır. Ön işlemden geçen veri seti üzerinde 3 farklı modelle yapılan testler incelenmektedir. Modellerin eğitimi için veri setinin %80'i, test için %20'si rastgele olarak seçilmiş ve her modele aynı eğitim ve test veri kümeleri uygulanmıştır. Böylelikle bu tez çalışmasında sahte kullanıcı yorumlarının tespitinde makine öğrenmesi algoritmaları ve derin öğrenme modelleri kullanılarak yapılan çalışmalar incelenmiş ve daha iyi performanslara ulaşılması amaçlanmıştır.

4.4.1. LSTM Modeli Eğitimi ve Test Sonuçları

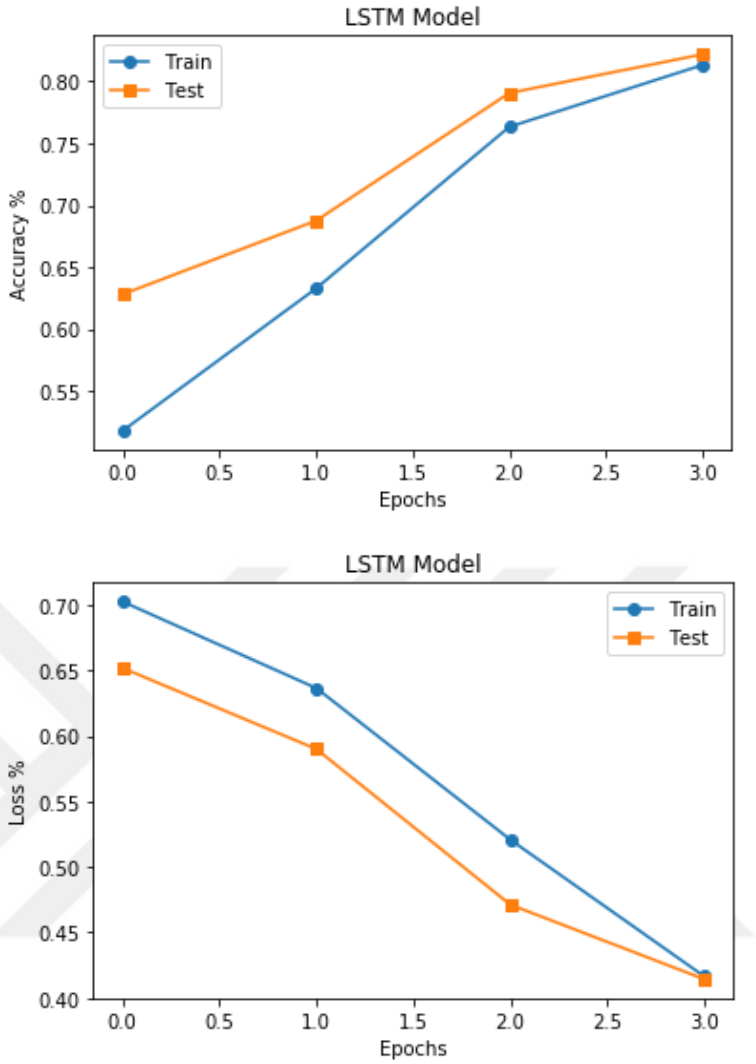
Derin öğrenme uygulamalarının en temel sorunlarından birisi aşırı öğrenmedir. Gizli katmanlarda tutulan parametrelerin sayısındaki artıştan kaynaklı ezberleme problemini önlemek için oluşturulan modelde giriş katmanında dropout yöntemi kullanılmaktadır. (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014)

Dropout yöntemi ile bırakılan nöronlar ileri ve geri yayılıma katılmazlar. Böylece, modele girdi her sunulduğunda model farklı bir mimariyi örnekler. (Krizhevsky, Sutskever, & Hinton, 2012) Model içinde kullanılan LSTM katmanın daha tutarlı ve daha hızlı bir öğrenim gerçekleştirmesini sağlamak için toplu normalleştirme (BacthNormalization) katmanı giriş katmanı sonrasına eklenmektedir. (Cooijmans, Ballas, Laurent, Gülçehre, & Courville, 2016) Bu yöntemle aşırı öğrenme problemi daha düşük bir seviyelerde tutulmaktadır.

Şekil 4.3'de çalışmada kullanılan tek katmanlı LSTM modelinin yapısı gösterilmektedir. LSTM katmanı çoktan-bire bir RNN eğiterek verilen yorumun gerçek bir yorum mu ya da sahte bir yorum mu olduğunu bulmaya çalışmaktadır. Veri setinde yapılan yorumların kelime uzunlukları birbirinden farklı olduğundan dolayı yorumlar 396 kelime ile sınırlandırılmıştır. Şekil 4.4'de modelin eğitim aşamalarında ki doğruluk(Accuracy) oranındaki artma ve hata(Loss) oranındaki azalma devir sayılarına göre grafikle gösterilmektedir. Her iki grafikte eğitim ve test performanslarının devir sayısı arttıkça birbirine yakınlaştığı görülmektedir. Bu durum aşırı öğrenme için kullandığımız dropout katmanın modelin aşırı öğrenmesine engel olduğunu göstermektedir.



Şekil 4. 3. LSTM model mimarisi



Şekil 4. 4. LSTM modeli eğitim(Train) ve doğrulama(Test) sonuçları

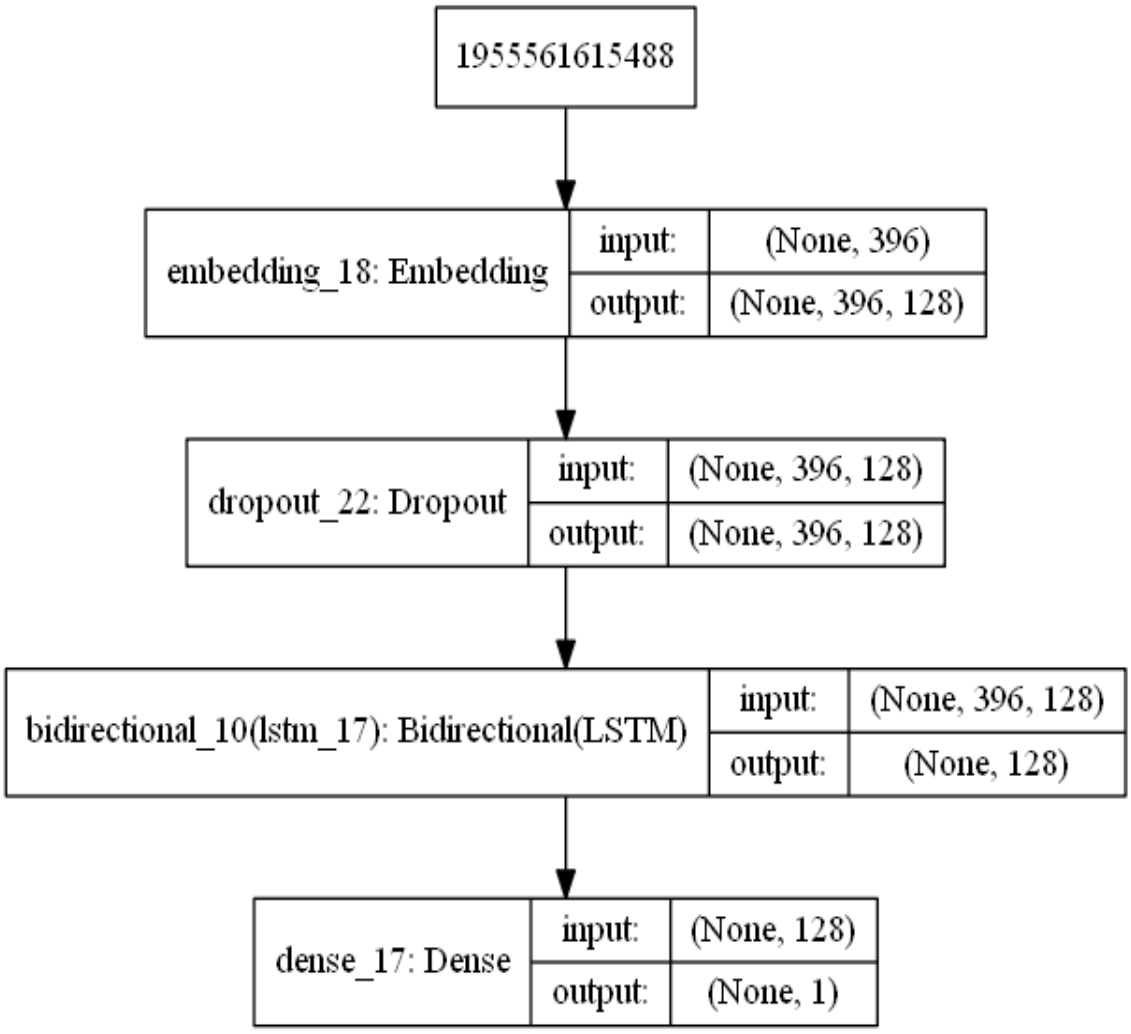
Yapılan eğitim ve testler sonrasında LSTM modelin değerlendirme sonuçları Çizelge 4.5'de verilmektedir. Model sahte yorumların tespitinde %82'lik bir doğruluk oranı sağlamaktadır.

Çizelge 4.5. LSTM Model test sonuçları

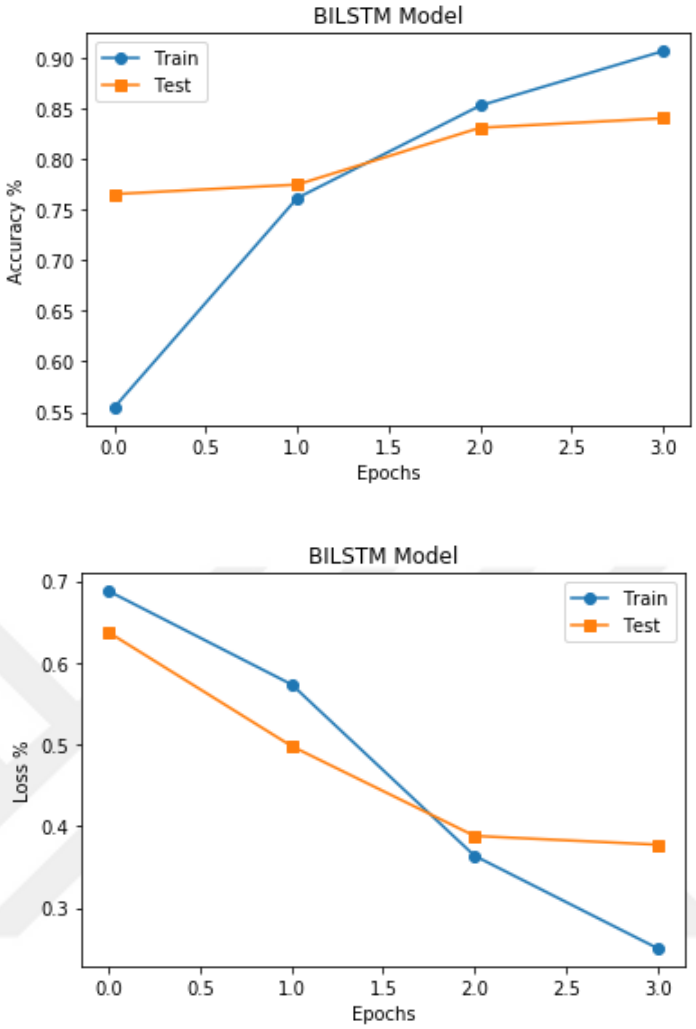
	Accuracy	**Precision**	**Recall**	**F1-score**
LSTM Model	0.82	0.82	0.82	0.81

4.4.2.BiLSTM Modeli Eğitimi ve Test Sonuçları

BiLSTM modelinin eğitimde ve test aşamasında veri kümesi değiştirilmeden gerçekleştirim yapılmıştır. BiLSTM'ler LSTM modelinin aksine sadece ileriye dönük hareket etmezler. Modelde ters yönde çalışan iki gizli LSTM katmanı bulunmaktadır. İki gizli katmanı da kullanarak aynı zamanda geriye ve ileriye doğruda hareket edebilirler. Bir katman soldan sağa kelimeleri öğrenirken, diğer katman sağdan sola doğru kelimeleri öğrenmektedir. Şekil 4.5'de çalışmada kullanılan tek katmanlı BiLSTM modelinin yapısı gösterilmektedir. Aşırı öğrenmeyi engellemek için dropout katmanı, LSTM model testinde olduğu gibi giriş katmanında kullanılmaktadır.



Şekil 4. 5. BiLSTM model mimarisi



Şekil 4. 6. BiLSTM modeli eğitim(Train) ve doğrulama(Test) sonuçları

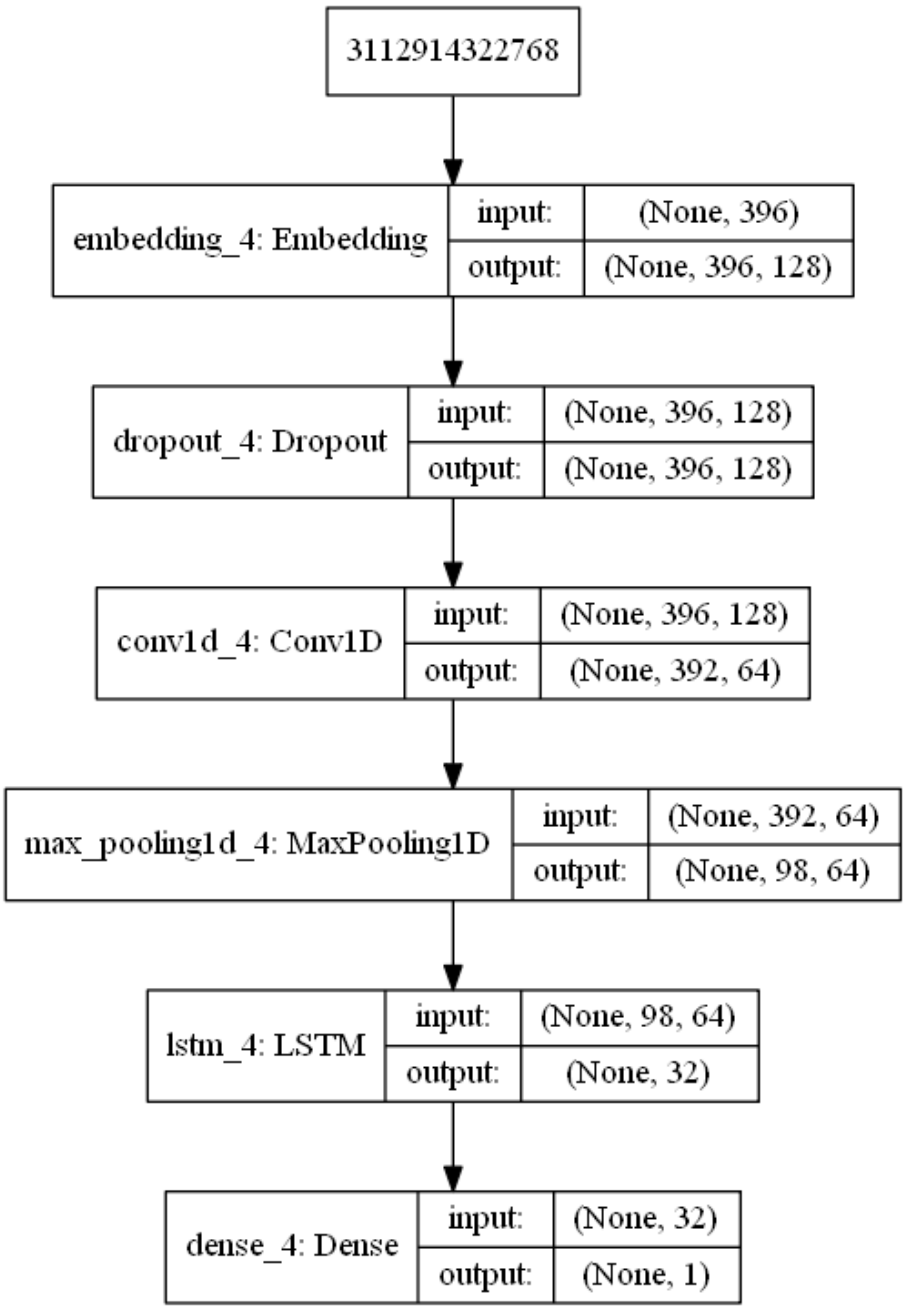
Şekil 4.6'da modelin eğitim ve test aşamalarındaki doğruluk ve hata oranları görülmektedir. Modelde dropout katmanı kullanılmasına rağmen 3. devirden(epochs) sonra aşırı öğrenme eğilime girdiği görünmektedir. Buna rağmen model Çizelge 4.6'da gösterildiği gibi %84 doğruluk oranına ulaşarak LSTM modelinden daha iyi bir performans sergilemiştir.

Çizelge 4.6. BiLSTM Model test sonuçları

	Accuracy	**Precision**	**Recall**	**F1-score**
BiLSTM Model	0.84	0.84	0.84	0.81

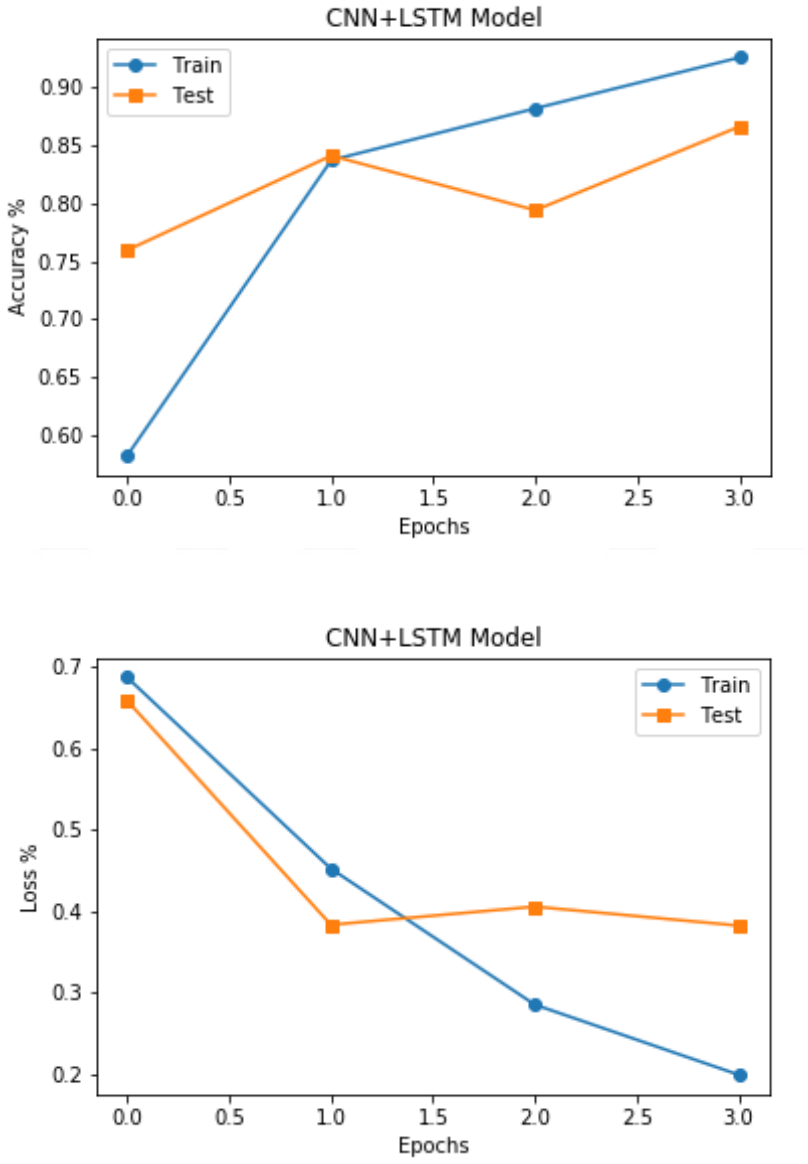
4.4.3.CNN+LSTM Modeli Eğitimi ve Test Sonuçları

CNN+LSTM modeli görüntülerin metinsel açıklamaları sınıflandırmak için kullanılan bir mimaridir. (Vinyals, Toshev, Bengio, & Erhan, 2014) CNN modeli, sıralı bilgileri öğrenmede zayıf kalmasına rağmen lokal özellik çıkarımında oldukça başarılıdır.



Şekil 4. 7. CNN+LSTM model mimarisi

Bir CNN+LSTM modeli, Şekil 4.7'de gösterildiği gibi giriş katmanın ardına özellik çıkarımda faydalı olabilecek bir CNN katmanı eklenmesi ile oluşmaktadır. Bu özellikleri zaman adımlarında yorumlamak içinde LSTM katmanı kullanılmıştır. Ağın hesaplama zamanını azaltmak için bu iki katmanın arasında maksimum havuzlama katmanı eklenmiştir. Şekil 4.8'de model üzerinde yapılan eğitim ve test sonrasın ulaşılan ölçümlerin grafiği verilmektedir.



Şekil 4. 8. CNN+LSTM modeli eğitim(Train) ve doğrulama(Test) sonuçları

Şekil 4.8’de verilen grafikleri incelediğimizde BiLSTM modelinde olduğu gibi 3. devirden sonra modelin aşırı öğrenme eğilimine girdiği görülmektedir. Buna rağmen model, 1. devir de aşırı öğrenme problemi olmadan %87 doğruluk oranına ulaştığı Çizelge 4.7’de gösterilmektedir.

Çizelge 4.7. CNN+LSTM Model test sonuçları

	Accuracy	**Precision**	**Recall**	**F1-score**
CNN+LSTM Model	0.87	0.86	0.86	0.85

BÖLÜM 5

SONUÇ

Bu tez çalışmasında 3 adet derin öğrenme modeli kullanarak çevrim içi sahte kullanıcı yorumlarının tespiti gerçekleştirilmiştir. Modellerin eğitimi sonrasında yapılan ölçümlerde kullanılan modellerin değerlendirmeleri yapıldığında %87 doğruluk oranıyla CNN+LSTM modeli sahte yorumların tespitinde en iyi performansı göstermektedir. LSTM modeli aşırı öğrenme problemi yaşamadan %82 doğruluk oranına ulaşmıştır. BiLSTM modeli %84 doğruluk oranı ile LSTM modelinden daha yüksek doğruluk oranı ulaşmış olsa da model aşırı öğrenme problemi yaşamaktadır. Çalışmada hedeflenen, Ott ve arkadaşlarının (Ott, Cardie, & Hancock, 2013) geleneksel makine öğrenmesi metotları kullanarak çevrim içi sahte kullanıcı yorumlarının tespitinde ulaştıkları %86'lık doğruluk oranı, bir Derin Öğrenme mimarisi olan CNN+LSTM modeli ile geçilmiştir.

Sahte yorumların tespiti üzerine yaptığımız literatür taramalarında bu alanda yapılan çalışmaların az olduğu görülmüştür. Derin öğrenme uygulamalarının geliştirilmesi için en temel sorunlardan biri olan güvenilir veri seti sorununun bu alanda da yaşandığını fark edilmiştir. Aynı zamanda sahte yorumların tespiti için yapılan çalışmalarda kullanılan veri setlerinin doğruluğunun sorgulanması gerekli olduğunu düşünülmektedir. Tez çalışmasında kullanılan veri seti, sınırlı sayıda yorumdan oluşmaktadır. Bu tez çalışması daha sonraki çalışmalarda az sayıda veri kümesi üzerinde daha yüksek doğruluk oranına ulaşabilecek derin öğrenme modellerinin geliştirilmesine yardımcı olabilecek niteliktedir. Literatürde derin öğrenme yöntem ve teknikleri kullanılarak sahte kullanıcı yorumlarının tespitinde Türkçe dil yapısına sahip, güvenilir ve doğruluğu kanıtlanmış herhangi bir veri seti üzerinde yapılmış bir çalışmaya

rastlanmaması bu çalışmanın ileride yapılabilecek Türkçe çalışmalara temel olabilecek niteliktedir.

KAYNAKLAR

D'Onfro, J. (2013). *A Whopping 20% Of Yelp Reviews Are Fake*. Mayıs 26, 2019 tarihinde Business Insider: https://www.businessinsider.com/20-percent-of-yelp-reviews-fake-2013-9 adresinden erşildi

Abadi, M., Agarwal, A., ve diğerleri. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv preprint arXiv:1603.04467*.

Alpaydın, E. (2010). *İntroduction to Machine Learning, Second Edition.* Cambridge, Massachusetts: The MIT Press.

Anaconda. (2019). Haziran 25, 2019 tarihinde https://www.anaconda.com/ adresinden erşildi

Bengio, Y., Simard, P., & Paolo , F. (1994, March). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks, 5*(2), 157 - 166. doi:10.1109/72.279181

Britz, D. (2015, Eylül 17). *Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs*. Mart 25, 2019 tarihinde http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/ adresinden erşildi

Broder, A. Z. (June 1997). On the resemblance and containment of documents. *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)* (s. 21-29). Salerno, Italy,: IEEE. doi:10.1109/SEQUEN.1997.666900

Clark, J. (2015, Ekim 26). *Google Turning Its Lucrative Web Search Over to AI Machines.* Mayıs 25, 2019 tarihinde Bloomberg: https://www.bloomberg.com/news/articles/2015-10-26/google-turning-its-lucrative-web-search-over-to-ai-machines adresinden erşildi

Cooijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç., & Courville, A. (2016). Recurrent batch normalization. *arXiv preprint arXiv:1603.09025*.

Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Najada , H. A. (2015). Survey of review spam detection using machine learning techniques. *Journal of Big Data, 2*(23). doi:10.1186/s40537-015-0029-9

Cui, Z., Ke, R., & Wang, Y. (2017). Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. *6th International Workshop on Urban Computing (UrbComp 2017).* Nova Scotia, Canada.

Gers, F. A., & Schmidhuber, J. (2001, Nov). LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks, 12*(6), 1333 - 1340. doi:10.1109/72.963769

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.

Gravesa, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks, 18*(5), 602 - 610.

Heydari, A., Tavakoli, M. a., Salim, N., & Heydari, Z. (2015, May 1). Detection of review spam: A survey. *Expert Systems with Applications, 42*(7), 3634-3642.

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

Jindal, N., & Liu, B. (2007). Review spam detection. *Proceedings of the 16th international conference on World Wide Web* (s. 1189-1190). ACM.

Jindal, N., & Liu, B. (2008). Opinion Spam and Analysis. *Proceedings of the 2008 International Conference on Web Search and Data Mining* (s. 219-230). Palo Alto, California, USA: ACM. doi:10.1145/1341531.1341560

Karakuş, B. A., Talo, M., Hallaç, İ. R., & Aydin, G. (2018, July 17). Evaluating deep learning models for sentiment classification. *Concurrency and Computation: Practice and Experience, 20*(21), e4783. doi: https://doi.org/10.1002/cpe.4783

Kawaguchi, K. (2000). A multithreaded software model for backpropagation neural network applications. *ETD Collection for University of Texas,El Paso*.

Keras. (2019, Haziran 15). *Keras: The Python Deep Learning library*. Haziran 15, 2019 tarihinde https://keras.io/ adresinden erşildi

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, (s. 1097-1105). Lake Tahoe, Nevada.

LeCun, Y., Bengio , Y., & Hinton, G. (2015, May). Deep learning. *Nature, 521*, 436-444. doi:https://doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324. doi:10.1109/5.726791

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (s. 253-256). Paris, France: IEEE. doi:10.1109/ISCAS.2010.5537907

Lopez , M. M., & Kalita, J. (2017). Deep Learning applied to NLP. *ArXiv, abs/1703.03091*.

Luca, M. (2011). Reviews, reputation, and revenue: The case of Yelp. com. *Harvard Business School NOM Unit Working Paper*, 12-16.

Maass, W., & Orponen, P. (1998, July). On the Effect of Analog Noise in Discrete-Time Analog Computations. *Neural Computation, 10*(5), 1071 - 1095. doi:10.1162/089976698300017359

Mayford, M., Siegelbaum, S. A., & Kandel, E. R. (2012, June). Synapses and Memory Storage. *Cold Spring Harbor Perspectives in Biology, 4*(6).

McCollum, P. (2019). *An Introduction to Back-Propagation Neural Networks*. Haziran 25, 2019 tarihinde Seattle Robotics Society: http://www.seattlerobotics.org/encoder/199811/neural.html adresinden erşildi

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics,* 115-133.

McGonagle, J., Williams, C., & Khim, J. (2019, Mayıs 15). *Recurrent Neural Network*. Haziran 15, 2019 tarihinde https://brilliant.org/wiki/recurrent-neural-network/ adresinden erşildi

Mitchell, T. (1997). *Machine Learning.* New York: NY:McGraw-Hill.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2014). Deep learning applications and challenges in big data analytics. *Journal of Big Data, 2*(1), 1-21.

Nasrabadi, N. M. (2007, October 1). Pattern Recognition and Machine Learning. *Journal of Electronic Imaging, 16*(4).

Ott, M. (2019, Haziran 25). *Deceptive Opinion Spam Corpus v1.4*. Haziran 25, 2019 tarihinde https://myleott.com/op-spam.html adresinden erşildi

Ott, M., Cardie, C., & Hancock , J. (2012). Estimating the prevalence of deception in online review communities. *Proceedings of the 21st international conference on World Wide Web* (s. 201-210). Lyon, France: ACM. doi:10.1145/2187836.2187864

Ott, M., Cardie, C., & Hancock, J. T. (2013). Negative deceptive opinion spam. *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies* (s. 497-501). Atlanta, Georgia: NAACL.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. *HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (s. 309-319). Portland, Oregon: Association for Computational Linguistics.

Powers, D. (2011). Evaluation: From precision, recall and fmeasure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies, 2*(1), 37-63.

Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., & Pande, V. (2015). Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*.

Rayana, S., & Akoglu, L. (2015). Collective Opinion Spam Detection: Bridging Review Networks and Metadata. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (s. 985-994). Sydney, NSW, Australia: ACM. doi:10.1145/2783258.2783370

Ren, Y., & Ji, D. (2019). Learning to Detect Deceptive Opinion Spam: A Survey. *IEEE Access, 7*, 42934-42945. doi:10.1109/ACCESS.2019.2908495

Ren, Y., & Zhang, Y. (2016). Deceptive Opinion Spam Detection Using Neural Network. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (s. 140-150). Osaka, Japan: The COLING 2016 Organizing Committee.

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review, 65*(6), 386-408.

Rumelhart, D. E., Hinton, G. E., & Williams , R. J. (1986, October). Learning representations by back-propagating errors. *Nature, 323*, 533–536.

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.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 3104-3112.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going Deeper With Convolutions. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (s. 1-9).

TensorFlow. (2019, Mayıs 20). *Introduction to TensorFlow*. Mayıs 20, 2019 tarihinde https://www.tensorflow.org/ adresinden erşildi

Tynan, D. (2017, Ağustos 09). *What is machine learning? Software derived from data*. Mayıs 25, 2019 tarihinde https://www.javaworld.com/article/3215266/what-is-machine-learning-software-derived-from-data.html adresinden erşildi

Udacity Deep Learning Team. (2019, Haziran 25). *Long Short-Term Memory Networks*. Mart 26, 2019 tarihinde https://www.udacity.com/course/deep-learning-nanodegree--nd101 adresinden erşildi

Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. *Advances in Neural Information Processing Systems*, (s. 2692-2700).

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2014). Show and Tell: A Neural Image Caption Generator. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (s. 3156-3164).

Wang, C. C., Day, M. Y., Chen, C. C., & Liou, J. W. (2018). Detecting spamming reviews using long short-term memory recurrent neural network framework. *Proceedings of the 2Nd International Conference on E-commerce, E-Business and E-Government* (s. 16-20). New York, NY, USA: ACM. doi:10.1145/3234781.3234794

Wang, Z., Zhang, Y., & Qian, T. (2017). *Fake Review Detection on Yelp.* Stanford University.

Yao, Y., & Huang, Z. (2016). Bi-directional LSTM Recurrent Neural Network for Chinese Word Segmentation. *International Conference on Neural Information Processing(ICONIP 2016). 9950*, s. 345-353. Springer, Cham. doi:978-3-319-46680-4

ÖZGEÇMİŞ

<table>
<tr><td colspan="4">Kenan TAŞAĞAL
kenantasagal@trakya.edu.tr</td></tr>
<tr><td colspan="3">Eğitim Durumu</td><td rowspan="4">  </td></tr>
<tr><td>2014-2016</td><td>Trakya Üniversitesi
Bilgisayar Mühendisliği</td><td>Edirne,
Türkiye</td></tr>
<tr><td>2004-2007</td><td>Mersin Üniversitesi
Tarsus Teknik Eğitim Fakültesi
Bilgisayar Öğretmenliği</td><td>Mersin,
Türkiye</td></tr>
<tr><td>2002-2004</td><td>Mersin Üniversitesi Tarsus Meslek Yüksek Okulu
Bilgisayar Programcılığı</td><td>Mersin,
Türkiye</td></tr>
<tr><td colspan="4">İş Deneyimi</td></tr>
<tr><td>2008-</td><td colspan="3">Bilişim Teknolojileri Öğretmeni-Edirne Milli Eğitim Müdürlüğü
Serhat İMKB Mesleki ve Teknik Anadolu Lisesi</td></tr>
<tr><td>2015</td><td colspan="3">Proje Yöneticisi / Veri Tabanı Yöneticisi
Meriç Uluslararası Sosyal ve Stratejik Araştırma Dergisi</td></tr>
<tr><td>2013-2014</td><td colspan="3">Yazılım Geliştirici-BestERA Yazılım</td></tr>
<tr><td>2009</td><td colspan="3">Yazılım Geliştirici-Eviminfiyati.com</td></tr>
<tr><td>2007-2008</td><td colspan="3">Yazılım Geliştirici -İst Modern Yazılım</td></tr>
<tr><td colspan="4">Sertifikalar</td></tr>
<tr><td>2010</td><td colspan="3">CCNA Discovery: Networking for Home and Small Businesses</td></tr>
<tr><td>2010</td><td colspan="3">IT Essentials: PC Hardware and Software</td></tr>
<tr><td>2008</td><td colspan="3">C&System Programmers Association Main Branch
Java Programming Language</td></tr>
</table>

TEZ ÖĞRENCİSİNE AİT TEZ İLGİLİ BİLİMSEL FAALİYETLER

K. Taşağal, Ö. Uçar, Detection of Fake User Reviews with Deep Learning, International Journal of Research in Engineering and Applied Sciences(IJREAS) Vol. 8 Issue 12, December -2018 ISSN (O): 2249-3905, ISSN(P): 2349-6525

Özlem Uçar , Kenan Taşağal, Using Lstm Neural Networks For Fake Online User Review Detection, International Scientific Conference, 16-17 November 2018, Gabrovo