

T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YARATICILIK ÜZERİNE DERİN ÖĞRENME YAKLAŞIMLARI
VE BİR ÖRNEK UYGULAMA

Burak BEYNEK

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Altan MESUT

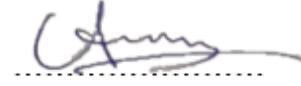
EDİRNE – 2020

Burak BEYNEK'in hazırladığı “**Yaratıcılık Üzerine Derin Öğrenme Yaklaşımları ve Bir Örnek Uygulama**” başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisliği Anabilim Dalında bir **Yüksek Lisans** tezi olarak kabul edilmiştir.

Jüri Üyeleri:

İmza

Dr. Öğr. Üyesi Altan MESUT



Dr. Öğr. Üyesi Bora ASLAN



Dr. Öğr. Üyesi Deniz TAŞKIN



Tez Savunma Tarihi:

Bu tezin **Yüksek Lisans** olarak gerekli şartları sağladığını onaylarım.



Dr. Öğr. Üyesi Altan MESUT

Tez Danışmanı

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı



Prof. Dr. Muşat 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.

24/01/2020

Burak BEYNEK

Yüksek Lisans Tezi

Yaratıcılık Üzerine Derin Öğrenme Yaklaşımları ve Bir Örnek Uygulama

T.Ü. Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

İnsanoğlu sürekli olarak kendisi gibi zeki, yaratıcı, duygusal başka varlıkların olup olmadığını merak etmiştir. İnsana benzeyen makinelerin yapılıp yapılamayacağı da hala çok tartışılan ve merak edilen bir konudur. Günümüzde derin öğrenme ağları kendi kendine öğrenme yetenekleri sayesinde bu konuya en yakın sayılabilecek bir araştırma alanıdır. Bu tez çalışmasında yapay sinir ağlarının bir alt başlığı olan derin öğrenme sistemleri yaratıcılık açısından incelenmiştir. Söz konusu olan derin öğrenme sistemini kullanan yapay zekâ çalışmaları girdi, çıktı ve çalışma prensipleri açısından yaratıcılık özelliğine sahip olup olmadıkları değerlendirilmiştir. Örnek bir test olarak DeepWriting isimli uygulamaya, Yüzüklerin Efendisi gibi üslup olarak zorlayıcı metinler İngilizce ve Türkçe dillerinde girdi olarak verilmiş sonuç olarak yeni bir metin elde edilmeye çalışılmıştır. Elde edilen sonuçlar ışığında derin öğrenme ağlarının yaratıcılık açısından potansiyeli hakkında fikir sahibi olunmuştur.

Yıl: 2020

Sayfa Sayısı: 56

Anahtar Kelimeler: Derin Öğrenme, Yapay Sinir Ağları, Yapay Zekâ, Yaratıcılık.

Masters Thesis

Deep Learning Approaches to Creativity and a Sample Application

Trakya University Institute of Natural Sciences

Computer Engineering Department

ABSTRACT

People have constantly wondered if there are other intelligent, creative, emotional beings like themselves. Whether human-like machines can be built is still a matter of much debate and wonder. Today, deep learning networks are a research area that can be considered closest to this subject thanks to their self-learning abilities. In this thesis, deep learning systems, a subheading of artificial neural networks, are examined in terms of creativity. Artificial intelligence studies using the deep learning system in question were evaluated in terms of creativity using input, output and working principles. As a sample test, the application called DeepWriting was re-performed, and compelling texts such as The Lord of the Rings were given in English and Turkish languages and a new text was obtained. In the light of the results obtained, an idea was gained about the potential of deep learning networks in terms of creativity.

Year: 2020

Number of Pages: 56

Keywords: Deep Learning, Artificial Neural Networks, Artificial Intelligence, Creativity

TEŐEKKÜR

Tez alıőmam sırasında kıymetli bilgi ve tecrübeleri ile bana yol gösteren ve desteęini esirgemeyen danıőmanım Dr. Öğr. Üyesi Altan MESUT'a, yüksek lisans eğitimim süresince bilgi ve tecrübelerini paylaşan Bilgisayar Mühendislięi bölümündeki tüm hocalarıma teşekkürlerimi sunuyorum.

alıőmalarım boyunca yardımını hiç esirgemeyen Kırklareli Üniversitesi Yazılım Mühendislięindeki deęerli hocalarım ve meslek arkadaşlarıma teşekkürü borç bilirim.

Bu günlere gelmemde emeęi geen ve hiçbir zaman desteęini esirgemeyen annem Prof. Dr. Nestrin BEYNEK, babam Dr. Öğr. Üyesi Hayrettin BEYNEK ve erkek kardeőim Barıő BEYNEK'e sonsuz teşekkür ederim.

İÇİNDEKİLER

ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
KISALTMALAR.....	ix
ÇİZELGELER.....	x
ŞEKİLLER.....	xi
BÖLÜM 1: GİRİŞ.....	1
BÖLÜM 2: YÖNTEMLER.....	4
2.1. Makine Öğrenmesi.....	4
2.2. Vaka-Bazlı Akıl Yürütme (Case-Based Reasoning: CBR).....	5
2.3. Kural Bazlı Sistemler (Rule-Based Systems RBS).....	7
2.4. Yapay Sinir Ağları.....	8
2.5. Reinforcement Learning (RL, Pekiştirmeli Öğrenme).....	11
2.6. Bilgisayarlı Görü (Computer Vision).....	11
2.7. Derin Öğrenme.....	12
2.8. Generative Adversarial Nets (GAN).....	15
BÖLÜM 3: YARATICILIK ÜZERİNE YAPAY ZEKÂ UYGULAMALARI.....	17
3.1. Resim Üzerine Uygulamalar.....	17
3.1.1. Pix2Pix.....	17
3.1.2. Plug & Play.....	18
3.1.3. Metinsel Görüntü Tanımı Oluşturma.....	19
3.1.4. Fotoğraf Stilini Değiştirme.....	21
3.1.5. Neural-Style.....	22
3.1.6. DeepDream.....	23
3.2. Müzik Üzerine Uygulamalar.....	23
3.2.1. Klasik Müzik Kompozisyonu Yazma.....	23

3.2.2. Vocaloid.....	25
3.3. Yazı Üzerine Uygulamalar	26
3.3.1. 1 the Road	26
3.3.2. Deep Writing.....	26
BÖLÜM 4: DENEYSEL ÇALIŞMA	28
4.1. TensorFlow Kütüphanesi.....	28
4.2. Materyal.....	29
4.3. Eğitim	30
4.4. Yeni Metin Elde Etme	32
4.5. Değerlendirme	33
BÖLÜM 5: SONUÇLAR.....	35
KAYNAKLAR	37
ÖZGEÇMİŞ	43
TEZ ÖĞRENCİSİNE AİT AKADEMİK MAKALELER.....	45

KISALTMALAR

AI	: Artificial Intelligence
BG	: Bilgisayarlı Görü
CBR	: Case-Based Reasoning
CNN	: Convolutional Neural Network
DDİ	: Doğal Dil İşleme
GAN	: Generative Adversarial Nets
Gİ	: Görüntü İşleme
KNN	: K-Nearest Neighbors
RBS	: Rule-Based Systems
RL	: Reinforcement Learning
Sİ	: Sinyal İşleme
SR	: Sağlık ve Robotik
TDK	: Türk Dil Kurumu
YSA	: Yapay Sinir Ağı
YZ	: Yapay Zekâ

ÇİZELGELER

Çizelge 2.1. Derin Öğrenme Üzerinde Çalışan Araştırma Ekipleri.....	15
--	----



ŞEKİLLER

Şekil 1.1. Deep-Blue vs Garry Kasparov	2
Şekil 2.1. Makine Öğrenmesi Adımları	5
Şekil 2.2. Vaka-Bazlı Akıl Yürütme Adımları	6
Şekil 2.3. Erne Yapay Sinir Ağı Modeli	9
Şekil 2.4. Yapay Sinir Ağları ile Çözölebilecek Problemler	10
Şekil 2.5. İlk Derin Öğrenme Çalışması	12
Şekil 2.6. Derin Öğrenme Adımları	13
Şekil 2.7. Derin Öğrenme Örneđi	14
Şekil 2.8. GAN çalışma yapısı	16
Şekil 3.1. Pix2Pix Dönüşümleri	18
Şekil 3.2. Plug & Play Dönüşümleri	19
Şekil 3.3. Resimlerde Obje Sınıflandırma	20
Şekil 3.4. Resimden Eylem ve Obje Çıkarsama	20
Şekil 3.5. Fotoğraf Stil Deđişikliği Örneđi	21
Şekil 3.6. Resim Stil Deđişikliği Örneđi	22
Şekil 3.7. DeepDream Örneđi	23
Şekil 3.8. Bilgisayar Tarafından Yazılmış Bir Parça	24
Şekil 3.9. Hatsune Miku Canlı Konseri	25
Şekil 3.10. Harf Tahmini Çalışma Prensipleri	27
Şekil 4.1. TensorFlow Çalışma Şeması	29
Şekil 4.2. İngilizce metin için kayıp/iterasyon grafiđi	31
Şekil 4.3. Türkçe metin için kayıp/iterasyon grafiđi	31

BÖLÜM 1

GİRİŞ

Bilim insanları Antik Yunandan beri kendi kendine düşünebilen makineler yapmayı düşlemiştir. Yunan mitolojisinde Pygmalion, Daedalus ve Hephaestus gibi figürler bu tarz makineler yapabilecek mucitler olarak tasvirlemiştir. Yapay Zekâ (Artificial Intelligence: AI) çalışmalarından yüzyıllar önce, programlanabilir bilgisayarların ilk tasarlandığı dönemlerde insanlar makinelerin kendi zekâsının olabileceğini öngörmüşlerdir.

Yapay Zekâ geliştirildiği ilk zamanlarda, sadece insanların çözmeye zorlandığı problemleri hızlı bir şekilde çözmek için kullanılırdı. Bu problemler genelde matematiksel kurallarla ya da biçimsel listeler ile tanımlanabiliyordu. Oysaki yapay zekânın gerçek başarısı insanların kolay bir şekilde yaptığı ama nasıl yaptığını tarif edemeyeceği problemleri çözebilecek olmasıydı. Bu problemler günlük hayatta bizim farkına varmadan çözebildiğimiz şeylerdir. Örneğin; yüzüne bakarak bir insanı tanımak, okuduğumuz yazıdan anlam çıkarmak ya da konuşulanları anlamak gibi. Yapay zekânın ilk zamanlarında başarılı olan pek çok uygulama aslında kısıtlanmış ve biçimsel bir çevrede çalışacak şekilde tasarlanmış ve gerçek dünya ile ilgili fazla bir bilgiye ihtiyaç duymamaktaydı. Mesela IBM'in Deep Blue satranç sistemi dünya şampiyonu Kasparov'u 1997 yılında yenmiştir (Şekil 1.1). Tabi ki satranç 64 konum ve üzerinde belirli kurallar ile hareket edebilen 32 taş ile çok basit bir dünyaya sahip. Dünya şampiyonu bir oyuncuyu yenebilecek bir bilgisayar yapmak tabi ki büyük bir başarı ancak satranç biçimsel bazı kurallar listelenerek tanımlanabilecek bir oyundur. Soyut ve biçimsel kurallar insanlar için anlaması güç olmasına rağmen bilgisayarlar için en kolay işlenecek şeylerdir (Goodfellow, Bengio & Courville, 2016).



Şekil 1.1. Deep-Blue vs Garry Kasparov

Bilgisayarlar en iyi satranç oyuncusunu yenmiş olabilirler ama günümüzde bir çocuğun bile yapabildiği insanları tanıma, konuşma gibi şeyleri yeni yeni yapabilecek hale gelmişlerdir. İnsanın günlük hayatı dünya hakkında inanılmaz büyüklükte bilgiye ihtiyaç duymaktadır. Bu bilginin birçoğu öznel ve sezgisel olduğu için bu bilgiyi biçimsel bir yapıya dönüştürmemiz çok zordur. Yapay zekânın temel problemlerinden biri de bu bilgiyi bilgisayara aktarmaktır. Pek çok yapay zekâ projesi günlük hayatı biçimsel bir dil ile tanımlamaya çalışmışlardır. Bilgisayar bu yöntem ile gelen bilgi üzerinde mantık kurallarını kullanarak yorum yapabilir. Bu çalışmaların hiç biri büyük bir başarı elde etmemiştir. Bunlardan en meşhur olan, Douglas Lenat tarafından 1984'te geliştirilmeye başlanan Cyc'nin yıllar süren çalışmaları sırasında insanlar dünyayı biçimsel listeler halinde bilgisayara anlatmaya çalışmışlardır (Lenat, Prakash & Shepherd, 1986). Ama Cyc bazı hikâyeleri anlayamamıştır. Mesela "Fred sabahları tıraş oluyor." cümlesinde tıraş makinesinin elektrikli bir alet olduğunu bildiği ve insanların elektronik parçası olamayacağını da bildiği için, Fred'in insan olmadığını sonucuna varmıştır.

Günümüzde daha zor problemlere çözüm aranmaktadır. Bu çözümler bilgisayarların problemleri deneyimleyip bu deneyimlerden öğrenmesini ve dünyayı hiyerarşik kavramlar ile anlayabilmesini gerektirmektedir. Bu tip problemlere çözüm

olarak derin öğrenme (deep learning) algoritmaları ortaya çıkmıştır. Derin öğrenmede her bir kavram kendinden daha basit birçok kavramın bağlanması ile oluşmaktadır. Bilgisayarın kendi deneyimleri ile öğrenmeye devam etmesi, aradaki eğitimci insan faktörünü ortadan kaldırmakta ve karmaşık kavramları kendi kendine daha basit kavramları yorumlayıp birleştirerek oluşturmasını sağlamaktadır. Eğer ki bu basit kavramların yorumlanması ve karmaşık olanların oluşumunu bir grafik çizerek gösterirsek katmanların çok fazla sayıda ve çok derine gittiğini görürüz. Derin öğrenmedeki ‘Derin’ kavramı buradan gelmektedir.

2012 yılında ImageNet Büyük Ölçekli Görsel Tanıma Yarışmasında (ImageNet Large Scale Visual Recognition Challenge) Alex Krizhevsky tarafından tasarlanan AlexNet isimli evrimsel sinir ağının (CNN), diğer adıyla derin öğrenme algoritması, büyük başarı kazanması sonucunda, derin öğrenme alanına dikkat çekerek gelecek çalışmalara öncü oldu (Krizhevsky, Sutskever & Hinton, 2012). İlerleyen yıllarda Görsel Geometri Grup Ağları (Visual Geometry Group Networks) daha derin mimariler tasarladı. GoogleNet derin öğrenme için başlangıç modülleri oluşturdu ve ResNet mimarileri ortaya çıktı (He, Zhang, Ren & Sun, 2015). Derin öğrenme algoritmaları yapay zekâ ve makine öğrenmesi ile yapılan sınıflandırma işlemlerinde kullanılmaya başlandı. Bu işlemlerdeki hata oranı yıl geçtikçe azaldı ve eskiden kullanılan kayan pencere gibi yöntemlerin yerini aldı.

Günümüzde derin öğrenme yöntemlerini kullanarak yaratıcı ya da türetici olarak nitelendirilen uygulamalar ortaya çıkmaktadır. Makinelerin yaratıcı olup olmayacağı halen büyük bir tartışma konusudur. Çalışmaları yapan araştırmacılar bazen kendi uygulamalarını yaratıcı bazen ise türetici olarak nitelendirmektedirler. IBM’in 30 yapay zekâ uzmanına sorarak hazırladığı bir yazıda bu sorulara cevap aranmış ama kesin bir sonuç elde edilmemiştir (The quest for AI creativity, 2019).

Bu çalışmada yaratıcı ya da türetici olarak nitelendirilen uygulamaları çalışma prensiplerine göre ve verdikleri sonuçlara göre değerlendirerek gerçekten yaratıcı mı yoksa türetici mi olduklarını değerlendirilmiştir. “Makineler yaratıcı olabilir mi?” sorusuna cevap aranmıştır.

BÖLÜM 2

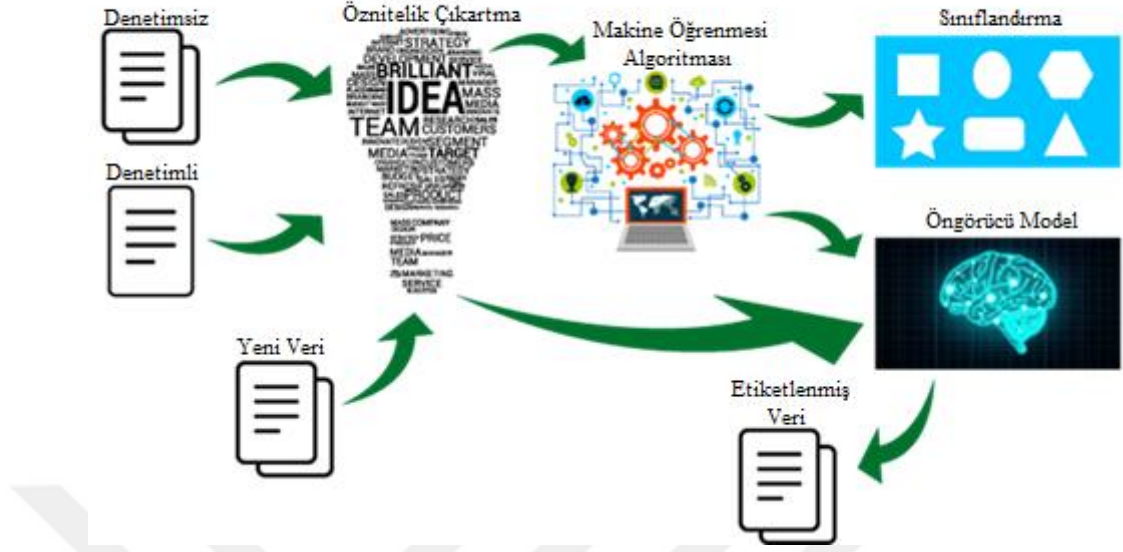
YÖNTEMLER

Geçmişten günümüze bilgisayarların zekileşmesi için pek çok farklı yöntem geliştirilmiştir. Bu yöntemlerden bazıları insanların yapabildiği şeyleri bilgisayarların yapmasını sağlarken bazıları ise bilgisayarların insanların dahi yapamayacağı işleri yapmasına olanak sağlamıştır. Bu bölümde yapay zekâ alanında kullanılan yöntemler anlatılmaktadır.

2.1. Makine Öğrenmesi

Makine öğrenmesi bilgisayar bilimlerinin bir alt dalıdır. Yapay zekâ ve sayısal öğrenme ile doğrudan ilişkili olan makine öğrenmesi 1959 yılından itibaren bir çalışma alanı olarak kabul edilmektedir. Makine öğrenmesi algoritmaları öğrenebilen ve tahmin yapabilen algoritmalardır. Makine öğrenmesinin keşfi ile bilgisayarlar gerçek dünya ile ilgili bilgi toplayabilme ve öznel karar verebilme yeteneğine kavuşmuştur. Makine öğrenmesi algoritmalarının performansı işlenen bilginin gösterim biçimine bağlıdır. Bilgisayar bilimlerinde arama, sıralama, veri toplama gibi işlemler verinin akıllıca yapılandırılıp etiketlenmesi ile hızlanmaktadır. Örneğin Arap rakamları ile işlem yapmak daha kolay iken Roma rakamları ile işlem yapmak daha uzun sürmektedir. Makine öğrenmesi kullanan sistemlerin eğitilmesi gerekir. Bu eğitim sayesinde sistemler karar verme yeteneğine kavuşur. Şekil 2.1'de makine öğrenmesinin işlem adımları yer almaktadır.

Makine Öğrenmesi



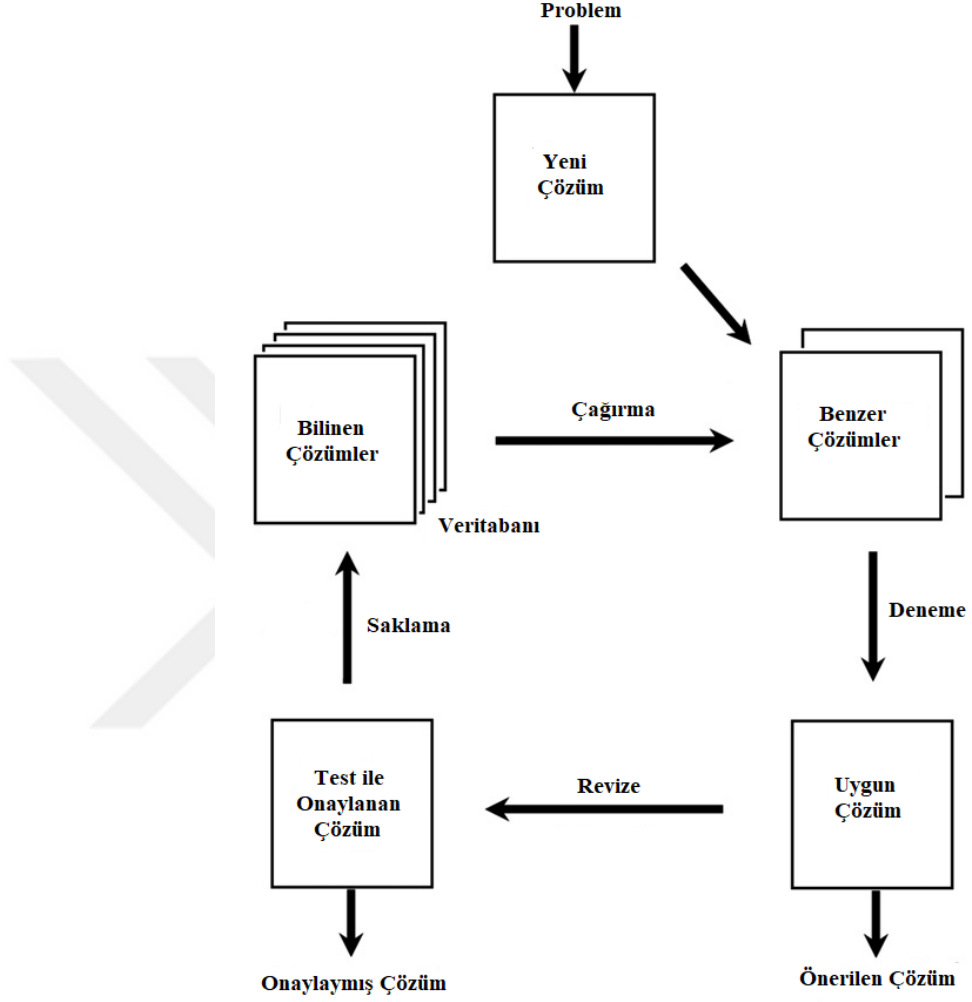
Şekil 2.1. Makine Öğrenmesi Adımları

Makine öğrenmesi yöntemlerinde eğitim denetimli (supervised), denetimsiz (unsupervised) ya da yarı denetimli (semi-supervised) olabilir. Denetimli eğitimde, etiketlenmiş ya da sınıflandırılmış veriler sisteme eğitim amacıyla girdi olarak verilir. Sistem sonuçlar ile veriler arasındaki ilişkileri oluşturur. Oluşturulan ilişki gelecekte gelen problemleri çözmek için kullanılır. Denetimsiz eğitimde, etiketlenmemiş ya da sınıflandırılmamış veriler sisteme eğitim amacı ile verilir. Sistem veriler arasında benzerlikler bularak verileri kümeler. Elde edilen kümeleme yöntemi gelecekteki problemlerin çözümünde kullanılır. Yarı denetimli eğitimde, hem denetimli hem denetimsiz eğitim yöntemleri bir arada kullanılır. Küçük bir kısmı etiketlenmiş ya da sınıflandırılmış veri kümesi eğitim amacı ile sisteme verilir. Sistem denetimsiz eğitimdeki gibi veriler arasında benzerlik bulur ancak oluşturulan benzerlikler etiketlenmiş veriler sayesinde kontrol edilir ve gereken ayarlamalar yapılır. Sonuç olarak elde edilen çözüm yöntemi gelecekteki problemleri çözmek için kullanılır. (Makine Öğrenimi, 2019)

2.2. Vaka-Bazlı Akıl Yürütme (Case-Based Reasoning: CBR)

CBR sistemleri problemleri, benzer problemlerin çözümleri benzer olur mantığı ile çözer (Fdez-Riverola & Corchado, 2004). Yeni çözümlere ulaşabilmek için birçok çözülmüş örnek probleme ihtiyaç duyar. CBR sistemleri problem çözümlerini tekrar

tekrar dener ve her tekrarda daha kolay bir çözüm öğrenir. CBR sistemleri aşağıdaki adımları uygulayarak problem çözer (Şekil 2.2) (Aamodt & Plaza, 1994).



Şekil 2.2. Vaka-Bazlı Akıl Yürütme Adımları

- Probleme benzer en güncel çözümleri veritabanından çağırır.
- Çağrılan çözümleri kullanarak elimizdeki probleme bir çözüm oluşturur.
- Deneme yanılma ya da test yöntemlerini kullanarak önerilen çözümü revize eder.
- Başarılı bir çözüm elde ettikten sonra elde edilen çözümü gelecek problemlerde kullanmak üzere saklar.

Çağırma işlemi sadece sözdizimsel ya da anlamsal benzerliklere bakarak yapılır. Sözdizimsel çağrılar kelime eşleştirme ile yapılır. Anlamsal çağrılar ise problem içerikleri karşılaştırılarak yapılır. Anlamsal çağırma yapan sistemlere gelişmiş CBR sistemleri de denir (Aamodt & Plaza, 1994). Genel olarak karşılaştırma işlemleri K En Yakın Komşu (K-Nearest Neighbors: KNN), tümevarım ya da bilgi-tabanlı algoritmalar kullanılarak yapılır (Salem & Katoua, 2013). Çağırma işleminden sonra çağırılan çözümler yeni probleme adapte edilir. Bu işlem yapısal değişikliklerle ya da çözümü türeterek yapılır (Watson & Marir, 1994). Yapısal değişiklikler eski çözümü modifiye ederek yapılır. Yeni çözüm türetmek ise bir ya da daha fazla algoritma, yöntem ya da kuralı yeni problemi çözmek için kullanarak yapılır. Önerilen çözüm gerekli görüldüğü takdirde yeniden revize edilir. Yeni çözüm onaylandıktan sonra veritabanında saklanır. Önceden saklanan ama gereksiz hale gelen çözümler silinir ya da başka çözümler ile birleştirilir (Fdez-Riverola & Corchado, 2004).

CBR sistemleri veritabanını güncellediğinden dolayı sürekli olarak başarı oranını ve performansını geliştirir. Pek çok değişken içeren büyük verileri etkili bir şekilde kullanır (Watson & Marir, 1994). Ancak CBR sistemleri daha önceden bilgi sahibi olmadığı bir problem ile karşılaştığında çözüm üretmez. Ayrıca CBR sistemleri gözlemlemeye kapalıdır. Sistemin işleyişi ve çıkarsamaları gözlemlenemez. Bu durum CBR sistemlerini çözüm yönteminin önemli olduğu ya da gözlemlenmesi gereken problemlerde kullanışız hale getirir. CBR sistemleri tanı, tahmin, kontrol ve planlama gereken problemlerde kullanışlıdır. Yangın müdahale planlaması (Avesani, Perini & Ricci, 2000), hava kalitesi takibi (Kalapanidas & Avouris, 2001) hava durumu tahmini (San Pedro, Burstein & Sharp, 2005) gibi alanlarda kullanılır.

2.3. Kural Bazlı Sistemler (Rule-Based Systems RBS)

RBS, problemleri uzman bilgilerinden elde edilen kurallar ile çözer (Hayes-Roth, 1985). Kurallar; koşullar ve eylemler olarak iki bölümden oluşur. Eğer (if) ve öyleyse (then) yapıları kullanılarak bir çıkarsama motoru oluşturulur. Bu motor problemin bilgilerini tutan bir bellek, örüntü eşleyici ve kural uygulayıcı parçalarından oluşur. Örüntü eşleyici bellekte tutulan bilgilerin hangi kurallar ile eşleşeceğine karar verir ve kural uygulayıcı o kuralı işleme alır. İşlenen kuraldan elde edilen yeni bilgiler belleğe eklenir. Eşle, seç ve uygula döngüsü eşlenen kural kalmayınca kadar devam eder (Ng

& Abramson, 1990). İleri ve geri zincirli olmak üzere iki tip RBS vardır (Winston, 1984). İleri zincirli RBS veri üzerinden hareket eder. Veriyi kurallar üzerinde eşleştirerek bir sonuç elde eder. Geri zincirli RBS sonuç üzerinden hareket eder. Hipotez noktasından başlayarak uygun kural sıralamasını bulup veri ile sonucun doğruluğunu sınar (Engelmore & Feigenbaum, 1993).

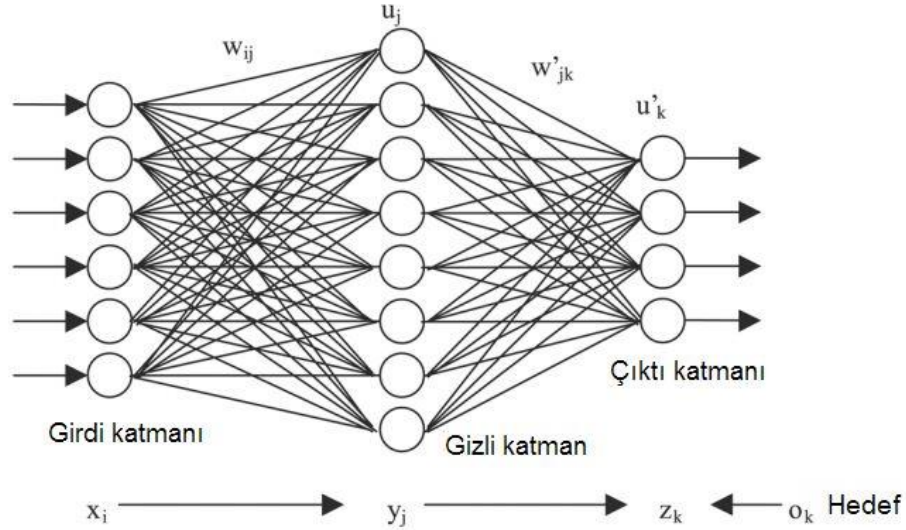
RBS anlaşılırdır, uygulaması ve sürdürülebilirliği kolaydır. Ama bilginin belirli kalıplara uyduğu, kuralların yazılabilir olduğu problemlerde kullanılabilir. Ayrıca RBS öğrenemez ve kendini geliştiremez. Kullanıcı ya da geliştirici tarafından yeni kurallar eklenebilir. RBS sadece yeterli ve doğru bilgi olduğu durumlarda kullanılabilir (Dhar & Stein, 1997). Bu yüzden kullanım alanları çok sınırlıdır. RBS uzmanların emin bir şekilde kural verebildiği ve bilinmeyen değişkenlerin az olduğu problemlerde en verimli çalışır. Etkileşimin çok olduğu karmaşık sistemlerde ve çözüm sürecinin tam olarak anlaşılmadığı problemlerde kullanışsızdır. RBS bitki ve hayvan tanıma, hastalık ve haşere tanıma gibi sınırlandırılmış problemlerde kullanılır (Mahaman, Passam, Sideridis & Yialouris, 2003) (Zetian, Feng, Yun & XiaoShuan, 2005).

2.4. Yapay Sinir Ağları

Yapay sinir ağları (YSA) insan beyninin işleyişini taklit eden sistemlerdir. Bir YSA'da uyum içinde çalışan pek çok işlem birimi yani nöron vardır. Bu nöronlar birbirleri ile belirli ağırlıklara sahip bağlantılar yani sinapslar ile bağlıdırlar. Bu bağlantıların sonucunda veriler arasındaki ilişkileri çıkartarak öğrenebilen bir ağ oluştururlar (Uzun, n.d.).

YSA giriş katmanı, çıkış katmanı ve birçok gizli katmandan oluşur. Bir nöron bir sonraki katmandaki bütün nöronlar ile bağlantılıdır (Şekil 2.3). Nöronlar kendilerine gelen veriyi alır, matematiksel bir fonksiyona sokarak veriyi dönüştürür ve oluşan sonucu bir sonraki katmandaki nöronlara gönderirler (Hammerstrom, 1993). YSA bağlantı ağırlıklarını belirleyebilmek için önceden bir eğitime ihtiyaç duyar. Bu eğitim denetimli ya da denetimsiz olabilir (Rodvold, McLeod, Brandt, Snow & Murphy, 2001). Denetimli eğitimde her eğitim döngüsünden sonra YSA'ya bütün girdilerin doğru sonuçları verilir. Bağlantı ağırlıkları YSA'nın bulduğu sonuçlar ile verilen sonuçların arasındaki hata oranını en aza indirecek şekilde değiştirilir. Denetimsiz eğitimde sadece girdiler verilir ve YSA'nın girdiler arasındaki ilişkileri karşılaştırarak girdileri kategorize etmesi

beklenir (Jain, Mao & Mohiuddin, 1996). Bazı YSA sistemleri hem denetimli hem de denetimsiz eğitimi aynı anda kullanabilir.

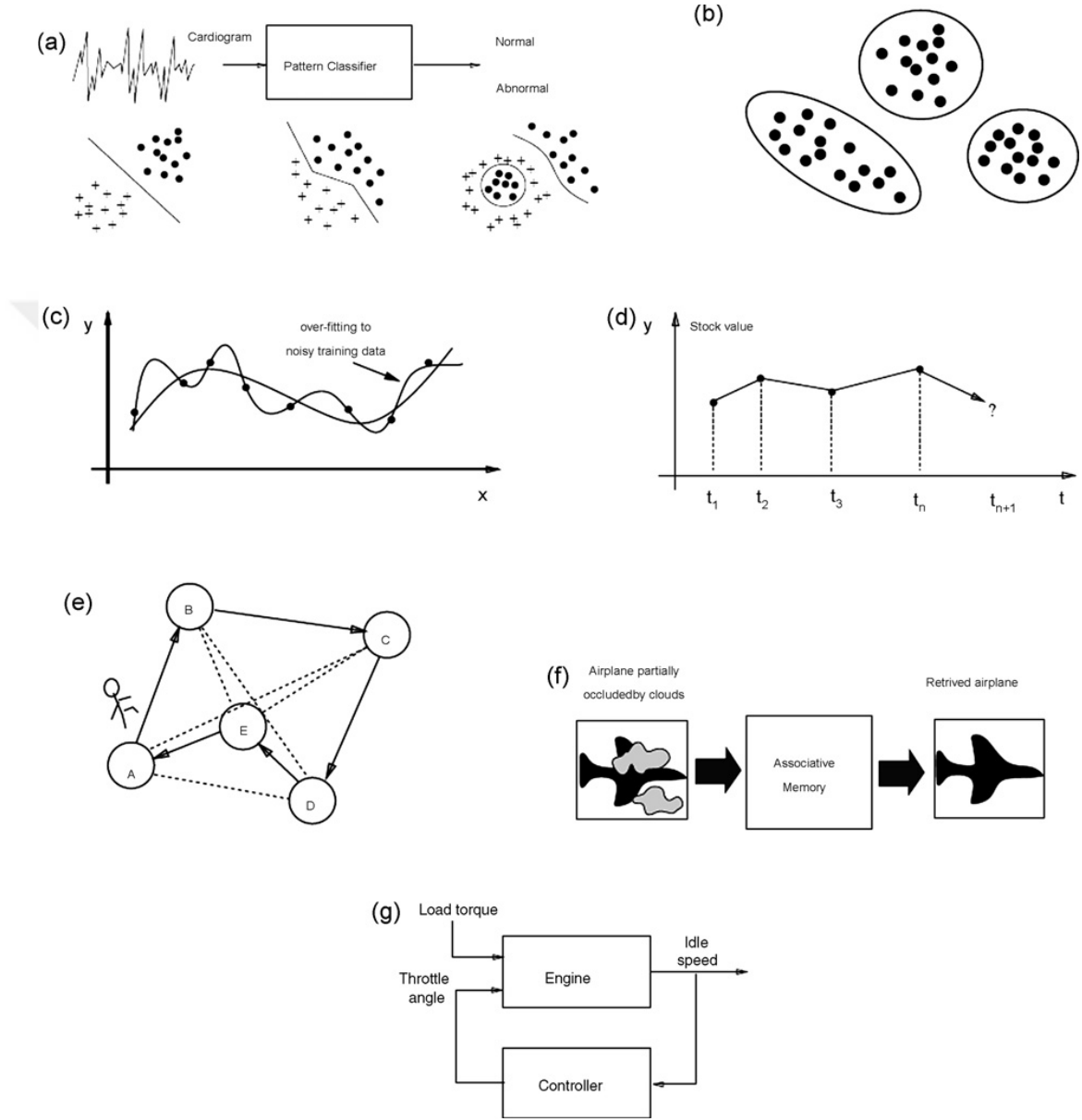


Şekil 2.3 Erne Yapay Sinir Ağı Modeli

YSA çözümün bilinmediği ya da ifade etmesinin zor olduğu veri ağırlıklı problemlerde kullanılır (Zhang & Stanley, 1997). Yapısı ve doğrusal olmayan hesaplama biçimi sayesinde veriyi paralel olarak işleyebilir ve veride olabilecek hatalardan az etkilenir. Hatalı veri setlerinde genelleme yaparak sonuca ulaşabilir. YSA kapalı bir sistem olduğu için çözüm adımları gözlemlenemez. Çözüm yolunun önemli olduğu problemlerde kullanışsızdır. Ayrıca YSA bir problemi çözmezse neden çözemediğini öğrenmemiz imkânsıza yakındır.

YSA 7 tipteki problemi çözebilir (Şekil 2.4) (Jain, Mao & Mohiuddin, 1996) Bunlar; sınıflandırma, kümeleme, fonksiyon seçimi, tahmin, optimizasyon, içerik çağırma ve işlem kontrolüdür. Sınıflandırma bir girdiyi daha önceden belirlenmiş sınıflardan birine atama işlemidir. Örnek olarak uydu görüntülerinden yeryüzü şekli sınıflandırma (Santiago Barros & Rodrigues, 1994) ya da kötü koku sınıflandırma (Onkal-Engin, Demir & Engin, 2005) verilebilir. Kümeleme denetimsiz sınıflandırma işlemidir. Girdiler ortak özelliklerine göre gruplara ayrılırlar. Örnek olarak akarsuların ekolojik özelliklerine göre sınıflandırması (Vellido, Marti, Comas, Rodriguez-Roda & Sabater, 2007) verilebilir. Fonksiyon seçimi ya da regresyon, eğitim verisi üzerinden bir

fonksiyon elde edilmesidir. Örnek olarak ozon yoğunluğunun hesaplanması (Sousa, Martins, Alvim-Ferraz & Pereira, 2007) ya da yeraltı sularındaki nitrat dağılımının hesaplanması (Almasri & Kaluarachchi, 2005) verilebilir.



Şekil 2.4 Yapay Sinir Ağları ile Çözülebilecek Problemler

2.5. Reinforcement Learning (RL, Pekiştirmeli Öğrenme)

RL'de öğrenme işlemi öğretici sistem ile ortam arasında bir etkileşim oluşturulması sayesinde gerçekleşir. Yani eğitim sırasında sistem deneme yanılma yolu ile öğrenir (Sutton & Barto, 2014). Öncelikle eğitim verisi üzerinde deneme yapar sonra sonuçlarının doğru olup olmadığına bakar. Sisteme doğru çözümler verilmez sadece bulunan sonucun doğru olup olmadığı söylenir. Bu yöntem 3 temel parçadan oluşur. Bunlar; Ortam, pekiştirme fonksiyonu ve değer fonksiyonudur. Ortam dinamik bir değişkendir ve zamana göre sürekli değişiklik gösterir. Herhangi bir zamandaki ortamın bir durumu için seçilen bir aksiyonun sonucunda pekiştirme yani ödül ya da ceza sinyalleri geri döndürülür (Kaelbling, Littman & Moore, 1996).

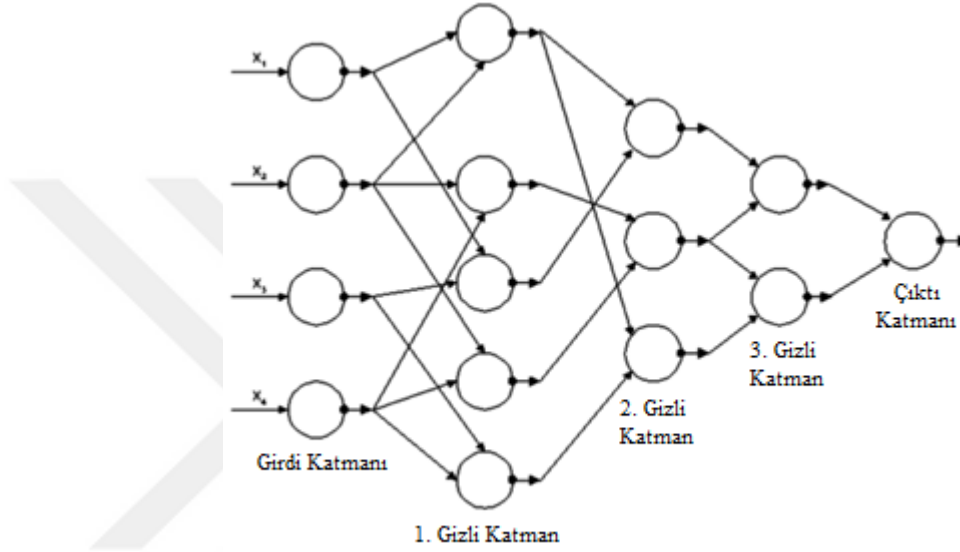
RL'nin kullanım alanları genelde robotik çalışmalar ya da dijital oyunlardır (Kaelbling, Littman & Moore, 1996). RL sürekli olarak davranış değiştirebilme özelliğine sahip olduğu için bu alanlarda etkili çalışır. RL kullanan sistemler sürekli verim ve başarı artışına sahip olurlar. RL'nin tek başına kullanım alanı sınırlıdır ancak başka yapay zekâ yöntemleri ile birleştirildiğinde pek çok alanda yüksek başarı elde eder. Örnek olarak insanların ya da hayvanların bireysel davranışlarının modellenmesi (Tsoularis & Wallace, 2005), arıların dolaşma düzenlerinin modellenmesi (Niv, Joel, Meilijson & Ruppın, 2002) gibi. RL sadece simülasyonlarda ya da laboratuvar ortamında çalışmaya uygundur. RL ile eğitilmiş bir sistem gerçek hayatta etkili çalışmaz (Abbeel, Quigley & Ng, 2006).

2.6. Bilgisayarlı Görü (Computer Vision)

Makinelerin, bilim ve teknolojiyi kullanarak insan görüşü ve gözlemlemesini taklit etmesine bilgisayarlı görü denir. Bilgisayarlı görü, makinelere resim ve videoları anlama, tanıma ve işleme gibi yetenekler kazandırır. İhtiyaca ve kullanım alanına göre insan görüşünü taklit etme, istatistiksel veri çıkartma ya da geometrik problemleri çözme gibi kullanım alanları da vardır. Kendi kendine işleyen bir görsel algı yaratmak için algoritmik temeller kullanılır. Dijital imge ve videolardan nitelikli veri elde etmeyi amaçlayan disiplinler arası bir çalışma alanıdır.

2.7. Derin Öğrenme

Derin öğrenme makine öğrenmesinin bir alt dalıdır. Derin öğrenme algoritmalarının temelini oluşturan ilk çalışmayı Ivakhnenko ve Lapa 1965 yılında yapmıştır (Ivakhnenko & Lapa, 1966). Bu çalışmada sadece birbiri ile yakından alakalı katmanlar arasında bir ilişki vardır. Bu ilişkiler istatistiksel yöntemler aracılığı ile seçilmiştir (Şekil 2.5).



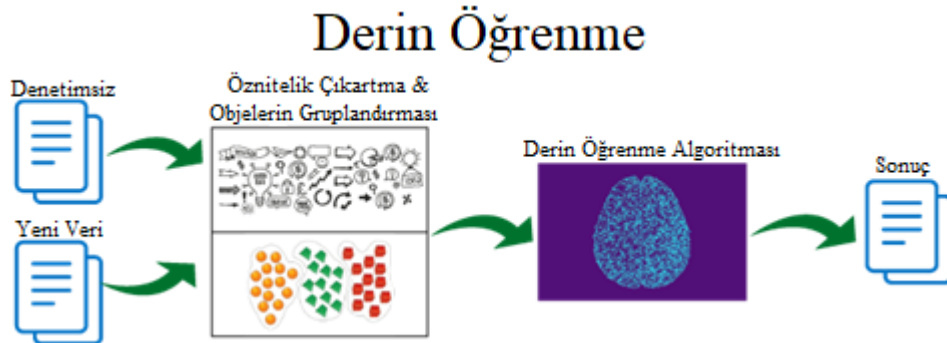
Şekil 2.5. İlk Derin Öğrenme Çalışması

1979 yılında Fukushima'nın yaptığı derin öğrenme mimarisine sahip neokognitron isimli çalışmada denetimsiz öğrenme kullanarak kendi kendini düzenleyebilen bir yapı geliştirilmiştir (Fukushima, 1980). 1989 yılına kadar derin öğrenme mimarilerinde geri bildirim mekanizması yoktur. Geri bildirim mekanizması olmadığı için derin öğrenme ağları öğrenmeyi tam olarak başaramamışlardır. 1989 yılında Yann LeCun el yazısı ya da baskı olarak yazılmış posta kodlarını okuyabilen bir uygulama geliştirmiştir. Bu uygulama geri bildirim özelliği olan bir derin öğrenme algoritması kullanmaktadır (LeCun, 1989). Başarılı sonuç vermesine rağmen eğitimi 3 gün süren bu uygulamanın pratikte kullanılamayacağı kararına varılmıştır. Aynı yıl içinde LeNet adı verilen başka derin öğrenme çalışması yapan Yann LeCun ve ekibi, bu yeni çalışmada el yazısı ya da baskı olarak yazılmış olan rakamları tanıyan bir sistem geliştirmişlerdir (LeCun, Boser & Denker, 1989). 1995 yılında Brendan Frey, Peter

Dayan ve Geoffrey Hinton tarafından yapılan bir çalışmada, bütün nöronları bir sonraki katman ile bağlı çok sayıda gizli katman içeren bir ağı eğitmenin mümkün olduğu ispatlanmıştır. (Hinton, Dayan, Frey & Neal, 1995). 1990'lı yıllarda yapay sinir ağları ve derin öğrenme algoritmaları teknolojinin yetersiz olmasından kaynaklanan hesaplama maliyetlerinin yüksekliği sebebiyle kullanılmamıştır. Bu algoritmalar yerine probleme özgü çalışan karar mekanizmaları ya da elle hazırlanmış ağırlıklara sahip daha basit özellikli ağ yapıları tercih edilmiştir.

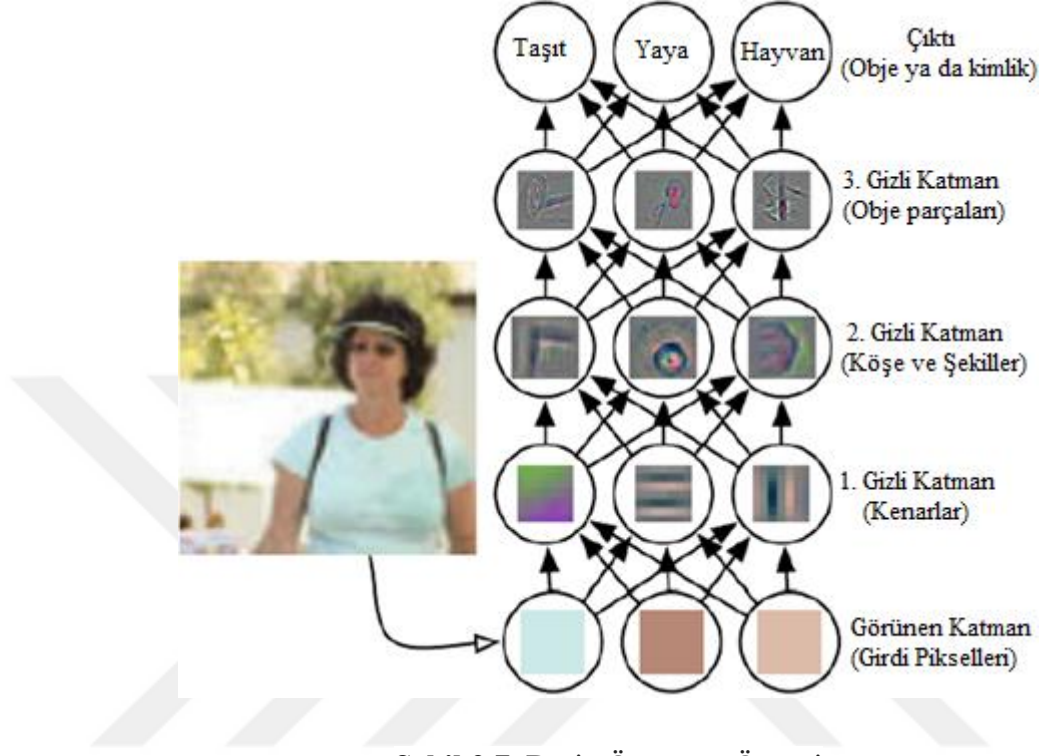
1990-2000 yılları arasında teknolojik gelişmelerin hızlanması ile bilgisayarların işlem gücü 1000 kat artmıştır. Bu artış beraberinde yapay sinir ağları alanı yeniden gündeme gelmiş ve 2000 yılında ilk defa derin öğrenme ifadesi kullanılarak çok katmanlı yapay sinir ağları ayrı bir araştırma konusu haline gelmiştir (Aizenberg, Aizenberg & Vandewalle, 2000).

Derin öğrenme özellikleri ifade ederken daha basit özellikler kullanmaktadır. Yani derin öğrenme bilgisayarın karmaşık kavramları daha basit kavramlardan tanımlamasını sağlamaktadır (Şekil 2.6). Her katman, kendinden önceki katmandaki çıktıyı girdi olarak alır (Deng & Yu, 2014). Algoritmalar denetimli veya denetimsiz olabilir. Derin öğrenme veri özelliklerinin karmaşık düzeyine göre katmanlara ayrılmış bir şekilde ifade edilerek öğrenilmeye dayanır. Karmaşıklığı yüksek olan özellikler, karmaşıklık seviyesi düşük olan özelliklerden türetilerek hiyerarşik bir yapı oluşturur. Bu yapıda her bir karmaşıklık seviyesine denk gelen bir soyutlama miktarı vardır. En düşük seviyede yani verinin girdiği noktada soyutlama yokken son gizli katmanda soyutlama en üst seviyededir (Bengio, 2009).



Şekil 2.6. Derin Öğrenme Adımları

Bilgisayarın ham veriden bilgi üretmesi çok zordur. Şekil 2.7’deki derin öğrenme modelinde bu durum resimdeki pikseller olarak ifade edilmiştir. Normal şartlarda bu piksellerden desenler oluşturulması ve sistemin o desenler üzerinden eğitilmesi gerekirdi.



Şekil 2.7. Derin Öğrenme Örneği

Deep learning modeli karmaşık desenleri basit desenlerden oluşturarak katmanlara bölmektedir. Şekil 2.7’de girdi verinin bize gözüken kısmıdır. Sonrasında gelen katmanlar gizli olarak etiketlenir çünkü bizim sisteme verdiğimiz veride bulunmayan değerler içerir. Bu gizli katmanlar girdi değerlerinden oluşan soyut veriler içerir. Bu gizli katmanları derin öğrenme modeli kendini sonuca ulaştıracak girdilerin ilişkileri arasından seçer. Şekil 2.7’de her katmanda oluşan sonuçlar görselleştirilmiştir. İlk katmanda parlaklık farkından kenarlar belirlenmiştir. İkinci katmanda ise kenarlardan köşeler ve şekiller oluşturulmuştur. Üçüncü katmada ise objeler ayrıştırılmıştır. Son olarak da sistem objelerden resmin neye ait olduğuna karar vererek sonucu oluşturmuştur (Goodfellow, 2016).

Derin öğrenme konusunda çalışma yapan birçok kurum, kuruluş ve şirket bulunmaktadır. Ayrıca bu alandaki başarılarından dolayı büyük yatırımlar alan başlangıç

şirketleri de olduğu görülmüştür. Bu alandaki çalışma grupları, bu ekiplerin önde gelen isimleri, kullandıkları kütüphaneler ve çalışma alanları Çizelge 2.1’de verilmiştir.

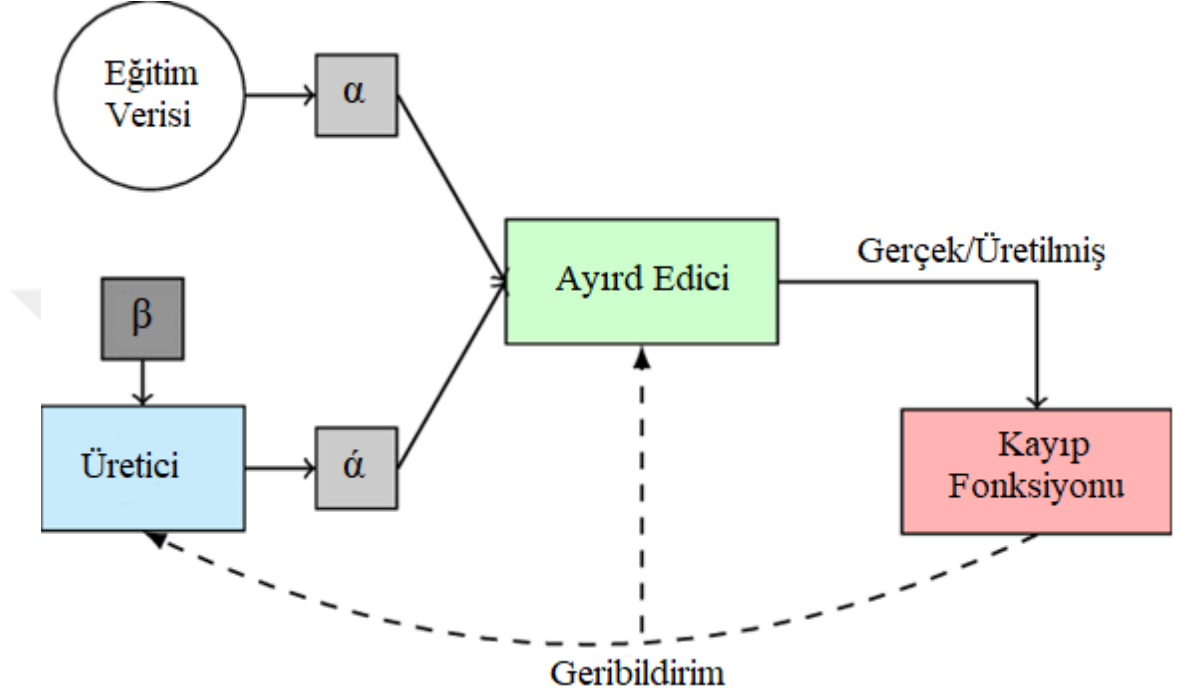
Çizelge 2.1. Derin Öğrenme Üzerinde Çalışan Araştırma Ekipleri

Araştırma Grupları ve Laboratuvarları	Ekip	Kütüphaneler	Çalışma Alanları
Toronto Üniv. - Makine Öğrenmesi Grubu	Geofrey Hinton Ruslan Salakhutdinov	Torch	DDİ, Gİ
Montréal Üniv. - MILA Lab.	Yoshua Bengio Pascal Vincent	Theano Pylearn2	DDİ, Gİ, Sİ
New York Üniversitesi - CILVR Lab.	Yann Lecun Rob Fergus	Theano	DDİ, BG, SR
Stanford Üniv. - SAIL ve SVL	Andrew Ng Fei-fei Li	TensorFlow Java	BG, DDİ, SR
Kaliforniya Üniv. - BAIR	Pieter Abbeel Trevor Darrell	Caffe	BG, DDİ, SR
Koç Üniv. - AI Lab.	Deniz Yuret	KNET	DDİ, Gİ
Google Research Google DeepMind	Jeff Dean Corinna Cortes	TensorFlow	Gİ, DDİ
Facebook - FAIR	Yann Lecun Larry Zitnick	Caffe2	DDİ, BG
Twitter - Cortex	Hugo Larochelle Clement Farabet	Torch	DDİ, Gİ, BG
Microsoft - DLTC	Xiaodong He Jianfeng Gao	Caffe	DDİ, BG
İsviçre U. B. Üniv. - IDSIA	Jurgen Schmidhuber Alessandro Antonucci	-	Gİ, SR
Baidu - Derin Öğrenme Enstitüsü (IDL)	Lin Yuanqing Wei Xu	-	DDİ, Gİ

2.8. Generative Adversarial Nets (GAN)

GAN derin öğrenme algoritmaları kullanan bir yapıdır. GAN yapısında 2 model bulunur. Birinci model Üretici (Generator: G), ikinci model ise ayırt edici (Discriminator: D) modelidir. Bu iki model eş zamanlı olarak birbirleri tarafından eğitilirler. D normal eğitim verisine ek olarak G’nin çıktılarını da eğitim verisi olarak alır. D’nin eğitim amacı gerçek verilerden G’nin ürettiği verileri ayırt etmektir. G’nin eğitim amacı ise D’nin hata oranını yükseltmektir. Şekil 2.8’de GAN’ın çalışma yapısı şematize edilmiştir. İdeal zamana ulaştığında D, G’den gelen verilerin tamamında hata yapar. Her iki model de

birbiri üzerinden eğitildiği için eğitim başlatıldıktan sonra bütün sistem sadece geribildirim ile ideal noktaya ulaşır. Herhangi ek bir destek mekanizmasına ihtiyaç duymaz (Goodfellow, vd., 2014).



Şekil 2.8. GAN çalışma yapısı

BÖLÜM 3

YARATICILIK ÜZERİNE YAPAY ZEKÂ UYGULAMALARI

Bu bölümde yapay zekâ yöntemleri kullanılarak yapılmış yaratıcılık niteliğine sahip olabilecek uygulamalar incelendi. Türk Dil Kurumuna (TDK) göre yaratıcılık şu şekilde tanımlanır; “sf. 1. Yaratma yeteneği olan, kreatif. 2. mec. Zekâ, düşünce ve hayal gücünden yararlanarak görülmeyen yeni bir şey ortaya koyan, yapan, kreatif”.

Bu bölümde tartışılan uygulamaların TDK'nın yaratıcılık tabirine ne kadar uygun olduğu araştırıldı. Uygulamalar sanatsal çalışma alanlarına göre ayrılarak incelendi.

3.1. Resim Üzerine Uygulamalar

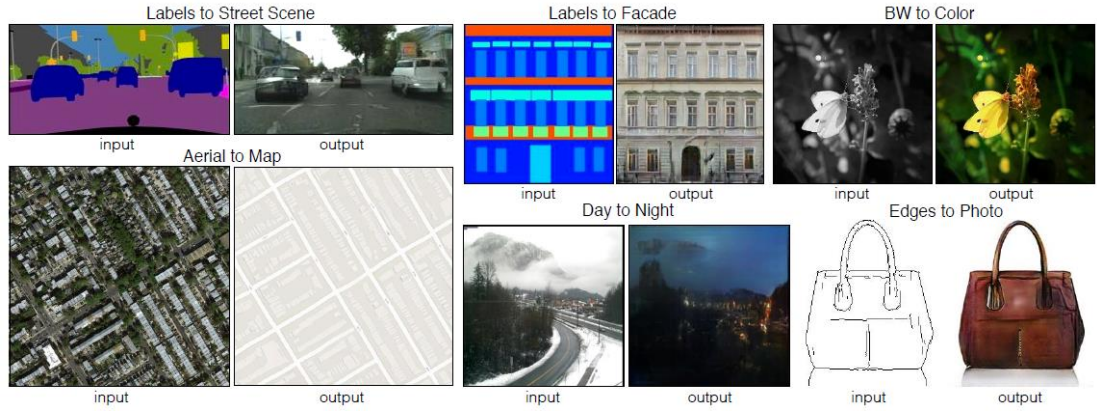
3.1.1. Pix2Pix

2018 yılında Phillip Isola, Jun-Yan Zhu, Tinghui Zhou ve Alexei A. Efros tarafından yapılan bir çalışmada otomatik imgeden imgeye dönüşüm yapılmıştır. Girdi olarak sadece resim şablonları verilerek sistemin bu şablonlardan objeler oluşturmaya imkân verilmiştir. Pix2pix adı verilen yazılımda verilen şablonlar karşılığında gerçekten kolay kolay ayırt edilemeyecek resimler elde edilmiştir. (Isola, Zhu, Zhou & Efros, 2018).

Bu sistem eğitim sırasında çıktıyı veriyi haritalayıp üzerine rastgele bir gürültü vektörü ekleyerek elde eder. Çıktının gerçek bir resimden ayırt edilmemesi esas alınarak uygunluk testi yapılır. Bu testi yapmak için benzer resimler ile çıktıları karşılaştıran bir test sistemi kullanılmıştır. Bu iki sistemin uyumlu çalışması sonucunda doğru bir eğitim elde edilmiş ve şablonlardan resme dönüşüm sağlanmıştır.

Pix2pix sistemi aşağıdaki dönüşümleri yapabilmektedir (Şekil 3.1):

- Etiketli Resim ↔ Fotoğraf
- Mimari Çizim → Fotoğraf
- Harita ↔ Kuşbakışı Fotoğraf
- Siyah Beyaz Fotoğraf → Renkli Fotoğraf
- Kenar → Fotoğraf,
- Skeç → Fotoğraf
- Gündüz → Gece
- Termal Fotoğraf → Renkli Fotoğraf



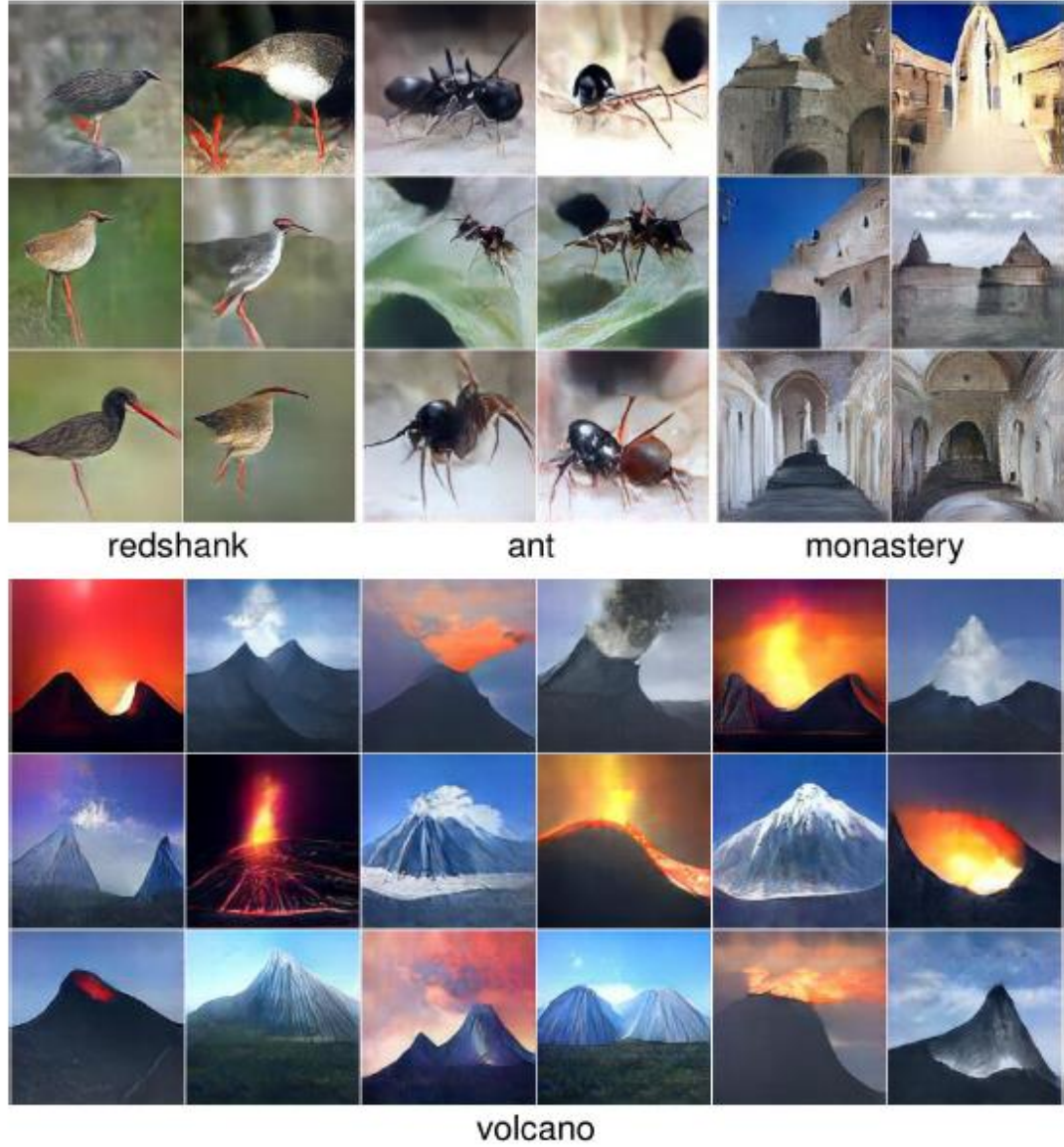
Şekil 3.1. Pix2Pix Dönüşümleri

3.1.2. Plug & Play

2017 yılında Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy ve Jason Yosinski tarafından yapılan Plug & Play adı verilen bir çalışmada verilen bir resim üzerinde bilgisayarın oynamalar yapmasına izin verilmiş ve gerçekten ayırt edilemeyecek yeni bir resim elde edilmeye çalışılmıştır. Bu çalışmada yüksek çözünürlüklü resimlerden aynı çözünürlüğe sahip yeni resimler elde etmek amaçlanmıştır. (Nguyen, Clune, Bengio, Dosovitskiy & Yosinski, 2017)

Bu çalışmada sistem girdi olarak verilen görüntüyü tarayarak obje çıkarımı yaptıktan sonra bulunan objeleri öğrenir. Sonra daha önceden sahip olduğu benzer görüntüler ile karşılaştırarak en benzer olanlarına göre girdiyi çok az değiştirir. Bu işlem belirli bir sayıda tekrar eder ve sonuç olarak yeni bir görüntü elde edilmiş olur (Şekil 3.2).

Normalde yüksek çözünürlüklü resimlerde bu işlem düzgün yapılamamaktadır. Ancak bu çalışmada orijinal resimdeki dominant objeler çok az değiştirildiği için çözünürlük yüksek olmasına rağmen resimlerde bozulma olmamaktadır.

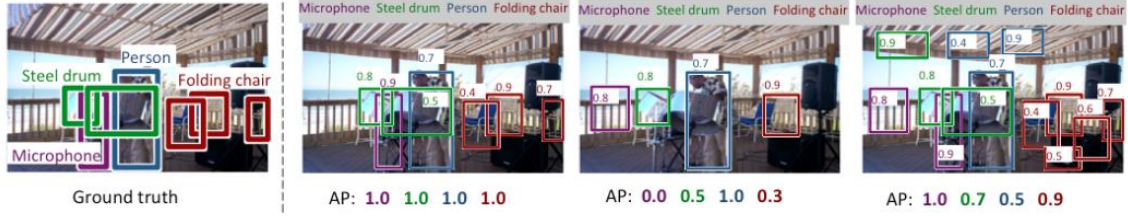


Şekil 3.2. Plug & Play Dönüşümleri

3.1.3. Metinsel Görüntü Tanımı Oluşturma

2015 yılında Andrej Karpathy ve Li Fei-Fei tarafından yapılan bir çalışmada verilen bir resmin sözel tasvirini veren bir sistem tasarlanmıştır. Bu konuda yapılan önceki çalışmalarda, sadece resimde bulunan objeler kategorilere göre etiketlenerek

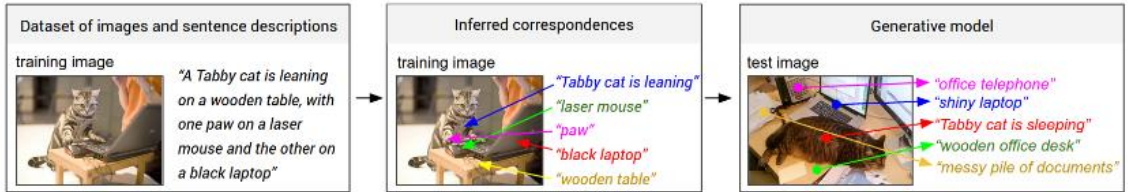
sınıflandırılmış ve Şekil 3.3'te görüldüğü gibi resmin üzerinde nerede hangi objenin olduğu etiketlenmiştir (Russakovsky vd., 2014).



Şekil 3.3. Resimlerde Obje Sınıflandırma

Önceki çalışmalarda herhangi bir eylem ya da durum belirtilmemiş resme herhangi bir yorum yapılmamıştır. Bazı çalışmalarda resim tasviri elde edilmiştir ancak bunlar belirli kalıplardaki cümlelerin içeriğe göre doldurulması şeklindedir. Bu çalışmalarda resimler bir iki cümle ile anlatılmak istenmiş ama karmaşık resimlerde bu durum hatalara yol açmıştır. Oysaki insan bir resme baktığında orada bulunan objelerden daha fazlasını görür. Bu yapılan çalışmada resmin insan gözünden görülüp yorumlanması hedeflenmiştir. Bunun için katı sınırlamalar, kalıplaşmış tahminler kullanılmadan resimde olanları mantık çerçevesi içerisinde doğal dile aktaran bir sistem geliştirilmiştir. (Karpathy & Fei-Fei, 2015)

Bu yöntemde resimdeki bölgeler ayrı ayrı yorumlanmış sonrasında ise bütün resmi anlatan bir metin elde edilmiştir. Resimler ve sözel tasvirleri derin öğrenme ağına eğitim verisi olarak verilmiş. Sonrasında eğitilen sistem karmaşık resimler üzerinde test edilmiştir (Şekil 3.4).



Şekil 3.4. Resimden Eylem ve Obje Çıkarsama

3.1.4. Fotoğraf Stilini Değiştirme

2017 yılında Fujun Luan, Sylvain Paris, Eli Shechtman & Kavita Bala tarafından yapılan çalışmada fotoğraf stillerini değiştiren bir derin öğrenme uygulaması yapılmıştır. Bu uygulama girdi olarak değiştirilmesi istenen bir fotoğraf ve artistik stil olarak benzetilmesi istenilen bir fotoğraf verilir. Çıktı olarak benzetilmek istenilen fotoğraf stiline sahip gerçek bir fotoğraf ile ayırt edilemeyecek bir fotoğraf verir. Şekil 3.5'te görüleceği gibi; gündüz/gece, mevsim, ışıklandırma ve artistik stil değişikliği yapabilmektedir (Luan, Paris, Shechtman & Bala, 2017).

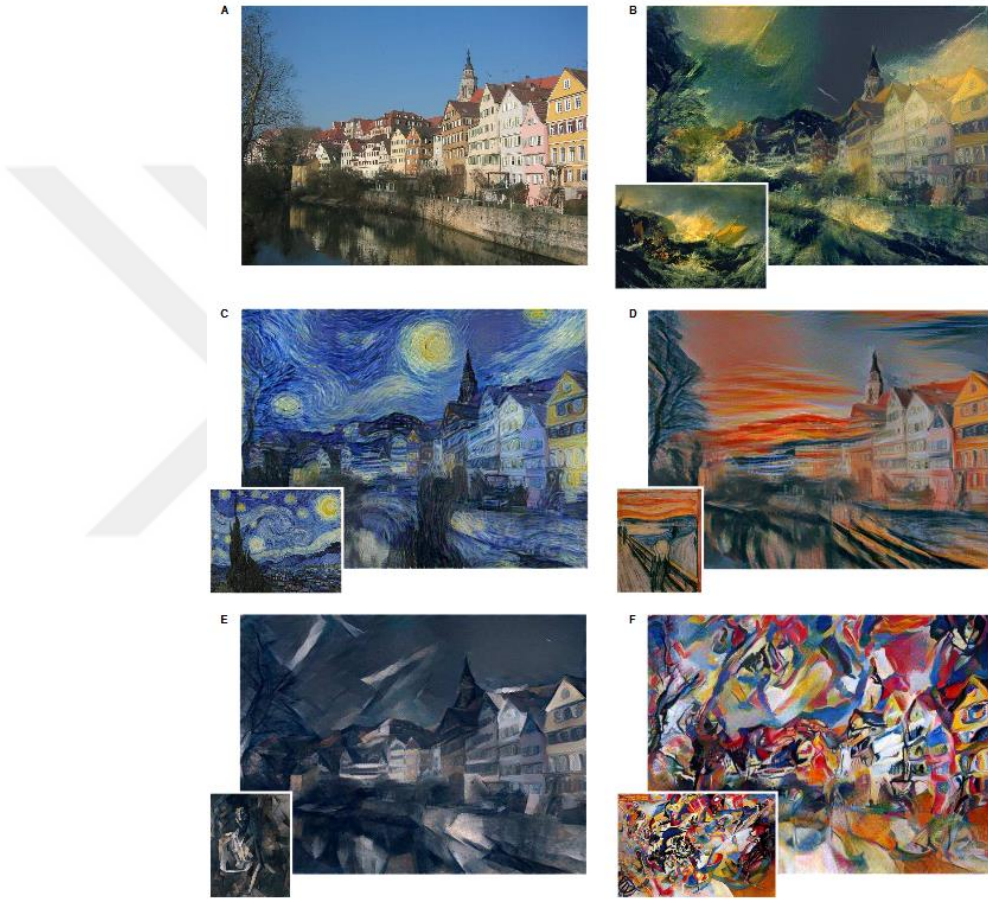


Şekil 3.5. Fotoğraf Stil Değişikliği Örneği

Eğitim ve test için 150 farklı fotoğraf kategorisine sahip DilatedNet Kütüphanesi (Chen, Papandreou, Kokkinos, Murphy & Yuille, 2016) kullanılmıştır. Önceki benzer çalışmalardan farkı daha gerçekçi sonuçlar vermesidir. Bunu sağlamak için 2 önemli değişiklik yapılmıştır. Birincisi bozulmaları önlemek için eğitim sırasında renk dönüşümleri sadece benzer renkler ile sınırlandırılmıştır. Yani renk uzayında ters düşen renkler birbirlerine dönüştürülmemiştir. İkincisi içerik uyumsuzluğunu engellemek için girdilerin aynı ya da benzer kategorilerde olduğu kontrol edilmiştir. Bu kontrolde veri setinde kullanılan 150 kategori temel alınmış ve girdiler bu kategorilere göre sınıflandırılmıştır. Bu özellik seçimlidir, istenildiğinde kapatılabilir. Referans resimler üzerinden benzetme işlemi yapıldığı için bu uygulama yaratıcı değil türeten bir uygulamadır.

3.1.5. Neural-Style

2016 yılında Leon A. Gatys, Alexander S. Ecker, Matthias Bethge tarafından resimlerde stil değişikliği yapan bir evrimsel yapay sinir ağı uygulaması geliştirilmiştir. Bu uygulamada girdi olarak değiştirilmesi istenen bir resim ve stili benzetilmesi istenen başka bir resim verilir. Çıktı olarak stil dönüşümü yapılmış bir resim elde edilir (Gatys, Ecker & Bethge, 2016).



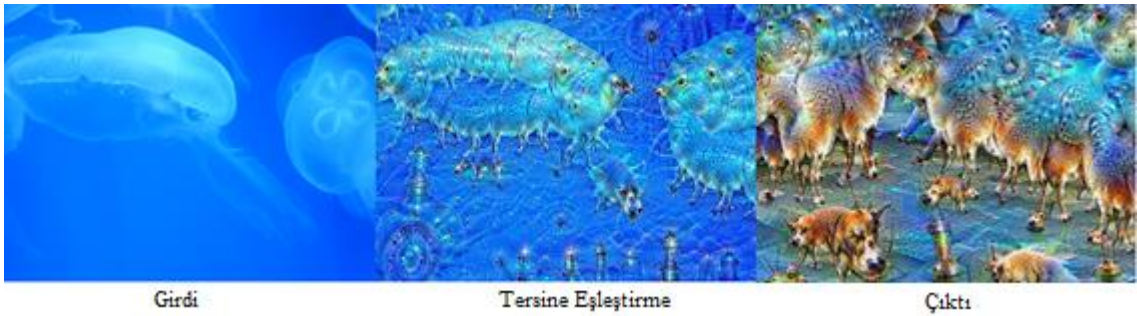
Şekil 3.6 Resim Stil Değişikliği Örneği

Girdi olarak verilen stil referans resmi veritabanında bulunan benzer resimler ile karşılaştırılır ve önceden etiketlenmiş bu resimlerden uygun stil ya da stiller belirlenir. Sonrasında değiştirilmesi istenilen resim içeriğine göre parçalara ayrılır ve her bir parça ya stil değişikliği uygulanır. Sonuç olarak değiştirilmesi istenilen resmin şablonuna uygun ama stili değişmiş bir resim elde edilir (Şekil 3.6).

Bu uygulamada resim stilleri derin öğrenme yöntemleri kullanılarak eğitilmiştir ancak resimlerin önceden etiketli oluşu ve stil değişikliği için CNN kullanılması bu uygulamayı yaratıcı değil türeten bir uygulama yapmaktadır.

3.1.6. DeepDream

DeepDream Google tarafından yapılmış bilgisayarlı görü ve derin öğrenme temelli bir uygulamadır. Alexander Mordvintsev tarafından gerçekleştirilen uygulama resimlerdeki özellikleri bulup geliştirerek bu özelliklerin başka resimler üzerinde aşırı işlenmesi sonucunda halüsinasyonlara benzeyen görünüm elde eder. DeepDream Inception filminden esinlenilerek ImageNet 2015 yarışması için tasarlanmıştır (DeepDream, 2019). Bilgisayarların hayal kurabilmesi fikri DeepDream'in internet ortamında yaygınlaşmasından sonra popüler hale gelmiştir. Google DeepDream uygulamasını açık kaynaklı hale getirip kaynak kodunu internette paylaştıktan sonra benzer çalışmalarda artış yaşanmıştır. Temelini resimlerde yüz tanıma, obje tanıma ya da sınıflandırma uygulamalarının tersten çalışmasına dayanır. Eğitildikten sonra verilen bir resim üzerinde oynama yaparak onu daha önceden öğrendiği başka objelere ya da yüzlere benzetmeye çalışır. Sonuç olarak saykodelik ve sürreal resimler elde eder (Şekil 3.7).



Şekil 3.7 DeepDream Örneği

3.2. Müzik Üzerine Uygulamalar

3.2.1. Klasik Müzik Kompozisyonu Yazma

2017 yılında Daniel D. Johnson derin sinir ağlarını kullanarak bilgisayarın yeni klasik piyano müzik kompozisyonları yazmasını sağlamıştır (Johnson, 2017). Bu uygulamayı oluştururken şu beş özelliğin olmasına dikkat etmiştir:

1. Sistemin zaman algısının olması gerekir. Çünkü şarkıların belirli bir uzunluğu vardır. Çok kısa ya da çok uzun olmamalıdır.
2. Sistemin stabil olması gerekir. Çünkü zaman içinde değişen bir sistemden her zaman aynı tarzda ve kalitede müzik elde edilemez.
3. Sistemin nota bağımsız olması gerekir. Çünkü parçalar kendi içlerinde iniş çıkışlara sahiptirler. Belirli tonlara odaklanmış sistemlerde hep aynı tonda parçalar çıkar.
4. Birden fazla notanın aynı anda çalınabilmesi ve bunun aynı akorttan çalınması gerekir. Çünkü tek bir piyanodan çalınan bir parçada aynı anda birçok nota çalınabilir ama piyanonun akordu değişmez.
5. Aynı notanın tekrar tekrar çalması ya da sürekli çalınması gerekir. Çünkü notaya tekrar tekrar basma ile notaya basılı tutma farklı ses oluşturur. Sistemin bunun farkında olması gerekir.

Bu sistem eğitirken var olan piyano şarkılarını 8 birimlik parçalara ayırarak her bir eğitim döngüsünde bu parçalardan rastgele 10 tanesini sisteme eğitmiştir. Eğitim sırasında aşırı öğrenmeyi engellemek için seyreltme yöntemi kullanılmıştır. Seyreltme yapmak için bazı şarkıların parçaları rastgele olarak gizlenmiştir. Şekil 3.8’de bu sistemin yazdığı bir parça bulunmaktadır.



Şekil 3.8. Bilgisayar Tarafından Yazılmış Bir Parça

3.2.2. Vocaloid

2004 yılında Yamaha Corporation tarafından piyasaya sürülen Vocaloid uygulaması bilgisayarların şarkı söylemesine imkân tanıdı (Vocaloid, 2020). İlk versiyonunda girdi olarak alınan şarkı sözlerini ve melodiyi sanal karakterlere söyleten bu uygulama Japonya’da kısa sürede popüler oldu. Şarkı söyleme sentezi (Singing Synthesis) adı verilen bu teknolojiye kullanıcı yazı olarak şarkı sözlerini ve nota olarak melodiyi uygulamaya girer. Sonrasında uygulama bulut ortamında bulunan ses ve şarkıcı kütüphanesinden şarkıya uygun bir ses tonu ve söyleme tarzı oluşturur. Bu oluşturulan tarz istendiğinde kaydedilip farklı parçalarda da kullanılabilir. Böylece sanal bir şarkıcı oluşturulmuş olur (Vocaloid, 2020).



Şekil 3.9 Hatsune Miku Canlı Konseri.

Günümüzde dünyanın en popüler sanal şarkıcısı Hatsune Miku, Vocaloid ortamında oluşturulmuş bir karakterdir. 2007 yılında Crypton Future Media, Inc. tarafından yaratılan karakterin bugün 10 adet şarkısı bulunmaktadır. YouTube ortamında 870 bin takipçiye sahip sanal karakter 90.000.000\$ ticari değere sahiptir. 2011 yılında çıkan World is Mine adlı şarkısı çıktığı ilk haftada iTunes global satış sıralamasında 7. sırayı almıştır. Japonya’da sanal ortam ile yaratılan sahneye çıkıp ilk konserini vermiştir (Şekil 3.9). Günümüzde dünya çapında canlı konserler vermektedir (Hatsune_Miku, 2015). Ses ve stil veritabanı üzerinden şarkılara uygun bir stil çıkartılması yaratıcı kabul

edilebilir. Ancak şarkıların ve melodilerin hazır alınması şarkının piyasaya sunulmadan önce düzenlenmesi yaratıcılığı azaltan faktörlerdir. Bu yüzden bu uygulamaya yaratıcılığa destek veren bir uygulama denilebilir.

3.3. Yazı Üzerine Uygulamalar

3.3.1. 1 the Road

2017 yılında Ross Goodwin tarafından yapılan bir çalışmada, arabaya takılan kamera, GPS ve mikrofondan alınan New York'tan New Orleans'a kadar yapılan yolculuğa ait veriler bir derin öğrenme ağına canlı olarak kaydedilmiştir (Goodwin, 2017). Yolculuk sırasında bu sistemden canlı olarak gördüklerini anlatması istenmiştir. Goodwin, yolculuğun sonunda aracının gözünden yazılanları derleyerek "1 the Road by an Artificial Neural Network" isimli kitabı yayınlamıştır (Goodwin, 2018). Kitabın ilk sayfasından bir alıntı:

It was seven minutes to ten o'clock in the morning, and it was the only good thing that had happened.

Sistemin eğitimini Foursquare'den elde edilmiş yüzlerce yolculuk kitabını kullanarak yapmıştır.

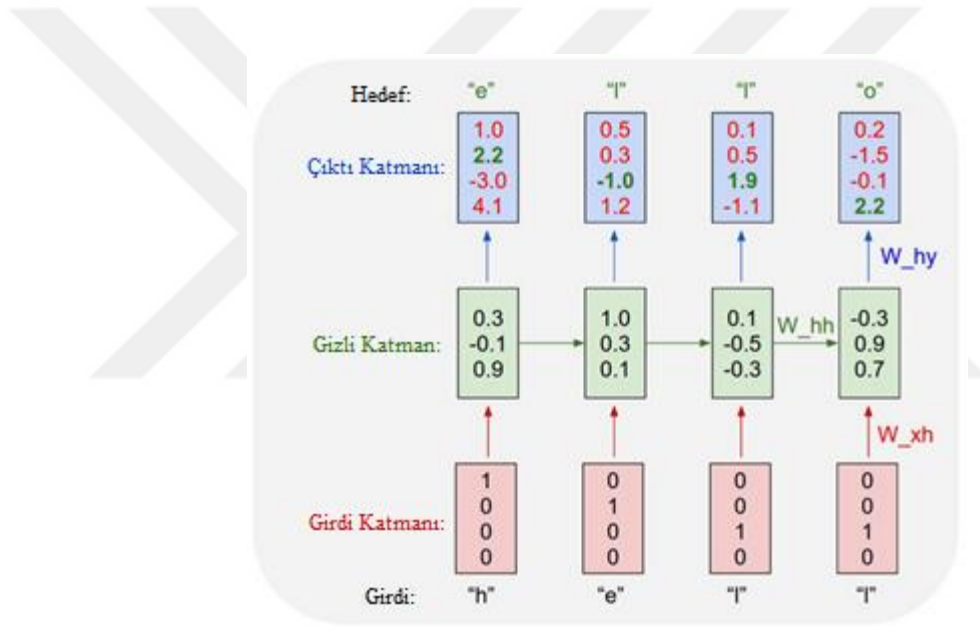
3.3.2. Deep Writing

2016 yılında Sung Kim ve Jack Jackson tarafından yapılan Deep Writing adı verilen bir çalışmada yazılardan cümle analizi yapılarak hikâyedeki karakterlerin yapacağı ya da söyleyebileceği şeyleri içeren yeni yazılar elde edilmiştir (Kim & Jackson, 2016). Önceki yazılardan cümle analizi yapan uygulama cümle yapısını art arda gelen kelimeleri tahmin ederek öğrenmektedir. Eğitim verisi olarak önceden yazılmış metinleri alıp o metinler üzerinde eğitildikten sonra yeni birçok cümle oluşturan bu uygulama anlam bütünlüğüne sahip bir hikâyeye ya da metin oluşturamamaktadır. Anlam bütünlüğü sadece cümle ile sınırlıdır. Örnek olarak Harry Potter serisi ile eğitilen bir sistemden şu cümleler oluşturulmuştur (2016):

"I'm afraid I've definitely been suspended from power, no chance – indeed?" said Snape. He put his head back behind them

and read groups as they crossed a corner and fluttered down onto their ink lamp, and picked up his spoon. The doorbell rang. It was a lot cleaner down in London.

Bu çalışma Andrej Karpathy'nin kelime temelli harf tahmini uygulaması üzerinden geliştirilmiştir (Karpathy, 2015). Bu uygulamada sisteme öğretilen kelimeler sayesinde yazı üzerinde gelecek bir sonraki harfin tahmin edilmesi sağlanmıştır. Örnek olarak "hell" yazıldığında bir sonraki gelecek harfin çok büyük olasılıkla 'o' olacağını tahmin eder (Şekil 3.10). Derin öğrenme teknolojisi ve tensorflow kütüphanesini kullanan bu uygulama İngilizce sözlüğü ile eğitilmiş ve bir sonraki gelecek harfi ağırlık tahmini ile bulmaktadır.



Şekil 3.10. Harf Tahmini Çalışma Prensibi

BÖLÜM 4

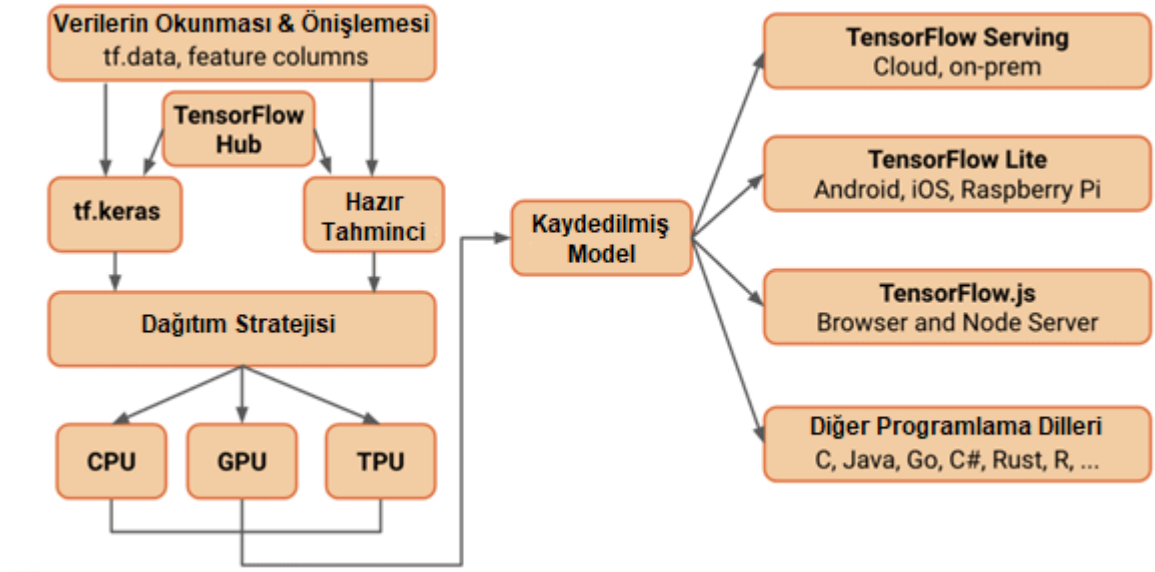
DENEYSEL ÇALIŞMA

Bu tez çalışmasında daha önceden geliştirilmiş olan Deep Writing uygulamasını yaratıcılık yönünden inceleyerek yeniden gerçekleştirildi. Yüzüklerin Efendisi Serisini İngilizce ve Türkçe dillerinde eğiterek uygulamanın dile olan bağımlılığını test ettik. Denemelerimizde Python programlama dili ve TensorFlow Kütüphanesi kullanıldı (Deep Writing, 2016).

4.1. TensorFlow Kütüphanesi

TensorFlow, Google tarafından geliştirilmiş Python programlama dilini kullanan açık kaynaklı bir derin öğrenme kütüphanesidir. 2015 yılında geliştirilmeye başlanan TensorFlow, 2017 yılında Apache 2.0 lisansı ile açık kaynaklı olarak piyasaya çıkartılmıştır. Birden fazla işlemci ya da ekran kartını aynı anda çalıştırabilme özelliği vardır. Linux, macOS, Windows, Android ve IOS ortamlarında kullanılabilir (TensorFlow, 2020).

2017 yılından itibaren TensorFlow Kütüphanesi Google tarafından desteklenen Keras Kütüphanesi ile birleştirilmiştir. Keras TensorFlow'un üzerinde çalışarak, daha hızlı çalışan, kullanıcı dostu, modüler ve genişletilebilir uygulamalar oluşturulmasına imkân sağlamıştır. Şekil 4.1'de Keras ve TensorFlow'un çalışma şeması verilmiştir (Keras, 2019).



Şekil 4.1. TensorFlow çalışma şeması

4.2. Materyal

Çalışmada eğitim verisi olarak J. R. R. Tolkien'in Yüzüklerin Efendisi Serisinin üç kitaplık tam metnini hem Türkçe hem de İngilizce olarak kullanıldı (Tolkien, 1954). Bu metni seçmemizin sebebi sistemin sınırlarını zorlamak ve gerçekten yaratıcı olup olamayacağını test etmektir.

İlk kitabı 1954 yılında yayımlanan Yüzüklerin Efendisi 150 milyonun üzerinde satışa sahip dünyada tüm zamanların en çok satan 2. romanıdır. Fantastik macera temalı bu seri türünün en iyisi kabul edilir ve çıktığı dönemden itibaren fantastik edebiyatın gelişmesini sağlamıştır (Yüzüklerin Efendisi, 2019)

Yüzüklerin Efendisi Serisi ağır bir üsluba sahiptir. Metin içinde sıra dışı ve özel isimlendirmelere sıklıkla rastlanır. Bu özellikleri Türkçe ve İngilizce metinleri için de geçerlidir. Örneğin; Miğfer Dibi - Helm's Deep, Minas Tirith, Bag End - Çıkın Çıkma, Fangorn - Ağaçsakal, Ayrıkvadi - Rivendell gibi. Bu farklı üslup eğitimin zorlaşmasına ve bu uygulamada yaratıcılığın olup olmadığı konusunda kesin bir yargıya varmamızı sağlayacaktır. İngilizce metin yaklaşık 470.000 sözcük ve 2.000.000 karakterden oluşmakta iken, Türkçe metin ise yaklaşık 350.000 sözcük ve 2.200.000 karakterden oluşmaktadır.

4.3. Eğitim

Her iki dildeki sistemi de eğitirken şu parametreler kullanıldı:

- Her bir gizli katman için 128 nöron,
- Toplamda 3 adet gizli katman,
- Tüm metin üzerinden 50 tam döngü,
- Öğrenme oranı: 0.002.

Bu uygulama eğitim sırasında metni kelime kelime tarayarak her bir cümlenin yapısını öğrenir. Cümle yapısını kelimelerin sıralanışını öğrenerek yani her bir kelimedenden sonra gelebilecek sonraki kelimeyi öğrenerek gerçekleştirir.

Modeli eğitmek için kullanılan Python kodu aşağıda verilmiştir;

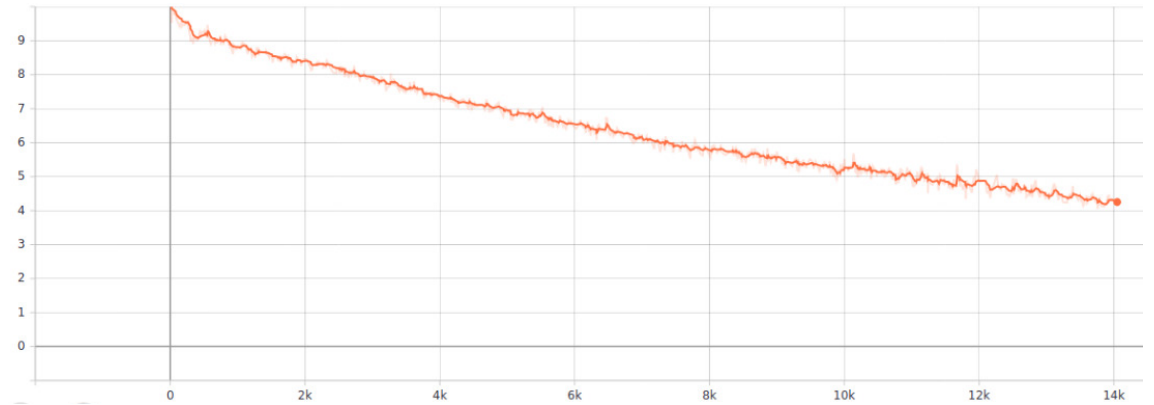
```
model = Model(args)
merged = tf.summary.merge_all()
train_writer = tf.summary.FileWriter(args.log_dir)
gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=args.gpu_mem)
with tf.Session(config=tf.ConfigProto(gpu_options=gpu_options)) as sess:
    train_writer.add_graph(sess.graph)
    tf.global_variables_initializer().run()
    saver = tf.train.Saver(tf.global_variables())
    for e in range(model.epoch_pointer.eval(), args.num_epochs):
        sess.run(tf.assign(model.lr, args.learning_rate *
                           (args.decay_rate ** e)))

        data_loader.reset_batch_pointer()
        state = sess.run(model.initial_state)
        speed = 0
        if args.init_from is None:
            assign_op = model.epoch_pointer.assign(e)
            sess.run(assign_op)
        if args.init_from is not None:
            data_loader.pointer = model.batch_pointer.eval()
            args.init_from = None
        for b in range(data_loader.pointer, data_loader.num_batches):
            start = time.time()
            x, y = data_loader.next_batch()
            feed = {model.input_data: x, model.targets: y,
                   model.initial_state: state, model.batch_time: speed}
            summary, train_loss, state, _, _ =
                sess.run([merged, model.cost, model.final_state,
                        model.train_op, model.inc_batch_pointer_op], feed)
            train_writer.add_summary(summary, e*data_loader.num_batches + b)
            speed = time.time() - start
    train_writer.close()
```

İngilizce metin ile eğitilen modelin 0.8 oranında yumuşatılmış kayıp/iterasyon grafiği Şekil 4.2’de verilmiştir. Derin öğrenmede öğrenmenin sınaması kayıp fonksiyonları ile gerçekleştirilir. Öğrenme değerleri farklılık gösterirse kayıp/iterasyon grafiğinde sıçramalar oluşur. Sistemler optimize edilerek her iterasyondaki kayıp bir miktar azaltılmıştır. Türkçe metin ile eğitilen modelin 0.8 oranında yumuşatılmış kayıp/iterasyon grafiği Şekil 4.3’de verilmiştir. Eğitim işlemi sonucunda yeni metinler elde etmek için kullanacağımız modeller kaydedilmiştir.



Şekil 4.2 İngilizce metin için kayıp/iterasyon grafiği



Şekil 4.3 Türkçe metin için kayıp/iterasyon grafiği

Şekil 4.2 ve 4.3’deki grafikler kıyaslandığında, İngilizce dilinde gerçekleşen eğitimde kaybın daha az olduğu görülür. Bu sonuçtan yola çıkılarak İngilizce dilinde yapılan eğitimin Türkçe dilinde yapılan eğitime göre daha başarılı sonuçlandığı söylenebilir.

4.4. Yeni Metin Elde Etme

Eğitim işlemi tamamlandıktan sonra yeni metin elde edebilmek için kaydettiğimiz modellerden belirli kelime uzunluklarına sahip örnekler oluşturuldu. Bu örnekler, eğitim sırasında öğrenilen cümle yapısına uyacak şekilde sıralanan kelimelerden oluşturuldu. Bu denemede karmaşık üsluba sahip bir metin öğretilmeye çalışıldığı için cümle yapılarında bazı bozukluklar olmaktadır.

Aşağıdaki Python kodu elde ettiğimiz modelden örnekleme yapmak için yürütülmüştür:

```
with open(os.path.join(args.save_dir, 'config.pkl'), 'rb') as f:
    saved_args = cPickle.load(f)
with open(os.path.join(args.save_dir, 'words_vocab.pkl'), 'rb') as f:
    words, vocab = cPickle.load(f)
model = Model(saved_args, True)
with tf.Session() as sess:
    tf.global_variables_initializer().run()
    saver = tf.train.Saver(tf.global_variables())
    ckpt = tf.train.get_checkpoint_state(args.save_dir)
    if ckpt and ckpt.model_checkpoint_path:
        saver.restore(sess, ckpt.model_checkpoint_path)
        for _ in range(args.count):
            print(model.sample(sess, words, vocab, args.n, args.prime,
                               args.sample, args.pick, args.width, args.quiet))
```

Elde edilen sonuçların çoğu aşağıdaki örneklerdeki gibi anlamsız kelime sıralamaları şeklinde bulunmuştur;

İNGİLİZCE:

```
Homesteads then was followed the course to the lean kitchen, and
passed across the Emyn Muil; and putting out the days and rode on
east and gathering into the din. The jored had been gone on, while
each is afraid. The cold sun was deserted; the face of Mordor was
far flames and terror sprang to through the aisle of later grey
trees lay in the Wood-elves' Street he saw himself enlarged, as
if things had gone down; terrible nearby made him at all.
```

TÜRKÇE:

Anlıyorum. Ama şimdi yine de çok ehven bir desteksindir. Hem ama ben bir süre önce bir kez başka önce bir şey duymadan, güvenlik içinde şaklattı. Sonra bir süre için kalkmazsa iyi daha dirseğimin üzerinden bir kaidesi olduğu bölgelerindeki daha uzun bir şey görülmüyordu. Ve bu konuda çok veledin bu çağrılara daha kabarık.

4.5. Değerlendirme

Sonuç olarak Deep Writing uygulamasının yaratıcı değil türetici bir uygulama olduğunu söyleyebiliriz. Öğretilen metinlerdeki kelimeler üzerinden tahmin yaparak yeni metinler elde etmiştir. Elde edilen metinlerin çoğunda cümle yapısı düzgün kurulmamıştır. Yapının doğru olmaması metinlerin saçma ve düzensiz olmasına sebep olmuştur. Yüzüklerin Efendisi'nin anlatım dilinin ağır olması ve çok sayıda özgün kelime içermesi uygulamanın doğru cümle yapısı oluşturmasını engelleyen sebeplerden biridir.

Türkçe ve İngilizce sonuçları kıyaslayacak olursak Türkçenin dil yapısı gereği İngilizceden daha karmaşık olması, uygulamanın kelime dizilişlerini öğrenmesine rağmen cümle yapısını doğru öğrenememesine sebep olmuştur. İngilizce dilinde cümle yapısı daha basit olduğu için uygulamanın kelimeleri tahmin ederek doğru cümle kurma ihtimali daha yüksektir.

Aşağıdaki örnekler elde edilen sonuçlar içerisinde en doğru yapıya sahip olanlardır;

İNGİLİZCE:

- 'If I suppose, he said looking holding to me through my fingers!' said Pippin.
- The threat was gone. The dark grip of Elrond tied years had belonged in swiftly growing but a whole black light stood upon him.
- 'Mithrandir and Galadriel,' said Sam. 'But I am afraid I have not seen the others?'

- 'But I have not seen the Ring-bearer' said Frodo.
- 'Well, do you do your best. Picking up all the business of a plant. By folk will be answered.' said Gandalf.
- 'I thought it was Lobelia. Then I don't know that he has been shot! He was a great horseman, and he seemed to see that he did not think that they had been stunned.'

TÜRKÇE:

- "O halde," diye sordu Gandalf. İmalarından ve uyarılarından hayalimde canlandırdığımdan ne kadar çok daha korkunç.
- Bunun üzerine Aragorn şöyle dedi: "Çoğunluk yat!"
- Çiftçi varisleri Gondor'a yakınlaştı. "Fakat çok şey göremiyorum" dedi Frodo.
- "Ama isterseniz," dedi Gandalf. O zaman onu bir süre bekleyin.
- "Hele demedim mi?" dedi Gandalf,

BÖLÜM 5

SONUÇLAR

Bu tez çalışmasında yaratıcı özelliği olabilecek yapay zekâ çalışmaları incelenmiştir. İncelenen çalışmaların çoğunda derin sinir ağları (derin öğrenme) yöntemi kullanılmıştır. Derin öğrenme yönteminin yapısı gereği yeni bir şey yaratması imkânsızdır. Ne yazık ki makineler yaratıcı olabilir mi sorusunun cevabı derin öğrenme değildir. Elde edilen sonuçlar her zaman yapay zekâ sisteminin daha önceden öğrendiği bilgilerden türetilmiştir. Ancak çalışmalarda kullanılan eğitim setleri ve sonuçlar kıyaslandığında, yapay zekâ sistemlerinin önceden öğrendiği bilgileri etkili bir şekilde kullanarak yeni sonuçlar elde ettikleri söylenebilir. Öğrenme kapasitelerinin genişliği sayesinde yeni olmasa dahi yeniliğe yakın, insanları kutunun dışında düşünmeye itecek, onlara ilham kaynağı olabilecek sonuçlar verdiklerini görebiliriz. Derin öğrenme sistemlerinin yaratıcı değil ama yaratıcılığı tetikleyen yardımcı sistemler oldukları söylenebilir.

İnsanlarda bulunan yaratıcılığı inceleyecek olursak yaratıcılığın kişiden kişiye farklılık gösterdiği görülür. Aynı melodiden esinlenen iki sanatçıdan biri kendine yeni bir şarkı yazarken öbürü yeni bir müzik türü yaratabilir. Günümüzdeki sanatçıları yaratıcılıklarının etkililiğine göre gruplandırıldığında sadece %5'inin çığır açan bir yaratıcılığa sahip olduğunu geriye kalan %95'lik gruba ait sanatçıların ise yaratıcı ama şu anda hali hazırda var olan trendlerin, kalıpların dışına çıkmadıkları söyleyebilir. 90'lı yıllarda yaşayan bilim insanlarından bazıları makinelerin insan görüşünü kesinlikle taklit edemeyeceğini savunmuşlardır. Ancak günümüzde bilgisayarlı görü ve derin öğrenme alanlarındaki gelişmeler sayesinde insanlardan daha iyi ve daha hızlı görüntü takibi yapan cihazlar kullanılmaktadır. Sonuç olarak teknolojinin gelişim hızını ve insanlardaki

yaratıcılığın bu kadar farklılık göstermesini göz önünde bulundurarak makinelerin yakın bir gelecekte insanların en düşük seviyesine denk bir yaratıcılığa sahip olabileceği söylenebilir.



KAYNAKLAR

- Aamodt A. & Plaza E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches, *AI Commun.* 7, 35–59.
- Abbeel P., Quigley M. & Ng A.Y., (2006). Using Inaccurate Models in Reinforcement Learning, *ICML '06: Proceedings of the 23rd international conference on Machine Learning P* 1–8.
- Aizenberg, I. N., Aizenberg, N. N., & Vandewalle, J. (2000). Multiple-Valued Threshold Logic and Multi-Valued Neurons. *Multi-Valued and Universal Binary Neurons, Boston, MA: Springer US*, 25–80.
- Almasri M.N. & Kaluarachchi J.J. (2005) . Modular neural networks to predict the nitrate distribution in ground water using the on-ground nitrogen loading and recharge data, *Environ. Modell. Softw.* 20, 851–871.
- Avesani P., Perini A. & Ricci F. (2000). Interactive case-based planning for forest fire management, *Appl. Intell.* 13, 41–57.
- Bengio, Y. (2009). Learning Deep Architectures for AI. *Trends Mach. Learn.* 2(1), 1–127.
- Chen L.-C., Papandreou G., Kokkinos I., Murphy K., & Yuille A. L. (2016). Deeplab: Semantic image segmentation with deeppconvolutional nets, atrous convolution, and fully connected, *arXiv:1606.00915*.
- DeepDream (2019, 12 Aralık) Wikipedia, The Free Encyclopedia içinde. 27 Aralık 2019 <https://en.wikipedia.org/wiki/DeepDream> adresinden erişildi.
- Deep Writing (2016). 5 Kasım 2019 tarihinde <https://github.com/hunkim/word-rnn-tensorflow> adresinden erişildi.

- Deng, L. & Yu, D. (2014). Deep Learning: Methods and Applications. *Trends® Signal Process.* 7 (3,4), 197–387.
- Dhar V. & Stein R., (1997). Intelligent Decision Support Methods. *The Science of Knowledge, Prentice Hall, New Jersey.*
- Engelmore R.S. & Feigenbaum E., (1993). Expert systems and artificial intelligence, in: Knowledge-Based Systems in Japan, <http://www.wtec.org/loyola/kb/>.
- Fdez-Riverola F. & Corchado J.M., (2004). Improved CBR system for biological forecasting, *EOAI, Workshop 23, Binding Environmental Sciences and Artificial Intelligence, Valencia, Spain.*
- Fukushima, K. N. (1980). A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* 36(4), 193–202.
- Gatys L. A., Ecker A. S. & Bethge M., (2016). Image Style Transfer Using Convolutional Neural Networks, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1063-6919.
- Goodfellow I. J., Bengio, Y. & Courville, A. (2016). *Deep Learning*. Erişim <http://www.deeplearningbook.org/>
- Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A. & Bengio Y., (2014). Generative Adversarial Networks, *arXiv:1406.2661*.
- Goodwin R., (2017). Automatic narration from Brooklyn to New Orleans, <https://github.com/rossgoodwin/wordcar>
- Goodwin R., (2018). 1 the Road by an Artificial Neural Network. America.
- Hammerstrom D., (1993). Working with neural networks, *IEEE Spectrum July*, 46–53.
- Hatsune Miku, (2015). 8 Ocak 2020 tarihinde https://vocaloid.fandom.com/wiki/Hatsune_Miku adresinden erişildi.
- Hayes-Roth F. (1985). Rule-based systems, *Commun. ACM* 28, 921–932.

- He K., Zhang X., Ren S. & Sun J. (2015). Deep Residual Learning For Image Recognition, *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, 1–9.
- Hinton, G. E., Dayan, P., Frey, B. J. & Neal, R. M. (1995). The wake-sleep algorithm for unsupervised neural networks. *Science*. 268(5214), 1158–61.
- Isola P., Zhu J., Zhou T. & Efros A. A. (2018). Image-to-Image Translation with Conditional Adversarial Networks, *arXiv:1611.07004*.
- Ivakhnenko, A. G. & Lapa, V. G. (1966). *Cybernetic Predicting Devices*. Indiana: Lafayette.
- Jain A.K., Mao J. & Mohiuddin K. (1996). Artificial Neural Networks: a tutorial, *IEEE Comput.* 29, 31–44.
- Johnson D. D. (2017). Generating Polyphonic Music Using Tied Parallel Networks, *LNCS, volume 10198*.
- Kaelbling L., Littman M. & Moore A., (1996). Reinforcement learning: a survey, *J. Artif. Intell. Res.* 4, 237–285.
- Kalapanidas E. & Avouris N. (2001). Short-term air quality prediction using a case-based classifier, *Environ. Modell. Softw.* 16, 263–272.
- Karpathy A. & Fei-Fei L. (2015) Deep Visual-Semantic Alignments for Generating Image Descriptions, *arXiv:1412.2306*.
- Karpathy A. (21 Mart 2015). *Multi-layer Recurrent Neural Networks (LSTM, GRU, RNN) for character-level language models* 10 Aralık 2019 tarihinde <http://karpathy.github.io/2015/05/21/rnn-effectiveness/> adresinden erişildi.
- Keras (2019, 9 Aralık) Wikipedia, The Free Encyclopedia içinde. 11 Ocak 2020 <https://en.wikipedia.org/wiki/Keras> adresinden erişildi.
- Kim S. & Jackson J. (Apr 28, 2016). *Multi-layer Recurrent Neural Networks (LSTM, RNN) for word-level language* 10 Aralık 2019 tarihinde <https://github.com/hunkim/word-rnn-tensorflow> adresinden erişildi.

- Krizhevsky, A., Sutskever I. & Hinton G.E. (2012). Imagenet Classification With Deep Convolutional Neural Networks, *NIPS*, 1097–1105.
- LeCun, Y. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* 1(4), 541–551.
- LeCun, Y., Boser, B. & Denker, J S. (1989). Handwritten Digit Recognition with a Back-Propagation Network. *Neural Computation.* 1(4), 541-551.
- Lenat, D., Prakash, M., Shepherd, M. (1986). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks, *AI Magazine*, 6(4), 65-85.
- Luan F., Paris S., Shechtman E. & Bala K., (2017). Deep Photo Style Transfer, *arXiv:1703.07511*.
- Mahaman B.D., Passam H.C., Sideridis A.B. & Yialouris C.P. (2003). DIARES-IPM: a diagnostic advisory rule-based expert system for integrated pest management in Solanaceous crop systems, *Agric. Syst.* 76, 1119–1135.
- Makine Öğrenimi (2019, 6 Kasım) Wikipedia, The Free Encyclopedia içinde. 4 Aralık 2019 https://tr.wikipedia.org/wiki/Makine_ogrenimi adresinden erişildi.
- Ng K.C. & Abramson B. (1990). Uncertainty management in expert systems, *IEEE Intell. Syst. Appl.* 5, 29–47.
- Nguyen A., Clune J., Bengio Y., Dosovitskiy A. & Yosinski J. (2017) Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space, *arXiv:1612.00005*.
- Niv Y., Joel D., Meilijson I. & Ruppin E., (2002). Evolution of reinforcement learning in foraging bees: a simple explanation for risk averse behaviour, *Neurocomputing* 44–46, 951–956.
- Onkal-Engin G., Demir I. & Engin S.N. (2005). Determination of the relationship between sewage odour and BOD by neural networks, *Environ. Modell. Softw.* 20, 843–850.

- Rodvold D.M., McLeod D.G., Brandt J.M, Snow P.B. & Murphy G.P., (2001). Introduction to artificial neural networks: taking the lid off the black box, *Prostate* 46, 39–44.
- Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A. C. & Fei-Fei L. (2014). Imagenet large scale visual recognition challenge, *arXiv:1409.0575*.
- Salem A.B. & Katoua H.S. (2013). Case-Based Reasoning Approach for Intelligent Tutoring Systems, *Wseas Vouliagmeni, Athens, Greece*.
- San Pedro J., Burstein F. & Sharp A. (2005). A case-based fuzzy multicriteria decision support model for tropical cyclone forecasting, *Eur. J. Oper. Res.* 160, 308–324.
- Santiago Barros M.S. & Rodrigues V. (1994). Nonlinear aspects of data integration for land-cover classification in a neural network environment, *Adv. Space Res.* 14, 265–268.
- Sousa S.I.V., Martins F.G., Alvim-Ferraz M.C.M. & Pereira M.C., (2007) Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations, *Environ. Modell. Softw.* 22, 97–103.
- Sutton R. & Barto A., (2014). *Reinforcement Learning: An Introduction*, İngiltere: Londra.
- TensorFlow (2020, 9 Ocak) Wikipedia, The Free Encyclopedia içinde. 11 Ocak 2020 <https://en.wikipedia.org/wiki/TensorFlow> adresinden erişildi.
- The quest for AI creativity, 2019. 8 Ocak 2020 tarihinde <https://www.ibm.com/watson/advantage-reports/future-of-artificial-intelligence/ai-creativity.html> adresinden erişildi.
- Tolkien J. R. R. (1954). *Lord of the Rings*. İngiltere.
- Tsoularis A. & Wallace J., (2005). Reinforcement learning for a stochastic automaton modelling predation in stationary model-mimic environments, *Math. Biosci.* 195, 76–91.

- Uzun E. (n.d.) *Makine Öğrenmesi Metotları*. https://www.e-adys.com/makine_ogrenmesi/artificial-neural-network-yapay-sinir-agi/
- Watson I. & Marir F. (1994). Case-Based, Reasoning: a review, *Knowl. Eng. Rev.* 9, 327–354.
- Winston P.H. (1984). Artificial Intelligence, second ed., *Addisin-Wesley Publishing Co., MA*.
- Vellido A., Marti E., Comas J., Rodriguez-Roda I. & Sabater F. (2007). Exploring the ecological status of human altered streams through Generative Topographic Mapping, *Environ. Modell. Softw.* 22, 1053-1065.
- Vocaloid (2020). 8 Ocak 2020 tarihinde <https://www.vocaloid.com/en/> adresinden erişildi.
- Vocaloid (2020, 2 Ocak) Wikipedia, The Free Encyclopedia içinde. 8 Ocak 2020 tarihinde <https://en.wikipedia.org/wiki/Vocaloid> adresinden erişildi.
- Yüzüklerin Efendisi (2019, 30 Kasım) Wikipedia, The Free Encyclopedia içinde. 20 Aralık 2019 https://tr.wikipedia.org/wiki/Yüzüklerin_Efendisi adresinden erişildi.
- Zetian F., Feng X., Yun Z. & XiaoShuan Z. (2005). Pig-vet: a web-based expert system for pig disease diagnosis, *Expert Syst. Appl.* 29, 93–103.
- Zhang Q. & Stanley S.J. (1997). Forecasting raw-water quality parameters for the North Saskatchewan River by neural network modelling, *Water Res.* 31, 2340–2350.

ÖZGEÇMİŞ

Kişisel Bilgiler

Doğum Tarihi: 22.12.1990

Doğum Yeri: Edirne, Türkiye

Medeni Durum: Bekâr

Eğitim

2015 - 2017, İzmir Yüksek Teknoloji Üniversitesi, İzmir, *Bilgisayar Mühendisliği Yüksek Lisans Eğitimi (Tamamlanmadı)*

Başlıca Eğitimini Aldığım Dersler: Pattern Recognition, Digital Image Processing, Biometric Recognition

2009-2015, Ege Üniversitesi, İzmir, *Bilgisayar Mühendisliği Lisans Eğitimi*

Bitirme Projesi: Optik Karakter Tanıma Sistemi.

Başlıca Eğitimini Aldığım Dersler: Algorithms And Programming, Data Structures, Linear Algebra And Numerical Methods, Database Management, Object Oriented Programming, Object Oriented Analysis And Design, Simulation And Modelling, Computer Graphics, Cloud Computing, Test Driven Development, Parallel Programming.

2005-2009, Edirne Süleyman Demirel Fen Lisesi, Edirne, *Lise Eğitimi*

İş Deneyimi ve Projeler

05.2017 - ... , Kırklareli Üniversitesi Yazılım Mühendisliği Bölümü, Araştırma Görevlisi

06.2016-02.2017, Ek İş, Unreal Engine ve Google VR teknolojisi kullanılarak apartman ve daire sanal turu tasarımı, VR reklam ve tanıtımlarını hazırlama.

10.2015-05.2016, İYTÜ Bilgisayar Mühendisliği Bölümü, Yardımcı Araştırmacı

113E107 numaralı TÜBİTAK destekli Classification of Objects in Traffic Scenes using Omnidirectional and PTZ Cameras projesi

07.2014-09.2014, KAAN Makine, Yardımcı ERP Uzmanı



TEZ ÖĐRENCİSİNE AİT AKADEMİK MAKALELER

Baysal K., Özcan M. O., Özdüven F. F., Beynek B., (2018). Nesnelerin İnterneti Tabanlı Bir Sera Takip Sistemi, *Electronic Journal of Vocational Colleges-November* pg 49-56.

Ozcan, M. O., Doyuran, T., Beynek, B. (2017). A Survey on the Use of Microsoft Kinect for Physical Rehabilitation, *International Journal of Computer Science and Mobile Computing, Vol 6 Issue 8, August*, pg. 128-132.