

T.C.
MUNZUR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



MUNZUR
ÜNİVERSİTESİ
2008

DIŞ ORTAM GÖRÜNTÜLERİNDEKİ İNSAN HAREKETLERİNİN DERİN
ÖĞRENME KULLANARAK SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ
Özlem İLHAN

Anabilim Dalı: Elektrik ve Elektronik Mühendisliği

DANIŞMAN
Dr. Öğr. Üyesi Özal YILDIRIM
TUNCELİ – 2018

T.C.
MUNZUR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**DIŞ ORTAM GÖRÜNTÜLERİNDEKİ İNSAN HAREKETLERİNİN DERİN
ÖĞRENME KULLANARAK SINIFLANDIRILMASI**

YÜKSEK LİSANS TEZİ
Özlem İLHAN
(12877037)

Anabilim Dalı: Elektrik ve Elektronik Mühendisliği

DANIŞMAN
Dr. Öğr. Üyesi Özal YILDIRIM
TUNCELİ – 2018

T.C.
MUNZUR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DIŞ ORTAM GÖRÜNTÜLERİNDEKİ İNSAN HAREKETLERİNİN DERİN
ÖĞRENME KULLANARAK SINIFLANDIRILMASI

Özlem İLHAN
YÜKSEK LİSANS TEZİ
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Bu tez / / 2018 tarihinde aşağıdaki jüri üyeleri tarafından **oybirliği/oyçokluğu** ile kabul edilmiştir.

İmza:.....

Dr. Öğr.Üyesi Özal
YILDIRIM (Munzur
Üniversitesi)

DANIŞMAN

İmza:.....

Doç. Dr. Ayşegül UÇAR
(Fırat Üniversitesi)

ÜYE

İmza:.....

Dr. Öğr.Üyesi Bilgin
ZENGİN
(Munzur Üniversitesi)

ÜYE

Bu tez, Enstitümüz Elektrik ve Elektronik Mühendisliği Anabilim Dalı'nda hazırlanmıştır.

Doç. Dr. Numan YILDIRIM
Enstitü Müdürü
İmza ve Mühür

NOT: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı "Fikir ve Sanat Eserleri Kanunu"ndaki hükümlere tabidir.

ÖZET

Bu tez çalışmasında, dış ortam görüntülerinde yer alan kişilerin hareketlerinin otomatik sınıflandırılması için derin öğrenme yaklaşımları önerilmiştir. Öncelikle, elde edilen görüntü içerisindeki kişilerin tespiti sağlanmıştır. Bu amaçla, literatürde yaygın olarak kullanılan önceden eğitilmiş derin nesne tespit aracı olan YOLO kullanılmıştır. Dış ortam görüntülerinin elde edilmesinde Google Street View ortamı kullanılmıştır. Daha sonra tespit edilen kişiler için hareket sınıfları oluşturulmuştur. Bu hareket sınıfları; sağa yürüyen, sola yürüyen, ayakta duran ve oturan şeklindedir. Böylece dış ortam görüntülerinden tespit edilen kişiler için kapsamlı bir veri seti oluşturulmuştur. Sınıfları belirlenen verilerin otomatik olarak tanınması işlemi için bir konvolüsyonel sinir ağı (KSA) modeli tasarlanmıştır. Tasarlanan bu konvolüsyonel sinir ağı modeli ile kişilere ait hareketlerin otomatik tanınması sağlanmıştır. Eğitimi tamamlanan bu sınıflandırıcı, YOLO nesne tespit sistemi ile birlikte kullanılarak giriş görüntüsü içerisindeki kişilerin hareketlerinin otomati olarak tanınması sağlanmıştır. Tez kapsamında, veri tabanı üzerinde üç ve dört sınıflı veri setleri oluşturularak önerilen sistemin performans değerlendirmeleri yapılmıştır.

Tez çalışması sonucunda, dış ortamdan elde edilen görüntüde yer alan kişilerin hareketlerine yönelik bir tanıma sistemi sunulmuştur. Böylece sürücüsüz araçlar ve robotik gibi dış ortamda insanlarla etkileşim içerisinde bulunacak uygulamalarda kullanılacak etkili bir tanıma sistemi sunulmuştur.

Anahtar Kelimeler: Derin öğrenme, nesne tespiti, insan hareketlerinin algılanması.

ABSTRACT

Classification of Human Movements in Outdoor Enviroments Using Deep Learning

In this thesis, deep learning approaches are proposed for automatic classification of actions of people in outdoor enviroments. Firstly, the identification of the persons in the obtained image is provided. For this purpose, YOLO, a pre-trained deep object detection tool widely used in the literature, is used. The Google Street View system is used to obtain the outdoor images. Movement classes were then created for the people detected. These movement classes; walking to the right, walking to the left, standing and sitting. Thus, a comprehensive data set has been created for people identified from outdoor images. A convolutional neural network (CNN) model is designed for the automatic recognition of classes of identified data. With this CNN model, automatic recognition of the movements of the person is provided. This trained classifier is used in combination with the YOLO object detection system to ensure that the movement of the persons in the input image is automatically recognized. Within the scope of the thesis, three and four class data sets were created on the database and performance evaluations of the proposed system were made.

As a result of the thesis work, a recognition system for the movements of people in the image obtained from the outside is presented. This provides an efficient recognition system that can be used in applications that interact with people in external environments such as autonomous vehicles and robotics.

Key words: Deep learning, object detection, perception of human movements

TEŞEKKÜRLER

Bu tez çalışmasının planlanması, araştırılması ve yürütülmesi sırasında yardım ve desteğini esirgemeyen, bilgi ve tecrübelerinden yararlandığım, yönlendirmesi ve bilgilendirmesi ile tezime şekil veren, her sıkıntıda yanına rahatlıkla gidebileceğim değerli danışmanım Dr. Öğr. Üyesi Özal YILDIRIM' a sonsuz teşekkürlerimi sunarım.

Çalışma sırasında, yoğun çalışmalarını arasında benden yardımlarını esirgemeyen Sayın Vedat TÜMEN hocama sonsuz teşekkürlerimi sunarım.

Her zaman yanımda olan desteğini hiç esirgemeyen değerli eşim Bilal Velat İLHAN'a ve aileme teşekkürlerimi sunarım. Her zaman danışabildiğim çalışma arkadaşım Şeriban BALKASOĞLU' na teşekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖZET	I
ABSTRACT	II
TEŞEKKÜR	III
İÇİNDEKİLER	IV
ŞEKİLLER LİSTESİ	VI
TABLolar LİSTESİ	VII
KISALTMALAR LİSTESİ	VIII
1. GİRİŞ	1
2. MATERYAL ve METOTLAR	3
2.1. Yapay Zeka.....	3
2.1.1. Yapay Sinir Ağları.....	5
2.1.1.1. Yapay Sinir Ağlarının Yapısı ve Temel Elemanları.....	6
2.1.1.2. Yapay Sinir Ağı Modeli ve Çok Katmanlı Algılayıcı.....	8
2.1.1.3. İleri Beslemeli Yapay Sinir Ağı.....	10
2.1.1.4. Geri Beslemeli Yapay Sinir Ağı.....	12
2.1.1.5. Yapay Sinir Ağlarının Avantajları ve Dezavantajları.....	13
2.1.1.6. Yapay Sinir Ağlarının Kullanım Alanları.....	15
2.2. Bulanık Mantık.....	16
2.2.1. Bulanık Mantığın Genel Özellikleri.....	17
2.2.2. Bulanık Sinirsel Sistemler.....	18
2.2.3. Bulanık Mantığın Uygulama Alanları	18
2.2.4. Bulanık Mantığın Avantajları ve Dezavantajları.....	18
2.3. Destek Vektör Makineleri.....	20
2.3.1. Destek Vektör Makinelerinin Avantajları ve Dezavantajları.....	20
2.4. Derin Öğrenme.....	21
2.4.1. Derin Öğrenmenin Çalışma Prensipleri.....	23
2.4.2. Derin Sinir Ağları.....	23
2.4.3. Derin İnanç Ağları	24
2.4.4. Yığın Otomatik Kodlayıcılar.....	25
2.4.5. Derin Öğrenme Yazılım Araçları.....	28
2.4.5.1. Torch.....	28
2.4.5.2. Caffe.....	29
2.4.5.3. TensorFlow.....	30
2.4.5.4. MatConvNet.....	31
2.4.5.5. Deeplearning4j.....	32
2.4.5.6. NVIDIA cuDNN.....	33
2.4.5.7. Theano.....	34
2.4.5.8. YOLO.....	34
2.4.5.8.1. Nesne Tespiti.....	35

2.5. Konvolüsyonel Sinir Ağları.....	35
2.5.1. Konvolüsyon Katmanı.....	37
2.5.2. Havuzlama Katmanı	37
2.5.3. Konvolüsyonel Sinir Ağı Matematiği.....	38
3. BULGULAR.....	41
3.1. Dış Ortam Görüntülerinden İnsan Hareketi Tespiti.....	41
3.1.1. Görüntü Veri Setleri.....	42
3.1.2. Kişilerin Tespit Edilmesi.....	43
3.1.3. Konvolüsyonel Sinir Ağı Modeli.....	44
3.1.3.1. Önerilen KSA modeli.....	44
3.1.3.2. Vggnet Modeli.....	45
3.1.3.3. Lenet Modeli.....	46
3.1.4. Deneysel Çalışmalar.....	46
3.1.4.1. Deneysel Çalışma-1: 4 Sınıflı Çalışma.....	47
3.1.4.2. Deneysel Çalışma-2: 3 Sınıflı Çalışma.....	49
4. TARTIŞMA.....	53
5. SONUÇLAR.....	55
KAYNAKLAR	56
ÖZGEÇMİŞ.....	61

SEKİLLER LİSTESİ

Sayfa No

Şekil 2.1. Yapay sinir ağlarının yapısı.....	7
Şekil 2.2. Yapay sinir hücrelerinin yapısı.....	7
Şekil 2.3. Yapay sinir ağı katmanları.....	10
Şekil 2.4. İleri beslemeli yapay sinir ağı örneği.....	11
Şekil 2.5. Çok katmanlı perceptron.....	12
Şekil 2.6. Geri beslemeli yapay sinir ağı	13
Şekil 2.7. Bulanık mantık örneği.....	17
Şekil 2.8. Derin öğrenme ve geleneksel makine öğrenmesi başarı oranları	22
Şekil 2.9. Derin sinir ağı yapısı.....	23
Şekil 2.10. Derin inanç ağlarının yapısı.....	25
Şekil 2.11. Otomatik kodlayıcı gösterimi.....	26
Şekil 2.12. Gizli katmanın giriş katmanı olarak alınması ve tekrar oluşturulması Gösterimi.....	27
Şekil 2.13. İki katmanlı yığın otomatik kodlayıcı örneği.....	28
Şekil 2.14. Torch ekran görüntüsü.....	29
Şekil 2.15. Caffe ekran görüntüsü.....	30
Şekil 2.16. Tensorflow ekran görüntüsü.....	31
Şekil 2.17. MatConvNet ekran görüntüsü.....	32
Şekil 2.18. Deeplearning4j ekran görüntüsü.....	33
Şekil 2.19. Basit bir KSA yapısı örneği.....	36
Şekil 2.20. İki boyutlu konvolüsyon işlemi gösterimi.....	37
Şekil 2.21. Giriş matrisi üzerinde max Pooling işlemi ve görüntü üzerinde örnek.....	38
Şekil 2.22. 5 katmanlı KSA yapısı	38
Şekil 3.1. Geliştirilen sistemin çalışma adımları görüntüsü.....	41
Şekil 3.2. Google Street view uygulamasından elde edilen görüntüler.....	42
Şekil 3.3. Google Street view uygulama görüntüsü.....	42
Şekil 3.4. Google Street view uygulamasından alınan görüntülerin YOLO ile tespit edildikten sonra parçalanmış halleri.....	43
Şekil 3.5. Önerilen Derin KSA Modelin blok gösterimi.....	45
Şekil 3.6. Deneysel çalışma-1 için parçalanmış görüntü örnekleri	47
Şekil 3.7. Dört sınıf için insan hareket tespitinde elde edilen KSA eğitim grafiği (%80-%10- %10).....	48
Şekil 3.8. Deneysel çalışma-2 için parçalanmış görüntü örnekleri.....	50
Şekil 3.9. Üç sınıf için insan hareket tespitinde elde edilen KSA eğitim grafiği (%70-%15- %15).....	51

TABLolar LİSTESİ

	<u>Sayfa No</u>
Tablo 3.1. Önerilen Derin KSA modelinin kod gösterimi.....	45
Tablo 3.2. Çalışmada kullanılan sınıflar ve bu sınıflara ait örnek görüntüler.....	46
Tablo 3.3. Deneysel çalışma-1 tüm deneysel sonuçların karşılaştırılması.....	48
Tablo 3.4. KSA modelinin test verileri üzerinde elde ettiği doğruluk matrisi (%80-%10-%10).....	49
Tablo 3.5. Deneysel çalışma-2 tüm deneysel sonuçların karşılaştırılması	50
Tablo 3.6. KSA modelinin test verileri üzerinde elde ettiği doğruluk matrisi (%70-%15-%15).....	51
Tablo 3.7. Önerilen KSA modelinin tüm deneysel çalışmadaki performansı.....	52

KISALTMALAR LİSTESİ

- YSA** : Yapay Sinir Ağı
DSA : Derin Sinir Ağı
DİA : Derin İnanç Ağı
KSA : Konvolüsyonel Sinir Ağı
DVM : Destek Vektör Makinesi
ÇKA : Çok Katmanlı Algılayıcı
KBM : Kısıtlı Boltzmanın makinesi
TBK : Tam Bağlı Katman
GPU : Grafik İşlem Birimi
BYD : Berkeley Yazılım Dağıtımı
MİB : Merkezi İşlem Birimi
GİB : Grafik İşlem Birimi
UPA : Uygulama Programlama Arayüzü
ABS : Otomatik Fren Sistemi
ASC : Otomatik Vites Kontrolü

1. GİRİŞ

Derin öğrenme, makinelerin; ses, görüntü ve metin gibi büyük verileri algılama ve anlamasına yardımcı olan ve hızla büyüyen yapay zeka geliştirmede en popüler yaklaşımdır. Derin öğrenme, yapısı bakımından insan beyninin çalışma prensibinden esinlenilmiştir. Birçok sinir ağı katmanını kullanan bilgisayarlar günümüzde, karmaşık durumları en az hatayla insanlar kadar iyi görme, onları iyi öğrenme ve onlara tepki verme bakımından oldukça başarılıdır. Günümüzde bu teknolojilerin uygulamaları daha çok anlama konusunda belirli görevler üzerine yoğunlaşmış ve bu konu üzerine kayda değer başarımlar elde edilmiştir. Çok önemli araştırma kuruluşları ve bazı şirketler, araştırmalarında ve ürünlerinde derin öğrenmeyi kullanmak için Grafik İşlem Birimleri (GİB)'nden yararlanmaktadırlar. Yeni gelişmelerin ortaya konulması için bilim adamları ve araştırmacılar bu alanda sürekli olarak çaba göstermekte ve bu alan için yatırım yaparak yeni araştırma gruplarını kurmaktadır. Derin öğrenme özellikle son zamanlarda yapay zeka alanında çığır açmıştır. Makineler; görüntü, ses, metin gibi büyük verileri kullanıp sınıflandırma, nesne tanıma, ses tanıma ya da dil çevirileri gibi özelleşmiş görevleri derin öğrenme kullanarak olabilecek en az hatayla yerine getirmeyi amaçlamaktadır.

Derin Öğrenme (Deng ve Yu, 2014), yapay sinir ağlarından (YSA) oluşan, derin bir mimarisi olan, gizli katman sayılarının artırıldığı ve her bir katmanda probleme yönelik özniteliklerin öğrenildiği yöntemler bütünüdür. Her bir katmanda probleme ait farklı seviyede öznitelikler öğrenilmekte ve öğrenilmiş olan bu öznitelikler üst katman için bir girdi oluşturmaktadır. Böylece en alt katmandan başlayarak en üst katmana kadar basitten karmaşığa özniteliklerin öğrenildiği bir yapı oluşturulmuş olmaktadır. Derin Öğrenme, yeni dönemlerin en önemli olan konuları arasında yer almaktadır. Google (Le, 2012), Facebook (Taigman, 2014), Baidu (Hannun, 2015), Microsoft (Deng ve yu, 2014) vb. kurumlar derin öğrenme konusunda önemli çalışmalar yürütmektedir.

YSA'lar, yeni olan bir hesaplama metodu değildir. Hemen hemen 60 yıllık bir geçmişe sahip olup (Rosenblatt, 1958)-(Farley ve Clark, 1954) farklı sahalarda kullanılmakta olan ve bilinen bir yöntemdir. YSA'lar, 1990'lara kadar kendine yer bulmuş ve doğrusal olmayan sistemler üzerinde başarılı olamamıştır. Böylelikle 2000' lere kadar önemli derecede bir gelişme sağlanamamıştır.

Geoffrey Hinton ve arkadaşları (2006), konuşma tanıma sorunlarında YSA'ları kullanmayı başarmış ve beraberinde YSA'ları tekrardan bilim çevrelerinde gündem konusu haline getirmişlerdir.

Konvolüsyonel sinir ağları (KSA); belgeler içerisindeki sayılar ve karakterleri tanıma (Lecun ve ark., 1998), yüz tanıma (Lawrence ve ark., 1997)-(Fasel ve Beat, 2002), insan hareketlerinin algılanması (ji ve ark., 2003), video sınıflandırma (Karpathy ve ark., 2014), sahne etiketleme (Farabet ve ark., 2013), nesne tanıma (Krizhevsky ve ark., 2012)-(Jarrett ve ark., 2009)-(Girshick ve ark., 2014) konuşma tanıma (Abdel Hamid ve ark., 2013)-(Abdel Hamid ve ark., 2012)-(Deng ve ark., 2013)-(Deng ve ark., 2013) ve bilgisayar oyunları (Silver ve ark., 2016) gibi alanlarda başarıyla kullanılmaktadır. KSA mimarisinin nesne ve görüntü sınıflandırmadaki başarımının yanında bütün bir resim içerisindeki nesnelere tanımda da başarılı olduğu görülmüştür.

Bir çalışmada, çevre seslerinin sınıflandırılması için bir KSA modeli kullanılmıştır. Yaptıkları çalışmada, çevre seslerinin sınıflandırılmasındaki performansı artırmak için veri artırma yöntemlerinden yararlanmışlardır. Veri kümesinin miktarı, önerilen artırma yöntemleri vasıtasıyla artırıldığında önerilen modelin performansının önemli ölçüde arttığını ve büyük oranda doğruluk gösterdiğini belirtmişlerdir (Salamon ve ark., 2014)-(Mesaros ve ark., 2016).

Bir çalışmada (Abdel Hamid ve ark., 2013), TIMIT ve Microsoft veri kümeleri üzerinde KSA uygulamaları yapılmıştır. KSA eğitiminin etkilerini test etmek için TIMIT üzerinde bir dizi deneysel çalışma yapılmıştır. Burada, KSA için ön eğitimin kullanılmasıyla elde edilen gelişimin, Tekrarlı Sinir Ağları (TSA)' dan daha az olduğu gözlemlenmiştir. Çalışmada, KSA'nın konuşma tanımda derin sinir ağı (DSA)'dan daha iyi bir performans sergilediği belirtilmiştir.

Bu tez çalışmasında, derin öğrenme yöntemlerinden KSA'lar kullanarak dış ortam görüntülerindeki kişilerin eylemlerinin otomatik sınıflandırılması sağlanmıştır. Bu amaçla, Google Street View uygulama ortamından kişilerin yer aldığı karmaşık ortam görüntüleri kullanılmıştır. Bu görüntüler üzerinde öncelikle kişilerin tespit edilerek bölümlenmesi sağlanmıştır. Bu amaçla, nesne tanıma uygulamalarında son zamanlarda ön plana çıkan YOLO derin öğrenme aracı kullanılmıştır. Bölümlenen kişi görüntüleri farklı eylem sınıflarına ayrılarak bir veri seti oluşturulmuştur. Daha sonra kişi eylemlerini otomatik sınıflandırmak için 25 katmanlı bir KSA modeli tasarlanmıştır. KSA modeli üzerinde uygulamalar yapılarak sonuçlar sunulmuştur.

2. MATERYAL ve METOT

2.1. Yapay Zeka

Yapay zeka, kimilerine göre insan gibi davranan insanoğlunun yerini alan elektromekanik robotlar olarak görülmektedir. Fakat bu konuda bilgili olup bu alanla ilgili olan herkes, insan ve makineler arasındaki mutlak farklılığın olduğu bilincinde olurlar. Makinelerin var olan teknoloji ile hiçbir şekilde insanoğlunun yapabildiği gibi yaratıcı olma, duygu ve davranışları algılama, anlama ve bunları aktarabilme becerisine sahip olamayacağı öngörülmektedir. Bununla birlikte, bilgisayarların belli başlı fiziksel davranışları yapabilen robotlar gibi makinalara yön verme ve veri hesaplama gibi uzmanlık alanları ile ilgili insan gibi düşünme süreci için akıl yürüten sistemlere beyin olma yeteneğine sahip olması olasıdır. Böylece uygulamalarda başarılı olup bu alana ticari getirisi de sağlanmıştır (URL14, 2014). Bilgisayarlar insanoğlunun yapabildiği bazı basit fonksiyonları daha iyi ve erken yapabilirler; matematik hesaplamalarında, rakamları ve harfleri işlemede, basit kararlar vermede, farklı veri saklama ve erişim işlevlerini görmede başarılıdırlar. Bu tür basit uygulamalarda, bilgisayarlar oldukça verimlidir ve genellikle insanların göstermiş olduğu performanstan daha iyi bir performans gösterirler. Çünkü evlerde kullanılmakta olan bilgisayarlar dahi birkaç haneli iki tane sayının çarpma işlemini nano saniyeler içinde yapabilmektedir. Benzer şekilde, herhangi bir kitap içerisindeki belirli bir kelimedeki kaç adet olduğunu insan algılamasının çok üzerinde bir başarıyla bulmaktadır. Bilgisayarlar, bu açıdan insanların düşünerek yapabileceği işlemlerinin bazı yönlerini oldukça kolaylaştırır ve hızlandırır. Yapay zeka, teknolojinin sunmuş olduğu uygulamalar ile elle yapılan karmaşık işlemleri otomatik olarak gerçekleştirmemize olanak sağlar. Bununla birlikte, diğer bilgisayar tabanlı bilgi sistemleri ile yapay zeka teknolojileri birlikte çalıştırılarak bilgisayarların başarıları ve uygulama alanları hızlı bir şekilde artırılmaktadır. Günümüzde, yapay zeka alanında oldukça önemli gelişmeler sağlanmış olmakla birlikte, bazı alanlarda araştırma düzeyi gelişmemiş durumdadır. Yapay zeka alanı ile ilgili araştırmacılar her geçen gün yapay zekayı yeniden tanımlayarak katkı sağlayabilecek yenilikler ortaya koymaktadır. Kimileri bu gelişmelerden yola çıkarak yapay zekanın tanımlanmasının imkansız olduğunu söylemektedir. Yapay zeka konusunda yapılan çalışmaların her birisinin çözümden daha çok yeni problemler ortaya koyduğu bile

söylenbilir. MIT (Massachusetts Technical University) araştırma grubunda yapay zeka ile ilgili çalışmaların öncüsü olan Marvin Minsky Yapay zekanın “hareket eden bir ufuk” gibi olduğunu ifade etmektedir. Yapay zeka ile ilgili çalışmaların 1960’lardan beri gündemde olmasına rağmen yapay zeka ile ilgili uygulamaların çok büyük boyutta hesaplama gücüne ihtiyaç duymasın nedeniyle çoğu araştırmacılar yapay zeka alanında yenilikler ortaya koyma çabalarında başarısız olmuşlardır. Fakat günümüzde bilgisayar teknolojisinde artan gelişmelerden dolayı yapay zeka alanında geniş çapta araştırma yapabilmek düşük maliyetli ve güçlü bilgisayarlar sayesinde mümkün bir hal almıştır. Böylelikle, yapay zeka konusunun bir alt dalı olan uzman sistemler konusu üzerinde şimdiden önemli gelişmeler sağlanarak, iş hayatında karar verme süreçlerinde uzman sistemlerden çok fazla yararlanıldığı görülmektedir. Birkaç yıl içerisinde yapay zeka uygulamalarına olan ilginin arttığı görülmüştür. Bu alanla ilgili uğraşları olanlar için bu konu büyük ilgi uyandırmıştır. Yapay zeka, bir çok uzun süreli yayın kapak konusu olarak gösterilmiştir ve bu alanda özel ekler yayınlanmıştır. Bu nedenle çok sayıda yapay zeka ile ilgili kitaplar ortaya çıkmıştır. Yapay zeka ile ilgili çok fazla rapor düzenli olarak yayınlanmış ve yapay zeka konusunda dünya genelinde konferanslar düzenlenmiştir.

Yapay zeka, bilgisayarları akıllı yapma fikrine dayanır. Yapay zeka, geleneksel bilgisayarlardan arınıp bilgisayarların daha fazla faydalı hal almasını isteyenler ve zekanın doğasını daha iyi çözmek isteyenler aracılığıyla uygulanmaktadır. Yapay zeka, zeka olarak bilinen akıllı davranışların makinalar vasıtasıyla yapılmasıdır. Yapay zeka, insan aklının çalışma şeklini göstermek için uğraş gösteren bir teoridir. Zekanın doğasıyla ilgilenenlerin hedefi, zekanın taklidini yapmak değil sistemi zeki duruma getirmektir.

1884 yapay zeka konusunda en öneki dönüm noktası olmuştur. 1884’te, Charles Babbage isminde bir bilim insanı, zeki davranışlar göstermesini beklediği mekanik makineler üzerinde deneyler yapmak için çalışmıştır. İnsan gibi zeki olamayacağında hemfikir olunan bu makinalar üzerinde çalışmalar sürdürülerek, 1950’ye kadar gelinmiştir. Bu tarihte, Shannon ismindeki bir bilim insanı, bilgisayarın santraç oynayabileceğini ileri sürmüştür. Sonuç olarak tarih öncesi aadı altında adlandırılan dönem başındaki çalışmalar, 1960’lı yılların başına kadar hızlı denmeyecek bir hız ile sürdürülmüştür (URL13, 2017).

2.1.1. Yapay Sinir Ağları

YSA'lar, sinir sistemlerinden esinlenerek insan beyninin başlıca özelliği olan öğrenme işlevini yapan sistemleridir. Yapay sinir ağları, birbirine bağlı yapay sinir hücrelerinden oluşmaktadır. Her bir bağlantının kendine özel bir ağırlık değeri bulunmaktadır ve ağda bulunan bilgi bu ağırlık değerlerinde gizli durumda olup ağa yayılmaktadır. YSA'lar, geleneksel hesaplama metotlarından farklı olan bir hesaplama metodu önermektedir. Bu hesaplama metodu, ortama uyum sağlayan, belirsizlikler ile karar verebilen, hatalara karşı toleranslı olan, eksik bilgi ile çalışabilen ve hayatın hemen hemen her alanında başarılı uygulamaları olan bir metottur. Tasarlanacak olan ağ yapısının belirlenmesi, ağ parametrelerin seçimi, belli bir standartının olmaması, problemlerin sadece sayısal bilgiler ile temsiledilmesi, eğitimin ne şekilde sonlandırılacağına bilinmemesi ve ağın davranışlarını net olarak tanımlayamamasına rağmen YSA'lara duyulan ilgi git gide artmaktadır. Yapay sinir ağları, her şerden önce sınıflandırma, veri sıkıştırma, sinyal filtreleme ve optimizasyon çalışmalarında en güçlü yöntemler arasında yer almaktadır. Optik karakter taşıma, ses tanıma, optimum rota belirleme, parmak izi tanıma, görüntü tanıma, malzeme analizi vb. çoğu alanda günlük hayatta görülebilecek başarılı örneklerle karşılaşmak olasıdır. Yapay sinir ağlarının tarihsel gelişimini göz önüne aldığımızda dönüm noktasının 1970 olduğu görülmektedir. 1970'ten önce de çok fazla araştırma yapılmıştır. Fakat 1969 yılında XOR sorununa çözüm bulunamaması sebebiyle araştırmaların durma noktasına geldiği görülmektedir. Sonrasında ise az sayıda araştırmacının çalışmaları sürdürüp XOR problemine çözüm bulmaları ile birlikte yapay sinir ağlarına tekrardan artan bir ilgi görülmüştür. Bunu takiben 10 yıl içerisinde birbirinden farklı çok sayıda yeni modeller geliştirilmiştir. Bununla birlikte aladanki çalışmalar laboratuvar ortamından çıkarak günlük hayatımızda kullanılan teknolojiler haline gelmiştir. Bu alanla ilgili çalışmalar, hem yapay zeka hem de donanım teknolojilerindeki ilerlemeler tarafından destek görmüştür. Bütün bunlara dayanarak, günümüzde bundan sonra herkes bilgisayarlarında öğrenebileceğini kabul etmekte ve bu teknolojiden yararlanmaktadır(Öztemel, 2012).

2.1.1.1. Yapay Sinir Ağlarının Yapısı ve Temel Elemanları

Yapay sinir ağları, sinir sisteminden esinlenip geliştirilmiştir. Biyolojik sinir hücreleri sinapslar vasıtasıyla birbirleriyle bağ kurmaktadır. Bir sinir hücresi işlemiş olduğu bilgileri akson vasıtasıyla diğer hücelere göndermektedir. Buna benzer bir şekilde yapay sinir hücreleri dış ortamdan alınan bilgileri bir fonksiyon ile toplar ve bu toplamı aktivasyon fonksiyonundan geçirerek çıkış alır. Elde edilen bu çıkış ağ bağlantıları üzerinden diğer hücelere gönderir. Farklı toplama ve aktivasyon fonksiyonları bulunmaktadır. Yapay sinir ağlarının birbirlerine bağlanmasını sağlayan bağlantı değerlerine ağırlık değerleri denmektedir. Süreç elemanları birbirlerine paralel biçimde üç katman halinde bir araya gelip bir ağ kurmaktadır. Bunlar;

- Girdi katmanı,
- Ara katmanlar,
- Çıktı katmanı.

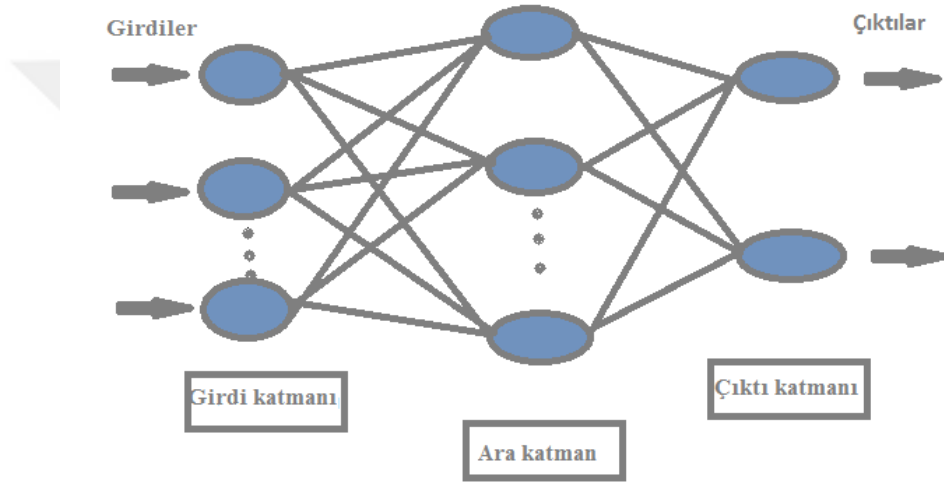
Bilgiler ağa ağın girişinde bulunan girdi katmanından verilmektedir. Bu bilgiler ara katmanda işlenerek oradan da çıktı katmanına gönderilmektedir. Bilgi işlemede ki amaç, ağın ağırlık değerleri kullanılarak ağa gelen bilgilerin çıktıya dönüştürülmesidir. Ağın girdileri için çıktıları doğru bir şekilde üretebilmesi bakımından ağırlık değerlerinin doğru olması gereklidir. Doğru ağırlıkların bulunması işlemine ağın eğitilmesi denmektedir. Ağırlık değerleri ilk etapta rastgele atanırlar. Daha sonra eğitim aşamasında, her bir örnek ağa gösterildiğinde ağırlıklar ağın öğrenme yaklaşımına göre değiştirilir. Daha sonra başka bir örnek ağa sunularak ağırlıklar yine değiştirilmekte ve en doğru değerleri bulmaya çalışılmaktadır. Ağ eğitim setindeki örneklerin tümü doğru çıktıları bulana kadar bu işlem sürekli tekrarlanır. Bu yapıldıktan sonra test setinde bulunan örnekler ağa gösterilir. Ağın test setindeki örneklere doğru cevaplar vermesiyle ağın eğitildiği kabul edilmektedir. Ağ için ağırlıklar belirlendikten sonra ağırlıkların hangi anlamları taşıdığı bilinmemektedir. Bu sebeple yapay sinir ağları için kara kutu tabiri yapılmaktadır. Ağırlıkların hangi anlama geldikleri bilinmemekle beraber ağın girdileri hakkındaki kararını bu ağırlıkları kullanarak vermesi, ağın zekasının bu ağırlıklarda gizlendiği söylenebilir. Ağın herhangi bir olayı öğrenmesi en uygun modeli seçmekle mümkündür. Bu güne kadar çok fazla yapay sinir ağı modeli gelişimi sağlanmıştır. Bir yapay sinir ağının modelini aşağıdaki bilgiler karakterize etmektedir.

- Ağdaki topolojisi,

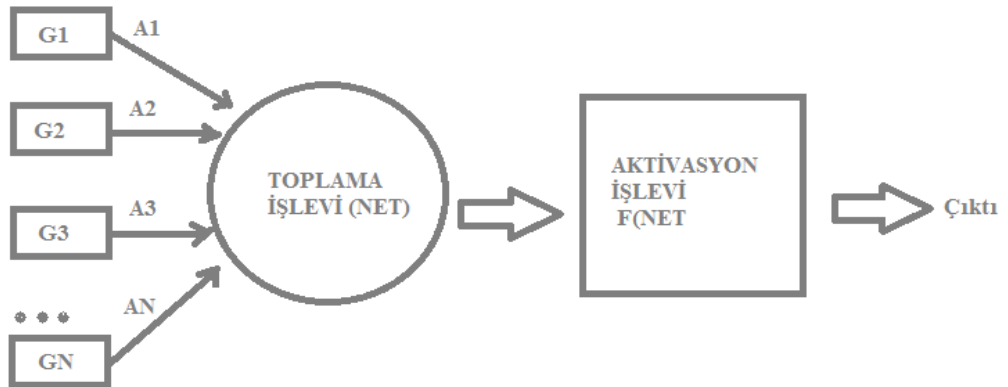
- Kullanılacak olan toplama fonksiyonu,
- Kullanılacak olan aktivasyon fonksiyonu,
- Öğrenmenin stratejisi,
- Öğrenmenin kuralı.

Geliştirilmiş olan modeller içinde en çok kullanılanları, LVO, ART ağları, tek ve çok katmanlı algılayıcılar, SOM, Elman ağı vb. ağlardır(URL15, 2017).

Şekil 2.1.'de YSA'lara ait bir örnek yapı gösterilmiştir. Şekil 2.2.'de ise yapay sinir hücrelerinin yapısı verilmiştir.



Şekil 2.1. Yapay sinir ağlarının yapısı (URL8, 2017)



Şekil 2.2. Yapay sinir hücrelerinin yapısı (URL8, 2017)

2.1.1.2. Yapay Sinir Ağı Modeli ve Çok Katmanlı Algılayıcı

Günümüzde en yaygın kullanılan yapay sinir ağı modeli çok katmanlı algılayıcı (ÇKA) ağlarıdır. Çok katmanlı algılayıcı ağlar, başta mühendislik olmakla birlikte bir çok alanda çözüm üretebilecek nitelikte ağlardır. Bu ağlar, XOR problemine çözüm bulabilmek için verilen uğraşlar neticesinde ortaya çıkmıştır. Bu ağlar girdi, ara ve çıktı olmak üzere 3 katmandan oluşur.

- **Girdi katmanı:** Bu katman dış ortam bilgilerini alır ve bu bilgilere herhangi bir işlem yapmadan ara katmana iletir.

- **Ara katmanlar:** Bu katmanda girdi katmanından gelen bilgiler işlenir. Tek bir ara katmanla bile çok fazla problemi çözmek mümkündür. Fakat ağda öğrenilmesi istenilen problemin girdi ve çıktı arasındaki ilişkisi doğrusal değilse ve karmaşıksa problemin çözümü için birden çok ara katmanda kullanılabilir.

- **Çıktı katmanı:** Bu katmanda ara katmandan gelen bilgiler işlenir ve ağa verilen girdi için ağ çıktı üretir. Bu çıktı dış dünyaya iletilir. Proses elemanlarının girdi ve çıktı katmanlarında kaç tane olması gerektiğine probleme bakılarak karar verilir. Girdi ve çıktı katmanlarında kaç tane proses elemanının olması gerektiğine probleme bakılarak karar verilir. Ağdaki ara katman sayısının ve ara katmanlardaki proses eleman sayısının kaç tane olması gerektiğini bilmenin hiçbir yolu yoktur. Bu sayı deneme yanılma yolu ile öğrenilmektedir. Girdi katmanında bulunan proses elemanların her biri ara katmanda bulunan proses elemanlarının hepsine bağlıdır. Ara katmanda bulunan proses elemanlarının her biri de çıktı katmanındaki proses elemanlarının hepsine bağlıdır. Bilginin akışı girdi katmanından ara katmana ve oradadan da çıktı katmanına doğru ileri yöndedir. Çok katmanlı algılayıcıların eğitilmesi “genelleştirilmiş delta kuralına göre gerçekleşmektedir. Çok katmanlı algılayıcı ağları öğretmenli olan öğrenme stratejileri kullandıklarından eğitim esnasında hem girdiler hem de o girdilere karşılık olan ağın üretmesi gereken çıktılar ağa gösterilirler. Kullanılan öğrenme kuralının felsefesi eğitim esnasında ağın ürettiği çıktılar ile üretmesi gereken çıktılar arasındaki farkın ağın ağırlıklarına dağıtılarak zaman içerisinde bu farkın en aza indirgenmesidir. Öğrenme esnasında başta ağa girdiler verilir ve bu girdilere karşılık gelen çıktılar üretilir. Bu işleme ileri doğru hesaplama denir. Daha sonra üretilen çıktı ve beklenen çıktı karşılaştırılır hatayı en aza indirmek amacıyla aradaki hata geriye doğru dağıtılarak en az hatayla çıkış üretilene kadar ağırlıklar değiştirilir. Buna da geriye doru hesaplama denmektedir.

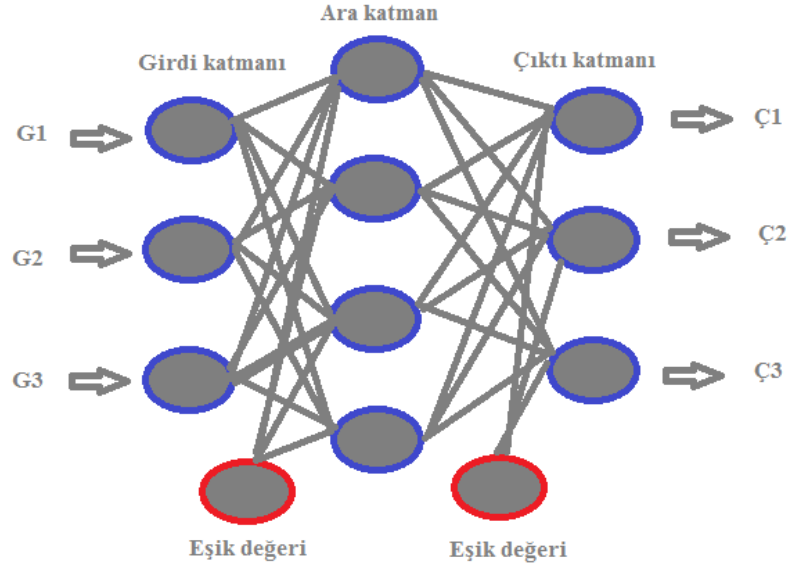
Çok katmanlı algılayıcıların olayları öğrenmesini etkileyen faktörler şunlardır.

- Örneklerin seçilmesi,
- Girdi ve çıktıların ağa sunulması,
- Girdi ve çıktılarının sayısal gösterimi,
- Ağırlıkların başlangıç değerlerinin atanması,
- Öğrenme ve momentum katsayılarının belirlenmesi,
- Örneklerinin ağa sunumu,
- Ağırlıklarının değiştirilme zamanı,
- Girdi ve çıktılarının ölçeklendirilmesi,
- Durdurma kriterinin belirlenmesi,
- Ağların büyütülmesi.

Çok katmanlı algılayıcı ağlarının planlayıcıları belirtilen faktörleri iyi bir şekilde değerlendirmeli ve sorunun çözümü için en uygun yaklaşımı kullanmalıdır. Çok katmanlı algılayıcı ağlarının performanslarını belirlemek için eğitimi bitirdikten sonra eğitimde göremediği örnekleri tekrar ağa göstererek bunlar ile ilgili ağın kararına bakılmalıdır. Eğer ağ daha önce görmediği örnekler için doğru cevaplar üretiyorsa ağın performansı iyidir ve ağ olayı öğrenmiştir diyebiliriz. Çok katmanlı algılayıcılar mühendislik problemlerinde göstermiş olduğu başarılı uygulamalar sayesinde dikkatleri üzerine toplamıştır. Özellikle;

- Sınıflama yapma,
- Tahmin etme,
- Tanıma yapma,
- Yorumlama yapma,
- Teşhis etme.

Konularında Birçok alanda başarılı uygulamalar görülmektedir.



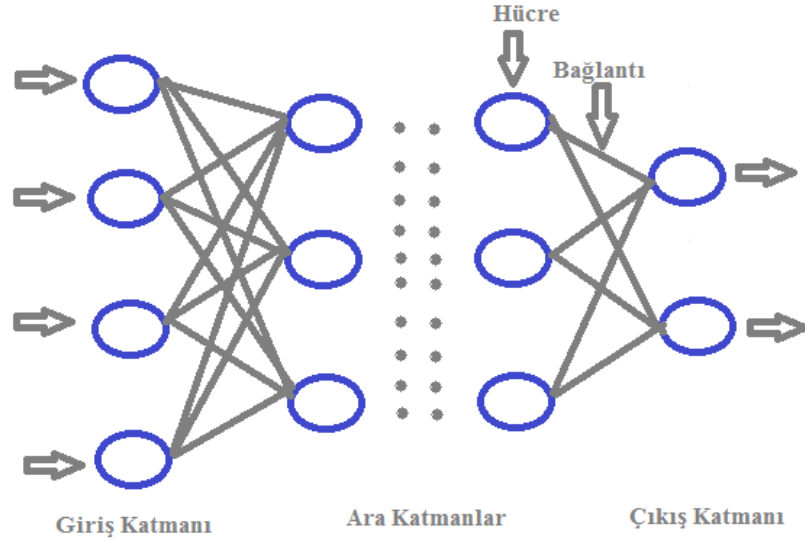
Şekil 2.3. Yapay Sinir Ağı Katmanları (URL7, 2017)

Şekil 2.3’de görüldüğü üzere bir sinir ağı girdi katmanı, ara katman ve çıktı katmanı olmak üzere 3 katmandan oluşmaktadır. Burada mavi ile gösterilenlerin her biri nöronları, oklar ise bağlantı noktalarını temsil etmektedir. Eğitim için girdi katmanında hazırlanan veri seti ağına verilir. Ağ ara katmanda bulunan bağlantı noktalarına öğrenilen olayların ağırlıklarını atar. Her bağlantı noktasının bir değerinin olmamasıyla birlikte bazı noktaların değeri sıfır olabilmektedir. Bağlantı noktalarındaki sıfır değerleri çıktımızı da sıfır yapmasını diye katmanlar arasında bir eşik değeri eklenmektedir.

Sonunda ağı test etmemiz gerekmektedir. Böylelikle veri setinden farklı olarak çıktı sonuçları olmayan bir test seti ağına gösterilirse ve ağ bize bu test setinin her olayı için bir çıktı değeri verir. Bu çıktı değerlerinin yorumlanmasıyla da sonuçlar elde edilir.

2.1.1.3. İleri Beslemeli Yapay Sinir Ağı

İleri beslemeli yapay sinir ağlarında tek yönlü olan sinyal akışı vardır. Bununla birlikte, ileri beslemeli sinir ağları çoğu katmanlar halinde organize edilmektedir. Üç katmanlı ileri beslemeli sinir ağı bir örneği Şekil 2.4’te gösterilmiştir. Bu ağ giriş katmanları, iki gizli katman ve bir çıkış katmanından oluşmaktadır (URL6, 2017).



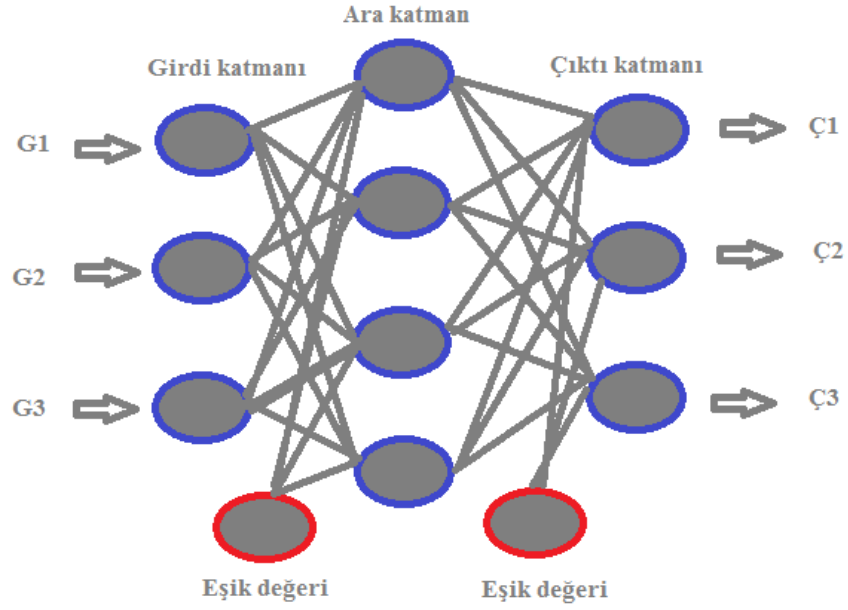
Şekil 2.4. İleri beslemeli yapay sinir ağı örneği(URL6, 2017)

İleri beslemeli yapay sinir ağlarında hücreler katmanlar şeklinde olur ve bir katmandaki hücrelerin ürettiği çıkışlar ağırlıklar üzerinden bir sonraki katmana giriş olarak sunulur. Giriş katmanı, dış ortamdan almış olduğu bilgileri hiçbir işleme sokmadan ara katmandaki hücrelere gönderir. Bilgi, ara ve çıkış katmanlarında işlenip ağ çıkışı belirlenir. Bu modeli ile ileri beslemeli ağlar, doğrusal olmayan bir fonksiyonu gerçekleştirir. İleri beslemeli üç katmanlı YSA'nın, orta katmanında yeterli sayıda hücre olmak şartıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştırabileceği gösterilmiştir. En çok bilinen algoritma olan geriye yayılım öğrenme algoritması, bu tarz yapay sinir ağların eğitiminde etkili olarak kullanılmaktadır. Ağa, hem örnekler hem de örneklerden elde edilmesi gereken çıktılar (beklenen çıktılar) verilmektedir. Ağ kendisine gösterilen örneklerden genellemeler yaparak problem uzayını temsil eden bir çözüm uzayı üretmektedir. Daha sonra gösterilen benzer örnekler için bu çözüm uzayı sonuçlar ve çözümler üretebilmektedir (URL6, 2017). İleri beslemeli ağa ilişkin iki farklı örnek şu şekilde verilebilir:

Tek katmanlı algılayıcılar: En basit ileri beslemeli sinir ağıdır. Bu ağ herhangi bir gizli katman içermez.

Çok katmanlı algılayıcılar: Bir ya da birden çok gizli katmanı olan sinir ağıdır. Tek katmanlı perceptronlara kıyasla daha kullanışlı olmalarından dolayı daha çok çok katmanlı perceptronlar üzerinde durulmaktadır. Tek bir katmanlı perceptron doğrusal fonksiyonları içerirken çok katmanlı bir perceptron doğrusal olmayan fonksiyonları öğrenebilmektedir.

Şekil 2.5'te tek bir gizli katmana sahip çok katmanlı bir perceptron görülmektedir. Tüm bağlantılarda ağırlıklar bulunmaktadır.

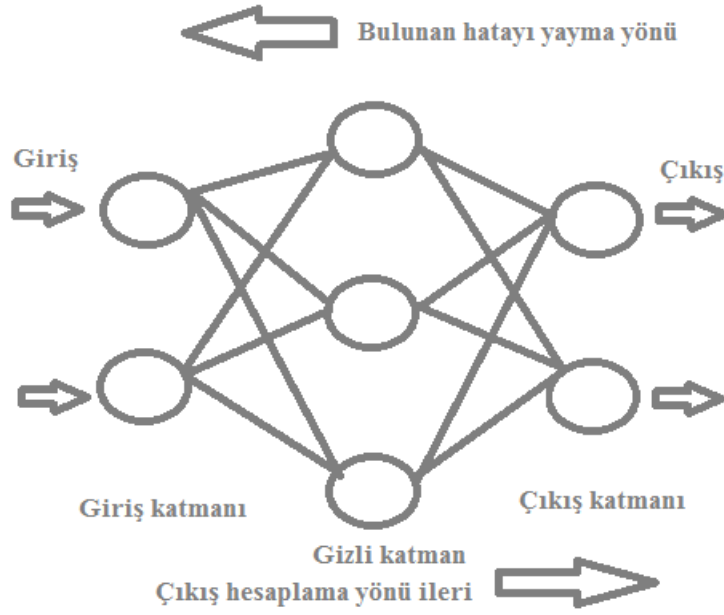


Şekil 2.5. Çok katmanlı perceptron (URL7,2017)

2.1.1.4. Geri Beslemeli Yapay Sinir Ağı

Geri beslemeli sinir ağı, ara katmanlardaki ve çıkıştaki verilerin, girişe ya da önceki ara katmanlara geri beslendiği bir ağ modelidir. Böylece, girişler ağ içerisinde hem ileri yönde hem de geri yönde aktarılmış olur.

Şekil 2.6'da bir geri beslemeli YSA yapısı gösterilmiştir.



Şekil 2.6. Geri beslemeli yapay sinir ağı (URL6, 2017)

Bu tip yapay sinir ağlarının dinamik hafızaları bulunmakta ve bir andaki çıkış hem o andaki hem de önceki girişleri yansıtmaktadır. Bu sebeple, bu ağ modeli önceden tahmin uygulamaları için çok uygundur. Geri beslemeli ağlar çeşitli tipteki zaman-serilerinin tahmininde oldukça başarı sağlamışlardır. Bu ağlara örnek olarak Hopfield, SOM (Self Organizing Map), Elman ve Jordan ağları verilebilir (URL6, 2017).

2.1.1.5. Yapay Sinir Ağlarının Avantajları ve Dezavantajları

YSA'ların bilinen avantajları şu şekildedir;

Bilgilerin ağıın tamamında saklanması: Yapay sinir ağları geleneksel programlamadakinin aksine bilgileri bir veri tabanında değil, ağıın tamamında yayılarak saklar. Bazı bilgilerin kaybolması ağıın çalışmaması için bir engel değildir.

Eksik bilgi ile çalışabilme: Yapay sinir ağları eğitimlerini tamamladıktan sonra eksik bilgi içeren veriler olsa bile çıktı üretebilirler. Ancak burada performansın düşmesi eksik bilginin önemine bağlıdır.

Hata tleransına sahip olma: Yapay sinir ađlarında bir yada daha ok hcrenin bozulması ađın ıktı retmesini engellemez. Bu zellik sayesinde ađlar hata toleransına sahip olur.

Dađıtık hafızaya sahip olma: Ađın renebilmesi iin rnekler belirlenir ve bu rnekler ađa gsterilerek istenen ıktılara gre ađın eđitimi sađlanır. Seilmiř olan rnekler ađın performansı ile dođru orantılıdır. Olaylar ađa btn ynleri ile gsterilmelidir. Aksi taktirde ađın yanlış ıktılar retmesi mmkndr.

Dereceli bozulma: Ađ herhangi bir sorunun ortaya ıkmasıyla birlikte hemen bozulmaz. Ađ zamanla yavař ve greceli bir řekilde bozulur.

Makina đrenmesi yapabilme: Yapay sinir ađları yorum gcne sahip olduđu iin olayları đrenerek benzer durumlar karřısında karar verebilme yeteneđine sahiptirler.

Paralel iřlem yeteneđi: Yapay sinir ađlarının aynı anda birok iři gerekleřtirebilecek sayısal kapasitesi vardır.

YSA'ların bilinen dezavantajları ise řu řekildedir;

Donanım bađımlı olması: Yapay sinir ađları yapısından dolayı paralel iřlem gcne sahip olan iřlemciler gerektirir. Bundan dolayı donanıma bađımlı alıřırlar

Ađın davranıřlarının aıklanamaması: Yapay sinir ađlarının en nemli problemidir. Yapay sinir ađları herhangi bir probleme zm sađladıđı zaman bunun nasıl ve niin olduđuna dair herhangi bir ipucu vermez. Bundan dolayı da ađa olan gven azalır.

Uygun ađ yapısının belirlenmesi: Yapay sinir ađlarının yapısını belirlemek iin herhangi zel bir kural yoktur. En uygun ađ yapısı deneyimle ve deneme yanılma yntemiyle sađlanır.

Problemin ađa gsterim zorluđu: Yapay sinir ađları nmerik bilgiler ile alıřabildiđinden dolayı problemlerin ađa tanıtılmadan nce nmerik bilgilere evrilmesi

gerekmektedir. Belirlenecek olan gösterim mekanizması ađın başarısını dođrudan etkileyecektir. Bu da kullanıcının kabiliyetine bađlıdır.

Ađın eđitim süresinin bilinmemesi: Eđitim ađın örnekler üzerindeki hataların belirli bir deđerin altına indirilmesi ile tamamlanır. Bu deđer bize en iyi sonuçları vermemektedir.

Gelişmekte olan yapay sinir ađlarının dezavantajlarının gün geçtikçe giderildiđi ve avantajlarının da her geçen gün arttıđı göz önünde bulundurulmalıdır. Bu sebeple yapay sinir ađları yaşamımızdaki önemini git gide artıracak ve bizim için vazgeçilmez bir hal alacaktır.

2.1.1.6. Yapay Sinir Ađlarının Kullanım Alanları

Yapay sinir ađlarının günümüzdeki kullanımı git gide hızlı bir şekilde artmaktadır. Günümüzdeki sorunlar karmaşık ve geleneksel algoritmalarla çözümlenemediklerinden dolayı zeki algoritmalara gereksinim duymaktadır. Yapay sinir ađları, karar verme mekanizması olarak kullanılmakla birlikte bu özellikleri de taşıdığından dolayı geniş çapta tercih edilmektedir. Yapay sinir ađı modelleri, sınıflandırmada çok iyi bir karakteristiđe sahiptir. Bu ađ sorgulanan bilginin hangi sınıfa ait olacağına cevabını vereceğinden dolayı verilen girdi üzerinde de bir karar verecektir. Bu özellikleri sayesinde yapay sinir ađları günümüzde bir çok alanda faaliyet göstermektedir. Bunlar; görüntü işleme, ses işleme, örüntü tanıma sinyal işleme vb. şeklindedir. Örneğin; sinyal işlemede sinyallerin iletildiđi frekans aralıklarındaki gürültülerin temizlenmesinde yapay sinir ađ modeli kullanılmaktadır.

Yukarıdaki özellikleriyle yapay sinir ađları günümüzde; sinyal işleme, görüntü işleme, ses algılama, örüntü tanıma gibi birçok alanda kullanılmaktadır. Sinyal işlemede, sinyallerin gönderildiđi frekans aralıklarında oluşan gürültülerin temizlenmesinde yapay sinir ađları kullanılmaktadır. Ses algılamanın genellikle güvenlik sistemlerinde ses kontrollü cihazlarda veya yazılım geliştirmede kullanımı görülmektedir (Öztemel, 2012).

2.2. Bulanık Mantık

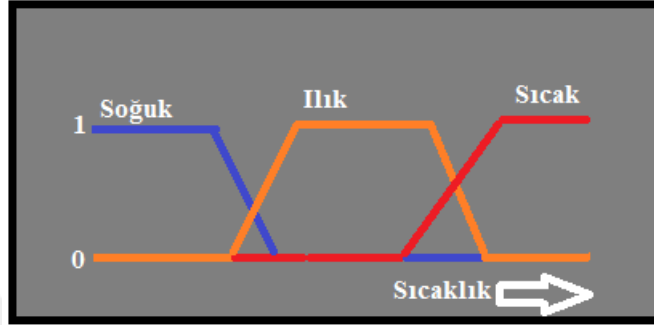
Günlük hayatımızda kullandığımız terimlerin birçoğu genellikle bulanık bir yapıya sahiptir. Bir şeyi tanımlamak, bir durumu açıklamak ve komut vermek için kullandığımız sayısal ya da sözel ifadeler bulanıklık içermektedir. Bunlara örnek olarak; yakın, uzak, kolay, zor, az, biraz, çok az, sıcak, soğuk, ılık, karanlık, çok karanlık, aydınlık, yavaş çok yavaş, hızlı, gibi birçok sözel terim bulunmaktadır. Biz bu tip kesinlik ifade etmeyen terimleri herhangi bir durum için karar verirken, durum hakkında yorum yaparken, bir durumu anlatmak için kullanırız. Havanın ısı durumuna göre sıcak, çok sıcak, soğuk, çok soğuk gibi terimler kullanırız. Bireyin yaş durumuna göre genç, orta yaşlı, yaşlı, çok genç ve çok yaşlı deriz. Bir nesnenin bize olan mesafesine göre yakın, çok yakın, az yakın, uzak, çok uzak deriz. Dinlediğimiz müziğin sesi yetersiz ise onu biraz artırırız, ses yeterinden fazla ise biraz azaltırız. Bunlar kesinlik göstermeyen durumlarda olaylara karşı nasıl davranıldığına ve olayların nasıl değerlendirilip yorumlanıldığına, nasıl tanımlanıp komut verildiğine dair örnekler..

Bulanık mantığın ve bu mantık kurallarını kullanan bulanık küme teorisinin Lotfi A. Zadeh tarafından geliştirilmekte olup 1965 tarihli orijinal makalesinde (Zadeh, 1965) yayınlanmasının ardından belirsizlik içeren sistemlerin incelenmesi yeni bir boyut kazanmıştır. 1965'te ortaya konulmasına rağmen, bulanık küme kavramı ancak 1970'li yılların ikinci yarısından sonra kullanılmaya başlanmıştır. Bunda özellikle Zadeh'in 1965 'teki ilk makalesinden (Zadeh, 1965) daha çok etkili olan ve bulanık mantığın belirsizlik içeren sistemlere uygulanabilirliğini açıklayan makaleleri (Zadeh, 1973)-(Zadeh, 1975) etkili olmuştur. Japonlar 1980'li yılların ikinci yarısından sonra ürünlerinde bulanık mantığı kullanmışlardır. Bundan dolayı bulanık mantık oldukça hızlı bir şekilde gelişerek, günümüzdeki zirvesine ulaşmıştır. Bulanık mantık uygulamalarına günümüzde neredeyse her alanda rastlamak mümkün olmuştur.

Bulanık mantığı Mamdani ve arkadaşlarının denetim sistemlerine ilk kez uygulamalarından dolayı bulanık mantığın en çok kullanıldığı alan olarak günümüzde önemli bir yer almıştır (Mamdani ve Assilian, 1975).

Klasik denetim sistemlerinden farklı olarak, sistemlerin matematiksel modeline gereksinim duymadan, sadece istenilen çıkışı üretecek şekilde sisteme uygulanan sinyalleri doğruladığından dolayı, bulanık denetimin işlemesi tıpkı usta bir insanın o sistemi denetlemesine benzetilmektedir (Altaş, 1999). Bulanık mantık ve bulanık küme işlemleri

uygulanarak makinelerin insanlar gibi davranıp karar vermesi gerçekleştirilebilir. Yapay sinir ağıları ya da genetik algoritmalarla bulanık mantığın destek görmesi sonucunda nöral bulanık sistemler ya da genetik bulanık sistemler gündeme gelmiştir. Bu nedenle akıllı sistemler de oldukça hızlı bir gelişme göstermeye başlamıştır (Liu, 1997)-(Farrell ve ark., 1997). Şekil 2.7' de bulanık mantık örneği görüntüsü verilmiştir.



Şekil 2.7. Bulanık mantık örneği

2.2.1. Bulanık Mantığın Genel Özellikleri

- Bulanık mantıkta, kesin değerlere dayanan düşünme yapısı kullanılmaz. Bunun yerine yaklaşık bir düşünme yapısı kullanılır.
- Bulanık mantıkta her şey $[0,1]$ aralığında belirlenmiş olan bir dereceleyle gösterilir.
- Bulanık mantıkta bilgi yakın, uzak, az, çok, küçük, büyük gibi dilsel ifadeler şeklinde olur.
- Her mantıksal sistem bulanık olarak ifade edilir.
- Bulanık çıkarım işlemi dilsel ifadeler arasında tanımlanan kurallar ile yapılır.
- Bulanık mantığın belirsiz ve tam girilmemiş olan bilgiler üzerinde işlem yapabilme yeteneği vardır.
- Bulanık mantık matematiksel modeli çok zor elde edilen sistemler için çok uygundur.

2.2.2. Bulanık Sinirsel Sistemler

Bulanık mantık ve yapay sinir ağlarının harmanlanmasından oluşan bulanık sinirsel sistemler örüntü tanıma, ses tanıma, sistem modellemesi yapma gibi alanlarda önemli derecede başarı gösterebilecek kapasiteye sahiptir.

Bulanık sinirsel sistemlerde ağın tasarımı üyelik fonksiyonlarının üretilmesini sağlayacak biçimdedir. İstenilen çıkış değeri bulunduğunda, üretilmiş olan üyelik fonksiyonları optimum kabul edilir ve yapay sinir ağları devre dışı bırakılır.

2.2.3. Bulanık Mantığın Uygulama Alanları

Günümüzde neredeyse her alanda uygulama olanağı bulan bulanık mantık, özellikle sanayi alanındaki uygulaması yaygındır. Japonlar bulanık mantığı özellikle bulaşık makineleri, çamaşır makineleri, elektrik süpürgeleri, video kameralara uygulamışlardır.

Bulanık mantık uygulamaları ilk olarak çimento sektöründe kullanılmaya başlanmıştır (Büyükkaracığan ve ark. 2011). Bu sektörde bulanık mantık yapısına uygun bir çok problem bulunmaktadır. Örneğin üretim esnasında ısıyı 5 derece yükselt veya azalt gibi kurallar yerine genellikle biraz yükselt veya biraz azalt gibi bulanık terimler kullanılmaktadır. Daha sonraları bulanık mantık, insansız uçakların kontrolünde, tren frenleme sistemlerinde, ABS (otomatik fren sistemi) ve ASC (otomatik vites kontrolü) kontrolünde kullanılmıştır.

2.2.4. Bulanık Mantığın Avantajları ve Dezavantajları

Bulanık mantığın bilinen avantajları şu şekildedir;

- Günlük hayatımızda sürekli karşılaştığımız belirsiz, karmaşık, zamanla değişip kesinlik içermeyen ve iyi tanımlanmamış sistemlerin denetlenmesine basit çözümler getirerek kolaylık sağlar.
- Eğer sistem basit bir matematiksel model ile tanımlanabilecek bir sistemse sadece geleneksel bir denetimin olması yeterli olacaktır. Fakat geleneksel bir mantığı karmaşık bir sisteme uygulamak hem oldukça zor hem de çok maliyetli olur. Bunun yanında

bulanık mantık denetimi geleneksel mantığa kıyasla sistemi daha iyi çözümlenmekle birlikte daha ekonomik bir yapıya da sahiptir.

- Bulanık mantık genelde küçük bir yazılım aracılığı ile daha hızlı bir şekilde sonuç alır. Bunun başlıca sebebi bulanık mantıkta işaretlerin bir ön işleme koyulmaları ve olabildiğince geniş bir alana yayılan değerlerin çok az sayıda üyelik fonksiyonlarına indirgenmeleridir. Bahsedilen az sayıdaki değerler üzerinde uygulanabilecek kural sayısının da az olması nedeniyle sonucu bulmak oldukça kolaylaşacaktır.

- Bu durum geleneksel bilgisayar ortamlarında bu şekildedir. Geliştirilmiş bir donanım ile sonucu daha erken bulmak mümkündür. Bazı özel firmalar uygulamalarında kullanmak istedikleri mikro bilgisayarların yetersiz kalacaklarını düşündükleri için bulanık denetim kullanmaya karar vermişlerdir. Bulanık denetim yazılım boyutlarının olabildiğince küçülmesini sağladığından dolayı dış bellek kullanımına gerek duyulmamıştır.

- Kullanıcı girişlerinden ve kullanıcı deneyimlerinden doğrudan faydalanabilmesine olanak sağlaması da bulanık mantık denetiminin sağladığı avantajlardandır.

- Otomatik araçlardaki vites değişimi motorun belli hızlara ulaşması sonucunda kendiliğinden gerçekleşir. Buna karşılık manuel vitesli bir araçlarda ise sürücü, yol, ağırlık ve kullanım tarzına göre belli durumlarda vites değiştirir. Subaru tarafından üretilen justy tipi otomobilde kullanılan aktarım biriminin değiştirilmesi, bir kayışın konumunun BM kullanılarak değiştirilmesi ile gerçekleştirilir. Böylece arabanın ivmesi ve performansı sürekli olarak düzenlenebilir hale gelir. Subaru, otomobilinde kullandığı BM üyelik fonksiyonlarını, otomobili test sürücülerine kullandırarak ve onlardan ivme ve performans açısından en iyi aktarım birimini öğrenerek düzenlemiştir. Bu konuda Honda ve Nissan firmaları da benzer çalışmalar yapmışlardır (Akdemir, 2011).

Bulanık mantığın bilinen dezavantajları şu şekildedir;

- Bulanık denetim için kullanılan kurallarda tecrübe oldukça önemlidir.
- Üyelik fonksiyonlarının seçimi için özel bir metod bulunmamaktadır. Bu fonksiyonların seçimi için en uygun metod deneme yanılmadır. Fakat bu çok uzun bir zamanda gerçekleşebilir.

- Denetlenen sistem için bir kararlılık analizi yapılamaz ve sistemin nasıl cevap vereceği önceden tahmin edilemez. Burada sistem için yapılacak tek şey benzetim çalışmasıdır.

2.3. Destek Vektör Makineleri

Değişkenler arasındaki örüntülerin ayırt edilmediği veri setlerindeki sınıflama problemleri için kullanılan oldukça etkili ve basit bir makine öğrenmesi yöntemidir. Bu yöntem sınıflama, değişkenler arasındaki ilişkiyi ölçme ve aykırı değer belirlemede kullanılabilen eğitici öğrenme çeşididir. Destek vektör makineleri eğitim verileri üzerinde öğrenerek yeni veriler için doğru tahminler yapmayı ve genellemeyi amaçlayan bir makine öğrenmesidir. İstatistiksel öğrenme teorisine ve yapısal risk minimizasyonuna dayanmaktadır.

2.3.1. Destek Vektör Makinelerinin Avantaj ve Dezavantajları

Destek vektör makinelerinin bilinen avantajları şu şekildedir;

- Yüksek oranda doğruluk gösterirler,
- Kompleks karar sınırlarını modelleyebilirler,
- Çok sayıda bağımsız değişkenle çalışabilirler,
- Hem lineer şekilde ayrılan hem de lineer şekilde ayrılamayan verilere uygulanabilirler,

Birçok yönteme kıyasla problem azdır.

Destek vektör makinelerinin bilinen dezavantajları şu şekildedir;

- Tahminleri olasılıksal olarak üretmezler. Yani nokta tahmini yaparlar.(Evet-Hayır, Var-Yok vb..),
- Çekirdek fonksiyonları Mercer teoremi olarak bilinen matematiksel bir kurala uymak zorundadır. Çekirdek fonksiyonları pozitif tanımlı sürekli simetrik fonksiyonlardır.

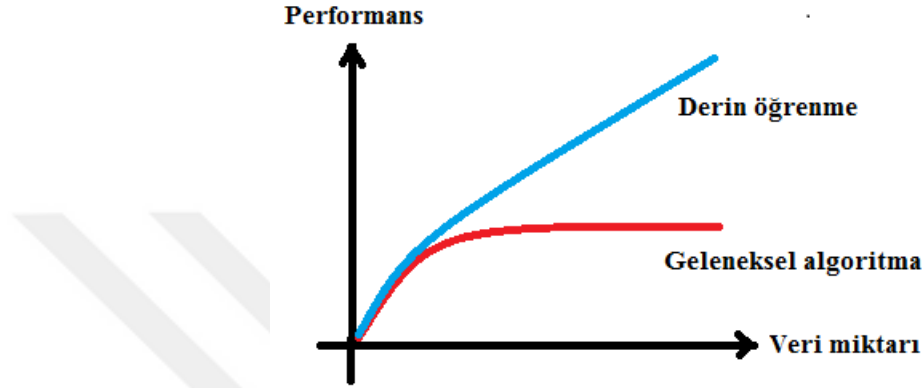
2.4. Derin Öğrenme

Derin öğrenme, yapay sinir ağları olarak adlandırılan beynin yapısından ve işlevinden esinlenmiş olan algoritmalarla alakalı makine öğrenmesinin bir alt dalıdır. Makine öğrenmesi, derin öğrenme ve yapay zeka ifadeleri birbirleri ile ilişkili iç içe kavramlardır (URL1, 2017). Yapay sinir ağlarının ilk geliştirilen modeli perceptron öğrenme algoritmasıdır. Derin öğrenme de perceptron öğrenme algoritmasının geliştirilmiş hali olarak düşünülmektedir. Geliştirilen ilk modelde nöron sayısı oldukça azdır ve bundan dolayı kompleks problemlerin çözülmesi olası değildir. Bu durumu örnekleyecek olursak, bu modelde ilk olarak “And” yapısının oluşturulmasında yüksek performans görülmektedir ve “OR” yapısının oluşturulmasında performans görülmemektedir. Çünkü “OR” yapısının oluşturulması için ek katmanlara gereksinim duyulmaktadır. Kompleks bir hal alan problemin yapay sinir ağları ile çözülmesi katman sayısının artırılması ile mümkündür. Katman sayısı arttıkça karar mekanizması daha da derinleşmektedir. Derin öğrenmeyi geleneksel yapay sinir ağı modellerinden ayıran en önemli özelliklerinden biri çok fazla sayıda katmanla çalışabilmesidir. Geleneksel yapay sinir ağlarında her hücre kendinden bir önceki ve bir sonraki katmanlarda bulunan tüm hücrelere bağlı durumdadır. Her bağlantıda hesap gerektiren matematiksel işlemler mevcuttur. Katman sayısında ve hücre sayısında artışın olabilmesi için yüksek miktarda CPU gücüne ihtiyaç duyulmaktadır. Derin bir ağ yapısının oluşturulması için kişisel bilgisayarlar içerisindeki CPU gücü tam anlamıyla yeterli değildir.

Son yıllarda, derin öğrenmedeki önemin çok fazla artmasındaki en önemli faktörlerden biri GPU’ların bu alan için elverişli olmasıdır. Günümüzdeki işlemci kapasitesi oldukça yüksek bilgisayarlarla sağlandığından dolayı çok büyük yapay sinir ağlarının eğitimi yapılmaktadır. Kısaca açıklayacak olursak, derin öğrenmenin uygulanabilmesi için GPU üzerinde yapılan teknolojik gelişmeler oldukça önemli bir husustur. Bu gelişmelerin sağladığı katkılar sayesinde büyük miktarda veriler ile sistem eğitilebilmektedir. Bununla birlikte büyük miktarda veriyle yapılan eğitimde performansta git gide artış göstermektedir. İşte derin öğrenme bu noktada geleneksel makine öğrenmesi tekniklerinden ayrılır. Çünkü geleneksel makine öğrenmesi uygulamalarında büyük miktarda veri, performansın biraz artmasına ve daha sonra performansın sabit bir şekilde kalmasına sebep olmaktadır.

Elimizde çok miktarda veri olmasının faydalarını kullanmamaktaydık. Elimizde çok miktarda veri olmasının avantajını kullanamamaktaydık. Verilerin artması ve artan veri

miktarı ile daha başarılı sistemlerin oluşturulabilmesi sayesinde derin öğrenme son yıllarda üzerinde oldukça çaba gösterilen popüler bir konu haline gelmiştir. Derin öğrenme ile geleneksel öğrenmesi algoritmalarının artan veri miktarına göre elde edebildikleri başarı oranları aşağıdaki grafikte verilmiştir. Şekil 2.8’ de derin öğrenme ve geleneksel makine öğrenmesi başarı oranları gösterilmiştir.



Şekil 2.8. Derin öğrenme ve geleneksel makine öğrenmesi başarı oranları

Son yıllarda, derin öğrenme kullanarak akıllı sistemler oluşturma, büyük teknoloji firmalarının gündeminde olan bir konu haline gelmiştir. Aşağıda bazı büyük teknoloji firmalarının derin öğrenme üzerine yaptıkları araştırmalara dair bilgiler sunulmuştur.

Google 2011 yılında derin öğrenme odaklı google beyin projesini başlattı, 2012 ortasında konuşma tanıma ürünlerine sinir ağlarını tanıttı ve 2013 yılının Mart ayında geoffrey hinton sinir ağlarına öncülük etti. Arama android, gmail, fotoğraf, tercüme etme ve kendi kendine sürülen otomobillere kadar uzanan 1000’den fazla derin öğrenme projesi vardır.

Microsoft, 2011’in ilk yarısı boyunca bing sesli arama ve x-Box sesli komutları da dahil olmak üzere, ticari konuşma tanımada derin öğrenme kullanmıştır. Şirket günümüzde fotoğraf araması, çeviri sistemleri, ses tanıma ve daha fazlası için sinir ağlarını kullanmaktadır. Lee şöyle diyor “sahip olduğu yaygın etkiyi göstermek zordur. Son yıllarda görüntü tanımayı başarmıştır ve eylülde, konuşma tanımadaki deneyleri düşük hata oranıyla bir rekor elde etmiştir.

Mayıs 2014’te Baidu daha önce araştırma laboratuvarlarını yönetmek için Google beyin projesini yönetmeye ve başlatmaya yardım eden Andrev Ng den yardım almıştır.

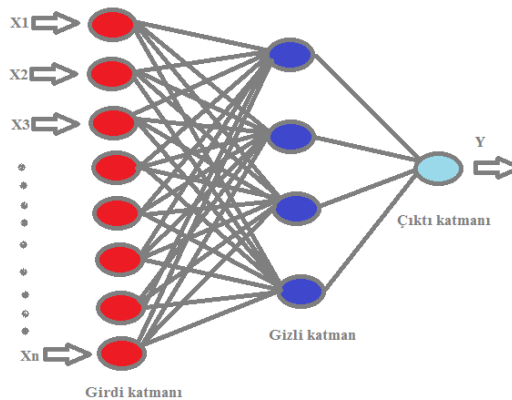
Çin'in önde gelen arama ve web servis sitesi olan Baidu konuşma tanıma, çeviri, fotoğraf arama ve kendi kendine giden araba projesi için sinir ağlarını kullanmıştır.

2.4.1. Derin Öğrenmenin Çalışma Prensipleri

Derin öğrenme, ayırt edici özelliklerini verilen çok sayıda girişe bakarak kendisi öğrenir. Eğer sistem yeteri kadar eğitilirse özellik öğrenme işlemi başarı ile sonuçlanır. Özellik öğrenme aşaması katmanlardan oluşmaktadır. Burada alt seviyedeki özellikler daha az ayırt ediciliğe sahiptir. Alt seviyedeki katmanların birleştirilmesinden oluşan üst seviye katmanlarındaki özellikler daha çok ayırt ediciliğe sahiptirler. Alt seviyedeki katmanlar daha anlamlı özellikler üretilebilmesi için bir alt yapı oluşturmaktadır. Bu tip bir öğrenme metodu, geleneksel makine öğrenmesi algoritmalarından farklıdır. Çünkü geleneksel makine öğrenmesi algoritmalarında insan tarafından belirlenmiş olan özellikler eğitimden önce hesaplanmalıdır. Hesaplanan bu özelliklere bakılarak öğrenme işlemi gerçekleştirilir.

Kısaca anlatacak olursak geleneksel algoritmalar insan bağımlı özelliklerle çalışmaktadır. Buna karşı olarak derin öğrenme insan bağımsız özellikler ile çalışmaktadır. Bu da derin öğrenme algoritmasının kendi kendine ayırt edici özellikleri öğrenmesi demektir. Derin öğrenme, bu kabiliyetinden dolayı oldukça büyük bir başarı göstermektedir (Jain ve ark., 1996).

2.4.2. Derin Sinir Ağları



Şekil 2.9. Derin sinir ağ yapısı (URL9,2017)

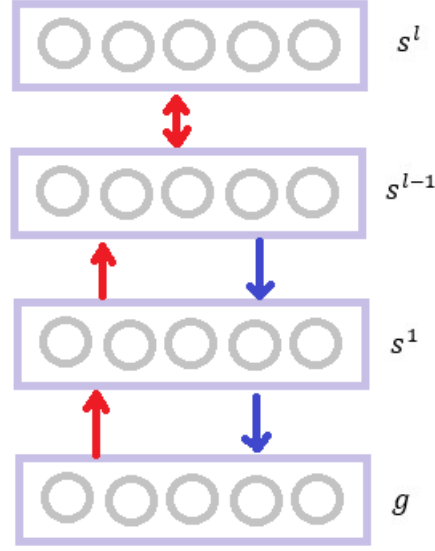
Derin sinir ağıları, birden çok gizli katmana sahip olan bir yapay sinir ağıdır. Geleneksel makineler, en fazla bir gizli katmandan oluşmakla birlikte, bir giriş ve bir çıkış katmanından oluşmaktadır. Derin öğrenmenin ise üçten fazla katmanı vardır. Mutlaka birden çok gizli katmana sahiptir. Derin sinir ağlarında, her giriş katmanı, kendinden önceki katmanın çıktısına bağlı olarak değişik özellik setleri üzerinde eğitilmektedir. Sinir ağları gelişme gösterdikçe, girişlerin tanıdıkları özelliklerin karmaşıklığı artmaktadır. Karmaşıklığın artış göstermesinin sebebi önceki katmandaki özelliklerin bir araya gelerek birleşmesidir. Bu ağ özelliği, verilerin çoğunluğunu oluşturan etiketsiz ve yapılandırılmamış verilerdeki gizli yapıları bulmaktadır. Gizli katman sayısının artırılması ile ağa derinlik özelliği kazandırılır. Fazladan bir nöronun bağlanması yapay sinir ağlarının önemli bir özelliğidir. Eğitim veri kümesinin lineer olarak ayrılabilen iki sınıfa sahip olması durumunda belli bir iterasyonda eğitimin gerçekleşmesinden sonra perceptron öğrenme işlemini gerçekleştirebilmektedir (Jain ve ark., 1996). Şekil 2.9’ da derin sinir ağı yapısı gösterilmiştir.

2.4.3. Derin İnanç Ağları

Derin inanç ağları (DİA), girişine uygulanan verinin hiyerarşik bir biçimde oluşturulmasını sağlayan graf-tabanlı modellerdir. Birden çok RBM’nin ard arda bağlanmasından meydana gelen derin inanç ağları, ağ yapısını oluşturan RBM’lerin belli bir düzenle eğitilmesinden dolayı iyi öğrenilirler. Böylelikle girişe uygulanan veri (x) ile aradaki saklı l katman arasında ortak bir olasılık dağılımı (2.1) deki gibi modellenir. (2.1) de s^{ℓ} , l saklı katmanda bulunan veriye karşılık gelen vektördür. g görünür giriş katmanındaki veriye karşılık gelen vektördür. Giriş katmanına eğitim maksadıyla öznitelik vektörleri uygulandığından

$$P(g, s^1 \dots, s^{\ell}) = \left(\prod_{k=0}^{\ell-2} p(s^k | s^{k+1}) \right) P(s^{\ell-1}, s^{\ell}) \quad (2.1)$$

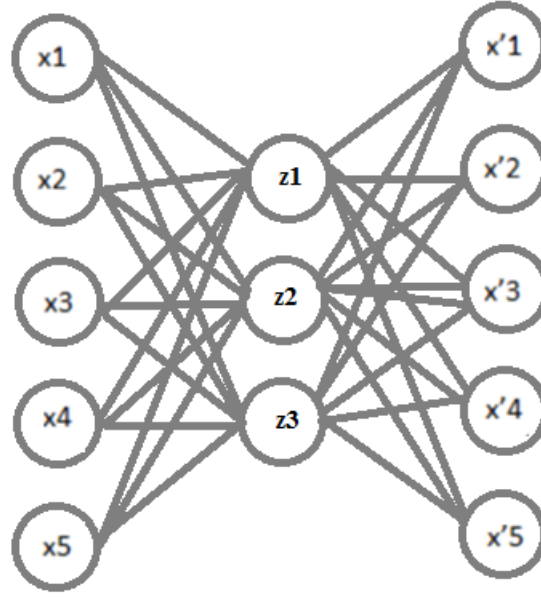
$k=0$ iken $g=s^0$ olarak alınır. $P(s^k | s^{k+1})$, k görünür katman için bir sonraki saklı katmanla oluşturduğu koşullu olasılıktır. $P(s^{\ell-1}, s^{\ell})$ ise son katmandaki görünür ve saklı RBM katmanları arasındaki ortak olasılık dağılımıdır. Şekil 2.10’ da derin inanç ağları yapısı gösterilmiştir.



Şekil 2.10. Derin inanç ağların yapısı (URL10, 2017)

2.4.4. Yığın Otomatik Kodlayıcılar

Oto kodlayıcılar yapay sinir ağı modelinden esinlenerek elde edilmiş denetimsiz bir makine öğrenme metodudur. Geleneksel bir yapay sinir ağı modelinde girdi katmanı, ara katman ve çıktı katmanı olmak üzere üç katman bulunmaktadır. Nöronlar ağın her katmanında belirli sayılarda bulunmaktadır. Veri seti üzerindeki özellik sayıları girdi katmanında bulunan nöron sayılarını tanımlamaktadır. Çıktı katmanındaki nöron sayısını belirleyen etken ise sonuçlanması beklenen sınıf sayısıdır. Gizli katman sayıları ve gizli katmanda bulunan nöron sayıları önceden belirlenmemiştir. Nöron sayıları genellikle deneme ve yanılma metodu ile ortaya koyulur. Yapay sinir ağlarının karmaşık olmayan veri setleri üzerindeki performansları oldukça iyidir fakat karmaşık veri setleri için yeteri kadar performans gösterilmemektedir. Oto kodlayıcı modeli ise şekil 2.11’de gösterildiği gibi daha karmaşık veri setlerinde performansı artırmak için benzer bir ağ modelini iyileştirerek kullanmıştır. Oto kodlayıcı modelinde genellikle gizli katmanda bulunan nöron sayısı girdi katmanında bulunan nöron sayısından azdır. Otomatik kodlayıcı modelini ileri beslemeli yapay sinir ağlarından ayırt eden ikinci önemli özeliği ise çıkış veri setinin giriş veri seti ile aynı olmasıdır. Böylece girdi katmanında bulunan nöron sayısı ile çıktı katmanında bulunan nöron sayısı birbirleriyle eşit durumda olur (Ng, 2011).



Şekil 2.11. Otomatik kodlayıcı gösterimi (Ng, 2011)

Oto kodlayıcılar teknik olarak bir sınıflama işlevi yapmazlar. Burada maksat N boyutunda bir özellik vektörünü en az kayıp olacak şekilde daha küçük bir boyuta düşürmektir. Dolayısıyla tüm özellikler öncelik olarak girdi katmanında okunur. Daha sonra bu katmanda bulunan bilgiler denklem 2.2’de gösterildiği gibi ara katmana iletilir. Denklem 2.2 ’de x_j girdi katmanındaki j’inci nöronun değerini, y_i gizli katmandaki i’inci nörona aktarılan değeri, n girdi katmanındaki nöron sayısını, w_{ji} girdi katmanındaki j’inci nörondan ara katmandaki i’inci nörona iletilen ağırlığı, f ise aktivasyon fonksiyonunu(örneğin: gauss, softmax, sigmoid) temsil etmektedir.

$$y_i = f(\sum_{j=1}^n x_j * w_{ji}) \quad (2.2)$$

İlk aşamadan sonra elde edilen değerler denklem 2.3’te gösterildiği gibi çıktı katmanına iletilerek, son değerler hesaplanır. Denklem 2.3’te x_j' çıktı katmanındaki j’inci nöronu, y_i gizli katmanındaki i’inci nöronu, w_{ij} gizli katmanındaki i’inci nörondan çıktı katmanındaki j’inci nörona iletilen ağırlığı, m gizli katmandaki nöron sayısını, f ise aktivasyon fonksiyonunu temsil etmektedir.

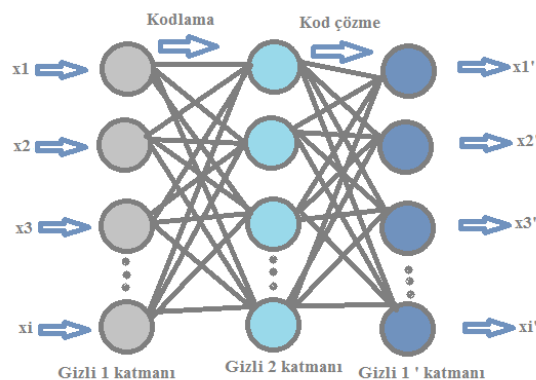
$$x_j' = f(\sum_{i=1}^m y_i * w_{ij}) \quad (2.3)$$

Oto kodlayıcı modelinde maksat, ilk iki aşama sonucunda üretilen x_j' değerinin, girdi katmanındaki x_j değerine benzer bir değer vermesidir. Bu iki değer birbirine yakın değerler olması için ağırlıklar geri yayılım algoritması aracılığı ile hesaplanıp sürekli olarak güncellenmelidir. Geri yayılım algoritması denklem 2.4'te verildiği gibi iki değer arasındaki farkın karesini minimize etmektedir.

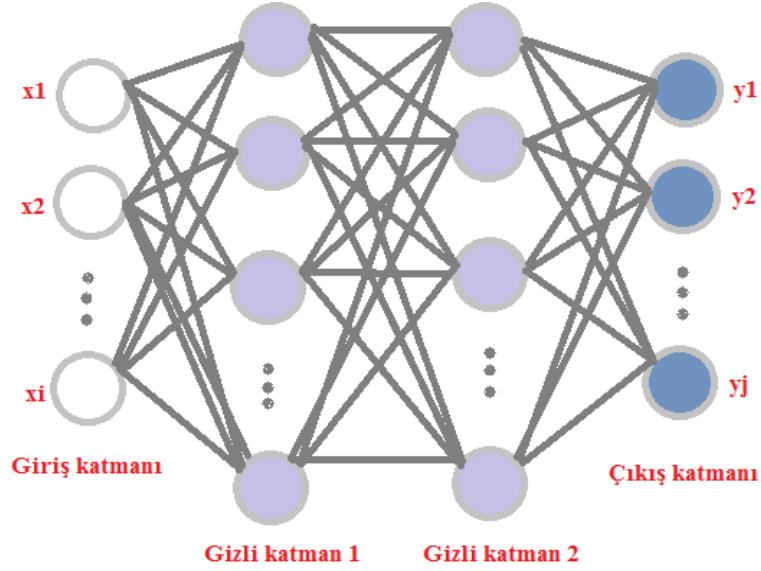
$$\min \sum (x_j' - x_j)^2 \quad i=1 \quad (2.4)$$

Derin öğrenme mimarilerinin tamamında olduğu gibi oto kodlayıcı modelinde de üstesinden gelinmesi gereken en önemli sorunlardan birisi ağırlık çok iyi bir şekilde eğitilmesidir. Aşırı eğitim ile eğitim seti için yüksek başarı oranı gösterilirken eğitim seti dışında kalan yeni veri setleri için aynı başarı gösterilememektedir. Bu soruna çözüm bulabilmek için birçok metot geliştirilmiştir. Bu metotlardan ilki iterasyon sayısına limit koymaktır. Öğrenme sırasında iterasyon sayısında artış sağlandıkça ağırlıklar eğitim veri setini sıfır hata verecek şekilde ezberleyip öğrenmektedir. Giriş veri setinde gürültü olması durumunda gürültünün de gerçek veri gibi modele dâhil edilmesine neden olacaktır. Başka bir metot ise, eğitim verisinin bir bölümünü doğrulama olarak kullanarak eğitim verilerinde ki hata oranı azaltılırken aynı şekilde geçerlilik verisindeki hata oranının azalıp azalmadığını kontrol edip eğitimin hızlı sonlandırılması hedeflenmektedir.

Şekil 2.12'de gizli katmanın giriş katmanı olarak alınması ve tekrar oluşturulması örneği verilmiştir. Şekil 2.13'te iki katmanlı yığın otomatik kodlayıcı örneği verilmiştir.



Şekil 2.12. Gizli katmanın giriş katmanı olarak alınması ve tekrar oluşturulması gösterimi



Şekil 2.13. İki katmanlı yığın otomatik kodlayıcı örneği

Yapay sinir ağlarında mevcut olan metotlar ile otomatik kodlayıcıların parametre güncellemesi sağlanmaktadır. Buradaki en önemli ayırt edici özellik gözetimli öğrenimde kullanılan etiket bilgisinden ziyade giriş katmanında bulunan verinin kullanılarak hatanın hesaplanmasıdır. Belirlenen hata, geri yayılım algoritması ve toplu eğitim inişi vb. optimizasyon metotları kullanılarak parametre güncellemelerinin yapılmasında kullanılır.

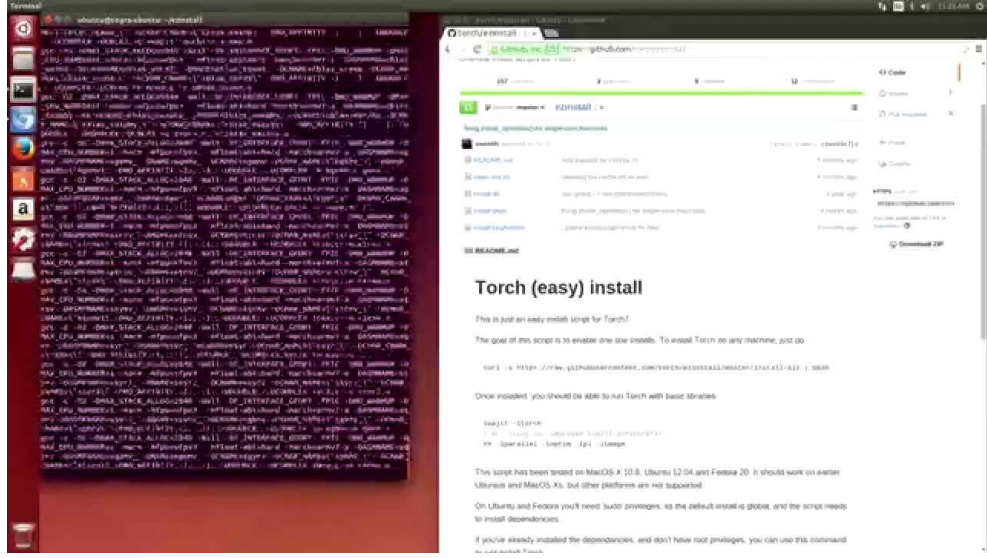
2.4.5. Derin Öğrenme Yazılım Araçları

Birçok yazılım aracı, derin öğrenme algoritmalarını bilgisayar diline çevirmek için kullanılmaktadır. Bu yazılım araçları Torch, Caffe, Tensorflow, MatConvNet, Deeplearning4j, NVIDIA, cuDNN ve Theano olarak verilmektedir.

2.4.5.1. Torch

Torch, GPU kullanmak için önde gelen makine öğrenme algoritmaları için bilimsel bir hesalama modelidir. Torch basit ve hızlı bir kodlama dili olan LuaJIT dili aracılığı ile ve C/CUDA temelli bir uygulama, kullanımı kolay bir dil yapısına sahip ve etkili olan açık kodlu bir kütüphanedir.

Şekil 2.14'te Torch aracına ait bir ekran görüntüsü verilmiştir.



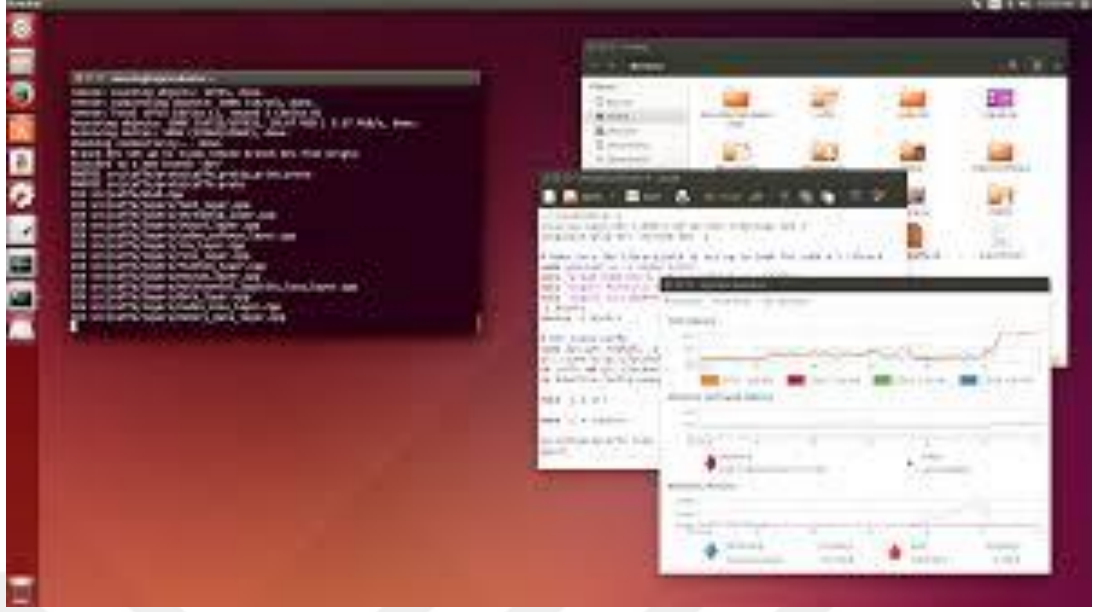
Şekil 2.14.. Torch ekran görüntüsü

Torch kullanımının temel amacı, bilimsel algoritmaların oluşturulmasında maksimum hıza ve esnekliğe sahip bir yapı göstermekle birlikte işlevleri oldukça basitleştiren bir yapıya sahiptir. Torch makine öğrenmesi, bilgisayar görmesi, sinyal işleme, ses işleme ve görüntü işleme gibi alanlarda geniş çapta desteklenen paket programlar sunmaktadır. Facebook, Google, Twitter, New York Üniversitesi ve daha bir çok şirket ve araştırma laboratuvarları tarafından kullanılmaktadır (CollobertRonan ve ark., 2017).

2.4.5.2. Caffe

Caffe, tasarım ve hız kavramları ile yapılan bir derin öğrenme modelidir. Bu model görü ve öğrenme merkezi ve topluluk katılımcıların desteğiyle geliştirilmiştir. Yangqing Jia bu projeyi California Berkeley Üniversitesi'nde doktorasını yaparken yapmıştır. BYD lisansı altında yayınlanmıştır.

Şekil 2.15'te Caffe aracına ait bir ekran görüntüsü verilmiştir.



Şekil 2.15. Caffe ekran görüntüsü

2.4.5.3. TensorFlow

Tensorflow, veri akış grafiklerini kullanarak numerik hesaplamalar yapabilen açık kaynak yazılım kütüphanesidir. Grafikte bulunan düğümler matematiksel işlemleri temsil etmektedir. Grafik kenarları ise grafikler arasındaki bağlantıyı sağlayan çok boyutlu veri dizilerine karşılık gelmektedir. TensorFlow, esnek yapısından dolayı bir UPA ile bir ya da daha çok GİB ile ilgili hesaplamaları yapabilmektedir. TensorFlow başlangıçta Google Machine Intelligence grubu içerisinde faaliyet gösteren Google Beyin takımında çalışan araştırmacılar ve mühendisler tarafından makine öğrenmesi ve derin sinir ağları araştırmalarını harmanlamak amacıyla geliştirildi. Diğer birçok alanda da genel olarak uygulanabilir seviyeye ulaşmıştır (URL2, 2017).

Şekil 2.16'da TensorFlow aracına ait bir ekran görüntüsü verilmiştir.

```
ap output locations for shuffle 0 to lappy.0xdata.loc:55853
16/06/13 17:46:23 INFO MapOutputTrackerMaster: Size of output statuses
for shuffle 0 is 175 bytes
16/06/13 17:46:23 INFO MapOutputTrackerMaster: Size of output statuses
for shuffle 0 is 175 bytes
16/06/13 17:46:23 INFO TaskSetManager: Finished task 2.0 in stage 5.0
(TID 46) in 5641 ms on lappy.0xdata.loc (1/3)
16/06/13 17:46:28 INFO TaskSetManager: Finished task 1.0 in stage 5.0
(TID 45) in 5672 ms on lappy.0xdata.loc (2/3)
16/06/13 17:46:28 INFO TaskSetManager: Finished task 0.0 in stage 5.0
(TID 44) in 5679 ms on lappy.0xdata.loc (3/3)
16/06/13 17:46:28 INFO TaskSchedulerImpl: Removed TaskSet 5.0, whose t
asks have all completed, from pool.
16/06/13 17:46:28 INFO DAGScheduler: ResultStage 5 (collect at <ipytho
n-input-7-12c43c5da018>:23) finished in 5.679 s
16/06/13 17:46:28 INFO DAGScheduler: Job 4 finished: collect at <ipyth
on-input-7-12c43c5da018>:23, took 10.428071 s
[17:46:29.815 NotebookApp] Saving file at /TensorFlowDeepLearning.ip
ynb
[17:46:29.815 NotebookApp] Saving untrusted notebook TensorFlowDeepL
earning.ipynb

total 222
-rw-r--r-- 1 arno staff 367B
drwxr-xr-x 7 arno staff 238B
drwxr-xr-x 14 arno staff 476B
-rw-r--r-- 1 arno staff 12K
-rw-r--r-- 1 arno staff 11K
-rw-r--r-- 1 arno staff 34K
-rw-r--r-- 1 arno staff 11K
drwxr-xr-x 6 arno staff 204B
drwxr-xr-x 3 arno staff 102B
drwxr-xr-x 5 arno staff 170B
drwxr-xr-x 6 arno staff 204B
drwxr-xr-x 476 arno staff 15K
-rw-r--r-- 1 arno staff 34K
.invnh

def train_nn(iterations, batch_size, use_tf_mnist=False):
def perPartition(it):
if not use_tf_mnist:
train_data = RowData(it)
test_data = train_data
else:
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets('MNIST_data', one_hot=True)
train_data = mnist.train
test_data = mnist.train

return create_nn(train_data, test_data, iterations, batch_size)

return perPartition

coeffs_per_node = train_df.mapPartitions(train_nn(ITERATIONS, BATCH_SIZE, USE_TF_MNIST)).collect

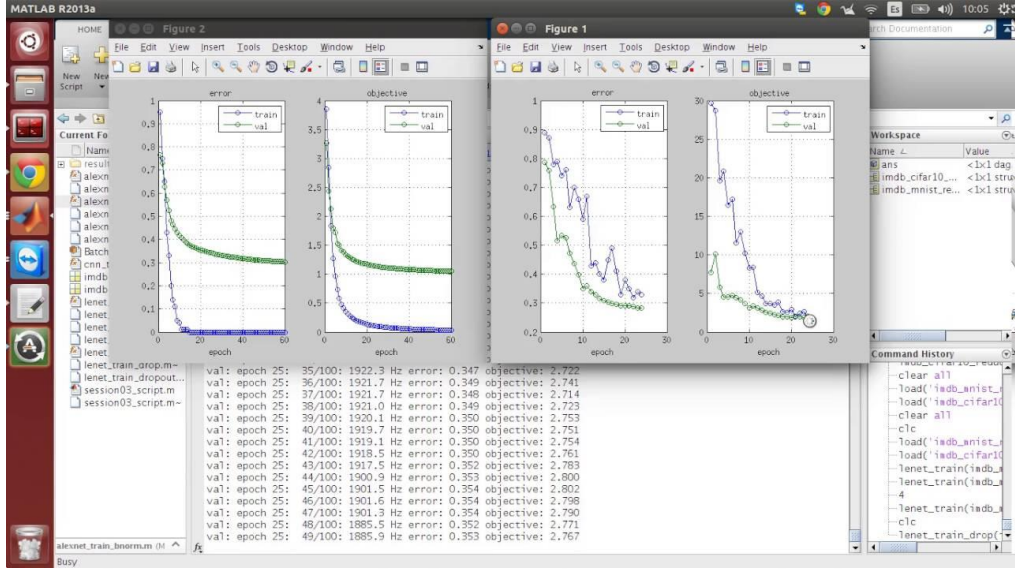
In [7]: # Now, we have the weights and biases for each node
print(len(coeffs_per_node)) ## Number of nodes
print(len(coeffs_per_node[0])) ## Number of weight and bias arrays
```

Şekil 2.16. Tensorflow ekran görüntüsü

2.4.5.4. MatConvNet

MatConvNet BYD lisansı altında yayınlanan açık kaynaklardan biridir. MatConvNet konvolüsyon, normalizasyon ve pooling gibi konvolüsyonel sinir ağı oluşturma bloklarını MATLAB komutlarına benzer şekilde basit bir dil yapısı ile sunmaktadır. Konvolüsyonel sinir ağı mimarilerinin oluşturulması için bu bloklar bir araya getirilip genişletilebilmektedir. Benzer blokların çoğu C++ ve CUDA ile yazılan optimize edilmiş MİB ve GİB uygulamalarını kullanmaktadır. Diğer taraftan GİB hesaplamaya yönelik MATLAB'ın doğal desteği, MATLAB ortamında yeni blokların yazılmasının hesaplamayla ilgili verimliliği korumak ile mümkün olduğu anlamına gelmektedir. Düşük seviyeli diller kullanarak yeni konvolüsyonel sinir ağı bileşenlerini yazmaya karşılık, yeni fikirleri test etmeyi önemli ölçüde hızlandırıp basitleştirmektedir. MATLAB kullanımını diğer alanlar ile ortak çalışma ortamı hazırlamaktadır. Örneğin; MatConvNet son zamanlarda Arizona Üniversitesinde gezegen bilimi alanında kullanılmıştır (Vedaldi ve Lenc, 2015).

Şekil 2.17'de MatConvNet aracına ait bir ekran görüntüsü verilmiştir.



Şekil 2.17. MatConvNet ekran görüntüsü

2.4.5.5. Deeplearning4j

Deeplearning4j (DL4J) Java ve Scala için yazılmış tecimsel nitelikli açık kaynak dağıtık derin öğrenme kütüphanesidir. Hadoop ve Spark ile entegre olan DL4J dağıtık MİB ve GİB'ler üzerinde iş ortamlarında kullanılmak üzere tasarlanmıştır. Ticari desteği Skymind şirketi tarafından sağlanmaktadır. Apache 2.0 lisansı altında yayınlanmıştır. DL4J dolandırıcılık tespiti, ağ ihlali tespiti, yüz/imge tanıma, ses arama, sesi metine dönüştürme gibi alanlarda kullanılabilir. DL4J sinir ağı modellerini Keras, TensorFlow, Caffe, Torch ve Theano gibi en çok gelişmiş sistemlerden aktarabilmektedir. DL4J'ler farklı sığ ağlardan her biri katman diye adlandırılan derin sinir ağlar oluşturabilme olanağı sağlamaktadır. DL4J'lerin esnekliği Spark ve Hadoop ile MİB ve GİB'ler üzerinde çalışan dağıtık ve üretim nitelikli bir yapıda ihtiyaç duyulduğu gibi KBM, diğer otomatik kodlayıcılar, konvolüsyonel sinir ağı ya da tekrarlı ağları birleştirebilme olanağı sunmaktadır (URL3, 2017).

Şekil 2.18'de Deeplearning4j aracına ait bir ekran görüntüsü verilmiştir.



Şekil 2.18. DeepLearning4j ekran görüntüsü

2.4.5.6. NVIDIA cuDNN

NVIDIA cuDNN (The NVIDIA CUDA® Deep Neural Network library), derin sinir ağları için GİB hızlı kütüphanedir. cuDNN; ileri ve geri konvolüsyon, örnek seçme, normalizasyon ve aktivasyon katmanları gibi standart rutinler için yüksek ayarlı uygulamaları sağlamaktadır. cuDNN NVIDIA derin öğrenme yazılım geliştiricinin bir parçasıdır. Derin öğrenme araştırmacıları sistem geliştiricileri yüksek performanslı GİB hızlandırma için cuDNN'e dayanmaktadır. cuDNN'de düşük seviyeli GİB performansı ayarlama zamanı harcamaktan ziyade sinir ağlarını eğitmeye ve yazılım uygulamaları geliştirmeye odaklanılmıştır. cuDNN; caffe, torch, theano ve tensorflow gibi yaygın olarak kullanılan derin öğrenme yapılarını hızlandırabilmektedir. Veri bilimcileri ve araştırmacıları derin öğrenme yapılarını ya da NVIDIA DIGITS indirerek cuDNN'den faydalanabilmektedir. DIGITS interaktif olarak veri yönetebilme ve eğitimi çoklu GİB'ler üzerinde gerçekleştirebilme olanağı sunmaktadır (URL4, 2017) .

2.4.5.7. Theano

BYD lisansı altında yayınlanan açık kaynak kodlu bir yapıdır ve MILA grubu tarafından Yoshua Bengio bulunduğu Montreal Üniversitesinde geliştirilmiştir. Adını Yunan bir matematikçiden almaktadır.

Theano çok boyutlu dizileri içeren matematiksel ifadelerin tanımlanması, optimize edilmesi ve değerlendirilmesine olanak sağlayan ve NumPy kütüphanesini kullanan bir Python kütüphanesidir. 2007'den beri geniş ölçekli hesaplama açısından yoğun bilimsel araştırmaları desteklemektedir. Derin öğrenmedeki büyük sinir ağları algoritmaları için gereken hesaplamaların yapılabilmesi için tasarlanmıştır. Kendi türünün ilk kütüphanelerinden biridir. Derin öğrenme araştırma ve geliştirme alanında endüstriyel standartlardan biridir. Özelliklerini inceleyecek olursak;

- Dinamik C kodu oluşturma olanağı sunması sebebi ile ifadeleri daha hızlı değerlendirmektedir.
- Geniş birim testi ve kendini doğrulama özelliği ile çoğu hatayı tespit ve teşhis edebilmektedir.
- Doğru yanıt alınmasını hız ve kararlılık optimizasyonu ile sağlamaktadır.
- Etkili sembolik farklılaşma sunar, yani bir ya da daha fazla girişli istenilen fonksiyonların türevini alır.
- GPU kullanımını yeterince anlaşılır olup MİB'ne göre 140 kat daha hızlı veri yoğunluklu hesaplamaları gerçekleştirebilmektedir (URL4, 2017)-(URL5, 2017).

2.4.5.8. YOLO

YOLO tam görüntüye tek bir sinir ağı uygulayarak görüntü içerisindeki nesnelerin tespitini yapmaktadır (Rendom, 2016). Bu ağ, görüntüyü segmentasyon yaparak her bölgenin sınır kutularını ve olasılıklarını tahmin eder. Bu model sınıflandırıcı tabanlı sistemlere göre birçok avantaja sahiptir. Model test etme sürecinde tüm görüntüyü inceler, bu şekilde tahmini görüntünün tamamından yapar. Ayrıca tek bir görüntü için binlerce ağ gerektiren, R-CNN gibi sistemlerin aksine, tek bir ağ değerlendirilmesiyle tahminleri yapar. Bu model son derece hızlıdır. R-CNN 'den 1000 kat ve Fast R-CNN 'den 100 kat daha hızlıdır. . YOLO'nun yapısı oldukça basittir. 24 konvolüsyon katmanından ve bununla

birlikte 2 tamamen bağı katmandan oluşmaktadır. Ağın aktivasyon işlevi çoklu sınırlayıcı kutular ve buna tekabül eden sınıf olasılıkları eş zamanlı olarak tek bir konvolüsyonel sinir ağı tarafından tahmin edilmiştir(4).

Geleneksel nesne tanıma yöntemleriyle karşılaştırıldığında YOLO'nun bir çok avantajı vardır.

- Birincisi, YOLO çok hızlıdır. Regrasyon sorununun algılanmasını çerçevelediğinden dolayı algoritmanın karmaşık bir hatta ihtiyacı yoktur.
- İkincisi, 2 aşamalı yöntemlerden daha çok gelişmiş olan YOLO, kutu sınırlarını ve etiketleri eş zamanlı olarak tespit ediyor.
- Üçüncüsü, YOLO nesnelerin genelleştirilebilir temsillerini ve görüntü hakkında evrensel sebeplerini öğrenir. Beklenmedik girdiler veya tamamen farklı etki alanı söz konusu olduğunda, YOLO' nun başarısız olması olasılığı azdır.

YOLO önceden eğitilmiş nesne tespit aracıdır. Belirli bir görüntüdeki nesnelere bulup bu nesnelerin ne olduğunu belirler. Geleneksel nesne tespit araçları kare başına ayrı ayrı yalnızca bir nesneyi algılayabilmektedir. Buna karşın YOLO bir karede birden çok nesneyi algılayabilen ve konumunu belirleyebilen bir sinir ağı modelidir (URL12, 2017).

2.4.5.8.1. Nesne Tespiti

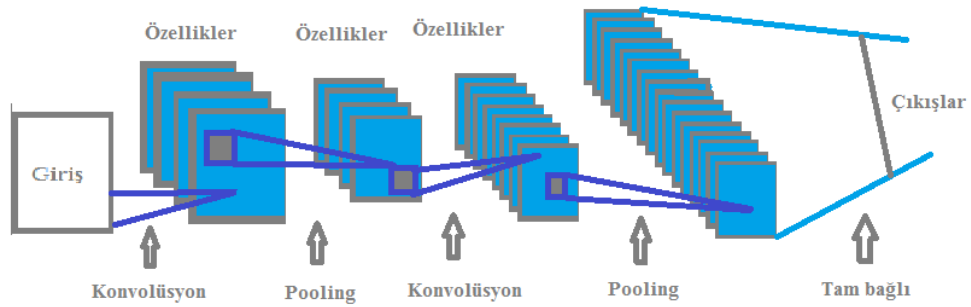
Nesne tespiti, belirli bir nesnenin sahne içerisindeki konumunun belirlenmesidir. Bu amaçla, hareketli bir nesnenin zaman ekseninde ardışık görüntülerdeki pozisyonunu ve oryantasyonunu saptamayı hedefler. Hareket analizi, sınıflandırılan nesne ya da nesnelerin hareketlerinin tanımlanması ve nesnenin görüntü içinde devam eden ilişkilerinin otomatik olarak saptanmasını hedeflemektedir (URL11, 2013).

2.5. Konvolüsyonel Sinir Ağları

Konvolüsyonel ağlar, bilinen ızgara benzeri bir topolojiye sahip verileri işlemek için özel bir sinir ağı türüdür. Konvolüsyonel ağlar, pratik uygulamalarda mükemmel derecede başarılı olmuştur. “Konvolüsyonel sinir ağı “ ismi ağın konvolüsyon denilen matematiksel bias işlemi kullandığını gösterir. Kıvrım, özel bir doğrusal işlem türüdür. Konvolüsyonel

ağlar, katmanların en azından birinde genel matris çarpımı yerine konvolüsyon sinir ağlarıdır.

Konvolüsyonel sinir ağları, ses, görüntü ve yapay zeka uygulamalarında sıkça kullanılan yöntemlerden biridir. KSA, birden fazla katmandan oluşan ve bu katmanlarda doğrusal olmayan fonksiyonların bulunduğu modellerdir. Geleneksel ileri beslemeli sinir ağlarında bir katmandaki bütün nöronlar bir sonraki katmanın bütün nöronlarına bağlıdır. Bu tür bağlı katmanlara tam bağlı katman denir. KSA'larda, tam bağlı katmanlara ek olarak, çıkışı üretmesi için giriş katmanına konvolüsyon işlemi uygulanır. Bu, giriş katmanındaki bütün bölgelerin bir sonraki katmanda bulunan nöronlara bağlı olduğu, yerel bağlantılara neden olur. Her katman farklı filtreler uygular ve sonunda bunların sonuçlarını birleştirir. Eğitim boyunca KSA'lar, filtrelerin içerdiğini (öznitelikleri), gerçekleştirilmesi istenen işleme göre kendisi otomatik olarak öğrenir. Böylece baştan sona doğru her katmanda veriye ilişkin, basitten karmaşığa, doğru öznitelikler öğrenilmiş olur. Şekil 2.19' da basit bir KSA yapısı örneği verilmiştir.



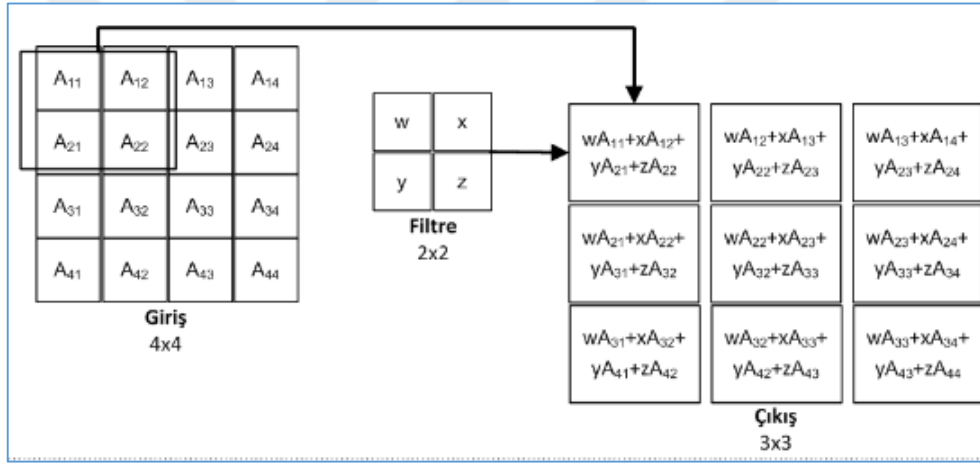
Şekil 2.19. Basit bir KSA yapısı örneği

Konvolüsyonel sinir ağları, görüntü sınıflandırmada büyük bir başarı gösterdiğinden geniş çapta benimsenmiştir. Konvolüsyonel sinir ağları karmaşık ve yoruma dayalı görüntü özelliklerini öğrenmede önemli başarı gösterdiğinden (Zeiler ve ark., 2014), günümüzde git gide daha fazla araştırmacı görüntü sınıflandırma ve benzeri alanlarda Konvolüsyonel sinir ağlarını kullanmaya başlamışlardır. Konvolüsyonel sinir ağı modelini anlamadaki zorluk sadece statik karelerde bulunan bilgileri değil, aynı zamanda karmaşık zamansal gelişimi anlamaktır. Arka plan karışıklığı, bakış açısı değişiklikleri ve gözlemlenen hareketteki dinamikler önemli derecede farklılık gösterdiğinden görüntü alırken bunlara dikkat edilerek

sınıflamak gerekir aksi halde başarılı bir şekilde sınıflamak oldukça zordur (Soomro ve ark., 2012).

2.5.1. Konvolüsyon Katmanı

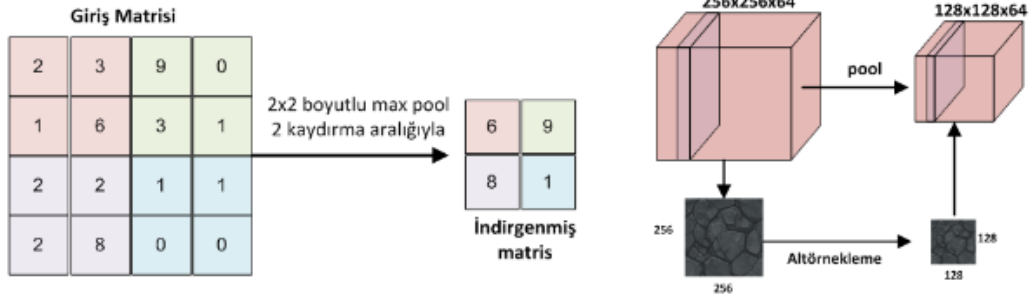
Konvolüsyonel sinir ağı kavramı, konvolüsyon işleminden türetilmiştir. Genel tanımıyla matris üzerine kayan bir pencere fonksiyonunun uygulanmasıdır. Konvolüsyonel sinir ağlarının en temel amacı giriş verisine ait özelliklerin çıkartılmasıdır. Konvolüsyon, giriş verilerinin küçük boyutlarını kullanarak verilerin özelliklerini öğrenme ve piksel değerleri arasındaki ilişkiyi korumaktır. Şekil 2.20’ de iki boyutlu konvolüsyon işlemi gösterimi verilmiştir.



Şekil 2.20. İki boyutlu konvolüsyon işlemi gösterimi

2.5.2. Havuzlama Katmanı

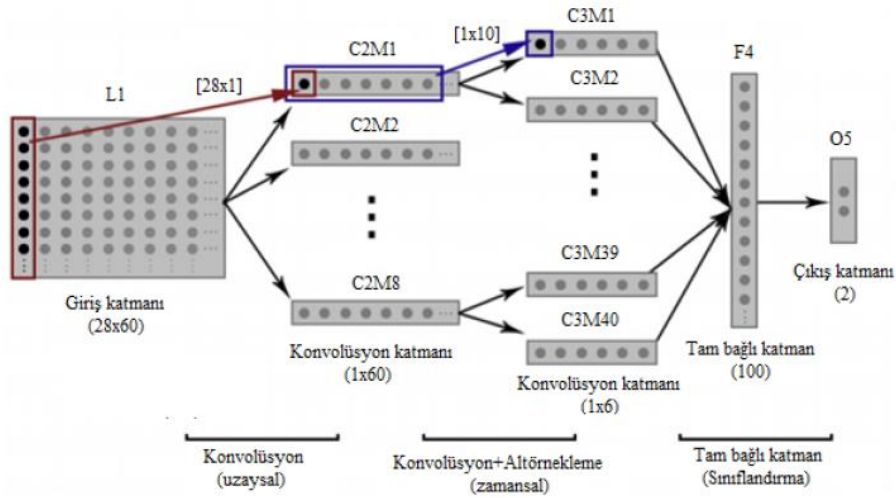
Konvolüsyonel sinir ağlarında havuzlama katmanları konvolüsyon katmanlarından sonra gelmektedir. Havuzlama katmanları girdi olarak verilen verilerin alt örnekleme işlemlerini yapar. Havuzlama işleminin sonucunda da sabit boyutlu bir matris elde edilmektedir. Burada çıkış boyutu önemli ölçüde düşürülür. Şekil 2.21’ de giriş matrisi üzerinde maxpooling işlemi ve görüntü üzerinde örnek gösterilmiştir.



Şekil 2.21. Giriş matrisi üzerinde maxpooling işlemi ve görüntü üzerinde örnek

2.5.3. Konvolüsyonel Sinir Ağı Matematiği

Konvolüsyonel sinir ağlarının matematiksel olarak anlaşılması amacıyla Tank ve arkadaşlarının yaptığı çalışmadaki yapı göz önünde bulundurulmuştur (Tang ve ark., 2017). Şekil 2.22’ de 5 katmanlı KSA yapısı verilmiştir.



Şekil 2.22. 5 katmanlı KSA yapısı (Tang ve ark., 2017)

Konvolüsyonel sinir ağlarında bir nöron $n(l, m, j)$ ile ifade edilir. Burada l katmanı, m özellik haritasını ve j bu özellik haritasındaki pozisyonu temsil etmektedir. Giriş ve çıkıştaki her bir nöron $x_m^l(j)$ ve $y_m^l(j)$ olur.

$$y_m^l(j) = f(x_m^l(j)) \quad (2.5)$$

Denklem 2.5'te f() aktivasyon fonksiyonuna karşılık gelmektedir. Hiperbolik tanjant fonksiyonu ilk iki gizli katman olan C2 ve C3 için aktivasyon fonksiyonu olarak kullanılır.

$$f(x)=\alpha \tan h (bx) \quad (2.6)$$

Denklem 2.6'da (LeCun, Bottou, Orr, & Müller, 2012) çalışmasına göre a=1.7159 ve b=2/3 olarak alınır. Sigmoid fonksiyonu son iki katman için aktivasyon fonksiyonu olarak kullanılır.

$$f(x)=\frac{1}{1+exp-x} \quad (2.7)$$

C2 ve C3 katmanlarındaki haritaların her bir nöronu aynı ağırlıklar dizisini paylaşmaktadır. Bu katmanlardaki nöronlar önceki katmandan nöronlar alt kümesine bağlıdır. Farklı katmanlardaki nöronlar arasındaki bilgi iletimi aşağıdaki gibi gerçekleşmektedir.

L1 katmanı için: Giriş matrisi $I_{N,T}$ ile temsil edilebilir.

C2 katmanı için: Konvolüsyon işlemi ve aktivasyon fonksiyonu vasıtasıyla bir önceki katmanın giriş özellik haritası çıkış özellik haritasına dönüştürülür.

$$y_m^2=f(\sum_{i=1}^{i \leq 28} I_{i,j} x k_m^2 + b_m^2(j)) \quad (2.8)$$

k_m^2 : [28 * 1] boyutlu bir konvolüsyonel çekirdeği $b_m^2(j)$: bias'ı temsil etmektedir.

C3 katmanı için: Bu katman C2 katmanına benzemektedir.

$$y_m^3=f(\sum_{i=1}^{i \leq 10} y_m^2[(j-1) \times 10 + i] x k_m^3 + b_m^3(j)) \quad (2.9)$$

Denklem 2.9'da k_m^3 : [1 x 10] boyutlu bir konvolüsyonel çekirdeği $b_m^3(j)$: bias'ı temsil etmektedir.

F4 katmanı için: Bu katman tamamen C3 katmanına bağlıdır.

$$y^4=f(y_i^3(p)w_i^4(p)+b^4(j)) \quad (2.10)$$

Denklem 2.10'da w_i^4 (p) C3 ve F4 katmanlarındaki nöronların ağırlık değeri ve b^4 (j): bias'a karşılık gelmektedir.

$$y^5 = f\left(\sum_{i=1}^{100} y^4(i)w^5(i) + b^5(i)\right) \quad (2.11)$$

Denklem 2.11'de w^5 (i) : F4 ve O5 katmanlarına ait nöronların ağırlık değeri ve b^5 (j) : bias'ı temsil etmektedir.

Her bir nöron için giriş ağırlıkları ve eşik değeri standart bir dağılım ile $\left[\pm \frac{1}{N_{n(l,m,i)}^{giris}} \right]$ civarında başlatılır. Burada ; $N_{n(l,m,i)}^{giris}$ n(l,m,i) girişinin sayısını ifade etmektedir. C2 ve C3 katmanlarına ait n_1 öğrenme oranı 2.12 ile tanımlanabilir.

$$\frac{2\lambda}{N_{n(l,m,o)}^{paylaşılan} \sqrt{N_{n(l,m,i)}^{giris}}} \quad (2.12)$$

2.12'de $N_{n(l,m,o)}^{paylaşılan}$ aynı ağırlık dizisi değerlerini paylaşan nöronların sayısına karşılık gelmektedir ve λ sabit katsayıdır. F4 ve O5 katmanlarına ait n_2 öğrenme oranı (2.13) ile tanımlanır.

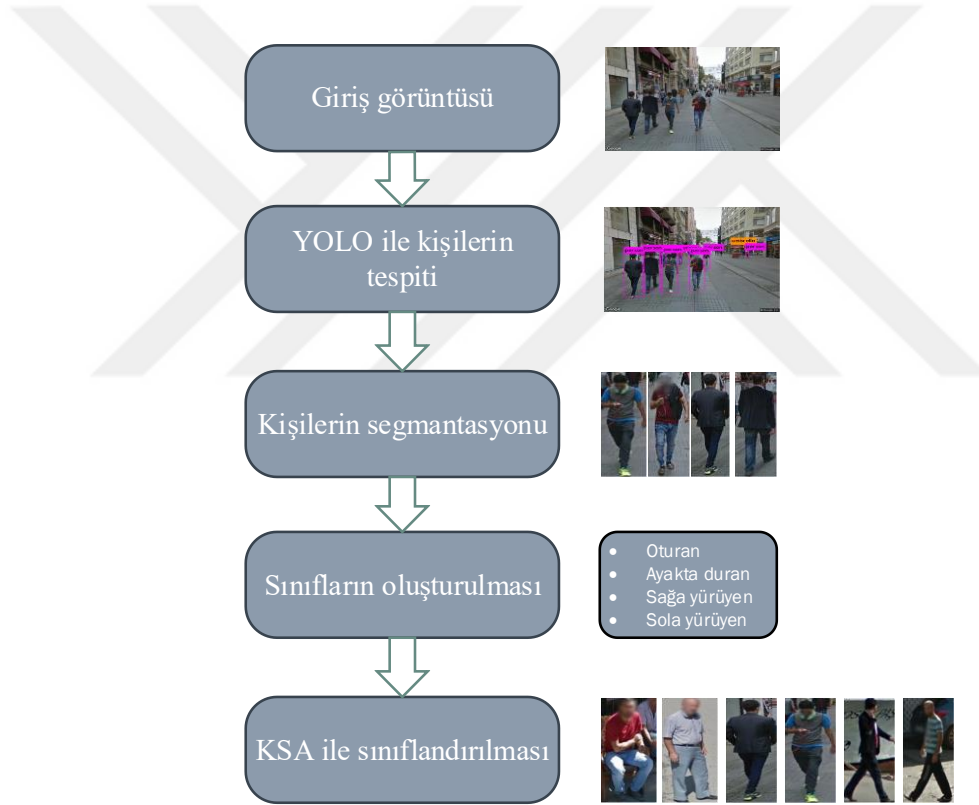
$$n_2 = \frac{\lambda}{\sqrt{N_{n(l,m,i)}^{giris}}} \quad (2.13)$$

Eğim iniş yöntemi ağırlıkları ve konvolüsyonel sinir ağlarının biasını ayarlamak ve düzeltmek için kullanılır.

3. BULGULAR

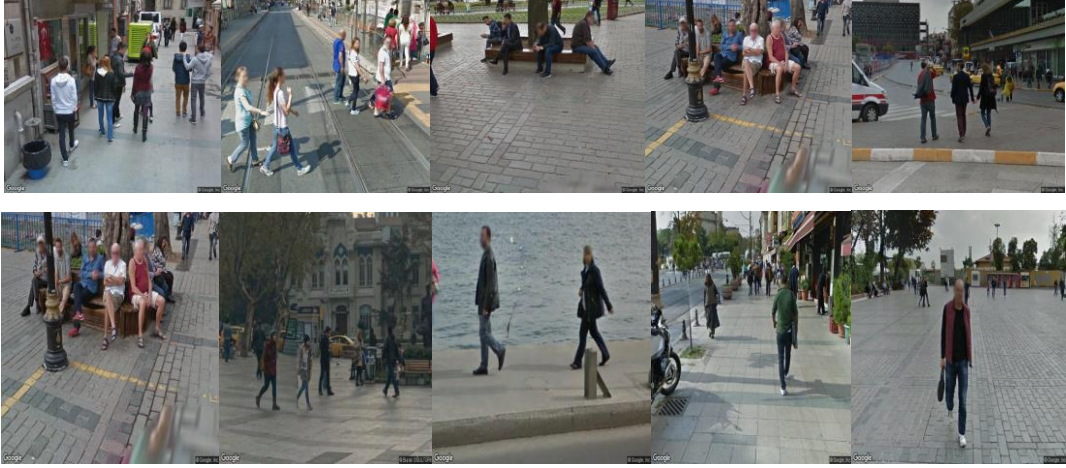
3.1. Dış Ortam Görüntülerinden İnsan Hareketi Tespiti

Bu tez çalışmasında, dış ortam görüntülerinden insan hareketlerinin tespiti için derin öğrenme tabanlı bir tanıma sistemi geliştirilmiştir. Bu tanıma sistemi ile Google Streetview uygulamasından elde edilen görüntülerdeki insanların hareketlerinin belirlenmesi sağlanmıştır. Tez kapsamında geliştirilen sistemin çalışma adımları Şekil 3.1’de gösterildiği gibidir.



Şekil 3.1. Geliştirilen sistemin çalışma adımları görüntüsü

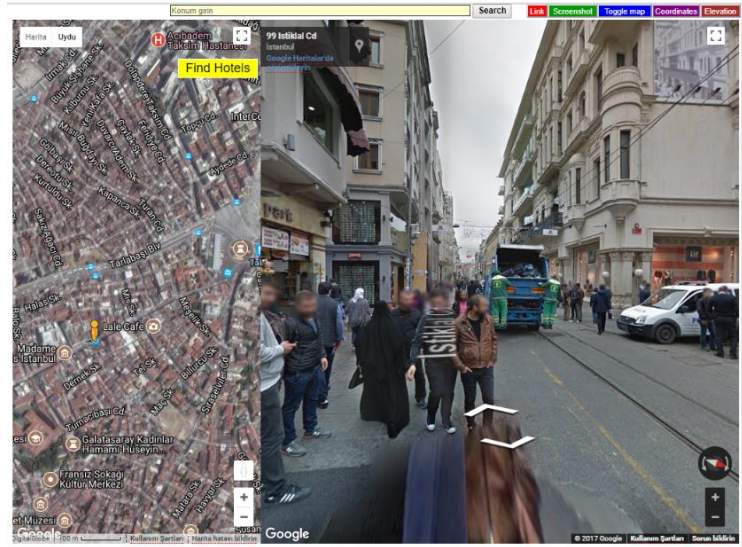
Google Street view uygulamasından elde edilen bazı görüntüler Şekil 3.2.’de verilmiştir. Öncelikle görüntülerin elde edilmesi ve bu görüntüler için bir veri tabanı oluşturulması sağlanmıştır. Daha sonraki aşamada bu görüntüler nesne tespit aracı olan YOLO ile tespit edilmiştir. Tespit edilen görüntüler için sınıf ayırımı yapılmıştır. Bu sınıflar; sağa, sola yürüyen, ayakta duran ve oturan şeklinde belirlenmiştir.



Şekil 3.2. Google street view uygulamasından elde edilen görüntüler

3.1.1. Görüntü Veri Setleri

Google'ın diğer ülkelerinde uzun zamandır kullanılan uygulaması street view özelliği, 22 Ekim 2015 tarihinde Türkiye'ye gelmiştir. Bu uygulama ile sokakları 360 derece görüntüleme imkanı sunan google bu hizmetini 2,5 saniyede de fotoğraf çekerek sunmaktadır. 15 kamerası olan cihaz tarafından ileri seviye GPS vasıtasıyla hazırlanan görüntüler kullanıma sunulmaktadır. Şekil 3.3'de, çalışmada kullanılan street view ekran görüntüsü verilmiştir.



Şekil 3.3. Google street view uygulama görüntüsü

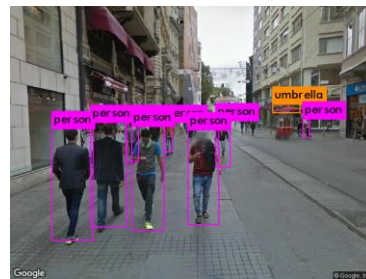
Bu çalışmada, Google street view uygulamasından toplamda 1000 adet görüntü alınarak kapsamlı bir veri tabanı oluşturulmuştur. Bu görüntüler içerisindeki kişiler nesne tespit aracı olan YOLO ile tespit edilip bölümlenmiştir. Böylece 2000 adet kişi görüntüsü elde edilmiştir. Elde edilen kişi görüntülerin boyutlarına bakılarak belli bir eşik değeri belirlenmiştir. Belirlenen yükseklik ve genişlik ölçülerine uygun olmayan görüntüler çıkarılıp tekrardan veritabanı güncellenmiştir. Son olarak güncellenen veritabanındaki görüntülerin ortalama genişlik ve yüksekliklerine göre üç farklı boyut için çalışmalar yapılmıştır. Bu veri kümeleri; yükseklik \times genişlik \times renk olarak $192 \times 72 \times 3$, $128 \times 64 \times 3$ ve $96 \times 36 \times 3$ şeklinde düzenlenmiştir.

3.1.2. Kişilerin Tespit Edilmesi

Orijinal görüntüler içerisindeki kişilerin tespit edilmesi için YOLO aracı kullanılmıştır. YOLO önceden eğitilmiş bir nesne tespit aracıdır. Belirli bir görüntüdeki nesnelere bulup bu nesnelere ne olduğunu belirler. Google street view uygulamasından alınan bir görüntü Şekil 3.4 (a)' da verilmiştir. Nesne tespit aracı olan YOLO ile bu görüntüdeki kişilerin tespiti ise Şekil 3.4 (b)' de verilmiştir. Tespit edilen görüntülerin parçalanmış halleri ise Şekil 3.4 (c)' de verilmiştir.



a. Orijinal görüntü



b. Kişilerin tespit edilmesi



c. Kişilerin bölümlenmiş halleri

Şekil 3.4. Google Street view uygulamasından alınan görüntülerin YOLO ile tespit edildikten sonra parçalanmış halleri

Sonraki adımda YOLO ile tespit edilen kişilerin parçalanmış olan görüntüleri için oturan, ayakta duran ve sağa, sola yürüyen şeklinde 4 sınıf oluşturulmuştur. Bu sınıflar için elde edilen bazı görüntüler Tablo 3.2.'de verilmiştir.

3.1.3. Konvolüsyonel Sinir Ağı Modeli

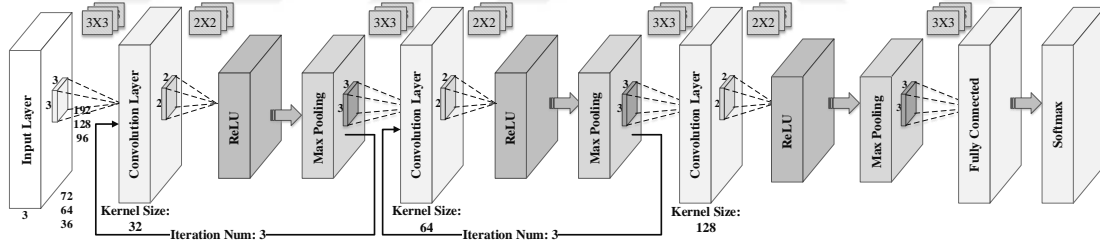
Sınıfların otomatik olarak belirlenebilmesi için bir konvolüsyonel sinir ağı yapısı geliştirilmiştir. Veritabanındaki tüm görüntüler YOLO sistemi ile parçalandıktan sonra insan hareketlerinin sınıflandırılması için konvolüsyonel sinir ağları kullanılmıştır. Belirlenen sınıflar için eğitim ve test veri setleri oluşturulmuştur. Sınıflarda bulunan verilerin bir kısmı eğitim bir kısmı test için kullanılmıştır. Bu oranlar değiştirilerek en yüksek performansın oluşumu hedeflenmektedir. Bu çalışmamızda önerdiğimiz model ile diğer modellerin verileri üzerindeki başarımları karşılaştırılmıştır.

3.1.3.1. Önerilen Konvolüsyonel Sinir Ağı Modeli

Verilerin otomatik bir şekilde sınıflandırılması için aşağıdaki konvolüsyonel sinir ağı modeli kullanılmıştır. Önerilen derin KSA modelinin kod gösterimi Tablo 3.1.'de gösterilmiştir. Model giriş katmanı, 7 adet iki boyutlu konvolüsyon, 7 adet MaxPooling katmanı, 8 adet aktivasyon katmanı ve üç adet tam bağlı katman olmak üzere 25 katmandan oluşmaktadır. Önerilen derin KSA modelinin blok gösterimi Şekil 3.5.'te gösterilmiştir.

Tablo 3.1. Önerilen Derin KSA Modelinin kod gösterimi

```
model.add(Conv2D(32, (3, 3), input_shape=(192, 72, 3), padding='SAME'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='SAME'))
model.add(Conv2D(32, (3, 3), padding='SAME'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='SAME'))
model.add(Conv2D(32, (3, 3), padding='SAME'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='SAME'))
model.add(Conv2D(64, (3, 3), padding='SAME'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='SAME'))
model.add(Conv2D(64, (3, 3), padding='SAME'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='SAME'))
model.add(Conv2D(64, (3, 3), padding='SAME'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='SAME'))
model.add(Conv2D(128, (3, 3), padding='SAME'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='SAME'))
model.add(Flatten())
model.add(Dense(2048))
model.add(Dense(2048))
model.add(Dense(4))
model.add(Activation('softmax'))
```



Şekil 3.5. Önerilen Derin KSA Modelin blok gösterimi

3.1.3.2. Vggnet Modeli

Visual Geometry Group (VGG) çalışma grubu tarafından, karmaşık görüntüleri tanımda kullanılmak üzere geliştirmiş oldukları 16 ve 19 katmanlı KSA modelleridir (Simonyan ve ark., 2014). VGG 1000 sınıflı görüntü işleme için geliştirilmiştir. Bu çalışmada ImageNet ILSVRC-2014 yarışması için görüntü sınıflandırma ve yerleştirme alanında yarışmışlar ve birinciliği almışlardır.

3.1.3.3. Lenet Modeli

Lenet modeli (LeCun ve ark., 1998) rakam tanıma için geliştirilmiştir derin bir modeldir. Bu modelin farklı katmanlara sahip örnekleri bulunmaktadır. 10 rakamı tanımak için geliştirilmiş fakat daha sonra nesne tanımadada da başarı gösterdiği görülmüştür.

3.1.4. Deneysel Çalışmalar

Tüm veritabanları eğitim, doğrulama ve test olmak üzere belli oranlarda üç parçaya bölünmüştür. Bu bölümlenme yapıları şu şekildedir;




%80 eğitim, %10 doğrulama ve %10 test,

%70 eğitim, %15 doğrulama ve %15 test,

%50 eğitim, %25 doğrulama ve %25 test

Deneysel çalışmalarda kullanılan sınıflar ve bu sınıflara ait bazı örnek görüntüler Tablo 3.2’de verilmiştir.

Tablo 3.2. Çalışmada kullanılan sınıflar ve bu sınıflara ait örnek görüntüler

Sınıf	Sınıf görüntü örnekleri	Veri sayısı
1) Oturan		198
2) Ayakta		234
3) Sağa yürüyen		228
4) Sola yürüyen		249

3.1.4.1. Deneysel Çalışma-1: 4 sınıflı çalışma

Deneysel çalışmaların ilkinde, dört sınıfa ait insan hareketleri bulunan bir veri seti üzerinde uygulamalar yapılmıştır. Deneysel çalışmada, 234 tane ayakta duran, 198 tane oturan, 249 tane sola yönelen ve 228 tane sağa yönelen olmak üzere toplamda 909 tane insan hareket verisi bulunmaktadır. Bu görüntülerin otomatik olarak sınıflandırılması için önerilen KSA modeli eğitilerek test verileri üzerine uygulanmıştır. Bu sınıflara ait YOLO ile parçalanmış bazı görüntü örnekleri Şekil 3.6’ da verilmiştir.



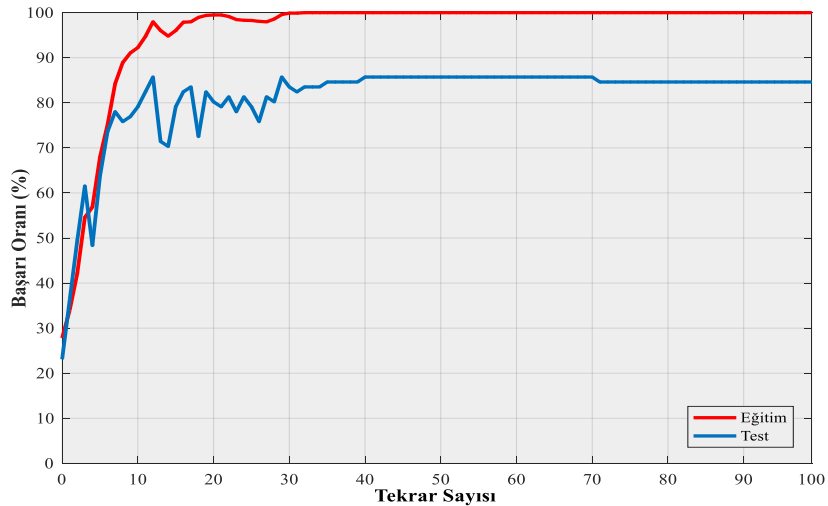
Şekil 3.6. Deneysel çalışma-1 için parçalanmış görüntü örnekleri

Çalışmada, belli oranlarda eğitim, doğrulama ve test verileri alınarak ayrı sonuçlar elde edilmiştir. Önerilen modelde elde edilen sonuçlar ile diğer modellerde elde edilen sonuçlar karşılaştırılmıştır. Önerilen model ile daha yüksek sonuçların elde edildiği gösterilmiştir. Eğitim, doğrulama ve test verileri değiştirilerek en iyi sonuçların elde edilmesi sağlanmıştır. Tablo 3.3.’te, deneysel çalışma-1 için farklı veri bölümlenme oranları ile elde edilen sınıflandırma sonuçları verilmiştir.

Tablo 3.3. Deneysel çalışma-1 tüm deneysel sonuçların karşılaştırılması

Görüntü Boyutu	Modeller	Sınıflandırma Başarımı (%)		
		Verisetleri		
		Eğitim – Doğrulama – Test (%)		
		80 – 10 – 10	70 – 15 – 15	50 – 25 – 25
192 × 72 × 3	Önerilen KSA	85.71	78.10	75.00
	Vgg	83.52	77.37	69.74
	Lenet	82.42	67.15	71.05
128 × 64 × 3	Önerilen KSA	71.43	74.45	68.86
	Vgg	70.33	75.94	69.30
	Lenet	71.43	74.45	67.54
96 × 36 × 3	Önerilen KSA	72.53	77.37	69.30
	Vgg	72.53	74.45	71.05
	Lenet	64.84	73.72	70.8

Tablo 3.3’ te görüldüğü gibi 4 sınıflı veriler üzerinde önerilen KSA modeli diğer modellere kıyasla daha iyi sonuçlar elde edilmiştir. Önerilen KSA modelinde %80 eğitim, %10 doğrulama ve %10 test veriseti üzerinde %85.71 ile en iyi sonuç gözlemlenmiştir. Diğer görüntü boyutları ve verisetlerinde de önerilen KSA modeli diğer modellere göre daha yüksek sonuç vermiştir. Şekil 5.7’de, %80 eğitim, %10 doğrulama ve %10 test şeklinde bölünen dört sınıflı veriseti üzerinde önerilen KSA modeli ile elde edilen eğitim ve doğrulama sonuçlarına ait bir grafik gösterilmiştir.



Şekil 3.7. Dört sınıf için insan hareket tespitinde elde edilen KSA eğitim grafiği (%80-%10-%10)

Şekil 3.7’de görüldüğü gibi derin KSA modelinin eğitimi için 100 tekrar aşaması kullanılmıştır. Grafikten görülebileceği gibi modelin eğitimi 30. Aşamada tamamlanırken doğrulama aşaması tam olarak başarıya ulaşamayıp 80. Aşamada başarımı %85.71 olarak ölçülmüştür. Eğitilen KSA modelinin test verileri üzerinde elde ettiği sınıflama sonuçlarına ait bir doğruluk matrisi verilmiştir. Tablo 3.4’ te önerilen KSA modelinin 4 sınıfa ait test verileri üzerinde elde ettiği doğruluk matrisi verilmiştir.

Tablo 3.4. KSA modelinin test verileri üzerinde elde ettiği doğruluk matrisi (%80-%10-%10)

Sınıflar	1. Ayakta	2.Oturan	3.Sağ Yön	4. Sol Yön
1. Ayakta	20	2	3	0
2. Oturan	3	21	1	0
3. Sağ Yön	2	1	16	1
4. Sol Yön	0	0	0	21

Doğruluk matrisinden görüldüğü gibi, ayakta duran sınıfa atanması gereken imge sayısının 25 olması gerekirken 2 imge oturan sınıfa 3 imge de sağa giden sınıfa yanlış olarak atılarak ayakta duran sınıfa 20 imge atanmıştır. Oturan sınıfa atanması gereken imge sayısının 25 olması gerekirken 3 imge ayakta duran sınıfa 1 imge de sağa giden sınıfa yanlış olarak atılarak oturan sınıfa 21 imge atanmıştır. Sağa giden sınıfa atanması gereken imge sayısının 20 olması gerekirken yanlış bir şekilde imge ataması yapılarak imgelerin 2’si ayakta duran Sınıfa 1’i oturan Sınıfa 1’i sola giden Sınıfa atanarak sağa giden Sınıfa 16 imge atanmıştır. Sola giden sınıf için herhangi bir yanlış atama yapılmayıp 21 imge atanmıştır.

3.1.4.2. Deneysel Çalışma-2: 3 Sınıflı Çalışma

Tezdeki ikinci deneysel çalışmada üç sınıfa ait insan hareketleri bulunmaktadır. Deneysel çalışmada, 234 tane ayakta duran, 198 tane oturan ve 477 tane hareketli olmak üzere toplamda 909 tane insan hareket verisi bulunmaktadır. Bu deneysel çalışmada, daha önceki deneysel çalışma-1 içerisindeki sağa ve sola hareket sınıfları hareketli adı ile tek bir sınıfta birleştirilmiştir. Bu sınıflara ait YOLO ile parçalanmış bazı görüntü örnekleri Şekil

3.8’ de verilmiştir. Bu görüntülerin otomatik olarak sınıflandırılması için önerilen KSA modeli eğitilerek test verileri üzerine uygulanmıştır.



Şekil 3.8. Deneysel çalışma-2 için parçalanmış görüntü örnekleri

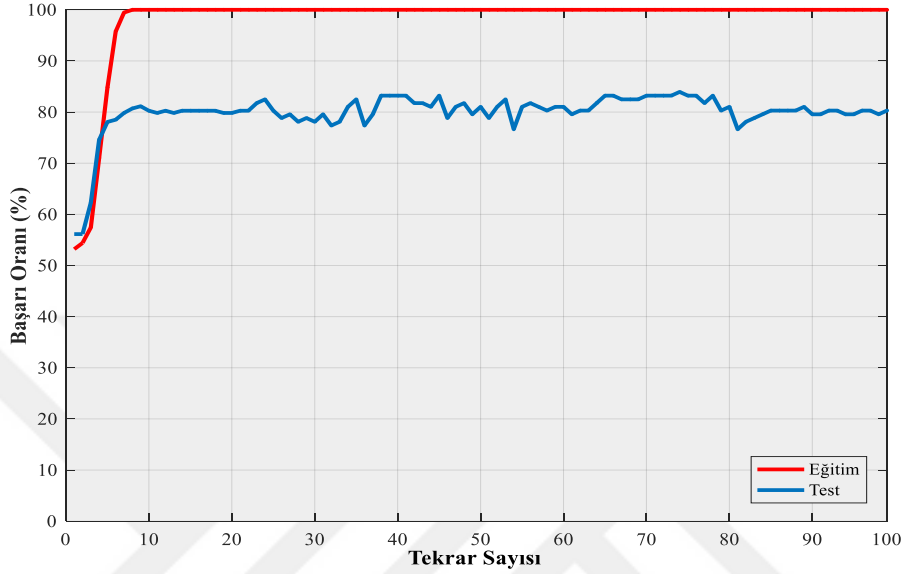
Çalışmada, belli oranlarda eğitim, doğrulama ve test verileri alınarak ayrı sonuçlar elde edilmiştir. Önerilen modelde elde edilen sonuçlar ile diğer modellerde elde edilen sonuçlar karşılaştırılmıştır. Önerilen model ile daha iyi sonuçların elde edildiği gösterilmiştir. Eğitim, doğrulama ve test verileri değiştirilerek en iyi sonuçların elde edilmesi sağlanmıştır. Tablo 3.5’ de, deneysel çalışma-2 için farklı veri bölümlenme oranları ile elde edilen sınıflandırma sonuçları verilmiştir.

Tablo 3.5. Deneysel çalışma-2 tüm deneysel sonuçların karşılaştırılması

Görüntü Boyutu	Modeller	Sınıflandırma Başarımı (%)		
		Verisetleri		
		Eğitim – Doğrulama – Test (%)		
		80 – 10 – 10	70 – 15 – 15	50 – 25 – 25
192 × 72 × 3	Önerilen KSA	81.32	83.94	81.58
	Vgg	78.02	81.02	76.75
	Lenet	74.73	77.37	69.30
128 × 64 × 3	Önerilen KSA	80.22	81.02	81.14
	Vgg	78.02	77.37	79.39
	Lenet	75.82	79.56	75.44
96 × 36 × 3	Önerilen KSA	79.12	81.02	80.26
	Vgg	75.82	81.75	80.70
	Lenet	78.02	77.38	76.75

Tablo 3.5’ de görüldüğü gibi üç sınıflı veriler üzerinde önerilen KSA modeli diğer modellere kıyasla daha başarılı sonuçlar vermektedir. Önerilen KSA modelinde %70 eğitim, %15 doğrulama ve %15 test veriseti üzerinde %83.94 ile en iyi sonuç gözlemlenmiştir. Diğer

görüntü boyutları ve verisetlerinde de önerilen KSA modeli diğer modellere göre daha yüksek sonuç vermiştir. Şekil 3.9’da, %70 eğitim, %15 doğrulama ve %15 test şeklinde bölünen üç sınıflı veriseti üzerinde önerilen KSA modeli ile elde edilen eğitim ve doğrulama sonuçlarına ait bir grafik gösterilmiştir.



Şekil 3.9. Üç sınıf için insan hareket tespitinde elde edilen KSA eğitim grafiği (%70-%15-%15)

Şekil 3.9’da görüldüğü gibi derin KSA modelinin eğitimi için 100 tekrar aşaması kullanılmıştır. Grafikten görülebileceği gibi modelin eğitimi 10. aşamada tamamlanırken doğrulama aşaması tam olarak başarıya ulaşamayıp 40. Aşamada modelin başarımı %83.94 olarak ölçülmüştür. Eğitilen KSA modelinin test verileri üzerinde elde ettiği sınıflama sonuçlarına ait bir doğruluk matrisi Tablo 3.6’da verilmiştir.

Tablo 3.6. KSA modelinin test verileri üzerinde elde ettiği doğruluk matrisi (%70-%15-%15)

Sınıflar	1. Ayakta	2. Oturan	3. Hareketli
1. Ayakta	17	1	2
2. Oturan	1	20	3
3. Hareketli	2	1	43

Doğrulama matrisinden görüldüğü gibi ayakta duran sınıfa atanması gereken imge sayısının 20 olması gerekirken 1 imge oturan sınıfa 2 imge de yürüyen sınıfa yanlış olarak atılarak ayakta duran sınıfa 17 imge atanmıştır. Oturan sınıfa atanması gereken imge sayısının 24 olması gerekirken 1 imge ayakta duran sınıfa 3 imge de hareketli sınıfa yanlış olarak atılarak oturan sınıfa 20 imge atanmıştır. Yürüyen sınıfa atanması gereken imge sayısının 46 olması gerekirken 2 imge oturan 1 imge de ayakta duran sınıfa yanlış atılarak yürüyen sınıfa 43 imge atanmıştır. Böylece modelin test verileri üzerinde elde ettiği genel başarı %83.94 olarak ölçülmüştür.

Tablo 3.7’de önerilen KSA modelinin tüm deneysel çalışmalarda başarı oranları gösterilmektedir. Tablodan görüldüğü gibi dört sınıflı veriseti üzerinde yapılan deneysel çalışmalarda eğitim, doğrulama ve test verileri belirli oranlarda değiştirilerek en iyi sonucun alınması hedeflenmiştir. Deneysel çalışma-1 için kullanılan verisetinde önerilen KSA modeli kullanılarak toplam verilerin %80’i eğitim, %10’u doğrulama ve %10’u test verisi olarak bölündüğünde % 85.71 ile en iyi sonuç gözlemlenmiştir.

Deneysel çalışmaların ikincisinde üç sınıflı bir veriseti üzerinde aynı şekilde eğitim, doğrulama ve test verilerinde belirli oranlarda değişiklik yapılarak en iyi sınıflandırma sonucunun elde edilmesi amaçlanmıştır. Tablodan görüleceği gibi üç sınıflı çalışma için verilerin %70’i eğitim, %15’i doğrulama ve %15’i test aşamasında kullanılarak %83.94 oranında yüksek bir tanıma başarımları sağlanmıştır.

Tablo 3.7. Önerilen KSA modelinin tüm deneysel çalışmalarda performansı

Deneysel Çalışma	Veriseti Eğitim – Doğrulama – Test (%)	Sınıflandırma Başarımı (%)		
		Görüntü Boyutu		
		192 × 72 × 3	128 × 64 × 3	96 × 36 × 3
Çalışma-1 (4 Sınıf)	80 – 10 – 10	85.71	71.43	72.53
	70 – 15 – 15	78.10	74.45	77.37
	50 – 25 – 25	75.00	68.86	69.30
Çalışma-2 (3 Sınıf)	80 – 10 – 10	81.31	80.22	79.12
	70 – 15 – 15	83.94	81.02	81.02
	50 – 25 – 25	81.58	81.14	80.26

4. TARTIŞMA

Bu tez çalışmasında, derin öğrenme yöntemlerinden KSA'lar kullanarak dış ortam görüntülerindeki kişilerin eylemlerinin otomatik sınıflandırılması sağlanmıştır. Bu görüntüler üzerinde öncelikle kişilerin tespit edilerek bölümlenmesi sağlanmıştır. Bu amaçla, önceden eğitilmiş olan YOLO derin öğrenme aracı kullanılmıştır. Bölümlenen kişi görüntüleri farklı eylem sınıflarına ayrılarak bir veri seti oluşturulmuştur. Daha sonra kişi eylemlerinin otomatik olarak sınıflandırılması için 25 katmanlı bir KSA modeli tasarlanmıştır. KSA modeli üzerinde uygulamalar yapılarak sonuçlar sunulmuştur. Önerilen KSA modeli, dört sınıflı veri seti üzerinde %85.71 oranında başarı elde etmiştir. Üç sınıflı veri seti üzerinde ise elde edilen sınıflandırma başarımı %83.94 olarak gözlemlenmiştir. Diğer çalışmalardan farklı olarak nesne tespit aracı olan YOLO ve nesnelerin otomatik olarak sınıflandırılması için KSA birlikte kullanılmıştır.

Literatürde nesne tespiti ve sınıflandırılması için birçok çalışma mevcuttur. Araç, insan, hayvan ve benzeri nesnelerin sınıflandırılması için YOLO derin öğrenme aracı veya derin öğrenme yöntemlerinden KSA'lar kullanılmıştır. Dong ve ark. (Dong ve ark., 2015), araç tiplerinin sınıflandırılmasında araçların önden görünüm görüntülerinden yarım-denetimli KSA kullanmışlardır. Sinir ağı, giriş olarak verilen araç görüntüsünün hangi araç sınıfına ait olabileceğinin olasılık değerini göstermektedir. Geliştirilmiş olan sistemin karmaşık görüntüler üzerinde iyi performans gösterdiği raporlanmıştır. Çalışma kapsamında, araçların sadece ön görünümüne sahip 9850 adet yüksek çözünürlüklü görüntüden oluşan (BIT-vehicle) veri setini oluşturmuşlardır. Önerdikleri sistemleri, gündüz koşullarında %96.1, gece koşullarında %89.4 doğruluk ile çalışmaktadır.

Tashiev ve ark. (Tashiev ve ark., 2017), gerçek zamanlı araç sınıflandırma üzerinde durmuşlardır. YOLO gerçek zamanlı nesne sınıflandırma çerçevesini kullanan bir sistem gerçekleştirilmiştir. Testler için iyi bilinen araç veri setini (BIT-vehicle) kullanmışlardır. 3106 veri setinin 2174'ü eğitim ve 932'si test veri seti olarak kullanılmıştır. Sistemin yüksek oranda performans gösterdiği gözlemlenmiştir.

Baccoubhe ve ark. (Baccoubhe ve ark., 2011), insan hareketlerini herhangi bir ön bilgi kullanmadan sınıflandırmayı öğrenen, tamamen otomatikleştirilmiş derin bir model önermişlerdir. Konvolüsyonel sinir ağlarının 3B'ye genişletilmesine dayanan sistemin ilk basamağı, otomatik olarak uzay zamansal özellikleri öğrenmektedir. Daha sonra

tekrarlanan bir sinir ağı her bir zaman adımı için öğrenilen özelliklerin zamansal gelişimini göz önüne alarak her diziyi sınıflandırmak üzere eğitilmiştir. KTH veri setinde yapılan deneysel sonuçlar; önerilen yaklaşımın mevcut derin modellerden daha iyi performans gösterdiğini ve ilgili çalışmalarla karşılaştırılabilir sonuçlar ürettiğini göstermişlerdir.

Bir çalışmada (Ji ve ark., 2012), derin öğrenme ile kontrolsüz ortamlarda eylemlerin tamamen otomatik olarak tanınması problemi ele almışlardır. Çalışmada, özellik belirleme sürecini otomatikleştiren ham girişlere doğrudan etki edebilen derin bir model olan KSA kullanılmıştır. Çalışmada 3B konvolüsyonel sinir ağı modeli geliştirilmiştir. Geliştirilen model ile insan eylemlerinin tespitinde iyi bir performans sağlanmıştır.



5. SONUÇLAR

Bu tez çalışmasında, başlangıçta derin öğrenme mimarileri ele alınmıştır. Derin öğrenmede kullanılan araçlarla ilgili genel bilgiler verilmiştir. Daha sonra derin öğrenme yaklaşımlarından konvolüsyonel sinir ağları kullanarak dış ortam görüntülerindeki insan hareketlerinin otomatik tanınması için bir uygulama yapılmıştır. Google Street View ortamından elde edilen görüntülerde, daha önceden eğitilmiş nesne tespit aracı olan YOLO ile kişilerin tespiti yapılmıştır. Elde edilen bir çok kişi görüntüsü dış ortamdaki hareketlerine göre dört sınıfta etiketlenmiştir. Etiketlendirilmiş verisetleri üzerinde önerilen KSA modelinin eğitimleri gerçekleştirilmiştir. Eğitilen modelin test verileri üzerindeki tanıma performansları değerlendirilmiştir. Performans değerlendirmelerinde, farklı görüntü boyutları ve farklı veri bölümlenme oranları kullanılmıştır. Önerilen KSA modeli, dört sınıflı veri seti üzerinde %85.71 oranında başarı elde etmiştir. Üç sınıflı veri seti üzerinde ise elde edilen sınıflandırma başarıyı %83.94 olarak gözlemlenmiştir. Bu sonuçlara göre, önerilen yöntemin dış ortam görüntüleri üzerinde kişi hareketlerinin tanınmasına yönelik uygulamalarda kullanılabilir bir yöntem olduğu söylenebilir.

Mevcut nesne tanıma yaklaşımı olan YOLO sadece görüntü içerisindeki kişilerin tespitini gerçekleştirmektedir. Bu çalışmada ise YOLO ile tespit edilen kişilerin o andaki hareketleri ile ilgili bir sınıflandırma işlemi gerçekleştirilmektedir. Bu uygulama YOLO sistemi ile entegre bir şekilde çalıştırılarak amaca yönelik geliştirilebilir bir yaklaşım sunmaktadır. Önerilen yöntemin başarımının artırılması için daha fazla veri kullanımı veya veri artırma yöntemlerinin kullanımı ileride sağlanacaktır. Ayrıca, video görüntüleri üzerindeki uygulamaları yapılarak modelin geliştirilmesi hedeflenmektedir.

KAYNAKLAR

- Abdel-Hamid, O., Deng, L., & Yu, D.,** 2013. Exploring convolutional neural network structures and optimization techniques for speech recognition. In Interspeech, 3366-3370.
- Abdel-Hamid, O., Mohamed, A.r., Jiang, H., & Penn, G.,** 2012. "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition", In 2012 IEEE internati Lawrence, S., Giles, C.L., Tsoi, a. C., & Back, A. D., 1997, "Face recognition: A convolutional neural-network approach", IEEE transactions on neural networks, 8(1), 98-113. Onal conference on Acoustics, speech and signal processing (ICASSP), 4277-4280.
- Abdel-Hamid, O., Mohamed, A. R., Jiang, H., & Penn, G.,** 2012. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, 4277-4280.
- Altaş, İ. H.,** 1999. Bulanık Mantık: Bulanıklılık Kavramı. Enerji, Elektrik, Elektromekanik-3e, 62, 80-85.
- Akdemir, A.,** 2011. Tedarik zinciri yönetiminde çok amaçlı optimizasyon (Doctoral dissertation) Yıldız Teknik Üniversitesi, Fen bilimleri enstitüsü, Yüksek Lisans Tezi, İstanbul.
- Büyükkaracığan, N., Ödük, M. N., Akyol, Y., & Hasan U. Z. U. N.,** 2011. Farklı çimento karışımları ile üretilen beton dayanımlarının bulanık mantık yöntemiyle araştırılması. Selçuk Üniversitesi Sosyal ve Teknik Araştırmalar Dergisi, (1), 79-86.
- Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A.,** 2011. Sequential deep learning for human action recognition. In International Workshop on Human Behavior Understanding, 29-39. Springer, Berlin, Heidelberg.
- Callan, R.,** 1998. The Essence of Neural Networks, 20-56.
- CollobertRonan, Clement Farabet, K. K. & Soumith, C.,** 2017. What is Torch? Available at: <http://torch.ch/>.
- Deng, L., & Yu, D.,** 2014. Deep learning: methods and applications. Foundations and Trends® in Signal Processing, 7(3-4), 197-387.
- Deng, L., Abdel-Hamid, O., & Yu, D.,** 2013. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 6669-6673.

- Deng, L., Li, J., Huang, J. T., Yao, K., Yu, D., Seide, F., ... & Gong, Y.,** 2013, May. Recent advances in deep learning for speech research at Microsoft. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 8604-8608.
- Dong, Z., Wu, Y., Pei, M., & Jia, Y.** 2015. Vehicle type classification using a semisupervised convolutional neural network. *IEEE transactions on intelligent transportation systems*, 16(4), 2247-2256.
- Farley, B. W. A. C., & Clark, W.,** 1954. Simulation of self-organizing systems by digital computer. *Transactions of the IRE Professional Group on Information Theory*, 4(4), 76-84.
- Farabet, C., Couprie, C., Najman, L., & LeCun, Y.,** 2013. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1915-1929.
- Fasel, B.,** 2002. Robust face analysis using convolutional neural networks. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, (2), 40-43.
- Farrell, J. A.,** 1997. Persistence of excitation conditions in passive learning control. *Automatica*, 33(4), 699-703.
- Fukushima, K., & Miyake, S.,** 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, 267-285. Springer Berlin Heidelberg.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J.,** 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580-587.
- Hinton, G. E., Osindero, S., & Teh, Y. W.,** 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
- Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... & Ng, A. Y.,** 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.
- Jain, A. K., Mao, J., & Mohiuddin, K. M.,** 1996. Artificial neural networks: A tutorial. *Computer*, 29(3), 31-44.
- Jarrett, K., Kavukcuoglu, K., & LeCun, Y.,** 2009. What is the best multi-stage architecture for object recognition?. In *Computer Vision, 2009 IEEE 12th International Conference on*, 2146-2153.
- Jia, Y.,** 2017. Caffe. Available at: <http://caffe.berkeleyvision.org/>.

- Krizhevsky, A., Sutskever, I., & Hinton, G. E., 2012.** Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097-1105.
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L., 2014.** Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1725-1732.
- Lai, J. C., Huang, S. S., & Tseng, C. C., 2010.** Image-based vehicle tracking and classification on the highway. In *Green Circuits and Systems (ICGCS), 2010 International Conference on*, 666-670.
- Le, Q. V., 2013.** Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 8595-8598.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P., 1998.** Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D., 1989.** Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 541-551.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P., 1998.** Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D., 1997.** Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1), 98-113.
- LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R., 2012.** Efficient backprop. In *Neural networks: Tricks of the trade*, 9-48. Springer Berlin Heidelberg.
- Liu, C. C., Pierce, D. A., & Song, H., 1997.** Intelligent system applications to power systems. *IEEE Computer Applications in power*, 10(4), 21-22.
- Mamdani, E. H., & Assilian, S., 1975.** An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1), 1-13.
- Mamdani, E. H., 1974.** Application of fuzzy algorithms for control of simple dynamic plant. In *Proceedings of the Institution of Electrical Engineers*, 121(12), 1585-1588.
- Mamdani, E. H., 1976.** Advances in the linguistic synthesis of fuzzy controllers. *International Journal of Man-Machine Studies*, 8(6), 669-678.
- Mesaros, A., Heittola, T., Virtanen, T., 2016.** TUT database for acoustic scene classification and sound event detection. In *Signal Processing Conference (EUSIPCO), 2016 24th European*, 1128-1132.

- Ng, G.S., Erdogan, S.S., Pan, W.N.,** 1993. Neural networks for voice recognition Networks. International Conference on Information Engineering '93. 'Communications and Networks for the Year 2000', Proceedings of IEEE Singapore International Conference on, 1(1), 383 -387.
- Ng, A.,** 2011. Sparse autoencoder. CS294A Lecture notes, 72(2011), 1-19.
- Öztemel, E.,** 2012. Yapay Sinir Ağları ; Papatya Yayıncılık Eğitim, xxii, 232.
- Rabiner, L. R., & Juang, B. H.,** 1993. Fundamentals of speech recognition, (14). Englewood Cliffs: PTR Prentice Hall.
- Redmon, J., & Farhadi, A.,** 2016. YOLO9000: better, faster, stronger. arXiv preprint arXiv:1612.08242.
- Rosenblatt, F.,** 1958. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.
- Salamon, J., Jacoby, C., & Bello, J. P.,** 2014. A dataset and taxonomy for urban sound research. In Proceedings of the 22nd ACM international conference on Multimedia, 1041-1044.
- Simonyan, K., & Zisserman, A.,** 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S.,** 2016. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484.
- Soomro, K., Zamir, A. R., & Shah, M. (2012).** UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.
- Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L.,** 2014. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1701-1708.
- Tang, Z., Li, C., & Sun, S.,** 2017. Single-trial EEG classification of motor imagery using deep convolutional neural networks. Optik-International Journal for Light and Electron Optics, 130, 11-18.
- Tashiev, İ., Kul, S., Şentaş, A., Küçükayvaz, F., Eken, S., Sayar, A., & Becerikli, Y.,** 2017. Konvolüsyonel Sinir Ağı Kullanarak Gerçek Zamanlı Araç Tipi Sınıflandırması Real-Time Vehicle Type Classification Using Convolutional Neural Network.
- URL-1,** 2018. <http://fortune.com/ai-artificial-intelligence-deep-machine-learning/>. (Erişim tarihi: 12.02.2018).

- URL-2**, 2018. Apronix Incorporated : <http://www.aptronix.com/fuzzynet> . (Erişim tarihi: 04:03:2018).
- URL-3**, 2018. Deep Learning for Java. Available at: <https://deeplearning4j.org/>. Erişim tarihi: 06:02:2018).
- URL-4**, 2018.Theano at a glance Available: <http://deeplearning.net/software/theano/introduction.html#contact-us>. (Erişim tarihi: 15:03:2018).
- URL-5**, 2018. Deep Learning Tutorial. Available at: <http://deeplearning.net/tutorial/deeplearning.pdf>. (Erişim tarihi: 20:02:2018).
- URL-6**, 2018. <http://www.derinogrenme.com/yapay-sinir-aglari/>. (Erişim tarihi: 18:03:2018).
- URL-7**, 2018. <http://www.elektrikport.com/>. (Erişim tarihi: 07:02:2018).
- URL-8**, 2018. <http://www.mmsrn.com/yapay-sinir-aglari/>. (Erişim tarihi: 06:03:2018).
- URL-9**, 2018. <http://www.psiko.gen.tr/yapay-sinir-aglari.html>. (Erişim tarihi: 09:02:2018).
- URL-10**, 2018. <http://docplayer.biz.tr/8068542-Istanbul-teknik-universitesi-elektrik-elektronik-fakultesi-derinlikli-ogrenme-ile-konusmaci-duygu-durumunun-siniflandirilmesi.html>. (Erişim tarihi: 02:04:2018).
- URL-11**, 2018. <http://sargetek.com.tr/2.html>. (Erişim tarihi: 17:03:2018).
- URL-12**, 2018. <https://iyarweb.wordpress.com/neural-compute-stick-object-detection-with-neural-networks/>. (Erişim tarihi: 13:03:2018).
- URL-13**, 2018. http://web.itu.edu.tr/~sonmez/lisans/ai/yapay_zeka_icerik1_1.6.pdf. (Erişim tarihi: 07:02:2018).
- URL-14**, 2018. <http://www.hasanbalik.com/projeler/bitirme/24.pdf>. (Erişim tarihi: 09:03:2018).
- URL-15**, 2018. <http://dergipark.gov.tr/http-ordergi-duzce-edu-tr-2279-sayfa-anasayfa>. (Erişim tarihi: 04:04:2018).
- Vedaldi, A., & Lenc, K.**, 2015. Matconvnet: Convolutional neural networks for matlab. In Proceedings of the 23rd ACM international conference on Multimedia, 689-692.
- Zadeh, L. A.**, 1965. Fuzzy sets. Information and control, 8(3), 338-353.
- Zadeh, L. A.**, 1973. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man, and Cybernetic, (3), 28-44.
- Zadeh, L. A.**, 1975. The concept of a linguistic variable and its application to approximate reasoning—I. Information sciences, 8(3), 199-249.

Zeiler, M. D., & Fergus, R., 2014. Visualizing and understanding convolutional networks.
In European conference on computer vision, 818-833. Springer, Cham.



ÖZGEÇMİŞ

21.11.1992 tarihinde Şırnak'ta doğdu. İlkokul eğitimini Şırnak Bener Cordan İlkokulu'nda, lise eğitimini Mardin İMKB Lisesi'nde tamamladı. 2012 yılında girdiği Munzur Üniversitesi Mühendislik Fakültesi Elektrik ve Elektronik Mühendisliği bölümünden 2016 yılında mezun oldu. 2016 yılında Munzur Üniversitesi Fen Bilimleri Enstitüsü Elektrik ve Elektronik Mühendisliği'nde yüksek lisansa başladı. Halen Munzur Üniversitesi'nde yüksek lisans eğitimine devam etmektedir.

