

**Öğrenci Adı Soyadı:** Buğra İLCİ

**Öğrenci Numarası:** 180101038

**Üniversite / Bölüm:** Yalova Üniversitesi – Bilgisayar Mühendisliği

**Proje:** Yapay Sinir Ağları Ders Projesi

**İÇERİK:**

1. Giriş
2. Kurulum ve Veri Setinin İndirilmesi
3. Veri Setinin ve Modelin (CNN) Oluşturulması
4. Modelin Test Edilmesi

**Yapay Sinir Ağları Proje Raporu**

1. **Giriş**

Neredeyse her insanın evcil hayvanı vardır. Kimisi balık besler kimisi köpek, fakat dünya genelinde en çok beslenen iki evcil hayvan **Kedi** ve **Köpek**’tir.

Projede, elde edilen veri setinin yapay sinir ağları kullanılarak modellenmesi ve sonuç elde edilmesi istenmektedir. Ben de Yapay Sinir Ağları dersi ile ilgili proje oluştururken kendim de evcil hayvan beslediğim için, yapay zekanın kedi ve köpek ayrımı yapabilmesini sağlayan bir model oluşturmak istedim.

***Kullanılan Veri Seti ve İçeriği***

Projemde Kaggle ’da bulunan ve 2013 yılında düzenlenen "**Kaggle Machine Learning**" yarışmasında kullanılan **Dogs vs. Cats** veri setini kullandım.

<https://www.kaggle.com/c/dogs-vs-cats/data>

Bu veri seti yaklaşık 25.000 adet kedi ve köpek resmi içermektedir.

***Kodlama için kullanılan uygulamalar***

* Anaconda
* Jupyter Notebook

***Verilerin işlenmesi için kullanılan kütüphaneler***

* KERAS
* Matplotlib
* Pandas
* Opencv-Python
* Numpy

1)

******

*Anaconda ücretsiz ve açık kaynaklı, Python ve R programlama dillerinin bilimsel hesaplama kullanımında paket yönetimini kolaylaştırmayı amaçlayan bir özgür ve açık kaynaklı dağıtımdır.*

* Paket sürümleri conda paket yönetim sistemi ile yönetilir.
* Anaconda dağıtımı Windows, Linux ve MacOS işletim sistemlerinde kullanılabilen veri bilimi paketleri içerir.

2)



*Jupyter Notebook, çeşitli programlama dilleri için etkileşimli bir ortam sağlayan açık kaynak kodlu bir programdır.*

* Aldığınız notları ve hesaplamalarınızı bir arada tutmak için kullanabileceğiniz en başarılı araçlardan biridir.
* Tekrar edilebilir araştırmada ve veri biliminde sıkça kullanılır

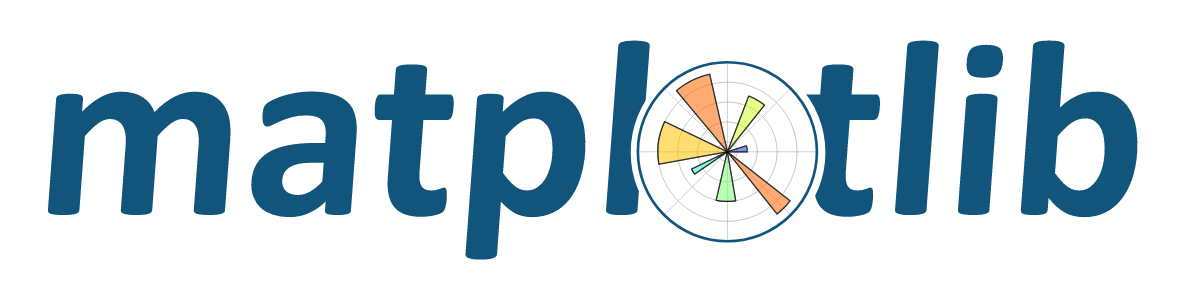
3)



*Keras, neredeyse her tür derin öğrenme modelini tanımlamak ve eğitmek için uygun bir yol sağlayan Python için bir derin öğrenme kütüphanesidir.*

* Kolay ve hızlı bir şekilde model oluşturmaya olanak sağlar.
* Bilgisayarlı görme modelleri için evrişimli sinir ağlarını (CNN), sürekli veriler içinse yinelemeli sinir ağlarını (RNN) destekler.
* Modelleri CPU ve GPU’da sorunsuz bir şekilde çalıştırır. Bu sayede istediğiniz zaman işlemleri GPU’da yapıp zaman kazanabilirsiniz.

4)



*Matplotlib, figürlerin görselleştirilmesinde biz analizcilere yardımcı olan 2 Boyutlu bir çizim kütüphanesidir.*

* Matplotlib en popüler görselleştirme kütüphanesidir.
* Bir figür üzerinde tamamen kontrol sağlanabilir.
* Matlab’ın grafik çizimine benzer bir his vermesi ve kıyasla daha kolay ve ücretsiz olması

5)

metin, vektör grafikler içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Pandas, “ilişkisel” ve “etiketli” verilerle çalışmayı kolay ve sezgisel hale getirmek için tasarlanmış hızlı, esnek ve etkileyici veri yapıları sağlayan bir Python paketidir.*

* Python’da pratik, gerçek dünya veri analizi yapmak için temel yapı taşı olmayı hedefler.
* Ayriyeten, her dilde mevcut olan en güçlü ve esnek açık kaynak veri analizi / manipülasyon aracı olmak gibi daha geniş bir amaca sahiptir.

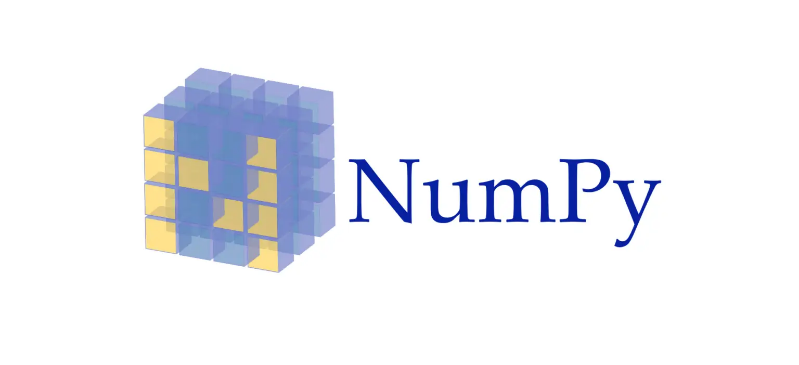
6)



*OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir.*

* OpenCV kütüphanesi içerisinde görüntü işlemeye ve makine öğrenmesine yönelik 2500’den fazla algoritma bulunmaktadır.
* Bu algoritmalar ile yüz tanıma, nesneleri ayırt etme, insan hareketlerini tespit edebilme, nesne sınıflandırma ve görüntü karşılaştırma gibi işlemler rahatlıkla yapılabilmektedir.

7)



*NumPy, Python programlama diline ait çok boyutlu dizilerle ve matrislerle çalışmamıza yardım eden ileri düzey matematiksel işlemler yapabileceğimiz bir Phyton kütüphanesidir.*

* Açık Kaynak ve numerik bir Python kütüphanesidir.
* Çok boyutlu dizin ve matris veri yapılarını içerir
* Pandas nesneleri büyük ölçüde Numpy ile bağlantılıdır. Pandas Numpy’i genişletir diyebiliriz.

1. **Kurulum ve Veri Setinin İndirilmesi**

***2.1 Veri Setini İndirmek***

Veri setine Kaggle üzerinden ulaşılabilir fakat ben bu veri setini Microsoft sağlayıcısını kullanarak indirdim.

<https://www.microsoft.com/en-us/download/details.aspx?id=54765>

***2.2 Kurulum***

Kurulum yaparken, Python - How to install KERAS library in Anaconda ?

<https://www.youtube.com/watch?v=MIkZ6cDE53w>

adlı videodan yararlandım ve gerekli kütüphanelerin kurulumlarını **Anaconda Prompt(CMD)** ile gerçekleştirdim.

*-conda install -c anaconda keras*

*-conda install jupyter*

*-conda install matplotlib*

*-conda install pandas*

*-pip install opencv-python*

1. **Veri Setinin ve Modelin (CNN) Oluşturulması**

***3.1 Kütüphanelerin koda eklenmesi***

Veri setini ve modeli oluşturmadan önce kullandığımız kütüphaneleri kodumuza eklemeliyiz.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

***3.2 Kedi ve Köpek verilerinin bulunduğu 786M ZIP dosyasının indirilmesi***

Veri setine Kaggle üzerinden ulaşılabilir fakat ben !curl fonksiyonu ile bu veri setini Microsoft sağlayıcısını kullanarak indirdim.



***3.3 Bozuk görüntülerin filtrelenmesi***

Çok sayıda gerçek dünya görüntü verisi ile çalışırken, verilerin arasında bozuk görüntülerin olması yaygın bir durumdur. Başlıklarında "JFIF" dizesini içermeyen kötü kodlanmış görüntüler filtreliyoruz.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

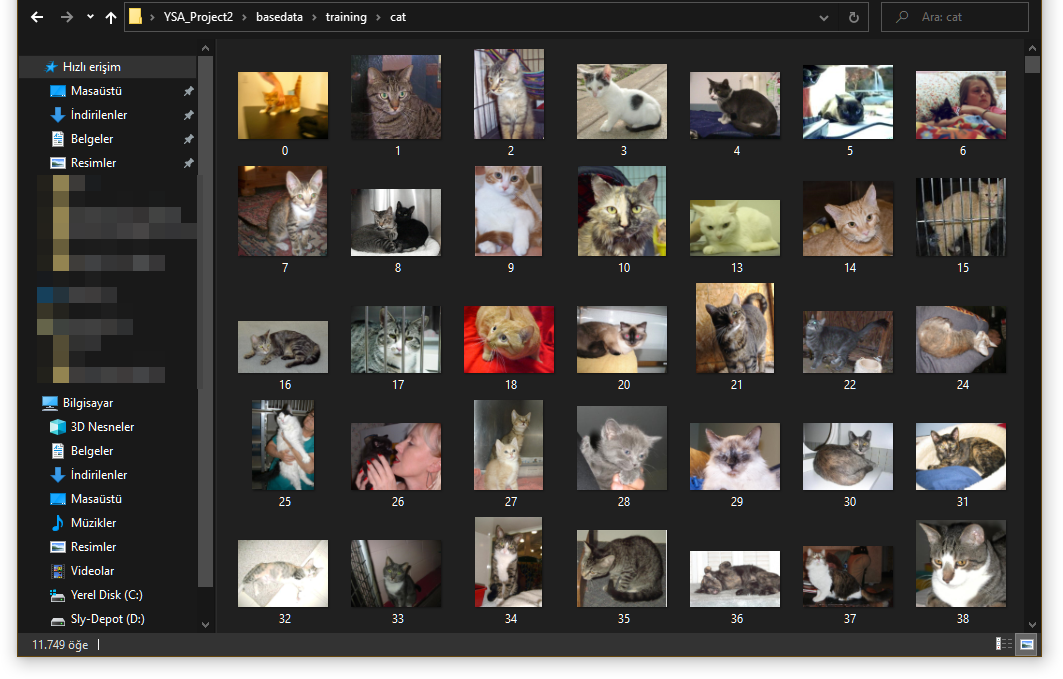
***3.4 Resimler filtrelendikten sonra yeniden dosyalanması***

Resimleri filtreledikten sonra Basedata dediğim bir ana klasörde topladım. Ardından bu klasörün içerisine Training, Validation ve Testing adlı üç adet alt dizin oluşturdum.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

***Örnek olarak Training içerisindeki kedi resimleri;***



***3.5 Resimlere ulaşılabildiğinin kontrol edilmesi***

Resimleri kod ile çekilebilmesinin kontrolü;

metin, ekran görüntüsü, kedi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resimleri başarı ile çekebildikten sonra 3 boyutlu matris verisini görüntüledim.

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

***3.6 Training ve Validation ’da kullanılacak veri setinin oluşturulması***

Öncelikle resimlerimiz RGB olduğu için ve RGB verisi 0-255 arasında olduğu için 1/255 olarak boyutlandırdık.



metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Toplamda 23422 adet resim buldu ve 2 adet sınıfa ayırdı, Kedi ve Köpek.

***Target size ‘ın belirlenmesi***

Sinir ağımıza birden farklı boyutta matris veremeyiz, her matrisin aynı boyutta olması lazım.

Bazı veriler 250x200, bazıları 300x400, kimisi ise 500x650 boyutlarında.

Bu yüzden boyutlarını tahminimce ortası olan 300x300 olarak ayarladım.

***Batch\_size ‘ın belirlenmesi***

Elimde yaklaşık 23.000 adet veri olduğu için ve bilgisayar hafızasını çok zorlamamak için 100 yaptım.

***Class mode ‘un ayarlanması***

Class\_mode 'u binary ayarlamamın sebebi kedi ve köpekleri karşılaştırmam. İki adet sonuç olduğu için 0 ve 1 olan binary 'i kullandım.

Ardından oluşan sınıfları kontrol ettim.

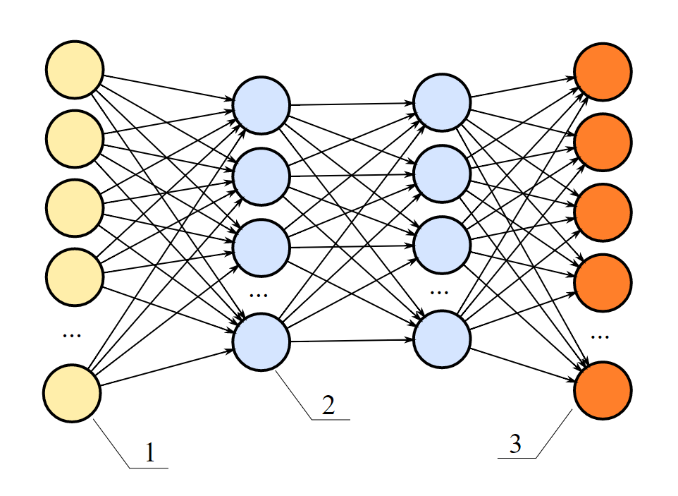


***3.7 Modelin (Convolutional Neural Network) Oluşturulması)***

Benim verilerim görüntülerden oluştuğu için ve veriler düzensiz olduğu için CNN algoritmasını kullanmaya karar verdim.

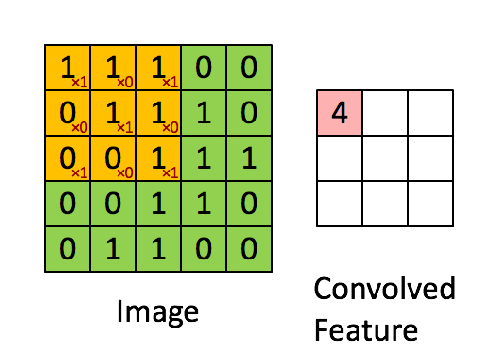
CNN genellikle görüntü işlemede kullanılan ve girdi olarak görselleri alan bir derin öğrenme algoritmasıdır.

* Farklı operasyonlarla görsellerdeki featureları (özellikleri) yakalayan ve onları sınıflandıran bu algoritma farklı katmanlardan oluşmaktadır.
* Convolutional Layer, Pooling ve Fully Connected olan bu katmanlardan geçen görsel, farklı işlemlere tabii tutularak derin öğrenme modeline girecek kıvama gelir.
* CNN modelleri oluştururken, unstructural (düzensiz) veri ile uğraştığımızdan klasik makine öğrenmesi algoritmalarına kıyasla veri ön işleme kısmında çok uğraşmamaktayız.

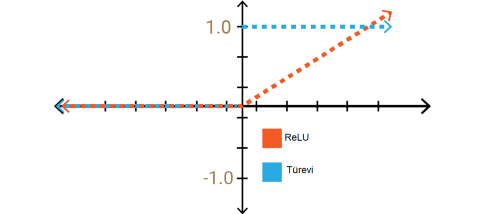


***Convolutional Layer***

Bu katman CNN’nin ana yapı taşıdır. Resmin özelliklerini algılamaktan sorumludur. Bu katman, görüntüdeki düşük ve yüksek seviyeli özellikleri çıkarmak için resme bazı fitreler uygular. Örneğin, bu filtre kenarları algılayacak bir filtre olabilir.

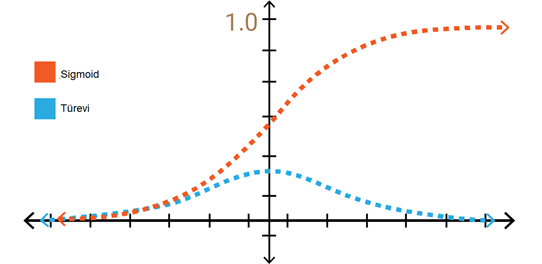


***ReLU (Rectified Linear Unit) Fonksiyonu***



Doğrultulmuş lineer birim (rectified linear unit- RELU) doğrusal olmayan bir fonksiyondur. ReLU fonksiyonu negatif girdiler için 0 değerini alırken, x pozitif girdiler için x değerini almaktadır.

***Sigmoid Fonksiyonu***



En yaygın kullanılan aktivasyon fonksiyonlarından birisidir, [0,1] aralığında çıktı üretir.

***Pooling Layer***

Bu katman, CovNet’teki ardışık convolutional katmanları arasına sıklıkla eklenen bir katmandır. Bu katmanın görevi, gösterimin kayma boyutunu ve ağ içindeki parametreleri ve hesaplama sayısını azaltmak içindir. Bu sayede ağdaki uyumsuzluk kontrol edilmiş olur. Birçok Pooling işlemleri vardır, fakat en popüleri max pooling’dir.

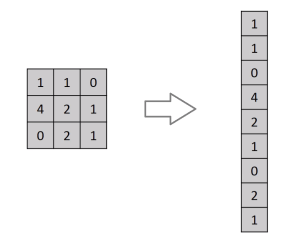
tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resimde gördüğünüz gibi, filtre, kapsadığı alandaki en büyük sayıyı alır. Bu sayede, sinir ağının doğru karar vermesi için yeterli bilgiyi içeren daha küçük çıktıları kullanmış olur.

***Flattening Layer***

Bu katmanın görevi basitçe, son ve en önemli katman olan Fully Connected Layer’ın girişindeki verileri hazırlamaktır. Genel olarak, sinir ağları, giriş verilerini tek boyutlu bir diziden alır. Bu sinir ağındaki veriler ise Convolutional ve Pooling katmanından gelen matrixlerin tek boyutlu diziye çevrilmiş halidir.



***Fully-Connected Layer(Dense)***

Bu katman ConvNet’in son ve en önemli katmanıdır. Verileri Flattening işleminden alır ve Sinir ağı yoluyla öğrenme işlemini geçekleştirir. Katmanın giriş vektöründe doğrusal bir işlemi vardır. Genellikle bunu doğrusal olmayan bir etkinleştirme işlevi izlemektedir.

Aşağıdaki kod modelin nasıl oluşturulacağını göstermektedir.

***metin içeren bir resim

Açıklama otomatik olarak oluşturuldu***

1. katman olan Conv2D(16,(3, 3), activation=’relu’, input\_shape=(300, 300, 3)),

Bu katmandaki:

* Conv2D(16,(3, 3) ‘teki 16 sayısı kaç adet filtreden oluşacağını belirler, 3,3 sayısı filtrelerimin boyutlarını ayarlar ve aktivasyon fonksiyonu olarak ‘Relu’ ‘yu kullandım.
* input\_shape=(300, 300, 3) vermemizin sebebi resimlerimizin boyutlarının 300x300 piksel olmasından kaynaklıdır, sondaki 3 ise RGB’den kaynaklı ve 3 boyutlu matris kullanacağımız içindir.

2. katman olan MaxPool2D((2, 2)),

Bu katmandaki:

* MaxPool2D((3, 3)) ‘teki 2,2 sayısı taranan matris boyutunu belirler.

3. ve 4. Katman, 1. Ve 2. Katmanın aynısıdır. Sadece 3. Katmandaki filtre sayısı 16 yerine 32 olarak belirlenmiştir.

5. katman olan Flatten(), Flattening Layer’ a örnektir

6. katman olan Dense(512), activation=’relu’),

Bu katmandaki:

* Dense(512), activation=’relu’), ‘teki 512 sayısı pozitif tamsayı olmalı ve çıktı uzayının boyutunu belirler, aktivasyon fonksiyonu olarak ‘Relu’ ‘yu kullandım.

7. katman olan Dense(1), activation=’sigmoid’),

Bu katmandaki:

* Dense(1), activation=’sigmoid’), ‘teki 1 sayısı pozitif tamsayı olmalı ve çıktı uzayının boyutunu belirler, aktivasyon fonksiyonu olarak ‘sigmoid’ ‘i kullandım.

***3.8 Modelin Eğitilmesi***

Modelin eğitimi yaklaşık 90 dakika sürdü. Elimde 23000 adet resim olması, milyonlarca RGB verisi ve yüksek miktarda katman olması sebebiyle uzun sürmüş olabilir.

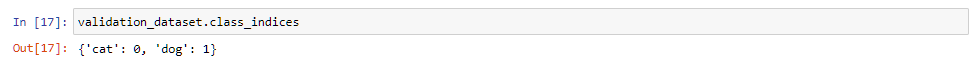
Bilgisayarımdan ötürü sıkıntı olduğunu düşünmüyorum çünkü en yeni sistemlerden birine sahibim; **İşlemci** = Ryzen 9 5900X **Ekran kartı** = Nvidia RTX 3080

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Modelimiz başlangıçta **0.66** oran ile başlarken ilerledikçe **0.9860** oranında bir accuracy ‘ye ulaştı. Bu oran çok mükemmel bir oran ve val\_accuracy ‘nin **0,9997** çıkması modelimizin ezberden çok gerçekten doğru öğrendiğini gösteriyor.

***3.9 Eğitimden sonra oluşan çıktıların kontrol edilmesi***



Eğer sonuç kedi ise 0, köpek ise 1 döndürecek.

1. **Modelin Test Edilmesi**

Aşağıdaki kod ile ‘testing’ dizininde bulunan resimlerin oluşturulan model ile test edilmesini içerir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Model test ettikten sonra eğer sonuç 0 ile “This is a CAT!” yazdırıyoruz.

Fakat sonuç 1 ise yani else, “This is a DOG!” yazdırıyoruz.