Deep Learning

CS 504

Fall-2019

Samarth Subramanya

Computer Science

Classifying Dogs vs Cats

I chose to apply neural network on cat vs dog dataset as I am working on human activity recognition in graduate project. Basically, recognizing cats and dogs is quite easy, but predicting it accurately was a difficult task. With the help of convolutional neural network, it made possible to train the model and predict accurately.

The dataset contains a lot of images of cats and dogs. Our aim is to make the model learn the distinguishing features between the cat and dog. Once the model has learned, i.e once the model got trained, it will be able to classify the input image as either cat or a dog.

The dataset is available in kaggle, the link for the dataset is

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

Implementation:

The project is implemented using Python 3. The framework involved is:

1. Keras- Tensorflow backend
2. OpenCV – for image operations
3. Build model.

With the increase in the resolution of the Image, the number of parameters in case of Simple Neural Networks becomes huge. With that many parameters, it's difficult to get enough data to prevent a simple neural network from overfitting. Moreover, the computational requirements and the memory requirements to train such a model is just a bit infeasible.
Keras Sequential model will be used to build a simple Convolutional Neural Network.

Convolutional Neural Network (CNN's) on the other hand implement what is called a convolution operation, which is one of the fundamental building blocks of a Convolutional Neural Network

Experimental Results:

The training set consists of 25,000 images of dogs and cats, train the algorithm on these sets and classify dog as 1 and cat as 0 in the test1 set. Our Training Accuracy comes out to be 82.53% and Validation Accuracy comes out to be 80.98%. Better result could not be achieved because of the limited computational power.

Convolution Neural Network:

A convolution layer applies a set of "sliding windows" across an image. These sliding windows are termed *filters*, and they detect different primitive shapes or patterns. In the convolution layer, the filters are passed across the input, row by row, and they activate when they detect their shape. Now, rather than treating each of the 12,288 pixel values in isolation, the "windows" treat them in small local groups. These sliding filters are how the CNN can learn meaningful features and locate them in any part of the image. My convolutional neural network has 9 layers including 3 convolution layers.

What is remarkable in convolutional neural network is that the layer has learned a variety of filters which can detect edges and separate elements of the picture. Most importantly, some filters isolate the cat from the background and plainly that's a useful thing, being able to isolate the animal from the rest of the image is step one towards identifying what it is.

Conclusion:

At the end my network got an accuracy of 82 %, I tried to tune and fit the model to get better accuracy, where in I could not get better than 82% as I was running on a low configuration system and processing power was low for the gpu.

I could have got a better result by implementing a bigger network with more layers and epochs, which could not be done because of the restricted processing power of my system.