Name : Shivam Yadav (yshivam9920@gmail.com)

Role : Deep Learning Intern @Synergy Labs

Task : Show us the use of transfer learning to train a model on less data (<1000 images) and still getting good results (>80% accuracy) to demonstrate use of transfer learning. You are free to pick a model of your choice and dataset of your choice.

My Approach :

Step 1 – Dataset Preparation

I have previously worked on the transfer learning with VGG16 Neural network architecture. Hence, I chose it for performing the transfer learning using CNN.

Dataset was taken from Kaggle which was having 4000 training data samples for cat and dog with 1000 test samples. So, I decided to use only the test samples for training which was exactly 1000 images and then for testing any if the samples can be taken from the training samples of the given dataset.

Step 2 – CNN Classification Model Building

For the NN model I used the VGG16 from Keras which had total 138,357,544 trainable parameters which is then customized as per our use i.e we are just considering the last 3 layers of the VGG16 and then try to change the fully connected layer as per our need. BatchNormalization for standardization of raw input variables so that they don’t deviate more from the standard deviation and Dropouts were introduced in order to decrease the computation time of the model. In the Last layer I am made the use of softmax activation as we are using the classification and 2 nodes for computing the results for the cat and dog labels.

Step 3 - Model Training

In the training part, I tried SGD and RMSprop as the optimizers with different permutations of the hyperparameters(learning rate and momentum) out of which RMSprop performed pretty good as compared to SGD.

Step 4 – Classifier Testing

Similarly, I took a fresh set of test samples. And tested the model on it.

And got a 91% accuracy for dog samples and 100% accuracy in cat samples.

\*\*

References : Stackoverflow(for some errors which I came across during this task), My Previous Personal Project(for referring the VGG16 implementation with the ImageDataGenerator which I used in my project earlier), machinelearningmastery.com (For referring what is batch normalization)

\*\*