CS-671

Deep Learning and its Applications

Assignment 2

Task 2

Multi Head Classification

submitted by

Team 5

Vishal Anand B16040

Aman Jain B16044

Yash Agrawal B16120

SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MANDI, MANDI

March 2019

1 Objective

The objective of this assignment is to design a non-sequential convolution neural network for classifying the line dataset created in Assignment 1. The network should have 4 outputs based on the 4 kind of variations(length, width, color, angle).

2 Model

The code for the complete model of the above task can be found at :
https://github.com/Vishal1541/DeepLearning/tree/master/Assignment2/Task2_MultiHead
Filename : task.ipynb

2.1 Architecture

The network architecture is divided in two parts a) Feature network and b) Classification heads. The feature network is extracting the required features from the input and the 4 classification heads attached to it are classifying four types of variations namely length, width, color, angle.

The complete architecture is implemented using Keras functional API because we require multiple outputs from single input.

2.1.1 Feature Network

Model with the following architecture was used to extract the features from the input :

1. 3x3 Convolutional Layer with 32 filters and stride of 1.
2. ReLU Activation Layer.
3. Batch Normalization Layer
4. 2x2 Max Pooling layer with a stride of 2
5. 1x1 Convolutional Layer with 8 filters and stride of 1.
6. ReLU Activation Layer.
7. Batch Normalization Layer.
8. Dropout Layer with a rate of 0.4
9. Layer to flatten the features

2.1.2 Classification heads

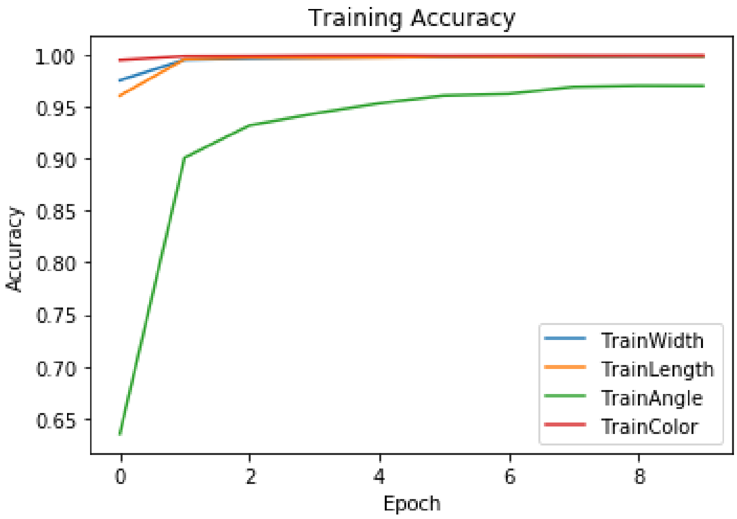
Model with the following architecture was used for classification heads :

1. Fully Connected layer with 512 output units.
2. ReLU Activation Layer.
3. Dropout Layer with a rate of 0.4
4. Batch Normalization Layer
5. Fully Connected layer with 1,1,1,12 output units respectively for length, width, color and angle classifications.
6. Sigmoid, sigmoid, sigmoid, softmax activation functions respectively for length, width, color and angle classifications.

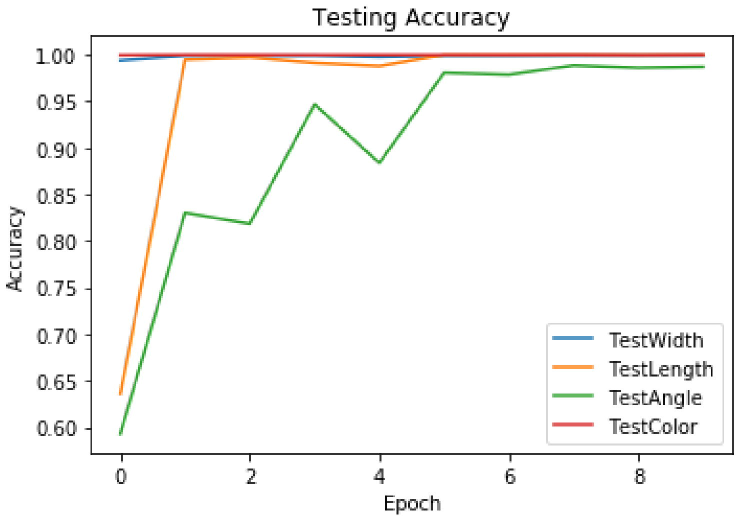
3 Results

3.1 Learning Curves

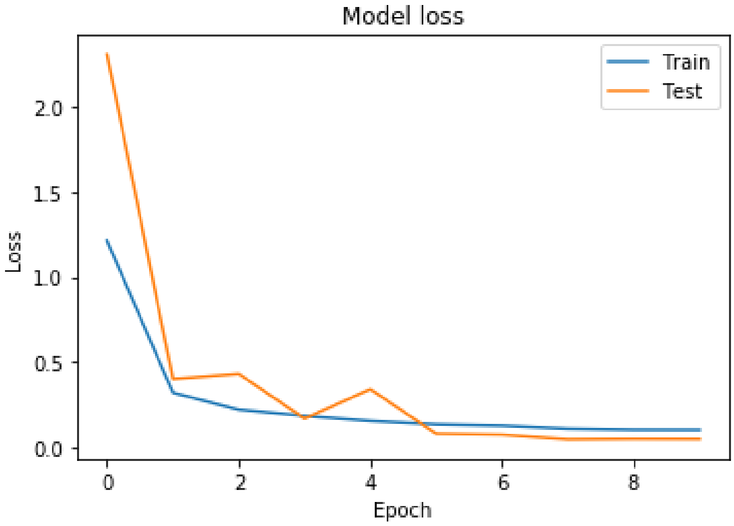
- Training Accuracy



- Testing Accuracy



- Loss



3.2 Accuracy

3.2.1 Separate Accuracy

Accuracy of each of the 4 classification heads separately are :

- Length : 0.9996
- Width : 0.9991
- Color : 0.9997
- Angle : 0.9869

3.2.2 Aggregate Accuracy

Aggregate accuracy is calculated by multiplying the accuracy of each of the classification heads because an image is correctly classified overall if it is correctly classified by each of the classification heads. Therefore, probability of correctly classifying by each of the classification heads is multiplied.

Aggregate accuracy : **0.9853**

3.3 F-Scores

F-Acores of each of the 4 classification heads separately are :

- Length : 1.0
- Width : 1.0
- Color : 1.0
- Angle : 0.9869

3.4 Confusion matrix

- for Angle :
  3044 143 0 0 0 0 0 0 0 0 0 13
  0 3197 3 0 0 0 0 0 0 0 0 0
  0 60 3140 0 0 0 0 0 0 0 0 0
  0 0 33 3167 0 0 0 0 0 0 0 0
  0 0 0 4 3189 7 0 0 0 0 0 0
  0 0 1 0 7 3191 1 0 0 0 0 0
  11 0 0 0 0 58 3131 0 0 0 0 0
  0 0 0 0 0 3 4 3188 3 0 2 0
  0 0 0 0 0 0 0 18 3176 6 0 0
  0 0 0 0 0 0 0 0 9 3127 64 0
  0 0 0 0 0 0 0 0 0 0 3193 7
  0 25 0 0 0 0 0 0 0 0 22 3153

4 Variations

The first model we tried was the one mentioned above and it gave us an accuracy of about 99% in the first attempt. The only variation we tried was with the number of epochs :

- Initially we gave number of epochs as 7 which was over-fitting the data because till 6th epoch the accuracy was increasing and on 7th it decreased.
- On decreasing the number of epochs to 6, the accuracy decreased by 2%.

- Finally we run our model with 10 number of because it seemed that on increasing the epoch the model might be able to train good and the same happened. It decreased in middle but at last the final accuracy was good.

5 Inferences

- The complex models such as multiple inputs, multiple outputs, shared layers, etc can be implemented using Keras functional API.
- The training phase of the model greatly depends on the architecture used for each of the feature network as well as the classification heads.
- The classification accuracy for angle classification was less when compared to other three classification heads because for a network it is difficult to predict the correct angle as the line created during 1st assignment was using pixels. The line was created as blocks of pixels one above the other, similar to stairs. This creates difficulty in correctly classifying the image as compared to other three.
- Increasing or decreasing the number of epochs might lead to over-fitting as well as under-fitting, therefore the model should be tested for different number of epochs for high accuracy.
- The number of dense layers or conv layers in fully connected network or convNet greatly determines the accuracy of the network. Increasing the number of layers beyond certain amount might also deacrease the accuracy because it makes our network deep and more prone to over-fitting.