CSE474/574: Introduction to Machine Learning (Fall 2017)

Project 4: Introduction to Deep Learning

Name: Prachi Jayakumar Patil
Person Number: 50247309
UBID: ppatil3

Objective:

Implementation of a convolution neural network to determine whether the person in a portrait image is wearing glasses or not.
CelebA dataset is used for the training and testing purposes.

Implementation:

1) **Extract feature values and labels from the data**:

The images are read using the scipy library of python and stored in numpy array

The labels are stored in the form of **one-hot vector representation** of dimension 1x2 such that

If the person is wearing glasses, labels = [0,1]

Else, labels = [1,0]

One-hot encoding significance:

Most machine learning algorithms require numerical input and output variables.

An integer and one hot encoding is used to convert categorical data to integer data.

2) Resolution of images:

The images are converted to **28x28 pixels**, then converted to grayscale images and then flattened.

3) **Size of dataset**:

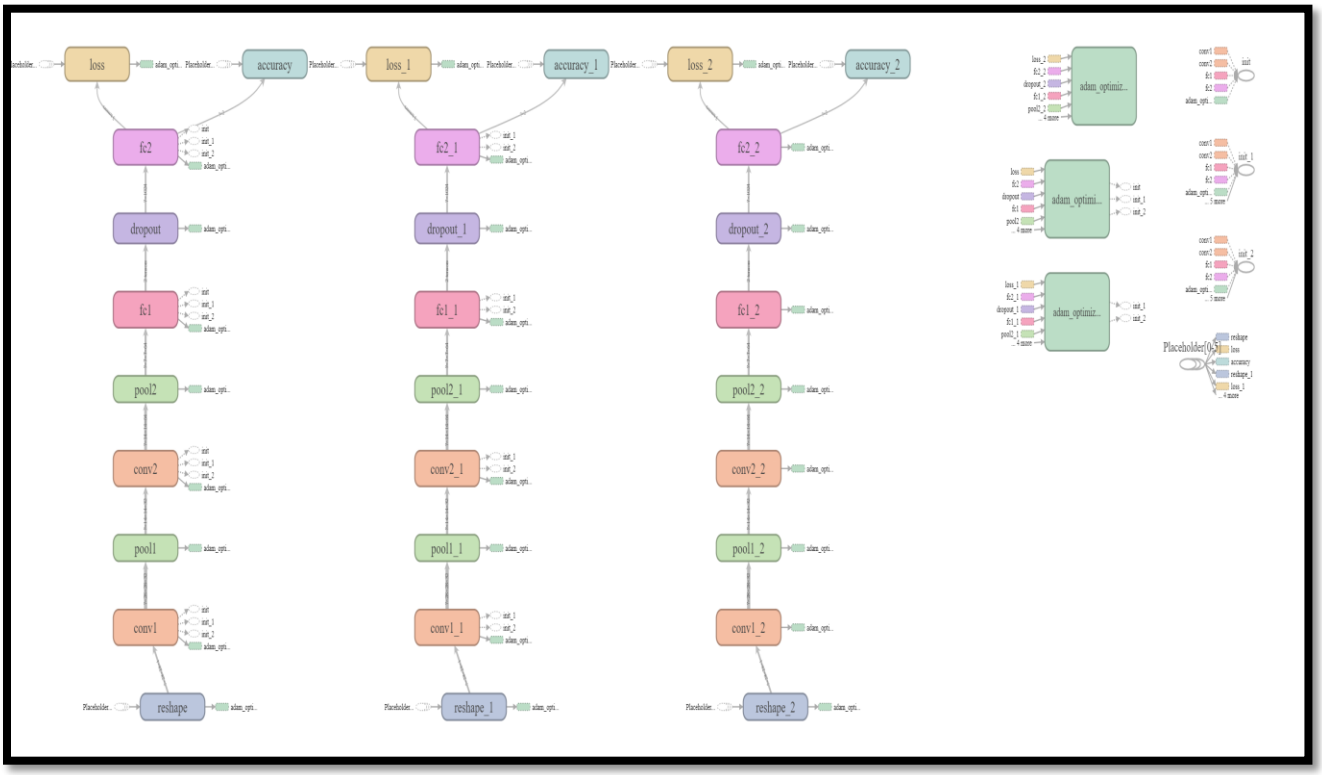
Complete celebA dataset i.e. 202599 is used.

4) Data Partition:

The data is partitioned into training and testing set in **90:10** proportion.

5) Neural Network Architecture:

1) **<u>Network woth 2 hidden layers:</u>**

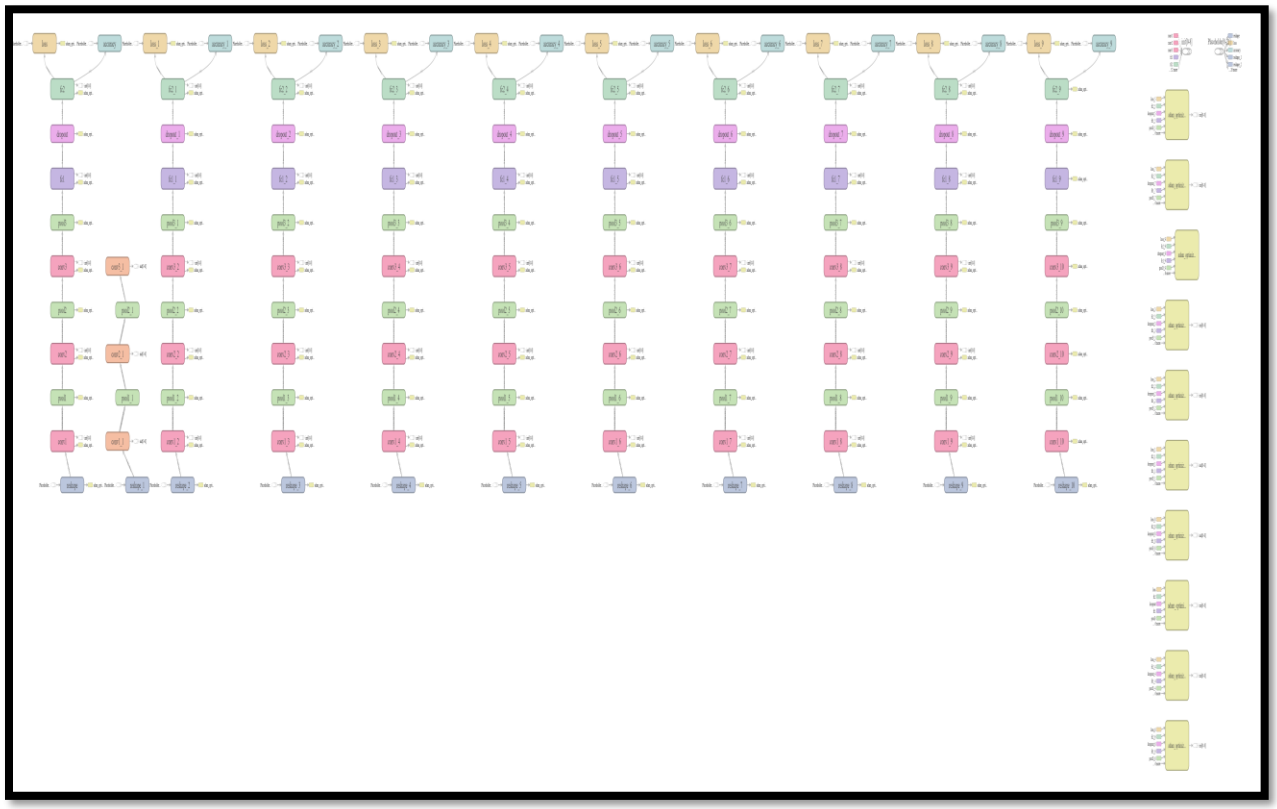


<u>Adding a layer:</u>

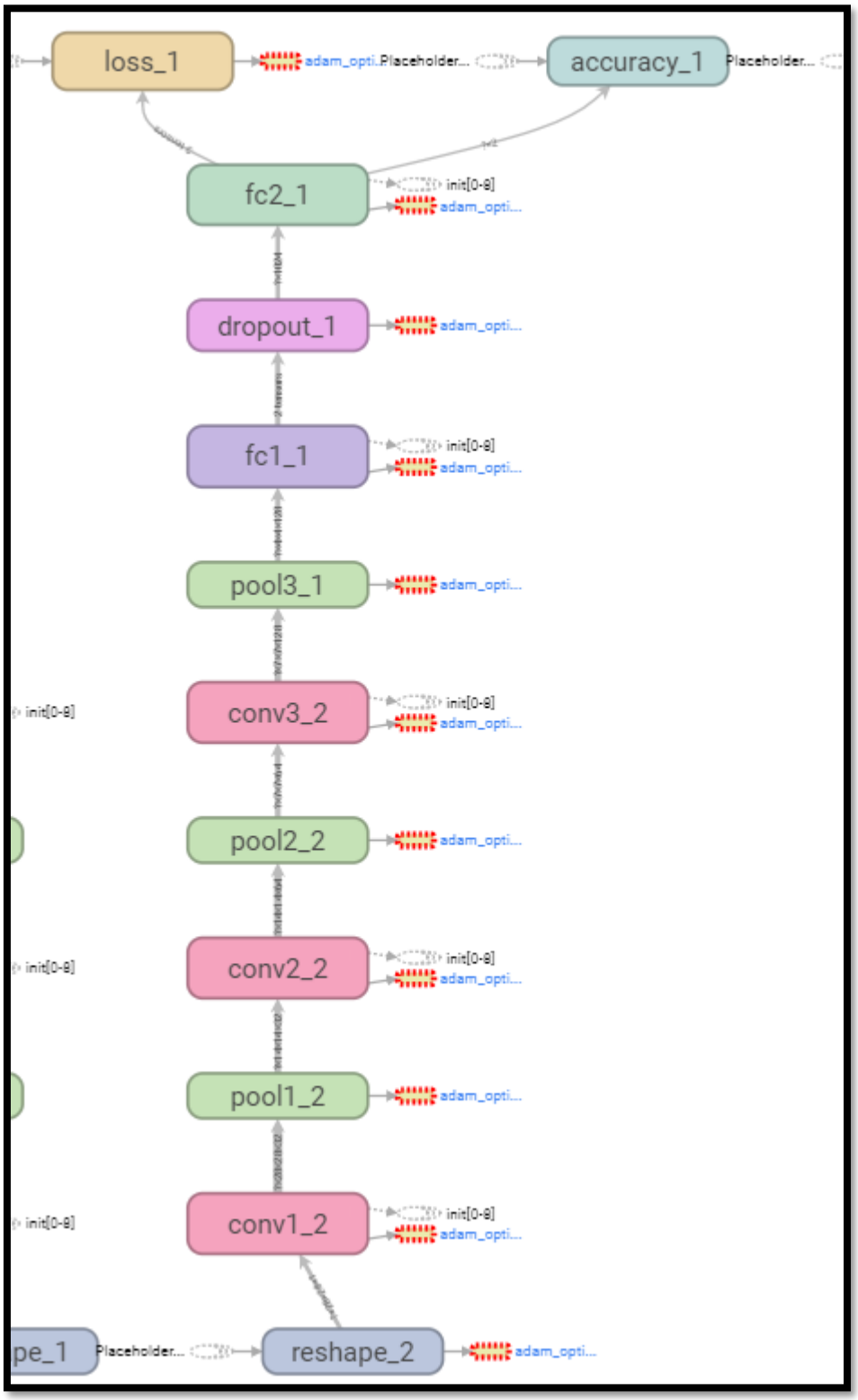
Adding layer increased the number of weights in the network, also the model complexity.

But without large dataset, it may lead to overfitting of training data and thus reduction in testing accuracy. So the layers added and tried were limited.

2) **Network with 3 layers**:

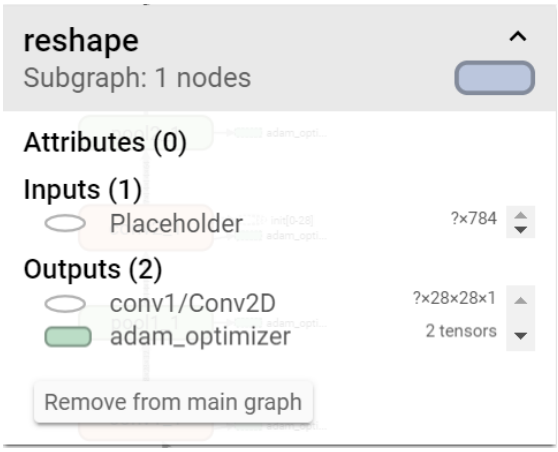


Zooming in:

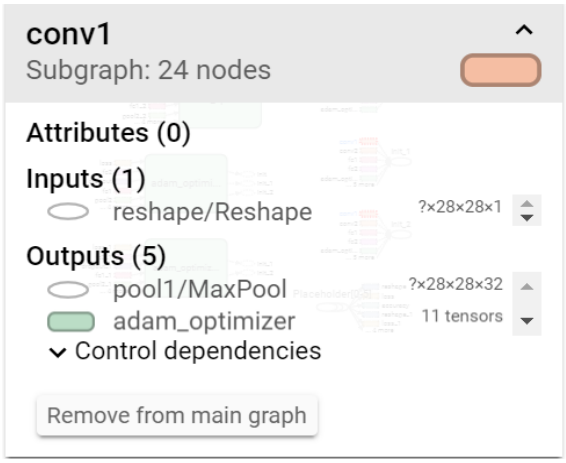

loss_1
adam_opti...
Placeholder...
accuracy_1
Placeholder...
fc2_1
init[0-8]
adam_opti...
dropout_1
adam_opti...
fc1_1
init[0-8]
adam_opti...
pool3_1
adam_opti...
init[0-8]
conv3_2
init[0-8]
adam_opti...
pool2_2
adam_opti...
init[0-8]
conv2_2
init[0-8]
adam_opti...
pool1_2
adam_opti...
init[0-8]
conv1_2
init[0-8]
adam_opti...
pe_1
Placeholder...
reshape_2
adam_opti...

All the layers and their corresponding features can be seen as below:

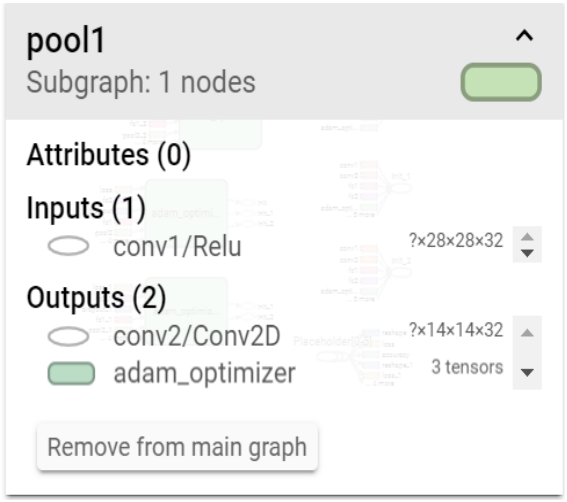
1) Reshape



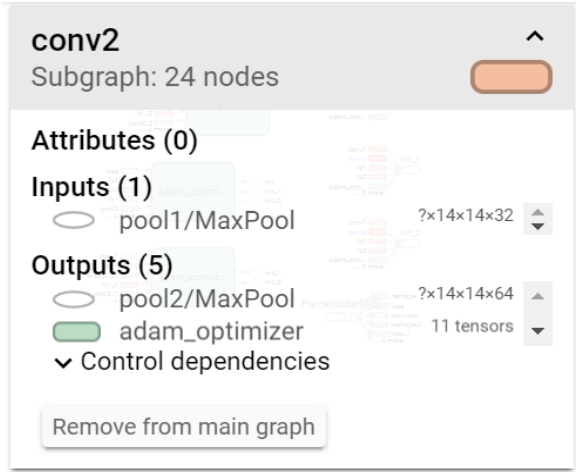
2) First convolutional layer - maps one grayscale image to 32 feature maps:



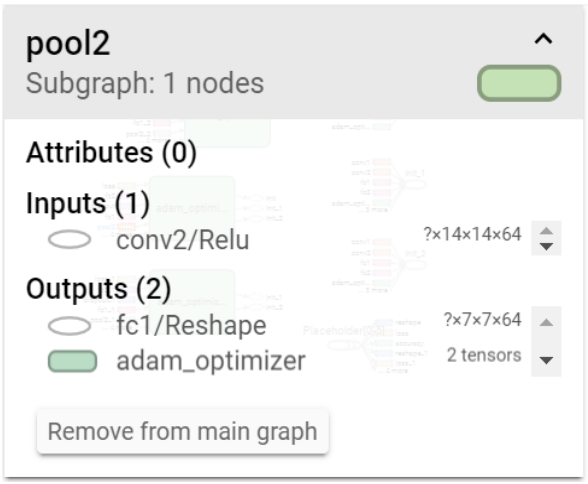
3) Pooling layer - downsamples by 2X.



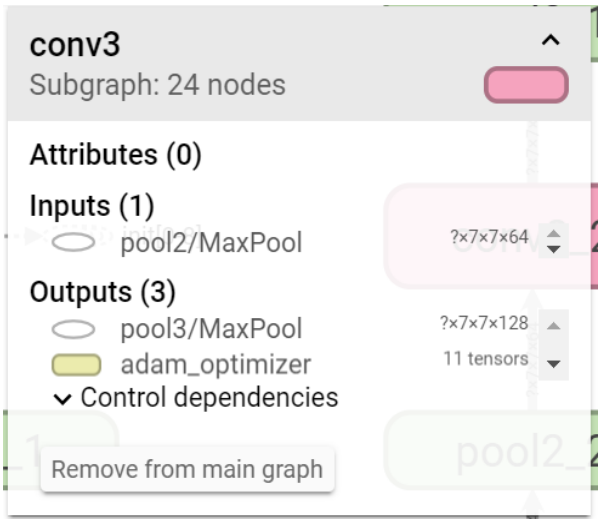
4) Second convolutional layer -- maps 32 feature maps to 64.



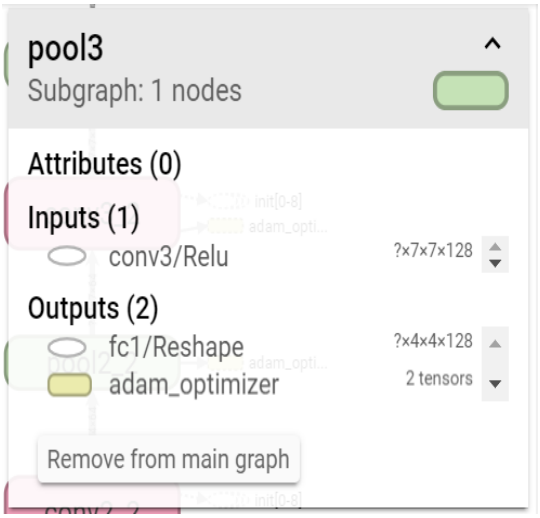
5) Second pooling layer.



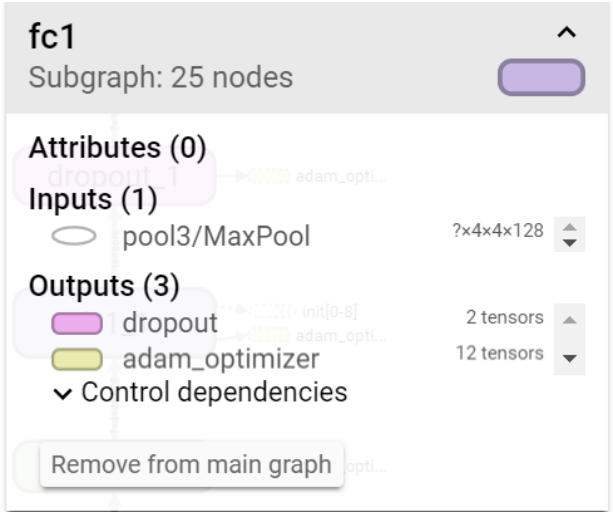
6) Third convolutional layer -- maps 64 feature maps to 128.



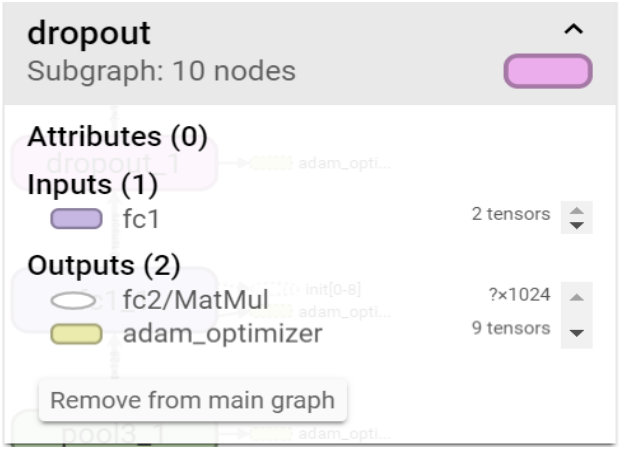
7) Third pooling layer.



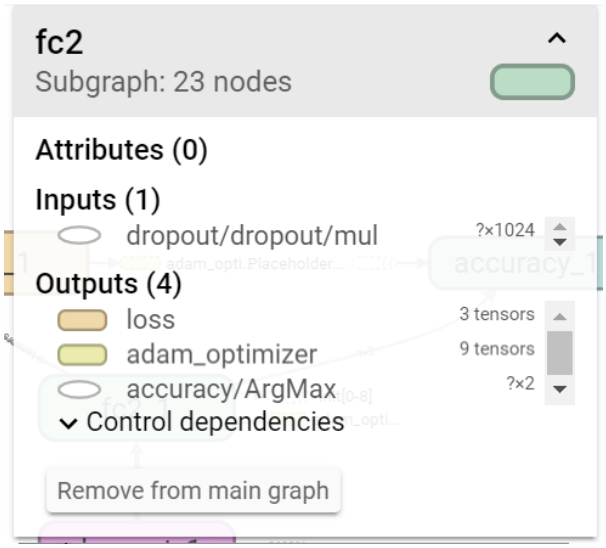
8) Fc1:



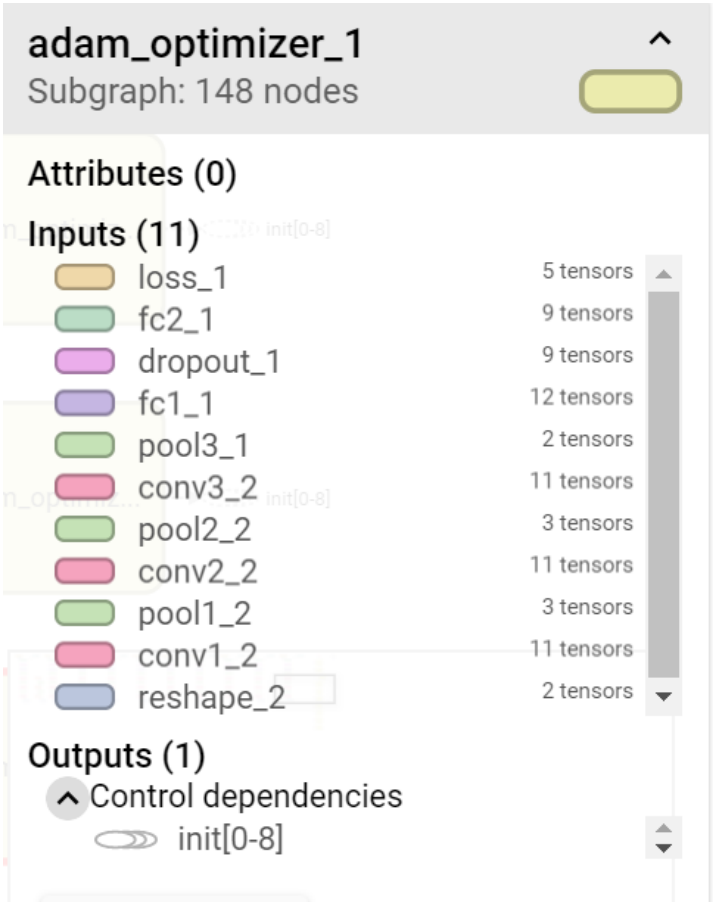
9) Dropout



10) Fc2:

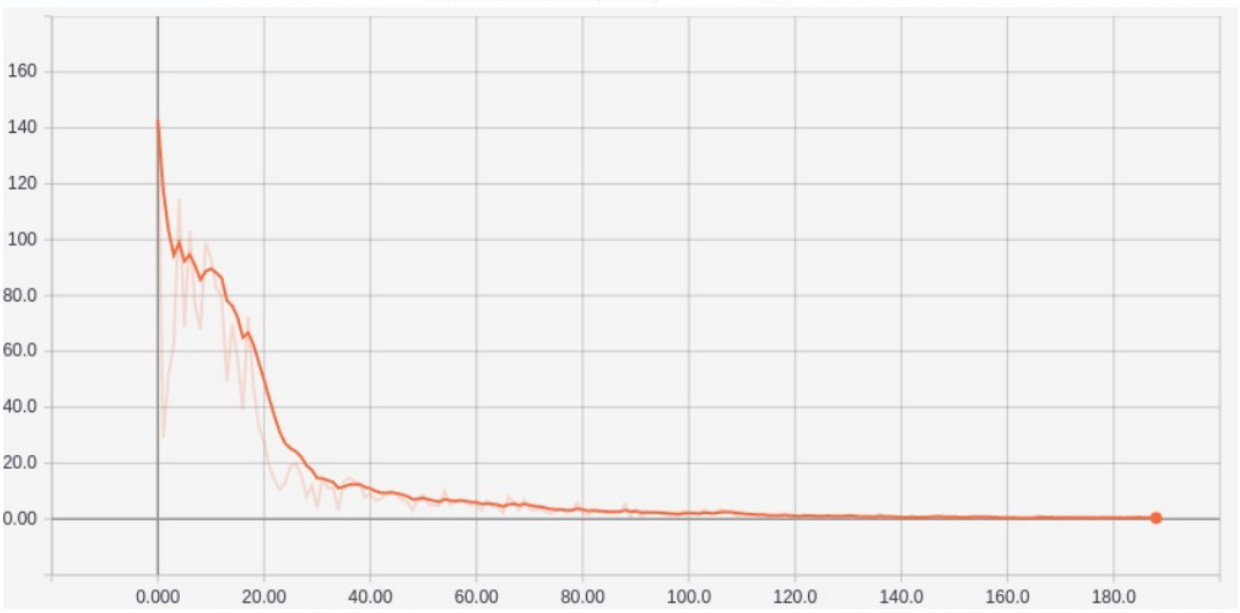


11) Adam Optimiser:

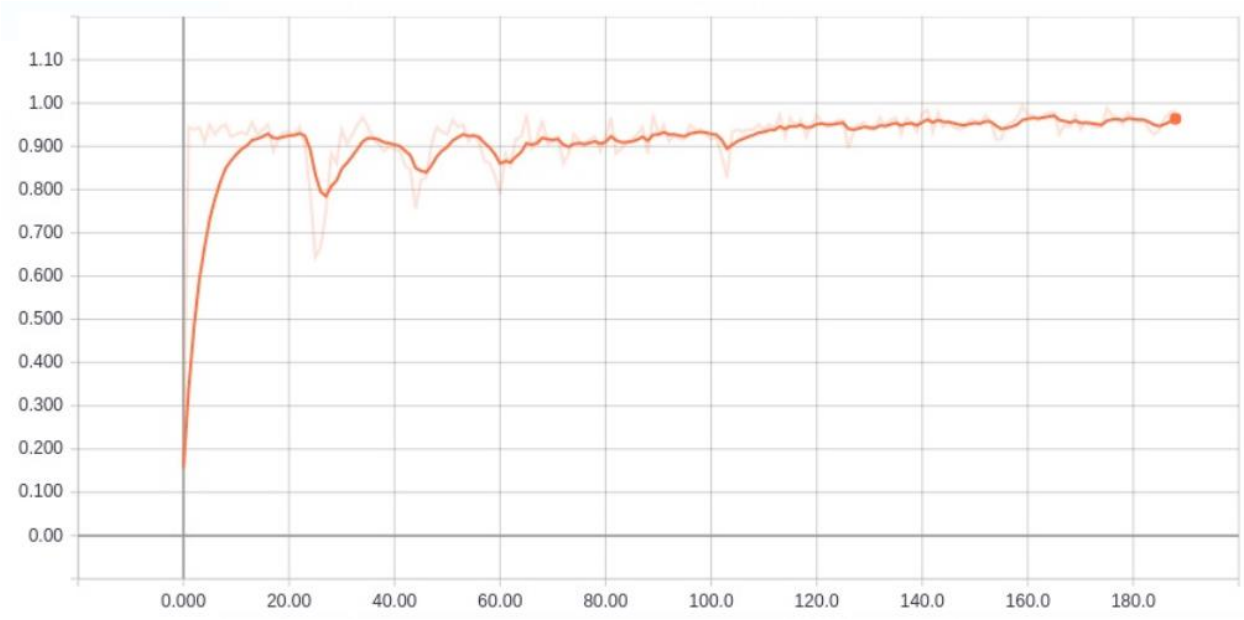


Graphs:

1) Cross Entropy Vs Iterations:



2) Accuracy Vs Iterations:



Hyper Tuning the parameters:

1) Number of epochs:

Number of epochs	Accuracy
10	92.7%
20	93.6%
30	94.8%

The number of epochs should be chosen carefully when training models. With a small number of epochs, the network may not be trained sufficiently. With a larger number of epochs, there is a chance of overfitting.

2) Dropout:

Increasing the number of layers in neural network leads to overfitting of the training data and poor testing performance of the testing data. Dropout is a regularization technique for handling this overfitting problem. Dropout is a 'learning less to learn better' technique.

Dropout was varied from 0.3 to 0.7 and it was observed that the accuracy is better at 0.5.

Retrain the model using higher resolutions. Does that improve the performance?

In this model, the resolution used is 28x28 pixels. I tried using the resolution 64x64 the accuracy increased slightly to 93.87%. But the neural network took larger computation time as compared to the existing model.

Use bigger sizes of the training set.

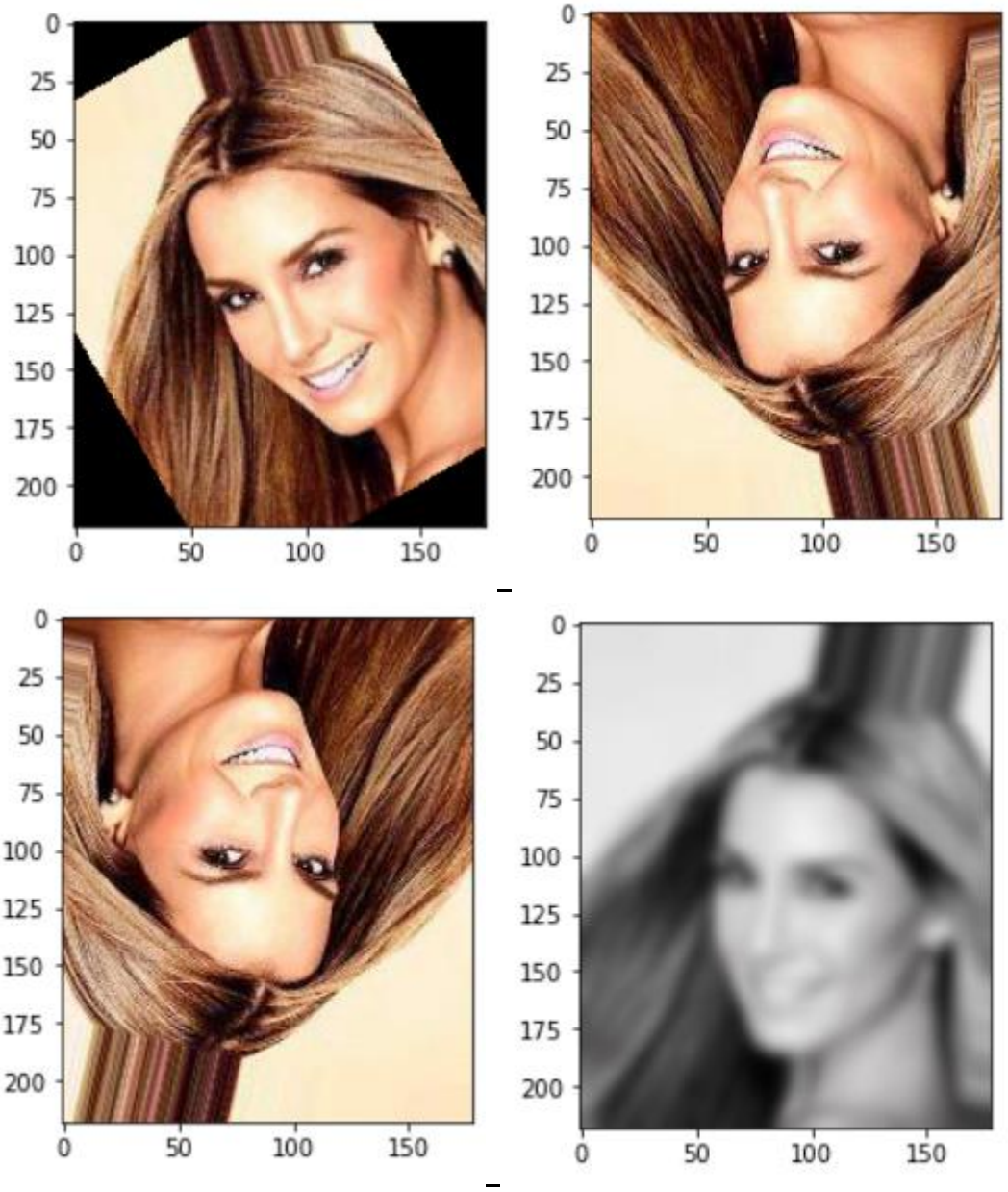
Size of dataset	Test Accuracy
1000	92.2%
50,000	92.5%
1,00,000	93.7%
2,00,000	93.8%

Increasing the training dataset leads to increase in the learned features by the neural network, thus the accuracy increases.

Data Augmentation:

Following operations are applied to add to dataset:

1) Rotate
2) Flip array up down
3) Flip array left to right
4) blur



Thus we can achieve data augmentation for increasing the size of training set to overcome the problem of overfitting. Since we have limited resources and limited time, it was not feasible to run the training on such large dataset.

Conclusion:

Successfully implemented the CNN architecture, tuned the hyper parameters and achieved the accuracy of 93.8%.