

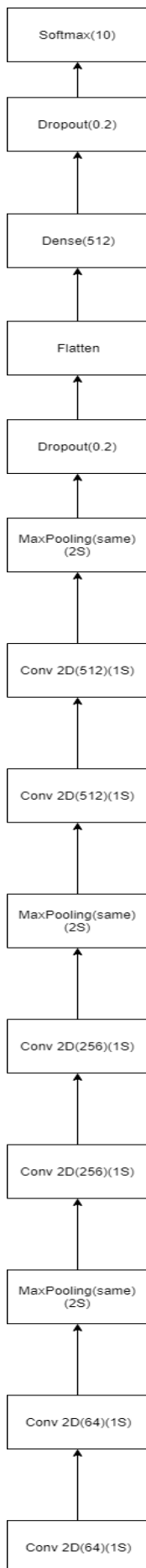
Report Assignment 2b

In this part I have experimented with 3 types of architectures. One is the normal CNN , other is the ResNet and the third one is the InceptionNet. In this part I have used Data Augmentation with the help of Keras Image Pre-processing tool , Image Data Generator. Here I have used horizontal flip, height shift of 0.1 and width shift of 0.1. I have used a validation split of 5000 i.e I have trained on 45000 examples and used the rest 5000 examples as validation set. In spite of setting Tensorflow seed I was not able to reproduce results as I am using GPU's.

Normal CNN

In the next page I have shown the architecture that I have used in this part. I have used The Relu Activation function and used dropout regularization to ensure that my network does not overfit the data. Using Adam optimizer, 500 batch_size ,I have trained the network using the following training policy. I trained for 20 epochs on data augmentation, then trained for 5 epochs on the original training data, then trained on augmented data for 10 epochs, then on training data for 5 epochs, then on augmented data for 7 epochs and lastly for 3 epochs on original training data. Using this I got an accuracy of 90.42% max on the test set and an average of around 90%. I have used all the default initializers as my accuracy was being affected adversely using other initializers.

Fig: Architecture of the normal CNN with regularization



ResNet Architecture

In this architecture I have mainly used 2 types of basic building blocks namely the Identity Block and the Convolutional Block.

The Identity Block: Here I have used 3 layers of convolution, Batch normalization, Activation Function(ReLu) and used a skip connection between input to first layer and the output of batch normalization of the third layer.

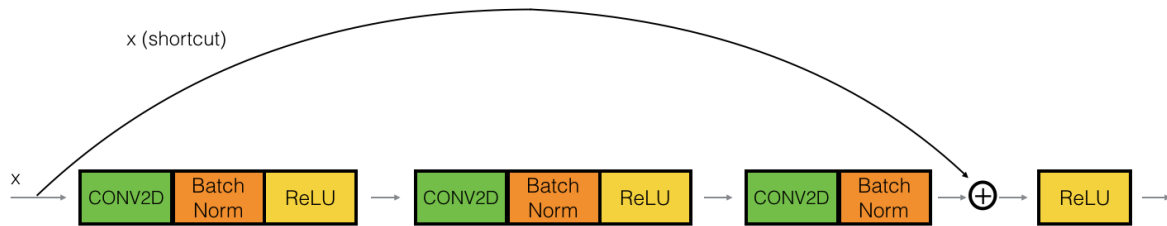


Fig: IDENTITY BLOCK(Coursera CNN course)

The Convolutional Block: Here the architecture is similar to the identity block but here we account for the fact where input of the first layer and output of the last layer have different dimensions. On the skip connection we use another convolution to make the dimensions same.

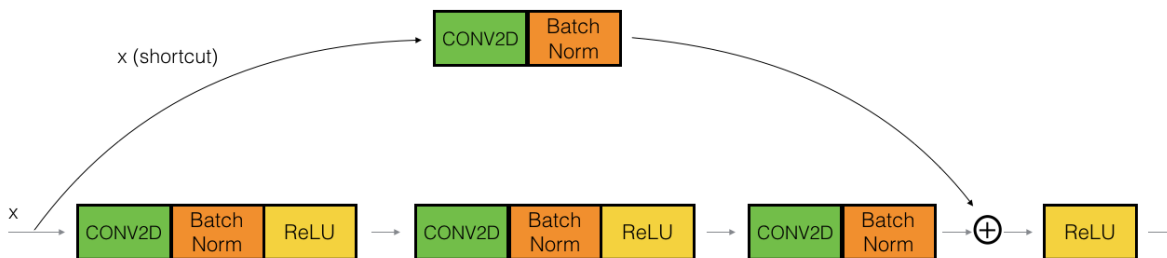


Fig: Convolutional Block(Coursera CNN Course)

I made the architecture given below. Since it was quite deep 1hr was not enough to completely train the network as after 1hr the accuracy was still increasing. I trained it for 10 epochs on augmented data and 5 epochs on original data and got an accuracy of 88.76% on the test data. This architecture has a natural regularization effect.

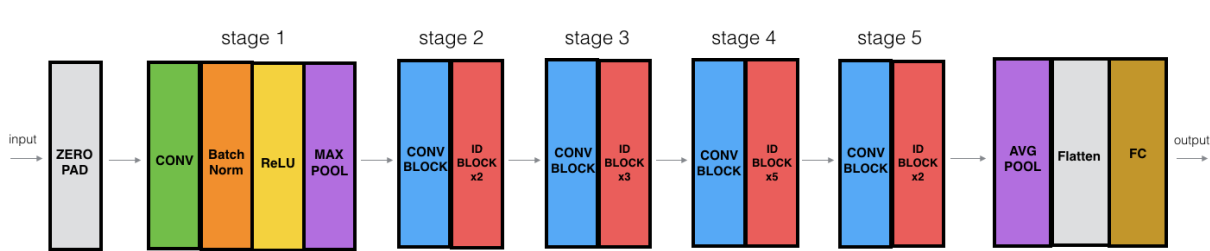


Fig: ResNet Architecture

InceptionNet Architecture



This is the Inception Architecture that I have used. I have used the ReLu Activation function at all places and used a dropout regularization to prevent overfitting. Then I have also used Data Augmentation to prevent overfitting. I have used the Adam optimizer, 500 batch size to train my model. After about 40 minutes of training on Kaggle GPU I got an accuracy of about 91.52% max and an average accuracy of about 91%. But it was not able to finish in 1hr on HPC. If I reduced the amount of training or number of layers my accuracy was suffering big time. So I chose not to include this as my architecture.

Overall the assignment was good and I had a lot to learn but the time constraint ruined it for me personally. We could have implemented much more deeper networks had we got a little bit of leeway on training time. From next time for deep learning please give us more freedom with training time. Moreover the GPU at HPC is very slow.

Fig: Inception Architecture implemented by me.