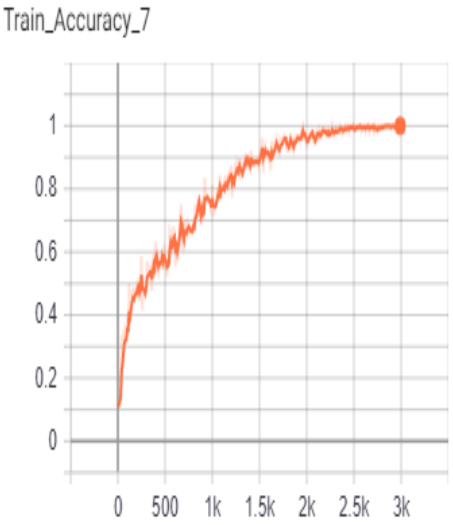
COMP 576 HW 2

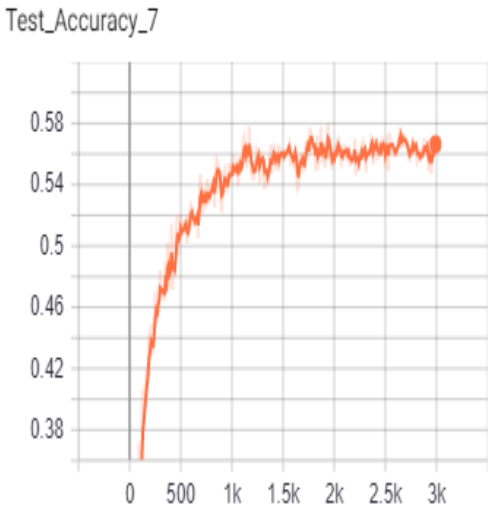
Jingyu FU (jf70)

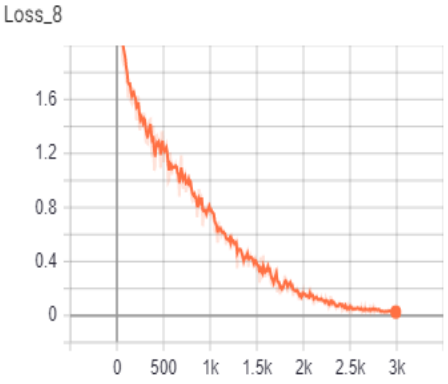
1 Visualizing a CNN with CIFAR10

b) Train LeNet5 on CIFAR10

First, I used Adam learning rate = 0.001



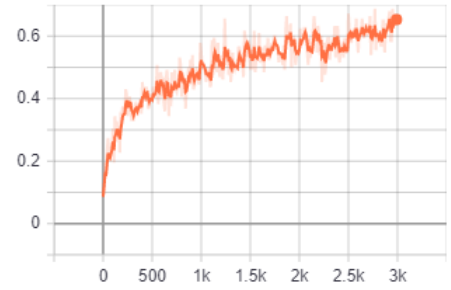




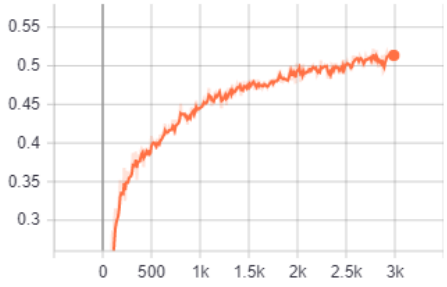
Find good hypermeters:

Tested Adam optimizer with leaning rate = 0.0001, it turns out that having a learning rate less than 0.001 will have negative effect for the model.

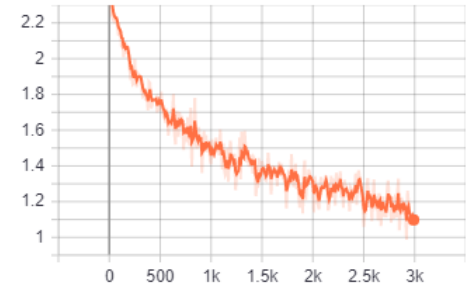
Train_Accuracy_8



Test_Accuracy_8

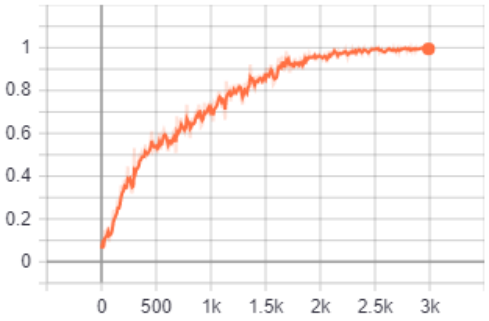


Loss_10

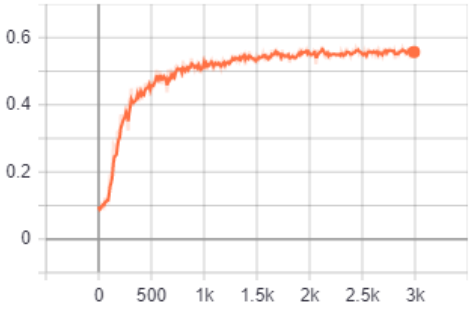


Second, I used RMSPropOptimizer with learning rate 0.001 momentum 0

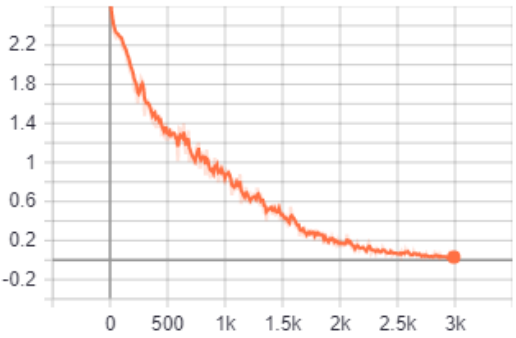
Train_Accuracy_9



Test_Accuracy_9

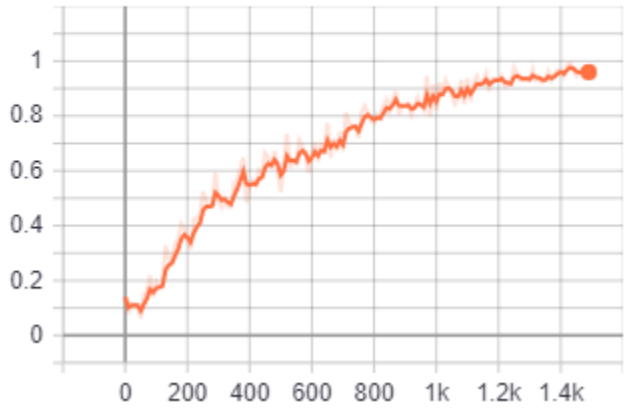


Loss_12

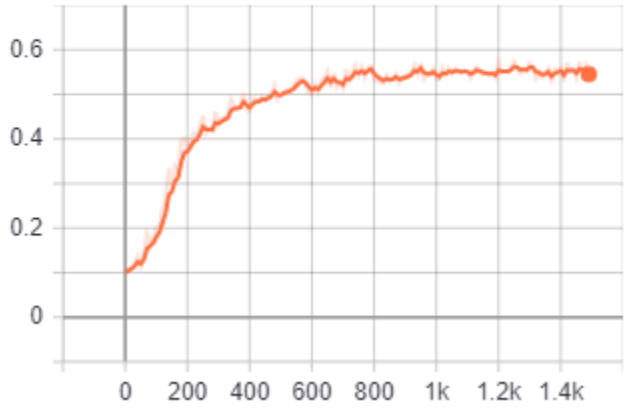


Third, I used RMSPropOptimizer with learning rate 0.001 momentum 0.5

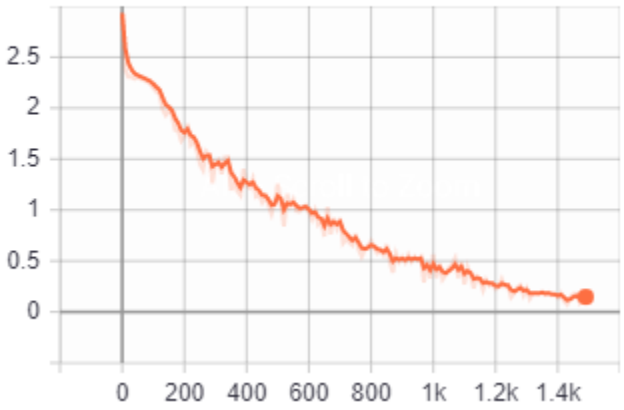
Train_Accuracy_11



Test_Accuracy_11



Loss_16

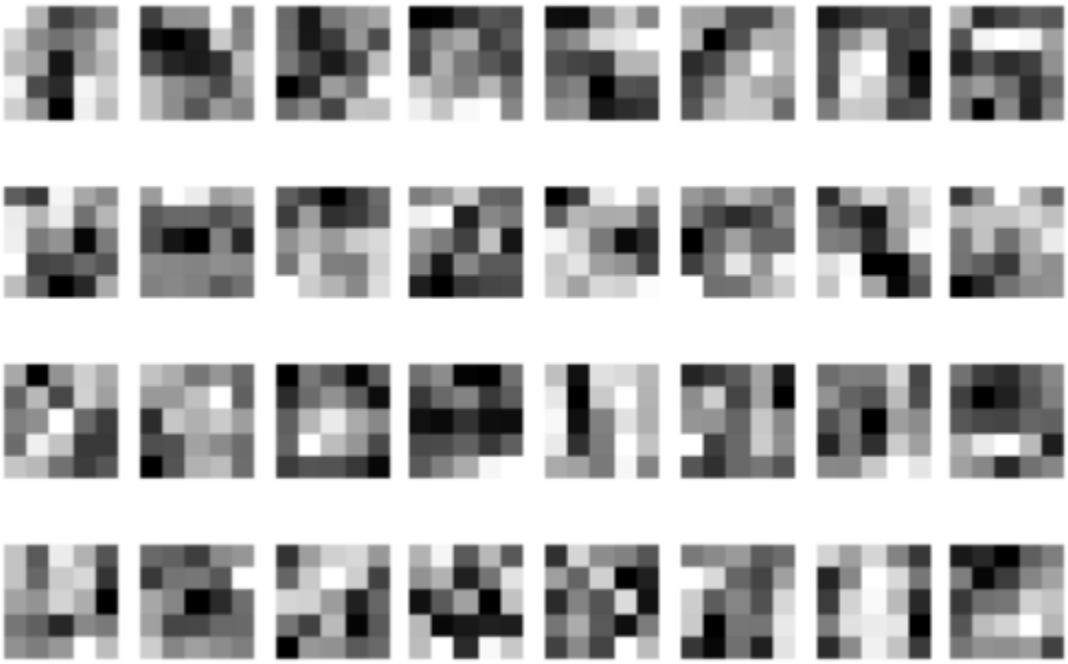


Conclusions:

It turns out that with the same leaning rate, the RMSpropOptimizer has better results and with higher momentum(less than 1), the result can be even better(train accuracy reaches 1 with less runs).

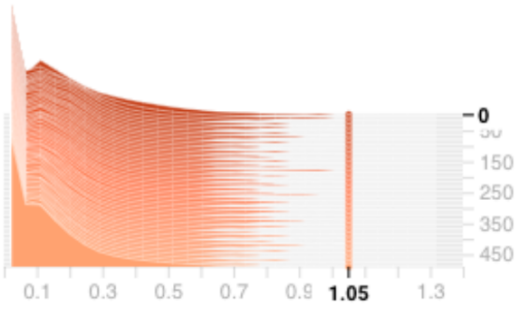
C) Visualize trained network

First layer's weights:

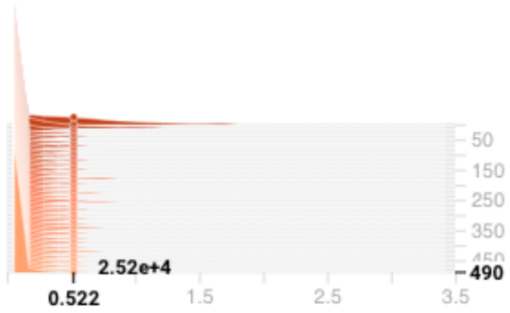


Stats for activations:

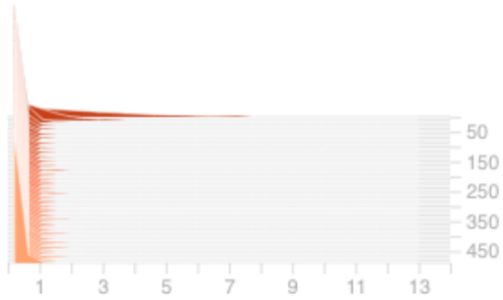
Layer1



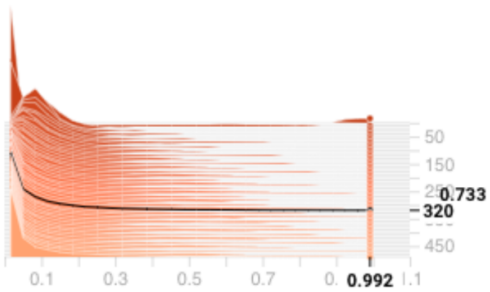
Layer2



Fc1



Fc2



2 Visualizing and Understanding Convolutional Networks

Key ideas:

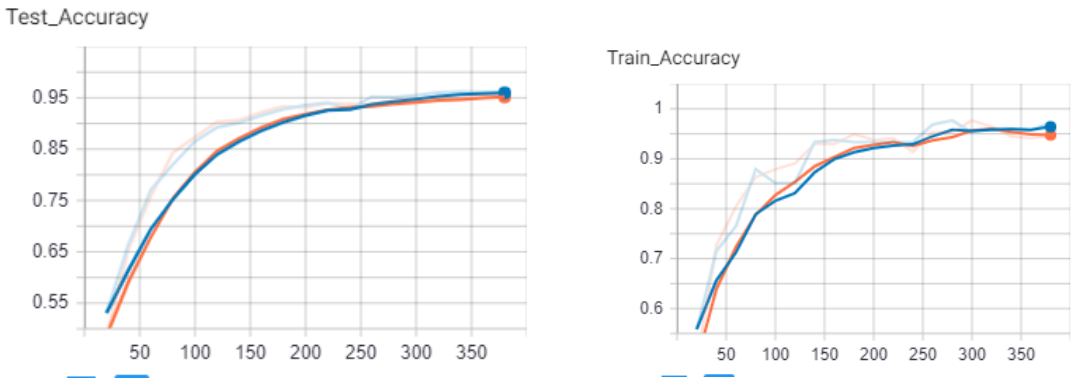
The main topic of this article is introducing a conventional technique to visualize, better understand, and probably improve convolutional networks. This article shows in detail how the ImageNet convolutional networks succeeded in the past, and it tries to figure out the reason behind its success. This new technique has a Deconvolutional Network, and this is made up of unpooling, rectification and transposed filtering stages. It can benefit the users in reversing the deep ImageNet models. After training a new model with the new technique, the model can be easily generalized for other datasets, which is the key strength of this novel technique.

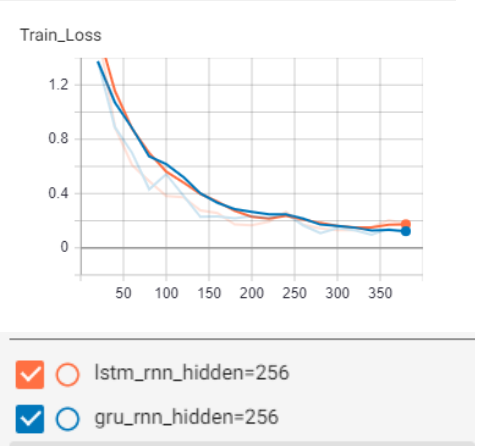
Methods:

The authors first introduce deconvnet, which is very useful to visualize and understand what the layers within a convnet see. After that, the paper experiments to see how is the convnet detecting the objects in image classification. Through covering different parts of the images and running the model, the results showed that the convet can localize the objects. Also, by changing the number of convolution and fully connected layers, the authors try to find the corresponding effects it will have on the final performance. In conclusion, the depth of the convet is the key factor that influences the overall performance of the model. In the end, the authors try to generalize the model they developed with different datasets by retraining the softmax classifier. The model has a really good result on some datasets and showed that this method is successful.

3 Build and Train an RNN on MNIST

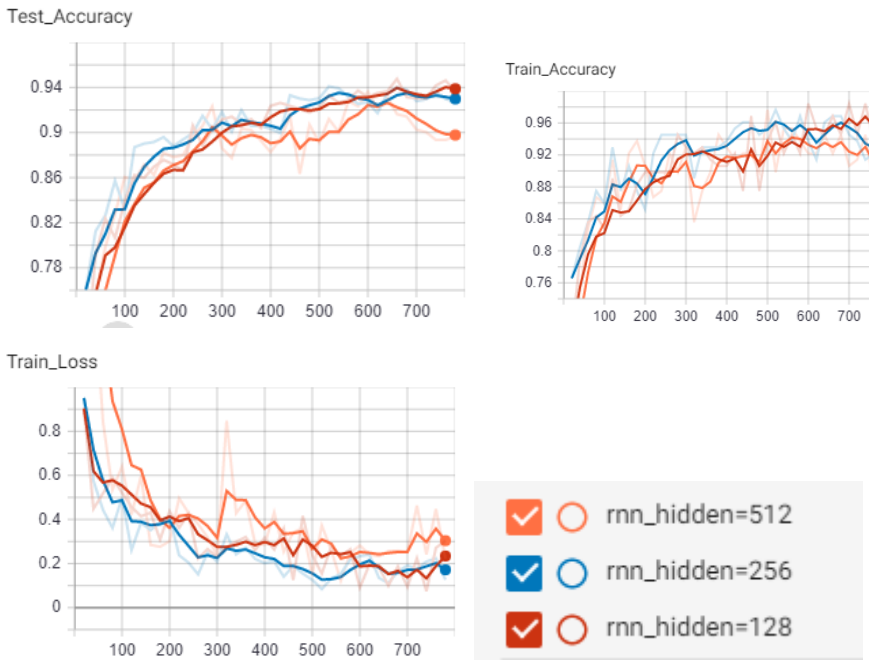
LSTM and GRU:





I used 256 hidden units for LSTM and GRU. Both of them are very effective and have good results. From the graph we can see that GRU is slightly better than LSTM but overall, they are both good.

Basic rnn with different hidden units:



With hidden units of 128 and 256, the results are better than with 512 units. And with 256 hidden units, the train loss graph shows the best result. Therefore, 256 should be the best option for this case.

RNN vs CNN:

After comparing using the stats, I think the RNN works better than the CNN in terms of accuracy. RNN may require less iterations to reach the high point of the accuracy but has the problem of overfitting.