Conclusion of the Results:

This file contains the conclusion of the result. All process and all line of codes are described in the code with inline comments and python comment for methods.

All functionality of the code and the process is described in the code with comment and python document for each method. The purpose of this document is just to summarize the result and visualization of the models with one or two or three convolutional layers.

The training and testing the results are enclosed in separated following files:

1. result_with_three_layers.txt: is a text file that contains the result of running the code in a case of having three convolutional layers.
2. result_with_two_layers.txt: is a text file that contains the result of running the code in a case of having two convolutional layers.
3. result_with_one_layer.txt: is a text file that contains the result of running the code in a case of having one convolutional layers. This result contains up to step 1700 for the accuracy changed in every 100 steps because of the time limit. But the accuracy of the model with one convolutional layer is calculated and reported for final trained model which is 0.7706.

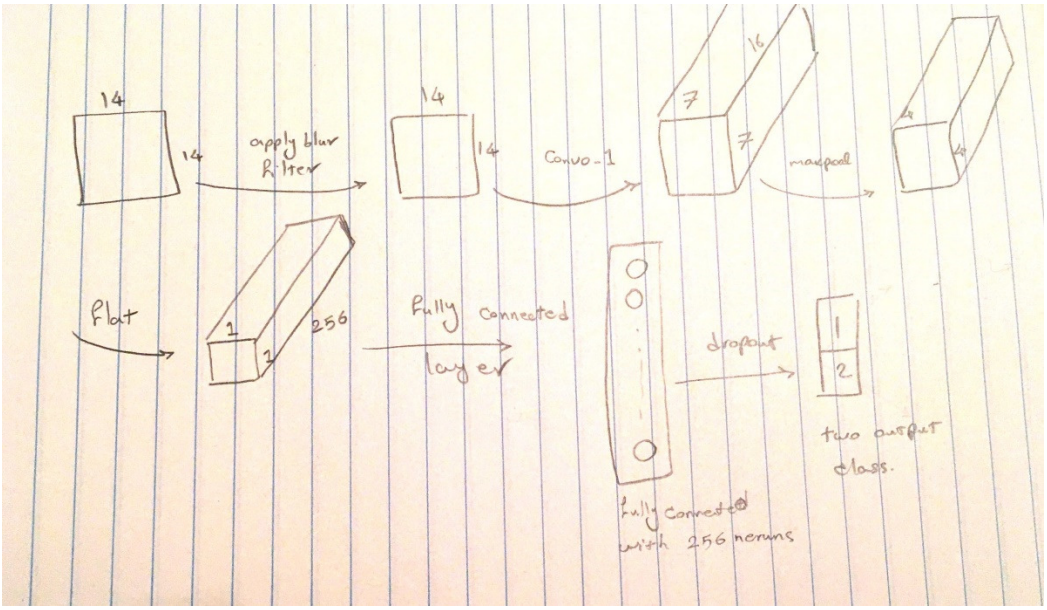
One Convolutional Layer:

The following image shows the visualization of the CNN architecture that I have with one convolutional layer. The process of the defining the tensorflow graph and the execution of it is described in the code.

In this case, after applying the blur filter we have 14 *14 images. Then we apply the convo_1 which is first module in our model as a convolutional module. The result of the convolutional operation with strides 2 * 2 and padding of same will be a set of features map in the shape of 7 * 7 * 16. After that, I defined a maxpoool module with kernel size of 2 * 2 and the strides of 2 * 2 and padding of same. The shape of the result will be 4 * 4 * 16.

Then, we have to flat the result in order to be able to use it as an input for fully connected layer. Flatted result be in shape of 1 * 1 * 256. Now, we can define a fully connected layer with 256 neurons. Then, we can define a dropout module and classified the result in two classes.

Note: all used module with their parameters and the benefit of using them is described in the code in detail as comments.



In this case the accuracy and execution time of the model after 3000 steps on MNIST dataset in a case that we have blurred images in 14 * 14 dimension is:

Accuracy: 0.7706

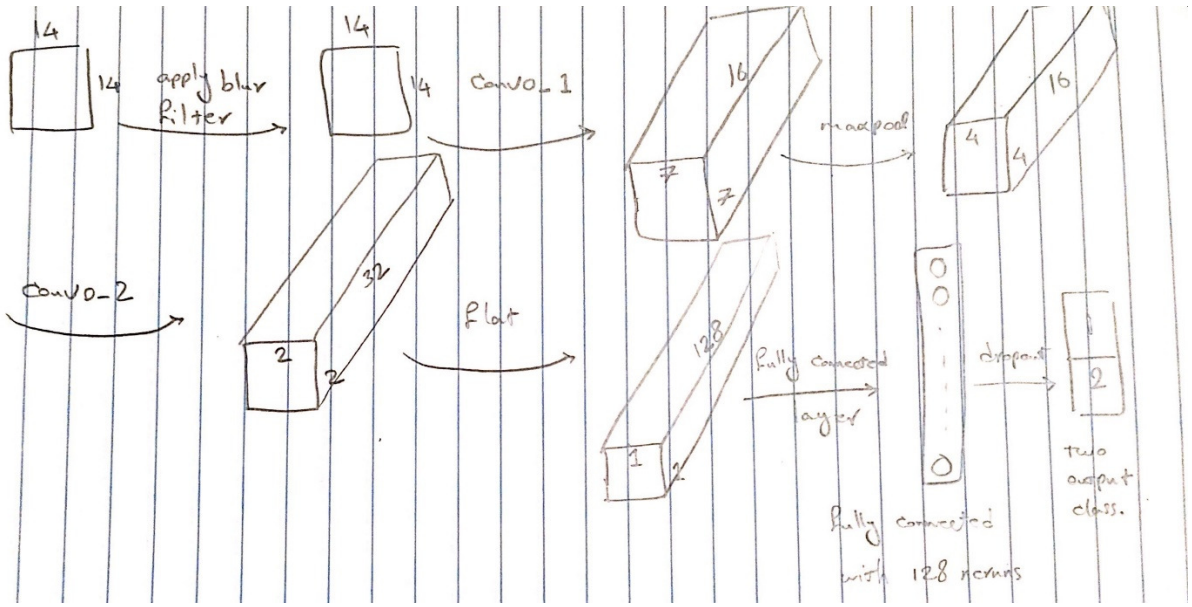
Two Convolutional Layers:

The following image shows the visualization of the CNN architecture that I have with two convolutional layers. In this case after first convolution and its maxpool module, we defined second convolutional module. In this case the input channel of the second convolutional layer is 16 and the out put channel is 32. The result of the convolutional operation with strides 2 * 2 and padding of same will be a set of features map in the shape of 7 * 7 * 16. After that, I defined a maxpoool module with kernel size of 2 * 2 and the strides of 2 * 2 and padding of same. The shape of the result will be 4 * 4 * 16.

In this case, after applying the blur filter we have 14 *14 images. Then we apply the convo_1 which is first module in our model as a convolutional module. The result of the convolutional operation with strides 2 * 2 and padding of same will be a set of features map in the shape of 7 * 7 * 16. After that, I defined a maxpoool module with kernel size of 2 * 2 and the strides of 2 * 2 and padding of same. The shape of the result will be 4 * 4 * 16. Then, we applied the second convolution module. The shape of the result will be 2 * 2 * 32. Following image shows the feature maps shape after convo_2.

Then, we have to flat the result in order to be able to use it as an input for fully connected layer. Flatted result be in shape of 1 * 1 * 128. Now, we can define a fully connected layer with 128 neurons. Then, we can define a dropout module and classified the result in two classes.

The following image shows the visualization of the CNN architecture that I have with two convolutional layers. The process of the defining the tensorflow graph and the execution of it is described in the code.



In this case the accuracy and execution time of the model after 3000 steps on MNIST dataset in a case that we have blurred images in 14 * 14 dimension is:

Accuracy: 0.9062

Execution time: 6071.086055994034

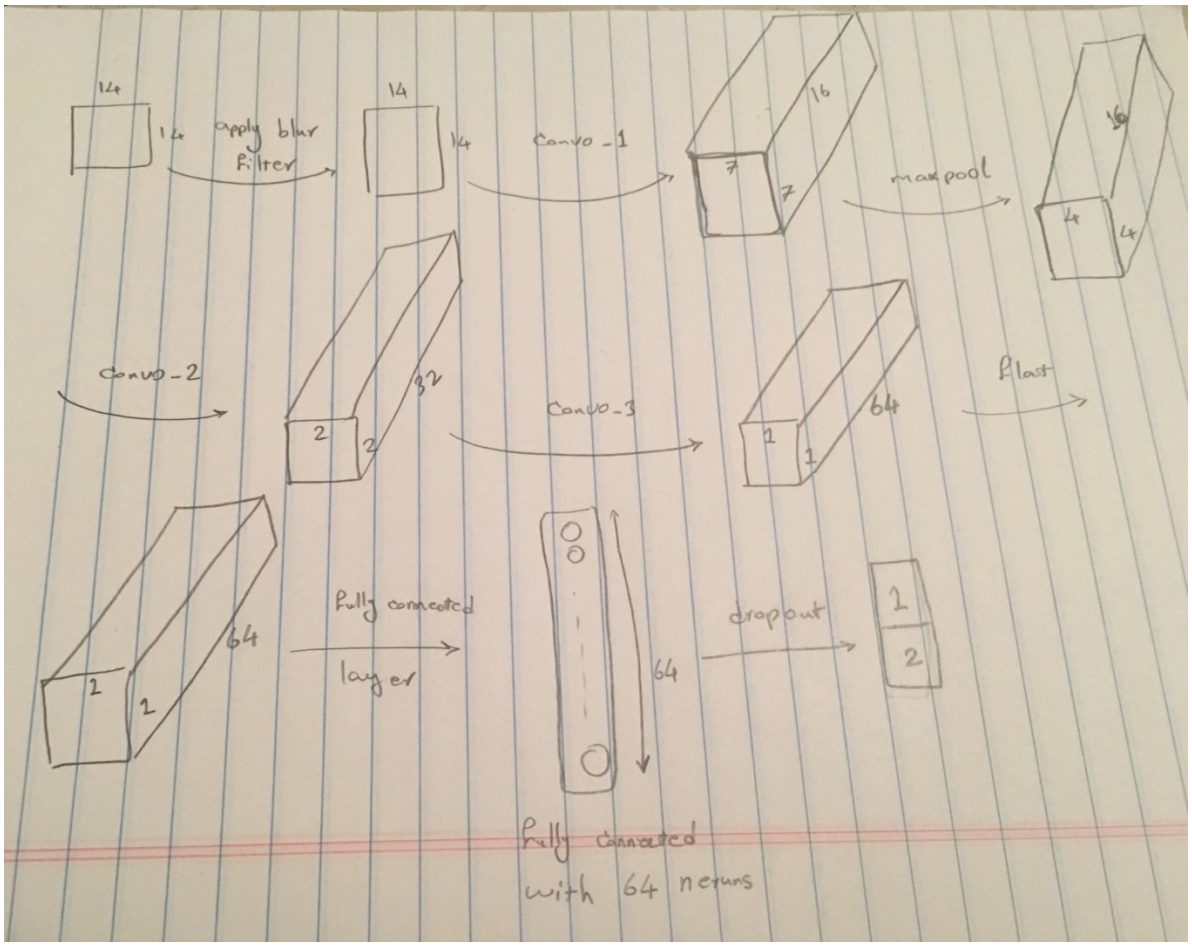
Note: reported time includes training and testing that invoked in each 100 steps. It is explained in the code.

Three Convolutional Layers:

The following image shows the visualization of the CNN architecture that I have with three convolutional layers. After applying the second convolution, we can define the third convolution module instead of flatting the result. After convo_2, the shape of feature maps is 2 * 2 * 32. By adding the third convolutional module, the feature map shape will be 1 * 1 * 64 with strides of 2 * 2 and padding of SAME.

Then, we have to flat the result in order to be able to use it as an input for fully connected layer. Flatted result be in shape of 1 * 1 * 64. Now, we can define a fully connected layer with 64 neurons. Then, we can define a dropout module and classified the result in two classes.

The following image shows the visualization of the CNN architecture that I have with two convolutional layers. The process of the defining the tensorflow graph and the execution of it is described in the code.



In this case the accuracy and execution time of the model after 3000 steps on MNIST dataset in a case that we have blurred images in 14 * 14 dimension is:

Accuracy: 0.9174

Execution time: 6531.145233154297

Note: reported time includes training and testing that invoked in each 100 steps. It is explained in the code.

Conclusion:

With three convolutional layer we reach to the best accuracy. However, the biggest execution time for training and testing in 3000 steps is also belongs to the model with three convolutional layers.

if we have two convolutional layers, the accuracy drops only 1% from 0.91 to 0.90, but the execution time for training and testing is 7.044% less than the model with the three convolutional layers.

The model with one convolutional layer has worst accuracy which is 0.7706. To conclude, if the we want to have maximum accuracy we can use model with three convolutional layers. But, if the accuracy which is one percent less than maximum accuracy is acceptable, we can use model with two convolutional layers. This samples show how deeper network with reasonable number of layers can increase the accuracy of the model. However, with having deeper network, we have more inference and training time.