

Assignment 3: CNN

Sunayana Gupta
1222389090
sgupt279@asu.edu

In this assignment, we had to vary the number of dense hidden layers, the number of neurons per hidden layer to see and also the other various parameters like filter and stride to understand why changes are occurring when we changed these numbers in the convolutional neural network(CNN) given to us for the given classification task.

I ran all the neural networks given in the reference material and decided to perform the experiment on the well-known Digit MNIST dataset. I started with the small neural network with one convolutional layer and with increasing accuracy increased the number of conv2d's to 4 to get an accuracy that is above the threshold of 99. Below is the neural network skeleton that I found to cross the threshold of 99.

```
#1. LAYER
model.add(Conv2D(filters = 32, kernel_size = (3,3), padding = 'Same', input_shape=(28, 28, 1)))
model.add(BatchNormalization())
model.add(Activation("relu"))

#2. LAYER
model.add(Conv2D(filters = 32, kernel_size = (3,3),strides=(1,1), padding = 'Same'))
model.add(BatchNormalization())
model.add(Activation("relu"))

model.add(MaxPool2D(pool_size=(2, 2)))

#3. LAYER
model.add(Conv2D(filters = 64, kernel_size = (3,3),strides=(1,1), padding = 'Same'))
model.add(BatchNormalization())
model.add(Activation("relu"))

#4. LAYER
model.add(Conv2D(filters = 64, kernel_size = (3,3), padding = 'Same'))
model.add(BatchNormalization())
model.add(Activation("relu"))
model.add(MaxPool2D(pool_size=(2, 2)))

#FULLY CONNECTED LAYER
model.add(Flatten())
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation("relu"))
model.add(Dropout(0.25))

#OUTPUT LAYER
model.add(Dense(10, activation='softmax'))
```

Figure 1 CNN model which crossed the threshold accuracy

Below are the experiments that I performed to reach this model.

In the model, we had 6 parameters to vary, namely the number of channels, the number of conv2d layers, the number of epochs, filter size, stride size, and max pool layer size. I tried training multiple neural networks and fixated on 10 epochs as that was making the error stable in the accuracy and number of epochs graph on all the models.

I decided to perform the neurons varying experiment along with varying the number of channels in the 4 conv2d layers. So I ran the above model with different channel numbers ranging from 1 to 128, and below are the graphs showing the results of the same.

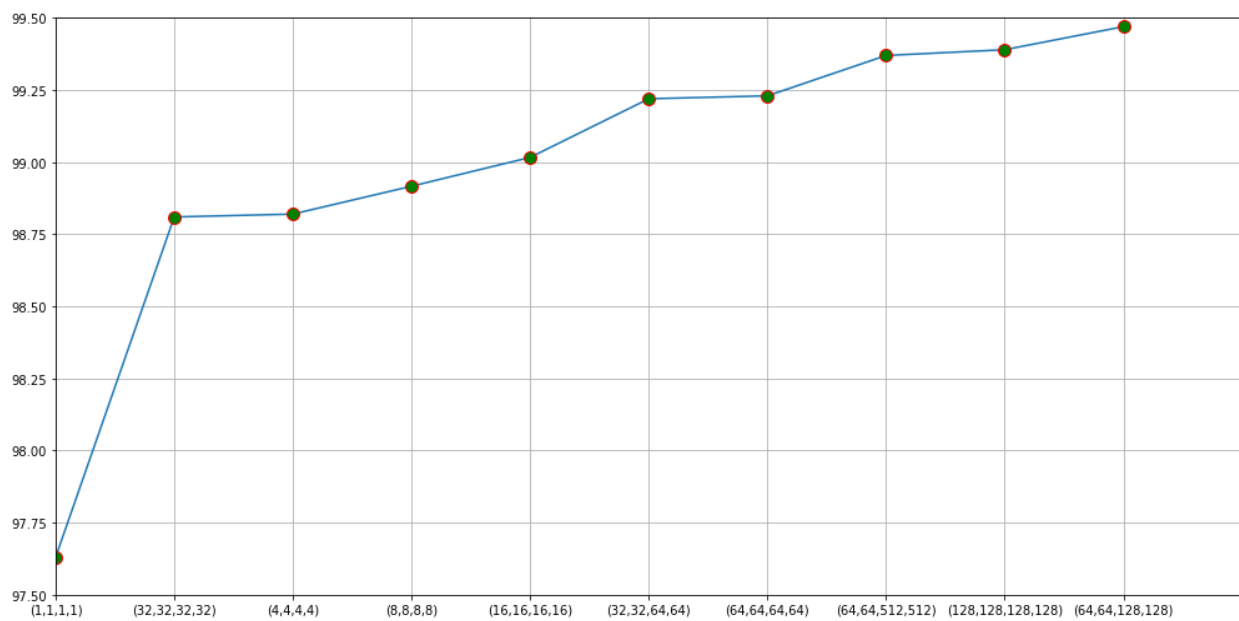


Figure 2 Graph showing accuracy vs number of channels for 4 conv2d layers

As we can see in this graph the accuracy was not varying much from 1,1,1,1 channels in all 4 layers to (64,64,128,128). But it steeply increased from 97.75 to 98.75 when I increased the channel to 32. And I had to increase the channels till (64,64,128,128) to get an accuracy of 99.25 which crossed the threshold boundary. As we can see the number of channels wasn't the most important factor as we saw in other models as the accuracy does not seem to vary much with this factor if other factors are kept constant.

The other interesting experiment was to keep the channels constant and vary the number of Conv2d layers. Below is the result I got by doing this experiment.

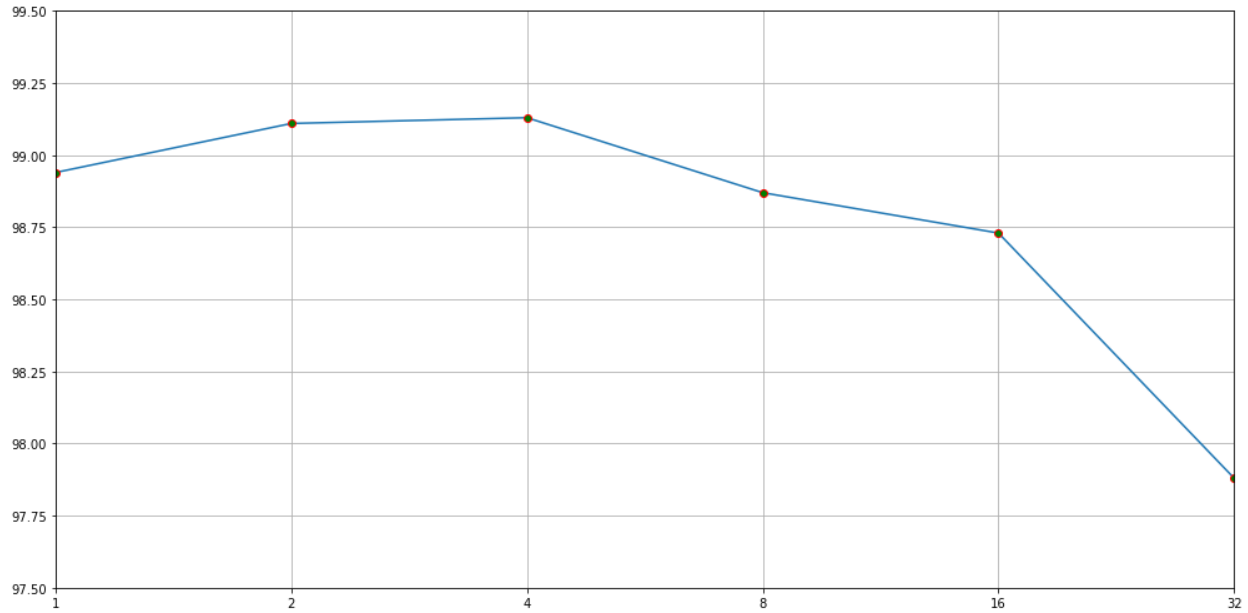


Figure 3 Graph showing accuracy vs num of layers in CNN model

Here we can see that the accuracy increased till 4 layers and after that it started decreasing. This is because there is more feature extraction that is required in this particular use case as the image itself does not have a that many features to train on. Therefore number of layers as 4 is the optimal layers for Digits MNIST and because of overfitting post this the accuracy started decreasing and kept on decreasing after after 4 and decreased steeply between 16 and 32 as it led to a lot of overfitting.

Next, I decided to vary the filter size and see the variations in accuracy in convolution if we increase the filter size to a lot then it would decrease the overlap between the convolutions and if we decrease it too much it would result in overfitting as every small pixel will be seen as an individual feature rather than a kind of blob of some pixels as features. Below is the result I got when I varied the filter size.

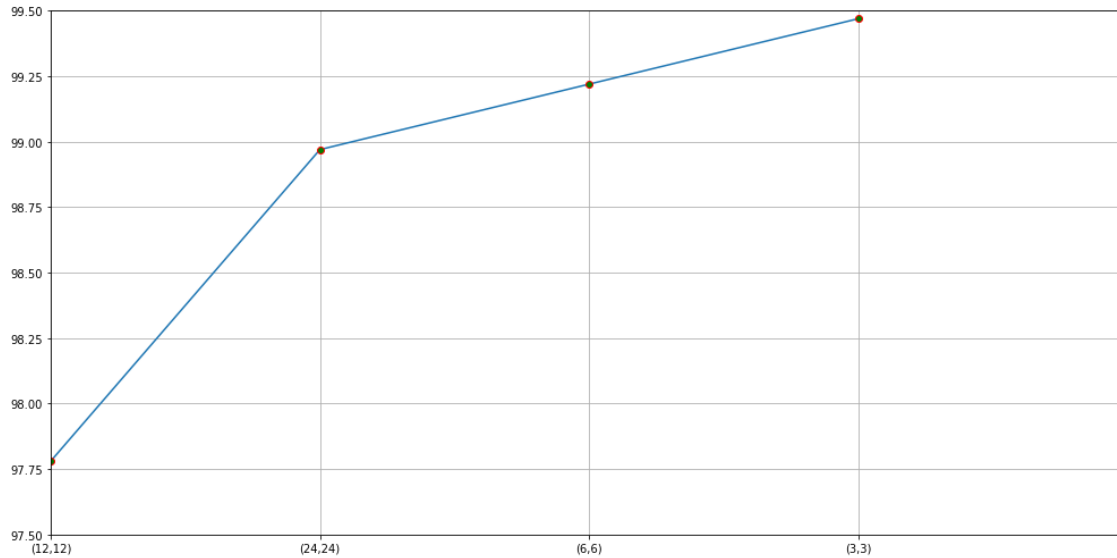


Figure 4 Graph showing accuracy vs filter size the CNN model

Since this dataset just had numbers so even when I increased the number of filters till 12,12 for an image size of 28,28 it didn't cause much of a decrement in the accuracy as there weren't a lot many features to be studied in the first place so even 12,12 gave a good accuracy of 97.75%. Further when I decreased it resulted in an increment of accuracy as expected. After this experiment, I decided to keep the filter size at 3,3 for further experiments.

The next interesting factor was stride to look at for me. Since that decides how many pixels will be skipped for the starting point of next convolution in the CNN. Decreasing this should increase the feature space by a lot and should result in a massive increase of accuracy. Below are some results that I obtained while doing this experiment.

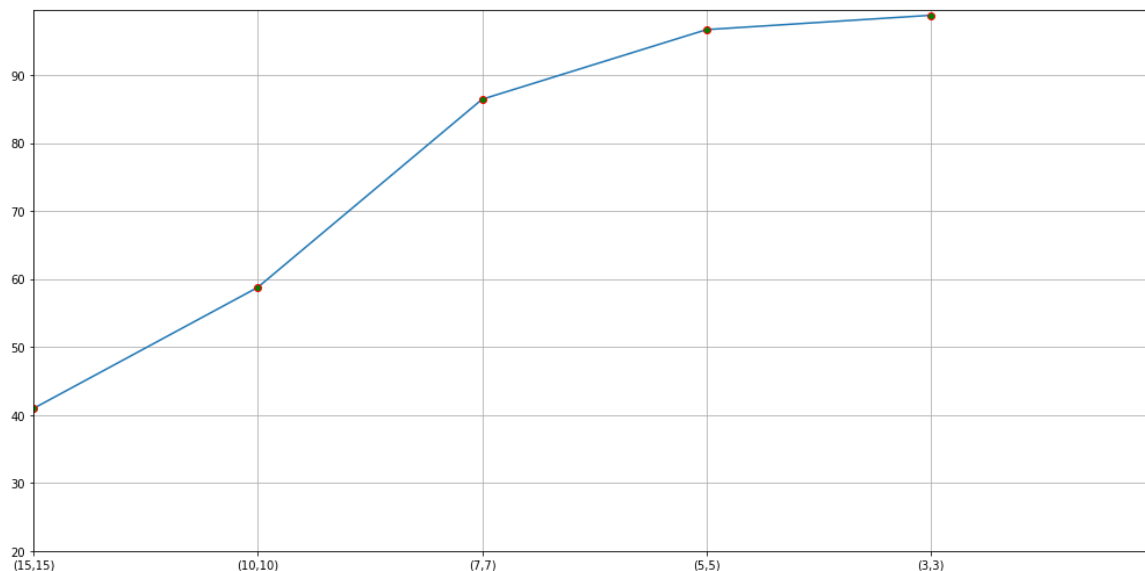


Figure 5 Graph showing accuracy vs stride size the CNN model

As we can see decreasing the stride increased the accuracy a lot as this parameter result in a large change in the feature space and hence the learning of the model. The stride of (3,3) gave a very good accuracy of 99.17% with lesser feature space compared to what we have in 1,1 stride. Hence this could be called as best stride size for the given problem.

Below are the URLs of public repository where I have uploaded the results and code

Code:

I have uploaded the code in the below public repository. Ran this in google Colab.

<https://github.com/sunayana17/CSE598IDL/blob/master/Assignment3/FinalCNNmodel.ipynb>

Results:

I have uploaded the results in GitHub repo:

Results for layer variation:

<https://github.com/sunayana17/CSE598IDL/blob/master/Assignment3/layervariation.csv>

Results for filter variation:

<https://github.com/sunayana17/CSE598IDL/blob/master/Assignment3/filtervariation.csv>

Results for stride variation:

<https://github.com/sunayana17/CSE598IDL/blob/master/Assignment3/stridevariation.csv>