Homework 3

In this assignment we shall discuss the design of my architecture for classifying the MNIST handwritten character set. The dataset comprises of 60000 training images with labels and 10000 test images. There are totally 10 different classes which belong to digits 0-9. The first and foremost step was to convert the images to greyscale and then normalize the pixel values between 0 and 1.

Next, we create our model which is composed of two blocks of a 3x3 convolution layer followed by a 2x2 max-pooling layer. The pooling layer controls the depth of the model and it significantly reduces the computational complexity. The convolution layers use the Relu activation function. After we have the input image passed through the above arrangement twice. We flatten the inputs, and we convert the 2d array into a vector and pass these inputs to a 32-neuron dense layer with output layer having 10 neurons. The model uses the Adam optimized gradient descent along with the Sparse Categorical Cross-entropy loss function. The learning rate selected was 0.001, Beta1 was 0.9, Beta2 was 0.999 and epsilon was equal to 1e-07. The model characteristics are specified below.

Layer(type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 16)	160
max_pooling2d	(None, 13, 13, 16)	0
conv2d (Conv2D)	(None, 11, 11, 16)	2320
max_pooling2d	(None, 5, 5, 16)	0
flatten (Flatten)	(None, 400)	0
dense (Dense)	(None, 32)	12832
dense (Dense)	(None, 10)	330

We have a total of 15,642 parameters to train, this model was trained for 75 epochs and the model achieved a test accuracy of 99.12%. The confusion matrix for our model is presented below.

```
[[ 975    0    1    0    0    0    2    1    1    0]
 [    0 1127    0    1    1    1    1    3    1    0]
 [    1    1 1020    2    0    0    2    6    0    0]
 [    0    0    0 1007    0    2    0    0    1    0]
 [    0    0    0    0  972    0    2    0    2    6]
 [    0    0    0    6    0  884    2    0    0    0]
 [    3    1    0    0    2    5  946    0    1    0]
 [    0    0    3    1    1    1    0 1017    1    4]
 [    3    0    2    2    0    0    0    1  964    2]
 [    1    0    0    0    5    2    0    0    1 1000]]
```

Now that we have established that this model achieved a 99.12 percent accuracy, we shall discuss some references which performed better than our model. We shall discuss the paper “Stochastic Optimization of Plain Convolutional Neural Networks with Simple methods”. The author achieved an MNIST classification accuracy of 99.79 percent and there are several differences in the author’s architecture and mine. Firstly, the author has used a dropout layer after every CNN and Maxpooling layer. The author uses a dropout factor of 0.4. Also, author has used a batch size of 256 whereas I have used 50. Also, the author has used 2048 neurons which are densely connected which gives the author about 1.5 million parameters to train. In our model, we have selected only 32 neurons which gives us about 15k parameters to train.

It can be understood that in order to further improve the accuracy of our model, we need to introduce a couple of dropout layers after our convolutional neural network layer and maxpooling layer block. Additionally, we need to make the network denser by introducing more neurons in the hidden layer.

Next, we evaluate the Convolutional Tsetlin Machine, it is a learning machine modeled after the Tsetlin Automaton. This architecture uses the Interpretable rule-based filters which incorporate the use of two image patches. Here, BX = [X-W/d] + 1 and BY = [Y-W/d] + 1, where W is the width of filter and d being the step size of the convolution. Further, it is followed by recognition which creates clauses which are then used to learn. For 8000 clauses per class, the model reached an accuracy of 99.33.