CS 250: Deep Learning Project II Report

Harsh Kumar
Mtech Res (CSA)
17082

February 21, 2020

The project required us to design CNN and NN classifier models to classify the FashionMNIST dataset.

NN Design:

My design consists of two fully connected layers and an output layer. The input layer takes flattened image in 784 nodes. The first hidden layer consists of 200 nodes followed by another hidden layer of 100 nodes. Finally, the output layer consists of 10 nodes for 10 categories of Fashion dataset. Each output of fully connected layer passes through ReLu activation function, except the output layer. The loss function used is CrossEntropyLoss and the optimizer used is Stochastic Gradient Descent with learning rate 0.001 and momentum value of 0.9

The choice of learning rate was arbitrary and worked well. The CrossEntropy loss function was used due to the categorical nature of the problem. The no. Of nodes for hidden layers was tried and tested before settling to the values. The model converges for 35 epochs.

CNN Design:

My design consists of one convolutional layer of kernel size 3x3 having 32 kernels, followed by a maxpool layer with kernel size 2x2 and stride of 2. This is followed by a fully connected layer of 100 hidden units and an output layer – 10 units.

The kernel size of 3x3 was selected arbitrary and is mostly used. The no. Of kernels was selected after trying with varying no. Of kernels.

The loss function used is CrossEntropyLoss and the optimizer used is StochasticGradientDescent with learning layer 0.001 and momentum value of 0.9. The choice of learning rate was arbitrary and worked well. The CrossEntropy function was used due to the categorical nature of the problem. The no. Of nodes for hidden layer was tried and tested before settling to the values. The model converges for 35 epochs.

Training Procedure

The training set consists of the FashionMNIST training data. For the NN model, the training images were flattened before sending to the model. In case of CNN model, the image was sent as it is. A batch size of 4 images was used. The models were trained for various no. Of epochs from 10 to 35. The loss and accuracy at various epochs were calculated and then the one with best results was selected.

Findings

We test for various no. Of nodes in the hidden layers and plot the accuracy. Also, we fix the no. Of nodes in hidden layers and plot the output of loss functions with varying no. Of epochs. The findings are shown below:

For 10 epochs: loss of nn = 0.183
For 35 epochs, loss of nn = 0.10

For 10 epochs accuracy of nn: 86.99
For 35 epochs accuracy of nn : 88.47

For 10 epochs: loss of cnn = 0.183
For 35 epochs, loss of cnn = 0.09

For 10 epochs accuracy of cnn: 88.99
For 35 epochs accuracy of cnn : 91.47