Deep Learning Lab 2018

Exercise 1: Feed Forward Neural Network

Name: Jatin Dhawan

Matriculation No. : 4523227

Introduction: Feedforward Neural Networks are also called artificial neural networks where the connection between different layers is only in forward direction and not in cycles. The information only travels forward in the network and no loops are allowed between the layers.The information first travels through the input nodes, then through the hidden nodes and finally through the output nodes. These networks are usually used for supervised learning where the output (also called as Labels) are given. Figure 1 shows the general architecture of a neural network.

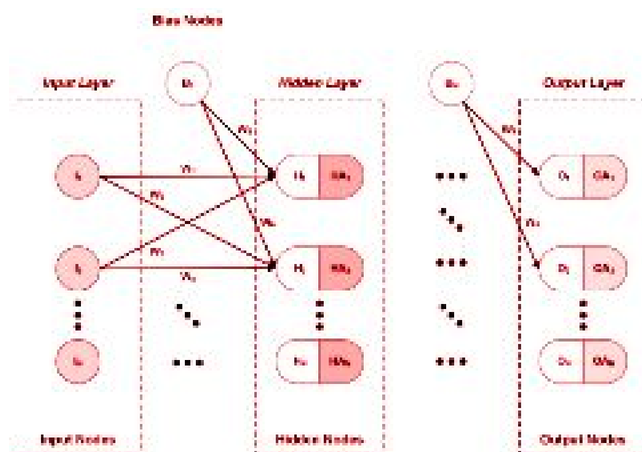

Figure 1: General architecture of a neural network

Implementation: In this exercise, we have implemented a network with 1 input layer, 3 hidden layers(called fully connected layers) with different activation functions(Relu, Sigmoid, tanh and softmax) and 1 output layer called Softmax output layer (depending on our loss function). Training of an artificial neural network is done in 4 steps:

1. **Initialization of weights and biases**: Normally one can also initialize all the weights to zero, which will make our model equivalent to a linear model and thus making all the layers symmetric. Therefore we have randomly initialized our weights and biases from a normal distribution with a mean of zero and with some standard deviation.
2. **Forward propagation**: Using the input, weights and biases, we compute the activation A and output Z with different activation functions for each layer.

3. **Loss function:** Different loss function are used to calculate the loss between the predicted output and actual output (target). Negative log likelihood (multiclass cross entropy) is used to find the loss.
4. **Back propagation:** After calculating the loss, we calculate the gradient with respect to our activations, output, weights and biases. With the help of gradients, weights are updated again for each layer

Results: After training the model for 20 epochs with training data of 50,000 images of 28*28 pixels each, and learning rates of 0.1 and 0.01 respectively, a training error of 0.5% and 6.43% is seen. The model was then trained on whole data including training and validation set, and tested on test data, a training error of 0.4% and a test error or 2.3 % was seen. Since the learning rate is not optimized, different errors are seen. Figure below shows how the validation and training error changes with epochs.

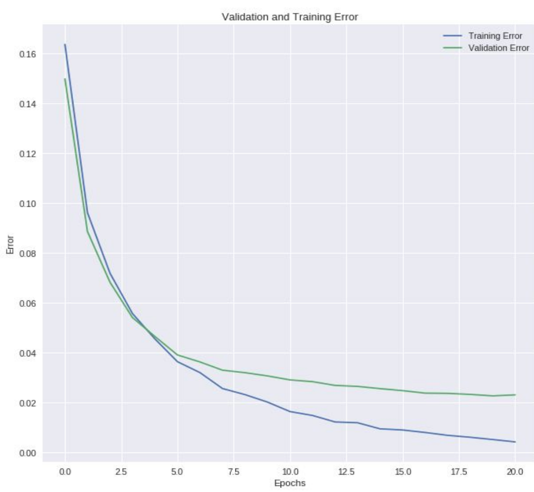

Figure 2 (Training vs Validation)
Training error: 0.55%
Validation error: 2.45%
Learning rate: 0.1
Time taken: 98.5s

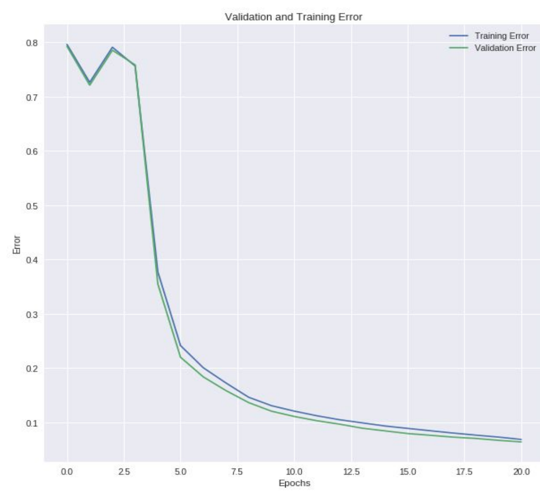

Figure 3 (Training vs Validation
Training error: 6.89%
Validation error: 6,43%
Learning rate: 0.01
Time taken: 96.2s

Bibliography

1	https://brilliant.org/wiki/feedforward-neural-networks/
3	https://www.learnopencv.com/understanding-feedforward-neural-networks/
2	https://medium.com/usf-msds/deep-learning-best-practices-1-weight-initialization-14e5c0295b94