

IE598 Deep Learning and Neural Networks

Pramod Srinivasan(psrnvsn2)

Question 1 : Single Layer Neural Network

1. Implement a single-layer neural network for the MNIST dataset and train it using stochastic gradient descent. Please include a printout of your code. This neural network should be implemented from scratch (i.e., implement the backpropagation yourself) in Python. Do not use Chainer, Tensorflow, Theano etc. to implement the neural network.
2. Describe the hyperparameters you tried and which ones performed the best (number of hidden units, learning rate, etc.). Examples of types of hidden units include sigmoidal and RELU. Sigmoidal hidden units are $\sigma(z) = \exp(z) / (1 + \exp(z))$ and RELU hidden units are $\sigma(z) = \max(z, 0)$.
3. Report test accuracy.

We first begin with the introduction of the 3-layer neural network with one input layer, one hidden layer and one output layer.

- The number of nodes in the input layer is determined by the dimensionality of the input data = 784
- The number of nodes in the output layer is decided by the number of classes = 10
- The number of units in the hidden layer is parameterized = {10, 100, 150, 200}

The neural network uses forward propagation to make predictions. Every layer consists of a linear operation (matrix multiplication) followed by a non-linear operation carried out by an activation function. We have chosen RELU (Rectified Linear Unit) in the first layer and SOFTMAX function in the final layer. summarized the following equations:

$$z_1 = w_1 X + b_1$$

$$a_2 = \text{relu}(z_1)$$

$$z_2 = w_2 a_2 + b_2$$

$$g = \hat{y} = \text{softmax}(z_2)$$

We note that w_1 , w_2 , b_1 and b_2 are network parameters needed to be learnt from the training data. The hidden layer nodes are tuned so as to minimize the error on the training data. The backpropagation works as follows :

$$\delta_3 = \hat{y} - y$$

$$\delta_2 = \delta_3 * W2 * \text{reluGrad}(z1)$$

Here reluGrad is the gradient of the relu function.

Subsequently, we calculate the gradients of each of these parameters w_1 , w_2 , b_1 and b_2 and update them with the learning rate $\eta = 0.001$. The number of epochs was chosen to be 35.

Final Test Accuracy : **97.76%**.

The console output for each epoch is given below :

```
At epoch number 0, train accuracy: 0.822716666667 test accuracy 0.9109
At epoch number 1, train accuracy: 0.906433333333 test accuracy 0.9142
At epoch number 2, train accuracy: 0.914766666667 test accuracy 0.9222
At epoch number 3, train accuracy: 0.922183333333 test accuracy 0.9267
At epoch number 4, train accuracy: 0.92945 test accuracy 0.9335
At epoch number 5, train accuracy: 0.9373 test accuracy 0.9383
At epoch number 6, train accuracy: 0.944883333333 test accuracy 0.9461
At epoch number 7, train accuracy: 0.951083333333 test accuracy 0.9513
At epoch number 8, train accuracy: 0.955966666667 test accuracy 0.9536
At epoch number 9, train accuracy: 0.959933333333 test accuracy 0.9588
At epoch number 10, train accuracy: 0.964033333333 test accuracy 0.9632
At epoch number 11, train accuracy: 0.967066666667 test accuracy 0.9646
At epoch number 12, train accuracy: 0.969966666667 test accuracy 0.9656
At epoch number 13, train accuracy: 0.972333333333 test accuracy 0.9678
At epoch number 14, train accuracy: 0.974166666667 test accuracy 0.9699
At epoch number 15, train accuracy: 0.97635 test accuracy 0.9698
At epoch number 16, train accuracy: 0.97815 test accuracy 0.9709
At epoch number 17, train accuracy: 0.979566666667 test accuracy 0.9728
At epoch number 18, train accuracy: 0.981083333333 test accuracy 0.9733
At epoch number 19, train accuracy: 0.982183333333 test accuracy 0.9742
At epoch number 20, train accuracy: 0.983233333333 test accuracy 0.9736
At epoch number 21, train accuracy: 0.984316666667 test accuracy 0.9755
At epoch number 22, train accuracy: 0.985166666667 test accuracy 0.9774
At epoch number 23, train accuracy: 0.986116666667 test accuracy 0.9754
At epoch number 24, train accuracy: 0.98685 test accuracy 0.9767
At epoch number 25, train accuracy: 0.9877 test accuracy 0.9761
At epoch number 26, train accuracy: 0.9887 test accuracy 0.9765
At epoch number 27, train accuracy: 0.98885 test accuracy 0.9777
At epoch number 28, train accuracy: 0.98985 test accuracy 0.9771
At epoch number 29, train accuracy: 0.99055 test accuracy 0.9779
At epoch number 30, train accuracy: 0.990633333333 test accuracy 0.9782
At epoch number 31, train accuracy: 0.991566666667 test accuracy 0.978
```

At epoch number 32, train accuracy: 0.9923 test accuracy 0.9774
At epoch number 33, train accuracy: 0.992833333333 test accuracy 0.9781
At epoch number 34, train accuracy: 0.993216666667 test accuracy 0.9767
At epoch number 35, train accuracy: 0.993616666667 test accuracy 0.9788