CS 535 Deep Learning Assignment 2

Zhou Fang

February 19, 2018

1 Calculation for network

Suppose the first hidden layer generating Z_1 and the final layer generating Z_2. ReLU function is RL. Sigmoid function is p. Cross entropy layer is Loss.
W_1 is a matrix with the number of rows equal to the input dimensionality and the number of columns equal to the number of hidden units. b_1 is a matrix with the number of rows equal to the number of hidden units and the number of columns equal to 1. W_2 is a matrix with the number of rows equal to the number of hidden units and the number of columns equal to 1.b_2 is a matrix with the number of rows equal to 1 and the number of columns equal to 1.
Forward process is

$$Z_1 = W_1^T x + b \tag{1}$$

$$RL = max(0, Z_1) \tag{2}$$

$$Z_2 = W_2^T RL + c \tag{3}$$

$$p = \frac{1}{1 + exp(-Z_2)} \tag{4}$$

$$Loss = ylogp + (1 - y)log(1 - p) \tag{5}$$

The backpropagation process to compute W_1 and W_2 is

$$W_2 = \frac{\partial Loss}{\partial p}\frac{\partial p}{\partial Z_2}\frac{\partial Z_2}{\partial RL} \tag{6}$$

$$W_1 = \frac{\partial Loss}{\partial p}\frac{\partial p}{\partial Z_2}\frac{\partial Z_2}{\partial RL}\frac{\partial RL}{\partial Z_1}\frac{\partial Z_1}{\partial x} \tag{7}$$

2 Gradient descent training

Stochastic mini-batch gradient descent performs updates on weights for every mini-batch size N_M from total training data, and then average loss function of these N_M numbers of training data to update weights.

$$\nabla \widetilde{E} \approx \sum_{i \in N_M} \frac{\partial L(f(x_i; W), y_i)}{\partial W}, N_M \subset 1, ..., n \tag{8}$$

The basic update momentum approach is shown below. μ is momentum, and η is learning rate. Every training data in mini-batch set will have an update in weight W. Average of all these update in weight W as partial of update to weight W. Finally, the update of weight W will be momentum times previous weight W minus learning rate times averaged updated subgradient weight.

$$W = \mu W - \eta \nabla_W J(W, (x^{(i:i+N_M)}, y^{(i:i+N_M)}) \tag{9}$$

$$J(W, (x^{(i:i+N_M)}, y^{(i:i+N_M)}) = \frac{1}{N_M} \sum_{i=0}^{N_M} J(W, (x_i, y_i)) \tag{10}$$

3 Train and test the network

20 epochs are performed and the best test accuracy is 80.20% that can be achieved when the numbers of hidden units is 100. learning rate is 0.01. Momentum is 0.8. mini-batch size is 10.

4 Training monitoring

In this section, we monitor and evaluate the training objective, testing objective, training error(train accuracy) and testing error(test accuracy). The numbers of hidden units is 100. learning rate is 0.001. Momentum is 0.8. mini-batch size is 10.

Table 1: Performance of MLP

epoch	train loss	test loss	train accuracy	test accuracy
1	0.524	0.527	74.95%	74.55%
2	0.479	0.496	77.70%	77.00%
3	0.453	0.483	79.20%	77.95%
4	0.432	0.476	80.62%	78.50%
5	0.414	0.471	81.79%	78.90%
6	0.398	0.471	82.42%	79.10%
7	0.383	0.472	83.16%	79.45%
8	0.368	0.472	84.00%	79.20%
9	0.355	0.476	84.71%	79.35%
10	0.343	0.481	85.47%	79.65%
11	0.331	0.487	85.80%	79.70%
12	0.319	0.494	86.41%	79.60%
13	0.307	0.501	87.08%	79.60%
14	0.295	0.510	87.62%	79.70%
15	0.282	0.519	88.08%	79.90%
16	0.272	0.530	88.53%	80.20%
17	0.259	0.540	89.22%	80.10%
18	0.247	0.552	89.73%	80.00%
19	0.233	0.563	90.26%	79.75%
20	0.221	0.576	90.68%	79.85%

5 Tuning parameters

5.1 Test accuracy with different numbers of batch size

The test accuracy with different number of batch size is shown in figure 1. Batch size can be 5, 10, 20, 100, 200, 1000, 2000. The numbers of hidden units is 100. Learning rate is 0.001. Momentum is 0.8. Batch size of 10 is chose to get the best test accuracy.

From figure 1, we can see, as increase of batch size, the worse performance it has. When batch size is small enough, there are no improvement to test accuracy.

5.2 Test accuracy with different learning rate

The test accuracy with different learning rate is shown in figure 2. Learning rate can be 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1. The numbers of hidden units is 100. Momentum is 0.8. Batch size is 10. Learning rate of 0.001 is chose to get the best test accuracy.

From figure 2, we can see, as increase of learning rate, the better performance it has. However, when learning rate is too large to be 0.1, test accuracy is not good because of overflow problem.

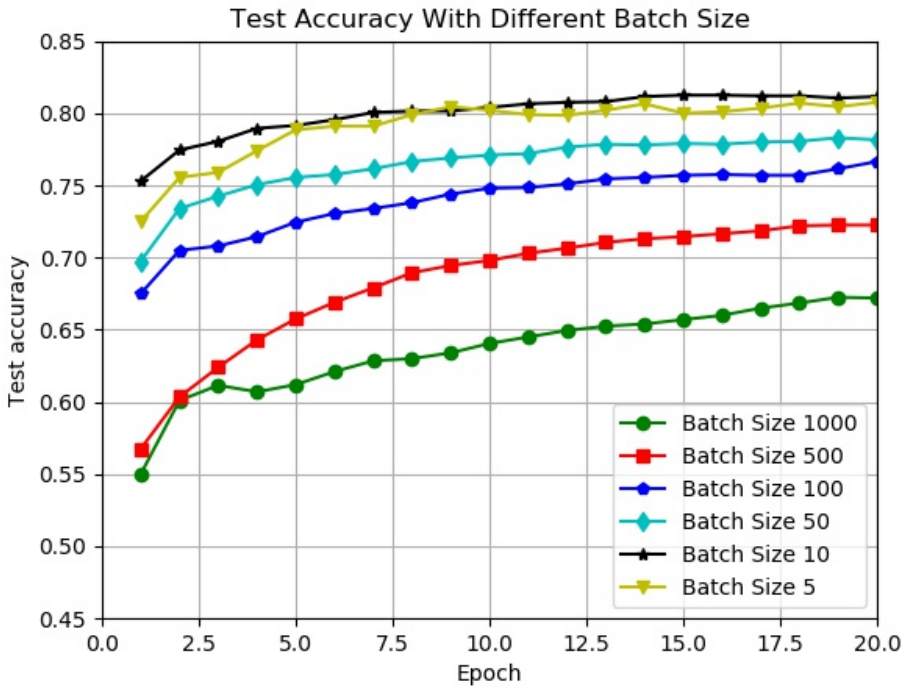

Figure 1: Test Accuracy With Different Batch Size

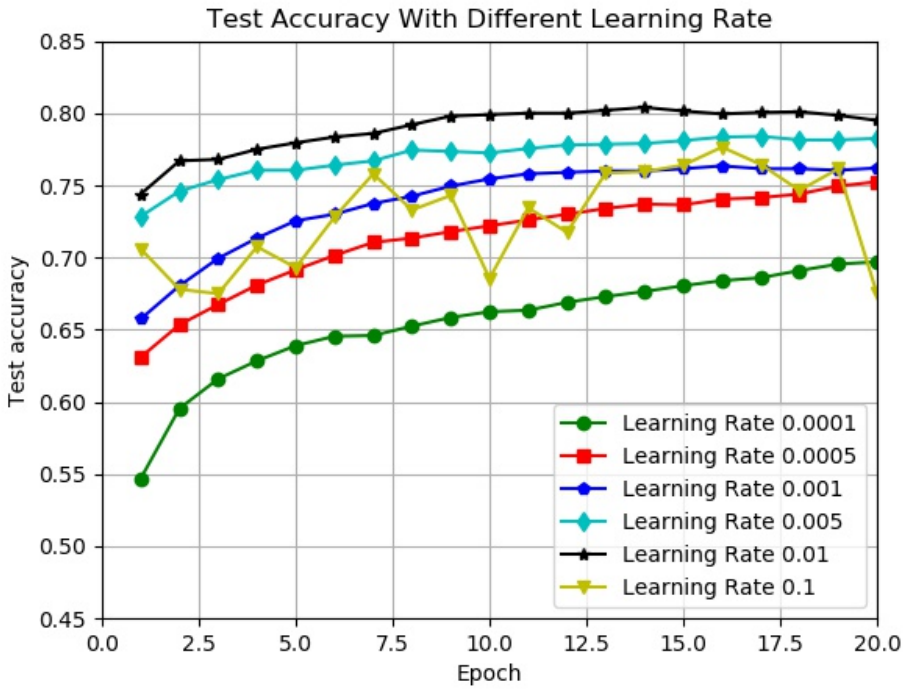

Figure 2: Test Accuracy With Different Learning Rate

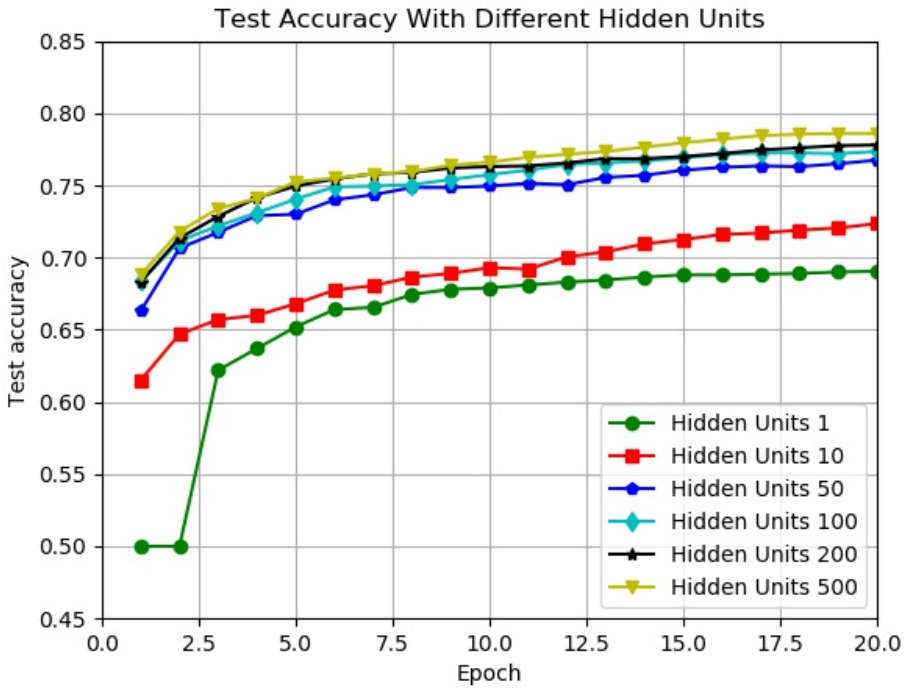

Figure 3: Test Accuracy With Different Hidden Units

5.3 Test accuracy with different numbers of hidden units

The test accuracy with different number of hidden units is shown in figure 3. Hidden units can be 1, 10, 50, 100, 200, 500. Learning rate is 0.001. Momentum is 0.8. Batch size is 10. Hidden units of 100 is chose to get the best test accuracy.

From figure 3, we can see, as increase of hidden units, the better performance it has. When hidden units size is large enough, there are no improvement to test accuracy. When hidden units is too large, it is very slow to run code.

6 Discussion

Finally, we choose batch size 10, learning rate 0.001, and hidden units 100 to get best test accuracy. The performance of the neural network is good that can achieve test accuracy above 80%.