Portland State University

Deep Learning: Computational Structures and Programming

Winter 2021

Project 1

Author:
Hermann Yepdjio

Professor:
Dr. Suresh Singh

January 18, 2021

Submission Report

1) Link to my colab notebook:
`https://colab.research.google.com/drive/1Bx4EqzWYnoHE4LQCUDEdUDcqPrIBqTMK?usp=sharing`

2) We implemented two different networks with the following configurations:

- **Network 1: one input layer, one hidden layer, and one output layer of 784, 1024, and 10 neurons respectively. The activation and loss functions used were ReLU and cross entropy respectively. We chose optim.SGD as the optimizer with momentum 0 and learning rate 0.001. We used a mini-batch size of 30.**
- **Network 2: similar to Network 1 except that we added another hidden layer of 1024 neurons.**

After training (for 10 epochs) and testing both models on the FashionMNIST dataset, we obtained accuracies of 80% and 79% respectively for Network 1 and Network 2. The fact that the latter performed slightly worse than the former despite having more neurons is probably due to the first having more neurons than it was needed. This probably has caused it to start memorizing rather than generalizing the training set.
Another explanation could be that, Network 2 needed to train for more epochs given its higher number of neurons. In fact, when looking at the loss after each epoch, both models started at $loss \approx 1.1$ at epoch 1, but Network 1 reached $loss \approx 0.09$ at epoch 10 while Network 2 was still at $loss \approx 0.4$.

3) Here, we used Network 2 mentioned above and adjusted some its parameters as indicated in Table 1. The results that we obtained as also recorded in the same table.
As we can see, the best accuracy (88%) was obtained when using ReLU activation function, $mini-batch-size = 1$ and $learning-rate = 0.01$ while the worst accuracy (10) mostly was obtained when using $learning - rate = 1.0$. The difference can be explained by the fact that a learning rate of 1.0 is probably too high and prevent the network to learn effectively during training because it is trying to learning too fast.

Table 1

Mini_Batch Size / Learning Rate	1 (ReLU)	10 (ReLU)	1000 (ReLU)	1 (Sig)	10 (Sig)	1000 (Sig)
1.0	10%	10%	10%	10%	86%	67%
0.1	10%	87%	81%	87%	86%	61%
0.01	88%	87%	67%	85%	82%	10%
0.001	87%	83%	43%	81%	62%	19%

4) In this part of the assignment we polluted the training set by Picking 9 sets of 1% of images from each of the 10 categories and adding them to the other 9 categories. We used the best model mentioned in the previous paragraph and we obtained exactly the same result (88% classification accuracy). However, if we look at the loss after each epoch, when the data is not polluted, the loss varies between 0.805 (epoch 1) and 0.203 (epoch 10). On the other hand when the data is polluted the loss varies between 1.045 (epoch 1) and 0.511 (epoch 10). Therefore, in the second scenario it is possible that the model realized that a small portion of data in each category did not belong that category and ignored that portion when updating the weights and bias.