CSCE4613/5613 – Artificial Intelligence

Fall 2021 Assignment 4

Name: Rashmitha Thoranala

Mail-id: rthorana@uark.edu

Id: 010962830

I did not use any late days for submission of assignment so, number of late days=0

Implementation:

I studied about the neural networks and the backpropagation methods in this assignment. I used Google collab as in this assignment requires implementation in Pytorch. The file Homework4.ipynb contains the details of implementation.

1. Boolean functions:

a) The formula for sigmoid function and its derivative is given below:

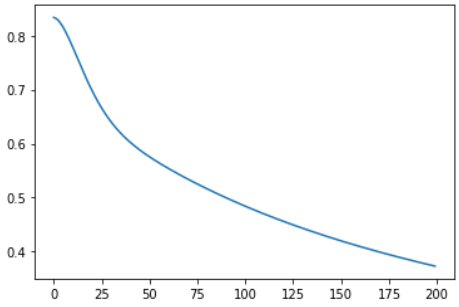
Sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

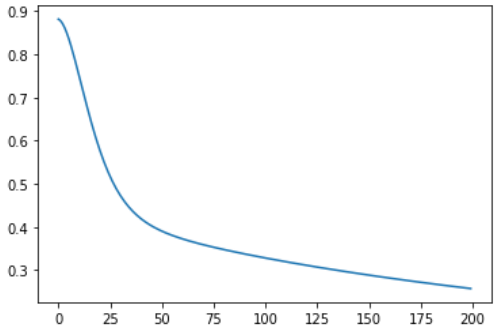
Derivative of sigmoid function:

$$\sigma'(x)=(1-\sigma(x))\sigma(x)$$

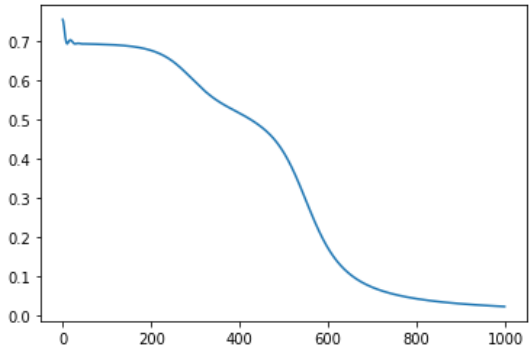
b) For AND operator, I used hidden dimensions as 0, number of iterations is 200, learning rate is 0.01 and threshold is 0.5. I got the final accuracy as 100.00. I plotted the training losses in the curve graph as:



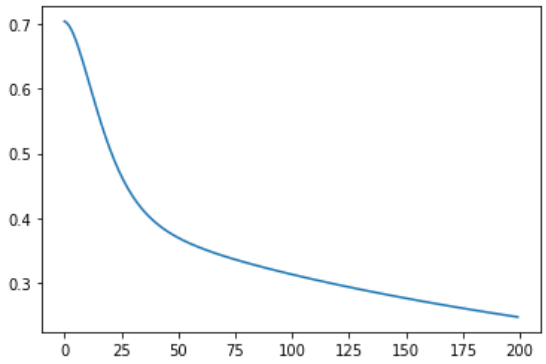
c) For OR operator, I used hidden dimensions as 0, number of iterations is 200, learning rate is 0.01 and threshold is 0.5. I got the final accuracy as 100.00. I plotted the training losses in the curve graph as:



d) For XOR operator, I used hidden dimensions as 2, number of iterations is 1000, learning rate is 0.1 and threshold is 0.5. I got the final accuracy as 100.00. I plotted the training losses in the curve graph as:



e) For NOR operator, I used hidden dimensions as 0, number of iterations is 200, learning rate is 0.01 and threshold is 0.5. I got the final accuracy as 100.00. I plotted the training losses in the curve graph as:



2) Digit Classification:

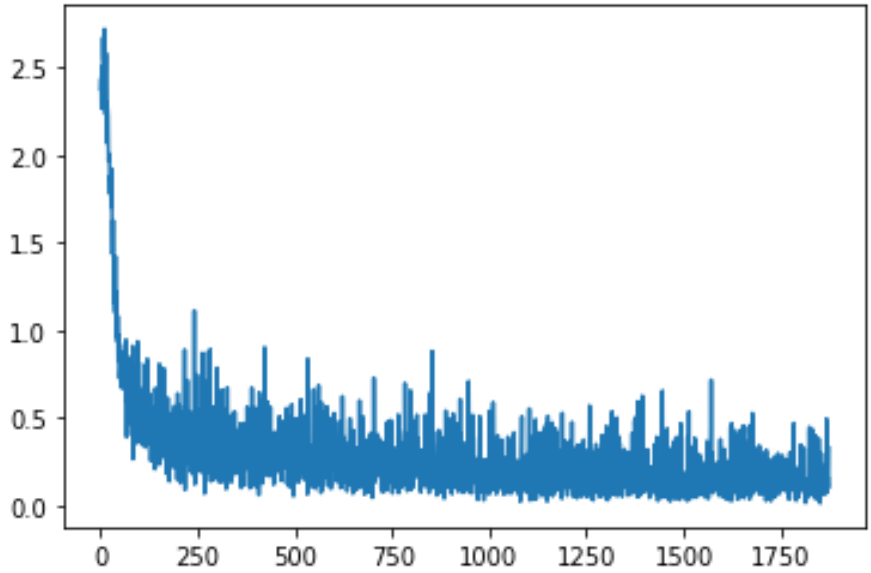
a) Formula of number of parameters in each layer = D_{in} * D_{out} + D_{out}

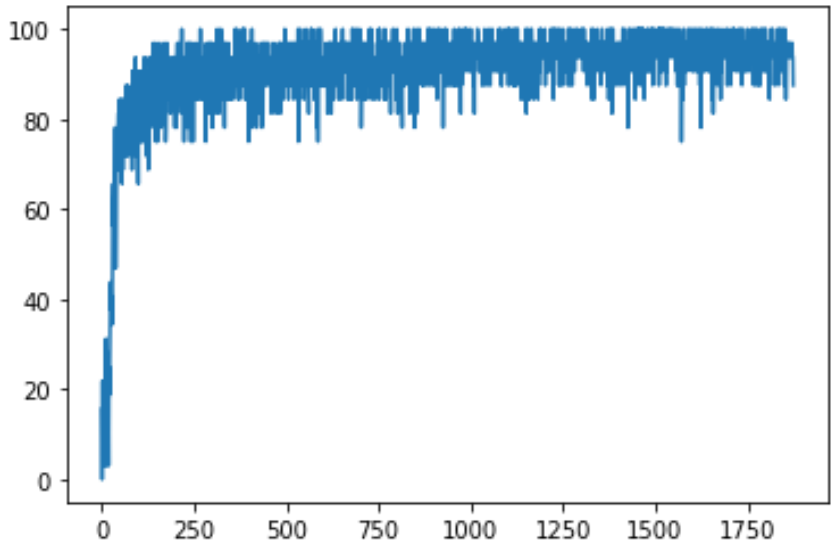
The number of parameters in the first layer = 100,480

The number of parameters in the second layer = 1290

Total number of parameters of the model = 101,770

b) I have configurated hyperparameters as batch_size = 32 and n_epochs = 1, learning_rate = 0.1and then I trained the model with these values. The average time is 0.0014. The plots for training losses and accuracies are shown below respectively:





c) The final accuracy is 95.83 and the average inference time is 0.0004. The confusion matrix is shown below:

```
Iter [1/10000]. Accuracy: 100.00
Iter [2001/10000]. Accuracy: 100.00
Iter [4001/10000]. Accuracy: 100.00
Iter [6001/10000]. Accuracy: 100.00
Iter [8001/10000]. Accuracy: 100.00
Final Accuracy: 95.83
avg_inference time : 0.0004
array([[ 955,    0,    8,    0,    1,    5,    7,    1,    3,    3],
       [   0, 1125,    2,    1,    1,    4,    3,    8,    4,    8],
       [   0,    2,  987,   10,    3,    0,    1,   20,    4,    1],
       [   2,    1,    7,  972,    0,   16,    1,    2,   19,   13],
       [   0,    0,    6,    0,  923,    1,    5,    1,    2,    9],
       [  10,    1,    1,    8,    1,  847,   11,    1,    9,   11],
       [   7,    3,    9,    2,    6,    8,  924,    0,    8,    1],
       [   3,    2,    6,   12,    3,    1,    1,  984,   11,    9],
       [   1,    1,    4,    4,    3,    6,    5,    0,  913,    1],
       [   2,    0,    2,    1,   41,    4,    0,   11,    1,  953]])
```

d) The predictions for 10 samples are:

```
label 7 is correctly predicted
label 2 is correctly predicted
label 1 is correctly predicted
label 0 is correctly predicted
label 4 is correctly predicted
label 1 is correctly predicted
label 4 is correctly predicted
label 9 is correctly predicted
label 5 is incorrectly predicted
label 9 is correctly predicted
```

3) Back Propagation:

a) I implemented forward and backward functions for sigmoid layer.

b) Implement forward and backward functions for a fully connected layer (dense layer).

c) I tested our implementation and used training and testing framework of previous implementation and got final accuracy as 91.83