جامعـــة خليفـــة
Khalifa University

COSC 606: Homework 2 Results

By - Adarsh Ghimire (100058927)

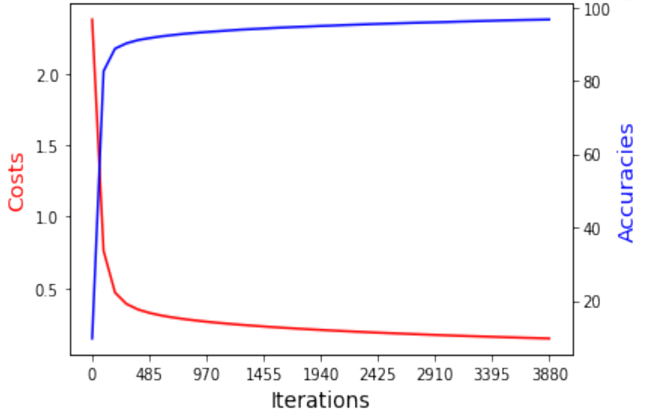
06 MARCH 2021

Batch Gradient Descent on MNIST Dataset Details and Results

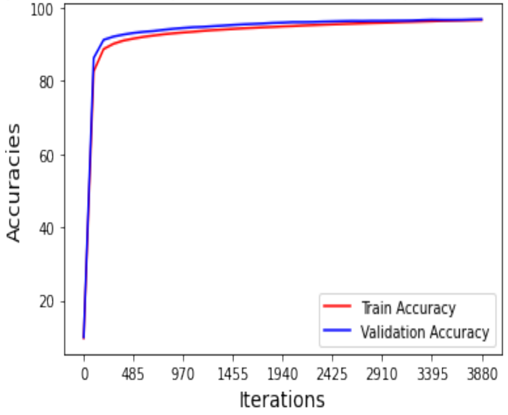
Network Details	**Network 1**	**Network 2**	**Network 3**
Hidden Layers / Neurons	1 / 64	2 / 64, 32	3 / 64, 32, 32
Network Initialization	Xavier	Xavier	Xavier
Learning rate	1.0	1.0	1.0
Epochs	4000	4000	4000
Training data size	54,000	54,000	54,000
Validation data size	6,000	6,000	6,000
Test data size	10,000	10,000	10,000
Hidden layers activation	Sigmoid	Sigmoid	Sigmoid
Output layer activation	Softmax	Softmax	Softmax
Loss	Categorical Cross entropy	Categorical Cross entropy	Categorical Cross entropy
Training Accuracy (%)	96.68	97.24	97.23
Validation Accuracy (%)	96.83	97.10	96.31
Test Accuracy (%)	96.08	96.26	95.94
Test Error Rate (%)	3.920	3.740	4.060

Network Convergence Graph

Network 1

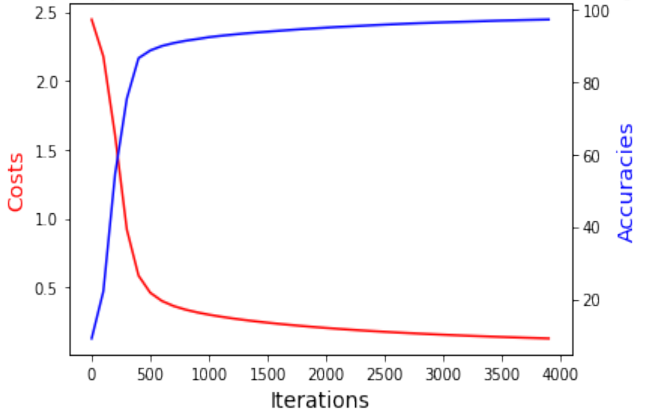


Training Cost/Accuracy over iterations for 1 hidden layer network with Learning rate =1.0

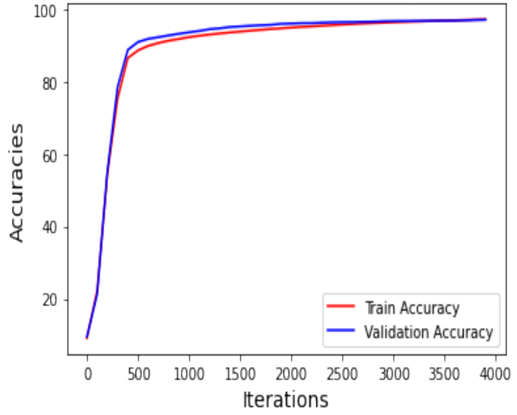


Training Accuracy/ Validation Accuracy vs Iterations for 1 hidden layer network with Learning rate =1.0

Network 2

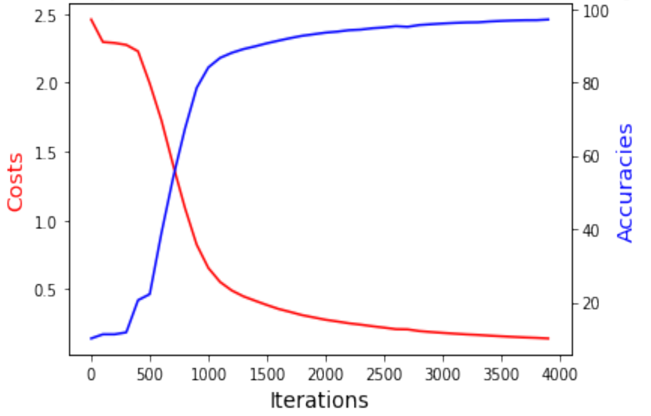


Training Cost/Accuracy over iterations for 2 hidden layer network with Learning rate =1.0

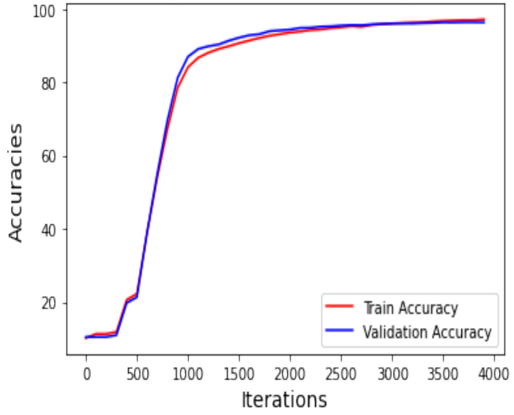


Training Accuracy/ Validation Accuracy vs Iterations for 2 hidden layer network with Learning rate =1.0

Network 3



Training Cost/Accuracy over iterations for 3 hidden layer network with Learning rate =1.0



Training Accuracy/ Validation Accuracy vs Iterations for 3 hidden layer network with Learning rate =1.0

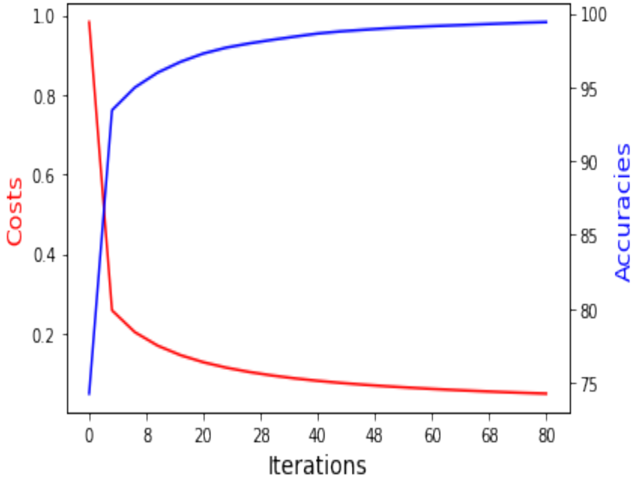
Mini-Batch Gradient Descent on MNIST Dataset Details and Results with Sigmoid

Network Details	**Network 1**	**Network 2**	**Network 3**
Hidden Layers / Neurons	1 / 64	2 / 64, 32	3 / 64, 32, 32
Network Initialization	Xavier	Xavier	Xavier
Learning rate	1.0	5.0	1.0
Epochs	100	500	100
Batch Size	128	128	128
Training data size	54,000	54,000	54,000
Validation data size	6,000	6,000	6,000
Test data size	10,000	10,000	10,000
Hidden layers activation	Sigmoid	Sigmoid	Sigmoid
Output layer activation	Softmax	Softmax	Softmax
Loss	Categorical Cross entropy	Categorical Cross entropy	Categorical Cross entropy
Training Accuracy (%)	99.41	99.03	99.67
Validation Accuracy (%)	97.77	97.13	97.46
Test Accuracy (%)	97.64	96.48	97.03
Test Error Rate (%)	**2.360**	3.520	2.970

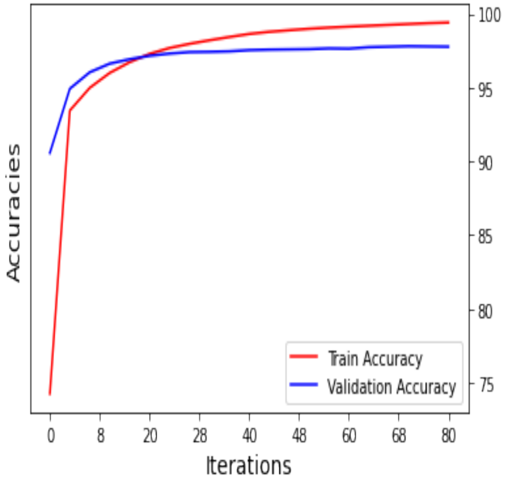
Sigmoid activated Network Convergence Graph

Network 1

Training Cost/Accuracy over iterations for 1 hidden layer network with Learning rate =1.0

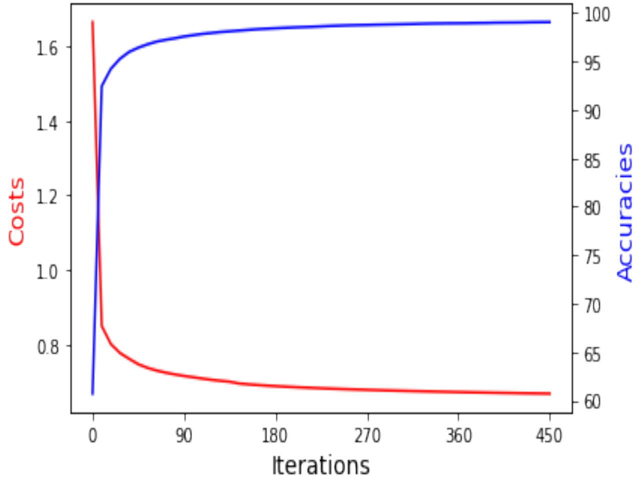


Training Accuracy/ Validation Accuracy vs Iterations for 1 hidden layer network with Learning rate =1.0

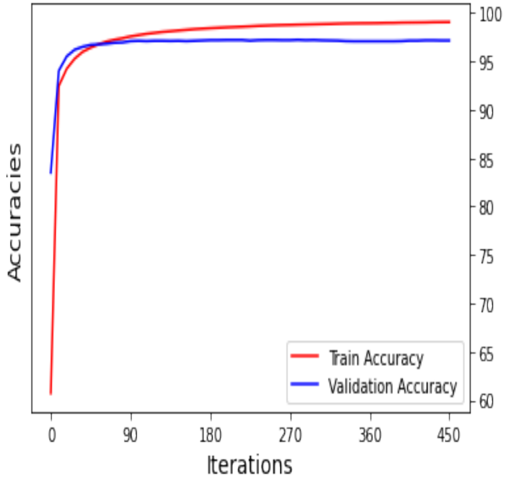


Network 2

Training Cost/Accuracy over iterations for 2 hidden layer network with Learning rate =5.0

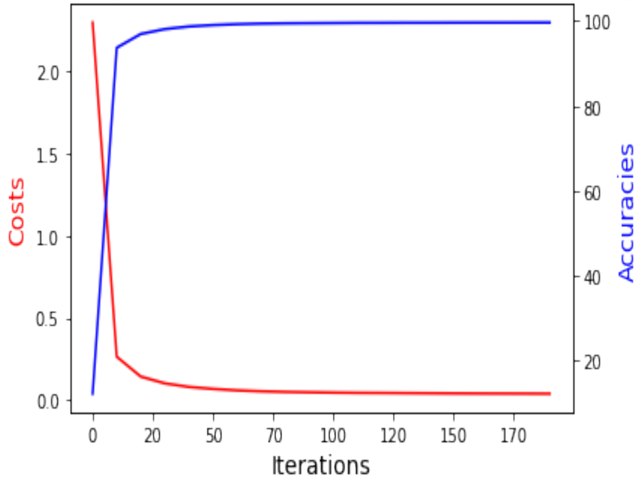


Training Accuracy/ Validation Accuracy vs Iterations for 2 hidden layer network with Learning rate =5.0

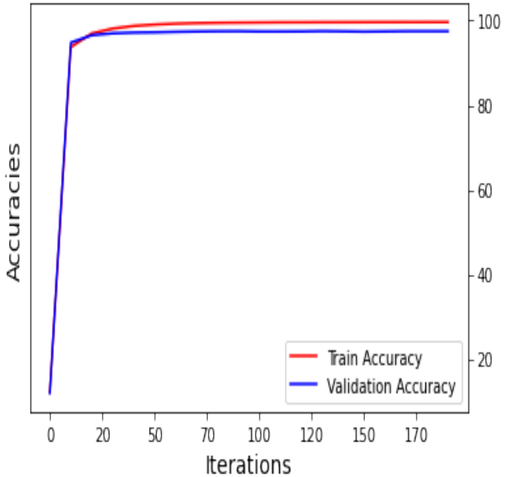


Network 3

Training Cost/Accuracy over iterations for 3 hidden layer network with Learning rate =1.0



Training Accuracy/ Validation Accuracy vs Iterations for 3 hidden layer network with Learning rate =1.0



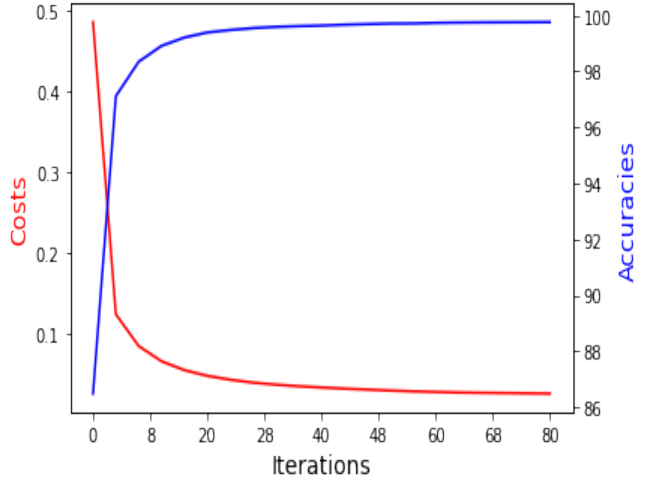
Mini-Batch Gradient Descent on MNIST Dataset Details and Results with ReLU

Network Details	**Network 1**	**Network 2**	**Network 3**
Hidden Layers / Neurons	1 / 64	2 / 64, 32	3 / 64, 32, 32
Network Initialization	Xavier	Xavier	Xavier
Learning rate	1.0	1.0	1.0
Epochs	100	100	100
Batch Size	128	128	128
Training data size	54,000	54,000	54,000
Validation data size	6,000	6,000	6,000
Test data size	10,000	10,000	10,000
Hidden layers activation	ReLU	ReLU	ReLU
Output layer activation	Softmax	Softmax	Softmax
Loss	Categorical Cross entropy	Categorical Cross entropy	Categorical Cross entropy
Training Accuracy (%)	99.76	99.03	99.69
Validation Accuracy (%)	98.02	97.58	97.83
Test Accuracy (%)	97.60	97.74	97.40
Test Error Rate (%)	2.400	**2.260**	2.600

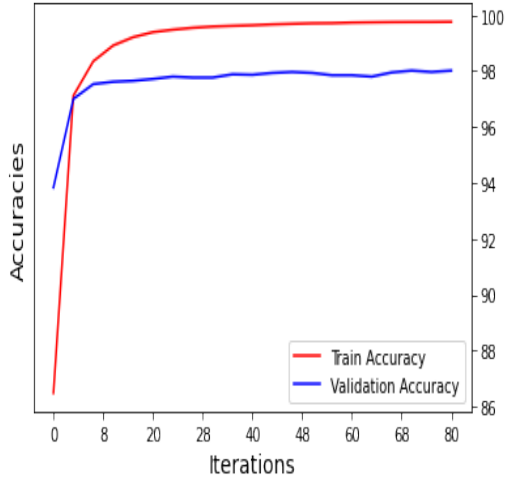
ReLU activated Network Convergence Graph

Network 1

Training Cost/Accuracy over iterations for 1 hidden layer network with Learning rate =1.0

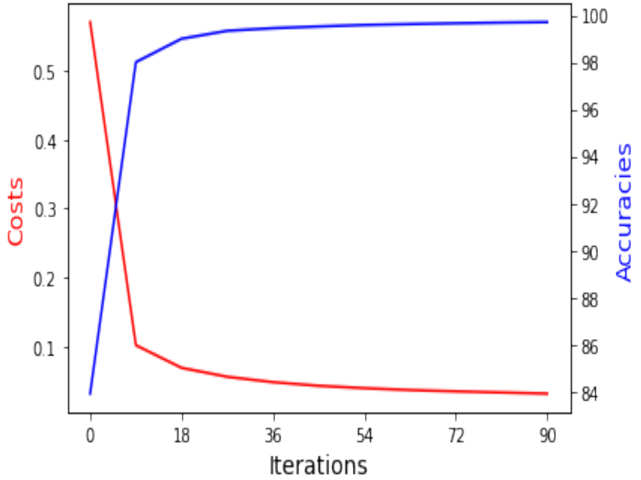


Training Accuracy/ Validation Accuracy vs Iterations for 1 hidden layer network with Learning rate =1.0

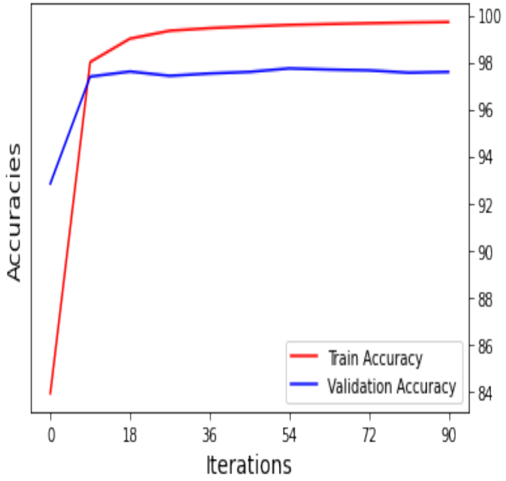


Network 2

Training Cost/Accuracy over iterations for 2 hidden layer network with Learning rate =1.0

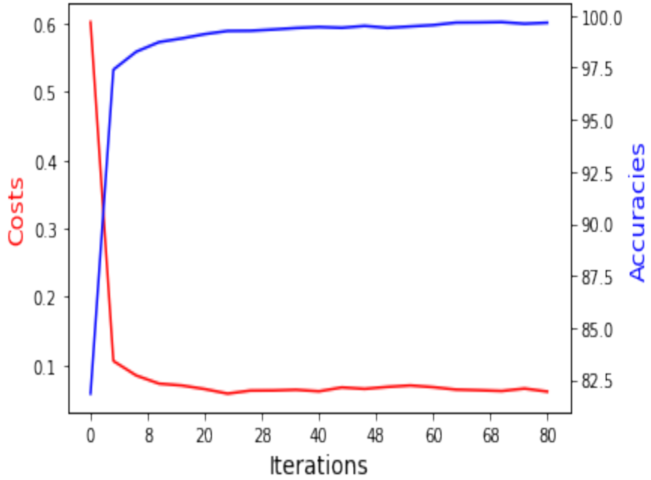


Training Accuracy/ Validation Accuracy vs Iterations for 2 hidden layer network with Learning rate =1.0

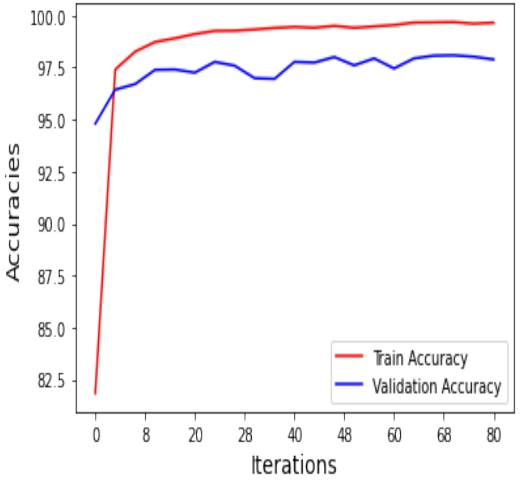


Network 3

Training Cost/Accuracy over iterations for 3 hidden layer network with Learning rate =1.0



Training Accuracy/ Validation Accuracy vs Iterations for 3 hidden layer network with Learning rate =1.0



Overall Comparison between best performing networks

Network Details	**Batch gradient descent based network**	**Sigmoid Activated Mini-batch network**	**ReLU activated Mini-batch network**
Hidden Layers / Neurons	2 / 64, 32	1 / 64	2 / 64, 32
Network Initialization	Xavier	Xavier	Xavier
Learning rate	1.0	1.0	1.0
Epochs	4000	100	100
Batch Size	54,000	128	128
Hidden layers activation	Sigmoid	Sigmoid	ReLU
Output layer activation	Softmax	Softmax	Softmax
Loss	Categorical Cross entropy	Categorical Cross entropy	Categorical Cross entropy
Training Accuracy (%)	97.24	99.41	99.03
Validation Accuracy (%)	97.10	97.77	97.58
Test Accuracy (%)	96.26	97.64	97.74
Test Error Rate (%)	3.740	2.360	**2.260**

Discussion

Regarding convergence rate :

- Batch Gradient Descent reaches towards convergence after 3000 epochs only and the reason is quiet obvious because the network needs to update its weights by looking at overall training data.
- Mini-batch Gradient Decent reaches convergence in less than 20 epochs, and the reason is because the weight updates happens for each mini batches, and it allows the network to update its weights by looking at the mini batches only at a time, resulting in overall convergence of the network in short amount of time.

Regarding Training and Validation Accuracy

- In my case, Training accuracy of the batch gradient descent network is less compared to mini-batch ones, which is due to the fact that the network was trained only for 4000 epochs in case of batch gradient descent which is not sufficient enough as can be seen from the diagram, that the network loss or accuracy has not plateaued completely.
- Validation and test accuracy of batch gradient descent based network is similar to training accuracy, as it can be generalized from the fact that, in batch network the whole training data is considered into account for weight updates, thus the network is more generalized during weight updates. It shows batch gradient descent network are more generalized, or does not over fit easily.
- Mini-batch network training accuracy is higher than validation accuracy since the weight updates has to be done for batches only which does not generalizes more. However, the convergence in minibatch is faster, because the weight updates that happens with the help of mini-batches resulting in faster updates of weights, and convergence of the network. It shows that the mini-batch networks overfits faster and thus requires some regularizations in order to avoid overfitting.

Discussion

Regarding Sigmoid vs ReLU

- Sigmoid vs ReLU based network, the convergence on ReLU is very fast which are less than 20 epochs for all the 1 to 3 hidden layer based network, and also the training time of the ReLU based network is also very small because of little less network complexity. Where as Sigmoid activated network minimum convergence is after 40 epochs only. In addition, the training time is also little bit more when the sigmoid was used.
- ReLU activated 2 hidden layer is giving the best performance among all the networks. Sigmoid activated single hidden layer is performing best compared to 2 and 3 hidden layers based network, however the ReLU activated all 1, 2, and 3 hidden layers based network are giving very high performance i.e. test error rate of 2.4%, 2.26% and 2.6% respectively. Sigmoid activated 1,2, and 3 hidden layers based network test error rate are 2.36%, 3.52%, and 2.97% respectively.

Regarding why 3 hidden layer network did not give the best result?

- 3 hidden layer network should have performed the best however due to the training data size which was not sufficient to train the large number of parameters of the network thus resulted in overfitting, and also the network suffered from vanishing gradient problem.

Regarding Overall test-error rate vs State of the art results

- My optimal model is ReLU activated, 2 hidden layer based network with test error rate of 2.260%, however the state of the art results are less than 1%. The reason for state of the art results to be higher than mine is because those models are using Convolutional Neural Networks for feature extraction from the images, however, I am only using MLP at pixel level.

جامعة خليفة
Khalifa University

Thank You

ku.ac.ae