Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

CSE4238: Soft Computing Lab

Assignment-2

Implementation of Deep Neural Network

Submitted by

Nazmus Shakib Shadin	17.01.04.119
Section	C
Lab Group	C1

September 5, 2021

Contents

1 Introduction

Bangla (Bengali) ranks fifth in Asia and it is also in the top ten spoken languages in the world. So, a huge number of people depend on this language for their day to day communication. Therefore, automatic recognition of Bengali handwritten characters and numeral digits are needed to be digitized for making the communication smoother. Many research works and models have been proposed to recognize Bengali handwritten characters and numeral digits so far, but still, a huge scopes there to improve this task in terms of accuracy and applicability.

Images of handwritten digits are different from natural images as the orientation of a digit, as well as similarity of features of different digits, makes confusion. Before we have this used this experiment using logistic regression. But in here, we will use deep neural network. Using this model, it will increase the accuracy. It provides a better performance.

In this assignment we will be exploring different bengali numerical dataset (Dataset C: As odd ID) and use to classify these digits. We will be using the following dataset (Dataset C) in this assignment.

We will tune the dataset(Dataset C) and compare it with another dataset (Dataset 2) on the basis of model loss and model accuracy.

2 Experiment-1

2.1 Hyperparameter

According to the question we set the hyperparameters as following:

- **Num of Hidden Layer:** 6
- **Number of nodes in hidden layers:** 200
- **Iterations:** 20000
- **Learning_rate:** 0.01
- **Batch_Size:** 20
- **Input_dim:** 28*28
- **Output_dim:** 10*10
- **Activation Function:** All ReLU
- **Optimizer:** Adamax
- **Loss:** CrossEntropy

2.2 Train Test Split Data

We have kept 80% of our data as training data and 20% of our data as testing data. In the given dataset (Dataset C) there are 24298 data. That means we have taken almost 19439 data as training data and 4860. From the dataset, we have only kept the column FileName and Digit.

2.3 Epoch Calculation

$$Epoch = \frac{Iterations}{\frac{total\_data}{batch\_size}} \tag{1}$$

Here, Iterations = 20000, total_data = 24298, batch_size = 20

so, Epoch = 16

2.4 Result Analysis

```
Epoch: 1
Iteration: 500. Loss: 2.299643039703369. Accuracy: 10.290183165260341
Epoch: 2
Iteration: 1000. Loss: 2.300184488296509. Accuracy: 9.63161144268368
Iteration: 1500. Loss: 2.318502187728882. Accuracy: 10.290183165260341
Epoch: 3
Iteration: 2000. Loss: 2.299276113510132. Accuracy: 10.290183165260341
Iteration: 2500. Loss: 2.306591510772705. Accuracy: 9.569870343692118
Epoch: 4
Iteration: 3000. Loss: 2.31643009185791. Accuracy: 10.104959868285656
Iteration: 3500. Loss: 2.3062214851379395. Accuracy: 9.919736571310969
Epoch: 5
Iteration: 4000. Loss: 2.299635410308838. Accuracy: 9.919736571310969
Iteration: 4500. Loss: 2.3084466457366943. Accuracy: 9.919736571310969
Epoch: 6
Iteration: 5000. Loss: 2.2909648418426514. Accuracy: 10.084379501955135
Iteration: 5500. Loss: 2.3201541900634766. Accuracy: 9.919736571310969
Epoch: 7
Iteration: 6000. Loss: 2.303091526031494. Accuracy: 10.084379501955135
Iteration: 6500. Loss: 2.30950927734375. Accuracy: 9.960897303972011
Epoch: 8
Iteration: 7000. Loss: 2.3026537895202637. Accuracy: 10.084379501955135
Iteration: 7500. Loss: 2.3019516468048096. Accuracy: 9.919736571310969
Epoch: 9
Iteration: 8000. Loss: 2.294377326965332. Accuracy: 9.63161144268368
Iteration: 8500. Loss: 2.318166732788086. Accuracy: 9.569870343692118
Epoch: 10
Iteration: 9000. Loss: 2.3051791191101074. Accuracy: 9.569870343692118
Iteration: 9500. Loss: 2.3011717796325684. Accuracy: 9.63161144268368
Epoch: 11
Iteration: 10000. Loss: 2.3072052001953125. Accuracy: 9.63161144268368
Iteration: 10500. Loss: 2.30485200881958. Accuracy: 9.63161144268368
Epoch: 12
Iteration: 11000. Loss: 2.3085746765136672. Accuracy: 10.187281333607737
Iteration: 11500. Loss: 2.307220220565796. Accuracy: 9.63161144268368
Epoch: 13
Iteration: 12000. Loss: 2.3001580238342285. Accuracy: 9.919736571310969
Iteration: 12500. Loss: 2.312523603439331. Accuracy: 10.290183165260341
Epoch: 14
Iteration: 13000. Loss: 2.309614896774292. Accuracy: 10.104959868285656
Iteration: 13500. Loss: 2.2922136783599854. Accuracy: 9.63161144268368
Epoch: 15
Iteration: 14000. Loss: 2.3146326541900635. Accuracy: 9.63161144268368
Iteration: 14500. Loss: 2.295405864715576. Accuracy: 9.569870343692118
Epoch: 16
Iteration: 15000. Loss: 2.2966742515563965. Accuracy: 9.63161144268368
Iteration: 15500. Loss: 2.2944118976593018. Accuracy: 9.919736571310969
```

Here, we can observe that the model according to the achieved the highest accuracy of 10.29% at iteration 1 and 3.

2.5 Loss Vs. Iteration Graph

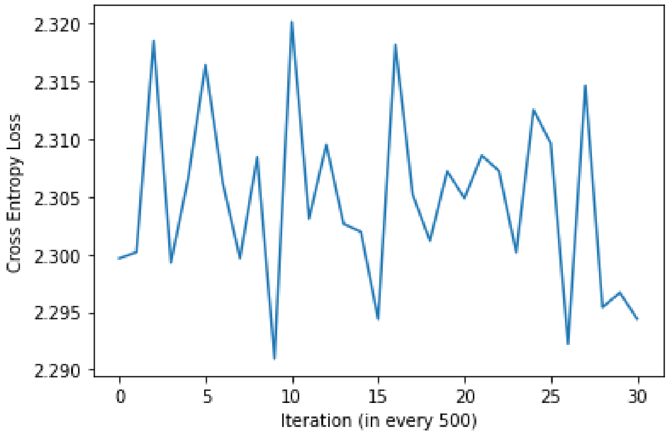

Figure 1: Loss vs. Iteration Graph for Experiment1

3 Experiment-2

3.1 Hyperparameter

According to the question we set the hyperparameters as following:

- **Num of Hidden Layer:** 6
- **Number of nodes in hidden layers:** 300
- **Iterations:** 30000
- **Learning_rate:** 0.01
- **Input_dim:** 28*28
- **Output_dim:** 10*10
- **Batch_Size:** 350
- **Activation Function:** All ReLU except the 3rd layer where we used Softmax
- **Optimizer:** Adamax
- **Loss:** CrossEntropy

3.2 Train Test Split Data

We have kept 80% of our data as training data and 20% of our data as testing data. In the given dataset (Dataset C) there are 24298 data. That means we have taken almost 19439 data as training data and 4860. From the dataset, we have only kept the column FileName and Digit.

3.3 Epoch Calculation

$$Epoch = \frac{Iterations}{\frac{total\_data}{batch\_size}} \tag{2}$$

Here, Iterations = 30000, total_data = 24298, batch_size = 350
so, Epoch = 432

3.4 Result Analysis

```
Iteration: 500. Loss: 1.7704663276672363. Accuracy: 38.56760650339576
Iteration: 1000. Loss: 0.9347795844078064. Accuracy: 64.972216505453
Iteration: 1500. Loss: 0.5606016516685486. Accuracy: 75.30356040337519
Iteration: 2000. Loss: 0.3942268192768097. Accuracy: 79.19324963984359
Iteration: 2500. Loss: 0.24090519547462463. Accuracy: 80.07820539205598
Iteration: 3000. Loss: 0.18159811198711395. Accuracy: 81.82753653015024
Iteration: 3500. Loss: 0.22715260088443756. Accuracy: 81.97159909446388
Iteration: 4000. Loss: 0.10203585028648376. Accuracy: 82.01275982712492
Iteration: 4500. Loss: 0.08291501551866531. Accuracy: 81.66289359950608
Iteration: 5000. Loss: 0.0763048380613327. Accuracy: 81.82753653015024
Iteration: 5500. Loss: 0.046442169696092606. Accuracy: 81.8892776291418
Iteration: 6000. Loss: 0.024692300707101822. Accuracy: 82.03334019345544
Iteration: 6500. Loss: 0.03883235529065132. Accuracy: 82.4861082527269
Iteration: 7000. Loss: 0.15453088283538818. Accuracy: 82.50668861905741
Iteration: 7500. Loss: 0.032452505081892014. Accuracy: 83.12409960897304
Iteration: 8000. Loss: 0.08138242363929749. Accuracy: 82.36262605474377
Iteration: 8500. Loss: 0.030961502343416214. Accuracy: 83.10351924264252
Iteration: 9000. Loss: 0.023177068680524826. Accuracy: 82.93887631199836
Iteration: 9500. Loss: 0.03790069743990898. Accuracy: 82.81539411401523
Iteration: 10000. Loss: 0.07143961638212204. Accuracy: 82.85655484667627
Iteration: 10500. Loss: 0.026217572391033173. Accuracy: 83.1858407079646
Iteration: 11000. Loss: 0.013134746812283993. Accuracy: 82.85655484667627
Iteration: 11500. Loss: 0.02663564495742321. Accuracy: 83.26816217328668
Iteration: 12000. Loss: 0.012636248953640461. Accuracy: 82.8771352130068
Iteration: 12500. Loss: 0.016069287434220314. Accuracy: 82.73307264869315
Iteration: 13000. Loss: 0.06165703386068344. Accuracy: 83.30932290594772
Iteration: 13500. Loss: 0.023508764803409576. Accuracy: 82.7742333813542
Iteration: 14000. Loss: 0.04574590176343918. Accuracy: 83.43280510393085
Iteration: 14500. Loss: 0.04782503843307495. Accuracy: 82.60959045071002
Iteration: 15000. Loss: 0.01713939569890499. Accuracy: 82.28030458942169
Iteration: 15500. Loss: 0.019093463197350502. Accuracy: 83.61802840090553
Iteration: 16000. Loss: 0.03344385325908661. Accuracy: 83.65918913356657
Iteration: 16500. Loss: 0.00880385935306549. Accuracy: 83.63860876723605
Iteration: 17000. Loss: 0.014661023393273354. Accuracy: 83.20642107429512
Iteration: 17500. Loss: 0.00748130539432168. Accuracy: 82.93887631199836
Iteration: 18000. Loss: 0.002354151802137494. Accuracy: 83.63860876723605
Iteration: 18500. Loss: 0.023125814273953438. Accuracy: 82.81539411401523
Iteration: 19000. Loss: 0.01748194545507431. Accuracy: 83.3916443712698
Iteration: 19500. Loss: 0.0018441157881170511. Accuracy: 83.26816217328668
Iteration: 20000. Loss: 0.009359064511954784. Accuracy: 83.1858407079646
Iteration: 20500. Loss: 0.011075960472226143. Accuracy: 82.8771352130068
Iteration: 21000. Loss: 0.06137864664196968. Accuracy: 82.52726898538793
Iteration: 21500. Loss: 0.013514259830117226. Accuracy: 82.69191191603211
Iteration: 22000. Loss: 0.010233347304165363. Accuracy: 83.2887425396172
Iteration: 22500. Loss: 0.0021437855903059244. Accuracy: 83.30932290594772
Iteration: 23000. Loss: 0.0068382821045815945. Accuracy: 84.29718048981272
Iteration: 23500. Loss: 0.013714092783629894. Accuracy: 83.67976949989709
Iteration: 24000. Loss: 0.022804323583841324. Accuracy: 82.91829594566784
```

Here, we can observe that our model achieved the highest accuracy of 84.30% at Iteration 23000 after multiple trial and error approach.

3.5 Loss Vs. Iteration Graph

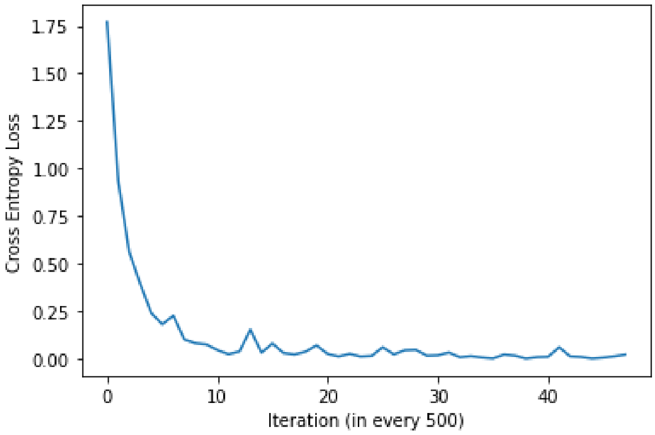

Figure 2: Loss vs. Iteration Graph for Experiment2

4 Comparison between two models for the first Dataset

Table 1: Hyparparameter Comparison between the Deep Neural Network on Dataset C

	layer	**Nodes**	**Iteration**	**Epoch**	**Lr**	**BS**	**Activation**	**Opt.**	**Loss**
Exp_1	6	200	20000	16	0.01	20	Relu	Adamax	Cross_Entropy
Exp_2	6	300	30000	432	0.01	350	Relu and Softmax	Adamax	Cross_Entropy

Here,
layer = Hidden_Layer
Lr = Learning_rate
BS = Batch_Size
Opt. = Optimizer.

The following Table-1 is showing the hyperparameter for the Dataset C for the both experiment (Experiment-1 and Experiment-2).

After applying the parameter for the experiment-1 we have achieved the max accuracy of 10.29% and the experiment-2 have achieved the max accuracy of 84.30% after being tuned from the first experiment.

5 Comparison Between the first Dataset (Dataset C) and the second Dataset(Dataset_2)

5.1 Dataset_2 Replacing the dataset of Experiment-1

In the following we will see loss and accuracy of the dataset-2 when it is replaced with the dataset of experiment-1.

```
Iteration: 500. Loss: 0.6053855419158936. Accuracy: 77.59
Iteration: 1000. Loss: 0.4763883948326111. Accuracy: 81.72
Iteration: 1500. Loss: 0.7361134886741638. Accuracy: 80.48
Iteration: 2000. Loss: 0.2924569547176361. Accuracy: 81.78
Iteration: 2500. Loss: 0.49476972222328186. Accuracy: 82.6
Iteration: 3000. Loss: 0.3280336558818817. Accuracy: 84.61
Iteration: 3500. Loss: 0.4236431121826172. Accuracy: 85.29
Iteration: 4000. Loss: 0.373003751039505. Accuracy: 84.12
Iteration: 4500. Loss: 0.6969588398933411. Accuracy: 86.18
Iteration: 5000. Loss: 0.6225858926773071. Accuracy: 85.22
Iteration: 5500. Loss: 0.5186593532562256. Accuracy: 85.52
```

```
Iteration: 6000. Loss: 0.3275323510169983. Accuracy: 86.3
Iteration: 6500. Loss: 0.6941832304000854. Accuracy: 86.39
Iteration: 7000. Loss: 0.28721290826797485. Accuracy: 87.15
Iteration: 7500. Loss: 0.17897240817546844. Accuracy: 86.26
Iteration: 8000. Loss: 0.3745700418949127. Accuracy: 85.04
Iteration: 8500. Loss: 0.37330248951911926. Accuracy: 87.26
Iteration: 9000. Loss: 0.38829493522644043. Accuracy: 87.35
Iteration: 9500. Loss: 0.3316674530506134. Accuracy: 86.78
Iteration: 10000. Loss: 0.20874185860157013. Accuracy: 85.76
Iteration: 10500. Loss: 0.5832291841506958. Accuracy: 87.9
Iteration: 11000. Loss: 0.30748093128204346. Accuracy: 87.6
Iteration: 11500. Loss: 0.09647516161203384. Accuracy: 86.82
Iteration: 12000. Loss: 0.5812944173812866. Accuracy: 86.55
Iteration: 12500. Loss: 0.4371141493320465. Accuracy: 87.48
Iteration: 13000. Loss: 0.1400289088487625. Accuracy: 87.83
Iteration: 13500. Loss: 0.10022695362567902. Accuracy: 87.3
Iteration: 14000. Loss: 0.3129619061946869. Accuracy: 87.66
Iteration: 14500. Loss: 0.10388094186782837. Accuracy: 86.93
Iteration: 15000. Loss: 0.44625115394592285. Accuracy: 87.64
Iteration: 15500. Loss: 0.18216651678085327. Accuracy: 87.59
Iteration: 16000. Loss: 0.3367074131965637. Accuracy: 87.6
Iteration: 16500. Loss: 0.51124107837677. Accuracy: 87.53
Iteration: 17000. Loss: 0.17913374304771423. Accuracy: 87.73
Iteration: 17500. Loss: 0.2732146680355072. Accuracy: 87.94
Iteration: 18000. Loss: 0.35739976167678833. Accuracy: 88.46
```

Here, we can see all the loss and accuracy of experiment-3 where we can observe that, it performs far more better than the experiment-1 on the same model because we in experiment-3 on the model of experience we have used the numerical values without images and nodes, hidden layers and other hyperparameters are compatible for this dataset (Dataset_2). Now let's see the model loss graph for experiment-1 and experiment-3.

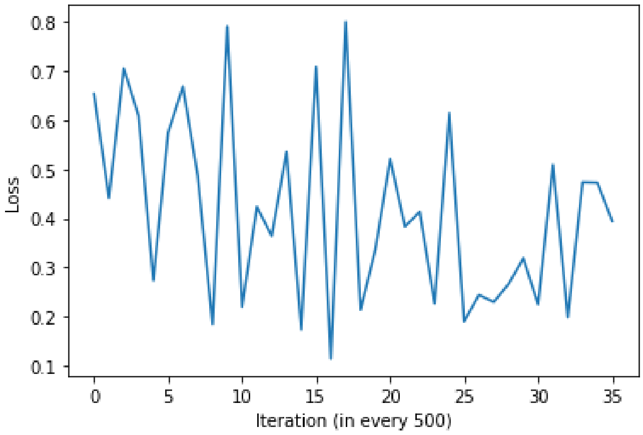

Figure 3: Loss vs. Iteration Graph for Experiment3 on model of experiment-1

From figure 3, and figure 4 we can observe that, the model loss for experiment-3 which is shown in figure-3 is better than the model loss for experiment-1 is shown in figure-4.

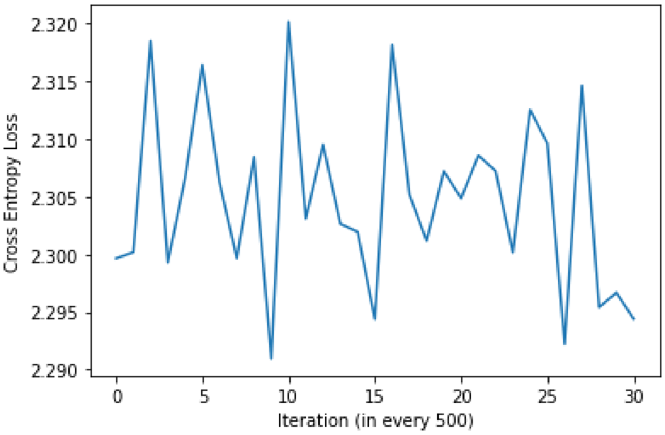

Figure 4: Loss vs. Iteration Graph for Experiment1

5.2 Dataset_2 Replacing the dataset of Experiment-2

In the following we will see loss and accuracy of the dataset-2 when it is replaced with the dataset of experiment-2.

```
Iteration: 500. Loss: 2.300928831100464. Accuracy: 10.24
Iteration: 1000. Loss: 2.303175687789917. Accuracy: 11.28
Iteration: 1500. Loss: 2.301696538925171. Accuracy: 12.32
Iteration: 2000. Loss: 2.301663875579834. Accuracy: 11.9
Iteration: 2500. Loss: 2.3015947341918945. Accuracy: 16.7
Iteration: 3000. Loss: 2.3014702796936035. Accuracy: 20.34
Iteration: 3500. Loss: 2.301339864730835. Accuracy: 23.43
Iteration: 4000. Loss: 2.301071882247925. Accuracy: 21.86
Iteration: 4500. Loss: 2.300731658935547. Accuracy: 24.45
Iteration: 5000. Loss: 2.300327777862549. Accuracy: 31.47
Iteration: 5500. Loss: 2.3003180027008057. Accuracy: 37.49
Iteration: 6000. Loss: 2.299665927886963. Accuracy: 32.54
Iteration: 6500. Loss: 2.2996432781219482. Accuracy: 34.39
Iteration: 7000. Loss: 2.299240827560425. Accuracy: 36.11
Iteration: 7500. Loss: 2.298677921295166. Accuracy: 41.36
Iteration: 8000. Loss: 2.297673225402832. Accuracy: 36.96
Iteration: 8500. Loss: 2.297006130218506. Accuracy: 39.46
Iteration: 9000. Loss: 2.296048641204834. Accuracy: 37.06
Iteration: 9500. Loss: 2.2944788932800293. Accuracy: 36.42
Iteration: 10000. Loss: 2.293665885925293. Accuracy: 43.32
Iteration: 10500. Loss: 2.2899725437164307. Accuracy: 32.61
Iteration: 11000. Loss: 2.2855467796325684. Accuracy: 33.3
Iteration: 11500. Loss: 2.281125068664551. Accuracy: 30.04
Iteration: 12000. Loss: 2.2671959400177. Accuracy: 28.07
Iteration: 12500. Loss: 2.2447078227996826. Accuracy: 25.57
Iteration: 13000. Loss: 2.20707631111145. Accuracy: 24.38
Iteration: 13500. Loss: 2.1219210624694824. Accuracy: 22.34
Iteration: 14000. Loss: 2.069288492202759. Accuracy: 25.21
Iteration: 14500. Loss: 2.058783531188965. Accuracy: 29.84
Iteration: 15000. Loss: 2.007864475250244. Accuracy: 36.21
Iteration: 15500. Loss: 2.012235164642334. Accuracy: 28.97
Iteration: 16000. Loss: 2.018775463104248. Accuracy: 28.03
Iteration: 16500. Loss: 2.020832061767578. Accuracy: 28.47
Iteration: 17000. Loss: 1.9872915744781494. Accuracy: 28.08
Iteration: 17500. Loss: 1.87290358543396. Accuracy: 30.4
Iteration: 18000. Loss: 1.8450592756271362. Accuracy: 35.38
Iteration: 18500. Loss: 1.747239112854004. Accuracy: 40.27
```

```
Iteration: 19000. Loss: 1.6455456018447876. Accuracy: 39.61
Iteration: 19500. Loss: 1.549673080444336. Accuracy: 45.02
Iteration: 20000. Loss: 1.4432774782180786. Accuracy: 48.82
Iteration: 20500. Loss: 1.3979227542877197. Accuracy: 50.61
Iteration: 21000. Loss: 1.2971867322921753. Accuracy: 49.58
Iteration: 21500. Loss: 1.6457173824310303. Accuracy: 41.91
Iteration: 22000. Loss: 1.1609513759613037. Accuracy: 53.43
Iteration: 22500. Loss: 1.0942823886871338. Accuracy: 55.21
Iteration: 23000. Loss: 1.0380196571350098. Accuracy: 62.01
Iteration: 23500. Loss: 0.9918182492256165. Accuracy: 62.12
Iteration: 24000. Loss: 0.9865191578865051. Accuracy: 59.61
Iteration: 24500. Loss: 1.022180438041687. Accuracy: 53.9
Iteration: 25000. Loss: 1.0446475744247437. Accuracy: 55.27
Iteration: 25500. Loss: 0.9382985830307007. Accuracy: 65.68
Iteration: 26000. Loss: 0.8100550770759583. Accuracy: 65.63
Iteration: 26500. Loss: 0.7715207934379578. Accuracy: 72.23
Iteration: 27000. Loss: 0.8018977642059326. Accuracy: 72.18
Iteration: 27500. Loss: 0.7401929497718811. Accuracy: 69.57
Iteration: 28000. Loss: 0.7200825810432434. Accuracy: 65.54
Iteration: 28500. Loss: 0.6162732243537903. Accuracy: 76.02
Iteration: 29000. Loss: 0.6694281697273254. Accuracy: 77.01
Iteration: 29500. Loss: 0.7687380313873291. Accuracy: 67.63
Iteration: 30000. Loss: 0.5619120597839355. Accuracy: 80.62
```

we have seen the loss and accuracy for experiment-3 above. Now we will compare them.
Here, we can see all the loss and accuracy of experiment-3 where we can observe that, it performs far more better than the experiment-1 on the same model because this kind of hyper parameters are compatible for images.
Now let's see the model loss graph for experiment-2 and experiment-3.

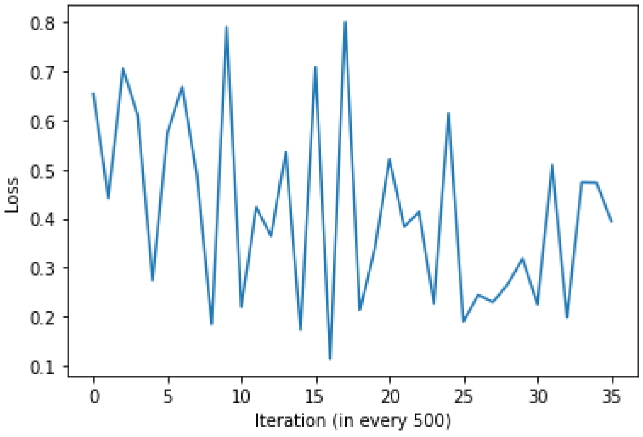

Figure 5: Loss vs. Iteration Graph for Experiment3 on model of experiment-2

From figure 5, and figure 6 we can observe that, the model loss for experiment-3 which is shown in figure-5 is less better than the model loss for experiment-2 is shown in figure-6.

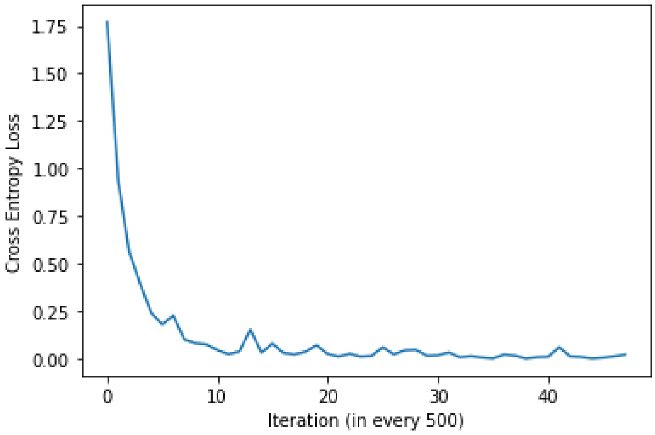

Figure 6: Loss vs. Iteration Graph for Experiment2

6 Github Repository

https://github.com/shadhin39/CSE4238_SoftComputingLab_Assignment2
The Above link is the github repository for the whole assignment.