

Ahsanullah University of Science and Technology  
Department of Computer Science and Engineering

Soft Computing  
CSE4238

Assignment-2 Report

---

Topic : Deep Neural Network (Linear Layer)

---

Submitted by

Sabbir Ahmed

17.01.04.011

September 5, 2021

# 1 Dataset Preparing

Download the [Dataset C](#). We have to prepare the dataset in following process:

```
class CustomDatasetmine(Dataset):
    def __init__(self, csv_file, root_dir, transform=None):
        self.annotations = pd.read_csv(csv_file)
        self.root_dir = root_dir
        self.transform = transform
    def __len__(self):
        return len(self.annotations)
    def __getitem__(self, index):
        img_path = os.path.join(self.root_dir, self.annotations.iloc[index,0])
        image = io.imread(img_path)
        y_label = torch.tensor(int(self.annotations.iloc[index, 3]))

        if self.transform:
            image=self.transform(image)
        return (image, y_label)
```

⇒ Read the CSV file for image paths and following labels

For Testing purpose 20% data has been splitted

↓

```
[ ] data = CustomDatasetmine("/content/training-c.csv", "/content/training-c", transforms.Compose([transforms.ToPILImage(), transforms.Resize(size=(32, 32)), transforms.ToTensor()]))
train_set, test_set = torch.utils.data.random_split(data, [19398, 4900])
```

## 1.1 Experiment 1:

These Parameters must be used. After performing according to the given parameters, accuracy is 9.51

```
# Hyperparameters
batch_size = 20
num_iters = 20000
input_dim = 32*32 # num_features = 784
num_hidden = 200
output_dim = 10

learning_rate = 0.01
```

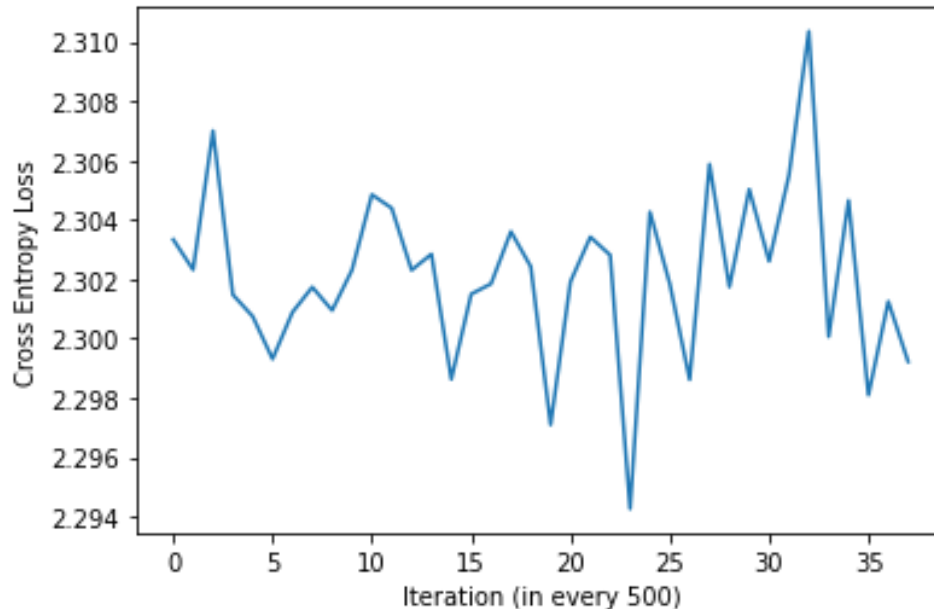
⇒ For fixed parameters

```
# Print Loss
print('Iteration: {}. Loss: {}. Accuracy: {}'.format(iter, loss, iter

Iteration: 500. Loss: 2.303332805633545. Accuracy: 9.428571428571429
Iteration: 1000. Loss: 2.3023149967193604. Accuracy: 9.551020408163266
Iteration: 1500. Loss: 2.3069963455200195. Accuracy: 9.73469387755102
Iteration: 2000. Loss: 2.301473617553711. Accuracy: 9.510204081632653
Iteration: 2500. Loss: 2.3007426261901855. Accuracy: 9.73469387755102
Iteration: 3000. Loss: 2.299307107925415. Accuracy: 9.510204081632653
Iteration: 3500. Loss: 2.3008809089660645. Accuracy: 9.877551020408163
Iteration: 4000. Loss: 2.3017284870147705. Accuracy: 10.10204081632653
Iteration: 4500. Loss: 2.3009445667266846. Accuracy: 9.510204081632653
Iteration: 5000. Loss: 2.3023054599761963. Accuracy: 9.510204081632653
Iteration: 5500. Loss: 2.3048458099365234. Accuracy: 9.510204081632653
Iteration: 6000. Loss: 2.3043887615203857. Accuracy: 9.73469387755102
Iteration: 6500. Loss: 2.3023006916046143. Accuracy: 10.89795918367347
Iteration: 7000. Loss: 2.3028416633605957. Accuracy: 9.877551020408163
Iteration: 7500. Loss: 2.2986245155334473. Accuracy: 9.510204081632653
Iteration: 8000. Loss: 2.3014936447143555. Accuracy: 9.510204081632653
Iteration: 8500. Loss: 2.3018410205841064. Accuracy: 9.510204081632653
Iteration: 9000. Loss: 2.303598403930664. Accuracy: 9.510204081632653
Iteration: 9500. Loss: 2.302419424057007. Accuracy: 9.73469387755102
Iteration: 10000. Loss: 2.297090768814087. Accuracy: 9.510204081632653
Iteration: 10500. Loss: 2.301913261413574. Accuracy: 9.510204081632653
Iteration: 11000. Loss: 2.303419589996338. Accuracy: 9.510204081632653
Iteration: 11500. Loss: 2.302805185317993. Accuracy: 9.510204081632653
Iteration: 12000. Loss: 2.2942585945129395. Accuracy: 9.877551020408163
Iteration: 12500. Loss: 2.3042595386505127. Accuracy: 9.73469387755102
Iteration: 13000. Loss: 2.301840305328369. Accuracy: 9.877551020408163
Iteration: 13500. Loss: 2.298611640930176. Accuracy: 9.73469387755102
Iteration: 14000. Loss: 2.3058714866638184. Accuracy: 9.510204081632653
Iteration: 14500. Loss: 2.3017358779907227. Accuracy: 9.73469387755102
Iteration: 15000. Loss: 2.305023670196533. Accuracy: 9.877551020408163
Iteration: 15500. Loss: 2.30260157558514404. Accuracy: 9.510204081632653
Iteration: 16000. Loss: 2.3054966926574707. Accuracy: 9.73469387755102
Iteration: 16500. Loss: 2.310349941253662. Accuracy: 9.510204081632653
Iteration: 17000. Loss: 2.300060510635376. Accuracy: 9.510204081632653
```

⇒ Accuracy 9.51

Graph :



## 1.2 Experiment 2:

For getting better result, parameters are changed as follows. The accuracy then increases to 18.75

```
# Hyperparameters
batch_size = 40
num_iters = 50000
input_dim = 32*32 # num_features = 784
num_hidden = 500
output_dim = 10

learning_rate = 0.01
```



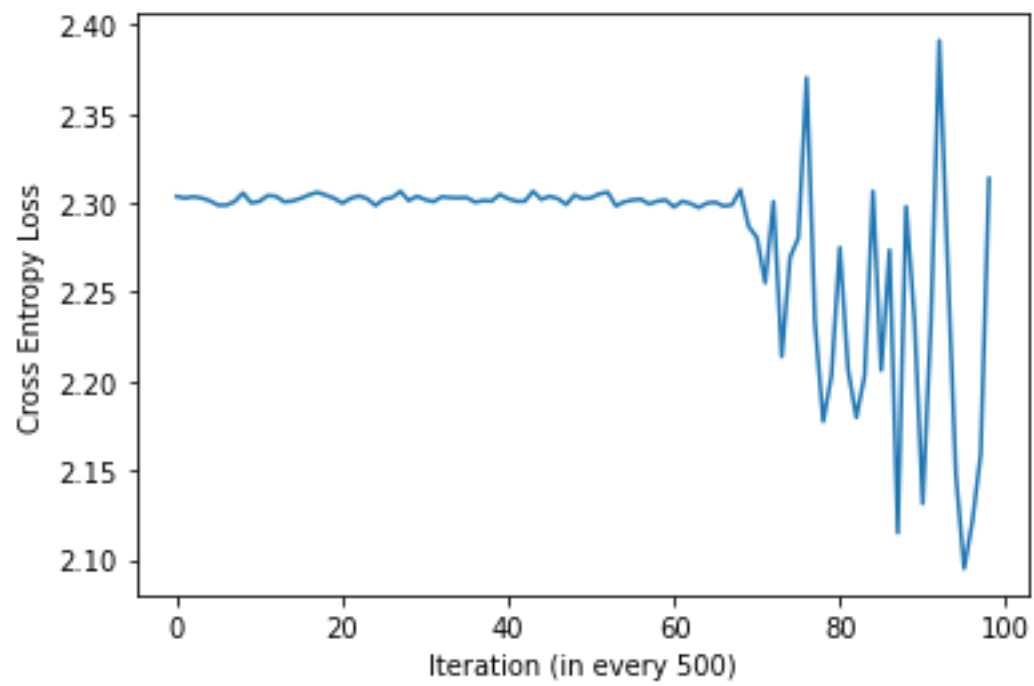
**Parameters**

```
Iteration: 40000. Loss: 2.2032856941223145. Accuracy: 18.244897959183675
Iteration: 40500. Loss: 2.275024175643921. Accuracy: 18.26530612244898
Iteration: 41000. Loss: 2.2065012454986572. Accuracy: 17.510204081632654
Iteration: 41500. Loss: 2.179683208465576. Accuracy: 16.877551020408163
Iteration: 42000. Loss: 2.2033376693725586. Accuracy: 19.081632653061224
Iteration: 42500. Loss: 2.306607723236084. Accuracy: 20.102040816326532
Iteration: 43000. Loss: 2.2061221599578857. Accuracy: 16.979591836734695
Iteration: 43500. Loss: 2.273656129837036. Accuracy: 16.163265306122447
Iteration: 44000. Loss: 2.1149027347564697. Accuracy: 16.979591836734695
Iteration: 44500. Loss: 2.2981419563293457. Accuracy: 14.714285714285714
Iteration: 45000. Loss: 2.235240936279297. Accuracy: 15.428571428571429
Iteration: 45500. Loss: 2.1313552856445312. Accuracy: 17.918367346938776
Iteration: 46000. Loss: 2.2336411476135254. Accuracy: 20.612244897959183
Iteration: 46500. Loss: 2.391308546066284. Accuracy: 20.632653061224488
Iteration: 47000. Loss: 2.2668001651763916. Accuracy: 20.285714285714285
Iteration: 47500. Loss: 2.147733688354492. Accuracy: 19.551020408163264
Iteration: 48000. Loss: 2.0949206352233887. Accuracy: 16.6734693877551
Iteration: 48500. Loss: 2.1216442584991455. Accuracy: 19.142857142857142
Iteration: 49000. Loss: 2.159135580062866. Accuracy: 19.081632653061224
Iteration: 49500. Loss: 2.3137574195861816. Accuracy: 18.775510204081634
```



**Accuracy**

Graph :



### 1.3 Experiment 3:

For getting even more better accuracy, parameters are changed as follows. The accuracy then increases to 70.1%

```
# Hyperparameters
batch_size = 70
num_iters = 80000
input_dim = 32*32 # num_features = 784
num_hidden = 800
output_dim = 10

learning_rate = 0.01
```



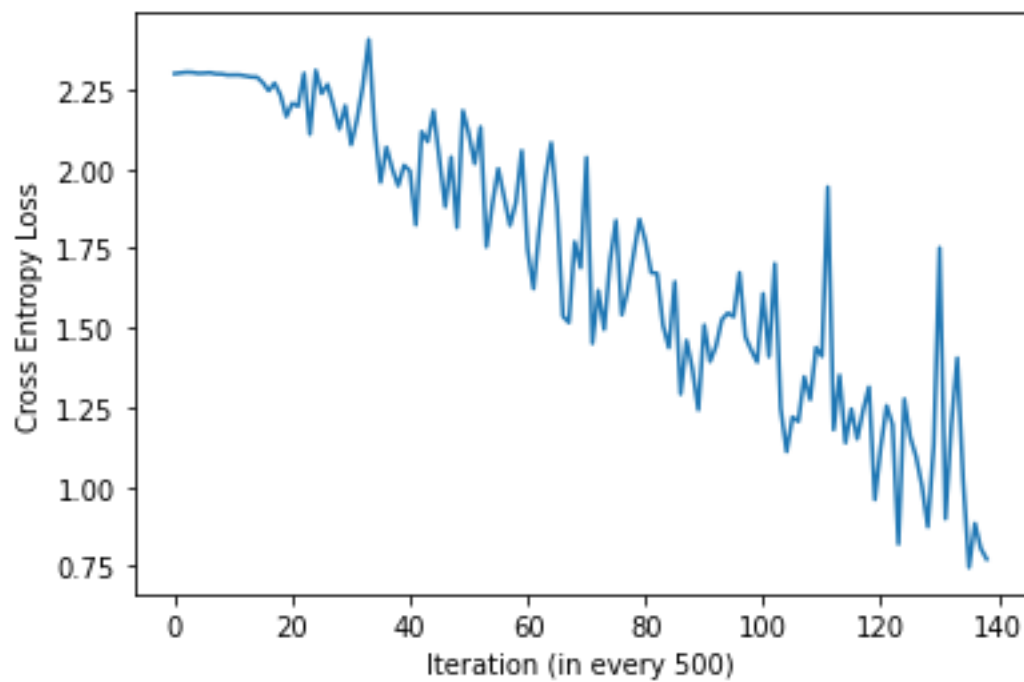
**Parameters**

```
Iteration: 64500. Loss: 1.674544095993042. Accuracy: 54.244897959183675
Iteration: 65000. Loss: 1.095276951789856. Accuracy: 53.12244897959184
Iteration: 65500. Loss: 1.0067182779312134. Accuracy: 58.83673469387755
Iteration: 66000. Loss: 1.3547345399856567. Accuracy: 53.95918367346939
Iteration: 66500. Loss: 1.0009421110153198. Accuracy: 58.12244897959184
Iteration: 67000. Loss: 0.8690697550773621. Accuracy: 64.18367346938776
Iteration: 67500. Loss: 0.97789067029953. Accuracy: 60.30612244897959
Iteration: 68000. Loss: 1.089109182357788. Accuracy: 65.75510204081633
Iteration: 68500. Loss: 0.8655075430870056. Accuracy: 66.61224489795919
Iteration: 69000. Loss: 1.0298471450805664. Accuracy: 65.28571428571429
Iteration: 69500. Loss: 1.0977097749710083. Accuracy: 41.42857142857143
Iteration: 70000. Loss: 0.7743734121322632. Accuracy: 63.775510204081634
Iteration: 70500. Loss: 1.0895328521728516. Accuracy: 66.3061224489796
Iteration: 71000. Loss: 1.243025302886963. Accuracy: 65.06122448979592
Iteration: 71500. Loss: 1.0204646587371826. Accuracy: 55.61224489795919
Iteration: 72000. Loss: 0.9451479315757751. Accuracy: 60.857142857142854
Iteration: 72500. Loss: 1.2282171249389648. Accuracy: 65.34693877551021
Iteration: 73000. Loss: 0.7281823754310608. Accuracy: 69.71428571428571
Iteration: 73500. Loss: 0.8693990707397461. Accuracy: 69.75510204081633
Iteration: 74000. Loss: 0.8023049831390381. Accuracy: 67.18367346938776
Iteration: 74500. Loss: 0.7983916401863098. Accuracy: 70.77551020408163
Iteration: 75000. Loss: 0.9436432123184204. Accuracy: 57.38775510204081
Iteration: 75500. Loss: 0.7644849419593811. Accuracy: 61.57142857142857
Iteration: 76000. Loss: 0.6863813996315002. Accuracy: 68.59183673469387
Iteration: 76500. Loss: 0.7616690993309021. Accuracy: 70.04081632653062
Iteration: 77000. Loss: 1.0928810834884644. Accuracy: 65.73469387755102
Iteration: 77500. Loss: 1.0226831436157227. Accuracy: 69.89795918367346
Iteration: 78000. Loss: 1.4621509313583374. Accuracy: 61.285714285714285
Iteration: 78500. Loss: 0.7927481532096863. Accuracy: 71.3061224489796
Iteration: 79000. Loss: 1.4409058094024658. Accuracy: 47.265306122448976
Iteration: 79500. Loss: 1.0531506538391113. Accuracy: 63.83673469387755
Iteration: 80000. Loss: 0.566940808296204. Accuracy: 70.63265306122449
```



**Final Accuracy 70%**

Graph :



### 3 Checking Models Performance by Another Dataset



#### 3.1 Dataset Preparing:

As like previous procedure dataset should be formatted after downloading [Dataset 2](#)

##### 3.1.1 Experiments with same parameters

For using the same parameters, but different dataset, accuracy is now as follows 84% , 91 %

#### 4 Comparison table:

Parameters	Dataset C	Dataset_2
 <pre># Hyperparameters batch_size = 20 num_iters = 20000 input_dim = 32*32 # num_features = 784 num_hidden = 200 output_dim = 10  learning_rate = 0.01</pre>	Accuracy 9.51%	Accuracy 88.31 %
 <pre># Hyperparameters batch_size = 70 num_iters = 80000 input_dim = 32*32 # num_features = 784 num_hidden = 800 output_dim = 10  learning_rate = 0.01</pre>	Accuracy 70.1%	Accuracy 91.02%

#### 5 GitHub Repository:

<https://github.com/sabbirahmedAUST/SoftComputing>