Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

CSE 4238

Soft Computing Lab

Assignment # 02

Submitted To,

Sanzana Karim Lora

Lecturer

CSE, AUST

Submitted By,

MD. Abu Yousuf Sajal

ID 170104025

Date of Submission: September 5, 2021

Dataset Analysis :

Introduction : Bangla Handwritten Digits

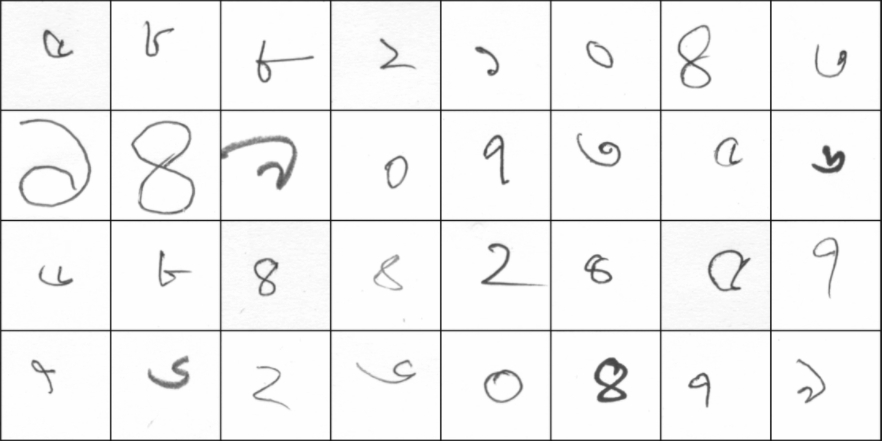
Name : Namta\_DB

Origin : OngkoDB

Database Name : Training-C

Database Size : 24,298 Samples

Contributing Team : BUET\_Backpropers



*A snapshot from DataSet*

Experiments’ Setup

Experiment #1:

*The setup of this experiment is based on the instructions given on the document. The aim of this experiment is to implement the task according to instructor’s setup and process from previous Lab materials.*

Trained & Tested on = Namta\_DB Training-C Database

Train & Test Ration = 4:1

Training Samples = 19456

Testing Samples = 4864

*Hyperparameters*

Image Size = 28\*28

batch\_size = 32

input\_dim = 784

number of hidden layers = 6

First Node = 200

Second Node = 200

Third Node = 200

Fourth Node = 200

Fifth Node = 200

Sixth Node = 200

output\_dim = 10

num\_epochs = 300

learning\_rate = 0.01

optimizer = SGD

Experiment #2:

This setup is based on previous one. But here I have modified hyperparameters as it was mentioned to do in assignment. The aim was to improve accuracy from previous experiment.

Trained & Tested on = Namta\_DB Training-C Database

Train & Test Ration = 4:1

Training Samples = 19456

Testing Samples = 4864

*Hyperparameters*

Image Size = 32\*32

batch\_size = 32

input\_dim = 1024

number of hidden layers = 5

First Node = 2048

Second Node = 1024

Third Node = 512

Fourth Node = 256

Fifth Node = 128

output\_dim = 10

num\_epochs = 300

learning\_rate = 0.1

optimizer = SGD

Basic Comparison Table

[*All the data generated from implementation*]

+--------------+----------------+----------------+

| Basis | Experiment # 1 | Experiment # 2 |

+--------------+----------------+----------------+

| DataSet | Namta\_DB | Namta\_DB |

+--------------+----------------+----------------+

| Train / Test | 19456 / 4864 | 19456 / 4864 |

+--------------+----------------+----------------+

Hyper Parameter Comparison Table

[*All the data generated from implementation*]

+------------------+----------------+----------------+

| HyperParameters | Experiment # 1 | Experiment # 2 |

+------------------+----------------+----------------+

| Image Size | 28 | 32 |

+------------------+----------------+----------------+

| Batch Size | 32 | 64 |

+------------------+----------------+----------------+

| Input Dimension | 784 | 1024 |

+------------------+----------------+----------------+

| Output Dimension | 10 | 10 |

+------------------+----------------+----------------+

| Hidden Layers | 6 | 5 |

+------------------+----------------+----------------+

| Epochs | 300 | 300 |

+------------------+----------------+----------------+

| Learning Rate | 0.01 | 0.1 |

+------------------+----------------+----------------+

| Optimizer | SGD | SGD |

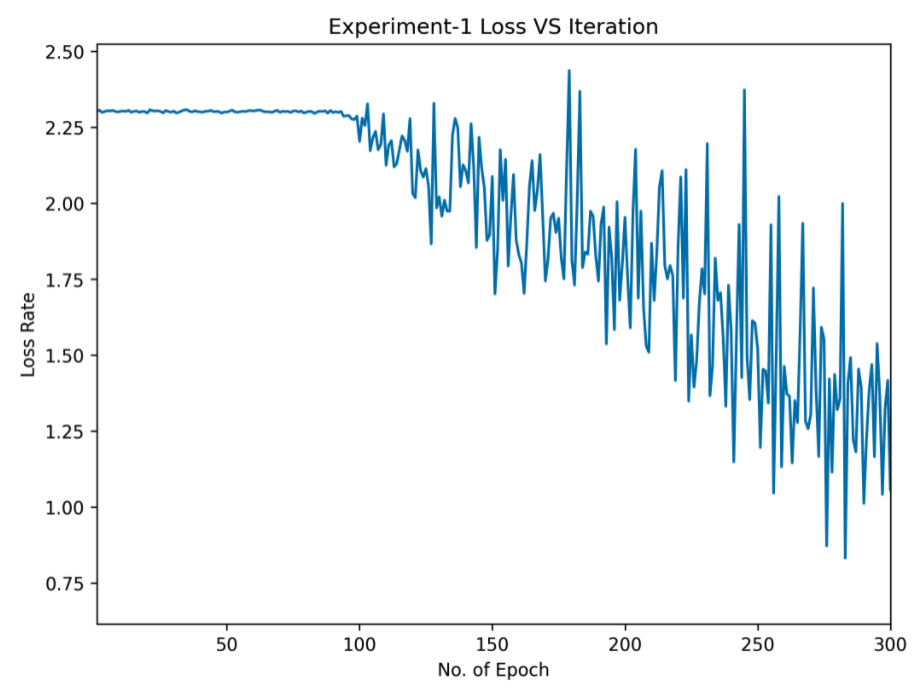
+------------------+----------------+----------------+

Loss VS Iteration Graph

Experiment #1

Initial Loss [Epoch 1] = 2.2993695735931396

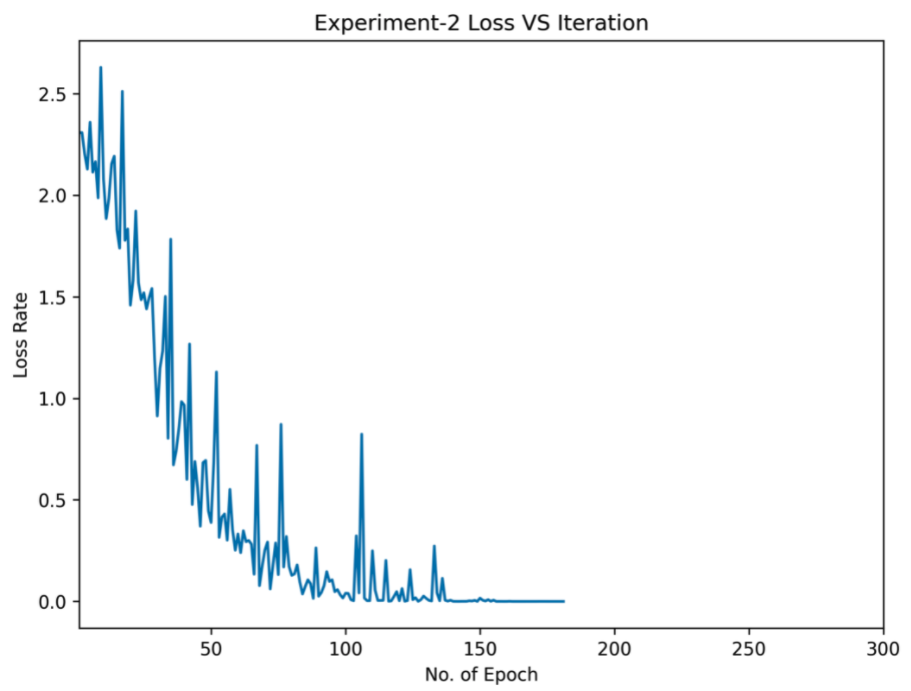
Final Loss [Epoch 300] = 1.072466254234314



Experiment #2

Initial Loss [Epoch 1] = 2.303194999694824

Final Loss [Epoch 300] = 0.0000605798268225044



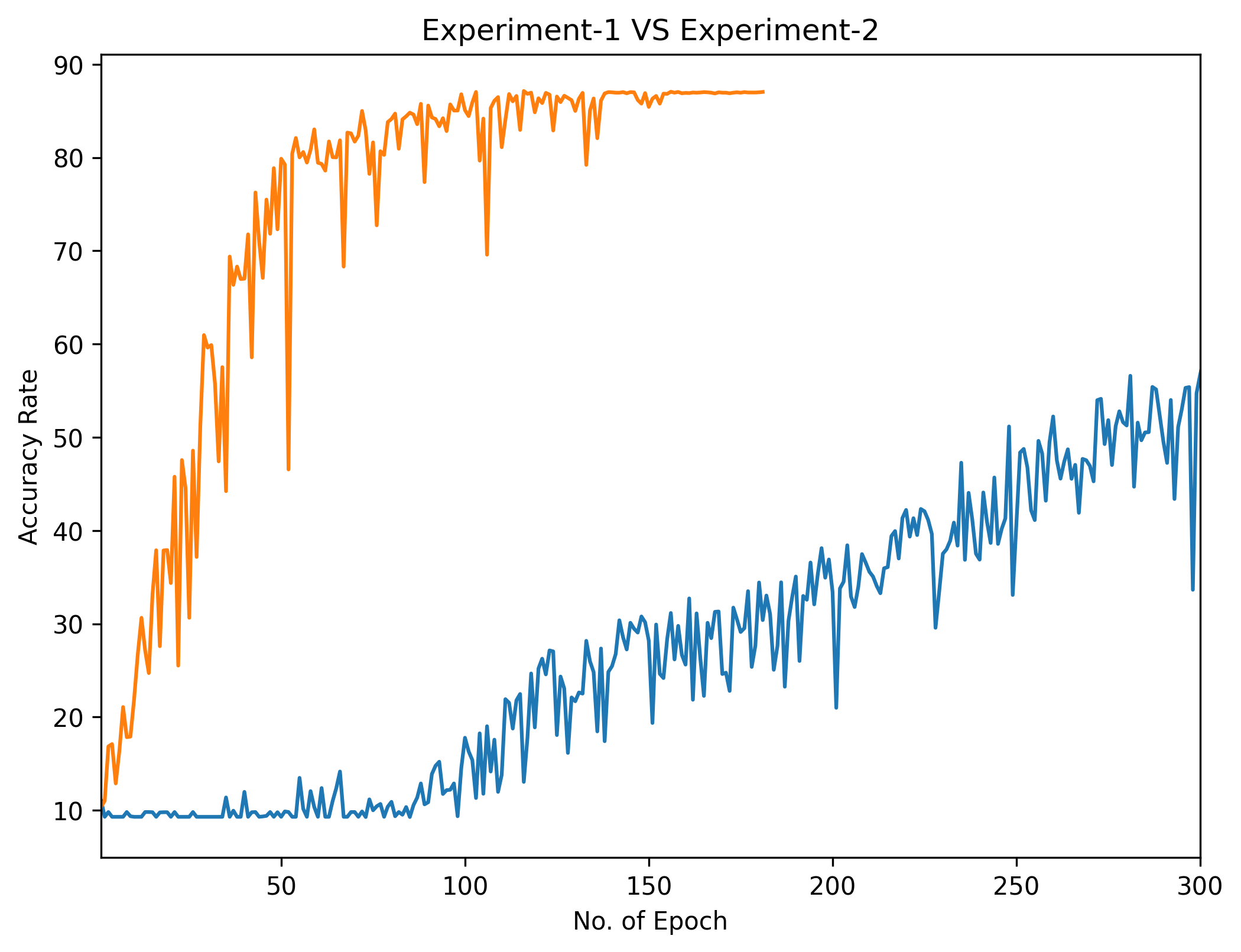
Experiment Result Analysis :

Accuracy :

Here is the final Verdict of our Experiment on Accuracy Metric,

Experiment #1 = 67.48%

Experiment #2 = 87.13%

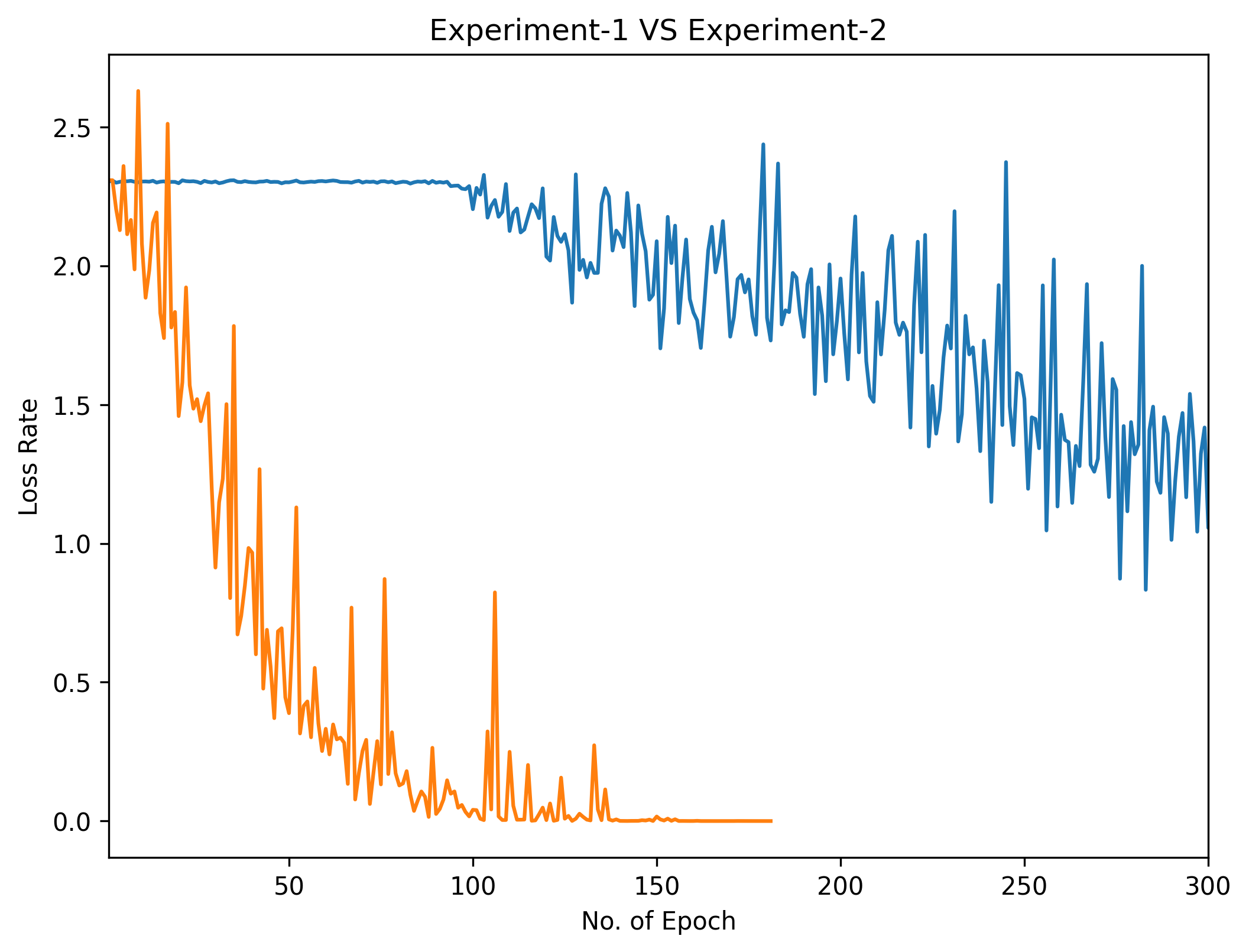


Loss Rates :

The Lowest possible loss value that could be generated is,

Experiment #1 = 9.28

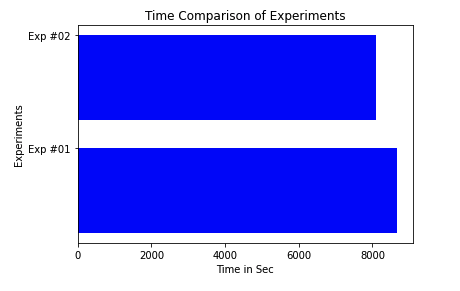
Experiment #2 = 0.000037



Time Comparisons

Experiment #1 = 8654.29 sec == 2h:24m:14s

Experiment #2 = 8088.36 sec == 2h:14m:48s



Conclution :

Finally the experiment has achieved better accuracy than previous. It could kept the goal line as mentioned in the instruction to achieve accuracy above 85%.

Not only its accuracy got better but also it could reduce the run time 10 min!

Code 🡪 [PyTorch Implementation](https://github.com/codexaxor/Soft_Computing/blob/a69bdfc7ace0def99050500fc01eb7bbcf847925/Assignment%202/170104025_exp_02.ipynb)