Winter 2023 COEN 6331 Assignment Report (I)

Jun Huang, Student ID: 40168167
Electrical and Computer Engineering, Concordia University
Email: jun.huang@concordia.ca

***Abstract*—This report presents the experiments and results of building a multilayer perception (MLP) network to observe the effect of different involved factors on the model's performance and the training process. The network is trained for a binary classification problem with two features as input. In this assignment, we examine the different effects of different structure, learning rate, optimizer settings, activation functions, and data normalization on the multilayer perception neuron network. We perform and observe the result of total thirty three experiments for different settings and list their training performance. Finally, based on those experiments, we locate the appropriate setting for the network.**

I. Problem Description

The problem that the model trying to predict if the data point (x_1, x_2) belongs to one of the two classes (C_1 and C_2) in the following coordinate system as Fig. 1 shown. The objective is to build a multilayer perception (MLP) neural network to solve this problem.

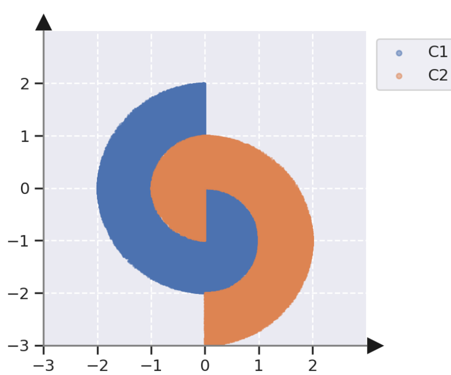

Fig. 1. Non-convex decision regions of the problem.

II. Data Collection

To collect the data, we separate the two decision regions into four regions as Fig. 2 shown. The original C_1 and C_2 regions are separated to C_1a, C_1b, C_2a, and C_2b. The C_1a is the left part of the circle with $(0, 0)$ as the center and radius of the range $[1, 2]$. The C_1b is the right part of the circle with $(0, -1)$ as the center and radius of the range $[0, 1]$. The C_2a is the left part of the circle with $(0, 0)$ as the center and radius of the range $[0, 1]$. The C_2b is the right part of the circle with $(0, -1)$ as the center and radius of the range $[1, 2]$.

Then the data collection of the problem becomes to collecting data points inside a circle with the given center point, radius range, and the range of feature x_1. We formulate

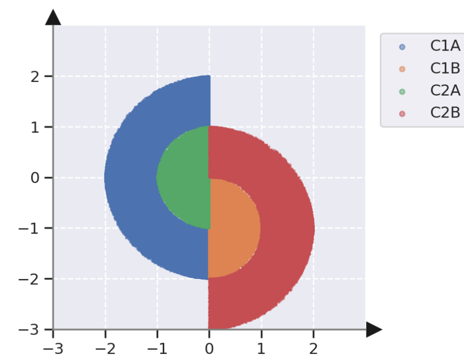

Fig. 2. Separated non-convex decision regions of the problem.

the data generation function based on the circle's equation in parametric form using the trigonometric functions [1] as following:

$$
\begin{aligned}
x_1 &= a + \cos(\theta) \times r \\
x_2 &= b + \sin(\theta) \times r
\end{aligned}
\tag{1}
$$

where (a, b) is the center of the circle, θ is the angle between $[0, 2\pi]$, and r is a random radius or the distance between (a, b) and (x_1, x_2). We then generate 21,004 data samples with two decision classes equally distributed.

III. Experiments and Results

According to the assignment specification, we set six experiments for the six factors involved in the training process: (1) the number of the neurons when building up a singlelayer perception (SLP) neural network; (2) the number of the layers in MLP; (3) the learning rate; (4) the momentum when using the stochastic gradient descent optimizer; (5) the normalization of the dataset; (6) the activation function. Further more, we fixed the loss function to ***Cross-Entropy Loss Function*** for all experiments.

All experiments are conducted under the enverionment of Macbook Pro with M1 chip instead of leveraging GPU. For the purpose of hyperparameters tuning, we split the dataset into three subsets: training set (80%), validation set (10%), and testing set (10%). We use the training set and the validation set to tune and validate the hyperparameters, and use the testing set for the performance testing with the fine-tuned hyperparameters.

A. Different Numbers of Neurons

This experiment compares the SLP network with different numbers of neurons. We fix the following hyperparameters: 0.0005 as learning rate, ReLU as activation function, and stochastic gradient descent as optimizer with 0.9 momentum. This experiment lists the numbers of neurons as follow: 1, 5, 10, 30, 50, 100, 500, and 1,000. The validation performance metrics are shown in Fig. 3. As the figure suggested, if the SLP network has 1, 5, or 10 neurons, it has a low performance within 200 epochs. If the SLP network has more than 50 neurons, it reaches good performance (over 90% accuracy). The performance will not improve much if the SLP has more than 100 neurons since they overlap in accuracy metric.

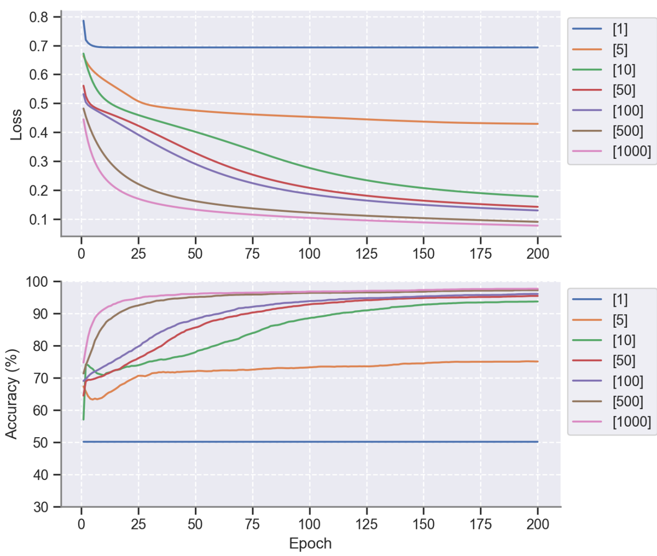

Fig. 3. Loss (L) and accuracy (ACC) performance through 200 epochs for different numbers of neurons. Each line's label is the number of the neuron.

B. Different Numbers of Layers

This experiment compares the MLP network with different numbers of layers under the fixed numbers of neurons in every layer. We fix the following hyperparameters: 0.0005 as learning rate, ReLU as activation function, and stochastic gradient descent as optimizer with 0.9 momentum. With fifty neurons in every layer, this experiment lists the number of layers as follow: 1, 3, 5, 7, and 9. The validation performance metrics are shown in Fig. 4. As the figure suggested, networks with more layers require more epochs to boost the learning.

C. Different Learning Rates

This experiment compares the SLP network with different learning rates. We fix the following hyperparameters: 0.0005 as learning rate, ReLU as activation function, and stochastic gradient descent as optimizer with 0.9 momentum. This experiment lists the learning rates as follow: 0.00001, 0.0001, 0.001, 0.01. The validation performance metrics are shown in Fig. 5. As the figure suggested, the learning rate should set with an appropriate value. Too big or too small will result to performance stagnation (such as the learning rate of 0.01) or slow learning speed (such as the learning rate of 0.00001).

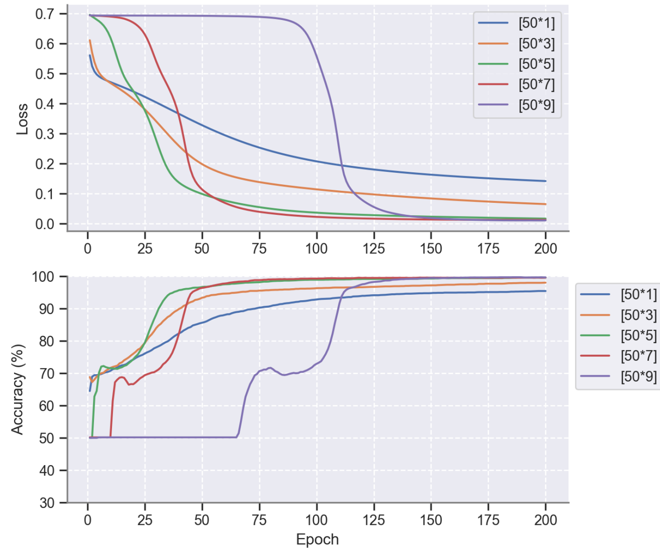

Fig. 4. Loss (L) and accuracy (ACC) performance through 200 epochs for different numbers of layers. The label of the line indicates the [50 * number of layers].

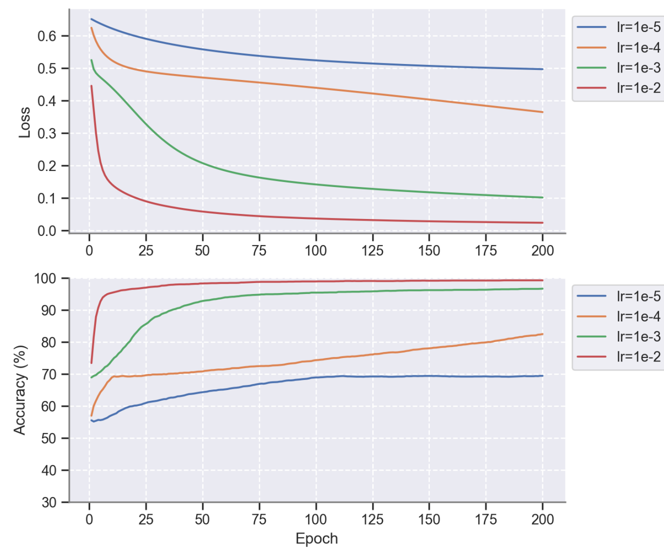

Fig. 5. Loss (L) and accuracy (ACC) performance through 200 epochs for different learning rates.

D. Different Momentum Setting

This experiment compares the MLP network with different momentum settings for the stochastic gradient descent as optimizer. We fix the following hyperparameters: 0.001 as learning rate, ReLU as activation function, and three layers with each has fifty neurons in the MLP network. This experiment lists the momentum as follow: 0.83, 0.86, 0.9, 0.93, 0.96, 1.0. The validation performance metrics are shown in Fig. 6. As the figure suggested, the momentum setting also affects the

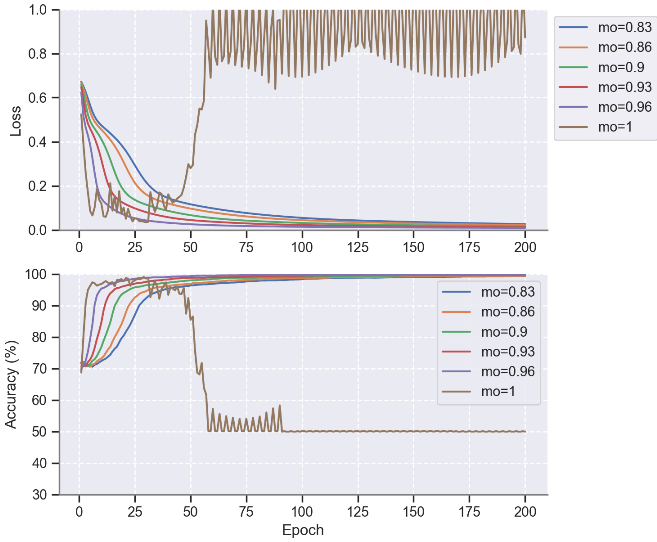

Fig. 6. Loss (L) and accuracy (ACC) performance through 200 epochs for different momentum settings.

optimization process and further affects the learning speed. Not like the learning rate, inappropriate momentum leads the back and forth floating of the loss result.

E. Data Normalization

This experiment compares the MLP network while using original training dataset and the normalized data set. We fix the following hyperparameters: 0.00005 as learning rate, ReLU as activation function, and three layers MLP network settings with each layer has thirty, fifty, and seventy neurons. This experiment lists the momentum as follow: orginal dataset and normalized data set. The validation performance metrics are shown in Fig. 7. As the figure suggested, the momentum

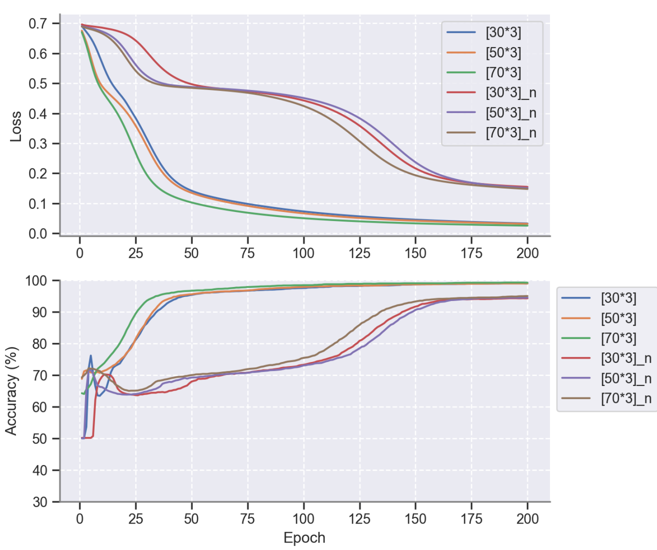

Fig. 7. Loss (L) and accuracy (ACC) performance through 200 epochs for different size of network with the original training data and the normalized data. Labels with '_n' indicate the normalized data result.

setting also affects the optimization process and further affects the learning speed. Not like the learning rate, inappropriate momentum leads the back and forth floating of the loss result.

F. Activation Fucntion

This experiment compares the MLP network with different activation functions. We fix the following hyperparameters: 0.00005 as learning rate, and three layers MLP network settings with each layer has thirty, fifty, and seventy neurons. This experiment lists the activation function as follow: ReLU and Sigmoid. The validation performance metrics are shown in Fig. 8. As the figure suggested, the sigmoid funcion is not

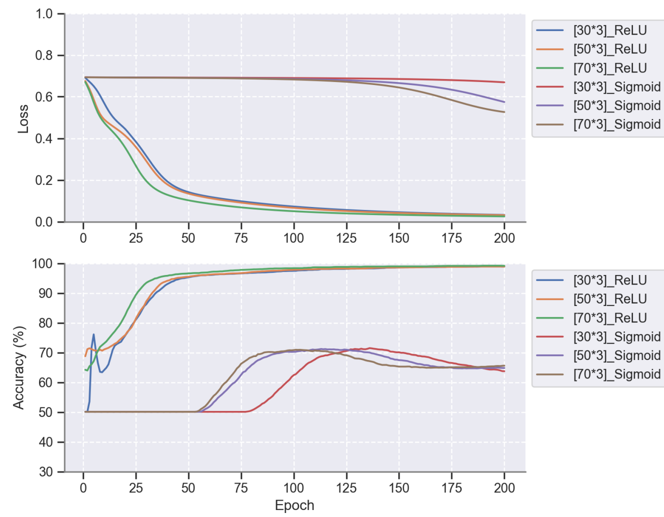

Fig. 8. Loss (L) and accuracy (ACC) performance through 200 epochs for different size of network with the original training data and the normalized data. Labels with '_n' indicate the normalized data result.

suitable for this classification problem.

G. Final Parameters and Network Structure

Finally, based on the experiments, we identify the hyperparameters as:

- number of neurons in each layer: 50
- number of layers: 3
- learning rate: 0.001
- momentum: 0.96
- normalization: false
- activation for each neuron: ReLU

After we train the MLP network with the above hyperparameters, we present performance metrics for the top then networks trained in the experiment in Table. I. We also present the confusion matrix for the final network in Fig. 9.

References

[1] "Wikipedia circle," Feb 2023. [Online]. Available: https://en.wikipedia.org/wiki/Circle

TABLE I
EVALUATION METIRCS FOR THE TOP-10 SETTINGS RANKED BY THE ACCURACY

Setting Label	***Accuracy***	***Precision***	***F1-Score***	***Recall***	***Top-1 Accuracy***	***Training Time in Seconds***
mo=0.93	0.996668	0.996198	0.996671	0.997146	0.996668	32.661339
mo=0.9	0.996192	0.995247	0.996194	0.997143	0.996192	37.016412
lr=1e-3	0.996192	0.995247	0.996194	0.997143	0.996192	32.549158
final	0.995716	0.994297	0.995716	0.997140	0.995716	32.983115
mo=0.96	0.995716	0.994297	0.995716	0.997140	0.995716	32.572386
[50*5]	0.994288	0.995247	0.994302	0.993359	0.994288	36.161434
mo=0.86	0.992861	0.993346	0.992874	0.992403	0.992861	32.075152
lr=1e-2	0.992385	0.998099	0.992439	0.986842	0.992385	34.220032
[70*3]	0.991909	0.992395	0.991924	0.991453	0.991909	35.596409
[70*3]_ReLU	0.991909	0.992395	0.991924	0.991453	0.991909	35.622054

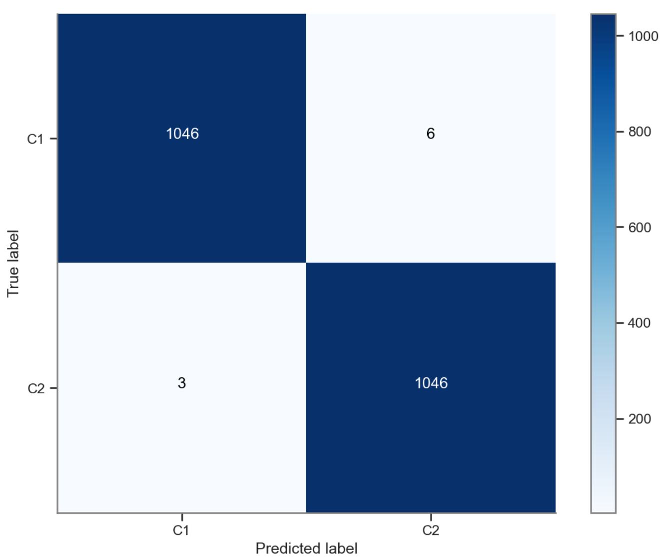

Fig. 9. Confusion matrix for the final network