# **Perceptron Predicts for Diabetes**

Yi Zhong

University of Adelaide

Adelaide, SA 5000

a1710371@adelaide.edu.au

**Abstract**

*This paper analyzes the performance of Single-Layer Perceptron (SLP) and Multi-Layer Perceptron (MLP) models in diabetes prediction, exploring the effects of different learning rates and training epochs on model accuracy and loss convergence. While MLPs are generally expected to outperform SLPs in capturing complex patterns, the results indicate that SLPs can achieve better accuracy and loss reduction in cases involving simpler, linear relationships. The study highlights the importance of model selection based on data complexity and demonstrates how hyperparameter tuning impacts the overall performance of both models.*

Git hub link:

<https://github.com/Pe3kaboo/Perceptron_Predict_diabetes>

# **Introduction**

The perceptron was first proposed by Frank Rosenblatt in 1958 as a simple neural network model for handling binary classification tasks [1]. Over time, the multi-layer perceptron (MLP) was developed to overcome the limitations of the single-layer perceptron (SLP), enabling the model to learn more complex non-linear patterns. In this paper, we compare the performance of SLP and MLP models, with the objective of evaluating how adjusting different parameters, such as learning rates and epochs, affects the performance of these models in predicting diabetes.

## **Diabetes data**

The dataset used in this study consists of various attributes that represent different health metrics of individuals, with each instance labeled as either diabetic (+1) or non-diabetic (-1). Each feature is normalized, and the data includes factors such as plasma glucose concentration, blood pressure, body mass index (BMI), and age, among others, which are used to predict the presence of diabetes (a binary classification task).

## **Process**

This paper first implements a basic single-layer perceptron (SLP) to examine its working principles and identify its limitations. Then, multi-layer perceptron (MLP) models with both a single hidden layer and two hidden layers are built to evaluate their performance in predicting diabetes, and the results are compared with those of the SLP. Through this comparison, the goal is to not only improve prediction accuracy but also to gain insights into how hyperparameter settings, such as learning rates, epochs, and the number of hidden layers, influence the overall performance of the models.

# **Model description**

In this section, a detailed overview of the models used is presented, including the single-layer perceptron (SLP) and multi-layer perceptron (MLP). The SLP is introduced as a baseline model to illustrate the fundamental principles of perceptron learning. To address the limitations of the SLP, the MLP, with one or more hidden layers, is used, enabling the model to capture more complex non-linear relationships in the data. The architecture, mathematical formulations, and learning processes of both models are discussed, followed by a comparison of their performance in predicting diabetes.

## **Single-layer perceptron (SLP)**

The single-layer perceptron (SLP) is designed to perform binary classification tasks by mimicking the basic functions of the brain. The SLP consists of three main components: sensory units (input layer), associative units (processing layer), and response units (output layer) [2].

In the SLP model, inputs from sensory units are connected to associative units in a many-to-many and random manner. Each sensory unit receives input (e.g., a pattern of light or other stimuli) and transmits a signal to the associative unit. The model calculates the weighted sum of the inputs, and if the total exceeds a predefined threshold, the associative unit activates and sends an output signal to the response unit [2].

Mathematically, SLP can be described as follows:

Where:

* represents the input features
* are the weights associated with the inputs
* is the bias term
* is the activation function

Since the data is categorized as 1 and -1 for diabetes and non-diabetes, the activation function can be defined as:

During training, the SLP adjusts its weights and bias through reinforcement learning, but its inability to solve non-linearly separable problems, like the exclusive OR problem, due to its linear decision boundary, led to the development of MLP, which uses hidden layers to capture non-linear patterns [2].

## **Multi-layer perceptron (MLP)**

Multilayer Perceptron (MLP) is a type of feedforward artificial neural network (ANN) and one of the most widely used neural network architectures. Unlike single-layer perceptron (SLP), which is limited to solving linearly separable problems, MLP can solve more complex, nonlinearly separable problems by introducing one or more hidden layers [3].

MLP consists of three types of layers:

* **Input layer**: receives the input features.
* **Hidden layer**: process information using non-linear activation functions. MLPs can have one or more hidden layers, with each neuron in the hidden layer connected to every neuron in the previous layer (fully connected). The complexity of the decision boundaries grows with the number of hidden layers and neurons [3].
* **Output layer**: produces the final prediction.

The signal flows in a forward direction, from input to output, without any loops, which characterizes MLP as a feed-forward network. The output of each neuron is computed as a weighted sum of its inputs passed through an activation function. The typical activation functions used in MLP are non-linear, such as the sigmoid or hyperbolic tangent function, allowing the network to capture non-linear patterns in the data [3]​.

The mathematical representation of the output of a neuron in the hidden layer is:

Where:

* is the output of the -th neuron in the output layer.
* are the weights between the hidden neuron and the output .
* is the bias of the output neuron

MLP is trained using the backpropagation algorithm, which adjusts weights by minimizing error, enabling it to model complex decision regions and approximate continuous functions with one hidden layer, while additional layers enhance its ability to capture intricate patterns [3].

# **Model implementation**

In this section, the implementation details of both the single-layer perceptron (SLP) and multi-layer perceptron (MLP) models are presented. The models were developed using Python, leveraging libraries such as NumPy and PyTorch for efficient computation. The SLP serves as the baseline model, while the MLP is tested with two configurations: one with a single hidden layer of 10 neurons, and another with two hidden layers, containing 10 and 5 neurons, respectively. Key aspects of the code, including model architecture, training procedures with different learning rates and epochs, and evaluation metrics, are discussed.

## **Hyperparameters and training setup**

In this section, the overall training setup and the hyperparameters used for both the single-layer perceptron (SLP) and multi-layer perceptron (MLP) models are outlined. Both models were trained using different combinations of learning rates (0.1, 0.01, 0.001) and epochs (100, 500), allowing for a comparison of their performance under varying conditions.

For the MLP, two configurations were used:

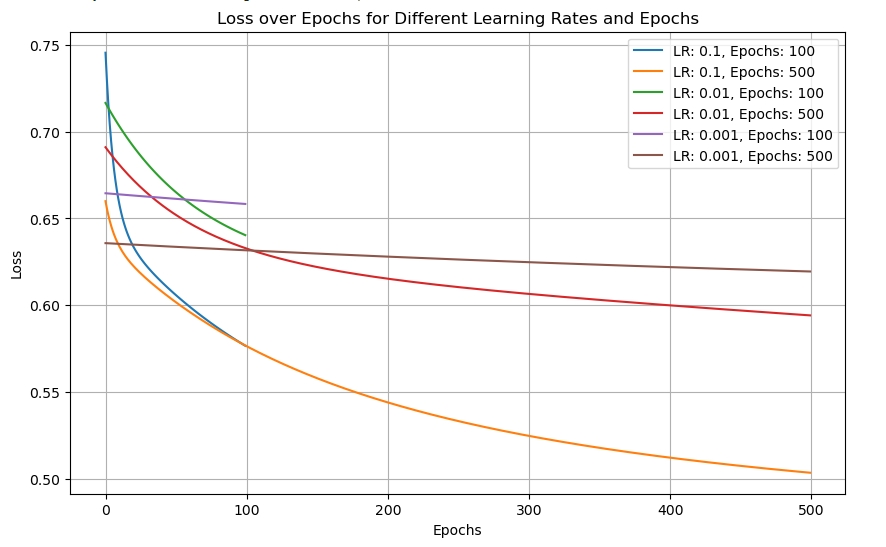
One Hidden Layer: 10 neurons in the hidden layer.

Two Hidden Layers: 10 neurons in the first hidden layer and 5 neurons in the second hidden layer.

These hyperparameters were chosen to examine how different learning rates, training durations, and hidden layer configurations affect the models' ability to minimize loss and improve accuracy. Both models were evaluated on the same test set for consistency.

## **SLP result**

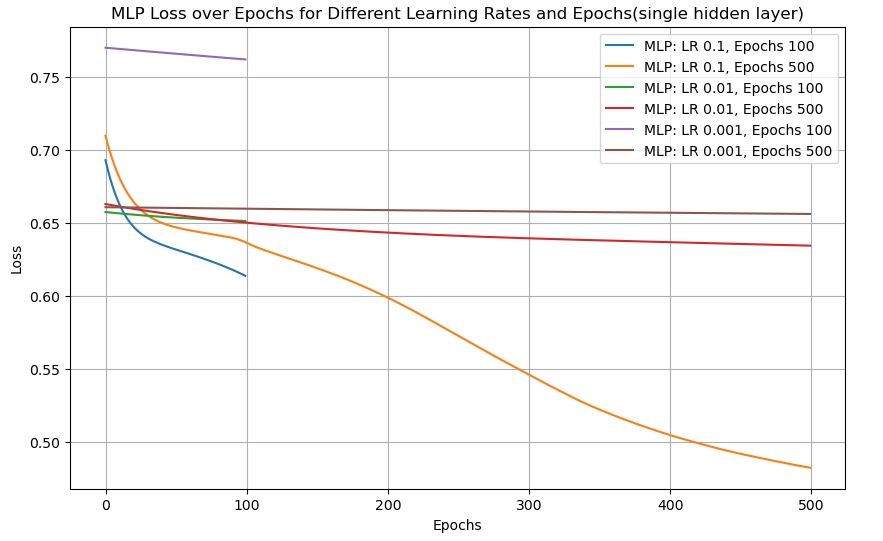
The Single-Layer Perceptron (SLP) model was trained with different learning rates and epochs, and the loss curves over the training process are visualized below to illustrate the model's convergence behavior.



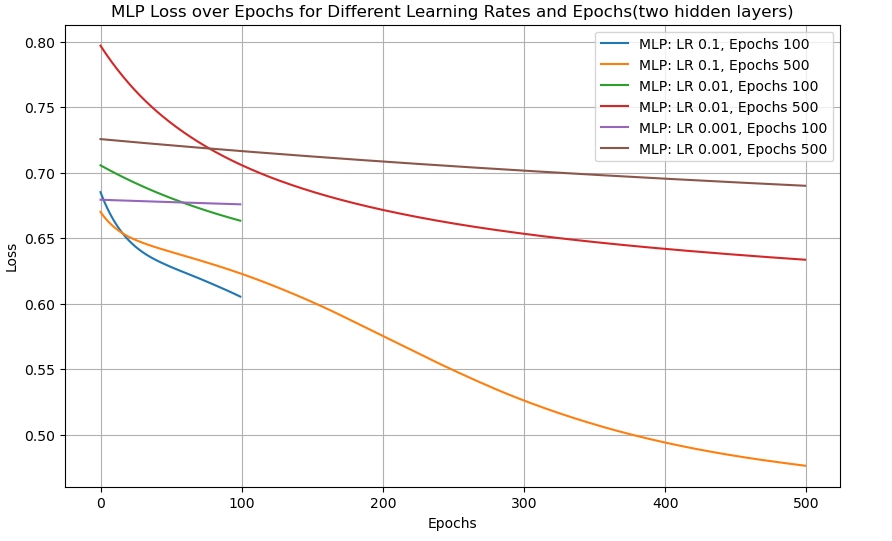
**Fig.1.** Loss over Epochs for Different Learning Rates and Epochs for the SLP Model

## **MLP result**

The Multi-Layer Perceptron (MLP) model, with both one and two hidden layer configurations, was trained using various learning rates and epochs, and the loss curves are visualized below to illustrate the model's convergence and performance compared to the SLP.



**Fig.2.** Loss over Epochs for Different Learning Rates and Epochs for the MLP (One Hidden Layer)



**Fig.3.** Loss over Epochs for Different Learning Rates and Epochs for the MLP (Two Hidden Layers)

## **Comparison of loss curves between SLP and MLP’s.**

## 

**Fig.4** Loss Curves between SLP and MLPs

This figure compares the loss curves for the Single-Layer Perceptron (SLP), MLP with one hidden layer, and MLP with two hidden layers. The comparison highlights the improved convergence and lower loss achieved by the MLP models, particularly the two hidden layer configuration, over the SLP. All models were trained with a learning rate of 0.1 and 500 epochs.

# **Result summary**

* **Single-Layer Perceptron (SLP)**

The SLP model was trained using different learning rates and epochs. With a learning rate of 0.1 and 500 epochs, the model achieved a test accuracy of 75.32%, showing that although the SLP performs reasonably well, its ability to capture more complex patterns is limited compared to the MLP models. The loss curves indicate a gradual reduction in loss, but with a slower convergence compared to the MLP models.

* **MLP with One Hidden Layer**

The MLP with one hidden layer (10 neurons) showed improved performance over the SLP. With a learning rate of 0.1 and 500 epochs, the model achieved a test accuracy of 72.08%. The model's loss reduced more quickly than the SLP, and the final loss value was lower, reflecting its ability to learn more complex representations.

* **MLP with Two Hidden Layers**

The MLP with two hidden layers (10 neurons in the first layer and 5 neurons in the second) demonstrated the best performance. With a learning rate of 0.1 and 500 epochs, the test accuracy reached 75.32%, matching the performance of the SLP but with faster convergence and a significantly lower final loss value. The loss curves indicate that the two hidden layer MLP was able to better fit the data over the course of training.

* **Comparison of SLP and MLP**

As shown in the comparison graph, both MLP models outperform the SLP in terms of faster convergence and lower final loss values, especially when trained with longer epochs and larger learning rates. The two hidden layer MLP consistently showed the best performance, indicating its superior ability to capture non-linear patterns in the data.

# **Conclusion**

This paper implemented and evaluated the performance of both Single-Layer Perceptron (SLP) and Multi-Layer Perceptron (MLP) models for predicting diabetes. The results clearly demonstrate the limitations of the SLP in handling complex, non-linear patterns, as reflected by its slower convergence and inability to reduce loss as efficiently as the MLP models. While the SLP achieved a test accuracy of 75.32%, its loss curve showed gradual convergence compared to the faster and more efficient MLP models.

The MLP with one hidden layer demonstrated improved performance, achieving a test accuracy of 72.08% and faster convergence than the SLP. However, the best results were obtained from the MLP with two hidden layers, which achieved the same accuracy of 75.32% as the SLP but with significantly lower loss and faster convergence.

Overall, the paper highlights the advantages of using deeper neural networks, such as MLPs, for binary classification tasks like diabetes prediction. The results show that adding hidden layers allows the model to better capture complex patterns in the data, leading to faster convergence and improved overall performance. Future work could explore optimizing the number of neurons in each hidden layer, experimenting with different activation functions, or applying these models to other classification tasks to further enhance their predictive power.

# **References**

1. R. D. L. Cruz, "Frank Rosenblatt’s Perceptron, the birth of the neural network," *Medium*, 2023. [Online]. Available: <https://medium.com/@robdelacruz/frank-rosenblatts-perceptron-19fcce9d627f>.
2. H. D. Block, "The Perceptron: A Model for Brain Functioning," \*Rev. Mod. Phys.\*, vol. 34, no. 1, pp. 123–135, Jan. 1962..