·

**Predict diabetes using Perceptron**

Hu Xiaolong

The University of Adelaide

Adelaide SA 5005

A1912235@adelaide.edu.au

**Abstract**

*This paper will provide process on implementing, describing, and testing a specific algorithm called Perceptron, which can be interpreted as a dense layer neural network, for predicting diabetes using the related diabetes dataset. And will compare the accuracy and analysis results between the single-layer perceptron and MLP models.*

# **Introduction**

Diabetes is a global health problem with an increasing prevalence rate, which has placed a heavy burden on individuals and society. According to the World Health Organization, diabetes has become one of the leading causes of death worldwide, and prevention and early diagnosis are essential to reduce the incidence [1]. Therefore, developing effective prediction models to identify high-risk individuals has become an important research direction in the field of public health.

## **Purpose**

This paper aims to predict diabetes using machine learning algorithms, especially single-layer perceptrons and multi-layer perceptrons (MLPs). Perceptrons are simple linear classifiers that can effectively perform binary classification tasks by learning the relationship between input features and target labels. Despite their simple structure, they still show good results in many practical applications. On the other hand, MLPs, as a more complex neural network structure, can capture nonlinear relationships in data through multiple hidden layers, thereby improving classification performance.

## **Process**

This paper will introduce basic single-layer perceptron to understand its working principle and limitations. Then, building an MLP model to explore its performance in diabetes prediction and compare it with the results of a single-layer perceptron. Through this process, hope to not only improve the prediction accuracy, but also gain a deeper understanding of the relationship between model performance and hyperparameter settings.

# **Method Description**

The following will explain the algorithm and the model used in detail, and provide examples to illustrate its effects and shortcomings. Some possible improvements and their working principles are also attached to facilitate implementation by others.

## **Single-layer perceptron**

The single-layer perceptron is one of the simplest neural network models suitable for linearly separable binary classification problems. Its basic structure includes an input layer and an output layer, where each node in the input layer corresponds to a feature, and the output node generates the prediction [2]. The functioning of the single-layer perceptron is as follows:

1. **Input Features**: The input data is represented as a feature vector x=[x\_1,x\_2,…,x\_n].
2. **Weights and Bias**: Each input feature is assigned a weight w\_i and a bias b.
3. **Linear Combination**: The weighted sum is calculated as
4. **Activation Function**: The linear combination is passed through an activation function to produce the output. For binary classification tasks, the step function is commonly used:
5. **Weight Update**: The weights and bias are updated using the perceptron learning rule when the prediction does not match the true label:

where η is the learning rate.

While the single-layer perceptron is simple to implement, it can only handle linearly separable problems and cannot solve complex non-linear classification tasks.

## **Multi-Layer Perceptron**

The multi-layer perceptron (MLP) is an extension of the neural network that includes at least one hidden layer, allowing it to capture non-linear relationships within the input data [3]. The structure and functioning of MLP is as follows:

1. **Network Structure**: An MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer’s nodes are interconnected by weights.
2. **Forward Propagation**:Input data is processed through each layer using weighted sums and activation functions, which may include ReLU, Sigmoid, or Tanh. The output layer produces the final prediction through an activation function.
3. **Loss Function**: An appropriate loss function (such as cross-entropy loss) is used to assess the model's prediction performance.
4. **Backpropagation**:Gradients for each weight are computed using the chain rule, and weights are updated using gradient descent. In addition, the backpropagation process involves calculating errors and updating weights layer by layer from the output layer back to the input layer.

By increasing the number of hidden layers and nodes, MLP can significantly enhance the model's expressive power, making it suitable for more complex tasks.

## **Improvement**

This paper gives result of impact of different hyperparameter settings (such as learning rate and number of hidden layers) on model performance. By systematically adjusting these parameters, it aim to improve predictive accuracy and gain deeper insights into how hyperparameter selection affects model training.

# **Experimental Analysis**

The following shows the data preprocessing steps, the experimental setup for testing the models, and the results obtained from different configurations of the Perceptron and Multilayer Perceptron (MLP) models.

## **Data Preprocessing**

The dataset used for this experiment was preprocessed before training the models. First, the labels with format of +1 and -1, were converted into 0 and 1 for binary classification. Since the number of features in each sample was expected to be 8, rows with a different number of features were filtered out. Additionally, no missing values ​​were found in the data, to ensure that the data was clean and ready for model training. Finally, the dataset was split into training and test sets, using an 80/20 ratio (test\_size=0.2) and a fixed random seed (random\_state=42) to ensure reproducibility.

## **Model Testing and Hyperparameters**

Both Single Layer Perceptron (SLP) and Multilayer Perceptron (MLP) were extensively tested with different hyperparameters such as learning rate and number of epochs. The following settings were tested:

**Learning rate: 0.1, 0.01, and 0.001**

**Number of epochs: 200 and 500**

The models were evaluated based on the loss reduction during training and the accuracy achieved on the test set.

## **SLP Results**

# **Final copy**

# **References**

1. WHO, “Diabetes,” *World Health Organization*, 2024. <https://www.who.int/health-topics/diabetes#tab=tab_1>
2. H. D. Block, "The Perceptron: A Model for Brain Functioning," *Rev. Mod. Phys.*, vol. 34, pp. 123–134, Jan. 1962.
3. A. J. Shepherd, *Second-order Methods for Neural Networks: Fast and Reliable Training Methods for Multi-layer Perceptrons*, 1st ed., Perspectives in Neural Computing, Springer, 1997.