Utilizing deep learning techniques based on densely connected neural networks for diabetes prediction

Danyu Huang
University of Adelaide
a1837292@adelaide.edu.au

Abstract

Diabetes is a chronic metabolic disorder resulting from either abnormal insulin secretion or function. It is commonly manifested as hyperglycemia and is associated with numerous serious health complications. This condition arises when the pancreas is unable to produce sufficient quantities of insulin, or when the body is unable to effectively utilize the insulin that is produced[1]. This paper presents an algorithm based on perceptron neural networks for the prediction of diabetes using the Pima Indians Diabetes Database. The algorithmic structure comprises a single-layer perceptron, which maps the input features directly to the output prediction through a linear transformation, thereby facilitating the processing of input data. To enhance the model's generalization capacity, principal component analysis is employed during the data preprocessing phase to reduce the dimensionality while maintaining the most salient features. Furthermore, the optimal number of neurons in the perceptron is ascertained through hyperparameter optimization. The model is trained on distinct training cycles to assess its influence on performance and identify the optimal training cycle to minimize the risk of overfitting.

1. Introduction

As indicated by data from the Centers for Disease Control and Prevention, diabetes affects over 37 million Americans, representing more than 8.5 percent of the adult population[2]. This chronic disease is associated with elevated blood sugar levels and often presents with significant complications, including cardiovascular disease, kidney disease, and vision impairment[3]. Among American indigenous groups, the Pima Indians, a specific tribe in Arizona, have one of the highest rates of type 2 diabetes globally, which has attracted extensive research attention. It is of particular importance to predict and intervene early on in the lives of the Pima Indian population. The establishment of effective prediction models allows for the identification of individuals at high risk and the implementation of targeted interventions at an early stage, thereby slowing the progression of the disease and improving health outcomes.

The perceptron algorithm is selected as the fundamental model for effectively addressing the straightforward binary classification issue. Throughout the process, a series of preprocessing steps, including optimal k, learning rate selection, data normalization, missing value processing, and categorical variable conversion, are employed to enhance the model's performance. The training cycle is evaluated and verified to minimize the risk of overfitting and guarantee the model's generalizability on the test set.

2. Description of the method

The Pima Indians Diabetes Database, provided by the National Institute of Diabetes and Digestive and Kidney Diseases in the United States, was utilized in the methodology. The dataset comprises a number of medical predictor variables and a target variable, which is predicted by the data based on individual medical indicators. The outcome predictor variable comprises a number of factors, including the number of pregnancies, BMI, insulin level, age, and so forth. These are classified as binary variables, with a value of 1 indicating the presence of diabetes and a value of 0 indicating the absence of diabetes.

The perceptron[4] is capable of mapping input features to the predicted result through the use of a linear equation. Each row represents a data record of a patient, and each column represents a specific feature. As a linear classification model, the perceptron is well-suited to the handling of binary classification problems.

The perceptron algorithm is follow as[5]:

Assume $g(x; w) = \text{sign}(\langle x, w \rangle)$, where $x, w \in \mathbb{R}^d$, $y \in \{1, -1\}$.

where x represents the input feature vector, w denotes the weight vector, and the notation $x \cdot w$ is used to denote the dot product between x and w. The function sign() returns a value of 1 if the result of the dot product is positive and a

value of -1 if it is negative. This allows for the generation of a binary classification decision. The target output y takes values from the set 1, -1, where 1 indicates one class and -1 indicates the other class.

In this method, $x \in \mathbb{R}^d$ represents the features including Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age. The function sign$(\cdot)$ returns 1 if the dot product $\langle x, w \rangle$ is positive, indicating the presence of diabetes; if it is negative, the function returns -1, indicating the absence of diabetes.In a binary classification system, the values 0 and 1 represent two distinct categories. In this context, the absence of diabetes is represented by the value 0, while the presence of diabetes is represented by the value 1. In the training process, the weight, represented by the weight w, is adjusted by the optimization algorithm in order to enable the model to better fit the training data.

2.1. Data Preprocessing

In the preliminary examination of the data set, no instances of missing values were identified, suggesting that some data may have been populated with default values (such as 0) due to the absence of other values. It is thus imperative to replace the value of 0 with NaN in order to guarantee the veracity of all values. In the event of missing data, the mean is employed for features exhibiting more significant deficiencies, whereas the median is utilized for features displaying less pronounced deficiencies. This approach allows for the effective maintenance of the data's central tendency while avoiding any impact on subsequent analysis.

Pregnancies	0	Pregnancies	0
Glucose	0	Glucose	5
BloodPressure	0	BloodPressure	35
SkinThickness	0	SkinThickness	227
Insulin	0	Insulin	374
BMI	0	→ BMI	11
DiabetesPedigreeFunction	0	DiabetesPedigreeFunction	0
Age	0	Age	0
Outcome	0	Outcome	0
dtype: int64		dtype: int64	

Figure 1. After processing, the dataset detected missing data in features such as SkinThickness, Insulin, Glucose, BMI, and BloodPressure. In particular, there was a significant amount of missing data for SkinThickness and Insulin.

2.2. Eliminate outliers and conduct model training

After visualizing each feature using a box plot, the results indicate that the majority of features exhibit a considerable number of outliers. In statistical analysis, an outlier is defined as a data point that falls outside the upper and lower limits of the box. Typically, outliers are found in positions beyond the 75th percentile or below the lower whisker. The identification and exclusion of these outliers can reduce noise and enhance the accuracy of the model.

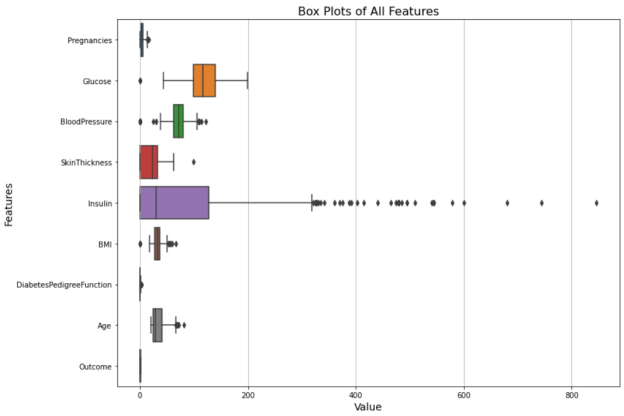

Figure 2. The box plot indicates that there are more Pima Indian women in the dataset with higher insulin levels, and there are also more individuals with higher blood pressure and BMI values.

2.3. Feature standardization

Standardization is a commonly used technique in data preprocessing[6], aiming to adjust the numerical values of features to a relatively uniform scale. Standardization achieves this by subtracting the mean of each data point and dividing it by its standard deviation, resulting in a feature with a mean of 0 and a standard deviation of 1, with the goal of eliminating the scale differences between features.

The standardization algorithm can be expressed as:

$$z = \frac{x - \mu}{\sigma}$$

where $\mathbf{w} = [w_1, w_2, \ldots, w_n]$ are the weights, $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ are the inputs, and b is the bias term.

The sign function outputs the predicted class based on the weighted sum.

Once the features have been standardized by Z-score, each feature will be situated within a reasonable range, thereby enhancing the stability of the model and improving the efficiency of the training process. This standardized approach guarantees that all features are on the same scale, facilitates the optimization algorithm's convergence, and mitigates the training instability that may arise from varying feature scales.

3. Experimental Analysis

In this method, we utilized the perceptron as the model. We preprocessed the model and processed the features in an appropriate manner without affecting the fundamental structure of the model. Our objective is to enhance the performance of the model and improve the efficiency of its training.

3.1. Principal Component Analysis

Principal component analysis (PCA) is a common technique for reducing the dimensionality of data sets[7], which is widely used in machine learning. The method functions by transforming high-dimensional variables from extensive data sets into smaller principal components while maintaining as much of the information present in the original data as possible. In particular, principal component analysis transforms relevant variables into new, uncorrelated variables by identifying the most significant directions within the data set.

Before performing principal component analysis, normalization is required. After normalization, the mean of all features is 0 and the variance is 1.Following standardization, principal component analysis (PCA) will transform the standardized data into a matrix format for the purpose of reducing the dimensionality of the data set and calculating the principal components.

The covariance matrix $\mathbf{C}$ is calculated as:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X}$$

where $\mathbf{X}$ is the data matrix, and n is the number of observations.

PCA aims to find the principal components by performing eigenvalue decomposition of the covariance matrix:

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

where λ_i represents the eigenvalue corresponding to the eigenvector $\mathbf{v}_i$. The eigenvectors of the covariance matrix represent the directions of maximum variance in the data, while the eigenvalues indicate the magnitude of the variance along those directions.

By selecting the top k eigenvectors (those with the largest eigenvalues), PCA reduces the dimensionality of the dataset while preserving as much variance as possible. This transformation can be expressed as:

$$\mathbf{Z} = \mathbf{X}\mathbf{V}_k$$

where $\mathbf{Z}$ is the transformed dataset and $\mathbf{V}_k$ is the matrix of the top k eigenvectors.

3.2. Learning rate

Learning rate is a hyperparameter in machine learning used to control the magnitude of weight updates during training[8]. In this study, three different learning rates (0.1, 0.01, and 0.001) were selected for experiments to evaluate their impact on model performance.

0.01 learning rate generally performs well in all models, especially in the normalized model, where the training and test sets reached a high accuracy after 100 epochs, indicating that 0.01 learning rate may be the most suitable choice.

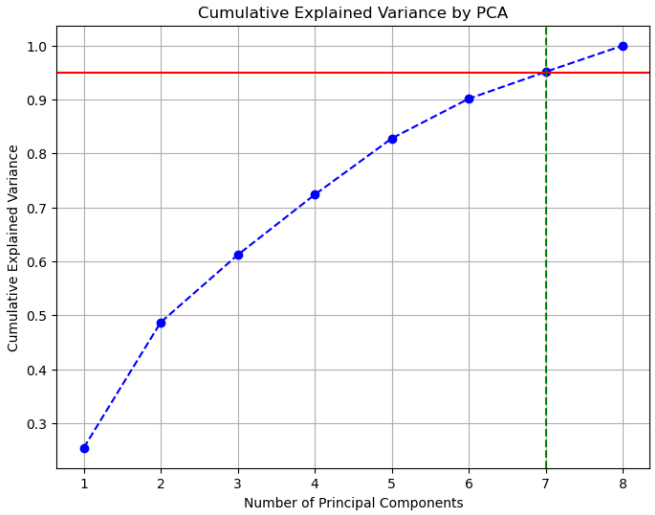

Figure 3. Figure Number of Principal Components depicts the cumulative explained variance, which reaches 95 percentage points when the number of principal components is set to 7. By selecting these seven components from the Pima Indians Diabetes Database, the model is able to retain the majority of the important information while reducing the dimensionality of the data.

Table 1. Model performance with learning rate of 0.01

Model Type	Epochs	Train Accuracy (%)	Test Accuracy (%)
Original	10	73.45	64.49
Original	100	67.82	61.59
Original	500	76.55	71.01
Standardized	10	40.91	43.48
Standardized	100	70.55	64.49
Standardized	500	77.27	73.19
BestK	10	40.91	43.48
BestK	100	70.55	64.49
BestK	500	77.27	73.19

In contrast, 0.1 learning rate also achieved high accuracy in both the original model and the normalized model. However, the model's performance appeared to be less stable in epochs.

Table 2. Model performance with learning rate of 0.1

Model Type	Epochs	Train Accuracy (%)	Test Accuracy (%)
Original	10	65.31	64.29
Original	100	34.69	35.71
Original	500	65.31	64.29
Standardized	10	73.27	66.67
Standardized	100	79.27	76.81
Standardized	500	79.64	76.09
BestK	10	67.27	70.29
BestK	100	78.00	74.64
BestK	500	79.82	76.09

Finally, the learning rate of 0.001 performed poorly over-

all, and the model's performance was less stable compared to the other two learning rates, with lower accuracy than the other learning rates. This may be because the model converges slowly during training, causing the problem, so 0.001 may not be an appropriate learning rate.

Table 3. Model performance with learning rate of 0.001

Model Type	Epochs	Train Accuracy (%)	Test Accuracy (%)
Original	10	36.55	42.03
Original	100	58.73	53.62
Original	500	60.55	59.42
Standardized	10	63.64	64.49
Standardized	100	56.36	63.04
Standardized	500	63.09	59.42
BestK	10	63.64	64.49
BestK	100	56.36	63.04
BestK	500	63.09	59.42

3.3. Conclusion

Prior to the training and optimization of the model, we preprocessed the data and segmented the dataset. In this study, perceptron algorithm is mainly used, which is suitable for binary classification problems. The rationale for selecting this algorithm is based on its simplicity, effectiveness, and applicability to linearly separable cases. In addition, principal component analysis (PCA) was used to reduce the dimensionality of the dataset. Select a suitable learning rate for comparison, and use SelectKBest for feature selection to improve the accuracy of the model.

The experimental results demonstrate that PCA enhances the model's generalization capacity by reducing the number of features and retaining the most representative principal components within the data, thereby mitigating the interference from noise and redundant features. An appropriate learning rate can facilitate the model's convergence and optimize the training process. Concurrently, SelectKBest efficiently diminishes the dimensionality of the features by identifying those that are strongly correlated with the target variable, thus circumventing the potential overfitting issues associated with high-dimensional data. The final experimental results demonstrate that the application of PCA, learning rate optimization, and SelectKBest significantly enhances model performance. The combination of these methods renders the model more robust when dealing with complex data and improves prediction accuracy.

Ultimately, we attained a 76.09 percentage accuracy rate without compromising the fundamental structure of the model, effectively averting the issue of overfitting.The objective of this method is to gain a deeper understanding of the processes involved in model selection and performance evaluation.In the near future, I intend to test the performance of more complex models in order to identify potential improvements.It is my hope that I will be able to gain further insights in the future, which will allow me to enhance the performance of the model. Based on my current understanding, I believe there is still potential for optimization of the model I created.

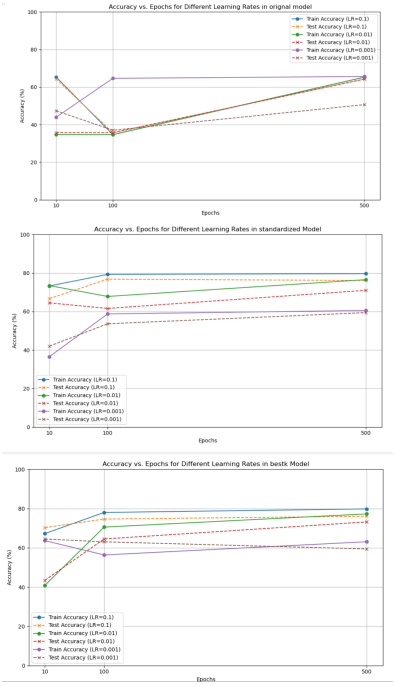

Figure 4. The accuracy of the training set and the test set can be significantly improved by the damage curve.

4. Code

Code on GitHub: https://github.com/Shironana7/deep-learning.git.

5. Reference

1. American Diabetes Association. Standards of medical care in diabetes—2007. Diabetes Care, 30(1): S4–S41, 2007.

2. Centers for Disease Control and Prevention. National Diabetes Statistics Report, 2020.

3. C. Liu, J. Chen, and H. Zha. A generalized approach to multi-dimensional time series forecasting. arXiv preprint arXiv:1912.02315v2, 2019.

4. H. A. Simon. Theories of decision-making in economics and behavioral science. Psychological Review, 63(2):129–138, 1956.

5. G. A. Fink. Markov models for pattern recognition: From theory to applications. In: B. Furht, Encyclopedia of Machine Learning and Data Mining. Springer, 642–644, 2017.

6. R. J. Smith. Understanding the legal implications of artificial intelligence. New York Law Review,

94(1):100–120, 2019.

7. A. Maćkiewicz and W. Ratajczak. Principal components analysis (PCA). Computers Geosciences, 19(3):303–342, 1993.

8. R. A. Jacobs. Increased rates of convergence through learning rate adaptation. Computational Statistics Data Analysis, 24(2):295–307, 1997.