A Single Layer Perceptron-based Approach for Diabetes Data Classification

Aditya D Pandey
The University of Adelaide
Adelaide, South Australia, Australia
aditya.pandey@student.adelaide.edu.au

Abstract

Diabetes is one of the leading causes of morbidity and mortality worldwide. Diagnosis of diabetes is a crucial step in the treatment of diabetes. In this study, we introduce an approach based on perceptron, for categorizing diabetes data by harnessing the capabilities of neural networks to enhance accuracy and efficiency. The dataset comprises of 768 women with age of more than 21 years. We leveraged the model to perform training and evaluating it on different learning rates. We achieved an accuracy rate of 80.78%, with an ideal learning rate of 0.01.

1. Introduction

Diabetes is a long-term condition that impacts individuals across the globe [14]). Precise categorization of diabetes information plays a role, in identifying it making diagnoses and devising effective treatment strategies [17].Over the years the application of machine learning methods has demonstrated encouraging outcomes in analyzing medical data particularly when it comes to classifying diabetes [8], [18], [3], [21]. This study introduces an approach based on perceptron, for categorizing diabetes data by harnessing the capabilities of neural networks to enhance accuracy and efficiency.

2. Method

In this section we will delve into the materials and methods utilized in the research study. Subsequently we will provide a breakdown of the dataset, methods and approach, in subsequent subsections.

2.1. Dataset

The dataset utilized in this research corresponds to the Pima Indian community situated near Phoenix, Arizona, a dataset that has been continuously examined since 1965 by the National Institute of Diabetes and Digestive and Kidney Diseases [24] PIMA Indian diabetes dataset (PIDD) consists of 9 attributes, 8 predictors and 1 class label. The dataset comprises of 768 women with age of more than 21 years. Above depicted are histogram plots and table with description and for each of attribute(s) of PIDD [16]

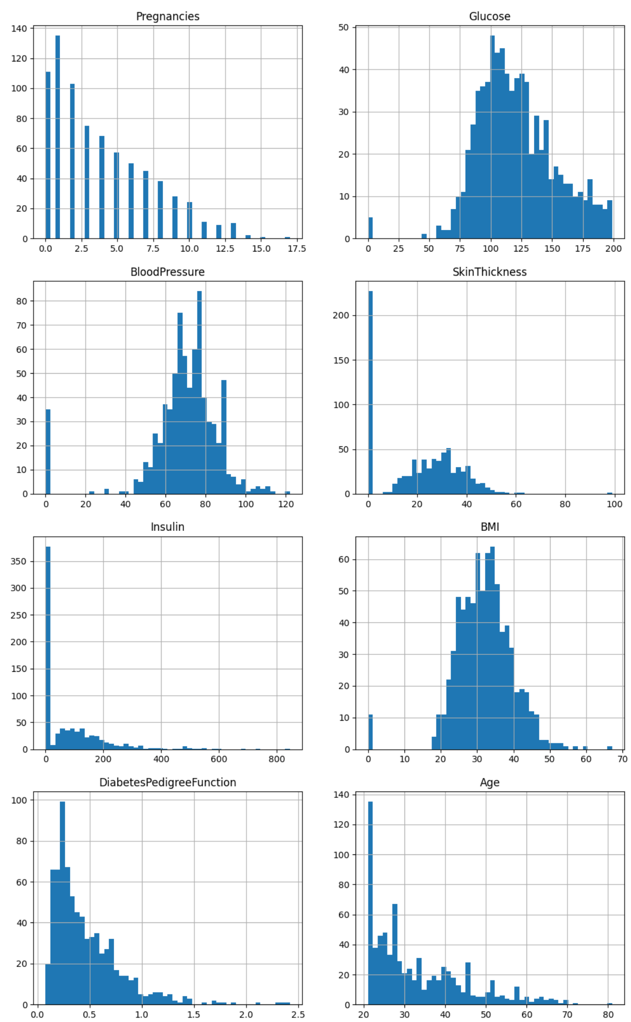

Figure 1. A histogram for each numerical attribute

Sr. No.	Selected Attributes from PIDD	Description of selected attributes	Range	Attribute Type	Data Type
1	Pregnancies	Number of times a participant is pregnant	0-17	Feature	Numerical
2	Glucose	Plasma glucose concentration a 2 hours in an oral glucose tolerance test	0-199	Feature	Numerical
3	Blood Pressure	It consists of Diastolic blood pressure (when blood exerts into arteries between heart)(mm Hg)	0-122	Feature	Numerical
4	Skin Thickness	Triceps skinfold thickness (mm). It is concluded by the collagen content	0-99	Feature	Numerical
5	Insulin	2-Hour serum insulin (mu U/ml)	0-846	Feature	Numerical
6	BMI	Body mass index	0-67.1	Feature	Numerical
7	Diabetes Pedigree Function	Synthesis of the diabetes mellitus history in relatives and the genetic relationship of those relatives to the subject[24]	0.078-2.42	Feature	Numerical
8	Age	Age of participant	21-81	Feature	Numerical
9	Outcome	Diabetes class variable, Yes represent the patient is diabetic and no represent patient is not diabetic	Yes/No	Target	Categorical

Table 1. Description of all PIDD attributes

2.2. Single-Layer Perceptron (SLP)

The Single-Layer-Perceptron is the most basic Artificial Neural Network architectures, the model was invented by Frank Rosenblat [22]

A Perceptron can be described as a linear threshold device that computes weighted sum of the coordinates of the pattern vector, compares the value with a threshold, and outputs +1 or -1 if the threshold is reached, threshold is identified as the activation function that we employ [23]. The below diagram depicts the concept [7]. SLP even though being the most elementary form of ANN still find use in multitude of fields of study, Alkhamees utilized optimized SLP for classification in context of fetal state detection [1], SLP to analyze laboratory data [5], for prediction of Bankruptcy [11], and data security [15]. The core formula behind SLP is

$$z = w_0 \dot{1} + w_1 x_1 + w_2 x_2 + ... + w_n x_s$$

As is depicted in Figure : 2, The inputs $x_0, x_1, ...x_n$ are multiplied with the individual weights $1, w_1, x_2, ...x_n$, further down the line they are summed up. Then the weighted sums are pass through a step function also known as activation function or transfer function, which as described below.

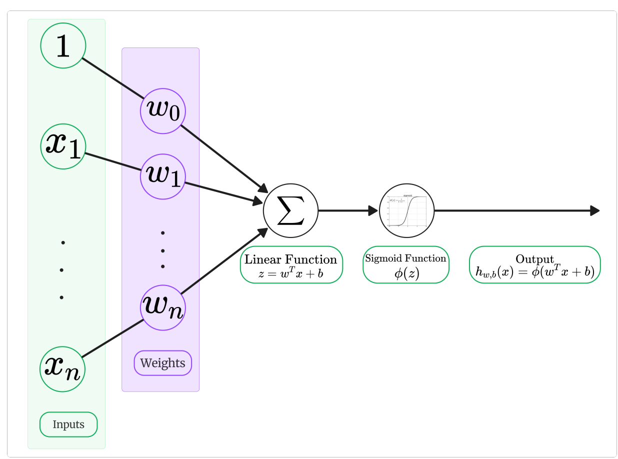

Figure 2. The Components of Single Layer Perceptron

$$\phi(z) = \frac{1}{1 + \exp -z}$$

In order to simplify the calculation and reduce the pro-

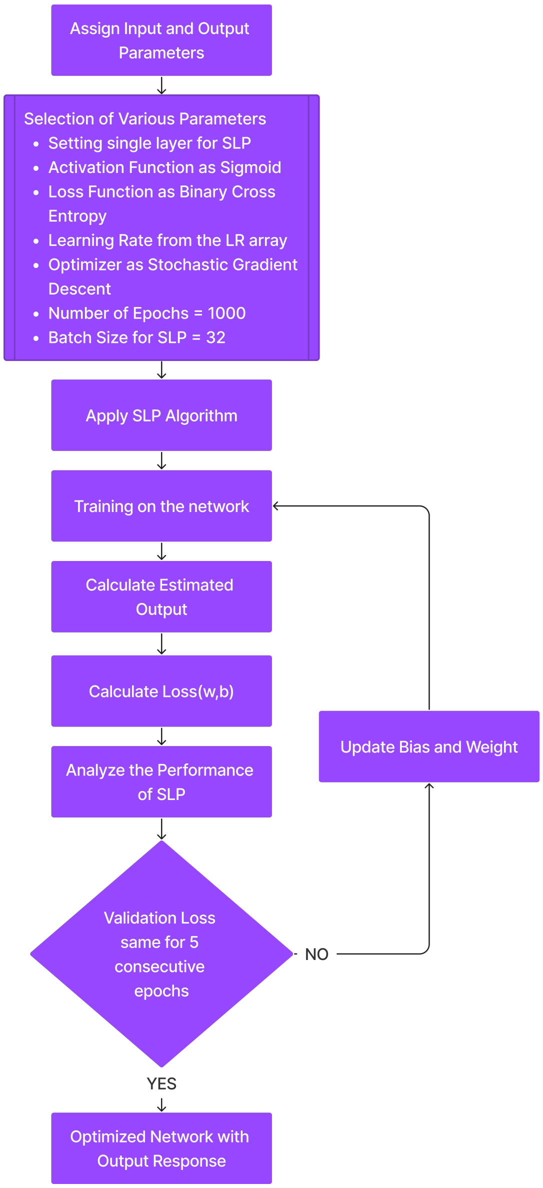

Figure 3. Flowchart for SLP Training

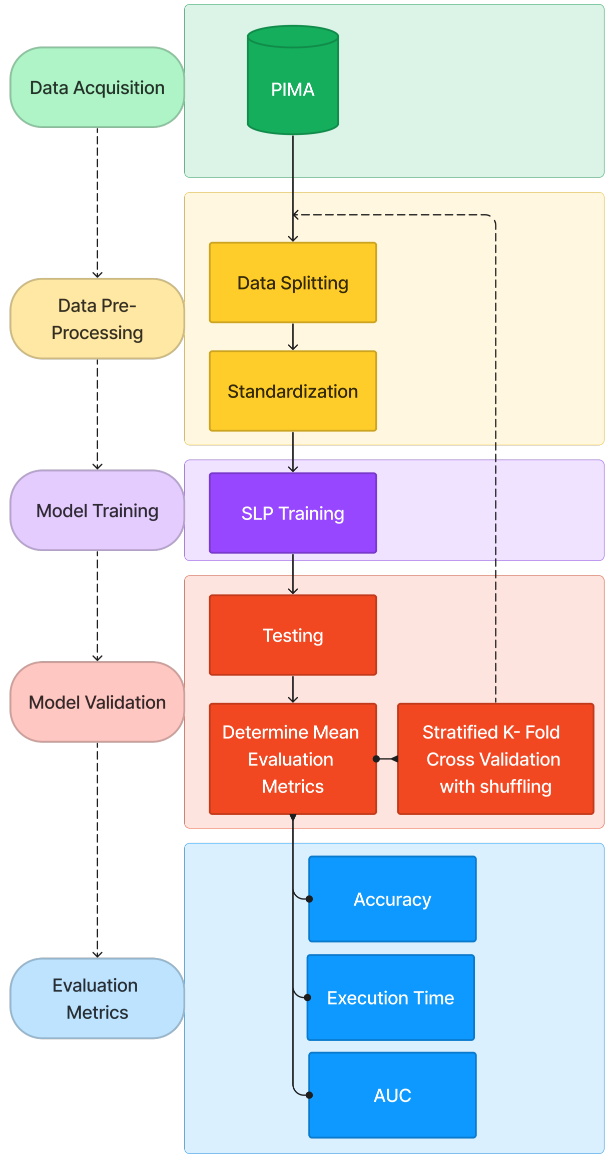

Figure 4. Flowchart for ML Pipeline

cessing delay, we employ the Sigmoid activation function a nonlinear activation function to provide clear probabilistic outputs, enabling straightforward interpretation for the binary classification. The next part involves training and updating of the weights, given by the equation below.

$$w_i^{\text{next step}} = w_{i,\,j} + \eta\,(y_i - \hat{y}_i)\,x_i$$

If training instances are linearly separable, Rosenblatt demonstrated that this algorithm would converge to a solution. This is called the perceptron convergence theorem [22]. As depicted above is the flowchart Figure: 3 for the Algorithm with respect to implementation in the code for PIDD.

2.3. Model Pipeline

The representative pipeline for the Training of the SLP Algorithm. The PID Dataset for this project is downloaded from online Platform Kaggle [13]. The data is split and standardized [4]. Further in the pipeline we train our SLP model as is described in Figure : 4, with Binary Cross Entropy to calculate the loss and Stochastic gradient descent as the optimizer. We set an epoch limit of 1000 but along with that we implement an early stopping call-back [20]), which would stop the model training if the validation loss tends to remain unchanged for 5 consecutive epochs.

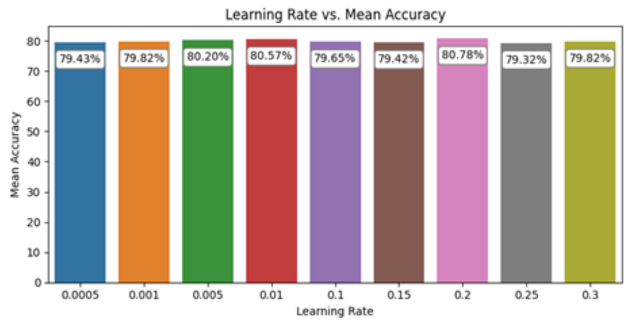

Figure 5. Accuracies of SLP Model at different Learning Rate

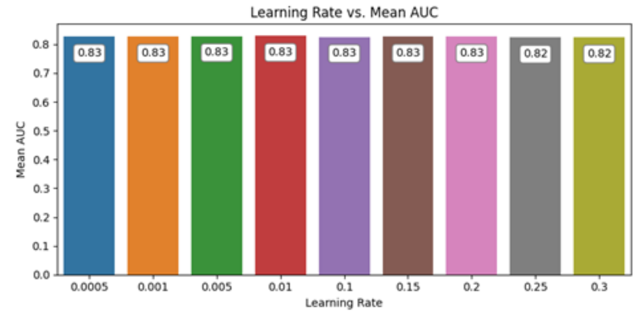

Figure 6. Mean AUC Score at different Learning Rates

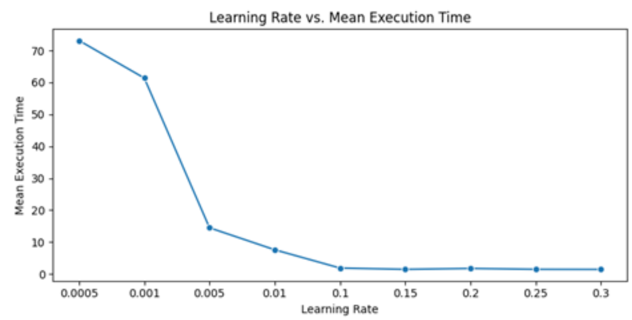

Figure 7. Enter Caption

2.4. Experimental Analysis

We evaluate the SLP Output using five-fold cross-validation method [6]), we divide the dataset into k=5 partitions with shuffling, one of these set is used for testing, other 4 are used for training. The outcome of each fold is used to compute mean metrics for evaluation.

We evaluate the model using metrics such as accuracy, Execution time, AUC Score

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Here, TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative. TP (TN) represents the number of observations in the positive (negative) class that are classified as positive (negative), and FP (FN) represents the number of observations in the negative (positive) class that are classified as positive (negative).

We train the SLP model with a 5-fold cross validation technique and it trains based on the learning rates which are 0.0005, 0.001, 0.005, 0.01, 0.1, 0.15, 0.2, 0.25, 0.35. The below graph in Figure 5, depicts that there is minor deviation in term of accuracy and the highest being 80.78 % which is obtained by 0.2 learning rate.

Area Under the Curve (AUC) Score is an indicator of the ability of classifier to distinguish between different classes [9]. It represents the area under the ROC curve, a graph where True positive rate is traced by the probabilities curve to different thresholds against the False positive rate. A higher value would imply better model performance. As per Figure 6, we can observe that the AUC score of approximately 0.83 across different Learning rates.

Execution Time represents the amount of time taken by CPU to train the algorithm for each fold, we then take mean execution time. A lower Execution time translates to lesser resources will be used by the algorithm.

As is evident from Figure 7, there is a reduction in Execution time as we increase the Learning rate up-till 0.1 after which the reduction in execution time is minimal. Though one must also note that the stopping condition of training of our algorithm pertains to Early stopping callback function, for which there is a condition of patience level of 5.

2.5. Conclusion and Future Work

This research is intended to determine the performance metrics of Single Layer Perceptron in classifying if the patient is diabetic or not. We used five-fold cross-validation into training and test sets to obtain mean values of evaluation metrics owing to the size of dataset which does not warrant the use of larger data splitting. We leveraged the model to perform training and evaluating it on different learning rates, based on that we achieved an accuracy rate of 80.78%, a learning rate of 0.01 would be ideal for further research. There is scope of achieving a higher accuracy if a more complex artificial neural network model is used for this dataset. The need for a larger dataset with balanced values for outcomes would be beneficial.

2.6. Code

My code and data is available at `https://github.com/aditya-524/DL_Ass1/blob/main/implementation_alpha.ipynb` The code utilizes the python packages as such matplotlib [12] & seaborn [25], pandas [26], numpy [10], scikit-learn [19], Keras [2].

References

[1] Bader Fahad Alkhamees. An optimized single layer perceptron-based approach for cardiotocography data classification. *International journal of advanced computer science applications*, 13(10):239–245, 2022.

[2] François Chollet et al. Keras. https://keras.io, 2015.

[3] Aman Darolia and Rajender Singh Chhillar. Analyzing three predictive algorithms for diabetes mellitus against the pima indians dataset. *ECS transactions*, 107(1):2697–2704, 2022.

[4] Lucas B.V. de Amorim, George D.C. Cavalcanti, and Rafael M.O. Cruz. The choice of scaling technique matters for classification performance. *Applied soft computing*, 133:109924–, 2023.

[5] J. J. Forsström, K. Irjala, G. Selén, M. Nyström, and P. Eiund. Using data preprocessing and single layer perceptron to analyze laboratory data. *Scandinavian journal of clinical laboratory investigation. Supplement*, 55(S222):75–81, 1995.

[6] Tadayoshi Fushiki. Estimation of prediction error by using k-fold cross-validation. *Statistics and computing*, 21(2):137–146, 2011.

[7] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow concepts, tools, and techniques to build intelligent systems*. O'Reilly, Sebastopol, CA, third edition. edition, 2023.

[8] Mariwan Ahmed Hama Saeed. Diabetes type 2 classification using machine learning algorithms with up-sampling technique. *Journal of Electrical Systems and Information Technology*, 10(1):8–10, 2023.

[9] D.J. Hand and C. Anagnostopoulos. When is the area under the receiver operating characteristic curve an appropriate measure of classifier performance? *Pattern recognition letters*, 34(5):492–495, 2013.

[10] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy. *Nature (London)*, 585(7825):357–362, 2020.

[11] Yi-Chung Hu. Bankruptcy prediction using electre-based single-layer perceptron. *Neurocomputing (Amsterdam)*, 72(13):3150–3157, 2009.

[12] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in science engineering*, 9(3):90–95, 2007.

[13] UCI Machine Learning. Pima indians diabetes database, 2016.

[14] Basith K. Moien Abdul, Muhammad J. Hashim, Kwan K. Jeffrey, Devi G. Romona, Halla Mustafa, and Juma A. Kaabi. Epidemiology of type 2 diabetes – global burden of disease and forecasted trends. *Journal of Epidemiology and Global Health*, 10(1):107–111, 03 2020. Copyright - © 2020. This work is licensed under http://creativecommons.org/licenses/by-nc/4.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2020-12-01.

[15] D Mualfah, Y Fatma, and R A Ramadhan. Anti-forensics: the image asymmetry key and single layer perceptron for digital data security. *Journal of Physics: Conference Series*, 1517(1):12106–, 2020.

[16] Huma Naz and Sachin Ahuja. Deep learning approach for diabetes prediction using pima indian dataset. *Journal of diabetes and metabolic disorders*, 19(1):391–403, 2020.

[17] World Health Organization. *Classification of diabetes mellitus*. World Health Organization, 2019.

[18] Radhanath Patra and Bonomali khuntia. Analysis and prediction of pima indian diabetes dataset using sdknn classifier technique. *IOP conference series. Materials Science and Engineering*, 1070(1):12059–, 2021.

[19] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12:2825–2830, 2011.

[20] Lutz Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural networks*, 11(4):761–767, 1998.

[21] Rajni Rajni and Amandeep Amandeep. Rb-bayes algorithm for the prediction of diabetic in "pima indian dataset". *International journal of electrical and computer engineering (Malacca, Malacca)*, 9(6):4866–4872, 2019.

[22] F Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, 1958.

[23] Kai-Yeung Siu, Amir Dembo, and Thomas Kailath. On the perceptron learning algorithm on data with high precision. *Journal of computer and system sciences*, 48(2):347–356, 1994.

[24] Jack W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, and R.S. Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings - Annual Symposium on Computer Applications in Medical Care*, pages 261–265, 1988.

[25] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[26] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.