

Predicting Diabetes Through Perceptions: A Novel Approach to Early Detection and Risk Assessment

Santiago Reyes Vargas – a1865001
The University of Adelaide

[SantRV/DLF-Diabilities-Perceptron: Predict Diabilities using perceptron. \(github.com\)](https://github.com/SantRV/DLF-Diabilities-Perceptron)

Abstract

This report delves into the development of a software solution for early diabetes prediction, focusing on the implementation of Perceptron (P) and Multi-layer Perceptron (MLP) models with varied parameter configurations. With diabetes mellitus affecting a significant global population and necessitating daily attention to lifestyle, diet, and blood glucose monitoring, exploring alternative indicators for early diagnosis is paramount.

Our study utilizes deep-learning techniques and data processing methodologies, including data normalization, layered architecture, and batch optimization, to analyze a dataset provided by the National Institute of Diabetes and Digestive and Kidney Diseases. This dataset comprises 768 records of females aged 21 and older of Pima Indian heritage, featuring eight key attributes and a binary target variable for diabetes diagnosis.

The Multi-layer Perceptron emerges as a promising model, demonstrating optimal performance with scheduled learning rates and a hidden layer of 16 nodes. Our findings showcase accuracy ranging from 60% to 75%, providing a potential avenue for early diabetes prediction.

This report serves as an exploration of data-driven methods for diabetes prediction and underscores the importance of alternative diagnostic indicators in the field of healthcare.

1. Introduction

Representing a series of metabolic conditions associated with hyperglycaemia¹ and caused by partial or total insulin insufficiency, diabetes mellitus affects over 422 million people globally which require daily attention to diet, lifestyle, and self-monitoring of blood glucose [1]. While this condition is generally diagnosed by measuring the plasma glucose concentration over a 2 – 3-month period [2], it has been suggested that related conditions include elevated adiposity², high insulin, age and blood pressure can be used for the assessment of diabetes existence on a patient [3].

With the aim of developing a software solution that could allow both patients and health practitioners to predict a diabetes diagnosis prior to measuring plasma glucose levels, this research paper focuses on the implementation of a Perceptron (P) and a Multi-layer Perceptron (MLP) model with varied parameters configurations. Hence, this paper builds on the experimentation conducted on the data with deep-learning techniques and data processing

¹ High amounts of glucose in the blood.

² Body fat concentration

methodologies which include data normalisation, layers architecture, and batch optimisation.

The dataset provided by the National Institute of Diabetes and Digestive and Kidney Diseases consists of a single csv file with 768 records of females aged 21 and over of Pima Indian heritage. The dataset contains 8 features namely: pregnancies, glucose, blood pressure, skin thickness, insulin levels, BMI, diabetes pedigree function, and age. One target feature is present as a binary value that indicates the diagnosis of diabetes. Therefore, the performance of the proposed models is limited to the data provided and can only provide a binary forecast, not a diagnostic result for all types of diabetes.

Preliminary data analysis suggests that the existence of strong correlations between glucose, blood pressure and skin thickness and a positive diabetes diagnosis, Figure 1. Also, it was found that the distribution of the data is skewed towards no-diabetes diagnosis, Figure 2. Hence, it becomes crucial to normalise the data during training to accelerate convergence and generalise the model given the low number of positive diabetes datapoints [4].

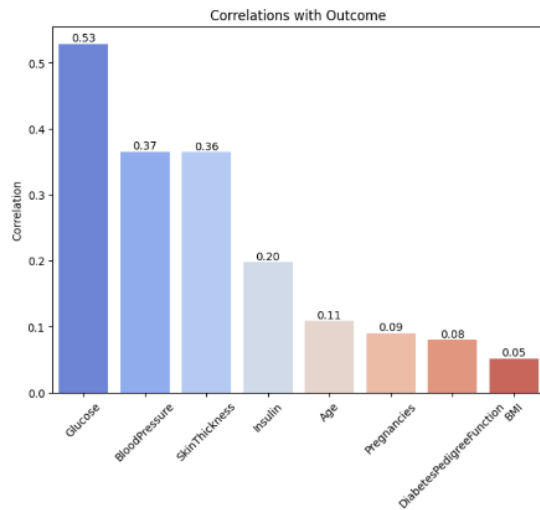


Figure 1 – Bar plot depicting the correlation between the data features and the binary diagnosis of diabetes.

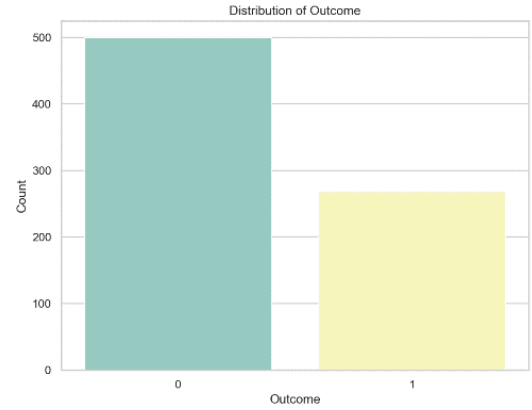


Figure 2 – Bar plot depicting the data distribution for the binary target feature of diabetes diagnosis.

2. Literature Background

Deep neural networks implementations have expanded across the medical field over the last two decades within the areas of medical image analysis, clinical diagnosis, and biological modelling [5]. The predictive diagnosis of diabetes has been addressed by numerous competitors, thus in this section we briefly review the current literature that aligns with this project's aims.

Machine learning approaches taken include the use of support vector machines (SVM) and K-means clustering algorithms for diagnosis. While research conducted by Alseema et al suggest that information on sex, smoking and family history of diabetes can improve predictions [6], the outcomes of studies by Abnoosian et al suggests that a framework with strong data pre-processing, k-folds cross-validation and grid search can achieve high average accuracy, precision, recall, F1-score, and AUC values of 0.9887, 0.9861, 0.9792, 0.9851, and 0.999, respectively [7].

In addition, the use of scheduled learning rates during model training seem to provide further performance improvement by allowing neural networks to escape local minima and expand the search space exploration over the epoch iterations.

This allows the model to find multiple local minima which can be then ensembled to further target a global minimum [8]

3. Methodology

The proposed neural networks were developed using Pytorch and Numpy and the main approach taken was to split data into a data loader with batch size of 5, 10, 50 and 100, normalise the data during training using dropout and batch Norm1, have a series of 0 to 3 hidden layers with nodes from 16 to 64, and to implement a linear learning rate scheduler.

3.1. Neural Network models

Three neural networks were developed with increasing complexity in the number of hidden layers, all are fully connected and have a one-node output layer that denotes the final binary prediction, Figure 3. For data pre-processing the models expect a target feature with values of 0 or 1 instead of -1 and 1 as provided in the dataset.

3.1.1. Perceptron:

The perceptron is the most basic model with one linear layer with 8 input nodes and one sigmoid activation function, Figure 3.

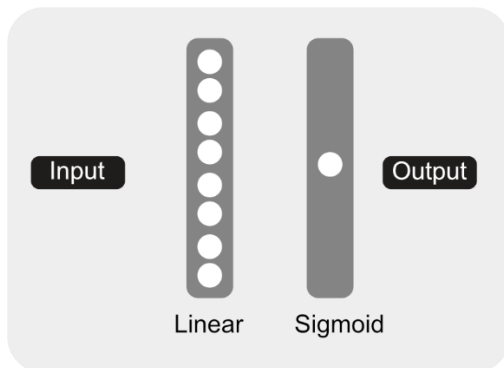


Figure 3 – High-level perceptron layer architecture.

3.1.2. Multilayer Perceptron:

This neural network incorporates a total of 6 layers, including one hidden layer with a default size of 16 nodes. It implements a ReLU and Sigmoid functions as well as a BatchNorm1d one. While ReLU leads to faster convergence during training compared to Sigmoid and Tanh, Sigmoid was used after the hidden layer to map a real value number to the range 0, and 1 as these problems require a binary classification, Figure 4.

This model implements regularization techniques to improve its training and generalization performances. Dropout helps to prevent overfitting by randomly deactivating a fraction of neurons during each forward and backward pass. Thus, zeroing out the weights of such neurons. On the other hand, batch normalization is used to accelerate the training and make the model more stable by subtracting the batch mean. In addition to dividing by the batch standard deviation following the scaling and shift of normalized values using learnable parameters.

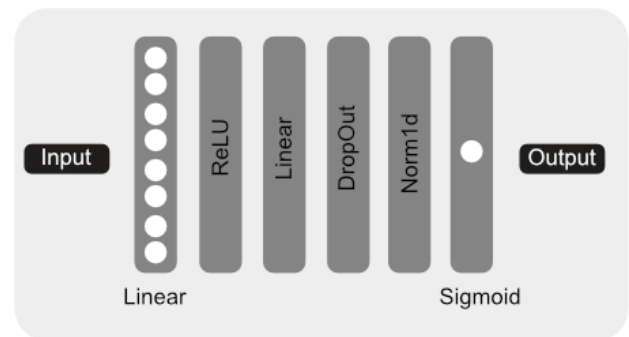


Figure 4 – High-level multi-layer perceptron (MLP) layer architecture.

3.1.3 Deep Multilayer Perceptron

This deep neural network consists of a minimum of 5 layers, 3 of which are hidden by default, one input layer and one output layer. Although the hidden layers can be set as a parameter to this model, the preset contains 2 layers with 16 nodes and 1 layer with 32 nodes. Similar to the multilayer perceptron, this model implements a ReLU and Sigmoid activator as well as Dropout and BatchNorm1d.

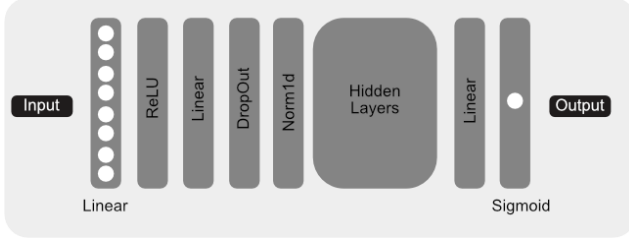


Figure 5 – High-level deep neural network or deep multi-layer perceptron (DeepMLP) layer architecture.

3.2 Model Evaluation

3.2.1 Binary Cross Entropy with Logits

Given that this model aims to predict a binary output, the binary cross entropy loss with logits was set as the criterion for the evaluation, which measures the similarity between the predicted probabilities and the true binary labels while applying a Sigmoid activation function on its output.

$$BCE(y, p) = -[y * \log(p) + (1 - y) * \log(1 - p)]$$

$y \rightarrow$ True binary label, 0 or 1

$p \rightarrow$ Predicted probability of example

$\log(x) \rightarrow$ Natural log of x

3.2.2 Stochastic Gradient Descent

This optimization function updates the model's parameters which comprises the weights and biases based on a random subset of the training data instead of the entire dataset. It seeks to find the set of parameters that minimizes the binary cross entropy function.

$$\theta(t + 1) = \theta(t) - \varepsilon * \nabla J(\theta; x(i), y(i))$$

$\theta(t) \rightarrow$ Model's parameteres at time 't'

$\varepsilon \rightarrow$ Learning rate

$\nabla J(\theta; x(i), y(i))$
 \rightarrow Gradient of loss J to gradients θ

3.2.3 Performance Metrics

To measure with a holistic approach the performance of the models, four metrics are set to facilitate its comparison. Firstly, the loss is computed to quantify the errors between the predicted and actual values utilizing the binary cross entropy function. Secondly, the accuracy is measured as the proportion of correctly classified examples out of the total number of samples.

In addition, precision measures the accuracy of positive predictions calculated as the ratio of true positives to the sum of true positives and false positives. Thus, indicating that the model predictions of a positive outcome are likely to be correct. In contrast, recall measures the ability of the model to capture all the positive instants of the data. Thereby indicating whether the model is effective at identifying positive instances while avoiding false negative errors [8].

A modular approach encapsulation was taken to implement these metric computations asynchronously using threads in the software application.

4. Experiments & Analysis

Based on the high number of potential parameter combinations that can be used within the development of a neural network, it can be argued that this constitutes an NP-hard problem. Hence, experimentation consisted of changing the number of hidden layers, batch size, learning rate, and data normalization across the three proposed models. In this section, the output of this experimentation was analysed to identify the parameters with the most impact on the model performance.

5.1 Hidden Layers

In this test, the multilayer perceptron and deep layer perceptron were tested with 1, 3, and 6 hidden layers. Additionally, the number of nodes within each layer was tested from 16, 32, and 64 nodes.

Overall, it can be seen that the multilayer perceptron with 16-layer nodes (ML16) achieved the lowest loss value of 0.46 when tested with a batch of

5 and an initial learning rate of 0.5 (see Figure 4). Likewise, its accuracy on the validation data seems to fluctuate between 0.4 to 1 when being tested on 300 and 200 iterations, respectively.

By contrast, deep layer perceptron with 32 nodes (DeepMLP32) achieved an accuracy ranging from 0.6 to 0.75 when tested with a batch size of 50 over 300 iterations, Figure 6. Yet, its loss function merely reached approximately 0.5. The deep multilayer perceptron with 64 nodes performed slightly worse, with a loss value ranging from 0.59 to 0.67 and a maximum accuracy of 0.47.

Therefore, it can be argued that although the multilayer perceptron with one hidden layer and 16 nodes reached the lowest loss value, the deep multilayer perceptron performed better on the accuracy front when tested on the testing data set with a batch of 100.

It can be seen that ML16 had a low precision that fluctuated between 0.18 and 0.22, as well as a high recall from 0.75 to 1. Compared to the DeepMLP64 with a precision from 0.82 to 0.88 and a recall from 0.5 to 0.82. Hence, ML16 experiences a situation of low precision and high recall. As a result, it can correctly identify many positive cases and generate many false positive predictions.

In the context of medicine, it can be argued that a high recall has a greater significance than precision. Because most true patients with diabetes would be identified with ML16 and DeepML64, the former would generate higher false positives owing to its low precision [5].

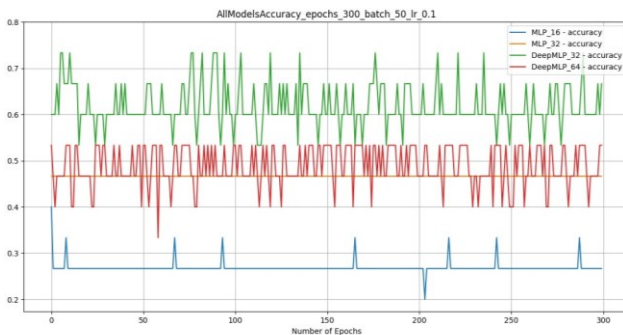


Figure 6 – Models accuracy over 300 epochs, 50 batch size and an initial learning rate of 0.1.

5.2 Dynamic Learning Rate

Also referred to as learning rate scheduling, this technique aims to improve the training process by adapting the learning rate over the epochs iterations. Thus, increasing or decreasing the step size at which the models' weights are updated during the gradient descent optimization [6].

This technique was incorporated into all the proposed models as a custom implementation of the learning rate decay technique. The learning rate decay technique involves gradually decreasing the learning rate when the lowest loss value is encountered by a factor of 0.5. Otherwise, if a better local minimum is not found within a hundred iterations, the learning rate is increased by a factor of 2, Figure 7.

This approach allows the networks to exploit the current local minimum by reducing the step size of the gradient. Likewise, it accelerates search space exploration if the current loss value is not improved over a given number of iterations. The algorithms increased the learning rate on the 100th iteration.

The results of this experiment include achieving faster convergence across all the models within the first hundred iterations. It was found that subsequent iterations did not significantly improve the models' performance despite an increasingly large learning rate.

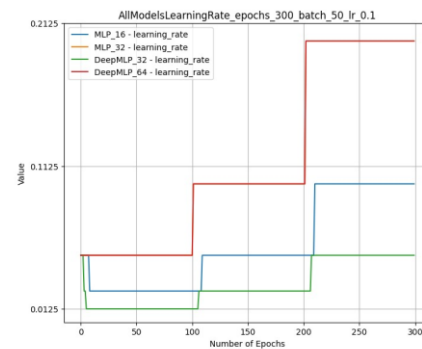


Figure 7 – Dynamic learning rate scheduling over 300 epochs, 50 batch size and 0.1 starting learning rate.

5.3 Data Normalization

This processing technique is used to rescale the values of features in a data set to a standard range or distribution. Thereby improving the performance of the models. The normalisation techniques include Dropout and BatchNorm1d.

In this experiment, Dropout was used to prevent overfitting by reducing the models' reliance on specific neurons during training. This method randomly deactivates neurons in a range of 0.2 to 0.5. This Dropout layer precedes the output layer of the multilayer perceptron.

Moreover, BatchNorm1d applies one-dimensional normalization on the fully connected layers preceding the activation functions. This technique allows the network to learn the optimal scale and shift for each feature by introducing two learnable parameters named Gamma for scaling and Beta for shifting.

Although a significant improvement was not identified with both techniques given the small data set, it can be expected that given a larger data set, their absence will lead to overfitting and poor performance on testing sets.

2. Conclusion

In this report, we conducted extensive experiments to develop neural network models for diabetes prediction. We tackled the challenge of navigating the vast parameter space of neural networks, recognizing its NP-hard nature. Our analysis focused on key parameters like hidden layers, batch size, learning rate, and data normalization across three models. Here are the key findings:

Hidden Layers: We explored the impact of hidden layers and node configurations. The Multilayer Perceptron with 16 nodes (ML16) achieved the lowest loss but showed some accuracy fluctuation. In contrast, the Deep Layer Perceptron

with 64 nodes (DeepMLP64) consistently exhibited strong accuracy on the testing dataset.

Precision and Recall: DeepMLP64 struck a balance between precision and recall, which is crucial in medical contexts. ML16 exhibited high recall but lower precision, indicating its potential to correctly identify positive cases but generate more false positives.

Dynamic Learning Rate: We introduced dynamic learning rate scheduling, which accelerated convergence in initial iterations but did not significantly improve performance in subsequent iterations.

Data Normalization: Techniques like Dropout and BatchNorm1d were explored to enhance model performance. While not transformative in a small dataset, these techniques are expected to be more effective in larger datasets, preventing overfitting.

2.1. Further improvements

Hyperparameter Tuning: Conduct a more exhaustive hyperparameter search to identify optimal configurations for improved model performance.

Larger Datasets: Test the models on larger datasets to better assess the impact of data normalization techniques.

Ensemble Models: Explore ensemble methods to combine the strengths of different models, potentially improving overall performance.

Clinical Validation: Conduct clinical validation to assess the models' real-world applicability and their potential to assist in early diabetes prediction.

In conclusion, while our experiments provide valuable insights, there is room for further optimization and real-world validation of the developed models. The next steps should focus on refining these models for practical application in diabetes prediction and healthcare.

References

- 1] WHO, "Diabetes," 5 April 2023. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/diabetes>. [Accessed 10 September 2023].
- 2] A. M. Egam and S. F. Dinneen, "Diabetes: Basic Facts - What is diabetes?," *Medicine*, vol. 50, no. 10. <https://www.sciencedirect.com/science/article/pii/S1357303922001797>, pp. 615-618, 2022.
- 3] J. A. Luchsinger, "Type 2 Diabetes and Related Conditions in relation to dementia: An opportunity for prevention?," *Journal of Alzheimer's Disease*, vol. 20, no. 11. <https://content.iospress-com.eu1.proxy.openathens.net/download/journal-of-alzheimers-disease/jad091687?id=journal-of-alzheimers-disease%2Fjad091687>, pp. 723-736, 2010.
- 4] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu and L. Shao, "Normalization techniques in training DNNs: Methodology, analysis and application," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 45, no. 8. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10056354>, p. 10173, 2023.
- 5] H. Malmgren , M. Borga and L. Niklasson, Artificial neural networks in medicine and biology, New York: Springer. <https://link.springer.com/book/10.1007/978-1-4471-0513-8>, 2000.
- 6] M. Alssema, D. Vistisen, W. M. Heymans, G. Nijpels, P. Z. Zimmet, P. Z. Shaw and M. Elisson, "Risk scores for predicting type 2 diabetes: using the optimal tool," *PubMed*, no. <https://pubmed.ncbi.nlm.nih.gov/21660635/>, 2011.
- 7] K. Abnoosian, R. Farnoosh and M. Behzadi, "Prediction of diabetes disease using an ensemble of machine learning multi-classifier models," *BMC Bioinformatics*, no. <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-023-05465-z#:~:text=Prediction%20of%20diabetes%20disease%20using%20an%20ensemble%20of,...%204%20Machine%20learning%20models%20Single%20models%20>, pp. 10-94, 2021.
- 8] J. Yang and F. Wang, "Auto-ensemble: An adaptive learning rate scheduling based deep learning model ensembling," *IEEE*, no. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9274468>, 2020.