Neural Networks for Classifying Diabetes

Hamza Alshamy

1 Perceptron Model

The perception has a single input layer, output layer, and no hidden layers or activation functions. As in all analyses, a train/test split (80%/20%) was utilized using `train_test_split`. The model was trained with the `Perceptron` class configured with a maximum of 1000 iterations (`max_iter`), and shuffling of data after each iteration (`shuffle=True`) to improve model robustness. Additionally, the dataset is imbalanced with respect to the outcome variable, diabetes. Specifically, out of 253,680 observations, 218,334 are non-diabetic, and only 35,346 are diabetic. Therefore, the class weight was set to `balanced` to adjust weights inversely proportional to class frequencies, addressing class imbalance in the dataset.

Observe from Table 1 the metrics for the Perceptron model. The Area Under the Receiver Operating Characteristic Curve (AUC) was computed to be 0.648, indicating a moderate discriminative ability. This was complemented by a Matthews Correlation Coefficient (MCC) of 0.16, suggesting a weak correlation between observed and predicted classifications. The accuracy of the model was around 0.61 for both the training and testing sets, indicating modest predictive performance. Lastly, there is a trade-off between precision and recall; the increase in recall to 0.62 came at the cost of a lower precision of 0.2. However, since the question at hand is of medical importance, flagging diabetics as such is of high importance.

Model	**AUC**	**Training Accuracy**	**Test Accuracy**	**MCC**	**Precision (Diab.)**	**Recall (Diab.)**
Perceptron	0.648	0.61	0.609	0.16	0.20	0.62

Table 1: Perceptron Model Performance Metrics

Given the low-moderate AUC score, below-guessing capabilities as shown by accuracy, and low MCC, the model does not perform well. Also, for visual representation of the AUC score, refer to Figure 1 below.

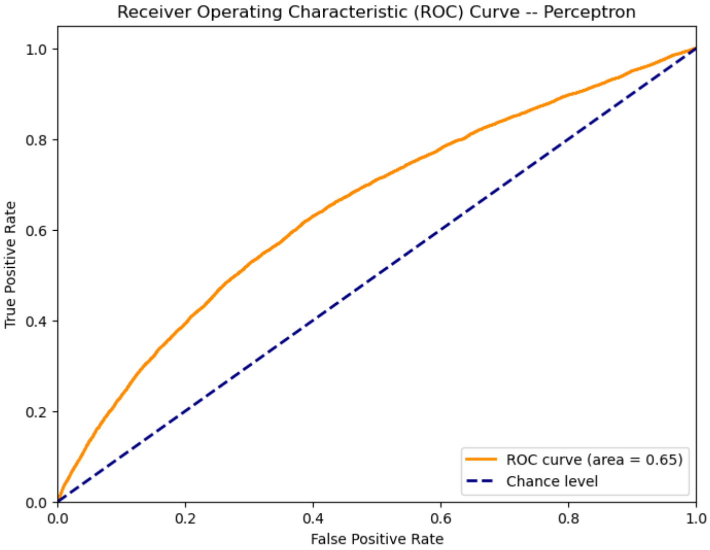

Figure 1: Receiver Operating Characteristic (ROC) – Perceptron

Lastly, you can refer to the Confusion Matrices in section 7 to view the confusion matrix produced by this model.

2 Feedforward Neural Network (DV: Diabetes)

The feedforward neural network FFNN was iteratively constructed with varying configurations, specifically altering the number of hidden layers and the type of activation functions used. Each model configuration was trained using a train/test split (80%/20%). The activation functions tested were ReLU (Rectified Linear Unit), Sigmoid, and no activation function. The complexity of the models was adjusted by varying the number of hidden layers from one to five. This approach allowed for the exploration of model performance across a spectrum of depth and non-linear processing capabilities. Since we are using the neural network for binary classification, training the model involved optimizing a binary cross-entropy loss function with a Stochastic Gradient Descent (SGD) optimizer, set at a learning rate of 0.01, across 100 epochs. The learning rate of 0.01 was used to follow the convention.

The 15 models were evaluated based on AUC, MCC, and accuracy metrics for both training and testing phases which can be fully seen in Table 4 in the Appendix 7 section. Figure 2 below shows the AUC score for all 15 models.

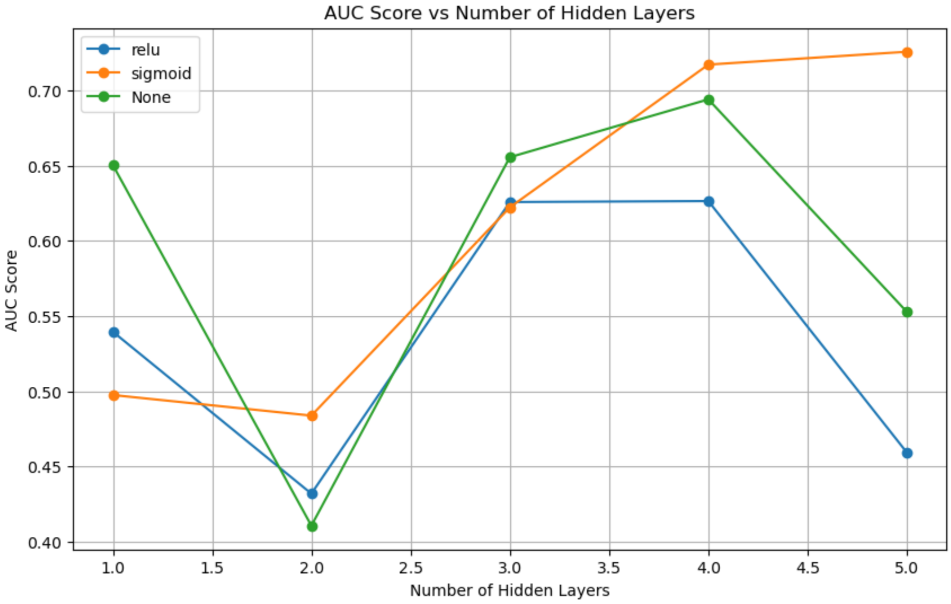

Figure 2: Receiver Operating Characteristic (ROC – Different Activation Functions for a Range of Hidden Layers)

Observe from Figure 2 that the AUC score varies with the number of hidden layers and is influenced by the activation function. For example, models utilizing the ReLU activation function generally performed better in terms of AUC compared to those using the Sigmoid function or no activation function, particularly as the number of hidden layers increased (up to a certain point). The highest AUC observed was 0.725, achieved by the model with five hidden layers using Sigmoid activation. All the MCC values were near zero (refer to Table 4) suggesting that the models do not account for the class imbalance well. However, the similarity of the training and test accuracy suggests the models are not overfitting.

Comparing the AUC score from the Perceptron model of 0.65 from Question 1 to the highest AUC score of 0.725 suggests that an FFNN model performs better. *However*, it is important to note, that the MCC for all the models in this analysis is lower – the imbalance is not taken into account. Therefore, only considering the AUC score as a metric for comparison can only show half the story.

3 Deep Neural Network

For a deep neural network, three hidden layers were used. This network consists of sequential layers with 128, 64, and 32 neurons respectively, each followed by ReLU activation functions to introduce non-linearity, enhancing the model's ability to learn complex patterns in the data.

Table 2 below shows some metrics to assess the deep network. The model's AUC score of 0.826 indicates a discriminative ability between diabetic and non-diabetic classes. Moreover, the similarity between the training and test accuracy suggests that the model is not overfitting. The AUC and training/testing accuracy are encouraging, however, upon looking at the MCC, precision, and recall, it seems that the model is prioritizing minimizing false positives over identifying positive cases. Since the question at hand is medical, an emphasis on identifying positive cases should be prioritized. This conflicting story between the AUC, MCC, and recall implies that the model is not taking the case imbalance into account.

Model	AUC	Training Accuracy	Test Accuracy	MCC	Precision (Diab.)	Recall (Diab.)
Deep Network	0.826	0.863	0.864	0.128	0.66	0.03

Table 2: Deep Network Performance Metrics for Question 3

For visualization, you can refer to the AUC below in Figure 3 or the confusion matrix in the confusion matrices section 7.

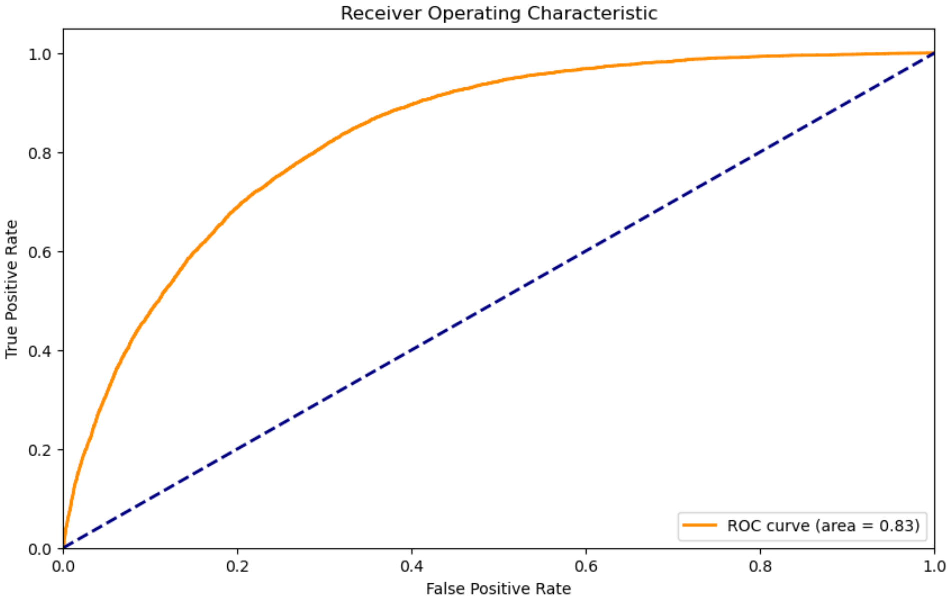

Figure 3: Receiver Operating Characteristic (ROC)

3.1 The Use of Convolutional Neural Networks (CNNs)

Given the nature of the dataset, which consists of structured and non-spatial features such as diabetes status, BMI, and socio-economic indicators (income bracket), it would be unconventional to use Convolutional Neural Networks (CNNs) as they are predominantly used in scenarios involving spatial data. CNNs can be used with data that exhibit spatial or temporal relationships, such as images or videos where pooling layers can effectively capture hierarchical patterns. Therefore, for the given dataset, utilizing CNN would likely introduce unnecessary complexity without enhancing the model's performance.

4 Feedforward Neural Network (DV: BMI)

To explore the relationship between activation functions and model accuracy in predicting BMI, a feedforward neural network with a single hidden layer was developed. The network architecture was designed with one input layer containing as many neurons as there are features in the dataset, a hidden layer with 128 neurons, and a single neuron in the output layer to predict BMI values. The model utilized three different activation functions: ReLU, Sigmoid, and none, to determine their effect on RMSE.

The data preprocessing included standardizing feature inputs and BMI (the outcome) values using `StandardScaler` to ensure that the network's input and output scales did not bias the training process. The network was trained using the Adam optimizer with a learning rate set to 0.01 across 100 epochs, minimizing the mean squared error between the predicted and actual BMI.

Figure 4 below, which shows the RMSE values across different activation functions, shows that the ReLU activation function resulted in the lowest RMSE of 0.905, indicating a more accurate BMI prediction compared to Sigmoid and no activation function.

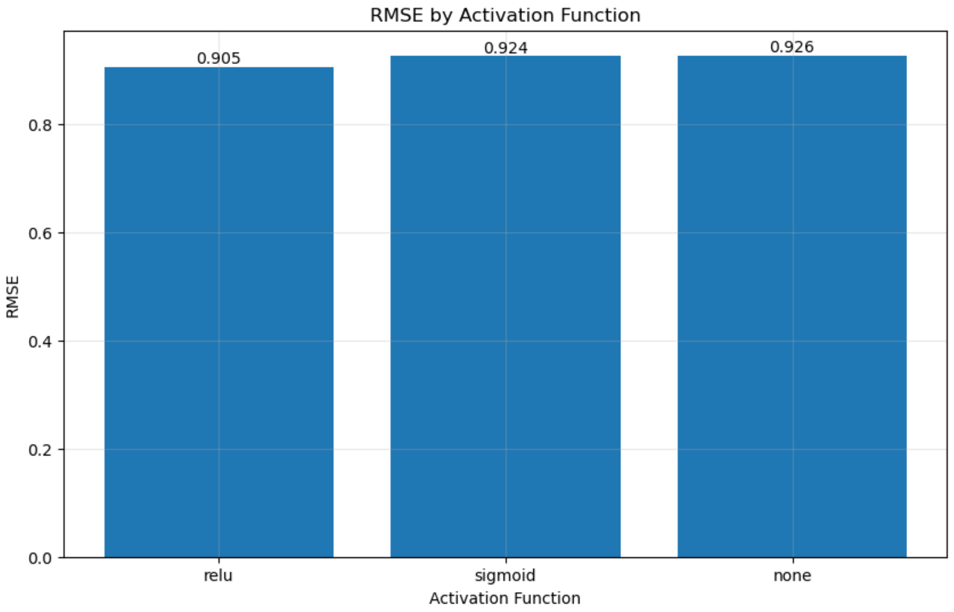

Figure 4: Loss Curve

Observe that the RMSE in the models with an activation function performs better than the model with no activation function. However, the difference is minimal. Therefore, while the RMSE is affected by which activation function is used, the impact is not substantial for this particular dataset and task.

5 Predicting BMI

To predict BMI from the rest of the dataset, a series of neural network models were developed. Each model was differentiated by the number of hidden layers, ranging from one to eight, and the type of activation function employed: none, sigmoid, ReLU, and softplus. Each model's performance was evaluated using Root Mean Squared Error (RMSE), optimized using the Adam optimizer, and trained under the Mean Squared Error (MSE) loss criterion.

Using 4 different activation functions across 8 hidden layers produced 32 models. The RMSE for all the models can be seen in Figure 6 in the Appendix 7 section. Figure 5 below shows the RMSE for all models.

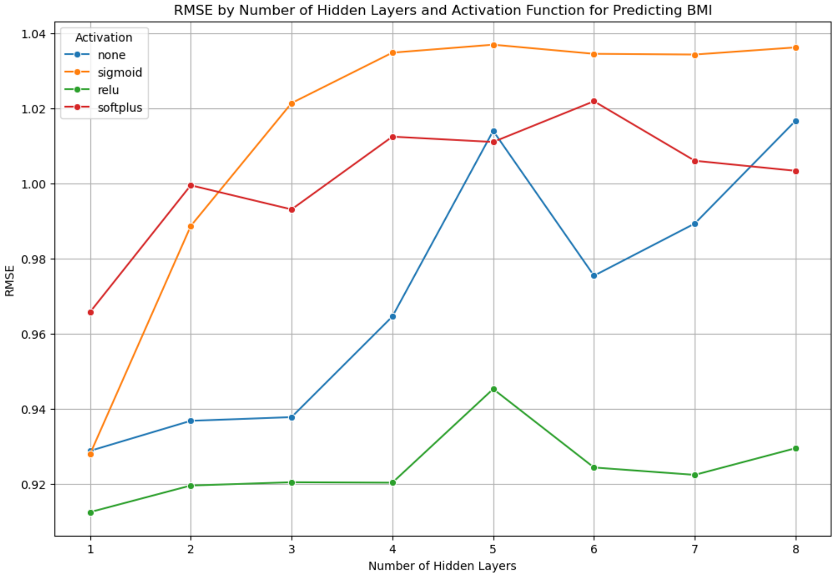

Figure 5: RMSE by Number of Hidden Layers and Activation Function for Predicting BMI)

The models with ReLU activation function consistently showed superior performance, with the lowest RMSE recorded at 0.913 for a single hidden layer configuration. As the number of layers increased, ReLU models maintained a relatively low RMSE, suggesting robustness in simpler configurations. Conversely, models without an activation function or with the softplus function exhibited higher RMSEs, particularly as more layers were added, indicating potential overfitting or inefficiencies in learning from complex models. Sigmoid functions demonstrated an increasing RMSE with more layers, possibly due to vanishing gradient issues

Generally, it can be seen that the RMSE is influenced by the activation function as well as the number of hidden layers.

6 Comparing Neural Networks to Classical Methods

To compare some of the classical methods (Logistic Regression, SVM, and tree-based methods) with Neural Networks, Table 3, below, presents the same metrics on the same dataset for classifying diabetes[1].

Model	**AUC**	**Training Accuracy**	**Test Accuracy**	**MCC**	**Precision (Diab.)**	**Recall (Diab.)**
Logistic	0.8287	0.7311	0.7303	0.363	0.31	0.78
SVM	0.8275	0.7272	0.7265	0.361	0.31	0.78
Decision Tree	0.7574	0.7632	0.7639	0.316	0.32	0.62
Random Forest	0.8255	0.7560	0.7529	0.367	0.33	0.74
AdaBoost	0.7919	0.7543	0.7558	0.345	0.32	0.69
Perceptron	0.648	0.61	0.609	0.16	0.20	0.62
Deep Network	0.826	0.863	0.864	0.128	0.66	0.03

Table 3: Models Performance Metrics

It is quickly apparent that the MCC for the classical methods are higher which means they take the class imbalance into account. Since the question at hand is medical, the stakes are high and the ability to distinguish classes and flag positive cases should be prioritized. Further, recall for the diabetes class is also higher (at the price of lower precision) which means that the classical methods do a better job of flagging diabetics. The AUC for the deep network can be compared to the classical methods, however, the classical methods excel in all other aspects. One significant advantage of neural networks is their scalability and adaptability to various types of data, which might not be fully leveraged in this traditional tabular dataset. There are no necessary visualizations here as all the metrics can be seen in the table above.

[1] The metrics for the classical methods were obtained from a previous analysis which can be found on my GitHub account.

7 Appendix

Activation Function	Hidden Layers	AUC	MCC	Training Accuracy	Testing Accuracy
relu	1	0.5396	0.0000	0.8603	0.8621
relu	2	0.4320	0.0000	0.8603	0.8621
relu	3	0.6258	0.0000	0.8603	0.8621
relu	4	0.6265	0.0000	0.8603	0.8621
relu	5	0.4593	0.0000	0.8603	0.8621
sigmoid	1	0.4975	0.0000	0.8603	0.8621
sigmoid	2	0.4838	0.0000	0.8603	0.8621
sigmoid	3	0.6223	0.0000	0.8603	0.8621
sigmoid	4	0.7172	0.0000	0.8603	0.8621
sigmoid	5	0.7258	0.0000	0.8603	0.8621
None	1	0.6504	0.0376	0.8602	0.8616
None	2	0.4107	0.0000	0.8603	0.8621
None	3	0.6556	0.0000	0.8603	0.8621
None	4	0.6941	0.0000	0.8603	0.8621
None	5	0.5528	0.0000	0.8603	0.8621

Table 4: Feedforward Neural Network Performance by Activation Function and Number of Hidden Layers (Question 2)

Hidden Layers	RMSE
1	0.928825
2	0.936835
3	0.937816
4	0.964587
5	1.013986
6	0.975445
7	0.989326
8	1.016687

(a) Activation Function: None

Hidden Layers	RMSE
1	0.927909
2	0.988669
3	1.021380
4	1.034796
5	1.036919
6	1.034486
7	1.034297
8	1.036198

(b) Activation Function: Sigmoid

Hidden Layers	RMSE
1	0.912529
2	0.919616
3	0.920492
4	0.920375
5	0.945303
6	0.924411
7	0.922448
8	0.929546

(c) Actication Function: ReLU

Hidden Layers	RMSE
1	0.965736
2	0.999508
3	0.993089
4	1.012447
5	1.011032
6	1.021867
7	1.006034
8	1.003357

(d) Activation Function: Softplus

Figure 6: Comparison of RMSE by Activation Function and Number of Hidden Layers

Confusion Matrices

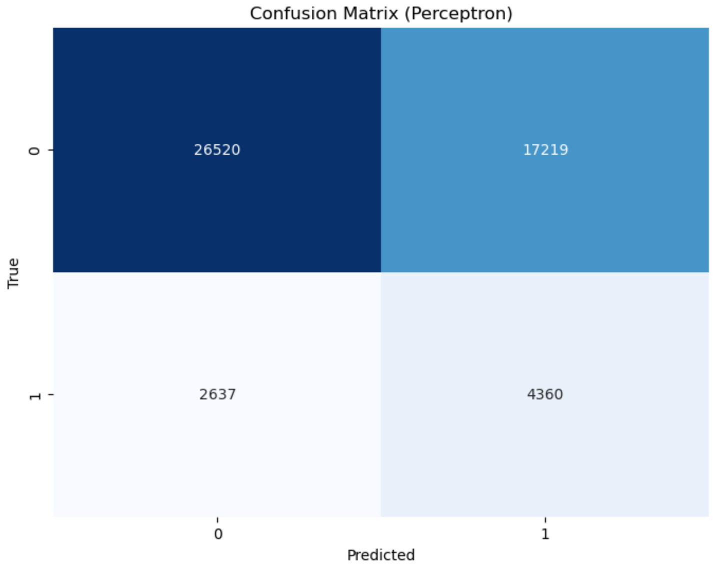

Figure 7: Confusion Matrix (Perceptron)

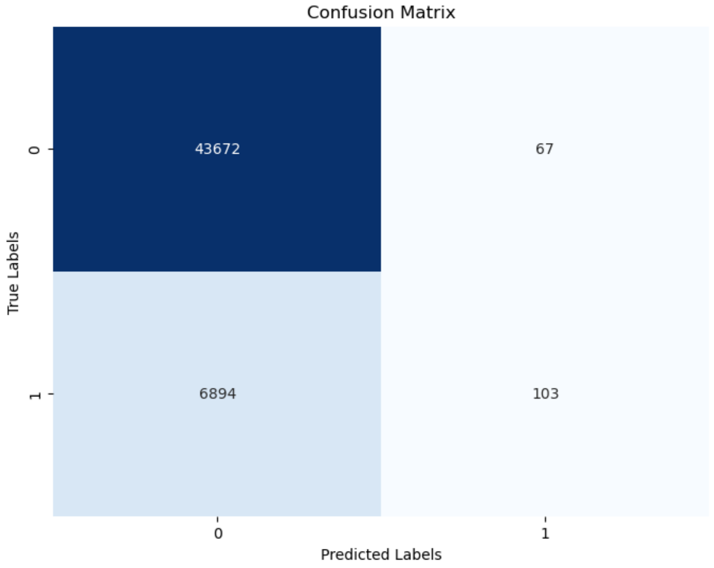

Figure 8: Confusion Matrix (Deep Neural Network)