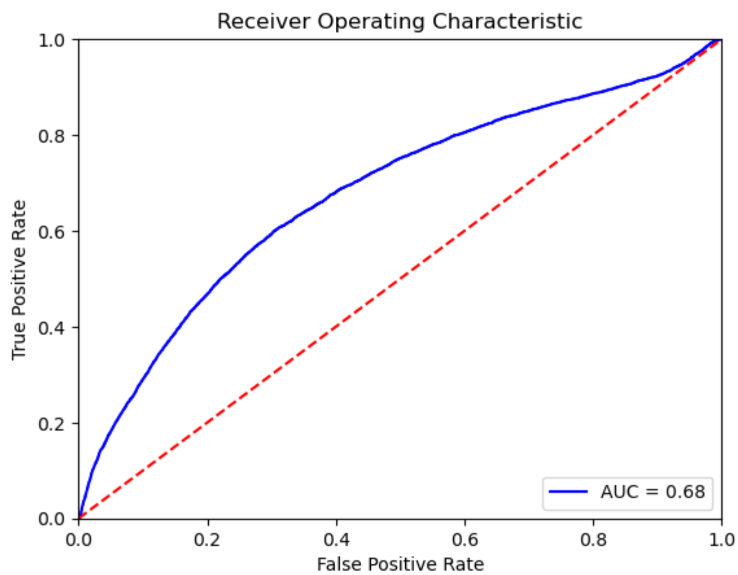
FML HW4

1. I trained one perceptron with one input layer, one output layer, no hidden layers and no activation functions. I normalized the training features (every column except Diabetes in the csv file). I then calculated the AUROC and drew the ROC graph.

The perceptron is the simplest neural network model. Normalizing improves convergence speed and balances the impact of different features. A higher AUROC usually means better performance.

AUROC is 0.678476701371747.

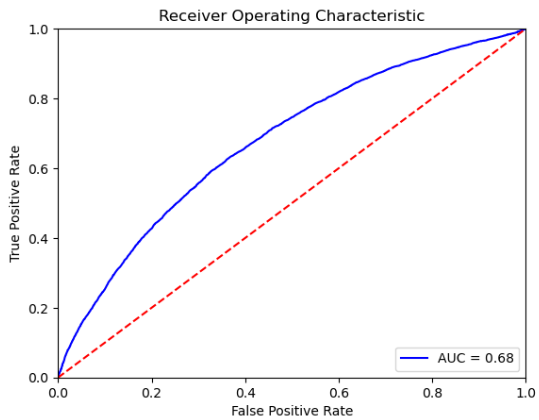


The AUROC is neither too low nor satisfyingly high. Perceptron is better than pure guessing.

2. I built two neural networks, one with a single hidden layer and the other with three hidden layers (the case with 2 hidden layers is reserved for question 3, though I don't quite understand why the HW is designed that way). Each has three versions with different activation functions: no activation functions, ReLU, and sigmoid function. So there are 6 AUROCs in total.

This allows me to explore the relationship between AUROC and the number of layers / activation functions used.

AUROC	No Activation Function	ReLU	Sigmoid
N_layers=1	0.81	0.80	0.79
N_layers=3	0.80	0.74	0.52

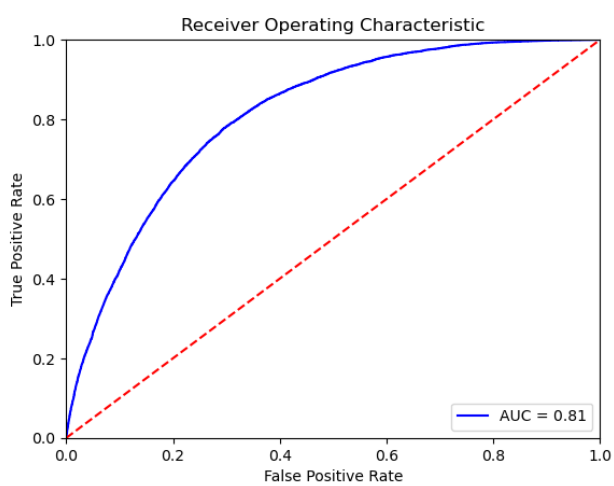


It seems that three hidden layers together don't outperform one hidden layer, possibly due to overfitting. Sigmoid function is somewhat weaker compared to no activation function or ReLU.

3. As discussed in the last question, now I build a neural network with two layers and one of three different activation functions.

CNN is mostly used for classification tasks involving 2D images. It uses kernels to take advantage of the fact that adjacent pixels in images are usually highly correlated. These kernels can reduce the size of input data and improve efficiency. But here each feature is a 1D column and 2D kernels do not work here. There are 1D kernels, but it is not a very common approach to use them for the data at hand.

The three AUROC values are 0.81, 0.78 and 0.68 for no activation function, ReLU and Sigmoid.



Two hidden layers' performance is between that of one hidden layer and three hidden layers, further strengthening the argument for potential overfitting. It seems that the original data contains some random errors/noises that might be captured by overly complicated models.

4. Now the training features are all columns including Diabetes but excluding BMI. A neural network with one hidden layer is built. It uses the Sigmoid activation function. It has one output neuron in the output layer. In each training epoch the training data are fed to the model in batches and the optimizer adjusts the weights of connections after each batch. Total squared error of each batch are added together and divided by the number of samples to get the RMSE.

The number of samples in the csv file is so large that there is not enough memory to calculate the RMSE of all prediction results at once, hence we use the batches. This is a regression task so only one neuron is required in the output layer.

RMSE is 106.49.

The RMSE is unreasonably high for this model. Some adjustments are probably required. Will look further into the issue.

5. Now use a deep neural network with 2 hidden layers and ReLU activation functions. Batch operations are still applied.

RMSE is 106.43, which shows that it is probably not the number of layers that matter in this problem.