

In this project, I have tried to predict whether a person has diabetes or not based on **8 parameters** such as **pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, pedigree function and age**. Non-diabetic and diabetic people are predicted as 0 and 1 respectively.

First, I split the dataset into train and test sets. The training set is used to build the model and the test set is used to evaluate the model that I built. The split was made to have **80** percent training samples and **20** percent testing samples. Then, I **one-hot encoded** the target variables so that the neural network can output the probabilities for both classes.

The neural network that I used has the following architecture:

- 1) **3 layers**, out of two are hidden layers and one is the output layer which outputs the probability of a sample to belong to each class.
- 2) The first hidden layer has **500** neurons each and I used "**sigmoid**" activation function in this layer whereas in the second hidden layer I have used **100** neurons with "**sigmoid**" activation function again.
- 3) In the output layer, I used "**softmax**" function to output the probability of a sample to belong to each class. This layer has **2** neurons since there are two classes that need to be predicted.

I used two additional hyper-parameters:

- 1) **Learning rate** of **0.001** which helps us to optimize our cost function, and
- 2) **Batch size**, which controls how many training samples are going in each iteration.

RESULTS:

Using the above architecture and hyper-parameters, I was able to achieve an accuracy of **0.72** on the test set.

NOTE:

Anyone running this code has the option to try different combination of hyper-parameters leading to different neural network architecture which would produce different accuracy results on the test set.