Project Documentation on Obtaining Best Decision surface

Major Steps:

- Import required libraries
- Load the dataset given
- Identify the relation (surface)
- Segregate the data
- Split the data
- Preprocess the data
- Building the Architecture
- Compiling
- Training the model
- Prediction
- Plotting the Best Decision Surface

Libraries to be imported:

1. Import Numpy
2. Import pandas
3. Import seaborn & Matplotlib
4. Import tensor flow
5. Import keras
6. Import sklearn

Identify the surface:

- Use Matplotlib and seaborn to identify the surface that we need to plot (Hint: use scatter plot).Surface: Linearsep.

- Perform little bit of EDA to understand the data (example: the distribution).

Data Segregation:

- Separate the input features and the target variable
- We have two input features of continuous and target variable as discrete (binary classification).

Split the data:

- Split the data into train and test (to be precise: Xtrain, Xtest, ytrain, ytest) using sklearn.
- While splitting the data provide either test_size or train_size.
- Test_size = 0.2

Data Preprocessing:

- Use data preprocessing techniques like Standardization and normalization to preprocess yourdata to common scale (to a specific range). I have used Standardization.
- Use sklearn library to preprocess your data. While performing preprocessing the data make sureto use fit_transform on training data and transform on test data (cause we don't our object to learn from unseen data).

➢ Building the Architecture:

- Step1: create a sequential model using the Class "Sequential" from keras module.
- Step2: Add layers to your sequential model. While adding the layers make sure to use 'Dense' layers, the optimal number of neurons, use appropriate activation function and pass the inputsonly for the first dense layer.
- Use the same activation function in the hidden layers and in the output layer use sigmoid/tanh(care must be taken while choosing).
- You can also add dropout's to your Architecture if your resultant model is over-fitting (by usingdropout's we can dismantle some percentage of the connection's).
- You can also set kernel initialize after appropriate analysis of your data.
- I have used 2 *hidden* layers and one output layer, Tanh activation function in hidden layers and *sigmoid* activation function in output layer.

➢ Compiling the architecture:

- Use the compile method available in keras to compile your sequential model.
- While compiling specify the optimizer, the loss, metric.
- Do the trial and error method to obtain the best optimizer for now!
- Optimizer that I have used is 'Adam'; loss is 'binary_crossentropy'; and metric is 'Accuracy'.

➢ Model Training:

- Fit the sequential model that you have created by simply saying 'model.fit'; you should fit yourmodel on the training data not the unseen data.
- Train your model using the transformed input data and the target variable, specify the numberof epochs (hint: try). I have used a total number of epochs as **160**.
- While training you can also provide the validation set of data which will further be split form your training data for the validation score to know whether your model is over-fitting or notjust by comparing train score and validation score.
- My train score turned out to be 80% and validation score 82%

➢ Prediction & Evaluation:

- Use the trained model to predict on the unseen data
- Evaluate your model using the appropriate evaluation technique on the actual and thepredicted values.
- I have predicted and evaluated the model's performance, got a score of 80% on the unseendata.
- There is no overfiting as the train score and the test score almost equal which proves that mymodel has learned well from the train data and predicted well on the unseen data.
- As we know for a best fit model the train and test score should be as high as possible.

Plot the best decision surface:

- Using mlxtend import plot_decision_regions and plot the best decision surface using yourmodel.
- To obtain the best decision surface repeat the processes multiple number of times (play withcode).

Conclusion:

- I was success in obtaining the best decision surface for the given data.