Deep Learning Assignment-1 Summary by Ammar

- Read about the problem

- import libraries

- import dataset

- Try to undetstand the dataset

- Check Null values

- Check duplicate observations

- Check dupliocate CustmerId OR RowNumber since they should be unique

- Drop:

- CustmerID

- RowNumber

- Surname, surname may worth a trial since maybe some members of a family follow have the same behaviour

- Convert some cateogrical variables to their proper type since they were exist as numerical

- See the distriubtion of our target classes. Two classes but 80:20 (imbalance)

- See the relationship between each couple of the variables using pairplot, and using different color for each class

- It seems that there are no unique areas for class 1 (most parts are also covered by class 0)

- No Association between each pair of variables

- Check correlation between each 2 variables

- No strong linear correlation between any pair of variables (including the target)

- For Cateogrical features, bar plots were done

- The most useful insight from these barplots is that when NumofProduct is 3 or 4, the closed account are more than the open

- We Check the box plots of the numeircal variables for each class

- EstimatedSalary will not help much alone to distinguish between the two calsses

- CreditScore in a smaller degree also seems will not help that much

- Age have many outlier (old people) which does not seem error values, maybe those people opened the account than maybe even forget to use it! etc. Therefore, we will keep them but will use Robust scaler instead of MinMAx or standard scaler

- Data PreProcessing:

- We split hte data into train & test iwth stratfying our target since it is imbalanced

- We create a pipline that do one-hot encoding for the two cateogrical variables (Gender, Geography) & Robust scaling which is used due to the high number of extreme values in age

- We Fit\_transform the trianing and just transform the testing to prevernt data leakage

- Build models (11 models were built): (The model to not be useless, it should give accuracy higher than predciting the most frequnet class which is around 80%)

- 1- Without class weight

- 2- With class weight (no tuining just use the method used in Tensorflow documentation https://www.tensorflow.org/tutorials/structured\_data/imbalanced\_data) --> No big differnece

- 3- with Class wieght but With higher learning rate (0.01) --> worse reuslt and very oscilating loss

- 4- Add dropout to preventr overfitting and with default learning rate again --> Similar train/testing score. However, underfitting situation

- 5- Add early stopping to prevent overfitting and with default learning rate again --> IT stooped after the 45th epoch and no need for comleting the training until the 1000th epoch

- 6- Change themonitoring from loss\_val to val\_recall

- We have to change the metric from accuracy to recall as well

- The aim is still to minimize the loss function but now we monitor the recall isntead of the accuracy

- The eraly stopping now is based on recall's validaiton improvement

- The prevous models were static (pre-defined layers and neurons)

- Now will emntion the models that uses Hypaprameter tuning for:

- Number of hidden layers

- Number of neurons of rach

- Dropout rate for each

- The optimizer

- The learning rate

- 7- Accuracy --> Accuracy(85%)

- 8- Accuracy with undersmapling --> Accuracy(84%) (undersampling is not suggested)

- 9- Recall --> Accuracy (24%) --> It almost predicted everyting as class 1

- 10 - PrecisionAtRecall we were looking for the best percision when recall is >=0.7 --> Accuracy (87%)

- The weird thign is that the recall were less than 70%, maybe it do that per mini-batch

- 11- SensitivityAtSpecificity we were looking for the best percision when Specificity is >=0.7 --> Accuracy (87%)

- For the final model, they were mutliple models with accuracy is 87% which is > 80% (the null accuracy), I'll choose the PrecisionAtRecall model since it is easy to interpret (similar to F1)

- We save the model and scaler to be applied during the inference

- We selected a random point and it was misclassfied, which confrims that out model btter than the null accuracy but stil need further improvements, by applyting imbalanced-data techniques like oversampling for the minroity class