Kiana Koohjani, Sahar Mohammadi

In this project we aim to make a classification model and train it for a dataset containing signals of brain when its normal and when a seizure is happening for the patient.

Preparing dataset

In 10 edf files including chb01\_03, chb01\_04, chb01\_15, chb01\_16, chb01\_18, chb01\_21, chb01\_26, chb02\_16, chb02\_16+ , chb02\_19 there are some parts that a seizure exist.

Each file has 23 signals but we only want signals labeled fz-cz and cz-pz which is numbered 16th and 17th. So, we read all the files and get the part from the moment a seizure starts until it ends. For example, in file chb01\_03 seizure starts at 2996 and ends at 3036. And the sampling frequency is 256, we get the cell array indexed 2996\*256 till 3036\*256 of these two signals.

Our data is slides of 5 seconds and for getting a better result in the network, we increase the slider index only 1 and we have an overlap of 4 seconds.

It gives us 564 seizure cells, so we need 564 normal cells. Since we want to peak 25% of normal data from a file that includes seizure, we need 141 normal cell from seizure files. And 423 from normal files. To prevent noisy data we get rid of the first 5% of the file.

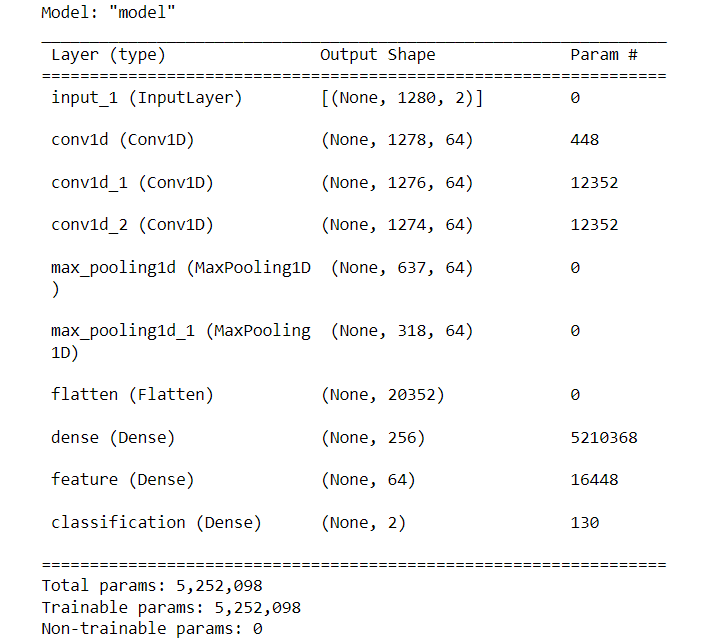
This will make our X and for Y or labels we make np arrays of 0 shaped 564\*1, and of 1 the shape, and then we concatenate them.

In order to classify the data, we first make a CNN model and train it with the data we have.

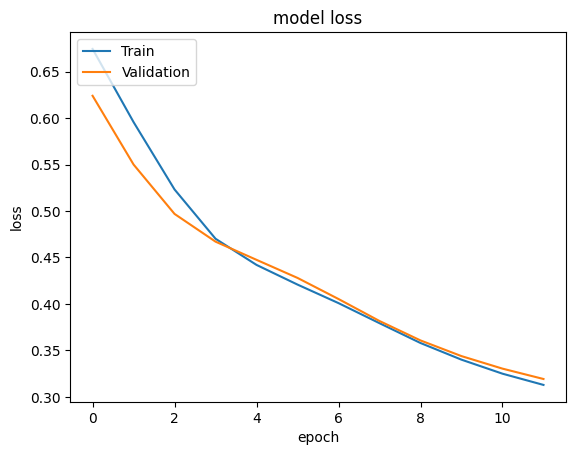
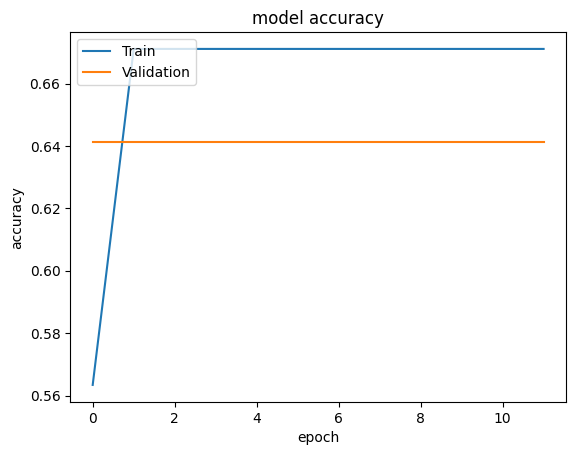
We built a model that consists of three convolutional layers, two max pooling layer, 1 flatten and two dense layers.

To check whether this model is best for our data or not, we built models with different layers as well but they all resulted in a worse outcome.

Summary of the CNN model:



Plots:



After training our model we discarded its last classification layer and extracted 64 features.

Alongside extracting features from this CNN model we calculate mode, median, mean, variance, standard deviation, kurtosis, coefficient of variation, skewness, min, max, PTP, TWOPP, PPS, shanon and reyni for the signals. We choose only some of this features by adding them to an array one by one and checking if the feature that has been added classifies more signals correctly and if it does we add that feature to the list of features that we select for our signal.

In the end we use random forest classification and the features that we selected to classify our signals.

After the final classification, we check whether its working good or not by drawing confusion matrix, calculating accuracy, precision, recall and f1.

The result:

average accuracy: 0.9988200589970502

average precision: 0.9982532751091704

average recall 1.0

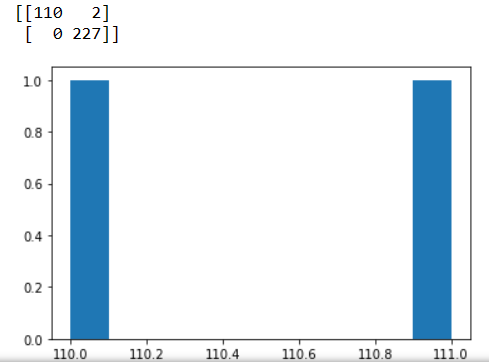
average F1 0.9991228070175439

The time of execution of above program is : 2304000 ms

We also calculate the false alarm and its rate, times when our model predicted a false positive (it wasn’t positive truly) and how long it continues this alarm.

Which only once it showed false alarm and the duration was 2 slide (10 seconds, two time repeatedly predicted false positive).

Confusion matrix and the histogram of false alarm:



So, if we want use our model to release a drug if there is seizure happening and the drug has side effect so we want to make sure that the seizure is really happening it should wait for 10 seconds to make sure that it’s not a false alarm.