**Machine Learning**

1. **Computational learning experiments**

**Featurer engineering**

We first did exam what the data look like. Our training data consists of three files, *written\_train.npy, spoken\_train.npy* and *match\_train.npy. written\_train.npy* is with a dimension of 45000x784. It contains 45000 images. As described in the assignment, ‘each image is given as 784-dimensional vector, which represents 28x28 pixel grayscale image. Pixel intensities range from 0 (black) to 255 (white)’. *spoken\_train.npy* is a 3-dimensional dataset. There are 45000 pieces of audios in it. These are spoken words. These audios are given as an array of MFCC features. Each piece of audio has a shape of Nx13, and 13 is the number of MFCC feature. N varies because of different length of each audio piece. We found out that the maximum number of N is 93. Hence, for these audio pieces with dimension, N, less than 93, we added zeros after the last ‘frame of speech’. Then we scaled *written\_train.npy* and *spoken\_train.npy* separately. The reason we scaled two datasets separately is because the features in numerical values of these two datasets are with different scales. We tried different scaling as well. It includes MinMax scaling, standard scaling, MaxAbsScaler and Normalized scaling. After the scaling is done, we concatenate *written\_train.npy*, *spoken*\_*train*.*npy* and *match\_train.npy* into one dataset. *written\_train.npy* and spoken\_train.npy are X\_train and *match\_train.npy* is y\_train.

**Learning algorithm(s) tried**

The algorithms that we tried are Perceptron, Multi Layer Perceptron (MLP), convolutional neural network (CNN) and Long short-term memory (LSTM). All algorithms are combined with over-sampling and with out over-sampling. In general, training algorithm with over-sampling gives higher accuracy score. Dataset are split into training set and test set. We use training set to train different algorithms with various parameter settings. Hyper-parameters are tuned on the test set.

**Parameter setting and tuning**

For Perceptron and CNN, JANNE ADDS HER ART…

For MLP, the number of hidden layers that are experiments is 2 and 3. The number of neurons is 100 with bias. For eath hidden layer, a drop out rate of 0.2 are used to avoid over-fitting. Optimizers are sgd and adam. Learning rate are 0.01 and 0.001. The weights of neurons are initial-ized by random numbers generated by a normal distribution with 0 mean and 0.05 standard deviation. Train data set are scaled first to have a better result for MLP. For each hidden layer and output layer, there is a bias. For hidden layers, the activation function is relu and for the output layer, the activation function is sigmoid. Epoch ranges from 20 to 100. Batch size is 5 and 10. The loss function is binary cross entropy because only two classes are predicted.

For CNN AND LSTM, EMILE ADDS HER ART…

**Discussion of performance**

Three submission were made to CodaLab. The first submission resulted in an accuracy of approximately 20%. Upon receiving this result, the result of MLP was uploaded to Codalab and the accuracy was around 90%. Then great efforts were put into improving the CNN model by the author, by tuning the parameters of the model further. Unfortunately, in spite of the efforts made by the author, the third submission to codalab only resulted in an accuracy of approximately 92% (slightly higher than the baseline of approximately 91%). Initially, the goal was to make as many submissions as possible to CodaLab. However, due to limited computational resources of the authors (possible due to the advanced age of the computers used for this assignment) and the limited time available, it was unfortunately not possible to get the accuracy rate higher.

Given more time and more resources, several approaches could have been taken to improve the submission results. Other validation schemes such as k-fold cross validation and “leave-one-out” could have been utilized to avoid overfitting, given enough time and resources. Furthermore, better hyper-parameter tuning might give better predictions, resulting higher accuracy score. Perhaps, simpler classification approaches such as k-nearest neighbors would also have resulted in a better accuracy at less computational and time costs.

In conclusion, this assignment has been a great learning opportunity in terms of how to (and how not to) approach a classification problem. Although, the authors are disappointed with accuracy generated from the results submitted to CodaLab, the challenge was very appreciated.

1. **Work done by group members**

* **Everyone**: Every team member was involved in creating a project roadmap, including discussion on how to pre-process data, individual coding on pre-processing, work division, and internal deadlines.
* Emile Jaspar: was responsible for CNN and LSTM.
* Janne: was responsible for Percetron and CNN.
* Sen Yang: was responsible for MLP, submitting the result to Codalab and writing the report.

1. **CodaLab account details**

Account: baobaodoudoudavid@hotmail.com