Mohammad Shahriar Hooshmand
Ph.D. Candidate
Department of Materials Science and Engineering
The Ohio State University

Note: Please note that the figures are captured with high resolution. Please zoom in to read the axes values.

a) In order to vectorize the data, we have two approaches: integer coding vs one hot coding. Typically, if the features do not have ordinal relation, it is better to divide them to separate features with the assigned Boolean values known as one hot coding. For example, for *mood* there is no such ordinal relation between *silly, happy* and *tired* so it's better to separate them.

   If there is an ordinal number, in some algorithms it's convenient to assign integer 1,2,3,… to the attributes. This continuum of numbers gives an order to the attributes where it makes difference which attributes happen before/after one another. While there is no restrict rule about it, it's always important to check how the separating affects the results. It has been shown that if we group the features that have ordinal relation (such as time instead of 7am,8am,9am,… as separate attributes, we group it to morning, afternoon, evening), this might affect the results (depending the algorithm) where one hot coding might give better results compared to integer coding even though there is ordinal relation.

   Our approach here is to start by hot encoding the *mood* and integer coding the *dayofweek,timeofday,timetoplay* and finally assigning the Boolean 0 or 1 to yes/no features. Then, we try different combinations of the first three features by hard coding and integer coding them and see what model gives the best accuracy. Details are explained in the next section.

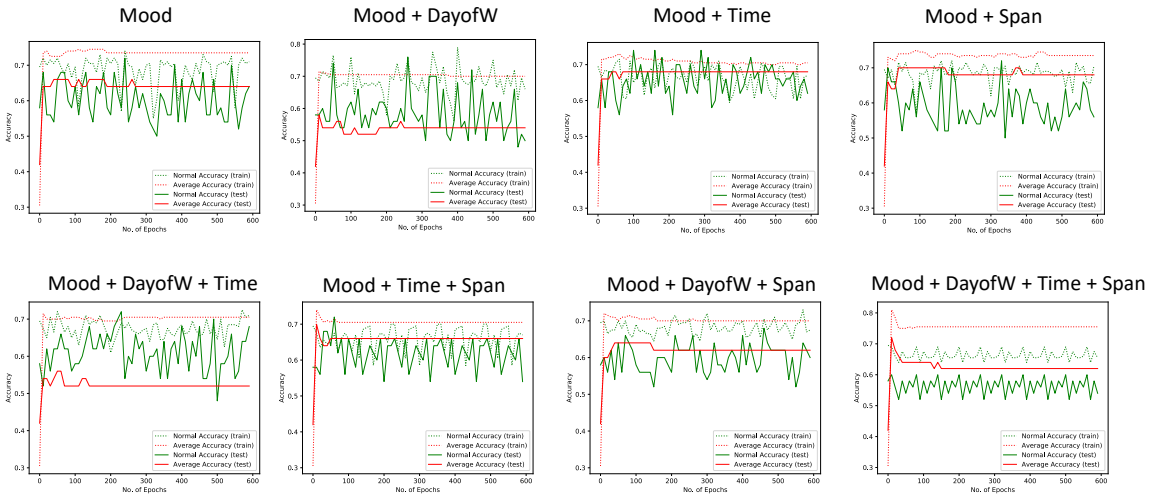
b) The code is implemented, and the results are as follows. For comparing the accuracies, the range of epochs considered up to 600. It is important to note that one might say one hot coding of possible non Boolean features (here the first four features) might always put us on the safe side an is better, however, this leads to the higher model complexity and prone to the overfitting. Here, we don't have a large dataset so we might not observe overfitting, however the best possible way to assure which features to categorize is to systematically study it.

   As we see, one hot coding of (mood, timeToplay:span) and (mood,timeofDay:time) gives comparatively better results on the test set. Separating all the first four features give the most stable with less fluctuation results while the test error is about 5% lower than the former cases. However, this one is very prone to the overfitting based on the literature as it's the most complex model. While the dataset we have is not huge, we do not see overfitting yet, though this model is not considered as a good model.

Email: hooshmand.1@osu.edu
Website: http://u.osu.edu/hooshmand.1

Mohammad Shahriar Hooshmand
Ph.D. Candidate
Department of Materials Science and Engineering
The Ohio State University

So, in overall one hot coding of **(mood, timeofday:time)** which gives the highest accuracy overt test data is chosen. the best results among the others.



Also, the results show that average method typically gives **higher accuracy** than the normal method. We see that normal method fluctuates a lot which is strange as it's not converging very well. It's expected to occur since it is a very naive form of stochastic gradient descent method with the unity learning rate which is very high and this causes large jumps around the minimum point. However, the average method gives more stable and typically better test accuracy compared to the normal method.

c) The average method after 600 epochs which gave the best accuracy has the following weight vector:

[0.44714407 0.45549153 -2.09855085 2.02208475 0.52361017 1.65069492 -5.32727966 5.21735593 0.5570678 -2.94501695 1.415 1.31771186 -2.08508475]

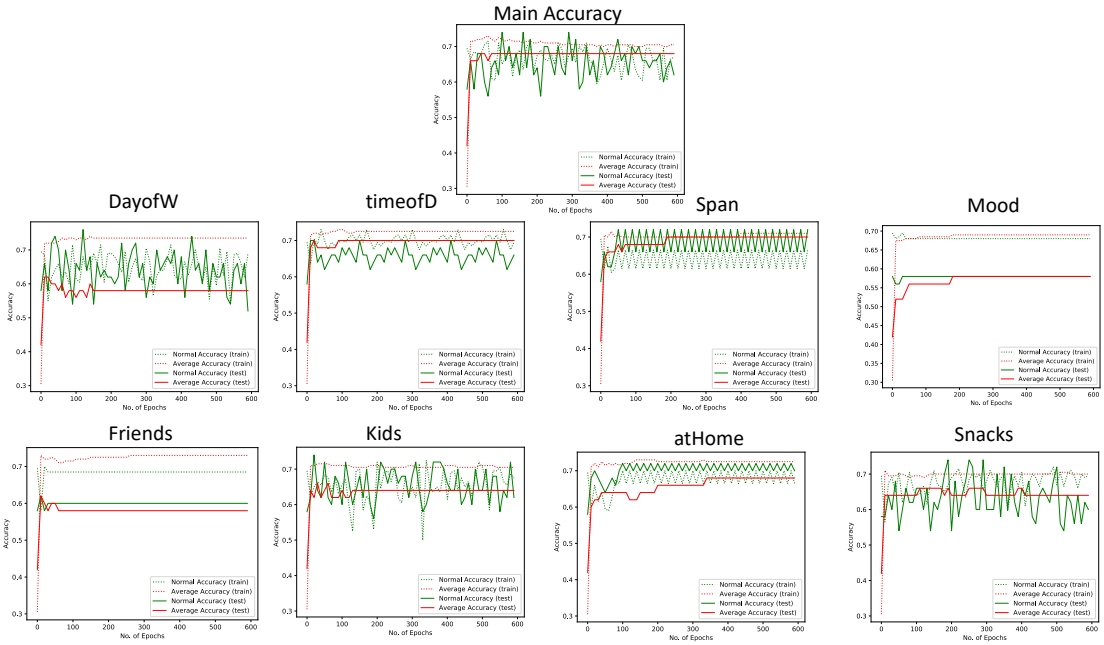
The format is:
[bias, dayofweek, morning, afternoon, evening, timetoplay, silly, happy, tired, friends, kids, athome, snack]

Based on the above features, silly and happy moods have the largest weight among the other parameters.

Email: hooshmand.1@osu.edu
Website: http://u.osu.edu/hooshmand.1

Mohammad Shahriar Hooshmand
Ph.D. Candidate
Department of Materials Science and Engineering
The Ohio State University

d) In implementing ablation method, every feature is dropped once and the accuracy is calculated over the training and test set. Here are the results. The top figure is considering all features (mood and time of day are one hot encoded) and the eight figures below it are the accuracies where that particular feature is dropped.



As it can be observed, removing Mood feature dropped the accuracy to the lowest value (57%). Friends visiting and day of week are the other two features that mostly affected the accuracy after mood.

e) Generally, feature selection can depend on many factors. That's the basis of different ablation techniques available in the literature. Typically, ablation methods are more systematic and are expected to magnify the effect of each feature on the final accuracy results in a more meaningful quantitative framework. In comparison, weights in the perceptron algorithm are kind of representative of the correlated effects of different features. So, the weight values are not necessarily indicative of individual isolated feature without the effect of any other features.

In addition, if the values of features are not normalized or not Boolean, the weights are updated proportional to the attributes of the features. As such, in a general dataset where the data is not necessarily Boolean or limited integers, comparing the weights would not be meaningful as they are updated to lead the convergence in proportional to their corresponding attributes.

Email: hooshmand.1@osu.edu
Website: http://u.osu.edu/hooshmand.1

Mohammad Shahriar Hooshmand
Ph.D. Candidate
Department of Materials Science and Engineering
The Ohio State University

f) One advantage of averaged perceptron is that we can maintain a running sum of the wrights. Problem with the normal perceptron is that we would like the perceptron not to make updates on every example which is the idea of averaged perceptron. Ideally, the averaged weight vector is updated when the actual weight vector is changed. In addition, the average method is kind of healing the effect of dataset ordering we encounter in the normal perceptron. That means the ordering of data affects the weight update over scanning through each sample though in the averaged method, the accumulative effect of each batch is used to update the weight at the end of the epoch.

So, the averaged perceptron is better than normal perceptron as it generalizes better to the test data. However, we should note that it is prone to overfitting more than the normal perceptron and we should make sure to stop training after proper amount of iterations.

Email: hooshmand.1@osu.edu
Website: http://u.osu.edu/hooshmand.1