

## Code Working :

**train\_net.py** : The program reads train.csv as input in dataframe data. Then for all the columns, ? Entries were replaced by mode of the respective column by first changing them to None and then using fillna function. Then for the columns whose dtypes are object , I replaced it with category type and then replaced it entries by using cat.codes so as to assign them specific integers.

Now I have X matrix from which I have removed id and salary columns and normalized all columns while Y contains all the salary outputs for it.

I have used 1 hidden layer and the number of hidden nodes in the layer can be changed by changing the variable nn\_hdim(I have put 50 nodes default in the hidden layer). Now neural network is trained on data using back propogation using tanh as the activation function. I have kept number of iterations to be 1000 although it can be changed easily. After this weights and bias are loaded to weights.txt.

**test\_net.py** : Program reads kaggle\_test\_data.csv and stores it into dataframe test\_data. Now from test\_data, id column is dropped and stored into test\_ids. Then it reads weights and biases from weights.txt as input. Now I transforms the input data similar to the transformation done in train\_net.py and then these wighths and biases are used in forward propogation on test\_data to get the predictions which are stored in predictions.csv.

## Feature engineering :

For all the columns, ? Entries were replaced by None and then filled by mode of that respective column using fillna function. Then the columns which dtypes are object, I replaced it with category type and then replaced it with cat.codes so as to assign them with specific integers.

## Comparison :

Comparing kaggle score predictions, my neural network performs better than decision tree classifier,MLP classifier and Adaboost classifier scores whose predictions are stored in predictions\_1.csv, predictions\_2.csv, predictions\_3.csv respectively.

Kaggle score of my submitted code : 0.79989

Kaggle score of predictions\_1.csv- 0.74163

Kaggle score of predictions\_2.csv- 0.75342

Kaggle score of predictions\_3.csv- 0.77277

## References/Citations :

1.[http://scikitlearn.org/stable/auto\\_examples/classification/plot\\_classifier\\_comparison.html](http://scikitlearn.org/stable/auto_examples/classification/plot_classifier_comparison.html) (for differnet classifiers in train\_standard.py)

2.<http://www.wildml.com/2015/09/implementing-a-neural-network-fromscratch/> (for neural network implementation)