**Using Deep Neural Network on Predicting Results of Law Cases**

Minghan Wang

06.25.2018

**Table of Contents:**

Introduction-------------------------------------------------------------3

Improvements compared to previous approaches-----------------3

Overall Methodology--------------------------------------------------3

Data Preprocessing-----------------------------------------------------4

Case Categorization----------------------------------------------------4

Building the Neural Net-----------------------------------------------6

Training the Neural Net-----------------------------------------------7

Model Result------------------------------------------------------------7

Future Improvements--------------------------------------------------8

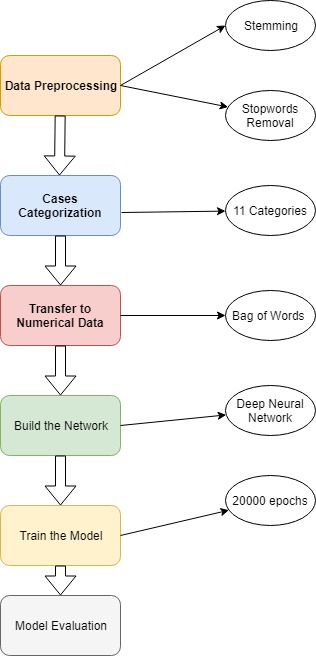
**Introduction:**

The documentation of each law cases are great amount of text data. If one is able to put them into good use, the data could provided great amount of information. In each law case, the result of it, whether it’s reversed or affirmed, is always related to the content of the case. The goal of this project is to use deep neural networks to predict the result of law cases.

**Improvements compared to previous approaches:**

Previous approaches only categorize the results of the cases into two groups, affirmed and reversed. However, there are actually 11 different results for the cases provided. Traditional machine learning model such as random forest or decision tree won’t have good performance on a ’11 class classification’ problem while using deep neural network model would be a better choice.

**Overall Methodology:**



Data Preprocessing:

**Stopwords Removal and Stemming**

The raw data was first tokenized, each case was tokenized into a list of words. And then the stopwords were removed while also stemmed.



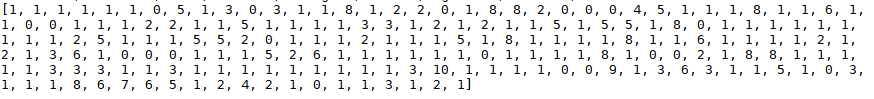
**Figure 1** Result of Case 1 being Stemmed and Stopwords Removed(Output too large, only part of the result was put)

**Case Categorization:**

Each law case document has recorded the results of that case. There are in totally 11 outcomes for the given cases.

|  |  |
| --- | --- |
| Case Result | Category Number |
| Reversed | 0 |
| Affirmed | 1 |
| Affirmed in part, reversed in part, and remanded. | 2 |
| Reversed and remanded | 3 |
| Affirmed in part, reversed in part, and vacated. | 4 |
| Vacated and remanded | 5 |
| Affirmed in part and reversed in part. | 6 |
| Appeal dismissed; Affirmed | 7 |
| Affirmed in part, vacated in part, and remanded. | 8 |
| Affirmed in part and modified in part. | 9 |
| Motion denied | 10 |

Above table shows the results of different cases provided and the corresponding category number that was used in the project. For example cases that were affirmed were categorized into group 1 while cases that were reversed were in group 0.

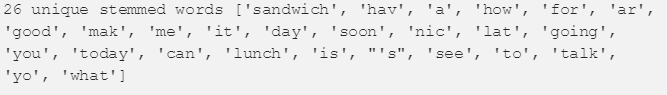


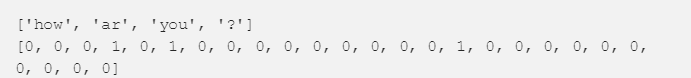
**Figure 2**

Figure 2 shows an array of length 200 which equals to the 200 cases provided. Each number in the array represents the results of the corresponding case. First index is 1 shows that case 1 is an affirmed case while case 8 is vacated and remanded as the eighth index is 5.

**Transfer Text Data to Numerical Data:**

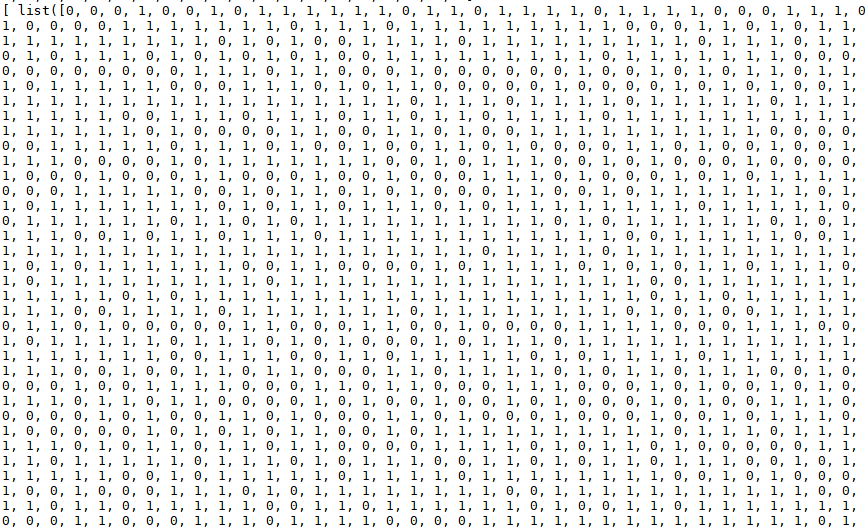
Because neural network model only takes in numerical data, the text data must be transferred into numerical data using some algorithms. The algorithm used in this project is the bag of words approach. The bag of word approach is a simple algorithm that transfers the text data into a binary array which only contains 0 and 1. The approach is to create an array contains the unique words in all the 200 law cases which in other words, a bag of words. And then for each case, create another array contains of 0 and 1s, where the indices are 1 when the corresponding words are in the bag of words and otherwise 0.





**Figure 3**

Figure 3 shows a simple example of how bag of words approach works. For example, the unique words in all documents are shown as above with size 26. Then “How are you ?” would be transferred into the above shown array. The word “how” are in the forth index in the first array, so the forth index in the second array is 1.

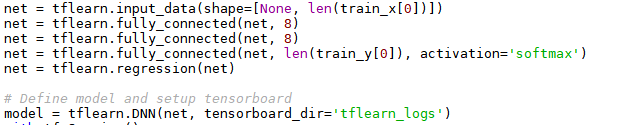




**Figure 4**

Figure 4 above shows an example of a binary representation of the case 1 using the bag of word approach. The second array shown in the figure with a length of 11 is the binary representation of the category this case is in. As show, case 1 is in group 1 which is affirmed.

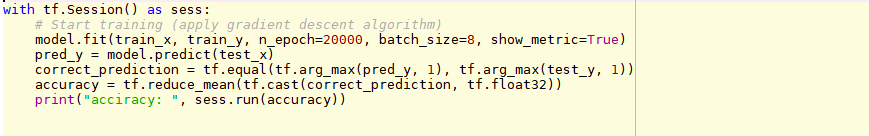
**Building the Neural Net:**



The network is built using the tensorflow and tflearn package. The neural net contains three fully connected layers where fully connected means each neurons of the current layer is connected to all the neurons in the previous layer. The activation function used was ‘softmax’ which would convert the numerical data into probabilities. And the output layer is set as a regression which will perform a regression to the input.

**Training the Neural Net:**

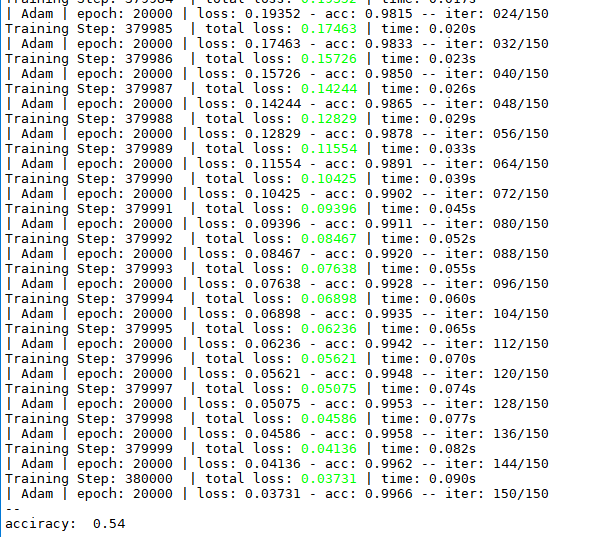
Before training the model, the processed numerical dataset is splitted into two set, a training set with 150 cases which were passed into the model as input and a testing set with 50 cases which were used to evaluate the accuracy of the model.



**Figure 5**

The model was trained on 20000 epochs. The number of epoch to be trained is adjustable.

**Model Result:**



**Figure 6**

The accuracy of the neural network is 0.54. The accuracy is not high enough because it is a 11 class classification problem instead of a common binary classification problem. But it is much higher than randomly guessing from the 11 outcomes which only has a 1/11 probability of getting right.

**Future Improvements:**

1. The size of the dataset only contains 200 cases which is not enough for neural networks to perform well. Once more cases are provided, the model could improve its accuracy.
2. Some contents in the cases are not relevant to the outcome of a case such as its headings or some people’s names. Those irrelevant text could be removed before being transferred into numerical data and being passed into the model.