Introduction

In this report, we want to precisely predict cuisine when given list of ingredients. Each ingredients English and we tried DNN and word stemming. At first, we modeled input ingredients as input of DNN's input layer. But after some observation on dataset, we decided to make a better modeling and, finally, we applied word stemming on input gredients. After changing the modeling, predict accuracy on test set increased upto 3%.

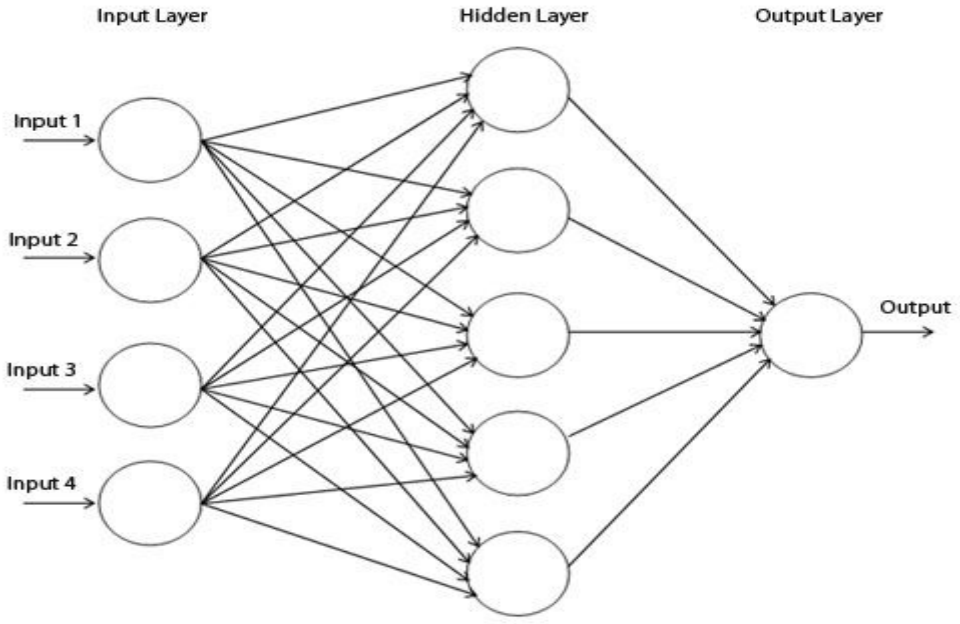
Background

Word stemming

Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. An example of stemming can be done to these words, cat, cats, catty, catlike. These 4 words commoly have meaning of cat, so the stemmer will extract 'cat' from these words. In our wokrs, we will use stemming to reduce the number of ingredients for modeling our inputs. In short, we want to consider both chicken and chickens as 'chicken'.

MLP(MultiLayer Perceptron)

MLP is quintessential deep learning model.



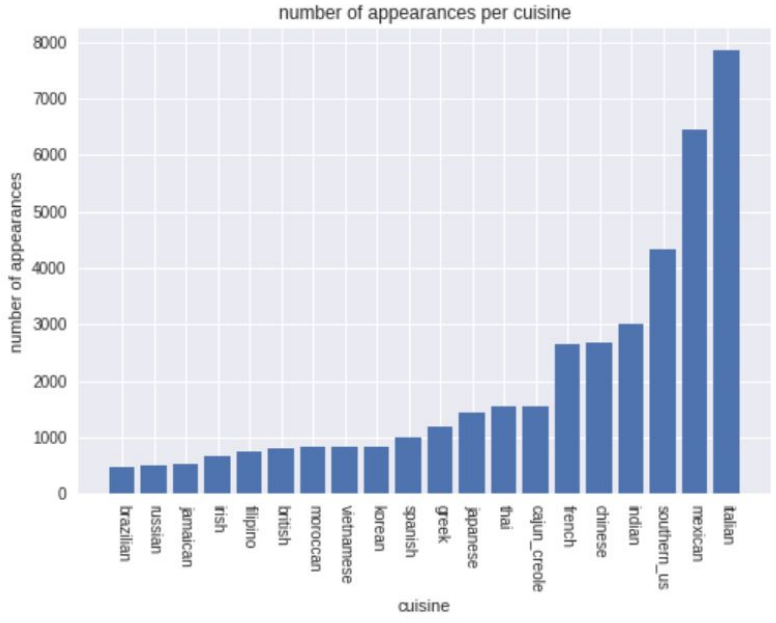
MLP has 3 types of layer, which is Input layer, hidden layer(one or more), Output layer. Input layer and output layer (denoted by its name) form as input and output of MLP. All layers

except input layer and output layer are called hidden layer. Hidden layer consists of hidden neuron that represents hidden feature of input. In basic MLP, all neurons in previous layer are connected to all neurons in next layer. We call this as ‘fully connected layer’. Inputs make signal from input layer and this propagate to output layer(forward propagation). After output comes from output layer, the loss between output make another signal. This signal propagate from output layer to input layer(backward propagation). When backpropagation, parameters of neurons in hidden layer are changed so that total loss become lower.

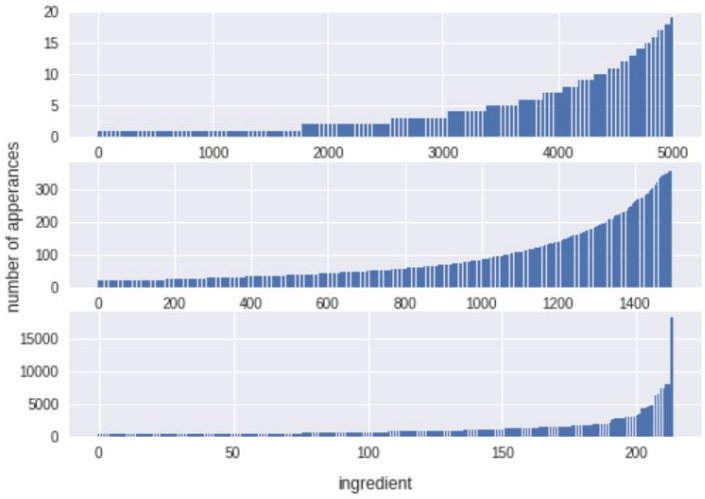
Methodology

Data analysis

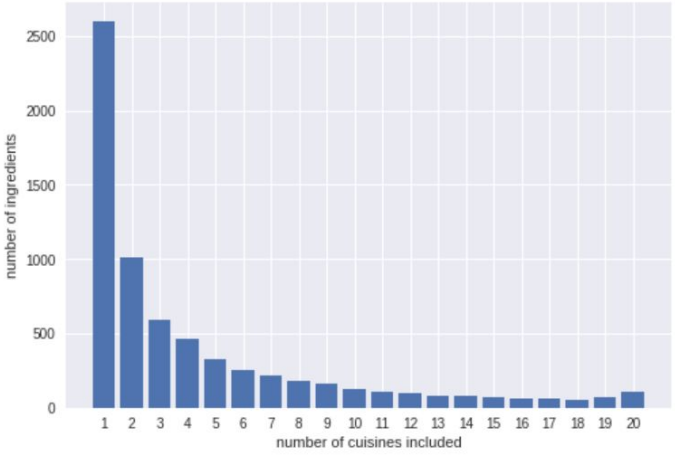
Number of data in train set is 39,774 and test set is 9,944. Total number of unique ingredients in train set is 6,714 and total number of unique cuisines(which is number of our output class) is 20. Average number of ingredients in cuisine is 10.77. If we modeling our input as 1d array of size 6,714 which is boolean data whether input has each ingredient, this array will be very sparse. Also, among 4,844 ingredients in test set, there are 423 ingredients that never appears at train set.



This graph describes cuisines in train set. Top 3 cuisines, which is italian, mexican, southerns_us occupy about 50% of train set data.



Above graph shows the number of appearance per ingredients. Most of all, about 1,800 ingredients is used only once and 3,500 ingredients appears less than 5 times. Considering total number of class is 20, this 3,500 ingredients may have few contribution on deciding cuisine.




```
number of ingredients included in all cuisines: 107
```

This shows how many ingredients are included at cuisines. (x, f(x)) means there are f(x) ingredients which is included at x cuisines. Especially, when x is 20, these ingredients are included at all cuisines. So these ingredients may have weak factor for deciding cuisine.

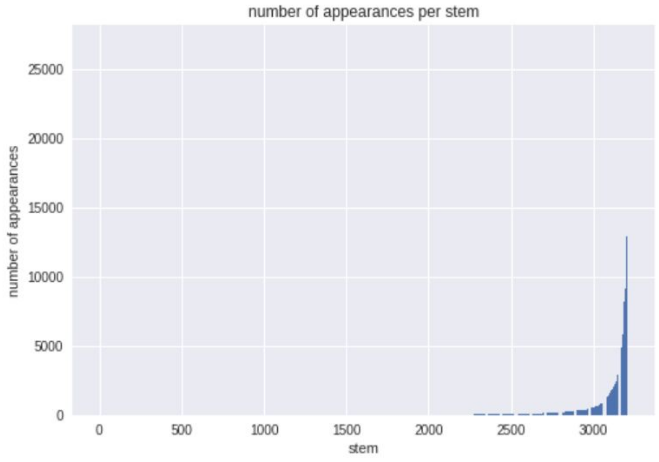
When we sum it up, if we use ingredients itself as a method of modeling inputs, it will be very sparse and, therefore, we need another modeling.

Apply stemming

```
fat-free reduced-sodium chicken broth
chicken stock
skinless chicken thighs
canned low sodium chicken broth
smoked chicken sausages
unsalted chicken stock
non fat chicken stock
gluten-free chicken stock
chicken feet
cooked chicken breasts
chicken thighs
bone in skinless chicken thigh
chicken fingers
canned chicken broth
25% less sodium chicken broth
rich chicken stock
chicken wings
knorr reduc sodium chicken flavor bouillon
gluten free chicken broth
chicken breasts
```

Above is few ingredients comprising the word ‘chicken’. As you can see, string it self is different but they commonly have the meaning ‘chicken’. We can guess(이 단어 좋지 않은거 같아, 추측하다? 예상하다? 좋은단어 추천좀) that many ingredients can have similar meaning.
So we will use ‘word stemming’ to extract important meaning from ingredient which is stem.

```
number of stems: 3211
```



After stemming, stem acts as new ingredients. We extract 3,211 stems from train set, and this is done by outer module, stemming. Stem’s distribution is almost similar with ingredients. This means that we can reduce stems again. About 1,500 stems appears less than 6 times. We made a variable ‘stem_reduction_factor’ so that we can control how may stems remain.

Experiment

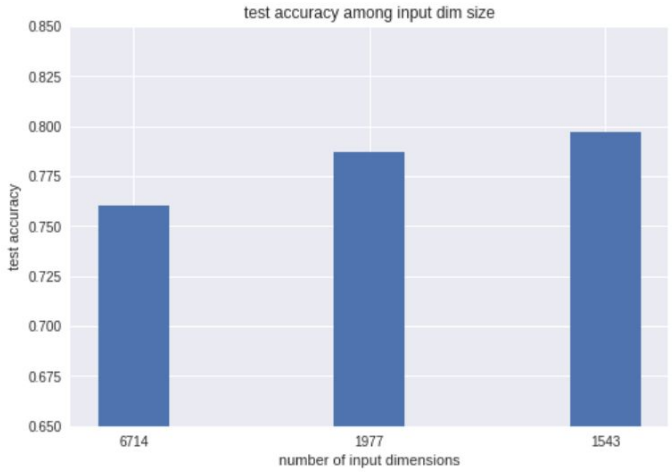
MLP setup

Our model has 2 hidden layers and output layers. Each hidden layer has 512 neurons. Activation function is leaky relu and for generalization performance(not to overfit) used dropout and l2 normalization. Also, we used he initializer which is good for relu(and its variant) activation. Also we did input conversion because we will use stem.

Split dataset

We divided train set into 75%(train set), 5%(validation set), 20%(test set) with random sampling.

Result



Without stemming, test accuracy was 0.76. But after stemming(based on ‘stem_reduction_factor’), test accuracy reaches at 0.797. Out model also used dropout and l2 normalization which increases test accuracy about 0.05.

Conclusion

In this workout, we tried to modeling input using word stemming. We used DNN as a prediction model and also used some ML techniques such as fine-tunning and retraining, dropout, l2 regularization, hyperparameter tunning to increase generalization. At last, we got accuracy 0.797 on test set data. Also this can be changed at every execution because train set and test set will be changed on each execution.