AI-POWERED SPAM CLASSIFIER

PHASE 4:

In this phase, We use various algorithm for spam classification.

Algorithms:

- Multi-layer perceptron
- K-Nearest Neighbour
- Decision Tree
- Support Vector Machine

1. Multi-Layer Perceptron:

Abstract:

Spam email is a kind of junk email. Recent years, with the rapid growth of the internet users, especially emails users, the spam emails have been regarded as a severe problem. there are many classification methods that can be used to detect spam, Naïve Bayesian and Decision tree for example. These methods all gains good performance in most of cases, but the true positive rate and the false positive rate of them are not good enough. In this paper we designed a neural network based spam classification algorithm to filter spam. Our model is a classical multi-player perceptron composed of two hidden layer, one input layer and one output layer. By applying different threshold to plot the roc curve, we demonstrate that our method outperforms most of existed method. We also demonstrate that ensemble learning will boost the whole method.

Keywords: Multi-layer perceptron, spam, ADAM, ROC

Introduction:

Spam email is a kind of junk email. Most of spam emails are commercial advertisements or contain disgusting contents like violence or porn. Some of them may contains viruses that might hurt receivers' computer. Recent years, with the rapid growth of the internet users, especially emails users, the spam emails have been regarded as a severe problem. Internet users' normal life has been effected. Under that cases, developing a technique to filter the spam has been a more and more important topic. In most of cases, spam can be filtered by a black list. By rejecting to receive email from specific addresses that have been annotated as the main source of spam, the number of spam in our mail box will decrease. This method only works under the condition that the source of spam is limited. However, when the spam sender builds a huge group of computers from allover the world that being controlled by trojans horse, it is no longer realistic to ban all of them.

- Multi-layer perceptron will transform the input into a complex feature space by the linear combination of input vector. The core point for the neural network to lean ingenious is the activation function:

$$f(x) = \frac{1}{1 + e(-x)} .$$

- In which x is the linear combination of this layer. But by applying this activation function, the gradient vanishing problem [10] might occurs, so in our model, we adopt the RELU function .

$$f\ x = \max 0, x .$$

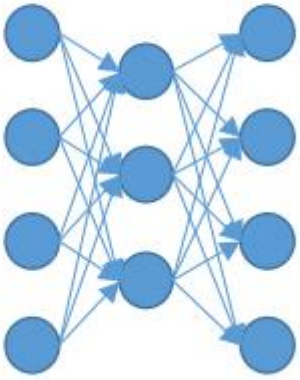
- This function punishes all the negative value to be zero, and this activation function will give us linear gradient during the training process. With the activation function shown above, we get the feed forward process as:

$$p = f(w^{(2)} f(w^{(1)} x + b^{(1)}) + b^{(2)}).$$

Regularization:

In most of cases, the training process of neural network will make the model fit training data perfectly, but the variance of model will be very high, which means the model was over fitted to the data, so here we adopt the L2 norm [11] of parameters to the neural network, that will make our cost function as:

$$J(\theta; x, y) = \frac{1}{2}\sum_{i=1}^{N} | y_i - \bar{y}_i |^2 + \frac{1}{2} w^{(1)T} w^{(1)} + \frac{1}{2} w^{(2)T} w^{(2)}.$$



This operation was implemented with weight decay parameters in pytorch optimizer. And the weight decay parameter is used to tuning the importance of the regularization part and model bias part. On the other hand, we also use dropout operation, preserving activation value the fixed probability, was proved to be equal to the L2 regularization [17,18].

Auto-encoder:

In traditional machine learning, the depth of neural network can't be too deep for the effect of gradient vanishing [4], which strongly constraint the performance of multi-layer perceptron. Hinton proposed a deep belief net [15], and Vincent proposed a stacked autoencoder [16] to automatically extract the features from the data, and enhance the performance of classifier.

The formation of autoencoder is very simple, and it was composed of a encoder and a decoder. The decoder plays as an inverse operation of decoder.

$$\text{code}=f_1 \quad f_2 \quad f_3(X)$$
$$\bar{X}=f_3^{-1} \quad f_2^{-1} \quad f_1^{-1}(\text{code})$$

When we can recover the X from its code with small error, we can claim that the code contains most of information of the original data, on the other hand, the code is a good representation for the original feature. When we concatenate this autoencoder to another classifier, like SVM, we will gain a deeper model, which will show good performance.

PROGRAM:

```
import pandas as pd
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Load the dataset
data = pd.read_csv('spam.csv')

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data['text'], data['label'],
random_state=0)

# Create an MLP classifier with two hidden layers of 16 and 8 neurons each
mlp = MLPClassifier(hidden_layer_sizes=(16, 8), max_iter=1000)

# Train the classifier on the training set
mlp.fit(X_train, y_train)

# Predict the labels of the test set
y_pred = mlp.predict(X_test)

# Print the accuracy of the classifier on the test set
print('Accuracy:', accuracy_score(y_test, y_pred))
```

OUTPUT:

	message	label
0	Go until jurong point, crazy.. Available only ...	0
1	Free entry in 2 a wkly comp to win FA Cup fina...	1
2	U dun say so early hor... U c already then say...	0
3	FreeMsg Hey there darling it's been 3 week's n...	1
4	As per your request 'Melle Melle (Oru Minnamin...	0

Conclusion:

With the help of multi-layer perceptron, we gains performance enhancement on the spam base dataset. And with given features, we can predict spam with over 90% accuracy, that beats lots of existed method. What's more, we found that with the help of ensemble learning method, the performance of classifier can be further boosted.

Future work:

Though we can use the bag of word model of IF-IDF model to extract feature of any given emails, but this method can not be utilized to represent a documents perfectly, which means these feature is not perfect for the classification task. We believe that by integrating the feature extracting task and classification task, the model can be trained in a whole, and gains better performance. These emerging deep learning method like CNN and LSTM will be good tools to solve this.

2. Support Vector Machine:

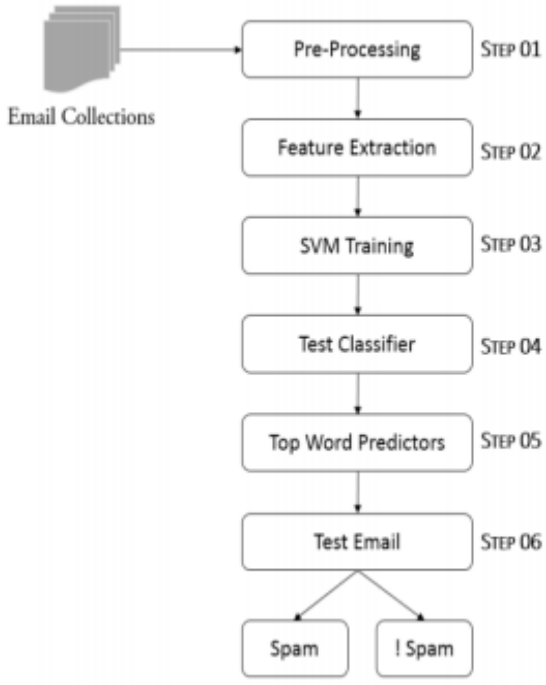
INTRODUCTION:

Recently unsolicited commercial / bulk e-mail, also known as spam, becomes a major problem on the Internet. Spam is a waste of time, storage space and communication bandwidth. The problem of spam and fraud e-mail has been increasing for years. In recent figures, 40% of all mail is spam that emails about 15.4 billion emails per day and costs Internet users about $ 355 million per year. Automatic e-mail filtering is the most effective way to deal with spam at the moment and there is a fierce competition between spammers and spam filtering methods. Now Spammers began using several tricky methods to overcome filtering methods such as using random sender addresses and / or adding random characters to the beginning or end of the message subject line.

PRELIMINARY AND PROBLEM STATEMENT :

A Spamming is one of the major and common attacks that accumulate a large number of compromised machines by sending unwanted messages, viruses, and phishing through email. We have chosen this project because now there are many people who are trying to fool you just by sending you fake e-mails, as if you have won 1000 dollars, deposit this amount in your account as soon as you open this link. Once done, they will track you and try to hack your information. Sometimes relevant e-mail is considered spam email. Unwanted email is harassing Internet consumers in ways such as:

- Important email messages were missed and / or delayed.
- Consumers seek ISP's frequent email delivery changes all the time.
- Internet performance and bandwidth loss.
- Millions of compromised computers.
- Loss of billions of dollars worldwide.
- Identification of theft.
- Increase in several viruses and Trojan horses.
- Spam can crash and affect the mail server and fill the hard drive.



Pre-processing:

The pre-processing step is used to remove noise from emails that are irrelevant and need not exist .

Pre□processing phase includes:

1. Removing Numbers
2. Remove special symbol
3. URL deletion

4. HTML tags separating task
5. Performing Word Steaming

Feature Extraction :

Feature extraction technique is used to extract important and relevant features from the email body. Feature replaces email 2D vector space features numbers. These features are mapped from the dictionary list.

SVM Training:

Email spam is used for training purposes. Training datasets contain spam content and are trained using this classifiers. After training, the classifier is ready to classify spam emails.

Test Classifier:

The classifier is tested with several training data to test it accurately. Until the proposed solution is obtained which gives 98% accuracy in classifying email. e. Test Email After the training phase is completed, the classifier is given a sample email as input to classify the email. Classifier Generates output in forms of 0 or 1, 1 means it is spam and 0 means it is not spam.

CONCLUSION:

In this study, we reviewed the general application in the field of machine learning approach and spam filtering. A review of the state of the art algorithm has been implemented to classify the message as either spam or ham. Efforts made by various researchers to solve the problem of spam through the use of machine learning classifiers were discussed. The development of spam messages was investigated over the years to avoid filters. The basic structure of the email spam filter and the processes involved in filtering spam emails were noted. The paper surveyed some of the publicly available datasets and performance metrics that can be used to measure the effectiveness of any spam filter.

The challenges of machine learning algorithms in efficiently handling the threat of spam were pointed out and a comparative study of machine learning techniques available in the literature.

We also revealed some open research problems related to spam filters. In general, the amount and amount of literature we reviewed suggests that significant progress has been made and will still be made in this area. After discussing open problems in spam filtering.

Our hope is that research students will use this paper as a spring board to conduct qualitative research in spam filtering using machine learning, deep learning, and deep adversarial learning algorithms.