Long and Short-Term Memory for Email Classification

Weifan Zhang
University of Dortmund
`weifan.zhang@tu-dortmund.de`

Abstract

How to more accurately identify spam has always been a difficult problem in the field of machine learning. The performance of various machine learning algorithms in the past in processing and identifying spam is not good, such as KNN, linear regression and so on. Due to the better effect of long and short-term memory (LSTM) in the field of time series data, LSTM has become the main machine learning algorithm for researching and identifying spam. In this paper, i will describe the development history of Recurrent Neural Networks (RNNs), their advantages and disadvantages, and their applications in the context of the modern computing environments. Later, i will introduce an experiment using LSTM for mail classification. Through this implementation, men will have a deeper understanding of recurrent neural networks, including the working principles of LSTM, and the underlying neural networks that are used to generate them.

1. Introduction

With the development of artificial intelligence, various advanced machine learning algorithms and models have been developed to help people understand the world around them. In modern society, we can use artificial intelligence to achieve many things that could not be achieved before. Like robot customer service, this is not false artificial intelligence has implemented many if-else statements in the past, only responding to messages with specific keywords. Today's intelligent robots can learn human language habits through big data, and give corresponding, fluent, reasonable answers that are closer to everyday communication. It is precisely because of these machine learning algorithms, especially LSTM, that we can now use these algorithms to better identify spam so that spam will interfere with our normal life as little as possible.

Next, i will introduce the use of recurrent neural networks to process human language so that machines can understand the meaning and emotions of human language.

2. History of Recurrent Neural Networks

Recurrent neural networks are a new model developed based on a neural network. To understand recurrent neural networks, we need to first understand what a neural network is.

2.1 Neural Networks

A neural network is a computational model of a neural network created by Warren McCulloch and Walter Pitts [9] (1943) based on mathematics and an algorithm called threshold logic.[10]

The following two pictures can help us better understand the working principles of neural networks. The neuron receives a series of inputs, processes them inside the neuron, and then produces corresponding outputs.

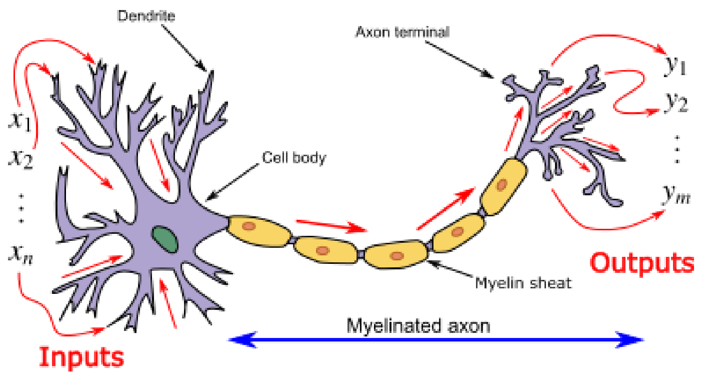


Figure 1: Neuron and myelinated axon, with signal flow from inputs at dendrites to outputs at axon terminals [3]

The neurons in the neural network are similar to the neurons of the human brain, so this computational model is called a neural network.

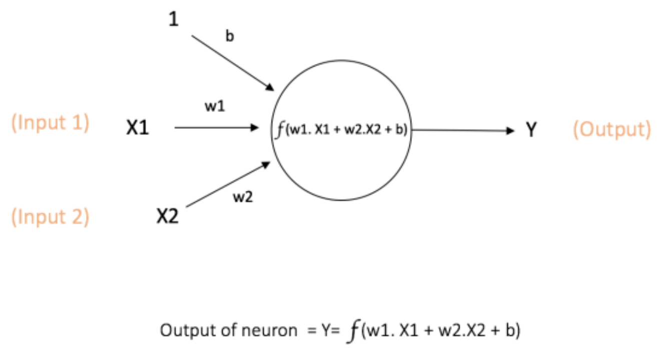


Figure 2: a single neuron [4]

The above network takes digital inputs X1 and X2 and has weights w1 and w2 associated with those inputs. Besides, there is another input 1 (called bias) associated with weight b. The calculation of the output Y of the neuron is shown in Figure 2. The function f is nonlinear and is called the activation function. The purpose of the activation function is to introduce nonlinearity into the output of the neuron. This is important because most real-world data is non-linear, and we want neurons to learn these non-linear representations. There are various activation functions, and each activation function has its role. Common activation functions include sigmoid (for binary classification), tanh, and so on.

2.2 Recurrent Neural Networks

The neurons in a traditional neural network are independent of each other. They have their input and output, and each neuron does not interfere with each other. This makes traditional neural networks incapable of complex tasks. RNNs is a relatively complex neural network model, and its development is based on David Rumelhart's work in 1986. [10] With the invention of the Elman network in 1990, RNNs were widely popularized. [7]

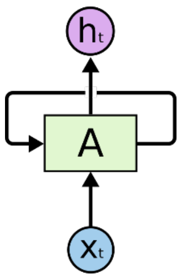


Figure 3: Recurrent Neural Networks have loops [5]

By observing the mathematical formula, we can know that the working principle of RNNs is roughly the same as that of a neural network. It is a combination of weight w and input x_t, and an bias b is added to get the output h_t. The difference is that RNNs will use the output h_{t-1} of the previous network as one of the input values of the next network. In this way, the previous information can be continuously transmitted along with the network, allowing RNNs to remember more things. A loop allows information to be passed from one step of the network to the next. [5]

$$h_t = tanh(W_x * x_t + W_h * h_{(t-1)} + b) \tag{1}$$

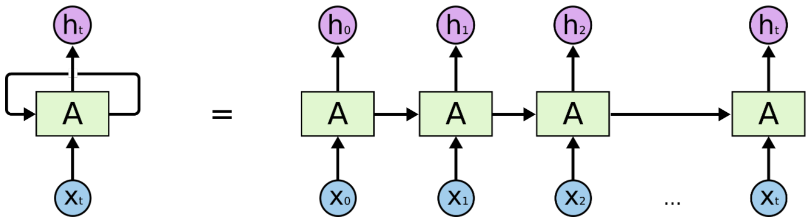


Figure 4: An unrolled recurrent neural network [5]

Because of the powerful functions of RNNs, RNNs are often used in applications such as text generation, machine translation, and speech recognition. But RNNs also have obvious shortcomings. RNNs have one major shortcomings: Cannot Capture Long-term Dependencies. Let us look at the following example:

If we want to predict the last word in the sentence "The grass is green", that is totally doable. But if we want to predict the last word in the sentence "I am French, (2000 words later), I speak fluent French".[5] We need to be able to remember long range dependencies. RNNs are bad at this. They forget the long term past easily. In theory, RNNs can solve the problem of "long-term dependence", but the actual performance is not good. But do not worry, this problem was solved in 1997. Long short-term memory networks were invented by Hochreiter and Schmidhuber in 1997 and set accuracy records in multiple application domains.[8]

2.3 Long and Short-Term Memory

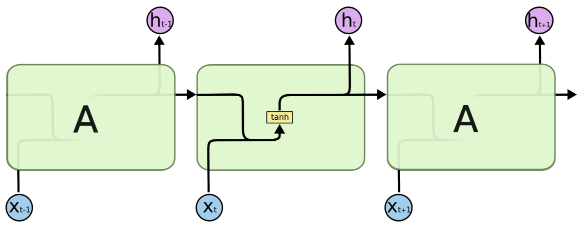


Figure 5: The repeating module in a standard RNN contains a single layer. [5]

All RNNs are repeating and chaining the modules of a neural network many times. In standard RNNs, this repeated module is a very simple structure, such as a single-layer tanh layer, as shown in Figure 5. LSTM also have such a chain structure, but the structure of the repeating module is different. Unlike only a single-layer neural network, LSTM have 4 layers.

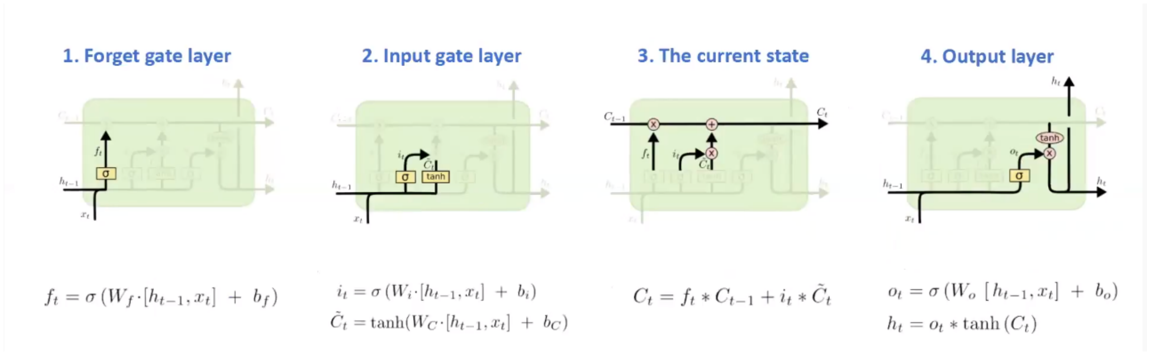

1. Forget gate layer
2. Input gate layer
3. The current state
4. Output layer
$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$
$h_t = o_t * \tanh(C_t)$

Figure 6: Long Short-Term Memory Works [5]

The working principle of LSTM can be divided into four steps:

1. Add a forgetting mechanism: We want the model to learn a distinguished forgetting/memory mechanism: when a new input comes, it needs to know what information to remember and what to discard.

2. Add a saving mechanism: For example, when the model sees a new picture, it needs to learn whether the information about the picture is worth using and saving.

3. So when a new input comes, the model must first forget any long-term memory information that it thinks is no longer needed. Then learn which parts of the new input are worth using and save them in your long-term memory.

4. Focus long-term memory in working memory: Finally, the model needs to learn which parts of long-term memory are immediately useful. The model learns which part to focus on, rather than always using full long-term memory.

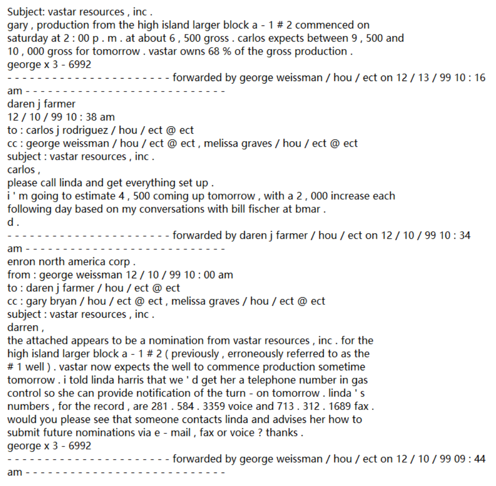
Compared with standard RNNs, LSTM transfers memory in a very precise way-using a specific learning mechanism: which parts of information need to be remembered, which parts of information need to be updated, and which parts of information need to be paid attention to. In contrast, RNNs will rewrite memory at every time step in an uncontrollable way.

3. Experiment

This experiment will use an email classification project to gain a deeper understanding of the usages and advantages of LSTM. The experimental process is roughly divided into the following 6 steps: data preprocessing, data transformation, modeling, prediction, visualization, and analysis. The programming language used in the experiment is Python, and the software used is PyCharm and TensorFlow.

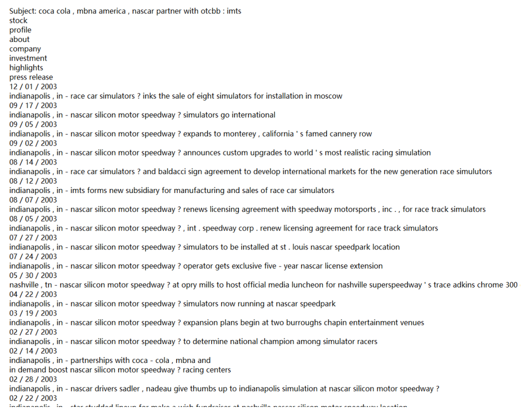
3.1 Data Preprocessing

The dataset for the experiment comes from the computer science course "Machine learning for natural language systems" at the University of Dortmund. The original data consists of .txt files that have been classified. These are emails with complete or incomplete content, as shown in the picture below. To make the data better accepted by the LSTM, we need to vectorize the data, which means that the content of each email is converted into a sequence that will be used in the next step.



```
Subject: vastar resources , inc .
gary , production from the high island larger block a - 1 # 2 commenced on
saturday at 2 : 00 p . m . at about 6 , 500 gross . carlos expects between 9 , 500 and
10 , 000 gross for tomorrow . vastar owns 68 % of the gross production .
george x 3 - 6992
- - - - - - - - - - - - - - - - - - - - - - forwarded by george weissman / hou / ect on 12 / 13 / 99 10 : 16
am - - - - - - - - - - - - - - - - - - - - - - - - - - -
daren j farmer
12 / 10 / 99 10 : 38 am
to : carlos j rodriguez / hou / ect @ ect
cc : george weissman / hou / ect @ ect , melissa graves / hou / ect @ ect
subject : vastar resources , inc .
carlos ,
please call linda and get everything set up .
i ' m going to estimate 4 , 500 coming up tomorrow , with a 2 , 000 increase each
following day based on my conversations with bill fischer at bmar .
d .
- - - - - - - - - - - - - - - - - - - - - - forwarded by daren j farmer / hou / ect on 12 / 10 / 99 10 : 34
am - - - - - - - - - - - - - - - - - - - - - - - - - - -
enron north america corp .
from : george weissman 12 / 10 / 99 10 : 00 am
to : daren j farmer / hou / ect @ ect
cc : gary bryan / hou / ect @ ect , melissa graves / hou / ect @ ect
subject : vastar resources , inc .
darren ,
the attached appears to be a nomination from vastar resources , inc . for the
high island larger block a - 1 # 2 ( previously , erroneously referred to as the
# 1 well ) . vastar now expects the well to commence production sometime
tomorrow . i told linda harris that we ' d get her a telephone number in gas
control so she can provide notification of the turn - on tomorrow . linda ' s
numbers , for the record , are 281 . 584 . 3359 voice and 713 . 312 . 1689 fax .
would you please see that someone contacts linda and advises her how to
submit future nominations via e - mail , fax or voice ? thanks .
george x 3 - 6992
- - - - - - - - - - - - - - - - - - - - - - forwarded by george weissman / hou / ect on 12 / 10 / 99 09 : 44
am - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

(a) Ham Email



```
Subject: coca cola , mbna america , nascar partner with otcbb : imts
stock
profile
about
company
investment
highlights
press release
12 / 01 / 2003
indianapolis , in - race car simulators ? inks the sale of eight simulators for installation in moscow
09 / 17 / 2003
indianapolis , in - nascar silicon motor speedway ? simulators go international
09 / 05 / 2003
indianapolis , in - nascar silicon motor speedway ? expands to monterey , california ' s famed cannery row
09 / 02 / 2003
indianapolis , in - nascar silicon motor speedway ? announces custom upgrades to world ' s most realistic racing simulation
08 / 14 / 2003
indianapolis , in - race car simulators ? and baldacci sign agreement to develop international markets for the new generation race simulutors
08 / 12 / 2003
indianapolis , in - imts forms new subsidiary for manufacturing and sales of race car simulators
08 / 07 / 2003
indianapolis , in - nascar silicon motor speedway ? renews licensing agreement with speedway motorsports , inc . , for race track simulators
08 / 05 / 2003
indianapolis , in - nascar silicon motor speedway ? , int . speedway corp . renew licensing agreement for race track simulators
07 / 27 / 2003
indianapolis , in - nascar silicon motor speedway ? simulators to be installed at st . louis nascar speedpark location
07 / 24 / 2003
indianapolis , in - nascar silicon motor speedway ? operator gets exclusive five - year nascar license extension
05 / 30 / 2003
nashville , tn - nascar silicon motor speedway ? at opry mills to host official media luncheon for nashville superspeedway ' s trace adkins chrome 300
04 / 22 / 2003
indianapolis , in - nascar silicon motor speedway ? simulators now running at nascar speedpark
03 / 19 / 2003
indianapolis , in - nascar silicon motor speedway ? expansion plans begin at two burroughs chapin entertainment venues
02 / 27 / 2003
indianapolis , in - nascar silicon motor speedway ? to determine national champion among simulator racers
02 / 14 / 2003
indianapolis , in - partnerships with coca - cola , mbna and
in demand boost nascar silicon motor speedway ? racing centers
02 / 28 / 2003
indianapolis , in - nascar drivers sadler , nadeau give thumbs up to indianapolis simulation at nascar silicon motor speedway ?
02 / 22 / 2003
```

(b) Spam Email

Figure 7: Original content of the email

We can see that whether it is a ham email or a spam email, the content contains many meaningless characters, such as the continuous "-", "/", "." etc. So, in addition to serializing the data, we should also delete these meaningless characters to make the model better trained.

The processed data is as shown in the table below. All contents in the email is put into one line, and each word block is separated by a space. Finally, add the corresponding label for it.In this experiment, we marked spam emails as 0 and ham emails as 1.

email	label
Subject: vastar resources inc gary production from the high island larger ...	1
Subject: looking for love tonight this is a great dating site ! ! ! please ...	0
Subject: urgent replyoverseas stake lotteryinternational spain batch ...	0
Subject: calpine daily gas nomination calpine daily gas nomination 1 doc.	1

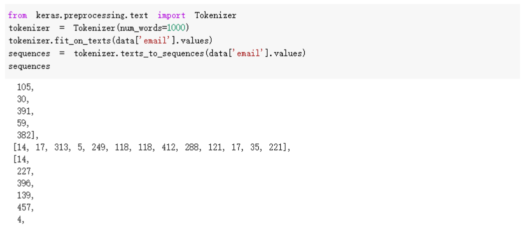
Table 1: Serialized data

3.2 Data Transformation

A simple string sequence is not well accepted by the LSTM model, so we need to further convert the data into a data structure that is more conducive to LSTM model training. And how to deal with this problem? In fact, there are already great solutions – Keras.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable error messages. It also has extensive documentation and developer guides.[2]

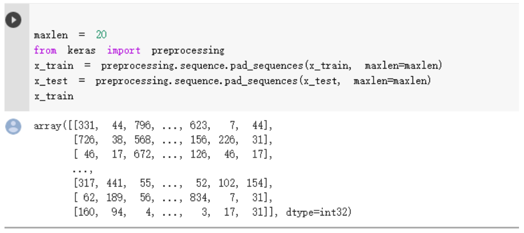
By using the functions test_to_sequences and pad_sequences in Keras, we can easily obtain data and data structures that can be better accepted by the LSTM model.



```
from keras.preprocessing.text import Tokenizer
tokenizer = Tokenizer(num_words=1000)
tokenizer.fit_on_texts(data['email'].values)
sequences = tokenizer.texts_to_sequences(data['email'].values)
sequences
```

```
  105,
  30,
  391,
  59,
  382],
 [14, 17, 313, 5, 249, 118, 118, 412, 288, 121, 17, 35, 221],
 [14,
  227,
  396,
  139,
  457,
  4,
```

(a) To Sequences



```
maxlen = 20
from keras import preprocessing
x_train = preprocessing.sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = preprocessing.sequence.pad_sequences(x_test, maxlen=maxlen)
x_train
```

```
array([[331,  44, 796, ..., 623,   7,  44],
       [726,  38, 568, ..., 156, 226,  31],
       [ 46,  17, 672, ..., 126,  46,  17],
       ...,
       [317, 441,  55, ...,  52, 102, 154],
       [ 62, 189,  56, ..., 834,   7,  31],
       [160,  94,   4, ...,   3,  17,  31]], dtype=int32)
```

(b) Reformat Sequences

Figure 8: Data Transformation

3.3 Modeling

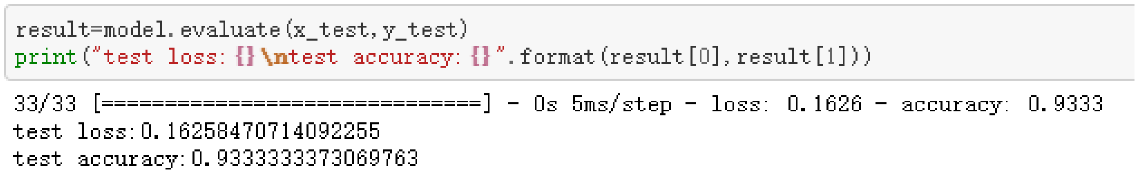
After the data is processed, we can start to train the data using the LSTM. In this experiment, i divided the data set at a ratio of 8:2, with 80% of the data as the training data set and 20% of the data as the test set.

The LSTM model used in the experiment comes directly from Keras. Each time the SpatialDropout1D layer is updated during training, the ratio of the input unit is randomly set to 0.2, which helps prevent overfitting. The activation function of the model is set to "sigmoid" because our classification results are binary data, either spam emails or ham emails, so using the sigmoid activation function is a good choice. For the same reason, the loss function is set to "binary_crossentropy".

I choose to use the "Adam" optimizer because RMSprop, Adadelta, and Adam are similar algorithms that do well in similar circumstances. Kingma et al.[6]show that their bias-correction helps Adam slightly outperform RMSprop towards the end of optimization as gradients become sparser. Insofar, Adam might be the best overall choice.Set 5 training cycles epochs = 5, batch_size = 32.

3.4 Prediction

After training the model, we can start to make predictions on the test data. The picture below shows the accuracy of the prediction.



```
result=model.evaluate(x_test,y_test)
print("test loss:{}\ntest accuracy:{}".format(result[0],result[1]))
```

```
33/33 [==============================] - 0s 5ms/step - loss: 0.1626 - accuracy: 0.9333
test loss:0.16258470714092255
test accuracy:0.9333333373069763
```

Figure 9: Result

We got a 93.3% prediction accuracy, which is a good result. But whether it is a good model, we need to observe other information. Such as F1 score, AUC, and so on.

3.5 Visualization and Analysis

precision	recall	f1-score	AUC
0.97	0.92	0.90	0.93

Table 2: result report

We know that the higher the values of precision, recall, and f1-score, the better the model. In addition, in order to prevent extreme situations, I also calculated the value of AUC. The value of AUC is between (0.5,1]. The closer to 0.5, the worse the model, and the closer to 1, the better the model. This shows that the model we trained is a good model. But we also need to observe the loss function.

Use matplotlib package to data visualization. In the following two pictures, the left side shows the accuracy function, and the right side shows the loss function.

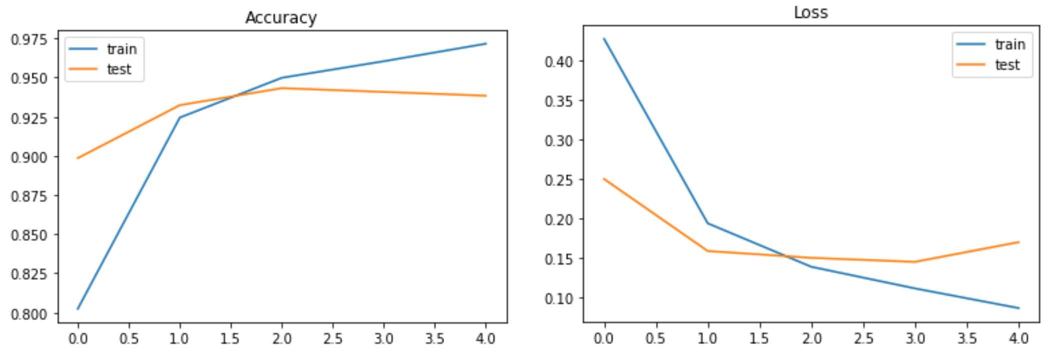
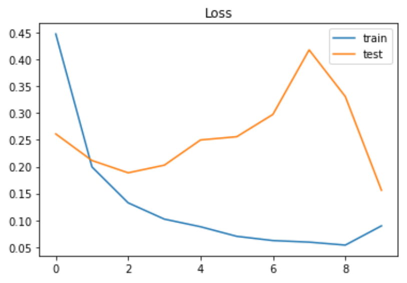
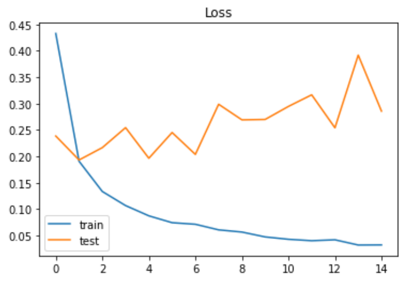


Figure 10: Graphic of Accuracy and Loss function

Normally, the trend of the train_loss line and the val_loss line should be the same. If their trends are consistent, then this model is good. But in my results, the trends of train_loss and val_loss are not consistent, so it can be inferred that there is a high probability of overfitting.

Next, try to use different parameters to achieve better results.

epochs	batch_size	accuracy	loss function
10	32	0.9487922787666321	
15	32	0.947826087474823	

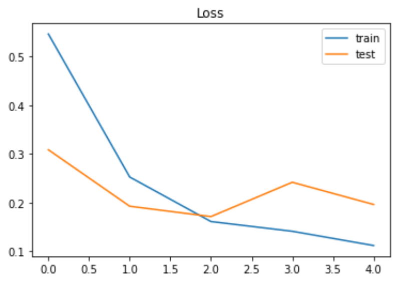
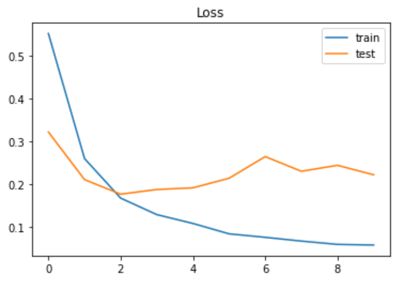
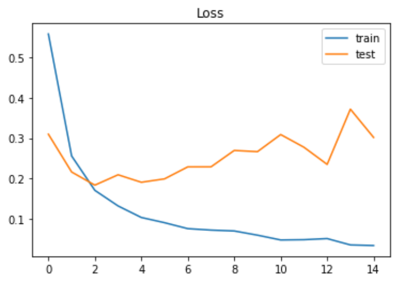
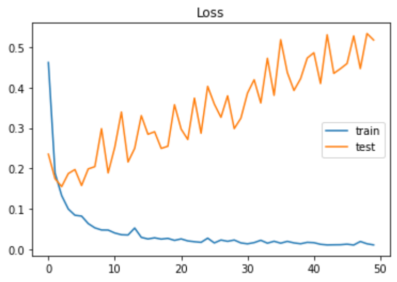
epochs	batch_size	accuracy	loss function
5	64	0.9246376752853394	
10	64	0.9439613819122314	
15	64	0.9420289993286133	
50	32	0.9326139453295158	

Table 3: Result

We can see that by changing the values of epochs and batch_size, the problem of overfitting cannot be solved, but it is getting worse. While looking through the Keras documentation, i found that there is a way to solve the problem of overfitting or underfitting - Callback Function: Early Stopping.[1]

By observing the graphs of these loss functions in the above experiment, i found that the cause of overfitting is that the epochs are too large. According to the picture of loss function, epochs should be set between 2-3.

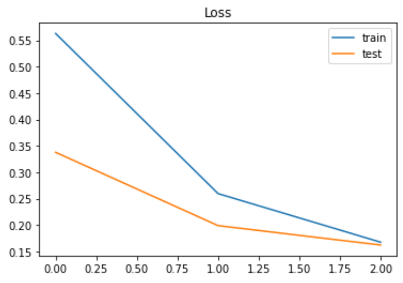
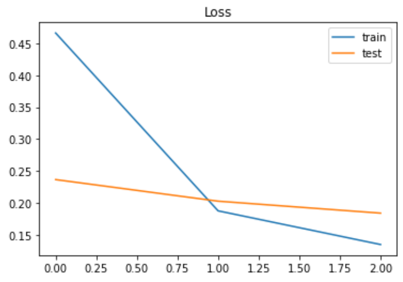
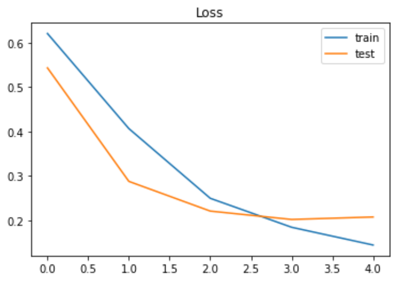
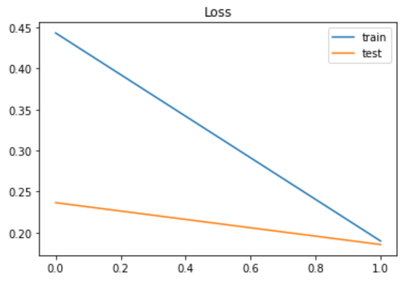
epochs	batch_size	accuracy	loss function
3	64	0.9314009547233582	
3	32	0.9420289993286133	
2	64	0.9285024404525757	
2	32	0.9236714839935303	

Table 4: Result

It seems that this idea is right, and the trends of val_loss and train_loss are roughly the same. So, the model with epochs=3 and batch_size=64 is a good model.

4. Conclusion

In this paper, i introduced the development history, advantages and disadvantages of neural networks and their corresponding solutions, especially LSTM. Through the experiment, we have roughly understood the principles and implementation process of LSTM. By comparing it to other natural language processing models, such as fastText, LSTM has a higher accuracy. In future work, we can also try to use algorithms such as GRU and LSTM-Autoencoder for spam identification. If we can find or develop a new algorithm of machine learning that can make it particularly good at identifying spam, it will be more convenient for people's lives.

5. Acknowledgments

Thank you to all the students and teachers who took this course with me for their comments and suggestions. Thanks to data provided by the University of Dortmund.

References

[1] callbacks. `https://keras-cn.readthedocs.io/en/latest/other/callbacks/`.

[2] Keras. `https://keras.io/`.

[3] Neural network. `https://en.wikipedia.org/wiki/Artificial_neural_network`.

[4] A quick introduction to neural networks. `https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/`.

[5] Understanding lstm networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

[6] Diederik P. Kingma; Jimmy Lei Ba ADAM. A method for stochastic optimization. 2015.

[7] Jeffrey L Elman. *Finding Structure in Time*, volume 14. 1990.

[8] Jürgen Hochreiter, Sepp; Schmidhuber. *Long Short-Term Memory*, volume 9. 1993.

[9] Warren; Walter Pitts McCulloch. *"A Logical Calculus of Ideas Immanent in Nervous Activity"*, volume 5. 1943.

[10] Geoffrey E.; Rumelhart David E Williams, Ronald J.; Hinton. *Learning representations by back-propagating errors*, volume 323. October 1986.