Mutual Fund Style Classification from Prospectus

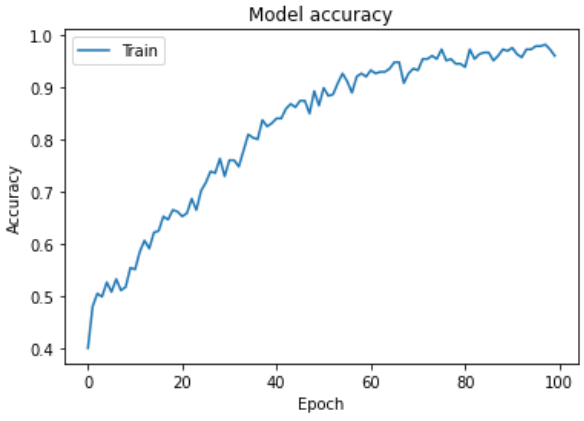
Xizhi Wang, Yuqing Liu, Haozhi Chen, Che Wang

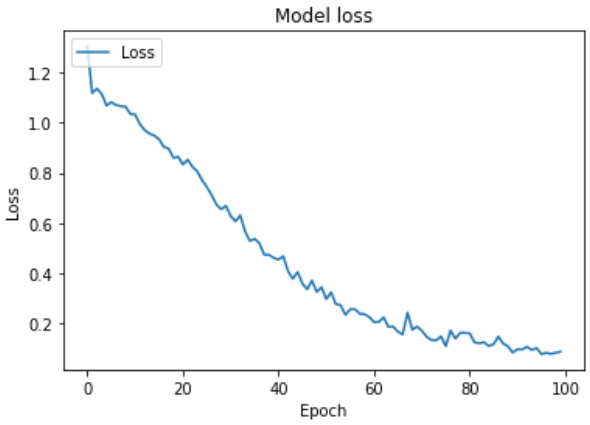
1. Executive Summary

The purpose of this paper is to discuss the application of mutual fund prospectus in determining the fund investment style out of 4 styles. With the two datasets provided, we managed to use a skip-gram model and built a word embedding dictionary. We further built knowledge bases and measured the distance between each summary to its knowledge base. Then we used a classification algorithm to predict the investment strategy for each fund.

2. Results and Discussion

2.1 the normal CNN model Accuracy and Loss:





As the epoch gets larger, the accuracy slowly goes above 0.9, and as the epoch gets larger, the loss of the model decreases to be lower than 0.1.

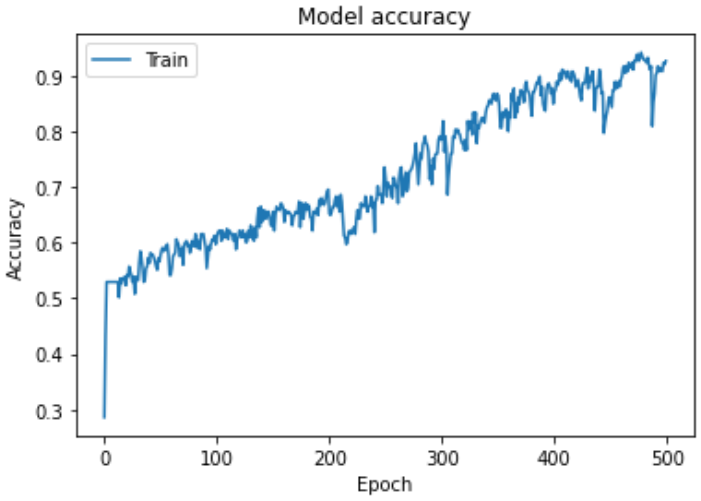
Accuracy score and classification report when using the model to predict the investment strategy of the test dataset:

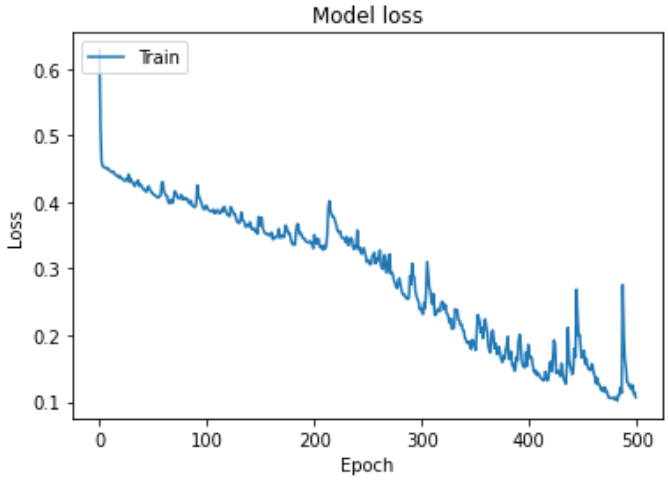
```
# print acc and report
print(accuracy_score(dummy_test_y,pred_y_CNN))


print(classification_report(dummy_test_y,pred_y_CNN))
```

```
0.6857142857142857
              precision    recall  f1-score   support

           0       0.71      0.74      0.73        27
           1       0.74      0.72      0.73        75
           2       0.56      0.59      0.58        37
           3       0.00      0.00      0.00         1

   micro avg       0.69      0.69      0.69       140
   macro avg       0.50      0.51      0.51       140
weighted avg       0.68      0.69      0.68       140
 samples avg       0.69      0.69      0.69       140
```

The accuracy of this model is 0.6857, we can notice from the classification report that the f1-score, recall, and precision for class 3(which is 'Long Short Funds(High risk)') are 0, which means the model failed to correctly predict this investment strategy.

2.2 RNN model Accuracy and Loss:





Accuracy score and classification report when using the model to predict the investment strategy of the test dataset:

```
# print acc and report
print(accuracy_score(dummy_test_y,pred_y_RNN))

print(classification_report(dummy_test_y,pred_y_RNN))
```

```
0.6785714285714286
              precision    recall  f1-score   support

           0       0.77      0.63      0.69        27
           1       0.76      0.71      0.73        75
           2       0.51      0.65      0.57        37
           3       1.00      1.00      1.00         1

   micro avg       0.68      0.68      0.68       140
   macro avg       0.76      0.75      0.75       140
weighted avg       0.70      0.68      0.68       140
 samples avg       0.68      0.68      0.68       140
```

The accuracy of this model is 0.6786, which is slightly lower than the first CNN model. The classification report shows that the f1-score, recall, and precision for class 3(which is 'Long Short Funds(Hight risk)') are 1, which means the model correctly predicted this investment strategy.

2.3 CNN model with different settings

We change the maxlen from 150 to 200. And we make class 3('Long Short Funds(Hight risk)') more obvious by changing the feature_train and feature_test.

```
feature_train[22] =feature_train[22]*0.1
feature_train[314] =feature_train[314]*0.1
feature_train[321]=feature_train[321]*0.1
feature_test[85] =feature_test[85]*0.1
```

Accuracy score and classification report when using the model to predict the investment strategy of the test dataset:

```
from sklearn.metrics import classification_report
print(classification_report(dummy_test_y,pred_y_CNN))
```

```
0.6357142857142857
              precision    recall  f1-score   support

           0       0.86      0.67      0.75        27
           1       0.75      0.59      0.66        75
           2       0.44      0.70      0.54        37
           3       1.00      1.00      1.00         1

   micro avg       0.64      0.64      0.64       140
   macro avg       0.76      0.74      0.74       140
weighted avg       0.69      0.64      0.65       140
 samples avg       0.64      0.64      0.64       140
```

As we can see, the accuracy score of this model is 0.6357, which is lower than the RNN model. However, according to the classification report, the precision, recall and f1-score of class 3 are 1, which means this model can correctly predict the 'Long Short Funds(Hight risk)' investment strategy of the test dataset.

In conclusion, the RNN model is the best model. Since it has the power to correctly predict the 'Long Short Funds(Hight risk)' investment strategy, it has a relatively higher accuracy score than the second CNN model.

---

3. Methodology

3.1 Procedure

We first processed the mutual fund summaries into a list of fund names and a list of summaries and saved them into a data frame. We then split the data into a training and testing set. We defined a stop words tokenizer to filter sentences in training set summaries, then we defined the parameters for skip-gram model. We then handle rare words by removing samples with less than pre-specified min_occurrence, giving unique id to words in the vocabulary. After pre-processing, we trained the skip-gram model and fitted the autoencoder with parameters. To build the knowledge base, we come up with a keywords list that contains the keyword that is related to the 4 investment strategies, we make sure that each keyword is also in the word2vec. Get the num_neighbors close

words of each keyword and combine them to create a knowledge base set. Then we use the match extraction method to produce the train_X and test_X. Use the column of 'Investment Strategy' to produce dummy_train_y and dummy_test_y both are an array of arrays of 4 dummy variables presenting 4 different investment strategies. Then we assign a unique index to each word in train_X and test_X to create the feature. After that, we create the embedding matrix. We use two CNN models and one RNN model to train our train set, and use these models to predict the test set's investment strategy, and report the performance of each model.

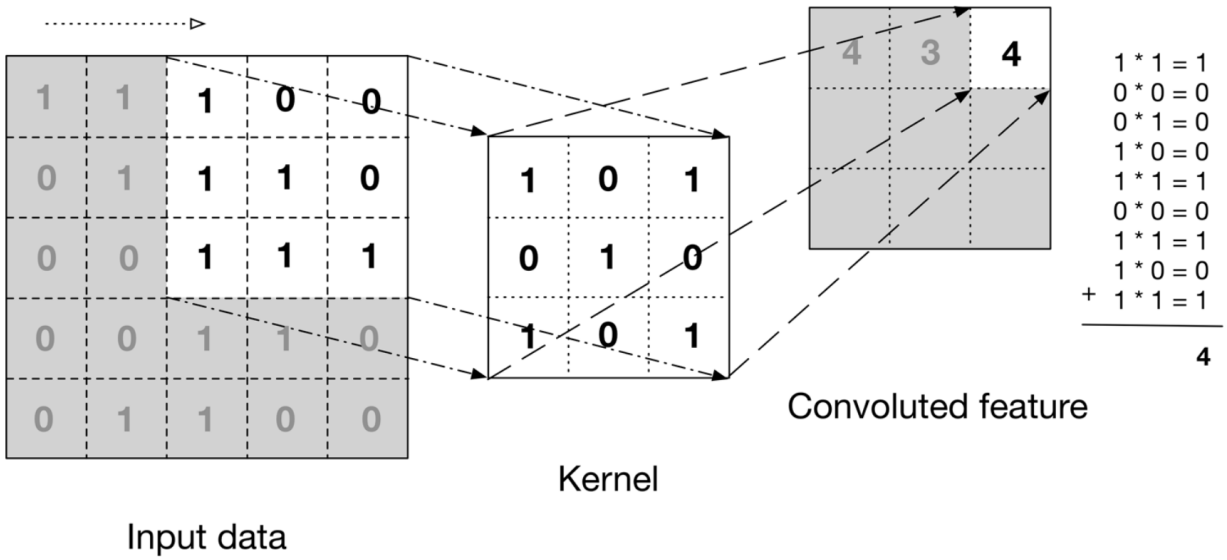
3.2 Convolutional Neural Networks

Introduction

In the past few decades, Deep Learning has proved to be a very powerful tool because of its ability to handle large amounts of data. The interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition. One of the most popular deep neural networks is Convolutional Neural Networks(CNN). CNN was first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc. The important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback for CNN at that period hence CNN was only limited to the postal sectors and it failed to enter the world of machine learning.

How CNN works

For simplicity, we will show you the following grayscale images to help understand how CNNs work.



We make the featured x in the train set and the dummy y in the train set to fit the CNN model with epochs and batch size being 100 and 32 respectively to build up:

```
CNN_history = CNN_model.fit(feature_train, dummy_train_y, epochs=100, batch_size=32)
```

Then we find out the training accuracy and the training loss value, in this case, we provide the following two plots to make our training result visible. After fitting the training set, we use it to make a prediction of our test set (prediction of the investment strategy).

3.3. Recurrent Neural Networks

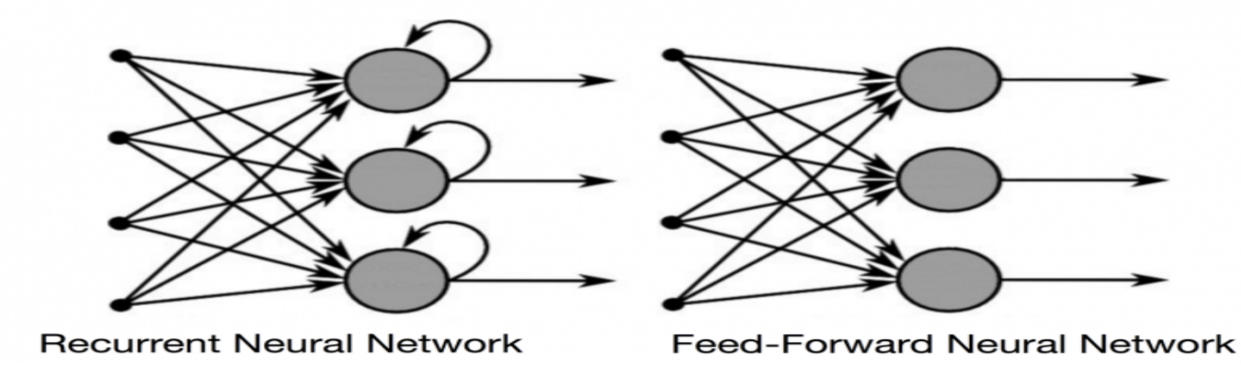
Introduction

RNNs are a powerful and robust type of neural network and belong to the most promising algorithms in use because it is the only one with internal memory.
Like many other deep learning algorithms, recurrent neural networks are relatively old. They were initially created in the 1980s, but they are widely used by people just in recent years. This is because of an increase in computational power along with the massive amounts of data that we now have to work with, and the invention of long short-term memory (LSTM) in the 1990s, which has really brought RNN to the foreground. Because of their internal memory, RNN can remember important things about the input they received, which helps to improve the precision in making predictions. This is why they're the preferred algorithm for sequential data like time

series, speech, text, financial data, audio, video, weather and much more. Recurrent neural networks can form a much deeper understanding of a sequence and its context compared to other algorithms.

How RNN works

The two images below illustrate the difference in information flow between an RNN and a feed-forward neural network.



In our project, we make the featured x in the train set and the dummy y in the train set to fit the RNN model with epochs and batch size being 500 and 32 respectively to build up:

```
RNN_history = RNN_model.fit(feature_train, dummy_train_y, epochs=500, batch_size=32)
```

Then making the prediction of the investment strategy by the same process as in CNN.

---

4. Appendix

Xing, H. (n.d.). NLP_Deep_Learning_Questrom_WS_03_2020.ipynb. Questromtools. Retrieved May 4, 2022, from https://questromtools.bu.edu/portal/site/SPRG22MF815D1E1/tool

CNN for deep learning: Convolutional Neural Networks. Analytics Vidhya. (2021, July 23). Retrieved May 4, 2022, from https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

Donges, N. (n.d.). *A guide to RNN: Understanding recurrent neural networks and LSTM Networks*. Built In. Retrieved May 4, 2022, from https://builtin.com/data-science/recurrent-neural-networks-and-lstm

---

5. Project Contribution

Task 1 – Task 3

1. Split the data into training and testing.

2. Following the NLP application in class, use the skip-gram model to build a word embedding dictionary from the mutual fund summaries in the training set.

3. Design a strategy to build knowledge bases associated with the aforementioned four investment strategies.

By Che Wang and Haozhi Chen

Task 4 – Task 5

4. Measure the distance of each summary to each knowledge base. Design a classification algorithm to predict the investment strategy of each fund.

5. Apply your classification algorithm to predict the investment strategy of each fund in the test data.

By Yuqing Liu and Xizhi Wang