Online News Popularity Prediction

Shuo Zhang

Reseach School of Computer Science, Australian National University,

2601 Canberra, AUSTRALIA

U6226993@anu.edu.au

Abstract. Due to the popularity of the Internet, online news has become an important tool for information sharing. Predicting the popularity of online news also became a hot topic since it could help editors design better news. The prediction can be treated as a binary classification problem where "popular" is one class and "unpopular" is the other one. Researchers have built many models to solve this problem including the neural network. In this paper, experiment on using bimodal distribution removal (Slade & Gedeon, 1993) to improve the performance of a neural network classifier has been done. But bimodal distribution removal showed no improvement. After that, using the evolutional algorithm(EA) to perform feature selection was tried. The best features being selected by evolutional algorithm improved the final score by 4%. The final system received 70% accuracy on test dataset which is 3% higher than the result of Fernandes, Vinagre and Cortez's research (2015).

Keywords: online news popularity, classification, neural network, bimodal distribution removal, evolutional algorithm, feature selection

1. Introduction

1.1 Background and Motivation

The Internet is an important tool for sharing messages. A recent survey has shown that around 50% of teenagers and adults in America choose to read online news in their daily life(Pew Research Center, 2018) in 2016. This percentage has increased a lot in the past few years. For reasons that more and more people read online news and editors want their news to be popular, it would be meaningful to build a system to predict whether a news will be popular or not. Such system can not only help editors find how they could improve their news but also can bring significant commercial value. Thus I choose the "online news popularity" data set for my experiment.

The online news popularity problem can be modeled as a binary classification problem. Which means taking a few features of the given news then predict whether it will be popular or not. There are two approaches in popularity prediction area: the first one uses features after the publication of the news and the second one doesn't. Using features after the publication could usually gain a high accuracy, but it is almost useless in the real life. Thus this paper will focus on the second one although the accuracy might be low.

Petrovic, Osborne and Lavrenko (2011) proposed a system to predict whether a tweet received retweets or not in 2011. They used some social features and characteristic of the content such as length, number of words and so on. Finally, their system achieved 47% F-1 score. Hensinger, Flaounas and Cristianini (2013) also tried to classify whether a tweet is appealing or not. They used SVM(support vector machine) with text features(e.g., keywords, BOW of the title) and other characteristics like publishing date to build their system. And they achieved 51% to 62% accuracy.

In this paper, a three-layer neural network classifier has been built and many methods have been tried to improve the performance of it. These methods are data normalization, bimodal distribution removal and evolutional algorithm.

1.2 Introduction to Evolutional algorithm

1.2.1 Essential components of natural selection theory

Evolutional algorithm was inspired by the theory of natural selection. It aims to improve the survive ability of individuals in a given environment. In the natural selection theory, individuals have their own chromosomes, these chromosomes will somehow determine their phenotypes in the environment. If the phenotype adapted the environment

well, then this individual will survive, otherwise not. For example, in the cold and freezing area, if a rabbit has a thick fur, then it can survive, but if it has a thin fur, it will die. The chromosome consists of genes. Individuals who survived in the environment can pass their genes to the next generation. In the above example, all the thick fur rabbits may pass their thick fur genes to the next generation. But the thin-fur genes will be reduced a lot because of the dead of thin-fur rabbits. After this procedure, the next generation will have more thick fur genes and thick fur rabbits, thus this generation will adapt the environment better. In order to create more phenotypes with limited genes. the crossover procedure appeared. Crossover means that before passing the chromosome to the child, the parents may exchange part of their genes. For example, if there are two rabbits, one of them is thin fur and run fast, the other one is thick fur and run slow. Then their child might be thick fur and run fast instead of simply copying the features of one of its parent. If the environment is cold as well as full of predators, the child who has thick fur and run fast can survive. With crossover, the whole generation has more probability to adapt to the environment. Another procedure to increase the diversity is the mutation, the mutation means each gene has a very low probability to change to an undetermined sequence. If a gene mutated, we might get a genotype (also phenotype) which never appeared before. For example, after mutating, if a rabbit obtained a gene which make its body stronger than the fox, then this rabbit is more likely to survive.

1.2.2 From natural selection theory to the evolutional algorithm

Now we talk about how we apply the theory of natural selection in the evolutional algorithm. The evolutional algorithm aims to find candidate solutions for a given problem, it is also an optimization process which improves the quality of the solutions step by step. In the natural selection example, the new generation of rabbits tends to adapt to the environment better and better. While in the evolutional algorithm, the new generation of solutions tends to solve the given problem better and better. The main components of an evolutional algorithm are: encoding, initialization, fitness function, selection operators and reproduction operators. The architecture of evolutional algorithm is shown in figure 1.1.

Encoding:

In our algorithm, the solution of the problem corresponds to the natural chromosome. The natural chromosome is a sequence of genes, while in the evolutional algorithm, we will encode each gene into a numerical representation then join them to form the artificial chromosome.

Initialization:

Initialization is used to initialize the chromosomes(solutions) of the first generation, we can process this step according to our prior knowledge of the problem. But most time we will randomly initialize the solutions and let them evolve generation by generation.

Fitness function:

Fitness is used to measure how well the solution solves our problem. Sometimes we also need to convert the genotype encoding into the phenotype encoding before calculating the fitness. The problem we need to solve corresponds to the environment in the natural selection. If the solution fits the problem well, the fitness function would return a high score, otherwise returning a low score. For example, in the rabbit example, the problem is “survive in the cold weather”, and the fitness function will give the solution “thick fur” a high score while the solution “thin fur” a low score.

Selection operator:

The selection operator is used to determine which individuals would survive while which one would be eliminated. There are many types of selection operators such as random selections, tournament Selection, rank-based selection and so on. Individuals with high fitness scores usually have more opportunities to survive while low score individuals tend to be eliminated. Just like the thick-fur rabbit survived while thin-fur rabbit died.

Reproduction operator:

The reproduction operator is used to produce chromosomes of the next generation. There are two operations to introduce diversities: crossover and mutation. Advantages of these two operations are the same as the natural example we mentioned before. The differences are: 1.we will do these operations on the numerical artificial chromosome. 2. We can do crossover on more than two parents.

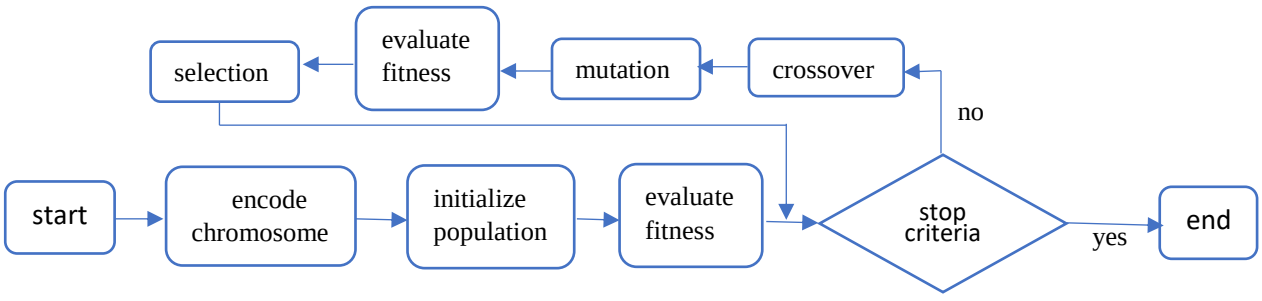


Figure 1.1 Architecture of evolutional algorithm

2. Dataset

The dataset is provided by Fernandes, Vinagre and Cortez(2015). They retrieved the news published by Mashable from the year 2013 to 2015. 39000 articles in total were downloaded. They have obtained 60 features and 1 target (number of shares) from these articles. Some of these features were extracted from HTML code. While several other natural language processing features were processed by the Latent Dirichlet Allocation (LDA) algorithm(Blei, Ng and Jordan, 2003). Parts of these features are shown in Table 1.1.

Feature	**type**	**Feature**	**type**	**Feature**	**type**
url	string	Num of keywords	number	Worst keyword	number
Num of images	number	Title subjectivity	ratio	Article categorgy	nominal
Num of images	number	Day of the week	nominal	Title sentiment	ratio

Table 1.1 part of features of the dataset (Fernandes, Vinagre and Cortez ,2015)

3. Method

3.1 Evaluation method

We use accuracy to evaluate the system since the task is binary classification problem. The formula of accuracy is:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{num\_correct\_prediction}{numz\_all\_samples} \quad (3.1)$$

(TP: true positive, TN: True negative, FP: False positive, FN: False negative)

It means the percentage of correct predictions among all test data, since we want more accurate predictions, thus the accuracy is a good evaluation method.

3.2 Processing the Data

The original data contains 60 features including url, rate of positive words, title sentiment polarity and so on. And 39797 samples is provided. All the features are numerical except url. Considering that url is not an important feature and there is no effective way to convert it from string to numerical, I delete this feature directly. The original target is the number of shares. In order to do the classification task, I adopted the method used by Fernandes, Vinagre and Cortez (2015), treating 1400 shares as the threshold, which means if the number of shares of a news was larger than 1400, then it was a popular news, otherwise unpopular. And I used number 1 to represent popular and 0 to represent unpopular in the system. Moreover, to compare my system with theirs, I separated the dataset as they did, 70% for training and 30% for testing.

3.3 Feature Scaling

Feature scaling is used to standardize the range of feature values. It can help the gradient descent converge faster and stable (Juszczak, Tax and Duin,2002). There are three kinds of scaling method: rescaling, mean normalization and standardization. Their mathematical formulas are showing below.

Rescaling:

$$\mathbf{x'} = \frac{\mathbf{x} - \mathbf{min}(\mathbf{x})}{\mathbf{max}(\mathbf{x}) - \mathbf{min}(\mathbf{x})}$$ where x is an original value, x' is the normalized value (3.2)

Mean normalization:

$$\mathbf{x'} = \frac{\mathbf{x} - \mathbf{mean}(\mathbf{x})}{\mathbf{max}(\mathbf{x}) - \mathbf{min}(\mathbf{x})}$$ where x is an original value, x' is the normalized value (3.3)

Standardization:

$$\mathbf{x'} = \frac{\mathbf{x} - \mathbf{mean}(\mathbf{x})}{\sigma}$$ where x is the original feature vector, σ is its standard deviation (3.4)

When I looked into the values of these features, some of them had the range around 0-2000 such as "the number of tokens of the news" while some other features only have range 0-10 like "the average token length". Huge difference of

these ranges inferred that feature scaling should be done to improve the performance of the system. I tried all these three methods above. The results were shown in the next part.

3.4 Building the Neural Network

The neural network has 59 neurons in the input layer(same as the number of features), 49 hidden neurons and 2 output neurons. When I looked into all the features, I found that some of them can be treated as the same type, for example, features “weekday_is_monday”, “weekday_is_tuesday” and so on can be regarded as the day of the week. On my own understanding, there’re 49 types of features in total. Thus I created 49 hidden neurons and hope these neurons can gather such types of information themselves. As for the loss function, since it was binary classification problem, cross-entropy could be ideal. I used sigmoid function as the active function, the network only contained one hidden layer, so there is no need to worry about gradient vanishing and gradient exploding problem.

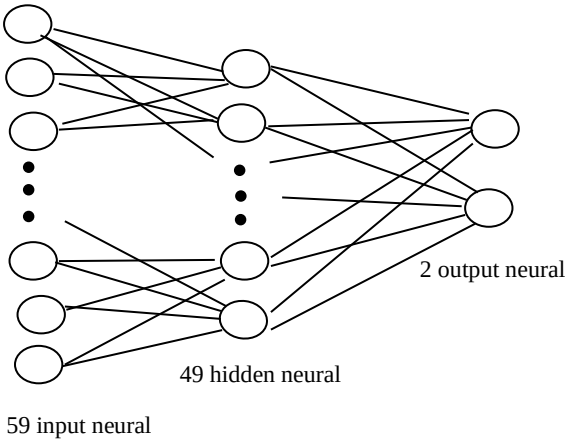


Figure 3.1 architecture of the neural network

3.5 Bimodal Distribution Removal

There are usually some outliers in the dataset. Remove these outliers can improve the generalization of the neural network. According to Geman, Bienenstock and Doursat(1992), the performance function in machine learning approaches and back-propagation of the neural network can be decomposed into bias part and variance part. With limited dataset, we may unable to decrease both the bias and the variance. Sometimes we can introduce bias to help our model fits the data well. Removing outliers belongs to this approach.

Slade and Gedeon(1993) proposed bimodal distribution removal method which can remove the outliers of real word data. As the neural network can identify outliers itself, Slade and Gedeon noticed that after around 200-500 epoch training, the error for all samples in the training dataset would form a bimodal distribution, the patterns within low peak were the data been learned well while patterns within high peak were outliers. Then we can find these outliers and remove them. The remaining dataset will improve the generalization of the neural network.

To realize this method, we need first calculate the mean of the errors($\bar{E}$) for all the training patterns Slade and Gedeon(1993), then choose the value of the parameter α to determine which pattern to remove. The formula is:

$$\text{if} \quad \text{error\_pattern\_i} >= \bar{E} + \alpha\sigma \quad (0 \le \alpha \le 1,\ \sigma \text{ is its standard deviation})$$
$$\text{then} \quad \text{remove the corresponding pattern.} \qquad (4)$$

We could repeat this approach every 50 epochs in order to let our system learn the new training set.

3.5 Evolutional algorithm

There are 59 features in total in the dataset. Some of them might be redundant and cause the neural network to perform worse. Thus we can apply evolutional algorithm to select more essential features from them. The whole method contains the following steps:

1. Encode the chromosome.

Since we have 59 features, each feature corresponding to one gene(the gene is encoded as 0 or 1, 1 represents using this feature, 0 represents not), the length of each chromosome will be 59. The format of the chromosome would be like [0 0 1 0 1 · · · 1 1 0 1]. The phenotype is the same as the genotype and we don’t need extra calculations.

2. Fitness function

Our aim is to obtain features which can gain the best test accuracy on the neural network. Thus the fitness function should return a higher score for a higher accuracy case. In this task, we will use the following formula:

score_solution_i = accuracy_i – min_accuracy (min_accuracy is the lowest accuracy among all the solutions in the same generation)

For each chromosome, we will train a new network and get the test accuracy. Since we will only use the selected features, thus the number of input neurons of the neural network will also be changed (the number of input neurons will equal to the number of selected features).

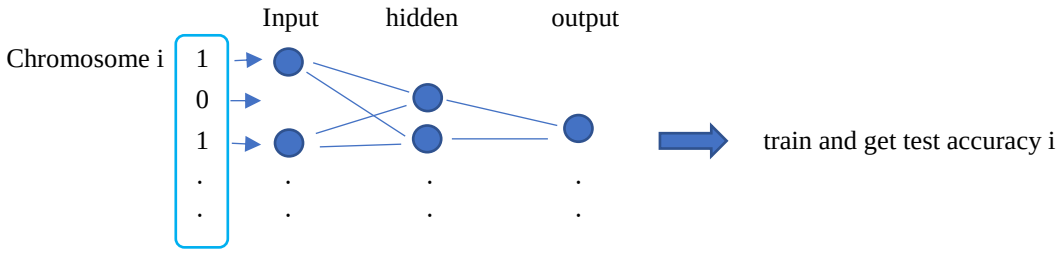


Figure 3.2. New network architecture

3. Selection operator

Before selecting the chromosome (solution), we need first calculate the probability of each solution being selected. We can use the following formula:

$$\text{Probability\_i} = \frac{\text{score\_solution\_i}}{\sum_{j=0}^{num\_solutions} score\_solution\_j}$$

With the above formula, all the probabilities will sum to one. Besides, the solution with high fitness score(accuracy) will have a high probability to be chosen(survive). Then we will generate n(number of population) solutions from this distribution. After this procedure, there will be more high accuracy solutions, fewer low accuracy solutions in the new generation.

4. Reproduction operator

As we mentioned in the introduction part, there are two reproduction operators: crossover and mutation. For the crossover, we will first set the crossover probability, if a chromosome is chosen to perform crossover, it will exchange part of its genes with another parent chromosome. The mutation operator is similar, we set the mutation rate first, if a gene is chosen to mutate, we will change it to 1 if it's originally 0 (change to 0 if it's originally 1).

3. Result and Discussion

3.1 Result and Discussion of all the Methods

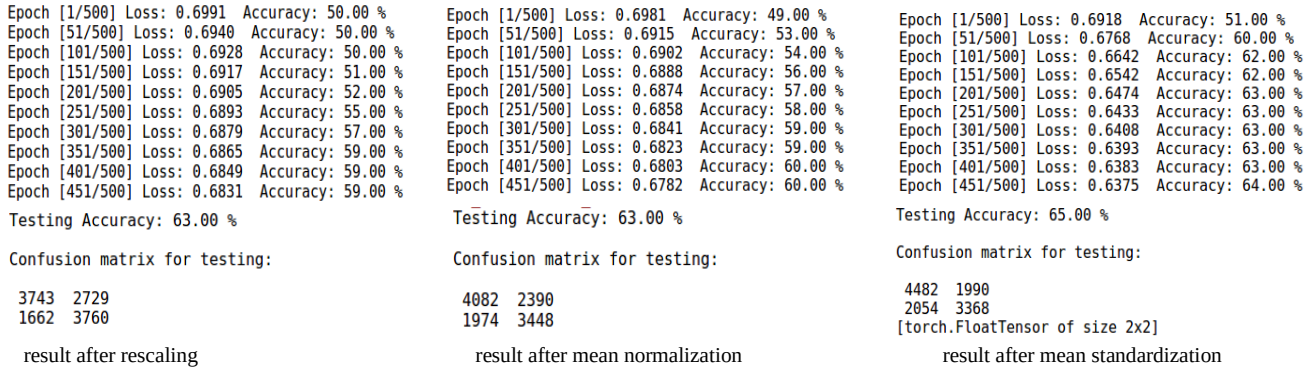
(1) Original:

The original system (without data normalization) only received 45% test accuracy which is even less than 50%. The result is shown in figure 3.1, the reason for such low accuracy might be that the huge difference of the feature ranges(0-2000 for "the number of tokens" while 0-10 for "the average token length") makes gradient doesn't work. Thus feature scaling should be done.

```
Epoch [1/500] Loss: 0.7309  Accuracy: 50.00 %
Epoch [51/500] Loss: 0.6934  Accuracy: 50.00 %
Epoch [101/500] Loss: 0.6933  Accuracy: 50.00 %
Epoch [151/500] Loss: 0.6931  Accuracy: 50.00 %
Epoch [201/500] Loss: 0.6931  Accuracy: 50.00 %   Testing Accuracy: 45.00 %
Epoch [251/500] Loss: 0.6930  Accuracy: 50.00 %
Epoch [301/500] Loss: 0.6929  Accuracy: 50.00 %   Confusion matrix for testing:
Epoch [351/500] Loss: 0.6929  Accuracy: 51.00 %
Epoch [401/500] Loss: 0.6928  Accuracy: 51.00 %       1  6471
Epoch [451/500] Loss: 0.6927  Accuracy: 51.00 %       5  5417
```

Figure 3.1 result of the original data

(2) Try feature scaling:
After trying three kinds of feature scaling methods(rescaling, mean normalization and standardization), the result shows that standardization achieves the best score of 65% accuracy. What's more, the loss in standardization method has been reduced much more than the other two methods. All the three methods have improved the accuracy a lot, which means feature scaling is really important for the neural network. Since standardization performed best, I chose to use it in all the following approaches.



result after rescaling | result after mean normalization | result after mean standardization

Figure 3.2 result after performing feature scaling

(3) Try bimodal distribution removal:
An intuitive reason for the low accuracy of 65% might be that the training data are quite noisy. Thus performing Bimodal Distribution Removal to remove the outliers should be a reasonable choice. As Slade and Gedeon(1993) addressed that the bimodal distribution usually formed after 200 epochs. Thus I started to remove outliers from the 300th epoch. For the formula of removing the pattern "if error_pattern_i >= $\bar{E}$ + α σ" I tried α to be 0.5, 0.8 and 1, but only 1 seemed suitable since other choices may remove too many patterns. And I remove outliers every 50 epochs in order to avoid removing too many patterns. The result was shown in figure 3.3.

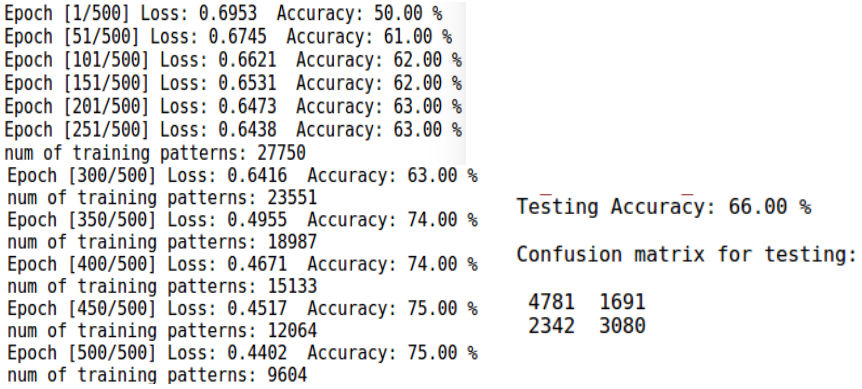


Figure 3.3 result after performing bimodal distribution removal

As we can see from the result. Although the loss has been reduced a lot and the training accuracy became much higher after performing bimodal distribution removal, the testing accuracy was only improved by 1%. This indicated that the bimodal distribution removal did not work as expected. The reason for the low training loss was that we removed the high loss patterns. And these high error patterns were also the patterns which our neural network predicted poorly, removing them would certainly increase the training accuracy. When we focusing on the remaining training patterns, we will find that we removed around 4000 patterns each time, such a large number means we removed too many for the dataset even we choose α to be 1. Considering the reason for this scenario, a reasonable explain would be that the loss of these patterns has not formed a bimodal distribution. The patterns been removed were mainly normal data rather than outliers.

(4) Evolutional algorithm + Bimodal distribution removal:
Since the bimodal removal itself cannot improve the performance effectively. We now need to try evolution algorithm to do the feature selection(there are 59 features being used, maybe some of them are redundant). The original dataset contains 39797 data. Due to the high computational complexity of the evolutional algorithm(we need to train a new network for each solution). It is almost impossible to run the algorithm on such a large dataset in a common CPU. So I

try to figure out whether this data set is redundant and whether the basic network can achieve the same outcome with only part of the data. I used only1000 data to do the experiment and it turned out that the test accuracy is exactly the same as using large data set. Changing the hidden layer into 2 neurons also showed no effect. Besides, using a larger learning rate(0.5) with fewer epochs(100) also achieved the same score as small learning rate(0.1) with more epochs(500). Since I changed the number of epochs, the bimodal distribution removal function should also be changed, since the learning rate is larger, the bimodal distribution would form early, so I choose to perform bimodal distribution removal at epoch 80, also in order not to remove so many patterns(as we mentioned before, bimodal distribution removal tends to remove too many patterns of this dataset), I set α to be 3, although it is larger than 1, it will not change the concept of bimodal distribution removal. With all these changes, the test accuracy is the same as before(around 65%, 66%). Since these changes will not affect the performance of the basic neural network. I applied them in the evolutional algorithm.

In the evolutional feature selection. I set the population size to be 20, the crossover rate to be 0.8,the mutation rate to be 0.002, and the number of generations to be 40. We will record the best solution in each generation and choose the one with the highest accuracy among all these best solutions. The accuracies of all solutions in some generations and the highest accuracy of each generation are shown below.

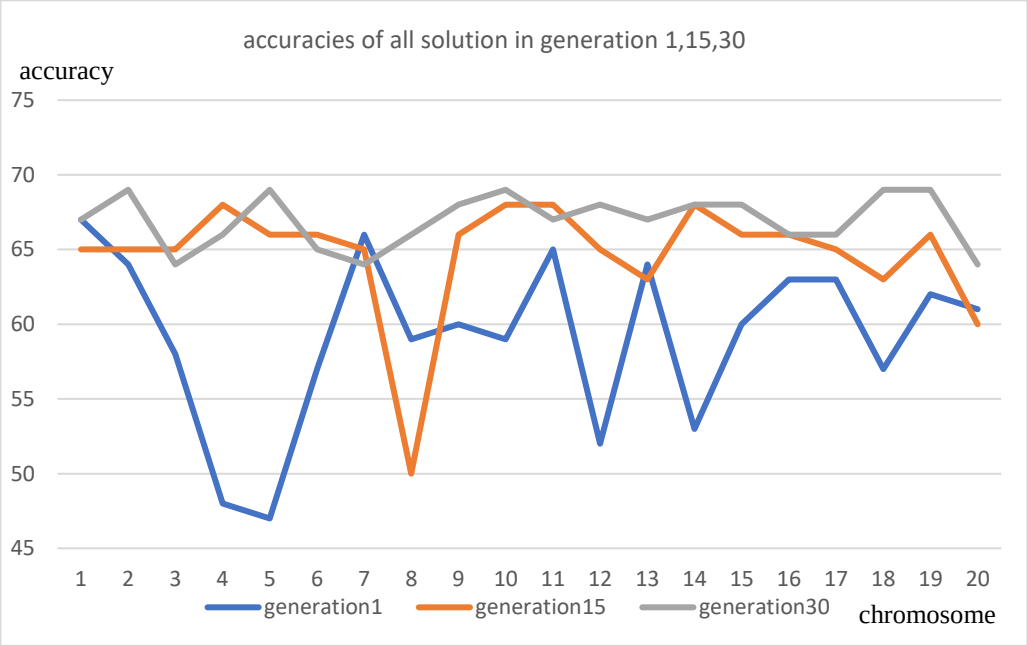


Figure 3.4 Accuracies of all solutions in generation 1,15,30

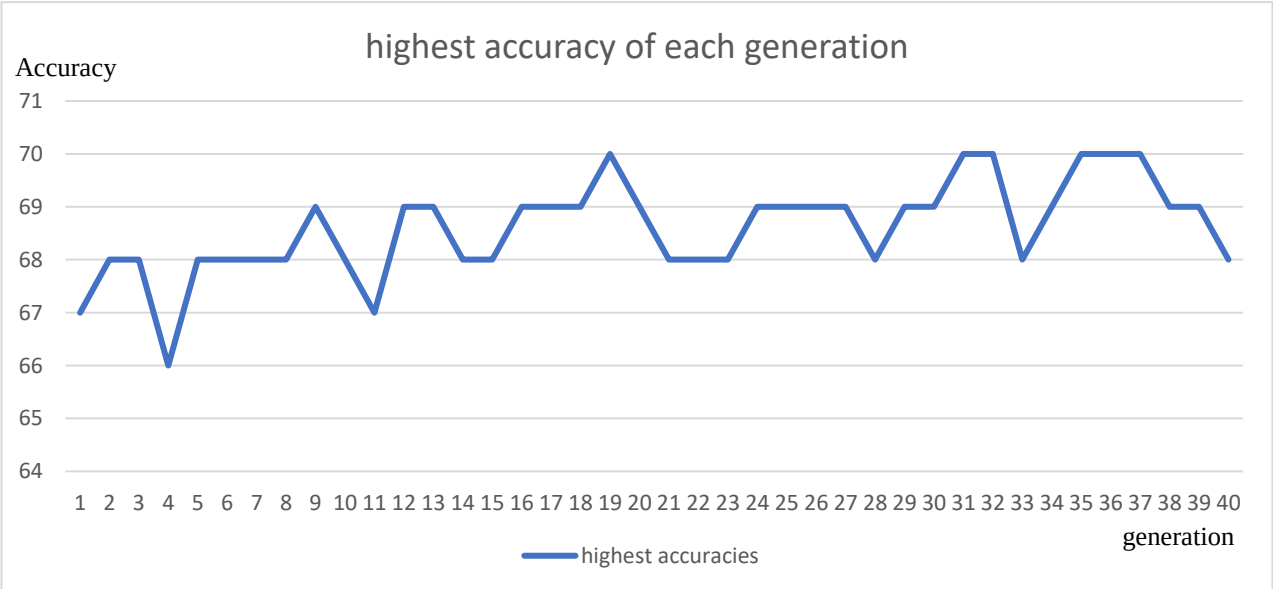


Figure 3.5 Highest accuracy of each generation

As we can see in Figure 3.4. the new generation tends to solve the problem better than before. At the first generation, the average accuracy of all the solutions is around 60%. At the 10th generation, the average accuracy is around 65% and at the 10th generation, the average accuracy has been increased to around 67%. But this is a tendency, which does not mean all the new generations will perform better than before, in this experiment, the average accuracy of the 40th generation is actually lower than the 30th generation. As for the highest accuracy in each generation, the evolutional algorithm is not as stable as we expected. The performance of the new generation is often worse than the previous generation. And the tendency that the highest accuracy was increased along time is also not so obvious. There are two possible reasons for this. The first one is that the crossover and mutation have changed a good solution. The second one is that the neural network itself is not stable, the test result may be a little different with different initializations. Even though the evolutional algorithm is not so stable, we can still get a better solution than the basic network using all the features. The best accuracy among all the generations is 70%, which is 4% percent higher than the score only use bimodal distribution removal. This indicates our evolutional feature selection did find the more essential features, so now we should look deeper into the chromosome which gained the high accuracy among all the generation. The chromosome is shown below

[1001011101101101111011011111100000000100111000111010110110 0]

Figure 3.5 Best chromosome among all the generation

Features been selected in this chromosome contains “Avg. keyword (avg. shares)”(index 27), “Avg. keyword (max. shares)”(index 26) , “Article category (Mashable data channel)”(index 13-18, except the bus channel, is not selected) and so on. And the day of the week (index 30-37) is not selected except “is weekend”(index 38). The author who provided the data set has ranked the importance of features according to their importance in the RF model (Fernandes, Vinagre, and Cortez, 2015), their result is shown below.

Feature	**Rank (#)**	**Feature**	**Rank (#)**
Avg. keyword (avg. shares)	0.0456 (1)	Closeness to top 1 LDA topic	0.0287 (11)
Avg. keyword (max. shares)	0.0389 (2)	Rate of unique non-stop words	0.0274 (12)
Closeness to top 3 LDA topic	0.0323 (3)	Article text subjectivity	0.0271 (13)
Article category (Mashable data channel)	0.0304 (4)	Rate of unique tokens words	0.0271 (14)
Min. shares of Mashable links	0.0297 (5)	Average token length	0.0271 (15)
Best keyword (avg. shares)	0.0294 (6)	Number of words	0.0263 (16)
Avg. shares of Mashable links	0.0294 (7)	Day of the week	0.0260 (18)
Closeness to top 2 LDA topic	0.0293 (8)	Number of words in the title	0.0161 (31)
Worst keyword (avg. shares)	0.0292 (9)	Number of images	0.0142 (34)
Closeness to top 5 LDA topic	0.0288 (10)	Number of videos	0.0082 (44)

Table 3.1. Ranking of features according to their importance in the RF model (Fernandes, Vinagre, and Cortez, 2015)

Comparing our selection with their result, we will find that features “Avg. keyword (avg. shares)”, “Avg. keyword (max. shares)”, “Article category (Mashable data channel)” which were selected by our evolutional algorithm have also been ranked high scores in their experiment. And the feature we abandoned “Day of the week” was ranked low scores. This means our evolutional feature selections did find the more essential features. With these essential features, the neural network can focus more on them by deleting unnecessary features and then achieve a better score.

Although we gained great result (70%) in this experiment, there are still some problems exist. The evolutional feature selection relies on the initialization a lot, sometimes it could select reasonable features and return a good score. But sometimes the whole process is unstable and could not give an effective solution. If most of the solutions in the first generation (initialized solutions) selected useless features, then the system might perform badly. This problem might be solved by increasing the number of population and the number of generations, but due to limited resources, we cannot do such a large amount of computations in our experiment.

3.2 Comparing with Other Approaches and Further Discussion

Fernandes, Vinagre and Cortez(2015) also worked on the same dataset for the binary classification problem (treating the pattern whose “num of shares” larger than 1400 as “popular” news). They have tried 5 methods: Random Forest(RF), Adaptive Boosting(AdaBoost), Support Vector Machine(SVM), K-Nearest Neighbors(KNN) and Naïve Bayes(NB).

Support Vector Machine(SVM) usually perform well in classification problem since it maximizes the margin between classes (Suykens and Vandewalle, 1999). The result of their experiments was shown in table 3.6.

Model	**Accuracy**
Random Forest(RF)	0.67
Adaptive Boosting(AdaBoost),	0.66
Support Vector Machine(SVM)	0.66
K-Nearest Neighbors(KNN)	0.62
Naïve Bayes(NB)	0.62

Table 3.6 result for Fernandes, Vinagre and Cortez(2015)'s approaches

Comparing their results with our approach. The best result (67% for Random Forest) is only 1% higher than our basic model with bimodal distribution removal. And with evolutional feature selection, we gained 70% accuracy which is 3% higher than their best score. Thus the performance of our system is good. However, we cannot say our system is definitely better than theirs since we only used part of the dataset in the evolutional feature selection. Besides, the evolutional feature selection requires much more computations.

4. Conclusion and Future Work

In this paper, we proposed a three-layer neural network to solve classification problem of online news popularity. Feature scaling, bimodal distribution removal and the evolutional algorithm have been tried to improve the performance of this system. Feature scaling has improved the testing accuracy for nearly 15% while bimodal distribution removal did not work effectively. Combining evolutional feature selection with bimodal distribution removal has further increased the score by 4%, the final system gained 70% accuracy which is 3% higher than the comparing approach.

However, the performance of the evolutional algorithm is unstable, it relies on the initialization a lot. With different initializations, the system may perform differently. If most of the solutions in the first generation (initialized solutions) selected useless features, then the result will be bad. Thus future works should focus on this problem. One way to solve this is to use a large number of population and more generations, but this will cost too much computation. Another way is finding a better initialization method instead of random initialization. We may first get some prior knowledge about these features then make plans according to them.

Reference

Pew ResearchCenter (2018). Pathway to News. retrieved from http://www.journalism.org/2016/07/07/pathways-to-news/

Tatar, A., de Amorim, M. D., Fdida, S., & Antoniadis, P. (2014). A survey on predicting the popularity of web content. Journal of Internet Services and Applications, 5(1), 8.

Petrovic, S., Osborne, M., & Lavrenko, V. (2011). RT to Win! Predicting Message Propagation in Twitter. ICWSM, 11, 586-589.

Hensinger, E., Flaounas, I., & Cristianini, N. (2013). Modelling and predicting news popularity. Pattern Analysis and Applications, 16(4), 623-635.

Fernandes, K., Vinagre, P., & Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. In Portuguese Conference on Artificial Intelligence (pp. 535-546). Springer, Cham.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

Juszczak, P., Tax, D., & Duin, R. P. (2002). Feature scaling in support vector data description. In Proc. ASCI (pp. 95-102). Citeseer.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.

Slade, P., & Gedeon, T. D. (1993). Bimodal distribution removal. In International Workshop on Artificial Neural Networks (pp. 249-254). Springer, Berlin, Heidelberg.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. Neural computation, 4(1), 1-58.

Čížek, P., & Víšek, J. Á. (2000). Least trimmed squares. In XploRe®—Application Guide (pp. 49-63). Springer, Berlin, Heidelberg.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. Neural processing letters, 9(3), 293-300.