Optimisation of the Artificial Neural Network for Snippet Length Classification

Shuqiang Jiang

Research School of Computer Science
Australian National University
Canberra, Australia
u6281911@anu.edu.au

Abstract. The snippet length of the search results on mobile devices is the same as that on desktops so far. Kim, Thomas, Sankaranarayana, Gedeon and Yoon (2017) believe that different snippet lengths will affect the user search behavior and user satisfaction on mobile devices, and prove this by statistical methods. This article will further explore the relation among snippet length, user search behavior and user satisfaction through the artificial neural networks. By applying the DecryptGISData technique, the K-Fold Cross-Validation, the batch normalization and the LSTM, this article achieve a classification model whose macro average precision, recall and f1 score are all over 0.75. This strongly proves that there is a mapping between snippet length and user search behavior and user satisfaction. This proof will promote the development of search engines, improve user experience and work efficiency on mobile devices.

Keywords: Neural networks · Mobile web searching · Classification

1 Introduction

Web searching is a common action in modern daily life. The search engine result pages (SERP) generally contains the title, URL, and snippet. The content in the snippets of mobile devices seems the same as desktops. However, the same snippet occupies more screens on mobile devices due to the display limitations. This result in that the SERP of mobile devices generally display 2 to 3 results while the desktops often shows 5 to 6 result on the SERP. Therefore, there is an assumption that the appropriate snippet length for mobile devices may be different from that for desktop computers.

Kim et al. (2017) conducted an experiment to study the effect of the snippet length for the user search behaviour and satisfaction on mobile device with three different snippet lengths and two different task types. They finally concluded that the long snippets will make users spend longer search time because they need read these snippets. In addition, users are not satisfied with long snippets. Although users may be accustomed to searching with typical snippets length, two to three lines of snippets can provide better service for mobile users.

This article will study deeply the the relationship between snippet length and user search behavior and user satisfaction based on the research and data from Kim et al.(2017). The research shows that the snippet length will affect the user 's searching time, eye fixation duration on title, url and snippets and user satisfaction. However, this conclusion is based on a simple statistical comparison, then is there a mapping between snippets length and user search behavior and user satisfaction? This article takes user search behavior and user satisfaction as input and then applys deep learning to do classification on snippet length to explore the mapping between them.

2 Method

This article take use of the DecryptGISData technique, which raised by Bustos and Gedeon (1995). This technique mainly consist of the pattern reduction, adding random noise and encoding, decoding and interpretation for the input and output data. In addition, this article designs three network models; One is an artificial neural network based on sigmoid activation function and batch normalization. Another is an artificial neural network based on relu activation function and dropout. The other is the LSTM network with L2 regularization. All of these models is trained with the batch size 32. Moreover, this article evaluate these model by k-Fold Cross-Validation and then choose the best model to train the final model.

2.1 Data Analysis and Selection

The data came from 24 participants and each participant completed twelve tasks where six informational tasks and six navigational tasks. Therefore, the raw dataset provide 288 patterns and 24 features.

The 'Snippet_length' represents the snippet length. The data type is String and there are three value: 'short' with one line, 'medium' with two or three lines, and 'long' with six to seven lines. This feature is chosen as the label

The 'Task_type' indicates the type of tasks including 'info' and 'nav' and the data type is String. This feature is important as user search behavior and user satisfaction run differently between informational tasks and navigational tasks. The 'Satisfaction' indicates the participant satisfaction using a 7-point Likert scale where 1 represents completely dissatisfied and 7 represents completely satisfied. The data type is integer and the value ranges from 1 to 7. The 'scroll' indicates whether the participant scroll the SERP. The data type is integer and the value ranges from 0 to 1. The 'Fixation_title', 'Fixation_URL', 'Fixation_snippet' and 'Fixation_total' are the eye fixation duration on title, URL, snippet and total and the data types are both float. The 'Log(F_title)', 'Log(F_URL)', 'Log(F_Snippet)' and 'Log(F_total)' are the logarithm value of those feature.

The 'Subject' represents the ID of the participants. The 'Task_num' is the ID of the tasks. This article attempt to find a general mapping between snippet

length and user search behavior and user satisfaction instead of for one particular participant or tasks. The 'StartTime' and 'Endtime_SERP' show the time of participant's start of the task and the time of participant's first click. The time interval is the value of 'Time to first click'. The 'Log(TTF)' is the logarithm value of 'Time to first click'. These features have the simlar contribution on the model with the eye fixation features. The 'Accuracy' represents whether the task is completed. These features may useful for analyzing the relation between snippet length and accuracy, but not for this article. The 'viewport' is an empty column. The 'Shown_T_num' is the order of the tasks shown to each participant which may be useful when apply the LSTM and RNN. However, in this stage the feature is not used as the data set is too small to train a complex network. The 'Clicked_rank' is the rank of search results which the participant clicked. This feature is related to the order of search results but not for this article. Li, Cheng, Wang, Morstatter, Trevino, Tang and Liu (2018) state that the unrelated features should be removed to reduce computational cost and accelerate the model convergence. Thus, these features are removed from data set.

In addition, there are alse 2880 lines date which consist of 288 sequence with 10 sequence length and each unit contains 8 feature. We only take use of the logarithm values Log10_Title, Log10_URL, Log10_Snippet and Log10_total. The other 4 features are the original value of them.

2.2 Data Imputation

There are missing values in 'Fixation_title', 'Fixation_URL', 'Fixation_snippet' and 'Fixation_total', so data dmputation is applied here. There are several methods for data dmputation such as deleting sample, filling with mean, median or mode, regression, hot deck imputation and K-Nearest Neighbor. The k-nearest neighbor method has been used in the surveys conducted of the US Bureau of Labor Statistics, the US Census Bureau and the Statistics Canada due to its simplicity and high accuracy (Zhang, 2012). Thus, this article mainly used the K-Nearest Neighbor. The detail implement is that using the KNNImputer class from scikit-learn.

2.3 Input Encoding

Firstly, the category features can be encode by integer encoding and one-hot encoding. However, the problem with integer encoding is that the category with higher integer become more important in the model which affects the results. Therefore, this article mainly choose one-hot encoding. The integer encoding is the control group to observing the effect of different encoding methods on the results. The category features which encoded as one-hot are 'Task_type' and 'scroll'. The 'Snippet_length' is the label, so its encoding method will not affect the classification network. This article encodes it as integer, because network of pytorch only accepts digital input. Although 'satisfaction' looks like category features, its actually is discrete data and the values indicate the different level of satisfaction, therefore the 'satisfaction' remain its represention as integer.

The continuous data with large range is often encoded as the logarithm value of itself, so that narrow down range. This is aimed to avoid features with large range influence the other features. For example, if some feature are ranged from 0 to 1 and a feature is ranged from 1000 to 10000, then this features may be contribute more to the model even though they have similar range of weight. In addition, this article also establish a new input feature 'ratio' which is the ratio between 'Fixation_title' and 'Fixation_snippet'. We believes that this feature can intuitively reflect the difference among snippet length, thereby accelerating the model convergence.

2.4 K-Flod Cross Validation

When the data set is small, cross validation can make full use the data to find suitable model parameters and prevent overfitting (Mosteller & Tukey, 1968)). In this article, there are only 288 lines data which is suitable to apply the K-Flod Cross Validation and the K is set as 12. Firstly, sampling randomly 60 lines as the validation set which not participating in the training process. Then the remained 228 lines is divided into 12 group which each has 19 values. By choosing one group as test set and the other groups as train set, there are 12 different train sets with corresponding test sets. Training all these combination can get 12 models. Finially, evaluating these models by the validation set and choosing the best model (Beacuse the Network A apply the batch normalization, the Section 4 Results and Discussion part only evaluate the experiments result of one mini-batch and the batch size is 32).

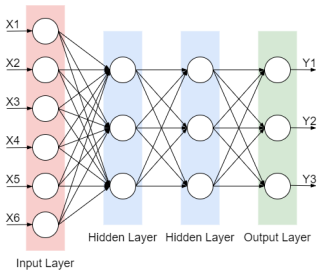

Fig. 1. the network A structure

2.5 Network A: sigmoid activation with batch normalization

As we know, the sigmoid activation function may cause a vanishing gradient problem.The root cause of this problem is that in the process of back propagation, due to the chain rule, the gradient of the front layers is affected by the rear layers. In addition, the gradient of the sigmoid function is very small when the x values outside the range from -5 to 5. Therefore, in the process backpropagating,

it is easy to appear that the gradients of some layer is very small, and then the gradients of the front layers which multiplied by these value also become very small. To solve this problem, there is a method called batch normalization (BN) which can solve this problem as well as avoid overfitting. The BN is nomalize the data for this batch before each activation layers, so that these data distribute around the area where the gradient of the sigmoid function is high. Thus, the vanishing gradient problem is controled. In addition, BN simplifies the process of parameter adjustment and reduces the requirements for data initialization. The inherent randomization from the minibatch statistics adds noise to the data and provides regularization so that effectively prevent the overfitting (Ioffe & Szegedy, 2015). The network structure is shown as fig.1.

2.6 Network B: relu activation with dropout

Apart from BN, using the relu activation function can also solve the vanishing gradient problem (Płaczek, & Płaczek, 2018). However, it cannot prevent overfitting, so the Dropout is added into the network. Dropout is an effective algorithm for training robust neural networks by avoiding the co-adaptation of feature detectors to prevent overfitting (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014). When forward propagation, stopping each neuron working with the probability p. This can improve the generalization of the model, as the model does not rely on certain local neuron and features. This work is similar to the pattern reduction.

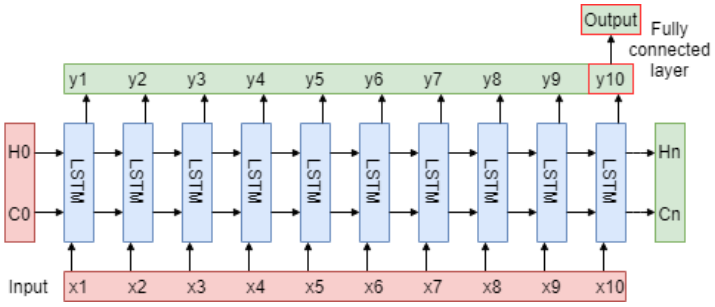

Fig. 2. the network C structure

2.7 Network C: LSTM with L2 regularization

In addition to the standard neural network, we also attempt to apply the LSTM network, because the dataset contains sequence features, and we want to take use of these features. RNN achieved an incredible success in processing sequence data, and the LSTM optimized RNN memory and forgetting functions so that achieved even better performance (Hochreiter & Schmidhuber, 1997).

When applying LSTM, we found that there is still an overfitting proble, so we apply the L2 regularization to prevent it. The main effect of the L2 norm is to prevent the model from overfitting and improve the generalization ability (Schmidhuber, 2015). The network structure is shown as fig.2.

2.8 The Loss Function and the Optimizer

The loss function adopts the Cross Entropy loss function which is the standard classification loss function. The optimizer is the torch.optim.Adam. Adam automatically adjust the learning rate to different parameters based on gradient statistics (Kingma & Ba, 2014). In addition, the updated learning rate can be limited to an approximate range according to the initial learning rate.

2.9 Evaluation methods for multi-class classification

This article evaluates the results with multiple criterias, including the Precision, Recall and F1-score of each classes, macro average and weighted average. When divide the prediction into negative and positive and divide the ground truth into true and false, the precision is the rate of true positive in all positive predictions. The recall is the rate of true positive in all true positive and false negative. The f1 score balance the precision and recall to evaluate all classes. Macro average and weighted average is used to caculate the average precision, recall and f1-score when the number of classes is more than two.

3 Results and Discussion

According to the DecryptGISData technique and the deep learning relevant knowledge, this article design 5 experiments which are running on simple network with raw data, simple network with encoded data, network A, network B and Network C. In addition, we apply the cross validation to envaluate and compare these model and then train the final model.

3.1 Experiment 1: Simple Network with Raw Data & Encoded Data

Motivation: The experiment 1 build a simple network as a control group without applying any technique. Then we build a simple network with encoded data. Therefore, the effect of data encoding can be found by comparing the result. In addition, when applying a new technique, its effect can be clearly compared with the simple network with encoded data.
Cross Validation result: The cross validation Result is shown in Table 1. The accuracy of the Simple Networks with Raw Data is only 0.38, while as we know, the random prediction on a three classes classification have 0.33 probability to win and this shows the poor performance. Compared with it, the simple networks with encoded data imporved 15.8% of the accuracy. This may due to that the features of encoded data are more obvious, and the problem of feature values

Network	Outline	accuracy (avg.)	precision (macro avg.)	recall (macro avg.)	f1-score (macro avg.)
Simple Network	Raw Data	0.38	0.39	0.40	0.35
Simple Network	Encoded Data	**0.44**	**0.46**	**0.44**	**0.42**

Table 1. The cross validation result of Experiment 1

with differeent ranges is solved. In addition, the raw data are more complex so that the network needs more parameters and the convergence of the model is slow. Therefore, data encoding can effectively improve the accuracy of a model.

3.2 Experiment 2: Network A

Motivation: This experiment builds the model by using the sigmoid activation function with batch normalization and trained with encoded data. Thus, the effect of Network A can be analysed by comparing the result with experiment 1.

Network	hidden nodes	accuracy (avg.)	precision (macro avg.)	recall (macro avg.)	f1-score (macro avg.)
Network A	3	0.48	0.50	0.50	0.47
Network A	4	**0.50**	**0.54**	**0.52**	**0.50**
Network A	5	0.43	0.46	0.45	0.44
Network A	6	0.49	0.49	0.50	0.48
Network A	7	0.44	0.45	0.46	0.44
Network A	8	0.39	0.38	0.40	0.38

Table 2. The cross validation result of Experiment 3

Cross Validation result: The accuracy of experiment 2 is higher than experiment 1 which shows the effect of dropout on overfitting. The accuracy 0.50 seems not so high but it is the average accuracy of the 12 models. As our dataset is small, some test set may not balance and result in unstable performance. This means there are some models achieve 0.65 accuracy while other models only have 0.3 accuracy and pull down the average accuracy. According to the Table 2, the relatively best hyperparameters is 4 hidden nodes.

3.3 Experiment 3: Network B

Motivation: This experiment builds the model by using the relu activation function with dropout and trained with encoded data. Thus, the effect of Network B can be found by comparing the result with experiment 1 and 2.

Network	hidden nodes	dropout prob.	accuracy (avg.)	precision (macro avg.)	recall (macro avg.)	f1-score (macro avg.)
Network B	**3**	**0.1**	**0.50**	**0.55**	**0.50**	**0.48**
Network B	3	0.2	0.44	0.48	0.45	0.39
Network B	3	0.3	0.40	0.40	0.41	0.37
Network B	4	0.1	0.43	0.45	0.45	0.40
Network B	4	0.2	0.48	0.53	0.49	0.45
Network B	4	0.3	0.46	0.49	0.45	0.42

Table 3. The cross validation result of Experiment 2

Cross Validation result: The accuracy of experiment 3 is similar to experiment 2. This may due to that the inherent randomization generated by mini-batch statistics can be used as a regularizer which making BN an alternative to dropout for preventing overfitting (Shen, Tian, Liu, Xu, & Tao, 2018). According to the Table 3, the relatively best hyperparameters are 3 hidden nodes with 10% probability dropout.

3.4 Experiment 4: Network C

Motivation: This experiment builds the model by using the LSTM network with L2 regularization and trained with encoded data. Thus, the effect of Network C can be found by comparing the result with experiment 1 to 3.

Network	hidden layer	hidden node	accuracy (avg.)	precision (macro avg.)	recall (macro avg.)	f1-score (macro avg.)
Network C	1	10	0.46	0.46	0.48	0.45
Network C	**1**	**12**	**0.48**	**0.50**	**0.47**	**0.46**
Network C	1	14	0.44	0.41	0.47	0.41
Network C	1	16	0.46	0.45	0.48	0.44
Network C	1	18	0.43	0.42	0.43	0.40
Network C	2	10	0.30	0.10	0.33	0.15

Table 4. The cross validation result of Experiment 1

Cross Validation result: The accuracy of experiment 4 is lower to experiment 3. This may due to that we donot have enough data to train the LSTM model. When we use a simple LSTM structure, it is underfitting and when we use the LSTM with complex structure, it is still underfitting as the dataset cannot support such many parameters. In addition, although the sequence length is 10, there are only about 1 to 4 lines contains value and the others are all zero for each sequence. This means over 2/3 data are just zero. Thus, it's hard to train

the lstm model on such a small and lossing value dataset. According to the Table 3, the relatively best hyperparameters are 1 hidden layer and 12 hidden nodes.

3.5 Result of the Final Model

Network	accuracy (avg.)	precision (macro avg.)	recall (macro avg.)	f1-score (macro avg.)
Simple Network	0.44	0.46	0.44	0.42
Network A	**0.50**	0.54	**0.52**	**0.50**
Network B	0.50	**0.55**	0.50	0.48
Network C	0.48	0.50	0.47	0.46

Table 5. The cross validation of each model with the relatively best parameters

Comparing the cross validation result of each model with best hyperparameters, we find the network A achieve the best performance. Thus, we train the final model based on the sigmoid activation function with batch normalization.

	precision	recall	f1-score	support
class short	0.64	0.90	0.75	10
class medium	0.78	0.64	0.70	11
class long	0.89	0.73	0.80	11
accuracy			0.75	32
macro avg	0.77	0.75	0.75	32
weighted avg	0.77	0.75	0.75	32

Fig. 3. The Evaluation result of the final model

The evaluation result of the final model is shown in Fig. 3. The accuracy is much higher than the cross validation result as we discussed above. In addition, the macro average precision, recall and f1 score are all over 0.75 which strongly indicate and prove that there is a mapping between snippet length and user search behavior and user satisfaction.

4 Conclusion and Future Work

This article explore the relation between snippet length and user search behavior and user satisfaction based on the research of Kim et al.(2017). By applying the DecryptGISData technique, the Cross-Validation and the batch normalization, this article achieve a classification model whose macro average precision,

recall and f1 score are all over 0.75. This proves that there is a mapping between snippet length and user search behavior and user satisfaction. In addition, by desigining experiments and comparing the results, this article also indicates the positive effect of DecryptGISData technique, the k-Fold Cross-Validation, Dropout, the batch normalization and the LSTM on artificial neural network.

4.1 Future Work

On the premise of avoiding from overfitting, it will be very difficult to continue to increse the accuracy of the model on such a small data set. Therefore, the next stage of work will mainly focus on the data collection. After collecting sufficient data, the author will attempt to deepen the network layers to improve the accuracy of the model.

References

1. Bustos R.A., Gedeon T.D. (1995) Decrypting Neural Network Data: A Gis Case Study. In: Artificial Neural Nets and Genetic Algorithms. Springer, Vienna
2. Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Comput. 9, 8 (November 15, 1997), 1735–1780.https://doi.org/10.1162/neco.1997.9.8.1735
3. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
4. Kim, J., Thomas, P., Sankaranarayana, R., Gedeon, T., & Yoon, H. (2017). What snippet size is needed in mobile web search? Paper presented at the 97-106. https://doi.org/10.1145/3020165.3020173
5. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
6. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R., Tang, J., & Liu, H. (2018). Feature selection: A data perspective. ACM Computing Surveys (CSUR), 50(6), 1-45. https://doi.org/10.1145/3136625
7. Mosteller, F. & Tukey, J. W. (1968). Data analysis, including statistics. In G. Lindzey & E. Aronson (eds.), Handbook of Social Psychology, Vol. 2, : Addison-Wesley.
8. Płaczek, S., & Płaczek, A. (2018). Learning algorithm analysis for deep neural network with ReLu activation functions. ITM Web of Conferences, 19, 1009. https://doi.org/10.1051/itmconf/20181901009
9. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.
10. Shen, X., Tian, X., Liu, T., Xu, F., & Tao, D. (2018). Continuous dropout. IEEE Transactions on Neural Networks and Learning Systems, 29(9), 3926-3937. https://doi.org/10.1109/TNNLS.2017.2750679
11. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). "Dropout: A simple way to prevent neural networks from overfitting", 15(1), 1929-1958.
12. Zhang, S. (2012). Nearest neighbor selection for iteratively kNN imputation. The Journal of Systems & Software, 85(11), 2541-2552. https://doi.org/10.1016/j.jss.2012.05.073