Classifying Target Page Position by a Neural Network with Explanation Mechanisms and Long Short-Term Memory RNNs

Zhouyang Jiang

Research School of Computer Science,
Australian National University
U6258659@anu.edu.au

Abstract. When using the search engine, the position of the target page may influence users' satisfaction, accuracy and efficiency. Our findings in predicting and classifying the target page position show that the users' satisfaction, accuracy and the time for them spending on searching are related to the position of the target page. When the relevant result is located at the front of the page, it will cost users less time to get the target result and the accuracy and satisfaction of the user are also higher. When building the neural network to do the classification, an explanation mechanism (Characteristic Input Method) and Long Short-Term Memory RNNs (LSTMs) are used to improve the neural network. From the results, it can be found that LSTMs can improve the performance of the neural networks. However, the explanation mechanism cannot improve the accuracy of the neural network in this case.

Keywords: Neural Network, Characteristic Input Method, Explanation Mechanisms, Search Engine, Classification, LSTMs, RNN

1 Introduction

Nowadays, the target page position of the search engine results pages (SERPs) become important for the search engine and its users. The reason is that the position of the target page will be related to users' satisfaction, accuracy and efficiency. Ghose [1] found that the rank of the results in the search engine will influence users' experience and the bidding rank for the searching results is common in search engines. Therefore, discovering the relationship between the target page position and users' experience is necessary.

To discover the relationship between the target page position and users' experience, in this report, the dataset of eyegaze-search3 from the research of Kim [2] was chosen. Here is a sample of the raw data:

Type	Target Position	Subject	Time to first click	Time to right click	Total time on SERPs	Task completion duration	Accuracy	Satisfaction	task Start	1st Click	Right Click	Task end	Scroll	Time with wrong WebDoc
H	1	1	8.189	8.189	8.189	15.69	1	5	173.11	181.299	181.299	188.8	0	0
H	3	1	36.195	36.195	36.195	59.595	1	3	90.405	126.6	126.6	150	1	0
H	5	1	63.374	63.374	63.374	105.789	1	4	544.611	607.985	607.985	650.4	1	0
H	6	1	21.18	90.693	62.88	101.993	0	3	323.807	344.987	414.5	425.8	0	27.813
H	8	1	7.982	42.195	42.195	54.695	0	3	461.005	468.987	503.2	515.7	0	0
H	10	1	44.794	44.794	44.794	56.994	1	3	208.306	253.1	253.1	265.3	1	0
H	1	2	34.546	34.546	34.546	47.148	1	7	1468.852	1503.398	1503.398	1516	1	0
H	3	2	91.754	91.754	91.754	104.989	1	7	2148.011	2239.765	2239.765	2253	1	0

Fig. 1. A sample of the raw data

From the raw data in Fig. 1, it can be found that the dataset shows the information of searching results, searching experience, and searching time.

In this report, the hypothesis of the research is an extension of the hypothesis H1 in Kim's research [2], which is that when the relevant result is located at the front of the page, it will cost users less time to get the target result and the accuracy and satisfaction of the user are also higher. So, in this study, to test the relationship, a neural network for classifying the target position in the result page from users' experience was built in PyTorch. And explanation mechanisms in Gedeon's research [3] and Long Short-Term Memory RNNs (LSTMs) were used to help the neural network do the classification task. The comparison was also made between the classification results before and after the explanation mechanisms and LSTMs were added.

2 Method

2.1 Data Pre-processing and Preparing

The first step is preparing and pre-processing the raw data. The data was loaded into data frame. Therefore, the data was preprocessed inside the data frame. When looking into the raw data, it can be found that there are several useless attributes and duplicated attributes all representing searching time. Therefore, after loading data, pre-processing data is necessary. After removing the unused attributes, it is Type, Target Position, Total Time on SERPs, Accuracy and Satisfaction that are finally kept where Target Position is the output value and others are the inputs. Type values are 'H' and 'V' where 'H' stands for horizontal control type and 'V' stands for vertical control type. Therefore, Type is a categoric data which can be normalized into binary numbers as 0 (H) and 1 (V). The reason why Type is chosen as one of the inputs is that Type is related to users' visual experience and the arrangement of the Target Position which is the output. As for the Target Position, as the output, it is an integer data of 6 values: 1, 3, 5, 6, 8, 10 which represents the position of the target page in the search engine results pages. Here is the image from Kim's study [2] that describes which location that each number stands for, with the horizontal control type and some additional annotations are added:

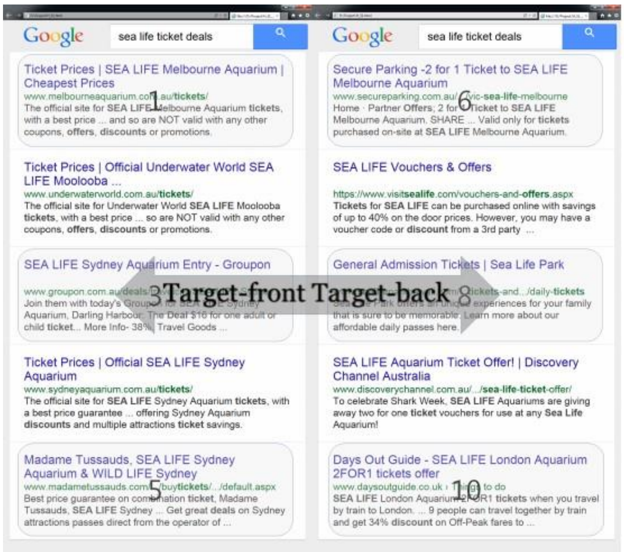

Fig. 2. Example of two subsequent pages of a SERP with the horizontal control type. 1, 3, 5 are regarded as Target-front. 2, 4, 6 are regarded as Target-back

From the image, it can be found that position numbers 1, 3, 5 can be regarded as target-front while 6, 8, 10 can be regarded as target-back. Therefore, similar to Type, Target Position can also be turned into 2 classes represented by binary numbers as 0 (target-front) and 1 (target-back). The other three attributes: Total time on SERPs, Accuracy and Satisfaction remain the same format as the ones in raw data, which shows the status information of time, accuracy and satisfaction in the hypothesis.

The next step is to divide the data into train data and the test data. In this case, train data and test data are divided randomly. 75% of the data are train data and 25% of the data are test data. The reason why 25% of the data are test data instead of 20% is that the size of this dataset is 288 which is small so that if only 20% of data were used, the test output will be unstable because of the outliers.

2.2 Neural Network Model Structure

After preparing the data, the next step is to define the neural network structure. As mentioned above, the classification target has 2 different classes: Target-front and Target-back. Therefore, a binary neural network for classification tasks was built. For the number of the inputs is 4, so there are 4 input neurons. And because it is a binary classification task, the number of the output neurons is 2. The learning rate is set to 0.01. Several different values of learning rate have been tried from 0.00001 to 0.1. And with several experiments, when learning rate is 0.01, the accuracy of the classification results is the highest and most stable. Here is the table that shows the accuracy of different learning rate for both normalized data and non-normalized data:

Table 1. Accuracy table with different learning rate while the number of hidden neurons is 10 and the number of the epochs is 500

Learning Rate	Testing Accuracy
0.00001	47.37%
0.0001	54.17%
0.001	58.57%
0.01	64.71%
0.1	60.49%

And the number of the hidden neurons is 10 and the number of the training epochs chosen is 500. The reason why the numbers of hidden neurons and the epochs are 10 and 500 is to avoid overfitting and underfitting. Here is the table that records the accuracy of different combinations of these parameters:

Table 2. Accuracy table with different number of the hidden neurons and the training epochs

Parameter	Training Accuracy	Testing Accuracy
hidden_neurons=5 num_epochs=100	52.97%	50.72%
hidden_neurons=10 num_epochs=500	63.01%	60.87%
hidden_neurons=15 num_epochs=2000	81.28%	52.17%

So, it can be found that, when hidden_neurons=5, num_epochs=100, the neural network is underfitting and when hidden_neurons=15, num_epochs=2000, it is overfitting. Therefore, 10 and 500 are suitable numbers for the hidden neurons and the epochs.

2.3 Performance Evaluation Techniques

The next step is to define the measures to calculate the performance of the neural network and predictions. As shown in the table above, accuracy is chosen as one of the techniques to measure the performance. The reason why accuracy can determine the performance is that it is a classification task, so that the prediction can be easily known whether it is correct by comparing with the actual targets. Therefore, the accuracy can show the percentage of the total data which has been correctly classified. One of the other reasons that accuracy was chosen instead of other techniques such as precision and recall is that the distributions of target classes are equally common. As mentioned above, the target classes are target-front and target-back. The probabilities of the target position located in these two classes are same which is 50%. To show the details of the distributions of the target positions, here is the table that shows the numbers of each targets class:

Table 3. Numbers of each target class

Target Position	Amount	Proportion
Target_front (1, 3, 5)	144	50%
Target_back (6, 8, 10)	144	50%
Total (1, 3, 5, 6, 8, 10)	288	100%

From the table, it can be found that the proportion of datapoints with each class is the same. So that accuracy can evaluate the performance of this binary classifier.

Another technique chosen to measure the performance is the number of the epochs to let the training accuracy achieve 75%. Although when training accuracy achieves 75% will cause the overfitting issue for the test data in this case, it can show the learning speeds for different networks. The number of the epochs will be used to compare the converge speeds between the original neural network and LSTMs.

2.4 Explanation Mechanisms

After the neural network has been implemented, the last step is to add explanation mechanisms to help the neural network do the classification. From Gedeon's paper [3], it can be found that the first step is to liken the input pattern to the characteristic input patterns and find the important ones. In this case, there are only 2 numeric inputs: Total Time on

SERPs and Satisfaction. To determine the importance of these two inputs, from Turner's research [4], the rate of change to the output should be considered, which means the gradient with respect to the input. The reason why the rate of change of the output neuron is related to the input neuron is that in Yoda's paper [5], it says that the output neuron's rate of change can be calculated by using chain rule of differentiation. Therefore, to find the gradients in tensor, autograd package in PyTorch can be used. Here is a sample of the gradients with respect to the Total Time on SERPs and Satisfaction in train data:

Table 4. Gradients with respect to the Total Time on SERPs and Satisfaction in train data

Gradient with respect to the Total Time	Gradient with respect to Satisfaction:
1.2036e-02	-2.9257e-02
5.3703e-03	-2.3288e-02
2.9608e-03	-8.8893e-03
7.9060e-03	4.4340e-01
3.4597e-03	-1.4777e-02
-5.3779e-03	2.1834e-02
1.7122e-02	-1.5998e-02

From the gradients, it can be found that most of the time, gradient with respect to the Total Time is greater than 0 and shows a positive peak while gradient with respect to Satisfaction is less than 0 and shows a large negative peak. Therefore, for the Characteristic pattern: ON target-back, Total Time is a characteristic ON input and Satisfaction is a characteristic OFF input. According to Gedeon's paper [3], after discovering the patterns, the next step is producing the rules from there inputs. In Turner's research [4], for the inputs showing a positive peak in the rate of change, the input value which can turn the gradient to 0 is considered as the boundary of the rule and the rule will be that the input should be greater than the boundary to turn the output on. Contrarily, for the inputs showing a negative peak in the rate of change, the input value which can turn the gradient to 0 are considered as the boundary of the rule and the rule will be that the input should be less than the boundary to turn the output on. So next step is to find the input values where the gradient is 0. However, autograd package in PyTorch cannot directly output the values according to gradients. To get the values approximately, the input value whose gradient is closest to 0 is considered as the boundary. In another word, the gradient who has the smallest absolute value shows the input can be regarded as the boundary of rules. After finding the rules of Total Time on SERPs and Satisfaction, these rules should be combined by the operator into an IF-THEN rule. ON target-back is a characteristic ON pattern so that according to Turner's procedures [4], for a characteristic ON pattern, inputs with a large peak in gradient should be combined by the conjunction operator. Therefore, the rules of Total Time on SERPs and Satisfaction are connected by the conjunction operator. Here is the example of rules generated:

Table 5. The generated rule

Characteristic pattern	Rule Set
ON target-back	(Total Time > 81) ∧ (Satisfaction < 5)

For train data is formed randomly, the boundaries generated vary depending on the train data. After the rule is generated, the last step is using this rule to predict the next most likely output. The reason why the rules are used directly is that from Gallant [6], it shows that explanation mechanisms are based on the neuron activation values which is inferred by the neural network. So, the classification done by the rule is quite straightforward. This rule in Table 5 means when the total time is larger than 81 and the satisfaction is less than 5, the target position will be classified to target-back class. It can be found from the generated rules, there is a negative correlation between target position and the satisfaction and a positive correlation between target postion and the total time, which can prove the hypothesis is true. Here is the example that shows that the neural network result and the way to calculate the explanation facility result:

Table 6. The neural network result and the explanation facility result

Row Number	209
Network Output	Target-back
Important Inputs	Total time: 93.074, Satisfaction: 4
Satisfied Rule Set	(Total Time > 81) ∧ (Satisfaction < 5)
Next Most Likely Output	Target-back

2.5 LSTMs

Another technique used to improve the original neural network is Long Short-Term Memory RNNs (LSTMs). For an RNN can be unrolled through time create a standard neural network with arbitrary input size, so that, different from the input size in the original neural network, the input size of LSTM will no longer be 4. Instead, the input with a size of 4 will be split into 4 single inputs with a size of 1 and the neural network will run 4 times recurrently to calculate the results. Therefore, the input size of LSTM will be 1 in torch.nn.LSTM(). And to record the total size of a whole input, another variable called time_step will be used. It means RNN will consider every 4 steps as a group and each step will have an input whose size is 1. And different from the original neural network, the number of the epochs will be changed into 200. It is because of the convergence speed which will be discussed in Results and Discussion section. And here is the table that records the accuracy of different number of epochs:

Table 7. The accuracy of different number of epochs

Parameter	Training Accuracy	Testing Accuracy
num_epochs=100	61.09%	59.70%
num_epochs=200	69.86%	67.09%
num_epochs=500	84.40%	57.14%

So, it can be found that, when num_epochs=100, the neural network is underfitting and when num_epochs=500, it is overfitting. Therefore, 200 is the suitable number for the epochs in this LSTMs.

3 Results and Discussion

3.1 Accuracy

Using the techniques mentioned above, the test data can be classified in 3 different ways. The first way is using the normal neural network. Second is the explanation mechanisms. And the last one is using LSTMs. The results of the testing accuracy for three techniques is shown:

Table 8. The testing accuracy for three techniques

Technique	Testing Accuracy
Original neural network	61.76%
Explanation mechanisms	57.35%
LSTMs	66.18%

Because the original neural network testing accuracy is 61.76%, it can be found that the hypothesis is true. It supports that the position of the relevant result will influence the user’s time to get the target result and their accuracy and satisfaction. There is a relationship between users’ experience and the target position. However, the testing accuracy is not that high. There might be several reasons. First, as mentioned above, the size of the data set is too small, which is only 288 rows of data. So, it causes that the train data and the test data is not enough. Small size of the train data will cause the overfitting issue and small size of the test data will cause the accuracy influenced by the outlier easily. Second, the users’ experience not only depends on the target position, but also varies from person to person. For example, the time to get the target result might be different for different person even if the result is at the same location. But in general, the testing accuracy can support the hypothesis.

However, surprisingly, the explanation accuracy is only 57.35% which is slightly lower than testing accuracy 61,76%. It means that, in this case, the explanation mechanisms will not improve the performance. Instead, it may reduce the accuracy. Compared with the procedure in Gedeon’s paper [3] to discover the reason why it does not improve the performance, it can be found that the raw data will be normalized to an interval from 0 to 1 while in eyegaze data, Accuracy and Satisfaction remain the same format as the ones in raw data. Therefore, to discover the influence of the normalization, the experiment using the normalized data has been done where the normalization strategy used is minmax normalization. The learning rate should also be adjusted because of normalization. Here are the results of the accuracy with different learning rates:

Table 9. The testing accuracy with different learning rates and with normalized data

Learning Rate	Testing Accuracy
0.00001	47.13 %
0.0001	46.25%
0.001	52.86%
0.01	50.85%
0.1	47.22%

Surprisingly, the highest accuracy is only 52.86%. So, the accuracy gets a further decrease and from the accuracy and the confusion matrix, it can be found that, with the normalized data, it will classify most of the data into one class, which means that the normalization cannot improve the performance of the explanation mechanisms in this case. Therefore, there should be other reasons that influence the accuracy. First, as mentioned above, the size of the data set is too small, so the training data is not enough which will cause the overfitting issue. Second, the reason could be that it is hard to find the input value when gradient equals 0, so in the experiment, the input value whose gradient is closest to 0 is considered as the boundary approximately. Third, it could be the values of satisfaction that influence the accuracy. From Kim's report [2], it says that the format of the satisfaction in eyegaze data set is a 7 point likert scale. And when looking into the data, it can be found that most of the satisfaction data is gathered in an interval from 4 to 7, which means the interval of the satisfaction is too small. Therefore, the rule related to the satisfaction data could be not accurate. And another reason could be the attributes for the rules in explanation mechanisms are not enough. In the neural network, there are 4 inputs: Type, Total Time, Accuracy and Satisfaction. However, in explanation mechanisms, the number of the inputs for generating the rules is only 2: Total Time and Satisfaction, because Type is categoric data which is turned into binary numbers 0 and 1 and Accuracy is a Boolean data which records whether it is accurate or not. Therefore, the generated rules may not show the real relationship, so that the accuracy will decrease.

And as for LSTMs network, compared with the original neural network in Table 8, it can be found that the testing accuracy becomes higher, from 61.76% to 66.18%. So that it can be found that with LSTMs, the performance of the classification will be improved, which got the same conclusion as Sundermeyer [9]. Sundermeyer found that the performance of the original approach could be considerably improved by LSTMs network. However, in Sundermeyer's experiment, the improvement will be 15%, while, in this case, the improvement of the accuracy is only 7%. The reason could be LSTMs is more suitable for language models and videos or images with an unknown length. So for this classification task, the LSTMs network is limited.

3.2 Convergence Speed:

As mentioned in 2.3, the other dimension of the performance is convergence speed, which is represented by the number of the epochs to let the training accuracy achieve 75%. Here is the number of the epochs of different technique:

Table 10. the number of the epochs to let the training accuracy achieve 75%

Technique	The number of the epochs
Original neural network	7300
Explanation mechanisms	7300
LSTMs	550

It can be found that the convergence speeds of the original neural network and explanation mechanisms are the same. The reason is that the original neural network and the explanation mechanisms share the same training part. Therefore, the speeds are the same.

However, after applying LSTMs, the convergence speed is much higher than original neural network. The number of the epochs to let the training accuracy achieve 75% is only 550, which is much smaller than 7300. And to visualize the convergence speed, here is the line plot that shows the difference of training accuracy between LSTMs and original neural network:

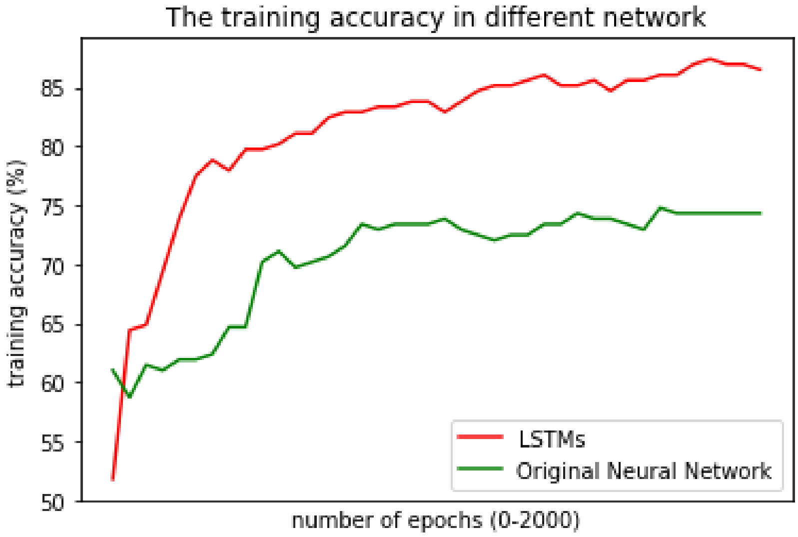

Fig. 3. training accuracy in different network

Therefore, it can be found that the convergence speed of LSTMs is much higher than the original neural network. This is also the reason why the number of epochs will be set to 200 instead of 500. For the learning speed of LSTMs is faster than original neural network. To avoid overfitting, 200 is enough for LSTMs, while the original 500 epochs will be too large for LSTMs which will cause overfitting.

4 Conclusion and Future Work

In summary, it can be found that when the relevant result is located at the front of the page, it will cost users less time to get the target result and the accuracy and satisfaction of the user are also higher, for using neural network, the target position can be classified correctly when inputting the users' experience. To improve the accuracy of this neural network, more data is required. However, in this case, when explanation mechanisms were added to help the classification, the performance of the classification will not be improved or become slightly worse. The reasons might be multiple among data structure, data type and data size. And when LSTMs is applied to the network, it can be found that the performance will be improved in both accuracy and learning speed dimensions. And to avoid the overfitting, the number of the epochs of LSTMs needs to be set smaller than the original neural network.

Considering the performance of the classification still has some limitations, there are several further analysis and experimentation can be done. For instance, more data should be added to test whether the performance of the neural network and explanation mechanisms can be improved. And since the explanation mechanisms currently cannot improve the performance, the specific reasons that cause the limitation could be found from the possible reasons mentioned above. And in the current study, the target positions are turning into 2 classes: target-front and target-back, which is too simple. Because different positions of target-front or target-back (1, 3, 5, 6, 8, 10) still have different weights. Therefore, binary classes may not enough to show the relationship. So, in the future study, the relationship between specific target positions can be explored.

5 Reference

1. Ghose, A., Ipeirotis, P. G., Li, B.: Examining the impact of ranking on consumer behavior and search engine revenue. Management Science 60(7), 1632-1654 (2014)
2. Kim, J., Thomas, P., Sankaranarayana, R., Gedeon, T., & Yoon, H.: Pagination versus scrolling in mobile web search. ACM, 751-760 (2016)
3. Gedeon, T. D., & Turner, S.: Explaining student grades predicted by a neural network. 1 609-612 vol.1. (1993)
4. Turner, H and Gedeon, TD.: Extracting Meaning from Neural Networks. Proceedings 13th Int. Con. on AI, Avignon (1993)

5. Yoda, M, Baba, K and Enbutu, I: Explicit representation of knowledge aquired from plant historical data using neural networks," International Joint Conference on Neural Networks, San Diego, vol. 3, 155-160 (1981)
6. Gallant, SI: "Connectionist expert systems," Communications of the ACM, vol. 31, no. 2, 152-169 (1988)
7. Bochereau, L and Bourgine, P: Expert systems made with neural networks, International Joint Conference on Neural Networks, vol. 2, 579-582, (1990)
8. Gedeon, TD and Bowden, TG: Heuristic pattem reduction, Proc. International Joint Conference on Neural Networks, Beijing, 449-453, (1992)
9. Sundermeyer, M., Ney, H., & Schlüter, R.: From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 23(3), 517-529. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), vol. 23, no. 3, 517-529 (2015)