Determining Search Task from Eye Gaze Data Using Casper

Adam Green

College of Engineering and Computer Science, Australian National University
u6047892@anu.edu.au

Abstract. Search engines such as Bing and Google have complex systems in place to ensure your searches return relevant results. In this, a model was attempted which can determine the task based only from the user eyes movements. To achieve this a model called Casper was used, Casper is a modified cascade network with an edited version of RPROP used. Casper was also extended to the domain of deep learning, however all attempts at this yielded poor results. Overall results were poor achieving an accuracy high of 58%. Further work needs to be done to investigate both the viability of the data and the ability of Casper to model it.

Keywords: Casper · Residual Networks · Search Engine Optimization.

1 Introduction

Search Engine Optimization commonly referred to as just SEO is the system by which search providers such as Google and Bing return relevant results to users based on their search query. These major search engines companies are constantly trying to ensure that a user's web search returns information which is relevant to the user's goals. Techniques such as user based custom search results [6] and search suggestions [2] have helped with this, although there still exists some mismatch between user web search goals and the results.

In addition to what the web search returns, there's also an interest into how the provider displays these results. Researchers have looked at the impact of snippet size (the length of the summary of the webpage) on a user's ability to correctly select the ideal result [5] . In this case test-subjects were given a set of tasks and not only was their response to the task recorded but their eye movements as well. There exist another dimension of SEO which involves studying the looking habits of user and placing results in regions more likely to garner their attention.

In this paper it will be attempted to determine what type of task a test subject was doing based purely off their gazing habits, this is will be implemented through the use of a Casper network [8] a variation of a cascade network .

Cascade Networks first introduced in 1990 [3] are relatively unknown today, with more popular network topologies dominating the field. Nevertheless Cascade networks in some cases provide an edge over other forms [6], or at least serve as a good introduction to the scope of different possible learning topologies. Seeing as Cascade networks are capable of successfully doing difficult classification tasks [1] it's valid to apply them to this task.

In simplest terms cascade networks start with only an input and output layer, then once these layers have reached a certain state a new hidden unit is added between them, this new hidden unit then receives the input as well as the output of any prior hidden neurons. Each iteration adds a new hidden unit to the network. The modifications which will be made to make Casper will be described in the following section.

Another aspect explored is how Casper, and cascade networks in general, can be used for deep learning. These networks dont easily translate to being used for deep learning. Particurly the way in which they have to be trained. The performance of these deep learning aproaches using Casper are presented.

First the dataset for the eye gaze data was analyzed and encoded into a favorable format. A Casper network was then trained to model a classification task of predicting based from the eye gaze data the topic. The results for multiple different Casper configurations will be analyzed and discussed. Finally a conclusion and recommendation on any extensions on this work will conveyed.

2 Method

2.1 Eye Gaze Dataset

The eye gaze dataset [5] contains multiple sheets, the sheet of interest is AOIs fixation. This data has records how long each participant looked at a certain area of the screen, for each task. The screen is broken up into ten regions each called areas of interest (AOI). The time within that area of interest the subject was looking at the URL, Title and Snippet was recorded in both second and log10 seconds. The two task types are navigation and info, navigation is for a task with a navigation goal and info for a task with an information goal.

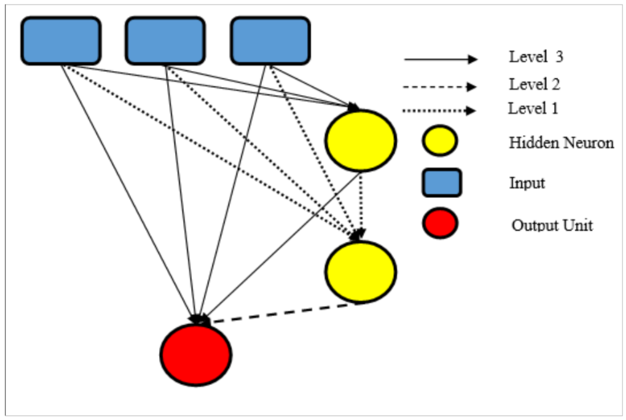
Examining the data it can be seen that many of the AOI receive no attention most of the time, in order to not have a sparse input and to stop inputs which are un useful are method to avoid this was created.

Figure 1 shows that most of the time spent is on the first three AOI, with almost none spent on AOI 6 to 10. To avoid using multiple sparse inputs AOI's 4 to 10 were summed for each task. This ensured that the model isn't receiving sparse inputs while still retaining information from the low use AOI's. This part of the process was done using python's panda's package.

Another task for the dataset was to change its shape to be usable for modelling. Initially the dataset had each AOI occupying its own row using pandas and numpy each row was horizontally stacked to form a 1-dimensional array. With each 1-dimensional array corresponding to a task done by a participant.

The 'info' and 'nav' labels were switch to a binary encoding, with 1 being for info and 0 for nav. Both the examples and labels were transformed into PyTorch tensors.

Fig. 1. Topology of a Casper model with two hidden units, the level of each weight has also been labeled.



2.2 Casper Model

The Casper model is a modified version of a cascade network, it has shown to be better at generalization and avoids overfitting [3], it was implemented in PyTorch.

A cascade network is created in a series of distinct iterations, with each iteration seeing the topography change due to the addition of a new hidden neuron. In its first iteration only the inputs and the output layers exist, after this is trained a hidden unit is added between the input and output layer. Next a hidden neuron is added to the model and its weights trained, a new hidden neuron not only receives the initial input but also receives a weight from all previous hidden neurons. The new hidden neurons output will be to the output units but also to every other hidden unit after it. This is shown in Figure 2, where the connections of each hidden layer (yellow circle) is shown.

Casper's topology is the same as a cascade network the change occurs in training, in a conventional cascade network after a hidden neuron is trained its weights are frozen. This is thought to effect the ability for a network to generalize. Casper's no longer freezes the weights, it instead uses a modified version of RPROP [7], a learning algorithm which bases weight changes only on the sign of the calculated gradient. RPROP's modified to have three distinct learning rates for weights based upon there topological position. Level 1 learning rate is for a the weight directly inputted into the most recently added neuron, level 2 learning rate is for the connection between the most recently added neuron and the output units, Level 3 are the remainder. It must be noted that these learning rates are only initially present and are changed as a result of RPROP, learning rates a re-initialized every time a new neuron is added. The different learning rates are supposed to minimize the amount of 'learning' the already existing neuron can do whilst still allowing them to train to some extent. Intuitively this should lead to them being more flexible and to prevent them being stuck with erroneous weights. The different levels are shown in Figure 2, notice all weights except for ones directly interacting with the new neuron having level 3 learning rates.

Another step proposed in Casper is to add simulated annealing to the weight decay, this paper has not implemented and the impacts on results is described in the discussion.

Another important factor is when to stop training a hidden neuron and add a new one, this was done the same as in Casper. If the loss has not decreased by more than one percent between a certain previous step and the current step training on the neuron was stop and a new one added.

2.3 Deep Residual Network

Cascade networks, unlike other networks, don't easily extend to become deep neural networks. For the purpose of this paper the generally known definition of Deep Learning will be used, i.e, a deep neural network is a network being more then 3 layers deep.

An approach used to use cascade networks succussfully for deep learning was explored. The first approach was to naively connect 3-7 Casper networks in series.

The second approach was to create a residual network [4] with Cascade networks being the consituent cells. A residual network after every cell concatenates both the previous cells output and the previous cells input. This has been shown to allow for deep neural networks acheive greater performance on some tasks [4]. Previously deep neural networks could only be five or six layers deep before any increase in depth would no longer increase performance. Residual networks allowed for deeper networks which increase perfromance to be made. A key note is that residual networks are primarily seen in the computer vision space, and see relatively little use in other areas.

The reasoning behind implementing a Cascade residual network in this case was to investigate wether or not Casper networks, like convultional networks, can see a performance increase by using residual connections.

Figure 3 shows a Casper cell with a residual connection. Note how the effective output is a concatenation of both the model input and output. Multiple network depths were experimented with. Figure 2 shows a 6 cell deep residual network using Casper as its cells.

3 Results

Models were trained on the eye gaze dataset for a classification problem to determining type of the task. Both different learning rate level values and number of added hidden neurons were tested and results tabulated. To produce a stable accuracy, each model was trained 25 times on datasets randomly distributed between training and test, with a roughly 80-20 split. The loss function chosen was binary cross entropy, and a sigmoidal activation functions was used for each layer.

For the deep Casper Residual networks multiple different depths were used, with different activation functions between cells used. The baseline naively connected deep Casper neural network also had many depths experimented. However both ultimately failed to have any degree of acceptable performance. The performance was never above random. They both failed to converge, with the loss never stabilising.

With the base Casper network acheiving a accuracy of 58% it, out of the three models tried, had the best performance.

4 Discussion

The overall result of all Casper implementations was poor, although there has been no other attempts with this dataset so it cannot be concluded that the poor results lie only on the Casper models. Overfitting was likely a cause of the poor ability of the model to generalize, simulated annealing may have been a solution to this.

Difficulties implementing the Casper method have the slim chance of introducing implementation errors. The lack of available datasets to benchmark and confirm Casper performance was also a contributing factor in this. Consequently it can be said that due to inabilities to confirm Casper performance there is a chance that there exists implementation errors.

Perhaps different loss functions would provide a better performance, although binary cross-entropy should be appropriate for this use case.

The mechanism whereby if the model hadn't decrease loss by more than one percent was flawed, in every case it was used overfitting was already occurring before it could add a new neuron. The training time of the models was quite fast with training speed remaining relatively linear based on hidden unit amount.

The relatively small data size may have impacted the ability of the model to generalize and caused it to overfit. The large amount of inputs compared to samples likely effected the models successfulness.

Intuitively its likely that when training the deep Casper models they struggled to converge due to new hidden units being added to cells at too high of a rate. It would be good to experiment on other methods to determine when to add a new neuron and where in the deep network to add it. Its likely our approach, of finding the cell with the smallest parameter change, was too simple and ineffective to select the optimum cell for correct training. Assuming that the other cells in the system were still rapidly changning their parameters adding a new neuron to the most stable cell likely increased the overall instablity in the network, leading to difficult in converging.

5 Conclusion

Comparing the Casper models results with other neural based techniques would allow for Caspers performance to be seen in context, providing clarity on whether the poor performance lies with the model or the dataset. Simulated annealing should also be implemented or some other weight decay mechanism, likely leading to increased generalization ability. More data would increase the models generalization performance as well as a broader range of test subjects.

The application of using deep learning in for the deep Casper networks was ulimtately unsuccussful. More investigation into this matter would likely be fruitful but it is recommended that a larger dataset be found first.

If a model is eventually capable of determine task type by eye gaze it's not too unlikely that Google or Bing may look into using user eye-gaze when they are searching, possibly improving the ability of search engines. Nevertheless this area is relatively unknown and of high commercial interest justifying any further research.

In conclusion experimental results of using eye gaze data to determine the task type was relatively unsuccessful, for any commercial application there would need to be a large increase in accuracy. However it cannot be said in certainty whether the Casper method or the dataset is the cause of the poor performance.

References

1. Bianucci, A.M., Micheli, A., Sperduti, A., Starita, A.: Application of cascade correlation networks for structures to chemistry. Applied Intelligence **12**(1-2), 117–147 (2000)
2. Bowman, D.E., Ortega, R.E., Hamrick, M.L., Spiegel, J.R., Kohn, T.R.: Refining search queries by the suggestion of correlated terms from prior searches (Dec 21 1999), uS Patent 6,006,225
3. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. In: Advances in neural information processing systems. pp. 524–532 (1990)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
5. Kim, J., Thomas, P., Sankaranarayana, R., Gedeon, T., Yoon, H.J.: What snippet size is needed in mobile web search? In: Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval. pp. 97–106 (2017)
6. Mysen, C.C., Verma, N., Chen, J.: Ranking custom search results (Jul 26 2011), uS Patent 7,987,185
7. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: IEEE international conference on neural networks. pp. 586–591. IEEE (1993)
8. Treadgold, N.K., Gedeon, T.D.: A cascade network algorithm employing progressive rprop. In: International Work-Conference on Artificial Neural Networks. pp. 733–742. Springer (1997)

Fig. 2. 6 layer Deep residual network with Casper network cells.

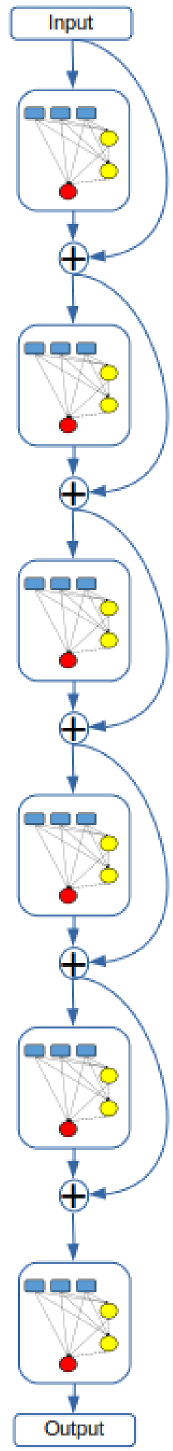


Fig. 3. Casper Residual Cell