Application of evolutionary algorithm in predicting votes of people based on eye gaze data with Progressive Image Compression technique

Dominic Daoyuan Yang (u6547564)

29/05/2021

Dataset and technique

- Dataset: dataset-image-manipulation
- Technique: task-NN07-ProgressiveImageComp-image-manipulation

Abstract

This second version extends version 1 by applying evolutionary algorithms. Evolutionary algorithm is applied with mutations. Results shows that evolutionary algorithm does help finding a better neural network.

Version 1 paper focuses on extending the additional goal of the paper that provided the dataset, which is, to find out if it is possible to determine what the participant will vote based on data collected by eyegaze of that participant. We will implement the technique technique paper - progressive image compression, and Distinctiveness Analysis, attempting to minimize the size of the neural network and improve the performance. We will do a brief review on the two papers given and the dataset provided and discuss the implementation of the technique and the application to solving this problem. The result shows no significant improvement directly on the performance. Progressive image compression, however, is useful in simplifying the neural network.

Introduction

The problem to tackle in this paper is using the data collected by eye gaze to predict if the image presented to the person is manipulated. We will compare the performance, learning speed and the outcome of neural networks with and without progressive image compression.

The dataset is an experiment done on predicting whether their neural network can predict if the image is manipulated and if the participant thinks the image is manipulated based on the data collected by using eyegaze to track the participant's eye movement Caldwell et al. (2015). And the technique chosen is progressive image compression. It is using a neural network to perform image compression and decompression by reducing the size of hidden layers to find the trade off between the degree of compression and the quality of the compressed image Gedeon and Harris (1992). In the next section we will go deeper into the problem and the technique.

The motivation for me is that predicting a person's thoughts just by tracking the eye movement has a large potential to be extended and applied. More research is needed in this field and is a good problem to apply progressive image compression technique on.

Method

Techniques

Progressive image compression technique can be used to reduce the size of a neural network while maintaining the functionality. It manipulates the sizes of the hidden layers, starting with a high (but still considerably low) number and gradually reduces while monitoring the performance of the neural network. With the result we can decide how much performance we want to trade for size.

The technique paper uses Distinctiveness Analysis to improve the speed and outcome of the neural network. The paper talked about dividing the pictures into non-overlapping pieces and using the pieces to train the neural network for image compression, but in this paper we will only focus on the other method that is useful to solving this problem, which is regularly checking the angular separations of nodes.

Implementation and analysis

To implement progressive image compression technique on this dataset, we use a standard neural network and start with 10 hidden neurons and gradually reduce the number.

The implementation of Distinctiveness Analysis is done by repeatedly checking on the weights of nodes. If two nodes are too similar to one another then reinitialize the weights of one node with random values. Since in this implementation the checking happens every 50 epochs, the nodes are well trained, and thus nodes that are close to each other are not likely to have happened by instance.

Results and discussion

Progressive image compression

The results we got without adding Distinctiveness Analysis:

Table 1: Distinctiveness Analysis Results

hidden_neurons	accuracy_.	avg.accuracy_.	An_example_confusion_matrix
50	65.66	64.98333	[[165. 13. 0.]
NA	64.31	NA	[80. 26. 0.]
NA	64.98	NA	[10. 3. 0.]]
45	59.93	62.62333	[[164. 14. 0.]
NA	62.96	NA	[77. 29. 0.]
NA	64.98	NA	[10. 3. 0.]]
40	63.97	62.96000	[[164. 14. 0.]
NA	59.93	NA	[76. 30. 0.]
NA	64.98	NA	[11. 2. 0.]]
35	65.32	63.86000	[[176. 2. 0.]
NA	65.99	NA	[104. 2. 0.]
NA	60.27	NA	[13. 0. 0.]]
30	64.31	61.39000	[[178. 0. 0.]
NA	59.93	NA	[106. 0. 0.]
NA	59.93	NA	[13. 0. 0.]]
25	63.97	61.27667	[[178. 0. 0.]
NA	59.93	NA	[106. 0. 0.]
NA	59.93	NA	[13. 0. 0.]]

Since accuracy is not the only indicator for the quality of the neural network, when the number of hidden neurons is under 35 the prediction is significantly biased and therefore not useful. For the Distinctiveness

Analysis part we are using 45 hidden neurons. The data shows that the accuracy doesn't change significantly, which means that reducing the number of hidden neurons doesn't affect the performance of this neural network much. But we can see the change in confusion matrix. As the num of hidden neurons decreases, there is a slight increase in the accurate predictions in the middle column (participants voting the image is manipulated), and as the number of hidden neurons keeps dropping the correct prediction in the middle column drops again. In this case the best trade off point of hidden layer size and quality of prediction is at around 45 hidden neurons.

Distinctiveness analysis

The accuracy comparison result:

Table 2: Accuracy Comparision

with.DA	X	without.DA	X.1
loss	accuracy	loss	accuracy
0.7829	59.93	0.7613	63.97
0.7902	59.93	0.7602	63.64
0.7991	59.93	0.7617	65.32
0.7792	61.62	0.7725	64.65
0.7736	59.93	0.767	64.65

We can see that with Distinctiveness Analysis (DA) added the performance and outcome of the neural network is not better in both loss and prediction accuracy.

However, from plotting the loss over time during training in figure1 figure2 and figure3 we can see that in the early stage of the learning process the slope of the learning curve is steeper with Distinctiveness Analysis added. Figure1 is a representative of all the identical curves without DA, figure2 and figure3 are two example curves with DA. This shows that DA does speed up learning, and in the early stage of the learning process DA goes far ahead the other.

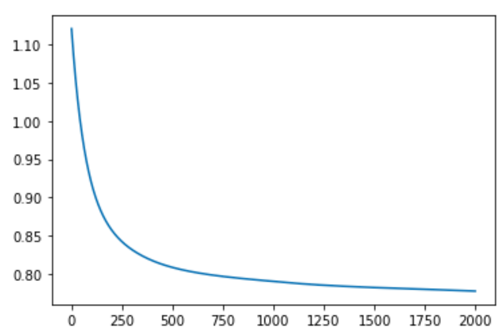

Figure 1: With DA

Extention with evolutionary algorithm (content of assignment 2)

Evolutionary algorithm is applied to the work metioned above. The concept is to simulate a natural evolution to the neural networks and apply survivor selection. The best performing neural nets survive and pass their genes (parameters) onto the next generation. Repeat this processor and after many generations the population will be filled with best performing genes under the target condition (selection).

In the implementation, a neural net with its parameters initialized randomly is fed into the evolution. In the first generation many randomly genes are added to the population to enrich the gene pool for variety.

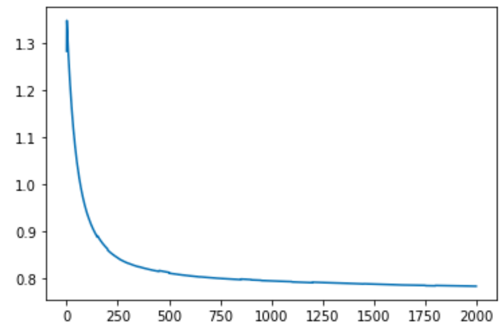

Figure 2: An example with DA

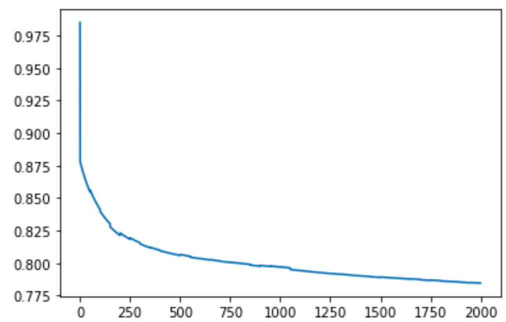

Figure 3: Another example with DA

Each generation after the first takes the top 3 best performing neural net and make children with their genes (refer to function `evolve` in code). Here due to limitation of time prediction accuracy is the only criteria determining performance. In `evolve` which generates the next generation of neural nets, `mutate` chooses a random gene in a random individual and randomly assign a value to that gene. This makes sure that the gene pool is always being added fresh blood for better performance.

Evolution results

Results running 10 generations:

```
### gen 0, children 1
[]
### gen 1, children 11
[59.93265993265993, 59.93265993265993, 59.93265993265993]
### gen 2, children 30
[59.93265993265993, 59.93265993265993, 59.93265993265993]
### gen 3, children 30
[59.93265993265993, 59.93265993265993, 59.93265993265993]
### gen 4, children 30
[59.93265993265993, 59.93265993265993, 59.93265993265993]
### gen 5, children 30
[61.61616161616162, 59.93265993265993, 59.93265993265993]
### gen 6, children 15
[59.93265993265993, 59.93265993265993, 59.93265993265993]
### gen 7, children 21
[59.93265993265993, 59.93265993265993, 59.93265993265993]
### gen 8, children 21
```

```
[62.96296296296296, 59.93265993265993, 59.93265993265993]
### gen 9, children 21
[59.93265993265993, 59.93265993265993, 59.93265993265993]
```

some strange behaviors can be observed. The best performing neural nets should be passed to the next iteration and in sometimes the performance of the next generation is not better than the previous one. It is confirmed that the best performing neural nets are passed to the next iterations. The only possible explanation to this is that the same neural net returns different performance results under the same condition. The reason to this has not been found. More debugging and reasoning needs to be done to solve this issue. But apart from that, evolutionary algorithm does help finding neural net parameters that suits the best under this conditions. Running on a different dataset and getting that issue metioned above solved would improve the results dramatically.

Conclusion and future work

Although we didn't have the original neural network used in dataset paper, we still managed to present the contrast and compare the performance with and without the technique implemented. qIn optimizing the size of the neural network Progressive Image Compression technique does give researchers a clear view on the tradeoff between the size of the hidden layer and the performance of the neural network. It is a technique that can be applied to many questions and help with neural network optimization. Distinctiveness analysis gives an impressive improvement in speeding up the learning process. By diverging the nodes that are approaching one another, the neural network learns more efficiently and has fewer redundant nodes.

Future work still needs to be done to improve the accuracy of the prediction of the neural networks in this paper. Neither of the techniques here improves the final outcome of the neural network, however, they do improve the performance of the neural network in different ways. Also a solution is needed for the dismatch in evolution that the same neural net performs differently under the same condition.

References

Caldwell, Sabrina, Tamás Gedeon, Richard Jones, and Leana Copeland. 2015. "Imperfect Understandings: A Grounded Theory and Eye Gaze Investigation of Human Perceptions of Manipulated and Unmanipulated Digital Images." In *Proceedings of the World Congress on Electrical Engineering and Computer Systems and Science*. Vol. 308.

Gedeon, TD, and D Harris. 1992. "Progressive Image Compression." In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, 4:403–7. IEEE.