Predicting the Result with Bidirectional Neural Network and LSTM-Bidirectional Recurrent Neural Network Method on Two Manipulation Datasets

Yuxuan Chen
Research School of Computer Science , Australian National University
u6902682@anu.edu.au

Abstract: Current input data and a better dataset always led to a better result in deep learning. This goal of the paper was to develop a bidirectional neural network and a bidirectional recurrent neural network to predict the eye gazes manipulation. The two neural network both performed a better result on the maximum accuracy than the previous work [8][2],reach to 86.75%. The thesis choose Eye-Gazes Manipulation and Caldwell Manipulation Images Timeseries created by Caldwell, S., Gedeon, T., Jones, R. and Copeland, L., 2015[2] as datasets. After cleaning and preprocessing the data, predict manipulation. Under the BDNN and the LSTM-BRNN training ,with two different datasets, showed good results. It is also worth noting that the size of the dataset is very important to the study. For the future studies ,in addition to optimizing the network structure and training methods, it is also necessary to expand the data set when necessary.

Keywords: Deep Learning; Bidirectional Neural Network (BDNN); Bidirectional Recurrent Neural Network (BRNN); Eye Gaze

1 Introduction

1.1 Motivation

Nowadays, neural network-based technology is rapidly evolving. Neural networks and deep learning methods are being used to study topics in an increasing number of disciplines. This is especially true in the field of visual analysis research. Emerging research in forensics and computer science is identifying best practices for determining whether images have been manipulated, but there is a lack of research on how we humans perceive or do not perceive these fraudulent images[3]. Images on the internet are updated on a daily basis, and with the advancement of image editing software, the images we see today are becoming more deceptive.

We hope to combine the eye gaze data from the image processing prediction experiment with the technology of the bidirectional neural network and bidirectional Recurrent Neural Network, in order to gain more insight into how to use this technology in this problem area and seek more suitable solutions.

1.2 Related Work

While in majority of the NN prediction on eye-gaze data set , most of them are using FNN [3]or RNN both are one-way neural networks. And the bidirectional neural network has better results in prediction. BDNNs can be used as a simulation tool for evaluating some major cognitive psychology theories such as prototypes and dual-code theory [1] like the topic this thesis focus on.

What's more, for the previous work from Razi which reach the accuracy upon 73%[8],Caldwell et al. got 65.7%,and Tan achieves 67.82%[2]. This article will optimize the results in different solutions.

1.3 Research objectives

1. Analyze and process data to make the data more in line with the style of deep learning.
2. Build a Bidirectional Neural Network and adjust the parameters and training process to make the results

better

3. Build a Bidirectional Recurrent Neural Network and adjust the parameters and training process to make the results better
4. Compare the results obtained by the neural network, and compare with the previous work and the analysis results of the data set, and get the improvement direction

2 Method

2.1 Data selection and preprocessing

2.1.1 Eye Gaze on Image Manipulation

As pervious discussed, we choose Eye-Gazes Manipulation by Caldwell, S., Gedeon, T., Jones, R. and Copeland, L., 2015[2] as our dataset. The purpose for this data is to compare eye gaze (inputs) to the two types of images: manipulated and unmanipulated (outputs). The eye gaze data is divided into the two groups: those who are viewing manipulated images and those who are viewing unmanipulated images[2].

There are 372 datapoints in the data set. Table 1 shows the feature name and the description. We use this data set in following experiments after preprocessing it.

Table 1. Description of the Eye Gaze on Image Manipulation[2].

Feature Name	Description
participant	the id number of the participant
num_fixs	the total number of fixations by the participant when looking at the image
fixs_duration	the total amount of time (in seconds) that the participant spent looking at the image
num_man_fixs	the total number of fixations by the participant when looking within the target area
Man_fixs_dur	the total amount of time (in seconds) that the participant spent looking within the target area
image_manipulated	whether the image the participant views is the manipulated or unmanipulated version, 0 = unmanipulated, 1 = manipulated
image	Number of the image
vote	the verbal opinion of the participant as to whether the image is manipulated or unmanipulated, 0 = voted unmanipulated , 1 = voted manipulated, 2 = don't know

Through a simple analysis of the data set, it is found that some of the columns are interference items for the experiment. At the same time, the data set is statistically analyzed by statistical methods.

The specific operation is:

1. Separate the data set from the excel file and save it in csv format.
2. Delete the “participant” columns when import the data set to the training file.
3. Delete the line 2 in the "vote" column. The reason is that 2 means that it is not certain, which will affect the result.

Perform grouping statistics on the data set before and after processing, and calculate the corresponding Precision, Recall, Specificity and Accuracy. Table 2 Shows the data distribution before and after the data set processing, and the statistical values after processing.

Table 2. Description of the data set before and after basic processing.

Table 2.1. Description of the numbers of the data set before and after basic processing.

Data Set	Description			Sum number
Original data set	image_manipulated	Vote	number	
	0	0	135	

		1	42	189
		2	12	
	1	0	92	
		1	87	188
		2	4	
Processed data set	0	0	135	177
		1	42	
	1	0	92	179
		1	87	

Table 2.2. Description of the data set after basic processing.

Data Set	Description			
Processed data set	precision	Recall	Specificity	Accuracy
	67.44%	48.60%	60.81%	62.35%

After finishing the simple data cleaning, the following operations were performed on the data before training:
1. Divide the dataset
2. Standardize numerical features
3. One-hot encoding for disordered type

2.1.2 Caldwell Manipulation nip Images Timeseries dataset

Caldwell Manipulation Images TimeSerises dataset include 31,114 data of the experiment. It contains 5 different Image ID with 80 participants. Each line of the data,has the X and Y pos ,and recorded the participant start time and end time.
The mean duration time of each image is showing in the table below

	Image_ID				
	10	11	12	13	14
Count	5923	6194	6885	6684	5428
Mean duration	0.274375	0.199833	0.200950	0.240699	0.246138

2.2 Technique and tool selection

In the whole experiment, in addition to the basic torch framework's packet function, pandas , numpy , matplotlib is also used.

2.2.1 Bidirectional Neural Network

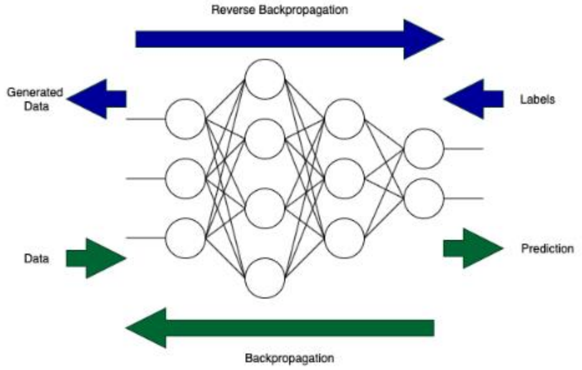

Fig1: Schematic diagram of BDNN structure

Bidirectional neural network experts optimize the neural network by using forward and back propagation techniques and adjusting the weight matrix of the bidirectional network[1].

2.2.2 Bidirectional Recurrent Neural Network

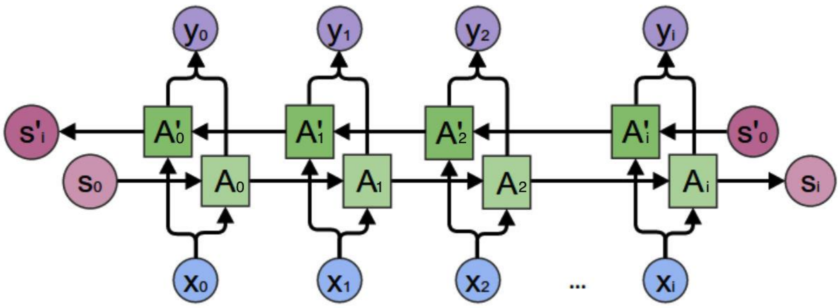

Fig2: Schematic diagram of BRNN structure

BRNN is a type of recurrent neural network. The basic RNN can only predict the output at the next moment based on the previous timing information, but some problems need to be linked to the previous and future states for common prediction. Then it comes BRNN.BRNN can achieve the above functions.

2.2.3 Long short-term memory (LSTM)

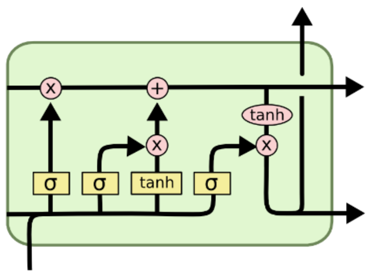

Fig3: Schematic diagram of LSTM structure

Long short-term memory (Long short-term memory, LSTM) is a special kind of RNN, mainly to solve the problem of gradient disappearance and gradient explosion during long sequence training. Simply put, compared to ordinary RNNs, LSTM can perform better in longer sequences[9].

2.2.4 Result Evaluation

We use accuracy as the result evaluation.
Accuracy is the most commonly used and most intuitive performance index, which is defined as follows:

$$accuracy = \frac{(TP + TN)}{(TP + FN + FP + TN)}$$

TP: True Positive. That means it is true,and set as positive.
FN: False Negative. That means it is false,and set as Negative.
FP: False Positive. That means it is false,and set as positive.
TN: True Negative. That means it is true,and set as Negative.

2.2.5 Experiment Environment

As for the technical equipment used in the experiment, the entire implementation was on JupyterLab with Python 3.7 and PyTorch version1.5.0 (CUDA 10.0) was used for neuron network traning The experiment was run on a desktop machine with AMD®Ryzen 5 3600 CPU, Nvidia GTX2060s GPU (6GB)

3 Result and Discussion

3.1 Experiment result analysis

3.1.1 Result of the Eye Gaze Manipulation with the BDNN and BRNN

Run each model 50 times, and calculate the average accuracy of different hyperparameter.

Table 3. Result of the accuracy.

Method	Description		
	Minimum Accuracy	Maximum Accuracy	Average Accuracy
Original data statistics	62.35%	62.35%	62.35%
BRNN	32.44%	85.14%	49.79%
BDNN	50.17%	86.75%	68.76%

Table 4.1: Table of the Hyperparameter and the accuracy in BDNN

Hyperparameter				Accuracy		
Size of the training and testing	Learning Rate	epoch	Batch_size	Minimum Accuracy	Maximum Accuracy	Average Accuracy
7:3	0.0001	100	64	50.17%	86.75%	68.76%
	0.001		64	24.08%	70.28%	56.18%
	0.0001		32	24.10%	73.49%	57.29%
	0.001		32	26.91%	77.11%	53.82%
6:4	0.0001		64	27.71%	71.49%	52.87%
	0.001		64	25.70%	85.54%	64.45%
	0.0001		32	28.51%	65.86%	45.78%
	0.001		32	24.90%	61.85%	46.18%

Table 4.2 :Table of the Hyperparameter in BRNN

Hyperparameter				Accuracy		
Size of the training and testing	Learning Rate	epoch	Batch_size	Minimum Accuracy	Maximum Accuracy	Average Accuracy
7:3	0.0001	100	64	21.76%	85.54%	42.33%
	0.001		64	22.89%	70.68%	48.82%
	0.0001		32	12.45%	81.53%	47.44%
	0.001		32	26.91%	71.49%	45.67%
6:4	0.0001		64	23.88%	71.49%	46.18%
	0.001		64	25.70%	81.93%	43.45%
	0.0001		32	32.44%	85.14%	49.79%
	0.001		32	38.96%	75.09%	46.99%

Visible from the Table 3,both of the BDNN and BRNN all have maximum accuracy that exceed the original data .The BDNN model even reach 86.75% as the maximum. It can be seen from the Training Loss and Training Accuracy images of the two models all have a convergence trend and a gentle trend after rising.

However the result of both models are very unstable. In the experiment, there have been many cases of normal accuracy without abnormal data. The reason will be discussed in 3.2 Limitation.

3.1.2 Result of the Caldwell Manipulation images TimeSeries with the LSTM-BRNN

Run each model 10 times, and calculate the average accuracy of different hyperparameter.

Table 5:Table of the Hyperparameter in LSTM-BRNN

Hyperparameter				Accuracy		
Size of the training and testing	Learning Rate	epoch	Batch_size	Minimum Accuracy	Maximum Accuracy	Average Accuracy
7:3	0.0001	100	64	38.08%	78.35%	53.88%
	0.001		64	42.43%	76.27%	51.24%
	0.0001		32	43.44%	85.03%	62.18%
	0.001		32	39.85%	79.88%	61.24%
6:4	0.0001		64	37.61%	77.79%	56.67%
	0.001		64	35.36%	82.53%	59.66%
	0.0001		32	42.31%	76.85%	55.99%
	0.001		32	44.69%	68.53%	60.08%

3.2 Discussion

The output of the model in the first experiment is very unstable. From the results of the experiments, it can be known that whether it is BDNN or BRNN, the loss and accuracy obtained during training are very unstable. No matter how you adjust the parameters, it is very unstable. However, with the second dataset ,the LSM-BRNN gives out very stable result and very easy to see the result.

The structure of the network is not the most suitable structure. For such a data set, if you want to make a prediction problem, because the data set needs to be predicted only 0 and 1, the results obtained will not be accurate enough.

Data set is too small for the first part of the experiment. The most important point is that the data set is too small. After data cleaning, there are only 300 available data, which is not enough to train a very stable neural network. So every time the output of the two networks is very uncertain

4 Conclusion and Future Work

From the pervious discussion, we found the BDNN and BRNN all have a good effect on the prediction of the data set. As the data was trained by neural network ,so that ,finding a suitable network is very important. For the BRNN ,the maximum accuracy reached 86% ,and the BDNN reached 85%,all have improvements than the pervious works.

Besides, with different dataset, BRNN have a significant improvement, this is because of the dataset changed, from a 300 lines small dataset into a 30,000 lines larger dataset.

For the future work,, in addition to optimizing the network structure and training methods ,with the Deep Learning method BRNN has be used in this essay ,in my own points, reinforcement learning can be used to strengthen the stability of the model, it is also necessary to expand the data set when necessary.

5 Reference

1. A. F. Nejad and T. D. Gedeon, "Bidirectional neural networks and class prototypes," Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, pp. 1322-1327 vol.3, doi: 10.1109/ICNN.1995.487348.

2. Caldwell, S., Gedeon, T., Jones, R. and Copeland, L., 2015, July. 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).

3. Tan, Z., Plested, J.: Classification of Humans' Perception of Manipulated and Unmanipulated Digital Images Using Feedforward Neural Network with Network Reduction Technique. ICONIP2019 Proceedings. (2019)

4. Caldwell, S.: Framing digital image credibility: image manipulation problems, perceptions and solutions, https://openresearch-repository.anu.edu.au/bitstream/1885/133844/1/Caldwell%20Thesis%202017.pdf, (2016)

5. Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE transactions on Signal Processing, 1997, 45(11): 2673-2681.

6. GitHub. 2020. *ljxw88/BDNN-for-SFEW*. [online] Available at: <https://github.com/ljxw88/BDNN-for-SFEW/tree/master/code_v1> [Accessed 26 April 2021].

7. Blog.csdn.net. 2021. *Pytorch study notes (7)-multi-layer neural network predicting housing prices (regression model) + GPU usage*.(in chinese) [online] Available at: <https://blog.csdn.net/weixin_41710905/article/details/103914549> [Accessed 27 April 2021].

8. Razi, S., 2020. *Optimizing Hyperparameters for Bidirectional Neural Networks using Evolutionary Algorithms*. [online] Users.cecs.anu.edu.au. Available at: <http://users.cecs.anu.edu.au/~Tom.Gedeon/conf/ABCs2020/paper/ABCs2020_paper_v2_183.pdf> [Accessed 26 April 2021].

9. 2021. *LSTM analysis*(in chinese) [online] Available at: <https://zhuanlan.zhihu.com/p/181634030> [Accessed 31 May 2021].

10. Vapnik, Vladimir. *The nature of statistical learning theory*. Springer science & business media, 2013.