Improving Machine Learning's Ability to Determine the Authenticity of the Anger Facial Expression

Yun Chao[1]

Research School of Computer Science,
Australian National University,
Canberra ACT 2601, Australia
u5264256@anu.edu.au

Abstract. This study extends from [3] and uses neural network input analysis techniques from [5] and deep learning algorithm, specifically long short-term memory (LSTM) recurrent neural networks, for the attempt on improving classification accuracy. [3] used several physiological signals, such as pupillary response and blood pressure, to analyse human's ability to distinguish between genuine and posed anger. In the first part of this research, I first built a two layered neural network and used the brute force and functional measure techniques from [5] to analyse the input variables. After adjusting the neural network by removing inputs or manually setting initial weights, I found that techniques from [5] could not improve the result achieved by the original study [3], which may be due to the small number of input variables available for adjustment. In the later part of this paper, I used a bigger set of pupil diameter data from [3] to perform classification with a LSTM algorithm and an ANN for comparison. The testing accuracy with the bigger data set with the ANN was able to reach 93%, which is a lot higher than the smaller data set with only six features. Meanwhile, the LSTM algorithm was able to achieve a higher accuracy than what was found by [3] at 96%.

Keywords: Input Analysis · Artificial Neural Network · Long Short-Team Memory

1 Introduction

Researchers have long been trying to find a method for machine learning to recognise and determine human's facial expressions. While numerous machine learning and neural network techniques have been developed, researched, and tested, no single best method can work perfectly in every situation. This study will extend on the research of [3] and investigate whether a different neural network analysis technique can improve the classification accuracy.

Scientists have been studying people's reactions towards other's facial expressions for decades. Variables such as the different facial expressions, the reaction time after observing the expressions, and the ability to determine the authenticity of the expressions have been topics of many research papers [2, 3, 7–9]. Similar to [3], [7] analysed people's ability to distinguish between genuine and posed smiles using verbal response and [8] studied the same topic using other physiological signals such as pupillary response and blood pressure. This study will be an extension of [3], which is about people's ability to distinguish between genuine and posed anger expression.

Artificial neural network, or ANN, has been called the black box method due to the lack of transparency in how features are selected and weighed in the process [11]. This causes the algorithm optimisation to be an extremely difficult task. It also makes it very hard for researchers to find the relative importance of the factors or variables of their studies [1]. Therefore, data scientists have been creating, researching, and adjusting many different techniques to overcome this difficulty and find a way to select features and analyse variables accurately and reliably. For example, statistical function, specifically curve of permanence, was introduced by [1], and was found to be a good tool for variable selection. [10] introduced the connection weight approach, which uses raw connection weights between any two layers of an ANN, and found it to accurately and consistently quantifying and ranking the variables' importance.

In addition, [5] introduced a new functional analysis of the weight matrix using the hidden neurons behavioural significance and a sensitivity measure that ranks the effects of perturbing rather than eliminating the inputs. However, they found that sensitivity of an input does not necessarily correlate with the importance of an input. In this paper, I focus on the functional analysis developed by [5].

On the other hand, recurrent neural networks (RNN), especially Long Short-Term Memory (LSTM), have been widely used to solve numerous types of machine learning problems. LSTM has been found to be a very effective and scalable model for sequential data machine learning. With a memory cell, LSTMs can maintain its state over time, and its gates control information that enters the cell [6].

In the first part of this study, I used the technique introduced in [5] to revisit the study done by [3] and find out if it is possible to improve the machine learning classification accuracy. I first used brute force adjustments to roughly analyse and rank the importance of each input of the ANN. Then, by finding the similarities between the activation vectors of the hidden layer in the ANN, I was able to compare the variables' importance using their functional measure [5]. Lastly, I adjusted my ANN using the combined rankings from the two methods then trained and tested with the given data set. However, since the data set from [3] has a small number of input variable, I did not expect the [5] techniques to work well in improving the classification results.

For the second part of this paper, I trained a LSTM on a larger data set and compared its classification accuracy with an ANN trained using the same data set. Since the larger data set consists of time series data, I expected the LSTM to perform better than a simple ANN. I also expected the ANN with the larger data set to achieve a higher accuracy than the first part of this paper.

2 Data and Methods

2.1 Data Description

This study extends from the research in [3] by attempting to improve classification accuracy using different machine learning techniques and algorithms. I used the same data set collected by [3] with the ANN technique introduced in [5] in the first part and a LSTM algorithm in the second part. In [3], they performed experiments by playing 20 videos in different sequences to 22 participants individually. While the participants are watching the videos, their eye gaze, pupil size, skin conductance, blood volume pulse, and heart rate would be measured by the Eye Tribe eye gaze tracker and the Empatica E4 bracelet on their wrist [4, 13].

The first data set consists a total of nine columns. One column is the label for the participants, six are the data variables collected from the participants, and two are the indicator of genuine or posed expression [3]. The research in [3] is an extension from [7] and they both used six of the same variables when classification training. The six input variables comprises of the following statistical features: mean, standard deviation, minimum, maximum, and the first and second differences of the process signals obtained from the eye tracker and the bracelet [3, 7]. Moreover, there are 20 participants' observations for each of the 20 videos, which means the total 400 total observations in the data set. Two participants' data were excluded due to complications with the equipments [3].

The second data set, which is much larger, only consists of pupil diameters data collected from the participants. The data set is composed of time series data. As the participants watch each video, their left and right pupil diameter are recorded separately in the data set by each time frame in the video [3]. The longest video has 186 time frames, with 20 participants and 20 videos, there are about 74,400 pupil diameters for both left and right eye.

2.2 ANN Methods on Smaller Data

The first data set provided is very clean and normalised to be used for an ANN. There are only a couple of things I adjusted in the data set before I started to build the ANN. First, I dropped two unnecessary columns, which are the identifiers for the observations and videos. Second is to make the identifier for genuine or posed expression a boolean column, so classification can be done more easily. With the data set pre-processed, I started setting up the structure of my ANN.

For the ANN structure, there are six input neurons, which are the variables in the data set, and two output neurons, which are the boolean values that indicates whether the expression is genuine (True) or posed (False). I built the ANN with only one hidden layer of 10 hidden neurons, set the learning rate of 0.01, and trained the ANN with 5000 epochs. I split the data set into two parts, where 80 per cent of the data are used to train the ANN and the remaining 20 per cent would test the ANN accuracy. This is to perform the holdout method of cross validation on the ANN. Moreover, the ANN is a two layered linear network with ReLU as the activation function and cross-entropy as the loss function.

For the first part of the [5] technique, I implemented the brute force analysis and rank the inputs. I removed each input one at a time to see the effects of the classification accuracy. In [5], they removed inputs by pairs because they believe it would have more impact on the outcome. However, since the data set in this study only has six variables, I thought dropping one variable would have enough effect. I also implemented two different models with different initial weights, Model A with random initial weights and Model B with uniform initial weights, to better compare the rankings.

In addition to the brute force analysis ranking, I also used the functional measure technique from [5]. For the functional measure, I used the cosine similarities between hidden activation vectors for each pair of inputs.

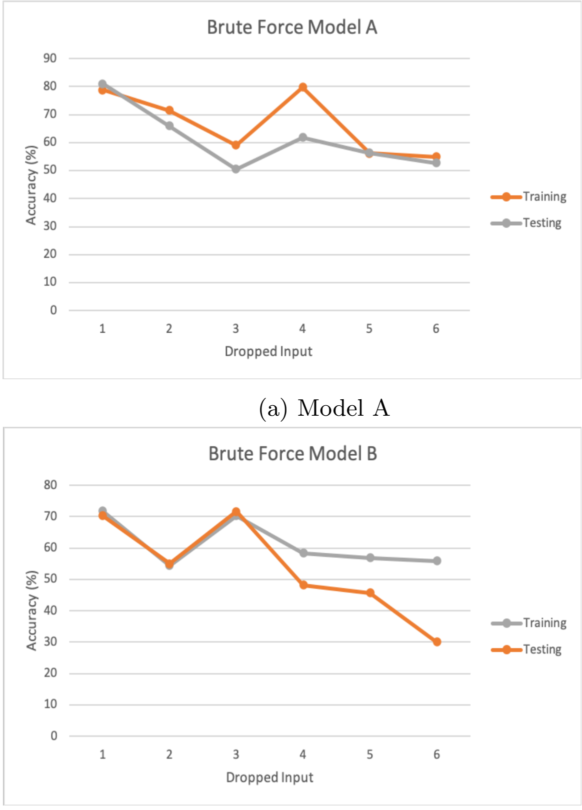
This helps me determine which of the inputs have similar affects on the output. [5] used the angles between the activation vectors to rank the inputs, but I found cosine similarity easier to implement. With the rankings from both brute force (Model A and B) and functional measure (Model C) analysis, I assigned a weight of 1 to the most significant and 0 to the least important inputs, with 0.2 incremented weights for the inputs in between. With the weighted significance, a combined input ranking (Model F) showed me the relative contributions of each variable. Lastly, I used this final input ranking to adjust my ANN by removing and setting initial weights for the least important inputs in attempt to improve the classification accuracy.

2.3 LSTM and ANN Methods on Larger data

There are two larger data sets, which are the left and right pupil diameters measurements for each participant watching each video. To use both data sets in a LSTM, I first combined the data sets and for each participant and each video, I combined the left and right pupil diameter (PD) into a tuple (left PD, right PD). Since a few participants' measurements were ignored and left blank in the data set due to measurement issues, I dropped those observations. I also set the empty time frames for the shorter video to zero, as LSTM cannot work with missing data. Then, for the algorithm to distinguish between videos with genuine and posed anger, I set the label for the videos as genuine and posed. Next, I built a LSTM, which samples from the model from time to time, whose the last step connects to a fully connected neural network to perform the classification. To compare the performance between LSTM and ANN algorithm, I used the same larger data set with the same ANN algorithm in the previous section without pruning.

3 Results and Discussion

3.1 ANN on Smaller Data Set



(a) Model A

(b) Model B

Fig. 1: Training and testing accuracy for brute force analysis models

First part of the technique is to use brute force by removing inputs one at a time. Fig.1 demonstrates the training and testing accuracy rates for the two different models created in brute force analysis. Model A and model B have different initial weights, randomised weights and uniform weights respectively. It is obvious from

Fig.1 that input 1 is the least important in both of the models, because of the relatively high training and testing accuracy rates observed when it is removed from the ANN. On the other hand, input 6 is the most important as the training and testing accuracy rates in both models are relatively low.

Using the accuracy rates in Fig.1, I ranked the inputs based on their significance, this is shown in Table 1. It indicates that the relative contribution for most inputs are the same between the two models, except for variable 3. Therefore, results from the second part of the technique needed to be considered before finalising the ranking.

Model	**Most Significant**					**Least Significant**
A	3	6	5	4	2	1
B	6	5	4	2	1	3

Table 1: Input Ranking after Brute Force Analysis

Using the functional measure analysis from [5], I calculated the cosine similarities of the activation vectors for the hidden layer. Table 2 shows the cosine similarities for each pair of the inputs. By the definition of cosine similarity, two vectors with the same orientation have the cosine similarity of one. If a pair of inputs has activation vectors with a cosine similarity of one, it means that the two inputs have the same function with regards to the ANN. As it can be seen in Table 2, most pairs of inputs have a cosine similarity close to one. While inputs 3 and 5 have the highest cosine similarity, 0.9998, input 5 also has the lowest cosine similarity of 0.5224 with input 1. This means that input 5 is relatively important to the ANN.

Input	**1**	**2**	**3**	**4**	**5**	**6**
1	1	0.8543	0.9496	0.9371	0.5224	0.8817
2		1	0.9980	0.9934	0.9968	0.9915
3			1	0.9838	0.9998	0.9940
4				1	0.9905	0.9922
5					1	0.9963

Table 2: Cosine similarities between input hidden activation vectors

Based on the cosine similarities shown in Table 2, I ranked the relative significance of the inputs shown as Model C in Table 3. With the ranking of Models A, B, and C, I assigned weights of 1 for the most significant inputs, 0 to the least significant, and 0.2 increments for the inputs in between. After calculating each input's weight, Model F in Table 3 is the finalised ranking of relative contribution of the variables in the data set.

With inputs 2 and 3 as the least important variables, I tried two methods to adjust the ANN for better classification accuracy. First, I removed input 2 and 3 individually then together from the ANN. These changes obtained training accuracy rates of 74.38, 77.33, and 72.96 per cent, and testing accuracy rates of 75, 69.23, and 75.27 per cent respectively. For the second method, I manually adjusted the initial weights according to the relative significance, with inputs 2 and 3 much lower than the other inputs. The manual initial weights achieved a testing accuracy rate of 73.68 per cent. These two accuracy rates are both far from the 95 per cent obtained by the original study in [3].

Model	**Most Significant**					**Least Significant**
A	3	6	5	4	2	1
B	6	5	4	2	1	3
C	1	5	6	2	4	3
F	6	5	4	1	3	2

Table 3: Input ranking after functional measure

3.2 LSTM and ANN on Larger Data Set

Due to the large amount of time series data in the second data set, it was unsurprising that the LSTM would perform well. Table 4, shows the loss in percentage for different adjustments to the LSTM model. I found that with 10 hidden neurons and learning rate of 0.05, the algorithm was able to achieve the best accuracy of 96%. Figure 2 shows the loss curve for the LSTM over 5,000 iterations and learning rate of 0.05. This LSTM algorithm was able to perform just a little better than [3]'s result of 95%.

Hidden Neurons	10	10	20	20
Number of RNN Steps	50	50	50	50
Learning Rate	0.01	0.05	0.01	0.05
Number of Iterations	5000	5000	5000	5000
Loss (%)	5.34	4.27	5.39	4.29

Table 4: LSTM Loss Tests

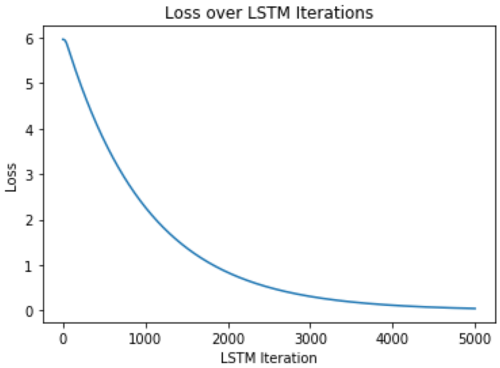


Fig. 2: LSTM Loss over 5000 Iterations with 10 Hidden Neurons and Learning Rate 0.05

To compare the LSTM model with the ANN, I used the same data set with the ANN built in the previous section. Table 5 shows the ANN testing accuracy with different number of epochs and learning rates. With the same pupil diameter data used in the LSTM, which included both eyes' data, the highest result that the ANN was able to achieve is a testing accuracy of 50%. However, with each eye's data separated the ANN was able to achieve 93% testing accuracy, which is significantly higher than the results from the first part of this study and surprisingly close to [3]'s results.

Epochs	**Learning Rate**	**Both Eyes**	**Left Eye**	**Right Eye**
3000	0.001	43%	91%	93%
3000	0.005	50%	92%	90%
1000	0.001	38%	70%	48%
1000	0.005	45%	93%	92%

Table 5: ANN Testing Accuracy Tests

4 Conclusion and Recommendation

This study used two different sizes of data set and both ANN and LSTM RNN in attempt to improve the classification accuracy from [3]. In the first part, brute force and functional measure analysis from [5] were performed on the smaller data set from [3]. After ranking the inputs, I was able to achieve the highest of 74.70 per cent testing accuracy rate by removing the least significant inputs. Although this accuracy rate is considerably high, it is nonetheless significantly lower than 95 per cent achieved by [3]. However, in the second part with the LSTM and the bigger data set, a classification accuracy of 96% was achieved. In addition, with the bigger data set, even ANN was able to achieve 93% accuracy. This shows that with the same algorithm, increasing data size can also increase accuracy.

The results from the first part agree with my hypothesis that using ANN and the technique from [5] would not improve the classification accuracy for the [3] study. I believe this lack of benefit from [5]'s methods may be due to the small number of input variables in the smaller data set from [3]. Adjusting or removing any inputs when there are only six inputs that contribute to the classification model is likely to impact the outcome greatly. Therefore, an input analysis method to improve the ANN with small number of input variables could be a future research topic.

The hypothesis for the second part was also confirmed by both LSTM and ANN achieving a significantly higher accuracy with the larger data set. For future studies on the second part of this paper, although the large size of data contributed a lot to the higher accuracy in this paper, I suggest using pruning methods to check and eliminate less significant data to improve processing time. As I could not discover a reason, another future recommendation would be to investigate why using both eyes' data achieved significantly lower accuracy than separating the two eyes' data with an ANN.

References

1. Alves, H., Valença, M.: Using Curves of Permanence to study the contribution of input variables in artificial neural network models: A new proposed methodology. BRICS Congress on Computational Intelligence & 11th Brazilian Congress on Computational Intelligence, 409–414 (2013)
2. Barlett, M. S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., Movellan, J.: Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavior. IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2**, 568-573 (2005)
3. Chen, L., Caldwell, S., Gedeon, T., Hossain, M. Z.: Are you really angry? Detecting emotion veracity as a proposed tool for interaction. Human - Nature, 412–416 (2017)
4. Empatica. 2017. For Researchers: E4 wristband. Retrieved from http://www.empatica.com.
5. Gedeon, T.: DATA MINING OF INPUTS: ANALYSING MAGNITUDE AND FUNCTIONAL MEASURES. School of Computer Science and Engineering, The University of New South Wales, (1996)
6. Greff, K.,Srivastava, R. K., Koutník, J., Steunebrink, B. R., Schmidhuber, J.: LSTM: A Search Space Odyssey. IEEE Transactions on Neural Networks and Learning Systems **28**(01), 2222–2232 (2017). https://doi.org/10.1109/TNNLS.2016.2582924.
7. Hossain, M. Z., Gedeon., T.: Discriminating Real and Posed Smiles: Human and Avatar Smiles. Human - Nature, 581 – 586 (2017). https://doi.org/10.1145/3152771.3156179
8. Hossain, M. Z., Gedeon., T.: Classifying Posed and Real Smiles from Observers' Peripheral Physiology. PervasiveHealth (2017). https://doi.org/10.1145/3154862.3154893.
9. Mather, M., Knight, M. R.: Angry Faces Get Noticed Quickly: Threat Detection is not Impaired Among Older Adults. Journal of Gerontology: PSYCHOLOGICAL SCIENCES **61B**(1), 54–57 (2006)
10. Olden, J. D., Joy, M. K., Death, R. G.: An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. Ecological Modelling **178**, 389–397 (2004)
11. Olden, J. D., Jackson, D. A.: Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. Ecological Modelling **154**, 135–150 (2002)
12. Pentoś, K.: The methods of extracting the contribution of variables in artificial neural network models – Comparison of inherent instability. Computers and Electronics in Agriculture **127**, 141–146 (2016)
13. The Eye Tribe. 2016. Our Big Mission, Retrieved from http://the.eyetribe.com.