Anger Detection Analysis By Deep Learning With Bimodal Distribution Removal

Rouyi Jin

Australian National University

Abstract. Anger is a strong emotional state and people sometimes pretend to be angry to achieve their goals. To detect the veracity of the genuine anger and the acted anger, machine based classification task is popularly employed nowadays. It is found that physiological signals, which are unintentional for humans, are favourable to other physical signals in emotion detection experiments. In this paper, artificial neural network is built and the main physiological signal - pupillary responses from the observers are extracted to an anger recognition task. To remove the noisy data points, the bimodal distribution removal technique is implemented to improve the results. Moreover, cross-validation is applied to train the dataset with generalisation and mini-batch is used to speed up the convergence and avoid the local minimum. The result confirms that the machine can identify the real and fake anger by physiological signals from the observer with a high accuracy. The mini-batch and LSTM largely boost the accuracy up to 90%, while the bimodal distribution removal leads to a smoother but slightly lower accuracy. However, rather than removing the outliers, BDR provides an early stopping for LSTM to avoid overfitting.

Keywords: Anger Detection · Artificial Neural Network · Bimodal Distribution Removal · Classification · Deep Learning · LSTM · Physiological signals · Pupillary Response.

1 Introduction

Emotions are complex states that lead to physical and physiological changes that affect behaviours, reactions and thoughts. Anger, one of the most common emotions in daily life is often considered as an intense and most destructive emotion [3]. It does not only evoke the emotional fluctuations for the expresser himself/herself, but also those who perceive the emotion.

It is proved that a neural network can identify the veracity of emotional expressions with 99.69 % accuracy [11]. In a classification task to recognize the real or fake anger, machine based method shows a greater accuracy than verbal responses. Generally, two methods are used: using physical signals or physiological signals [10]. Physical signals like facial expressions [16] and words are considered not reliable enough because it can be easily affected by subjective bias such as a fake smile. Physiological signals are consciously activated and therefore cannot be easily controlled [10]. An increasing number of empirical research [6,14] revealed that it is sufficient to detect the emotion veracity by the physiological signals of the expresser in a machine classification task.

Previous papers [6,14] are focused on the physiological signals of the expresser, however, [7] suggested that studying the perceiver's physiological signals can be another direction. More recently, [4] used perveicer's pupillary response (PR), galvanic skin response (GSR), and blood volume pulse (BVP) to classify the real and posed smiles and found a high accuracy of 93.7% from PR while 59% of verbal responses. [1] conducted an anger detection experiment by studying the pupillary responses from those who perceive the genuine or the acted anger with 95% accuracy. It is suggested that an artificial neural network can distinguish real or fake anger by learning pupillary dilation from different observers [12]. All of the results indicated that the pupillary response can predict the real smile and genuine anger at a high accuracy.

However, it is possible that the noisy data points in the training set will affect the model. Detecting the outliers and remove them is one of the efficient way to reduce the negative impacts. Thus, to further explore the efficiency of classifying the veracity of the anger by observer's physiological signals, this paper adopts the similar data structure with [1,4] in different techniques and models to analyze.

Firstly, a simple feedforward network is built and later improved by long-short term memory (LSTM) network, which is a recurrent neural network. To differentiate the two networks, I will name the simple feedworward network as feedforward in the paper and the other is LSTM. Mini-batch method to avoid local minimum and k-fold cross-validation to fully use the dataset are applied. Furthermore, the bimodal distribution removal technique is implemented so that some outliers can be removed without further influence in the training. Lastly, the effectiveness of BDR on both networks will be illustrated along with the exploration of reasons.

2 Methods

The recognition of genuine anger and fake anger is formulated into a time series classification task and I design different neural networks for this problem. It is aimed to figure out the effectiveness of the network and the

influence of removing the outliers during the training. A detailed network architecture is shown in Fig. 1. The raw data is preprocessed before the normalization. During the training, an outlier detection technique is applied and the training stops when the loss function converges so as to avoid over-fitting.

This section will briefly explain the dataset used in the experiment (section 2.1) and how I deal with the raw data (section 2.2). In the training part, a one layer long-short memory network is used (section 2.3) and the loss function with backpropagation and optimizer are introduced (section 2.4). Some other techniques, namely bimodal distribution removal (section 2.5), K-fold cross-validation and mini-batch (section 2.6) are implemented to ameliorate the network. Lastly, evaluation methods will be recommended to measure how well the model is in this anger classification task (section 2.7).

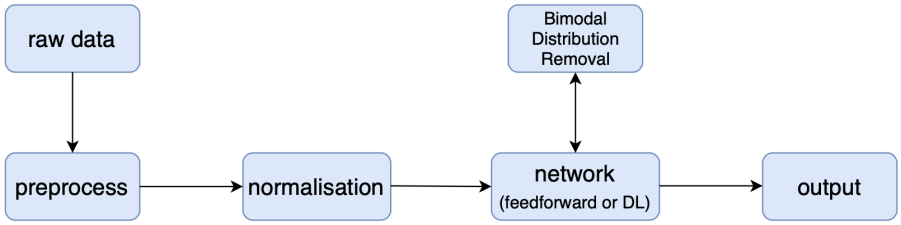

Fig. 1: Overall Architecture of the Network Implemented

2.1 Dataset Description

In this paper, the anger(timeseries) dataset is used for two reasons. Firstly, since detecting the emotion can be largely used in crime investigations, psychology and other industries [6], recognizing the genuine or fake emotion is important. And computer based emotion detection is much more accurate than humans [1]. Secondly, most of the previous research focused on the expresser's physiological signal changes while this dataset is the signals collected from the observers, which is interesting and novel. I am willing to figure out whether the observer's physiological signal could be trained to tell the veracity of anger.

The dataset used is the pupillary diameter of each participants' left and right eyes. The data was recorded while watching 20 videos - 10 of them are genuine anger and 10 fake anger. The pupillary diameter is recorded every $\frac{1}{60}$ seconds. And if the the participants eyes blink, 0 is used. However, not everyone watched all the videos, thus there are total 390 samples with different lengths due to the different videos.

2.2 Data Preprocessing

In the raw data, since the length of videos are different, the 0 padding is used to fill in to keep the length at the same level. The mean pupillary diameter of left and right eyes is used, however, there are eye blinks during the experiment. It is noticed that sometimes only the left eye is recorded while sometimes only the right eye is recorded. Instead of using the average of the participates watching the same video, I use the left data to fill the right and vice versa. Because the pupillary diameter has a large difference among the multi-cultural background participants and the data collected from the same person in the meantime should not lead to large difference.

Since all the raw data are numerical, it is convenient to normalize. Normalization plays an important role before training the data in a classification task in neural network. As Jayalakshmi [5] mentioned, normalization process can greatly affect on preparing the input data to be suitable for training and also accelerates the training. There are two common methods to process normalization. Formula 1 is the z-score normalization (standardization) and formula 2 is called min-max normalization.

$$x = \frac{x - \mu}{\sigma} \tag{1}$$

$$x = \frac{x - min}{max - min} \tag{2}$$

The min-max normalization uses the same scale while the outlier can have a large impact. However, the z-score normalization considers the influence from outliers but does not produce with the exact same scale. Therefore, due to the large difference among different features, I use z-score normalization for all the input features and it is calculated by formula 1, where μ is the mean of the feature and σ is the standard deviation.

2.3 Long-Short Term Memory (LSTM)

Recurrent neural network is one of the artificial neural networks and usually used in sequence learning. Compared with feed-forward neural networks, RNN provides feedback, which is capable of learning short term dependencies. Because the cell is used to remember the short-term values. However, sometimes longer dependencies are required where the LSTM network makes it possible. Long-short term memory is an artificial recurrent neural network (RNN) architecture, which is commonly used in deep learning field. Therefore, the best way to solve this time series classification problem is LSTM.

A general LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. And the LSTM cells are the hidden layer for the network. In this paper, 1 hidden layer is used. The biggest difference between LSTM and RNN is LSTM solves the longer term dependencies. For example, say a RNN can generate y3 based on x1 and x2, however if y10 is related to x1 and x2, the RNN can not effectively solve it. The LSTM enables to work out this situation. To avoid the overfitting, a dropout can be added in the output layer. Dropout is a regularization technique that drops neurons.

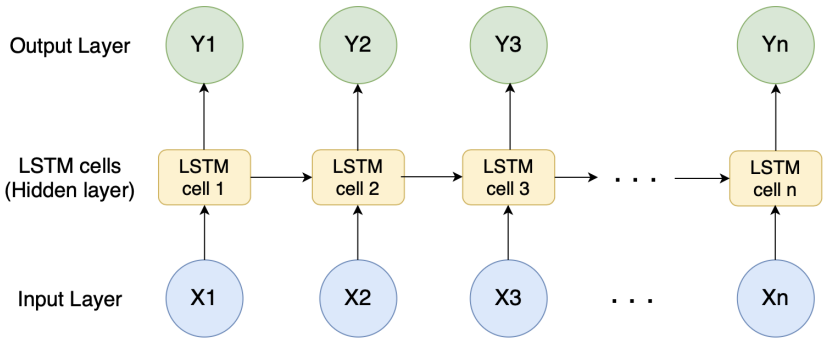

Fig. 2: Detailed Structure of LSTM network

2.4 Loss Function and Optimizer

The loss function calculates the difference between the expected output and the predicted output. There are several commonly used loss functions, namely mean squared error(MSE) and cross-entropy, that can be applied. It was shown that the cross-entropy has significant advantage over the MSE method in a classification task [9]. Thus the cross-entropy is used in this paper. It is calculated by formula 3, where N is the total number of training set, y_i is the real label and p_i is the output of the neuron. For example, if the label y_i is 1, $(1-y_i)log(1-p_i)$ is 0 and try to make $p_i = 1$ to minimize the loss. In minimizing the loss, the backpropagation can not be ignored.

$$L = \frac{1}{N}\sum_{i} -[y_i log(p_i) + (1-y_i) log(1-p_i)] \tag{3}$$

The backpropagation is a computation on the gradient of the loss function with respect to the weights and then updates the weights in the network. It can be considered as an optimization process. The weight is iteratively updated by deducting the calculated gradient times the learning rate. The learning rate is the determination of the step size in each iteration towards to the minimum loss. A large learning rate will possibly lead to a divergent behaviour in the loss function.

An optimizer is tied with the loss function so as to update the model in response to the output of the loss function. I adopt the first-order gradient-based optimization: Adam (adaptive moment estimation) algorithm, which is implemented straight forward [8]. Compared with SGD optimization, Adam only requires the first-order derivative that cost much less memory [8] and it performs much faster.

2.5 Bimodal Distribution Removal Technique

Outliers can largely affect the training in a neural network and it is important to identify and remove them. Bimodal distribution removal (BDR) is one of the methods that can clean up the noisy training sets during the training and provide a halting criterion to prevent overfitting [15].

The BDR is not started until the normalised variance of errors over the training set is below 0.1. Then take those with a value higher than the average error (δ_{ts}) as a subset and calculate the mean (δ_{ss}) and the standard deviation (σ_{ss}) of this subset. The criterion of permanently remove the recognized noisy data is the patterns from the subset with an error $\geq \delta_{ss} + \alpha\sigma_{ss}$ where α is in a closed interval between 0 and 1. The removal is

repeated every 50 training epochs until the normalised variance of errors is less than 0.01. And thus the whole training will be halted [15].

The idea of BDR comes from the frequency distributions of errors for all the training patterns and usually there will be a large variance in the beginning [15]. With the training moves on, most of the errors drop very quickly while a small amount of the errors remain, which are considered as outliers.

2.6 K-fold Cross-Validation and Mini-batch

K-fold validation enables each data point to have a chance to be validated against through crossing over the training set and validation sets in a successive mode [13]. In this paper, 3-fold cross-validation is used. Firstly, the dataset is divided into 3 splits randomly with similar sizes. Each set is used as validation test once and the rest sets are for training. The average of 3 validation tests is considered as the final generalised accuracy. Thus, all the data are used for both training and test, contributing to an unbiased result.

The mini-batch method is also used in training because it speeds up the convergence [2]. Mini-batch is a variation of the stochastic gradient descent method that splits the training set into small batches and each batch is trained to calculate the losses and update the weight. It enables the model to update the batch gradient in a higher frequency such that a more robust convergence is achieved and the local minima can be avoided. It also offsets the disadvantages in stochastic gradient descent and batch gradient descent, enabling a more computationally efficient process with robust convergence.

2.7 Evaluation method

Training the model plays a significant role in classification, however, the evaluation methods are absolutely indispensable. Other than the accuracy, I take F1 score to evaluate the model.

	Genuine (Predicted)	Acted (Predicted)
Genuine (Actual)	True Positive (TP)	False Negative (FN)
Acted (Actual)	False Positive (FP)	True Negative (TN)

Table 1: Confusion Matrix for F1-Score Evaluation

True Positive (TP): Correctly predict the genuine anger.

True Negative (TN): Correctly predict the acted anger.

False Positives (FP): When the actual is acted anger, but predicted as genuine anger.

False Negatives (FN): When the actual is genuine anger, but predicted as acted anger.

Accuracy is the ratio of correct predicted observations to total observations. Precision is the ratio of correctly predicted genuine anger to the total number of prediction as genuine. Recall is the ratio of correctly predicted genuine anger to the total number of actual genuine anger.

F1 score is the weighted average of precision and recall as calculated by formula 4.

$$\text{F1 Score} = 2 \times \frac{recall \times precision}{recall + precision} \tag{4}$$

By evaluating the model with F1 score and accuracy, we would know how well the model explains the dataset.

3 Results and Discussion

This section will evaluate a feedforward neural network and a LSTM network and discuss the effectiveness and impact of the techniques mentioned in section 2. And it mainly answers 3 questions below:

(1) What is the best combination of two layers hidden neurons in feedforward network? How the batch size affects the result and running time in LSTM?

(2) How does the minibatch and BDR technique influence the feedforward model? What about the performance in LSTM network and the impact of BDR technique on LSTM?

(3) Does the BDR technique effectively remove the outliers during the training process and result in a better accuracy in predicting the anger? If not, what are the possible reasons that the BDR is not working?

3.1 Hyper-parameter Tuning

Hyper-parameters in the neural network play an essential role. It can largely affect the running time and the final results. In this experiment, I will tune the hidden units for the feedforward network and the batch size for the LSTM. The accuracy is averaged by 6 experiments and each accuracy is the average of k folds.

hidden units	10 x 5	10 x 10	15 x 10	**20 x 10**	20 x 15
accuracy(%)	72.39	71.86	69.28	**76.01**	68.77

Table 2: Tuning the Number of Neurons in Two Hidden Layers in Combination

Experiments	batch size = 16		batch size = 8		batch size = 4	
	Accuracy(%)	Time(s)	Accuracy(%)	Time(s)	Accuracy(%)	Time(s)
1	89.12	15.49	90.33	25.33	90.17	50.12
2	82.67	16.19	87.67	28.00	89.33	47.86
3	85.33	16.45	89.67	27.32	91.24	46.83
4	86.33	16.39	90.67	27.64	90.33	50.83
5	87.21	16.81	91.67	27.33	88.67	51.14
6	80.33	15.19	89.67	27.05	89.33	49.92
average	85.11	16.27	89.94	27.12	89.78	46.36

Table 3: Tuning the batch size of LSTM network

The bold columns in table 2 are the outstanding hyper-parameters for feedforward network. It is obvious that the 20 x 10 combination for the hidden units produces the best results. From table 3, when the minibatch size is 4 and 8, the accuracy do not have large difference. However, considering the running time, size 8 takes 27.12 seconds on average to run 3 folds cross-validation, which is half of the time spent when size is 4. Thus, the mini batch size is set to be 8.

3.2 Evaluation of Feedforward Network

Motivation: Before implementing the RNN, exploration on feedforward network is presented. The dataset used is processed from the raw data. Specifically, the features include the mean and standard deviation of pupillary, mean of the absolute values of the first and second differences of the processed signals [4] and another 2 features are extracted from principle component analysis. It is still a balanced dataset.

In a stochastic gradient descent network training, it has a tendency to stuck in a local minimum. Mini-batch means the training only takes a subset of the data during one iteration and some noisy steps would be added in face of a local minimum. This is why mini-batch is popularly used these days. In my experiment, the mini-batch is firstly considered to improve the original network.

Experiment Design: There are two models for the experiment: a normal ANN and a normal ANN with mini-batch training. Both of them contain 20 neurons in the first hidden layer and 10 neurons in the second hidden layer. In the mini-batch model, a 32 batch size is set for learning in each iteration. The networks are trained with the same hyper-parameters such as 0.01 learning rate and 500 epochs to have a better comparison.

To evaluate the two models, the accuracy and F1-Score are recorded. To generalise the models, I also implement 5-fold cross-validation in both designs, which generates a more unbiased result. The accuracy and F1-Score are calculated on the average of the validation tests in 5 models due to the 5-fold cross-validation. It is expected that by adding the mini-batch method, the result would be better and the model should have a generalisation. The last two columns in the table is an ANN with minibatch and BDR so that the comparison would be more clear in section 3.4.

Results: Table 4 shows the accuracy and F1 score in different experiments. It is apparent that the mini-batch in the training greatly improves the performance. The normal ANN has a small variance in the accuracy but much higher variance in F1 score. The average of a normal ANN is about 65.6% accuracy and 64.3% of F1 score while adding the minibatch increases the result by 13% in both evaluations.

The noteworthy improvement from mini-batch can be explained by the sufficient avoidance of local minimum. The normal training method can be stuck in a local minima and some noisy gradients are needed to jump out of a local minimum of the loss function and heading to the global minimum.

Experiments(%)	Normal ANN		ANN with minibatch		minibatch + BDR	
	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score
1	64.89	65.69	81.38	80.65	73.02	71.06
2	64.54	65.83	82.33	81.31	71.24	70.19
3	67.13	65.07	77.42	75.97	69.74	68.08
4	64.54	60.29	75.74	74.38	72.26	70.99
5	66.49	67.21	75.75	75.02	74.26	72.79
6	65.81	61.91	78.11	77.49	72.60	70.69
average	65.57	64.33	78.46	77.47	72.10	70.62
stdev	0.012	0.070	0.079	0.085	0.016	0.015

Table 4: Accuracy and F1-score for ANN and ANN with Mini-Batch and Mini-Batch with BDR

3.3 Evaluation of LSTM Network

Motivation: Nowadays, based on various backgrounds, the performance of different neural networks can be varied. For example, convolutional neural network is commonly used in image data while recurrent neural network is suitable for sequence input data.

The raw data are collected in a time series, which indicates the data has dependencies. Thus, a recurrent neural network should be considered. However, it is noticed that the dimension of data is large and RNN is involved in gradient vanishing and exploding problem. LSTM can greatly avoid those issues and therefore considered as the most suitable model for this anger detection classification problem.

Experiment Design: The LSTM network contains only one hidden layer and the hidden size is set to be 35. Considering the largest input has a length of 186, and the time interval is $\frac{1}{60}$ second, the input size is set as 31, which is around 0.5 second. Thus, the time step is 6 for each sample. The learning rate is 0.01 and the total epoch is 100 because the LSTM model converges in a much smaller training epochs. The mini-batch is also implemented during the training and the batch size is set at 8.

I use a 3-fold cross-validation so as to generalise the results. The dataset is split into 3 folds and each time one of the fold is selected to be the test set. The recorded accuracy and f1-score in table 5 are the average of the 3 tests. The running time is also recorded to show the effect of BDR. The time is the total time for 3 folds of running.

The result for LSTM with BDR technique is in the table for the convenience of comparison in section 3.4.

Experiments	LSTM			LSTM + BDR		
	Accuracy(%)	F1 score(%)	Time(s)	Accuracy(%)	F1 score(%)	Time(s)
1	89.67	93.19	25.33	87.33	88.52	12.68
2	90.67	92.31	28.00	86.33	86.14	13.69
3	90.33	91.13	27.32	86.33	86.18	13.69
4	89.67	90.79	27.64	86.33	86.14	14.59
5	89.33	92.61	27.33	88.63	90.88	19.13
6	92.00	93.42	27.05	89.40	91.13	15.29
average	90.28	92.09	27.12	87.22	88.16	14.76
stdev	0.010	0.010	0.93	0.011	0.013	2.53

Table 5: Accuracy and F1-score for LSTM and LSTM with BDR

Results: From table 5, it is apparently that the LSTM has a remarkable improvement with a 12% increment on average compared with feedforward network. LSTM does not only increase the accuracy and f1 score, but also the stabilizes the evaluation. This is because the variance of the accuracy in feedforward is 0.07 while in LSTM, it is 0.01. The reason for this rapid increase can be the dependencies of the input playing an important role in training.

3.4 Effectiveness of BDR

Motivation: It is general to have some outliers in the dataset when conducting an experiment, and multiple techniques such as absolute criterion method and least median squares are deployed to minimize the impact

from those noisy points. Bimodal distribution removal is designed for outlier detection and addresses all the weaknesses of the methods mentioned earlier [15].

Experiment Design: Based on the mini-batch and 5-fold cross-validation in feedforward, 3-fold cross-validation in LSTM, I further add the BDR to remove the outliers in the anger dataset. However, to find the suitable α, which is the coefficient for the subset in permanently removing the pattern (i.e. α is in the decision formula: error $\geq \delta_{ss} + \alpha\sigma_{ss}$), I firstly have an experiment in α selection. All the hyper-parameters are the same as previous experiments to control variables: the hidden neurons are 20 x 10 for the two layers, the learning rate is 0.01, batch size for mini-batch is 32 in feedforward. And in LSTM, there is one hidden layer, batch size is 8, with 0.01 learning rate and 35 hidden dimensions. After choosing the α, the model is built to make comparisons. It is expected to stop the model earlier and the result should be better. In table 6, it is obvious that the α does not tell a large difference in LSTM model. From the feedforward row, both 0.8 and 0.9 are extraordinary, however, the 0.8 has a larger standard deviation through multiple trainings (0.8 has a standard deviation of 0.03 in accuracy while 0.9 is 0.015) and thus α is set to be 0.9.

α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
feedforward accuracy(%)	66.19	69.87	70.01	71.90	71.46	70.21	71.82	67.88	72.11	72.10	71.86
LSTM accuracy(%)	90.20	86.67	90.03	88.67	89.03	90.23	91.01	89.33	90.67	89.67	89.33

Table 6: Selecting the Coefficient for Variance used in Outlier Detection

Results and discussion: It is expected that the BDR has two functionalities: first, it should effectively remove the detected outliers to improve the performance, and second, after removing the noisy data points, the BDR provides an early stopping when the model is convergent.

In the feedforward neural network, training with a BDR technique does not improve the model, on the contrary, the result is lowered by 6%. Additionally, there is no early stopping noticed during the experiments in spite of the converged error. Both of the results reject the hypothesis that the result would be improved while the training would be stopped once the loss function is converged. However, the BDR smooths the result because adding the BDR generates a much smaller standard deviation at 0.016 compared with 0.079 in a normal ANN with mini-batch. This means the BDR generates a more stable result.

In LSTM, the time of the model with BDR (14.76s) is almost half of the time used without BDR(27.12s), which indicates an early stopping happened. The early stopping can avoid overfitting problems. Thus, though the accuracy and f1 score are both decreased about 3%, the overfitting is avoided. However, by inspections on the number of data points, there is no removal actions observed.

3.5 Discuss the Reasons on Under Performed BDR

Error Distribution During Training: Based on previous experiments, in the feedforward network, the BDR removes some data points but there is no early stopping. In LSTM, BDR does not remove any data but do halt the training when error converges.

To further investigate the reasons that BDR is not working as expected, I take a look at the error distribution in the very beginning (Fig.3, 5) as well as the one during training (Fig.4, 6). The two error distributions contain all the errors in the training. Those error distributions are found to be common through multiple experiments and thus I consider them as generalised error distribution. Surprisingly, the error distribution before the training in both networks shows a great normalization, implying that the dataset is not very noisy. Moreover, the error distribution of feedforward network at epoch 450 shows a bimodal error distribution. In LSTM, the error at epoch 30, which is an early stage, does not show a bimodal distribution, indicating the network is training well. However, as stated in [15], the BDR is employed in a training set whose error distribution in the early training (0-100 epochs) is almost bimodal and the variance will drop sharply after a few training epochs. Both of the results are on the contrary of the explanations in [15]. Therefore, I generate the 3 hypothesis. Since the LSTM is halted early, the third hypothesis is for the feedforward.

Hypothesis 1: the model is well trained without outliers and error converges at an early stage, where the BDR does not have contributions.

Hypothesis 2: the variance of normally distributed error at epoch 0 is not large and model will step into the outlier detection in the beginning of the training. This will lead to the model be under trained.

Hypothesis 3: the training will not stop though it is converged because the variance is still larger than 0.01 due to the bimodal error distribution.

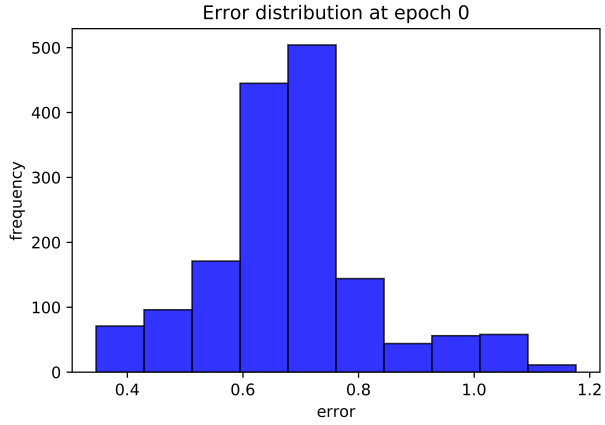

Fig. 3: Training Error Distribution at epoch0 normal

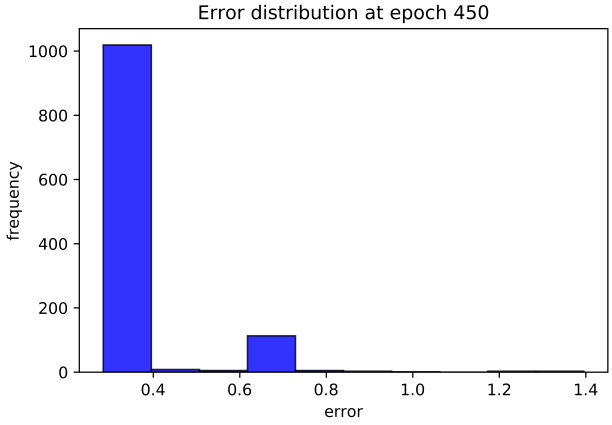

Fig. 4: Training Error Distribution at epoch450 normal

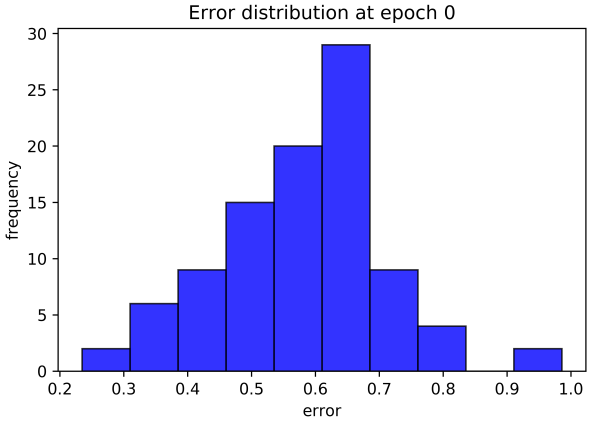

Fig. 5: Training Error Distribution at epoch0 LSTM

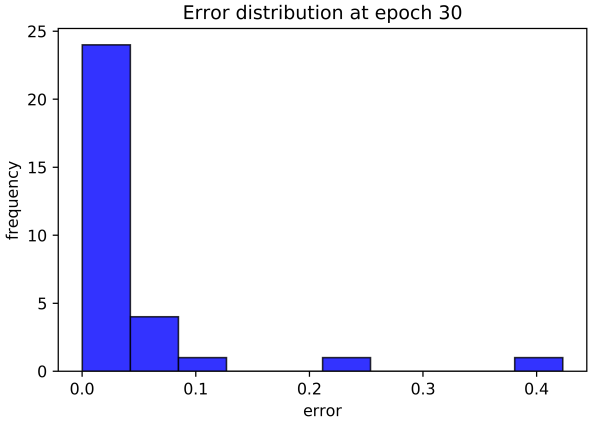

Fig. 6: Training Error Distribution at epoch30 LSTM

Model is Well-Trained before BDR: This part mainly focuses on the LSTM model. Because in feedforward network, the BDR effectively removes some outliers.

The BDR technique is triggered when the variance is between 0.01 and 0.1. To learn the evolvement of LSTM variance during the training, Fig. 7 is drawn. It is noticed that the variance is decreasing overall during the training until the error is converged after about 40 epochs, where the variance is less than 0.01. Considering the error distribution in Fig. 6, after 30 epochs, the distribution shows a great tendency to convergence. There is no bimodal error distribution and t LSTM is training on the track. The two figures reveal that the LSTM is well trained and the BDR provides early stopping rather than removing the noisy data points. When the data is explained by the model in a short time, then it is less likely to contain outliers. This accepts the first hypothesis.

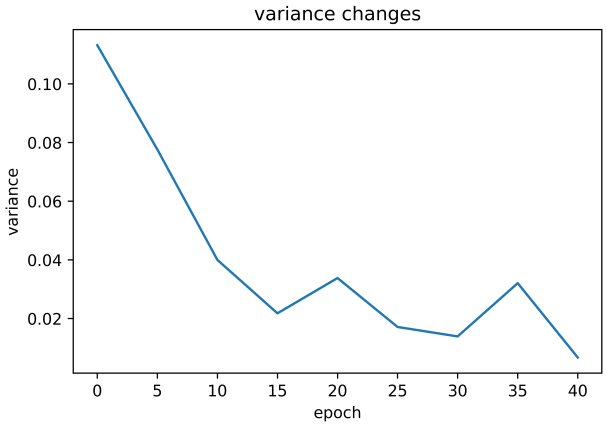

Fig. 7: Variance for One Fold with BDR in LSTM

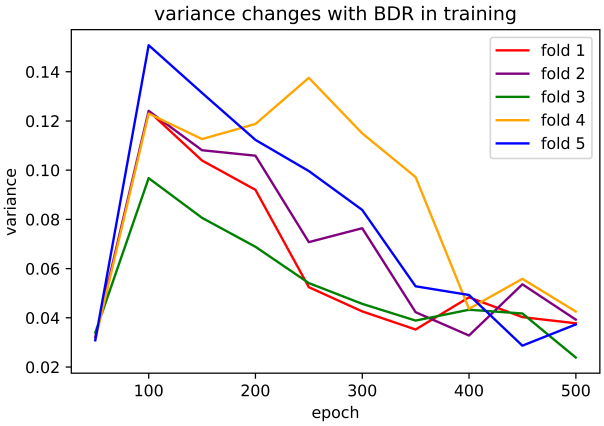

Fig. 8: Variance for Each Fold with BDR in feedforward

Ineffective Termination and Early Outlier Detection: Since it is noticed that in with a BDR in LSTM, the training process is halted when the variance is less than 0.01. Thus, this part will analyze situations in feedforward network.

To validate my hypothesis made, the error variance in the very beginning is tracked and all of the variances are greater than 0.01 and most of them are less than 0.1. It accepts the second hypothesis that the training steps into the outlier detection in the beginning. Starting the outlier detection in a early stage will possibly remove some important data points that can well train the model.

Besides, the bimodal error distribution will always be achieved between epoch 100 and 200, where loss function is converged. While running the experiments with BDR, I record the variance changes for each cross-validation set and generate the Fig.8. The variances during the training are always between 0.01 and 0.1 and converges at 0.02, indicating that the training will not be halted and the training will always step into outlier detection every 50 epoch. This validates the hypothesis 3 made earlier that the training is not stopped when it should be due to the large variance caused by bimodal error distribution.

The last two hypotheses made are accepted by inspecting the Fig.8. This result forms a preliminary knowledge that this dataset is not noisy before the training in a neural network and thus, the BDR is not very suitable to enhance the performance.

Tesing Accuracy With Remaining Data Numbers: To further investigate the accuracy change in feedforward, I explore the relationship between the remaining number of data points and the accuracy. I expect that adding the BDR would improve overall performance through removing the correct noisy points. Thus, if the BDR is efficient, then the accuracy will be higher when there is less training input, indicating a negative coefficient of the relationship is desired. However, from Fig. 9, the relationship is be explained by $y = 0.00025x + 65.59\%$ with a R-square 0.039. It reveals that there is a positive relationship but only explains 3.9% of the accuracy variance by the remaining inputs. Therefore, I reject the original hypothesis that BDR can improve the testing accuracy by removing the noisy points.

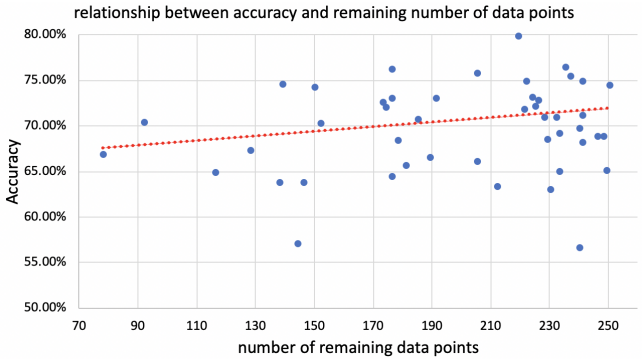

Fig. 9: Relationship between Testing Accuracy and Number of Data points Remaining

4 Conclusion and Future Work

This paper evaluates the effectiveness of a two-hidden layer neural network and a LSTM network with the help of an outlier detection technique named bimodal distribution removal in a time series classification task. I choose the k-fold cross-validation to make the results more generalised and also determine whether the mini-batch would improve the overall performance. It is shown that LSTM performs much better than the feedforward network. Lastly, I concentrate on the impact analysis based on the performance of BDR.

The mini-batch is proved to be advantageous in improving the accuracy. From the comparisons of two kinds of neural networks, the LSTM stands out with a high accuracy at 90%. However, compared with the higher accuracy (95%) in previous research [1], more techniques can be employed to improve the current accuracy in the future. For example, in the data pre-processing stage, the uneven in data length is simply solved by 0 padding, which is possible to have impacts in the training. It would be better to pack the padded sequence so that the actual length of data is used. Moreover, the data input could also be considered from other perspectives. It is recommended to change the input dimension from 1-D to 2-D to train the network. Specifically, instead of using the mean of pupillary diameter of left and right eyes, the input can be (left, right).

In the second experiment, I expect the BDR could halt the training when the error converges and improve the accuracy at the same time. However, the result does not show any enhancement in accuracy though it does stop training when the error converges but only in LSTM model. To figure out the reasons that BDR is not working as expected, a deeper analysis to the error distribution and the variance evolvement during the training process is conducted. As a result, the normal error distribution at the beginning indicates the training set is not noisy and the BDR may not be suitable. And the bimodal error distribution in the feedforward training suggests the variance is still large when the loss is converged. The halted LSTM indicates the model has been well trained in a early stage before the necessary implementation of BDR. Lastly, the positive relationship between the number of remaining inputs and accuracy demonstrates that a higher accuracy will be caused by less data points removed by BDR. It is suggested to check the error distribution before implementing BDR.

References

1. Chen, L., Gedeon, T., Hossain, M.Z., Caldwell, S.: Are you really angry? detecting emotion veracity as a proposed tool for interaction. In: Proceedings of the 29th Australian Conference on Computer-Human Interaction. p. 412–416. OZCHI '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3152771.3156147, https://doi.org/10.1145/3152771.3156147
2. Cotter, A., Shamir, O., Srebro, N., Sridharan, K.: Better mini-batch algorithms via accelerated gradient methods. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 24, pp. 1647–1655. Curran Associates, Inc. (2011), http://papers.nips.cc/paper/4432-better-mini-batch-algorithms-via-accelerated-gradient-methods.pdf
3. Ellis, A.: Anger: How to live with and without it. Hachette UK (2019)
4. Hossain, M.Z., Gedeon, T.: Classifying posed and real smiles from observers' peripheral physiology. In: Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare. p. 460–463. PervasiveHealth '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3154862.3154893, https://doi.org/10.1145/3154862.3154893
5. Jayalakshmi, T., Santhakumaran, A.: Statistical normalization and back propagation for classification. International Journal of Computer Theory and Engineering **3**(1), 1793–8201 (2011)
6. Kim, C.J., Chang, M.: Actual emotion and false emotion classification by physiological signal. In: 2015 8th International Conference on Signal Processing, Image Processing and Pattern Recognition (SIP). pp. 21–24 (2015)
7. Kim, J., André, E.: Emotion recognition based on physiological changes in music listening. IEEE transactions on pattern analysis and machine intelligence **30**(12), 2067–2083 (2008)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Kline, M., Berardi, L.: Revisiting squared-error and cross-entropy functions for training neural network classifiers. Neural Comput. Appl. **14**(4), 310–318 (Dec 2005). https://doi.org/10.1007/s00521-005-0467-y, https://doi.org/10.1007/s00521-005-0467-y
10. Lin Shu, Jinyan Xie, M.Y.Z.L.Z.L.D.L.X.X., Yang, X.: A review of emotion recognition using physiological signals. Sensors (Basel) **18**(7) (2018)
11. Qin, Z., Gedeon, T., Caldwell, S.: Neural networks assist crowd predictions in discerning the veracity of emotional expressions. In: International Conference on Neural Information Processing. pp. 205–216. Springer (2018)
12. Qin, Z., Gedeon, T., Chen, L., Zhu, X., Hossain, M.Z.: Artificial neural networks can distinguish genuine and acted anger by synthesizing pupillary dilation signals from different participants. In: International Conference on Neural Information Processing. pp. 299–310. Springer (2018)
13. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-Validation, pp. 532–538. Springer US, Boston, MA (2009). https://doi.org/10.1007/978-0-387-39940-9$_5$65, https://doi.org/10.1007/978-0-387-39940-9_565
14. Santamaria-Granados, L., Munoz-Organero, M., Ramirez-González, G., Abdulhay, E., Arunkumar, N.: Using deep convolutional neural network for emotion detection on a physiological signals dataset (amigos). IEEE Access **7**, 57–67 (2019)
15. Slade, P., Gedeon, T.D.: Bimodal distribution removal. In: International Workshop on Artificial Neural Networks. pp. 249–254. Springer (1993)
16. Zhang, Y., Yang, Z., Lu, H., Zhou, X., Phillips, P., Liu, Q., Wang, S.: Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation. IEEE Access **4**, 8375–8385 (2016)