Deception Detection using Neural Network with Bimodal Distribution Removal and Genetic Algorithm

Xueyuan Tang[1],

[1] Research School of Computer Science, Australian National University, u6022988@anu.edu.aum

Abstract. Gaining information for facial images is a popular area these years. By Granger causality method, previous study has extract 20 features from facial thermal imaging and are proven to be effectively connective in deception detection. The ability to make accurate classification of deception becomes necessary and achievable for neural network area. Bimodal distribution removal is a method to clean up outliers while training. It can not only overcome shortcoming of traditional removal methods, but also has higher training speed and can terminate training automatically. And genetic algorithm can optimize initial parameters of network to overcome the too quick halting problem of BDR and perform effectively in backpropagating the gradients. After applying the two methods, the resulted model shows clearly higher accuracy.

Keywords: Bimodal distribution removal, Genetic Algorithm, Neural network, deception detection,

facial thermal imaging, Granger causality

1 Introduction

Some researchers (Amin, Mohammad, Ali and Tom) [1] have applied a modified multivariate Granger causality (GC) method to quantify the effective connectivity of facial region with deceptive anxiety. In their research, 20 causality features were extracted from thermal images which are collected by a mock crime scenario. Four traditional machine learning classifiers based on those features were proven to have more than 87% accuracy on detecting deception.

However, there is no method of cleaning outliers mentioned in Amin's paper after extracting features. And no network-based classifiers have been implemented for comparison, which makes the accuracy possibly not close enough to optimal.

Neural network classifier is believed to have higher robustness in dealing with loss and noise of input data and be close to optimal complicate non-linear relationship (Thomas, 2019) [2]. As a discriminative model, it is expected to obtain different performance with models of Amin's team. Therefore, it is a valuable attempt to implement such a model for comparison.

The traditional noise removal methods, such as Least Median Squares and Least Trimmed Squares, are considered to be unsuitable in this case due to the lack of adequate patterns (only 31 objects are collected) and knowledge of priori on the number of outliers, or overfitting caused by unstably increase of training time.

Bimodal distribution removal (BDR) is introduced here to overcome the shortcomings above. Firstly, this noise removal method allows network to identify and remove outliers while training rather than preprocessing of train set. So valuable information of outliers can be extracted before removal to make full use of small size of objects. Secondly, BDR provides a halting criterion to stop training earlier, which can dramatically decrease training time to prevent overfitting.

Furthermore, genetic algorithm (GA) is introduced as a method to optimize initial parameters including weights and biases. Inappropriate initialization can cause vanishing or exploding gradient, and makes network hard to converge consequently (Poonam, 2019)[5]. What is more serious is that the halting criterion of BDR can let training stop unexpectedly if initial parameters cause variance of error too small from the beginning. Applying GA ensures our model will process gradient decrease efficiently and experience enough times of epoch in training.

Design of problem:

- Topic: Implement a neural network classifier with BDR and Genetic Algorithm to discriminate between deceptive and truthful subjects based on 31 objects from Amin's research[1].
- Input to model: 20*1 features which are extracted by GC method from a thermal image.

- Output from model: 0-1 Boolean value based on single double value output from network. Equal 1 if the input object is classified to be deception group.

This work facilitates the following:

- Build a basic 4-layer Multilayer Perceptron (two hidden layers) classifier for deception detection. Provide a comparison target with Amin's model[1] and further optimized models.
- Basic optimal tools are implemented at first to indicate general effect of neural network. Use average of classification accuracies in 20 times of built model with same hyperparameter to find proper optimal methods. Use the final version as a simple result of traditional neural network without BDR.
- Implement BDR to the final version. Compare the results and analyse the advantages and shortcomings of BDR in deception detection.
- Implement GA to the improved version. Compare the results and analyse the advantages and shortcomings of BDR in deception detection.
- Generally conclude the feasibility of neural network in deception detection. Provide constructive suggestions on improvements of such models.

2 Methods

2.1 Assumptions.

Basic network structure:
In this paper we will utilize a feed-forward MLP of four layers. The structure will be fully connected with no lateral, backward or multilayer connections. Cross Entropy error will be used in loss function and back-propagation will be used in training step. ReLU and Sigmoid function is used as activation function for hidden layers and output layer respectively. Training will stop if loss change proportion is less than 0.001.

Performance measure:
10-fold cross validation is implemented. For each test, we will repeat 10-fold cross validation for 5 to 50 times based on time cost and utilize the average of output (mean error and accuracy for valid set) as the final scores of a model.

2.2 Normal Neural network.

Firstly, pre-process for data is executed, including checking if dataset is balanced, 0-1 scaling, z-score and normalization. After observing train set, apply these methods if they are believed to be suitable to the situation in this paper. Compare the accuracy before and after implementation to decide whether it should be kept for further experiment.

Secondly, the number N, M of neurons in first and second hidden layers in the network is decided, while the input layer has 20 neurons and output layer has 1 neuron to satisfy the input and output size. In this section, N={2, 3,…,17} and M={1, 2,…,16} was tested by pair. Learning rate is set 0.05 by default. Training will also stop if epoch reaches maximum (500). Compare the performance of different pair of (N, M) to choose the one which is close to optimal. Meanwhile, adjust upbound of epoch if necessary.

Note that implementing other traditional network tricks like L1, L2 normalize and Dropout is still encouraged. However, this paper will focus on the impact of BDR and GA, so such methods will not be discussed specifically.

2.3 BDR method for noise clean

The basic idea of BDR is use error if every pattern to build a frequency curve to find a Bimodal distribution (Slade & Gedeon) [4]. The patterns around the left (usually high) peak are believed to be outliers. Those outliers can be identified by variance and mean value of error effectively. After certain times (50 by default) of epoch, they will be permanently removed from train set. But since in early training they will still be used, useful information from those outliers will still be collected. Furthermore, when variance of error is lower than a threshold (0.01 by default), training is halted to prevent overfitting.

Parameters will be used as the same as the default values in Slade's research. Every 50 epoch, if variance of error is lower than 0.1, BDR will be executed. In the first step, pattern with error larger than mean error will be taken out. For the subset, a new mean error value v_ss will be calculated. Patterns whose error e_p satisfies formula (1) will be identified as outliers and removed from training set.

$$e_p \geq v_{ss} + \alpha * t_{ss}, \qquad (1)$$

where t_ss is the standard deviation of the subset. α is a parameter related to threshold of outlier identification and satisfies $0 \leqslant \alpha \leqslant 1$.
Furthermore, training will be stopped if the variance of error v_ss satisfies formula (2).

$$threshold \geq v_{ss}, \qquad (2)$$

where threshold decided the minim value of variance that allows continuing training.

Firstly, an error curve of normal network model should be drawn to check if there exists bimodal which satisfies the apply conditions of BDR. If there exist two peaks, implement BDR structure. Then, tests should be done with different value of parameter α in formula (1) to detect the value which is close to optimal. Compare with former version to see the impact on accuracy. Additionally, draw the frequency of objects which are preserved (identified as not outliers) chart and observe for analyse the role of identifying outliers. Finally, compare the final model performance with that of Amin's team[1].

2.4 GA method for initialization of weights and biases between neurons.

Genetic algorithm is a search heuristic which reflects the process of natural selection (Vijini, 2020)[6]. The basic idea is to select parents which have higher fitness score, use them to reproduce new population (children) to find the best individual for certain environment (problem). The algorithm can be generally divided into five phases, initial population, fitness function, selection, crossover and mutation. More details about terminologies can refer to *Introduction to Evolutionary Computation* (Andries, 2007)[7].

For this paper, all parameters of weights and biases among different neurons in MLP will be represented as Genes respectively. These genes are collected into a string which is called Chromosome. The target of GA in this paper is to find the individual who carries best Chromosome (i.e. make resulted model have highest accuracy or lowest error)

For initial population, Genes (weights and biases) will be generated randomly ranging from 0 to 1 and combined to be the Chromosome for an individual. Repeat producing until the population size reaches a certain number. To obtain a close-optimal population size Z, tests will be executed for Z from 1 to 11, then select the suitable one which achieves a trade off between final accuracy and time cost as the larger the population is, the larger time cost will be and it might be unacceptable if Z is too high.

For fitness function, using Chromosome as the initial weights and bias to train our MLP. Choosing the Chromosome of highest accuracy of resulted model for the fitness function.

For selection, this paper will use Elitism by default as the target is for the individual with highest performance no matter which generation it belongs to.

For crossover, this paper will use two-point crossover with simple discrete recombination by default. The cross rate from 0.1 to 0.7 will be tested respectively and choose the rate with highest performance.

For mutation, since Gray Coding will be used to deal with continuous float value for initial parameters, simple binary mutation operator will be utilized. Mutate rate from 0.1 to 0.5 will be tested and chosen as above.

3 Results and Discussion

3.1 Code environment

- Python 3.7.1 [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
- IPython 7.2.0 -- An enhanced Interactive Python.
- Jupyter Notebook 5.7.4

3.2 Normal MLP.

Firstly, we run the preprocess of dataset. After observation, 15 out of 31 objections are labeled as deceptive, which means it is close enough to be balanced, so rebalancing methods such as Under-sampling need not to be utilized. 0-1 scale and z1-socre have been implemented to reduce unnecessary importance put on special values. As mentioned above, BDR method will be utilized later and it is aimed at removing outliers while training. Consequently, outlier removal will not be processed in this stage.

Secondly, the number N, M of neurons in first and second layer is decided. N={2, 3,..16} and M={1, 2,..15} was tested respectively and the results are shown in Figure 1. The reason why no higher number is tested is that hidden neurons number is always believed to be less than input neurons to avoid overfitting, especially when train data set is small. Learning rate is set 0.05 by default. Training will also stop if epoch reaches maximum (500). Average of 5 times of 10-Fold cross-validation will be used as the scores.

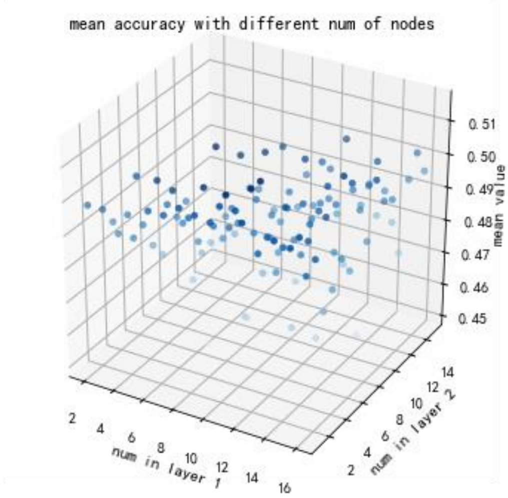

Fig 1. Performance of models with different number combination of neurons in hidden layers

As can be shown in Figure 1. No clear impact can be observed for different combinations of neurons number as the accuracy only varies slightly between 0.514167 and 0.451667. Such tiny difference can be caused by randomness of initial parameters or validation set split. Therefore, we choose N, M as (3, 2) since too complicate net structure can lead to much higher time cost of GA and overfitting for such a tiny dataset size.

Meanwhile, the maximum for times of epoch of training to stop is 201, which is far away from current upbound 500. So there is no need of adjustment for max times of epoch.

Clearly, the accuracy of simple MLP is quite low, which almost shows no difference between direct guessing from uniform distribution. Moreover, after observing the error curve of both train and validation set, it is found that curves for most models keep in similar value after small times of epoch, which show high possibility that back-propagation is stuck in local minimal. To overcome such situations, more tricks, such as optimizing initial parameters are required.

3.3 BDR method for noise clean

Firstly, we build a frequency curve to see whether a bimodal distribution can be found in Amin's training set[1]. The curve is shown in Figure 1 in Appendix.

Luckily, two peaks can be observed clearly in the distribution though the heights are similar. Since it can be regarded as a bimodal distribution, it satisfies the condition of Slade's paper. Further implementation can be done.

3.3.1. Assumption of BDR

Parameters will be used as the same as the default values in Slade's research. Every 50 epoch, if variance of error is lower than 0.1, BDR will be executed. In the first step, pattern with error larger than mean error will be taken out. For the subset, a new mean error value v_ss will be calculated. Patterns whose error is larger than v_ss+α * t_ss will be identified as outliers and removed from training set, where t_ss is the standard deviation of the subset. α is a parameter related to threshold of outlier identification and satisfies 0≤α ≤1. Furthermore, training will be stopped if the variance of error is less than 0.01 by default.

Table 1. Impact of BDR with different parameter α

parameter α	None	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
accuracy	0.51	0.568556	0.577072	0.575194	0.552544	0.56825	0.57825	0.573233	0.573233

In this paper, the code is implemented and shown in added file. The impact of BDR with different α are shown in Table 1. The scores are also the average of 5 times of 10-fold cross validations. In α = 0.6 tab, we can see that the accuracy is increased from 0.51 to around 0.578. Meanwhile, the accuracy after applying BDR is always higher than before. Therefore, the positive impact of BDR can be admitted in this paper. On the other hand, there is no clear trend of accuracy with different α . Consequently, we choose α = 0.6 by default for following discussion, since as the parameter highest accuracy, it has more possibility to be the optimal choice.

3.3.2. Shortcoming of BDR for small size dataset

However, while observing the error curve after, it is found that some models halted training immediately after 4 times of epoch (i.e. the fixed minimum of times by default) while training error is still decreasing with a significant speed. A typical example is shown in Figure 2.

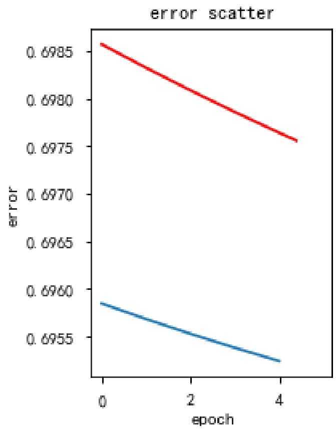

Fig 2. Example of error curve of a model after implementing BDR.

A reasonable explain is the negative impact from halting criterion of BDR. As mentioned above, training will be stopped if the variance of error is less than 0.01, even if model is still learning quickly. The original intention is to avoid the case when BDR is executed for many times, some objections with valuable information have been removed and continuing training will only lead to overfitting. An optional solution is to set halting criterion less strictly (e.g. reduce it from 0.01 to 0.001). However, such action could make more objections identified as outliers and removed, and for such a tiny dataset (31 objections are usually thought to be small for a net), removing actions have to be considered more carefully as a single objection can contain necessary information which can significantly impact the final performance. Compared to find a trade off value which is thought to be hard, increasing variance by adjusting initial weights and biases to decrease the possibility of triggering halting criterion in early time is more feasible.

3.4 GA method for initial weights and biases

Firstly, suitable population size NIND is decided. Table 2 shows related information for NIND varying from 2 to 11.

Table 2. Impact of GA with different population size NIND

NIND	Generation	best Acc in GA	mean of Acc	max of Acc	time cost
11	38	0.9	0.737083333	0.866666667	1217.02358
8	19	0.85	0.618333333	0.75	1648.19834
7	47	0.841666667	0.665416667	0.775	1771.8164
10	21	0.841666667	0.69875	0.783333333	1450.19911
9	4	0.783333333	0.6575	0.75	1734.19121
4	49	0.775	0.605416667	0.708333333	397.217552
6	17	0.766666667	0.570833333	0.675	411.643006
5	17	0.75	0.572916667	0.683333333	805.012443
2	12	0.5	0.485833333	0.491666667	24.6185797
3	20	0.5	0.4875	0.5	66.4788631

Column 'best Acc in GA' is the accuracy of the Chromosome (initial weight) selected by GA, and Table 2 is sorted by this column in descending order. Column 'mean of Acc' and 'max of Acc' is the mean and maximum accuracy while applying such Chromosome manually after GA, and their value is the average of 20 times of 10-fold cross validation. Column 'time cost' is the time cost for GA with corresponding NIND. And Column 'Generation' shows which generation produces the individual with selected Chromosome in GA.

With the increase of NIND, clearly, although the value in 'best Acc in GA' is usually higher than that in 'max of Acc', which could be caused by the randomness of validation split, the mean accuracy after implementing GA is significantly increased. It can be inferred that if we continue to increase population size, higher accuracy can be achieved by the trend. However, an explosion of time cost will happen even the required times of generation is less than 50. Therefore, this paper will not keep testing and compromise to NIND=11, but trying higher NIND is encouraged if physical conditions permit in the future.

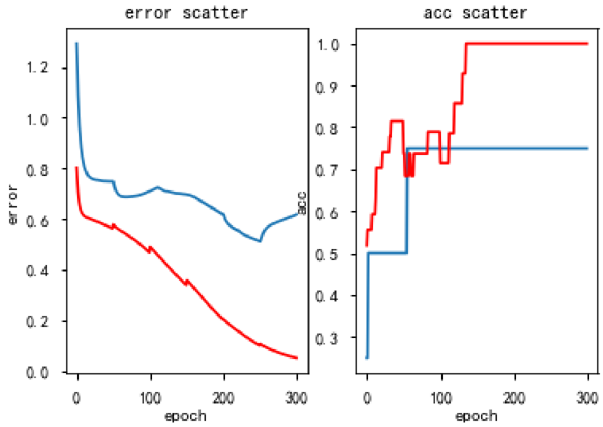

Fig 3. Example of error and accuracy curve of a model after implementing GA.

A reasonable worrying is whether GA just finds initial parameters which have been highly close to optimized parameters, just like using brute search for best parameters. If that happens, there is no need for network to train since initial parameters are indeed nearly the same as final parameters, and MLP will lose much value since training will never be needed.

To tweak the worrying, we collect the times of epoch after implementing GA's initial parameters. If the worrying is true, models should stop training very early as no clear change can be made in back-propagation, or keep training but the error is no longer reduced. Luckily, most models show they still keep training for more than 100 times of epoch, and error curve shows that training still reduce error effectively.

A typical example of that above is shown in Figure 3. In Figure 3, blue curves represent the values from validation set while red curves represent that from training set. Obviously, compared to previous model in Figure 2, applying GA dramatically increases the performance and remedies the disadvantage of BDR method.

Secondly, we test for obtain best crossover rate. Result is shown in Table 3 with rate from 0.1 to 0.7. It can be seen that accuracy is highest when crossover rate is 0.5, so it is used for further discussion. Note that due to physical limit of experiment tools, only discrete recombination is implemented in this paper. Linear recombination with floating-point representation is encouraged to test in the future.

Thirdly, we test for obtain best mutation rate. Result is shown in Table 4 with rate from 0.1 to 0.5. It can be seen that accuracy is highest when mutation rate is 0.3. After applying the hypothesis, we obtain the final GA for this MLP.

Table 3. GA with different crossover rate.

Crossover rate	0.7	0.5	0.3
accuracy	0.741666	0.825	0.775

Table 4. GA with different mutation rate.

Mutation rate	0.5	0.3	0.1
accuracy	0.741666	0.8	0.783333

Figure 4 provides how GA optimizes the initial parameters. Clear increase of accuracy can be seen with the increase of generations at the beginning. After reaching 18th generations, accuracy stays in a comparatively stable value.

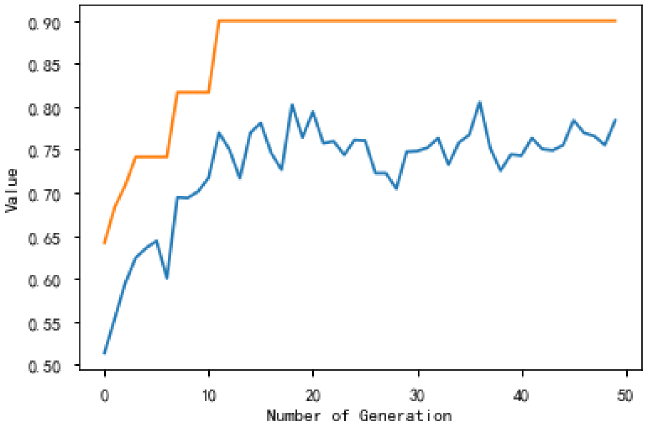

Fig 4. Accuracy curve in GA for different generations.

Orange curve in Figure 4 stands for the caught highest accuracy in history while blue curve stands for the mean of accuracy for current population in the generation.

In conclusion, GA is believed to be an effective tool for optimizing initial weights and biases and remedy the shortage of BDR method. A weak point of GA is that high time cost is required if population size is large. The efficiency of the combination of BDR and GA needs to be judged by comparing with other advanced optimizing methods for MLP in the future.

4 Conclusion and Future Work

4.1 Conclusion and improvement

Generally, BDR method provides stable and positive effect on increasing accuracy of normal neural network deception classifier based on features from facial thermal imaging. But for train set with small size or less actual outliers inside, the identified outliers might be random and inaccurate. No clear principles of parameter α in formula (1) can be found in that case. However, generally, implementing BDR will increase the accuracy firmly. Meanwhile, a 2-hidden layer MLP classifier with BDR is still not clearly lower than that of Amin's team [1] and the early halting problem is worth alert.

Halting problem can be effectively fixed by GA for initial parameters. Although spent time is comparatively high for this algorithm, the performance of model benefits dramatically from it. The mean of improved accuracy is very close to that in Amin's team, while the maximum accuracy has already exceeded it.

More complicated structure is recommended to build in the future. Related improvement concludes adding more hidden layers and adjust for more neurons inside. Moreover, applying advanced model such as 1-deminsion CNN is also an alternative choice. Meanwhile, increasing population size of GA and traditional tricks such as dropout is also encouraged if possible since the result has been proven to be expected higher if implement properly.

In addition, GA with float-point representation with multi Chromosomes, multi parents and other crossover methods are recommended to implement for close-optimal result in the future. On the other hand, rather than optimize whole initial weight set, focus on one weight or weights of certain layer is also an alternative way to improve performance of GA.

Furthermore, sampling from more objects to increase size of train set is required if possible. Since available data set now is not enough, it is hard to make full use of some optimal method which is related distribution and prior.

Lastly, increasing the times of test to obtain higher accurate scores. This paper utilizes the average of 5-10 times of cross validation. More times of test can be done in high speed machines and systems.

4.2 Value in future work

The major achievement in this paper is to find the potential problem of BDR method and confirm the feasibility of the combination of GA and BDR method. Early halting problem in BDR can happen if initial weights and biases are set unproperly, which make error variance so small that let BDR stop the model's training too early. The optimized initial parameters by GA can avoid such problem. Since there is possibility of the problem in other models, combining BDR with GA provides a practical direction for optimization.

Another discovery is to prove the priority of combination of BDR method with GA. For traditional noise removal methods, it might have negative or limited impact if given the size of train set is too small. But Table 2 shows that our method still dramatically optimize the performance in the tough case. Therefore, it is recommended to implement BDR with GA in most network build if possible to get benefit.

5 References

1. Derakhshan, A., Mikaeili, M., Nasrabadi, A. M., & Gedeon, T. (2019). Network physiology of 'fight or flight' response in facial superficial blood vessels. *Physiological Measurement*, *40*(1), 014002. doi: 10.1088/1361-6579/aaf089
2. Thomas, M., (2019, April 19). Neural Networks: Advantages and Applications. Retrieved from https://www.marktechpost.com/2019/04/18/introduction-to-neural-networks-advantages-and-applications/
3. Standard Score (z-score). (2013). doi: 10.4135/9781473979161
4. Slade, P., & Gedeon, T. D. (1993). Bimodal distribution removal. New Trends in Neural Computation Lecture Notes in Computer Science, 249–254. doi: 10.1007/3-540-56798-4_155
5. Ligade, P. (2019, September 3). Why cautiously initializing neural nets matters? Retrieved from https://towardsdatascience.com/what-is-weight-initialization-in-neural-nets-and-why-it-matters-ec45398f99fa
6. Mallawaarachchi, V. (2020, March 1). Introduction to Genetic Algorithms - Including Example Code. Retrieved from https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3
7. Engelbrecht, A. P. (2007). Computational intelligence: an introduction. Chichester, England: John Wiley & Sons.

6 Appendix:

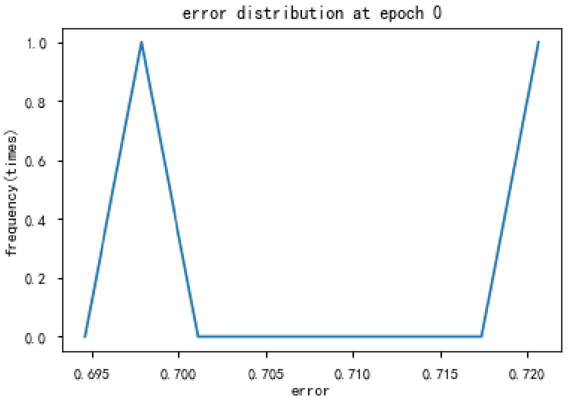

Fig 3. Example of error distribution at epoch 0.