Effectiveness Analysis of Neural Networks Using "Casper" Method in Judging the Authenticity of Angry Expressions Based on Changes of Pupils

Yuliang Ma

Research School of Computer Science,
Australian National University,
u6462980@anu.edu.au

Abstract. Progressive RPROP cascade correlation ("Casper") is an improved version of the cascade correlation ("Cascor") method, which uses a progressive method to add neurons to the network for training with varying learning rates. This paper explores the generalization ability of the "Casper" method by solving a problem of judging the authenticity of angry expressions based on changes of pupils with both the "Casper" method and a simple 2-layer fully connected neural network. The results show that the performance of "Casper" in this scenario is slightly inferior to the two-layer model, but accuracy of both models is acceptable. This paper speculates that one possible reason is that the "Casper" performs better when the data is extremely patterned such as the "Two Spirals". Also, using other methods or a combination of other methods to solve the issue in this scenario may improve the accuracy.

Keywords: Casper; Neural network; Effectiveness analysis

1 Introduction

Both "Casper" and "Cascor" are neural network construction methods that gradually add neurons to the network while training, which means that the structure and scale of the network will gradually become more complex and larger. These methods contrast with the traditional method of directly designing the network structure before training. The intention of the approach is to make the neurons that are added earlier to the network learn as many data patterns as possible, so that the neurons that are added later can focus on reducing the loss that the previous neurons cannot further reduce in the previous training. Therefore, the final model structure will match the complexity of the data (Kwok & Yeung, 1997).

The difference between "Casper" and "Cascor" is mainly reflected in how to add neurons to the network. "Cascor" "freezes" the weights of all the neurons added

before, and then the training process will be divided into two stages: first training the weight of this newly added neuron to maximize the correlation between its output and the remaining loss. Then training the weights of all neurons connected to the output units to minimize the overall loss (Fahlman & Lebiere, 1990). Although "Cascor" has been proven to perform well (Fahlman & Lebiere, 1990), it may result in too many redundant units being added and therefore causes the model being too large because it always has some "frozen" weights. (Kwok & Yeung, 1993). And its performance on some regression and classification problems is sometimes not satisfactory (Hwang, You, Lay & Jou, 1996). Differently, "Casper" uses a gradient descent algorithm called RPROP as the optimizer (Treadgold & Gedeon, 2006). RPROP allows different units in the neural network to use different learning rates (Riedmiller & Braun, 2002). The basic idea is that when a new neuron is about to be added, the weight associated with it will be given the maximum learning rate. The earlier the neuron is added, the lower the learning rate it is given. Usually, the entire neural network will be divided into three regions for 3 different learning rates. During training, RPROP will calculate all weights in a manner like back propagation. Since there is no "frozen" unit in "Casper", it is more flexible. Therefore, its performance in some tests, such as "Two Spirals Benchmark", is better than "Cascor" (Treadgold & Gedeon, 2006).

It cannot be ignored that, compared with other neural network algorithms, "Casper" is not widely used at present. There are not many papers on the effect evaluation of the algorithm. Therefore, this article intends to test the generalization ability of "Casper" in a specific situation.

The data set used in this model comes from a short paper from Australian National University (Chen, Gedeon, Hossain & Caldwell, 2017). It is pointed out in this article that changes in human pupils are more suitable to be used as indicators than changes in voices for judging whether people are angry or pretending to be angry. The prediction accuracy rate of the model constructed by the computer based on pupil data reached 95%, while the accuracy rate of human judgment for the same data set was only 60% (Chen, Gedeon, Hossain & Caldwell, 2017). The short article does not give details of the model it used. This paper chooses the same data set because the data is not extremely patterned like the "Two Spirals" (Fahlman & Lebiere, 1990). It is difficult for a simple 2-layer linear neural network models to use the "Two Spirals", which makes it not practical to compare with "Casper" and illustrate the generalization ability of "Casper". Using "anger" dataset is easier to compare between the two models.

2 Method

2.1 Data Pre-processing

The "Anger" data set describes the pupil changes of 22 students watching 20 different videos. Each video has a label to indicate whether the angry expression in the video is real or fake. Each video has 20 frames. This data set records the indicators of pupil change of all volunteers watching each frame. Therefore, the data set has 400 rows

plus a row of attribute names which is a total of 401 rows. The names of the attributes are the serial number of the video and the pupil's "Mean", "Std", "Diff1", "Diff2", "PCAd1", "PCAd2" and the label of that video. Therefore, the data set has the above 8 attributes plus a column index which is a total of 9 columns.

Table 1. The first ten rows of the original "anger" data set.

	Video	Mean	Std	Diff1	Diff2	PCAd1	PCAd2	Label
O1	T1	0.8431838	0.18523455	0.01220263	0.11444589	0.01739607	0.10219691	Genuine
O2	T1	0.8592473	0.10634656	0.003125	0.21364198	0.01705165	0.10749746	Genuine
O3	T1	0.84861888	0.12974385	0.00668682	0.05941109	0.01784606	0.10847376	Genuine
O4	T1	0.86716408	0.0970407	0.00201824	0.26251194	0.01739305	0.10733325	Genuine
O5	T1	0.94206699	0.03381895	0.00162462	0.29762421	0.0191801	0.10784395	Genuine
O6	T1	0.95101243	0.0268145	0.00234299	0.24802852	0.0196541	0.11008328	Genuine
O7	T1	0.92783216	0.08106683	0.00830452	0.05821458	0.0237639	0.11740363	Genuine
O8	T1	0.89223757	0.07270113	0.00366045	0.2463566	0.01872797	0.11715638	Genuine
O9	T1	0.85548774	0.0647545	0.00690465	0.18894512	0.02074209	0.11527459	Genuine
O10	T1	0.93101366	0.03563209	0.00322212	0.29963703	0.01977804	0.11563746	Genuine

Based on the different issues discussed, this article uses this data set in two ways. First, this paper discusses the performance of "Casper" algorithm compared with ordinary two-layer neural network. In this question, this article treats the pupil data of each frame as a data point and ignores which video the frame comes from. During training, a single frame of data is input to predict the whether the expression in the picture is genuine. The advantage of this is that a relatively large amount of data can be obtained for training and testing (total amount of 400). In addition, it is mentioned in the source paper of this data set that the prediction accuracy of the model built by the team reached 95% (Chen, Gedeon, Hossain & Caldwell, 2017). But this accuracy is based on treating the entire video as a data point rather than a single frame. In the following content, this paper will also study the performance of "Casper" and simple two-layer networks in the same task regarding the whole video as input. In order to do that, this article reshapes the data of 20 frames from a same video into a one-dimensional vector as a data point. This means that the entire data set contains only 20 data. This has brought a significant problem, that is, the size of the data set is too small, which brings difficulties to training and testing.

In the process of data pre-processing, redundant data such as index columns, column name rows, and video tags are first deleted. Then this article uses label encoder to encode the label as "0" or "1" and implements normalization by columns.

Unlike the traditional method of using the "Sklearn" library to randomly divide the data, this article introduces a special method for training, verification and test set division. The outcome is that in the three sets, the proportions of true and false labels are each half. Also, in the training set, these two labels will appear alternately, thus constructing a "balanced" data set. This method has been proved that it can improve accuracy because the model will not be biased due to the uneven distribution of the data set.

2.2 Models Description

In this paper, two models are used, namely the progressive neural network constructed by “Casper” and the simple two-layer network. “Casper” is based on a published academic paper (Treadgold & Gedeon, 2006), and the two-layer network simply uses the model provided in the “Pytroch” library as the basis.

In the “Casper” algorithm, the network is initialized as a simple two-layer structure without hidden neurons. After the training starts, and a certain number of epochs, the training will come to a “checkpoint”. At this point, if the overall loss cannot continue to decrease enough compared to the situation at the last “checkpoint”, one new neuron is added according to the rules and the optimizer is updated. The hyperparameters in the model are the upper limit of the number of hidden neurons “k” and the upper limit of the number of epochs, “n_epochs”, and the “p” used to determine the interval between “checkpoints”, which is generally set to 5. The testing result shows that the “p” value does not have a significant impact on the result. The value of “k” will be determined through validation.

In this paper, in order to compare the performance of the structures of the two models, rather than to determine which model is more competent for the task in this scenario, the parameters of the simple two-layer neural network model are set as consistent as possible with the “Casper” algorithm, and the two models share the same data set that has undergone the same processing. The number of hidden neurons in the double-layer model is also set to “k”. In addition, since the “Casper” algorithm often stops because it first reaches the upper limit of the number of neurons set instead of reaching the maximum number of epochs, the number of epochs for a simple two-layer network is set to be relatively smaller.

Finally, in the original “Casper” paper, the activation function is the “Sigmoid” (Treadgold & Gedeon, 2006). However, this paper finds that the signal transmission is often not obvious during training, which leads to the slow convergence speed of the loss function. This article tested several other activation functions, and finally decided to use the “LeakyRelu” function. Similarly, adding a 30% dropping rate in front of hidden neurons has also been found to be helpful for improving accuracy and convergence speed. Both settings are also applied to the two-layer network structure. In addition, this article also adds a momentum of 0.1 to the optimizer of “Casper” to prevent training from being trapped in some local minimums.

2.3 Hyperparameter Setting

The hyperparameters involved in this article mainly include the division ratio of training, validation and test data sets, the upper limit of the number of hidden neurons “k”, and the upper limit of the number of epochs “n_epochs”.

As mentioned above, this article uses two data processing methods to solve two problems. For comparing the performance of “Casper” and ordinary neural networks, this article uses a processing method that can result in 400 data. The training set accounts for 60%, and the validation set, the test set each account for 20%. For exploring the ability of “Casper” and ordinary neural networks of solving problems in

this specific situation, this article uses a processing method that can result in 20 data. The training set accounts for 80%, and the validation set, the test set each account for 10%.

For the first question, this article uses validation process to find the best “k” value for “Casper”. The method is: first set the “k” value and “n_epochs” at a larger level and use a dedicated data set for verification every time when the algorithm is about to add a new neuron. Then record the training loss and verification loss. After the algorithm terminates, plot a figure of the changes in training loss and verification loss as the number of neurons increases.

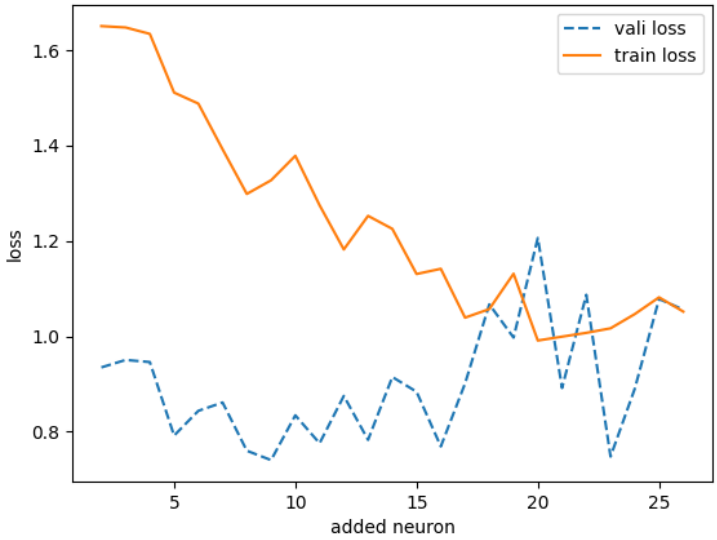

Fig. 1. During the training process of the "Casper" method, the training loss and verification loss change as the number of neurons added to the network increases.

The above figure is drawn when the value of “k” is 25 and the value of “n_epoch” is 5000. After many experiments, this article found that the validation loss always fluctuates sharply and rises after more than 15 hidden neurons have been added in the network, but it had been fluctuating in a small range before then. Experiments have also proved that when “k” is set to 15, the model can achieve the best accuracy.

When the “k” value is set to 15, the training of “Casper” always stops after reaching the upper limit of the “k” value after about 2000 epochs. Therefore, this article sets the “n_epoch” value of “Casper” to 2500, and the “n_epoch” value of the two-layer network model is set to a slightly smaller value of 1700.

Under this setting, we can see that the training loss of “Casper” and the two-layer network model has shown a reasonable decline.

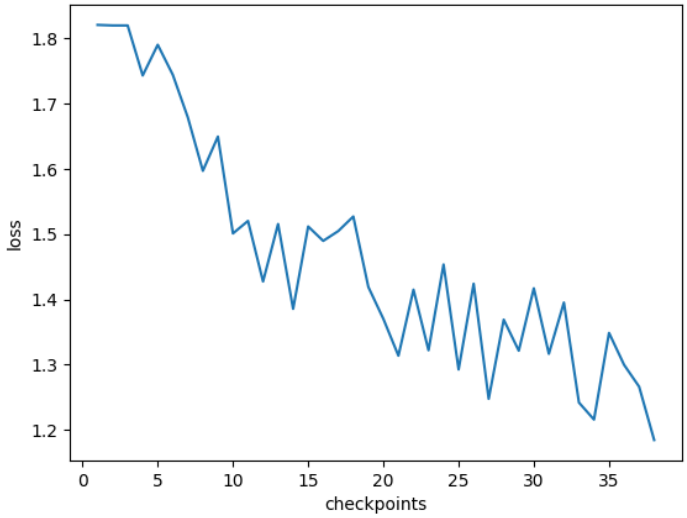

Fig. 2. During the training process of the "Casper" method, the training loss changes at every checkpoint.

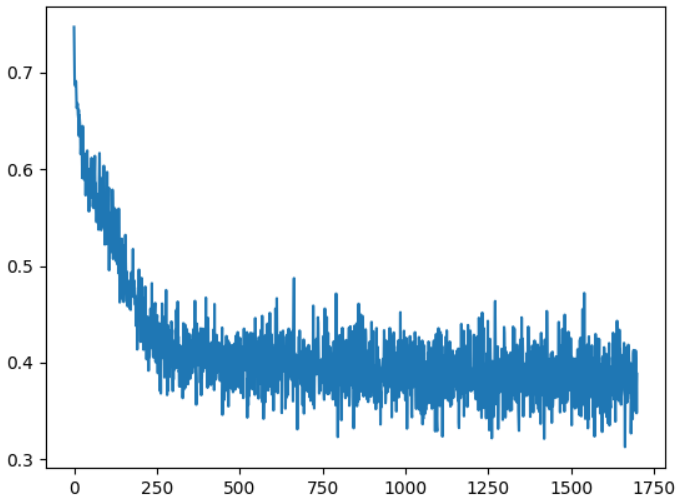

Fig. 3. In the training process of the simple two-layer neural network model, the training loss changes in every epoch.

Similarly, this article has made similar settings for the two models for the second approach. However, due to the increase in data features, the “k” value is set to 20.

Due to the reduction in the amount of data, the value of “n_epochs” is 1500 for “Casper” and 100 for double-layer networks.

3 Results and Discussion

This article uses an independent test set to test the model. The testing indicators include confusion matrix, test loss and the accuracy. The performance of all models on the confusion matrix and test loss is normal, which shows that there is no problem with the construction and operation of the model. Therefore, the analysis of the results in this section is mainly for accuracy.

For the first data processing method, each frame is regarded as a data point, and the label is determined based on every separate frame image. Under this setting, use the same training set and test set for the “Casper” and two-layer neural network and let them run for 10 times (note that every time the data set will change), and then record the mean accuracy of the two models. The accuracy of “Casper” is 68.9%, and the two-layer network model is 74.8%. The accuracy of “Casper” is slightly lower.

This accuracy is based on a single frame image. In actual tasks, images from the same video must have the same label. But the two models designed in this problem did not learn this information. Therefore, their accuracy of about 70% is acceptable.

This result also shows that if a data set is not difficult to process with conventional neural network models, it may not be a good idea to blindly introduce “Casper” into the neural network. The “Casper” algorithm is more complex, requires more computing power, and runs slower. It is true that “Casper” has the advantage that it is not easy to “forget” the data patterns that have been seen, but this also causes it to be more inclined to introduce more biases and thus reduce the accuracy. Perhaps only when the data set contains patterns that are difficult to handle with conventional models such as the “Two Spirals” (Fahlman & Lebiere, 1990), “Casper” can show its superiority.

For the second data processing method, each video containing 20 frames is regarded as a data point, and the label is determined based on all the frames in the video. But under this setting, both models performed very badly. According to a similar method to the above, the average accuracy of “Casper” measured is only 35%, while the accuracy of the two-layer network is 40%.

The reason for this phenomenon is also obvious, that is, the data set is too small. For each video, in this method, this paper reshapes a 20-row matrix from the original data set into a one-row vector, which reduces the data set by 20 times, resulting in only 20 data left in the entire data set. Only 10 of them are used for training, the remaining 5 are used for verification, and the other 5 are used for testing. The excessive number of interneurons and epochs leads to overfitting of the model. This led to the inevitable failure of this method.

The original intention of this processing method is to make the model learn the relationship between different frames within a video, so as to improve accuracy, because it learns additional information that other methods cannot learn. But from the results, the over-fitting phenomenon caused by the reduction of data completely

overwhelms the improvement brought by this setting. Maybe a larger data set is more suitable for applying this method.

The original article which constructed this data set claimed that their model accuracy reached 95% (Treadgold & Gedeon, 2006), but they did not give detailed information about the model used. However, it is undeniable that, as mentioned above, when the frame-by-frame prediction accuracy rate reaches 70%, there are indeed other ways to further improve the accuracy with a combination of frames, which will be explained in the future work section.

4 Conclusion and Future Work

This article first introduces the principle of the "Casper" algorithm and implements a neural network model accordingly. The article tested the practicability of "Casper" by using a data set named "anger", and the results showed that in this specific situation, the accuracy of the model built by "Casper" was slightly lower than that of a simple two-layer fully connected neural network. It is deduced that "Casper" may be more suitable for solving problems in highly patterned data sets that are difficult to capture by traditional linear neural networks, such as the "Two Spirals". At the same time, the article tries to solve the problem of "judge the truth of anger in the video based on pupil changes", and the results show that the approach selected in this article is not suitable for the data set. However, based on the research in this article, the accuracy of single-frame prediction has reached about 70%. This result can be applied in subsequent work, that is, to solve "how to use all the frames in a video to predict the label of the video". A brand-new approach is to input all 20 frames contained in the video into a simple double-layer network model to obtain 20 labels. Then let these 20 frames vote to determine the final label. This is very likely to get an accuracy much higher than 74.8%, which is the mean testing accuracy given by the 2-layer neuron network model that employs the frame-by-frame approach of prediction in this article.

References

1. Chen, L., Gedeon, T., Hossain, M., & Caldwell, S. (2017). Are you really angry?. Proceedings Of The 29Th Australian Conference On Computer-Human Interaction. doi: 10.1145/3152771.3156147
2. Fahlman, S., & Lebiere, C. (1990). The Cascade-Correlation Learning Architecture. Advances In Neural Information Processing, II(1990), 524-532.
3. Hwang, J., You, S., Lay, S., & Jou, I. (1996). The cascade-correlation learning: a projection pursuit learning perspective. IEEE Transactions On Neural Networks, 7(2), 278-289. doi: 10.1109/72.485631
4. Kwok, T., & Yeung, D. (1997). Constructive algorithms for structure learning in feedforward neural networks for regression problems. IEEE Transactions On Neural Networks, 8(3), 630-645. doi: 10.1109/72.572102
5. Riedmiller, M., & Braun, H. (2002). A direct adaptive method for faster backpropagation learning: the RPROP algorithm. IEEE International Conference On Neural Networks. doi: 10.1109/icnn.1993.298623

6. Treadgold, N., & Gedeon, T. (2006). A Cascade network algorithm employing Progressive RPROP. Lecture Notes In Computer Science, (1240:733-742). doi: 10.1007/BFb0032532

Appendix:

“Approach 1 Casper & SimpleNN.py” is for the first data processing method.
“Approach 2 Casper & SimpleNN.py” is for the second data processing method.
“Anger.xlsx” is the original dataset.