Angry or not? A threshold-varying neural network approach for binary classification

Y. Gao

Research School of Computer Science
Australian Natioinal University
u7016951@anu.edu.au

Abstract. The usage of neural network could realize the automation of facial expression detection and has achieved high accuracy than human classification. In this paper, training models are built to tell whether the anger of participans is genuine or posed based on the dataset provided with simple statistics feature of recorded anger video. A 85% accuracy is acheived by TVNN as the baseline of our research. Further, a Convolution Neural Network (CNN) is invovled to extract features from raw datapoints of pupil but not the statistic ones. Compared to the baseline, *Convolutional TVNN*(C-TVNN) got up to 10% improvement for the accuracy. Numerical results show that with selected threshold C-TVNN can acheive a final accuracy very close to Pupilllary method. The optimal threshold of decision boudary is selected according to the result of experiments with a range of threshold changing.

Keywords: Binary Classification· Anger Recognition · Convolutional Neural Network· Threshold-Varying

1 Introduction

Machine learning was widely used in the field of image processing and pattern recognition. With the development of neural network, since its flexibility and high accuracy, it is further used in the research of facial recognition like emotion detection. Some early works like [1] [2] adopt machine learning into emotion recoginition. More sigificant literatures found that deep neural networks can extract high-level features and further improves performance[3][4].

In this paper, our dataset is a collection of 400 records of pupil size with corresponding tags. A simple 4-layer linear network is firstly trained with 6 statistic features of original record data. To distinguish between real anger and posed one, output layer has only one node with a threshold as a decision boundary to tell the genuine anger expression. Based on this baseline, a deep convoluntional network is involved to extract more detailed features directly from original data records. As an optimization of model, different threholds are selected and simulized. With the changing threshold from 0.2 to 0.9, a significant improvement of accuracy in the test set has been monitored. Multiple loss function is also discussed in this paper.

2 Main Method

2.1 Model Structure

In this paper, the anger data is from literature[5] and data is from 20 videos which are seperately in 20 different phases. So the total number of dataset contains 400 pairs of data, each of which contains over 100 time-sequenced records of the pupil size from both left and right eyes. In this paper, as a baseline of research we first introduce TVNN method. The six statistic features are calculated from original datasets. Then features are feeded into input layer of linear model. And then we also introduce a C-TVNN method using a deep convolutional neural network as another method for feature extraction. The ouput will also be feed to linear layer. Both models produce output as one-node score used to decide whether the anger is realy or not.

2.1.1 TVNN

As a concept for discussing the availability of neural network for anger detection based on the anger dataset, a simple neural network with three layers is built. The video aspect is ignored so we only calculate 6 statistic features of the raw datasets, which contain “Mean”,” Std”,” Diff1”,” Diff2”, “PCAd1”, “PCAd2” columns. So, the input layer contains 6 nodes. The hidden layers are 6*14 and 14*14 linear layers. The second layer is set to improve the model performance. The output layer is a 14*1 linear layer. Between the first and second hidden layer, and between second hidden layer and output layer, relu function is used to prune the network. After the output linear layer, a sigmoid function is added to rescale the output to (0,1), in which the threshold will be discussed later.

2.1.2 C-TVNN

In the experiments of TVNN we found that the accuracy performance is not so good as some previous works like [5]. As analysis it is becuase the 6 statisc-faced inputs can not present the detailed features of the original data. To avoid a loss of information, we involve a deep convolutional neural network to extract the features. The adopted convolutional neural network includes 3 convolutional layers. Input is the padded datas from left and right eye of the same person, which can be organized as a 2-chanel 186 vector.The first layer is 1-d convolution from 2 chanel to 8 channel with stride size 2. Then the second layer is 8 chanel to 24 channel 1-d convolution with stride size 2. And The third layer is also 1-d covolutional layer from 24 to 72 channel with stride size 2. To improve the convergence speed and reduce the impact of data sparsity, the datas are activated by a relu function betwwen layers. The raw data from eyes are redundant so maxpool is also used to down-sample input representation and also prevent overfitting. The number of maxpool layers need to be optimized and chosen or there will be loss of informations, which will be discussed in section 3.

The output coming out from the convolutional layers is a batch of 72 channels feature data. It will be flatterned and feeded to the fully connected layer. Then the data will be processed through 2 linear layers and get the final output. The number of nodes in final output will be 1 or 2, depending on the method adpoted to make the final decision.

2.2 Data Pre-processing

Dataset in this paper is 400 sets of time-series datas from both the right and left eye. We need to divide them to the training set and test set to see the training performance and prevent overfitting. In the provided dataset we also note that the range of 6 statistic features are quite far from each other. Even for the raw data, the range and length of each dataset are quite differernt from each other. As a result, the raw data should be normalized and padded to get a uniformed orgnization.

2.2.1 Training and Test Segment

Since the dataset has 400 pairs of data and labels, it is randomly seperated as training and test set, and the ratios are 70% and 30%. We didn't involve a validation set in the training like liturate[5] as a sign of stop since it's hard to set the value because as discussed in later this paper, the result doesn't converge to a expected level until a large number of epoch. During the training, data in testset is not involved.

2.2.2 Normalization

The original range of all 6 feature is included in Table.1. Moreover, as is showned in Figure.1 the raw dataset from left and right eye are also in different ranges. This kind of imbalance in training dataset will cause numerical unstability during the training. And also the diversity in range will lead to a longer way for the model to get to its universal minimum loss. As a result, we normalize the datasets before feeding it to the training model. The feature values will be scaled to the same range [0,1] without distorting their relative position.

Table 1: Range of 6 features for TVNN

	Mean	Std	Diff1	Diff2	PCAd1	PCAd2
Max	0.9834	0.3584	0.044	0.4182	0.0838	0.2393
Min	0.5829	0.0080	0.0011	0.0219	0.0103	0.0611

The normalization function is define as below:

$$N(v_k) = \frac{v_k - min(v)}{max(v) - min(v)} \tag{1}$$

where v is the group of original values of a feature, v_k is the k-th value in this group, $min(v)$ and $max(v)$ are minimum and maximum values in this group, $N(v_k)$ is the normalized value.

2.2.3 Padding

The raw data of each person are indfferent length of time as they are records from different videos, which makes it difficult to feed to the model. In our reserach we pad the time-series dataset with 0 at the tail to the maximum length among differernt persons. In our dataset the maximum length is 186 so input data is a 2-chanel vector with length 186 points.

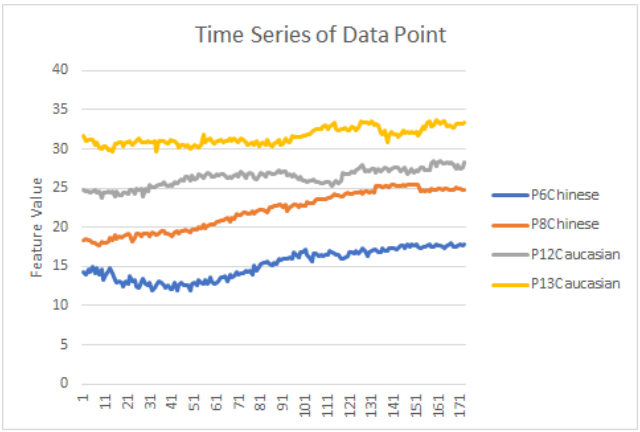

Fig. 1: Range of 4 sample dataset in C-TVNN

2.3 Threshold

The output tag is encoded as 1(genuine) or 0(posed). However, in our model, the result we get from the output layer is one-node vlue from a sigmod, which is located in the range (0,1). Therefore, a threhold is involved to to determine the final prediction between 0 or 1 based on a threhold. The rule to determined 1 or 0 follows below equation2:

$$label(x) = \begin{cases} 1, & x > threshold \\ 0, & otherwise \end{cases} \tag{2}$$

where x is the output of model, which is the output of the last linear layer add a sigmoid layer to rescale the output to (0,1). Note that this threshold is not fixed value. We can optimize the performance of our model by ajusting the threshold from 0 to 1. In Section 3 we will also discuss the choice of threhold and its impact on accuracy and convergence.

2.4 Loss Function

In this paper, we compare and make a choice between two kinds of outputs. One is a 1-node scoring ourput and judge it with threhold as de disicion boundry. And the other one is 2-node output presenting the probability of the result 'pose' and 'genuine'. So two kinds of loss function is used for calculating the loss and updating hyperparameters.One is cross entrophy and the other is binary cross entrophy. For classification task, cross entrophy is a widely used loss function. Since this anger-or-not problem only have 2 knid of oupt, the targets should be either 1 or 0. As a matter we can involve Binary Cross Entrophy Loss (BCELoss) in pytorch to meaure the error between tags and the predicted output. However, from another output, our neural network has two-node output, which can be treated as a scoring of probability for both 2 targets. Cross Entrophy Loss fucntion can also be used to choose from 2 tags (0 or 1).

Cross Entrophy Cross Entrophy Loss is popular in the calssification problems with multipule calsses, in which output are linked to a log-softmax function and a NLLoss in its encapsulation. And finally the losses will be averaged across all classes. The loss function can be decribed as:

$$L(x, class) = weight[class](-x[class] + log(\sum_j (exp(x[j])))) \tag{3}$$

where $class$ is the target label and x is its corresponding ouput from our model.

Binary Cross Entrophy Loss Cross entrophy[9] is a measurement of the difference between different probabilities distribution in information theory field and binary cross entrophy is specially used for "True or False" classification. Pytorch has a built-in module for calculation binary cross entrophy loss[11] with input from sigmoid function. The loss calculation is as below equation4:

$$l(x, y) = L = l_1, ..., {l_n}^T, l_n = -\omega_n[y_n.logx_n + (1 - y_n).log(1 - x_n)] \tag{4}$$

2.5 Test Process and Parameter

In each epoch, after training the model with training dataset, the test will also be performed with test dataset. Accuracy result of test will be recorded for each epoch along with the state dictionary of the model. When all epoche is done, the result with the largest test accuracy will be selected as the final training result. This training process will be conducted with different threshold on the dicision.After a few test with small number of epoch and batch size, we pick epoch =20000 and batch size = 15. After varying threshold to determin the labe as 1 or 0 from 0.2 to 0.8, the accuracy on testset will be recorded throughout the training process for every epoch. Loss function with MSE loss will also be compared with BCE loss.

2.5.1 Ground Truth

Before the result of the experiment is discussed, a ground truth needs to be set to compare the performance of the neural network. After checking the data in test set, we found that the ratio of "POSED" data is 0.5583 , so a random label assignment with all 1s or 0s would give a accuracy of 55.83%, so the output accuracy of the model must be higher than 55.83% to show the effectiveness.

Besides, as the experient result of literature[5], the accuracies of human classification and the model classification, which are 60% and 95%, still need to be compared in this paper.

2.5.2 Learning Rate

After running a small number of epoches and seeing the trend of loss decreasing, we set three level of learning rate. The basic level of learning rate is 0.01 and happens when the output accuracy is less than 67%. 0.01 is chosen as it is a relatively large learning rate and could speed up the gradient decreasing at the beginning when the loss is ralatively large. After the accuracy is larger than or equals 67%, the optimizer will decrease the learning rate by 90% to 0.001. Not until the accuracy is as large as 77%, the learning rate will change to 0.00001 and keep it until the tranining is finished.

3 Result and Discussion

3.1 Threshold adjusting experiment

In this section we compare the accuracy performance of TVNN and C-TVNN choosing differernt threshold as the decision boundry. Numerical results are shown in Figure.**??** From the test result we can see, with the threshold changing from 0.2 to 0.8, the output accuracy first increase then decrease, for both TVNN and C-TvNN. The peak happens in the rage 0.4 to 0.6. The maximum accuracy is 85% for TVNN at 0.4 and 94.8% C-TVNN at threshold 0.34. We adpoted a sigmoid function at the final output layer. With the largest gradient the middle range of it outputs can better seperate the input diferernce.

Table 2: Accuracy Comparation

	Verbal response	Pupillary	TVNN	C-TVNN
Accuracy	60%	95%	85%	94.8%

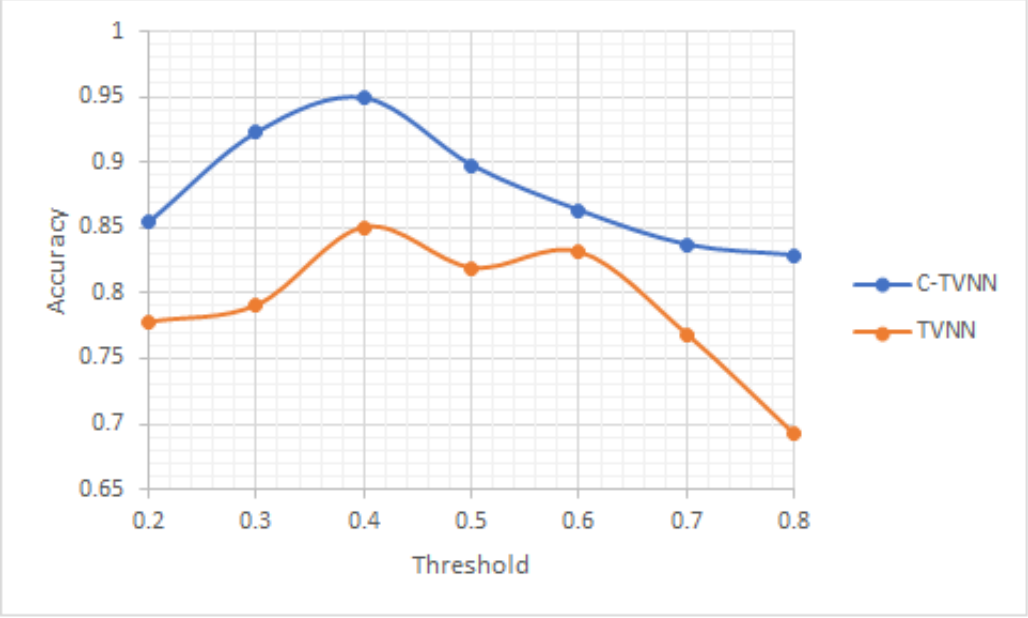

Fig. 2: Maximum Accuracy with Different Threshold

It is obvious that 6 statistic features in TVNN can not reflect every detail information contains in the raw data. So the inforamtion loss from the input portal limits the final performace of this method. But for C-TVNN the features are better extracted with the deep convolitional neural network. As a result we can find that C-TVNN can acheive a up to 10% accuracy improvement at every choice of threhold compared to TVNN.

The result is compared to the results for verbal response and pupillary response trained machine classifiers proposed in literature [5]. As we can see in below table the best performance of our model is 85%, which is better than the human verbal repsonse (60%) but it achieves an acuuracy lower than pupilary method(95%).

3.2 Convergence experiment

In this section we present the convergence of loss function for both 2 methods with the same decision threhold at 0.4. From convergence plot of TVNN, we can see the best performance of the model shows as early as around 10000 epoch, after which even though the loss on the trainset is still decreasing. The performance becomes worth and the model shows signs of overfitting. The plot from C-TVNN also shows a similar phenomenon.

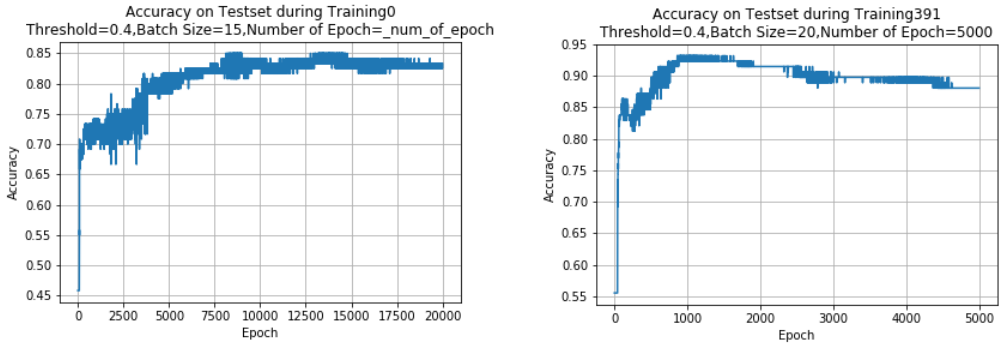

Fig. 3: Convergence comparision between TVNN(Left) and C-TVNN(Right)

The plot also shows that C-TNVV reaches it optimal accuracy after 1000 epoches which is much faster than TNVV (8000 epoches).

3.3 Feature Extraction Experiment

In C-TVNN method, there is redundency in input raw data maxpooling is added between convolitional layers.Since our dataset is not so redundant as pixels in picture, we find that maxpooling is not necessarily added at every slot between convolutional layers. To many max pool will cause serious information loss and impacts the accuracy. As is showned in table below we also compare the accuracy performace with 1, 2, and 3 maxpoolings in our model. So the best choice is to add it between the first and second convolutional layer. The redundency is eliminated and meanwhile accuracy is not impacted. Any further maxpooling will lose the necessary features for the training.

Table 3: Maxpooling Impacts

	No Maxpooling	1 Maxpolling	2 Maxpooling	3 Maxpooling
Accuracy	92.33%	92.0%	82.90%	70.08%

The information loss are also observed when wo adjust the size of our convolutional layers. Here we provide 2 options of Convolutional Layer. First one is 2-chanel to 4-chanel then 16-chanel and then 32-chanel (option 1) and another one is 2-chanel to 8-chanel then 24 chanel and then 72 chanel (option 2). Besides that, all other variables in network are same. The Relu is add between layers and maxpooling is added at the output of first layer. From below results, option 2 with larger size of convolutional layers extracts the features better and archive higher accuracy. Option 1 converge very fast but it is easy to overfit the training sets.

Table 4: Maxpooling Impacts

	Option 1	Option 2
Accuracy	88.9%	82.9%
Max Accuracy after	33 epoches	400 epoches

3.4 Output Score Experiment

In our work, after data is processed by convolutional layer and fully connected layer, it can also produce a 2-node ouput. Each node is the probability of the class 0 or 1, so this problem can be treated as a classification.

We also provide the numerical result of classification model as below. It shares the same convolutional layer with C-TVNN and only difference is the output nodes. We can see that the accuracy and convergence speed are almost the same.

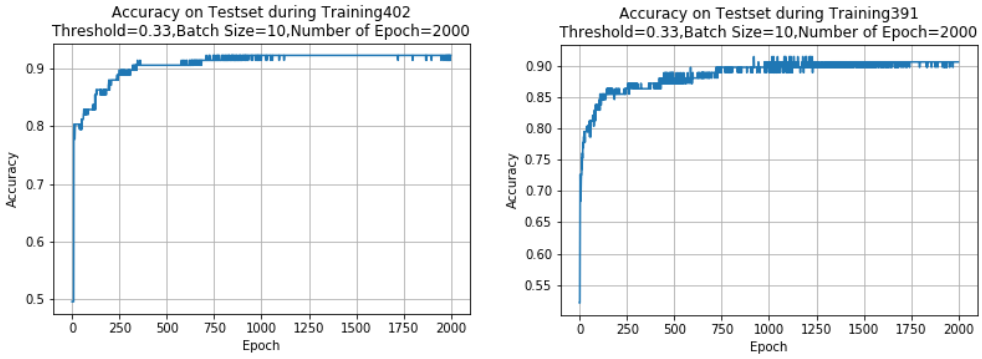

Fig. 4: Convergence comparision between C-TVNN(Left) and 2-Classification(Right)

4 Conclusion and Future Work

In summary, this paper provided 2 methods of Threshold-Varying Neural Network on anger dataset to distinguish the posed anger from genuine ones. A simple linear model TVNN is training based on 6 statistic features of the original datasets. The final ouput is a score ranged from 0 to 1 and be processed to 0 or 1 based on a fine-adjusted threhold. To further extract the detailed feature we introduce C-TVNN which involve 3 layer deep convolutional network. Numerical reuslts show that C-TVNN acheives an highest accuracy of 94.8%, which is 10% better than TVNN (85%) and quite close to existing methods(95%).

For threhold adjusting, the results curve shows the accuracy will first increase then decrease as the threhold goes from 0.2 to 0.8. The peak of both TVNN and C-TVNN appears around 0.4, which is the most effective part seperating different input features. The Convergence experiment also shows the convergence behalviour of both 2 methods. And C-TVNN is much faster on the way reaching the optimal accuracy. Experiments are also performed with feature extraction topics. Results show that insufficient size and over maxpooling can both cause information loss and further impact the accuracy. The last experiment is for comparation between C-TVNN and classification method. Results show that accuracy and convergence performance are quite close between C-TVNN and 2-classification.

In our research, we found that the inputs neurons in our model is generally accpeted and processed without any selection. Some of the input neurons are not making effective contributions to the final ouput and inteferering each other. So evolutionary algorithm is a good direction to research in next stage.

References

1. Oh-Wook Kwon, Kwokleung Chan, Jiucang Hao, Te-Won Lee (2003.). Emotion Recognition by Speech Signals. In EUROSPEECH. (pp.125-128).
2. C. Shan, S. Gong, and P. W. McOwan (2009.). Facial expression recognition based on local binary patterns:A comprehensive study. In Image and Vision Computing. (pp.803–816).
3. Kun Han, Dong Yu, Ivan Tashev (2014. September). Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine.In INTERSPEECH. (pp.14-18).
4. D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide (2013.). Feature learning in deep neural networks-studies on speech recognition tasks.In arXiv preprint arXiv. (1301.3605).
5. Chen, L., Gedeon, T., Hossain, M. Z., Caldwell, S. (2017, November). Are you really angry? detecting emotion veracity as a proposed tool for interaction. In Proceedings of the 29th Australian Conference on Computer-Human Interaction (pp. 412-416). ACM.
6. L.K. Milne1, T.D. Gedeon1 and A.K. Skidmore: Classifying dry sclerophyll forest from augmented satellite data : Comparing neural network, decision tree and maximum likelihood. In: Proc. 6th Australian Conference on Neural Networks, ACNN'95, pp.160-163. (1995)
7. Karen Simonyan, Andrew Zisserman:Very Deep Conbolutional Networkds ForOR Large-scale Image Recognition. In:ICLR (2015)
8. Karen Simonyan, Andrew Zisserman:Classifying Posed and Real Smiles from Observers' Peripheral Physiology. In:Health '17: Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for HealthcareMay 2017 Pages 460–463
9. Jason Brownlee:A Gentle Introduction to Cross-Entropy for Machine Learning https://machinelearningmastery.com/cross-entropy-for-machine-learning/ Last update on October 21, 2019
10. A Gentle Introduction to Threshold-Moving for Imbalanced Classificationhttps://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/. Last Updated on February 12, 2020
11. https://pytorch.org/docs/master/nn.html