The Comparison of Facial Expression Recognition Based on Resnet and BP Neural Network (Based on Pytorch)

Xincheng Xu[1]

[1] Research School of Computer Science, Australian National University, Australia
u6920920@anu.edu.au

Abstract. Facial expression recognition plays an important in the systems of expression analysis. In this paper, I firstly make facial expression recognition training based on the histogram of oriented gradients (HOG) and the pyramid of histogram of oriented gradients (PHOG)[1] through BP neural network, and then used the pruning neuron technology to reduce the redundancy of hidden layer neurons. Later, I use Resnet to identify the original facial expressions in the SFEW database[1].By comparing the accuracy of the two models, I find that the Resnet model performs better.After that, I will optimize the existing model and use other advanced models to make the recognition.

Keywords: BP Neural Network, Resnet, Facial Expression Recognition

1 Introduction

Facial expression is the most direct and effective mode of emotion recognition. Automatic facial image analysis has been a long standing research problem in computer vision[1].Facial expression analysis involves the measurement of facial movement and the recognition of facial expressions, which are generated by changes in a person's facial muscles that transmit the individual's effects to the observer[2]. It has many human-computer interaction applications, such as fatigue driving detection and real-time facial expression recognition on the mobile phone. As early as the 20th century, Ekman and other experts proposed seven basic expressions through cross-cultural research, namely anger, fear, disgust, happiness, sadness, surprise and neutral.

The first dataset contains the categories of facial expressions in SFEW database[1], the first 5 principal components of Local Phase Quantization (LPQ) features and the first 5 principal components of Pyramid of Histogram of Gradients (PHOG) features. PHOG operator can extract the gradient information of the image, so it effectively describes the details and structure information of the image. LPQ operator can effectively describe the feature information of the face image. If we combine them, we can get the detail information and feature information of the image. For pruning hidden units, we can guarantee a consistent level of functionality of units in the compression layer based on their distinctiveness, and can progressively reduce the size of the compression layer for the desired level of image quality[3,4].

The second dataset contains 675 original images in SFEW database[1]. Each image is a color image and the size of the image is 720 * 576. The ResNet model is used to extract discriminative features robust to the resolution change[6]. Hence, we use ResNet model to do the classification.

This paper mainly classifies seven facial expressions by using the five most important features of PHOG operator and LPQ operator in the first dataset and use the technique in [4] to prune the redundant units. At the same time, we also use Resnet model to extract main features from the second dataset and use these features to make the face emotion classification. Finally, we will compare the accuracy of these two models.

2 Method

2.1 Method for Implement Face Emotion Classification(PHOG & LPQ)

The basic model of face emotion classification for PHOG & LPQ is a BP neural network. The specific structure, pruning strategy and optimization method are as follows:

2.1.1 Structure of the BP Neural Network

The BP neural network consists of 1 input layer, 1 hidden layer and 1 output layer. All connections are from units in one level to the subsequent one, with no lateral, backward or multilayer connections. Each unit has a simple weighted

connection from each unit in the layer above. The activation function of the hidden layer adopts the sigmoid activation function. The activation function of the output layer is sigmoid activation function.

2.1.2 Pruning Strategy

To reduce the redundant units, we need to prune the neurons in the hidden layer.

In this paper, we will use for loop to delete one hidden unit at each time. In the process of pruning neurons, we need to delete neurons whose output is always 0 or 1 or neurons with the similar or opposite function.We need the output of the hidden layer to detect these hidden units.

To detect the neurons whose output is always 0, we set a threshold at 0.01. if the sum of the output of a particular hidden neuron for each pattern is less than 0.01. We can assume that the output of this neuron is always 0. Then, we should delete it. To detect the neurons whose output is always 1, we set the threshold value at 0.9*675 (675 represents a total of 675 patterns), if the sum of the output of a particular hidden neuron for each pattern is greater than 0.9*675. We can assume that the output of this neuron is always 1. Then, we should delete it.

To detect hidden neurons with the same or opposite function, we can calculate the correlation between each neurons. The correlation coefficients range from -1 to 1. The greater the absolute value of the correlation coefficient, the stronger the correlation between the two neurons[5].So we should find the neurons in the hidden neurons that are highly correlated i.e. the absolute value of the correlation coefficient is greater than 0.9 and delete one of them.

To make the model converge faster, the weight of the neural network in the next training is initialized to the weight after the current neuron deletion.

This process is repeated until when we cannot find hidden units whose output is always 0 or 1 or the similar or different hidden units.

2.1.3 Optimization of the BP Neural Network

By training the neural network, we find that the training accuracy and testing accuracy of this model are both low. This means that the model is in an underfitting state. Based on the underfitting state of this model, I modify the model as follows:

Normalize the Input Data

The optimal solution is not equivalent to the original solution after the model is unevenly scaled in each dimension. Through normalization, the model parameters can be avoided to be dominated by data with too large range. The normalization can also improve the convergence rate of the model. In this way, the model becomes more optimized under the same number of iterations. For the data whose data distribution itself is skewed distribution(the first feature in LPQ), I first conduct logarithmic processing on the original data and then normalize it .The normalized results are shown in Fig.1.

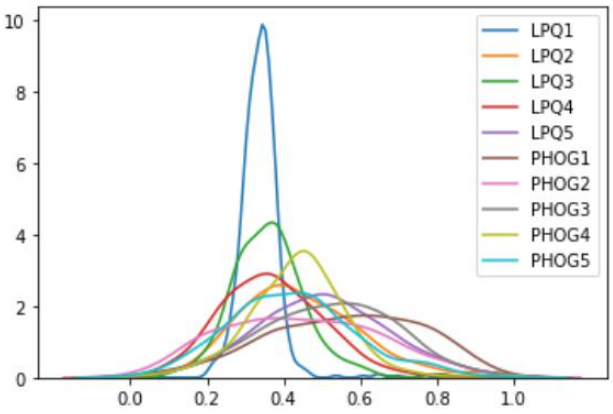

Fig. 1. The distribution of data after preprocessing

Implement K-fold Cross Validation

K-fold cross validation means we first divide the dataset into k subsets. Each subset is given a test set and the rest as the training set. The cross validation is repeated k times, each time a subset is selected as a test set, and the average cross validation recognition accuracy of k times is taken as the result. By doing this, We can have every data in the dataset tested and therefore get a model with stronger generalization results. There is very little data in the dataset, so I set k to

be 20. The purpose of this is to make as much data as possible for each training so that the model can better fit the data. The disadvantage is that the training time will be increase.

Use Adam Optimizer

The advantage of Adam optimizer is that it can comprehensively consider the first-order moment estimation and second-order moment estimation of gradient and calculate the update step size. The Adam optimizer is computationally efficient and memory intensive. Through it, the parameter update can not be affected by the scaling transformation of the gradient. It can automatically adjust the learning rate and is suitable for the problem of sparse gradient or high noise gradient. The comparison of SGD, SGD with momentum and Adam is shown in Fig.2.

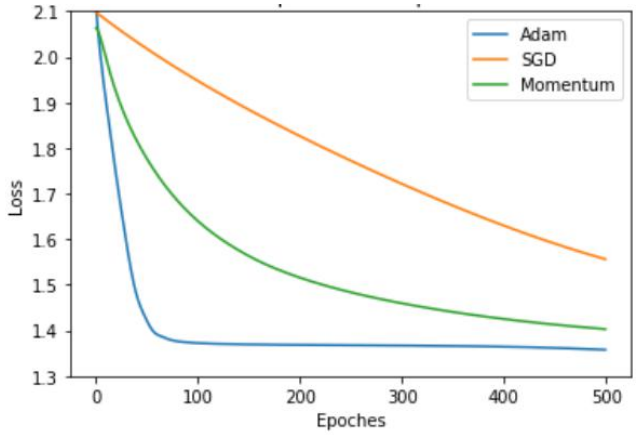

Fig. 2. The comparison of 3 different optimizer

2.2 Method for Face Emotion Classification(Image Dataset)

The basic model of face emotion classification for image dataset is the Resnet model. The specific structure is as follows:

2.2.1 Useful Functions

Conv2d(In_channels, Out_channels, Kernel_size, Stride, Padding) is used to convolve the image in two dimensions. 'In_channels' means the number of channels in the input image. 'Out_channels' means the number of channels produced by the convolution. 'Kernel_size' means the size of the convolving kernel. 'Stride' means the stride of the convolution. 'Padding' means zero-padding added to both sides of the input.

BatchNorm2d is used to normalize the data, so that the data will not be too large before Relu resulting in unstable network performance[7].

Maxpool is to reduce the impact of useless information. It is often seen in the shallow layer of the network, because the first few layers contain a lot of irrelevant information for the image.

Global average pooling can be used to replace the full connection layer. It avoids the need for a large number of parameters in the full connection layer. Another advantage of replacing the full connection layer is that it can support input of any size.

2.2.2 Structure of the Resnet Model

After the number of network layers reaches a certain degree, the errors of training and testing are higher and no longer decline. This is because the network is too deep for the model to optimize well. This is known as gradient extinction or gradient explosion. In theory, deep network is at least no larger than shallower network in terms of training loss, because each deep network can be regarded as a shallow network formed by adding some layers. Therefore, if we make these newly added layers into an identity mapping layer, the final result will be the optimal solution of shallow network. Resnet is the perfect solution to this problem.

The total structure of the Resnet model I used is shown in Fig.3 and it can be divided into 3 parts. For the first part, we put the images into model and use convolution,normalization and maxpool to extract preliminary features. Then, put

the features into building blocks. I set 8 building blocks and only show the first two building blocks. The solid and dotted curve lines on Fig.3 represent “shortcut connections”. It can be used when connections is an identity mapping without introducing additional parameters and computational complexity. For the last part, we use the global average pooling to replace the fully connected network. Finally, we set the number of output units to be 7 to do the face emotion classification.

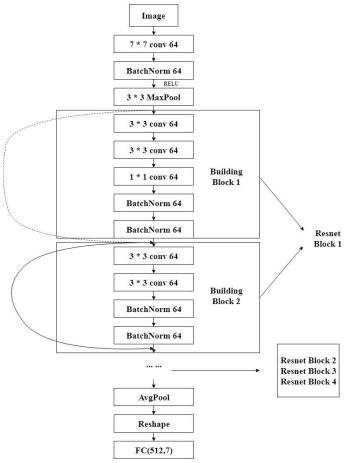

Fig. 3. The total structure of the Resnet model

3 Results and Discussion

3.1 Result of Pruning Hidden Units in BP Neural Network

Fig.4 shows the changing relationship between the number of pruned neurons and the value of loss. This graph shows that when we reduce the redundant hidden units and train again, the loss will not increase.

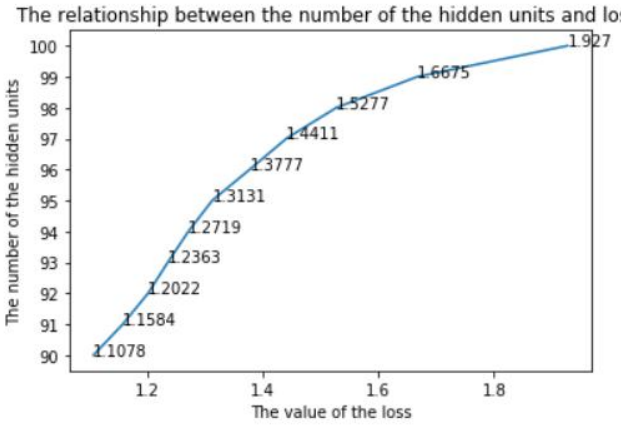

Fig. 4. The variation of loss as the number of neurons decreases

3.2 Result of the Comparison of BP Neural Network and Resnet model(Loss and Accuracy)

I use loss and accuracy as two criteria to compare the result of the BP neural network and the Resnet model. The results are shown in Fig.5 and Fig.6. As we can see, the training and test accuracy in the BP neural network are 41.36% and 11.62% respectively, which are both low. This means that this model has an under-fitting problem. However, for the Resnet model, the training loss is very small while the test loss is very high. Also, the training accuracy is around 95% but the test accuracy is only about 30%. So the Resnet model has an over-fitting problem.

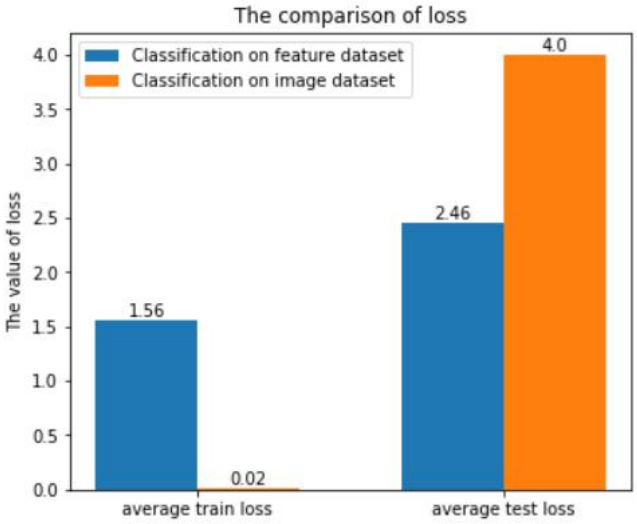

Fig. 5. The comparison of loss

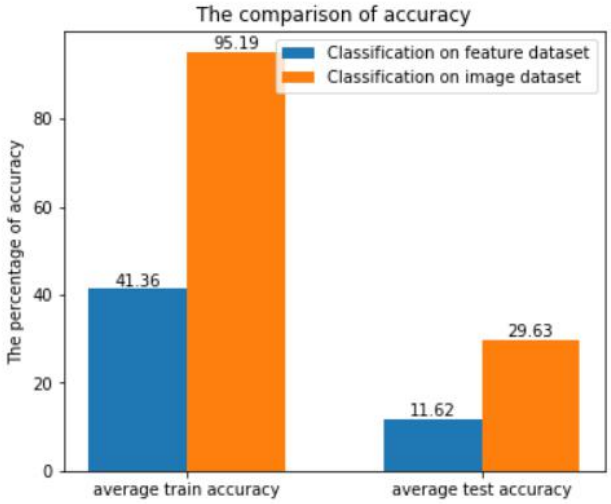

Fig. 6. The comparison of accuracy

3.3 Result of the Comparison of BP Neural Network and Resnet model(Precision, Recall and Specificity)

I calculate the precision, recall and specificity of each label for these two models. The results are shown in Table 1 and Table 2. By comparison, we can find that the precision, recall and specificity in the Resnet model are all higher than those in the BP neural network. Also, the precision, recall and specificity in the Resnet model are all higher than those in [1](except the recall of "Angry" and "Neural").

Table 1. The precision, recall and specificity for different labels(BP Neural Network)

	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Precision	0.03	0.02	0.14	0.27	0.08	0.14	0.05
Recall	0.03	0.01	0.16	0.29	0.07	0.15	0.06
Specificity	0.84	0.92	0.83	0.86	0.85	0.84	0.82

Table 2. The precision, recall and specificity for different labels(Resnet Model)

	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Precision	0.25	0.36	0.20	0.36	0.23	0.36	0.47
Recall	0.16	0.25	0.62	0.33	0.14	0.32	0.32
Specificity	0.99	0.99	0.94	0.99	0.98	0.98	0.99

4 Conclusion and Future Work

It can be seen from our results that the BP neural network has an under-fitting problem and the Resnet model has an over-fitting problem. Later, we can try to increase the complexity of the BP neural network to solve the issue. Also, we can address the over-fitting problems in Resnet by using dropout layer and doing regularization processing. We can also use some different models like CNN and GoogLeNet[8] to make the classification and then compare the accuracy with the Resnet model.

References

1. F. Zhou, F. De la Torre, and J. Cohn. Unsupervised discovery of facial events. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR'10, pages 2574–2581, 2010.
2. Dhall, Abhinav, et al. "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark." 2011 IEE E International Conference on Computer Vision Workshops (ICCV Workshops). IEEE.
3. Gedeon, T. D., Catalan, J. A., & Jin, J. Image Compression using Shared Weights and Bidirectional Networks. In Proceedings 2nd International ICSC Symposium on Soft Computing (SOCO'97) (pp. 374-381).
4. Gedeon, T. D., & Harris, D. (1992, June). Progressive image compression. In Neural Networks, 1992. IJCNN., International Joint Conference on (Vol. 4, pp. 403-407). IEEE.
5. Turner, H and Gedeon, TD "Extracting Meaning from Neural Networks," Proceedings 13th International Conference on AI, vol.1, pp. 243-252, Avignon, 1993.
6. Z. Lu, X. Jiang and A. Kot, "Deep Coupled ResNet for Low-Resolution Face Recognition," in IEEE Signal Processing Letters, vol. 25, no. 4, pp. 526-530, April 2018, doi: 10.1109/LSP.2018.2810121.
7. George Philipp, Dawn Song, Jaime G. Carbonell, "Gradients explode - Deep Networks are shallow - Resnet explained," in ICLR 2018 Workshop Submission.
8. Khan, R.U., Zhang, X. & Kumar, R. Analysis of ResNet and GoogleNet models for malware detection. J Comput Virol Hack Tech 15, 29–37 (2019).