Face Emotion Classification with Image and extract features

Can Yang[1]
Research School of Computer Science, Australian National University, Australia
U6921035@aun.edu.au

Abstract: Face expressions recognition is the most direct and effective way of emotion recognition. It is an indispensable part of human-machine interaction. Therefore, I used a static facial expression database Static Facial Expressions in the Wild (SFEW) and intend to train a model to better classify the face emotions. I tried the original images and the data obtained through feature extraction to classify facial expressions separately. The purpose of this paper is to present the results of classifying different data sets and to discuss methods for optimizing loss and accuracy.

Keyword: Classify Face Emotion, Compare image data and feature data.

1. Introduction

In this paper, we will construct a better face emotion classification model. The results will be compared with the literature. Facial expressions are very important non-verbal means of communication. Human emotions are divided into 7 categories: Angry, Disgust, Fear, Happy, Neutral, Sad and Surprise. [2] Facial expression recognition is a cross-cutting technology that spans the fields of artificial intelligence, neurology and psychology, etc. And has wide applications in human-machine interaction, psychoanalysis and clinical medicine. For example, psychological states such as depression can be inferred from a person's facial expressions to detect a patient's clinical progress based on a pain test, etc.

There are many ways in which we can solve the classification problem. There are traditional machine learning algorithms such as SVM, decision tree, logistic regression and even k-nearest neighbor. Each of these methods can address the classification issue. But more advanced neural network approaches may have better results. Since the neural network is inspired by biological neurons, they are more 'talented' in learning. So, I used neural networks to classify facial expressions.

Here are the original image and the extracted feature data. For the different data format we need to use different methods to de the classification. For the original image, CNN should be used for feature extraction and followed by a classification layer. For data that has been extracted features, it is not necessary to use CNN. Because it is not need to perform feature extraction, so all it has to do is classify based on the feature data. The classification of different forms of data must have different results, so I will compare them with the results in the literature. And finally we get a better model for facial expression classification.

2. Method

2.1 Face Emotion Classification with feature data

Preprocessing data

In the feature dataset, there is the label and features extract by two different ways. The label here is clearly categorical data, with seven categories from 1-7 representing different face emotions. And there are two different ways of obtaining the features of the image, they are Local Phase Quantization (LPQ) and Pyramid of Histogram of Gradients (PHOG).[5] There are five features were extracted from each method, and I constructed neural networks to train them individually. Since we only got the processed data by PHOG and LPQ. The local phase quantisation (LPQ) extract the detailed

information of the image. And the PHOD extract the gradient information of the image. I used the data combine PHOG and LPQ features. Because they are features extracted according to different methods, so combining them can get more comprehensive and complete features.

Then I found that these data are uneven, and I want to standardize the data to make them evenly distributed. The below diagram shows the standardized data.

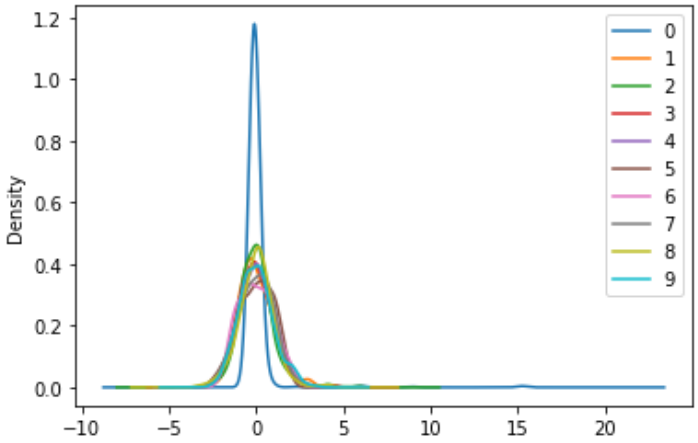

Fig. 4. This figure shows that the standardized dataset.

After that, I think the data should be segmented. So, I randomly split the data into training sets and test sets. In the process of training the neural network, I used the mini-batch method to train the model. The mini-batch gradient decent method divides the data into several batches, and updates the parameters according to the batch. In this way, a group of data in a batch jointly determines the direction of the gradient, so it is not easy to deviate and decrease the randomness. On the other hand, because the number of samples in the batch is much smaller than the entire data set, the amount of calculation is not very large.

Construct the Neural Network

This is followed by the most important part, the construction of the classification neural network. At first, I tried to build a neural network with two hidden layers. I found its accuracy to be low and tried many ways to optimize the model, however, it didn't improve obviously. Then I found out that almost all the values in the results would predict into one category, which shows that the model is indeed very flawed. This problem was solved after changing the number of hidden layer of the neural network to be one. All predictions won't be the same when there's only one hidden layer. By probing, I think this is mainly due to the fact that the initial data is so small, with only 10 features, that deeper neural networks will cause features to be hidden by hidden layers. Until that point I was naive enough to think that the more hidden layers in a neural network, the better this neural network would be at making predictions. So I finally construct a neural network with one hidden layer. The below diagram shows the structure of neural network.

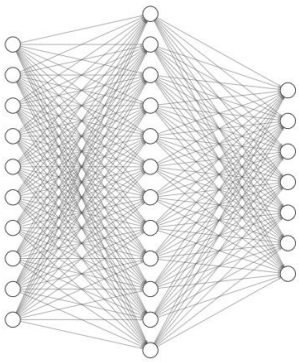

Fig. 5. This figure shows the structure of the classification neural network.

Pruning the NN

Distinctiveness analysis has been used to analyze the function of units in the pattern space to determine whether there are redundant units, to accelerate the learning speed by ensuring the distinctness of the unit functions, and to improve the network's resistance to destruction and robustness.[6]

The distinctiveness of the hidden unit is determined by the unit output activation vector on the pattern presentation set.[3] We only need to calculate the result of each pattern after passing through the neural unit of the hidden layer and its activation function (that is, the data input to the output layer), and the output activation vector of each hidden layer unit is the result of cumulative of all pattern after the calculation above. That is, for each hidden unit we construct a vector of the same dimensionality as the number of patterns in the training set, each component of the vector corresponding to the output activation of the unit.[3] By comparing these vectors we can get the vectors that should be removed most. The vector that should be removed most is the one with all values of 0. Such a vector is useless in a neural network, and the angle of computation with other vectors is almost 0. Then the very similar vectors, similar vectors with similar functions and opposite vectors with inverse functions. By calculating the angle between these vectors, we can know how similar they are. Set a threshold that removes all vectors smaller than it one by one.

2.2 Face Emotion Classification with Image

CNN is best at images processing. It is inspired by the human visual nervous system. It can effectively reduce the dimensionality of images and retain the image features. And in this paper we use GoogLeNet for image classification. GoogLeNet better simulates the biological neural network.

In general, the most direct way to improve neural network performance is to increase the depth and width of the neural network, which means a huge amount of parameters. However, the huge amount of parameters that are prone to overfitting will also greatly increase the complexity of calculation. The core idea of GoogLeNet believes that the solution to the above two drawbacks is to convert fully connected or even general convolutions into sparse connections. On the one hand, the connection of the real biological nervous system is also sparse, on the other hand, literature[7] shows that for a large-scale sparse neural network, it can be constructed layer by layer by analyzing the statistical characteristics of activation values and clustering highly relevant outputs An optimal network. This indicates that the bloated sparse network may be simplified without losing performance.

Due to the large amount of image data, I compressed all the images to a size of 96 * 96 for the convenience of calculation, so that the data processing can be faster, and a lot of time can be saved during training. Then I randomly divided the data set into 80% training set and 20% test set.

Inception

1. Using different sizes of convolution kernels means different sizes of receptive fields, and finally stitching means the fusion of features of different scales;

2. The reason why the size of the convolution kernel is 1, 3 and 5 is mainly to facilitate the alignment. After setting the convolution step stride = 1, as long as pad = 0, 1, 2, respectively, then the features of the same dimension can be obtained after convolution, and then these features can be directly spliced together;

3. The further back the network, the more abstract the features, and the larger the receptive field involved in each feature, so as the number of layers increases, the proportion of 3x3 and 5x5 convolutions also increases.

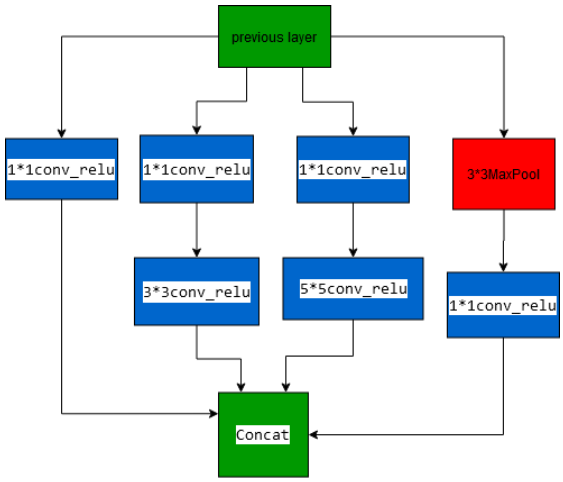

Figure 1. This figure shows the structure of inception block.

GoogLeNet

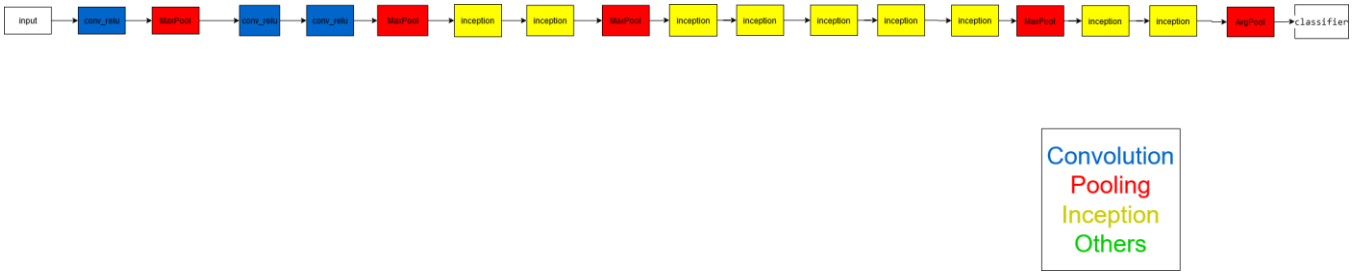

Figure 2. This figure shows the structure of GoogLeNet.

1. GoogLeNet uses a modular structure (Inception structure) to facilitate addition and modification;

2. The network finally uses average pooling to replace the fully connected layer. The idea comes from NIN (Network in Network). It turns out that this can increase the accuracy by 0.6%. However, a full connection layer was actually added at the end, mainly to facilitate flexible adjustment of the output.

Optimizer

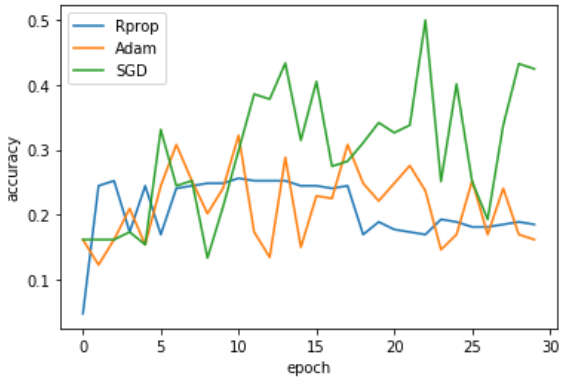

Figure 3. This figure shows the accuracy of models trained by different optimizers.

I tried different optimizers and found that SGD works best. Rprop will quickly reach its best situation, and then almost maintain that as the number of training increases, the accuracy even deteriorate. Adam has a lot of fluctuations. During the training process, it has been fluctuating between 0.1 and 0.3, which is very unstable, and the accuracy has not been improved to higher. Although the SGD fluctuates, the overall accuracy rate is constantly increasing. And it tops out at more than 50% accuracy, which is even higher than the accuracy in the paper.

3. Results and Discussion

Even though I have tried many ways to optimize the neural network, the results are still unsatisfactory, and we can see that the accuracy of the model is still not high. I think the reason for not making good predictions is that the external environment of the image in SFEW dataset is very complicated. It is the wild facial expression obtained from the movie. Because there are many interference factors, its accuracy cannot be very high.

There are the mean accuracy by train 30 times.

Table 1. the train and test mean accuracy of Image and feature data

	Train	Test
Feature data	98.80%	21.26%
Image	98.69%	39.90%

The table above shows the Image classification and feature classification after training 30 epochs. We can see that the average accuracy of GoogleNet using images for classification is almost 40. However, the accuracy of model using features data for classification can only reach more than twenty.

This shows that the model trained using the original image is still better than the feature extraction data, although the image data is quite large.

We can see from the two pictures below that the feature classifier accuracy of the training set is very high from the beginning. And it almost reach 100%. However the accuracy of test set is not good. It fluctuates between 0.2 and 0.3, and has not significantly improve. For GoogLeNet, the accuracy of the training set and the accuracy of the test set are not high at first. After about 15 times training, the accuracy of the training set has reached more than 90%, and the accuracy of the test set is also more than 30%. And it can reach more than 50%, which is higher than the accuracy rate in the literature. And it seems that it still has increasing tendency. This shows that it is feasible and necessary to use all the images for training. In this way, facial expressions can be better classified.

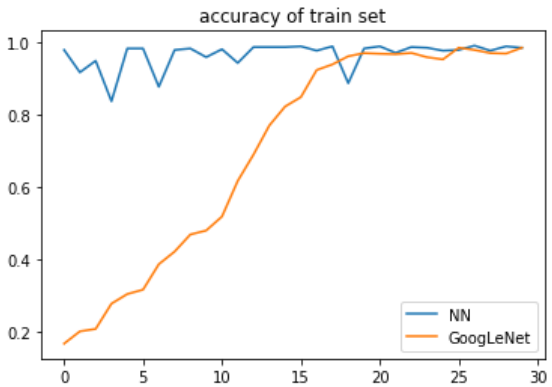

Fig. 6. This figure shows that the accuracy of training set by NN and GoogLeNet.

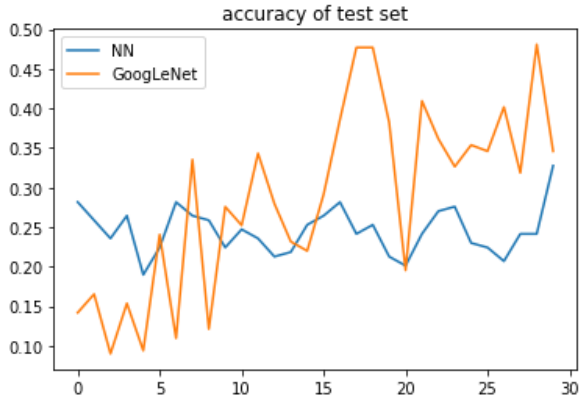

Fig. 7. This figure shows that the accuracy of test set by NN and GoogLeNet.

I compared my model with the data in the literature and calculated Table 2. I found that most of my data is higher than the original literature. This shows that my model is more suitable for SFEW facial expression classification than the model it uses. That's also reflects the benefits of using original image data rather than feature data.

Table 2. the precision recall and specificity of each class

	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Precision	0.68	0.50	0.56	0.54	0.18	0.63	0.20
Recall	0.63	0.26	0.42	0.61	0.12	0.63	0.37
Specificity	0.99	0.99	0.99	0.98	0.99	0.99	0.96

4. Conclusion and Future Work

We have built a convolutional neural network which is used to classify the face emotion based on SFEW dataset. Although the neural network I have constructed is not very good in terms of evaluation, I intend to identify faces in the images. Due to the complex environment of the SFEW data set, the model will always produce deviations when extracting features. This causes us to be unable to accurately recognize facial expressions. If we recognize the face part in the picture and use it as data, the classification accuracy can be improved very well. Using face extraction as data preprocessing may improve the robustness and reliability of the model.

Reference

1. T.D. Gedeon & D. Harris , F.: PROGRESSIVE IMAGE COMPRESSION
2. T.D. Abhinav Dhall, Roland Goecke, Simon Lucey, Tom Gedeon , F.: Static Facial Expression Analysis in Tough Conditions: Data, Evaluation Protocol and Benchmark
3. T.D. Gedeon & D. Harris , F.: "NETWORK REDUCTION TECHNIQUES"
4. Gedeon, TD & Harris, D, “Creating Robust Networks,” IJCNN, Singapore, 1991.
5. Abhinav Dhall ; Akshay Asthana ; Roland Goecke ; Tom Gedeon , "Emotion recognition using PHOG and LPQ features", Face and Gesture 2011
6. Harris, D, Gedeon, TD, “Adaptive insertion of units in feed-forward neural networks,” Neuro-Nîmes, 4th Int.Conf. on Neural Networks and their Applications, 1991.
7. Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. CoRR, abs/1310.6343, 2013