Facial Emotion classification using convolution neuron network*

Zhijie Guo

Austrilian National University
u6919082@anu.edu.au

Abstract. Facial emotion classification has always been a popular topic and have been put into practice using lots of methods. Here I use convolution neuron network to classify the static facial images based on the dataset face-emotion (The static database has been extracted from the temporal dataset Acted Facial Expressions in the Wild (AFEW))[1]. I did the classification using the images in the dataset. I used a 5-layer convolution neural network to do the job. I trained the network with training set and evaluated the result using accuracy(the probability that the prediction is correct).

Keywords: Facial Emotion classification · Convolution neuron network · neuron network.

1 Introduction

1.1

Image classification has always been an important problem in machine learning. Convolution neural network (CNN) can simulate the establishment of biological visual perception mechanism, and it can carry out supervised learning and unsupervised learning. The convolution kernel parameters shared by the underlying layer and the sparse characteristics of inter layer connections make the convolution neural network have less computation. I used very simple and basic CNN to implement our classification task in order to test the latter technique. It is a feed-forward network with five layers (Figure 1).

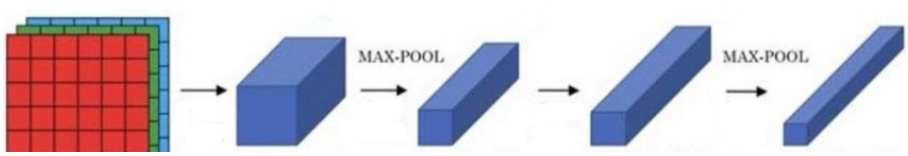

Fig. 1.

* Supported by organization x.

1.2

The first and third layers are convolution layer, the second and fourth layers are pooling layers, and the last layer is output layer. Each layer contains parameters that can be trained, and each layer has some feature maps. An input feature is extracted in each layer using a convolution filter, and each feature map has neurons. Although the eventual result is not satisfying, my results are still better than my previous report. The number of hidden neurons is also hard for me to determine and requires further study.

2 Method

Fig. 2.

2.1 Pre-process

First I transform the RGB images to grey-scale images because grey-scale images only have one channel and is easier to handle. Then I reshape the images to be 90*72 which is originally 720*576 and is too big. And gaussian filter is applied to reduce the noise thus keep the network from the its influence but the performance is influenced so I gave up the filter. Than the data is shuffled and split into training data and testing data with proportion 8:2.

2.2 Network

Then I built the network with similar structure of LeNet5[2], and the structure is shown in Fig.1 above. It's a feed-forward network containing 5 layers: conv1, pool1, conv2, pool2, out. The first and third is convolution layers and the second and fourth is pooling layers and each layer have some feature maps and parameters to be trained. And then I chose the hyper parameters by testing the combinations below.

Table 1. Combinations

Output channel	Conv1	Conv2
1	16 output channel	32 output channel
2	12 output channel	24 output channel
3	20 output channel	40 output channel
4	6 output channel	12 output channel

Table 2. Results

data	1	2	3	4
train loss	0.1154	0.2693	0,1263	0.7089
train accuracy	99.23	97.21	98.17	85.66
test loss	0.1154	0.2693	0,1263	0.7089
test accuracy	36.32	32.89 n	31.13	25.24

2.3

So the first combination performs the best on training and testing set. But due to the time that training takes is too long so i just tested each combination once and the result may be slightly different if it's carried multiple times.

3 Result

3.1 Training Accuracy and Loss

The training accuracy surges at the beginning and converges after 600 epochs, and loss keeps decreasing. Fig.3 and Fig.4

3.2 Testing Accuracy and Loss

The testing accuracy surges at the beginning and fluctuate after 800 epochs, and loss keeps decreasing. Fig.5 and Fig.6

3.3 Overall

The result is quite compared with my last report and I think the limit maybe come from the data set that the face is not edxtracted perfectly.

4 Conclusion

4.1

I built a convolution neural network by imitating the lenet5 structure[2] based oi sfew database, and the effect and accuracy of the convolution neural network for facial expression are tested. And the result is quite satisfactory compared to my previous report using simple 3-layer neuron network. And I think later I should use some other techniques like pruning and input encoding to improve the result.

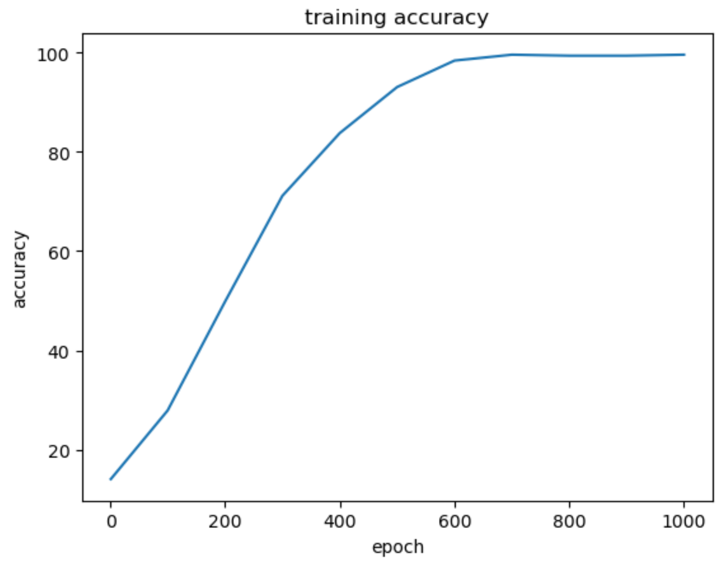
training accuracy
accuracy
epoch
100
80
60
40
20
0
200
400
600
800
1000

Fig. 3.

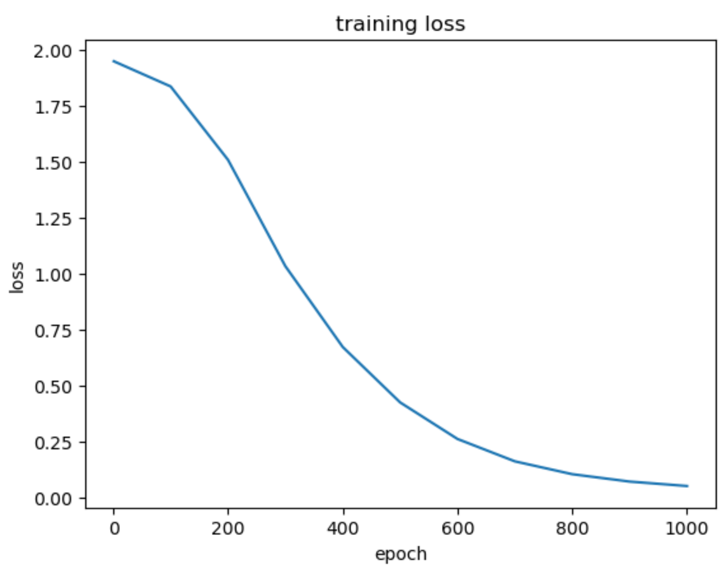
training loss
loss
epoch
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
1000

Fig. 4.

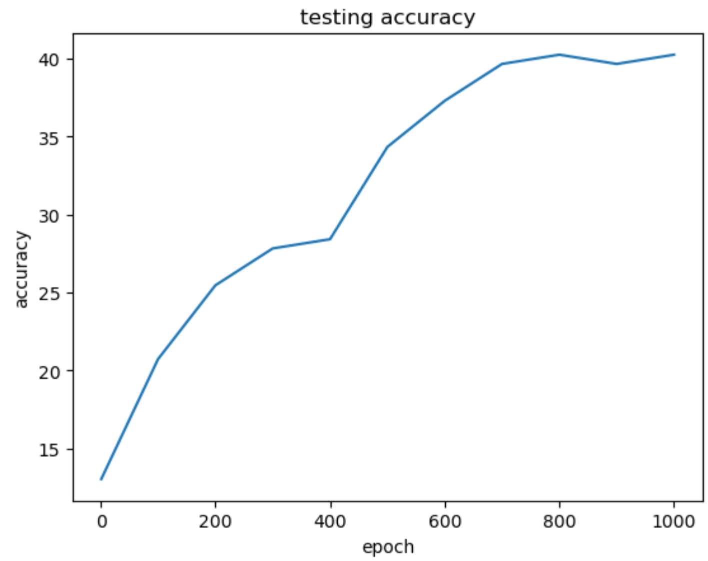
testing accuracy
accuracy
epoch
40
35
30
25
20
15
0
200
400
600
800
1000

Fig. 5.

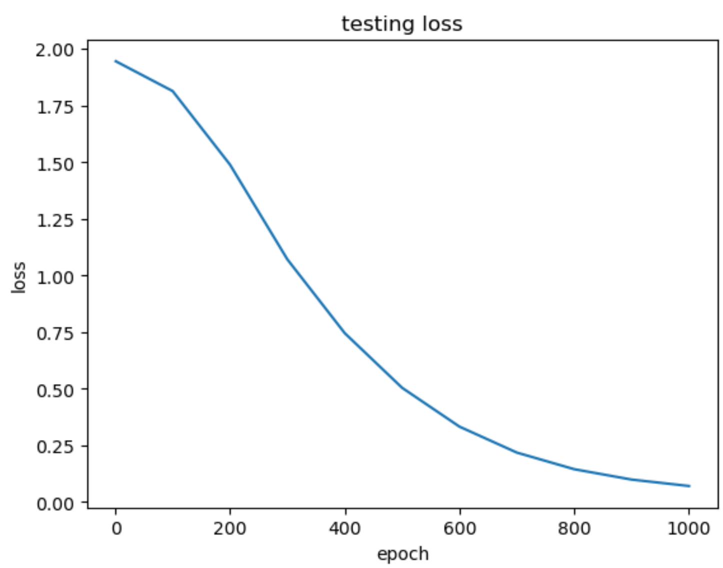
testing loss
loss
epoch
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
1000

Fig. 6.

References

1. Author, Abhinav Dhall, Roland Goecke, Simon Lucey, Tom Gedeon, “Static Facial Expression Analysis in Tough Conditions: Data, Evaluation Protocol and Benchmark” , 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, 2011.
2. Author, Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, ”Gradient-based learning applied to document recognition,” in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.