# Facial expression classification and progressive image compression technology application based on *SFEW* database

Tian Luan
Australian National University
Acton, ACT, Australia
u6920351@anu.edu.au

**Abstract.** Classification problem on images is always important problems in machine learning. In past people learn a lot about how to classify the facial expression in standard and strict constrained environment. Here we use convolutional neural networks to classify facial expressions in wild based on SFEW database. SFEW database is Static Facial Expressions in the Wild, extracted from facial expressions database AFEW, which are extracted from movies. How to determine the number of hidden neurons at the same time also is the important problem of the study of the neural network, Progressive image compression (PIC) technology proposes a technique that allows us to identify similarities in the functions of different neurons, so that we can prune hidden neurons according to whether they are similar or not. Here we are inspired by this theory that we can find a standard to prune hidden units and we choose the importance as the standard of pruning the neural network.

**Keywords:** facial expressions classification, convolutional neural network, hidden unit prune

## 1 Introduction

Classification problem on images is always important problems in machine learning. Convolutional Neural Network (CNN) imitation biological visual perception mechanism building, can undertake supervised learning and unsupervised learning, the convolution kernel parameters of sharing in the underlying layer and interlayer connection of sparse characteristics enables the convolutional neural network to have a smaller amount of calculation. To facilitate our testing of the latter techniques, we use very easy and basic CNN to implement our classification task. It is a feed-forward

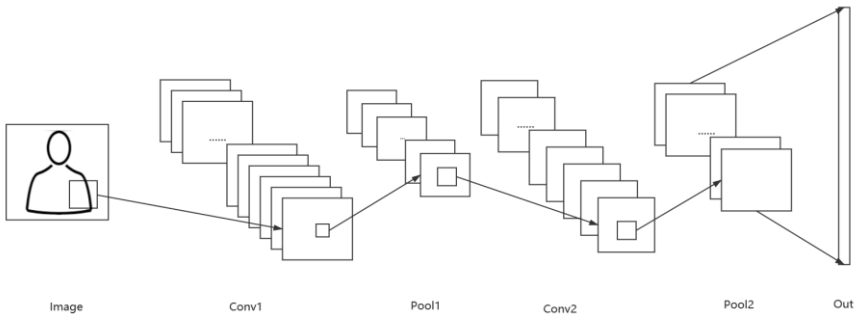

Fig. 1. The structure of convolutional neural network

network which only contains five layers (Fig.1.). The first and third layer are the convolution layers, the second and fourth layers are the pooling layers and the last layer is the output layer. Each layer contains trainable parameters, and each layer has multiple Feature maps. Each Feature Map extracts an input Feature through a convolution filter, and then each Feature Map has multiple neurons. It is hard to get satisfactory results, but the result we get still much better than the result we get in the previous paper *"Facial expression classification and progressive image compression technology application based on SFEW database"*, and it can help us to verify the performance of technique inspired by PIC technology [1]. How to determine the number of hidden neurons at the same time is another important question which we want to research, obviously if we can't use more than the minimum number of neurons we will not be able to get a good result, but too much neurons will lead to slow training speed. So we got the idea from PIC technology to prune the neurons. In PIC tech, we calculate the angle of the output from neurons and prune neurons by similarity. So we will first test the technology in Image Compression neural network. And then similar as PIC we decided to prune the neurons of CNN with a standard. Here we choose importance as the standard, because in convolution neural network, the feature maps in convolution layer are not identical, to evaluate similarity of them is very difficult, but we can evaluate the importance of each feature map, and prune the neural network by comparing importance. We will test this technology in our convolutional neural network. We will use two databases for meeting the above requirements, first is SFEW only with LPQ and PHOG, the second one is the SFEW database only contains images.

# 2 Method

## 2.1 Progressive Image Compression

Firstly, we want to test neuron pruning technology in *PROGRESSIVE IMAGE COMPRESSION* [1]. We only want to test prune technology so we do not care about the real meaning of the features in the database and the structure of the compression network. To test it, we construct a very simple neural network. It is a feed-forward network which only contains three layers (Fig.1.) all connections are from units in one level to the subsequent one, with no lateral, backward or multilayer connection. Each unit has a simple weighted connection from each unit in the layer above. The hidden layer uses a sigmoid function as the activation function. Input and output layer use linear function, the structure of the neural network is shown in Fig.2. Because in compression neural network we need to recover the compressed data, so the input size and output size are the same and they are both 11 because after we drop the description columns we have 11 feature columns. To compress the data,the hidden layer should consists fewer units than input layer and output layer, here we use 10 units at first. we use back-propagation method to update the weights and train model.

After training the network, we take the hidden layer output as vectors and check the angle between these vector pairs in pattern space. To prune the hidden layer we use distinctiveness to check if we should prune a unit or not. We choose 15° as the criterion for whether the two vectors are similar. That means if the angle is smaller than 15° or larger than 165°(They are complementary), we removed one of them. We also remove the units which outputs are all zero, that means they are not useful so we remove them. The weight vector of the unit which is removed is added to the weight vector of the unit which remains. Then we check the loss of the neural network as the evaluation criteria. We also use 20% data as test data to test the performance and evaluation criteria is the loss.

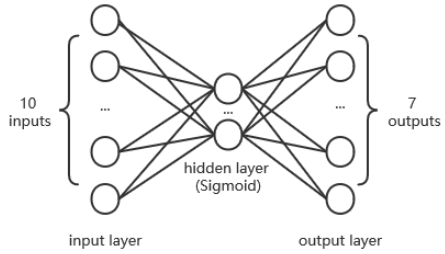


Fig. 2. The structure of Image Compression Neural Network

## 2.2 Facial Expression Classification Convolutional Neural Network

First we check some samples from the SFEW dataset and some of them will be shown follow:

Fig.3. Some Samples from SFEW

Then we convert them from RGB image to gray-scale image so we can handle them more easily be cause the gray-scale image only has one channel. Because the sizes of images are all 720*576 it is too big for us to train the neural network in the future, we resize these images to 90*72. We also try to use Gaussian filter to process the images but the result is not good so we do not gaussian blur the images here. And we shuffle the data because the raw data put the same facial expression together. To accelerate the convergence speed of the training step, we also normalize the image data. Now we have preprocessed these images. 20% data will be randomly chosen as the test data.

Then we built the convolutional neural network which imitated the structure of LeNet5 [4]. We use the network structure in Fig.1. It is a feed-forward network which

only contains five layers (Fig.1.). The first and third layer are the convolution layers, the second and fourth layers are the pooling layers and the last layer is the output layer. Each layer contains trainable parameters, and each layer has multiple Feature maps. Each Feature Map extracts an input Feature through a convolution filter, and then each Feature Map has multiple neurons. To determine the hyper parameters of the neural network, we test different output channel and input channel combination, the result shows in following table:

(1)conv1 layer: 16 output channel conv2 layer: 32 output channel
(2)conv1 layer: 12 output channel conv2 layer: 24 output channel
(3)conv1 layer: 20 output channel conv2 layer: 40 output channel
(4)conv1 layer: 6 output channel conv2 layer: 12 output channel

* Above are the combinations we have tried
* We train 700 epochs because the result basically converges at this time.
* We run 5 times and calculate the average result

| | Combination (1) | Combination (2) | Combination (3) | Combination (4) |
|---|---|---|---|---|
| Average Train Loss | 0.12 | 0.28 | 0.11 | 0.71 |
| Avg Train Accuracy | 99.01% | 97.83% | 99.80% | 84.01% |
| Average Test Loss | 0.12 | 0.28 | 0.11 | 0.71 |
| Avg Test Accuracy | 36.69% | 33.28% | 32.32% | 27.21% |

At last we choose the combination (1), because it has the best performance and the fast convergence speed can effectively save our training time. Then the input of conv1 layer has 1 channel because the images are gray-scale, the output channel is 16, the input of conv2 layer is 16 and output channel is 32. Pooling layers we choose here are both max pooling and each size is 2 and 3. Because Facial Expression Classification is also a classification problem, so we also choose the cross entropy loss as the loss function. For optimizer, we found the Adam has the best performance and we determine to use it at last.

### 2.3 Neurons Pruning in Facial Expression Classification Neural Network

We were inspired by *PROGRESSIVE IMAGE COMPRESSION* [1] to come up with this standard of neuron pruning, using importance as the evaluation which neurons need to be pruned. Here we use L2-norm to calculate the importance of every convolution kernel, this method is also mentioned in *Pruning optimization based on deep convolution neural network* [2]. The importance of kernels will be calculated as:

$$I^i = \sqrt{\frac{1}{k^2}\sum_{x}^{k}\sum_{y}^{k} a_{x,y}^2}$$

Here k is the kernel size, $\alpha$ is a weight in the kernel i. Then we picked out the kernel with the minimum importance and then set its weight to 0 for pruning. Here we only apply this pruning technology on the first convolution layer for test. We also set a threshold to determine when we need to stop pruning, which is when the loss

changed larger than 0.1 we will stop pruning. Also, 20% data will be chosen as the test data, for the evaluation standard, we choose loss and accuracy.

# 3 Results and Discussion

## 3.1 Result in Progressive Image Compression

Fig.4. shows the result of using PIC method to compress the SFEW database and the relationship between the number of hidden units and the smallest angle between hidden units, in order to clearly compare the image in *PROGRESSIVE IMAGE COMPRESSION* [1], we set the loss and minimum angle to the x-axis and the number of hidden neurons to the y-axis. Here for testing purposes we do not use 15° and 165° to dropout the units. The result we get is the loss is shown as the following table:

| Number of units | Loss | Min Angel |
|---|---|---|
| 9 | 0.0064 | 0.15° |
| 8 | 0.0198 | 0.197° |
| 7 | 0.0549 | 0.61° |
| 6 | 0.105 | 1.08° |
| 5 | 0.16 | 1.76 |
| 4 | 0.32 | 9.07° |
| 3 | 0.53 | 18.54° |
| 2 | 0.569 | 62.64° |

At Last, test loss is 0.1523.

It is clear that the loss is increased while the smallest angle is decreased. Figure 5. is from *PROGRESSIVE IMAGE COMPRESSION* [1]. Compare the two figures, we can find that the result is similar and we actually prune the hidden layer while keeping the loss.

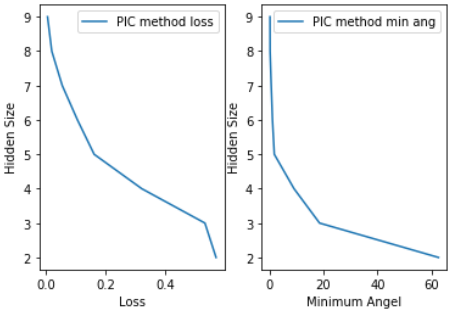


Fig.4. The loss and min angle of PIC

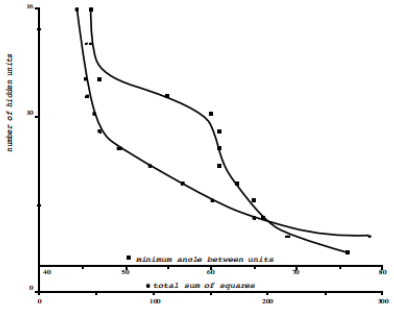

Fig.5. Units vs image quality, unit significance

### 3.2 Result of Facial Expression Classification Convolutional Neural Network

We choose one of our training processes to show the result of facial expression classification convolutional neural network. We use accuracy and loss to evaluation the network. The following table shows the result (We train 700 epochs):

| Epochs | Train Loss | Train accuracy |
|---|---|---|
| Initial | 1.9478 | 16.60% |
| 100 | 1.8552 | 22.53% |
| 200 | 1.5861 | 43.48% |
| 300 | 1.1564 | 66.01% |
| 400 | 0.8196 | 80.43% |
| 500 | 0.5781 | 89.13% |
| 600 | 0.4056 | 93.67% |
| 700 | 0.2831 | 97.04% |

At last test accuracy is 37.8698224852071%.

Obviously, there is a big gap between the accuracy of the test and the accuracy of the training, but it is better than the result of the previous paper. This phenomenon may be caused by neural network overfitting, but I think this is also because our training data and test data are in unconstrained situation, so the face location and the location of the facial features are not stable, it makes difficult for the convolution layer of neural network to extract features. I think if we want to improve the accuracy, we should combine this technology with facial recognition technology and do another data preprocessing.

### 3.3 Result of Neurons Pruning in Facial Expression Classification Neural Network

The following table shows how the trian loss and train accuracy changed with the number of pruned neurons by using our technology. It is clearly that when we delete the convolution kernels which has low importance, loss and accuracy won't change too much, here the loss changed from 0.28 to 0.55 and the accuracy changed from 97% to 81%, if we obey the threshold we set (If the loss changed more than 0.1, we will stop), the train loss will be 0.33 and accuracy will be 95%, we lose almost no performance.

| Number of Pruned Neurons | Train Loss | Train accuracy | Minimum Importance |
|---|---|---|---|
| 0 | 0.2831 | 97.04% | 0.528 |
| 1 | 0.2831 | 97.04% | 0.579 |
| 2 | 0.2831 | 97.04% | 0.7116 |
| 3 | 0.3295 | 95.26% | 0.7119 |
| 4 | 0.5569 | 81.82% | 0.7413 |

After pruning the test loss is 0.2841 and accuracy is 34.32%, also we lose almost no performance. The following figures Fig.6., Fig.7. and Fig.8. visually shows how

the trian loss and train accuracy changed with the number of pruned neurons by using our technology:

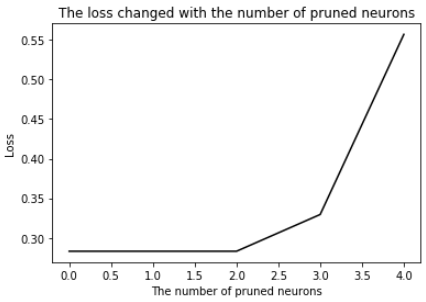


Fig.6.

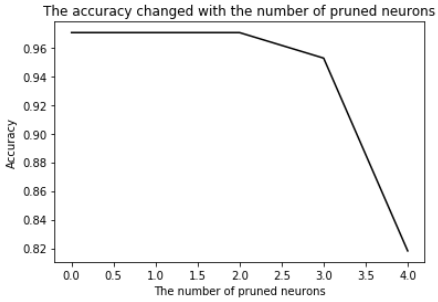


Fig.7.

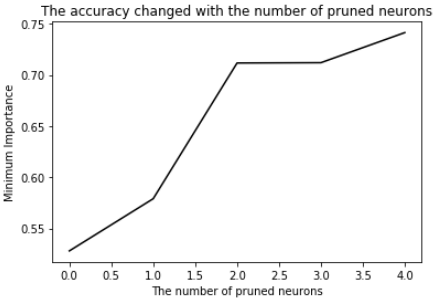


Fig.8.

## 4 Conclusion and Future Work

Firstly, we established a very simple neural network to verify the neuron pruning technique in *PROGRESSIVE IMAGE COMPRESSION* [1], then we confirmed that this idea is feasible. After that we built a convolutional neural network based on SFEW database and imitated LeNet5 structure [4], and tested the effect and accuracy of facial expression convolutional neural network on images which are under unconstrained environment. Finally, we use the inspiration from *PROGRESSIVE IMAGE COMPRESSION* [1] to come up a technology that regard importance as a standard of pruning the neural network, and test this technology on our convolutional neural network to prune some unimportant convolution kernels. The results told us this technology can be effective in the preservation of neural network function at the same time reduce the neurons in convolutional neural network, improve the training speed and reusability. Due to the simple structure of our convolutional neural network, this neuronal pruning technology based on importance can be extended to other convolutional neural networks.

## 5 The References Section

## References


**[1] T.D. Gedeon1 & D. Harris, "PROGRESSIVE IMAGE COMPRESSION", Neural Networks, 1992. IJCNN., International Joint Conference on (Vol. 4, pp. 403-407). IEEE.**

**[2] Ma Zhinan，Han Yunjie，Peng Linyu，et al. Pruning optimization based on deep convolution neural network[J]. Application of Electronic Technique，2018，44(12): 119-122，126.**

**[3] 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.**

**[4] 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.**

**[5] Enzo Tartaglione, Andrea Bragagnolo, and Marco Grangetto, "Pruning artificial neural networks: a way to find well-generalizing, high-entropy sharp minima" , arXiv: 2004.14765v1**

**[6] Ragav Venkatesan, Gurmurthy Swaminathan, Xiong Zhou, and Runfei Luo, "Out-of-the-box Channel Pruned Networks", arXiv: 2004.14584v1**

**[7] Viacheslav M. Osaulenko, "BINARY AUTOENCODER WITH RANDOM BINARY WEIGHTS", arXiv: 2004.14717v1**

**[8] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, Yasaman Khazaeni, "FEDERATED LEARNING WITH MATCHED AVERAGING", ICLR 2020**