Classification of Static Facial Expression Using Feedforward Neural Network with Network Reduction Technique

Xinyi Han
School of Computer Science, Australian National University
u6212385@anu.edu.au

Abstract: This research provides two methods which are LPQ and PHOG to extract important data as initial data of face images. All these face images are be labeled to 7 types of human emotions. The goal of this study was to develop a feedforward neural network to classify 7 types of human emotions. Meanwhile research aim to using a network reduction technique to minimize computation resources usage. When the implementation of the strategy, the study result shows that using the normalized data to training neural networks can prevent network overfitting, otherwise long iteration time would cause network achieved higher training accuracy and lower testing accuracy. Use K-Fold training neural network is better than just use splitting data one time to train the network. The study shows that in the K-Fold training use one database and one neural network but try to adapt to different training sets which equivalent to an expanded database and improve the generalization ability of the neural network. Due to low training/testing accuracy, research use another method, which is K neighbor classifier in machine learning, comparison, and contrasting with the feedforward neural network. Finally, to improve the performance of neural networks, the researcher converts 7 types of emotion into 4 types of emotion.

Keywords: Feedforward neural network, Pruning, Network reduction, Data outliers, Data normalizing, Data balance, K-Fold, Split data, K neighbor classifier.

1. Introduction

Original datasets have two methods to collect important parameters of 700 images which are selected from 957 images of screenshots of 37 films. For two methods, each method provides 5 parameters, the method LPQ is based on computing the short-term Fourier transform (STFT) on a local image window(Dhall, 2011), the method PHOG (pyramid of the histogram of oriented gradients) (Dhall, 2011) is to use counts occurrences of gradient orientation in localized portions of an image. The data provided by each method include side effects which are unconstrained facial expressions, varied head poses, large age range, occlusions, varied focus, different resolution of the face, and the simulation to real-world illumination. These side effects both exist in the laboratory environment and film making environment, however, film making environment has more natural facial expressions, more natural poses, and the close to real-world illumination, because actors always perform better than volunteers' performance which is in the laboratory environment.

Measurement of facial motion is a more scientific method of facial expression analysis (Ambadar, 2002) because some expression is reflected in the change in a person’s facial muscles. But the datasets for this study is to use Static Facial Expressions (which named SFEW (Dhall,2011)). The reason for facial expression analysis methods is image-based, not video-based is that encodes the facial dynamics is more complex, needs more parameters in motion datasets, need temporal data.

Separate database to two different data collection method datasets and remove outliers for training correctness and normalized two datasets except the ‘label’ attribute in datasets for preventing the feedforward neural network overfitting and for datasets easier fitting in those activation functions which including in the network. Then derive two datasets as LPQ.csv file and PHOG.csv file.

Create a three layers feedforward neural network which belongs to the deep neural network and considers with huge computation tasks during the network training, the researcher pruning the second hidden layer to minimize computation resources usage. And the method for pruning the network is called network reduction technique (Gedeon,1991) which including two main ideas which are removed constancy and similar hidden units.

After defined the neural network and pruning method, the experiment needs to read a dataset LPQ or PHOG dataset and then use K-Fold to split the dataset which could improve the generalization ability of the neural network. The training process is to use one database and one neural network but try to adapt to multiple different training sets which equivalent to an expanded database.

However, the accuracy is still keeping at a very lower level even if the neural network finished training processing. So, this research using another method K neighbor classifier in Machine learning to prove the neural network is well trained. Therefore, the researcher wishes to get a similar lower level accuracy.

Finally, to get a useful feedforward neural network which could classification different types of human emotion, this research does the last attempt which is convert 7 types of emotion into 4 types of emotion and then training and testing.

2. Method

2.1 Data set

- **Data outliers remove**

  Aim to remove outliers from datasets, firstly need a generalize understanding this datasets. Box plot (**Figure 1**) can help researchers notice if this variable column has outliers and how its distribution.

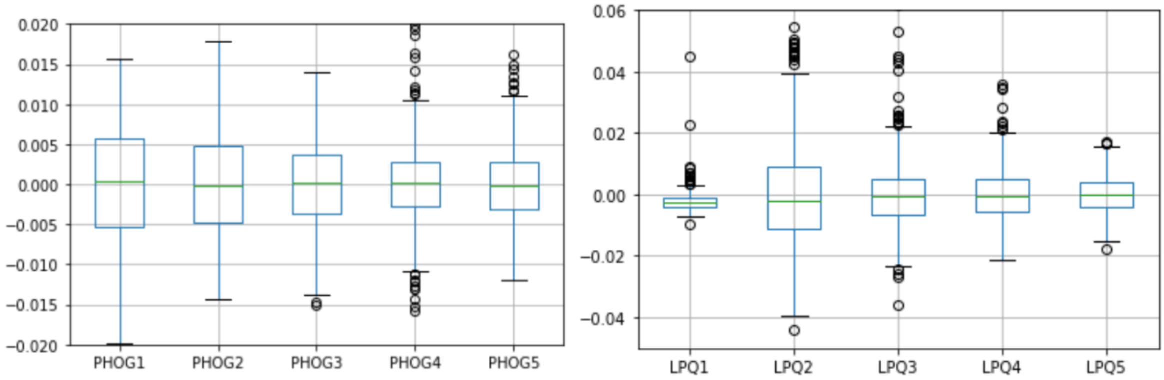


Figure 1: Box plot for 10 columns and circle represent outliers.

After ensures outliers exists in datasets, then research should compute a range (Q1-1.5*IQR, Q3+1.5*IQR). If a data point out of this range, the researcher need remove whole row from this dataset.

For every column

1. Get ***upper quartile*** value of this attribute

   Think as 75%

2. Get ***lower quartile*** value of this attribute

   Think as 25%

3. Compute ***IQR***

   IQR = upper_quartile – lower_quartile

4. ***Range***

   (Q1-1.5*IQR, Q3+1.5*IQR)

In this experiment we get result:

LPQ:

range for lower limit

[-0.0073043000000000005, -0.039428, -0.023511, -0.02113, -0.015000999999999999]

range for higher limit

[0.0031383, 0.03954, 0.022438, 0.020406999999999998, 0.015872999999999998]

PHOG:

range for lower limit

[-0.019926, -0.014275999999999999, -0.013817, -0.010799, -0.011857]

range for higher limit

[0.015622, 0.017791, 0.013952, 0.010498, 0.011144]

- **Data normalized**

  Normalization is the process of scaling individual samples to have unit norm. This process can be useful if you plan to use a quadratic form such as the dot-product or any other kernel to quantify the similarity of any pair of samples. This assumption is the base of the Vector Space Model often used in text classification and clustering contexts.

1. Import
   from sklearn import preprocessing
2. Normalized
   X = preprocessing.normalize(..., norm='l2') # …must be a array or list (DataFrame should convert df.values)

The function normalize provides a quick and easy way to perform this operation on a single array-like dataset, either using the l1 or l2 norms:

$$\text{L2:} \quad \|x\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}}$$

- **Data with K-Fold**
  1. Define a variable that would control the K-Fold model splitting data to **n** sets for the researcher easier to change it if the researcher wants to train the neural network with different numbers of the fold.
  2. Then create the KFold model and fit the X of datasets to the model.
  3. Then according to n to create 4 empty lists for the X training dataset, Y training dataset, X testing dataset, and Y testing dataset. For example: X_train_= [[] for x in range(n)]
     If **n** = 4:
     Out: X_train_ = [[], [], [], []]
  4. Get splitting results which are the index of datasets and then collect data according to the index list.
  5. Covert data type to torch.FloatTensor for X and torch.LongTensor for Y.
  6. Finally get n pairs of training X, Y, and testing x, y which are ready for training.

2.2 Neuron pruning method

The network reduction technique has two main ideas (Gedeon,1991) which are removed constancy and similar hidden units. Aim to minimize computation resource usage, therefore, this technique only removes units which function is similar to other units. In this research, we set all weights of hidden units to 1 as initial which means no hidden units are being removed now and set the weight to 0 to those units were pruned. This research uses three layers of Feedforward neural network but only pruning for the second hidden layer.

For **constancy** method define:

i. Input is the second layer output.
ii. Set a mask to correct shape. The length of the mask must equivalent to the numbers of hidden units.
iii. Every element in the second layer output could think as a vector, and we need to convert these vectors to its unit vector. $u = \frac{\vec{v}}{\|\vec{v}\|}$
iv. Iterate the second layer output and compute the dot production of current vector with all other vectors.

$$\overrightarrow{v_1} \cdot \vec{v}_2 = \|\overrightarrow{v_1}\| \times \|\overrightarrow{v_2}\| \times cos(\theta)$$

v. Get angle and represent it in degree form:

$$deg\,ree = arccos(\overrightarrow{v_1} \cdot \overrightarrow{v_2}) \times \frac{180^0}{\Pi}$$

vi. If degree < 0.1 then we think these two vectors are constancy.
vii. Keep the current vector and remove another hidden unit.

For **similar** method define:

If two weights absolute subtraction is less than 0.001, then we think two hidden neurons are similar and always keep the current vector and remove another hidden unit.

Here degree limited as 0.1 and the similarity number is 0.001, it can be changed if researchers want to study how their changes influence the accuracy.

2.3 Feedforward model design

Three layers of feedforward neural network need to define three linear functions and two activation functions which means the last layer in the neural network does not has an activation function. And this research selects nn.Hardtanh() as activation function type which type is proved that is the best suitable activation function type in ‘Static Facial Expression Analysis in Tough Conditions uses three layers feedforward network’ report (Xinyi, 2020). Then define a forward function in the net class for a forward pass and the result can use for both loss computation and pruning in training processing.

After the net class and pruning method are both being well defined we start training.

Training:

1. Create a feedforward neural network, an entropy type loss function, and an SGD type optimizer. These are used for the rest of the steps.

i. Iteration K-Fold training pair X training datasets and Y training datasets.

ii. In every iteration:
    1. Create an empty list for this pair of X and Y.
    2. Iteration the number of epoch times:
        a. Fit X in the same neural network and get predict Y.
        b. Use loss function to compute this epoch loss value.
        c. Loss list collects this epoch loss value.
        d. If epoch % 100 = = 0:

            Compute and print accuracy: number of correctness / total number of predict Y

        e. Back propagation and updating weights
        f. If epoch % 400 = = 0:
            Similar pruning for hidden out units in fc2
            Constancy pruning for hidden out units in fc2.
        g. Plot loss changes with the number of epochs.

iii. Done

2.4 Comparison & contrasting

This research is aim to use another method K neighbor classifier in Machine learning to prove the network is well trained. Therefore, we design a K-Fold training and prediction in the same .ipynb file with K-Fold training and testing in the neural network. From this file, research would get two classification methods training differences in every different pair of splitting datasets.

And also design a splitting training and prediction in another .ipynb file to get generalized results for the whole database and also easier to find the best suitable number of the neighbors got training model.

2.5 Type combine from 7 to 4

To get a useful feedforward neural network, this research attempt to convert 7 types of emotion into 4 types of emotion, and this research decides to keep Fear, Happy, Neutral three emotions classes, and other emotion classes Angry, Disgust, Sad, Surprise to another new class. Research has this decision because Fear, Happy, Neutral three emotions classes always have higher test accuracy in every K-Fold splitting dataset.

3. Results and Discussion

3.1 Data normalized vs Balance data

The last report which named ‘Static Facial Expression Analysis in Tough Conditions uses three layers feedforward network’ report (Xinyi, 2020) is using balance splitting data. This research is used normalize dataset and this part learning code is in ‘normalized data.ipynb’ file. The code did not change any method except using normalize datasets to training and testing neural networks.

From **Figure 2** we would see the loss value in left image convergence perfectly and training accuracy achieves 97.78%, however, the neural network testing accuracy is only 20.63%. This result shows the neural network is overfitting, but no idea how to stop training due to loss value convergence perfectly. In the last report researcher manual trial many times and get a fuzzy range of high accuracy limit value.

But the loss value in the right of the image has frequently shake since loss value over 1.7742 and training accuracy achieve

28.89%. Therefore, if training of neural network stop at loss value is 1.7742, the neural network will be not overfitting and the testing accuracy is 23.03% which improves 3%.

In conclusion, normalize would improve model performance and can help research prevent overfitting.

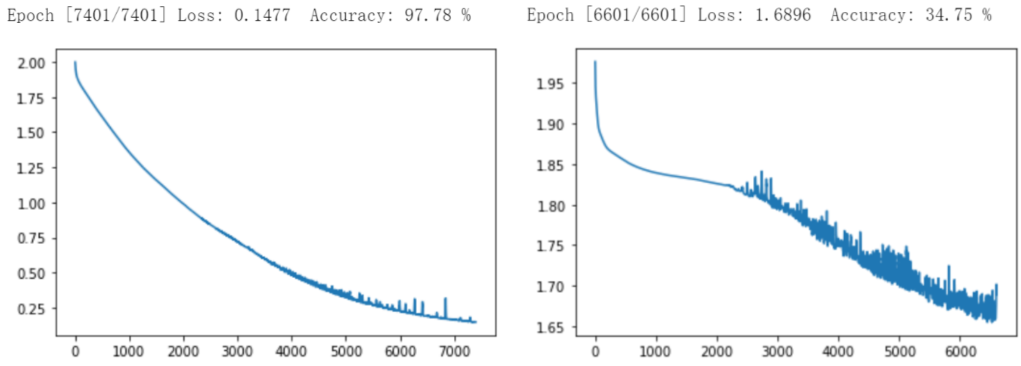


Figure 2

3.2 K-Fold vs once splitting

In this research 'K-Fold data in neural networks for LPQ datasets.ipynb' and 'K-Fold data in neural networks for PHOG datasets.ipynb' is used K-Fold to split database. And 'normalized data.ipynb' is using one time splitting the database.

And In this experiment, the result of **Figure3** shows that the K-Fold method does not need manual trial suitable iteration time. When research use split data, researchers may worry about higher iteration times may overfitting and lower iteration times the model may not training well. However, use the K-Fold method just need selects an approximately iteration times, if the first K-Fold pair of X, Y does not train enough time, the second pair and other pairs will training this network repeated and the loss value always convergence perfectly and does not appear loss value frequently shake.

That is because K-Fold training use one database and one neural network but tries to adapt to different training sets which equivalent to an expanded dataset and improve the generalization ability of the neural network.

Figure3 LPQ for n = 10

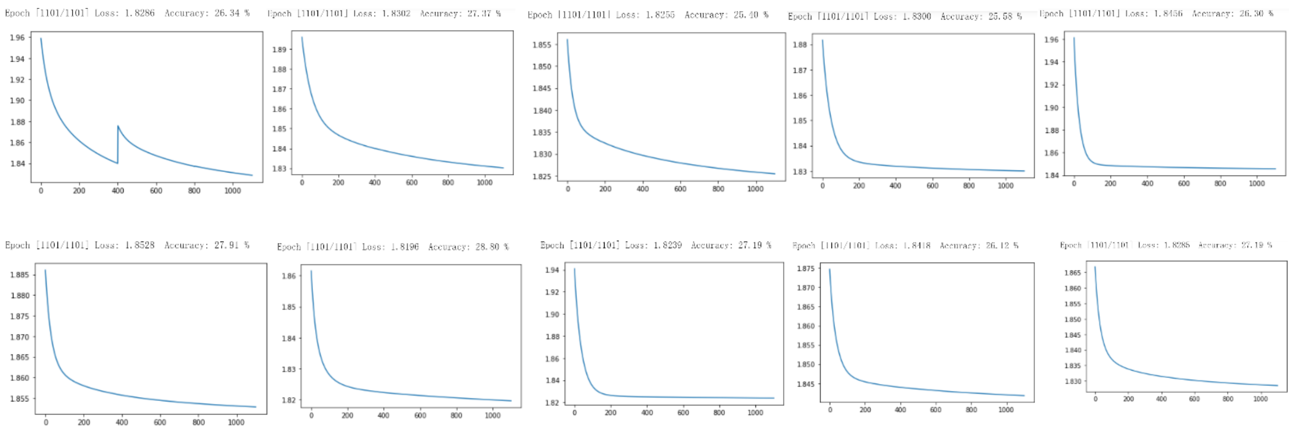
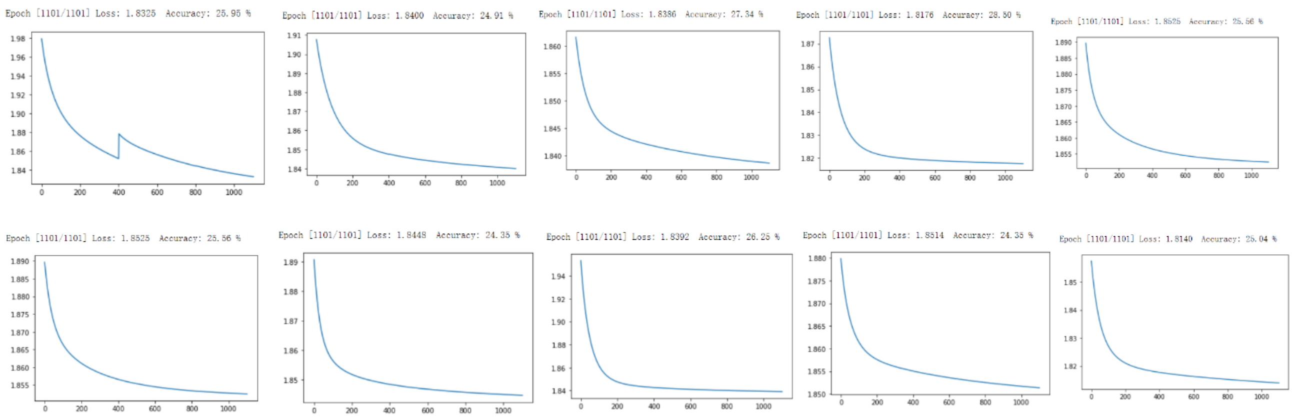


Figure3 PHOG for n = 10



3.3 K-Fold n splitting

Table 1 try to find the best number of the database will be split, however, the result shows the neural network is robustness. Although a new training process change number of database splitting part the testing accuracy are similar.

Table 1

Accuracy	K-Fold: n				
	6	7	8	9	10
LPQ	Training: 29.15 % Testing: 24.42%	Training: 28.89 % Testing: 24.74 %	Training: 27.94% Testing: 24.41%	Training: 27.17% Testing: 24.15%	Training: 27.19% Testing: 24.48%
PHOG	Training: 26.12% Testing: 23.02%	Training: 26.63% Testing: 22.83%	Training: 25.58% Testing: 22.42%	Training: 25.25% Testing: 22.58%	Training: 24.53% Testing: 22.57%

3.4 Neuron pruning experiment

The **Table2** result shows that these limit number does not influence the testing accuracy much unless pruning limit number is too large that pruning useful hidden unit. Therefore, we would enlarge two pruning limit numbers as much as possible to minimize computation resources usage.

Table 2

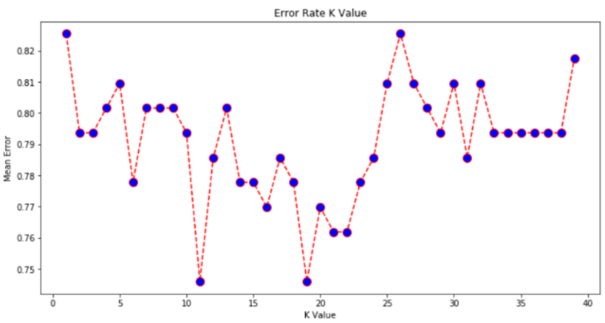
Try to find best constancy degree limit number and similar limit value:

	constancy degree limit number	similar limit value	Test Accuracy
LPQ	0.1 (degree)	0.0001	24.48%
	0.5	0.0005	24.64%
	1	0.001	24.80%
	1.5	0.0015	24.31%
	3	0.003	24.32%
	10	0.001	24.32%
	10	0.003	24.32%
	15	0.003	15.01%
PHOG	0.1 (degree)	0.0001	22. 88%
	0.5	0.0005	22.57%
	1	0.001	22.89%
	1.5	0.0015	22.57%
	3	0.003	23.36%
	10	0.003	22.26%
	15	0.003	15.47%

3.5 K neighbor classifier vs Feedforward neural network

K number of neighbors selected processing example is showing below **Figure4**. And the result for the KNN model, which is for LPQ datasets, selects using 19 number of neighbors to compute.

Figure4 LPQ



	precision	recall	f1-score	support
0. 0	0. 23	0. 37	0. 28	19
1. 0	0. 14	0. 08	0. 10	13
2. 0	0. 35	0. 37	0. 36	19
3. 0	0. 27	0. 42	0. 33	19
4. 0	0. 25	0. 16	0. 19	19
5. 0	0. 23	0. 17	0. 19	18
6. 0	0. 23	0. 16	0. 19	19
accuracy			0. 25	126
macro avg	0. 24	0. 25	0. 23	126
weighted avg	0. 25	0. 25	0. 24	126

For general model result: very close to each other

The neural network and KNN classifier model test accuracy both close to 25% which shows in 'KNN Clustering for LPQ datasets.ipynb' code file.

For every splitting datasets: two testing accuracy is not very close for every different splitting datasets. Which is showing in **Table3.** And same as KNN classifier model performance better because it has more average testing accuracy for every different splitting datasets and also has higher total accuracy.

Table3 LPQ

Testing for nth part	Feedforward neural network	KNN caasifier
1	22.22%	35%
2	19.35%	26%
3	20.97%	29%
4	46.77%	37%
5	56.45%	42%
6	38.71%	45%
7	14.52%	32%
8	17.74%	35%
9	6.45%	16%
10	0.00%	0%
Total	24.32%	29.7%

3.6 Converts 7 types of emotion into 4 types of emotion

For the LPQ method the test accuracy promotion from 24.32% to 55.16% (which accuracy gets from the neural network model). And for the PHOG method the test accuracy promotion from 22.42% to 56.63% (which accuracy gets from the neural network model). Which data shows on the bottom of 'K-Fold data in neural networks for LPQ datasets.ipynb' file and 'K-Fold data in neural networks for PHOG datasets.ipynb' file.

4. Conclusion and Future work

In conclusion, normalize would improve model performance and can help research prevent overfitting. K-Fold training use one database and one neural network but tries to adapt to different training sets which equivalent to an expanded dataset and improve the generalization ability of the neural network. To minimize computation resources usage researcher would enlarge two pruning limit number as much as possible as long as the test accuracy is not significantly lower than before pruning. Reducing the variety of classifications can significantly improve the accuracy of feedforward neural networks. In this research KNN classifier performance a bit better than neural networks. In future researcher would continuously working with improve test accuracy but focus on:

- The relationship between different normalize methods and test accuracy
- The relationship between different loss function in neural network and test accuracy
- The relationship between different classes combine to reduce the variety of classifications and test accuracy
- Data reduce dimension with test accuracy
- Recollect database

References

Ambadar, Z., & Schooler, J., & Cohn, J. (2002) Deciphering the enigmatic face: The importance of facial dynamics to interpreting subtle facial expressions. *Psychological Science*, 16(5):403–410, 2005.

Dhall, A., & Goecke, R., & Lucey, S., & Gedeon, T.D. (2011). Acted Facial Expressions in the Wild Database. *In Technical Report*.

Dhall, A., & Goecke, R., & Lucey, S., & Gedeon, T. (2011). Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. Proceedings of the *IEEE International Conference on Computer Vision*(2106-2112. 10.1109/ICCVW.2011.6130508).

Gedeon, T. D., & Harris, D. (1991). Network reduction techniques. In *Proceedings International Conference on Neural Networks Methodologies and Applications* (Vol. 1, pp. 119-126).

Xinyi, H. (2020). Static Facial Expression Analysis in Tough Conditions use three layers feedforward network. In *3RD Anu Annual Bio-inspired Computing Student Conference Report.*