# The Comparation Between Weight Magnitude and Evolutionary Algorithms in Unique and Significant Input Recognition

Xiyong Tian[1]
[1] School of Computing, Australian National University
u6782003@anu.edu.au

**Abstract.** In this article I use the Static Facial Expression in the Wild(SFEW) dataset to classify different emotions based on different facial expressions. The first part of the article introduces the pre-processing method to deal with the dataset and build a multilayer neural network. The second part introduces a magnitude measures and find one method to determine which inputs are unique and significant. The third part of this article introduces the Evolutionary Algorithms to find the best combination of significant attributes. The results illustrate that the magnitude measures in weight can figure out which is the most significant attribute, though the robust perform weakly. Though the Evolutionary Algorithm can tell the best combination of significant attitude, it is hard to know the most significant one or the least significant one.

**Keywords:** SFEW, multi-layer neural network, magnitude measures,

## 1 Introduction

Facial expression is wildly used in emotion recognition. In our daily life, emotions can be predicted by the facial expression and it is common for us to change our behavior based on other's feeling [1]. Sometimes, the computers could catch more information from the facial expression than humans. Therefore, more details which are not easy for human to pay attention can be caught by computers[2]. The facial expression are wildly used in criminal investigation, medical research, sociology and so on. Thus, in this article, I will use the Static Facial Expression in the Wild(SFEW) dataset [3] to do the emotion recognition.

Static Facial Expression in the Wild(SFEW) dataset is a emotion recognition dataset [3]. It is a sub-dataset extracted from Acted Facial Expression in the Wild(AFEW) dataset which contains 987 video clips [4]. The SFEW dataset contains 700 images and seven labels of the emotions, which are angry, sad, happy, disgust, the neural, supervising, and fear. The dataset extracts 10 principal components. Five of them are Local Phase Quantization (LPQ) features and Five of others are Pyramid of Histogram of Gradients (PHOG) features. Meanwhile, the LPQ features are calculated by:

$$d = \sum_{i=1}^{k} 2^{i-1} I(P, N_i) \tag{1.1}$$

$$\text{Where } I(p,\ N_i) = \begin{cases} 1 & if\ c < N_i \\ 0 & otherwise \end{cases}$$

Moreover, the accuracy of the LPQ is 43.71% as well as the PHOG is 46.28% [3].

In this article, I will use two different methods, Weight Magnitude and Evolutionary Algorithms, to recognize the unique and significant input of this dataset and discuss the different between these two methods.

## 2 Methodology

### 2.1 Data Pre-processing

The data was stored in csv file and I picked the first line and the first column of the dataset to be the header of the table. Then transfer the first line of the labels from 1 to 10 to represent the principle component of LPQ and PHOG. I transferred second column of the dataset from 0 to 6, which represents different emotion label angry, sad, happy, disgust, the neural, supervising, and fear to fit the 7 output neurons.

$$\frac{x-\bar{x}}{\sigma} \tag{2.1}$$

In order to avoid gradient vanishing and gradient explosion, the formular in 2.1 are used to normalize the dataset between one and minus one.

### 2.2 Model Building

In this article, we use 2 different methods, Weight Magnitude and Evolutionary Algorithms, to find the significant input. The first method needs to use weight of the neural network and the second method needs to use neural network to find the best chromosome. Thus, the first thing in this article is to build a neural network model. Here we use multilayer neural network, which contains 10 inputs, 30 hidden neurons, and 7 output neurons. I used 10 inputs to correspond the 10 features and use 7 outputs to represent 7 different emotions, angry, sad, happy, disgust, the neural, supervising, and fear. The learning rate of the model is 0.01 and to decrease the epoch of the model, I used the minibatch method to decrease the epoch from 10000 to 500. In the forward pass, SoftMax activation function is used to estimate the non-linear model after the hidden layers.

### 2.2 Magnitude Measures

In order to identify the significance of the input, I use the weight magnitude measures, which mentions by Gedeon[6]. In this paper, the author calculate the contribution of the input and hidden layer and also calculate the contribution of the hidden neuron and output neuron. The calculation of the input neuron and the hidden neuron is

$$P_{ir} = \frac{|w_{ir}|}{\Sigma_{p=1}^{ni}|w_{pr}|} \tag{2.1}$$

Meanwhile, the calculation of the output neuron and the hidden neuron is

$$P_{rk} = \frac{|w_{rk}|}{\Sigma_{p=1}^{nh}|w_{pk}|} \tag{2.2}$$

Therefore, we combine this together and we can get the contribution of input neuron and the output neuron is

$$Q_{ik} = \Sigma_{r=1}^{nh}(P_{ir} \times P_{rk}) \tag{2.3}$$

In this first formula, $W_{ij}$ represents the weight between i the input and j hidden, the sum of $W_{pr}$ is the sum of all weight between input neuron and the hidden neuron. Meanwhile, $W_{rk}$ represents the weight between r hidden and k output neuron, the sum of $W_{rk}$ is the sum of all weight between the hidden neuron and the output neuron. Therefore, in this method, we don't need to consider the sign of the contribution, which means the weight of Q is related to the magnitude of the contribution. And we can use such a method to identify the importance of the input.

### 2.3 Evolutionary Algorithms

Boubenna and Lee mention that Evolutionary Algorithms as a bionic algorithm, is a simulation of changing chromosomes during evaluation[7]. It contains select, crossover, and mutate during the evaluation. In the Evolutionary Algorithms, first we select each part of the chromosomes from parents

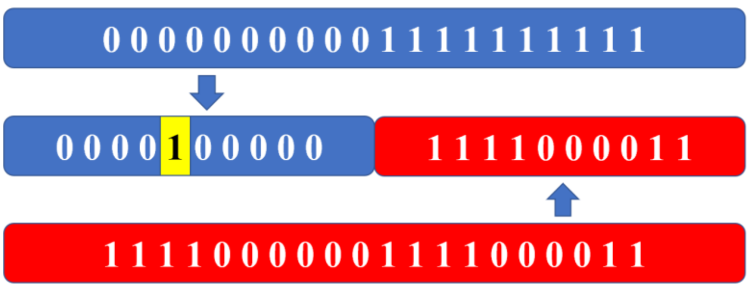


**Fig. 1.** The process of Evolutionary Algorithm. The top of the chromosome and the bottom of the chromosome are the parents' chromosomes and the middle one is child chromosome.

The fig 1 illustrates the processes of the Evolutionary Algorithm. Firstly we pick two chromosomes from our parents. Then we select each part of the chromosome and combine them together, which is called crossover. Meanwhile, the chromosome may have mutation during the crossover. Finally we get a new chromosome, which belongs to our child.

## 3 Results

### 3.1 Training Accuracy and Test Accuracy

In the 2 layer neural network layer, the training accuracy after 500 epochs is

| Times | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Training accuracy | 90.79 % | 92.08 % | 96.27 % | 94.12 % | 94.43 % |

**table. 1.** The training accuracy after 500 epochs, data randomly choose for 5 times

Meanwhile, the testing accuracy after 500 epochs

| Times | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Test accuracy | 19.85 % | 22.22 % | 20.29 % | 20.00 % | 22.73 % |

**table. 2.** The test accuracy after 500 epochs, data randomly choose for 5 times

Table 1 and table 2 illustrate the training accuracy and testing accuracy after training 500 epochs on multilayer neural network. The average of the training accuracy is 93.54%, and the average test accuracy is 21.02%.

Without the pre-processing, the training accuracy after 500 epochs is

| Times | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Training accuracy | 93.88 % | 94.62 % | 92.92 % | 91.62 % | 88.09 % |

**table. 3.** The training accuracy after 500 epochs, data without pre-processing randomly choose for 5 times

Meanwhile, the testing accuracy after 500 epochs

| Times | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Test accuracy | 19.21 % | 22.96 % | 13.82 % | 27.01 % | 23.33 % |

**table. 4.** The test accuracy after 500 epochs, data without pre-processing randomly choose for 5 times

Table 3 and table 4 illustrate the training accuracy and testing accuracy after training 500 epochs on multilayer neural network without pre-processing. the average of the training accuracy is 92.23%, and the average test accuracy is 21.27%. The accuracy illustrates stability under pre-processing.

### 3.2 Contribution of Inputs

After doing the weight magnitude measures, the contribution of 10 inputs are demonstrated as table3.

| | top1 | top2 | top3 | top4 | top5 | top6 | top7 | top8 | top9 | top10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | 7 | 5 | 8 | 0 | 4 | 9 | 2 | 3 | 1 |
| 2 | 9 | 8 | 0 | 6 | 7 | 4 | 3 | 2 | 5 | 1 |
| 3 | 9 | 5 | 7 | 6 | 8 | 3 | 4 | 0 | 2 | 1 |
| 4 | 7 | 6 | 8 | 9 | 0 | 4 | 2 | 3 | 5 | 1 |
| 5 | 9 | 7 | 8 | 6 | 4 | 3 | 2 | 0 | 5 | 1 |
| 6 | 9 | 8 | 5 | 6 | 7 | 0 | 2 | 4 | 3 | 1 |
| 7 | 7 | 8 | 9 | 6 | 5 | 4 | 0 | 2 | 3 | 1 |
| 8 | 9 | 7 | 8 | 3 | 5 | 4 | 2 | 6 | 0 | 1 |
| 9 | 9 | 5 | 7 | 8 | 6 | 3 | 0 | 4 | 2 | 1 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 9 | 0 | 4 | 3 | 6 | 8 | 7 | 5 | 1 | 2 |

**table. 4.** the contribution of 10 inputs in 10 times, top1 means the most significant and the top10 means the least significant.

Table 4 shows the contribution of 10 inputs in 10 epochs, top1 means the most significant and the top10 means the least significant. From the above table, the 5 most significant inputs and the 5 least significant inputs shows like below

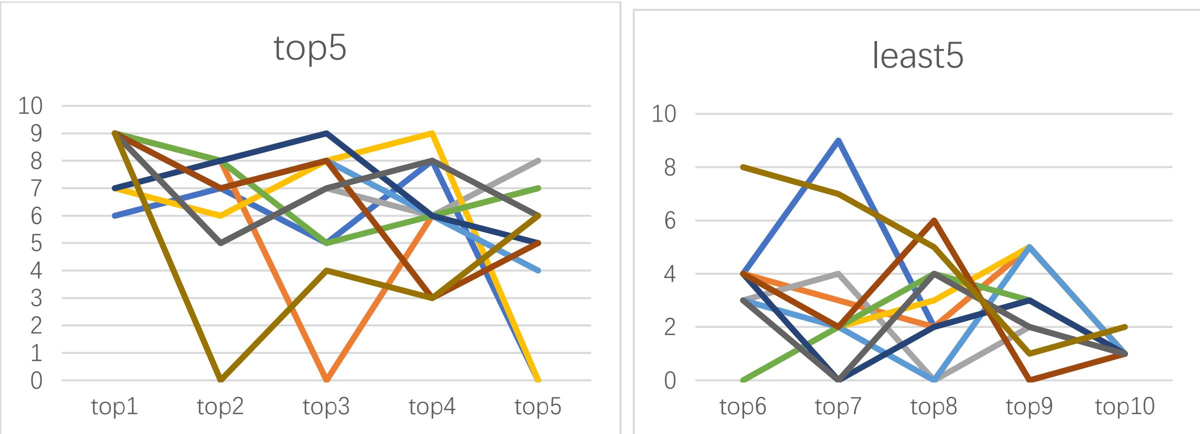


**chart. 1.** the contribution of 10 inputs in 10 times, top1 means the most significant and the top10 means the least significant.

According to the chart, we can see though the model is not robust, the top1 and the top10 significant input is clear.

### 3.3 Best pop accuracy

| Times | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Test accuracy | 20.53 % | 20.00 % | 20.29 % | 23.21 % | 21.29 % |
| Best pop accuracy | 25.16 % | 25.83 % | 28.26 % | 27.38 % | 28.38 % |
| Best pop | [0 0 0 1 0 1 1 1 1 1] | [0 0 0 0 0 0 0 1 1 1] | [0 1 1 0 1 1 1 1 1 0] | [0 1 0 1 1 0 0 1 1 0] | [1 1 0 1 1 1 1 1 0 1] |

**table. 5.** the accuracy after pick several significant attributes.

Table 5 demonstrate the accuracy after the Evolutionary Algorithm. We can get the best combination of the attributions after the evolution. Then we choose these attributions to train the neural network model and get the test accuracy. The average test accuracy after the evolution is 27.00%, which has a little bit increase comparing with using all dataset.

## 4 Discussion

According to the results, magnitude measures can measure which attribute has the most contribution in the picture. Though the model has weak robust, the result still tells that the contribution of attribute 6 to 10 is larger than the contribution of attribute 1 to 5. The result also tells that the $10^{th}$ attribute is a significant attribute because if we focus on the second attribute, we can see that it occupies most of the time in the top one contribution. This also illustrates in the Evolutionary Algorithm, when we see the best combination of the attributes, it always contains the $10^{th}$ attribute. That means the magnitude measure method works in this situation, though the robust perform weakly.

For Evolutionary Algorithm, we can see the best combination of the attribute. The advantage of the Evolutionary Algorithm is that we can tell the significant combination of attributes and we can see that the accuracy is increasing comparing with using all data. However, we can not tell the most significant attribute or the least significant attribute in these 10 attributes.

## 4 Conclusion and Future Work

In this paper, to compare with which method is better on input recognition, we use magnitude measures on weight method and Evolutionary Algorithm. I used pre-processing to deal with the data and use multilayer neural network to train the model getting the weight between inputs and hidden neurons, and hidden neurons and outputs.

The results illustrate that the magnitude measures in weight can figure out which is the most significant attribute, though the robust perform weakly. Though the Evolutionary Algorithm can tell the best combination of significant attitude, it is hard to know the most significant one or the least significant one. Therefore, in the future more methodology should be tried such as functional measures or sensitivity analysis to figure out how to find the unique and significant attribute.

## References

1. A. Samara *et al*, "Affective state detection via facial expression analysis within a human–computer interaction context," Journal of *Ambient Intelligence and Humanized Computing*, vol. 10, (6), pp. 2175-2184, 2019.
2. J. Yu, L. Li and J. Zou, "Realistic emotion visualization by combining facial animation and hairstyle synthesis," *Multimedia Tools and Applications*, vol. 76, (13), pp. 14905-14919, 2017.
3. A. Dhall *et al*, "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," in 2011, . DOI: 10.1109/ICCVW.2011.6130508.
4. A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. "Acted Facial Expressions in the Wild Database". (2011). Distributed by Australian National University. https://cs.anu.edu.au/few/AFEW.html.
5. A. Bosch, A. Zisserman and X. Munoz, "Representing shape with a spatial pyramid kernel," in 2007, . DOI: 10.1145/1282280.1282340.
6. T. D. Gedeon. "Data mining of inputs: analysing magnitude and functional measures", *Int J Neural Syst*. In 1997;8(2):209-218. DOI:10.1142/s0129065797000227
7. Boubenna and D. Lee, "Image-based emotion recognition using evolutionary algorithms," Biologically Inspired Cognitive Architectures, vol. 24, pp. 70-76, 2018.