

Facial Expression Recognition using Neural Network with Regularized Back-propagation Algorithm

Ashish Kumar Dogra

Research Scholar
Department of ECE,
Lovely Professional University,
Phagwara, India

Nikesh Bajaj

Assistant Professor
Department of ECE,
Lovely Professional University,
Phagwara, India

Harish Kumar Dogra

Research Scholar
Department of ECE,
Gyan Ganga Institute of
Technology & Sciences,
Jabalpur, India

Abstract

Since decades, in the field of face expression recognition, many researchers have been developing numerous new techniques. These developments are being fueled by numerous advances in computer vision. Such advancement in the field of computer vision holds a promise of reducing error rate in face expression recognition system. This paper proposes an automatic facial expression recognition system using neural network with regularized back-propagation algorithm. The Cohn-Kanade database [5],[8] have been used having six different types of expression. The database is highly imbalance so face detector using Viola Jones [2] have been used to crop and balance the database. Once balanced, a neural network approach used, obtained training set accuracy 99.32% and testing accuracy 91.9595%. The paper has got very good result of individual expression like happy and surprise expression, testing accuracy up to 100%. The reason why neural network is used, it might someday be able to learn in a manner similar to humankind.

Index Terms

Back-propagation algorithm, Machine learning and Neural Network.

1. INTRODUCTION

In the past few years, facial expression recognition system has engrossed a significant interest in the facial analysis community, as it plays an important role in Human-computer intelligent interaction (HCII), there is a need for the computer to naturally interact with the user, as human interaction takes place. Many more applications such as video conferencing, virtual reality, and user profiling and customer satisfaction studies for broadcast and web services, require efficient facial expression recognition in order to achieve the desired results. Therefore impact of the FER (facial expression recognition system) on the above mentioned application areas are constantly growing. Traditionally computer interface ignores user affective states, which lead to large loss of important information for the interaction process. To recognize the user affective state, human-centered interfaces must interpret non-verbal behavior such as body gesture, voice and facial movements. Facial expression is the most natural way for human beings to communicate emotions. Since decades, several efforts had been made by researchers in the field of facial expression

recognition. Ekman [1] has defined six basic universal emotions (Anger,

disgust, fear, happy, sadness, surprise). Moreover these six emotions can be form complex emotion by combining them. Every emotion has different types of facial movements that facial movements make them different from each other. These facial movements are called facial action units (AU's) and the corresponding code is known as Facial Action Coding system [3].

This paper is mainly focuses on expression detection on the Cohn-Kanade database. In Cohn-Kanade database, people are well posed and emotion is natural. This is the important step towards real world condition. That is why we propose machine learning concept using neural network with regularized back-propagation algorithm in an original framework.

1.1 Challenge Context.

In the literature, there are so many system has been proposed by researchers but all of them lack a common evaluation protocol. The FERA challenge, organized in conjunction with the IEEE Conference 2011 on Face and Gesture recognition [4].

The FERA challenge had been divided into two sub-challenges: 1) AU detection, 2) Emotion detection

We did research on emotion detection sub-challenge on Cohn-Kanade. The learning algorithm used on this challenge is directly deal with pixels that act as features. The only thing needed is making a system and train it in such a manner so that the approach used will provide the expected classification rates.

1.2 Overview of the proposed method

There are so many existing approaches have been proposed in the literature like geometric based methods, appearance-based methods and combination of both appearance and geometric methods [3]. Geometric based methods try to extract the facial shape information and the location of salient points. Appearance based methods extract facial texture information like wrinkles, furrows and bulges. These methods rely on Gabor wavelet.

To perform emotion detection, select neural network for their ability to find optimal hyper plane for non linear classification problems. The system has to trained in such a manner so that the

system is able to classify expression easily. Neural Network is actually a pretty old idea, which has fallen out of favor for a while. But today it is state of art technique for many different machine algorithms. The reason why we use neural network, it might someday be able to learn in a manner similar to humankind. In this paper, rigorous experiments have been performed on Cohn-Kanade database having each expression has its own different label.

2. EXPERIMENTAL APPROACH

Experimental approach explains the methods used for the accomplishment of the recognition of facial expression.

2.1 Visualization of Dataset

Neural network is the most powerful learning algorithm that was originally motivated by the goal of having machine that can mimic the brain. In our case, 444 training examples have been used, where each training example is a 50 pixel by 50 pixel grayscale image of the face expressions. Each pixel is represented by a floating point number indicating the grayscale intensity at that location. The 50 by 50 grid of pixels is unrolled into a 2500-dimensional vector. Each of these training examples becomes a single row in our data matrix say X. This gives us a 444 by 2500 matrix X where every row is a training example for a face expression image.

$$X = \begin{bmatrix} \text{--transpose of } x(1)\text{--} \\ \text{--transpose of } x(2)\text{--} \\ \text{--transpose of } x(3)\text{--} \\ \vdots \\ \text{--transpose of } x(n)\text{--} \end{bmatrix}$$

The second part of the training set is 444-dimensional vector y that contains labels for the training set from label 1 to 6

$$Y = \begin{bmatrix} -1- \\ -1- \\ \cdot \\ \cdot \\ -6- \\ -6- \end{bmatrix}$$

For our case, we randomly select the first 100 face expression from the datasheet is shown in Figure 1.

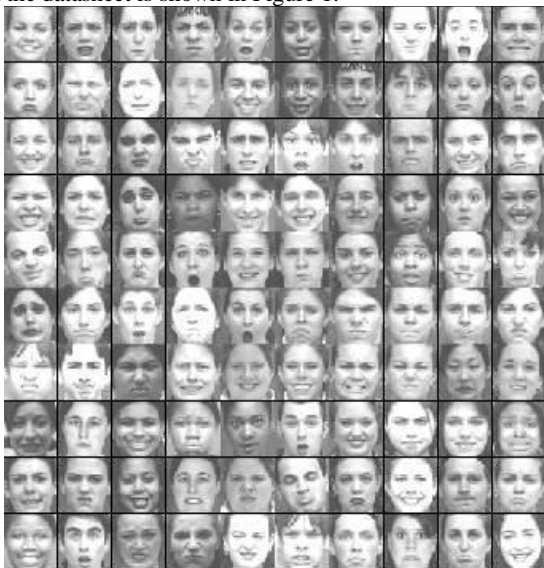


Figure 1: Randomly selected data from the datasheet.

2.2 Neural network model representation and apply onto dataset.

The neural network model is shown in Figure 2. It has three layers- an input layer, a hidden layer and an output layer. Since we have 50 by 50 pixels, this gives us 2500 input layer units (not containing the extra bias unit which always outputs +1). Every node in the hidden layer computes the features from the input layer.

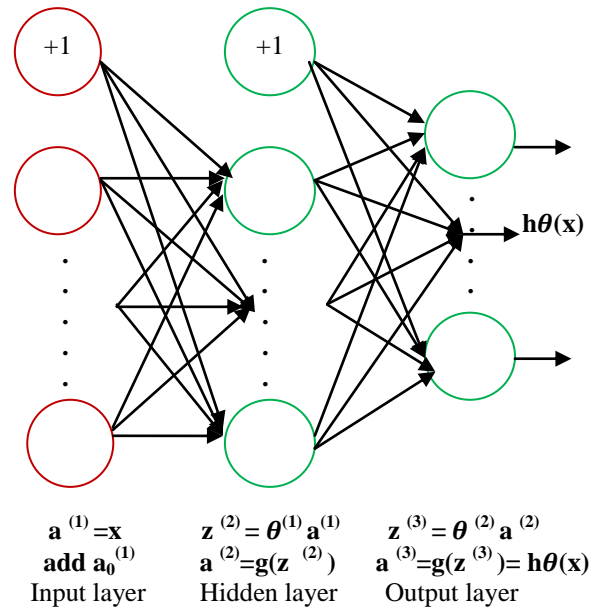


Figure 2: Neural network model representation.

2.3 Regularized cost function

The cost function for neural network with regularization is given by

$$J(\theta) = \sum_{i=1}^m \sum_{k=1}^K [-y_k^{(i)} \log((h\theta(x^{(i)}))_k) - (1-y_k^{(i)}) \log((h\theta(x^{(i)}))_k)] + \frac{\lambda}{2m} [\sum_{j=1}^{200} \sum_{k=1}^{2500} (\theta_{j,k}^{(1)})^2 + \sum_{j=1}^6 \sum_{k=1}^{200} (\theta_{j,k}^{(2)})^2]$$

where $h\theta(x^{(i)})$ is computed as shown in Figure 2 and for our case the value of K is 6 is the total number of possible labels. Note that the $(h\theta(x^{(i)}))_k = a_k(3)$ is the activation value of the kth output unit. Note we should chose λ value carefully, for our case we use 220 value of . Recall we have original labels in the variable y were 1, 2,...,6, for the purpose of training a neural network, we need to recode the labels as vectors containing only value 0 or 1, so that

$$y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Angry Disgust Fear Happy Sadness Surprise

2.4 Back-propagation algorithm.

Back-propagation algorithm is a common method of training artificial neural network. It is a supervised learning method, and generalization of the delta rule (for more details, we invite readers to go through). It requires a dataset of the desired output for many inputs, making up the training set. The back-propagation is shown

in Figure 3.

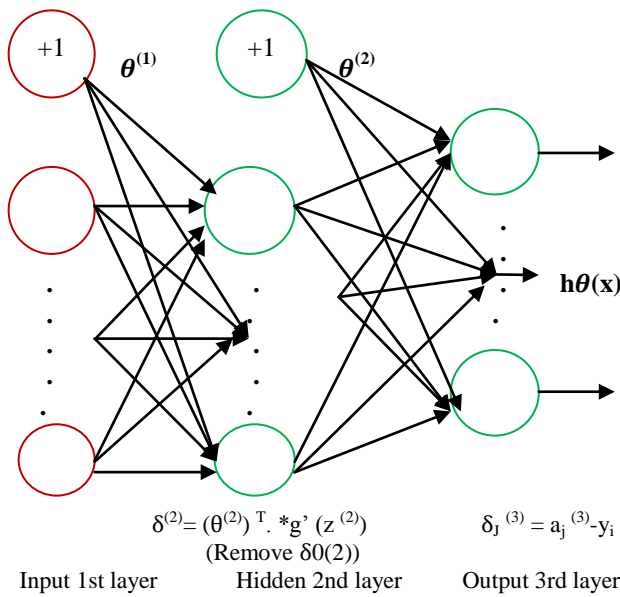


Figure 3: Back propagation updates.

Now implement the back-propagation. Given a training examples $(x^{(i)}, y^{(i)})$, run a “forward pass” to compute all the activations throughout the network, including the output value of the hypothesis $h\theta(x)$. Then, for each node j in layer 1, compute an error term $\delta_j^{(1)}$ that measures how much that node was responsible for any errors in our output.

For an output node, directly measure the difference between the network’s activation and the true target value, and use that to define $\delta_j^{(3)}$ since layer 3 is the output layer. For hidden units, we will compute $\delta_j^{(l)}$ based on a weighted average of the error terms of the nodes in layer $(l+1)$.

In detail Back-propagation algorithm as follows:

1) Set the input layer’s values $(a^{(l)})$ to the t th training examples $x^{(l)}$. Perform feed-forward propagation, computing the activations $(z^{(2)}, a^{(2)}, z^{(3)}, a^{(3)})$ for layer 2 and 3. Remember that you need to add a +1 term to ensure that the vector activations for layers $a^{(1)}$ and $a^{(2)}$ also include the bias unit.

2) For each output unit k in output layer 3, set

$$\delta z^{(3)} = (a_k^{(3)} - y_k)$$

Where $y_k \in \{0, 1\}$ indicating whether the current example belongs to class k ($y_k=1$), or if it is belong to a different class ($y_k=0$).

3) For hidden layer $l=2$, set

$$\delta_k^{(2)} = (\theta^{(2)})^T \cdot \delta_k^{(3)} \cdot g'(z^{(2)})$$

4) Accumulate the gradient from this example using the following formula. Remember that we should remove $\delta_0^{(2)}$.

$$\Delta^{(l)} = \Delta^{(l)} + \delta^{(l+1)} (a^{(l)})^T$$

5) Obtain the regularized gradient for the neural network cost function by dividing the accumulated gradients by $1/m$:

$$\partial / \partial \theta_{ij}^{(l)} J(\theta) = D_{ij}^{(l)} = \frac{1}{m} \Delta^{(l)} + \frac{1}{m} \theta_{ij}^{(l)}$$

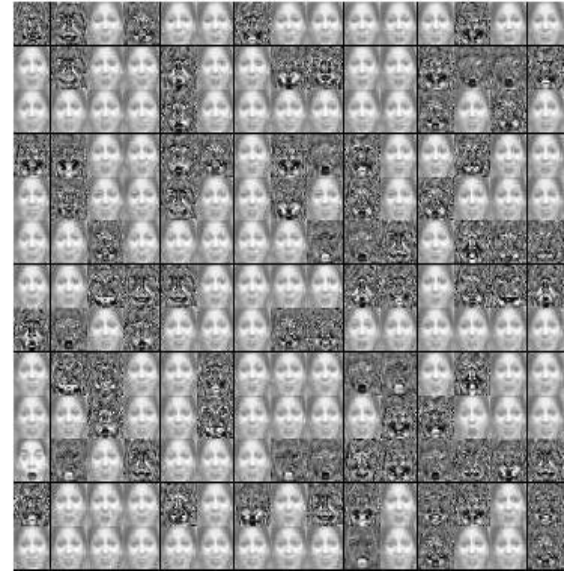


Figure 4: Visualization of learned feature by neural network.

- Further it can be visualize what neural network is learning by displaying the hidden units to see what features they are capturing in the data. The features are shown in Figure 4.
- The learning curve during training the system is shown in Figure 5 and it shows how our system learns as increase the number of iteration increases, the cost function get decreases until it local optima is achieved.

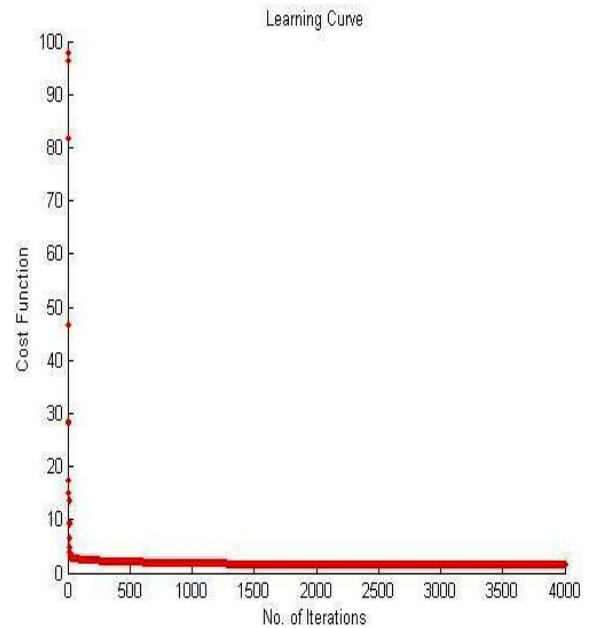


Figure 5: Learning curve (cost function versus Number of iterations.)

3. EMOTION RECOGNITION RESULTS

In this section, evaluation of our original framework for emotion detection. The objective of expression detection task is to label each sequence with one of the six expression classes: Anger, Disgust, Fear, Happy, Sadness and Surprise.

Classification rates of our framework for the case of PS (person specific) are shown in Table 3. The overall training and testing accuracy is shown in Table 1.

Table 1: Training and testing accuracy

| | |
|--------------|------------------|
| Training Set | Testing Accuracy |
| 99.3237% | 91.9595% |

Table 3: Our framework classification rates for six discrete emotions

| Emotions (PS) | Classification rates |
|---------------|----------------------|
| Anger | 69.230769% |
| Disgust | 93.333333% |
| Fear | 88.888889% |
| Happy | 100.00000% |
| Sadness | 91.666667% |
| Surprise | 100.00000% |
| Overall | 91.959500% |

4. CONCLUSION

The proposed original framework using advanced machine learning algorithm to perform emotion detection is discussed throughout. Experimental setups show that classification rates for emotion recognition using neural network. The used approached

have got very good result as we can see in Table 1, Table 2. However, the overall results remain insufficient, expression like angry are detected with less accuracy. So there is still room for improvement. We believe that experimental results could be significantly increased by using neural network.

5. REFERENCES

- [1] J. Russell and J. Dols, the Psychology of Facial Expression. Cambridge, U.K.: Cambridge Univ. Press, 1997, ser. Studies in Emotion
- [2] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.
- [3] Z. Zeng, M. Pantic, G. Roisman, and T. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 39–58, Jan. 2009.
- [4] M. Valstar, B. Jiang, M. Méhu, M. Pantic, and K. Scherer, "The first facial expression recognition and analysis challenge," in Proc. IEEE Int. Conf. FG, 2011, pp. 921–926.
- [5] Y. Tian, T. Kanade, and J. Cohn, "Facial expression analysis," in Handbook of Face Recognition. New York: Springer-Verlag, 2005, pp. 247–276.
- [6] Kanade, T., Cohn, J., & Tian, Y. (2000). "Comprehensive database for facial expression analysis, Proceedings of the fourth IEEE International Conference on Automatic Face and Gesture Recognition(FG'00), Grenoble, France, 46-53.