Machine Learning-CS GY 6033

Detection of Human Emotion using Artificial Neural Networks and Particle Swarm Optimization

Rachna Ramkumar (RR3656)

NEW YORK UNIVERSITY

Abstract

Human Emotions are often complex and most often go unnoticed but carry so much value. The idea is to use Machine Learning in order to detect human emotions in order to make use of the information.

- This paper explores ANNs and particle swarm optimization at their maximum capacity by implementing the use of CK+ dataset.

- The reason to apply the PSO algorithm is because it involves the optimization of the cost function of a cogeneration system also satisfactory results were obtained demonstrating the efficiency of the PSO method.

Introduction

Emotions display more than just feelings; they show the actual intentions of the particular person. These intentions can be used for a wide array of applications. According to different surveys, verbal components convey one-third of human communication, and nonverbal components convey two-thirds. Among several nonverbal components, by carrying emotional meaning, facial expressions are one of the main information channels in interpersonal communication.

Dataset

Since the Cohn-Kanade (CK) database was released for the purpose of promoting research into automatically detecting individual facial expressions, it has become one of the most widely used testbeds for algorithm Training and Validation. Even though it was extremely useful, it posed three limitations that become apparent over time:

1) While AU codes are well validated, emotion labels are not, as they refer to what was requested rather than what was actually performed,

2) The lack of a common performance metric against which to evaluate new algorithms, and

3) Standard protocols for common databases have not emerged.

As a consequence, the CK database has been used for both AU and emotion detection, comparison with benchmark algorithms is missing, and use of random subsets of the original database makes meta-analyses difficult.

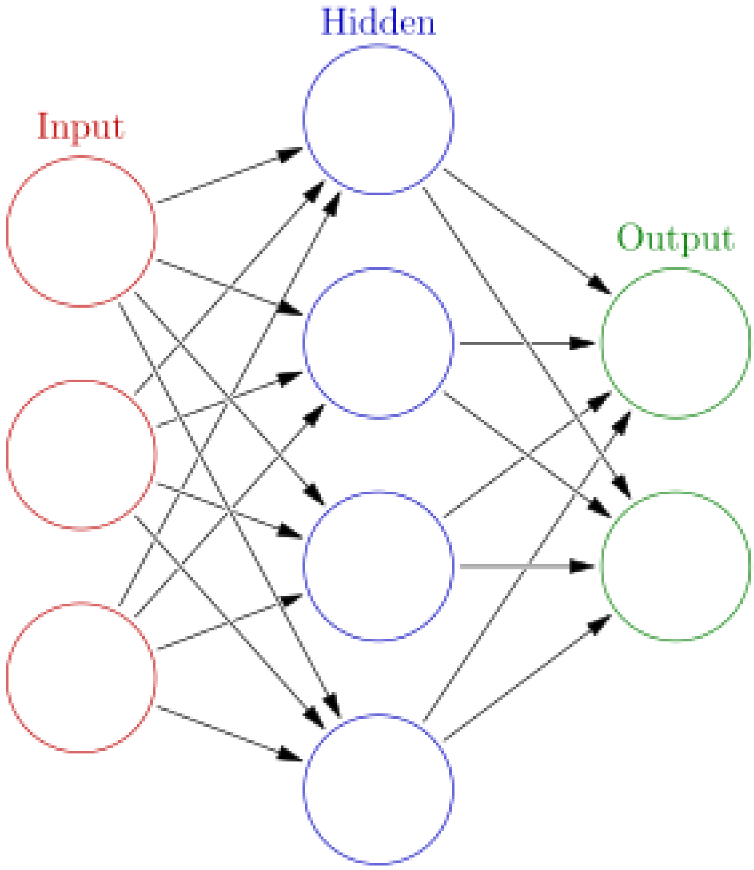
To address these and other concerns the Extended Cohn-Kanade (CK+) database was introduced. Extended Cohn-Kanade (CK+) includes both posed and non-posed (spontaneous) expressions and additional types of metadata. For posed expressions, the number of sequences is increased from the initial release by 22% and the number of subjects by 27%.

Algorithms

Artificial Neural Network

An Artificial Neural Network is an information processing model that is inspired by the way biological nervous systems, such as the brain, process information. They are loosely modelled after the neuronal structure of the mammalian cerebral cortex but on much smaller scales.

In simpler terms it is a simple mathematical model of the brain which is used to process nonlinear relationships between inputs and outputs in parallel like a human brain does every second.



Neural networks learn things in exactly the same way as the brain, typically by a feedback process called back-propagation.
Over time back-propagation causes the network to learn by making the gap between the output and the intended output smaller to the point where the two exactly match, so the neural network learns the correct output.

Whereas in this paper, Particle Swarm Optimisation has been used in the place of Back Prorogation.

Particle Swarm Optimisation

The particle swarm optimization (PSO) algorithm, proposed by Kennedy and Eberhart , is a metaheuristic algorithm based on the concept of swarm intelligence capable of solving complex mathematics problems existing in engineering .

It is of great importance noting that dealing with PSO has some advantages when compared with other optimization algorithms, once it has fewer parameters to adjust, and the ones that must be set are widely discussed in the literature .

Ghandi et al., presents an approach that applies a modified version of the Particle Swarm Optimization (PSO) algorithm, which is called Guided Particle Swarm Optimization (GPSO). It tracks down the movements of facial action units (AUs) that are placed on the face of a subject and captured in video clips. The algorithm is implemented and tested on video clips that contain all the six basic emotions, namely happy, sad, surprise, disgust, anger and fear. The success rate averaged at around 85.3%.

Project Concept

Face Extraction

The proposed system works on extracting the facial features from facial images and seven features are calculated from those facial features such as mouth curve, length of mouth, height of mouth, length of left and right eyebrows and length of eye and height of eye.

The generated Values from the CK+ Dataset were stored as a data frame and then sent to a csv file. Each emotion had its own spreadsheet and finally all of them were merged together. This was done twice, once for the Training Dataset and the second for the validation dataset.

	mouth_curve	mouth_length	mouth_height	eyebrow_l	eyebrow_r	eye_length	eye_height	label
0	-0.100000	9.0	-10.0	0.153846	-0.666667	4.0	-6.0	7.0
1	-0.111111	13.0	-11.0	0.050000	-1.000000	4.0	-2.0	7.0
2	-0.333333	13.0	-17.0	-0.045455	-0.875000	4.0	-3.0	7.0
3	-0.333333	10.0	-7.0	0.000000	-0.800000	3.0	-4.0	7.0
4	-0.000000	12.0	-5.0	0.111111	-1.250000	3.0	-3.0	7.0
...	...	...	...	...	...	...	...	...
480	-0.250000	9.0	-10.0	0.181818	-1.333333	2.0	-2.0	7.0
481	-0.300000	15.0	-14.0	-0.125000	-1.000000	4.0	-2.0	7.0
482	-0.250000	12.0	-8.0	-0.083333	-0.888889	3.0	-2.0	7.0
483	-0.200000	14.0	-18.0	0.080000	-0.777778	3.0	-2.0	7.0
484	-0.000000	14.0	-8.0	0.045455	-1.000000	4.0	-4.0	7.0

Swarm Intelligence

ANN with particle swarm optimisation in the place of backpropagation is implemented. It results in the decrease of loss function values and makes sure that the swarm particles don't get biased towards the local minima.

However, the only disadvantage is that PSO requires lot of resources. As in this project made use of a 100 particles for 10,000 iterations, even with a powerful computing tool it took almost 4 hours to formulate. Even though increasing the particles would increase the accuracy, It consumes a huge amount of time or expensive computing tools.

```
from pyswarms.utils.functions import single_obj as fx
options = {'c1': 0.5, 'c2': 0.5, 'w': 0.9}

# Call instance of PSO
dimensions = (7* 20) + (20 * 7) + 20 + 7
optimizer = ps.single.GlobalBestPSO(n_particles=100, dimensions=dimensions, options=opt

# Perform optimization
cost, pos = optimizer.optimize(f, iters=10000)
```

```
17250 17250
17250
17250 17250
17250
17250 17250
17250
17250 17250
17250
17250 17250
17250
17250 17250
17250
17250 17250
17250
17250 17250
17250
17250 17250
17250
17250 17250
17250
17250 17250
```

Network Architecture

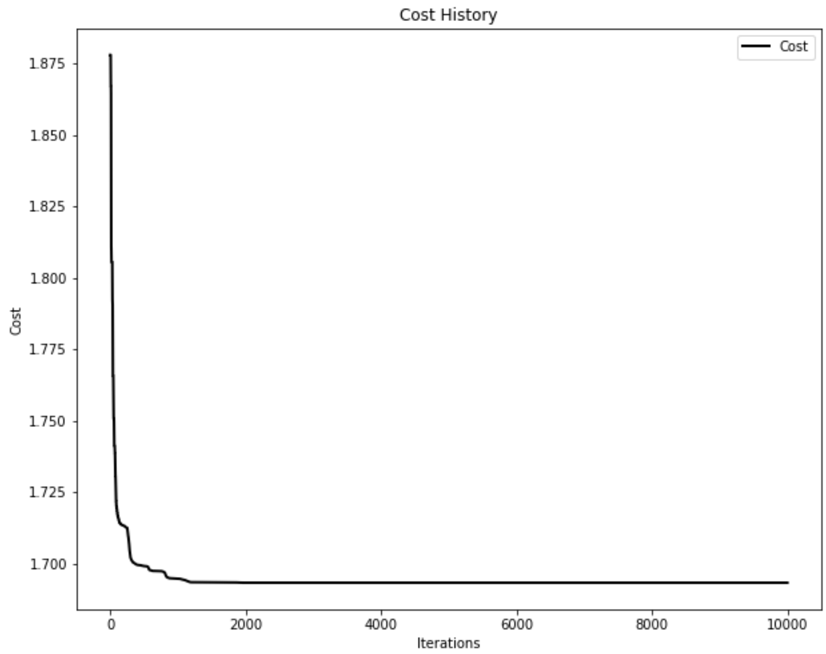
```
def predict(X, params):
    """
    Use the trained weights to perform class predictions.

    Inputs
    ------
    X: numpy.ndarray
        Input Iris dataset
    pos: numpy.ndarray
        Position matrix found by the swarm. Will be rolled
        into weights and biases.

    # Neural network architecture
    n_inputs = 4096
    n_hidden1 = 200
    n_hidden2 = 150
    n_hidden3 = 100
    n_classes = 5
    # Roll-back the weights and biases

    W1 = pos[0:819200].reshape((n_inputs,n_hidden1))
    b1 = pos[819200:819400].reshape((n_hidden1,))
    W2 = pos[819400:849400].reshape((n_hidden1,n_hidden2))
    b2 = pos[849400:849550].reshape((n_hidden2,))
    W3 = pos[849550:864550].reshape((n_hidden2,n_hidden3))
    b3 = pos[864550:864650].reshape((n_hidden3,))
    W4 = pos[864650:865150].reshape((n_hidden3,n_classes))
    b4 = pos[865150:865155].reshape((n_classes,))
    """
    # Neural network architecture
    n_inputs = 7
    n_hidden = 20
    n_classes = 7

    # Roll-back the weights and biases
    W1 = params[0:140].reshape((n_inputs,n_hidden))
    b1 = params[140:160].reshape((n_hidden,))
    W2 = params[160:300].reshape((n_hidden,n_classes))
    b2 = params[300:307].reshape((n_classes,))

    # Perform forward propagation
    z1 = X_Train.dot(W1) + b1  # Pre-activation in Layer 1
    a1 = dlrelu(z1)      # Activation in Layer 1
    z2 = a1.dot(W2) + b2 # Pre-activation in Layer 2
    logits = z2          # Logits for Layer 2

    y_pred = np.argmax(logits, axis=1)
    return y_pred
```

The Artificial Neural Networks was implemented using above architectures,

3 layers with 7 nodes in the input layer,20 nodes in the hidden layer and 7 nodes in the output layer which had the highest accuracy of all the three.

Results

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9439a480d0>
```




```
pyswarms.single.global_best: 100%|██████████|10000/10000
, best_cost=1.69
2020-08-07 22:38:26,064 - pyswarms.single.global_best -
INFO - Optimization finished | best cost: 1.693221006035
326, best pos: [ 5.19982896e-01  1.24066362e+00  5.61606
572e-01  3.94350613e-01
  7.30899141e-01  5.51598090e-01  5.64102273e-01  1.3232
8011e-02
  9.54371984e-01  8.68860921e-01  1.43720666e-02  9.7817
3869e-01
  1.00130894e+00  1.52571587e-01  6.35147287e-01  1.0577
5321e+00
.
.
.
.
.
..................
 -5.44970090e-03  4.72990560e-01  6.06094622e-01  1.1163
3494e+00
  3.63997779e-01  5.32698403e-01  6.15853744e-01  1.4562
1365e+00
  1.03592556e+00  1.55742974e-02  9.35783799e-01  5.8219
3879e-01
```

```
  7.81233168e-01  8.86706162e-01  3.85063524e-01  1.0023
8066e+00
  1.18645084e+00  7.17041648e-01  1.34514342e+00  1.4699
1571e+00
  1.00702889e+00  3.44772640e-01  6.13451421e-01  2.9035
2320e+00
  6.75639917e-01  3.56203623e-01  6.25867563e-01  1.1086
5699e+00
  9.01565761e-01  7.16527552e-01  9.14006090e-01  1.0340
4718e+00
  7.23955676e-01  1.05981498e-01  1.06148153e+00]
```

Conclusion

In this work, we have proposed a novel approach based on artificial neural networks and particle swarm optimization to detect the emotions from facial images by means of feature extraction and calculating feature values from the extracted features.

Implementing particle swarm optimization has decreased cost function values and loss of ANN. Another advantage of using PSO is that it is not biased towards local minima of the cost values.

References

- Ghandi, Bashir Mohammed, R. Nagarajan, and Hazry Desa. "Facial Emotion Detection using Guided Particle Swarm Optimization (GPSO)." International Conference on Man Machine Systems (ICoMMS 2009). 2009.
- Bruno Seixas Gomes de Almeida and Victor Coppo Leite (December 3rd 2019). Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems, Swarm Intelligence - Recent Advances, New Perspectives and Applications, Javier Del Ser, Esther Villar and Eneko Osaba, IntechOpen,
- https://medium.com/technology-invention-and-more/everything-you-need-to-know-about-artificial-neural-networks-57fac18245a1
- Chiranjeevi, Pojala, Viswanath Gopalakrishnan, and Pratibha Moogi. "Neutral face classification using personalized appearance models for fast and robust emotion detection." IEEE Transactions on Image Processing 24.9 (2015): 2701-2711.