Classification of happy faces with Logistic Regression

Daniel Viteri Noguera
Universidad de los Andes
d.viteri@uniandes.edu.co

1. Introduction

One of the main challenges in the computer vision area is the classification of images. It can be define as the labeling of an image into one or more predefined classes [1]. Human facial recognition is an example of a classification problem, and is of great interest to social networks, security companies and automobile industries. Furthermore, a more specific challenge is the recognition of facial expression, that can be used to detect fatigue in drivers.

Throughout the years, many methods have been proposed, including, for example, the histogram of oriented gradients (HOG) as descriptors to train SVM or K-means classifiers [2]. However, improvement in the results of these methods were not significant towards the half of the last decade, before the introduction of deep learning. Logistic regression is a basic machine learning concept that can be used to recognise facial expressions, which is the aim of this work.

To fulfill this task, the FER2013 database, evaluated in the Kaggle challenge [3] will be used. This consists of 48X48 gray-scale images with 7 different expressions (Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral). The training set consists of 28,709 examples, while the test set consists of 3,589. Sample data can be seen in figure 1. As observed, the images have centered faces, from which some are occluded in a minor level, making it not a difficult dataset.

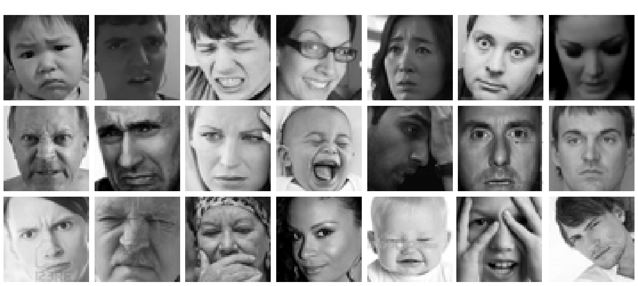

Figure 1. Sample images from the FER dataset.

2. Method

The logistic regression algorithm uses a linear equation with independent predictors, whose parameters are trained with the training dataset. Unlike linear regression, logistic regression uses the sigmoid function to predict values in a range from 0 to 1 [4]. Later, this output is set in a binary manner, setting the optimal threshold to finally make the prediction. Values above this threshold (included), will be 1, while values below it will be 0.

This model learned by setting the initial parameters randomly, and predicting the value of a batch of the training set. From these predictions, the loss is calculated and through the stochastic gradient descent optimizer, the parameter are learnt to minimize the error function. The magnitude of this variation is controlled by the learning rate, which was fixed to 0.00001.

The training set was divides into train and validation, where validation is simply the last 30 percent of the original training set. In order to choose the best batch size value, this was varied from 50 to 950 with steps of 50. When the loss was the minimum for the validation set, the batch size was saved, therefore obtaining the optimal in the range that was evaluated.

3. Discussion

The processes described in this work were supposed to be run in the available server's CPU, and that it was not in the capacity to run all the experiments that the users wanted. An epoch, for example, would take at least a minute to run, therefore limiting each individual work. For this reason, the researcher computer was used for the current paper's work.

After testing different values for the batch size to identify the optimal one, the result was that with 950 images per batch, the validation set loss would be minimum in the range evaluated. The reason for this, is that there are about 28,000 training images that could be used to obtain better results. However, we failed to identify if the trade off between better results and memory usage was worth it, because there was not a threshold set to determine if the bigger the batch, the significantly better the result. This, means that even if there was a small variation in the loss of the validation set, the algorithm would identify this as an improvement and save this value, regardless if adding images to the batch was worth it or not. For this reason, convergence was also not analysed.

Also, as the algorithm identified this minimum loss of the validation set, the trained parameters for the "optimal" batch size were saved in order to used them in the test set. The confusion matrix and ACA of the test set can be seen in figure 2.

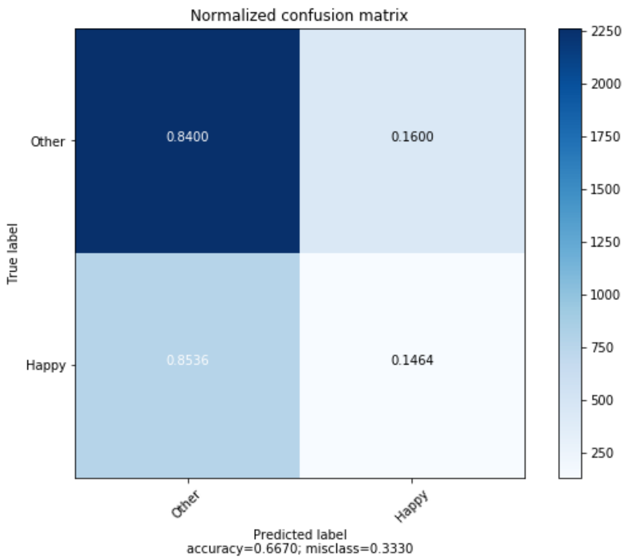

Figure 2. Normalized confusion matrix and ACA for the happiness classification.

These results, were obtained after determining the optimal threshold when the sigmoid function converted the regression predictions into values between 0 and 1. Starting from 0.5, and going upwards with a step of 0.1, the better results were obtained with a threshold of 0.99, which means that only values above 0.99 were taken as a happy face. This was made in the test set, taking into account that in the model was taught by only considering the minimization of the validation set loss.

Knowing that the FER dataset is not a complicated set, an ACA 0.67 is not a good result. However, this can be explain bearing in mind that the only parameter tuned up was the batch size, and the learning rate was set to a fixed value, due to insufficient time caused by limited processing resources. Also, since there are 7 different emotions in the data set, the other classes that were not happiness could be very different among them. This, would add complexity to the task of recognizing happy faces. However, this ACA proofs that the model is actually learning and could be improved.

The learning rate, if modified, would have provided information about its ability to make the model converge faster or slower. Depending on the initialization values of the model, the loss function could reach a local minimum that may need certain learning rate, to then converge in a global minimum. However, if it is to large, divergence may be obtained, making it impossible to train the model. It would, in deed, be wise to re-scale the learning rate to search for an optimal value. However, this could not be generalized since it can depend on the initialization parameters of the model.

References

[1] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", Neural Computation, vol. 29, no. 9, pp. 2352-2449, 2017. Available: 10.1162/neco_a_00990 [Accessed 7 April 2019]. 1

[2] I. Revina and W. Emmanuel, "A Survey on Human Face Expression Recognition Techniques", Journal of King Saud University - Computer and Information Sciences, 2018. Available: 10.1016/j.jksuci.2018.09.002 [Accessed 7 April 2019]. 1

[3] "Challenges in Representation Learning: Facial Expression Recognition Challenge — Kaggle", Kaggle.com, 2019. [Online]. Available: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data. [Accessed: 07- Apr- 2019]. 1

[4] R. Ghandi, "Introduction to Machine Learning Algorithms: Logistic Regression", Hacker Noon, 2019. [Online]. Available: https://hackernoon.com/introduction-to-machine-learning-algorithms-logistic-regression-cbdd82d81a36. [Accessed: 07- Apr- 2019]. 1