Emotion detection with Logistic Regression

Laura Munar Acosta
Universidad de los Andes
l.munar10@uniandes.edu.co

Maria Ana Ortiz
Universidad de los Andes
ma.ortiz1@uniandes.edu.co

1. Introduction

Emotion recognition is an integral part of quantitative studies of human behavior. The emerging areas of human behavioral signal processing and behavioral informatics offer new analytical tools to support a variety of applications, including the design of natural human machine interfaces (HMI). Emotionally-cognizant human-computer and human-robot interfaces promise a more responsive and adaptive user experience. In real life situations, behavioral computing must reconcile information in the context of a situated interaction [3].
Many applications can benefit from an accurate emotion recognizer. For example, customer care interactions (with a human or an automated agent) can use emotion recognition systems to assess customer satisfaction and quality of service. Other tasks that rely on observational coding of human interaction, such as in therapeutic settings can benefit from robust emotion recognition. These applications can benefit from the design of a robust emotion recognition scheme, which should also be easily adaptable to different interaction scenarios [3].
The logistic regression model has become a widely used and accepted method of analysis of binary outcome variables. This popularity stems from the availability of easily used software in both mainframe and microcomputer packages and the ease of interpretation of the results of the fitted model, be it used for estimating probabilities and/or odds ratios. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables [2].
The present work aims to study the capability of the Logistic Regression model to classify emotions in the Fer2013 dataset. The data consists of 48x48 pixel grayscale images of faces. The training set consists of 28,709 examples and public test set used consists of 3,589 examples.

2. Materials and Methods

2.1. Model

Method for model is based on a logistic regression model, a first approach to machine learning algorithms. It had a linear layer for estimation of parameters (W and b) that will best fit for prediction of parameters. For training model it was set a early stop in case error in the actual epoch is greater that error in the past epoch. Parameters such as number of images per batch, learning rate, and number of epochs where variated. Differences on models are explained on the next two sections.

2.1.1 Smiling Classifier

For the model that was trained to detect smiling faces over any other of the 6 emotions remaining on the dataset, function for defining classification was a sigmoid, that set a probability number between $[0, 1]$ and then this probability was used for creating precision and recall curves reported on results section. As mentioned evaluation of the method was precision and recall curves and F-measure.

2.1.2 Multiple Emotions Classifier

For the model to classify the 7 emotions on the dataset loss function and function for classification was change to a cross entropy (2) and softmax functions (1).

$$p_i = \frac{e_i^a}{\sum_{K-1}^{N} e_k^a} \tag{1}$$

$$H(y,p) = -\sum_i y_i log(p_i) \tag{2}$$

3. Results

In table 1 are presented the experiments done in the case where were only two emotions (happy/not happy).This experiments were performed in order to see the influence of the hyperparameters.
The behavior of the increase of the number of epochs in the loss in presented in figure 1.

Table 1. Results on the experiments done in the binary case

# Epochs	**Learning Rate**	**Batch size**	**Fmax**
10	0,00010	50	0,3832
100	0,00010	50	0,3995
1000	0,00010	**50**	0,4021
1000	0,00010	**150**	0,4026
1000	**0,00010**	**300**	0,4047
1000	0,00010	**400**	0,3954
1000	0,00010	**600**	0,3980
1000	0,00010	**800**	0,3980
1000	**0,001**	300	0,3998
1000	**0,01**	300	0,3937

Also, the figure 2 presents the Precision Recall Curve for the best method.

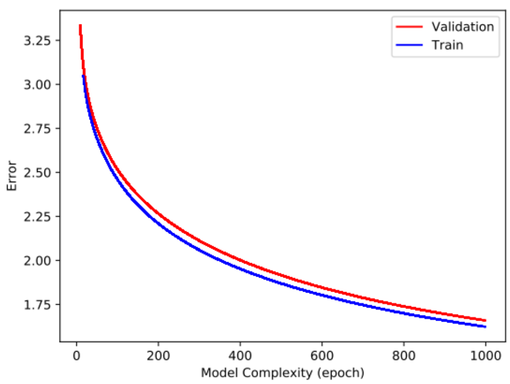

Figure 1. Influence of number of epochs in the loss of the model

On the other hand, below is presented the loss for the trained set and validation set for the multiclass model.

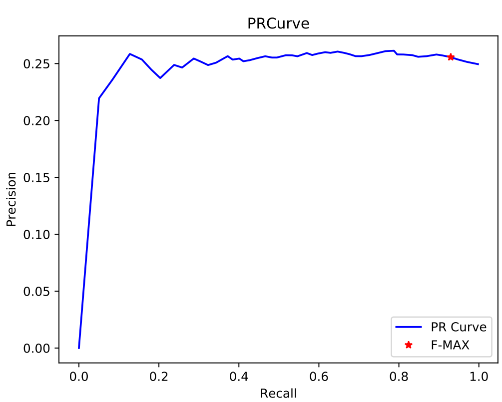

Figure 2. Precision and recall curve for best method

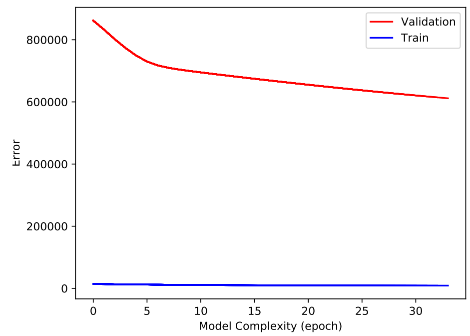

Figure 3. Error lost for multiple class

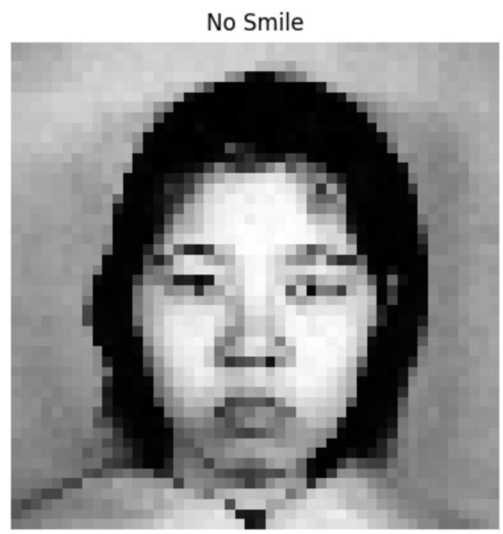

Figure 4. Result In the wild

Figure 5. Result In the wild

4. Discussion

4.1. Hyperparameters influence

As it can be seen on table 1, an increase in the number of epochs results on a better performance of the Logistic Regression model. This is explained because the epochs are a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. The number of epochs is traditionally large, often hundreds or thousands, allowing the learning algorithm to run until the error from the model has been sufficiently minimized [1]. The figure 1 corroborates what was mentioned before. Here, it can be seen that the loss decreases as the epochs increases. In a case with unlimited computational resources, this graph is useful to see if there is an overfitting of the model and that will represent an increase in the validation loss but not on the train set.
On the other side, the behavior of batch size hyperparameter is different. The best performance is for 300 batch size but an increase above this number results in a decrease in the performance. Considering that the batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters; the decrease on the performance is explained on the fact that the model may be changing its internal model parameters to fit outliers; so, when the test set is evaluated the performance is lower [1].
Finally, the learning rate parameter controls how much we are adjusting the weights of our network with respect the loss gradient. The lower the value, the slower we travel along the downward slope. Furthermore, the learning rate affects how quickly our model can converge to a local minima [4]. In table 1 is corroborated what was mentioned before because the lowest value of learning rate resulted in the greatest performance of the model.

4.2. Best Method - Binary case

According to Figure 2, the best performance on the smiling classifier gives a 0.4 Fmax value. The accuracy of this method could be improved using different models to train. This figure shows that for lower recall values the precision is low too. And this could be explained on the fact that since the model is a binary case, for this combination of low precision and recall the model is not recognizing well the False Negative and the True Negatives in neither class. In figures 4 and 5 is presented the evaluation of the Smiling method in two images taken in real life. As it is seen, both pictures were correctly predicted.

4.3. Multiple class model

In Figure 3 it can be seen that for a multiple class model the loss is considerable bigger than for the binary case. This may tell us that there are greater number of epochs more to try out, although this will increase the computational time. Meaning that for a 6 class problem, the model is having complications to converge. Considering what was mentioned above, a different method or model should be tried with this problem. Additionally, in future studies a hyperparameter influence should be performed in order to establish if it was the configuration of the parameters what made the model fail to converge.

5. Conclusions

Learning rate hyper-parameter has a high effect on metrics, mainly because as mention it tells how fast the model learned, so high values lowers metrics.
Scale is a factor that has an effect for in the wild images, mainly because classifier has an specific size.
As a future work, is highly recommended to use different models to describe the multiclass problem. Additionally, if the computational resources are available, the tuning of hyperparameters will also be useful.

References

[1] J. Brownlee. What is the difference between a batch and an epoch in a neural network?, 2019.

[2] D. W. Hosmer and S. Lemesbow. Goodness of fit tests for the multiple logistic regression model. *Communications in statistics-Theory and Methods*, 9(10):1043–1069, 1980.

[3] C.-C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayanan. Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, 53(9-10):1162–1171, 2011.

[4] H. Zulkifli. Understanding learning rates and how it improves performance in deep learning, 2019.

Annexes