Lab 10: LogisticReg

Adrian S. Volcinschi M.
Universidad de los Andes
https://uniandes.edu.co/en
as.volcinschi@uniandes.edu.co

Abstract

The problem of human emotion recognition has been around for almost 30 years now and it is yet to be fully solved. The idea of this paper is to see how good are basic machine learning methods at classifying 7 basic human emotions (happy, sad, angry, disgusted, neutral, feared, and surprised) in the FER2013 dataset. The problem was first implemented as a binary problem of happy vs not happy with logistic regression and then adapted to a multiclass problem with Softmax - Cross-Entropy. The results for the binary problem seem to be promising (ACA = 0.52) considering it used only pixel intensities and information whereas the multiclass problem seems really hard to fulfill with this approach with a poor result (ACA = 0.14).

1. Introduction

We as humans tend to analyze a person as soon as we look at them. We can tell sometimes if a person is happy or not based on the look of their face, but it is not certain all of the times. Because of this, many people have tried to create models that can tell the emotions a person is feeling based on a photo of their face. One of the first approaches was realized by De Silva et al. [1] in 1997 which used pictures and audio as a multi-modal approach to detect emotions. More recent works have implemented deep learning methods such as deep neural networks [5] and recursive neural networks [2] to achieve a solution for this problem.

The purpose of this paper is then to proof if using the most basic concepts of machine learning such as Logistic Regression and Softmax - Cross-Entropy can help us tackle this problem. The idea behind this machine learning methods is to construct a set of parameters theta that can be calculated through a gradient approach of our data and labels and then be updated to get a better result in the Cost function. In other words, the idea is to find some parameters that reduce our error in classification by an iterative method using derivatives of the Cost function to find which parameter needs to be adjusted more for a better performance [3].

2. Materials and Methods

To train the emotion classifier the FER2013 [7] dataset was used. This dataset consists of 35,888 human faces with different angles and emotions. For this paper, the whole dataset was used following the partition proposed by the author: 28709 train and 3589 test. Additionally, all the photos are grayscale images of size 48x48. Some of the train images are shown in Figure 1

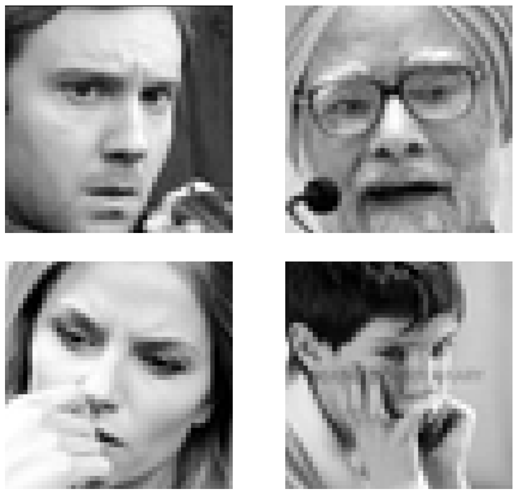


Figure 1: Pictures of the FER 2013 dataset

2.1. Logistic Regression

The raw 48x48 images were transformed into a unique vector of size 2304. The labels go from 0 to 6 depending on the emotion the person has on the image. For the first part, a binary classification approach was intended. This meant that labels were 1 for happy emotion and 0 for any other emotion. The idea of the Logistic Regression is to calculate some W, of size mxn, and b. Then with those parameters the model can receive an input vector of size 1xm so that it returns a nx1 vector where n depends on the amount of outputs the model should have. In this particular case, m is going to be 2304 and n is going to be 1 since the output is only a 1 for happy and 0 for not happy. This number can be any real number so a normalization function is needed to get a number between 0 and 1 that can then be thresholded to obtain a label 0 or 1. The normalization function used is the Sigmoid function shown in Equation 1 [6].

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

2.2. Descending Gradient

The W and b are going to be calculated through a descending gradient method which begins with a random initialization for W and b. When both, W and b, are determined, they are used to predict the labels of the train images. When the predicted labels are calculated they are used to calculate a gradient between them and the true labels. The W and b are updated using the gradient and then through a loss function, in this case a logistic cost function shown in Equation 2, it is calculated how much better are the new labels with the new W and b [6].

$$Cost(pred, gt) = \begin{cases} -log(pred) & gt = 1 \\ -log(1-pred) & gt = 0 \end{cases} \tag{2}$$

2.3. Softmax - Cross-Entropy

The previously defined model can be adapted from a binary classifier to a multiclass classifier. This can be done by changing the sigmoid function that returns a number between 0 and 1 to a Softmax functions that returns a 1xn vector, where n is the amount of classes, with each element of the vector being the probability of the object to be of that class. The implementation for this approach was based on the code of Usman Malik [4] where he implements it as a single layer neuronal network. The code was then adapted to fit on this problem.

2.4. Evaluation Method

For the binary problem two different metrics were implemented. On one hand, the Precision-Recall curve was calculated alongside the F-measure. On the other hand, the ACA for the confusion matrix of 2x2 was calculated to get an average precision of classification.For the multiclass problem the only metric implemented is the ACA for the confusion matrix of 7x7.

3. Results

All the results shown are with a random seed 1 to guarantee consistency in the results.

3.1. Loss, Learning Rate, and Batch Size

We can estimate how good is our algorithm is learning by looking at the loss at the end of each epoch. For this paper the models were trained with 40000 epochs, 0.00001 of learning rate, and a batch size of 100. If we look at Figure 2 it is clear that the loss is decreasing constantly in the binary problem whereas the loss in the multiclass problem, presented in Figure 3, seems to stabilize. What this means is that the loss of the binary problem can go down even more if more epochs were added. On the other hand, the loss for the multiclass problem seems to be steady as it might have found a local minimum.

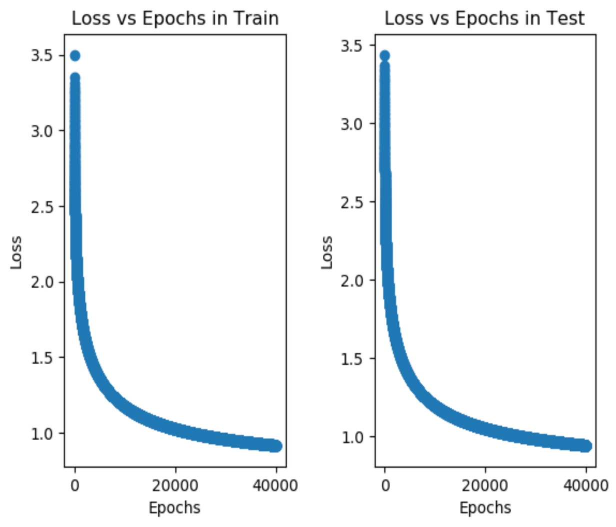


Figure 2: Loss vs Epochs in the binary problem with constant learning rate

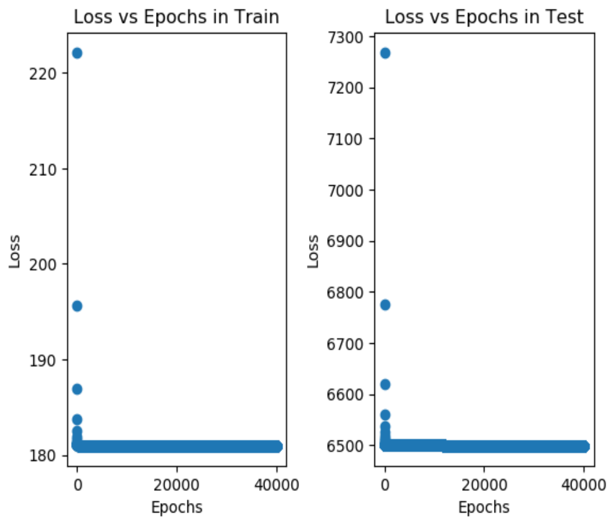


Figure 3: Loss vs Epochs in the multiclass problem with constant learning rate

Another attempt to increase convergence was done by starting with a high learning rate (0.01) and every 500 epochs reduce it by a factor of 10 using only 5000 epochs. The loss convergence for this approach can be seen in Figure 4. It is to be expected that a better result is obtained since we are using a random initialization it can be far from the minimum so it is better to take big steps at the start for more coarse changes and reduce it as epochs go by to refine the parameters. It is clear from Figure 4 that convergence is faster this way and the same results are obtained.

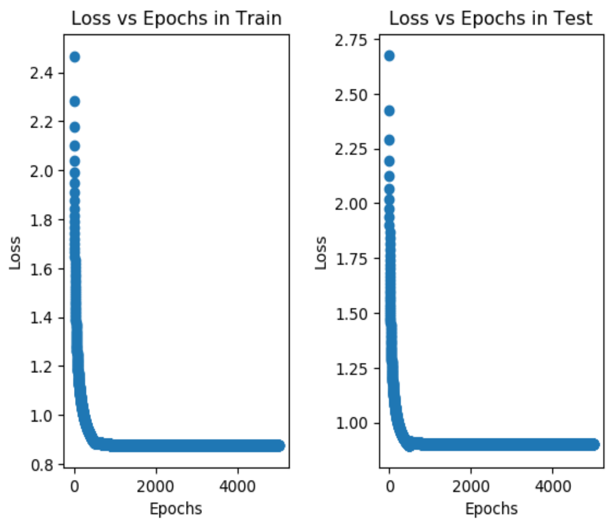


Figure 4: Loss vs Epochs in the binary problem with changing learning rate

The parameter of batch size is also important since a low batch size is not going to let our model train on a representative sample. The dream is to be able to train with the whole dataset but due to memory limitations this can't be done and therefore the need for batch sizes.

3.2. Threshold

The threshold is an important parameter to play with since it is the one that regulates the amount of classifications labeled as happy or not in the binary case. This parameter was varied between 0.4 and 0.6 as shown in Table 1. It can be seen that a low threshold would give a worst result because it would let more not happy faces to go through whereas a high threshold would be too selective and limit the amount of happy faces.

Table 1: ACA variation when the hyperparameter threshold is variated

Threshold	0.4	0.5	0.6
ACA	0.5056	**0.5078**	0.5062

3.3. F-measure

Since the binary model is going to give us a probability between 0 and 1 a Precision-Recall curve can be calculated. This curve, shown in Figure 5, can tell us that the method has an overall steady precision throughout the recall spectrum. With this into account, we can see that the F-measure for this model is 0.4 which is not bad considering it is trying to classify a person as happy or not based on only pixel intensities information.

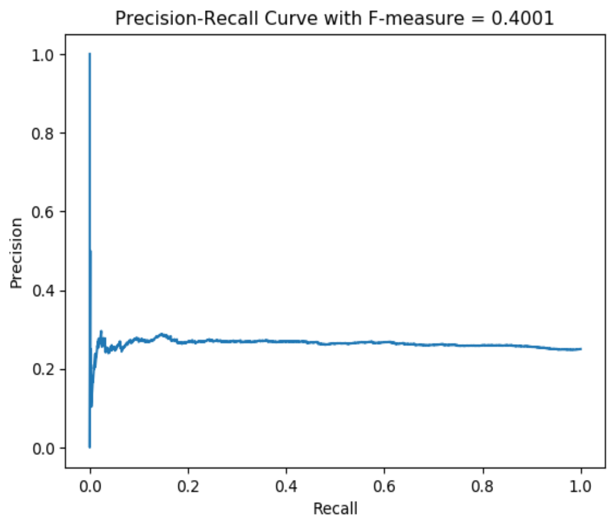


Figure 5: PR curve obtained for the binary problem

3.4. ACA

With the results of the F-measure shown previously we can tell that the algorithm has an overall good precision. Since the ACA of a confusion matrix is taking only into account recall and not precision, it is to be expected that the ACA is higher than the previously obtained F-measure that takes into account precision. As it can be seen in Figure 6 the ACA for the binary problem is higher than the F-measure of before. Considering it only has the pixel's intensity as information it is really good to obtain a better result than just classifying randomly (0.5).

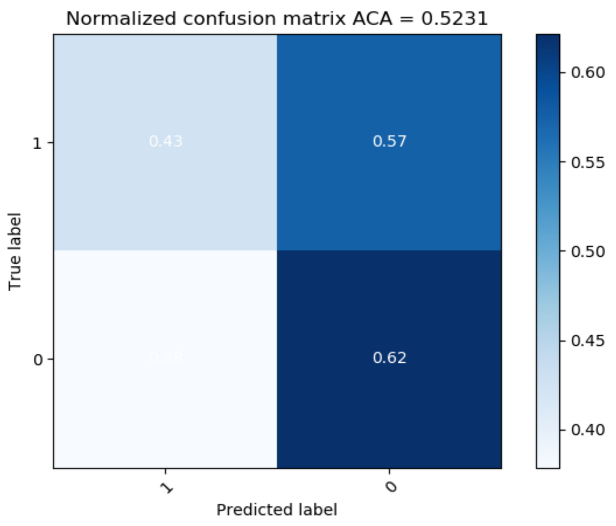


Figure 6: Confusion Matrix for the binary problem

If we take a look now at Figure 7 it is clear that this approach is not really good for a multiclass problem. The ACA obtained, 0.1368, is lower than a random classification (0.1429).

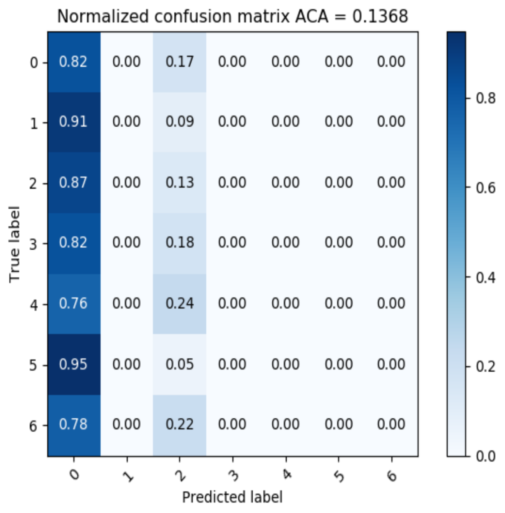


Figure 7: Confusion Matrix for the multiclass problem

3.5. Additional Test Images

Some additional test images were taken from the internet and a cellphone. These pictures can be seen in Figure 8. The purpose of this is to see if the algorithm is robust enough to classify different faces that are not from the dataset. The results can be observed in Figures 9 and 10.

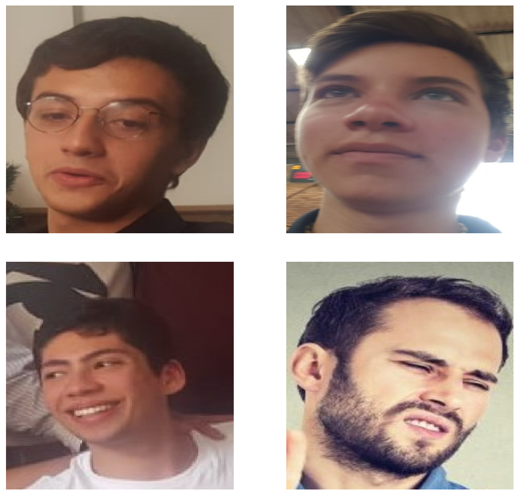
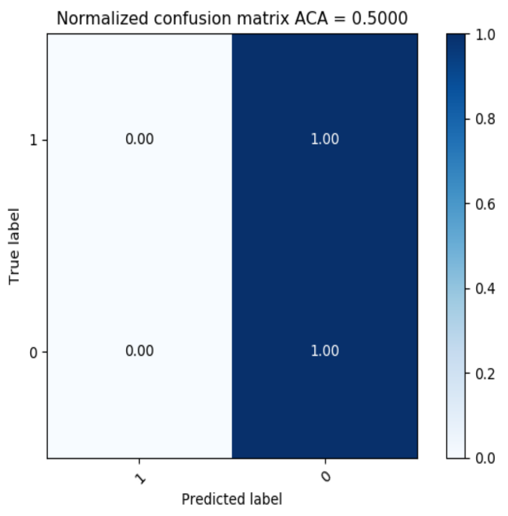
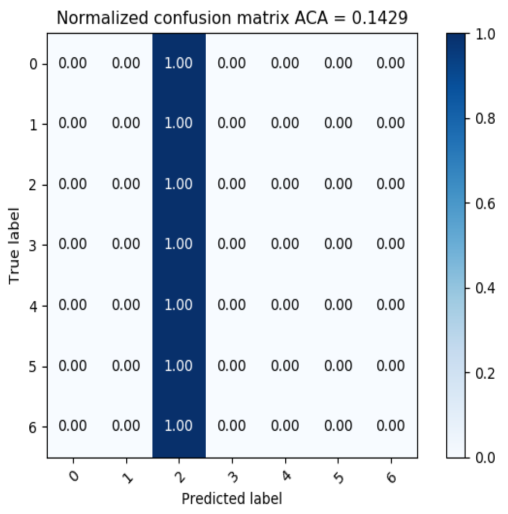


Figure 8: Additional extra pictures for testing



	1	0
1	0.00	1.00
0	0.00	1.00

Figure 9: Confusion Matrix for binary problem with extra photos



	0	1	2	3	4	5	6
0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
1	0.00	0.00	1.00	0.00	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00	0.00	0.00
3	0.00	0.00	1.00	0.00	0.00	0.00	0.00
4	0.00	0.00	1.00	0.00	0.00	0.00	0.00
5	0.00	0.00	1.00	0.00	0.00	0.00	0.00
6	0.00	0.00	1.00	0.00	0.00	0.00	0.00

Figure 10: Confusion Matrix for multiclass problem with extra photos

Looking at Figure 10 it is clear that the algorithm is not good for more natural images. Additionally, if the performance was bad in the testing set here it is shown that it fails miserably. The last hope for this model would be in the binary problem but as it is depicted in Figure 9 it is not a good result either.

4. Conclusions

Considering the model only had the pixel's intensity as information to classify as happy or not it did a decent job. If it had some sense of space or shape it could perform even better. Perhaps, if the same problem was done twice using a re scaled image it could boost the performance a bit up. Nevertheless, the model is not suited to handle this problem on the state it is. The multiclass problem failed because of the complexity of the problem itself, 7 emotions based on faces, and the simplicity of the model, only intensity information.

References

[1] L. C. De Silva, T. Miyasato, and R. Nakatsu. Facial emotion recognition using multi-modal information. In *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing. Theme: Trends in Information Systems Engineering and Wireless Multimedia Communications (Cat.*, volume 1, pages 397–401. IEEE, 1997.

[2] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal. Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 467–474. ACM, 2015.

[3] R. Gandhi. Introduction to machine learning algorithms: Logistic regression, May 2018.

[4] U. Malik. Creating a neural network from scratch in python: Multi-class classification, Oct 2018.

[5] A. Mollahosseini, D. Chan, and M. H. Mahoor. Going deeper in facial expression recognition using deep neural networks. In *2016 IEEE Winter conference on applications of computer vision (WACV)*, pages 1–10. IEEE, 2016.

[6] S. Swaminathan. Logistic regression - detailed overview, Mar 2018.

[7] R. Verma. fer2013, May 2018.