Detecting facial expressions

Julio Nicolás Reyes Torres
Universidad de los Andes
jn.reyes10@uniandes.edu.co

Juan David Triana
Universidad de los Andes
jd.triana@uniandes.edu.co

Abstract

This article presents the implementation of two statistic concepts (loss functions) to train a model, which is able to detect facial expressions from a dataset of faces images. One model is used to detect in binary form a happy or non-happy face, and the other one, to multiclass to determine the 7 different facial expressions from database. The theory of the models are presented, like so the optimizer employed to find the minimum global. Finally, the results of the implementation are presented for both loss functions to determine the main differences, performances and the advantages of each one.

1. Introduction

This article presents the training of a detection model with two different "loss functions". The goal is to identify facial expressions as a happy, angry, disgust, sad face, the images semantically have an emotional representation. The loss functions implemented were "Logistic Regression" for binary representation and "Softmax Cross-Entropy" for many classes.

Logistic Regression was used to train the detection model among two categories to identify a Happy and non-Happy face. Instead, Softmax Cross-Entropy function is used to determine all the different facial expression of database. In total, there are 7 kind of facial expressions.

In both cases the most popular optimization method "Stochastic Gradient Descent" is implemented to looking for a global minimum.

2. Methods

2.1. Data description

The Database is FER2013. It was introduced by Pierre-Luc Carrier and Aaron Courville in 2013. The photos are of 48x48 pixels in grayscale and they are well organized, each face is more or less centered with a similar size. The faces are divided among 7 categories. [5]

- Categories: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral.
- Grayscale images
- Size: 48x48 pixels
- train.csv file: "emotion": numeric code from 0 to 6 that represents the emotion and "pixels": A string surrounded in quotes for each image.

2.2. Logistic Regression Model

Also known as " like log odds " and " logit ", is a concept taken from the statistics applied to predict the binary result of a variable. Logistic regression is the model that implements the function used at the core of the method, the "logistic function". Initially, it was used in different biological sciences, for instance, in ecology to describe properties of population growth, they discovered a quickly rising and maxing out at the carrying capacity of the environment. [1]

The logistic function also knows as "sigmoid function" has the property of taking any real number and mapping it into a function with limits that tends to 0 and 1. Following, the equation (1) that represents it and in figure (1) is shown its shape.

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

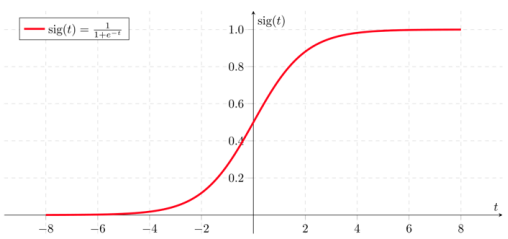

Figure 1: **Sigmoid function**

A logistic regression model will finally predict the probability of a class, that is, we are modeling the probability that an input (X) belongs to the default class (Y = 1), we can write this formally as:

$$P(X) = P(Y = 1|X)$$

So, a logistic regression is a linear method but each prediction is transformed with the logistic function to identify the different categories. That is, instead of understand the predictions as a linear combination of the inputs as happens with a linear regression, here the context is different. Specifically, with the "cost function", is important the method, a linear regression uses the "mean squared error', however, if this cost function is used with logistic regression, it will be a non-convex function and the optimizer "Gradient descent" will not converge into the global minimum. [8, 1, 6]

2.3. Softmax function

The "softmax function", also known as "normalized exponential function" can be interpreted as a generalization of the "Logistic function", which maps a k-dimensional vector z of arbitrary real values to a k-dimensional vector $\sigma(z)$ of real values in range $(0, 1]$, that is, is the normalization of the input vector with the s um of all the exponentials. [4, 6]

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{2}$$

2.4. Cross-Entropy

The "Cross-Entropy" analytically indicates the distance between the model output distribution, and the original distribution that really is. The equation that represents it, is given by:

$$H(p, q) = -\sum_{x} p(x)\ log(q(x)) \tag{3}$$

This function is widely used instead of mean squared error in classification problems, because in the case of cross-entropy function, the change is pretty significant using the logarithm. In a network, cross-entropy allows to asses into very small errors and work to eliminate them. In general, this function is a much more efficient optimization tool than mean squared error. [7, 2]

2.5. Stochastic Gradient Descent

This is an optimization algorithm based using convex functions, the goal is to find a local minimum. The adjust of the parameters is done using the "Gradient" which measure how much change the output of a function depending of the variation of the inputs.

In a function the gradient is equivalent to the slope, that is, a partial derivative with respect to its inputs. The behavior is related with the changing of the variable, if we have a high gradient, the slope is steep and the model is faster to learn. But if the slope is zero, simply the model is going to stop. [3]

3. Results

3.1. Logistic Regression Model

The results of the method "linear regression model" are shown on this subsection. There were some parameters to vary in this part, which could give a different insight of how change the results depending of the given parameters. One of these parameters was the number of epochs implemented in the algorithm. This is understand as the number of cycles of error reassessment on every iteration of the cost function evaluation on the validation and train sets. The following figure shows an interaction among epochs and cost error, regarding train and validations sets.

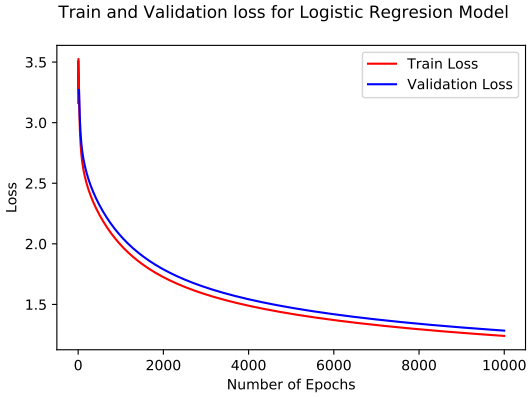

Figure 2: **Train and validation loss for Logistic Regression Model along of time epochs.**

3.2. Cross-Entropy

The results of the Cross-Entropy, this algorithm is based to use the logarithm to identify small errors to eliminate them, it is cataloged as an efficient optimization tool.

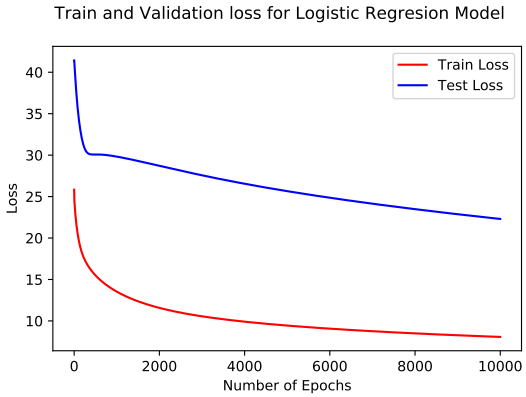

Figure 3: **Train and validation loss for Cross-Entropy along of time epochs.**

4. Discussion

The logistic regression model predicts the probability of a class, in figure (2) is presented the prediction error depending of the number of epochs. This graph shows the behavior of the model to train and test the facial emotions detector in a huge number of epochs, the tend is to decrease the mistakes according with a more quantity of epochs.

The model that implements Cross-Entropy keeps the tend to decrease the number of mistakes along epoch time, thought the number of the prediction error is high, that happens because of the detection among all classes, the model discriminate the probability of being a specific class among different possibilities.

An important hyperparameter was the “batch size”, which has a significantly response because of the execution time; that is, a higher batch size it will contain a higher number of samples to work in an epoch, and the execution is faster.

The learning rate is another fundamental hyperparameter, it is very important because this is the number that represents the “speed” of learning. In other words, is the rate to advance into a model. A high value implicates that the model pass fast between the function but it can leave an important point and maybe never converge. In our implemented algorithm the best value was to 0,0001.

5. Conclusions

- In the implemented models and with the parameters used, the test loss never found an optimum point to determine an over-fitting, instead, both models “logistic regression” and“cross-entropy” always were decreasing this prediction error.

- Both Cross-entropy as logistic regression is widely used in classification problems, the way to determine the probability is finer in cross-entropy because it has the ability to detect small error to eliminate, it is an efficient optimization tool.

References

[1] J. Brownlee. Logistic Regression for Machine Learning.

[2] P. Dahal. Classification and Loss Evaluation - Softmax and Cross Entropy Loss.

[3] N. Donges. Gradient Descent in a Nutshell – Towards Data Science.

[4] hamza Mahmood. Softmax Function, Simplified – Towards Data Science.

[5] Kaggle. Challenges in Representation Learning: Facial Expression Recognition Challenge.

[6] Statistics Solutions. What is Logistic Regression?

[7] SuperDataScience Team. Convolutional Neural Networks (CNN): Softmax & Cross-Entropy.

[8] S. Swaminathan. Logistic Regression — Detailed Overview – Towards Data Science.