Film Rating Prediction Using Neural Networks

Juliana Henao
Ingeniería Matemática
Universidad EAFIT)
Medellín, Colombia
jhenaoa4@eafit.edu.co

***Abstract*—Film Industry is one of the most important industries in the world, making a big impact in our society. But film making requires many resources. Producers are interested in finding the future commercial performance of a film project. In this project, a set of neural networks predict if a film will be nominated to an Oscar Award. This neural networks try to find patterns within the films that have the most award success. This research compares neural networks, making combinations of number of neurons per layer and hyper-parameters, on the data set formed by the processing some characteristics of a film.**

***Index Terms*—neural network, classification, films**

I. Introduction

Cinematography is a way of art that everyone has enjoyed in their lives, the seventh art has a direct impact in society, culture and economics. That is one reason why Film Industry is one of the most important in the world and worth billions of dollars. But it is well known that the cost of the production of a film can be very high, and this budget affects directly the audience rating and the performance in the awards, this is why the accuracy when predicting financial performance is important when making an investment.

One of this applications in Motion Picture Industry is the prediction of what films people will want to see. Researchers from film studio 20Th Century Fox say that understanding detailed audience composition is important for movie studios that invest in stories of uncertain commercial outcome [1]. This film positioning can be based in many different criteria, based on genre, on directors value, synopses and others. Besides, the analysis of audience success based on genre can be done in many different ways, by analysing different components of a movie or any kind of motion picture. Other interesting researches in this topic are made by Y. J. Lim and Y. W. Teh in the paper "Variational bayesian approach to movie rating prediction" [3], and the work of J. D. Mcauliffe and D. M. Blei in 2008 [4].

Data science is a very useful tool to find implicit patterns that are intuitively perceived but difficult to grasp it self. So that, Artificial Intelligence has reached the enough development to be applied to creative industries like music, painting, literature and of course cinematography, which seemed impossible a few decades ago.

For this reasons, there is a big opportunity of research in this field. In this project the approach that is going to be taken is to predict whether a movie is going to be nominated to an Oscar or not, using neural networks. It is interesting to research if there is a relation between the main characteristics of a film and its awards.

II. Conceptual Framework

In this research it will be taken several hypothesis classes of neural networks to model the space the data generates. Every hypothesis is a neural network with different number of hidden layers and different number of neurons per layer.

The model consists on figuring out the optimal weights to fit the data. It emulates the external stimuli input which is weighted w_{ij} to generate a signal $v_j(n)$. The signal is then transformed by an activation function $\phi_j(v_j(n))$ that is the neuron capacity to provide a response $y_j(n)$. The activation function for this research is the sigmoid function, given by the equation bellow.

$$P(v) = \frac{1}{1+e^{-v}} \tag{1}$$

The architecture of the neural networks is given by all the combinations of L hidden layers, with $L = 1, 2, 3$; and the number of neurons per hidden layer is $l_i = 1, ..., 5$. Also, this architecture has n neurons in the input layer and m neurons in the output layer. With a sample of size N.

Since there is no possibility of defining learning guarantees for the neural networks, the sample is divided into training, validation and testing. The stop criteria for the neural network is given by the tolerance to error, which is measured

Finally, it is important to define the set of events ω, where lies the population of the problem.

III. Method

The data used for this research is from the combination of two Kaggle data sets [2]. The resulting data set contains the following features.

- Title of the movie.
- Rating, a categorical variable indicating the Motion Picture Association film rating.

- Genres, a categorical variable indicating the main genre of the movie.
- Duration of the movie in minutes.
- Number of years since the film premiere.
- Votes in IMDb.
- Score in IMDb.
- Country, 1 if the country is The United States, 0 if not.
- Budget of the film.
- Nominated variable, 1 if it was nominated to an Oscar, 0 if not.

A. *Sampling*

In order to choose a good and balanced set to train and test de models, it is important to shuffle and sample the data according to the following algoritm.

Algorithm 1 Shuffle and Sampling Algorithm

0: **Input:** Matrix $M \in \mathbb{R}^{mxn}, k \in (0,1)$
0: **Output** M_{train}, M_{train}
0: Permute index of M using distributions (normal, uniform, gamma)
0: $M \leftarrow$ Shuffle M according to the permutation
0: Reset index
0: $M_{train} \leftarrow$ Take k percent of M
0: $M_{test} \leftarrow$ Take $1-k$ percent of M
0: Return M_{train}, M_{train} =0

B. *Visualization*

The target for this exercise is the variable of nomination, which is a binary variable, indicating if the movie is nominated to an Oscar. In Figure 2, there is a scatter plot for each pair of features. It can be seen that there is not any lineal relation between this variables.

Fig. 1. Scatter plot of each pair of features

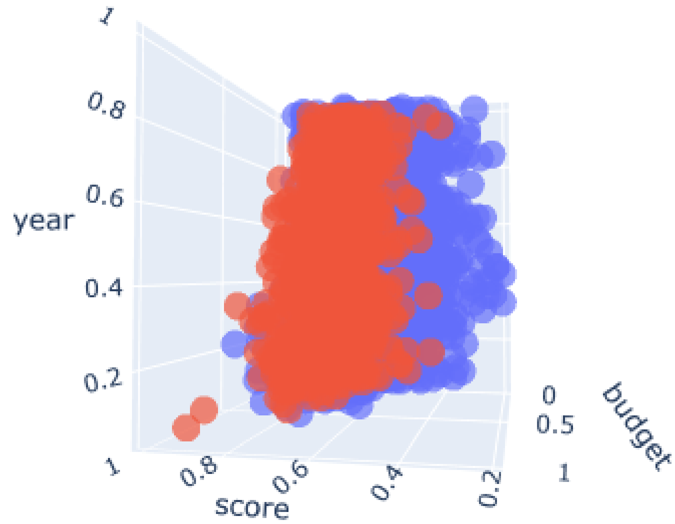

Fig. 2. Scatter plot 3 variables, where the blue color represents the non nominated movies and red represents the nominated ones.

Also, in Figure 3, it is shown the proportion of each class. Clearly we have imbalanced classes, the percentage of films with good rating is much bigger that the films with bad rating.

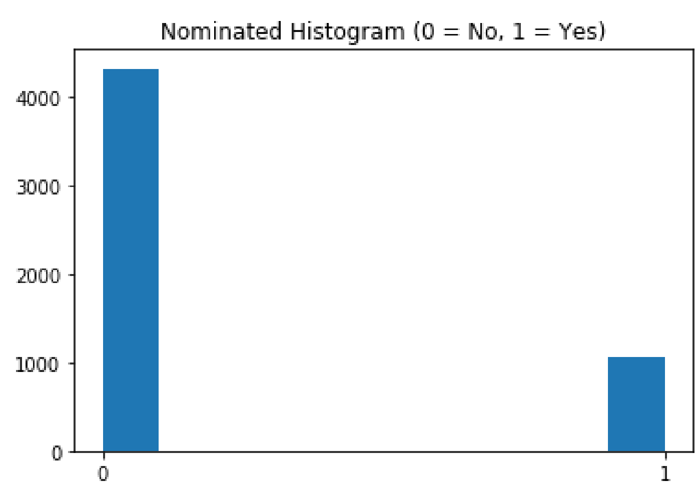

Fig. 3. Frequency of the two classes, nominated to Oscar (1), non nominated to Oscar (0)

C. *Learning considerations*

First, it will be decided the final disposition of the sample. The sample is divided in training, testing sets, giving 60% and 40% respectively to those sets.

As mentioned above, different neural network architectures will be compared by varying the number of layers and neurons per layer. Tests will also be performed for 5 values of η, $\eta = 0.01, 0.1, 0.2, 0.5, 0.9$.

IV. Results

In the following tables and figures, it can be seen the performance of the methods used to classify the dataset for each number of hidden layers.

A. One hidden layer

The most remarkable architectures for a single hidden layer are shown below. It can be noted that for a single neuron architecture in the hidden layer with a learning rate of 0.01 Figure 4, the gradients change until they reach 0. In this architecture the error does not change; Figures 5 and 6.

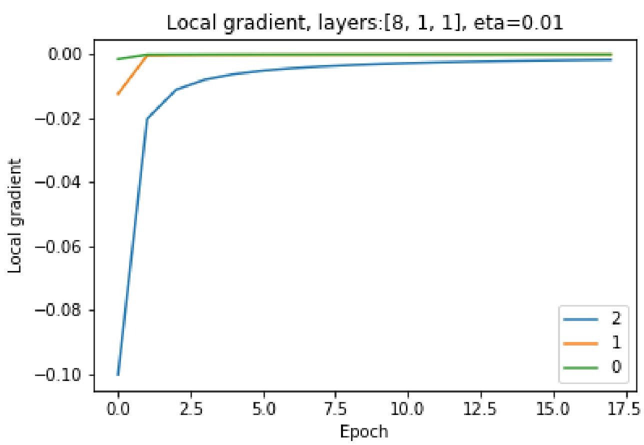

Fig. 4. Local gradient

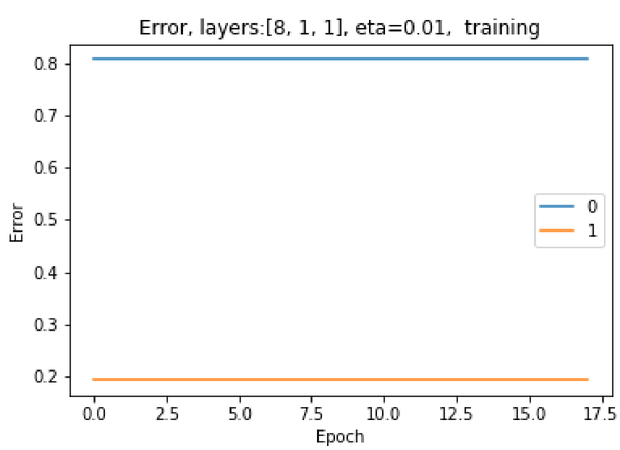

Fig. 5. Training Error

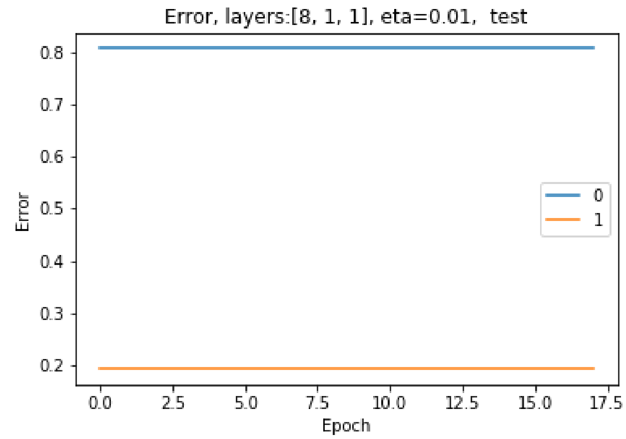

Fig. 6. Testing Error

In the other hand, for 5 neurons in the hidden layer with a learning rate of 0.1 Figure 7, the gradients reach 0 much faster and the algorithm stops running. The error does not change either for the non nominated class, but it goes to 0 for the nominated one; Figures 8 and 9.

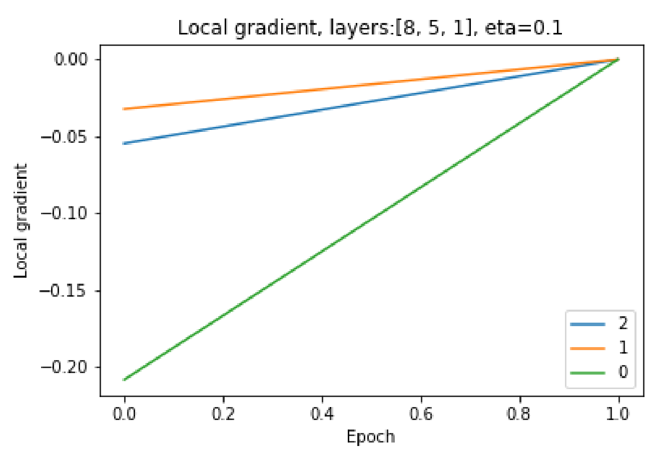

Fig. 7. Local gradient

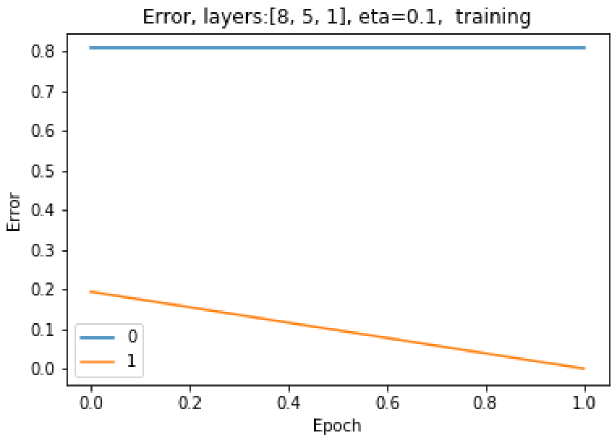

Fig. 8. Training Error

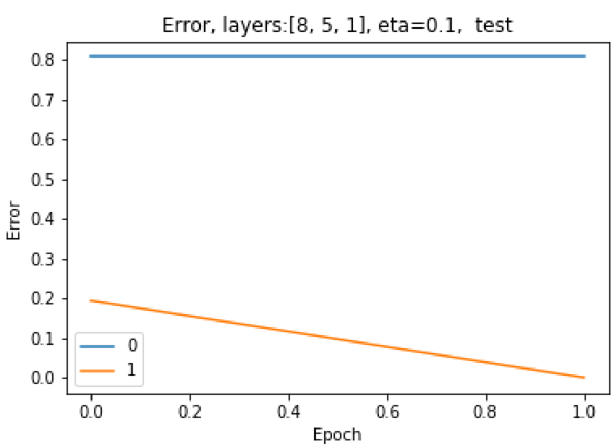

Fig. 9. Testing Error

With the same architecture, but increasing the learning rate to 0.9, the hidden and the output layer, start with a local gradient of 0, which is not good for the learning process, it does not learn at all.

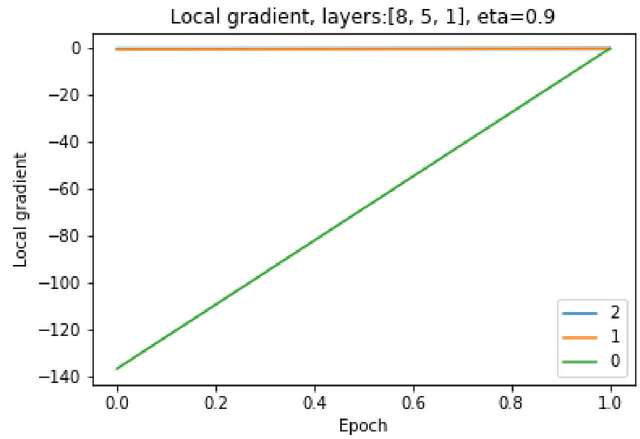

Fig. 10. Local gradient

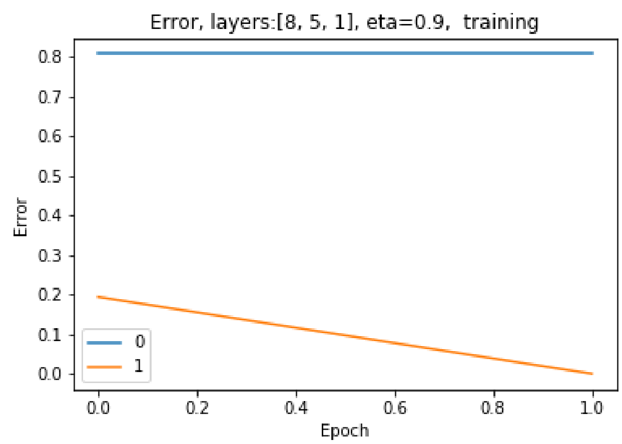

Fig. 11. Training Error

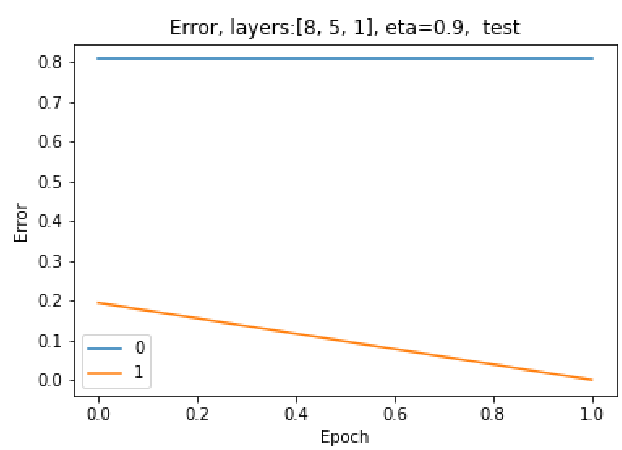

Fig. 12. Testing Error

B. Two hidden layers

In the case of two hidden layers. The architecture of a single neuron per layer and learning rate of 0.01, it can be seen the learning process for every layer until they converge, Figure 13. The training error and testing error does not changes, Figure 14 and 15.

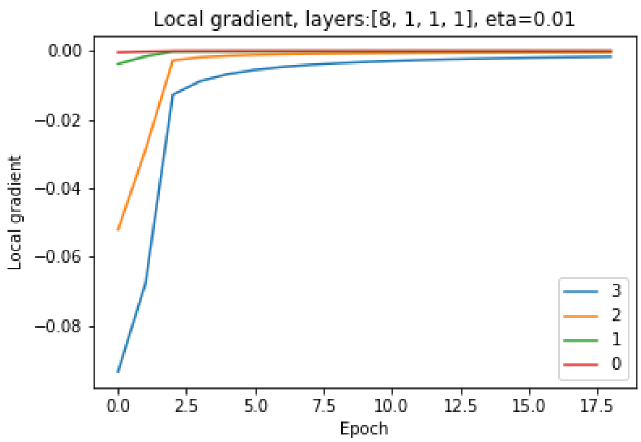

Fig. 13. Local gradient

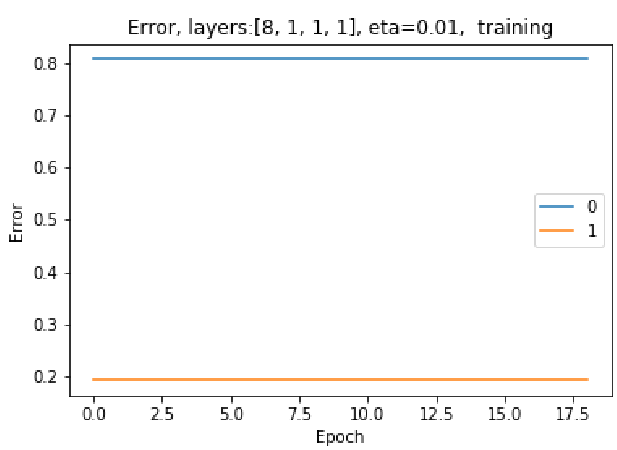

Fig. 14. Training error

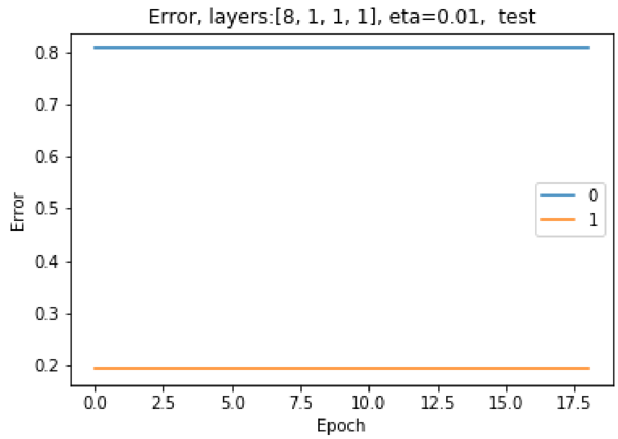

Fig. 15. Testing error

With the same architecture, but learning rate of 0.9, the algorithm does not show a learning process because of the quick convergence. And the error does not change for the non nominated class, but it goes to 0 for the nominated one; Figures 17 and 18

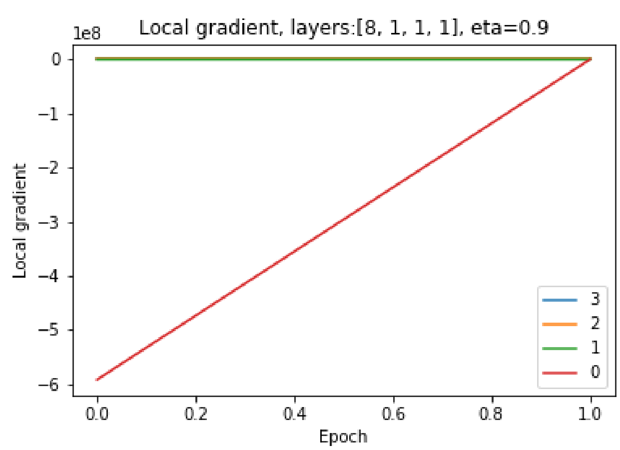

Fig. 16. Local gradient

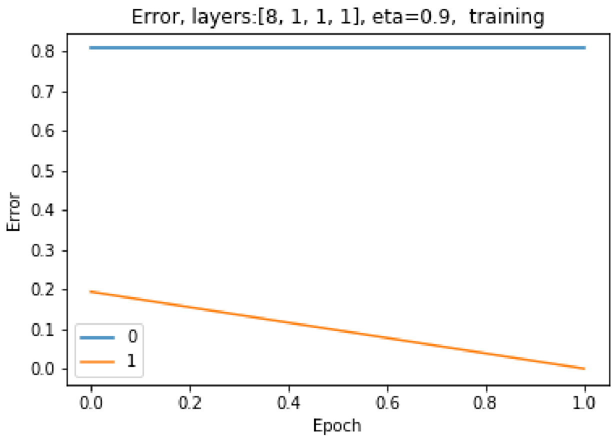

Fig. 17. Training error

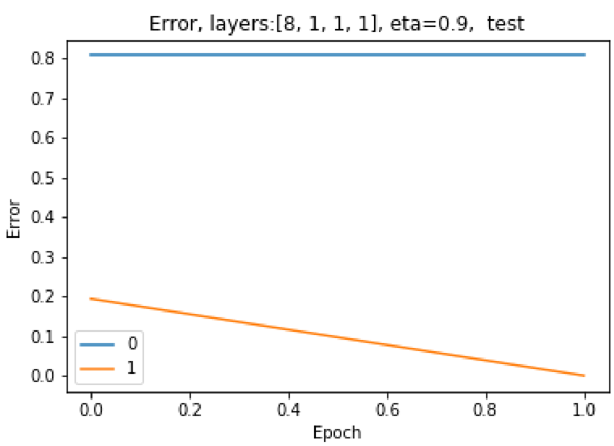

Fig. 18. Testing error

C. Three hidden layers

A similar behaviour is shown for the gradients and the errors in the architecture of three hidden layers. With one neuron per layer and 5 in the last hidden layer, with a small learning rate there is a slow and significant learning process (Figures 19 and 25); with a larger rate, there is no process and converges immediately (Figure 22).

The errors for this architectures performs the same that for two layers, Figures 20 to 27

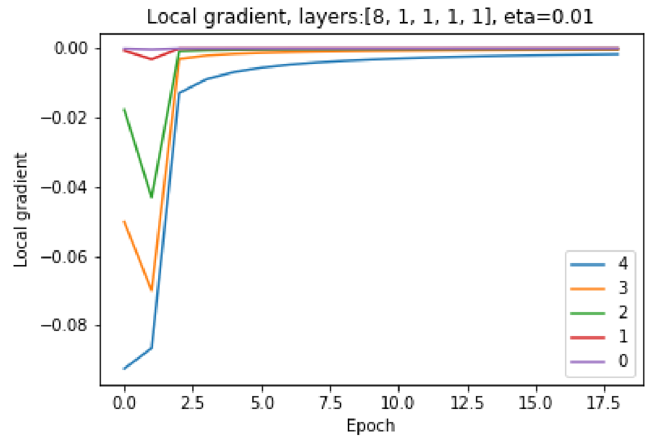

Fig. 19. Local gradient

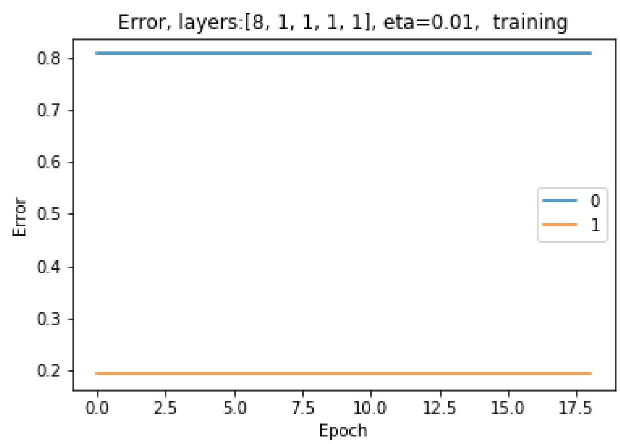

Fig. 20. Training Error

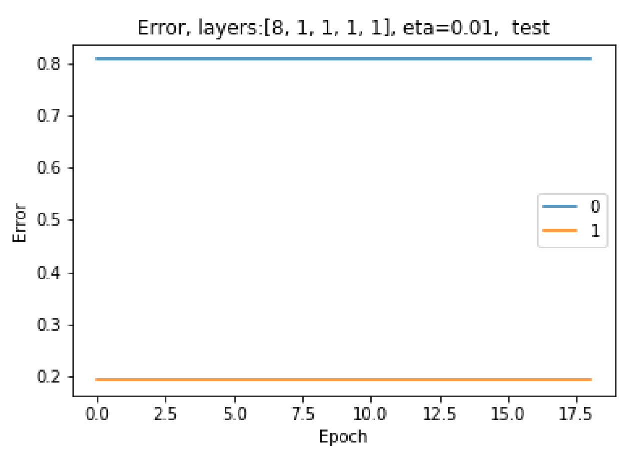

Fig. 21. Testing Error

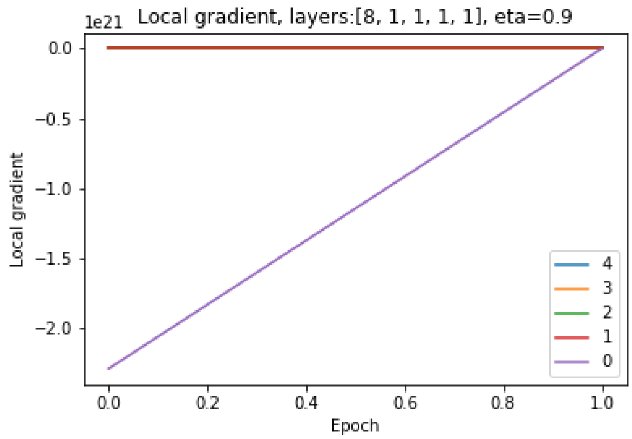

Fig. 22. Local gradient

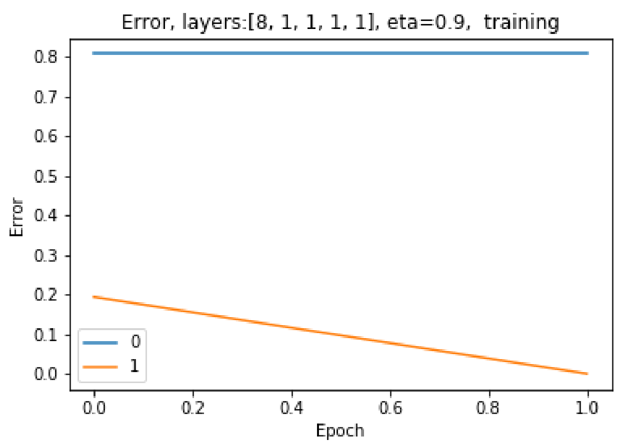

Fig. 23. Training Error

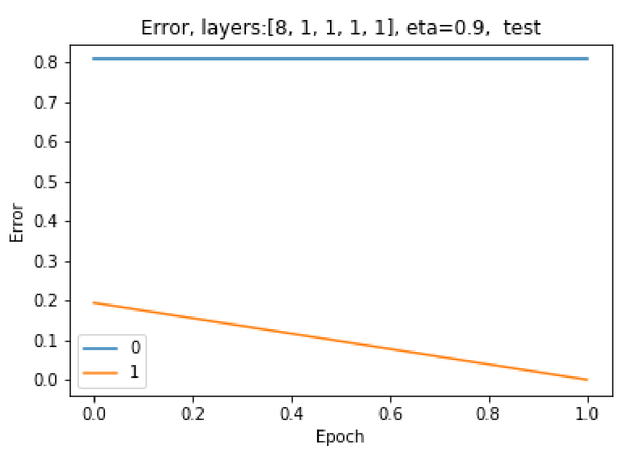

Fig. 24. Testing Error

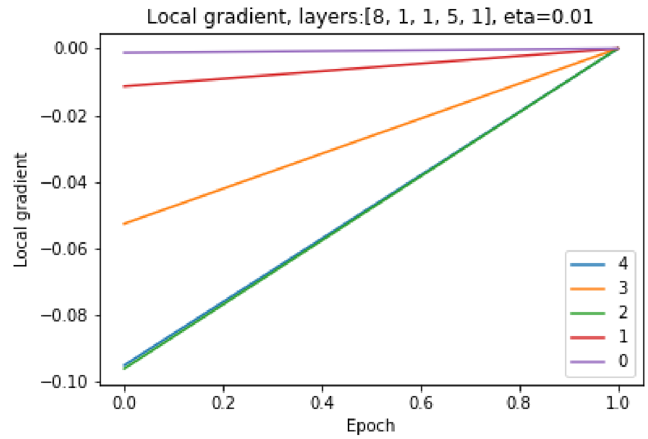

Fig. 25. Local gradient

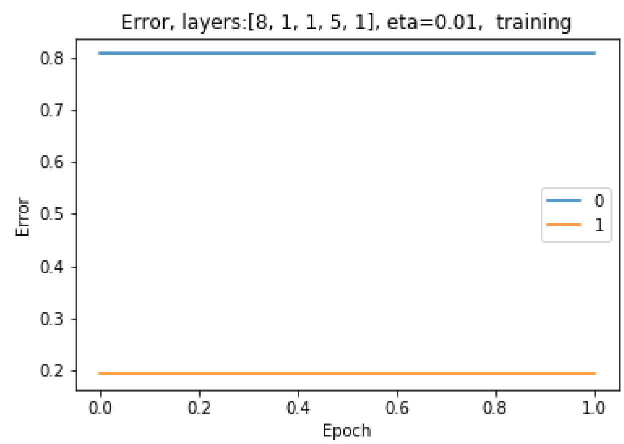

Fig. 26. Training error

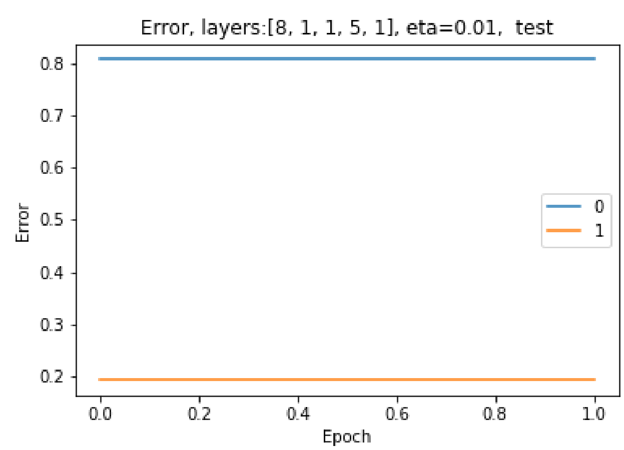

Fig. 27. Testing error

V. Conclusions

In this work we can remark that the sampling is very important for the learning process, specially with unbalanced classes. It was found that having unbalanced classes affects a lot the performance of the algorithm.

Another important finding is that, as the principle of parsimony states, a greater number of layers or neurons does not necessarily imply a better result. Also, that it is important to choose the correct learning rate, on that depends the learning process of the network, but it can be more computationally expensive.

The lack of bias in this model also affects the error because the model won´t take in count the intrinsic bias of the data.

References

[1] Hsieh, Cheng-Kang, Campo, Miguel, Taliyan, Abhinav, Nickens, Matt, Pandya, Mitkumar, JJ, Espinoza. 2018. Convolutional Collaborative Filter Network for Video Based Recommendation Systems. ArXiv.

[2] IMDb. 2021. Sort by Popularity - Most Popular Movies and TV Shows tagged by keywords.

[3] Y. J. Lim and Y. W. Teh, "Variational bayesian approach to movie rating prediction," in Proceedings of KDD Cup and Workshop, vol. 7, 2007, pp. 15–21.

[4] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in Advances in Neural Information Processing Systems, 2008, pp. 121–128.