Underfitting & Overfitting

Miguel Ángel Banda Del Valle [1*], Jose Ángel Pertuz Montes [2*]

[1, 2] Department of Engineering, Universidad Tecnológica de Bolívar, Parque Industrial y Tecnológico Carlos Vélez Pombo Km 1 Vía Turbaco, Cartagena, Colombia
[*]mbanda@utb.edu.co
[*]pertuzj@utb.edu.co

Abstract: In training algorithms, there are certain factors to keep in mind to provide results more in line with a certain prediction model, among these factors is the fact of trying to make models with too many or too few calculation parameters, which can lead to issues of overfitting or underfitting, which currently certain methods have been developed to counteract the effects that these produce in the programs. Some of these techniques and their working principles are described below.

1. Introduction

Nowadays, the amount of information that is produced and stored daily in a digital format, in social networks, educational entities, informative organizations, among others, is of great significance. The content of this information represents data of interest to the world population in general and, to a greater extent, to companies and governments. Within this same area, the emergence of Text Mining has provided tools that lead to a better interpretation in decision making based on the available information.

This field addresses problems such as categorization and grouping of texts, identification of patterns, selection of information of interest, whose solution addresses the extraction and analysis of connections between concepts.

The structuring of the content of the texts in question is achieved by means of a representation model, on which the algorithms and identification processes applied depend.

In this paper, we will describe the most used text representation models, the principles on which each one is based and the possible deviations in the resulting interpretation.

2. Regularization

When we talk about regularization, we are talking about a regression technique that is implemented with the objective of restricting or minimizing the fitted loss function to discourage the learning of a more complex model to prevent possible overfitting or underfitting. In other words, one can fit the machine learning model on a given data set and reduce the errors in it.[1]

There are two main regularization techniques:

1. Ridge regularization: which is governed by the following formula.

$$Cost\ function = Loss + \lambda * \sum \parallel w \parallel^2$$

Wherein,

$$Loss = sum\ of\ the\ squared\ residuals$$

$$\lambda = penalty\ for\ the\ errors$$

$$W = slope\ of\ the\ curve/line$$

Seeks to modify overfitted or underfitted machine learning models by minimizing and calculating the coefficients, through the cost function of the ridge regression. The larger the penalty λ, the smaller the magnitude of the coefficients.

2. Lasso Regression: which is governed by the following formula

$$Cost\ function = Loss + \lambda * \sum \parallel w \parallel$$

Wherein,

$$Loss = sum\ of\ the\ squared\ residuals$$

$$\lambda = penalty\ for\ the\ errors$$

$$W = slope\ of\ the\ curve/line$$

Like the Ridge regularization, it also seeks to modify overfitted or underfitted machine learning models by minimizing the coefficients, only instead of squaring the magnitude of the coefficients, it takes the actual values.[2]

3. Early Stopping

Early stopping appears as a solution that seeks to avoid overfitting in machine learning models. This stopping method works once the performance of the validation dataset starts to stop or in other case when it decreases with respect to the performance of the validation dataset in the previous training time, it immediately stops iterating and stops the algorithm. The gain is a maximization of the generalization power of the learning model. [3]

An advantage of early stopping is that it provides a guide to how many iterations can be allowed before the learning algorithm begins to suffer from overfitting. [4]

4. Pruning

Pruning appears as a regularization technique to avoid overfitting, in machine learning language, this technique seeks to eliminate everything that is redundant or unimportant in a search model in order to reduce the difficulty and time of inference. [5]

5. Dropout

Dropout is a technique or method applied in machine learning algorithms to prevent overfitting during the training of predictive systems, it is a regularization method in which the model is trained so that it does not learn an interdependent set of feature weights.

In simpler words, it is based on randomly eliminating layers or units in the training network, which translates into an adaptability that will not be excessive on the part of certain layers or units, in order to later compare predictions of these poorly adapted units with the predictions made by the layers that were not eliminated, i.e., those that were trained. Therefore, these algorithms require a greater number of iterations to achieve their objective.

Basically, this method is based on a principle of co-adaptation, which provides robustness to the training, because the layers in the process of adaptation are correcting errors of previous layers. It is a model to reduce overfitting, which is noisy and in which probabilities of greater or lesser responsibility are assumed for the inputs that result in each stage of the process.

Some studies show that Dropout improves the performance of neural networks in supervised vision tasks, computational biology, document classification, result retrieval, speech recognition, among others.

6. Regularize the weights

In many cases of algorithm training, the programs associate input weights to the outputs and in certain occasions the existence of a single range of weights produces instability in the algorithm's objective, meaning an overadjustment of the algorithm in which inputs with small changes can cause large changes in the results.

The weight is understood as the level of specifications that the program can understand, so that if it is at an excessive level, the training is over-fitting, requiring the same level of specifications for all inputs.

The method of regularizing the associated weights, based on keeping them small, is used as a technique to correct the overfitting and provide better generalization of the data in question by changing the loss calculation used in the optimization of the program.

The algorithm finally uses weights smaller or no larger than necessary to provide good performance.

7. References

[1] P. Gupta. "Regularization in Machine Learning". Medium. https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a (Accessed October 2021).

[2] Simplilearn. "Regularization in Machine Learning || Simplilearn". Simplilearn.com. https://www.simplilearn.com/tutorials/machine-learning-tutorial/regularization-in-machine-learning (Accessed October 2021).

[3] "What is "early stopping" in machine learning?" Artificial Intelligence Stack Exchange. https://ai.stackexchange.com/questions/16/what-is-early-stopping-in-machine-learning [Accessed October 2021].

[4] "ⓘ Early stopping. In machine learning, early stopping is a for". en.google-info.org. https://amp.en.google-info.org/213214/1/early-stopping.html [Accessed October 2021].

[5] "What Is Pruning In ML/AI?" Analytics India Magazine. https://analyticsindiamag.com/what-is-pruning-in-ml-ai/ (accedido el 2 de octubre de 2021).

[6] *A Gentle Introduction to Dropout for Regularizing Deep Neural Networks*. (s. f.). Machine Learning Mastery. https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/

[7] *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. (s. f.). Journal of Machine Learning Research. https://jmlr.org/papers/v15/srivastava14a.html

[8] Budhiraja, A. (2016, 15 de diciembre). *Learning Less to Learn Better—Dropout in (Deep) Machine learning*. Medium. https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5