International Journal of Advance in Computer Science & its application – IJCSIA 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 7 : Issue 2 [ISSN : 2250-3765] - Publication Date: 25 June , 2018

Regularization in the Process of Developing an Artificial Neural Network

Imanol Bilbao, Javier Bilbao

***Abstract*—The process for classifying a set of data can depend on several variables, which can have a non-very direct relation among them. Using mathematical techniques such as regression is one of the most accepted methods. Moreover, in the last time and when number of data is higher and concepts like deep learning are applied, artificial neural networks (ANN) are taking into account as a method to solve these classification systems. But, when these ANN are used, some problems must be resolved in order to obtain good results. Some of these problems are overfitting and underfitting. In this paper, an approach to the resolution of them by means of regularization is dealt with.**

***Keywords*—artificial neural networks, linear regression, logistic regression, regularization, overfitting, underfitting**

I. Introduction

An ANN is an independent computational element set (neurons), totally interconnected to each other, that work autonomously but synchronously with the other elements.

Each neuron receives impulses from other neurons and gives them a certain importance or specific weight. After that, the neuron transmits the sign to other neurons, or even to itself (feedback). Restrictions on the number and the connections among neurons limit the type and application field of the ANN.

In order to a correct work of a neural network, that is, to give the correct outputs from a certain input set, the ANN must learn by means of training the guidelines of the necessary calculations to be performed. Normally, these process is based on examples or learning pattern.

The training of a neural network is the modification of the weights in order to achieve that each neuron gives the correct answer or output, in all situations that it has to learn.

Imanol Bilbao
University of the Basque Country (UPV/EHU)
Spain

Javier Bilbao
University of the Basque Country (UPV/EHU), Engineering School, Applied Mathematics Department
Spain
javier.bilbao@ehu.eus

It is proved that a multilayer feedforward artificial neural network with one or more than one hidden layer is enough to approximate any non-linear continuous function in a closed interval, provided enough neuron exists in each layer [1]. For example, load flow resolution is a non-linear problem, and therefore, theoretically it may be solved by means of artificial neural networks (ANN). In the fields of load flow and of optimization of load flow some satisfactory tests have been implemented [2], [3]. In these tests the application of ANN improves the results obtained by conventional algorithms of load flow.

The own makeup of the neural networks is not very clear to determinate the connection between inputs and outputs, and the incorporation to the model of empiric knowledge is an arduous task. So, although it is true that the computational speed may be increased and changes in data may be adapted, it is difficult to find a training method that guarantees a total convergence of the ANN in the all cases that the network can study. Moreover, the learning time and the number of necessary patterns that we have to give to the ANN in its training are factors that are not optimized enough [4].

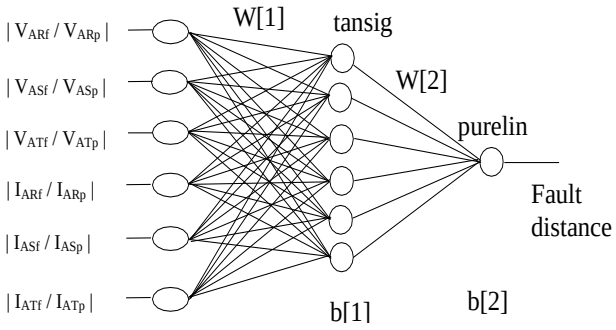

Figure 1. ANN structure example.

The back-propagation learning rule is used in perhaps 80% to 90% of practical applications. Improvement techniques can be used to make back-propagation more reliable and faster. Various improvement techniques were applied to the different network architectures tested, and it was concluded that the most suitable training method for the architecture selected was the back-propagation method based on the Levenber-Marquardt optimisation technique. This technique is more sophisticated than the gradient descent used in the back-propagation technique. [5]

International Journal of Advance in Computer Science & its application – IJCSIA 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 7 : Issue 2 [ISSN : 2250-3765] - Publication Date: 25 June , 2018

II. Regression models

When we want to evaluate the relationship between a variable that for us gives rise to special interest (dependent variable that is usually called y) with respect to a set of variables (independent variables, which are called x1, x2, ..., xn), hypothesis tests normally do not give us enough information about the overall relation of all of them, given that the typical contrasts of hypotheses are based on relations of 2 variables, where the possibility of other variables of interest is not taken into account and where the meaning of the relationship is bidirectional. This is when the application of the regression models is appropriate and convenient. The regression models allow to evaluate the relationship between a variable (dependent) with respect to other variables as a whole (independents). The regression models are expressed as follows:

$$y = f(x_1, x_2, ..., x_n) + \varepsilon \quad (1)$$

The main objective of creating a regression model can be, for example, to evaluate how the change in certain characteristics (independent variables) affects another particular characteristic (dependent variable), called the explanatory model; or our objective could be to try to estimate or approximate the value of a characteristic (dependent variable) in function of the values that can take together another set of characteristics (independent variables), called then model for predictive purposes.

There are several options for estimating a regression model, and we can stand out, because the ease of application and interpretation, the linear regression model and the logistic regression model. Taking into account the type of variable that we want to estimate (dependent variable or response) we will apply a regression model or the other one. Simply put, when the dependent variable is a continuous variable, the most frequently used regression model is linear regression; whereas when the variable of interest is dichotomous (that is, it takes two values such as yes/no, male/female), logistic regression is normally used.

Regression models, normally linear or logistic regression as we have said, can be used for two purposes:

1) prediction, when the researcher's interest is to predict in the best possible way the dependent variable, using a set of independent variables, and

2) estimation, when the interest is focused on estimating the relation of one or more independent variables to the dependent variable.

The result of a predictive model is the model itself, while in an estimative model it is the estimation of the coefficient of the variable of interest. The second case is the most frequent in complex studies where it is tried to find determinant factors of a process.

At the present, neural networks are widely known for use in machine and deep learning and also in modelling complex problems such as image recognition. In this way, it is possible to train a neural network to perform classification of some data or regression. Artificial neural networks are easily adapted to regression problems. Any type of statistical model can be resemble a neural network if these ANN use adaptive weights and can approximate non-linear functions of their inputs. Therefore, neural network regression is adapted to problems where traditional regression models do not fit appropriately a good solution.

III. Overfitting and underfitting

It says that there is overfitting when the error of the proposed curve referred to the data is zero or almost zero, that is, the proposed curve fits all or almost all data with zero error. It seems that it would be perfect, but we have to take into account that the shape of the proposed curve can be of different types and, what is more frequent, it is possible that our data have some noise. So, if our curve is too good that fits all our data without any error we must contemplate the possibility that the proposed curve is capturing the noise of the data. In neural networks, sometimes overfitting happens when the model shows low bias but high variance. Normally, overfitting is a result of an excessively accurate or complicated model, and it can be avoided by fitting several models and using techniques as validation or cross-validation in order to compare their predictive accuracies on test data.

It says that there is underfitting when the error of the proposed curve referred to the data is enough high in several of the values of our data or the average of the error of the whole curve is high. In this case, we say that the proposed curve cannot capture the underlying trend of our data (the curve does not fit the data as we expected or wanted or needed) and we must find another solution for fitting the data. In order to compare the predictive accuracies of the model, techniques as validation or cross-validation are very useful also to avoid underfitting. In neural networks, sometimes underfitting happens when the model shows low variance but high bias. Normally, underfitting is a result of an excessively simple model.

Both overfitting and underfitting lead to poor predictions on new data sets.

A. *Overfitting*

Both in the case of linear regression and logistic regression, we can have hypotheses that fit perfectly or very well to the training data but do not reflect well the trend of the model, or perhaps fail to generalize to new examples. This often happens when we have a high number of input parameters which results in very complicated functions with many unnecessary curves and angles.

In the case of linear regression, if we are trying to relate, for example, the manufactured units of one product with their price, we could happen to the following with a high number of parameters. We could obtain the next model:

International Journal of Advance in Computer Science & its application – IJCSIA 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 7 : Issue 2 [ISSN : 2250-3765] - Publication Date: 25 June , 2018

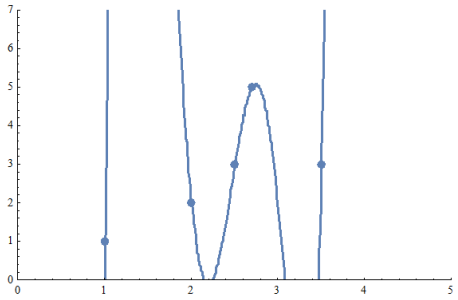

Figure 2. Overfitting effect.

When in fact we would prefer a model more similar to

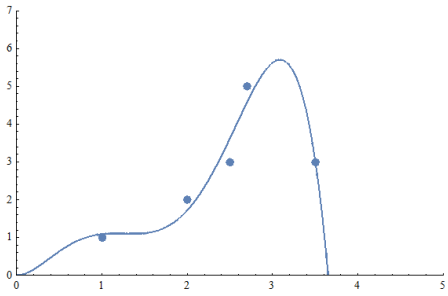

Figure 3. Smooth regression.

In the case of logistic regression this can also happen to us. With too many parameters we would get

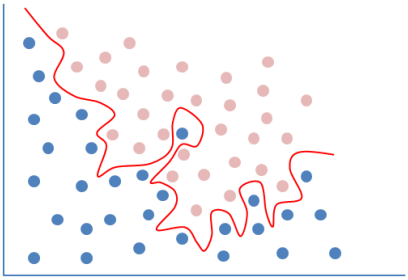

Figure 4. Overfitting effect with several data.

When in reality we would prefer a model like

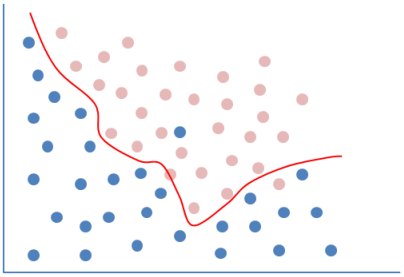

Figure 5. Appropriate regression with several data.

In the case of facing an overfitting problem we can reduce the number of parameters manually, analyzing which are more important and, therefore, will preserve, and which seem of secondary level and we can eliminate. We can also try to do this automatically using some technique like PCA. Unfortunately, in this way we will always be losing information.

Another possible option for solving it is to use the regularization technique while maintaining all variables. This technique works well when we have many input parameters and each contributes "a little" in the prediction, because what it tries to decrease the magnitudes of all the weights in the ANN.

B. Underfitting

In the case of linear regression as well as logistic regression we can have hypotheses that fit very poorly to the training data and that consequently they will also do with new examples. This often happens when we have a very poor number of input parameters which results in functions that are too simple that do not estimate the data well.

In the case of linear regression, if we are trying to relate, for example, the manufactured units of one product with their price, we could happen to the following with an insufficient number of parameters. We could get a model like:

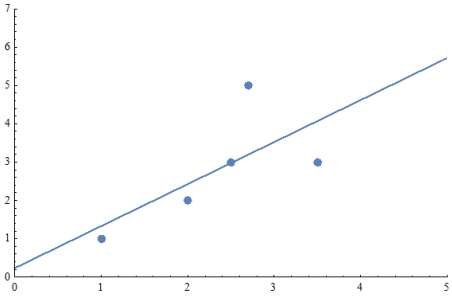

Figure 6. Underfitting effect.

When in fact we would prefer a model more similar to

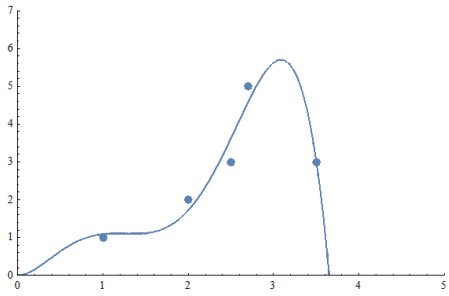

Figure 7. Smooth regression.

In the case of logistic regression this can also happen to us. With too many parameters we would get

International Journal of Advance in Computer Science & its application – IJCSIA 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 7 : Issue 2 [ISSN : 2250-3765] - Publication Date: 25 June , 2018

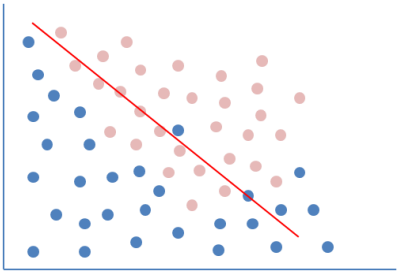

Figure 8. Underfitting effect with several data.

When in reality we would prefer a model like

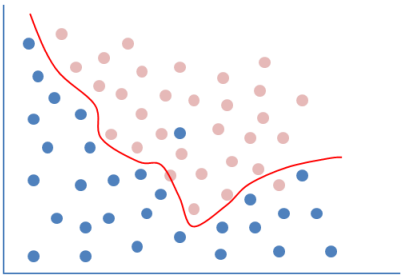

Figure 9. Appropriate regression with several data.

IV. Regularization

If we have an overffiting problem in our hypothesis function and we know which parameters should be less important, we can try to reduce the weight of those terms by increasing the cost of the magnitude of the weight θj associated with those parameters. Thus, the process of minimizing the cost function would try to minimize the magnitude of the θj that should influence less.

Generalizing this idea, making the different θj small or close to zero, it helps to have simpler hypotheses and with less angulations and, therefore, less propitious to overffiting. This is especially true if the contribution to the prediction of the different input parameters is similar.

It is impossible to know in advance which parameters may be more relevant in our hypothesis and in what proportion, so in regularization we will treat all parameters equally, trying to obtain the smallest possible values of θj and thus reduce the influence of its associated parameters. Since the term θ_0 is not associated with any parameter, it is not usually regularized.

With this purpose, we add to the cost function that we are going to use an extra regularization term, an additional cost associated with the magnitude of the weights, in the form of $\lambda\sum_{j=1}^{n}\theta_j^2$. This term will produce, when calculating the gradient, an extra term in each component of the gradient of $\lambda\cdot\theta_j$ (from j = 1 to n, the first term θ_0 will not be modified), and will thus also minimize each θj (from j = 1 to n, without taking into account the first term θ_0) [7].

λ is a positive parameter called regularization parameter. It controls the influence of regularization on the whole process, controlling how much importance is given to a good adjustment of training data, and how much importance to minimize the different θj. If λ = 0 , there will be no regularization. As λ takes on larger values, the influence of regularization will increase, resulting in smoother curves.

An excess in the value of λ may lead to hypotheses too soft, and even of constant value, which will end up incurring underfitting, so some care is needed when choosing this parameter.

A. *Regularized linear regression*

In the case of linear regression the cost function with the regularization term would be,

$$J(\vec{\theta}) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(\vec{\theta}\vec{x}^{(i)} - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right] \qquad (2)$$

which would modify the terms of the gradient in

$$\frac{\partial}{\partial\theta_j}J(\vec{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\left(\vec{\theta}\vec{x}^{(i)} - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}\theta_j \qquad (3)$$

except for the case of j = 0 that would not be modified.

This causes the weights to be updated at each step using

$$\theta_j = \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(\vec{\theta}\vec{x}^{(i)} - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}\theta_j\right] =$$

$$= \theta_j\left(1 - \alpha\frac{\lambda}{m}\right) - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(\vec{\theta}\vec{x}^{(i)} - y^{(i)}\right)x_j^{(i)} \qquad (4)$$

The regularization term produces an effect of multiplying: $\theta_j\left(1-\alpha\frac{\lambda}{m}\right)$. As the subtraction is usually less than unity, the effect of this multiplication is, as discussed above, to decrease the weight θj in each step before applying the second term in conventional form.

From the point of view of the normal equations we can interpret it as the inverse process, by first decreasing the values of the weights by the result of pre-multiplying first $\vec{\theta}$ by λL, where $L = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$ (similar to the unit

matrix of (n + 1) × (n + 1) but with the first element null), so that $\vec{\theta}_{new} = \vec{\theta} + \lambda L\vec{\theta}$. Thus, the normal equation would be:

$$X^T\left(X\vec{\theta}_{new}\right) = X^T\vec{y} \implies (X^TX + \lambda L)\vec{\theta} = X^T\vec{y} \qquad (5)$$

The matrix $X^TX + \lambda L$ is always invertible, even though the matrix X^TX is not, so the solution to this equation is always unique and we can find it in a single step by means of:

$$\vec{\theta} = (X^TX + \lambda L)^{-1}X^T\vec{y} \qquad (6)$$

v. Conclusions

Artificial neural networks are created before the advent of computers and they are capable of learning after a process of training. In the fields like recommendation systems and machine learning, neural networks are used with the aim of classification. The process usually involves methods of regression, typically linear or logistic, which has to solve some problems such as overffiting or underfitting. In this paper a solution of regularization is presented.

References

[1] Suykens J.A.K., Vandewalle J.P.L., De Moor B.L.R., Arificial Neural Networks for Modelling and Control of Non-Linear Systems, Kluwer Academic Publishers, 1996.

[2] Nguyen T.T., Neural Network Load Flow, IEE Proceedings Generation, Transmission and Distribution, Vol.142, No.1, 1995, pp. 51-58.

[3] Nguyen T.T., Neural Network Optimal Power Flow, Proceedings of the Fourth International Conference on Advances in Power System Control, Operation & Management, IEE Conf. Publ. 450, 1997, pp. 266-271.

[4] J. Bilbao, E. Bravo, M. Rodríguez, O. García, C. Varela, P. González, N. M. Tabatabaei, Neural Networks for Load Flow, Scientific Bulletin of the University of Pitesti, Series Electronics and Computer Science, number 8, vol. 2, 2008, 1-6.

[5] A.J. Mazon, I. Zamora, J. Gracia, J. Bilbao, J.R. Saenz, Falneur: ANN based software to fault location in electrical transmission lines, IASTED International Conference on Applied Informatics 2001

[6] A. Bluman, Elementary Statistics, A Brief Version, Ed. Cram 101 Textbook Reviews, 2017.

[7] A.Y. Ng, Preventing Overfitting of Cross-Validation Data, Carnegie Mellon University, Stanford Artificial Intelligence Laboratory, Stanford University, http://ai.stanford.edu/~ang/papers/cv-final.pdf,

About Author (s):

Imanol Bilbao obtained the master in Telecommunications Engineering from University of the Basque Country, Spain, in 2006. He is also Master of Advanced Study (MAS) in Distributed generation, renewable energy and power system by the same university. At present he is lecturer at the department of Electric Engineering of that university.
He has additional training in Neural Networks for Machine Learning by University of Toronto, Machine Learning by Stanford University, etc.

Javier Bilbao obtained the degree in Electrical Engineering from University of the Basque Country, Spain, in 1991. At present he is Ph.D. in Applied Mathematics and professor at the department of Applied Mathematics of that university.
He has been General Chairman of some conferences of WSEAS organization. Current and previous research interests are: Distribution overhead electrical lines compensation, Optimization of series capacitor batteries in electrical lines, Modelization of a leakage flux transformer, Losses in the electric distribution Networks, Artificial Neural Networks, Modelization of fishing trawls, E-learning, Noise of electrical wind turbines, Light pollution, Health risk of radiofrequencies.
Prof. Bilbao is the General Chairman of the International Conferences on Engineering and Mathematics (ENMA) and member of the committees of the Technical and Physical Problems of Power Engineering (TPE) International Conferences.