

Performance Functions Alternatives of Mse for Neural Networks Learning

Mohamed M. Zahra,
Electrical Engineering Department,
Al-Azhar University- Cairo- Egypt,

Mohamed H. Essai,
Electrical Engineering Department,
Al-Azhar University-Qena- Egypt,

Ali R. Abd Ellah,
Electrical Engineering Department,
Al-Azhar University-Qena- Egypt,

Abstract

Recently multilayer feed-forward neural networks are often used in several fields, as industrial modeling, universal function approximations, and as classifiers. These supervised neural networks are commonly trained by a traditional backpropagation learning algorithm, which minimizes the mean squared error (Mse) of the training data. All previous efforts has been exerted to find alternatives of Mse in the presence of outliers (noisy data), however Mse is not robust in presence of outliers that may pollute the training data. For first time we aim in our paper to present M-Estimators as performance functions alternatives of Mse Performance function in the case of using high quality clean data. We compared between Mse and M-estimators in two applications crab classification, and function approximation.

KeyWords -- Robust Statistics, Feed-Forward Neural Networks, M-Estimators, Classification, Function Approximation.

1. Introduction

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the connections between elements largely determine the network function. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Neural networks have been trained to perform complex functions in various fields, including pattern recognition, system identification, function approximation, classification, speech recognition, computer vision, and control systems. Neural networks can also be trained to solve problems that are difficult for conventional computers or human being.

Feed-forward neural networks are commonly trained by the traditional back propagation.

It is common to use the back propagation learning algorithm based on the minimization of the mean square error Mse for the training data. The use of Mse in data modeling is commonly known as the

least mean squares LMS method. The basic idea of LMS is to optimize the fit of a model with respect to the training data by minimizing the square of residuals. Mean squared error Mse is the preferred measure in many data modeling techniques. Tradition and ease of computation account for the popularity of Mse.

Our main idea is to find alternative performance functions (cost function) instead of Mse performance function in order to optimize the neural networks training in case of high quality clean data in other word non-corrupted data (outliers free). We will exploit a family of robust statistics estimators called M-estimators as alternatives.

Recently many researches exploited M-estimators in order to robustify the NN learning process [2],[3], in the presence of contaminated data. However, they did not study the performance of these robust M-estimators in noise free (clean) data.

The objective of our contribution is to introduce M-estimators for first time as alternatives of Mse performance function in the case of using trusted clean data in the learning process.

The outline of this paper is as follows. Section (2) presents M-estimator as alternative performance function to Mse, and shows some common M-estimators. Section (3) back propagation learning algorithm based M-estimators. Section (4) discusses the function approximation by neural networks. Section (5) discusses the classification. Section (6) gives our experimental results by comparing the performance of various M-estimators and Mse in terms of accuracy in case of clean data.

2. M-Estimators

M-estimators have gained popularity in the neural networks community[6]. Let r_i be the residual of the i th datum, i.e. the difference between the i th observation and its fitted value. The standard least-squares method tries to optimize the training data by minimize

$\sum_i r_i^2$ but The M-estimators try to minimize the error by replacing the squared residuals r_i^2 by another function of the residuals, yielding

$$\text{Min } \sum_i \rho(r_i) \quad (1)$$

Where $\rho(\cdot)$ is a symmetric, positive-definite

function with a unique minimum at zero, and is chosen to be less increasing than square. Table 1, lists a few commonly used M-estimators and their influence functions. M-estimators' influence functions can be illustrated graphically in Fig. 1, and Fig. 2

Table 1: Some commonly used M-estimator

Type	$\rho(r)$	$\psi(r)$	$\omega(r)$
L2	$r^2 / 2$	r	1
L1	$ r $	$\text{Sgn}(r)$	$\frac{1}{ r }$
Fair	$c^2 \left[\frac{ r }{c} - \log\left(1 + \frac{ r }{c}\right) \right]$	$\frac{r}{1 + r /c}$	$\frac{1}{1 + r /c}$
Huber	$\begin{cases} r^2/2 & \text{if } r \leq k \\ k(r - k/2) & \text{if } r \geq k \end{cases}$	$\begin{cases} r & \\ k \cdot \text{sgn}(r) & \end{cases}$	$\begin{cases} 1 & \\ k/ r & \end{cases}$
Cauchy	$\frac{c^2}{2} \log(1 + (r/c)^2)$	$\frac{r}{1 + (r/c)^2}$	$\frac{1}{1 + (r/c)^2}$
Geman-McClure	$\frac{r^2/2}{1 + r^2}$	$\frac{r}{(1 + r^2)^2}$	$\frac{1}{(1 + r^2)^2}$
LMLS	$\log(1 + \frac{1}{2}r^2)$	$\frac{r}{1 + \frac{1}{2}r^2}$	$\frac{1}{1 + \frac{1}{2}r^2}$

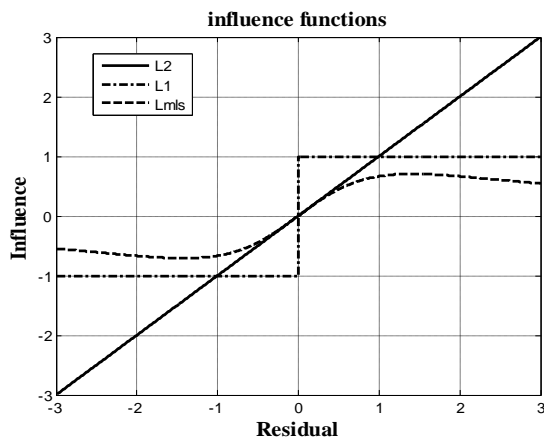


Figure 1: The influence functions for L2, L1 and LMLS estimators

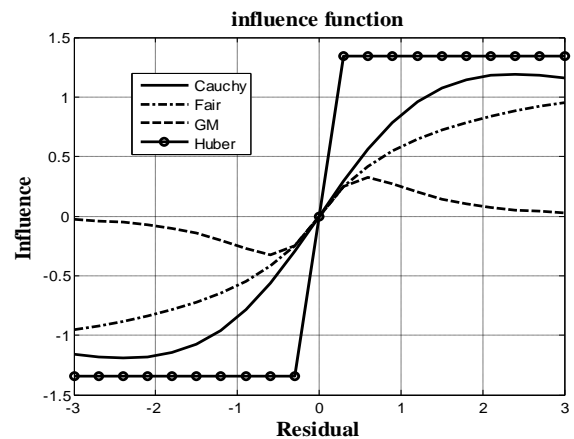


Figure 2: The influence functions for Cauchy, Fair, GM and Huber estimators.

3. Backpropagation Learning Algorithm Based M-Estimators

To implement the traditional learning algorithm based on M-estimators concept, all want to do is replacing the squared residuals r_i^2 by another function of the residuals, yielding

$$E = \sum_i \rho(r_i) \tag{2}$$

Where ρ is asymmetric, positive definite function with a unique minimum at zero, and is chosen to be less increasing than square.

4. Function Approximation Using Neural Networks

Numerous engineering problems in signal processing, computer vision, and pattern recognition can be abstracted into the task of approximating an unknown function from a training set of input-output pairs. It is hypothesized that the input vector and the output vector are related by an unknown function f such that

$Y = f(x) + e$ The output noise deviation (e) is a random vector due to the imprecise measurements made by physical devices in real world environments.

The function approximation task can be summarized as to find an estimator \hat{f} of f such that some metric of approximation error is minimized [8].

5. Classification

Classification is a multivariate technique concerned with data cases (i.e. observations) assigning [5], [7] to one of a fixed number of possible classes (represented by nominal output variables). The goal of classification is to sort observations into two or more labeled classes. The emphasis is on deriving a rule that can be used to optimally assign new objects to the labeled classes.

In statistics, where classification is often done with logistic regression or a similar procedure, the properties of observations are termed explanatory variables.

A large number of input variables can present severe problems for pattern recognition systems. One technique to alleviate such problems is to combine input variables together to make a smaller number of new variables called features.

In the terminology of pattern recognition, classifications are known as the training set and future cases form the test set and our primary measure of success is the error or (misclassification) rate.

Classification problems can be seen as particular cases of function approximation, where for classification problems the functions which we seek to approximate are the probabilities of membership of the different classes expressed as functions of the input variables. Many of the key issues which need to be addressed in tackling pattern recognition problems are concerned to classification

6. Simulation Results

In this section, the performance of feed-forward neural networks (FFNN) trained with back propagation learning algorithm that uses M-estimators as performance functions, and of FFNN trained with back propagation learning algorithm that uses tradition Mse performance function are evaluated in two different applications mentioned above (function approximation and classification).

6.1. Crab classification

Neural networks introduced as proficient classifiers and are particularly well suited for addressing non-linear problems. Given the non-linear nature of real world phenomena, like crab classification, neural networks is certainly a good candidate for solving the problem.

In this section we attempt to build a classifier that can identify the sex of a crab from its physical measurements. Six physical characteristics of a crab are considered: species, frontal lip, rear width, length, width and depth [9].

For comparison constructed classifier each time will be trained using one of M-estimators performance functions, and traditional Mse performance function.

The six physical characteristics will be organized as input matrix to a neural network where i^{th} column of this matrix contains six elements representing crab's features (species, frontal lip, rear width, length, width and depth), and the sex of the crab will be organized as target matrix, where each corresponding column of the target matrix will have

two elements. Female crabs are represented with a one in the first element, male crabs with a one in the second element. Given an input, matrix, the neural network then will be tuned to produce the desired target outputs (process of neural network training). After this process it is expected that NN will have ability to identify if the crab is male or female [9].

6.1.1. Crab classification results

The classification performances of the classifiers trained using candidated M-estimators, and traditional Mse-performance functions given in Table 2.

It is clear that classifiers trained using both Cauchy, Fair, and GM performance functions have identical percentage of correct classification as Mse-performance function, while LMLS (Least Mean Log of Squares) has percentage of correct classification equal to 96.7%, which is not far from others. Both L1 and Huber have less percentage of correct classification in comparison with others.

Table 2: Mse and M-estimators comparison

Performance Function	Percentage of correct classification
MSE	100%
CAUCHY	100%
FAIR	100%
LMLS	96.7%
GM	100%
L1	80%
HUBER	80%

6.2. Function approximation

In this section, the proper performance of neural networks trained with M-estimators, and traditional Mse performance functions was tested to approximate the function

$$y = |x|^{2/3} \quad (3)$$

This example is proposed in [1],[2],[3],[4]. The neural network architecture considered is a two layer feed-forward with ten hidden neurons. A total of 501 training patterns were generated by sampling the independent variable in the range [-2, 2], and using Eq(3) to calculate the independent variable.

6.2.1. Result

To compare the performances of all above mentioned performance functions, we use root mean square error (RMSE) of each model,

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (t_i - y_i)^2}{N}} \quad (4)$$

Where the target t_i is the actual value of the function at x_i and y_i is the output of the network given x_i as its input.

The neural networks trained with high quality clean data for 500 epochs. The results presented below are the average response of trainings. This was done to take into account the different initial values of weights and bias at the beginning of each training.

Table.3, shows RMSE values for all mentioned performance functions. It is clear from tabulated results that, both LMLS, Cauchy, and GM performance functions, have approximately semi equal RMSE values with Mse one. In this case Huber performance function, provides so poor performance in comparison with others.

Table 3: Mse, and M-estimators RMSE comparison.

Performance function	RMSE
MSE	0.0104
LMLS	0.0106
L1	0.0156
FAIR	0.0132
CAUCHY	0.0107
GM	0.0117
HUBER	0.6121

Conclusion

In this paper we introduced a family of robust statics M-estimators as alternative performance functions of Mse one. It is well known that this family provided high reliability for robust NN training in the presence of contaminated data. Based on the mentioned above result we recommend this family of estimators as a good alternative of Mse performance function, in the presence of high quality clean data too.

References

- [1] A. V. Pernia-Espinoza, J. B. Ordieres-Mere, F. J. Martinez de Pison, and A. Gonzalez-Marcos "Tao-robust backpropagation learning algorithm" Neural Networks, vol. 1, pp. 1–14, 2005.
- [2] M.T El-Melegy, M. Essai and A. Ali, "Robust training of Artificial feedforward neural networks", Springer, vol. 1, pp. 217–242, Jun. 2009.
- [3] M.T El-Melegy, " RANSAC Algorithm with Sequential Probability Ratio Test for Robust Training of Feed-Forward Neural Networks", IEEE, International Joint Conference on Neural Networks (IJCNN), pp 3256-3263, July 31 - August 5-2011

- [4] Andrzej Rusiecki, " Robust Learning Algorithm Based on Iterative Least Median of Squares", Springer, pp 145-160, 15-may-2012
- [5] Bishop, C.M. (1995) Neural Networks for Pattern Recognition. Oxford: Clarendon Press
- [6] Zhengyou Zhang. Parameter Estimation Application Techniques :A Tutorial with to Conic Fitting, Oct-1995
- [7] Ripley, B.D., (1996) "Pattern Recognition and Neural Networks. Cambridge: Cambridge University press"
- [8] Sangit, Chatterjee Matthew Laudato, "Statistical Applications of Neural Networks" 1995
- [9] Mark Hudson Beale, Martin T. Hagan, HowardB. Demuth, Neural Network Toolbox™ 7 User's Guide