ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 167-174

Control of complexity in learning with perturbed inputs

Grandvalet Yves, Canu Stéphane, Boucheron Stéphane *

Heudiasyc, U.R.A. C.N.R.S. 817
Université de Technologie de Compiègne
Centre de Recherches de Royallieu
B.P. 649, 60206 Compiègne Cedex, France

Laboratoire de recherche en informatique
C.N.R.S. Université Paris-Sud, Bât. 490
91405 Orsay Cedex, France

Abstract. This paper considers the problem of function approximation from scattered data when using multilayered perceptrons. We present a new algorithm based on the principle of Inputs Perturbation, improving the generalization performances of backpropagation. In this algorithm, a new parameter is introduced in order to allow a control of the complexity of the fit. This parameter balances the bias versus the variance independently of the smoothness of the inputs density estimate. The tuning capacity of the algorithm is illustrated by experimental evidences.

1. Introduction

In this paper, we consider the training of a multi layered perceptron (MLP) for solving a regression estimation problem. The sample Z_ℓ gathers independent identically distributed observations drawn from the law of the random variable $Z = (X, Y)$: $z^i = (x^i, y^i) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}$, $(i = 1, \ldots, \ell)$.

Solving the regression estimation problem is defined as minimizing a cost function C over all function $f \in \mathcal{F}$, where $\mathcal{F}$ is the space defined by the architecture of the net. The cost C is the expectation of a loss function l:

$$f^* = \underset{f \in \mathcal{F}}{\text{Argmin}}\, C(f) \quad , \quad C(f) = \mathbb{E}_Z \left[l(Y, f(X)) \right] \tag{1}$$

In real-life applications, C is usually not computable as the density p_Z of Z is unknown. An empirical computable cost is then minimized using the sample Z_ℓ.

*Yves Grandvalet and Stéphane Canu are with Heudiasyc, Stéphane Boucheron is with Laboratoire de recherche en informatique.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 167-174

The classical empirical cost C_{emp} circumvents the estimation of p_Z, using the uniform strong law of large numbers:

$$C_{emp}(f) = \frac{1}{\ell}\sum_{i=1}^{\ell} l(y^i, f(\boldsymbol{x}^i)) \quad \text{, assuming } \lim_{\ell\to\infty} \max_{f\in\mathcal{F}} |C_{emp}(f) - C(f)| = 0 \quad (2)$$

Treating the regression estimation problem with a finite size sample by direct minimization of (2) usually induces a poor generalization to previously unseen patterns as the network overfits the data.

Numerous theoretical approaches, based on the Occam Razor Principle, have been devised to overcome this problem.

Concurrently, some heuristics have been proposed to avoid overfitting, by implicitly minimizing the complexity of the network. One of them, especially attractive since without additive computational cost, consists in applying inputs perturbation (IP) to the network during training.

The first part of this paper introduces the IP technique and recalls some theoretical results to exhibit the need of a new parameter to control the complexity of the function f^*. The introduction of this parameter in the IP algorithm is then developed and the complexity tuning capacity of this new algorithm is illustrated on a simulated and a real data set.

2. Inputs perturbation

The principle of the IP algorithm is that the original training sample Z_ℓ can be duplicated n fold by adding some noise $\boldsymbol{\eta}$ to the inputs $\boldsymbol{x}^i$. When n is large enough, minimizing C_{emp} on the enlarged sample C_{IP} becomes virtually equivalent to minimizing a mathematical expectation:

$$C_{IP}(f) \simeq \mathbb{E}_{\boldsymbol{\eta}}\left[\frac{1}{\ell}\sum_{i=1}^{\ell} l\left(y^i, f(\boldsymbol{x}^i + \boldsymbol{\eta})\right)\right] \quad (3)$$

It has been shown experimentally that IP could improve dramatically the generalization ability of MLP's [5]. Theoretically, two frameworks permit to explain this increase: Regularization theory and kernel density estimation. The scope of this paper is limited to this last theoretical framework.

The distribution of the noise $\boldsymbol{\eta}$ is considered in [1], [4], and [6] as a kernel used to approximate the distribution p_X of the inputs data $\boldsymbol{x}$. With this approximation $\widehat{p}_X$, the cost C_{IP} is equivalent to the true cost C:

$$C_{IP}(f) \simeq \int_{\mathbb{R}^d} \sum_{i=1}^{\ell} l\left(y^i, f(\boldsymbol{x})\right) \widehat{p}(\boldsymbol{x})\, d\boldsymbol{x} \quad \text{with} \quad \widehat{p}(\boldsymbol{x}) = \frac{1}{\ell}\sum_{i=1}^{\ell} \varphi\left(\boldsymbol{x} - \boldsymbol{x}^i\right) \quad (4)$$

Where φ is the distribution of the noise.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 167-174

If the loss l is quadratic, $l(y^i, f(\boldsymbol{x}^i)) = (y^i - f(\boldsymbol{x}^i))^2$, and if the function space in which f is chosen is not restricted to MLP's, the optimal function f_{IP} minimizing C_{IP} is shown to be the Nadaraya-Watson smoother [2, 3]:

$$f_{IP}(\boldsymbol{x}) = \frac{\sum_{i=1}^{\ell} y^i \varphi(\boldsymbol{x} - \boldsymbol{x}^i)}{\sum_{j=1}^{\ell} \varphi(\boldsymbol{x} - \boldsymbol{x}_j)} \tag{5}$$

Using the IP algorithm with a quadratic loss function is equivalent use the smoother f_{IP} as a target by minimizing:

$$C_{IP}(f) \simeq \int_{\mathbb{R}^d} \left(f(\boldsymbol{x}) - f_{IP}(\boldsymbol{x})\right)^2 \, \widehat{p}_X(\boldsymbol{x}) \, d\boldsymbol{x} \tag{6}$$

The MLP is used to approximate a smoother which has good convergence properties for $\ell \to \infty$ or for dense inputs data. Its role is to supply a fit which is computationally cheaper than f_{IP} to evaluate for large sample size ℓ.

The tuning parameters of IP are the shape and the covariance matrix of the noise density φ. In the statistic community, the shape of the kernel used by a smoother is usually considered to be unimportant compared to the choice of its width, *i.e* the covariance matrix [2]. Although asymptotic study provides optimal kernels, for practical problems, the choice of the kernel has not much influence on the mean squared error (1).

The width of the kernel may be thought as the inputs range for which the outputs are correlated. It assigns a scale to locality in the fit. It may be chosen *a priori* thanks to our prior knowledge of data, or *a posteriori* by cross-validation or any other resampling procedure.

But, once the width is chosen, the complexity of the function f_{IP} is fixed. When the noise amplitude is small compared to the inputs spacing, local means punctual, and f_{IP} is a bin smoother with as many bins as the number of distinct inputs. When the noise amplitude is large compared to the inputs range, local means global, and f_{IP} computes the mean of the outputs. For a given sample, locality forces the complexity of the smoother.

The modification of the IP algorithm we describe below consists in introducing a new parameter which allows the control of the complexity of the smooth f for any choice of the noise amplitude.

3. Complexity controlled inputs perturbation

3.1. Introducing the control parameter

The IP algorithm returns a function f^* which minimizes the mean square error when the density of the inputs is approached by the kernel estimator $\widehat{p}_X(\boldsymbol{x})$ given in (4). The complexity control of the function f^* we propose

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 167-174

here is executed by balancing the measures of fit and smoothness given by a regularization term. Indeed, the cost C_{IP} expressed in (3) can be decomposed as follows:

$$\begin{aligned} C_{IP}(f) &\simeq \frac{1}{\ell}\sum_{i=1}^{\ell}\left(y^i - \mathbb{E}_{\eta}\left[f(\boldsymbol{x}^i+\boldsymbol{\eta})\right]\right)^2 \\ &\quad +\mathbb{E}_{\eta}\left[\frac{1}{\ell}\sum_{i=1}^{\ell}\left(f(\boldsymbol{x}^i+\boldsymbol{\eta}) - \mathbb{E}_{\eta}\left[f(\boldsymbol{x}^i+\boldsymbol{\eta})\right]\right)^2\right] \end{aligned} \tag{7}$$

By introducing a regularization (positive) parameter λ in equation (7), we become able to balance the fit represented by the first term versus the smoothness constraint given by the second term. The tuning of λ allows the adjustment of the function complexity independently of the size of the neighborhood defining the locality of the fit. Thus, we become able to compare different functions (of same complexity) corresponding to different priors (size of neighborhoods).
The corresponding cost is as follows:

$$\begin{aligned} C_{\lambda IP}(f) &= \frac{1}{\ell}\sum_{i=1}^{\ell}\left(y^i - \mathbb{E}_{\eta}\left[f(\boldsymbol{x}^i+\boldsymbol{\eta})\right]\right)^2 \\ &\quad +\lambda\ \mathbb{E}_{\eta}\left[\frac{1}{\ell}\sum_{i=1}^{\ell}\left(f(\boldsymbol{x}^i+\boldsymbol{\eta}) - \mathbb{E}_{\eta}\left[f(\boldsymbol{x}^i+\boldsymbol{\eta})\right]\right)^2\right] \end{aligned} \tag{8}$$

This cost can also be decomposed in C_{IP} and the regularization term $\Omega(f)$ as follows:

$$C_{\lambda IP} = C_{IP} + (\lambda - 1)\ \Omega(f) \tag{9}$$

$$\text{with } \Omega(f) = \mathbb{E}_{\eta}\left[\frac{1}{\ell}\sum_{i=1}^{\ell}\left(f(\boldsymbol{x}^i+\boldsymbol{\eta}) - \mathbb{E}_{\eta}\left[f(\boldsymbol{x}^i+\boldsymbol{\eta})\right]\right)^2\right] \tag{10}$$

Minimizing the regularization term Ω of $C_{\lambda IP}$, is rather tricky as it involves an expectation.
The function f being an MLP, it is parameterized by its weight vector $\boldsymbol{w}$. At each step of the optimization procedure, the gradient of Ω with respect to $\boldsymbol{w}$ is computed. We have thus to use a cheap estimate of $\mathbb{E}_{\eta}\left[f(\boldsymbol{x}^i+\boldsymbol{\eta})\right]$. We choose $f(\boldsymbol{x}^i)$, which is the exact value for linear functions.

However $f(\boldsymbol{x}^i)$ should not be modified when used as an estimate of $\mathbb{E}_{\eta}\left[f(\boldsymbol{x}^i+\boldsymbol{\eta})\right]$. We therefore consider the following approximation of Ω:

$$\widehat{\Omega}(f) = \mathbb{E}_{\eta}\left[\frac{1}{\ell}\sum_{i=1}^{\ell}\left(f(\boldsymbol{x}^i+\boldsymbol{\eta}) - F^i\right)^2\right] \quad \text{with } F^i = f(\boldsymbol{x}^i) \text{ fixed} \tag{11}$$

This estimate yields the following approximation of $C_{\lambda IP}$ in (8):

$$C_{\lambda IP} \simeq \mathbb{E}_{\eta}\left[\frac{1}{\ell}\sum_{i=1}^{\ell}\left(\lambda\ f(\boldsymbol{x}^i+\boldsymbol{\eta}) + (1-\lambda)\ F^i - y^i\right)^2\right],\ F^i = f(\boldsymbol{x}^i) \text{ fixed} \tag{12}$$

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 167-174

N.B: minimizing $C_{\lambda IP}$ in (12) should not be considered as a relaxation method somewhere in between minimizing C_{emp} in (2) and C_{IP} in (3). Indeed, if $\lambda = 1$, then $C_{\lambda IP} = C_{IP}$, but if $\lambda = 0$, then $C_{\lambda IP}$ is not C_{emp} as it is a constant, in this case, f is completely undetermined. The fact that in (12) $C_{\lambda IP}$ is not minimized with respect to $F^i = f(\boldsymbol{x}^i)$ is crucial.

The optimal solution $f_{\lambda IP}$ of (12) is obtained in a similar manner to f_{IP} (5):

$$f_{\lambda IP}(\boldsymbol{x}) = \frac{\sum_{i=1}^{\ell} \left(y^i - (1-\lambda)\ f_{\lambda IP}(\boldsymbol{x}^i)\right) \varphi(\boldsymbol{x} - \boldsymbol{x}^i)}{\lambda \sum_{k=1}^{\ell} \varphi(\boldsymbol{x} - \boldsymbol{x}^k)} \tag{13}$$

This expression of $f_{\lambda IP}$ is convenient as the corresponding smoothing matrix can easily be exhibited, thus allowing a the calculation of the number of degrees of freedom *df* [3]. Assuming that $\mathrm{Var}[Y|X] = \sigma^2$, *df* is defined by the mean error and bias on $\{\boldsymbol{x}^i\}_{i=1}^{\ell}$:

$$df = \frac{\ell}{\sigma^2} \mathbb{E}_Y \left[\frac{1}{\ell} \sum_{i=1}^{\ell} \left(Y^i - f(\boldsymbol{x}^i)\right)^2 - \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\mathbb{E}_Y[Y|X = \boldsymbol{x}^i] - f(\boldsymbol{x}^i)\right)^2 \right] \tag{14}$$

As for IP, training an MLP with (12) is equivalent to use $f_{\lambda IP}$ as a target to determine f:

$$C_{\lambda IP}(f) \simeq \int_{\mathbb{R}^d} \left(f(\boldsymbol{x}) - f_{\lambda IP}(\boldsymbol{x})\right)^2\ \widehat{p}_X(\boldsymbol{x})\ d\boldsymbol{x} \tag{15}$$

With $\widehat{p}_X$ defined in (4).

As for IP, this minimization is done without the computation of $f_{\lambda IP}$. If the sample size ℓ is small, the computation of the function $f_{\lambda IP}$ should be directly carried out, but for large samples, the MLP is used as the "compiler" of $f_{\lambda IP}$ which becomes computationally expensive to evaluate.

We summarize below the sketch of the algorithm.

3.2. Algorithm

The algorithm for the complexity controlled inputs perturbation is simply using the backpropagation (BP) algorithm. At each iteration of an on-line BP, the following operations are carried out:

draw $\boldsymbol{\eta}$ **from the noise distribution** φ
compute the outputs $f(\boldsymbol{x}^i)$ **and** $f(\boldsymbol{x}^i + \boldsymbol{\eta})$
compute error $= \left(\lambda\ f(\boldsymbol{x}^i + \boldsymbol{\eta}) + (1-\lambda)\ f(\boldsymbol{x}^i) - y^i\right)^2$
compute gradient $= 2\lambda \left(\partial f(\boldsymbol{x}^i + \boldsymbol{\eta})/\partial \boldsymbol{w}\right) \left(\lambda\ f(\boldsymbol{x}^i + \boldsymbol{\eta}) + (1-\lambda)\ f(\boldsymbol{x}^i) - y^i\right)$
modify weights $\boldsymbol{w} = \boldsymbol{w} - \rho$ **gradient**

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 167-174

This algorithm can be seen as a Monte Carlo method for minimizing (15). Note that, as for IP, it could be applied to any parameterized function f whose parameters are determined by an iterative optimization algorithm. Compared to IP, this algorithm is more expensive since two output computations are required to compute the gradient. Nevertheless, the additional computing cost is small since evaluations of f are rapid.

4. Simulations

In this section, we illustrate the complexity tuning capacity of the algorithm on a simple simulated data set and on the motorcycle data set given in [2]. The noise η is Gaussian of zero mean and standard deviation σ. For all smoothers, the parameters σ and λ are set in order to get a constant df (14).

Figure 1 shows the simulated data set example. It was created to zoom in the different types of possible solutions $f_{\lambda IP}$.

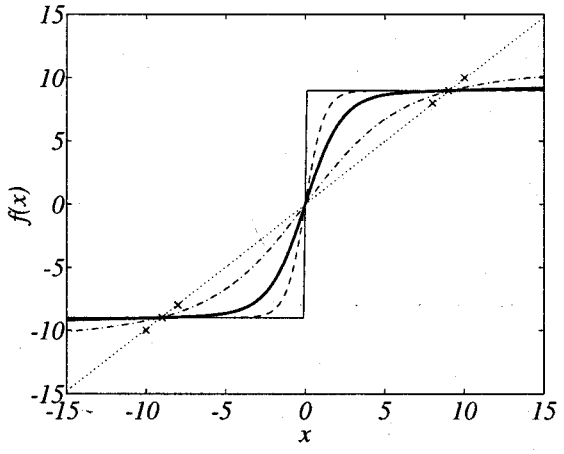

Figure 1: Simulated data set. Various smoother with 2 degrees of freedom: light solid line: $\sigma \simeq 0.4$, $\lambda = 10^4$; dashed line: $\sigma \simeq 3$, $\lambda = 10^3$; thick solid line: $\sigma \simeq 5$, $\lambda = 1$; dashdot line: $\sigma \simeq 7$, $\lambda = 10^{-1}$; dotted line: $\sigma \simeq 30$, $\lambda = 10^{-3}$.

When σ is small compare to the inputs spacing, the fit is a (local) bin smoother, with as many bins as df (here two). The bins limit is located at the midway of the most distant input data.

When σ is large compared to the input range, the fit is a (global) polynomial of degree $(df - 1)$, here a regression line.

All situations between local and global fit are possible when σ varies, thanks to the tuning of λ, df remains constant.

Figure 2 displays some solutions of the fit of the motorcycle real data set. This

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 167-174

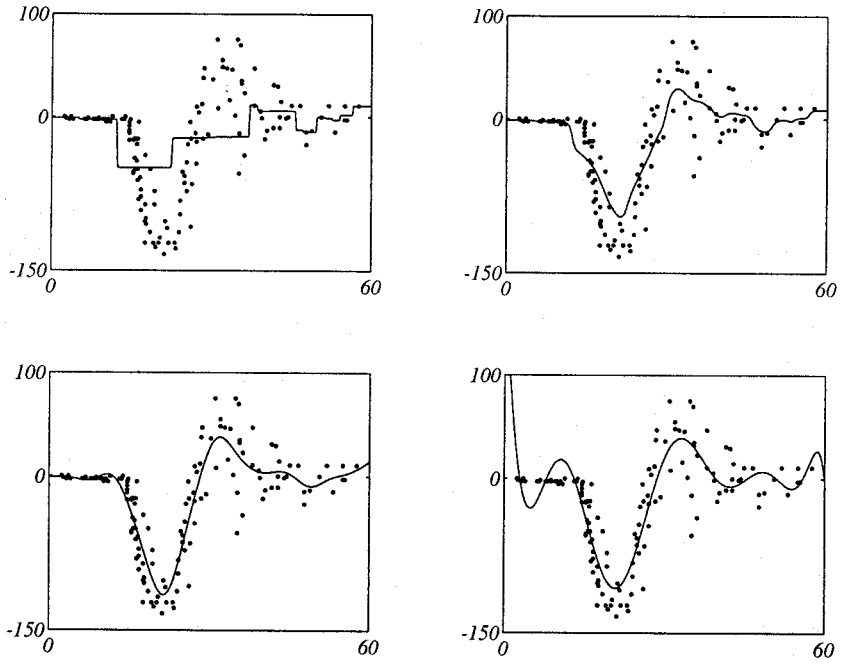

Figure 2: Motorcycle data set. Various smoother with 10 degrees of freedom: left-upper corner: $\sigma \simeq 0.25$, $\lambda = 10^6$; right-upper corner: $\sigma \simeq 0.8$, $\lambda = 10$; left lower corner: $\sigma \simeq 3$, $\lambda = 10^{-1}$; right-lower corner: $\sigma \simeq 21$, $\lambda = 10^{-10}$.

set is interesting for smoothing as the data are irregularly placed and that the dispersion of the output is varying along the abscissa. The two extreme cases (bin smoother and global polynomial fit) are represented.

For the bin smoother, the number of data points in each bin varies from one to more than thirty points. The delimiter of the bins being the largest gaps in the inputs data.

The computation of the residual of the four fits is a fair way of comparing the candidate solutions as they all verify $df \simeq 10$. The best result is obtained for $\sigma = 3$ and $\lambda = 0.1$, where the resulting curve is very close to a spline fit. The locality set by σ corresponds to the data.

5. Conclusion

The introduction of a new parameter in the IP algorithm allows a more flexible smoothing. The scale of locality (ranging from punctual to global), is set by the width of the noise distribution. The new parameter balances then the fit versus the smoothness of the regressor to control the complexity of the solution.

Thus, it sets the number of degree of freedom of the smoother returned by the algorithm. Moreover, the form of the optimal smoother permits to calculate this number of degrees of freedom.

The computation of the residuals obtained from different fits with same number of degrees of freedom is then used to select the best prior (scale of locality) according to data.

However, the evaluation of the optimal smoother becomes computationally expensive for large samples. In this case, an MLP trained with the proposed algorithm will converge towards the optimal smoother. It can be considered as the smoother's "compiled" version.

The Nadaraya-Watson smoother performs poorly in high dimensional inputs spaces. In those spaces, neighborhood containing little data become large [2], so that the smoother becomes close to a global averaging. The control of the complexity carried out by our algorithm enables to build a smoother which behaves well for irregularly spaced data. We suppose then that it behaves correctly in high dimensional spaces. This conjecture should be checked on high dimensional benchmarks.

References

[1] P. Comon. Classification supervisée par réseaux multicouches. *Traitement du Signal*, 8(6):387–407, 1992.

[2] W. Härdle. *Applied nonparametric regression*, volume 19 of *Economic Society Monographs*. Cambridge University Press, New York, 1990.

[3] T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, Redwood City, 1990.

[4] L. Holmström and P. Koistinen. Using additive noise in back-propagation training. *IEEE Transactions on Neural Networks*, 3(1):24–38, Jan 1992.

[5] J. Sietsma and R.J.F. Dow. Creating artificial neural networks that generalize. *Neural Networks*, 4(1):67–79, 1991.

[6] A.R. Webb. Functional approximation by feed-forward networks: A least-squares approach to generalization. *IEEE Transactions on Neural Networks*, 5(3):363–371, 1994.