Robustness in Neural Networks

Cesare Alippi
Politecnico di Milano, Italy

Manuel Roveri
Politecnico di Milano, Italy

Giovanni Vanini
Politecnico di Milano, Italy

INTRODUCTION

The robustness analysis for neural networks aims at evaluating the influence on accuracy induced by perturbations affecting the computational flow; as such it allows the designer for estimating the resilience of the neural model w.r.t perturbations. In the literature, the robustness analysis of neural networks generally focuses on the effects of perturbations affecting biases and weights. The study of the network's parameters is relevant both from the theoretical and the application point of view, since free parameters characterize the "knowledge space" of the neural model and, hence, its intrinsic functionality.

A robustness analysis must also be taken into account when implementing a neural network (or the intelligent computational system into which a neural network is inserted) in a physical device or in intelligent wireless sensor networks. In these contexts, perturbations affecting the weights of a neural network abstract uncertainties such as finite precision representations, fluctuations of the parameters representing the weights in analog solutions (e.g., associated with the production process of a physical component), ageing effects or more complex, and subtle uncertainties in mixed implementations.

BACKGROUND

The sensitivity/robustness issue has been widely addressed in the neural network community with a particular focus on specific neural topologies. In particular, when the neural network is composed of linear units, the relationship between perturbations and the induced performance loss can be obtained in a closed form (Alippi & Briozzo, 1998). Conversely, when the neural topology is non-linear, we have either to assume the small perturbation hypothesis or particular assumptions about the stochastic nature of the neural computation (e.g., see Alippi (2002a), Alippi et al. (1998), and Pichè, 1995); unfortunately, such hypotheses are not always satisfied in real applications. Another classic approach requires expanding the neural computation with Taylor around the nominal value of the trained weights. A subsequent linearized analysis follows, which allows the researcher to solve the sensitivity issue problem (Pichè, 1995). This last approach has been widely used in the implementation design of neural networks where the small perturbation hypothesis abstracts small errors introduced by finite precision representations of the weights (Dundar & Rose, 1995; Holt & Hwang, 1993). Again, the validity of the analysis depends on the validity of the small perturbation hypothesis.

Differently, other authors avoid the small perturbation assumption by focusing the attention on very specific neural network topologies and/or by introducing particular assumptions regarding the distribution of perturbations, internal neural variables, and inputs as done for Madalines neural networks (Alippi, Piuri, & Sami, 1995; Stevenson, Winter, Widrow, 1990).

Some other authors tackle the robustness issue differently by suggesting techniques leading to neural networks with improved robustness ability by acting on the learning phase (e.g., see Alippi, 1999) or by introducing modular redundancy (Edwards & Murray, 1998); though, no robustness indexes are suggested there. The robustness of neural networks with respect to hardware implementations were also studied in Hereford and Kuyucu (2005) and Nugent, Kenyon, and Porter (2004) where authors proposed evolutionary and adaptive approaches.

Again, the study of robustness over training time has been evaluated for neural networks in the large, without assuming the small perturbation hypothesis (Alippi, Sana, & Scotti, 2004). In this direction, other authors have addressed the issue of the robustness analysis during the training phase (Manic & Wilamowski, 2002; Qin Juanyin, Wei Wei, & Wang Pan, 2004) by suggesting a genetic approach or by considering the use of the regression theory.

An overview of the sensitivity issues in neural networks can be found in Ng, Yeung, Xi-Zhao, and Cloete, (2004).

In this article, we suggest a robustness/sensitivity analysis in the large (i.e., without assuming constraints on the size or nature of the perturbation); as such, the small perturbation hypothesis becomes only a subcase of the theory. The suggested

sensitivity/robustness analysis can be applied to ALL neural network models (including recurrent neural models) involved in system identification, control signal/image processing and automation-based applications without any restriction to study the relationship between perturbations affecting the knowledge space and the induced accuracy loss.

A ROBUSTNESS ANALYSIS IN THE LARGE

In the following we consider a generic neural network implementing the $\hat{y} = f(\theta, x)$ function where $\hat{\theta}$ is the weight vector of the trained neural network.

In several neural models, and in particular in those related to system identification and control, the relationship between the inputs and the output of the system are captured by considering a regression vector φ, which contains a limited time-window of actual and past inputs, outputs, and -possibly- predicted outputs.

Of particular interest are those models, which can be represented by means of the model structures $\hat{y}(t) = f(\varphi)$ where function $f(\cdot)$ is a regression-type neural network, characterized by N_φ inputs, N_η non-linear hidden units, and a single effective linear/non-linear output (Hassoun, 1995; Hertz, Krog, & Palmer, 1991; Ljung, 1987; Ljung, Sjoberg, & Hjalmarsson, 1996).

We denote by $y_\Delta(x) = f_\Delta(\theta, \Delta, x)$ the mathematical description of the perturbed computation and by $\Delta \in D \subseteq \Re^p$ a generic p-dimensional perturbation vector, a component for each independent perturbation affecting the network weights of model $\hat{y}(t)$. The perturbation space D, is characterized in stochastic terms by providing the probability density function pdf_D.

To measure the discrepancy between $y_\Delta(x)$ and $y(t)$ or $\hat{y}(t)$ we consider a generic loss function $U(\Delta)$. A common example for U is the Mean Square Error –MSE- loss function:

$$U(\Delta) = \frac{1}{N_x} \sum_{i=1}^{Nx} (y(x_i) - \hat{y}(x_i, \Delta))^2 \qquad (1)$$

but a generic Lebesgue measurable loss function with respect to D can be taken into account (Jech, 1978). The formalization of the impact of perturbation on the performance function can be simply derived as:

Definition: Robustness Index

We say that a neural network is robust at level $\bar{\gamma}$ in D, when the robustness index $\bar{\gamma}$ is the minimum positive value for which:

$$U(\Delta) \le \bar{\gamma}, \ \forall \Delta \in D, \ \forall \gamma \ge \bar{\gamma}. \qquad (2)$$

Immediately, from the definition of robustness index, we have that a generic neural network NN1 is more robust than another NN2 iff $\bar{\gamma}_1 < \bar{\gamma}_2$; the property holds independently from the topology of the two neural networks.

The main problem related to the determination of the robustness index $\bar{\gamma}$ is that we have to compute $U(\Delta)$, $\forall \Delta \in D$ if we wish a tight bound. The $\bar{\gamma}$-identification problem is therefore intractable from a computational point of view if we relax all assumptions made in the literature as we do.

To deal with the computational aspect we associate a dual probabilistic problem to (2):

Robustness Index: Dual Problem

We say that a neural network is robust at level $\bar{\gamma}$ in D with confidence η, when $\bar{\gamma}$ is the minimum positive value for which:

$$\Pr(U(\Delta) \le \bar{\gamma}) \ge \eta \ \text{ holds } \forall \Delta \in D, \ \forall \gamma \ge \bar{\gamma}. \qquad (3)$$

The probabilistic problem is weaker than the deterministic one since it tolerates the existence of a set of perturbations (whose measure according to Lebesgue is 1-η) for which $u(\Delta) > \bar{\gamma}$. In other words, not more than 100η% of perturbations $\Delta \in D$ will generate a loss in performance larger than $\bar{\gamma}$.

Probabilistic and deterministic problems are "close" to each other when we choose, as we do, η=1.

The non-linearity with respect to Δ and the lack of a priori assumptions regarding the neural network do not allow computing (2) in a closed form for the general perturbation case. The analysis, which would imply testing $U(\Delta)$ in correspondence with a continuous perturbation space, can be solved by resorting to probability according to the dual problem and by applying randomised algorithms (Alippi, 2002b; Bai, Tempo, & Fu, 1997; Tempo & Dabbene, 1999; Vidyasagar, 1996, 1998) to solve the robustness/sensitivity problem.

RANDOMIZED ALGORITHMS AND PERTURBATION ANALYSIS

In the following we denote by $p_\gamma = \Pr\{U(\Delta) \le \bar{\gamma}\}$ the probability that the loss in performance associated with perturbations in D is below a given—but arbitrary—value γ.

Probability p_γ is unknown, it cannot be computed in a form for a generic U function and neural network topology, and its evaluation requires exploration of the whole perturbation space D.

Anyway, the unknown probability p_γ can be estimated by sampling D with N independent and identically distributed samples Δ_i (intuitively a sufficiently large random sample explores the space); extraction must be carried out according