Neurocomputing xx (xxxx) xxxx–xxxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A neural network sensitivity analysis in the presence of random fluctuations

K.G. Kapanova*, I. Dimov, J.M. Sellier

IICT, Bulgarian Academy of Sciences, Acad. G. Bonchev str. 25A, 1113 Sofia, Bulgaria

ARTICLE INFO

Communicated by M.T. Manry

Keywords:
Neural networks
Feedforward networks
Random search
Simulated annealing
Sensitivity analysis
Function approximation

ABSTRACT

In this work we investigate the sensitivity analysis of the noise applied to a neural network. The goal of this study is to try to understand the network's behavior and the outputs sensitivity from the perturbation of weights during the training process. From an engineering point of view, noise is perceived as detrimental to the system and the quality of the output, but in biological neural systems, we can observe noise fluctuations, which possess certain abilities to improve information processing. By means of sensitivity analysis tools we show quantitatively the acceptable level of noise which provides optimal solution to the random fluctuations without sacrificing the behavior of the network. The three different indicators utilized in this endeavor allow us to observe whether the noise variance is detrimental or beneficial, and whether it acts as a source of fluctuation.

1. Introduction

In the past twenty years, Artificial Neural Networks (ANNs) have been implemented as a powerful tool in various fields - e.g. in process modeling and control, computer science, environmental control and big data analytics, where information is complex and voluminous, thus hard to analyze in full detail. From the inception of the idea about the artificial neuron, first presented by McCulloch and Pitts (1943) [1], the field has expanded to encompass many different ANNs paradigms [2]. Among them, the most widely used ones are the back-propagation multi-layer network and the recurrent neural network [3,4].

Generally speaking, ANNs have incorporated some indispensable structures of biological neurons, more precisely - input, processing, learning, as well as output. Neural Networks process information and operate in a similar manner to biological neurons, where the neural structure is specified by the connection between the elements. Thus one can develop an ANN following biological principles - signal, neural architecture, processing, output. Likewise, researchers have addressed the issue on the performance variability from “random or unpredictable fluctuations and disturbances” [5] and the possible benefits noise in neural systems can introduce [6]. Biologically speaking, noise is naturally found in neural systems, with certain abilities to improve information processing [5–8]. In engineering systems on the other hand, noise is perceived as detrimental to the system and the quality of the output. To the best of our knowledge, no definitive answer has been given in regards to the exploitation of naturally occurring fluctuations in neural systems, the cause of the occurrence and whether any potential advantages exist. One might consider that the addition of noise could contribute to the network's performance from the oscillations at certain levels for one noise range, while in other cases this oscillation could be detrimental [9].

Analysis of the addition of noise to the weights during the training process of a recurrent neural network have been performed in [10], with the aim to reach a possible improvement of convergence through the search in the weight space. Various strategies exist to adding noise in the system: noncumulative and cumulative (with additive and multiplicative effect) either per time step or per string. Such synaptic noise addition could be beneficial in problems where the network is stuck in local minima and provide a possible escape mechanism. Similarly, in [11], noise has been inserted in the weights during training of a multilayer perceptron network in order to explore the effects on the fault-tolerance of the system and on the feasibility of learning acceleration and effects on the error function.

A different approach to this optimization problem is by randomly probing the space of weights through the addition of a certain level of noise to all weights. In order to achieve this, the network is initialized. Computation of the error after weights selection is performed and a second search for another batch of weights is executed. The two errors are compared and the one with the smaller error is selected. Similar approach has been implemented in [12], where a deep neural network for speech recognition system is trained in a noise-aware environment. The noise is applied only in the input layer (in the signal) either by representing the noise through environmental information or by removing it in order to reduce variability.

One may consider the case of hardware representation of a neural network, for example the experiments by Hewlett-Packard and IBM,

* Corresponding author.
E-mail addresses: kapanova@parallel.bas.bg, kkapanova@gmail.com (K.G. Kapanova).

http://dx.doi.org/10.1016/j.neucom.2016.10.060
Received 11 August 2015; Received in revised form 29 July 2016; Accepted 18 October 2016
Available online xxxx
0925-2312/

with Memristors [13]. The reasonable expectation that the many core Memristor devices are enclosed in a limited space for the network, thus observing more electrical current activity and random fluctuations that could affect the performance of the system in various ways.

In this paper, we present a sensitivity analysis tools to represent an accurate mathematical approach to establish the level of noise in a network and whether it provides improvement to the system or is just a source of fluctuations. The use of Sensitivity Analysis tools allows us to study how the noise variation affects the behavior of the network and the sensitivity of the output in relation to the noise fluctuation. The sensitivity coefficients suggest the variance of the network's outputs due to changes in the weight parameters of the system. While in certain situations, the effect of the noise in the input signals could be predictable, no study (to the best of our knowledge) has been carried out explaining the level of influence of the noise and the consequences to the functionality of the network. One should recognize that considerable sensitivity to the noise in the weights can substantially change the performance of the system. Vice versa, in a case of negligible sensitivity, one may expect little change in the performance. Additionally, we can quantitatively show that intermediate levels of noise improve the network's performance, even if it is counter intuitive. We selected several measures, or in other words indicators, to explore the sensitivity of a neural network over the introduction of noise in the weight space. To explore this questions, three different indicators are used: the Euclidean distance (L_2), the maximum norm (L_∞) and the cosine-similarity (L_{cos}) to establish the scope of the solutions. The decision to apply those three indicators is two-fold. On one hand, considering one or two indicators would provide a partial picture of the sensitivity spectrum, thus limiting our understanding on the noise influence. On the other hand, implementing more indicators, thus increasing the level of information, could potentially dilute the overall understanding of the indicators and how the presence of noise can be analyzed and potentially exploited in neural networks. Moreover, the three specific indicators have been successfully executed in other sensitivity studies [14]. In specific, considering two weight vectors from the weight-space, the cosine similarity provides us with understanding about their orientation, the Euclidean distance will serve to introduce the possible directions and magnitudes of the selected weights, while the L_∞ will give us the parameter of their maximum departure. For every case we perform, we start from the same initial conditions in order to have duplicable and comparable results.

The paper is organized as follows. In the next section we introduce the methods used for the implementation of such an approach. Then we provide several numerical experiments to validate our approach. Further analysis of the robustness of the system when randomly searching for the weight space is needed. We believe that this study shows clearly that Sensitivity Analysis tools can provide a reliable way to study the presence of complex external effects such as random fluctuations and their effect on the performance of a neural network.

2. Methods and implementation

The field of Artificial neural networks has provided many different implementations of networks. Depending on the specific purpose of the network, decision about the number of layers, the type of neuron connections, the appropriate activation functions, and the training process (optimization problem) should be contemplated by developers [15].

In this work, as a first step towards eventual practical applications in the measurement of the sensitivity analysis in the presence of noise in the neural network, we have developed a three layer feedforward fully connected neural network (see Fig. 1). In our specific implementation, the first layer contains one input neuron, the hidden layer comprises of four neurons, with one output neuron in the final layer. In the current network model, every neuron is fully connected to the neurons in the next layer. The selected activation function is $tanh(x)$ with a range $[-1, 1]$. One should note that the sensitivity analysis tools are not limited to this particular architecture, activation function or training method.

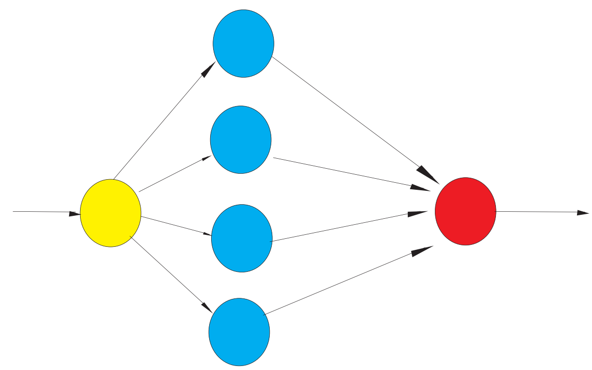

Fig. 1. Current design of the neural network implemented in this paper. Neurons are represented as nodes, the connections between the neurons are illustrated as directed edges. The six neurons are positioned as follows: one input neuron, four neurons in the hidden layer, and one output neuron.

Although many learning algorithms exist, in this particular work, we have selected a training process based on the simulated annealing method. This particular technique is a meta-heuristic search algorithm, which uses analogy with thermodynamics and more specifically the cooling of metals. The strategy consist of two steps: firstly, the effective temperature is increased to the maximum value, secondly it is slowly being decreased, allowing for the redistribution of particles until they find themselves in the ground state of the solid (thus representing a solution to the optimization problem) [16]. At each step, the algorithm provides two possible solutions (the current energy an a newly selected one) , which are compared. Most of the time the solution that decreases the energy of the algorithm is accepted, even though some increasing temperature solutions are accepted in some cases with the goal for a possible escape from a local optima during the search for the global one. The probability of such an event occurring depends on the energy parameter, which decreases with each step, thus with time fewer non-optimal solutions will be accepted. The probability of a move for a point can be described as

$$Pr[accept] = e^{\frac{-\Delta E}{T}}, \tag{1}$$

with ΔE describing the difference between the actual energy and the energy before the move, while T represents the effective temperature of the system.

Once the optimization algorithm is set, we then set the level of randomization (noise addition) for the network in a straightforward manner, described in the pseudo code:

```
Run the Network
1. Begin Simulated Annealing
2. Set the level of randomization
3. repeat
  3.1 Select random weight w'
  3.2 Randomly generate second set of weights (w'')
   by adding certain percentage of noise to the solution
  3.3 Calculate error
  3.4 If error_new < error_old select the new weights
  3.5 else keep the previous solution
4. Continue until stopping criteria is met
```

One should recognize that depending on the selected training algorithm, the optimization strategy will provide different weights solutions. Our goal by conducting sensitivity analysis is to try and understand the network's behavior when noise is introduced in the weights space and what is the sensitivity of the output considering this perturbation of the weights. Thus, the selection of a training method is not limited to Simulated Annealing and other techniques like back-