Neurocomputing 151 (2015) 49–54

Contents lists available at ScienceDirect

Neurocomputing

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

Robustness of artificial metaplasticity learning algorithm

Diego Andina*, Santiago Torres-Alegre, Martín J Alarcón, Juan I Seijas, Marta de-Pablos-Álvaro

Group for Automation in Signals and Communications, Technical University of Madrid, Madrid, 28040, Spain

ARTICLE INFO

Article history:
Received 28 April 2014
Accepted 18 July 2014
Available online 18 October 2014

Keywords:
Synapse
Plasticity
Metaplasticity
Backpropagation
Artificial metaplasticity
Learning
Information entropy

ABSTRACT

Artificial metaplasticity learning algorithm is inspired by the biological metaplasticity property of neurons and Shannon's information theory. It is based on the bio-inspired hypothesis that neurons do not learn in the same amount (metaplasticity of biological learning) from unfrequent patterns than from common ones, as the former are expected to contain more information than the latter (information entropy concept). On MLPs, the artificial metaplasticity can be formulated as an improvement in regular backpropagation algorithm by using a variable learning rate affecting all the weights in each iteration step and so resembling heterosynaptic plasticity of biological neurons. The variable rate involves statistical inference on the training set and it is common to successfully assume Gaussian distribution for the training patterns. Nevertheless, Gaussian assumption may diverge from the real one and using statistical information extracted from the training patterns may be necessary. In this research, robustness to significative variations on Gaussian assumption is evaluated using input sets generated with different probability distributions. For the cases where Gaussian assumption shows to degrade learning, a general algorithm is applied. This algorithm takes advantage of the inherent statistical inference performed by the MLP through the *a posteriori* probabilities of input patterns estimation provided by its outputs. The generality of this last algorithm for any input distribution is then demonstrated.

1. Introduction

Synaptic plasticity is a concept related to memory and learning processes and widely used in the fields of biology, neuroscience, physiology, neurology and others [1,2]. It refers to the ability of the synapse between two neurons to modulate its efficiency. It involves different cellular processes that modify the synaptic efficiency and can cause an enhancement in synaptic strength (Long-term Potentiation) or a reduction (Long-term Depression) [3–5]. Metaplasticity was introduced by Abraham, as a higher level of plasticity, *the plasticity of plasticity* [6], respective of the neurons activation frequency or postsynaptic activity. If time-averaged level of postsynaptic is high, by metaplasticity property, reinforcement is lower than if time-averaged level of postsynaptic firing is low.

Artificial metaplasticity (AMP) term was first introduced by Andina et al. [7]. The method focuses on the amount of variation in artificial synaptic strength or *weights* in an Artificial Neural Network (ANN). Andina postulated that high postsynaptic activity has to be related to highly frequent excitations (frequent input classes in an artificial model). As the value of postsynaptic activation is crucial for determining the amount of variation in the biological synaptic strength, also the statistical distribution of patterns is crucial in the amount of variation in the artificial synaptic weight. Then, patterns of higher probability to occur will correspond to time-averaged higher frequent activations and therefore to lower reinforcement of artificial weights. Hence, during the learning of an ANN (called "training phase"), the AMP assigns higher values for updating the artificial synaptic weights in the less probable patterns than in the ones with higher probability. A more detailed study on the biological plausibility of AMP learning algorithm can be found in [8,9].

AMP concept is applicable to any ANN, although implementation has only been tested for the Multilayer Perceptron type (MLP), referred as AMMLP. The implementation and application of AMMLP trained by Backpropagation Algorithm (BP) was presented in [7] and some of the applications are detailed in [10–12].

In AMP implementation, a weighting function $f_X^*(x)$, related to the probability distribution of the input vectors is applied to the cost function to be minimized to find the proper ANN weight values (called "training phase" of ANN design). In order to model the probability density function of the input patterns, Gaussian hypothesis is valid in most cases:

$$f_X^*(x) = \frac{A}{\sqrt{(2\pi)^n}.e^{B\sum_{i=1}^{n} x_i^2}} \quad (1)$$

being $A, B \in R$ parameters to scale the Gaussian function. They have to be determined empirically for each set of patterns used in the

* Corresponding author.
E-mail address: andina@gc.ssr.upm.es (D. Andina).

http://dx.doi.org/10.1016/j.neucom.2014.07.075
0925-2312/© 2014 Published by Elsevier B.V.

training phase (the so-called "training patterns"). n is the number of components $x_i \in R^n$ (where R^n is the n-dimensional space, i.e. $x=(x_1, x_2, \ldots, x_n)$, $x_i \in R^1$, $i=1,2,\ldots,n$) of the input vector $x \in X$. X is the training set of all vector x used to train the ANN.

The AMMLP algorithm can then be formulated as an improvement in ordinary BP algorithm by using a variable learning rate $\eta(x)$ in the training phase affecting the weights in each iteration step based on an estimation of the real distribution of training patterns. That is if $l, j, i \in N$ are the MLP layer, node and input counter respectively (see Fig. 1), for each component of the weight matrix $W(t)$, $w_{ij}^{(l)}(t) \in R$, the weight reinforcement for each single training pattern is

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) - \eta(x)\frac{\partial E[W(t)]}{\partial w_{ij}} = w_{ij}^{(l)}(t) - \eta\frac{1}{f_X^*}\frac{\partial E[W(t)]}{\partial w_{ij}} \tag{2}$$

where $\eta \in R^+$ is a learning rate parameter and $E[W(t)]$ the error function to be minimized.

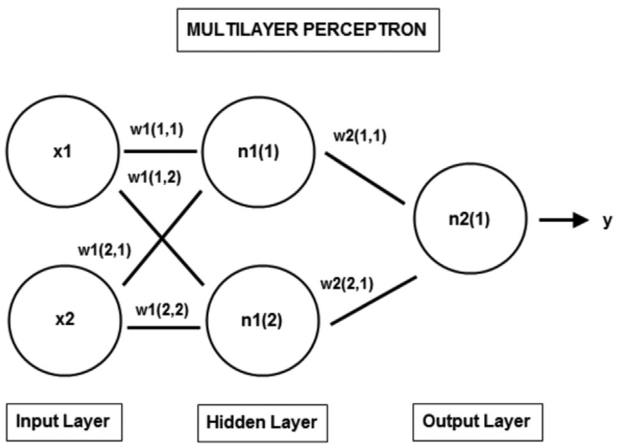

Fig. 1. The three-layer MLP used in this research.

2. Multilayer perceptron structure

In order to easily compare and represent important parameters of the AMP algorithm with regular BP, a simple MLP (2/2/1) (2 inputs, 2 nodes in the hidden layer and one output node) is used. Activation function is sigmoidal with values ranging in $(0,1)$, as it is well know that it allows the output to be an inherent estimation od the ANN of the *a posteriori* probabilities of input patterns [13]. A scheme of this neural network is shown in figure below:

Each weight of the network is $wL(I,j)$, and

- L: layer, $L=1$ for hidden layer and $L=2$ for output layer.

Table 1
Comparison between BP and AMMLP for Gaussian distribution.

	Backpropagation				Artificial Metaplasticity				
Errors	0.3750	0.3750	0.3750	0.3750	0.1250	0.6250	0.6250	0.4375	0.2500
$dw1(1,1)$	−0.0023	−0.0010	−0.0049	−0.0361	−0.0177	−0.0309	−0.0286	0.0451	0.0223
$dw1(1,2)$	0.0009	0.0004	0.0039	0.0509	0.0089	−0.3470	−0.2534	−0.0753	−0.0204
$dw1(2,1)$	0.0028	0.0011	0.0059	0.0431	0.0212	−0.0665	−0.0616	0.0971	0.0480
$dw1(2,2)$	−0.0011	−0.0005	−0.0047	−0.0609	−0.0107	−0.7468	−0.5454	−0.1621	−0.0439
$dw2(1,1)$	−Q.Q081	−D.D017	−0.0064	−0.0130	−Q.Q083	2.Q895	1.4078	0.4477	D.1208
$dw2(2,1)$	−0.0032	−0.0008	−0.0052	−0.0931	−0.0036	1.4972	0.4919	0.0720	0.0114
$w1$			−3.9933 −3.1234				0.2493	0.0409	
			5.4884 3.1668				−3.5997	−1.8993	
w2			− 5.7263 4.5436				0.5560	−3.9251	
n1(1)	0.6139	0.5070	0.3896	0.1080	0.3934	0.4571	0.3093	0.2096	0.1975
$n1(2)$	0.2428	0.2313	0.3128	0.7715	0.1731	0.3275	0.1081	0.0337	0.0186
$n2(1)$	0.1225	0.0595	0.1384	0.8211	0.1579	0.3003	0.3763	0.6828	0.8477
Iterations			2, 7				1, 7		

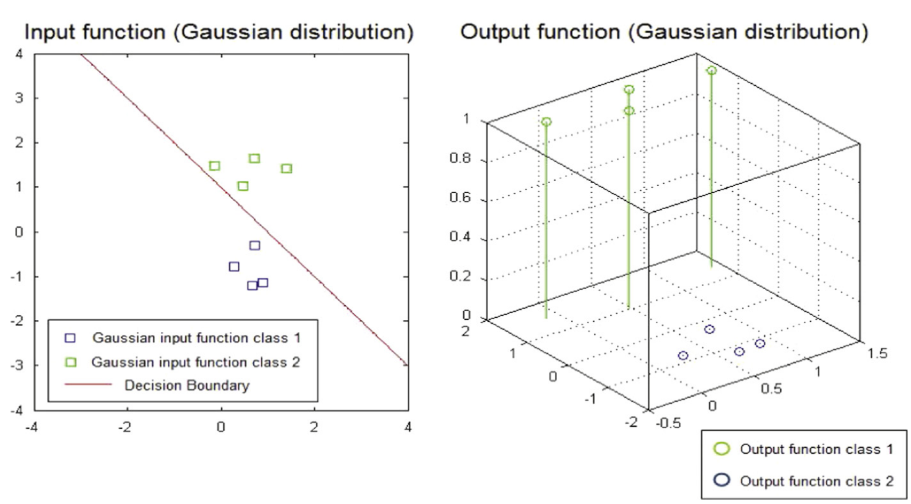

Fig. 2. Input function: two-dimensional Gaussian distribution with mean 0 and variance 1. Output function: evaluates if the sum of the two dimensions of each element is above 1 (class 2) or below 1 (class 1).