

AMSOM: artificial metaplasticity in SOM neural networks—application to MIT-BIH arrhythmias database

Santiago Torres-Alegre · Juan Fombellida · Juan Antonio Piñuela-Izquierdo · Diego Andina

Abstract

Artificial metaplasticity is the machine learning algorithm inspired in the biological metaplasticity of neural synapses. Metaplasticity stands for plasticity of plasticity, and as long as plasticity is related to memory, metaplasticity is related to learning. Implemented in supervised learning assuming input patterns distribution or a related function, it has proved to be very efficient in performance and in training convergence for multidisciplinary applications. Now, for the first time, this kind of artificial metaplasticity is implemented in an unsupervised neural network, achieving also excellent results that are presented in this paper. To compare results, a modified self-organization map is applied to the classification of MIT-BIH cardiac arrhythmias database.

Keywords Metaplasticity · AMMLP · AMP · Feature extraction · Artificial neural network · Self-organizing maps · AMSOM

1 Introduction

Biological synaptic plasticity is the ability of the synapse between two neurons to modulate its efficiency, and its key relevance in biological brains and intelligence is well known. Biological metaplasticity concept was introduced by Abraham [1] as a form of persistent synaptic plasticity induced by synaptic or cellular activity. Metaplasticity is thus a higher-order form of synaptic plasticity. Metaplasticity means a higher level of plasticity, the plasticity of the plasticity itself related to changes in synaptic efficiency

that can cause an enhancement or a reduction in synaptic strength [2, 3].

A proposal of artificial metaplasticity (AMP) was generally introduced by Andina et al. [4] and detailed for supervised artificial neural networks (ANN). After relevant results that can be found in the literature, now, for the first time, this kind of artificial metaplasticity is implemented in an unsupervised neural network, achieving also excellent results that are presented in this paper. To compare results, a modified self-organization map is applied to the classification of MIT-BIH cardiac arrhythmias database.

This paper is organized as follows: in Sect. 2, artificial metaplasticity in supervised neural networks is briefly described and references are given to enter in details. Then, in Sect. 2.1 artificial metaplasticity implemented in self-organization map (Kohonen's network) and applied to arrhythmias classification is detailed. In Sect. 3, data preparation and network structure selection are described. Results of experiments are presented in Sect. 4, followed by a discussion in Sect. 5, and finally ending with appropriate conclusions in Sect. 6.

2 Artificial metaplasticity in neural networks

In the learning phase, ANN weights are adjusted to obtain a specific performance or task. These weights are adjusted iteration by iteration. In this network learning process AMP is implemented by modifying the way the weights change by introducing a variable function in this process. Ropero-Pelaez [5], Andina [4] and Marciano-Cedeño [6] have introduced and modeled the biological property of metaplasticity in the field of ANNs, obtaining excellent results in multidisciplinary applications. They apply an approach that connects metaplasticity and Shannon's information theory, which establishes that less frequent patterns carry more information than frequent patterns [7]. In this AMP model, greater modifications in the synaptic weights are produced with less frequent patterns and fewer modifications with more frequent ones. Biological metaplasticity favors synaptic strengthening for low-level synaptic activity, while the opposite occurs for the high-level activity.

The artificial metaplasticity on multilayer perceptron supervised gradient-descent learning algorithm (AMMLP) tries to improve backpropagation algorithm (BPA) by including a variable learning rate in the training phase affecting the weights in each iteration step, that is the metaplasticity, instead of the uniform plasticity that applies in the classical BPA [8]. If $s, j, i \in \mathbb{N}$ are the MLP layer, node and input counter, respectively, $W(t)$ weight matrix $w_{ij}^{(s)}(t) \in \mathbb{R}$, η is the learning ratio, and then we can express the weight reinforcement in each iteration t as:

$$w_{ij}^{(s)}(t+1) = w_{ij}^{(s)}(t) - \eta \frac{\partial E^*[W(t)]}{\partial w_{ij}^{(s)}} = w_{ij}^{(s)}(t) - \eta \frac{1}{f_X^*} \frac{\partial E[W(t)]}{\partial w_{ij}^{(s)}} \quad (1)$$

Note that f_X^* is a function to improve the learning process that implements the metaplasticity concept. One possibility tested by several authors [4, 6, 9] is to implement an AMMLP based on an estimation of the distribution of the input patterns being then the function f_X^* :

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

where N is the number of neurons in the MLP input layer, and parameters A and $B \in \mathbb{R}^+$ are algorithm optimization values empirically determined which depend on the specific application of the AMMLP algorithm. Notice that to apply the proposed f_X^* function patterns are normalized.

As the outputs of an MLP are approximations of the *a posteriori* probability of the different classes [10], another option of implementing AMMLP was presented by the authors of this contribution [11] based on the estimation of

a posteriori probability density function considering f_X^* as follows:

$$\hat{y}_L \cong P(H_i/x) = f_X^*(x) \quad (3)$$

where \hat{y}_L is the output of the neuron that estimates the *a posteriori* probability of the class. It can be seen that Eq. 3 takes advantage of the inherent probabilities estimation of the MLP for each input class.

It has to be noticed that in the first steps of the training the outputs of the MLP do not provide yet any valid estimation of the probabilities, so the training may not converge. In practice, there are rarely instability problems of this kind, but if they occur, it is then better in these first steps of training, either to apply ordinary BPA training or to use another valid weighting function till BPA starts to minimize the error objective.

2.1 Artificial metaplasticity in SOMs

SOMs, self-organizing maps, are ANNs proposed by Kohonen [12, 13] where competitive learning is carried out, hence only one neuron is activated in each iteration, the so-called *winner takes all neuron*, and they are characterized by the formation of a topographic map of the input patterns. In a SOM, the neurons are normally displaced in a two-dimensional lattice and a weight matrix is randomly set.

The learning process follows the following rule.

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t)h_{j,i}(t)(\mathbf{x} - \mathbf{w}_j(t)) \quad (4)$$

where \mathbf{w}_j is the vector of weights of neuron j and \mathbf{x} is the input pattern of the correspondent iteration and $h_{j,i}$ is a neighborhood function that determines which neurons, besides, the 'winner takes all neuron', update its weights.

It should be noted that the learning rate η is not a constant but time dependent $\eta(t)$, and it decreases as the training phase evolves. As the training phase advance, the weights of each neuron tend to define a centroid of the input patterns that activates that neuron.

The modification that is proposed to introduce artificial metaplasticity in these kinds of neural networks is to modify the learning process in the same way it has been done in the MLP case assuming f_X^* to be Gaussian form. So, what we introduce in the learning ratio is the factor $1/f_X^*$ being the new updating weights rule:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \frac{1}{f_X^*} \eta(t)h_{j,i}(t)(\mathbf{x} - \mathbf{w}_j(t)) \quad (5)$$

with,

$$f_X^*(x) = \frac{A}{\sqrt{(2\pi)^N \cdot e^B \sum_{i=1}^N x_i^2}} \quad (6)$$

We define the ANN which implement this learning process as AMSOM (artificial metaplasticity self-organizing map). As in the AMMLP case patterns are normalized to $(-1, 1)$ interval.

3 Materials and methods

3.1 Data preparation

The MIT-BIH Arrhythmia Database was the first generally available set of ECGs (electrocardiograms) test material for evaluation of arrhythmia detectors [14]. Database contains 48 half-hour excerpts of two-channel, 24-hour, studied by the BIH Arrhythmia Laboratory. 1000 annotated ECG beats which contain 4 different waveforms related to cardiac arrhythmias target, Normal beat (N); premature ventricular contraction (PVC); right bundle branch block (RBBB) and left bundle branch block (LBBB) have been used in this research. In Table 1, the eleven feature descriptors for characterizing the cardiac arrhythmias are presented. Components of ECG signal could be seen in Fig. 1.

The data set is formed by 1000 patterns divided equally in four classes with 250 patterns each (N , PVC, RBBB and LBBB). We denominate, respectively, these classes as H_1 , H_2 , H_3 , and H_4 . In order to apply metaplasticity using the function f_X^* , previously presented database patterns are normalized to $(-1, 1)$ interval. To obtain results statistically independent of the distribution of the patterns, a tenfold cross-validation evaluation method has been considered. By using this method, the possible dependence of the results with the distribution of the samples in the training or performance evaluation sets is eliminated. All the samples are used to train the networks and to evaluate

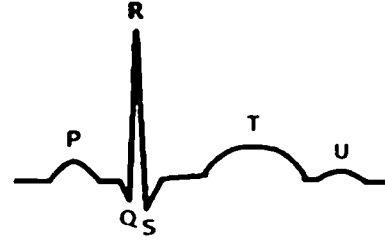


Fig. 1 ECG signal components

their performance in different executions of the experiment for the same initial neural networks. Average values are calculated to establish the final performance results.

3.2 Network structure selection

For the case of the supervised AMMLP ANN, an 11/9/4 network structure is selected, that is, 11 input neurons, one for each relevant feature selected, 9 neurons in the hidden layer, and 4 output neurons to represent the four possible classes.

Regarding the AMMLP training phase, the following parameters are considered:

- Learning rate $\eta = 1$.
- Activation function is sigmoidal with value in the interval $(0, 1)$.
- Initialize all weights in weight matrix W randomly between $(-0.5, 0.5)$
- if epochs = 200 stop training
- if mean squared error, $MSE = 0.01$ stop training
- for Gaussian AMMLP case $A = 39$ and $B = 0.5$ are empirically selected.

For the case of the SOM and AMSOM networks different structures are compared:

Table 1 Feature descriptors

Attributes	Meaning
Duration P	The width of the P wave
PR interval	The distance between the beginning of the P wave and the beginning of QRS
QRS complex	The distance between the beginning of the Q wave and the end of the S wave
Duration T	The width of the T wave
ST segment	The distance between the end of the S wave or R and the beginning of the T wave
QT interval	The distance between the beginning of QRS and the end of the T wave
RR previous: RRp	The distance between the R peak of the present beat and the R peak of the previous beat
RR next: RRn	RRn: the distance between the R peak of the present beat and the R peak of the following beat
RDI (delay of the deflexion)	From the beginning of QRS to the top of the latest wave of positivity R peak
Beat duration	The distance between the beginning of the P wave and the end of the T wave
RRp/RRn.	The ratio RRp/RRn

- Network size: 6×4 and 9×8 .
- epochs: 200 and 500.
- Learning rate: initially $\eta = 0.2$ at the end of the training phase $\eta = 0.001$.
- Initialize all weights in weight matrix W randomly between $(-1, 1)$.
- Neighborhood function: rectangular function.
- Winner neuron determination: Euclidean distance between input pattern and weights vector of each neuron
- metaplasticity parameters: $A = 5750$ and $B = 0.5$ are empirically selected.

4 Results

In this section, we present the results obtained in this research. All the network structures were trained and tested with the same data and validated using tenfold cross-validation. The eleven attributes detailed in Table 1 were used as the networks inputs. The SOM, AMSOM and AMMLP proposed as classifiers for cardiac arrhythmias were implemented in MATLAB by using functions from the Neural Networks Toolbox as *feedforwardnet*, with *traingdx* training function, for the AMMLP cases, and *selforgmap* for the SOM cases. Due to the impossibility to introduce the modifications related to the metaplasticity algorithm in the learning part of the process in these standard MATLAB functions, we had to build our own codes instead of using the ones included in the Toolbox. It is important to remark that patterns representing the four classes are alternately presented to the network in the training phase. If not, results are clearly worst.

4.1 Performance indicators

To evaluate the performance of the classifiers, three performance indicators defined as follows are used: Sensitivity (SE) = $(\frac{TP}{TP+FN}) \times 100$, Specificity (SP) = $(\frac{TN}{TN+FP}) \times 100$ and Accuracy (AC) = $(\frac{TP+TN}{TP+TN+FP+FN}) \times 100$, where TP, TN, FP, and FN stand for true positive, true negative, false positive and false negative, respectively.

4.2 Performance evaluation

For test results to be more valuable, a tenfold cross-validation is used in all of our experiments by separating the selected 1000 samples randomly into 10 subsets with 100 records each and then taking each subset as test data in turns. This cross-validation method is used among the researchers because it minimizes the bias associated with the random sampling of the training. The classification

algorithm is trained and tested k times. In each case, one of the folds is taken as test data and the remaining folds are added to form training data. Thus, k different test results exist for each training-test configuration. The average of these results provides the test accuracy of the algorithm.

In Table 2, results are presented, and SOM, AMSOM, AMMLP1 and AMMLP2 stand, respectively, for standard self-organizing map, artificial metaplasticity SOM, MLP implementing Gaussian function to modify the weights of the network and MLP implementing the output of the network to modify the weights. AMMLP1 and AMMLP2 results were presented in [9, 11]. The models are evaluated based on the accuracy measures discussed above (classification accuracy, sensitivity and specificity). The results were achieved using tenfold cross-validation for each model and are based on the average results obtained from the test data set for each fold.

The evolution of parameters specificity, sensitivity, and accuracy for the best networks of this study can be seen in Fig. 2. For AMMLP1, AMMLP2 and SOM 6×4 we present the evolution till 300 epochs because best results are obtained after 200 epochs. However, for the AMSOM 9×6 evolution for 500 epochs is shown because this last network improves its performance between 200 and 500 epochs. It can be seen that SOM 6×4 is the fastest ANN in reaching higher values of specificity and maintains the best performance for this parameter (99.70). AMMLP2 behaves similar to SOM 6×4 in terms of specificity where AMMLP obtains the worst performance. Finally, AMSOM 9×6 begins lower in terms of specificity, but as the number of epochs increases it almost reaches the value of SOM 6×4 (99.6). Regarding the sensitivity, AMSOM 9×6 and AMMLP1 present the best performance, while AMMLP shows slightly worse performances and finally SOM 6×4 performs clearly worse in terms of specificity than the other neural networks. In terms of accuracy, AMMLP1, AMMLP2, and AMSOM 9×6 get better values than traditional SOM 6×4 network.

Table 2 SOM and AMSOM results

Classifier	Network	Epochs	SP	SE	AC
SOM	6×4	200	99.70	96.26	97.20
AMSOM	6×4	200	92.00	93.86	93.30
SOM	9×6	200	97.00	97.15	97.10
AMSOM	9×6	200	98.40	98.13	98.20
SOM	6×4	500	99.70	96.67	97.50
AMSOM	6×4	500	92.20	93.25	92.90
SOM	9×6	500	99.60	96.87	97.20
AMSOM	9×6	500	99.60	98.84	99.04
AMMLP1	11/9/4	200	97.79	98.30	98.25
AMMLP2	11/9/4	200	99.60	98.10	98.70

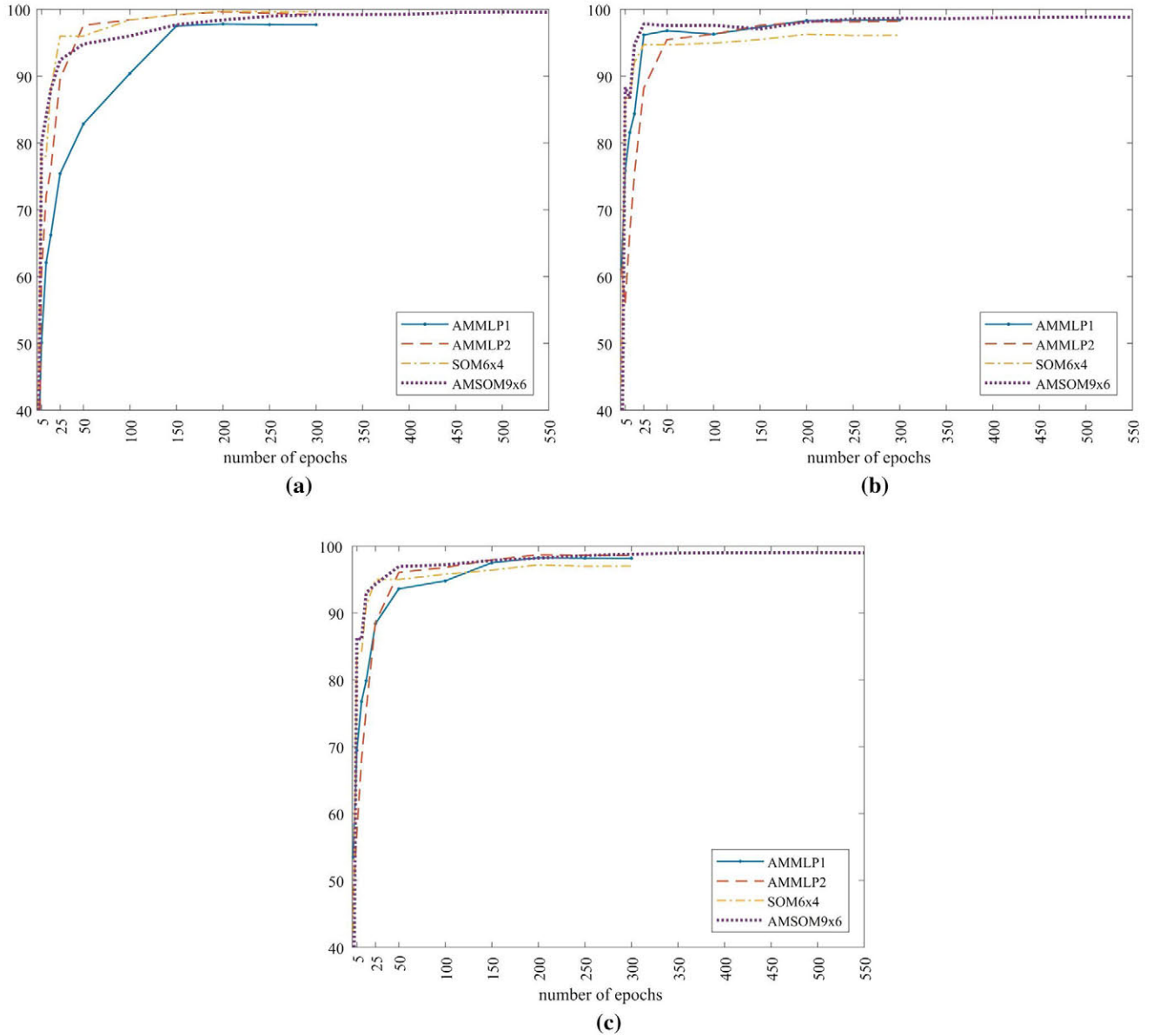


Fig. 2 Performance evolution: **a** specificity, **b** sensibility and **c** accuracy

For comparison purposes, Table 3 gives the classification accuracies of our proposed methods and other methods applied to the same database. As it can be seen from the results, AMMLP2 and AMSOM methods are among the best in classification accuracy. We are going to briefly summarize the different approaches carried out by the researchers related in Table 3. Minami et al. [15] applied the Fourier Transform to the QRS complex obtaining the different frequency components to train an MLP Neural network with BPA for classification purpose. This idea was evolved by Ghorbanian et al. [19] who used continuous wavelet transform instead of fourier transform. The computed wavelet coefficients were used to train the MLP with BPA. Owis et al. [16] applied principal component analysis

(PCA) and Independent Component Analysis (ICA) to extract the number of components that best characterizes the ECG beat signals. This idea was evolved by Yu et al. [17] by using Fast-ICA algorithm to process the QRS signals and then training a Probabilistic Neural Network formed by a radial bases function layer and a competitive layer with the obtained patterns. Alonso et al. [20], Elhaj et al. [21] and Shanshan et al. [23] applied Support Vector Machines instead of Neural Networks for ECG classification after a process of feature selection. Alonso et al. use 13 parameters training patterns that can be divided into three types: time-morphological parameters, frequency parameters and complex parameters. Eljah et al. generates 28 components training patterns using Discrete Wavelet

Table 3 Classification accuracies obtained with our method and other classifiers from the literature

References	Method	Accuracy
Minami et al. [15]	Fourier-NN	98.00
Owis et al. [16]	Blind source separation	96.79
Yu et al. [17]	ICA-NN	98.71
Benchaib et al. [18]	MLP BPA	95.12
Ghorbanian et al. [19]	CWT-NN	99.17
Benchaib et al. [9]	AMMLP1	98.25
Alonso et al. [20]	SVM	99.10
Torres et al. [11]	AMMLP2	98.70
Elhaj et al. [21]	SVM-RBF	98.91
Kyranyaz et al. [22]	1D-CNNs	99.00
Shanshan et al. [23]	FE-SVM	98.41
In this study	SOM	97.50
In this study	AMSOM	99.04

Transform (DWT) + PCA, ICA, and Higher Order Statistics (HOS) to train a SVM with Gaussian kernel (SVM-RBF). Shamsan et al. obtain 30 parameters for each training pattern using a projection matrix with the ECG signal. Recently Kiranyaz [22] have applied the new Convolutional Neural Networks (CNNs) to this problem obtaining remarkable results. The proposed neural structure combines CNN layers with final MLP layers.

4.3 learning evolution

In this section we present some error evolution graphics of the different ANNs proposed. AMMLP networks Error evolution is evaluated at the end of every epoch of training.

In each epoch 900 patterns are presented to the network. In Fig. 3, error evolution is presented for AMMLP1 and AMMLP2.

In the SOM case it has to be notice that we cannot specifically talk about an error because there is no output expected for the neuron so with the objective of observing the correct evolution of the training phase we have used the average in an epoch of the distance between the input pattern and the weights of the winning neuron in each iteration. This learning evolution is showed in Fig. 4, for SOM 6×4 , AMSOM 6×4 , and AMSOM 9×6 .

5 Discussion

As seen in Table 2 results obtained by the SOM are better than the ones obtained by the AMSOM in the case of the 6×4 networks no matter the number of epochs. When we increase the network dimensions to 9×6 , the AMSOM results improve being better than SOM results which remain similarly. It has to be noticed that the AMSOM results also improve in the 9×6 network when increasing the epoch to 500, greater number of epochs does not improve the results. When increasing the size of the network to 12×8 results do not improve. Notice that in this case in the evaluation phase some input patterns activate neurons of the network that has not been activated in the training process so there is no possibility to classify them. Regarding the evolution of specificity, sensitivity and accuracy parameters evolve quickly till 50 epochs and the evolution is much softer stopping in 200 epochs for the majority of the studied networks except for the AMSOM 9×6 .

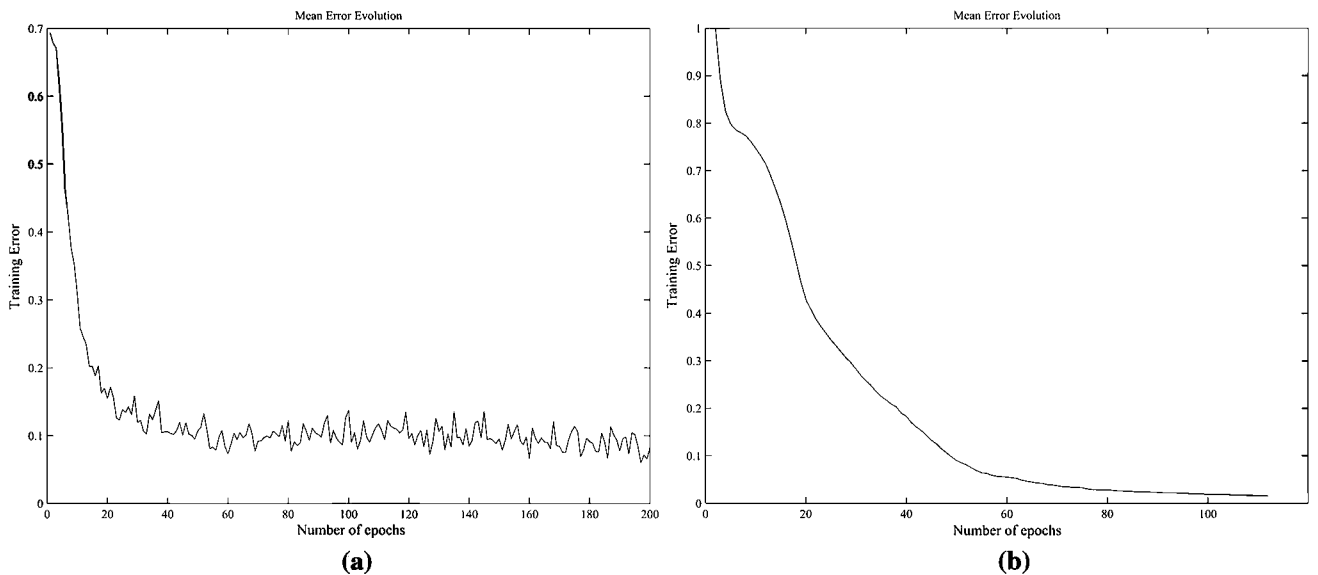


Fig. 3 Error evolution: **a** AMMLP1 and **b** AMMLP2

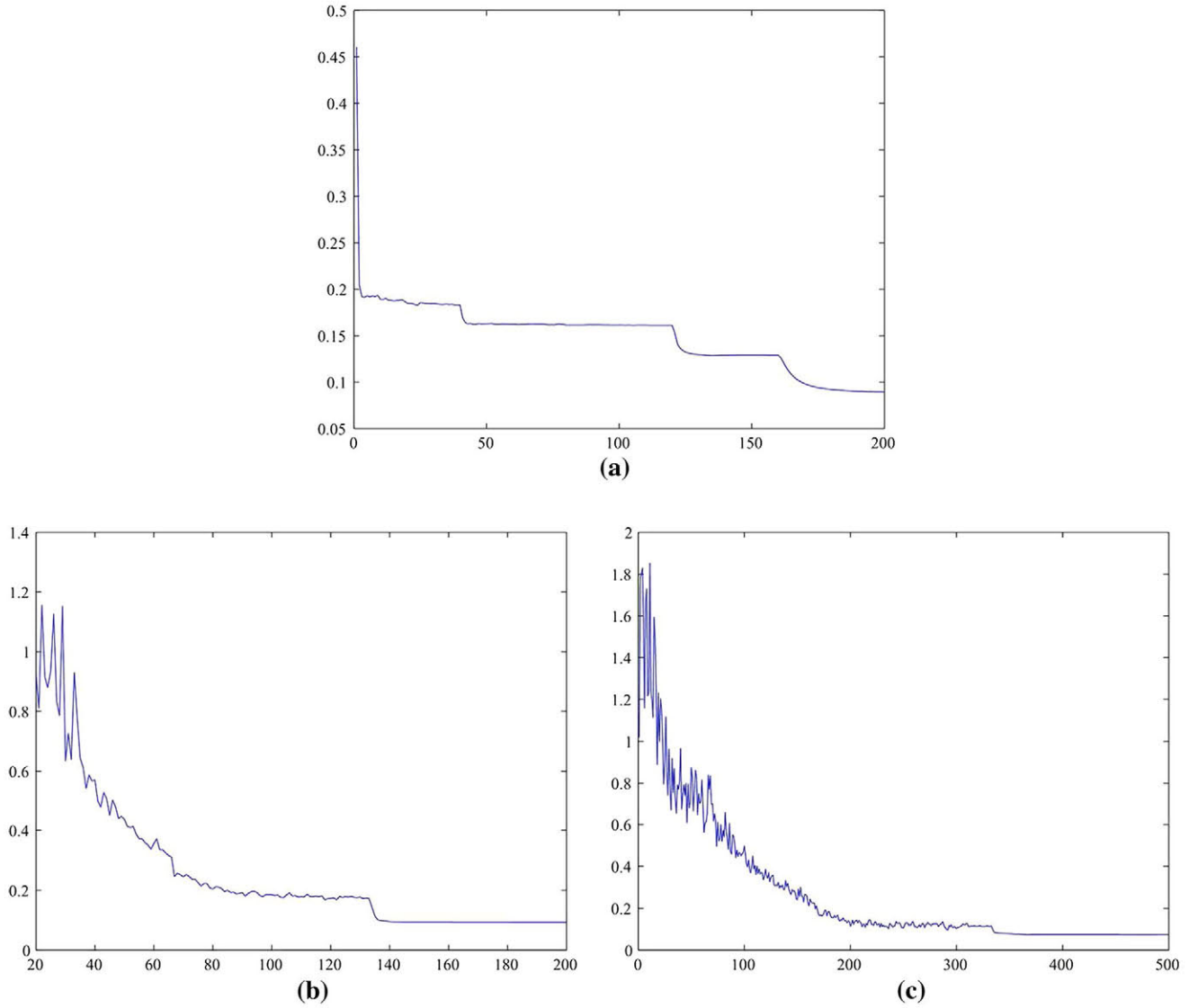


Fig. 4 learning evolution: **a** SOM 6×4 , **b** AMSOM 6×4 and **c** AMSOM 9×6

Regarding the learning evolution view as the average of the distance from the input pattern to the weights of the winning neurons, this distance begins lower in the case of SOMs than in AMSOMs and evolves differently in both cases. For the SOM networks, the evolution is quicker because in the case of AMSOM in some iterations weight vector can be move away from the input pattern recuperating this proximity in the following iterations. Although the learning evolution is slower in AMSOM case, classification rates are better. It also can be seen that at the beginning of the training phase different neurons are activated for each input pattern, but as the training evolves winner neuron becomes stable. Particularity in the AMSOM case, we can observe more variability in the winner neuron than in the SOM case. For the supervised networks case we can observe that the training error decrease quicker in

AMMLP1 than in AMMLP2 but AMMLP2 reaches the final value first. We can also observe that for AMMLP1 case there are a lot of peaks in the error evolution but for AMMLP2 case the error evolution is softer.

6 Conclusions

In this contribution, AMP, artificial metaplasticity, in SOM neural networks has been proposed as a new method to improve training and performance in unsupervised neural networks obtaining satisfactory results. It is remarkable to note that the idea of artificial metaplasticity in unsupervised neural networks is totally new and opens a new potential for the unsupervised ANNs results. Proposed implementation of AMP has been applied to the problem of

arrhythmia classification. It has been shown that although in reduced size SOM networks results were better than those of AMSOM, when increasing the size of the network the AMSOM results become better. The AMSOM results are among the best of the state-of-the-art published algorithms applied to this problem similarly to the case of the AMMLPs algorithms applied to MLP networks. Regarding the learning evolution, AMPs classifiers progress with more variability, although performance results are better. The results indicate that the use of AMP algorithms in unsupervised (SOM) and supervised (MLP) networks is an alternative option for cardiac arrhythmias detection and could be used as a computer-aided detection system for second opinion by physicians, when making their diagnostic decisions.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abraham WC (1996) Activity-dependent regulation of synaptic plasticity (metaplasticity) in the hippocampus. In: *The hippocampus: functions and clinical relevance*. Elsevier, Amsterdam, pp 15–26
2. Jedlicka P (2002) Synaptic plasticity, metaplasticity and BCM theory. *Bratisl Med J* 103(4):137–143
3. Ropero-Pelaez J, Andina D (2012) Do biological synapses perform probabilistic computations? *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2012.08.042>
4. Andina D, Alvarez-Vellisco A, Jevtic A, Fombellida J (2009) Artificial metaplasticity can improve artificial neural network learning. *Intell Autom Soft Comput Spec Issue Signal Process Soft Comput* 15(4):681–694
5. Kinto E, Del-Moral-Hernandez E, Marcano-Cedeño A, Ropero-Pelaez J (2007) A preliminary neural model for movement direction recognition based on biologically plausible plasticity rules. In: *Proceeding IWINAC 2007*, vol 45, no 28, pp 628–636
6. Marcano-Cedeño A, Quintanilla-Dominguez J, Andina D (2011) Breast cancer classification applying artificial metaplasticity algorithm. *Neurocomputing* 74(8):1243–1250
7. Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech. J.* 27:379–423
8. Haykin S (1995) *Neural networks a comprehensive foundation*. MacMillan College Publishing Company, New York
9. Benchaib Y, Marcano-Cedeño A, Torres-Alegre S, Andina D (2013) Application of artificial metaplasticity neural networks to cardiac arrhythmias classification. *Lect Notes Comput Sci* 79(30):181–190
10. Ruck DWH, Rogers SK, Kabrisky M, Oxley ME, Suter BW (1990) The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Trans Neural Netw* 1(4):296–298
11. Torres-Alegre S, Fombellida J, Piñuela JA, Andina D (2015) Artificial metaplasticity: application to MIT-BIH arrhythmias database. *Lect Notes Comput Sci* 91(7):133–142
12. Kohonen T (1982) Self-organized formation of topologically correct feature maps. *Biol Cybern* 43:59–69
13. Kohonen T (1982) Analysis of a simple self organizing process. *Biol Cybern* 44:135–140
14. Moody GB, Mark RG (2001) The impact of the MIT-BIH arrhythmia database. *IEEE Eng Med Biol Mag* 20(3):45–50
15. Minami K, Nakajima H, Toyoshima T (1999) Real-time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network. *IEEE Trans Biomed Eng* 46(2):179–185
16. Owis MI, Youssef ABM, Kadah YM (2002) Characterization of ECG signals based on blind source separation. *Med Biol Eng Comput* 40(5):557–564
17. Yu SN, Chou KT (2008) Integration of independent component analysis and neural networks for ECG beat classification. *Expert Syst Appl* 34(4):2841–2846
18. Benchaib Y, Chikh M (2009) A Specialized learning for neural classification of cardiac arrhythmias. *J Theor Appl Inf Technol* 6(1):81–89
19. Ghorbanian P, Jalali A, Ghaffari A, Nataraji A (2009) An improved procedure for detection of heart arrhythmias with novel pre-processing techniques. *Expert Syst* 29(5):478–491
20. Alonso-Atienza F, Morgado E, Fernandez-Martinez L, Garcia-Alberola A, Rojo-Alvarez J (2014) Detection of life-threatening arrhythmias using feature selection and support vector machines. *IEEE Trans Biomed Eng* 61(3):832–840
21. Elhaj F, Salim N, Harris A, Swee T, Ahmed T (2016) Arrhythmia recognition and classification using combined linear and nonlinear features of ECG signals. *Comput Methods Programs Biomed* 127:5263
22. Kiranyaz S, Ince T, Gabbouj M (2016) Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Trans Biomed Eng* 63:664675
23. Shanshan C, Wei H, Zhi L, Jian L, Xingjiao G (2017) Heartbeat classification using projected and dynamic features of ECG signal. *Biomed Signal Process Control* 31:165–173