Neurocomputing 150 (2015) 43–49

Contents lists available at ScienceDirect

Neurocomputing

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

τ-NEAT: Initial experiments in precise temporal processing through neuroevolution

Pilar Caamaño, Francisco Bellas *, Richard J. Duro

Integrated Group for Engineering Research, Universidade da Coruña, Coruña, Spain

ARTICLE INFO

Article history:
Received 28 October 2013
Received in revised form
26 March 2014
Accepted 7 April 2014
Communicated by Chennai Guest Editor
Available online 2 October 2014

Keywords:
Neuroevolution
NEAT
Time delay neural network
Time series

ABSTRACT

Neuroevolution of Augmenting Topologies (NEAT) has been a very successful algorithm for evolving Artificial Neural Networks (ANNs) that adapt their structure and processing to the task that is required from them. However, this algorithm is not always reliable when handling time related processes and this may be due to its lack of explicit temporal elements within its structure. Of course, NEAT can handle time dependent phenomena through the use of recurrences within the networks it builds, but it is well known that simple recurrences do not easily allow for precise temporal processing due to the history effect they induce on the networks. Many authors have argued for the introduction of other mechanisms, which are also present in natural systems, such as variable or trainable propagation delays in the synapses of the networks that must deal with precise temporal processing. In this paper, we carry out an initial study of a new implementation of NEAT called τ-NEAT that includes the possibility of introducing variable delays in the synapses of the networks NEAT constructs. These delays can affect both, regular direct synapses or recurrent connections. To evaluate the performance of this implementation several tests are carried out over different types of temporal functions and the results of the traditional version of NEAT and τ-NEAT are compared.

1. Introduction

The NEAT (neuroevolution of augmenting topologies) algorithm [1,2] is a widely tested approach for evolving the weights and structure of an artificial neural network (ANN). Its operation is based on the use of history markers in genes to promote crossover between similar topologies. Thus, species or niches in the population are preserved by avoiding reproduction between historically different individuals. Moreover, NEAT starts with simple feed-forward ANNs that contain only input and output neurons and it incrementally increases their complexity through structural mutation operators, the *add connection* mutation and the *add node* mutation [3]. This way, a designer does not need to predetermine the architecture and number of nodes of the ANN needed for a given task or function, and this is very useful when applying ANNs to problems and domains where it is not easy to predetermine the difficulty of the task. In this type of situations, when the network is too small, the function is learnt without too much detail and large errors may arise. On the other hand, when too many nodes are used, if one is not very careful, overtraining may easily occur leading to a lack of generalization capabilities of the networks in the task. As a consequence, having an algorithm like NEAT that automatically and simultaneously grows and trains a network until its function is achieved with the required goodness is of great importance. In fact, this algorithm has been successfully applied to very different domains going from data classification [4,5] to evolutionary robotic design [6], but its main application field has been that of learning in dynamic domains, like video games [7–9] or vehicle crash simulation [10].

Time dependent processing is quite important in many applications and NEAT is able to manage time dependent phenomena through recurrent or feedback connections between neurons that can be inserted using the *add connection* mutation operator. Consequently, NEAT intrinsically supports the generation and training of classical Recurrent Neural Networks, which are quite adept at working with dynamic processes that depend on sequences of events.

However, classical recurrent neural networks (RNN) present several drawbacks when dealing with problems that require precise timing [11], especially when modeling the underlying structure of complex time series, and different approaches have been developed to address them [11,12]. One of the most popular consists in mimicking nature and modeling the length of the synapses through the introduction of synaptic time delays both in the direct and in the recurrent connections, leading to what have been called time delay recurrent neural networks (TDRNN) [13–15].

* Corresponding author at: Integrated Group for Engineering Research, Universidade da Coruña, Campus de Esteiro, Ferrol, Spain.
E-mail addresses: pcsobrino@udc.es (P. Caamaño), francisco.bellas@udc.es (F. Bellas), richard@udc.es (R.J. Duro).

http://dx.doi.org/10.1016/j.neucom.2014.04.077
0925-2312/

The use of delays when modeling dynamic processes is supported by a series of theoretical results in the signal processing literature such as the embedding theorem [16,17]. This theorem states that given an unknown dynamic system:

$$x(n+1) = F(x(n))$$

where $x(n)$ is the d-dimensional state vector of the system and F some function; the information of which we obtain by sampling an observable in time, that is through a temporal series given by

$$Y(n) = h(x(n)) + r(n)$$

being $h()$ a scalar function and $r()$ a noise term, the geometric structure of the multivariate dynamics of the system in the absence of noise can be extended from observable $Y(n)$ to a D-dimensional space as

$$Y_R(n) = [Y(n), Y(n-\tau), \ldots, Y(n-(D-1)\tau)]^T$$

where τ is called the embedding delay.

What is important here is that this theorem states that having an observable $Y(n)$ that corresponds to one component of an unknown dynamical system, the dynamic reconstruction of the signal (in other words a precise model) is possible using vector $Y_R(n)$ from a dimension D onwards. In fact, it can be said that this dimension must verify:

$$D \geq 2d + 1$$

which is a sufficient, but not necessary, condition. The minimum dimension D that permits this reconstruction is called the embedding dimension.

Note that what this means is that when we have a dynamic system characterized by a measured signal, it is only necessary to embed this signal in a higher dimensional space of dimension D by taking D samples of the signal spaced by τ in order to make it predictable or, in other words, its unambiguous modeling feasible.

Thus, according to the embedding theorem the only problem now becomes how to obtain D and τ [18]. Translating it to our problem, the challenge is to obtain these values autonomously for a signal or process that is being modeled. Basically, how to obtain the number of points that must be considered and the temporal spacing between them when they are regularly spaced.

Going one step further, one could hypothesize that, in many cases, lower dimensional embedding spaces could be used if the samples were not evenly spaced in time and, consequently, if one considered an uneven distribution of delays, a lower number of points would be necessary to disambiguate many dynamic processes. In fact several authors have already hinted towards this conclusion [19].

Thus, to produce a good intrinsic model of a signal, it is necessary to be able to determine what points of the signal must be processed together. This is done in time delay based neural networks, and, in particular, in TDRNNs by using the synapses as delayers of signals, in other words, the signal transferred from one neuron to another suffers a delay that is proportional to a value that characterizes the synapse connecting the neurons. These delays are a sort of representation of the different lengths these connections could present, which would have a bearing on the time signals would take to traverse them. Therefore, when a neuron is processing its inputs, it is really processing points of the signal that correspond to different times. Several authors have reported quite interesting results using this approach in fields such as robotics [20,21] or dynamic control [22,23].

However, the TDRNN related algorithms that have been developed do not provide for these networks to be grown and adapt their topology and weights to the problems they are faced with. That is, the designer must usually decide on the number of neurons and the architecture of the TDRNN and a training algorithm is used to provide values for the synaptic weights and delays. This takes us back the problem of how to determine the right size and connectivity of the network, and an obvious solution would be to adapt NEAT so that it can automatically and incrementally generate TDRNN type networks.

Thus the question that is posed here is whether adding the capability of introducing synaptic delays to NEAT, which leads to an algorithm we have called τ-NEAT, can improve the response of the ANNs it produces. That is, we aim to analyze whether such a higher degree of temporal processing is beneficial for NEAT when applied to tasks that involve complex precise temporal patterns, in particular, to time series prediction tasks or precise signal modeling tasks. In other words, does the introduction of time delays allow NEAT to produce better signal modelers?

The paper is organized as follows. Section 2 deals with the formal description of the τ-NEAT algorithm. Section 3 contains the comparison experiments that have been performed using chaotic temporal series and that show how τ–NEAT outperforms NEAT in these complex cases. Finally, Section 4 is devoted to the presentation of the main conclusions of this study.

2. The τ-NEAT neuroevolutionary algorithm

To allow for the introduction of time delays, the original NEAT algorithm was extended, thus creating the τ-NEAT algorithm. τ-NEAT is basically a neuroevolutionary algorithm for growing neural networks that may include recurrent connections and synaptic delays. Fig. 1 displays the structure of a general or prototypic neural network that τ-NEAT may obtain, where it can be observed that it now includes a synaptic delay τ_{ij}, in addition to the synaptic weight w_{ij} corresponding to the synapse between neurons i and j. In fact, this time delay is modeled through a buffer containing the last n input values to that synapsis.

The basic operation of NEAT is described in [1] and it has been slightly modified. Mainly, the synaptic delays have been included in the NEAT chromosome and their value is applied over the buffer establishing a sort of length of the synaptic connection. These synaptic delays are integers, and as such they are included in the chromosome of the networks within the synaptic encoding. This implies that they must have their own parametric mutation operator. Thus, in terms of evolution the τ-NEAT approach works

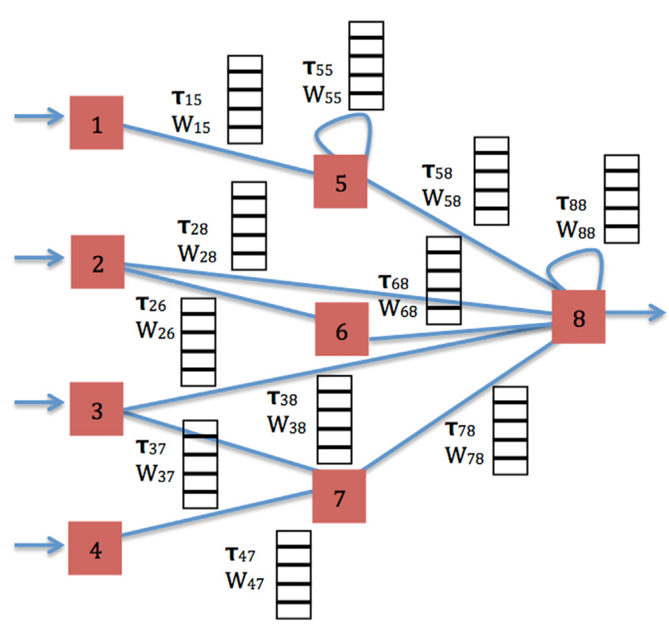

Fig. 1. Structure of a τ-NEAT neural network.