Neurocomputing 143 (2014) 302–311

Contents lists available at ScienceDirect

Neurocomputing

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

Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction

Christopher Smith, Yaochu Jin *

Department of Computing, University of Surrey, Guildford GU2 7XH, United Kingdom

ARTICLE INFO

Article history:
Received 19 March 2014
Received in revised form
30 May 2014
Accepted 31 May 2014
Communicated by V. Palade
Available online 12 June 2014

Keywords:
Hybrid multi-objective evolutionary algorithms
Time series prediction
Ensembles
Selection
Recurrent neural networks

ABSTRACT

Ensembles have been shown to provide better generalization performance than single models. However, the creation, selection and combination of individual predictors is critical to the success of an ensemble, as each individual model needs to be both accurate and diverse. In this paper we present a hybrid multi-objective evolutionary algorithm that trains and optimizes the structure of recurrent neural networks for time series prediction. We then present methods of selecting individual prediction models from the Pareto set of solutions. The first method selects all individuals below a threshold in the Pareto front and the second one is based on the training error. Individuals near the knee point of the Pareto front are also selected and the final method selects individuals based on the diversity of the individual predictors. Results on two time series data sets, Mackey-Glass and Sunspot, show that the training algorithm is competitive with other algorithms and that the final two selection methods are better than selecting all individuals below a given threshold or based on the training error.

1. Introduction

Time series data can be regarded as any information that varies with time and time series forecasting can be described as projecting this time series data into the future [1]. Understanding the behavior of a dynamic system and using this information to predict its future behavior can be very useful and it has been applied to the prediction of electricity demand [2], solar data [3], finance forecasting [4,5] and hydrological forecasting [6].

To confidently predict time series data the dynamics of the system needs to be accurately modeled. Recurrent neural networks (RNNs) are considered to be suitable for time series prediction as they have both feed-back and feed-forward connections. A form of memory is incorporated into the networks, with the states of the neurons from previous iteration steps being stored and used to influence the prediction of data at future iterations. An ensemble of predictors can be used to produce a confident prediction and involves combining many different models to give the final prediction. An ensemble can include information that is not contained in a single model [7] and each member can produce different errors. Ensembles have been shown to provide better generalization performance than single models and the result is a more confident final prediction [8]. Ensembles of RNNs should therefore be well suited for time series prediction.

Evolutionary algorithms (EAs) use a population of solutions to solve a problem, which makes them ideal for creating potential individual predictors in an ensemble, as they result in a population of solutions. EAs have been used to train neural networks [9–12], however, some members in the final population may not be suitable. Therefore, selecting the right models in the population to ensemble is very important.

There has not been much discussion in the literature on how to select ensemble members from the Pareto set of solutions. For example in [13], two slightly different methods for generating ensemble members were reported. The first variant splits training data into two subsets, and the training errors on the two subsets are used as two objectives. In the second variant, random noise is added to the training data set to form the second objective. However, all Pareto-optimal solutions are used to construct ensembles and no selection strategy has been discussed. Therefore, this paper investigates different selection methods for time series prediction tasks. A hybrid multi-objective evolutionary algorithm (H-MOEA) is used to train RNNs and determine their optimal structure and four selection methods are investigated to select individual models from a filtered population of solutions. Firstly, all filtered members are selected. Then members with a low training accuracy and members located near the knee point of the Pareto front are considered. Finally, members with a large degree of diversity in the filtered population are selected. The normalized performance gain (NPG) [14] and the ambiguity term of the error-ambiguity decomposition [15] are used to determine the knee point and the diversity terms, respectively.

* Corresponding author.
E-mail address: yaochu.jin@surrey.ac.uk (Y. Jin).

http://dx.doi.org/10.1016/j.neucom.2014.05.062
0925-2312/

Two time series data sets are used to test the training algorithm and selection methods. The Mackey-Glass is a simulated chaotic time series, while the Sunspot is a real-world time series. The prediction performance is compared with a number of other methods in the literature that have used similar parameter settings.

The rest of the paper is organized as follows. Background of recurrent neural networks and their training methods are presented in Section 2. Information on the specific training and selection algorithms are presented in Section 3. Section 4 presents the experiments, results and discussions. Section 5 concludes the paper and provides details of future work.

2. RNNs for time series prediction

2.1. Time series data and dynamic reconstruction

Univariate time series data is any data set with only one variable. Dynamic reconstruction is concerned with establishing a model that "captures the underlying dynamics" of a system that uses univariate data and can be used to determine future values [16]. The reconstruction vector, $y_R(n) = [y(n), y(n-\tau), .., y(n-(D-1)\tau)]^T$, is defined in terms of the time series observable output $y(n)$ and its delayed versions [16]. τ is the normalized embedding delay. D corresponds to the minimum number of data points needed for dynamic reconstruction and is known as the embedding dimension.

Taken's Theorem states that $D \geq 2d+1$, where d is the dimension of the state space of the system [16]. The value of D may not be known and although increasing D can improve prediction, it could also introduce noise or imperfections into the system. It is therefore desirable to keep D to a minimum. Various values for the embedding delay and dimension have been used for time series prediction; $D=3/\tau=2$ [17], $D=4/\tau=6$ [18] and $D=3/\tau=7$ [19] and values have also been determined using an evolutionary algorithm [20]. However, specific parameter values can be problem dependent.

Feed-forward neural networks [6,21], recurrent radial basis functions [22] and fuzzy models [23] have all been used for time series prediction and have used various embedding delay and dimension values.

2.2. Recurrent neural networks

Neural networks are nonlinear models used to approximate solutions to complex problems and can be used to model any nonlinear function. They acquire knowledge of the system or environment that they are embedded in through observations and use them to train the network [16]. Recurrent neural networks (RNNs) are dynamical systems that are specifically designed for temporal problems, as they have both feed-back and feed-forward connections.

More specifically, a form of memory is incorporated in RNNs, with the states of the neurons from previous iteration steps being stored and used to influence the prediction of future iterations. RNNs have been shown to out perform feed-forward neural networks on time series tasks [24] and have been empirically shown to be successful on time series data sets [17–19,25].

The overall structure of a RNN consists of synaptic connections between the input, hidden and output layers of neurons. Knowledge is represented in a network by the values of these synaptic connections. The states of the neurons are dependent on these free parameters, the inputs to the neurons and the states of the neurons at previous time steps [26]. A RNN can have copies of any neuron in the network from the previous time-step and they can be used to influence the prediction of data at future iterations. The objective of learning is to train the network by adjusting the connection weight values, over several training epochs, to reduce the output error of the network. Training moves the error towards a minimum point on the error surface, which has the free parameters of the network as its coordinates [16].

Gradient descent (GD) [19,27], single-objective evolutionary algorithms (SOEAs) [25,28] and multi-objective evolutionary algorithms (MOEAs) [20,29,30] have been used to train RNNs. Hybrid approaches to neural network training that combine global and local search techniques have also been used [31–33]. Of these hybrid methods, some have used Baldwinian [31] and others Lamarckian learning [34]. The global search is used to find suitable starting weight values and the local search to fine tune them to their optimal value. As well as training the networks weight values, MOEAs can also optimize the structure (number of connections) of the network [11,35].

2.3. Ensemble member generation and selection

When constructing an ensemble it is important that each individual model is both accurate and diverse [36]. There is always a trade-off between these two characteristics [37] and this is summarized by the error-ambiguity decomposition presented by Krogh and Vedelsby [15]. Eq. (1) summarizes this relationship, showing that the generalization error of an ensemble (E) is based on the weighted average of the individual generalization errors ($\overline{E}$) and the weighted average of ambiguities ($\overline{A}$):

$$E = \overline{E} - \overline{A} \tag{1}$$

By reducing each individual's generalization error and increasing their ambiguity, the overall generalization error of the ensemble will reduce. However, by increasing the ambiguity of an individual predictor there is an increase in the individual's error.

Diverse ensemble members can be either implicitly or explicitly created. Different data samples, network parameters and initialization methods, as well as using different learning algorithms have all been used to implicitly create diverse ensemble members [36,37]. To explicitly create diverse neural network ensembles, the ADDEMUP [12], DIVACE [10] and regularization [11] algorithms have all been used.

The use of an MOEA to create diverse ensemble members is very attractive, as the fitness functions can be specifically chosen to optimize conflicting objectives, with the resultant Pareto-optimal solutions providing a trade-off between these objectives and a set of optimal solutions [38].

There may be one model in the Pareto set that is able to perform better than an ensemble, however, there is no clear way of selecting this individual model [37]. There may also be unsuitable/infeasible solutions in the final Pareto set, so a subset may provide better performance and it has been suggested that to ensemble many of the individual members can be better than ensembling them all [39]. Therefore, an MOEA can be used as an indicator of which solutions to use in the ensemble and MOEAs have been used to successfully design neural networks for a variety of problems [7,14].

Ranking all the individual models, based on some criteria or through the use of an optimization process, has been suggested as possible methods of selection [36]. The method used to combine the selected ensemble members is also very important and as stated in [36], the variance as well as the bias of learning algorithms may be reduced through an optimal combination.

The mean of a number of predictors [6,40] or the weighted mean [4,6] of the final output has been used to combine the individual predictors. The weighted median [41] and the weighted sum [42] have also been used, among many others.

Section 3 provides information on the specific H-MOEA and selection algorithms used in this work.