Ensembles of Neural Oscillators

Danil Koryakin, Fabian Schrodt, Martin V. Butz

Cognitive Modeling, Department of Computer Science,
University of Tübingen, Sand 14, 72076 Tübingen, Germany

Abstract. Modularization is a promising direction for the further development of artificial neural networks (ANNs), and a large variety of modularized ANNs have been proposed. Possibly the main advantage of modularization is that, due to the wiring and learning mechanisms, different modules in the ANN can be biased towards developing particular problem solution substructures – allowing the incorporation of a priori problem knowledge. We present a modular Echo-State Network (mESN) architecture, where modules process independent recurrences. The structure enables the modeling of complex, periodic functions by means of an additive combination of elementary oscillations. We compare the mESN to monolithic networks on problems of different complexity and confirm superior performance. Finally, we sketch out potential applications and future work directions.

Keywords: Modular Neural Network, Recurrent Neural Networks, Echo State Network, Periodic Dynamics

1 Introduction

Artificial Neural Networks (ANNs) are very general machine learning systems, which have been applied in many areas of science and engineering. To tune ANNs for solving particular problems, researchers have proposed numerous ANN architectures, with different topologies, types of neurons involved, learning mechanisms, and modularizations by constraining neural connections.

Modularity typically groups neurons in an ANN towards specific functionalities and restricts communication between the groups of neurons in a particular manner. For example, in [7] and [8] ANNs are composed of several feed-forward neural networks. These modules are mediated by a gating network, which chooses the output of the most relevant module. The modules are combined either hierarchically or recursively depending on the problem to be solved. Such modular networks have been applied to engineering applications, such as trajectory modeling [8] and recognition [3, 7]. More complex modularized ANNs have been considered in cognitive science for understanding processes in the human brain [9] or for controlling complex humanoid robots [1].

In this work we propose a modular, recurrent ANN (RNN) architecture for modeling time series. Although our architecture generally allows the usage of any type of RNN as a module, we focus on Echo-State Networks (ESNs). Thus, we call the investigated architecture modular ESN (mESN). As it was shown in [10], despite their compact size, ESNs are able to model quite complex oscillators. They are flexible and can be driven either by a signal from alternative input neurons or operate as autonomous oscillators.

The possibility to use ESNs as autonomous oscillators makes them especially suitable for applications where once learned target dynamics must be reproduced on request. In such applications the networks can serve as a memory for complex time series. Compactness of ESNs would bring added value to such devices. Despite this potential benefit, applications for input-less reproductions of time series are sparse. They include applications for cyclic rehearsal [4], [5] and for artificial problems [4], [5], [11].

In the next section we detail the mESN architecture. Section 3 evaluates the mESN in the task of reproducing of mixtures of periodic signals and analyzes its performance. Section 4 sketches possible practical applications of the presented model. In the conclusions, we summarize our work and draw future research perspectives.

2 Modular ESN

The modular Echo-State Network (mESN) consists of several modules, each of them being an ESN. These modules do not have direct connections to each other and therefore operate independently. The mESN is shown schematically in Figure 1. The output of an mESN is computed as a linear combination of outputs of its modules as follows.

$$o(n) = \sum_{i=1}^{M} r_i o_i(n) \tag{1}$$

where M is the number of modules, $o_i(n)$ is the output of the i^{th} module at time step n, and r_i is its responsibility for generating the output.

Using standard ESNs as modules in mESN is particularly advantageous for modeling time series that consist of multiple oscillators. ESNs have been shows to be able to produce accurate, oscillating output signals for a long period of time without further external stimulation, solely drive by their own output feedback. In mESN, the output signal $o_i(n)$ of one ESN module i is computed as follows:

$$o_i(n) = \mathbf{W}_{OUT,i}[\mathbf{f}_i\left(\mathbf{W}_i \mathbf{x}_i(n-1) + \mathbf{W}_{OFB,i} o_i(n-1)\right)], \tag{2}$$

where n is the current time step, $\mathbf{x}_i(n-1)$ is the vector of reservoir states at the previous time step, $\mathbf{W}_i$ is the matrix of reservoir weights, $\mathbf{f}_i$ is the vector of activation functions of all reservoir neurons, $\mathbf{W}_{OUT,i}$ is the matrix of output weights, and $\mathbf{W}_{OFB,i}$ is the matrix of output feedback weights. Expression (2) is derived from the more general expression of ESNs, ignoring signals from possibly additional input neurons. The general expression for updating an ESN output, rules for designing the matrix $\mathbf{W}$ as well as a training procedure for the standard ESNs can be found in [6].

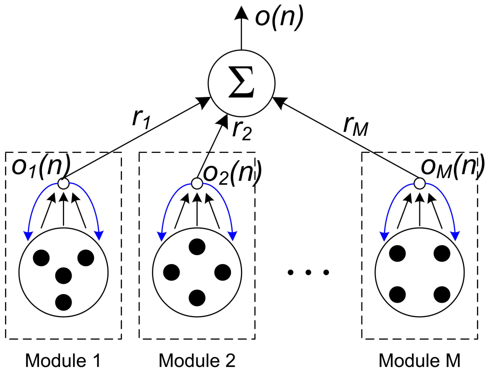

Fig. 1. Structure of the mESN consisting of M modules. Responsibilities r_i of the modules define a portion of each module in the total network output $o(n)$, which is computed as a linear combination of the individual module outputs.

The design of the mESN requires the choice of macro parameters for each ESN module as well as a the independent training of the output weights $\mathbf{W}_{OUT,i}$. To train the modules, a preliminary decomposition of a training sequence Y into a set of sequences is necessary:

$$Y = (Y_1 \oplus Y_2 \oplus ... \oplus Y_M) \tag{3}$$

where Y_i is the time series assigned to the i^{th} module. The operator $\oplus$ denotes an application-dependent split of every value $y(n)$ of the sequence Y into a set of values $y_i(n)$, each of them belonging to the corresponding component Y_i at the current time step n.

In our study the decomposition was performed by an expert using a priori knowledge about the target time series. Thus, additional information about a problem was incorporated into the model during the design phase. A similar approach was described elsewhere [12, 2]. In these studies feedforward neural network modules were trained independently of each other on already segmented sequences. In our study, ESN modules were also trained independently of each other using the training procedure

$$\mathbf{W}_{OUT,i} = \mathbf{M}_i^{-1}\mathbf{T}_i, \tag{4}$$

where $\mathbf{M}_i^{-1}$ is the inverted matrix of states of the i^{th} module on its training sequence and $\mathbf{T}_i$ is the corresponding sequence of target outputs.

Modularity provides an opportunity to choose the macro parameters independently for each module. These parameters include the module's reservoir size, a spectral radius of the matrix $\mathbf{W}_i$, and an interval for initialization of the feedback weights $\mathbf{W}_{OFB,i}$. An independent parameter choice is especially useful for time series that consist of components with very different characteristics, such as very slow and fast dynamics – where larger and smaller spectral radii are more suitable, respectively – or differing component complexities – where the reservoir sizes should be adapted accordingly [10].

3 Experiments

In this section we demonstrate the merit of mESNs when modeling time series of different complexity. We focus on smooth sine waves and triangular signals. The latter are not continuously differentiable and consequently more challenging. We compared the performance of our mESN implementation with our corresponding single-reservoir ESN implementation.

3.1 Experimental Setup

In the experiments we focus on dynamic reservoirs that were randomly generated for different combinations of ESN parameter settings. These parameters were reservoir size, connectivity of a dynamic reservoir, and scaling of the feedback weights $\mathbf{W}_{OFB}$. Ranges of connectivity and of $\mathbf{W}_{OFB}$ were the same for standard ESNs and for the mESNs, and were set to $\{0.1, 0.2, ..., 1.0\}$ and $\{10^{-10}, 10^{-9}, ..., 3.9, 4.0\}$, respectively. Ranges of reservoir sizes were different for the evaluated ESNs and mESNs. Modules in mESNs were equipped with only 4 to 10 reservoir neurons. The single-reservoir in the ESNs had from 10 to 100 neurons. For each combination of the parameter values, 500 networks were generated, trained, and evaluated on a test sequence. Networks with the smallest normalized root mean squared errors (NRMSE) were chosen as ESN modules for the mESN. The mESN performance was compared with the performance of the best single-reservoir ESN.

We considered three sets of target dynamics. The first time series was composed of sine waves, the second was composed of triangular signals, and the third was composed of a mixture of the two. For each target dynamics, a sequence of 700 time steps was generated, which was split into a washout sequence (the first 100 time steps), a training sequence (following 300 steps), and a test sequence (the last 300 time steps).

Mixtures of sine waves are known in the literature as Multiple Superimposed Oscillators (MSO). The number of sine waves defines the complexity of a dynamics: more sine waves constitute more difficult dynamics. The whole family of the MSO dynamics can be described as follows:

$$y(n) = \sum_{i=1}^{s} sin(\alpha_i n), \tag{5}$$

where s is the number of sine waves and α_i specify their respective frequencies. We generated the sequences for the following standard set of the frequencies $\alpha_1 = 0.2$, $\alpha_2 = 0.311$, $\alpha_3 = 0.42$, $\alpha_4 = 0.51$, $\alpha_5 = 0.63$, $\alpha_6 = 0.74$, $\alpha_7 = 0.85$, and $\alpha_8 = 0.97$. Because of the smoothness of the individual components, the MSOs are moderately non-linear and continuously differentiable over the whole time axis.

Like the MSOs, the mixtures of triangular signals (MTS) are linear combinations of their components. Each component is a periodic triangular signal characterized by a period and an amplitude. The amplitude of all components was one. The periods were set to the following integers {32, 24, 20, 12, 8, 4}, which yields periods similar to the MSO ones. Despite this similarity, the MTS are more difficult because of much higher non-linearity at peaks of the triangular signals. Figure 2 shows curves of the least and most complex MTS dynamics, MTS2 and MTS6. As can be seen, an MTS with more components resembles a chaotic attractor. But in contrast to known chaotic attractors, both MSOs and MTSs have internal structures that are very suitable for modularization.

For example, MTS2 consists of two distinct components Y_1 and Y_2, with periods 32 and 24 time steps, respectively. To model MTS2 with mESN, two ESN modules are employed. Each module is then trained on one of the components and the best ESN is chosen, respectively, by means of the stochastic search procedure detailed above. A dynamic reservoir for every candidate was updated using the formula (2) at every time step of the washout and training sequences. The output weights $\mathbf{W}_{OUT,i}$ of module i were trained using formula (4) given the training sequence Y_i, that is, the corresponding target values $o_i(n)$ and $\mathbf{T}_i$, which appear in formulae (2) and (4), were taken from the sequence Y_i.

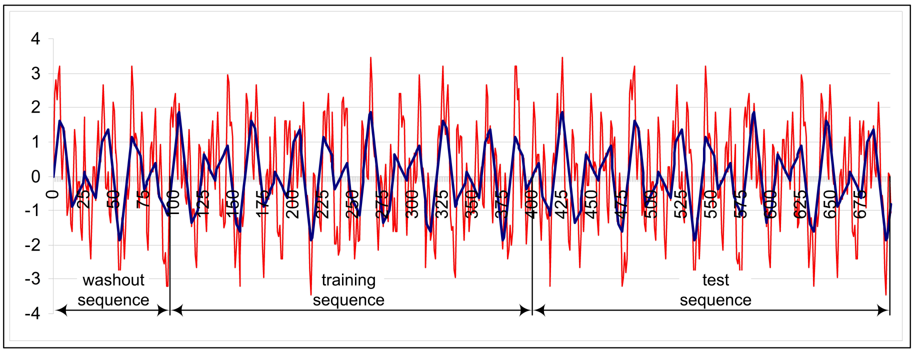

Fig. 2. Mixtures of triangular signals showing the simplest (MTS2, blue curve) and most complex dynamics (MTS6, red curve) considered. The sequences are split into washout, training and test intervals.

4 Results

Table 1 shows reached performances and sizes of the best mESNs and ESNs found for time series with two, four, six, and eight dynamic components. Performance of the standard ESNs varies over a wide range from 10^{-12} to 10^{-2}. Standard ESNs had big difficulties on more complex dynamics and could not model the most difficult target MST4.4 at all. At the same time, as expected,

Table 1. Performance of the best mESN and ESN on the respective target dynamics of different complexity. MSOs are mixtures of sine waves, "TriX" stands for a mixture of X different triangular signals, and "MSTX.Y" stands for a combination of X sine waves with Y triangular signals. The number of neurons is the summed number of reservoir neurons in all modules in the best mESN / ESN found for the respective problem.

mESN			
Number of Components	MSO & NRMSE & size	Triangular & NRMSE & size	MST & NRMSE & size
2	MSO2: 5.62×10^{-10} with 8 neurons	Tri2: 7.99×10^{-7} with 10 neurons	MST1.1: 8.08×10^{-7} with 10 neurons
4	MSO4: 6.47×10^{-10} with 16 neurons	Tri4: 8.37×10^{-7} with 18 neurons	MST2.2: 8.63×10^{-7} with 18 neurons
6	MSO6: 7.83×10^{-10} with 24 neurons	Tri6: 9.45×10^{-7} with 26 neurons	MST3.3: 1.04×10^{-6} with 26 neurons
8	MSO8: 1.07×10^{-9} with 32 neurons	-	MST4.4: 1.59×10^{-6} with 34 neurons
ESN			
Number of Components	MSO & NRMSE & size	Triangular & NRMSE & size	MST & NRMSE & size
2	MSO2: 2.51×10^{-12} with 5 neurons	Tri2: 2.42×10^{-6} with 30 neurons	MST1.1: 3.29×10^{-6} with 20 neurons
4	MSO4: 5.72×10^{-8} with 9 neurons	Tri4: 7.31×10^{-4} with 90 neurons	MST2.2: 3.16×10^{-3} with 90 neurons
6	MSO6: 8.43×10^{-5} with 14 neurons	Tri6: 1.83×10^{-3} with 100 neurons	MST3.3: 5.80×10^{-2} with 90 neurons
8	MSO8: 2.73×10^{-4} with 68 neurons	-	MST4.4: no ESN could model the dynamics

the complexity of the time series had only a minor impact on the performance of the mESNs. Their test errors varied only within four orders of magnitude. Also the most difficult dynamic, MST4.4, which consists of four sine components and four triangular components, was solved with high accuracy. The high performance of the mESNs was also reached thanks to an individual choice of critical parameters for each component, which was especially helpful on mixtures of MSOs and MTSs. Whereas the sine waves needed similar $\mathbf{W}_{OFB}$ settings below 10^{-6}, some triangular signals required reservoir neurons to operate in a saturation range with $\mathbf{W}_{OFB}$ up to 3.8. Such a wide parameter spread is infeasible for a single-reservoir ESN.

Besides the performance gain, modularity is very favorable for the generation of compact models of complex target dynamics. The largest reduction in model size was observed in the sequence "Tri4", where mESN required 18 neurons whereas 90 neurons were needed with standard ESNs. In order to reach higher performance, a single-reservoir ESN increases a variety of reservoir states through formation of complementary neural paths. This automatically requires a larger reservoir. On the contrary, in the mESN splitting target components leads to decoupling of internal dynamics. As a result, smaller ESN modules provide a sufficient variety of internal dynamics for the corresponding target component. However, splitting the components causes a slight size overhead in mESNs. This can be seen in the easiest target dynamics, such as MSO2, where mESN required 8 neurons while a single-reservoir ESN solved the problem best with only 5 neurons.

5 Discussion

The experiments showed advantages of an mESN ensemble over a single-reservoir ESN. The main advantage is the possibility to decouple internal dynamics from each other. Like other modular architectures, it allows incorporating a priori knowledge to do a focused choice of modules' parameters and to reach higher accuracy with more compact models. Besides that, such an organization offers flexibility and robustness for potential applications. Modules of different types can be plugged into the ensemble. Switching off a malfunctioning module allows avoiding an abrupt reduction in system performance.

The mESN is useful for modeling periodic patterns of any complexity and any period, especially when consisting of different components. Currently we see at least three potential applications. The first one is an analysis of a time series through mESN synchronization. It will produce an mESN whose active ESN modules will indicate which components are present and how they are mixed with each other.

The second application is a flexible control of a robot arm shown in Figure 3. Its end effector may be required to draw a complex trajectory periodically. Each ESN module may be linked to its own joint and produce a specific trajectory independently of the other modules – leading to the generation of the overall, target trajectory with the end effector. Alternatively, the end-effector trajectory may be controlled by selectively switching mESN components on and off over time, possibly enabling the generation of digits on any surface and with any surface orientation that is reachable with the robot arm.

Another relevant application is the generation of central pattern generators (CPG), where primitive rhythmic signals are combined into more complex patterns. An mESN will represent a CPG with a population of tiny ESN modules.

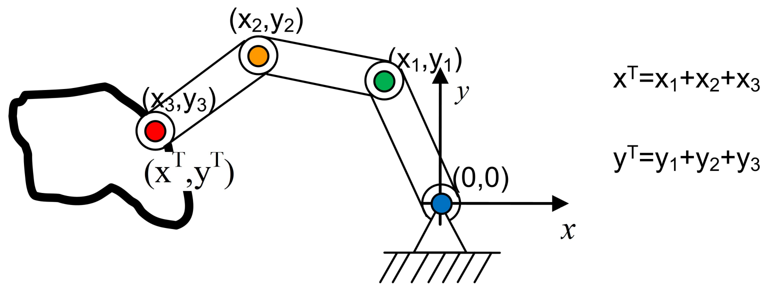

Fig. 3. mESN-based loopless control of a robot arm. Coordinates (x^T, y^T) of the end effector (red point) on a trajectory (thick line) are a sum of projections of coordinates of individual joints. (x_i, y_i) are coordinates of i^{th} joint in a coordinate system with the origin at the $(i-1)^{th}$ joint. Coordinates (x_i, y_i) are output of the i^{th} ESN module.

A linear combination of their outputs will be used to tune the CPG to a target behavior. CPG parameterization may be realized by augmenting the individual neural oscillators with input neurons.

6 Conclusions and Outlook

In this paper we presented the modularized echo state network architecture mESN. In mESN modules are combined without a switching block common for mixtures of experts. The independent operation of the neural oscillators realizes component decoupling, enabling local parameter and meta parameter optimization for each module and time series component. The proposed model is useful for practical applications that deal with decomposable and switching processes. Currently, we are investigating possibilities to tune an ensemble of neural oscillators after changes in the target dynamics, such as amplitude and phase. Moreover, we are working on automatizing the mESN modularization.

References

1. Byadarhaly, K.V., Perdoor, M.C., Minai, A.A., A modular neural model of motor synergies, Neural Networks 32, pp. 96–108 (2012)
2. Gradojevic, N., Gencay, R., Kukolj, D., Option Pricing With Modular Neural Networks, IEEE Transactions On Neural Networks 20, pp. 626–637 (2009)
3. Happel, B.L.M., Murre, J.M.J., The Design and Evolution of Modular Neural Network Architectures, Neural Networks 7, pp. 985–1004 (1994)
4. Holzmann, G., Hauser, H., Echo State Networks with Filter Neurons and a Delay&Sum Readout, Neural Networks 23, pp. 244–256 (2010)
5. Jaeger, H., The "echo state" approach to analysing and training recurrent neural networks, Technical Report 148, German National Research Institute for Computer Science (2001)
6. Jaeger, H., A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach, Technical Report 159, German National Research Center for Information Technology (2002)
7. Jacobs, R.A., Jordan, M.I., Barto, A.G., Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks, Cognitive Science 15, pp. 219–250 (1991)
8. Jordan, M.I. Jacobs, R.A., Hierarchical mixtures of experts and the EM algorithm, Proceedings of 1993 International Joint Conference on Neural Networks, pp.1339–1344 (1993)
9. Kharratzadeh, M., Shultz, T., Neural-network modelling of Bayesian learning and inference, 35th Annual Conference of the Cognitive Science Society, pp. 2686–2691. Berlin, Germany, Cognitive Science Society (2013)
10. Koryakin, D., Lohmann, J., Butz M.V., Balanced echo state networks, Neural Networks 36, pp. 35–45 (2012)
11. Reinhart, F., Steil, J., Reservoir regularization stabilizes learning of Echo State Networks with output feedback, In Proceedings of ESANN2011, European Symposium on Artificial Neural Networks (2011)
12. Soltani, S., On the use of the wavelet decomposition for time series prediction, Neurocomputing 48, pp. 267–277 (2002)