Grouped Multi-Layer Echo State Networks with Self-Normalizing Activations

Robert Wcisło[1] and Wojciech Czech[1][0000−0002−1903−8098]

Institute of Computer Science, AGH University of Science and Technology, Kraków, Poland
czech@agh.edu.pl

Abstract. We study prediction performance of Echo State Networks with multiple reservoirs built based on stacking and grouping. Grouping allows for developing independent subreservoir dynamics, which improves linear separability on readout layer. At the same time, stacking enables to capture multiple time-scales of an input signal by the hierarchy of non-linear mappings. Combining those two effects, together with a proper selection of model hyperparameters can boost ESN capabilities for benchmark time-series such as Mackey Glass System. Different strategies for determining subreservoir structure are compared along with the influence of activation function. In particular, we show that recently proposed non-linear self-normalizing activation function together with grouped deep reservoirs provide superior prediction performance on artificial and real-world datasets. Moreover, comparing to standard tangent hyperbolic models, the new models built using self-normalizing activation function are more feasible in terms of hyperparameter selection.

Keywords: Echo State Network · Self-Normalizing Activation · Reservoir Computing · Deep ESN.

1 Introduction

Echo State Networks (ESN), being the leading representative of Reservoir Computing (RC) framework are intensively studied in last years owing to convenient learning paradigm and promising results in modeling dynamic systems. They are classified as a constrained variant of Recurring Neural Networks (RNNs), for which weights of hidden layer are not trained but initialized once, typically randomly [8]. ESNs were successfully applied for univariate/multivariate time series prediction and currently provide proven framework for modeling difficult chaotic systems [11]. Most recent applications of ESNs include medical multivariate time series [3], human activity sensor data [10], electrical load [2] or robot navigation [4].

In recent years, we observe growing number of works addressing the role of reservoir topology. In particular, deep ESNs were introduced as a robust variant of ESN utilizing the effect of layering, which allows to capture multi-scale time characteristics of an input signal [7]. Different structural constraints can

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_9

be applied to the graph of hidden layer links resulting in grouped reservoirs [6], growing subreservoirs organized in blocks [12], small-world reservoirs with segregated input/output nodes [9] or deep reservoirs with simple non-radom structure [5]. The models exhibiting non-trivial, decoupled topology were proven to outperform shallow ESNs in time-series prediction on the most popular benchmark datasets. At the same time, important results regarding non-linear activation function were published in [13]. The authors introduced new, non-linear activation function called Self-Normalizing Activation (SNA). The SNA function projects network pre-activations onto hyperspehere and ensures stable behaviour of ESN, which cannot enter chaotic regime and is less sensitive to perturbations of hyperparameters.

Motivated by recent findings regarding reservoir structure as well as promising properties of SNA, we study multi-layer ESNs with SNA and *tanh* activations. We also propose hybrid ESN architecture with layerying and grouping, then based on benchmark datasets, show that it can achieve superior prediction performance comparing to shallow architectures with the same number of nodes and deep architectures with *tanh* activation function. The key contributions of this work are new multi-layer grouped ESN architecture with SNA activation and their comparison with previously analyzed deep, grouped and shallow ESNs, as well as standard models such as LSTM or moving average. In addition, we present a modular software framework for ESN configuration and testing together with the associated open-source Python library. The library (AutoESN) allows for building different types of ESNs including stacked, grouped, growing and provides implementation of kernel and SNA activation functions.

2 ESN architectures

Herein, we describe shallow ESNs and provide details on SNA function together with the comment on its parameter *activation radius*. We also present new, grouped multi-layer architecture with SNA activation, which will be tested against standard ESN models with tangent-hyperbolic activation further in experimental section.

2.1 Shallow Echo State Network

Shallow Echo State Network is defined using following state transition equations:

$$\boldsymbol{x}(n) = (1-\alpha)\boldsymbol{x}(n-1) + \alpha\tilde{\boldsymbol{x}}(n) \tag{1}$$

$$\tilde{\boldsymbol{x}}(n) = f(\boldsymbol{W}_{in}[1;\boldsymbol{u}(n)] + \boldsymbol{W}_x\boldsymbol{x}(n-1)) \tag{2}$$

$$\boldsymbol{y}(n) = \boldsymbol{W}_{out}[1;\boldsymbol{u}(n);\boldsymbol{x}(n)] \tag{3}$$

where $\boldsymbol{u}(n) \in \mathbb{R}^{N_u \times 1}$ is an input signal, $\boldsymbol{y}^t(n) \in \mathbb{R}^{N_y \times 1}$ - an output signal, $n \in \{1,\ldots,T\}$ - discrete time, T - a number of data points in the training dataset, N_x - a number of nodes in the reservoir, N_u - dimensionality

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_9

of an input, N_y - dimensionality of output, $\boldsymbol{x}(n) \in \mathbb{R}^{N_x \times 1}$ - vector of reservoir node activations, $\boldsymbol{y}(n) \in \mathbb{R}^{N_y \times 1}$ - network output (trained ESN model), $\boldsymbol{W}_{in} \in \mathbb{R}^{N_x \times (N_u+1)}$ - input weight matrix, $\boldsymbol{W}_x \in \mathbb{R}^{N_x \times N_x}$ - recurrent weight matrix, $\boldsymbol{W}_{out} \in \mathbb{R}^{N_y \times (1+N_u+N_x)}$ - readout weight matrix, $f : \mathbb{R}^{N_x} \to \mathbb{R}^{N_x}$ - activation function, $\alpha \in (0, 1]$ - leaking rate and $[\cdot;\cdot]$, $[\cdot;\cdot;\cdot]$ - vertical vector (matrix) concatenation operators. Most typically, $\boldsymbol{W}_{in}$ is fully-connected or dense, while $\boldsymbol{W}_x$ is sparse. Spectral radius ρ is used to scale initial weights of $\boldsymbol{W}_x$ as follows: generating random weights, calculating maximal absolute eigenvalue of $\boldsymbol{W}_x$, dividing all elements of $\boldsymbol{W}_x$ by this value and scaling obtained matrix with ρ. In case of *tanh* activation function, which currently dominates in practical applications of ESNs, spectral radius and input scaling were identified as the most important hyperparameters affecting non-linearity and memory. Nevertheless, in case of different activation functions such as SNA, the observations are different.

2.2 Self-Normalizing Activation function on Hyper-Sphere

ESN models were originally built using tangent hyperbolic activation functions, which are applied element-wise on a pre-activation vector. More recently, the self-normalizing activation (SNA) function was proposed as the robust alternative to *tanh*, providing stable predictions and reducing model sensitivity to hyperparameters [13]. It is defined as follows:

$$\boldsymbol{a}(n) = \boldsymbol{W}_{in}[1; \boldsymbol{u}(n)] + \boldsymbol{W}_x \boldsymbol{x}(n-1), \tag{4}$$

$$\tilde{\boldsymbol{x}}(n) = r \frac{\boldsymbol{a}(n)}{\|\boldsymbol{a}(n)\|}, \tag{5}$$

where $\boldsymbol{a}(n)$ is pre-activation vector, $\tilde{\boldsymbol{x}}(n)$ is post-activation vector used in Eq. 1, $\|\cdot\|$ is norm and r is an activation radius - the parameter of SNA function. SNA is the global function (neuron state depends on the states of other neurons), which projects pre-activation vector $\boldsymbol{a}(n)$ into $(N_x - 1)$-dimensional hyper-sphere of radius r. One of interesting SNA properties is that it brings non-linearity and at the same time ensures high memory capacity - on the level of the one exhibited by linear activations. Therefore, it seems to be the best solution for memory non-linearity trade-off. In addition, SNA guarantees that ESN does not display chaotic behaviour as the maximum Lapunov exponent is always zero. Those important characteristics of SNA were proven theoretically and also confirmed by experiments [13]. It can be shown that memory capacity of reservoir with SNA depends on the product $r\rho(\boldsymbol{W}_x)$, therefore in the experimental setup one of those parameters can be fixed. At the same time, scaling factor $\frac{r}{\|\boldsymbol{a}(n)\|}$ (see Eq. 5) significantly reduces the effect of input scaling s. So far, SNA was tested on shallow ESN architectures. Considering its global nature, which can further enrich dynamics reflected by subreservoirs, we decided to study the effects of introducing SNA in more complex ESN architectures.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_9

2.3 Grouped Deep Echo State Network

Combining deep, decoupled ESNs [6] with Self-Normalizing Activation function, we propose new architecture called grouped deep ESN (*gdESN*), which uses two-dimensional organization of subreservoirs. Each subreservoir can be characterized by different hyperparameters: size, leaking rate, activation radius, spectral radius, input scaling and sparsity. The state transitions of *gdESN* are described using equations:

$$\boldsymbol{x}^{(i,j)}(n) = (1 - \alpha^{(i,j)})\boldsymbol{x}^{(i,j)}(n-1) + \alpha^{(i,j)}\tilde{\boldsymbol{x}}^{(i,j)}(n), \tag{6}$$

$$\tilde{\boldsymbol{x}}^{(i,j)}(n) = f(\boldsymbol{W}_{in}^{(i,j)}[1; \boldsymbol{v}^{(i,j)}(n)] + \boldsymbol{W}_{x}^{(i,j)}\boldsymbol{x}^{(i,j)}(n-1)), \tag{7}$$

$$\boldsymbol{v}^{(i,j)}(n) = \begin{cases} \boldsymbol{u}(n) & \text{if } j = 1 \\ \boldsymbol{x}^{(i,j-1)}(n) & \text{if } j > 1, \end{cases} \tag{8}$$

where N_g is the number of groups ($1 < i \leq N_g$), N_l - the number of layers ($1 < j \leq N_l$) and $\alpha^{(i,j)}$ is leaking rate for i-th group, j-th layer. For *gdESN* with SNA, the function $f \equiv SNA$ but in experimental section we also consider hyperbolic tangent activations.

$$\boldsymbol{y}(n) = \boldsymbol{W}_{out}[1; \boldsymbol{x}^{(1,1)}(n); \ldots; \boldsymbol{x}^{(1,N_l)}(n); \ldots; \boldsymbol{x}^{(N_g,1)}(n); \ldots; \boldsymbol{x}^{(N_g,N_l)}(n)] \tag{9}$$

The node activations $\boldsymbol{x}^{(i,j)}(n)$ from all subreservoirs are concatenated, therefore $\boldsymbol{W}_{out} \in \mathbb{R}^{N_y \times (N_x N_g N_l + 1)}$. The new architecture allows for utilizing decoupled dynamics of grouped ESNs (*gESN*), ordered time-scale organization of deep ESNs (*dESN*) and stability of SNA at the cost of more difficult hyperparameter optimization. Nevertheless, thanks to properties of SNA, the search space can be reduced by fixing input scaling and spectral radius. Moreover, we also gain better computational efficiency. For the fixed total number of neurons N, multiple subreservoirs reduce computational complexity of state updates with $\mathcal{O}(N^2)$ being the cost of matrix-vector multiplication for shallow architecture and $\mathcal{O}(N^2/(N_g N_l))$ - the cost for *gdESN* with N_g groups and N_l layers.

3 AutoESN library

As a software contribution of this work we created open-source Python library called AutoESN. This library uses PyTorch and provides comprehensive tools for configuring and training decoupled ESN architectures. Comparing to other libraries, like DeepESN [1] our software is based on PyTorch framework, has GPU support and is more flexible in terms of reservoir configuration. DeepESN supports deep and shallow ESNs as well as weights initialization and fixed activation functions, which cannot be easily adapted.

In case of AutoESN, we developed the tool for creating different types of decoupled ESNs (including *gdESN*), which enables to mix reservoirs with different kinds of readouts and different custom activation functions. The library consists

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_9

of two core modules: `Reservoir` and `Readout`. At the core of `Reservoir` module the user can find `GroupedDeepESN` class, which enables creating vanilla ESNs, as well as grouped, deep and grouped deep architectures. Weights initialization and activation functions are separated from reservoir logic. This way, changing regular ESN into subreservoir ESN or into the one having more complex structures of the weight matrices (for example arbitrary graphs) can be achieved with just a few extra lines of code. One can also easily create and exchange activation functions (SNA function is supported natively). The code and documentation of AutoESN library can be accessed on `https://github.com/Ro6ertWcislo/AutoESN`.

4 Results

Herein, we present experimental comparison of shallow, deep and grouped ESN architectures (including newly proposed *gdESN*) with SNA activation function. The test setup is based on univariate time series prediction for selected benchmark datasets: Mackey Glass (MG) System, Monthly Sunspot Series and Multiple Superimposed Oscillators (MGO). The details regarding each dataset as well as hyperparameter selection can be found under `https://github.com/Ro6ertWcislo/AutoESN`.

The activation functions considered in this work exhibit different sensitivity to hyperparameters. The behaviour of tangent hyperbolic function highly depends on input scaling, while for SNA its effect is reduced due to normalization factor. Therefore, in experiments we used two different hyperparameter setups as an input to grid search. Each tested model, shallow or decoupled had the total number of 1000 neurons (with the small deviations resulting from subreservoir integer sizes). The best configuration was selected based on minimal NRMSE achieved for validation dataset.

This section describes the results of one step ahead prediction of univariate time series. The experiments on memory capacity and input scaling are documented on the Github page. In addition to ESN models, we also show the best results obtained for simple moving average and LSTM model (with parameters optimized using validation dataset).

Time-Series Prediction The results of one step ahead prediction obtained for three datasets (MG, Sunspot, MSO) are presented in Table 1. We report average and minimal value of NRMSE achieved for relevant test sets by 8 different models. Each row of the table presents the result obtained by particular model for the best hyperparameter configuration, which was discovered using grid search on validation set. In general, all SNA architectures outperform the ones based on *tanh* activation both in terms of average and minimal prediction error. The only exception is minimal NRMSE for MG achieved by *dESN* with *tanh* activation. Nevertheless, the average NRMSE obtained by the same model is from 2.86 to 4 times worse than the results for SNA models. The primacy of SNA architectures is especially visible for MSO dataset, where the best average

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_9

Architecture	MG		MSO		Sunspot	
	Min	Avg	Min	Avg	Min	Avg
ESN tanh	0.01785	0.05586	0.01608	0.03113	0.3615	0.3651
gESN tanh	0.02217	0.06678	0.01525	0.02028	0.3450	0.3669
dESN tanh	**0.00579**	0.05240	0.00517	0.01013	0.3902	0.4024
gdESN tanh	0.02035	0.05864	0.00512	0.01128	0.3760	0.4036
ESN SNA	0.00768	0.01749	0.00026	0.00306	0.3318	0.3353
gESN SNA	0.00779	**0.01310**	0.00055	0.00303	0.3430	0.3610
dESN SNA	0.00847	0.01834	0.00023	**0.00163**	0.3321	**0.3332**
gdESN SNA	0.00700	0.01511	**0.00019**	0.00240	**0.3283**	0.3623
LSTM	0.12166	0.10887	0.03816	0.03990	0.34036	0.34428
Moving average	0.21674	-	0.48106	-	0.35459	-

Table 1. Mackey Glass, MSO and Sunspot one step ahead predicion NRMSE results for all architectures on the test set (5 runs). The architectures with smallest NRMSE in terms of both average and minimal value were bolded. ESN denotes regular shallow architecture with 1000 neurons.

NRMSE *tanh* result is 3.3 times worse than the worst SNA result. The same ratio for MG dataset is 2.86, while for Sunspot it equals 1.01. The selection of hyperparameters for SNA ESNs is cheaper as only the activation radius and regularization factor are adjusted, apart from the layers and groups, which are common for all decoupled models, regardless of activation type. ESNs based on *tanh* are sensitive to hyperparameters, what requires more exhaustive search of parameter space. Comparing the sizes of grid-search spaces we have 960 (*tanh*) vs. 450 (SNA) trials for baseline shallow models, 4800 vs. 2250 for *dESN* and *gESN* models and 9600 vs. 4500 for *gdESN* models. The new *gdESN* model achieved the best minimal NRMSE result for MSO and Sunspot datasets. In case of average NRMSE, it acquired the second best result for MG and MSO. Besides, the Sunspot time-series test was more difficult for *gdESN*, which obtained the worst result from the four decoupled models with SNA. From the perspective of average NRMSE, *dESN* with SNA is the best model, failing only on MG dataset. Overall, the models *gdESN* and *dESN* with SNA are best-suited to predictions on the given benchmark datasets. The hyperparameters selected by grid-search for *gdESN* with SNA are as follows: $N_l = 3$, $N_g = 3$, $r = 50$, $\beta = 0.5$ (MG), $N_l = 3$, $N_g = 2$, $r = 1400$, $\beta = 1.0$ (MSO), $N_l = 3$, $N_g = 2$, $r = 1450$, $\beta = 1.0$ (Sunspot).

5 Conclusion

Combining self normalizing activation function with stacking and grouping of subreservoirs allows for creating robust ESN prediction models, which outperform similar models with *tanh* activation. Grouped Deep SNA architecture proposed in this work can be regarded as a generalization of the previous approaches to reservoir decoupling and forms feasible framework for configuring hyperparameters, which reflect right memory vs. non-linearity balance. The architectures with multiple subreservoirs increase the size of hyperparameter search space per

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_9

se, but at the same time they enrich reservoir dynamics for the model. Besides, application of SNA function reduces the number of hyperparameters, which have to be adjusted and provides high stability of predictions. In general, for SNA, dividing reservoir into subreservoirs decreases memory capacity (see the results published on Github) but at the same time reduces computational complexity of state activations updates. Our future work will include analysis of interplay between reservoir structure (non-random weighted graphs) and SNA activation function. We also plan to extend our analysis of *gdESN* SNA architecture to different benchmark time-series.

Acknowledgements The research presented in this paper was financed from the funds assigned by Polish Ministry of Science and Higher Education to AGH University of Science and Technology.

References

1. Deepesnpy. `https://github.com/lucapedrelli/DeepESN`
2. Bianchi, F.M., De Santis, E., Rizzi, A., Sadeghian, A.: Short-term electric load forecasting using echo state networks and pca decomposition. Ieee Access **3**, 1931–1943 (2015)
3. Bianchi, F.M., Scardapane, S., Løkse, S., Jenssen, R.: Reservoir computing approaches for representation and classification of multivariate time series. IEEE Transactions on Neural Networks and Learning Systems (2020)
4. Chessa, S., Gallicchio, C., Guzman, R., Micheli, A.: Robot localization by echo state networks using rss. In: Recent Advances of Neural Network Models and Applications, pp. 147–154. Springer (2014)
5. Gallicchio, C., Micheli, A.: Reservoir topology in deep echo state networks. In: International Conference on Artificial Neural Networks. pp. 62–75. Springer (2019)
6. Gallicchio, C., Micheli, A., Pedrelli, L.: Deep reservoir computing: A critical experimental analysis. Neurocomputing **268**, 87–99 (2017)
7. Gallicchio, C., Micheli, A., Pedrelli, L.: Design of deep echo state networks. Neural Networks **108**, 33–47 (2018)
8. Jaeger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. science **304**(5667), 78–80 (2004)
9. Kawai, Y., Park, J., Asada, M.: A small-world topology enhances the echo state property and signal propagation in reservoir computing. Neural Networks **112**, 15–23 (2019)
10. Palumbo, F., Gallicchio, C., Pucci, R., Micheli, A.: Human activity recognition using multisensor data fusion based on reservoir computing. Journal of Ambient Intelligence and Smart Environments **8**(2), 87–107 (2016)
11. Pathak, J., Wikner, A., Fussell, R., Chandra, S., Hunt, B.R., Girvan, M., Ott, E.: Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. Chaos: An Interdisciplinary Journal of Nonlinear Science **28**(4), 041101 (2018)
12. Qiao, J., Li, F., Han, H., Li, W.: Growing echo-state network with multiple subreservoirs. IEEE transactions on neural networks and learning systems **28**(2), 391–404 (2016)
13. Verzelli, P., Alippi, C., Livi, L.: Echo state networks with self-normalizing activations on the hyper-sphere. Scientific reports **9**(1), 1–14 (2019)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_9