ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Bidirectional deep-readout echo state networks

Filippo M. Bianchi[1], Simone Scardapane[2], Sigurd Løkse[1] and Robert Jenssen[1] *

1- Machine Learning Group – UiT the Arctic University of Norway

2- Sapienza University of Rome – Dept. of Information Engineering, Electronics and Telecommunications (DIET)

Abstract. We propose a deep architecture for the classification of multivariate time series. By means of a recurrent and untrained reservoir we generate a vectorial representation that embeds temporal relationships in the data. To improve the memorization capability, we implement a bidirectional reservoir, whose last state captures also past dependencies in the input. We apply dimensionality reduction to the final reservoir states to obtain compressed fixed size representations of the time series. These are subsequently fed into a deep feedforward network trained to perform the final classification. We test our architecture on benchmark datasets and on a real-world use-case of blood samples classification. Results show that our method performs better than a standard echo state network and, at the same time, achieves results comparable to a fully-trained recurrent network, but with a faster training.

1 Introduction

Reservoir computing (RC) is an established paradigm for modeling nonlinear temporal sequences [1]. In machine learning tasks, echo state networks (ESNs) are the most common RC models, wherein the input sequence is projected to a high-dimensional space through the use of a (fixed) nonlinear recurrent reservoir [1]. Learning is performed by applying simple linear techniques in the high-dimensional reservoir space. The lack of flexibility in the recurrent part is balanced by a range of advantages, including faster training compared to other recurrent neural networks (RNNs). In tasks requiring a limited amount of temporal memory, ESNs achieve state-of-the-art results in many real-world scenarios constrained by time budgets, low-power hardware and limited data [2]. On the other hand, fully-trained RNNs trade architectural and training complexity with more accurate representations and a larger memory capability [3].

We propose an architecture for time series classification called Bidirectional Deep-readout ESN (BDESN), which combines the training speed of RC with the accuracy of trainable RNNs. We equip our model with a *bidirectional reservoir* to curtail the short-term memory limitation in ESNs. Bidirectional architectures have been successfully applied in RNNs to extract temporal features from the time series that accounts also for dependencies very far in time [4]. Differently from the classic linear readout found in ESNs, we generate the desired output

*This work is partially funded by the Norwegian Research Council FRIPRO grant no. 239844 on the *Next Generation Learning Machines* and IKTPLUSS grant no. 270738 *Deep Learning for Health*.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

by processing the temporal features with a deep feedforward network trained with gradient descent. Although we loose a closed form solution, the whole architecture can still be trained quickly, since the recurrent part is fixed and the time consuming unfolding procedure [5] is avoided. Additionally, we can leverage recent advancements in deep learning to speed up the learning [6]. The temporal features generated by the bidirecitonal reservoir are projected into a subspace before being fed into to feedforward network, to reduce model and training complexity [7]. Deep feedforward networks on top of RNNs demonstrated to enhance the representative power of the state space of a recurrent model [8]. However, so far their application in reservoir computing has been limited, due to difficulties in training [1]. We note that our model differs from the recently proposed deep ESNs [9, 10], which are built by concatenating multiple (fixed) reservoirs, with no modification of the adaptable part. The two ideas are in fact complementary, and can in principle be used together.

We empirically validate our model against standard ESNs and fully-trainable RNNs on multiple benchmark datasets. We show that BDESN vastly outperforms an ESN, achieving a competitive accuracy with respect to RNNs, while being orders of magnitude faster to train.

2 Methodology

The BDESN assigns to a multivariate time series (MTS) $\boldsymbol{x} = \{\mathbf{x}_t\}_{t=0}^{T}$ a class label $\boldsymbol{c}$ through a three-step procedure. First, a special recurrent *reservoir* generates a representation of $\boldsymbol{x}$ that has fixed (large) size and embeds temporal features. Then, a *dimensionality reduction algorithm* projects the reservoir outputs in a lower dimensional space. Finally, a *multilayer perceptron* (MLP) classifies the vectorial representation of $\boldsymbol{x}$. In Fig. 1 we depict the whole architecture, whose details are discussed in the following.

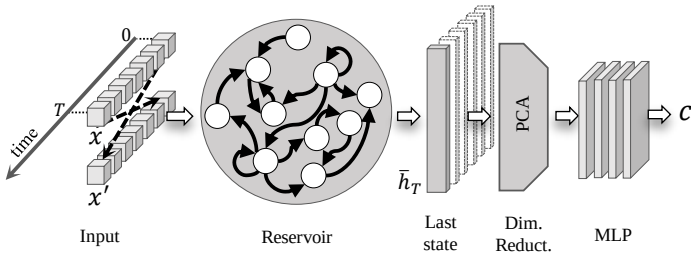

Fig. 1. Schematic illustration of the BDESN architecture.

2.1 Extracting features with a reservoir

The reservoir is governed by the following state-update equation

$$\mathbf{h}_t = f(\mathbf{W}^h \mathbf{h}_{t-1} + \mathbf{W}^i \mathbf{x}_t), \tag{1}$$

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

where $\mathbf{h}_t$ is the internal time-dependent state, which combines the current input $\mathbf{x}_t$ and the previous computations $\mathbf{h}_{t-1}$. The function f is a nonlinear activation (usually a tanh), $\mathbf{W}^h$ is a sparse matrix that defines the recurrent self-connections in the reservoir, and $\mathbf{W}^i$ defines input connections. Both matrices are randomly generated and left untrained, and the reservoir behavior is controlled mainly by three hyperparameters. These are the state size N, the spectral radius ρ of $\mathbf{W}^h$, and scaling of the inputs ω. Through an optimal tuning of these hyperparameters the reservoir produces rich dynamics and its internal states can be used to solve many prediction and classification tasks [1].

The last state $\mathbf{h}_T$ generated by the reservoir, after the whole input $\boldsymbol{x}$ is processed, is a high-level representation of fixed size that embeds the temporal dependencies of $\boldsymbol{x}$. Since the reservoir trades its internal stability with a vanishing memory of the past inputs [11], at time T the state maintains scarce information about the first inputs. To alleviate this issue, we feed to the same reservoir also the MTS in reverse order, $\boldsymbol{x'} = \{\mathbf{x}_{T-t}\}_{t=0}^{T}$ and we generate a new representation $\mathbf{h'}_T$ that is more influenced by the first inputs. A final representation is obtained by concatenating the two states, $\bar{\mathbf{h}}_T = [\mathbf{h}_T; \mathbf{h'}_T]^T$. A bidirectional reservoir has been recently used for time series prediction [12]. However, the architecture proposed in this work is significantly different, mainly because we deal with classification tasks.

2.2 Dimensionality Reduction

A dimensionality reduction method maps data from high to a lower dimensional space, usually relying on unsupervised criteria. In a classification setting supervised techniques can also be adopted to improve the dimensionality reduction procedure, although we do not consider them in this paper.

Popular algorithms are principal component analysis (PCA) and kernel PCA, which project data on the first d eigenvectors of a covariance matrix. PCA and kPCA have been already successfully applied to ESN reservoir states, improving its prediction and modeling capabilities [7]. In BDESN, dimensionality reduction provides a strong regularization that prevents overfitting in the classifier operating on the reservoir states.

2.3 Multilayer Perceptron

In a standard ESN, the output layer is a linear readout that is quickly trained by solving a convex optimization problem. However, we hypothesize that a simple linear model does not possess sufficient representational power for modeling the high-level embeddings resulting from our reservoir space. For this reason, in BDESN the classification is performed by a MLP with L layers on the representations generated at the previous steps. The number of layers in the MLP thus determines a "feedforward" depth to the RNN [8]. Deep MLPs are known for their capability of disentangling factors of variations from high-dimensional spaces, and can nowadays be trained efficiently with the use of sophisticated regularization techniques. In particular, we leverage two procedures, namely

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

dropout and the minimization of the L_2 norm of the model parameters (i.e., weight decay). Thanks to the dimensionality reduction step, we can greatly reduce the number of parameters in the MLP, hence its variance, speed up the computation and obtain a faster convergence during training.

3 Experiments

To evaluate the proposed architecture, we compare the classification accuracy and the training time of BDESN with two different RNNs. The first is a standard ESN, configured with the same reservoir (only unidirectional) of BDESN and with linear readout trained with ridge regression. The second is a network where the vectorial representations are generated by a recurrent layer of gated recurrent units (GRUs), which are fed into a MLP equivalent to the one of BDESN. Contrarily to the reservoir, the GRU layer is trained supervisedly.

Each RNN depends on different hyperparameters, which are optimized by means of random search. The optimized RNNs are trained 10 times, using random and independent model initializations on a Intel Core i7-6850K CPU@4GHz. The code is implemented in Tensorflow and is publicly available online, along with a detailed description of the experimental setup and validation procedure[1].

3.1 Benchmark datasets

We perform classification of MTS from different datasets, whose details are reported in Tab. 1.

Dataset	$\#V$	**Train**	**Test**	$\#C$	T_{min}	T_{max}	**Source**
DistPhal	1	400	139	3	80	80	UCR
ECG	2	100	100	2	39	152	UCR
Libras	2	180	180	15	45	45	UCI
Ch.Traj.	3	300	2558	20	109	205	UCI
Wafer	6	298	896	2	104	198	UCR
Jp.Vow.	12	270	370	9	7	29	UCI

Table 1. Time series dataset details. Column 2 to 5 report the number of variables ($\#V$), samples in training and test set, and number of classes ($\#C$), respectively. T_{min} is the length of the shortest MTS in the dataset and T_{max} the longest MTS.

Classification accuracy on the test sets and training time (in minutes) obtained by each RNN are reported in Tab. 2.

Results show that BDESN consistently performs better than ESN, by obtaining an increment of the accuracy up to 30 percentage points (Character Trajectory dataset). BDESN outperforms GRU, sometimes slightly, in several classification tasks, except in the Wafer dataset. Average GRU accuracy is higher also in ECG, but the difference is not significant due to the high standard deviation in the results. While BDESN is slower than ESN, it can still be trained up to 70 times faster than GRU, providing a huge impact in terms of the required computational resources.

[1] `github.com/FilippoMB/Bidirectional-Deep-reservoir-Echo-State-Network`

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Table 2. Classification accuracy and training time (minutes) obtained by each RNN on different datasets in 10 independent runs. Best average accuracies are in bold.

Dataset	ESN		GRU		BDESN	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
DistPhal	71.5±3.3	0.11	69.4±2.4	34.33	**73.7**±0.91	1.15
ECG	65.6±4.63	0.03	**79.1**±4.3	10.35	78.5±1.5	0.22
Libras	76.2±2.77	0.03	72.7±4.8	5.27	**78.4**±1.9	0.53
Ch.Traj.	43.4±2.72	0.23	72.1±6.4	40.28	**77.6**±5.53	0.63
Wafer	89.7±0.9	0.06	**98.2**±0.9	35.38	90.3±0.8	0.76
Jp. Vow.	85.8±2.34	0.06	94.45±1.88	6.03	**94.72**±1.12	0.58

Fig. 2. Cross entropy loss assumed by GRU and BDESN during training. In BDESN, the loss is more stable and, in average, assumes lower values.

Even if GRU during the training phase may reach a lower cross entropy on the training set, in BDESN the average loss is usually lower and the training is more stable. For example, in Fig. 2 we report the cross entropy values reached during training on the Japanese Vowels dataset. We observe that GRU is more unstable and converges to less accurate values than BDESN.

3.2 Classification of blood samples time series

We test our method also on a real-world dataset of blood measurements collected from patients undergoing a gastrointestinal surgery at University Hospital of North Norway in 2004–2012. Each patient in the dataset is represented by a MTS of 10 blood samples measurements extracted within 20 days after surgery. We consider the problem of classifying patients with and without surgical site infections from their blood samples. The MTS contains missing data, corresponding to measurements not collected for a given patient in one day of the observation period, which are replaced with mean-imputation. The dataset consists of 883 MTS, of which 232 are relative to patients with infections.

RNN	F1 score	Time (mins)
ESN	0.734±0.041	0.01
GRU	**0.779**±0.018	10.54
BDESN	0.761±0.054	1.87

Table 3. F1 score and time of the different models on the blood samples dataset. Best average result is in bold.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

In Tab. 3 we report the F1 score and time obtained by the different models on the blood samples data. Since the dataset is imbalanced, F1 score in this case is a more suitable measure of performance than accuracy. We observe that the performance of GRU and BDESN are comparable and both are better than ESN. GRU mean F1 score is somehow higher and, by looking at the standard deviation, we notice its performance to be slightly more stable. However, GRU trades this moderate improvement with a significant increment in training time, which is 1 and 2 orders of magnitude higher than BDESN and ESN, respectively.

4 Conclusions

We proposed a novel recurrent architecture for time series classification, the bidirectional deep-readout ESN, which captures dependencies in the data forward and backward in time, by means of a large untrained reservoir. The representation yielded by the reservoir is firstly regularized with PCA and then classified by a MLP trained supervisedly. Our approach trains much faster than a RNN configured with GRU cells to generate the same kind of representations. We tested our architecture on several benchmark datasets and one real-world use-case of time series classification. Results indicate that BDESN performance is much better than ESN and sometime even better than a fully trained GRU network.

References

[1] M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 2009.

[2] S. Scardapane and D. Wang. Randomness in neural networks: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2017.

[3] I. Sutskever, O. Vinyals, and Q. V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*. 2014.

[4] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 2005.

[5] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen. *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis*. Springer, 2017.

[6] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[7] S. Løkse, F. M. Bianchi, and R. Jenssen. Training echo state networks with regularization through dimensionality reduction. *Cognitive Computation*, 2017.

[8] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

[9] C. Gallicchio and A. Micheli. Deep reservoir computing: A critical analysis. In *ESANN*, 2016.

[10] C. Gallicchio, A. Micheli, and L. Pedrelli. Deep reservoir computing: a critical experimental analysis. *Neurocomputing*, 2017.

[11] F. M. Bianchi, L. Livi, and C. Alippi. Investigating echo-state networks dynamics by means of recurrence analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[12] A. Rodan, A. F. Sheta, and H. Faris. Bidirectional reservoir networks trained using SVM+ privileged information for manufacturing process modeling. *Soft Computing*, 2017.