Recurrence Enhances the Spatial Encoding of Static Inputs in Reservoir Networks

Christian Emmerich, René Felix Reinhart, and Jochen Jakob Steil

Research Institute for Cognition and Robotics (CoR-Lab),
Bielefeld University, Universitätsstr. 25, 33615 Bielefeld, Germany
{cemmeric,freinhar,jsteil}@cor-lab.uni-bielefeld.de
http://www.cor-lab.de

Abstract We shed light on the key ingredients of reservoir computing and analyze the contribution of the network dynamics to the spatial encoding of inputs. Therefore, we introduce attractor-based reservoir networks for processing of static patterns and compare their performance and encoding capabilities with a related feedforward approach. We show that the network dynamics improve the nonlinear encoding of inputs in the reservoir state which can increase the task-specific performance.

Keywords: reservoir computing, extreme learning machine, static pattern recognition.

1 Introduction

Reservoir computing (RC), a well-established paradigm to train recurrent neural networks, is based on the idea to restrict learning to a perceptron-like read-out layer, while the hidden reservoir network is initialized with random connection strengths and remains fixed. The latter can be understood as a "random, temporal and nonlinear kernel" [1] providing a suitable mixture of both spatial and temporal encoding of the input data in the network's hidden state space. This mixture is based upon three key ingredients illustrated in Fig. 1: (i) the projection into a high dimensional state space, (ii) the nonlinearity of the approach and (iii) the recurrent connections in the reservoir. On the one hand, the advantages of a nonlinear projection into a high dimensional space are beyond controversy: so-called kernel expansions rely on the concept of a nonlinear transformation of the original data into a high dimensional feature space and the subsequent use of a simple, mostly linear, model. On the other hand, the recurrent connections implement a short-term memory by means of transient network states. Due to this

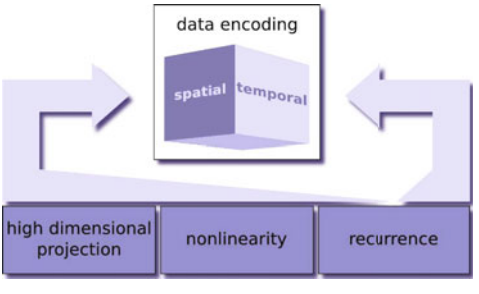

Fig. 1. Key ingredients of RC

K. Diamantaras, W. Duch, L.S. Iliadis (Eds.): ICANN 2010, Part II, LNCS 6353, pp. 148–153, 2010.

short-term memory, reservoir networks are typically utilized for temporal pattern processing such as time-series prediction, classification and generation [2]. In principle, short term memory can also be implemented in a simpler fashion, e.g. by an explicit delay-line. But we point out that the *combination of spatial and temporal encoding* makes the reservoir approach powerful and can explain the impressive performance on various tasks [3, 4]. It remains nevertheless unclear how the network dynamics influence the spatial encoding of inputs.

Our hypothesis is that the *dynamics* of the reservoir network enhances the spatial encoding of *static* inputs by means of a more nonlinear representation, which consequently improves the task-specific performance. Moreover, we expect an improved performance when applying larger reservoirs, i.e. when using an increased dimensionality of the nonlinear feature expansion. We systematically test the contribution of the network dynamics to the spatial encoding independently from its temporal effects by using attractor-based computation and by considering purely static input patterns, A statistical analysis of the distribution of the network's attractor states allows to access the qualitative difference of the encoding caused by the network's recurrence indepently of the task-specific performance.

2 Attractor-Based Computation with Reservoir Networks

We consider the three-layered network architecture depicted in Fig. 2, which comprises a recurrent hidden layer (reservoir) with a large set of nonlinear neurons. The input, reservoir and output neurons are denoted by $\mathbf{x} \in \mathbb{R}^D$, $\mathbf{h} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^C$, respectively. The reservoir state is governed by discrete dynamics

$$\mathbf{h}(t+1) = \mathbf{f}\left(\mathbf{W}^{inp}\,\mathbf{x}(t) + \mathbf{W}^{res}\,\mathbf{h}(t)\right), \tag{1}$$

where the activation functions f_i are applied componentwise. The reservoir neurons have sigmoidal activation functions such as $f_i(x) = tanh(x)$, whereas the output layer consists of linear neurons, i.e. $\mathbf{y}(t) = \mathbf{W}^{out}\,\mathbf{h}(t)$.

Learning in reservoir networks is restricted to the read-out weights $\mathbf{W}^{out}$. All other weights are randomly initialized and remain fixed. In order to infer a desired input-to-output mapping from a set of training examples $(\mathbf{x}_k^T, \mathbf{y}_k^T)_{k=1,\ldots,K}$, the read-out weights $\mathbf{W}^{out}$ are adapted such that the mean square error is minimized. In this paper, we use ridge regression: For all inputs $\mathbf{x}_1, \ldots, \mathbf{x}_K$ we collect the corresponding reservoir states $\mathbf{h}_k$ as well as the desired output targets $\mathbf{y}_k$ column-wise in a reservoir state matrix $\mathbf{H} \in \mathbb{R}^{N\times K}$ and a target matrix $\mathbf{Y} \in \mathbb{R}^{C\times K}$, respectively. The optimal read-out weights are determined by the least squares solution with a regularization factor $\alpha \geq 0$: $\mathbf{W}^{out} = \mathbf{Y}\mathbf{H}^T\left(\mathbf{H}\mathbf{H}^T + \alpha\mathbf{1}\right)^{-1}$.

The described network architecture in combination with the offline training by regression is often referred to as echo state network (ESN) [2]. The potential of the ESN approach depends on the quality of the input encoding in the reservoir. To adress that issue, Jaeger proposed to use weights drawn from a uniform distribution in $[-a, a]$, where often a sparsely connected reservoir is preferred.

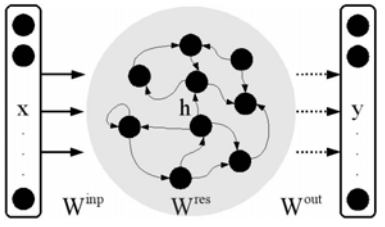

Fig. 2. Reservoir network

Algorithm 1. Convergence algorithm

Require: get external input $\mathbf{x}_k$
1: **while** $\Delta\mathbf{h} > \delta$ and $t < t_{max}$ **do**
2: apply external input $\mathbf{x}(t) = \mathbf{x}_k$
3: execute network iteration (1)
4: compute state change $\Delta\mathbf{h} = ||\mathbf{h}(t) - \mathbf{h}(t-1)||^2$
5: $t = t+1$
6: **end while**

In addtion, the reservoir weight matrix $\mathbf{W}^{res}$ is scaled to have a certain spectral radius $\lambda_{\max}$. There are two basic parameters involved in this procedure: the reservoir's weight connectivity or density $0 \leq \rho \leq 1$ and the spectral radius $\lambda_{\max}$, which is the largest absolute eigenvalue of $\mathbf{W}^{res}$.

In this paper, an attractor-based variant of the echo state approach is used, i.e. we map the inputs $\mathbf{x}_k$ to the reservoir's related attractor states $\bar{\mathbf{h}}_k$: The input neurons are clamped to the input pattern $\mathbf{x}_k$ until the network state change $\Delta\mathbf{h} = ||\mathbf{h}(t+1) - \mathbf{h}(t)||^2$ approaches zero. This procedure is condensed in Alg. 1. As a prerequisite it must hold that the network always converges to a fix point attractor, which is related to a scaling of the reservoir's weights such that $\lambda_{\max} < 1$. The resulting attractor states $\bar{\mathbf{H}}$ are used for training.

Note that an ESN with a spectral radius $\lambda_{\max} = 0$ or with zero reservoir connectivity ($\rho = 0$) has no recurrent connections at all. Then, the ESN degenerates to a feedforward network with randomly initialized weights. In [5], this special case of RC has been called extreme learning machine (ELM). As our intention is to investigate the role of the recurrent reservoir connections, this feedforward approach obviously is the non-recurrent baseline of our recurrent model and we present all results in comparison to this non-dynamic model.

3 Key Ingredients of Reservoir Computing

We investigate the influence of the key ingredients of RC on the network performance for several data sets (Tab. 1) in a static pattern recognition scenario. Except for Wine, all data sets are not linearly seperable and constitute nontrivial classification tasks. The introduced models are used for classification of each data set. We represent class labels c as a 1-of-C coding in the target vector $\mathbf{y}$ such that $y_c = 1$ and $y_i = -1\ \forall i \neq c$. For classification of a specific input pattern, we apply Alg. 1 and then determine the estimated class label $\hat{c}$ from the network output $\mathbf{y}$ according to $\hat{c} = \arg\max_i y_i$. All results are obtained by either partitioning the data into several cross-validation sets or using an existing partition of the data into training and test set and are averaged over 100 different network initializations. We use normalized data in the range $[-1, 1]$.

Role of Reservoir Size and Nonlinearity

Fig. 3 shows the impact of the reservoir size to the network's recognition rate for a fully connected reservoir, i.e. $\rho = 1.0$, with $\lambda_{\max} = 0.9$ and α as in Tab. 1. The number of correctly classified samples increases strongly with the number of hidden neurons. On the one hand, this result shows that the projection of the input into a high-dimensional network state space is crucial for the reservoir approach: The performance of very small reservoir networks degrates to the performance of a linear model (LM). We observe also a saturation of the performance for large reservoir sizes. It seems that the random projection can not improve the separability of inputs in the network state space anymore. On the other hand, note that the nonlinear activation functions of the reservoir neurons are crucial as well: Consider an ELM with linear activation functions, then the inputs are only transformed linearly in a high dimensional representation. Hence, the read-out layer can only read from a linear transformation of the input and the classification performance is thus not affected by the dimensionality of that representation. Consequently, the combination of a random expansion and the non-linear activation functions is essential.

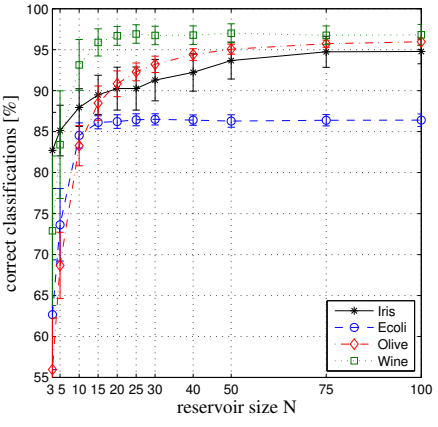

Fig. 3. Classification performance depending on the reservoir size N

Role of Reservoir Dynamics

In this section, we focus on the role of reservoir dynamics and restrict our studies on the Iris data set. We vary both the spectral radius $\lambda_{\max}$ of the reservoir matrix $\mathbf{W}^{res}$ and the density ρ for a fixed reservoir size of $N = 50$. Note again that we obtain an ELM for $\lambda_{\max} = 0$ or $\rho = 0$. Fig. 4 reveals that for recurrent networks the recognition rate increases significantly with the spectral radius $\lambda_{\max}$ and surpasses the performance of the non-recurrent networks with the same parameter configuration. Interestingly enough, this is not true for the weight density in the reservoir: adding more than 10% connections inbetween the hidden neurons has only marginal impact on the classification performance, i.e. two many connections neither improve nor detoriate the performance. Note that there is a trade-off in an increased

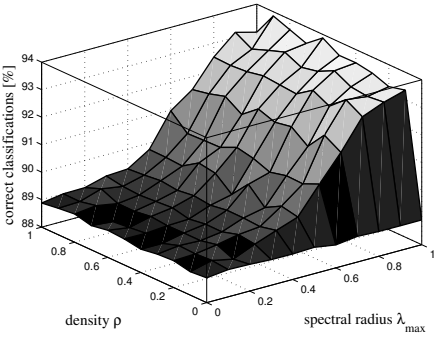

Fig. 4. Recognition rate for the Iris data set depending on $\lambda_{\max}$ and ρ

number of iterations the network needs for settling in a stable state, which correlates with the spectral radius $\lambda_{\max}$.

4 On the Distribution of Attractor States

We give a possible explanation for the improved performance caused by the recurrent connections. Our hypothesis is that these connections spread out the network's attractors to a spatially broader distribution than a non-recurrent approach is capable of, which results in an increasingly nonlinear hidden representation of the network's inputs. Because we perform linear readout, it is reasonable to analyse the encoding $\bar{\mathbf{H}}$ by linear methods, namely PCA. Given the dimension D of the data, we expect the hidden representation to encode the input information with a significantly higher number of relevant principle components (PCs). Therefore, we calculate the shift of information or energy content from the first D PCs to the remaining $N-D$ PCs. Let $\lambda_1 \geq \ldots \geq \lambda_N \geq 0$ be the eigenvalues of the covariance matrix $Cov(\bar{\mathbf{H}})$. We calculate the normalized cumulative energy content of the first D PCs by $g(D) = (\sum_{i=1}^{D} \lambda_i)/(\sum_{i=1}^{N} \lambda_i)$, which measures the relevance of the first D PCs. The case of $g(D) < 1$ implicates a shift of the input information to additional PCs, because the encoded data then spans a space with more than D latent dimensions. If $g(D) = 1$, no information content shift occurs, which is true for any linear transformation of data.

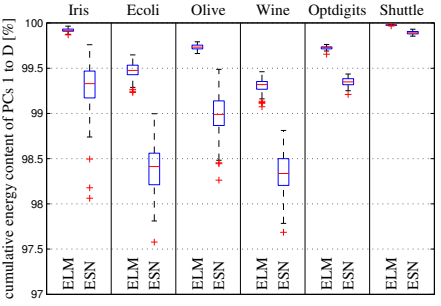

Fig. 5. Normalized cumulative energy content $g(D)$ of the first D PCs

Fig. 5 reveals that both approaches are able to encode the input data with more than D latent dimensions. In the case of an ELM, the information content shift is solely caused by its nonlinear activation functions. For recurrent networks, we observe the forecasted effect: The cumulative energy content $g(D)$ of the first D PCs of the attractor distribution is significantly lower for reservoir networks than for ELMs. That is, a reservoir network redistributes more of the existing information in the input data onto the remaining $N-D$ PCs than the feedword approach. This effect, cuased by the recurrent connections, shows the enhanced spatial encoding of inputs in reservoir networks and can explain the improved performance (cf. Fig. 4).

We remark that the introduced measure $g(D)$ does not strictly correlate with the task-specific performance. Although the ESN reassigns a greater amount of information content on the last $N-D$ PCs than the ELM (cf. Fig. 5), this does not improve the generalization performance for every data set (cf. Tab. 1).

Table 1. Mean classification rates with standard deviations

	data set properties			**classification rate** [%] (*L*-**fold cross-validation**)				**network properties**		
	D	C	K	L	**LM**	**ELM**	**ESN**	N	a	α
Iris [6]	4	3	150	10	83.3	88.9 ± 0.7	92.7 ± 2.1	50	0.5	0.001
Ecoli [6]	7	8	336	8	84.2	86.6 ± 0.5	86.4 ± 0.6	50	0.5	0.001
Olive [7]	8	9	572	11	82.7	95.3 ± 0.5	95.0 ± 0.7	50	0.5	0.001
Wine [6]	13	3	178	2	97.7	97.6 ± 0.7	96.9 ± 1.0	50	0.5	0.1
Optdigits [6]	64	10	5620	-	92.0	95.9 ± 0.4	95.8 ± 0.4	200	0.1	0.001
Statlog Shuttle [6]	9	7	58000	-	89.1	98.1 ± 0.2	99.2 ± 0.2	100	0.5	0.001

5 Conclusion

We present an attractor-based implementation of the reservoir network approach for processing of static patterns. In order to investigate the effect of recurrence on the spatial input encoding, we systematically vary the respective network parameters and compare the recurrent reservoir approach to a related feedforward network. The reservoir dynamics result in an increased nonlinear representation of the input patterns in the network's attractor states which can be advantageous for the separability of patterns in terms of static pattern recognition. In temporal tasks that also require a suitable spatial encoding, the mixed spatio-temporal representation of inputs is crucial for the functioning of the reservoir approach. Incorporating the results reported in [3, 4], we conclude that the spatial representation is not detoriated by the temporal component.

References

[1] Verstraeten, D., Schrauwen, B.: On the Quantification of Dynamics in Reservoir Computing. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009. LNCS, vol. 5768, pp. 985–994. Springer, Heidelberg (2009)

[2] Jaeger, H.: The echo state approach to analysing and training recurrent neural networks. Technical Report 148, German National Research Center for Information Technology (2001)

[3] Verstraeten, D., Schrauwen, B., Stroobandt, D.: Reservoir-based techniques for speech recognition. In: Proc. IEEE IJCNN, pp. 1050–1053 (2006)

[4] Ozturk, M., Principe, J.: An associative memory readout for ESNs with applications to dynamical pattern recognition. Neural Networks 20(3), 377–390 (2007)

[5] Huang, G., Zhu, Q., Siew, C.: Extreme learning machine: Theory and applications. Neurocomputing 70(1-3), 489–501 (2006)

[6] Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`

[7] Forina, M., Armanino, C.: Eigenvector projection and simplified nonlinear mapping of fatty acid content of italian olive oils. Ann. Chem. (72), 125–127 (1982)