Power and Limits of Recurrent Neural Networks for Symbolic Sequences Processing

Matej Makula*
Institute of Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
makula@fiit.stuba.sk

Abstract

A recurrent neural network is a class of neural network where connections between neurons form a directed cycle. These so-called recurrent connections allow spreading information about past neural activities in network, which enables to process temporal inputs. Although they are theoretically equivalent to Turing machines, widespread use is restricted due to computational expensive training and lack of knowledge of internal representation mechanism in this class of networks.

Our thesis studies properties of recurrent neural networks while processing symbolic inputs. We focused mainly on their relation and description of their behavior in terms of dynamical systems. We describe the dynamics of randomly initialized neural network and its relation to Markov prediction models of variable length. In the main part of our work, we present usability of methods for visualization, clusterization and the state space analysis as an effective tool for thorough study of recurrent networks capabilities on prediction tasks.

In experimental part of our thesis, we focus on studying changes that emerge in training. We are mostly interested in the change of naïve Markovian dynamics of randomly initialized network during training in relation to various factors such as input sequence, training algorithm, network architecture, number of hidden units, etc.

We focused not only on simple recurrent network before and after training, but also on the computational capabilities of the new approach called echo state networks. It uses large randomly initialized neural reservoir, which dynamics is the subject of our interest. We demonstrate benefits and constrains of this currently popular approach based on the results of our experiments and differences identified after recurrent networks training.

Categories and Subject Descriptors

I.2.6 [**Artificial intelligence**]: Learning; I.2.6 [**Artificial intelligence**]: Learning—*connectionism, neural networks*

Keywords

recurrent neural networks, symbolic sequences, dynamical systems, analysis of dynamics, state space visualization

1. Introduction

A recurrent neural network is a class of neural network where connections between neurons form a directed cycle. These so-called recurrent connections allow spreading information about past neural activities in network, which enables to process temporal inputs.

Various methods can be used for recurrent neural network adaptation. Straightforward solution is to use basic error back-propagation method as was used in [3]. However, this simple method does not have to provide adequate solution especially in case of difficult problems. To calculate exact gradient one has to take information from previous time steps into account. Two basic algorithms can manage this. The first one, which unfolds network in time, is back-propagation through time (BPTT). Each weight has several copies that represent time the weight was used in forward calculation [27]. The second algorithm is real time recurrent learning (RTRL). In this case gradient is calculated sequentially in each time step - in real time [28].

Nowadays so-called Kalman filtering methods (KF) become more popular. Classical Kalman filter is effective recursive filter that estimates the state of linear system from a series of noisy observations [9]. Application of KF to nonlinear problems, such as weights adaptation of recurrent neural network, can be done by linearization using Taylor series [26]. This approach is called extended Kalman filter (EKF).

Despite the highest robustness of mentioned algorithms adaptation of recurrent networks is still difficult. One most known problem is so-called vanishing gradient problem, which causes network training to fail on problems where temporal dependencies are spanning many time steps. The solution to this problem is to use specific architecture

*Recommended by thesis supervisor: Assoc. Prof. Ľubica Beňušková
Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on July 8, 2009.

with long short-term memory (LSTM), which uses specific memory cells with single recurrent connection and input/output gates to store information for reasonable time [6].

Since the year 2000 new approaches, represented by echo state networks (ESN) and liquid state machines (LSM), appeared in the field of recurrent neural network [7, 13]. ESN and LSM are neural networks with large untrained random "reservoir" of simple computational elements that are used to preprocess input signal. The weights of output neurons can be trained to read out extracted features.

Although recurrent neural networks are theoretically equivalent to Turing machines, widespread use is restricted due to computational expensive training and lack of knowledge of internal representation mechanism in this class of network.

2. Recurrent Neural Networks

Historical development in artificial neural networks resulted in various directions in research activities. The title of our work *"Power and limits of recurrent neural networks for symbolic sequences processing"* indicates that the basic objective is to bring new insight on the principles of their behavior. We presume that recurrent network is not a black box, which uses adjustable weights to solve particular problems. Instead, we analyze their internal mechanisms and study the way they solve particular problems. Our work is aimed on description of their behavior in the terms of dynamical systems. We assume, that the most important regimes of behavior can be described by studying neural network state space [2, 18]. Activities of network units represent internal network state, encode temporal information and designate the way network transforms input to desired output. In the case of symbolic sequences, limited set of input symbols allows effective analysis of network behavior. More precisely, each symbolic input is related to the network state transformation. This transformation can be studied as dynamical system, which basic characteristics are determined by so-called fixed points [24].

The first part of our analysis was an analysis of the dynamics of untrained randomly initialized recurrent network. It has been known for some time that untrained neural network provides similar results as Markovian prediction models [20]. This phenomenon called Markovian architectural bias of recurrent network originates from initialization of networks weights to small values. Small values cause recurrent network activities (i.e. network state) to organize in the state space the same way as prediction contexts in variable length Markovian models (VLMM) [21, 22].

The basic motivation of our work is to understand how neural network extracts knowledge about temporal dependencies from input sequence to its internal representation. More precisely, we are interested in the transformation of naïve Markovian dynamics in relation to various factors such as input sequence, training algorithm, network architecture, number of hidden units, etc. For this purpose, we use three basic methods of analysis:

Visualisation of network state clusters, and state trajectories in different locations of network state space.

Clusterization, which allows us to analyze discrete version of network state space.

Fixed points analysis, which studies network state transformations while processing individual inputs.

Previously published works used mentioned methods to analyze dynamics of small networks with two or three hidden units [19, 1, 23]. Our work differs from these research activities in the fact that we focused on the networks with more hidden units. Processing of complex sequences is usually possible only with large networks. This allows us to focus on problems which have not been studied thoroughly yet.

3. Recurrent Networks as Dynamical Systems

Recurrent neural networks can be characterized as discrete time dynamical system that have input, output and state variables [2, 18, 24, 19]. The most important part of network in our studies is hidden layer. Activities of units in the hidden layer can be thought of as an internal network state. For example the state **s** of recurrent network is vector of real values composed of hidden unit activities. It is a point in multidimensional space, in which number of dimensions is equal to the number of hidden units. Basic property of state space model is that actual state position has to provide enough information to predict dynamical system evolution [5].

The impact of network weights initialization to network dynamics can be visualized by bifurcation diagram (Figure 1). Axis y of bifurcation diagram represents average value of activities of state neurons, i.e. values $s_{avg}(t) = \frac{1}{N}\sum_{i=0}^{N} s_i(t)$ at time steps $t = 100$ to $t = 150$. Randomly initialized weights were than scaled by coefficient $\alpha \in (0, 20)$.

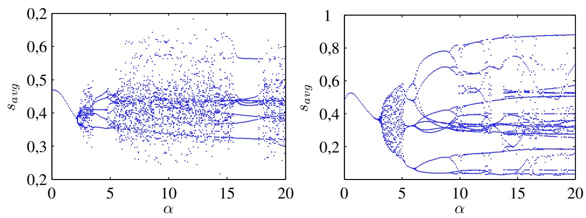

Figure 1: Bifurcation diagrams of two randomly initialized recurrent networks. Symbol α stands for the scale ratio of recurrent weights.

If the network weights are initialized to small values, i.e. $\alpha < 1$, state dynamics is influenced by the stable fixed point – attractor, which contracts states. If α is increased enough, bifurcation occurs and dynamics is interleaving between limit cycles and chaotic behaviors, respectively.

Markovian dynamics of randomly initialized RNN is quite simple and can be explained by the iterated function theory [10, 11]. Processing of constant input vector results in stabilization of network state in fixed point – attractor. Since each input symbol is encoded with different input vector, network state is stabilized in different region for each input symbol. Processing of symbolic sequence thus creates fractal structure (Figure 2) [11, 10].

Organization of clusters in network state before training corresponds to structure of prediction contexts of variable

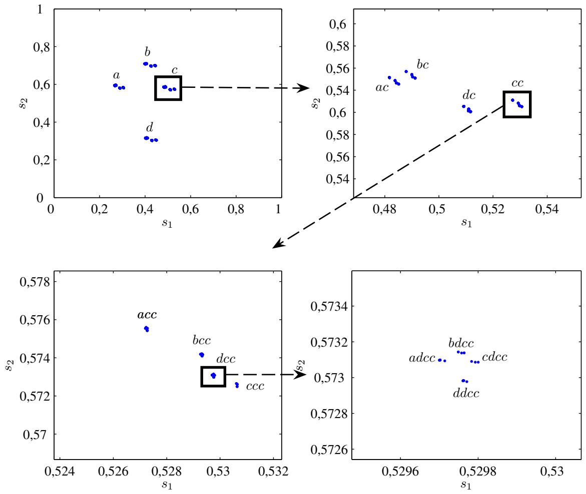

Figure 2: State space of naïve network while processing random sequence of four symbols a, b, c and d. Network state encodes information about symbol presented at the time t, four cluster top level. Information about previous input symbols is also encoded: symbol at the time $t-1$ (nested clusters at 1st level), symbol at time $t-2$ (nested clusters at 2nd level), etc.

length markov model, which is called Markovian architectural bias of recurrent neural networks [21, 22]. During training process, network weights usually expand from stable interval. Consequent bifurcations result in emergence of complex behavior. Analysis of dynamic emerged in training process is the main objective of our work.

4. Methods for Network Dynamics Analysis

Fixed point is a network state $\tilde{\mathbf{s}}$ which does not change in time.

$$\tilde{\mathbf{s}} = \mathbf{F}(\tilde{\mathbf{s}}) \tag{1}$$

Fixed points in the network state space have essential impact on the state trajectory of trained network (i.e. its dynamics). Every fixed point can be characterized as an attractor, repelor or saddle point (Figure 3). When network state space contracts to a fixed point, that point is an attractor (Figure 3a), otherwise that point is repelling (Figure 3b). In some cases, the repelling point may be contracting in one direction and expanding in another direction, so we call it a saddle point (Figure 3c).

Although, recurrent neural networks are not linear (nonlinearity is introduced by activation function), linearization can be performed to study character of fixed points. The eigenvalues λ_i and eigenvectors $\mathbf{v}_i$ of Jacobian $\mathbf{J}(\tilde{\mathbf{s}})$ (partial derivative matrix) calculated at fixed point $\tilde{\mathbf{s}}$ express how system changes in time [12]. They have to satisfy following condition $\mathbf{J}(\tilde{\mathbf{s}}) \cdot \mathbf{v}_i = \lambda_i \cdot \mathbf{v}_i$ and $\mathbf{v}_i \neq 0$, i.e. they are expressing direction and intensity of linear contraction/expansion of linearized system. If the complex eigenvalue λ_i lies in the unit circle, fixed point is contracting in the direction $\mathbf{v}_i$, otherwise is repelling. Moreover, the non-zero imaginary part of eigenvalue is a sign of rotation around the fixed point. The negative value of λ_i indicates that the state is driven to/from fixed point by a 2–periodic oscillations (Figure 3c, left).

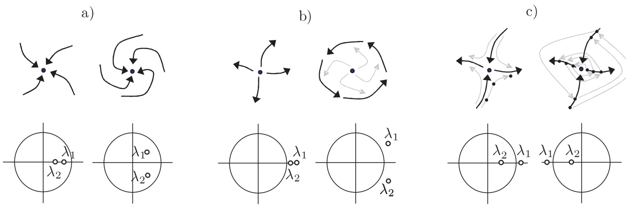

Figure 3: State space topology near the fixed point $\tilde{\mathbf{s}}$ and its relation to the eigenvalues λ_i of partial derivative matrix $\mathbf{J}(\tilde{\mathbf{s}})$. a) attractor – state trajectories lead state to fixed point; b) repelling point; c) saddle – repelling in one direction and contracting in other.

Analysis of network dynamics before, during and after training is not and simple task. To get and valuable insight one has to examine many parameters. In our work we use three basic methods of analysis.

Visualization of state space. Visualization is used to draw network state space, i.e. clusters of states and corresponding trajectories representing internal representation of network. State space of larger network is visualized with linear projection to 2D/3D subspace which is created by using two or three principal components of PCA.

Clusterization and neural prediction model analysis. Discretization of network state space creates finite set of clusters. Building of prediction models over this discrete version of state space helps us to get overview on changes of network dynamics. Similar process can also be used to compare different training algorithms.

Fixed point analysis. Localization and analysis of fixed points in the network dynamics identify changes in training process. To identify both unstable and stable fixed points Newton-Raphson method was used. Linearization and consequent visualization of state trajectories revealed their impact to state trajectories.

5. Adaptation of Recurrent Neural Networks

We performed several experiments related to previously published results. The first set of experiments were focused on analysis of networks trained on next symbol prediction tasks of three linguistic sequences: $a^n b^n$, *Christiansen-Chater* and *Elman language*. We also used neural prediction machines to compare two different approaches of network training: self-organizing BCM and simple recurrent network trained by EKF.

There are three known distinctive solutions for networks with two and three hidden units trained on next symbol prediction of language $a^n b^n$: monotonic, oscillating and exotic [25, 1]. The first solution contains two attracting points in opposite corners of the state space (Figure 4a). While processing a and b inputs, the network state is moving between these two attractors, which can be interpreted as symbol counting. Oscillating dynamics, which is known to achieve some kind of generalization, is performed by attractor and one saddle point (Figure 4b). In this case, counting is performed by oscillating towards the attractive point (a-system) and from the saddle point (b-system).

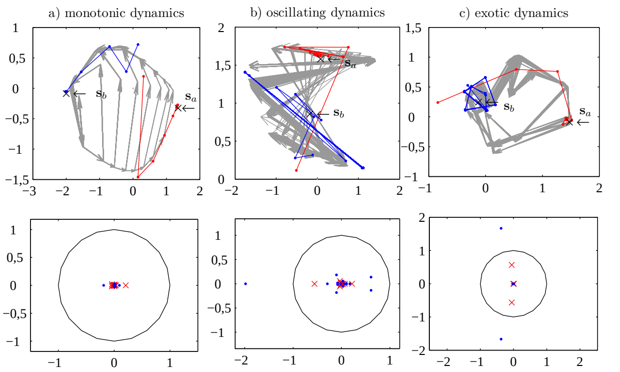

Figure 4: Three typical solutions of counter in the SRN state space: a) monotonic counter – two attractive points; b) oscillating counter – attractor s_a and oscillating saddle s_b; c) exotic counter – attractor s_a and unstable fixed point s_b. First and second network have 15 hidden units. Third network has 5 hidden units. The eigenvalues of Jacobian matrix calculated at s_a and s_b points are shown as red crosses and blue dots, respectively.

Introduced methods of analysis allow us to study and identify characteristic solutions in networks with high number of hidden units [15]. Our results showed bias of BPTT training towards simpler – monotonic solution. On contrary, dynamics composed of both stable and unstable fixed point dominated in the case of EKF training.

Similar method was used to analyze networks trained on next symbol prediction of languages *Christiansen-Chater* and *Elman*. In case of *Christiansen-Chater* sequence, we identified that the training changes position of attractors and increases distances between them. This allows network to spread clusters in the network state space in optimal way. Moreover, we found out, that symbols used in identical temporal contexts (e.g. words of the same grammatical category) rearrange their attractive points in the same part of the state space.

Complex chaotic characteristics of *Laser* sequence and deeper recursion in *Elman language* requires network to store longer temporal information. This results in the change of attractor stability and/or creation of new fixed points. These new points allow network to drive its state in different parts of state space in different manner and thus trained networks outperform Markovian prediction models.

6. Echo State Networks

In the second part of our work, we focused on the echo state neural networks. They use large untrained randomly initialized dynamical reservoir to preprocess input sequence (Figure 5). Output layer of ESN is used to transform this preprocessed input from dynamical reservoir to desired output sequence. An essential condition for the ESN approach is that dynamical reservoir must produce meaningful response, i.e. network state must be an "echo" of input signal. This is achieved by rescaling network weights to small values, which is apparently similar to architectural bias condition, i.e. initialization with small weights.

We used methods for dynamical system analysis to identify relation between Markovian architectural bias and

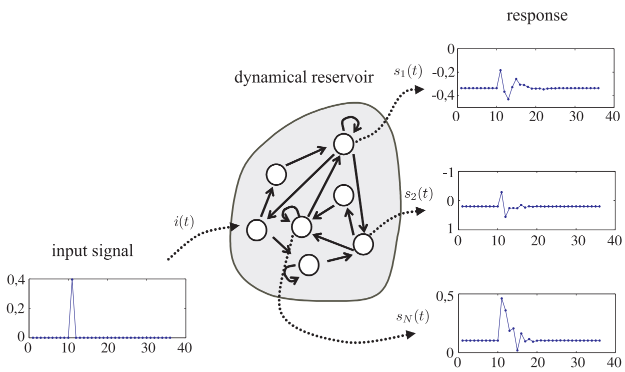

Figure 5: Dynamical reservoir preprocesses input sequence and creates potentially interesting response. Sparsely connected units in hidden layer thus produce richer version of input sequence [7].

crucial echo state property of ESN. Especially random initialization of dynamical reservoir indicates that echo state networks use Markovian architectural bias of dynamical reservoir to produce desired output [17].

It was shown, that existence of echo state in dynamical reservoir can be achieved by setting weights in dynamical reservoir to small values [7]. This procedure set absolute value of spectral radius of matrix **M** to value less than 1. Matrix **M** is composed of random values of reservoir neurons weights. Value of spectral radius has essential impact on existence of echo state and consequently influences global memory characteristics of ESNs [8, 16].

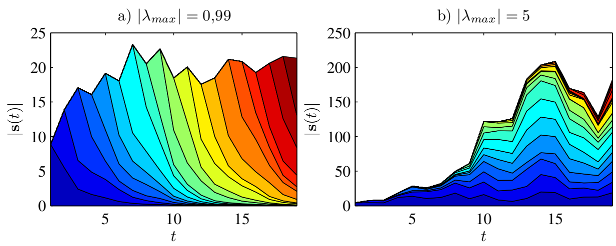

Figure 6: Visualization of the state vector components. Each colour represents component of state vector corresponding to information from specific time step. Dark (blue) colour represents component of state vector encoding input from time step $t = 1$, light (red) corresponds to information from time step $t = 19$. Forgetting factor depends on the value of spectral radius $|\lambda_{max}|$ of the matrix M.

The impact of actual input to components of state vector can be easily analyzed. We set input at analyzed time step to zero and calculate distance between old and new state trajectory during several time steps. Impact of inputs was closely related to the value of spectral radius $|\lambda_{max}|$. For values less than 1 exponential forgetting curve appeared (Figure 6a). It means, that inputs from the beginning of sequence have only small impact on network state at the time step $t = 20$. Forgetting factor was related to the value of spectral radius of matrix **M**. The slowest descent appears if value of $|\lambda_{max}|$ is close to 1 and steepest descent for values close to 0. If matrix is scaled to value greater than 1, echo states property is not satisfied (Figure 6b).

7. Computational Capabilities of Dynamical Reservoirs

When processing symbolic inputs it is convenient to initialize input weights in the network according to co-occurrence statistics of input symbols [4]. Therefore, symbols used in similar temporal context have similar impact on activities of neurons in dynamical reservoir. This results in better generalization for symbols that did not appear in training set. Network using foregoing approach for input weight initialization is called ESN+.

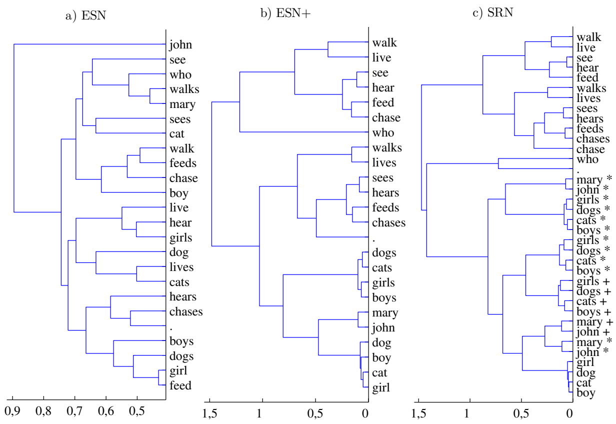

Figure 7: Distances of fixed points in the state space of: a) ESN; b) ESN+; c) SRN. In the case of SRN dynamics of some input symbols is composed of cobination of several fixed points. Attractors are marked with symbol * and saddles with symbol +. For examples dynamical systems of words *john* a *mary* are composed of two attractors and one saddle point.

Our proposal was to enhance described initialization with random linear transformation of input weights. Analysis of state space organization shows that this simple enhancement allows us to adjust distances between attractors of input symbols according to specific task (Figure 7). Results of our experiments indicate that ESN+ with linear transformation provides better performance compared ESN+ or classical ESN. On the other hand analysis of classical simple recurrent network trained by algorithm EKF shows that adaptation creates in the state space configuration of both stable and unstable fixed points, which allows network to outperform ESN and ESN+ approaches (Figure 7c).

Because results indicate that performance of ESN approach is closely related to initialization of reservoir before training, we decided to simplify structure of reservoir connections. It is based on removing recurrent connections from reservoir. Neurons are labeled with numbers, and recurrent connections exist only between neurons with lower index. More precisely network is unable to use recurrent connections to store temporal information for unlimited time. Instead, the number of hidden units in reservoir limits its memory.

Results of our experiments showed similar performance of ESN and feed forward ESN on three different prediction tasks. Potential of FF-ESN is hidden in scaling, i.e. adding and removing neuron with maximal index. In classical ESN, removing of single neuron has significant impact on reservoir dynamic. In FF-ESN, neuron with maximal index does not affect network dynamics, i.e. dynamical reservoir can be easily resized.

First technical report about ESN described simple enhancement of training procedure based on state wobbling [7]. It adds small noise to input sequence, which helps network to increase its prediction capabilities. Moreover, if predicted output is used as input for next time step, network stably reproduces desired periodical sequence. Analysis of state space of trained ESN while generating periodical sequence helps us to identify relation between stability of solution and output error and noise. We also suggest enhancement that uses online adaptation to find stable solution even without state wobbling. Online training process allows us to use actual output (with prediction error) as input signal and thus network can find stable solution.

8. Conclusion

Our experiments concentrated on the changes in the network dynamics while processing symbolic sequences. We can conclude that fixed-point Markovian dynamics changes during training on temporal sequences. If network is trained on simple task, attractive points rearrange their positions to distinct parts of state space. This change spreads state clusters in the state space in an optimal way.

Moreover, if different symbols in input sequence are used in similar temporal context, corresponding fixed points are located in the same space after training. The change of fixed-point stability was related to deeper recursion in linguistic time series or longer temporal dependence in chaotic sequences, respectively. Consequently, new fixed points are created, and network can drive state in distant parts of state space in a different way.

In the next part of our work, we focused on new approach in the field of recurrent neural networks – echo state neural networks. They are based on idea of using large reservoir of randomly interconnected units. We performed several experiments that studied relationship between Markovian architectural bias and echo state networks. Random initialization of dynamical reservoir and consequent rescaling of reservoir weights to small values creates contractive dynamics, which is based on single attractor. The absence of adaptation of reservoir dynamic has major impact on the computational capabilities of echo state networks.

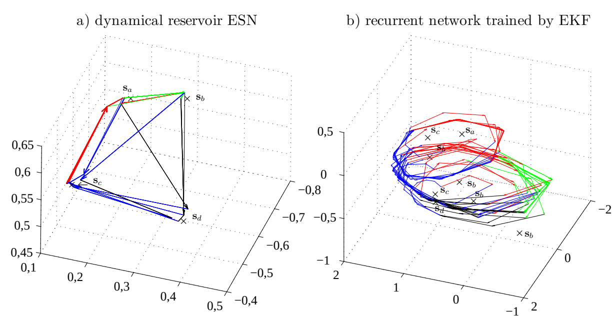

Figure 8: Network state trajectory while processing chaotic sequence *Laser*: a) randomly initialized reservoir ESN; b) classical network trained by EKF. The state change is colored according to presented symbol: a – red; b – green; c – blue; d – black.

To demonstrate stated fact we visualize state trajectory of ESN and classical recurrent network trained by EKF algorithm on next symbol prediction task *Laser*. Markovian dynamics of randomly initialized reservoir of ESN includes four attractive points corresponding to four input symbols (Figure 8a). This organization of state space is present after each random initialization regardless of number of hidden units in reservoir [14]. Position of ESN state in the state space clusters thus always changes according to Markovian fractal structure (Figure 2).

Dynamics of network trained by EKF algorithm is different. Dynamics changes and stability of fixed points is changed and new fixed points appear. Some input symbols thus employ several fixed points that drive state in distant parts of the space in a different way (e.g. symbols b, c on Figure 8b).

This allows network to move its state on chaotic orbit, which consequently allows network to store information about distant input symbols and thus overcome well-known information latching problem. ESN with large dynamical reservoir thus never achieve performance that can be achieved by classically adopted small recurrent network.

Acknowledgements. This work was supported by the VG–1/0848/08 grant.

References

[1] M. Bodén and J. Wiles. Context-free and context-sensitive dynamics in recurrent neural networks. *Connection Science*, 12(3):197–210, 2000.

[2] M. Casey. The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. *Neural Computation*, 8(6):1135–1178, 1996.

[3] J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[4] S. L. Frank and M. Čerňanský. Generalization and systematicity in echo state networks. In *Proceedings of the 30th Cognitive Science Conference, Washington DC, USA*, pages 733–738, 2008.

[5] S. Haykin. *Neural networks – a comprehensive foundation*. Prentice-Hall, Upper Sadle River, New Jersey 07458, United States, 1994.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

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

[8] H. Jaeger. Short term memory in echo state networks. Technical Report GMD Report 152, German National Research Center for Information Technology, 2001.

[9] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[10] J. F. Kolen. The origin of clusters in recurrent neural network state space. In *Proceedings from the Sixteenth Annual Conference of the Cognitive Science Society*, pages 508–513. Hillsdale, NJ: Lawrence Erlbaum Associates, 1994.

[11] J. F. Kolen. Recurrent networks: state machines or iterated function systems? In D. S. T. J. L. E. M. C. Mozer, P. Smolensky and A. Weigend, editors, *Proceedings of the 1993 Connectionist Models Summer School*, pages 203–210. Erlbaum Associates, Hillsdale, NJ, 1994.

[12] Y. A. Kuznetsov. *Elements of applied bifurcation theory (2nd ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1998.

[13] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[14] M. Makula. Dynamical reservoir of echo state network. In J. Kelemen and V. Kvasnička, editors, *Cognition and Artificial Life VI*, pages 273–281, 2006.

[15] M. Makula and Ľ. Beňušková. Analysis and visualization of the dynamics of recurrent neural networks for symbolic sequences processing. In *Proceedings of the ICANN 2008, Part II*, pages 577–586. Springer-Verlag, 2008.

[16] M. Makula and M. Čerňanský. Neural networks based on echo states. In *Proceedings of the Cognition and Artificial Life IV*, pages 117–127, 2004.

[17] M. Makula, M. Čerňanský, and Ľ. Beňušková. Approaches based on markovian architectural bias in recurrent neural networks. In *SOFSEM 2004: Theory and Practice of Computer Science*, volume 2932/2003, pages 257–264. Springer-Verlag, 2004.

[18] C. Moore. Dynamical recognizers: real-time language recognition by analog computers. *Theoretical Computer Science*, 201(1–2):99–136, 1998.

[19] P. Rodriguez, J. Wiles, and J. L. Elman. A recurrent neural network that learns to count. *Connection Science*, 11:5–40, 1999.

[20] P. Tiňo, M. Čerňanský, and Ľ. Beňušková. Markovian architectural bias of recurrent neural networks. In *Intelligent Technologies – Theory and applications*, pages 17–23. IOS Press, 2002.

[21] P. Tiňo, M. Čerňanský, and Ľ. Beňušková. Markovian architectural bias of recurrent neural networks. *IEEE Transactions on Neural Networks*, 15(1):6–15, 2004.

[22] P. Tiňo and B. Hammer. Architectural bias in recurrent neural networks: Fractal analysis. *Neural Computation*, 15(8):1931–1957, 2003.

[23] P. Tiňo, B. G. Horne, and C. L. Giles. Attractive periodic sets in discrete time recurrent networks (with emphasis on fixed point stability and bifurcations in two-neuron networks). *Neural Computation*, 13(6):1379–1414, 2001.

[24] P. Tiňo, B. G. Horne, C. L. Giles, and P. C. Collingwood. Finite state machines and recurrent neural networks – automata and dynamical systems approaches. In E. Dayhoff and O. Omidvar, editors, *Neural Networks and Pattern Recognition*, page 171220. Academic Press, 1998.

[25] B. Tonkes, A. Blair, and J. Wiles. Inductive bias in context-free language learning. In *Proceedings of the Ninth Australian conference on Neural Networks*, pages 52–56, 1998.

[26] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report NC27599-3175, Department of Computer Science, University of North Carolina, Chapel Hill, Dec. 1997.

[27] P. Werbos. Backpropagation through time; what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[28] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280, 1989.

Selected Papers by the Author

M. Makula and Ľ. Beňušková. Interactive visualization of oligomer frequency in DNA. *Computing and Informatics*. in press, 2009.

M. Čerňanský, M. Makula and Ľ. Beňušková. Organization of the state space of a simple recurrent network before and after training on recursive linguistic structures. *Neural Networks*, 20, 2, pages 236–244, 2007.

M. Makula and Ľ. Beňušková. Analysis of state space of RNNs trained on a chaotic symbolic sequence. *Neural Network World*, 13, 3, pages 267–276, 2003.

M. Makula and Ľ. Beňušková. Analysis and visualization of the dynamics of recurrent neural networks for symbolic sequences processing. In *Lecture Notes in Computer Science*, Proceedings of the ICANN 2008, Part II, pages 577–586. Springer-Verlag, 2008.

M. Čerňanský, M. Makula and Ľ. Beňušková. Improving the state space organization of untrained recurrent networks. newblock In *Lecture Notes in Computer Science*, Proceedings of the INNS–NNN Symposia, pages 671–678, 2008.

M. Čerňanský, M. Makula, P. Trebatický, P. Lacko, Text correction using approaches based on Markovian architectural bias. In *EANN 2007, Proc. of the 10th Int. Conf. on Engineering Applications of Neural Networks*, pages 221–228, 2007.

M. Čerňanský and M. Makula. Feed-forward echo state networks. In *IEEE International Joint Conference on Neural Networks*, 3, pages 1479–1482, 2005.

M. Čerňanský and M. Makula. Processing linguistic sequences using recurrent neural networks. In *Cognition and Artificial Life VIII*, pages 81–86, 2008. (in Slovak)

M. Čerňanský and M. Makula. Processing symbolic sequences using echo state networks. In *Cognition and Artificial Life VII*, pages 93–98, 2007. (in Slovak)

M. Makula. Dynamical reservoir of echo state network. In *Cognition and Artificial Life VI*, pages 273–281, 2006. (in Slovak)

J. Kriška and M. Makula. Neuroevolution of augmenting topologies. In *Cognition and Artificial Life V*, pages 637–650, 2005. (in Slovak)

M. Makula, M. Čerňanský and Ľ. Beňušková. Approaches based on Markovian architectural bias in recurrent neural networks. In *Lecture Notes in Computer Science*: SOFSEM 2004 Theory and Practice of Computer Science, 2932/2003. Springer-Verlag, pages 257–264, 2004.

M. Čerňanský, M. Makula and Ľ. Beňušková. Processing symbolic sequences by recurrent neural networks trained by Kalman filter based algorithms. In *SOFSEM 2004: Theory and Practice of Computer Science*, 2, pages 58–65, 2004.

M. Makula and M. Čerňanský. Neural networks based on echo states. In *Cognition and Artificial Life IV*, pages 117–128, 2004. (in Slovak)

M. Čerňanský and M. Makula. Memory of randomly initialized recurrent neural networks. In *Cognition and Artificial Life IV*, pages 363–371, 2004. (in Slovak)

M. Makula and M. Čerňanský. Evaluation of neural network performance based on entropy. In *Student EEICT 2004: Proceedings of the international conference and competition of students scientific works*, pages 172–175, 2004.

M. Čerňanský and M. Makula. Extended Kalman filter approximation for training long short–term memory networks. In *ELITECH 2003. The sixth conference on Electrical engineering and information technology for PhD students*, pages 94–96, 2003.

M. Makula and M. Čerňanský. Echo state networks – wobbling trick. In *ELITECH 2003. The sixth conference on Electrical engineering and information technology for PhD students*, pages 100–102, 2003.

M. Makula. Visualization and analysis of the dynamics of recurrent neural networks for symbolic sequences processing. In *ELITECH 2002: 5th Scientific Conf. on Electrical Engineering & Information Technology for Ph.D. Students*, pages 103–105, 2002.