Recurrent neural networks with Iterated Function Systems dynamics

Peter Tiňo* **Georg Dorffner†**
Austrian Research Institute for Artificial Intelligence
Schottengasse 3
A-1010 Vienna, Austria
Email: petert,georg@ai.univie.ac.at

Abstract

We suggest a recurrent neural network (RNN) model with a recurrent part corresponding to iterative function systems (IFS) introduced by Barnsley [1] as a fractal image compression mechanism. The key idea is that **1)** in our model we avoid learning the RNN state part by having *non-trainable* connections between the context and recurrent layers (this makes the training process less problematic and faster), **2)** the RNN state part codes the information processing states in the symbolic input stream in a well-organized and intuitively appealing way. We show that there is a direct correspondence between the Rényi entropy spectra characterizing the input stream and the spectra of Rényi generalized dimensions of activations inside the RNN state space. We test both the new RNN model with IFS dynamics and its conventional counterpart with trainable recurrent part on two chaotic symbolic sequences. In our experiments, RNNs with IFS dynamics outperform the conventional RNNs with respect to information theoretic measures computed on the training and model generated sequences.

1 Introduction

It has been recognized for some time that conventional gradient based recurrent neural network (RNN) training methods such as BPTT or RTRL (see, for example, [9]) suffer from an exponentially decaying error signal when learning to latch important information pieces separated by longer time lags [10, 4]. Loosely speaking, RNNs latch important information by driving neural activations to a vicinity of a hyperbolic attractor. The norm of the Jacobian matrix of the RNN state transition part is smaller than 1 "near" hyperbolic attractors. When propagating the error signal through long time lags, the amplitude of the propagated signal vanishes exponentially fast.

Several approaches to deal with the problem have been proposed, such as the use of non-gradient optimization learning techniques [4], the extension of synaptic weights with a memory mechanism [15], or the enforcement of a constant non-vanishing error flow through specially designed "neural" units [11]. In this paper we suggest a new simple (but potentially powerful) alternative. Recurrent part of our RNN model corresponds to

*also with *Department of Computer Science and Engineering, Slovak University of Technology, Ilkovicova 3, 812 19 Bratislava, Slovakia*
†also with *Dept. of Medical Cybernetics and Artificial Intelligence, Univ. of Vienna*

Iterative Function Systems (IFS) introduced by Barnsley [1] as a fractal image compression mechanism. The key idea is that in our model we avoid learning the RNN state part by having *non-trainable* connections between the context and recurrent layers. This makes the training process less problematic and faster [10, 4, 7]. We refer to RNNs with non-trainable recurrent part driven by IFS dynamics as *iterative function system networks* (IFSNs).

There is yet another benefit in introducing RNNs with IFS dynamics. We show (in the multifractal theory framework) that the way IFSN organizes its state space reflects the input stream characteristics in a precise mathematical sense.

We test both the IFSNs and the conventional RNNs with trainable recurrent part on two chaotic symbolic sequences. In our experiments, IFSNs outperform the conventional RNNs with respect to information theoretic measures computed on the training and model generated sequences.

2 Preliminaries

We consider sequences $S = s_1 s_2 ...$ over a finite alphabet $\mathcal{A} = \{1, 2, ..., A\}$ generated by stationary information sources. The sets of all sequences over $\mathcal{A}$ with a finite number of symbols and exactly n symbols are denoted by $\mathcal{A}^+$ and $\mathcal{A}^n$, respectively. By S_i^j, $i \leq j$, we denote the string $s_i s_{i+1} ... s_j$, with $S_i^i = s_i$. Denote the (empirical) probability of finding an n-block $w \in \mathcal{A}^n$ in S by $\hat{P}_n(w)$. A string $w \in \mathcal{A}^n$ is said to be an allowed n-block in the sequence S, if $\hat{P}_n(w) > 0$. The set of all allowed n-blocks in S is denoted by $[S]_n$.

Statistical n-block structure in a sequence S is usually described through generalized entropy spectra. The original distribution of n-blocks, $\hat{P}_n(w)$, is transformed to the "twisted" distribution [25] (also known as the "escort" distribution [3])

$$Q_{\beta,n}(w) = \frac{\hat{P}_n^\beta(w)}{\sum_{v \in [S]_n} \hat{P}_n^\beta(v)}. \tag{1}$$

The entropy rate

$$h_{\beta,n} = \frac{-\sum_{w \in [S]_n} Q_{\beta,n}(w) \log Q_{\beta,n}(w)}{n} \tag{2}$$

of the twisted distribution $Q_{\beta,n}$ approximates the thermodynamic entropy density [25]

$$h_\beta = \lim_{n \to \infty} h_{\beta,n}. \tag{3}$$

Varying the parameter β amounts to scanning the original n-block distribution $\hat{P}_n$ – the most probable and the least probable n-blocks become dominant in the positive zero ($\beta = \infty$) and the negative zero ($\beta = -\infty$) temperature regimes, respectively. Varying β from 0 to ∞ amounts to a shift from all allowed n-blocks to the most probable ones by accentuating still more and more probable subsequences. Varying β from 0 to $-\infty$ accentuates less and less probable n-blocks with the extreme of the least probable ones. We note that the thermodynamic entropy densities are closely related to the β-order Rényi entropy rates [21] (cf. [25]).

We represent the n-blocks $u = u_1 u_2 ... u_n \in \mathcal{A}^n$ as points

$$u(x) = u_n(u_{n-1}(...(u_2(u_1(x)))...)) =$$

$$(u_n \circ u_{n-1} \circ ... \circ u_2 \circ u_1)(x), \quad x \in X,$$

where $X = [0,1]^N$, $N = \lceil \log_2 A \rceil$, and the maps $1, 2, ..., A$,

$$i(x) = kx + (1-k)t_i, \quad t_i \in \{0,1\}^N, \tag{4}$$

with $t_i \neq t_j$ iff $i \neq j$, act on X with a contraction coefficient $k \in (0, \frac{1}{2}]$. For $Y \subseteq X$, $u(Y) = \{u(x) |\ x \in Y\}$.

Denote the center $\{\frac{1}{2}\}^N$ of X by x_*. Given a sequence $S = s_1 s_2 ...$ over $\mathcal{A}$, we define its chaos game representation $CGR_k(S)$ as a sequence of points

$$CGR_k(S) = \left\{ S_1^{i+n-1}(x_*) \right\}_{i \geq 1}. \tag{5}$$

The chaos game representation codes the subsequence suffix structure in the following sense: if $v \in \mathcal{A}^+$ is a suffix of length $|v|$ of a string $u = rv$, $r, u \in \mathcal{A}^+$, then $u(X) \subset v(X)$, where $v(X)$ is an N-dimensional hypercube of side length $k^{|v|}$. Hence, the longer is the common suffix shared by two subsequences, the closer lie the end points of their chaos game representations. Our notion of chaos game representation of symbolic sequences is related to the original chaos game representation of DNA sequences proposed by Jeffrey [12].

3 Neural network architectures

The more conventional recurrent neural network (RNN) architecture we used (see figure 1) has an input layer $I^{(t)} = (I_1^{(t)}, ..., I_A^{(t)})$ with A neurons (to which the one-of-A codes of input symbols from $\mathcal{A} = \{1, ..., A\}$ are presented, one at a time), a hidden non-recurrent layer $H^{(t)}$, a hidden recurrent layer $R^{(t+1)}$, and an output layer $O^{(t)}$ having the same number of neurons as the input layer. Activations in the recurrent layer are copied with a unit time delay to the context layer $R^{(t)}$ that forms an additional input. At each time step t, the input $I^{(t)}$ and the context $R^{(t)}$ determine the future context

$$R_i^{(t+1)} = g\left(\sum_{j,k} W_{i,j,k} I_j^{(t)} R_k^{(t)} + T_{i,j}^R \right), \tag{6}$$

and the output $O^{(t)}$ (via a hidden layer $H^{(t)}$)

$$H_i^{(t)} = g\left(\sum_{j,k} Q_{i,j,k} I_j^{(t)} R_k^{(t)} + T_{i,j}^H \right),$$

$$O_i^{(t)} = g\left(\sum_{j} V_{i,j} H_j^{(t)} + T_i^O \right).$$

Here, g is the standard logistic sigmoidal function. $W_{i,j,k}$, $Q_{i,j,k}$ and $T_{i,j}^R$, $T_{i,j}^H$ are second-order real valued weights and thresholds, respectively. $V_{i,j}$ and T_i^O are the weights and

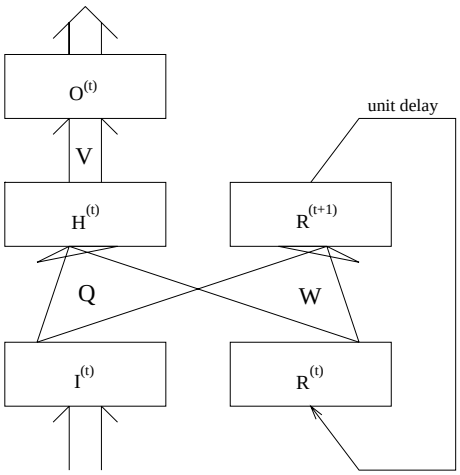

Figure 1: Recurrent neural network architecture. The recurrent weights W can either be learned, or fixed prior to the training process, in which case the activation functions in the recurrent layer $R^{(t+1)}$ are linear and the recurrent part $[I^{(t)} + R^{(t)} \to R^{(t+1)}]$ of the network implements an affine IFS.

thresholds, respectively, associated with the hidden to output layer connections.

The proposed IFSN has the same architecture with the exception that the recurrent neurons' activation function is linear and the weights W and thresholds T^R are fixed, so that the network dynamics (6) is given by (4): x is the current state $R^{(t)}$ and $i(x)$ is the next state $R^{(t+1)}$, provided the input symbol at time t is i. Such a dynamics is equivalent to the dynamics of the IFS (4) driven with the symbolic sequence appearing at the network input. The attractor of the IFS is either the whole hypercube $X = [0, 1]^N$ (when $k = 1/2$), or a Sierpinski sponge [13] (when $k < 1/2$). The trainable parts of IFSNs form a feed-forward architecture $[I^{(t)} + R^{(t)} \to H^{(t)} \to O^{(t)}]$.

The input sequence S feeding the IFSN is translated in the network recurrent part into the chaos game representation $CGR_k(S)$. The $CGR_k(S)$ forms clusters of state neuron activations, where points lying in a close neighborhood code histories with a long common suffix (e.g. histories that are likely to produce similar continuations), whereas histories with different suffices (and potentially different continuations) are mapped to activations lying far from each other. This organization of IFSN state space corresponds to an underlying (and often unstated) assumption about the network output smoothness. In addition, in RNN community dealing with symbolic tasks, it is rather common to stabilize RNN performance by performing a vector quantization on recurrent neuron activations [8, 24, 6, 19, 17, 5]. In our model, dense areas in the IFSN state space correspond to contexts with long common suffices and are given more attention by the vector quantizer. Consequently, more information processing states of the extracted machines are devoted to these potentially "problematic" contexts. This directly corresponds to the idea of variable memory length Markov models [22, 23], where the length of the past history considered in order to predict the future is not fixed, but context dependent.

4 Experiments

We tested the performance of our model on two data sets. The first data set is a long sequence (10.000 items) of differences between the successive activations of a real laser in a chaotic regime. The laser activations were retrieved from http://www.cs.colorado.edu/~andreas/Time-Series/SantaFe.html. The sequence was quantized into a symbolic stream over four symbols corresponding to low and high positive/negative laser activity changes: $[0, 50)$, $[50, 200]$, $(-64, 0]$ and $[-200, -64)$ correspond to symbols 1,2,3 and 4 respectively. The second data set is an artificial sequence of 15.000 points generated by iterating the logistic map $F(x) = rx(1-x),\ x \in [0,1]$ with the control parameter r set to the Misiurewicz parameter value $r \approx 3.9277370012867...$ [25]. Symbolic sequence over two symbols was obtained through a generating partition defined by the critical point 1/2.

We used the data sets to train both the IFSNs and the RNNs with trainable recurrent weights[1]. The networks were trained to perform the next symbol prediction. Trained models are compared via thermodynamic entropy spectra (eq. (2)) computed on the training and model generated sequences. The recurrent networks were used to generate sequences on their own by initiating them with the first 10 training sequence symbols and then generating the continuation with respect to the output probabilities

$$P_i^{(t)} = \frac{O_i^{(t)}}{\sum_{j=1}^{A} O_j^{(t)}}, \quad i = 1, ..., A.$$

The symbol generated at time t appears at the network input at time $t+1$.

As expected [8, 6, 19, 17, 5], finite state RNN representations constructed through vector quantization of trained networks' state space [20] have better modeling behaviour than the networks themselves. The graph in figure 2 shows the 6-block[2] based entropy spectra: the bold line with diamonds corresponds to the training sequence, dashed lines with squares and crosses correspond to sequences generated by extracted stochastic machines from trained conventional RNNs with 2 and 5 recurrent neurons[3], respectively, solid line with triangles corresponds to the IFSN based[4] machine. The machines were of comparable size of about 20 states.

Figure 3 presents the entropy spectra for the logistic map experiment. The spectra are based on 10-block statistics. There are 3 recurrent and 3 non-recurrent hidden neurons in the "conventional" RNN and the corresponding extracted machine had 5 states. IFSN has 1 recurrent[5] and 3 non-recurrent hidden neurons and the extracted machine has 6 states.

The IFSNs outperform the conventional RNNs on high probability 6-blocks[6] in the laser data experiment, and beat the conventional RNN on rare 10-blocks in the logistic

[1] the recurrent weights were trained using the RTRL procedure

[2] the block length for computing the entropy spectra statistics is chosen with respect to two criteria: 1. the blocks should be as long as possible, 2. the blocks should not be too long so that the statistics over n-blocks is significant. For more details see [20]

[3] the 2- and 5-recurrent neuron RNNs have 3 and 6 hidden non-recurrent neurons, respectively

[4] the IFSN has 3 hidden non-recurrent neurons and, obviously, 2 recurrent neurons ($A = 4$ and $N = \lceil \log_2 A \rceil = 2$)

[5] the training sequence is a binary sequence

[6] on rare 6-blocks, the performance of IFSN is slightly worse than that of the 2-recurrent neuron conventional RNN

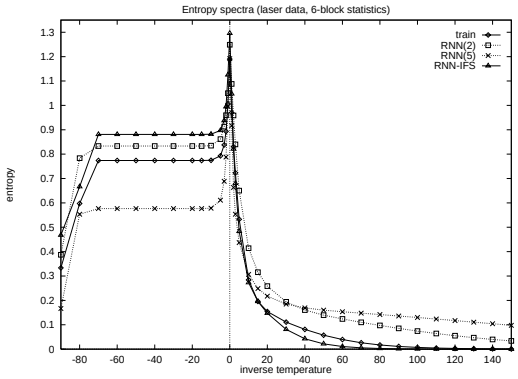

Figure 2: Entropy spectra (based on 6-block statistics) of the training sequence and model generated sequences in the laser data experiment. Labels `RNN(2)`, `RNN(5)` and `RNN-IFS` correspond to the conventional RNNs with 2 and 5 recurrent neurons, and IFSN with 2 recurrent neurons, respectively. The spectrum of the training sequence is labeled by `train`.

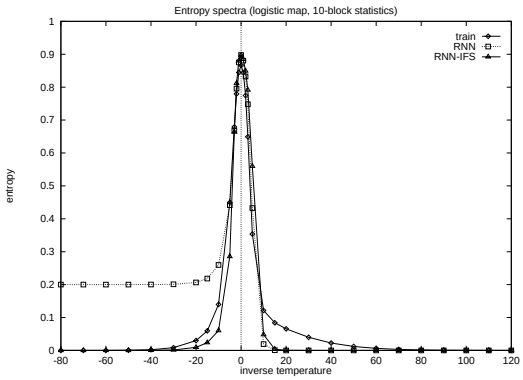

Figure 3: Logistic map experiment. Shown are entropy spectra of the training and model generated sequences. The spectra are based on 10-block statistics. The labels on spectra are analogical to the labels in the previous figure.

map experiment.

5 Theoretical description of state representations in IFSNs

Although previous approaches to the analysis of RNN state space organization did point out the correspondence between IFSs and RNN recurrent part $[I^{(t)} + R^{(t)} \rightarrow R^{(t+1)}]$ [16, 14], due to nonlinearity of recurrent neurons' activation function, they did not manage to provide a deeper insight into the RNN state space structure (apart from observing an apparent fractal-like clusters corresponding to nonlinear IFS attractor). In the following, we show that the IFSN state space organization reflects statistical characteristics of the input stream in a strict mathematical sense.

The space $\mathcal{A}^\omega$ of all one-sided infinite sequences over the alphabet $\mathcal{A} = \{1, ..., A\}$, endowed with the standard n-cylinder based metric d_K

$$\forall u, v \in \mathcal{A}^\omega; \quad d_K(u, v) = \sum_{i=1}^{\infty} \frac{|u_i - v_i|}{K^i}, \quad K > 1,$$

forms a metric space $(\mathcal{A}^\omega, d_K)$.

Assume, the input sequence is a "typical" sequence generated by a stationary ergodic information source $\mathcal{M}$. Its topological entropy is equal (up to a scaling factor $\log K$) to the fractal dimension, in the metric space $(\mathcal{A}^\omega, d_K)$, of the set of all sequences generated by $\mathcal{M}$ [2]. Although not done in this paper, we can prove that (when $K = 1/k$) the attractor Q of IFSN dynamics, endowed with the Euclidean metric d_E, forms a metric space (Q, d_E) that is metrically equivalent to $(\mathcal{A}^\omega, d_K)$, and hence share the same fractal dimensions. It follows that, in the limit of infinite length input stream, the fractal dimension of the set of recurrent neurons' activations directly corresponds to the topological entropy of the information source $\mathcal{M}$.

More generally, the information source induces a measure on the set of sequences it generates. The *dynamical* Rényi entropy spectra on generated sequences coincide with the *static* multifractal spectra of generalized Rényi dimensions (in $(\mathcal{A}^\omega, d_K)$) of the set of all sequences generated by $\mathcal{M}$. Consequently, the *static* multifractal generalized Rényi dimensions of recurrent neurons' activations directly correspond to the *dynamical* Rényi entropy spectra of typical sequences generated by $\mathcal{M}$. The complete proof will be published elsewhere.

6 Conclusion

We propose to reduce (although not completely eliminate) the information latching problem in training recurrent neural networks (RNN) [10, 4] by introducing a novel RNN architecture with non-trainable recurrent weights. The dynamics of our RNN model corresponds to iterative function systems used in chaos game representations of symbolic sequences [18, 12]. RNNs with IFS dynamics are related to the variable memory length Markov models.

We present a sketch of the proof that the fixed dynamics of our RNN model does indeed represent the relevant statistical features of training sequences. In particular, the sequence space is metrically equivalent to the Euclidean space of recurrent neurons' activations and the Rényi entropy spectrum of the training sequence coincides with the multifractal spectrum of generalized Rényi dimensions of the recurrent neurons' activations.

In the experiments with two chaotic sequences, our RNN model performs at least as well as the conventional RNNs. Yet, unlike conventional RNNs, the RNNs with IFS dynamics can be trained as feed-forward networks in a faster and less complicated manner.

Acknowledgements

This work was supported by the Austrian Science Fund (FWF) within the research project "Adaptive Information Systems and Modeling in Economics and Management Science" (SFB 010). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Science and Transport.

References

[1] M.F. Barnsley. *Fractals everywhere*. Academic Press, New York, 1988.

[2] L. Barreira, Y. Pesin, and J. Schmeling. On a general concept of multifractality: Multifractal spectra for dimensions, entropies, and lyapunov exponents. multifractal rigidity. *Chaos: an Interdisciplinary Journal of Nonlinear Science*, 7(1):27–53, 1996.

[3] C. Beck and F. Schlogl. *Thermodynamics of chaotic systems*. Cambridge University Press, Cambridge, UK, 1995.

[4] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[5] M.P. 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.

[6] S. Das and M.C. Mozer. A unified gradient–descent/clustering architecture for finite state machine induction. In J.D. Cowen, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 19–26. Morgan Kaufmann, 1994.

[7] K. Doya. Bifurcations in the learning of recurrent neural networks. In *Proc. of 1992 IEEE Int. Symposium on Circuits and Systems*, pages 2777–2780, 1992.

[8] C.L. Giles, C.B. Miller, D. Chen, H.H. Chen, G.Z. Sun, and Y.C. Lee. Learning and extracting finite state automata with second–order recurrent neural networks. *Neural Computation*, 4(3):393–405, 1992.

[9] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison–Wesley, Redwood City, CA, 1991.

[10] J. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Master's thesis, Institut fur Informatik, Technische Universitat, Munchen, 1991.

[11] J. Hochreiter and J. Schmidhuber. Long short term memory. Technical Report FKI-207-95, Fakultat fur Informatik, Technische Universitat, Munchen, 1995.

[12] J. Jeffrey. Chaos game representation of gene structure. *Nucleic Acids Research*, 18(8):2163–2170, 1990.

[13] R. Kenyon and Y. Peres. Measures of full dimension on affine invariant sets. *Ergodic Theory and Dynamical Systems*, 16:307–323, 1996.

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

[15] T. Lin, B.G. Horne, P. Tino, and C.L. Giles. Learning long-term dependencies in narx recurrent neural networks. *IEEE Transactions on Neural Networks*, 7(6):1329–1338, 1996.

[16] P. Manolios and R. Fanelli. First order recurrent neural networks and deterministic finite state automata. *Neural Computation*, 6(6):1155–1173, 1994.

[17] 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 J.E. Dayhoff and O. Omidvar, editors, *Neural Networks and Pattern Recognition*, pages 171–220. Academic Press, 1998.

[18] P. Tiňo and M. Koteles. Modeling complex sequences with neural and hybrid neural based approaches. Technical Report STUFEI-DCSTR-96-49, Slovak University of Technology, Bratislava, Slovakia, September 1996.

[19] P. Tiňo and J. Sajda. Learning and extracting initial mealy machines with a modular neural network model. *Neural Computation*, 7(4):822–844, 1995.

[20] P. Tiňo and V. Vojtek. Modeling complex sequences with recurrent neural networks. In G.D. Smith, N.C. Steele, and R.F. Albrecht, editors, *Artificial Neural Networks and Genetic Algorithms*, pages – to appear. Springer Verlag Wien New York, 1998.

[21] A. Renyi. On the dimension and entropy of probability distributions. *Acta Math. Hung.*, (10):193, 1959.

[22] D. Ron, Y. Singer, and N. Tishby. The power of amnesia. In *Advances in Neural Information Processing Systems 6*, pages 176–183. Morgan Kaufmann, 1994.

[23] D. Ron, Y. Singer, and N. Tishby. The power of amnesia. *Machine Learning*, 25, 1996.

[24] R.L. Watrous and G.M. Kuhn. Induction of finite–state automata using second–order recurrent networks. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 309–316, 1992.

[25] K. Young and J.P. Crutchfield. Fluctuation spectroscopy. In W. Ebeling, editor, *Chaos, Solitons, and Fractals, special issue on Complexity*, 1993.