On early stages of learning in connectionist models with feedback connections

Peter Tiňo
School of Computer Science
University of Birmingham
Birmingham B15 2TT, UK
P.Tino@cs.bham.ac.uk

Barbara Hammer
University of Osnabrück
D-49069 Osnabrück, Germany
hammer@informatik.uni-osnabrueck.de

Abstract

We have recently shown that when initiated with "small" weights, many connectionist models with feedback connections are inherently biased towards Markov models, i.e. even prior to any training, dynamics of the models can be readily used to extract finite memory machines (Tiňo, Čerňanský, & Beňušková 2004; Hammer & Tiňo 2003). In this study we briefly outline the core arguments for such claims and generalize the results to recursive neural networks capable of processing ordered trees. In the early stages of learning, the compositional organization of recursive activations has a Markovian structure: Trees sharing a top subtree are mapped close to each other. The deeper is the shared subtree, the closer are the trees mapped.

Introduction

There is a considerable amount of literature devoted to connectionist processing of sequential symbolic structures. For example, researchers have been interested in formulating models of human performance in processing linguistic patterns of various complexity (e.g. (Christiansen & Chater 1999)).

It has been known for some time that when training connectionist models with feedback connections to process symbolic sequences, activations of recurrent units display a considerable amount of structural differentiation even *prior to learning* (Christiansen & Chater 1999; Kolen 1994b; 1994a; Manolios & Fanelli 1994). Following (Christiansen & Chater 1999), we refer to this phenomenon as the *architectural bias of connectionist models*.

We have recently shown, both empirically and theoretically, the meaning of the architectural bias: when initiated with "small" weights, many connectionist models with feedback connections are inherently biased towards Markov models, i.e. even prior to any training, the model dynamics can be readily used to extract finite memory machines (Hammer & Tiňo 2003; Tiňo, Čerňanský, & Beňušková 2004). In this paper we wish to expose these ideas to the community interested in cognitive aspects of connectionist modeling and to show how the previous results can be generalized to a wider class of compositional connectionist models, namely recursive neural networks (RecNN) capable of processing tree-structured data (Sperduti & Starita 1997; Frasconi, Gori, & Sperduti 1998).

Connectionist models with feedback connections are initialized as finite memory processors

Most connectionist models of sequence processing are endowed with memory structure in the form of delay feedback connections among a subset of hidden units called recurrent neurons. From an abstract point of view, we consider such models non-autonomous dynamical systems on activation space of recurrent units, $\mathcal{R}$, i.e. systems evolving in time on $\mathcal{R}$ with dynamics dependent on external inputs.

Assume inputs driving the model are taken from a finite alphabet $\mathcal{A}$ of A symbols. Then given a symbol $s \in \mathcal{A}$ presented at the input at time t and activations of recurrent units from the previous step, $\mathbf{R}^{(t-1)}$, activations of the recurrent units at time t, $\mathbf{R}^{(t)}$, are updated via a map $f_s : \mathcal{R} \to \mathcal{R}$,

$$\mathbf{R}^{(t)} = f_s(\mathbf{R}^{(t-1)}).$$

This notation can be extended to sequences over $\mathcal{A}$ by postulating for empty string e, $f_e = Id$[1] , and for any symbol $s \in \mathcal{A}$ and any finite string U over $\mathcal{A}$ (including e), $f_{Us} = f_s \circ f_U$.

Given an input sequence $s_1 s_2 ...$ over $\mathcal{A}$, and initial recurrent activation vector $\mathbf{R}^{(0)}$, the recurrent activations after t time steps are

$$\mathbf{R}^{(t)} = f_{s_1 ... s_t}(\mathbf{R}^{(0)}).$$

Now, it is the case that in many connectionist models the dynamic maps f_s, $s \in \mathcal{A}$, are initialized as contractive mappings. This is due to the usual strategy of starting with small connection weights. When all f_s, $s \in \mathcal{A}$, are contractions, there is a constant $0 \leq C < 1$, such that for all $s \in \mathcal{A}$ and for all $\mathbf{R}, \mathbf{R}'$,

$$|f_s(\mathbf{R}) - f_s(\mathbf{R}')| \leq C \cdot |\mathbf{R} - \mathbf{R}'|. \quad (1)$$

To show that contractive dynamic maps f_s lead to Markovian organization of recurrent activations $\mathbf{R}$, consider a

[1] Id is the identity map

string $S = s_1 s_2 ... s_r$ over $\mathcal{A}$. For any two strings U, V over $\mathcal{A}$ we have (Tiňo, Čerňanský, & Beňušková 2004)

$$|f_{US}(\mathbf{R}) - f_{VS}(\mathbf{R})| \leq C^r \cdot |f_U(\mathbf{R}) - f_V(\mathbf{R})| \quad (2)$$
$$\leq C^{|S|} \cdot diam(\mathcal{R}), \quad (3)$$

where $diam(\mathcal{R})$ is diameter of the activation space $\mathcal{R}$ of recurrent units. Hence, no matter what state $\mathbf{R}^{(0)}$ the model is initiated with, the neural codes $f_P(\mathbf{R}^{(0)})$, $f_Q(\mathbf{R}^{(0)})$ of two sequences P, Q sharing a long common suffix S will lie close to each other. Furthermore, the longer is the common suffix shared by P and Q, the closer lie $f_P(\mathbf{R}^{(0)})$ and $f_Q(\mathbf{R}^{(0)})$.

Now, imagine that while observing an input stream $S = s_1 s_2 ...$ we have at our disposal only a finite memory length L, i.e. all we can process are the most recent L input symbols. We can ask a question: How different would be recurrent activations $\mathbf{R}^{(t)}$ (coding histories of inputs up to time t) if instead of processing the whole sequence S we processed only the last L symbols of S, starting in the initial state $\mathbf{R}^{(0)}$? By the analysis above, the difference would be at most $C^L \cdot diam(\mathcal{R})$, which can be made arbitrarily small by increasing the finite memory length L.

The types of models studied above processed sequential data, but more involved generalized neural models for processing structured tree-like data have also appeared in the literature (Sperduti & Starita 1997; Frasconi, Gori, & Sperduti 1998). The idea is simple. Sequences can be considered ($k = 1$)-ary trees, where each node has at most one child. The processing flows in a bottom-up fashion, from leaves[2] to the root. Having computed the state information $\mathbf{R}^{(t-1)} \in \mathcal{R}$ about what nodes have already been processed so far, based on the label of the current node (symbol s_t at time t), we determine the updated representation $\mathbf{R}^{(t)} \in \mathcal{R}$ of the history of symbols seen so far, including s_t, as $\mathbf{R}^{(t)} = f_{s_t}(\mathbf{R}^{(t-1)})$. Initial activation in the leaf is $\mathbf{R}^{(0)} \in \mathcal{R}$. When processing ($k > 1$)-ary ordered trees, in order to determine activation vector $\mathbf{R}^{(n)} \in \mathcal{R}$ for a node n labeled by symbol $s \in \mathcal{A}$, fed by k child nodes n_1, n_2, ..., n_k, we need to take into account state informations $\mathbf{R}^{(n_1)} \in \mathcal{R}$, $\mathbf{R}^{(n_2)} \in \mathcal{R}$, ..., $\mathbf{R}^{(n_k)} \in \mathcal{R}$, that code processed sub-trees rooted in the k child nodes. So the generalized "state-transition" map $f_s : \mathcal{R}^k \to \mathcal{R}$ has now the form

$$\mathbf{R}^{(n)} = f_s(\mathbf{R}^{(n_1)}, \mathbf{R}^{(n_2)}, ..., \mathbf{R}^{(n_k)}).$$

The initial activation vectors in leaves are, as before, $\mathbf{R}^{(0)} \in \mathcal{R}$. Once the whole tree has been recursively processed, the activation vector in the root, $\mathbf{R}^{(root)} \in \mathcal{R}$, codes the topology and node labelings of the tree. An illustration of recursive processing of a binary tree is presented in figure 1.

Consider trees G_1, G_2 that *share a common top subtree G of depth d*. In other words, up to certain depth d, the top structures near the root in G_1 and G_2 are identical and equal to G. The trees G_1, G_2 differ only "bellow" G, i.e. at levels

[2] in case of sequences there is only one leaf corresponding to the first symbol in the sequence

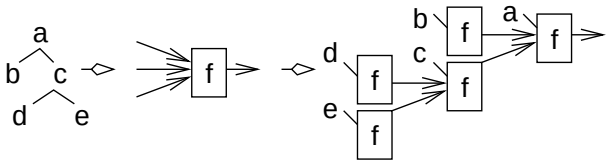

Figure 1: Example of a computation with a recursive function for a given binary tree.

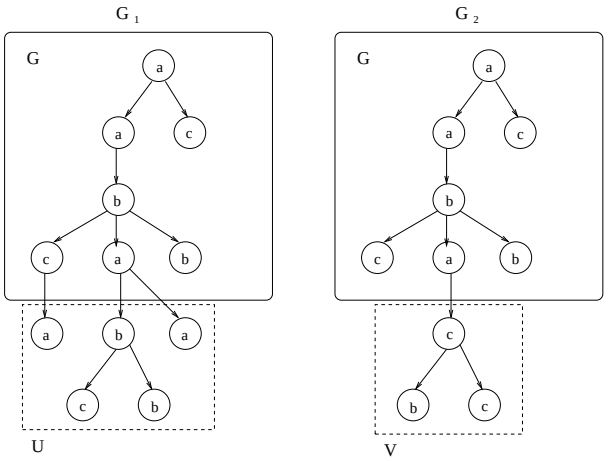

Figure 2: Example of depth-6 trees G_1, G_2 sharing a top-level tree G of depth 4. G_1 and G_2 differ only in subgraphs U, V feeding the leaves of G and the differences appear at levels 5 and 6.

exceeding the depth of G. As an example, consider trees G_1, G_2 in figure 2. The trees share the top portion G of depth $d = 4$ and differ only in subgraphs U, V feeding the level-4 leaves of G. The differences between G_1 and G_2 appear only at levels 5 and 6.

Again, assume that all the maps f_s, $s \in \mathcal{A}$, are contractions, i.e. there is a constant $0 \leq C < 1$, such that for each $s \in \mathcal{A}$ and for all $\bar{\mathbf{R}} = (\mathbf{R}^{(n_1)}, \mathbf{R}^{(n_2)}, ..., \mathbf{R}^{(n_k)})$, $\bar{\mathbf{R}}' = (\mathbf{R}'^{(n_1)}, \mathbf{R}'^{(n_2)}, ..., \mathbf{R}'^{(n_k)})$, we have

$$|f_s(\bar{\mathbf{R}}) - f_s(\bar{\mathbf{R}}')|_{\mathcal{R}} \leq C \cdot |\bar{\mathbf{R}} - \bar{\mathbf{R}}'|_{\mathcal{R}^k}. \quad (4)$$

Note that the Euclidean norms appearing in the above equation are defined on spaces of different dimensionality.

Denote by $\mathbf{R}^{(G_1)}$ and $\mathbf{R}^{(G_2)}$ the activation vectors in the roots of graphs $G_!$ and G_2, respectively. By realizing that for

$$|(\mathbf{R}^{(n_1)}, \mathbf{R}^{(n_2)}, ..., \mathbf{R}^{(n_k)}) - (\mathbf{R}'^{(n_1)}, \mathbf{R}'^{(n_2)}, ..., \mathbf{R}'^{(n_k)})|_{\mathcal{R}^k}$$

we have

$$|\bar{\mathbf{R}} - \bar{\mathbf{R}}'|_{\mathcal{R}^k} \leq \sum_{j=1}^{k} |\mathbf{R}^{(n_j)} - \mathbf{R}'^{(n_j)}|_{\mathcal{R}}$$
$$\leq k \cdot \max_j |\mathbf{R}^{(n_j)} - \mathbf{R}'^{(n_j)}|_{\mathcal{R}} \quad (5)$$

and using arguments analogous to those employed in the case of sequential data, we arrive at

$$|\mathbf{R}^{(G_1)} - \mathbf{R}^{(G_2)}| \leq k^{d-1} \cdot C^d \cdot diam(\mathcal{R})$$

$$= \quad C \cdot (k \cdot C)^{d-1} \cdot diam(\mathcal{R}).$$

It follows that, as long as $C < 1/k$, no matter what subgraphs of G_1, G_2 lie "bellow" the shared top tree G, the representations $\mathbf{R}^{(G_1)}$ and $\mathbf{R}^{(G_2)}$ of G_1 and G_2 will lie close to each other. Furthermore, the deeper is the common top tree G shared by G_1 and G_2, the closer lie the representations $\mathbf{R}^{(G_1)}$, $\mathbf{R}^{(G_2)}$. So when the maps f_s, $s \in \mathcal{A}$, are contractions with contraction coefficient $C < \frac{1}{k}$, the recursive neural network will have the "Markovian" coding property mapping trees sharing deep top subtrees close to each other.

Discussion and conclusion

Recently, we have extended the work of Kolen and others (e.g. (Christiansen & Chater 1999; Kolen 1994b; 1994a; Manolios & Fanelli 1994)) by pointing out that when initiated with "small" weights, connectionist models for processing sequential data are inherently biased towards Markov models, i.e. *even prior to any training*, the internal dynamics of the models can be readily used to extract finite memory machines (Tiňo, Čerňanský, & Beňušková 2004; Hammer & Tiňo 2003). In other words, even without any training, or in the early stages of learning, the recurrent activation clusters are perfectly reasonable and are biased towards finite-memory computations. We have also shown that in such cases, a rigorous analysis of fractal encodings in the model space can be performed (Tiňo & Hammer 2004).

Our analysis is general and applies to any connectionist model for processing sequential data that is initialized with contractive dynamics. In fact, this initialization strategy is quite common and alternative initialization schemes would need to be properly justified, as a-priori introduction of more complicated dynamical regimes (evolving e.g. along periodic orbits) can complicate the training process (Tiňo, Čerňanský, & Beňušková 2004). We plan to evaluate Markovian models extracted from connectionist models in the early stages of learning along the lines of (Tiňo, Čerňanský, & Beňušková 2004) (e.g. using a neural self-organizing map as a quantizing extraction tool) in the context of connectionist studies of language learning/processing in the cognitive science community.

In this paper we have further extended our results to a more general class of connectionist models with feedback connections, namely recursive neural networks (RecNN) (Sperduti & Starita 1997; Frasconi, Gori, & Sperduti 1998) capable of processing data represented as ordered trees. We have shown that the notions of finite memory and Markovian state organization generalize to processing trees by RecNN. In the early stages of learning, the compositional organization of recursive activations has a Markovian structure: Trees sharing a top subtree are mapped close to each other. The deeper is the shared subtree, the closer are the trees mapped to each other.

References

Christiansen, M., and Chater, N. 1999. Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science* 23:417–437.

Frasconi, P.; Gori, M.; and Sperduti, A. 1998. A general framework of adaptive processing of data structures. *IEEE Transactions on Neural Networks* 9(5):768–786.

Hammer, B., and Tiňo, P. 2003. Neural networks with small weights implement finite memory machines. *Neural Computation* 15(8):1897–1926.

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

Kolen, J. 1994b. Recurrent networks: state machines or iterated function systems? In Mozer, M.; Smolensky, P.; Touretzky, D.; Elman, J.; and Weigend, A., eds., *Proceedings of the 1993 Connectionist Models Summer School*. Hillsdale, NJ: Erlbaum Associates. 203–210.

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

Sperduti, A., and Starita, A. 1997. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks* 8(3):714–735.

Tiňo, P., and Hammer, B. 2004. Architectural bias in recurrent neural networks: Fractal analysis. *Neural Computation* 15(8):1931–1957.

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