Lazy Neural Network Learning for Building Symbolic Transducers

Stefan Wermter
University of Sunderland
Computing and Information Systems
Sunderland SR6 0DD, United Kingdom
stefan.wermter@sunderland.ac.uk

Abstract

Recurrent artificial neural networks can provide essential computational models and systems for bridging the gap between neuroscience and cognitive science. However, it is essential to understand better how a recurrent network learns and what it represents after learning. This paper describes new dynamic methods for the interpretation of recurrent neural networks. While most previous work on interpretation has focused on interpreting the final state of non-recurrent networks, we particularly focus on the process of learning as well as the final state of recurrent networks. Analyzing the dynamics of the learning of simple recurrent networks we found a "lazy learning" strategy which led to neural representations after learning which can be described as symbolic transducers.

Introduction

The interpretation of recurrent networks is more difficult than the interpretation of non-recurrent feedforward networks since the previous context in recurrent networks has an important dynamic effect within these networks. The internal states in recurrent networks do not only depend on the input but also on the internal state of the local memory [1, 2, 3]. Therefore, the focus has been primarily on smaller recurrent networks and artificially generated data. For instance, an interesting current approach interprets the training of a simple recurrent network with two input, two output and two internal elements to learn the sequence $a^n b^n$ [6]. It has been found that the network behaved like a spiral which moved to and from a fix point. While this seems a plausible interpretation of the behavior of recurrent networks trained for learning the sequences $a^n b^n$ different interpretations have to be expected if we move to different tasks and data sets closer to real-world scenarios.

In the past we have developed a large system for spoken language analysis which makes extensive use of simple recurrent networks [5, 4]. The spoken input is recognized by a speech recognizer and analyzed at syntactic, semantic and dialog levels. Furthermore, cognitive constraints like incremental analysis, parallel syntax and semantics, robust processing of corrections, etc are part of the system. However, so far we did not focus on the interpretation of the learning process and the interpretation of the neural knowledge. Here we are primarily interested in a detailed interpretation of the learning behavior as well as a symbolic interpretation of the learned knowledge after training.

In this paper we will focus on the analysis of the dynamic learning behavior of simple recurrent networks using a syntactic transformation task. The task for the network is to process sentences and associate their syntactic categories at the phrasal level, e.g. noun phrase, prepositional phrase etc. In order to interpret the behavior of recurrent networks it is not only essential that they learn a certain task but it is also important how the network reaches its performance.

Global Learning Behavior

Often, the interpretation of the learning behavior is just demonstrated with the learning curve of the overall error reduction over time. We believe that this is just the first step of a more detailed analysis although this learning curve provides first hints about the performance of a network over the training time. In general within our large spoken language system we have trained networks with many sentences using a corpus of several thousand words. For the sake of demonstrating the detailed learning behavior we will focus on the analysis of 15 of these sentences with 76 words from a domain of meeting arrangements. Here we concentrate on these sentences with 76 words in order to demonstrate a detailed single analysis of the underlying patterns.

The actually occurring syntactic basic categories are noun (n), verb (v), adverb (a), adjective (j), preposition (p), determiner (d) and pronoun (u). The abstract phrasal categories are noun group (ng), verb group (vg), and prepositional group (pg). The task of the recurrent network is to learn to assign phrasal categories based on basic syntactic categories for supporting a robust flat understanding of spontaneously spoken language. Below we show some translated example utterances from the original German meeting corpus [5] together with the syntactic categories at the basic and the phrasal level.

- I (u → ng) thought (v → vg) in (r → pg) the (d → pg) next (j → pg) week (n → pg)
- That (u → ng) is (v → vg) the (d → ng) Thursday (n → ng) after (r → pg) Easter (n → pg)

090000

Based on these seven basic syntactic and three phrasal syntactic categories we use a simple recurrent network (SRN) [1] with seven input units, three internal units and three output units (the networks in the actual system contain more categories and have been trained with several thousand words, but for illustration purposes we restrict ourselves to this smaller network). The learning rate was 0.05, momentum 0.9. The weight updates were performed incrementally after each training pattern. Each training pattern consisted of the basic syntactic category at the input layer and the abstract phrasal category at the output layer. 200000 patterns were presented to this simple recurrent network and figure 1 shows the learning curve with the overall sum squared error over time.

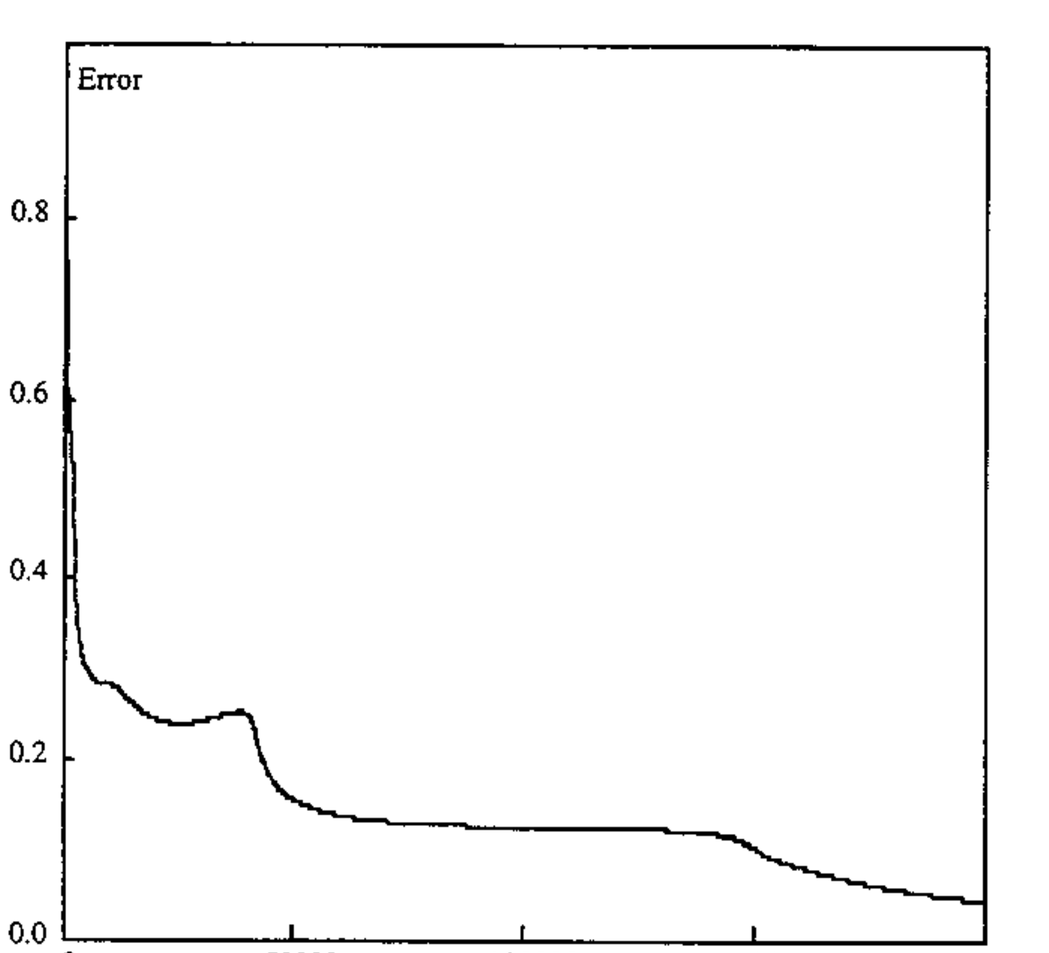

Figure 1: Learning curve for abstract syntactic categorization

The learning curve shows that the speed of learning is quite different. Furthermore, we can see different steps during the learning process. In the beginning, learning proceeds quickly, but later learning is slower and it takes longer times to make significant improvements. For instance, between 70000 and 140000 it seems that learning is about to finish before there is a final significant improvement. In the following subsection we will focus on a more detailed analysis and the reasons of these various learning steps.

Stepwise Dynamic Analysis of the Learning Behavior

First, we will examine how the network reaches its performance. We start the analysis directly after the random initialization of the weights. This is the state before learning starts.

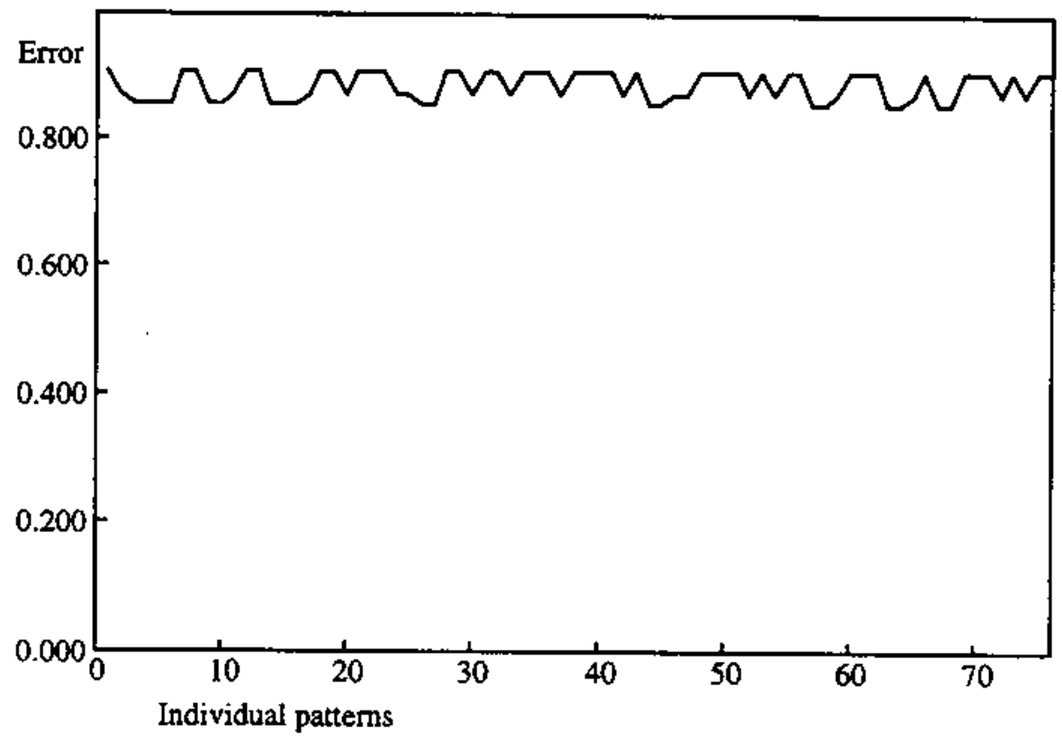

Figure 2: Performance for individual patterns before learning

We want to give an overview over the overall performance for all input patterns at different time steps. Therefore we show the error for each of the 76 patterns of the demonstration set at different time steps. Figure 2 shows the individual error of 76 patterns before training. Based on the random initialization all patterns show a relatively high error. For a random start initialization it is to be expected that the values of an output element differ from the desired value 0 or 1 by 0.5. Therefore the expected error for an individual pattern for three output elements is $\sqrt{0.5^2 + 0.5^2 + 0.5^2} = 0.866$. This expected error value is confirmed in this figure.

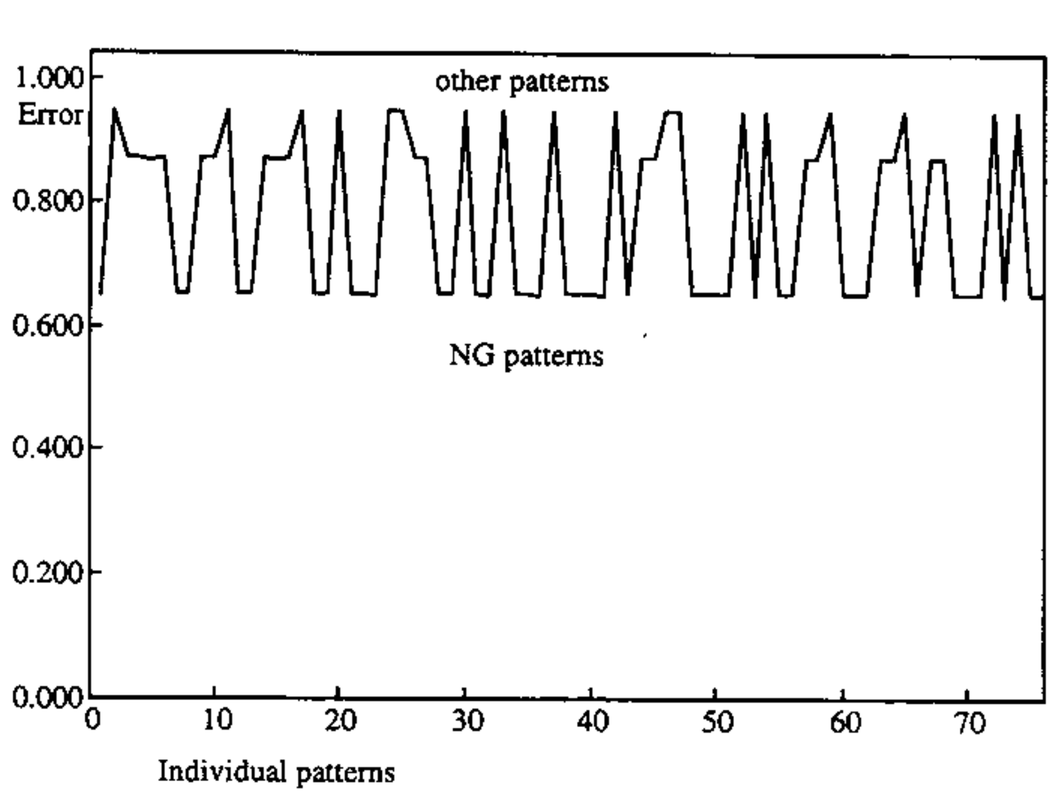

Figure 3: Performance for individual patterns after 100 training patterns

As shown in figure 1 the error decreases quickly at the start of the training. The state after 100 patterns of the

000061

training set is shown in figure 3. First, we can observe that after 100 patterns of training, the error for some of the shown 76 patterns could be reduced significantly. Other patterns still show a high error. Obviously, the network has started to learn pattern selectively.

A more detailed analysis revealed that the patterns with a lower error are exactly those patterns which belong to the noun group NG. After only 100 patterns the network has recognized that the global error can be minimized significantly by focusing on the NG patterns since these patterns occur more frequently than for instance prepositional groups or verb groups. Therefore, at first the network has learned a constant mapping of all patterns to the noun group. That is, all patterns are classified as noun groups since this reduces the overall error most at this stage. This explains why certain patterns in figure 3 still show a high error and others a low error. Those are exactly the patterns which have been classified correctly as noun groups.

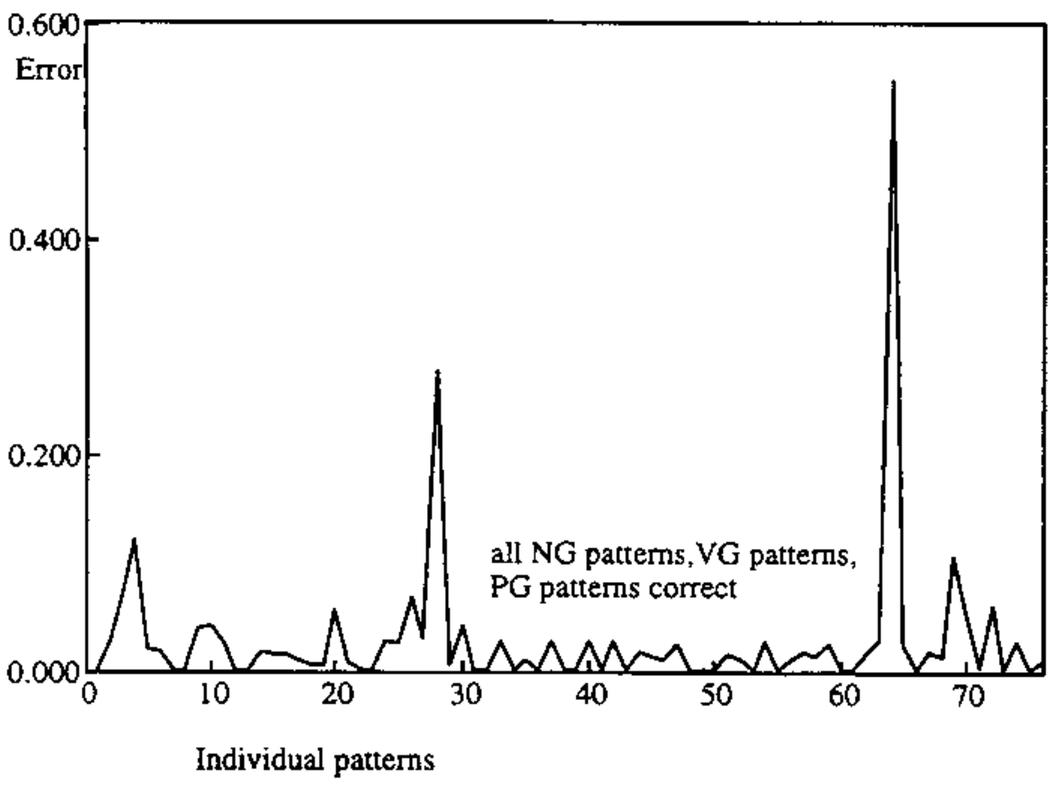

Figure 4: Performance for individual patterns after 150000 training patterns

After 150000 patterns all regularities have been learned as shown in figure 4. In general the network pursues a conservative learning strategy which we call *lazy learning*. First simple and often occurring generalizations of one category are learned. Only when the network cannot minimize its error significantly other often occurring categories are integrated. And only when all patterns have been learned which did not require previous local context those patterns are learned which require context to make correct category assignments for otherwise ambiguous category assignments. Finally certain exceptions are learned. During this conservative learning strategy it may be possible that the overall error increases briefly in order to reach a better overall state later.

Symbolic Interpretation of Neural Networks as Transducers

After we have analyzed how and why the network arrived at its certain performance, we now turn to the interpretation of the learned knowledge itself. A number of techniques like hierarchical cluster analysis of activation values or weight representations as Hinton diagrams have been used in the past to represent the knowledge of trained neural networks. However these techniques do not allow for a good representation of sequences of pattern assignments. Furthermore, they do not support a symbolic interpretation of the underlying knowledge. Therefore, we show a different technique of describing the knowledge within a recurrent network.

A symbolic transducer can be extracted from our recurrent network, which assigns to each input vector of basic syntactic categories a new output vector of phrasal categories depending on the previous context. In our network the internal state and the context was represented by a three-dimensional vector. For simplicity each strict symbolic interpretation of a three-dimensional vector can take 2^3, that is 8 states.

For a symbolic interpretation of the network we presented all patterns from the training set and stored the internal state vectors at the hidden layer of the network. Our demonstration network and the used data material was kept small enough in order to illustrate this process although in practice we have used much larger networks (e.g. several thousand patterns rather than 76 demonstration patterns). For each output vector and for each state vector the next corner preference was determined using the Euclidean distance metric. We assigned a symbolic abstract syntactic phrase category to each output vector and a symbolic number identifier to each state vector.

Figure 5 shows the learned knowledge of the network as an extracted strict symbolic transducer, sometimes also called a Moore machine. The corner nodes represent the eight strict states, the center node represents the start state of the transducer. At the edges we find the symbols for the single transductions. Input and output categories are separated by a colon, e.g. $d : ng$ means that - starting from the source state of this edge - a determiner preference d is assigned to a noun group preference ng and the transduction is made to the end state of this edge.

More detailed (less detailed) transducers can be provided if the state and output vectors are mapped to more (less) nodes. Therefore general abstraction level of such a symbolic transducer can be quite flexible. The symbolic transducer represents an abstraction of the detailed network knowledge but this abstraction also hides some of the numerical complexity and allows a symbolic direct interpretation which provides a summary of the network behavior.

000062

In the extracted transducer we can see some clear regularities at certain states. For instance the transductions to state 100 are primarily responsible for the assignments to the prepositional group (pg). Another example are the transductions to state 010 and to state 000, which are primarily responsible for the verbal group (vg) assignment. Furthermore, figure 5 shows the example transductions for the sentence "I thought in the next week". Beginning with the start state at the center we see the transduction $u : ng$ for the word "I" which assigns the noun group ng to the pronoun u. Then, $v : vg$ assigns a verb group vg to the verb "thought". Finally the transductions $r : pg$ $d : pg$ $j : pg$ $n : pg$ assign the prepositional group "pg" to the sequence "in the next week".

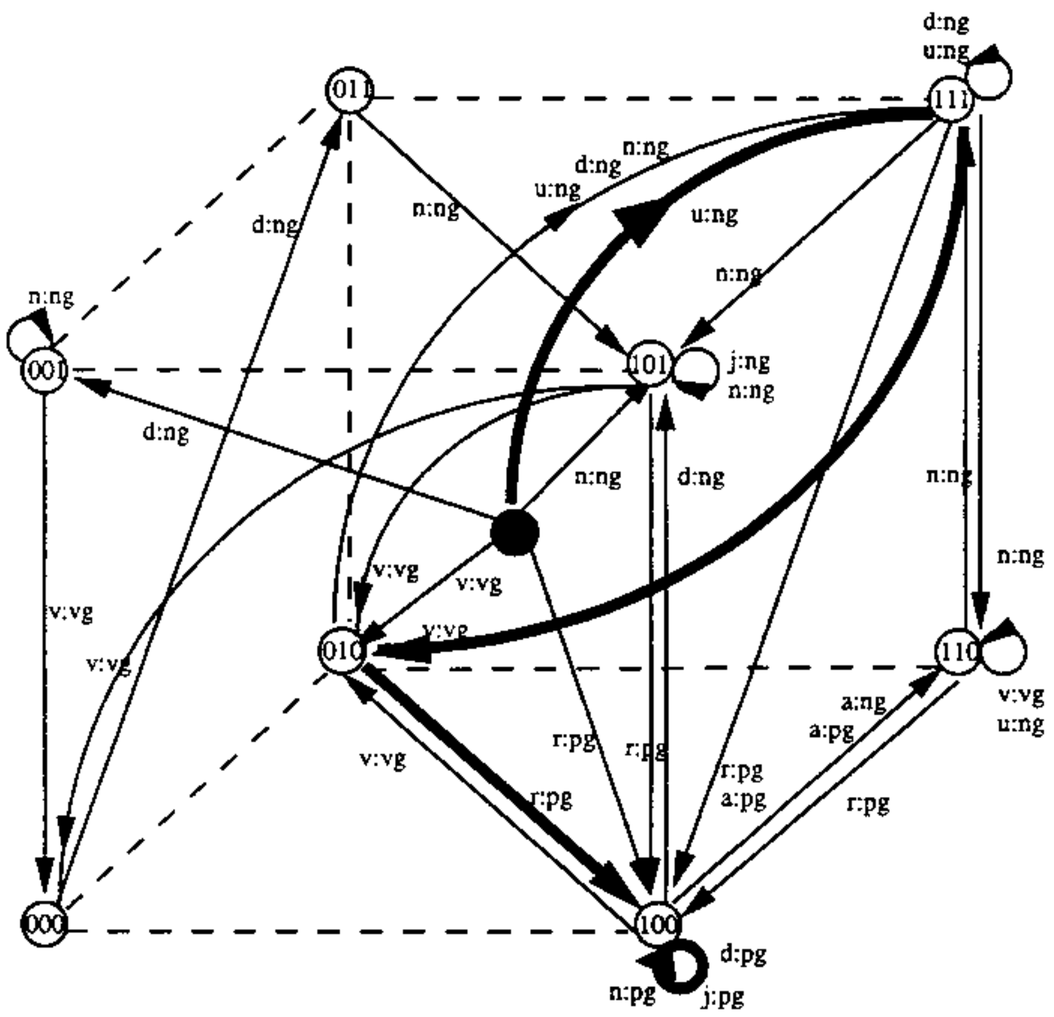

Figure 5: Symbolic interpretation of a recurrent network for the translated sentence "I thought in the next week".

In general there are no designated final states, since the network - and the extracted symbolic transducer - produce output as long as input is provided. This transducer behavior is therefore quite different from other extraction procedures [2], which are based on acceptors for artificial languages.

One advantage of this symbolic interpretation is the higher abstraction level for the recurrent network which makes it easier to understand. The original network contains more detailed knowledge in the numerical weights and activations, but it is not possible to see the declarative sequential symbolic knowledge which this network represents. The extraction of a symbolic transducer allows a better understanding of the learned sequential knowledge in a more explicit manner.

Conclusion

We have described new dynamic methods for the interpretation of recurrent neural networks. For building larger models of spoken language processors we believe it is essential to understand better the process of learning in recurrent networks. We have demonstrated that networks which have been used in large real world architectures of spoken language analysis show a conservative learning strategy which also has been observed in different other tasks in human performance. After learning, this "conservative learning" strategy led to neural representations which can be described as symbolic transducers. Furthermore, these transducers allow for a much better interpretation of the sequential network knowledge compared to common analysis using hierarchical clustering or Hinton diagrams. Finally this better symbolic interpretation of a neural network also holds a lot of potential for models which could bridge the large gap between neural representations in the brain and symbolic reasoning and processing in human behavior.

References

[1] J. L. Elman. Language as a dynamical system. In R. F. Port and T. van Gelder, editors, *Mind as motion: explorations in the dynamics of cognition*, pages 195–225. MIT, Cambridge, MA, 1995.

[2] C. Lee Giles and C. W. Omlin. Extraction, insertion and refinement of symbolic rules in dynamically driven recurrent neural networks. *Connection Science*, 5:307–337, 1993.

[3] C. W. Omlin and C. L. Giles. Extraction of rules from discrete-time recurrent neural networks. *Neural Networks*, 9(1):41–52, 1996.

[4] S. Wermter and M. Meurer. Building lexical representations dynamically using artificial neural networks. In *Proceedings of the International Conference of the Cognitive Science Society*, pages 802–807, Stanford, 1997.

[5] S. Wermter and V. Weber. SCREEN: Learning a flat syntactic and semantic spoken language analysis using artificial neural networks. *Journal of Artificial Intelligence Research*, 6(1):35–85, 1997.

[6] J. Wiles and J. Elman. Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. In *Proceedings of the AAAI Workshop on Computational Cognitive Modeling: Source of the Power*, Portland, Oregon, 1996.

000063