The Processing & Recognition of Symbol Sequences

Mark W. Andrews (mwa1@cornell.edu)
Department of Psychology; Uris Hall
Ithaca, NY 14853 USA

Abstract

It is proposed that learning a language (or more generally, a sequence of symbols) is formally equivalent to reconstructing the state-space of a non-linear dynamical system. Given this, a set of results from the study of nonlinear dynamical systems may be especially relevant for an understanding of the mechanisms underlying language processing. These results demonstrate that a dynamical system can be reconstructed on the basis of the data that it emits. They imply that with minimal assumptions the structure of an arbitrary language can be inferred entirely from a corpus of data. *State-Space reconstruction* can implemented in a straightforward manner in a model neural system. Simulations of a recurrent neural network, trained on a large corpus of natural language, are described. Results imply that the network sucessfully recognizes temporal patterns in this corpus.

Introduction

Complex pattern recognition is often characterized by means of a simple geometric analogy. Any object or pattern may be described as a single point in a high-dimensional space. For example, a square grayscale image that is 256 pixels in length, may be described as a point in the 256^2 dimensional space of all possible images. A collection of such images is a set of points in this space. If these patterns are not entirely random, this set will reside in a subspace of lower dimensionality. To learn the structure of these images, an organism or machine must discover a compact parametric representation of this subspace. This might take the form of, for example, finding a reasonably small set of basis vectors that will span the subspace and projecting each image onto these vectors. Having done this, each image can be classified in terms of a new and more meaningful coordinate system. You effectively describe 'what is there' in terms of 'what is known'.

This geometric approach is routinely employed in the study of visual object recognition, but may easily be extended to a wide range of categorization and classification tasks. In almost all cases, however, the patterns under study have been multi-dimensional *static* patterns. In contrast, the study of *temporal* pattern recognition using this or related approaches has not been well-developed. For example, one of the most widely employed techniques for temporal pattern recognition, Hidden Markov Models are limited in their generality due to their fundamental inability to handle patterns above a certain complexity. This absence of general models for temporal pattern recognition is evident in the study of human language processing, which traditionally has eschewed serious consideration of statistical learning and pattern recognition.

This paper aims to introduce a general framework for the study of temporal pattern recognition. This is developed in the context to language processing, but it could be extended in a straightforward manner to most other cases of temporal patterns. First, a general characterization of the problem of language learning and language processingis proposed. Then, some recent results in the study of nonlinear dynamical systems are described. These are seen as being especially relevant for an understanding of the mechanisms underlying temporal pattern recognition, especially with regard to language processing. Finally, simulations with a recurrent neural network are described, which suggest successful pattern recognition of English sentences.

The processing of symbol sequences

A paradigm for the study temporal pattern processing, especially language processing, has developed as a result of the deep relationship between formal languages and abstract automata (Chomsky 1963)[1]. Any language (or more generally, any sequence of symbols), can be described as the product of a particular automaton. By this account, learning a language is equivalent to identifying a particular automaton on the basis of a sample of the language that it generates. More formally, an automaton A is specified by the quadruple $\langle X, Y, F, G \rangle$. X and Y are sets known as the state and the output spaces, respectively. The functions $F: X \mapsto X$ and $G: X \mapsto Y$

---

[1] The correspondence between formal languages and abstract automata can be summarized by the so-called Chomsky hierarchy: Classes of automata that are increasingly restrictive versions of the Turing machine produce classes of languages described by increasingly restrictive generative grammars. The *regular* languages R are produced by strictly finite automata, the *context-free* languages CF are produced by pushdown stack automata, the *context-sensitive* languages CS are produced by linear bounded automata and the *recursively enumerable* languages RN are produced by unrestricted Turing machines. $R \subset CF \subset CS \subset RN$, and likewise for their corresponding automata.

are the state-transition and the output functions, respectively. Beginning at time t_0 and continuing until t_∞, the sentence-generator A constantly changes from one state in X to the next, according to its state-transition function F. At each transition, a symbol from the set Y is emitted, according to its output function G.

A langage learner attempts to identify the nature of this automaton on the basis of a sample of the language that it generates. That is to say, the language learner is exposed to a finite sequence of Y and from this must attempt to identify $A = \langle X, Y, F, G \rangle$. Having attained knowledge of A, the learner is said to have full knowledge of the structure of the language. The learner has the capability to produce all the sentences of the language, including the infinite number of sentences that were never seen. Likewise, the learner has the capability to parse the syntactic form of any sentence of the language. This ability also extends to the infinite number of never-encountered sentences. As syntactic parsing is a necessary precondition for the interpretation of language, it is said that the language has been learned once knowledge of its grammar has been attained.

While the correspondence between formal languages and automata has allowed the problem of language learning to be given an explicit characterization, these automata $A = \langle X, Y, F, G \rangle$ have always been taken to be discrete parameter systems. To continue this paradigm it is useful to demonstrate the correspondence between generative grammars and continuous as well as discrete automata. Within nonlinear dynamical systems theory, the study of *symbolic dynamics* has made apparent the relationships between formal languages, generative grammars and continuous dynamical systems. Symbolic dynamics refers to the practice of coarse-coding the ambient state-space of a dynamical system into a finite set of subspaces and assigning a symbol to each. Whenever the system enters a partition, the assigned symbol is emitted. In this way, the trajectories of the dynamical system can be represented as strings of symbols. Unless the system is entirely stochastic, only a certain subset of strings will occur. It can be shown that these strings define a language and the system producing them can be described by a generative grammar (Bai-Lin & Wei-Mou 1998).

The relationship between languages, grammars and dynamical systems has been further described by Tabor (1998). In that work, and in Tabor (2000), the computational capacities of a pushdown stack automaton were identified with those of a stochastic dynamical system, based on an *iterated function system*. This was used to demonstrate the recognition of context-free languages by a simple $2-$dimensional dynamical system. Following Tabor's approach, it is reasonable to propose that any language (or any symbol sequence) generating process may be legitimately described as a *continuous* as well as a *discrete* system. Accordingly, and by keeping a strict analogy with the automaton $A = \langle X, Y, F, G \rangle$ described above, it is possible to introduce the corresponding continuous system, A' defined by the quadruple $\langle \vec{x}, y, f, g \rangle$.

By introducing $A' = \langle \vec{x}, y, f, g \rangle$, the language generating process is being explicitly defined as a nonlinear dynamical system. For example, the system may be described by a set of coupled differential equations

$$\dot{x}_i = f(\vec{x}, \delta)$$

where $\vec{x}$ is the system's state and $\dot{x}_i$ is a vector-field defined on an m-dimensional manifold M. δ is a unspecified stochastic element in the system. The *language* being produced by this system is a result of the coarse-coding function

$$y = g(\vec{x}),$$

where y is a variable representing the *symbols* of the language. However, there are still formal similarities between the discrete automaton $A = \langle X, Y, F, G \rangle$ and its continuous counterpart $A' = \langle \vec{x}, y, f, g \rangle$. $\vec{x}$ is the state-space of A' and y is a variable representing its output. The function $f : \vec{x} \mapsto \vec{x}$ describes the state evolution of the system while $g : \vec{x} \mapsto y$ is an output function. In fact, the only essential distinction between $A = \langle X, Y, F, G \rangle$ and $\mathcal{A} = \langle \vec{x}, y, f, g \rangle$ is that in the latter case the state-space $\vec{x}$ is continuous, rather than discrete, and the evolution of the system described by f is smooth, preventing discontinuous leaps through space.

State-Space Reconstruction

A language learner can be said to be attempting to identify the process generating the language. If this generating process is described as a continuous dynamical system, the objective is to model the dynamical system $\dot{x} = f(\vec{x})$ on the basis of the language it outputs, $y_{t_0}, \ldots, y_t$. Prima facie, this problem is widely intractable. The symbols to which the learner is exposed do not identify the state of the system. They are a product of the composition of two unknown and probably non-linear functions, f and g. However, it may be fruitful to consider the analogy between this problem and a more general problem encountered in the experimental analysis of complex systems. For example, a scientist observing a sequence of individual measurements from a complex physical process (e.g. a fluid in turbulent motion) may be interested in understanding the properties of the underlying system. In the absence of prior knowledge and without loss in generality, the system can be taken to be a stochastic dynamical system, whose functional form is completely unknown. The scientist must infer its functional form on the basis of the measurement data alone. One of the more remarkable outcomes of dynamical systems theory is that in many general cases this problem is tractable. In virtue of the analogy, the manner by which this is done may also elucidate the problem of language learning.

Packard, Crutchfield, Farmer & Shaw (1980) were first to demonstrate that a dynamical system could be reconstructed entirely on the basis of its output. They proposed that *any* time-series of quantities measured from a dynamical system may be sufficient to construct a model

that preserves its essential structure. Takens (1981) developed and clarified the mathematical evidence for this proposal. This was considerably generalized by Sauer, Yorke & Casdagli (1991), and more recently Stark, Broomhead, Davies & Huke (1997) have extended these results to the more general case of stochastic dynamical systems.

Sauer et al. (1991) have suggested that the foundations of these ideas are to be found in differential topology. For example, a seminal theorem in this field (Whitney 1936) is that any m-dimensional manifold M can be mapped by a diffeomorphism[2] into Euclidean space $\mathbb{R}^d$ if $d > 2m + 1$. Moreover, the subset of all possible smooth maps from M to $\mathbb{R}^{2m+1}$ that are also diffeomorphisms is both open and dense in the function space. As Sauer et al. (1991) point out a single measurement of a dynamical system is a map from the system's state to the real line. As such, the significance of Whitney's result is that *almost every*[3] set of $2m + 1$ independent measurements of a dynamical taken simultaneously is suffcient to reconstruct the dynamical system in the measurement-space. The manifold M and its vector-field $\dot{x}$ are *embedded* in the measurement-space.

The more recent result by Takens (1981) may be understood in in terms of this embedding theorem. Takens considers the case of a dynamical system $f(\vec{x}, \delta) \colon M \mapsto M$ and the *delay-coordinate* map, $D \colon M \mapsto \mathbb{R}^{2m+1}$. This map D is defined as simply a time-series of scalar measurements $z = \{y_t, y_{t+1}, \ldots, y_{t+2m}\}$ obtained from this system, where $y = g(\vec{x})$. It is clear that

$$z = \{y_t, y_{t+1}, \ldots, y_{t+2m}\} = \{g(\vec{x}_t), g \circ f(\vec{x}_t), \ldots, g \circ f^{2m}(\vec{x}_t), \}$$

where f^n is the composition of f n-times. In other words, the sequence z of $2m + 1$ measurements $y = g(\vec{x})$ is in fact a function of a single point or state $\vec{x}$ of the hidden dynamical system. The *delay-coordinate* map D maps each state $\vec{x}$ of the hidden system to a point in $\mathbb{R}^{2m+1}$. Takens (1981) demonstrated that with minimal assumptions about the hidden dynamical system[4], the set of *delay-coordinate* maps D that are also diffeomorphisms is both open and dense in the space of maps D. In *almost every* case, the hidden dynamical system is *embedded* within the delay-coordinate measurement space.

Sauer et al. (1991) have considerably elaborated the Takens (1981) embedding theorem. They define both a

---

[2]A *diffeomorphism* from M to N is a one to one map, where the map and its inverse are diferentiable.

[3]The fact that the set of maps that are also diffeomorphisms is an *open* subset of the function space means that any arbitrarily small perturbation of a diffeomorphism is also a diffeomorphism. The fact that the set is *dense* means that every point in the function space is arbitrarily close to a diffeomorphism. In addition, Sauer et al. (1991) have shown that *almost every* map in the function space is a diffeomorphism, in that the complement to this subset is of measure zero. In other words, the likelihood of an arbitrary map also being a diffeomorphism is *probability one*, or infinitely likely.

[4]In particular, it is assumed that the dynamical does not contain periodic orbits that are exactly equal to (or exactly twice) the sampling rate of the measurement function $y = g(\vec{x})$.

delay coordinate map $D' \colon M \mapsto \mathbb{R}^s$, where s is an integer arbitrarily greater than $2m + 1$, and a smooth transformation of this map, $\phi \colon D' \mapsto \mathbb{R}^{2m+1}$. In the spirit of Takens (1981), Sauer et al. (1991) demonstrate that the set of these composite functions $\phi \circ D' \colon M \mapsto \mathbb{R}^{2m+1}$ that are also diffeomorphism is open and dense in the function space.

The theorems of Takens (1981) and Sauer et al. (1991) apply to deterministic dynamical systems. These are systems whose entire future evolution can be be determined from precise knowledge of the system's state. As real world systems are inevitably coupled with sources of external noise, the generality of these theorems may seem limited. Stark et al. (1997) have shown, however, that the embedding theorems can be generalized to a much less restricted class of stochastic dynamical systems. They consider a discrete time system where at each time step one of k different discrete-time maps $f_\omega \colon M \mapsto M$ is chosen, where $\omega = 1, \ldots, k$. As in Takens (1981), they define the *delay-coordinate* map, $D \colon M \mapsto \mathbb{R}^{2m+1}$ and show that in the stochastic systems under consideration the set of maps D that are also diffeomorphism is open and dense in the function space.

State-Space reconstruction in neural systems

While these results are obviously important for the general problem of nonlinear time-series analysis, their relevance for the problem of language learning may be limited. The problem of language learning does not fit neatly into the scenarios considered by Takens (1981), Sauer et al. (1991) and Stark et al. (1997). This is primarily due to the fact that the output of the language generating dynamical system is a sequence of symbols rather than a real-valued scalar. In addition, the stochastic system considered by Stark et al. (1997) might not be general enough to describe the arbitrary stochastic dynamical system that is here taken to be the language generator. More importantly, these theorems consider and explain certain *sufficient* conditions and do not lead naturally to a general algorithmic procedure for reconstructing state-space. For example, Taken's theorem demonstrates that the coordinate space of $2m + 1$ scalar measurements is sufficient to embed the generating dynamical system of dimensionality m. Practically, however, this just means that the coordinate space of a *finite* number of scalar measurements is sufficient for the embedding. It does not indicate how it can be known that an embedding has in fact occurred. What is necessary, therefore, is an *objective function* that may be optimized to produce a reconstruction of the dynamical systems from its outputs.

Crutchfield & Young (1989) introduce ε-machines as a general procedure for state-space reconstruction. They propose that the state of the ε-machine uniquely corresponds to the state of a dynamical system emiting a symbol sequence if it can be shown that its state renders the future of the symbol sequence conditionally independent of its past. In other words, if the probability distribu-

tion over future sequences of symbols is independent of the past symbols *given* the state of the ε-machine, then the ε-machine uniquely labels the state of the dynamical system generating the symbols. The ε-machine can then be taken as a model of the unseen symbol generating dynamical system[5].

On the basis of the embedding theorems and the ε-machine of Crutchfield & Young (1989), an objective function for state-space reconstruction may be introduced. The objective is to *model* the dynamical system $f(\vec{x}, \delta): M \mapsto M$, and this can be defined as learning a structure-preserving map from the manifold M to a second topological *model-space* N. If the probability distribution over sequences of symbols emitted by the dynamical system defined on M is independent of its past symbols *given* the state of the *model-space* N, then N smoothly and uniquely labels the state of the dynamical system generating the symbols. The trajectory of states on N can then be taken as a model of the unseen symbol generating dynamical system N. This idea may be illustrated by means of a neural system.

A system of cortical neurons can be minimally modeled by a set of n coupled nonlinear differential equations,

$$\dot{y}_i = -y_i + \sum_{j=1}^{j=n} w_{ij}\sigma(y_i) + I_i,$$

where σ is a smooth and monotonic transfer function, y_i is the soma potential of neuron i, resulting from a weighted sum of its inhibitory and excitatory inputs. I is the external input to the system. Clearly, this system is a dynamical system defined on a n-dimensional manifold N. In addition, the state of this system $\vec{y}_t$ at a given time t is a function of both its present input I_t and, through the action of its recurrent synapses, the history of previous input, $\{I_{t_0}, \dots, I_t\}$. In other words, the system's state at any given time is a smooth function of an entire sequence of inputs. This can be represented by the correspondence $\vec{y}_t = \psi(I_{t_0}, \dots, I_t)$. If the sequence of inputs $\{I_{t_0}, \dots, I_t\}$ represents the output $I_t = g(\vec{x}_t)$ of dynamical

system $f(\vec{x}, \delta)$ then it is clear that

$$\vec{y}_t = \psi(I_{t_0}, \dots, I_t) = \psi\left(g(\vec{x}_{t_0}), g \circ f(\vec{x}_{t_0}), \dots, g \circ f^t(\vec{x}_{t_0})\right),$$

where f^t is the composition of f t times. The state $\vec{y}$ of the neural system is a function of the state $\vec{x}$ of the hidden dynamical system.

The neural system $\dot{y}_i$ on N is a diffeomorphism of the dynamical system $\dot{x}$ on M, if the state $\vec{y}$ smoothly and uniquely labels the state $\vec{x}$. If the future inputs to the neural system are stochastically independent of the past inputs, *given* the state $\vec{y}$ of the system then N and M are diffeomorphically equivalent. If the probability of the future inputs to the neural system, conditioned on its state $\vec{y}$, is not further sharpened by acquiring information about the previous inputs to the system then there is a structure preserving map between the two systems.

Network simulations

In this paper, it is taken that a language (or a sequence of symbols) is produced by a continuous dynamical system. To learn this dynamical is to learn the statistical structure of the language. By hypothesis, this can be accomplished by embedding the hidden dynamical system in a second model space. To maximize prediction of future states given present ones is effectively to seek such an embedding. As such, it should be the case that if a recurrent neural network is trained on a corpus of natural language (in the now familiar style introduced initially by Elman (1990)) it should develop a state space that is a model of the generating process of the language. One manifestation of this would be that sentences, judged (by human observers) to be structurally similar, should also be clustered in the state space of the neural system.

To explore this hypothesis further, a simulation of an idealized neural system was performed by implementing the system of coupled equations,

$$\dot{y}_i = -y_i + \sum_j w_{ij}\sigma(y_i) + \theta_i + \sum_k w_{ik}I_k,$$

$$O_l = \sigma\left(\sum_i w_{li}(y_i)\right),$$

$$\sigma(\zeta) = \left(1 + e^{-\zeta}\right)^{-1},$$

where y_i is the state of the neuron and can be viewed as representing its mean soma potential, θ_i is a bias term and I_i is external input. O_l is the output of the system which "reads off" the recurrent network. There were 120 neurons in the recurrent network. The input was a 250 dimensional bit vector, described below. The output was likewise a 250 dimensional vector. For the purposes of computer simulation, a difference equation was used,

$$y_i^{t+\Delta t} = (1 - \Delta t)y_i^t + \Delta t \sum_j w_{ij}\sigma(y_i) + \Delta t\theta_i^t + \Delta t \sum_k w_{ik}I_k^t,$$

---

[5]In a dynamical system, the entire evolution of the system is described by its trajectory from $t_{-\infty}$ through t_0 to t_∞. The future trajectories of the system are conditionally independent of the past, *given* the present state of the system. In the ideal case of a deterministic and autonomous system, the future trajectory of the system, $X[t_0, t_\infty)$, can in principle be determined from the present state of the system, $X(t_0)$. Absolute knowledge of the system's state X at t_0 provides absolute knowledge of the future trajectory $X[t_0, t_\infty)$. No information about the system's prior trajectory $X(t_{-\infty}, t_0]$ is necessary. In a stochastic dynamical system (where, for example, at irregular points in time there is coin toss of an k-sided coin to choose between k different set of differential equations), a similar situation occurs. While the future is not entirely predictable on the basis of the present state in this system, no *increase* in information about the future is gained by knowing the past. In other words, the future trajectory of the system is stochastically independent of the past, *given* the state of the system. The case of a stochastic system can be seen to generalize to the case of a dynamical system driven by external input.

This was obtained by an approximation of its continuous counterpart. Δt was a variable parameter which could be manipulated for finer approximations of the underlying continuous system.

The data-set used for network learning was a corpus of natural language amounting to over 10 million words. The corpus comprised 14,000 documents, the average length of each document being approximately 700 words. All documents were in a plain-text and untagged format. They were obtained from publicly available electronic text archives on the internet[6]. No explicit criteria were used when selecting documents other than that cover a wide range of subject matters such as science, social science, literature, children's stories, history, law and politics.

Altogether, the entire corpus contained a vocabulary of 115,000 words. Of these, a set of 50,000 accounted for over 99% of the total number of words in the corpus. Only the members of this set were used for training the network, the infrequent words having been deleted. Each of these 50,000 words was coded by being randomly assigned to a unique bit vector of 247 zeros and 3 ones (there are over 2.5 million possible combinations to choose from). While this random coding scheme introduced some spurious correlations between words, the average correlation between words was close to zero[7].

The network was presented with the entire corpus as a sequence of words, one word at a time. The network was trained to predict its future word-input given its present word-input. The synaptic weight parameters were adapted using the continuous version of back-propagation through time due to Pearlmutter (1989). In this procedure, the minimum of the cross-entropy objective function was sought by calculating the derivatives of this function with respect to each weight parameter at each time "tick" Δt of the 50 previous time steps.

With a learning rate parameter of .01, and a Δt parameter of .25, the network was trained for 46 passes through the corpus. At this time, the learning rate parameter was annealed to .001, and the Δt parameter was lowered to .1. Training was continued for another five passes through the entire corpus. The performance of the network at predicting future words could be adequately assessed using the a method of ratios between squared errors,

$$R_i = \frac{\sum_t \left(d_i^t - y_i^t\right)^2}{\sum_t \left(d_i^t - d_i^{t-1}\right)^2},$$

where d_i^t is the target or to-be-predicted outcome for neuron i at time t. The denominator of this ratio specifies the sum squared differences between the target outcome and the target at the previous step. This ratio is useful as the best prediction a random-walk model can make would

be to predict the same value for the future as is obtained at the present. Thus, if the ratio is greater than 1.0 the network is performing worse than a chance model. At values less the 1.0, the network is performing better than a chance model. A value approaching 0, would indicate perfect predictive accuracy.

On the final pass through the corpus, the mean performance ratio for the training data was .4767. Furthermore, a validation set which comprised 1000 unseen documents was prepared. The mean performance ratio on this set was .4989. These values indicate substantial predictive performance and generalization abilities by the network. They compare very favorably to mean performance ratios usually obtained in non-linear time series prediction tasks (Weigend & Gershenfeld 1993).

Discriminant function analysis

If a neural network learns the statistical structure of the language , its state space should have topological organization based on a similarity principle. For example, sentences that are similar in content should cluster in compact neighborhoods of the state space. An ideal experimental test of this would be to have reliable human judges classify a large set of sentences on the basis of their content, and then to compare this with a network's classification of the same set of sentences. To the extent that the network's classifications are close to those of human judges, the network would have met a behavioral criterion for language comprehension.

To adequately assess generalization abilities, a large set of sentences would be required. Such an experiment would be laborious to conduct. Fortunately, however, data-sets of labeled or categorized documents (rather than sentences) are readily obtainable, as these are regularly used as benchmark tests of text categorization techniques. In the experiment conducted here, sentences were extracted from labeled documents. Sentences were then assigned to the semantic class of the document from which they came. For example, sentences taken from a document assigned to the class 'motorcycling' would themselves be assigned to the semantic class 'motorcycling'. In this way, a large set of sentences could assigned a plausible, although somewhat limited, interpretation. The data-sets were the Reuters-21578 newswire data-set, the 20 newsgroups data-set[8] , and then a third set which was compiled for the purpose of this experiment from 6000 documents obtained from the Library of Congress, which had been previously classified by their Dewey Decimal categories

An appropriate test of the network's representational capacities would be to assess the probability that a sentence from a given semantic class would be assigned correctly to that class. To do this a linear discriminant function was used to divide the state space into (simply connected and convex) sub-regions based on semantic class. The discriminant function is a straightforward

---

[6]The main sources of the electronic texts were, Project Gutenberg, the Etext Archives, and archives.org.

[7]A more valid distributed code based the actual orthography of English words has been used by the author in previous simulations, but these will be reported here.

[8]The two data sets are available on the internet. See http://www.cs.cmu.edu/ textlearning and http://www.research.att.com/ lewis

Table 1: Accuracy of sentence classification.

Data Set	Accuracy
Library	83%
Reuters	75%
Newsgroups	69%
Mean	76%

linear transformation of the state space, such that the centroids of "training" sentences labeled by their class are made maximally distant from one another. The network's ability to categorize by semantic class can be assessed for a "test" set of sentences by assessing the probability that a given sentence from a certain semantic class would be correctly assigned to that class. The measure used was Mahalanobis distance. This measure is approximately proportional to an estimate of the posterior probability that a given sentence will correctly assigned to its appropriate class. 5000 sentences from each of the three data-set were used in this test. The results are illustrated in Table 1.

These accuracy rates are suitably high, and in fact compare favorably to state-of-the-art categorization methods which use similar or identical data-sets (Nigam, Mccallum, Thrun & Mitchell 2000). It is reasonable to conclude from this that the state space of a recurrent neural network trained to predict word sequences becomes organized on basis on semantic similarity. Sentences and texts that are semantically similar are clustered into compact neighborhoods which can be discriminated by a simple linear function.

Conclusion

Temporal pattern recognition is not as theoretically sophisticated as its multidimensional and static counterpart. Here, an approach to temporal pattern learning is introduced that is based on recent results from dynamical systems theory. It is proposed that the reconstruction of system generating a language (or symbol sequence) is adequate for learning the statistical structure of temporal data. It is proposed that state-space reconstruction can be carried out in a straightforward manner in a recurrent neural network. Results showing pattern recognition of English sentences by the network are provided. These results are similar in kind to those obtained by Elman (1990) and in the many works that followed this paradigm. It is believed that the appropriate explanation of these now familar sets of results, is that the recurrent neural network has reconstructed the language generating process. Sentences that were produced by similar trajectories in the original systems are now modelled by similar trajectories in the recurrent neural network. It is clear, however, that this is not a definitive demonstration of state-space reconstruction and a more detailed analysis of temporal pattern learning using formal grammars is being currently undertaken (Andrews 2001).

References

Andrews, M. W. (2001), Language learning by state space reconstruction. Manuscript in preparation.

Bai-Lin, H. & Wei-Mou, Z. (1998), *Applied Symbolic Dynamics and Chaos*, World Scientific, Singapore.

Chomsky, N. (1963), Formal properties of grammars, *in* R. D. Luce, R. R. Bush & E. Galanter, eds, 'Handbook of mathematical psychology', Vol. 2, John Wiley and Sons, Inc., New York and London, pp. 323–418.

Crutchfield, J. P. & Young, K. (1989), 'Inferring statistical complexity', *Physical Review Letters* **63**(2), 105–108.

Elman, J. L. (1990), 'Finding structure in time', *Cognitive Science* **14**, 179–211.

Nigam, K., Mccallum, A., Thrun, S. & Mitchell, T. (2000), 'Text classification from labeled and unlabeled documents using em', *Machine Learning* **39**(2/3), 103–135.

Packard, N., Crutchfield, J. P., Farmer, J. & Shaw, R. (1980), 'Geometry from a time series', *Physical Review Letters* **45**(9), 712–716.

Pearlmutter, B. (1989), 'Learning state space trajectories in recurrent neural networks', *Neural Computation* **1**, 263–269.

Sauer, T., Yorke, J. A. & Casdagli, M. (1991), 'Embedology', *Journal of Statistical Physics* **65**(3–4), 579–616.

Stark, J., Broomhead, D. S., Davies, M. E. & Huke, J. (1997), 'Takens embedding theorem for forced and stochastic systems', *Nonlinear Analysis, Theory, Methods and Applications* **30**(8), 5303–5314.

Tabor, W. (1998), Dynamical automata, Technical report, Department of Computer Science, Cornell University.

Tabor, W. (2000), 'Fractal encoding of context-free grammars in connectionist networks', *Expert Systems* **17**(1), 41–56.

Takens, F. (1981), Detecting strange attractors in turbulence, *in* D. Rand & L.-S. Young, eds, 'Dynamical Systems and Turbulence', Springer-Verlag, Berlin and Heidelberg, pp. 366–381.

Weigend, A. S. & Gershenfeld, N. A., eds (1993), *Time series prediction: Forecasting the future and understanding the past*, Vol. 15, Addison-Wesley, Reading, MA 01867.

Whitney, H. (1936), 'Differentiable manifolds', *Annals of Mathematics* **37**(3), 645–680.