A Model of Symbol Grounding in a Temporal Environment

Brian T. Bartell Garrison W. Cottrell

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, Ca. 92093-0114

Abstract

Recent work by researchers [Harnad 90], [Cottrell, et. al. 90] has focused attention on the Symbol Grounding Problem, which can be paraphrased as follows: if the symbols in a symbol processing system have no computable relationship with the objects or constructs they are to denote in the world, how can the symbols have a non-trivial semantics?

We present a recurrent neural network model which learns to generate symbolic categorizations of the temporal characteristics of its stimulus environment. The Symbol Grounding Problem is addressed by relating the learned categories directly to the perceptual input, and by analyzing the representation space constructed by the network to perform the task. We demonstrate that such a grounded system can exhibit useful generalization, and that the internal representation of the symbolic classes is usefully different than the traditional predicate logic approach.

1 Introduction

We wish to investigate how a neural network can learn symbolic classifications of data with strongly temporal features, and how these classifications relate to traditional symbolic approaches to class membership. By grounding the system in an analog environment, a semantics can be ascribed to the learned symbolic classifications. This enables us to analyze the representational system with direct reference to the network's environment. A relevant contrast which we will draw exists between the all-or-nothing nature of a logic predicate P, which segments the world into all things $P(x)$ and all things $\neg P(x)$, and the graded and textured potential of PDP representations.

This paper presents a simple recurrent neural network (SRN) which was trained to generate sequences of symbols (which may be interpreted as words from a very simple lexicon) classifying sequences of perceptual input originating from an environment. The symbols generated, although from a small set, constitute classifications of an environment which is both analog valued and which has strongly temporal features. Therefore, the network must correctly learn the temporal regularities in order to successfully generalize to novel sequences from the same continuous space.

2 Task

The stimulus environment is defined by a (possibly infinite) set of movies presentable to the network, each of which consists of a sequence of visual images in a retinotopic (or some alternate more abstract) representation. Each image is a static snap-shot of the world. At each discrete time step, a single image from the current active movie sequence is presented to the network. Only one movie is active at a time. The particular environment used in the current experiments involves movies in which a (billiard) ball rolls around a square table using a starting position and velocity randomly determined for each movie, and bounces off the table's walled edges. A single movie consists of 20 snap-shots of the ball in successive positions on the table. An image is presented to the network using 2 nodes, representing the ball's $< x, y >$ position. The table walls are located at ± 0.8 along both the x and y axes. Velocities δx and δy are randomly chosen in the range $[-0.3, +0.3]$.

0-7803-0164-1/91/0000-0805$01.00©1991 IEEE

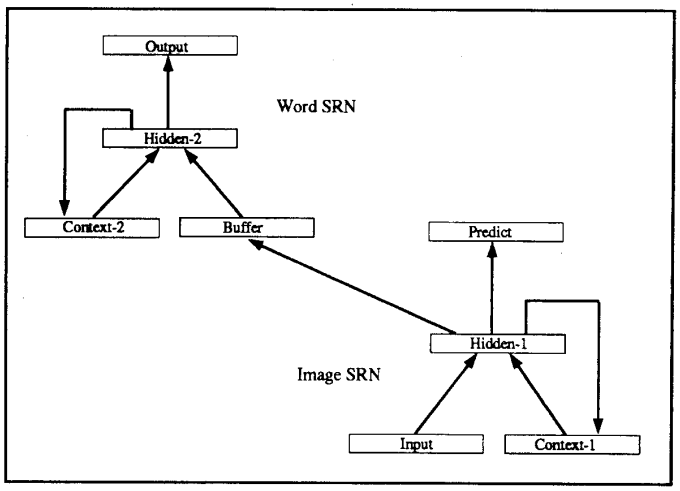

Figure 1: The Movie Description Network (MDN) architecture, described in the text.

Descriptions of the environment are sequential enumerations of temporal features of the movies. In the "billiard ball" world, descriptions are sequences of the form: "rolling { up | down } and { right | left } { slowly | quickly } period", where the correct choice from each pair is instantiated deterministically based on the current trajectory of the ball. One symbol is generated at the Output layer each time step. Note that the net must learn non-trivial temporal features of the image sequences (e.g. relative rate of motion) in order to generate an accurate description, and that these features are not present in any single image. Additionally, since the perceptual input is rational valued, the network cannot simply memorize a finite corpus – an important distinguishing feature between this work and the work of others in the field (most notably [Allen 90]).

Each word is encoded using a local representation across the Output units. Thus, the representation for each word has a single unique node with a value +0.8, and all other nodes with a -0.8 value. During word generation, the unit with the highest activation value determines which word is output by the network.

3 Recurrent Neural Network Model

The network architecture is depicted in Figure 1. Because of the task which the network attempts to solve, it is called the Movie Description Network (MDN). In the figure, each labeled rectangle represents a layer of typical connectionist processing units.[1] Arrows pointing generally upward represent uni-directional weighted links which fully connect the source and target layers to propagate activation values, and which are trained using non-temporal back-propagation [Rumelhart 86]. Arrows pointing downward are one-to-one copy links. The MDN architecture is motivated by the work of previous researchers (e.g. [Elman 88] [Allen 90] [StJohn 90]) in connectionist natural language processing.

The network operates as follows: a visual snap-shot of the world is presented to the net at the Input units, and the lower portion of the network (the Image SRN) is trained to predict the next visual state (similar to Elman's word prediction task [Elman 88]); at random intervals, the upper portion of the net (the Word SRN) is trained to generate the sequence of words which describes the current world based on the activations propagated forward to the Buffer layer. The Buffer

[1] Each node generates an output signal equivalent to a sigmoid, bounded between -1.0 and 1.0, computed on the inner product of the node's weight vector and input activations, plus bias.

Decision Type	Trained Range	Extended Movie	$\delta x, \delta y$ in $[-0.6, +0.6]$	$\delta x, \delta y$ in $[-0.9, +0.9]$
Up/Down	92%	94%	89%	81%
Right/Left	90	92	87	77
Quick/Slow	84	90	96	96
syntax	100	100	100	100

Table 1: Percent of classifications performed correctly, by decision type, for the trained $\delta x, \delta y$ range and on untrained ranges and movie sequence lengths.

activations remain fixed during the generation of a single description sequence. Word errors are propagated from the Output layer through to the Input layer. Although the primary task is to generate symbolic classification sequences, prediction training in the Word SRN is used to help constrain the information content in the Hidden-1 layer (similar to the "hints" used in [Wiles 90]).

4 Performance

The MDN was trained for 7 iterations through 50,000 randomly generated movies, using $\alpha = 0.9$ and $\eta = 0.001$ for 2 epochs, $\eta = 0.0005$ for the remaining 5. 20 units were used in each of the internal layers: Hidden-1, Buffer, and Hidden-2.

A summary of the classification performance of the network is provided in Table 1. Each column summarizes performance for a test consisting of 250 randomly generated movies presented to the network, with a single description extracted and analyzed. The "Extended Movie" test summarizes performance when the description is generated after 50 steps of the movie. The high correctness rate indicates that the network has generalized in time, since the network was only trained on length 20 movie sequences. All other tests sampled the description after 7 steps. The third and last columns in the table test for generalization to ball velocities outside of the trained $\delta x, \delta y$ range (i.e. in ± 0.3). Although performance degrades in these cases, it is gradual and follows the same pattern as for the trained movies. Figure 2 depicts locations of "left/right" word errors plotted in the ball's velocity space (for the case $\delta x, \delta y$ in $[-0.9, +0.9]$). Classification errors for the other decision types ("up/down" and "quickly/slowly") are similarly clustered around decision boundaries and show good generalization performance in the central regions of each class.

Note that only semantic class errors are made; the syntax of the description sequences was correct on every test. Classification performance can be improved by further training, although generalization performance for direction classifications degrades more rapidly.

5 Discussion

We wish to investigate the character of the trained network's internal representations, with specific emphasis on:

- visualizing the grounded symbols in terms of the network's environment, and
- contrasting the representational space used by the network with traditional binary symbolic classes.

Most of the analysis will consider the Hidden-1 layer, since this layer must encode all information about the environment for the network to perform the primary description and secondary prediction tasks successfully. The Buffer layer is also considered since it encodes class information for the description task.

The first, and most obvious, characteristic of the activations on these layers is that the values are not limited to a small range of values. Rather, the complete range is used. This is expected at the Hidden-1 layer, because of the continuous nature of its prediction task, but also is the case of

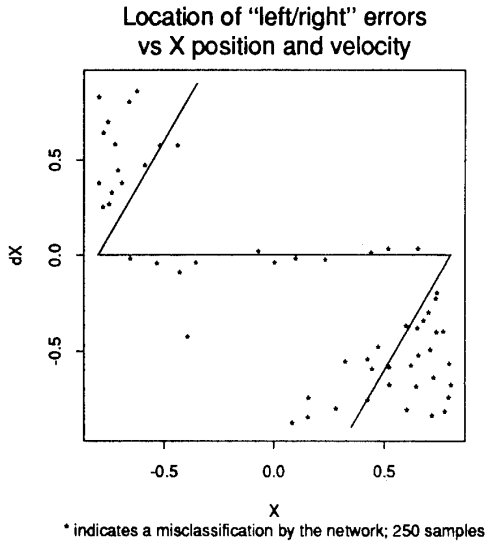

Figure 2: Classification errors occur almost exclusively around the decision boundaries, even during generalization to velocities (δx) larger than trained (± 0.3). The horizontal line at $\delta x = 0.0$ denotes the "left/right" boundary. Bouncing (where $x = \pm 0.8$) also separates the classes; diagonal lines indicate regions in which the ball will appear to move one direction (due to sampling the world in discretely timed images) but must be classified as moving in the opposite direction.

the Buffer layer. This precludes using a finite state framework for analyzing the recurrence, which is common in the literature [Sun, et. al. 90]. Information is also distributed at the Hidden-1 layer, with node activation values correlated with combinations of the environment parameters $x, y, \delta x$, and δy. However, the Buffer layer has learned a much more local, although still graded, information encoding, with nodes well correlated with single parameters δx and δy, and with one node correlated specifically with ball speed (a non-trivial feature).

One method for visualizing the grounded symbols is by sampling the network describing a set of movies, and then plotting the classification with respect to the environmental features present. Figure 3 (left) presents this result for the "slow/quick" discrimination. This can be thought of as an approximate extensional semantics for the two symbols, since we define the symbols (e.g. predicate $SLOW()$) with respect to a sample of the set of things (X) which are classified positively ($SLOW(X)$). Note that a single position in velocity space can be classified in two opposite classes at different times, because the recurrent context of the network will be different for different movies.

We may also push the representation further by asking which set of environment feature values is most typical for each class. One simple process for constructing prototypes for each of the symbols is to average the Hidden-1 activation vectors over a sample of vectors which all generated the same symbolic description. These prototype Hidden-1 vectors can then be placed in the Image SRN and propagated through the network to examine the network's next image prediction and to check for accurate symbol generation. This process was performed for all atomic symbols (up, down, right, left, quick, and slow) as well as for a composite: up and right. Symbol generation performance was correct in all cases except for "quick", which generated "slow" instead. Predicted motion for each of the prototypes is displayed in Figure 3 (right), including the up-right composite labeled as "up-right-1" ("up-right-2" is discussed below). Note that the prediction for "quick" is anything but; in fact, the true speed of this prototype is 0.058, well below the 0.240 decision boundary speed.

We may also examine the relationship of other sampled activation vectors to the prototypes. Table 2 lists mean distances (in 20D euclidean space) and standard deviations between the proto-

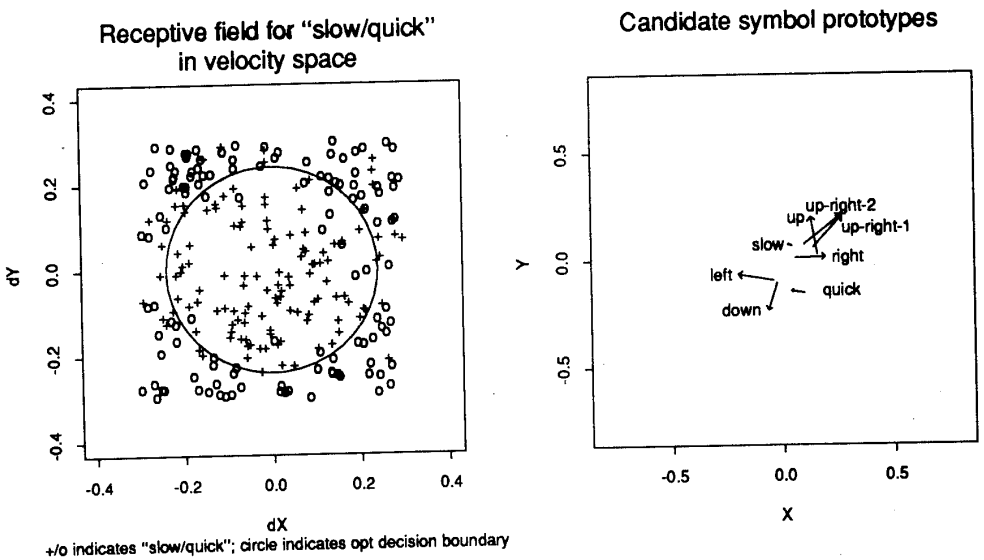

Figure 3: Two methods for visualizing the meaning of the network's symbols in terms of its grounding environment. The left graph depicts the receptive fields for the symbols "slow" and "quick", in terms of sampled ball velocities. The right graph presents possible prototypes for each of the classes, including two "up and right" compositions, described in the text.

	Up/Right	Right	Down/Right	Down	Down/Left	Left	Up/Left	Up
Mean	1.717	1.903	2.108	2.318	2.584	2.385	2.155	1.884
SDev	0.549	0.561	0.508	0.514	0.390	0.487	0.495	0.566

Table 2: Mean and standard deviation of euclidean distances between the prototypical "up and right" representation and other possible segmentations by class.

typical "up and right" vector and vectors classified by the network in other ways. The increase in distance as the sample vector class moves semantically farther from the prototype indicates that external similarity judgments between representations are possible using a simple euclidean distance metric. However, the large standard deviations suggest interference due to other free dimensions in the system (mainly x and y ball positions, which share the Hidden-1 representation space).

Another technique for constructing prototypes is to use the method of principle components analysis for decomposing a very large sample of data vectors (in our case, vectors of activation) into a set of orthonormal vectors spanning the data space. The spanning vectors are chosen by principle components such that the first component vector is along the axis of maximum variance in the data, the second along the second largest variance, etc. This decomposition can potentially localize the highly distributed representations in a set of hidden units (e.g. Hidden-1), and particular components can often be correlated with environment features (empirically). Thus, by letting a base PC vector $\vec{v}$ be equal to middle component values along each axis which we are uninterested in, and letting the component values of $\vec{v}$ which are correlated with particular features of interest in the environment take on non-mean values, we can calculate a prototype vector $\vec{p} = \vec{v} * R^T$, where R is the principle component rotation matrix, and T the transpose operator. This method was performed, with prototype "up and right" displayed as "up-right-2" in Figure 3.

These two methods for constructing prototypical symbol vectors, and the analysis of distances between representations, assume that the representational space used by the network is essentially euclidean in form. This is certainly explicit in the calculation of euclidean distance, but also is implicit in the averaging operation over representation vectors. By averaging, we assume that

symbol representations are clustered in this space. However, it is obvious that these assumptions are not valid in general, taking the calculated "quick" prototype as an example (see Figure 3). It is important to note that the non-linearities in the network (and the additional layers between the Hidden-1 layer and the Output layer) allow arbitrary encodings of the information at Hidden-1, and a convenient euclidean space is therefore not necessarily emergent.

6 Conclusion

The Movie Description Network is an architecture for learning a limited descriptive symbolic vocabulary for an analog temporal environment. The labels which the network learns in order to generate correct descriptions of the varied trajectories of a billiard ball appear functionally symbolic, similar to predicate logics; however, the internal representations are distributed and continuous. A Euclidean interpretation for the representation space was examined, and was found to offer insights in general, but was inadequate in at least one case. We have demonstrated that symbols can be grounded in a system's environment, and that such a grounded system is able to generalize to patterns outside its training set.

Acknowledgements: The authors wish to thank Jeff Elman, William Hart, Gregory Werner, and the members of GURU: Tim Ash, Steven Biafore, Dave Demers, Margaret Myers, Mai Nguyen, Mark Plutowski, Fu-Sheng Tsung, and Keiji Yamada, for useful discussions. Simulations were performed using a modified SunNet simulator [Miyata 87]. The primary author is supported by a Cubic Fellowship, and by Peregrine Systems, Inc., Carlsbad, Ca.

References

[Allen 90] Allen, R. B., *Connectionist Language Users.* Tech Report TR-AR-90-402. February 1990. Bell Communications Research.

[Cottrell, et. al. 90] Cottrell, G. W., Bartell, B. T., Haupt, C. *Grounding Meaning in Perception.* German Workshop on Artificial Intelligence (GWAI), 1990.

[Elman 88] Elman, J. L., *Finding Structure in Time.* CRL Technical Report 8801. April 1988. Center for Research in Language, San Diego.

[Sun, et. al. 90] Sun, G. Z., Chen, H. H., Lee, Y. C., and Giles, C. L. *Recurrent Neural Networks, Hidden Markov Models and Stochastic Grammars.* Vol I. IJCNN San Diego, June 1990.

[Harnad 90] Harnad, S., *The Symbol Grounding Problem.* Physica D 1990, Volume 42, Number 1-3, pp. 335-346.

[Miyata 87] Miyata, Y., *SunNet Version 5.2: A Tool for Constructing, Running, and Looking into a PDP Network in a Sun Graphics Window.* ICS Tech Report 8708. December 1987. Institute for Cognitive Science, San Diego.

[Rumelhart 86] Rumelhart, D. E., McClelland, J. L., and the PDP Research Group. Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol I. 1986. The MIT Press, Cambridge.

[StJohn 90] St. John, M. F., *The Story Gestalt: Text Comprehension by Cue-based Constraint Satisfaction.* Preprint, submitted to the 1990 Conference of the Cognitive Science Society.

[Wiles 90] Wiles, J. S., and Bloesch, A. *Patterns of Activation are Operators in Recurrent Networks.* Tech. Report No. 189. University of Queensland, Australia. October 1990.