1987

Scaling properties of coarse-coded symbol memories

Ronald Rosenfeld
Carnegie Mellon University

David S. Touretzky
Artificial Intelligence and Psychology Project.

Follow this and additional works at: http://repository.cmu.edu/compsci
NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making
of photocopies or other reproductions of copyrighted material. Any copying of this
document without permission of its author may be prohibited by law.
SCALING PROPERTIES OF COARSE-CODED SYMBOL MEMORIES

Technical Report AIP-10

Ronald Rosenfeld and David S. Touretzky

Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA 15213

29 September 1987

This research was supported by the Computer Sciences Division, Office of Naval Research and DARPA under Contract Number N00014-86-K-0678. Reproduction in whole or in part is permitted for purposes of the United States Government. Approved for public release; distribution unlimited.
Coarse coded memories have appeared in several neural network symbol processing models, such as Touretzky and Hinton’s distributed connectionist production system DCPS, Touretzky’s distributed implementation of Lisp S-expressions on a Boltzmann machine, and St. John and McClelland’s PDP model of case role defaults. In order to determine how these models would scale, one must first have some understanding of the mathematics of coarse coded representations. For example, the working memory of DCPS, which stores triples of symbols and consists of 2,000 units, can hold roughly 20 items at a time out of a 15,625-symbol alphabet. How would DCPS scale if the alphabet size were raised to 50,000? With the current alphabet size, how many units would have to be added simply to double the working memory capacity to 40 triples? We present some analytical results related to these questions.
Scaling Properties of Coarse-Coded Symbol Memories

Ronald Rosenfeld
David S. Touretzky

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Summary

Coarse coded memories have appeared in several neural network symbol processing models, such as Touretzky and Hinton's distributed connectionist production system DCPS [6,7], Touretzky's distributed implementation of Lisp S-expressions on a Boltzmann machine, BoltzCONS [8,9], and St. John and McClelland's PDP model of case role defaults [4]. In order to determine how these models would scale, one must first have some understanding of the mathematics of coarse coded representations. For example, the working memory of DCPS, which stores triples of symbols, consists of 2,000 units, can hold roughly 20 items at a time out of a 15,625-symbol alphabet. How would DCPS scale if the alphabet size were raised to 50,000? With the current alphabet size, how many units would have to be added simply to double the working memory capacity to 40 triples? We present some analytical results related to these questions.

A coarse coded symbol memory in its most general form is defined by two parameters: the alphabet size $a$ and the number of units, $N$. Each unit has a "receptive field" containing some subset of the alphabet. Symbols are stored in memory by turning on all the units in whose receptive field they fall. Thus, symbols are represented as distributed patterns of activity, and the units are said to be "coarsely tuned" because each participates in the representation of more than one symbol. However, our units' receptive fields are not restricted to contiguous subregions of a multidimensional feature space as are the "value units" of [1,2,3,5]. They are instead random subsets of a one-dimensional symbol space.

A symbol is deemed present if all its receptors are active (our analysis easily generalizes to a weaker criterion). As items are added, "ghost" symbols eventually appear; these are symbols which were not stored, but appear because all their receptors are shared with symbols that were stored. The capacity or "span" of a memory is the number of symbols $k$ that can be stored before ghosts appear. (A localist representation, where $k = a = N$, is very inefficient for sparse memories with a large alphabet.)

In this analysis we assume that symbols have a uniform receptor set size $L$, and that each of the
\[ P_{\text{ghost}}(N, L, k, \alpha) = 1 - \sum_{c=0}^{N} T_{N, L}(k, c) \left[ 1 - \left( \frac{c}{N} \right) \right]^{a-k} \]  

(1)

\[ T_{N, L}(k, c) \] is the probability that exactly \( c \) units will be active after \( k \) symbols have been stored. It is defined by Equation 2:

\[ T_{N, L}(k, c) = \sum_{a=0}^{L} T(k - 1, c - a) \cdot \frac{\binom{N-(c-a)}{a} \binom{c-a}{L-a}}{\binom{N}{L}} \]  

(2)

\[ T_{N, L}(0, c) = 0 \] for all \( c \).

The optimal pattern size with respect to \( N \), \( k \), and \( \alpha \) can be determined by binary search on Equation 1. However, this may be expensive for large \( N \). A good initial estimate is the \( L \) that maximizes the following expression:

\[ \frac{\left( N \left[ 1 - (1-L/N)^k \right] \right)}{\binom{N}{L}} \]  

(3)

We have constructed coarse coded memories of various sizes and measured their capacities experimentally. The results show good agreement with the predicted values.

We present graphs of the relationships between \( N \), \( k \), \( \alpha \), and \( P_{\text{ghost}} \) for optimum pattern sizes, as determined by Equation 1. A representative graph is shown in Figure 1. The results show an exponential relationship between \( \alpha \) and \( N/k \). Thus, for a fixed alphabet size, the span is proportional to the number of units. For \( P_{\text{ghost}} = 0.01 \) the relationship is:

\[ \alpha = e^{0.468 \frac{N}{k} - 4.76} \]  

(4)

We compare the capacity obtained using our probabilistic, random receptive fields approach with that of two other approaches which guarantee a specified span: a binary coding scheme, and an approach where the overlap between any two patterns is bounded by \( \lfloor (L - 1)/k \rfloor \).
References


Figure 1: A graph of $\log \alpha$ vs. $N$ for even $k$ values (memory capacity) from 2 to 20. $p = 0.01$. 