In Proceedings of the 12th International Joint Conference on Artificial Intelligence, edited by John Mylopoulos and Ray Reiter, Morgan Kaufmann, San Mateo, CA, 1991.

Holographic Reduced Representations: Convolution Algebra for Compositional Distributed Representations

Tony Plate
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada, M5S 1A4
tap@ai.utoronto.ca

Abstract

A solution to the problem of representing compositional structure using distributed representations is described. The method uses circular convolution to associate items, which are represented by vectors. Arbitrary variable bindings, short sequences of various lengths, frames, and reduced representations can be compressed into a fixed width vector. These representations are items in their own right, and can be used in constructing compositional structures. The noisy reconstructions given by convolution memories can be cleaned up by using a separate associative memory that has good reconstructive properties.

1 Introduction

Distributed representations [Hinton, 1984] are attractive for a number of reasons. They offer the possibility of representing concepts in a continuous space, they degrade gracefully with noise, and they can be processed in a parallel network of simple processing elements. However, the problem of representing compositional structure in distributed representations has been for some time a prominent concern of both followers and critics of the connectionist faith [Fodor and Pylyshyn, 1988; Hinton, 1990] .

Using connectionist networks, e.g., back propagation nets, Hopfield nets, Boltzmann machines, or Willshaw nets, it is easy to represent associations of a fixed number of items. The difficulty with representing compositional structure in all of these networks is that items and associations are represented in different spaces. Hinton [1990] discusses this problem and proposes a framework in which "reduced descriptions" are used. This framework requires that a number of vectors be compressed (reduced) into a single vector of the same size as each of the original vectors. This vector acts as a reduced description of the set of vectors and itself can be a member of another set of vectors. The reduction must be reversible so that one can move in both directions in a part-whole hierarchy. In this way, compositional structure is represented. However, Hinton does not suggest any concrete way of performing this reduction mapping.

Some researchers have built models or designed frameworks in which some some compositional structure is present in distributed representations. For some examples see the papers of Touretzky, Pollack, or Smolensky in [AIJ, 1990].

In this paper I propose a new method for representing compositional structure in distributed representations. Circular convolution is used to construct associations of vectors. The representation of an association is a vector of the same dimensionality as the vectors which are associated. This allows the construction of representations of objects with compositional structure. I call these *Holographic Reduced Representations* (HRRs), since convolution and correlation based memories are closely related to holographic storage, and they provide an implementation of Hinton's [1990] reduced descriptions. I describe how HRRs and error correcting associative item memories can be used to build distributed connectionist systems which manipulate complex structures. The item memories are necessary to clean up the noisy items extracted from the convolution representations.

2 Associative memories

Associative memories are used to store associations between items which are represented is a distributed fashion as vectors. Nearly all work on associative memory has been concerned with storing items or pairs of items.

Convolution-correlation memories (sometimes referred to as holographic-like) and matrix memories have been regarded as alternate methods for implementing associative memory [Willshaw, 1981; Murdock, 1983; Pike, 1984; Schönemann, 1987]. Matrix memories have received more interest, probably due to their relative simplicity and their higher capacity in terms of the dimensionality of the vectors being associated.

The properties of matrix memories are well understood. Two of the best known matrix memories are "Willshaw" networks [Willshaw, 1981] and Hopfield networks[Hopfield, 1982]. Matrix memories can be used to construct auto-associative (or "content addressable") memories for pattern correction and completion. They can also be used to represent associations between two vectors. After two vectors are associated one can be used as a cue to retrieve the other.

There are three operations used in associative memories: encoding, decoding, and trace composition. The

encoding operation takes two item vectors and produces a memory trace (a vector or a matrix). The decoding operation takes a memory trace and a single item (the cue), and produces the item that was originally associated with the cue, or a noisy version thereof. Memory traces can be composed by addition or superposition. The decoding operation will work with this sum of individual traces, but the retrieved items may be noisier. In some models encoding and decoding are linear, e.g., Murdock [1983], in others decoding is non-linear, e.g., Hopfield [1982], and in others all the operations are non-linear, e.g., Willshaw [1981].

To illustrate this, let $\mathbb{I}$ be the space of vectors representing items [1], and $\mathbb{T}$ be the space of vectors or matrices representing memory traces. Let $\boxtimes : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{T}$ be the encoding operation, $\triangleright$: $\mathbb{I} \times \mathbb{T} \rightarrow \mathbb{I}$ be the decoding operation, and $\boxplus : \mathbb{T} \times \mathbb{T} \rightarrow \mathbb{T}$ be the trace composition operation. Let $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$, $\mathbf{e}$, and $\mathbf{f}$ be item vectors, and let T_i be memory traces.

The association of two items $\mathbf{a}$ and $\mathbf{b}$ is represented by the trace $T_1 = \mathbf{a} \boxtimes \mathbf{b}$. We can recover $\mathbf{a}$ from T_1 by using the decoding operation on T_1 and the cue $\mathbf{b}$: $\mathbf{b} \triangleright T_1$ gives $\mathbf{a}$, or a noisy version of it. Noisy versions of $\mathbf{b}$ can also be used as cues. Depending on the properties of the particular scheme, the retrieved vector will be more or less similar to $\mathbf{a}$.

A trace can represent a number of associations, e.g., $T_2 = (\mathbf{a} \boxtimes \mathbf{b}) \boxplus (\mathbf{c} \boxtimes \mathbf{d}) \boxplus (\mathbf{e} \boxtimes \mathbf{f})$. An item from any pair can be used as a cue to recover the other item of the pair, e.g., $\mathbf{c} \triangleright T_2$ gives a noisy version of $\mathbf{d}$. The noisiness of the recovered vector increases with the number of associations stored in a single memory trace. The number of associations that can be represented usefully in a single trace is usually referred to as the *capacity* of the memory model.

In matrix memories the encoding operation is the outer product, and in convolution memories the encoding operation is convolution. Addition and superposition have both been used as the trace composition operation in matrix and convolution memories.

2.1 Convolution-correlation memories

In nearly all convolution memory models the aperiodic convolution operation has been used to form associations.[2] The aperiodic convolution of two vectors with n elements each results in a vector with $2n-1$ elements. This result can be convolved with another vector (recursive convolution); and if that vector has n elements, the result has $3n-2$ elements. Thus the resulting vectors grow with recursive convolution. This same growing property is exhibited in a much more dramatic form by both matrix memories and Smolensky's [1990] tensor product representations.

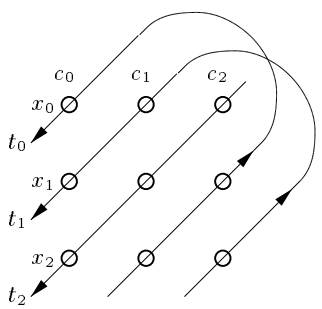
Researchers have used three solutions to this problem of growth with recursive associations - (a) limit the depth of composition (Smolensky [1990]), (b) discard elements outside the n central ones (Metcalfe [1982]), and (c) use infinite vectors (Murdock [1982]).

The growth problem can be avoided entirely by the use of circular convolution, an operation well known in signal processing. The result of the circular convolution of two vectors of n elements has just n elements. Since circular convolution does not have the growth property, it can be used recursively in connectionist systems with fixed width vectors.

There is a close relationship between matrix and convolution memories. A convolution of two vectors (whether circular or aperiodic) can be regarded as a compression of the outer product of those two vectors. The compression is achieved by summing along the transdiagonals of the outer product.

Circular convolution, denoted by the symbol $\circledast$, is illustrated in Figure 1. Each of the small circles represents the product of a pair of elements from $\mathbf{a}$ and $\mathbf{b}$, and these are summed along the indicated diagonals. While the circular convolution operation is straightforward, what is remarkable is that circular correlation, $\circledcirc$, (illustrated in Figure 2) is its approximate inverse.[3] If a pair of vectors is convolved together to give a "memory trace", then one member of the pair, the "cue", can be correlated with the trace to produce the other member of the pair. Suppose we have a trace which is the convolution of the cue with another vector, $\mathbf{t} = \mathbf{c} \circledast \mathbf{x}$. Then correlation allows the reconstruction of a noisy version of $\mathbf{x}$ from $\mathbf{t}$ and $\mathbf{c}$: $\mathbf{y} = \mathbf{c} \circledcirc \mathbf{t}$ where $\mathbf{y} \approx \mathbf{x}$. The correlation operation also has aperiodic and circular versions, which are approximate inverses for the respective convolution operations.

Circular convolution is defined as: $\circledast : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}^n$, such that $\mathbf{t} = \mathbf{c} \circledast \mathbf{x}$ where $t_j = \sum_{k=0}^{n-1} c_k x_{j-k}$. Subscripts are interpreted modulo n, which gives the operation its circular nature.



$$\mathbf{t} = \mathbf{c} \circledast \mathbf{x}$$

$$\begin{aligned} t_0 &= c_0x_0 + c_2x_1 + c_1x_2 \\ t_1 &= c_1x_0 + c_0x_1 + c_2x_2 \\ t_2 &= c_2x_0 + c_1x_1 + c_0x_2 \end{aligned}$$

$$t_j = \sum_{k=0}^{n-1} c_k x_{j-k}$$

for $j = 0$ to $n-1$

(Subscripts are modulo-n)

Figure 1: Circular convolution represented as a compressed outer product for $n = 3$.

Circular correlation is defined as: $\circledcirc : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}^n$, such that $\mathbf{y} = \mathbf{c} \circledcirc \mathbf{t}$ where $y_j = \sum_{k=0}^{n-1} c_k t_{k+j}$.

Convolution can be computed in $O(n \log n)$ time using Fast Fourier Transforms (FFTs). The method is simple and well known: $\mathbf{a} \circledast \mathbf{b} = f^{-1}(f\mathbf{a} \odot f\mathbf{b})$, where f is a discrete Fourier transform (its range is a vector of complex numbers), f^{-1} is its inverse, and $\odot$ is the component-wise multiplication of two vectors.

[1] There are usually distributional constraints on the elements of the vectors, e.g., the elements should be drawn from independent distributions.

[2] The exception is the non-linear correlograph of Willshaw [1981], first published in 1969.

[3] Under certain conditions on the distribution of elements in the vectors.