ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 189-204

Variable Binding in a Neural Network using a Distributed Representation.

Antony Browne and John Pilkington.

Centre for Information Engineering, School of Electrical, Electronic & Information Engineering, South Bank University, London SE1 0AA. England. E-mail brownea@uk.ac.sbu.vax.

Abstract. The operation of variable binding has been difficult to achieve using connectionist networks. Some localist connectionist models achieve it in a limited form, but are open to the criticism that they are merely implementations of classical symbol processing systems. Until now no connectionist system using a distributed representation has attempted to perform the complex form of variable binding called unification. We describe a connectionist system that performs unification directly on a distributed representation.

Introduction.

Researchers in the connectionist camp work with models using two broad types of representation, those using localist representations (including marker-passing models) and those using distributed representations. In models using localist representations, the representations themselves coincide with computational tokens, in such networks a single unit would represent a single concept.They have been popular with many AI researchers, however there is nothing in these models that does not appear in classical symbolic processing models, and localist connectionist systems have been criticised as being mere implementations of these. A rigorous definition of the notion of distribution used in distributed connectionist models was put forward by Van Gelder (1992). The representations formed in a standard feed-forward network can be considered as distributed as they are context dependent compositions of the input constituents (Sharkey, 1992). It has been argued that distributed representations cannot display compositionality (Fodor & Pylyshyn,1988), but this position has been weakened by some recent connectionist models which demonstrate a form of functional compositionality when performing structure-sensitive tasks (Van Gelder, 1990). In one of these a model using distributed representations was used to perform the passivisation of English sentences (Chalmers, 1990). Syntactic transformations were produced by this model which acted directly on the compressed distributed representations of sentences, without passing through any stages of decomposition (extraction) of the constituents. This, together with other model that also perform syntactic operations on distributed representations (Chrisman, 1991; Niklasson & Sharkey, 1991), demonstrate that functionally compositional distributed representations can perform structure sensitive operations without extracting the individual

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 189-204

constituents, and provide evidence that distributed connectionist systems can be both compositional and systematic.

Variable Binding and Unification.

Variable binding is usually understood as a process in which symbols are designated to represent entities.The ability to handle variables seems to be essential for connectionist systems if they are ever to perform complex reasoning tasks, so a demonstration of variable binding by a system using a fully distributed representation would provide additional evidence for the expressive power of these models. In addition, if it to be demonstrated that distributed representations can support cognitive processes equivalent to those demonstrated by symbolic systems, then it will have to be shown that theorem proving is within their scope. Modern ideas on theorem proving were pioneered by Robinson (1965), who developed a method for theorem proving using resolution. Central to this method was unification of first-order terms, for which an algorithm was constructed and proven correct. Unification can best be described as a complex form of pattern matching and variable binding, where both the pattern and the datum (which the pattern is to be matched against) can contain variables.The unification problem is usually stated in the following context: Considering two terms of logic built up from function symbols (**f,a,...**) variables (**X,Y,Z...**), and constants (**a,b,c,...**), is there a substitution of terms for variables that will make the two terms identical? For example, considering the two terms **f(X,Y) & f(g(Y,a),h(a))** they are said to be unifiable since binding the variable **X** to the term **g(h(a),a)** and the variable **Y** to the term **h(a)** will make both terms look like **f(g(h(a),a),h(a))** which is called the most general unifier (M.G.U., intuitively the simplest of their unifiers). If we try to unify the two terms **X = g(X)** we find that this is unsolvable, since the variable **X** cannot be equal to a term which contains **X** as a proper subterm. This is often referred to as the occurs check problem since we have to check whether **X** occurs in **g(X)** in order to solve the unification problem. Many of the implementations of unification attempted so far using connectionist representations have been part of larger inference systems such as production systems, and all have performed unification using localist representations (Ballard & Hayes, 1984; Sun, 1992). An effective connectionist unification algorithm has been constructed using a localist representation by Hölldobler (1990).

Performing Unification on Distributed Representations.

We have used Hölldobler's scheme to produce a symbolic representation for terms to be unified, and have forced an autoassociative network to produces a distributed representation. Subsequently, we have performed unification directly on the distributed representations. A simple alphabet of constants and variables was chosen to generate the term pairs to be unified, consisting of two constants **a** and **f**, and two variables **X** and **Y**. Four positions were used when generating the compound terms, **functor, 1, 1.1** and **2**. This allowed two-argument compound terms such as **f(X,Y)** or

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 189-204

f(a(X),Y) to be generated. All possible permutations of the members of this alphabet were generated to produce a set of two-argument terms, with the restriction that only **f** or **a** appear at the functor position of a compound term (it is not allowable to have an uninstantiated variable as the principal functor of a compound term). These terms were then combined in pairs to generate all of the 9216 term pair permutations possible for this alphabet, and this set of term pairs was then partitioned into training and test sets for the network.The input representation signifying the two terms before unification was designed with two different sets of binary units, one set of 16 units for each of the terms in the term pair, giving 32 units in total. An example of the patterns generated on one of these sets of units is shown in Figure 1 for the term **f(a(X),Y)**. In this representation units were given a value of 1 if a particular functor, constant or variable was present at that argument position, and a value of 0 otherwise. The output representation had 21 units, an extra row of 4 units as compared with the pre-unification terms used to signify the propagation of a nested binding to the second argument position, and an occurs check unit which was set to 1 when occurs check failure was detected. Unification failure was signified by setting all the units to 0, with the exception of occurs check failure where all the units were set to 0 and the occurs check unit set to 1. In the example shown in Figure 2 the two spotted units are those that become active after the unification of **f(a(X),Y) & f(Y,Y)** due to the propagation of the binding **Y = a(X)** to the second argument position.The distributed representations for the pre-unification terms were generated by autoassociating them in a standard three-layer backpropagation network.This resulted in a distributed representation of each term pair being formed on the hidden layer of network A when that term pair was presented as input. This process was repeated for the unification results, the results for all the possible permutations being autoassociated in another network (network B). A third network (network C) was then designed to actually perform the unification on the distributed representations formed on the hidden layer of network A by taking these as input, and producing as output the correct distributed representations that were expected on the hidden layer of network B. A backpropagation network was again chosen, but in this case a 4 layer network was used, as this was found empirically to give better results. This network was only trained on the training corpus, the test corpus was held in reserve to test the generalisation performance of the network. Figure 3 shows the complete system with all three networks and their relative positions in the system.

Experimental Results.

Network C correctly unified 97% of the term pairs in the training set and generalised to 95% of the terms in the test set. The term pairs incorrectly unified were scattered randomly throughout both the training corpus and the test corpus, indicating that the network is not having difficulty with a particular type of term. As a practical unification machine for theorem proving, this figure may not be acceptable. However the intention behind developing this model was to demonstrate variable binding on distributed representations. As such can be judged to have fulfilled that task even

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 189-204

though it has not achieved 100% accuracy.

Discussion.

This model demonstrates that distributed representations can support variable binding and unification. In addition it has demonstrated functional compositionality by showing that a complex structure-sensitive processes can be performed on a distributed representation. The performance of unification is a more complex task than those achieved previously using distributed representations, as it involves the generation of new arguments and the propagation of these to new argument positions. For example, consider the unification of the term pair **a(X,Y) & a(a(Y),a)** to give **a(a(a),a)**, here the new term **a(a)** has been generated by the unification, and the constant **a** has been propagated to the 2.1 argument position. In addition, although some previous tasks performed on distributed representations have been context sensitive (Chrisman, 1991; Niklasson & Sharkey, 1991), unification of terms with an arity of greater than 1 can be said to be sensitive to the context of context, as one constituent must be evaluated in the context of another constituent, which must then be evaluated in the context of other constituents. For example in the term pair **f(a(X),X) & f(Y,Y),** firstly **a(X)** in the first argument position of the first term must be evaluated against **Y** in the first argument position of the second term, and then **Y** (given it's new context **a(X)**) in the second argument position of the second term must be evaluated in the context of the **X** in the second argument position of the first term. This gives **a(X) = Y = X**, but **a(X) = X** is illegal so an occurs check failure must be signified. The performance of unification using these representations adds further evidence that distributed connectionist representations can demonstrate compositionality and support structure-sensitive tasks.We are currently investigating methods of performing a resolution step on the distributed representations that represent the variable bindings generated by network C. Resolution is a powerful inference rule (so powerful that it is the only rule needed for a sound and complete system of logic), and amounts to an algebraic cancelling process. From two clauses that contain a complementary pair of literals (literals that are exactly alike except that one is negated and the other is not), using resolution we can deduce a third clause that is the disjunction of all the remaining literals of the two clauses after striking out the complimentary pair. Unification is used to produce the required variable substitutions, for example the resolution **p(X)** and **not (p(a))** requires the binding **X = a** as only this value would produce the desired contradiction. The performance of this inference rule by an extension of our model will add further evidence that distributed connectionist representations can perform complex reasoning tasks.

References.

Ballard, D.H. and Hayes, P.J (1984) Parallel logical inference, *Proceedings of the 6th annual conference of the cognitive science society,* pp. 114-123.

Chalmers, D.J (1990) Syntactic transformations on distributed representations, *Connection Science* **2**(1 & 2) pp. 53-62.

Chrisman, L (1991) Learning recursive distributed representations for holistic computation, *Connection science* **3**(4) pp. 345-365.

Fodor, J.A and Pylyshyn, Z (1988) Connectionism and cognitive architecture: a critical analysis, *Cognition* **28** pp. 3-71.

Hölldobler, S (1990) A connectionist unification algorithm, *Technical Report TR-90-012*, International Computer Science Institute, Berkeley, U.S.A.

Niklassson, L. and Sharkey, N.E (1992) Connectionism and the issues of compositionality and systematicity, In R.Trappl (Ed.) *Cybernetics and systems*, Kluwer academic press, Dordrecht.

Robinson, J.A (1965) A machine-oriented logic based on the resolution principle. *Journ. ACM* **12** pp. 23-41.

Sun, R (1992) On variable binding in connectionist networks, *Connection Science* **4**(2) pp. 93-124.

Sharkey, N.E (1992) The ghost of the hybrid: a study of uniquely connectionist representations, *AISB Quarterly* **79** pp. 10-16.

Van Gelder, T.J (1992) Defining 'Distributed Representation', *Connection Science* **4**(3&4) pp.175-192.

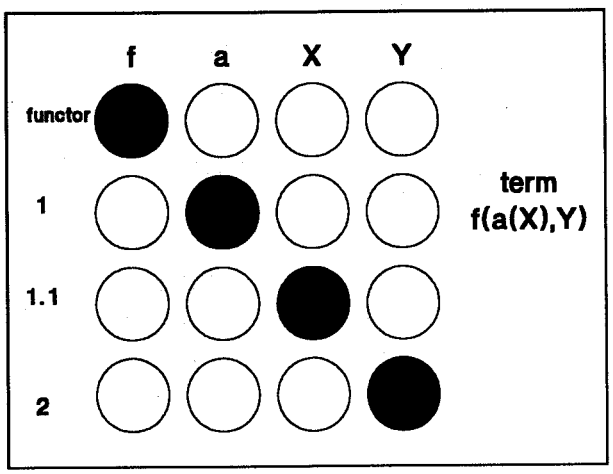

Figure 1: Example term **f(a(X),Y)**.

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 189-204

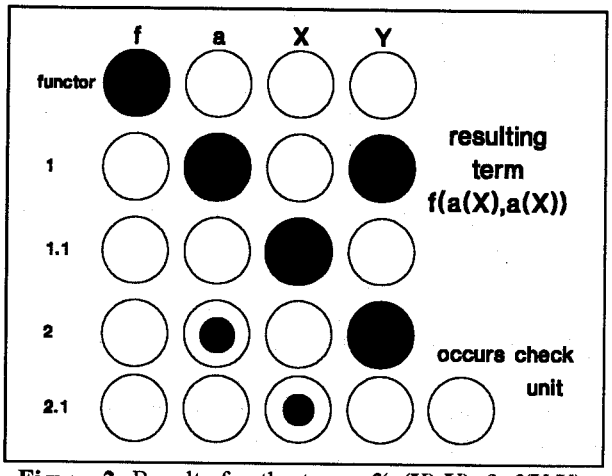

Figure 2: Results for the terms **f(a(X),Y) & f(Y,Y)**.

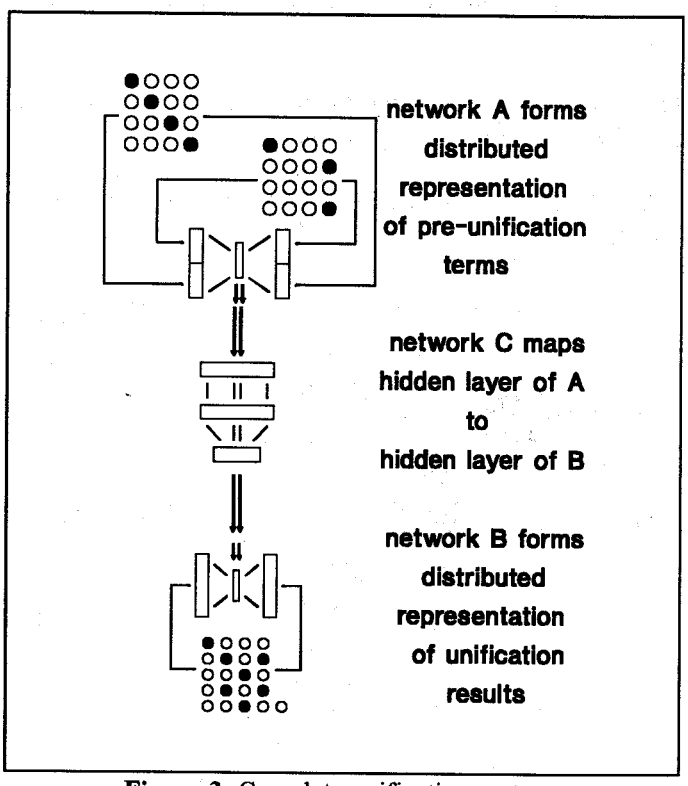

Figure 3: Complete unification system.