A Connectionist Model for Predicate Logic Reasoning Using Coarse-Coded Distributed Representations

Sriram G. Sanjeevi[1,*] and Pushpak Bhattacharya[2,**]

[1] Dept. of Comp. Science & Engg.,N.I.T. Warangal, Warangal 506004, 91-0870-2430440
sgsanjeevi@yahoo.com
[2] Dept. of Comp. Science & Engg.,I.I.T. Bombay, Mumbai 400076, 91-22-5767718
pb@cse.iitb.ac.in

Abstract. In this paper, we describe a model for reasoning using forward chaining for predicate logic rules and facts with coarse-coded distributed representations for instantiated predicates in a connectionist frame work. Distributed representations are known to give advantages of good generalization, error correction and graceful degradation of performance under noise conditions. The system supports usage of complex rules which involve multiple conjunctions and disjunctions. The system solves the variable binding problem in a new way using coarse-coded distributed representations of instantiated predicates without the need to decode them into localist representations. Its performance with regard to generalization on unseen inputs and its ability to exhibit fault tolerance under noise conditions is studied and has been found to give good results.

1 Introduction

Traditionally reasoning systems using predicate logic have been implemented using symbolic methods of artificial intelligence. Connectionist methods of implementation of reasoning systems describe an alternative paradigm. Among the connectionist systems they use two types of representational schemes. They are 1) localist and 2) distributed representational schemes.

Localist representational schemes represent each concept with an individual unit or neuron. In the distributed representational schemes [3] each unit or neuron is used in representation of multiple concepts and multiple units or neurons are used to represent a single concept. In the literature, some localist methods for reasoning using connectionist networks have been described. The connectionist inference system *SHRUTI* [1], [5] described a localist method where temporal synchrony was used to create bindings between variables and entities they represent. *CONSYDERR* [2] described another localist method for variable binding and forward reasoning. It used an assembly or a set of interconnected nodes to represent each predicate $p(x_1.....x_k)$. Since, these systems used localist representations, advantages of distributed representations are not obtainable by them and hence the motivation for a distributed representation based reasoning system.

---

[*] Asst. Professor
[**] Professor

R. Khosla et al. (Eds.): KES 2005, LNAI 3682, pp. 732–738, 2005.

2 Rule and Fact Base

Our system represents and reasons with predicate logic rules and facts. Following are rules and facts we use.

1. *give(x, y, z) —> own(y, z);*
2. *own(y, z) —> donate(y, z);*
3. *own(y,z)* $\wedge$ *wantstobuy(w, z)*$\wedge$*(hasrequiredmoney(w, m)* $\vee$ *hasgoodcreditrating(w)) —> cansell(y,w,z);*
4. *give(John,Mary,Book-1);*
5. *give(John,Chris,Book-2);*
6. *wantstobuy(Walter,Book-2);*
7. *hasrequiredmoney(Walter,Money);*
8. *hasgoodcreditrating(Walter);*

Our system uses the above rule base and makes inferences shown below.

1. *own(Mary,Book-1);*
2. *donate(Mary,Book-1);*
3. *own(Chris,Book-2);*
4. *cansell(Chris,Walter,Book-2);*

3 Forward Reasoning Using Connectionist System

In this paper we see how to accomplish the forward reasoning for predicate calculus facts and rules using neural networks which operate on coarse coded distributed representations. Each fact of predicate p_i is represented by a vector $\mathbf{v_{ij}}$. The vector $\mathbf{v_{ij}}$ is a k dimensional vector which stores the coarse coded representation of predicate fact. The different instantiations of predicate p_i are each represented by separate vector $\mathbf{v_{ij}}$ where j varies from *1* to *m* where *m* is the number of vectors in predicate p_i table.

We describe here, briefly with an example how forward reasoning using localist representations [6],[7] is made using a connectionist system. Let us consider the *rule 1:give(x,y,z)—> own(y,z)* from the knowledge base. The localist pattern for the LHS of rule 1 can be written as *0001 001 001 001 1*. The first 4 bit value denotes the predicate *give*, the next 3 bit value denotes an object getting bound to variable *x* '*John'*, the next 3 bit value denotes an object getting bound to variable *y*, *'Mary'* and the next value denotes, '*Book-1'*. The last bit indicates the truth value of predicate *give*. This instantiation will activate rule *1* and make variables on the right hand side of the rule *'y'* and *'z'* be assigned the values *'001'* and '*001*' representing the objects *'Mary'* and ' *Book-1'* respectively. Because of the rule activation the localist pattern representation for RHS will be *0010 001 001 1* denoting *own(Mary,Book-1)*. This triggers the rules whose left hand sides match RHS of rule 1 and through this forward chaining, forward reasoning using localist representations is accomplished. In *Table 1* and *2* below we show samples of localist vectors for some of the predicates in the rule base.

Table 1. Shows a sample of localist tuples used by predicate *give*

S.No of Tuple	Predicate 'id' code	Localist Value of x	Localist Value of y	Localist Value of *z*	Truth Value of Predicate
215	00001000000	0000100000	0000100000	0000010000	00001