Outline

Finish discussion of EBNN

Symbolic and Neural Network EBL

KBANN

Summary: Inductive-Analytical methods for Neural Network learning

Copyright Tom Mitchell, 1994

The Explanation Based Learning Problem

Given:

- Instance space
- Hypothesis space
    - Symbolic: horn clauses (first order)
    - Neural net: networks (propositional)
- Target concept
- Training examples
- Domain Theory
    - Symbolic: horn clauses
    - Neural net: networks

Determine:

- Hypothesis that fits training examples and domain theory

Copyright Tom Mitchell, 1994

Explanation Based Learning Method

For each training example:

1. *Explain* how training example satisfied target function
   - Symbolic: chain together horn clauses
   - Neural net: chain together neural networks
2. *Analyze* explanation to determin relevance of features
   - Symbolic: weakest preconditions
   - Neural net: derivative of target fn wrt example features
3. *Refine* hypothesis
   - Symbolic: add new rule with weakest preconditions
   - Neural net: update weights analytically *and inductively*

Copyright Tom Mitchell, 1994

Differences between Symbolic, ANN EBL

Correctness and completeness (coverage) of target function

- Symbolic: Always correct$_*$, Increasingly complete
- Neural net: Always complete, Increasingly correct

Scaling up to complex target functions

- Symbolic: Problems = cost of matching many rules, coverage
- Neural net: Problem = inability to closely approximate target function in fixed size representation

Copyright Tom Mitchell, 1994

More Differences

Mechanisms for learning new features

- Symbolic: analytically, from single explanation
- Neural net: hidden units defined statistically from many examples

Permanence of memory:

- Symbolic: never forget learned rules
- Neural net: current weights (knowledge) gets overwritten

Human comprehensibility

- Symbolic: More understandable

Copyright Tom Mitchell, 1994

KBANN Overview

- **Purpose:** use propositional domain theory to guide ANN learning
- **Key idea:**
  - analytically construct ANN equivalent to domain theory,
  - then inductively refine it using Backpropagation
- **Applications:** Identifying promoter sequences in DNA, predicting protein folding
- **Reference:** Shavlik and Towell, "An approach to combining expalnation-based and neural learning algorithms", *Connection Science*, (1), pp. 233-255, 1989.
- **Related efforts:** Neural network tuning of Fuzzy rules (e.g., Fu)

Copyright Tom Mitchell, 1994

KBANN: The Problem

Given:

- Training examples of target concept (boolean)
- Approximate domain theory (propositional horn clauses)

Determine:

- Target concept (ANN) that fits the data, biased by the domain theory

Copyright Tom Mitchell, 1994

Insert

slides from the Shavlik/Towell paper

Copyright Tom Mitchell, 1994

Converting the Domain Theory to an ANN

1. For each example attribute, create an input node
2. For each rule in DT, create a node that activates when the rule matches
    a. Inputs to this node are the rule antecedents
    b. Assign weight W to non-negated antecedents
    c. Assign weight -W to negated antecedents
    d. Set threshold to W(n-.5), where n is number of non-negated antecedents
3. Add further connections among units. Connect each unit at depth i to all units at depth i+1. Assign weights of approximately 0 to these.

Copyright Tom Mitchell, 1994

Lessons from KBANN

Main idea:

- Approximate (propositional) theory can initialize ANN
- Inductive weight tuning can overcome deficiencies of theory
- Outperforms inductive Backpropagation, given reasonable domain theory

Some limitations:

- Propositional domain theory
- Learns only boolean concepts (not real-valued functions)

Copyright Tom Mitchell, 1994