# Explanation Based Learning

- Generalizes better from less data, more prior knowledge
- In pure form, requires *perfect* prior knowledge
- What problems provide perfect prior knowledge?
    - Learning search control

Copyright Tom Mitchell, 1994

# A Simple Domain Theory for "Good Move"

---

*Domain independent part:*

Good-move(s,a) $\leftarrow$ Successor(s,a,snew)
Solvable(snew)

Solvable(s) $\leftarrow$ Goal-State(s)

Solvable(s) $\leftarrow$ (There-Exists a) Successor(s,a,snew)
Solvable(snew)

*Domain specific part:*

Goal-state(s) $\leftarrow$ $Distance_2$=0

Successor(s,$A_{turn+10}$,snew) $\leftarrow$ $Theta_1$(s)=a
$Theta_1$(snew)=a+10
$Distance_1$(s)=b
$Distance_1$(snew)=b
...

Successor(s,$A_{forward+5}$,snew) $\leftarrow$ ...

...

**Copyright Tom Mitchell, 1994**

# Explanation-based Learning of Control Knowledge in PRODIGY

---

[Minton, 1988]

PRODIGY

- STRIPS-like problem solver
- Search control choices:
    - choose node
    - choose subgoal
    - choose operator
    - choose bindings

Learns variety of specialized target concepts:

- Operator-succeeds
- Goal-fails
- Operator-results-in-goal-interference
- Goal-results-in-goal-interference
- ...

**Copyright Tom Mitchell, 1994**

# EBL of Control Knowledge in PRODIGY

---

Learn from successes and failures

Empirically evaluate utility of learned rules

- Delete those whose cost>benefit

Learns useful control knowledge for variety of domains

**Copyright Tom Mitchell, 1994**

# Example Rule Learned via EBL

---

[Minton, 1988]

Target concept: Node-results-in-goal-interference

Learned rule:

```
IF (AND (CURRENT-NODE ?node)
        (CANDIDATE-GOAL ?node (ON ?x ?y))
        (CANDIDATE-GOAL ?node (ON ?w ?x)))
THEN (PREFER GOAL (ON ?x ?y) TO (ON ?w ?x))
```

Copyright Tom Mitchell, 1994

# A Problem with Learning Control Rules

---

The more rules you learn, the slower you get...

Problem: Need *useful* rules, not necessarily the full *truth*!

Approaches to handling this:

- Monitor utility, and delete worthless rules
    - e.g., PRODIGY
- Index rules cleverly
    - e.g., SOAR, RETE
- Learn efficient approximate rules
    - e.g., Tadepalli LAZY EBL

**Copyright Tom Mitchell, 1994**

# Utility Evaluation in PRODIGY

---

Utility = (AvrSavings x ApplicFreq) - AveMatchCost

Utility considered:

- At learning time: only create rules with high estimated utility
- During use: collect empirical data, and prune low utility rules

Effect:

| | Blocks Domain | Robot Domain | Sched Domain |
|---|---|---|---|
| **Learning opps** | **328** | **588** | **946** |
| **Rules generated** | **69** | **130** | **243** |
| **Useful rules** | **19** | **30** | **37** |

**Copyright Tom Mitchell, 1994**

# EBL in PRODIGY: Summary

- Successful learning of control knowledge across domains
- Domain independent plus domain specific prior knowledge
- Utility analysis is critical

Copyright Tom Mitchell, 1994

# What We Would Like

A single learning method that operates over entire spectrum:

- Given perfect domain theory, generalizes accurately
- Given no domain theory, generalizes like inductive learning
- Degrades gracefully with quality of domain theory
- Learns its own domain theory
- Accomodates noisy data

**Copyright Tom Mitchell, 1994**

# EBNN: The Problem

Given:

- Training examples of target function (real-valued)
- Approximate domain theory (previously learned ANN's)

Determine:

- Target function (ANN) that fits the training data and the domain theory

Copyright Tom Mitchell, 1994

# EBNN Overview

- **Purpose:** use previously learned neural networks to guide learning subsequent neural networks
- **Key idea:**
  - use neural network domain theory to explain each observed example
  - analyze explanation to determine relevance of example features
  - train target network to fit analytically determined relevance, and observed data
- **Applications:** Acquiring robot control strategies, robot perception
- **Reference:** [Mitchell and Thrun, 92]
- **Related efforts:** This is ANN analog of symbolic EBL [DeJong 86], [Mitchell 86]. Weight tuning algorithm is based on Simard's TangentProp [Simard 92].

Copyright Tom Mitchell, 1994

# Explanation-based Neural Network Learning

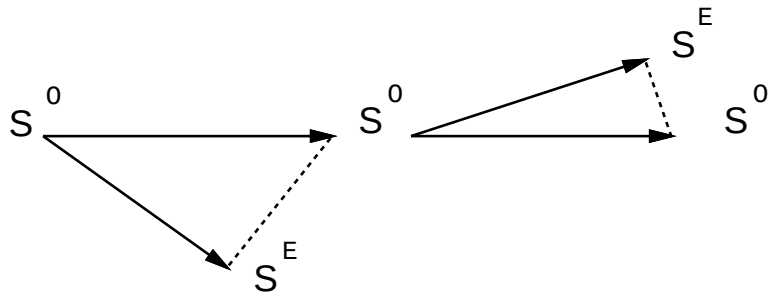
EBNN

For each observed example:

1. Explain:
   - Show how example satisfied target concept, using domain theory
2. Find dependency of target concept on training example features:
   - Extract derivative of target concept activation wrt training example activations
3. Update representation of target concept:
   - (inductive) Update weights to better fit observed training example
   - (analytical) Update weights to better fit analytically determined derivative
   - Weight the analytical component by observed quality of explanation

Copyright Tom Mitchell, 1994

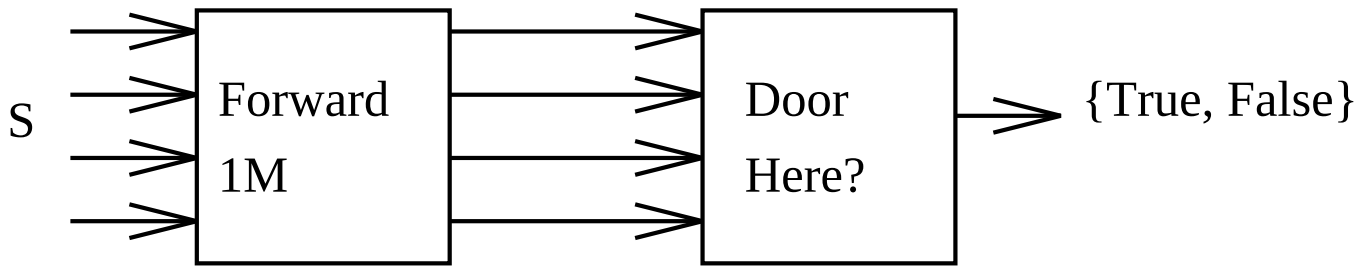
## How Strongly Should EBNN Trust the Explanation?



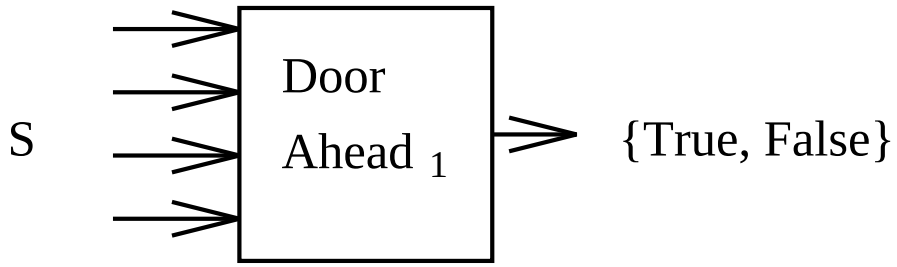

Heuristic LOB*: Weight analytical component of learning inversely with product of explanation errors

# Robot Perception: Domain Theory and Target Funct

**Domain Theory:**

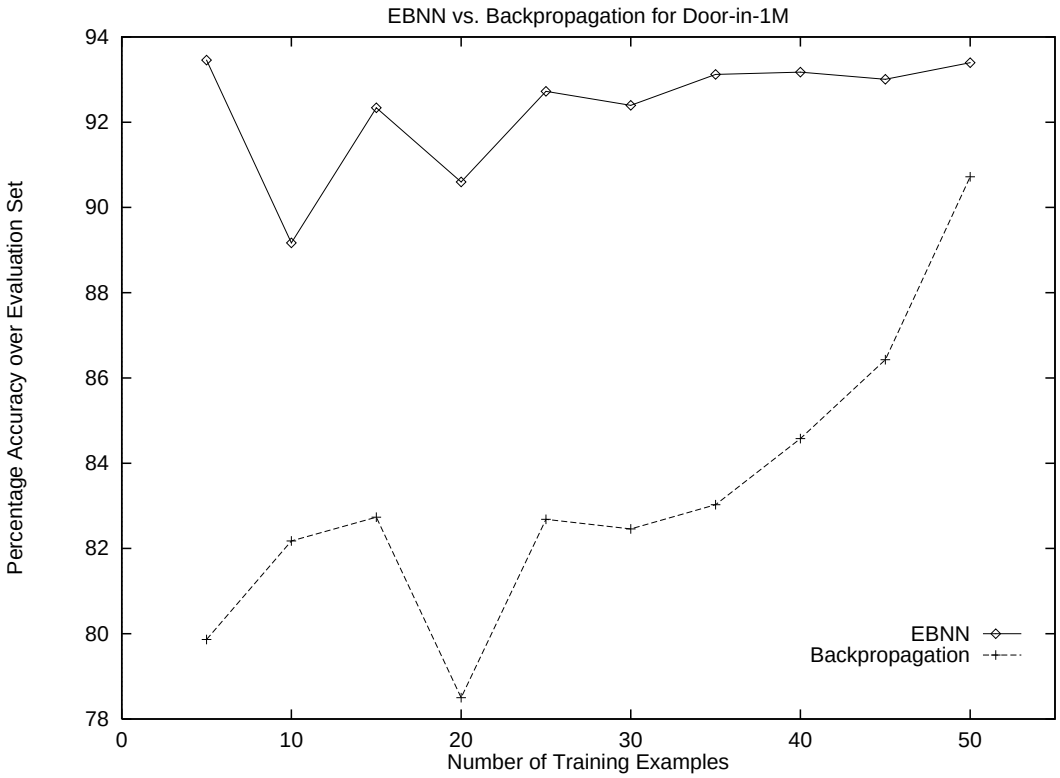



**Target Function:**




Copyright Tom Mitchell, 1994

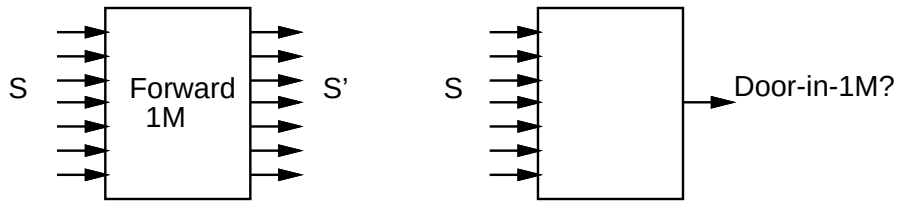
# Results



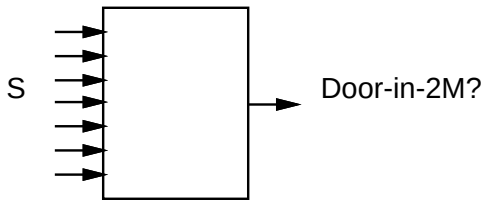

Copyright Tom Mitchell, 1994

# Using Learned Door-in-1M? to Learn Door-in-2M?

**Previously Learned Networks:**




**Target Function:**




Copyright Tom Mitchell, 1994

# Lessons from EBNN

Main idea:

- Previously learned ANN's explain examples of new target function
- Derivatives from explanation capture DT dependencies
- Target ANN weights adjusted fit to target values (inductive) and target derivatives (analytical)
- Outperforms inductive Backpropagation, robust to DT errors

Some limitations:

- ANN domain theory ~propositional

Copyright Tom Mitchell, 1994