Combining Inductive and Analytical Learning

[Read Ch. 12]

[Suggested exercises: 12.1, 12.2, 12.6, 12.7, 12.8]

- Why combine inductive and analytical learning?
- KBANN: Prior knowledge to initialize the hypothesis
- TangetProp, EBNN: Prior knowledge alters search objective
- FOCL: Prior knowledge alters search operators

Inductive and Analytical Learning

Inductive learning	**Analytical learning**
Hypothesis fits data	Hypothesis fits domain the
Statistical inference	Deductive inference
Requires little prior knowledge	Learns from scarce data
Syntactic inductive bias	Bias is domain theory

What We Would Like

Inductive learning ⟷ Analytical learning

Plentiful data
No prior knowledge

Perfect prior knowledge
Scarce data

General purpose learning method:

- No domain theory $\rightarrow$ learn as well as inductive methods
- Perfect domain theory $\rightarrow$ learn as well as PROLOG-EBG
- Accomodate arbitrary and unknown errors in domain theory
- Accomodate arbitrary and unknown errors in training data

Domain theory:

$$\text{Cup} \leftarrow \text{Stable, Liftable, OpenVessel}$$
$$\text{Stable} \leftarrow \text{BottomIsFlat}$$
$$\text{Liftable} \leftarrow \text{Graspable, Light}$$
$$\text{Graspable} \leftarrow \text{HasHandle}$$
$$\text{OpenVessel} \leftarrow \text{HasConcavity, ConcavityPointsUp}$$

Training examples:

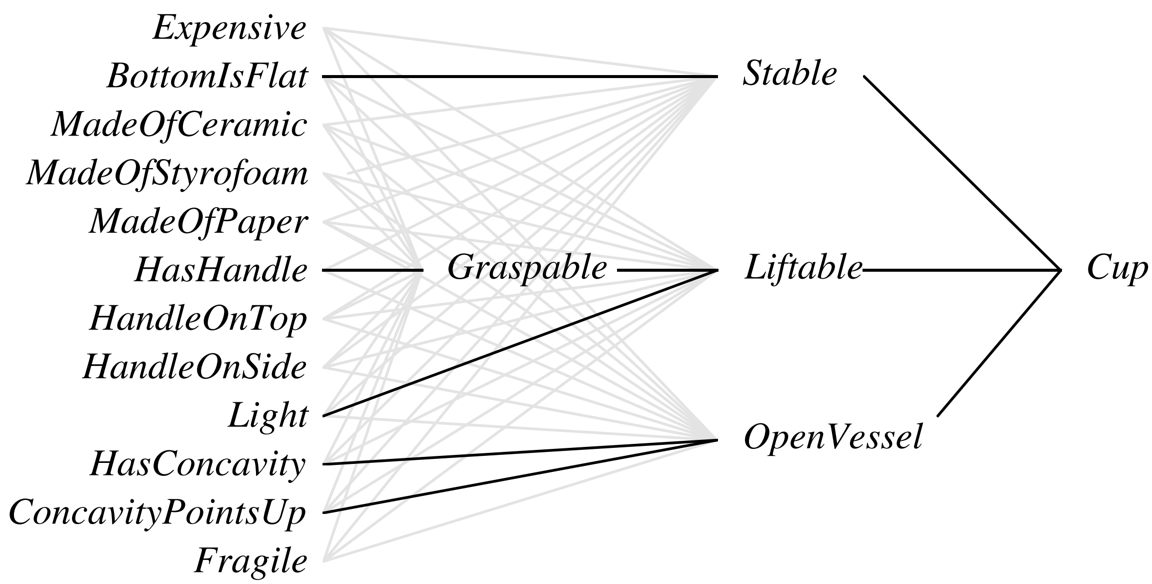
	Cups				Non-Cups					
BottomIsFlat	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$			$\surd$
ConcavityPoints Up	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$		$\surd$	$\surd$		
Expensive	$\surd$		$\surd$				$\surd$		$\surd$	
Fragile	$\surd$	$\surd$			$\surd$	$\surd$		$\surd$		$\surd$
HandleOnTop					$\surd$		$\surd$			
HandleOnSide	$\surd$			$\surd$					$\surd$	
HasConcavity	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$		$\surd$	$\surd$	$\surd$	$\surd$
HasHandle	$\surd$			$\surd$	$\surd$		$\surd$		$\surd$	
Light	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$		$\surd$	
MadeOfCeramic	$\surd$				$\surd$		$\surd$	$\surd$		
MadeOfPaper				$\surd$					$\surd$	
MadeOfStyrofoam		$\surd$	$\surd$			$\surd$				$\surd$

KBANN

KBANN (data D, domain theory B)

1. Create a feedforward network h equivalent to B
2. Use BACKPROP to tune h to fit D

Neural Net Equivalent to Domain Theory



Creating Network Equivalent to Domain Theory

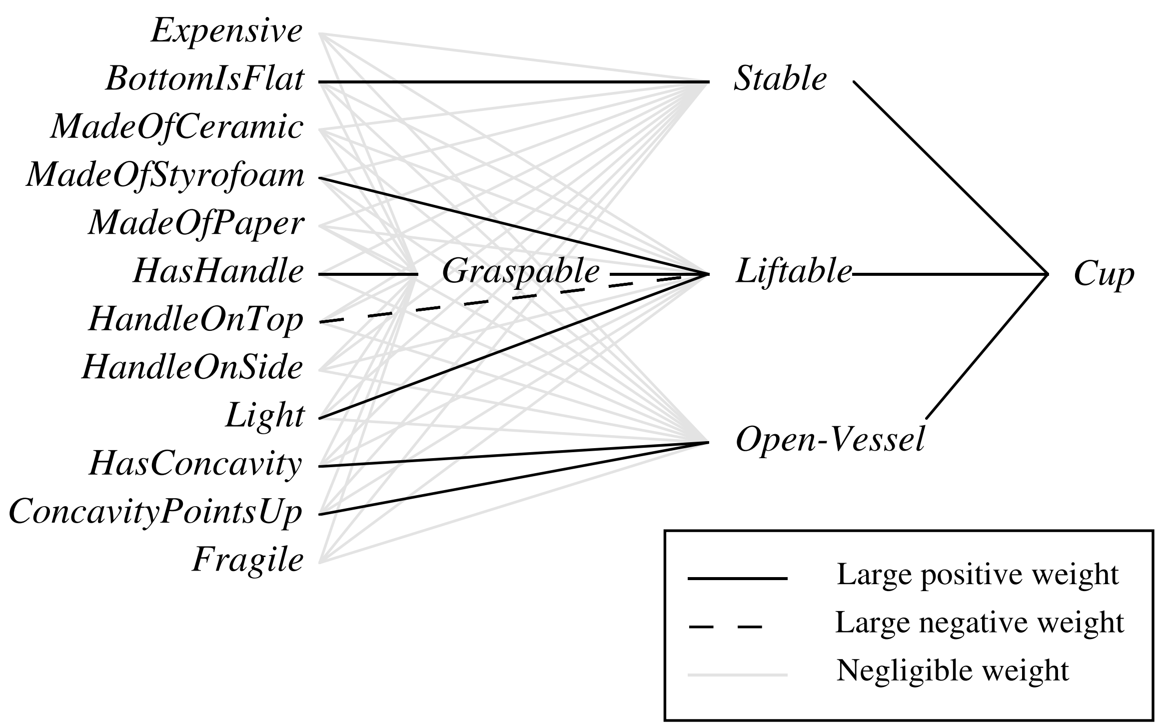
Create one unit per horn clause rule (i.e., an AND unit)

- Connect unit inputs to corresponding clause antecedents
- For each non-negated antecedent, corresponding input weight $w \leftarrow W$, where W is some constant
- For each negated antecedent, input weight $w \leftarrow -W$
- Threshold weight $w_0 \leftarrow -(n - .5)W$, where n is number of non-negated antecedents

Finally, add many additional connections with near-zero weights

$$Liftable \leftarrow Graspable, \neg Heavy$$

Result of refining the network



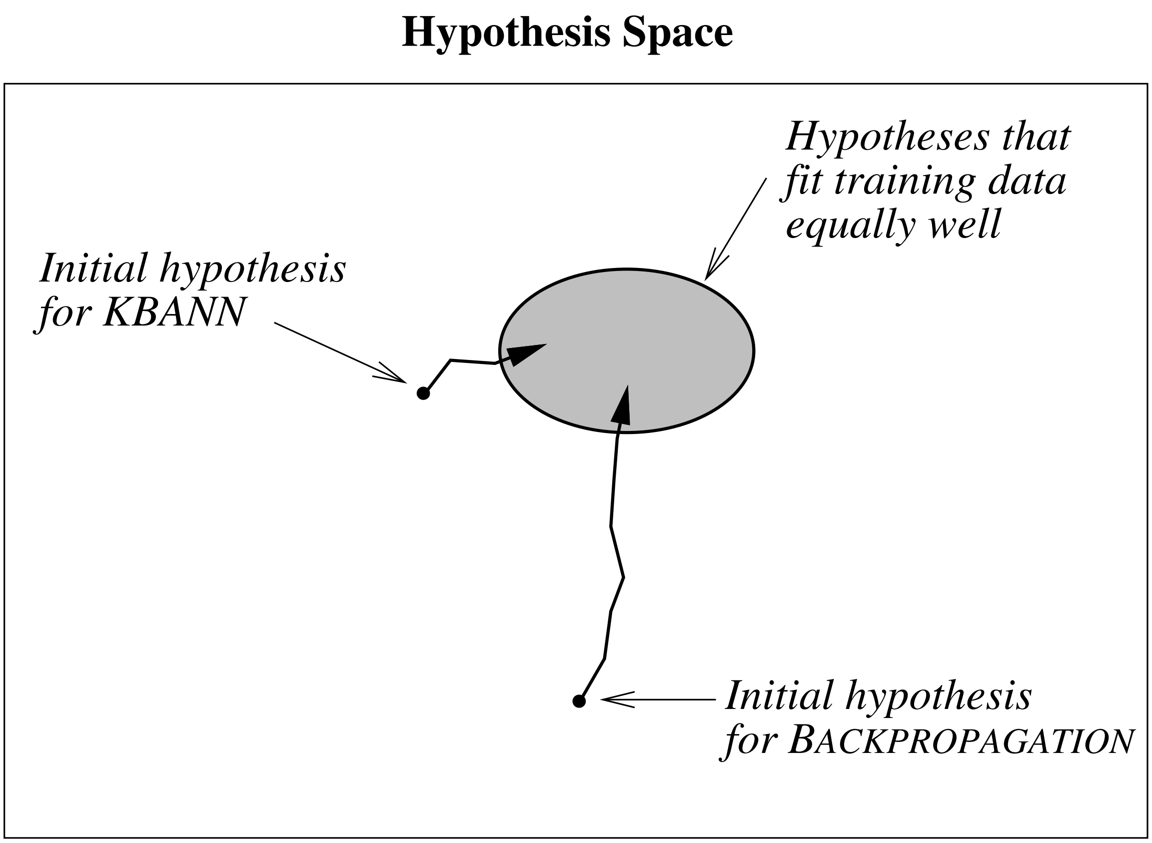
KBANN Results

Classifying promoter regions in DNA
leave one out testing:

- Backpropagation: error rate 8/106
- KBANN: 4/106

Similar improvements on other classification, control tasks.

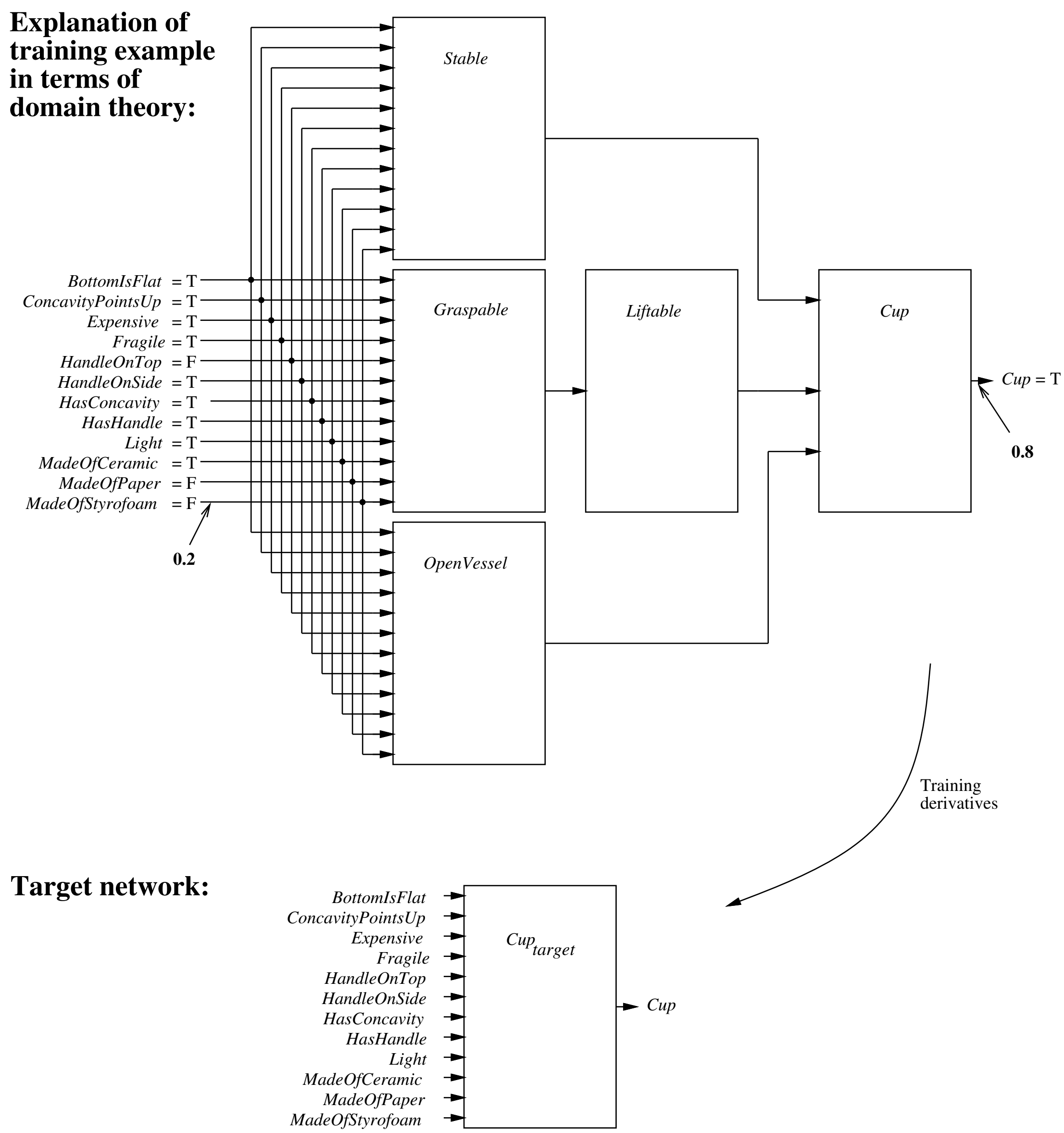
Hypothesis space search in KBANN



EBNN

Key idea:

- Previously learned approximate domain theory
- Domain theory represented by collection of neural networks
- Learn target function as another neural network


Explanation of training example in terms of domain theory:
Stable
Graspable
Liftable
Cup
OpenVessel
BottomIsFlat = T
ConcavityPointsUp = T
Expensive = T
Fragile = T
HandleOnTop = F
HandleOnSide = T
HasConcavity = T
HasHandle = T
Light = T
MadeOfCeramic = T
MadeOfPaper = F
MadeOfStyrofoam = F
0.2
Cup = T
0.8
Training derivatives
Target network:
BottomIsFlat
ConcavityPointsUp
Expensive
Fragile
HandleOnTop
HandleOnSide
HasConcavity
HasHandle
Light
MadeOfCeramic
MadeOfPaper
MadeOfStyrofoam
Cup_target
Cup

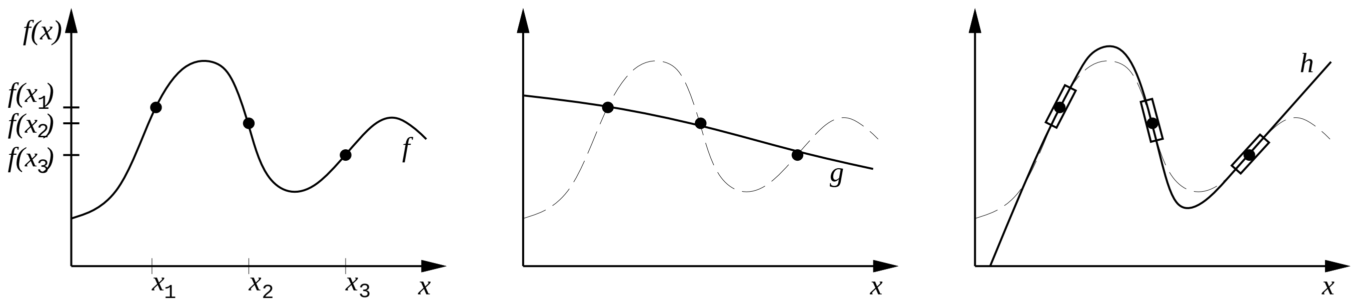
Modified Objective for Gradient Descent

$$E = \sum_i \left[(f(x_i) - \hat{f}(x_i))^2 + \mu_i \sum_j \left(\frac{\partial A(x)}{\partial x^j} - \frac{\partial \hat{f}(x)}{\partial x^j} \right)^2_{(x=x_i)} \right]$$

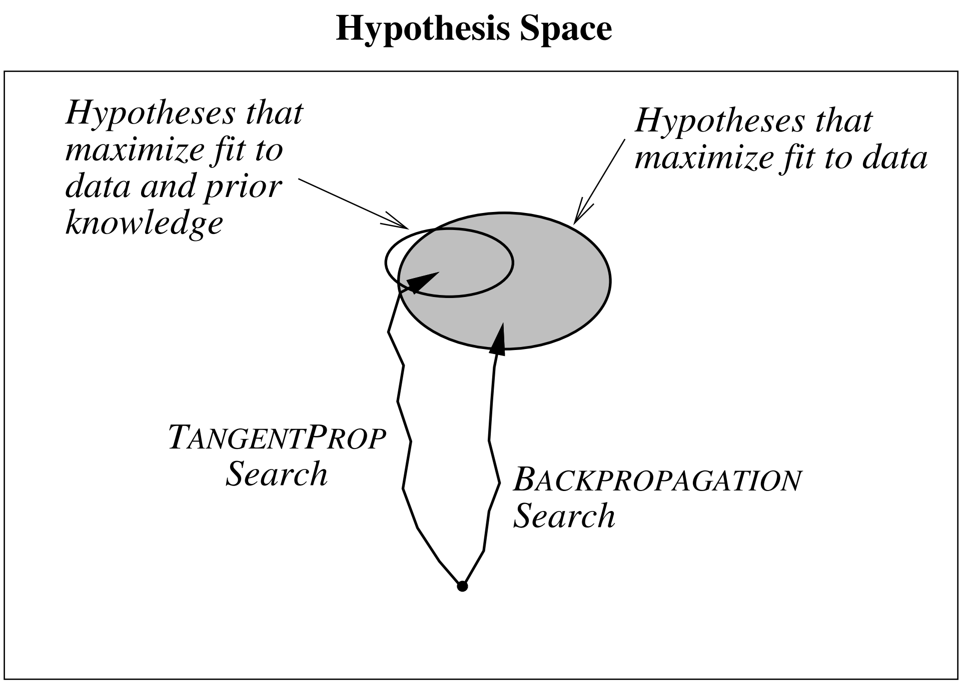
where

$$\mu_i \equiv 1 - \frac{|A(x_i) - f(x_i)|}{c}$$

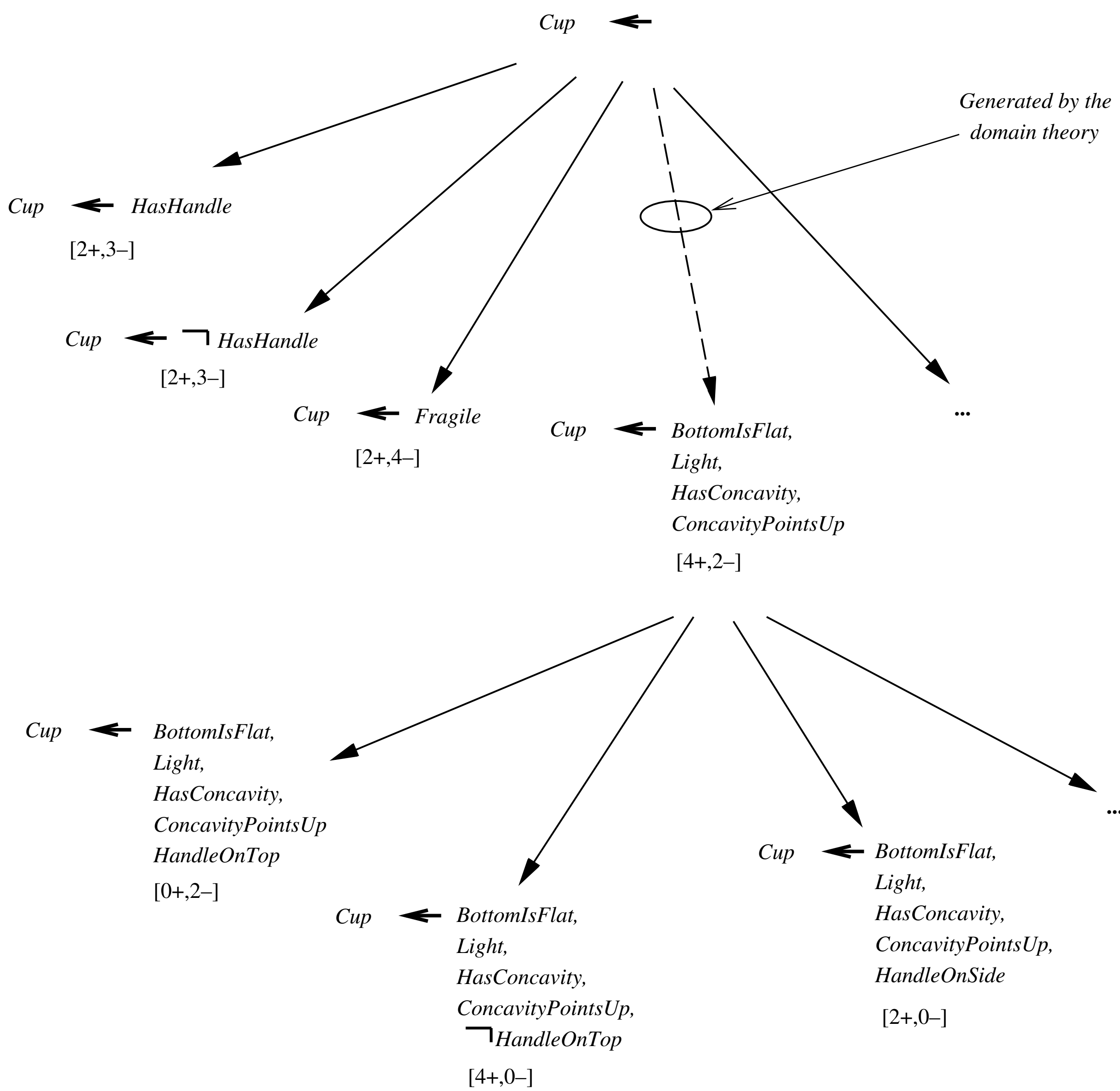
- $f(x)$ is target function
- $\hat{f}(x)$ is neural net approximation to $f(x)$
- $A(x)$ is domain theory approximation to $f(x)$


f(x)
f(x1)
f(x2)
f(x3)
f
x1
x2
x3
x
g
x
h
x

Hypothesis Space Search in EBNN



Search in FOCL



FOCL Results

Recognizing legal chess endgame positions:

- 30 positive, 30 negative examples
- FOIL: 86%
- FOCL: 94% (using domain theory with 76% accuracy)

NYNEX telephone network diagnosis

- 500 training examples
- FOIL: 90%
- FOCL: 98% (using domain theory with 95% accuracy)