Criterionusedtodeterminethecorrectfinaltreesize

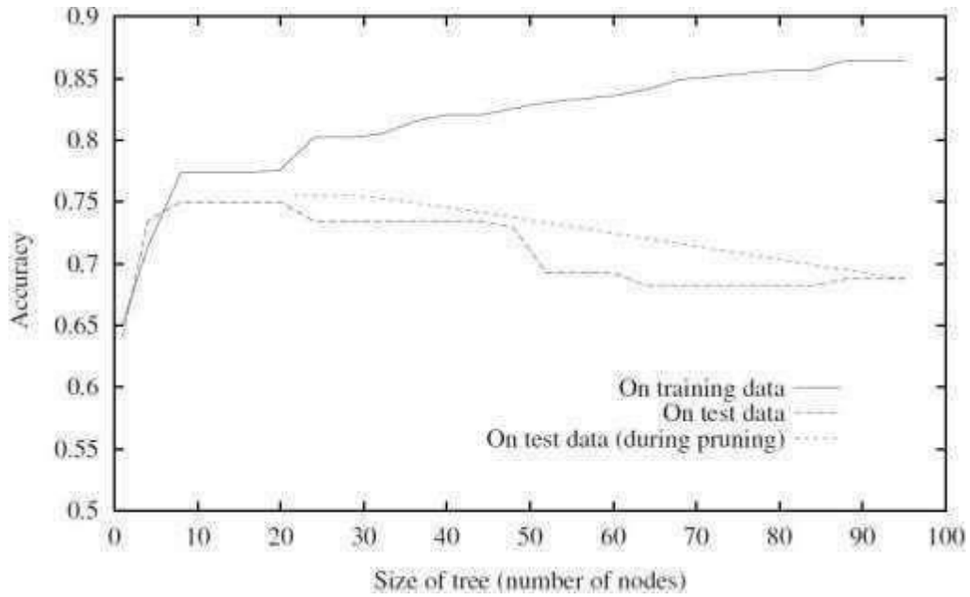
- Useaseparatesetofexamples,distinctfromthetrainingexamples,toevaluatetheutilityof post- pruningnodesfrom thetree
- Usealltheavailabledatafortraining,butapply*astatisticaltest*toestimatewhetherexpandi ngpruning)a particularnodeis likelyto produceanimprovementbeyond thetrainingset (or
- Usemeasureofthecomplexityforencodingthetrainingexamplesandthedecisiontree,h altinggrowth of the tree when this encoding size is minimized. This approach istheMinimumDescriptionLength called
  - *MDL–*

Minimize:size(tree)+size(misclassifications(tree))

Reduced-ErrorPruning

- ***Reduced-error pruning***, is to consider each of the decision nodes in the tree to becandidates for pruning
- ***Pruning*** a decision node consists of removing the subtree rooted at that node,making it a leaf node, and assigning it the most common classification of thetrainingexamples affiliated with that node
- Nodesareremovedonlyiftheresultingprunedtreeperformsnoworsethan-theoriginaloverthevalidationset.
- Reduced error pruning has the effect that any leaf node added due to coincidentalregularitiesinthetrainingsetislikelytobeprunedbecausethesesamecoincidencesare unlikelyto occur inthe validationset

Theimpactofreduced-errorpruningontheaccuracyofthedecisiontreeisillustratedinbelowfigure



- The additional line in figure shows ***accuracy over the test examples*** as the tree is pruned. Whenpruning begins, the tree is at its maximum size and lowest accuracy over the test set. As pruningproceeds,thenumberofnodesisreducedandaccuracyoverthetestsetincreases.
- The available data has been split into three subsets: the training examples, the validation examplesused for pruning the tree, and a set of test examples used to provide an unbiased estimate ofaccuracyoverfutureunseen examples.Theplotshowsaccuracyoverthetrainingandtestsets.

Summary

- Statisticaltheoryprovidesabasisforestimatingthetrueerror ($error_D h$)ofhypothesish,basedonitsobservederror($error_S h$) overasampleSofdata.
- In general, theproblem of estimatingconfidence intervals isapproachedbyidentifyingtheparametertobeestimated($error_D h$)and anestimator($error_S h$)forthisquantity.
- Becausetheestimatorisarandomvariableitcanbecharacterisedbyt heprobabilitydistributionthatgovernsitsvalue.

unbiasedestima$_2$tor,theobservedvalueoftheestimatorislikelyto varyfromoneexperimenttoanother.

Thevarianceofthedistributiongoverningtheestimateisl
ikelyto orcharacteriseshowwidelythis

Confidenceintervalscanthenbecalculatedbydeterminingthein
terval that containsthedesired probabilitymass under
thisdistribution.

Acause ofestimationerroristhevarianceintheestimate.Evenwithan

Prodigy and Explanation Based Learning

Prodigy defines a set of target concepts to learn, e.g., which operator given the current state takes you to the goal state?

An example of a rule learned by Prodigy in the block-stacking problem is:

IF One subgoal to be solved is On(x,y) AND
One subgoal to be solved is On(y,z)
THEN Solve the subgoal On(y,z) before On(x,y)

Prodigy and Explanation Based Learning

The rationale behind the rule is that it would avoid a conflict when stacking blocks.

Prodigy learns by first encountering a conflict, then explaining the reason for the conflict and creating a rule like the one above.

Experiments show an improvement in efficiency by a factor of two to four.

Problems with EBL

- ✓ The number of control rules that must be learned is very large.
- ✓ If the control rules are many, much time will be spent looking for the best rule.

Utility analysis is used to determine what rules to keep and what rules to forget.

Prodigy:

328 possible rules ⟶ 69 pass test ⟶ 19 were retained

Problems with EBL

- ✓ Another problem with EBL is that it is sometimes intractable to create an explanation for the target concept.

  For example, in chess, learning a concept like:
  "states for which operator A leads to a solution"
  The search here grows exponentially.

Summary

- Different from inductive learning, analytical learning looks for a hypothesis that fit the background knowledge and covers the training examples.

- Explanation based learning is one kind of analytical learning that divides into three steps:
  a. Explain the target value for the current example
  b. Analyze the explanation (generalize)
  c. Refine the hypothesis

Summary

- Prolog-EBG constructs intermediate features after analyzing examples.
- Explanation based learning can be used to find search control rules.
- In all cases we depend on a perfect domain theory.

Machine Learning

Chapter 12. Combining Inductive and Analytical Learning

Tom M. Mitchell

Inductive and Analytical Learning

Inductive learning

- Hypothesis fits data
- Statistical inference
- Requires little prior knowledge
- Syntactic inductive bias

Analytical learning

- Hypothesis fits domain the
- Deductive inference
- Learns from scarce data
- 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 → learn as well as inductive methods
- Perfect domain theory → learn as well as Prolog-EBG
- Accomodate arbitrary and unknown errors in domain theory
- Accomodate arbitrary and unknown errors in training data

Domain theory:

Cup ← Stable, Liftable, Open Vessel

Stable ← BottomIsFlat

Liftable ← Graspable, Light

Graspable ← HasHandle

Open Vessel ← HasConcavity, ConcavityPointsUp

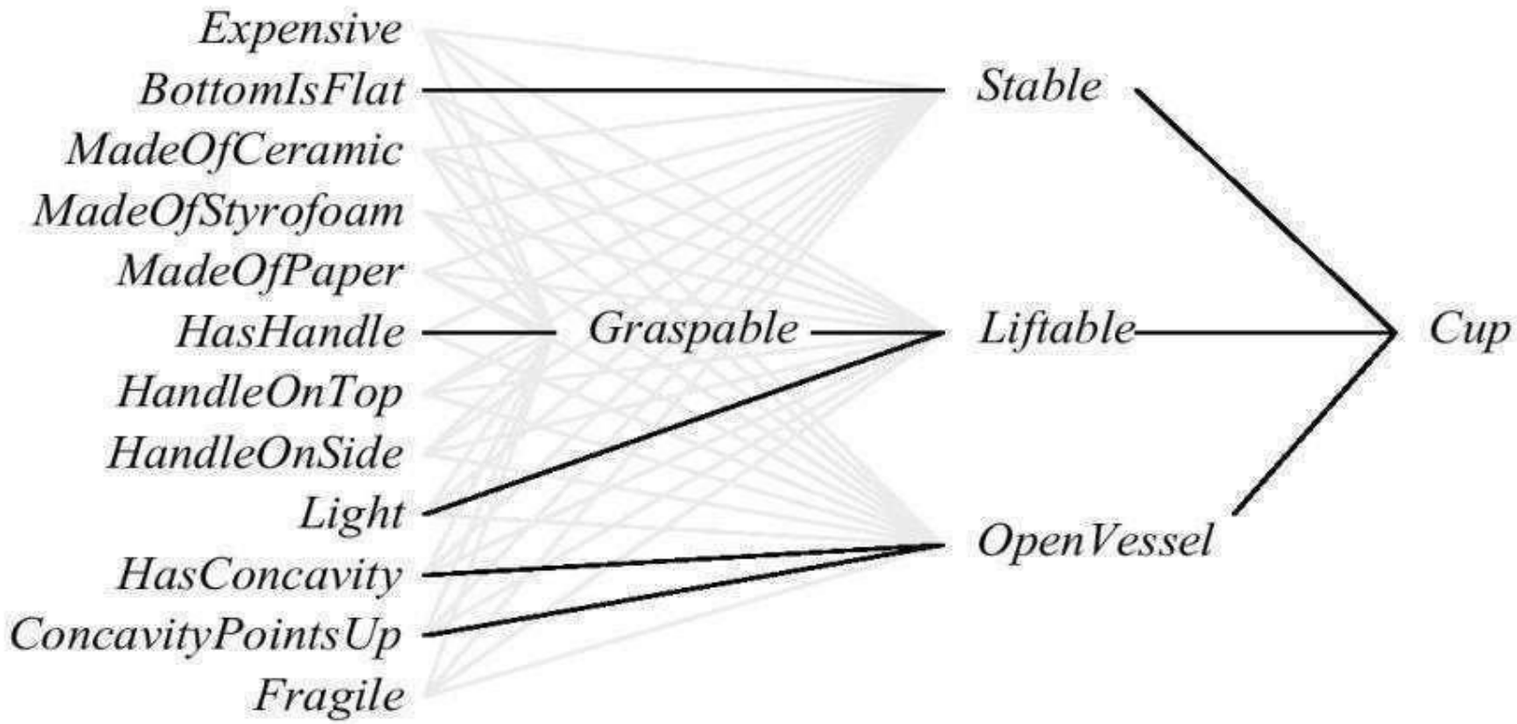
Training examples:

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

KBANN

- KBANN (data D, domain theory B)
  1. Create a feedforward network h equivalent to B
  2. Use BACKPROP to tune h to t D

Neural Net Equivalent to Domain Theory



Creating Network Equivalent to Domain Theory

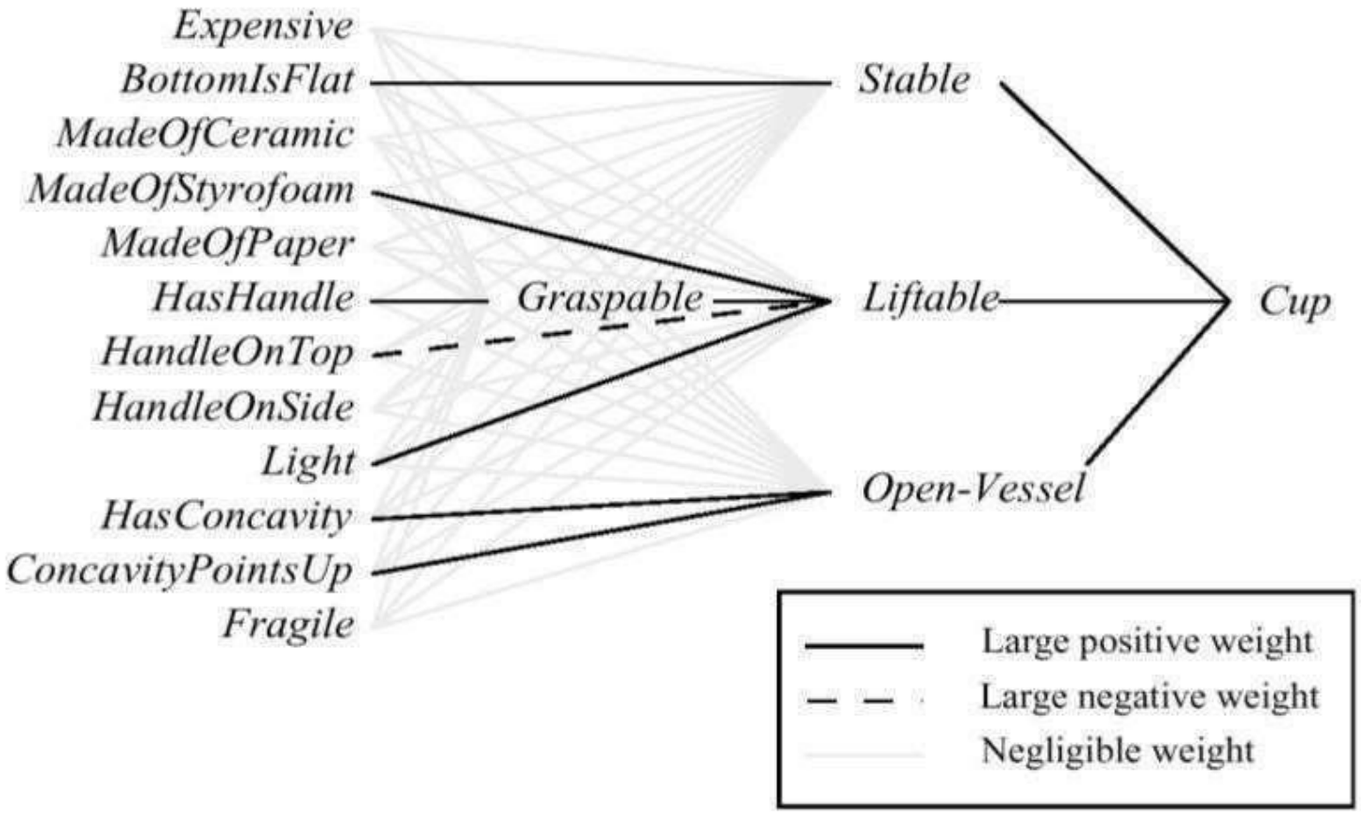
Create one unit per horn clause rule (i.e., an AND unit)

- 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



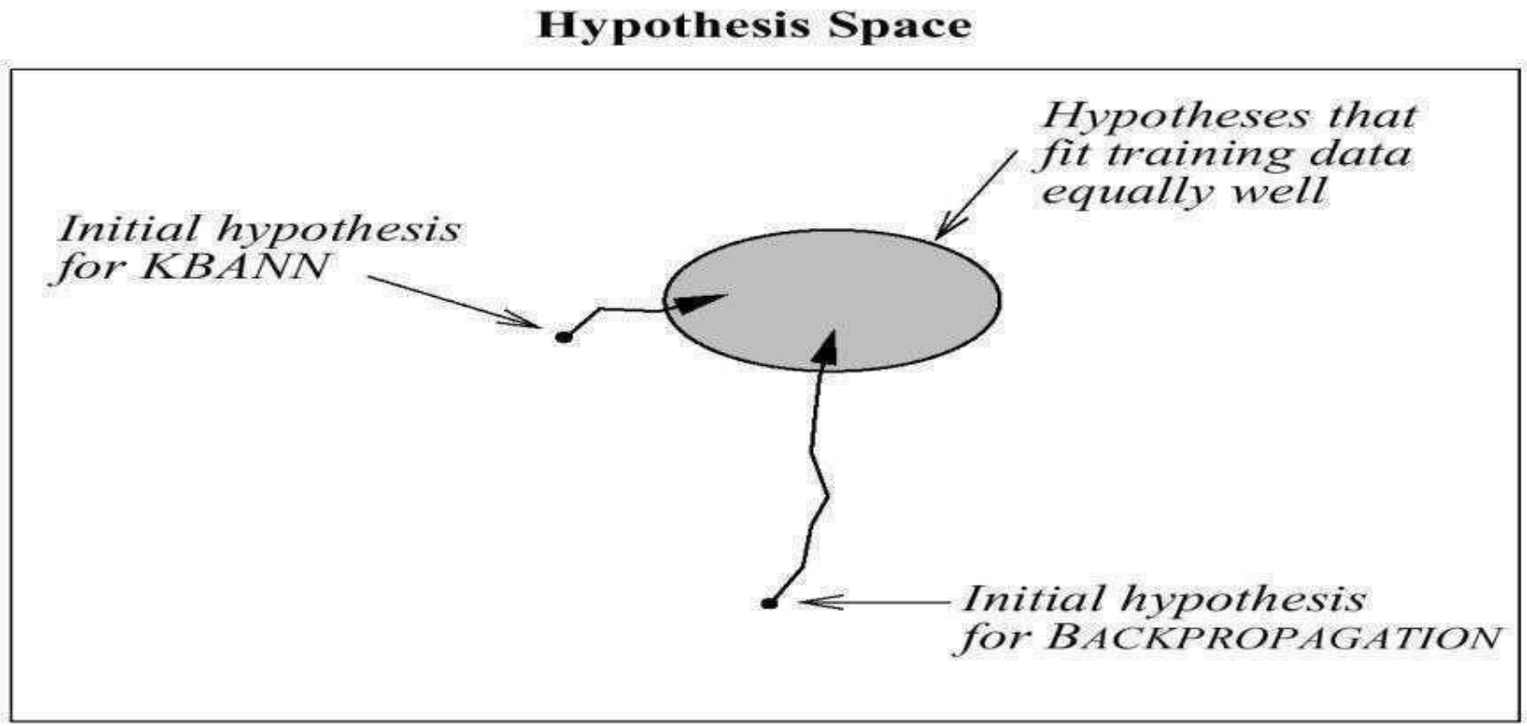
KBANN Results

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

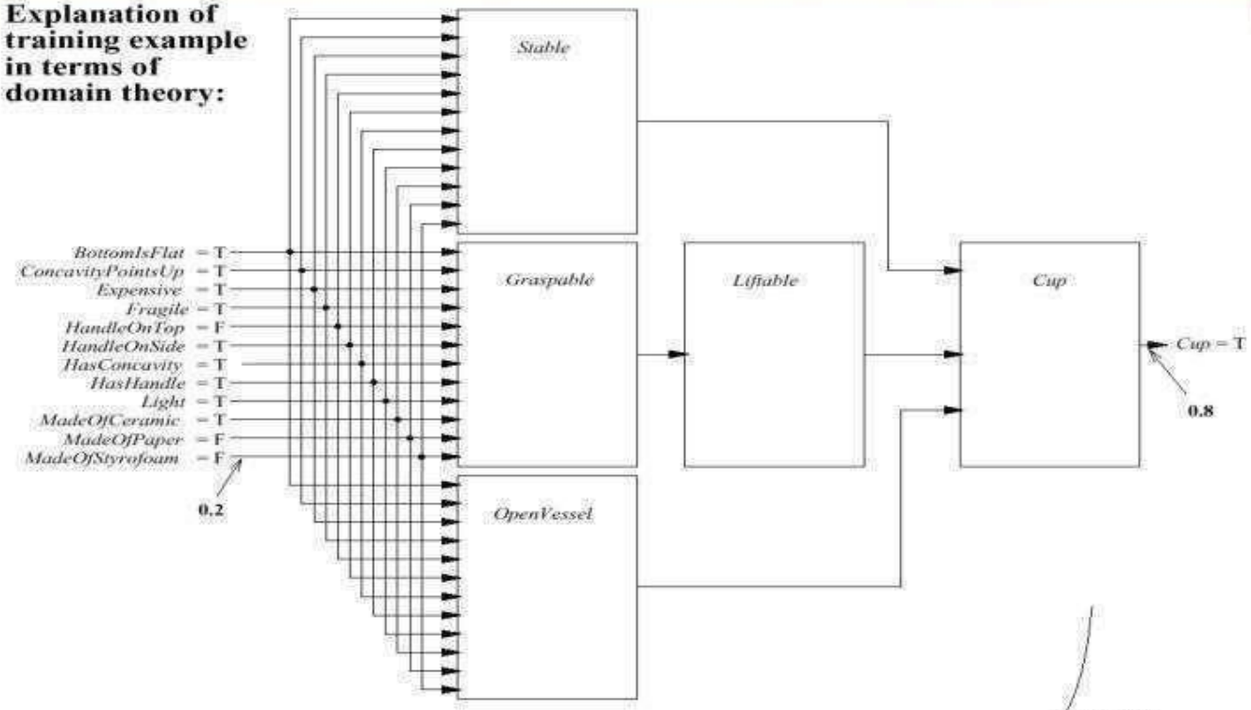


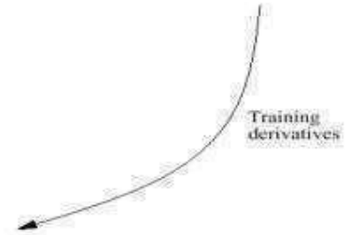
EBNN

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:





Target network:

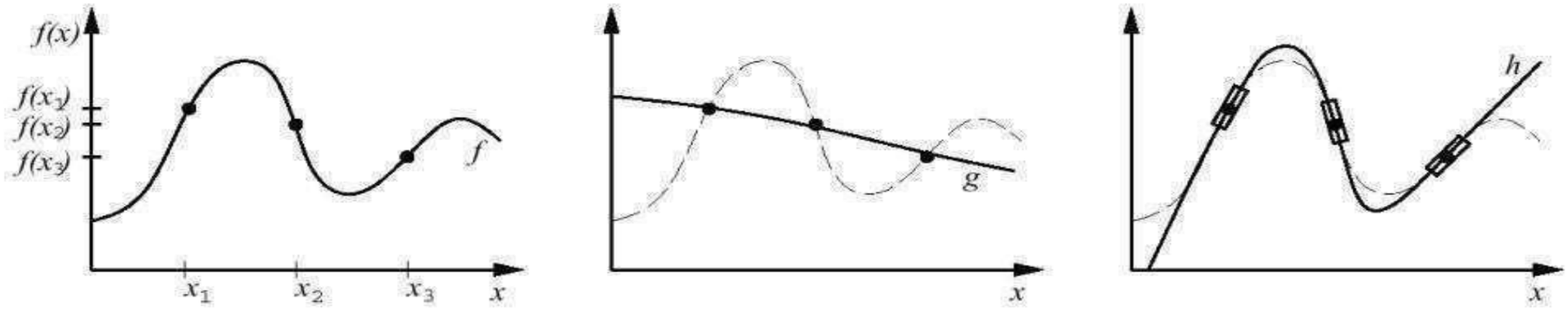
Modified Objective for Gradient Descent

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

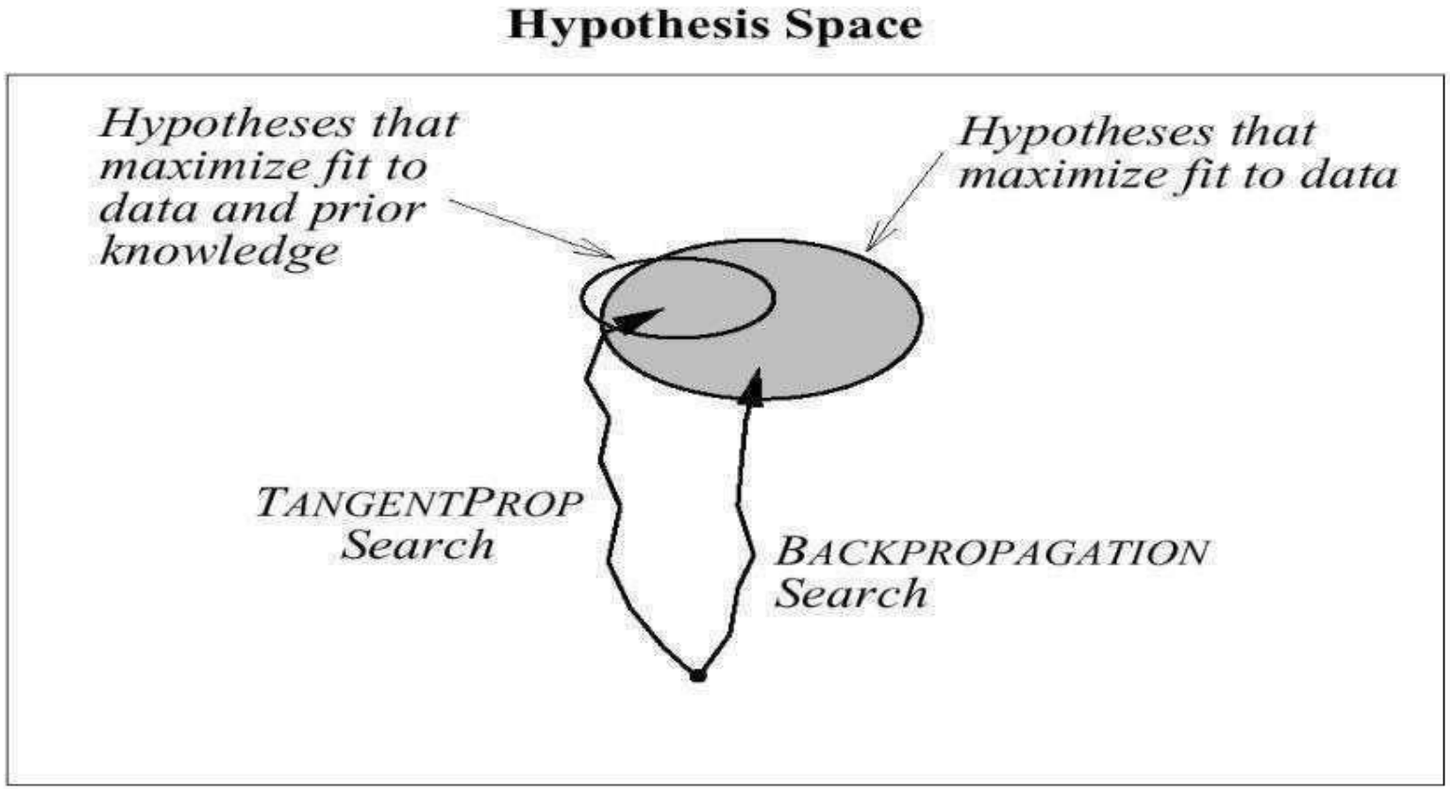
where

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

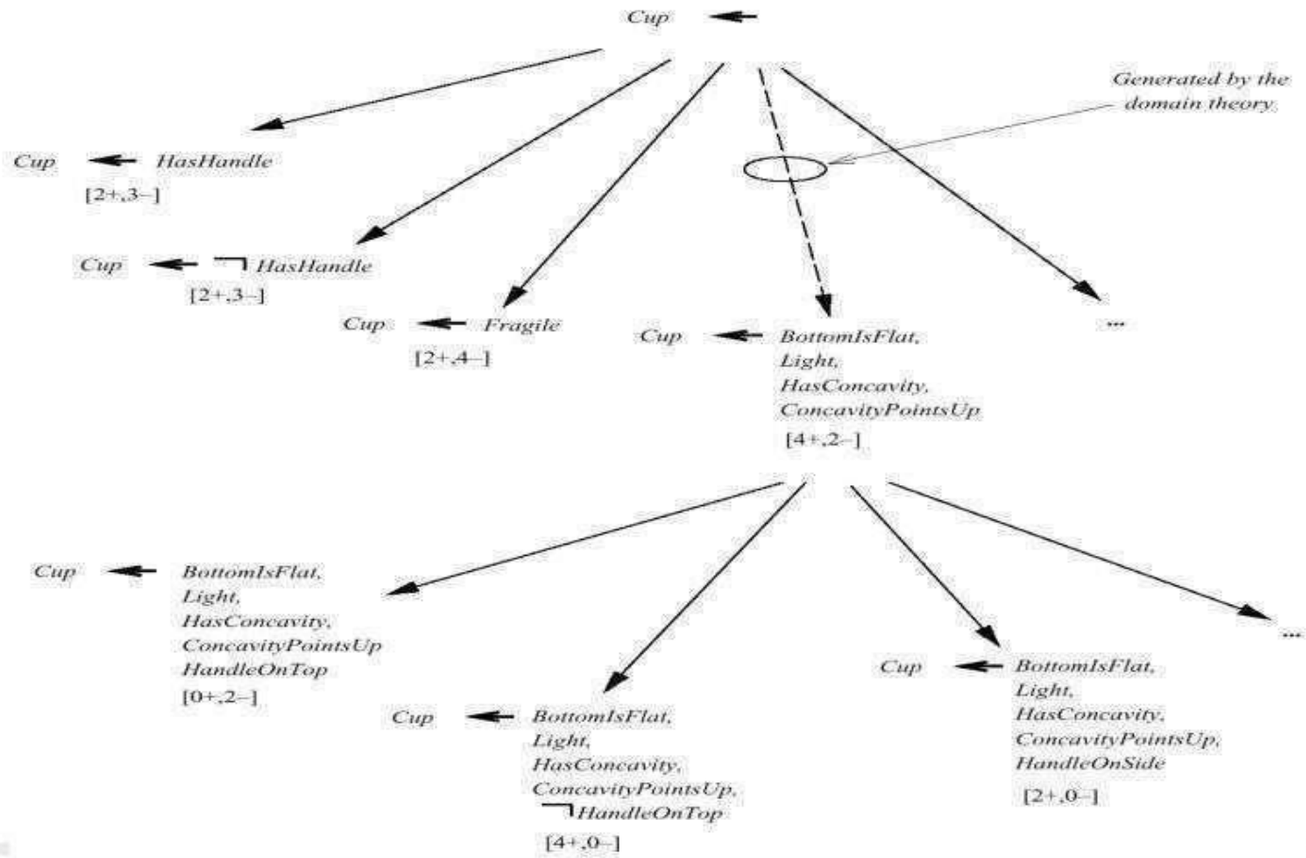
- $f(x)$ is target function
- $\hat{f}(x)$ is neural net approximation to $f(x)$
- $A(x)$ is domain theory approximation to $f(x)$


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



THANKYOU