Unit-5

Analytical Learning

① Introduction (Inductive-Analytical approaches to learning)

- Inductive learning requires a certain no. of training problems to achieve a given level of accuracy.
- In this learning method, the generalization is done by observing training examples and identifying their features.
- On the basis of the features they are classified as +ve and -ve training examples.

→ learning algorithm → hypothesis

dataset (must be large)

Analytical Learning:

- Analytical learning uses prior knowledge and deductive reasoning to increase the training examples.
- It contains domain theory which explains the training examples. So, it's called as explanation based learning.
- By using these explanations, analytical learning is more accurate than inductive

→ Learning Algorithm → hypothesis

(dataset)

prior knowledge

Inductive - Analytical Learning

- In this learning method we combine both analytical learning and inductive learning in order to gain the benefits of both approaches.
- In this method, the domain theory contains the explanations of training examples by the induction method.
- These explanations are used to perform predictions.

Learning problem:

Given: A set of training examples 'D' possibly containing an error.
- A domain theory 'B' containing errors.
- A space of candidate hypothesis 'H'

Determine: A hypothesis that best fits the training example and domain theory.

→ A best fit hypothesis is a hypothesis which contain zero errors.

→ There are 2 types of errors:
① Errors due to training examples.
② Errors due to domain theory

→ Total errors can be given as

$$\underset{h \in H}{\operatorname{argmin}} \; k_D \, error_D(h) + k_B \, error_B(h)$$

Here, k_D, k_B are weights that are present for training example and domain theory.
They can be calculated by using Bayesian method.

	Inductive Learning	Analytical Learning
Goal:	Hypothesis fits data	Hypothesis fits domain theory
Justification:	Statistical inference	Deductive inference
Advantage:	Requires little prior knowledge	Learns from scarce data

Comparision (Inductive & Analytical):

- Purely analytical learning methods offer the advantage of generalising more accurately from less data by using prior knowledge to guide learning. However, they can be misled when given incorrect or insufficient prior knowledge.
- Purely inductive methods offer the advantage that they require no explicit prior knowledge & learn regularities based solely on training data. However, they can fail when given insufficient training data, & can be misled by the implicit inductive bias they must adopt in order to generalize beyond the observed data.

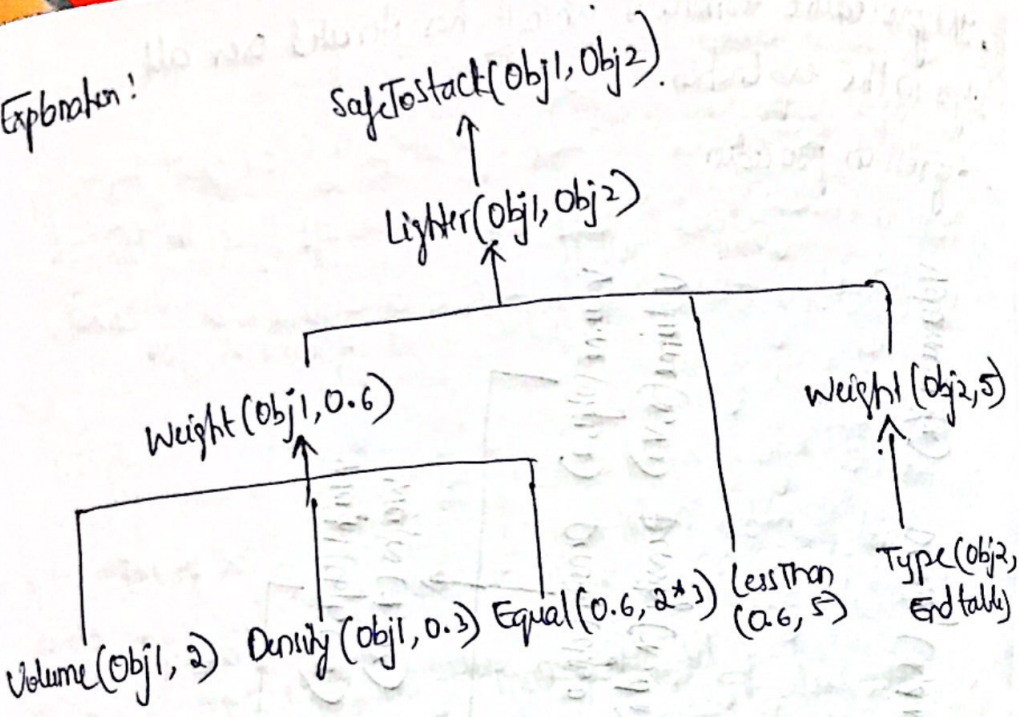
Hypothesis Space Search: We can characterise most learning methods as search algorithms by describing the hypothesis space H they search, the initial hypothesis 'h0' at which they begin their search, the set of search operators 'O' that define individual search steps, and the goal criterion 'G' that specifies the search objective.

② Learning with perfect domain theories:

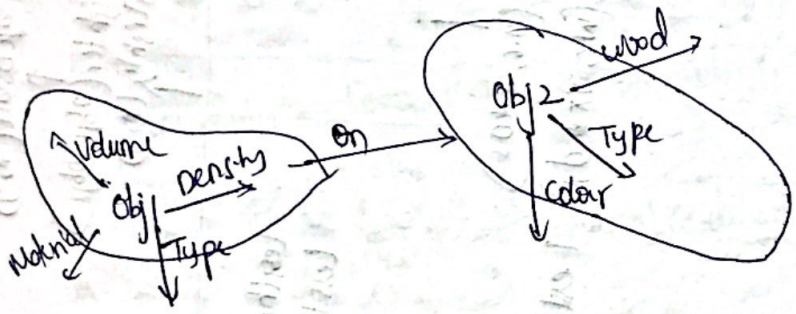
② PROLOG-EBG:

- A domain theory is said to be correct, if each of its assertions is a truthful statement about the world.
- A domain theory is said to be complete with respect to a given target concept and instance space, if the domain theory covers every positive example in the instance space.
- PROLOG-EBG algorithm is a sequentially covering algorithm that considers the training data incrementally.
- For each new positive training example that is not yet covered by a learned Horn clause, it forms a new Horn clause by:
  (1) Explaining the training example.
  (2) Analysing this explanation to determine whether it's an appropriate generalisation or not.
  (3) Refine the current hypothesis by adding a new Horn clause rule to cover this positive example, as well as other similar instances.
- PROLOG-EBG computes the most general rule by computing the weakest preimage (initial assertion) of the explanation.
- PROLOG-EBG computes the weakest preimage of the target concept with respect to the explanation, using a general procedure called regression.

Explanation:

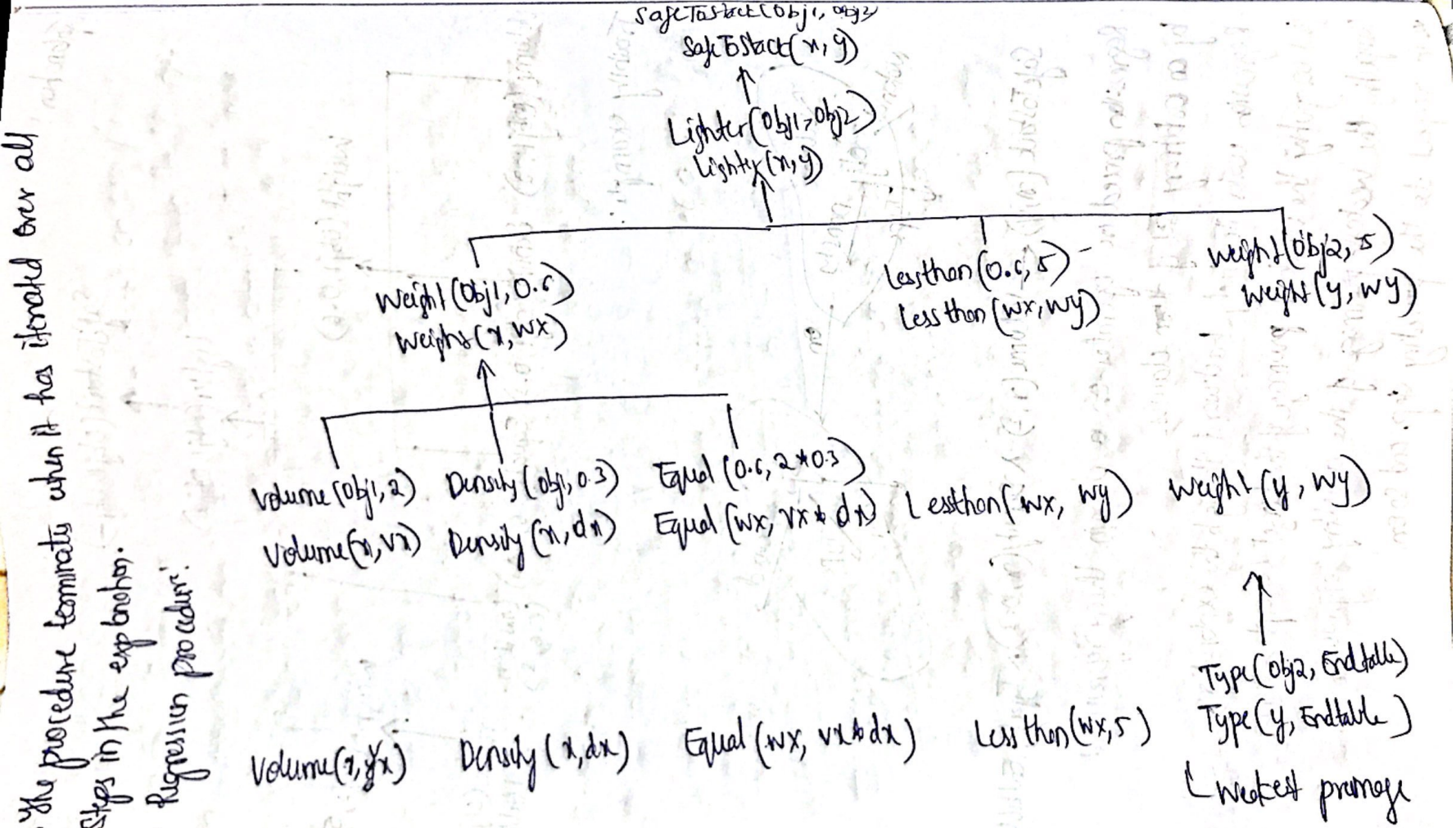


Trainbly example:



$SafeToStack(x,y) \leftarrow Volume(x,3) \wedge Density(x,0.3) \wedge Type(y,Endtable)$

- Regression procedure operates on a domain theory represented by an arbitrary set of Horn clauses.
- Regression works iteratively backward through the explanation, first computing the weakest preimage of the target concept, then computing the weakest preimage of the resulting expressions with respect to the preceding step, and so on.



- The procedure terminates when it has iterated over all steps in the explanation.
- Regression procedure.

PROLOG-EBG!

PROLOG-EBG (TargetConcept, TrainingExamples, DomainTheory)

- LearnedRules ← {}
- Pos ← the positive examples from TrainingExamples
- foreach PositiveExample in Pos that is not covered by LearnedRules, do

1. Explain:
   - Explanation ← an explanation (proof) in terms of the DomainTheory that PositiveExample satisfies the TargetConcept

2. Analyze:
   - SufficientConditions ← the most general set of features of PositiveExample sufficient to satisfy the TargetConcept according to the Explanation.

3. Refine:
   - LearnedRules ← LearnedRules + NewHornClause, where NewHornClause is of the form

     TargetConcept ← SufficientConditions

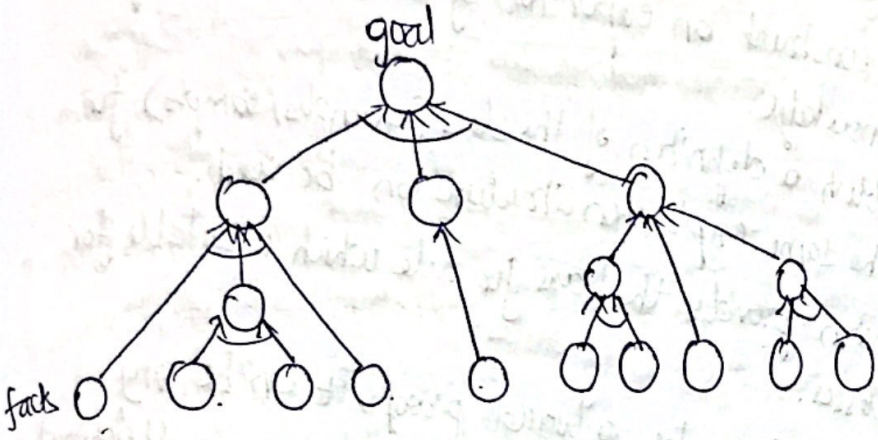
- Returned learned rules.

③ Explanation based learning (EBL):

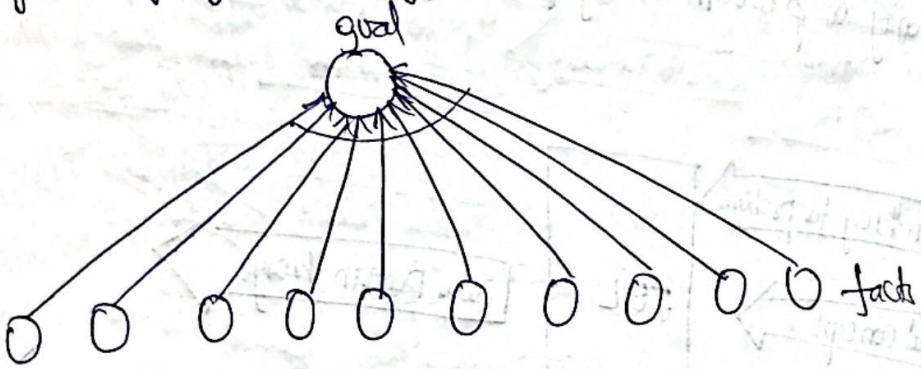
- Exploration-Based learning (EBL) system accepts an example (i.e a training example) and explains what it learns from the example.
- The objective of EBL is to formulate a generalised exploration of the goal concept.
- It has two steps:
  - first, explain method
  - second, generalise method
- During the first step, the domain theory is used to prune away all the unimportant aspects of training examples with respect to the goal concept.
- The second step is to generalise the exploration as far as possible while still describing the goal concept.
- EBL requirements:
  (1) A formal statement of the goal concept to be learned
  (2) At least one positive training example of the concept
  (3) Domain theory, which relates to the concept and training examples.
  (4) Criteria for concept operation
- Exploration Based learning (EBL) is a principled method for exploiting available domain knowledge to improve supervised learning.

Standard approach to EBL:

An exploration (detailed proof of goal)



After learning (go directly from facts to solution)



. EBL computes a generalisation of the training example, to describe the goal concept.

. It also satisfies the operational criteria

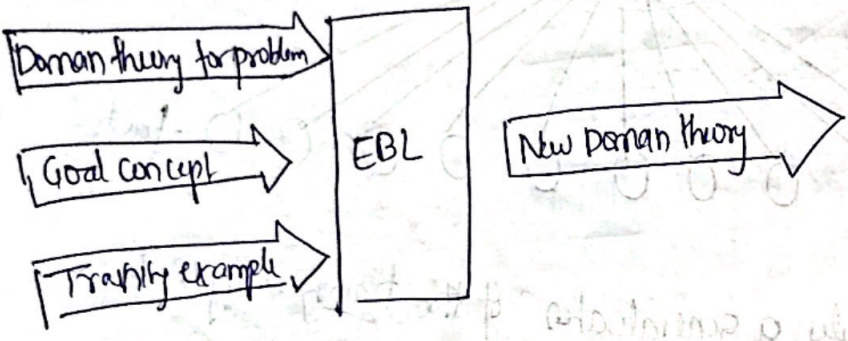
EBL- Memorisation:

. Once the rule is identified then that should be memorised.

. Whenever it finds an example, immediately it will apply the same rule (taken from cache memory) on the example.

Extracting general rules from examples:

- The basic idea behind EBL is
  - first to construct an explanation of the observation using prior knowledge.
  - to establish a definition of the class (examples/samples) for which the same explanation structure can be used.
- This definition provides the basis for rule which is suitable for all the classes.
- The "explanation" can be a logical proof, it can be any reasoning or problem solving process, whose steps are well defined.



Domain theory:
Set of rules and facts that are used to explain how the example is a case of the goal concept.

Goal concept:
A high-level description of what program is supposed to learn.

Training Example:
What the learning sees in the world.

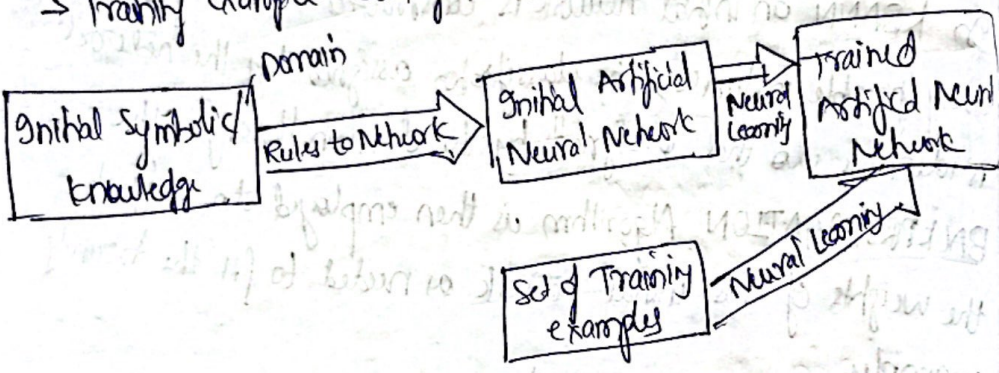
④ Using prior knowledge to initialise the hypothesis

KBANN Algorithm: (Combining Inductive & Analytical Learning)

- One approach to perfectly fit the domain theory using prior knowledge is to initialise the hypothesis to perfectly fit the domain theory, then inductively refine this initial hypothesis as needed to fit the training data. This approach is used by the KBANN (Knowledge Based Artificial Neural Network) to learn artificial neural network.
- In KBANN an initial network is constructed so that for every possible instance, the classification assigned by the network is identical to that assigned by the domain theory. The BACKPROPOGATION Algorithm is then employed to adjust the weights of this initial network as needed to fit the training examples.
- The KBANN algorithm clarifies the initialise-the-hypothesis approach using domain theories. It assumes a domain theory represented by a set of propositional non-recursive Horn clauses.
- Given:
  - A set of training examples.
  - A domain theory consisting of non-recursive, propositional Horn clauses.
- Determine:
  - An artificial neural network that fits the training examples.

- The two stages of KBANN algorithm are first to create an artificial neural network that perfectly fits the domain theory and second, to use the BACKPROPOGATION algorithm to refine this initial network to fit the training examples.

- KBANN is a hybrid learning system (combine both domain knowledge + training examples)
  - → Domain knowledge - set of rules
  - → Training example - training set of neural network.



- KBANN algorithm Combination
  1) rules-to-network algorithm.
  2) Refiner algorithm

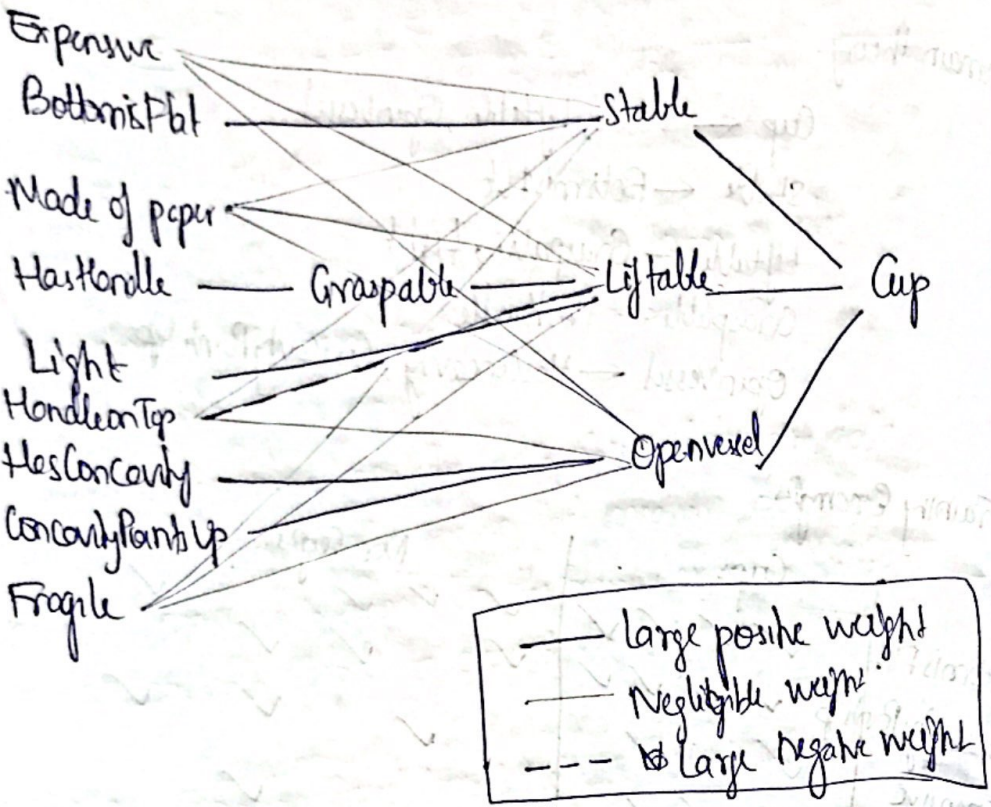
- Ex: Learning Task: Identify Cup

Domain theory:

Cup ← Stable, Liftable, OpenVessel
Stable ← BottomIsFlat
Liftable ← Graspable, Light
Graspable ← HasHandle
OpenVessel ← HasConcavity, ConcavityPointsUp

Training Examples:

	Cups				Non Cups			
BottomIsFlat	✓	✓	✓	✓	✓	✓	✓	✓
ConcavityPointsUp	✓	✓	✓	✓	✓		✓	✓
Expensive	✓		✓			✓	✓	✓
Fragile	✓	✓	✓		✓		✓	
~~Stable~~ Light	✓	✓	✓	✓	✓			✓
~~BottomIsFlat~~					✓	✓		
~~Handle on Top~~	✓				✓		✓	✓
HasHandle	✓	✓	✓	✓	✓	✓		✓
HasConcavity	✓	✓	✓	✓	✓	✓		
MadeOfPaper				✓	✓	✓	✓	



⑤ Using prior knowledge to alter the search objective

TangentProp algorithm:

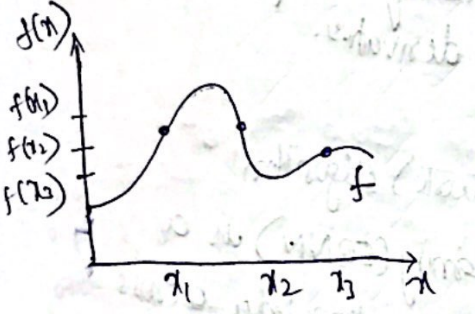
- TangentProp accommodates domain knowledge expressed as ~~derivatives~~ derivatives of the target function with respect to transformations of its inputs.
- Prior knowledge is to incorporate it into the error criterion, minimized by gradient descent, so that the network must fit a combined function of the training data and domain theory.

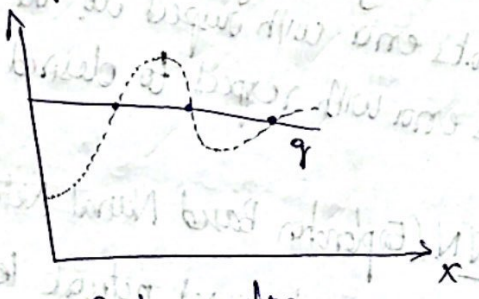
6) Handwritten character recognition.

- TangentProp algorithm trains a neural network to fit both training values and training derivatives
- Each training example consists of a pair $(x_i, f(x_i))$
- The tangentprop algorithm assumes various training derivatives of the target function.
- If each instance x_i is described by a single real value, then each training example may be of the form

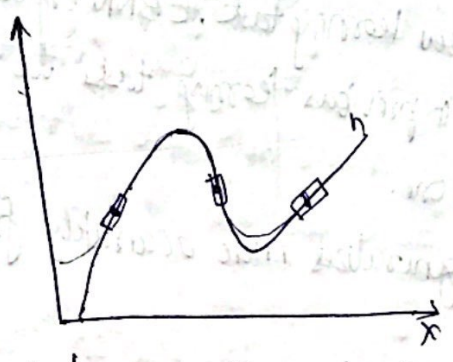
$$\left(x_i, f(x_i), \frac{\partial f(x)}{\partial x}\Big|_{x_i}\right) \Rightarrow [3\text{tuples}]$$

slope





Backpropogation
(It considers data in the form of 2tuples only)



Graph for TangentProp algorithm
(Considers 3 tuples training data.)
(Due to consideration of slope, it gives more accurate results) than Backpropogation algo.

- The modified error function in TangentProp is

$$E = \sum_i \left[(f(x_i) - \hat{f}(x_i))^2 + \mu \sum_j \left(\frac{\partial f(s_j(\alpha, x_i))}{\partial \alpha} - \frac{\partial \hat{f}(s_j(\alpha, x_i))}{\partial \alpha} \right)^2 \right]_{\alpha=0}$$

$\mu \Rightarrow$ constant value.

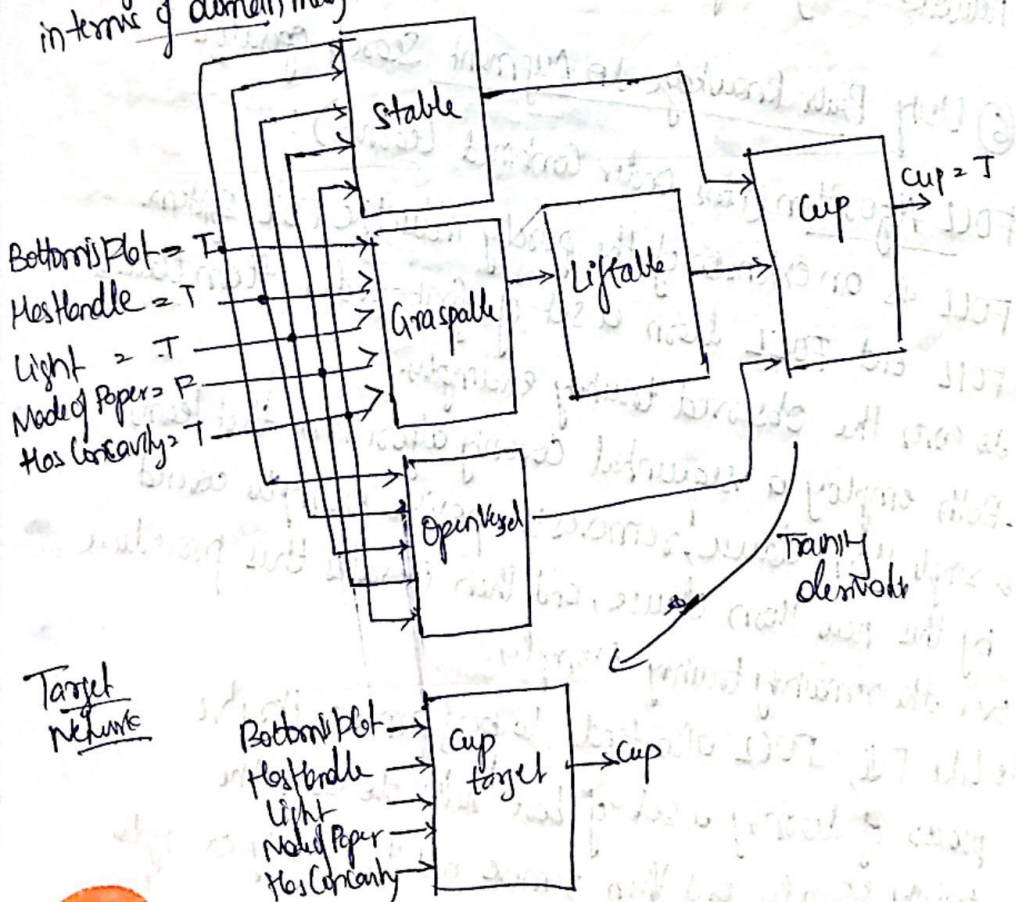
- TangentProp uses prior knowledge in the form of desired derivatives of the target function with respect to transformations of its inputs.
- Combines this prior knowledge with observed training data, by minimizing the objective function that measures both the network's error with respect to the training example values and its error with respect to desired derivatives.

EBNN (Explanation Based Neural Network) algorithm:

- Explanation-based neural network learning (EBNN) is a machine learning algorithm that transfers knowledge across multiple learning tasks.
- When faced with a new learning task, EBNN exploits domain knowledge accumulated in previous learning tasks to guide generalisation in the new one.
- As a result, EBNN generalizes more accurately from less data.

- EBNN builds on TangentProp algorithm in two significant ways.
- First, instead of relying on the user to provide training derivatives, EBNN computes training derivatives itself for each observed training example.
- Second, EBNN addresses the issue of how to weight the relative importance of the inductive (training data) and analytical (domain theory, prior knowledge) components of learning.
- The value of c_i is chosen independently for each training example.

Explanation of training examples in terms of domain theory.



- The top portion of this figure depicts an EBNN domain theory for the target function cup, with each rectangular block representing a distinct neural network in the domain theory.
- Some networks take the outputs of other networks as their inputs (e.g the rightmost network labeled cup takes its inputs from the outputs of the Stable, Liftable and Open Vessel networks)
- Thus, the networks can be chained together to infer the target function value for the input instance.
- EBNN makes use of the domain theory networks to learn the new target function. It does not alter the domain theory networks during the process.

(6) Using Prior knowledge to Augment Search Operators:

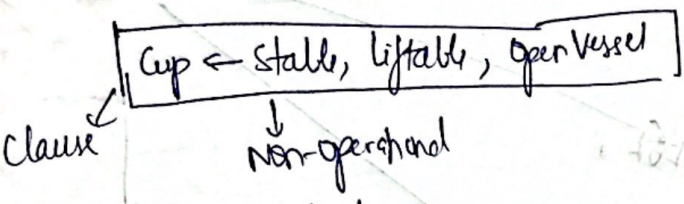
FOCL Algorithm (First Order Combined Learner):

- FOCL is an extension of the purely inductive FOIL system.
- FOIL and FOCL learn a set of first-order Horn clauses to cover the observed training examples.
- Both employ a sequential covering algorithm that learns a single Horn clause, removes the positive examples covered by this new Horn clause, and then iterates this procedure over the remaining training examples.

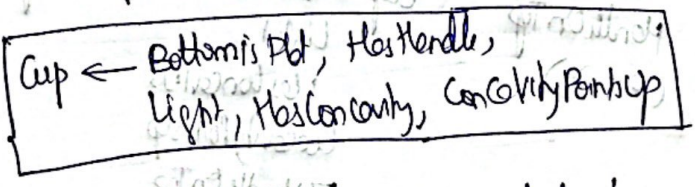
i.e Like FOIL, FOCL also tends to perform an iterative process of learning a set of best rules to cover the training examples and then remove all the training examples

covered by that best rule (using sequential covering algorithm).

- A literal is called operational if it can describe the training example properly leading to the output hypothesis. Literals that occur only as intermediate features in the domain theory (Stable, Liftable, OpenVessel), but not as primitive attributes of the instances, are considered non-operational.
- Non-operationals are evaluated in the same manner as operationals in FOCL.

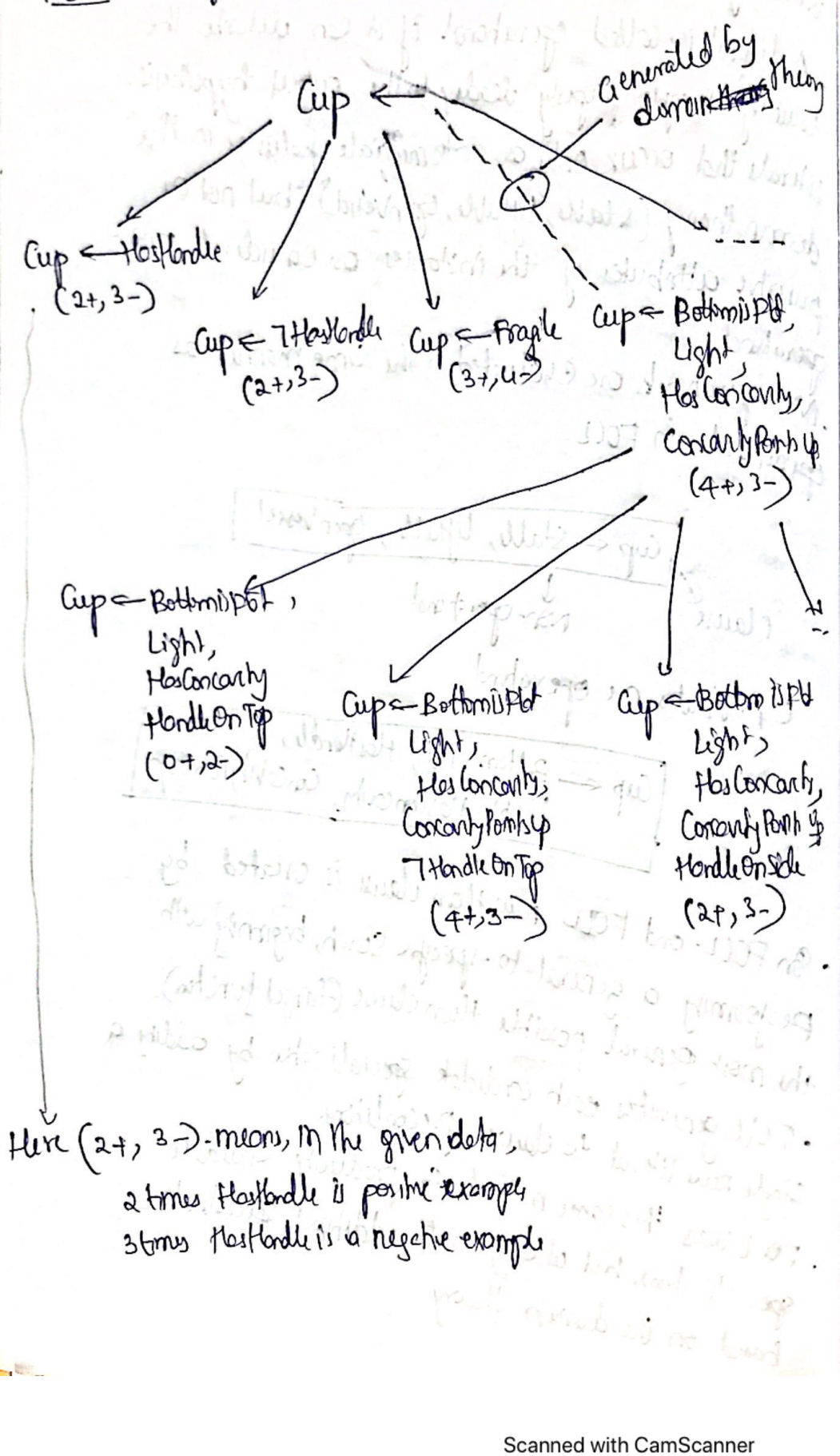


(Attributes are operational



- In FOCL and FOIL new Horn clause is created by performing a general-to-specific search, beginning with the most general possible Horn clause (target function).
- FOIL generates each candidate specialization by adding a single new literal to clause preconditions.
- FOCL uses this same method for producing candidate specialisations, but also generates additional specialisations based on the domain theory.

FOCL: Example (domain theory + training examples)



Here (2+, 3-) means, in the given data,
2 times HasHandle is positive example
3 times HasHandle is a negative example