Inductive learning:- (ML Algo)

In Inductive learning, the learner is given hypothesis space H and set of training examples D = $\{x_n, f(x_n)\}$

where,

x_n = instance

$f(x_n)$ = target value (for instance x_n)

The learner must select a output hypothesis 'h' from hypothesis space H.

The expected hypothesis 'h' (from hypothesis space H) must be consistant with the training examples. D.

short ans (Inductive learning train a model to generalize predictions based on examples or observations) Eg:- Decision tree, KNN, NN

- commonly used in supervised learning with labled data.

Analytical Learning:-

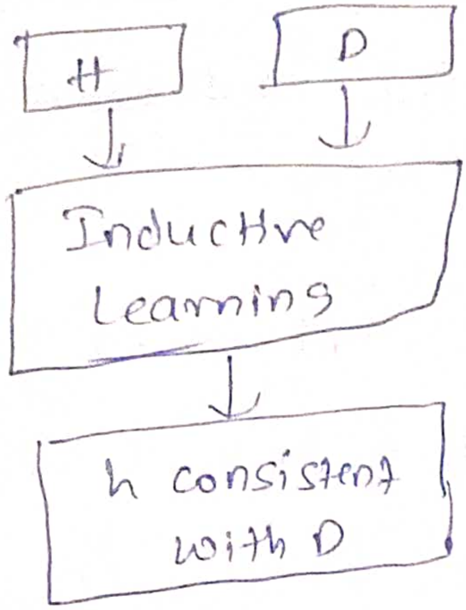
In analytical learning, the input to the learner has same hypothesis space H and training examples D same like inductive learning. Other than them it has an extra input, i.e domain theory B.

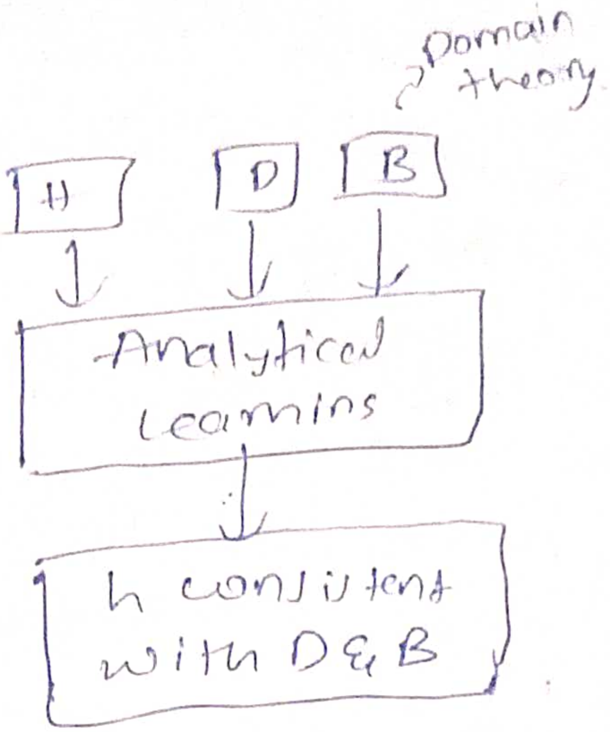
* domain theory B consists of background knowledge that can be used to explain the observed training examples.

* Hence, the expected output of the learner is a hypothesis h from hypothesis space H is consistent with both training example and domain theory B.

Inductive and analytical learning

difference





→ 2 inputs given to learner • Hypothesis space (H) • Training Examples (D)	→ 3 inputs given to learner • Hypothesis space (H) • Training Example (D) • Domain theory (B)
→ Hypothesis fits the data	→ Hypothesis fits the domain theory
→ statistical inferance	→ deductive inferrance.
→ requires little prior knowledge	→ learns from scarce data.

Domain Theory:-

A domain theory is said to be correct if each of its assertions (facts) is a truthful statement about the world.

A domain theory is said to be complete with respect to a given target concept and instance, if it convers every positive example in the instance space.

Learning with perfect domain theories : PROLOG - EBG

- Prolog:-

→ It is a logic programming language and declarative programming language.

→ Logic is expressed as relations (called as facts & Rules)

→ Core hart of Prolog lies at the logic being applied.

→ formulation/ computation is performed by running a query over these relations.

PROLOG - EBG

EBG → Explation based generalization.

EBG is a particular type of analytical approach which uses prior knowledge to identify the relavent features of training Examples from the irrelavent. So the Examples can be generalized based on logical reasoning (rather than Statistical reasoning).

Properties of PROLOG-EBG

→ It is sequential covering algorithm.

→ It is a deductive learning system, which assump domain knowledge is correct and complete.

→ Produces justified general hypothesis by using prior knowledge to analyze individual examples.

→ It assumes that domain theory is correct and complete. But, if the domain theory is incorrect or incomplete, the resulted learned concept may also be incorrect.

→ It computes the most general rule (by computing the weakest pre-image of the explanation) that can be justified by the explanation.

PROLOG-EBG Algorithm

PROLOG-EBG (Target concept, TrainingExam-ples, Domain Theory)

- LearnedRules ← { }
- Pos ← Positive examples from Training Examples.
- For each positive Example in Pos that is not covered by Learned Rules, do

1. Explain:
   - Explanation ← An explanation (proof) in terms of the Domain Theory That positive Examples satisfies the Target-concept

2. Analyze:-
   - SufficientConditions ← The most general set of features of positiveExample sufficient to satisfy the Target concept according to Explanation.

3. Refine:
   - LearnedRules ← LearnedRules + NewHornclause, where · NewHornclause

is of the form.

TargetConcept ← Sufficient Conditions.

- Return LearnedRules.

Explanation Based learning :- (EBL)

→ It is a form of analytical learning.

→ It has the ability to learn from single training instance.

→ Instead of taking more Examples, it focuses to learn a single, specific example

→ An EBL System accepts an example (i.e, training example) and explains what it learns from the Example.

→ EBL System takes only the relevant aspects of the training.

→ This explanation is translated into Particular form so that a problem solving program can understand in simple words, the explanation is generalized so that it can be used to solve the other problems.

Remarks on Explanation - Based Learning :-

- Produces justified general hypotheses by using prior knowledge to analyze individual examples.
- Explanation determines relevant attributes (features)
- Regressing allows deriving more general constraints (weakest preimages).
- Learned Horn clause corresponds to a sufficient condition to satisfy target concept.
- Prolog-EBG implicitly assumes the complete and correct domain theory.

• Perspectives of EBL.

- EBL as theory-guided generalization of example.
- EBL as example-guided reformation of theories.
- EBL as "just" restating what the learner already "knows"

Deductive learning

* It builds a model using logical principles and steps.
* The model adheres to predefined rules and processes to produce predictions based on new, unexplored data.
* Often used in system relying on first-order logic
* Example:- Prolog-based systems

Explanation-Based learning (EBL)

of search control knowledge

→ It's a method used in problem solving systems like PRODIGY.

→ Its objective is to EBL aims to distill effective search control knowledge from experience.

→ The process is EBL uses prior knowledge to reduce the complexity of the hypothesis space to be searched, thereby reducing sample complexity and improving generalization accuracy of the learner.

→ The PRODIGY uses EBL to learn from a variety of phenomena, including solutions, failures, and goal ~~inst~~ interactions.

→ It introduces a definition of "operationality" based on the utility of a learned rule, and gives methods for evaluating it dynamically.

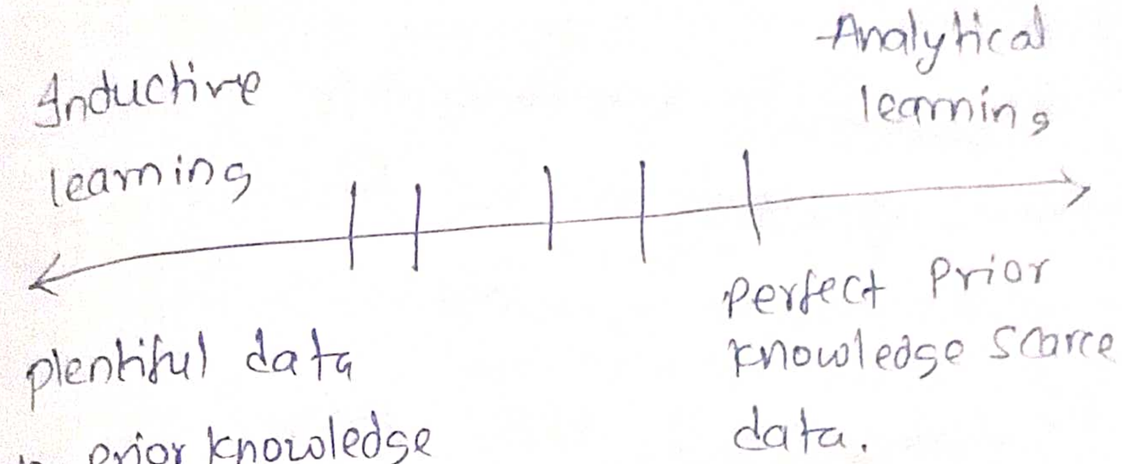
→ An Algorithm, EBS, is presented and proved correct.

In essence, EBL of search control knowledge is about learning from the successes and failures of past problem-solving attempts to improve future performance. It's a detailed examination of the learning component of the PRODIGY problem-solving system. This approach significantly extends the application of EBL methods.

Motivation

The two approaches work well for different types of Problem. Inductive and Analytical learning.

How to combine the two into a single algorithm that captures the best aspect of both?



Most practical problems lie some-where between these two extremes.

- In analyzing a database of medical records.
- In analyzing a stock market database.

<u>Desirable properties</u>

- Given no domain theory, it should learn at least as effectively as purely inductive methods.
- Given a perfect domain theory, it should learn at least as effectively as purely analytical methods.

- Given an imperfect domain theory and imperfect training data, it should combine the two to outperform either purely inductive or purely analytical methods.
- It should accommodate an unknown level of error in the training data and in the domain theory.

Inductive - Analytical Approaches to Learning

- The learning problem.
  - Given
    - A set of training examples D, possibly containing errors.
    - A domain theory B, possibly containing errors.
    - A space of candidate hypotheses H.
  - Determine
    - A hypothesis that best fits the training examples and domain theory.

- Tradeoff

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

- $error_D(h)$: proportion of Examples from D that are misclassified by h.

- $error_B(h)$: probability that h will disagree with B on the classification of a randomly drawn instance

- Learning methods as Search Algorithms
  - H: Hypothesis space
  - h_0: Initial hypothesis
  - O: set of search operators
  - G: Goal criterion.

- Use prior knowledge to
- Derive an initial hypothesis h_0 from which to begin the search.
  - KBANN.
- Alter the objective G of the hypothesis space search.
  - TangentProp, EBNN.

- Alter the available search steps (operator O)
  - FOCL

KBANN Algorithm:- {Knowledge-Based Artificial Neural Networks}

→ use prior knowledge to initialize the hypothesis to perfectly fit the domain theory.

→ then inductively refine this initial hypothesis as needed to fit the training data.

Given:-

- A set of training examples.
- A domain theory consisting of nonrecursive, propositional Horn clauses.

Determine

- An artificial neural network that fits the training examples, biased by the domain theory.

2 steps

Analytical step

Create an artificial neural network that perfectly fits the domain theory

Back propagation to refine the initial Network to fit the training examples

Inductive step

Algorithm

KBNN(Domain_Theory, Training_Examples)

Domain_Theory: set of propositional, nonrecursive Horn clauses.

Training_Examples: set of (input output) pairs of the target function.

Analytical step: create an initial network equivalent to the domain theory.

1. For each instance attribute create a network input.

2. For each Horn clause in the Domain_Theory, create the domain theory.

- Connect the inputs of this unit to the attributes tested by the clause antecedents.
- For each non-negated antecedent of the clause, assign a weight of w to the corresponding sigmoid unit input.
- Set the threshold weight wo for this unit to $-(n-0.5)w$, where n is the number of non-negated antecedents of the clause.

3. Add additional connections among the network units, connecting each network unit at depth i from the input layer to all network units at depth i+1. Assign random near-zero weights to these additional conditions.

Inductive step

- Apply the Backpropagation algorithm to adjust the initial network weights to fit the training examples.

Example

Training examples

	Cups	Non-cups
BottomIsFlat	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓
ConcavityPointsUp	✓ ✓ ✓ ✓	✓ ✓ ✓
Expensive	✓ ✓	✓ ✓
Fragile	✓ ✓	✓ ✓ ✓
HandleOnTop	✓ ✓	✓ ✓ ✓
HandleOnSide	✓ ✓	✓ ✓ ✓
HasConcavity	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
HasHandle	✓ ✓ ✓	✓ ✓ ✓ ✓
Light		
Made of cermic		✓ ✓ ✓
Made of Paper	✓	✓
Made of Styrofoam	✓ ✓	✓ ✓ ✓

Domain Theory:-

Cup ← Stable, liftable, Openvessel

Stable ← BottomIsflat

Liftable ← Graspable, Light

Openvessel ← HasConcavity, Concavity Points up.

In Analatical step

- A nural network equivalent to the domain theory.
  - created in the frist stage of the KBANN.

- sigmoid output value >=0.5 it true
<0.5 is false.

Expensive
Bottom is Flat — Stable
weight = w
Made of ceramic
Made of styrofoam
Weight = ~0
Made of Paper.
Has Handle — Graspable — liftable — cup
Handle on top
Handle on side
Light
Has concavity — openvessel
Concavity points up
Fragile

Threshold weight w_0 = -1.5w

Now Next in Inductive step

Expensive
Bottom is Flat — Stable
made of Ceramic
made of Styroforam
made of paper — liftable — Cup.
Has Handle — Graspable
Handle on top -
Handle on side
Light
Has conacvity — open-vessal
Concavity Points up
Fragile

— large positive weight
--- large negative weight

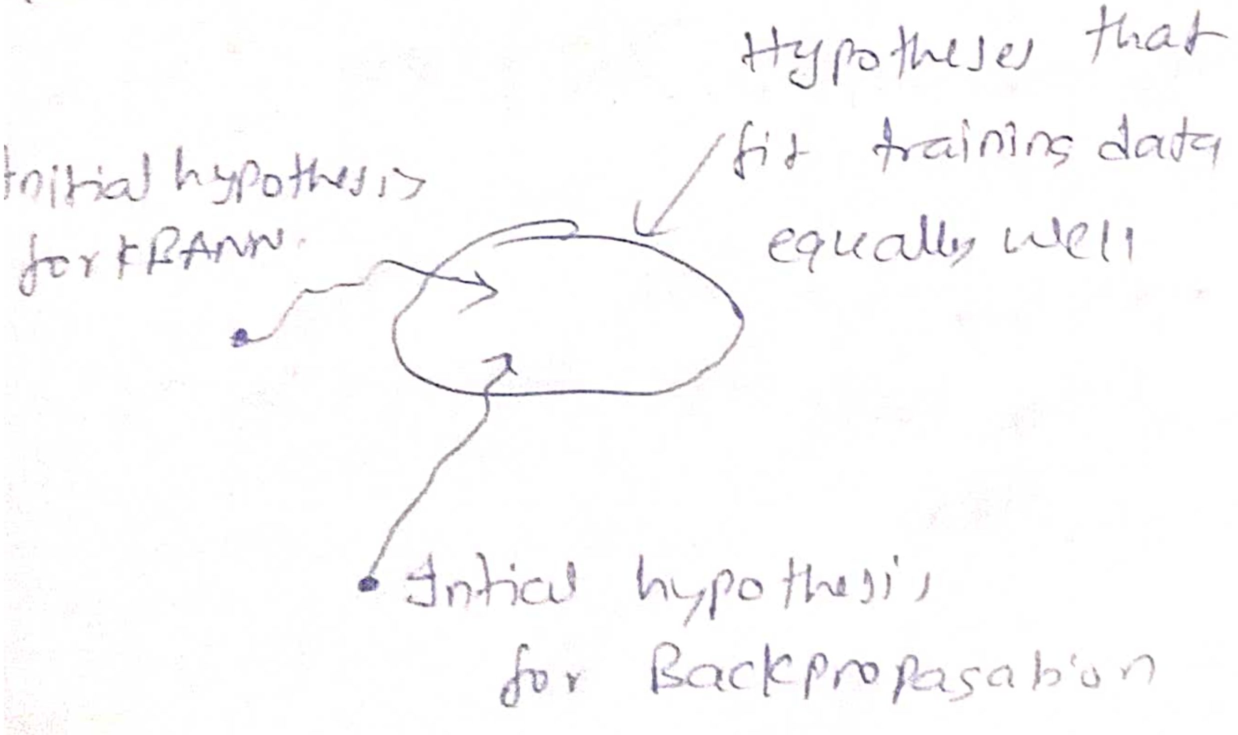
Benefits of KBNN

* Generalizes more accurately than Backpropagation.
* initialize the hypothesis.

Limitations in KBANN

* Accomodate only propositional domain theories.
* Misled when given high inaccurate domain theories.

Hypothesis space search in KBANN



Tangent Prop Algorithm

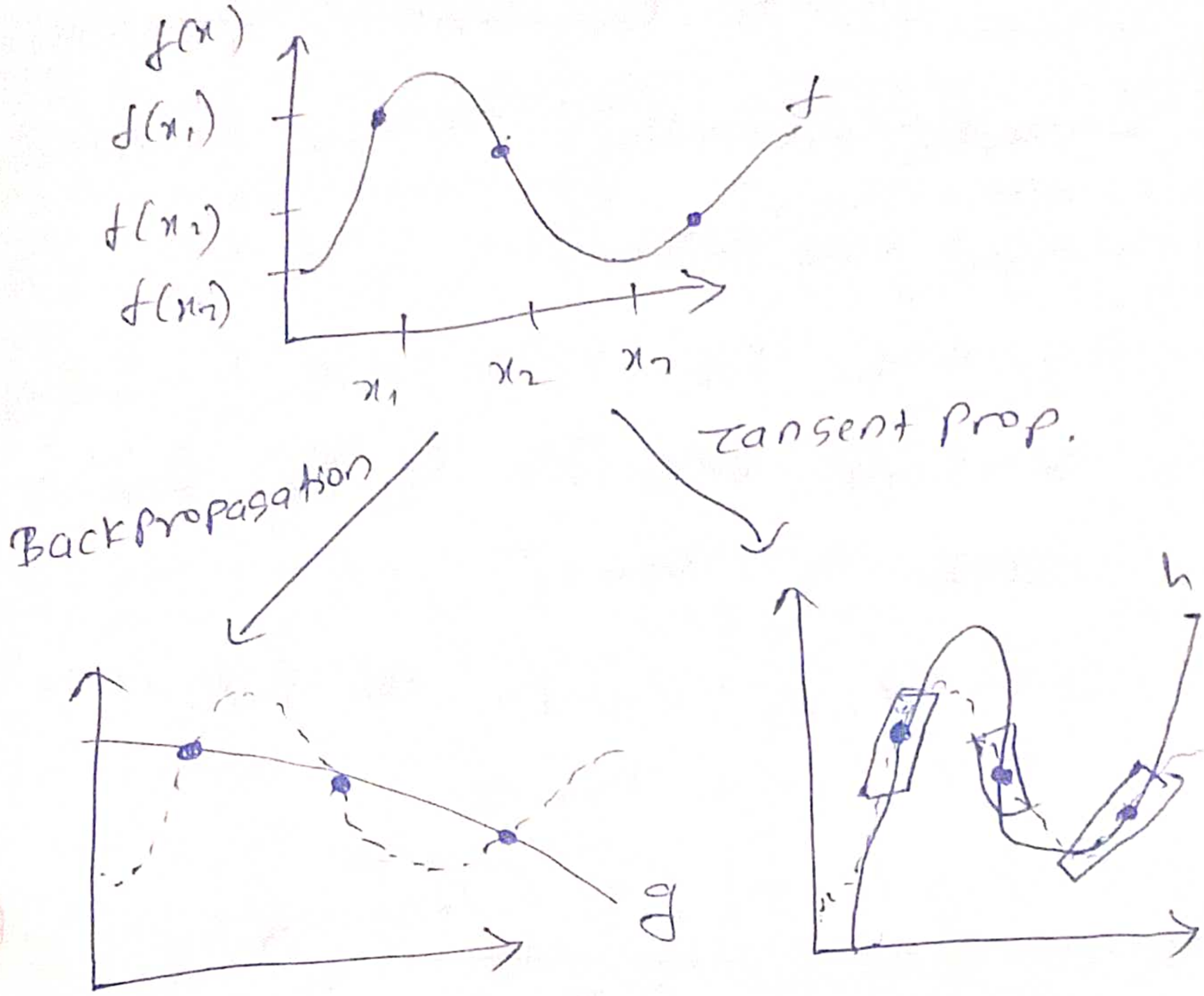
- Prior knowledge is to Derivatives of the target function.
- Trains a neural network to fit both training values and training derivatives.
- Tagent prop & EBNN.
  - output form purely inductive · methods
    - characters and object recognition.
    - Robot perception and control tasks.
- Training Examples

- up to now: $\langle x_i, f(x_i) \rangle$

- In TangentProp: $\langle x_i, f(x_i), \frac{\partial f(x)}{\partial x}\Big|_{x_i} \rangle$

- Assume various training derivative of the target function are also provided.

. Intuitively



The Learner has a better chance to correctly generalize from the sparse training data.

- Accept training derivatives with respect to various transformations of the input X.

Example
- Learning to recognize handwritten characters.

- Input x: An Image containing a single handwritten character.
- Task: correctly classify the character
- Prior knowledge
  - "The target function is invariant to small rotations of the character within the image."
  - $s(\alpha, x)$: Rotates the image x by α degrees.

$$\frac{\partial f(s(\alpha, x_i))}{\partial \alpha} = 0$$

- BACKPROPAGATION.
  - Performs gradient decent to attempt to minimize the sum of squared errors.

$$E = \sum_i (f(x_i) - \hat{f}(x_i))^2$$

- TangentProp
  - Accept multiple transformations
    - Each transformation must be of the form $s_j(\alpha, x)$

- α : continuous parameter
- s_j : Differentiable, $s_j(0, x) = x$

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

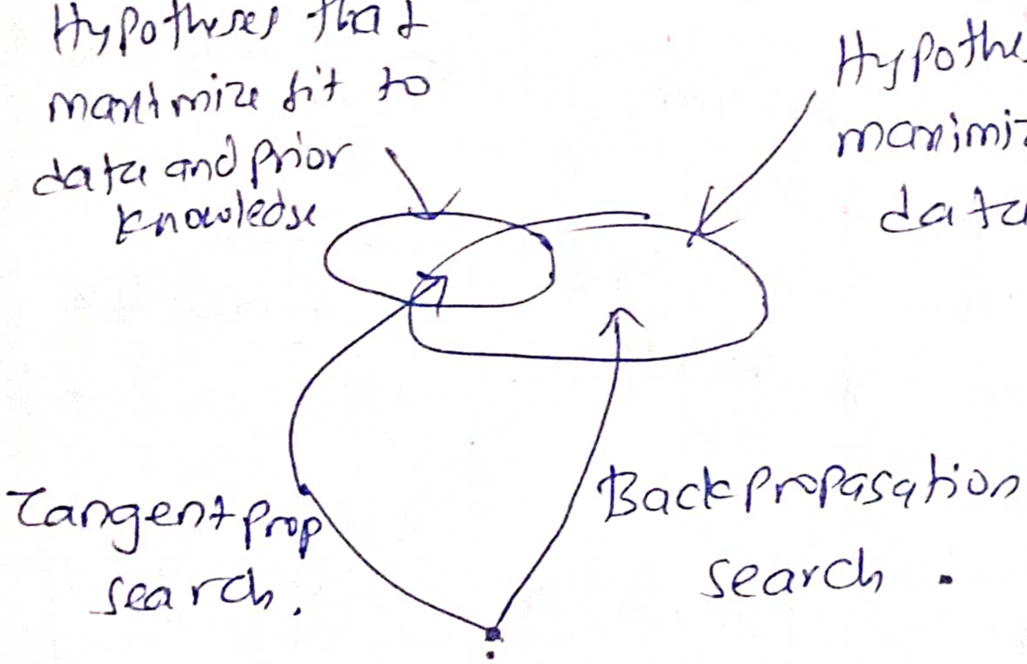
- μ : Constant.

- Recogonizing handwritten characters. (1992)
- Images containing a single digit 0-9
- Prior knowledge
  - classification of a character is invarient of vertical and horizontal translation.

Training set size	Percent error on test set	
	TangentProp	BackPropagation
10	34	48
20	17	33
40	7	18
80	4	10
160	0	3
320	0	0

- the behavior of algorithm is Sensitive to μ.
- Not robust ~~of~~ to errors in the prior knowledge.
  - Degree of error in the training derivatives is unlikely to be known in advance.

Hypothesis Space Search in Tangentprop



EBNN (Explanation-Based Neural Network Learning)

- Automatically select values for μ on a example-by-example basis in order to address the possibility of incorrect prior knowledge.

- using the prior knowledge to alter the search objective.

- Builds on TangentProp.
  - compute training derivatives itself for each examples.
  - "How to weight the relative importance of the inductive and analytic components of learning".

→ Determined by itself :

Given

- training example : $\langle x_i, f(x_i) \rangle$
- Domain Theory : represented as a set of previously trained neural networks.

Determine

- A new neural network that approximates the target function f
- This learned network is trained to fit both the training examples

and training derivatives of f extracted from the domain theory.

Algorithm

Create a feedforward Network → Initialize small random weights

↓

Determines training derivatives → Predict the value $A(x_i)$ of the target function for training example x_i using domain theory network

↓ Analyze the weights and activation of the domain theory network

↓ Extract derivatives of $A(x_i)$

↓

Train the target network

↳ Error function

Error function

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

First term ↓ Inductive constraint (that the hypothesis must fit the training data)

Second term ↓ Analytical constraint (that the hypothesis must fit the training derivatives) where $\mu_i = 1 - \dfrac{|A(x_i) - f(x_i)|}{c}$

x_i = The i th training instance.

$A(x)$: The domain theory prediction for input x.

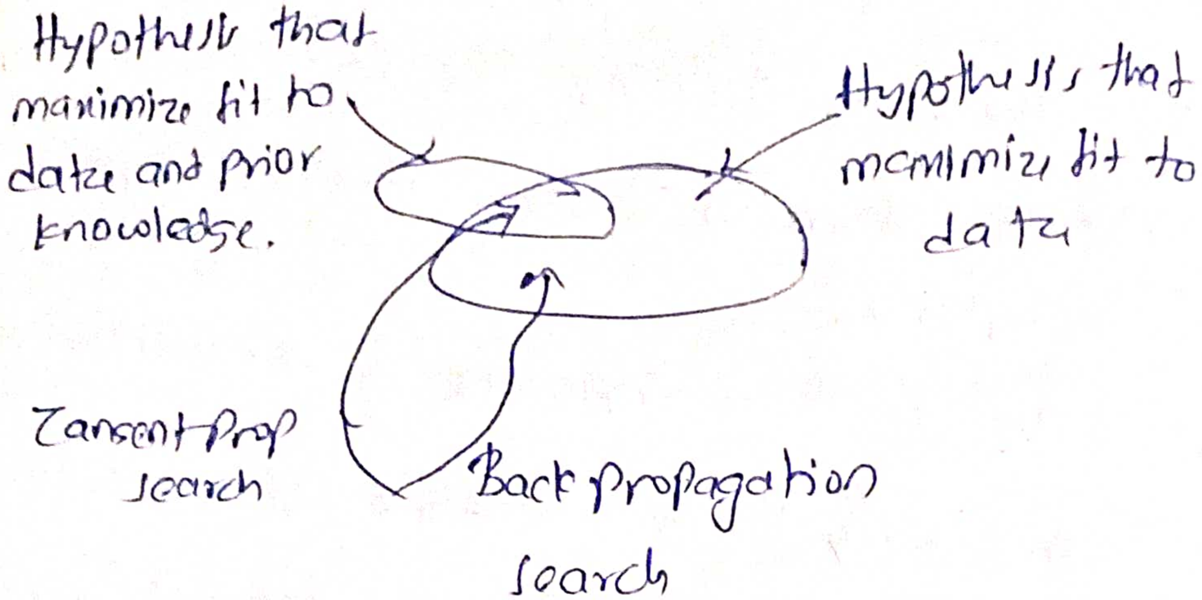
x^j: The j th component of the vector x.

c: A normalizing constant ($0 \le \mu_i \le 1$, for all i)

Remarks

- Domain theory: Expressed as a set of Previously learned Neural Networks
- training derivative: How the target function value is influenced by a small change to attribute value
- μ_i: Determined independently for each training example, based on how accurately the domain theory predicts the training value for the example

Hypothesis Space Search in EBNN



EBNN vs PROLOG-EBG

	EBNN	PROLOG-EBG
Explanation:-	Training derivatives	weakest Preimage
Domain:	Neural Network	Horn clause
theory	Imperfect	Perfect.
Size	Learns a fixed size neural network	Learns a growing set of Horn clause

FOCL (First order combined Learner)

→ using Prior knowledge to augment search operators.

→ Extension of the purely FOIL (First order Inductive Learning)

	FOCL	FOIL
Generating candidate specialization	FOIL + Additional Specializations based on the domain theory	Add a single new literal to the clause precondition
	Learn a set of first-order Horn clauses covering	sequential Algorithm

→ Operational

→ If a literal is allowed to be used in describing an output hypothesis.

→ Nonoperational

– If a literal occur only as intermed-iate features in the domain theory.

Algorithm

- Generating Candidate Specializations.

Selects one of the domain theory clause

↓

Nonoperational literal is replaced

↓

Prune the preconditions of h unless pruning reduces classification accuracy over training examples

- **Remarks:-**

- Horn clause of the form

$$\boxed{c \leftarrow O_i \wedge O_b \wedge O_f}$$

O_i: An initial conjunction of operational literals.

O_b: A conjunction of operational literals.

O_f : A final conjunction of operational literals.

→ uses both a syntatic ~~generalization~~ generation of candidate specialization and a domain theory driven generation of candidate specialization at each step.

→ Hypothesis Space of FOCL