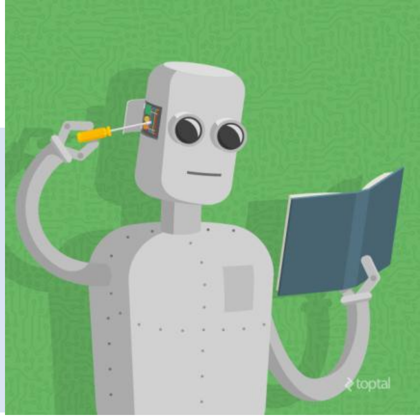
Machine Learning (15CS73)



Text Book :

Tom M. Mitchell, Machine Learning, India Edition 2013, McGraw Hill

Lecture Notes :https://thyagumachinelearning.blogspot.com/

ಡಾ|| ತ್ಯಾಗರಾಜು ಜಿ.ಎಸ್
Head Dept of CSE
SDMIT Ujire

Modules

- **Module1 :** Well posed learning problems, Designing a Learning system, Perspective and Issues in Machine Learning. Concept Learning: Concept learning task, Concept learning as search, Find-S algorithm, Version space, Candidate Elimination algorithm, Inductive Bias.
- **Module2**: **Decision tree representation**, Appropriate problems for decision tree learning, Basic decision tree learning algorithm, hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning
- **Module3: Artificial Neural Networks**: Introduction, Neural Network representation, Appropriate problems, Perceptron's, Backpropagation algorithm.
- **Module4**: Introduction, **Bayes theorem**, Bayes theorem and concept learning, ML and LS error hypothesis, ML for predicting probabilities, MDL principle, Naive Bayes classifier, Bayesian belief networks, EM algorithm
- **Module5:** Motivation, **Estimating hypothesis accuracy**, Basics of sampling theorem, General approach for deriving confidence intervals, Difference in error of two hypothesis, Comparing learning algorithms. **Instance Based Learning**: Introduction, k-nearest neighbor learning, locally weighted regression, radial basis function, cased-based reasoning, **Reinforcement Learning**: Introduction, Learning Task, Q Learning .

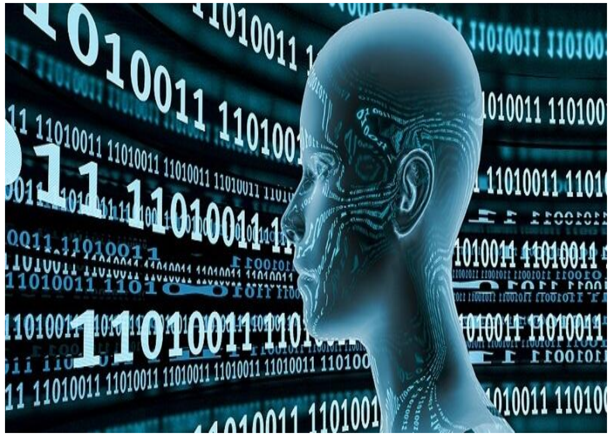
Assignment

- Watch You tube videos and understand the following
  - What is Checkers Game ?
  - History of Checkers Game.
  - Rules of Checkers Game.
  - How to play the Checkers Game.

Module 1 Chapter1

Text Book : Machine Learning by Tom M Mitchell (Chapter 1)

Dr Thyagaraju G S
Head Dept of CSE
SDMIT Ujire



Syllabus

1. Introduction
2. Well Posed Problems
3. Designing a Learning System
4. Perspective and Issues in Machine Learning
5. Summary

1.Introduction

- How to program computers to learn?

  Learning: Improving automatically with experience

- Example: Computers learning from medical records which treatments are most effective for new diseases.

State of Art / Some successful applications of machine learning

- Learning to recognize spoken words (Lee, 1989; Waibel, 1989).
- Learning to drive an autonomous vehicle (Pomerleau, 1989).
- Learning to classify new astronomical structures (Fayyad et al., 1995).
- Learning to play world-class backgammon (Tesauro 1992, 1995).
- Predicting Recovery rates of pneumonia patients
- Detecting Fraudulent use of credit cards
- Playing games at the level of humans

Some disciplines of their influence on machine learning

- Artificial intelligence
- Bayesian methods
- Computational complexity theory
- Control theory
- Information theory
- Philosophy
- Psychology
- Neurobiology
- Statistics

What is Machine Learning ?

- Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.
- **Machine learning focuses on the development of computer programs** that can access data and use it learn for themselves.

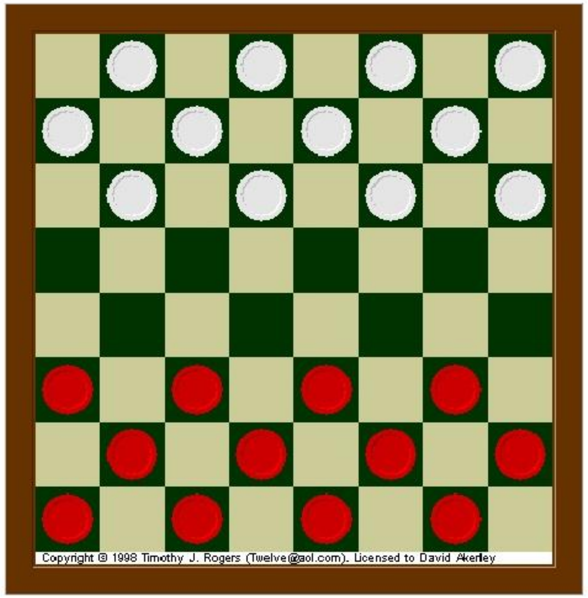
2.Well-Posed Learning Problems

- The study of Machine learning is about writing software that improves its own performance with experience
- **Definition [Mitchell]**: A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T,** as measured by **P**, improves with experience **E**.

Example1 : A Checkers Learning Problem

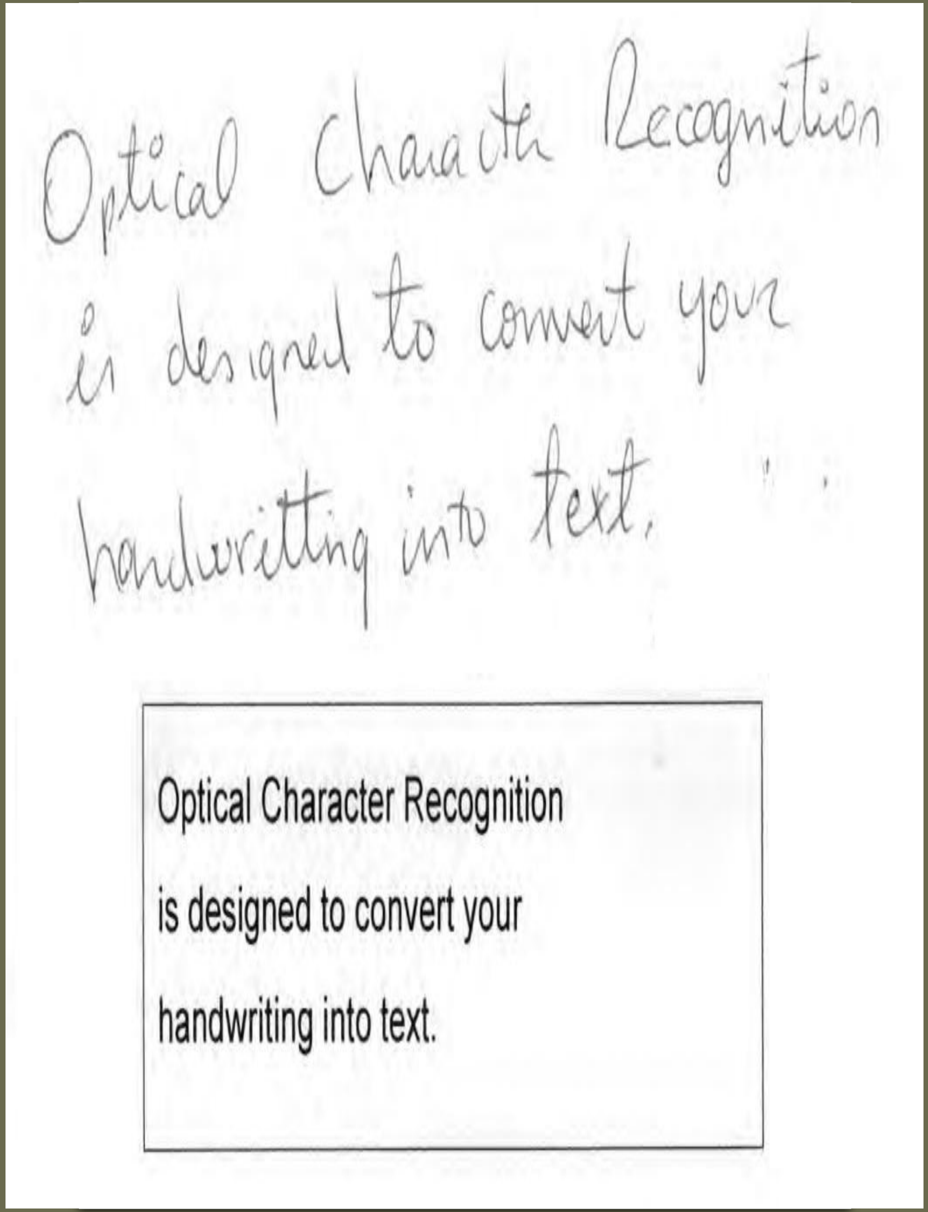
- **Task T** = Playing checkers
- **Performance Measure P** Percentage of games won against opponent
- **Training Experience E** = Playing practice games against itself

Checker



Example2: A Handwriting Recognition Learning Problem

- **Task T** = Recognizing and classifying handwritten words
- **Performance Measure P** Percentage of words correctly classified
- **Training Experience E** = A database of handwritten words with given classification



Example3 : A Robot driving learning program

- **Task T =** Driving on public four lane highways using vision sensors
- **Performance Measure P =** Average distance handled before error (as judged by human overseas)
- **Training Experience E** = A sequences of images and steering commands recorded while observing a human driver .

3. DESIGNING A LEARNING SYSTEM

Steps to design a learning system

1. Problem Description
2. Choosing the Training Experience
3. Choosing the Target Function
4. Choosing a Representation for the Target Function
5. Choosing a Function Approximation Algorithm
    1. ESTIMATING TRAINING VALUES
    2. ADJUSTING THE WEIGHTS
6. The Final Design

3.1 Problem Description: A Checker Learning Problem

- **Task T:** Playing Checkers
- **Performance Measure P:** Percent of games won against opponents
- **Training Experience E:** To be selected ==> Games Played against itself

3.1 Choosing the Training Experience (1)

- Training experience impacts on success or failure of the learners. The attributes of training experience
- Key Attributes –Will the training experience provide *direct* or *indirect* feedback?
  - **Direct Feedback:** system learns from examples of individual checkers board states and the correct move for each
  - **Indirect Feedback**: Move sequences and final outcomes of various games played
    - *Credit assignment problem*: Value of early states must be inferred from the outcome

- Second Attribute : Degree to which the learner controls the sequence of training examples
  - **Teacher selects** informative boards and gives correct move
  - **Learner proposes** board states that it finds particularly confusing. Teacher provides correct moves
  - **Learner controls** board states and (indirect) training classifications

3.1 Choosing the Training Experience (2)

- Third Attribute : How well the training experience **represents the distribution** of examples over which the final system performance ***P*** will be measured
  - If training the checkers program consists only of experiences played against itself, it may never encounter crucial board states that are likely to be played by the human checkers champion
  - Most theory of machine learning rests on the assumption that the distribution of training examples is identical to the distribution of test examples

Partial Design of Checkers Learning Program

- A checkers learning problem:
  - Task *T*: playing checkers
  - Performance measure *P*: percent of games won in the world tournament
  - Training experience *E*: games played against itself
- Remaining choices
  - The exact type of knowledge to be learned
  - A representation for this target knowledge
  - A learning mechanism

3.2 Choosing the Target Function (1)

- *Target function:* *ChooseMove* : $B \rightarrow M$
- **Alternative target function**
  - An evaluation function that assigns a numerical score to any given board state
  - $V : B \rightarrow \Re$ (where $\Re$ is the set of real numbers)
    - $V(b)$ for an arbitrary board state b in B
      - if b is a final board state that is won, then $V(b) = 100$
      - if b is a final board state that is lost, then $V(b) = -100$
      - if b is a final board state that is drawn, then $V(b) = 0$
      - if b is not a final state, then $V(b) = V(b')$, where b' is the best final board state that can be achieved starting from b and playing optimally until the end of the game

Choosing the Target Function (2)

- V(b) gives a recursive definition for board state b
  - Not usable because not efficient to compute except is first three trivial cases
  - *nonoperational* definition
- Goal of learning is to discover an operational description of *V*
- Learning the target function is often called **function approximation**
  - Referred to as $\hat{V}$

3.3 Choosing a Representation for the Target Function -1

- Choice of representations involve trade offs
  - Pick a very expressive representation to allow close approximation to the ideal target function V
  - More expressive, more training data required to choose among alternative hypotheses

3.3 Choosing a Representation for the Target Function -2

- Use linear combination of the following board features:
  - *x1*: the number of black pieces on the board
  - *x2*: the number of red pieces on the board
  - *x3*: the number of black kings on the board
  - *x4*: the number of red kings on the board
  - *x5*: the number of black pieces threatened by red (i.e. which can be captured on red's next turn)
  - *x6*: the number of red pieces threatened by black

3.3 Choosing a Representation for the Target Function -3

$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

Partial Design of Checkers Learning Program

- A checkers learning problem:
  - Task *T*: playing checkers
  - Performance measure *P*: percent of games won in the world tournament
  - Training experience *E*: games played against itself
  - *Target Function*: *V*: Board $\rightarrow \Re$
  - *Target function representation*

$$\hat{V}(b) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6$$

3.4. Choosing a Function Approximation Algorithm

- To learn we require a set of training examples describing the board *b* and the training value $V_{train}(b)$
  - Ordered pair $\langle b, V_{train}(b) \rangle$

$$\langle \langle x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0 \rangle, +100 \rangle$$

3.4.1 Estimating Training Values

- Need to assign specific scores to *intermediate* board states
- Approximate intermediate board state *b* using the learner's current approximation of the next board state following *b*

$$V_{train}(b) \leftarrow \hat{V}(Successor(b))$$

  - Simple and successful approach
  - More accurate for states closer to end states

3.4.1 Adjusting the Weights

- Choose the weights w_i to best fit the set of training examples
- Minimize the squared error E between the train values and the values predicted by the hypothesis

$$E \equiv \sum_{\langle b, V_{train}(b)\rangle \in training examples} \left(V_{train}(b) - \hat{V}(b)\right)^2$$

- Require an algorithm that
  - will incrementally refine weights as new training examples become available
  - will be robust to errors in these estimated training values
- Least Mean Squares (LMS) is one such algorithm

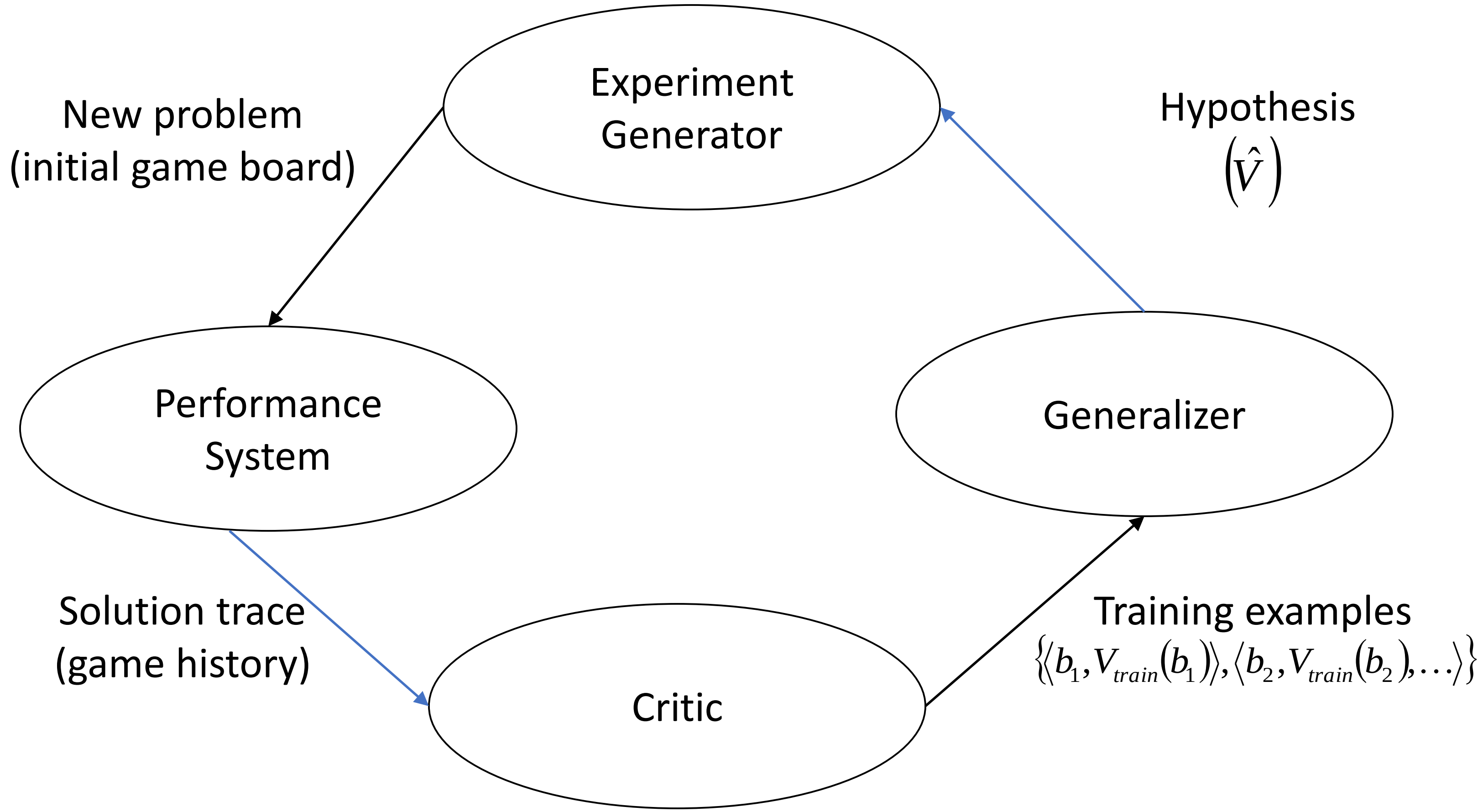
LMS Weight Update Rule

- For each train example $\langle b, V_{train}(b) \rangle$
  - Use the current weights to calculate
  - For each weight w_i, update it as $\hat{V}(b)$

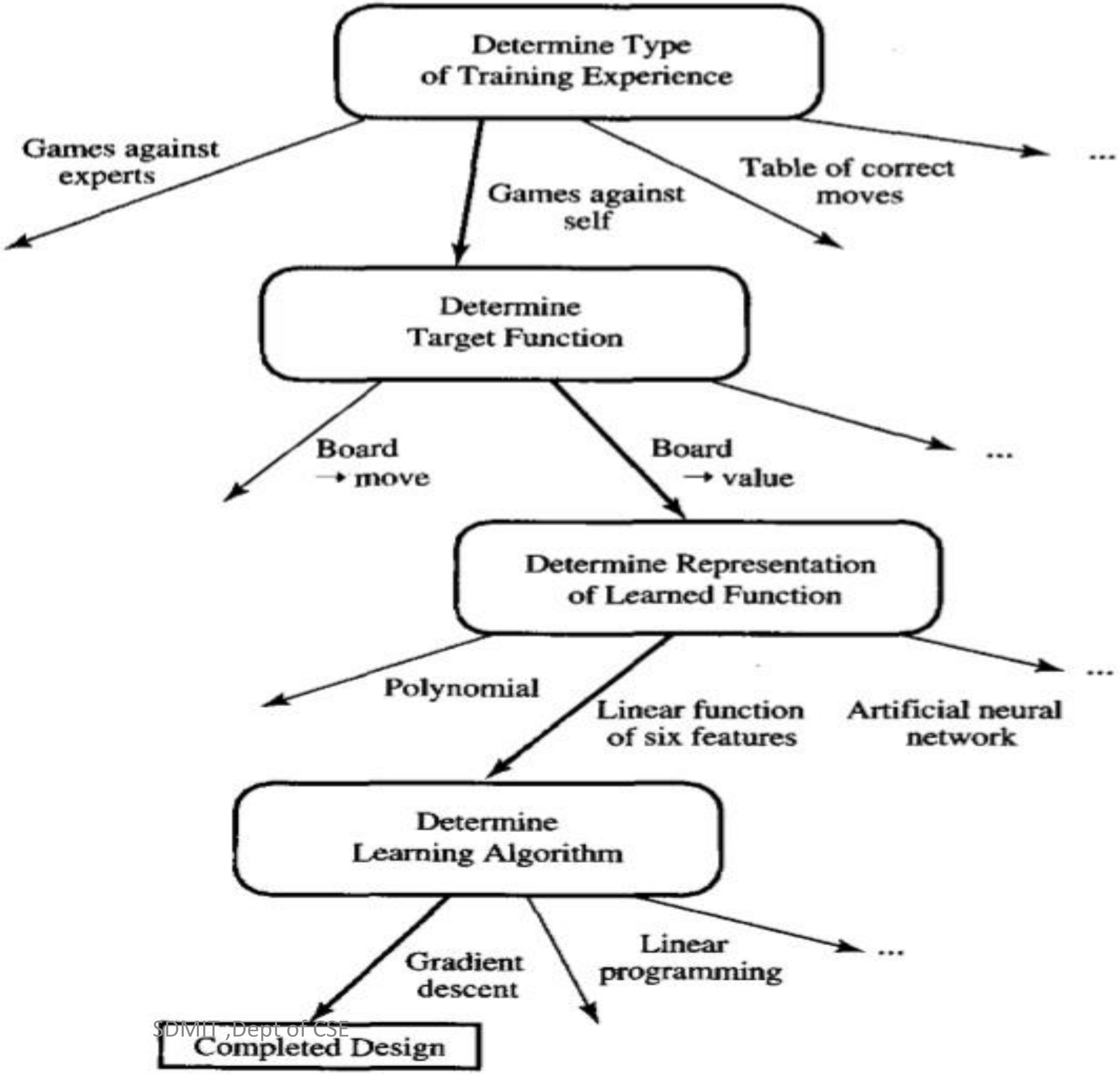
$$w_i \leftarrow w_i + \eta \left(V_{train}(b) - \hat{V}(b) \right) x_i$$

  - where
    - η is a small constant (e.g. 0.1)

3.5 Final Design



Summary of choices in designing the checkers learning program



4.0 Perspectives and Issues in Machine Learning

4.1 Perspectives in Machine Learning

- One useful perspective on machine learning is that it involves searching a very large space of possible hypotheses to determine one that best fits the observed data and any prior knowledge held by the learner.

4.2 Issues in Machine Learning (i.e., Generalization)

1. What algorithms exist for learning general target functions from specific training examples?
2. In what settings will particular algorithms converge to the desired function, given sufficient training data?
3. Which algorithms perform best for which types of problems and representations?

4.2 Issues in Machine Learning (i.e., Generalization)

4. How much training data is sufficient?
5. What general bounds can be found to relate the confidence in learned hypotheses?
6. When and how can prior knowledge held by the learner guide the process of generalizing from examples?
7. Can prior knowledge be helpful even when it is only approximately correct?

4.2 Issues in Machine Learning (i.e., Generalization)

8. What is the best strategy for choosing a useful next training experience, and how does the choice of this strategy alter the complexity of the learning problem?
9. What is the best way to reduce the learning task to one or more function approximation problems?
10. How can the learner automatically alter its representation to improve its ability to represent and learn the target function?

How to play checkers Game?

How To Play Checkers

Summary (Continued)

1. Introduction
2. Well Posed Problems
3. Designing a Learning System
4. Perspective and Issues in Machine Learning

Question Bank M1.1

1. **Define Machine Learning. Discuss with examples Why Machine Learning is Important.**
2. **Discuss with examples some useful applications of machine learning**
3. **Explain how some areas/disciplines have influenced the Machine learning.**
4. **Define Learning Program for a given Problem. Describe the following problems with respect to Tasks, Performance and Experience:**
    1. **Checkers Learning Problems**
    2. **Handwritten Recognition Problem**
    3. **Robot Driving Learning Problem**
5. **Describe in detail all the steps involved in designing a Learning Systems**
6. **Discuss the Perspective and Issues in Machine Learning.**