Def. AI:

Artificial intelligence is that activity devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment.

Def AI Effect:

The great practical benefits of AI applications and even the existence of AI in many software products go largely unnoticed by many despite the already widespread use of AI techniques in software. This is the AI effect.

Example: A modern calculator is assumed to have no AI whereas when its previous version, Arithometer was discovered it was said to have AI in doing calculations.

Machine Learning:

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data.

Deep Learning:

Deep Learning is classified as a class of machine learning algorithm that use a cascade of many layers of nonlinear processing units for feature extraction and transformation. It is based on the (unsupervised) learning of multiple levels of features or representations of the data. Higher level features are derived from lower level features to form a hierarchical representation.

Agents:

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.

Rationality:

A rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Chinese room experiment:

The Chinese room argument holds that a program cannot give a computer a "mind", "understanding" or "consciousness", regardless of how intelligently or human-like the program may make the computer behave.

Suppose that artificial intelligence research has succeeded in constructing a computer that comfortably passes the Turing test: it convinces a human Chinese speaker that the program is itself a live Chinese speaker.

The question Searle wants to answer is this: does the machine literally "understand" Chinese? Or is it merely simulating the ability to understand Chinese?

Searle can sit in a room and perform the program instructions step-by-step manually. However, Searle would not be able to understand the conversation.

Types of agents:

**Rational agent:** A rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

**Omniscient agent:** An omniscient agent knows the actual outcome of its actions and can act accordingly; but omniscience is impossible in reality.

**Simple reflex agent:** These agents select actions on the basis of the current percept, ignoring the rest of the percept history.

**Model-based agent:** current percept is combined with the old internal state to generate the updated description of the current state, based on the agent’s model of how the world works. Knowledge about “how the world works”—whether implemented in simple Boolean circuits or in complete scientific theories—is called a model of the world.

**Goal-based agent:**

As well as a current state description, the agent needs some sort of goal information that describes situations that are desirable. Example: The reflex agent brakes when it sees brake lights. A goal-based agent, in principle, could reason that if the car in front has its brake lights on, it will slow down. Given the way the world usually evolves, the only action that will achieve the goal of not hitting other cars is to brake.

**Utility-based agent:**

An agent’s utility function is essentially an internalization of the performance measure. If the internal utility function and the external performance measure are in agreement, then an agent that chooses actions to maximize its utility will be rational according to the external performance measure.

**Learning agent:**

The learning element uses feedback from the "critic" on how the agent is doing and determines how the performance element should be modified to do better in the future.

Properties of Environment:

**Fully observable vs. partially observable:** If an agent’s sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. If the agent has no sensors at all then the environment is unobservable.

**Single agent vs. multi-agent:** example, an agent solving a crossword puzzle by itself is clearly in a single-agent environment, whereas an agent playing chess is in a two agent environment.

**Deterministic vs. stochastic:** If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic.

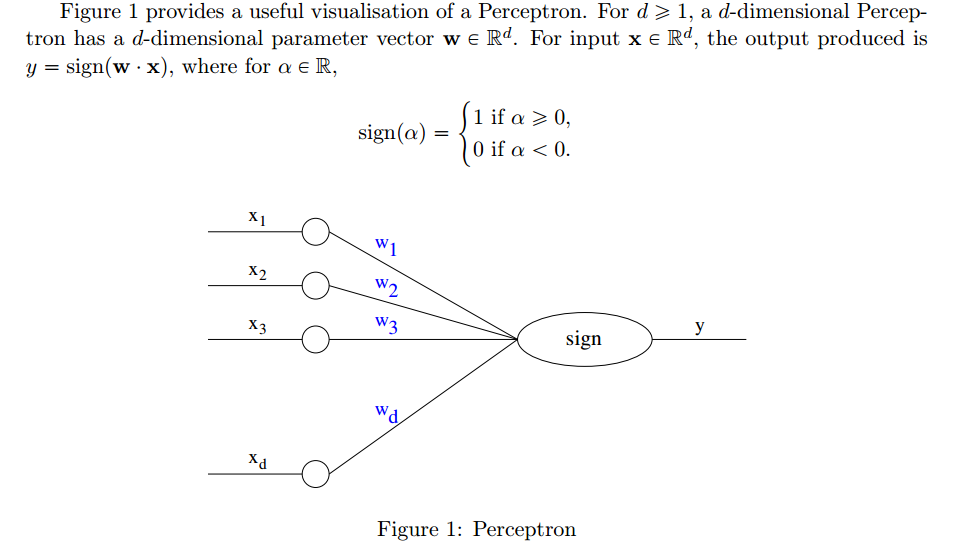
**Episodic vs. sequential:** In an episodic the next episode does not depend on the actions taken in previous episodes. In sequential environments, on the other hand, the current decision could affect all future decisions.

**Static vs. dynamic:** If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static.

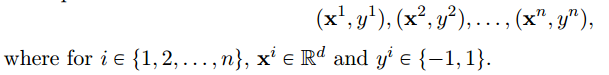
**Discrete vs. continuous:** The discrete/continuous distinction applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent. Example: Chess also has a discrete set of percepts and actions. Taxi driving is a continuous-state and continuous-time problem.

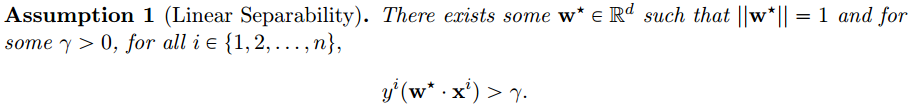
**Known vs. unknown:** In a known environment, the outcomes for all actions are given. Obviously, if the environment is unknown, the agent will have to learn how it works in order to make good decisions.

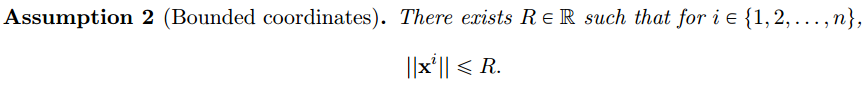
Perceptron Learning Algorithm: read from slides classnotes-1

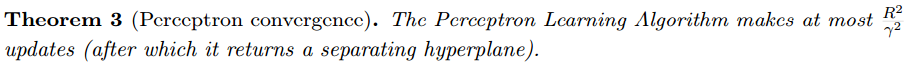
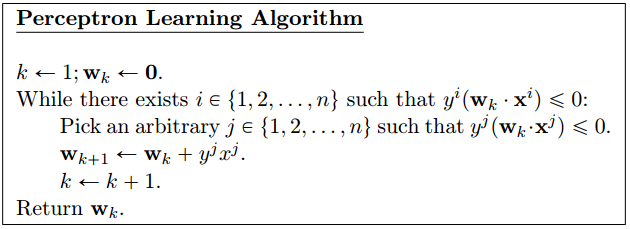


Input:

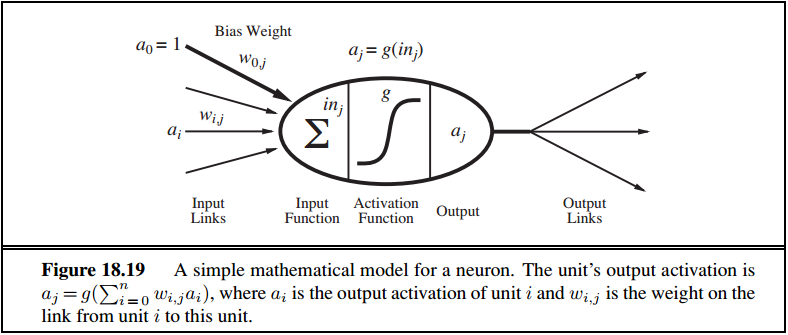


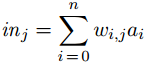


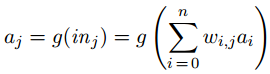




Neural Networks:

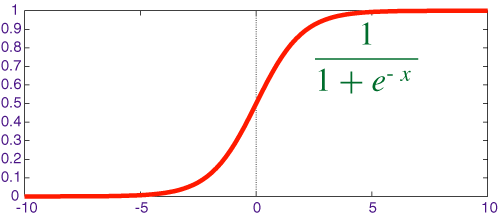






If the activation function g is hard then it is called **perceptron** else if a logistic function is used it is called as **sigmoid perceptron**.

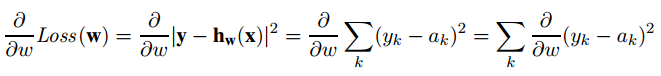
**Sigmoid activation function**:



A **feed-forward network** has connections only in one direction—that is, it forms a directed acyclic graph.

A **recurrent network**, on the other hand, feeds its outputs back into its own inputs.

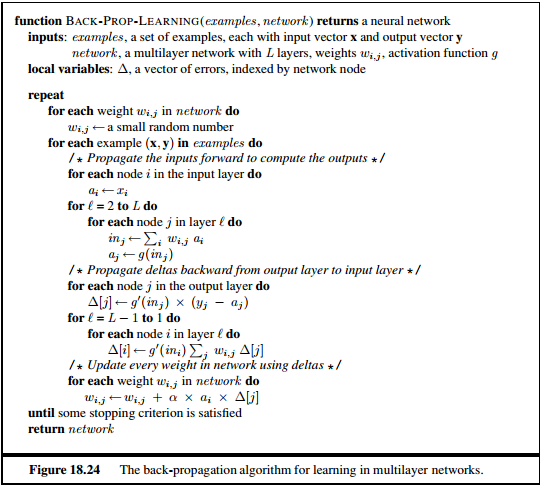
Perceptrons are single layer feed-forward network.





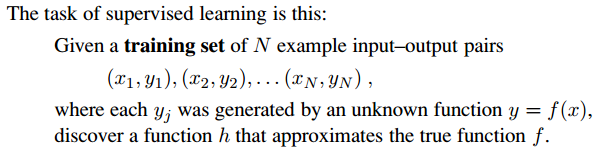




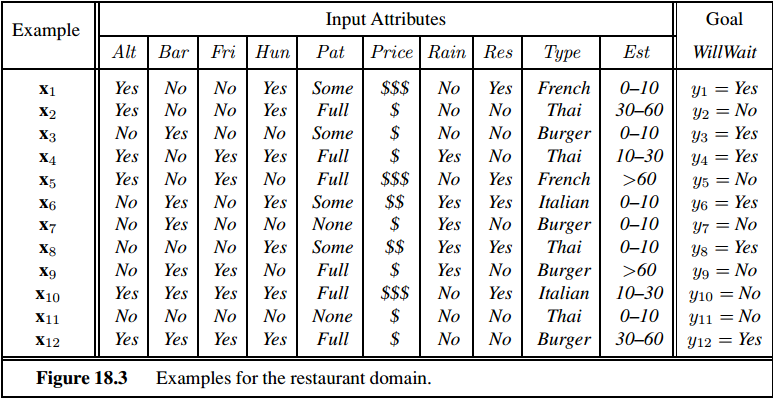


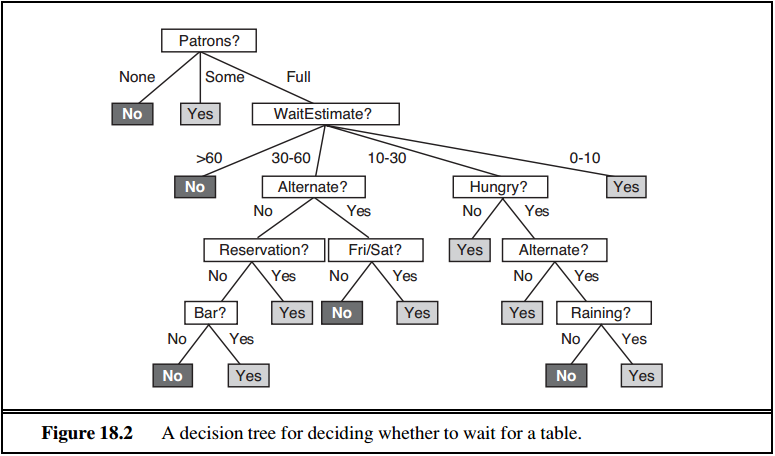


Supervised Learning:

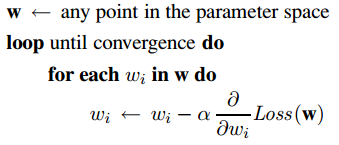


Decision Trees:





Gradient Descent:

 alpha is called learning factor.

Cross-validation (k-fold cross validation):

In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged to produce a single estimation.

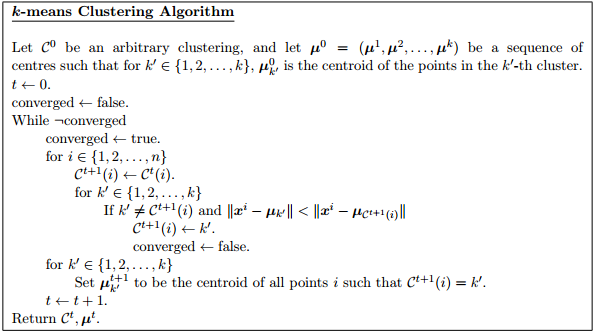
**Advantage**: all observations are used for both training and validation, and each observation is used for validation exactly once.

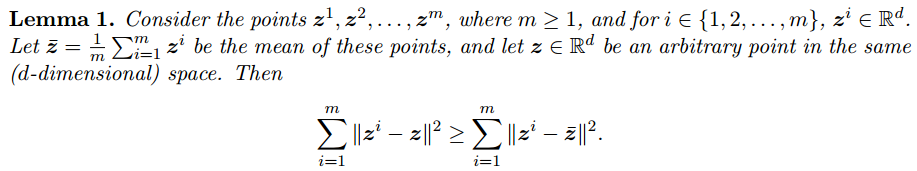
Nearest-neighbor method of classification:

The class(y) of a query point is decided by taking average of its k-nearest neighbors.

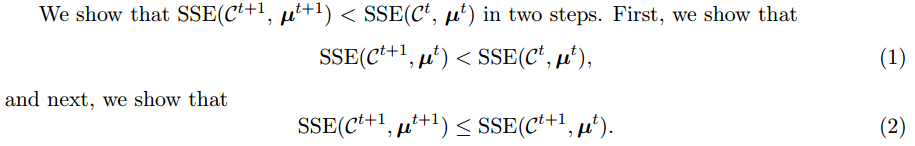
Distance here can be calculated here using various methods. (E.g. Euclidean distance)

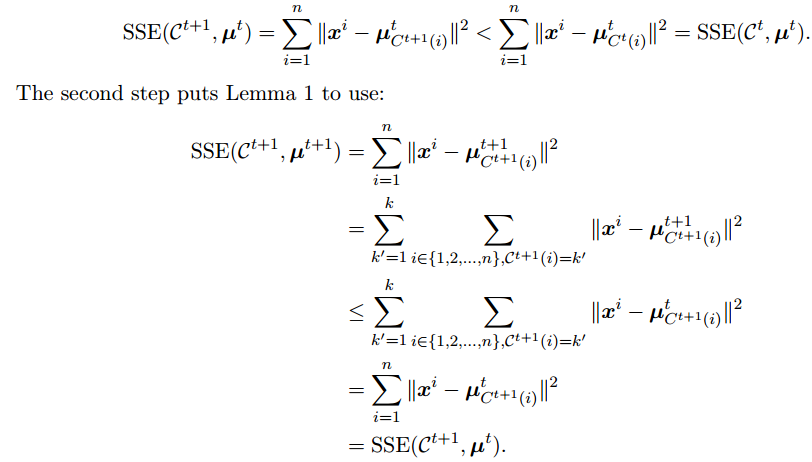
K-mean clustering Problem: Read from classnote-2



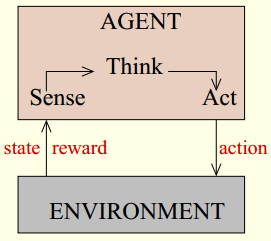






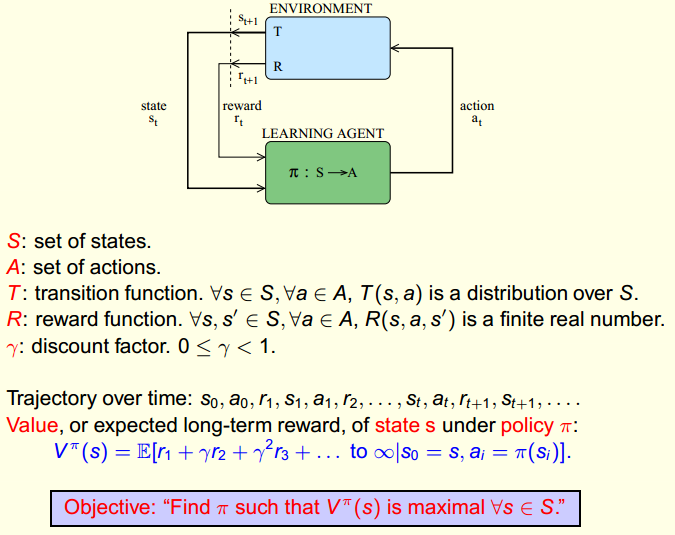


Reinforcement Learning:

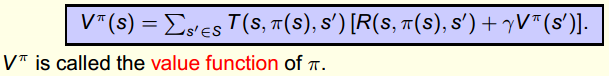


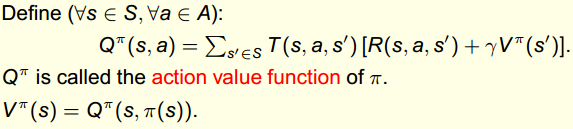
**Sequential Decision making situations**: situations in which a decision is made, an event occurs, another decision is made, another event occurs, and so on.

Markov’s Decision problem:



Bellman’s Equation:

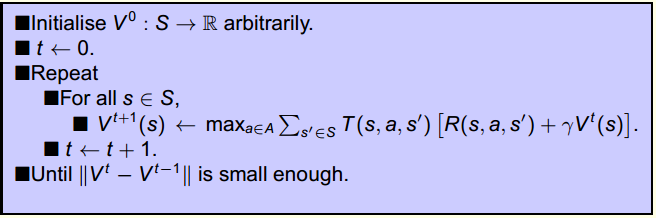




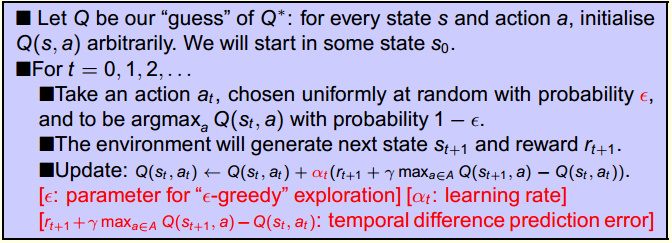
Bellman’s Optimality Equations:



**Value Iteration method to solve Bellman’s Optimality Equations:** (planning problem, computationally efficient)



**Q-learning**: a temporal difference learning algorithm: (Learning problem, sample efficient)



**Challenges**: presence of state aliasing and generalisation

Tree-search methods:

