LEARNING

Module 4

What is Learning?

- **learning** refers to the process by which a system improves its performance at a task over time through experience or data, rather than through explicit programming.
- AI systems learn patterns, make decisions, and adapt to new situations by analyzing and processing data

Forms of learning based on feedback

Consider a class of problems in which from a collection of input-output pairs a machine learns a function that predicts the output for new inputs.

- Unsupervised
- Reinforced
- Supervised
- Semi-supervised

Supervised learning

- In supervised learning the agent observes some example input-output pairs and learns a function that maps from input to output.
- Example: Predicting house prices based on size, location, etc.
- Common Algorithms: Linear regression, decision trees, support vector machines, neural networks.

Unsupervised Learning

- In unsupervised learning the agent learns patterns in the input even though no explicit feed back is supplied.
- Find hidden structure or patterns in data.
- Example: Grouping customers based on purchasing behavior (clustering).
- Common Algorithms: K-means, hierarchical clustering, PCA, autoencoders.

Reinforcement learning

- In reinforcement learning the agent learns from a series of reinforcements- rewards or punishments.
- Learn to make sequences of decisions to maximize cumulative reward.
- Example: A robot learning to walk, or an AI playing chess.
- Common Algorithms: Q-learning, Deep Q-Networks (DQN), policy gradients, actor-critic methods.

semi-supervised learning

- A small amount of labeled data + a large amount of unlabeled data.
- Improve learning accuracy by combining supervised and unsupervised approaches.
- Example: Image classification with few labeled images and many unlabeled ones.

Forms of learning based on different ways of acquiring knowledge

- Rote learning
- Learning by taking advice
- Learning in problem solving
- Inductive learning
- Explanation based learning
- Discovery
- Analogy

Learning Type	How Knowledge is Acquired	Key Features	AI Examples
1. Rote Learning	Memorizing input-output associations or facts exactly as given	- No reasoning or generalization- Direct recall- Simple storage and retrieval	Lookup tables, basic flash memory systems
2. Inductive Learning	Generalizing rules or models from specific examples	- Data-driven- Learns from multiple instances- Subject to overfitting/underfitting	Decision trees, neural networks, supervised ML

3. Explanation-Based Learning (EBL)	Generalizing from a single example using background knowledge and logical inference	- Requires prior domain knowledge- Learns by understanding causal structure	Deductive learners, logic-based agents
4. Analogical Learning	Learning by comparing new situations with previously solved similar ones	- Case-based reasoning- Relies on similarity and analogy- Human-like problem solving	Case-based reasoning systems, analogical inference in problem-solving
5. Discovery Learning	Actively identifying patterns or structure in data through exploration and hypothesis formation	- Often unsupervised- Novel rule or structure finding- Open-ended	Clustering algorithms, scientific discovery systems

6. Problem-Solving Learning	Learning to solve complex tasks by exploring solutions, often involving planning or search	- Goal-directed - -Often involves reasoning, planning, or decision trees	Planning agents, search-based solvers, STRIPS planners
7. Advice-Based Learning	Learning by incorporating external expert advice or guidance	- Can accelerate learning- - May come from humans or rules - Blends knowledge with learning	Interactive learning systems, learning from human feedback

Rote Learning

- Rote learning is the basic learning activity. Rote learning is a memorization technique based on repetition.
- It is also called memorization because the knowledge, without any modification is, simply copied into the knowledge base.
- As computed values are stored, this technique can save a significant amount of time.
- When Computer stores a piece of data, it is performing rudimentary form of learning
- In data caching, we store computed values so that we do not need to recompute
- When computation is more expensive than Recall, this strategy save significant amount of time
- Caching has been used in AI programs to produce performance improvement
- Such Caching is known as Rote learning
- It need
    - Organized storage of information
    - Generalization

Learning from advice

- There are two basic approaches to advice taking:
- Take high level, abstract advice and convert it into rules that can guide performance elements of the system. Automate all aspects of advice taking.
- Develop sophisticated tools such as knowledge base editors and debugging. These are used to aid an expert to translate his expertise into detailed rules. Here the expert is an integral part of the learning system

Learning from advice

- FOO (First Operational Operationaliser) is a programme developed in 1983 which accepts advice for playing hearts, a card game. It tries to convert high level advice (principles, problems, methods) into effective executable (LISP) procedures.

Learning in Problem Soving

- There are three basic methods in which a system can learn from its own experiences.

1. Learning by parameter adjustment

- Many programs rely on an evaluation procedure to summarise the state of search etc.
- Game playing programs provide many examples of this.
- However, many programs have a static evaluation function. So the basic idea of idea of parameter adjustment is to:

a. Start with some estimate of the correct weight settings.
b. Modify the weight in the program on the basis of accumulated experiences.
c. Features that appear to be good predictors will have their weights increased and bad ones will be decreased.

2. Learning with macro-operators

- Thebasic idea here is similar to rote learning: "Avoid expensive recomputation".
- Sequences of actions that can be treated as a whole are called macro-operators.

3. Learning by chunking

- Chunkinginvolves ideas similar to macro-operators and originates from psychological ideas on memory and problem solving. Its computational basis is in production systems.
- Chunking is the process of forming generalized rules or "chunks" from repeated experiences during problem solving.

Inductive Learning

- This involves the process of learning by example– where a system tries to induce a general rule from a set of observed instances.
- Inductive learning involves classification, that is, assigning to a particular input the name of a class to which it belongs.
- Classification is important to many problem solving tasks.
- A learning system has to be capable of evolving its own class descriptions. But it is sometimes difficult to construct class definitions by hand.
- This is particularly true in domains that are not well understood or that change rapidly.
- The task of constructing class definitions is called induction or concept learning

Explanation Based Leaning

- Humans appear to learn quite a lot from one example. The basic idea of explanation based learning is to use results from one example problem solving effort the next time around.
- An explanation based learning accepts four kinds of input:
- **A training example**: what the learning program “sees” in the world.
- **A goal concept**: A high level description of what the program is supposed to learn.
- **An operational criterion**: A description of which concepts are usable.
- **A domain theory**: A set of rules that describe relationships between objects and actions in a domain.
- From this, EBL computes a generalisation of the training example that is sufficient to de scribe the goal concept and also satisfies the operationality criterion.

Discovery

- Discovery is a restricted form of learning in which one entity acquires knowledge without the help of a teacher.
- Discovery can be of two types:
- Theory driven discovery
- Data driven discovery

Learning by Analogy

- Analogy involves a mapping between what might appear to be two dissimilar concepts.
- For example, consider the following statement: “Finding a good man is like finding a needle in a haystack.” This involves a mapping between two worlds: One, a world of men, good men and searching for good men; two, a world of small objects, a haystack and searching for a needle.
- There are two methods of analogical problem solving in AI:
  - transformational analogy and
  - derivational analogy.

Neural net Learning

- An Artificial Neuron Network (ANN), popularly known as Neural Network is a computational model based on the structure and functions of biological neural networks.
- It is like an artificial human nervous system for receiving, processing, and transmitting information.

- structure of a neuron in the human nervous system

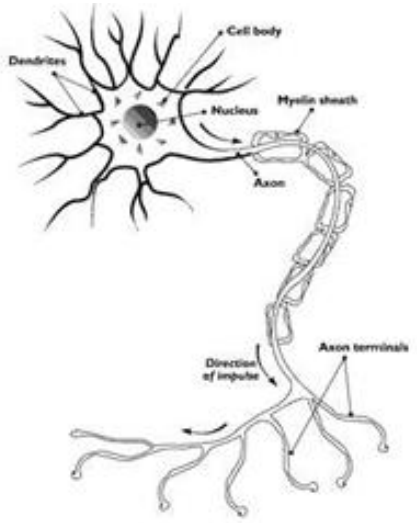


Figure 9.2: The structure of a neuron

- In an ANN,each neuron receives one or more inputs and produces one or more outputs.
- In its simplest form known as a perceptron a neuron takes a weighted sum of its inputs and produces an output 1 if the sum is greater than a certain pre-defined threshold value and 0 otherwise.

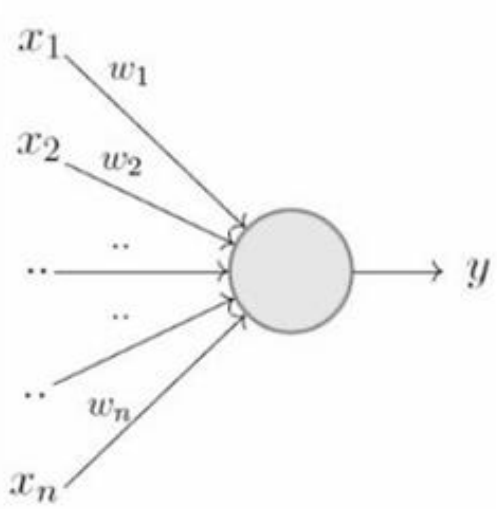
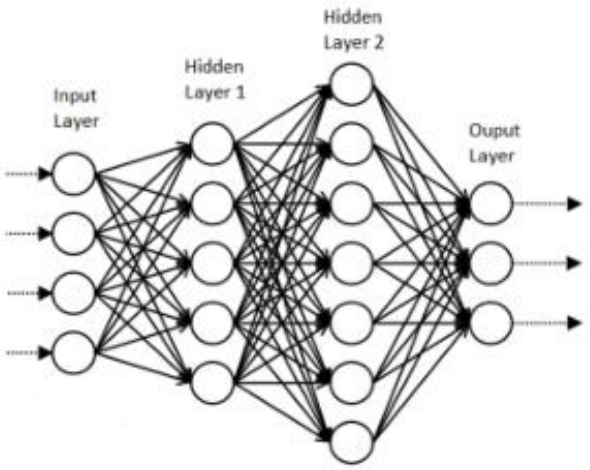


Figure 9.3: The model of a perceptron

$$y = \begin{cases} 1 & \text{if } w_1x_1 + w_2x_2 + \cdots + w_nx_n > w \\ 0 & \text{otherwise} \end{cases}$$

where w is some threshold value

- A neural networks consists of a number of neurons structured into different layers where each circle represents a neuron.
- Basically, there are three different layers in a neural network:



- 1. Input layer: All the inputs are fed in the model through this layer.
- 2. Hidden layers: There can be more than one hidden layers which are used for process ing the inputs received from the input layers.
- 3. Output layer: The data after processing is made available at the output layer

Learning in neural network

- During ANN learning, to change the input/output behavior, we need to adjust the weights.
- Hence, a method is required with the help of which the weights can be modified.
- These methods are called Learning rules, which are simply algorithms or equations.
- The objective is to find a set of weight matrices which when applied to the network should hopefully map any input to a correct output.

Learning in neural network

- In neural networks, learning is the process of modifying the values of the weights and the threshold.
- The learning takes place through an iterative process of “going and return” by the layers of neurons. The “going” is a forward propagation of information and the “return” is a backward propagation of information.

Applications

- 1. vehicle control, trajectory prediction
- 2. natural resource management
- 3. pattern recognition (radar systems, face identification, signal classification, 3D re construction, object recognition and more)
- 4. sequence recognition (gesture, speech, handwritten and printed text recognition)
- 5. medical diagnosis
- 6. finance (e.g. automated trading systems
- 7. data mining
- 8. visualization
- 9. machine translation
- 10. e-mail spam filtering

Genetic Algorithm

- A genetic algorithm is an algorithm inspired by the process of natural selection in the evolution of organisms.
- Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems.

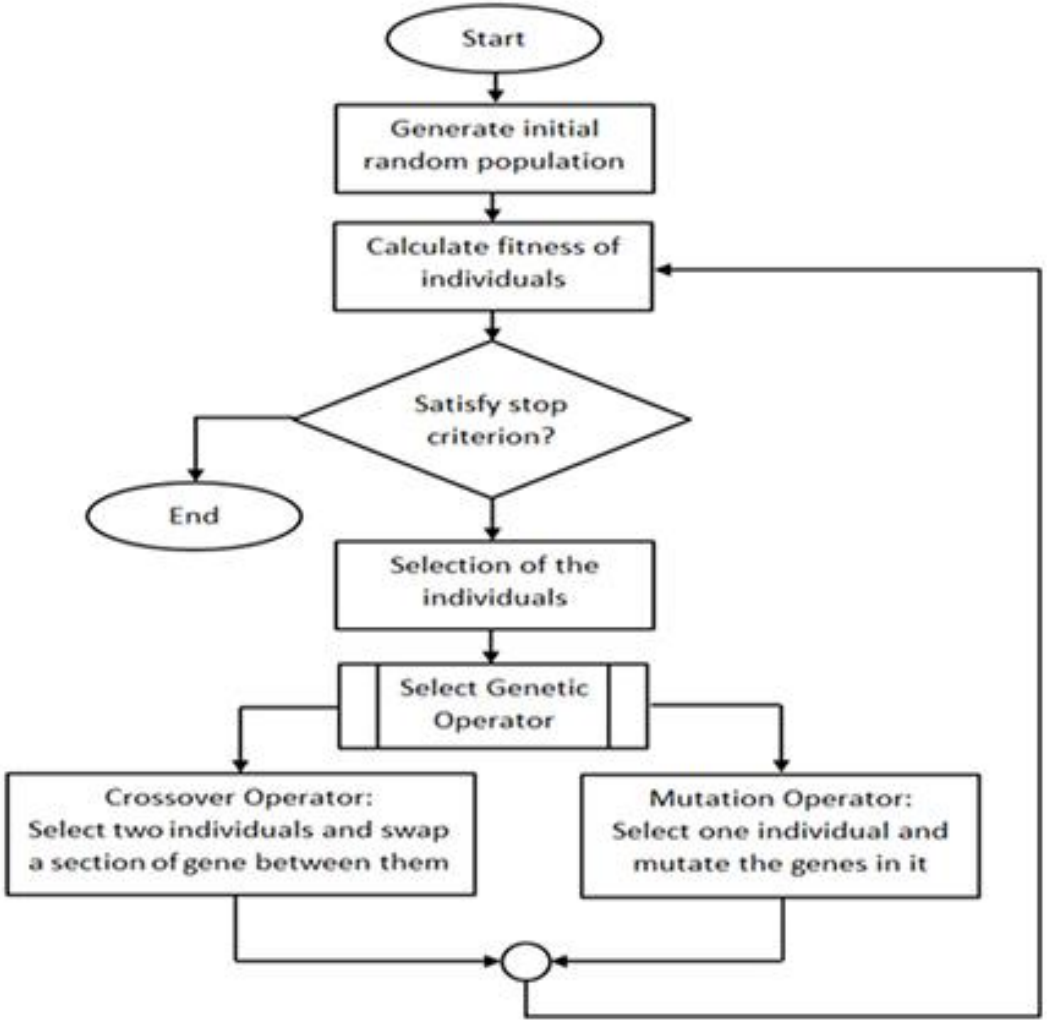


Figure 9.5: A genetic algorithm flowchart

- In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (called its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s. The evolution usually starts from a population of randomly generated individuals, and is an iterative process.

- The population in each iteration called a generation.
- In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved.
- The more fit individuals are randomly selected from the current population, and each individual's genome (the set of chromosomes) is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. The chromosome representation, selection, crossover (or, recombination), mutation, and fitness function computation are the key elements of the genetic algorithm.