Machine Learning
(Reinforcement learning and Neural Networks)

Prof K R Chowdhary

MBM University

September 20, 2024

Reinforcement Learning (RL)

⇒ Human learning is through feedback of their actions in the real-world. We always learn by interactions with the teacher (environment), which is in the form of *cause* and *effect* relations.

⇒ The environment or the world around us is like a teacher, but its lessons are often difficult to detect or grasp or analyze. The best example is learning by a dog where good actions are rewarded and bad actions are discouraged. RL has four components:

- policy,
- a reward function,
- a value mapping, and
- a model of environment.

⇒ The "reward function" is a relationship between *state* and *goal*, it maps each state into a reward measure, and indicate the need of that action to achieve the goal.

⇒ A RL system may not have a teacher to respond each action, the learner creates a policy to interpret feedback.

⇒ With the objective to maximize the expected reward, RL algorithms attempt to learn policies.

⇒ *State-space* of real-world problems contains infinitely large number of possible states features. So, the designer of any such task must pickup only the most relevant features. Say, for task: "Travel to Mumbai for work, and the weather report for New Delhi is not likely to be relevant."

⇒ From these we construct a feature set, and break it down into a number of subsets, so that each subset can learn specific concept of the domain. Some concepts/ feature-set subsets may be more important than others.

Example

Formulate a task of navigation to be carried out by an agent, with a goal to investigate best plan to go from point A to point B, and may choose a path and transport method of walking, driving, taxi,..

The feature-set of the agent's state-space may include,

- Positions of A and B,
- Raining (yes/no),
- Type of shoes of agent,
- Agent is with umbrella (Yes/no),
- Current time, and
- Day of week.

⇒ Using positions as features, the agent can learn the concept of position and basic path planing. The features: raining, shoes, ..., are useful in learning as how the weather governs the policy, and the features: time and day in a week may be useful in learning to handle traffic.

⇒ A conventional approach to solve this problem through RL is to learn in a six dimensional space (positions, raining, shoes, umbrella, time, weekday) when all the features are taken into account.

Some functions in Reinforcement Learning

In RL system, an agent recognizes itself in some state $p \in S$, then takes some action $a \in A$, and then recognizes itself in a new state q. The q is decided by the agent's *transition function* T:

$$T(S \times A) \rightarrow S. \quad (1)$$

Also, the agent receives a reward r for arriving to q, based on the *reward function* R:

$$R(S \times A) \rightarrow \mathbb{R}. \quad (2)$$

A *value function* $V^{\pi}(p)$ is based on average sum-of- rewards received when an agent starts in p, and enter into q, following a policy π. Relation between *value function* and *optimal policy* $V^*(p)$ is:

$$\forall \pi, p : V^*(p) \geq V^{\pi}(p). \quad (3)$$

The RL used in many real-world domains, where discounted total reward is optimized.

⇒ Artificial Neural Networks (ANN) is a brain model.

⇒ The science of machine learning is mostly experimental as there is a no universal learning algorithm yet. Given a number of tasks, none can make a computer to learn every task well.

⇒ We are fairly good at general learning abilities due to which we are able to master number of tasks, like playing chess and playing cards.

⇒ A knowledge-acquisition algorithm is always required to be tested on learning tasks and data.

⇒ There is no method to prove that the given algorithm will be consistently better for all the situations.

⇒ These arguments suggest and might serve as inspirations for building machines with some form of general intelligence.

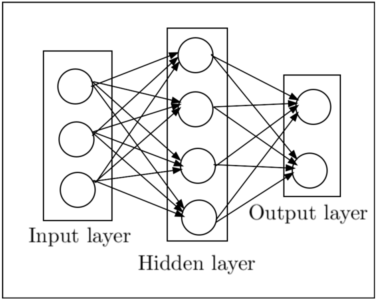
⇒ Basic unit of brain for performing the computation is a cell, called *neuron*, each one of them sends a signal to other neurons through very small gaps between the cells, called *synaptic clefts*.

⇒ The property of any neuron of sending a signal through this gap, and the amplitude of the signal together, is called as *synaptic strength*.

⇒ As a neuron learns enough, its synaptic strength increases, and in that situation, if it is stimulated by an electrical impulse, there are better chances that it would send messages to its neighboring neurons.

⇒ A neural network consists of two or more layers: an input layer, zero or more hidden layers, and an output layer. Data is ingested through the input layer, modified in the hidden layers and the output layer, based on the weights applied. Using *iterations*, a neural network continuously adjusts and makes inferences until some stopping pt. reached.

Neural Networks: Architecture of a simple NN



⇒ Majority of these algorithms are based on *supervised learning*.

⇒ As an example, let picture of sunrise is associated with caption: "Sunrise". Hence, goal of learning algorithm is: take the photo as an input, and produce name of object in the image, as output, i.e., "sunrise."

⇒ The process of transforming an input to the output is math *function*.

⇒ Synaptic strength (a numerical value) produce this function, which is the solution to the learning through ANN.

⇒ In fact, when we want to teach to the algorithm "what the sunrise is", then the algorithm should recognize any sunrise, even the one for which we have not trained it! This is the goal of ML.

NN: Properties

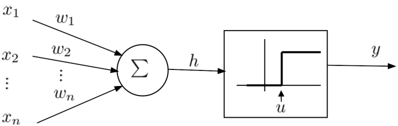
Important properties:

- Learning ability,
- Massive parallelism,
- Adaptability,
- Fault tolerance,
- Distributed representation and computation,
- Generalization ability, and
- Low energy consumption.

Thus, ANN has many applications. Although their models vary, the most common is in learning and computation, where ANN is the basic processing unit. Each processing unit has following properties:

- an activity-level to represent neuron's polarization state,
- an output value represents firing rate,
- a set of input connections,
- synapses on the cell and its dendrite,
- a bias value to represent an internal resting level of a neuron, and
- output connections to represent neuron's *axonal* projections.

⇒ Each connection of a neuron has an associated weight, called *synaptic strength* or weight w_i of *ith* input, that influences the effect of the incoming input on the activity of the unit. The weight is either +ve (excitatory) or -ve (inhibitory).



⇒ The basic model for artificial neuron with binary threshold is shown above.

⇒ The mathematical neuron computes the weight as the *sum* of its *n*-input signals $x_1, \ldots, x_n$.
⇒ The generated output is 1, if the *sum* > *threshold* u, else, output is 0. The o/p is:

$$y = \theta\Big(\sum_{i=1}^{n} w_i x_i - u\Big), \qquad (4)$$

⇒ The $\theta(.)$ is *unit step* function at 0. Threshold u is other weight; $w_0 = -u$ is attached to the neuron with a constant input $x_0 = 1$.

An overly simplified one of true biological neuron

⇒ A properly chosen weights allows a synchronous arrangement of such neurons to perform universal computations.

⇒ There is a crude analogy of this neuron model to biological neurons as follows: wires and interconnections model the *axons* and *dendrites*, respectively, in the biological neuron.

⇒ The *connection weights* in here correspond to *synapses* in biological neuron, and threshold function approximates the activity in biological neuron.

⇒ The ANNs can be considered as weighted directed graphs, where artificial neurons act as nodes, and directed edges with weights are connections between neurons, and between outputs and inputs of neuron.

⇒ Based on the connection pattern an ANN can be classified as:

- *Feed-forward networks:* The direct graphs have no loops.
- *Recurrent feedback networks:* Have loops, due to feedback connections.