Artificial Intelligence: Week 1

1 What is an AI?

There are various definitions of an AI, ranging from thinking humanly and rationally to acting humanly and rationally. The *turing test*, is test in which a human interrogator interacts with a machine, sending it messages back and forth, and a machine passes if it fools the human into thinking that the messages are being sent to them by a human. For this a computer needs: **natural language processing, knowledge representation, automated reasoning and machine learning**. To pass the *total turing test* a computer would additionally need **computer vision and robotics.**

2 Intelligent Agents

An **agent** is just something that acts and a **rational agent** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome. **Percept** means the agent's perceptual inputs at any given time, and a **percept sequence** is the complete history of everything the agent has ever perceived. The **agent function** is an abstract mathematical description that maps a given percept to an action; an **agent program** is a concrete implementation of the agent function, running within some physical system. *It is better to design a performance measure according to what one wants in an environment, then how one wants an agent to behave.*

The proper definition of a rational agent is *for each possible percept sequence, 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.* An **omniscient** agent knows the actual outcomes of its actions.

2.1 The Nature of Environments

An environment is defined as **PEAS**: performance measure, environment, actuators and sensors. There are different types of environments, namely:

- **Observable vs Partially-Observable**: it is observable when the agents' sensors have complete access to the environment's state at all times

- **Single agent vs Multi agent**: there could be multiple agents in an environment. There is also a question of what must be considered an agent. This gives way to the concept of **competitive** vs **cooperative** environments.

- **Deterministic vs Stochastic**: If the next state can be completely determined by the current state and the action executed by the agent, then it is deterministic; and stochastic otherwise. An environment is **uncertain** if it is not fully observable or not deterministic. Note that a **nondeterministic** environment is one where each action is characterized by its possible outcomes, but no probabilities are attached to them.

- **Episodic vs Sequential**: In an episodic environment the agent's experience is divided into atomic episodes. The next episode doesn't depend on the action taken in the previous episode.

- **Static vs Dynamic**: If an environment can change when an agent is deliberating, then it's dynamic, and is static otherwise. If the environment doesn't change when deliberating but the performance score does, then we call it **semi-dynamic.**

- **Discrete vs Continuous**: The distinction here applies to the state of the environment, the way time is handled and the percepts and actions of the agent. For example, chess having a discrete set of states; the same doesn't apply for taxi driving.

- **Known vs Unknown**: This applies to the agent's state of knowledge about the "laws of physics" of the environment. Note that it's possible that a known environment is partially observable like solitaire. Conversely, an environment can also be unknown and fully observable, like in a video game, one can see the state but one doesn't know the control until one tries to play.

2.2 The Structure of Agents

There are four basic types of agent programs:

- **Simple Reflex Agents**: Agents that select the current action based on the current precepts and ignoring the rest of the precept history. It is also important to note that these types of agents are usually implemented in a fully-observable environment.

- **Model-Based Reflex Agents**: The best way to handle a partially observable environment is to keep some sort of an internal representation of the aspects of the environment not currently observable. Therefore, an agent should have some sort of knowledge about how the world works and the agents who have said knowledge are called model-based agents.

- **Goal-Based Agents**: These types of agents consider how close they get to a goal, in addition to have a model of how their environment works. These types of agents are also quite flexible as they can update their actions on-the-fly depending on their goals and the feedback they get from the environment.
- **Utility-Based Agents**: Since the previous model does not differentiate between how it gets to its goal, and which state would make it more happy, it is not efficient. Therefore a utility function is needed to determine just that i.e. it is an internalization of its performance measure. The previous model will also fail when there are conflicting goals or when there are several goals the agent can aim for, none of which can be achieved with certainty; in both cases, a utility function can dictate which action to take to maximize expected utility.

There are different ways an agent can represent the world around it:

- **Atomic**: Each state is of the world is indivisible: it has no internal structure.
- **Factored**: Each state is split up into a fixed set of variables and attributes, each of which can have a value. Uncertainty can also be represented in this representation.
- **Structured**: This type of representation has objects and their relationship with each other.