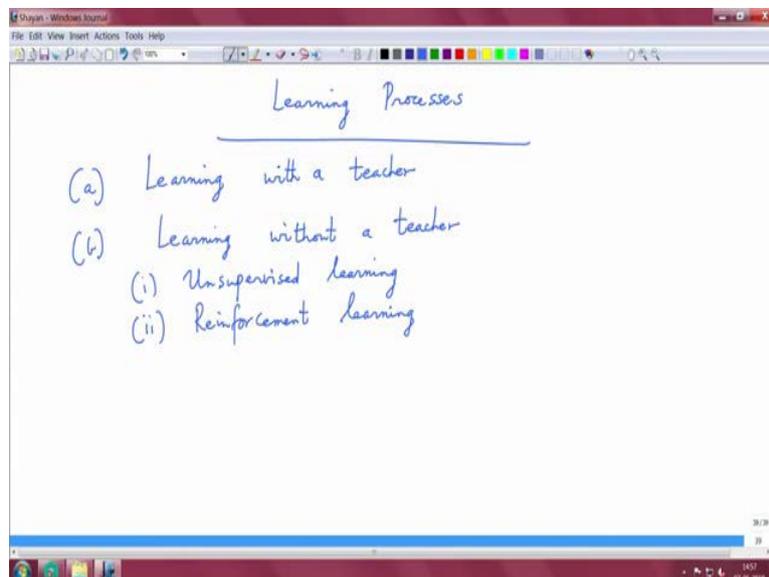


Neural Networks for Signal Processing – I
Prof. Shayan Srinivasa Garani
Department of Electrical Systems Engineering
Indian Institute of Science, Bengaluru

Lecture – 07
Learning Processes

Let us get started with learning processes.

(Refer Slide Time: 00:33)

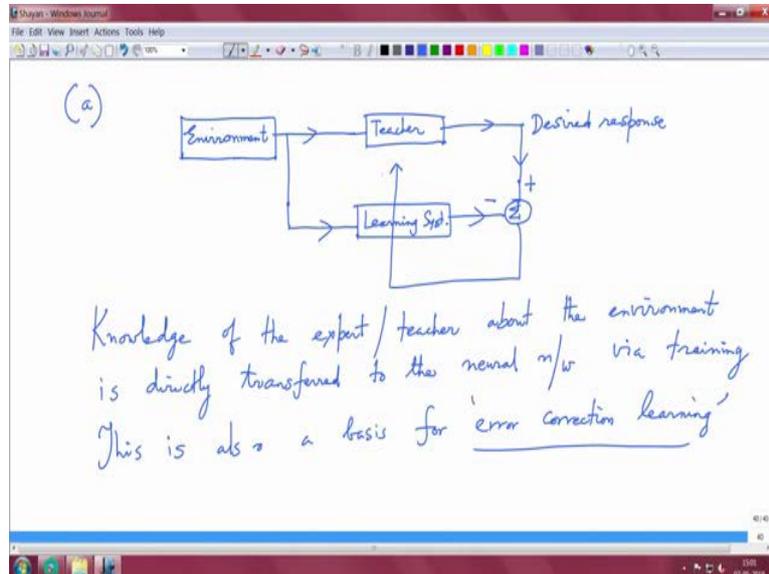


When considering learning processes, we encounter several methods. Firstly, there's supervised learning, where a teacher provides labeled examples to guide the learning process. Secondly, there's unsupervised learning, where the system identifies patterns and structures in data without explicit guidance. Lastly, reinforcement learning involves learning through trial and error, where actions are reinforced or discouraged based on their outcomes. Visualizing these processes through a block diagram allows us to grasp their mechanics and appreciate how each method contributes uniquely to the learning process.

Let's delve into the first learning mechanism, which involves learning with the guidance of a teacher. Imagine an environment, perhaps a plant or any engineering system, where outputs are generated. These outputs are then sent to a teacher, an expert, who provides desired responses. Simultaneously, these outputs are also fed into a learning system. Initially, the learning system

doesn't know the correct responses to produce. It generates outputs, and the expert teacher provides feedback, indicating desired responses.

(Refer Slide Time: 01:46)



This feedback creates an error signal that the learning system aims to minimize. This process drives the adaptation of the learning system to better align its outputs with the desired responses provided by the teacher.

So, what happens is that the knowledge held by the teacher about the environment is directly imparted to the neural network through a training procedure. The synaptic weights of the network, in essence, encode the environmental variables learned from the teacher's expertise. This forms the foundation of error correction learning.

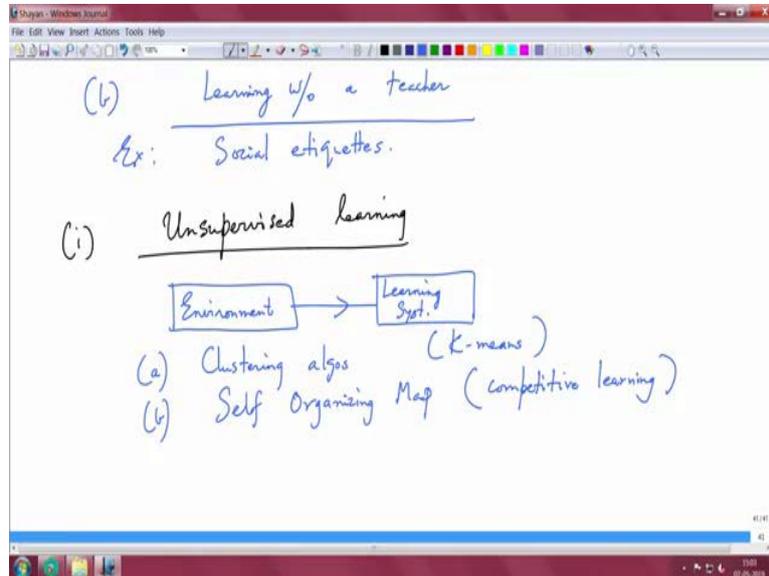
The critical details include formulating the objective function. Here, we have an error signal where the learning system produces a response that deviates from the desired response. Determining the objective of this error metric influences the geometric nature of the learning problem, shaping how the system adapts and improves over time.

So, the other facet of learning without direct guidance is akin to learning social etiquettes. In this scenario, behaviors are learned through observation and immersion in the environment, without explicit instruction.

Translating this concept into a neural network learning process involves two main approaches: unsupervised learning and reinforcement learning. Let's delve into unsupervised learning first.

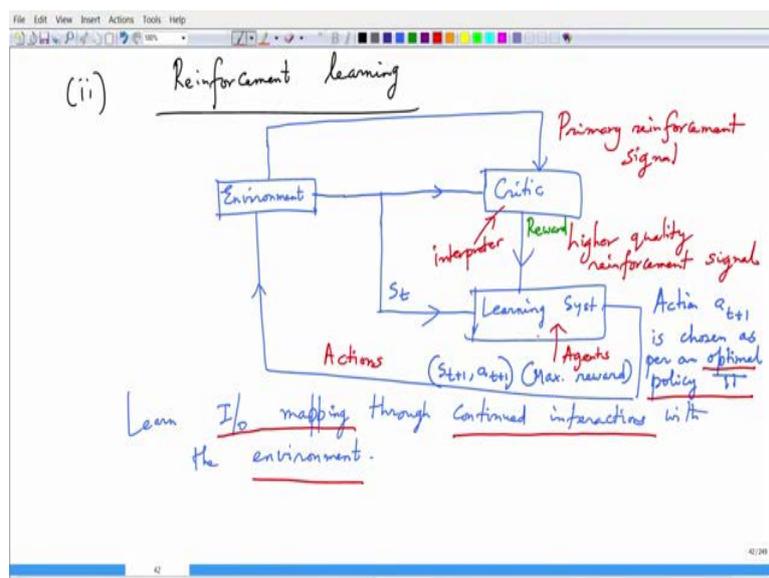
In unsupervised learning, the system interacts with the environment, extracting feature vectors autonomously. These feature vectors are then input into the learning system.

(Refer Slide Time: 05:42)



A prime example of unsupervised learning can be found in clustering algorithms. A specific example is the self-organizing map (SOM), which represents a more complex form of unsupervised learning. While SOM does perform clustering, it goes beyond traditional clustering methods by incorporating a competitive learning mechanism.

(Refer Slide Time: 08:45)



To distinguish SOM from general clustering algorithms like the rudimentary k-means, consider the following: When input patterns from the environment are fed into a learning system, the system clusters these inputs into distinct classes and partitions the output space. This process occurs without the guidance of a teacher or a predefined desired response. Instead, the network autonomously learns these classifications through a suitable learning mechanism, enabling it to identify and organize patterns in the data.

Now, let's explore the concept of reinforcement learning. Before diving into the schematic of the learning process itself, I'd like to motivate this task with a few examples.

Imagine we have a robot with four, six, or even eight legs that must navigate through a complex terrain. In this scenario, the robot receives rewards for taking specific actions that enable it to traverse this environment successfully. As the robot continuously learns from the state of the environment, it adjusts its actions to optimize how it moves or crawls across the terrain.

Another example is learning music from a teacher. In this case, the actions I take aim to maximize my reward, which would be learning more songs during a music session. To learn more songs, I need to take certain optimal steps, such as singing what I'm taught accurately. This process involves feedback to the environment, which in this case is the teacher. Additionally, a critic can be part of this learning experience, offering feedback and guidance. Imagine the music learning scenario, where the teacher provides constructive criticism to help refine the learning process.

I have a teacher guiding me in music, and as I learn, I take specific actions during the lesson. Meanwhile, a judge, who observes the interaction between the teacher and myself, provides rewards based on my performance. These rewards are a consequence of the actions I've taken. This scenario exemplifies a learning system where I aim to optimize my actions to maximize the rewards, or the expected amount of rewards, I receive in a statistical sense.

With these motivating examples in mind, let's delve into the systematic process of reinforcement learning. We begin with an environment, this could be any context within artificial intelligence, whether it's learning music, navigating a terrain, or even a child exploring and sampling their surroundings. The state of the environment, denoted as s_t , is provided to both the critic and the learning system.

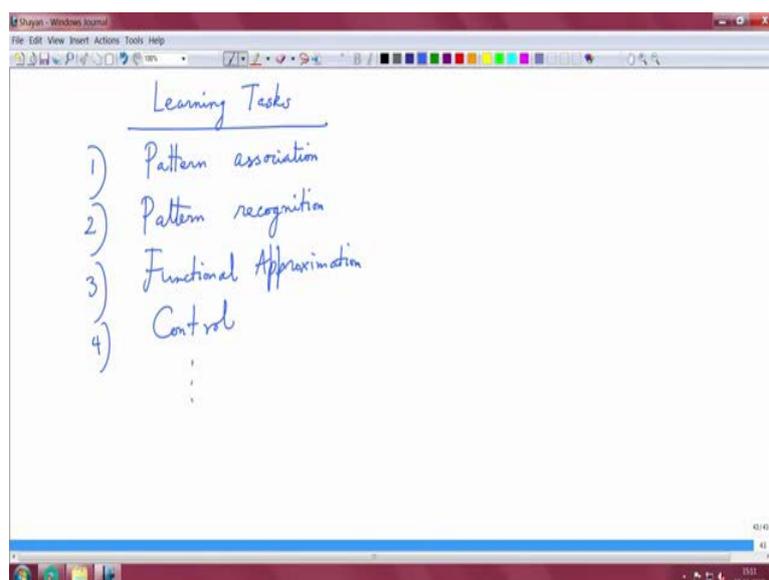
Other environmental variables can also be fed to the critic, which we refer to as the primary

reinforcement signal. In this context, the critic could be a judge who observes the environment and assigns a reward to the learning system, or agent, based on its performance. This agent performs actions according to an optimal policy, π , to maximize the reward. During this process, the action a_{t+1} is taken to maximize the rewards, leading to a change in the environmental state to s_{t+1} . The new state s_{t+1} is then fed back into the environment, creating a continuous reinforcement loop.

Learning occurs through an input-output mapping driven by continued interactions with the environment. In this case, the environment can be likened to a music teacher, with the state changing at each small time step t , where s_t progresses from $t = 0$ to some final time T . The agent, which is learning music, takes actions guided by the rewards it receives from the critic. The objective is to maximize the expected rewards over time, with each action a_{t+1} being chosen optimally to achieve this goal.

This reward-action system, encompassing reward, action, and state, is dynamically evolving through continuous interactions with the environment, forming the basis of how learning occurs in reinforcement learning. This mode of learning leverages the principles of reinforcement to adapt and improve over time. Optimal decisions within this framework can often be mathematically formulated using Markov decision processes and other stochastic process theories, which are selected based on the specific random processes we encounter over time.

(Refer Slide Time: 15:24)

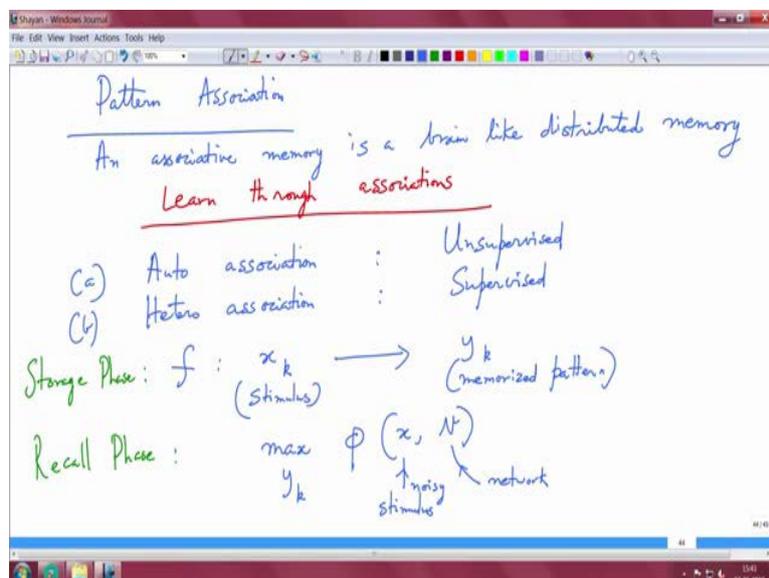


This overview completes our discussion on the details of reinforcement learning. With this

understanding, you can set up fascinating gaming problems and other applications that utilize the principles of reinforcement learning. The topic itself is rich and complex, offering ample material for exploration in a dedicated course.

Now, let's explore the different types of learning tasks, which are akin to the processes occurring in biological systems, such as our brain. These tasks include pattern association, pattern recognition, functional approximation, and control, among others. Most AI problems tend to fall into one of these four categories, though there are additional categories of learning as well. Let's examine these learning tasks qualitatively, and as we delve into the algorithms, we will gain a quantitative understanding of what we are discussing.

(Refer Slide Time: 16:39)



The first concept to explore is pattern association. This involves an associative memory, which functions as a distributed memory system akin to the brain's own mechanisms. In this process, learning occurs through associations. But what does learning through associations mean? Essentially, you repeatedly input data into a network, allowing it to learn over time. This involves two phases: storage and recall. During the storage phase, the network learns and stores the information. When a stimulus similar to the input is presented, it triggers a response, reflecting what the network learned during the learning process.

Let's delve into the two modes of pattern association: auto-association and hetero-association. Auto-association is typically unsupervised, whereas hetero-association is usually supervised.

If we describe this process in terms of mapping, we have a stimulus x_k and a memorized pattern y_k that corresponds to this stimulus. Here, k represents the pattern, and the mapping occurs from x_k to y_k , which constitutes the storage phase. Additionally, there is a recall phase, where a noisy version of the stimulus is presented to evoke the learned response.

It is not exactly the original stimulus x_k that the network learns, but rather a noisy version of x_k . The goal is to optimize the retrieval of memorized patterns by presenting this noisy version to the network. This noisy input acts as a stimulus that the network has learned through training. When this noisy version is input, the network must retrieve the memorized pattern and output what it learned as y_k . This process is known as auto-association.

A brief note: I made a minor error previously by referring to the function with the same notation, f , as used earlier. During the recall phase, you might use a different function, which can be denoted as ϕ . In this phase, the objective is to maximize ϕ , which evaluates all memorized patterns and returns the pattern closest to the noisy stimulus.