Neural Networks for Signal Processing – I

Prof. Shayan Srinivasa Garani

Department of Electronic Systems Engineering

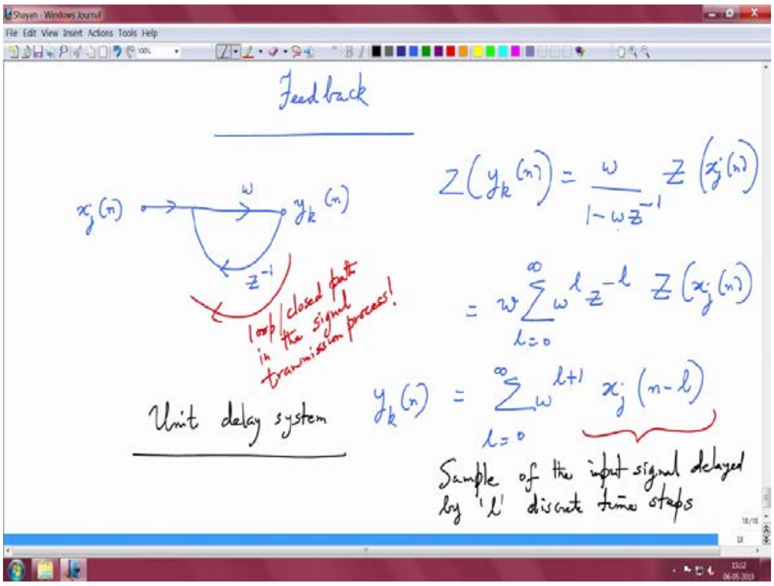
Indian institute of Science, Bengaluru

Lecture – 04

Feedback and network architectures

Let us delve into the role of feedback in non-linear systems. When there are closed paths for signal transmission, we refer to this as feedback. In 1975, Freeman stated that feedback is present in nearly every part of the nervous system of an animal. Thus, feedback is both central and fundamental to all systems. To better understand this concept, let us examine it through a signal flow graph.

(Refer Slide Time: 00:58)



Let's consider $x_j(n)$, a feedforward weight w, and $y_k(n)$. There is a unit delay in the feedback, forming a closed path in the signal transmission process. This closed loop is an example of a unit delay system, which is commonly seen in transfer functions discussed in literature.

Before we delve deeper, it's important to understand the motivation for incorporating feedback into these systems. As mentioned by Freeman in 1975, feedback is present in nearly every part of the nervous system of animals. This theory posits that feedback plays a crucial role in the functionality of neural networks, and there is substantial evidence supporting this.

When we reflect on our actions and inferences, the role of feedback becomes evident. For instance, when making a decision while grading an exam paper, I don't simply grade the paper once and stop. Instead, I consider the overall performance, especially if the student's performance is borderline. This process involves revisiting and reassessing the situation, which is essentially a feedback loop. By doing so, I adjust my criteria and thresholds accordingly, illustrating the fundamental importance of feedback in decision-making processes.

Feedback plays a crucial role in our decision-making processes, thinking, memory, and learning. It is an essential concept, but let's return to our primary discussion.

Let's analyze the Z-transform of $y_k(n)$. Simplifying the system, we see a feedforward gain w and a feedback path z^{-1}. Applying the Z-transform, we get:

$$Y(z) = \frac{W}{1 - Wz^{-1}} \cdot X_j(z)$$

We can further simplify this by expanding the denominator polynomial into a power series:

$$Y(z) = W \sum_{l=0}^{\infty} (Wz^{-1})^l \cdot X_j(z)$$

This can be rewritten as:

$$Y(z) = \sum_{l=0}^{\infty} W^{l+1} z^{-l} \cdot X_j(z)$$

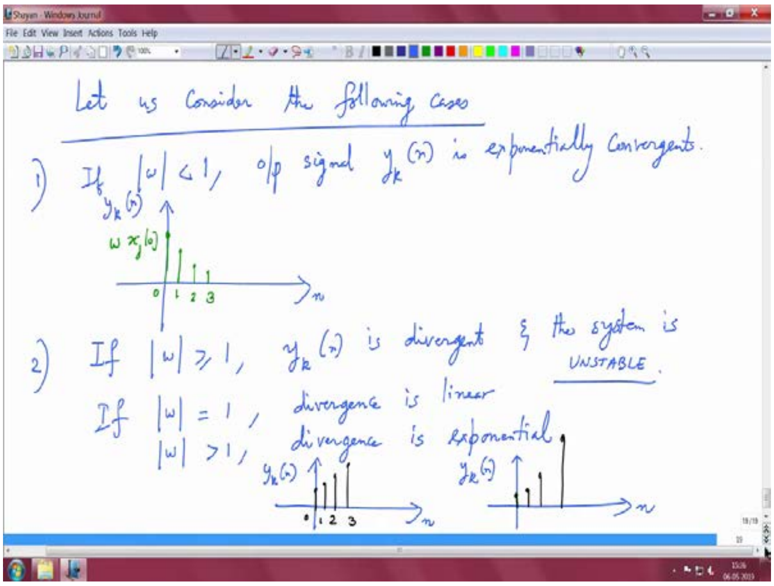
Taking the inverse Z-transform, we get:

$$y_k(n) = \sum_{l=0}^{\infty} W^{l+1} \cdot x_j(n - l)$$

This expression emphasizes that the input signal $x_j(n)$ is sampled and delayed by l discrete time steps, then scaled accordingly. This indicates that the sample at time n depends not only on the input at time n, but also on past inputs. This is the essence of feedback.

Fundamentally, this means there is memory associated with the system due to the pole present in the Z-transform equation. Feedback introduces this memory, allowing the system to retain information about previous inputs, which is crucial for learning and adaptive processes.

(Refer Slide Time: 06:55)



Let's delve into the following scenarios:

1. When |w| < 1:

The output signal $y_k(n)$ is exponentially convergent. This means if you plot $y_k(n)$ against time n, you'll observe it declining exponentially. Initially, at time step n = 0, you have w \cdot $x_j(0)$. As time progresses (at time steps 1, 2, 3, etc.), the signal starts to decrease exponentially.

2. When $|w| \geq 1$:

The behavior of $y_k(n)$ changes significantly, indicating divergence and instability in the system.

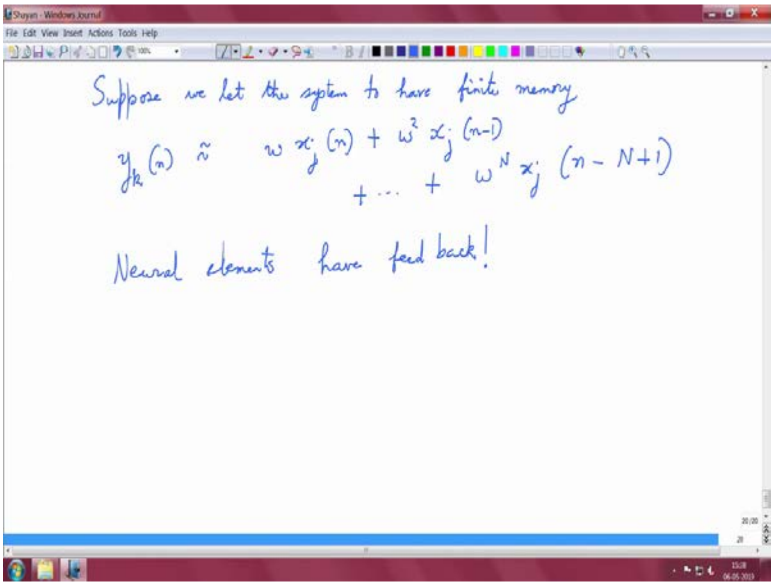
For |w| = 1:

The divergence is linear. A simple sketch would show a linear increase over time, depicting this linear divergence.

For |w| > 1:

The divergence is exponential. A sketch in this case would reveal a steep exponential rise, illustrating exponential divergence.

(Refer Slide Time: 10:23)



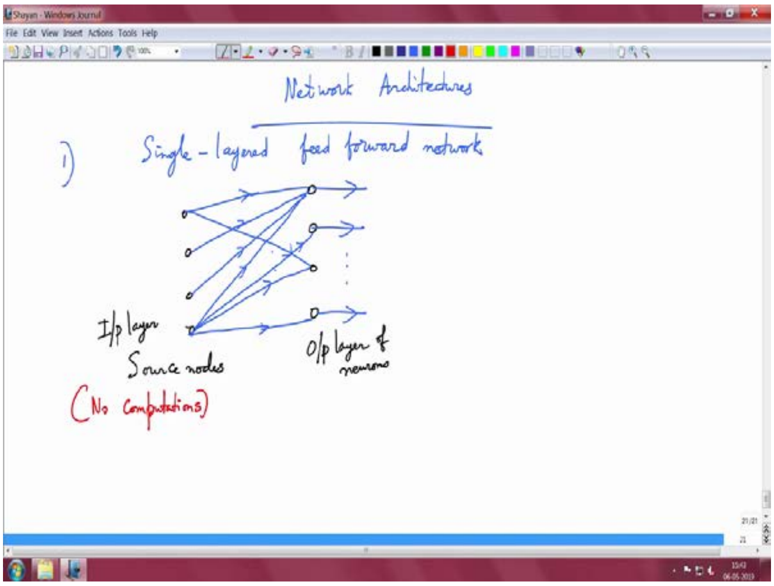
Let's consider allowing the system to possess finite memory, a crucial aspect that simplifies our analysis. We approximate $y_k(n)) \; as \; (wx_j(n) + w^2x_j(n-1) + \cdots + w^nx_j(n-N+1)$, where we let it span N time steps. These terms scale accordingly with w^2 and so forth.

This representation inherently includes feedback. All neural elements incorporate this feedback mechanism. Finite memory implies that inputs are combined only up to a specified past time step, weighted by associated scale factors, to determine the output signal within this memory range. This feedback mechanism is integral to neural network operations, facilitating dynamic adjustments based on past inputs up to a defined memory limit.

And in general, if you think about non-linear systems analysis of the dynamics, a particularly non-linear dynamics is very complicated and when we think about networks that is when you think about network architectures, where we have recurrence in neural networks, we have these recurrent neural networks where there is feedback mechanism from the output or the input at that step we can think about architectures as to what is the kind of memory that we want to bring in and how many units of memory we want to bring in and we can look at these sophistications, when we revisit the network architectures.

Having discussed the feedback mechanism, neural network representations is digraphs and the neuronal models, we are sort of ready to understand networks network architectures right.

(Refer Slide Time: 13:03)



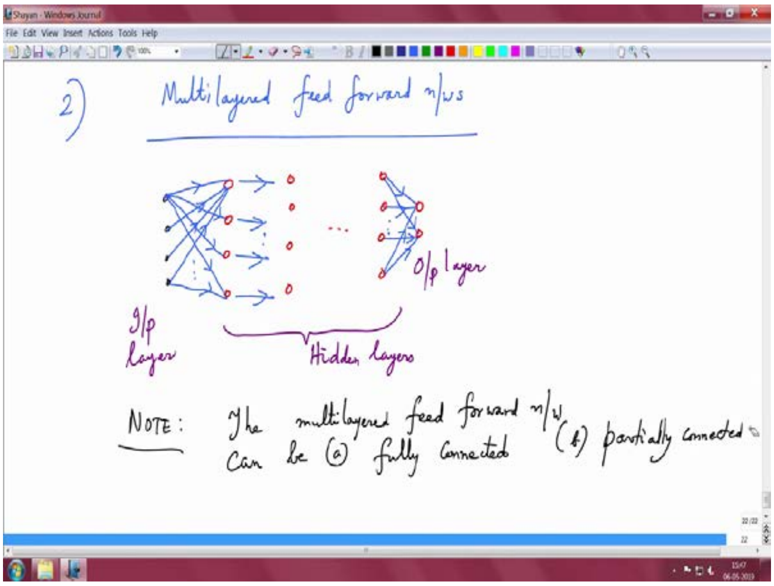
Let's explore the structure of the first network architecture, which is a single-layer feedforward network. This is the simplest form of a feedforward network. In this architecture, we have an input layer and an output layer of neurons connected by weights. It's important to note that this network is fully connected, meaning each neuron in the input layer is connected to every neuron in the output layer.

Visualizing this, the input layer acts as the source nodes where no computation occurs; they simply receive inputs. These inputs are then transmitted through synaptic links to the neurons in the output layer, where computations take place. Here, each connection represents a synaptic link responsible for transmitting and transforming the input signals into outputs through computational processes.

This single-layer feedforward network serves as a foundational model. We can extend this concept to more complex architectures by introducing multiple layers of neurons, known as multi-layered feedforward networks. This expansion allows for deeper and more intricate computations and transformations of data as it passes through successive layers of interconnected neurons.

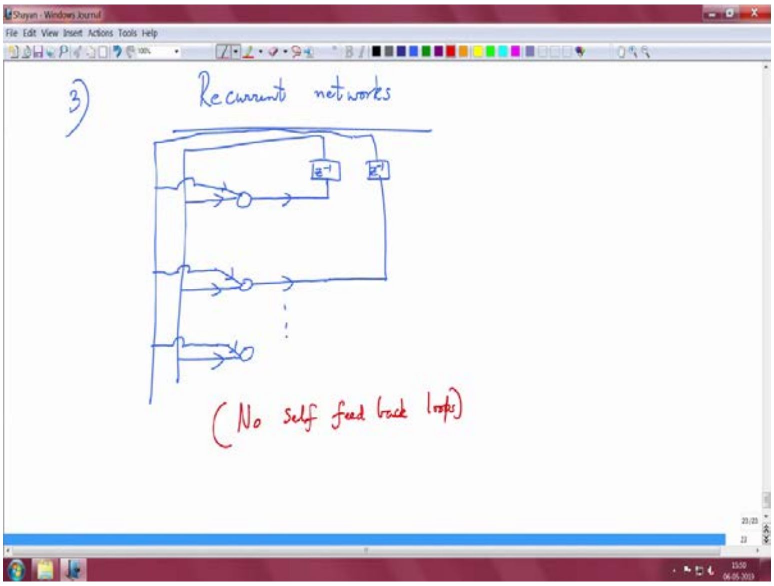
The concept here is quite straightforward. We begin with the source nodes at the input, typically marked in red for the hidden layers. Connections extend from these source nodes to the first hidden layer, and so forth, often fully interconnected. This pattern continues across various hidden layers, each layer interconnected with the next. From the final hidden layer, connections spread to the output layer, forming a network where all possible connections are considered.

(Refer Slide Time: 15:26)



It's crucial to note that a multi-layered feedforward network can either be fully connected or partially connected. This distinction significantly impacts learning capabilities. A fully connected network tends to offer generalization abilities, whereas a partially connected one may specialize in particular tasks. The choice between these configurations depends on the specific application and the desired knowledge representation within the network.

(Refer Slide Time: 19:24)



In a straightforward feedforward network like this, there are no feedback loops. Information flows strictly in one direction, from input to hidden layers to output. There are also no lateral connections between neurons within the same layer, nor is there feedback from the output back to preceding layers.

Introducing feedback into the network leads us into the realm of recurrent networks, where connections allow information to circulate back within the network, enabling more complex temporal dynamics and the potential for learning sequences and patterns over time.

Let's delve into the concept of recurrent networks and visualize their architecture. In this example, consider a scenario where the output signal is delayed and fed back into the inputs. This feedback loop allows the network to integrate past information into the current computation.

Similarly, another output can be delayed and fed back, extending the network's capability. While this specific example doesn't include self-feedback loops, theoretically, recurrent networks can incorporate such loops. The presence of these feedback mechanisms profoundly influences the network's learning capabilities.

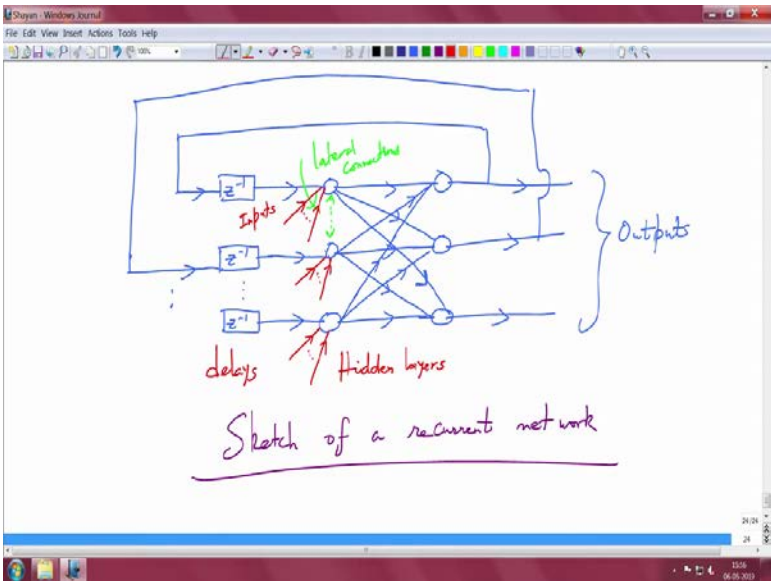
We can generalize this concept further by extending delays not only to outputs but also within hidden layers, interconnecting them to form a complex network of feedback paths. This architecture enables the network to process sequential data and capture temporal dependencies, essential for tasks such as time-series prediction and sequence modeling.

These delays represent the hidden layers, allowing for the stacking of multiple layers as needed. The outputs are then connected via a feedback mechanism through delays to influence the processing elements within the hidden layers.

This sketch illustrates an example of a recurrent network, highlighting the feedback mechanism through these delays from outputs back to inputs. The network's complexity can vary with different configurations of feedback connections, complicating its architecture. Notably, this recurrent network architecture typically does not include lateral connections between processing elements within the same layer.

When discussing lateral connections, we refer to connections between processing elements within the same layer. These connections can facilitate feedback and other interactions across the network. Typically, in a straightforward recurrent network, lateral connections are not employed. This distinction is crucial to clarify.

(Refer Slide Time: 21:52)



In the sketch presented earlier, while all the connections were detailed, what was not explicitly shown were the inputs. These inputs, marked in red, are crucially placed at the input layer. They can be uniformly provided to all neurons, which represents one architectural approach. Alternatively, inputs can be selectively allocated to individual neurons, offering another configuration.

What remains fundamental is the presence of inputs at the input layer, followed by hidden layers and outputs that feedback through these delays. This encapsulates the basic structure of a recurrent network. Remember, while lateral connections are absent in this design, inputs, delays, and outputs are integral components. This concludes our overview of the recurrent network architecture. We stop here.