Neural Networks for Signal Processing – I

Prof. Shayan Srinivasa Garani

Department of Electronics System Engineering

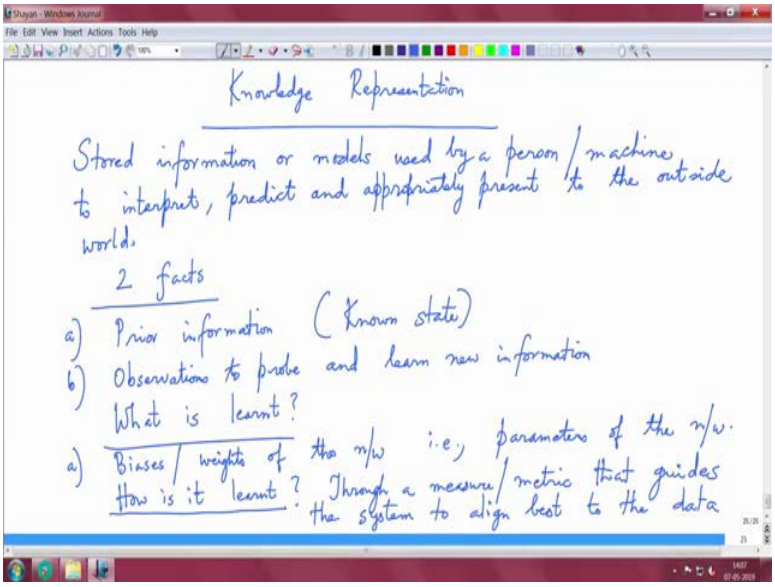
Indian Institute of Science, Bengaluru

Lecture – 05

Knowledge Representation

When we consider the concept of knowledge representation, what exactly do we mean by it? Knowledge representation refers to the stored information within a model that can be utilized to interpret, predict, and appropriately present information to the outside world. To reiterate, knowledge representation involves the use of stored information or models by a person or machine to interpret, predict, and effectively present data to the outside world.

(Refer Slide Time: 01:00)



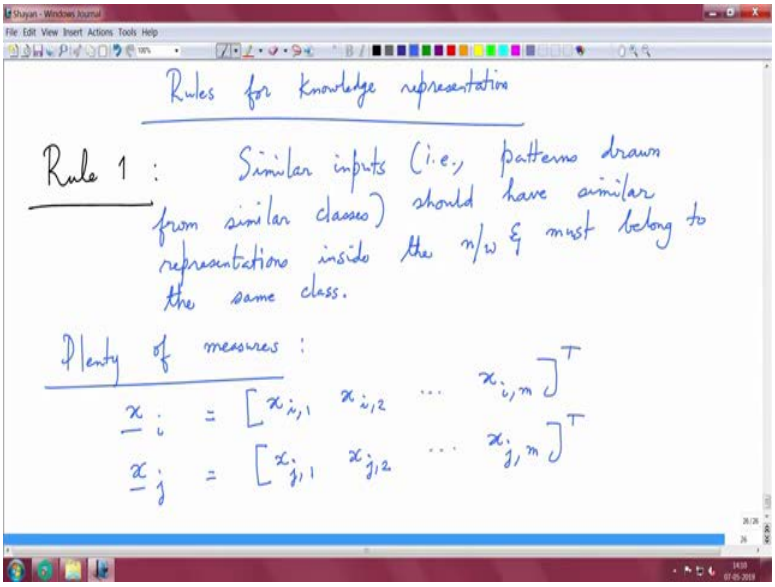
Let's write this down clearly. Knowledge representation refers to stored information or models used by a person or machine to interpret, predict, and appropriately present information to the outside world. When we think about knowledge representation, two key aspects come to mind. First, we need prior information or some known state that the machine can learn and store. Second, we need observations to probe and learn new information.

So, what exactly are we learning? This question naturally arises. We recognize that there is an environment providing certain features and signals from which information can be extracted

and stored within the network. During this process, we are learning the biases and weights of the network. In other words, we are learning the parameters of the network.

You might then ask, how is this learning achieved? Learning is facilitated through a measure or metric that guides the system to align optimally with the data. When we think about any learning process, there is an objective metric that drives the system to learn the information, ensuring it aligns best with the data. This is a crucial statement. We can delve into the specifics of quantification as we introduce the measures for knowledge representation.

(Refer Slide Time: 05:23)



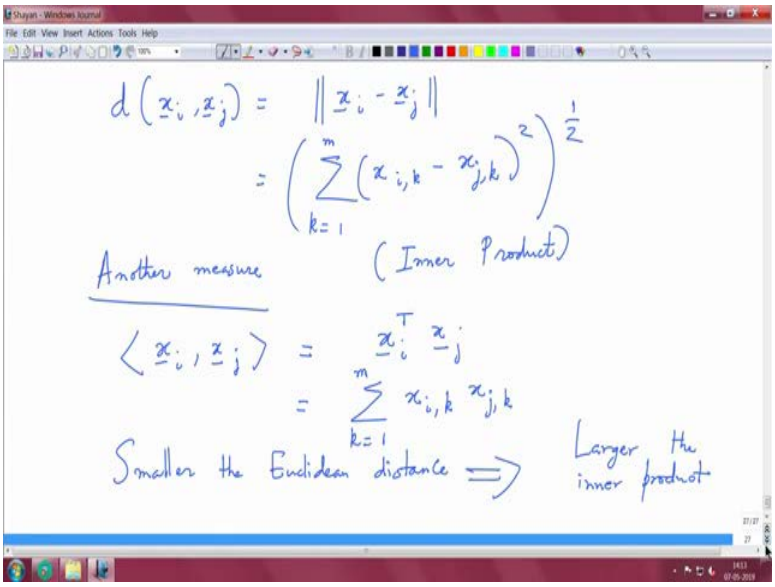
Let's begin with some fundamental rules for knowledge representation. Rule 1 is essential: Similar inputs, meaning patterns drawn from similar classes, should have similar representations within the network. Furthermore, these inputs must be classified into the same group.

Consider an example where we want a learning machine to learn and categorize colors. The machine should group similar colors together. Similarly, if we want the machine to learn about different animals, it should categorize animals accordingly. This necessitates that similar inputs have analogous representations within the network.

There are numerous measures one can consider for this purpose. Let's discuss a couple of these metrics. To start, consider a vector $\mathbf{x_i}$ with m attributes. This vector has coordinates x_{i1}, x_{i2}, …, x_{im}. Now, take another vector $\mathbf{x_j}$, with coordinates x_{j1}, x_{j2}, …, x_{jm}. We assume that both

vectors share the same coordinate dimensions.

(Refer Slide Time: 08:09)



Let's formulate a distance metric between the two vectors $\mathbf{x_i}$ and $\mathbf{x_j}$, specifically using the L_2 norm distance. This is given by the following equation:

You take the difference between the corresponding coordinates x_{ik} and x_{jk}, square these differences, sum them over all the attributes k from 1 to m, and then take the square root of this sum.

Mathematically, this can be expressed as:

$$|\mathbf{x_i} - \mathbf{x_j}|_2 = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}$$

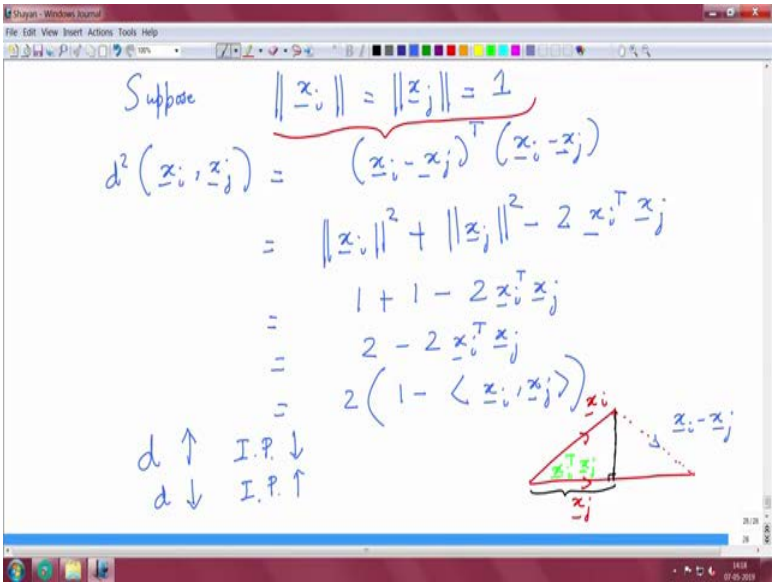
The L_2 norm is a common metric, but there are various norms to consider. In mathematical methods and techniques for signal processing, there's an extensive discussion on different norms and the geometry behind them.

Another important measure is the inner product. The inner product between two vectors $\mathbf{x_i}$ and $\mathbf{x_j}$ is given by the familiar dot product. It is denoted as $\mathbf{x_i^T}\ \mathbf{x_j}$, which represents the pairwise product of the coordinates k for these vectors i and j, summed over k from 1 to m:

$$\mathbf{x_i}^{\mathsf{T}\mathbf{x_j}} = \sum_{k=1}^{m} x_{ik} x_{jk}$$

There is a connection between the inner product metric and the distance metric. Typically, the smaller the Euclidean distance, the larger the inner product. We'll explore this relationship in more detail shortly.

(Refer Slide Time: 10:45)



Let's delve into this concept quickly. Suppose the norms of both vectors, $\mathbf{x_i}$ and $\mathbf{x_j}$, are 1. We consider the square of the distance between $\mathbf{x_i}$ and $\mathbf{x_j}$. This distance can be expressed in terms of the inner product, specifically the induced norm of the difference vector squared, given by $(\boldsymbol{x_i} - \boldsymbol{x_j})^{\mathsf{T}}(\boldsymbol{x_i} - \boldsymbol{x_j})$.

Essentially, we take the deviation of these two vectors, $\mathbf{x_i}$- $\mathbf{x_j}$, which forms our error vector, and then compute the induced self-norm. Expanding this expression, we have:

$$(\mathbf{x_i} - \mathbf{x_j})^{\mathsf{T}}(\mathbf{x_i} - \mathbf{x_j}) = |\mathbf{x_i}|^2 + |\mathbf{x_j}|^2 - 2\mathbf{x_i}^{\mathsf{T}}\mathbf{x_j}$$

Given that the norms of $\mathbf{x_i}$and $\mathbf{x_j}$ are both 1, we simplify this to:

$$|\mathbf{x_i}|^2 + |\mathbf{x_j}|^2 - 2\mathbf{x_i}^{\mathsf{T}}\mathbf{x_j} = 1 + 1 - 2\mathbf{x_i}^{\mathsf{T}\mathbf{x_j}} = 2 - 2\mathbf{x_i}^{\mathsf{T}}\mathbf{x_j}$$

If you're familiar with linear algebra, you'll recognize that $\mathbf{x_i}^{\mathsf{T}}\mathbf{x_j}$ is equal to $\mathbf{x_j}^{\mathsf{T}}\ \mathbf{x_i}$ since it's a

scalar product. Therefore, the expression simplifies to:

$$|\mathbf{x_i} - \mathbf{x_j}|^2 = 2 - 2\mathbf{x_i}^T\mathbf{x_j}$$

Now, we will utilize the property that the individual norms are 1. Thus, we have:

$$|\mathbf{x_i} - \mathbf{x_j}|^2 = 2 - 2\mathbf{x_i}^T\mathbf{x_j}$$

Simplifying this further, we get:

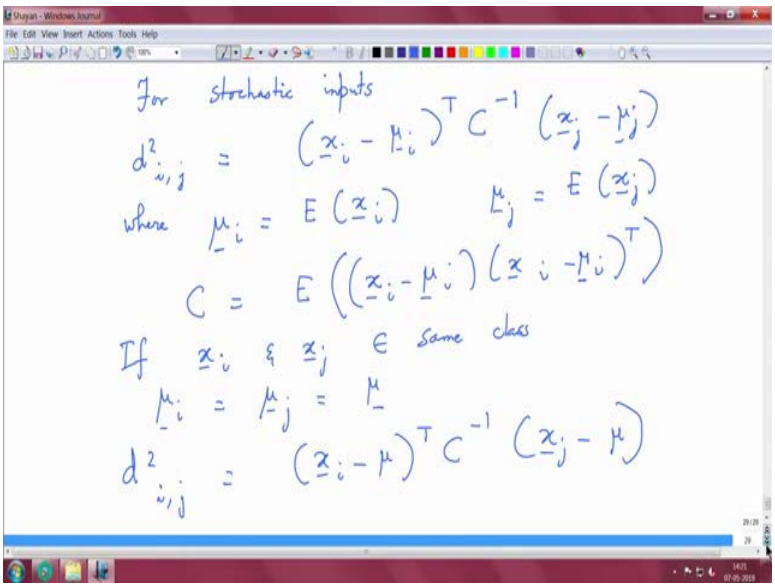
$$|\mathbf{x_i} - \mathbf{x_j}|^2 = 2\left(1 - \mathbf{x_i}^T\mathbf{x_j}\right)$$

This equation clearly indicates that as the distance d increases, the inner product $\mathbf{x_i}^T\ \mathbf{x_j}$ decreases, and conversely, as d decreases, the inner product increases.

To understand this geometrically, consider vector $\mathbf{x_i}$ and another vector $\mathbf{x_j}$. We can complete the triangle, similar to high school geometry, by projecting $\mathbf{x_i}$ onto $\mathbf{x_j}$. This projection represents the length, and the difference vector is given by $\mathbf{x_i}$ - $\mathbf{x_j}$.

Alternatively, we can complete the parallelogram and consider the vector $\mathbf{x_i}$ - $\mathbf{x_j}$. The relationship between the distance and the angle measure becomes apparent: as the distance increases, the inner product decreases, and as the distance decreases, the inner product increases. This highlights the inverse relationship between the distance and the inner product.

(Refer Slide Time: 15:42)



For stochastic inputs, the squared distance between two vectors $\mathbf{x_i}$and $\mathbf{x_j}$ is linked via the covariance matrix. This relationship is given by:

$$(\mathbf{x_i} - \mu_i)^{\top}\mathbf{C}^{-1}(\mathbf{x_j} - \mu_j)$$

Here, $\mathbf{C}$ represents the inverse covariance matrix, and μ_i and μ_j are the means. For simplicity, we can assume μ_i and μ_j are the same, denoted as μ.

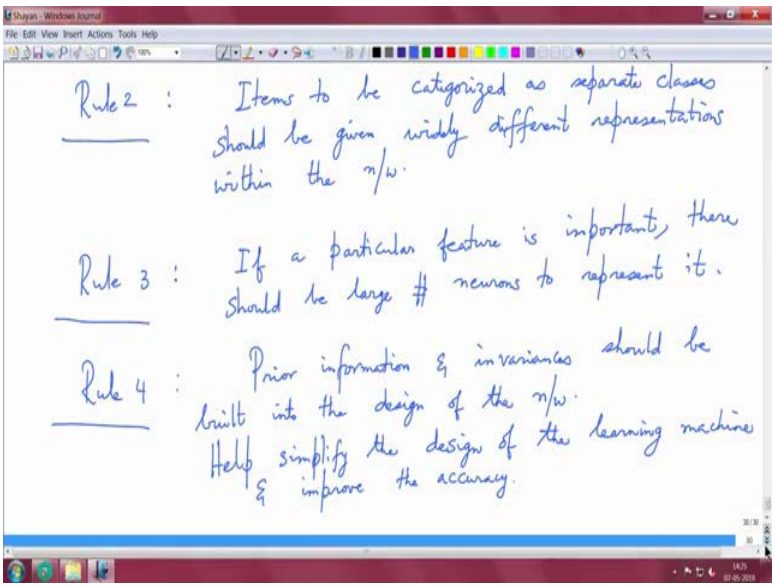
Now, μ_i is the expected value of $\mathbf{x_i}$. Suppose $\mathbf{x_i}$belongs to a certain class with a specific distribution; μ_i is its statistical mean, which is also a vector. Similarly, μ_j is the expected value of $\mathbf{x_j}$, representing the statistical mean over $\mathbf{x_j}$.

The covariance matrix $\mathbf{C}$ is defined as the expectation of $(\mathbf{x_i}\text{- }\mu_i)$ $(\boldsymbol{x_i} - \mu_i)^{\top}$. If the points $\mathbf{x_i}$ and $\mathbf{x_j}$ come from the same class, their means μ_i and μ_j are identical. Therefore, if $\mathbf{x_i}$and $\mathbf{x_j}$ belong to the same class, μ_i equals μ_j because the statistical mean is taken over the same distribution. Consequently, μ_i and μ_j are both μ.

Given this, the distance can be simplified to:

$$(\mathbf{x_i} - \mu)^{\top}\mathbf{C}^{-1}(\mathbf{x_j} - \mu)$$

(Refer Slide Time: 19:06)



One might question the introduction of the covariance matrix. Consider $\mathbf{x_i}$for all values of i.

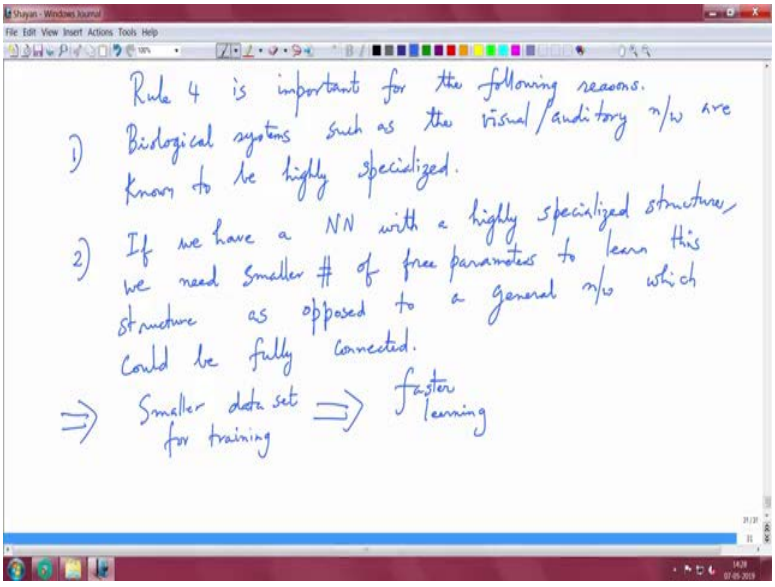
The indices i and j are subsumed within this definition of the matrix. Without this assumption, we would need to look at the joint expectation of the random variables $\mathbf{x_i}$ and $\mathbf{x_j}$ if they belonged to different classes.

Let's start with Rule 3. Rule 3 states that items to be categorized into separate classes should be given significantly different representations within the network. However, I believe this should be Rule 2, considering the previous rule was Rule 1. So, Rule 2 asserts that items to be categorized into separate classes must have distinctly different representations to avoid any ambiguity between the two classes.

Next, we have Rule 3. This rule stipulates that if a particular feature is important, there should be a large number of neurons dedicated to representing it. This is a heuristic rule, implying that an important feature requires more neurons to accurately represent it due to its significance.

Finally, we come to Rule 4. This rule emphasizes that prior information and invariances should be integrated into the design of the neural network. This is crucial because embedding prior information and invariances into the network design can simplify the design process and enhance accuracy. This is a pivotal statement, and it's important to incorporate these concepts within a mathematical framework to understand what we mean by prior information and invariances, and how they can be integrated into the network design.

(Refer Slide Time: 22:44)



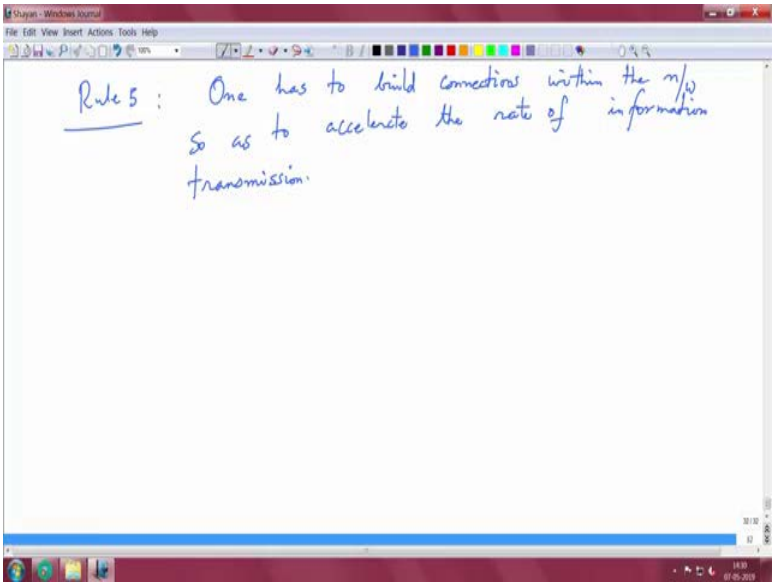
There's a biological rationale behind Rule 4 that I'd like to explain. Rule 4 holds significance

for several reasons. Firstly, biological systems, such as the visual and auditory networks, are highly specialized. Consider the visual system: despite variations like a rotated or elongated image of a cat, we can still recognize it as a cat due to the built-in invariances within our learning networks. This inherent capability is crucial.

Secondly, when a neural network corresponds to a highly specialized structure, it requires fewer parameters to learn effectively compared to a general network that might be fully connected. This means that with a smaller dataset for training, learning is faster.

It's important not to confuse this with the rule mentioned earlier, which suggests allocating more neurons to important features. An important feature can be crucial in both a general and a specialized network, but in a specialized network, the structure is simpler with fewer neurons and less complexity, as it doesn't need to be fully connected to generalize across all types of data. This distinction is an essential empirical consideration when designing networks.

(Refer Slide Time: 26:54)



Let's discuss Rule 5, which emphasizes the importance of building connections within the network to enhance the speed of information transmission. As we learn and process data through the network, optimizing connections can accelerate information flow. By eliminating unnecessary connections, we streamline the transmission of information across directed graphs.

This approach contrasts with maintaining a very general structure where all possible connections exist, requiring more complex reasoning about the nature of these connections.

When we consider deep neural networks and how connections are integrated, this principle becomes particularly relevant.