Neural Networks for Signal Processing-I

Prof. Shayan Srinivasa Garani

Department of Electronic System Engineering

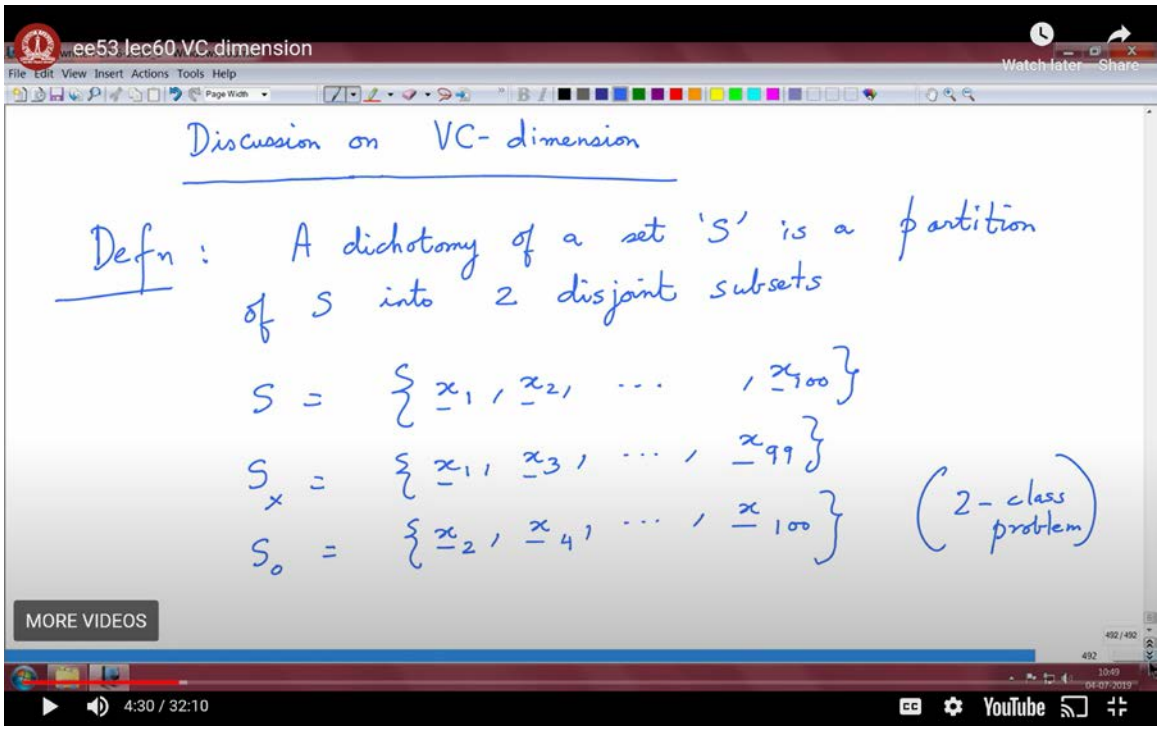
Indian Institute of Science, Bengaluru

Lecture – 60

VC Dimension

In today's lecture, we will explore the concept of VC dimension, focusing on how it relates to the complexity of a classifier in distinguishing between different classes of data. Our motivation arises from understanding the relationship between the number of data points in our dataset and the type of classifier needed to solve the classification problem. Specifically, we are interested in whether there is a connection between the quantity of data points and the complexity of the learning machine required to effectively classify these points.

(Refer Slide Time: 04:30)

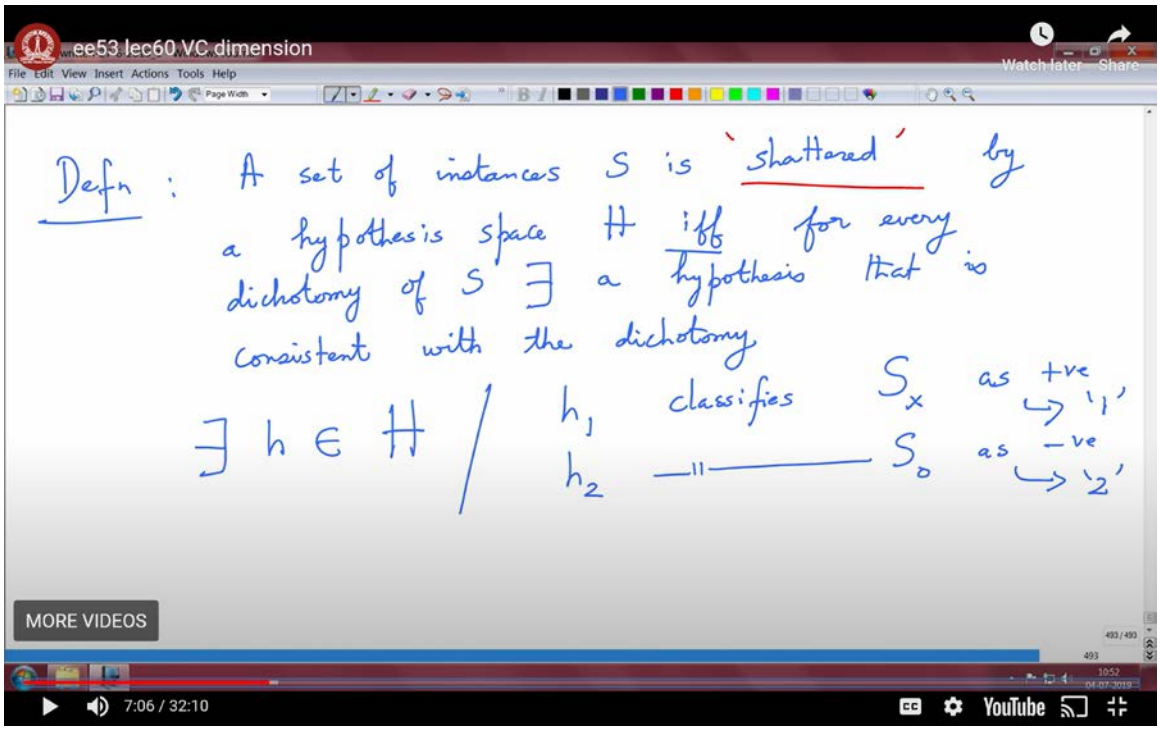


To start, we'll cover some formal definitions that will lay the groundwork for our discussion, followed by illustrative examples to better understand the underlying concepts.

First, let's define dichotomy: a dichotomy of a set S is a partition of S into two disjoint subsets. For example, consider a set S consisting of points $x_1, x_2, \ldots, x_{100}$. We can partition S into two subsets: S_{cross} , which includes the points with odd indices $(x_1, x_3, \ldots, x_{99}), and\ S_{\text{circle}}$, which includes the points with even indices $(x_2, x_4, \ldots, x_{100})$.

If we label the points in S_{cross} as 1 and the points in S_{circle} as 2, this labeling represents a dichotomy of the set S. Thus, x_1 gets the label 1, x_3 gets the label 1, x_2 gets the label 2, and so on. This setup demonstrates a basic two-class problem.

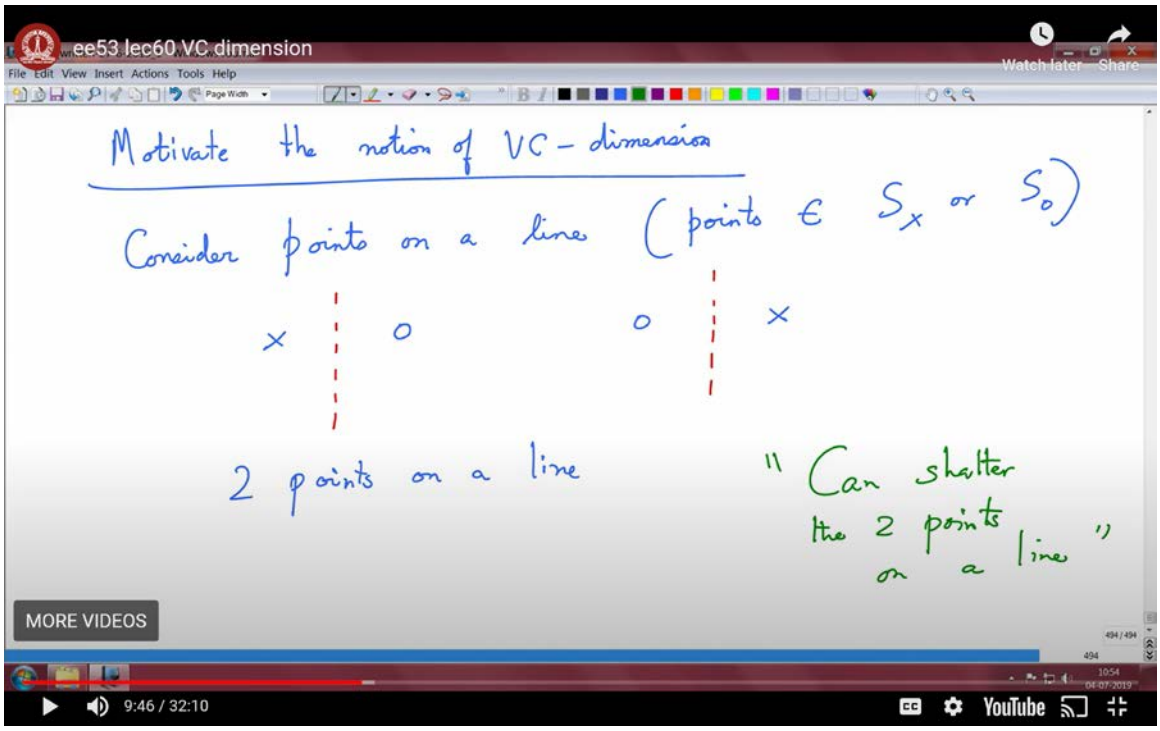
(Refer Slide Time: 07:06)



Next, we introduce the concept of shattering. A set of instances S is said to be shattered by a hypothesis space $\mathcal{H}$ if, for every possible dichotomy of S, there exists a hypothesis in $\mathcal{H}$ that is consistent with that dichotomy. In other words, for every way of partitioning S into two subsets, there is a hypothesis within $\mathcal{H}$ that correctly classifies the points according to that partition. For instance, if a hypothesis H_1 classifies all points in S_{cross} as positive (label

1) and a hypothesis H_2 classifies all points in S_{circle} as negative (label 2), then S is said to be shattered by the hypothesis space $\mathcal{H}$.

(Refer Slide Time: 09:46)



To build our intuition, we'll examine examples involving points on a line and in a two-dimensional plane. This exploration will help us understand how VC dimension relates to classifiers. As you might already know, the VC dimension (Vapnik-Chervonenkis dimension) measures the capacity or complexity of a learning machine.

Let's start with points on a line, where these points belong to either class S_{cross} or S_{circle}. By examining such examples, we will develop a deeper understanding of VC dimension and its implications for classifier complexity.

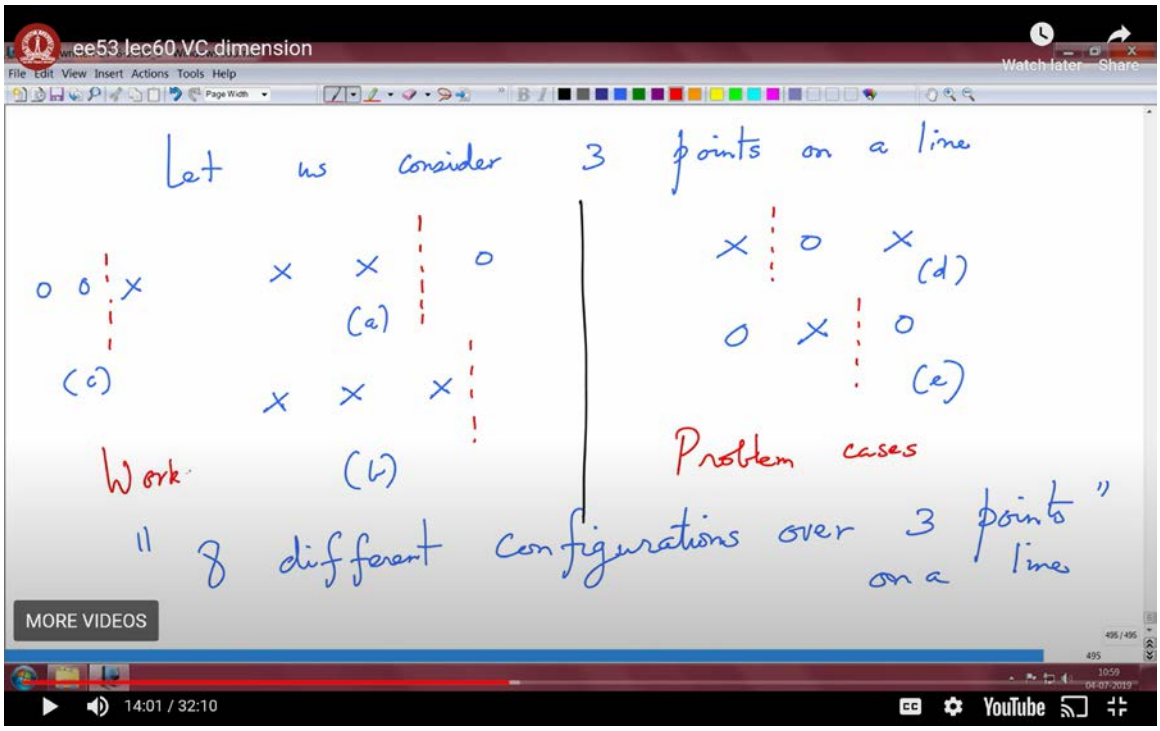
Let's delve into some examples to illustrate the concept of shattering with points on a line and a plane.

First, let’s consider two points, which we'll denote as a cross and a circle. Clearly, these two points can be separated by a line, regardless of its orientation. Whether the cross is on

the left and the circle on the right, or vice versa, we can always draw a line to separate them. Thus, we can easily shatter these two points using a single line.

Now, let's extend this to three points on a line. For instance, we could have configurations like a cross, a cross, and a circle, or a cross, a circle, and a cross, and so on. There are a total of 8 possible configurations for three points.

(Refer Slide Time: 14:01)



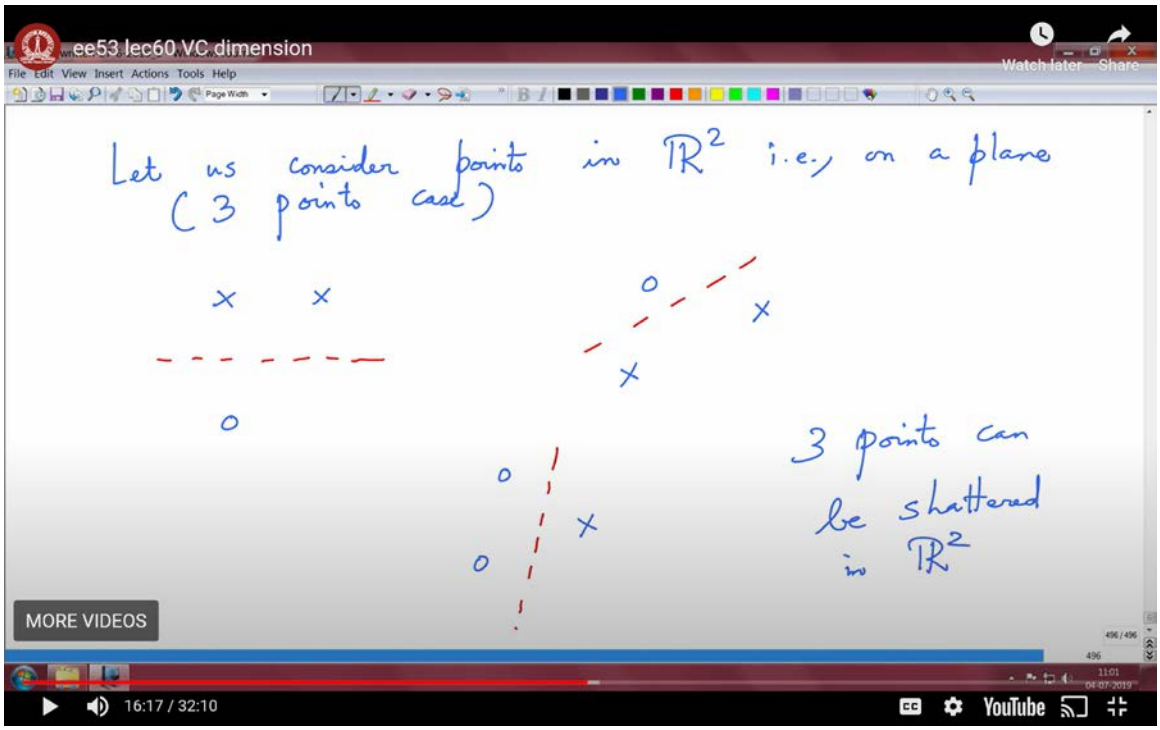
When we analyze these configurations, we observe that certain arrangements are easily separable with a single line. For instance, if all three points are labeled consistently (e.g., all crosses or all circles), a line can easily separate them. However, more complex configurations, such as alternating labels (e.g., cross, circle, cross) or overlapping configurations where points are mixed, present challenges.

In particular, configurations where the points are interspersed, such as a cross, a circle, and a cross or a circle, a cross, and a circle, are problematic. In these cases, a single line cannot separate all points correctly without misclassification. For example, if you attempt to separate a cross on one side of the line and a circle on the other side with points positioned

on both sides of the boundary, you will encounter classification errors. This means that some configurations cannot be shattered perfectly by a single line.

This illustrates a fundamental limitation: while two points are straightforward to shatter, three points introduce complexity. As you can see, the problem cases arise with certain configurations of points, and these are precisely the cases where we encounter classification errors. This observation aligns with the boundary condition discussed in Kovar's theorem, which suggests that not all configurations can be shattered by a single line.

(Refer Slide Time: 16:17)



Now, let's build on this understanding and explore similar concepts with points on a plane.

Now, let's explore the possibilities with points on a plane to deepen our intuition about shattering. Consider two different classes of points: crosses and circles.

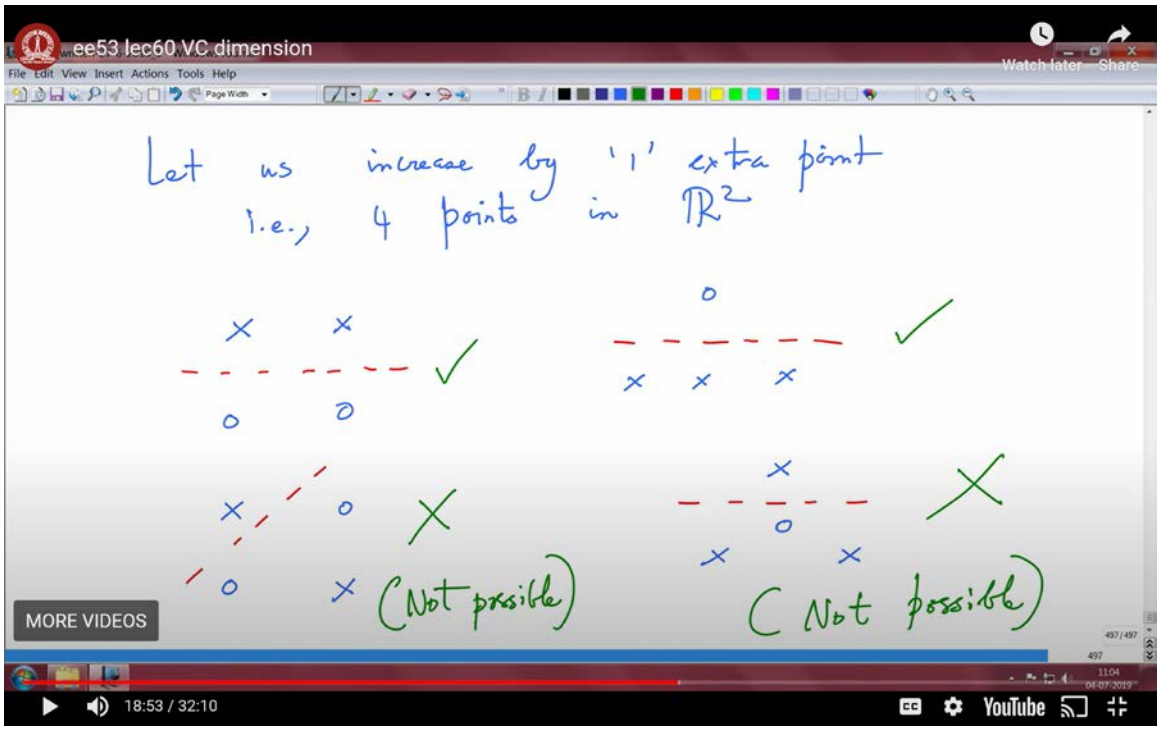
Let's start with configurations involving just a few points. For instance, in Configuration A, where we have a set of points that can be clearly separated by a hyperplane, the

separation is straightforward. This means we can easily shatter these points with a hyperplane.

Moving on to three points, let's look at different configurations. If we have three points arranged in various ways, such as a cross, a circle, and another cross, perhaps forming a rotated triangle, there's no issue with shattering these points using a line. Even if we rearrange them to have two circles and one cross, we can still separate them with a line. Thus, three points in a plane (R^2) can be shattered without any problems.

Now, let's increase the number of points to four and examine how it affects our ability to shatter them. We need to explore various configurations for four points in R^2. For example, if we have a simple configuration where four points are arranged in a way that can be separated by a line, then there's no issue.

(Refer Slide Time: 18:53)

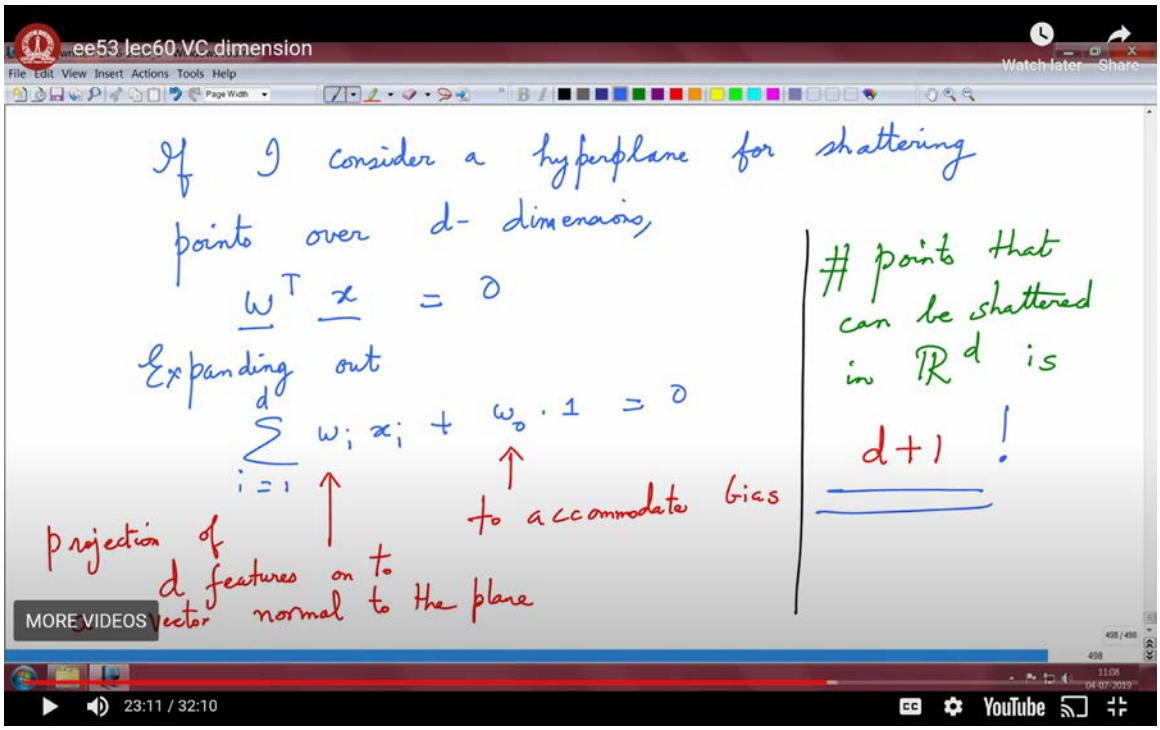


However, consider a more complex scenario such as the XOR problem, which we've encountered in discussions about classifiers, including multilayer perceptrons (MLPs), support vector machines (SVMs), and radial basis functions. In the XOR problem, if we

have a configuration where one class of points is in a diagonal pattern (e.g., crosses in opposing corners) and the other class is also in opposing corners, it is clear that a single line cannot separate these points. The presence of points from one class on both sides of the boundary makes it impossible to shatter them with just one line.

Similarly, if we place four points such that three are arranged in a triangle and the fourth point is at the centroid of this triangle, a single line cannot separate these points properly. Regardless of how the line is positioned, it's impossible to achieve perfect separation.

(Refer Slide Time: 23:11)

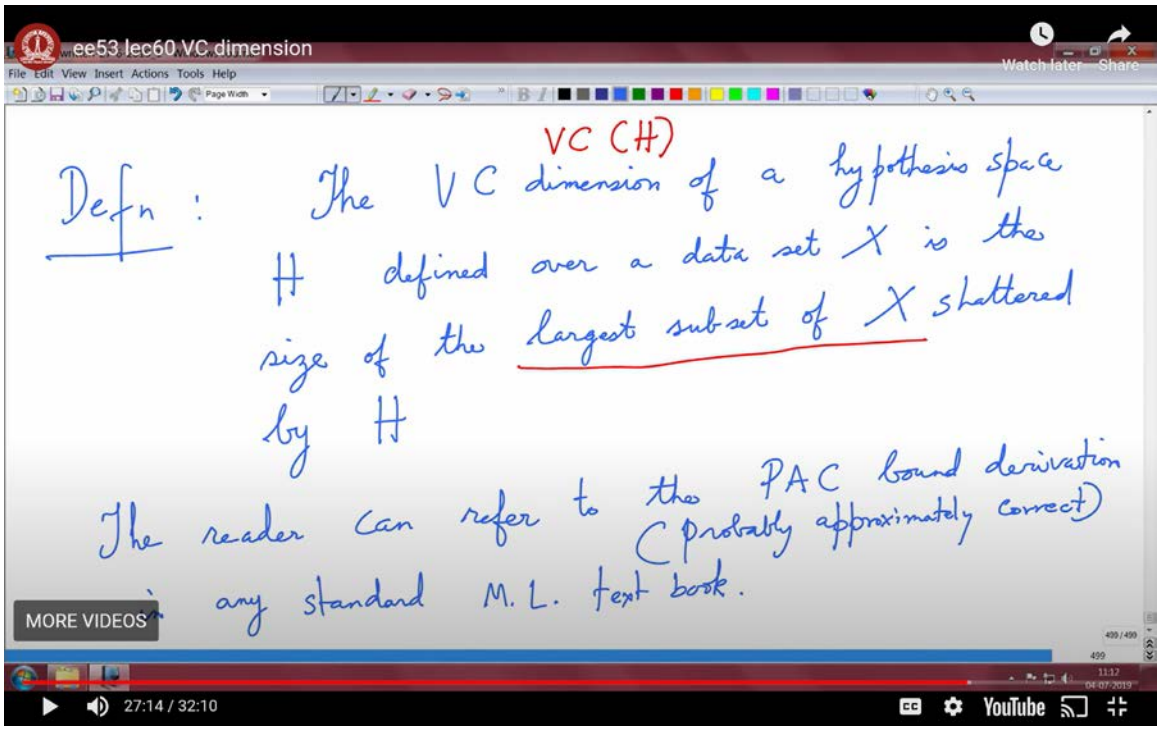


This analysis highlights a connection between the dimension d and the number of points with respect to linear classifiers. In one dimension, two points can be easily separated, but introducing a third point introduces complications. In two dimensions, three points are manageable, but adding a fourth point brings issues. You can extend this analysis to three dimensions and beyond to further explore the relationship between dimensionality and the number of points for linear classifiers.

Let's consider the case of 5 points and analyze various configurations to generalize our understanding. For instance, if you have D features and a bias term, then with D + 1 points, you can shatter any configuration. To illustrate, if you're working in one dimension (a line), two points can be separated easily. In two dimensions (a plane), three points can be managed without issues. Extending this to D dimensions, D + 1 points can be shattered, but beyond that, you might encounter configurations that are impossible to separate.

This observation is crucial for developing intuition about the capabilities of classifiers in various dimensions. However, formal proof of this result is left as an exercise for the reader.

(Refer Slide Time: 27:14)



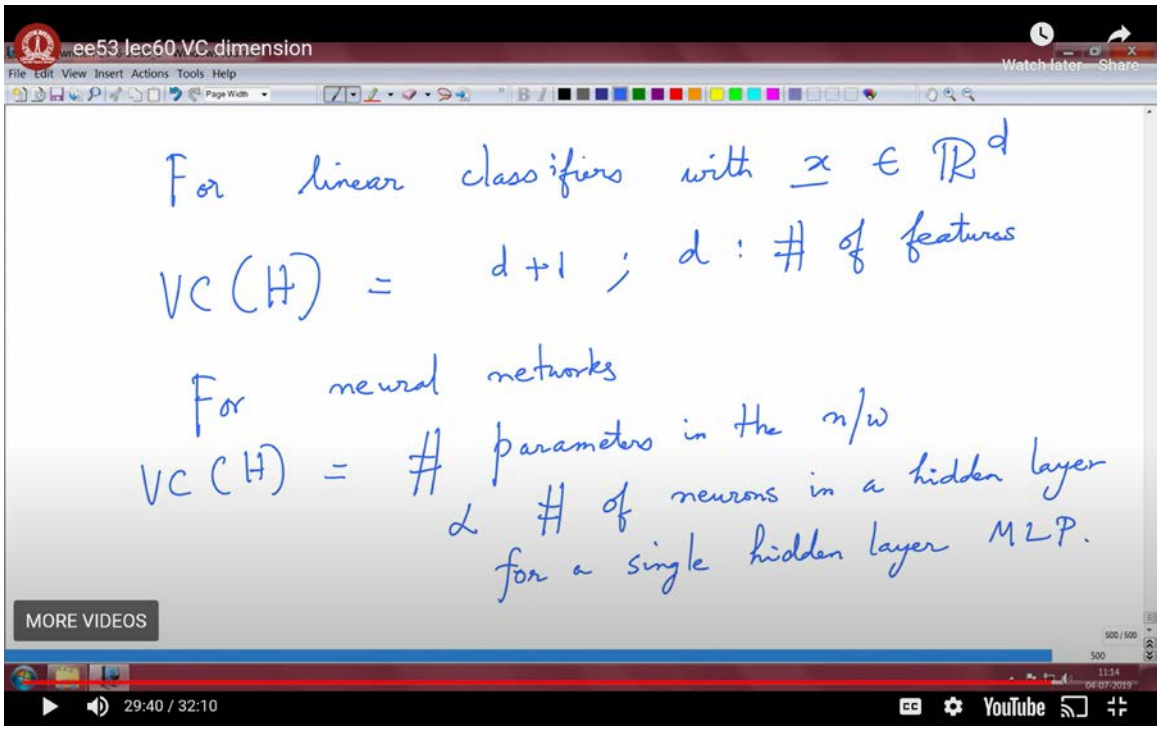
To summarize, when using a hyperplane for shattering points in D dimensions, the equation of the hyperplane can be written as $w^\top x + w_0 = 0$. Here, $w^\top x$ represents the dot product of the weight vector w and the feature vector x, and w_0 is the bias term. This equation can be expanded to $\sum_{i=1}^{D} w_i x_i + w_0 = 0$, which accommodates the bias term and represents the projection of D features onto a vector normal to the hyperplane.

From our analysis, we can conclude that the maximum number of points that can be shattered in $R^D isD + 1$, which is a key result.

With this understanding, we can define the VC dimension. The VC dimension of a hypothesis space H defined over a dataset X is the size of the largest subset of X that can be shattered by H. This is an important concept because it links the complexity of a learning machine to its ability to shatter data points and the number of points in the training set.

In formal terms, the VC dimension of a hypothesis space is the largest number of points that can be shattered by the hypotheses in that space. This dimension provides insight into the capacity of the hypothesis space and its ability to classify data.

(Refer Slide Time: 29:40)



I haven't provided a procedure for calculating the VC dimension, but a key result in machine learning theory is the PAC (Probably Approximately Correct) bound. This bound connects the error in the training set with the VC dimension, constants, and the cardinality of the hypothesis space. For more details on the PAC bound, refer to standard machine learning textbooks or resources on statistical decision theory and statistical learning theory.

Essentially, as you work with higher dimensions, the VC dimension helps you understand how the number of points that can be shattered relates to the complexity of the learning machine.

Let's discuss the types of learning machines and their VC dimensions. For linear classifiers operating in a d-dimensional space, the VC dimension of the hypothesis space is d + 1, where d is the number of features. In contrast, for neural networks, the VC dimension is closely related to the number of parameters in the network, specifically proportional to the number of neurons in a hidden layer.

Consider a single-layer perceptron or a single hidden layer in a multi-layer perceptron (MLP). In neural networks, the complexity or capacity to shatter points is influenced by the number of neurons in the hidden layer. This was demonstrated through the XOR problem, where we examined how different neurons can respond to varying inputs. For example, neurons might respond to inputs being greater than or equal to 1, or to combinations of inputs meeting certain criteria. This exercise highlighted how different configurations of neurons can create various decision regions within the hypothesis space.

The VC dimension for neural networks depends on the number of hidden neurons and can be considered proportional to the number of parameters in the network. Importantly, a single hidden layer in a neural network can approximate any function, demonstrating that multiple layers are not strictly necessary. Multiple layers are used to balance complexity with approximation error. Thus, the VC dimension for a neural network with a single hidden layer is proportional to the number of hidden neurons or parameters in that layer.

In summary, while I haven’t provided a detailed derivation of how to compute the VC dimension, since it would require an advanced understanding of neural network theory, you should appreciate the key takeaway: the VC dimension is a measure of the machine's capacity to shatter points and is directly linked to the number of data points and the complexity of the learning machine.