Neural Networks for Signal Processing-I

Prof. Shayan Srinivasa Garani

Department of Electronic System Engineering

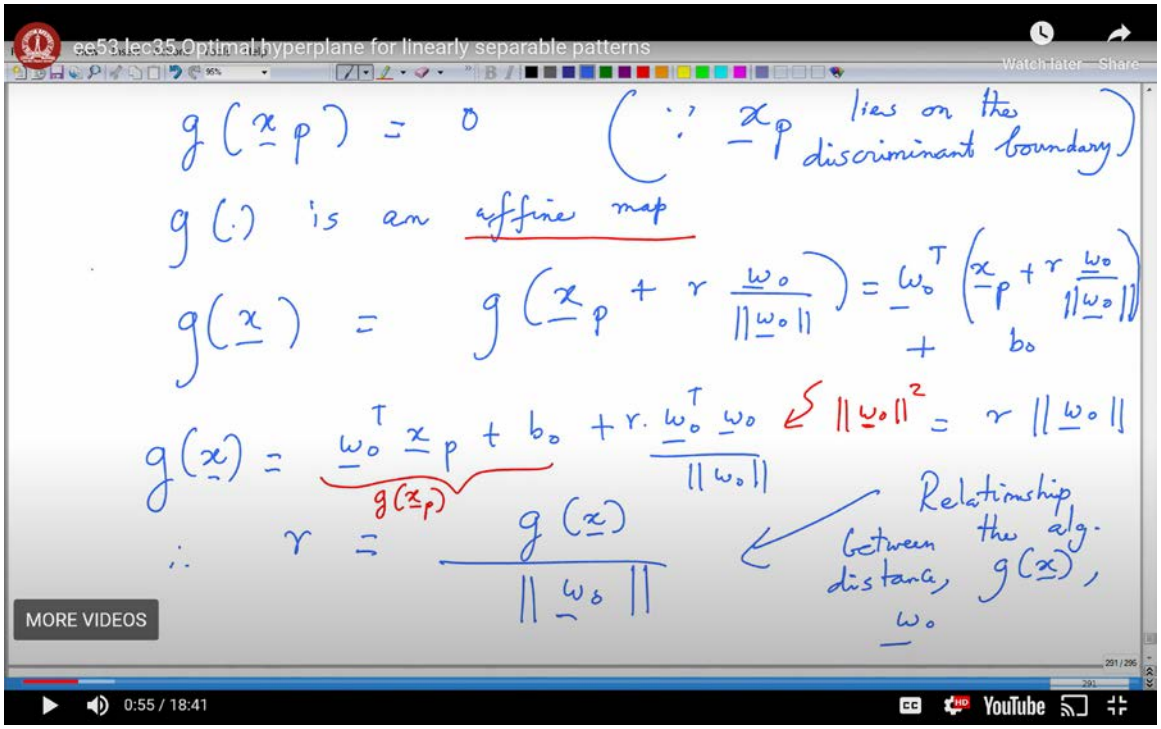
Indian Institute of Science, Bengaluru

Lecture – 35

Optimal Hyperplane for Linearly Separable Patterns

In our previous discussion, we explored the geometry of vectors relative to the normal equation of the hyperplane. Next, we need to delve into the specifics of constructing the optimal hyperplane for this linear machine. Let's proceed with this.

(Refer Slide Time: 00:55)



Recall that g(x) represents the discriminant function. As discussed earlier, a vector x can be decomposed into two components: one that lies in the direction of the hyperplane and one that is normal to it. Specifically, x can be expressed as $x = x_p + r\frac{W_0}{|W_0|}$, where x_p is the projection of x onto the hyperplane and r is the distance from x to the hyperplane.

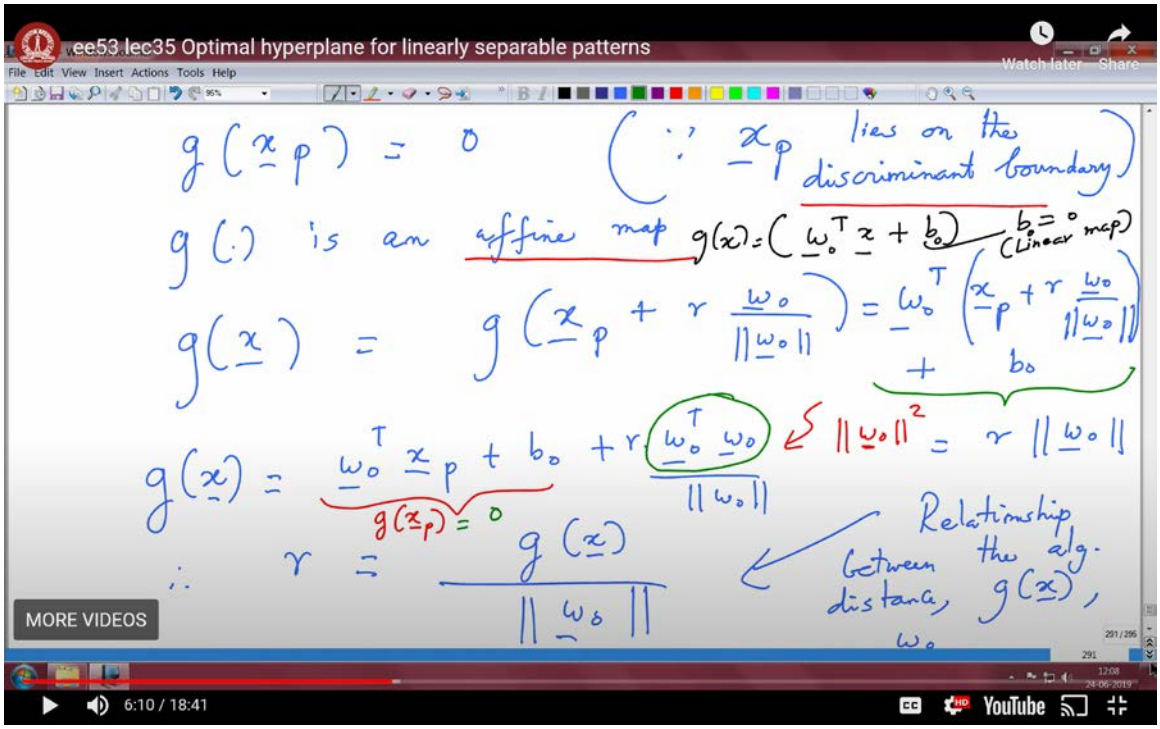
Evaluating the discriminant function g on x_p, which lies on the hyperplane, yields $g(x_p) = 0$. This is crucial because x_p is precisely on the decision boundary, so $g(x_p)$ must be zero.

The discriminant function g(x) is an affine map given by $g(x) = W_0^T x + b$. It's important to note that if b (the bias term) is not zero, the function is affine; if b were zero, it would be a linear map.

Now, let's evaluate g at the point x. Given $x = x_p + r\frac{W_0}{|W_0|}$, we need to compute g(x):

$$g(x) = W_0^T x + b$$

(Refer Slide Time: 06:10)



Substitute x into the equation:

$$g(x) = W_0^T \left(x_p + r\frac{W_0}{|W_0|} \right) + b$$

Expanding this, we get:

$$g(x) = W_0^T x_p + r\frac{W_0^T W_0}{|W_0|} + b$$

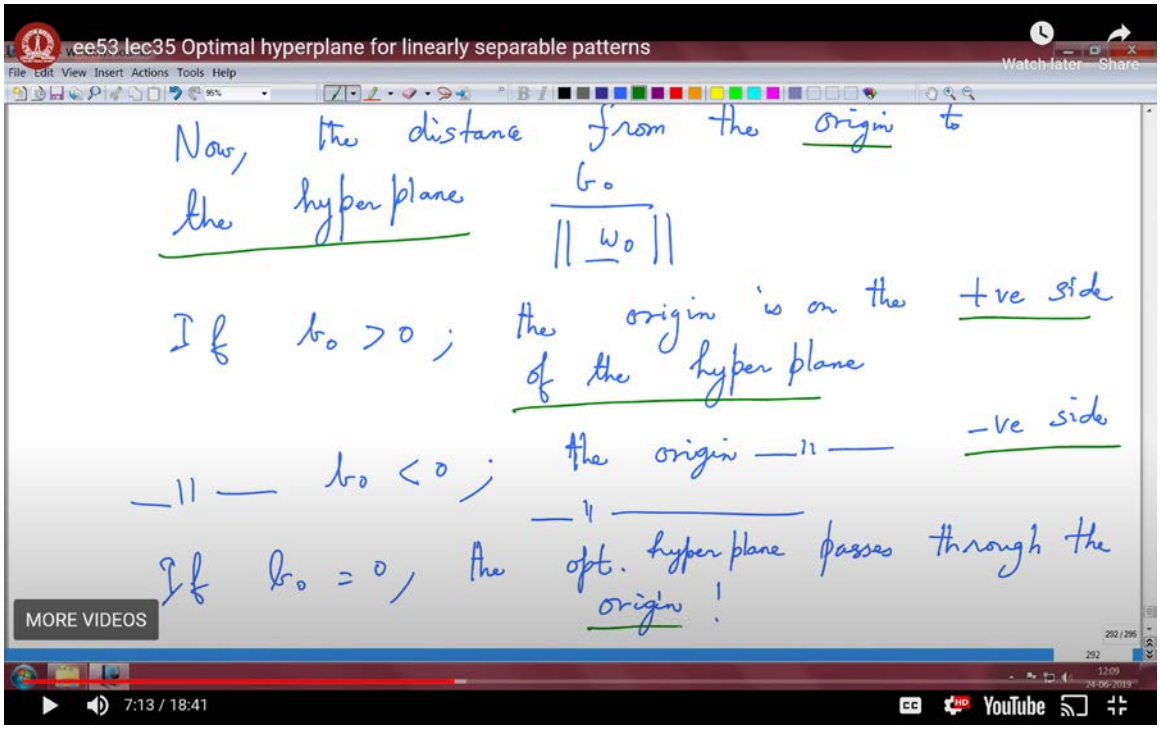
Rewriting it slightly:

$$g(x) = W_0^T x_p + b + r\frac{W_0^T W_0}{|W_0|}$$

Here, $W_0^T x_p + bisg(x_p)$, which is zero because x_p is on the decision boundary. Therefore:

$$g(x) = 0 + r\frac{W_0^T W_0}{|W_0|}$$

(Refer Slide Time: 07:13)



Since $W_0^T W_0 = |W_0|^2$, this simplifies to:

$$g(x) = r|W_0|$$

Thus, g(x) simplifies to r $|W_0|$, where r is the algebraic distance from the point x to the hyperplane.

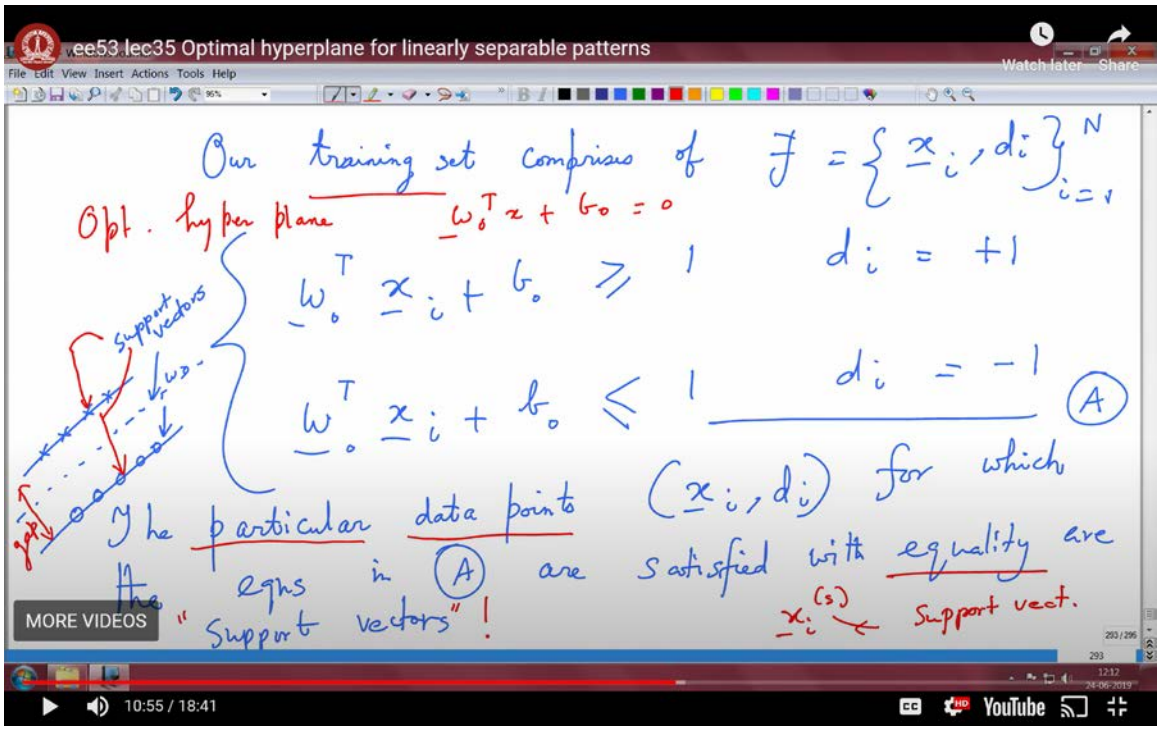
Let's break down the geometry of this problem. We have previously determined that the discriminant function g(x) is given by $g(x) = W_0^T x + b_0$. Since $g(x_p)$ for the point x_p on the hyperplane is zero, this simplifies to:

$$g(x) = r|W_0|$$

where r is the algebraic distance from the point x to the hyperplane. Therefore, we can express the algebraic distance r as:

$$r = \frac{g(x)}{|W_0|}$$

(Refer Slide Time: 10:55)



This provides a direct relationship between the algebraic distance, the discriminant function g(x), and the norm of the weight vector W_0.

Next, let's determine the distance from the origin to the hyperplane. This distance can be computed straightforwardly as:

$$\frac{b_0}{|W_0|}$$

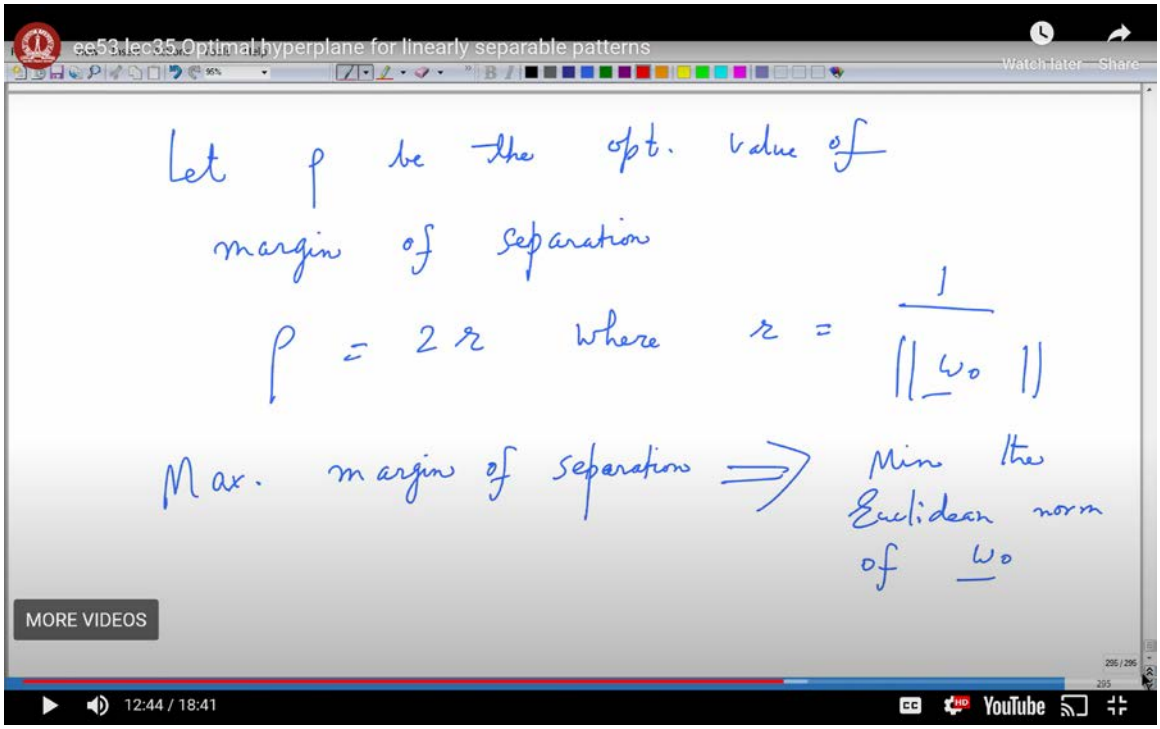
If b_0 is positive, the origin is on the positive side of the hyperplane. Conversely, if b_0 is negative, the origin is on the negative side. If b_0 is zero, the hyperplane passes through the origin, which is an important geometric insight.

Our training set consists of points (x_i, d_i), where d_i is the label, either +1 or -1, and x_i are the feature vectors from i = 1 to n.

Let's examine the geometry of this setup. The equation for our optimal hyperplane is given by:

$$W_0^T x + b_0 = 0$$

(Refer Slide Time: 12:44)



For classification, if:

$$W_0^T x_i + b_0 \geq 1$$

then the label d_i is +1. If:

$$W_0^T x_i + b_0 \leq -1$$

then d_i is -1. If the value equals 1 or -1, the label is not determined by the equation alone, and it is equally likely to be +1 or -1.

Importantly, there are specific data points x_i for which:

$$W_0^T x_i + b_0 = \pm 1$$

These data points are called support vectors. Support vectors are those points that lie exactly on the margin boundaries. In other words, they satisfy the equation with equality and are crucial in defining the optimal hyperplane.

Consider the case where the data is linearly separable. This means there is a non-zero margin or gap between the two classes, which allows for a clear separation.

Visualize the two classes: one set of points (say, x) and another set (say, circles). The support vectors are those points lying exactly on the boundaries of the margin, indicated by bold blue lines. The hyperplane W_0 separates these support vectors.

It's important not to confuse the equation of the bold margin lines with the dotted line representing the hyperplane. The dotted line's parameters are W_0 and b_0, while the bold lines represent planes defined by $W_0^T x + b_0 = \pm 1$. Support vectors are those that satisfy this margin condition with equality and lie on the critical boundary of the margin.

Consider a support vector x_s. For simplicity, we may omit the subscript i in this context.

Let's consider evaluating $g(x_s)$, the discriminant function for a support vector x_s. The function is given by:

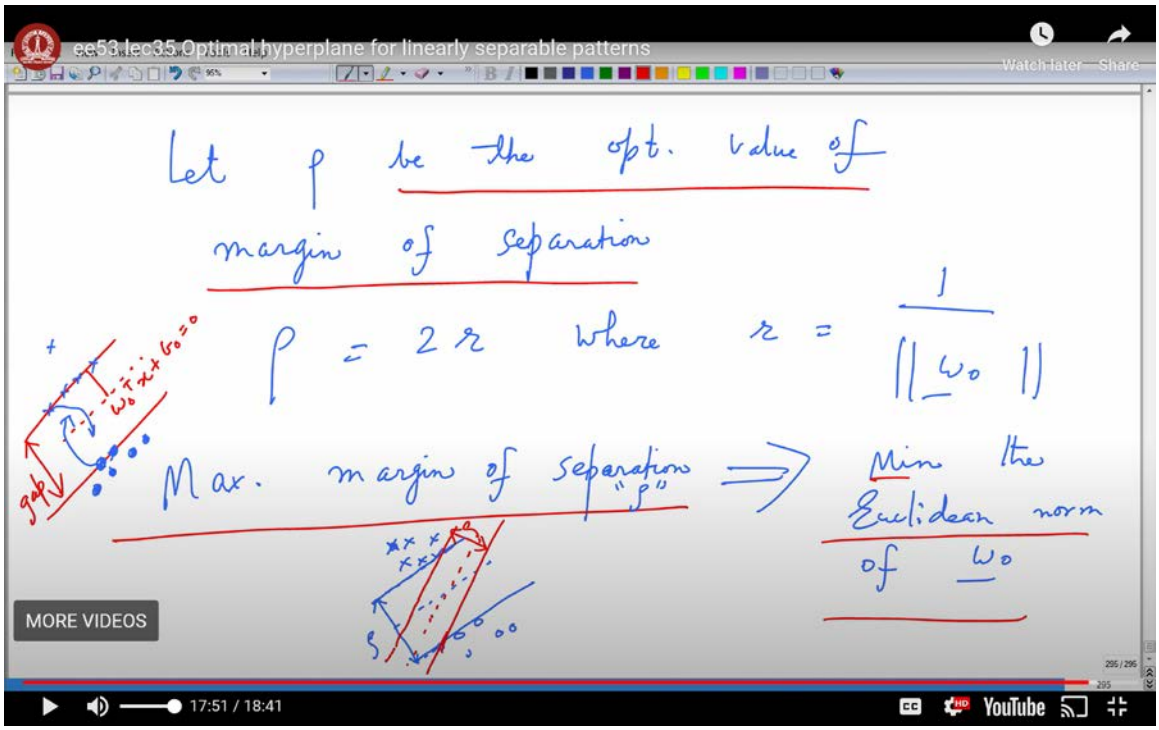
$$g(x_s) = W_0^T x_s + b_0$$

For a support vector with an associated label of -1, $g(x_s)$ will be -1. Conversely, if the label is +1, $g(x_s)$ will be +1. The algebraic distance r from the support vector x_s to the optimal hyperplane is given by:

$$r = \frac{g(x)}{|W_0|}$$

where g(x) is either +1 or -1, depending on the label, and $|W_0|$ is the norm of the weight vector. Therefore, if the label is +1, $r = \frac{1}{|W_0|}$; if the label is -1, $r = -\frac{1}{|W_0|}$. The sign of r reflects which side of the hyperplane the support vector lies on, and the magnitude is scaled by $\frac{1}{|W_0|}$.

(Refer Slide Time: 17:51)



The margin of separation, which is the distance between the two hyperplanes defined by $W_0^T x + b_0 = \pm 1$, is 2r. Since $r = \frac{1}{|W_0|}$, the margin of separation becomes:

$$\text{Margin} = 2 \times \frac{1}{|W_0|}$$

Maximizing this margin is crucial. The larger the margin, the better the separation between the classes. Why is this important? Consider a scenario where there is noise. Imagine you have some points represented by crosses and others by circles. If noise affects your data, some circles might end up on the side where crosses are and vice versa.

By maximizing the margin, you are effectively accounting for potential noise in your training set. A larger margin means that even if noise shifts some data points, the boundary remains robust. Thus, optimizing the margin helps improve the model's generalization performance, as it accounts for variations and ensures that the hyperplane remains effective even when the test set contains points that might be near or on the other side of the margin boundary.

Maximizing the margin of separation, denoted by ρ, is effectively equivalent to minimizing the Euclidean norm of the weight vector W_0. This is because the margin and the Euclidean norm of W_0 have a reciprocal relationship: as one increases, the other decreases.

However, it's crucial to remember that while we are optimizing the margin of separation over all the points in the training set, this process does not directly provide a hyperplane with a unique direction or orientation. By minimizing the Euclidean norm of W_0, which is a scalar quantity, we can have vectors with different orientations but the same norm. For instance, if we have a set of points with crosses and circles, and we fix a margin ρ, we can draw multiple hyperplanes with different orientations that maintain the same margin.

To illustrate, consider the case where we have a margin ρ defined by a dotted hyperplane and solid lines separating two classes. Even if we draw other hyperplanes, represented in red with a different orientation, as long as they achieve the same margin ρ and maintain linear separability, they are valid solutions. This means that while optimizing the maximum margin of separation leads us to minimize the Euclidean norm of W_0, it does not determine the exact orientation of W_0.

This approach is an improvement over the perceptron algorithm, which lacked an objective metric to uniquely define the solution. In the perceptron method, we did not have a clear way to maximize or minimize a specific quantity that could guide us to a unique solution.

This takeaway highlights the advantage of using support vector machines (SVMs) in providing a well-defined optimization metric. In the next steps, we will formulate the optimization problem within the primal-dual framework for SVMs and work towards solving it.