Neural Networks for Signal Processing-I

Prof. Shayan Srinivasa Garani

Department of Electronic System Engineering

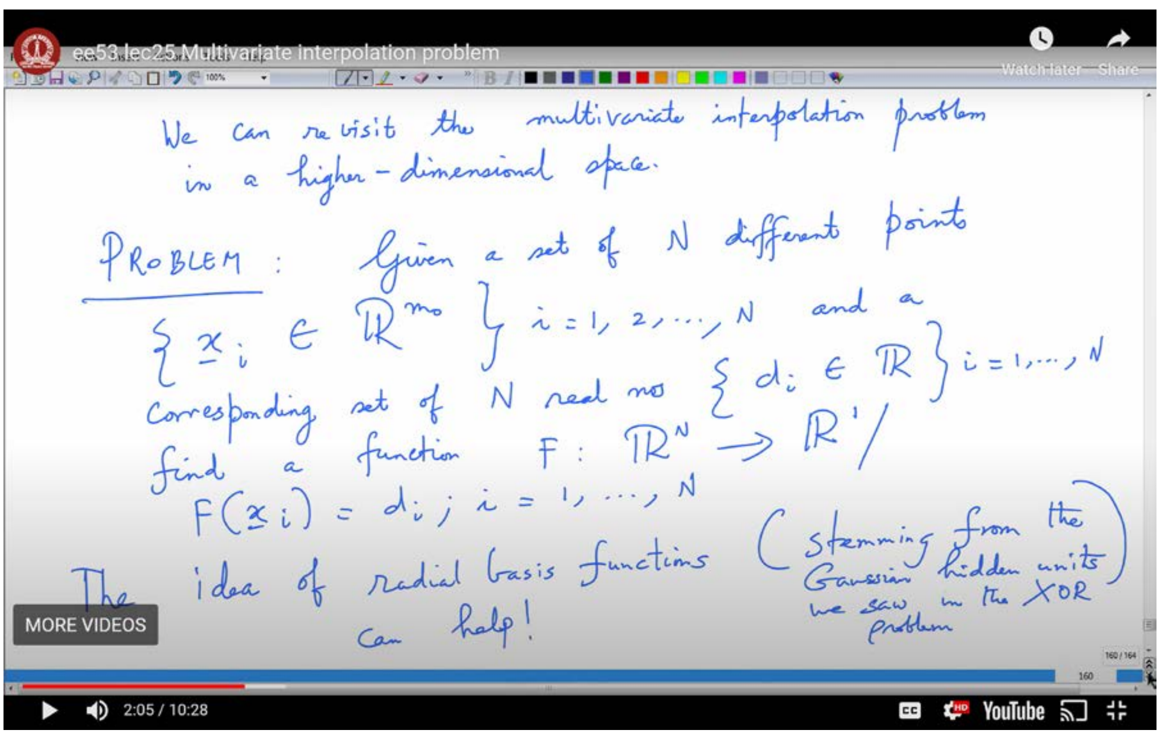
Indian Institute of Science, Bengaluru

Lecture – 25

Multivariate Interpolation Problem

In the last lecture, we revisited the XOR problem that used Gaussian units. Here, we transformed the input space non-linearly and successfully separated the points in the feature space using a perceptron. Now, we're motivated to use these Gaussian units as a basis to solve the interpolation problem in a multidimensional setting, which we will explore here.

(Refer Slide Time: 02:05)

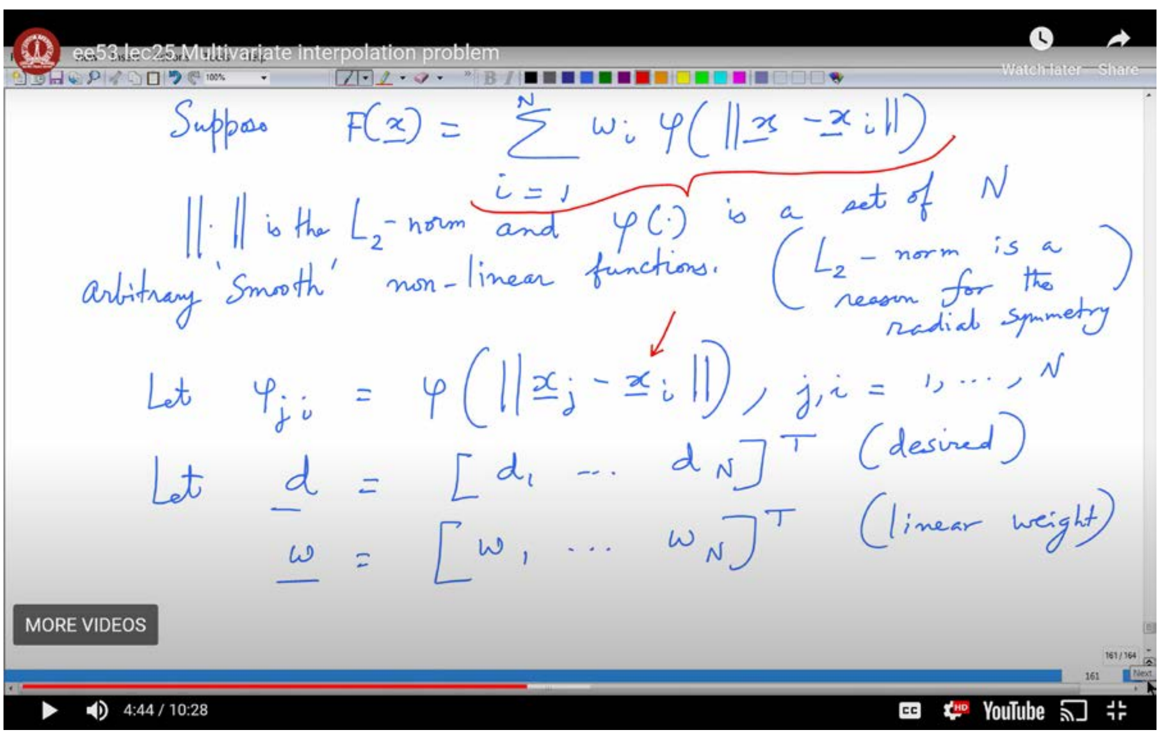


The problem statement is as follows: Given a set of n different points x_i belonging to R^{M_0}, where M_0 is the dimension we've chosen, and n such points i = 1, 2, …, n, along with a corresponding set of n real numbers d_i (which are all scalars), we are interested in finding a function $F: R^n \rightarrow R$ such that $F(x_i) = d_i$ for all these n points.

We would like to use the radial basis functions (RBFs) that we employed earlier while solving the XOR problem. This function F acting on the data point X can be written as a linear combination of these non-linear functions φ acting on the data points. We assume that these non-linear functions are smooth and look at the Euclidean norm between the data point and some chosen centers x_i. This sets up the problem as a linear combination, and our goal is to solve for the weights w_i, where i = 1, …, n.

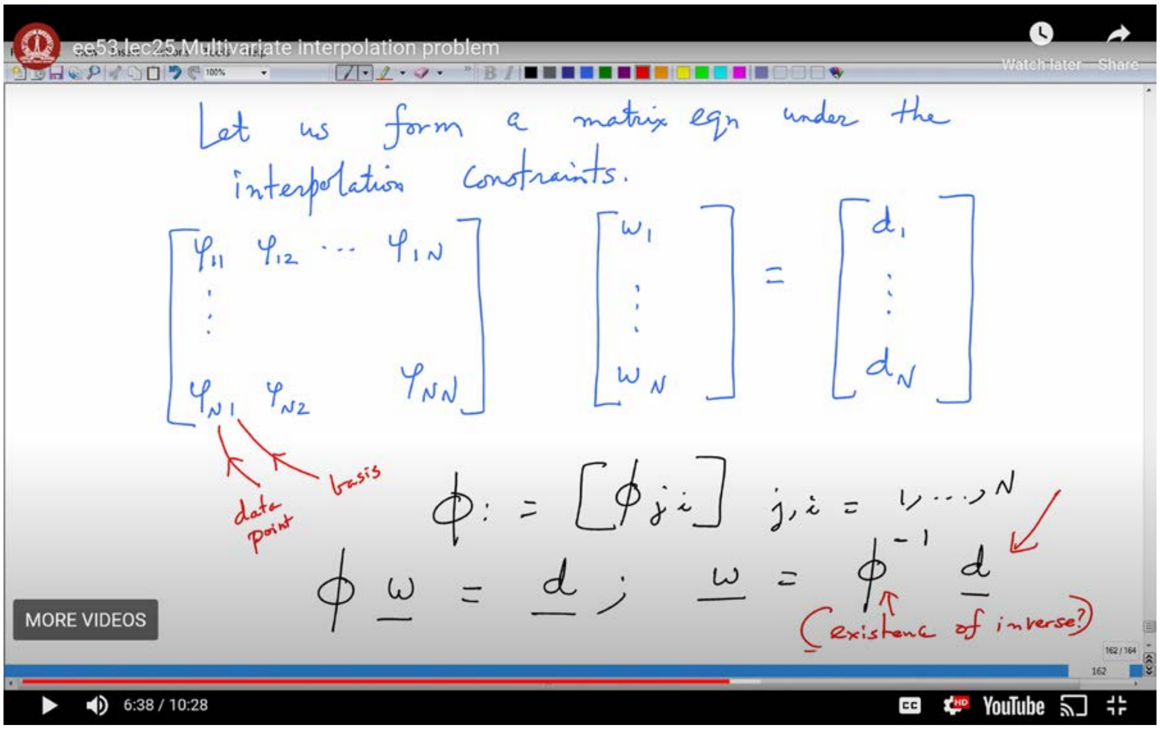
At this stage, you might wonder why we are introducing the L_2 norm within this non-linear function. The L_2 norm has radial symmetry and is isotropic, meaning it does not skew the direction. We want these basis functions to exhibit radial symmetry, which is why we refer to them as radial basis functions (RBFs). The Gaussian function is a prime example of an RBF.

(Refer Slide Time: 04:44)



The function $\varphi_{j,i}$ is given by $\varphi(x_j - x_i)$. First, we fix the centers x_i. Looking at this expression, $\varphi(x - x_i)$, the variable x can vary by selecting the points x_j for j = 1, …, n. Thus, i ranges from 1 to n, and j also ranges from 1 to n, giving us the desired vector d_1, d_2, …, d_n.

(Refer Slide Time: 06:38)

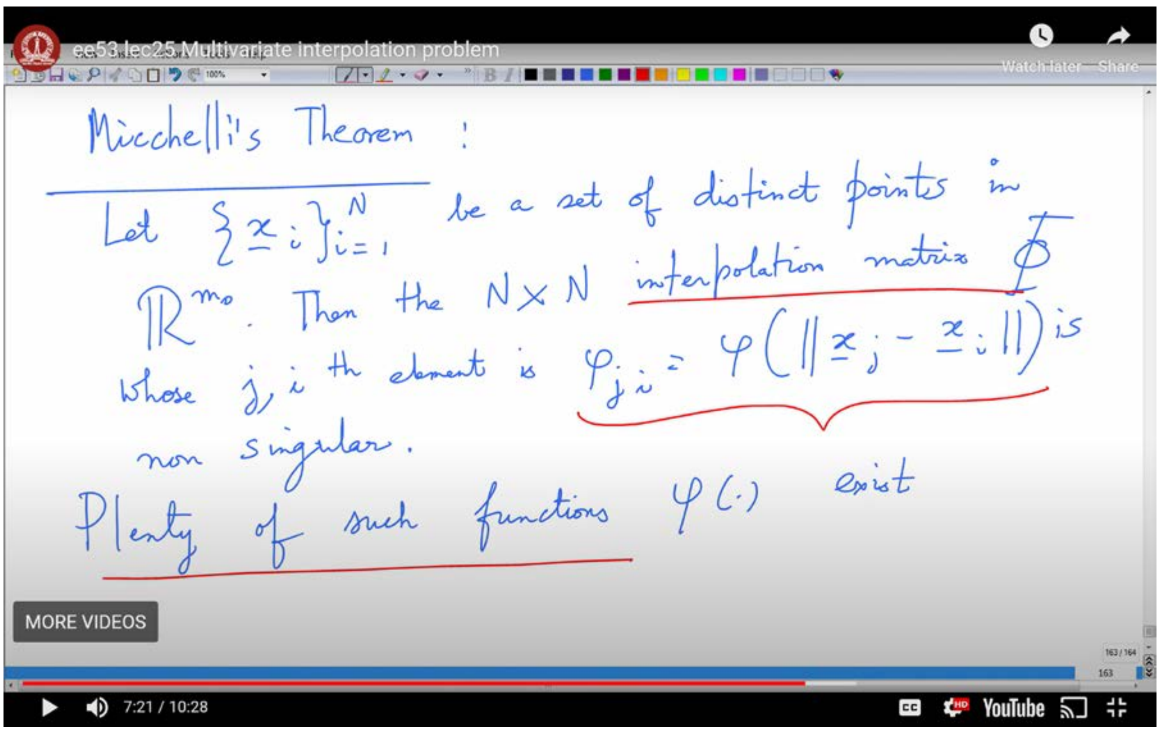


So, these scalars are stacked to form a vector, and we also have the linear weights. Now, if we express this problem in the form of a matrix, it becomes quite straightforward. We have elements $\varphi_{11}, \varphi_{12}, \varphi_{13}, \ldots, \varphi_{1n}$ and weights w_1, w_2, w_3, …, w_n. Examining this matrix, each row corresponds to an equation, such as $d_1 = \varphi_{11}w_1 + \varphi_{12}w_2 + \cdots + \varphi_{1n}w_n$, and similarly for all the points from 1 to N. Thus, we form a matrix φ comprising elements φ_{ji}, where j and i range from 1 to n.

This leads to the matrix equation $\varphi w = d$, where w is the weight vector and d is the desired vector. The straightforward solution to this is $w = \varphi^{-1}d$. You might wonder about the existence of φ^{-1}. This is crucial because φ depends on the choice of the non-linear function φ you started with, which you used to populate the entries in this matrix. For φ to be invertible, the original non-linear function must have certain properties.

Fortunately, there is a theorem by Micheli that states: Let X_i, i = 1 … n, be a set of distinct points in R^{M_0}. Then the $n \times n$ interpolation matrix φ, whose ji-th entry is φ_{ji}, given by $\varphi(|x_j - x_i|)$, is non-singular. Many such functions exist, giving us the freedom to choose the basis functions without being restricted to a particular one.

(Refer Slide Time: 07:21)



Here are a few examples:

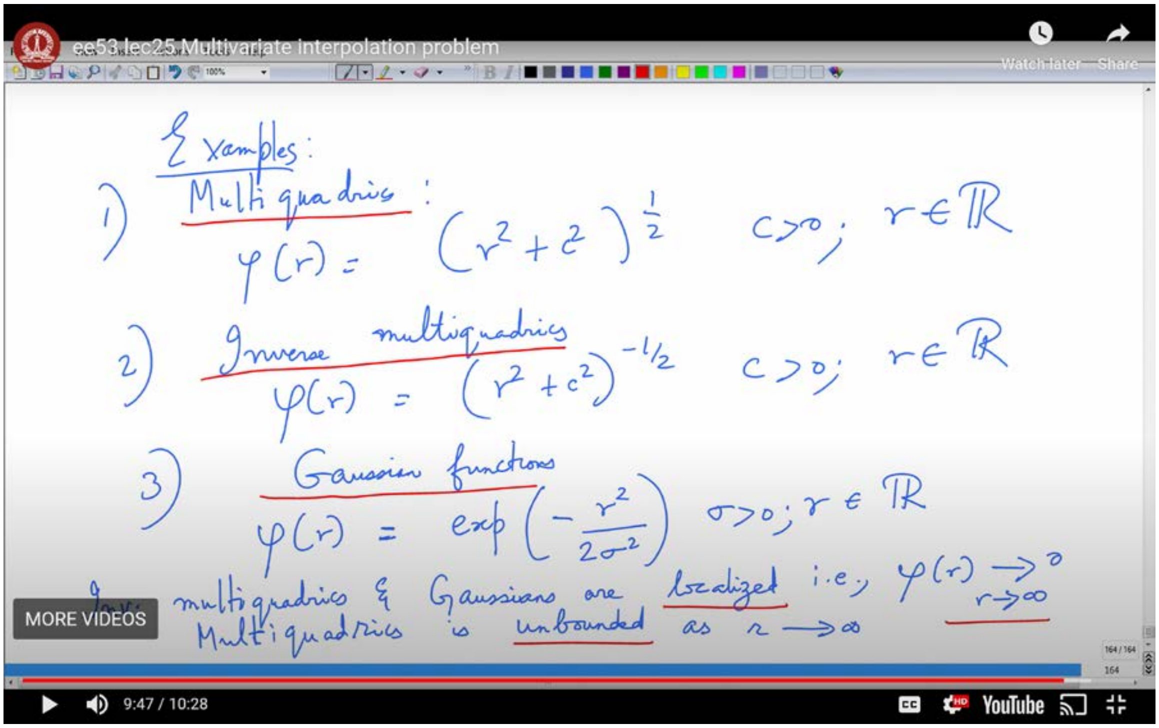
1. Multiquadrics: $\varphi(r) = \sqrt{r^2 + c^2}$, where c > 0 and r is a real number.

2. Inverse Multiquadrics: The reciprocal of the multiquadrics.

3. Gaussian Functions: In the form $\varphi(r) = e^{-(r^2/2\sigma^2)}$, without normalization factors like those in Gaussian distributions.

One thing to note is not to confuse this Gaussian function with the Gaussian distribution. We haven't included a normalization factor here. Typically, in Gaussian distributions, people tend to include constants such as $\frac{1}{\sqrt{2\pi\sigma^2}}$, but here we treat it as a general form from the exponential family.

(Refer Slide Time: 09:47)



These functions, such as inverse multiquadrics and Gaussians, are localized, meaning that φ(r) tends to 0 as r approaches infinity. In other words, if you choose a very large radius, the function becomes localized or shrinks to 0, which has interesting consequences. This behavior is significant in learning, particularly when dealing with the neighborhood radius. For example, in self-organizing maps, we use Gaussian units, and as we fine-tune the map, the radius shrinks to 0. This creates a duality between r = 0 and $r = \infty$, which is crucial because when r goes to infinity, the function becomes localized. Conversely, multiquadrics are unbounded and tend to infinity as r goes to infinity.

Therefore, it's important to put bounds on the multiquadrics function if we use it within our interpolation problem. With this, we complete the interpolation problem, a vital topic as we will revisit it in the context of the RBF (Radial Basis Function) network. At the output layer of the RBF network, we will need to solve this particular problem to determine the synaptic weights connecting the hidden units to the output layer. This concludes our discussion for now, and we will revisit this problem later in the course.