Neural Networks for Signal Processing-I

Prof. Shayan Srinivasa Garani

Department of Electronic System Engineering

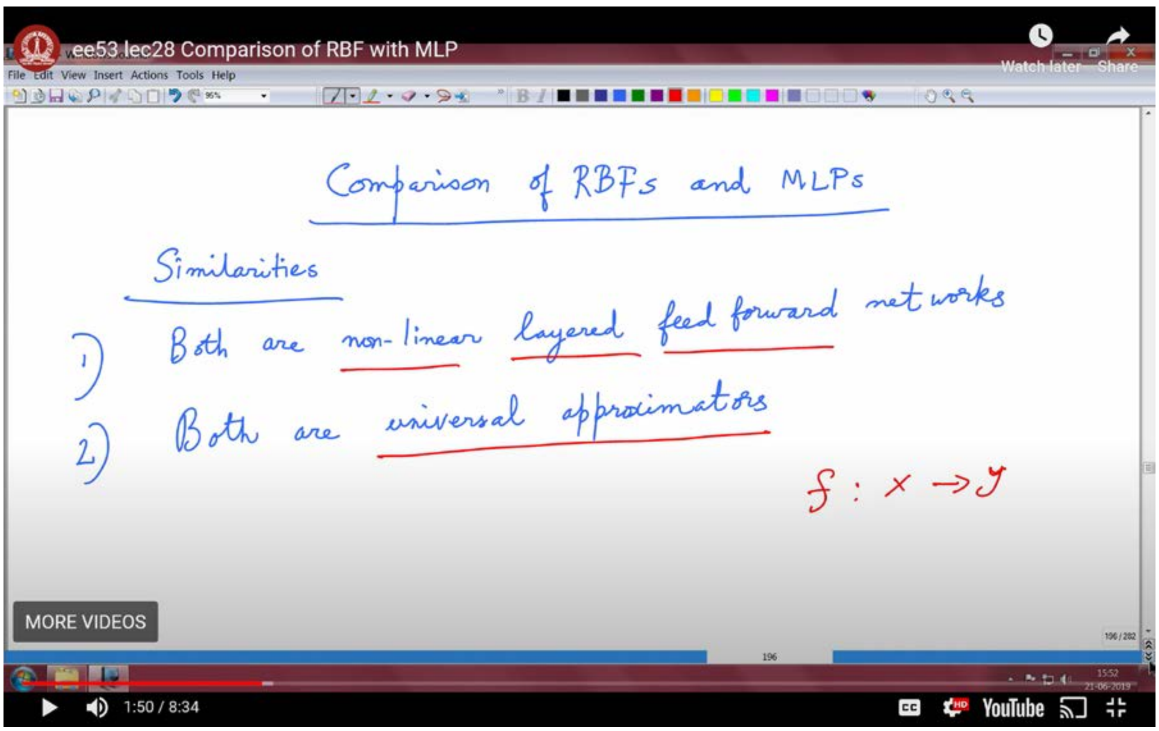
Indian Institute of Science, Bengaluru

Lecture – 28

Comparison of RBF with MLP

Having studied both Multilayer Perceptrons (MLPs) and Radial Basis Function (RBF) networks, let's explore the similarities and differences between these two approaches.

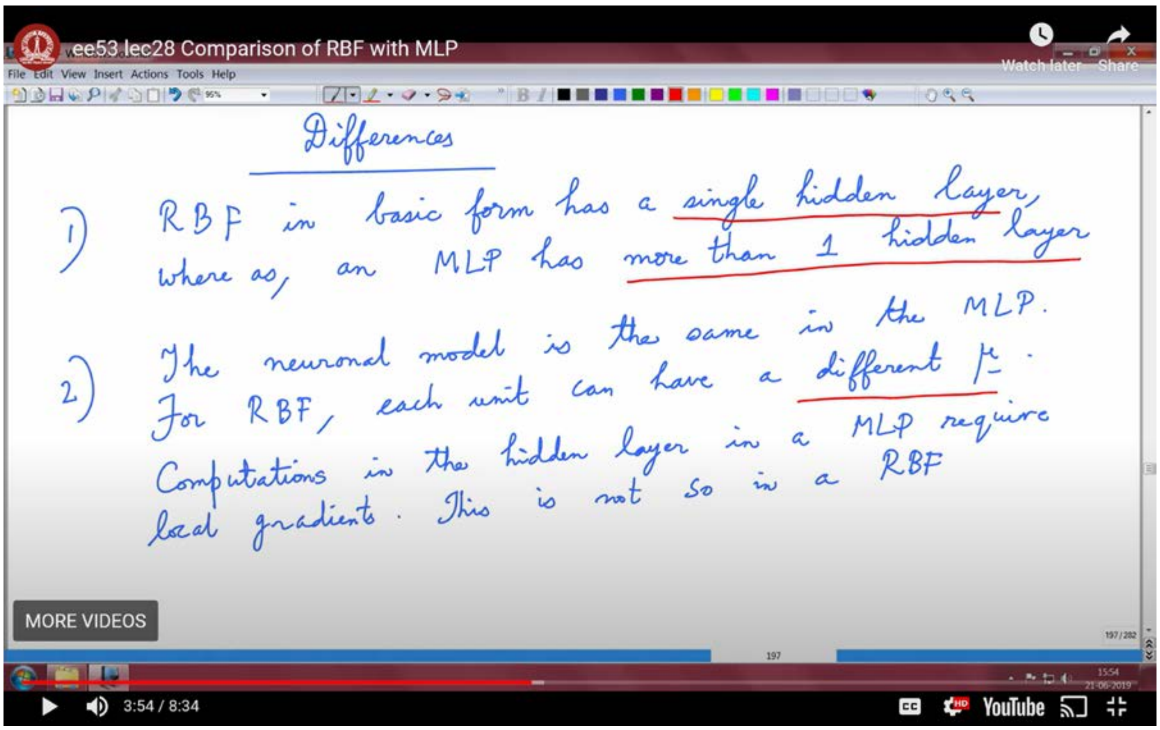
(Refer Slide Time: 01:50)



Similarities:

Both MLPs and RBF networks are non-linear, feedforward networks. In an RBF network, you typically have just one hidden layer, while in an MLP, multiple hidden layers can be present. However, both architectures maintain a layered structure. They are called feedforward networks because the information flows unidirectionally from the input layer to the output layer, with no feedback loops involved. The non-linearity in these networks arises due to the activation functions; MLPs use non-linear activation functions across their layers, while RBF networks incorporate non-linear elements within the radial basis functions themselves. Importantly, both MLPs and RBF networks are considered universal approximators, meaning that they can approximate any continuous function f mapping inputs x to outputs y.

(Refer Slide Time: 03:54)



Differences:

Let's now delve into the key differences between these networks:

1. Network Structure:

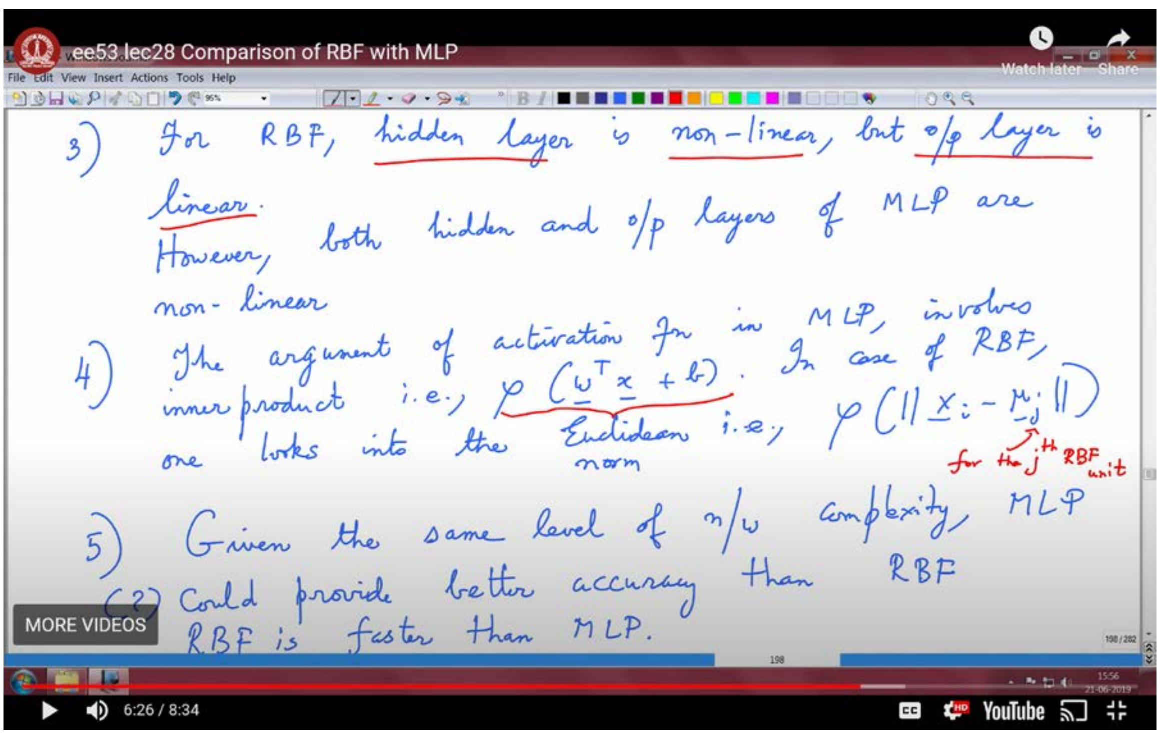
The most fundamental distinction lies in their structural design. The RBF network, in its basic form, consists of a single hidden layer. This simplicity is central to how the RBF is constructed. In contrast, an MLP generally has multiple hidden layers. While a single hidden layer is theoretically sufficient for universal approximation in both MLPs and

RBFs, the use of multiple layers in MLPs becomes crucial when we need to derive context or hierarchical features, as is common in Convolutional Neural Networks (CNNs) and other deep learning architectures. Thus, this introduces a significant morphological difference between the two networks: RBF networks typically operate with a single hidden layer, whereas MLPs utilize multiple layers to capture complex patterns.

2. Neuronal Model:

Another important difference lies in the neuronal models employed. In MLPs, every neuron across all layers typically shares the same activation function. However, in an RBF network, each radial basis function unit can have a different parameter, μ, which depends on the cluster mean calculated during training. While μ can vary, it's common practice to assume a uniform σ^2 (the variance) across all units for simplicity and practical purposes.

(Refer Slide Time: 06:26)



3. Computation in the Hidden Layer:

The way computations are performed in the hidden layers also differs significantly. In

MLPs, hidden layer computations require the calculation of local gradients, which are used to propagate errors backward through the network during training. This backpropagation process adjusts the synaptic weights based on these local gradients. In contrast, RBF networks do not rely on local gradients within the hidden layer. Instead, the output is a linear combination of the radial basis functions, and optimization is performed directly on the output weights, simplifying the training process.

These differences highlight the distinct approaches of MLPs and RBF networks in terms of structure, neuron modeling, and computational methods within the hidden layers. Each technique offers unique advantages depending on the specific application and requirements.

In an RBF network, the hidden layer is non-linear, while the output layer is linear—a significant structural distinction. This is in contrast to the multilayer perceptron (MLP), where both the hidden and output layers are non-linear. Because of this, one could argue that MLPs might possess slightly better generalization capabilities, as the output layer's non-linearity contributes to the overall complexity of the model.

In an MLP, the activation function's argument involves calculating the inner product between the input vector and the weights. This operation is represented by $\phi(W^T x + \text{bias})$, where the inner product $W^T x$ is implicitly computed. However, in an RBF network, the process differs. Instead of computing an inner product, the RBF network calculates the Euclidean norm between the input vector and the parameter μ_j of the j-th RBF unit.

Another critical difference, though more nuanced, is the potential accuracy of these networks given the same level of complexity. Heuristically, one might argue that the MLP could offer better accuracy than the RBF network. This is due to the non-linear processing elements in the MLP's output layer, which may enhance its generalization abilities. However, this is not a definitive rule and depends heavily on the data and specific conditions.

It's essential to note that RBF networks generally offer superior computational speed compared to MLPs. This speed advantage arises from the ability to compute inverses using recursive least squares, which accelerates the process. In contrast, MLPs, especially those with multiple layers, require sequential error propagation through each layer, from the input to the synaptic weights of the first hidden layer, leading to a longer processing time.

Due to these differences, RBF networks are typically faster than MLPs, but the optimal choice of algorithm ultimately depends on the dataset and the network complexity you can tolerate. This choice can be further refined through experimentation to align with the specific requirements of your solution.

Moreover, when implementing these algorithms in hardware, the MLP may introduce latency because computations must sequentially pass through each layer before reaching the synaptic weights connected to the input and the first hidden layer. In contrast, RBF networks can be implemented with more efficient hardware designs, potentially reducing latency.

With these considerations in mind, we will conclude this discussion and continue exploring regression estimation problems related to radial basis functions.