Radial Basis Function Networks

Sanjay Singh*†

*Department of Information and Communication Technology
Manipal Institute of Technology, MAHE,Manipal-576104, INDIA
sanjay.singh@manipal.edu

†Centre for Artificial and Machine Intelligence (CAMI)
MAHE, Manipal-576104, INDIA

March 2, 2019

Some definitions

- **Stochastic approximation** methods are a family of iterative optimization algorithms that attempts to find zeros or extrema of function which can not be computed directly, but only estimated via noisy observation
- **Curve fitting** is the process of constructing a curve or mathematical function that has the best fit to a series of data points, possibly subject to constraints. Curve fitting can involve either interpolation where an exact fit of the data is required or smoothing, in which a smooth function is constructed that approximately fits the data.

- Supervised learning in MLP
  - Recursive technique of stochastic approximation, e.g., backpropagation
  - Design of neural networks as a curve-fitting (approximation) problem in high dimensional space, e.g., RBF
- Curve-fitting
  - Finding a surface in a multidimensional space that provides a best fit to the training data
  - "Best fit" measured in some statistical sense
  - RBF is an example: hidden neurons forming an arbitrary basis for the input patterns when they are expanded into the hidden space. These basis are called radial basis function

Radial Function

Radial function is a function defined on a Euclidean space $\mathbb{R}^n$, whose value at each point depends only on the distance between that point and the origin.

Radial Basis Function

A **radial basis function (RBF)**[a] is a real-valued function whose value depends only on the distance from the origin, so that $\phi(x) = \phi(||x||)$ or alternatively on the distance from some point c, called a center, so that $\phi(x, c) = \phi(||x - c||)$. Any function that satisfy the property $\phi(x) = \phi(||x||)$ is called a radial function.

[a] Wikipedia. *Radial basis function — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Radial_basis_function&oldid=741801190. [Online; accessed 1-October-2016]. 2016.

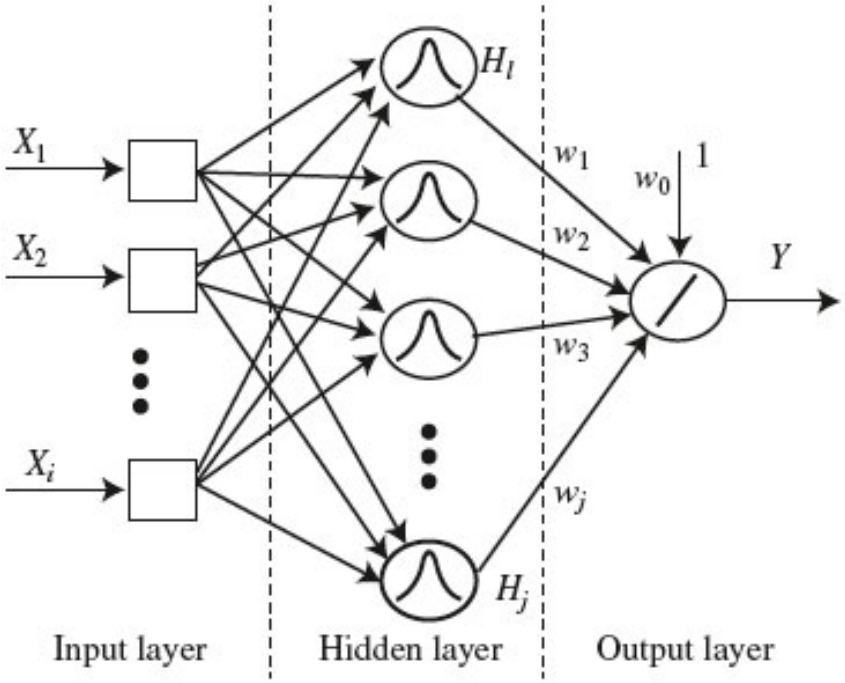


Figure 1: Radial Basis Function Network

Cover's Theorem on Separability of Patterns

Theorem

A complex pattern-classification problem cast in a high dimensional space nonlinearly is more likely to be linearly separable than in a low dimensional space[a].

[a] T. M. Cover. "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition". In: *IEEE Transactions on Electronic Computers* EC-14.3 (1965), pp. 326–334. ISSN: 0367-7508. DOI: 10.1109/PGEC.1965.264137.

Basic idea-non-linearity maps points in the input space to a hidden space that has a higher dimension than the input space. Once proper mapping is done, simple algorithms can be used to find the separating hyperplane.

φ-Separability of Patterns

- N input patterns $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in m_0 dimensional space
- The inputs belong to two classes $\mathcal{C}_1$ and $\mathcal{C}_2$: they form a dichotomy
- The dichotomy is separable wrt a family of surfaces if a surface exists in the family that separates the points in class $\mathcal{C}_1$ from $\mathcal{C}_2$
- For each $\mathbf{x} \in X$, define an m_1-vector $\{\varphi_i(\mathbf{x}) | i = 1, 2, \ldots, m_1\}$ by

$$\varphi(\mathbf{x}) = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \ldots, \varphi_{m_1}(\mathbf{x})]^T$$

  that maps inputs in m_0 dim space to the hidden space of m_1 dim space called hidden space, and the space spanned by these functions is called the hidden space or feature space
- A dichotomy is φ-separable if there exists an m_1 dim vector $\mathbf{w}$ such that

$$\mathbf{w}^T\varphi(\mathbf{x}) > 0, \quad \mathbf{x} \in \mathcal{C}_1,$$

$$\mathbf{w}^T\varphi(\mathbf{x}) < 0, \quad \mathbf{x} \in \mathcal{C}_2$$

  with separating hyperplane

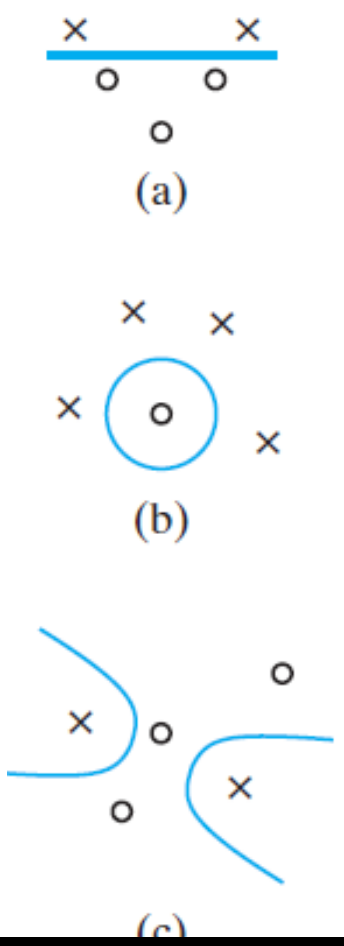
$$\mathbf{w}^T\varphi(x) = 0$$

Cover's Theorem Revisited

- Given a set X of N inputs picked from the input space independently, and suppose all the possible dichotomies of X are equiprobable
- Let $P(N, m_1)$ be the probability that a particular dichotomy picked at random is φ-separable, where the class of surfaces has m_1 degrees of freedom
- In this case

$$P(N, m_1) = \begin{cases} \left(\frac{1}{2}\right)^{N-1} \sum_{m=0}^{m_1-1} \binom{N-1}{m} & N > m_1 - 1 \\ 1 & N \leq m_1 - 1 \end{cases}$$ where

$$\binom{l}{m} = \frac{l!}{(l-m)!m!}$$



Three examples of φ-separable dichotomies of different sets of five points in two dimensions:

(a) linearly separable dichotomy

(b) spherically separable dichotomy

(c) quadrically separable dichotomy

Cover's Theorem:Interpretation

- Separability depends on
  1. particular dichotomy, and
  2. distribution of patterns in the input space
- Probability $P(N, m_1)$ states that the probability of being φ-separable is equivalent to the cumulative binomial distribution corresponding to the probability that $(N-1)$ flips of a fair coin will results in $(m_1 - 1)$ or fewer heads
- In sum, Cover's theorem has two basic ingredients:
  - Nonlinear mapping to hidden space with $\varphi_i(\mathbf{x}) \quad i = 1, 2, \ldots, m_1$
  - High dimensionality of hidden space compared to the input space $(m_1 > m_0)$
- Corollary: A maximum of $2m_1$ patterns can be linearly separated by a hidden space of dimensionality m_1

XOR Problem

To illustrate the idea of φ-separability of patterns

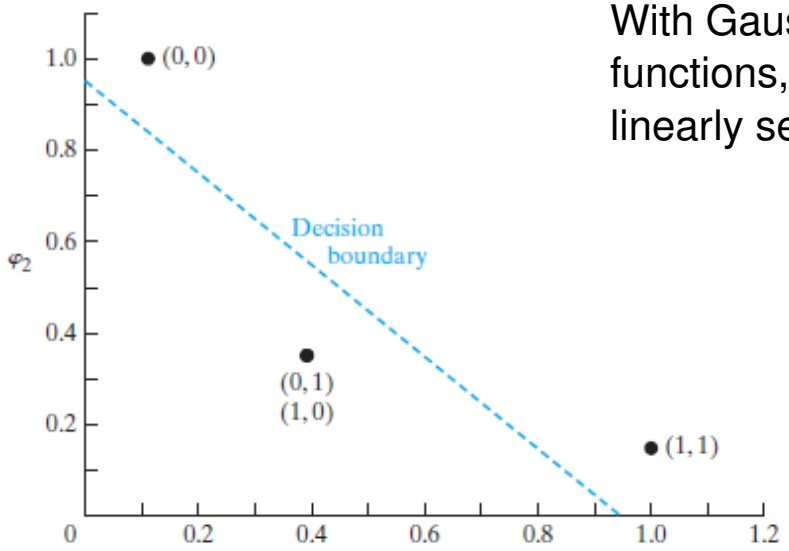
- Define a pair of Gaussian hidden function as follows

$$\begin{aligned} \varphi_1(\mathbf{x}) &= e^{-\|\mathbf{x}-\mathbf{t}_1\|^2}, \quad \mathbf{t}_1 = [1,1]^T \\ \varphi_2(\mathbf{x}) &= e^{-\|\mathbf{x}-\mathbf{t}_2\|^2}, \quad \mathbf{t}_2 = [0,0]^T \end{aligned} \tag{1}$$

- Compute for φ_1 and φ_2 for four input patterns

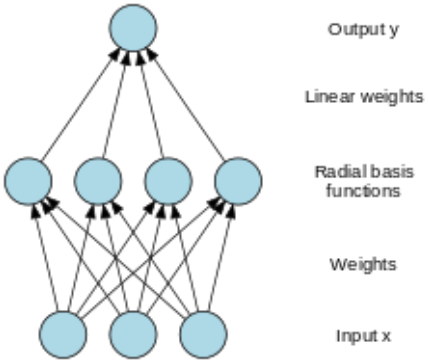
Table 1: Specification of the Hidden Functions for the XOR Problem

Input Pattern x	First Hidden Function φ_1	Second Hidden Function φ_2
(1,1)	1	0.1353
(0,1)	0.3678	0.3678
(0,0)	0.1353	1
(1,0)	0.3678	0.3678



With Gaussian hidden functions, the inputs become linearly separable

RBF Learning

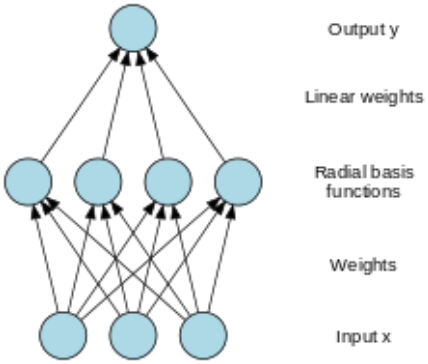


RBF learning problem boils down to two tasks:

- How to determine the parameters associated with the radial-basis function in the hidden layer $\varphi_i(x)$ (e.g., the center of the Gaussians)
- How to train the hidden-to-output weight?

RBF as an Interpolation Problem

- As per the Cover's theorem, a nonlinear mapping is used to transform a nonlinearly separable classification problem into a linearly separable one
- In similar ways, we may use a nonlinear mapping to transform a difficult nonlinear filtering problem into an easier one that involves linear filtering



- RBFN can be considered as a function map $s : \mathbb{R}^{m_0} \mapsto \mathbb{R}^1$
- Map s can be thought of as hypersurface $\Gamma \subset \mathbb{R}^{m_0+1}$
- Surface Γ is a multidimensional plot of output as a function of input
- Practically, Γ is unknown and training data is contaminated with noise
- Training and generalization phase of Γ
  - **Training**: fit hypersurface Γ to the training data points
  - **Generalization**: interpolate between data points, along the reconstructed surface Γ
- We are led to the theory of multivariable interpolation in high dimensional space

Interpolation Problem in Strict Sense

Given a set of N different points $\{\mathbf{x}_i \in \mathbb{R}^{m_0} | i = 1, 2, \ldots, N\}$ and a corresponding set of N real numbers $\{d_i \in \mathbb{R}^1 | i = 1, 2, \ldots, N\}$, find a function $F : \mathbb{R}^{m_0} \mapsto \mathbb{R}^1$ that satisfies the interpolation condition

$$F(\mathbf{x}_i) = d_i, \quad i = 1, 2, \ldots, N \tag{2}$$

- For strict interpolation, the interpolating surface (i.e., function F) is constrained to pass through all the training data
- Radial-basis functions (RBF) technique consists of choosing a function F of the following form

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|) \tag{3}$$

  where $\{\varphi_i(\|\mathbf{x} - \mathbf{x}_i\|) | i = 1, 2, \ldots, N\}$ is a set of N functions, known as radial-basis functions
- Known data points $\mathbf{x}_i \in \mathbb{R}^{m_0}$, $i = 1, 2, \ldots, N$ are taken to be the centers of RBF

- Eq (3) can be written as

$$F(\mathbf{x}_j) = \sum_{i=1}^{N} w_i \varphi(\|\mathbf{x}_j - \mathbf{x}_i\|) \tag{4}$$

- From eq(2) and eq(4) we can write

$$\sum_{i=1}^{N} w_i \varphi(\|\mathbf{x}_j - \mathbf{x}_i\|) = d_j \tag{5}$$

From eq(5) we may write

$$\begin{aligned}
w_1\varphi(\|\mathbf{x}_1 - \mathbf{x}_1\|) + w_2\varphi(\|\mathbf{x}_1 - \mathbf{x}_2\|) + \ldots + w_N\varphi(\|\mathbf{x}_1 - \mathbf{x}_N\|) &= d_1 \\
w_1\varphi(\|\mathbf{x}_2 - \mathbf{x}_1\|) + w_2\varphi(\|\mathbf{x}_2 - \mathbf{x}_2\|) + \ldots + w_N\varphi(\|\mathbf{x}_2 - \mathbf{x}_N\|) &= d_2 \\
\vdots &= \vdots \\
w_1\varphi(\|\mathbf{x}_N - \mathbf{x}_1\|) + w_2\varphi(\|\mathbf{x}_N - \mathbf{x}_2\|) + \ldots + w_N\varphi(\|\mathbf{x}_N - \mathbf{x}_N\|) &= d_N
\end{aligned} \tag{6}$$

- Eq(6) can be written as

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \tag{7}$$

where $\varphi_{ji} = \varphi(\|\mathbf{x}_j - \mathbf{x}_i\|),\ (j,i) = 1, 2, \ldots, N$

- Let $d = [d_1, d_2, \ldots, d_N]^T$ and $w = [w_1, w_2, \ldots, w_N]^T$ represents the desired response vector and linear weight vector
- Let $\Phi = \{\varphi_{ji} | (j,i) = 1, 2, \ldots, N\}$ and call it interpolation matrix
- Eq(7) can be written as $\Phi w = d$
- Assuming that Φ is nonsingular so that Φ^{-1} exists, we can solve for w i.e., $w = \Phi^{-1} d$

- How to ensure that Φ is nonsingular?
- Non-singularity of Φ is guaranteed by **Micchelli's Theorem**

Micchelli's Theorem

Let $\{\mathbf{x}_i\}_{i=1}^N$ be a set of distinct points in $\mathbb{R}^{m_0}$. Then $N \times N$ interpolation matrix Φ, whose ji-th element is $\varphi_{ji} = \varphi(\|\mathbf{x}_j - \mathbf{x}_i\|)$, is non-singular.

- When $m_1 < N$, we can find w that minimizes

$$\mathcal{E}(w) = \sum_{i=1}^{N} (F(\mathbf{x}_i) - d_i)^2$$

where $F(\mathbf{x}) = \sum_{k=1}^{m_1} w_k \varphi_k(\mathbf{x})$

- The solution involves the pseudoinverse of Φ

$$w = \underbrace{(\Phi^T \Phi)^{-1} \Phi^T}_{\text{pseudo inverse}} d$$

here Φ is $N \times m_1$ matrix

Typical RBFs

There is a large class of RBF covered by the Micchelli's theorem, following are of particular interest for RBFN.

- **Multiquadrics**:

$$\varphi(r) = (r^2 + c^2)^{1/2} \qquad \text{for some } c > 0 \text{ and } r \in \mathbb{R}$$

- **Inverse multiquadrics**:

$$\varphi(r) = \frac{1}{(r^2 + c^2)^{1/2}} \qquad \text{for some } c > 0 \text{ and } r \in \mathbb{R}$$

- **Gaussian functions**:

$$\phi(r) = exp(-\frac{r^2}{2\sigma^2}) \qquad \text{for some } \sigma > 0 \text{ and } r \in \mathbb{R}$$

Radial-Basis Function Networks

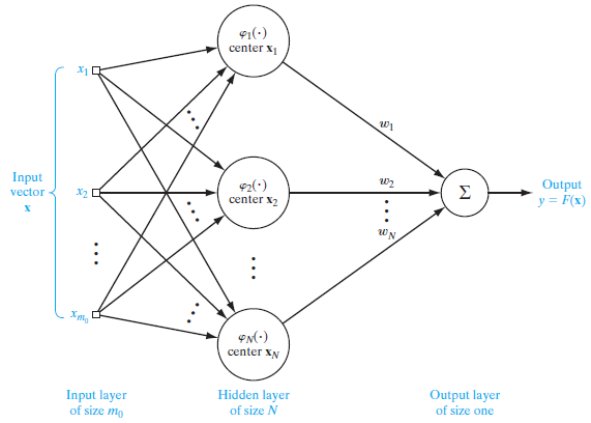


Figure 2: Structure of an RBF network, based on interpolation theory.

- **Input layer**: m_0 source nodes, where $\mathbf{x} \in \mathbb{R}^{m_0}$
- **Hidden layer**:$m_1 = N$, and each unit described by $\varphi_j(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{x}_j\|)$
- **Output layer**: $m_2 = 1$, however m_2 can be anything but $m_2 < m_1$

Now onwords the RBF for hidden units will be defined by

$$\begin{aligned} \varphi_j(\mathbf{x}) &= \varphi(\|\mathbf{x} - \mathbf{x}_j\|) \\ &= exp\left(-\frac{1}{2\sigma_j^2}\|\mathbf{x} - \mathbf{x}_j\|^2 \right), \quad j = 1, 2, \ldots, N \end{aligned} \tag{8}$$

- Formulation of RBFN via interpolation is neat
- Training sample $\{\mathbf{x}_i, d_i\}_{i=1}^N$ is typically noisy
- Use of interpolation based on noisy data could lead to misleading results
- We need a different approach for design of an RBFN
- There is redundancy of neurons in the hidden layer by virtue of the redundancy that may be inherent in the training sample
- It is a good design practice to make size of hidden layer a fraction of the size of training sample

Function approximation realized by both RBFN structure has same mathematical form

$$F(\mathbf{x}) = \sum_{j=1}^{K} w_j \varphi(\mathbf{x}, \mathbf{x}_j)$$

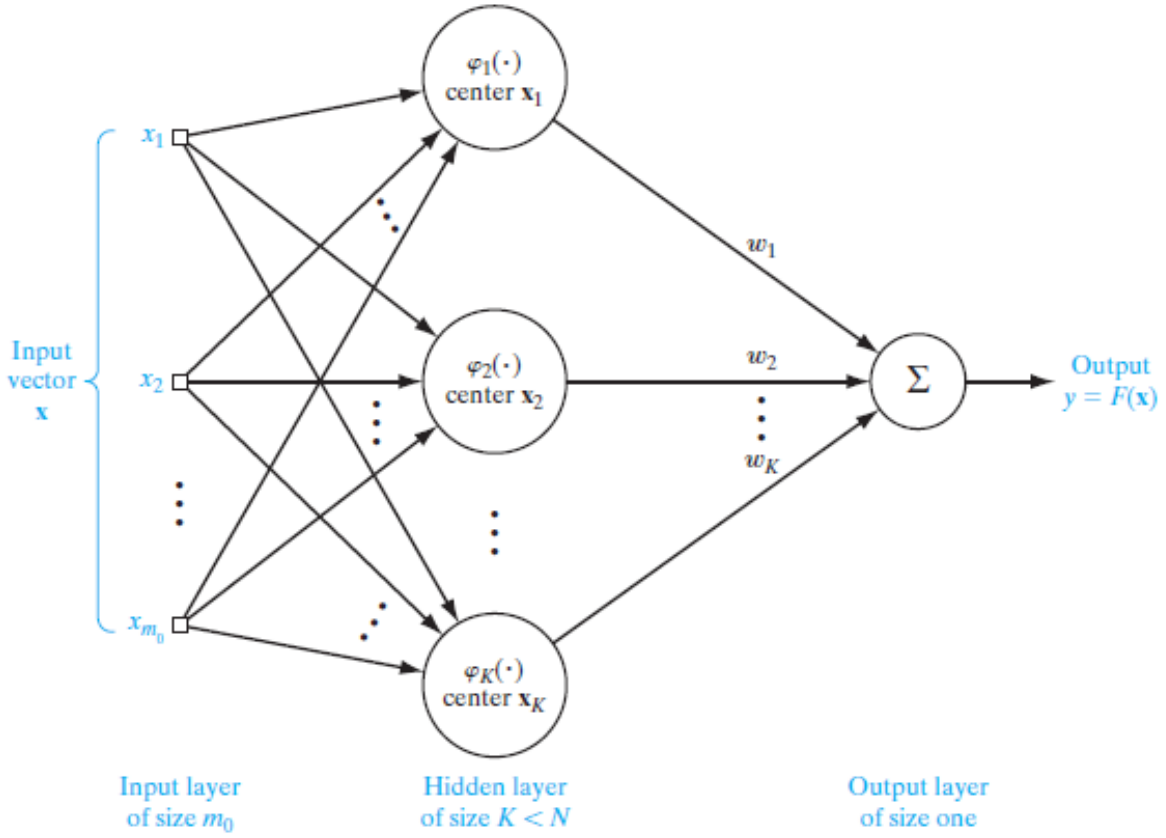
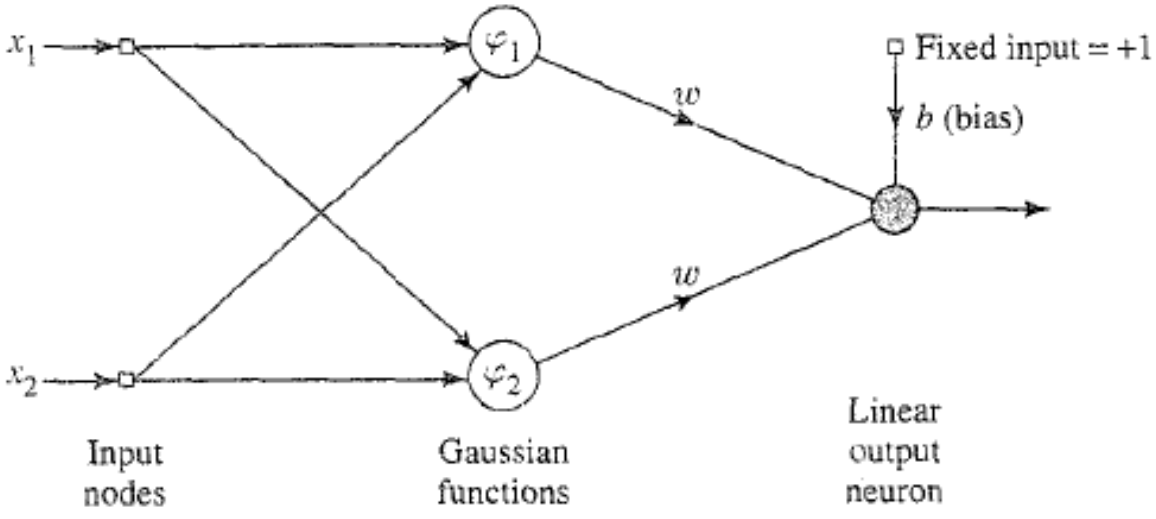


Figure 3: Structure of a practical RBF network.

XOR Problem (Revisited)



- RBFN to be investigated consist of a pair of Gaussian functions, defined as

$$\varphi(\|\mathbf{x} - \mathbf{t}_i\|) = exp(-\|\mathbf{x} - \mathbf{t}_i\|^2), \ i = 1, 2$$

where the centers $\mathbf{t}_1 = [1, 1]$ and $\mathbf{t}_2 = [0, 0]$

- The input-output relation of the network is defined by

$$y(\mathbf{x}) = \sum_{i=1}^{2} w\varphi(\|\mathbf{x} - \mathbf{t}_i\|) + b$$

- To fit the training data of Table 1, we require that

$$y(\mathbf{x}_j) = d_j, \quad j = 1, 2, 3, 4$$

- Let $\varphi_{ji} = \varphi(\|\mathbf{x}_j - \mathbf{t}_i\|), \quad j = 1, 2, 3, 4; i = 1, 2$
- Using Table 1 in above relation we get following equation written in matrix form as

$$\varphi w = d$$

where $\varphi = \begin{bmatrix} 1 & 0.1353 & 1 \\ 0.3678 & 0.3678 & 1 \\ 0.1353 & 1 & 1 \\ 0.3678 & 0.3678 & 1 \end{bmatrix}$, $w = [w, w, b]^T$ and $d = [0, 1, 0, 1]^T$

- The problem here is overdetermined in the sense that we have more data points than free parameters
- $w = \varphi^+ d = (\varphi^T \varphi)^{-1} \varphi^T d$
- $\varphi^+ = \begin{bmatrix} 1.8292 & -1.2509 & 0.6727 & -1.2509 \\ 0.6727 & -1.2509 & 1.8292 & -1.2509 \\ -0.9202 & 1.4202 & -0.9202 & 1.4202 \end{bmatrix}$
- Finally we get, $w = [-2.5018, -2.5018, 2.8404]$, which completes the specification of the RBFN

Comparison of RBFN with MLP

RBFN and MLP are examples of nonlinear layered feedforward networks

- A RBFN has a single hidden layer, whereas an MLP may have more than one
- Computational nodes of an MLP, located in a hidden layer or an output layer, share a common neuronal model. Computational nodes of RBFN have different neuronal model
- Argument of the activation function of each hidden neuron in RBFN computes the Euclidean norm between the input vector and the center of that unit. Meanwhile, the activation function of each hidden unit in an MLP computes the inner product of the input vector and synaptic weight vector of that unit
- MLPs construct global approximations to nonlinear input-output mapping. RBFN uses exponentially decaying localized non-linearities to construct local approximations to nonlinear input-output mappings.