RADIAL BASIS FUNCTION NETWORKS

COVER'S THEOREM ON THE SEPARABILITY OF PATTERNS

When a radial-basis function (RBF) network is used to perform a complex pattern classification task, the problem is basically solved by transforming it into a high dimensional space in a nonlinear manner. The underlying justification is found in Cover's theorem on the seperability of patterns, which, in qualitative terms, may be stated as follows (Cover, 1965):

"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."

Consider a family of surfaces where each naturally divides an input space into two regions.

Let $\mathcal{X}$ denote a set of N patterns (vectors) $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, each of which is assigned to one of two classes $\mathcal{X}_1$ and $\mathcal{X}_2$ This dichotomy (binary partition) of the points is said to be separable with respect to the family of surfaces if a surface exists in the family that separates the points in the class $\mathcal{X}_1$ from those in the class $\mathcal{X}_2$For each pattern $\mathbf{x} \in \mathcal{X}$ define a vector made up of a set of real-valued functions $\{\varphi_i(\mathbf{x}) | i = 1, 2, \ldots, m_1\}$ as shown by

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

Suppose that the pattern x is a vector in an m_0-dimensional input space. The vector $\boldsymbol{\varphi}(\mathbf{x})$ then maps points in mo-dimensional input space into corresponding points in a new space of dimension m_1. We refer to $\varphi_i(\mathbf{x})$ as a hidden function, because it plays a role similar to that of a hidden unit in a feed forward neural network. Correspondingly, the space spanned by the set of hidden functions $\{\varphi_i(\mathbf{x})\}_{i=1}^{m_1}$ is referred to as the hidden space or feature space.

A dichotomy $\{\mathcal{X}_1, \mathcal{X}_2\}$of $\mathcal{X}$ is said to be $\boldsymbol{\Phi}$-separable if there exists an m0 dimensional vector w such that we may write (Cover, 1965)

$$\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) > 0, \qquad \mathbf{x} \in \mathcal{X}_1$$

$$\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) < 0, \qquad \mathbf{x} \in \mathcal{X}_2$$

The hyperplane defined by the equation

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

describes the separating surface in the Φ-space (i.e., hidden space).

The inverse image of this hyperplane, that is,

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

defines the separating surface in the input space.

Consider a natural class of mappings obtained by using a linear combination of r-wise products of the pattern vector coordinates. The separating surfaces corresponding to such mappings are referred to as rth-order rational varieties. A rational variety of order r in a space of dimension mo is described by an rth degree homogeneous equation in the coordinates of the input vector x, as shown by

$$\sum_{0 \le i_1 \le i_2 \le \cdots \le i_r \le m_0} a_{i_1 i_2 \ldots i_r} x_{i_1} x_{i_2} \ldots x_{i_r} = 0$$

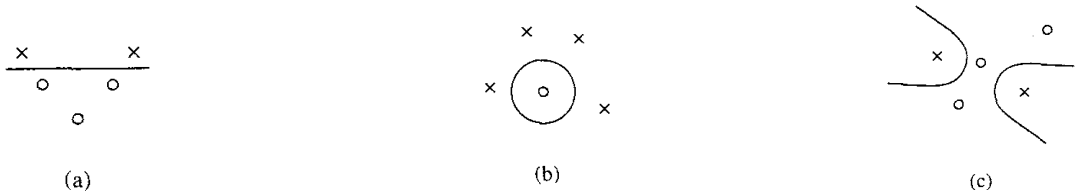
where x_i is the ith component of input vector x, and x_0 is set equal to unity in order to express the equation in a homogeneous form.

An rth order product of entries x_i of x that is, $x_{i_1} x_{i_2} \ldots x_{i_r}$ is called a monomial. For an input space of dimensionality mo there are

$$\frac{(m_0 - r)!}{m_0! r!}$$

monomials.

Examples of the type of separating surfaces are hyperplanes (first-order rational varieties), quadrices (second-order rational varieties), and hyperspheres (quadrics with certain linear constraints on the coefficients). These examples are illustrated in following Figure for a configuration of five points in a two-dimensional input space.



(a) (b) (c)

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.
In general, linear separability implies spherical separability which implies quadric separability; however, the converses are not necessarily true.

Let $P(N, m_1)$ denote the probability that a particular dichotomy picked at random is φseparable, where the class of separating surfaces chosen has m, degrees of freedom.
We may then state that

$$P(N, m_1) = \left(\frac{1}{2}\right)^{N-1} \sum_{m=0}^{m_1-1} \binom{N-1}{m}$$

where the binomial coefficients comprising N - 1 and m are themselves defined for all integer I and m by

$$\binom{l}{m} = \frac{l(l-1)\cdots(l-m+1)}{m!}$$

Equation (5.5) embodies the essence of Cover,' separability theorem for random patterns
2 It is a statement of the fact that the cumulative binomial distribution corresponding
to the probability that (N - 1) flips of a fair coin wilI result in $(m_1 - .1)$ or
fewer heads.

Cover's theorem on the separability of patterns encompasses two basic ingredients:

1. Nonlinear formulation of the hidden function defined by $\varphi_i(\mathbf{x})$, where x is the input vector and $i = 1, 2, \ldots, m_1$

2. High dimensionality of the hidden space compared to the input space; that dimensionality is determined by the value assigned to m_1 (i.e., the number of hidden units).

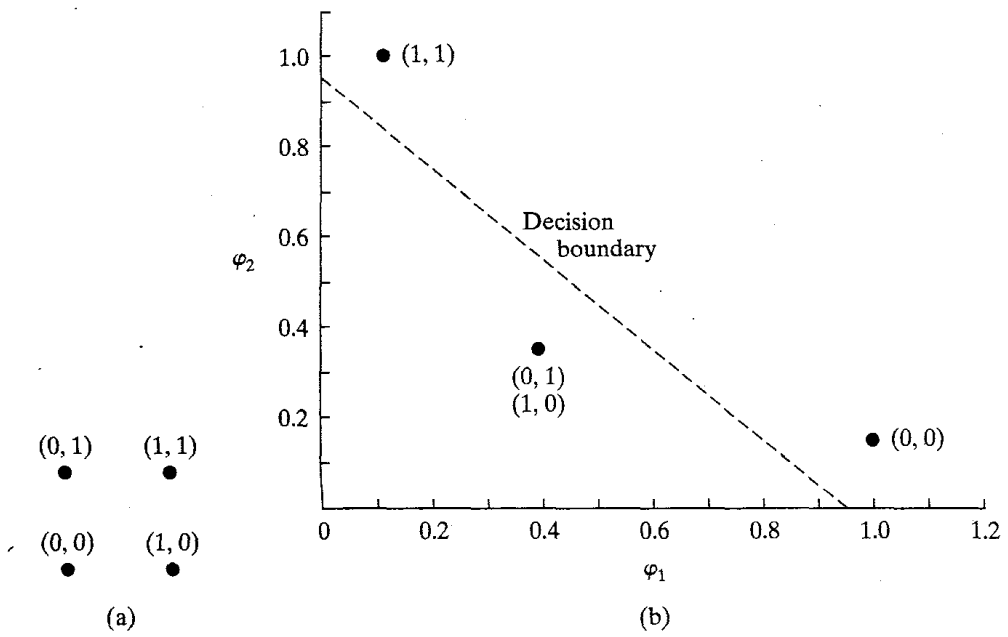
Example. The XOR Problem

To illustrate the significance of the idea of φ-separability of patterns, consider the simple yet important XOR problem. In the XOR problem there are four points (patterns): (1, 1), (0, 1),(0, 0), and (1, 0), in a two-dimensional input space, as depicted in Fig. a. The requirement is to construct a pattern classifier that produces the binary output 0 in response to the input pattern (1, 1), or (0, 0), and the binary output 1 in response to the input pattern (0, 1) or (1. 0). Thus points that are closest in the input space, in terms of the Hamming distance, map to regions that are maximally apart in the output space.

Define a pair of Gaussian hidden functions as follows:

$$\varphi_1(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{t}_1\|^2}, \qquad \mathbf{t}_1 = [1, 1]^T$$
$$\varphi_2(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{t}_2\|^2}, \qquad \mathbf{t}_2 = [0, 0]^T$$

We may then construct the results summarized in Table for the four different input patterns of interest. The input patterns are mapped onto the $\varphi_1 - \varphi_2$ plane as shown in Fig. b.
Here we now see that the input patterns (0, 1) and (1 , 0) are linearly separable from the remaining input patterns (1 , 1) and (0, 0). Thereafter, the XOR problem may be readily solved by using the functions $\varphi_1(\mathbf{x})$ and $\varphi_2(\mathbf{x})$ as the inputs to a linear classifier such as the perceptron.
In this example there is no increase in the dimensionality of the hidden space compared to the input space. In other words, nonlinearity exemplified by the use of Gaussian hidden functions is sufficient to transform the XOR problem into a linearly separable one.



(a) The four patterns of the XOR problem (b) Decision-making diagram.

Input Pattern, $\mathbf{x}$	First Hidden Function, $\varphi_1(\mathbf{x})$	Second Hidden Function, $\varphi_2(\mathbf{x})$
(1,1)	1	0.1353
(0,1)	0.3678	0.3678
(0,0)	0.1353	1
(1,0)	0.3678	0.3678

TABLE : Specification of the Hidden Functions for the XOR Problem of Example

INTERPOLATION PROBLEM

Basically, a nonlinear mapping is used to transform a nonlinearly separable classification problem into a linearly separable one. In a similiar way, we may use a nonlinear mapping to transform a difficult nonlinear filtering problem into an easier one that involves linear filtering.

Consider then a feedforward network with an input layer, a single hidden layer, and an output layer consisting of a single unit. We have purposely chosen a single output unit to simplify the exposition without loss of generality. The network is designed to perform a nonlinear mapping from the input space to the hidden space, followed by a linear mapping from the hidden space to the output space. Let m_0 denote the dimension of the input space. Then, in an overall fashion, the network represents a map from the mo-dimensional input space to the single-dimensional output space, written as

$$s: \mathbb{R}^{m_0} \rightarrow \mathbb{R}^1$$

We may think of the map s as a hypersurface (graph) $\Gamma \subset \mathbb{R}^{m_0+1}$,just as we think of the elementary map $s: \mathbb{R}^1 \rightarrow \mathbb{R}^1$where $s(x) = x^2$, as a parabola drawn in $\mathbb{R}^2$ space. The surface Γ is a multidimensional plot of the output as a function of the input. In a practical situation, the surface Γ is unknown and the training data are usually contaminated with noise.

The training phase and generalization phase of the learning process may be respectively viewed as follows (Broomhead and Lowe, 1988):

→The **training phase** constitutes the optimization of a fitting procedure for the surface Γ, based on known data points presented to the network in the form of input-output examples (patterns).

→The **generalization phase** is synonymous with interpolation between the data points, with the interpolation being performed along the constrained surface generated by the fitting procedure as the optimum approximation to the truesurface Γ.

The interpolation problem, in its strict sense, may be stated:

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 $F: \mathbb{R}^N \rightarrow \mathbb{R}^1$ that satisfies the interpolation condition:

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

For strict interpolation as specified here, the interpolating surface (i.e., function F) is constrained to pass through all the training data points.

The radial-basis functions (RBF) technique consists of choosing a function F that has the following form (Powell, 1988):

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

Where $\{\varphi(\|\mathbf{x} - \mathbf{x}_i\|) | i = 1, 2, \ldots, N\}$ is a set of N arbitrary (generally nonlinear) functions, known as radial-basis functions, and $\| \cdot \|$ denotes a norm that is usually Euclidean. The known data points $\{\varphi(\|\mathbf{x} - \mathbf{x}_i\|) | i = 1, 2, \ldots, N\}$ are taken to be the centers of the radial basis functions.

Inserting the interpolation conditions of above two equations, we obtain the following set of simultaneous linear equations for the unknown coefficients (weights) of the expansion $\{w_i\}$:

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

Where

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

Let

$$\mathbf{d} = [d_1, d_2, \ldots, d_N]^T$$

$$\mathbf{w} = [w_1, w_2, \ldots, w_N]^T$$

The N-by-1 vectors d and w represent the desired response vector and linear weight vector, respectively, where N is the size of the training sample.

Let $\mathbf{\Phi}$ denote an N-by-N matrix with elements φ_{ji}:

$$\mathbf{\Phi} = \{\varphi_{ji} | (j, i) = 1, 2, \ldots, N\}$$

We call this matrix the interpolation matrix. We may then rewrite equation in the compact form

$$\mathbf{\Phi w} = \mathbf{x}$$

Assuming that $\mathbf{\Phi}$ is nonsingular and therefore that the inverse matrix $\mathbf{\Phi}^{-1}$exists,

$$\mathbf{w} = \mathbf{\Phi}^{-1}\mathbf{x}$$

The vital question is: How can we be sure that the interpolation matrix $\mathbf{\Phi}$ is nonsingular? The answer to this question is given in the following important theorem.

Micchelli's Theorem

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

There is a large class of radial-basis functions that is covered by Micchelli's theorem; it includes the following functions that are of particular interest in the study of RBF networks:

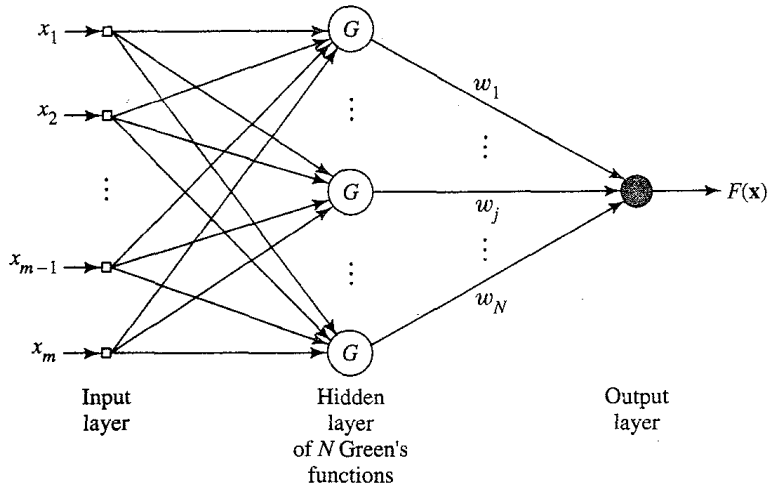
1. Multiquadrics:
   $$\varphi(r) = (r^2 + c^2)^{1/2} \quad \text{for some } c > 0 \text{ and } r \in \mathbb{R}$$
2. Inverse multiquadries:
   $$\varphi(r) = \frac{1}{(r^2 + c^2)^{1/2}} \quad \text{for some } c > 0 \text{ and } r \in \mathbb{R}$$
3. Gaussian functions:
   $$\varphi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad \text{for some } \sigma > 0 \text{ and } r \in \mathbb{R}$$

The inverse multiquadrics and the Gaussian functions share a common property: They are both localized functions, in the sense that $\varphi(r) \to 0$ as $r \to \infty$. In both of these cases the interpolation matrix $\mathbf{\Phi}$ is positive definite. By contrast, the multiquadrics are nonlocal in that $\varphi(r)$ becomes unbounded as $r \to \infty$ and the corresponding interpolation matrix $\mathbf{\Phi}$ has N-1 negative eigen values and only one positive eigenvalue, with the result that it is not positive definite (Micchelli,1986).

REGULARIZATION NETWORKS

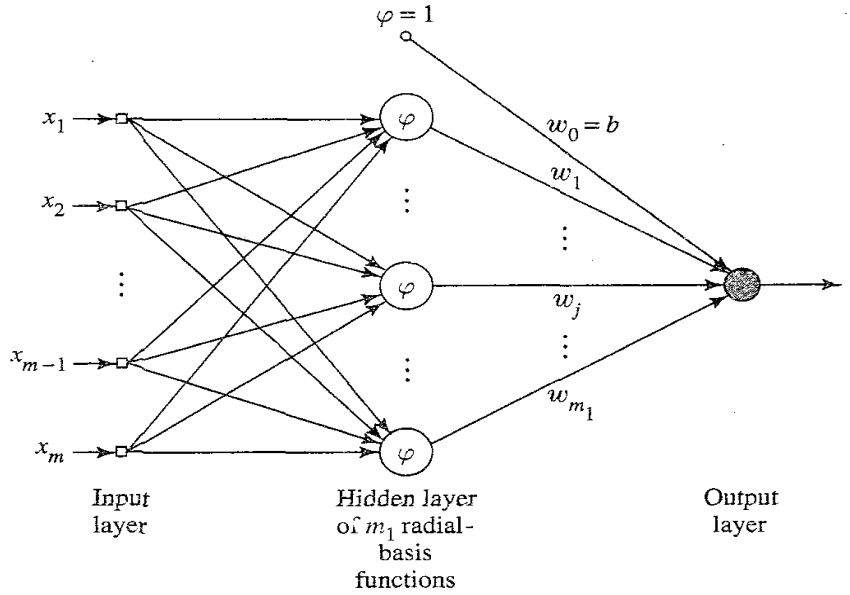
A regularization network consists of three layers. The first layer is composed of input (source) nodes whose number is equal to the dimension mo of the input vector x (i.e., the number of independent variables of the problem). The second layer is a hidden layer, composed of nonlinear units that are connected directly to all of the nodes in the input layer. There is one hidden unit for each data point Xi i = 1 , 2, . . . , N, where N is the size of the training sample. The activation functions of the individual hidden units are defined by the Green's functions. Accordingly, the output of the ith hidden unit is $G(\mathbf{x}, \mathbf{x}_i)$. The output layer consists of a single linear unit, being fully connected to the hidden layer. By "linearity" we mean that the output of the network is a linearly weighted sum of the outputs of the hidden units. The weights of the output layer are the unknown coefficients of the expansion, defined in terms of the Green's functions $G(\mathbf{x}, \mathbf{x}_i)$ and the regularization parameter λ.

Following Figure 5.4 depicts the architecture of the regularization network for a single output. Clearly, such an architecture can be readily extended to accommodate any number of network outputs desired.



Regularization network.

The regularization network shown in following diagram assumes that the Green's function $G(\mathbf{x}, \mathbf{x}_i)$ is positive definite for all i. Provided that this condition is satisfied, which it is in the case of the $G(\mathbf{x}, \mathbf{x}_i)$ having the Gaussian form, then the solution produced by this network will be an "optimal" interpolant in the sense that it minimizes the functional $\mathscr{E}(F)$.



Radial-basis function network

The regularization network has three desirable properties (Poggio and Girosi, 1990a):
1. The regularization network is a universal approximator in that it can approximate arbitrarily well any multivariate continuous function on a compact subset of $\mathbb{R}^{m_0}$, given a sUfficiently large number of hidden units.
2. Since the approximation scheme derived from regularization theory is linear in the unknown coefficients. it follows that the regularization network has the best approximation property. This means that given an unknown nonlinear function f, there always exists a choice of coefficients that approximates f better than all other possible choices.
3. The solution computed by the regularization network is optimal. Optimality here means that the regularization network minimizes a functional that measures how much the solution deviates from its true value as represented by the training data.

REGULARIZATION THEORY

In 1963, Tikhonov proposed a new method called regularization for solving ill-posed problems. The basic idea of regularization is to stabilize the solution by means of some auxiliary nonnegative functional that embeds prior information about the solution. The most common form of prior information involves the assumption that the input-output mapping function is smooth, in the sense that similar inputs correspond to similar outputs.
To be specific, let the set of input-output data available for approximation be described by
Input signal: $\mathbf{x}_i \in \mathbb{R}^{m_0}, \quad i = 1, 2, \ldots, N$
Desired response: $d_i \in \mathbb{R}^1, \quad i = 1, 2, \ldots, N$
Let the approximating function be denoted by F(x).
Basically, Tikhonov's regularization theory involves two terms:

1. Standard Error Term: This first term, denoted by $\mathscr{E}_s(F)$, measures the standard error (distance) between the desired (target) response d_i, and the actual response y_i for training example i = 1, 2, . . . , N. Specifically, we define

$$\mathscr{E}_s(F) = \frac{1}{2} \sum_{i=1}^{N} (d_i - y_i)^2$$
$$= \frac{1}{2} \sum_{i=1}^{N} [d_i - F(\mathbf{x}_i)]^2$$

2. Regularizing Term: This second term, denoted for $\mathscr{E}_c(F)$, depends on the "geometric" properties of the approximating function F(x). Specifically, we write

$$\mathscr{E}_c(F) = \frac{1}{2} \|\mathbf{D}F\|^2$$

where D is a linear differential operator. Prior information about the form of the solution [i.e., the input-output mapping function F(x)] is embedded in the operator D, which naturally makes the selection of D problem-dependent. We also refer to D as a stabilizer because it stabilizes the solution to the regularization problem, making it smooth and thereby satisfying the property of continuity.
Thus, the symbol $\|\cdot\|$ denotes a norm imposed on the function space to which DF(x) belongs.
The function f(x) used here denotes the actual function that defines the underlying physical process responsible for generating the set of input-output data pairs.
The quantity to be minimized in regularization theory is

$$\mathscr{E}(F) = \mathscr{E}_s(F) + \lambda \mathscr{E}_c(F)$$
$$= \frac{1}{2} \sum_{i=1}^{N} [d_i - F(\mathbf{x}_i)]^2 + \frac{1}{2} \lambda \|\mathbf{D}F\|^2$$

Where λ is a positive real number is is called the regularization parameter and $\mathscr{E}(F)$ is called the Tikhonov functional.
The minimizer of the Tikhonov functional $\mathscr{E}(F)$ (i.e., the solution to the regularization problem) is denoted by $F_\lambda(\mathbf{x})$.

Frechet Differential of the Tikhonov Functional

The principle of regularization may now be stated as:

Find the function $F_\lambda(\mathbf{x})$ that minimizes the Tikhonov functional $\mathscr{E}(F)$, defined by

$$\mathscr{E}(F) = \mathscr{E}_s(F) + \lambda\mathscr{E}_c(F)$$

where $\mathscr{E}_s(F)$ is the standard error term, $\mathscr{E}_c(F)$ is the regularizing term, and λ is the regularization parameter.

To proceed with the minimization of the cost functional $\mathscr{E}(F)$, we need a rule for evaluating the differential of $\mathscr{E}(F)$.

The Frechet differential of the functiona $\mathscr{E}(F)$ is formally defined by:

$$d\mathscr{E}(F, h) = \left[\frac{d}{d\beta}\mathscr{E}(F + \beta h)\right]_{\beta=0}$$

where h(x) is a fixed function of the vector x.

A necessary condition for the function F(x) to be a relative extremum of the functional $\mathscr{E}(F)$ is that the Frechet differential $d\mathscr{E}(F, h)$ must be zero at F(x) for all $h \in \mathcal{H}$, as shown by

$$d\mathscr{E}(F, h) = d\mathscr{E}_s(F, h) + \lambda d\mathscr{E}_c(F, h) = 0$$

where $d\mathscr{E}_s(F, h)$ and $d\mathscr{E}_c(F, h)$ are the Frechet differentials of the functionals $\mathscr{E}_s(F)$ and $\mathscr{E}_c(F)$.

→Evaluating the Frechet differential of the standard error term $\mathscr{E}_s(F, h)$ we have

$$\begin{aligned} d\mathscr{E}_s(F, h) &= \left[\frac{d}{d\beta}\mathscr{E}_s(F + \beta h)\right]_{\beta=0} \\ &= \left[\frac{1}{2}\frac{d}{d\beta}\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i) - \beta h(\mathbf{x}_i)]^2\right]_{\beta=0} \\ &= -\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i) - \beta h(\mathbf{x}_i)]h(\mathbf{x}_i)|_{\beta=0} \\ &= -\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i)]h(\mathbf{x}_i) \end{aligned}$$

Hence, in light of the Riesz representation theorem, we may rewrite the Frechet differential $d\mathscr{E}_s(F, h)$ in the equivalent form

$$d\mathscr{E}_s(F, h) = -\left(h, \sum_{i=1}^{N}(d_i - F)\delta_{\mathbf{x}_i}\right)_{\mathcal{H}}$$

The symbol $(\cdot,\cdot)$ used here stands for the inner (scalar) product of two functions in $\mathcal{H}$ space.

where $\delta_{\mathbf{x}_i}$. denotes the Dirac delta distribution of x, centered at $\mathbf{X}_i$; that is $\delta_{\mathbf{x}_i}(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_i)$

→Evaluation of the Frechet differential of the regularizing term $\mathscr{E}_c(F)$ is as follows.

$$\begin{aligned} d\mathscr{E}_c(F, h) &= \frac{d}{d\beta}\mathscr{E}_c(F + \beta h)|_{\beta=0} \\ &= \frac{1}{2}\frac{d}{d\beta}\int_{\mathbb{R}^{m_0}}(\mathbf{D}[F + \beta h])^2 d\mathbf{x}|_{\beta=0} \\ &= \int_{\mathbb{R}^{m_0}}\mathbf{D}[F + \beta h]\mathbf{D}h\, d\mathbf{x}|_{\beta=0} \\ &= \int_{\mathbb{R}^{m_0}}\mathbf{D}F\,\mathbf{D}h\, d\mathbf{x} \\ &= (\mathbf{D}h, \mathbf{D}F)_{\mathcal{H}} \end{aligned}$$

where $(\mathbf{D}h, \mathbf{D}F)_{\mathcal{H}}$ is the inner product of the two functions Dh(x) and DF(x) that result from the action of the differential operator D and h(x) and F(x), respectively.

Euler-Lagrange Equation

Given a linear differential operator D, we can find a uniquely determined adjoint operator, denoted by $\tilde{\mathbf{D}}$, such that for any pair of functions u(x) and v(x) which are sufficiently differentiable and which satisfy proper boundary conditions, we can write

$$\int_{\mathbb{R}^m} u(\mathbf{x})\mathbf{D}v(\mathbf{x})d\mathbf{x} = \int_{\mathbb{R}^m} v(\mathbf{x})\tilde{\mathbf{D}}u(\mathbf{x})d\mathbf{x}$$

This equation is called Green's identity; it provides a mathematical basis for defining the adjoint operator $\tilde{\mathbf{D}}$ in terms of the given differential D.

We may make the following identifications:

$$u(\mathbf{x}) = \mathbf{D}F(\mathbf{x})$$

$$\mathbf{D}v(\mathbf{x}) = \mathbf{D}h(\mathbf{x})$$

Using Green's identity, we may rewrite in the equivalent form

$$d\mathscr{E}_c(F, h) = \int_{\mathbb{R}^{m_0}} h(\mathbf{x})\tilde{\mathbf{D}}\mathbf{D}F(\mathbf{x})d\mathbf{x}$$

$$= (h, \tilde{\mathbf{D}}\mathbf{D}F)_{\mathscr{H}}$$

where $\tilde{\mathbf{D}}$ is the adjoint of **D**.

Substituting the Frechet differentials of $d\mathscr{E}_s(F, h)$ and $d\mathscr{E}_c(F, h)$ in $d\mathscr{E}(F, h)$, we may now express the Frechet differential $d\mathscr{E}(F, h)$ as

$$d\mathscr{E}(F, h) = \left(h, \left[\tilde{\mathbf{D}}\mathbf{D}F - \frac{1}{\lambda}\sum_{i=1}^{N}(d_i - F)\delta_{\mathbf{x}_i}\right]\right)_{\mathscr{H}}$$

Since the regularization parameter λ is ordinarily assigned a value somewhere in the open interval [0, ∞] , the Frechet differential $d\mathscr{E}(F, h)$ is zero for every h(x) in $\mathscr{H}$ space if and only if the following condition is satisfied in the distributional sense:

$$\tilde{\mathbf{D}}\mathbf{D}F_\lambda - \frac{1}{\lambda}\sum_{i=1}^{N}(d_i - F)\delta_{\mathbf{x}_i} = 0$$

or equivalently,

$$\tilde{\mathbf{D}}\mathbf{D}F_\lambda(\mathbf{x}) = \frac{1}{\lambda}\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i)]\delta(\mathbf{x} - \mathbf{x}_i)$$

This equation is the Euler-Lagrange equation for the Tikhonov functional $\mathscr{E}(F)$ that defines a necessary condition for the Tikhonov functional $\mathscr{E}(F)$ to have an extremum at $F_\lambda(\mathbf{x})$.

Green's Function

Let $G(\mathbf{x}, \boldsymbol{\xi})$ denote a function in which both vectors x and $\boldsymbol{\xi}$ appear on equal footing but for different purposes: x as a parameter and $\boldsymbol{\xi}$ as an argument.

For a given linear differential operator L, we stipulate that the function $G(\mathbf{x}, \boldsymbol{\xi})$ satisfies the following conditions:

1. For a fixed $\boldsymbol{\xi}$, $G(\mathbf{x}, \boldsymbol{\xi})$ is a function of x and satisfies the prescribed boundary conditions.

2. Except at the point x = $\boldsymbol{\xi}$, the derivatives of $G(\mathbf{x}, \boldsymbol{\xi})$ with respect to x are all continuous; the number of derivatives is detennined by the order of the operator L.

3. With $G(\mathbf{x}, \boldsymbol{\xi})$ considered as a function of x, it satisfies the partial differential equation $\mathbf{L}G(\mathbf{x}, \boldsymbol{\xi}) = 0$ everywhere except at the point x = $\boldsymbol{\xi}$, where it has a singularity. That is, the function $G(\mathbf{x}, \boldsymbol{\xi})$ satisfies the following partial differential equation

$$\mathbf{L}G(\mathbf{x}, \boldsymbol{\xi}) = \delta(\mathbf{x} - \boldsymbol{\xi})$$

where, as defined previously, $\delta(\mathbf{x} - \boldsymbol{\xi})$ is the Dirac delta function positioned at the point x = $\boldsymbol{\xi}$.
The function $G(\mathbf{x}, \boldsymbol{\xi})$ thus described is called the Green's junction for the differential operator L.
The Green's function plays a role for a linear differential operator that is similar to that for the inverse matrix for a matrix equation.
Let $\varphi(\mathbf{x})$ denote a continuous or piecewise continuous function of $\mathbf{x} \in \mathbb{R}^{m_0}$. Then the function

$$F(\mathbf{x}) = \int_{\mathbb{R}^{m_0}} G(\mathbf{x}, \boldsymbol{\xi})\varphi(\boldsymbol{\xi})d\boldsymbol{\xi}$$

is a solution of the differential equation
$\mathbf{L}F(\mathbf{x}) = \varphi(\mathbf{x})$ where $G(\mathbf{x}, \boldsymbol{\xi})$ is the Green's function for the linear differential operator L.
To prove the validity of F(x) as a solution, apply the differential operator L

$$\mathbf{L}F(\mathbf{x}) = \mathbf{L}\int_{\mathbb{R}^{m_0}} G(\mathbf{x}, \boldsymbol{\xi})\varphi(\boldsymbol{\xi})d(\boldsymbol{\xi})$$

$$= \int_{\mathbb{R}^{m_0}} \mathbf{L}G(\mathbf{x}, \boldsymbol{\xi})\varphi(\boldsymbol{\xi})d\boldsymbol{\xi}$$

The differential operator L treats $\boldsymbol{\xi}$ as a constant, acting on the kernel $G(\mathbf{x}, \boldsymbol{\xi})$ only as a function of x. Here we get

$$\mathbf{L}F(\mathbf{x}) = \int_{\mathbb{R}^{m_0}} \delta(\mathbf{x} - \boldsymbol{\xi})\varphi(\boldsymbol{\xi})d\boldsymbol{\xi}$$

Finally, using the sifting property of the Dirac delta function, namely

$$\int_{\mathbb{R}^{m_0}} \varphi(\boldsymbol{\xi})\delta(\mathbf{x} - \boldsymbol{\xi})d(\boldsymbol{\xi}) = \varphi(\mathbf{x})$$

Finally we obtain $\mathbf{L}F(\mathbf{x}) = \varphi(\mathbf{x})$.

Solution to the Regularization Problem

While solving the Euler-Lagrange equation, set

$$\mathbf{L} = \tilde{\mathbf{D}}\mathbf{D}$$

and

$$\varphi(\boldsymbol{\xi}) = \frac{1}{\lambda}\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i)]\delta(\boldsymbol{\xi} - \mathbf{x}_i)$$

Then we may use following to write

$$F_\lambda(\mathbf{x}) = \int_{\mathbb{R}^{m_0}} G(\mathbf{x}, \boldsymbol{\xi})\left\{\frac{1}{\lambda}\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i)]\delta(\boldsymbol{\xi} - \mathbf{x}_i)\right\}d\boldsymbol{\xi}$$

$$= \frac{1}{\lambda}\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i)]\int_{\mathbb{R}^{m_0}} G(\mathbf{x}, \boldsymbol{\xi})\delta(\boldsymbol{\xi} - \mathbf{x}_i)d\boldsymbol{\xi}$$

where in the last line we have interchanged the order of integration and summation.
Finally, using the sifting property of the Dirac delta function, we get the desired solution to the Euler-Lagrange equation as follows:

$$F_\lambda(\mathbf{x}) = \frac{1}{\lambda}\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i)]G(\mathbf{x}, \mathbf{x}_i)$$

The above equation states that the minimizing solution $F_\lambda(\mathbf{x})$ to the regularization problem is a linear superposition of N Green's functions.

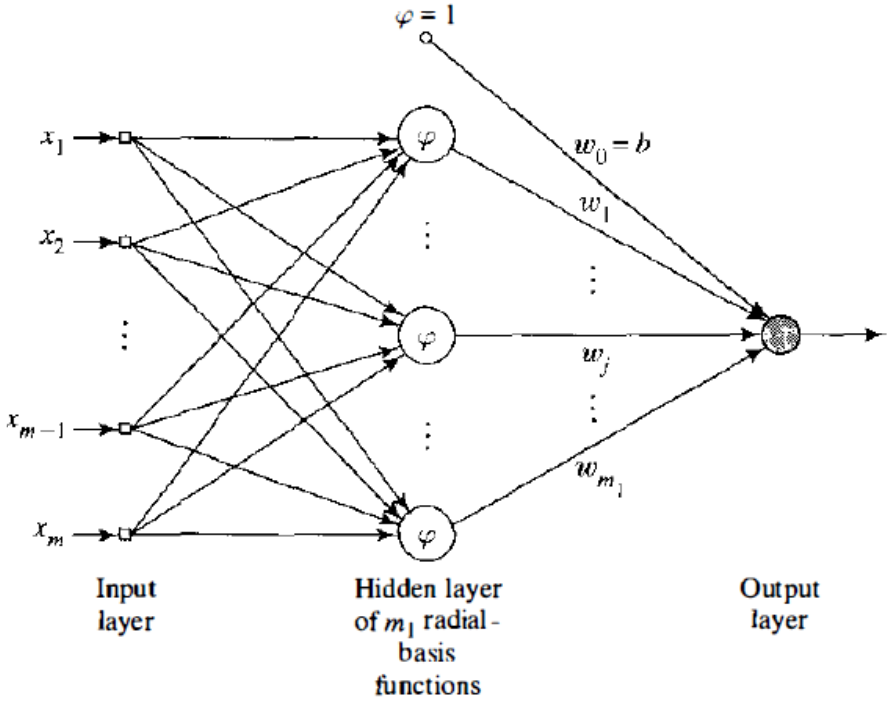
RADIAL BASIS FUNCTION NETWORKS

Radial Basis Function network was formulated by Broomhead and Lowe in 1988. Since Radial basis functions (RBFs) have only one hidden layer, the convergence of optimization objective is much faster, and despite having one hidden layer RBFs are proven to be universal approximators. RBF networks have many applications like function approximation, interpolation, classification and time series prediction. All these applications serve various industrial interests like stock price prediction, anomaly detection in data, fraud detection in financial transaction etc.

A **radial basis function network** is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters.

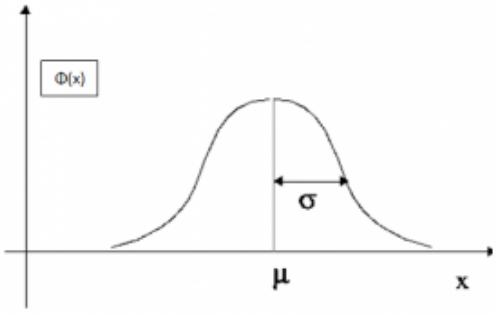
Architecture of RBF

RBF network is an artificial neural network with an input layer, a hidden layer, and an output layer. The Hidden layer of RBF consists of hidden neurons, and activation function of these neurons is a Gaussian function. Hidden layer generates a signal corresponding to an input vector in the input layer, and corresponding to this signal, network generates a response.



RBF Network Design and Training

To generate an output, neuron process the input signal through a function called activation function of the neuron. In RBF activation function of hidden neuron is ϕ(X) i.e. for an input vector X output produced by the hidden neuron will be ϕ(X).



$$\phi(x) = e^{-\frac{(x-\mu)^2}{\sigma^2}}$$

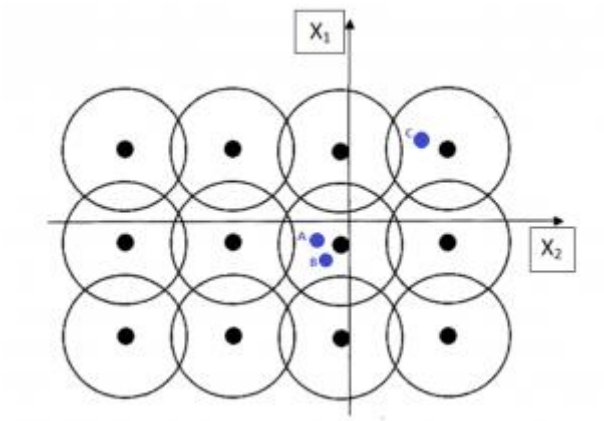
Above figure represents a Gaussian neural activation function for 1-D input x with center (mean) μ . Here ϕ(x) represents the output of Gaussian node for given value of x. It can be clearly

seen that the signal strength decreases as the input (X) move away from the center. The range of the Gaussian function is determined by ?, and output beyond the range is considered to be negligible.

The basic idea of this model is that the entire feature vector space is partitioned by Gaussian neural nodes, where each node generates a signal corresponding to an input vector, and strength of the signal produced by each neuron depends on the distance between its center and the input vector. Also for inputs lying closer in Euclidian vector space, the output signals that are generated must be similar.

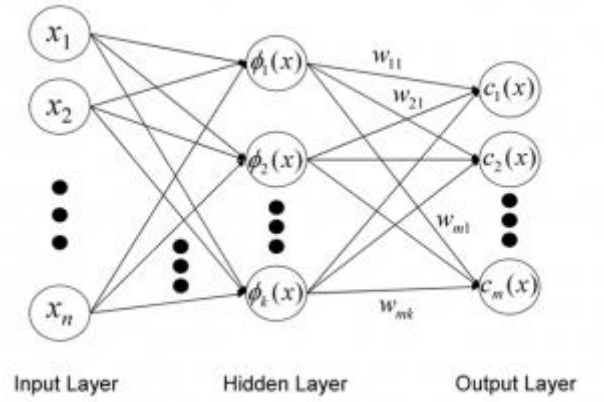
$$\phi(X) = e^{\frac{-||X-\mu||^2}{\sigma^2}}$$

Here, μ is center of the neuron and ϕ(X) is response of the neuron corresponding to input X.



In above figure circles represent Gaussian neural nodes and boundary of circles represents the range of the corresponding nodes also known as the receptive field of neurons. Here 2-D vector space is partitioned by 12 Gaussian nodes. Every input vector activates the collective system of neurons to some extent and the combination of these activations enables RBF to decide how to respond. Above configuration of neurons will generate similar output signals for input vectors A and B whereas for C output generated will be quite different.

In RBF architecture, weights connecting input vector to hidden neurons represents the center of the corresponding neuron. These weights are predetermined in such a way that entire space is covered by the receptive field of these neurons, whereas values of weights connecting hidden neuron to output neurons are determined to train the network.



It's appropriate if vectors lying close in the Euclidian space falls under the receptive field of the same neuron; therefore centers of hidden neurons are determined using K-Means clustering.

k Means Clustering algorithm:

- Choose the number of cluster centers "K".
- Randomly choose K points from the dataset and set them as K centroids of the data.
- For all the points in the dataset, determine the centroid closest to it.
- For all centroids, calculate the average of all the points lying closest to the same centroid.

- Change the value of all the centroids to corresponding averages calculated in (4).
- Go to (3) until convergence.

The range of receptive fields is chosen such that entire domain of input vector is covered by the receptive field of the neurons. So, the value of sigma is chosen according to maximum distance "d" between two hidden neurons.

$$\sigma = \frac{d}{\sqrt{2M}}$$

Where d is the maximum distance between two hidden neurons and M is the number of hidden neurons.

Mathematical Development

Let g_{ij} be an element of matrix G representing the output of j^{th} neuron for i^{th} input vector and W_{ij} be an element of matrix W representing weight connecting i^{th} output neuron to j^{th} hidden neuron. In RBF, the activation function of output neuron is linear i.e. "g(z)= z" where z is the weighted summation of signals from hidden layer. Multiplying i^{th} row of G with j^{th} columns of W does the weighted summation of signals from the hidden layer which is equal to signal produced by j^{th} output neuron.

GW= T

Where T is a column vector and i^{th} row contains the target value (actual desired output) of i^{th} training vector.

From above equation, by method of pseudo inverse

$$W = (G^T G)^{-1} G^T T$$

where G^T is the transpose of matrix G.

Algorithm

- Define the number of hidden neurons "K".
- set the positions of RBF centers using K-means clustering algorithm.
- Calculate σ using equation (2)
- Calculate actions of RBF node using equation (1)
- Train the output using equation (3)

RBF v/s MLP

MLPs are advantageous over RBFs when the underlying characteristic feature of data is embedded deeply inside very high dimensional data sets. For example, in image recognition, features depicting the key information about the image is hidden inside tens of thousands of pixel. For such training examples, the redundant features are filtered as the information progress through the stack of hidden layers in MLPs, and as a result, better performance is achieved.

Having only one hidden layer RBFs have much faster convergence rate as compared to MLP. For low dimensional data where deep feature extraction is not required and results are directly correlated with the component of input vectors, then RBF network is usually preferred over MLP. RBFs are universal approximators, and unlike most machine learning models RBF is a robust learning model.

APPROXIMATION PROPERTIES OF RBF

Radial-basis function networks exhibit good approximation properties of their own, paralleling those of multilayer perceptrons.

Universal Approximation Theorem

Let $G: \mathbb{R}^{m_0} \to \mathbb{R}$ be an integrable bounded function such that **G** is continuous and

$$\int_{\mathbb{R}^{m_0}} G(\mathbf{x}) d\mathbf{x} \neq 0$$

Let $\mathcal{S}_G$ denote the family of RBF networks consisting of functions $F: \mathbb{R}^{m_0} \to \mathbb{R}$ represented by

$$F(\mathbf{x}) = \sum_{i=1}^{m_1} w_i G\left(\frac{\mathbf{x} - \mathbf{t}_i}{\sigma}\right)$$

Where $\sigma > 0, w_i \in \mathbb{R}$ and $\mathbf{t}_i \in \mathbb{R}^{m_0}$. for $i = 1, 2, ..., m_1$.
We may then state the universal approximation theorem for RBF networks:

For any continuous input-output mapping function f(x) there is an RBF network with a setof centers $\{\mathbf{t}_i\}_{i=1}^{m_1}$ and a common width $\sigma > 0$ such that the input-output mapping functionF(x) realized by the RBF network is close to f(x) in the L_P norm, $p \in [1, \infty]$.

Curse of Dimensionality

In addition to the universal approximation property of RBF networks, there is the issue of the rate of approximation attainable by these networks that must be considered. we recall that the intrinsic complexity of a class of approximating functions increases exponentially in the ratio m_o/s, where mo is the input dimensionality (i.e., dimension of the input space) and S is a smoothness index measuring the number of constraints imposed on an approximating function in that particular class. Bellman's curse of dimensionality tells us that, irrespective of the approximation technique employed, if the smoothness index S is maintained constant, the number of parameters needed for the approximating function to attain a prescribed degree of accuracy increases exponentially with the input dimensionality mo. The only way that we can achieve a rate of convergence independent of the input dimensionality mv, and therefore be immune to the curse of dimensionality, is for the smoothness index S to increase with the number of parameters in the approximating function so as to compensate for the increase in complexity.

Function Space	Norm	Approximation Technique
$\int_{\mathbb{R}^{m_0}} \|\mathbf{s}\| \tilde{F}(\mathbf{s}) d\mathbf{s} < \infty$	$L_2(\Omega)$	(a) multilayer perceptrons
where $\tilde{F}(\mathbf{s})$ is the *multidimensional* Fourier transform of the approximating function $F(\mathbf{x})$		$F(\mathbf{x}) = \sum_{i=1}^{m_1} a_i \varphi(\mathbf{w}_i^T \mathbf{x} + b_i)$ where $\varphi(.)$ is the sigmoid activation function
Sobolev space of functions whose derivatives up to order $2m > m_0$ are integrable	$L_2(\mathbb{R}^2)$	(b) RBF networks: $F(\mathbf{x}) = \sum_{i=1}^{m_1} a_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{t}_i\|^2}{2\sigma^2}\right)$

Two Approximation Techniques and Corresponding Function Spaces with the Same Rate of convergence $O(1/\sqrt{m_1})$ where m_1 is the Size of the Hidden Space.

Naturally, the constraints imposed on these two approximating techniques are different, reflecting the different paths followed in their formulations. In the case of RBF networks, the result holds in the Sobolev space of functions whose derivations up to order $2m > m_0$ are integrable. In other words, the number of derivatives of the approximating function that are integrable is required to increase with the input dimensionality m_0 in order to make the rate of convergence independent of m_0