Plan of Class 4

- Radial Basis Functions with moving centers;
- Multilayer Perceptrons;
- Projection Pursuit Regression and ridge functions approximation;
- Radial Basis Functions and dimensionality reduction;
- From Additive Splines to ridge functions approximation;
- Dual kernels: global versus local approximation;

Regularization approach

A smooth function that approximates the data set $D = \{(\mathbf{x}_i, y_i) \in R^d \times R\}_{i=1}^N$ can be found minimizing the functional:

$$H[f] = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 + \lambda \int_{R^d} d\mathbf{s} \, \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}$$

where $\tilde{G}$ is a positive, even, function, decreasing to zero at infinity, and λ a small, positive number.

Main result

(Duchon, 1977; Meinguet, 1979; Wahba, 1977; Madych and Nelson, 1990; Poggio and Girosi, 1989; Girosi, 1992)

The function that minimizes the functional

$$H[f] = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 + \lambda \int_{R^d} d\mathbf{s} \; \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}$$

has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) + \sum_{\alpha=1}^{k} d_\alpha \gamma_\alpha(\mathbf{x})$$

where G is conditionally positive definite of order m, $\tilde{G}$ is the Fourier transform of G and $\{\gamma_\alpha\}_{\alpha=1}^{k}$ is a basis in the space of polynomials of degree $m-1$.

Computation of the coefficients

The coefficients are found by solving the linear system:

$$(G + \lambda I)\mathbf{c} + \Gamma^T \mathbf{d} = \mathbf{y}$$

$$\Gamma \mathbf{c} = 0$$

where I is the identity matrix, and we have defined

$$(\mathbf{y})_i = y_i \ , \quad (\mathbf{c})_i = c_i \ , \quad (\mathbf{d})_i = d_i$$

$$(G)_{ij} = G(\mathbf{x}_i - \mathbf{x}_j) \ , \quad (\Gamma)_{\alpha i} = \gamma_\alpha(\mathbf{x}_i)$$

Radial Basis Functions

If the function $\tilde{G}$ is radial ($\tilde{G} = \tilde{G}(\|\mathbf{s}\|)$ the smoothness functional ϕ is rotationally invariant, that is

$$\phi[f(\mathbf{x})] = \phi[f(R\mathbf{x})] \quad \forall R \in O(d)$$

where $O(d)$ is the group of rotations in d dimensions.

The regularization solution becomes the so called Radial Basis Functions technique:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x})$$

Radial Basis Functions

Define $r = \|\mathbf{x}\|$

$G(\mathbf{x}) = e^{-r^2}$	gaussian
$G(\mathbf{x}) = \sqrt{r^2 + c^2}$	multiquadric
$G(\mathbf{x}) = \frac{1}{\sqrt{c^2+r^2}}$	inverse multiquadric
$G(\mathbf{x}) = r^{2n+1}$	multivariate splines
$G(\mathbf{x}) = r^{2n} \ln r$	multivariate splines

Some good properties of RBF

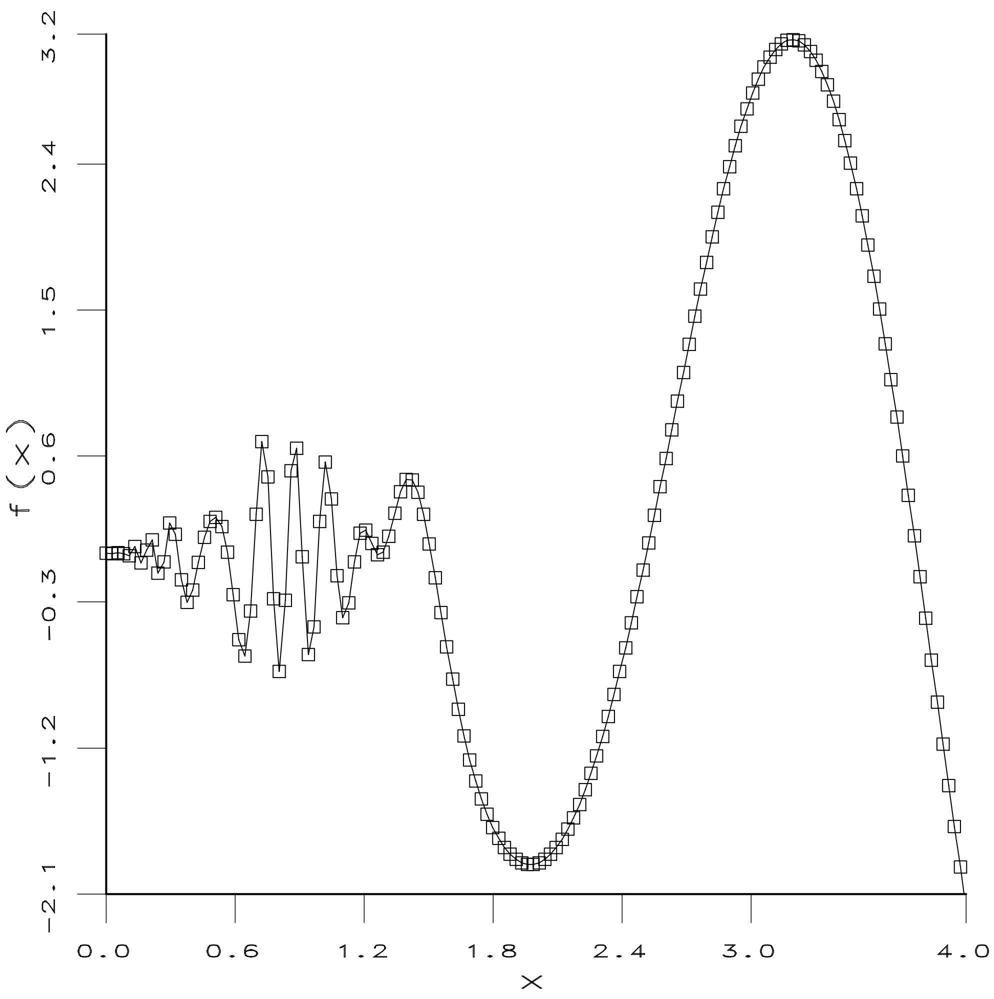
- Very well motivated in the framework of regularization theory;
- The solution is unique and equivalent to solve a linear system;
- Amount of smoothness is tunable (with λ);
- Radial Basis Functions are *universal approximators*;
- Huge amount of literature on this subject;
- (Interpretation in terms of "neural networks");
- Biologically plausible;
- Simple interpretation in terms of "smooth look-up table";
- Similar to other non-parametric techniques, such as nearest neighbor and kernel regression;

Some not-so-good properties of RBF

- Computationally expensive ($O(N^3)$ where $N =$number of data);

- Linear system often badly ill-conditioned;

- The same amount of smoothness is imposed on different regions of the domain;



This function has different smoothness properties in different regions of its domain

Least Squares Regularization Networks

We look for an *approximation* to the regularization solution:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i)$$

$$\Downarrow$$

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{x} - \mathbf{t}_\alpha)$$

where $n << N$ and the vectors $\mathbf{t}_\alpha$ are called *centers*.

(Broomhead and Lowe, 1988; Moody and Darken, 1989; Poggio and Girosi, 1989)

Least Squares Regularization Networks

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{x} - \mathbf{t}_\alpha)$$

Suppose the centers $\mathbf{t}_\alpha$ have been fixed.

How do we find the coefficients c_α?

$$\Downarrow$$

Least Squares

Least Squares Regularization Networks

Define

$$E(c_1, \ldots, c_n) = \sum_{i=1}^{N} (y_i - f^*(\mathbf{x}_i))^2$$

The least squares criterion is

$$\min_{c_\alpha} \; E(c_1, \ldots, c_n)$$

The problem is convex and quadratic in the c_α, and the solution satisfies:

$$\frac{\partial E}{\partial c_\alpha} = 0$$

Least Squares Regularization Networks

$$\frac{\partial E}{\partial c_\alpha} = 0$$

$$\Downarrow$$

$$G^T G\ \mathbf{c} = G^T\ \mathbf{y}$$

where we have defined

$$(\mathbf{y})_i = y_i\ , \quad (\mathbf{c})_\alpha = c_\alpha\ , (G)_{i\alpha} = G(\mathbf{x}_i - \mathbf{t}_\alpha)$$

Therefore $\mathbf{c} = G^+\mathbf{y}$, where

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

is the *pseudoinverse* of G.

Least Squares Regularization Networks

Given the centers $\mathbf{t}_\alpha$ we know how to find the c_α.

How do we choose the $\mathbf{t}_\alpha$?

1. a subset of the examples;

2. by a clustering algorithm (k-means, for examples);

3. by least squares ("moving centers");

Centers as a subset of the examples

Fair technique. The subset is a random subset, which should reflect the distribution of the data.

Not many theoretical results available.

Main problem: how many centers?

Main answer: we don't know. Cross validation techniques seem a reasonable choice.

Finding the centers by clustering

Very common. However it makes sense only if the input data points are clustered.

No theoretical results.

Not clear that it is a good idea, especially for pattern classification cases.

Moving centers

Define

$$E(c_1, \ldots, c_n, \mathbf{t}_1, \ldots, \mathbf{t}_n) = \sum_{i=1}^{N} (y_i - f^*(\mathbf{x}_i))^2$$

The least squares criterion is

$$\min_{c_\alpha, \mathbf{t}_\alpha} \; E(c_1, \ldots, c_n, \mathbf{t}_1, \ldots, \mathbf{t}_n)$$

The problem is not convex and quadratic any-more: expect multiple local minima.

Moving centers

:-) Very flexible, in principle very powerful;

:-) Some theoretical understanding;

:-(Very expensive computationally due to the local minima problem;

:-(Centers sometimes move in “weird” ways;

Some new approximation schemes

Radial Basis Functions with moving centers is a particular case of a function approximation technique of the form:

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha H(\mathbf{x}, \mathbf{p}_\alpha)$$

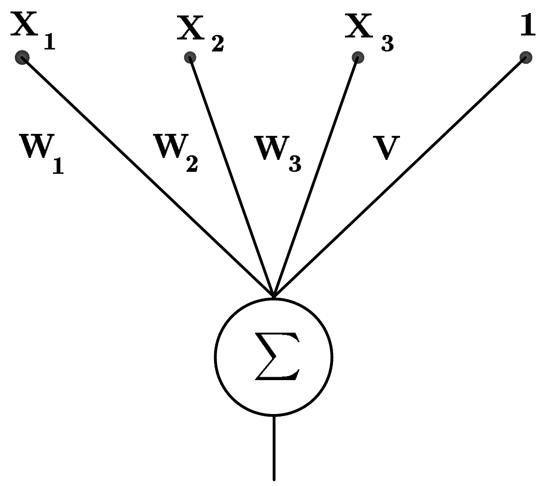
where $\{\mathbf{p}_\alpha\}_{\alpha=1}^{n}$ is a set of parameters, which can be estimated by least squares techniques.

Radial Basis Functions corresponds to the choice

$$H(\mathbf{x}, \mathbf{p}_\alpha) = G(\|\mathbf{x} - \mathbf{p}_\alpha\|)$$

The simplest approximation technique is *linear*:

$$f^*(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + v$$



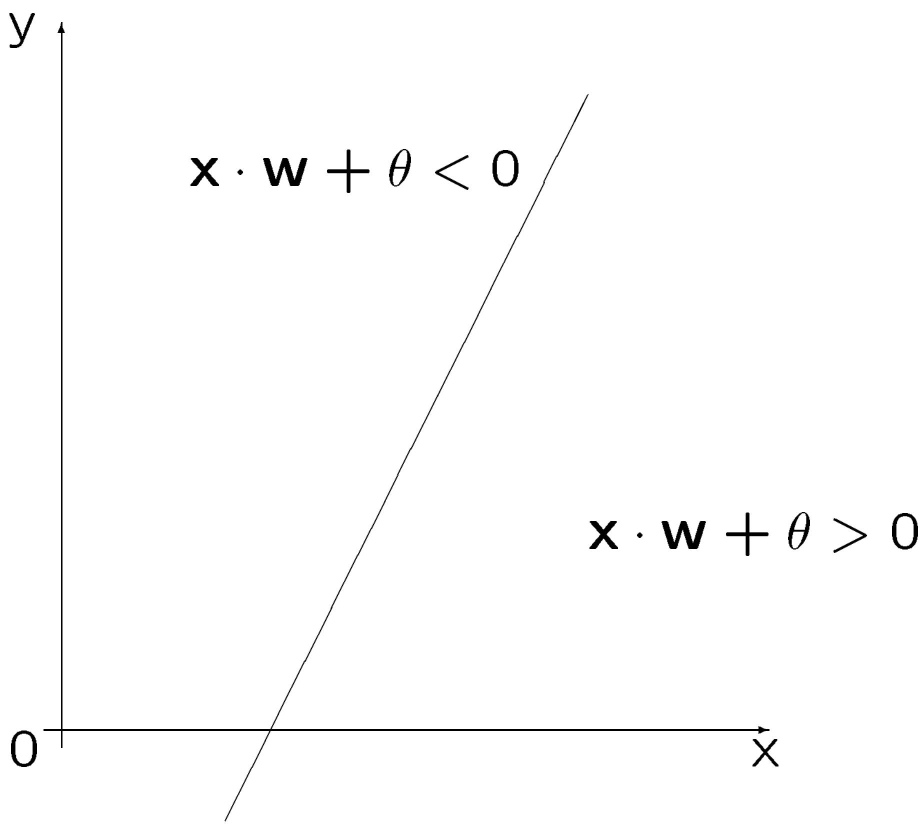
$$\mathbf{f^* = W_1 X_1 + W_2 X_2 + W_3 X_3 + V}$$

Perceptrons

For binary outputs (classification problems) Perceptrons are more appropriate:

$$f^*(\mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + \theta)$$

σ is a step function

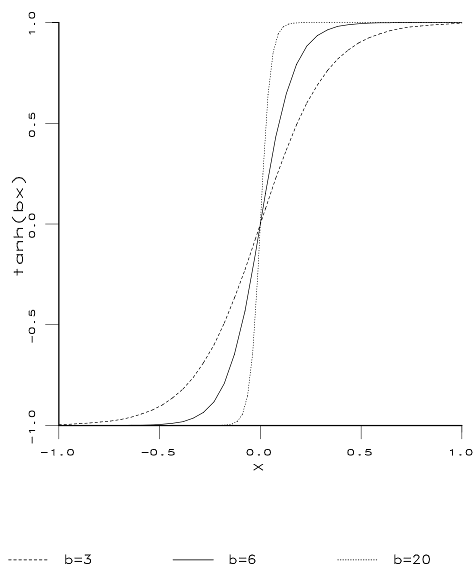

y
$\mathbf{x} \cdot \mathbf{w} + \theta < 0$
$\mathbf{x} \cdot \mathbf{w} + \theta > 0$
0
x

If the output is real, but bounded, the function σ may be a sigmoid

Examples of sigmoids

$$f(x) = \frac{1}{1 + e^{-\beta x}}$$

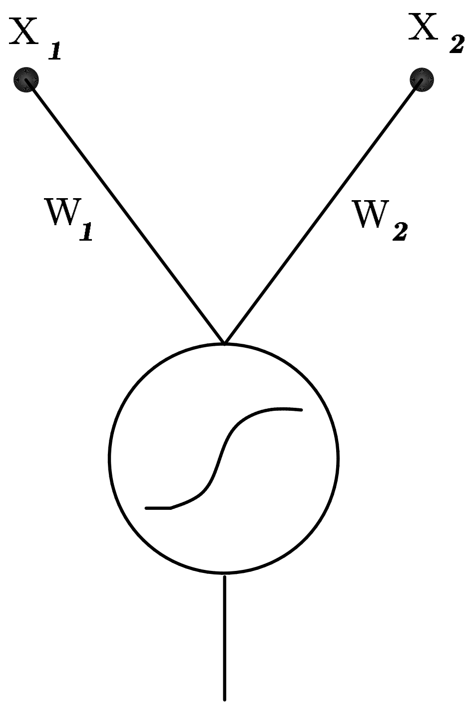
$$f(x) = \tanh(\beta x)$$



The sigmoidal function tanh(bx) for different values of b.

Notice that the function approaches a step function as b increases.

Network representation



$$Y = \sigma(W_1 X_1 + W_2 X_2)$$

$$\sigma(X) = \frac{1}{1 + e^{-X}}$$

Multilayer Perceptrons

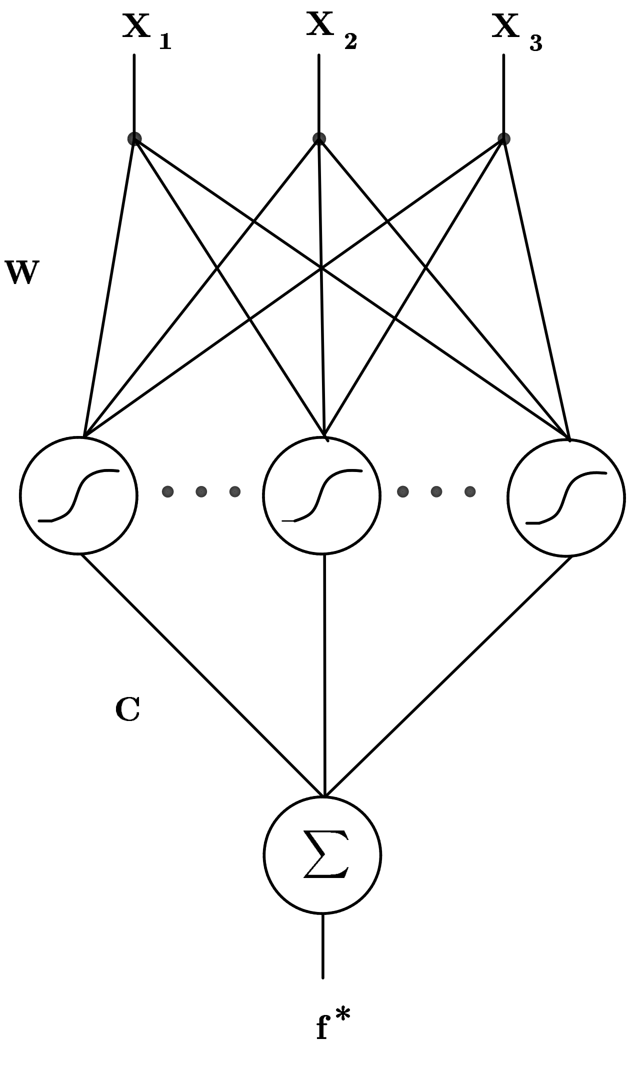
A linear superposition of simple Perceptrons:

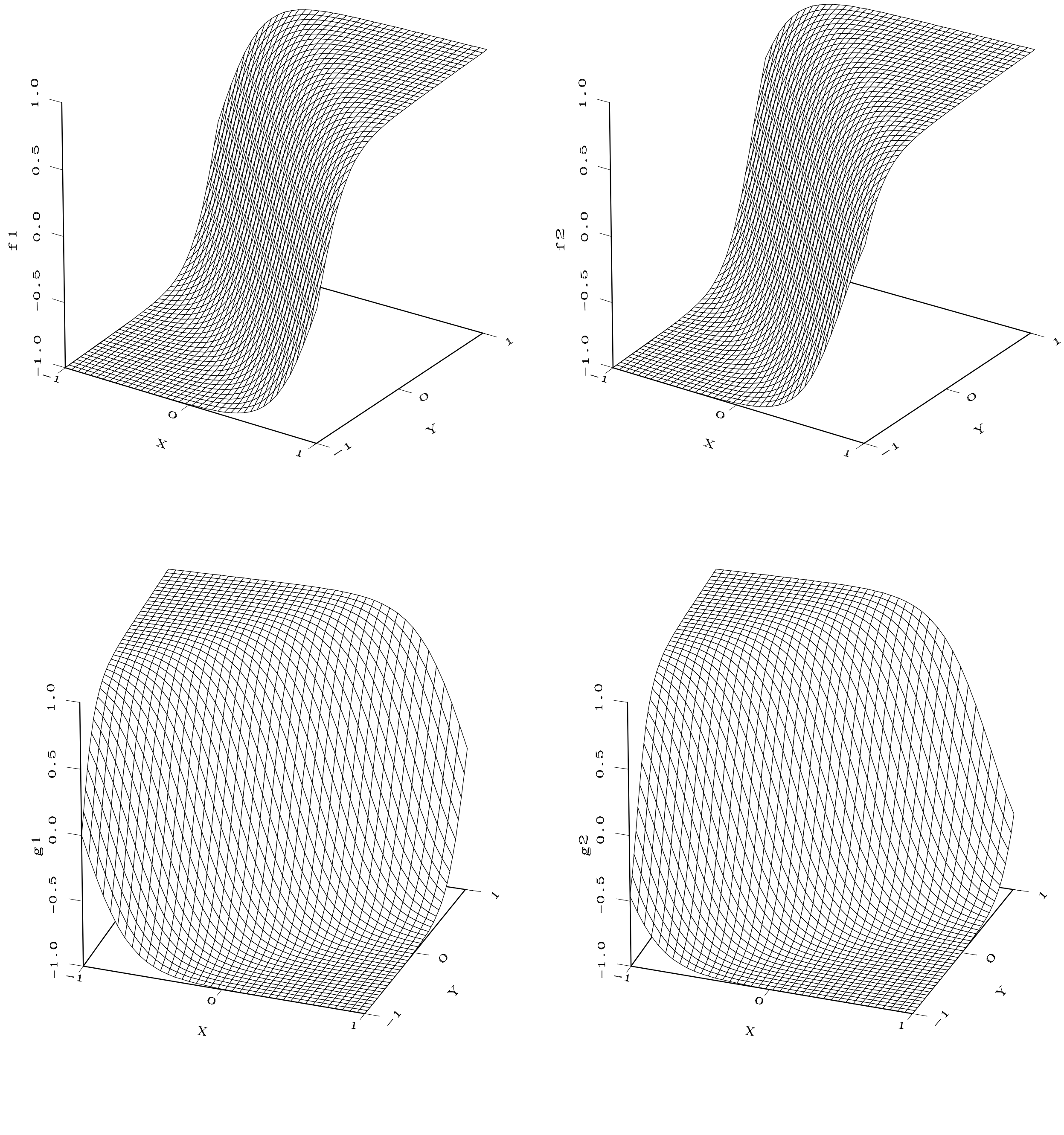
$$f^*(\mathbf{x}) = \sum_{i=1}^{n} c_i \sigma(\mathbf{x} \cdot \mathbf{w}_i + \theta_i)$$

or more complex superpositions:

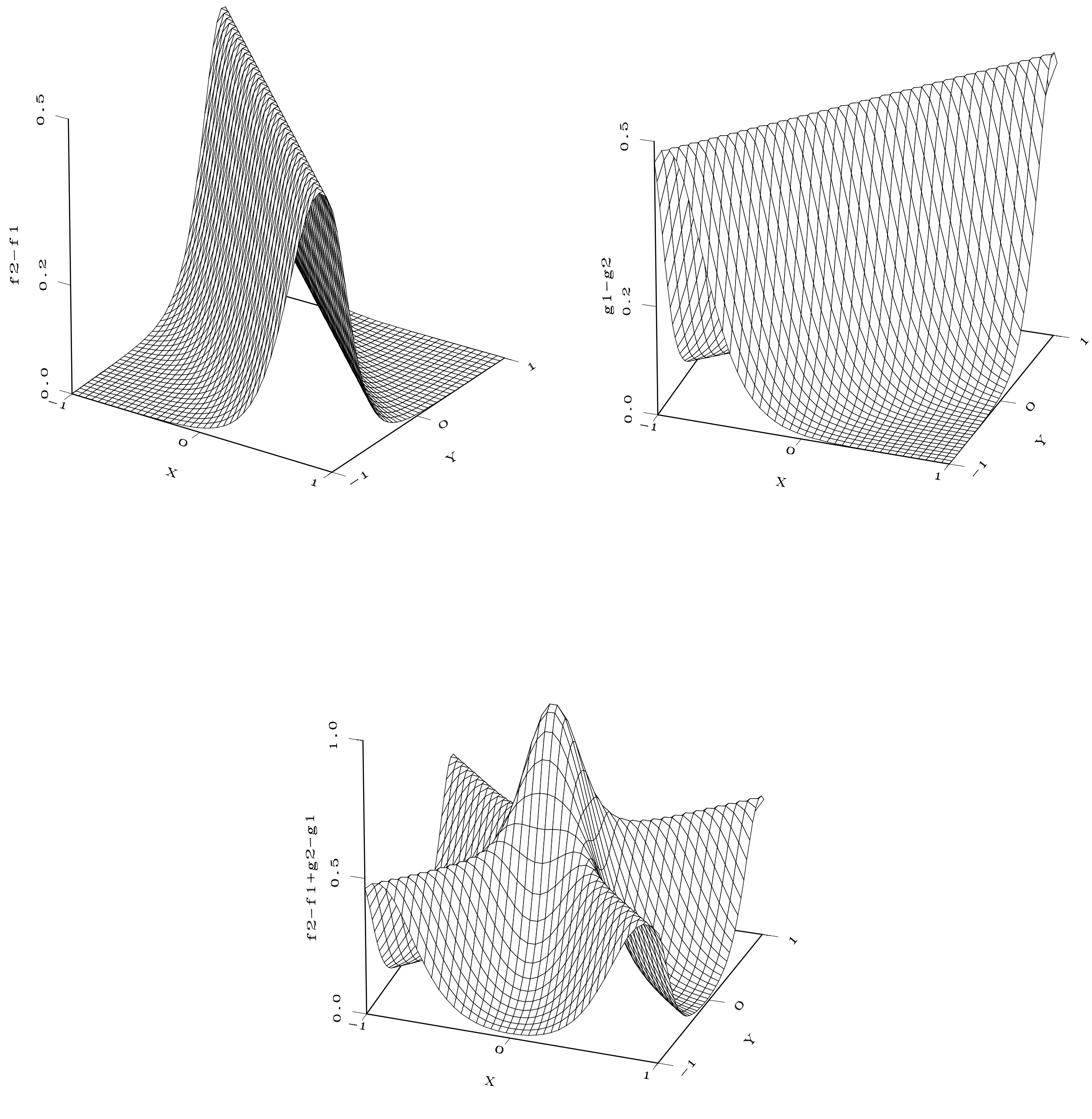
$$f^*(\mathbf{x}) = \sum_{j=1}^{k} a_j \sigma(\sum_{i=1}^{n} c_i^j \sigma(\mathbf{x} \cdot \mathbf{w}_i + \theta_i))$$

One hidden layer Perceptron





$f1$, $f2$, $g1$ and $g2$ are the output of different sigmoid units.



Output of different units combined together in different ways.

Theorem: (Weierstrass) The set $P_{[a,b]}$ of all polynomials is dense in $C[a,b]$. In other words, given an $h \in C[a,b]$ and $\epsilon > 0$, there is a polynomial p for which:

$$|p(x) - h(x)| < \epsilon \quad \text{for all} \quad x \in [a,b] \ .$$

Theorem: (Cybenko, 1989) Let σ be any sigmoidal function and I_d the d-dimensional cube $[0,1]^d$. Then finite sums of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} c_i \sigma(\mathbf{x} \cdot \mathbf{w}_i + \theta_i)$$

are dense in $C[I_d]$. In other words, given an $h \in C[I_d]$ and $\epsilon > 0$, there is a sum, $f(\mathbf{x})$, of the above form, for which:

$$|f(\mathbf{x}) - h(\mathbf{x})| < \epsilon \quad \text{for all} \quad x \in I_d \ .$$

Theorem: Let G a Radial Basis Function function and I_d the d-dimensional cube $[0,1]^d$. Then finite sums of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} c_i G(\mathbf{x} - \mathbf{t}_i)$$

are dense in $C[I_d]$. In other words, given an $h \in C[I_d]$ and $\epsilon > 0$, there is a sum, $f(\mathbf{x})$, of the above form, for which:

$$|f(\mathbf{x}) - h(\mathbf{x})| < \epsilon \quad \text{for all} \quad x \in I_d \ .$$

Results of this type are very common in approximation theory, and can hold under very weak assumptions.

Density in the space of continuous functions can be considered as a *necessary* condition that every approximation scheme should satisfy.

However, results of this type do not say anything about interpolation/approximation of sets of sparse data points.

MLP Training

The parameter of a MLP are found by solving the following minimization problem:

$$\min_{c_j, \mathbf{w}_j, \theta_j} = \sum_{i=1}^{l} \left(y_i - \sum_{j=1}^{n} c_j \sigma(\mathbf{x}_i \cdot \mathbf{w}_j + \theta_j) \right)^2$$

Very difficult: many parameters and multiple local minima.

Gradient descent

In order to minimize the function $F(\mathbf{w})$ one can use the following iterative procedure:

$$\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} - \epsilon_n \nabla F(\mathbf{w}^{(n)})$$

- choice of ϵ_n may be difficult;
- can be very slow;
- converges to local minimum: restart procedure is needed;

Stochastic Gradient Descent

Consider the sequence

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{\epsilon}{n}(\nabla F(\mathbf{w}_n) + \mathbf{s}_n)$$

where $\{\mathbf{s}_n\}$ is a sequence of random vectors with zero mean. Under certain assumptions this sequence converges to a local minimum of F.

Stochastic Gradient Descent

Consider the case in which $F(\mathbf{w}) = \frac{1}{l}\sum_{i=1}^{l} f_i(\mathbf{w})$ and choose $\mathbf{s}_n$ as

$$\mathbf{s}_n = \nabla f_{\xi_n} - \nabla F \ .$$

where $\{\xi_n\}$ is a sequence of random variables that take values in $\{1, \ldots, l\}$ with uniform probability distribution. The stochastic gradient descent sequence becomes:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{\epsilon}{n}\nabla f_{\xi_n}(\mathbf{w}_n)$$

This is called *on-line learning*.

Does this sequence converge to a local minimum of F?

Stochastic gradient descent (backpropagation)

Theorem (Girosi and Marroquin, 1993)

Let $F(y) : \mathbf{R}^k \mapsto \mathbf{R}$ be of the form:

$$F(y) = f_0(y) + \frac{1}{l} \sum_{i=1}^{l} f_i(y)$$

where f_i are differentiable functions whose gradient is bounded and satisfies the following Lipschitz condition:

$$\|\nabla f_i(y) - \nabla f_i(y')\| \leq M\|y - y'\| \; i = 0, \ldots, l$$

for some positive number M. Let $\{y_n\}$ be the sequence

$$y_{n+1} = y_n - a_n(\nabla f_0(y_n) + \nabla f_{\xi_n}(y_n)) + a_n \eta_n \tag{1}$$

where $\{\xi_n\}$ is a sequence of random variables that take values in $\{1, \ldots, l\}$ with uniform probability distribution, $\{a_n\}$ is a sequence of scalars satisfying:

$$\sum_{i=1}^{\infty} a_n = \infty \quad \text{and} \quad \sum_{i=1}^{\infty} a_n^2 < \infty$$

and $\{\eta_n\}$ is a sequence of random variables that is bounded and converges to zero with probability one. Assume that $\{y_n\}$ is bounded and that S is a locally asymptotically stable point of the ordinary differential equation:

$$\dot{y} = -\nabla F \tag{2}$$

with domain of attraction A_S. Then, if $y_n \in G$ for all n, for some compact domain $G \subset A_S$, $\{y_n\}$ converges to S with probability one.

Stochastic gradient descent (backpropagation)

Theorem (Girosi and Marroquin, 1993), easy version.

Let $F(\mathbf{w}) : \mathbf{R}^k \mapsto \mathbf{R}$ be of the form:

$$F(\mathbf{w}) = \frac{1}{l}\sum_{i=1}^{l} f_i(\mathbf{w})$$

Let us consider the following sequence:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{\epsilon}{n}\nabla f_{\xi_n}(\mathbf{w}_n)$$

where $\{\xi_n\}$ is a sequence of random variables that take values in $\{1,\ldots,N\}$ with uniform probability distribution.

This sequence converge to the same fixed point of the sequence:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \epsilon\nabla F(\mathbf{w}_n)$$

that is a local minimum of F.

Stochastic gradient descent (backpropagation)

The reason for which the previous theorem holds is that

$$E[\nabla f_{\xi_n}] = \frac{1}{l}\sum_{i=1}^{l} \nabla f_i = \nabla F$$

where we used the assumption that $P(\xi_n) = \frac{1}{l}$

Multilayer Perceptrons

:-) Approximation schemes of this type have been successful in a number of cases;

:-(Interpretation of the approximation technique is not immediate;

:-) Multilayer Perceptrons are "universal approximator";

:-) Some theoretical results available;

:-(Computationally expensive, local minima;

:-— Motivation of this technique?

Multilayer perceptrons are particular cases of the more general **ridge function approximation** techniques:

$$f(\mathbf{x}) = \sum_{\mu=1}^{n} h_\mu(\mathbf{x} \cdot \mathbf{w}_\mu)$$

in which we have chosen

$$h_\mu(x) = c_\mu h(x + \theta_\mu)$$

for some fixed function h (sigmoid).

Projection Pursuit Regression (Friedman and Stuetzle, 1981; Huber, 1986)

Look for an approximating function of the form

$$f(\mathbf{x}) = \sum_{\mu=1}^{n} h_\mu(\mathbf{x} \cdot \mathbf{w}_\mu)$$

where $\mathbf{w}_\mu$ are unit vectors and h_μ are functions to be fitted to the data (often cubic splines, trigonometric polynomials or and Radial Basis Functions).

The underlying idea is that all the information lies in few, one dimensional projections of the data.

The unit vectors $\mathbf{w}_\mu$ are the "interesting projections" that have to be pursued.

PPR algorithm

- Assume that the first $\nu - 1$ terms of the expansion have been determined, and define the residuals

$$r_i^{(\nu-1)} = y_i - \sum_{\mu=1}^{\nu-1} h_\mu(\mathbf{x}_i \cdot \mathbf{w}_\mu)$$

- Find the ν-th vector $\mathbf{w}_\nu$ and the ν-th function h_ν that solve

$$\min_{\mathbf{w}_\nu, h_\nu} \sum_{i=1}^{N} (r_i^{(\nu-1)} - h_\nu(\mathbf{x}_i \cdot \mathbf{w}_\nu))^2$$

- Go back to the first step and iterate

Motivation of PPR

Cluster analysis of high-dimensional set of data points D: Diaconis and Freedman (1984) show that *for most high-dimensional clouds, most one-dimensional projections are approximately normal*.

In cluster analysis normal means "uninteresting".

Therefore there are *few interesting projections*, that can be found maximizing a "projection index" that measures deviation from normality.

Extensions of Radial Basis Functions

- Different variables can have different scales: $f(x, y) = y^2 \sin(100x)$;
- Different variables could have different units of measure $f = f(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})$;
- Not all the variables are independent or relevant: $f(x, y, z, t) = g(x, y, z(x, y))$;
- Only some linear combinations of the variables are relevant: $f(x, y, z) = \sin(x + y + z)$;

Extensions of regularization theory

A priori knowledge:

- the relevant variables are linear combination of the original ones:

$$\mathbf{z} = \mathbf{W}\mathbf{x}$$

  for some (possibly rectangular) matrix $\mathbf{W}$;

- $f(\mathbf{x}) = g(\mathbf{W}\mathbf{x}) = g(\mathbf{z})$ and the function g is smooth;

The regularization functional is now

$$H[g] = \sum_{i=1}^{N} (y_i - g(\mathbf{z}_i))^2 + \lambda\phi[g]$$

where $\mathbf{z}_i = \mathbf{W}\mathbf{x}_i$.

Extensions of regularization theory (continue)

The solution is

$$g(\mathbf{z}) = \sum_{i=1}^{N} c_i G(\mathbf{z} - \mathbf{z}_i) \ .$$

Therefore the solution for f is:

$$f(\mathbf{x}) = g(\mathbf{W}\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{W}\mathbf{x} - \mathbf{W}\mathbf{x}_i)$$

If the matrix $\mathbf{W}$ were known, the coefficients could be computed as in the radial case:

$$(G + \lambda I)\mathbf{c} = \mathbf{y}$$

where

$$(\mathbf{y})_i = y_i \ , \quad (\mathbf{c})_i = c_i \ , (G)_{ij} = G(\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j)$$

and the same criticisms of the Regularization Networks technique apply, leading to *Generalized Regularization Networks*:

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{W}\mathbf{x} - \mathbf{W}\mathbf{t}_\alpha)$$

Since $\mathbf{W}$ is not known, it could be found by *least squares*. Define

$$E(c_1, \cdots, c_n, \mathbf{W}) = \sum_{i=1}^{N} (y_i - f^*(\mathbf{x}_i))^2$$

Then we can solve:

$$\min_{c_\alpha, \mathbf{W}} \; E(c_1, \cdots, c_n, \mathbf{W})$$

The problem is not convex and quadratic anymore: expect multiple local minima.

From RBF to HyperBF

When the basis function is radial the Generalized Regularization Networks becomes

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\|\mathbf{x} - \mathbf{t}_\alpha\|_{\mathbf{w}})$$

that is a *non radial basis function* technique

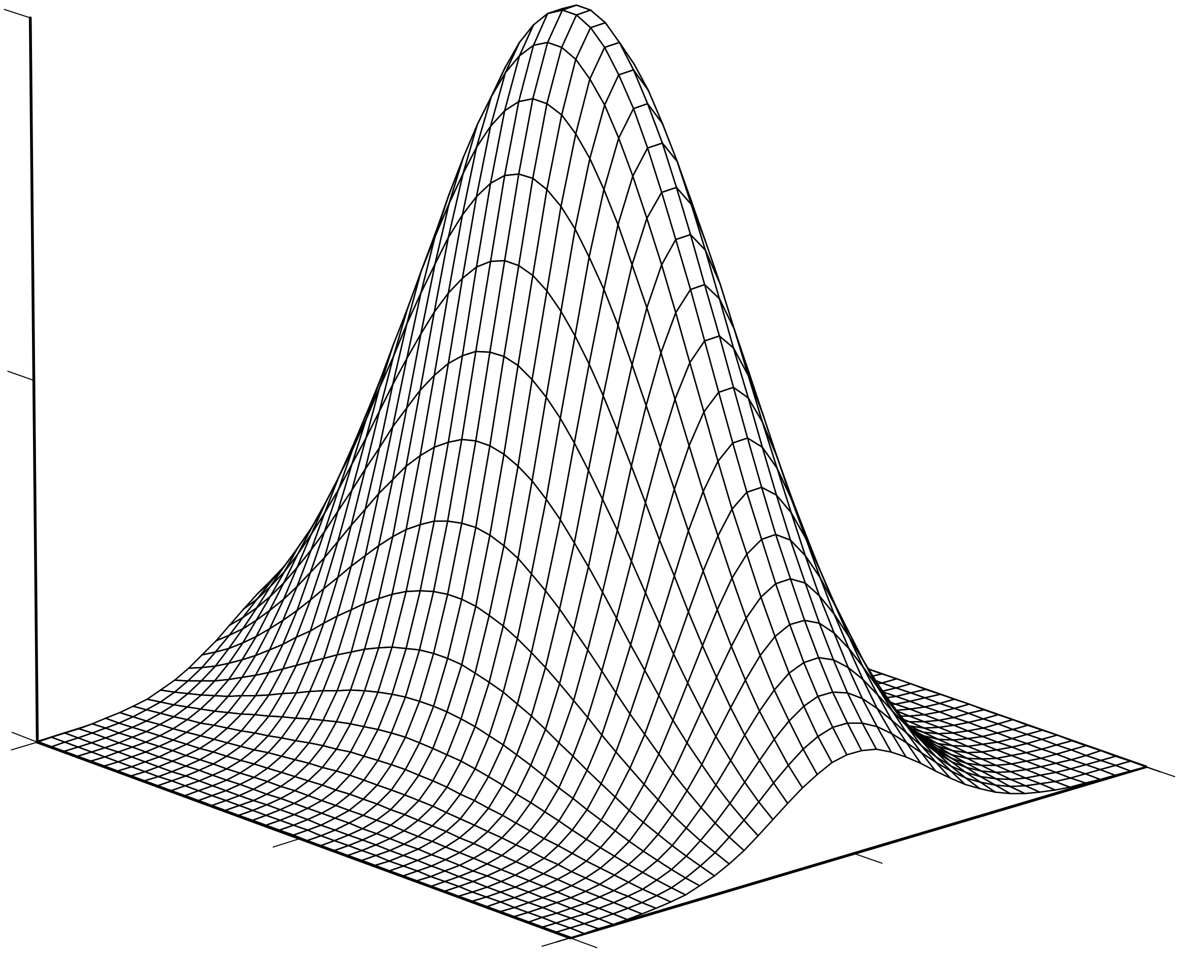
Least Squares

1. $\min_{c_\alpha} \; E(c_1, \ldots, c_n)$

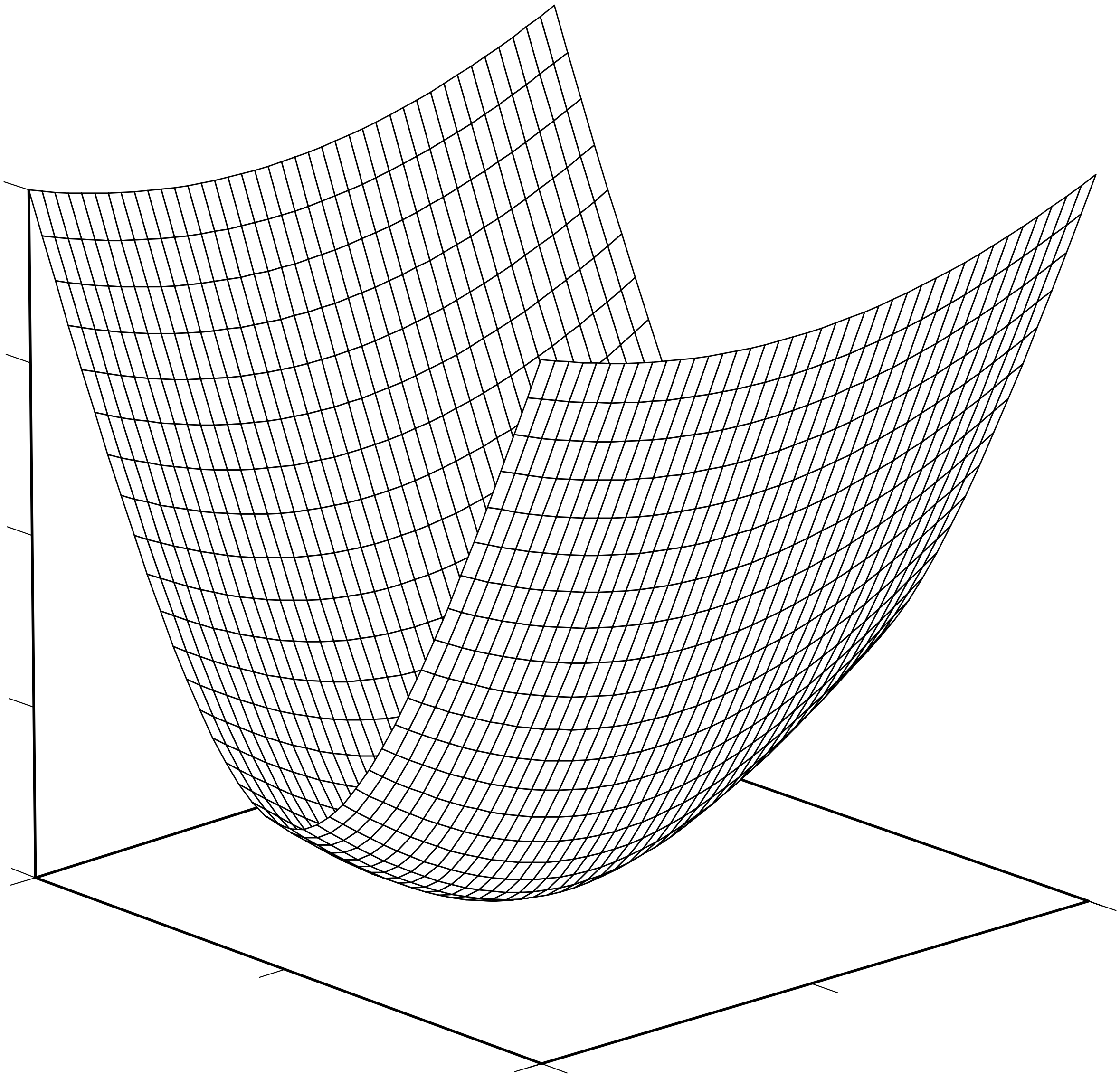
2. $\min_{c_\alpha, \mathbf{t}_\alpha} \; E(c_1, \ldots, c_n, \mathbf{t}_1, \ldots, \mathbf{t}_n)$

3. $\min_{c_\alpha, \mathbf{W}} \; E(c_1, \cdots, c_n, \mathbf{W})$

4. $\min_{c_\alpha, \mathbf{t}_\alpha, \mathbf{W}} E(c_1, \ldots, c_n, \mathbf{t}_1, \ldots, \mathbf{t}_n, \mathbf{W})$



A non radial gaussian function



A non radial multiquadric function

Additive models

An additive model has the form

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} f_\mu(x^\mu)$$

where

$$f_\mu(x^\mu) = \sum_{i=1}^{N} c_i^\mu G(x^\mu - x_i^\mu)$$

In other words

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} \sum_{i=1}^{N} c_i^\mu G(x^\mu - x_i^\mu)$$

Extensions of Additive Models

If we have less centers than examples we obtain:

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} f_\mu(x^\mu)$$

where

$$f_\mu(x^\mu) = \sum_{\alpha=1}^{N} c_\alpha^\mu G(x^\mu - t_\alpha^\mu)$$

In other words

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} \sum_{\alpha=1}^{N} c_\alpha^\mu G(x^\mu - t_\alpha^\mu)$$

Extensions of Additive Models

If we now allow for an arbitrary linear transformation of the inputs:

$$\mathbf{x} \rightarrow \mathbf{W}\mathbf{x}$$

where $\mathbf{W}$ is a $d' \times d$ matrix, we obtain:

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} \sum_{\alpha=1}^{n} c_{\alpha}^{\mu} G(\mathbf{x} \cdot \mathbf{w}_{\mu} - t_{\alpha}^{\mu})$$

where $\mathbf{w}_{\mu}$ is the μ-th row of the matrix $\mathbf{W}$,

Extensions of Additive Models

The expression

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} \sum_{\alpha=1}^{n} c_\alpha^\mu G(\mathbf{x} \cdot \mathbf{w}_\mu - t_\alpha^\mu)$$

can be written as

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} h_\mu(\mathbf{x} \cdot \mathbf{w}_\mu)$$

where

$$h_\mu(y) = \sum_{\alpha=1}^{n} c_\alpha^\mu G(y - t_\alpha^\mu)$$

This form of approximation is called **ridge approximation**

From the extension of additive models we can therefore justify an approximation technique of the form

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} \sum_{\alpha=1}^{n} c_\alpha^\mu G(\mathbf{x} \cdot \mathbf{w}_\mu - t_\alpha^\mu)$$

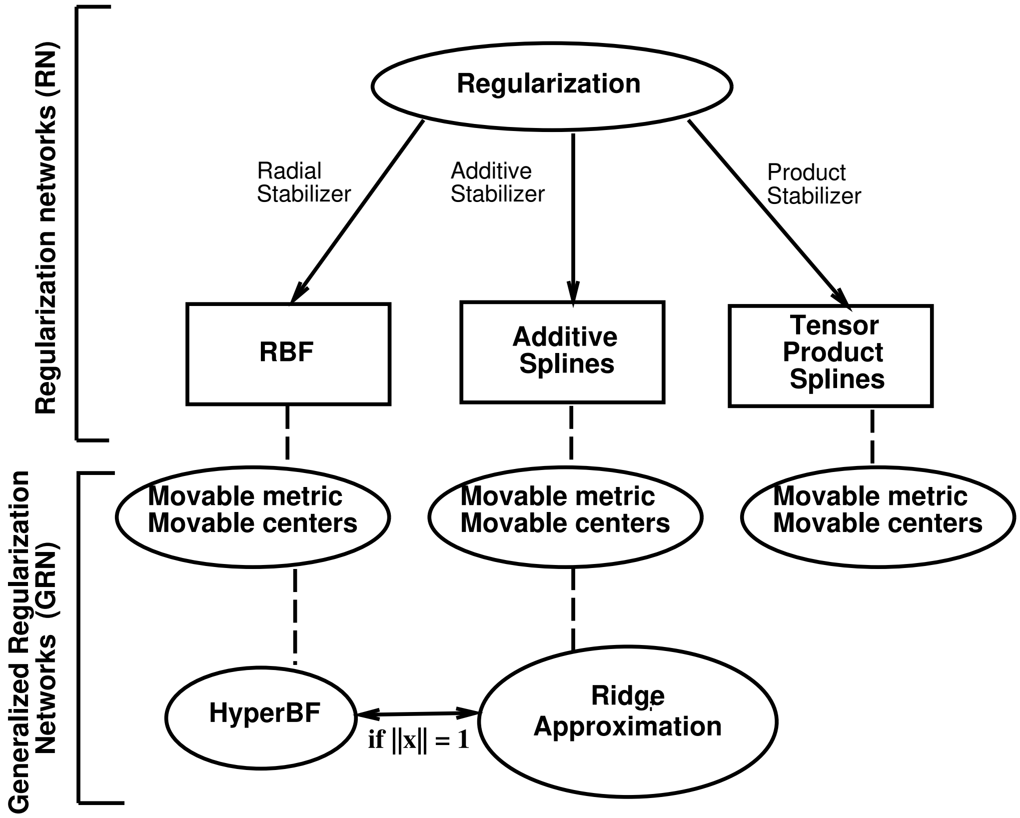
Particular case: $n = 1$ (one center). Then we derive the following technique:

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} c^\mu G(\mathbf{x} \cdot \mathbf{w}_\mu - t\mu)$$

which is a Multilayer Perceptron with a Radial Basis Functions G instead of the sigmoid function.

Notice that the sigmoid function is not a Radial Basis Functions

Regularization Networks



Memory based learning versus Regularization networks

- kernel regression: no complex global model of the world is assumed. Many simple local models instead.

- Regularization networks: fairly complex global model of the world

Are these two techniques related?

Least square Regularization networks

A model of the form

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{x} - \mathbf{t}_\alpha)$$

is assumed and the parameters c_α and $\mathbf{t}_\alpha$ are found by

$$\min_{c_\alpha, \mathbf{t}_\alpha} E[\{c_\alpha\}, \{\mathbf{t}_\alpha\}]$$

where

$$E[\{c_\alpha\}, \{\mathbf{t}_\alpha\}] = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2$$

Least square Regularization networks

The coefficients c_α and the centers $\mathbf{t}_\alpha$ have to satisfy the conditions:

$$\frac{\partial E}{\partial c_\alpha} = 0 \ , \qquad \frac{\partial E}{\partial \mathbf{t}_\alpha} = 0 \quad \alpha = 1, \ldots, n$$

The equation for the coefficients gives:

$$c_\alpha = \sum_{i=1}^{N} H_{\alpha i} y_i$$

where

$$H = (G^T G)^{-1} G^T \ , \qquad G_{i\alpha} = G(\mathbf{x}_i - \mathbf{t}_\alpha)$$

Substituting the expression for the coefficients in the regularization network we obtain

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} \sum_{i=1}^{N} H_{\alpha i} y_i G(\mathbf{x} - \mathbf{t}_\alpha) = \sum_{i=1}^{N} y_i \sum_{\alpha=1}^{n} H_{i\alpha}^T G(\mathbf{x} - \mathbf{t}_\alpha)$$

$$\boxed{f(\mathbf{x}) = \textstyle\sum_{i=1}^{N} y_i b_i(\mathbf{x})}$$

where we have defined

$$b_i(\mathbf{x}) = \sum_{\alpha=1}^{n} H_{i\alpha}^T G(\mathbf{x} - \mathbf{t}_\alpha)$$

The representation above is called the "dual representation" and the basis function $b_i(\mathbf{x})$ are called "dual kernels".

Memory based learning versus Regularization networks

$$\boxed{f(\mathbf{x}) = \sum_{i=1}^{N} y_i b_i(\mathbf{x})}$$ *Regularization networks*

$$\Updownarrow$$

$$\boxed{f(\mathbf{x}) = \frac{\sum_{i=1}^{N} w_i(\mathbf{x}) y_i}{\sum_{i=1}^{N} w_i(\mathbf{x})}}$$ *Kernel regression*

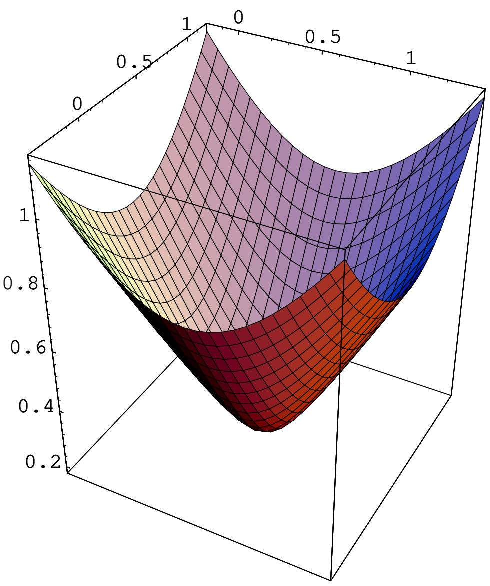
In both cases the value of f at point $\mathbf{x}$ is a weighted average of the values at the data points.

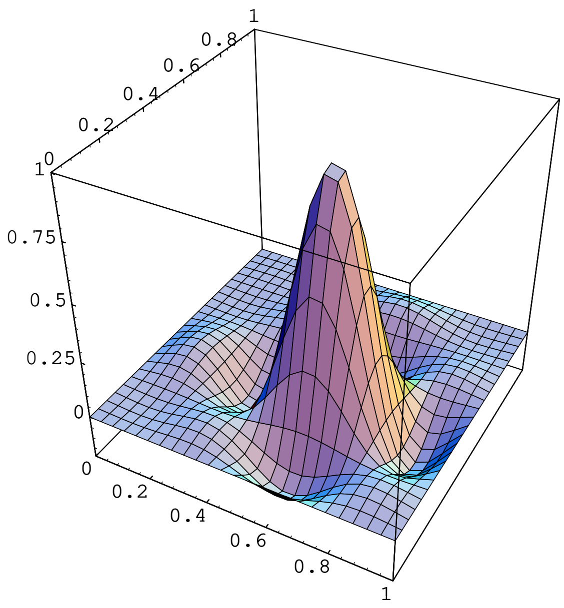
Memory based learning versus Regularization networks

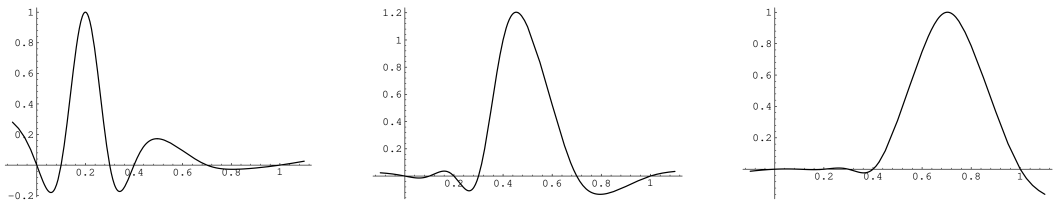
In kernel regression the kernels are chosen to be bell-shaped.

If regularization networks and kernel regression are similar the equivalent kernels b_i should be bell-shaped too.

Equivalent kernels for the standard regularization solution with multiquadric basis functions







From (Girosi, Jones and Poggio, 1993)

The equivalent kernels for standard regularization theory are more easily understood in the case in which an infinite number of data is available, that is the function g to be approximated is known:

$$H[f] = \int_{R^d} d\mathbf{x} \; (f(\mathbf{x}) - g(\mathbf{x}))^2 + \lambda \phi[f]$$

In this case the solution is simply

$$f = g * H$$

where H is *always* a bell-shaped function such that

$$\tilde{H}(\mathbf{s}) = \frac{\tilde{G}(\mathbf{s})}{\tilde{G}(\mathbf{s}) + \lambda}$$

Memory based learning versus Regularization networks

It has been proved (Silverman, 1984) that the equivalent kernels for one dimensional cubic splines are bell-shaped and very well localized.

The bandwidth of the equivalent kernel is inversely proportional to the density of the data.

This result can be generalized to higher order splines (Silverman, 1984) and probably to all regularization networks.