Optmization Methods for Machine Learning
Radial Basis function

Laura Palagi

http://www.dis.uniroma1.it/~palagi

Dipartimento di Ingegneria informatica automatica e gestionale A. Ruberti
Sapienza Università di Roma

Via Ariosto 25

SAPIENZA
UNIVERSITÀ DI ROMA

Interpolation problem

Given p distinct points in R^n:

$$X = \{x^i \in R^n,\ i = 1, \ldots, p\},$$

and a corresponding set of real numbers

$$Y = \{y^i \in R,\ i = 1, \ldots, p\}.$$

The interpolation problem consists in finding a function $f : R^n \to R$, in a given class of real functions $\mathcal{F}$, which satisfies:

$$f(x^i) = y^i \qquad i = 1, \ldots, P. \tag{1}$$

SAPIENZA
UNIVERSITÀ DI ROMA

Interpolation properties

For $n = 1$ the Interpolation pb. can be solved explicitly using polynomials

$$f(x) = \sum_{i=0}^{P-1} c_i t^i$$

For $n > 1$, the 2-layer MLP with g not polynomial satisfies

$$\sum_{j=1}^{P} v^j g({w^j}^T x^i - b^j) = y^i, \quad i = 1, \ldots, P$$

for some $w^j \in R^n$, and $v^j, b^j \in R$.

MLP can approximate arbitrarily well a continuous function provided that an arbitrarily large number of units is available.

SAPIENZA
UNIVERSITÀ DI ROMA

Interpolation properties

Being an universal approximator may be not enough from theoretical point of vew. An important property is the

existence of a best approximation

Informally: given a function $\bar{f}$ belonging to some set of functions $\mathcal{F}$ and given a subset $\mathcal{A}$ of $\mathcal{F}$ find an element of $\mathcal{A}$ which is closest to $\bar{f}$. If $d(f, g)$ is the distance between two elements f, g in $\mathcal{F}$, we consider the problem

$$d^*_{\mathcal{A}} = \inf_{a \in \mathcal{A}} d(\bar{f}, a)$$

If there exists $a^* \in \mathcal{A}$ that attains the infimum, namely $d^*_{\mathcal{A}} = d(\bar{f}, a^*)$ then a^* is the *best approximation* to $\bar{f}$ from $\mathcal{A}$.

SAPIENZA UNIVERSITÀ DI ROMA

Best approximation properties

MLP does not have the best approximation property.

Consider other approximation scheme based on Radial Basis functions (RBF)

$$\phi(\|x - x^j\|)$$

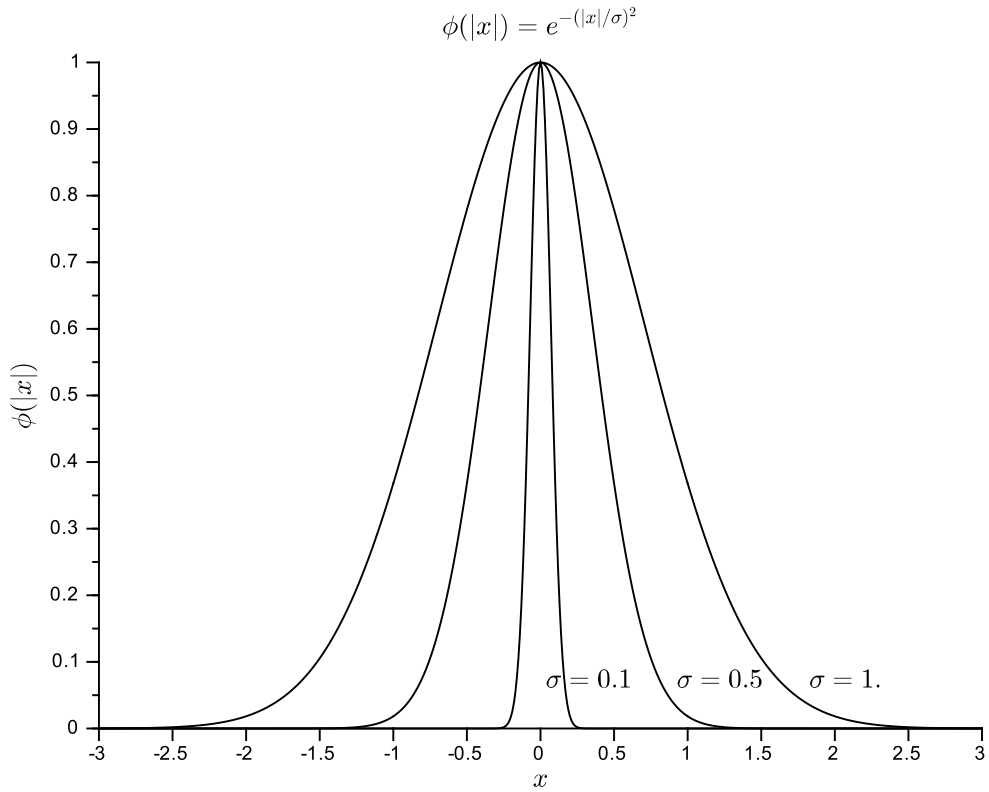
with $j = 1, \ldots P$.
$\phi : R^+ \to R$ is a suitable continuous function, called *radial basis function* since it is assumed that the argument of ϕ is the radius $r = \|x - x^j\|$.

SAPIENZA
UNIVERSITÀ DI ROMA

Gaussian

$$\phi(r) = e^{-(r/\sigma)^2}$$

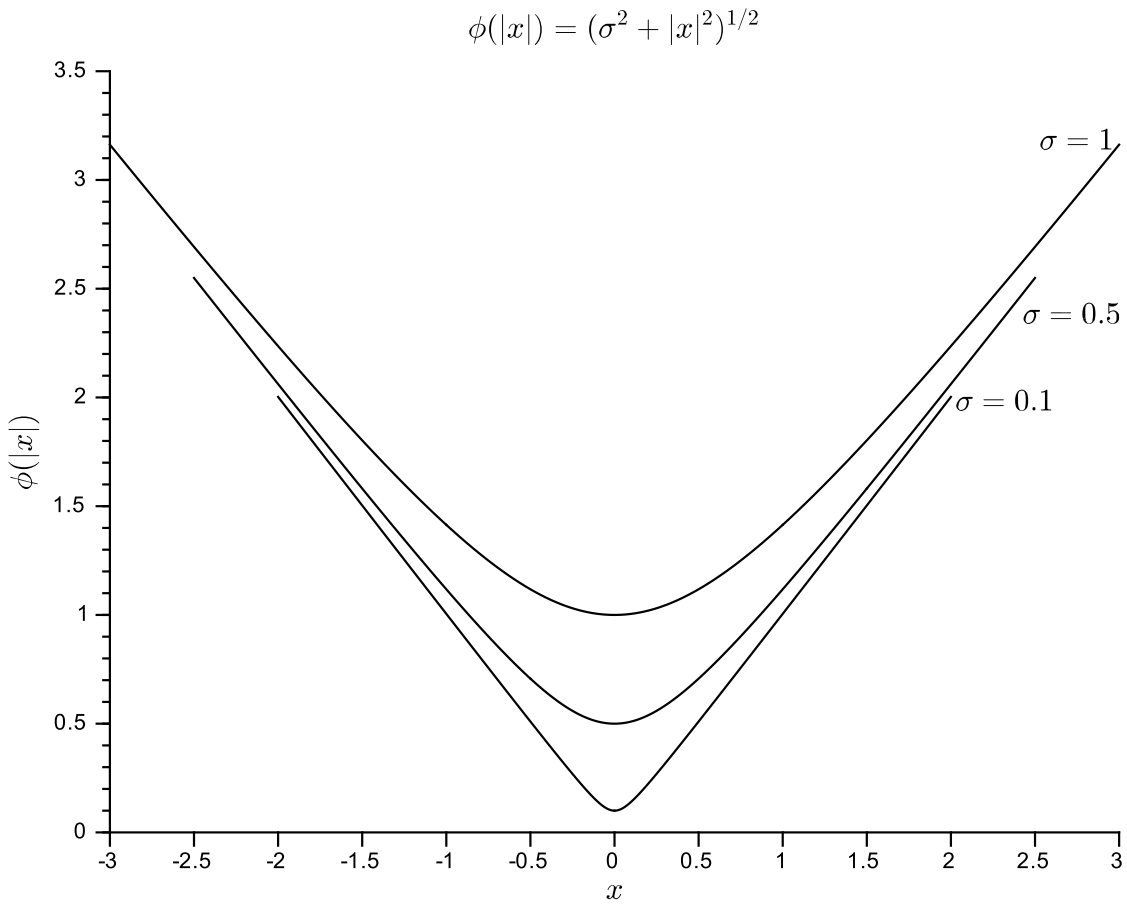
with $r > 0$



SAPIENZA
UNIVERSITÀ DI ROMA

Multiquadric

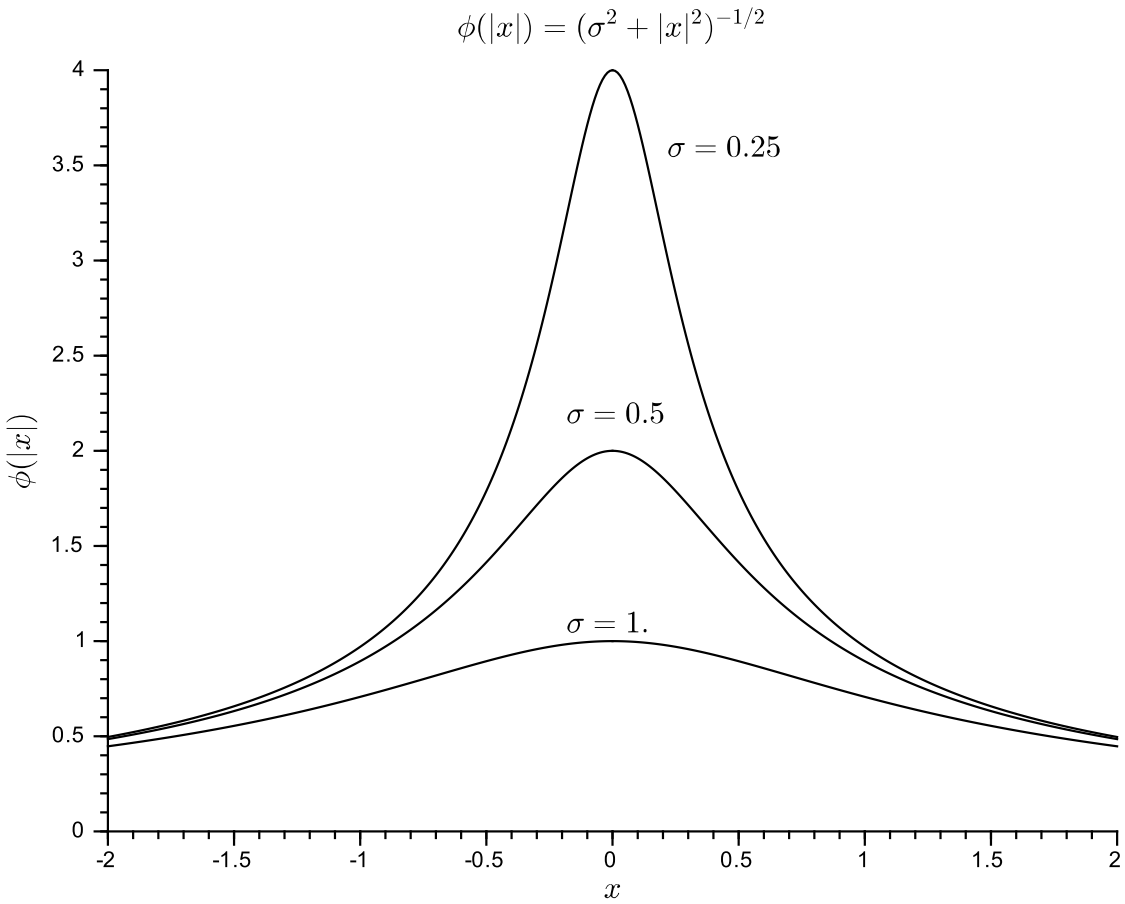
$$\phi(r) = (r^2 + \sigma^2)^{1/2}$$



SAPIENZA
UNIVERSITÀ DI ROMA

Inverse Multiquadric

$$\phi(r) = (r^2 + \sigma^2)^{-1/2}$$



SAPIENZA
UNIVERSITÀ DI ROMA

Other RBF

$$\begin{aligned} \phi(r) &= r && \text{linear spline} \\ \phi(r) &= r^3 && \text{cubic spline} \\ \phi(r) &= r^2 \log r, && \text{thin plate spline.} \end{aligned}$$

Interpolation by RBF

Given p distinct points in R^n:

$$X = \{x^i \in R^n,\ i = 1, \ldots, P\},$$

and consider functions of the form

$$f(x) = \sum_{j=1}^{P} w_j \phi(\|x - x^j\|), \tag{2}$$

where the data points $x^j \in X$ are the so called *centers* and the coefficients $w_j \in R$ are the *weights*.

SAPIENZA
UNIVERSITÀ DI ROMA

Interpolation by RBF

By imposing the interpolation conditions we get:

$$\sum_{j=1}^{P} w_j \phi(\|x^i - x^j\|) = y^i, \quad i = 1, \ldots, P. \tag{3}$$

It is a linear system of P equations in P unknowns. Let define the vectors $w = \begin{pmatrix} w_1 & \cdots & w_P \end{pmatrix}^T$, and $y = \begin{pmatrix} y_1 & \cdots & y_P \end{pmatrix}^T$, and the symmetric $P \times P$ matrix Φ with elements

$$\Phi_{i,j} = \phi(\|x^i - x^j\|), \quad 1 \leq i, j \leq P,$$

system (3) cam be written as:

$$\Phi w = y.$$

SAPIENZA
UNIVERSITÀ DI ROMA

Matrix Φ is non singular, provided that $P \geq 2$, that the interpolation points $x^j, j = 1, \ldots, P$ are distinct and using

- Gaussian (Φ positive definite)
- the multiquadric
- the inverse multiquadric (Φ positive definite)
- linear spline

Thus, the interpolation problem $\Phi w = y$ admits a unique solution. When ϕ pos. def. it can be computed by minimizing the (strictly) convex quadratic function in $\mathbb{R}^P$

$$F(w) = \frac{1}{2} w^T \Phi w - y^T w,$$

whose gradient is given by $\nabla F(w) = \Phi w - y$.

From Interpolation to approximation properties

Because of the remarkable properties of the RBFs, the RBF method is one of the most often applied approaches in multivariable interpolation.

This has motivated the attempt of employing RBFs also within approximation algorithms for the solution of classification and regression problems in data mining.

SAPIENZA
UNIVERSITÀ DI ROMA

Regularized RBF neural networks

Suppose that the set $\{(x^p, y^p),\ p = 1, \ldots, P\}$ of data has been obtained by random sampling of a function belonging to some space of functions $\mathcal{X}$ in the presence of noise

This problem of recovering the function or an estimate of it from the set of data is clearly ill posed since it has an infinite number of solutions.

In order to choose one particular solution we need to have some a priori knowledge of the function that has to be reconstructed.

The most common form of a priori knowledge consists in assuming that the function is smooth in the sense that two similar inputs correspond to two similar outputs.

Regularized RBF neural networks

The solution can be obtained from a variational principle which contains both the data and smoothness information.

Smoothness is a measure of the "oscillatory" behavior of f. Within a class of differentiable functions, one function is said to be smoother than another one if it oscillates less. A smoothness functional $\mathcal{E}_2(f)$ is defined and we consider

$$\min_f \mathcal{E}(f) = \mathcal{E}_1(f) + \lambda \mathcal{E}_2(f) = \frac{1}{2} \sum_{i=1}^{P} [y^i - f(x^i)]^2 + \lambda \mathcal{E}_2(f),$$

where the first term is enforcing closeness to the data and the second smoothness while the *regularization parameter* $\lambda > 0$ controls the tradeoff be tween these two terms.

Regularized RBF neural networks

It can be shown that for a wide class of smoothness functionals $\mathcal{E}_2(f)$, the solutions of the minimization all have the same form

$$\sum_{i=1}^{P} w_i \phi(\|x - c^i\|) = y,$$

Centers coincides with inputs

$$c^i = x^i, \quad i = 1, \ldots, P$$

and weights solve the regularized system

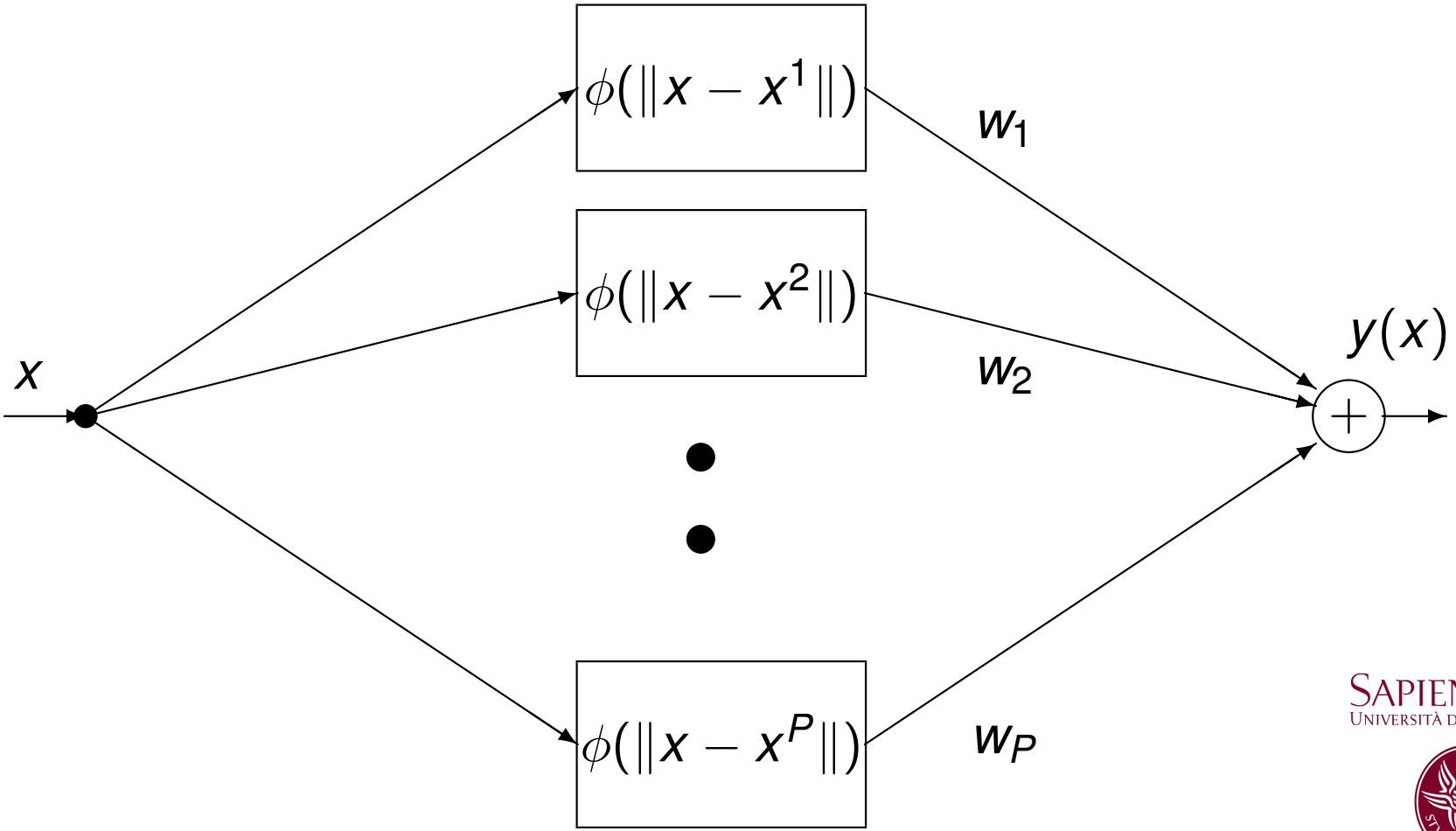
$$(\Phi + \lambda I) w = y$$

where

$$\Phi = \{\Phi_{ij}\}_{i,j=1,\ldots,P} = \{\phi(\|x^i - x^j\|)\}_{i,j=1,\ldots,P}$$

SAPIENZA
UNIVERSITÀ DI ROMA

2-layer Regularized RBF network



SAPIENZA
UNIVERSITÀ DI ROMA

2-layer Regularized RBF network

- RBF are *universal approximator*: any continuous function can be approximated arbitrarily well on a compact set, provided a sufficiently large number of units, and for an appropriate choice of the parameters
- RBF possess the *best approximation property*, namely there exists the best approximation and in most cases (under assumptions often satisfied) is unique (RBF is linear in parameters w)
- The value of λ can be selected by employing cross validation techniques and this may require that system $(\Phi + \lambda I)w = y$ is solved several times.

SAPIENZA
UNIVERSITÀ DI ROMA

2-layer Generalized RBF network

When P is very large, the cost of constructing a regularized RBF network can be prohibitive. Indeed, the computation of the weights $w \in R^P$ requires the solution of a possible ill conditioned linear system, which costs $O(P^3)$.

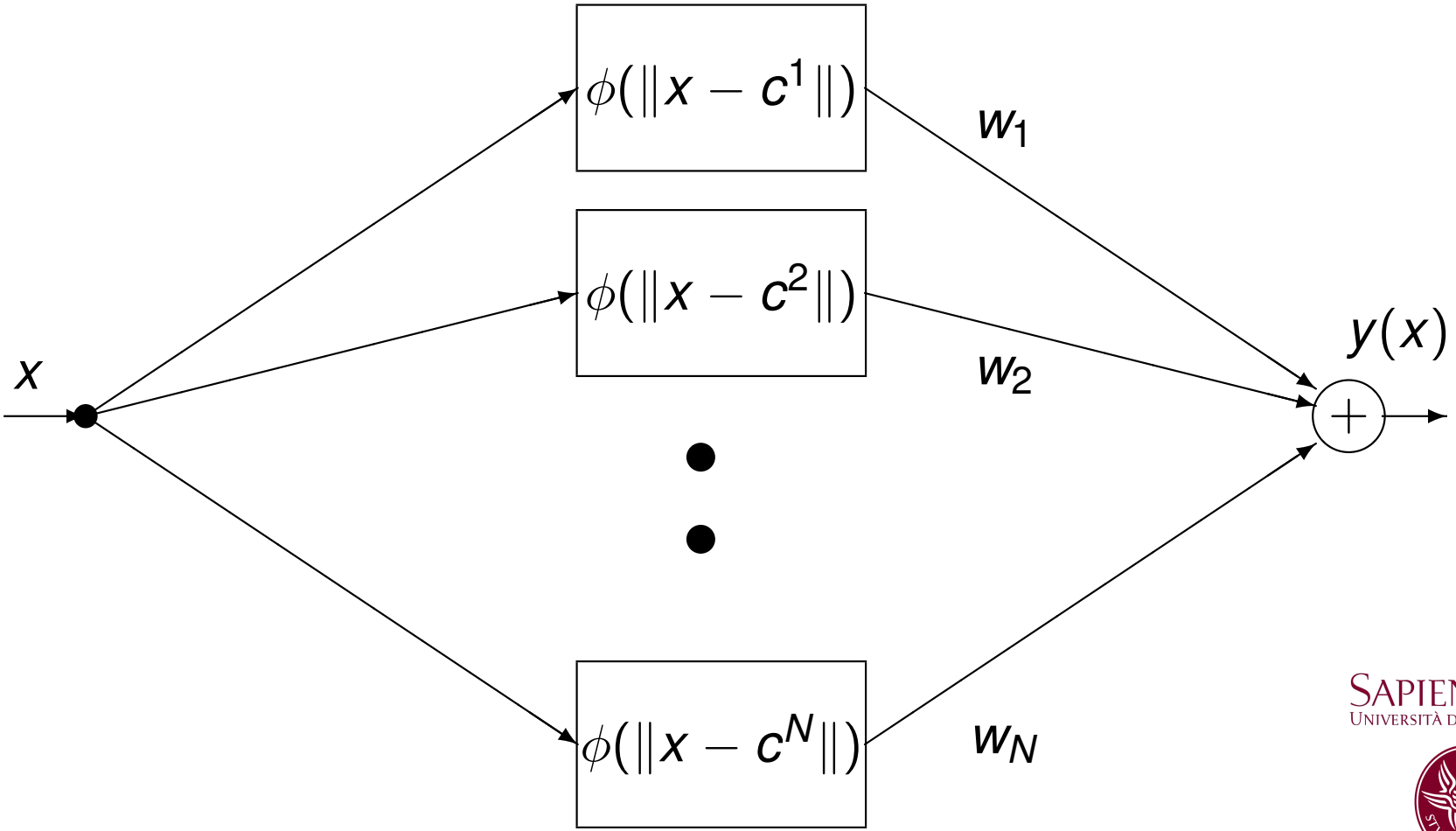
Generalized RBF neural network are used where the number N of neural units is much less than P.

The output of the network can be defined by

$$y(x) = \sum_{j=1}^{N} w_j \phi_j(\|x - c_j\|), \tag{4}$$

where both the *centers* $c_j \in R^n$ and the weights w_j $j = 1, \ldots, N$ must be selected appropriately.

2-layer Generalized RBF network



SAPIENZA
UNIVERSITÀ DI ROMA

2-layer Generalized RBF network

- GRBF are *universal approximator*: any continuous function can be approximated arbitrarily well on a compact set, provided a sufficiently large number of units, and for an appropriate choice of the parameters
- GRBF may NOT possess the *best approximation property*. However if the centers are fixed, the approximation problem becomes linear with respect to w and the existence of a best approximation is guaranteed
- in he general case, both the centers and the weights are treated as variable parameters and the approximation is *nonlinear*
- As $N << P$, GRBF performs inherently a *structural stabilization* which may prevent the occurrence of overtraining.

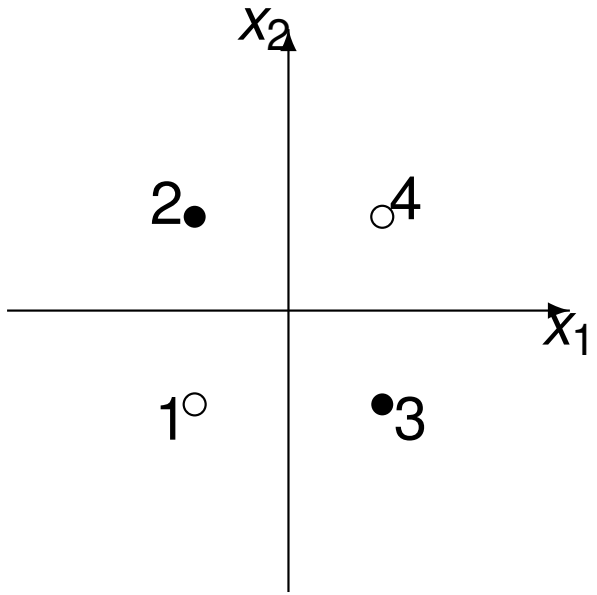
SAPIENZA UNIVERSITÀ DI ROMA

An example: Exclusive OR

The logical function XOR

XOR

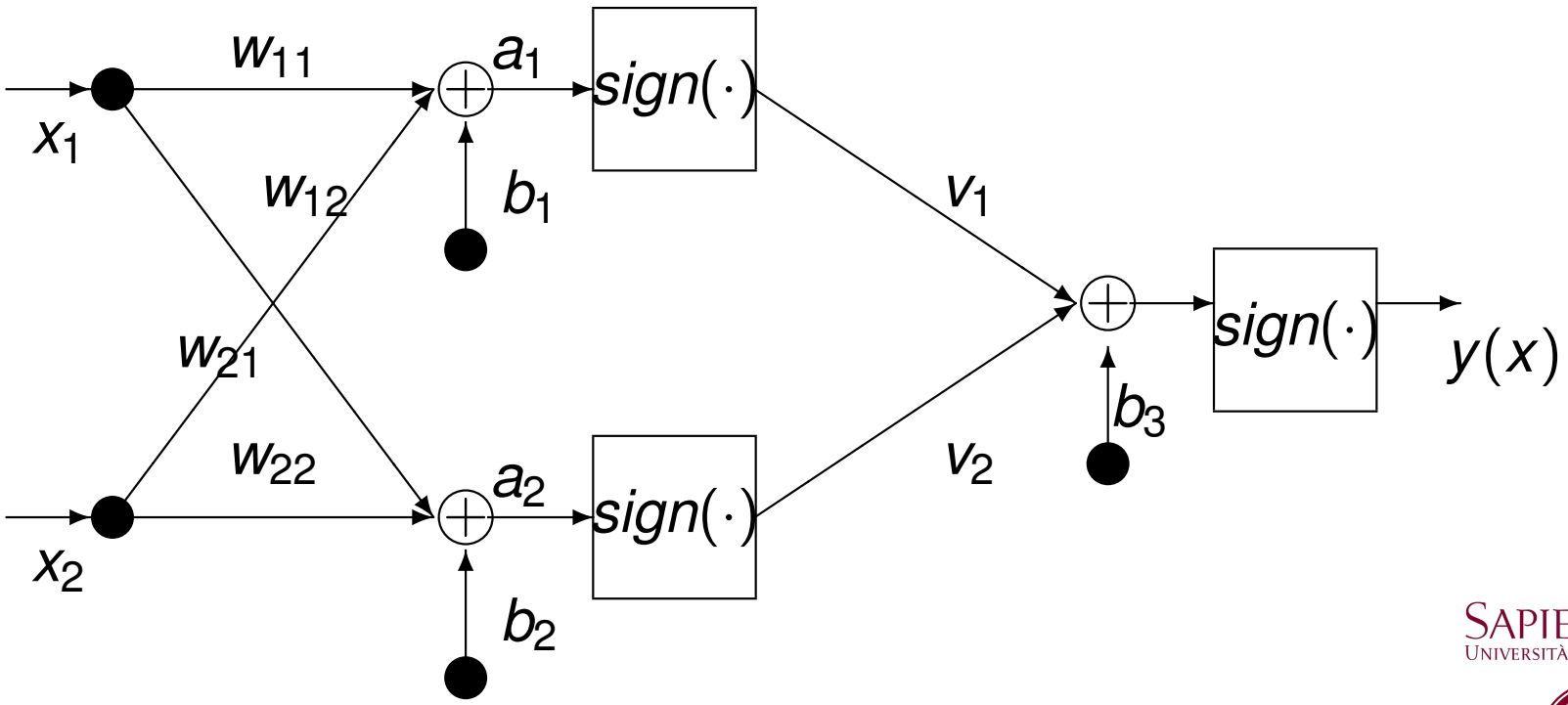
p	x_1	x_2	y^p
1	-1	-1	-1
2	-1	1	1
3	1	-1	1
4	1	1	-1



Perceptron (linear separator) doesn't work

SAPIENZA
UNIVERSITÀ DI ROMA

Two layer MLP



SAPIENZA
UNIVERSITÀ DI ROMA

Two layer MLP

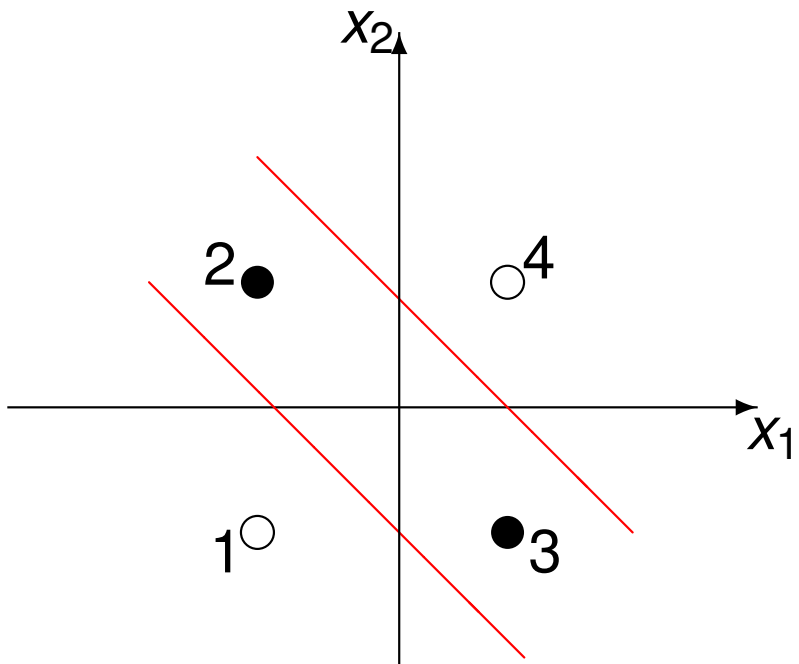
Choose $w_{11} = w_{22} = 1$ and $w_{12} = w_{21} = -1$, $b_1 = b_2 = -1$, $v_1 = v_2 = 1$ $b_3 = 0.1$ (output bias). We get

$$a_1 = x_1 - x_2 - 1 \qquad z_1 = sign(a_1)$$
$$a_2 = -x_1 + x_2 - 1 \qquad z_2 = sign(a_2)$$
$$y = sign(z_1 + z_2 + 0.1)$$

input p	a_1	a_2	z_1	z_2	$z_1 + z_2 + 0.1$	y
1	-1	-1	-1	-1	-1.9	-1
2	-3	1	-1	1	0.1	1
3	1	-3	1	-1	0.1	1
4	-1	-1	-1	-1	-1.9	-1

Two layer MLP

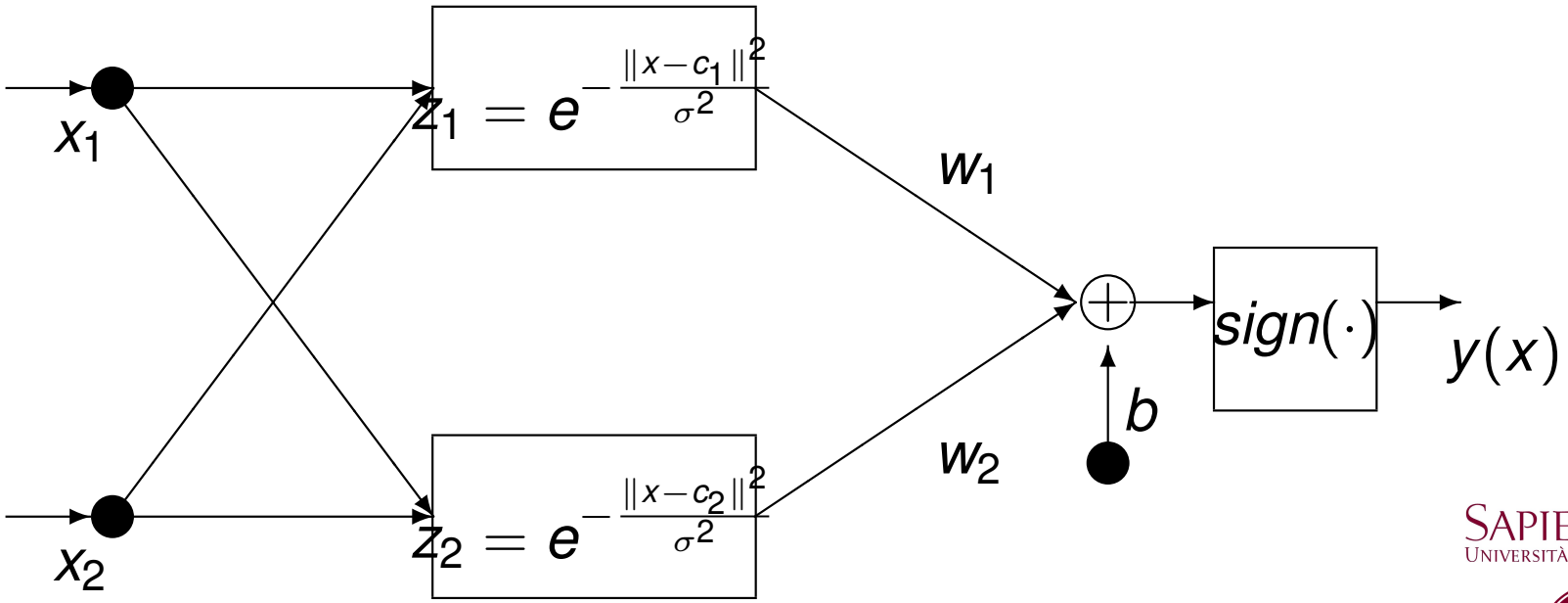
This MLP network with two hidden nodes realizes a nonlinear separation (each hidden node describes one of the two lines). The output node combines the outputs of the two hidden layer.



SAPIENZA
UNIVERSITÀ DI ROMA

RBF network

Consider a RBF network with two units ($N = 2$) with centers c_1, c_2 and assume the activation function is a gaussian $g_j = e^{-(\|x-c_j\|/\sigma)^2}$



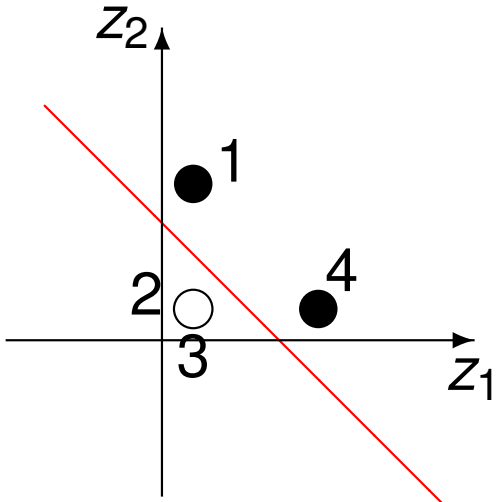
SAPIENZA UNIVERSITÀ DI ROMA

RBF network

Choose $\sigma = \sqrt{2}$ and $c_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $c_2 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$ We transform the problem into a linearly separable form.

XOR

p	$e^{-\frac{\|x-c_1\|^2}{\sigma^2}}$	$e^{-\frac{\|x-c_1\|^2}{\sigma^2}}$	y^p
1	e^{-4}	1	-1
2	e^{-2}	e^{-2}	1
3	e^{-2}	e^{-2}	1
4	1	e^{-4}	-1



SAPIENZA
UNIVERSITÀ DI ROMA

The output takes the form

$$y(x) = w_1 e^{-\frac{\|x-c_1\|^2}{\sigma^2}} + w_2 e^{-\frac{\|x-c_2\|^2}{\sigma^2}} + b$$

Minimizing the training error

$$\min_{w,b} \sum_{p=1}^{4} (y(x^p) - y^p)^2$$

we get the optimal solution (w^*, b^*) that gives $E = 0$

$$\begin{pmatrix} w_1 \\ w_2 \\ b \end{pmatrix} = \begin{pmatrix} -2,675065656 \\ -2,675065656 \\ 1,72406123 \end{pmatrix}$$

and the RBF network has been trained.