

# L'intelligenza biologica

## Reti Neurali con funzione di attivazione a base radiale

Alberto Borghese  
Università degli Studi di Milano  
Laboratory of Applied Intelligent Systems (AIS-Lab)  
Dipartimento di Scienze dell'Informazione  
[borgnese@dsi.unimi.it](mailto:borgnese@dsi.unimi.it)



A.A. 2003-2004

1/49

<http://homes.dsi.unimi.it/~borgnese>



## Sommario



### .Le Radial Basis Function Networks.

Modalità di apprendimento nelle reti RBF.

Apprendimento Ibrido.

Apprendimento ispirato alla teoria dei segnali.

Apprendimento multi-scala.

A.A. 2003-2004

2/49

<http://homes.dsi.unimi.it/~borgnese>

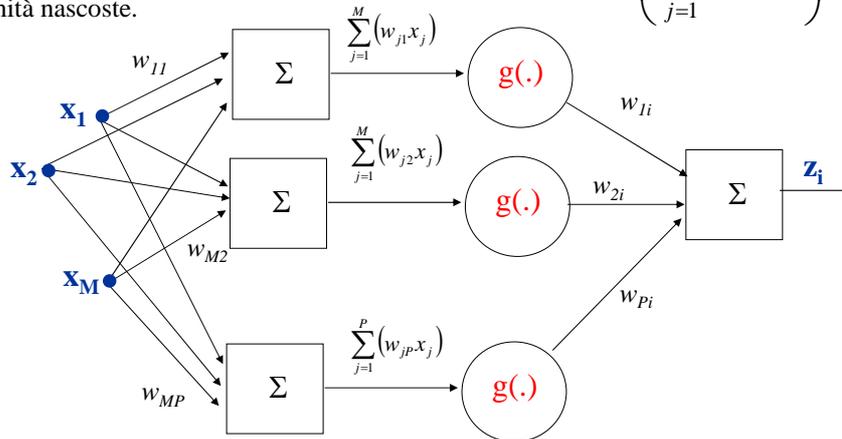


# MLP : Multi-layer Perceptron



- M ingressi
- N uscite
- P unità nascoste.

$$h_i = g\left(\sum_{j=1}^M (w_{ij}x_j)\right)$$



Difficult to make it learn (almeno quando  $g(\cdot)$  è non lineare, vedi logistica)!

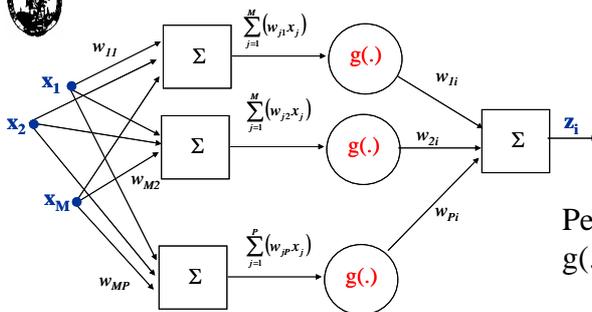
A.A. 2003-2004

3/49

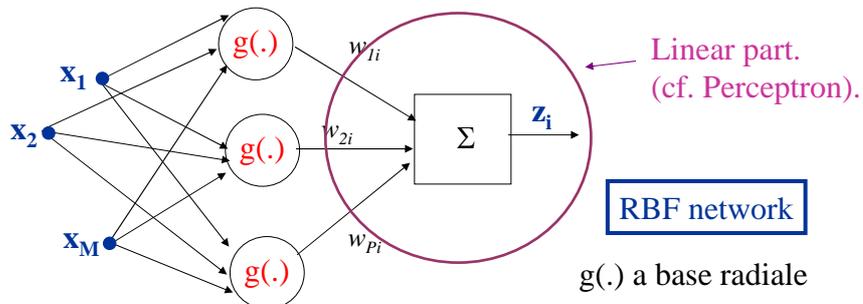
<http://homes.dsi.unimi.it/~borgnese>



# RBF: una forma differente di non-linearità



Perceptrone multi-layer  
 $g(\cdot)$  sono sigmoidali.



A.A. 2003-2004

4/49

<http://homes.dsi.unimi.it/~borgnese>



## Esempi di RBF network



$$z_i = \sum_{j=1}^N w_{ji} \frac{e^{-\frac{\|\mathbf{x}-\mathbf{x}_j\|}{\sigma_j^2}}}{(\sqrt{\pi\sigma_j})^D} \quad \leftarrow \text{Gaussiana}$$

$$z_i = \sum_{j=1}^N w_{ji} \|\mathbf{x}-\mathbf{x}_j\|^{2n+1} = \sum_{i=1}^N w_{ji} r_j^{2n+1} \quad \leftarrow \text{Spline}$$

$$z_i = \sum_{j=1}^N w_{ji} \|\mathbf{x}-\mathbf{x}_j\|^{2n} \ln(\|\mathbf{x}-\mathbf{x}_j\|) \quad \leftarrow \text{Spline}$$

Sono tutte funzioni di attivazione con base Radiale (simmetria circolare).

A.A. 2003-2004

5/49

<http://homes.dsi.unimi.it/~borghese>



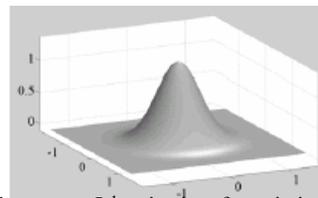
## Funzioni di attivazione a simmetria radiale



Supponiamo che vale:  $g(x) = g(Rx)$ , con R matrice di rotazione. Allora, si dice che  $g(x)$  è invariante a rotazione, ovvero si ha simmetria radiale.

Per le funzioni di attivazione Gaussiane, questo vuol dire passare dalla formulazione in cui si utilizza una Gaussiana non simmetrica:

$$z = \sum_{j=1}^M w_j e^{-(\mathbf{x}-\mathbf{x}_j)\Sigma_j(\mathbf{x}-\mathbf{x}_j)^T}$$



Alla formulazione a simmetria radiale:  $z = \sum_{j=1}^N w_j \frac{e^{-\frac{\|\mathbf{x}-\mathbf{x}_j\|}{\sigma_j^2}}}{(\sqrt{\pi\sigma_j})^D}$  (ha norma  $L^1$  unitaria, cf. statistica)

Questo riflette il fatto che non si hanno assunzioni a-priori sull'importanza delle variabili: tutte le direzioni hanno la stessa "importanza".

ghese



## Approssimazione e RBF networks: ipotesi



Apprendimento da esempi è in molti casi equivalente ad approssimare una **iper-superficie** (multi-variabile).

Dati  $\{P(x_1, x_2, x_3, \dots, x_M, x_{M+1})\}$  posso scrivere il mio set di esempi come:  $\{P(\mathbf{x}, s(\mathbf{x}))\}$  con  $s = x_{M+1} \in \mathbb{R}^1$  e  $\mathbf{x} \in \mathbb{R}^M$ .

$$\mathbb{R}^M \rightarrow \mathbb{R}^1$$

Ad esempio nello spazio 3D la mia funzione rappresenterà l'altezza di una superficie punto per punto:  $s = s(x, y)$ .

E' una ricostruzione  $2\frac{1}{2}$  D ottenuta come **somma pesata** di Gaussiane.

A.A. 2003-2004

7/49

<http://homes.dsi.unimi.it/~borgnese>

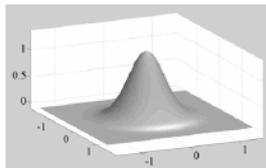


## Funzioni quasi-locali



Ispirazione biologica. **Campo recettivo** di un neurone sensoriale (e.g. I neuroni della coclea sono sensibili ad una banda di frequenza, i neuroni della corteccia somatosensoriale primaria ad una regione del corpo, alcuni neuroni della corteccia visiva sono sensibili all'orientamento dei contorni...).

Ciascuna unità risponde ad input in una regione limitata, ovvero ciascun input attiva solo un certo numero di unità: quelle il cui input è vicino significativamente al centro dell'unità stessa.



- Quasi-locale: output va velocemente al di fuori di una regione.
- Possibilità di definire un “campo recettivo”.

A.A. 2003-2004

8/49

<http://homes.dsi.unimi.it/~borgnese>



## Sommario



Le Radial Basis Function Networks.

**Modalità di apprendimento nelle reti RBF.**

Apprendimento Ibrido.

Apprendimento ispirato alla teoria dei segnali.

Apprendimento multi-scala.



## Apprendimento in una RBF network



Supervised learning.

Regolarizzazione.

Approccio Bayesiano.

Schemi ibridi.

Teoria del filtraggio.

$$z_i = \sum_{j=1}^N w_{ji} \frac{e^{-\frac{\|\mathbf{x}-\mathbf{x}_j\|}{\sigma_j^2}}}{(\sqrt{(\pi\sigma_j)^D}}$$

*Dato un insieme  $\{z_k = S(\mathbf{x}_k)\}$  trovare i parametri tali che la iper-superficie risultante “fitta meglio” i dati.*



## Supervised learning style



Definisco una funzione costo  $J(\cdot)$ , funzione dei parametri  $\alpha$ .

$$S(P | \alpha) = \sum_{j=1}^M w_j g(P - P_j | \sigma_j) \quad \min_{\{\alpha\}} J(a) = \min_{\{\alpha\}} \left( S[(P_k | \alpha) - s_k]^2 \right)$$

Il minimo di  $J(\cdot)$  si ha quando le derivate rispetto ai parametri si annullano → Tecniche di ottimizzazione tramite gradiente.

Non ci sono vincoli sui parametri. I valori di  $\sigma$  tendono a crescere ed i centri a concentrarsi al centro o a respingersi fuori dal campo recettivo. Minimi locali.



## Regolarizzazione



Si aggiungono ipotesi sulla funzione (stabilizzatori della soluzione).

Regolarizzazione -> Problema variazionale:

$$H[s] = \min_{\{s(\cdot)\}} \left( \sum_k (s(\mathbf{x}_k) - y_k)^2 + \lambda \Phi[s(\cdot)] \right)$$

Misura la fedeltà della soluzione  $f(\cdot)$  ai campioni, grado di interpolazione.

Stabilizzatore (e.g. Penalizza le variazioni brusche), grado di smoothness.

$\lambda$  è un parametro scelto dall'utente.



## Approccio Bayesiano



Supponiamo che i nostri **esempi**:  $\{\mathbf{x}_k; s_k\}$  siano ottenuti **campionando** in modo randomico una iper-superficie  $s(\cdot)$  non nota, in presenza di rumore:  $s_k = s(\mathbf{x}_k) + \varepsilon_k$  ( $i=1, \dots, N$ ). Gli  $\varepsilon_k$  sono indipendenti, sono funzioni randomiche che rappresentano il **rumore** con una sua distribuzione statistica.

Supponiamo di considerare la iper-superficie  $s(\cdot)$  come una realizzazione di un campo randomico con una certa distribuzione a-priori, che indichiamo con  $P[s(\cdot)]$ . Questo rappresenta la conoscenza **a-priori** sul campo di iper-superficie da cui è estratta. Si può utilizzare per dare un **vincolo soft** alla soluzione, favorendo quelle superfici che meglio soddisfano le ipotesi a priori sul campo delle iper-superfici.



## Approccio Bayesiano: formalizzazione



Definisco  $s(\cdot)$  la mia iper-superficie e la associo ad una funzione di probabilità da determinare e  $Q = \{\mathbf{x}_k; s_k\}$  l'insieme dei dati.

- $P[s|Q]$  – Probabilità **condizionata** della funzione  $s$ , dati i campioni  $Q$ .
- $P[Q|s]$  – Probabilità **condizionata** di ottenere i dati  $Q$  da un modello  $s(\cdot)$ : probabilità di ottenere i valori  $\{s_k\}$  campionando la funzione nei  $\{\mathbf{x}_k\}$ : rappresenta un modello dell'errore.
- $P[s(\cdot)]$  – Probabilità **a-priori** sul campo randomico  $s(\cdot)$ .

Dal teorema di Bayes si ottiene:  $P[s(\cdot)|Q] = P[Q|s(\cdot)] P[s(\cdot)]$



## Approccio Bayesiano: le ipotesi



- $P[Q|s(\cdot)]$  – Assumiamo che il rumore di misura ( $\varepsilon_k$ ) sia normalmente distribuito con varianza  $\sigma$ .

$$P[Q | s(\cdot)] = e^{-\frac{1}{2\sigma^2} \sum_{k=1}^N (z_k - s(\mathbf{x}_k))^2}$$

- $P[s(\cdot)]$  – Viene scelto in modo analogo al termine di regolarizzazione: diamo probabilità alta a funzioni che hanno costo basso:

$$P(s(\cdot)) = e^{-a\phi[s(\cdot)]}$$

Tramite la regola di Bayes, otteniamo:

$$P(s(\cdot) | Q) = e^{-\frac{1}{2\sigma^2} \sum_{k=1}^N (y_k - s(\mathbf{x}_k))^2} e^{-a\phi[s(\cdot)]} = e^{-\frac{1}{2\sigma^2} \left[ \sum_{k=1}^N (y_k - s(\mathbf{x}_k))^2 + a\sigma^2\phi[s(\cdot)] \right]}$$

Un modo classico di stimare  $s(\cdot)$  è mediante la MAP,  $s(\cdot)$  che minimizza l'esponente  $\rightarrow$  problema di regolarizzazione con  $\lambda = 2\sigma^2\alpha$ . se



## Sommario



Le Radial Basis Function Networks.

Modalità di apprendimento nelle reti RBF.

**Apprendimento Ibrido.**

Apprendimento ispirato alla teoria dei segnali.

Apprendimento multi-scala.



## Learning ibrido



Supponiamo di ricostruire una ipersuperficie  $S(P)$ : I parametri  $\{\alpha\}$  da determinare in una RBF network Gaussiana sono:

$$S(P | \alpha) = \sum_{j=1}^M w_j G(\|P - P_j\| | \sigma_j)$$

- Numero di unità ( $M$ ), Posizione  $\{P_j\}$  e standard deviation  $\{\sigma_j\}$  delle unità. **Definiscono la struttura della rete.**
- Pesi  $\{w_j\}$  sinaptici (perceptrone con unità di attivazione Gaussiane).



## Determinazione della posizione



$$S(P | \alpha) = \sum_{j=1}^M w_j G(P - P_j | \sigma_j)$$

***Per determinare la posizione delle unità:***

- Definizione del numero di unità.
- Posizionamento delle unità mediante clusteringin: (k-means, fuzzy-clustering,...).

Filosofia: vado ad inserire più unità dove ci sono più dati.



## Determinazione dell'ampiezza delle unità $\{\sigma_j\}$



$$S(P|\alpha) = \sum_{j=1}^M w_j G(P - P_j | \sigma_j)$$

Si utilizzano euristiche di tipo P-nearest-neighbour.

E.g. primo ordine:  $\sigma = \langle \Delta x_{\alpha\beta} \rangle$

Dove  $\alpha, \beta$  sono tutte le coppie di centri ottenute associando ad  $\alpha$  il centro più vicino.

$\sigma$  controlla il grado di sovrapposizione tra due Gaussiane vicine.

La media può essere calcolata localmente: P-loc-nearest-neighbor e quindi ogni unità avrà una sua ampiezza caratteristica.

A.A. 2003-2004

19/49

<http://homes.dsi.unimi.it/~borgnese>



## Determinazione dei pesi sinaptici



$$S(P|\alpha) = \sum_{k=1}^M w_k G(P - P_k | \sigma_k)$$

Determinata la struttura della rete, per ogni input avrò un output che è **funzione lineare dei pesi**.

Utilizzo la tecnica utilizzata per il perceptrone. Quale?

Ricordiamo che le reti RBF sono chiamate anche perceptroni non lineari o Gaussiani.

NB rilevanza dal punto di vista biologico e computazionale. Ho una struttura fissa: funzioni con campo recettivo locale, con queste unità posso realizzare funzioni complesse semplicemente agendo sui pesi. Le unità mi forniscono già una pre-elaborazione dell'input locale in spazio ma anche in frequenza.

se



## Risultati da Moody & Darken



Esperimento di riconoscimento automatico del parlato.

Analizzo in tempo reale lo spettro per estrarre le formanti.

Dalle formanti voglio riconoscere le vocali (sillabe).

Seconda formante [Hz]



Prima formante [Hz]

Posizione e standard deviation delle unità della rete (20 unità). Queste unità mi ricostruiscono una superficie (pre-processing) su cui poi andrò ad applicare tecniche di clustering per individuare le diverse vocali.



## Sommario



Le Radial Basis Function Networks.

Modalità di apprendimento nelle reti RBF.

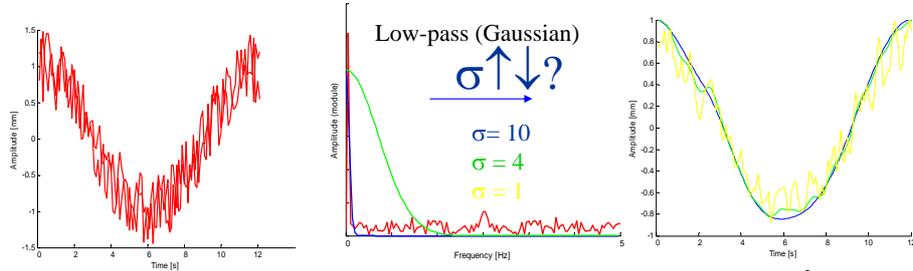
Apprendimento Ibrido.

**Apprendimento ispirato alla teoria dei segnali.**

Apprendimento multi-scala.



## Filtraggio e convoluzione



Consideriamo il filtro Gaussiano:  $h(x; x_c) = G(x - x_c; \sigma) = e^{-\frac{(x-x_c)^2}{\sigma^2}}$   
 La posizione della Gaussiana è definita da  $x_c$ , la sua ampiezza da  $\sigma$ .

$$s_{out}(x_k) = 1/M \sum_{n=-M}^M G(x_n) s_{in}(x_n - x_k) = G(x) * s_{in}(x)$$

La Gaussiana attenua in modo progressivamente maggiore le armoniche con frequenze più elevate (filtraggio passa-basso)

ise



## Applicazione: 3D scanner



- Pair of video-cameras + standard laser pointer.
- The range data are obtained by "painting" the surface manually.
- Set of range data, which is denser where required.
- High precision in spot localization (cross-correlation, bright image).

$$\{z_k = s(x_k, y_k)\}$$

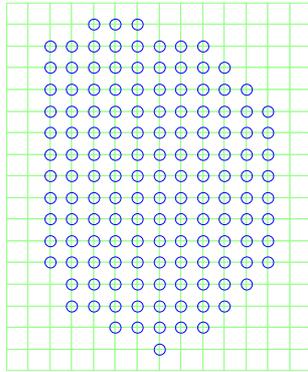
Drawback: High scanning time.



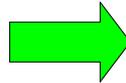
## The RBF model



$$s(P) = \sum_{k=1}^M w_k G(P - P_k | \sigma_k)$$



Grid support



Simple computation

### Structural parameters to be set:

- M – Number of units.
- $\{P_k\}$  – Position of the units.
- $\sigma_k$  – Gaussian width.

### Linear parameters:

- $\{w_k\}$  – “Weights”.



## RBF e Filtri



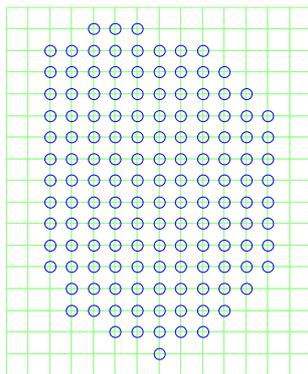
$$S(P) = \sum_k w_k G(P - P_k | \sigma_k)$$

Gaussiane equispaziate su un grid.

$$|P_{k+1} - P_k| = \Delta P_k$$

$$\sigma_k = \sigma \quad \forall k$$

Gaussiane normalizzate  $\|g(\cdot)\| = 1$



$$s(P) = \sum_{k=1}^M s_{m_k} G(P - P_k | \sigma) =$$

$$s_{m_1} G(P - P_1 | \sigma) + s_{m_2} G(P - P_2 | \sigma) + \dots =$$

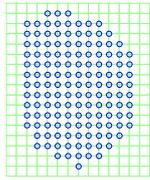
$$s_m(\cdot) * G(P - \cdot | \sigma)$$

$G(\cdot)$  costituisce un filtro passa-basso:  $S(P) = S_m(\cdot) * G(P - \cdot | \sigma)$

$$w_k = s_k = s(P_k)$$

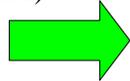


## Role of the scale $\sigma$

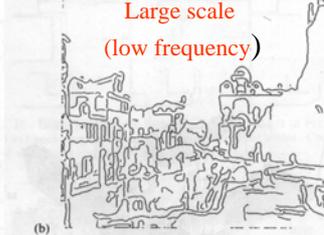
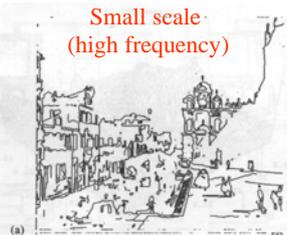


Linear filter  
(low-pass)

$$S(P) = \sum_k^M s_k G(P - P_k | \sigma)$$



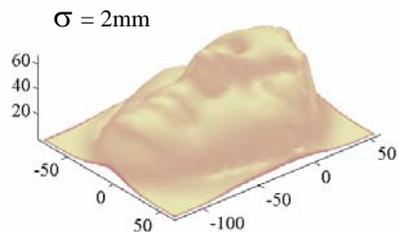
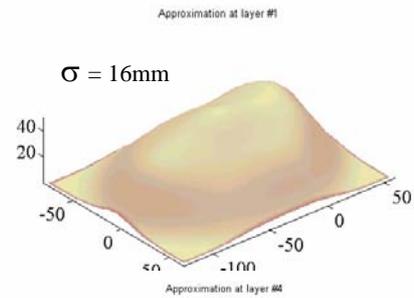
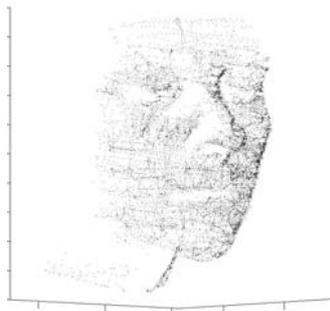
$\sigma$  - Scale of the filter (bandwidth)



Setting  $\sigma$ , we set the spatial frequency of the reconstruction.



## Ricostruzione da range data a scala diversa



$$S(P) = \sum_k^M s_k G(P - P_k | \sigma)$$



## Come determinare i parametri strutturali



Si determina il dettaglio desiderato (frequenza spaziale),  $\sigma$ .

Al diminuire di  $\sigma$ , aumenta la frequenza spaziale -> per il teorema di Shannon aumenta la frequenza di campionamento (vicinanza tra due Gaussiane).

$$\sigma \rightarrow v_{\max} \rightarrow v_S \rightarrow \Delta P_k$$

Valori di esempio:  $\Delta P = 0.6826 \sigma$  equivalente ad un 75% di overlap tra Gaussiane adiacenti.



## Computation of the $\{s_k\}$

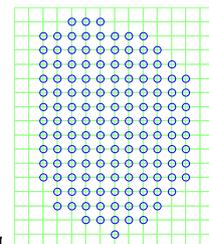


$$S(P) = \sum_k^M s_k G(P - P_k | \sigma)$$

We do not have the height of the surface in the grid crossing,  $\{s_k\}$ , and we must **estimate** it.

$$s_k = \frac{\sum_m s_m(P_m) f(|P_m - P_k|)}{\sum_m f(|P_m - P_k|)}$$

We can apply weighted mean estimate where each measured point,  $s_m = s(P_m)$  is weighted with a function,  $f(|\cdot|)$ , of its **distance** from  $P_k$ .



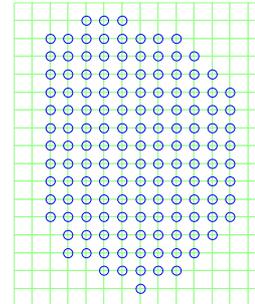


## Riassunto sulla configurazione delle RBF network



$$S(P) = \sum_k^M s_k G(P - P_k | \sigma)$$

- Scelta di un valore di  $\sigma$  adeguato.
- Calcolo dell'”impaccamento” delle Gaussiane:  $\sigma \rightarrow \Delta P_k \rightarrow M$  (teoria del filtraggio).
- Stima dell'altezza della superficie nei grid crossings.



$$s_k = \frac{\sum_m s_m(P_m) f(|P_m - P_k|)}{\sum_m f(|P_m - P_k|)}$$



## Problems



$$S(P) = \sum_{k=1}^M s_k G(P - P_k | \sigma) \quad s_k = \frac{\sum_m s_m(P_m) f(|P_m - P_k|)}{\sum_m f(|P_m - P_k|)}$$

### Related to the choice of $\sigma$ :

- No information on spatial frequency content.
- Frequency variability.  $\sigma$  should be chosen small enough to reconstruct the finest details (high packing). Few points close to  $P_k$  to estimate  $s_k = S(P_k)$ . may result in other regions.

### No control on the reconstruction error.

### How many and which points use to estimate $\{s_k\}$ ..

The approach as it is has little adaptivity and little hope to be real-time.



## Sommario



Le Radial Basis Function Networks.

Modalità di apprendimento nelle reti RBF.

Apprendimento Ibrido.

Apprendimento ispirato alla teoria dei segnali.

**Apprendimento multi-scala.**



## Strategia costruttiva di $s(P)$



Inizio con una scala molto ampia e valuto il residuo.

Approssimazione del residuo, dove presente, ad una scala più piccola.

Calcolo di un secondo residuo

.....

Fino a che l'errore di approssimazione soddisfa il mio criterio di approssimazione (cross-validation, contenuto in frequenza del rumore, rischio empirico o strutturale...).

NB Le Gaussiane vengono inserite negli incroci della griglia **solamente** dove l'errore (residuo) è sopra soglia (*calcolo del residuo locale*).



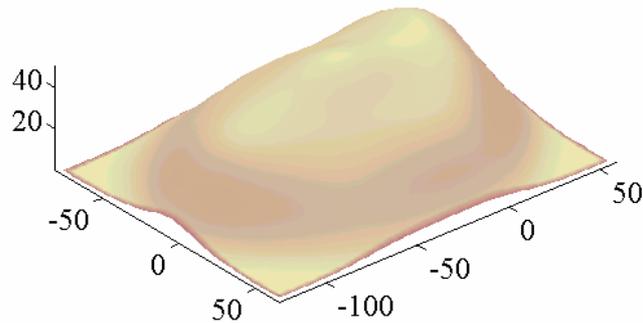
## Costruzione di un primo livello a scala ampia



$\sigma_1 = 16\text{mm}$

Approximation at layer #1

$$s^I(P) = \sum_{k=1}^{M_1} s_k G(P - P_k | \sigma_1)$$



A.A. 2003-2004

35/49

<http://homes.dsi.unimi.it/~borgnese>



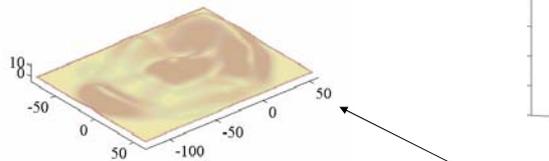
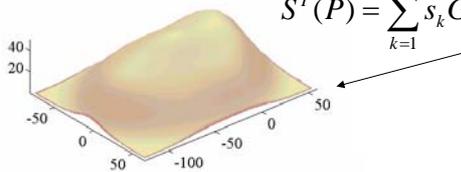
## Calcolo del residuo



Residuo della ricostruzione:  $r^I = \{z_m - S^I(P)\} \forall P$  misurato,  $P_m$ .

Approximation at layer #1

$$S^I(P) = \sum_{k=1}^{M_1} s_k G(P - P_k | \sigma_1)$$



Residuo:  $\{z_m - S^I(P_m)\}$

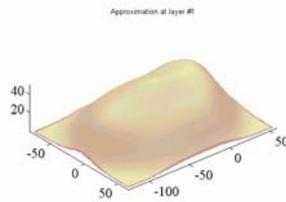
A.A. 2003-2004

36/49

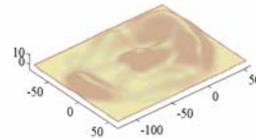
<http://homes.dsi.unimi.it/~borgnese>



## Error evaluation



Continuous surface at a low scale ( $\sigma = 16\text{mm}$ )



Residual at the sampled points

The residual is computed in the points sampled over the surface (range data).

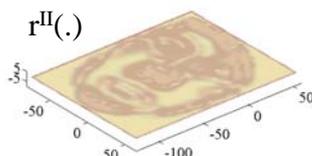
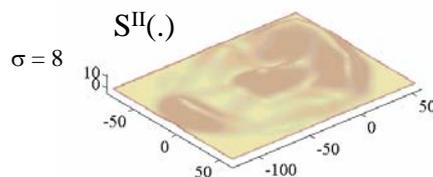
The error is evaluated using an **integral metric**: 
$$e^I = \frac{\sum_m |r^I(P_m)|}{N}$$
 to avoid outliers (spike reproduction).



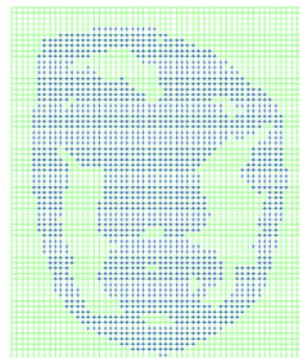
## Costruzione di un secondo livello gerarchico



At layer 1:  
$$e_k^I(P_k) = \frac{\sum_m |r^I(P_m) - S^I(P_m)|}{N_k}$$
 Viene calcolato un valore di "errore" per ogni grid crossing, k  
Output of layer #2



Layer #2



Nei livelli successivi si approssima il residuo = si aggiunge dettaglio.



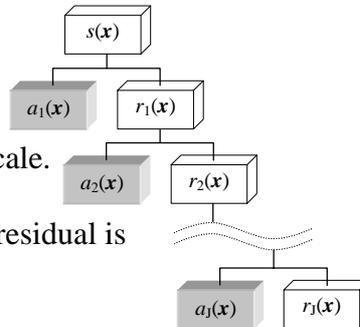
## The HRBF configuration algorithm



- 1) Start with a large scale and reconstruct the surface.
- 2) Compute the residual.

Do:

- 2) Create a denser grid (lower scale).
- 3) Compute the residual at this lower scale.
- 4) Evaluate the residual.
- 5) Insert the Gaussians only where the residual is under-threshold.



Until:

- 6) Until the residual goes under threshold.

Reconstruction is obtained by adding the approximations:  $s(\mathbf{x}) = \sum_j a_j(\mathbf{x})$

Each approximation is obtained as:  $a_j(\mathbf{P}) = \sum_k s_{kj} G(\mathbf{P} - \mathbf{P}_{k_j} | \sigma_j)$

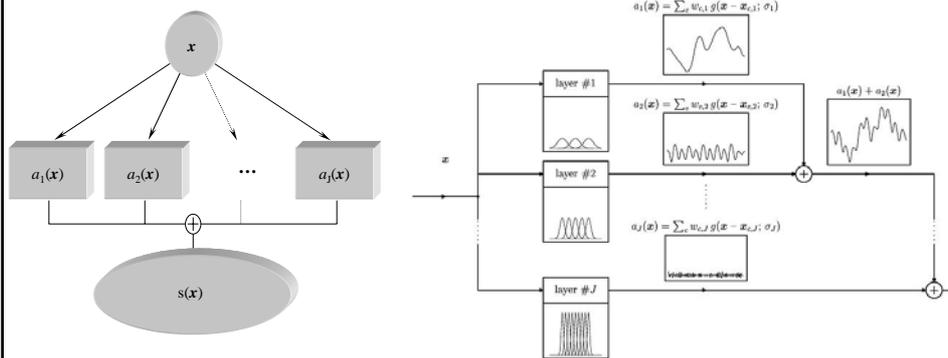
A.A. 2003-2004

39/49

<http://homes.dsi.unimi.it/~borgnese>



## Sintesi della superficie



- Diversi strati (reti RBF).
- Ciascuno strato opera ad una certa scala.
- Gli strati non sono completi ma allocano Gaussiane solo dove l'errore locale è sopra-soglia.

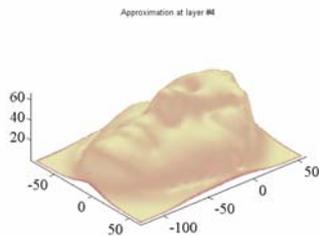
A.A. 2003-2004

40/49

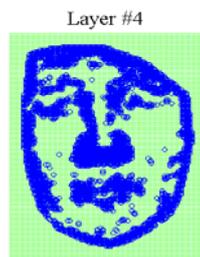
<http://homes.dsi.unimi.it/~borgnese>



## HRBF Networks



Noise



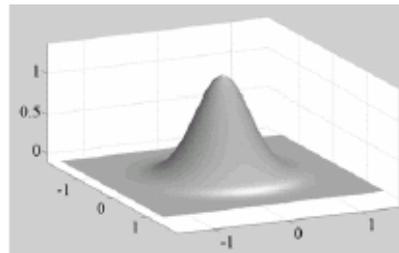
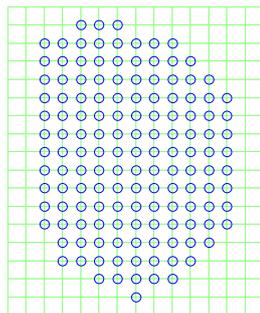
A.A. 2003-2004

41/49

<http://homes.dsi.unimi.it/~borgnese>



## Sfruttamento della quasi-localita'



Definisco un campo recettivo, RF, per ogni Gaussiana (grid crossing). E.g. Il quadrato centrato nella Gaussiana e di dimensione  $2\Delta P$ . I campi recettivi sono parzialmente sovrapposti.

**Operazioni quasi-locali sui punti all'interno dei campi recettivi:**

- Calcolo dell'altezza della funzione nell'incrocio.
- Calcolo dell'errore locale.

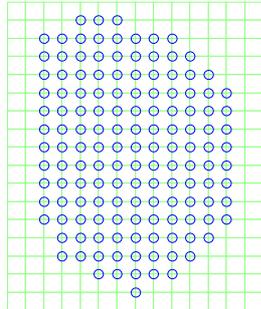
A.A. 2003-2004

42/49

<http://homes.dsi.unimi.it/~borgnese>



## Where locality is interesting?



$$S(P) = \sum_{k=1}^M s_k G(P - P_k | \sigma)$$

Estimate of the Surface in the grid crossings:

$$s_k = \frac{\sum_m s_m(P_m) f(|P_m - P_k|)}{\sum_m f(|P_m - P_k|)}$$

Computation of the error from the residual for each grid crossing:

$$e_k = \frac{\sum_m |r(P_m)|}{N_k}$$

Questi calcoli possono essere ristretti solamente i punti all'interno del campo recettivo della k-esima Gaussiana.

A.A. 2003-2004

43/49

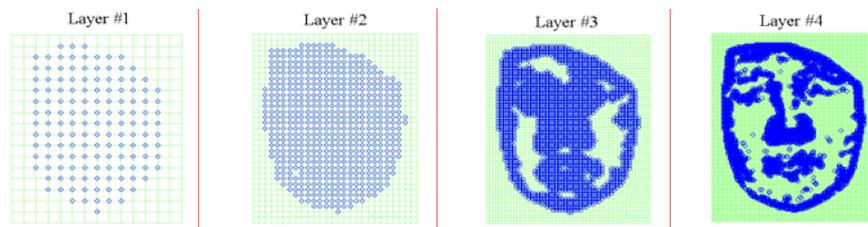
<http://homes.dsi.unimi.it/~borgese>



## Risultato: sparse approximation



Grids do not need to be complete, but Gaussians can be inserted only in those crossings where the error is over threshold.



We obtain a sparse approximation (less dense units, where less dense sampling and larger details).

Local control of the reconstruction error. Error globally under threshold.

A.A. 2003-2004

44/49

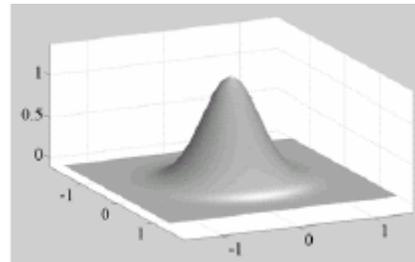
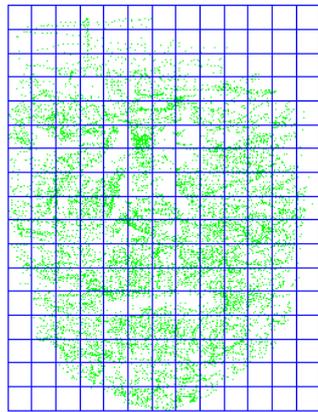
<http://homes.dsi.unimi.it/~borgese>



## Representation of data locality



Approximate the Receptive field with squares (Manhattan distance).



→ Quad-tree subdivision

Data are partitioned into parallelepipeds supported by the grid at the lowest (largest) scale.

A.A. 2003-2004

45/49

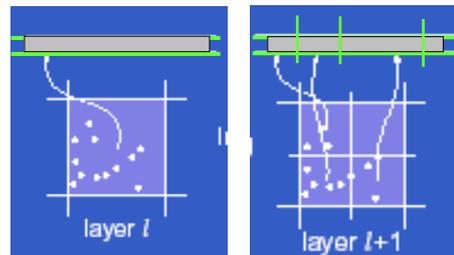
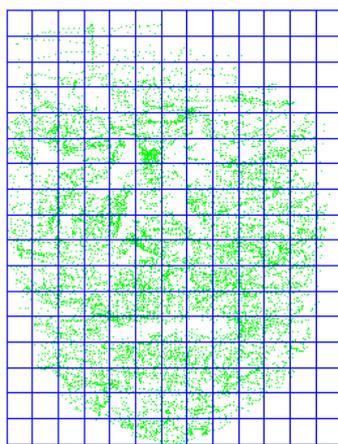
<http://homes.dsi.unimi.it/~borgese>



## Efficient data storage



Quad-tree subdivision by means of in-place sorting.



Block of range data associated to the grid square

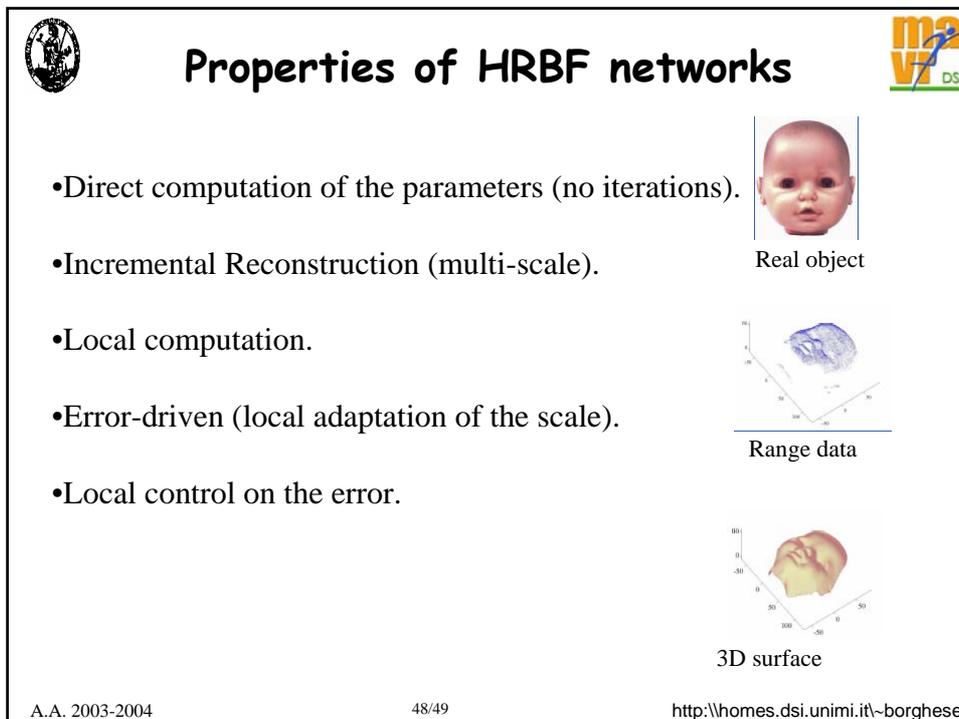
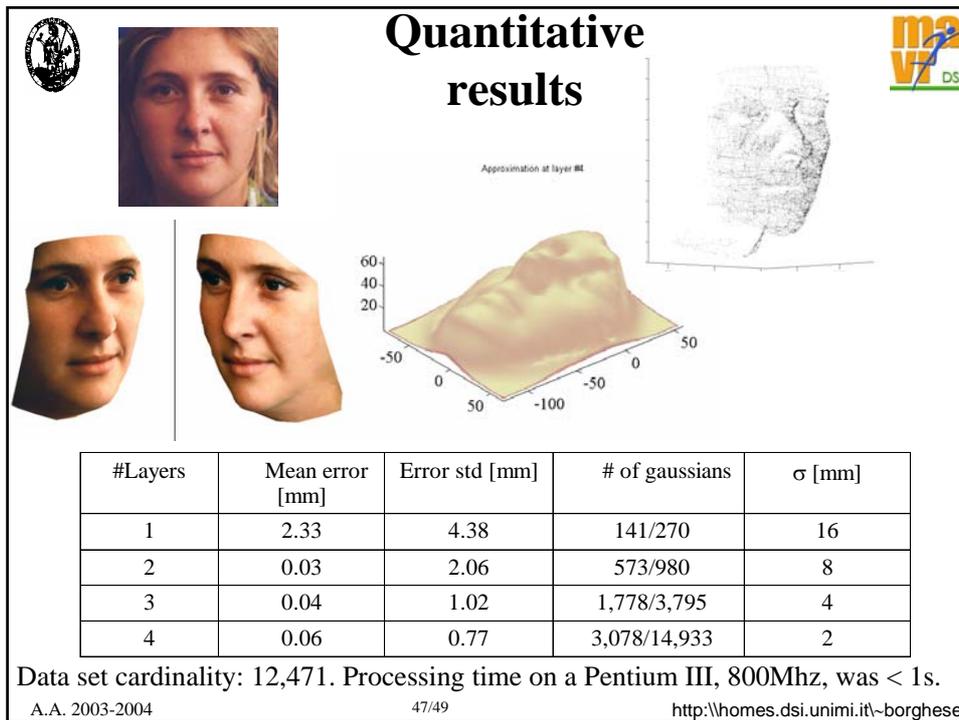
Data can be retrieved directly.

At every layer, only data inside the receptive field have to be sorted.

A.A. 2003-2004

46/49

<http://homes.dsi.unimi.it/~borgese>





## Sommario



Le Radial Basis Function Networks.

Modalità di apprendimento nelle reti RBF.

Apprendimento Ibrido.

Apprendimento ispirato alla teoria dei segnali.

Apprendimento multi-scala.