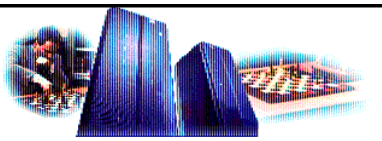
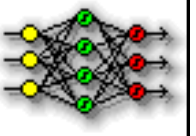




Sistemi Intelligenti
Reti convoluzionali e On-line learning

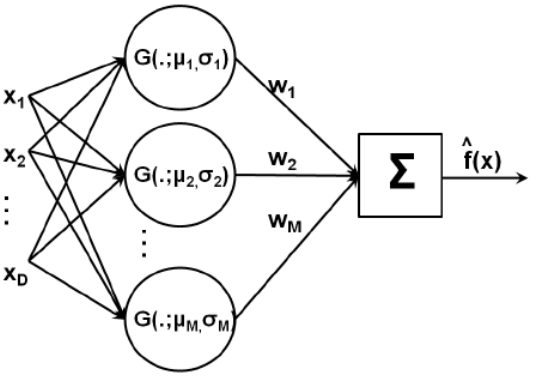
Alberto Borghese
Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
Dipartimento di Informatica
Alberto.borghese@unimi.it

Riassunto

- Reti convoluzionali
- Modelli multi-scala
- Valutazione di un modello
- Modelli multi-scala on-line

Convolutional layer

Connessionism. Simple processing units combined with simple operations to create complex functions.



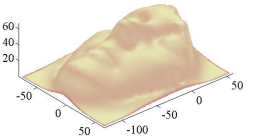
When G(.) are not equally spaced -> RBF Networks. Perceptron

Costruzione di modelli continui

- **Le fuzioni di base sono equispaziate (posizionate su una griglia) e tutte con gli stessi parametri (in questo caso σ).**
- Struttura di supporto semplificata (griglia – funzioni di base - Il concetto di Base di uno spazio funzionale in analisi matematica è definito mediante certe proprietà di approssimazione che qui non consideriamo, consideriamo solo l'idea intuitiva).
- Il concetto di base è simile a quello dei "replicating kernels" in Machine Learning.

$$z(p(x,y)) = \sum_i w_i G(p, p_i; \sigma)$$

Approssimazione continua con un numero di elementi finito



Combinazione lineare di funzioni di base

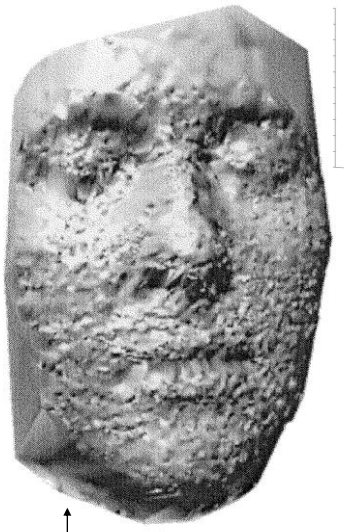
Da calcolare

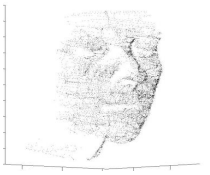
Funzione di base (fissate)

- Δp_i, σ dati

In generale, mappatura da $R^D > R$

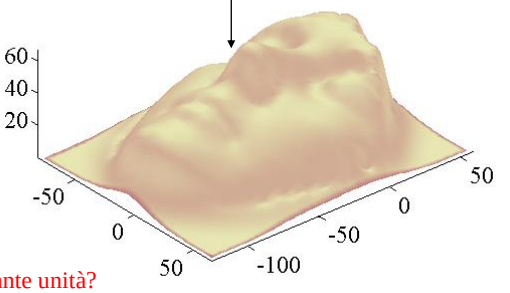
Applicazione: scanner 3D





z = f(x,y | w) – alto-rilievo

Buona approssimazione



Quante unità?
Quanto ampie (valore di σ)?

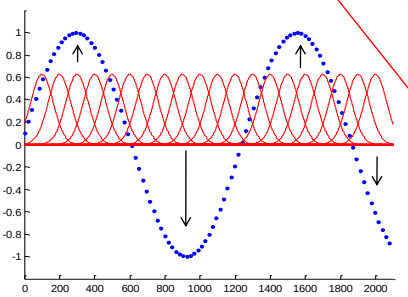
Problema dell'overfitting dovuto a sovra-parametrizzazione

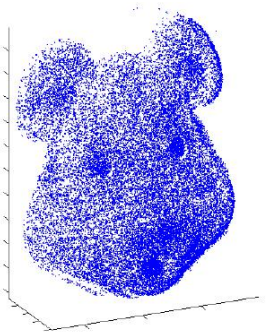
Advantages and issues

w_k

$$\hat{f}(x) = \sum_{k=1}^{N} f_k G(x; x_k, \sigma)\Delta x = \frac{\Delta x}{\sqrt{\pi}\,\sigma} \sum_{k=1}^{N} f_k e^{-\frac{(x-x_k)^2}{\sigma^2}}$$

Filters interpolates data
(introduce generalization)
and reduce noise but...





Height of the surface on a grid crossing, f_k, is not known in general

Points clouds (data are not equally spaced)

Gridding

$$z = \hat{f}(x) = \sum_{k=1}^{N} f_k\, G(x; x_k, \sigma)\Delta x = \frac{\Delta x}{\sqrt{\pi}\,\sigma} \sum_{k=1}^{N} f_k e^{-\frac{(x-x_k)^2}{\sigma^2}} = \sum_{k=1}^{N} w_k e^{-\frac{(x-x_k)^2}{\sigma^2}}$$

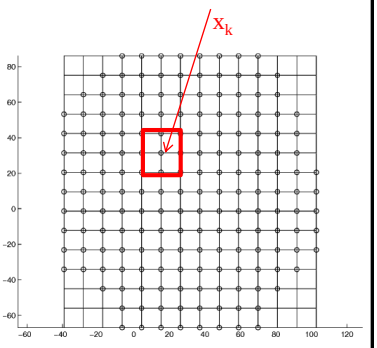
Gaussians equally spaced and distributed over a grid. How can we determine their associated weight, w_k, from points clouds?

Local estimators. Nadaraya Watson estimators. *Lazy learning (cf. K-NN)*

$$f(x_k) = \frac{\sum_i z_i K_\sigma(x_i, x_k)}{\sum_i K_\sigma(x_i, x_k)} = \frac{\sum_i z_i e^{\frac{(||x_i - x_k||)^2}{\sigma^2}}}{\sum_i e^{\frac{(||x_i - x_k||)^2}{\sigma^2}}}$$

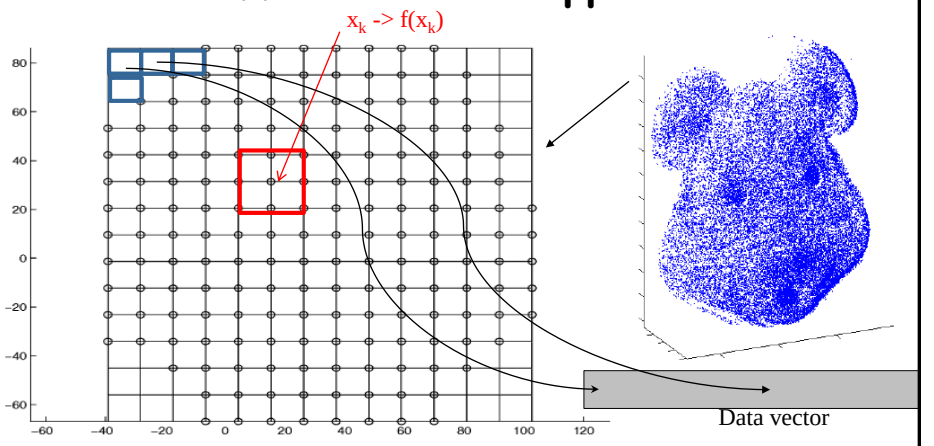
$K_\sigma(.)$ Gaussiana

Troncamento dell'ampiezza del "campo recettivo"
Dati $\{z_i=f(x_i), x_i\}$ all'interno di un numero di celle vicine.



Parzen-windows estimator

Efficient data support



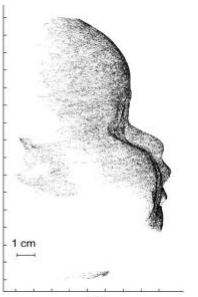
Data comes into a linear vector and are sorted into **quads** each centered in each Gaussian center ➔ Each quad points to a position in the data vector ➔ **in-place ordering inside the vector**, by data position.
The receptive field of x_k is constituted of 4 quads and the data considered for estimating w_k are those inside those quads.

Example: 3D scanner

Which scale?

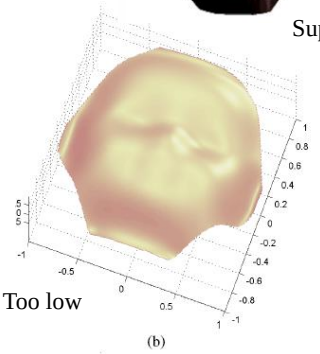
Too high

(a)



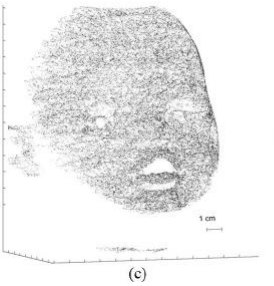
(b)

Supervised learning



Too low

(b)



(c)

(d)

Riassunto

- Reti convoluzionali
- Modelli multi-scala
- Valutazione di un modello
- Modelli multi-scala on-line

Filters and bases

$$\hat{f}(x) = f_i * G(x - x_{k_i}; \sigma) = \sum_{i=1}^{N} w_i G(x - x_{k_i}; \sigma)$$

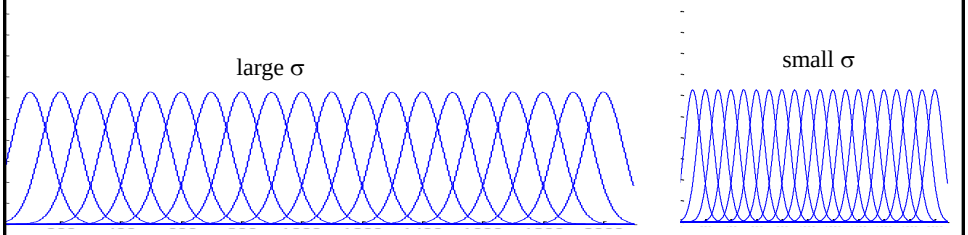
con funzioni di base normalizzate:

Riesz basis, the approximation space is characterized by the scale of the basis that determines the amplitude of the space.

A sequence of spaces can be defined according to σ:

σ_0 -> V_0; σ_1 -> V_1; σ_2 -> V_2….

The number of representable functions increases.



Incremental strategy

- Acquire more data in the more complex areas, less smooth, higher frequency.
- Acquire less data in the less complex areas, more smooth, lower frequency.

$$\hat{f}(x) = \sum_{k=1}^{N} f_k G(x; x_k, \sigma)\Delta x = \frac{\Delta x}{\sqrt{\pi}\,\sigma} \sum_{k=1}^{N} f_k e^{-\frac{(x-x_k)^2}{\sigma^2}}$$

- Can we use a single Δx? ➔ A single value of σ?

- Large σ, large spacing, few Gaussians, little detail.
- Small σ, tight spacing, many Gaussians, lots of details.

Why not using the highest σ?

- Not known
- Not enough data inside the receptive field of all the Gaussians (more data where little details concentrate).

Incremental approximation with local adaptation of the scale σ.

Resolution, Δx and σ

- Low resolution, large distance,
- High resolution, small distance, $\Delta x > 2\nu_{Max}$

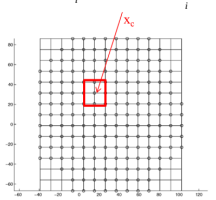
σ determines the amount of overlap. It determines also the frequency content of the Gaussian G(.).

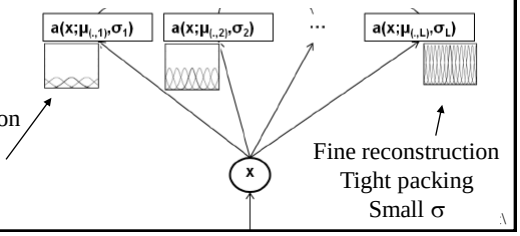
Once σ (or Δx is defined) the grid is also defined.

The height of each Gaussian, $\hat{f}(x_c)$, can be computed.

$$\hat{f}(x_c) = \frac{\sum_i y_i K_\sigma(x_i, x_c)}{\sum_i K_\sigma(x_i, x_c)} = \frac{\sum_i y_i e^{-\frac{\|x_i - x_c\|^2}{\sigma^2}}}{\sum_i e^{-\frac{\|x_i - x_c\|^2}{\sigma^2}}}$$

$$\hat{f}(x) = \sum_{k=1}^{N} f_k G(x; x_k, \sigma)\Delta x = \frac{\Delta x}{\sqrt{\pi}\,\sigma} \sum_{k=1}^{N} f_k e^{-\frac{(x-x_k)^2}{\sigma^2}}$$





Coarse reconstruction
Loose packing
High σ

Fine reconstruction
Tight packing
Small σ

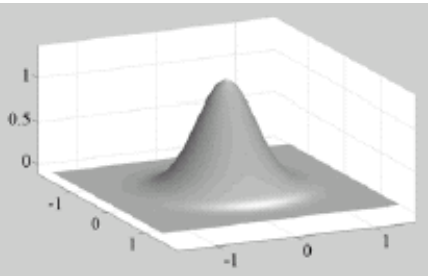
Starting from low resolution

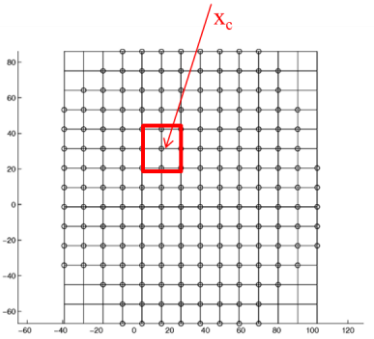
How many points to consider? The Gaussian has infinite support.

Apply local estimator to the data points in the neighbourhood of a grid crossing (Gaussian center) to compute $f_k = \hat{f}(x_{ck})$.

Quad support makes this operation easy.

$$\hat{f}(x) = \sum_{k=1}^{N} f_k G(x; x_k, \sigma)\Delta x$$



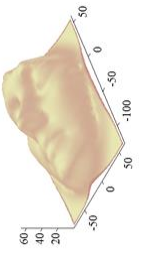


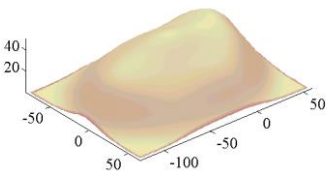
We can obtain a «poor» reconstruction

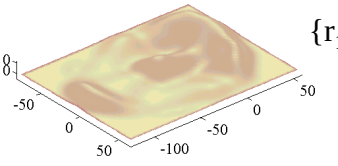
Little detail. Large scale. But it is a start. It can be seen as a modified support for successive approximations.

Regular grid with few Gaussians largely spaced with large σ

What can be done?





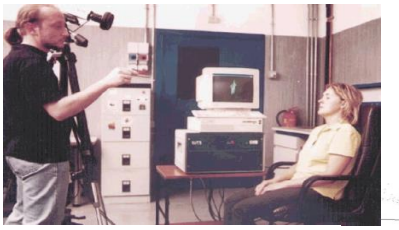


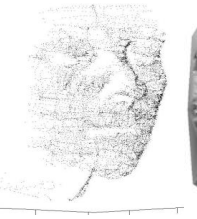
$\{r_1(\mathbf{x})\}$

We evaluate the **residual** for **each data point**: $r_1 = dist\left(y_m, \hat{f}(x_m)\right)$

E.g.: $r_1 = \left(y_m - \hat{f}(x_m)\right)^2$ $r_1 = \left|y_m - \hat{f}(x_m)\right|$

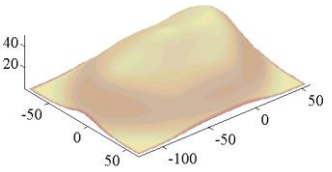
Is the residual adequate?

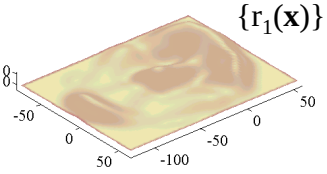


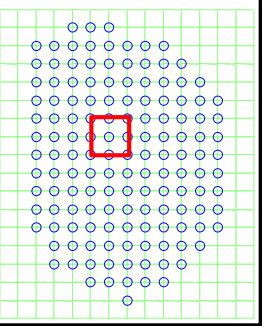


Approximation at layer #1

$\{r_1(\mathbf{x})\}$



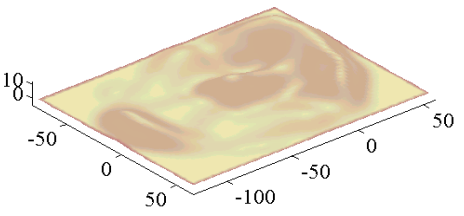




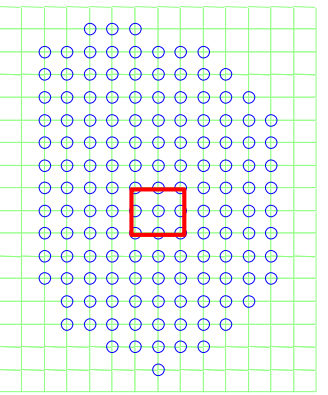
For each Gaussian t**he integral of the residual inside the "receptive field"** of the Gaussian, is assumed as local approximation error associated to it. , is computed inside its "receptive field":

$$R(x_c) = \frac{\sum_m r_m}{N_k}$$

How can we evaluate the local adequacy of the reconstruction?



$$R(x_c) = \frac{\sum_m r_m}{N_k}$$



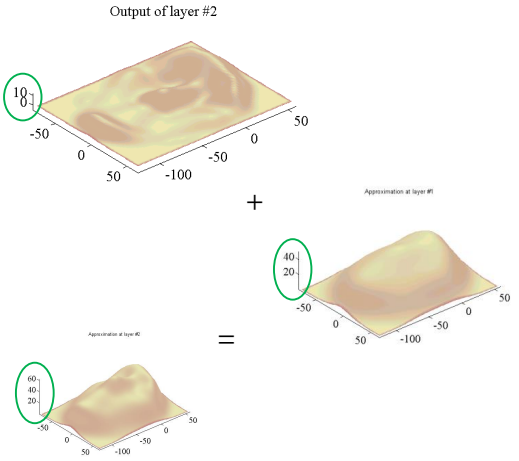
We compare the local residual with a threshold derived from:
- Degree of approximation
- Noise: RMS.

We aim to have a **residual error** that is **uniformly** under a given threshold.

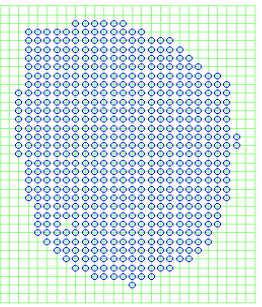
Layer 2

Input are the residuals of previous layer, $r_{1,m}$= $\left|y_m - \hat{f}_1(x_m)\right|$

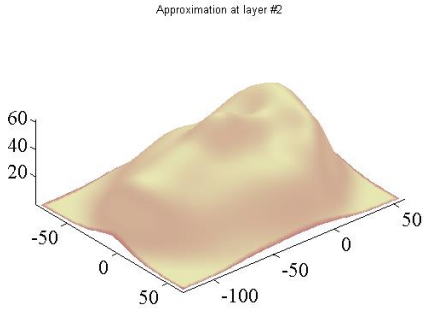
Output is a layer that approximates $r_{1,m}$: $f_2(x_m) \rightarrow r_{1,m}$



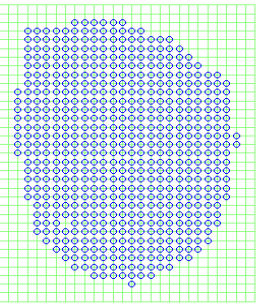
Layer #2



Evaluation of Layer 2



Layer #2



$$\widehat{f(x)}^{II} = \sum_{j=1}^{2} \sum_{k} f_{j,k} G\left(x - x_{j,k} \middle| \sigma_j\right)$$

First approximation + first residual

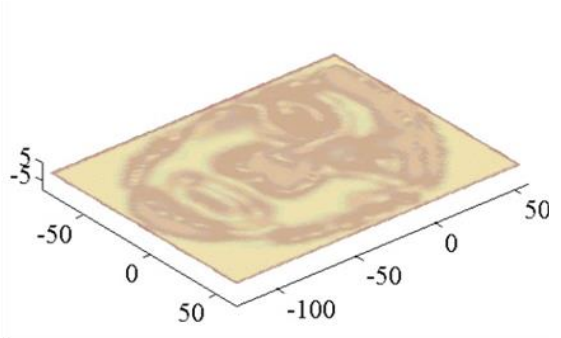
$$R(x_c) = \frac{\sum_m \left| y_m - \widehat{f(x)}^{II} \right|}{N_k}$$

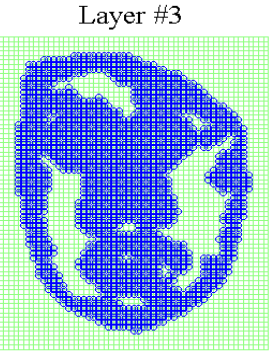
More packed Gaussians. More details. But…
There should be enough points to have a reliable local estimate of Gaussian height.

Layer 3

Input are the residuals of previous layer, $r_{2,m} = \left| y_m - \widehat{f(x)}^{II} \right|$

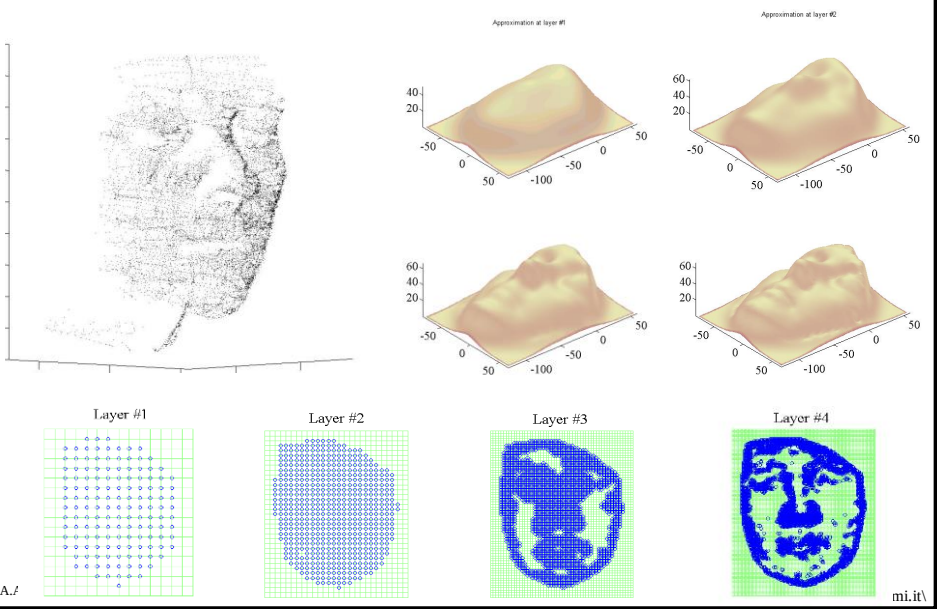
Output is a layer that approximates $r_{2,m}$: $f^{III}(x_m) \rightarrow r_{2,m}$



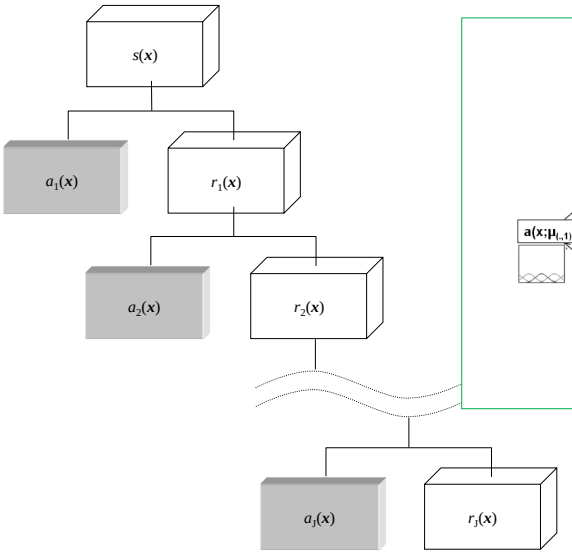


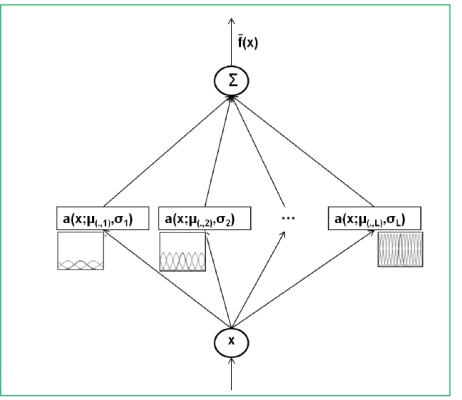
Sparse approximation in the third layer with $\sigma = \sigma_3$.

Applicazione alla regressione



Hierarchy construction

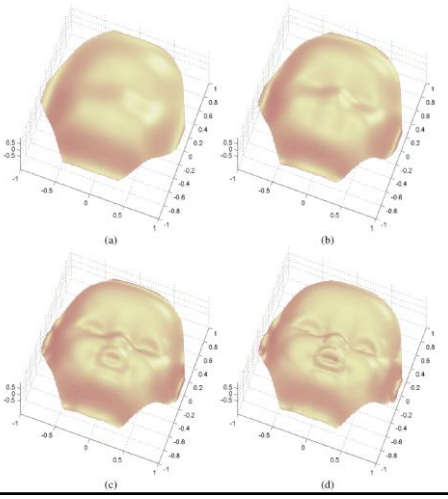


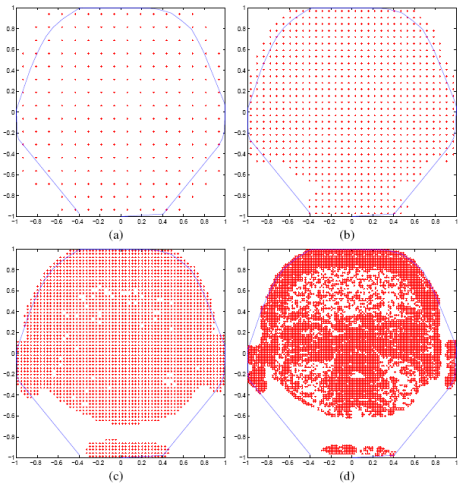


It generates a set of hierarchical layers

How to operate on large sets of data?

Recursive splitting of the data inside each quad -> local re-ordering of the data.





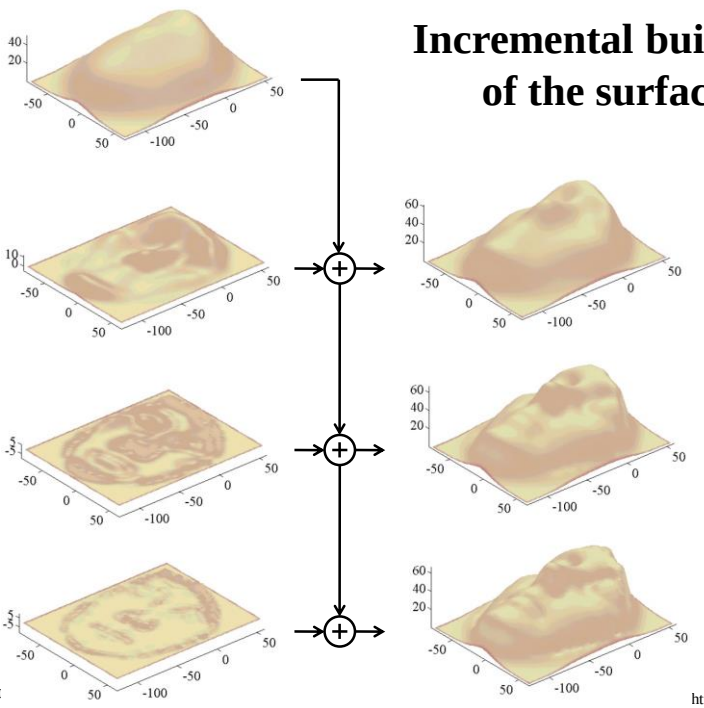
Characteristics of HRBF networks

- Local operations.
- Hierarchy of approximations.
- Local adaptation of the scale (Not fully occupied layers)
- Adaptive allocation of the resources
- Uniform convergence to a residual error
- No hyper-parameters have to be set

- Residual bias is recovered in the next layers.
- Relatively dense data sets are required to obtain a robust local estimate.
- Riesz basis, with a high degree of redundancy between the coefficients. The angle between two approximating spaces is not 90, but it is considerably smaller

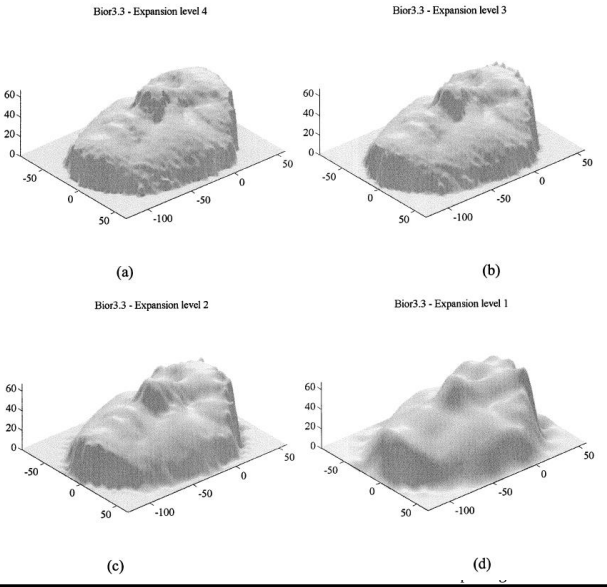
$$\cos\,\alpha_j = \sup_{f(\cdot)\in V_j, h(\cdot)\in V_{j+1}} \frac{\langle f(\cdot), h(\cdot)\rangle}{\|f(\cdot)\|_2 \|h(\cdot)\|_2} = \cos\alpha_{j-1}.$$

Incremental building of the surface



Pyramidal reconstruction

- Decrease scale from coarse (level 1) to fine (level 4).
- Which is the adequate scale?
- Which model is the closest to the true model?



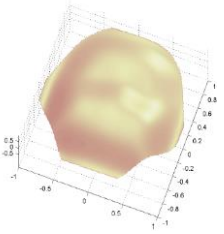
Beyond Wavelet

Portilla et al., Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain, 2003.

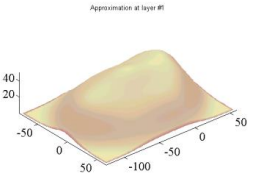
Coefficients reduction through a model of the noise.

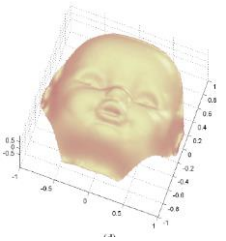
RBF and Wavelet have excellent for CUDA implementation as all bases with limited support.

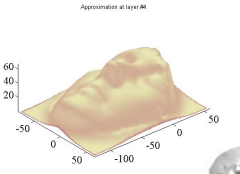
Underfitting e overfitting



Quanti parametri?







Quante unità?

Come valutiamo?

Riassunto

- Reti convoluzionali
- Modelli multi-scala
- Valutazione di un modello
- Modelli multi-scala on-line

How to classify the error introduced by a model?

Does it cover the input domain (in all dimensions – **dimensionality discovery**)?

This is not enough to obtain a good model!! Is the model good enough?

Is a model that produces 0 error on the data a good model?

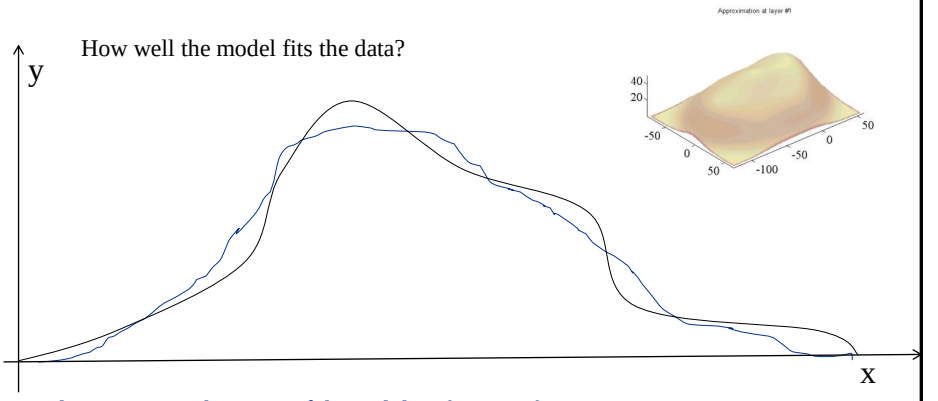
Total quality is an inspection system that detects all the defects. No defected objects are left behind.

The model should be properly tuned to the data. Errors can be classified in:
- Bias
- Variability

Relationship between number of parameters and bias/variability

Bias

How well the model fits the data?



Blue represents the curve of the real data $\{x_{true}, y_{true}\}$
Black represents the curve produce by the model: $y = f(x)$

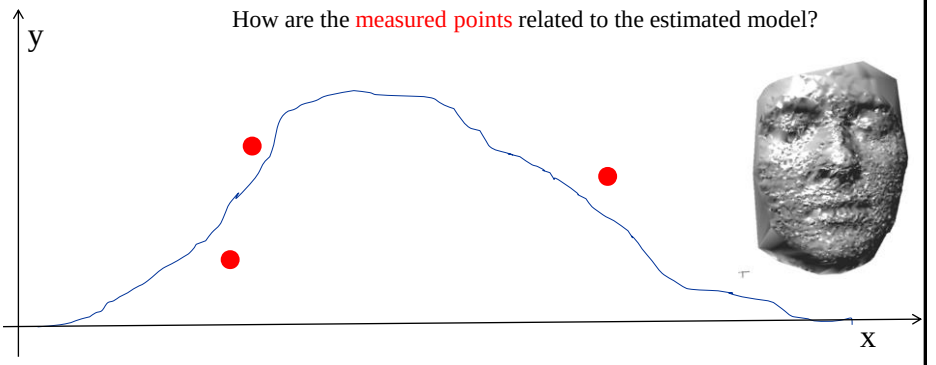
How good is the model?

Model output

We can compute the error: $dist(y_{true}, y=f(x_{true}))$, for instance Euclidean distance.
As such it is the **bias of the model**.

Variability

How are the measured points related to the estimated model?

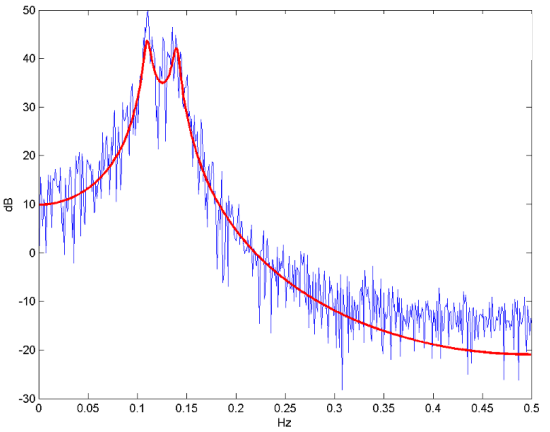


Given $P_m(x_m,y_m)$ and y = f*(x), the true data behavior, the error is:
dist(y_m,y=f*(x_m)), for instance Euclidean distance.
As such **variability** is the **measurement error**.

If variability goes to zero, bias increases and overfitting arises (model fits data and the noise too).
In a good model, variability tends to the statistics of the measurement noise
(cf. regularization parameter setting).

Variability

How are the measured points related to the estimated model?



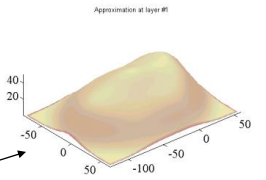
Given $P_m(x_m,y_{mes})$ and y = f(x), the true data behavior, the error is measured as: dist(y_m,f(x_m)), for instance Euclidean distance. It is associated to measurement error.

If variability goes to zero, bias increases and overfitting arises.
In a good model, variability tends to the statistics of the data noise.

Bias and variability

Bias and variability trade-off

Bias is the distance of the model curve from the true curve, **that is unknown**. It is the model error.



High bias

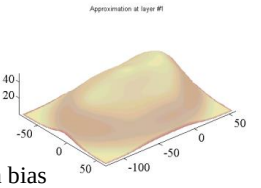
Variability is the distance of the true curve, **that is unknown**, from the **measured data.** It is the measurement error.

High variability

Parametric model
Error vs number of parameters

Bias and variability trade-off

Bias is the distance of the model curve from the true curve, **that is unknown**. It is the model error.



High bias
Low number of parameters

If we have **few parameters**, we can reproduce only the outline of the data (under-fitting -> bias).

Variability is the distance of the true curve, **that is unknown**, from the **measured data.** It is the measurement error.

High variability
High number of parameters

If we have many parameters we can reproduce the fastest variations, that are due to noise (over-fitting ->variability).

Cross validation

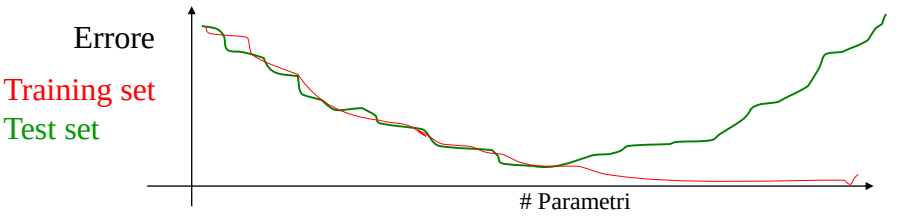
- I dati vengono suddivisi in due sotto-insiemi: training (costruzione del modello) e test.
- Errore sull'insieme di training = Errore sull'insieme di test.
- L'insieme di test è rappresentativo dei dati che potranno essere presentati in futuro al modello.

Si vuole evitare che il modello si specializzi troppo sui pattern di training e non sia in grado di interpolare correttamente su altri dati (e.g. dati di test).

- La procedura viene ripetuta k volte su sottoinsiemi diversi estratti a caso: k-fold cross-validation.

Scelta empirica del numero di parametri

*Il numero di parametri viene aumentato fino a quando **entrambi** gli errori diminuiscono.*



Parto da un modello con pochi parametri e aumento il numero di parametri o viceversa.

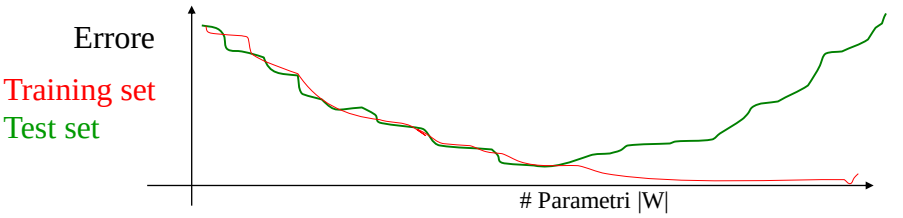
Approccio simile a quello utilizzato nel criterio di discrepanza.

Scelta teorica

Quale funzione costo minimizzo? Come posso inserire l'informazione di complessità del modello nella funzione costo?

Penalizzo i modelli con tanti parametri. Aggiungo nel calcolo della distanza tra dati e modello (funzione costo) un termine che cresce con il numero dei parametri -> Regularization with Reproducible Hilbert Kernels as regularizers.

$$w = \underset{w}{\operatorname{argmin}} \left(\sum_i \|f(x_m) - y_m\|^2 + \lambda g(\lfloor W \rfloor) \right)$$

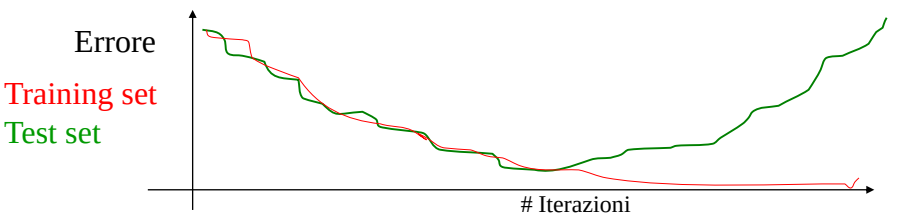


Semi-convergenza

Ipotesi: ho un insieme di parametri sufficientemente ampio.

Semi-convergenza: non porto l'algoritmo fino alla convergenza nel punto di ottimo ma arresto le iterazioni prima ("early stopping")

Il modello non sarà perfettamente aderente ai dati, ma il residuo sarà tendenzialmente l'errore di misura.



Riassunto

- Reti convoluzionali
- Modelli multi-scala
- Valutazione di un modello
- Modelli multi-scala on-line

Dati e apprendimento

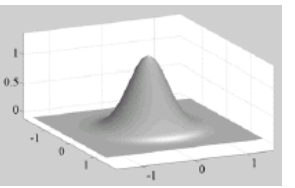
- Apprendimento batch. Ho a disposizione tutti i dati o lotti (batch) di dati. Costruzione del modello una-tantum.
- Apprendimento on-line. Ho a disposizione un dato alla volta. Costruzione del modello incrementale.

Active sampling. Dirigo l'attenzione in certe regioni dello spazio dei dati e lì raccolgo più dati -> modulo di pianificazione dell'acquisizione dei dati.

On-line single layer

- Each new point, $\{x_k,y_k\}$, contributes to the estimate of the function height inside the receptive fields of the associated Gaussians.
- The estimate of $f(x_c)$ has to be recomputed:

$$f(x_c) = \frac{\sum_{i=1}^{N_c}\left(f(x_i)e^{\frac{-(x_i-x_c)^2}{\sigma^2}}\right)}{\sum_{i=1}^{N_c}\left(e^{\frac{-(x_i-x_c)^2}{\sigma^2}}\right)}$$



- On-line strategy: numerator and denominator are updated separately.

On-line estimate of $f(x_c)$

$$Num(x_c) = \sum_{i=1}^{N_c}\left(f(x_i)e^{\frac{-(x_i-x_c)^2}{\sigma^2}}\right)+\left(f(x_{new})e^{\frac{-(x_{new}-x_c)^2}{\sigma^2}}\right)=$$

$$Num'(x_c) = \sum_{i=1}^{N_c+1}\left(f(x_i)e^{\frac{-(x_i-x_c)^2}{\sigma^2}}\right)$$

$$Den(x_c) = \sum_{i=1}^{N_c}\left(e^{\frac{-(x_i-x_c)^2}{\sigma^2}}\right)+\left(e^{\frac{-(x_{new}-x_c)^2}{\sigma^2}}\right)=$$

$$Den'(x_c) = \sum_{i=1}^{N_c+1}\left(e^{\frac{-(x_i-x_c)^2}{\sigma^2}}\right)$$

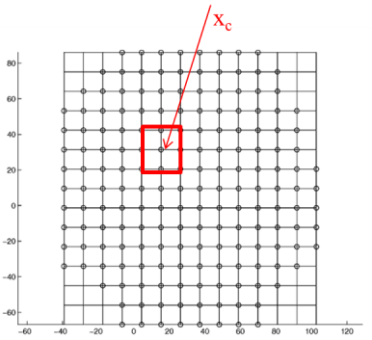
$$f(x_c) = \frac{Num'(x_c)}{Den'(x_c)}$$

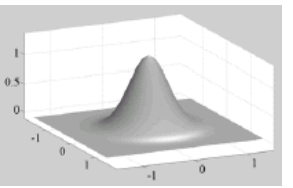
Few computations are required for any number of point: recursive estimate of $f(x_c)$

For each new point a new term is added and the ratio is recomputed only for the Gaussians whose receptive field contains that point.

For Nc -> ∞

$$\lim_{Nc\to\infty} \frac{Num'(x_c)}{Den'(x_c)} = E(f(x_c))$$

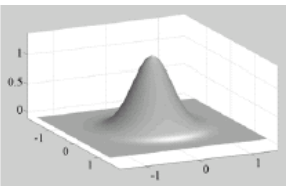


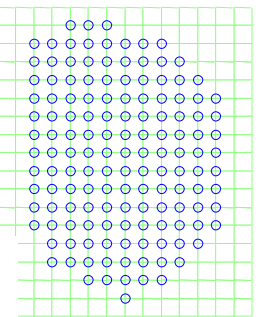


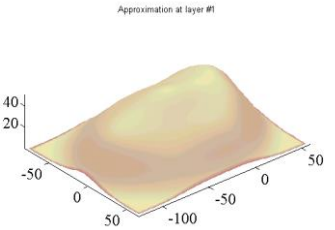
How many points are required to get a good estimate of Num'(x_c) e Den'(x_c)?
Experimental answer.

Is it sufficient to obtain a good reconstruction?

First layer



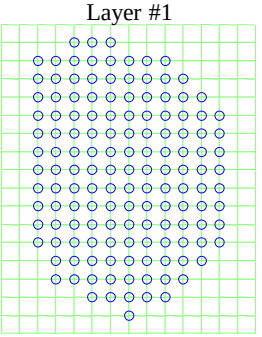




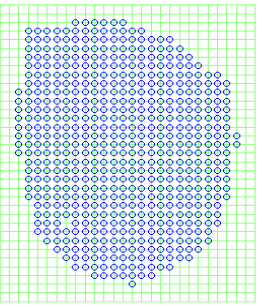
Asymptotically, we cannot obtain anything better than this.
Few Gaussians, large scale.

How to move to next layer?

Layer #1



Layer #2



A reliable estimate of f_c on Layer 1 ~~is~~ a reliable estimate of f_c on Layer 2.

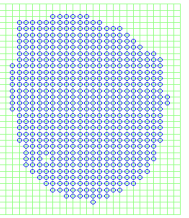
The points that belong to the receptive field of fc on Layer 1 are split over the different quads in Layer 2.

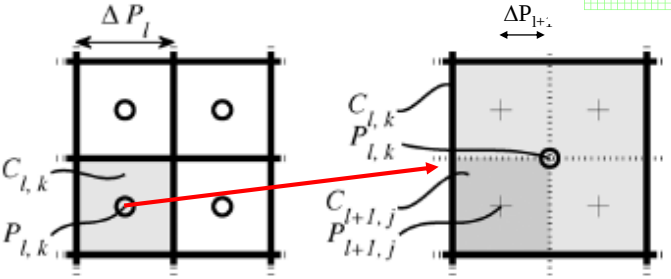
When shall we start to estimate f_c in Layer 2?

Local operations

- Local split of each quad is achieved when:
  - Residual is higher than threshold
  - **Enough points have been sampled**
- **4 new Gaussians are generated at the higher level**

Layer #2





1 quad

4 quads

On-line estimate of $f(x_c)$ on new layer

$$Num'(x_c) = \sum_{i=1}^{N_c+1} \left(r(x_i) e^{\frac{-(x_i - x_c)^2}{\sigma^2}} \right)$$

$$Den'(x_c) = \sum_{i=1}^{N_c+1} \left(e^{\frac{-(x_i - x_c)^2}{\sigma^2}} \right)$$

This requires that the points $\{x_i\}$ inside the receptive field of the 4 Gaussians created are extracted from the $\{x_i\}$ of the Guassian of the current layer that we have split.

How?

In-place ordering of the $\{x_i\}$ associated to the quad of the Gaussian of the current layer such that they are distributed in the quads of the four Gaussians created.

The approximation of the residual, $a(x_c)$ is initialized with few points. Computation of Num'(x_c) and Den'(x_c) for the four new Gaussians requires little effort.

$$r(x_c) = \frac{Num'(x_c)}{Den'(x_c)}$$

On-line version

- Data do not arrive all together (batch)
- One data at a time.
- **Growing while scanning**

2 min video

Riassunto

- Reti convoluzionali
- Modelli multi-scala
- Valutazione di un modello
- Modelli multi-scala on-line