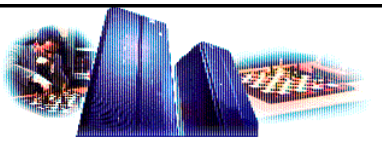
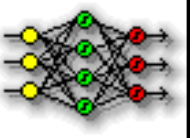




Sistemi Intelligenti
Supervised learning

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

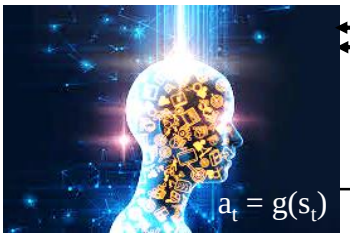
Riassunto

- Supervised learning
- Predictive regression
- Regressione multi-scala

Apprendimento con rinforzo

Agent

What the world is like now (internal representation)?

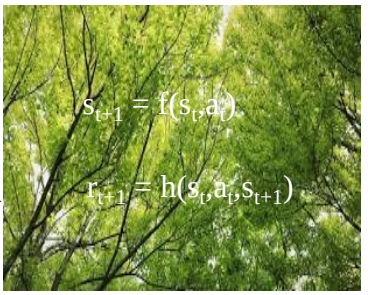


What action should I choose now? (policy)

Which is the value of my action (value function)?

s_t s_{t+1} r_{t+1} a_t

Environment



L'ambiente fornisce un'informazione **puntuale**, di tipo **qualitativo**, ad esempio success or fail.

Apprendimento non supervisionato

Agent

What is «s» similar to?

s

X No output

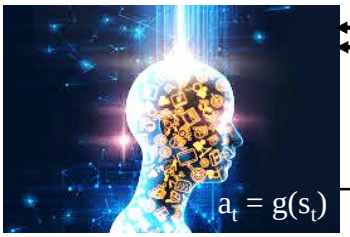
Environment

No feed-back/information from the environment

Apprendimento Supervisionato

Agent

What the world is like now (internal representation)?



What action should I choose now? (policy)

Which is the suggested action?

s_{t+1}

$\mathbf{a^D_t}$

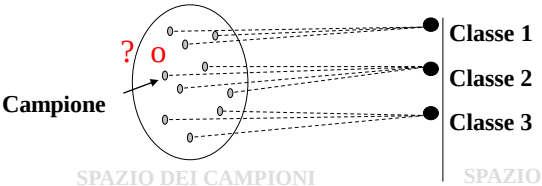
$a_t = g(s_t)$

Environment

L'ambiente fornisce (non sempre) un'informazione **puntuale**, di tipo **quantitativo,** tipicamente l'azione desiderata, $\mathbf{a_t^D}$, per quello stato $\mathbf{s_t}$. **Learning with a teacher => generalizzazione**

Classificazione

Mappatura dello spazio dei campioni nello spazio delle classi.



SPAZIO DEI CAMPIONI / DELLE FEATURE (CARATTERISTICHE)

SPAZIO DELLE CLASSI (identificate da un'etichetta)

Classifico ("aiuto alla clusterizzazione")

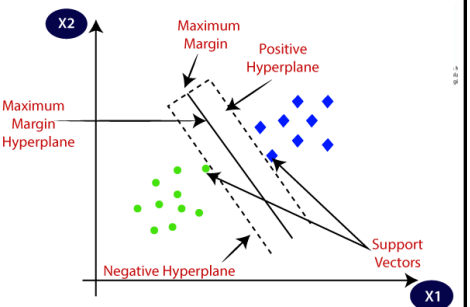
Uscita del processo di classificazione è un insieme discreto e finito di etichette.

Le classi sono la base per potere applicare la logica…
E' un processo di clusterizzazione guidato (dal teacher).

Classificazione algoritmi

- **Boosting**. Si utilizza un insieme di classificatori binari, dove ciascun classificatore lavora su una **singola feature** e decide si'/no l'appartenenza del dato a una classe. La classificazione avviene prendendo la **maggioranza** del voto dei classificatory semplici.
- **Reti neurali**. Approccio black-box generale.
- **Support Vector Machine**. **Partizionano lo spazio dei dati** (o delle feature) calcolando la linea di separazione che massimizza il margine, cioè che passa più lontana dai punti delle due classi. La linea può essere una spezzata (lineare) oppure una curva (non-lineare).

➔ Corso di "Metodi di apprendimento"



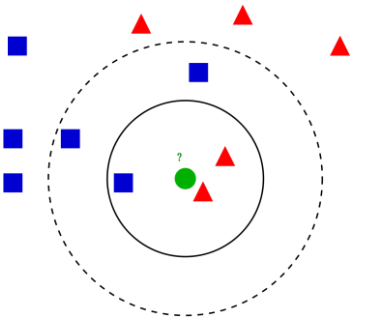
Algoritmo K-NN

K-Nearest Neighbour

Definisco una **misura di vicinanza** (campo recettivo).

Per ogni input, considero i K dati più vicini per i quali è già stata prescritta una classificazione.

Scelgo l'azione combinando questi K dati (max, soft-max, combinazione lineare o non-lineare, pesata con la distana o con il campo recettivo).



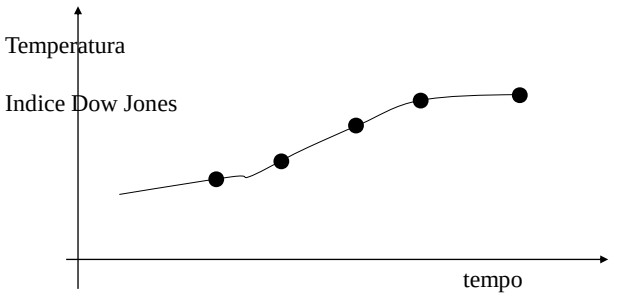
Consideriamo il punto verde e vogliamo classificarlo blu o rosso.
Consideriamo la distanza Euclidea come misura di vicinanza.
Consideriamo la funzione maggioranza per la decisione.

Se consideriamo il dato più vicino -> rosso (NN)
Se consideriamo i 2 dati più vicini -> rosso (2-NN)
Se consideriamo I 5 dati più vicini -> blu (5-NN).

Riassunto

- Supervised learning
- Predictive regression
- Regressione multi-scala

Regressione predittiva



Quanto vale la temperature (indice Dow Jones) all'istante successivo (futuro)?

Controllo della portata di un condizionatore in funzione della temperatura. "Imparo" una funzione continua a partire da alcuni campioni: devo imparare ad **predire** (regressione = predictive learning).

Applicazioni alle serie temporali: per esempio andamento della borsa, previsioni del tempo,… **(estrapolazione)**.
Applicazioni alla ricostruizione di mani-fold **(interpolazione)**

Modello predittivo

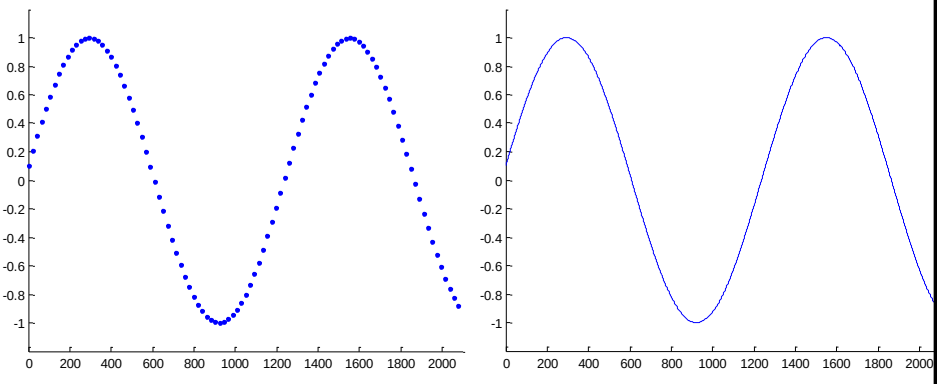
a = π(s | w) π è una funzione dello stato s e dipende dei parametri w
Q = Q(s, a | w)
z = f(u | w)

u – causa-input-stato => z – effetto-output-azione

Control / Classification / Prediction: determine {z} from {u},{w}
Inverse problem: determine cause {u} from {z},{w}
Inverse problem: Identification: determine {w} from {u}, {z} - *Learning*

f(u | w) è un **modello**, rappresentazione di una realtà: policy, Value function, Environment… Utilizzeremo il modello per il controllo / classificazione / predizione una volta calcolati i valori di {w}

Modello parametrico



I punti vengono fittati perfettamente da una sinusoide: y = **A** sin(**ω**x + **ϕ**). Devo determinare solo i 3 parametri della sinusoide (non lineare), i cui valori sono: **ω** = 1/200, **ϕ** = 0.1, **A** = 1. I parametri hanno un **significato semantico:** frequenza, fase e ampiezza (picco-picco). Dai punti {x, z_d} -> {ω, ϕ, A}.

Ma se non si sa che abbiamo una sinusoide…

I modelli (semi-)parametrici

- L'approssimazione è ottenuta mediante funzioni "generiche", dette di **base,** soluzione molto utilizzata nelle NN e in Machine learning (replicating kernels). E' anche associato all' approccio «black-box» in cibernetica. Non si hanno informazioni sulla struttura dell'oggetto che vogliamo rappresentare.
- E' anche l'idea che sta alla base delle Reti Neurali Artificiali

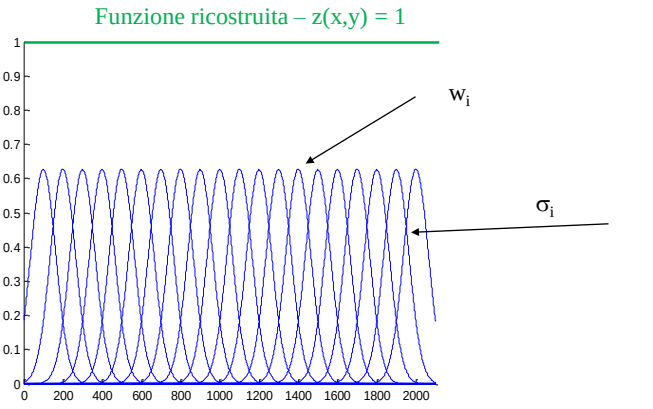
$$z(p(x,y)) = \sum_i w_i G(p(x,y), p_i(x,y), \sigma_i)$$

Combinazione lineare di funzioni di base

Da calcolare per ogni funzione di base:
- Parametro (peso)
- Posizione
- Ampiezza

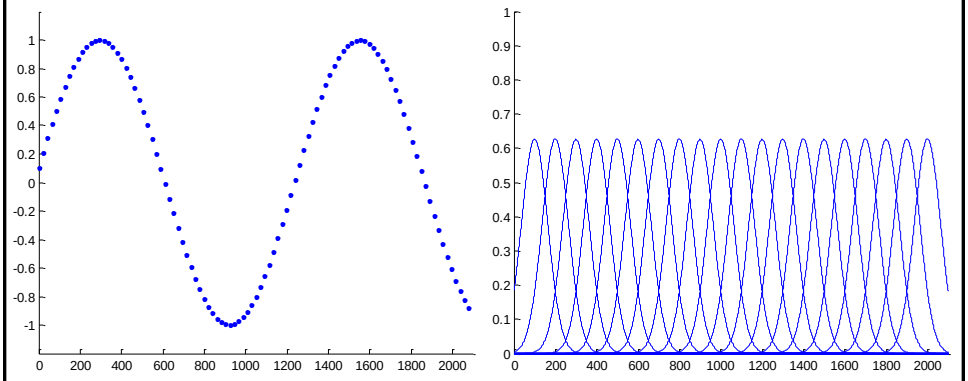
Funzione lineare di funzioni di base

Caso particolare: Funzioni di base, funzioni equispaziate



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

Approssimazione mediante un modello (semi-) parametrico (lineare) nello spazio 2D

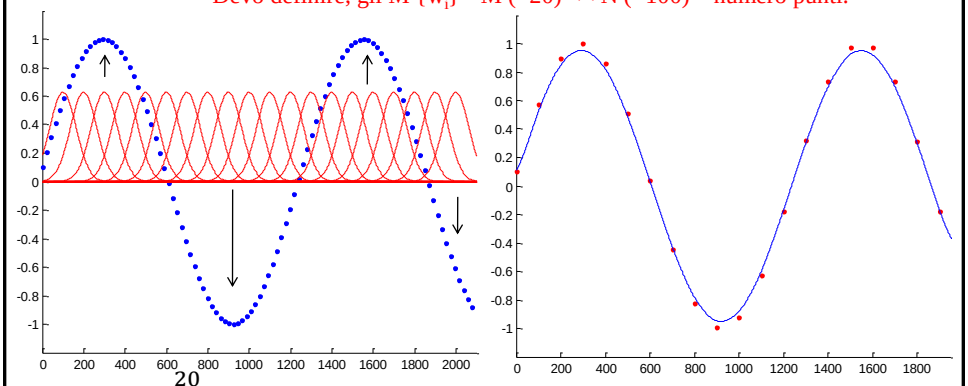


Sinusoide y = **A** sin(**ω**x + **ϕ**) con **ω** = 1/200, **ϕ** = 0.1, **A** = 1

Vogliamo fittare i punti con l'insieme di 20 Gaussiane riportate a destra che costituiscono una base. In questo caso hanno tutte σ = 90. Posso? Come le utilizzo?

Funzionamento di un modello parametrico (lineare)

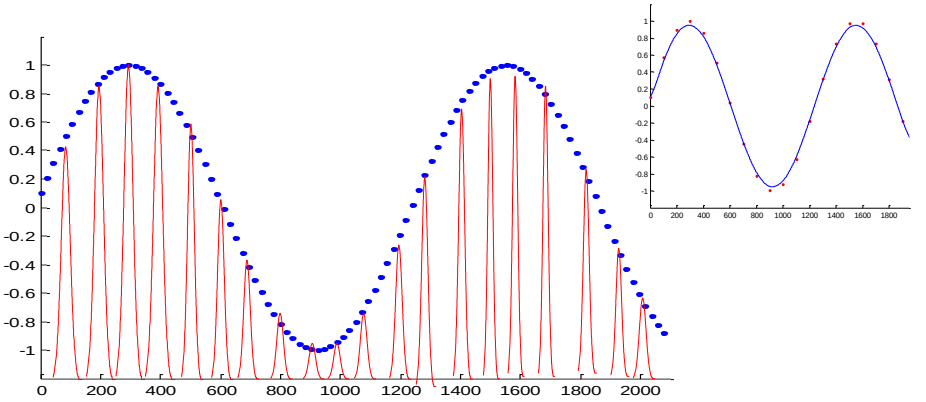
Devo definire, gli M $\{w_i\}$ - M (=20) << N (=100) – numero punti.



$$y(x) = \sum_{k=1}^{20} w_k G(x - x_k ; 90)$$

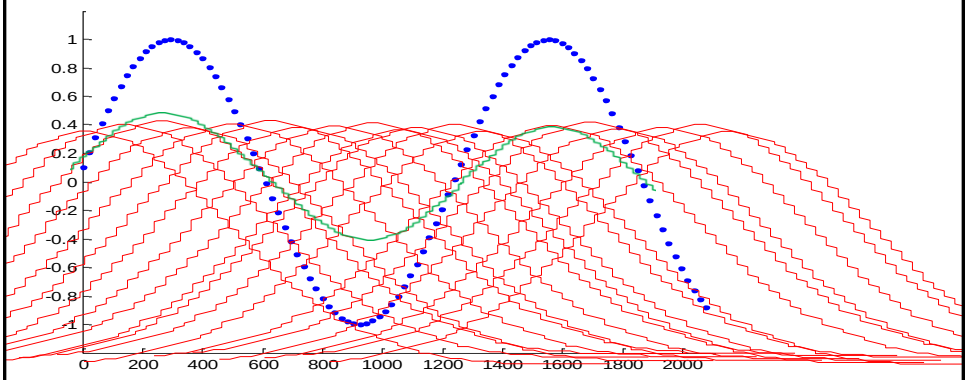
I σ sono tutti uguali ed uguali a 90°, le Gaussiane sono equispaziate.
C'è una relazione tra σ e spaziatura.
Le Gaussiane sono note tutte a priori, devono essere definiti i pesi w_k.

Ruolo di σ – σ piccolo



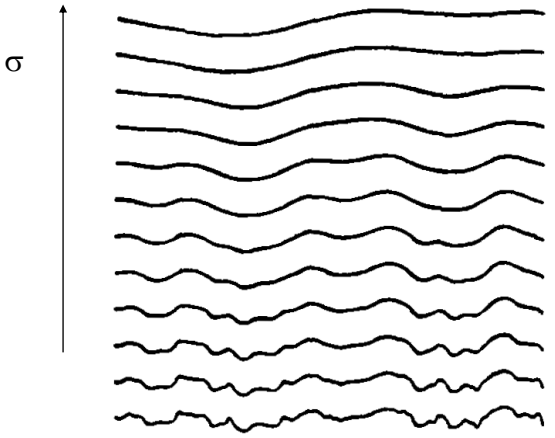
La ricostruzione continua rossa è molto lontana da una sinusoide!!
Le Guassiane devono essere sovrapposte (circa 30% - cf. Teorema di Shannon)
σ piccola vuol dire un numero maggiore di Gaussiane – impaccamento maggiore.

Ruolo di σ – σ grande



La ricostruzione in verde risulta smussata – sarebbe sufficiente un numero minore di Gaussiane.
Le Guassiane devono essere sovrapposte (circa 30% - cf. Teorema di Shannon)
σ grande vuol dire un numero minore di Gaussiane – impaccamento minore.

Effect of σ on a convolutional layer



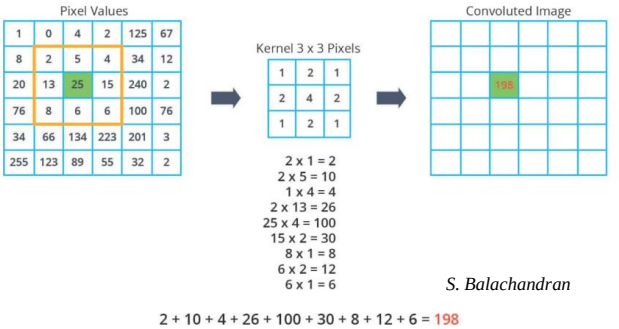
- Diminuisce il numero di Gaussiane /
- Aumenta la spaziatura
- Diminuisce σ

Original signal

Gaussian convolutional layer applied to a 1-D signal

- Non solo interpolazione, anche filtraggio.

Convolution operator



Discrete convolution with a Gaussian kernel:

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

Model as a filter (convolution)

- Convolution: $\hat{f}(x) = \int_{\mathbb{R}} f(c)\, G(x-c|\sigma)\, \mathrm{d}c = f(x) * G(x;\sigma)$

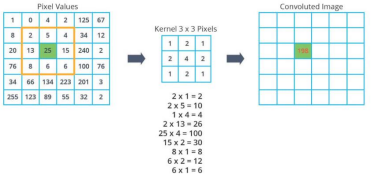
we can construct output up to a certain scale (level of detail), provided an adequate small value of σ.

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

The construction of the output, $\hat{f}(x)$, if G(.) is normalized, is obtained through digital filtering.

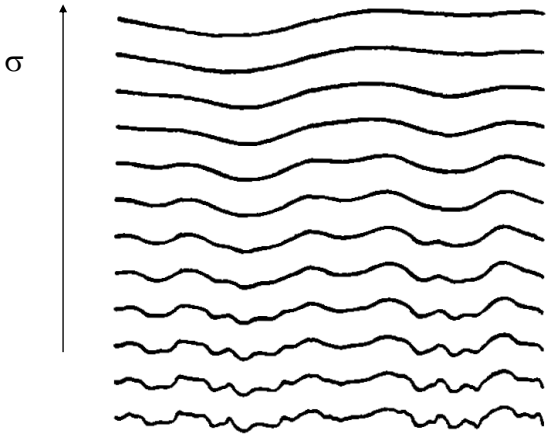
Extrapolation beyond the sample points. Continuos reconstruction.

It reconstructs the details of f(.) up to a given scale.



Convolutional networks.

Effect of a convolutional layer



- Diminuisce il numero di Gaussiane /
- Aumenta la spaziatura
- Diminuisce σ

Original signal

Gaussian convolutional layer applied to a 1-D signal

Filters and bases

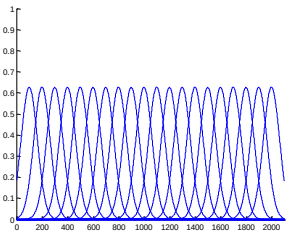
$$\hat{f}(x) = f_i * G(x - x_{k_i}; \sigma) = \sum_{i=1}^{N} w_i G(x - x_{k_i}; \sigma)$$

Con funzioni di base normalizzate:

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

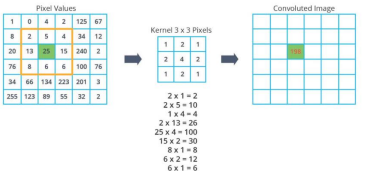
Normalized Gaussians, filter = weighed sum of **shifted (normalized) basis functions**.
Basis representation. Approximation space.
No amplification takes place: If $\{w_i\} = k \; \forall$ => f(x) = k

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

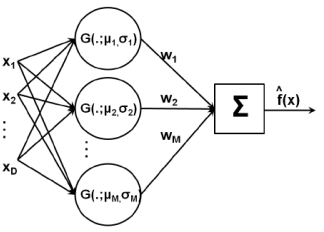


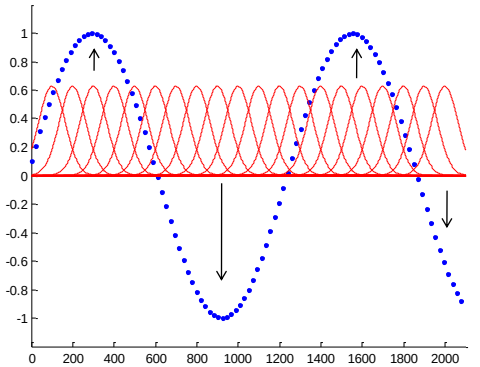
Different views

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



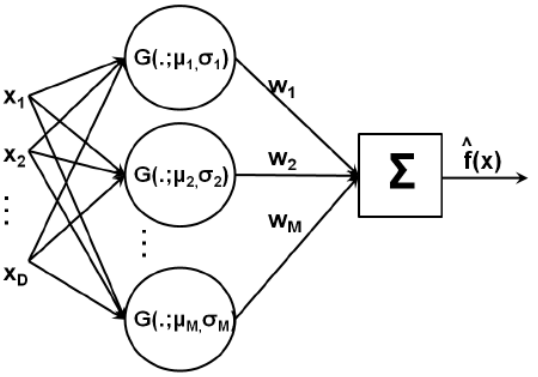
- **NN – RBF networks – perceptron**
- Digital filters
- Functional bases





Convolutional layer

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



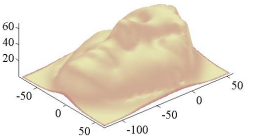
When G(.) are not equally spaced -> RBF Networks. Perceptron

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

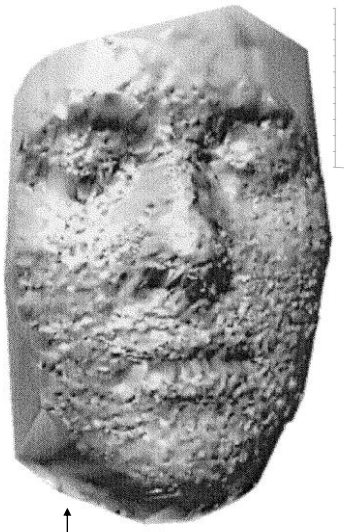


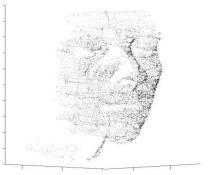
Combinazione lineare di funzioni di base

Da calcolare

Funzione di base (fissate)

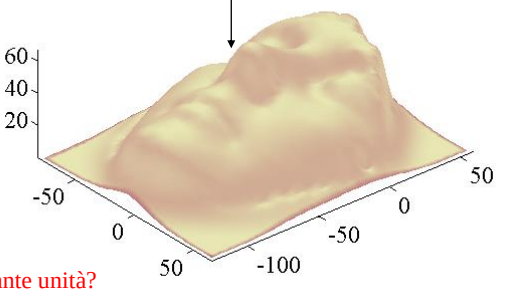
Applicazione: scanner 3D





z = f(x,y | w) – alto-rilievo

Buona approssimazione



Quante unità?

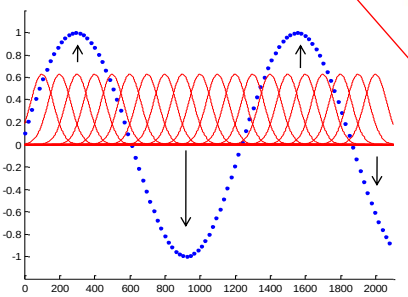
Problema dell'overfitting dovuto a sovra-parametrizzazione

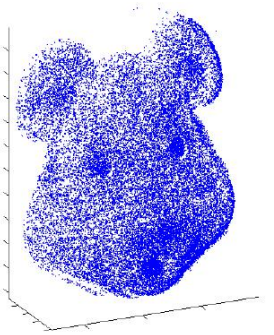
Advantages and issues

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

Gridding

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

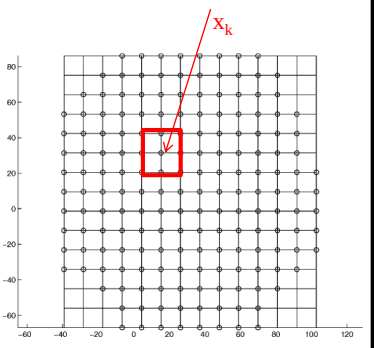
Gaussians equally spaced and distributed over a grid. How can we determine their associated weight, w_k, from points clouds?

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

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

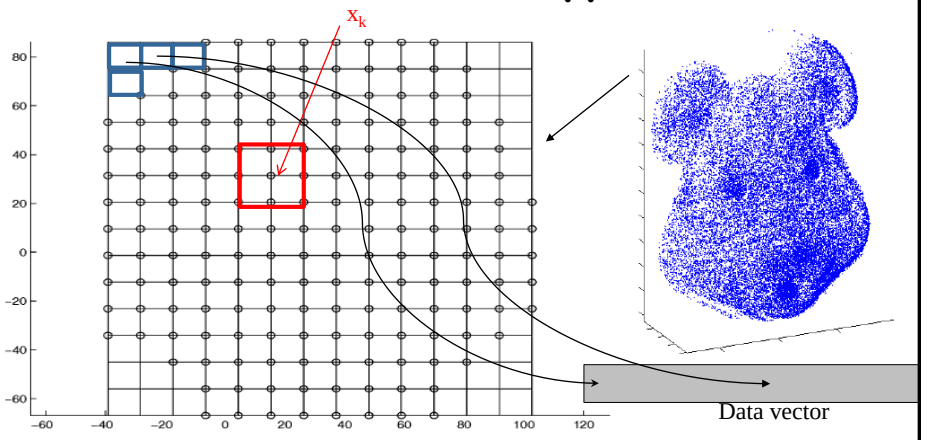
$K_\sigma(.)$ Gaussiana

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



Parzen-windows estimate

Efficient data support

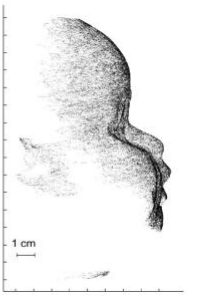


Data comes into a linear vector and are sorted into **quads** each centered in each Gaussian center ➔ Each quad points to a position in the data vector ➔ **in-place ordering inside the vector**, by data position.
The receptive field of x_k is constituted of 4 quads and the data considered for estimating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

Example: 3D scanner

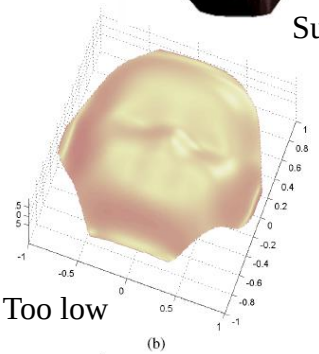
Which scale?

Too high



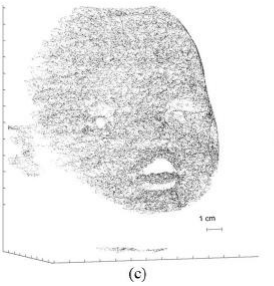
(b)

Supervised learning



Too low

(b)



(c)

(d)

Riassunto

- Supervised learning
- Predictive regression
- Regressione multi-scala

Filters and bases

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

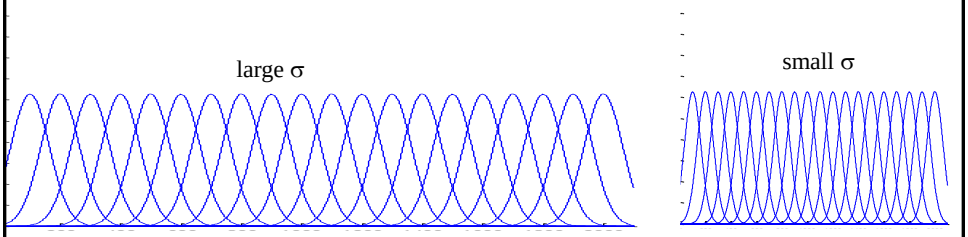
con funzioni di base normalizzate:

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, $\Delta x > 2\nu_{Max}$

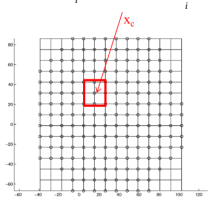
σ determines the amount of overlap. It determines also the frequency content of the Gaussian G(.).

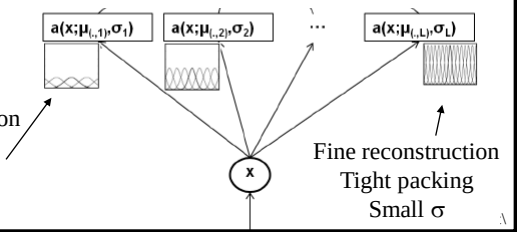
Once σ (or Δx is defined) the grid is also defined.

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

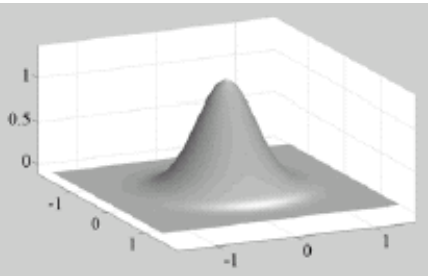
Starting from low resolution

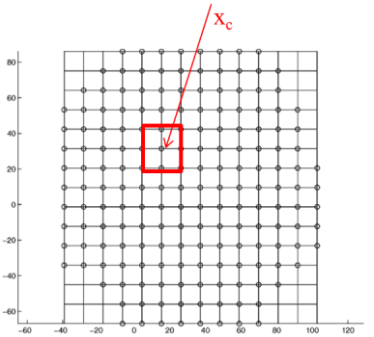
How many points to consider? The Gaussian has infinite support.

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.

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



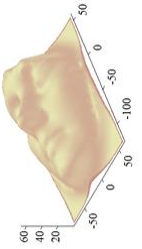


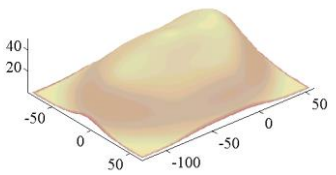
We can obtain a «poor» reconstruction

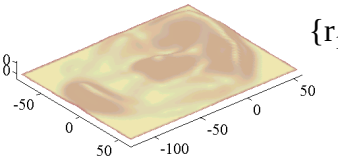
Little detail. Large scale. But it is a start. It can be seen as a modified support for successive approximations.

Regular grid with few Gaussians largely spaced with large σ

What can be done?





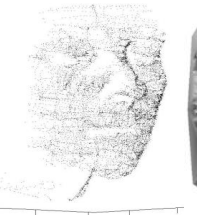


$\{r_1(\mathbf{x})\}$

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

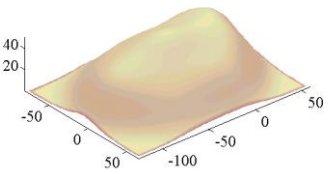
E.g.: $r_1 = \left(y_m - \hat{f}(x_m)\right)^2$ $r_1 = \left|y_m - \hat{f}(x_m)\right|$

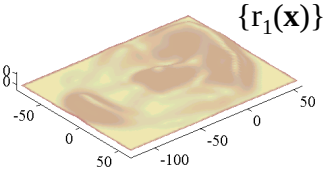
Is the residual adequate?

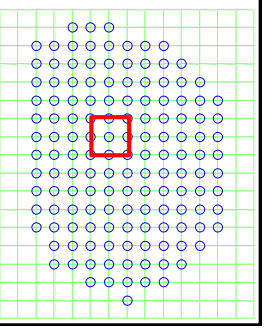


Approximation at layer #1

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



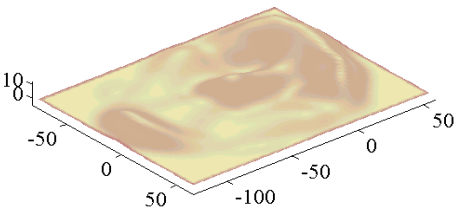




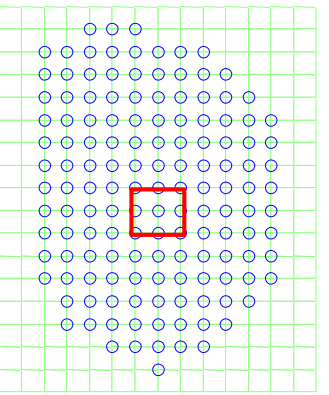
For each Gaussian t**he integral of the residual inside the "receptive field"** of the Gaussian, is assumed as local approximation error associated to it. , is computed inside its "receptive field":

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

How can we evaluate the local adequacy of the reconstruction?



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



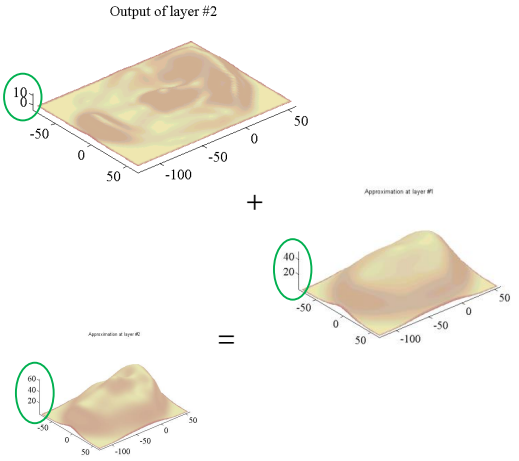
We compare the local residual with a threshold derived from:
- Degree of approximation
- Noise: RMS.

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

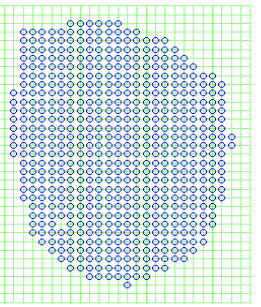
Layer 2

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

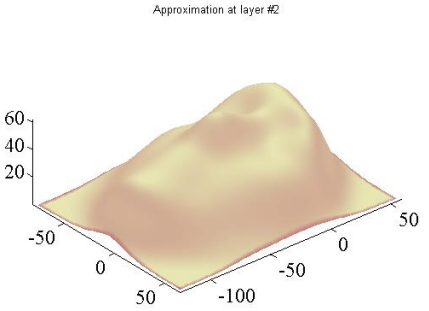
Output is a layer that approximates $r_{1,m}$: $f_2(x_m) \rightarrow r_{1,m}$



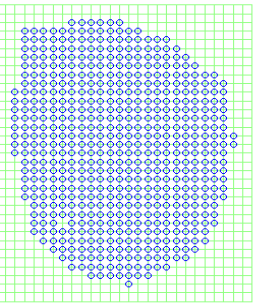
Layer #2



Evaluation of Layer 2



Layer #2



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

First approximation + first residual

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

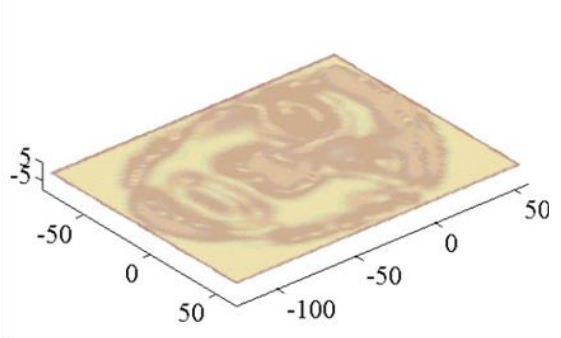
More packed Gaussians. More details. But…

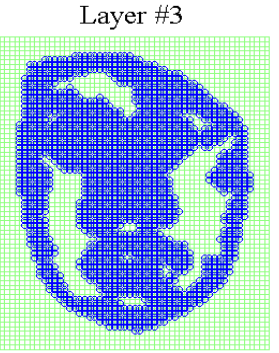
There should be enough points to have a reliable local estimate of Gaussian height.

Layer 3

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

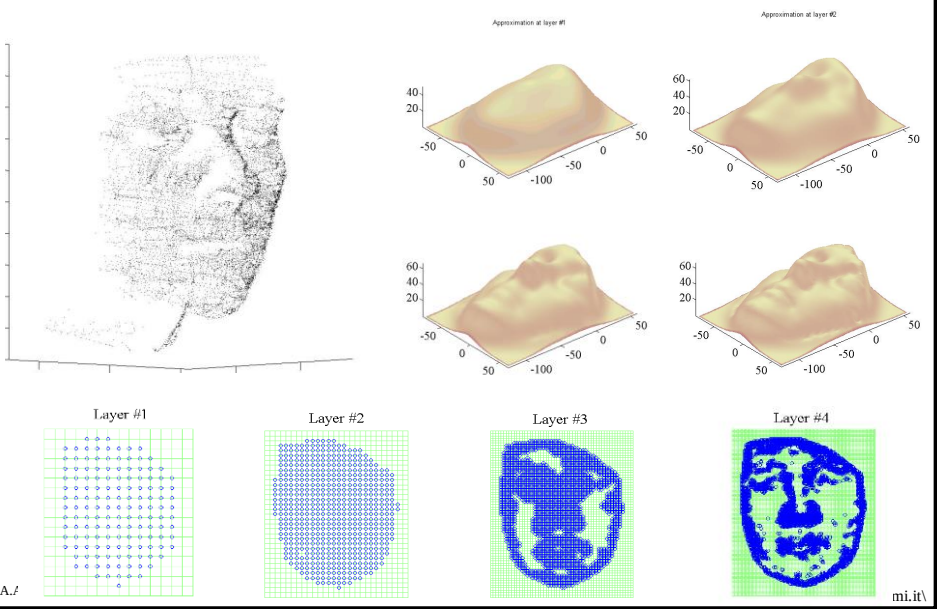
Output is a layer that approximates $r_{2,m}$: $f^{III}(x_m) \rightarrow r_{2,m}$



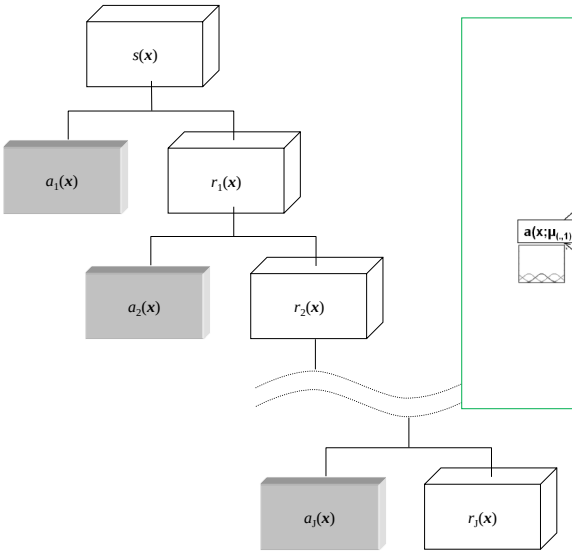


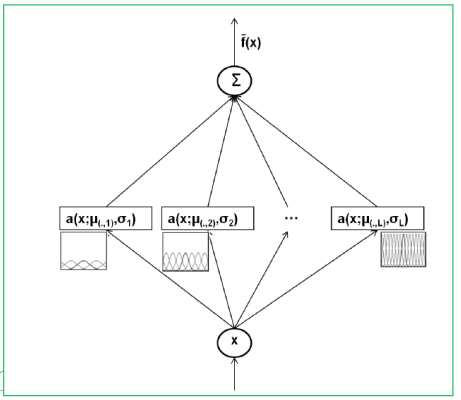
Sparse approximation in the third layer with $\sigma = \sigma_3$.

Applicazione alla regressione



Hierarchy construction

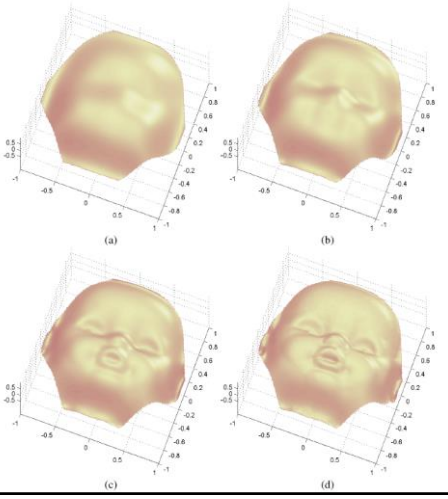


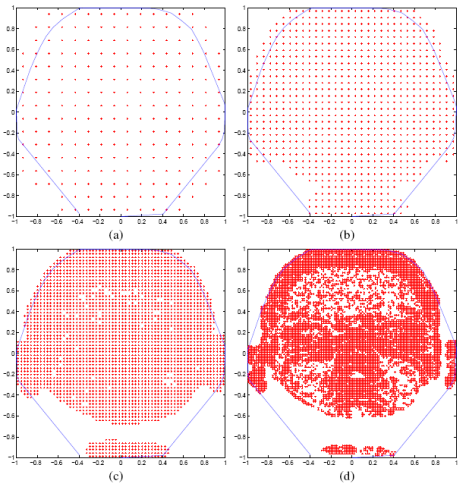


It generates a set of hierarchical layers

How to operate on large sets of data?

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





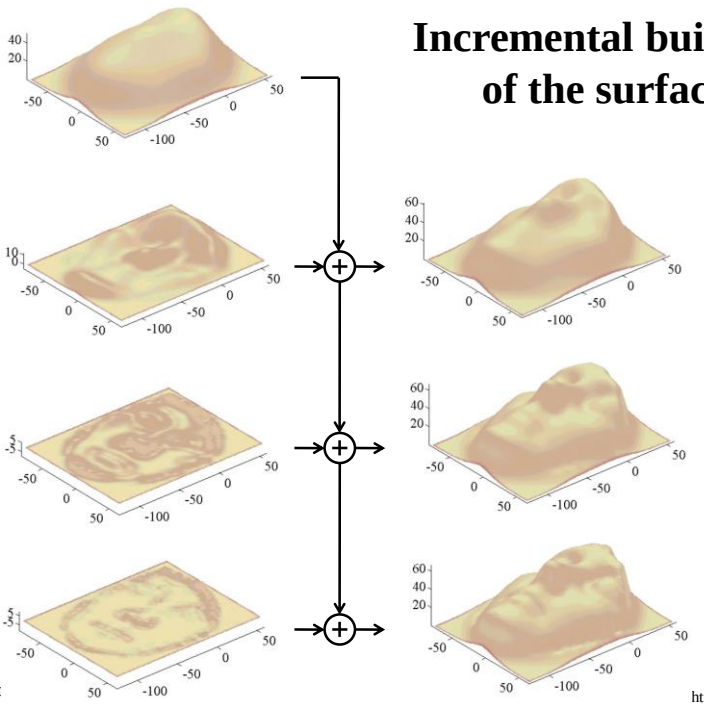
Characteristics of HRBF networks

- 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

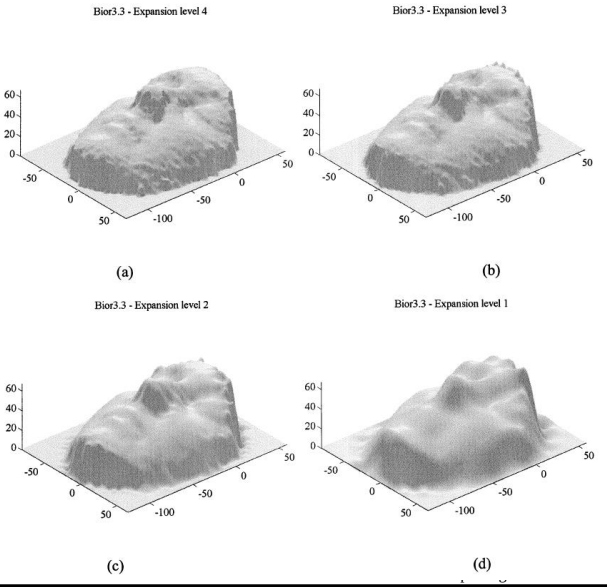
$$\cos\ \alpha_j = \sup_{f(\cdot)\in V_j, h(\cdot)\in V_{j+1}} \frac{\langle f(\cdot), h(\cdot)\rangle}{\|f(\cdot)\|_2 \|h(\cdot)\|_2} = \cos\alpha_{j-1}.$$

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.

Riassunto

- Supervised learning
- Predictive regression
- Regressione multi-scala