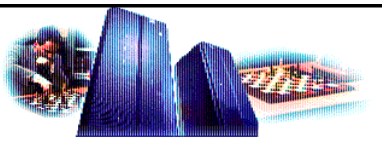
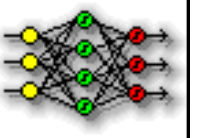




# Sistemi Intelligenti
# Supervised learning

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



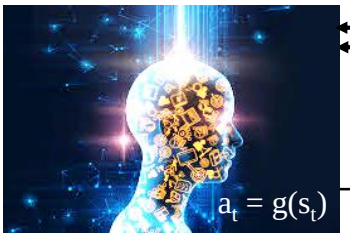
## Riassunto

- Supervised learning; predictive regression
- Regressione multi-scala
- Versione on-line
- Valutazione del modello

# Apprendimento con rinforzo

**Agent**

What the world is like now (internal representation)?




What action should I choose now? (policy)

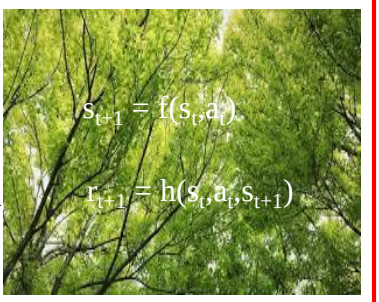
Which is the value of my action (value function)?

$s_t$ $s_{t+1}$

$r_t$ $r_{t+1}$

$a_t$

**Environment**



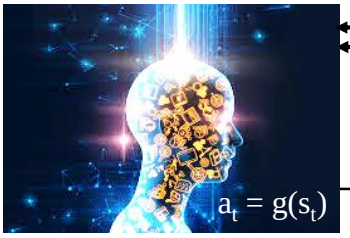

L'ambiente fornisce un'informazione **puntuale**, di tipo **qualitativo**, ad esempio success or fail.



# Apprendimento Supervisionato

**Agent**

What the world is like now (internal representation)?




What action should I choose now? (policy)

Which is the suggested action?

$s_t$

$a^D_t$

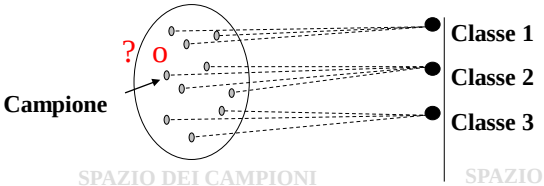
$a_t = \pi(s_t)$

**Environment**

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

# Classificazione e regressione

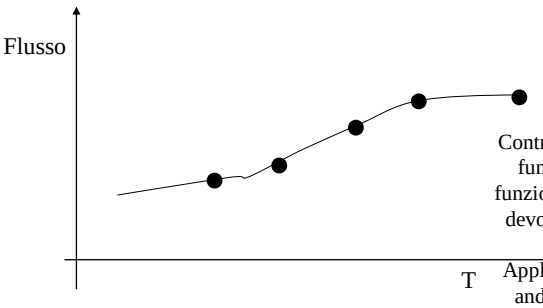
*Mappatura dello spazio dei campioni nello spazio delle classi.*




SPAZIO DEI CAMPIONI / DELLE FEATURES (CARATTERISTICHE)

SPAZIO DELLE CLASSI (identificate da un'etichetta)

Classifico ("aiuto alla clusterizzazione")




Quanto vale ?

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

Applicazioni alle serie temporali: ad esempio andamento borsa, previsioni del tempo,….



# Classificazione

- Boosting. Si utilizza un insieme di classificatory binary, dove ciascun classificatore lavora su una singola feature. La classificazione avviene prendendo la maggioranza di voto dei classificatory.
- Reti neurali. Approccio black-box generale.
- Support Vector Machines. Calcolo 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

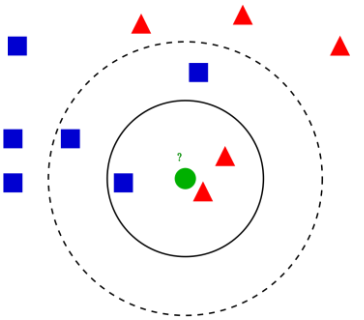
# Algoritmo K-NN

**K-Nearest Neighbour**

Definisco una misura di vicinanza (campo recettivo).

Per ogni input, considero i K dati più vicini per i quali è stata prescritta un'azione.

Scelgo l'azione combinando questi K dati (max, soft-max, combinazione lineare o non-lineare).




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
Se consideriamo i 2 dati più vicini -> rosso
Se consideriamo I 5 dati più vicini -> blu.



# Modello per prendere la decisione

$a = \pi(s \mid w)$

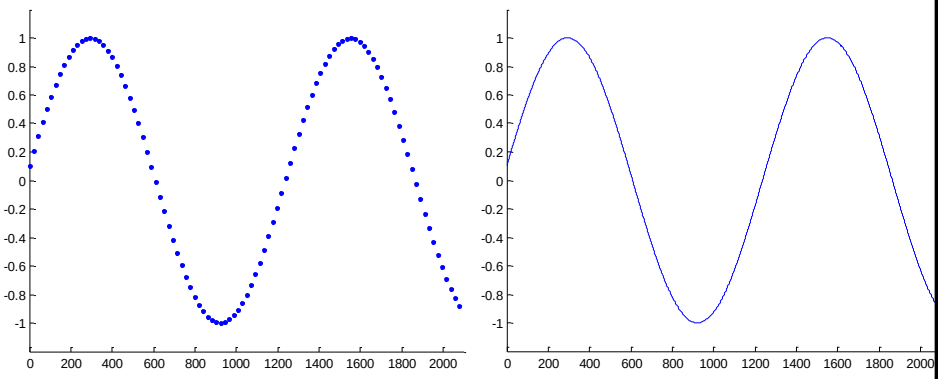
$z = f(u \mid w)$


u – causa-input => z – effetto-output

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

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

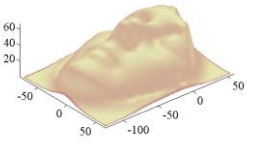
Da calcolare per ogni funzione di base:
- Peso
- Ampiezza
- Posizione

# Modelli supportati da una base

- **Le fuzioni di base sono equispaziate (posizionate su una griglia) e tutte con gli stessi parametri (in questo caso σ). S**
- Struttura di supporto semplificata.
- (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



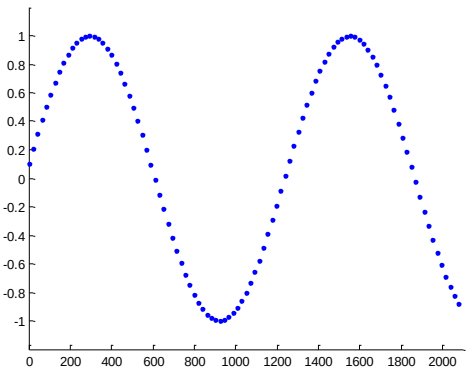

Combinazione lineare di funzioni di base

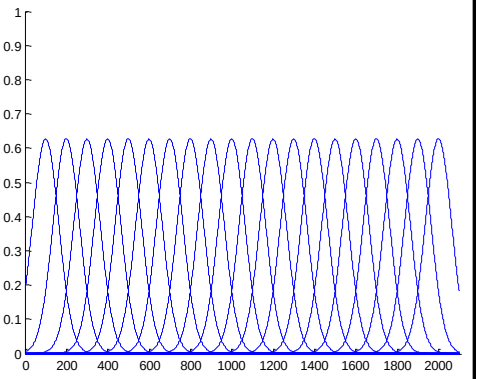
Da calcolare

Funzione di base (fissate)



# Approssimazione mediante un modello (semi-)parametrico (lineare)



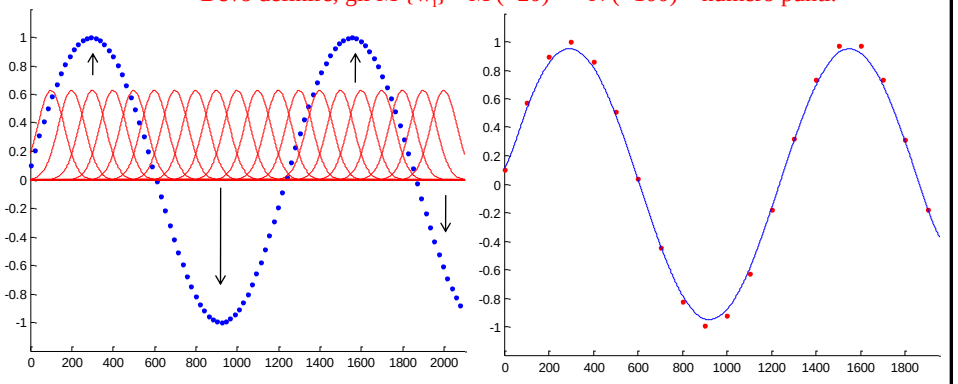




Sinusoide y = **A** sin(**ω**x + **ϕ**) con **ω** = 1/200, **ϕ** = 0.1, **A** = 1

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.



$$y(x) = \sum_{i=1}^{20} w_i G(x - x_{o_i}; 90^o)$$

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.



# 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.

# Filters and bases

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

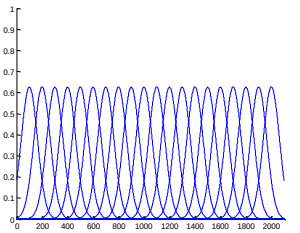
Con funzioni di base normalizzate:

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

$\frac{\Delta x_k}{\sqrt{\pi}\sigma}$ 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.




A sequence of spaces can be defined according to σ:

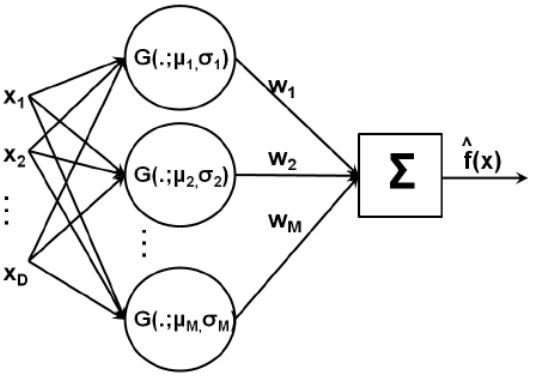
$\sigma_0$ -> $V_0$; $\sigma_1$ -> $V_1$; $\sigma_2$ -> $V_2$….

The number of representable functions increases.



# RBF Network

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

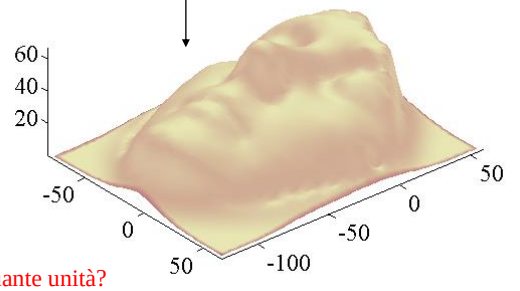



Perceptron

# Esempio: scanner 3D

z = f(x,y | w) – alto-rilievo

Buona approssimazione




Quante unità?

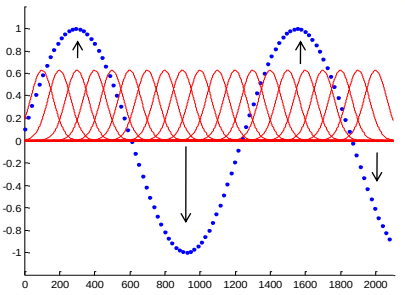
Problema dell'overfitting dovuto a sovra-parametrizzazione

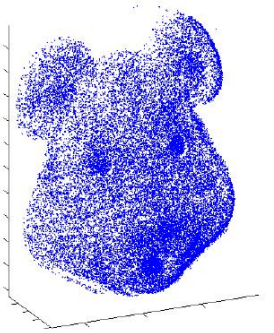


# Advantages and issues

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

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



**Height of the surface on a grid crossing is not known in general.**

Points clouds

# Gridding

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

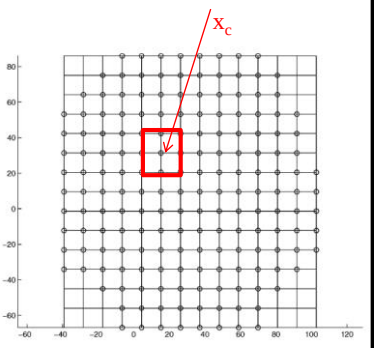
Gaussians equally spaced and distributed over a grid. How can we determine $w_k$ from points clouds?

Local estimators. Nadaraya Watson estimator. *Lazy learning*.

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

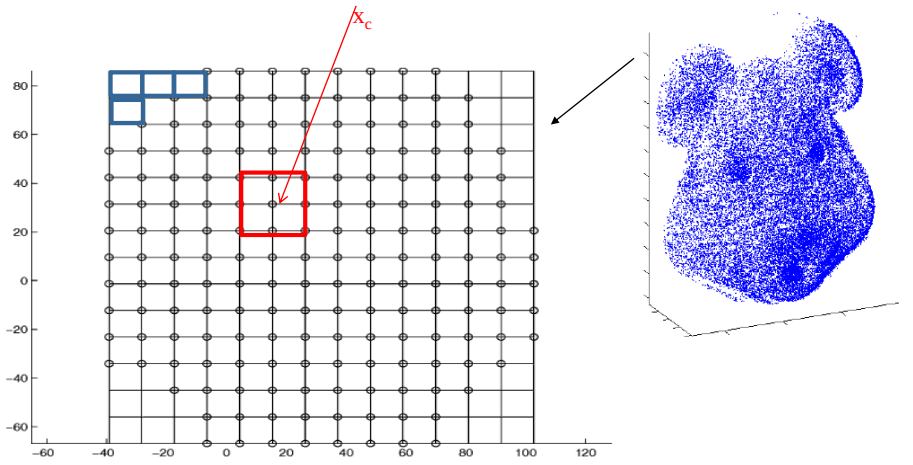
$K_\sigma(.)$ Gaussiana
Troncamento dell'ampiezza del "campo recettivo"
Dati $\{f(x_i), x_i\}$ all'interno di un numero di celle vicine.

Parzen-window approximation






# Efficient data support




Data are distributed to quads with Gaussian centers as vertexes.
Data are collected inside a vector ➔ in-place ordering inside the vector by position.
The receptive field of xc is constituted of 4 quads and the data considere are those inside those quads.

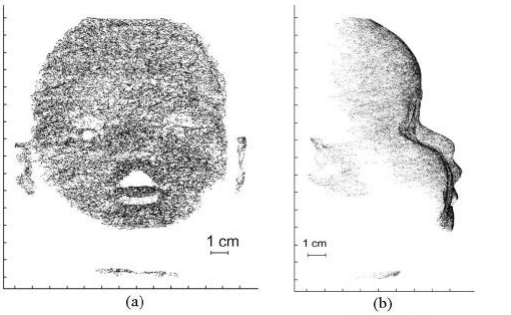
# Example: 3D scanner

- Properties:

- Redundancy.

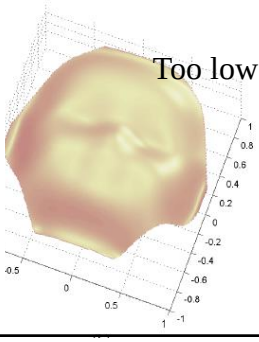
- Riesz basis (unique representation, given the height in the grid crossings).

**Which scale?**




Too high

Too low





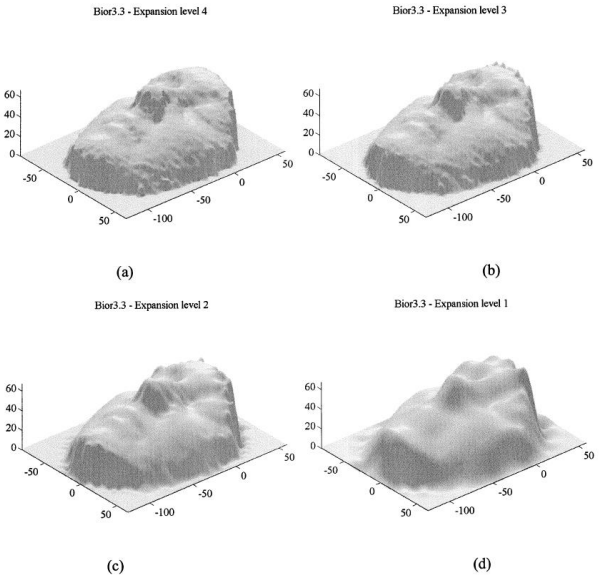
http:\borghese.di.unimi.it\

# Riassunto

- Supervised learning: predictive regression
- Regressione multi-scala
- Versione on-line
- Valutazione del modello

# 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?






# 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, small distance, $1/\Delta x > 2\nu_{Max}$
- Δx = width of the domain of definition

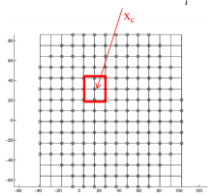
σ determines the amount of overlap. It determines also the frequency content of the Gaussian G(.).

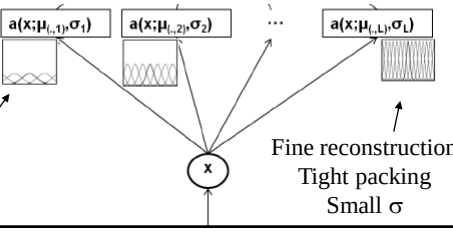
Once σ (or Δx is defined) the grid and mesh size are also defined.

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



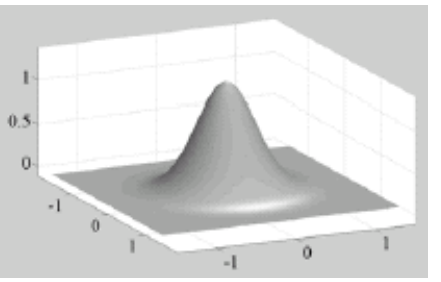
# Starting from low resolution

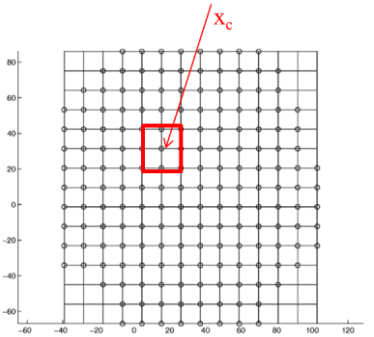
How many points to consider? The Gaussian has infinite support.

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.

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



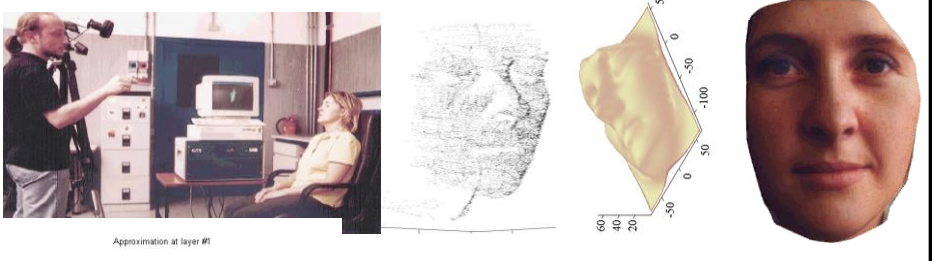
# We can obtain a «poor» reconstruction

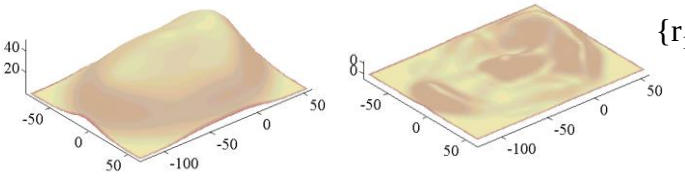
Little detail. Large scale. But it is a start. It can be seen as a modified support for successive approximations.


Regular grid with few Gaussians largely spaced with large σ



# What can be done?




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

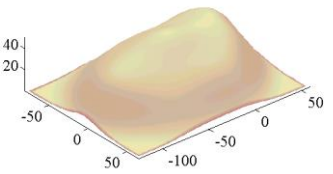
We evaluate the residual for **each data point**: $r_1 = dist\left(y_m, \hat{f}(x_m)\right)$

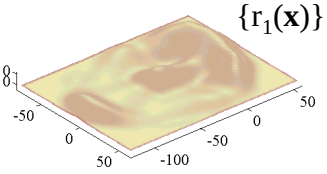
E.g.: $r_1 = \left(y_m - \hat{f}(x_m)\right)^2$ $r_1 = \left|y_m - \hat{f}(x_m)\right|$

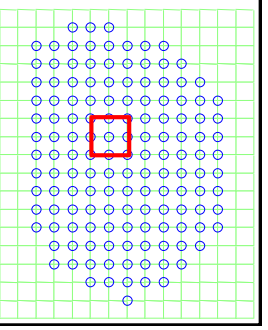
# Is the residual adequate?

Approximation at layer #1

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



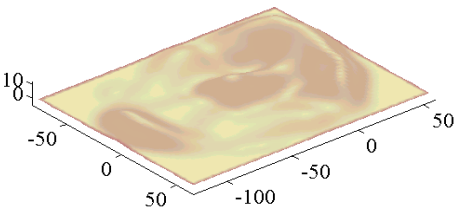



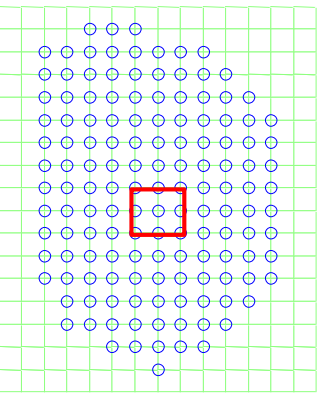
For each Gaussian t**he integral of the residual inside the "receptive field"** of the Gaussian, is assumed as local approximation error associated to it. , is computed inside its "receptive field":

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

# 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 it with a threshold derived from:
- Degree of approximation
- Noise: RMS.

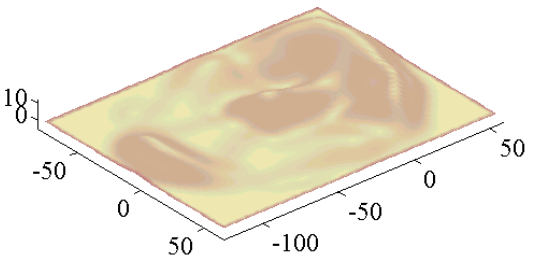
We aim to have an error that is **uniformly** under a given threshold.

# Layer 2

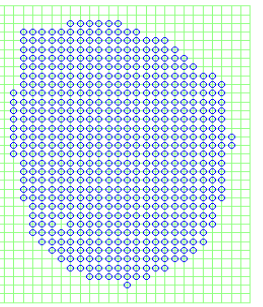
Input are the residuals of previous layer, $r_{1,m}$= $\left|y_m - \hat{f}_1(x_m)\right|$

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

Output of layer #2



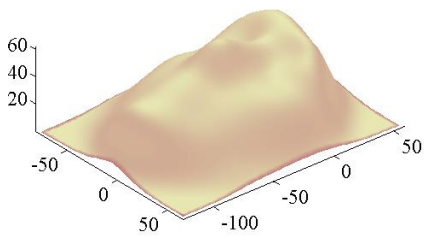

Layer #2



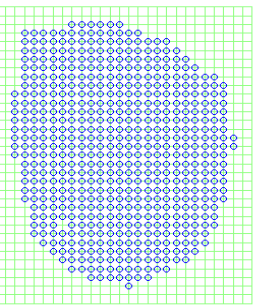


# Evaluation of Layer 2

Approximation at 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

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

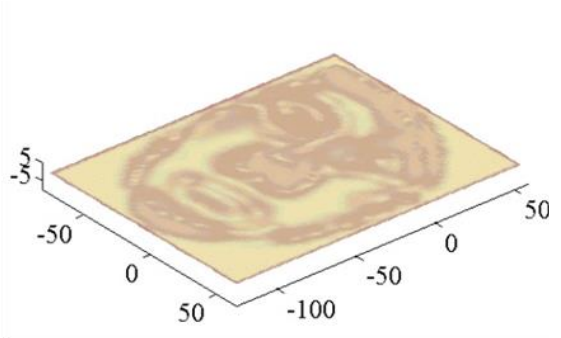
More packed Gaussians. More details. But…

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

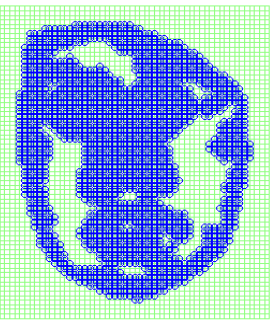
# Layer 3

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

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



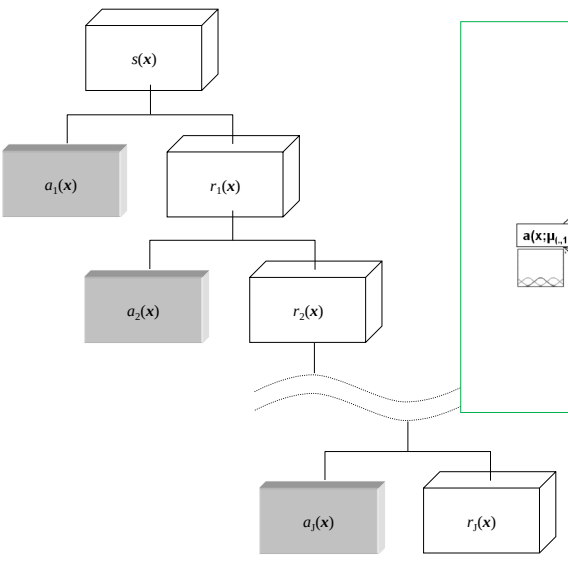

Layer #3

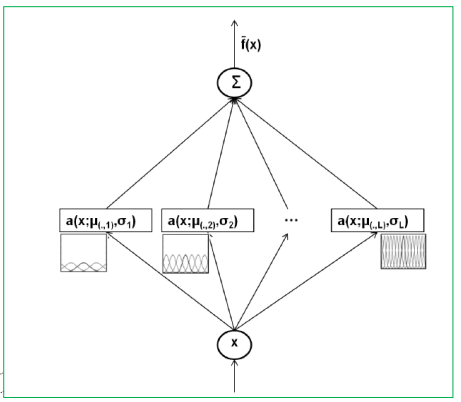


Sparse approximation in the third layer with $\sigma = \sigma_3$.



# Hierarchy construction

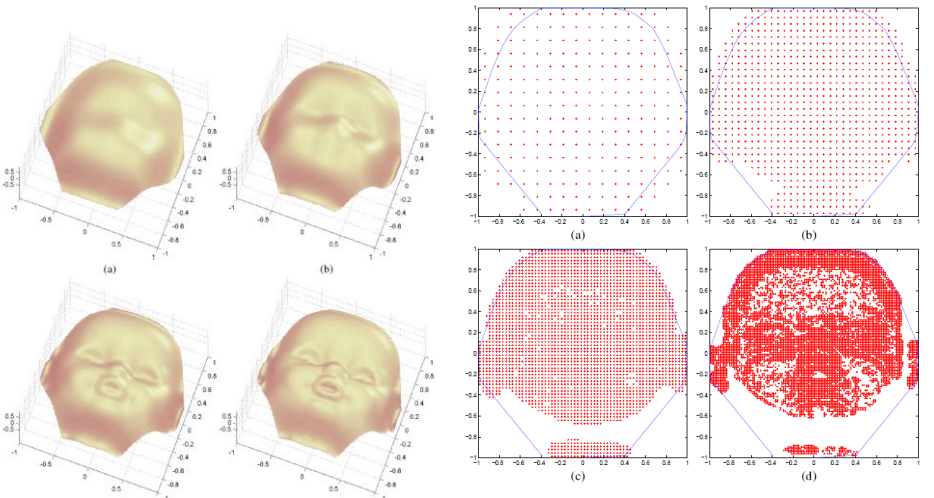



It generates a set of hierarchical layers

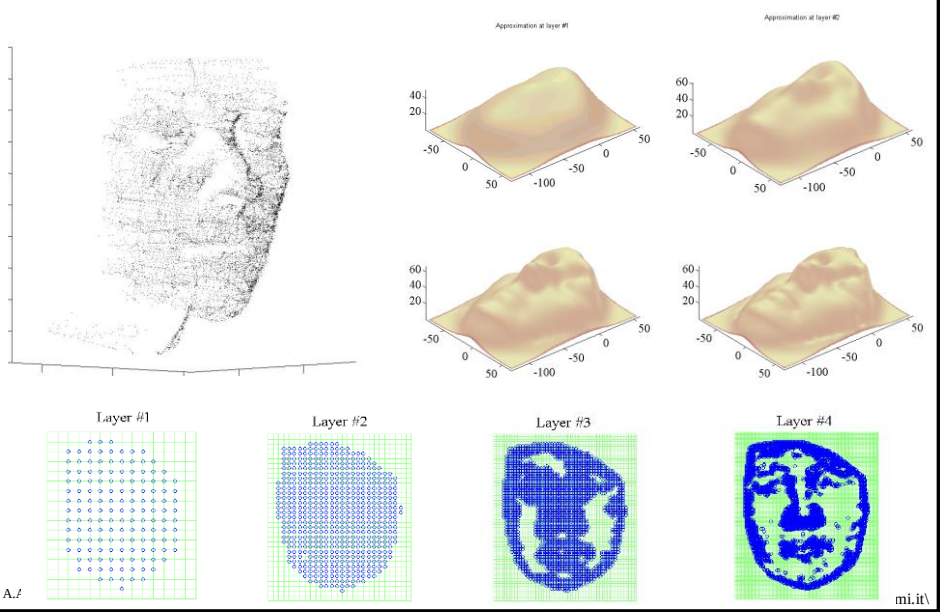
# How to operate on large sets of data?

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




http:\borghese.di.unimi.it\

# Applicazione della regressione




A.Borghese, mi.it\

# Characteristics of HRBF networks

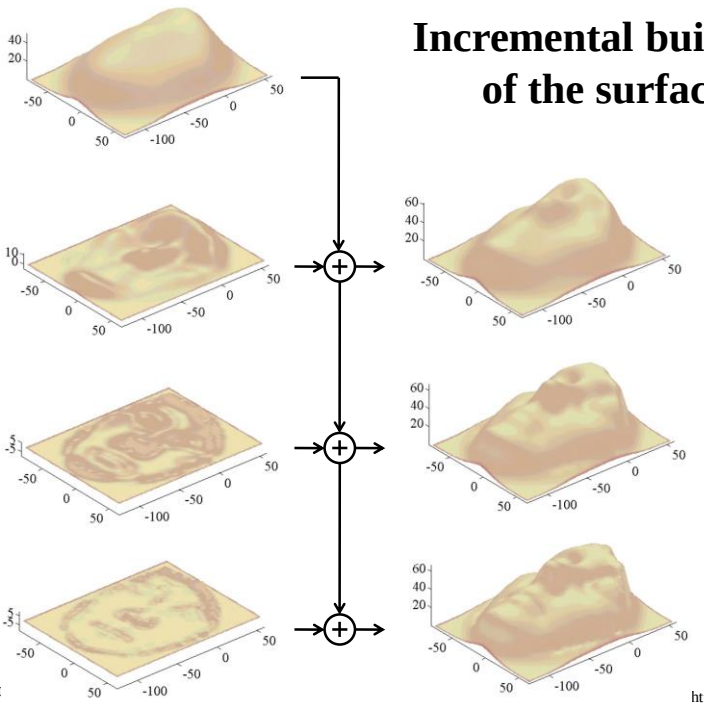
- Local operations.
- Hierarchy of approximations.
- Local adaptation of the scale (Not fully occupied layers)
- Adaptive allocation of the resources
- Uniform convergence to a residual error
- No hyper-parameters have to be set

- 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



# Riassunto

- Supervised learning: predictive regression.
- Regressione multi-scala
- Versione on-line
- Valutazione del modello



# On-line version

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

2 min video

hrbf_online.wmv

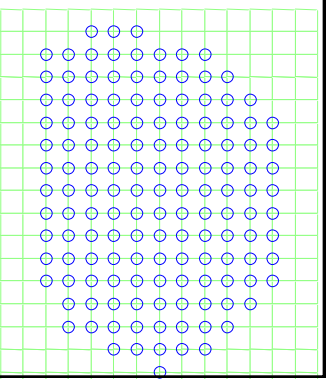
# Observation

- Each new point, $y_k=f(x_k)$, modifies at least the coarsest approximation, $f_1(.)$, around the point $\{x_k, y_k\}$ and possibly the more detailed approximations, if present.
- The height of the affected Gaussians should be recomputed.

Recomputation can be simplified. Numerator and denominator are stored separately.

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

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

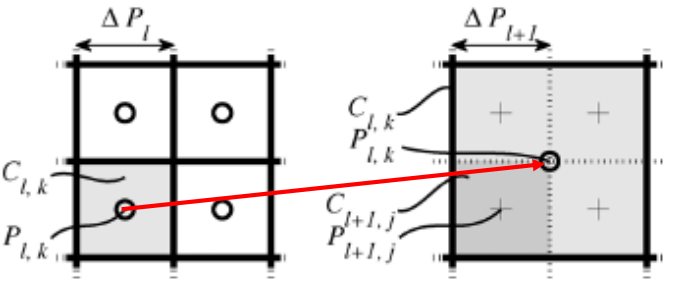




# Local operations

- 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**

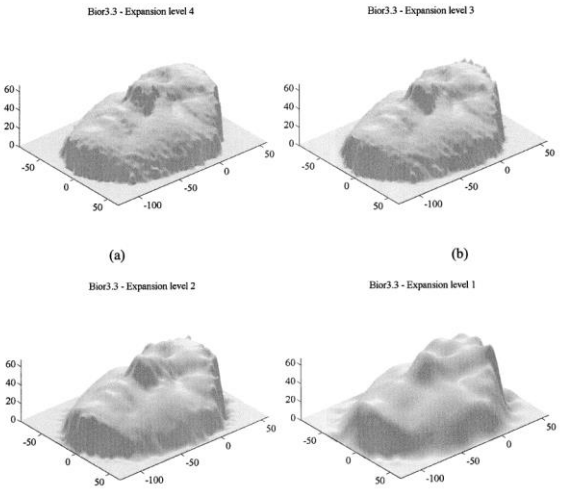
1 quad




4 quad

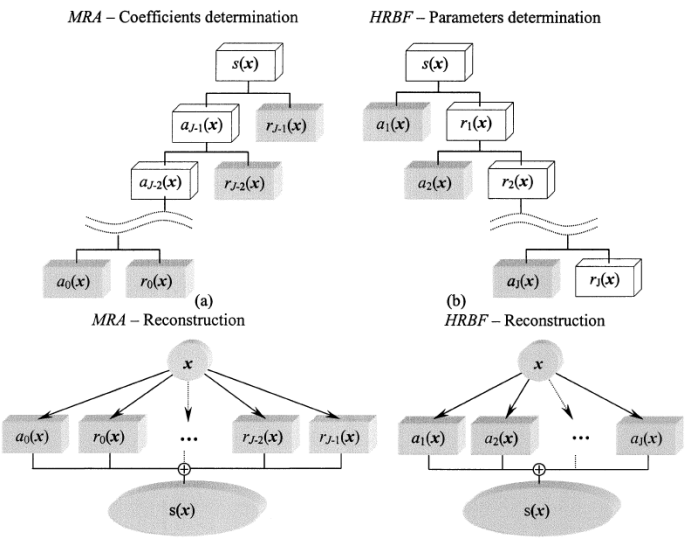
# Comparison with Wavelets

- Fast incorporation of the content (high angles between approximating spaces -> 90 degrees)
- No control on the residual.






# Comparison with Multi-Resolution Analysis (wavelet)



# 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.

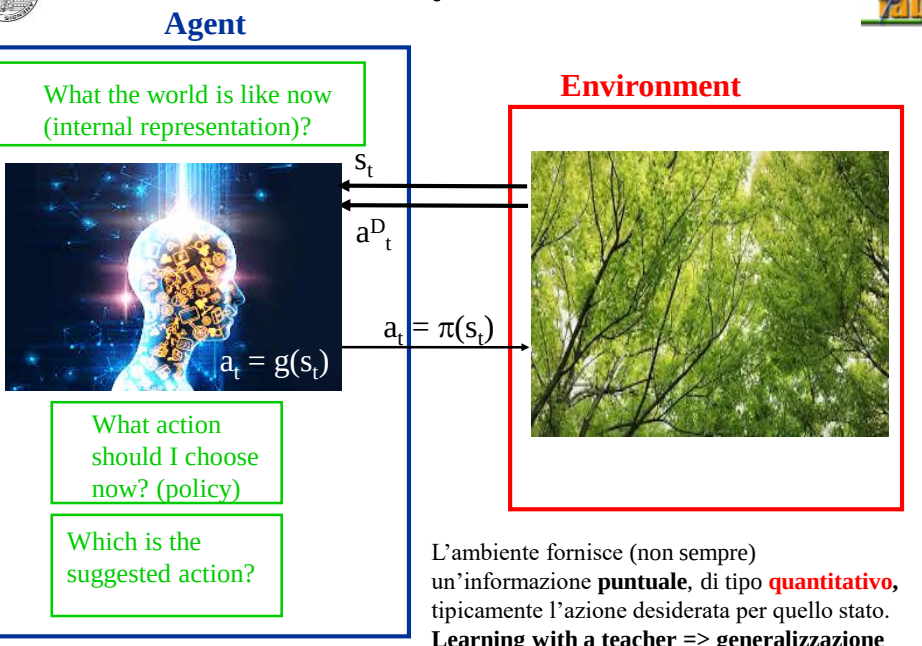
RBF and Wavelet have excellent for CUDA implementation as all bases with limited support.



# Riassunto

- Supervised learning: predictive regression.
- Regressione multi-scala
- Versione on-line
- Valutazione del modello

# Quando $\{a^D_t\}$ sono sufficienti?



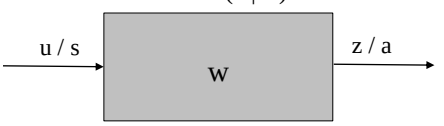

L'ambiente fornisce (non sempre) un'informazione **puntuale**, di tipo **quantitativo,** tipicamente l'azione desiderata per quello stato.
**Learning with a teacher => generalizzazione**



# Valutazione della bontà del modello

$a = \pi(s \mid w)$

$z = f(u \mid w)$




u – causa-input => z – effetto-output

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 valtori di {w}

Quando il modello è buono?

# How to classify the error introduced by a model?

Is the model good enough?

Does it have enough parameters? (under-parameterization)
Does it cover the input domain (in all dimensions – dimensionality discovery)?

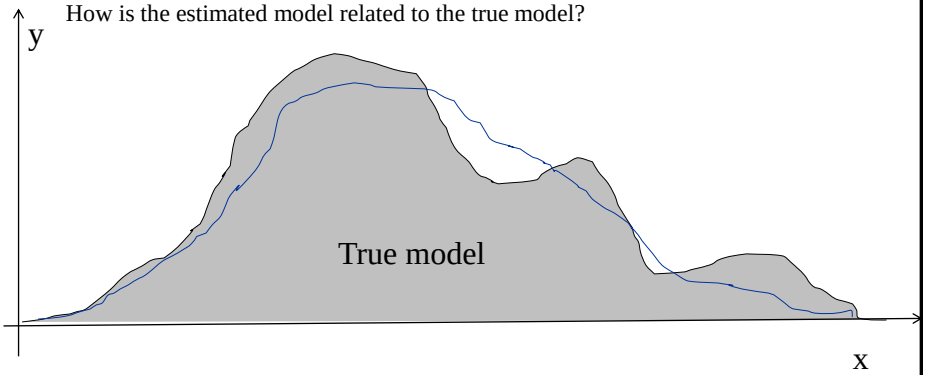
This is not enough to obtain a good model!!
The model should be properly tuned to the data

Source of errors:
- Bias
- Variability



# How to classify the error introduced by a model?

How is the estimated model related to the true model?

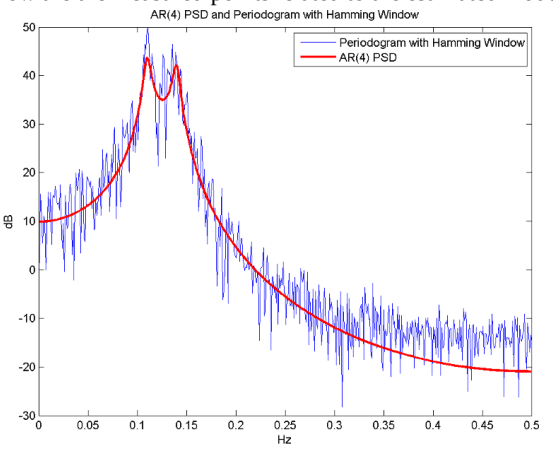



Bias and variability trade-off

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

# Variability

How are the measured points related to the estimated model?




Given $P_{mes}(x_{mes},y_{mes})$ and $y = f(x)$, the error is measured as: $dist(y_{mes},f(x_{mes}))$, for instance Euclidean distance. It is associated to measurement error.

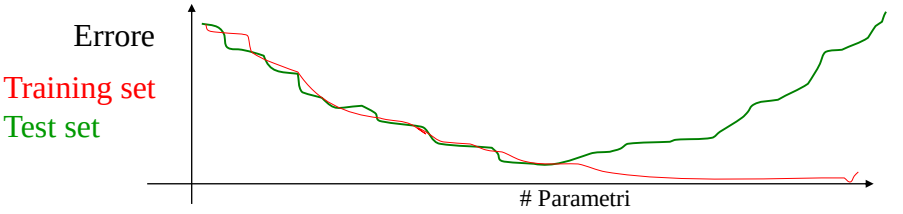
If variability goes to zero, bias increases and overfitting arises.
**In a good model, variability tends to the statistics of the data noise.**

# Scelta empirica – cross-validation

Cross-Validation - Errore sull'insieme di training = Errore sull'insieme di test.

*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).*

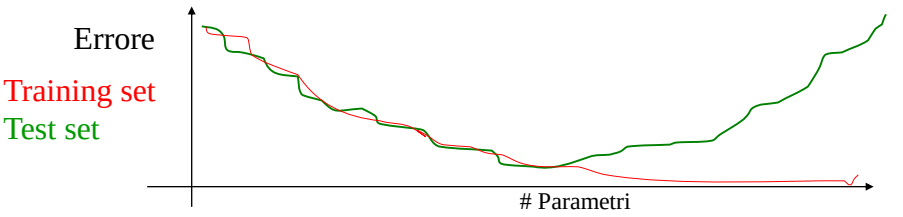
*Il numero di parametri viene aumentato fino a quando **entrambi** gli errori diminuiscono.*



# Scelta teorica

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

*Penalizzo i modelli con tanti parametri. Regularization with Reproducible Hilbert Kernels as regularizers.*

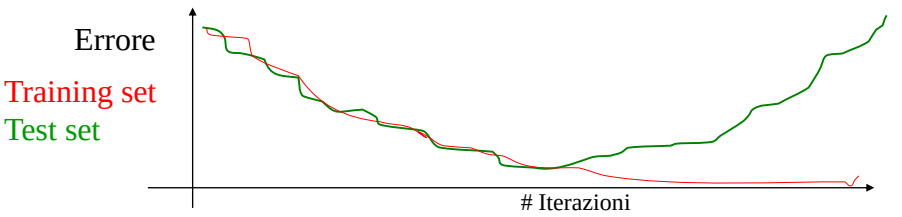




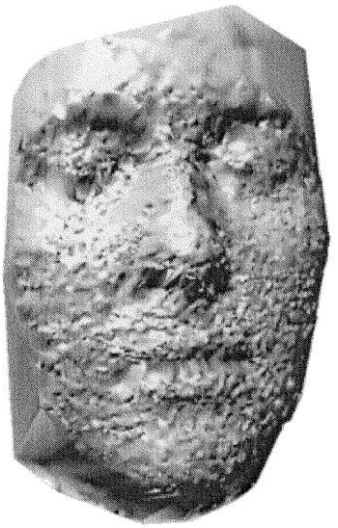

# Altri approcci

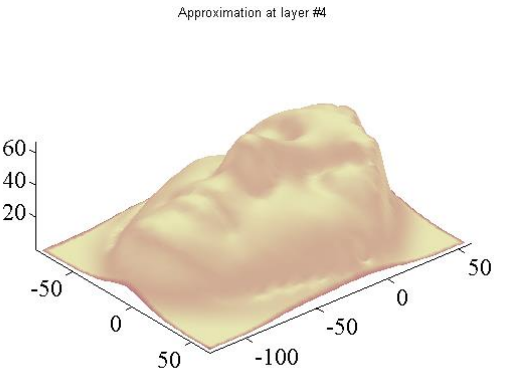
**Semi-convergenza**: non porto l'algoritmo fino alla convergenza nel punto di ottimo ma arresto le iterazioni prima.

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



# Problema dell'overfitting dovuto a sovraparametrizzazione






Quante unità?



# I vari tipi di apprendimento

$x(t+1) = f[x(t), a(t)]$ Ambiente

$a(t) = \mathbf{g}[x(t)]$ Agente

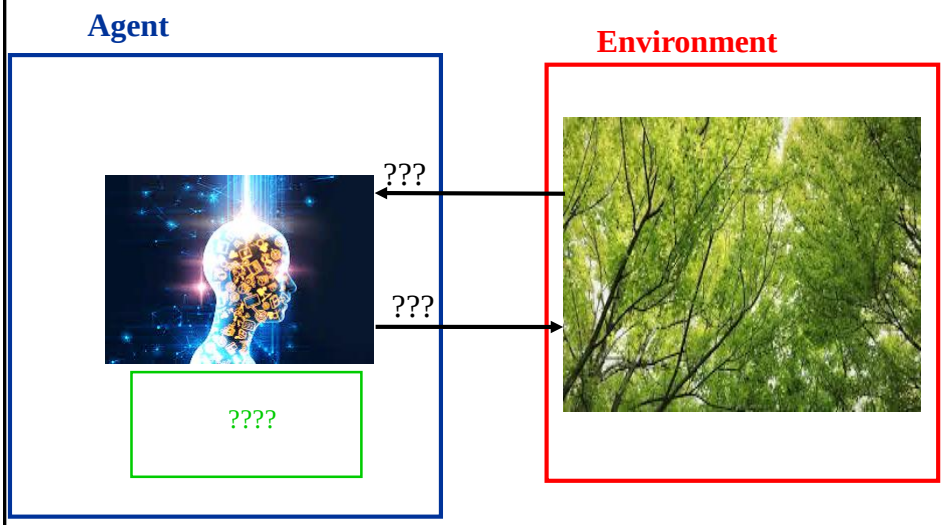
**Supervisionato** (learning with a teacher). Viene specificato per ogni pattern di input, il pattern desiderato in output.

**Semi-Supervisionato**. Viene specificato solamente per **alcuni** pattern di input, il pattern desiderato in output.

**Non-supervisionato** (learning without a teacher). Estrazione di similitudine statistiche tra pattern di input. Clustering. Mappe neurali.

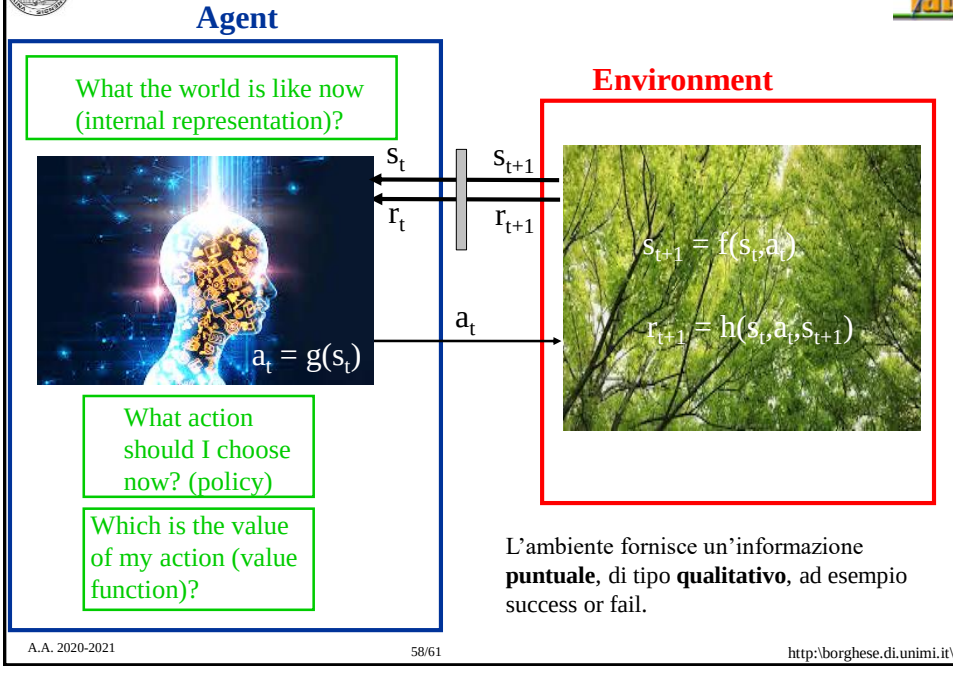
**Apprendimento con rinforzo** (reinforcement learning, learning with a critic). L'ambiente fornisce un'informazione puntuale, di tipo qualitativo, ad esempio success or fail.

# Schematic diagram of an agent



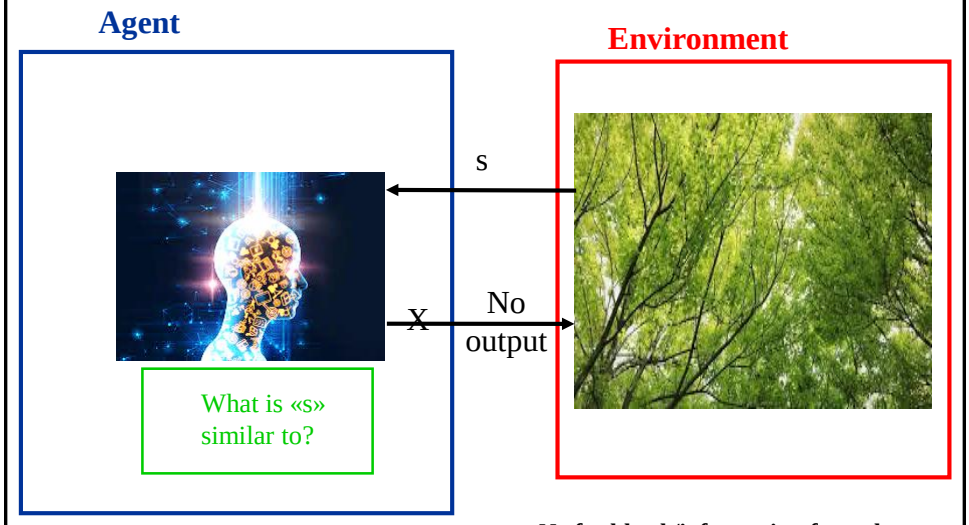



# Apprendimento con rinforzo




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

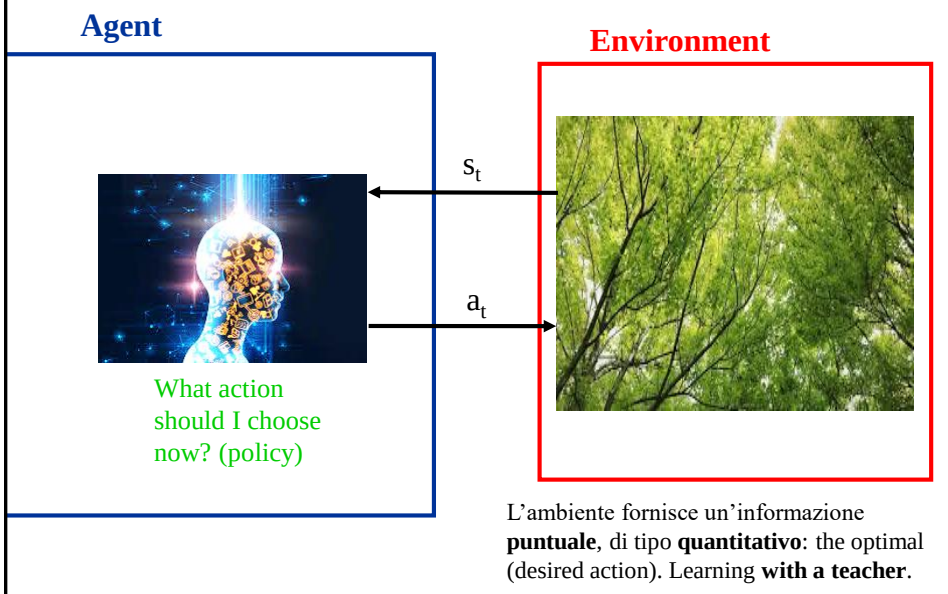
# Apprendimento non supervisionato




**No feed-back/information from the environment**



# Apprendimento supervisionato




L'ambiente fornisce un'informazione **puntuale**, di tipo **quantitativo**: the optimal (desired action). Learning **with a teacher**.

# Riassunto

- Supervised learning: predictive regression.
- Regressione multi-scala
- Versione on-line
- Valutazione del modello