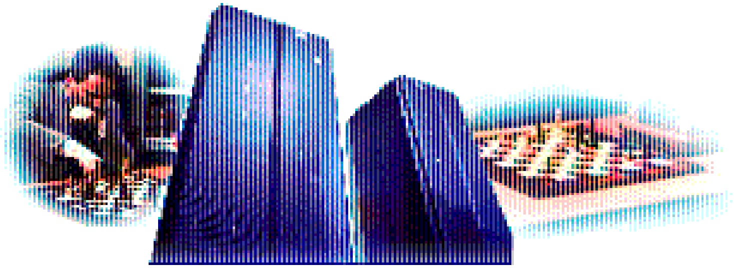
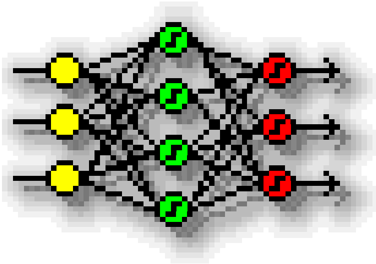




Sistemi Intelligenti
Soft Clustering

Alberto Borghese

Università degli Studi di Milano

Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)

Dipartimento di Informatica

alberto.borghese@unimi.it

Riassunto

- Soft Clustering
- Mappe di Kohonen e altri utilizzi del clustering
- I modelli

K-means: addestramento

Inizializzazione $\mathbf{C}_j$

↓

Classificazione $\mathbf{X}_i$

↓

Aggiornamento $\mathbf{C}_j$

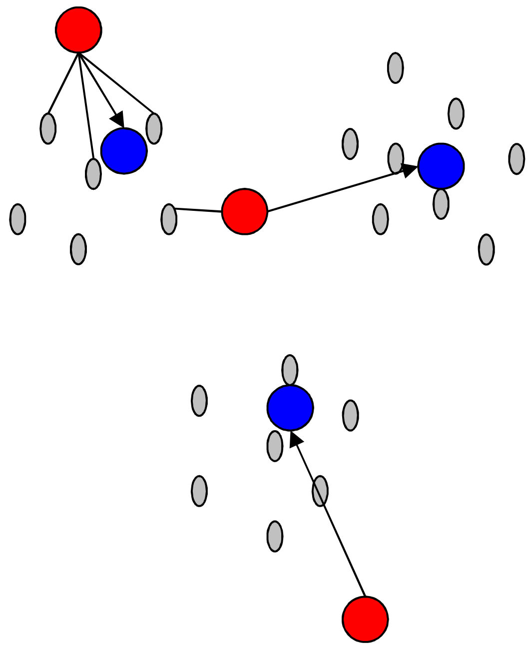
↓

I prototipi $\mathbf{C}_j$ si sono spostati significativamente?

SI' → Classificazione $\mathbf{X}_i$

NO ↓

Termine addestramento



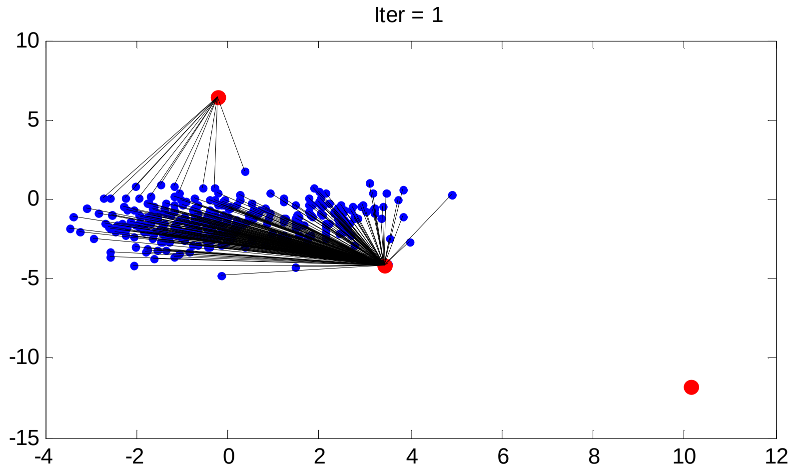
Aggiornamento $\mathbf{C}_j$: baricentro degli $\mathbf{X}_i$ classificati da $\mathbf{C}_j$.

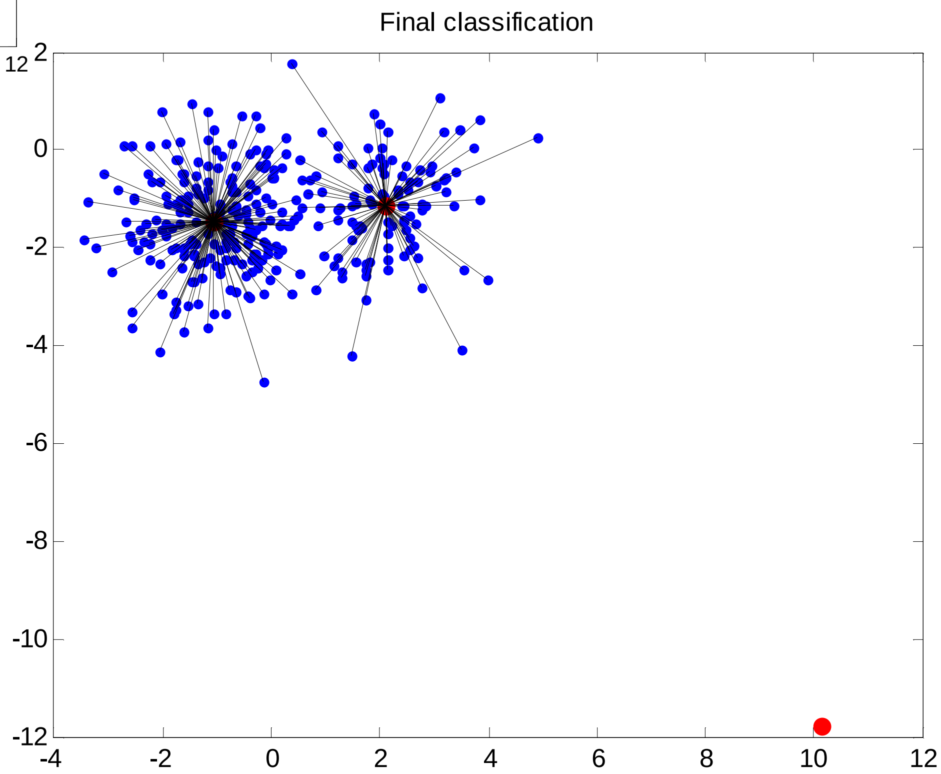
K-means::limiti

- Partitional, hard, deterministic;
- Veloce, semplice da implementare;
- Trova un minimo locale della funzione $f=\Sigma_j\Sigma_i[dist(x_i,prot_j)] / N_j$;
- Il risultato dipende dall'inizializzazione!
- Possono essere usati altri metodi (es. GA) per inizializzare K-means… es. GA per la minimizzazione di f, effettuano una ricerca globale, ma sono lenti!

Sw in Matlab available

Bad initialization





Principles of soft-clustering

- I centroidi vengono spostati e non posizionati
- Lo spostamento dei centroidi avviene analizzando iterativamente tutti i dati
- Lo spostamento viene ridotto via via che l'apprendimento procede

Competitive learning

- All'interazione k- esima, si presenti al sistema **un (1)** dato, $\mathbf{X_i}$;

- Aggiornamento di tutti i prototipi $\mathbf{W}_j$ ("neuroni")

- Generalized competitive Learning Rule:
  - $\Delta \mathbf{W_j} = \eta_k \Lambda_k(i,j) (\mathbf{X_i} - \mathbf{W_j})$

AGGIORNAMENTO PESI (POSIZIONE) DEI NEURONI

$\Lambda_k(i,j)$ è una funzione "campo recettivo"

- $\Lambda_k(i,j) = \exp(-||\mathbf{X_i} - \mathbf{W_j}||^2 / 2\sigma_k^2)$
  - (spazio dei dati
- $\Lambda_k(i,j) = \exp(-||f(\mathbf{X_i}) - f(\mathbf{W_j})||^2 / 2\sigma_k^2)$
  - (spazio delle feature)

FUNZIONI DI VICINATO

Competitive learning

- All'interazione k- esima, si presenti alla rete **un (1)** dato $\mathbf{X_i}$; al termine dell'apprendimento.

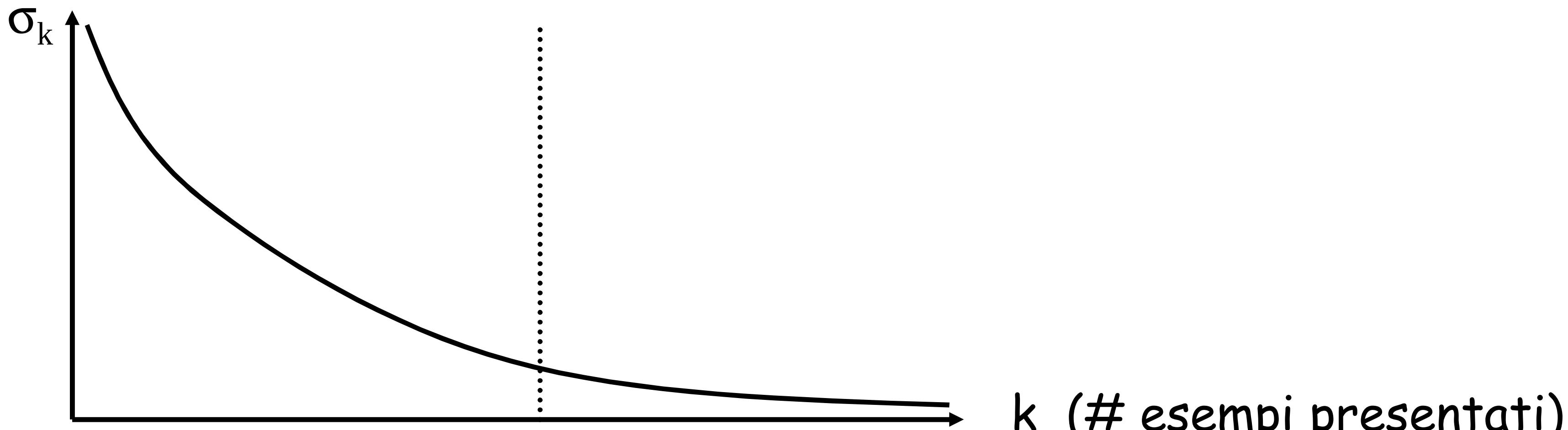
- Unità vincente (associazione):

  j^* t.c. $||\mathbf{W_{j^*}} - \mathbf{X_i}|| = \min_j ||\mathbf{W_j} - \mathbf{X_i}||$

UNITA' VINCENTE

Anche qui viene indotta una tessellazione di Voronoj dallo spazio da tutte le unità vincenti.

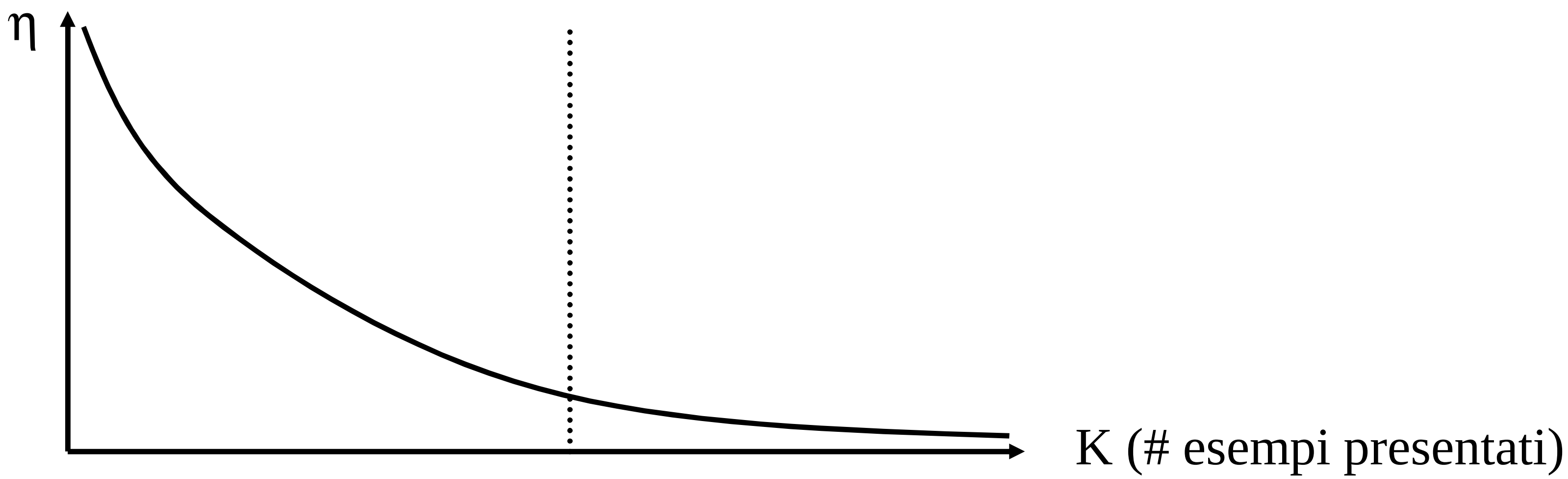
Funzione di vicinato nel tempo



$$\Delta \mathbf{W_j} = \eta_k \, \Lambda_k(i,j) \, (\mathbf{X_i} - \mathbf{W_j})$$

Procedendo nell'addestramento della rete, un neurone perde la capacità di spostare i suoi vicini.

Learning rate nel tempo



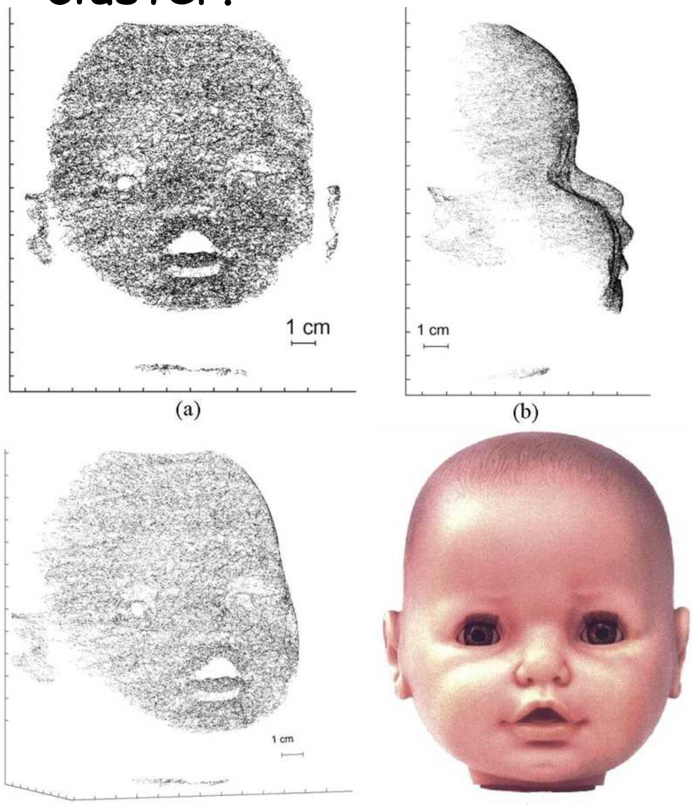
$$\Delta \mathbf{W_j} = \eta_k \Lambda_k(i,j) (\mathbf{X_i} - \mathbf{W_j})$$

Procedendo nell'addestramento della rete, i pesi dei neuroni perdono la possibilità di muoversi => rete più stabile.

Competitive learning
(“First search then converge”)

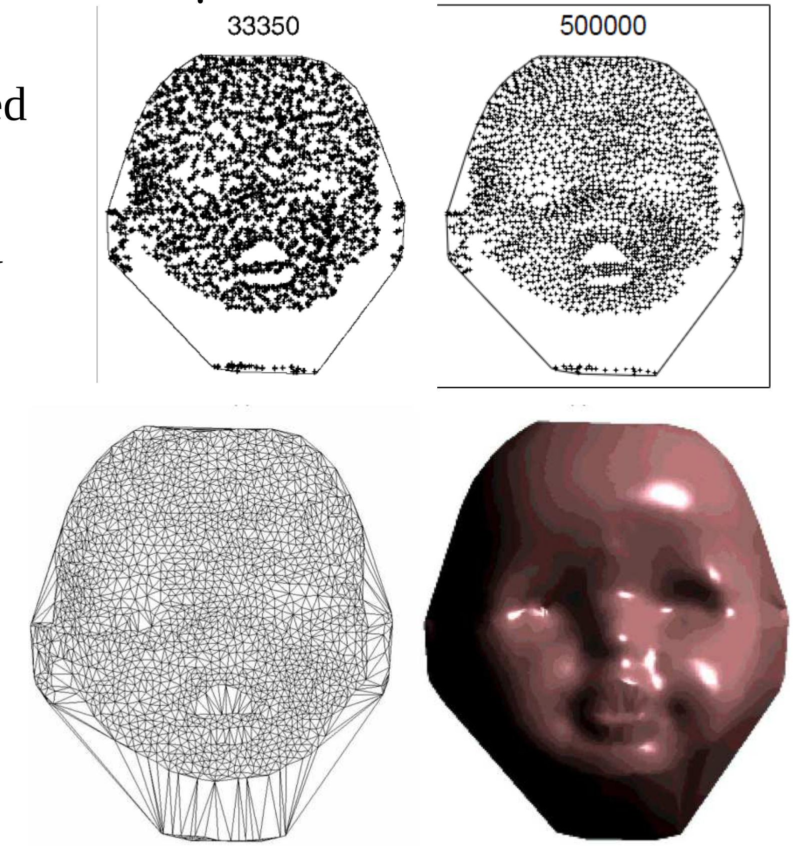
1) **ORDERING PHASE**: η, σ **grandi**; ogni neurone può spostarsi molto verso l’ingresso $\mathbf{X}_i$; il neurone trascina con sé i vicini; in tale fase la rete si dispiega nello spazio R^N “spargendo” i suoi neuroni.

2) **TUNING PHASE**: η, σ **piccoli**; ogni neurone si muove da solo; è una fase di raffinamento in cui vengono raggiunti con precisione i centri dei cluster.



(a) (b) (c) (d)

100,000 sampled points reconstructed with 2,000 RV



(c) (d)

Soft-clustering

$\Delta \mathbf{W}_j = \eta_k \Lambda_k(i,j) (\mathbf{X}_i - \mathbf{W}_j)$

$\Lambda_k(i,j)$ è l'elemento chiave. I "Campi recettivi" dei diversi neuroni sono parzialmente sovrapposti.

In "Competitive clustering" $\Lambda_k(i,j)$ è una Gaussiana nello spazio dei dati o dei prototipi (mappe di Kohonen).

In "Neural-gas" $\Lambda_k(i,j)$ è una ranking function nello spazio dei dati e dei prototipi.

In "Fuzzy c-means" $\Lambda_k(i,j)$ è realizzata mediante membership function nello spazio dei dati.

I problemi del soft-clustering

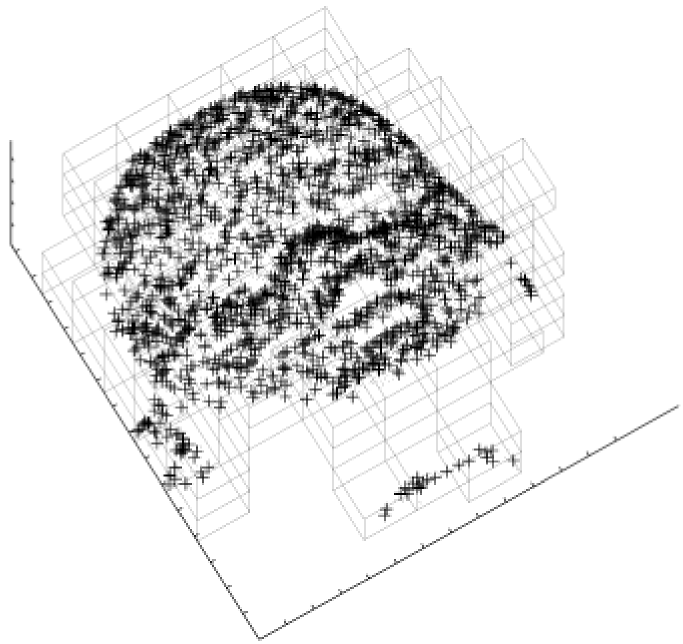
Dead-units: sono centroidi che non vengono aggiornati da un certo passo, k, in poi.

Inizializzazione guidata dai dati.

$$\rho_{Centroid} = \rho_{Data}{}^{\gamma} \quad with \quad \gamma = \frac{D}{D+2}$$

Partition of the input space and distribution of the number of centroids inside each box through a partitioning function:

$$M_k = M \frac{N_k{}^{\gamma}}{\sum_k N_k{}^{\gamma}}$$



Minimi locali.

Tipo di apprendimento

COMPETITIVE LEARNING. *Apprendimento competitivo. Dato un certo input, le unità* ***competono*** *tra loro per "aggiudicarsi" l'input.*

*Questo meccanismo può essere hard. Nel caso estremo: "**winner-take-all**", "spara" un solo neurone per volta (grandmother cell). Opppure può essere soft, le unità raggiungono un grado diverso di "vincita".*

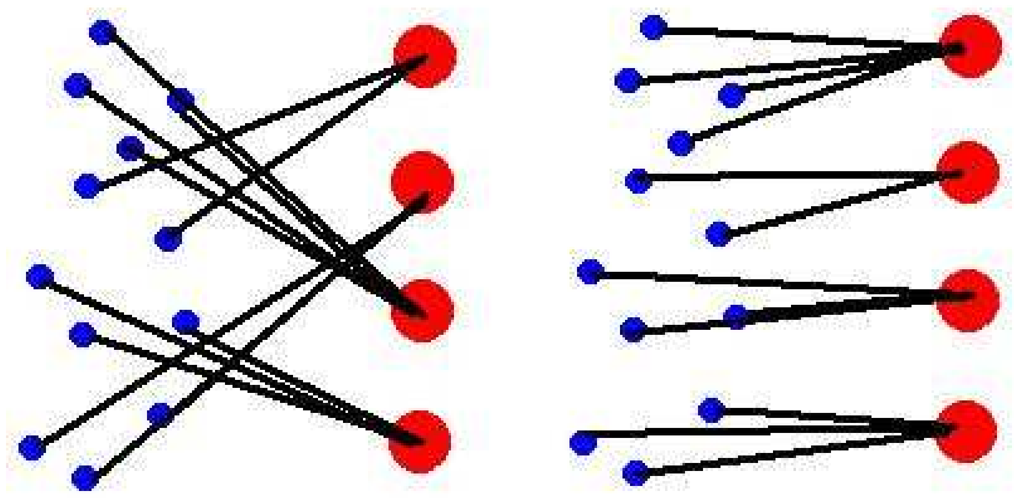
Winner-take-all ➔ *hard approach*

More than one winner → *soft approach*

Riassunto

- Soft Clustering
- Mappe di Kohonen e altri utilizzi del clustering
- I modelli

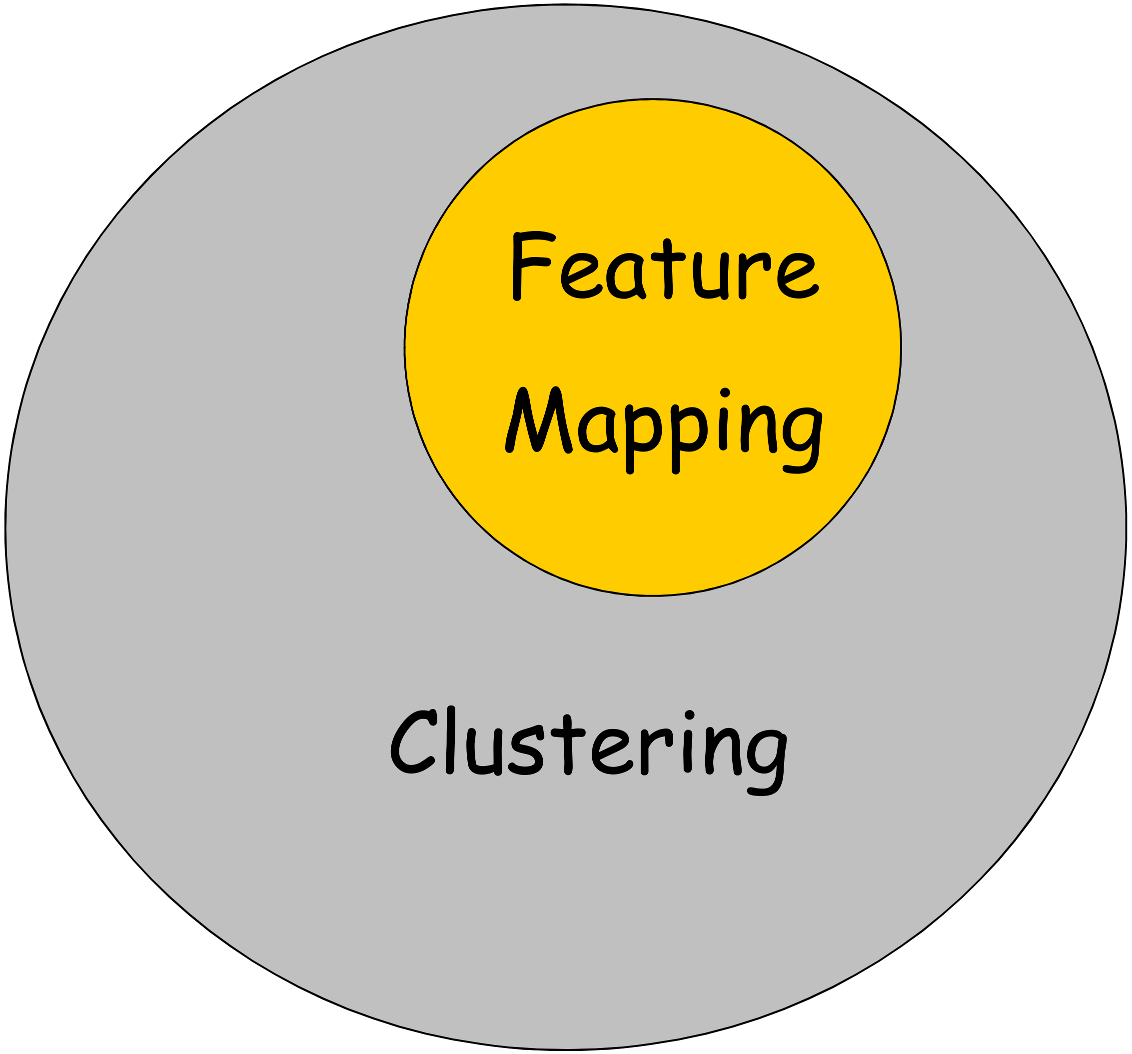
Clustering vs. feature mapping



Clustering   Feature Mapping

A differenza di quanto accade con il clustering, nel feature mapping <u>vengono preservate le relazioni tologiche</u> tra i dati.

Clustering e feature mapping

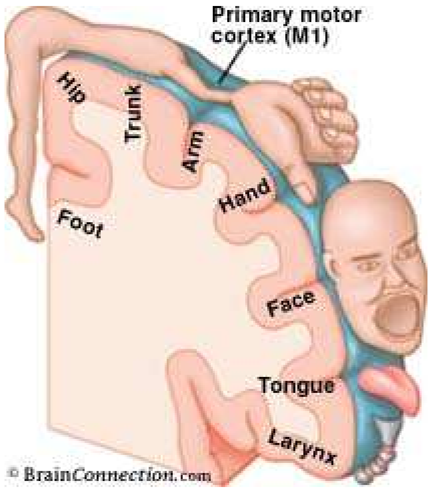


Feature mapping

Con il Feature Mapping si dà importanza alla posizione dei prototipi (xxx-topia, Homunculus motorio);

A uscite contigue corrispondono configurazioni d'ingresso contigue;

Il clustering opera una trasformazione tra lo spazio degli ingressi e lo spazio delle uscite (prototipi) che preserva le relazioni di vicinanza tra i vari elementi.



Self Organizing Maps (SOM)

MODELLI ISPIRATI ALLA BIOLOGIA

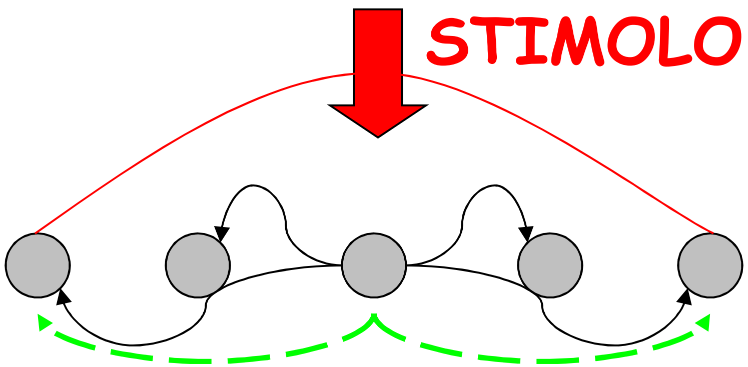
NO SUPERVISORE

Una *struttura neurale autoorganizzante* per il *feature mapping.*

XXX-TOPIA (COME NELLE STRUTTURE BIOLOGICHE)

L'ispirazione biologica : Il campo recettivo

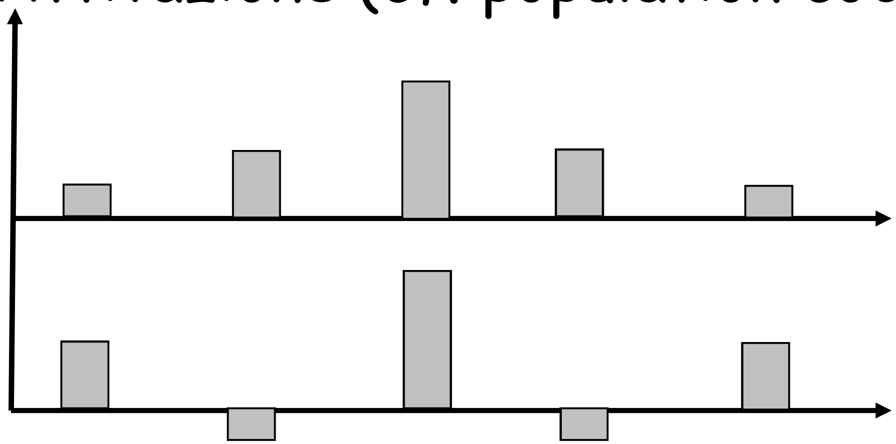
- **Sinapsi eccitatorie** verso i neuroni vicini;
- **sinapsi inibitorie** verso i neuroni lontani.



Sinapsi eccitatorie

Sinapsi inibitore

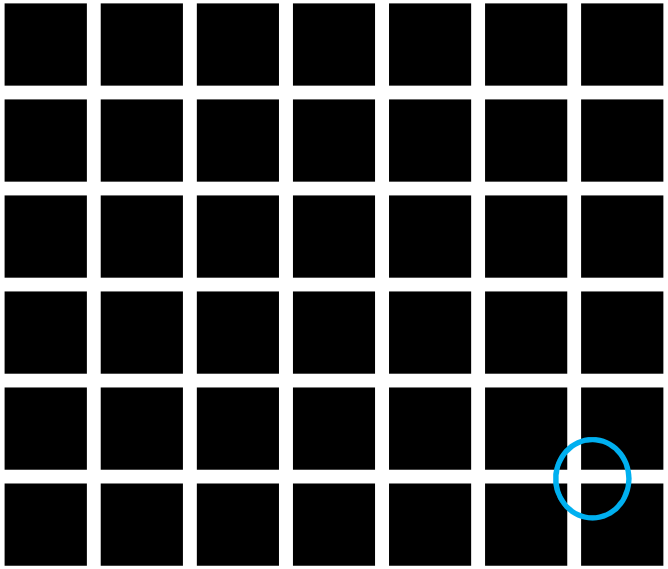
Attivazione (cf. population code)



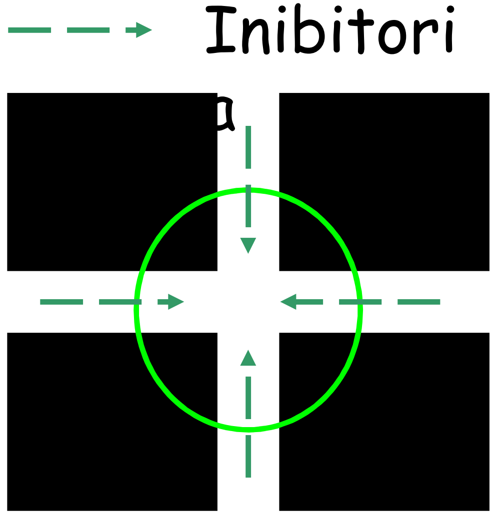
In assenza di sinapsi inibitorie

In presenza di sinapsi inibitorie

Il campo recettivo: effetto Hermann

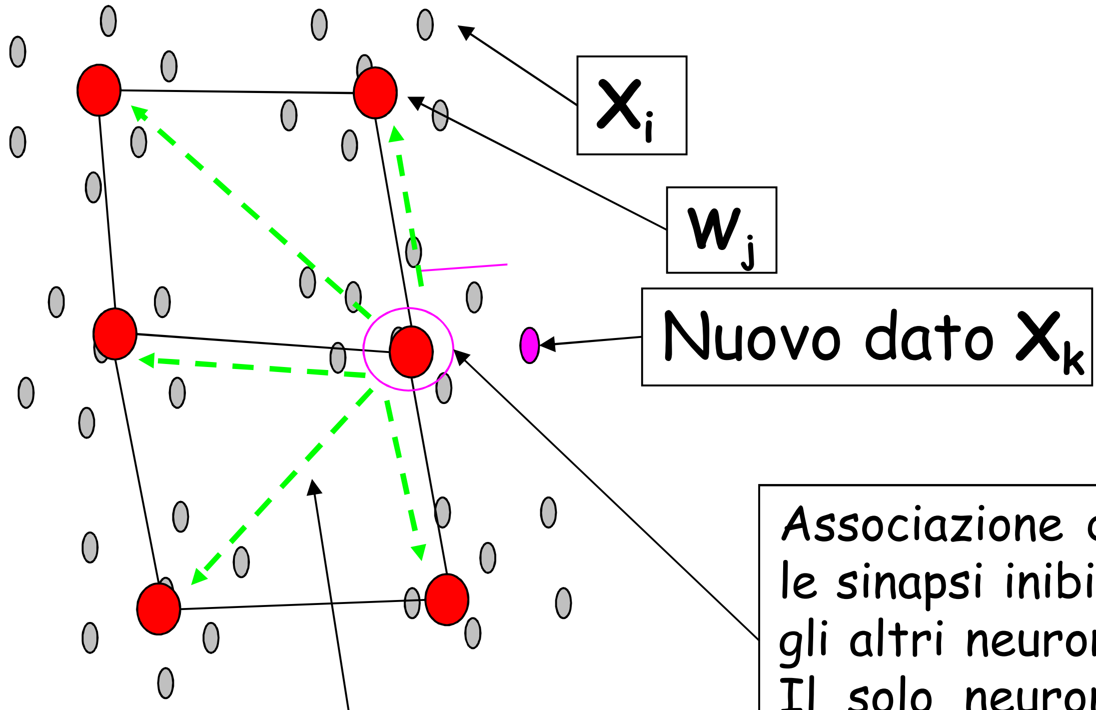


Pallini neri agli incroci delle linee bianche.



Il neurone centrale viene "spento" dai neuroni vicini attivi → generazione della zona scura.

SOM: competitive learning in versione "hard"



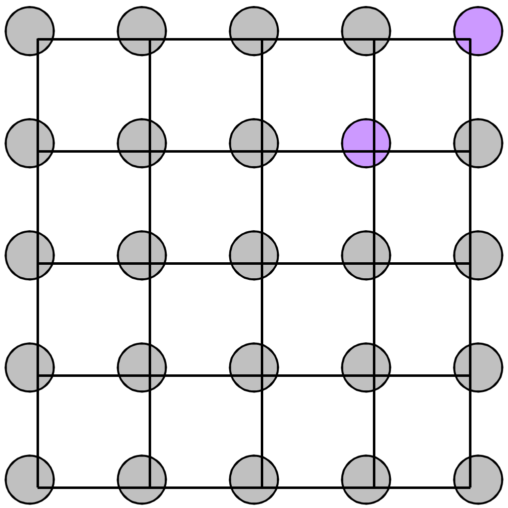
Associazione di $\mathbf{X_k}$ alla classe $\mathbf{W_l}$; le sinapsi inibitorie spengono tutti gli altri neuroni (*winner takes all*). Il solo neurone che "spara", $\mathbf{W_l}$, individua la classe di $\mathbf{X_k}$.

Connessione inibitoria
"spegne gli altri neuroni"

Mapping $R^N \rightarrow R^M$, con $M \leq N$ (da spazio delle caratteristiche a spazio delle classi).

SOM: organizzazione topologica

I neuroni della SOM sono ordinati topologicamente nello spazio dei neuroni (es. griglia ordinata in R^2). In tale spazio viene definita la distanza tra neuroni.



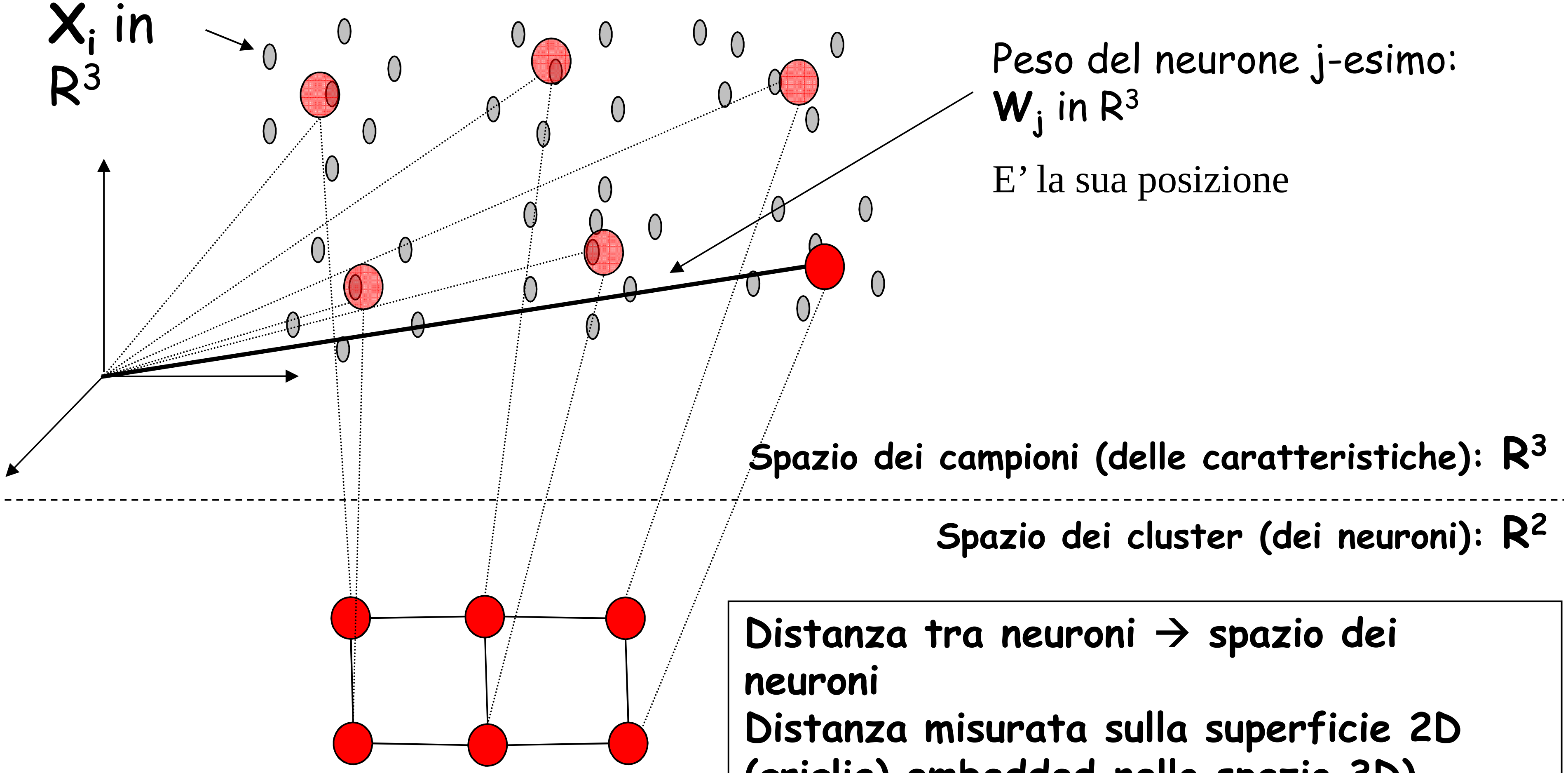
La distanza tra i due neuroni (nello spazio dei neuroni) è:

$\sqrt{(\Delta x^2 + \Delta y^2)} = 1.4142$ [Metrica Euclidea]

$|\Delta x| + |\Delta y| = 2$ [Manhattan]

...

SOM: pesi dei neuroni



Distanza tra neuroni → spazio dei neuroni
Distanza misurata sulla superficie 2D (griglia) embedded nello spazio 3D)

SOM: addestramento

- Siano $\mathbf{X_1}$, …, $\mathbf{X_D}$ i dati di addestramento (per semplicità, definiti in R^3);

- siano $W_\mathbf{1}$, …, $W_\mathbf{K}$ i *prototipi* di K classi, definiti anch'essi in R^3; ogni *prototipo* identifica il peso di un neurone della SOM ($\mathbf{W_j}$);

- lo schema di associazione adottato sia il seguente: "$\mathbf{X_i}$ appartiene a $W_\mathbf{j}$ se e solo se $W_\mathbf{j}$ è il *prototipo (peso del neurone)* più vicino a $\mathbf{X_i}$, nello spazio dei campioni (delle caratteristiche, R^3)";

- l'algoritmo di addestramento permette di determinare i pesi dei neuroni (le posizioni dei *prototipi*) $W_\mathbf{j}$ mediante successive approssimazioni;

- L'algoritmo di addestramento tiene conto della topologia dei neuroni nello spazio dei neuroni (*feature mapping*).

SOM: output

- All’interazione k- esima, si presenti alla rete il dato $\mathbf{X_i}$;
- Unità vincente (associazione):
  j^* t.c. $\|\mathbf{W_{j^*}}-\mathbf{X_i}\| = \min_j \|\mathbf{W_j} - \mathbf{X_i}\|$
- Uscita:
  - $u_{j^*} = 1$, se $j = j^*$
  - $u_j = 0$ se $j != j^*$

UNITA’
VINCENTE

SOM e competitive learning

- All'interazione k- esima, si presenti alla rete il dato $\mathbf{X_i}$;

- Unità vincente (associazione):

  j^* t.c. $\|\mathbf{W_{j*}}-\mathbf{X_i}\| = \min_j \|\mathbf{W_j} - \mathbf{X_i}\|$

UNITA' VINCENTE

- Competitive Learning Rule (SOM, Kohonen '81):
  - $\Delta \mathbf{W_j} = \eta_k \Lambda_k(j^*,j)\,(\mathbf{X_i} - \mathbf{W_j})$

    AGGIORNAMENTO PESI DEI NEURONI

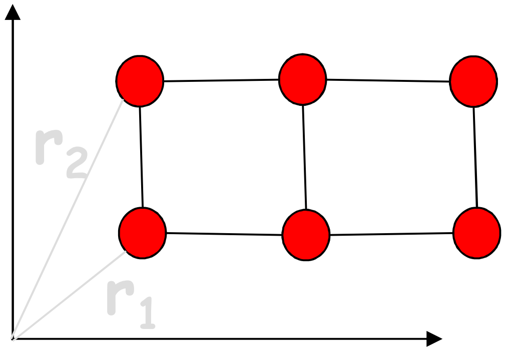
  - $\Lambda_k(j^*,j) = \exp(-\|\mathbf{W_{j*}} - \mathbf{W_j}\|^2 / 2\sigma_k^2)$

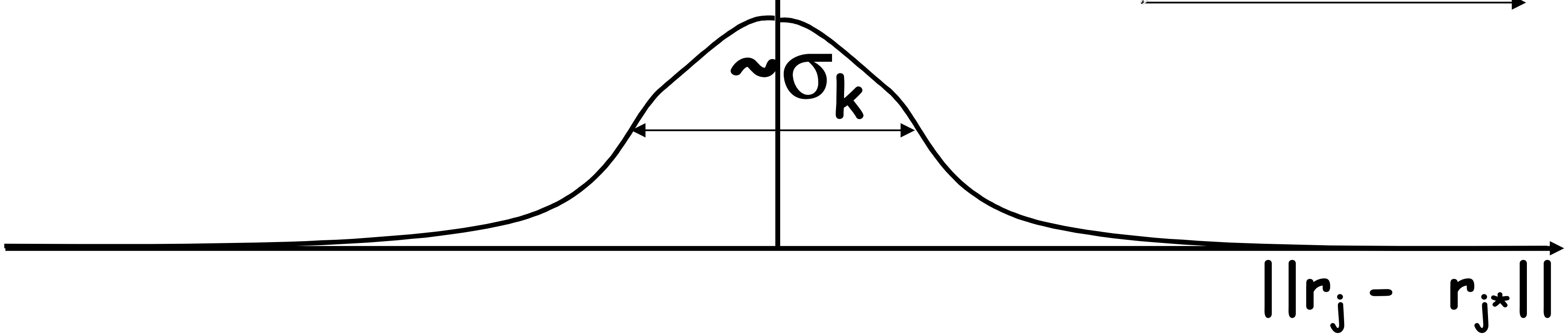
    FUNZIONE DI VICINATO

    Calcolata nello spazio dei prototipi

Funzione di vicinato

$$\Lambda_k(j,j^*) = \exp(-|\mathbf{r}_j - \mathbf{r}_{j^*}|^2 / 2\sigma_k^2)$$



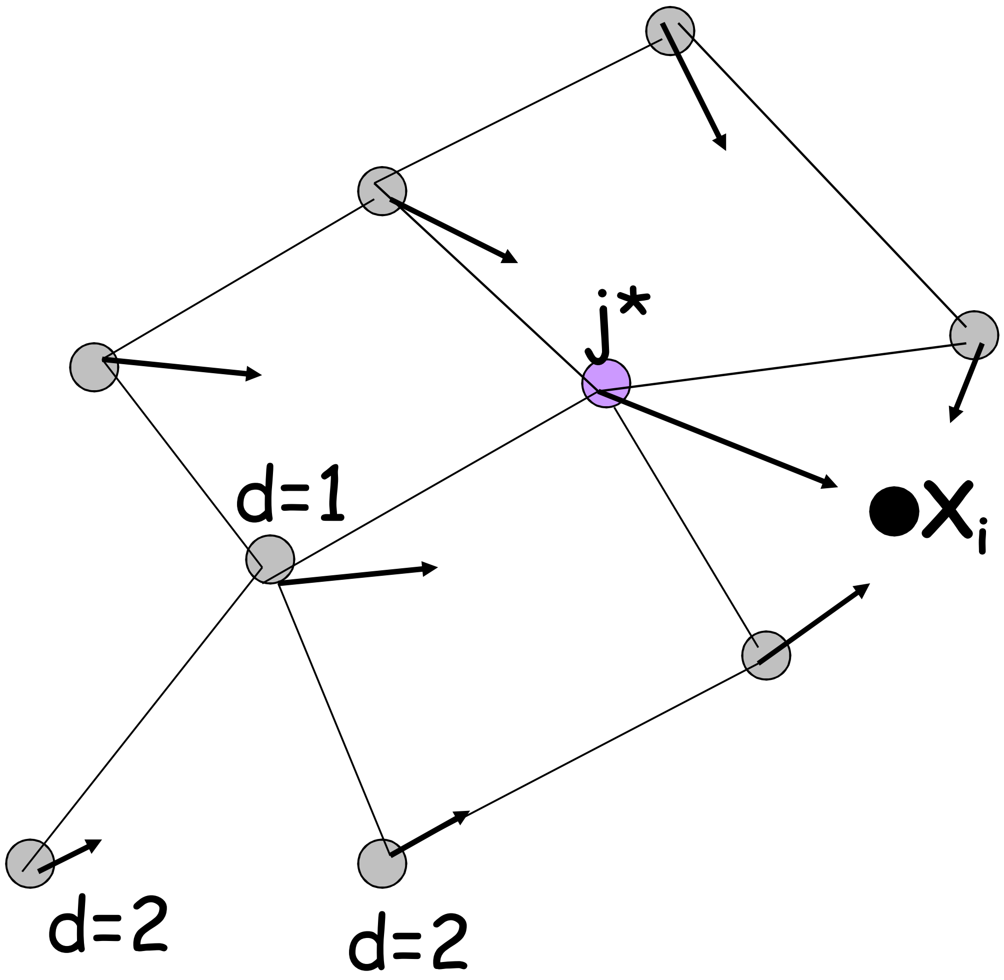


Neurone lontano dal neurone vincente: $\Lambda_k(j,j^*) \to 0$, $k \to oo$

$\sigma_k \to 0$, l'esponenziale va rapidamente a 0, solo il neurone vincente, j*, viene aggiornato.

$\sigma_k \to oo$ l'esponenziale rimane costantemente = 1. aggiorno tutti i neuroni di una quantità pari a: $\Delta \mathbf{W_j} = \eta_k (\mathbf{X_i} - \mathbf{W_j})$

SOM: addestramento

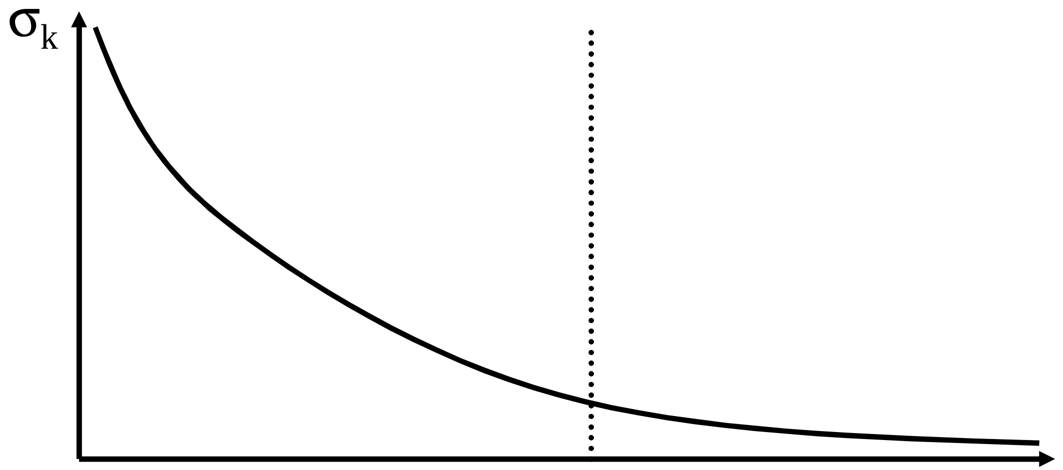


In definitiva:

-Il neurone vincente si sposta verso $\mathbf{X}_i$, trascinando i vicini in proporzione alla loro distanza nello spazio delle feature.

- L'ordinamento dei pesi dei neuroni nello spazio dei dati è simile all'ordinamento dei neuroni nello spazio dei neuroni.

Funzione di vicinato nel tempo

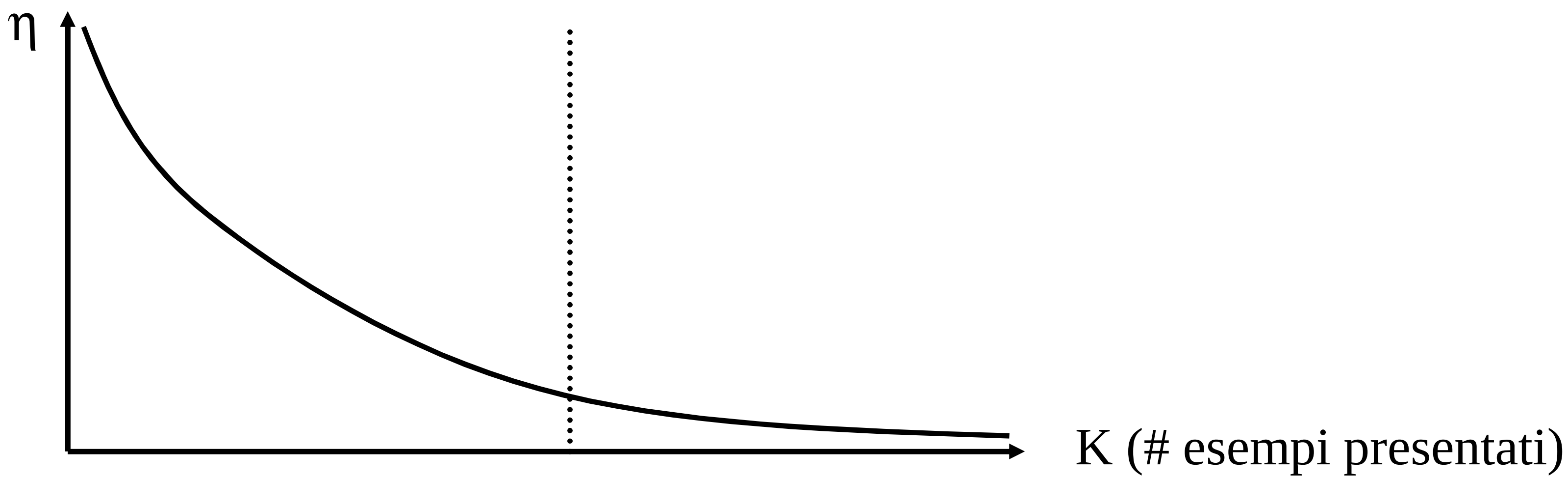


$$\Delta \mathbf{w_j} = \eta_k \Lambda_k(j,j^*) (\mathbf{X_i} - \mathbf{w_j}) \rightarrow \mathbf{w'} = \mathbf{w} + \eta_k \Lambda_k \mathbf{X_i} - \eta_k \Lambda_k \mathbf{w}$$

$$\Lambda_k(j,j^*) = \exp(-||\mathbf{r_j} - \mathbf{r_{j*}}||^2 / 2\sigma_k^2)$$

Procedendo nell'addestramento della rete, un neurone perde la capacità di spostare i suoi vicini.

Learning rate nel tempo

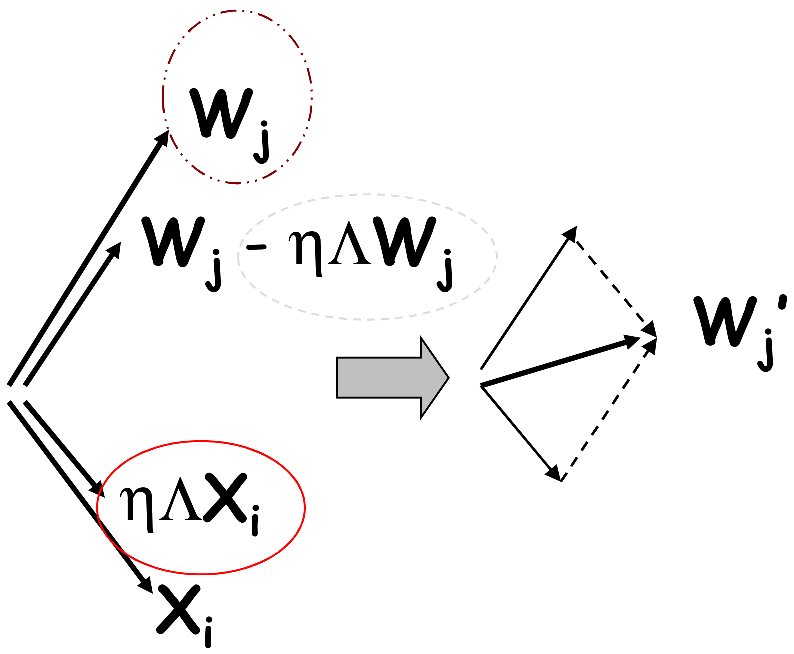


$$\Delta \mathbf{w_j} = \eta_k \Lambda_k(j,j^*)(\mathbf{X_i} - \mathbf{w_j}) \rightarrow \mathbf{w'} = \mathbf{w} + \eta_k\Lambda_k\mathbf{X_i} - \eta_k\Lambda_k\mathbf{w_i}$$

Procedendo nell'addestramento della rete, i pesi dei neuroni perdono la possibilità di muoversi => rete più stabile.

Competitive Learning come forze contrapposte

$$\Delta \mathbf{W}_j = \eta_k \Lambda_k(i,j)\,(\mathbf{X}_i - \mathbf{W}_j) \rightarrow \mathbf{W}_j' = \mathbf{W}_j + \eta_k\Lambda_k(i,j)\mathbf{X}_i - \eta_k\Lambda_k(i,j)\mathbf{W}_j$$



Il peso $\mathbf{W}_j$ del neurone vincente j si sposta verso l'ingresso presentato $\mathbf{X}_i$.

$-\eta\mathbf{W}_j$ evita che il peso $\mathbf{W}_j$ cresca a dismisura.

SOM: addestramento

- L'addestramento avviene presentando alla rete i vettori (dati) $\mathbf{X}_i \in R^N$ per un numero di epoche E;

- Per ogni esempio presentato $\mathbf{X}_i$ vengono aggiornati i pesi dei neuroni della rete;

- Durante l'addestramento il learning rate η e la neighborhood distance σ decrescono;

- Se presentiamo alla rete un nuovo esempio $\mathbf{X}_i$ alla fine dell'addestramento, la rete lo associa al neurone vincente (clustering);

In più:

- Categorie simili sono rappresentate da neuroni vicini (feature mapping).

SOM: problemi

-E' necessario scegliere η, σ, numero di epoche, durata della ordering phase $\rightarrow$ metodi empirici(!);

- Scelta della topologia e del numero di neuroni corretti;

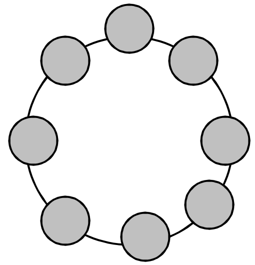
- I dati di addestramento devono presentare una certa ridondanza;

- Unità "morte" (dead-units);

SOM per ordinamento

Spazio dei dati $\mathbf{X}_i$ (e dei pesi $\mathbf{w}$) : R^3

Topologia della SOM : circolare



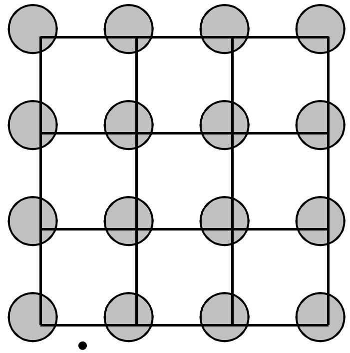
Parametri di addestramento : # neuroni, $\eta(t)$, $\sigma(t)$, ...

Movie here (with RealPlayer)

SOM per ricostruzione 3D

Spazio dei dati **q** (e dei pesi **w**) : R^3

Topologia della SOM : griglia 2D



Parametri di addestramento : # neuroni = 10x10, 0.5 $\rightarrow$ 0.1 lin, 10 $\rightarrow$ 1 lin, ...

Movie here

SOM per ricostruzione 3d

Problemi:

- Oscillazioni della rete all'inizio dell'addestramento;

Sol.: Scelta accurata di $\eta(t)$, $\sigma(t)$

- Raggiungimento dei confini della superficie aperta;

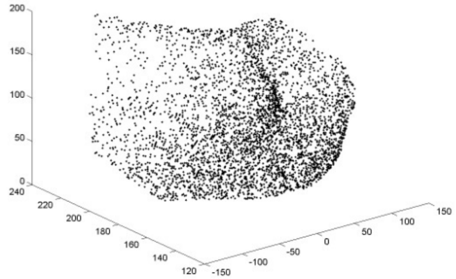
Sol.: Boundary First Method + η, σ modificati

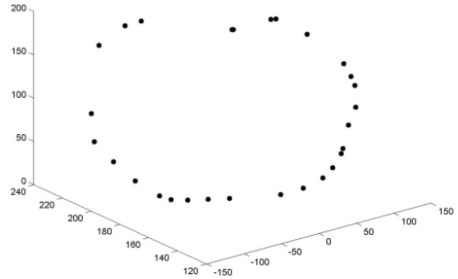
- Numero insufficiente di neuroni;

Sol.: Parametrizzazione della points cloud

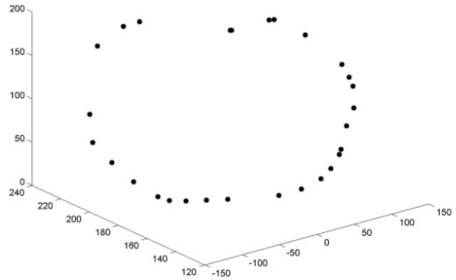
Boundary First Method

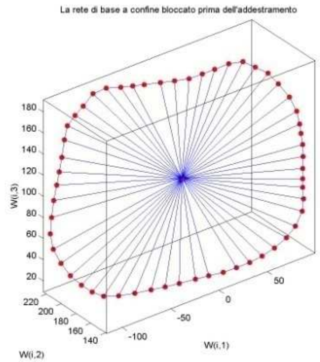
1) Individuazione dei punti di confine della superficie



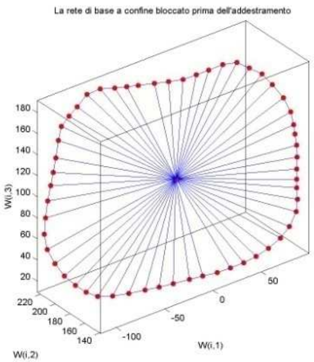


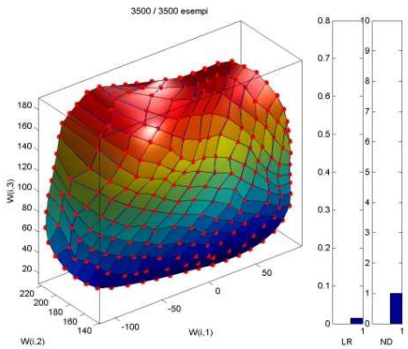
2) Posizionamento dei neuroni di bordo





3) Addestramento della SOM (neuroni di bordo bloccati), η,σ maggiori ai lati



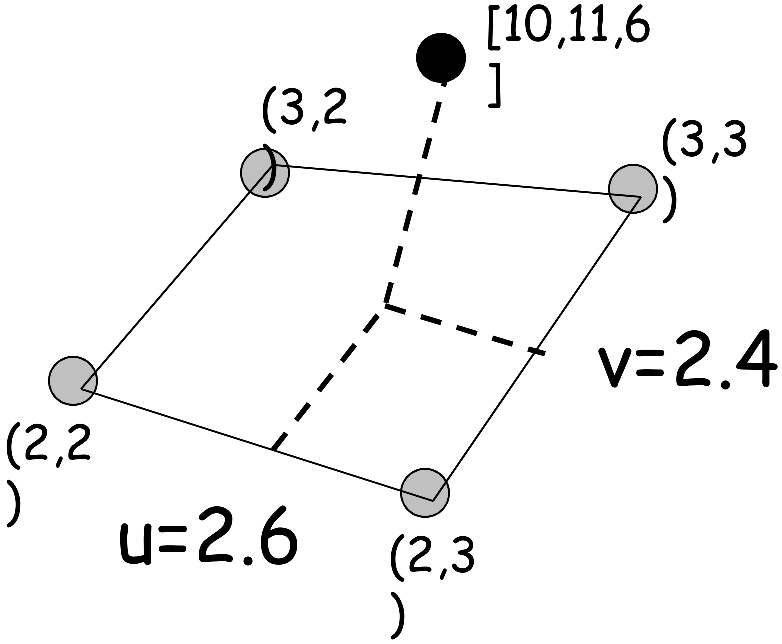


SOM per ricostruzione 3D

- Movie here
- Movie here

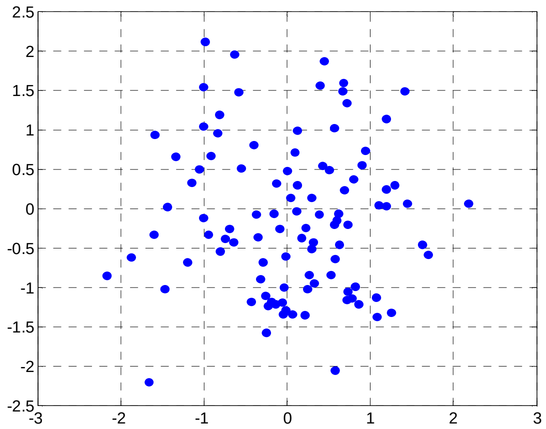
Parametrizzazione Points Cloud

- Ad ogni punto 3D [x,y,z] vengono assegnate le coordinate 2D [u,v] corrispondenti nello spazio 2D della SOM tramite una proiezione (parametrizzazione)



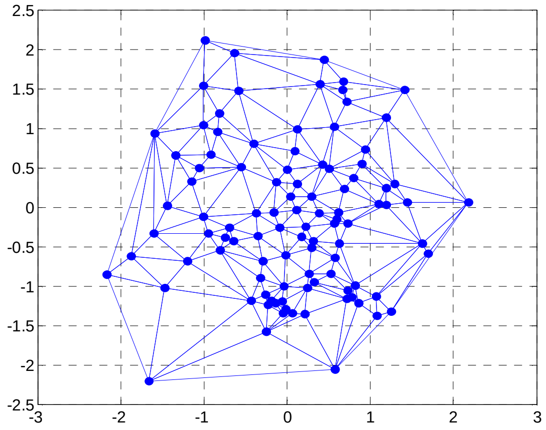
Triangolazione punti in [u,v] (Delaunay)

- Dato un set di punti nello spazio 2D, è possibile trovare un set di triangoli tale per cui (triangolazione di Delaunay):
  - ogni punto si trova al vertice di uno o più triangoli
  - nessun punto si trova all'interno di un triangoli
- Utile in computer graphics.



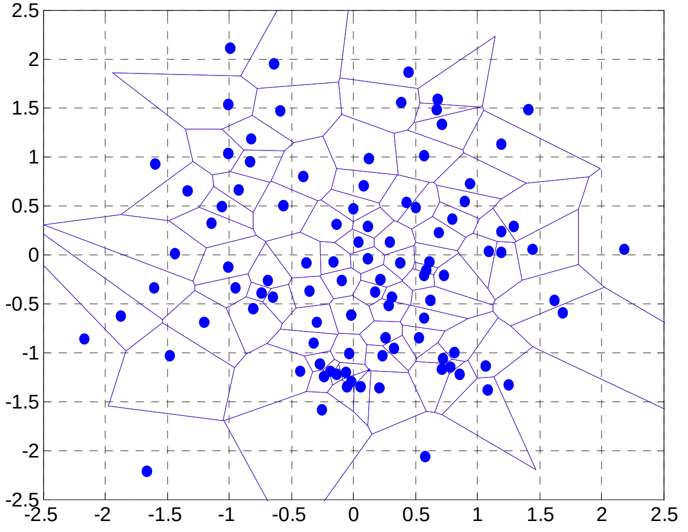
- In modo più rigoroso:

  a Delaunay triangulation for a set P of points in the plane is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). Delaunay triangulations **maximize the minimum angle of all the angles of the triangles** in the triangulation; they tend to **avoid "sliver" triangles**.

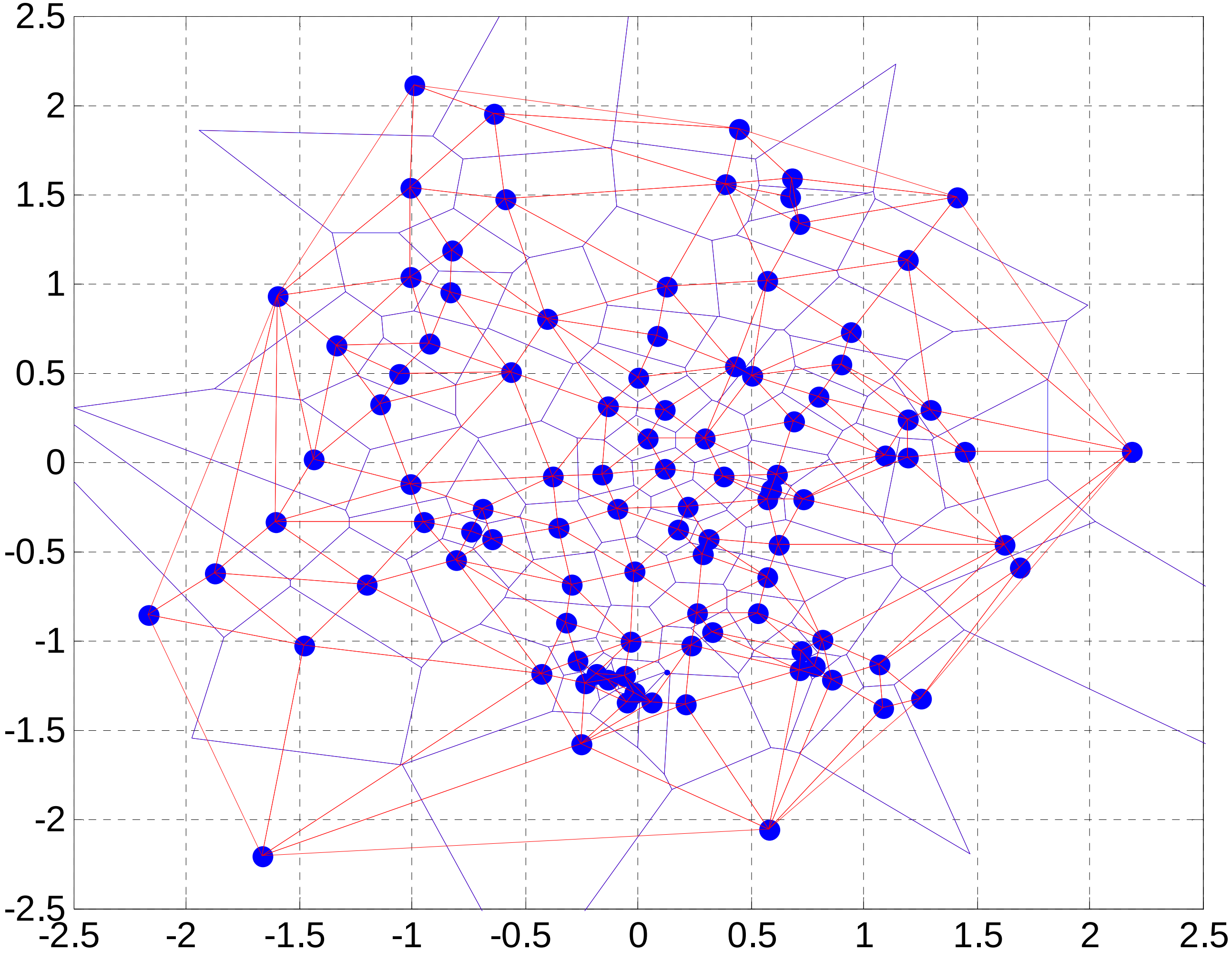


Delaunay e Voronoi

- Duale della triangolazione di Delaunay → Tessellazione di Voronoi.
- Suddivisione “regolare” dello spazio.

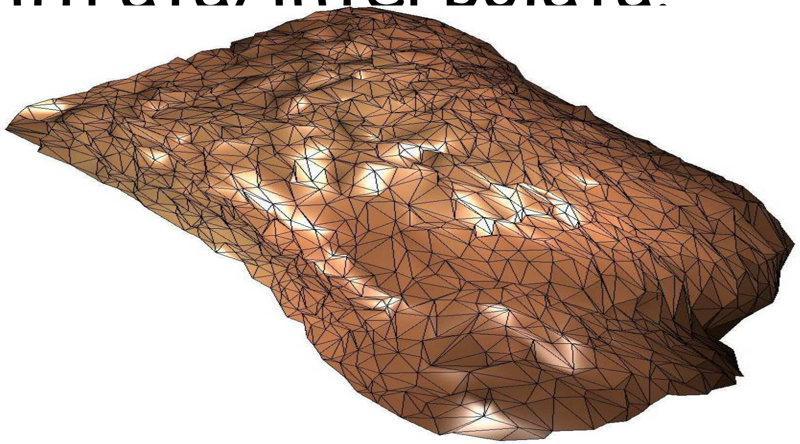


Dealunay & Voronoi

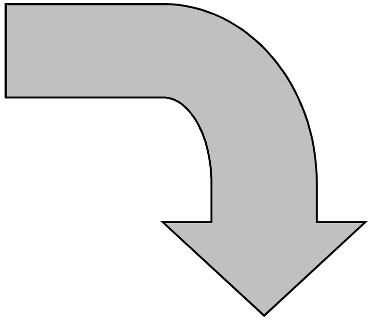


Parametrizzazione Points Cloud

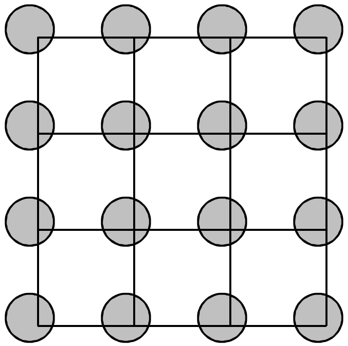
I punti 2D [u,v] possono essere triangolati (ricostruzione a mesh di triangoli). La mesh viene poi filtrata/interpolata.

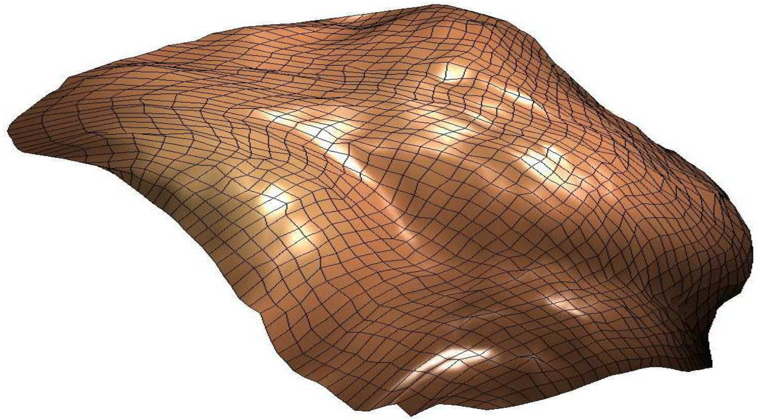


Ricostruzione a mesh di triangoli



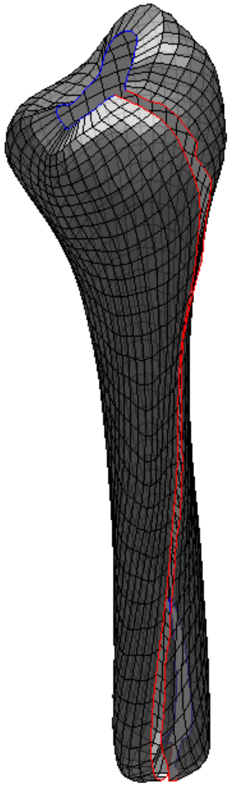
Filtraggio o interpolazione

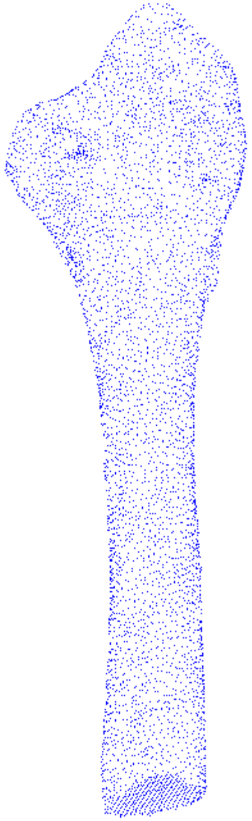


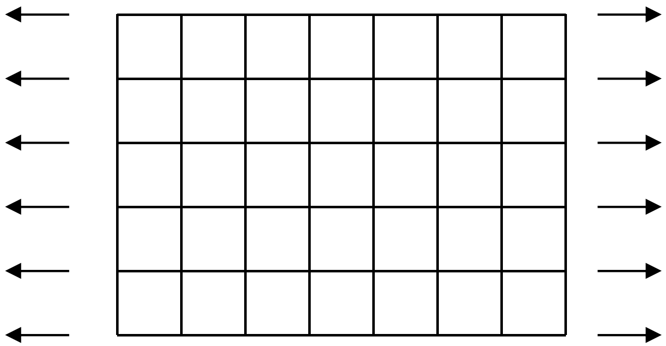


Ricostruzione Finale

Tibia Left

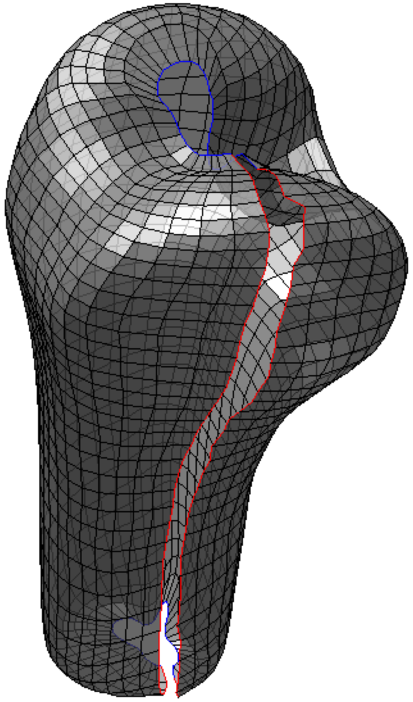


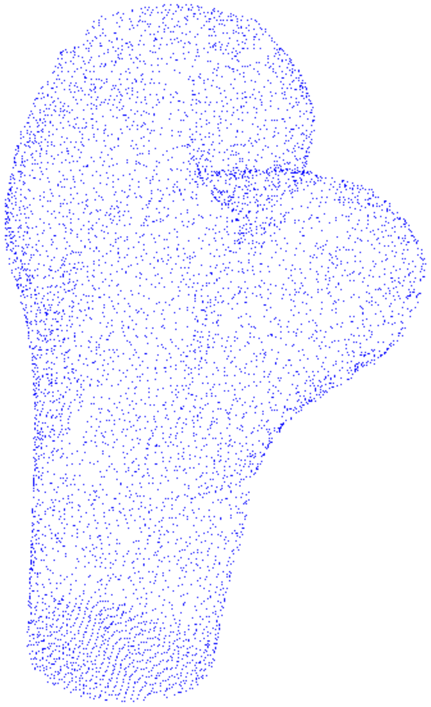


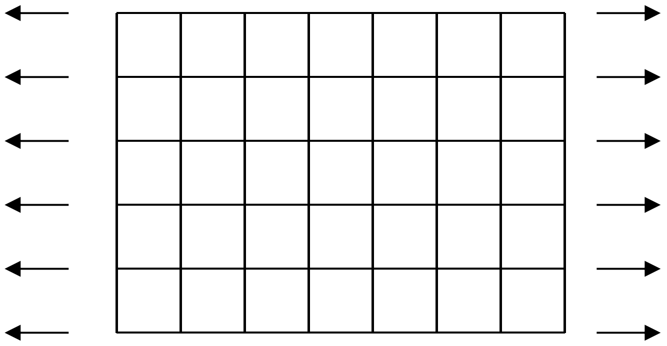


Topologia: cilindro

Femoral Bone Left







Topologia: cilindro

Variant SOM

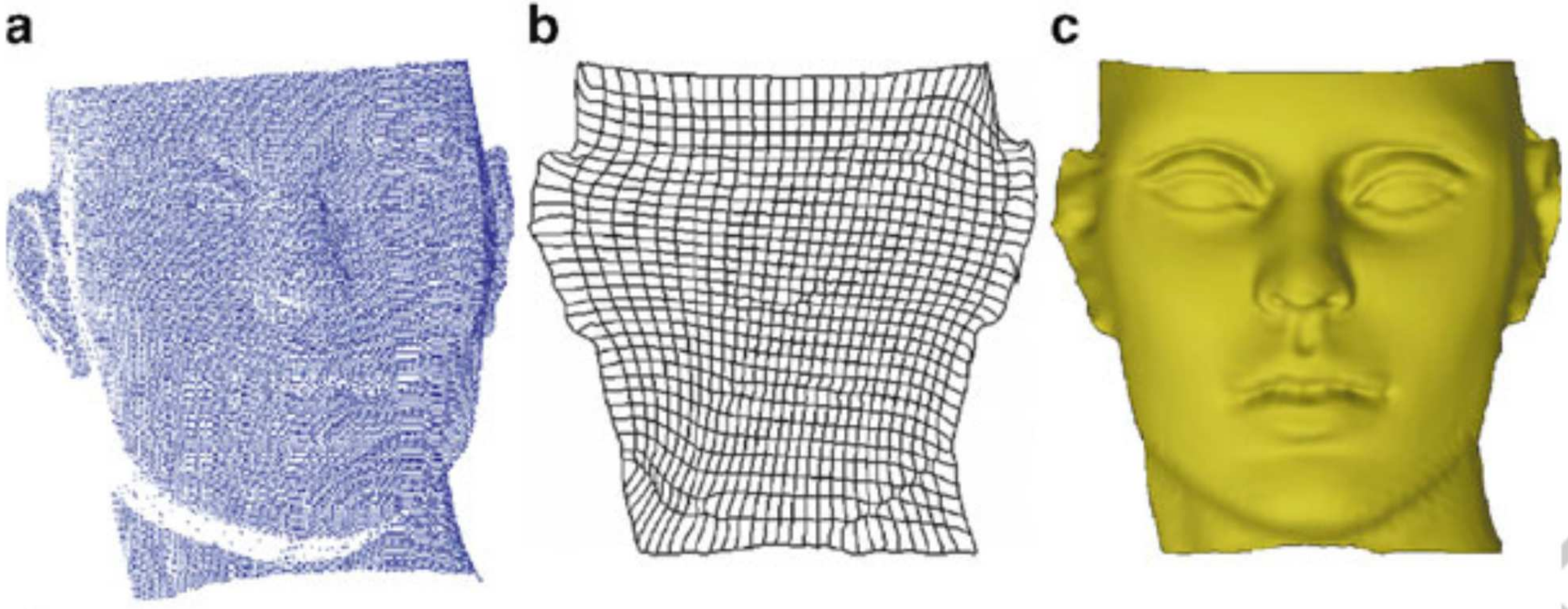
- SELF CREATING MAP
    - Aggiunta di un neurone:
    - Vicino al neurone vincente con f maggiore;
    - Vicino al neurone con curvatura massima;

-SOM SUPERVISIONATE (per la classificazione);

- ...

Self Organizing Maps approach Barhak et al., 2001

- A regular pre-defined mesh (map) is «attracted» and «deformed (warped)» by the points



$$w_j(i+1) = w_j(i) + \eta \cdot h_j(k) \cdot (p - w_j(i))$$

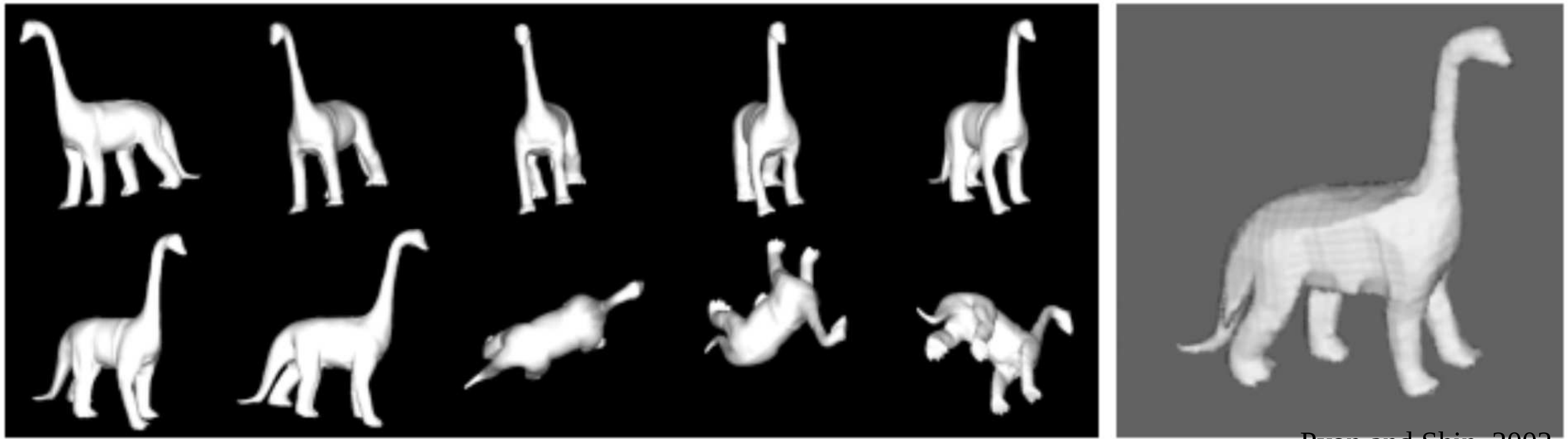
w grid intersection p data point h(k) neighbour region (k is its amplitude)

Other methods

Baloons

Level sets

Volume carving



Pyon and Shin, 2002

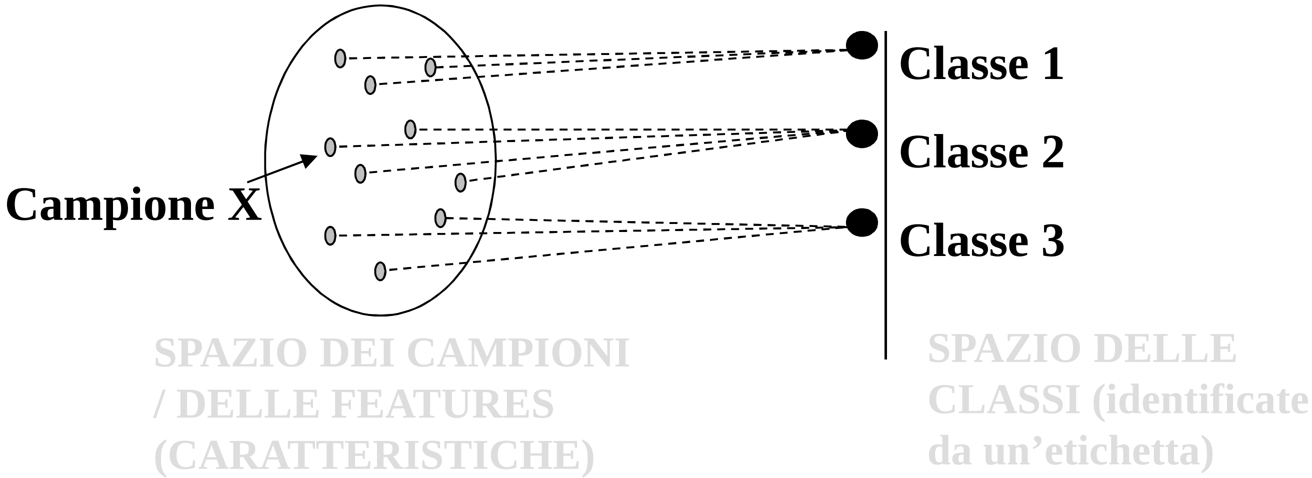
Kinect fusion = camera motion + 3D reconstruction

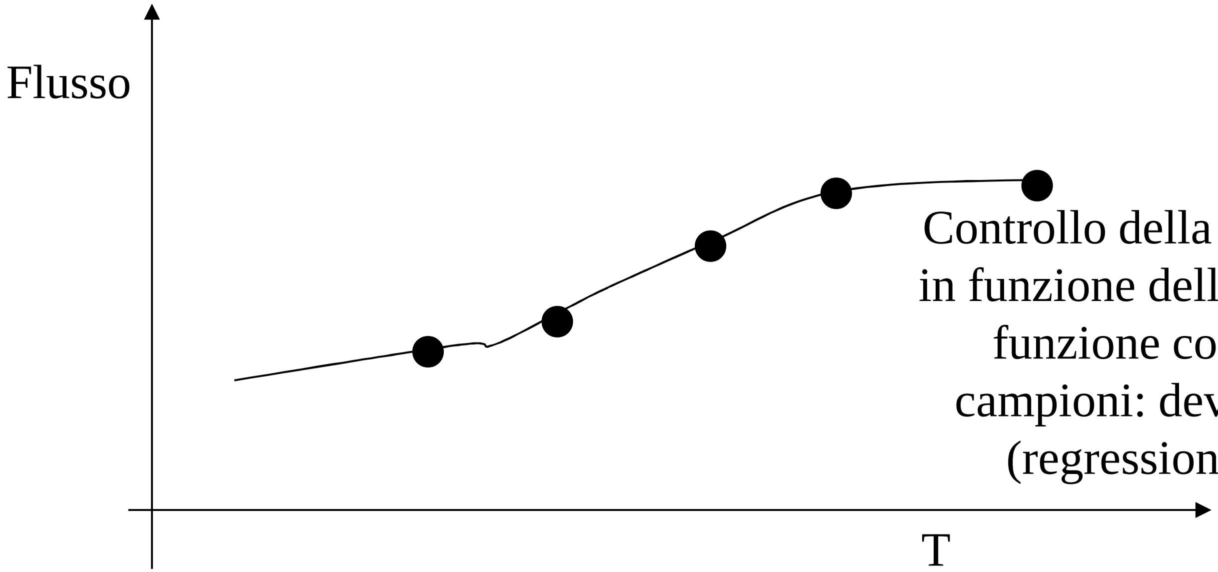
Riassunto

- Soft Clustering
- Mappe di Kohonen
- I modelli

Classificazione e regressione

Mappatura dello spazio dei campioni nello spazio delle classi.





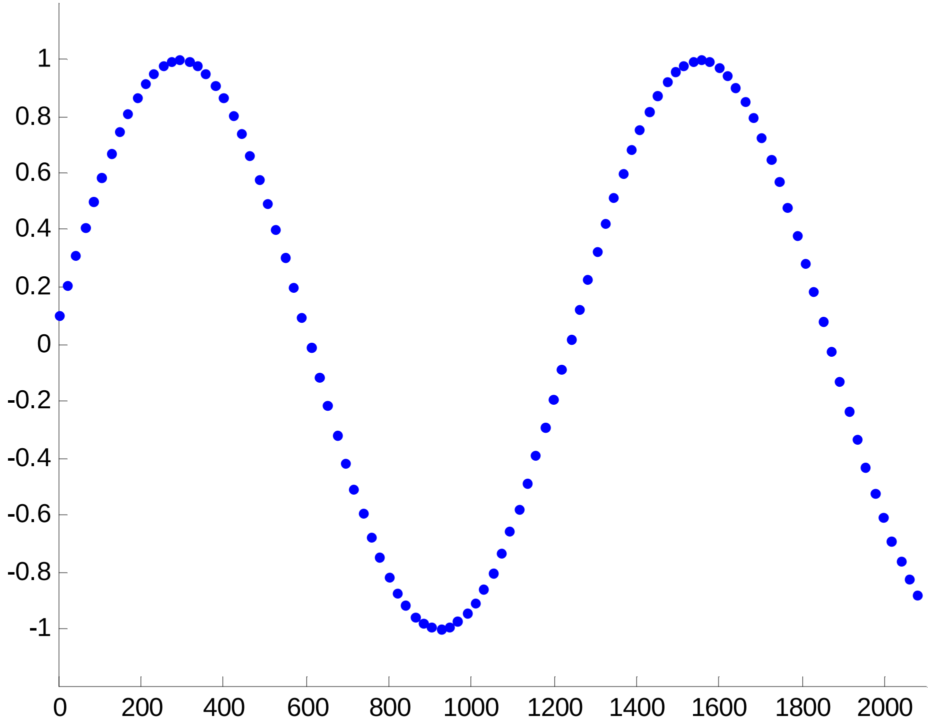
Controllo della portata di un condizionatore in funzione della temperatura. "Imparo" una funzione continua a partire da alcuni campioni: devo imparare ad **interpolare** (regressione = predictive learning).

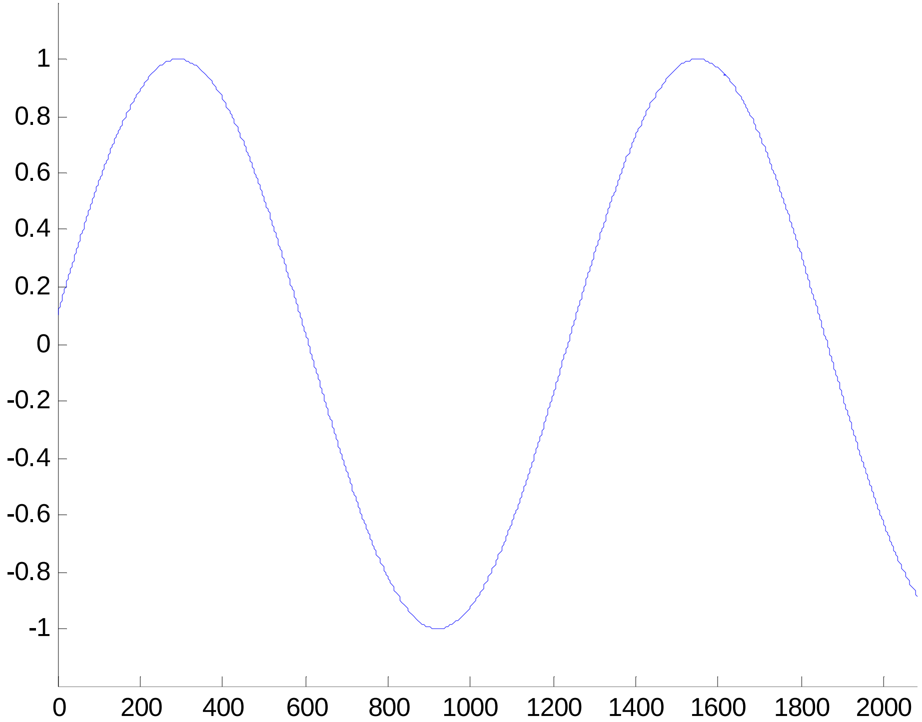
Ruolo dei modelli

- **Identificazione:** stimo i parametri di un modello a partire dai dati: identifico il modello.

- **Utilizzo:** utilizzo il modello per inferire informazioni su nuovi dati (controllo, regressione predittiva, classificazione).

Modello parametrico



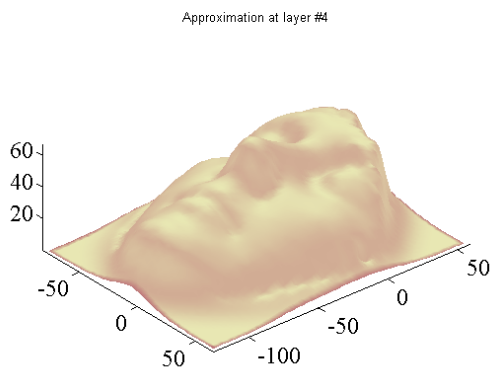


I punti vengono fittati perfettamente da una sinusoide: $y = \mathbf{A} \sin(\omega x + \phi)$. Devo determinare solo i 3 parametri della sinusoide (non lineare), i cui valori ottimali sono: $\omega = 1/200$, $\phi = 0.1$, $A = 1$. I parametri hanno un significato semantico.

I modelli semi-parametrici

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

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

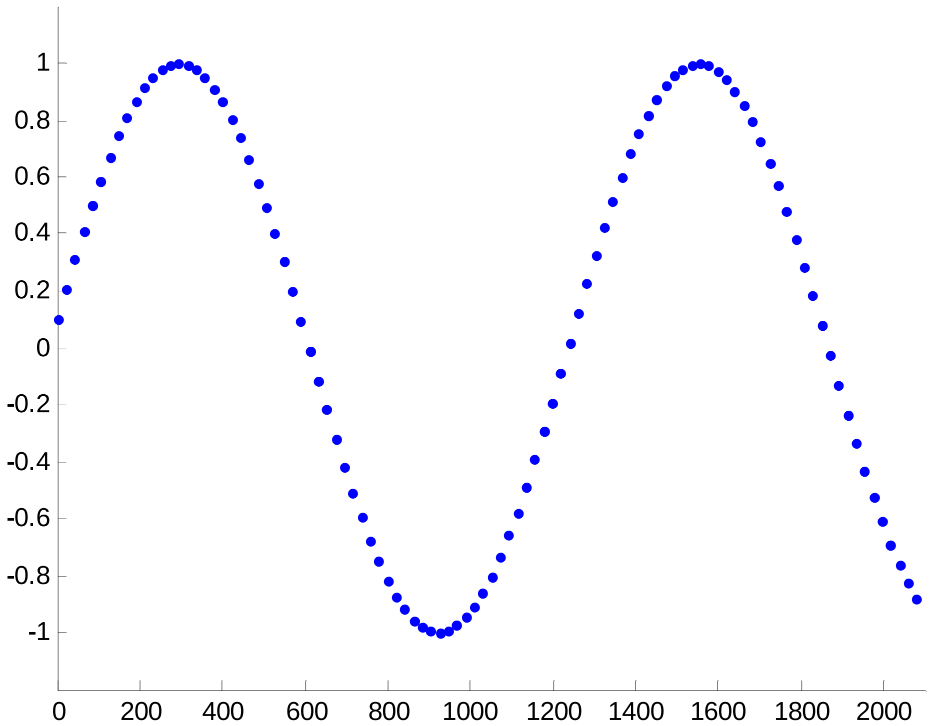


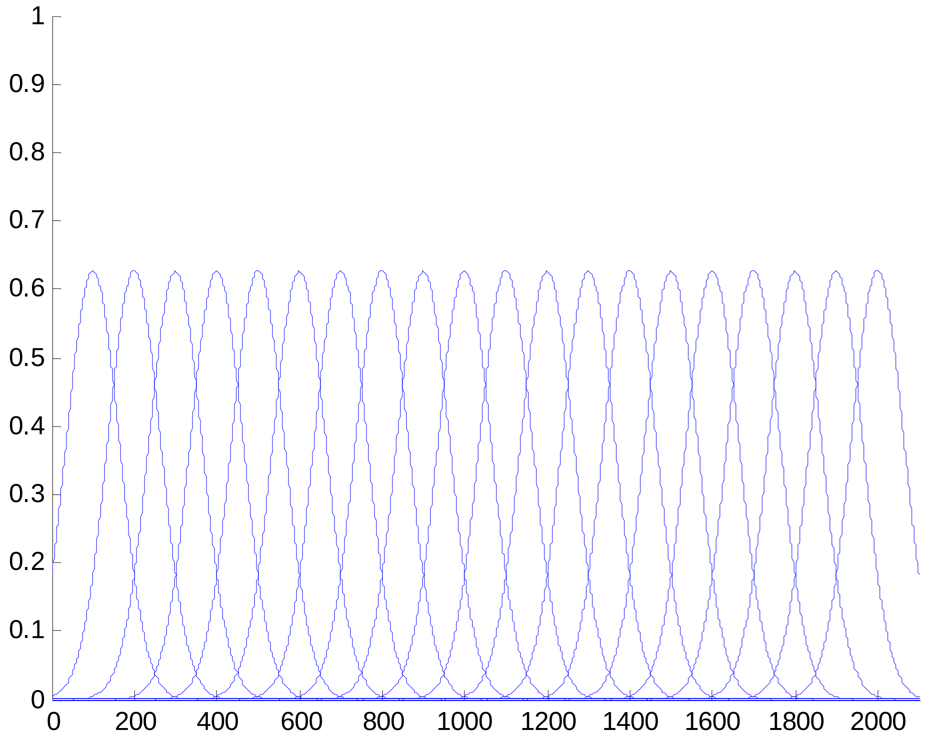
Combinazione lineare di funzioni di base

Da calcolare

Funzione di base (fissate)

Approssimazione mediante un modello semi-parametrico (lineare)

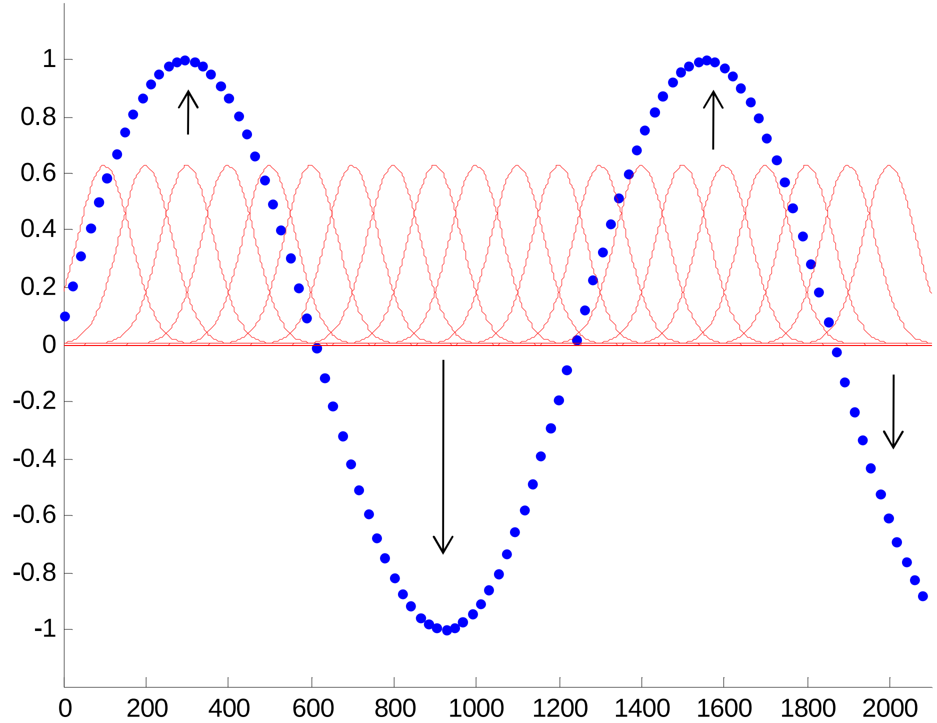


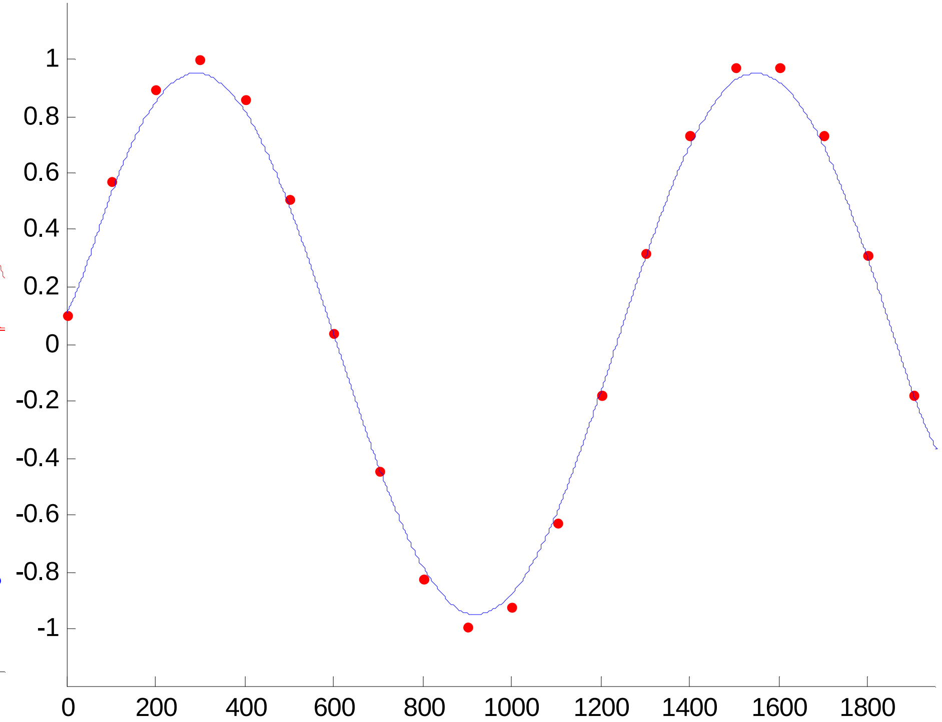


Sinusoide y = **A** sin(ωx + ϕ) con $\omega = 1/200$, $\phi = 0.1$.

Vogliamo fittare i punti con l'insieme di Gaussiane riportate sulla dx. In questo caso hanno tutte $\sigma = 90$. Come le utilizzo?

Funzionamento di un modello semi-parametrico (lineare)





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

Devo definire, gli M $\{w_i\}$.

3 << M << N – numero punti.

I σ sono tutti uguali ed uguali a 90^o, le Gaussiane sono equispaziate.

Le Gaussiane sono note tutte a priori, devono essere definiti i pesi.

Modelli lineari e non lineari

Classificazione alternativa dei modelli. Vengono utilizzate classi molto diversi di algoritmi per stimare i parametri di questi due tipi di modelli.

$$z(p(x,y)) = f(x) = \sum_i w_i x$$

f(.) è funzione lineare nei $\{w_i\}$

$$z(p(x,y)) = \sum_i f_i\left(p;w\right)$$

f(.) è funzione non lineare

e.g. f(.) = $e^{w\,x}$

f(.) = x ln(w x)

....

Riassunto

- Soft Clustering
- Mappe di Kohonen
- I modelli