



# Clustering & feature mapping

Iuri Frosio  
frosio@dsi.unimi.it

146



## Info...

- La lezione di venerdì' 19 si terrà in aula GAMMA alle ore 15.30.

246



# Classificazione

Descrizione numerica dell'oggetto:

altezza, colore, forma, posizione, ...

SPAZIO DEI CAMPIONI /  
DELLE CARATTERISTICHE



Classificatore



Classificazione dell'oggetto:

(classe A, classe B, ...)

SPAZIO DELLE CLASSI

346



## A cosa serve la classificazione?

- Compressione dati (telecomunicazioni, immagini, ...);
- segmentazione (bio)immagini;
- riconoscimento automatico;
- controllo robot;
- pattern recognition;
- ricostruzione superfici;
- ...

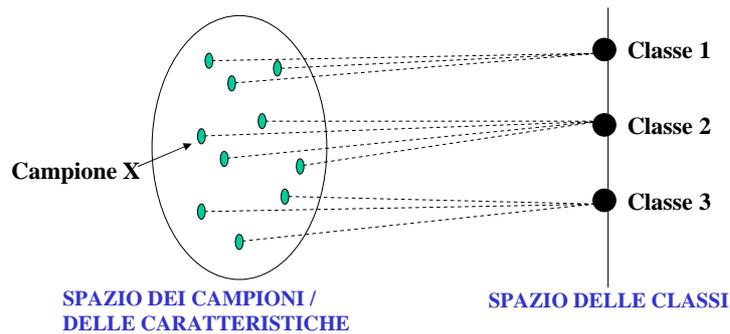
446



# Classificazione

Un'interpretazione geometrica:

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



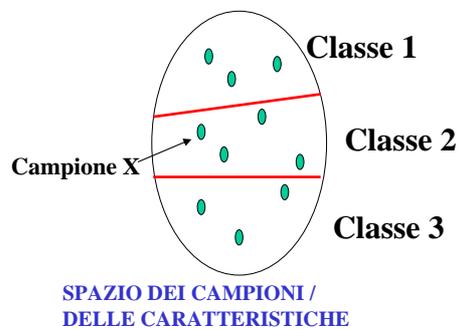
546



# Classificazione

Un'altra interpretazione geometrica:

*Partizione dello spazio dei campioni.*

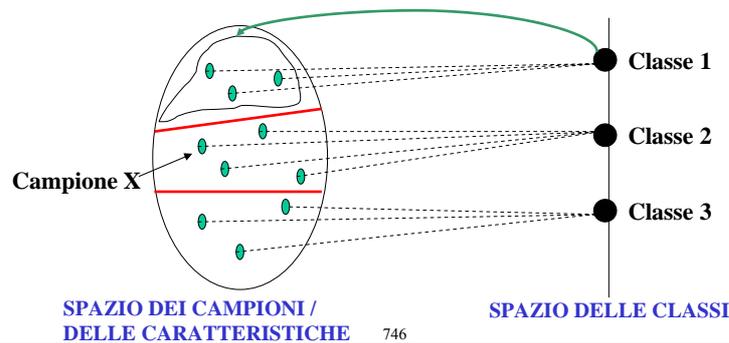


646



## Osservazione

- La classificazione è un'operazione *non reversibile*, per cui la trasformazione inversa sarà possibile solo con una predeterminata approssimazione.



## Componenti di un classificatore

- 1) **Classificatore** (“Schema” di classificazione);
- 2) **Algoritmo di addestramento** per il classificatore (per la definizione delle regioni associate a ciascuna classe e dei loro confini);
- 3) **Dati** per l’addestramento del classificatore;
- 4) [*Supervisore per l’addestramento del classificatore*].



## Scopo dell'addestramento

Scopo **dell'algoritmo di addestramento** è di far apprendere al **classificatore** la distribuzione statistica dei **dati** di addestramento in relazione alle classi, con o senza l'aiuto del **supervisore**. Una volta effettuato l'addestramento, il solo **classificatore** può essere utilizzato per la classificazione di nuovi dati, non presentati in fase di addestramento (*proprietà di generalizzazione*).

946



## Addestramento supervisionato / non supervisionato.

- **Addestramento supervisionato** (in presenza di supervisore):
  - i dati di addestramento sono già divisi in classi di appartenenza.
- **Addestramento non supervisionato** (non è necessaria la presenza del supervisore):
  - il sistema *classificatore* + *algoritmo di classificazione* effettua in maniera automatica una partizione dei dati di addestramento.

IL SUPERVISORE PUO' EFFETTUARE CONTROLLI A POSTERIORI PER VERIFICARE CHE LA PARTIZIONE OTTENUTA SIA CORRETTA.

1046



## In questa lezione...

- Clustering parametrico non supervisionato (K-means);
- Reti neurali per il feature mapping non supervisionato (SOM).

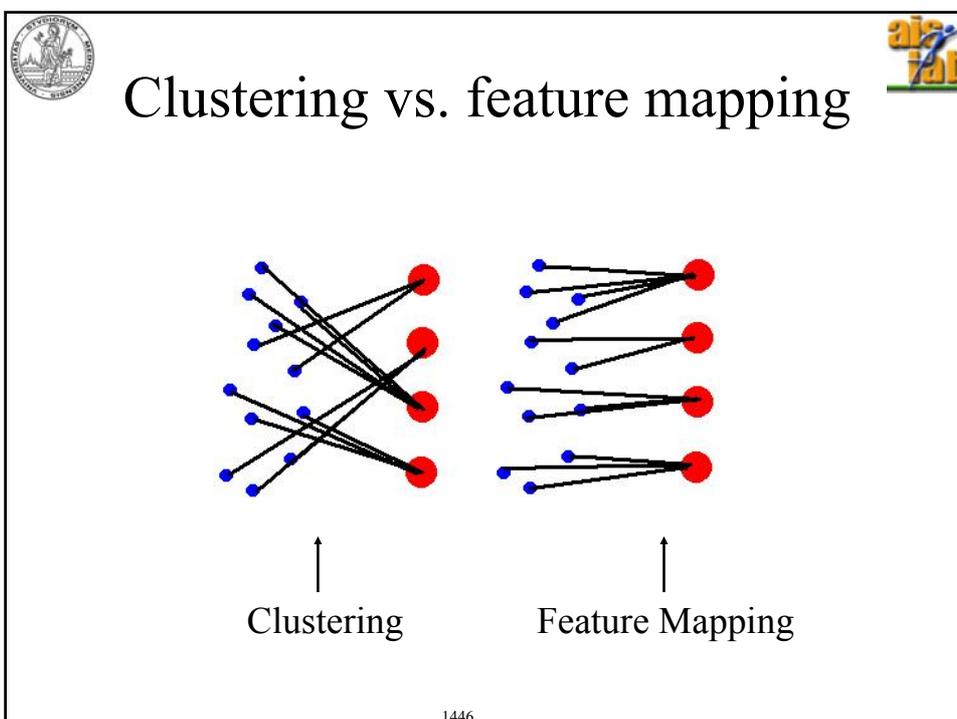
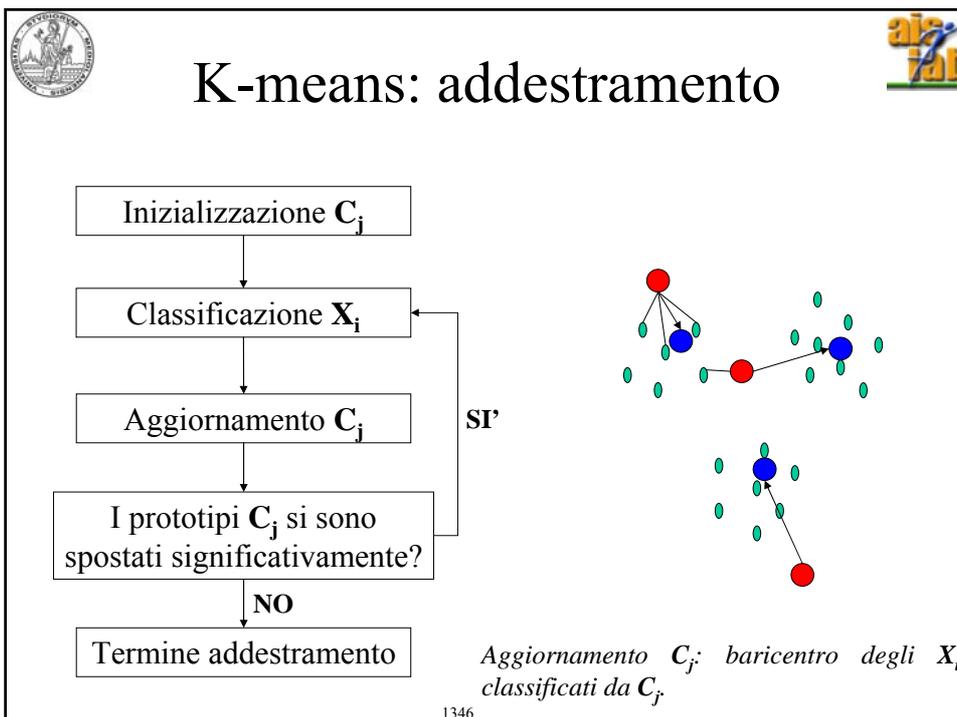
1146



## K-means: framework

- Siano  $\mathbf{X}_1, \dots, \mathbf{X}_D$  i dati di addestramento (per semplicità, definiti in  $\mathbb{R}^2$ );
- siano  $\mathbf{C}_1, \dots, \mathbf{C}_K$  i *prototipi* di  $K$  classi, definiti anch'essi in  $\mathbb{R}^2$ ; ogni *prototipo* identifica il baricentro della classe corrispondente;
- lo schema di classificazione adottato sia il seguente: “ $\mathbf{X}_i$  appartiene a  $\mathbf{C}_j$  se e solo se  $\mathbf{C}_j$  è il *prototipo* più vicino a  $\mathbf{X}_i$ ”;
- l'algoritmo di addestramento permette di determinare le posizioni dei *prototipi*  $\mathbf{C}_j$  mediante successive approssimazioni.

1246





## Feature mapping

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

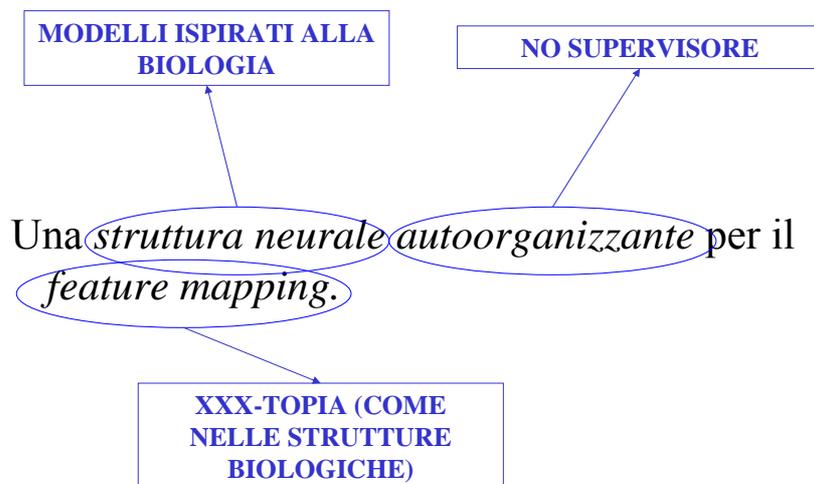
a uscite contigue corrispondono configurazioni d'ingresso contigue;

il classificatore opera una trasformazione tra lo spazio degli ingressi e lo spazio delle uscite (categorie) che preserva le relazioni di vicinanza tra i vari elementi.

1546



## Self Organizing Maps (SOM)



1646



## Tipo di apprendimento

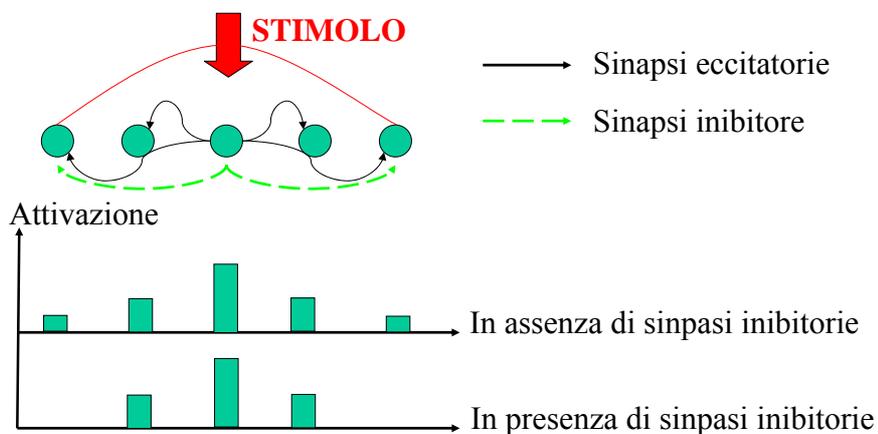
**COMPETITIVE LEARNING:** es. self organizing map - clustering e feature mapping - nell'addestramento non è necessario dare le "uscite desiderate" - "spara" un solo neurone per volta.

1746



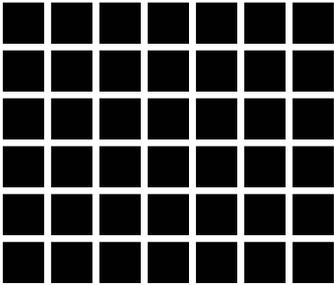
## Il campo recettivo

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



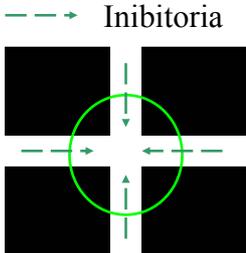
1846

 **Il campo recettivo: effetto Hermann** 



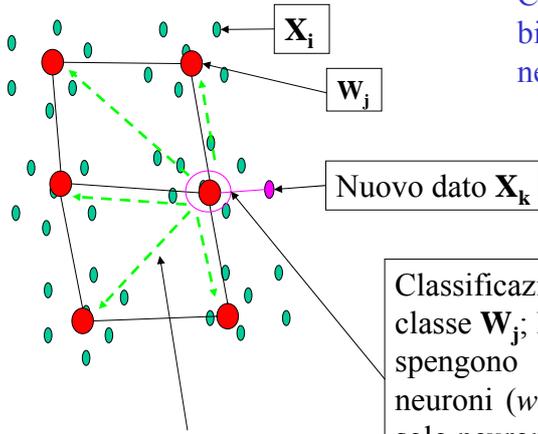
Pallini neri agli incroci delle linee bianche.

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



1946

 **SOM: comportamento atteso** 



Corrispondenza biunivoca tra neuroni e classi.

Nuovo dato  $X_k$

Connessione inibitoria

Classificazione di  $X_k$  nella classe  $W_j$ ; le sinapsi inibitorie spengono tutti gli altri neuroni (*winner takes all*). Il solo neurone che “spara”,  $W_j$ , individua la classe di  $X_k$ .

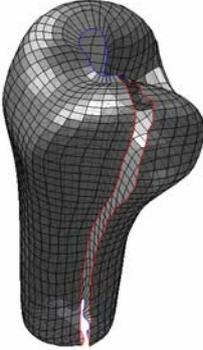
Mapping  $R^N \rightarrow R^M$ , con  $N \geq M$  (da spazio delle caratteristiche a spazio delle classi).

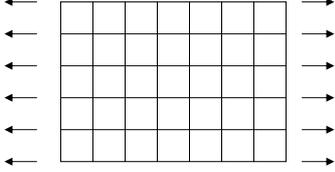
2046




## Femoral Bone Left

$\mathbb{R}^3 \rightarrow \mathbb{R}^2$





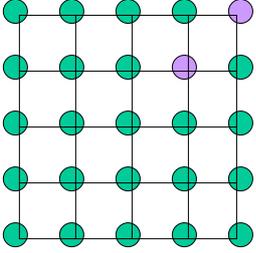
Topologia: cilindro

2146




## SOM: organizzazione topologica

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



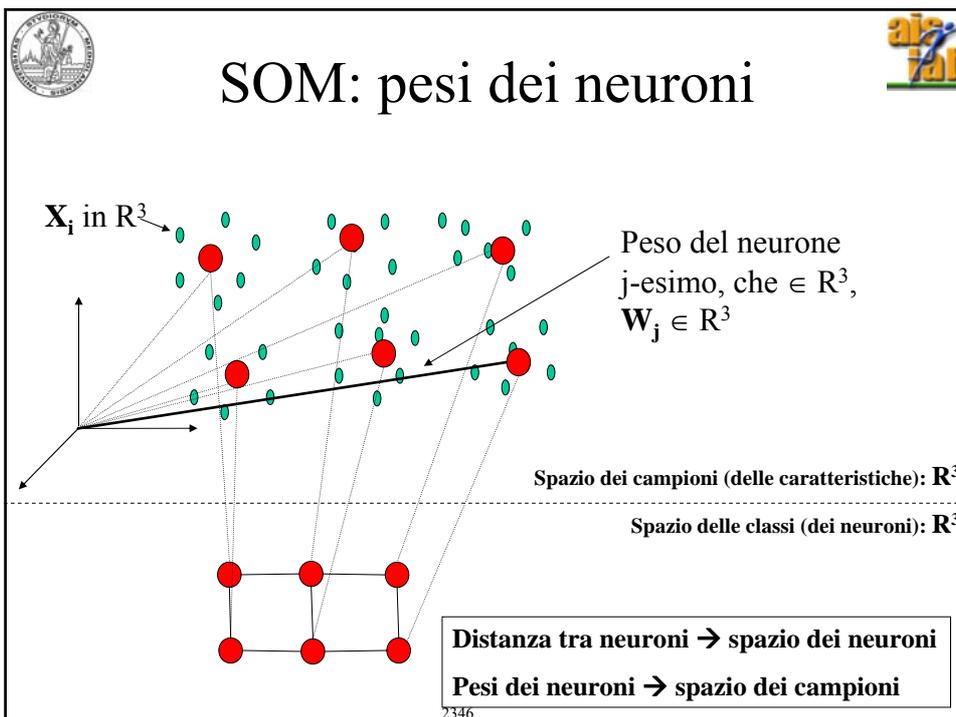
La distanza tra i due neuroni (nello spazio dei neuroni) è:

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

$|\Delta x| + |\Delta y| = 2$  [Manhattan distance,  $L^1$ ]

...

2246



- 
- SOM: addestramento**
- Siano  $\mathbf{X}_1, \dots, \mathbf{X}_D$  i dati di addestramento (per semplicità, definiti in  $\mathbb{R}^3$ );
  - siano  $\mathbf{W}_1, \dots, \mathbf{W}_K$  i *prototipi* di  $K$  classi, definiti anch'essi in  $\mathbb{R}^3$ ; la posizione di ogni *prototipo* identifica il peso di un neurone della SOM ( $\mathbf{W}_j$ );
  - lo schema di classificazione adottato sia il seguente: “ $\mathbf{X}_i$  appartiene a  $\mathbf{W}_j$  se e solo se  $\mathbf{W}_j$  è il *prototipo* (*peso del neurone*) più vicino a  $\mathbf{X}_i$ , nello spazio dei campioni (delle caratteristiche,  $\mathbb{R}^3$ )”;
  - l'algoritmo di addestramento permette di determinare i pesi dei neuroni (le posizioni dei *prototipi*)  $\mathbf{W}_j$  mediante successive approssimazioni;
  - L'algoritmo di addestramento tiene conto della topologia dei neuroni nello spazio dei neuroni (*feature mapping*).
- 2446



## SOM: addestramento

- All'interazione k- esima, si presenti alla rete il dato  $\mathbf{X}_i$ ;
- unità vincente, **winning unit**, (classificazione):  
 $j^*$  t.c.  $\|\mathbf{W}_{j^*} - \mathbf{X}_i\| = \min_j \|\mathbf{W}_j - \mathbf{X}_i\|$
- uscita:
  - $u_{j^*} = 1, u_j = 0, j \neq j^*$
- competitive Learning Rule (SOM, Kohonen '81):
  - $\Delta \mathbf{W}_j = \eta_k \Lambda_k(j, j^*) (\mathbf{X}_i - \mathbf{W}_j)$
  - $\Lambda_k(j, j^*) = \exp(-\|\mathbf{r}_j - \mathbf{r}_{j^*}\|^2 / 2\sigma_k^2)$

UNITA' VINCENTE  
(CLASSIFICAZIONE)

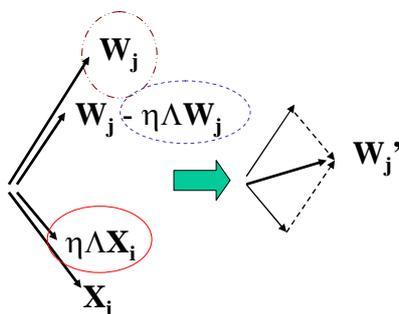
AGGIORNAMENTO  
PESI DEI NEURONI

FUNZIONE DI  
VICINATO



## Competitive Learning

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



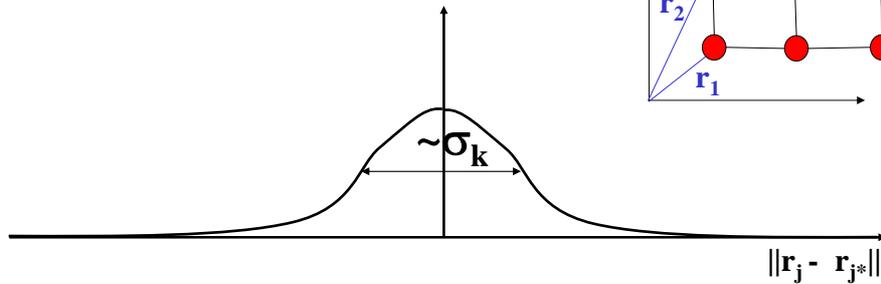
Il peso  $\mathbf{W}_j$  del neurone vincente  $j$  si sposta verso l'ingresso presentato  $\mathbf{X}_i$ .

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



## Funzione di vicinato

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



Neurone lontano dal neurone vincente:  $\Lambda_k(j, j^*) \rightarrow 0$ .

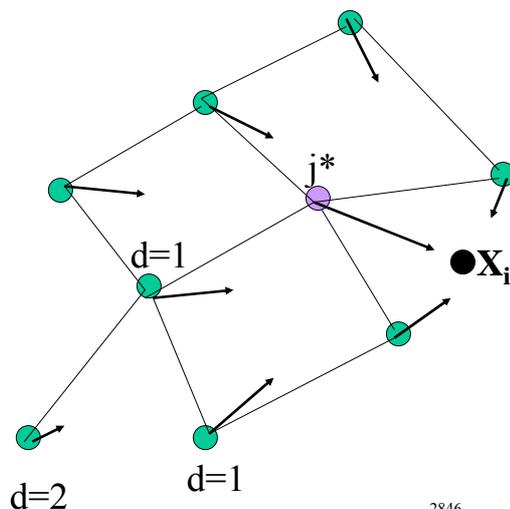
Aggiornamento del neurone:  $\mathbf{w}' = \mathbf{w} + \eta_k \Lambda_k(\mathbf{X}_i - \mathbf{w}_j) \rightarrow 0$ .

Tutti i pesi dei neuroni sono modificati alla presentazione di  $\mathbf{X}_i$ .

2746



## SOM: addestramento



In definitiva:

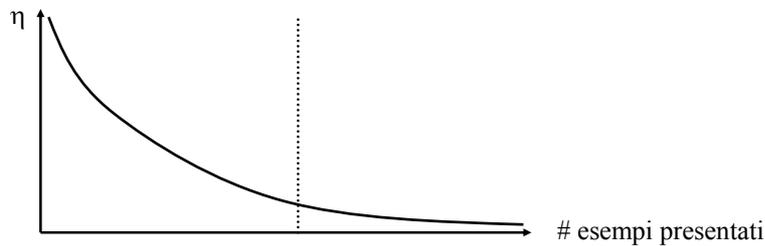
- Il neurone vincente si sposta verso  $\mathbf{X}_i$ , trascinando i vicini.

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

2846



## Learning rate



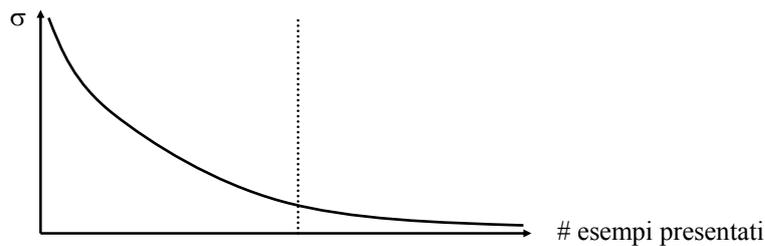
$$\Delta \mathbf{w}_j = \eta_k(t) \Lambda_k(j, j^*) (\mathbf{X}_i - \mathbf{w}_j) \rightarrow \mathbf{w}' = \mathbf{w} + \eta_k(t) \Lambda_k \mathbf{X}_i - \eta_k(t) \Lambda_k \mathbf{w}_j$$

Procedendo nell'addestramento della rete, i pesi dei neuroni "perdono" la possibilità di muoversi.

2946



## Neighborhood distance



$$\Delta \mathbf{w}_j = \eta_k \Lambda_k(j, j^*) (\mathbf{X}_i - \mathbf{w}_j) \rightarrow \mathbf{w}' = \mathbf{w} + \eta_k(t) \Lambda_k(t) \mathbf{X}_i - \eta_k(t) \Lambda_k(t) \mathbf{w}$$

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

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

3046



## SOM: addestramento



- L'addestramento avviene presentando alla rete i vettori (dati)  $\mathbf{X}_i \in \mathbb{R}^N$  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  $\eta$  e la neighborhood distance  $\sigma$  decrescono;
- Se presentiamo alla rete un nuovo esempio  $\mathbf{X}_i$  alla fine dell'addestramento, la rete lo classifica (neurone vincente);
- Categorie simili sono rappresentate da neuroni vicini (feature mapping).

3146



## SOM: addestramento



- 1) **ORDERING PHASE**:  $\eta$ ,  $\sigma$  **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  $\mathbb{R}^N$ ;
- 2) **TUNING PHASE**:  $\eta$ ,  $\sigma$  **piccoli**; ogni neurone si muove da solo; è una fase di raffinamento in cui vengono raggiunti con precisione i centri dei cluster.

3246



## SOM: addestramento



### Problemi:

- E' necessario scegliere  $\eta$ ,  $\sigma$ , 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";

3346



## SOM: addestramento



### Parametri caratteristici della SOM:

- # neuroni,  $\eta(t)$ ,  $\sigma(t)$ ;
- Durata ordering & tuning phase, epoche;
- Topologia della SOM (neuroni in  $R^M$ );
- Spazio dei dati  $\mathbf{q}$  ( $R^N$ ) e dei pesi  $\mathbf{w}$ ;

3446

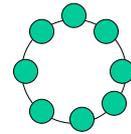


## SOM per ordinamento



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

Topologia della SOM : circolare



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

[Movie here](#)

3546

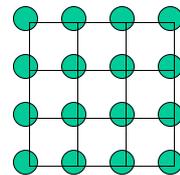


## SOM per ricostruzione 3D



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

Topologia della SOM : griglia 2D



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

[Movie here](#)

3646



## 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 +  $\eta$ ,  $\sigma$  modificati**
- Numero insufficiente di neuroni;  
**Sol.: Parametrizzazione della points cloud**

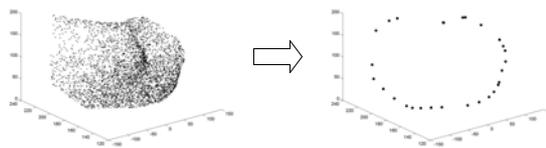
3746



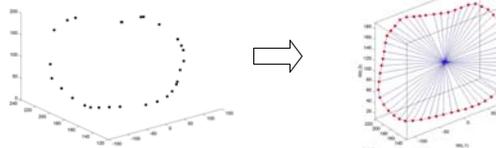
## Boundary First Method



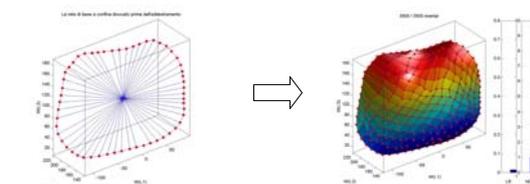
1) Individuazione dei punti di confine della superficie



2) Posizionamento dei neuroni di bordo



3) Addestramento della SOM (neuroni di bordo bloccati),  $\eta, \sigma$  maggiori ai lati



3840



## SOM per ricostruzione 3D



- [Movie here](#)
- [Movie here](#)

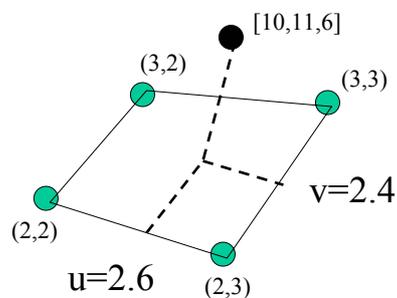
3946



## 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)



4046



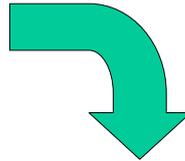
## Parametrizzazione Points Cloud



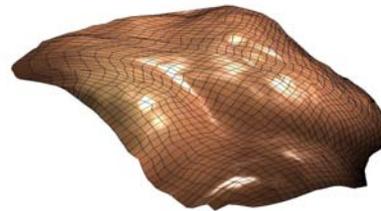
I punti 2D [u,v] possono essere triangolati (ricostruzione a mesh di triangoli). La mesh viene poi filtrata/interpolata.



Ricostruzione a mesh di triangoli



Filtraggio o interpolazione

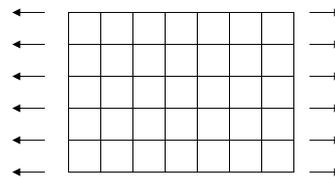
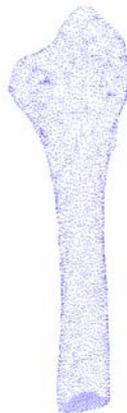


Ricostruzione Finale

4146



## Tibia Left



Topologia: cilindro

4246



## Varianti SOM



- SELF CREATING MAP
  - Aggiunta di un neurone:
    - Vicino al neurone vincente con  $f$  maggiore;
    - Vicino al neurone con curvatura massima;
- ADAPTIVE RESONANCE THEORY (reti ART);
- SOM SUPERVISIONATE;
- ...

4346



## Tesi: Virtual Art / ricostruzione 3D



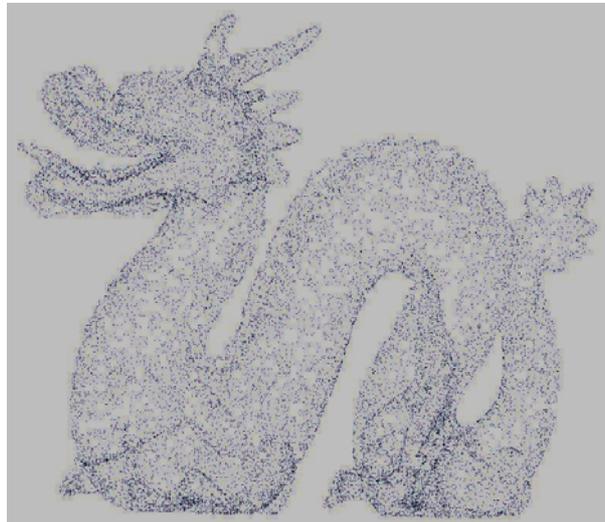
Ricostruzione tramite SOM di superfici nello spazio 3D dalla topologia complessa:

- Scelta della corretta topologia della SOM;
- Possibilità di usare più SOM (problemi di giunzione);
- Problemi nella generazione della points cloud;

4446



## Tesi: Virtual Art / ricostruzione 3D



4546



## Bibliografia



- J. Hertz, A. Krogh, R. G. Palmer,  
“Introduction to the theory of neural  
computation”, Santa Fe Institute Editorial  
Board, 1990  
– Feature mapping & SOM: cap. 9.4

4646