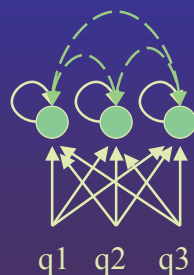


IL COMPETITIVE LEARNING E LE RETI NON SUPERVISIONATE...

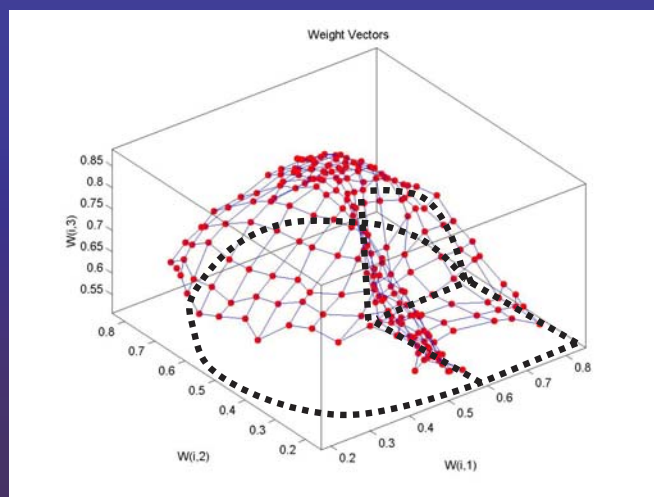


Frosio Iuri

Laboratorio MAVR

frosio@dsi.unimi.it

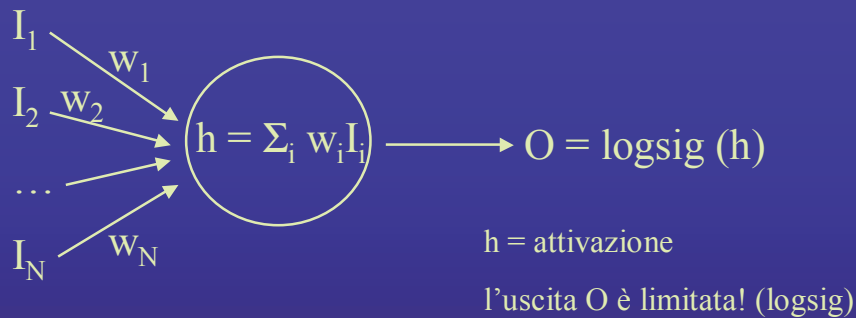
...APPLICAZIONI PER LA RICOSTRUZIONE 3D



Diverse tipologie di ANN

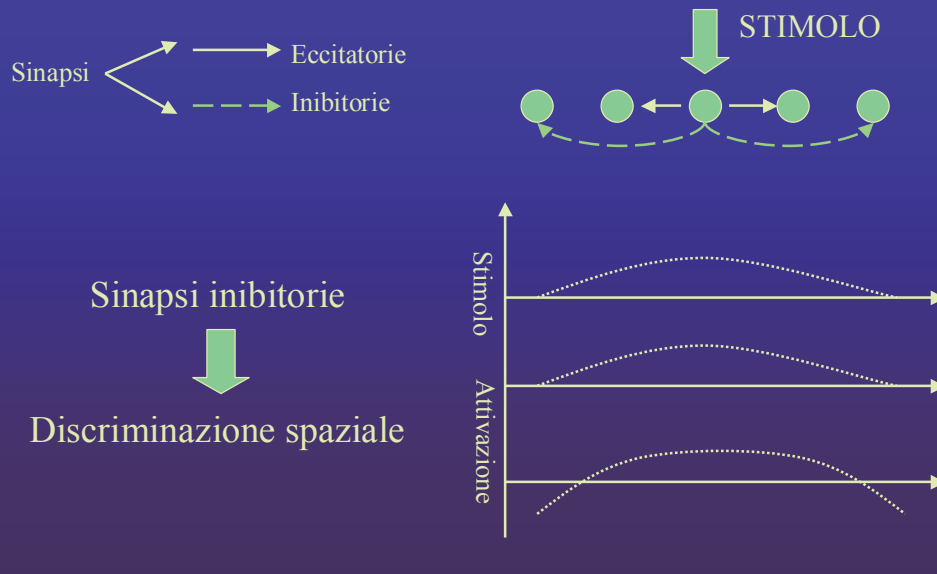


Il neurone artificiale

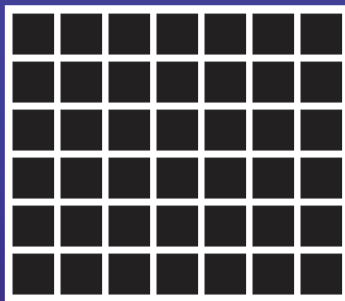


Modello semplificato del neurone reale
Strumento di calcolo o modello neurofisiologico?

Il campo recettivo



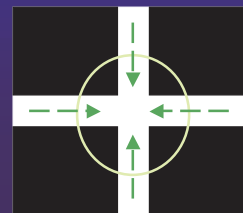
Il campo recettivo: effetto Hermann



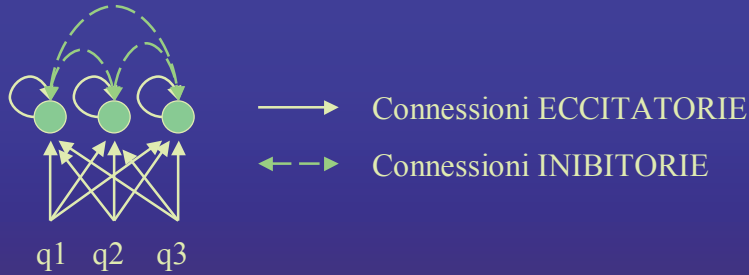
Vedo pallini neri agli incroci delle linee bianche !!!

Il neurone centrale viene “spento” dai neuroni vicini attivi: si genera il punto nero.

- - - ► Inibitoria

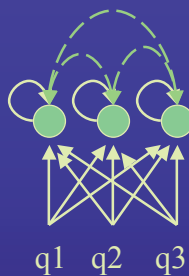


Reti non supervisionate: apprendimento competitivo



Ogni neurone ha connessioni inibitorie verso
gli altri neuroni

Competitive Learning 1: CL Rule



Attivazione : $h_i = \sum_j w_{ij} q_j = \mathbf{w}_i \mathbf{q}$

Unità vincente : i^* t.c. $\mathbf{w}_{i^*} \mathbf{q} > \mathbf{w}_j \mathbf{q} \quad \forall j$

Uscita : $u_{i^*} = 1, u_j = 0, j \neq i^*$

Competitive Learning Rule 1:

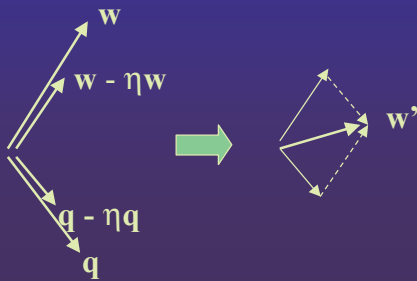
$$\Delta w_{i^*j} = \eta (q_j - w_{i^*j})$$

$$\Delta w_{ij} = \eta u_i (q_j - w_{ij})$$

Competitive Learning 1: CL Rule

$$\Delta w_{i^*j} = \eta (q_j - w_{i^*j}) \quad \text{WINNER TAKES ALL}$$
$$\Delta w_{ij} = \eta u_i (q_j - w_{ij}) \quad \text{HEBB'S LAW } [u_i q_j]$$

$$\Delta w_i = \eta u_i (\mathbf{q} - \mathbf{w}_i) \rightarrow \mathbf{w}_i' = \mathbf{w}_i + \eta u_i \mathbf{q} - \eta u_i \mathbf{w}_i$$



- w_{i^*} si sposta verso q ;
- $-\eta w$ evita che w cresca a dismisura.

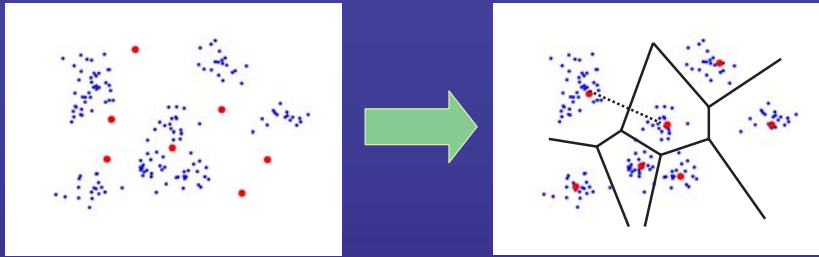
Competitive Learning 1: CL Rule

- Presentazione Q ingressi q (normalizzati);
- La rete sceglie il neurone vincente i^* ($w_{i^*} \sim q$);
- w_{i^*} viene aggiornato in direzione di q ;
- Il learning rate η decresce (convergenza);



CLUSTERING

Competitive Learning 1: Clustering



- Partizione del dataset in CLUSTER;
- Tessellazione di Voronoi di R^N ;
- Compressione dati;
- Divisione in classi.

Competitive Learning 2: Learning Vector Quantization (Kohonen)

Versione supervisionata della quantizzazione vettoriale:

$$\Delta w_{ij} = \eta u_i (q_j - w_{ij}) \quad \text{se la classe è corretta}$$
$$\Delta w_{ij} = -\eta u_i (q_j - w_{ij}) \quad \text{altrimenti}$$

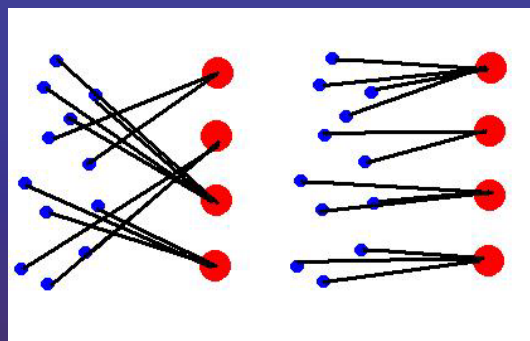
Competitive Learning 3: Feature Mapping

-Si dà importanza alla posizione dei neuroni (xxx-topia, Homunculus sensitivo);

-Uscite contigue \Leftrightarrow configurazioni d'ingresso contigue;

- Trasformazione (spazio degli ingressi) \rightarrow (spazio delle uscite (categorie)) t.c.sono preservate le relazioni di vicinanza tra i vari elementi.

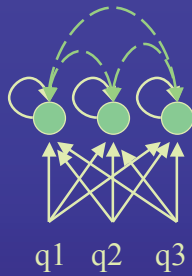
Competitive Learning 3: Feature Mapping



Clustering "normale"

Feature Mapping

Competitive Learning 3: S.O.M.



Attivazione : $h_i = \sum_j w_{ij} q_j = \mathbf{w}_i \mathbf{q}$

Unità vincente: i^* t.c. $|\mathbf{w}_{i^*} - \mathbf{q}| < |\mathbf{w}_j - \mathbf{q}|, j \neq i^*$

Uscita : $u_{i^*} = 1, u_j = 0, j \neq i^*$

Competitive Learning Rule 3 (S.O.M. - Kohonen '81):

$$\Delta \mathbf{w}_i = \eta_k \Lambda_k(i, i^*) (\mathbf{q} - \mathbf{w}_i)$$

$$\Lambda_k(i, i^*) = \exp(-|\mathbf{r}_i - \mathbf{r}_{i^*}|^2 / 2\sigma_k^2) \quad [\text{funzione di vicinato}]$$

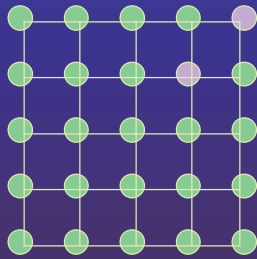
Competitive Learning 3: S.O.M.

$$\Delta \mathbf{w}_i = \eta_k \Lambda_k(i, i^*) (\mathbf{q} - \mathbf{w}_i) \rightarrow \mathbf{w}' = \mathbf{w} + \eta_k \Lambda_k \mathbf{q} - \eta_k \Lambda_k \mathbf{w}_i$$

- Spostamento di \mathbf{w} verso \mathbf{q} [$+\eta_k \Lambda_k \mathbf{q}$];
- Controllo di $\|\mathbf{w}\|$ [$-\eta_k \Lambda_k \mathbf{w}_i$].
- Aggiornamento su TUTTI i neuroni;
- $\Lambda_k(i, i^*)$ modula $\|\Delta \mathbf{w}_i\|$.

Competitive Learning 3: S.O.M.

- Neuroni topologicamente ordinati (es. griglia ordinata in \mathbb{R}^2).
- $\Lambda(i,j) \sim$ distanza “topologica”.



Distanza tra due neuroni:

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

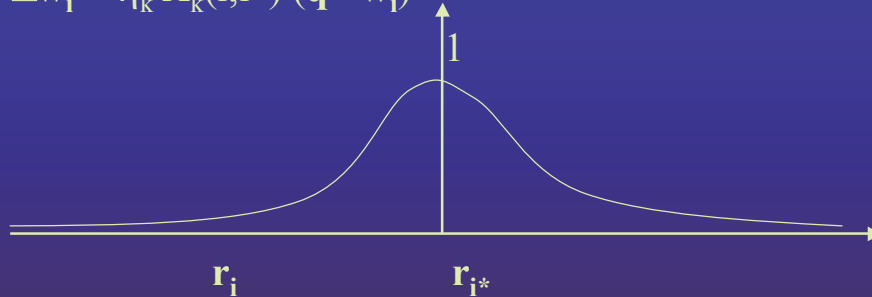
$$|\Delta x| + |\Delta y| = 2 \quad [\text{Manhattan}]$$

...

Competitive Learning 3: S.O.M.

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

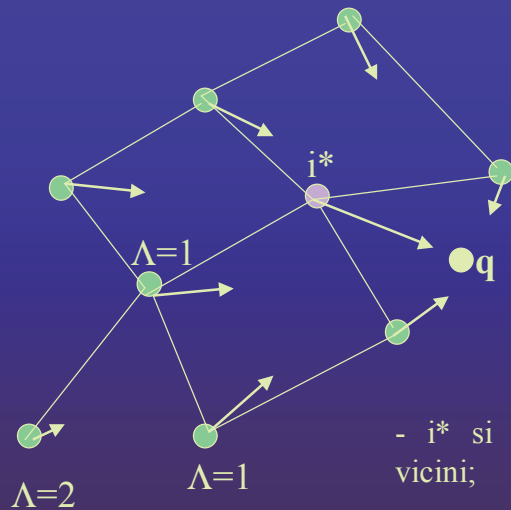
$$\Delta \mathbf{w}_i = \eta_k \Lambda_k(i, i^*) (\mathbf{q} - \mathbf{w}_i)$$



$$\|\mathbf{r}_i - \mathbf{r}_{i^*}\| \text{ grande} \Rightarrow \Lambda_k(i, i^*) \rightarrow 0 \Rightarrow \|\Delta \mathbf{w}_i\| \rightarrow 0$$

$$\|\mathbf{r}_i - \mathbf{r}_{i^*}\| \text{ piccolo} \Rightarrow \Lambda_k(i, i^*) \rightarrow 1 \Rightarrow \|\Delta \mathbf{w}_i\| \rightarrow \|\Delta \mathbf{w}_{i^*}\|$$

Competitive Learning 3: S.O.M.



- i^* si sposta verso q , trascinando i vicini;

- Ordinamento w in $R^N \cong$ Ordinamento neuroni in R^M .

Competitive Learning 3: S.O.M.

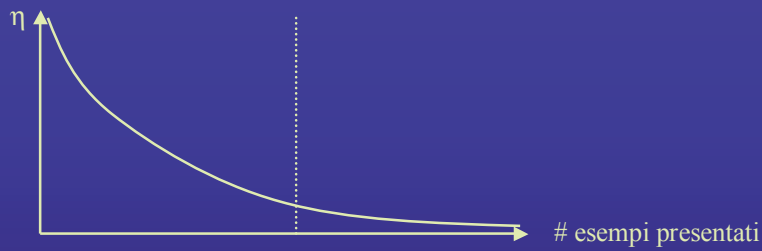
ADDESTRAMENTO

- Presentazione di Q dati $q \in R^N$, per un numero di epoche E ;
- Per ogni q : aggiornamento di tutti i pesi w ;
- Learning rate η , neighborhood distance σ decrescenti.



- Rete addestrata = classificatore
- Categorie simili \sim neuroni vicini.

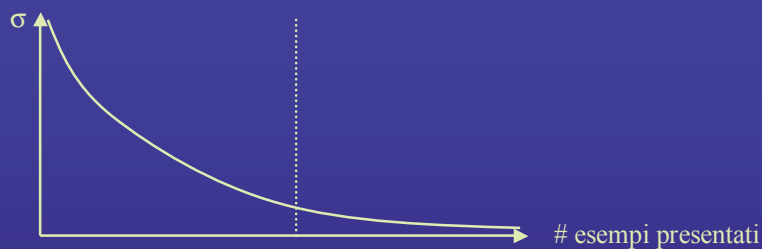
Competitive Learning 3: S.O.M.



$$\Delta \mathbf{w}_i = \eta_k \Lambda_k(i, i^*) (\mathbf{q} - \mathbf{w}_i) \rightarrow \mathbf{w}' = \mathbf{w} + \eta_k \Lambda_k \mathbf{q} - \eta_k \Lambda_k \mathbf{w}_i$$

I pesi dei neuroni perdono la possibilità di muoversi!

Competitive Learning 3: S.O.M.



$$\Delta \mathbf{w}_i = \eta_k \Lambda_k(i, i^*) (\mathbf{q} - \mathbf{w}_i) \rightarrow \mathbf{w}' = \mathbf{w} + \eta_k \Lambda_k \mathbf{q} - \eta_k \Lambda_k \mathbf{w}_i$$

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

I neuroni perdono la capacità di spostare i neuroni vicini.

Competitive Learning 3: S.O.M.

Fasi addestramento SOM

1) ORDERING PHASE: η , σ grandi:

$\|\Delta \mathbf{w}\|$ grandi;

Alta interazione topologica;

Dispiegamento in \mathbb{R}^N .

2) TUNING PHASE: η , σ piccoli:

$\|\Delta \mathbf{w}\|$ piccoli;

Interazione topologica bassa o nulla;

Raffinamento verso i centri dei cluster in \mathbb{R}^N .

Competitive Learning 3: S.O.M.

Problemi

- Scelta di $\eta(t)$, $\sigma(t)$, #epoche, durata ordering phase \rightarrow metodi empirici(!);
- Scelta topologia;
- Scelta #neuroni;
- Ridondanza dati di addestramento;
- Unità "morte";

Altri parametri caratteristici

- Spazio dei neuroni, \mathbb{R}^M ;
- Spazio dei dati \mathbf{q} (\mathbb{R}^N) e dei pesi \mathbf{w} ;

Applicazioni SOM

Riduzione della dimensione dello spazio dei dati:

$$\mathbf{q} \in \mathbb{R}^N \rightarrow \text{SOM} \rightarrow \mathbf{i}^* \in \mathbb{R}^M$$

Applicazioni:

- clustering (feature mapping);
- classificazione (segmentazione bioimmagini);
- compressione dei dati (telecomunicazioni);
- ricostruzione superfici nello spazio 3D;
- controllo robot, pattern recognition, ...

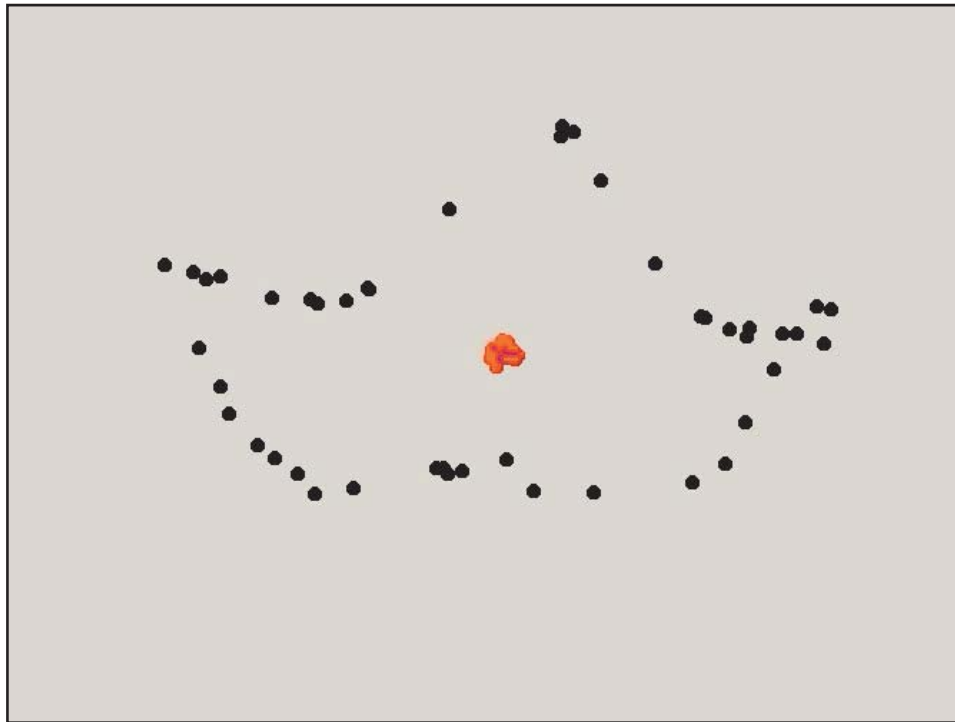
Esempi SOM 1: Ordinamento

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

Topologia della SOM : circolare



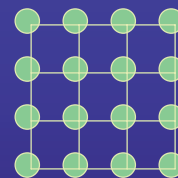
Parametri di addestramento : # neuroni, $\eta(t)$, $\sigma(t)$, ...



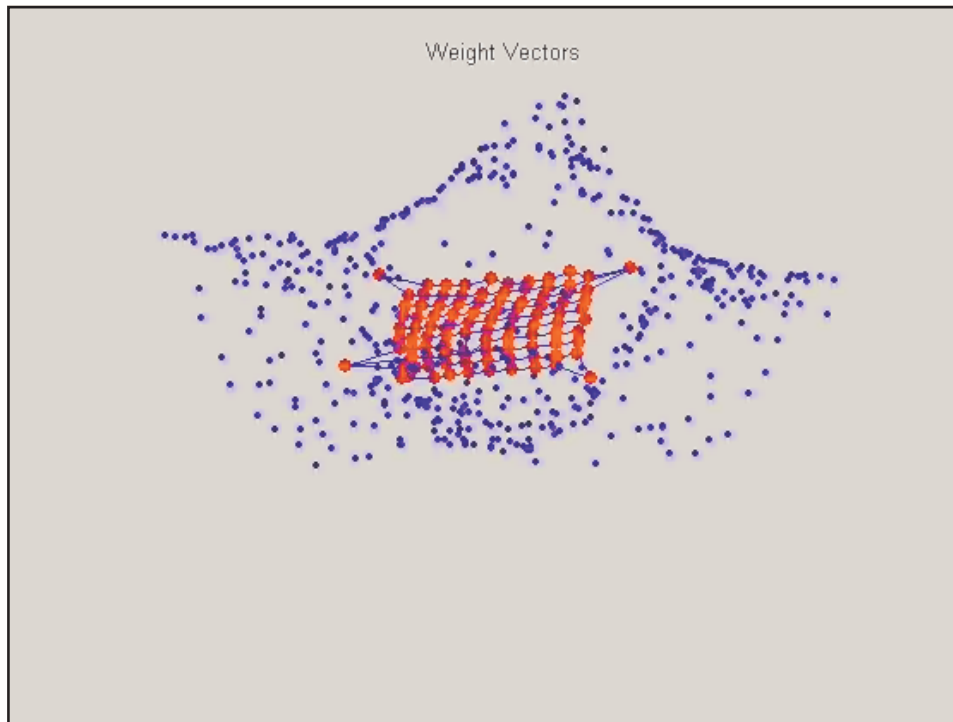
Esempio SOM 2: Ricostruzione

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, ...



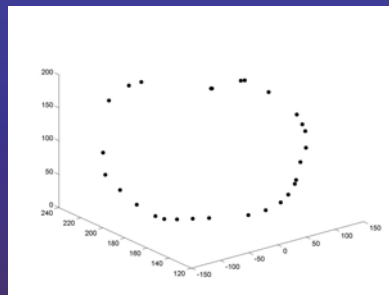
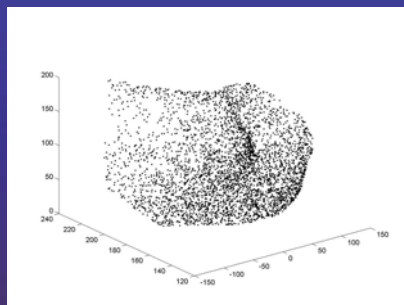
Applicazioni SOM: ricostruzione

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

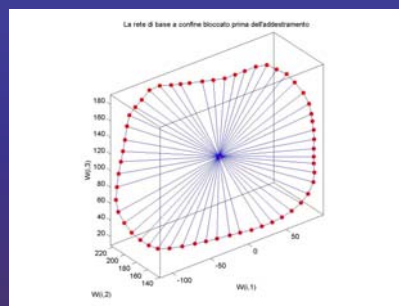
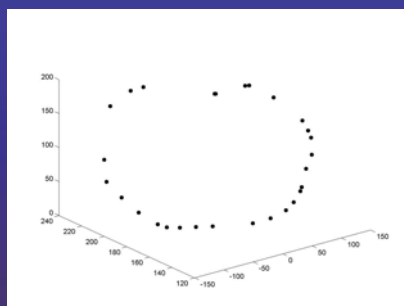
Boundary First Method

1) Individuazione dei punti di confine della superficie



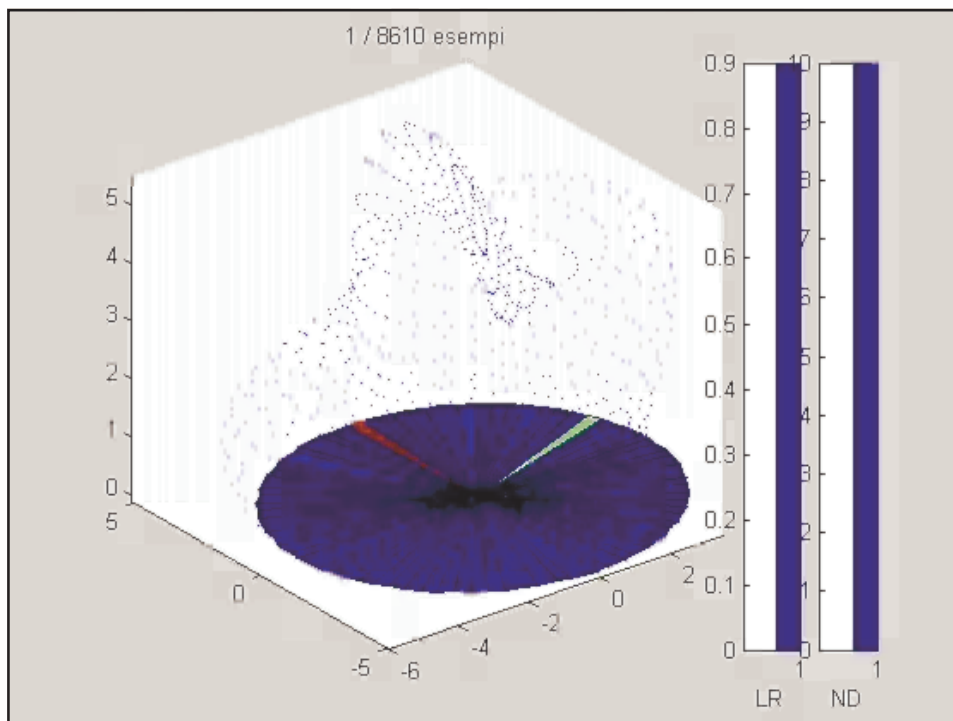
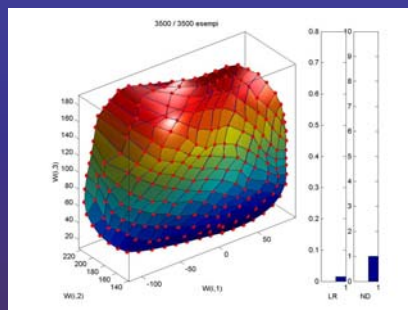
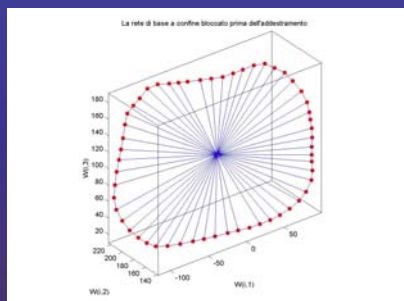
Boundary First Method

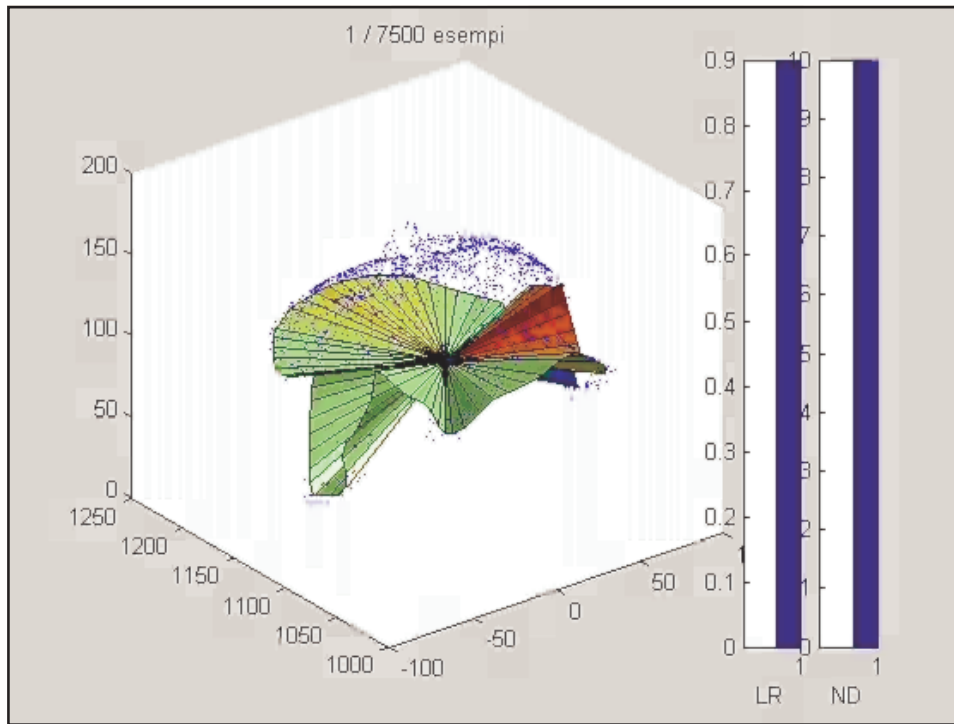
2) Posizionamento dei neuroni di bordo



Boundary First Method

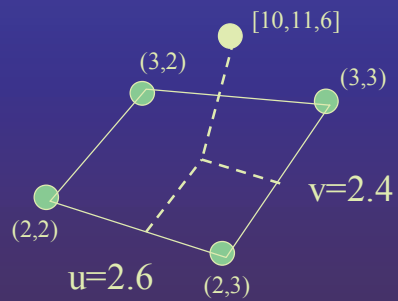
3) Addestramento della SOM (neuroni di bordo bloccati), η, σ maggiori ai lati





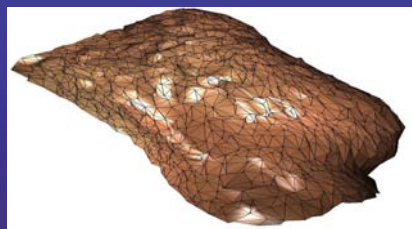
Parametrizzazione Points Cloud

- $q \in \mathbb{R}^3$, $q=[x,y,z] \rightarrow c \in \mathbb{R}^2$, $c=[u,v]$ (parametrizzazione)

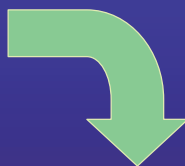


Parametrizzazione Points Cloud

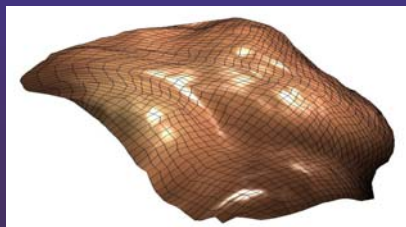
- 1) Triangolazione punti 2D
- 2) Filtraggio interpolazione mesh 3D



Ricostruzione a mesh di triangoli



Filtraggio o
interpolazione



Ricostruzione Finale

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;
- ...

Tesi: Virtual Art



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

Problemi

Scelta corretta topologia.

Approccio multi SOM
(problemi di giunzione).

Sistemi multicamera e
ricostruzione da video.

Generazione points cloud.