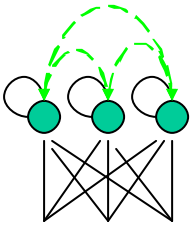
IL COMPETITIVE LEARNING E LE RETI NON SUPERVISIONATE…



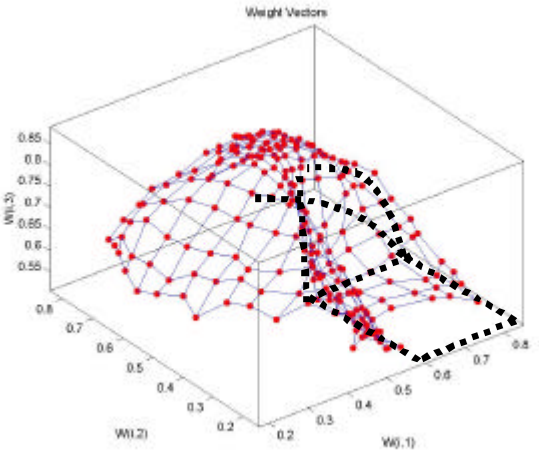
q1 q2 q3

Frosio Iuri

Laboratorio MAVR

frosio@dsi.unimi.it

…APPLICAZIONI PER LA RICOSTRUZIONE 3D



Diverse tipologie di ANN

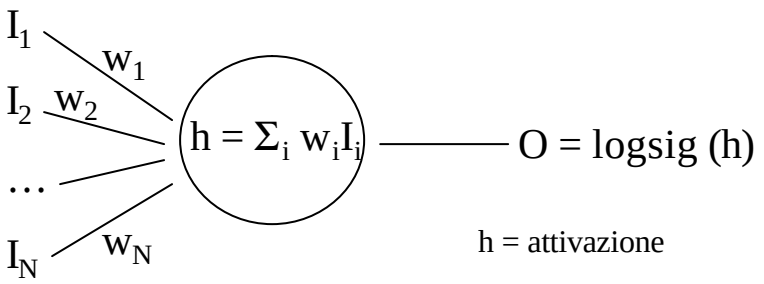
ANN

***SIGNAL-TRANSFER NETWORKS**: multilayer perceptron, … - simili a fdt - esistono varianti retroazionate e tempovarianti*

***STATE-TRANSFER NETWORKS** (memorie associative) : rete di Hopfield, … - sono in grado di "ricordare" gli esempi usati in addestramento (attrattori) - addestramento con Hebb*

COMPETITIVE LEARNING: self organizing map, … - effettuano il clustering dei dati (ma non solo!) - nell'addestramento non è necessario dare le "uscite desiderate"

Il neurone artificiale



h = attivazione

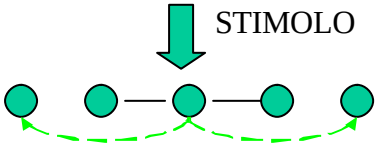
l'uscita O è limitata! (logsig)

Modello semplificato del neurone reale

Strumento di calcolo o modello neurofisiologico?

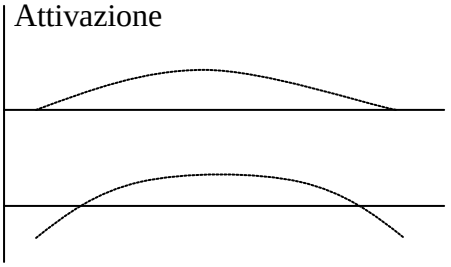
Il campo recettivo

- Ogni neurone è dotato di **sinapsi eccitatorie** (verso i neuroni vicini) ed **inibitorie** (verso i neuroni lontani);



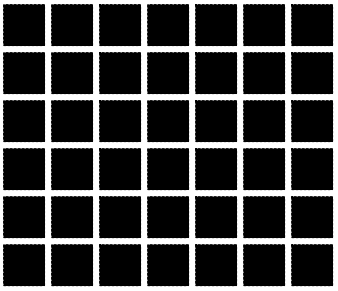
Eccitatorie

Inibitorie



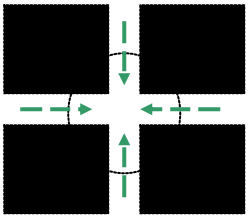
LA PRESENZA DI SINAPSI INBITORIE PERMETTE DI DISCRIMARE MEGLIO LA ZONA DI PROVENIENZA DELLO STIMOLO

Il campo recettivo: effetto Hermann



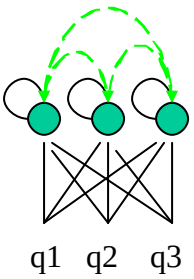
Vedo pallini neri agli incroci delle linee bianche !!!

Inibitoria



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

Reti non supervisionate: apprendimento competitivo

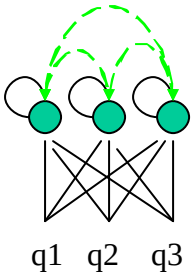


Connessioni ECCITATORIE

Connessioni INIBITORIE

Ogni neurone ha connessioni inibitorie verso gli altri neuroni

Competitive Learning 1: CL Rule



Attivazione : $h_i = \Sigma_i w_{ij} q_j = \mathbf{w_i q}$

Unità vincente : i* t.c. $\mathbf{w_{i*} q} > \mathbf{w_j q}$ $\forall j$

cioè: $h_{i*} > h_j$

Uscita : $u_{i*} = 1$, $u_j = 0$, j!=i*

Competitive Learning Rule 1:

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

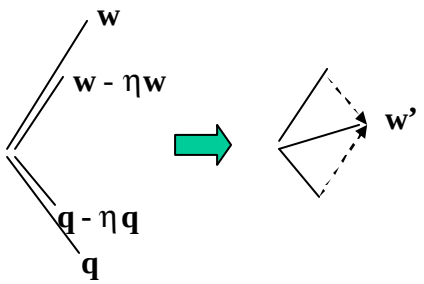
$\Delta w_{ij} = \eta u_i (q_j - w_{ij})$

Competitive Learning 1: CL Rule

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

$\Delta w_{ij} = \eta\, u_i\,(q_j - w_{ij})$ **HEBB'S LAW [$u_i q_j$]**

$D\mathbf{w_i} = \eta\, u_i\,(\mathbf{q} - \mathbf{w_i}) \;\; ® \;\; \mathbf{w_i}' = \mathbf{w_i} + \eta u_i \mathbf{q} - \eta u_i \mathbf{w_i}$



- Il peso **w** del neurone vincente i* si sposta verso l'ingresso presentato **q**;

- -η**w** evita che il peso **w** cresca a dismisura.

Competitive Learning 1: CL Rule

- Presentiamo alla rete gli ingressi **q** (normalizzati);

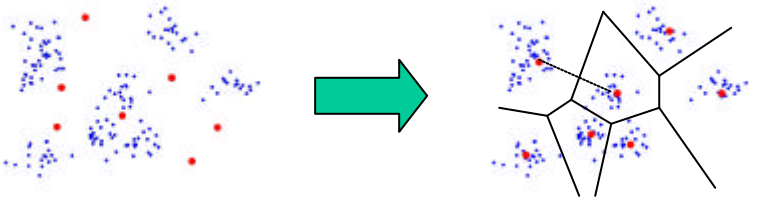
- Per ogni ingresso la rete sceglie il neurone vincente i* (il neurone con peso **w** "più simile" a **q**);

- Il peso $\mathbf{w_{i^*}}$ del neurone vincente i* viene aggiornato in direzione di **q**;

- Durante l'addestramento il learning rate η diventa più piccolo;

- Effetto: ogni neurone si sposta verso il centro di un cluster dei dati presentati (**CLUSTERING / QUANTIZZAZIONE VETTORIALE**).

Competitive Learning 1: Clustering



- Il dataset viene diviso in CLUSTER;
- Tessellazione di Voronoi dello spazio;
- Compressione dei dati;
- Divisione in classi.

Competitive Learning 2: Learning Vector Quantization (Kohonen)

E' una versione supervisionata della quantizzazione vettoriale:

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

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

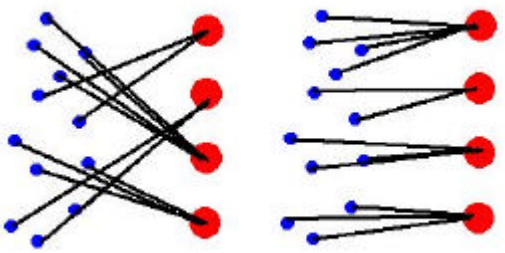
Competitive Learning 3: Feature Mapping

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

- A uscite contigue corrispondono configurazioni d'ingresso contigue;

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

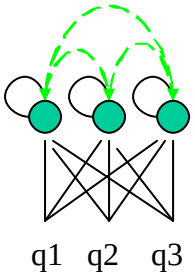
Competitive Learning 3: Feature Mapping



Clustering "normale"

Feature Mapping

Competitive Learning 3: S.O.M.



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

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

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

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

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

Competitive Learning 3: S.O.M.

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

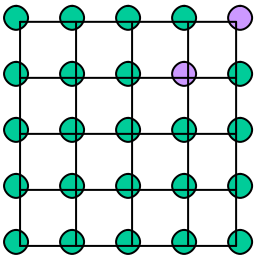
- Il neurone si sposta verso l'ingresso presentato [$+ \eta_k\Lambda_k\mathbf{q}$];

- [$-\eta_k\Lambda_k\mathbf{w}_i$] evita che il peso $\mathbf{w}_i$ cresca a dismisura.

L'AGGIORNAMENTO VIENE FATTO SU TUTTI I NEURONI, NON SOLO SUL NEURONE VINCENTE i*. L'AGGIORNAMENTO E' MODULATO DALLA FUNZIONE DI VICINATO $\Lambda_k(i,i^*)$.

Competitive Learning 3: S.O.M.

I neuroni della SOM sono ordinati topologicamente nello spazio dei neuroni (es. griglia ordinata in R^2). In tale spazio viene calcolata la funzione di vicinato $\Lambda(i,j)$.



La distanza tra i due neuroni è:

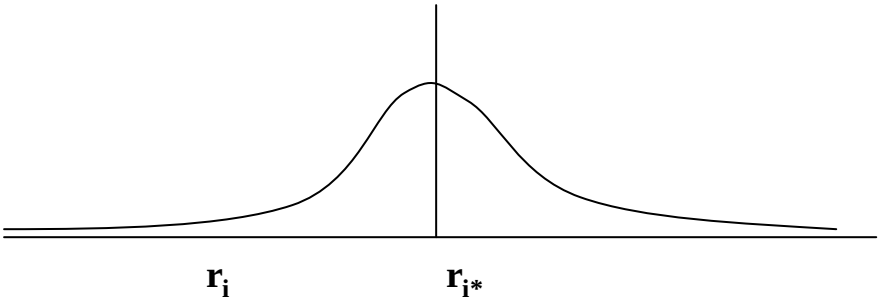
$\text{sqrt}(\Delta x^2 + \Delta y^2) = 1.4142$ [Metrica Euclidea]

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

...

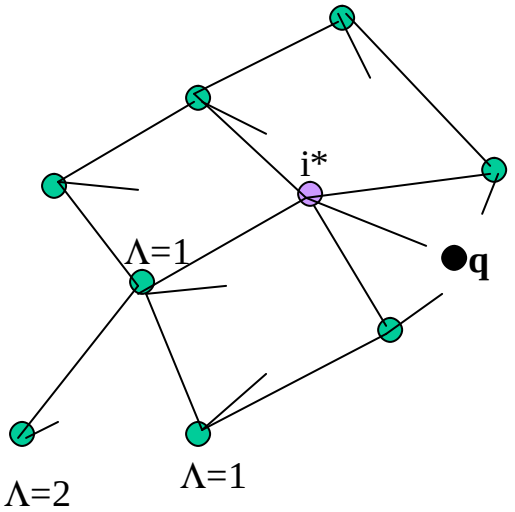
Competitive Learning 3: S.O.M.

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



Per un neurone lontano dal neurone vincente $\Lambda_k(i,i^*) \rightarrow 0$; l'aggiornamento del neurone [$\mathbf{w'} = \mathbf{w} + \eta_k \Lambda_k(\mathbf{q} - \mathbf{w_i})$] è piccolo!

Competitive Learning 3: S.O.M.



In definitiva:

-Il neurone vincente si sposta verso **q**, trascinando i vicini

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

Competitive Learning 3: S.O.M.

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

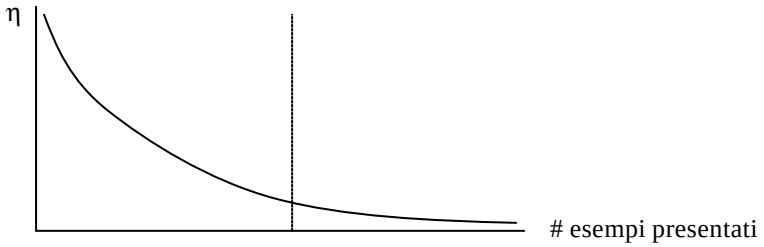
- Per ogni esempio presentato q vengono aggiornati i pesi dei neuroni della rete;

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

- Se presentiamo alla rete un nuovo esempio q ala fine dell'addestramento, la rete lo classifica (neurone vincente);

- Categorie simili sono rappresentate da neuroni vicini.

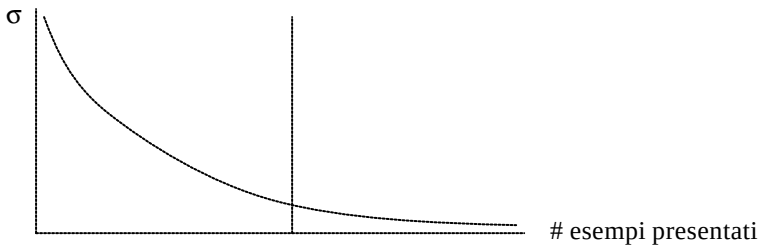
Competitive Learning 3: S.O.M.



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

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

Competitive Learning 3: S.O.M.



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

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

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

Competitive Learning 3: S.O.M.

Addestramento SOM

1) ORDERING PHASE: h, s **grandi**; ogni neurone può spostarsi molto verso l'ingresso q; il neurone trascina con sé i vicini; in tale fase la rete si dispiega nello spazio R^N;

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

Competitive Learning 3: S.O.M.

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";

Competitive Learning 3: S.O.M.

Parametri caratteristici della SOM:

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

Applicazioni SOM

Riduzione della dimensione dello spazio dei dati:

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

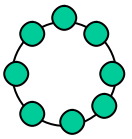
Applicazioni:

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

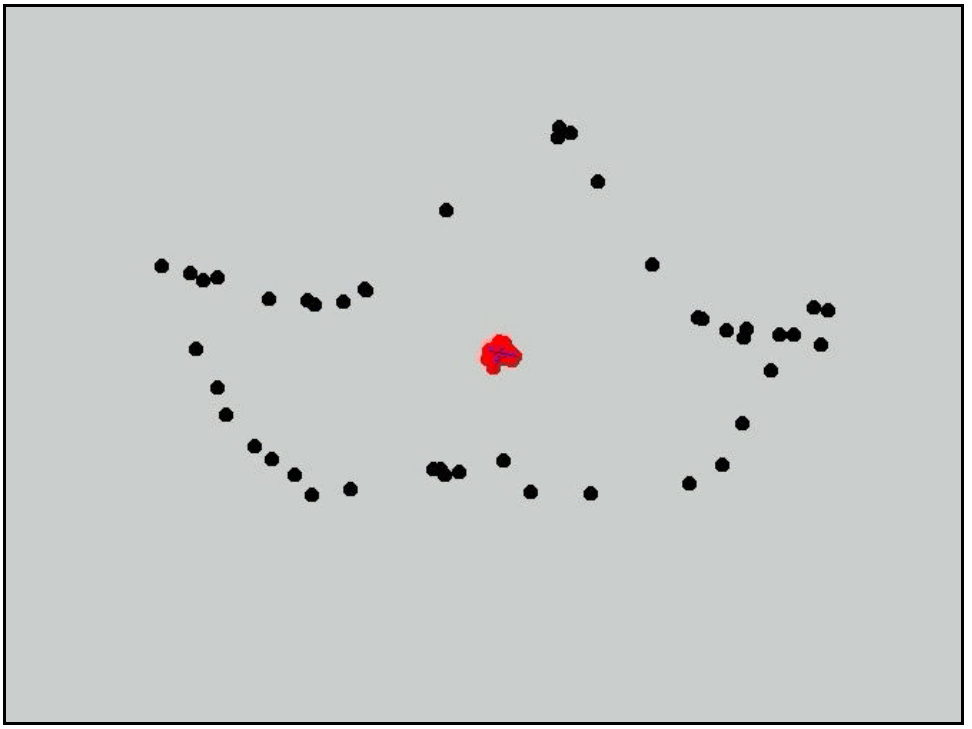
Esempi SOM 1: Ordinamento

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

Topologia della SOM : circolare



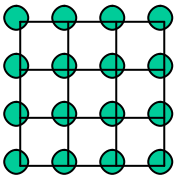
Parametri di addestramento : # neuroni, $\eta(t)$, $\sigma(t)$, ...



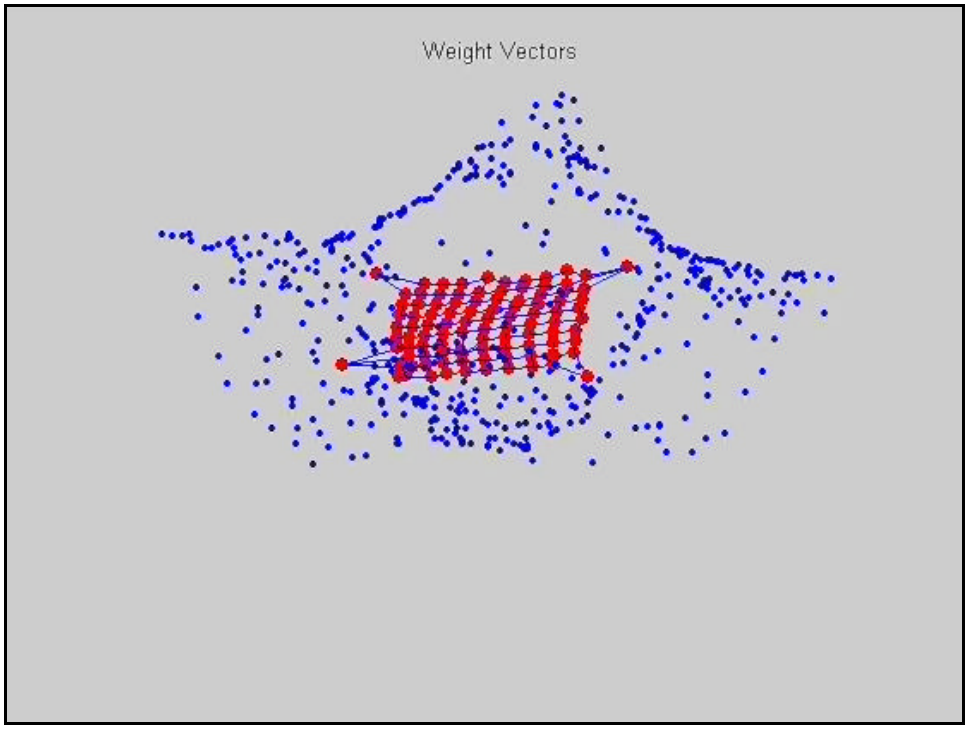
Esempio SOM 2: Ricostruzione

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

Topologia della SOM : griglia 2D



Parametri di addestramento : # neuroni = 10x10, 0.5 → 0.1 lin, 10 → 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;

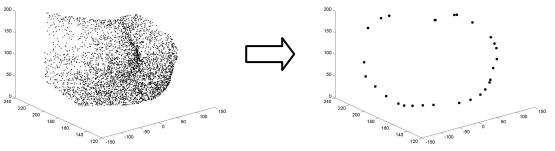
  Sol.: Boundary First Method + η, σ modificati

- Numero insufficiente di neuroni;

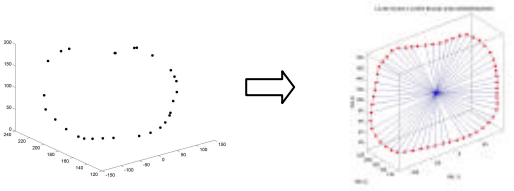
  Sol.: Parametrizzazione della points cloud

Boundary First Method

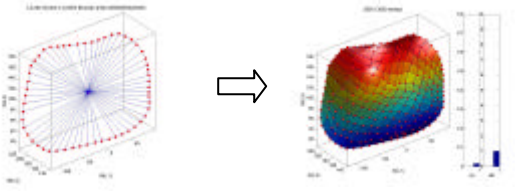
1) Individuazione dei punti di confine della superficie

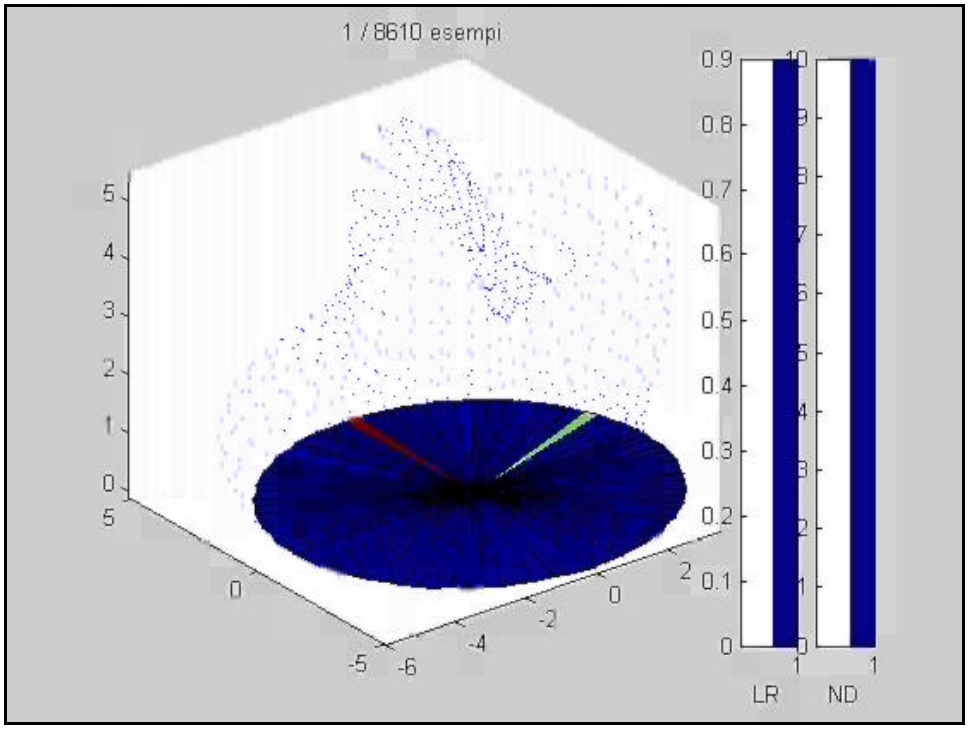


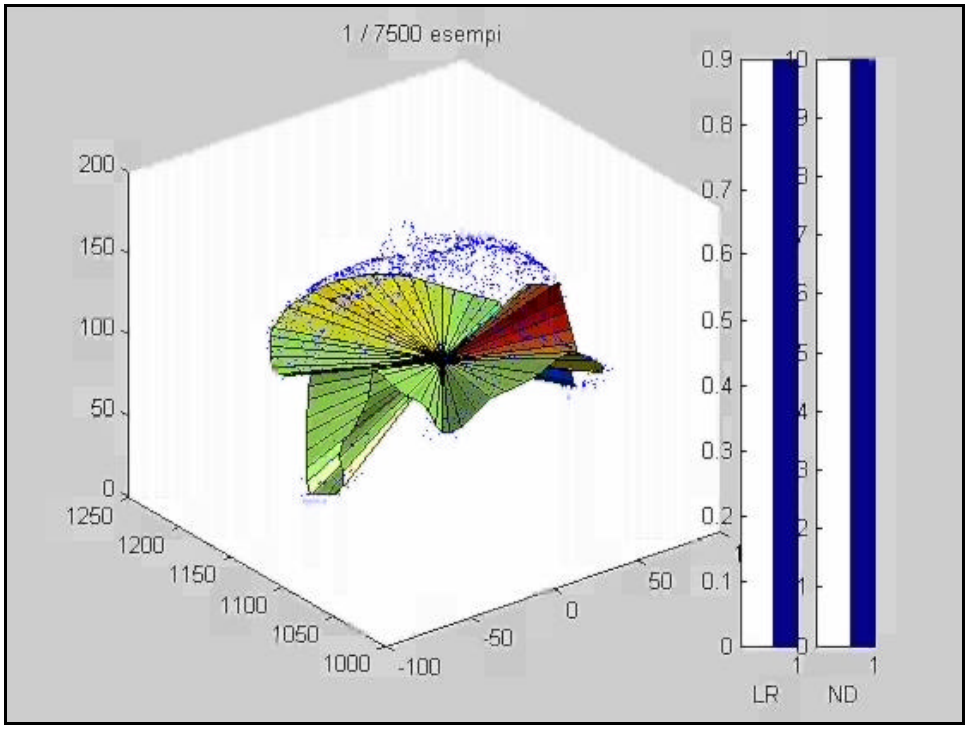
2) Posizionamento dei neuroni di bordo



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

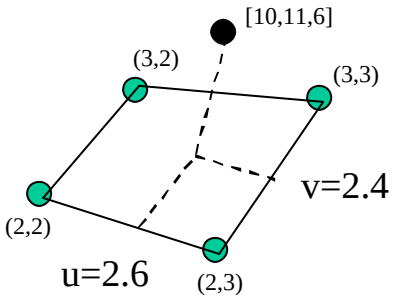



1 / 8610 esempi
LR
ND


1 / 7500 esempi
LR
ND

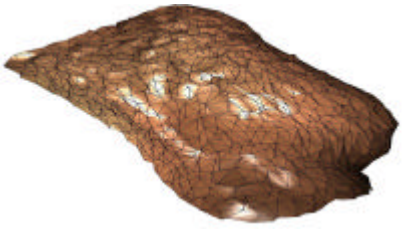
Parametrizzazione Points Cloud

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



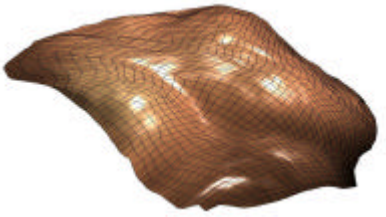
Parametrizzazione Points Cloud

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



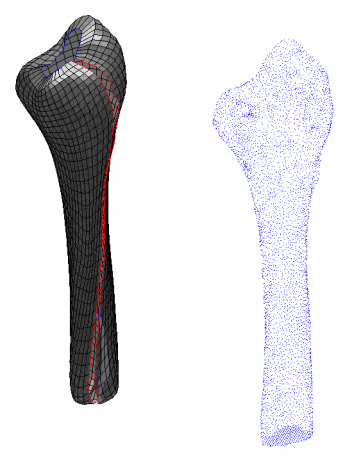
Filtraggio o interpolazione

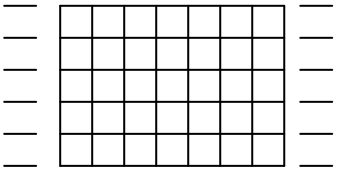
Ricostruzione a mesh di triangoli



Ricostruzione Finale

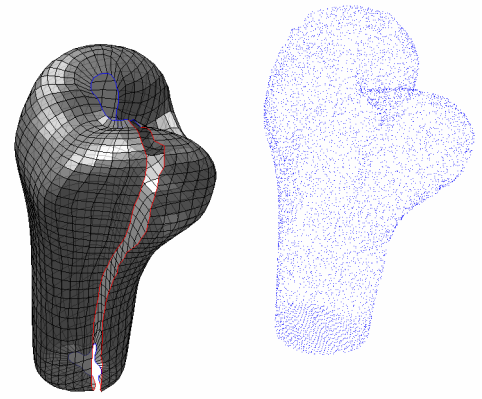
Tibia Left

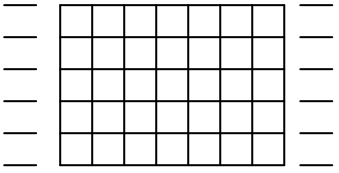




Topologia: cilindro

Femoral Bone Left





Topologia: cilindro

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:

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

Progetto Ricostruzione 3D…