

# RETI NEURALI

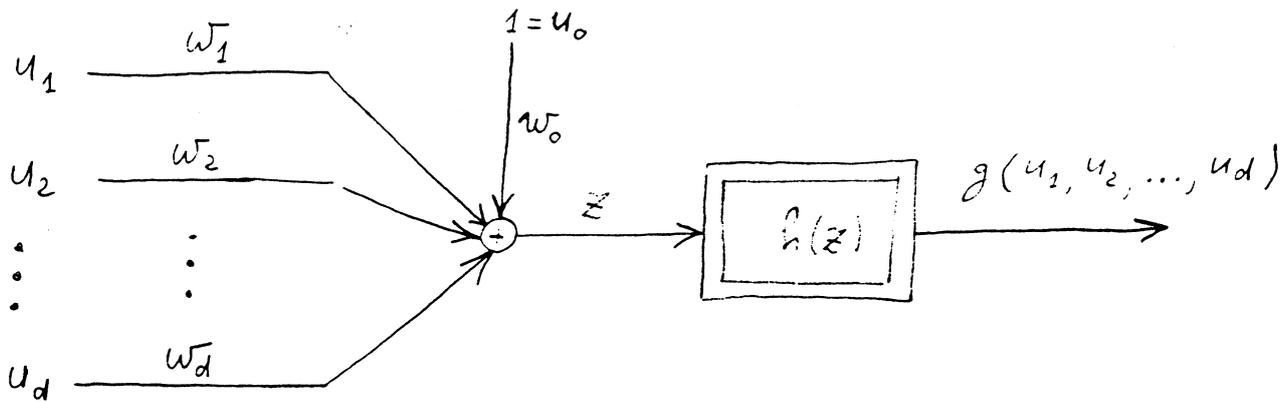
## *Contenuti:*

- Reti di perceptroni
- Reti neurali a base radiale
- Esempio: Stima del rendimento volumetrico  $\eta$
- Conclusioni

*Motivazione:* Modelli flessibili e modulari (= costituiti dalla interconnessione di blocchi elementari) per rappresentare relazioni del tipo  $y = g(u_1, u_2, \dots, u_d)$ . (identificazione a scatola nera di modelli nonlineari).

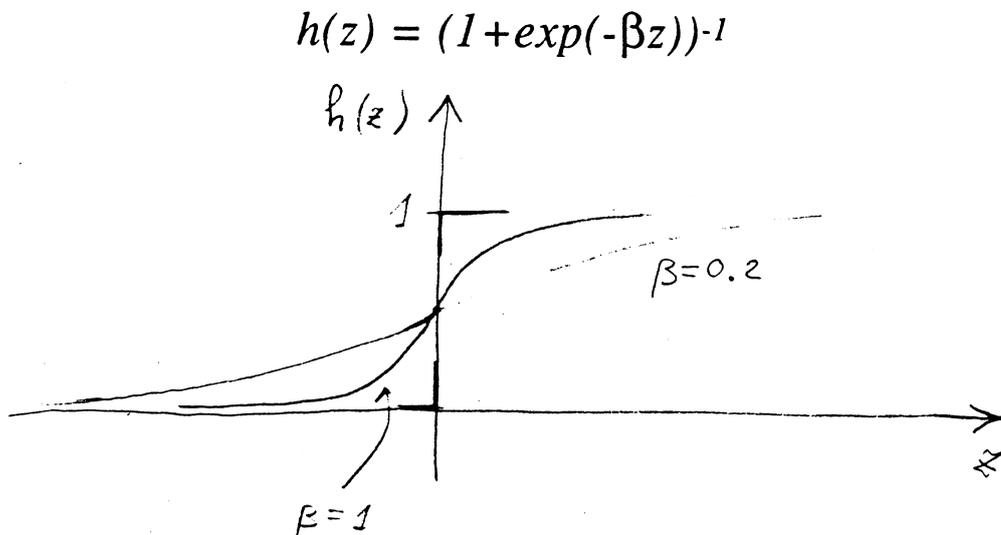
# RETI DI PERCETTRONI

Il perceptrone (Rosenblatt 1958):

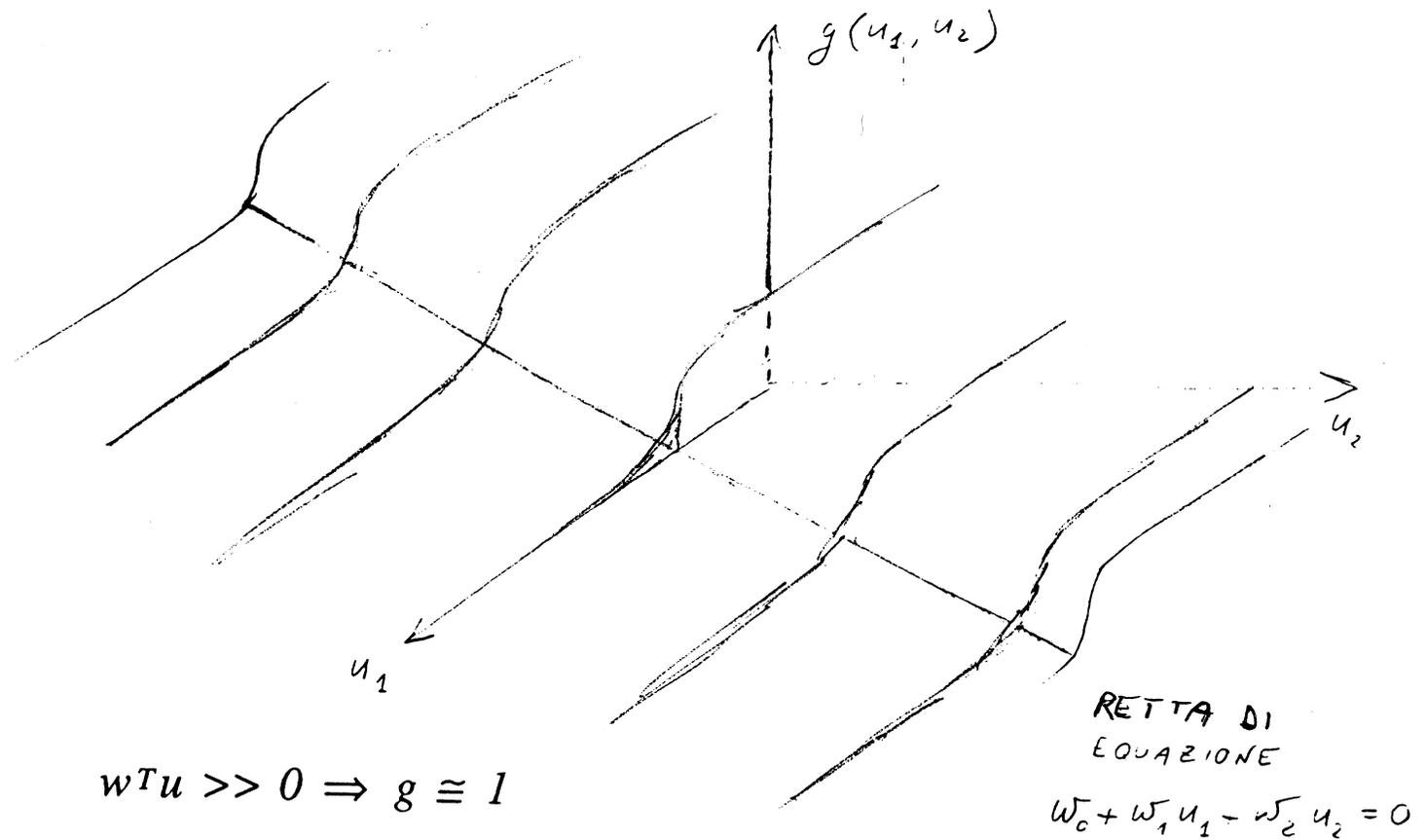


$$g(u_1, u_2, \dots, u_d) = h \left( \sum_{i=0}^d w_i u_i \right) = h(w^T u)$$

- $w = [w_0 \ w_1 \ w_2 \ \dots \ w_d]^T$  : vettore dei pesi (è l'equivalente di  $\theta$ )
- $h(z)$  è tipicamente una nonlinearietà di tipo sigmoide.  
Per esempio:



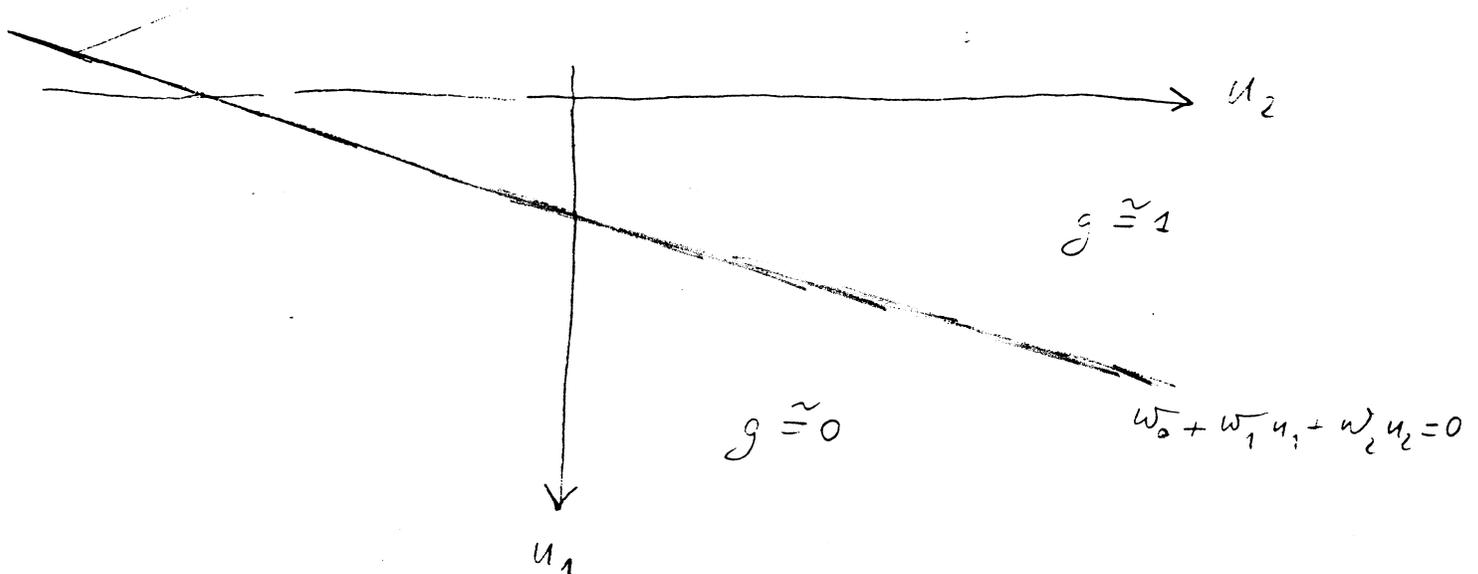
## Interpretazione grafica del perceptrone



$$w^T u \gg 0 \Rightarrow g \cong 1$$

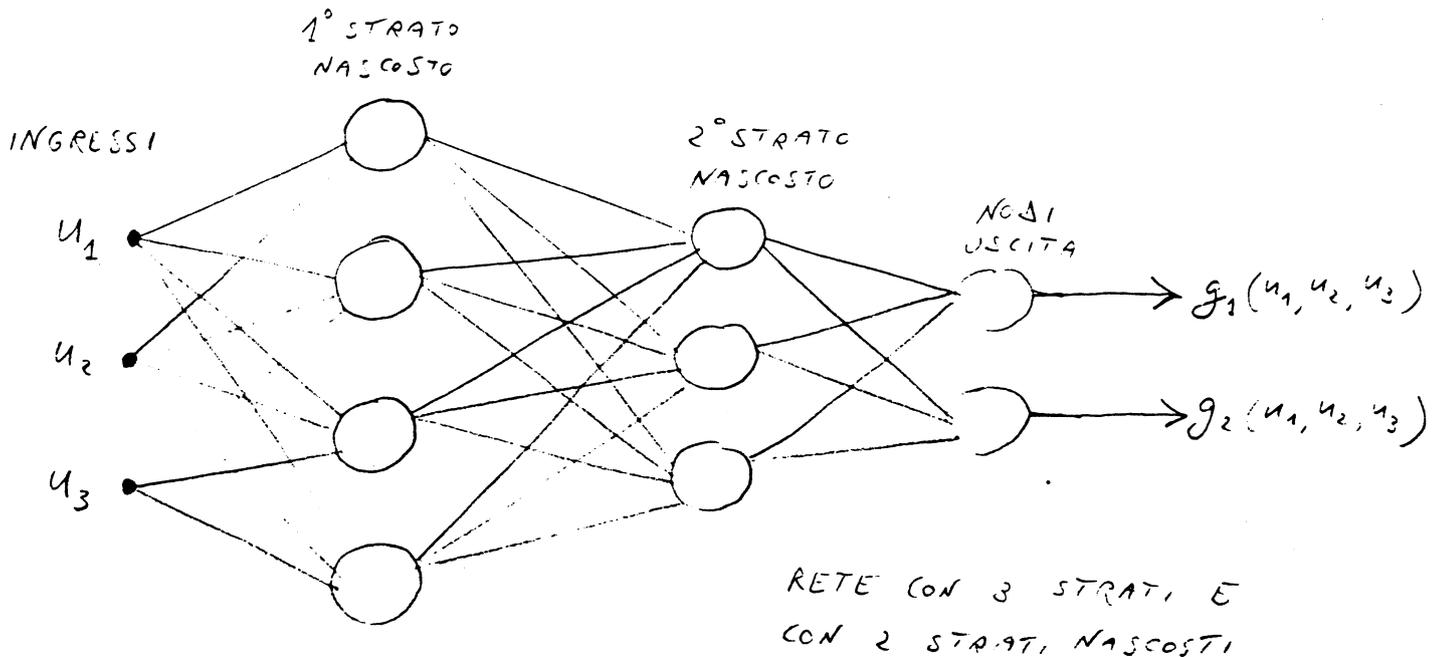
$$w^T u \ll 0 \Rightarrow g \cong 0$$

Il singolo perceptrone divide lo spazio delle variabili indipendenti in due regioni (classificatore).



Idea: combinare in cascata più perceptron

*Rete neurale di perceptron a più strati  
(MLP NN: Multi-Layer Perceptron Neural Network)*



*Teorema (Cybenko 1989):* Una rete con 2 strati (= con 1 strato nascosto) è un *approssimatore universale*, può cioè approssimare in modo arbitrariamente preciso ogni funzione continua  $g(u_1, u_2, \dots, u_d)$  (pur di prendere un numero sufficiente di neuroni).

*Nota:* il teorema vale anche se nei nodi di uscita la nonlinearietà sigmoide è sostituita da una relazione lineare.

*Come stimare i parametri (= pesi  $w_i$ )?*

$g(u, \theta)$  è funzione nonlineare di  $\theta$



problema di stima nonlineare

*Criterio:* Minimizzazione di SSR

*Algoritmi:* Quelli già visti, tra cui:

- Metodo del gradiente (chiamato "backpropagation")
- Gauss-Newton
- etc.

*Difficoltà numeriche:* Problema di ottimizzazione non lineare che può avere vari minimi locali. E' bene ripetere gli algoritmi iterativi partendo da inizializzazioni di volta in volta diverse. Richiede molto tempo.

## *Glossario*

*Teoria dell'identificazione*

*Reti neurali*

Modello  $\leftrightarrow$  Rete neurale

Parametri  $\leftrightarrow$  Pesi

Stimare i parametri  $\leftrightarrow$  Addestrare la rete

Dati per l'identificazione  $\leftrightarrow$  Esempi, training set

Sovraparametrizzazione  $\leftrightarrow$  Overtraining

*Problema fondamentale:* Scelta del n° di neuroni e della struttura della rete.

*Osservazione:* Per valutare la bontà di un modello, tra tutti i criteri visti, il più applicabile è la crossvalidazione dividendo i dati in due gruppi.

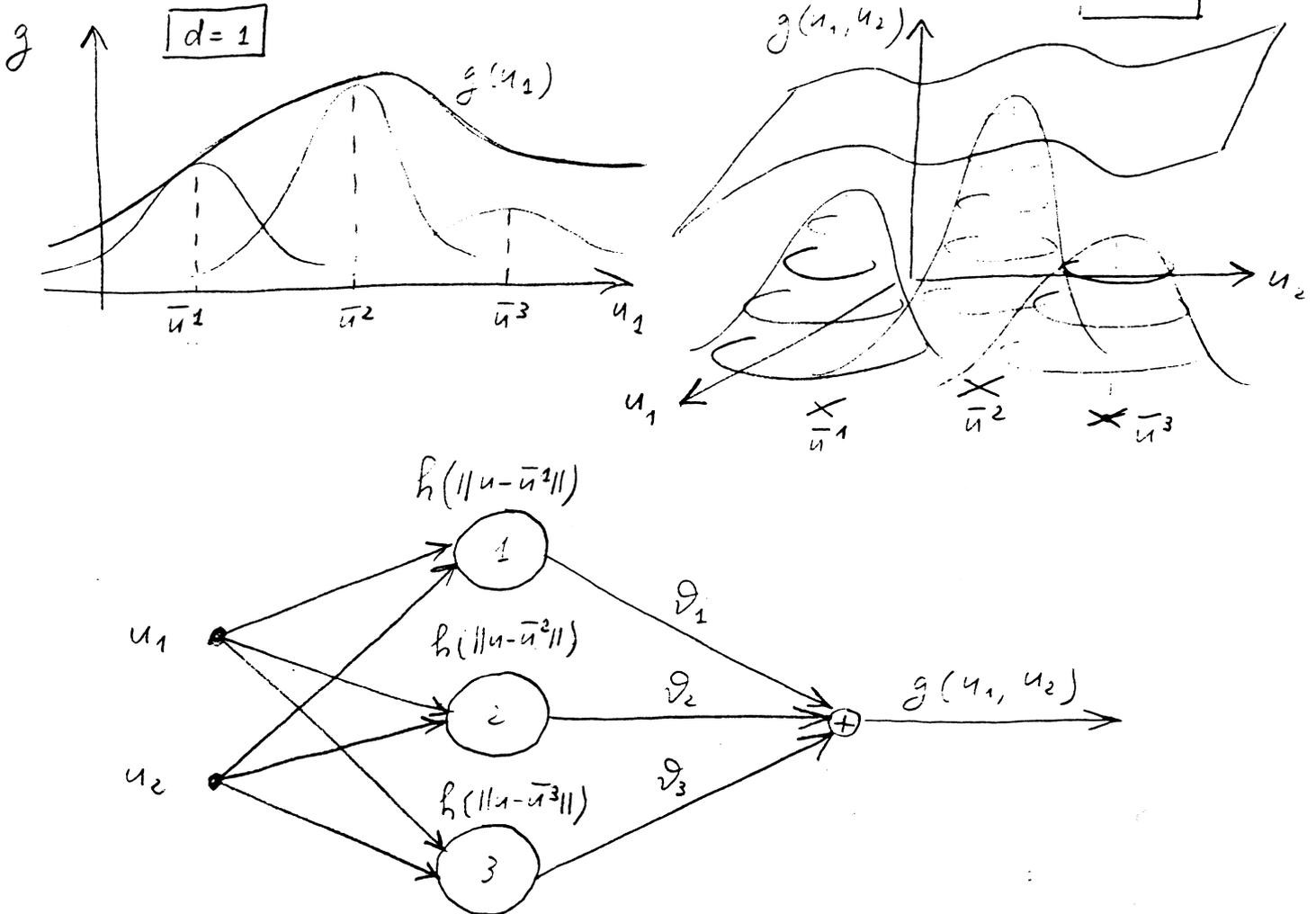
*Alcune tecniche basate sulla crossvalidazione:*

- Comincio con pochi neuroni e ne stimo i parametri. Poi "faccio crescere" la rete aggiungendo altri neuroni. Per decidere quando fermarmi uso la crossvalidazione.
- Optimal brain damage: uso una rete sovradimensionata e distruggo nodi e/o pesi che contano poco ("pruning") fino a quando riesco a migliorare  $SSR^V$ .
- Uso una rete sovradimensionata e la addestro con un algoritmo iterativo monitorando i progressi mediante crossvalidazione. Invece di arrivare alla convergenza, mi fermo quando le prestazioni non migliorano più.
- ...

*Regola empirica:* Dato che n° parametri  $\gg$  n° neuroni, è bene che il n° neuroni sia  $\ll$  del numero  $N$  dei dati.

# RETI NEURALI A BASE RADIALE

*Idea:* Rappresento  $g(u_1, u_2, \dots, u_d)$  come una combinazione lineare di funzioni "a campana".



$$g(u_1, u_2, \dots, u_d) = \sum_{k=1}^n \theta_k h(\|u - \bar{u}^k\|)$$

Se la forma  $h(\cdot)$  delle campane è fissata, i parametri liberi sono i centri  $\bar{u}^k$  e le ampiezze  $\theta_k$ .

Tipicamente,  $h(r)$  è una funzione a simmetria radiale:

- $h(r) = e^{-\frac{r^2}{2c^2}}$
- $h(r) = \sqrt{r^2+c^2}$
- $h(r) = (r^2+c^2)^\beta$ ,  $0 < \beta < 1$
- $h(r) = (r^2+c^2)^{-\beta}$ ,  $\beta > 0$

$c, \beta$ : parametri liberi che caratterizzano  $h(r)$ .

*Osservazioni:*

- Anche le Radial Basis Function Neural Networks (RBF NN) sono approssimatori universali.
- Parentela con i modelli fuzzy (basta interpretare  $h(r)$  come una "membership function").
- Fatto interessante: una volta note le coordinate dei centri  $\bar{u}_k$ , la stima di  $\theta$  è un problema lineare (risolvibile mediante LS).

*Idea:* Spezzo il problema dell'addestramento in due fasi:

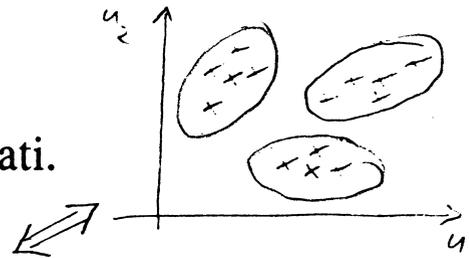
- 1) Posizionamento dei centri  $\bar{u}_k$
- 2) Stima delle ampiezze  $\theta_k$

La forma di  $h(r)$  viene di solito scelta a priori (sono comunque possibili aggiustamenti dei parametri  $c, \beta$ ).

## Tecniche per il posizionamento dei centri:

### (i) $n^\circ$ dei centri fissato a priori

- Griglia uniforme (inutile però mettere tanti centri dove ho pochi dati).
- Scelta casuale di un sottoinsieme dei dati.
- Divido i dati  $\{u^{(k)}\}$  in  $n$  "grappoli" con un opportuno algoritmo di "clustering" (K-MEANS, per esempio) e uso come centri i baricentri dei grappoli.
- Algoritmi più sofisticati: ottimizzazione simultanea di tutti i parametri (coordinate  $\bar{u}^k$  dei centri e ampiezze  $\theta_k$ ). Difficoltà numeriche (stima nonlineare). Ne vale la pena?



### (ii) $n^\circ$ dei centri non fissato a priori

- Stepwise regression considerando come regressori candidati tutte le funzioni radiali centrate in corrispondenza degli  $N$  dati.

*Scelta del  $n^\circ$  di centri:* E' consigliabile ricorrere alla crossvalidazione (si procede per tentativi cercando di minimizzare  $SSR^v$ ).

## ESEMPIO: STIMA DEL RENDIMENTO VOLUMETRICO $\eta$

Basandosi sui risultati dell'identificazione mediante modelli polinomiali sono state considerate solo  $n$ ,  $p_{man}$  come variabili indipendenti

$$\eta = \eta(n, p_{man})$$

### *RBF*

- $h(r)$ : gaussiana
- Posizionamento dei centri mediante K-MEANS
- n° dei neuroni e parametro  $c$  scelti mediante crossvalidazione

### *MLP*

- $h(z) = \tanh(z)$
- Stima dei pesi mediante backpropagation.
- Poiché il risultato della stima dipende dall'inizializzazione dei pesi, la backpropagation è stata ripetuta usando inizializzazioni casuali.

*Sia per RBF che MLP è essenziale normalizzare le variabili indipendenti  $n$ ,  $p_{man}$ .*

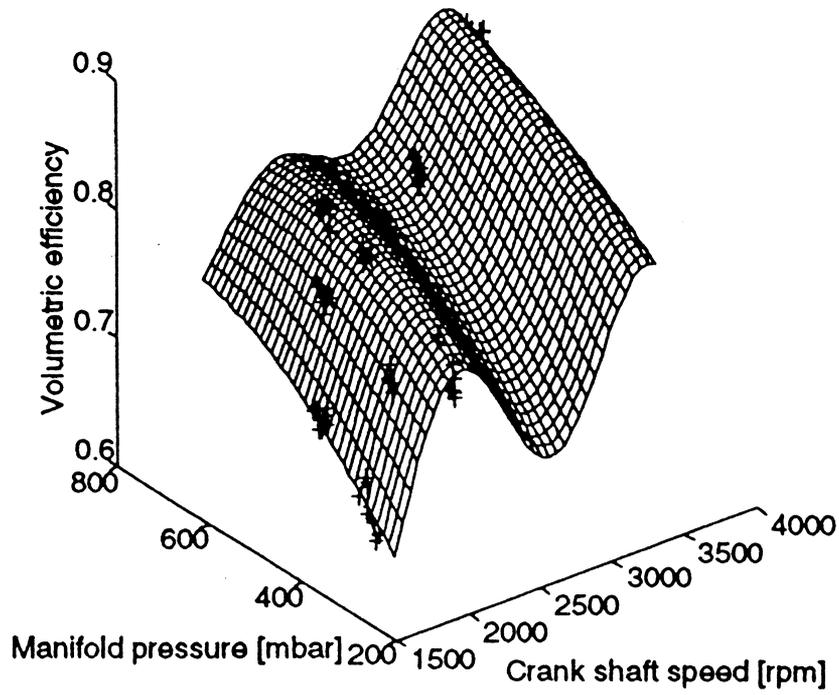


Figure 9: Radial basis function neural-network identification of  $\eta_v$  as a function of  $n$ ,  $p_{man}$  (8 neurons). The symbol “+” indicates the validation data.

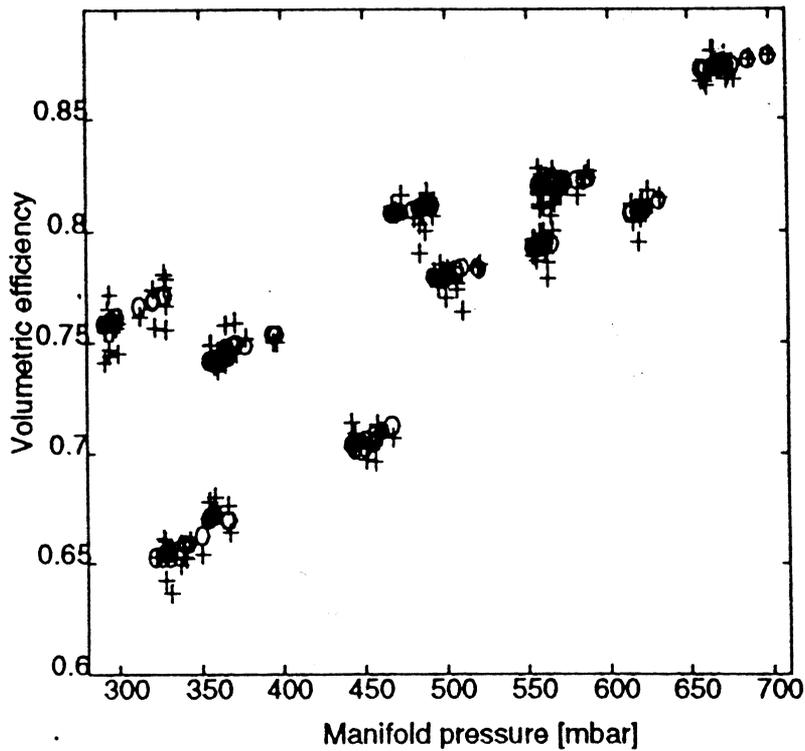


Figure 10: Radial basis function neural-network identification of  $\eta_v$ . Validation (+) versus predicted (o) data.

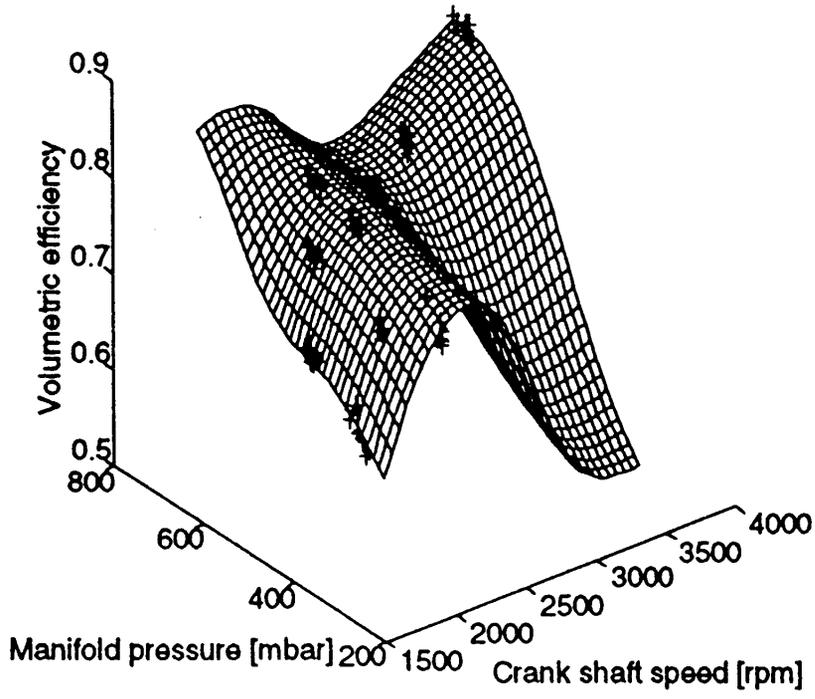


Figure 11: Multi-layer perceptron identification of  $\eta_v$  as a function of  $n$ ,  $p_{man}$  (5 neurons). The symbol “+” indicates the validation data.

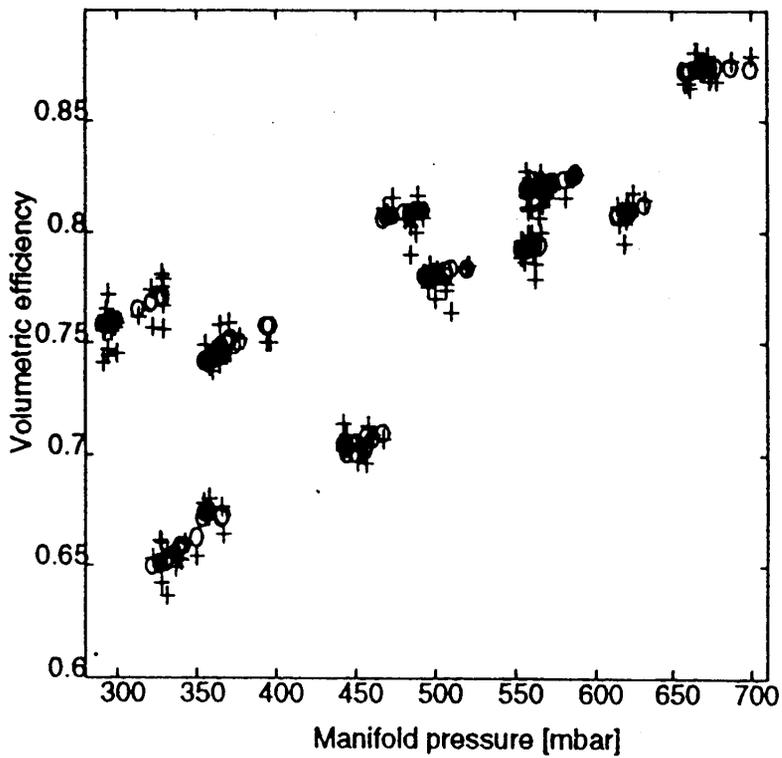


Figure 12: Multi-layer perceptron identification of  $\eta_v$  (5 neurons). Validation (+) versus predicted (o) data

**Table 3. RBF neural network identification: sum of squared residuals (SSR) and largest standard deviation of identified  $b_t$  parameters ( $M=5-9$ ).**

Number of neurons	SSR	Largest SD
5	0.0079	2.7%
6	0.0077	31.2%
7	0.0075	28%
8	0.0071	27%
9	0.0072	11.8%

**Table 4. MLP neural network identification: sum of squared residuals (SSR) ( $M=3-7$ ).**

Number of neurons	SSR
3	0.0072
4	0.0072
5	0.0069
6	0.0075
7	0.0074

**Table 5. Performances of the different identification method: SSR on the validation data and SD of residuals.**

Identification method	SSR	SD of residuals
<i>Additive model</i>	0.0076	0.0070
<i>Polynomial model</i>	0.0107	0.0083
<i>RBF (8 neurons)</i>	0.0071	0.0068
<i>ML (5 neurons)</i>	0.0069	0.0067

## CONCLUSIONI

- Reti MLP: strumento versatile. Problemi di minimi locali e di tempo di calcolo.
- Reti RBF: una volta fissati i centri, l'addestramento si riduce ad una stima  $LS$ . lineare Problema: posizionamento dei centri.
- Nella stima del rendimento volumetrico le reti neurali danno risultati (leggermente) migliori dei modelli polinomiali, ma la loro identificazione è più complicata.
- Per identificare "a scatola nera" una funzione  $g(u_1, u_2, \dots, u_d)$  senza conoscerne a priori la struttura funzionale, è necessario avere molti dati, soprattutto al crescere di  $d$  ("maledizione della dimensionalità"). Se, grazie alle leggi fisiche, conosco la struttura di  $g(u_1, u_2, \dots, u_d, \theta)$  a meno di (pochi) parametri incogniti  $\theta$ , posso ottenere risultati migliori mediante la stima parametrica. Però, se  $d$  è "piccolo" rispetto al n° di dati disponibili, l'identificazione a scatola nera mi consente di scavalcare tutti i problemi di modellistica fisica.