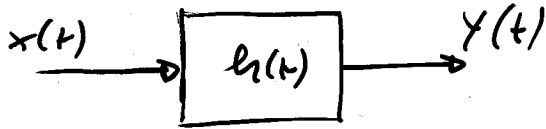


# 5. LA TEORIA DEI CIRCUITI ELE AHN

Tempo continuo (TC) - circuiti L.P.C.C.

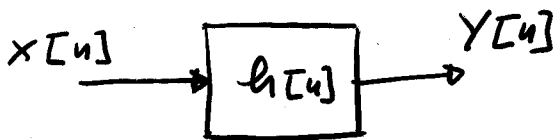


legame in-ant

$$y(t) = \int_{-\infty}^{\infty} h(\tau) x(t-\tau) d\tau \quad *$$

$$y(t) = \sum_{i=0}^p b_i x^{(i)}(t) + \sum_{j=1}^q a_j y^{(j)}(t)$$

Tempo discreto (TD)



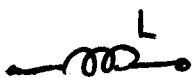
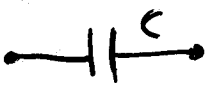
legame in-ant

$$y[n] = \sum_{k=0}^N h[k] x[n-k] \quad n=0, 1, 2, \dots$$

$$y[n] = \sum_{i=0}^p d_i x[n-k] + \sum_{j=1}^q c_j y[n-k]$$

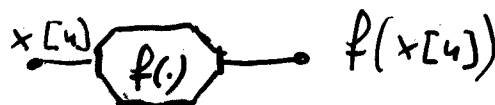
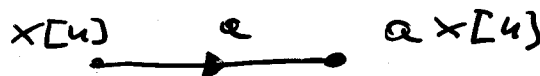
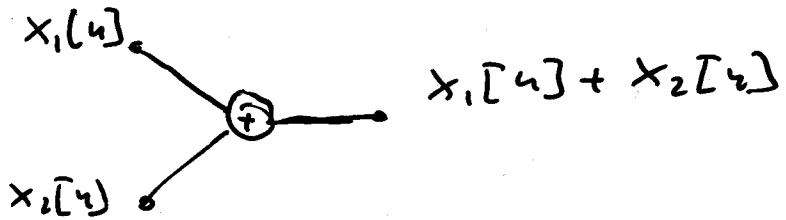
Rappresentazione grafica circuiti

TC

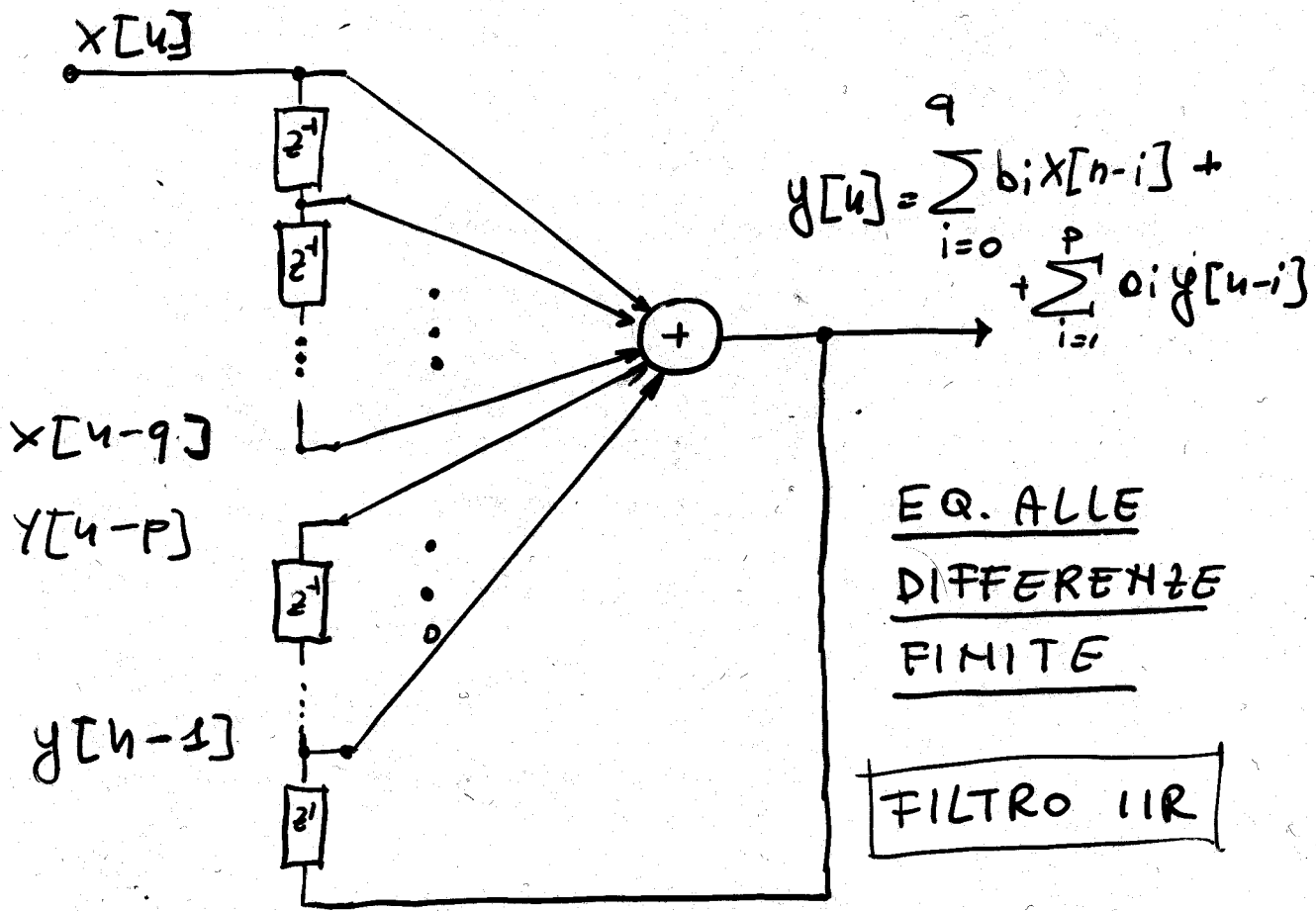
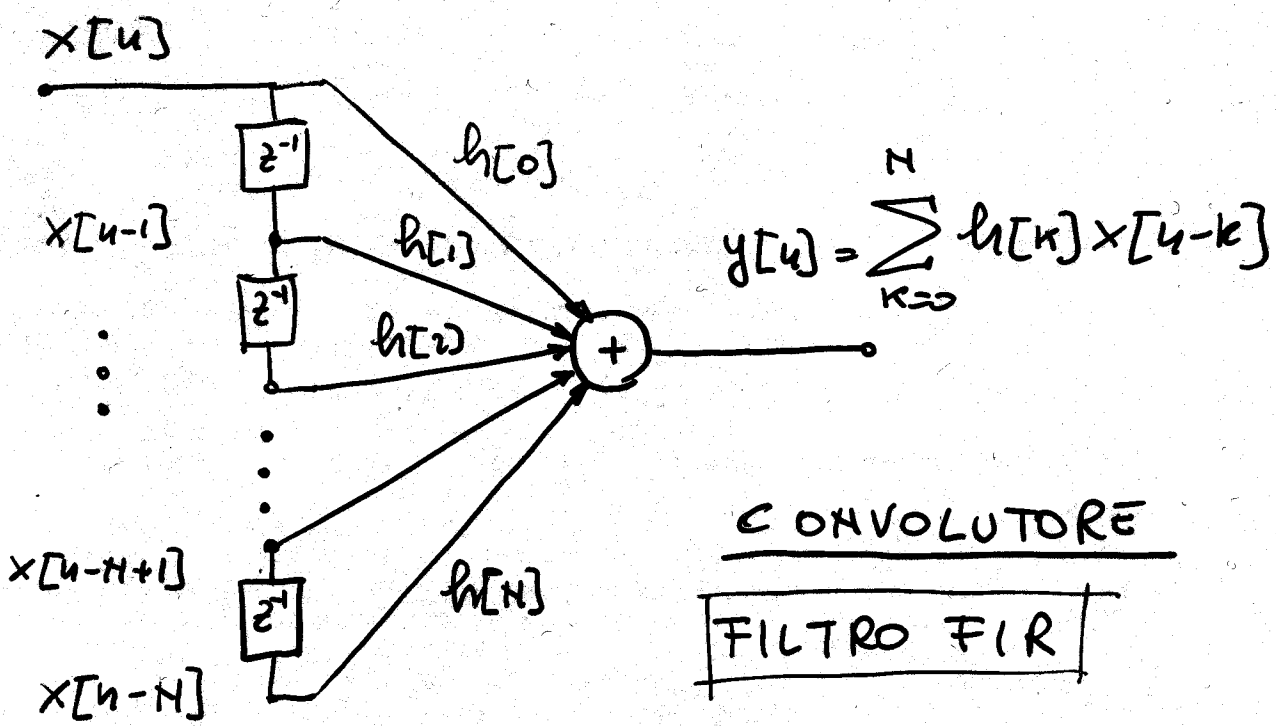


⋮

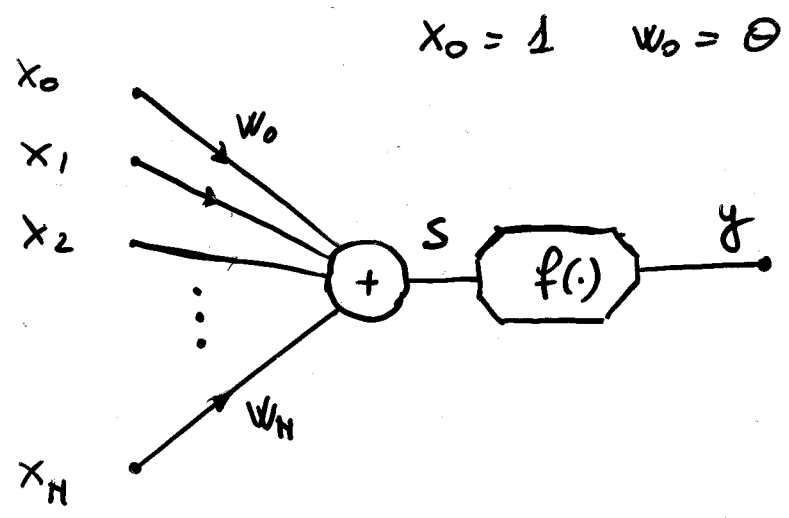
TD



ES

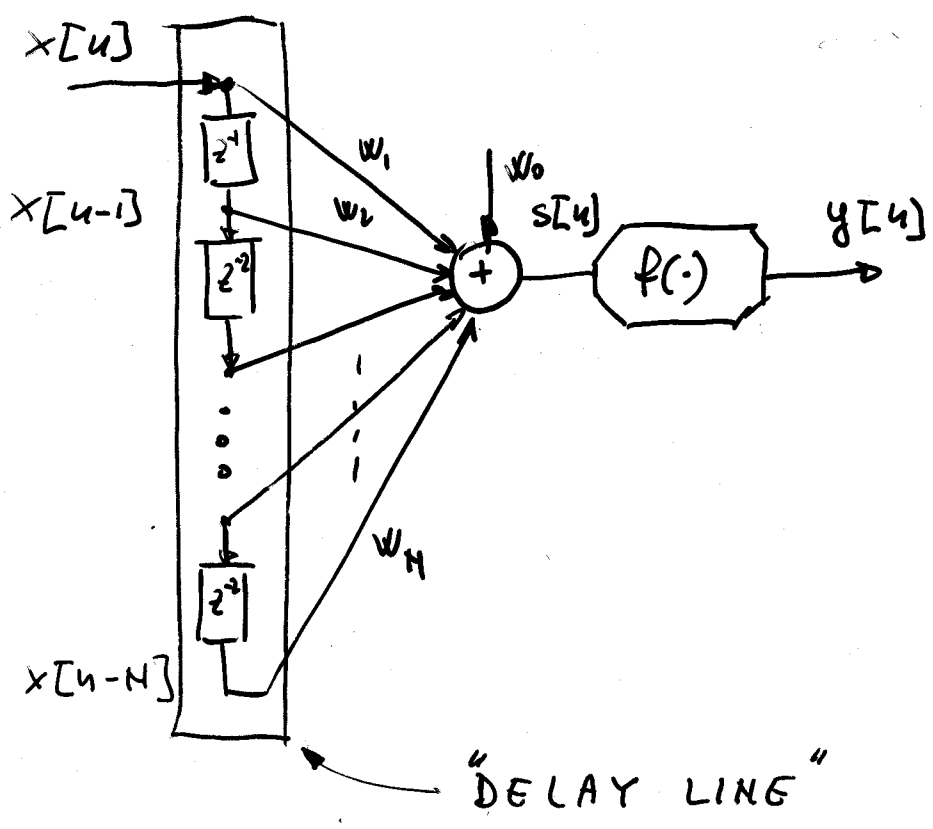


ES



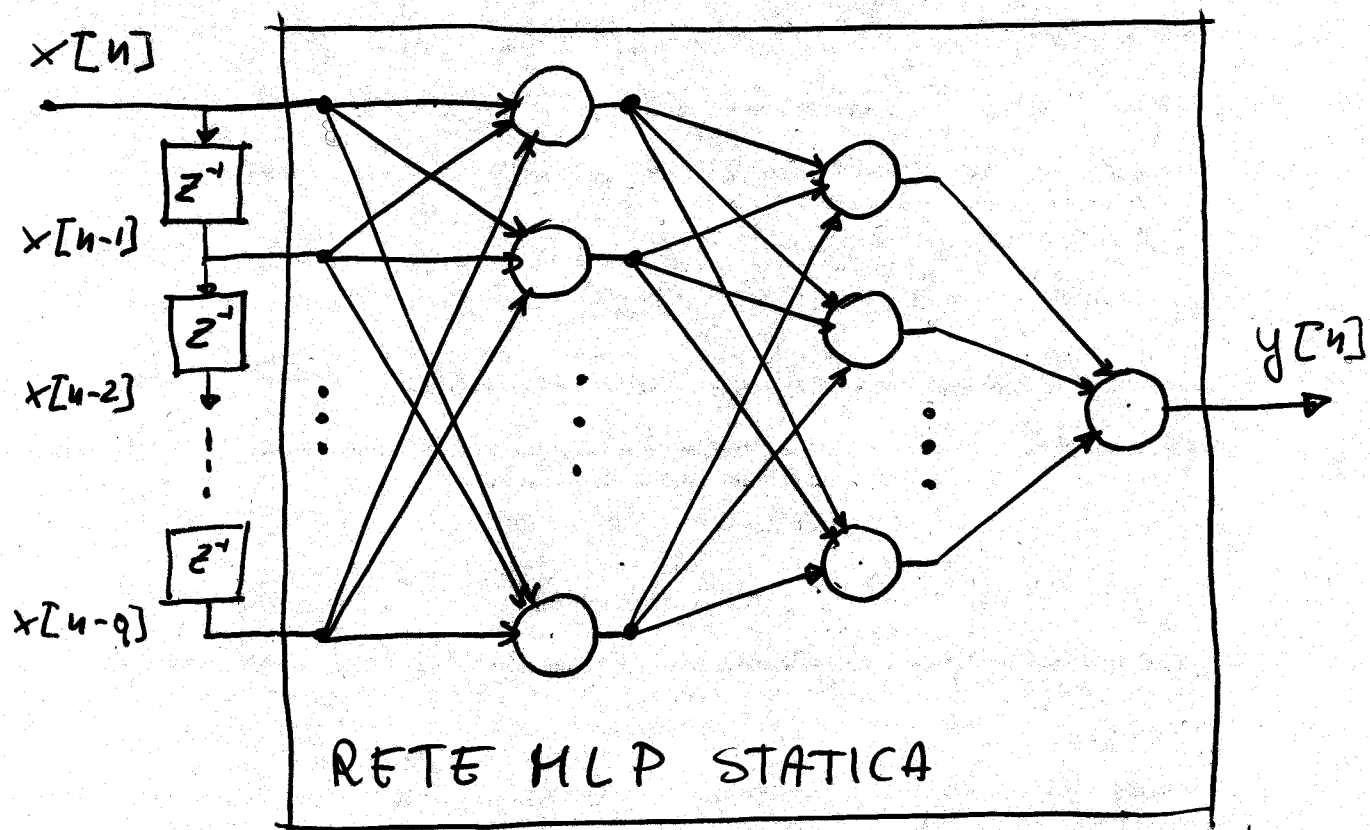
NEURONE  
FORMALE

introduces la variable tempo



# RETI STATICHE CON DINAMICA ESTERNA

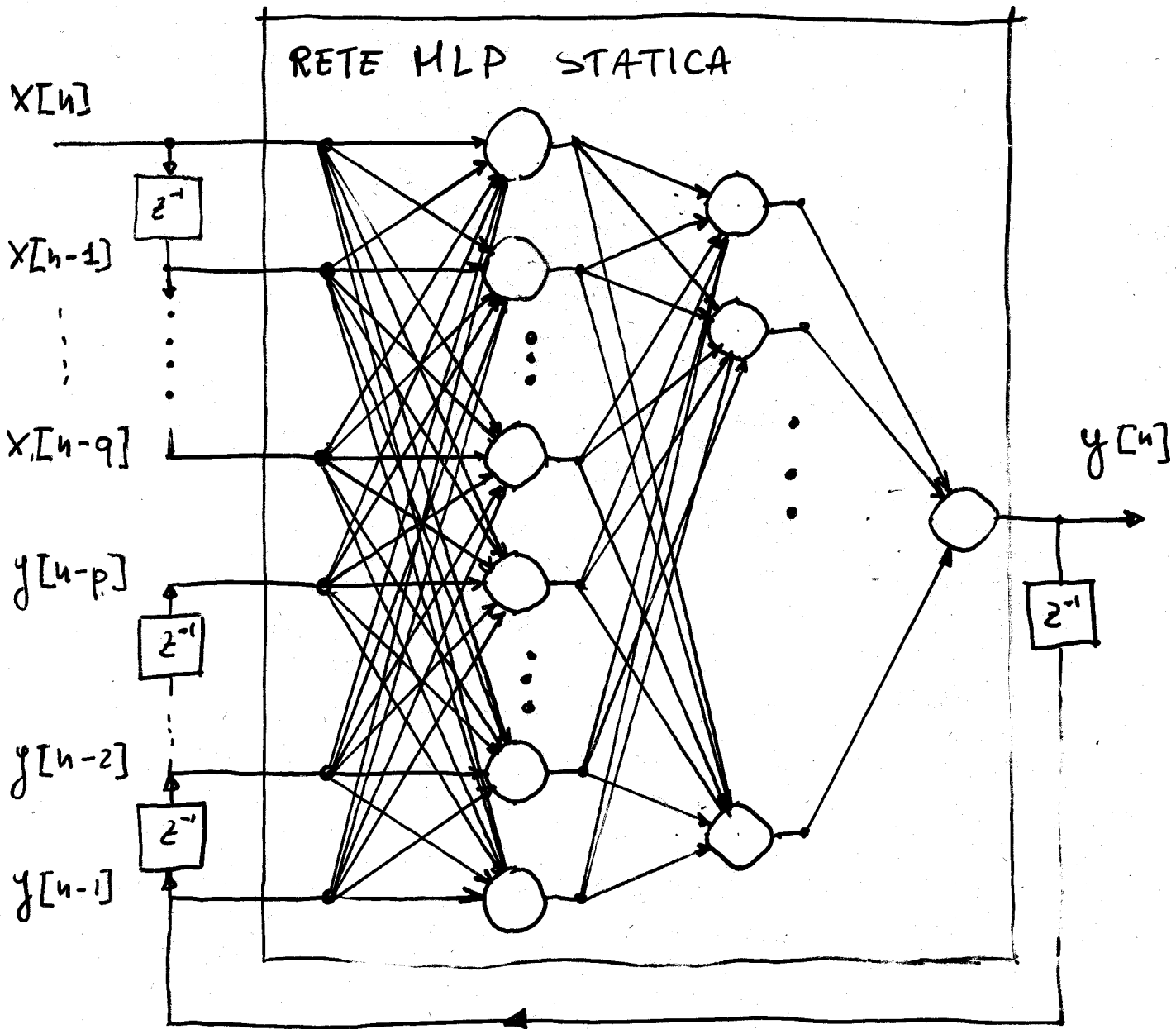
## Reti con dinamica FIR



$$y[n] = F(x[n], x[n-1], \dots, x[n-q])$$

Realizza un circuito FIR non lineare

# Rete ricorrente con dinamica esterna IIR



$$y[n] = F(x[n], x[n-1], \dots, x[n-q], y[n-1], \dots, y[n-p])$$

Realizza un circuito IIR non lineare

## 6. PROPRIETA' MATEMATICHE DELLE ANN DINAMICHE

- Reti ricorrenti con dinamica temporale FIR o IIR sono circuiti non lineari adattativi

- TEOREMA: dovuto a T. Chen, H. Chen (1993)

Sia  $\mathcal{U}$  un insieme compatto in  $C[t_a, t_b]$ ,  
 $F$  un funzionale continuo definito in  $\mathcal{U}$ ,  
 $\sigma(\cdot)$  una sigmoide generalizzata limitata (non necessariamente continua)  $\Rightarrow \forall \varepsilon > \phi$

$\exists$   $m+1$  punti  $t_a < t_1 < t_2 \dots < t_m < t_b$ ;

$\exists$  un intero  $N$ ,

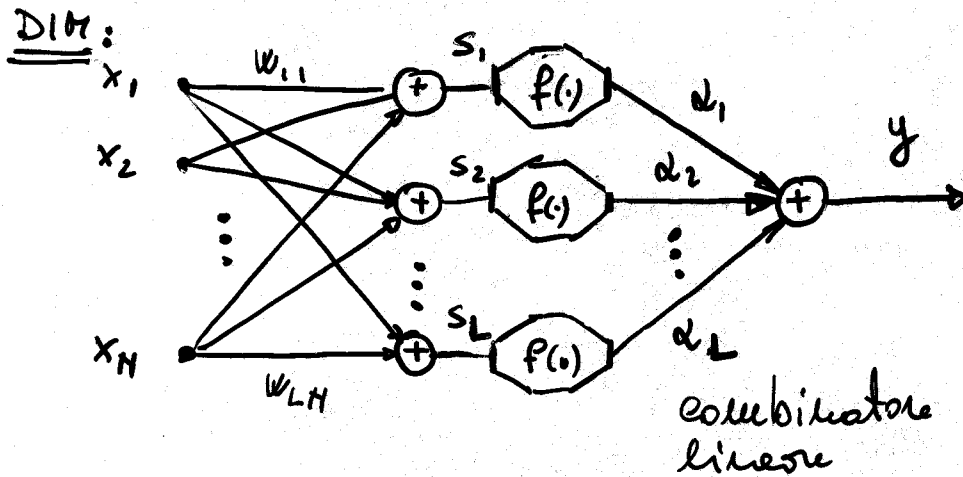
$\exists \alpha_i, \theta_i, w_{ij} \in \mathbb{R}$  con  $i=1, \dots, N, j=0, \dots, m \dots$

$$\left| F[x(t)] - \sum_{i=1}^N \alpha_i \sigma \left( \sum_{j=0}^m w_{ij} x(t_j) + \theta_i \right) \right| < \varepsilon \quad \square$$

Conseguente: Una rete con dinamica temporale FIR è un approssimatore universale per ogni funzionale continuo in  $\mathcal{U}$  compatto in  $C[t_a, t_b]$

Teorema:

Una rete neurale con un solo strato nascosto è rappresentabile con una serie di Volterra troncata.



$$\text{sviluppo } f(s_i) = \frac{1}{1 + e^{-s_i}} = \sum_{k=0}^P c_k s^k + o(s^{P+1})$$

$$y = \sum_{i=1}^L \alpha_i \left[ c_0 + c_1 \sum_{j=1}^N w_{ij} x_j + c_2 \left( \sum_{j=1}^N w_{ij} x_j \right)^2 + \dots + c_p \left( \sum_{j=1}^N w_{ij} x_j \right)^p \right]$$

utilizzo la relazione

$$\left( \sum_{j=1}^N w_{ij} x_j \right)^p = \sum_{j_1=1}^N \dots \sum_{j_p=1}^N w_{ij_1} x_{j_1} \dots w_{ij_p} x_{j_p} \Rightarrow$$

$$y = \sum_{i=1}^L \alpha_i c_0 + \sum_{j_1=1}^N \left( \sum_{i=1}^L \alpha_i c_1 w_{ij_1} \right) x_{j_1} +$$

$$+ \sum_{j_1=1}^N \sum_{j_2=1}^N \left( \sum_{i=1}^L \alpha_i c_2 w_{ij_1} w_{ij_2} \right) x_{j_1} x_{j_2} + \dots +$$

$$+ \sum_{j_1=1}^N \dots \sum_{j_p=1}^N \left( \sum_{i=1}^L \alpha_i c_p w_{ij_1} \dots w_{ij_p} \right) x_{j_1} \dots x_{j_p}$$

sostituendo le quantità:

$$h_0 = \sum_{i=1}^L \alpha_i c_0$$

$$h_{j_1} = c_1 \sum_{i=1}^L \alpha_i w_{ij_1}$$

$$h_{j_1 j_2} = c_2 \sum_{i=1}^L \alpha_i w_{ij_1} w_{ij_2}$$

⋮

$$h_{j_1 \dots j_p} = c_p \sum_{i=1}^L \alpha_i w_{ij_1} \dots w_{ij_p}$$

NUCLEI DI VOLTERRA

Si ottiene l'espressione della serie di Volterra di ordine  $p$  a  $N$  ingressi.

$$\begin{aligned} y = & h_0 + \sum_{j_1=1}^N h_{j_1} x_{j_1} + \sum_{j_1=1}^N \sum_{j_2=1}^N h_{j_1 j_2} x_{j_1} x_{j_2} + \dots + \\ & + \sum_{j_1=1}^N \sum_{j_2=1}^N \dots \sum_{j_p=1}^N h_{j_1 j_2 \dots j_p} x_{j_1} x_{j_2} \dots x_{j_p} \end{aligned}$$

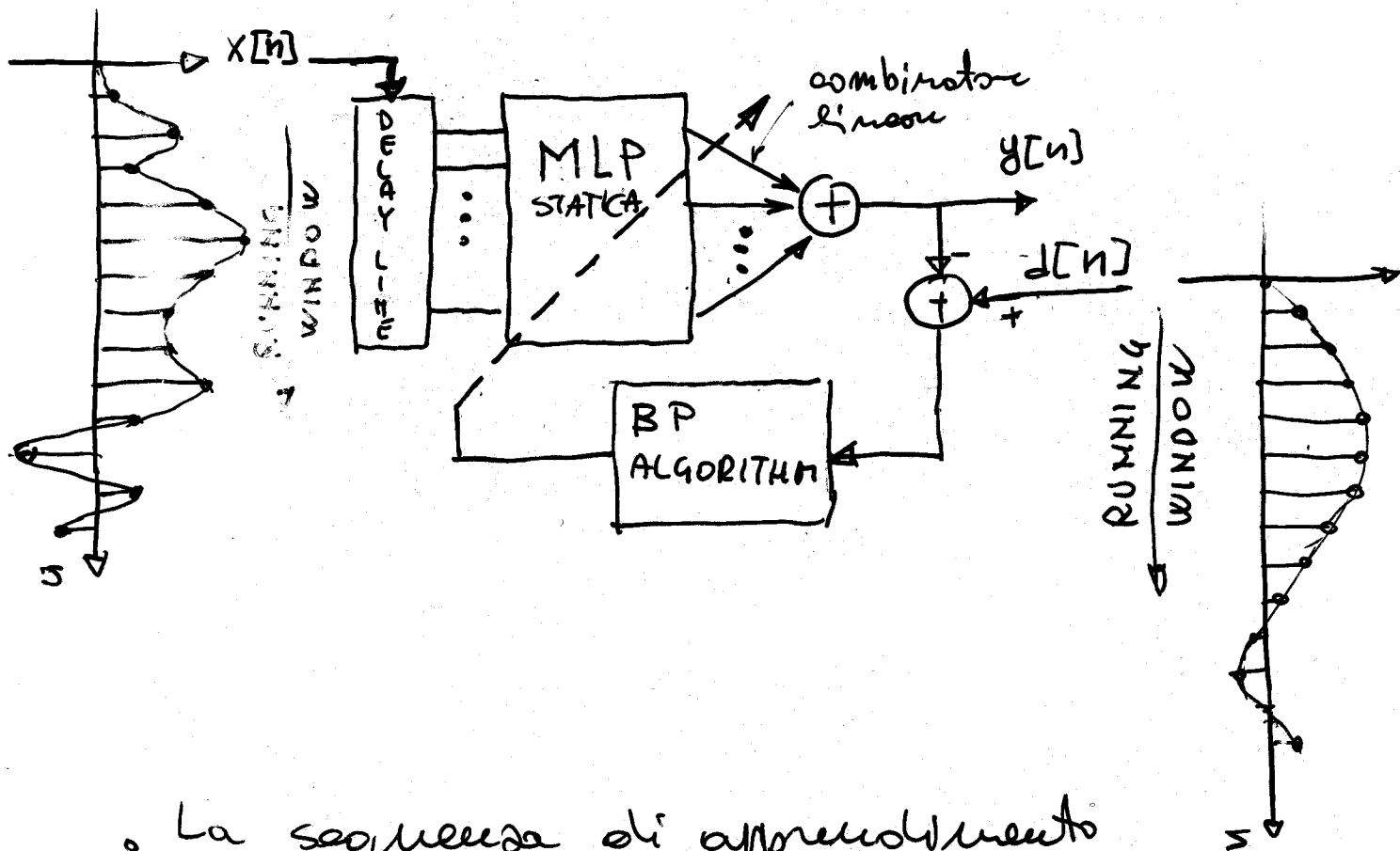
C.V.D.

conseguente: stessa del precedente teorema  
+ dualità tra MLP e  
serie di Volterra.

# Apprendimento in reti statiche e/o dinamiche esterne

## • Reti FIR

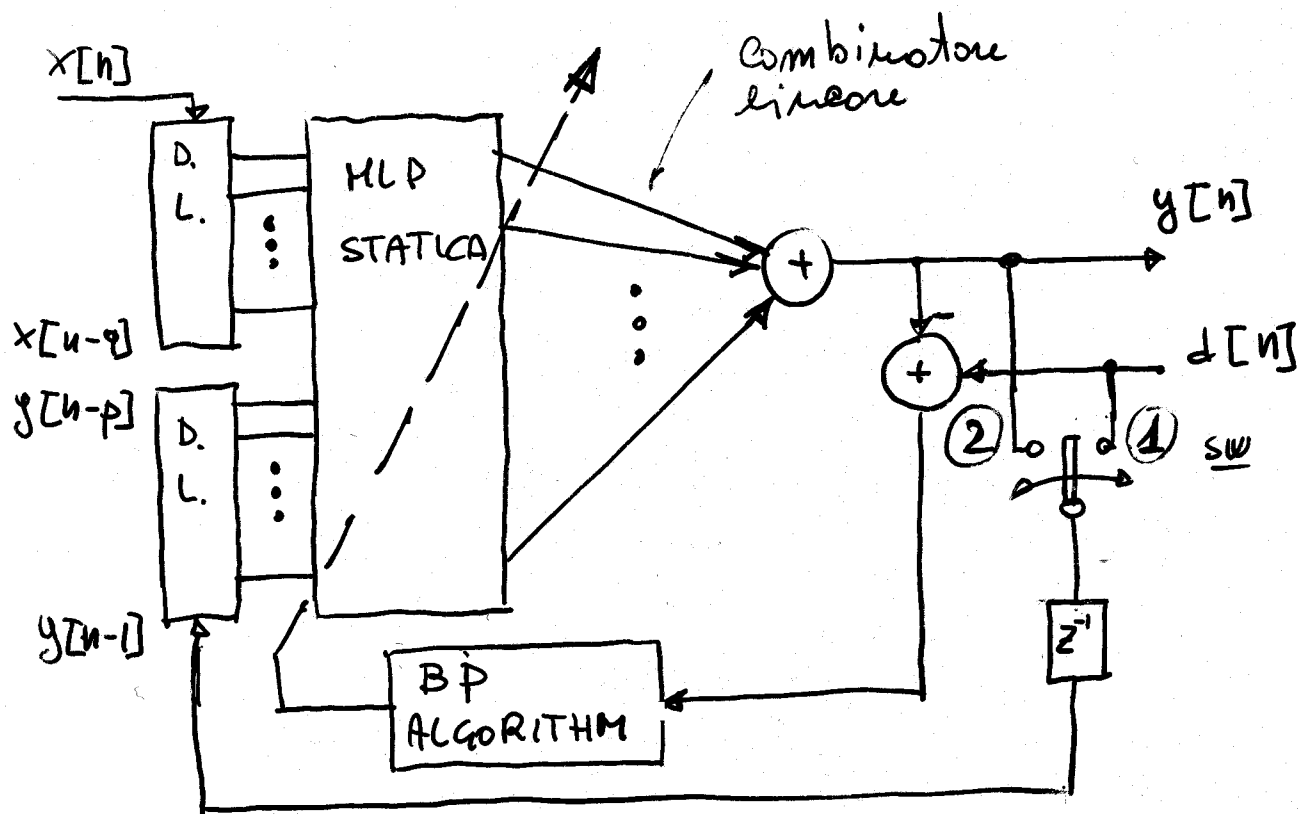
L'apprendimento avviene come, per i filtri adattativi lineari, facendo scorrere il segnale all'ingresso e in uscita della rete. La BP è identica al caso statico



- La sequenza di apprendimento è presentata alla rete per un numero "adeguato" di volte o fino a quando l'errore  $E$  risulta  $E < \text{criterio}$  fissato a priori.

# Reti IIR

L'apprendimento nelle reti ricorsive IIR può essere effettuato con due diversi approcci:



(1.) "Equation-error": il segnale di retroazione coincide con il segnale  $d[n]$  dell'uscita desiderata (sw pos 1).

(2.) "Output-error": il segnale di retroazione coincide con la uscita  $y[n]$  della rete. (sw pos 2)

# RETI NEURALI CON DINAMICA INTERNA

- I neuroni sono dotati di una propria dinamica.
- Ogni sinapsi è realizzata come un filtro FIR o IIR.
- L'algoritmo di apprendimento per reti MLP dinamiche (DMLP), generalizzazione della BP, chiamato "backpropagation through time" (BPT), è non causale.
- Approssimazioni on-line della BPT, ausentano una complessità minore con prestazioni paragonabili alla BPT.
- Le prestazioni di reti con dinamica interna sono sensibilmente superiori alle reti statiche con dinamica esterna.

## 8. COMPLESSITÀ DELLE RETI NEURALI

### • Complessità strutturale

È riferita alle dimensioni, in termini di n° di neuroni e n° di connessioni necessarie alla rete per un determinato "mapping" tra ingresso e uscita

### • Complessità computazionale

Correlata alla complessità strutturale, indica il tempo necessario alla rete per calcolare l'uscita (complessità FORWARD), per calcolare la BP (complessità BACKWARD), e per l'intera fase di apprendimento.

⇒ Nelle applicazioni pratiche la complessità strutturale e/o computazionale ha un ruolo di prim'importanza.

⇒ Diminuendo la complessità strutturale in genere si ha anche una diminuzione della complessità computazionale.

- La funzione di attivazione ha un ruolo importante per quanto riguarda: la complessità (computazionale e strutturale), e le capacità di generalizzazione delle rete

- Nelle implementazioni circuitali numeriche, la complessità della fase forward, non dipende dal tipo di funzione di attivazione.

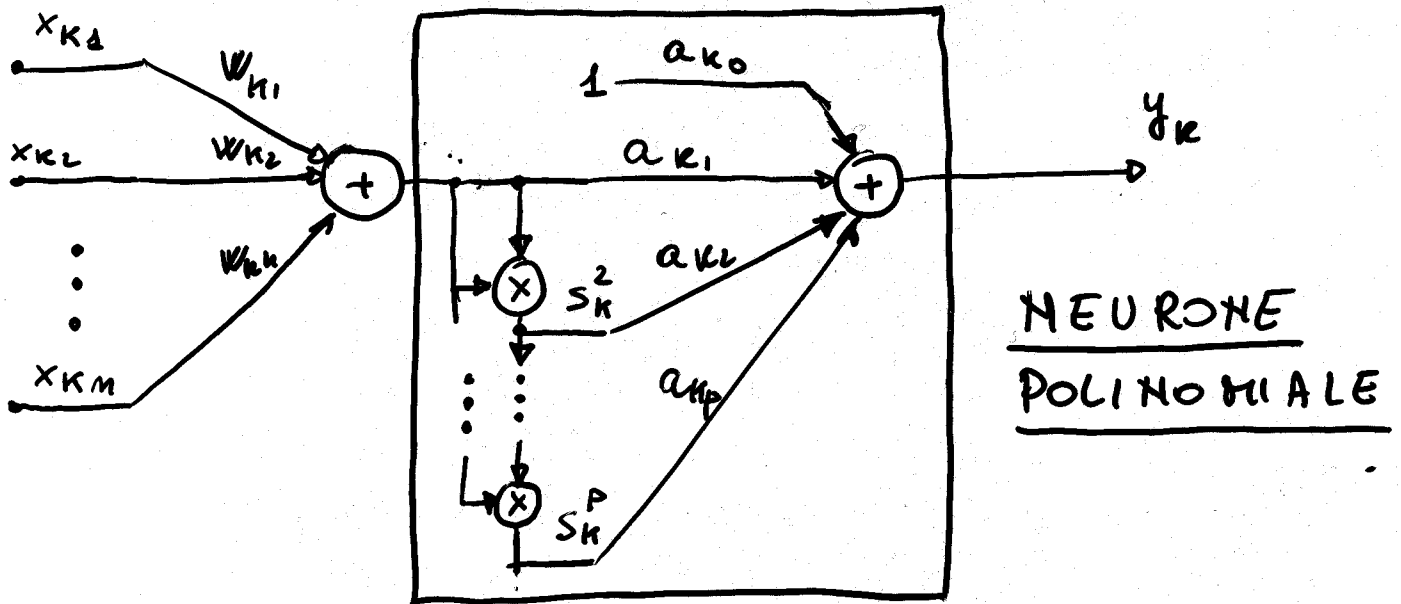
- Una funzione di attivazione adattativa consente, a parità di capacità di elaborazione, una rete neurale con una complessità sensibilmente minore.

- Alcuni tipi di funzione di attivazione adattativa:

- Sigmoide (con slope e gain adattivi).
- Polinomiali
- Tabellone (solo numeriche)
- Sigmoide Generalizzate

# 9. ANN CON FUNZIONE DI ATTIVAZIONE ADATTATIVA

## Reti neurali polinomiali (APNN)



(Adaptive Polynomial Activation Function)

### • Apprendimento:

determinazione dei vettori w e A

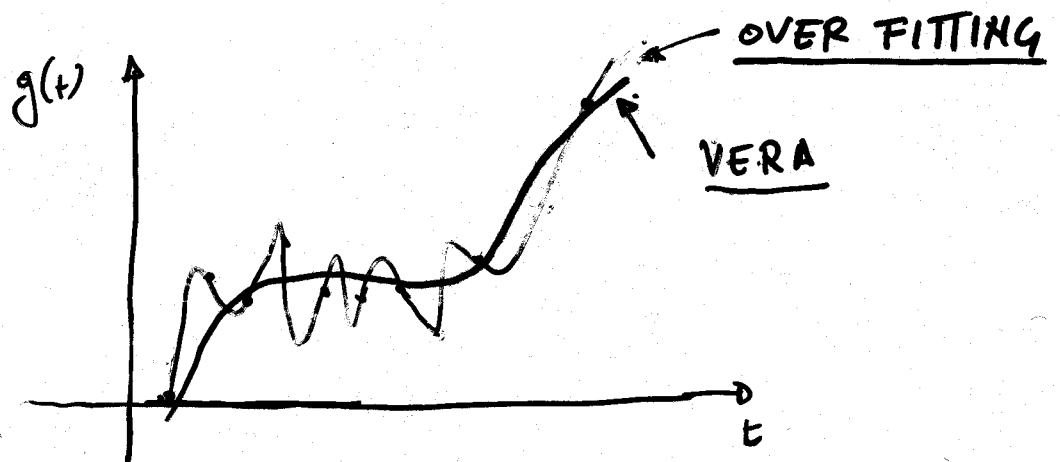
$$\begin{cases} \underline{w}_{k+1} = \underline{w}_k + \Delta \underline{w}_k \\ \underline{A}_{k+1} = \underline{A}_k + \Delta \underline{A}_k \end{cases}$$

$$\frac{\partial \epsilon_k^2}{\partial w_j} = -2 \epsilon_k \left( \frac{\partial f(\cdot)}{\partial s_k} \right) x_j = -2 \epsilon_k \left( \sum_{i=1}^p i \cdot a_i s_k^{i-1} \right) x_j$$

$$\frac{\partial \epsilon_k^2}{\partial a_j} = -2 \epsilon_k \left( \frac{\partial f(\cdot)}{\partial a_j} \right) = -2 \epsilon_k s_k^j$$

# Caratteristiche APNN

- Realizzato una serie di Voltana traseata con vimeoli dovuti alla topologia delle rete.
- Buone prestazioni in termini di complessità.
- Tendenza all'"over-fitting" tipica degli approssimatori polinomiali

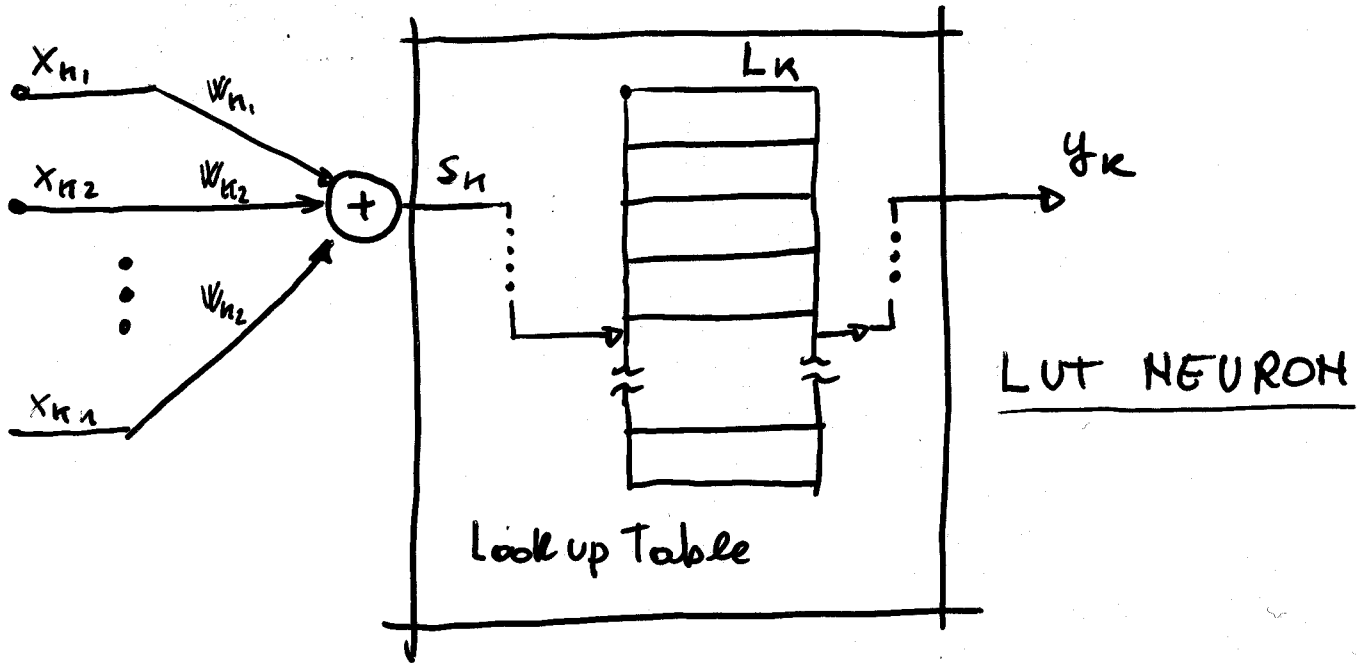


La rete tende ad adattarsi ai dati disponibili ma non "impara" il modello sottostante.

⇒ scarse capacità di generalizzazione.

# Reti neurali tabellari (LUT-NN)

- La  $f(\cdot)$  è realizzata in forma numerica tabellare



- Apprendimento

$w$   $\rightarrow$   $\Delta$ -rule

LUT  $\rightarrow$  gradiente numerico

- Vantaggi:

implementazioni HW veloci della fase di Apprendimento.

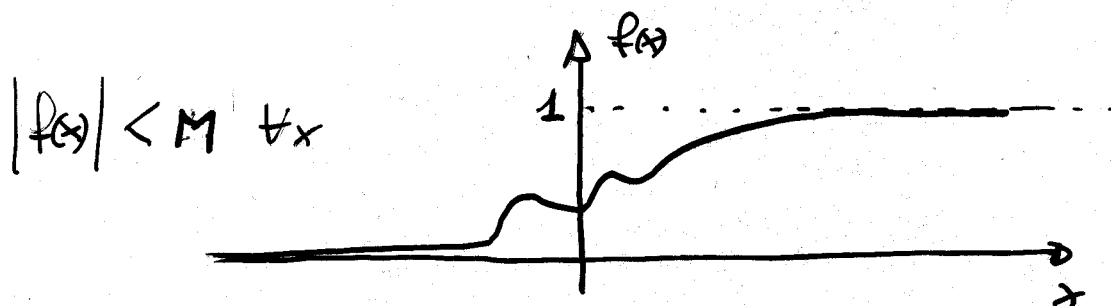
- Svantaggi:

- non adatta a problemi di elaborazione di sequenze
- possibile instabilità numerica

# Reti neurali Sigmoide Generalizzate (SG-NH)

- Sigmoide generalizzate

$$f(x) = \begin{cases} 1 & x \rightarrow \infty \\ 0 & x \rightarrow -\infty \end{cases}$$



- Le reti neurali SG sono degli approssimatori universali (Teor. Hornik, Stinchcombe, White).

- Supera i problemi delle reti polinomiali

- limitatezza

- overfitting

⋮

- Supera i problemi delle LUT

- elevato S/H per signal processing

- stabilità numerica

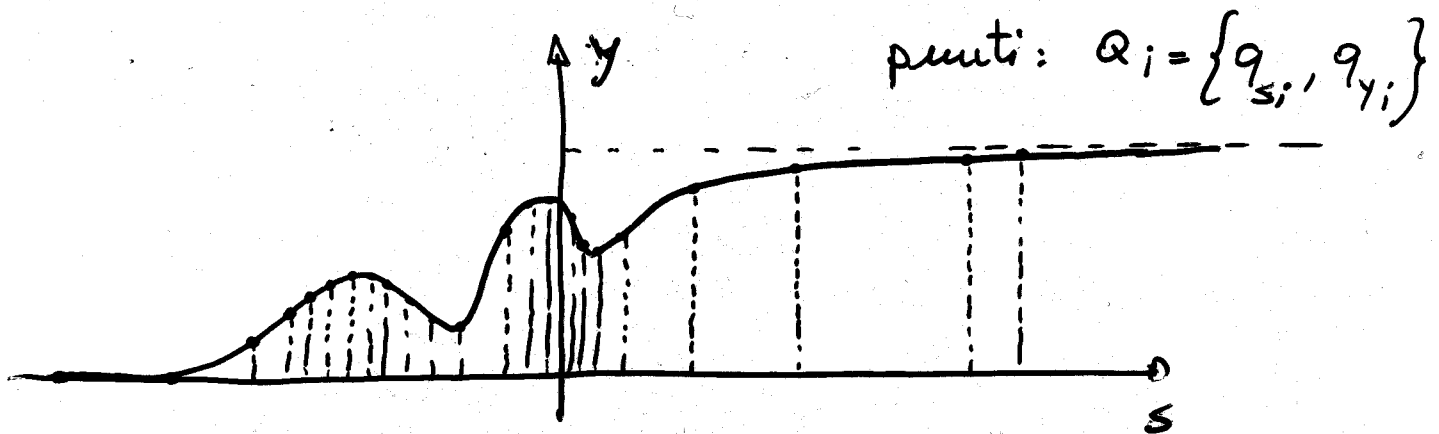
- funzione continua

⋮

# Struttura Neurale SG

- Possibile soluzione:

la funzione di attivazione è generata con uno schema di interpolazione di una LUT-NEURON



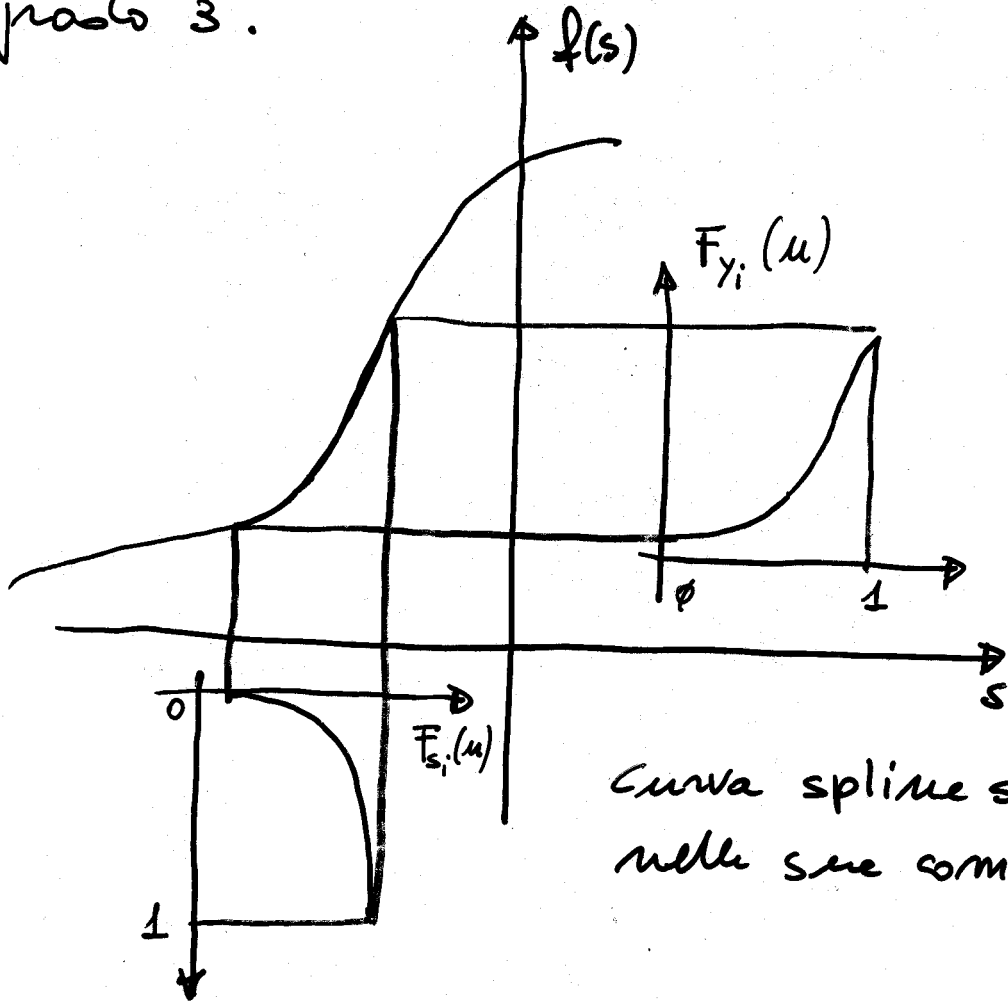
- Approccio usato:

interpolazione locale su base spline di Catmull-Rom.

$$f(s) = \bigcup_{i=0}^{n-3} F_i(u)$$

$$F_i(u) = \{F_{s_i}(u), F_{y_i}(u)\} \quad u \in [0, 1]$$

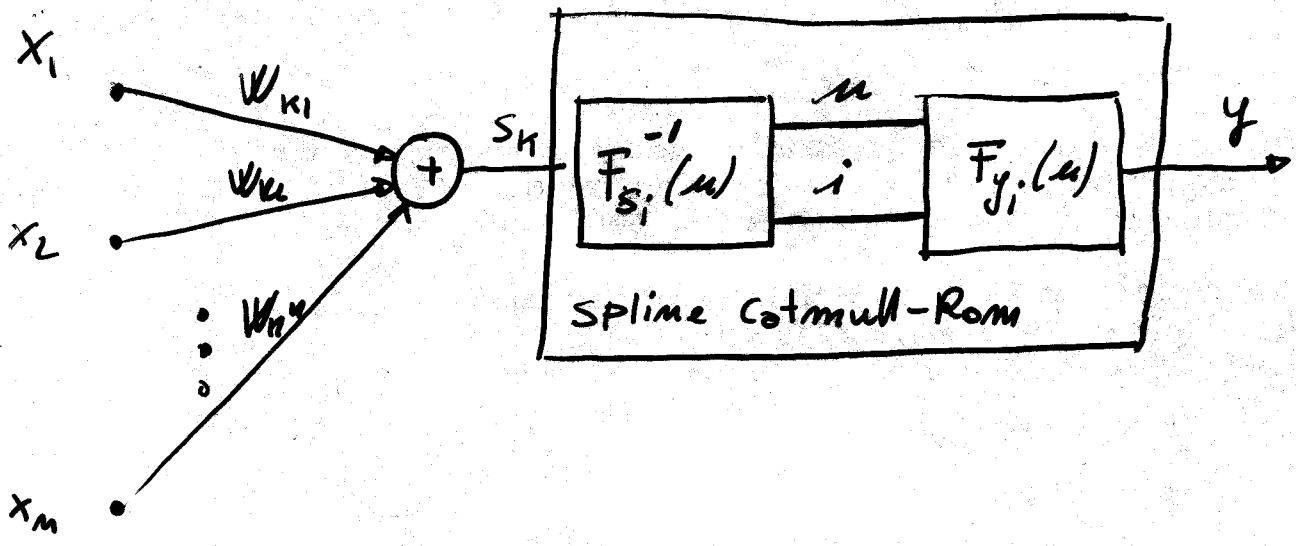
La  $F_i(u)$  è una funzione polinomiale di grado 3.



Curva spline separate nelle sue componenti

$$F_{s_i}(u) = [u^3 \ u^2 \ u \ 1] \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & \emptyset & 1 & \emptyset \\ \emptyset & 2 & \emptyset & \emptyset \end{bmatrix} \begin{bmatrix} q_{s1} \\ q_{s2} \\ q_{s3} \\ q_{s4} \end{bmatrix}$$

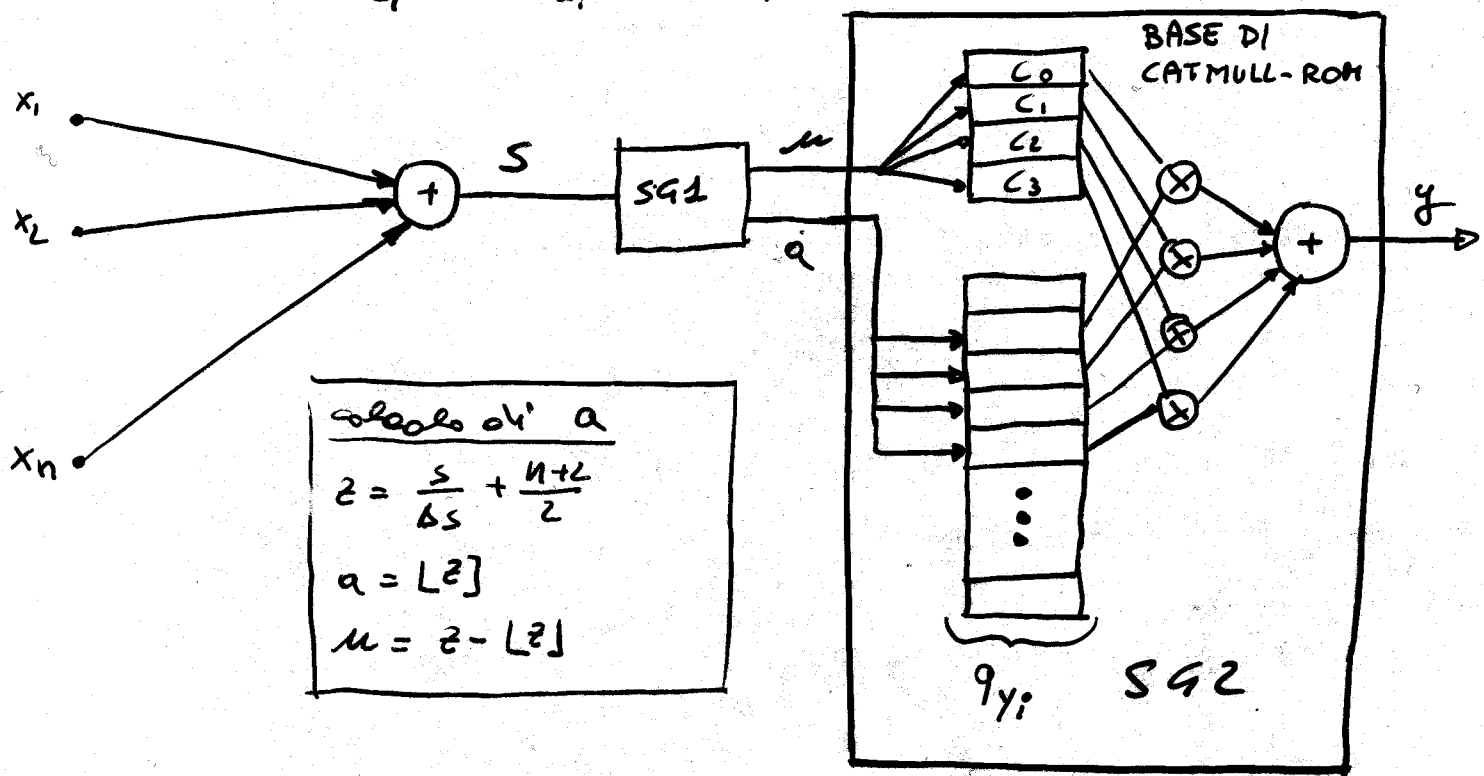
$$F_{y_i}(u) = [u^3 \ u^2 \ u \ 1] \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & \emptyset & 1 & \emptyset \\ \emptyset & 2 & \emptyset & \emptyset \end{bmatrix} \begin{bmatrix} q_{y1} \\ q_{y2} \\ q_{y3} \\ q_{y4} \end{bmatrix}$$



→ sviluppando le precedenti espressioni:

$$\begin{cases} F_{S_i}^{-1}(u) = s = c_1 u + c_0 \quad (\text{per hp lineare} \Rightarrow \text{invertibile}) \\ F_{Y_i}(u) = b_3 u^3 + b_2 u^2 + b_1 u + b_0 \end{cases}$$

$$y = F_{Y_i}(u) = \frac{b_3}{c_1^3} s^3 + \left( \frac{b_2}{c_1^2} - \frac{3b_3 c_0}{c_1^3} \right) s^2 + \left( \frac{3b_3 c_0^2}{c_1^3} - \frac{2b_2 c_0}{c_1^2} + \frac{b_1}{c_1} \right) s + \left( \frac{b_2 c_0^2}{c_1^2} - \frac{b_3 c_0^3}{c_1^3} - \frac{b_1 c_0}{c_1} + b_0 \right)$$



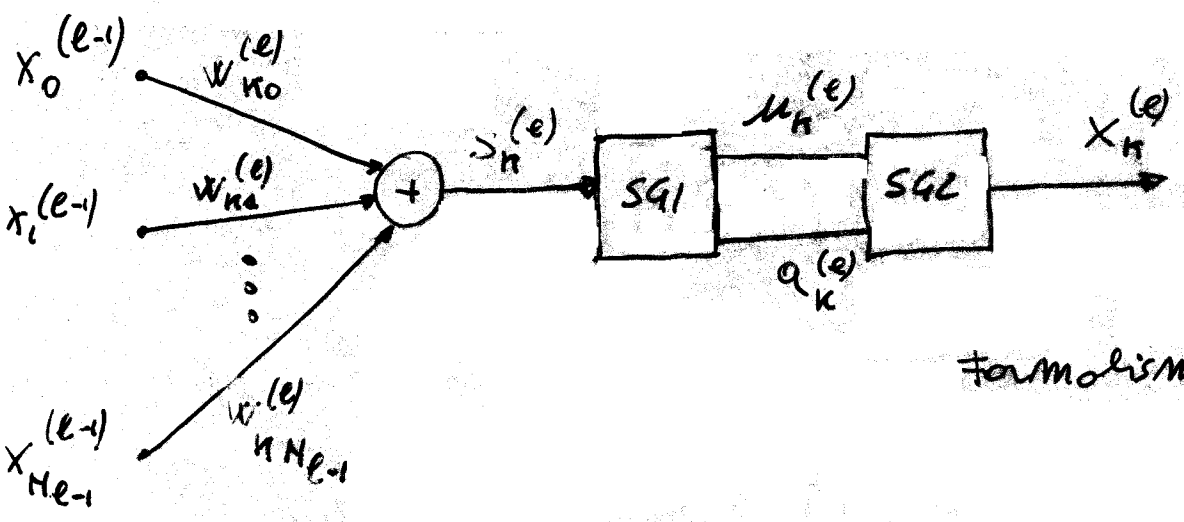
calcolo di a

$$z = \frac{s}{b_3} + \frac{u+2}{2}$$

$$a = \lfloor z \rfloor$$

$$u = z - \lfloor z \rfloor$$

# Algoritmo di apprendimento su $n$ neurone SG



Formalismo

$$\begin{cases} w_{kj}^{(l)} [t+1] = w_{kj}^{(l)} [t] + \Delta w_{kj}^{(l)} [t] \\ q_{k, (a_k^{(l)} + m)}^{(l)} [t+1] = q_{k, (a_k^{(l)} + m)}^{(l)} [t] + \Delta q_{k, (a_k^{(l)} + m)}^{(l)} [t] \end{cases}$$

$m = 0, \dots, 3$

## $\Delta w$

$$e_k^{(l)} = \begin{cases} (d_k - x_k^{(l)}) & l = M \\ \sum_{P=1}^{N_{l+1}} s_P^{(l+1)} w_{Pk}^{(l+1)} & l = M-1, \dots, 1 \end{cases}$$

$$s_k^{(l)} = e_k^{(l)} \left( \frac{\partial F_{k, a_k^{(l)}}^{(l)}(\mu)}{\partial \mu} \Big|_{\mu = \mu_k^{(l)}} \right) \frac{\Delta}{\Delta s}$$

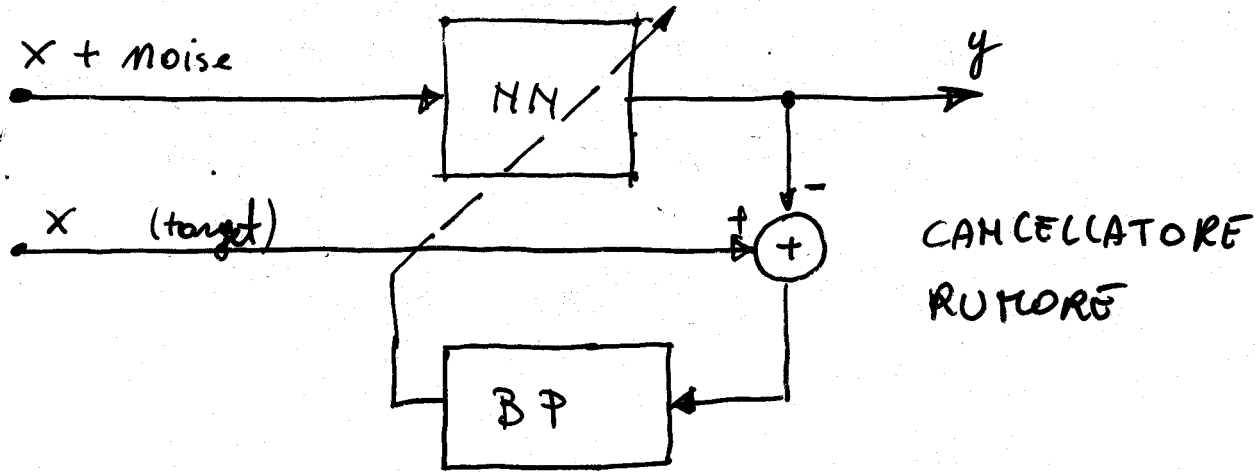
$$\Delta w_{kj}^{(l)} = \eta s_k^{(l)} x_j^{(l-1)}$$

## $\Delta q$

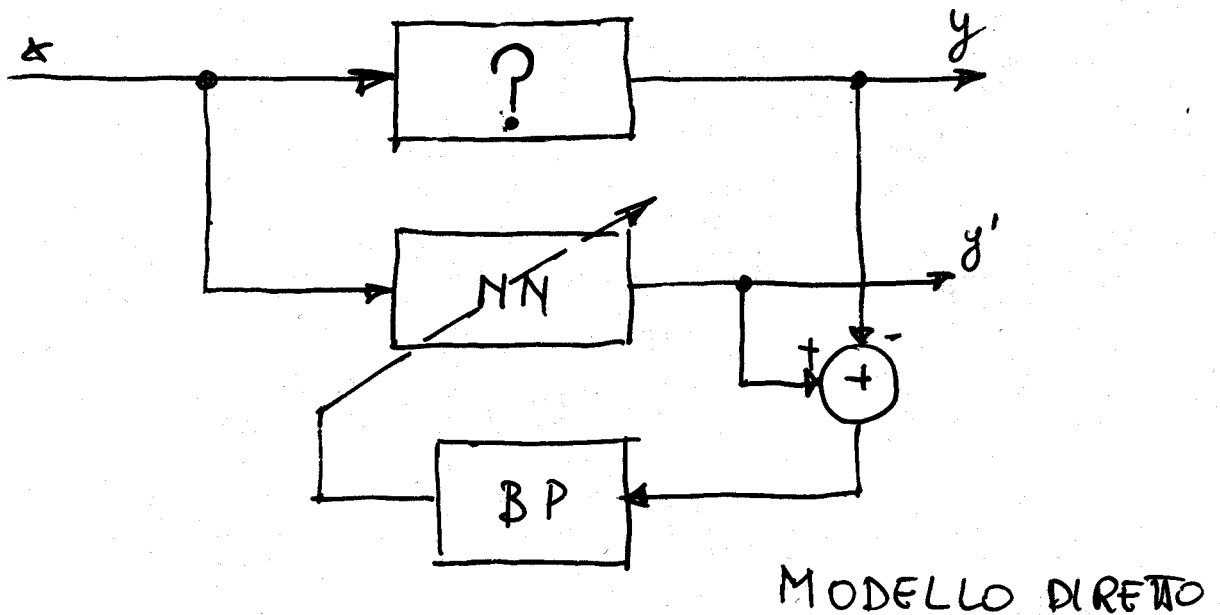
$$\Delta q_{k, (a_k^{(l)} + m)}^{(l)} = \mu_q e_k^{(l)} \left( \frac{\partial F_{k, a_k^{(l)}}^{(l)}(\mu)}{\partial q_{k, (a_k^{(l)} + m)}^{(l)}} \right) = \mu_q e_k^{(l)} c_{k, m}^{(l)}(\mu_k^{(l)})$$

# 10. APPLICAZIONI ANN PER S.P.

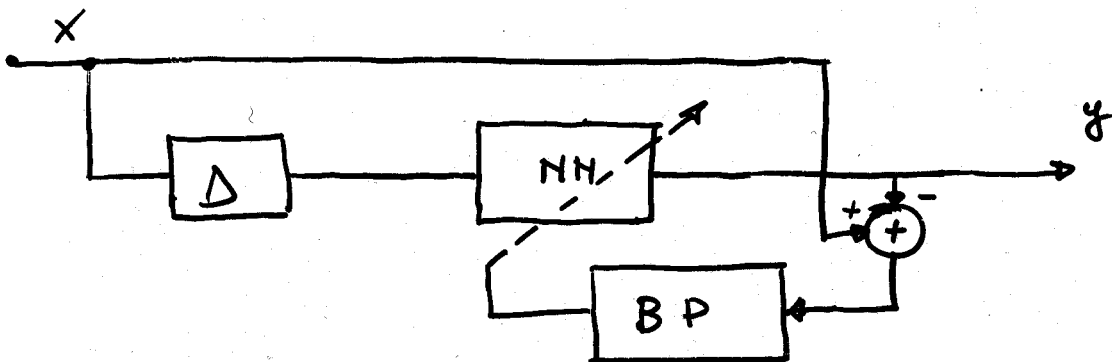
## • Filtraggio adattativo non lineare



## • Identificazione



## • Predizione



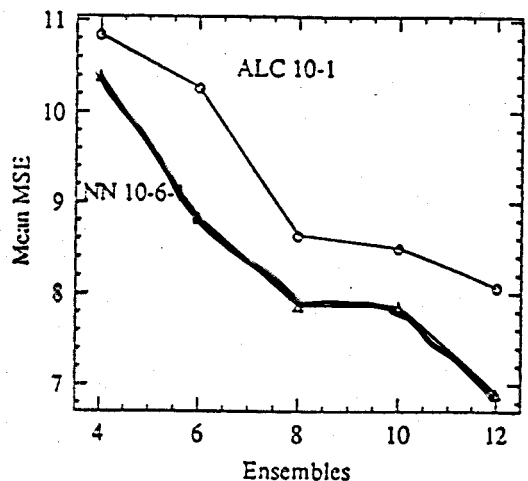
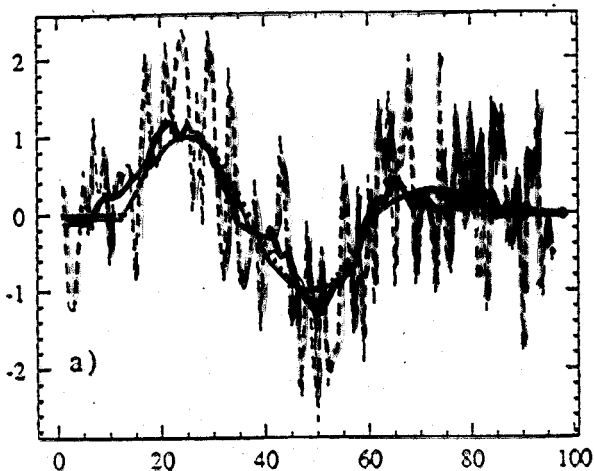
## APPLICAZIONI

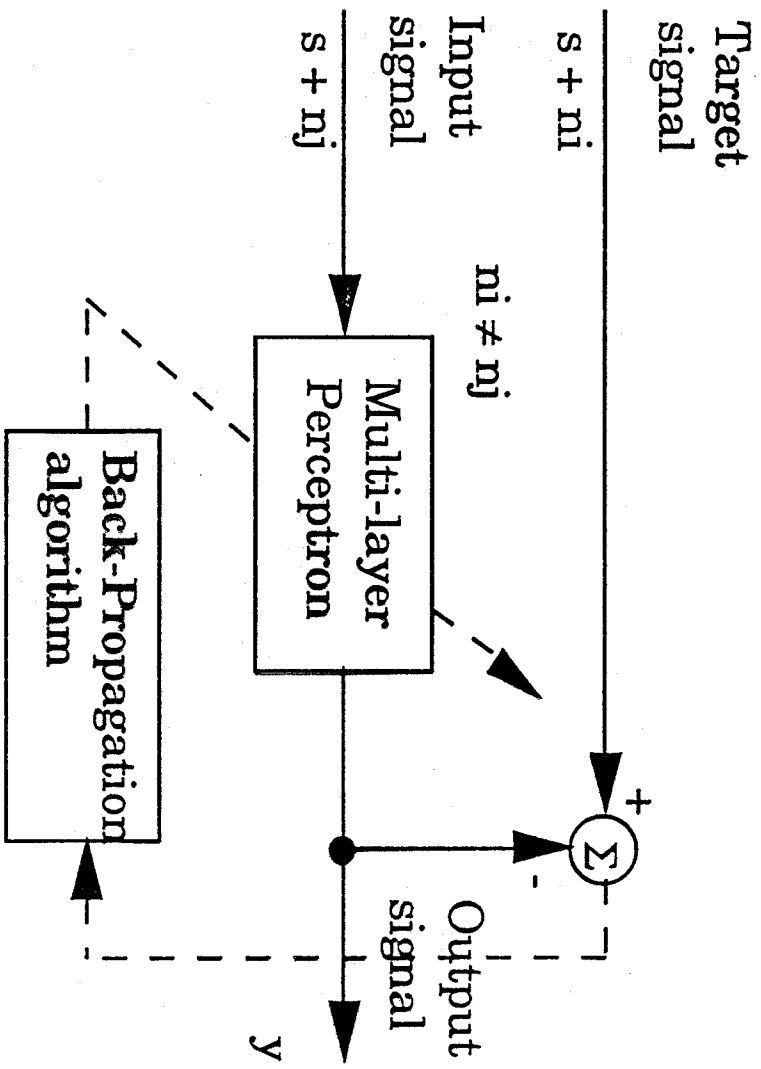
utilizzazione delle reti MLP in problemi applicativi quali la predizione di processi non lineari, il filtraggio adattativo non lineare, la quantizzazione vettoriale, etc.;

### Esempio: filtraggio di potenziali evocati

Problema: il potenziale cerebrale evocato (EP), un segnale elettroencefalografico ottenuto in risposta a semplici stimoli, è immerso in un rumore di fondo anche non stazionario con rapporto S/N inferiore allo zero.

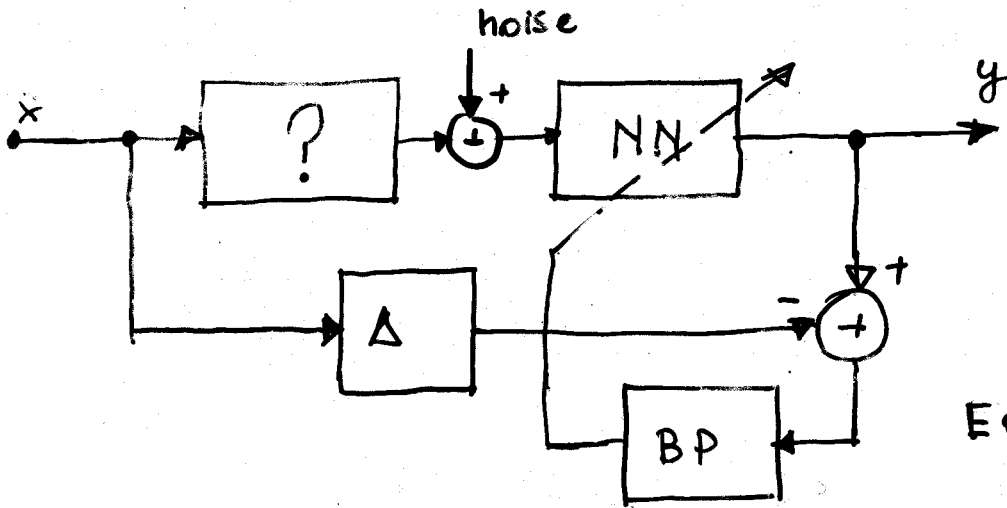
Contributo: filtraggio adattativo non lineare mediante reti MLP con particolari topologie e con algoritmo di apprendimento modificato.



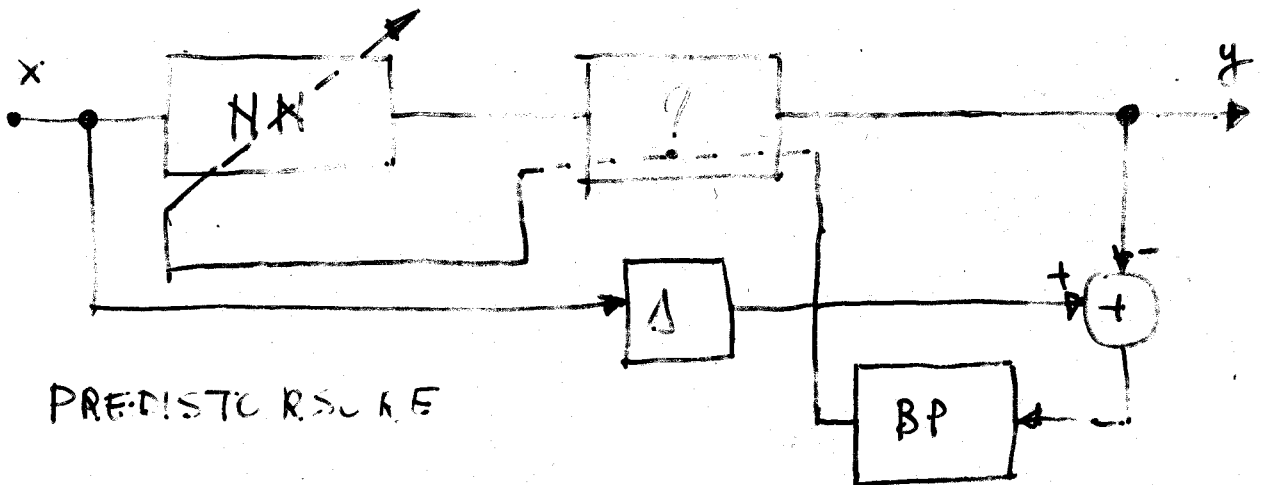


**Scheme of the proposed adaptive architecture for FIP processing.**

# Modello inverso



EQUALIZZATORE



PREDISTORSORE

Nel caso del predistorsore per la back propagation  
occorre avere un stima anche molto approssi-  
mata del sistema incognito

# Predizione di serie temporale caotica

• serie di Mackey-Glass

$$x^o(t) = \frac{a x(t-\tau)}{1 + x^c(t-\tau)} - b x(t)$$

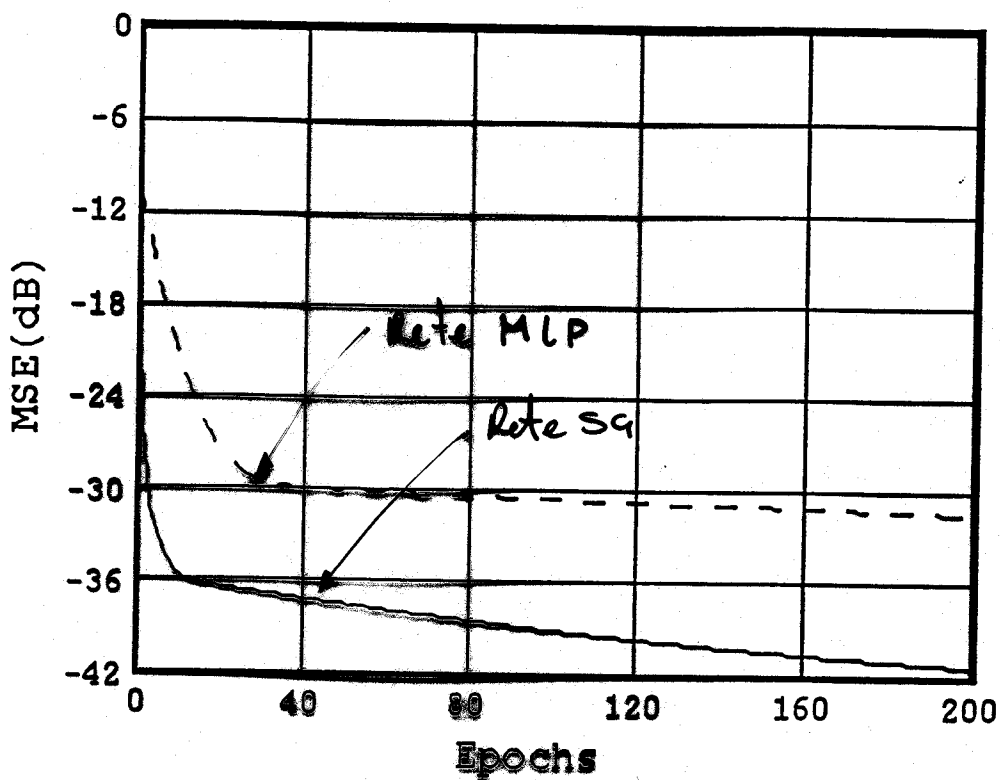
$$a = 0.2$$

$$b = 0.1$$

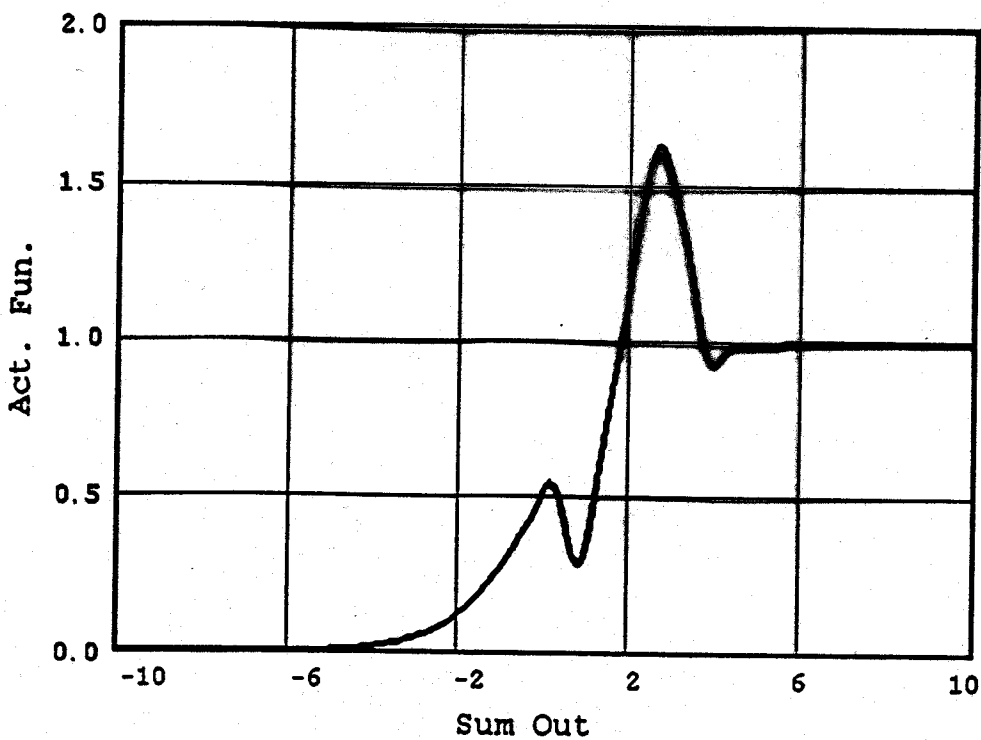
$$c = 10$$

$$\tau = 80$$

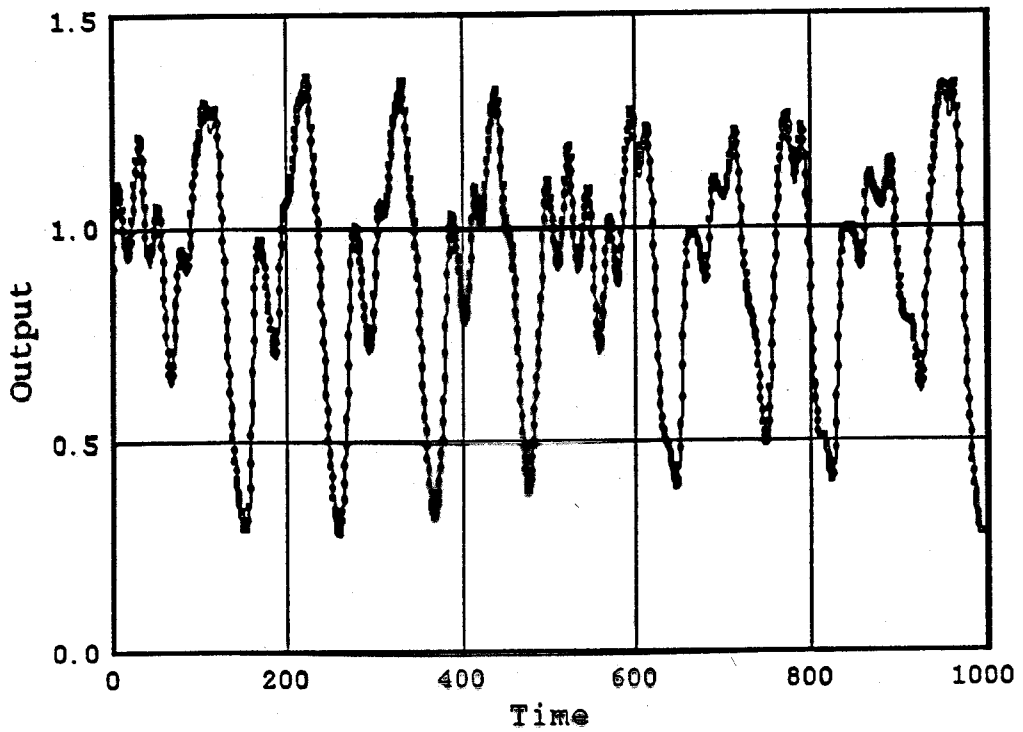
Reti:  $\left\{ \begin{array}{l} \text{SG-NET } 6 \text{ ingressi;} \\ \text{MLP } 2-1 \text{ (stesso n° per. lib.)} \end{array} \right. \eta = 0.1$



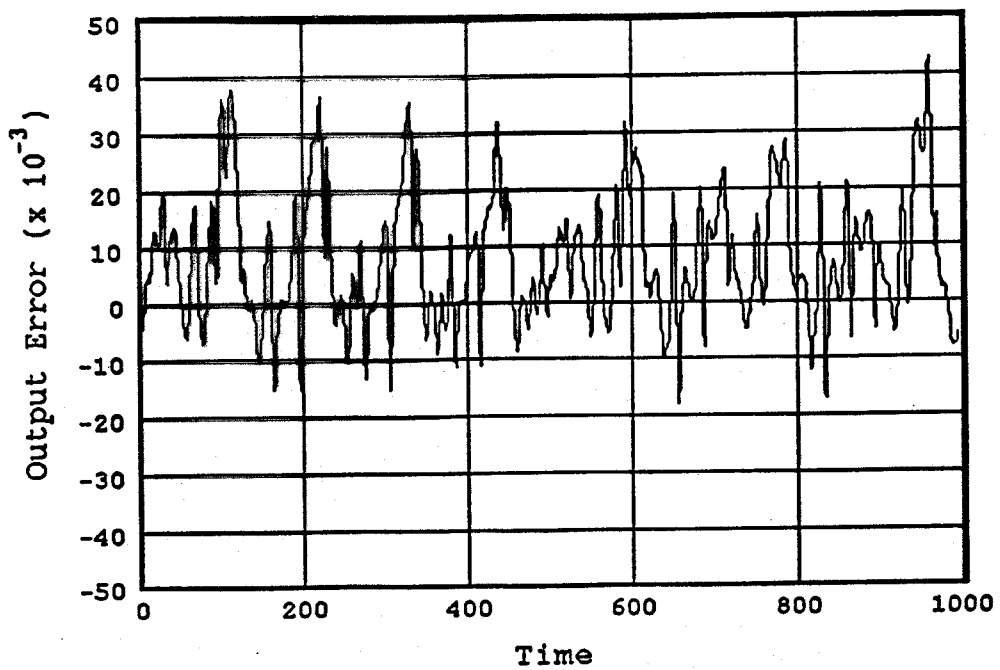
*andamento dell'MSE per le due reti per la serie Mackey-Glass*



*funzione SG per la serie Mackey-Glass*

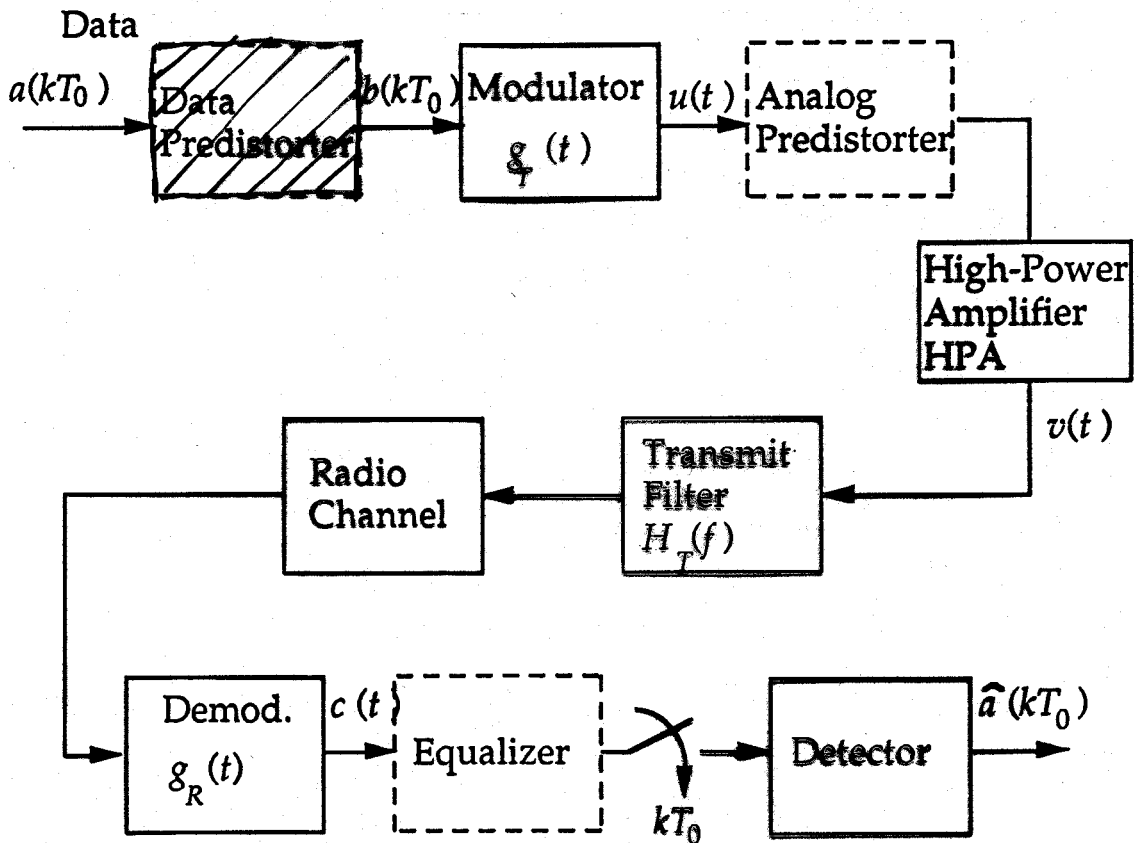


*output della rete SG nel test Mackey-Glass*



*scarto tra serie vera e serie predetta*

# BLOCK DIAGRAM OF THE DIGITAL RADIO LINK (BASEBAND MODEL)

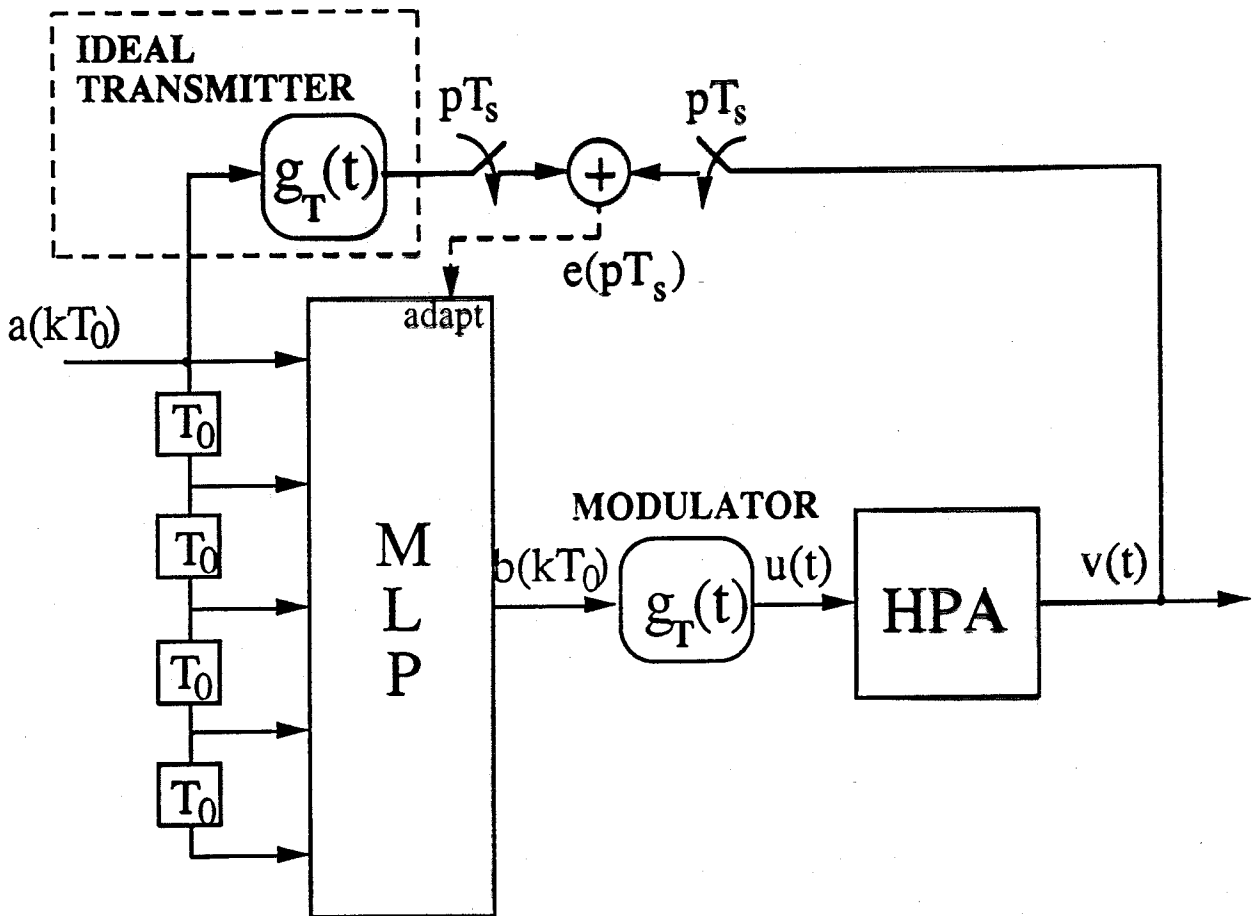


**Two approaches for compensating the non-linear effects of the HPA:**

**A) NON-LINEAR EQUALIZATION**

**B) PREDISTORTION AT THE TRANSMITTER**

## A NEURAL NETWORK DATA PREDISTORTION APPROACH

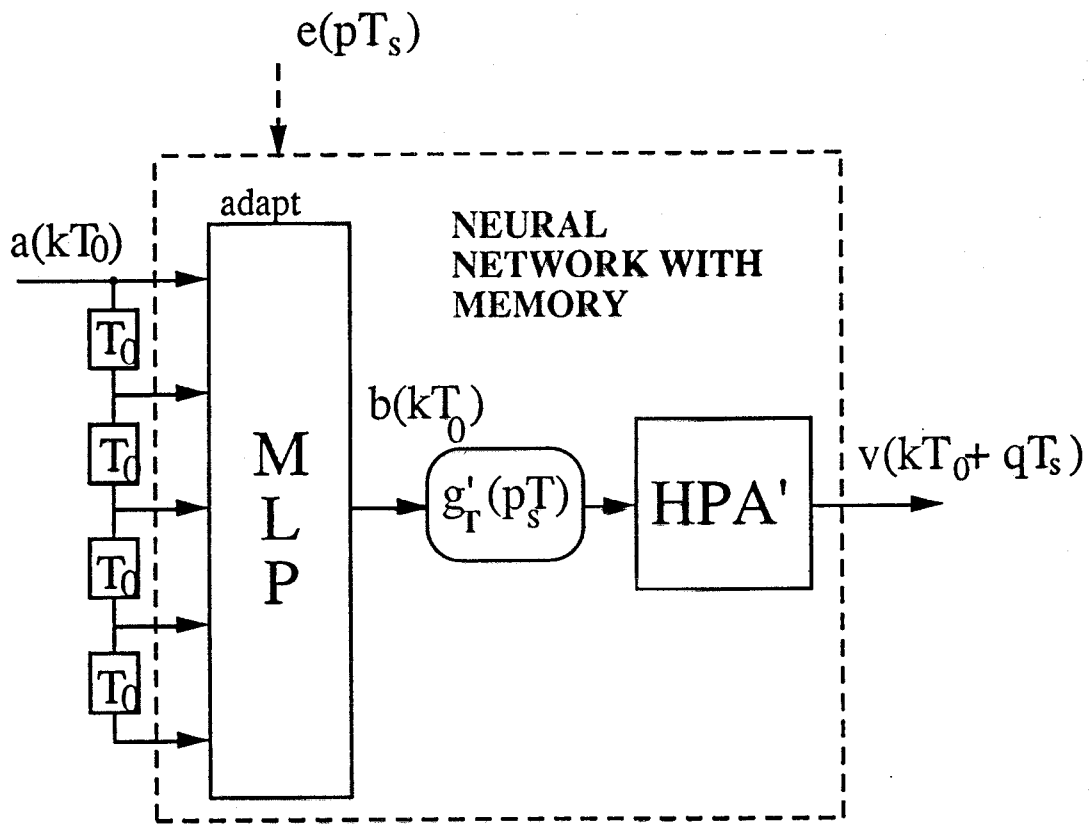


- \* The output can be sampled with  $T_s$  less than  $T_0$ .
- \* A greater control of the spectrum of  $v(t)$  can be exercised.

(The Multilayer Perceptron can be realized by a RAM)

## ADAPTATION SCHEME

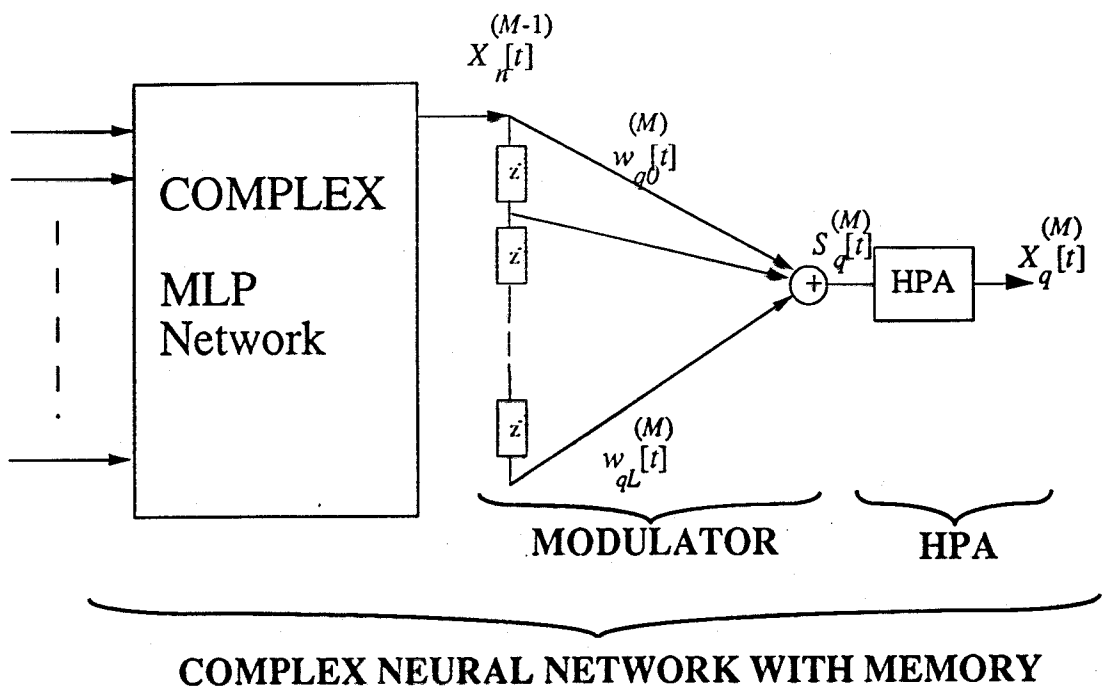
- \* The analog signal  $u(t)$  and  $v(t)$  are replaced by oversampled discrete signals with period  $T_s$ .
- \* The entire system may be modeled as a multirate neural network with memory.



- \* An extension of the backpropagation algorithm can be used to determine the predistorter parameters.
- \* Only rough models of the modulator and HPA are needed; in our case we assume:  
 $g'(t) = g_T(t)$  and  $HPA' = HPA$

# NEURAL NETWORKS WITH FIR SYNAPSES

- \* Data sequences are complex.
- \* The proposed NN model has a dynamic behavior because it includes neurons with FIR synapses
- \* A novel algorithm (FIR Dynamics BP) has been developed in order to train the network



# THE COMPLEX ACTIVATION FUNCTION

$$\text{SGM}(Z) = \text{sgm}(Z_R) + j \text{sgm}(Z_I)$$

where:

$$\text{sgm}[a] = \frac{2G}{1 + e^{-a}} - G$$

- \* Bounded everywhere, smooth and very similar to the real-valued sigmoidal function;
- \* Collapse to the classical sigmoid, if restricted to the real axis.

