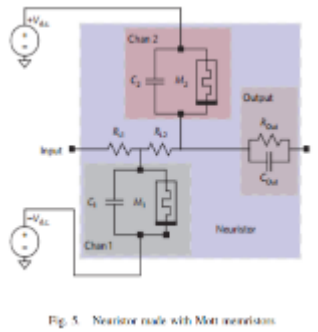
Notes on Neural Networks

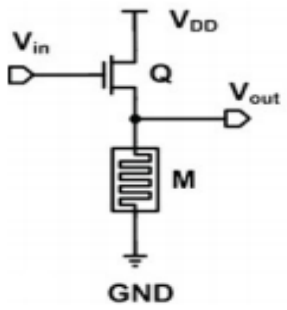
An approach
+ Breadboard
- Blackboard

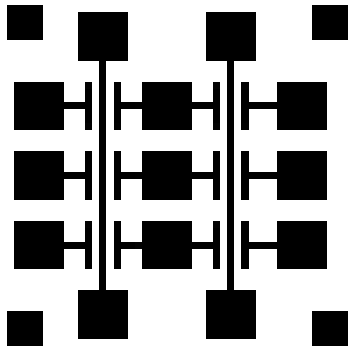
Some sparse nouns from E.B.Carne, 1965,
«Artificial Intelligence Techniques», MacMillan/Spartan:

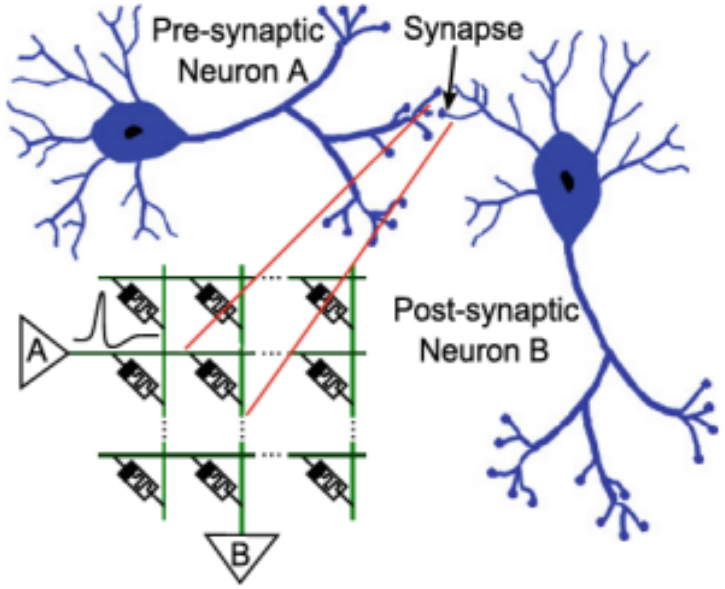
Neuristor, electrology, bionics, active_intellect logical_deduction, trial_and_error, intuition, inference representativity, ontology, knowledges, mindedness, formalism, model'ize.

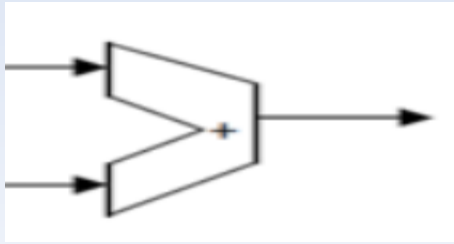
Resistive communications based on neuristors
David Alejandro Trejo Pizzo https://arxiv.org/abs/1705.03008











"FIELD PROGRAMMABLE DSP ARRAYS" –
Signal & Image Processing : An International Journal (SIPIJ) Vol.4, No.2, April 2013
DOI : 10.5121/sipij.2013.4204

'Filter Functions' (FIR, IIR etc.) and 'Linear Transforms' (DFT, FFT, DCT, DWT etc.).

$$Y = \sum_{j=1}^{B-1} [\sum_{i=1}^{N} x_{ij} a_i] 2^{-j} + \sum_{i=1}^{N} (-x_{i0}) a_i$$

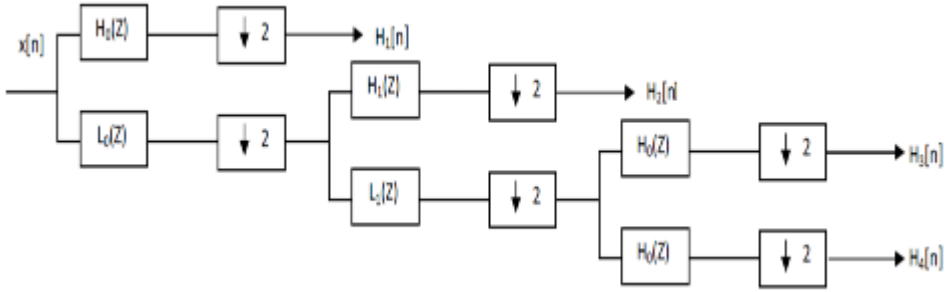
Finite Impulse Response Filter

$$y[n] = \sum_{k=0}^{L-1} x[k] \, c[n-k]$$

Infinite Impulse Response Filter

$$y[n] = \sum_{i=0}^{L-1} a[l] \, x[n-l] + \sum_{m=1}^{L-1} b[m] \, y[n-m]$$

Discrete Wavelet Transform



$$W_L(n, j) = \sum_m W_L(m, j - 1)\, h_0(m - 2n)$$

$$W_H(n, j) = \sum_m W_L(m, j - 1)\, h_1(m - 2n)$$

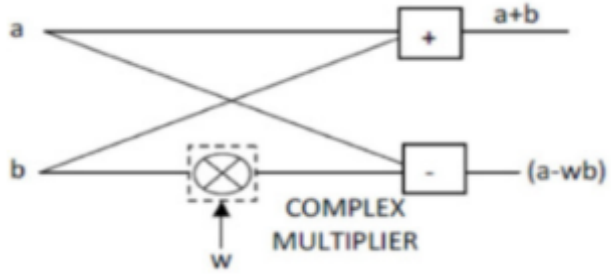
Fast Fourier Transform

$$R + jI = (a + ib)(\cos\theta + i \sin\theta)$$

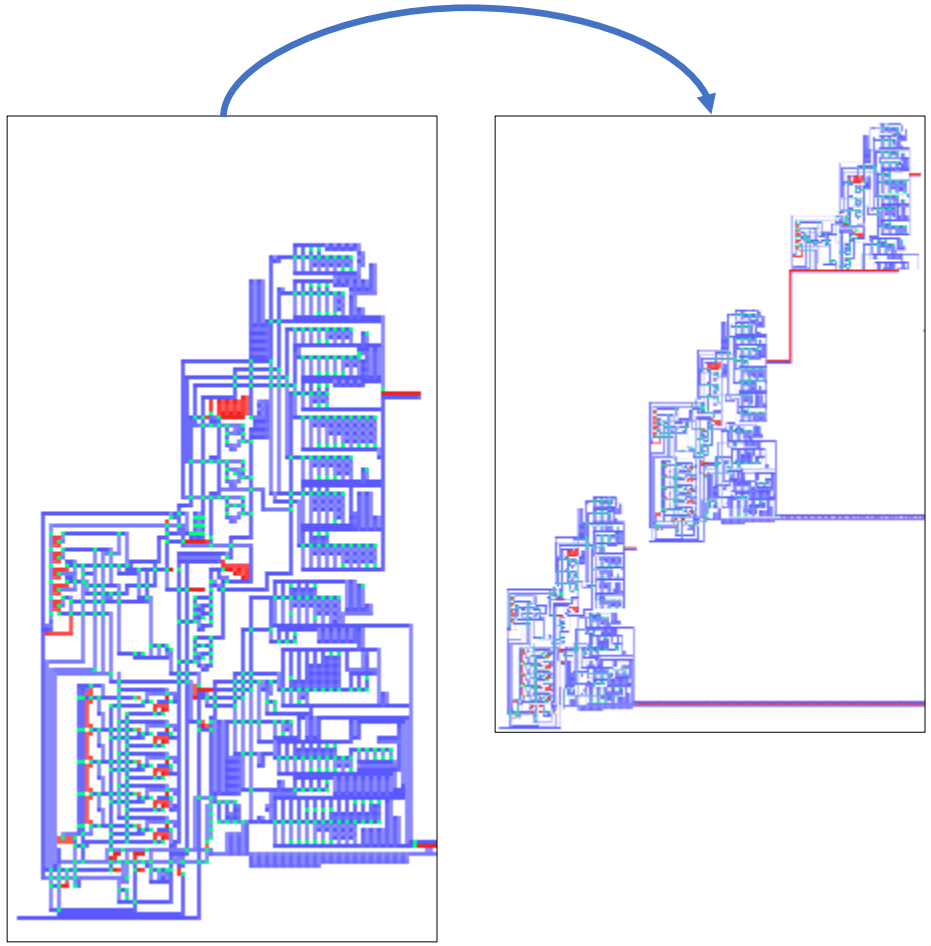
$$R = (\cos\theta - \sin\theta)\, b + \cos\theta\, (a - b)$$

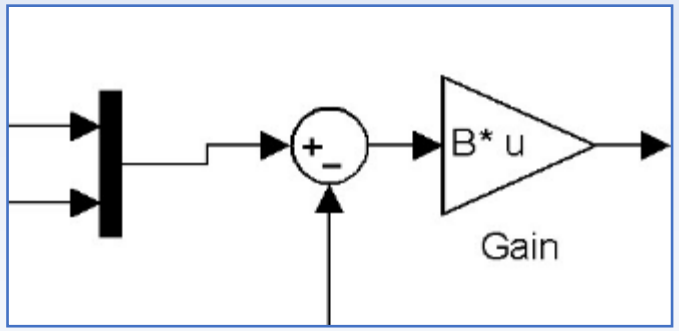
$$I = (\cos\theta + \sin\theta)\, a - \cos\theta\, (a - b)$$

$$X[k] = \sum_{n=0} x[n]\, e^{-(j2\pi kn / N)}$$



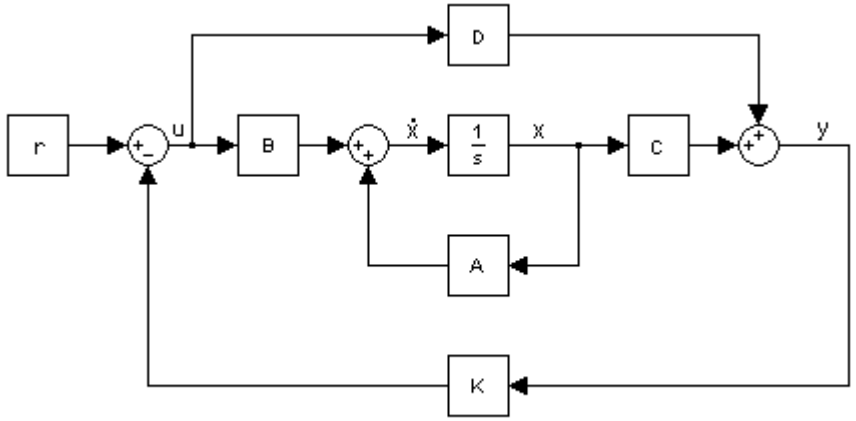
Costruttore universale di von Neumann, 1966 Arthur W. Burks





State-space representation

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$$



Mapping: $S(a_i,i_i)$

S means that "Entity a_i transport information i_i"

Direct Mapping: $S_i(a_i,i_i) > S_j(a_j,i_j)$ $S_i > S_j$

S_i transport a "precedence" over the m S_j

(has a direction)

Comparison: $S_i(a_i,i_i) = S_j(a_j,i_j)$ $S_i = S_j$

The information of S_i and S_j are comparable.

Recognition: Information maps to a reference model

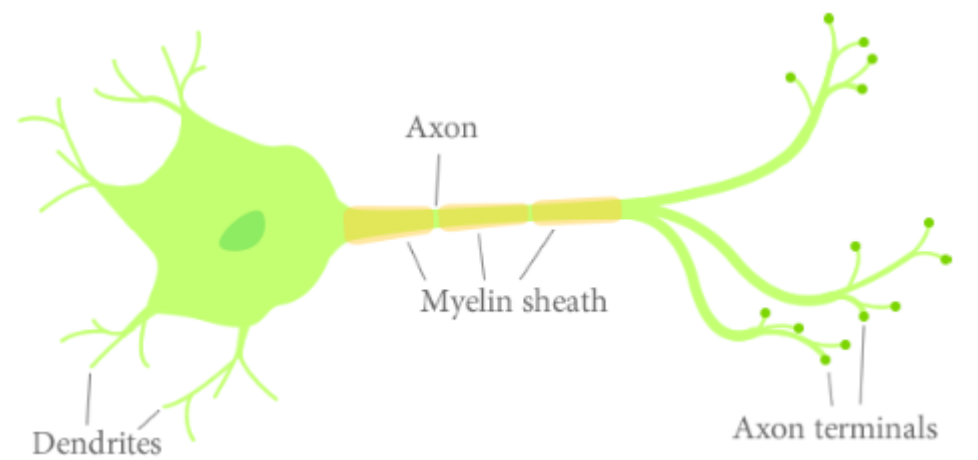
Problem: A Mapping to a "not-Yet-Known" model

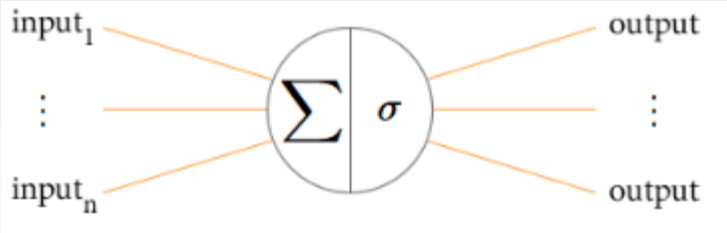
Decision: A Problem promoted by a "precedence" law

Prediction: A "not-yet-mapped" model promoted as reference by a "precedence" law

Liu, Notes on real-time Decisional Trees, 2019, in prep.

http://scs.ryerson.ca/~aharley/neural-networks/

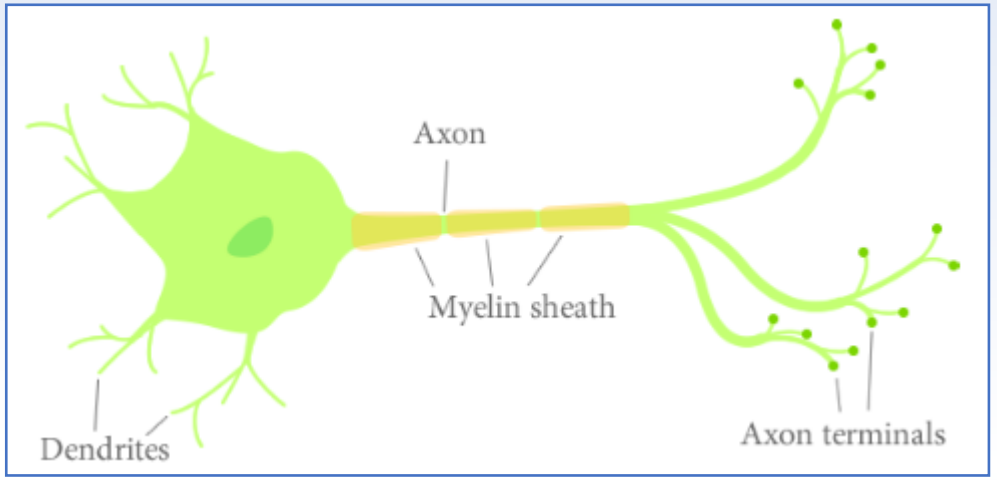


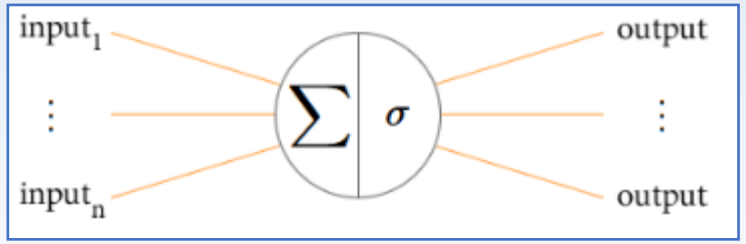


$$\sigma(x) = \begin{cases} 0, & \text{if } x < \text{threshold}, \\ 1, & \text{if } x >= \text{threshold}, \end{cases}$$

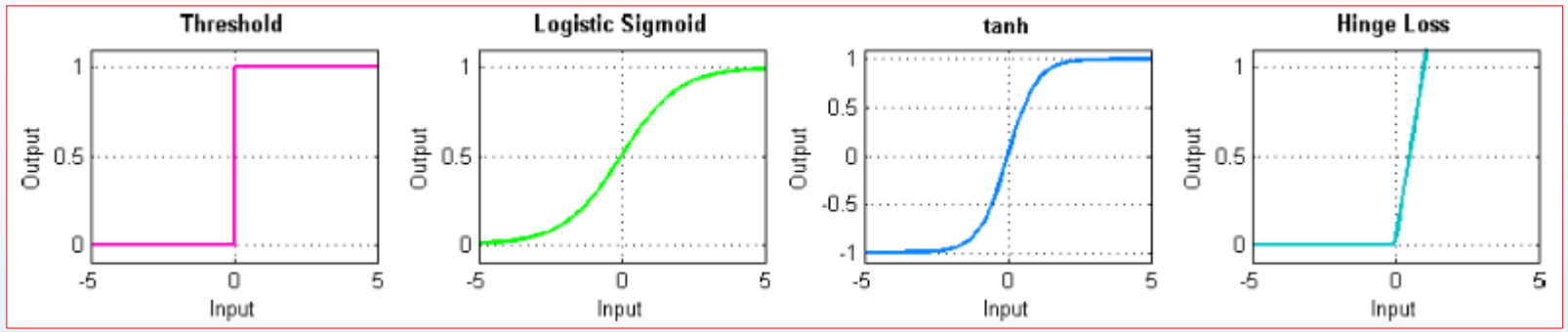
$$f\left(\boxed{0}\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad f\left(\boxed{1}\right) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \ldots, \quad f\left(\boxed{9}\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

http://scs.ryerson.ca/~aharley/neural-networks/





$$\sigma(x) = \begin{cases} 0, & \text{if } x < \text{threshold}, \\ 1, & \text{if } x >= \text{threshold}, \end{cases}$$

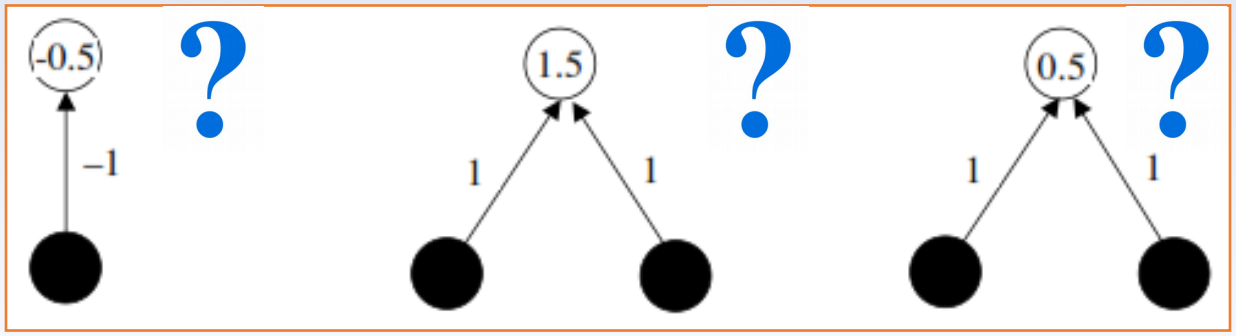


$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

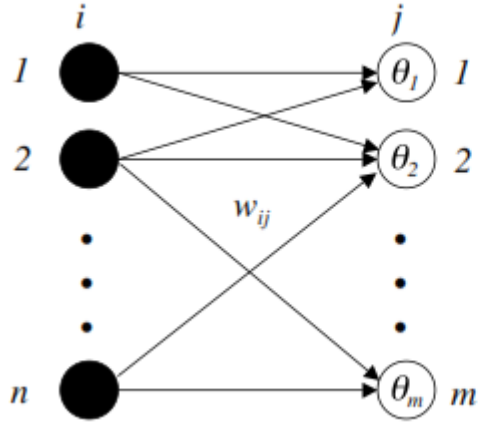
$$O = \sigma(\phi_1 w_1 + \ldots + \phi_n w_n + b).$$

McCulloch-Pitts Neuron

$$out_i = step(\sum_{k=1}^{n} in_{ki} - \theta_i)$$



The Perceptron



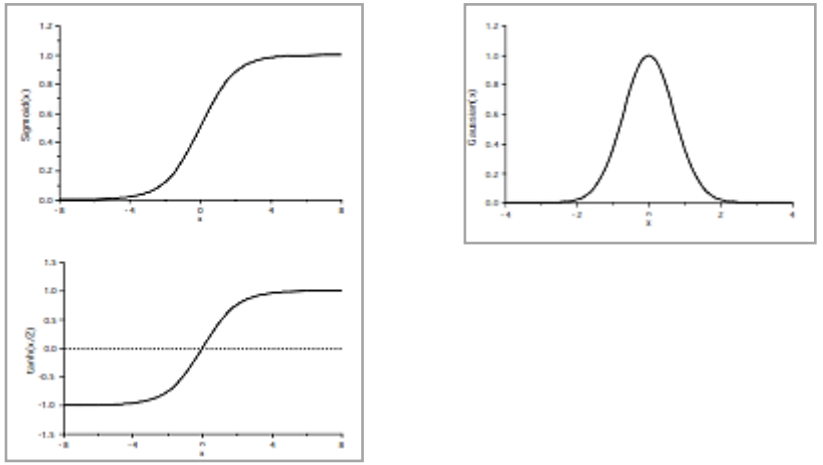
$$out_j = step(\sum_{i=1}^{n} in_i w_{ij} - \theta_j)$$

XOR ?

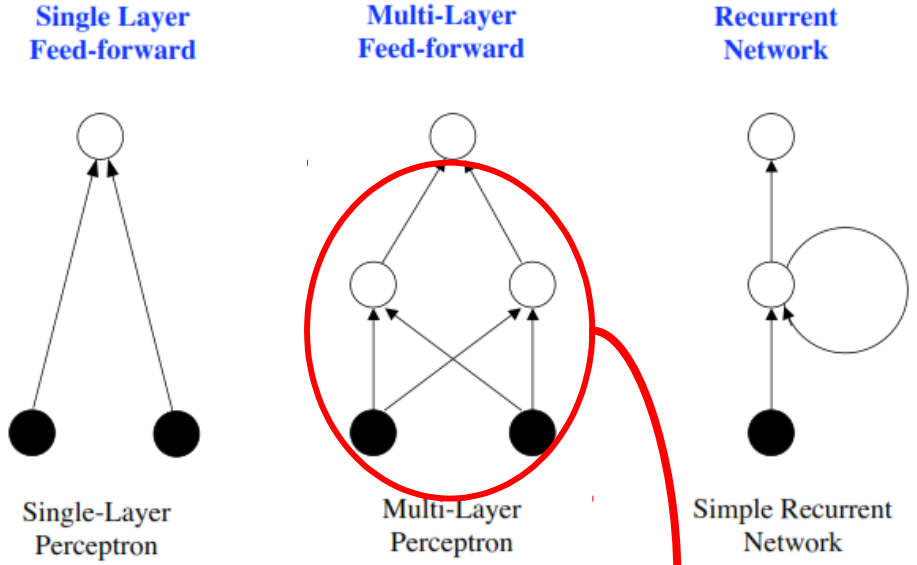
non-linearly separable.

© John A. Bullinaria, 2015

Activation/Transfer Function



Examples of Network Architecture Types



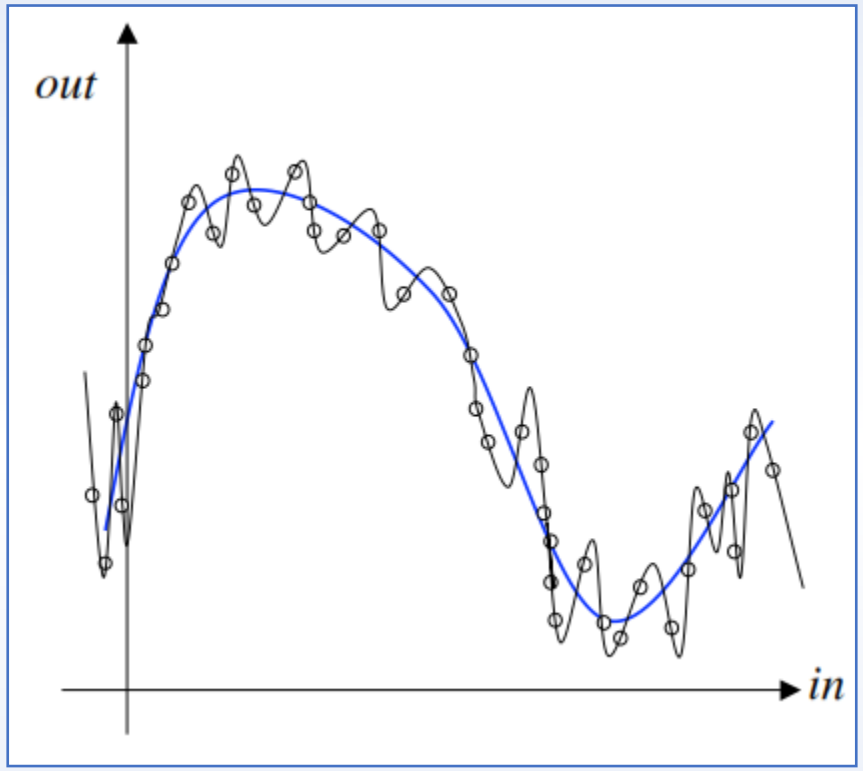
Decision Hyperplanes and Linear Separability

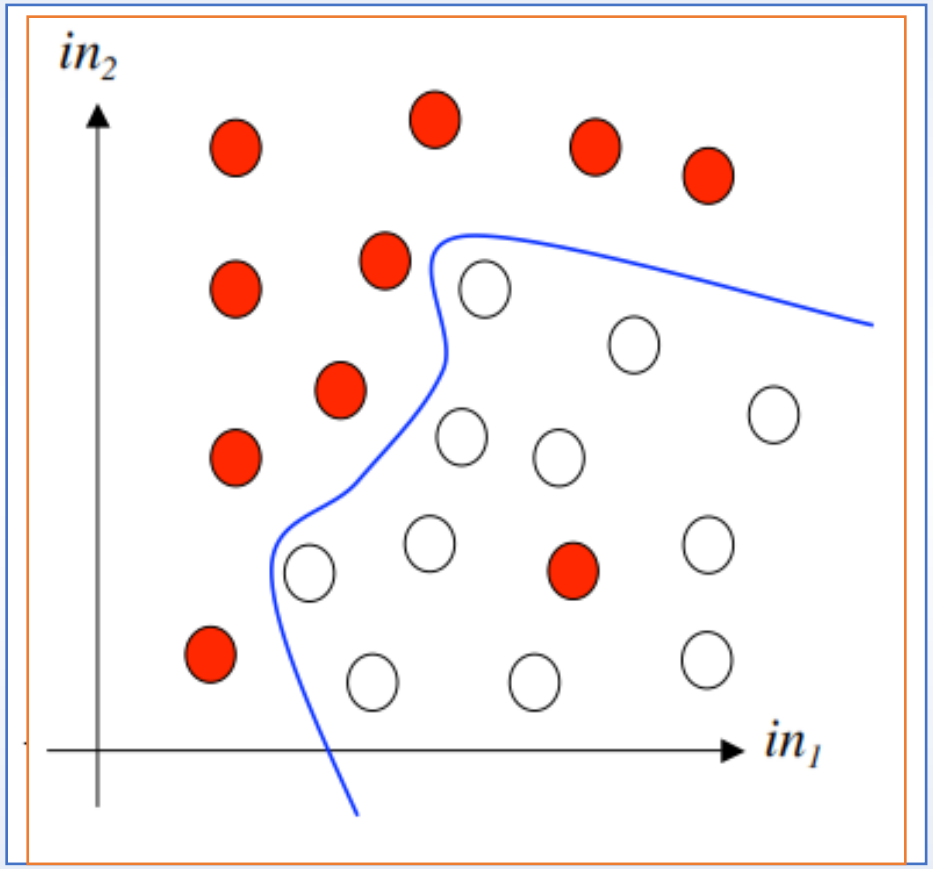
$$w_1 in_1 + w_2 in_2 + \ldots + w_n in_n - \theta = 0$$

network weights at time t are $w_{ij}(t)$ $\quad w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$

© John A. Bullinaria, 2015

Generalization in Function Approximation





© John A. Bullinaria, 2015

Basic Neuron Equation

$$out_j = f\left(\sum_{i=1}^{n} w_{ji} in_i - \theta_j\right)$$

Sigmoid/Logistic Activation Function

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$

Sum Squared Error and Cross Entropy Cost Functions

$$E_{SSE}(\{w_{kl}\}) = \tfrac{1}{2}\sum_p \sum_j \left(targ_j^p - out_j(in_i^p)\right)^2$$

$$E_{CE}(\{w_{ikl}\}) = -\sum_p \sum_j \left[targ_j^p .\log\left(out_j\left(in_i^p\right)\right) + \left(1 - targ_j^p\right).\log\left(1 - out_j\left(in_i^p\right)\right)\right]$$

Gradient Descent – General Weight Update Equation

$$\Delta w_{kl} = -\eta \frac{\partial E(\{w_{ij}\})}{\partial w_{kl}}$$

Gradient Descent Updates for Single Layer Perceptron

$$\Delta w_{kl} = \eta \sum_p \left(targ_l - out_l\right).in_k$$

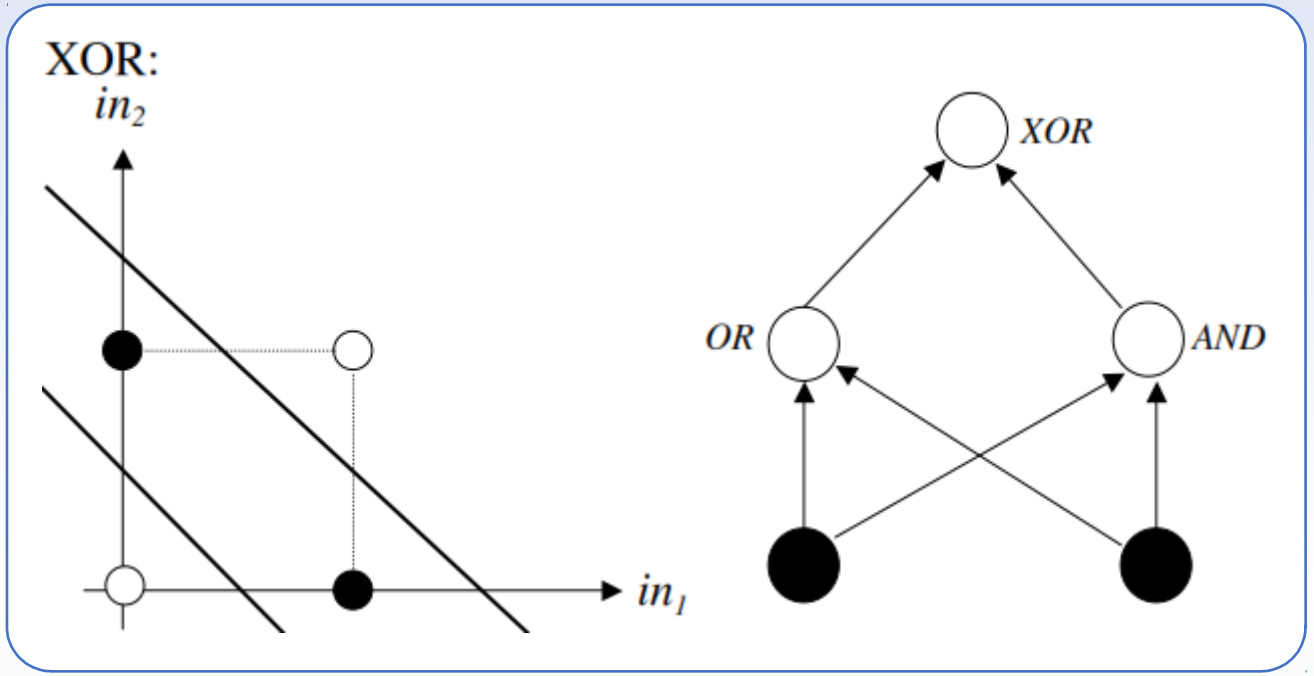
Back-propagation with Momentum

$$delta_k^{(N)} = \left(targ_k - out_k^{(N)}\right)$$

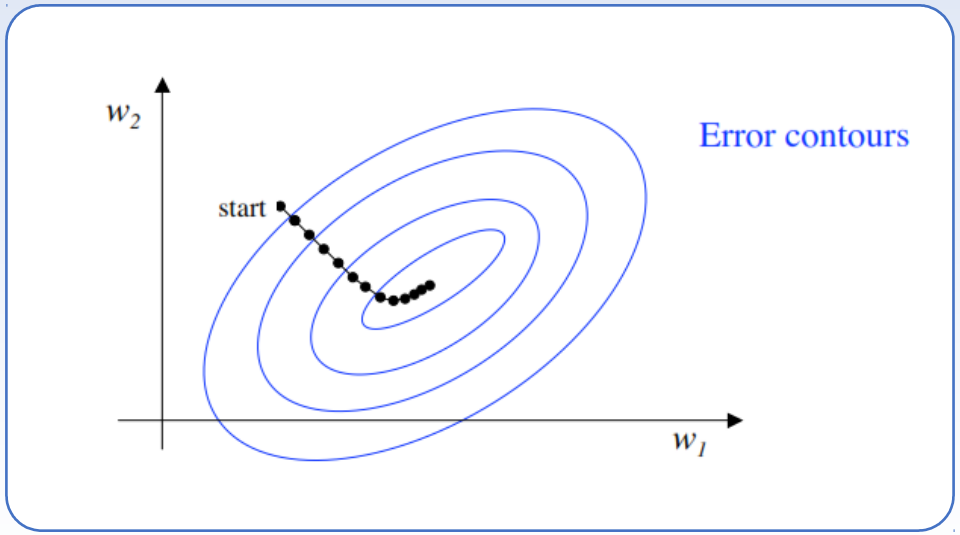
$$delta_k^{(n)} = \left(\sum_k delta_k^{(n+1)}.w_{lk}^{(n+1)}\right).f'\left(\sum_j out_j^{(n-1)} w_{jk}^{(n)}\right)$$

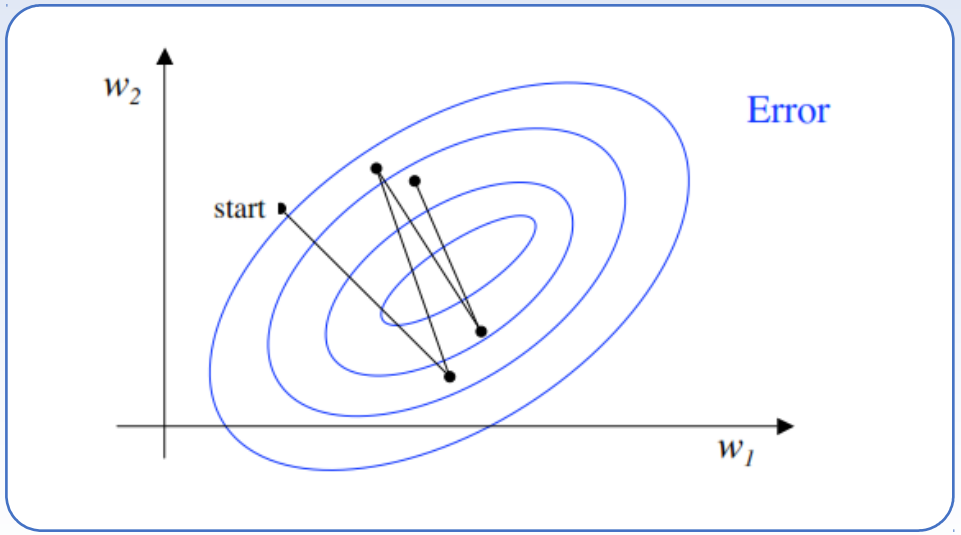
$$\Delta w_{kl}^{(n)}(t) = \eta \sum_p delta_l^{(n)}(t).out_k^{(n-1)}(t) + \alpha.\Delta w_{kl}^{(n)}(t-1)$$

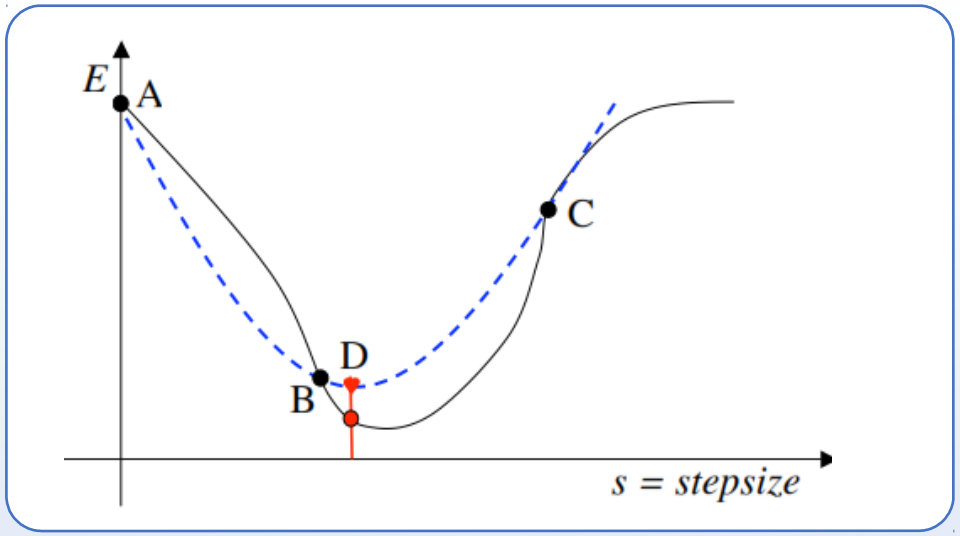
© John A. Bullinaria, 2015



© John A. Bullinaria, 2015



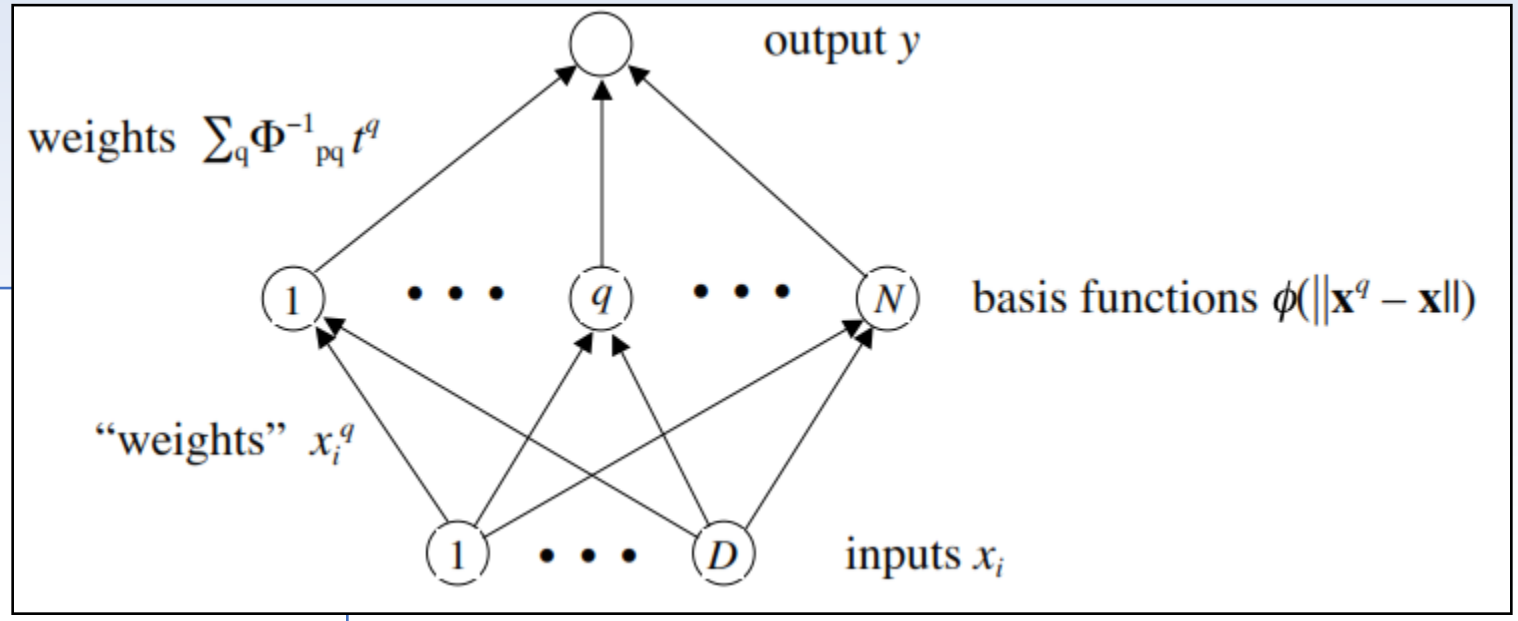


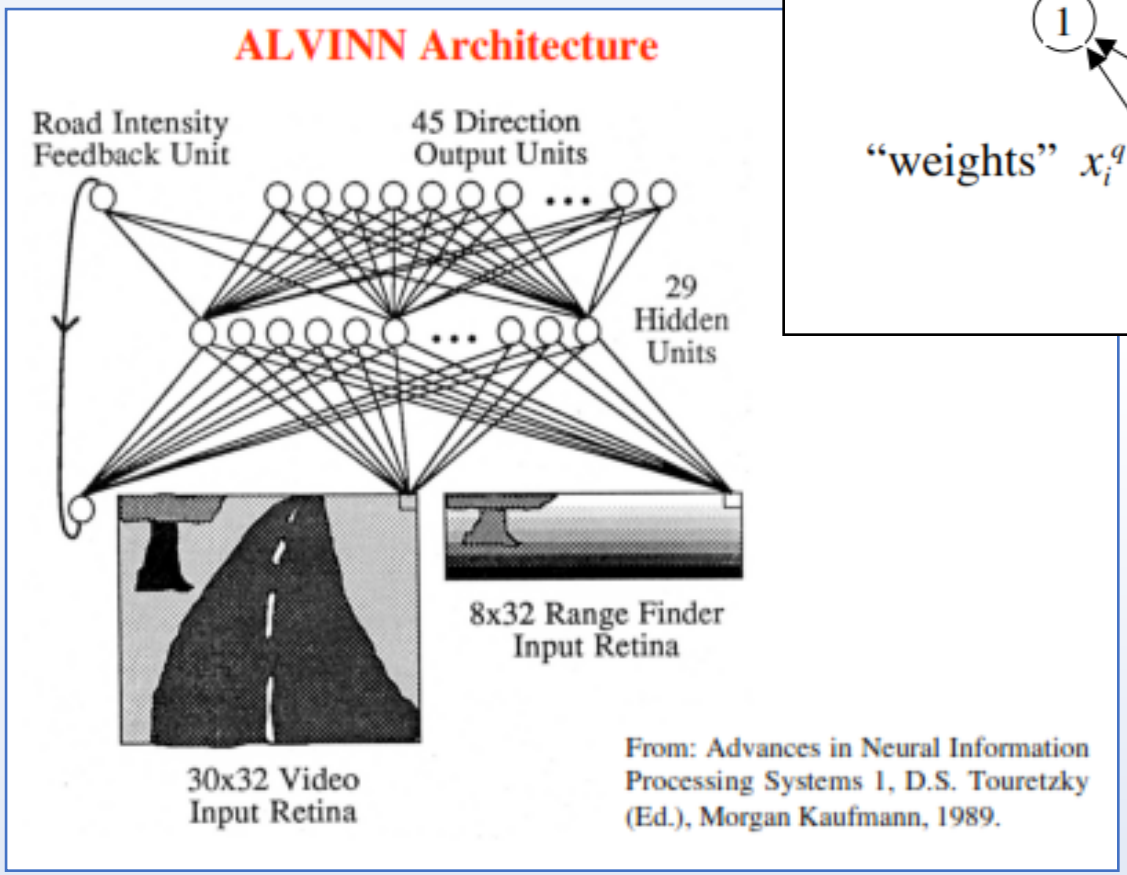


© John A. Bullinaria, 2015

Heurystic Greedy Bettor

Congettura 1:
Qualunque funzione differenziabile con spettro di potenza finito,
 ammette forma Taylor o .
Congettura 2:
Delle precedenti funzioni la ricerca degli ottimi marginali in domini chiusi
 è un problema lineare.
Tesi :
~~Il delta passo conviene che sia verso il baricentro alle differenze pesate tramite entropia~~





Radial Basis Function Networks

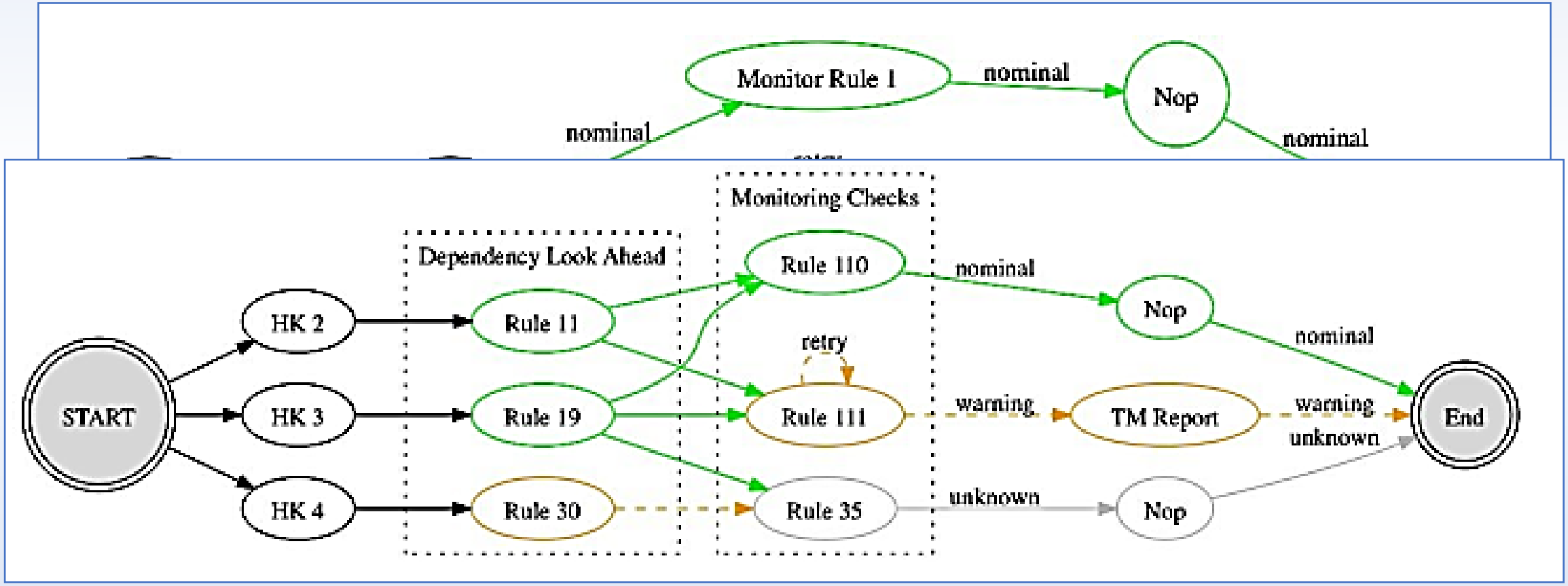
© John A. Bullinaria, 2015

Gaussian Radial

$$f(\mathbf{x}) = \sum_{p=1}^{N} w_p \phi_p(\mathbf{x}) = \sum_{p=1}^{N} w_p \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}^p\|^2}{2\sigma^2}\right)$$

Herschel-SPIRE satellite instrument: configurable On-Board Software for autonomous and real time operation

Housekeeping Parameters Collection and Monitoring



Scige J. Liu
Proc. of SPIE Vol. 8442 844245-1

Herschel-SPIRE satellite instrument: configurable On-Board Software for autonomous and real time operation

Housekeeping Parameters Collection a

Soft and Hard limits for analog parameters
Fail_Low < Warn_Low < Normal < Warn_High < Fail_High

Fail_Low Limit	Warn_Low Limit	Normal Range	Warn_High Limit	Fail_High Limit
Failure	Warning	Normal	Warning	Failure

Table 13-5 – Monitored HK Value Ranges

Note:
The WARNING/FAILURE state is triggered when the value is equal to limit:
{Limit} <= {Value} <= {Limit}
The NORMAL state is triggered when the value is inside the boundary limits:
{Limit} < {Value} < {Limit}

Analog Parameter

Fail_High_Limit
Warn_High_Limit
Warn_Low_Limit
Fail_Low_Limit
Action_NW
Action_WN
Action_NF
Action_FN
Action_WF
Action_FW
Reserved
Reserved

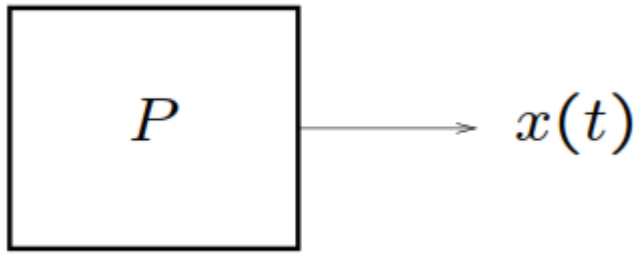
Reserved
N_Dep
Dep 1
...
Dep N (up to N=16)

Digital Parameter

N_Fail_Values	
Fail_Val2	Fail_Val1
Fail_Val4	Fail_Val3
Fail_Val6	Fail_Val5
Fail_Val8	Fail_Val7
Fail_Val10	Fail_Val9
Fail_Val12	Fail_Val11
Fail_Val14	Fail_Val13
Fail_Val16	Fail_Val15
Action_NF	
Action_FN	
Reserved	

Scige J. Liu, Proc. of SPIE Vol. 8442 844245-1

by Dave Touretzky and Kornel Laskowski



Discrete Phenomena

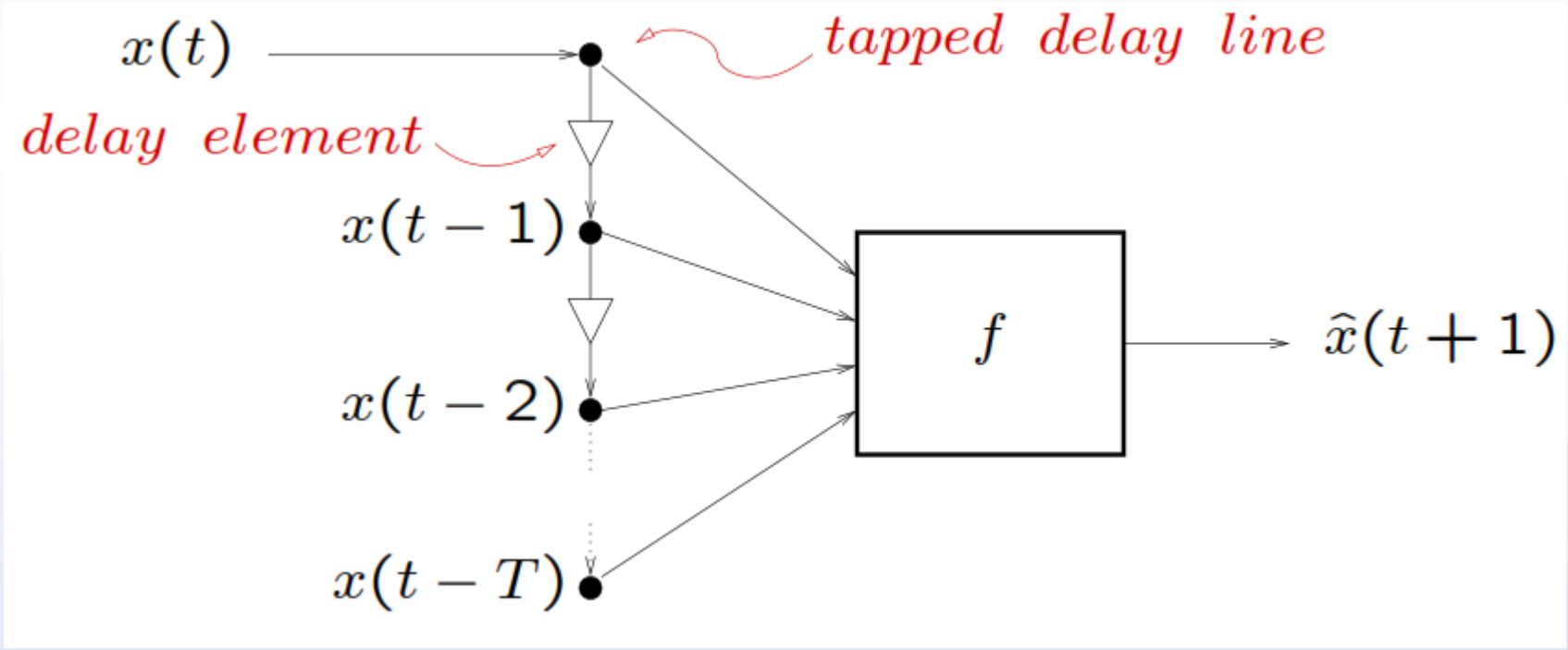
$$\{\ x(t_0),\ x(t_1),\ \cdots\ x(t_{i-1}),\ x(t_i),\ x(t_{i+1}),\ \cdots\ \}$$

Continuous Phenomena

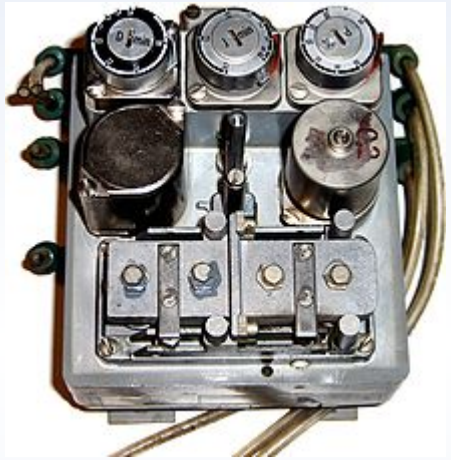
$$\{x[t]\} = \{x(0), x(\Delta t), x(2\Delta t), x(3\Delta t), \cdots\}$$

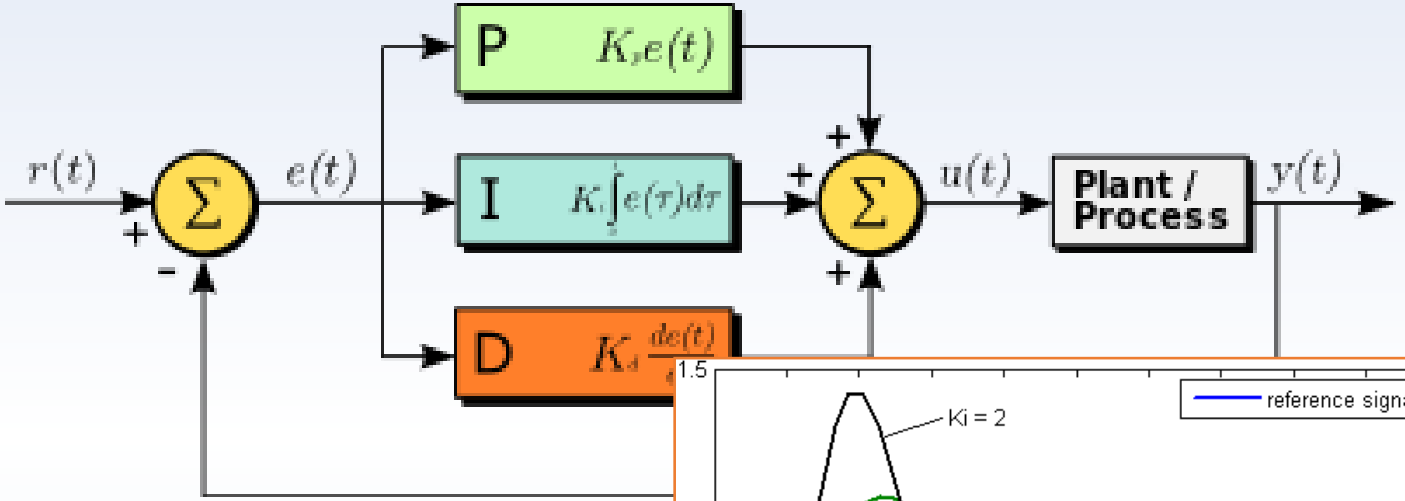
$$\hat{x}[t+s] = f(\ x[t],\ x[t-1],\ \cdots\ )$$

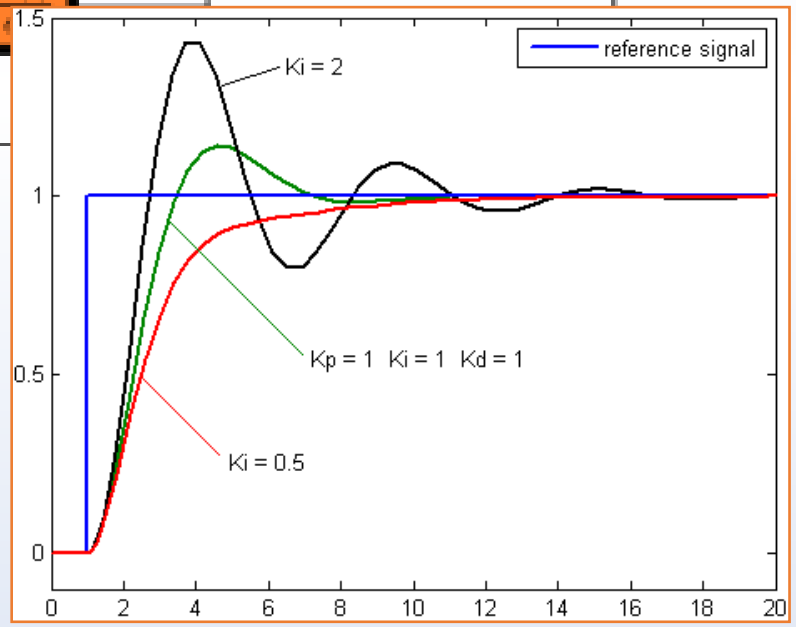
NN as Filter / Controller



PID controller - proportional–integral–derivative controller







$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(t')\,dt' + K_\mathrm{d} \frac{de(t)}{dt}$$

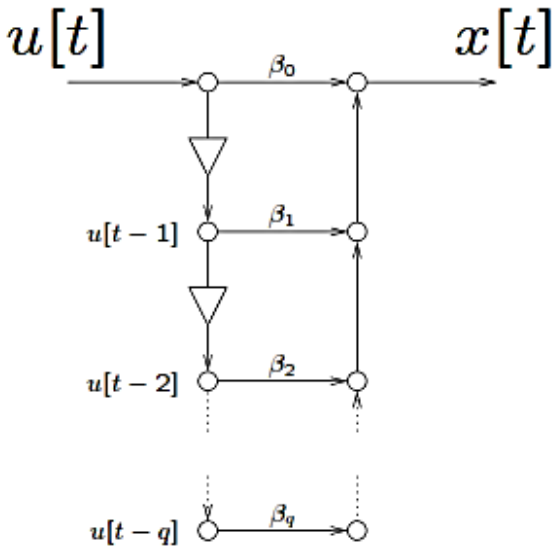
Finite Impulse Response (FIR) Filters

$$x[t] = \sum_{i=0}^{q} \beta_i \, u[t-i]$$

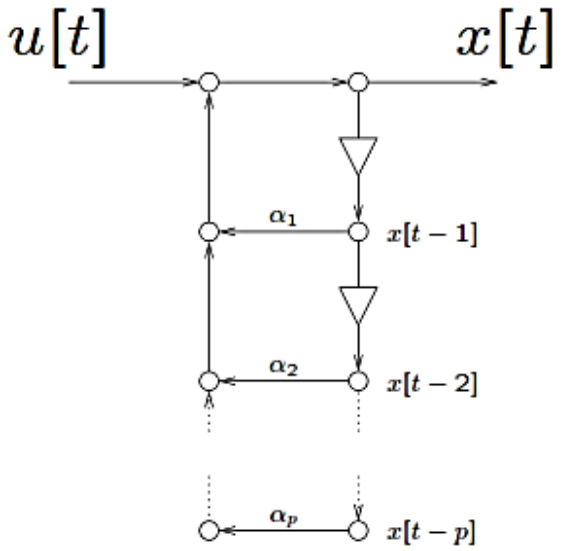
Infinite Impulse Response (IIR) Filters

$$x[t] = \sum_{i=1}^{p} \alpha_i \, x[t-i] + u[t]$$

In DSP notation:



FIR



IIR

Moving Average (MA[q])

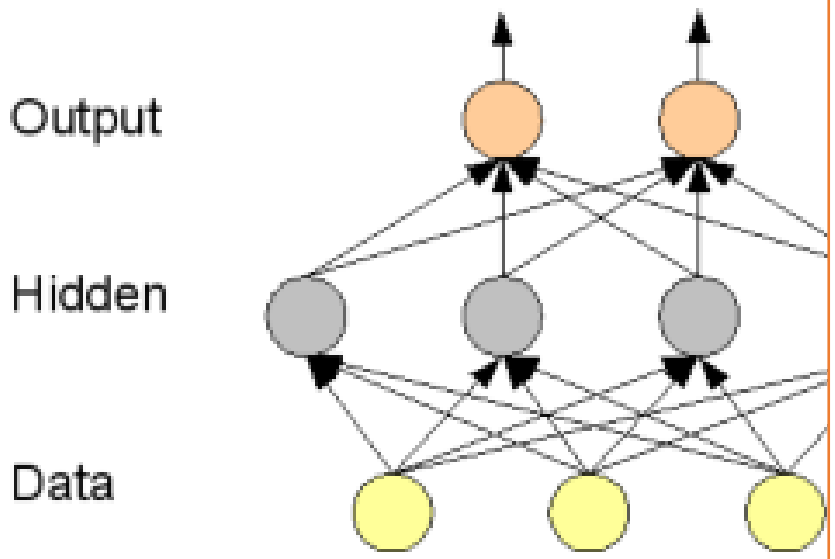
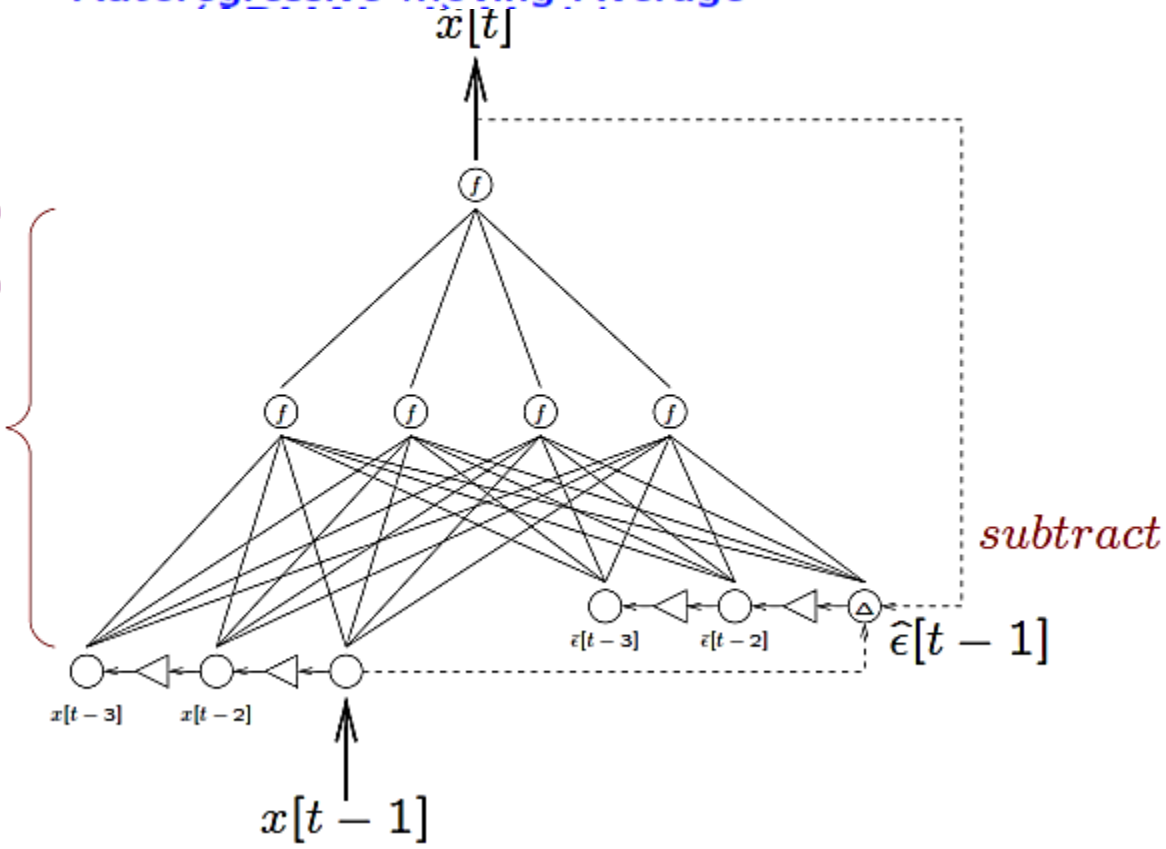


Figure 2: A feed-forward neural network

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} w_{ik}^{l-1} y_i^{l-1}$$

Nonlinear ARMA[p, q] Models

Autoregressive Moving Average



$$\hat{x}[t] = \sum_{i=1}^{p} \alpha_i x[t-i] + \sum_{i=1}^{q} \beta_i \hat{\epsilon}[t-i]$$