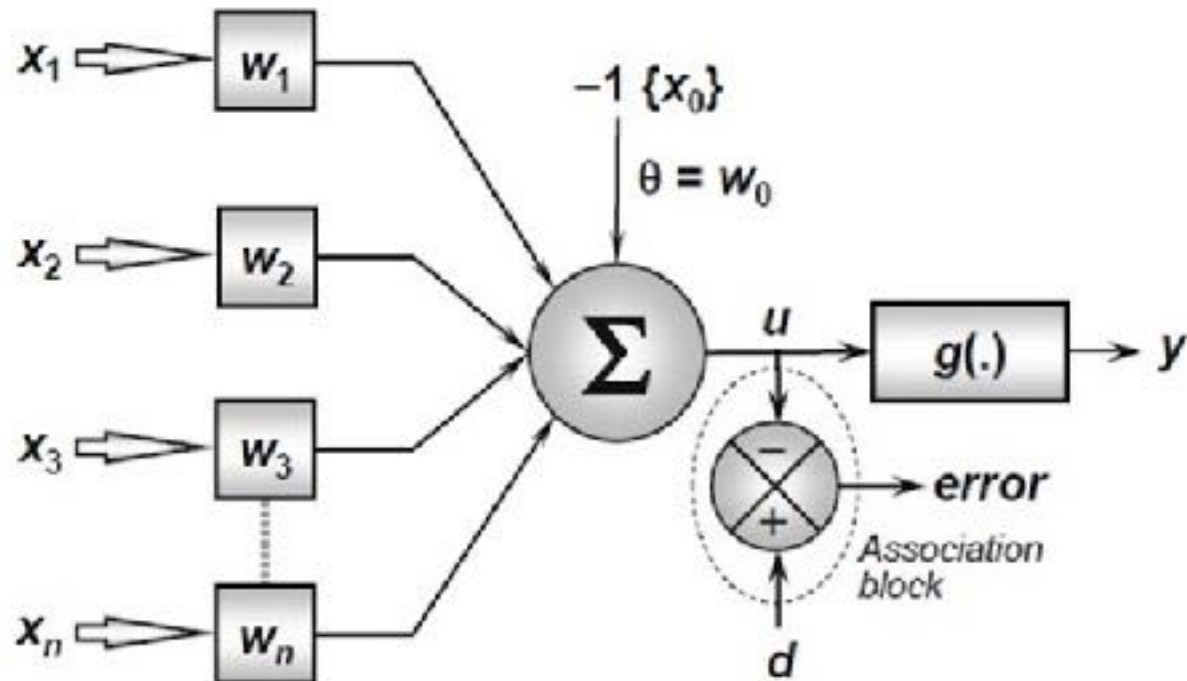


Redes Lineales

ADALINE - Adaptive Linear Element



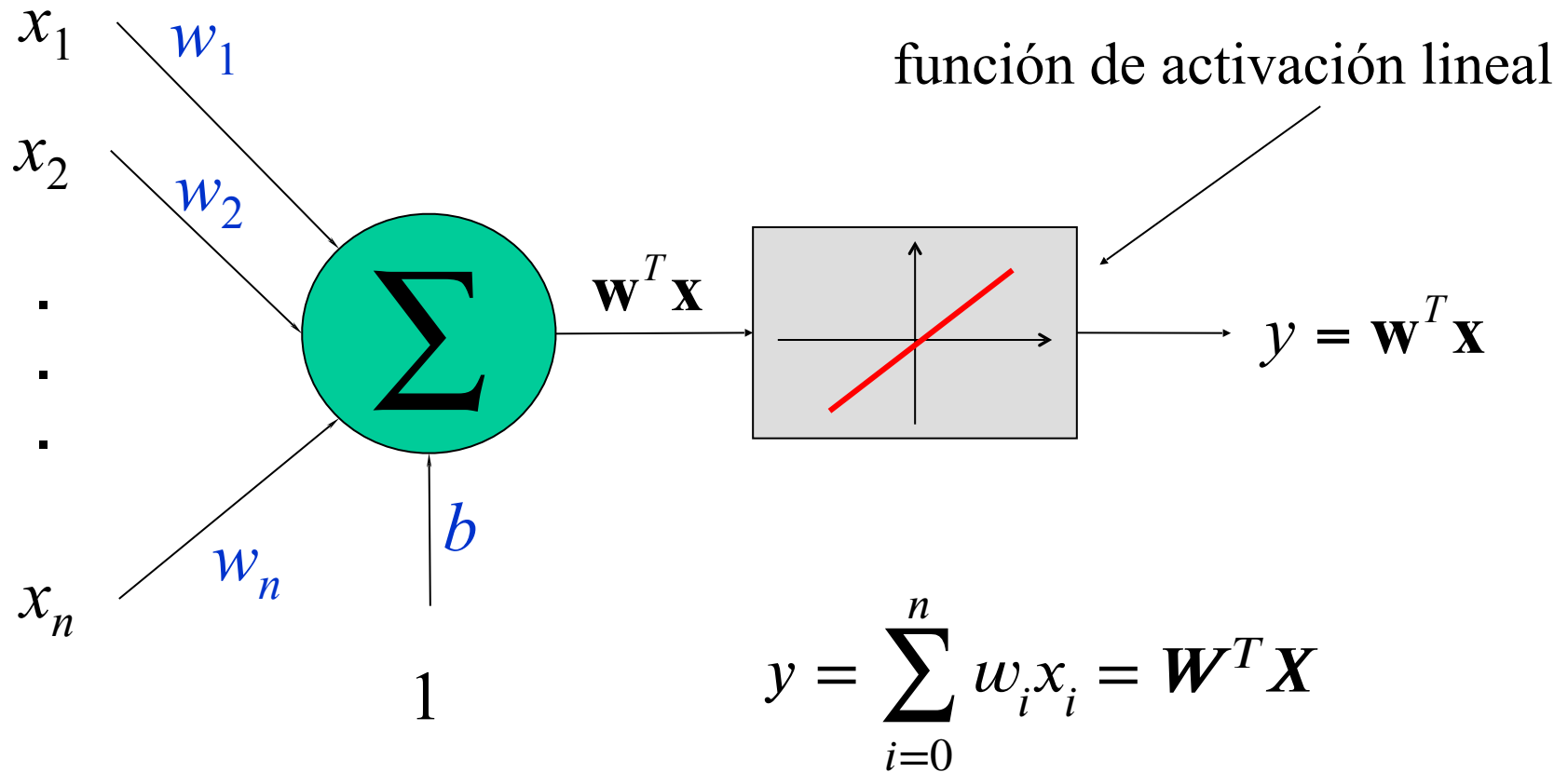
ADALINE (Adaptive Linear Element)

- ❑ Estructura
- ❑ Características
- ❑ Regla de aprendizaje Widrow-Hoff
- ❑ Ejemplo

ADALINE (Adaptive Linear Element)

- Desarrollada por el profesor Bernard Widrow y su alumno Ted Hoff (1960), en la universidad de Stanford. El modelo está basado en la Neurona de McCulloch-Pitts.
- Similar al perceptron, una ADALINE es capaz de clasificar patrones linealmente separables.
- Pueden procesar señales analógicas utilizando una función de activación lineal.
- La principal diferencia esta en como se corrige la señal de error.

Estructura de la red Adaline



Adaline características

- Utiliza la regla Delta de Widrow & Hoff o regla del mínimo error cuadrado medio (LMS, *Least Mean Squared*)
- La regla delta se basa en la búsqueda del mínimo de una expresión del error entre la salida deseada y la salida lineal obtenida.
- Tanto ADALINE como el perceptron sufren de la misma limitación, solo pueden resolver problemas linealmente separables.

Adaline características

- El algoritmo LMS es mas poderoso que la regla de aprendizaje del perceptron de F. Rosenblatt.
- El perceptron y ADALINE, MADALINE, son conocidas como redes de una capa o redes simples.
- A diferencia del perceptron, en el cual la frontera de solución está muy cerca de la solución y es muy sensible a ruido, ADALINE garantiza la convergencia a la solución que categoriza correctamente los patrones de entrenamiento.

Adaline características

CARACTERISTICA	PERCEPTRON	ADALINE
FUNCION DE ACTIVACION	escalón, signo	lineal
PROBLEMAS	linealmente separables	linealmente separables
RESPECTO AL RUIDO	sensible	minimiza el ruido
ALGORITMO DE APRENDIZAJE	regla del perceptron, error amplificado	LMS

Regla de Aprendizaje Widrow-Hoff

Aprendizaje Widrow-Hoff

Widrow & Hoff propone su regla desde la técnica de mínimos cuadrados, basado en el descenso por gradiente (algoritmo de optimización), considerado lento pero versátil, para la búsqueda de los pesos W que minimicen el error.

$$w^{new} = w^{old} + \Delta w$$

Para minimizar, se escoge

$$\Delta \mathbf{w} = -\alpha \nabla E$$

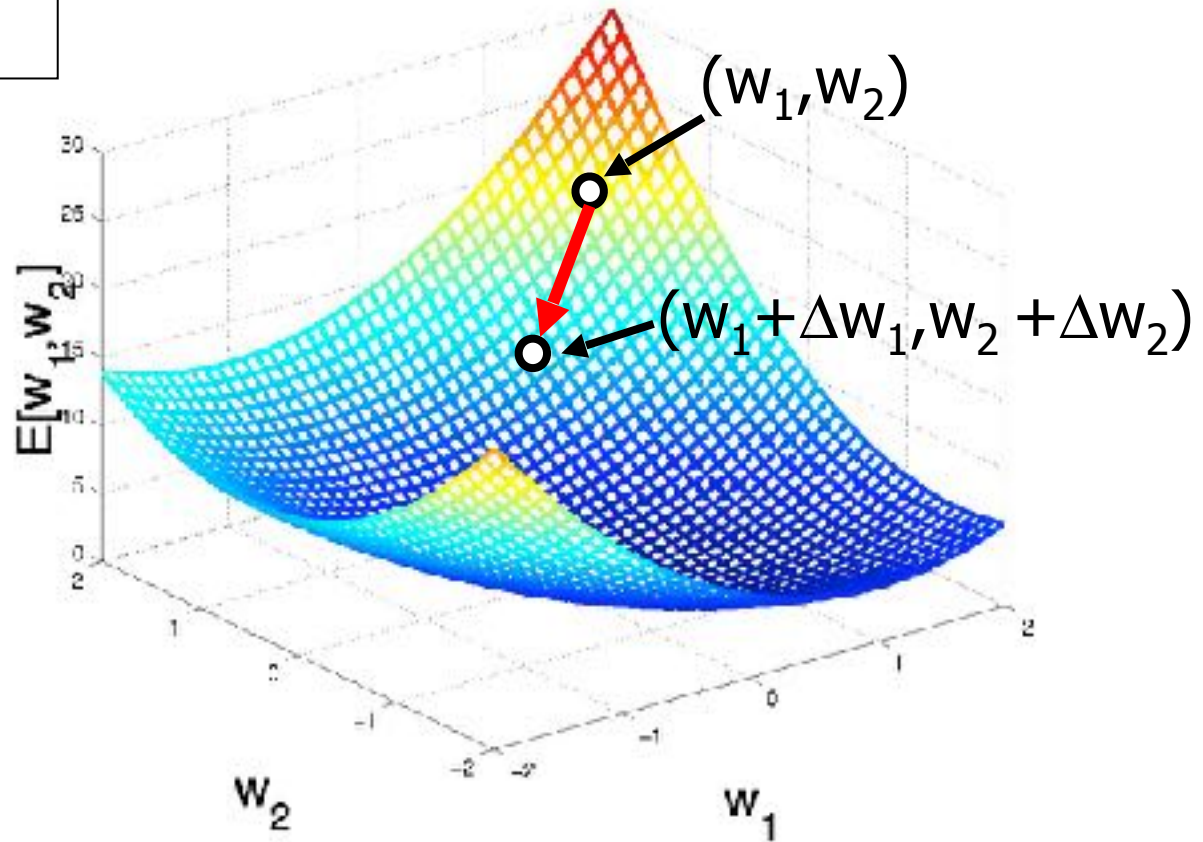
$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Descenso por el gradiente

$$\Delta \mathbf{w} = -\alpha \nabla E$$

Es uno de los algoritmos de optimización más populares en aprendizaje automático, particularmente por su uso extensivo en el campo de las redes neuronales.

Gradient descent es un método general de **minimización** para cualquier función.



Regla de aprendizaje

- Salida para el k-ésimo vector de entrada de la red lineal,

$$y_k = \sum_{i=0}^n w_i x_{ki}$$

- Si t_k es la salida deseada, entonces el error es,

$$\varepsilon_k = t_k - y_k$$

Regla de aprendizaje

- El error cuadrático medio, o valor esperado del error, se define en la forma:

$$E = \sum_{k=1}^L E_k = \frac{1}{2} \sum_{k=1}^L (t_k - y_k)^2$$

L: número de vectores de entrada

Regla de aprendizaje

- La actualización de pesos basada en el descenso por el gradiente, esta dada por

$$\Delta w_i = -\alpha \frac{\partial E_k}{\partial w_i}$$

- Aplicamos la regla de la cadena para el cálculo del gradiente,

$$\Delta w_i = -\alpha \frac{\partial E_k}{\partial w_i} = -\alpha \frac{\partial E_k}{\partial y_k} \frac{\partial y_k}{\partial w_i}$$

Regla de aprendizaje

- Cálculo del primer término,

$$\frac{\partial E_k}{\partial y_k} = \frac{\partial}{\partial y_k} \left[\frac{1}{2} (t_k - y_k)^2 \right]$$

$$\frac{\partial E_k}{\partial y_k} = - (t_k - y_k) = - \varepsilon_k$$

Regla de aprendizaje

- Calculando el segundo término,

$$\frac{\partial y_k}{\partial w_i} = \frac{\partial}{\partial w_i} \left(\sum_{i=0}^n w_i x_{ki} \right)$$

$$y_k = w_0 x_{k0} + w_1 x_{k1} + w_2 x_{k2} + \cdots + w_n x_{kn}$$

$$\frac{\partial y_k}{\partial w_i} = x_k$$

Regla de aprendizaje

- Reuniendo los resultados, la actualización de pesos está dada por,

$$\Delta w_i = -\alpha \left(-\varepsilon_k x_k \right) = \alpha \varepsilon_k x_k$$

- Así,

$$w_i(k+1) = w_i(k) + \Delta w_i$$

Algoritmo de entrenamiento

1. Seleccionar los pesos iniciales w_1, w_2, \dots, w_m y el umbral b con números aleatorios.
2. Presentar las entradas un patrón x_k . Calcular la salida de la red en la iteración k .

$$y_k = \sum_{i=0}^n w_i x_{ki}$$

3. Calcular el error con respecto a la salida deseada

$$\epsilon_k = t_k - y_k$$

Algoritmo de entrenamiento

4. Actualización de los pesos

$$w_i(k + 1) = w_i(k) + \alpha \varepsilon_k x_k$$

5. Incrementar k en uno, volver al Paso 2 y repetir el proceso hasta la convergencia.

Por ejemplo; “Si el error cuadrático medio es más pequeño un umbral se termina la iteración de aprendizaje”

Matlab

% Red lineal

net = linearlayer;

% Entrenamiento

net = train(net, P, T);

% Simulación

y = net(p)

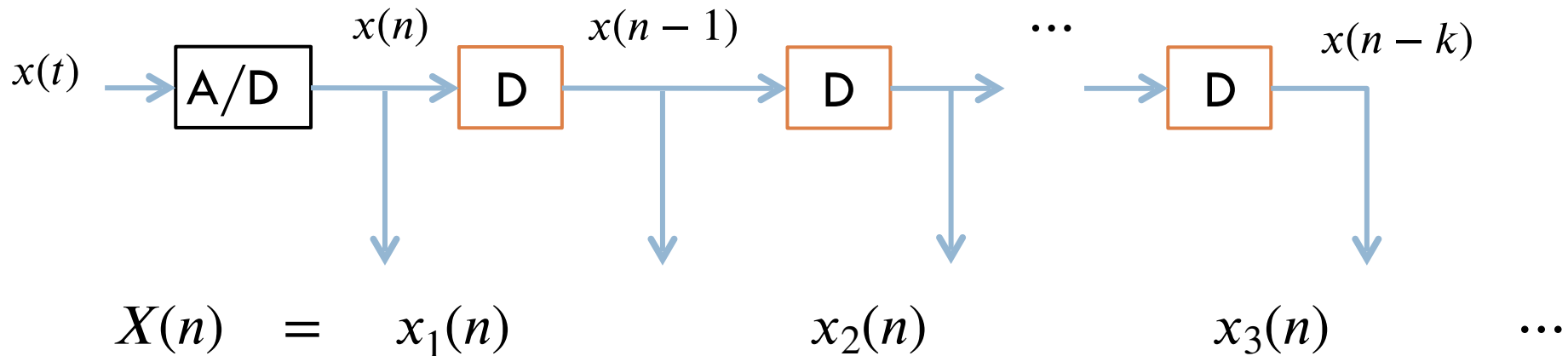
Vectores de señales

- En el procesamiento de señales en tiempo real una señal analógica pasa por un conversor A/D, el cual produce muestras de la señal.
- Estas muestras pueden agruparse en un vector de k elementos de la señal de entrada,

$$X(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-k)]$$

Vectores de señales

- Este vector, de la muestra actual y las $p-1$ pasadas, es creado por una línea de retardo.



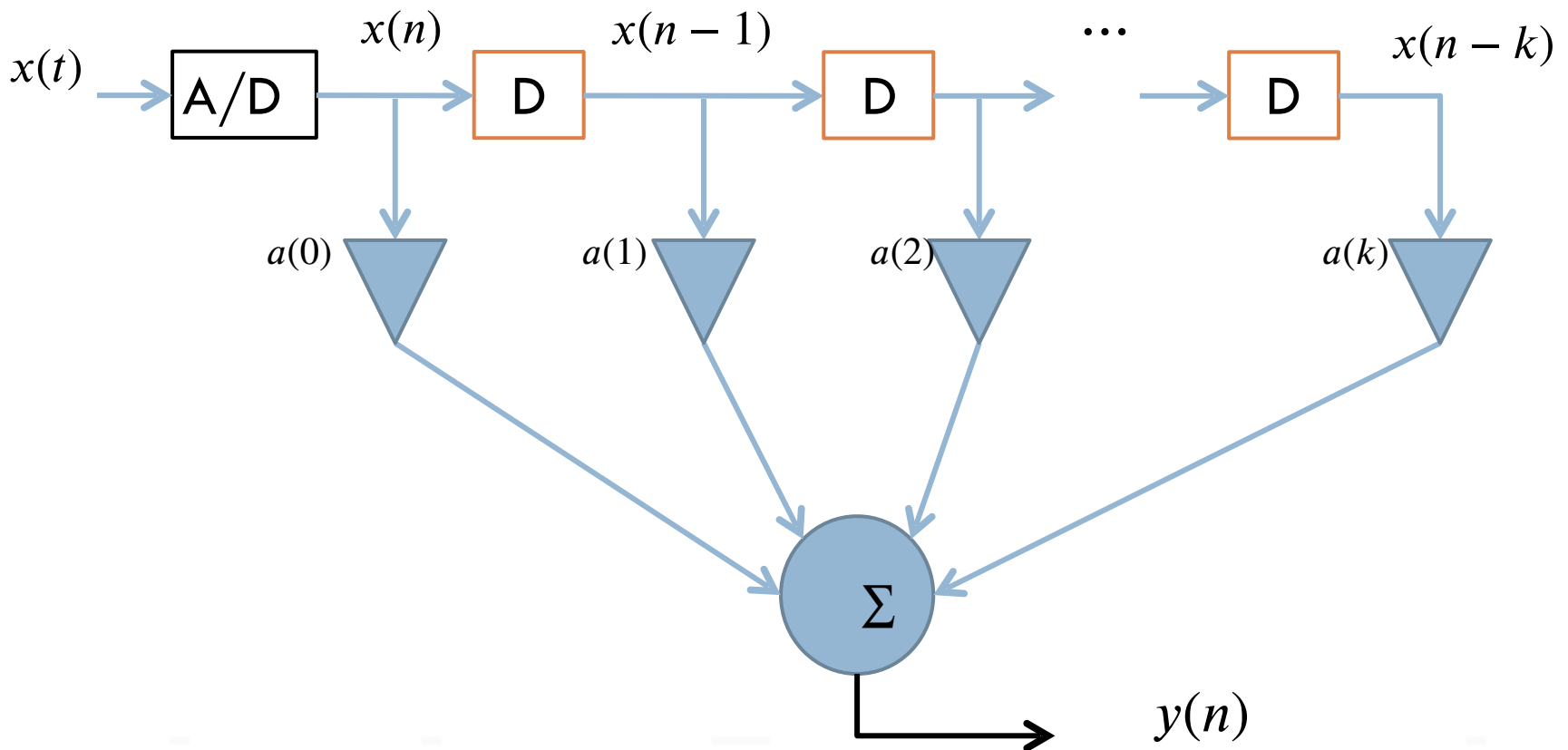
Vectores de señales

- **FIR (Respuesta al Impulso Finita):** La respuesta en el tiempo es la suma pesada de un número finito de muestras presentes y pasadas.

$$y(n) = \sum_{k=0}^{N-1} a(k)x(n - k)$$

Filtro FIR

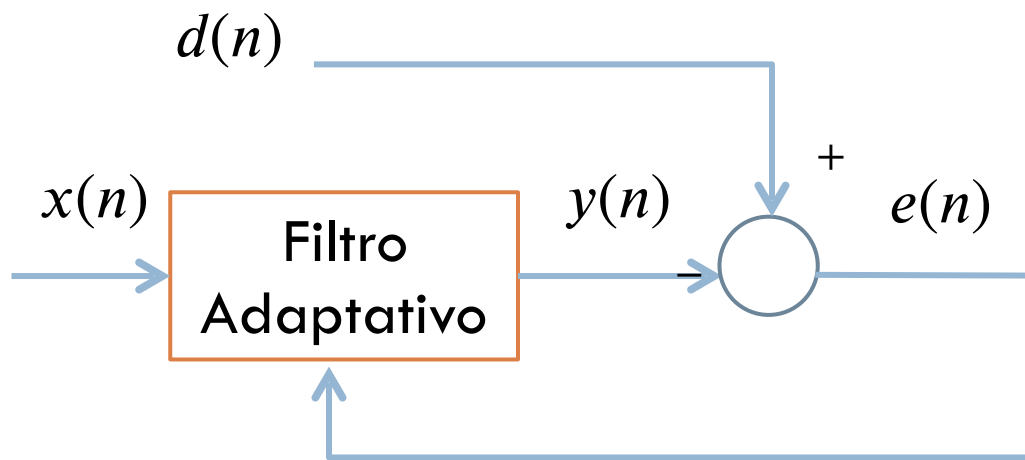
$$y(n) = \sum_{k=0}^{N-1} a(k)x(n-k]$$



Filtro FIR

- Si conectamos las salidas de los elementos de retardo a las sinapsis de una Adaline, resulta la estructura de procesamiento de señales conocida como **Filtro digital lineal de orden k FIR** (Finite-Impulse-response)

Filtro Adaptativo

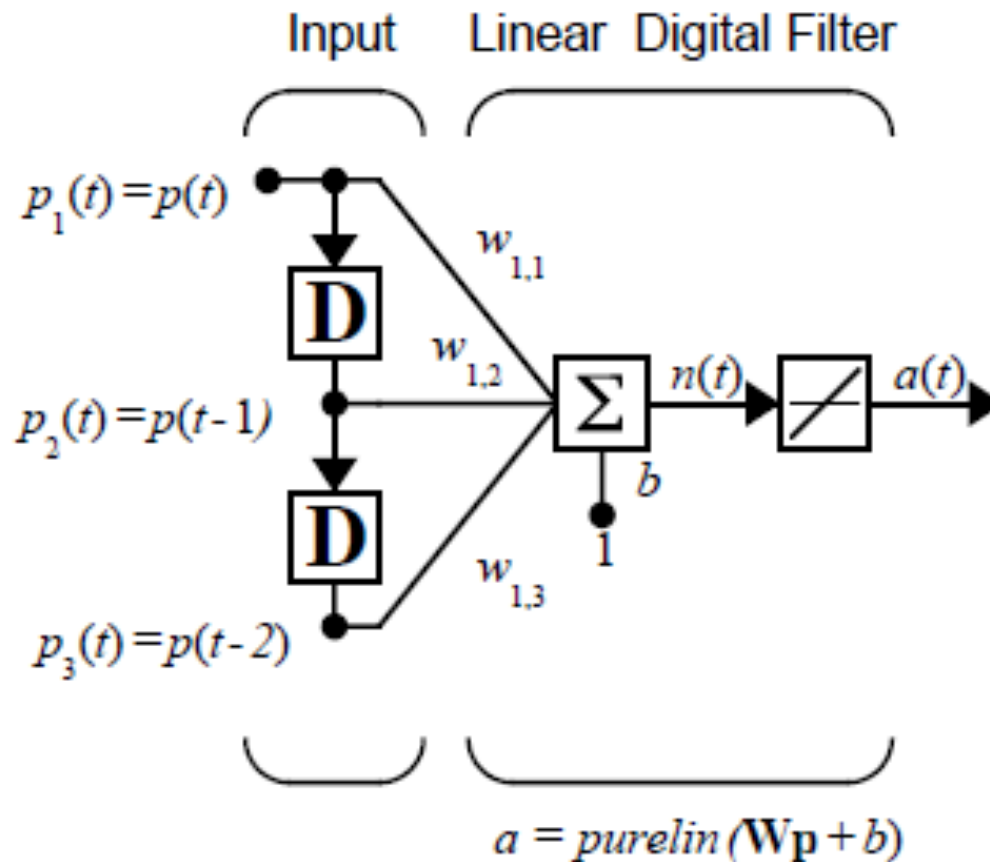


$$y(n) = \sum_{k=0}^{N-1} a(k)x(n-k)$$

$$e(n) = d(n) - y(n)$$

El objetivo es hacer que la señal de error sea cero, para ello el sistema debe configurarse para que, a partir de la señal de entrada $x(n)$ se genera la salida $y(n)$ de forma que sea igual a la señal deseada $d(n)$.

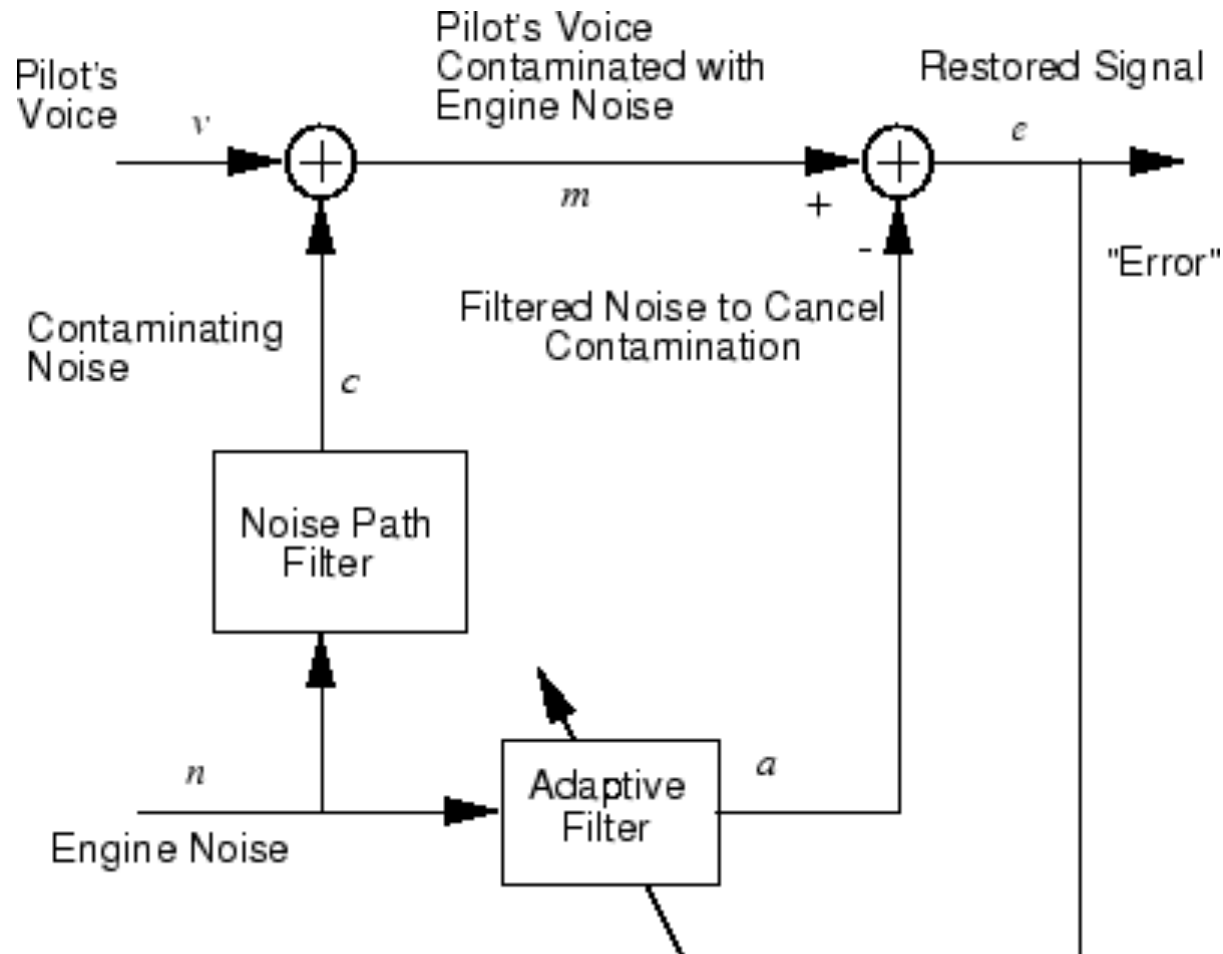
Adaline como un filtro adaptativo FIR



Filtro digital lineal de orden p FIR (Finite-Impulse-Response)

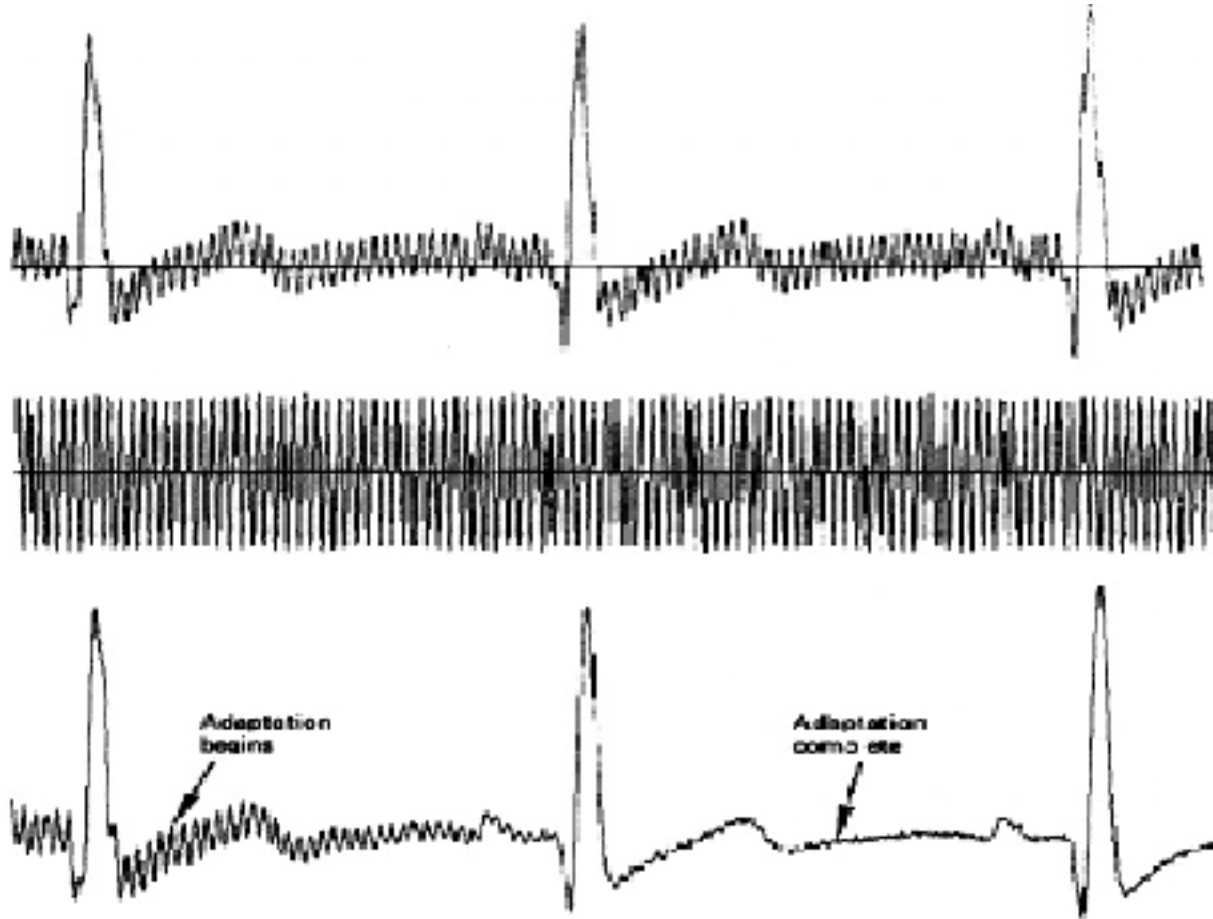
Cancelación adaptativa de ruido

- Una señal útil $u(n)$ es perturbada por un ruido, $x(n)$.
 - Por ejemplo, la voz de un piloto en un avión perturbada por el ruido originado desde el motor
- El ruido es coloreado por un filtro FIR desconocido
- El problema consiste en filtrar el ruido con el fin de obtener un estimado de la señal original



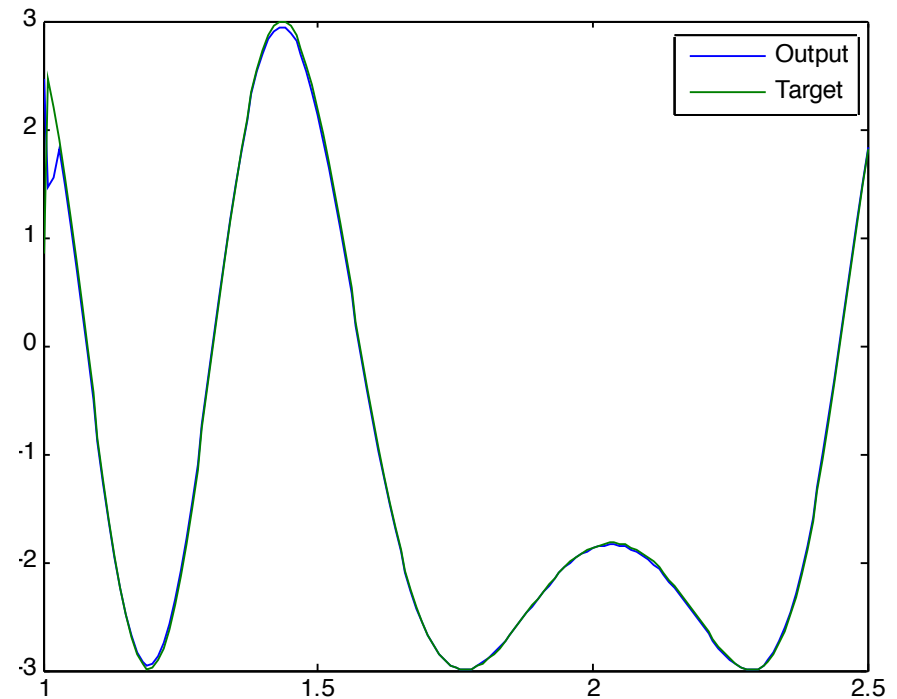
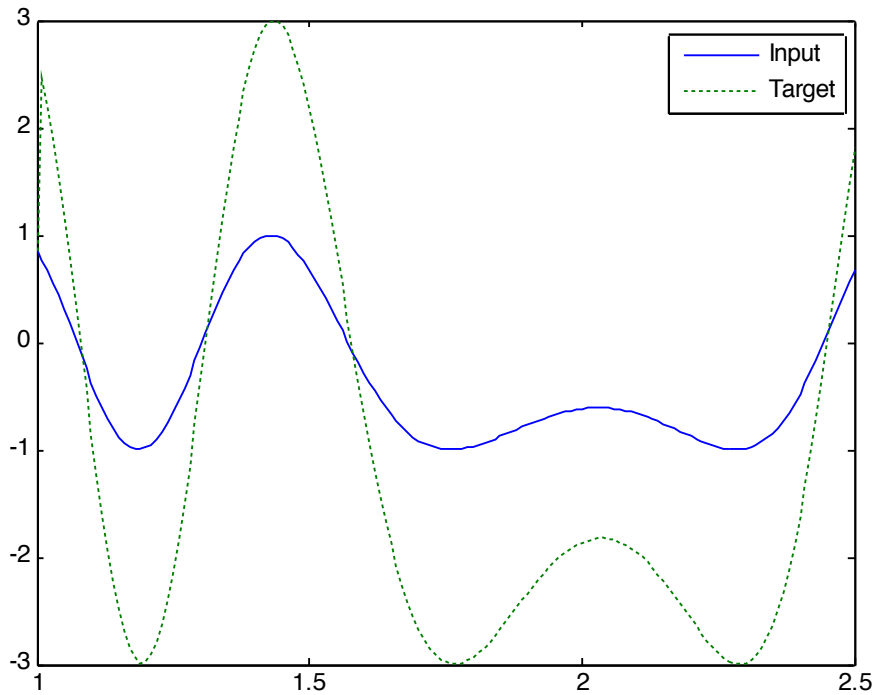
Adaptive Filter Adjusts to Minimize Error.
This removes the engine noise from contaminated
signal, leaving the pilot's voice as the "error."

Cancelación de la interferencia de 60hz en un ECG



Ejemplo de Predicción

- En este ejemplo se configura una Adaline para predecir una señal 1-D (serie de tiempo)



¿ Preguntas ?

