

Filtrado Óptimo y Adaptativo

Algoritmo de Máxima Pendiente

- Tanto los problemas de filtrado óptimo como los de estimación o regresión lineal se resuelven de manera óptima empleando la solución de Wiener, las Ecuaciones Normales

$$\underbrace{\begin{pmatrix} R_{xx}[0] & R_{xx}[1] & \cdots & R_{xx}[N-1] \\ R_{xx}[1] & R_{xx}[0] & & \\ \vdots & & \ddots & \\ R_{xx}[N-1] & & & R_{xx}[0] \end{pmatrix}}_{\mathbf{R}} \underbrace{\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{pmatrix}}_{\mathbf{w}} = \underbrace{\begin{pmatrix} R_{dx}[0] \\ R_{dx}[-1] \\ \vdots \\ R_{dx}[-(N-1)] \end{pmatrix}}_{\mathbf{p}}$$

$$\mathbf{R}\mathbf{w} = \mathbf{p}$$

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$$

- En problemas de tiempo real (filtrado, por ejemplo) el coste puede ser demasiado alto
 - Cálculo de las autocorrelaciones y correlaciones cruzadas
 - Inversión de la matriz
- Alternativa: usar métodos iterativos, que aprovechan la información contenida en el error

- Esquemas iterativos:
 - Métodos de resolución de problemas basados en el acercamiento progresivo a una solución deseada.
 - Directamente relacionados con:
 - Filtrado adaptativo
 - Maximización o minimización de funciones
 - Búsqueda heurística de soluciones
- Caracterización del problema:
 - Punto de partida $\mathbf{x}$ (o datos de partida, señal de partida...).
 - Se busca un conjunto de variables $\mathbf{w}_{opt}$ que, aplicados a $\mathbf{x}$ por medio de una función f conocida, conduzcan a un punto deseado d .

$$d = f(\mathbf{x}, \mathbf{w}_{opt})$$

- En un método iterativo:
 - Dispondremos de unos valores $\mathbf{w}(i)$ que corresponderán a:

$$\tilde{d}(i) = f(\mathbf{x}, \mathbf{w}(i))$$

- El error cometido es una función de la distancia entre d y $\tilde{d}(i)$
- Cada iteración hará una corrección de los valores $\mathbf{w}(i)$ de forma que $\mathbf{w}(i+1)$ genere una solución más cercana de la deseada.

$$\text{Dist}(d, \tilde{d}(i + 1)) < \text{Dist}(d, \tilde{d}(i))$$

- El tipo de método depende de cómo se acerque la solución a la deseada:
 - Métodos basados en la superficie de error (optimización).
 - Métodos basados en la exploración del conjunto de soluciones (búsqueda).
 - Métodos basados en el incremento de información.

- En un método iterativo:
 - Dispondremos de unos valores $\mathbf{w}(i)$ que corresponderán a:

$$\tilde{d}(i) = f(\mathbf{x}, \mathbf{w}(i))$$

- El error cometido es una función de la distancia entre d y $\tilde{d}(i)$
- Cada iteración hará una corrección de los valores $\mathbf{w}(i)$ de forma que $\mathbf{w}(i+1)$ genere una solución más cercana de la deseada.

$$\text{Dist}(d, \tilde{d}(i + 1)) < \text{Dist}(d, \tilde{d}(i))$$

- El tipo de método depende de cómo se acerque la solución a la deseada:
 - Métodos basados en la superficie de error (optimización).
 - Métodos basados en la exploración del conjunto de soluciones (búsqueda).
 - Métodos basados en el incremento de información.

Superficie de error

- La superficie de error es una representación del error cometido por cada posible solución (óptima o no) al problema.
- La solución óptima corresponderá al mínimo de la superficie de error (si existe un mínimo global).
- La superficie de error contiene mucha información que permite solucionar la búsqueda de la solución óptima según diferentes planteamientos.

Por ejemplo:

- Utilización del **gradiente** de la superficie de error
 - Máxima pendiente
 - Método de Newton
 - LMS
 - Etc.

Superficie de error Filtrado óptimo

- El error cometido por un conjunto de coeficientes puede expresarse de la siguiente manera:

$$e[n] = d[n] - \hat{d}[n] = d[n] - \sum_{k=0}^{N-1} w_k x[n-k] = d[n] - \mathbf{w}^T \mathbf{x}$$

$$\mathbf{w} = \{w_0, w_1 \dots w_{N-1}\}^T$$

$$\mathbf{x} = \{x[n], x[n-1] \dots x[n-N+1]\}^T$$

- Creamos una función $J(\mathbf{w})$ que proporcione la esperanza del error cuadrático para cualquier conjunto de coeficientes

$$J(\mathbf{w}) = E[e^2[n]]$$

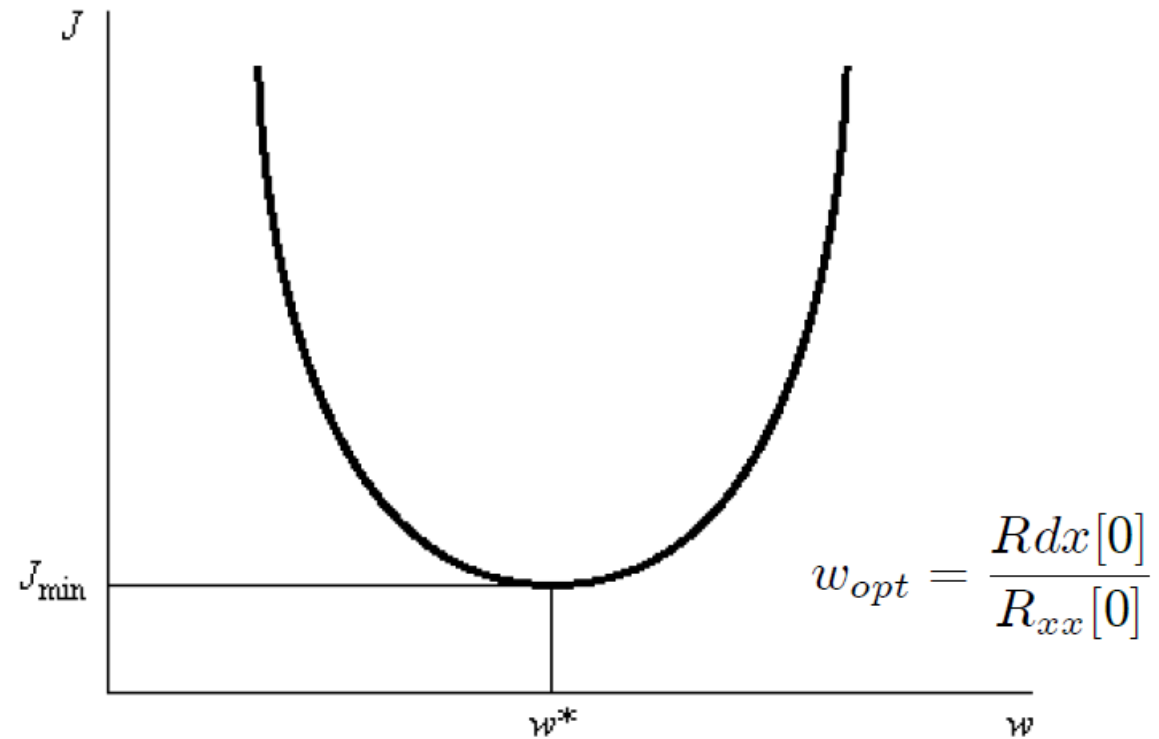
$$J(\mathbf{w}) = E[(d[n] - \mathbf{w}^T \mathbf{x})^2]$$

$$J(\mathbf{w}) = E[d^2[n]] - 2\mathbf{w}^T \mathbf{p} + \mathbf{w}^T \mathbf{R} \mathbf{w}$$

Superficie de error
Filtrado óptimo

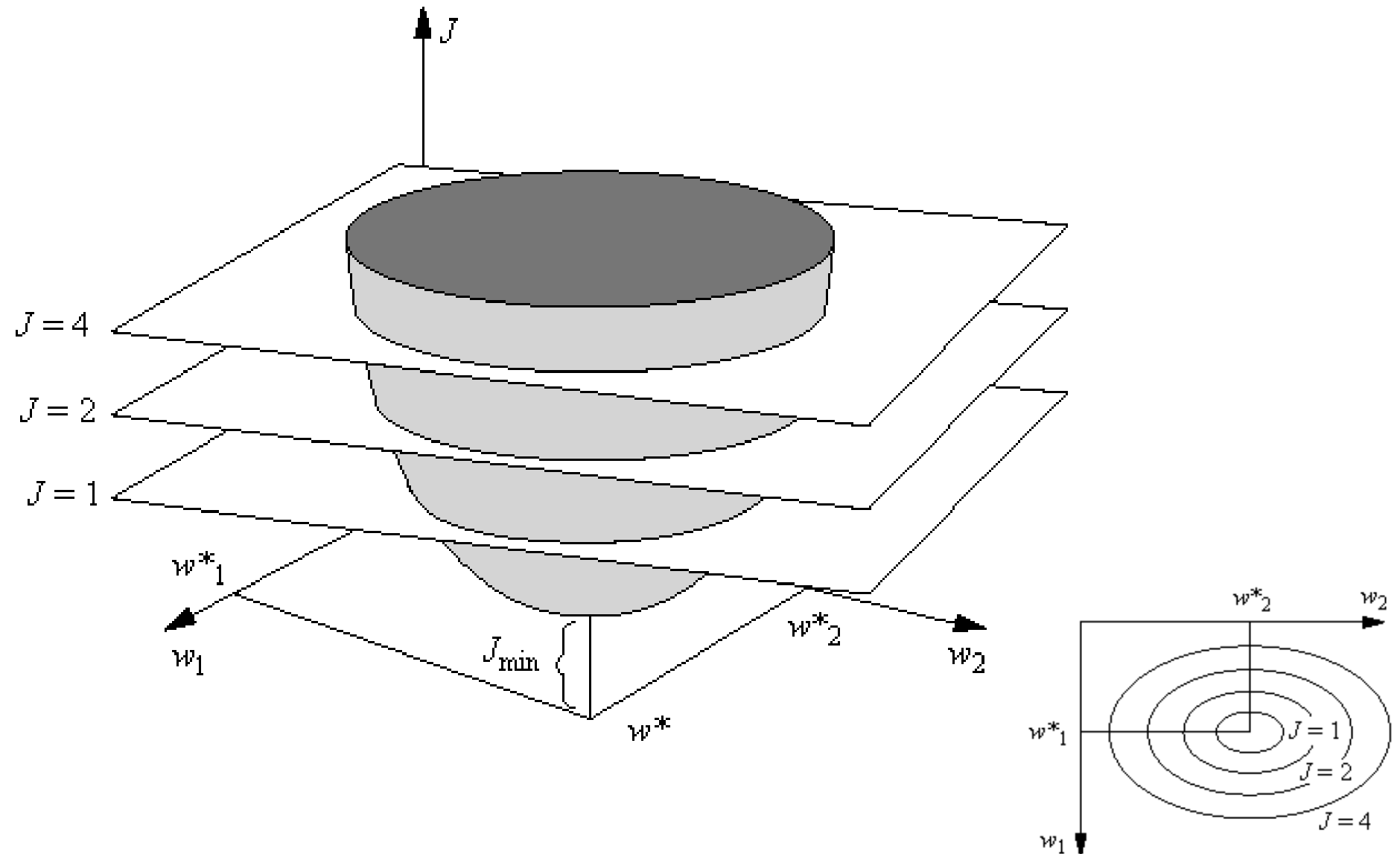
¿Qué forma tiene esta función? Filtro de 1 coeficiente

$$J(\mathbf{w}) = E[d^2[n]] - 2\mathbf{w}^T \mathbf{p} + \mathbf{w}^T \mathbf{R} \mathbf{w}$$



Superficie de error
Filtrado óptimo

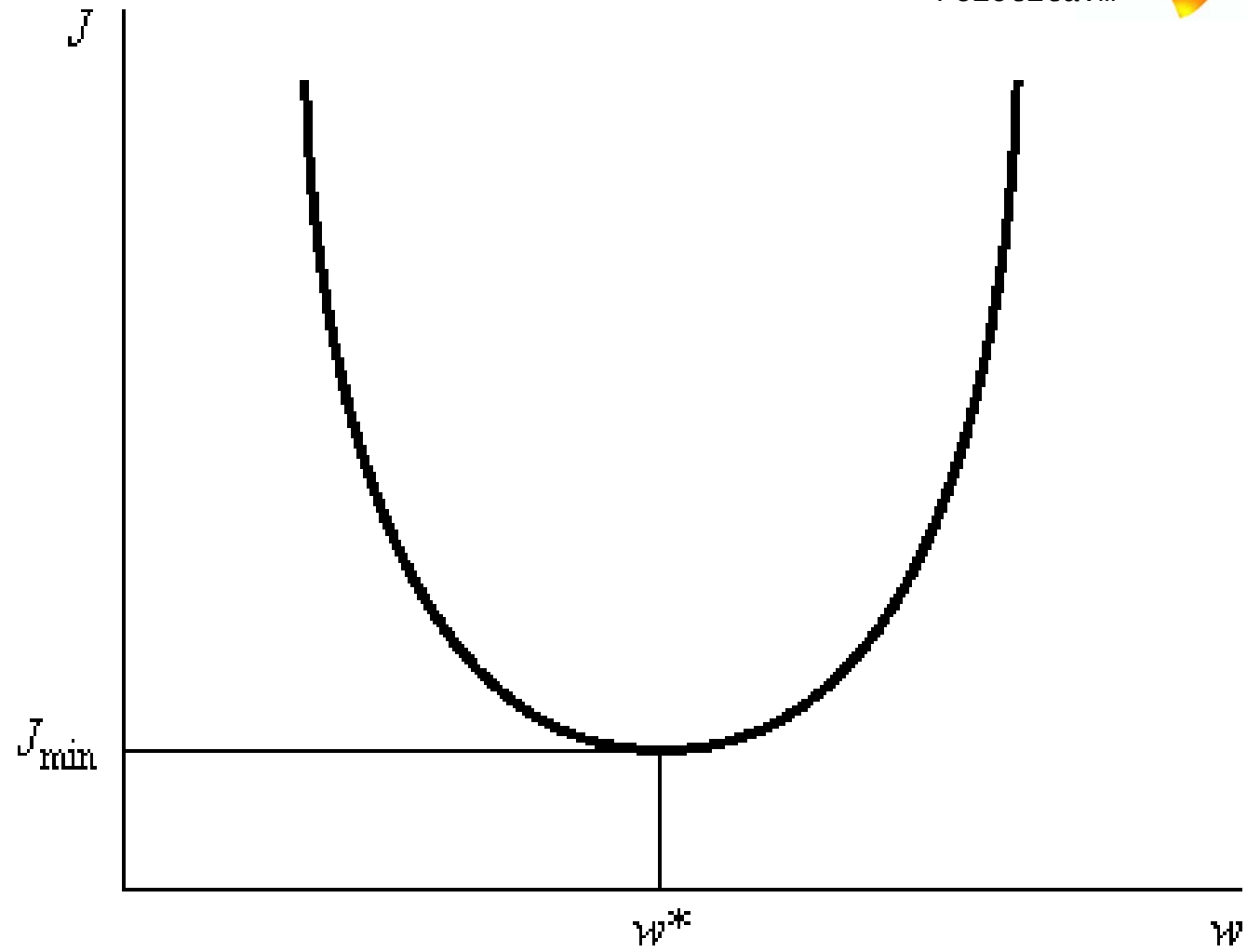
- Si se emplea un filtro de dos coeficientes:



Superficie de error Filtrado óptimo

- Se puede comprobar que la superficie de error es cuadrática para cualquier orden del filtro
- Tiene siempre un único mínimo correspondiente al conjunto óptimo de coeficientes para el orden del filtro especificado
- El mínimo siempre corresponde con el conjunto de coeficientes que satisfacen el principio de ortogonalidad, es decir, que son solución de las ecuaciones normales

- Algoritmo de la “pelotita”:



Superficie de error

- Dado que para este problema la superficie de error es una parábola con un único mínimo, podemos buscar ese mínimo para resolver el problema en lugar de hacerlo analíticamente.
- Para ello, emplearemos la información del gradiente.
 - Cuanto más lejos nos encontramos del óptimo:
 - Mayor es el gradiente
 - Se precisa mayor corrección en el parámetro libre
 - Cuanto más cerca nos encontramos del óptimo:
 - Menor es el gradiente
 - Se precisa menor corrección en el parámetro libre

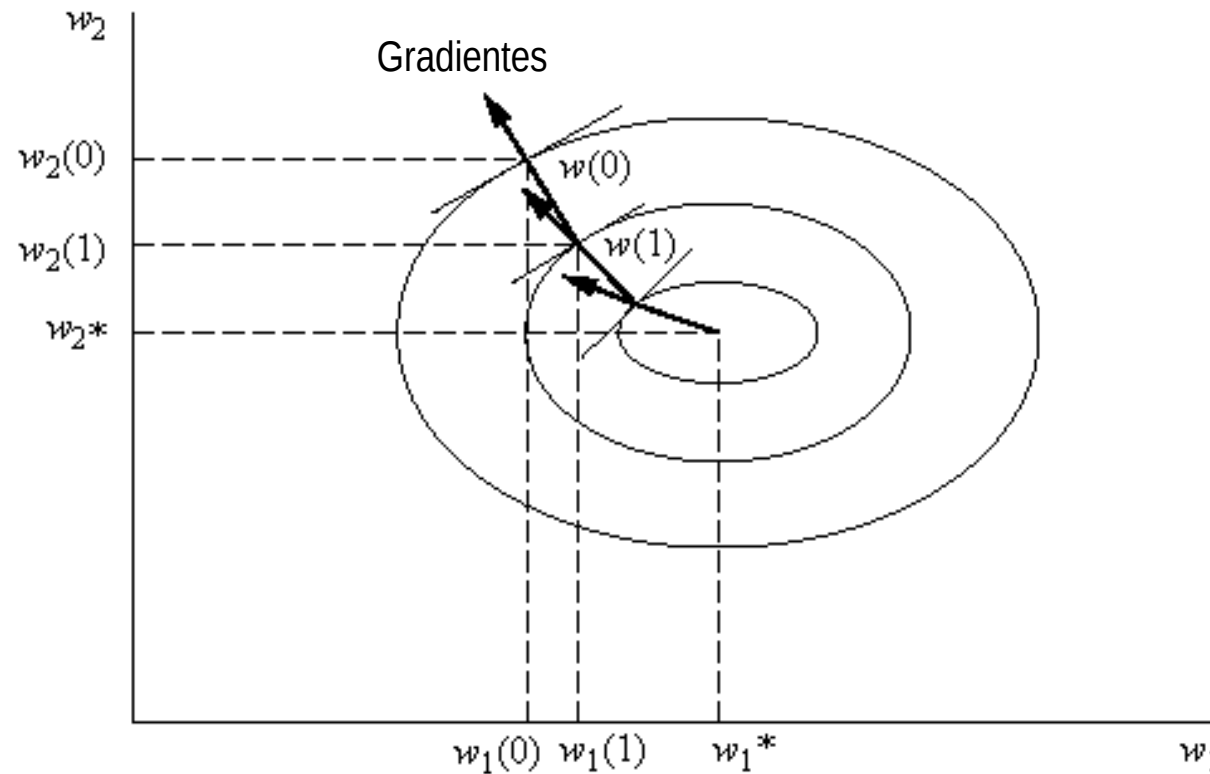
⇒ Corrección proporcional al gradiente y de signo contrario a éste:

$$w(k + 1) = w(k) - \mu \nabla J(k)$$

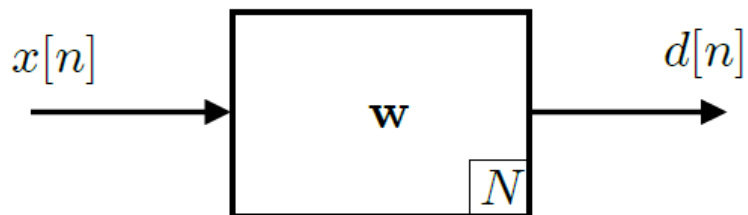
Superficie de error

- Este método se denomina **algoritmo de máxima pendiente** o **descenso de gradiente**
- Se puede generalizar para D variables

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \nabla \mathbf{J}(k) \qquad \nabla \mathbf{J} = \left[\frac{\partial J}{\partial w_0}, \dots, \frac{\partial J}{\partial w_D} \right]^T$$



Algoritmo de máxima pendiente



$$\mathbf{w} = \{w_0, w_1 \dots w_{N-1}\}^T$$

$$\mathbf{x} = \{x[n], x[n-1] \dots x[n-N+1]\}^T$$

$$\hat{d}[n] = \sum_{k=0}^{N-1} w_k x[n-k] = \mathbf{w}^T \mathbf{x}$$

$$J(\mathbf{w}) = E[d^2[n]] - 2\mathbf{w}^T \mathbf{p} + \mathbf{w}^T \mathbf{R} \mathbf{w}$$

- El algoritmo de máxima pendiente queda

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu[\mathbf{R}\mathbf{w}(k) - \mathbf{p}]$$

El algoritmo tiende al óptimo, que era

$$\mathbf{R}\mathbf{w} = \mathbf{p}$$

Algoritmo de máxima pendiente

$$J(\mathbf{w}) = E[d^2[n]] - 2\mathbf{w}^T \mathbf{p} + \mathbf{w}^T \mathbf{R} \mathbf{w}$$

- El algoritmo de máxima pendiente queda

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu[\mathbf{R}\mathbf{w}(k) - \mathbf{p}]$$

- También se puede comprobar que la siguiente expresión es equivalente

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu E[e[n] \cdot \mathbf{x}]$$

Algoritmo de máxima pendiente

Algoritmo de máxima pendiente

$$J(\mathbf{w}) = E[d^2[n]] - 2\mathbf{w}^T \mathbf{p} + \mathbf{w}^T \mathbf{R} \mathbf{w}$$

- El algoritmo de máxima pendiente queda

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu[\mathbf{R}\mathbf{w}(k) - \mathbf{p}]$$

- También se puede comprobar que la siguiente expresión es equivalente

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu E[e[n] \cdot \mathbf{x}]$$

- Cuando se alcanza la solución óptima, no hay corrección de los coeficientes

$$\mathbf{R}\mathbf{w} = \mathbf{p}$$

- Eso significa que, según la segunda expresión, el óptimo se alcanza cuando se cumple que

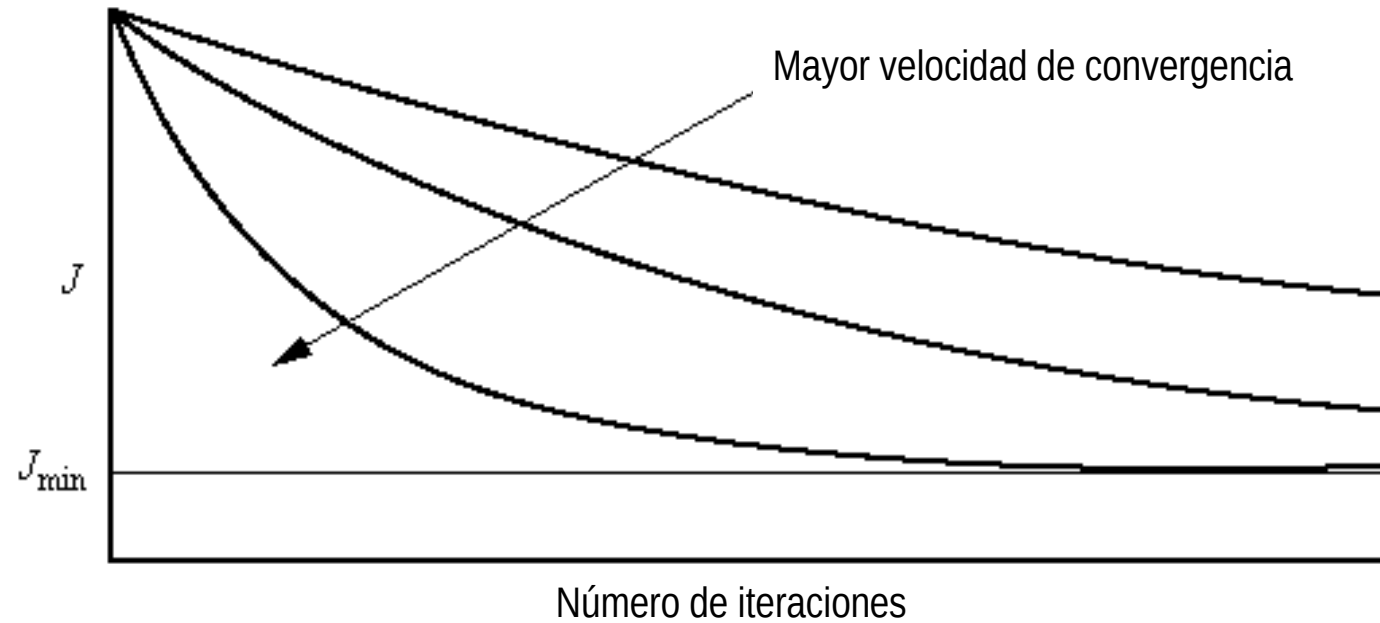
$$E[e[n] \cdot \mathbf{x}] = 0$$

$$E[e[n] \cdot x[n-k]] = 0, \quad k = 0 \dots N-1$$

- **Principio de Ortogonalidad**

Convergencia

- Curvas de aprendizaje de un algoritmo



- Desajuste:

$$M = \frac{J_{final} - J_{min}}{J_{min}}$$