Redes Multicapa (cont)

Fernando Lozano

Universidad de los Andes

8 de septiembre de 2017

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

  - Si generamos los pesos independientemente $\sim N(0, \sigma^2)$

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

  - Si generamos los pesos independientemente $\sim N(0, \sigma^2)$

$$a = \sum_{i=0}^{d} w_i x_i$$

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

  - Si generamos los pesos independientemente $\sim N(0, \sigma^2)$

$$a = \sum_{i=0}^{d} w_i x_i \Rightarrow \mathbb{E}a =$$

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

  - Si generamos los pesos independientemente $\sim N(0, \sigma^2)$

$$a = \sum_{i=0}^{d} w_i x_i \Rightarrow \mathbb{E}a = 0, \quad \mathbb{E}a^2 =$$

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

  - Si generamos los pesos independientemente $\sim N(0, \sigma^2)$

$$a = \sum_{i=0}^{d} w_i x_i \Rightarrow \mathbb{E}a = 0, \quad \mathbb{E}a^2 = \sigma^2 d$$

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

  - Si generamos los pesos independientemente $\sim N(0, \sigma^2)$

$$a = \sum_{i=0}^{d} w_i x_i \Rightarrow \mathbb{E}a = 0, \quad \mathbb{E}a^2 = \sigma^2 d$$

  - Deseable $a \sim 1$

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

  - Si generamos los pesos independientemente $\sim N(0, \sigma^2)$

$$a = \sum_{i=0}^{d} w_i x_i \Rightarrow \mathbb{E}a = 0, \quad \mathbb{E}a^2 = \sigma^2 d$$

  - Deseable $a \sim 1 \Rightarrow \sigma \propto \frac{1}{\sqrt{d}}$

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

  - Si generamos los pesos independientemente $\sim N(0, \sigma^2)$

$$a = \sum_{i=0}^{d} w_i x_i \Rightarrow \mathbb{E}a = 0, \quad \mathbb{E}a^2 = \sigma^2 d$$

  - Deseable $a \sim 1 \Rightarrow \sigma \propto \frac{1}{\sqrt{d}}$
- Problema:

Inicialización de pesos

- Folklore: inicializar $\mathbf{w}$ con valores aleatorios pequeños.
  - Evitar problemas por simetrías.
  - Fuera de zona de saturación de la sigmoide.
  - No muy pequeños: región no lineal.
- Cuál es el tamaño apropiado?
  - Suponga que se reescala $\mathbf{x}$ de forma que

$$\mathbb{E}x_i = 0, \quad \mathbb{E}x_i^2 = 1$$

  - Si generamos los pesos independientemente $\sim N(0, \sigma^2)$

$$a = \sum_{i=0}^{d} w_i x_i \Rightarrow \mathbb{E}a = 0, \quad \mathbb{E}a^2 = \sigma^2 d$$

  - Deseable $a \sim 1 \Rightarrow \sigma \propto \frac{1}{\sqrt{d}}$
- Problema:muestrear Gaussiana en altas dimensiones.

Ejemplo

- En una dimensión:

$$y = \sum_{i=0}^{N-1} v_i \sigma(w_i x + w_{0i})$$

Ejemplo

- En una dimensión:

$$y = \sum_{i=0}^{N-1} v_i \sigma(w_i x + w_{0i}) = \sum_{i=0}^{N-1} y_i$$

Ejemplo

- En una dimensión:

$$y = \sum_{i=0}^{N-1} v_i \sigma(w_i x + w_{0i}) = \sum_{i=0}^{N-1} y_i$$

- Con $\sigma(z) = \tanh(z)$ ∼lineal en $[-1,\ 1]$.

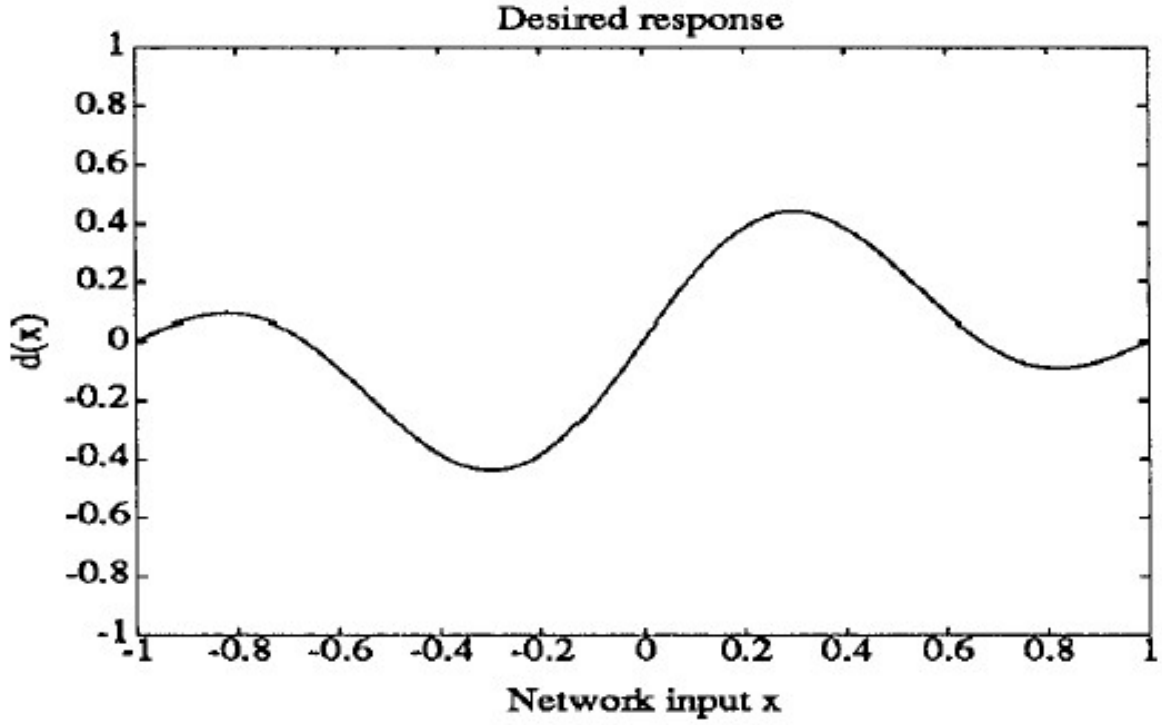


Figure 1: Desired response for first example

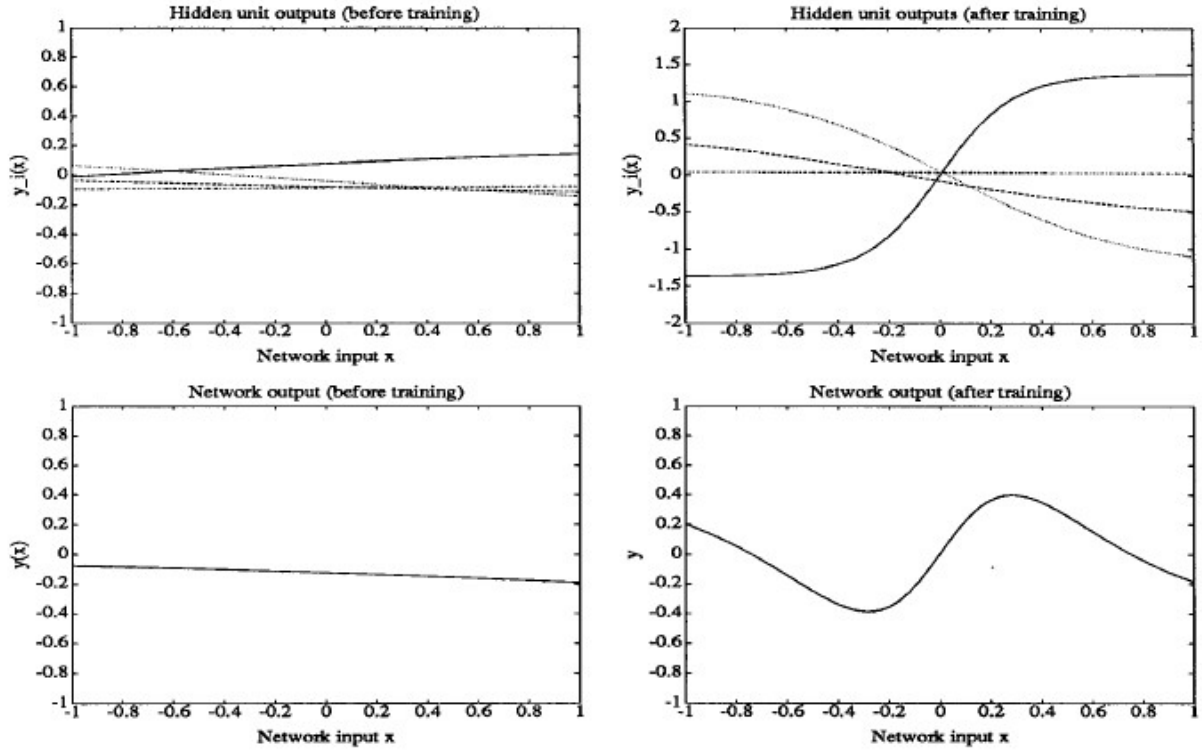


Figure 2: Outputs of network and hidden units before and after training with weights initialized to random values between -0.5 and 0.5

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).
- $\sigma(w_i x + w_{0i}) \approx w_i x + w_{0i}$ en $[-1,\ 1]$:

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).
- $\sigma(w_i x + w_{0i}) \approx w_i x + w_{0i}$ en $[-1,\ 1]$:

$$-1 < w_i x + w_{0i} < 1$$

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).
- $\sigma(w_i x + w_{0i}) \approx w_i x + w_{0i}$ en $[-1,\ 1]$:

$$-1 < w_i x + w_{0i} < 1 \Rightarrow -\frac{1}{w_i} - \frac{w_{0i}}{w_i} < x < \frac{1}{w_i} - \frac{w_{0i}}{w_i}$$

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).
- $\sigma(w_i x + w_{0i}) \approx w_i x + w_{0i}$ en $[-1,\ 1]$:

$$-1 < w_i x + w_{0i} < 1 \Rightarrow -\frac{1}{w_i} - \frac{w_{0i}}{w_i} < x < \frac{1}{w_i} - \frac{w_{0i}}{w_i}$$

- Intervalo de longitud $\frac{2}{w_i}$

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).
- $\sigma(w_i x + w_{0i}) \approx w_i x + w_{0i}$ en $[-1,\ 1]$:

$$-1 < w_i x + w_{0i} < 1 \Rightarrow -\frac{1}{w_i} - \frac{w_{0i}}{w_i} < x < \frac{1}{w_i} - \frac{w_{0i}}{w_i}$$

- Intervalo de longitud $\frac{2}{w_i} \Rightarrow w_i \sim N$

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).
- $\sigma(w_i x + w_{0i}) \approx w_i x + w_{0i}$ en $[-1,\ 1]$:

$$-1 < w_i x + w_{0i} < 1 \Rightarrow -\frac{1}{w_i} - \frac{w_{0i}}{w_i} < x < \frac{1}{w_i} - \frac{w_{0i}}{w_i}$$

- Intervalo de longitud $\frac{2}{w_i} \Rightarrow w_i \sim N$
- Es conveniente permitir sobrelapamiento: $w_i = 0{,}7N$.

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).
- $\sigma(w_i x + w_{0i}) \approx w_i x + w_{0i}$ en $[-1,\ 1]$:

$$-1 < w_i x + w_{0i} < 1 \Rightarrow -\frac{1}{w_i} - \frac{w_{0i}}{w_i} < x < \frac{1}{w_i} - \frac{w_{0i}}{w_i}$$

- Intervalo de longitud $\frac{2}{w_i} \Rightarrow w_i \sim N$
- Es conveniente permitir sobrelapamiento: $w_i = 0{,}7N$.
- Centros aleatorios en$[-1,\ 1]$:

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).
- $\sigma(w_i x + w_{0i}) \approx w_i x + w_{0i}$ en $[-1,\ 1]$:

$$-1 < w_i x + w_{0i} < 1 \Rightarrow -\frac{1}{w_i} - \frac{w_{0i}}{w_i} < x < \frac{1}{w_i} - \frac{w_{0i}}{w_i}$$

- Intervalo de longitud $\frac{2}{w_i} \Rightarrow w_i \sim N$
- Es conveniente permitir sobrelapamiento: $w_i = 0{,}7N$.
- Centros aleatorios en$[-1,\ 1]$:

$$-\frac{w_{0i}}{w_i} \sim U[-1,\ 1]$$

Método de Nguyen-Widrow

- Idea: Dividir región de interés en N segmentos (uno para cada neurona).
- $\sigma(w_i x + w_{0i}) \approx w_i x + w_{0i}$ en $[-1,\ 1]$:

$$-1 < w_i x + w_{0i} < 1 \Rightarrow -\frac{1}{w_i} - \frac{w_{0i}}{w_i} < x < \frac{1}{w_i} - \frac{w_{0i}}{w_i}$$

- Intervalo de longitud $\frac{2}{w_i} \Rightarrow w_i \sim N$
- Es conveniente permitir sobrelapamiento: $w_i = 0{,}7N$.
- Centros aleatorios en$[-1,\ 1]$:

$$-\frac{w_{0i}}{w_i} \sim U[-1,\ 1] \Rightarrow w_{0i} \sim U[-|w_i|,\ |w_i|]$$

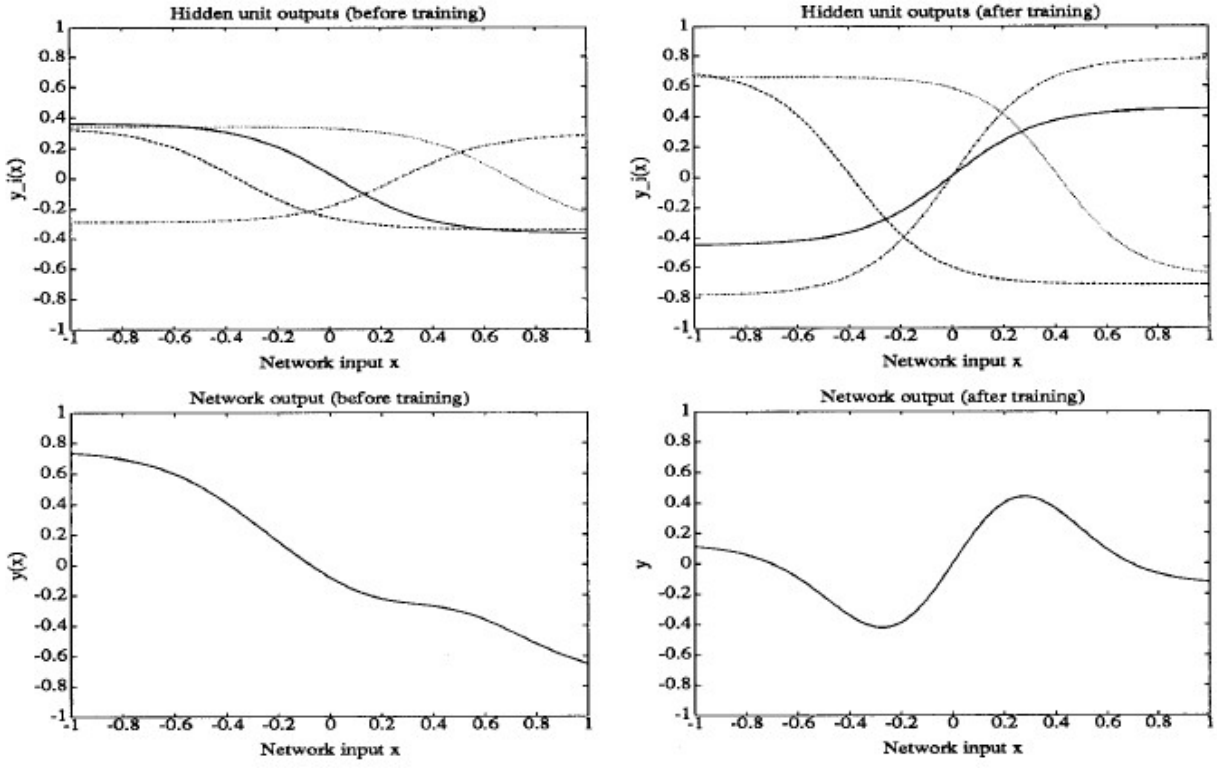
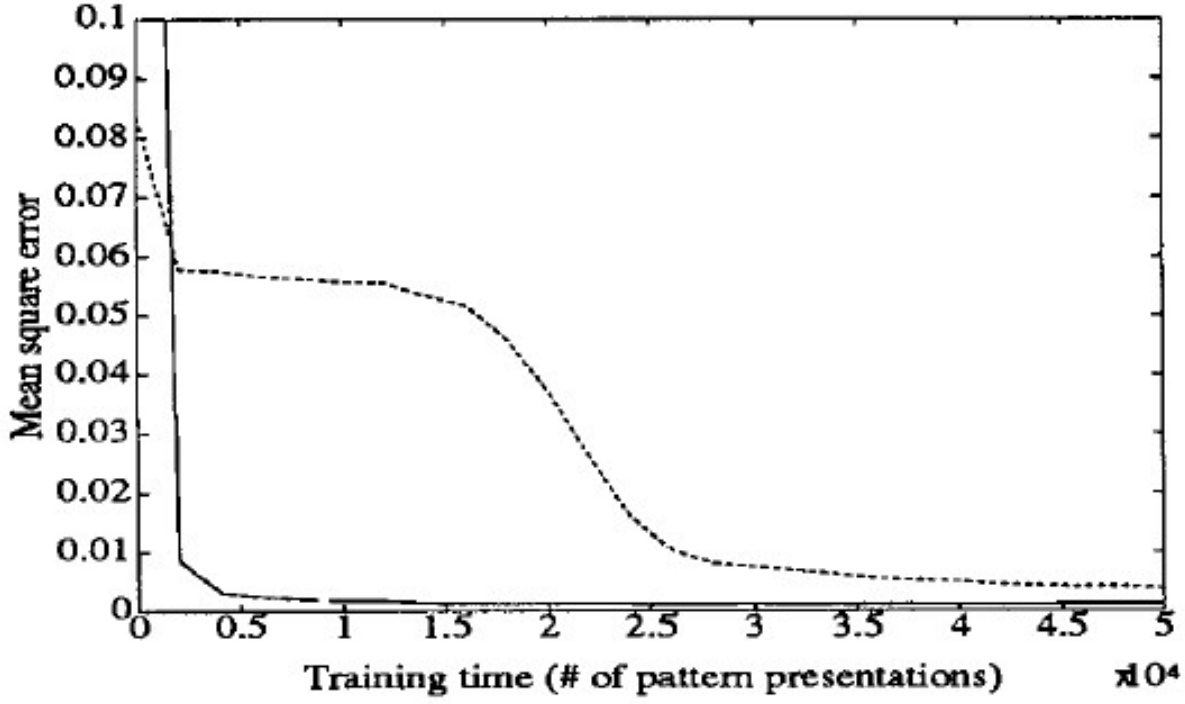


Figure 3: Outputs of network and hidden units before and after training with weight initialized by method described in text


0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Mean square error
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Training time (# of pattern presentations)
x10⁴

- En múltiples dimensiones:

$$y = \sum_{i=0}^{N-1} v_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle + w_{0i})$$

- En múltiples dimensiones:

$$y = \sum_{i=0}^{N-1} v_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle + w_{0i}) = \sum_{i=0}^{N-1} y_i(\mathbf{x})$$

- En múltiples dimensiones:

$$y = \sum_{i=0}^{N-1} v_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle + w_{0i}) = \sum_{i=0}^{N-1} y_i(\mathbf{x})$$

- Consideramos la transformada de Fourier: $y_i(\mathbf{x}) \leftrightarrow Y(\boldsymbol{\omega})$:

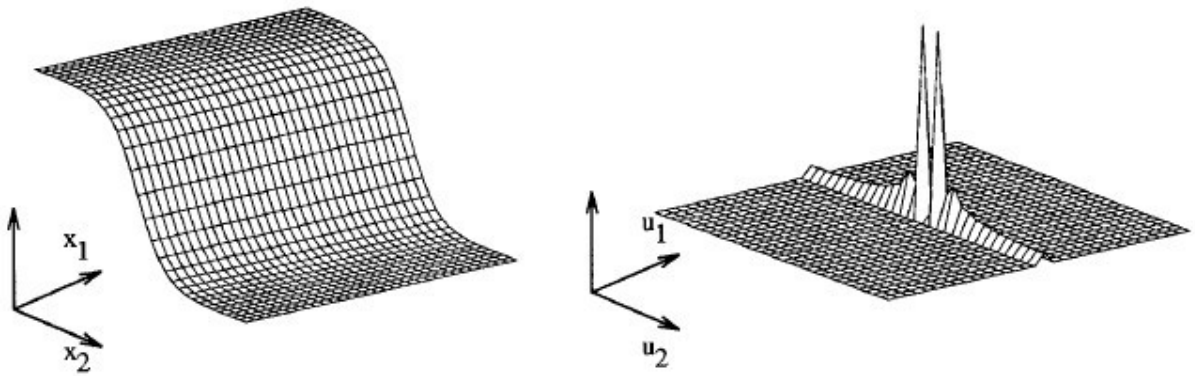


Figure 5: A $y_i(X)$ and its 2-D Fourier transform

- En múltiples dimensiones:

$$y = \sum_{i=0}^{N-1} v_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle + w_{0i}) = \sum_{i=0}^{N-1} y_i(\mathbf{x})$$

- Consideramos la transformada de Fourier: $y_i(\mathbf{x}) \leftrightarrow Y(\boldsymbol{\omega})$:

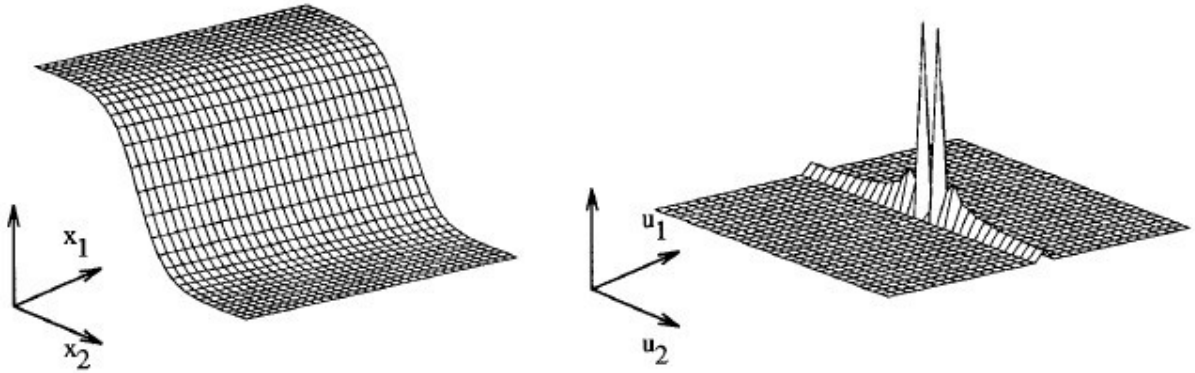


Figure 5: A $y_i(X)$ and its 2-D Fourier transform

- Dirección determinada por $\mathbf{w}_i$.

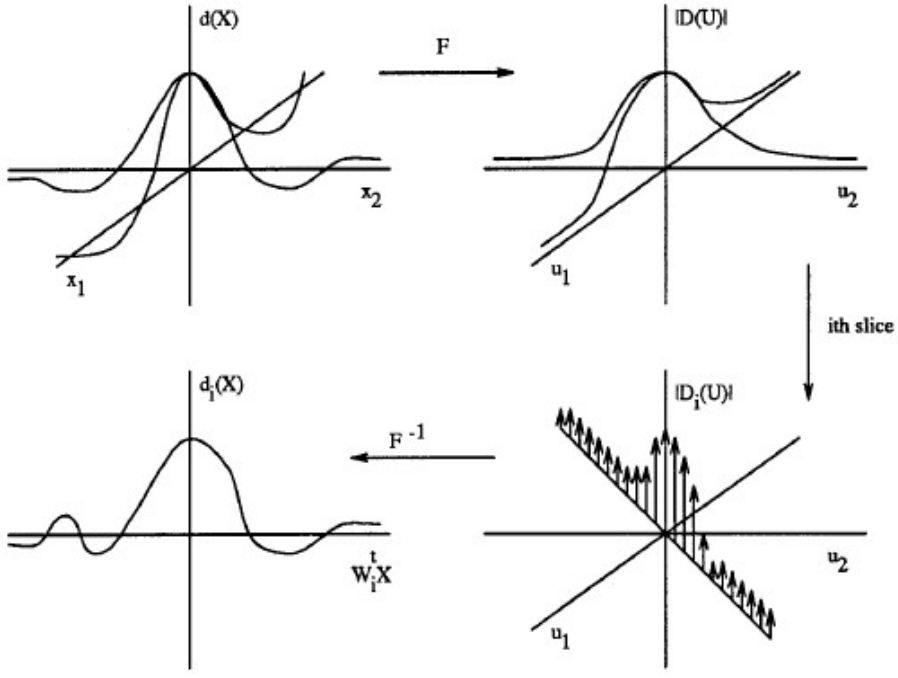


Figure 6: $d(x)$, its Fourier transform $D(U)$, a slice $D_i(U)$ of $D(U)$, and the inverse transform $d_i(X)$ of $D_i(U)$

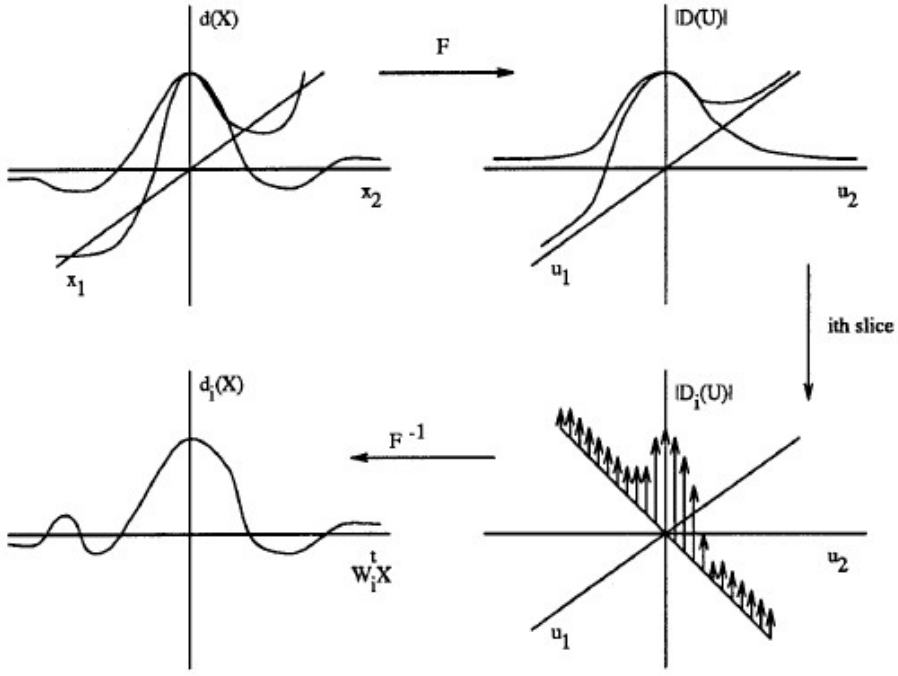


Figure 6: $d(x)$, its Fourier transform $D(U)$, a slice $D_i(U)$ of $D(U)$, and the inverse transform $d_i(X)$ of $D_i(U)$

- Asociar neuronas a tajadas de la transformada de Fourier de $d(\mathbf{x})$.

- Con S tajadas e I intervalos por tajada:

$$N = S \times I$$

- Con S tajadas e I intervalos por tajada:

$$N = S \times I = I^{d-1} \times I = I^d$$

- Con S tajadas e I intervalos por tajada:

$$N = S \times I = I^{d-1} \times I = I^d$$

- Para $\mathbf{x} \in [-1,\ 1]^d$,

- Con S tajadas e I intervalos por tajada:

$$N = S \times I = I^{d-1} \times I = I^d$$

- Para $\mathbf{x} \in [-1,\ 1]^d$,
  1. $\mathbf{w}_i \sim U[-1,\ 1]^d$ (direcciones aleatorias).

- Con S tajadas e I intervalos por tajada:

$$N = S \times I = I^{d-1} \times I = I^d$$

- Para $\mathbf{x} \in [-1,\ 1]^d$,
  1. $\mathbf{w}_i \sim U[-1,\ 1]^d$ (direcciones aleatorias).
  2. Normalizar $\|\mathbf{w}_i\| = 0{,}7I = 0{,}7N^{1/d}$.

- Con S tajadas e I intervalos por tajada:

$$N = S \times I = I^{d-1} \times I = I^d$$

- Para $\mathbf{x} \in [-1,\ 1]^d$,
  1. $\mathbf{w}_i \sim U[-1,\ 1]^d$ (direcciones aleatorias).
  2. Normalizar $\|\mathbf{w}_i\| = 0{,}7I = 0{,}7N^{1/d}$.
  3. $w_{0i} \sim U[-\|\mathbf{w}_i\|,\ \|\mathbf{w}_i\|]$.

- Con S tajadas e I intervalos por tajada:

$$N = S \times I = I^{d-1} \times I = I^d$$

- Para $\mathbf{x} \in [-1,\ 1]^d$,
  1. $\mathbf{w}_i \sim U[-1,\ 1]^d$ (direcciones aleatorias).
  2. Normalizar $\|\mathbf{w}_i\| = 0{,}7I = 0{,}7N^{1/d}$.
  3. $w_{0i} \sim U[-\|\mathbf{w}_i\|,\ \|\mathbf{w}_i\|]$.
- En la práctica se debe escalizar $\mathbf{x}$.

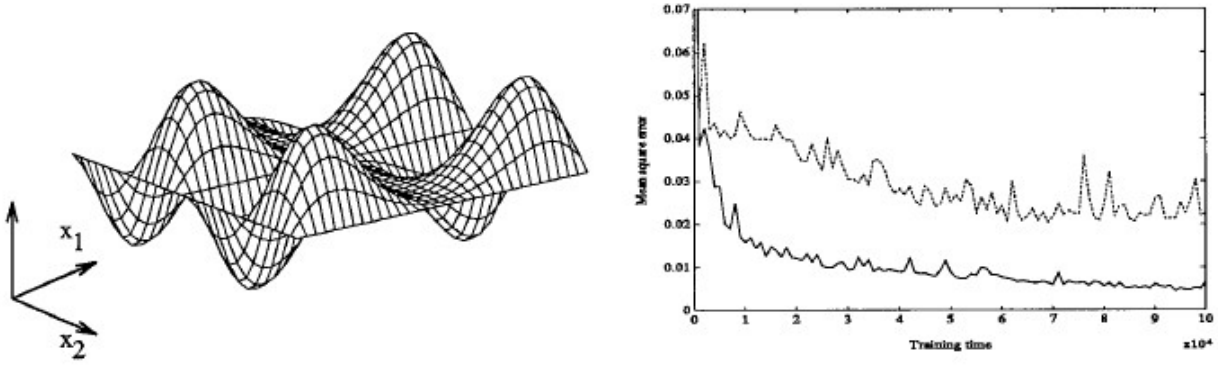


Figure 7: A 2-D desired function and the learning curves that resulted in training a neural net to approximate it. The solid curve is due to the training of a net initialized as described in the text. The dashed curve is due to a net whose weights are initialized to random values between -0.5 and 0.5

Escogencia de la tasa de aprendizaje

- Consideramos la función:

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \mathbf{Q}(\mathbf{w} - \mathbf{w}^*), \qquad \mathbf{Q} > 0$$

Escogencia de la tasa de aprendizaje

- Consideramos la función:

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \mathbf{Q}(\mathbf{w} - \mathbf{w}^*), \qquad \mathbf{Q} > 0$$

(buena aproximación cerca al mínimo local $\mathbf{w}^*$, si suponemos $E(\mathbf{w}^*) = 0$)

Escogencia de la tasa de aprendizaje

- Consideramos la función:

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \mathbf{Q}(\mathbf{w} - \mathbf{w}^*), \qquad \mathbf{Q} > 0$$

  (buena aproximación cerca al mínimo local $\mathbf{w}^*$, si suponemos $E(\mathbf{w}^*) = 0$)

- Iteración de Backprop (batch):

$$\Delta \mathbf{w}_k = \mathbf{w}_k - \mathbf{w}_{k-1} = -\mu \nabla E(\mathbf{w}_{k-1})$$

Escogencia de la tasa de aprendizaje

- Consideramos la función:

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \mathbf{Q}(\mathbf{w} - \mathbf{w}^*), \qquad \mathbf{Q} > 0$$

(buena aproximación cerca al mínimo local $\mathbf{w}^*$, si suponemos $E(\mathbf{w}^*) = 0$)

- Iteración de Backprop (batch):

$$\Delta \mathbf{w}_k = \mathbf{w}_k - \mathbf{w}_{k-1} = -\mu \nabla E(\mathbf{w}_{k-1})$$

- En la aproximación cuadrática:

$$\Delta \mathbf{w}_k = -\mu \mathbf{Q}(\mathbf{w}_{k-1} - \mathbf{w}^*)$$

- Suponga que $\mathbf{Q}$ tiene valores propios λ_i y vectores propios correspondientes $\mathbf{u}_i$:

- Suponga que $\mathbf{Q}$ tiene valores propios λ_i y vectores propios correspondientes $\mathbf{u}_i$:

$$\mathbf{w} - \mathbf{w}^* = \sum_i \alpha_i \mathbf{u}_i$$

- Suponga que $\mathbf{Q}$ tiene valores propios λ_i y vectores propios correspondientes $\mathbf{u}_i$:

$$\mathbf{w} - \mathbf{w}^* = \sum_i \alpha_i \mathbf{u}_i$$

donde $\alpha_i =$

- Suponga que $\mathbf{Q}$ tiene valores propios λ_i y vectores propios correspondientes $\mathbf{u}_i$:

$$\mathbf{w} - \mathbf{w}^* = \sum_i \alpha_i \mathbf{u}_i$$

donde $\alpha_i = \langle \mathbf{w} - \mathbf{w}^*, \mathbf{u}_i \rangle$.

- Suponga que $\mathbf{Q}$ tiene valores propios λ_i y vectores propios correspondientes $\mathbf{u}_i$:

$$\mathbf{w} - \mathbf{w}^* = \sum_i \alpha_i \mathbf{u}_i$$

donde $\alpha_i = \langle \mathbf{w} - \mathbf{w}^*, \mathbf{u}_i \rangle$.

- Luego

$$\Delta \mathbf{w}_k = (\mathbf{w}_k - \mathbf{w}^*) - (\mathbf{w}_{k-1} - \mathbf{w}^*)$$

- Suponga que $\mathbf{Q}$ tiene valores propios λ_i y vectores propios correspondientes $\mathbf{u}_i$:

$$\mathbf{w} - \mathbf{w}^* = \sum_i \alpha_i \mathbf{u}_i$$

donde $\alpha_i = \langle \mathbf{w} - \mathbf{w}^*, \mathbf{u}_i \rangle$.

- Luego

$$\Delta \mathbf{w}_k = (\mathbf{w}_k - \mathbf{w}^*) - (\mathbf{w}_{k-1} - \mathbf{w}^*) = \sum_i (\alpha_i^{(k)} - \alpha_i^{(k-1)}) \mathbf{u}_i$$

- Suponga que $\mathbf{Q}$ tiene valores propios λ_i y vectores propios correspondientes $\mathbf{u}_i$:

$$\mathbf{w} - \mathbf{w}^* = \sum_i \alpha_i \mathbf{u}_i$$

donde $\alpha_i = \langle \mathbf{w} - \mathbf{w}^*, \mathbf{u}_i \rangle$.

- Luego

$$\Delta \mathbf{w}_k = (\mathbf{w}_k - \mathbf{w}^*) - (\mathbf{w}_{k-1} - \mathbf{w}^*) = \sum_i (\alpha_i^{(k)} - \alpha_i^{(k-1)}) \mathbf{u}_i$$

- o en términos del gradiente:

$$\Delta \mathbf{w}_k = -\mu \mathbf{Q} (\mathbf{w}_{k-1} - \mathbf{w}^*)$$

- Suponga que $\mathbf{Q}$ tiene valores propios λ_i y vectores propios correspondientes $\mathbf{u}_i$:

$$\mathbf{w} - \mathbf{w}^* = \sum_i \alpha_i \mathbf{u}_i$$

donde $\alpha_i = \langle \mathbf{w} - \mathbf{w}^*, \mathbf{u}_i \rangle$.

- Luego

$$\Delta \mathbf{w}_k = (\mathbf{w}_k - \mathbf{w}^*) - (\mathbf{w}_{k-1} - \mathbf{w}^*) = \sum_i (\alpha_i^{(k)} - \alpha_i^{(k-1)}) \mathbf{u}_i$$

- o en términos del gradiente:

$$\Delta \mathbf{w}_k = -\mu \mathbf{Q}(\mathbf{w}_{k-1} - \mathbf{w}^*) = -\mu \sum_i \lambda_i \alpha_i^{(k-1)} \mathbf{u}_i$$

- Suponga que $\mathbf{Q}$ tiene valores propios λ_i y vectores propios correspondientes $\mathbf{u}_i$:

$$\mathbf{w} - \mathbf{w}^* = \sum_i \alpha_i \mathbf{u}_i$$

donde $\alpha_i = \langle \mathbf{w} - \mathbf{w}^*, \mathbf{u}_i \rangle$.

- Luego

$$\Delta \mathbf{w}_k = (\mathbf{w}_k - \mathbf{w}^*) - (\mathbf{w}_{k-1} - \mathbf{w}^*) = \sum_i (\alpha_i^{(k)} - \alpha_i^{(k-1)}) \mathbf{u}_i$$

- o en términos del gradiente:

$$\Delta \mathbf{w}_k = -\mu \mathbf{Q}(\mathbf{w}_{k-1} - \mathbf{w}^*) = -\mu \sum_i \lambda_i \alpha_i^{(k-1)} \mathbf{u}_i$$

- Comparando:

$$\alpha_i^{(k)} = (1 - \mu\lambda_i)\alpha_i^{(k-1)}$$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow$$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to$$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to \mathbf{w}^*$$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to \mathbf{w}^*$$

- $\mu \Uparrow \Rightarrow$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to \mathbf{w}^*$$

- $\mu \uparrow\uparrow \Rightarrow$ convergencia más rápida.

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to \mathbf{w}^*$$

- $\mu \uparrow\uparrow \Rightarrow$ convergencia más rápida.
- Valor máximo?

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to \mathbf{w}^*$$

- $\mu \uparrow\uparrow \Rightarrow$ convergencia más rápida.
- Valor máximo?

$$|1 - \mu\lambda_i| < 1$$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to \mathbf{w}^*$$

- $\mu \uparrow\uparrow \Rightarrow$ convergencia más rápida.
- Valor máximo?

$$|1 - \mu\lambda_i| < 1 \Rightarrow \mu < \frac{2}{\lambda_{\text{máx}}}$$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to \mathbf{w}^*$$

- $\mu \uparrow\uparrow \Rightarrow$ convergencia más rápida.
- Valor máximo?

$$|1 - \mu\lambda_i| < 1 \Rightarrow \mu < \frac{2}{\lambda_{\text{máx}}}$$

- Con $\mu \approx \frac{2}{\lambda_{\text{máx}}}$, convergencia es governada por:

$$\left(1 - 2\frac{\lambda_{\text{mín}}}{\lambda_{\text{máx}}}\right) = \left(1 - \frac{2}{\kappa}\right)$$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to \mathbf{w}^*$$

- $\mu \uparrow\uparrow \Rightarrow$ convergencia más rápida.
- Valor máximo?

$$|1 - \mu\lambda_i| < 1 \Rightarrow \mu < \frac{2}{\lambda_{\text{máx}}}$$

- Con $\mu \approx \frac{2}{\lambda_{\text{máx}}}$, convergencia es governada por:

$$\left(1 - 2\frac{\lambda_{\text{mín}}}{\lambda_{\text{máx}}}\right) = \left(1 - \frac{2}{\kappa}\right)$$

- $\kappa \uparrow\uparrow \Rightarrow$

- En T iteraciones:

$$\alpha_i^{(T)} = (1 - \mu\lambda_i)^T \alpha_i^{(0)}$$

- Si garantizamos $|1 - \mu\lambda_i| < 1$, tenemos que cuando $T \to \infty$,

$$\alpha_i^{(T)} \to 0 \Rightarrow \mathbf{w}^T \to \mathbf{w}^*$$

- $\mu \uparrow\uparrow \Rightarrow$ convergencia más rápida.
- Valor máximo?

$$|1 - \mu\lambda_i| < 1 \Rightarrow \mu < \frac{2}{\lambda_{\text{máx}}}$$

- Con $\mu \approx \frac{2}{\lambda_{\text{máx}}}$, convergencia es governada por:

$$\left(1 - 2\frac{\lambda_{\text{mín}}}{\lambda_{\text{máx}}}\right) = \left(1 - \frac{2}{\kappa}\right)$$

- $\kappa \uparrow\uparrow \Rightarrow$convergencia puede ser muy lenta!

Variaciones de Backpropagation

- Heurísticas:

Variaciones de Backpropagation

- Heurísticas:
  - Momentum.

Variaciones de Backpropagation

- Heurísticas:
  - Momentum.
  - Tasa de aprendizaje variable.

Variaciones de Backpropagation

- Heurísticas:
  - Momentum.
  - Tasa de aprendizaje variable.
  - Backpropagation resistente (resilient).

Variaciones de Backpropagation

- Heurísticas:
  - Momentum.
  - Tasa de aprendizaje variable.
  - Backpropagation resistente (resilient).
- Técnicas de Optimización:

Variaciones de Backpropagation

- Heurísticas:
  - Momentum.
  - Tasa de aprendizaje variable.
  - Backpropagation resistente (resilient).
- Técnicas de Optimización:
  - Dirección de búsqueda

Variaciones de Backpropagation

- Heurísticas:
  - Momentum.
  - Tasa de aprendizaje variable.
  - Backpropagation resistente (resilient).
- Técnicas de Optimización:
  - Dirección de búsqueda
    - Quasi Newton

Variaciones de Backpropagation

- Heurísticas:
  - Momentum.
  - Tasa de aprendizaje variable.
  - Backpropagation resistente (resilient).
- Técnicas de Optimización:
  - Dirección de búsqueda
    - Quasi Newton
    - Gradiente Conjugado

Variaciones de Backpropagation

- Heurísticas:
  - Momentum.
  - Tasa de aprendizaje variable.
  - Backpropagation resistente (resilient).
- Técnicas de Optimización:
  - Dirección de búsqueda
    - Quasi Newton
    - Gradiente Conjugado
  - Técnicas de búsqueda de línea.

Momentum

- Método para evitar caer en un mínimo local espúreo.

Momentum

- Método para evitar caer en un mínimo local espúreo.
- Tiene en cuenta información de gradiente local más la tendencia reciente en la superficie de error:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \nabla_{\mathbf{w}} E|_{\mathbf{w}_k} + \eta(\mathbf{w}_{k-1} - \mathbf{w}_{k-2})$$

Momentum

- Método para evitar caer en un mínimo local espúreo.
- Tiene en cuenta información de gradiente local más la tendencia reciente en la superficie de error:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \nabla_{\mathbf{w}} E|_{\mathbf{w}_k} + \eta(\mathbf{w}_{k-1} - \mathbf{w}_{k-2})$$

- En una región de la superficie de error con poca curvatura, y gradiente aproximadamente constante:

$$\begin{aligned} \Delta \mathbf{w} &\approx -\mu \nabla_{\mathbf{w}} E(1 + \eta + \eta^2 + \dots) \\ &= -\frac{\mu}{1-\eta} \nabla_{\mathbf{w}} E \end{aligned}$$

Momentum

- Método para evitar caer en un mínimo local espúreo.
- Tiene en cuenta información de gradiente local más la tendencia reciente en la superficie de error:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \nabla_{\mathbf{w}} E|_{\mathbf{w}_k} + \eta(\mathbf{w}_{k-1} - \mathbf{w}_{k-2})$$

- En una región de la superficie de error con poca curvatura, y gradiente aproximadamente constante:

$$\begin{aligned} \Delta \mathbf{w} &\approx -\mu \nabla_{\mathbf{w}} E(1 + \eta + \eta^2 + \dots) \\ &= -\frac{\mu}{1-\eta} \nabla_{\mathbf{w}} E \end{aligned}$$

- En cambio, en una región de curvatura grande en la cual el descenso de gradiente es oscilatorio, las contribuciones sucesivas del término de momentum tienden a cancelarse, resultando en una tasa de aprendizaje efectiva cercana a μ.

Tasa de aprendizaje variable

- El comportamiento del método de gradiente es altamente sensitivo al valor de la tasa de aprendizaje:

Tasa de aprendizaje variable

- El comportamiento del método de gradiente es altamente sensitivo al valor de la tasa de aprendizaje:
  - Si es muy alta, puede causar oscilaciones e inestabilidad.

Tasa de aprendizaje variable

- El comportamiento del método de gradiente es altamente sensitivo al valor de la tasa de aprendizaje:
  - Si es muy alta, puede causar oscilaciones e inestabilidad.
  - Si es muy pequeña, el aprendizaje es muy lento.

Tasa de aprendizaje variable

- El comportamiento del método de gradiente es altamente sensitivo al valor de la tasa de aprendizaje:
  - Si es muy alta, puede causar oscilaciones e inestabilidad.
  - Si es muy pequeña, el aprendizaje es muy lento.
- La tasa de aprendizaje óptima cambia durante el proceso de entrenamiento.

Tasa de aprendizaje variable

- El comportamiento del método de gradiente es altamente sensitivo al valor de la tasa de aprendizaje:
  - Si es muy alta, puede causar oscilaciones e inestabilidad.
  - Si es muy pequeña, el aprendizaje es muy lento.
- La tasa de aprendizaje óptima cambia durante el proceso de entrenamiento.
- En cada paso la tasa de aprendizaje se modifica:

Tasa de aprendizaje variable

- El comportamiento del método de gradiente es altamente sensitivo al valor de la tasa de aprendizaje:
  - Si es muy alta, puede causar oscilaciones e inestabilidad.
  - Si es muy pequeña, el aprendizaje es muy lento.
- La tasa de aprendizaje óptima cambia durante el proceso de entrenamiento.
- En cada paso la tasa de aprendizaje se modifica:
  - Si $\frac{E_{k+1}}{E_k} > \beta$ (típicamente $\beta = 1{,}04$) los nuevos pesos se descartan, y la tasa de aprendizaje se reduce a $\mu_{nueva} = \alpha * \mu_{vieja}$ (típicamente $\alpha = 0{,}7$).

Tasa de aprendizaje variable

- El comportamiento del método de gradiente es altamente sensitivo al valor de la tasa de aprendizaje:
  - Si es muy alta, puede causar oscilaciones e inestabilidad.
  - Si es muy pequeña, el aprendizaje es muy lento.
- La tasa de aprendizaje óptima cambia durante el proceso de entrenamiento.
- En cada paso la tasa de aprendizaje se modifica:
  - Si $\frac{E_{k+1}}{E_k} > \beta$ (típicamente $\beta = 1{,}04$) los nuevos pesos se descartan, y la tasa de aprendizaje se reduce a $\mu_{nueva} = \alpha * \mu_{vieja}$ (típicamente $\alpha = 0{,}7$).
  - Si $E_{k+1} < E_k$ la tasa de aprendizaje se incrementa a $\mu_{nueva} = \gamma * \mu_{vieja}$ (típicamente $\gamma = 1{,}05$).

Tasa de aprendizaje variable

- El comportamiento del método de gradiente es altamente sensitivo al valor de la tasa de aprendizaje:
  - Si es muy alta, puede causar oscilaciones e inestabilidad.
  - Si es muy pequeña, el aprendizaje es muy lento.
- La tasa de aprendizaje óptima cambia durante el proceso de entrenamiento.
- En cada paso la tasa de aprendizaje se modifica:
  - Si $\frac{E_{k+1}}{E_k} > \beta$ (típicamente $\beta = 1{,}04$) los nuevos pesos se descartan, y la tasa de aprendizaje se reduce a $\mu_{nueva} = \alpha * \mu_{vieja}$ (típicamente $\alpha = 0{,}7$).
  - Si $E_{k+1} < E_k$ la tasa de aprendizaje se incrementa a $\mu_{nueva} = \gamma * \mu_{vieja}$ (típicamente $\gamma = 1{,}05$).
- Esta técnica se conoce como bold driver(chofer atrevido).

Resilient Backpropagation

- La pendiente de la función de activación sigmoidal se acerca a cero cuando la entrada se hace grande.

Resilient Backpropagation

- La pendiente de la función de activación sigmoidal se acerca a cero cuando la entrada se hace grande.
- Esto hace que en ciertas regiones de la superficie de error el gradiente tenga magnitud muy pequeña, aunque los pesos estén lejos de sus valores óptimos.

Resilient Backpropagation

- La pendiente de la función de activación sigmoidal se acerca a cero cuando la entrada se hace grande.
- Esto hace que en ciertas regiones de la superficie de error el gradiente tenga magnitud muy pequeña, aunque los pesos estén lejos de sus valores óptimos.
- En estas regiones el progreso del procedimiento de descenso de gradiente se hace muy lento.

Resilient Backpropagation

- La pendiente de la función de activación sigmoidal se acerca a cero cuando la entrada se hace grande.
- Esto hace que en ciertas regiones de la superficie de error el gradiente tenga magnitud muy pequeña, aunque los pesos estén lejos de sus valores óptimos.
- En estas regiones el progreso del procedimiento de descenso de gradiente se hace muy lento.
- En Resilient Backpropagation (Rprop) se intenta eliminar este efecto.

Resilient Backpropagation

- La pendiente de la función de activación sigmoidal se acerca a cero cuando la entrada se hace grande.
- Esto hace que en ciertas regiones de la superficie de error el gradiente tenga magnitud muy pequeña, aunque los pesos estén lejos de sus valores óptimos.
- En estas regiones el progreso del procedimiento de descenso de gradiente se hace muy lento.
- En Resilient Backpropagation (Rprop) se intenta eliminar este efecto.
- En la actualización de los pesos se utiliza únicamente el signo de las derivadas de primer orden.

Resilient Backpropagation

- La pendiente de la función de activación sigmoidal se acerca a cero cuando la entrada se hace grande.
- Esto hace que en ciertas regiones de la superficie de error el gradiente tenga magnitud muy pequeña, aunque los pesos estén lejos de sus valores óptimos.
- En estas regiones el progreso del procedimiento de descenso de gradiente se hace muy lento.
- En Resilient Backpropagation (Rprop) se intenta eliminar este efecto.
- En la actualización de los pesos se utiliza únicamente el signo de las derivadas de primer orden.
- El valor del cambio de cada peso es determinado por un valor separado de actualización.

Resilient Backpropagation

- La pendiente de la función de activación sigmoidal se acerca a cero cuando la entrada se hace grande.
- Esto hace que en ciertas regiones de la superficie de error el gradiente tenga magnitud muy pequeña, aunque los pesos estén lejos de sus valores óptimos.
- En estas regiones el progreso del procedimiento de descenso de gradiente se hace muy lento.
- En Resilient Backpropagation (Rprop) se intenta eliminar este efecto.
- En la actualización de los pesos se utiliza únicamente el signo de las derivadas de primer orden.
- El valor del cambio de cada peso es determinado por un valor separado de actualización.

- Este valor de actualización se modifica en cada iteración de la siguiente forma:

- Este valor de actualización se modifica en cada iteración de la siguiente forma:
  - Se incrementa por un valor ρ_{inc} cuando la derivada del error con respecto a ese peso tiene el mismo signo en dos iteraciones sucesivas

- Este valor de actualización se modifica en cada iteración de la siguiente forma:
  - Se incrementa por un valor ρ_{inc} cuando la derivada del error con respecto a ese peso tiene el mismo signo en dos iteraciones sucesivas.
  - Se decrementa por un valor ρ_{dec} cuando la derivada del error con respecto a ese peso cambia en iteraciones sucesivas.

- Este valor de actualización se modifica en cada iteración de la siguiente forma:
  - Se incrementa por un valor ρ_{inc} cuando la derivada del error con respecto a ese peso tiene el mismo signo en dos iteraciones sucesivas.
  - Se decrementa por un valor ρ_{dec} cuando la derivada del error con respecto a ese peso cambia en iteraciones sucesivas.
  - Si la derivada es cero, el valor de actualización no cambia

- Este valor de actualización se modifica en cada iteración de la siguiente forma:
  - Se incrementa por un valor ρ_{inc} cuando la derivada del error con respecto a ese peso tiene el mismo signo en dos iteraciones sucesivas.
  - Se decrementa por un valor ρ_{dec} cuando la derivada del error con respecto a ese peso cambia en iteraciones sucesivas.
  - Si la derivada es cero, el valor de actualización no cambia.
  - Si el peso continúa cambiando en la misma dirección por varias iteraciones,ρ_{inc} se incrementa.