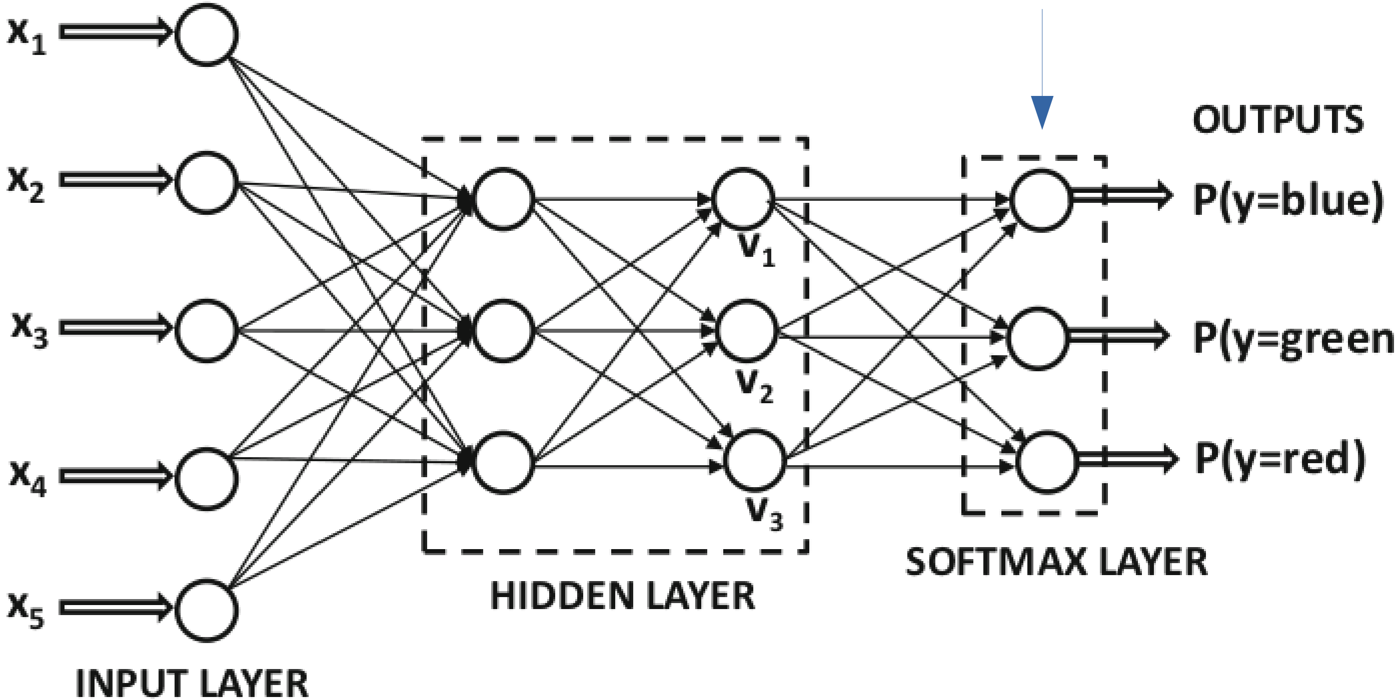
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

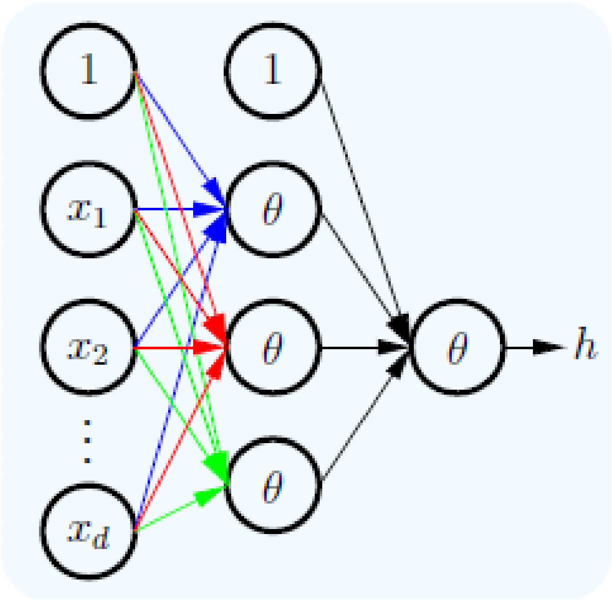
Minería de Datos

Extensión de MLP a *K* clases

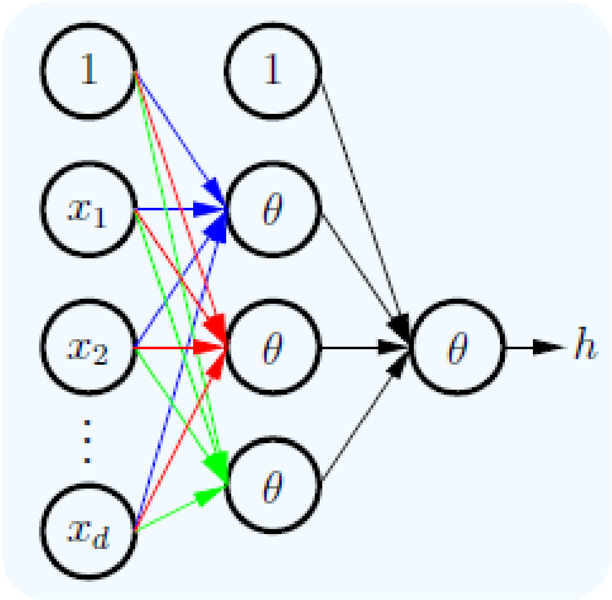
activación
logística



Funciones aproximadas por una red feed-forward

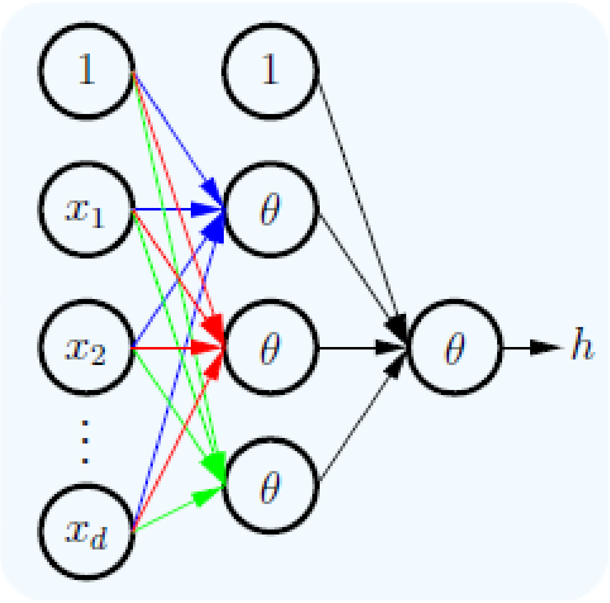


Funciones aproximadas por una red feed-forward



$$h(\mathbf{x}) = \theta\left(w_{01}^{(2)} + \sum_{j=1}^{m} w_{j1}^{(2)}\theta\left(\sum_{i=0}^{d} w_{ij}^{(1)} x_i\right)\right)$$

Funciones aproximadas por una red feed-forward



$$h(\mathbf{x}) = \theta\left(w_{01}^{(2)} + \sum_{j=1}^{m} w_{j1}^{(2)}\theta\left(\sum_{i=0}^{d} w_{ij}^{(1)} x_i\right)\right)$$

Se usa también la siguiente notación (simplificada):

$$\mathrm{W}^{(1)}$$

$$h(\mathbf{x}) = \theta\left(w_0 + \sum_{j=1}^{m} w_j \theta\left(\mathbf{v}_j^{\mathrm{T}}\mathbf{x}\right)\right)$$

Y para dimensionalidad: $d^{(1)} = m$

Capacidad de una red feed-forward

Para una cantidad de neuronas suficientemente grande, podemos lograr:

$$E_{\text{in}} \sim 0$$

Sin embargo, no podemos asegurar que $E_{\text{in}} \approx E_{\text{out}}$.

Capacidad de una red feed-forward

Para una cantidad de neuronas suficientemente grande, podemos lograr:

$$E_{\text{in}} \sim 0$$

Sin embargo, no podemos asegurar que $E_{\text{in}} \approx E_{\text{out}}$.

Se puede mostrar que para esta red:

$$h(\mathbf{x}) = \theta\left(w_{01}^{(2)} + \sum_{j=1}^{m} w_{j1}^{(2)} \theta\left(\sum_{i=0}^{d} w_{ij}^{(1)} x_i\right)\right)$$

la dimensión VC está acotada por:

$$d_{\text{VC}} \leq (\text{const}) \cdot md \log(md).$$

si la activación es la función signo en ambas capas.

Capacidad de una red feed-forward

Si la red usa estas funciones de activación:

$$h(\mathbf{x}) = \theta \left(w_{01}^{(2)} + \sum_{j=1}^{m} w_{j1}^{(2)} \theta \left(\sum_{i=0}^{d} w_{ij}^{(1)} x_i \right) \right)$$

$\mathrm{sign}(x)$ $\quad\quad$ $\tanh(\cdot)$

la dimensión VC es:

$$d_{\mathrm{VC}} = O(md(m+d)).$$

¿Cuál de las dos redes tiene mayor expresividad?

Capacidad de una red feed-forward

Si la red usa estas funciones de activación:

$$h(\mathbf{x}) = \theta\left(w_{01}^{(2)} + \sum_{j=1}^{m} w_{j1}^{(2)}\theta\left(\sum_{i=0}^{d} w_{ij}^{(1)} x_i\right)\right)$$

$\mathrm{sign}(x)$ $\qquad$ $\tanh(\cdot)$

la dimensión VC es:

$$d_{\mathrm{VC}} = O(md(m+d)).$$

¿Cuál de las dos redes tiene mayor expresividad?

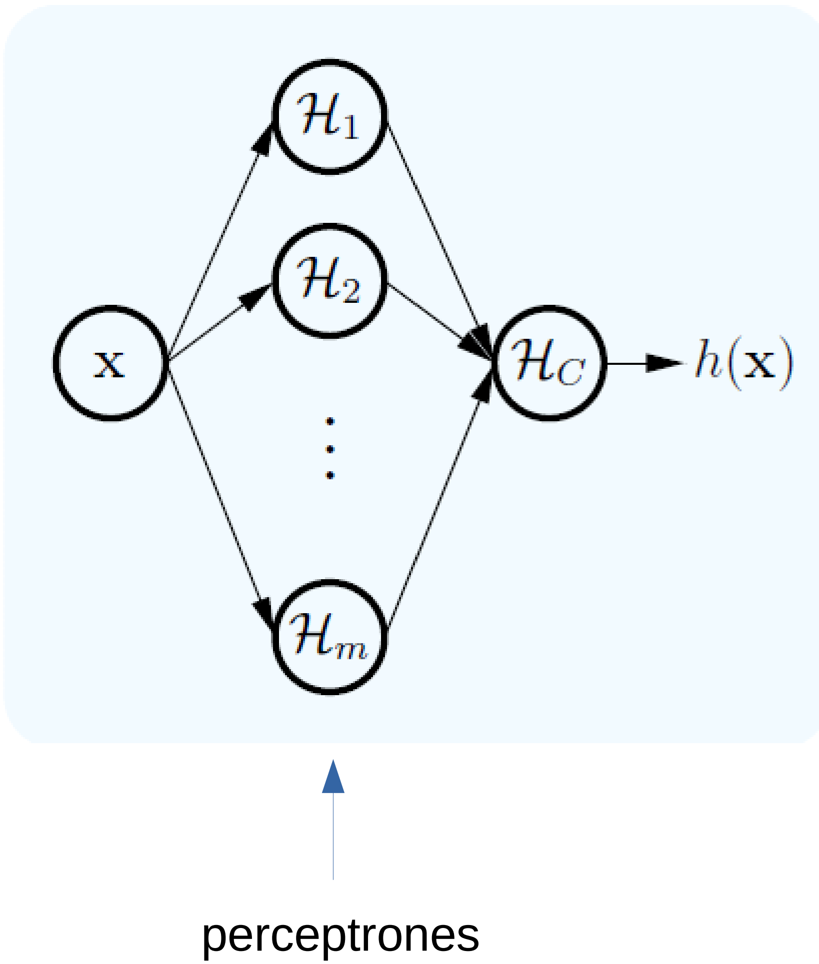
$$md\log(md) \leq \ md(m+d)$$

Usar $\tanh(\cdot)$ aumenta la expresividad de la red.

Capacidad de una red feed-forward

¿De dónde viene $d_{\mathrm{VC}} \leq (\mathrm{const}) \cdot md \log(md).$?

Consideremos el siguiente conjunto de hipótesis:

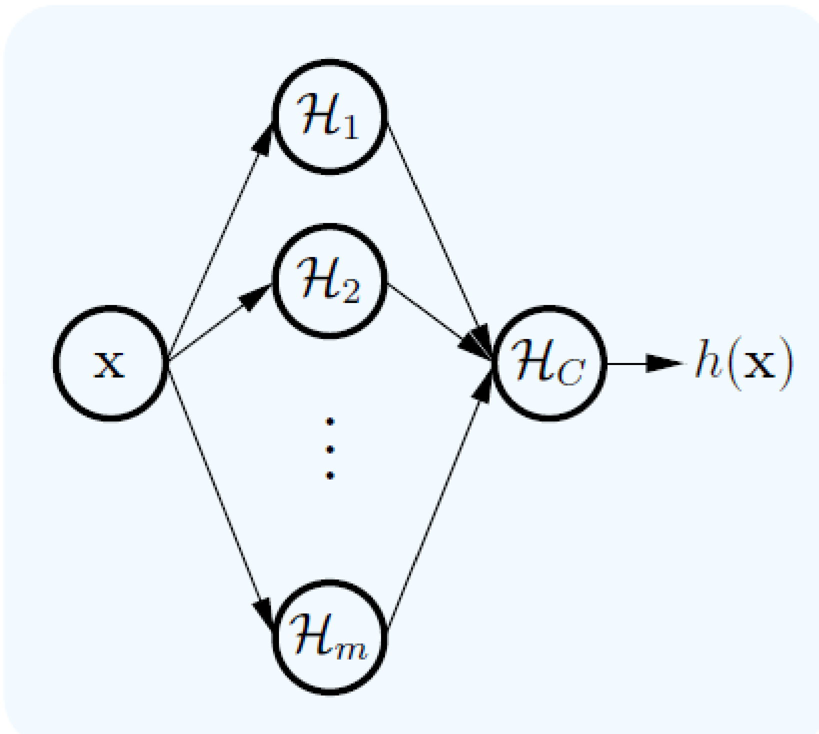


$$h(\mathbf{x}) = h_C(h_1(\mathbf{x}), \ldots, h_m(\mathbf{x}))$$

Capacidad de una red feed-forward

¿De dónde viene $d_{\mathrm{VC}} \leq (\mathrm{const}) \cdot md\log(md).$?

Consideremos el siguiente conjunto de hipótesis:



$$h(\mathbf{x}) = h_C(h_1(\mathbf{x}), \ldots, h_m(\mathbf{x}))$$

perceptrones

Supongamos que la dimensión VC de $\mathcal{H}_i$ es d_i y la de $\mathcal{H}_C$ es d_c .

Capacidad de una red feed-forward

Fijemos $\mathbf{x}_1, \ldots, \mathbf{x}_N$ y $h_1, \ldots, h_m$. Las hipótesis son ahora funciones base que definen una transformación hacia $\mathbb{R}^m$,

$$\mathbf{x}_1 \rightarrow \mathbf{z}_1 = \begin{bmatrix} h_1(\mathbf{x}_1) \\ \vdots \\ h_m(\mathbf{x}_1) \end{bmatrix} \quad \cdots \quad \rightarrow \mathbf{z}_N = \begin{bmatrix} h_1(\mathbf{x}_N) \\ \vdots \\ h_m(\mathbf{x}_N) \end{bmatrix}.$$

Capacidad de una red feed-forward

Fijemos $\mathbf{x}_1, \ldots, \mathbf{x}_N$ y $h_1, \ldots, h_m$. Las hipótesis son ahora funciones base que definen una transformación hacia $\mathbb{R}^m$,

$$\mathbf{x}_1 \to \mathbf{z}_1 = \begin{bmatrix} h_1(\mathbf{x}_1) \\ \vdots \\ h_m(\mathbf{x}_1) \end{bmatrix} \quad \cdots \quad \to \mathbf{z}_N = \begin{bmatrix} h_1(\mathbf{x}_N) \\ \vdots \\ h_m(\mathbf{x}_N) \end{bmatrix}.$$

Los vectores z son vectores binarios en $\mathbb{R}^m$.

Dado que la red tiene flexibilidad para encontrar $h_1, \ldots, h_m$, podemos acotar por arriba el número posible de diferentes $\mathbf{z}_1, \ldots, \mathbf{z}_N$.

Capacidad de una red feed-forward

Fijemos $\mathbf{x}_1, \ldots, \mathbf{x}_N$ y $h_1, \ldots, h_m$. Las hipótesis son ahora funciones base que definen una transformación hacia $\mathbb{R}^m$,

$$\mathbf{x}_1 \rightarrow \mathbf{z}_1 = \begin{bmatrix} h_1(\mathbf{x}_1) \\ \vdots \\ h_m(\mathbf{x}_1) \end{bmatrix} \quad \cdots \quad \rightarrow \mathbf{z}_N = \begin{bmatrix} h_1(\mathbf{x}_N) \\ \vdots \\ h_m(\mathbf{x}_N) \end{bmatrix}.$$

Los vectores z son vectores binarios en $\mathbb{R}^m$.

Dado que la red tiene flexibilidad para encontrar $h_1, \ldots, h_m$, podemos acotar por arriba el número posible de diferentes $\mathbf{z}_1, \ldots, \mathbf{z}_N$.

Las primeras componentes de los vectores z están dadas por:

$$h_1(\mathbf{x}_1), \ldots, h_1(\mathbf{x}_N)$$

Capacidad de una red feed-forward

Fijemos $\mathbf{x}_1, \ldots, \mathbf{x}_N$ y $h_1, \ldots, h_m$. Las hipótesis son ahora funciones base que definen una transformación hacia $\mathbb{R}^m$,

$$\mathbf{x}_1 \rightarrow \mathbf{z}_1 = \begin{bmatrix} h_1(\mathbf{x}_1) \\ \vdots \\ h_m(\mathbf{x}_1) \end{bmatrix} \quad \cdots \quad \rightarrow \mathbf{z}_N = \begin{bmatrix} h_1(\mathbf{x}_N) \\ \vdots \\ h_m(\mathbf{x}_N) \end{bmatrix}.$$

Los vectores z son vectores binarios en $\mathbb{R}^m$.

Dado que la red tiene flexibilidad para encontrar $h_1, \ldots, h_m$, podemos acotar por arriba el número posible de diferentes $\mathbf{z}_1, \ldots, \mathbf{z}_N$.

Las primeras componentes de los vectores z están dadas por:

$$h_1(\mathbf{x}_1), \ldots, h_1(\mathbf{x}_N)$$

que es una dicotomía sobre $\mathbf{x}_1, \ldots, \mathbf{x}_N$ implementada por h_1 . Dado que la dimensión VC de $\mathcal{H}_1$ es d_1 , existen a lo más N^{d_1} dicotomías. Es decir, existen a lo más N^{d_1} formas de elegir las primeras componentes de z.

Capacidad de una red feed-forward

Fijemos $\mathbf{x}_1, \ldots, \mathbf{x}_N$ y $h_1, \ldots, h_m$. Las hipótesis son ahora funciones base que definen una transformación hacia $\mathbb{R}^m$,

$$\mathbf{x}_1 \to \mathbf{z}_1 = \begin{bmatrix} h_1(\mathbf{x}_1) \\ \vdots \\ h_m(\mathbf{x}_1) \end{bmatrix} \quad \cdots \quad \to \mathbf{z}_N = \begin{bmatrix} h_1(\mathbf{x}_N) \\ \vdots \\ h_m(\mathbf{x}_N) \end{bmatrix}.$$

Los vectores z son vectores binarios en $\mathbb{R}^m$.

Dado que la red tiene flexibilidad para encontrar $h_1, \ldots, h_m$, podemos acotar por arriba el número posible de diferentes $\mathbf{z}_1, \ldots, \mathbf{z}_N$.

Las primeras componentes de los vectores z están dadas por:

$$h_1(\mathbf{x}_1), \ldots, h_1(\mathbf{x}_N)$$

que es una dicotomía sobre $\mathbf{x}_1, \ldots, \mathbf{x}_N$ implementada por h_1 . Dado que la dimensión VC de $\mathcal{H}_1$ es d_1 , existen a lo más N^{d_1} dicotomías. Es decir, existen a lo más N^{d_1} formas de elegir las primeras componentes de z.

→ Recordar que:

$$m_{\mathcal{H}}(N) \leq N^{d_{\text{vc}}} + 1.$$

Capacidad de una red feed-forward

Luego, el número total de asignaciones para $\mathbf{z}_1, \ldots, \mathbf{z}_N$ es:

$$\prod_{i=1}^{m} N^{d_i} = N^{\sum_{i=1}^{m} d_i}$$

Capacidad de una red feed-forward

Luego, el número total de asignaciones para $\mathbf{z}_1, \ldots, \mathbf{z}_N$ es:

$$\prod_{i=1}^{m} N^{d_i} = N^{\sum_{i=1}^{m} d_i}$$

Cada una de estas asignaciones puede ser dicotomizada en a lo más N^{d_c} formas. Cada una de estas dicotomías para una asignación particular entrega una dicotomía para los datos $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Luego, el número máximo de dicotomías implementables sobre los datos es acotada superiormente por el producto:

$$m(N) \leq N^{d_c} \cdot N^{\sum_{i=1}^{m} d_i} = N^{d_c + \sum_{i=1}^{m} d_i}.$$

Capacidad de una red feed-forward

Luego, el número total de asignaciones para $\mathbf{z}_1, \ldots, \mathbf{z}_N$ es:

$$\prod_{i=1}^{m} N^{d_i} = N^{\sum_{i=1}^{m} d_i}$$

Cada una de estas asignaciones puede ser dicotomizada en a lo más N^{d_c} formas. Cada una de estas dicotomías para una asignación particular entrega una dicotomía para los datos $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Luego, el número máximo de dicotomías implementables sobre los datos es acotada superiormente por el producto:

$$m(N) \leq N^{d_c} \cdot N^{\sum_{i=1}^{m} d_i} = N^{d_c + \sum_{i=1}^{m} d_i}.$$

Para un MLP de dos capas:

$$d_i = d + 1$$

$$d_c = m + 1$$

$$\rightarrow D = d_c + \sum_{i=1}^{m} d_i = m(d+2) + 1 = O(md)$$

Capacidad de una red feed-forward

Luego, el número total de asignaciones para $\mathbf{z}_1, \ldots, \mathbf{z}_N$ es:

$$\prod_{i=1}^{m} N^{d_i} = N^{\sum_{i=1}^{m} d_i}$$

Cada una de estas asignaciones puede ser dicotomizada en a lo más N^{d_c} formas. Cada una de estas dicotomías para una asignación particular entrega una dicotomía para los datos $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Luego, el número máximo de dicotomías implementables sobre los datos es acotada superiormente por el producto:

$$m(N) \leq N^{d_c} \cdot N^{\sum_{i=1}^{m} d_i} = N^{d_c + \sum_{i=1}^{m} d_i}.$$

Para un MLP de dos capas:

$$d_i = d + 1$$

$$d_c = m + 1$$

$$\rightarrow D = d_c + \sum_{i=1}^{m} d_i = m(d+2) + 1 = O(md)$$

Recordar que:

$$m_{\mathcal{H}}(N) \leq N^{d_{\text{vc}}} + 1.$$

→ Al menos separa $\Omega(md)$ puntos.

Capacidad de una red feed-forward

El análisis anterior puede extenderse para encontrar una cota superior ajustada:

$$d_{\mathrm{VC}} \leq 2D \log_2 D \qquad \text{(propuesto)}$$

Para la MLP de dos capas, entonces:

$$d_{\mathrm{VC}} = O(md \log(md))$$

Capacidad de una red feed-forward

El análisis anterior puede extenderse para encontrar una cota superior ajustada:

$$d_{\text{VC}} \leq 2D \log_2 D \qquad \text{(propuesto)}$$

Para la MLP de dos capas, entonces:

$$d_{\text{VC}} = O(md \log(md))$$

Las neuronas de la capa oculta no deben ser demasiadas pero deben ser suficientes para ajustar el modelo a los datos.

Capacidad de una red feed-forward

El análisis anterior puede extenderse para encontrar una cota superior ajustada:

$$d_{\mathrm{VC}} \leq 2D \log_2 D \qquad \text{(propuesto)}$$

Para la MLP de dos capas, entonces:

$$d_{\mathrm{VC}} = O(md \log(md))$$

Las neuronas de la capa oculta no deben ser demasiadas pero deben ser suficientes para ajustar el modelo a los datos.

En general, es una buena decisión que el número de neuronas crezca sublinealmente con los datos. Por ejemplo:

$$m \approx \frac{1}{d}\sqrt{N}. \longrightarrow d_{\mathrm{VC}} = O(\sqrt{N} \log \sqrt{N})$$

Capacidad de una red feed-forward

Recordar:

$$m_{\mathcal{H}}(N) \leq N^{d_{\text{vc}}} + 1 \sim N^{d_{\text{vc}}}$$

$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + O\left(\sqrt{\frac{d_{\text{vc}} \log N}{N}}\right)$$

Si $m \approx \frac{1}{d}\sqrt{N}$. $\longrightarrow$ $d_{\text{VC}} = O(\sqrt{N} \log \sqrt{N})$

De esta manera si $N \to \infty$, $E_{\text{out}} \to E_{\text{in}}$.

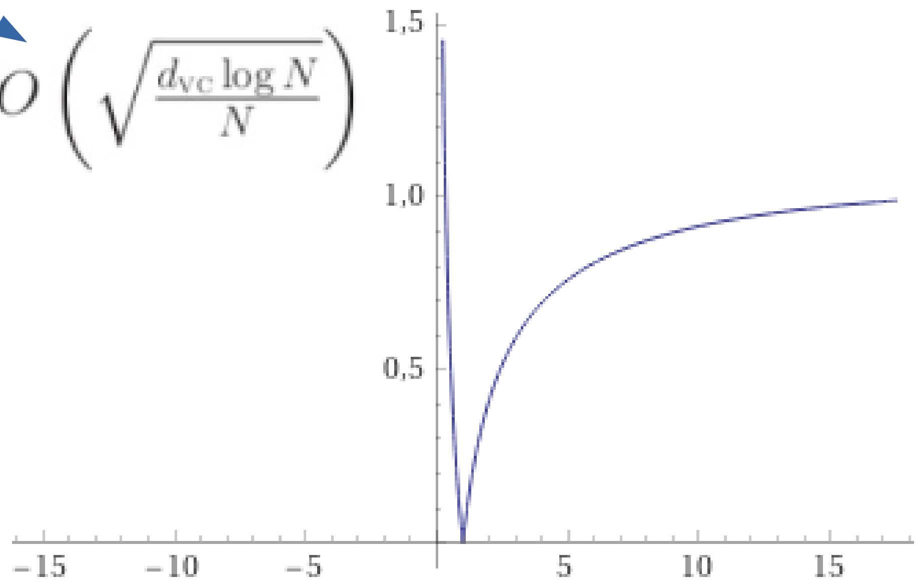
Capacidad de una red feed-forward

Si $m \approx \frac{1}{d}\sqrt{N}$. $\longrightarrow$ $d_{\mathrm{VC}} = O(\sqrt{N}\log\sqrt{N})$

De esta manera si $N \to \infty,\ E_{\text{out}} \to E_{\text{in}}$.

Complejidad paramétrica

$$O\left(\sqrt{\frac{d_{\mathrm{VC}}\log N}{N}}\right)$$



$$\sqrt{\frac{\sqrt{x}\ \log(\sqrt{x}\,)\log(x)}{x}}$$

(x de −15,5 a 17,5)

Aspectos prácticos en entrenamiento

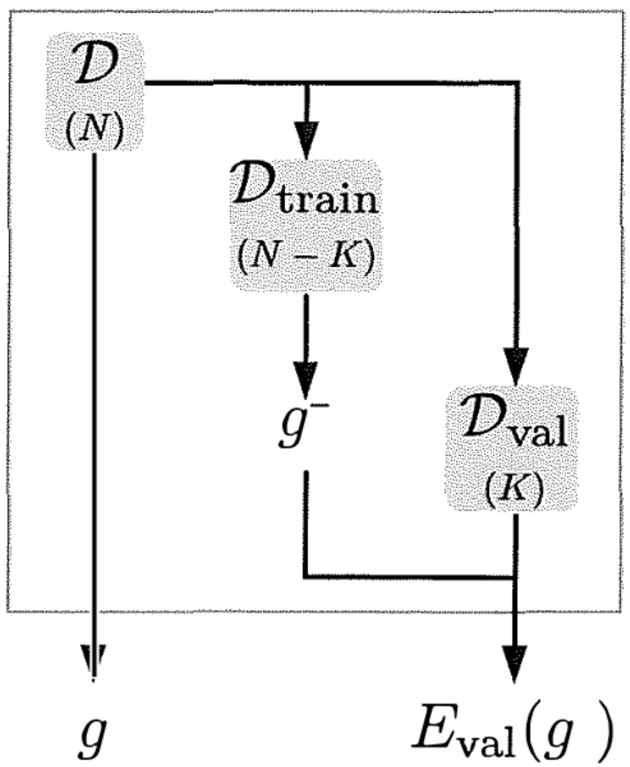
Regularización: penalizar el uso innecesario de parámetros del modelo, evitando el sobreajuste.

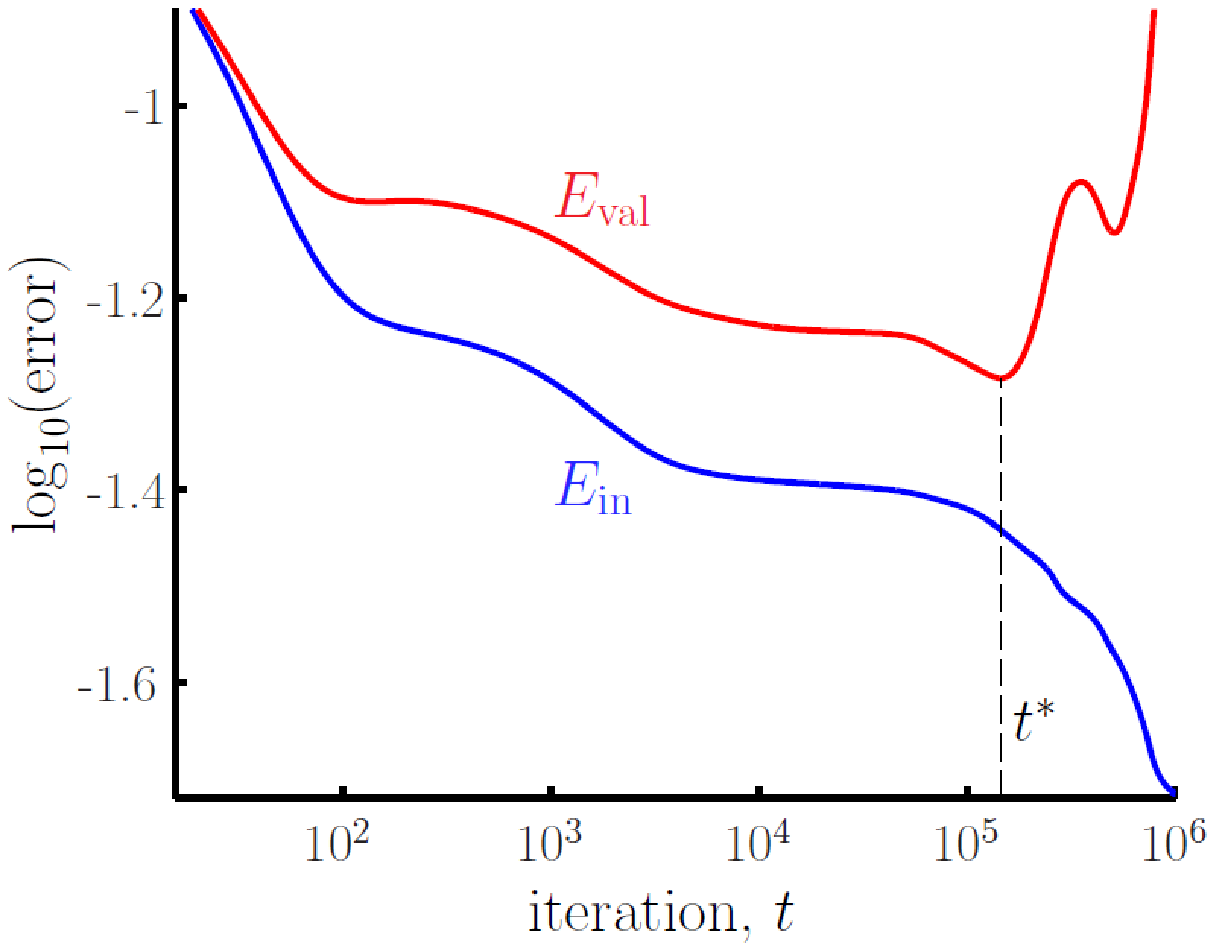
$$E_{\text{aug}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(h(\mathbf{x}_n;\mathbf{w}) - y_n)^2 + \frac{\lambda}{N}\sum_{\ell,i,j}(w_{ij}^{(\ell)})^2$$

$$\frac{\partial E_{\text{aug}}(\mathbf{w})}{\partial \mathrm{W}^{(\ell)}} = \underset{\substack{\uparrow \\ \text{backpropagation}}}{\frac{\partial E_{\text{in}}(\mathbf{w})}{\partial \mathrm{W}^{(\ell)}}} + \frac{2\lambda}{N}\mathrm{W}^{(\ell)}$$

Aspectos prácticos en entrenamiento

Early stopping: agregar un set de validación para monitorear durante el entrenamiento el error fuera de muestra.



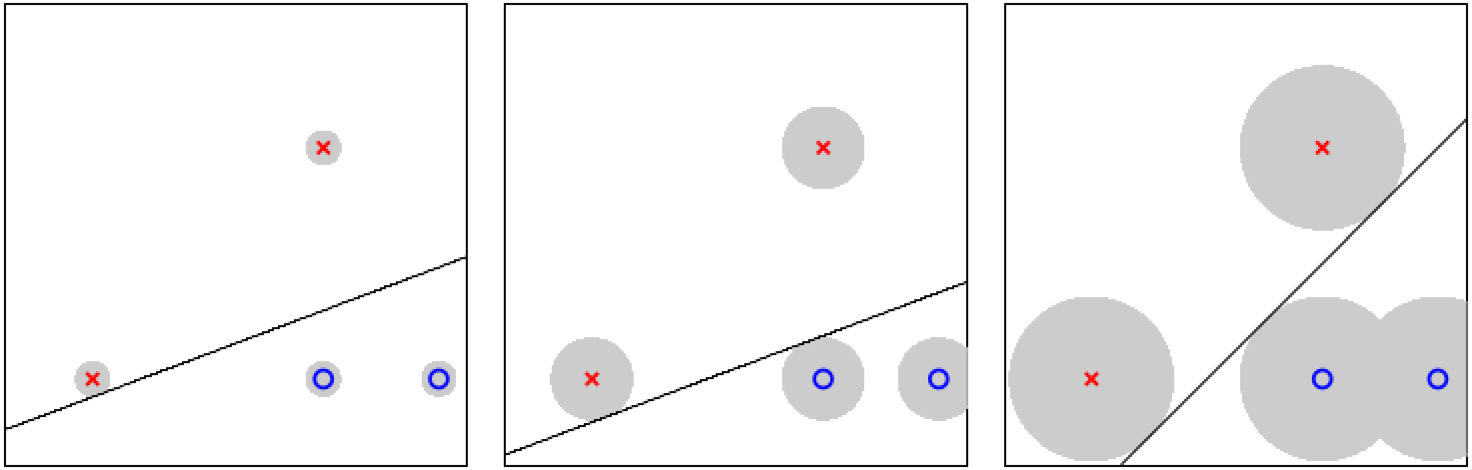


Learning with kernels

Separadores y datos ruidosos

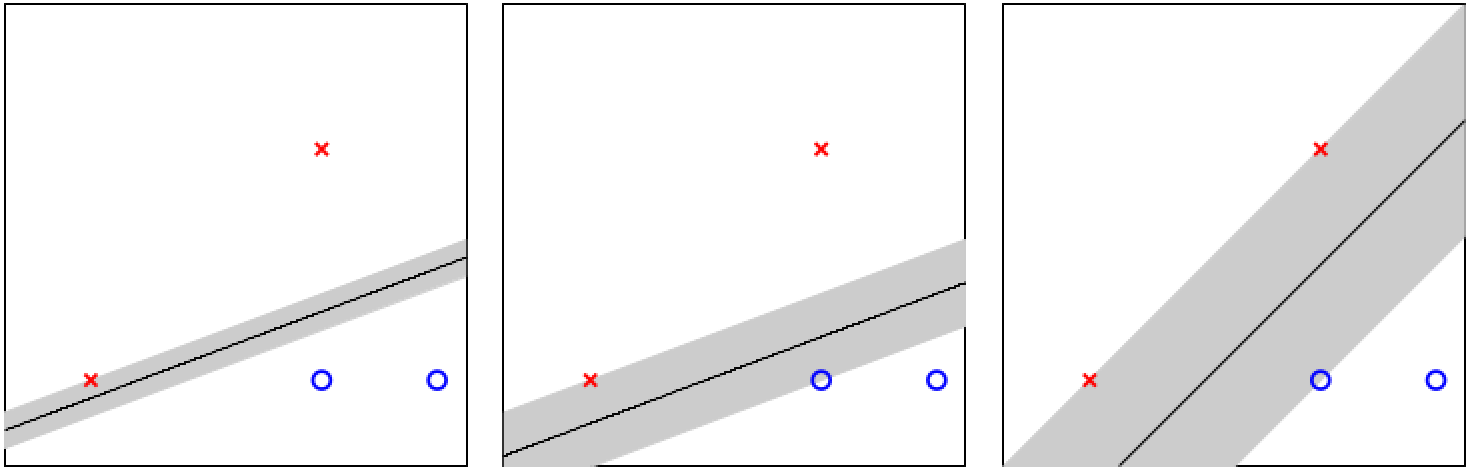
- Los datos pueden ser ruidosos (ruido de medición).
- Nuestros modelos debieran ser robustos a datos ruidosos.

Idea: La robustez al ruido tiene relación con considerar un margen de error para las mediciones.



Separadores y datos ruidosos

Una idea análoga a márgenes para datos consiste en trabajar con **hiperplanos gruesos**, agregando un margen al separador.



Separadores y datos ruidosos

Para trabajar con un hiperlano grueso, podemos usar el sesgo de una manera ingeniosa.

<u>Hiperplano estrecho</u>

$$\mathbf{x} \in \{1\} \times \mathbb{R}^d; \;\; \mathbf{w} \in \mathbb{R}^{d+1}$$

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}.$$

$$\text{signal} = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

El sesgo se codifica como una dimensión más

<u>Hiperplano grueso</u>

$$\mathbf{x} \in \mathbb{R}^d; \;\; b \in \mathbb{R}, \;\; \mathbf{w} \in \mathbb{R}^d$$

$$\text{bias } b$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}.$$

$$\text{signal} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$$

El sesgo aditivo interviene en el espacio de representación

Separadores y datos ruidosos

Hiperplano grueso

$$\mathbf{x} \in \mathbb{R}^d;\ b \in \mathbb{R},\ \mathbf{w} \in \mathbb{R}^d$$

bias b

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}.$$

$$\text{signal} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$$

sesgo