Tema 4 – Redes de Neuronas

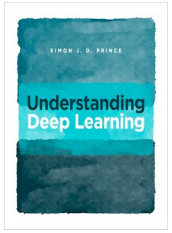
Aprendizaje Automático II - Grado en Inteligencia Artificial
Universidad Rey Juan Carlos

Iván Ramírez Díaz
ivan.ramirez@urjc.es

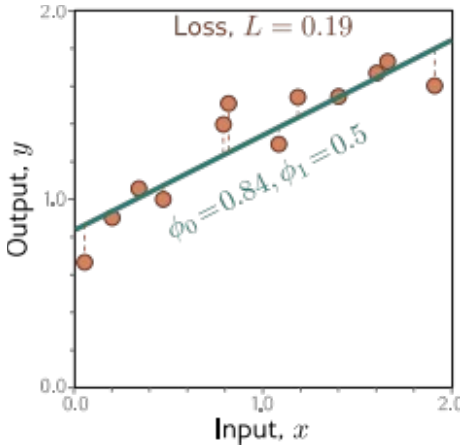
José Miguel Buenaposada Biencinto
josemiguel.buenaposada@urjc.es

Bibliografía

- Understanding Deep Learning. Capítulo 3.



Ejemplo: regresión lineal 1D

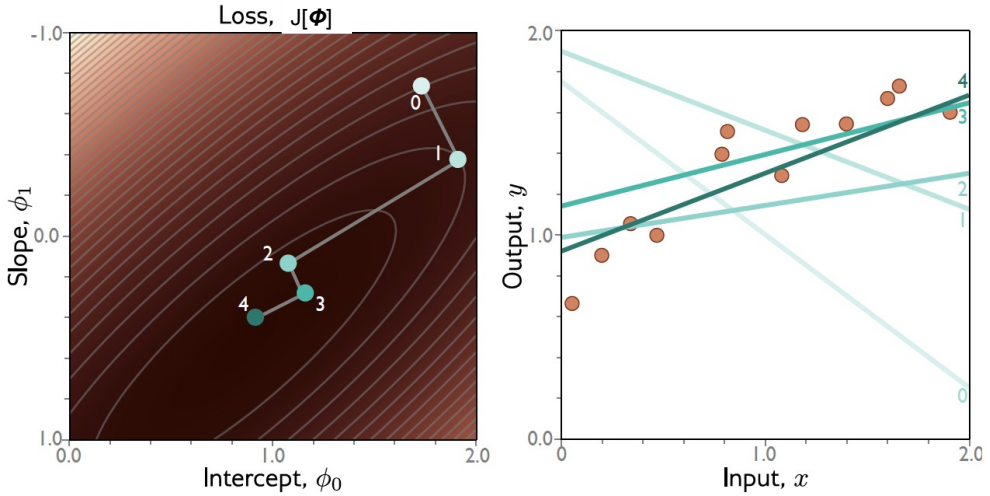


- Función de coste:

$$J(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\left(f(\boldsymbol{x_i};\boldsymbol{w}) - y_i\right)^2$$
$$= \frac{1}{N}\sum_{i=1}^{N}\left(w_1 x_i + w_0 - y_i\right)^2$$

Función de coste basada en el “error cuadrático medio” (“Mean squared error”) o problema de “mínimos cuadrados” (“least squares problem”)

Ejemplo: regresión lineal 1D



4.1 Redes de Neuronas

- Ejemplo de red, 1 entrada, 1 salida
- El teorema de aproximación universal
- Más de una salida
- Más de una entrada
- Caso general
- Número de regiones
- Terminología

4.1 Redes de Neuronas

- Ejemplo de red, 1 entrada, 1 salida
- El teorema de aproximación universal
- Más de una salida
- Más de una entrada
- Caso general
- Número de regiones
- Terminología

Redes de Neuronas (no profundas)

- Un modelo de regresión univariante es muy limitado
  - Queremos describir entradas/salidas que no sean líneas
  - Queremos múltiples entradas
  - Queremos múltiples salidas

- Redes de Neuronas (no profundas)
  - Suficientemente flexibles para describir cualquier relación funcional compleja entre la entrada y salida.
  - Cualquier número de entradas
  - Cualquier número de salidas

Regresión lineal vs Red de Neuronas

- Regresión lineal univariante (1D):

$$\begin{aligned} y &= \mathrm{f}[x, \boldsymbol{\phi}] \\ &= \phi_0 + \phi_1 x \end{aligned}$$

- Ejemplo de red de neuronas:

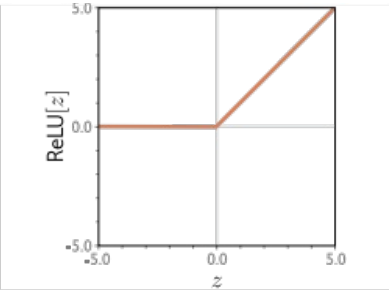
$$\begin{aligned} y &= \mathrm{f}[x, \boldsymbol{\phi}] \\ &= \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11} x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21} x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31} x] \end{aligned}$$

Ejemplo de Red de Neuronas

$$y = \mathrm{f}[x, \boldsymbol{\phi}]$$
$$= \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$

Función de activación

$$\mathrm{a}[z] = \mathrm{ReLU}[z] = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}.$$



Rectified Linear Unit

(Un tipo de función de activación)

K Fukushima. Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements. IEEE Tran. On Sys. Science and Cybernetics. 1969

Ejemplo de Red de Neuronas

$$y = \mathrm{f}[x, \boldsymbol{\phi}]$$
$$= \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$

Modelo con 10 parámetros

$$\boldsymbol{\phi} = \{\phi_0, \phi_1, \phi_2, \phi_3, \theta_{10}, \theta_{11}, \theta_{20}, \theta_{21}, \theta_{30}, \theta_{31}\}$$

- Representa una familia de funciones
- Los parámetros determinan la función particular
- Conocidos los parámetros se puede realizar la inferencia (ejecutar la función para una entrada x).

Ejemplo de Red de Neuronas

$$y = \mathrm{f}[x, \boldsymbol{\phi}]$$
$$= \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$

Modelo con 10 parámetros

$$\boldsymbol{\phi} = \{\phi_0, \phi_1, \phi_2, \phi_3, \theta_{10}, \theta_{11}, \theta_{20}, \theta_{21}, \theta_{30}, \theta_{31}\}$$

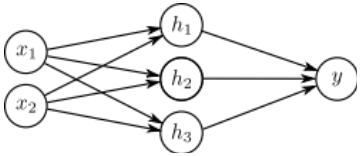
- Dado un conjunto de entrenamiento $D = \{x_i, y_i\}$
  - Podemos definir una función de pérdida $L[\boldsymbol{\Phi}]$
  - Cambiar los parámetros para que minimice la pérdida esperada.

Dos entradas

- 2 entradas, 3 unidades ocultas, 1 salida

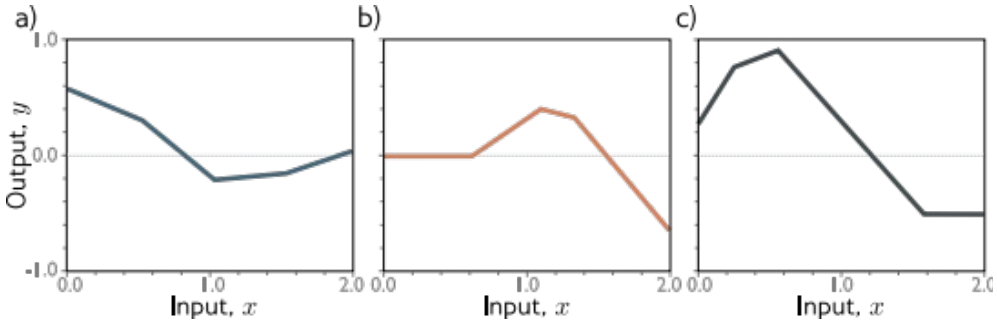
$$h_1 = \mathrm{a}[\theta_{10} + \theta_{11}x_1 + \theta_{12}x_2]$$
$$h_2 = \mathrm{a}[\theta_{20} + \theta_{21}x_1 + \theta_{22}x_2]$$
$$h_3 = \mathrm{a}[\theta_{30} + \theta_{31}x_1 + \theta_{32}x_2]$$

$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$



Ejemplo de Red de Neuronas

$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x].$$



Función lineal a trozos con 3 puntos de articulación

Unidades ocultas (Hidden units)

$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x].$$

Se puede descomponer en 2 partes:

$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$

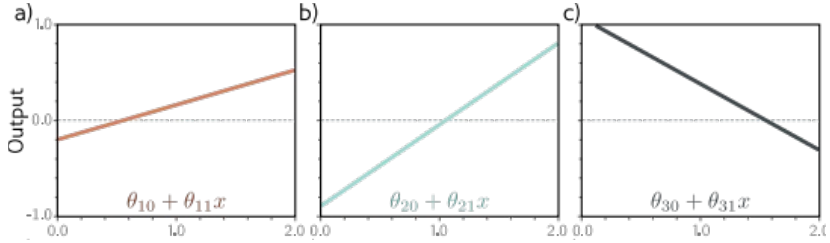
donde

Unidades ocultas

$$\begin{cases} h_1 = \mathrm{a}[\theta_{10} + \theta_{11}x] \\ h_2 = \mathrm{a}[\theta_{20} + \theta_{21}x] \\ h_3 = \mathrm{a}[\theta_{30} + \theta_{31}x] \end{cases}$$

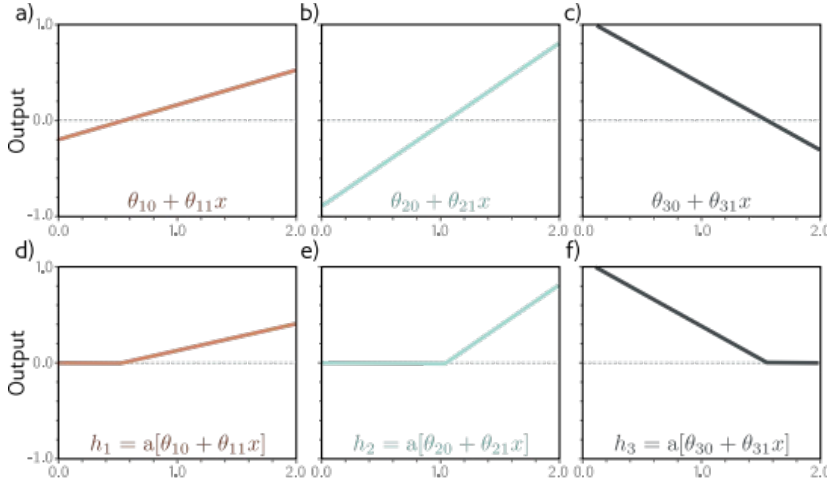
Ejemplo de Red de Neuronas: componentes

1. Calcular tres funciones lineales:



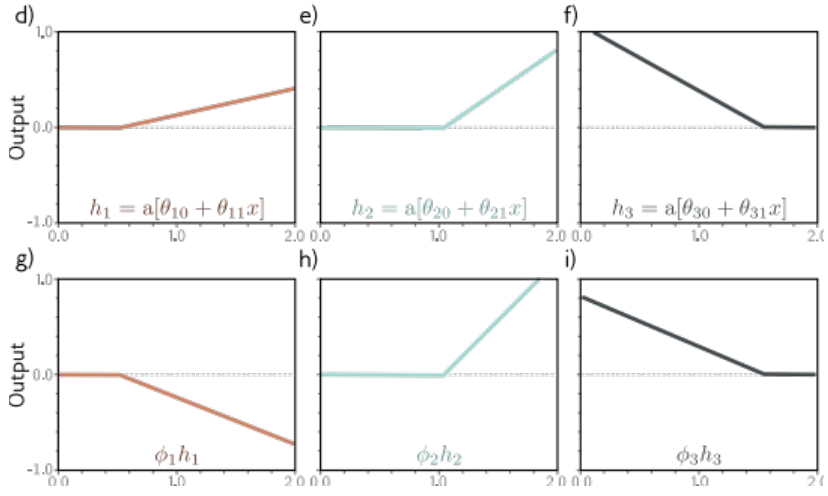
Ejemplo de Red de Neuronas: componentes

2. Pasar por la activación ReLU → valor unidades ocultas



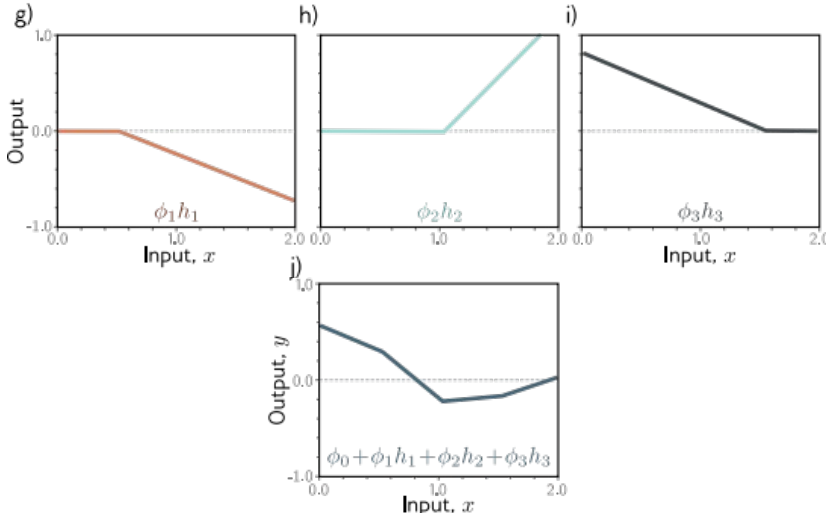
Ejemplo de Red de Neuronas: componentes

2. Pesar las unidades ocultas



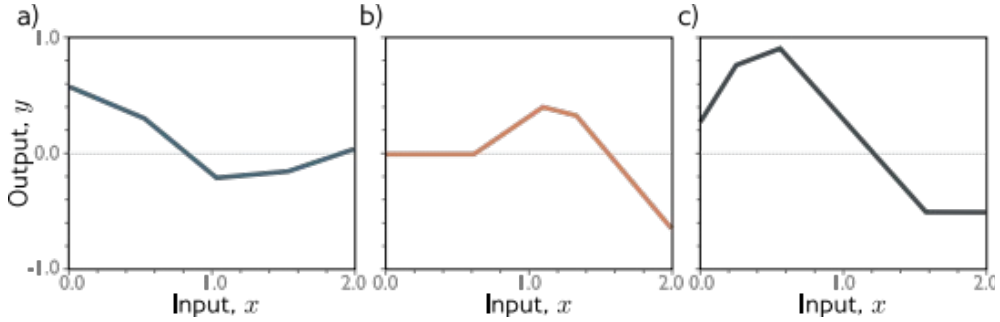
Ejemplo de Red de Neuronas: componentes

2. Pesar las unidades ocultas



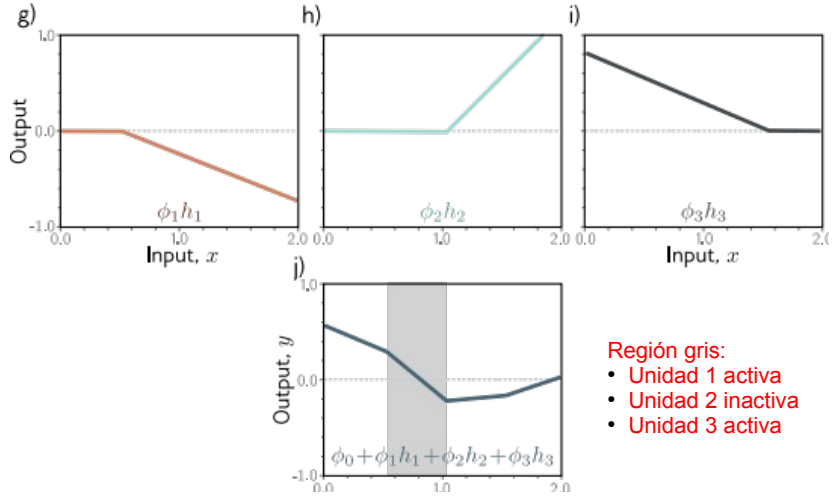
Ejemplo de Red de Neuronas

$$y = \phi_0 + \phi_1 a[\theta_{10} + \theta_{11}x] + \phi_2 a[\theta_{20} + \theta_{21}x] + \phi_3 a[\theta_{30} + \theta_{31}x].$$



Ejemplo de red de neuronas = función lineal a trozos
1 "punto de articulación" por cada función ReLU

Patrón de activación = qué unidad oculta está activada

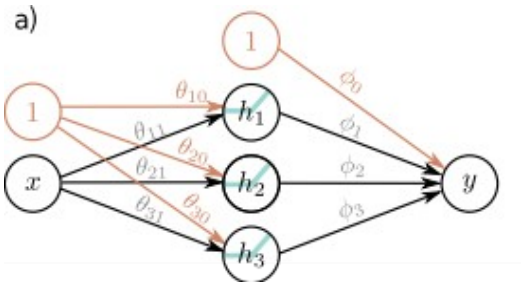


Región gris:
- Unidad 1 activa
- Unidad 2 inactiva
- Unidad 3 activa

Representación de Redes de Neuronas

$$h_1 = \mathrm{a}[\theta_{10} + \theta_{11}x]$$
$$h_2 = \mathrm{a}[\theta_{20} + \theta_{21}x]$$
$$h_3 = \mathrm{a}[\theta_{30} + \theta_{31}x]$$

$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$

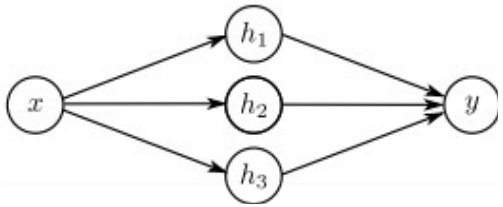


Cada parámetro (sobre el arco) multiplica a su origen y se suma a su destino.

Representación de Redes de Neuronas

$$h_1 = \mathrm{a}[\theta_{10} + \theta_{11}x]$$
$$h_2 = \mathrm{a}[\theta_{20} + \theta_{21}x]$$
$$h_3 = \mathrm{a}[\theta_{30} + \theta_{31}x]$$

$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$



¡Es un grafo de cómputo como los que vimos en el Tema 2!

4.1 Redes de Neuronas

- Ejemplo de red, 1 entrada, 1 salida
- El teorema de aproximación universal
- Más de una salida
- Más de una entrada
- Caso general
- Número de regiones
- Terminología

Número de unidades en ocultas

- Con 3 unidades ocultas:

$$h_1 = \mathrm{a}[\theta_{10} + \theta_{11}x]$$
$$h_2 = \mathrm{a}[\theta_{20} + \theta_{21}x]$$
$$h_3 = \mathrm{a}[\theta_{30} + \theta_{31}x]$$

$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$

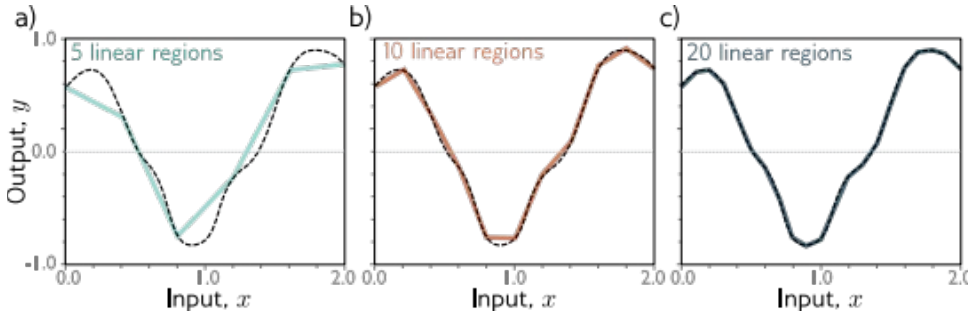
- Con D unidades ocultas:

$$h_d = \mathrm{a}[\theta_{d0} + \theta_{d1}x]$$

$$y = \phi_0 + \sum_{d=1}^{D} \phi_d h_d$$

Con suficientes unidades ocultas ...

… podemos aproximar cualquier función 1D con precisión arbitraria.



Teorema de aproximación universal

"Prueba formal de que, con un número suficiente de unidades ocultas, una red de neuronas puede describir cualquier función continua en un subconjunto compacto de $\mathbb{R}^D$ con precisión arbitraria"

- Prueban la existencia una red que puede representar cualquier función. A veces se necesita un número exponencialmente grande de unidades ocultas.
- No prueban si esa red se puede encontrar mediante un algoritmo de aprendizaje.

4.1 Redes de Neuronas

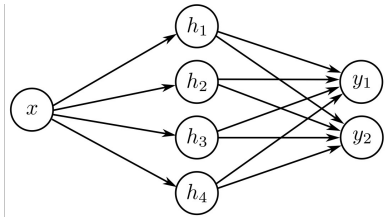
- Ejemplo de red, 1 entrada, 1 salida
- El teorema de aproximación universal
- Más de una salida
- Más de una entrada
- Caso general
- Número de regiones
- Terminología

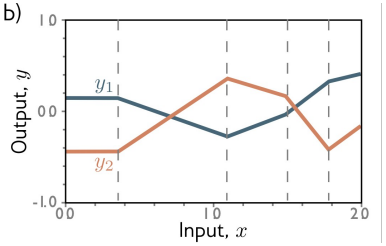
Dos salidas

- 1 entrada, 4 unidades ocultas, 2 salidas

$$
\begin{aligned}
h_1 &= \mathrm{a}[\theta_{10} + \theta_{11}x] \\
h_2 &= \mathrm{a}[\theta_{20} + \theta_{21}x] \\
h_3 &= \mathrm{a}[\theta_{30} + \theta_{31}x] \\
h_4 &= \mathrm{a}[\theta_{40} + \theta_{41}x]
\end{aligned}
$$

$$
\begin{aligned}
y_1 &= \phi_{10} + \phi_{11}h_1 + \phi_{12}h_2 + \phi_{13}h_3 + \phi_{14}h_4 \\
y_2 &= \phi_{20} + \phi_{21}h_1 + \phi_{22}h_2 + \phi_{23}h_3 + \phi_{24}h_4
\end{aligned}
$$

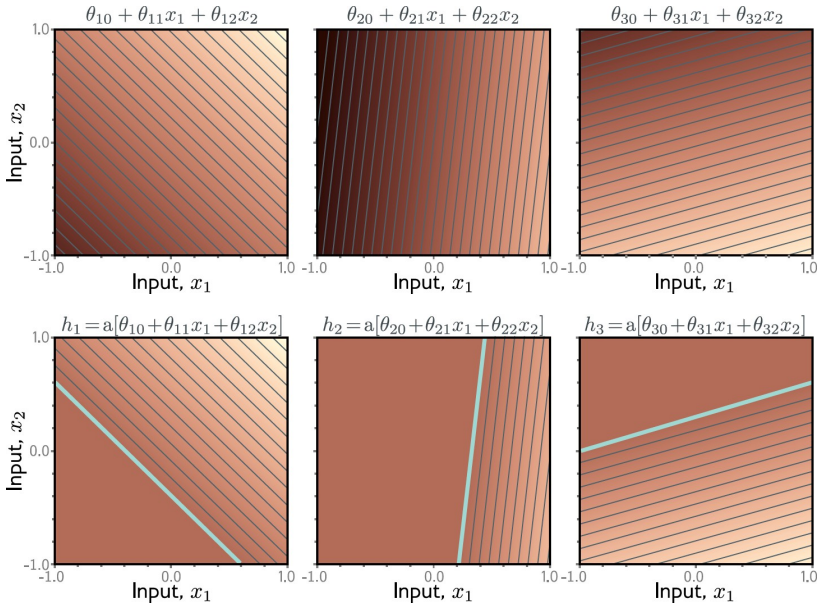




4.1 Redes de Neuronas

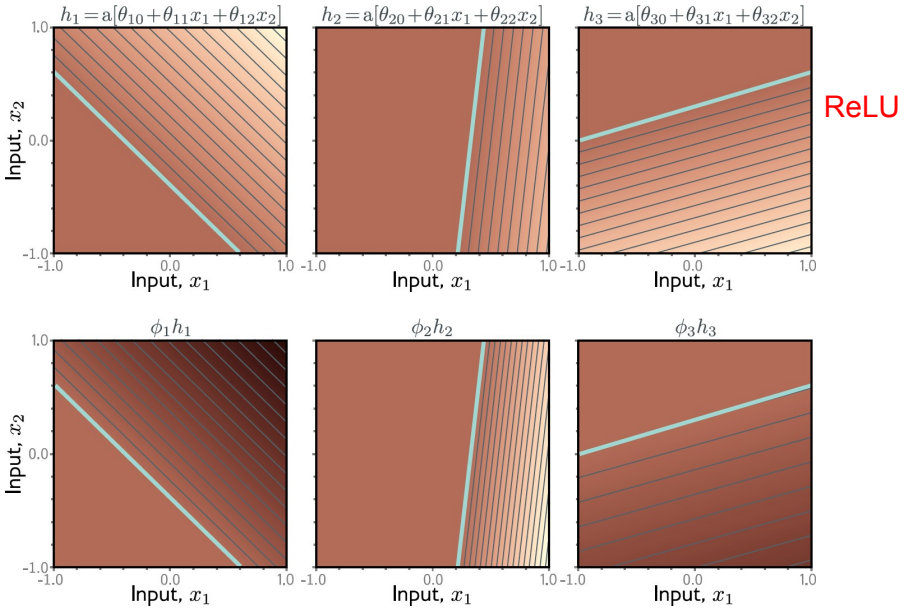
- Ejemplo de red, 1 entrada, 1 salida
- El teorema de aproximación universal
- Más de una salida
- Más de una entrada
- Caso general
- Número de regiones
- Terminología

Dos entradas



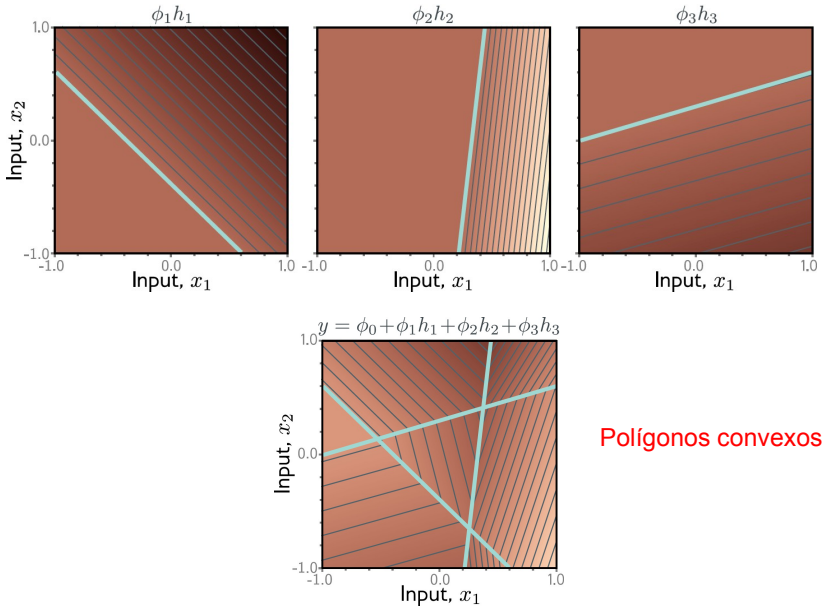
ReLU

Dos entradas



ReLU

Dos entradas

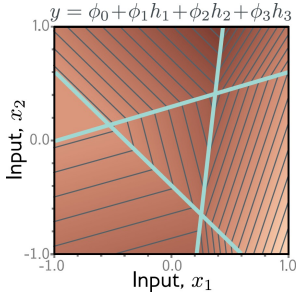


ReLU

Polígonos convexos

Pregunta 1

- Para el caso 2D, ¿qué ocurriría si tenemos 2 salidas?
- Si esta es una de las dos salidas,



- ¿Qué aspecto tendría la otra?

4.1 Redes de Neuronas

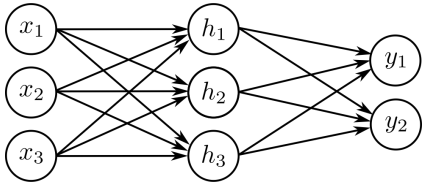
- Ejemplo de red, 1 entrada, 1 salida
- El teorema de aproximación universal
- Más de una salida
- Más de una entrada
- Caso general
- Número de regiones
- Terminología

D_i entradas, D unidades ocultas y D_o salidas

- D_i, Entradas, D unidades ocultas, y D_o salidas

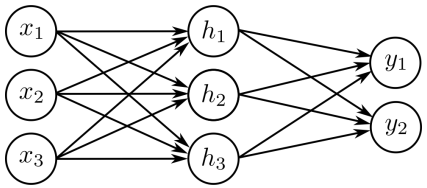
$$h_d = \mathrm{a}\left[\theta_{d0} + \sum_{i=1}^{D_i} \theta_{di} x_i\right] \qquad y_j = \phi_{j0} + \sum_{d=1}^{D} \phi_{jd} h_d$$

- Ejemplo: 3 entradas, 3 unidades ocultas, 2 salidas



Pregunta 2

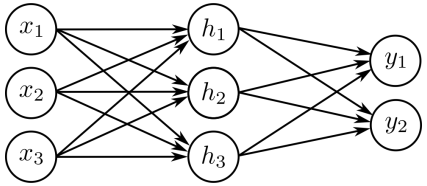
- ¿Cuántos parámetros tiene este modelo?



$$\boldsymbol{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \theta_{10} \\ \theta_{20} \\ \theta_{30} \end{bmatrix} \qquad \boldsymbol{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} a(z_1) \\ a(z_2) \\ a(z_3) \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} + \begin{bmatrix} \phi_{10} \\ \phi_{20} \end{bmatrix}$$

Pregunta 2

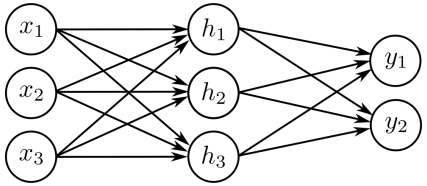
- ¿Cuántos parámetros tiene este modelo?



$$\boldsymbol{z}=\boldsymbol{\Omega}_0\boldsymbol{x}+\boldsymbol{\beta}_0 \qquad \boldsymbol{h}=\begin{bmatrix}h_1\\h_2\\h_3\end{bmatrix}=\begin{bmatrix}a(z_1)\\a(z_2)\\a(z_3)\end{bmatrix} \qquad \boldsymbol{y}=\begin{bmatrix}y_1\\y_2\end{bmatrix}=\begin{bmatrix}\phi_{11} & \phi_{12} & \phi_{13}\\ \phi_{21} & \phi_{22} & \phi_{23}\end{bmatrix}\begin{bmatrix}h_1\\h_2\\h_3\end{bmatrix}+\begin{bmatrix}\phi_{10}\\\phi_{20}\end{bmatrix}$$

Pregunta 2

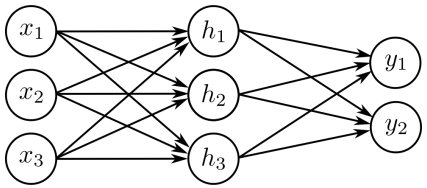
- ¿Cuántos parámetros tiene este modelo?



$$\boldsymbol{z}=\boldsymbol{\Omega}_0\boldsymbol{x}+\boldsymbol{\beta}_0 \qquad \boldsymbol{h}=a(\boldsymbol{z}) \qquad \boldsymbol{y}=\begin{bmatrix}y_1\\y_2\end{bmatrix}=\begin{bmatrix}\phi_{11} & \phi_{12} & \phi_{13}\\ \phi_{21} & \phi_{22} & \phi_{23}\end{bmatrix}\begin{bmatrix}h_1\\h_2\\h_3\end{bmatrix}+\begin{bmatrix}\phi_{10}\\\phi_{20}\end{bmatrix}$$

Pregunta 2

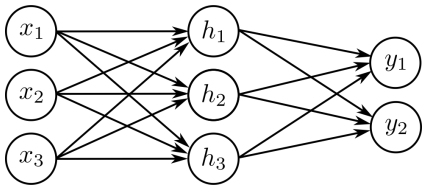
- ¿Cuántos parámetros tiene este modelo?



$$\boldsymbol{z}=\boldsymbol{\Omega}_0\boldsymbol{x}+\boldsymbol{\beta}_0 \qquad \boldsymbol{h}=a(\boldsymbol{z}) \qquad \boldsymbol{y}=\boldsymbol{\Omega}_1\boldsymbol{h}+\boldsymbol{\beta}_1$$

Pregunta 2

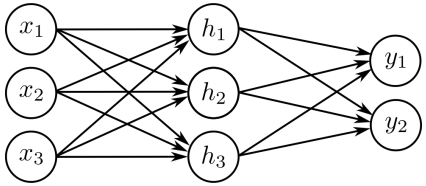
- ¿Cuántos parámetros tiene este modelo?



$$\boldsymbol{y}=\boldsymbol{\Omega}_1 a\left(\boldsymbol{\Omega}_0\boldsymbol{x}+\boldsymbol{\beta}_0\right)+\boldsymbol{\beta}_1$$

Pregunta 3

- ¿Por qué a() tiene que ser una función no lineal?

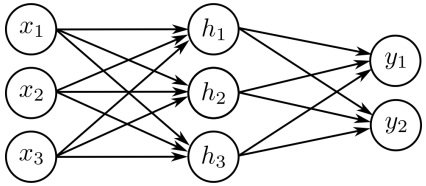


- Supongamos que a(**z**) = $\boldsymbol{\Omega_a}$ **z**

$$y = \boldsymbol{\Omega}_1 a\left(\boldsymbol{\Omega}_0 x + \boldsymbol{\beta}_0\right) + \boldsymbol{\beta}_1$$

Pregunta 3

- ¿Por qué a() tiene que ser una función no lineal?

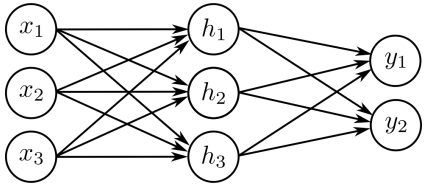


- Supongamos que a(**z**) = $\boldsymbol{\Omega_a}$ **z**

$$\begin{aligned} y &= \boldsymbol{\Omega}_1 \boldsymbol{\Omega}_a\left(\boldsymbol{\Omega}_0 x + \boldsymbol{\beta}_0\right) + \boldsymbol{\beta}_1 \\ &= \boldsymbol{\Omega}_1 \boldsymbol{\Omega}_a \boldsymbol{\Omega}_0 x + \boldsymbol{\Omega}_1 \boldsymbol{\Omega}_a \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 \end{aligned}$$

Pregunta 3

- ¿Por qué a() tiene que ser una función no lineal?



- Supongamos que a(**z**) = $\boldsymbol{\Omega_a}$ **z**

$$\begin{aligned} y &= \boldsymbol{\Omega}_1 \boldsymbol{\Omega}_a\left(\boldsymbol{\Omega}_0 x + \boldsymbol{\beta}_0\right) + \boldsymbol{\beta}_1 \\ &= \underbrace{\boldsymbol{\Omega}_1 \boldsymbol{\Omega}_a \boldsymbol{\Omega}_0}_{\boldsymbol{\Omega}_{0a1}} x + \underbrace{\boldsymbol{\Omega}_1 \boldsymbol{\Omega}_a \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1}_{\boldsymbol{\beta}_{0a1}} \end{aligned}$$

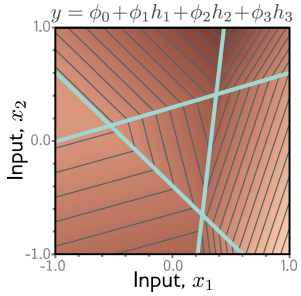
¡Si a(z) es lineal el modelo resultante **no sería mejor que un modelo lineal**!

4.1 Redes de Neuronas

- Ejemplo de red, 1 entrada, 1 salida
- El teorema de aproximación universal
- Más de una salida
- Más de una entrada
- Caso general
- Número de regiones
- Terminología

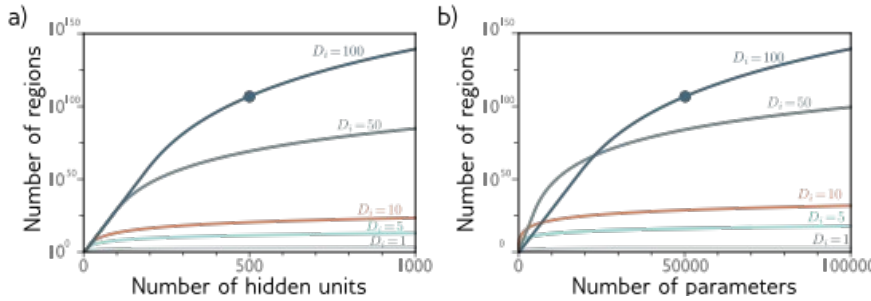
Número de regiones en la salida

- En general, cada salida consiste en politopos convexos con D dimensiones
- Con 2 entradas y 3 salidas, obtenemos 7 polígonos:



Número de regiones en la salida

- En general, cada salida consiste en politopos convexos con D dimensiones
- ¿Cuántas regiones?



Punto en azul = 500 unidades ocultas, 10^{107} regiones, 51001 parámetros

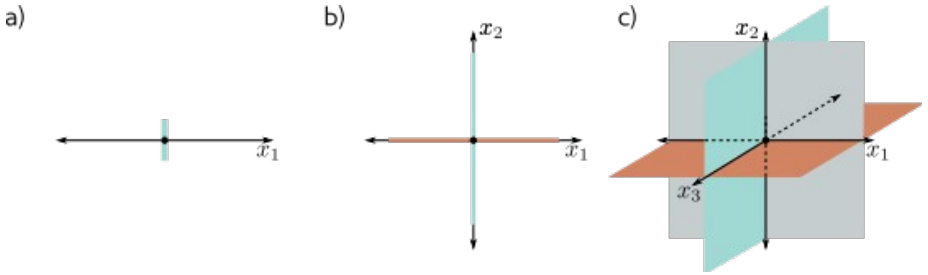
Número de regiones

- D_i número de entradas, D número de unidades ocultas
- Número de regiones creada por $D_i \leq D$ (Zavlasky,1975) como mucho es:

$$\sum_{j=0}^{D_i} \binom{D}{j}$$

- ¿Cómo de grande es eso? En redes de neuronas casi siembre $D_i < D$ y crean entre 2^{D_i} y 2^D regiones.

Si n.º entradas igual al n.º de unidades ocultas



Entrada 1D con 1 unidad oculta crea 2 regiones (un punto)

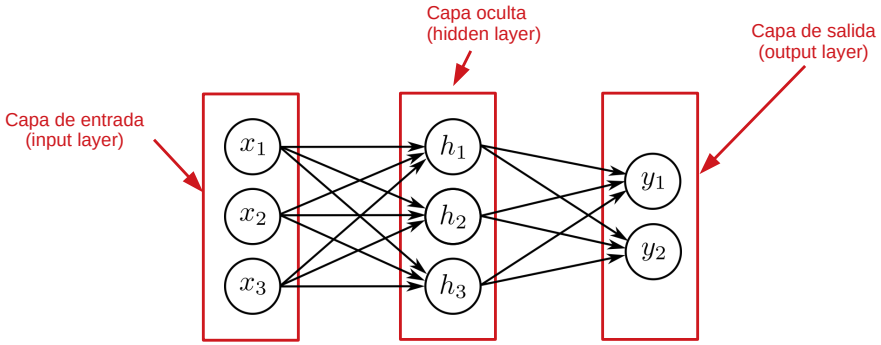
Entrada 2D con 2 unidades ocultas crea 4 regiones (dos líneas)

Entrada 3D con 3 unidades ocultas crea 8 regiones (3 planos)

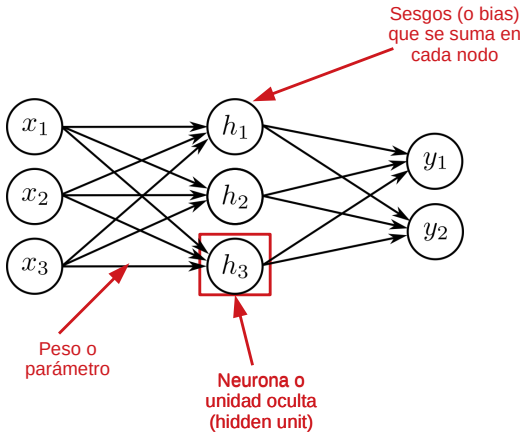
4.1 Redes de Neuronas

- Ejemplo de red, 1 entrada, 1 salida
- El teorema de aproximación universal
- Más de una salida
- Más de una entrada
- Caso general
- Número de regiones
- Terminología

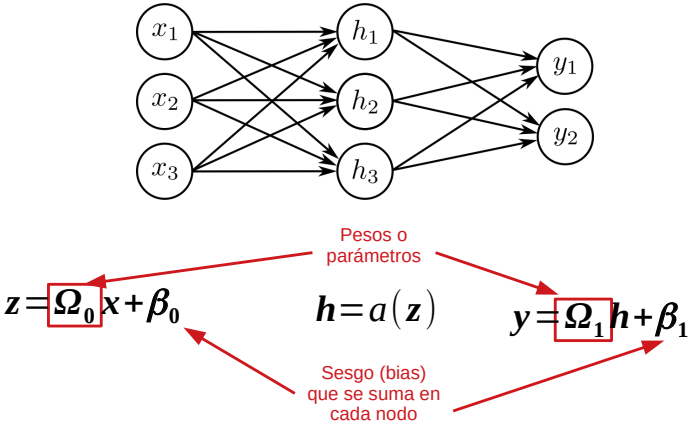
Terminología



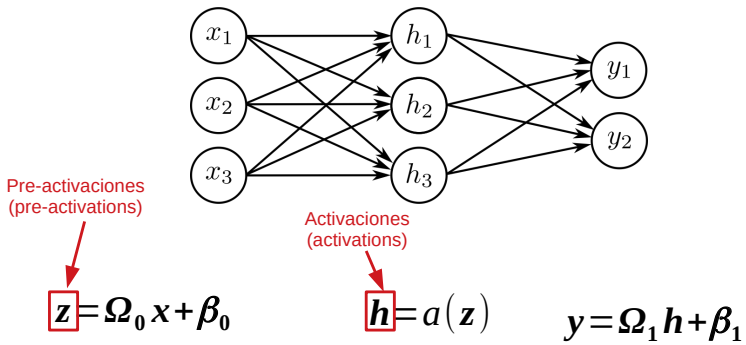
Terminología



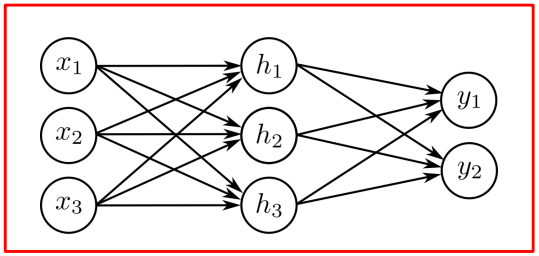
Terminología



Terminología



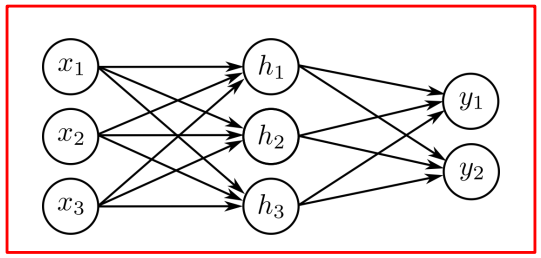
Terminología



Feedforward network (no hay ciclos en el grafo)

Fully Connected Network (todo lo de una capa esta conectado a todo en la siguiente)

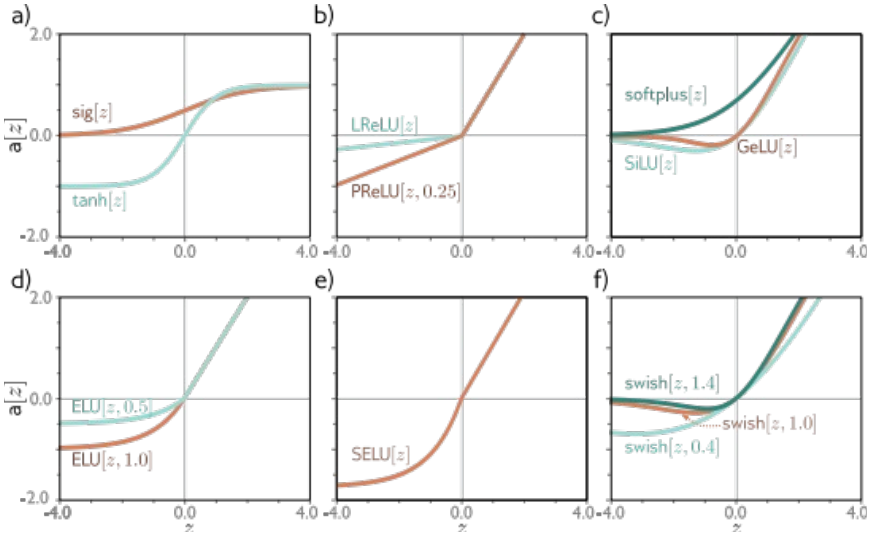
Terminología



Una capa oculta → Shallow Neural Nework

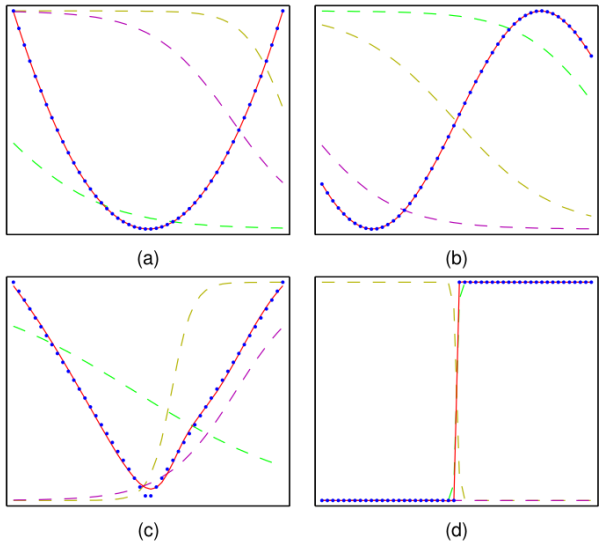
Más de una capa oculta → Deep Neural Network

Otras funciones de activación

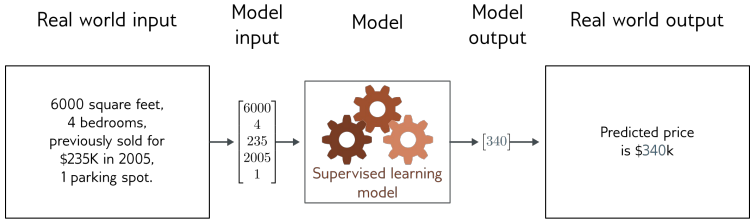


Otras funciones de activación

Figure 6.10 Illustration of the capability of a two-layer neural network to approximate four different functions: (a) $f(x) = x^2$, (b) $f(x) = \sin(x)$, (c), $f(x) = |x|$, and (d) $f(x) = H(x)$ where $H(x)$ is the Heaviside step function. In each case, $N = 50$ data points, shown as blue dots, have been sampled uniformly in x over the interval $(-1, 1)$ and the corresponding values of $f(x)$ evaluated. These data points are then used to train a two-layer network having three hidden units with tanh activation functions and linear output units. The resulting network functions are shown by the red curves, and the outputs of the three hidden units are shown by the three dashed curves.



Regresión



- Tenemos un modelo que puede:
  - Tomar un n.º arbitrario de entradas
  - Estimar un n.º arbitrario de salidas
  - Modelar una función de complejidad arbitraria de la entrada

$$h_d = \mathrm{a}\left[\theta_{d0} + \sum_{i=1}^{D_i} \theta_{di} x_i\right] \qquad y_j = \phi_{j0} + \sum_{d=1}^{D} \phi_{jd} h_d$$