Redes neuronales para matemáticos

EPM

Georgy Nuzhdin

Contenidos

¿Qué son las redes?

Una **red neuronal** es un conjunto de neuronas que transmiten una señal de entrada (una imagen, un número, un sonido...) hacia una señal de salida.
Las neuronas se organizan en capas; cada neurona de una capa está conectada a todas las de la capa siguiente. Cada conexión tiene un peso. Matemáticamente es un grafo multipartito ponderado, y biológicamente está inspirado en las conexiones neuronales del cerebro.

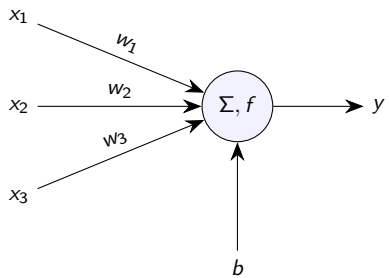
La neurona artificial

Una *neurona artificial* transforma un vector de entradas en una salida en tres pasos:

1. Cada entrada x_i se multiplica por un *peso* w_i (positivo, negativo o nulo).
2. Se suman los productos y se añade un *sesgo* b:

$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b.$$

3. Se aplica una *función de activación* f: $y = f(z)$.



La función de activación más sencilla: el escalón

La función escalón reproduce una decisión binaria:

$$H(z) = \begin{cases} 1 & \text{si } z > 0, \\ 0 & \text{si } z \leq 0. \end{cases}$$

Con ella, la neurona *dispara* (sale 1) cuando la suma ponderada es positiva, y *permanece muda* (sale 0) en caso contrario.

La idea fundamental: la neurona traza una recta (o un hiperplano) en el espacio de entradas y separa los puntos según en qué lado caen.

Problema 1.1

Enunciado

Considera una neurona con tres entradas, pesos $w_1 = 0{,}5$, $w_2 = -1$, $w_3 = 0{,}8$ y sesgo $b = -0{,}2$, con función escalón.

1. ¿Cuál es la salida con entradas $(x_1, x_2, x_3) = (1, 1, 0)$?
2. ¿Y con entradas $(x_1, x_2, x_3) = (0, 0, 1)$?

Problema 1.1 – solución

Solución

(a) Entrada $(1, 1, 0)$:

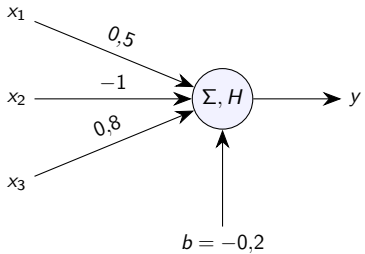
$$z = 0{,}5 - 1 + 0 - 0{,}2 = -0{,}7.$$

$H(-0{,}7) = 0$: *no dispara*.

(b) Entrada $(0, 0, 1)$:

$$z = 0 + 0 + 0{,}8 - 0{,}2 = 0{,}6.$$

$H(0{,}6) = 1$: *dispara*.



La misma neurona con dos entradas distintas: en (a) los pesos negativo y positivo se compensan y queda por debajo del umbral; en (b) la tercera entrada (peso 0,8) basta para superarlo.

Problema 1.2

Enunciado

Considera una neurona con tres entradas y parámetros $w_1 = w_2 = w_3 = 1$, $b = -1{,}5$, con función escalón. Construye la tabla con la salida y para todas las combinaciones de entradas binarias $(x_1, x_2, x_3) \in \{0, 1\}^3$ y di qué función lógica está implementando.

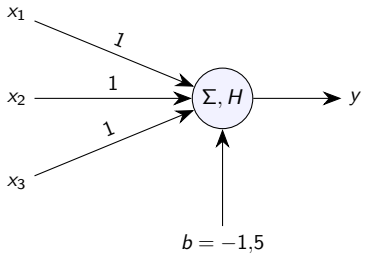
Problema 1.2 – solución

Solución

$z = x_1 + x_2 + x_3 - 1{,}5$. Dispara cuando $\sum x_i \geq 2$.

x_1	x_2	x_3	y
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	1

Función *mayoría*: sale 1 si y sólo si al menos dos entradas valen 1.



Como los tres pesos coinciden, la neurona sólo cuenta *cuántas* entradas están activas.

Diseñar los pesos

La neurona no piensa por sí sola: pensamos nosotros eligiendo los pesos. Para ver cómo, vamos a forzar a una neurona a reproducir la puerta lógica AND:

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Idea geométrica: en el plano (x_1, x_2) buscamos una recta $w_1 x_1 + w_2 x_2 + b = 0$ que deje el punto $(1, 1)$ a un lado (semiespacio positivo) y los otros tres al lado contrario.

Problema 2.1

Enunciado

Encuentra una pareja de pesos w_1, w_2 y un sesgo b tales que una neurona con función escalón implemente la puerta AND. Verifica explícitamente la salida para las cuatro combinaciones posibles. ¿Es la solución única?

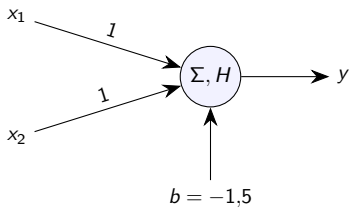
Problema 2.1 – solución

Solución

Una elección sencilla: $w_1 = w_2 = 1$, $b = -1{,}5$.

- $(0,0)$: $z = -1{,}5 \Rightarrow H = 0$.
- $(0,1), (1,0)$: $z = -0{,}5 \Rightarrow H = 0$.
- $(1,1)$: $z = 0{,}5 \Rightarrow H = 1$. ✓

No es única: cualquier $\lambda(1, 1, -1{,}5)$ con $\lambda > 0$ vale, y también cualquier recta que pase entre $(1,1)$ y los otros tres vértices.



La recta $x_1 + x_2 = 1{,}5$ separa $(1,1)$ del resto.

Problema 2.2

Enunciado

Encuentra una pareja de pesos w_1, w_2 y un sesgo b que implementen la puerta lógica OR con activación escalón:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Verifica la salida para las cuatro entradas. ¿Cómo cambia el sesgo al pasar de AND a OR?

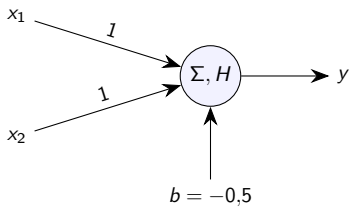
Problema 2.2 – solución

Solución

Una elección: $w_1 = w_2 = 1,\ b = -0{,}5$.

- $(0,0)$: $z = -0{,}5 \Rightarrow H = 0$.
- $(0,1), (1,0)$: $z = 0{,}5 \Rightarrow H = 1$.
- $(1,1)$: $z = 1{,}5 \Rightarrow H = 1$. ✓

Manteniendo los pesos, el sesgo **sube de** $-1{,}5$ **a** $-0{,}5$: la neurona se vuelve más *permisiva* (basta una entrada activa para disparar).

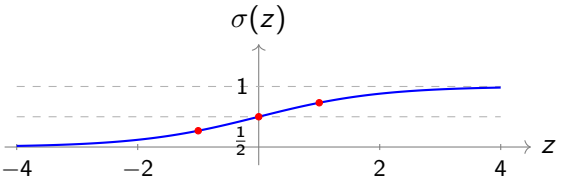


El sesgo desplaza la recta separadora: con $b = -0{,}5$, sólo $(0,0)$ queda al otro lado.

La sigmoide: una activación derivable

La función escalón funciona, pero **no es derivable**. Para que la red pueda aprender mediante derivadas necesitamos una activación suave: la sigmoide,

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$



La sigmoide aplasta cualquier número real al intervalo $(0, 1)$: una probabilidad o una çonfianza".

Propiedades clave:

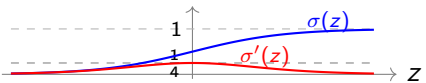
- $\sigma(0) = \frac{1}{2}$.
- Estrictamente creciente.
- $\sigma(z) \to 0$ cuando $z \to -\infty$.
- $\sigma(z) \to 1$ cuando $z \to +\infty$.
- Derivable en todo $\mathbb{R}$.

La derivada elegante de la sigmoide

La derivada de la sigmoide se expresa de manera muy elegante en términos de la propia σ:

$$\boxed{\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big)}$$

Esta fórmula es la razón por la que la sigmoide aparece en todas partes: si ya conocemos la salida $y = \sigma(z)$, calcular el gradiente es *trivial*: basta multiplicar $y(1 - y)$.



También conviene recordar la simetría

$$\sigma(-z) = 1 - \sigma(z),$$

que ahorra muchas cuentas.

Importante: $\sigma'(z)$ es máxima en $z = 0$ (valor $\frac{1}{4}$) y *cae a cero* cuando z es muy grande o muy pequeño. Esto se llama *saturación*.

Problema 3.1

Enunciado

1. Calcula $\sigma(0)$, $\sigma(1)$ y $\sigma(-1)$ con tres cifras decimales.
2. Verifica con esos valores la simetría $\sigma(-z) = 1 - \sigma(z)$.

Problema 3.1 – solución

Solución

- $\sigma(0) = 1/(1+1) = 0{,}500$.
- $\sigma(1) = 1/(1+e^{-1}) = 1/(1+0{,}3679) \approx 0{,}731$.
- $\sigma(-1) = 1/(1+e) = 1/3{,}7183 \approx 0{,}269$.

Comprobamos la simetría: $\sigma(1) + \sigma(-1) \approx 0{,}731 + 0{,}269 = 1{,}000$. ✓

Más valores útiles para tener en la cabeza: $\sigma(2) \approx 0{,}881$, $\sigma(-2) \approx 0{,}119$, $\sigma(3) \approx 0{,}953$.

Problema 3.2

Enunciado

Demuestra la fórmula

$$\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big)$$

a partir de la definición $\sigma(z) = (1 + e^{-z})^{-1}$. (Pista: aplica la regla de la cadena.)

Problema 3.2 – solución

Solución

Por la regla de la cadena, derivando $(1+e^{-z})^{-1}$,

$$\sigma'(z) = -(1+e^{-z})^{-2}\cdot(-e^{-z}) = \frac{e^{-z}}{(1+e^{-z})^2}.$$

Reescribimos:

$$\sigma'(z) = \frac{1}{1+e^{-z}}\cdot\frac{e^{-z}}{1+e^{-z}} = \sigma(z)\cdot\big(1-\sigma(z)\big),$$

donde hemos usado $\dfrac{e^{-z}}{1+e^{-z}} = 1-\dfrac{1}{1+e^{-z}} = 1-\sigma(z)$. $\square$

Ejemplo numérico: en $z=1$, $\sigma'(1) = 0{,}731\cdot 0{,}269 \approx 0{,}197$.

Medir el error

Para que la red *aprenda* hace falta cuantificar cuánto se equivoca. Si la red predice y y queríamos un objetivo $\hat{y}$, la *pérdida cuadrática* es

$$E(y) = \tfrac{1}{2}(y - \hat{y})^2.$$

- $E \geq 0$ siempre.
- $E = 0$ exactamente cuando la red acierta.
- El factor $\frac{1}{2}$ es cosmético: hace que la derivada salga limpia.

Cuando hay varios ejemplos $(x^{(k)}, \hat{y}^{(k)})$, la pérdida total es la suma:

$$E_{\text{total}} = \sum_k E_k.$$

Problema 4.1

Enunciado

Una neurona con sigmoide tiene pesos $w_1 = 1$, $w_2 = 0$ y sesgo $b = -1$. La entrada es $(x_1, x_2) = (1, 0)$ y el objetivo es $\hat{y} = 1$. Calcula la salida y y la pérdida E.

Problema 4.1 – solución

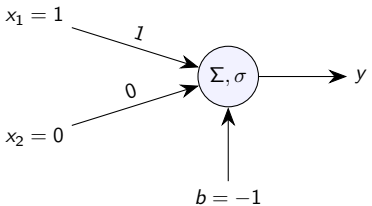
Solución

$$z = 1 \cdot 1 + 0 \cdot 0 + (-1) = 0.$$

$$y = \sigma(0) = 0{,}5.$$

$$E = \tfrac{1}{2}(0{,}5 - 1)^2 = \tfrac{1}{2} \cdot 0{,}25 = 0{,}125.$$

La neurona está justo en el centro de la sigmoide: máxima incertidumbre. Como queríamos $\hat{y} = 1$, hay que empujarla hacia arriba.



$\hat{y} = 1, \;\; y = 0{,}5, \;\; E = 0{,}125.$

Problema 4.2

Enunciado

Una neurona sigmoide tiene los pesos de la puerta AND: $w_1 = w_2 = 1$, $b = -1{,}5$. Calcula la salida y y la pérdida E para los siguientes pares entrada-objetivo:

1. Entrada $(1, 1)$, objetivo $\hat{y} = 1$.
2. Entrada $(0, 0)$, objetivo $\hat{y} = 0$.

¿Es perfecta la imitación de AND con sigmoide?

Problema 4.2 – solución

Solución

(1) $(1,1)$, $\hat{y} = 1$:

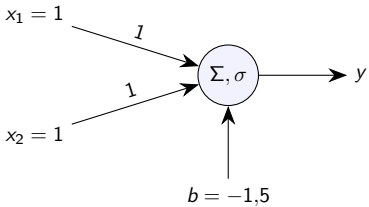
$$z = 0{,}5, \;\; y = \sigma(0{,}5) \approx 0{,}622, \;\; E \approx 0{,}071.$$

(2) $(0,0)$, $\hat{y} = 0$:

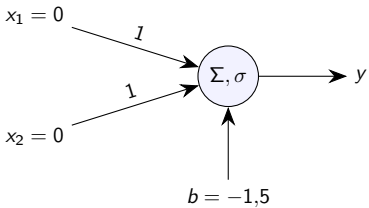
$$z = -1{,}5, \;\; y \approx 0{,}182, \;\; E \approx 0{,}017.$$

La sigmoide **no** reproduce AND exactamente: queda $E > 0$ aunque la red esté del lado correcto. Es el precio a pagar por tener una activación derivable.

Caso (1):



Caso (2):



Descenso del gradiente: bajar la colina

Imagen mental: la pérdida E es una *colina* en el espacio de los pesos. Queremos llegar al fondo.

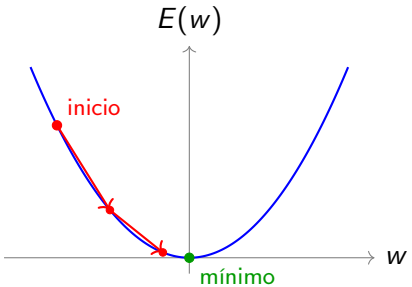
La derivada parcial $\partial E/\partial w_i$ es la *pendiente local* en la dirección de w_i. Para bajar, nos movemos al revés.

Regla de actualización:

$$w_i \longleftarrow w_i - \eta \frac{\partial E}{\partial w_i}$$

$\eta > 0$ es la *tasa de aprendizaje* (tamaño del paso).

Demasiado grande: nos pasamos. Demasiado pequeña: tardamos eternamente.

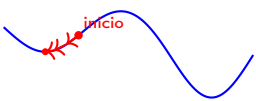


Bajamos la colina paso a paso.

La tasa de aprendizaje η: dos valles

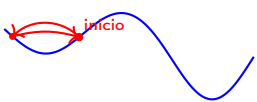
La pérdida puede tener **varios mínimos**. El tamaño del paso η decide a cuál llegamos. Consideremos una colina con dos valles (uno más profundo) y un punto de partida cerca del valle *alto*:

η pequeña



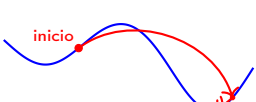
Cae al valle más cercano: queda atrapado en un *mínimo local*.

η mediana



Salta al simétrico dentro del *mismo valle*: oscila sin bajar.

η grande, luego pequeña



Salta sobre la barrera, *baja* η y desciende al valle profundo.

Lección práctica: en todo entrenamiento moderno se empieza con η relativamente grande para explorar el paisaje y se reduce gradualmente para afinar. Esta receta se llama *learning rate decay*.

Repartir el error entre los pesos

Cuando la red se equivoca, ¿**a qué peso** hay que culpar? Y, sobre todo, ¿**cuánto** debe cambiar cada uno?

Metáfora: un equipo toma una decisión por mayoría ponderada. Si la decisión es mala, el responsable es quien más *influyó*.

Para una neurona sigmoide, el reparto se hace en **tres factores**:

$$\frac{\partial E}{\partial w_i} = \underbrace{(y - \hat{y})}_{\substack{\text{error} \\ \text{global}}} \cdot \underbrace{y(1-y)}_{\substack{\text{sensibilidad} \\ \text{de } \sigma}} \cdot \underbrace{x_i}_{\substack{\text{influencia} \\ \text{local}}} .$$

- **Error global** $(y - \hat{y})$: lo mismo para todos los pesos. Cuanto más se equivoca la red, más cambian todos.
- **Sensibilidad** $y(1-y)$: cuánto se transmite el error a través de la sigmoide. Si la neurona está *saturada* ($y \approx 0$ o $y \approx 1$), apenas cambia nada.
- **Influencia local** x_i: cuánto contribuyó este peso al resultado. Si $x_i = 0$, el peso no se toca.

Repartir el error: ejemplo numérico

Cuatro neuronas idénticas (sigmoide), todas con error $(y - \hat{y}) = -0{,}5$, pero distintos z y distintas entradas:

caso	z	y	$y(1-y)$	x_1	$\partial E/\partial w_1$
A	0	0,500	0,250	1	$-0{,}500 \cdot 0{,}250 \cdot 1 = -0{,}125$
B	0	0,500	0,250	0	$-0{,}500 \cdot 0{,}250 \cdot 0 = \ \ 0{,}000$
C	3	0,953	0,045	1	$-0{,}500 \cdot 0{,}045 \cdot 1 \approx -0{,}023$
D	-3	0,047	0,045	1	$-0{,}500 \cdot 0{,}045 \cdot 1 \approx -0{,}023$

- En **A**, los tres factores tiran al máximo $\Rightarrow$ paso grande.
- En **B**, $x_1 = 0$ *anula* todo el aprendizaje en ese peso.
- En **C** y **D**, la sigmoide *saturada* ahoga el aprendizaje aunque haya error.

Esta es la lección clave: el aprendizaje se reparte por las **tres puertas** a la vez, y cualquiera de ellas puede bloquearlo.

Las fórmulas para una neurona sigmoide

Recogiendo lo anterior, los gradientes son:

$$\boxed{\frac{\partial E}{\partial w_i} = (y - \hat{y}) \cdot y(1-y) \cdot x_i} \qquad \boxed{\frac{\partial E}{\partial b} = (y - \hat{y}) \cdot y(1-y)}$$

Observa: el sesgo b se actualiza igual que un peso cuya entrada fuese siempre 1. (Ese es exactamente el papel del bias.)

Receta para un paso de aprendizaje:

1. Forward: calcular z y luego $y = \sigma(z)$.
2. Comparar con el objetivo: $(y - \hat{y})$.
3. Calcular cada gradiente con la fórmula.
4. Actualizar: $w_i \leftarrow w_i - \eta \cdot \partial E / \partial w_i$.

Problema 5.1

Enunciado

Continuando con los datos del Problema 4.1 ($w_1 = 1$, $w_2 = 0$, $b = -1$, entrada $(1, 0)$, objetivo $\hat{y} = 1$, donde $y = 0{,}5$ y $E = 0{,}125$), aplica un paso de descenso del gradiente con tasa de aprendizaje $\eta = 1$. Calcula los nuevos pesos w_1', w_2', b' y verifica que la nueva pérdida es menor.

Recordatorio de las fórmulas

$$\frac{\partial E}{\partial w_i} = (y - \hat{y}) \cdot y(1 - y) \cdot x_i, \qquad \frac{\partial E}{\partial b} = (y - \hat{y}) \cdot y(1 - y),$$

$$w_i \longleftarrow w_i - \eta \frac{\partial E}{\partial w_i}.$$

Problema 5.1 – solución

Solución

$(y - \hat{y}) = -0{,}5, \;\; y(1-y) = 0{,}25.$

$$\frac{\partial E}{\partial w_1} = (-0{,}5)(0{,}25)(1) = -0{,}125,$$

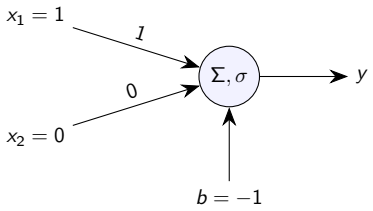
$$\frac{\partial E}{\partial w_2} = 0, \quad \frac{\partial E}{\partial b} = -0{,}125.$$

Nuevos parámetros con $\eta = 1$:

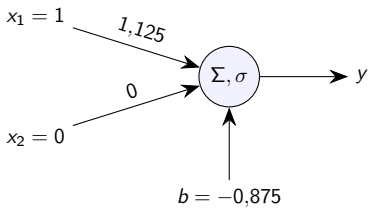
$$w_1' = 1{,}125, \;\; w_2' = 0, \;\; b' = -0{,}875.$$

Verificación: $z' = 0{,}25$, $y' \approx 0{,}5622$,
$E' \approx 0{,}0958 < 0{,}125$. ✓

Antes: $E = 0{,}125$



Después: $E \approx 0{,}096$



Problema 5.2

Enunciado

Continuamos con la neurona tras el primer paso del Problema 5.1: $w_1 = 1{,}125$, $w_2 = 0$, $b = -0{,}875$, entrada $(1, 0)$, objetivo $\hat{y} = 1$. Aplica un **segundo** paso de descenso del gradiente con $\eta = 1$. Calcula los nuevos pesos y la nueva pérdida.

Problema 5.2 – solución

Solución

$z = 0{,}25$, $y \approx 0{,}5622$, $(y-1) \approx -0{,}4378$, $y(1-y) \approx 0{,}2461$.

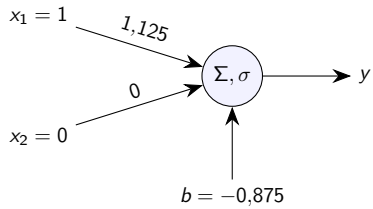
$$\frac{\partial E}{\partial w_1} \approx -0{,}1078, \quad \frac{\partial E}{\partial b} \approx -0{,}1078.$$

Nuevos: $w_1'' \approx 1{,}233$, $w_2'' = 0$, $b'' \approx -0{,}767$.

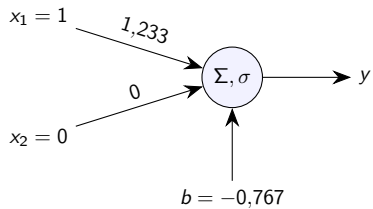
$z'' \approx 0{,}466$, $y'' \approx 0{,}614$, $E'' \approx 0{,}0744 < 0{,}0958$. ✓

Observa: el paso es *algo más pequeño* que el del Problema 5.1 ($-0{,}1078$ vs $-0{,}125$). Como y se acerca a 1, $y(1-y)$ baja: la neurona se va saturando y el aprendizaje se ralentiza.

Antes: $E \approx 0{,}096$



Después: $E \approx 0{,}074$



¿Por qué hace falta más de una neurona?

Una sola neurona sólo sabe trazar una recta en el plano de entradas. Eso basta para AND y OR, pero **no** para problemas como XOR o reconocer dígitos manuscritos.

Solución: conectar varias neuronas en *capas*. La salida de las neuronas de una capa es la entrada de la siguiente.

La red más pequeña que resuelve XOR es una red $2 \to 2 \to 1$: dos entradas, una capa oculta con dos neuronas y una neurona de salida.

La red $2 \to 2 \to 1$

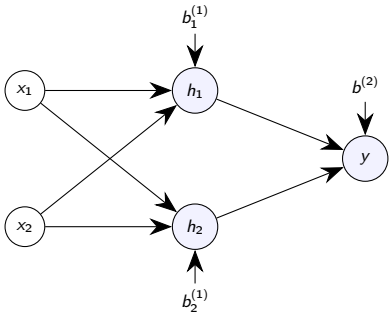
Capa oculta:

$$h_i = \sigma\Big(W_{i1}^{(1)} x_1 + W_{i2}^{(1)} x_2 + b_i^{(1)}\Big),\ i = 1, 2.$$

Capa de salida:

$$y = \sigma\Big(W_1^{(2)} h_1 + W_2^{(2)} h_2 + b^{(2)}\Big).$$

Cada arista lleva su propio peso. Cada neurona oculta o de salida lleva su propio sesgo. En total, esta red tiene $4 + 2 + 2 + 1 = 9$ parámetros entrenables.



Problema 6.1

Enunciado

Considera la red $2 \to 2 \to 1$ con pesos

$$W^{(1)} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \; b^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \; W^{(2)} = (1,1), \; b^{(2)} = -0{,}5,$$

y sigmoide en todas las neuronas. Calcula la salida y ante la entrada $(x_1, x_2) = (1, 0)$.

Problema 6.1 – solución

Solución

Capa oculta:

$$z_1^{(1)} = 1, \;\; h_1 = \sigma(1) \approx 0{,}731,$$

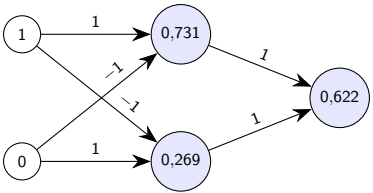
$$z_2^{(1)} = -1, \;\; h_2 = \sigma(-1) \approx 0{,}269.$$

Capa de salida:

$$z^{(2)} = 0{,}731 + 0{,}269 - 0{,}5 = 0{,}5,$$

$$y = \sigma(0{,}5) \approx 0{,}622.$$

$h_1 + h_2 = 1$ exactamente, gracias a $\sigma(-z) = 1 - \sigma(z)$.



Activaciones tras el forward pass.

Problema 6.2

Enunciado

Para la misma red del Problema 6.1, calcula la salida y ante la entrada $(x_1, x_2) = (1, 1)$. ¿Qué observas comparando con el resultado del Problema 6.1?

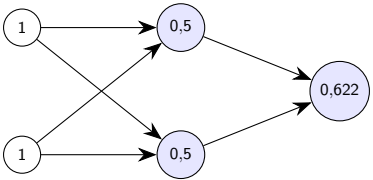
Problema 6.2 – solución

Solución

Capa oculta: $z_1^{(1)} = 1 - 1 = 0$, $h_1 = \sigma(0) = 0{,}5$; análogamente $h_2 = 0{,}5$.

Salida: $z^{(2)} = 0{,}5 + 0{,}5 - 0{,}5 = 0{,}5$, $y \approx 0{,}622$.

Observación: la salida es *la misma* que en el Problema 6.1, pero por motivos distintos: allí las neuronas ocultas eran *asimétricas* (una excitada, una inhibida) y aquí son *simétricas* (ambas a medias). La red da la misma respuesta a entradas distintas: estos pesos no distinguen $(1, 0)$ de $(1, 1)$.



Retropropagación: cómo aprende la red entera

Con una capa oculta los pesos son más numerosos. La técnica para calcular todos los gradientes es la *retropropagación*: aplicar la regla de la cadena **empezando por la salida** y propagando los errores hacia atrás capa a capa.

Idea clave: es exactamente el mismo reparto del error"que vimos en una neurona, sólo que la *influencia local* de un peso oculto pasa a través de la cadena de neuronas que vienen después.

Señal de error en la salida:

$$\delta^{(2)} = (y - \hat{y}) \cdot y(1 - y).$$

Gradientes en la salida:

$$\frac{\partial E}{\partial W_i^{(2)}} = \delta^{(2)} \cdot h_i, \qquad \frac{\partial E}{\partial b^{(2)}} = \delta^{(2)}.$$

Retropropagación a la capa oculta

Señal de error en la oculta: cada neurona oculta i recibe $\delta^{(2)} \cdot W_i^{(2)}$ desde la salida (su "parte"del error global), multiplicado por su derivada local $h_i(1 - h_i)$:

$$\delta_i^{(1)} = \delta^{(2)} \cdot W_i^{(2)} \cdot h_i(1 - h_i).$$

Gradientes en la oculta:

$$\frac{\partial E}{\partial W_{ij}^{(1)}} = \delta_i^{(1)} \cdot x_j, \qquad \frac{\partial E}{\partial b_i^{(1)}} = \delta_i^{(1)}.$$

Patrón general: en cada capa,

$$\text{gradiente del peso} = \underbrace{\delta_{\text{destino}}}_{\substack{\text{error que llega} \\ \text{a la neurona}}} \times \underbrace{\text{entrada del peso}}_{\substack{\text{influencia} \\ \text{local}}}.$$

Aplicable a cualquier número de capas.

Problema 7.1

Enunciado

Para la red del Problema 6.1 (recuerda: $y \approx 0{,}622$, $h_1 \approx 0{,}731$, $h_2 \approx 0{,}269$) con objetivo $\hat{y} = 1$, calcula los gradientes en la **capa de salida**: la señal $\delta^{(2)}$ y las derivadas parciales $\partial E/\partial W_1^{(2)}$, $\partial E/\partial W_2^{(2)}$, $\partial E/\partial b^{(2)}$.

Problema 7.1 – solución

Solución

Señal de error en la salida:

$$\delta^{(2)} = (0{,}622 - 1)(0{,}622)(1 - 0{,}622) \approx (-0{,}378)(0{,}235) \approx -0{,}0888.$$

Gradientes:

$$\frac{\partial E}{\partial W_1^{(2)}} = \delta^{(2)} h_1 \approx (-0{,}0888)(0{,}731) \approx -0{,}0649,$$

$$\frac{\partial E}{\partial W_2^{(2)}} = \delta^{(2)} h_2 \approx (-0{,}0888)(0{,}269) \approx -0{,}0239,$$

$$\frac{\partial E}{\partial b^{(2)}} = \delta^{(2)} \approx -0{,}0888.$$

Todos negativos $\Rightarrow$ los tres pesos hay que *subirlos* para reducir el error. Y el peso conectado a h_1 cambia más que el conectado a h_2 porque $h_1 > h_2$: h_1 "tuvo más voz.$^{\text{en}}$ la decisión final.

Problema 7.2

Enunciado

Para la misma red del Problema 7.1, calcula ahora los gradientes en la **capa oculta**: las señales $\delta_1^{(1)}, \delta_2^{(1)}$ y las derivadas parciales $\partial E/\partial W_{ij}^{(1)}$, $\partial E/\partial b_i^{(1)}$ para $i, j \in \{1, 2\}$.

Datos: entrada $(x_1, x_2) = (1, 0)$, $\delta^{(2)} \approx -0{,}0888$, $W^{(2)} = (1, 1)$, $h_1 \approx 0{,}731$, $h_2 \approx 0{,}269$.

Problema 7.2 – solución

Solución

Señales: $\delta_i^{(1)} = \delta^{(2)} W_i^{(2)} h_i(1-h_i)$, así que

$$\delta_1^{(1)} \approx (-0{,}0888)(1)(0{,}731)(0{,}269) \approx -0{,}0175,$$

$$\delta_2^{(1)} \approx (-0{,}0888)(1)(0{,}269)(0{,}731) \approx -0{,}0175.$$

(Coinciden por la simetría $h_1(1-h_1) = h_2(1-h_2)$ ya que $h_1 + h_2 = 1$.)
Con $x_1 = 1$, $x_2 = 0$:

$$\frac{\partial E}{\partial W_{11}^{(1)}} \approx -0{,}0175, \quad \frac{\partial E}{\partial W_{12}^{(1)}} = 0,$$

$$\frac{\partial E}{\partial W_{21}^{(1)}} \approx -0{,}0175, \quad \frac{\partial E}{\partial W_{22}^{(1)}} = 0,$$

$$\frac{\partial E}{\partial b_1^{(1)}} \approx -0{,}0175, \quad \frac{\partial E}{\partial b_2^{(1)}} \approx -0{,}0175.$$

Los pesos asociados a $x_2 = 0$ no se mueven: con entrada nula no hay manera de saber su efecto.

¿Cuándo deja de aprender la red?

El descenso del gradiente itera $w \leftarrow w - \eta \cdot \partial E / \partial w$. La red está en un *estado estable* (o punto fijo) cuando **todos los gradientes son nulos**: las actualizaciones no cambian nada.

Geométricamente, un estado estable es un punto crítico de la pérdida: mínimo, máximo o punto silla. Bien diseñada la red, los puntos a los que llega el descenso son mínimos.

En el caso lineal (sin activación) podemos resolver el estado estable analíticamente. Y obtenemos algo familiar: la regresión por mínimos cuadrados.

Una neurona lineal

Para una neurona lineal $y = wx$ entrenada sobre datos $\{(x_k, \hat{y}_k)\}_{k=1}^{N}$ con pérdida cuadrática total

$$E(w) = \tfrac{1}{2} \sum_{k=1}^{N} (wx_k - \hat{y}_k)^2,$$

el estado estable cumple $dE/dw = 0$:

$$\sum_k (wx_k - \hat{y}_k) x_k = 0 \quad \Longrightarrow \quad w^* = \frac{\sum_k \hat{y}_k \, x_k}{\sum_k x_k^2}.$$

Esa fórmula es la pendiente de la recta de regresión por mínimos cuadrados que pasa por el origen.

Problema 8.1

Enunciado

Para una neurona lineal $y = wx$ entrenada con pérdida cuadrática:

1. Sobre los datos $\{(1,2), (2,4), (3,6)\}$, encuentra w^* y $E(w^*)$.
2. Sobre los datos $\{(1, 2{,}1), (2, 3{,}9), (3, 6{,}1)\}$, encuentra w^* y $E(w^*)$. ¿Qué interpretación tiene w^*?

Problema 8.1 – solución

Solución

Caso (1): $\sum x_k^2 = 14$, $\sum \hat{y}_k x_k = 28$, así que $w^* = 28/14 = 2$. Como los puntos están sobre la recta $y = 2x$, $E(w^*) = 0$.

Caso (2): $\sum x_k^2 = 14$, $\sum \hat{y}_k x_k = 2{,}1 + 7{,}8 + 18{,}3 = 28{,}2$, $w^* \approx 2{,}014$.

$$E(w^*) \approx \tfrac{1}{2}(0{,}00735 + 0{,}01653 + 0{,}00327) \approx 0{,}0136.$$

w^* es la pendiente de la *regresión por mínimos cuadrados* a través del origen. La neurona lineal entrenada por descenso del gradiente está calculando una regresión.

Problema 8.2

Enunciado

Considera ahora una neurona afín $y = wx + b$ entrenada con pérdida cuadrática sobre los datos $\{(0, 1), (1, 3), (2, 5)\}$. Encuentra el estado estable (w^*, b^*) y el valor de la pérdida en ese punto. ¿Reconoces estos puntos?

Problema 8.2 – solución

Solución

Las dos ecuaciones de estado estable son

$$\sum (wx_k + b - \hat{y}_k)x_k = 0, \qquad \sum (wx_k + b - \hat{y}_k) = 0.$$

Con $\sum x_k = 3$, $\sum x_k^2 = 5$, $\sum \hat{y}_k = 9$, $\sum \hat{y}_k x_k = 13$, $N = 3$, queda el sistema

$$5w + 3b = 13, \qquad 3w + 3b = 9 \iff w + b = 3.$$

Restando: $2w = 4$, así que $w^* = 2$, $b^* = 1$. Los tres puntos están sobre $y = 2x + 1$, así que $E(w^*, b^*) = 0$. La neurona afín *es* regresión lineal en su forma completa.

Lo que habéis hecho

En estos 16 problemas habéis ejecutado el algoritmo entero que entrena cualquier red neuronal moderna:

1. Propagar entradas hacia adelante (problemas 1, 2, 4, 6).
2. Comparar la salida con un objetivo (problema 4).
3. Calcular gradientes con la regla de la cadena (problemas 5, 7).
4. Actualizar pesos en sentido contrario al gradiente (problemas 5, 7).
5. Iterar hasta llegar a un estado estable (problema 8).

La diferencia con una red real es la **escala**: aquí lo hacéis a mano para una neurona o tres; en una red real hay millones de pesos y miles de millones de pasos. Pero el algoritmo es exactamente el mismo.

Próxima sesión: laboratorio virtual con red ajustable, nube de puntos y descenso del gradiente en directo.