

Aprendizaje profundo

REDES DENSAS (PERCEPTRÓN MULTICAPA)

Gibran Fuentes-Pineda

Agosto 2023

Aproximación de la compuerta lógica XOR (\oplus)

- Minsky y Papert mostraron que no era posible aproximar una compuerta lógica XOR con una sola neurona con función de activación escalón unitario

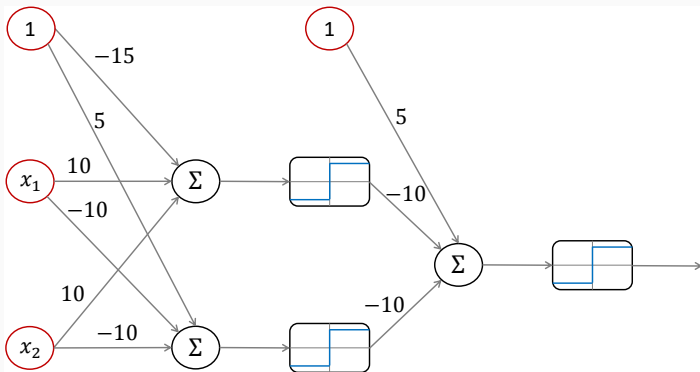
x_1	x_2	$x_1 \oplus x_2$
0	0	0
1	0	1
0	1	1
1	1	0

- Es aproximarla con múltiples neuronas conectadas entre sí (ver <https://playground.tensorflow.org/>).

Múltiples neuronas en capa

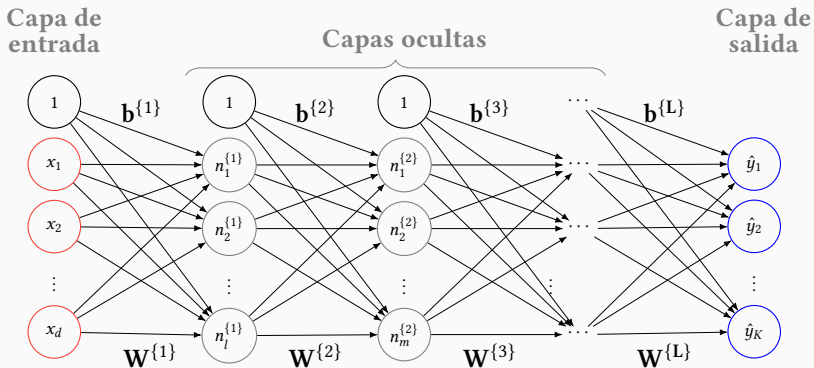
- Podemos usar la fórmula

$$x_1 \oplus x_2 = \neg((x_1 \wedge x_2) \vee \neg(x_1 \vee x_2))$$

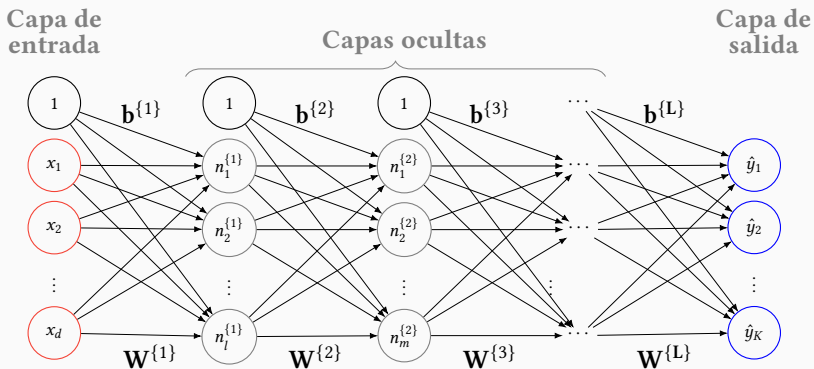


Red neuronal densa

- Está compuesta por capas densas o completamente conectadas



Red neuronal densa



Red neuronal densa: características

- Aproximadores universales (con 1 sola capa oculta con un número finito de neuronas^{1,2})
- Frecuentemente sobreparametrizados³
- Usualmente empleados como bloques de clasificación (no tan profundos) en conjunto con otros tipos de capas

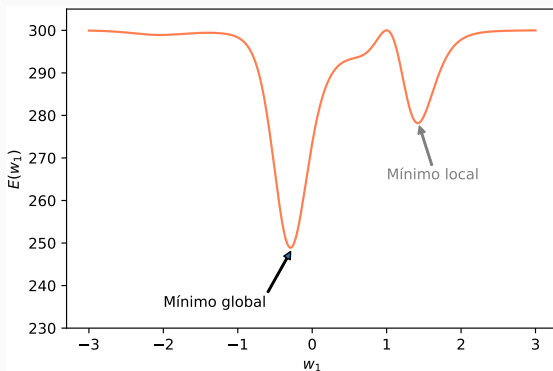
¹ Cybenko. Approximation by Superpositions of a Sigmoidal Function, 1989

² Hornik et al. Multilayer Feedforward Networks are Universal Approximators, 1989.

³ Allen-Zhu et al. Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers, 2020.

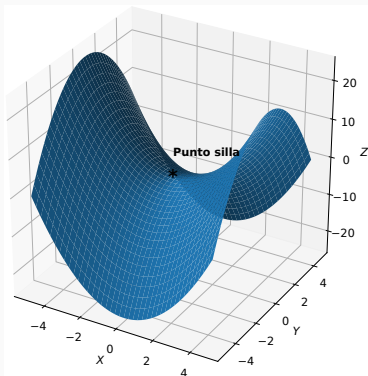
Pérdida para redes neuronales multicapa (1)

- Para múltiples capas de neuronas la función de pérdida no es convexa (ver <https://losslandscape.com/explorer>)



Pérdida para redes neuronales multicapa (2)

- Pueden existir varios puntos sillas



Entrenamiento de redes neuronales multicapa

- Aunque en problemas convexos SGD aproxima al GD, en la práctica se ha observado que en el entrenamiento de redes neuronales SGD encuentra mejores soluciones, especialmente con minilotes pequeños^{4,5,6}
- Problema: calcular eficientemente las derivadas parciales respecto a los pesos y sesgos de las capas ocultas

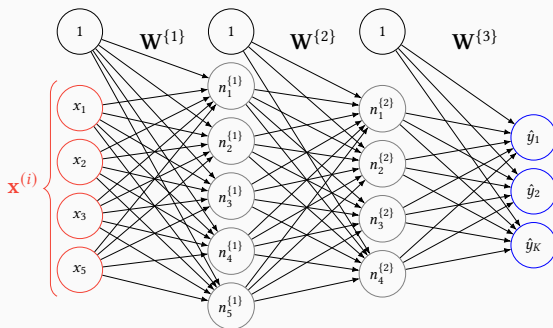
⁴ Kleinberg et al. An Alternative View: When Does SGD Escape Local Minima?, 2018

⁵ Zhu et al. The Anisotropic Noise in Stochastic Gradient Descent: Its Behavior of Escaping from Sharp Minima and Regularization Effects, 2019.

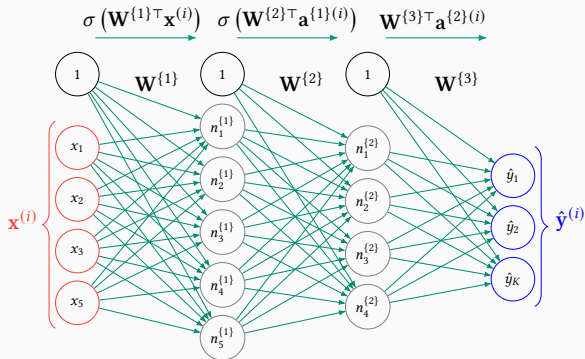
⁶ Keskar et al. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima, 2017.

Propagación hacia adelante de red densa (1)

- Considera una red densa con 1 capa de entrada, 2 capas ocultas con 5 y 4 neuronas con función de activación sigmoide y 3 neuronas de salida con función de activación lineal.



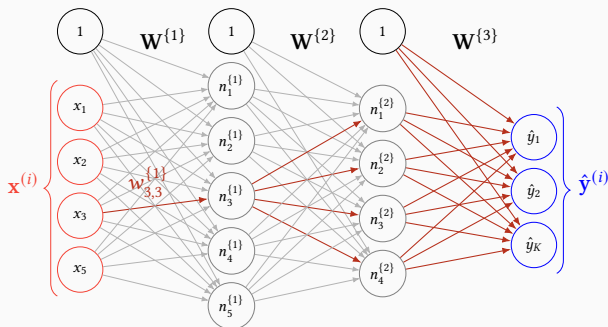
Propagación hacia adelante de red densa (2)



$$\hat{\mathbf{y}}^{(i)} = \mathbf{W}^{(3)\top} \cdot \sigma \left(\mathbf{W}^{(2)\top} \cdot \overbrace{\sigma \left(\mathbf{W}^{(1)\top} \cdot \mathbf{x}^{(i)} \right)}^{\mathbf{a}^{(1)(i)}} \right)$$

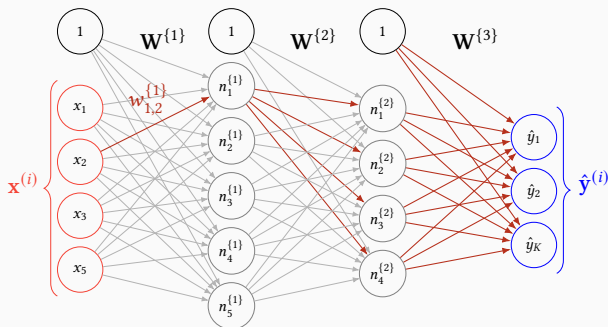
Propagación hacia adelante de red densa (3)

- ¿Cómo calculo los gradientes de la función de pérdida respecto a los pesos de la primera capa?



Propagación hacia adelante de red densa (4)

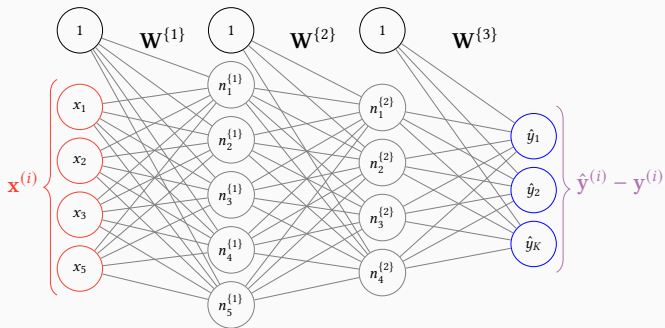
- ¿Cómo calculo los gradientes de la función de pérdida respecto a los pesos de la primera capa?



1. Propagamos cada entrada $\mathbf{x}^{(i)}$ hacia adelante para generar la correspondiente salida $\hat{\mathbf{y}}^{(i)}$
2. Calculamos derivadas parciales de la pérdida respecto a cada peso y sesgo capa por capa, empezando con la de salida y propagándolas hacia atrás para calcular las de la capa anterior

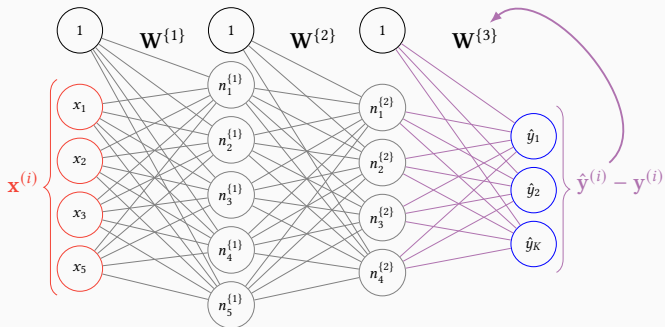
Propagación hacia atrás (1)

- Presuponiendo que se busca minimizar la función de pérdida de error cuadrático medio (ECM)



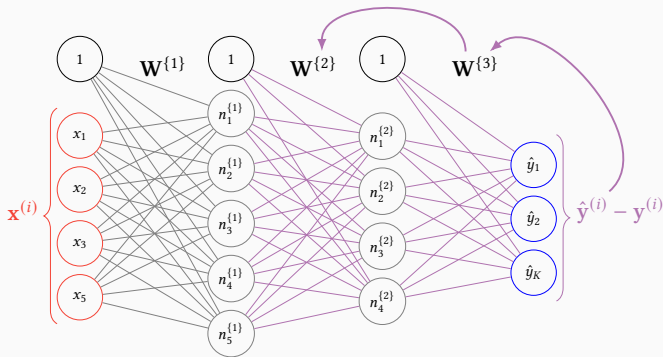
Propagación hacia atrás (2)

- Se calcula el error de la predicción y se propaga hacia la última capa



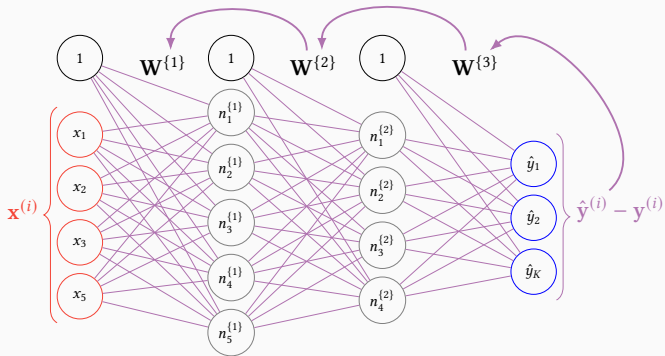
Propagación hacia atrás (3)

- Se calcula el error de la última capa usando el error de la predicción y se propaga hacia la penúltima capa



Propagación hacia atrás (4)

- Se calcula el error de la penúltima capa usando el error de la última capa y se propaga hacia la primera capa



Ejemplo: propagación hacia adelante

- Considera una red densa con 1 capa de entrada, 1 capa oculta con M neuronas sigmoide y 1 neurona de salida con activación lineal.
- La propagación hacia adelante estaría dada por

$$\mathbf{z}^{\{2\}} = \mathbf{W}^{\{1\}} \cdot \mathbf{x}^{(i)}$$

$$\mathbf{a}^{\{2\}} = \phi(\mathbf{z}^{\{2\}})$$

$$\mathbf{z}^{\{3\}} = \mathbf{W}^{\{2\}} \cdot \mathbf{a}^{\{2\}}$$

$$\hat{y}^{(i)} = \phi(\mathbf{z}^{\{3\}})$$

- Presuponemos una tarea de regresión y la función de pérdida

$$ECM(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2} \sum_{i=1}^N (y^{(i)} - \hat{y}^{(i)})^2$$

Ejemplo: retropropagación (1)

- Calculamos el gradiente de la función de pérdida con respecto a $\mathbf{W}^{\{2\}}$ de la siguiente forma

$$\begin{aligned}\frac{\partial ECM}{\partial \mathbf{W}^{\{2\}}} &= \frac{\partial \sum_i \frac{1}{2} (y^{(i)} - \hat{y}^{(i)})^2}{\partial \mathbf{W}^{\{2\}}} \\ &= \frac{\sum_i \partial \frac{1}{2} (y^{(i)} - \hat{y}^{(i)})^2}{\partial \mathbf{W}^{\{2\}}} \\ \frac{\partial \frac{1}{2} (y^{(i)} - \hat{y}^{(i)})^2}{\partial \mathbf{W}^{\{2\}}} &= (y^{(i)} - \hat{y}^{(i)}) \cdot \left(-\frac{\partial \hat{y}^{(i)}}{\partial \mathbf{W}^{\{2\}}} \right) \\ &= (y^{(i)} - \hat{y}^{(i)}) \cdot \left(-\frac{\partial \hat{y}^{(i)}}{\partial z^{\{3\}}} \cdot \frac{\partial z^{\{3\}}}{\partial \mathbf{W}^{\{2\}}} \right) \\ &= \underbrace{-(y^{(i)} - \hat{y}^{(i)}) \cdot \frac{\partial \hat{y}^{(i)}}{\partial z^{\{3\}}}}_{\delta^{\{3\}}} \cdot \mathbf{a}^{\{2\}}\end{aligned}$$

Ejemplo: propagación hacia atrás (1)

- Calculamos el gradiente de la función de pérdida respecto a $\mathbf{W}^{\{1\}}$ de la siguiente forma

$$\begin{aligned}\frac{\partial ECM}{\partial \mathbf{W}^{\{1\}}} &= \frac{\partial \sum_i \frac{1}{2} (y^{(i)} - \hat{y}^{(i)})^2}{\partial \mathbf{W}^{\{1\}}} \\ &= \frac{\sum_i \partial \frac{1}{2} (y^{(i)} - \hat{y}^{(i)})^2}{\partial \mathbf{W}^{\{1\}}} \\ \frac{\partial \frac{1}{2} (y^{(i)} - \hat{y}^{(i)})^2}{\partial \mathbf{W}^{\{1\}}} &= (y^{(i)} - \hat{y}^{(i)}) \left(-\frac{\partial \hat{y}^{(i)}}{\partial \mathbf{W}^{\{1\}}} \right) \\ &= (y^{(i)} - \hat{y}^{(i)}) \left(-\frac{\partial \hat{y}^{(i)}}{\partial \mathbf{z}^{\{3\}}} \cdot \frac{\partial \mathbf{z}^{\{3\}}}{\partial \mathbf{W}^{\{1\}}} \right) \\ &= \underbrace{-(y^{(i)} - \hat{y}^{(i)}) \cdot \frac{\partial \hat{y}^{(i)}}{\partial \mathbf{z}^{\{3\}}}}_{\delta^{\{3\}}} \cdot \frac{\partial \mathbf{z}^{\{3\}}}{\partial \mathbf{W}^{\{1\}}} = \delta^{\{3\}} \cdot \frac{\partial \mathbf{z}^{\{3\}}}{\partial \mathbf{W}^{\{1\}}}\end{aligned}$$

Ejemplo: propagación hacia atrás (2)

$$\begin{aligned} &= \delta^{\{3\}} \cdot \left(\overbrace{\frac{\partial z^{\{3\}}}{\partial a^{\{2\}}}}^{w^{\{2\}}} \cdot \frac{\partial a^{\{2\}}}{\partial w^{\{1\}}} \right) \\ &= \delta^{\{3\}} \cdot w^{\{2\}} \cdot \left(\frac{\partial a^{\{2\}}}{\partial w^{\{1\}}} \right) \\ &= \delta^{\{3\}} \cdot w^{\{2\}} \cdot \left(\frac{\partial a^{\{2\}}}{\partial z^{\{2\}}} \cdot \underbrace{\frac{\partial z^{\{2\}}}{\partial w^{\{1\}}}}_{x^{(i)}} \right) \\ &= \delta^{\{3\}} \cdot w^{\{2\}} \cdot \frac{\partial a^{\{2\}}}{\partial z^{\{2\}}} \cdot x^{(i)} \end{aligned}$$

Diferenciación automática

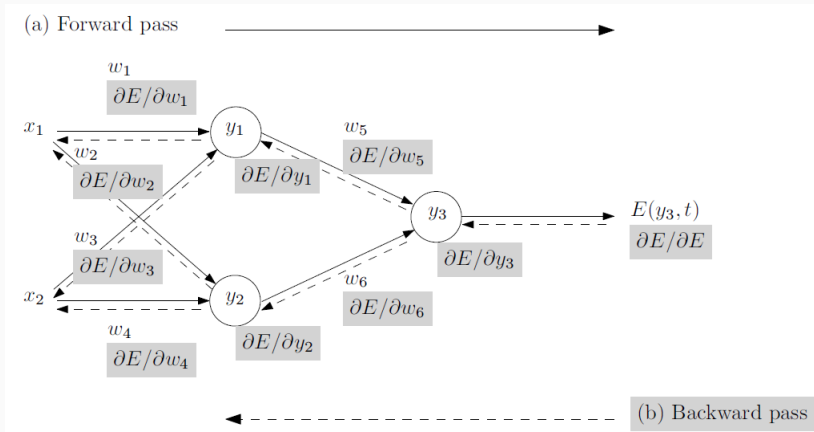


Imagen tomada de Baydin et. al. Automatic Differentiation in Machine Learning: a Survey, 2018.

Características generales de las redes neuronales densas

- Aproximadores universales (con 1 sola capa oculta con un número finito de neuronas^{7,8)}
- Frecuentemente sobreparametrizados⁹
- Usualmente empleados como bloques de clasificación (no tan profundos) en conjunto con otros tipos de capas

⁷ Cybenko. Approximation by Superpositions of a Sigmoidal Function, 1989

⁸ Hornik et al. Multilayer Feedforward Networks are Universal Approximators, 1989.

⁹ Allen-Zhu et al. Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers, 2020.