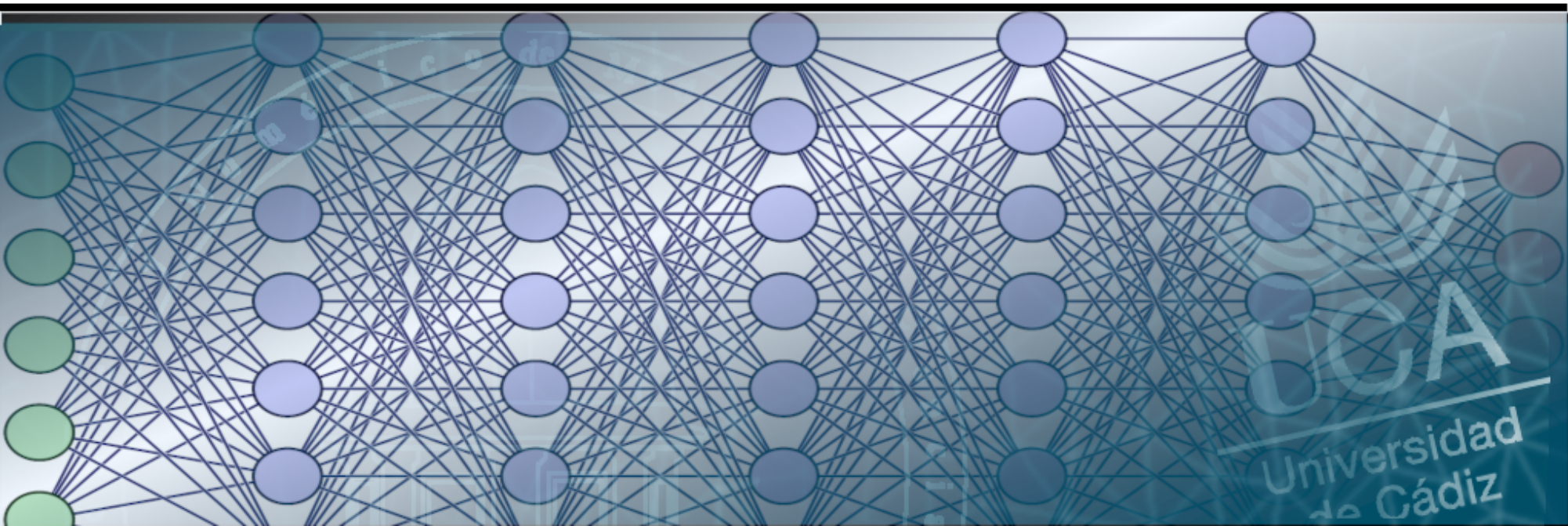
Neural Networks

Álex Pérez Fernández, Rafa Rodríguez Galván

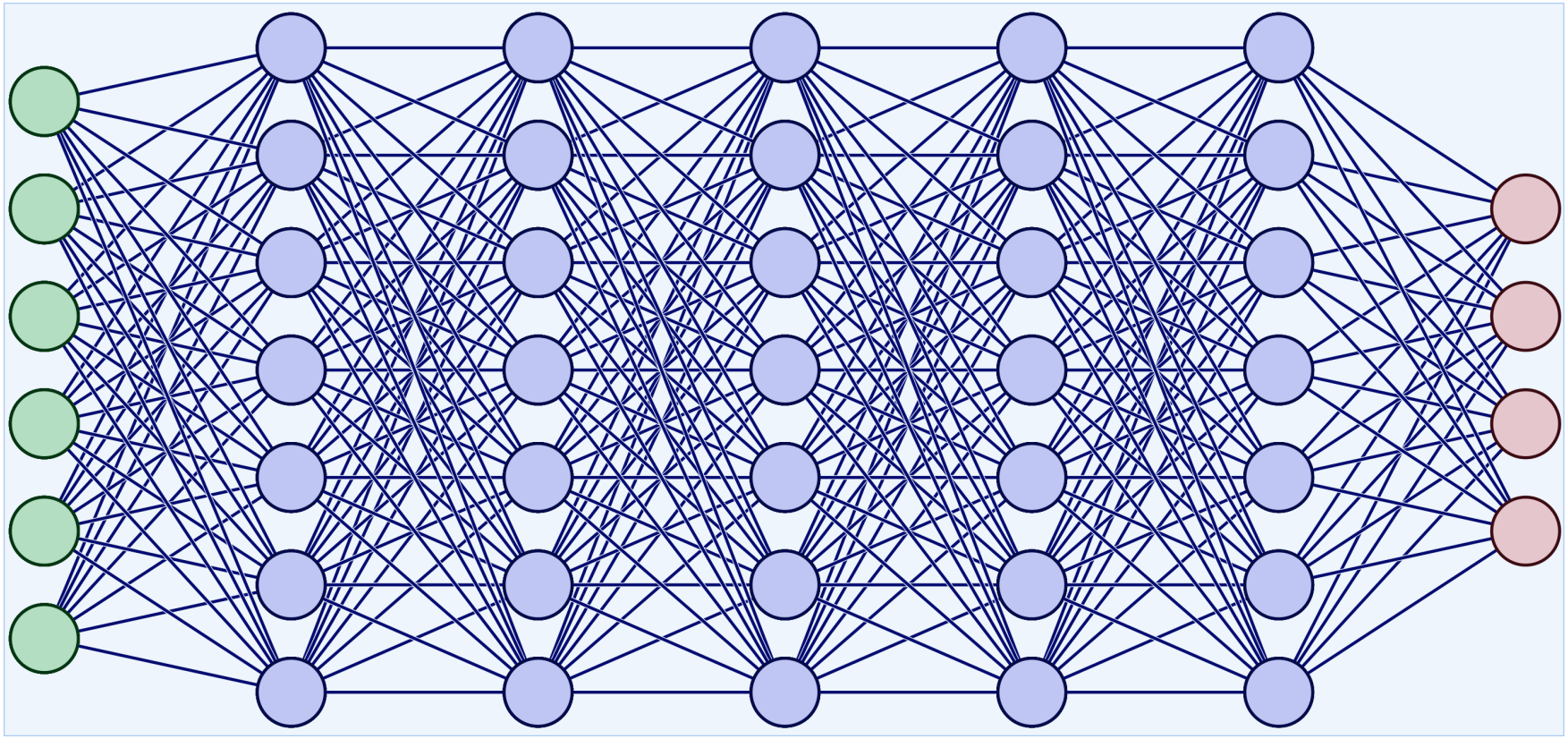
March 2, 2024



Section 1

Neural Networks

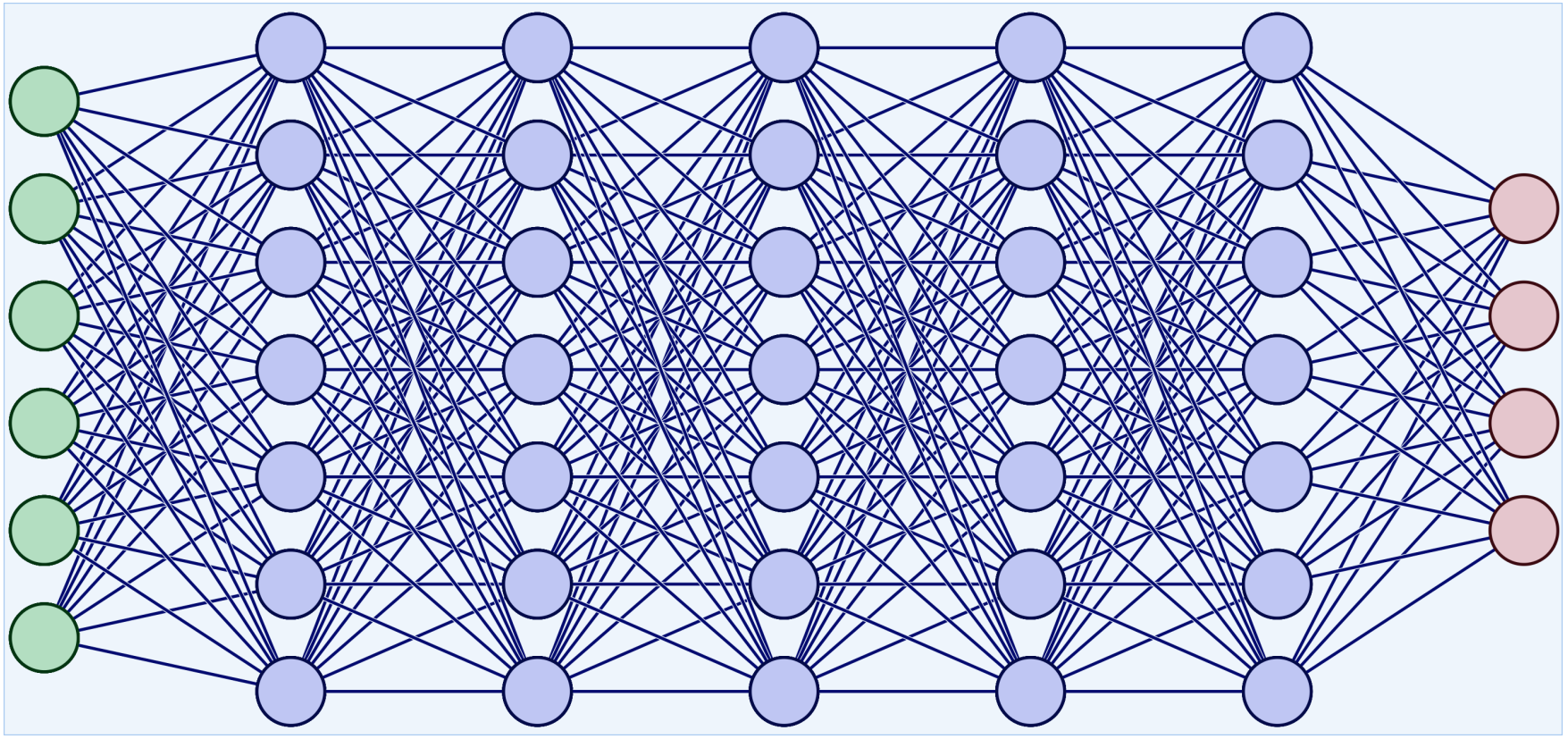
Neural Networks...



...are mathematical artifacts:

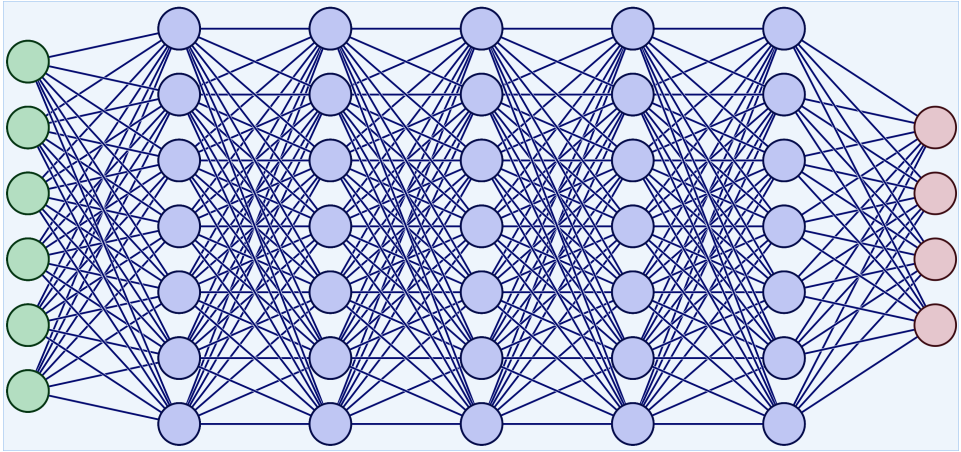
$$x \mapsto f_1(x) \mapsto f_2 \circ f_1(x) \mapsto \cdots \mapsto f_L \circ \cdots \circ f_2 \circ f_1(x) = y$$

Neural Networks...



...are mathematical artifacts:

$$x \mapsto f_1(x) \mapsto f_2 \circ f_1(x) \mapsto \cdots \mapsto f_L \circ \cdots \circ f_2 \circ f_1(x) = y$$



Definición:
una **Red Neuronal** (*RN* o *NN*) es una función $f_{NN} : \mathbb{R}^n \to \mathbb{R}^m$ del tipo:

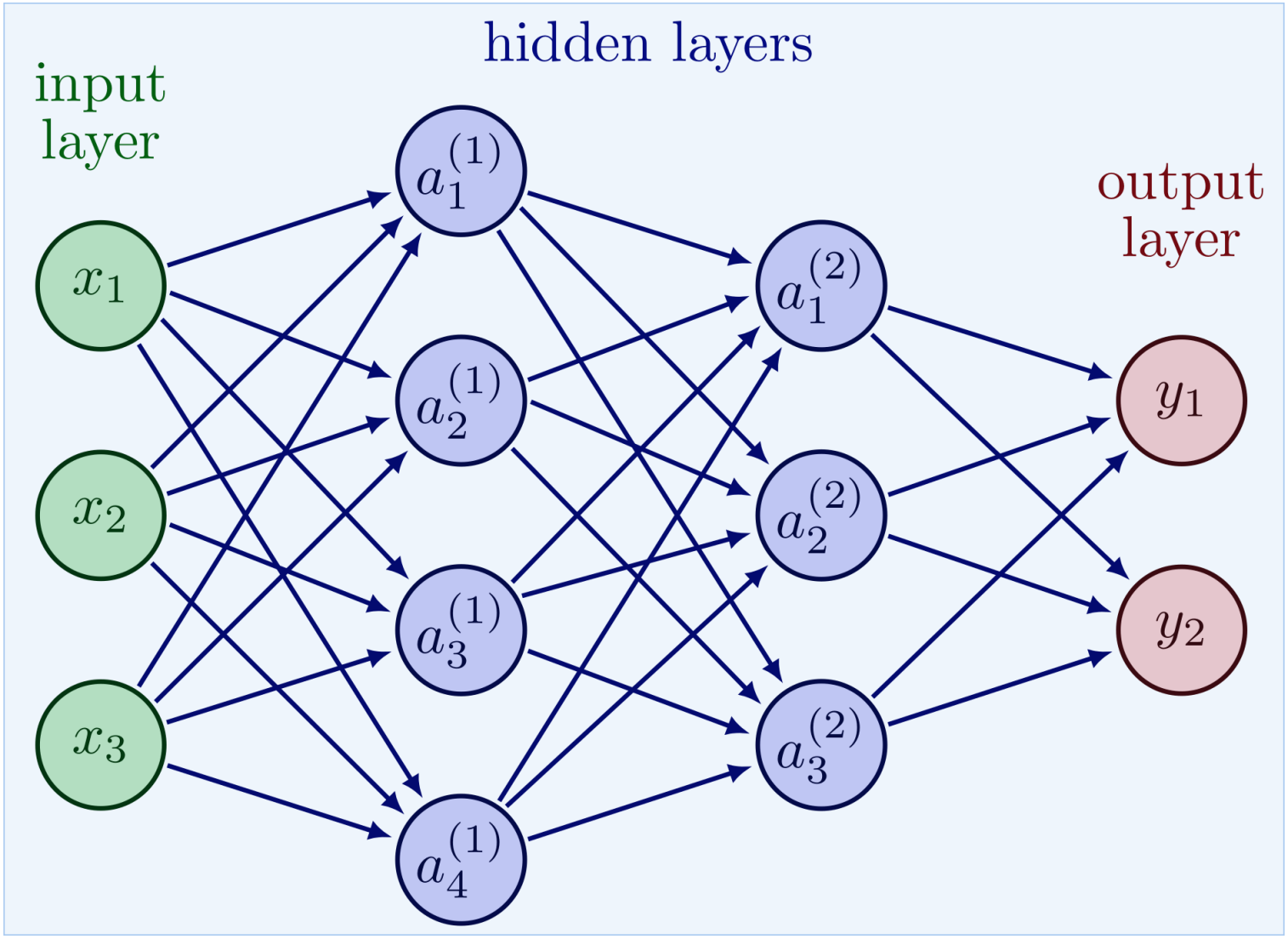
$$y = f_{NN}(x) = f_L \circ \cdots \circ f_2 \circ f_1(x).$$

Donde...

- Cada función f_i se llama una **capa** (entrada $\to$ oculta $\to$ salida)
- Cada capa f_i está compuesta por un nº variable de **neuronas**
- Cada neurona depende un conjunto de **parámetros**, que determinarán a la RN

$\star$ La RN de la figura se dice de tipo «*feed forward*» o prealimentada

Un ejemplo



$f_{NN} : \mathbb{R}^3 \to \mathbb{R}^2$con 2 capas ocultas de 4 y 3 neuronas

Neurona o perceptrón simple

Cada neurona j de una capa oculta f_i (o de salida y_i) es una función[1]:

$$\mathrm{x} \in \mathbb{R}^{N_i} \to a_j^{(i)}(\mathrm{x}) \in \mathbb{R},$$

composición de

- una función afín con parámetros $\mathrm{w} = (w_1, \ldots, w_{N_i})$ y b
- una función no lineal σ, llamada «función de activación»

$$a_j^{(i)}(\mathrm{x}) = \sigma(w_1 x_1 + w_2 x_2 + \cdots + w_{N_i} x_{N_i} + b) =$$
$$= \sigma\left(\sum_{k=1}^{N_i} w_k x_k + b\right) = \sigma(\mathrm{w} \cdot \mathrm{x} + b)$$

[1] Donde N_i es el número de neuronas de la capa $i - 1$

Con más propiedad...

Para aligerar la notación se omitieron los índices correspondientes a la capa, i, y a la neurona, j. Debería ser:

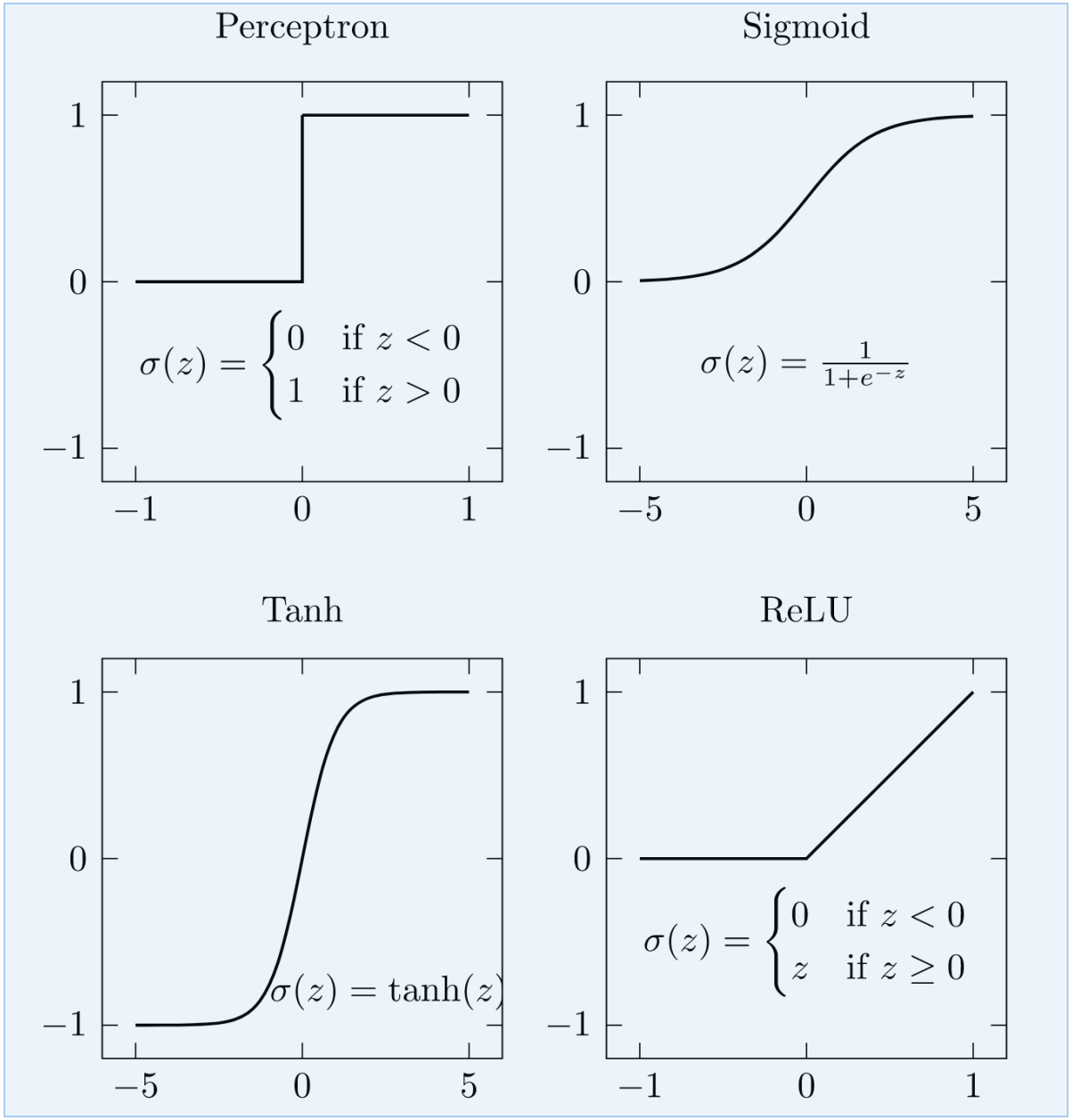
$$a_j^{(i)}(\mathrm{x}) = \sigma\Big(\sum_{k=1}^{N_i} w_{j,k}^{(i)} x_k + b_j^{(i)}\Big).$$

Así, si $\mathrm{W}^{(i)}$ denota a la matriz de valores $w_{j,k}^{(i)}$, y $\mathrm{b}^{(i)}$ es el vector $(b_j^{(i)})$, podemos escribir a tod la capa i como:

$$\mathrm{f}_i(\mathrm{x}) = \boldsymbol{\sigma}_i(\mathrm{W}^{(i)}\mathrm{x} + \mathrm{b}^{(i)})$$

La RN está determinada por los parámetros $\mathrm{W}^{(i)}$, los desplazamientos $\mathrm{b}^{(i)}$ y las funciones de activación $\boldsymbol{\sigma}_i$

Funciones de activación



Section 2

Ajuste de los parámetros

Aprendizaje supervisado

- En redes supervisadas, se dispone de datos de entrenamiento, formados por un conjunto de valores de entrada $\widehat{x}$, junto con los resultados asociados, $\widehat{y}$
- Es usual disponer además de datos de test, `xtest`, `ytest`

Función de coste y entrenamiento de la red neuronal

- El proceso de entrenamiento de la red neuronal consiste en determinar los parámetros (pesos, $w_{j,k}^{(i)}$ y desplazamientos, $b_j^{(i)}$) que minimizan un funcional, "función de coste", sobre los datos de entrenamiento:

$$\Theta^* = \text{argmin}\{J(\Theta; \widehat{x}, \widehat{y}), \quad \Theta = \left(w_{j,k}^{(i)}, b_j^{(i)}\right)\}$$

- La función de coste varía con cada tipo de red neuronal. Por ejemplo, en problemas de regresión se suelen usar mínimos cuadrados ("MSE: minimum mean square error"):

$$J(\Theta; \widehat{x}, \widehat{y}) = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} (\widehat{y}_i - f_{NN}(\widehat{x}_i))^2$$

Algoritmos de minimización

- Dificultades para la minimización: complejidad del funcional de coste, grandes valores de N_{data}
- Enormes requerimientos de cálculo para el entrenamiento, uso de grandes ordenadores, GPUs
- Se suelen utilizar algoritmos de tipo descenso de gradiente[2]

$$\Theta_{k+1} = \Theta_k - \ell_r \nabla_\Theta J(\Theta_k; \hat{x}, \hat{y}), \quad \ell_r \text{: "Learning Rate"}$$

- Necesidad de derivar de forma eficiente: diferenciación automática[3]
- Algoritmos de gradiente estocástico[4]: en cada paso, se calcula el gradiente pero sólo en un subconjunto aleatorio de datos

[2] `https://en.wikipedia.org/wiki/Gradient_descent`
[3] `https://en.wikipedia.org/wiki/Automatic_differentiation`
[4] `https://en.wikipedia.org/wiki/Stochastic_gradient_descent`