ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN PROFESIONAL

Diplomado en Programación y Aplicaciones de Python

Aplicaciones en Ciencia de Datos e Inteligencia Artificial

Profesor:
Francisco Pérez Galarce

Evaluaciones

Evaluación escrita de conceptos	**20%**	
• 2 controles (contenido teórico e implementación)	10%	**2 de 2**
• Prueba final del curso	10%	**Próxima semana (hasta el 22/12)**
Desarrollo de tareas de programación	**80%**	
• 2 actividades de implementación en clases	20%	**2 de 2**
• 2 Mini proyectos	40%	**1 de 2** **Próxima semana (hasta el 22/12)**
• Repositorio en Github	20%	**Próxima semana (hasta el 22/12)**

Fechas de evaluaciones

Fecha	Actividad/Evaluación
29-10-24	Introducción al aprendizaje de máquina: exploración y procesamiento de datos con Python **Actividad 1 (No evaluada)**
05-11-24	Aprendizaje supervisado con Python : regresiones **Actividad 2 (Evaluada)**
12-11-24	**Actividad 2 (Evaluada)** **Control 1**
19-11-24	Aprendizaje supervisado con Python naive Bayes y métricas de evaluación **Mini Proyecto 1**
26-11-24	Aprendizaje supervisado con Python : decision tree, random forest **Mini Proyecto 1**
03-12-24	Aprendizaje no supervisado con Python: k-means **Actividad 4 (Evaluada) – Control 2**
10-12-24	Redes Neuronales I
17-12-24	Redes Neuronales II **Mini Proyecto 2 / Prueba Final / Portafolio en Github**

- Introducción al aprendizaje de máquinas.
- Procesamiento de Datos
- Aprendizaje supervisado.
- Aprendizaje no supervisado
- Redes Neuronales

Hasta hoy!!

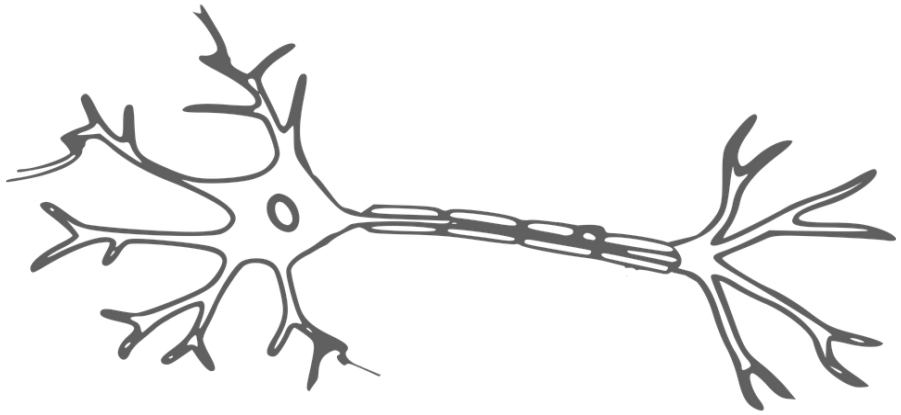
Red neuronal artificial

Profesor: **Francisco Pérez Galarce**

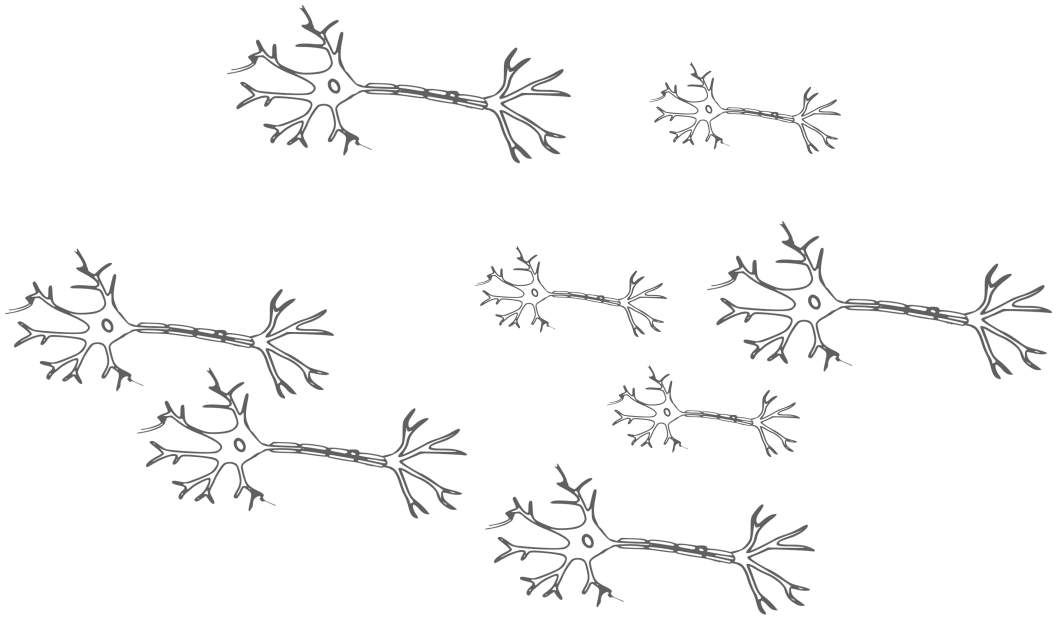
Neurona artificial

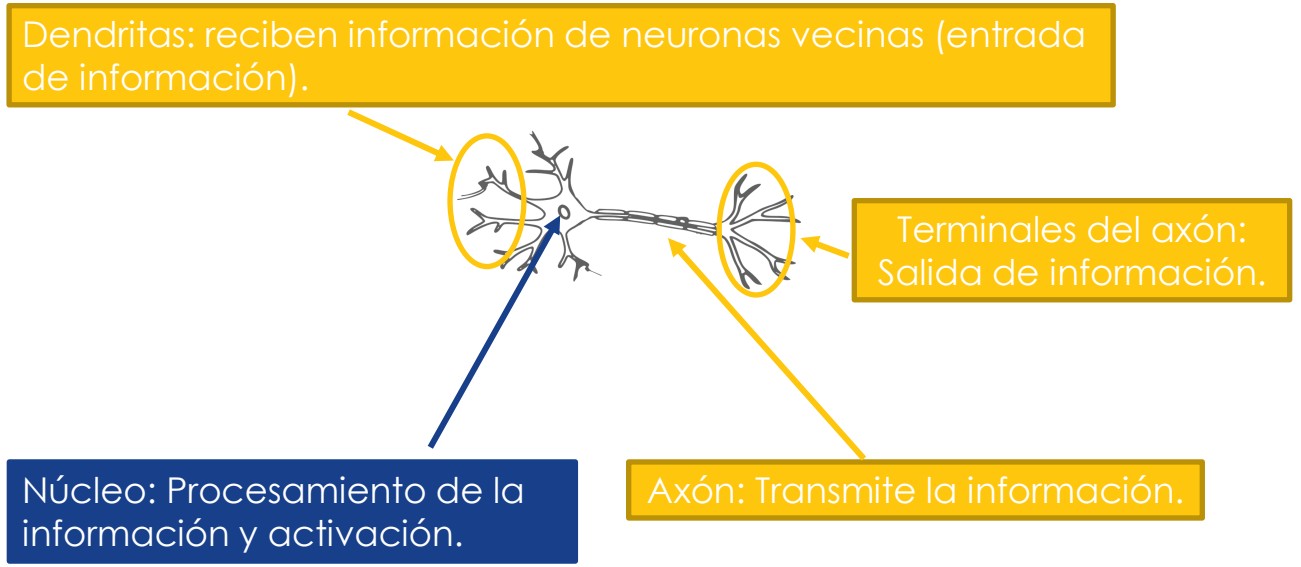
Inicialmente inspiradas en funcionamiento de una neurona humana

Emite señales como impulsos (modelo matemático trata de imitar)

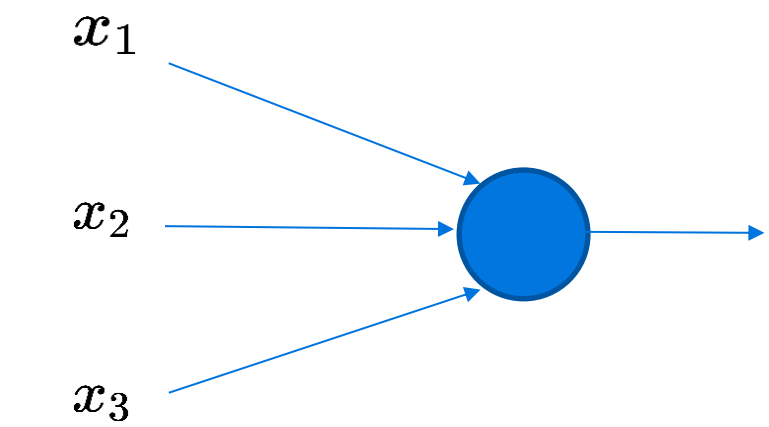


Neurona




Dendritas: reciben información de neuronas vecinas (entrada de información).
Terminales del axón: Salida de información.
Núcleo: Procesamiento de la información y activación.
Axón: Transmite la información.

Neurona artificial

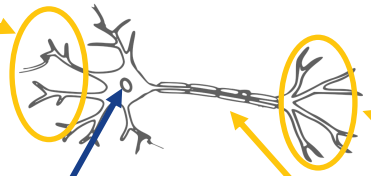


Entrada: datos

Activación

Salida: predicción

Dendritas: reciben información de neuronas vecinas (entrada de información).

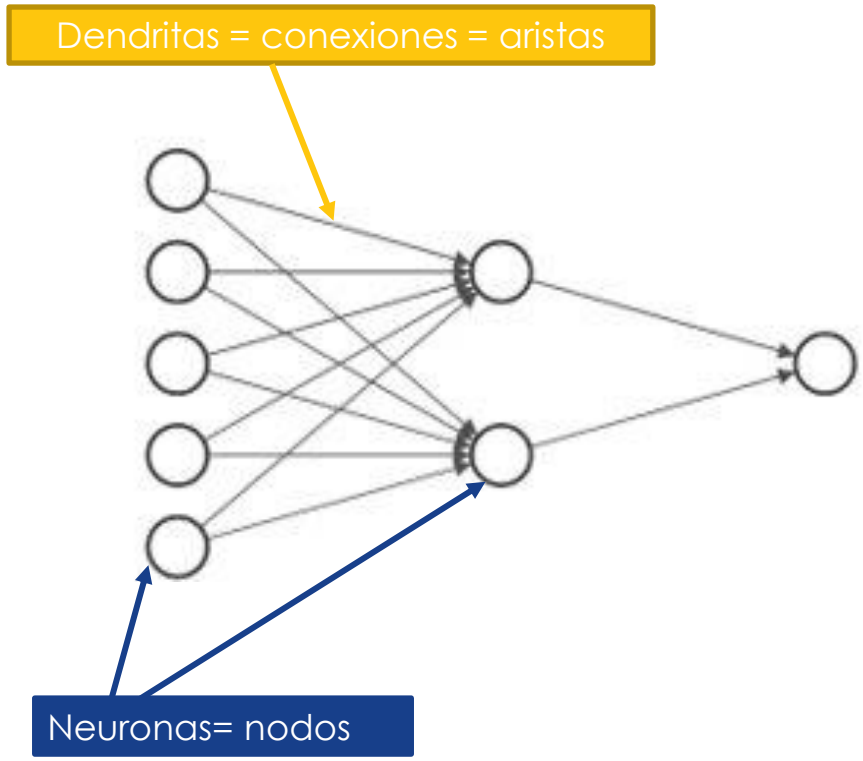


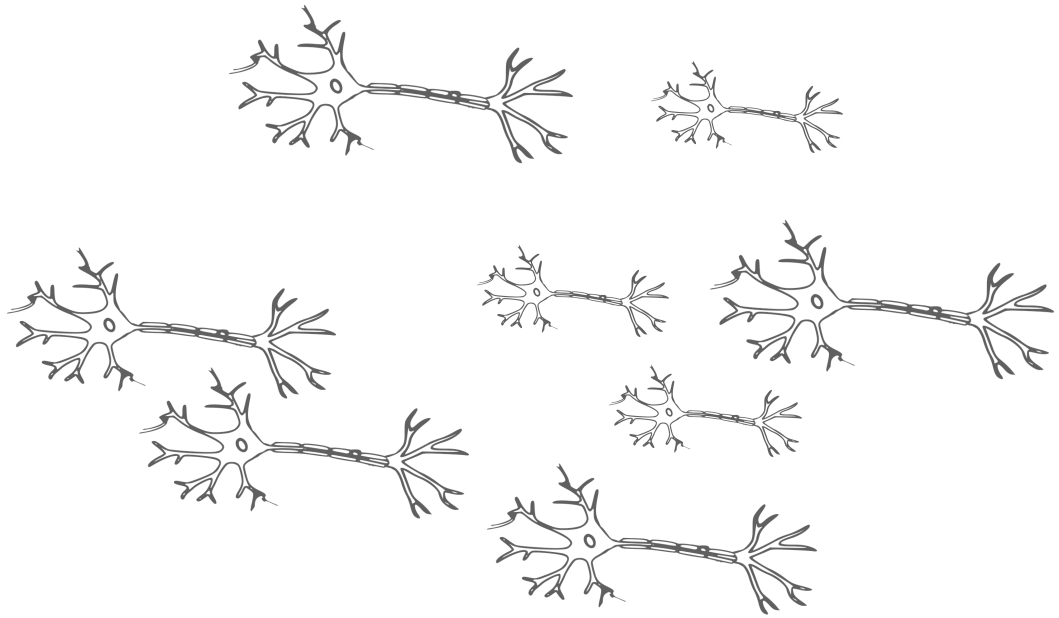
Terminales del axón: Salida de información.

Núcleo: Procesamiento de la información y activación.

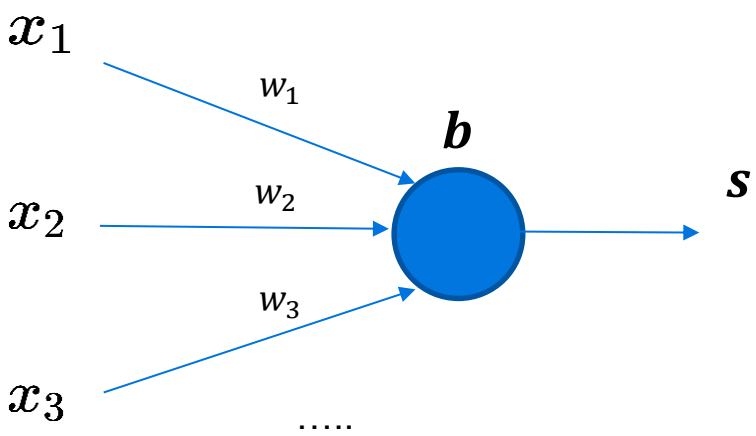
Axón: Transmite la información.

Redes neuronales artificiales





Neurona / perceptrón



El componente fundamental de una red neuronal es una neurona.

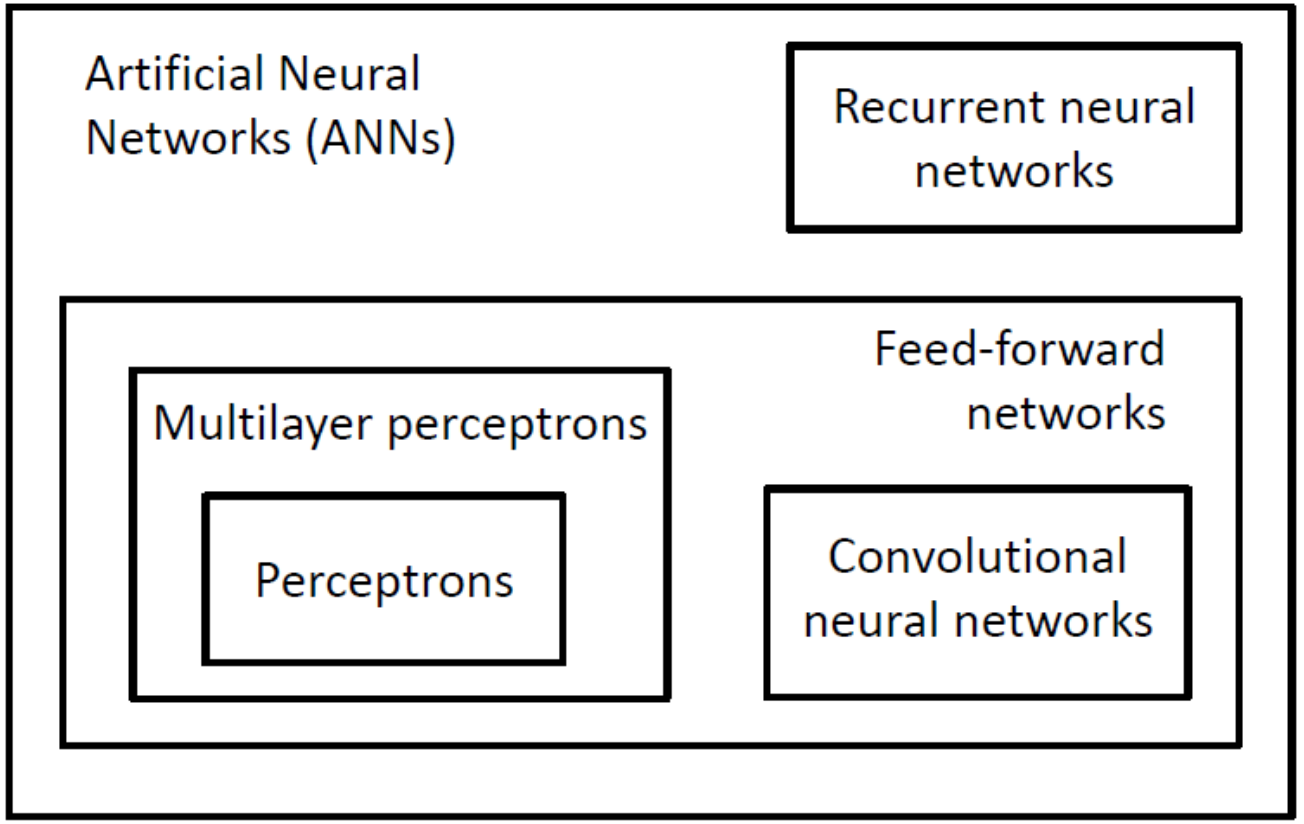
En ella, se multiplica cada valor de entrada por un peso entrenable y se suma una constante.

Luego se define una función que activa la reunona o no dado el producto obtenido previamente.

$$f(x_1 * w_1 + x_2 * w_2 + \ ... + x_m * w_m + \text{b}) = \text{f}(\sum_{i=1}^{m} w_i x_i + b)$$

$f(s)$ = función de activación

Tipos de redes neuronales



Profesor: **Francisco Pérez Galarce**

Redes neuronales artificiales

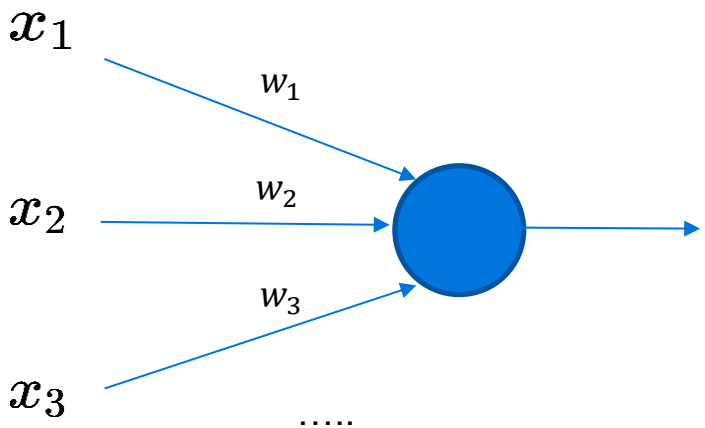
Al valor obtenido se le suma una constante (*bias*).

$$W^T X + b$$

$$W^T \in \mathbb{R}^{1\times 3} \quad X \in \mathbb{R}^{3\times 1}$$

Al resultado se le aplica la función de activación *f*.

$$f(W^T X + b\ )$$



¿Qué pasa si agregamos otra neurona?

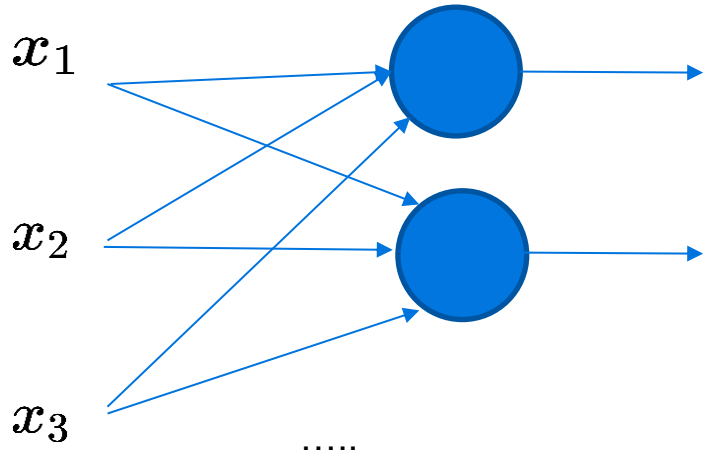
Al valor obtenido se le suma una constante (*bias*).

$$W^T X + b$$

$W^T \in \mathbb{R}^{2\times 3}$ $X \in \mathbb{R}^{3\times 1}$

Al resultado se le aplica la función de activación *f*.

$$f(W^T X + b\ )$$



¿Qué pasa si agregamos n neurona?

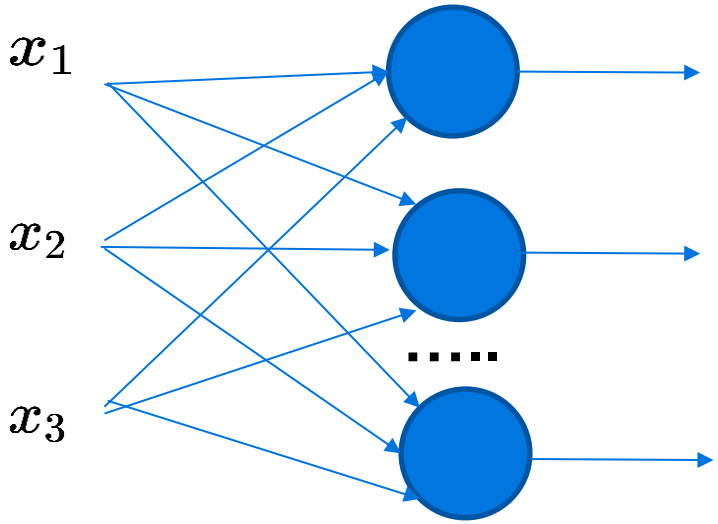
Al valor obtenido se le suma una constante (*bias*).

$$W^T X + b$$

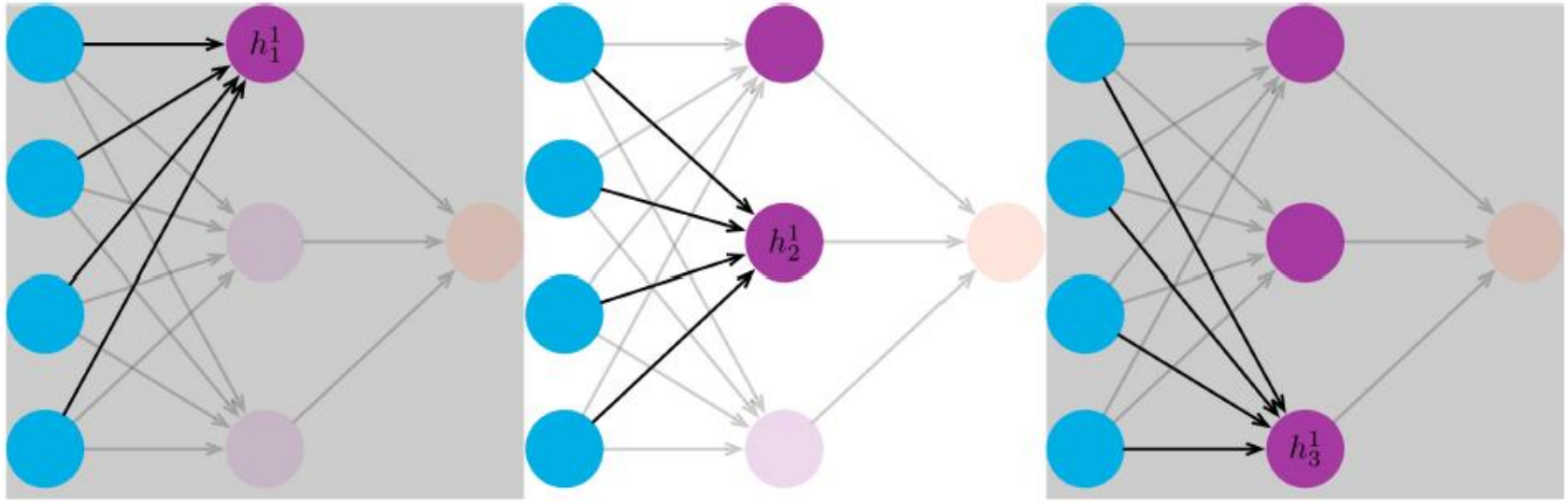
$$W^T \in \mathbb{R}^{n \times 3} \quad X \in \mathbb{R}^{3 \times 1}$$

Al resultado se le aplica la función de activación *f*.

$$f(W^T X + b\ )$$



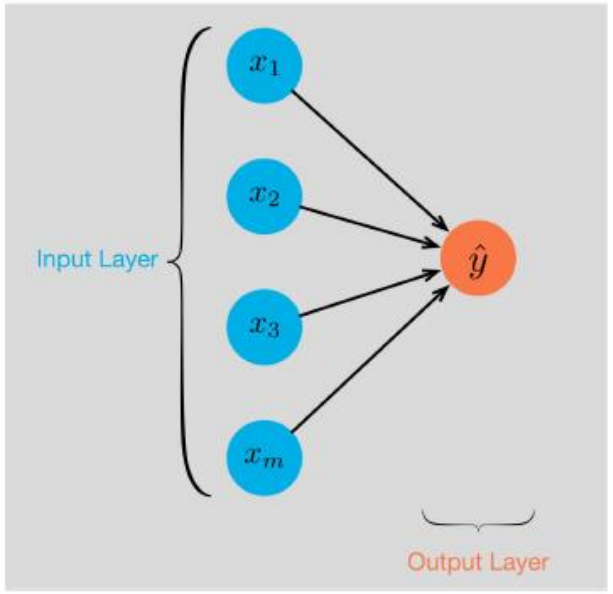
Entendiendo los pesos

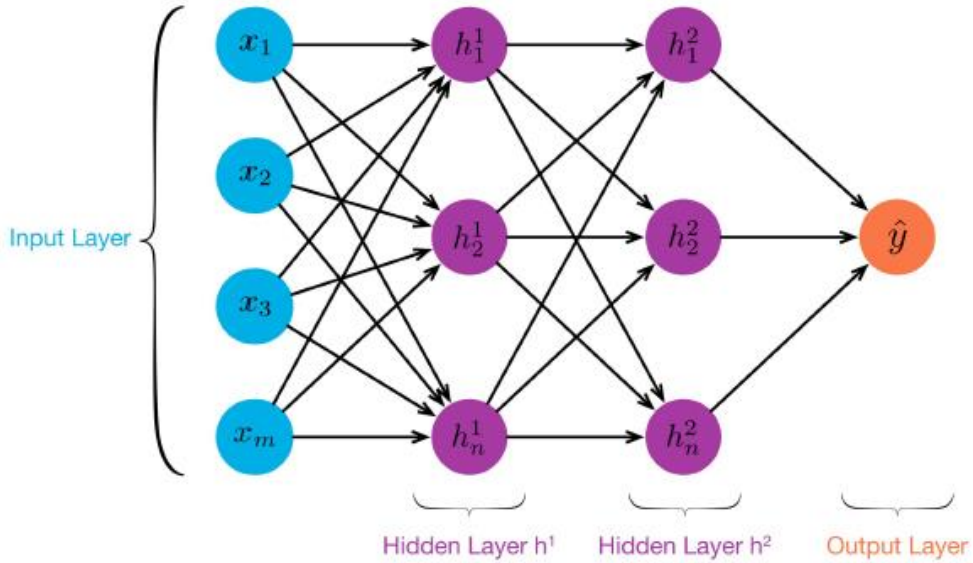


$$\begin{bmatrix} h_1^1 \\ h_2^1 \\ \dots \\ h_n^1 \end{bmatrix} = \begin{bmatrix} w_{11}^1 & w_{12}^1 & \dots & w_{1n}^1 \\ w_{21}^1 & w_{22}^1 & \dots & w_{2n}^1 \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1}^1 & w_{m2}^1 & \dots & w_{mn}^1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Redes neuronales artificiales

Al aumentar el número de neuronal se potencia su expresividad pudiendo ser considerado un excelente estimador de funciones de acuerdo al Teorema de Aproximación Universal (Cybenko, 1989).



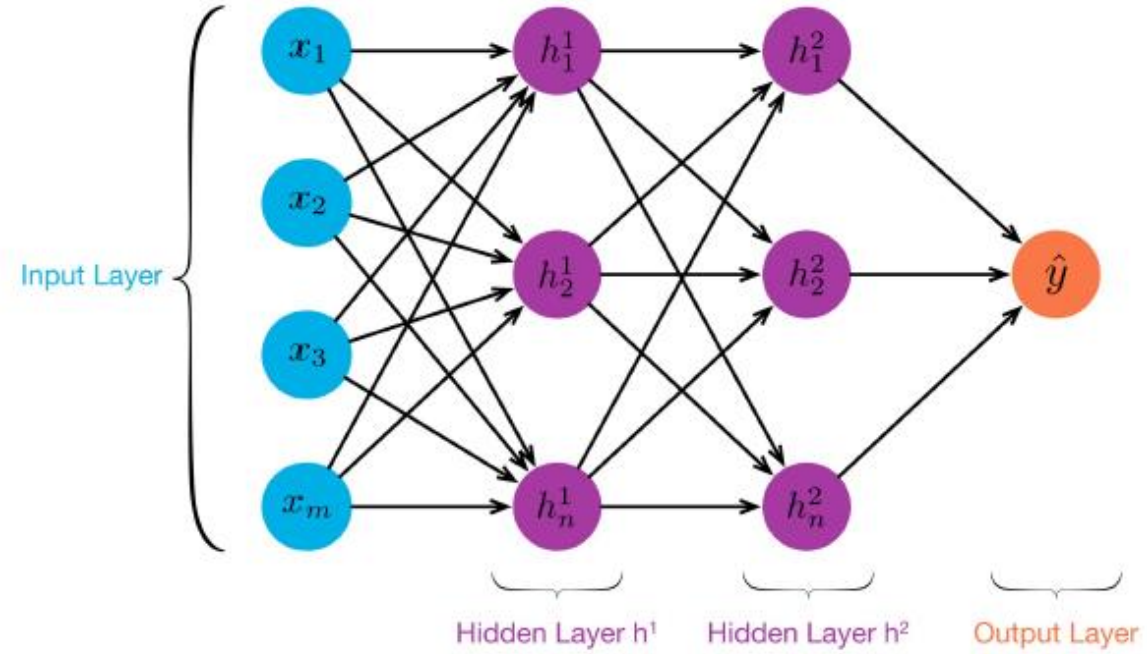


Profesor: **Francisco Pérez Galarce**

Redes neuronales artificiales

Salida del modelo

Usada para clasificación o regresión



Profesor: **Francisco Pérez Galarce**

Profundidad en una red neuronal

Lineal

$$h = \boldsymbol{w}^T \boldsymbol{x}$$

$$\boldsymbol{h} = \boldsymbol{W}^T \boldsymbol{x}$$

$$\boldsymbol{h} = \boldsymbol{W}_2^T \boldsymbol{W}_1^T \boldsymbol{x}$$

$$\boldsymbol{h} = \boldsymbol{W}_3^T \boldsymbol{W}_2^T \boldsymbol{W}_1^T \boldsymbol{x}$$

$$\boldsymbol{h} = \boldsymbol{W}_N^T \ldots \boldsymbol{W}_2^T \boldsymbol{W}_1^T \boldsymbol{x}$$

No lineal

$$h = f(\mathbf{w}^T \mathbf{x})$$

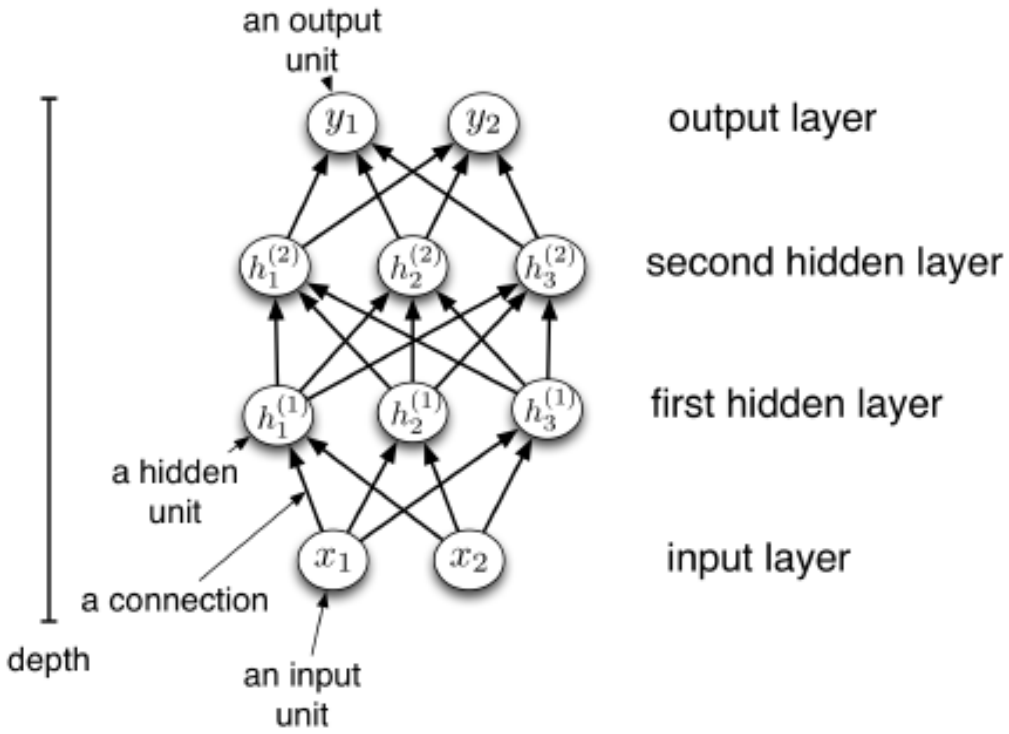
$$\mathbf{h} = f(\mathbf{W}^T \mathbf{x})$$

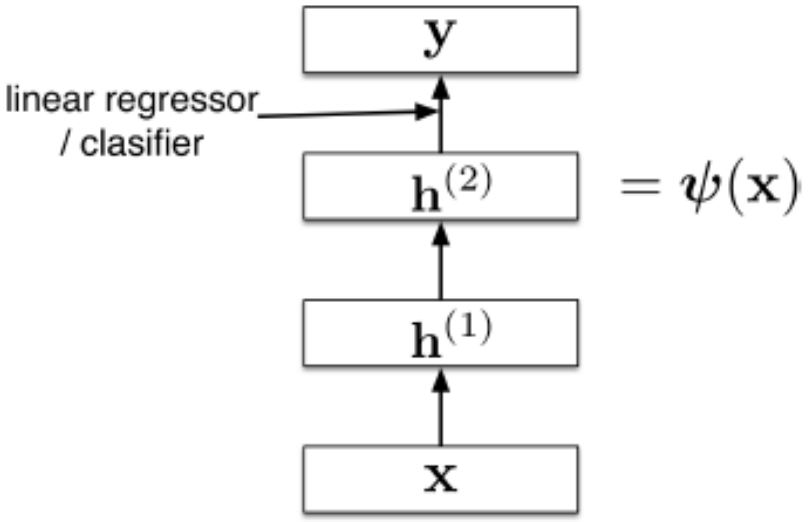
$$\mathbf{h} = f(\mathbf{W}_2^T f(\mathbf{W}_1^T \mathbf{x}))$$

$$\mathbf{h} = f(\mathbf{W}_3^T f(\mathbf{W}_2^T f(\mathbf{W}_1^T \mathbf{x})))$$

$$\mathbf{h} = f(\mathbf{W}_N^T .. f(\mathbf{W}_2^T f(\mathbf{W}_1^T \mathbf{x})))$$

Componentes de la red neuronal

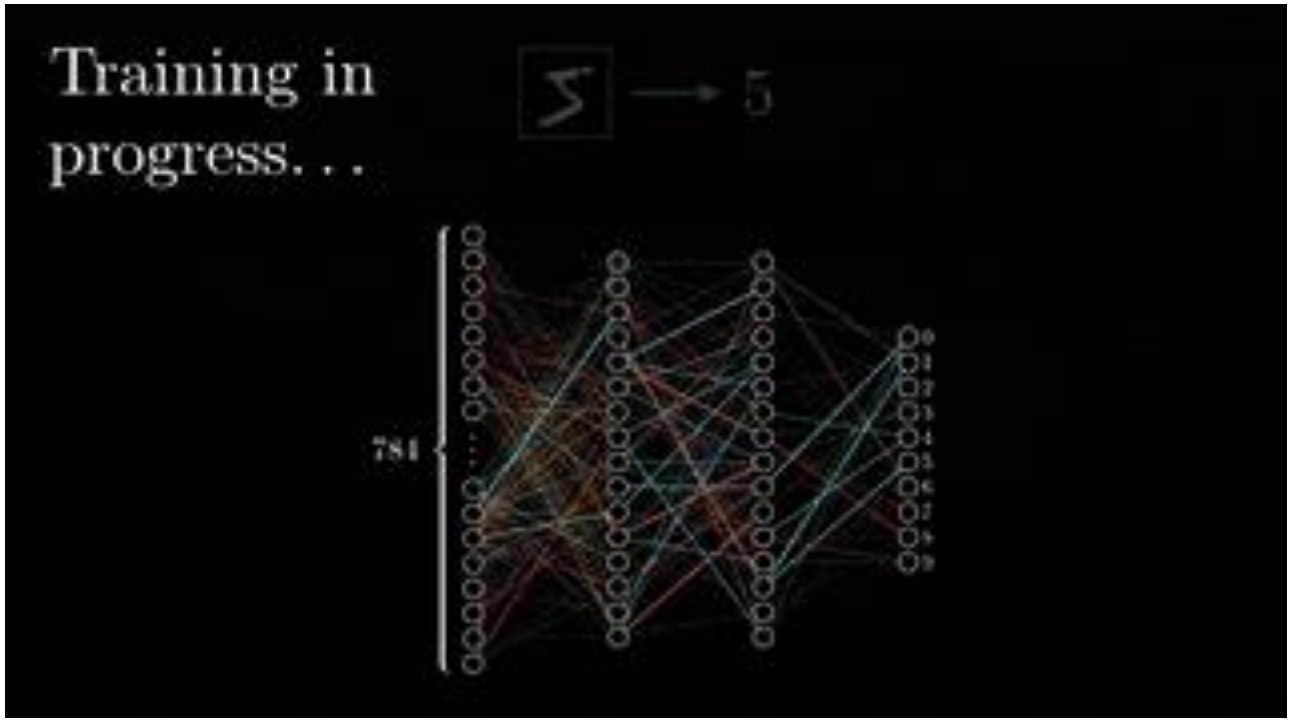




Conceptos para la definición y entrenamiento de una red neuronal

Profesor: **Francisco Pérez Galarce**

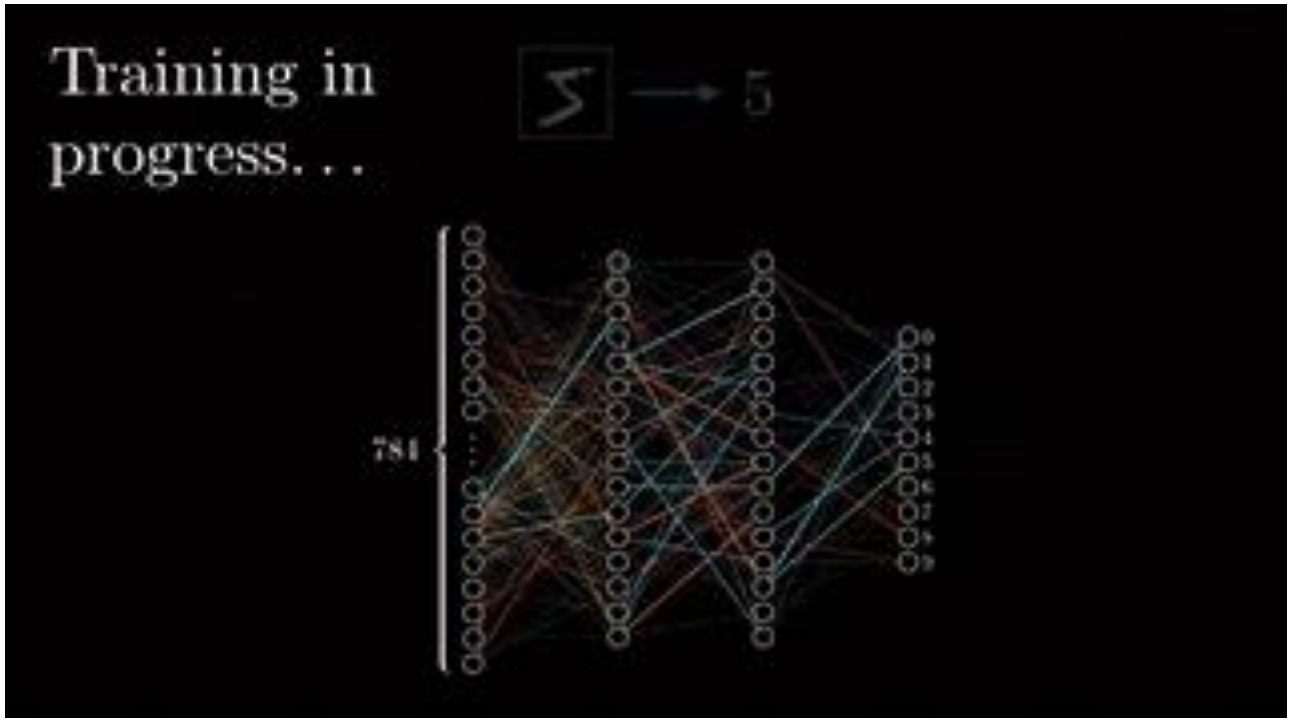
Algunos componentes de una red neuronal



- Funciones de activación
- Función de pérdida
- Algoritmos de optimización
- Batch y épocas
- Tipo de capa
- Backpropagation

Profesor: **Francisco Pérez Galarce**

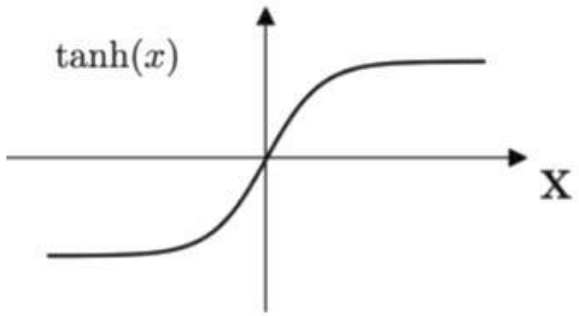
Algunos componentes de una red neuronal



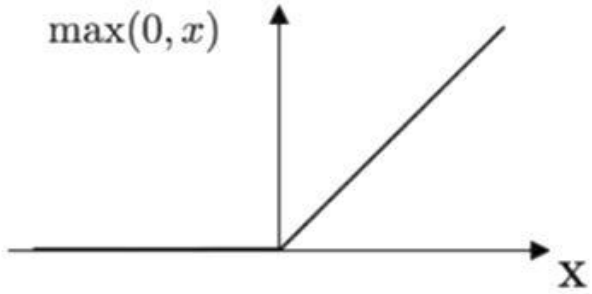
- Funciones de activación
- Función de pérdida
- Algoritmos de optimización
- Batch y épocas
- Tipo de capa
- Backpropagation

Funciones de activación

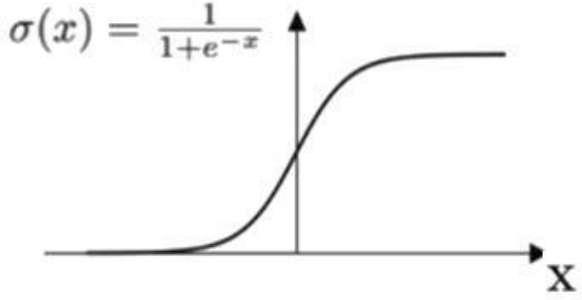
Tanh



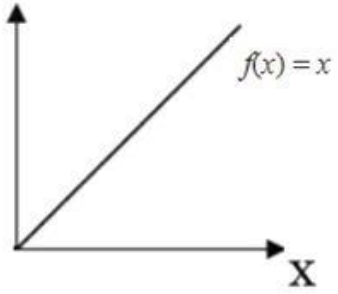
ReLU



Sigmoid



Linear



Profesor: **Francisco Pérez Galarce**

Última capa

Para clasificación binaria, podemos usar la función sigmoide:

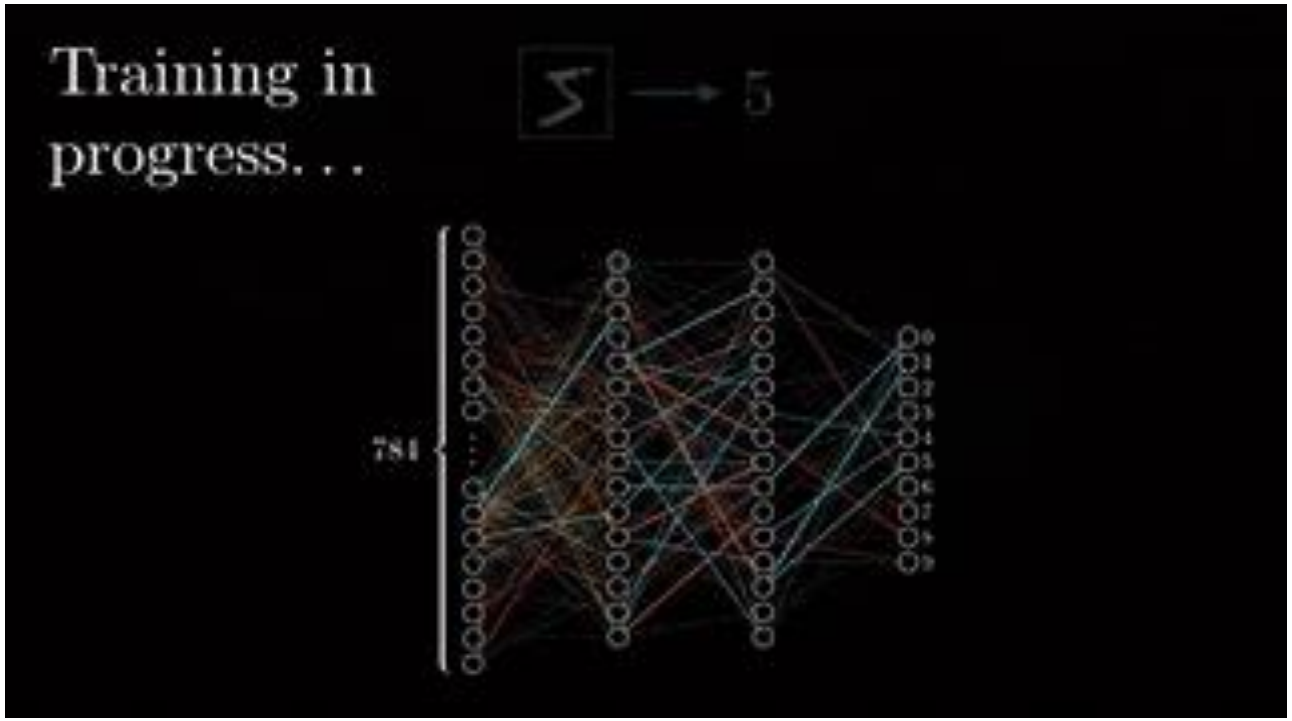
$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Redes neuronales artificiales: Última capa

Para clasificación multiclase con K posibles categorías, Podemos usar la función *softmax*.

$$softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

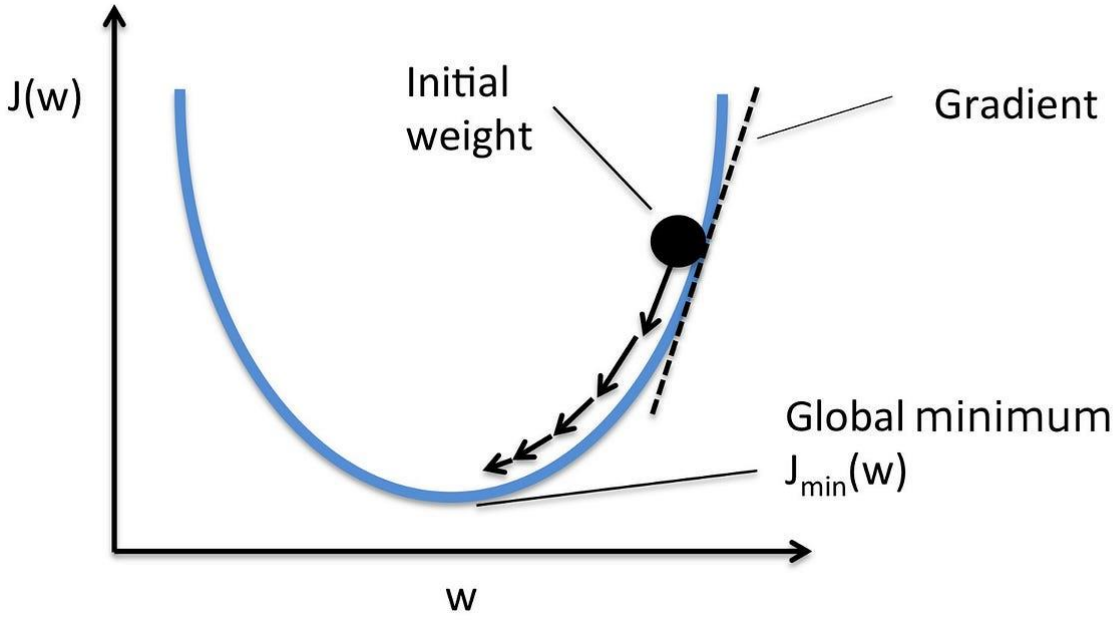
Algunos componentes de una red neuronal

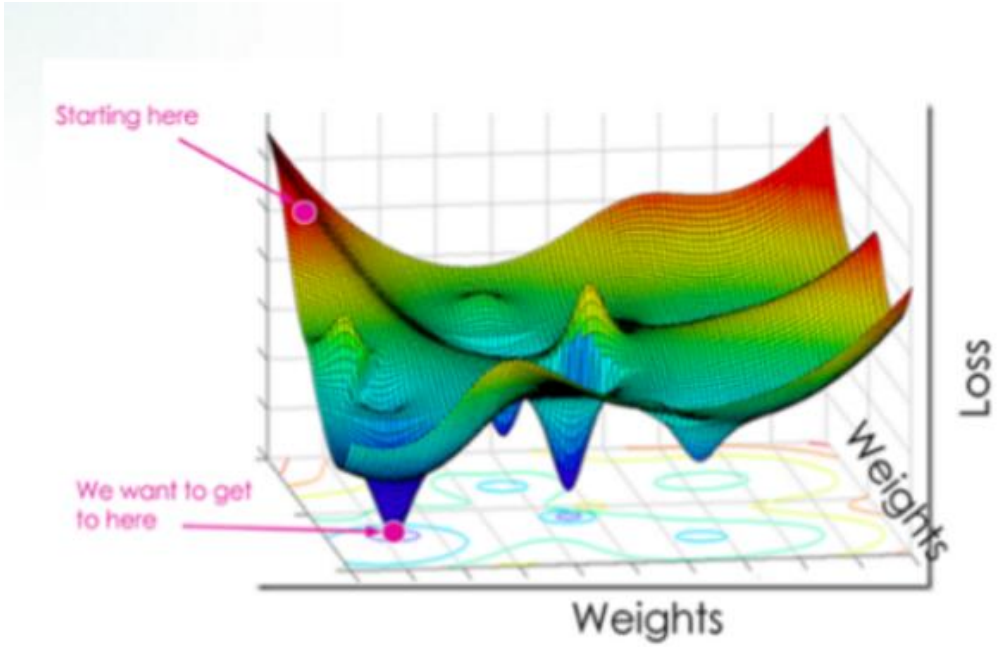


- Funciones de activación
- Función de pérdida
- Algoritmos de optimización
- Batch y épocas
- Tipo de capa
- Backpropagation

Profesor: **Francisco Pérez Galarce**

Redes neuronales artificiales: Función de pérdida





Profesor: **Francisco Pérez Galarce**

Redes neuronales artificiales: Función de pérdida

Para clasificación se utiliza la entropía cruzada o "binary cross-entropy", que se define como

$$Loss = \sum_{i=1}^{N} y_i \log \hat{y}_i$$

Donde y_i es el valor a observado e $\hat{y}_i$ es el valor de la predicción.

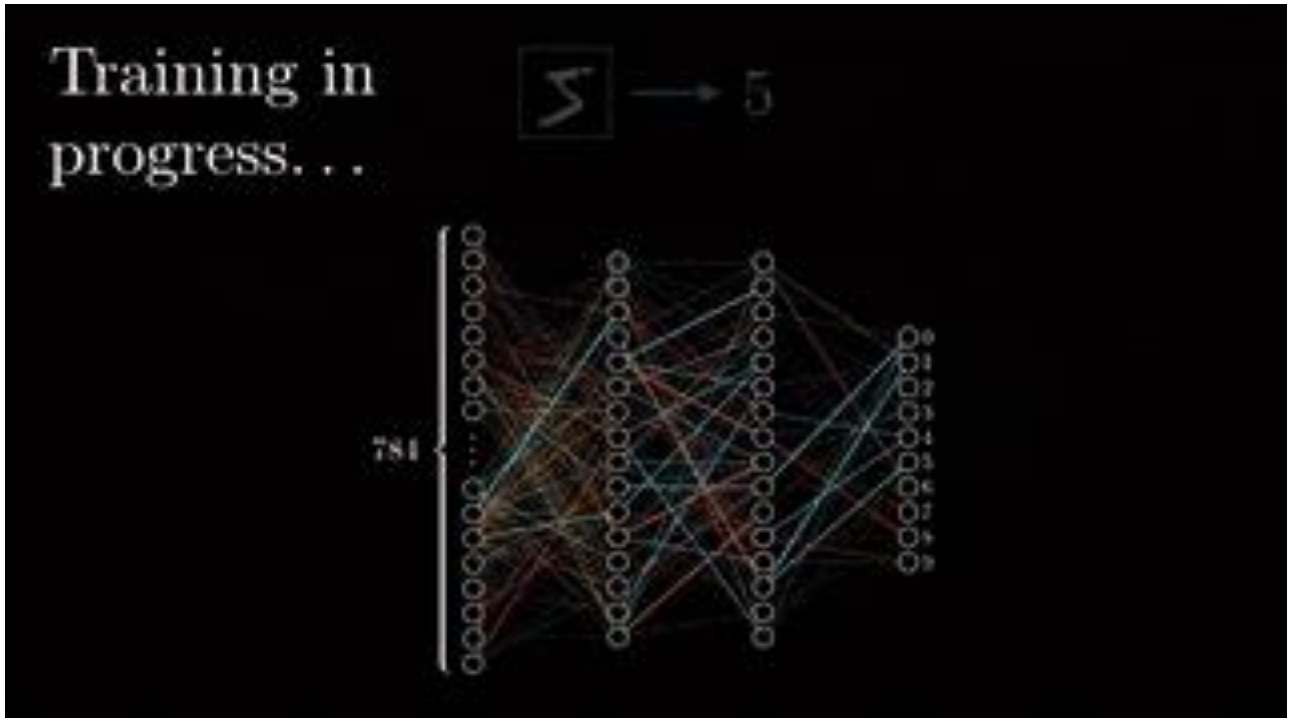
Redes neuronales artificiales: Función de pérdida

Para regresión se utiliza el "mean squared error" o MSE

$$Loss = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

Donde y_i es el valor observado e $\hat{y}_i$ es el valor de la predicción.

Algunos componentes de una red neuronal

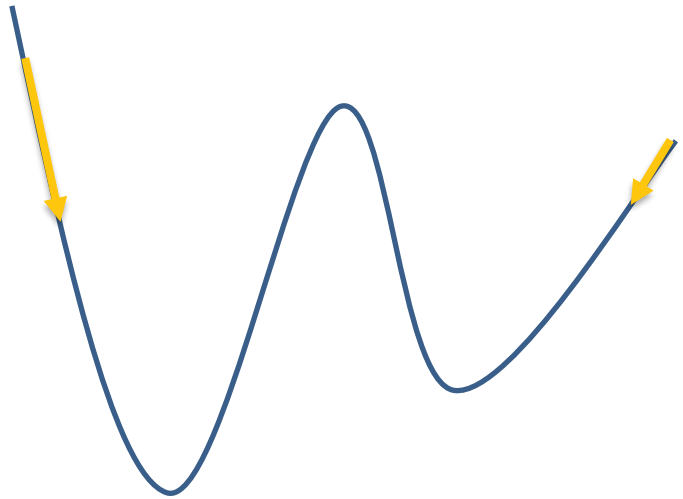


- Funciones de activación
- Función de pérdida
- Algoritmos de optimización
- Batch y épocas
- Tipo de capa
- Backpropagation

Profesor: **Francisco Pérez Galarce**

Redes neuronales artificiales: Entrenamiento

El **learning rate**, nos permite elegir cuánto avanzar en cada iteración del proceso de optimización.

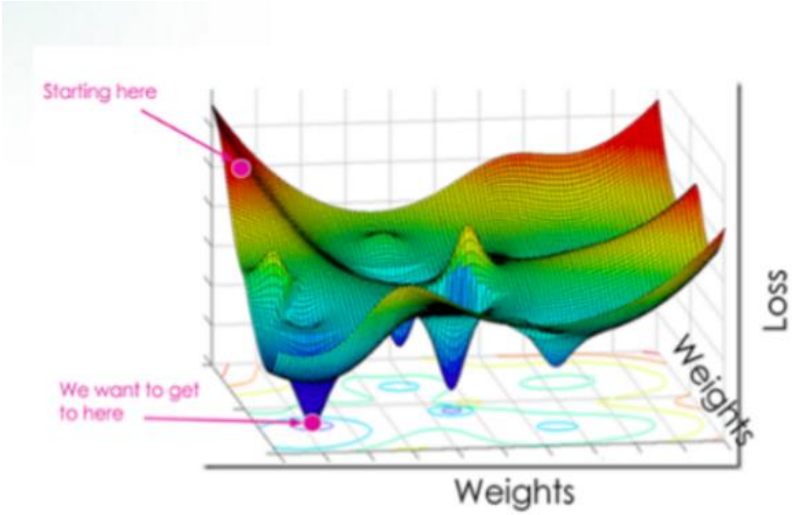


Redes neuronales artificiales: Entrenamiento

El descenso de gradiente permite optimizar los parámetros, pero no es el único algoritmo.

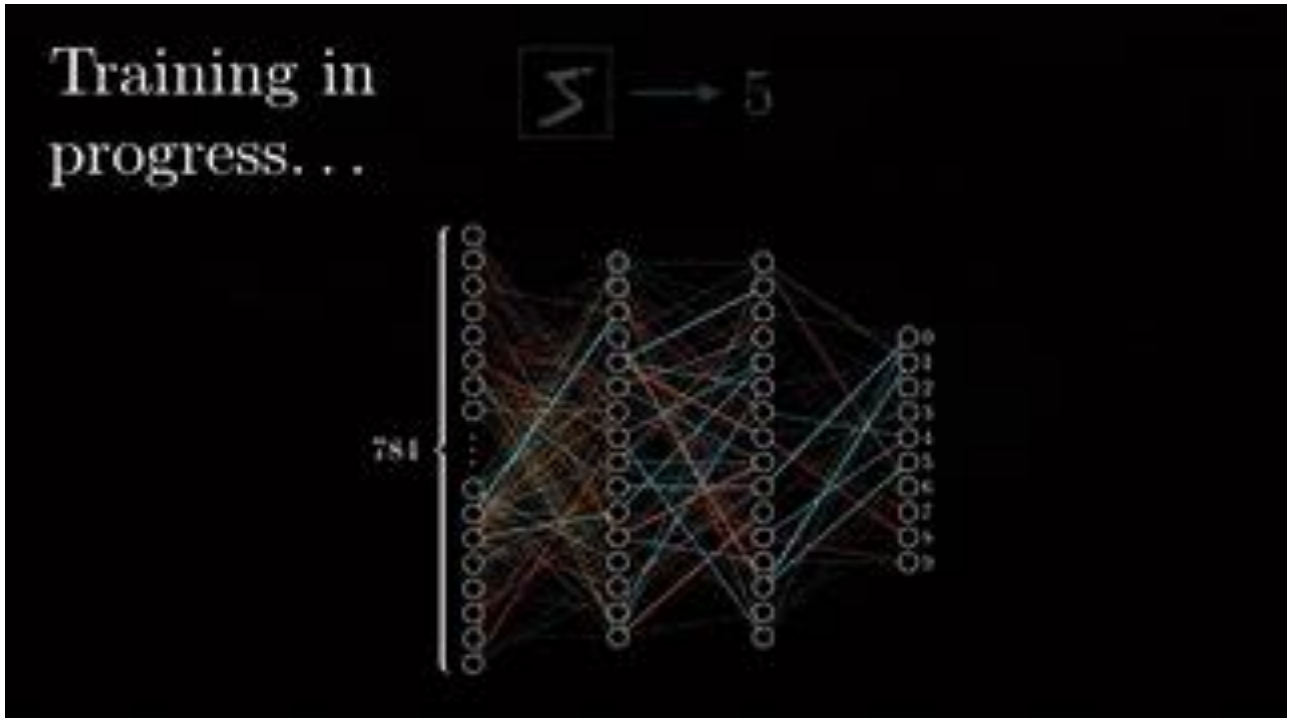
Algunos son:

- Adam
- Adagrad
- RMSprop
- AdamW



Ver listado de optimizadores en pytorch: https://pytorch.org/docs/stable/optim.html

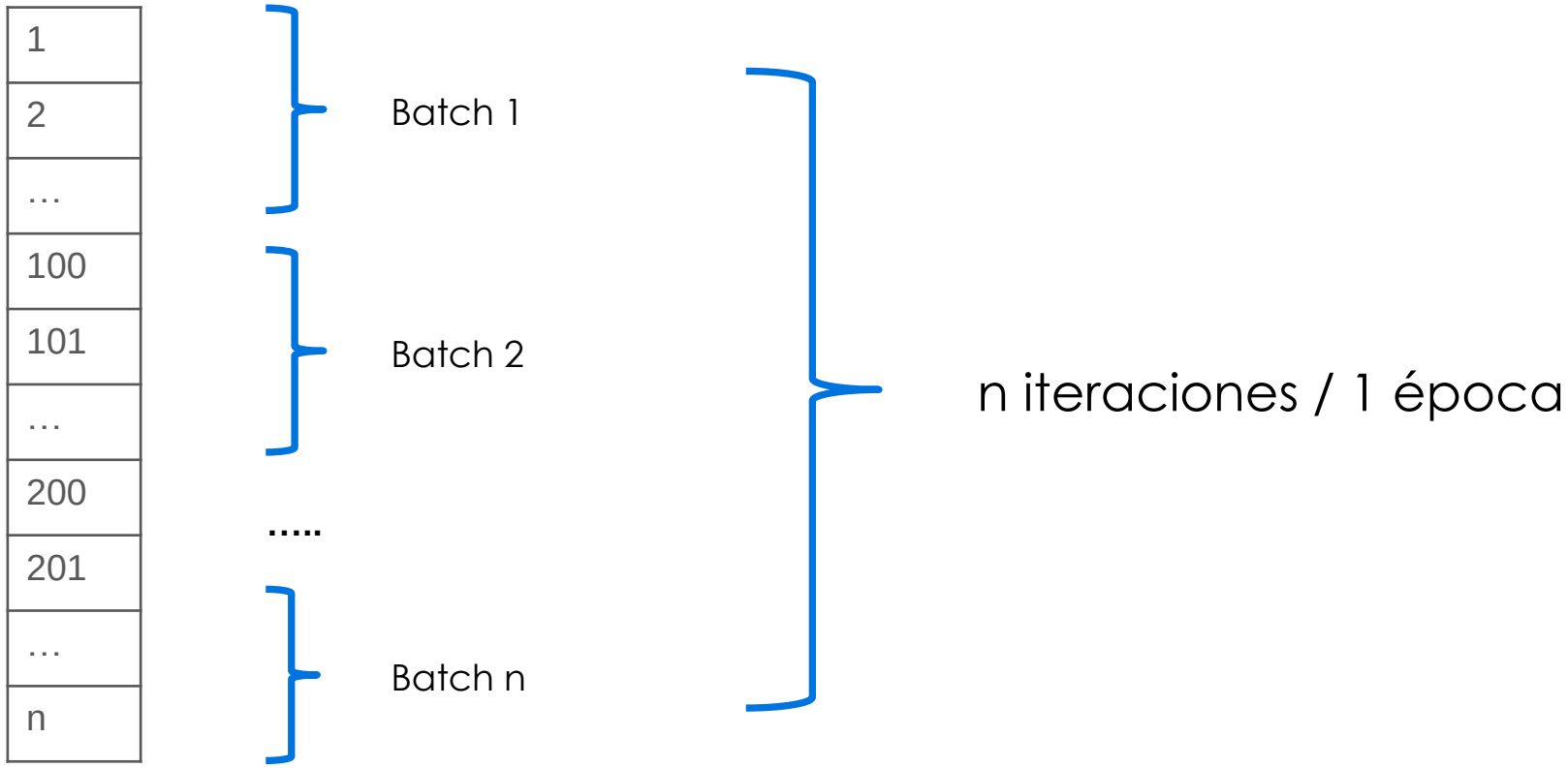
Algunos componentes de una red neuronal



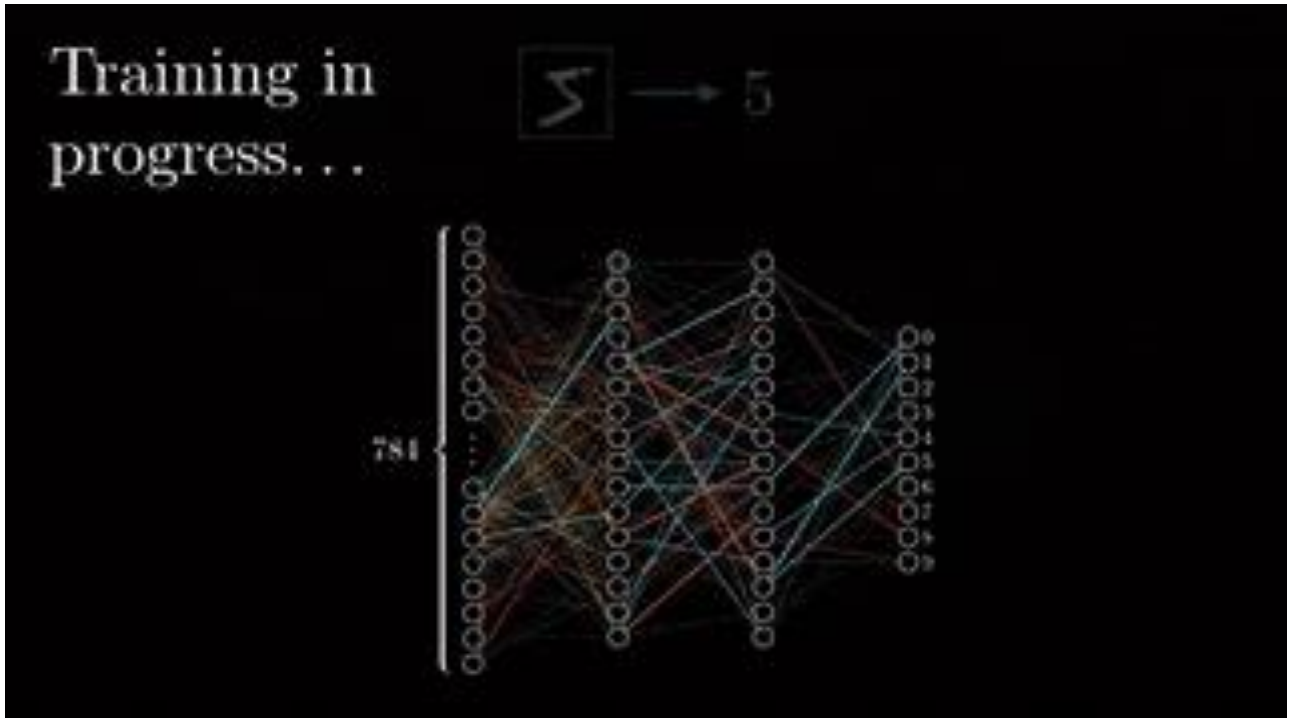
- Funciones de activación
- Función de pérdida
- Algoritmos de optimización
- Batch y épocas
- Tipo de capa
- Backpropagation

Profesor: **Francisco Pérez Galarce**

Redes neuronales artificiales: Entrenamiento



Algunos componentes de una red neuronal

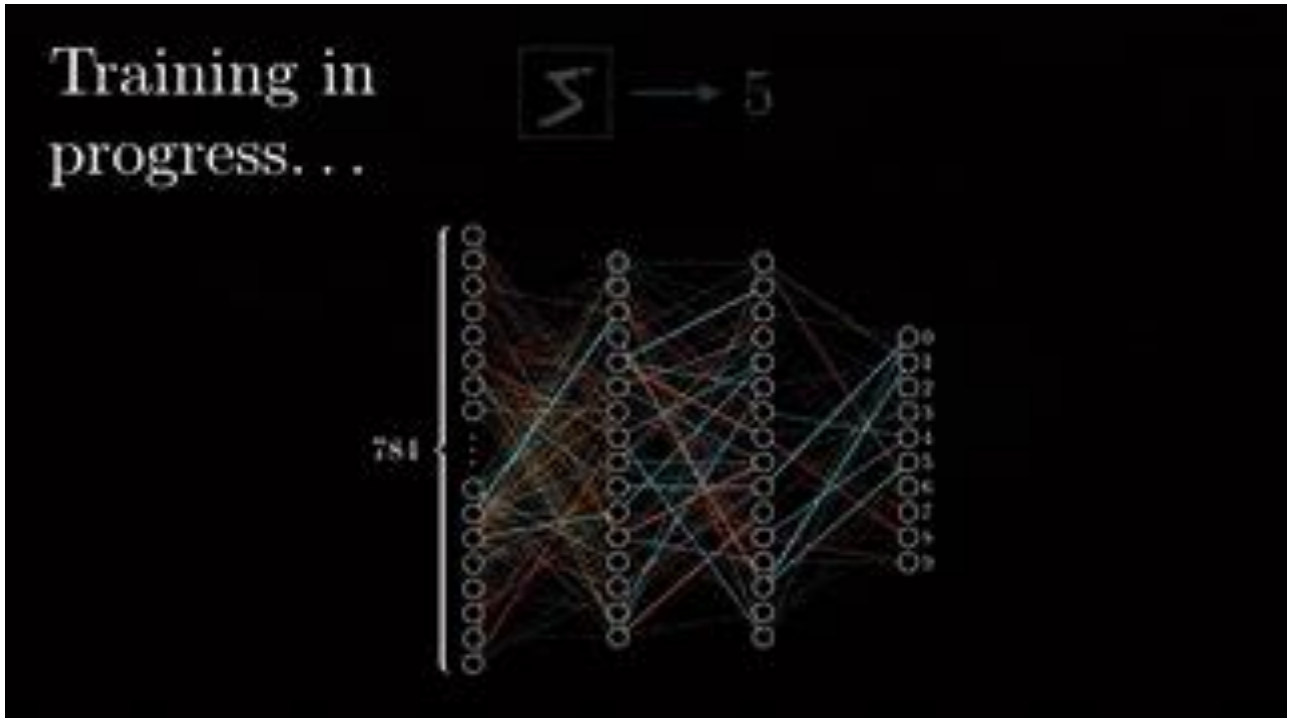


- Funciones de activación
- Función de pérdida
- Algoritmos de optimización
- Batch y épocas
- Tipo de capa
- Backpropagation

Redes neuronales artificiales

- Convolution layer
- Pooling layer
- Rectified Linear Unit
- Fully connected
- SoftMax

Algunos componentes de una red neuronal



- Funciones de activación
- Función de pérdida
- Algoritmos de optimización
- Batch y épocas
- Tipo de capa
- Backpropagation

Profesor: **Francisco Pérez Galarce**

Redes neuronales artificiales: Entrenamiento

Backpropagation nos permite encontrar los pesos de forma eficiente.

$$W^{t+1} = W^{t} - \eta \nabla_{W} Loss$$

- W^{t+1}: Parámetro actualizado
- W^{t}: Valor actual
- η: Learning rate
- $\nabla_{W} Loss$: Derivada de la Loss

ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL

www.educacionprofesional.ing.uc.cl