



Universitat d'Alacant
Universidad de Alicante



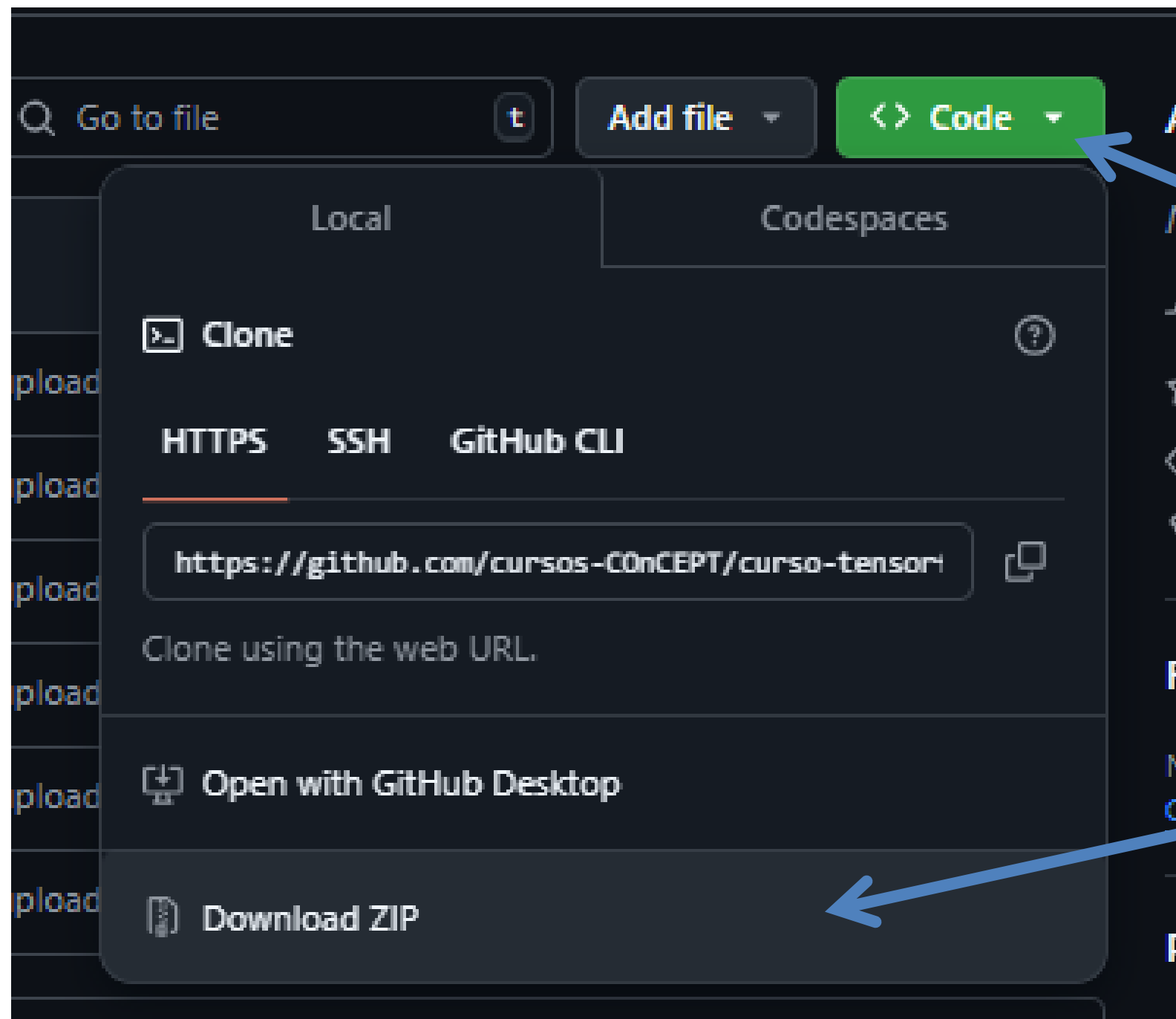
Mes Cultural y de
Divulgación EPS 2025

Introducción al *machine learning* con TensorFlow



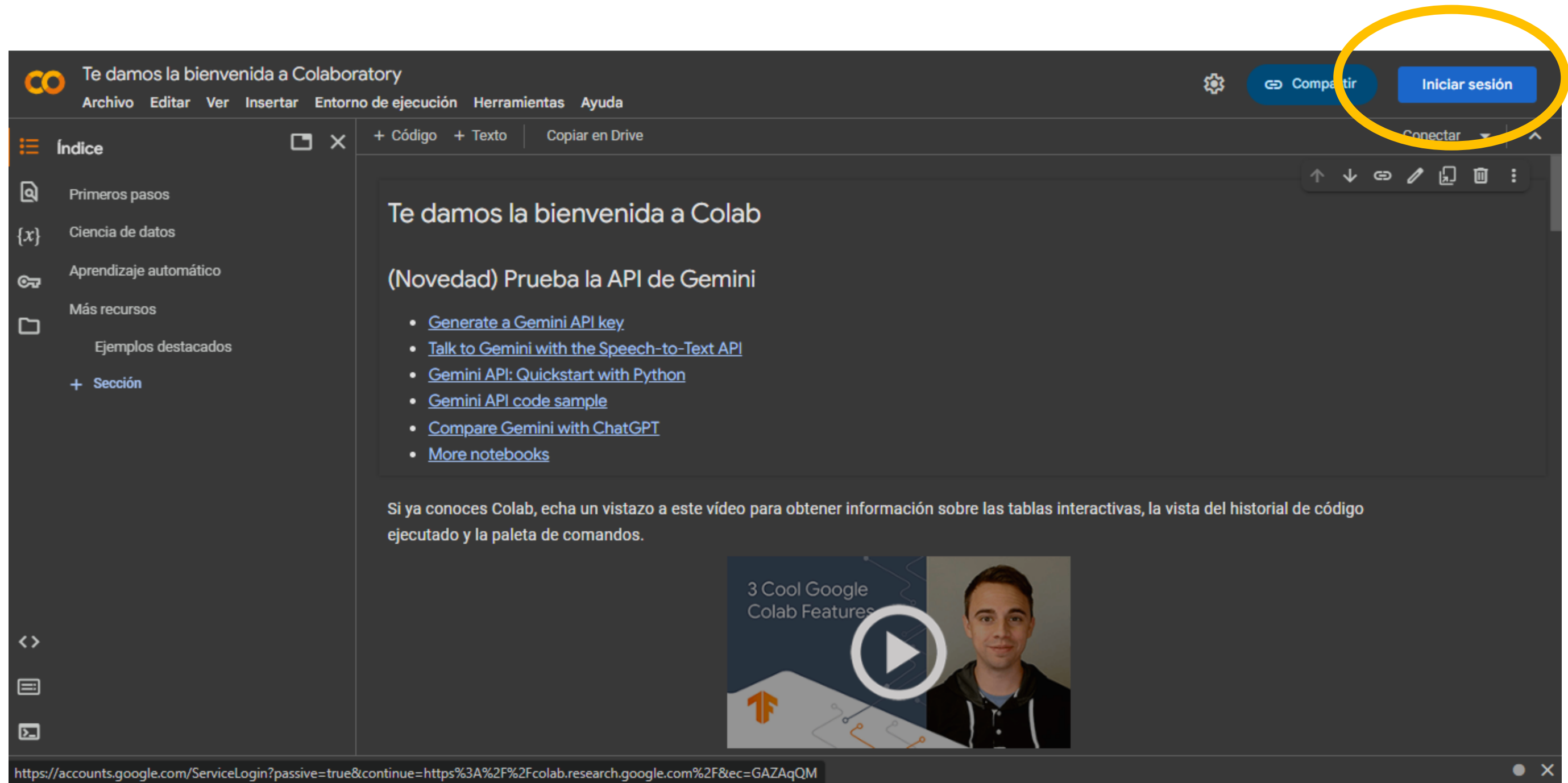
Materiales del taller:

<https://github.com/cursos-COnCEPT/curso-tensorflow>



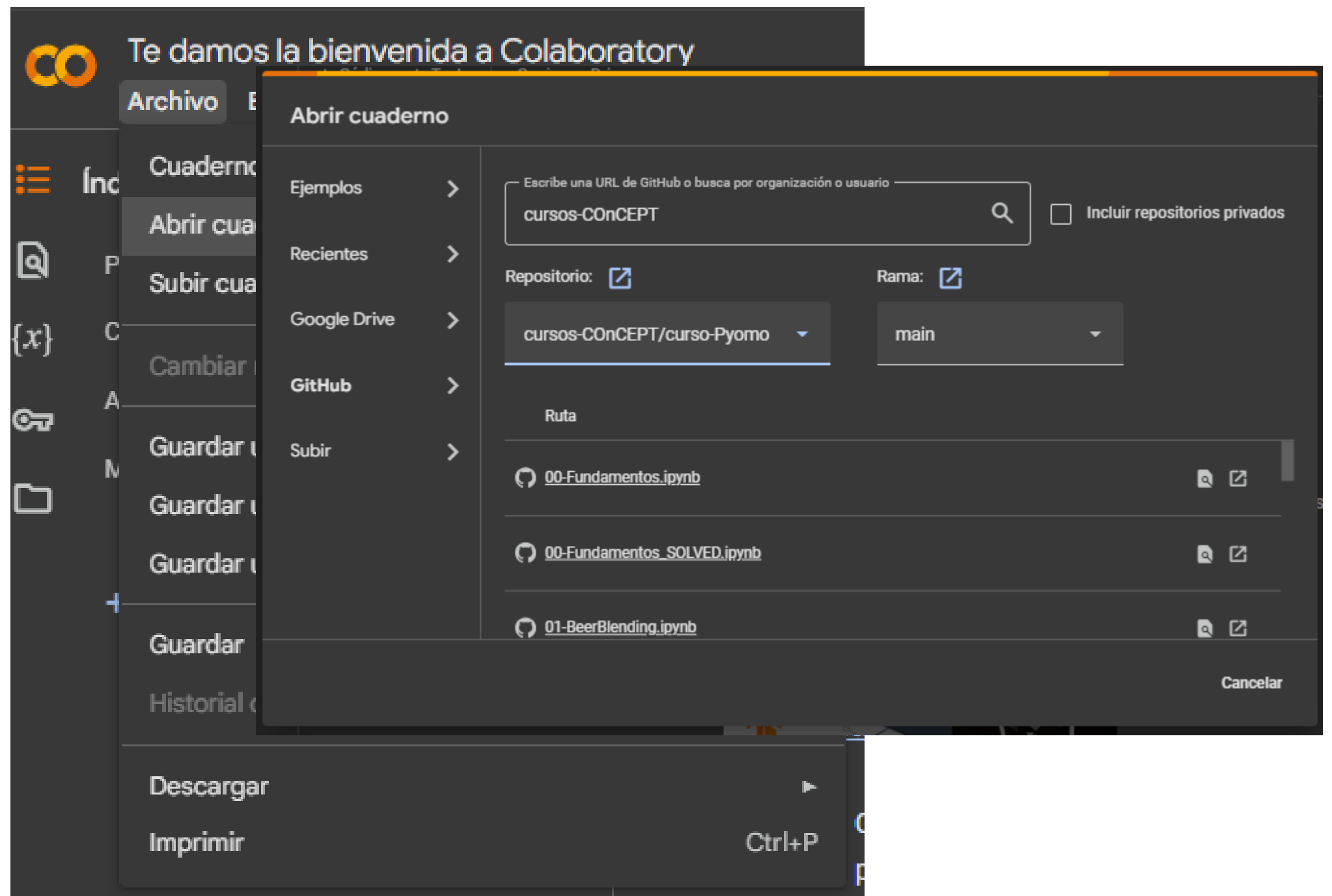
1- Iniciar sesión en GoogleColab

<https://colab.research.google.com/>



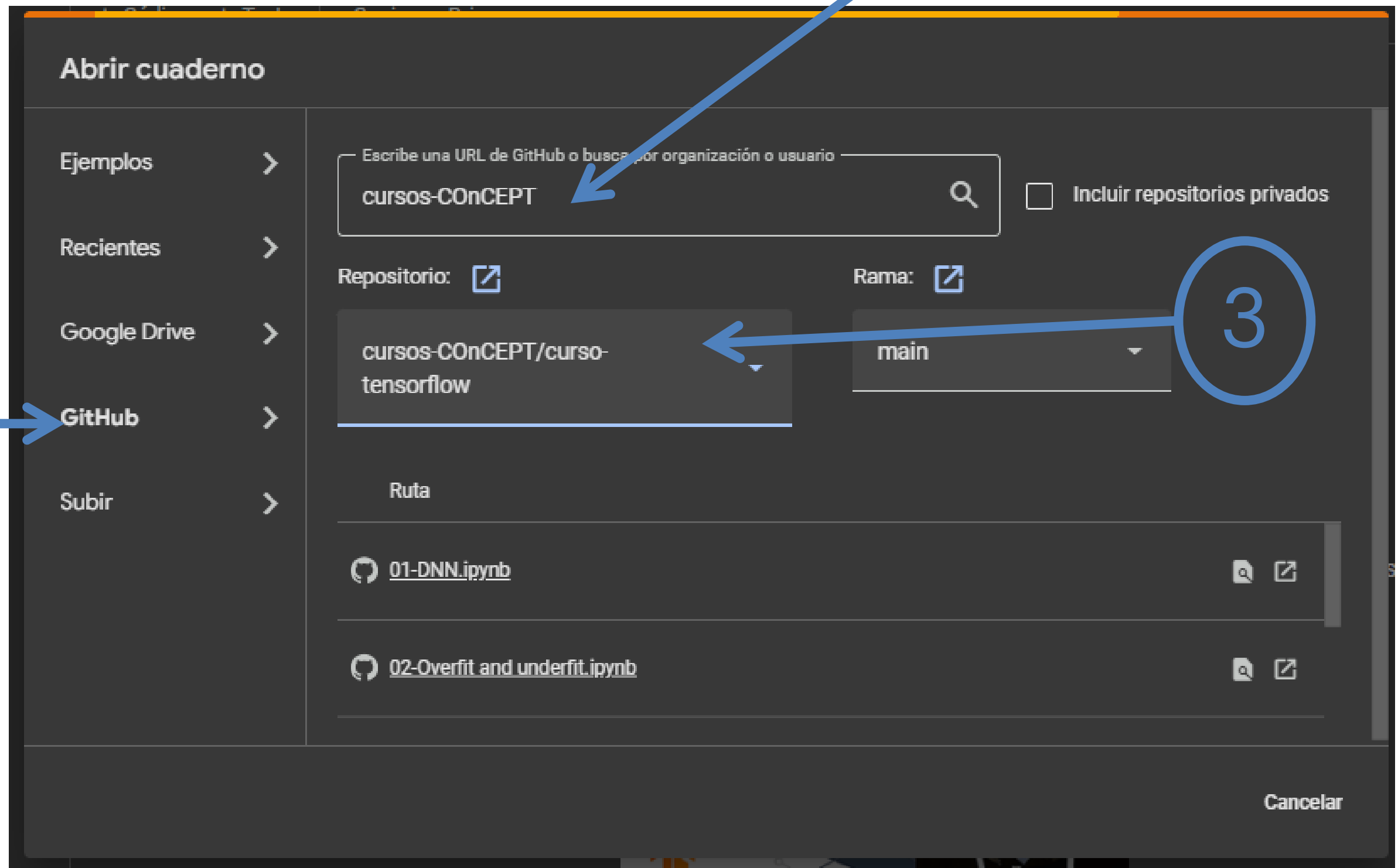
2 - Abrir archivos

Archivo >>> Abrir cuaderno



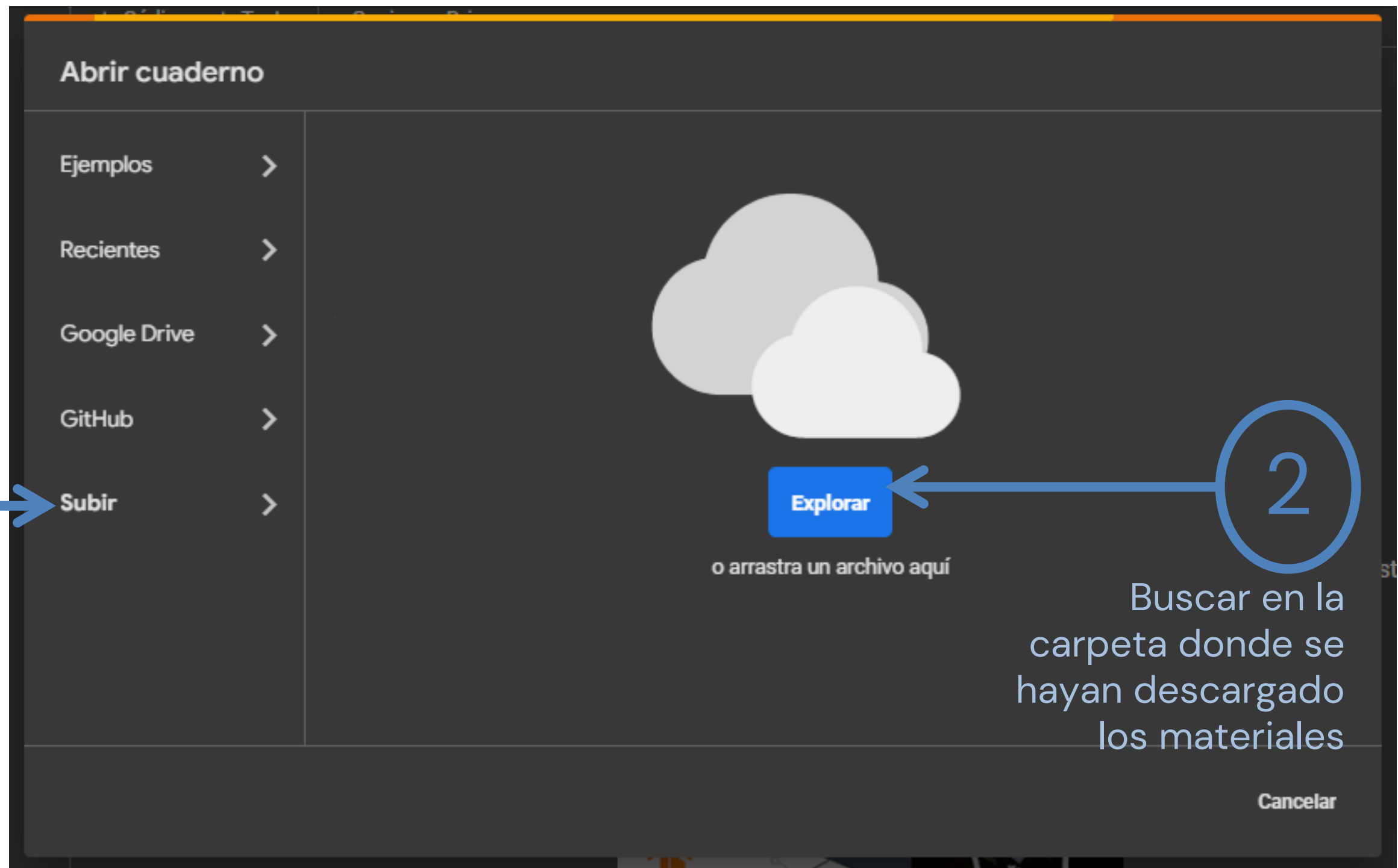
2 - Abrir archivos

Desde GitHub



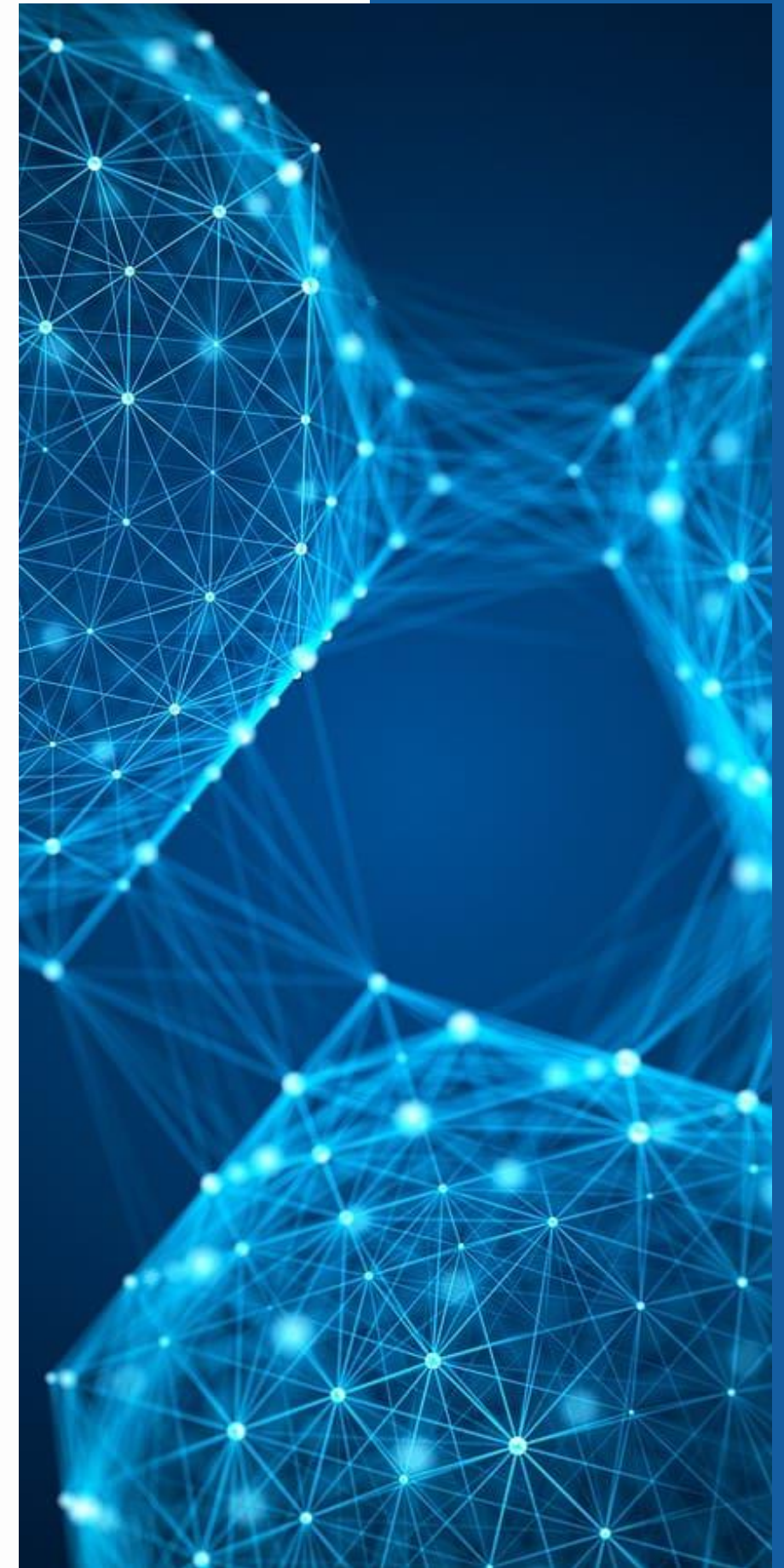
2 - Abrir archivos


Desde el ordenador (si GitHub no funciona)



Índice

- ▶ Conceptos básicos
- ▶ Tipos de aprendizaje
- ▶ Redes neuronales
- ▶ ¿Qué es TensorFlow?
- ▶ Pasos para entrenar un modelo
- ▶ Sobreajuste y subajuste
- ▶ Ajuste de hiperparámetros



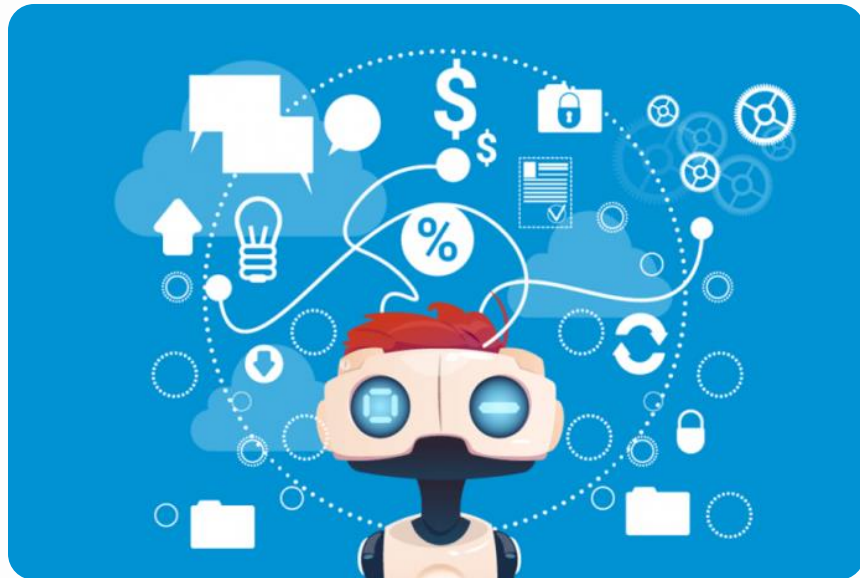
A black and white photograph of John McCarthy, a man with a beard and glasses, wearing a suit and tie. He is sitting at a desk, playing a game of chess. In the background, there is a large computer system with many reels of magnetic tape. The text "IBM 7090" is visible on a label. The image is partially covered by a blue semi-transparent overlay on the left side, which contains the quote and the name "John McCarthy".

“La ciencia y la
ingeniería
de hacer máquinas
inteligentes,
especialmente
programas de
computadora
inteligentes”

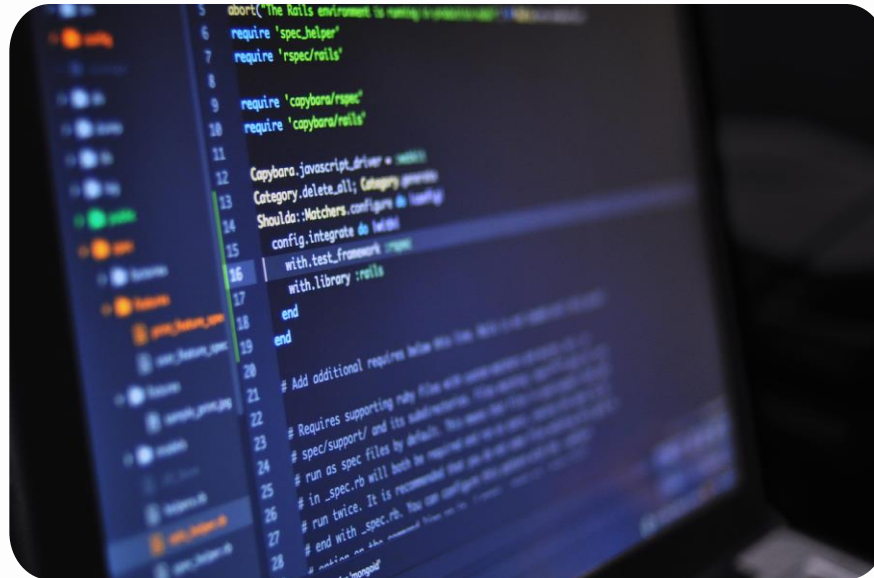
John McCarthy

1956

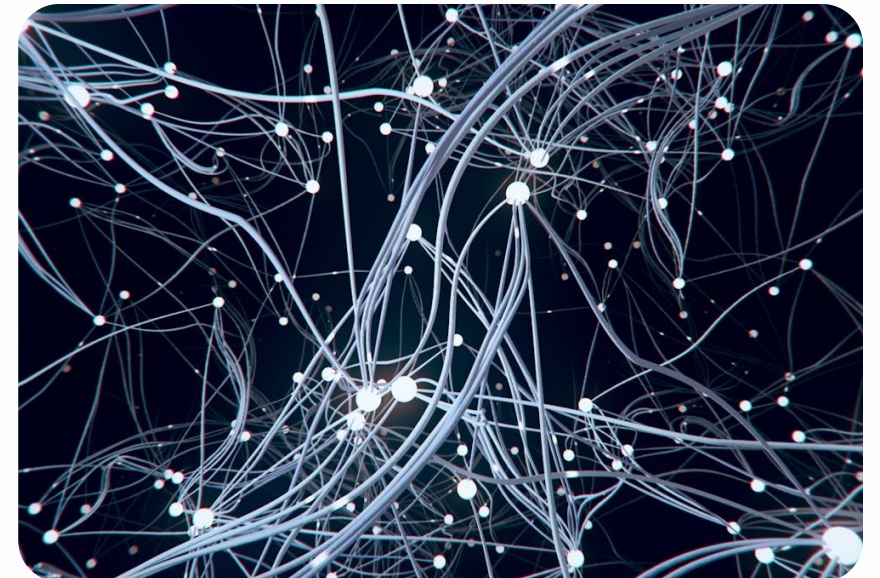
¿Es lo mismo?



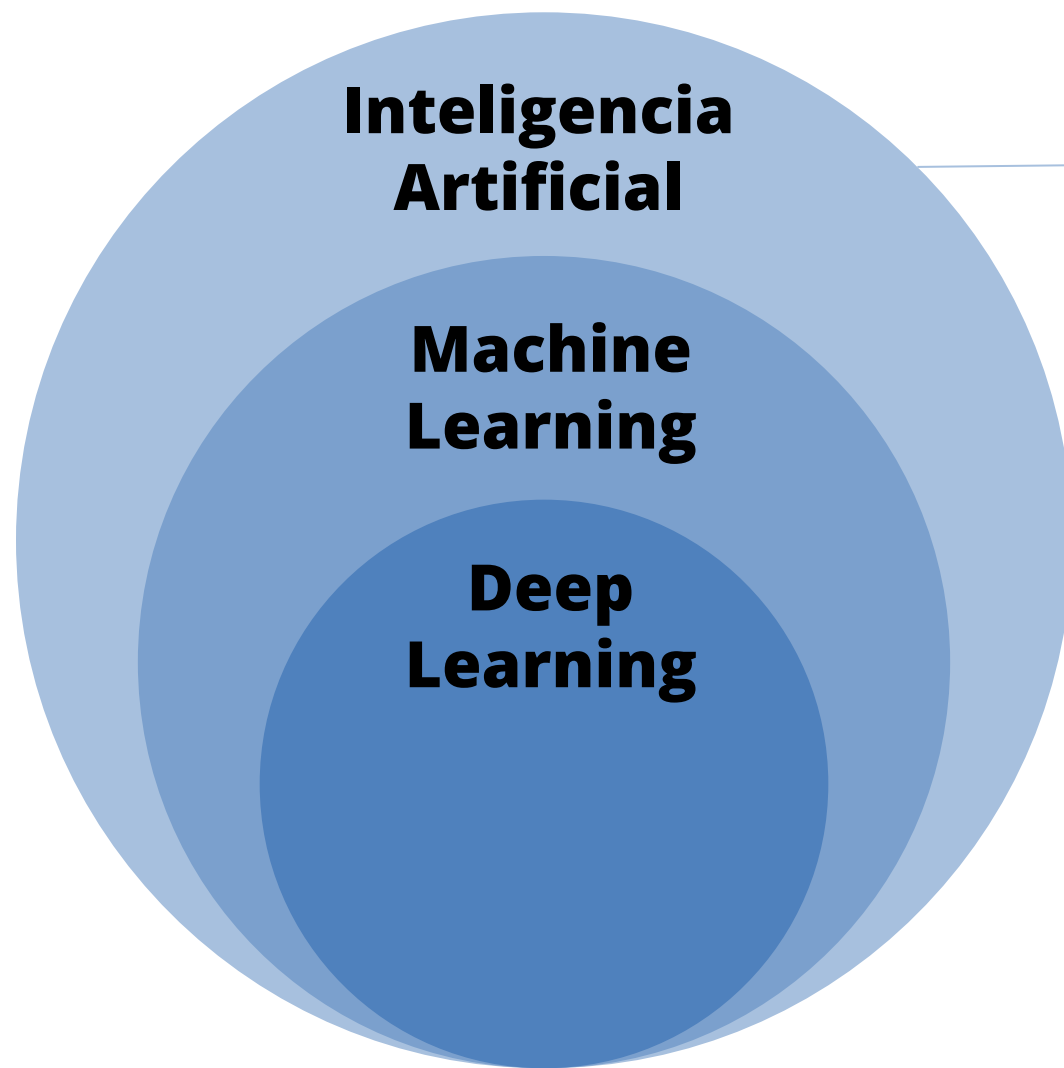
Inteligencia Artificial



Machine Learning



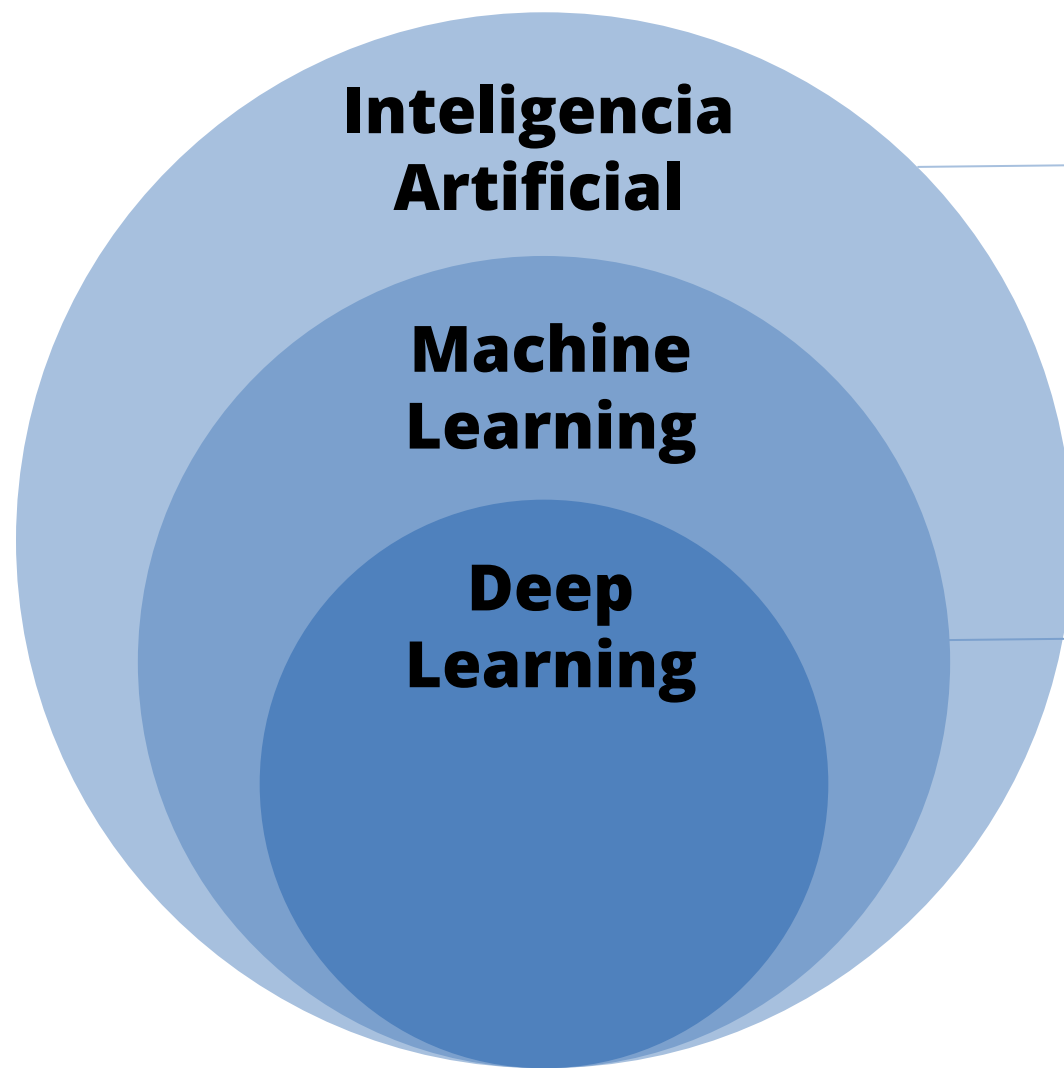
Deep Learning



Rama de la informática orientada a conseguir que las máquinas **simulen comportamientos humanos y tomen decisiones** de forma autónoma

- Chatbots
- Filtros email
- Coches autónomos
- Reconocimiento facial





Rama de la informática orientada a conseguir que las máquinas **simulen comportamientos humanos** y **tomen decisiones** de forma autónoma

Subcampo de la IA basado en la **estadística** para construir **modelos predictivos** que aprenden de **grandes cantidades de datos**.



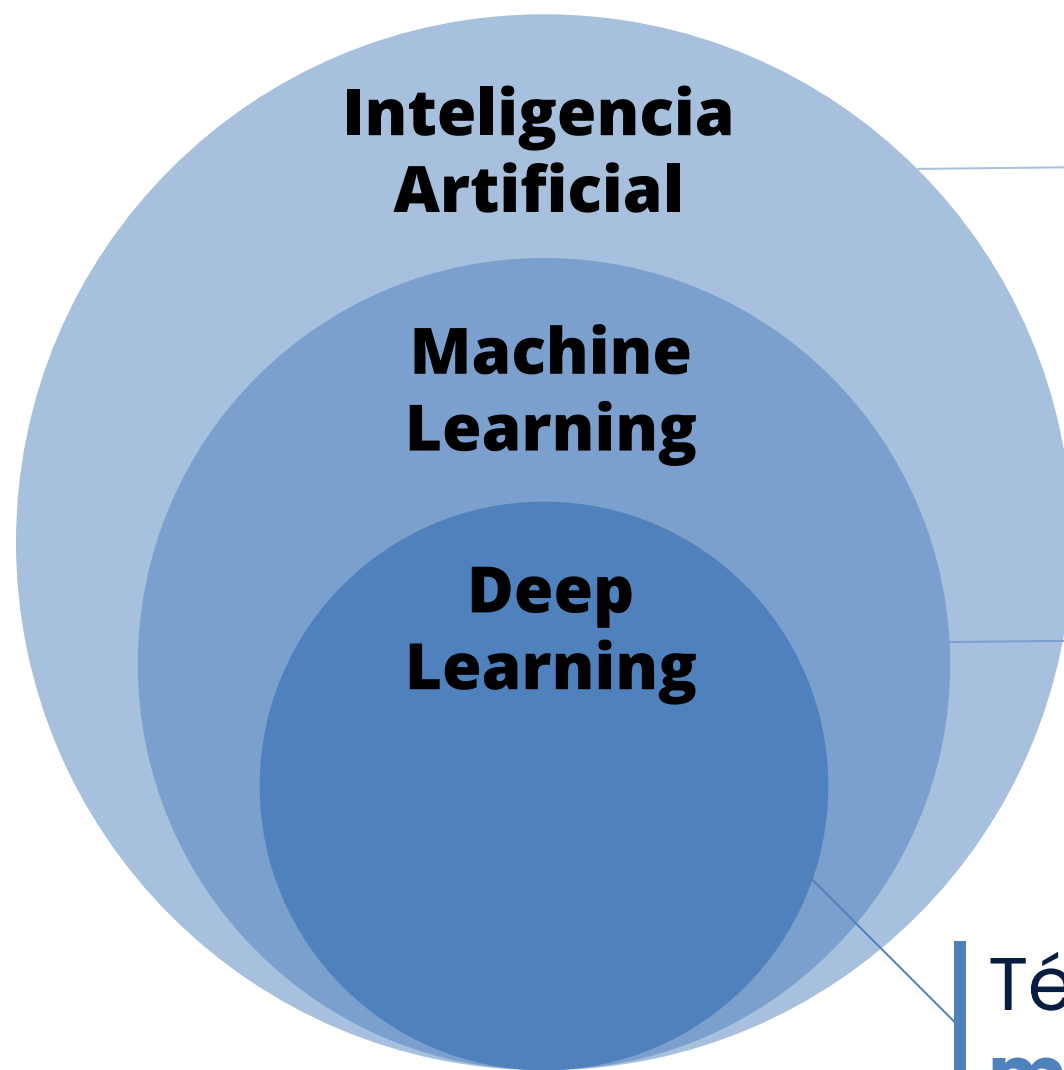
Past data



Learns from past data



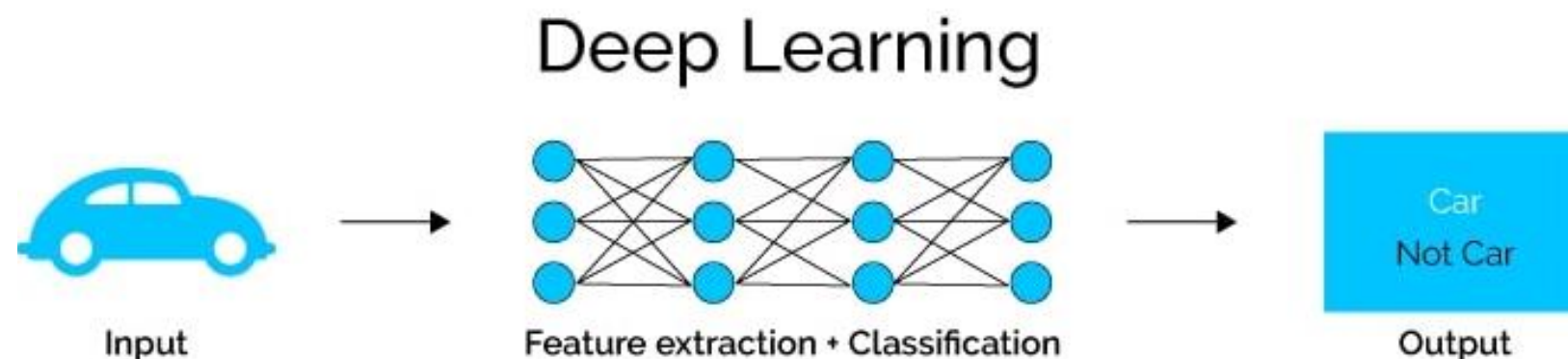
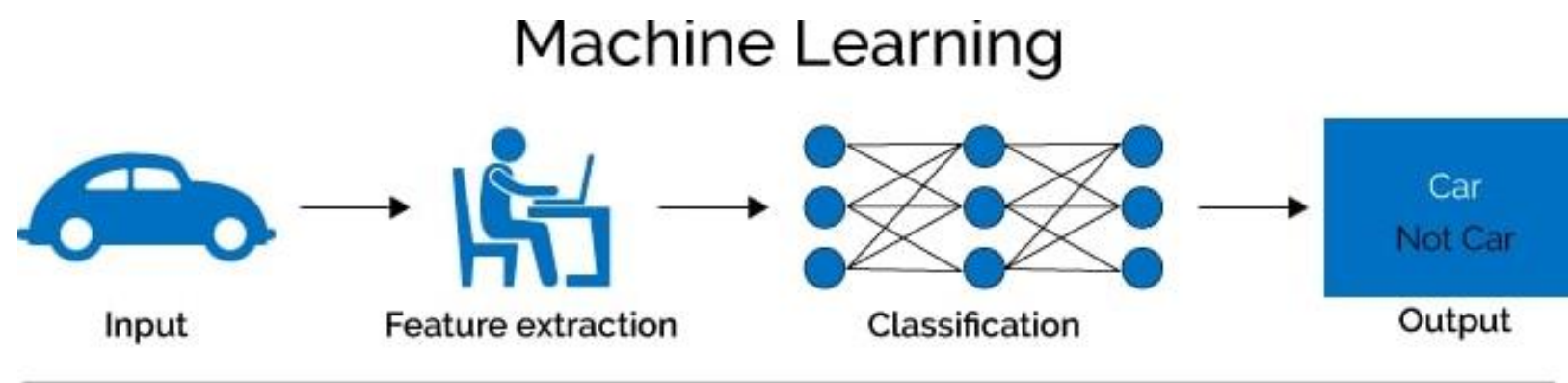
Predicts the output



Rama de la informática orientada a conseguir que las máquinas **simulen comportamientos humanos y tomen decisiones** de forma autónoma

Subcampo de la IA basado en la **estadística** para construir **modelos predictivos** que aprenden de **grandes cantidades de datos**.

Técnica de ML capaz de aprender **patrones más complejos** utilizando **redes neuronales**.



Tipos de aprendizaje automático

Aprendizaje supervisado

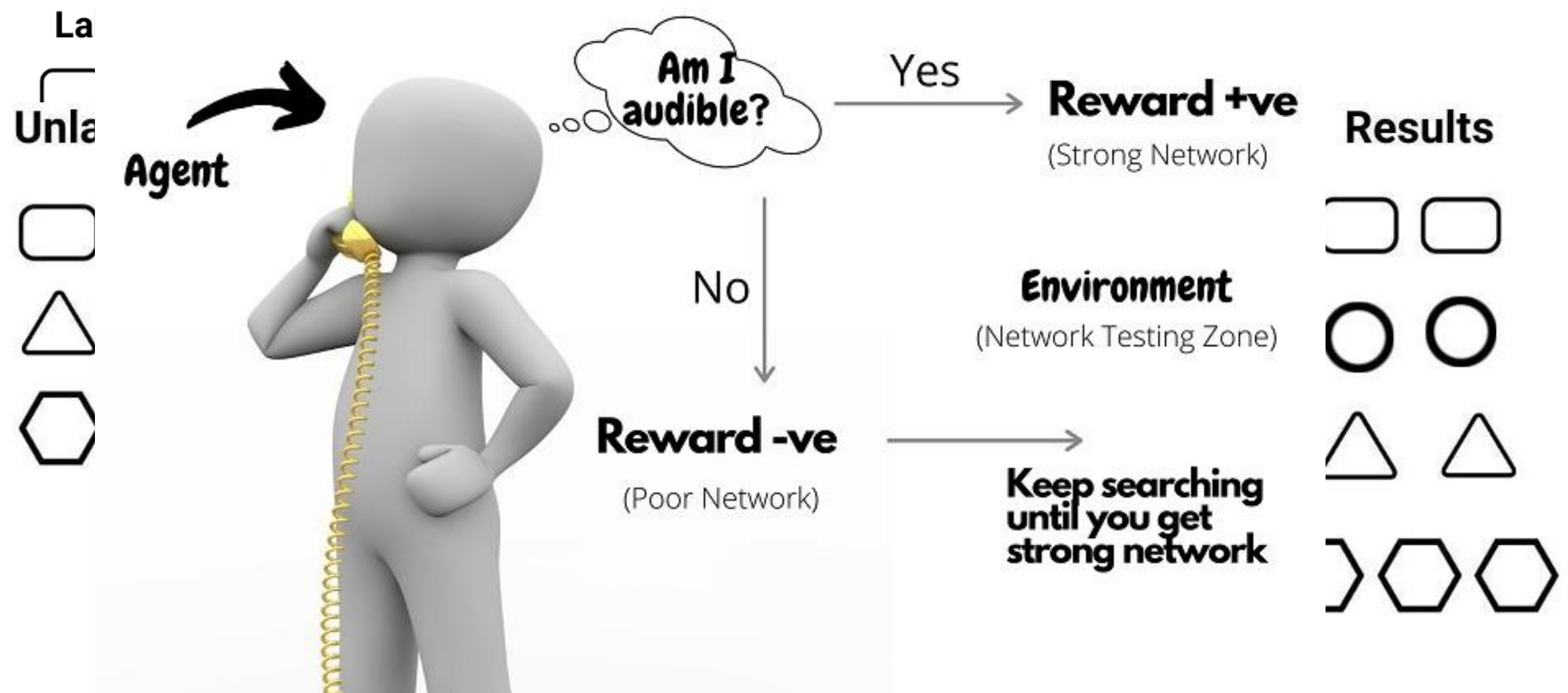
Relaciona variables entrada – salida deseada

Aprendizaje no supervisado

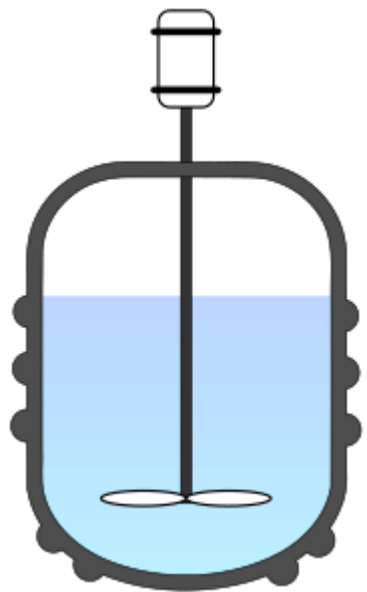
Descubre patrones en los datos de entrada

Aprendizaje por refuerzo (RL)

Método de prueba y error



Redes neuronales



T



F_0



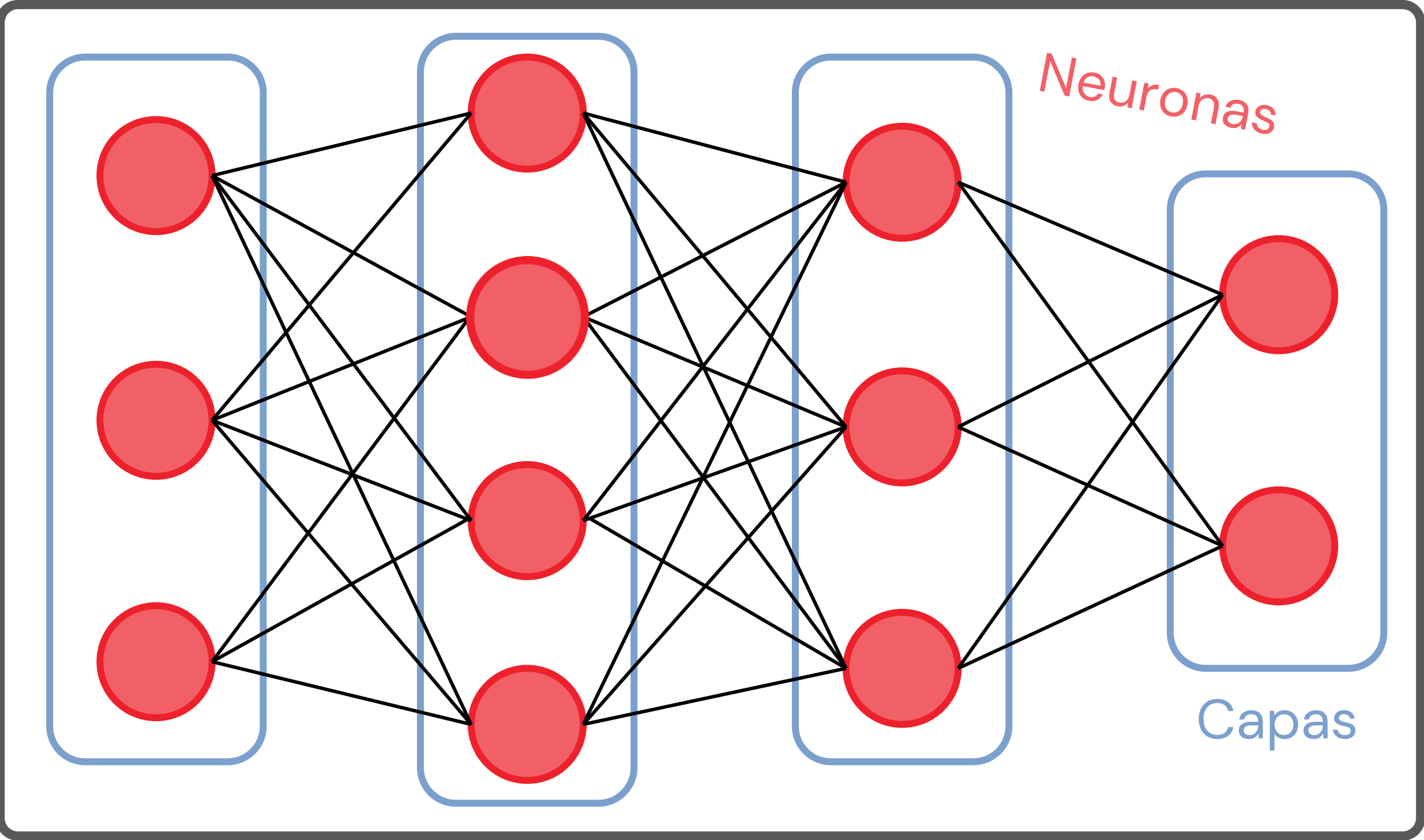
C_0



Pureza

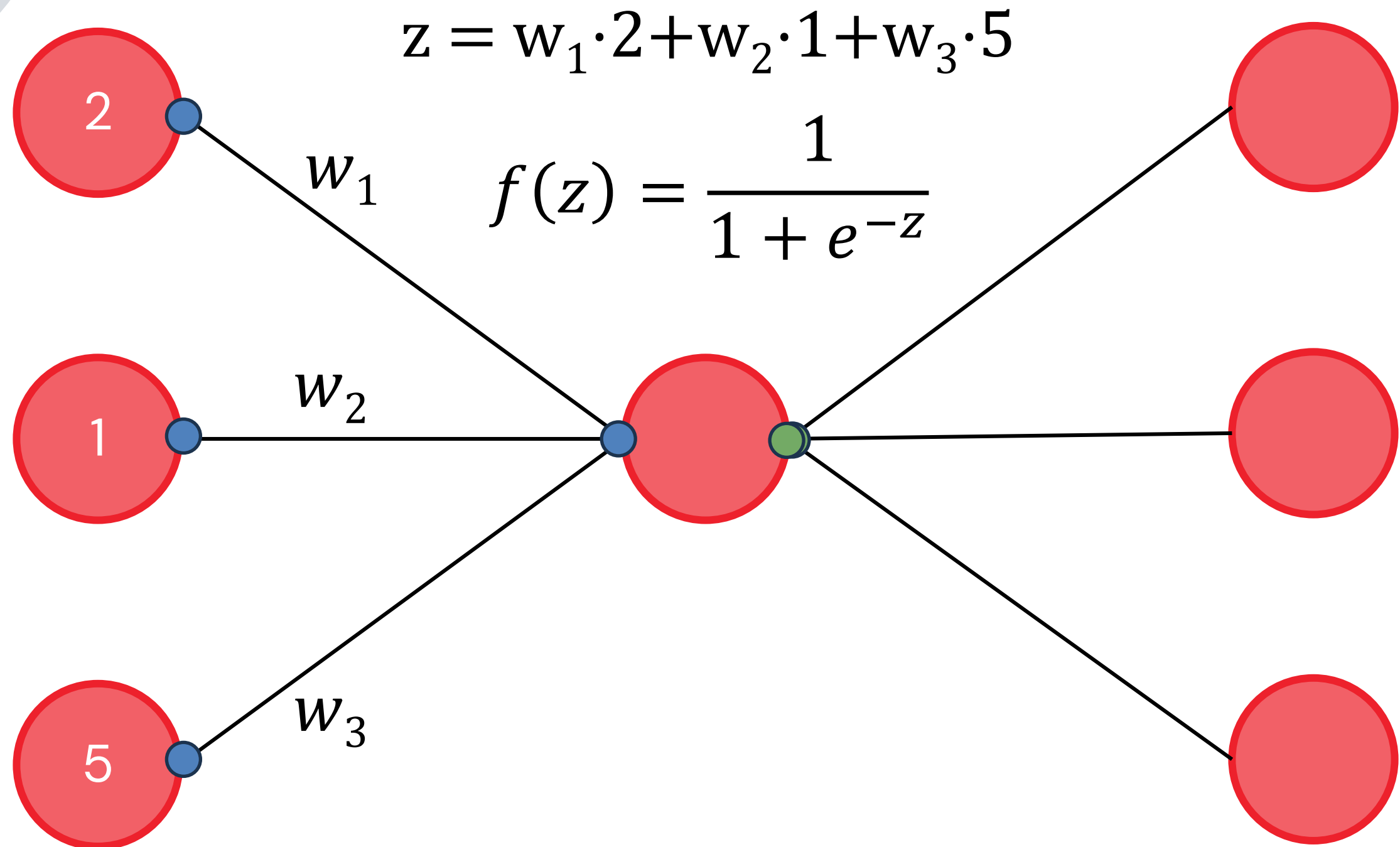


Viscosidad



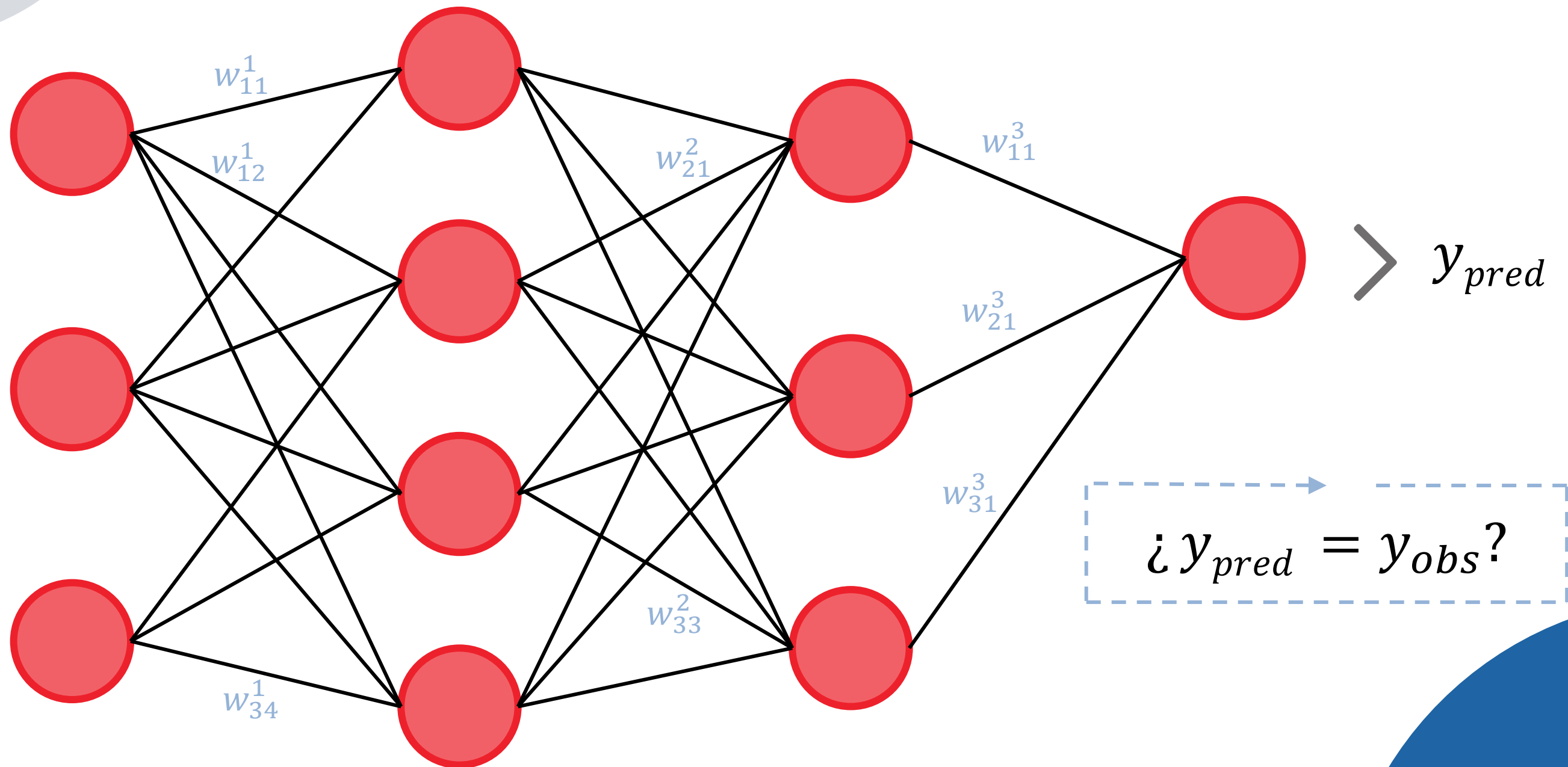
Redes neuronales

Forward propagation



Redes neuronales

Backpropagation



Redes neuronales

Backpropagation

- Para resolver el siguiente problema de optimización...

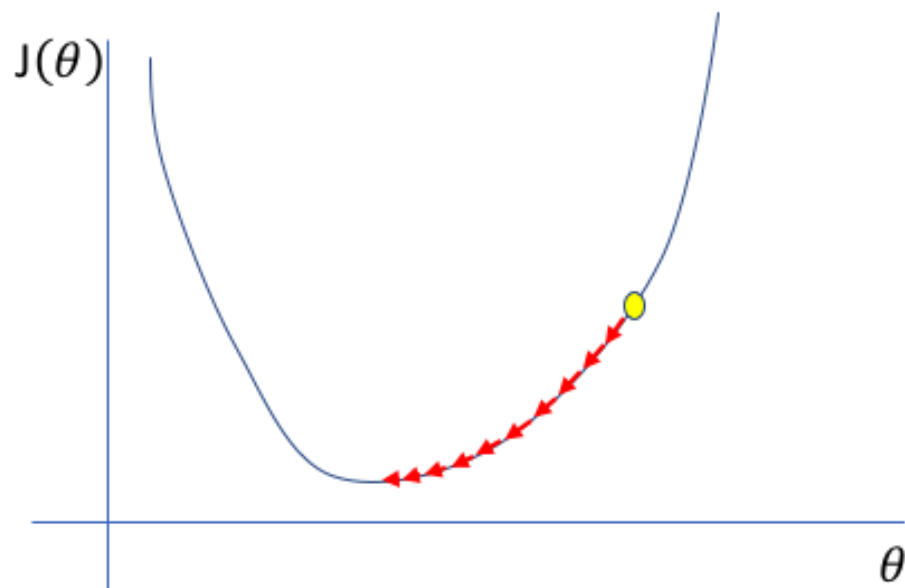
$$\min \mathcal{L} = \sum_{i=1}^{N_{points}} (y_{obs,i} - y_{pred,i})^2$$

- ... se utilizan algoritmos basados en el **método del descenso del gradiente**

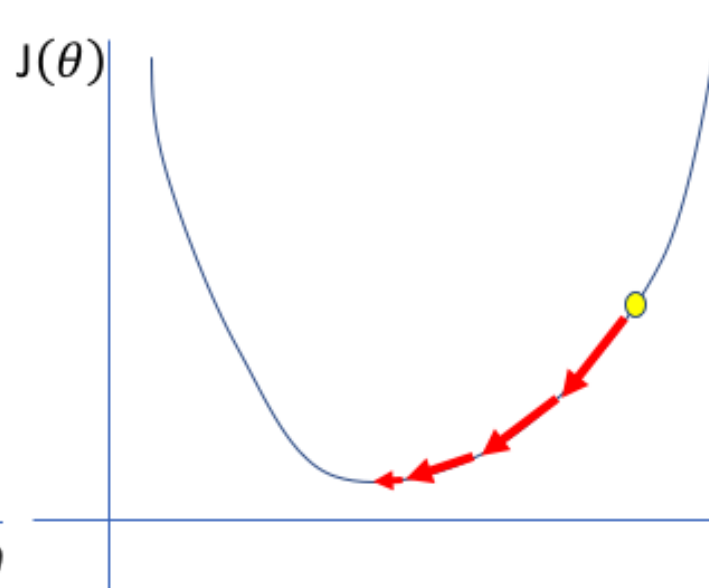
$$w_{new} = w_{old} - \alpha \frac{\delta \mathcal{L}}{\delta w}$$

Tamaño del paso
o *learning rate*

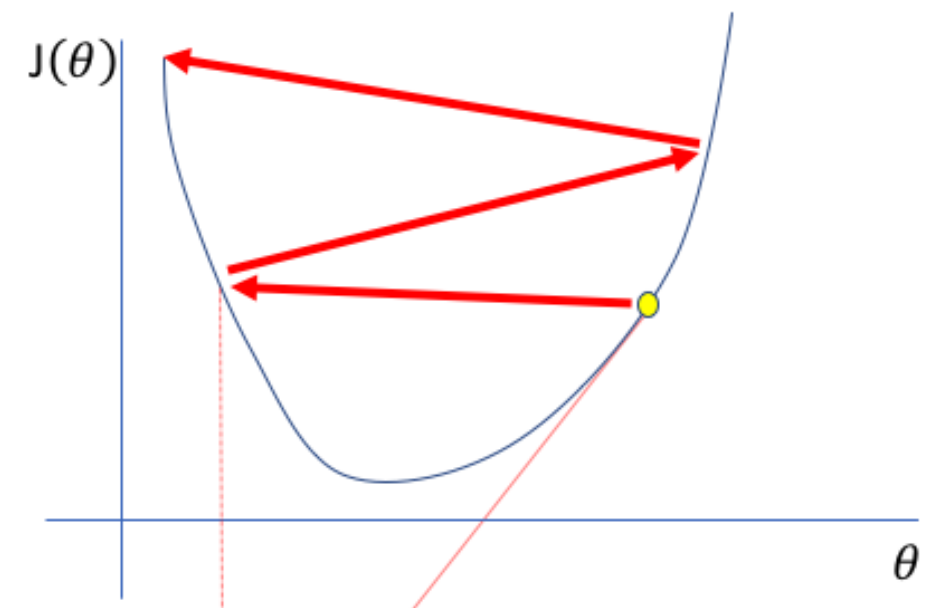
Too low



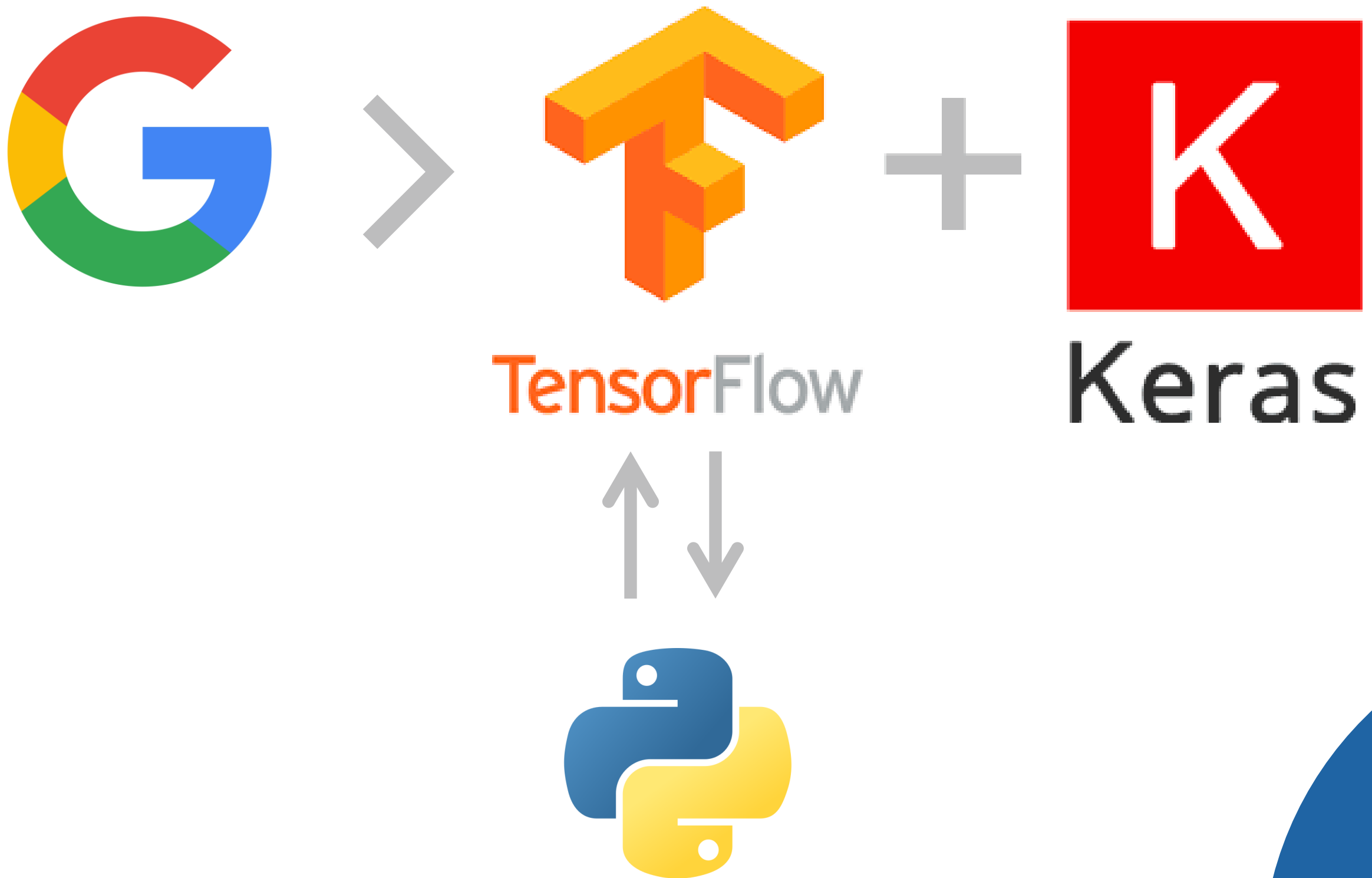
Just right



Too high



Introducción a TensorFlow

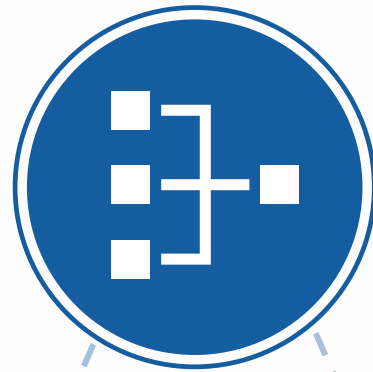


7 pasos para construir un modelo

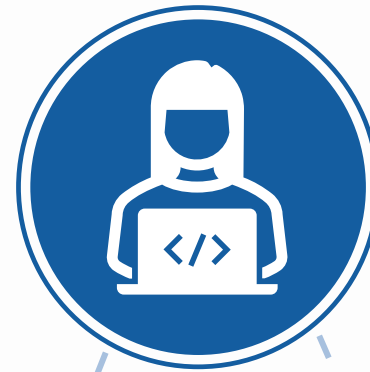
**Recopilación
de datos**



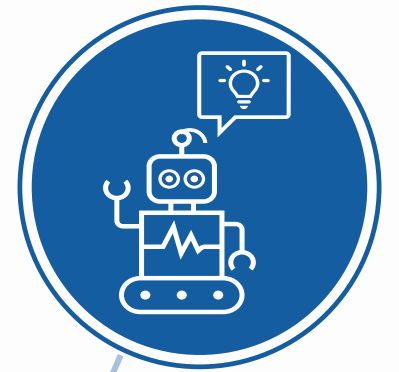
**Selección del
modelo**



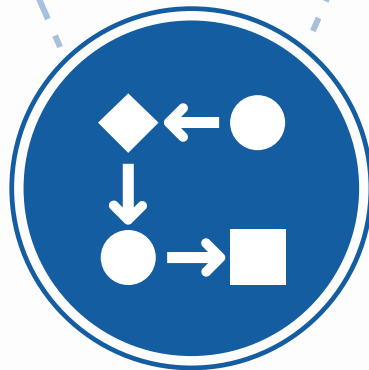
**Evaluación del
modelo**



Predicción



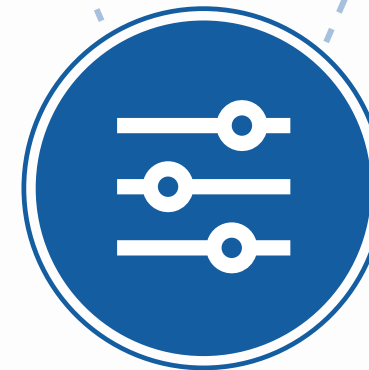
**Preprocesamiento
de datos**



**Entrenamiento
del modelo**

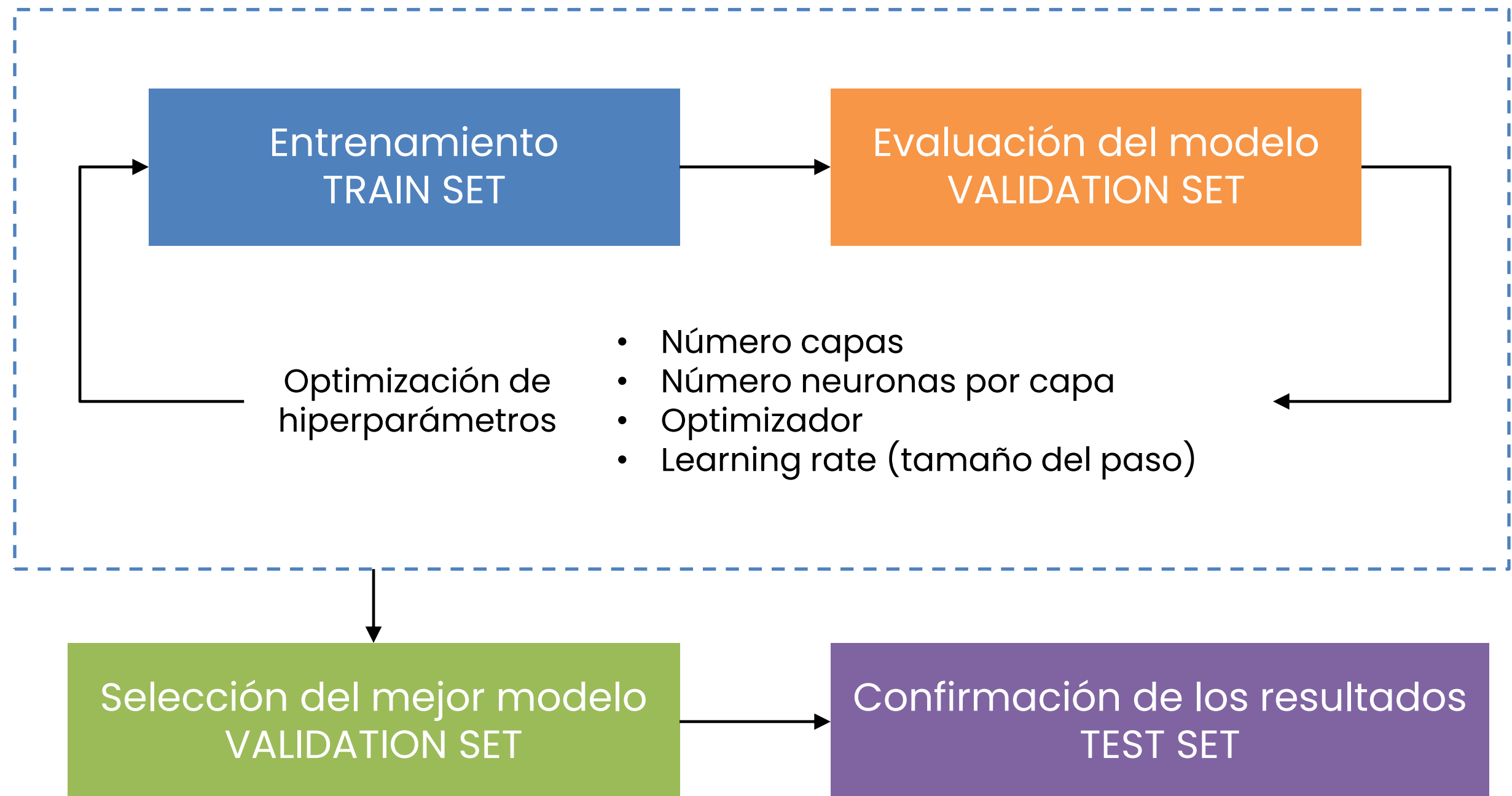


**Ajuste de
hiperparámetros**



7 pasos para construir un modelo

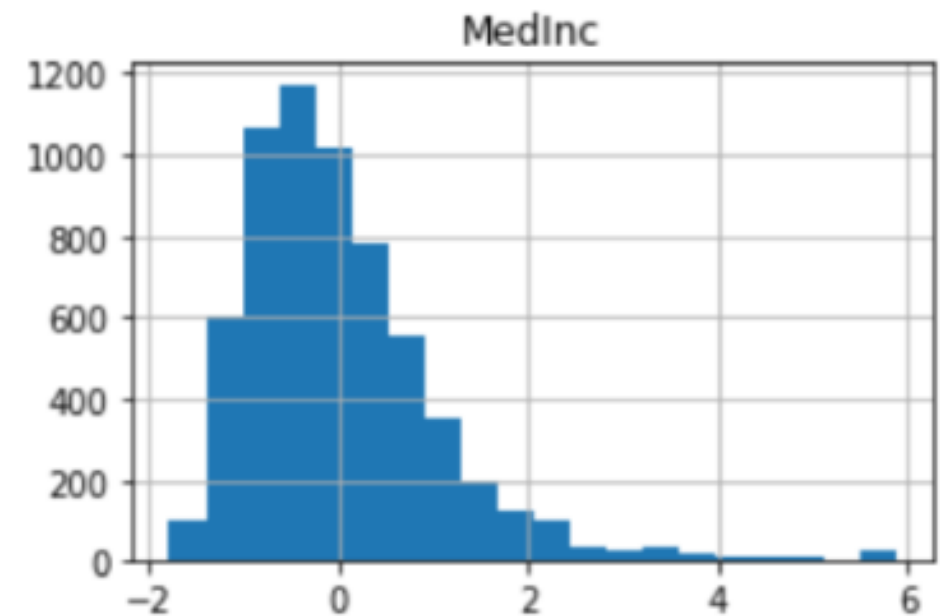
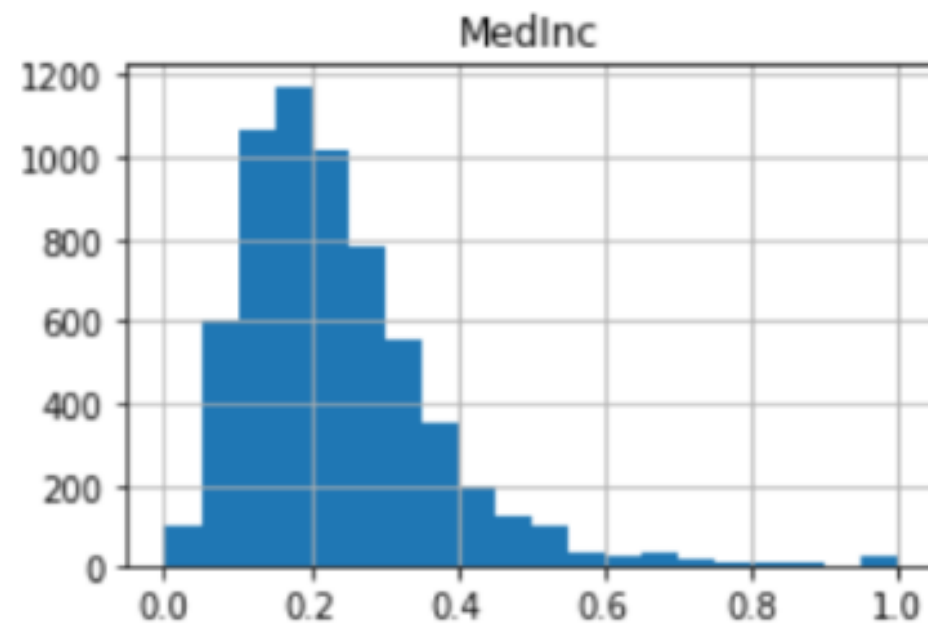
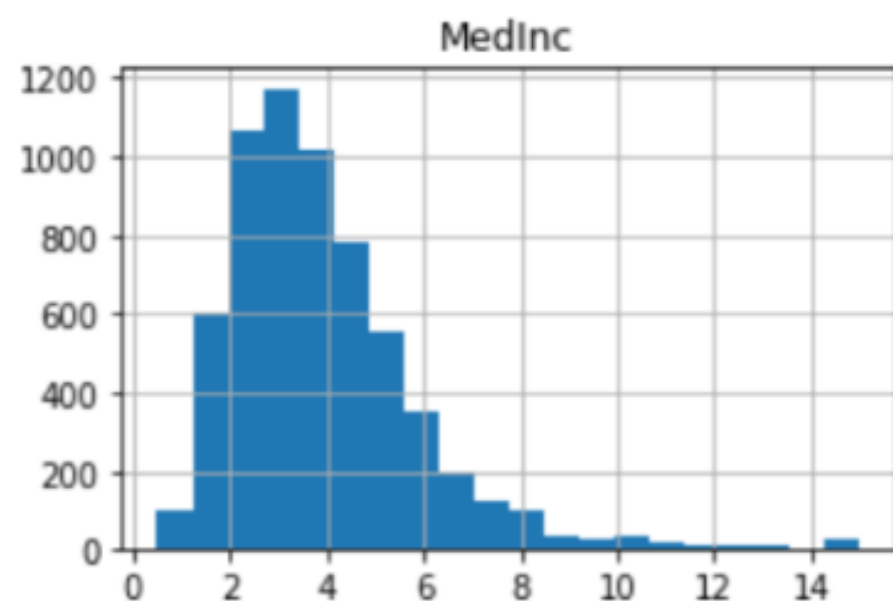
Preprocesamiento de datos: train-test-validation split



7 pasos para construir un modelo

Preprocesamiento de datos: Escalado

- Cambio en la magnitud de las variables sin afectar su distribución.
- Facilita convergencia algoritmo optimización



Min-max

$$X_{scaled} = \frac{X - X_{min}}{X - X_{max}}$$

Estandarización

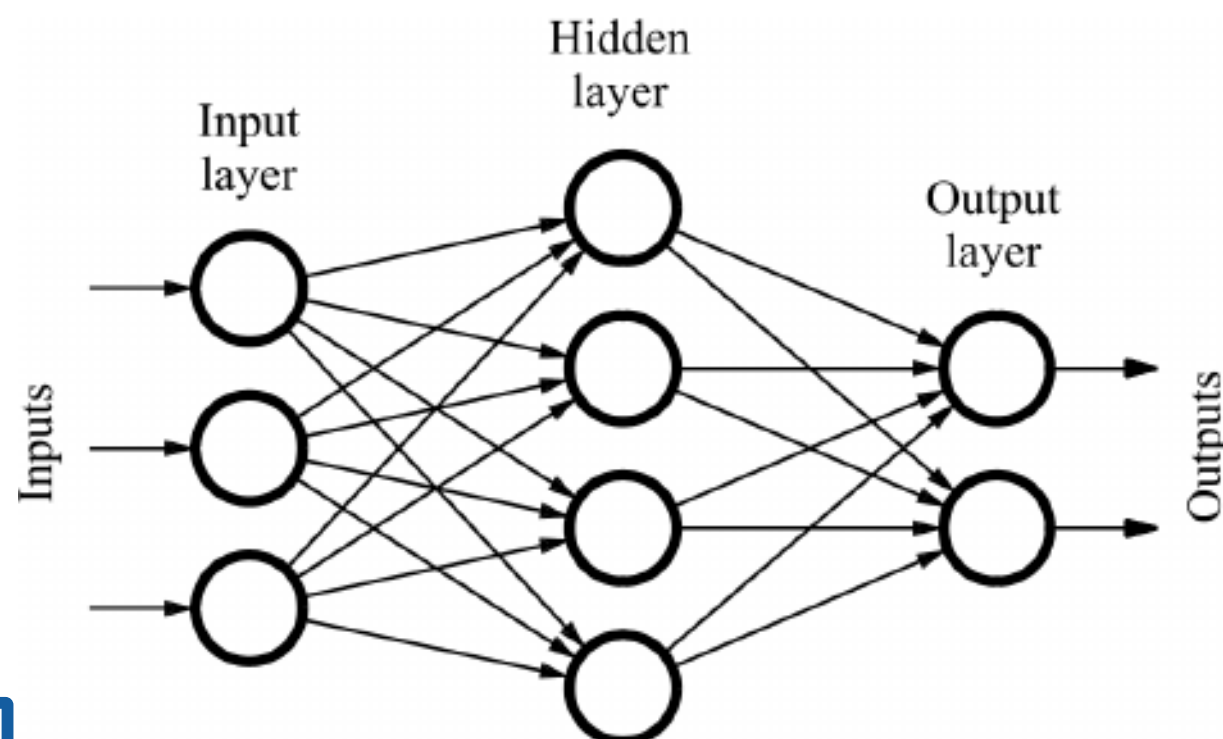
$$X_{scaled} = \frac{X - \mu_X}{\sigma_X}$$

7 pasos para construir un modelo

Selección del modelo

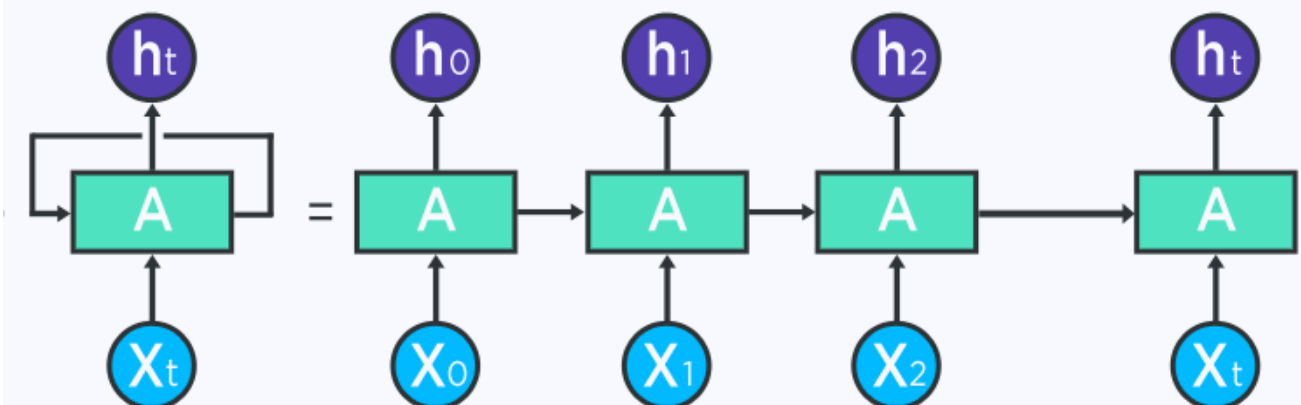
Dense neural networks

```
keras.layers.Dense(  
    units,  
    activation=None,  
    use_bias=True,  
    kernel_initializer="glorot_uniform",  
    **kwargs  
)
```



Recurrent neural networks

```
keras.layers.RNN(  
    cell,  
    return_sequences=False,  
    return_state=False,  
    go_backwards=False,  
    stateful=False,  
    unroll=False,  
    zero_output_for_mask=False,  
    **kwargs  
)
```



7 pasos para construir un modelo

Entrenamiento, evaluación y predicción

Método `compile`

1

```
model.compile(  
    optimizer="rmsprop",  
    loss=None,  
    metrics=None,  
)
```

Método `fit`

2

```
model.fit(  
    x=None,  
    y=None,  
    batch_size=None,  
    epochs=1,  
    verbose="auto",  
    callbacks=None,  
    validation_split=0.0,  
    validation_data=None,  
    shuffle=True,  
)
```

Método `predict`

3

```
model.predict(  
    x,  
    batch_size=None,  
    verbose="auto",  
    steps=None,  
    callbacks=None,  
)
```

Método `evaluate`

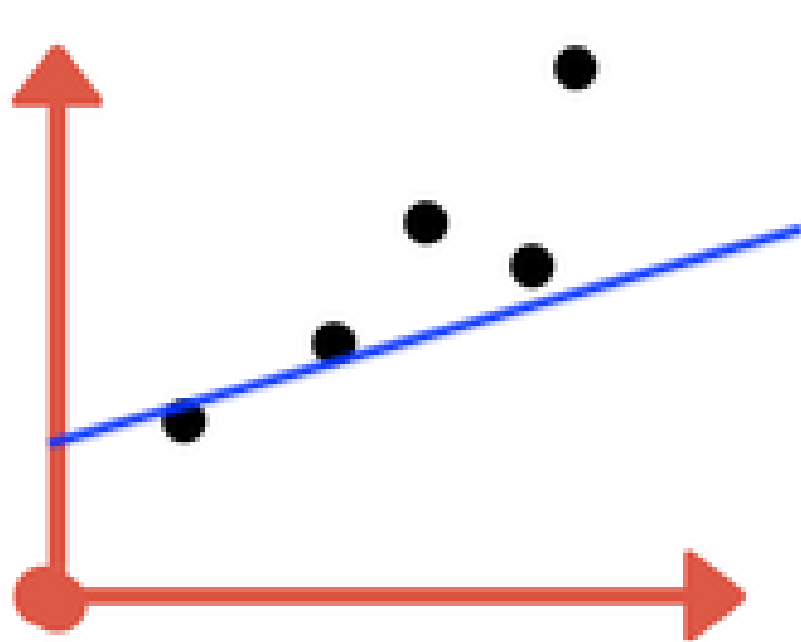
4

```
model.evaluate(  
    x=None,  
    y=None,  
    verbose="auto",  
    callbacks=None,  
    return_dict=False,  
)
```

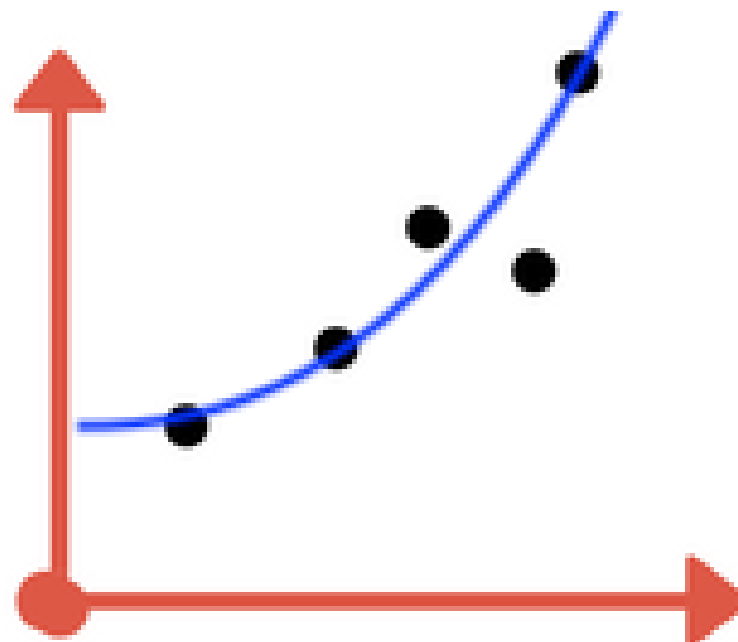


Sobreajuste y subajuste

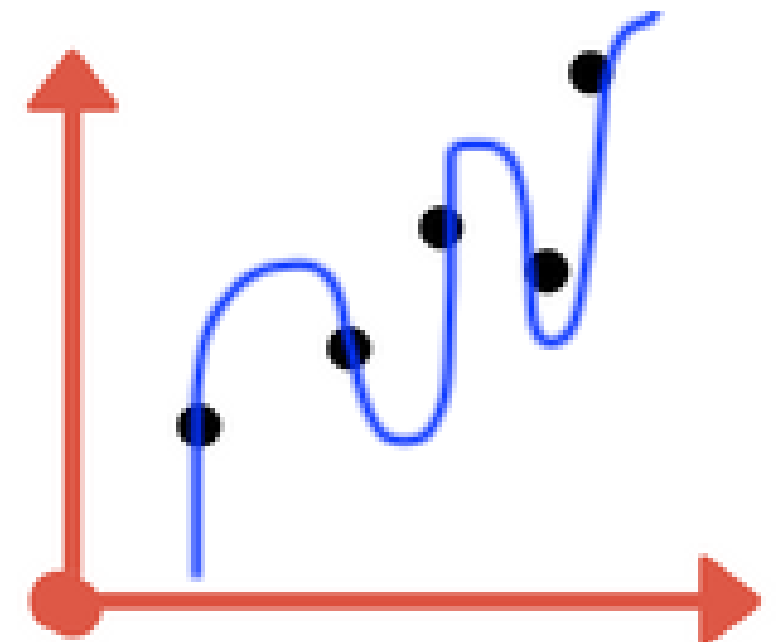
Complejidad del modelo



underfitting



correcto



overfitting

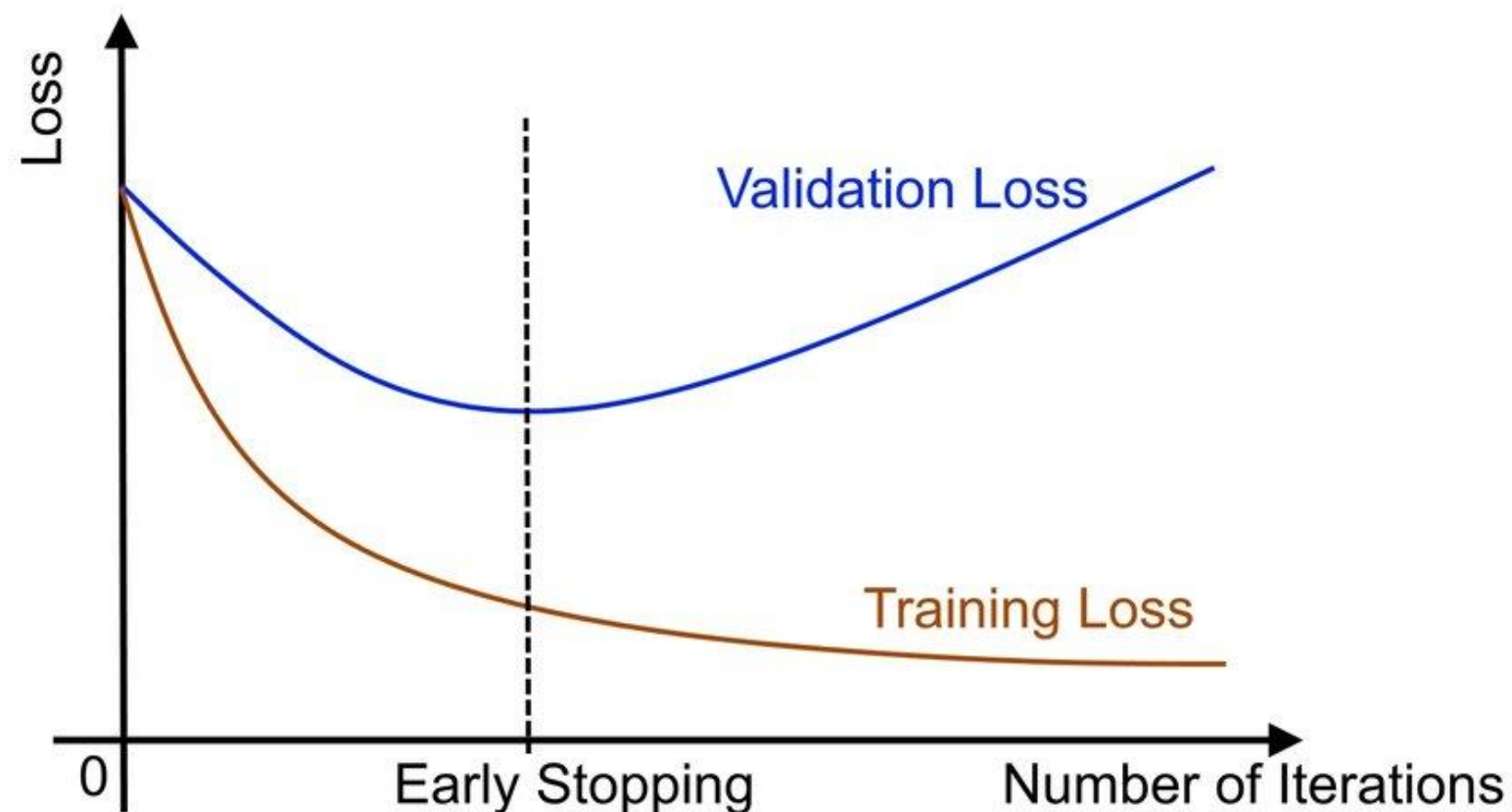
Sobreajuste y subajuste

¿Cómo evitarlo?

Early stopping

- Interrupción del entrenamiento cuando el error en validación empeora.

```
keras.callbacks.EarlyStopping(  
    monitor="val_loss",  
    min_delta=0,  
    patience=0,  
    verbose=0, restore_best_weights=False,  
    start_from_epoch=0,  
)
```



Sobreajuste y subajuste

¿Cómo evitarlo?

Early stopping

Regularización

- Penalización añadida en *loss function* proporcional al valor de los pesos

```
layer = layers.Dense(  
    units=64,  
    kernel_regularizer=regularizers.L1L2(l1=1e-5, l2=1e-4),  
    bias_regularizer=regularizers.L2(1e-4),  
    activity_regularizer=regularizers.L2(1e-5)  
)
```

$$\mathcal{L} = \sum_{i=1}^{N_{points}} (y_{obs,i} - y_{pred,i})^2$$

$$+ \lambda \sum_{j=1}^{N_{param}} |w_j|$$

L1
Lasso

$$+ \lambda \sum_{j=1}^{N_{param}} w_j^2$$

L2
Ridge



Sobreajuste y subajuste

¿Cómo evitarlo?

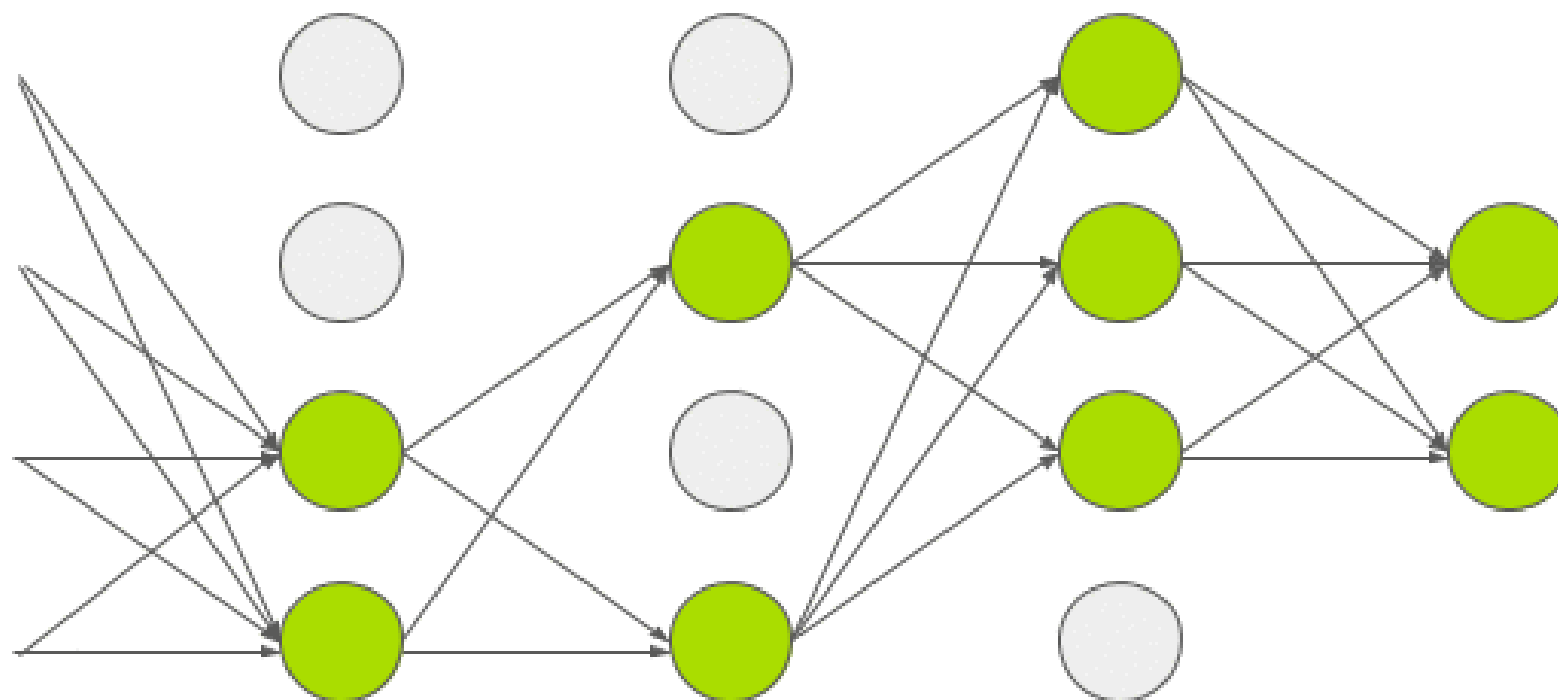
Early stopping

Regularización

Dropout

- Desactivación aleatoria de neuronas durante el entrenamiento.

`keras.layers.Dropout(rate, noise_shape=None, seed=None, **kwargs)`



Ajuste de hiperparámetros

Hiperparámetros

Definición: parámetro cuyo valor se utiliza para controlar el proceso de entrenamiento.

Suelen fijarse **antes de iniciar el entrenamiento.**

La **mejor combinación** se encuentra mediante un **proceso iterativo.**

01

Definir el espacio de búsqueda

```
hp = keras_tuner.HyperParameters()  
  
print(hp.Int("neuronas",  
            min_value=32,  
            max_value=512,  
            step=32  
      )  
    )
```



Ajuste de hiperparámetros

Hiperparámetros

Definición: parámetro cuyo valor se utiliza para controlar el proceso de entrenamiento.

Suelen fijarse **antes de iniciar el entrenamiento**.

La **mejor combinación** se encuentra mediante un **proceso iterativo**.

01

Definir el espacio de búsqueda

02

Seleccionar el tipo de búsqueda

- Random Search
- Grid Search
- Bayesian Search
- Hyperband Search

```
tuner = keras_tuner.RandomSearch(  
    hypermodel=build_model,  
    objective="val_accuracy",  
    max_trials=3,  
    executions_per_trial=2,  
    overwrite=True,  
)
```



Ajuste de hiperparámetros

Hiperparámetros

Definición: parámetro cuyo valor se utiliza para controlar el proceso de entrenamiento.

Suelen fijarse **antes de iniciar el entrenamiento**.

La **mejor combinación** se encuentra mediante un **proceso iterativo**.



01

Definir el espacio de búsqueda

02

Seleccionar el tipo de búsqueda

03

Comenzar la búsqueda

- Método **search** utiliza los mismos argumentos que el método **fit**.

```
tuner.search(  
    x_train,  
    y_train,  
    epochs=2,  
    validation_data=(x_val, y_val)  
)
```

Ajuste de hiperparámetros

Hiperparámetros

Definición: parámetro cuyo valor se utiliza para controlar el proceso de entrenamiento.

Suelen fijarse **antes de iniciar el entrenamiento.**

La **mejor combinación** se encuentra mediante un **proceso iterativo.**

01

Definir el espacio de búsqueda

02

Seleccionar el tipo de búsqueda

03

Comenzar la búsqueda

04

Inspeccionar los resultados

```
models = tuner.get_best_models(num_models=2)
best_model = models[0]
best_model.summary()
```





Universitat d'Alacant
Universidad de Alicante



Mes Cultural y de
Divulgación EPS 2025

Introducción al *machine learning* con TensorFlow



Isabela Fons

Estudiante de doctorado
Grupo COnCEPT

 isabela.fons@ua.es

 INSTITUTOS UNIVERSITARIOS II – P2
(0204P2049)