

# Hoja informativa: Redes totalmente conectadas

## Práctica

```
# Entrenamiento de regresión lineal en Keras
# importa Keras
from tensorflow import keras

# crea el modelo
model = keras.models.Sequential()
# indica cómo está organizada la red neuronal
# unidades: el número de neuronas en la capa
# input_dim: el número de inputs en la capa
model.add(keras.layers.Dense(units=1, input_dim=features.shape[1]))
# indica cómo está entrenada la red neuronal
model.compile(loss='mean_squared_error', optimizer='sgd')

# entrena el modelo
model.fit(features, target)
```

```
# Entrenamiento de regresión logística en Keras
# importa Keras
from tensorflow import keras

# crea el modelo
model = keras.models.Sequential()
# indica cómo está organizada la red neuronal
# unidades: el número de neuronas en la capa
# input_dim: el número de inputs en la capa
# activación: función de activación
model.add(keras.layers.Dense(units=1, input_dim=features_train.shape[1],
                             activation='sigmoid'))
# indica cómo está entrenada la red neuronal
model.compile(loss='binary_crossentropy', optimizer='sgd')

# entrena el modelo
model.fit(features, target)
```

```
# Entrenamiento de redes neuronales totalmente conectadas en Keras
# importa Keras
from tensorflow import keras
```

```

# crea el modelo
model = keras.models.Sequential()
# indica cómo está organizada la red neuronal
# Tenemos dos capas: la primera tiene 10 neuronas, la segunda tiene una
# unidades: el número de neuronas en la capa
# input_dim: el número de inputs en la capa
# activación: función de activación
model.add(keras.layers.Dense(units=10, input_dim=features_train.shape[1],
                             activation='sigmoid'))
model.add(keras.layers.Dense(units=1, activation='sigmoid'))

# indica cómo está entrenada la red neuronal
model.compile(loss='binary_crossentropy', optimizer='sgd', metrics=['acc'])

# entrena el modelo
model.fit(features, target)

```

```

# Trabajar con imágenes en Python
import numpy as np
from PIL import Image

# Importar la imagen
image = Image.open('image.png')
image_array = np.array(image)
print(image_array)

# Trazar la imagen
plt.imshow(image_array)

# Trazar la imagen en blanco y negro
plt.imshow(image_array, cmap='gray')

# Agregar la barra de color a la imagen
plt.colorbar()

```