

Curso de IA

# Capítulo 8. Cuestionario

**Para estudiantes**

© 2023 SAMSUNG. Todos los derechos reservados.

La Oficina de Ciudadanía Corporativa de Samsung Electronics posee los derechos de autor de este documento.

Este documento es una propiedad literaria protegida por la ley de derechos de autor, por lo que está prohibida su reimpresión y reproducción sin permiso.

Para utilizar este documento fuera del plan de estudios de Samsung Innovation Campus, debe recibir el consentimiento por escrito del titular de los derechos de autor..

1. ¿Cuál es el problema que se plantea cuando el número de nodos es pequeño al analizar redes neuronales artificiales?

- 1 Si el número de nodos es pequeño, el número de operaciones a procesar es pequeño, por lo que se realiza rápidamente.
- 2 No se pueden crear límites de decisión complejos al realizar un modelo de análisis.
- 3 No se puede realizar el algoritmo de retropropagación que ajusta pesos y umbrales.
- 4 El número de nodos no tiene nada que ver con el modelo de análisis.

2. ¿Cuál es el concepto que se describe a continuación?

**Matriz multidimensional con 3 componentes: Rango, Forma, Tipo**

R= Un tensor.

3. ¿Qué ocurre si el error de validación aumenta constantemente cuando graficamos el error de validación para cada época utilizando el método de descenso de pendiente por lotes? Además, ¿cómo se puede resolver este problema?

R= lo que ocurre es un sobreajustándose y se puede resolver usando un Dropout, que apaga aleatoriamente algunas neuronas en cada iteración para evitar dependencia excesiva en ciertos patrones.

4. Compara el rendimiento de una CNN y una ANN, propón las 2 arquitecturas y evalúa cual tiene mejor desempeño. Puedes tomar el siguiente código como ejemplo:

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import matplotlib.pyplot as plt
import numpy as np

# Cargar el dataset CIFAR-10
(x_train, y_train), (x_test, y_test) = keras.datasets.cifar10.load_data()

# Normalización de los datos
x_train, x_test = x_train / 255.0, x_test / 255.0
```

5. Aplicar la función softmax cuando la salida de la red neuronal es  $(0.4, 2.0, 0.001, 0.32)^T$  y escribir el resultado.
6. Forme un modelo de predicción de series temporales basado en RNN con referencia al siguiente código.

```
importar pandas como pd
importar numpy como np
desde sklearn.model_selection importar train_test_split
desde sklearn.preprocessing importar MinMaxScaler
importar tensorflow como tf
importar matplotlib.pyplot como plt

# read data# Leer datos
df = pd.read_csv('data_boston.csv', header='infer', encoding='latin1')
df = df[['PRICE']]

# scale input & X, y
scaler = MinMaxScaler()
ts_scaled = scaler.fit_transform(df)

# scale
ts_scaled_2 = ts_scaled.reshape(1, -1, 1)

# training parameters
batch_size = 1
n_epochs = 1000
learn_rate = 0.0001

# model
model = tf.keras.Sequential()
model.add(tf.keras.Input(shape=(None, 1)))

# completar el modelo basado en la rnn y la capacitación
```