

Actividad 4

Aplicación de Redes Neuronales LSTM
en Series Temporales

Aplicación de Redes Neuronales LSTM en Series Temporales



Ahora que los pasos para construir una red neuronal LSTM para pronosticar series de tiempo están claros vamos a aplicarlo en un problema un poco más grande.

```
from keras.models import Sequential
from keras.layers import LSTM, Dense
import numpy as np
import matplotlib.pyplot as plt
```



Aplicación de Redes Neuronales LSTM en Series Temporales



Generación de datos de serie de tiempo sintética:

Se generan datos de serie de tiempo sintética creciente con ruido aleatorio agregado.

```
# Generar datos para una serie de tiempo sintética
np.random.seed(0) # Fijar semilla para reproducibilidad
n_samples = 200 # Número de puntos de datos en la serie de tiempo
t = np.arange(n_samples) # Valores de tiempo

# Crear una serie de tiempo sintética (en este caso, creciente)
time_series = 0.1*t + np.random.randn(n_samples)
```



Aplicación de Redes Neuronales LSTM en Series Temporales



División de datos:

Los datos de serie de tiempo se dividen en conjuntos de entrenamiento y prueba, con un 80% de los datos destinados al entrenamiento y el 20% restante al prueba.

```
train_size = int(len(time_series) * 0.8) # Tamaño del conjunto de entrenamiento
train_data, test_data = time_series[:train_size], time_series[train_size:]
```



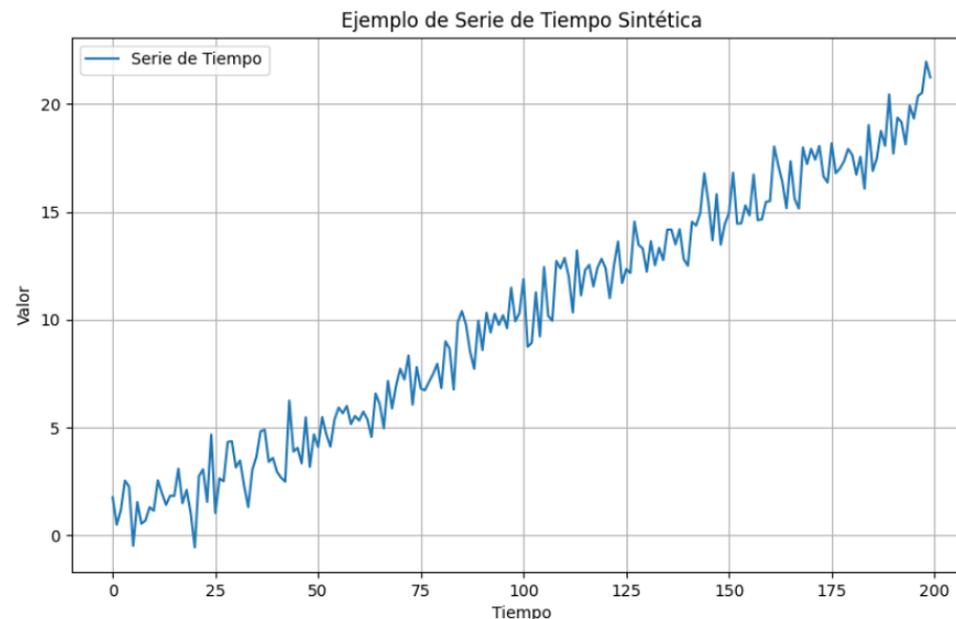
Aplicación de Redes Neuronales LSTM en Series Temporales



Visualización de la serie de tiempo:

Se grafica la serie de tiempo sintética para visualizar los datos.

```
# Graficar la serie de tiempo
plt.figure(figsize=(10, 6))
plt.plot(t, time_series, label='Serie de Tiempo')
plt.title('Ejemplo de Serie de Tiempo Sintética')
plt.xlabel('Tiempo')
plt.ylabel('Valor')
plt.legend()
plt.grid(True)
plt.show()
```



Aplicación de Redes Neuronales LSTM en Series Temporales



Preparación de datos para LSTM:

Se define una función llamada `create_dataset` que crea secuencias de datos de entrada y salida a partir de la serie de tiempo original. Estas secuencias se utilizarán para entrenar el modelo LSTM.

```
# transformamos nuestros datos en secuencias
def create_dataset(data, seq_length):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i:i + seq_length])
        y.append(data[i + seq_length])
    return np.array(X), np.array(y)

# Definimos el número de pasos de tiempo para la serie temporal
seq_length = 16

# Creamos conjuntos de entrenamiento y prueba en formato 3D
X_train, y_train = create_dataset(train_data, seq_length)
X_test, y_test = create_dataset(test_data, seq_length)
```



Aplicación de Redes Neuronales LSTM en Series Temporales



Construcción del modelo LSTM:

Se crea un modelo secuencial en Keras y se agregan capas LSTM con 32 unidades cada una, utilizando la función de activación ReLU. Se añade una capa densa para la salida.

```
# Construcción del modelo LSTM
model = Sequential()
model.add(LSTM(32,
              activation='relu',
              input_shape=(seq_length, 1),
              return_sequences=True))
model.add(LSTM(32, activation='relu'))
model.add(Dense(1))
```



Aplicación de Redes Neuronales LSTM en Series Temporales



Compilación del modelo:

El modelo se compila utilizando el optimizador Adam y la función de pérdida de error cuadrático medio (MSE).

```
model.compile(optimizer='adam', loss='mse')
```

Entrenamiento del modelo:

El modelo se entrena utilizando los datos de entrenamiento durante 100 épocas.

```
# Entrenamiento del modelo  
model.fit(X_train, y_train, epochs=100)
```

Evaluación del modelo:

Se evalúa el desempeño del modelo utilizando los datos de prueba, calculando la pérdida obtenida por el modelo.



```
# Evaluación del modelo  
Evaluación = model.evaluate(X_test, y_test)  
print(Evaluación)
```

Aplicación de Redes Neuronales LSTM en Series Temporales



Predicción:

Se realizan predicciones utilizando el modelo entrenado en los datos de prueba.

```
print("y_test: \n",y_test)
# Predicción
y_pred = model.predict(X_test)
print("y_pred: \n",y_pred)
```



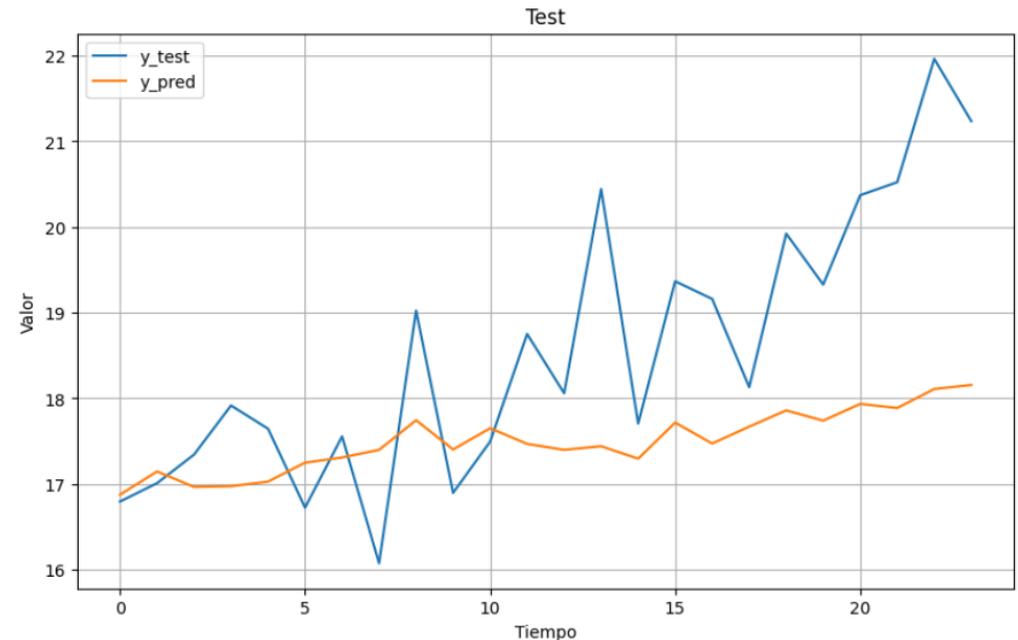
Aplicación de Redes Neuronales LSTM en Series Temporales



Visualización de las predicciones:

Se grafican los datos de prueba (`y_test`) junto con las predicciones (`y_pred`) para comparar el rendimiento del modelo. Las dos series de tiempo se visualizan en el mismo gráfico para facilitar la comparación.

```
# Graficar de la data de test vs pronostico la serie de tiempo
plt.figure(figsize=(10, 6))
plt.plot(y_test, label='y_test')
plt.plot(y_pred, label='y_pred')
plt.title('Test')
plt.xlabel('Tiempo')
plt.ylabel('Valor')
plt.legend()
plt.grid(True)
plt.show()
```



Ejercicios y preguntas relacionadas con el ejercicio anterior proporcionado

1. Conceptos básicos de LSTM y series de tiempo:

- ¿Qué es una serie de tiempo y por qué es importante en el análisis de datos?
- Describe brevemente cómo funciona una red neuronal LSTM y cuál es su ventaja para modelar series de tiempo.

2. Preprocesamiento de datos:

- ¿Por qué es importante dividir los datos de serie de tiempo en conjuntos de entrenamiento y prueba?
- ¿Qué hace la función `create_dataset` en el código?

3. Construcción y entrenamiento del modelo:

- Explica la estructura del modelo LSTM creado en el código.
- ¿Qué significan el parámetros `epochs` en la función `fit`?



Ejercicios y preguntas relacionadas con el ejercicio anterior proporcionado

4. Evaluación del modelo:

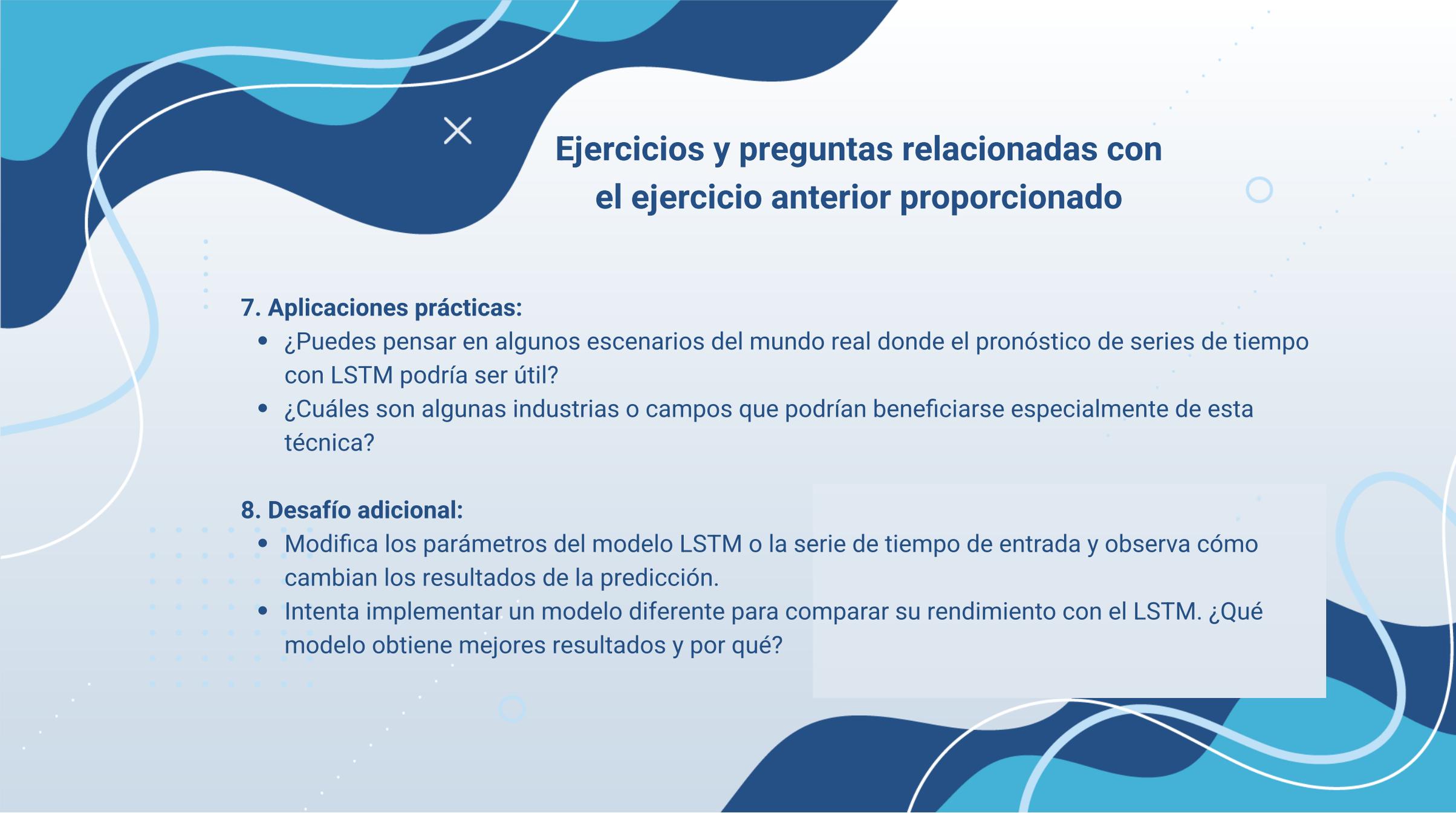
- ¿Por qué es importante evaluar el modelo con datos de prueba después del entrenamiento?
- ¿Qué medida se utiliza para evaluar el rendimiento del modelo en este caso?

5. Predicción y visualización:

- ¿Qué representa la variable `y_pred` en el código?
- ¿Qué significan los gráficos resultantes de la predicción? ¿Qué conclusiones se pueden sacar de ellos?

6. Experimentación y ajuste del modelo:

- ¿Cómo podrías mejorar el rendimiento del modelo LSTM en este caso?
- ¿Qué pasaría si aumentarás el número de unidades LSTM o la cantidad de épocas de entrenamiento?



Ejercicios y preguntas relacionadas con el ejercicio anterior proporcionado

7. Aplicaciones prácticas:

- ¿Puedes pensar en algunos escenarios del mundo real donde el pronóstico de series de tiempo con LSTM podría ser útil?
- ¿Cuáles son algunas industrias o campos que podrían beneficiarse especialmente de esta técnica?

8. Desafío adicional:

- Modifica los parámetros del modelo LSTM o la serie de tiempo de entrada y observa cómo cambian los resultados de la predicción.
- Intenta implementar un modelo diferente para comparar su rendimiento con el LSTM. ¿Qué modelo obtiene mejores resultados y por qué?



TIC

▶ **TALENTO**
TECH

AZ | **PROYECTOS**
EDUCATIVOS

