

LSTM cho bài toán dự báo nhiệt độ 1 cột

1. Import các thư viện:

```
import tensorflow as tf
from keras.layers import LSTM, Dense
import pandas as pd
```

Các thư viện TensorFlow, Keras (dùng để xây dựng mô hình LSTM), và Pandas (dùng để xử lý dữ liệu) được import. `drive` từ Google Colab cho phép truy cập Google Drive để đọc dữ liệu.

2. Đọc file CSV:

```
from google.colab import drive
import pandas as pd
drive.mount('/content/drive')

csv_file = "/content/drive/My Drive/ĐỒ ÁN DEEP/climate.csv"
df = pd.read_csv(csv_file, sep=',')
data = df.iloc[:, [2]].values
print(data)
```

Tập dữ liệu `climate.csv` chứa thông tin về khí hậu được tải từ Google Drive. Cột thứ hai của dữ liệu được chọn và lưu vào `data`.

3. Chuẩn hoá dữ liệu:

```
def normalize(data, train_split):
    data_mean = data[:train_split].mean(axis=0)
    data_std = data[:train_split].std(axis=0)
    return (data - data_mean) / data_std
```

Hàm `normalize` chuẩn hoá dữ liệu, đưa giá trị trung bình về 0 và độ lệch chuẩn về 1, chỉ áp dụng trên tập huấn luyện (`train_split`).

4. Tách dữ liệu huấn luyện và kiểm tra:

```
train_split = int(0.715 * int(df.shape[0]))
data = normalize(data, train_split)
train_data = data[:train_split]
val_data = data[train_split:]
```

Tập dữ liệu được chia thành tập huấn luyện (71.5% dữ liệu) và tập kiểm tra, sau khi chuẩn hoá.

5. Xây dựng dữ liệu chuỗi thời gian:

```
#Xác định phương thức dự báo
past = 720
future = 72
step = 6
batch_size = 256
```

Các tham số xác định chuỗi thời gian **past** (số điểm dữ liệu quá khứ), **future** (số điểm dự báo), **step** (bước nhảy), và **batch_size**.

6. Tạo tập dữ liệu huấn luyện và kiểm tra từ chuỗi thời gian:

```
dataset_train = tf.keras.preprocessing.timeseries_dataset_from_array(
    x_train,
    y_train,
    sequence_length=sequence_length,
    sampling_rate=step,
    batch_size=batch_size,
)
```

```
dataset_val = tf.keras.preprocessing.timeseries_dataset_from_array(
    x_val,
    y_val,
    sequence_length=sequence_length,
    sampling_rate=step,
    batch_size=batch_size,
)
```

timeseries_dataset_from_array giúp tạo các mẫu dữ liệu theo chuỗi thời gian, từ **x_train** và **x_val**.

7. Xây dựng mô hình LSTM:

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Input(shape=(sequence_length, 1)))
model.add(tf.keras.layers.LSTM(32))
model.add(tf.keras.layers.Dense(1))
```

Model: "sequential_4"

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 32)	4,352
dense_1 (Dense)	(None, 1)	33

Total params: 4,385 (17.13 KB)

Trainable params: 4,385 (17.13 KB)

Non-trainable params: 0 (0.00 B)

Mô hình Sequential gồm:

- **Input layer:** Nhận dữ liệu chuỗi thời gian với độ dài `sequence_length`.
- **LSTM layer:** Với 32 đơn vị, để học các mẫu chuỗi thời gian.
- **Dense layer:** Đầu ra 1 đơn vị dự báo giá trị liên tục.

8. Huấn luyện mô hình:

```
epochs = 10
history = model.fit(
    dataset_train,
    epochs=epochs,
    validation_data=dataset_val,
    #callbacks=[es_callback, modelckpt_callback],
)
```

Mô hình được huấn luyện với tập huấn luyện và đánh giá trên tập kiểm tra qua `epochs` lần.

```
Epoch 1/10
1172/1172 ————— 203s 171ms/step - loss: 0.3286 - val_loss: 0.1606
Epoch 2/10
1172/1172 ————— 201s 171ms/step - loss: 0.1492 - val_loss: 0.1601
Epoch 3/10
1172/1172 ————— 201s 171ms/step - loss: 0.1404 - val_loss: 0.1655
Epoch 4/10
1172/1172 ————— 199s 170ms/step - loss: 0.1374 - val_loss: 0.1658
Epoch 5/10
1172/1172 ————— 198s 169ms/step - loss: 0.1365 - val_loss: 0.1648
Epoch 6/10
1172/1172 ————— 200s 171ms/step - loss: 0.1346 - val_loss: 0.1623
Epoch 7/10
1172/1172 ————— 203s 173ms/step - loss: 0.1334 - val_loss: 0.1595
Epoch 8/10
1172/1172 ————— 259s 171ms/step - loss: 0.1326 - val_loss: 0.1562
Epoch 9/10
1172/1172 ————— 199s 169ms/step - loss: 0.1317 - val_loss: 0.1536
Epoch 10/10
1172/1172 ————— 201s 171ms/step - loss: 0.1308 - val_loss: 0.1525
```

9. Hàm vẽ đồ thị:

```
import matplotlib.pyplot as plt

def visualize_loss(history, title):
    # Lấy thông tin loss từ history
    loss = history.history["loss"]
    val_loss = history.history["val_loss"]

    # Tạo danh sách các epochs
    epochs = range(len(loss))

    # Vẽ biểu đồ
    plt.figure()
    plt.plot(epochs, loss, "b", label="Training loss")
    plt.plot(epochs, val_loss, "r", label="Validation loss")
    plt.title(title)
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()
    plt.show()

# Giả sử bạn đã huấn luyện mô hình và có biến history
# history = model.fit(...) # Kết quả huấn luyện
visualize_loss(history, "Training and Validation Loss")
```

Hàm `visualize_loss` để vẽ biểu đồ của hàm mất mát trên tập huấn luyện và kiểm tra, giúp kiểm tra sự hội tụ của mô hình.

