

Bài tập 8: huấn luyện mô hình LSTM

1. Import các thư viện

```
python
Sao chép mã
import tensorflow as tf
from keras.api.layers import LSTM, Dense
from keras.api.preprocessing import timeseries_dataset_from_array
import pandas as pd
import matplotlib.pyplot as plt
```

- tensorflow và keras: Xây dựng và huấn luyện mô hình LSTM.
- pandas: Quản lý và xử lý dữ liệu dạng bảng.
- matplotlib: Vẽ biểu đồ để trực quan hóa kết quả huấn luyện.

2. Đọc và chuẩn bị dữ liệu

```
python
Sao chép mã
df = pd.read_csv('climate.csv', sep=',')
data = df.iloc[:, [2]].values
dataN
```

- `pd.read_csv('climate.csv')`: Đọc dữ liệu từ tệp `climate.csv`.
- `data = df.iloc[:, [2]].values`: Lấy giá trị từ cột thứ ba trong bảng dữ liệu.

3. Hàm chuẩn hóa dữ liệu

```
python
Sao chép mã
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 hóa dữ liệu bằng cách trừ giá trị trung bình và chia cho độ lệch chuẩn của tập huấn luyện.

4. Chia tập dữ liệu huấn luyện và kiểm tra

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

MSSV: B2113343

Họ tên: Phạm Như Thịnh

- `train_split`: Xác định chỉ số để chia dữ liệu thành tập huấn luyện (chiếm 71.5%) và tập kiểm tra (28.5%).
- `normalize(data, train_split)`: Chuẩn hóa dữ liệu.
- `train_data` và `val_data`: Chia dữ liệu thành tập huấn luyện và tập kiểm tra.

5. Xác định các tham số dự báo

```
python
Sao chép mã
past = 720
future = 72
step = 6
batch_size = 256
```

- `past`: Số lượng thời điểm trong quá khứ dùng làm đầu vào cho mô hình (720 thời điểm).
- `future`: Số thời điểm trong tương lai cần dự đoán.
- `step`: Khoảng cách lấy mẫu, dùng để giảm số lượng dữ liệu.
- `batch_size`: Số lượng mẫu trong mỗi lần huấn luyện.

6. Tạo tập dữ liệu huấn luyện và kiểm tra

```
python
Sao chép mã
start = past + future
end = start + train_split
x_train = train_data
y_train = data[start:end]
sequence_length = int(past / step)
dataset_train = timeseries_dataset_from_array(
    x_train,
    y_train,
    sequence_length=sequence_length,
    sampling_rate=step,
    batch_size=batch_size,
)
```

- `timeseries_dataset_from_array`: Tạo các đoạn chuỗi thời gian từ dữ liệu huấn luyện.
- `sequence_length`: Chiều dài của chuỗi thời gian ($720/6 = 120$).

7. Tạo tập dữ liệu kiểm tra

```
python
Sao chép mã
x_end = len(val_data) - past - future
label_start = train_split + past + future
x_val = val_data[:x_end]
y_val = data[label_start:]
dataset_val = timeseries_dataset_from_array(
    x_val,
    y_val,
    sequence_length=sequence_length,
```

MSSV: B2113343

Họ tên: Phạm Như Thịnh

```
sampling_rate=step,  
batch_size=batch_size,  
)
```

- Tạo tập dữ liệu kiểm tra `dataset_val` giống cách tạo `dataset_train`.

8. Xây dựng mô hình LSTM

```
python  
Sao chép mã  
from keras import Input  
learning_rate = 0.5  
model = tf.keras.Sequential()  
model.add(Input(shape=(sequence_length, 1)))  
model.add(LSTM(32))  
model.add(Dense(1))  
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=learning_rate)  
, loss="mse")  
model.summary()
```

- `Sequential` API để xây dựng mô hình.
- `LSTM(32)`: Thêm lớp LSTM với 32 đơn vị.
- `Dense(1)`: Thêm lớp đầu ra.
- `model.compile`: Cấu hình mô hình với Adam optimizer và hàm mất mát MSE.

9. Huấn luyện mô hình

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

- `model.fit`: Huấn luyện mô hình trên tập `dataset_train` với `epochs=10`.

10. Hàm trực quan hóa mất mát huấn luyện

```
python  
Sao chép mã  
def visualize_loss(history, title):  
    loss = history.history["loss"]  
    val_loss = history.history["val_loss"]  
    epochs = range(len(loss))  
    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")
```

MSSV: B2113343

Họ tên: Phạm Như Thịnh

```
plt.ylabel("Loss")
plt.legend()
plt.show()
visualize_loss(history, "Training and Validation Loss")
```

- `visualize_loss`: Vẽ biểu đồ cho mất mát của tập huấn luyện và tập kiểm tra để đánh giá hiệu suất mô hình qua các epoch.