MSSV: B2113343
Họ tên: Phạm Như Thịnh

Thực Hành 8: xây dựng và huấn luyện mô hình LSTM để dự đoán từ tiếp theo trong chuỗi văn bản

1. Import các thư viện

```
python
Sao chép mã
import tensorflow as tf
import numpy as np
import collections
from keras.api.layers import LSTM, Dense
```

- `tensorflow` và `keras`: Sử dụng TensorFlow và các lớp LSTM, Dense từ Keras để xây dựng mô hình LSTM.
- `numpy`: Hỗ trợ các thao tác trên mảng.
- `collections`: Hỗ trợ đếm và tạo từ điển các từ và tần số xuất hiện của chúng trong văn bản.

2. Đọc dữ liệu từ tệp văn bản

```
python
Sao chép mã
def read_data(fname):
    with open(fname) as f:
        content = f.readlines()
        content = [x.strip() for x in content]
        words = []
        for line in content:
            words.extend(line.split())
    return np.array(words)
```

- Hàm `read_data` đọc nội dung của tệp `toto.txt`, sau đó xóa khoảng trắng và chia các dòng thành các từ riêng lẻ. Kết quả trả về là một mảng `words` chứa các từ trong tệp.

3. Tạo từ điển `word2id` và `id2word`

```
python
Sao chép mã
def build_dataset(words):
    count = collections.Counter(words).most_common()
    word2id = {}
    for word, freq in count:
        word2id[word] = len(word2id)
        id2word = dict(zip(word2id.values(), word2id.keys()))
    return word2id, id2word
```

- Hàm `build_dataset` tạo từ điển `word2id` ánh xạ từ thành ID, và từ điển `id2word` ánh xạ ID thành từ. Số ID này được sắp xếp dựa trên tần suất từ xuất hiện.

MSSV: B2113343
Họ tên: Phạm Như Thịnh

4. Đọc dữ liệu và tạo tập từ điển

```
python
Sao chép mã
data = read_data('toto.txt')
print(data)
w2i, i2w = build_dataset(data)
vocab_size = len(w2i)
```

- Gọi hàm `read_data` để đọc tệp và lưu vào `data`.
- Gọi hàm `build_dataset` để tạo từ điển `w2i` (word to id) và `i2w` (id to word).
- `vocab_size`: Số lượng từ trong từ điển.

5. Chuẩn bị dữ liệu huấn luyện

```
python
Sao chép mã
timestep = 3
X, Y = [], []
for i in range(timestep, len(data)):
    X.append([w2i[data[k]] for k in range(i-timestep, i)])
    Y.append(w2i[data[i]])
encoded_data = [w2i[x] for x in data]
X = encoded_data[:-1]
Y = encoded_data[timestep:]
X_training_np = np.array(X)
y_training_np = np.array(Y)
```

- `timestep`: Độ dài chuỗi đầu vào để dự đoán từ tiếp theo.
- `X` và `Y`: Tạo ra dữ liệu đầu vào `X` (chuỗi từ) và đầu ra `Y` (từ cần dự đoán tiếp theo).
- `encoded_data`: Mã hóa toàn bộ dữ liệu thành các ID.
- `X_training_np` và `y_training_np`: Chuyển `X` và `Y` sang dạng mảng numpy.

6. Tạo tập dữ liệu chuỗi thời gian

```
python
Sao chép mã
train_data = tf.keras.preprocessing.timeseries_dataset_from_array(
    X_training_np, y_training_np, sequence_length=timestep, sampling_rate=1
)
```

- `timeseries_dataset_from_array` giúp chia dữ liệu thành chuỗi với độ dài cố định là `timestep`.

7. Xây dựng mô hình LSTM

```
python
Sao chép mã
model = tf.keras.Sequential()
model.add(LSTM(512, return_sequences=True, input_shape=(timestep, 1)))
model.add(LSTM(512, return_sequences=False))
```

```
model.add(Dense(vocab_size))
```

- Mô hình `Sequential` LSTM với 2 lớp LSTM có 512 đơn vị và một lớp Dense đầu ra.
- `Dense(vocab_size)`: Số lượng đầu ra bằng với kích thước từ điển (`vocab_size`), mỗi đầu ra dự đoán một từ trong từ điển.

8. Biên dịch và huấn luyện mô hình

```
python
Sao chép mã
model.compile(optimizer='adam',

loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
model.fit(train_data, epochs=500)
```

- `optimizer='adam'`: Bộ tối ưu hóa Adam.
- `SparseCategoricalCrossentropy`: Hàm mất mát phù hợp cho bài toán phân loại đa lớp.
- `epochs=500`: Huấn luyện mô hình trong 500 lần lặp.

9. Hàm mã hóa câu

```
python
Sao chép mã
def encode(sent):
    encoded_sent = [w2i[w] for w in sent.split()]
    encoded_sent = np.array(encoded_sent).reshape(1, timestep, 1)
    return encoded_sent
```

- Hàm `encode` mã hóa chuỗi `sent` đầu vào thành dạng số và định dạng lại để phù hợp làm đầu vào cho mô hình.

10. Dự đoán từ tiếp theo

```
python
Sao chép mã
pred = model.predict(encode("had a general"))
pred_word = i2w[np.argmax(pred)]
print(pred_word)
```

- `model.predict`: Dự đoán từ tiếp theo cho chuỗi "had a general".
- `np.argmax(pred)`: Lấy chỉ số của từ có xác suất cao nhất.
- `pred_word`: Tra từ dự đoán trong từ điển `i2w`.

11. Thực hiện dự đoán với một chuỗi khác

```
python
Sao chép mã
```

MSSV: B2113343
Họ tên: Phạm Như Thịnh

```
pred = model.predict(encode("a general council"))
pred_word = i2w[np.argmax(pred)]
print(pred_word)
```

- Dự đoán từ tiếp theo cho chuỗi "a general council" và in kết quả.