BÀI TẬP 5

Due date: 26/09/2023

Xử lý ngôn ngữ tự nhiên
[3] Chollet, Chapter 6: Deep learning for text and sequences

5.1. Hiển thị quá trình học với keras

   a. Trình bày hiểu biết của bạn vê hiển thị quá trình học với keras
   https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/
   b. Chạy ví dụ sau và giải thích

```
=====================================================
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import matplotlib.pyplot as plt
import numpy as np
# load pima indians dataset
dataset = np.loadtxt("pima-indians-diabetes.csv", delimiter=",")
# split into input (X) and output (Y) variables
X = dataset[:,0:8]
Y = dataset[:,8]
# create model
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# Compile model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# Fit the model
history = model.fit(X, Y, validation_split=0.33, epochs=150, batch_size=10, verbose=0)
# list all data in history
print(history.history.keys())
# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

==

5.2. Xử lý ngôn ngữ tự nhiên

a. Trinh bày hiểu biết về xử lý ngôn ngữ tự nhiên
[3] Chollet, Chapter 6: Deep learning for text and sequences
https://machinelearningmastery.com/what-are-word-embeddings/
https://www.geeksforgeeks.org/word-embeddings-in-nlp/
https://www.turing.com/kb/guide-on-word-embeddings-in-nlp

b. Chạy ví dụ sau và giải thích

===================================

```
from keras.datasets import imdb
from keras import preprocessing
from keras.layers import Embedding
#from tensorflow.keras.models import Sequential
#from tensorflow.keras.layers import Dense
from keras.models import Sequential
from keras.layers import Flatten, Dense

import matplotlib.pyplot as plt
import numpy as np

max_features = 10000
maxlen = 20
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
x_train = preprocessing.sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = preprocessing.sequence.pad_sequences(x_test, maxlen=maxlen)

model = Sequential()
model.add(Embedding(10000, 8, input_length=maxlen))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
model.summary()
history = model.fit(x_train, y_train,
            epochs=10,
            batch_size=32,validation_split=0.2)

plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```