

NHẬP MÔN HỌC SÂU THỰC HÀNH

MỤC LỤC

Phần 1: CÀI ĐẶT NGÔN NGỮ LẬP TRÌNH VÀ THƯ VIỆN	1
1. Ngôn ngữ lập trình Python	1
2. Cài đặt các thư viện cần cho học sâu dựa trên Python	1
3. Công cụ lập trình cho Python (Integrated Development Environment – IDE)....	1
Phần 2: NỘI DUNG THỰC HÀNH	3
1. Mạng tích chập ConvNet.....	3
2. LSTM	4

Phần 1: CÀI ĐẶT NGÔN NGỮ LẬP TRÌNH VÀ THƯ VIỆN

1. Ngôn ngữ lập trình Python

Python là ngôn ngữ lập trình bậc cao đa năng. Triết lý thiết kế của nó nhấn mạnh khả năng đọc mã bằng cách sử dụng thụt lề đáng kể.

Python có kiểu động và thu gom rác. Ngôn ngữ này hỗ trợ nhiều mô hình lập trình, bao gồm lập trình cấu trúc (đặc biệt là lập trình thủ tục), lập trình hướng đối tượng và lập trình chức năng.

Python 2.0 được ra mắt vào năm 2000. Python 3.0 được ra mắt vào năm 2008, là bản sửa đổi lớn không hoàn toàn tương thích ngược với các phiên bản trước đó. Python 2.7.18, được phát hành vào năm 2020, là bản phát hành cuối cùng của Python 2.

Python liên tục được xếp hạng là một trong những ngôn ngữ lập trình phổ biến nhất và được sử dụng rộng rãi trong cộng đồng học máy.

2. Cài đặt các thư viện cần cho học sâu dựa trên Python

Cài đặt các thư viện sau để chạy được các bài tập thực hành:

- Keras
- TensorFlow
- Scikit Learn
- Pandas
- Pickle
- Matplotlib

3. Công cụ lập trình cho Python (Integrated Development Environment – IDE)

Có nhiều công cụ để lập trình ngôn ngữ Python như Pycharm, Spyder, Visual Studio Code, Jupyter Notebook, hoặc dịch vụ dựa trên web của Google là Google Colab.

3.1. Pycharm

Pycharm là IDE Python có số lượng người dùng đông đảo nhất hiện nay, được phát hành lần đầu ra công chúng vào ngày 3 tháng 2 năm 2010 bởi JetBrains.

IDE này được phát hành đa nền tảng trên Windows, MacOS và Linux.

Pycharm có một giao diện tối giản, trực quan và hỗ trợ mạnh mẽ mọi thứ cần để lập trình Python phù hợp với nhiều đối tượng người lập trình.

Hiện nay nhà phát hành có sẵn 3 phiên bản Pycharm bao gồm:

- Phiên bản Community được cấp phép từ Apache (100% miễn phí)
- Phiên bản Education (Miễn phí đối với người dùng có email do nhà trường đăng ký)
- Phiên bản Professional (Có phí) với phiên bản này có nhiều tính năng nâng cao như quản trị cơ sở dữ liệu, Framework phổ biến tích hợp như: Django, Flask, Google App, Engine, Pyramid

3.2. Visual Studio Code

Visual Studio Code (VS Code) là một Code Editor mạnh mẽ có khả năng mở rộng với các plugin phong phú để trở thành một IDE thực thụ, được phát triển đa nền tảng bởi Microsoft. Phần mềm này được xuất bản lần đầu vào ngày 29 tháng 4 năm 2015

VS Code với các plugin phong phú có thể hỗ trợ hầu hết loại ngôn ngữ lập trình hiện có trên thị trường, trong đó bao gồm cả Python. Phần mềm lập trình này cũng hỗ trợ các thuật toán tự động hoàn thành mã thông minh (IntelliSense), Bugger và terminal để khởi chạy các lệnh trên server.

3.3. Spyder

Spyder là một IDE Python hỗ trợ nhiều chức năng cho khoa học dữ liệu được phát hành lần đầu tiên vào ngày 18 tháng 10 năm 2019. Đây là một trong những phần mềm lập trình Python chuyên cho data science có chứng chỉ MIT.

3.4. Gom cụm

Google Colab là một dịch vụ miễn phí từ Google cho phép viết và chia sẻ mã Python. Người dùng có thể lưu trữ và chạy Notebook trên đám mây thông qua trình duyệt mà không cần kích hoạt cấu hình bộ đệm phức tạp cho máy tính cá nhân.

Phần 2: NỘI DUNG THỰC HÀNH

1. Mạng tích chập ConvNet

Xây dựng mạng tích chập và thử nghiệm trên bộ dữ liệu MNIST.

a) Khai báo thư viện sử dụng:

```
from keras import layers
from keras import models
from keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
```

b) Xây dựng mô hình:

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
                        input_shape=(28, 28, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

c) Chuẩn bị dữ liệu

```
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images = train_images.reshape((60000, 28, 28, 1))
train_images = train_images.astype('float32') / 255
test_images = test_images.reshape((10000, 28, 28, 1))
test_images = test_images.astype('float32') / 255
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

d) Train và test

```
model.fit(train_images, train_labels, epochs=3, batch_size=64)

test_loss, test_acc = model.evaluate(test_images, test_labels)
print(test_acc)
```

Bài tập thực hành.

- Bổ sung thêm các lệnh để vẽ ra các giá trị accuracy và loss của quá trình train và validation khi cho epoch = 20. Trình bày kết quả thu được vào báo cáo.
- Thay đổi cấu trúc mô hình và các tham số để thu được kết quả tốt hơn chương trình mẫu. Trình bày chương trình và kết quả vào báo cáo.

2. LSTM

Xây dựng mạng tích chập và thử nghiệm trên bộ dữ liệu IMDB

a) Khai báo thư viện sử dụng

```
from keras.datasets import imdb
from keras.layers import LSTM
from keras.models import Sequential
from keras.preprocessing import sequence
from keras.layers import Embedding, Dense
import matplotlib.pyplot as plt
from keras.layers import Dropout
```

b) Khai báo biến và xây dựng mô hình

```
max_features = 10000
maxlen = 500
batch_size = 256
model = Sequential()
(input_train, y_train), (input_test, y_test) = imdb.load_data(
    num_words=max_features)
input_train = sequence.pad_sequences(input_train, maxlen=maxlen)
input_test = sequence.pad_sequences(input_test, maxlen=maxlen)
model.add(Embedding(max_features, 32))
model.add(LSTM(32))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
```

c) Train và test

```
model.compile(optimizer='rmsprop',  
              loss='binary_crossentropy',  
              metrics=['acc'])  
  
history = model.fit(input_train, y_train,  
                    epochs=10,  
                    batch_size=batch_size,  
                    validation_split=0.2)  
  
test = model.evaluate(input_test, y_test, verbose=1)
```

d) Đọc accuracy và loss

```
test_acc = test[1]  
test_loss = test[0]  
print('Loss: ')  
print(test_loss)  
print('Accuracy: ')  
print(test_acc)
```

Bài tập thực hành.

- a) Bổ sung thêm các lệnh để vẽ ra các giá trị accuracy và loss của quá trình train và validation khi cho epoch = 20. Trình bày kết quả thu được vào báo cáo.
- b) Thay đổi cấu trúc mô hình và các tham số để thu được kết quả tốt hơn chương trình mẫu. Trình bày chương trình và kết quả vào báo cáo.