|  |  |  |
| --- | --- | --- |
|  | BAN CƠ YẾU CHÍNH PHỦ  “HỌC VIỆN KỸ THUẬT MẬT MÃ” | Mẫu 2 |

BÁO CÁO CHUYÊN ĐỀ SỐ 4.5.5

“Áp dụng mô hình học sâu DNN”

NHIỆM VỤ: “Nghiên cứu và ứng dụng nền tảng học sâu để xây dựng hệ thống phát hiện mã độc trực tuyến”.

Mã số: 06/2022/CB.

Cơ quan chủ trì: Học viện Kỹ thuật Mật mã

Chủ nhiệm: ThS. Lê Đức Thuận

Hà Nội - 2023

|  |  |  |
| --- | --- | --- |
|  | BAN CƠ YẾU CHÍNH PHỦ  “HỌC VIỆN KỸ THUẬT MẬT MÃ” |  |

BÁO CÁO CHUYÊN ĐỀ SỐ 4.5.5

“Áp dụng mô hình học sâu DNN”

NHIỆM VỤ: “Nghiên cứu và ứng dụng nền tảng học sâu để xây dựng hệ thống phát hiện mã độc trực tuyến”.

Mã số: 06/2022/CB.

Cơ quan chủ trì: Học viện Kỹ thuật Mật mã

Chủ nhiệm: ThS. Lê Đức Thuận

|  |  |
| --- | --- |
| **Người thực hiện chuyên đề** | **Cơ quan chủ trì** |
| *(Họ tên và chữ ký)* | *(Họ tên và chữ ký)* |

Hà Nội - 2023

MỤC LỤC

[MỤC LỤC 1](#_Toc129729600)

[DANH MỤC HÌNH ẢNH 2](#_Toc129729601)

[Deep Neural Networks 3](#_Toc129729602)

[1.1. Neural & Neural Network 3](#_Toc129729603)

[1.2. Deep Neural Network & Deep Learning 4](#_Toc129729604)

[1.3 Ứng dụng của Deep Learning & Deep Neural Network 5](#_Toc129729605)

[1.4 Thực nghiệm 6](#_Toc129729606)

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Mô phỏng nơron thần kinh con người trên mô hình toán học 3](#_Toc129729594)

[Hình 1.2 Sơ đồ mô phỏng cấu trúc một neural network đơn 4](#_Toc129729595)

[Hình 1.3 Cấu trúc của Deep Neural Network 4](#_Toc129729596)

[Hình 1.4 Sơ đồ mô phỏng Neural Network & Deep Learning Neural Network 5](#_Toc129729597)

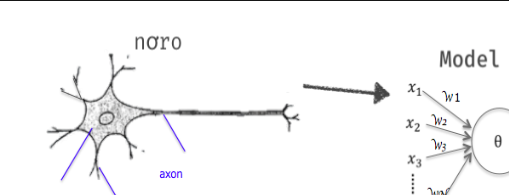
[Hình 1.5: Thông tin mô hình trong quá trình huấn luyện 10](#_Toc129729598)

[Hình 1.6 Biểu đồ thông số của mô hình sau khi huấn huyện 11](#_Toc129729599)

Deep Neural Networks

* 1. Neural & Neural Network

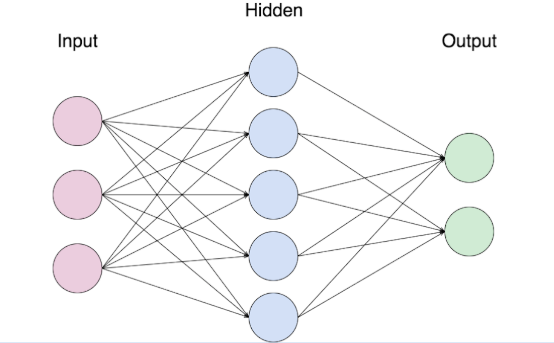
*Neural* là mô hình toán học mô phỏng và biểu hiện cho một số chức năng của nơron thần kinh trong não bộ con người.



Hình 1.1 Mô phỏng nơron thần kinh con người trên mô hình toán học

*Neural Network*: là tập hợp của nhiều Neural, các tập hợp này tạo thành một cấu trúc hệ thống gồm các lớp mà ta còn gọi là các layer. Trong đó, ta sẽ thấy một Neural Network đơn (Simple Neural Network) hay một đơn vị Neural Network thường có các lớp: Input layer (lớp đầu vào), Hidden layer (lớp ẩn) và Output layer (lớp đầu ra).

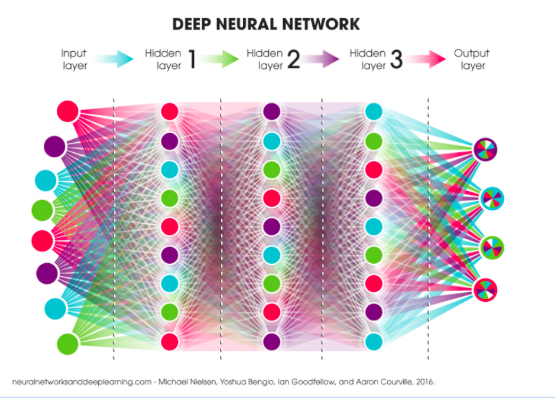
Các lớp này có nhiệm vụ xử lý tín hiệu nhận được theo thứ tự lớp sau nhận giá trị output của lớp trước để tiến hành. Việc các lớp này xử lý theo cách nào thường phụ thuộc vào từng yêu cầu khác nhau. Số lượng các lớp ẩn (hidden layer) là không giới hạn. Số lớp ẩn và cách xử lý ở từng lớp kể trên sẽ quyết định kết quả và hiệu quả của công việc cần xử lý.



Hình 1.2 Sơ đồ mô phỏng cấu trúc một neural network đơn

* 1. Deep Neural Network & Deep Learning

Để hiểu về *Deep Learning*, trước tiên ta cần tìm hiểu về *Deep Neural Network*.



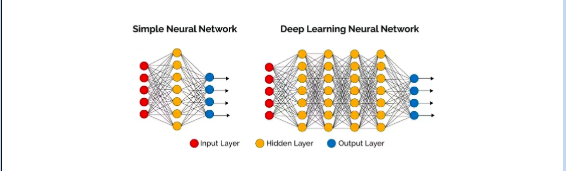
Hình 1.3 Cấu trúc của Deep Neural Network

***Deep neural network*** là hệ thống cấu trúc thần kinh phức tạp gồm nhiều đơn vị neural network mà trong đó, ngoài các lớp nguồn vào (input), nguồn ra (output) thì có hơn một lớp ẩn (hidden layer). Mỗi lớp này sẽ thực hiện một kiểu phân loại và sắp xếp riêng trong một quá trình ta gọi là “phân cấp tính năng” và mỗi lớp đảm nhiệm một trọng trách riêng, output của lớp này sẽ là input của lớp sau.

*Deep Neural Network* được xây dựng với mục đích mô phỏng hoạt động não bộ phức tạp của con người và được áp dụng vào nhiều lĩnh vực khác nhau, mang lại thành công và những hiệu quả đáng kinh ngạc cho con người.

***Deep Learning*** là khả năng máy tự đào sâu học hỏi nhờ vào công nghệ Deep Neural Network để từ đó xử lý và giải quyết những dữ liệu phi định danh hay phi cấu trúc.

*Deep Learning* là một phần của phương pháp Machine Learning (Học Máy), đại diện cho một hình thức cụ thể của phương pháp này, là nơi mà các loại hình công nghệ sẽ dùng mọi mặt của lĩnh vực trí tuệ nhân tạo nhằm tìm kiếm và phân loại cũng như sắp xếp thông tin theo nhiều cách vượt xa các giao thức nguồn vào và nguồn ra ban đầu. Ta có thể hình dung Deep Learning với quá trình mô phỏng hoạt động não bộ của chúng ta thông qua các mô hình toán học.



Hình 1.4 Sơ đồ mô phỏng Neural Network & Deep Learning Neural Network

1.3 Ứng dụng của Deep Learning & Deep Neural Network

*Deep Neural Network* được ứng dụng trong lĩnh vực *Deep Learning* để xử lý và giải quyết vấn đề. *Deep Learning* được ứng dụng trong thực tế cuộc sống, với một số ví dụ như:

* *Dịch thuật*: Phương pháp học sâu (*deep learning algorithms*) sẽ dịch các ngôn ngữ, giúp khách du lịch hay những người có nhu cầu về công việc có thể dễ dàng trao đổi, tương tác với nhau mà không cần phải chật vật như xưa.
* *Phương tiện không người lái*: Nghe có vẻ viễn tưởng nhưng trên thực tế những chiếc máy bay không người lái và những xe hơi tự hành giờ đây không còn xa lạ với đời sống của chúng ta. Nhờ *deep learning*, sau khi các thuật toán tiếp nhận và xử lý các bộ dữ liệu sẽ có khả năng hành động giống như con người, “nhìn” được thực tế đường đi, di chuyển, dừng lại hay tránh các xe khác.
* *Trợ lý ảo & các dịch vụ chatbots/bots*: Các nhà cung cấp dịch vụ trực tuyến giờ đây thường dùng các trợ lý ảo hay các dịch vụ chatbots, bots để chăm sóc khách hàng. Việc này không chỉ mang lại hiệu quả về chi phí mà còn nâng cao được năng suất nhờ vào số lượng câu hỏi được giải đáp cũng như tốc độ phản hồi tăng lên nhanh chóng.
* *Y học*: Thông qua phương pháp deep learning và công nghệ deep neural network, bệnh nhân giờ đây được chẩn đoán, kê đơn một cách nhanh chóng và chính xác hơn. Các công ty, tổ chức trong lĩnh vực y tế ngày càng đầu tư nhiều vào loại hình công nghệ này nhằm giảm tải cho các bác sĩ và giải phóng áp lực cho bệnh viện hay các cơ sở khám chữa.
* *Mua sắm & Giải trí*: Đây chính là kết quả có được nhờ Deep Learning và công nghệ Deep Neural Network mà ta đang đề cập.

1.4 Thực nghiệm

Dựa trên các đặc điểm trên, chúng ta sẽ áp dụng mô hình học máy DNN để huấn luyện cũng như chạy thực nghiệm việc phân loại mã độc với dữ liệu đầu vào là các đặc trưng. Để có thể sử dụng được thư viện, ta tiến hành cài đặt các thư viện cần thiết thông qua câu lệnh *pip install pandas tensorflow scikit-learn matplotlib gspread gspread-pandas gspread-dataframe*.

Sau khi cài đặt xong, ta tiến hành nạp các thư viện cần thiết vào chương trình.

import keras

import numpy as np

import pandas as pd

from keras.layers import Input, Dense, Flatten, Dropout

from keras.models import Model

from keras.layers.convolutional import Conv2D

from keras.layers.pooling import MaxPooling2D

from keras.layers.merge import concatenate

from keras.utils import to\_categorical

from keras.callbacks import TensorBoard, EarlyStopping, ModelCheckpoint

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, classification\_report, confusion\_matrix

import matplotlib.pyplot as plt

from google.colab import auth

auth.authenticate\_user()

import gspread

import gspread\_dataframe

from oauth2client.client import GoogleCredentials

from keras.layers.normalization import BatchNormalization

Tiếp theo, ta tiến hành nạp dữ liệu sử dụng để huấn luyện và kiểm tra từ file có định dạng .csv vào chương trình thông qua thư viện pandas đồng thời biến đổi dữ liệu đọc được sang dạng ma trận thông qua sử dụng thư viện numpy.

files = []

for i in range(0, 10):

path = r”dataset/file-${i}.csv”

data = pd.read\_csv(path, header=None, skiprows=1)

data = np.array(data)

files.append(data)

Tiếp theo, khởi tạo các ma trận chứa dữ liệu được sử dụng để huấn luyện mô hình bằng cách chia dữ liệu đã nạp thành 10 phần, lấy 1 phần làm dữ liệu kiểm tra, 2 phần dữ liệu xác thực và còn lại làm dữ liệu để huấn luyện.

index = 1

train = np.concatenate([files[i%10] for i in range(index + 2, index + 10)], axis=0)

test = files[(index+1)%10]

val = files[index]

x\_train = train[:, 3:].astype(np.float32)

y\_train = train[:, 2].astype('int')

del train

x\_val = val[:, 3:].astype(np.float32)

y\_val = val[:, 2].astype('int')

del val

x\_test = test[:, 3:].astype(np.float32)

y\_test = test[:, 2].astype('int')

del test

Sau khi đã chuẩn bị xong các thông số cũng như dữ liệu huấn luyện cho mô hình, ta tiến hành xây dựng mô hình để huấn luyện. Đầu tiên, ta khởi tạo một mô hình DNN.

input = Input(shape=(x\_val.shape[1],))

batch1 = BatchNormalization()(input)

hidden1 = Dense(16300, activation='relu')(batch1)

dropout1 = Dropout(0.5)(hidden1)

batch2 = BatchNormalization()(dropout1)

hidden2 = Dense(14600, activation='relu')(batch2)

dropout2 = Dropout(0.5)(hidden2)

batch3 = BatchNormalization()(dropout2)

hidden3 = Dense(12900, activation='relu')(batch3)

dropout3 = Dropout(0.5)(hidden3)

batch4 = BatchNormalization()(dropout3)

hidden4 = Dense(11200, activation='relu')(batch4)

dropout4 = Dropout(0.5)(hidden4)

batch5 = BatchNormalization()(dropout4)

hidden5 = Dense(9500, activation='relu')(batch5)

dropout5 = Dropout(0.5)(hidden5)

batch6 = BatchNormalization()(dropout5)

hidden6 = Dense(7800, activation='relu')(batch6)

dropout6 = Dropout(0.5)(hidden6)

batch7 = BatchNormalization()(dropout6)

hidden7 = Dense(6100, activation='relu')(batch7)

dropout7 = Dropout(0.5)(hidden7)

batch8 = BatchNormalization()(dropout7)

hidden8 = Dense(4400, activation='relu')(batch8)

dropout8 = Dropout(0.5)(hidden8)

batch9 = BatchNormalization()(dropout8)

hidden9 = Dense(2700, activation='relu')(batch9)

dropout9 = Dropout(0.5)(hidden9)

batch10 = BatchNormalization()(dropout9)

hidden10 = Dense(1000, activation='relu')(batch10)

dropout10 = Dropout(0.5)(hidden10)

batch = BatchNormalization()(dropout10)

output = Dense(228, activation='softmax')(batch)

model = Model(inputs=input, outputs=output)

Ở đây, ta sử dụng 10 lớp Dense được tạo bởi lớp BatchNormalization với kích thước đầu vào lần lượt là 16300, 14600, 12900, 11200, 9500, 7800, 6100, 4400, 2700, và 1000. Mỗi lớp đều sử dụng thuật toán là *relu* được truyền vào dưới dạng tham số activation và kết hợp cùng với 1 lớp Dropout. Lớp cuối cùng ta sử dụng một lớp Dense được tạo từ lớp BatchNormalization với số lượng dữ liệu đầu vào là 228 và thuật toán được dùng là softmax nhằm lấy lượng kết quả có xác suất cao nhất từ trên xuống.

Kế tiếp, ta bắt đầu tiến hành huấn luyện mô hình.

model.compile(

loss='categorical\_crossentropy',

optimizer='adam',

metrics=['accuracy']

)

history = model.fit(

x\_train,

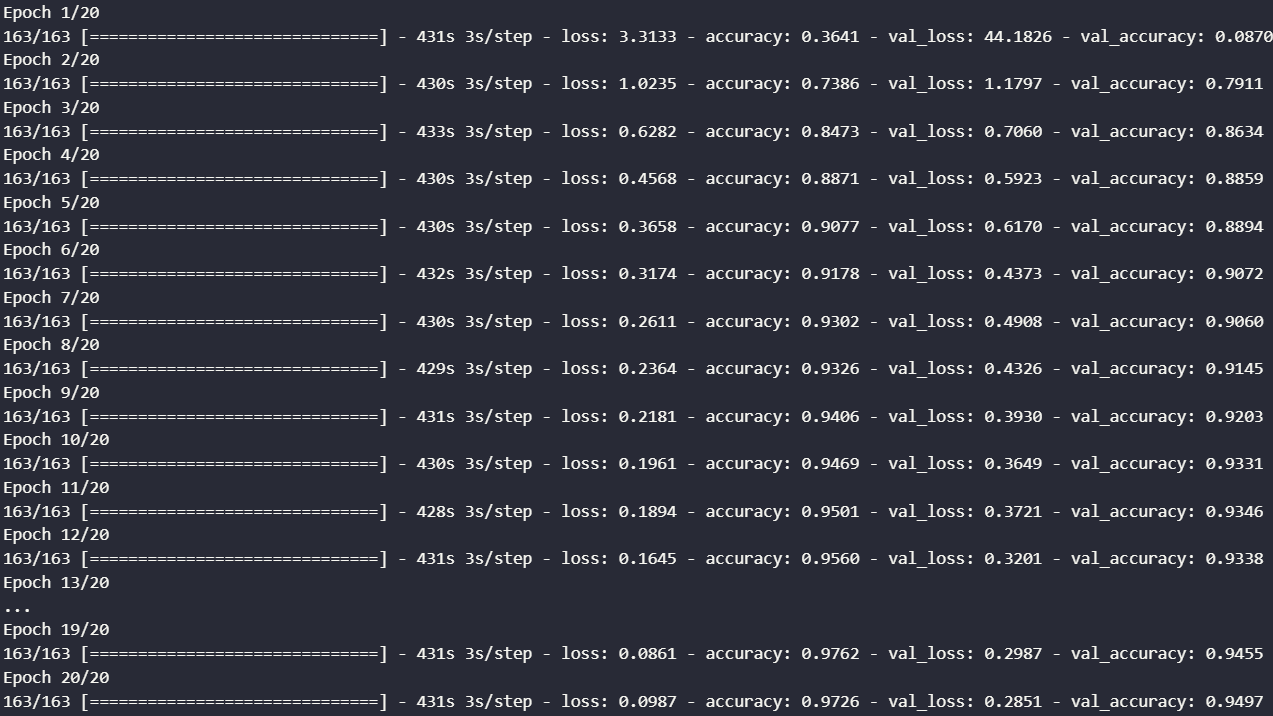
y\_train,

epochs=20,

batch\_size = 128,

validation\_data=(x\_val, y\_val)

)



Hình 1.5: Thông tin mô hình trong quá trình huấn luyện

Sau khi huấn luyện mô hình hoàn tất, chúng ta có thể đánh giá hiệu năng phân loại của mô hình thông qua việc sử dụng hàm *classification\_report* của thư viện sklearn*.*

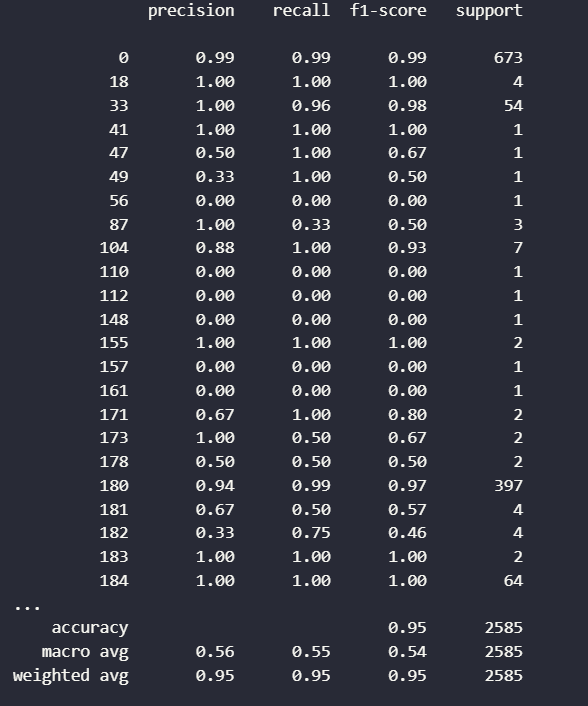
y\_pred\_DNN = model.predict(x\_test)

y\_pred = np.argmax(y\_pred\_DNN, axis=-1)

# report = classification\_report(list(y\_test), list(y\_pred), output\_dict=True)

report = classification\_report(list(y\_test), list(y\_pred))

print(report)



Hình 1.6 Biểu đồ thông số của mô hình sau khi huấn huyện