

MỘT PHƯƠNG PHÁP TƯƠNG TÁC VỚI DỊCH VỤ ĐIỆN TOÁN Đám Mây AWS TỪ RSTUDIO A METHOD OF INTERACTING WITH THE AWS CLOUD SERVICE FROM RSTUDIO

NGUYỄN VĂN TÂM

Học viên cao học Khoa CNTT, Trường Đại học Sư phạm Kỹ thuật Vĩnh Long

Email: tamvn135@gmail.com

Nhận bài (Received): 25/9/2022; Phản biện (Reviewed): 28/10/2022; Chấp nhận (Accepted): 26/11/2022

TÓM TẮT

Học máy và học sâu là lĩnh vực nghiên cứu hiện nay được rất nhiều nhà khoa học quan tâm. Tuy nhiên chúng ta phải thừa nhận rằng thực tế là sẽ không bao giờ có đủ tài nguyên CPU hoặc RAM cho các phân tích của mình. Việc sử dụng điện toán đám mây là lựa chọn rất hợp lý trong thời điểm hiện nay. Tuy nhiên thao tác trên đám mây và chạy các mô hình của bạn chưa thật sự dễ dàng đối với những người mới tìm hiểu, nên nếu bạn có thể khởi chạy và tương tác với dịch vụ đám mây ngay trên Rstudio thì thật tiện lợi. Xuất phát từ suy nghĩ này tôi giới thiệu một phương pháp tương tác với đám mây từ Rstudio. Trong bài viết này, tôi tập trung vào việc tiếp cận thực hành tạo ra một cách riêng để thực hành học sâu mạng nơ-ron nhân tạo trên đám mây AWS. Kết quả của nghiên cứu này mở ra một hướng tiếp cận với lập trình học sâu với AWS, cung cấp thêm một sự lựa chọn cho những người muốn tham gia lĩnh vực này.

Từ khóa: Điện toán đám mây, học máy, học sâu.

ABSTRACT

Machine learning and deep learning are the current research areas of great interest to many scientists. However, we must admit that the reality is that there will never be enough CPU or RAM resources for our analysis. The use of cloud computing is a very reasonable choice in the present time. However, manipulating the cloud and running your models is not really easy for novices, so it will be convenient if you can launch and interact with the cloud service right on Rstudio. With this in mind, a method of interacting with the cloud from Rstudio is recommended. In this article, I focus on taking a hands-on approach creating a unique way to practice deep learning of artificial neural networks in the AWS cloud. The results of this study open up an approach to deep learning programming with AWS, providing an additional choice for those looking to enter the field.

Keywords: Cloud computing, machine learning, deep learning.

1. GIỚI THIỆU

Trong mạng nơ-ron, mạng nơ-ron tích chập (ConvNets hay CNNs, Convolutional

Neural Network) là một trong những phương pháp chính để thực hiện nhận dạng hình ảnh, phân loại hình ảnh. CNN

được sử dụng rộng rãi trong một số lĩnh vực như phát hiện đối tượng, nhận dạng khuôn mặt [1], Mạng nơ-ron nhân tạo (Neural Network - NN) là một mô hình lập trình mô phỏng cách thức hoạt động của mạng nơ-ron thần kinh [2]. Kết hợp với các kỹ thuật học sâu (Deep Learning - DL), mạng nơ-ron nhân tạo đang trở thành một công cụ rất mạnh mẽ mang lại hiệu quả tốt nhất cho nhiều bài toán khó như nhận dạng ảnh, giọng nói hay xử lý ngôn ngữ tự nhiên [3][4]. Điều đó cũng kích thích sự tìm hiểu những người nghiên cứu mới và kể cả những nhà nghiên cứu chuyên nghiệp trong nhiều lĩnh vực khác. Có rất nhiều nghiên cứu trước đây trong việc thực hiện phân lớp ảnh với mạng nơ-ron tích chập nhưng chủ yếu xoay quanh tìm hiểu mạng nơ-ron, các kỹ thuật phân tích, nghiên cứu các ứng dụng thực tế từ mạng nơ-ron nhân tạo như: Conceptual understanding of convolutional neural network-a deep learning approach [5], Deep Learning Approach for Automatic Classification of X-Ray Images using Convolutional Neural Network [6].v.v...

Tuy nhiên chưa có nghiên cứu nào giới thiệu đầy đủ về việc chạy mô hình mạng nơ-ron nhân tạo trên nền điện toán đám mây trực tiếp từ R. Vấn đề đặt ra là làm thế nào để tương tác trực tiếp đến cấu trúc của điện toán đám mây từ R, sử dụng dịch vụ lưu trữ trên đám mây để chứa dữ liệu chạy cho các mô hình, chạy thực nghiệm một số mô hình mạng nơ-ron để biểu diễn kết quả thực nghiệm đó.

Trong bài viết này tôi trình bày một hướng thực hành học sâu mạng nơ-ron tích chập trên nền điện toán đám mây. Cách tiếp cận này được thực hiện trên cơ sở nghiên cứu tài liệu chọn phương pháp đơn giản nhất để tiến hành thực nghiệm.

Khi sử dụng Rstudio kết nối với máy chủ đám mây điều này giúp chủ động hơn, bằng cách chỉ tương tác với Rstudio mà vẫn đạt được các mục đích khởi động máy chủ chạy các thuật toán hay các mô hình mà không cần phải đăng nhập vào dịch vụ đám mây như cách mà người dùng đám mây thường làm. Nhật ký thể hiện trực tiếp trên phiên làm việc của Rstudio hoặc có thể xem lại nhật ký bằng cách đăng nhập lên đám mây và vào server Rstudio để xem.

Phần tiếp theo của bài viết được bố cục như sau: Trong Mục 2, trình bày những tìm hiểu về điện toán đám mây AWS, kỹ thuật kết nối R với AWS. Mục 3, nói về CNN tổng thể, giới thiệu 3 mô hình cơ bản của CNN. Mục 4, trình bày thực nghiệm chạy phân loại ảnh trên đám mây AWS. Cuối cùng là kết luận ở Phần 3.

2. NỘI DUNG

2.1. Đám mây AWS

AWS (Amazon Web Services) là nền tảng dịch vụ điện toán đám mây mang đến khả năng tính toán, lưu trữ cơ sở dữ liệu, phân phối nội dung và các chức năng khác [7]. Trong bài viết này chúng ta sử dụng AWS để tạo ra một server Rstudio và dịch vụ lưu trữ S3 để phục vụ cho việc chạy thực nghiệm của chúng ta ở phần sau.

2.2. Thao tác với đám mây AWS truyền thống

Qua nghiên cứu về việc sử dụng các dịch vụ AWS cho thấy với cách thao tác này toàn bộ đều được thực hiện trên giao diện web của nhà cung cấp. Từ việc đăng ký tài khoản sử dụng dịch vụ, tạo ra một máy chủ, các sổ tay ghi chép cho đến thực hiện việc đẩy dữ liệu lên các máy chủ ảo để chạy các mô hình máy học. Chúng ta đều phải thực hiện qua nhiều công đoạn, duyệt

qua các cửa sổ giao diện thực hiện các tùy chọn có sẵn.

Tóm lại nhà cung cấp đã đạt được nhiều tiến bộ và thành công lớn trong việc đem dịch vụ của mình đến với người dùng một cách đơn giản, dễ sử dụng nhất, mặc dù vậy vẫn còn nhiều khâu, nhiều bước cầu hình để sử dụng dịch vụ.

2.3. Kỹ thuật tương tác với AWS từ Rstudio

Để thực hiện tương tác với dịch vụ đám mây AWS từ R trong nghiên cứu cần tạo một tài khoản AWS, khi đã đăng ký thành công, đăng nhập vào tài khoản AWS của mình và vào bảng điều khiển quản lý IAM để tạo ra cặp khóa dùng để truy cập lên dịch vụ đám mây AWS từ R. Lưu ý rằng chỉ có thể thấy thông tin cặp khóa của mình vừa tạo chỉ một lần để lưu nó lại vào một nơi an toàn với chế độ bí mật [8].

Bây giờ trong nghiên cứu này đã có thể thực hiện tương tác từ R đến dịch vụ đám mây AWS. Từ đây chúng ta có thể thực hiện các thao tác như: tạo máy chủ, tạo thùng chứa dữ liệu, đẩy dữ liệu và chạy các mô hình máy học thông qua các đoạn mã.

Chúng ta thực hiện đăng nhập thông qua các biến môi trường chứa thông tin đăng nhập.

```
library(aws.ec2)

Sys.setenv(
  "AWS_ACCESS_KEY_ID" =
    "mykey",
  "AWS_SECRET_ACCESS_
    KEY" = "mysecretkey",
  "AWS_DEFAULT_REGION" =
    "us-west-1"
)
```

Với cách thao tác này chúng ta có thể thực hiện tại máy tính chúng ta một cách dễ dàng và chủ động.

2.4. Tổ chức dữ liệu trên AWS

Trong nghiên cứu này sử dụng dịch vụ lưu trữ đơn giản S3 của điện toán đám mây AWS để chứa dữ liệu cho các mô hình học sâu trong bài viết này. Tạo một thùng chứa cho dữ liệu với tên mặc định là `sagemaker-<aws-region-name>-<aws account number>`.

```
session <- sagemaker$Session()

bucket <- session$default_bucket()

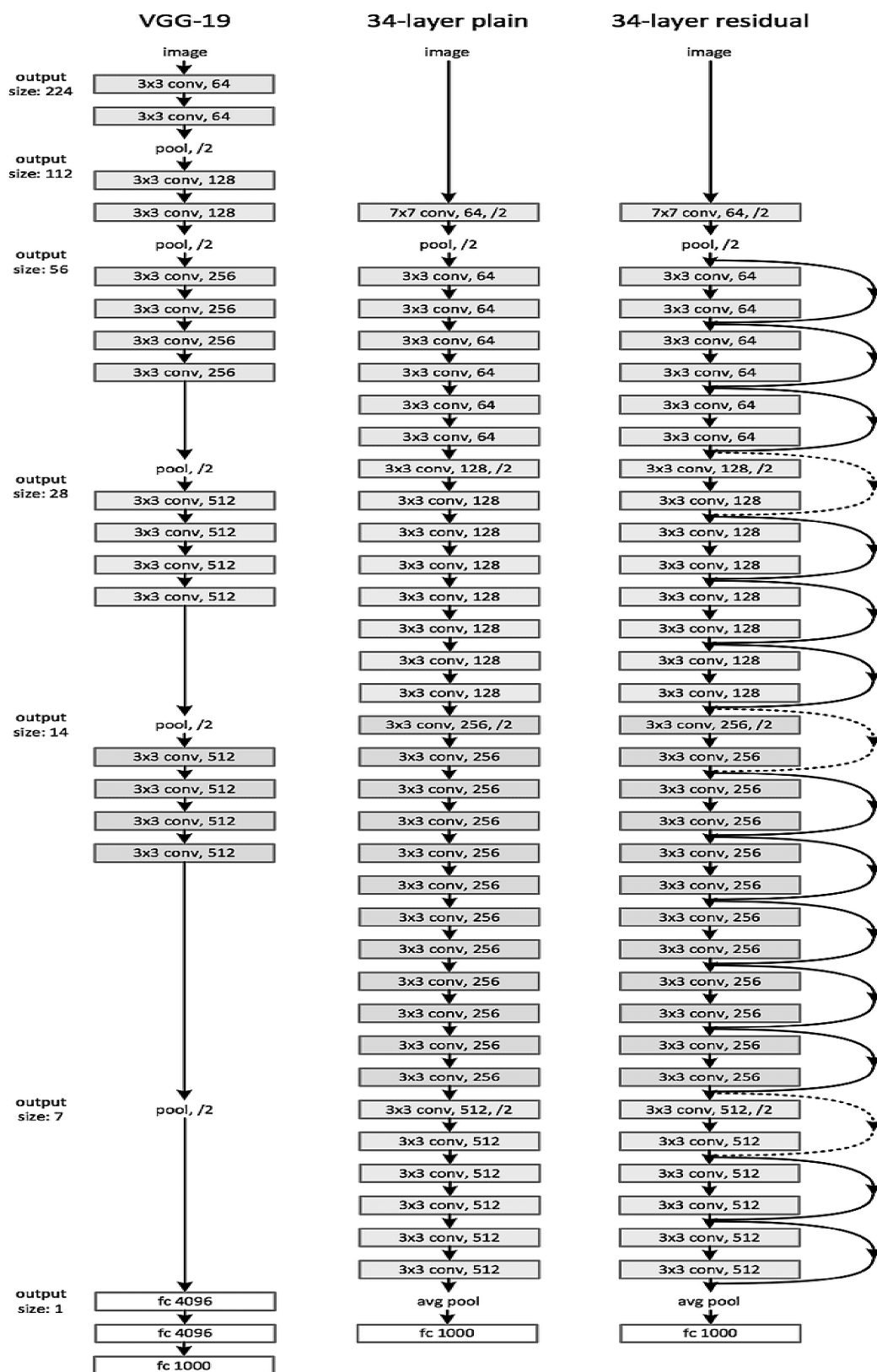
prefix <- 'r-batch-transform'
```

Sau khi chạy câu lệnh trên chúng ta đã có được ngăn chứa dữ liệu trên AWS với tên mặc định cho tài khoản của tôi là: `sagemaker-us-east-1-575208616377`. Chúng ta sẽ dùng đến nó khi chạy các mô hình thực nghiệm.

2.5. Mạng thần kinh tích chập

Mạng thần kinh tích chập (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Học sâu tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh [9][10].

- **VGG19**: VGG – 19 là mạng nơ-ron tích chập được đào tạo trên hơn một triệu hình ảnh từ cơ sở dữ liệu imagenet đây là cơ sở hình ảnh bao gồm 14 triệu hình ảnh được tổ chức theo hệ thống phân cấp wordnet, đây là mạng trực quan giúp sinh viên nghiên cứu và những người khác trong lĩnh vực nghiên cứu hình ảnh và thị giác. Chúng ta quan sát Hình. 1. Mô tả mô hình VGG-19.



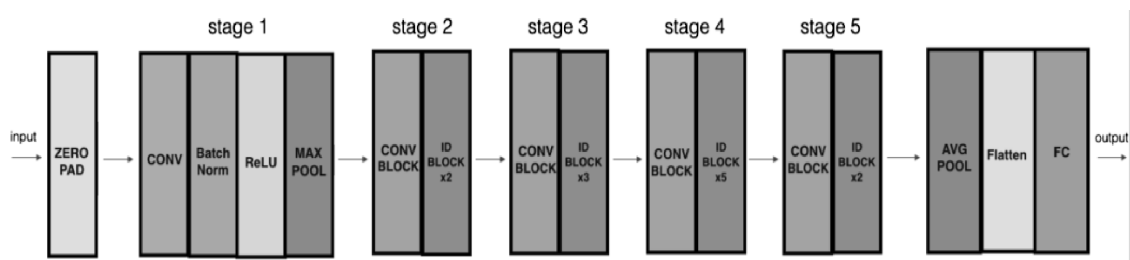
Hình 1. Mô tả mô hình VGG-19

Mạng lưới này bao gồm 19 lớp sâu và có thể phân loại hình ảnh thành hàng nghìn loại đối tượng như động vật hoặc đồ vật [9] v.v... Hình ảnh RGB có kích thước cố định ($224 * 224$) được cung cấp làm đầu vào cho mạng này, điều đó có nghĩa là ma trận có hình dạng ($224, 224, 3$). Quá trình tiền xử lý duy nhất đã được thực hiện là họ trừ đi giá trị RGB trung bình từ mỗi pixel, được tính toán trên toàn bộ tập huấn luyện. Các hạt nhân có kích thước ($3 * 3$) được sử dụng với kích thước sải chân là 1 pixel, điều này cho phép chúng bao phủ toàn bộ hình ảnh.

Đệm không gian đã được sử dụng để duy trì độ phân giải không gian của hình

ảnh. Tổng hợp tối đa được thực hiện trên cửa sổ $2 * 2$ pixel với slide 2. Tiếp theo là Đơn vị tuyến tính được chỉnh sửa (ReLU) để giới thiệu tính phi tuyến tính nhằm làm cho mô hình phân loại tốt hơn và cải thiện thời gian tính toán vì các mô hình trước đó sử dụng hàm tanh hoặc hàm sigmoid điều này tỏ ra tốt hơn nhiều so với những điều đó. Đã triển khai ba lớp được kết nối đầy đủ, trong đó hai lớp đầu tiên có kích thước 4096 và sau đó là lớp có 1000 kênh để phân loại ILSVRC 1000 chiều và lớp cuối cùng là chức năng softmax.

- ResNet50: Để có cái nhìn tổng thể về ResNet chúng ta quan sát Hình 2. Mô tả chi tiết kiến trúc mạng nơ ron ResNet.



Hình 2. Mô tả chi tiết kiến trúc mạng nơ ron ResNet

“ID BLOCK” trong hình trên là viết tắt của từ Identity block và ID BLOCK x3 nghĩa là có 3 khối Identity block chồng lên nhau. Nội dung hình trên như sau :

Zero-padding : Input với (3,3)

Stage 1 : Tích chập (Conv1) với 64 filters với shape (7,7), sử dụng stride (2,2). BatchNorm, MaxPooling (3,3).

Stag 2 : Convolutional block sử dụng 3 filter với size $64 \times 64 \times 256$, $f=3$, $s=1$. Có 2 Identity blocks với filter size $64 \times 64 \times 256$, $f=3$.

Stage 3 : Convolutional sử dụng 3

filter size $128 \times 128 \times 512$, $f=3$, $s=2$. Có 3 Identity blocks với filter size $128 \times 128 \times 512$, $f=3$.

Stage 4 : Convolutional sử dụng 3 filter size $256 \times 256 \times 1024$, $f=3$, $s=2$. Có 5 Identity blocks với filter size $256 \times 256 \times 1024$, $f=3$.

Stage 5 : Convolutional sử dụng 3 filter size $512 \times 512 \times 2048$, $f=3$, $s=2$. Có 2 Identity blocks với filter size $512 \times 512 \times 2048$, $f=3$.

The 2D Average Pooling : sử dụng với kích thước (2,2).

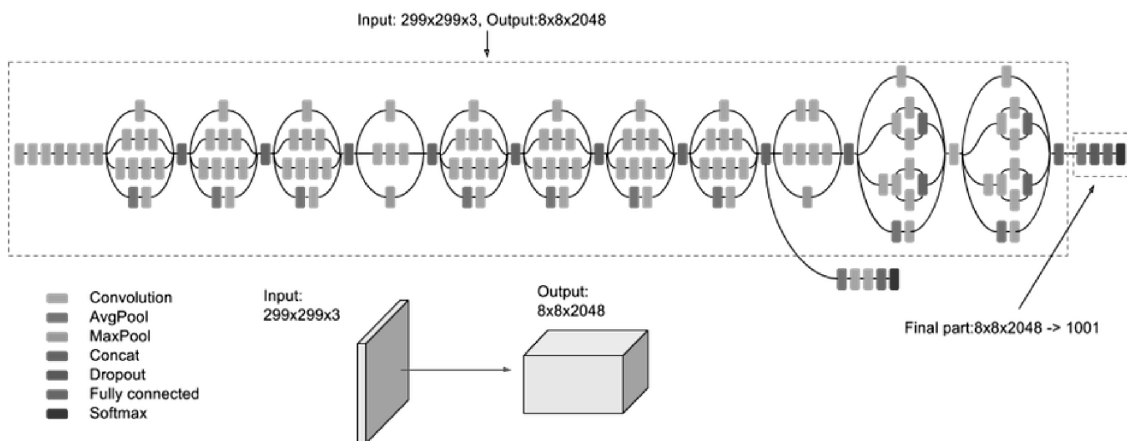
The Flatten.

Fully Connected (Dense) : sử dụng softmax activation.

- InceptionV3:

Inception v3 ban đầu được phát hành

vào năm, nó có tổng cộng 42 lớp và tỷ lệ lỗi thấp hơn so với các phiên bản tiền nhiệm. Sau khi thực hiện tất cả các tối ưu hóa Inceptionv3 được thể hiện như Hình 3. Mô hình InceptionV3.



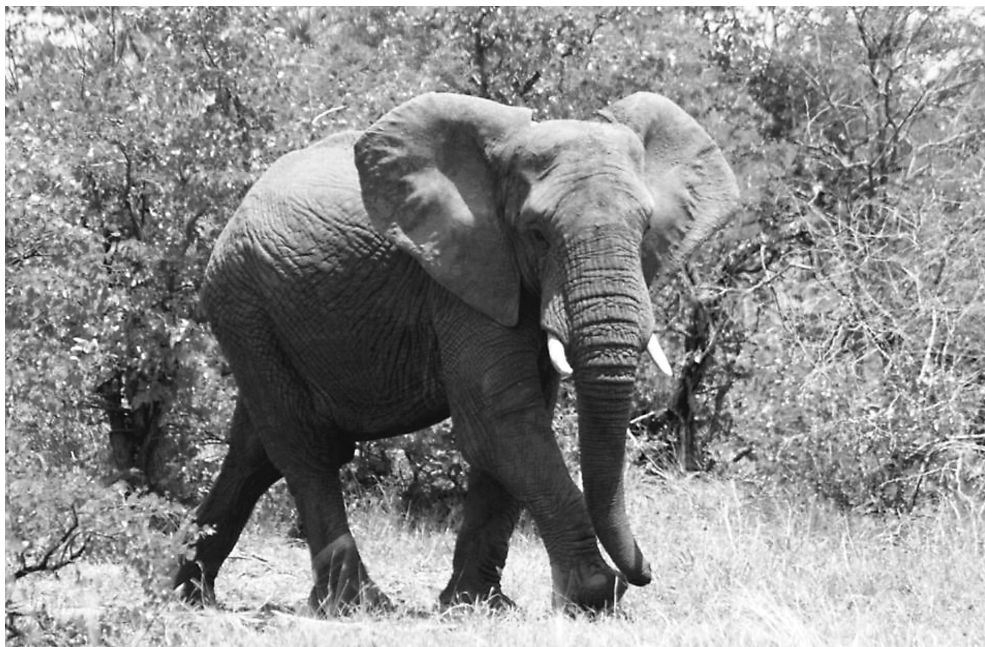
Hình 3. Mô hình InceptionV3

Inception V3 khởi động được tạo thành từ 42 lớp, cao hơn một chút so với các mẫu V1 và V2 khởi động trước đó. Nhưng hiệu quả của mô hình này thực sự rất ấn tượng. Chúng ta sẽ đi sâu vào nó một chút, nhưng

trước khi chúng ta hãy xem chi tiết các thành phần mà mô hình Inception V3 được làm bằng gì.

2.6. Thực nghiệm

2.6.1 Chọn dữ liệu



Hình 4. Hình ảnh một con voi đang đi trong rừng cây được lấy ngẫu nhiên trên mạng

Trong bài viết này, chúng tôi tiến hành thực nghiệm với hình ảnh lấy ngẫu nhiên trên internet là một con voi đang đi trong rừng cây như. Thực hiện đưa hình ảnh về đúng định dạng huấn luyện của mô hình là 224x224 pixel.

2.6.2 Công cụ

Trong thực nghiệm của nghiên cứu này sử dụng ngôn ngữ lập trình R trong Rstudio, một tài khoản điện toán đám mây AWS để có thể cung cấp máy chủ “t2.micro” và dịch vụ lưu trữ S3, tệp hình ảnh định dạng jpg, mô hình phân loại ảnh

Restnet 50 và các gói thư viện “aws.ec2”, keras, TensorFlow.

2.6.3 Kịch bản chạy

Hình ảnh ngẫu nhiên được chuẩn bị với kích thước 224x224 để phù hợp với mô hình. Từ R tạo kết nối đến máy chủ trên AWS. Sau đó, thực hiện đọc hình ảnh lưu trên S3. Cuối cùng, chúng tôi sẽ chạy phân lớp ảnh với mô hình: ResNet50 đã được huấn luyện với dữ liệu cá tra dựng. Kết quả như hình bên dưới.

- **Phân loại các lớp ImageNet với ResNet50:**

	class_name	class_description	score
1	n02504013	Indian_elephant	0.79484349
2	n01871265	tusker	0.11192206
3	n02504458	African_elephant	0.08656283

Hình 5. Kết quả chạy thực nghiệm với ResNet50

Dựa trên kết quả này cho thấy “n02504013 Indian_elephant 0.79484349” là kết quả cao hơn. Nghĩa là với hình ảnh đầu vào ban đầu khi cho phân loại với mô

hình ResNet50 thì giá trị n02504013 là đáng tin cậy hơn.

- **Trích xuất các tính năng từ một lớp trung gian tùy ý với VGG19:**

img	<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=224x224 at 0x18300001F70>
values	
base_model	<tensorflow.python.keras.engine.functional.Functional object at 0x000001835CE73E80>
block4_pool_features	Large array (100352 elements, 803 kb)
num [1, 1:14, 1:14, 1:512]	0 44.1 0 0 0 ...
img_path	"elephant.jpg"
model	<tensorflow.python.keras.engine.functional.Functional object at 0x000001835CE73910>
x	Large array (150528 elements, 1.2 MB)
num [1, 1:224, 1:224, 1:3]	-84.9 -66.9 -54.9 -19.9 -11.9 ...

Hình 6. Trích xuất tính năng từ một lớp

Kết quả trích xuất tính năng từ một lớp trung gian tùy ý cho thấy tại lớp 0x000001835CE73910 có 150528 các yếu

tố được trích xuất.

- **Tinh chỉnh InceptionV3 trên một nhóm lớp mới:**

```

1 input_2
2 conv2d
3 batch_normalization
4 activation
5 conv2d_1
6 batch_normalization_1
7 activation_1
8 conv2d_2
9 batch_normalization_2
10 activation_2
11 max_pooling2d
12 conv2d_3
13 batch_normalization_3
14 activation_3
15 conv2d_4
16 batch_normalization_4
17 activation_4
18 max_pooling2d_1
19 conv2d_8
20 batch_normalization_8
21 activation_8
22 conv2d_6
23 conv2d_9
24 batch_normalization_6
25 batch_normalization_9
26 activation_6
27 activation_9
28 average_pooling2d
29 conv2d_5
30 conv2d_7
31 conv2d_10
32 conv2d_11
33 batch_normalization_5
34 batch_normalization_7
35 batch_normalization_10
36 batch_normalization_11
37 activation_5
38 activation_7
39 activation_10
40 activation_11
41 mixed0
42 conv2d_15
43 batch_normalization_15
44 activation_15
45 conv2d_13
46 conv2d_16
47 batch_normalization_13
48 batch_normalization_16
49 activation_13
50 activation_16
51 average_pooling2d_1
52 conv2d_12

```

Hình 6. Kết quả tinh chỉnh InceptionV3 trên một nhóm lớp mới

Input_tensor: tensor Keras tùy chọn (tức là đầu ra của layers.Input()) để sử dụng làm đầu vào hình ảnh cho mô hình. Input_shape: bộ hình dạng tùy chọn, chỉ được chỉ định nếu include_top là Sai (nếu không thì hình dạng đầu vào phải là (299, 299, 3) (với channels_last định dạng dữ liệu) hoặc (3, 299, 299) (với channels_first định dạng dữ liệu). Nó phải có chính

xác 3 kênh đầu vào và chiều rộng và chiều cao không được nhỏ hơn 139. Ví dụ: (150, 150, 3) sẽ là một giá trị hợp lệ. Optional number of classes to classify images into, only to be specified if “include_top” is True, and if no “weights” argument is specified.

Nhìn chung cả 3 mô hình đều cho độ chính xác rất cao. Tùy vào từng trường hợp, lĩnh vực ứng dụng mà chọn mô hình phù hợp để đạt được kết quả tốt hơn.

2.6.4 Thảo luận

Để làm tốt h trợc sâu trên đám mây AWS, chúng ta cần có hiểu biết về hoạt động của chúng, hiểu về thuật toán học sâu, hiểu về lĩnh vực nghiên cứu trong thực tế và sự sáng tạo. Tất cả phải được kết hợp hài hòa để tạo ra một mô hình học sâu hoàn chỉnh. Trong nghiên cứu này, chúng tôi chỉ sử dụng một số mô hình học sâu từ mạng nơ ron tích chập. Nhưng trên thực tế, có rất nhiều loại mô hình học sâu, vì vậy để có cái nhìn toàn diện hơn, chúng ta có thể áp dụng cách tiếp cận này cho các loại mô hình học sâu khác.

3. KẾT LUẬN

Chúng tôi đã cố gắng trình bày một phương pháp tiếp cận tương tác với các dịch vụ đám mây AWS từ R để học sâu bằng một quy trình đơn giản cụ thể để tạo ra các sản phẩm từ mô hình học sâu, cung cấp một cái nhìn hệ thống cho những người mới tìm hiểu về lĩnh vực học sâu và những người làm trong các lĩnh vực khác muốn thực hiện nó. Tuy chưa đầy đủ nhưng nó đã tạo ra những bước đi đầu tiên và hứng thú cho việc nghiên cứu chuyên sâu trong thời gian sắp tới.

TÀI LIỆU THAM KHẢO:

- [1]. dominhhai.github.io, <https://dominhhai.github.io/vi/2018/04/nn-intro/>, last accessed 2022/06/20.
- [2]. viblo.asia, <https://viblo.asia/p/nn-mang-no-ron-nhan-tao-neural-networks-bWrZn6dwZxw>, last accessed 2022/06/20.
- [3]. careerlink.vn, <https://www.careerlink.vn/cam-nang-viec-lam/kien-thuc-kinh-te/neural-network-la-gi-vai-tro-va-ung-dung-cua-neural-network/>, last accessed 2022/06/20.
- [4]. r-bloggers.com, <https://www.r-bloggers.com/2018/06/interacting-with-aws-from-r/>, last accessed 2022/06/20.
- [5]. viblo.asia, <https://viblo.asia/p/tim-hieu-ve-convolutional-neural-network-va-lam-mot-vi-du-nho-ve-phan-loai-anh-aWj53WXo56m>, last accessed 2022/06/20.
- [6]. iq.opengenus.org, <https://iq.opengenus.org/vgg19-architecture/>, last accessed 2022/06/20.
- [7]. iq.opengenus.org, <https://iq.opengenus.org/inception-v3-model-architecture/>, last accessed 2022/06/20.
- [8]. AWS Homepage, <http://www.aws.amazon.com/vi/>, last accessed 2022/06/30.
- [9]. Mondal, S., Agarwal, K., Rashid, M., Deep Learning Approach for Automatic Classification of X-Ray Images using Convolutional Neural Network, 2019.
- [10]. Indolia, S., Goswani, K. A., S.P. Mishra, Asopa, P., : Conceptual understanding of convolutional neural network-a deep learning approach, June 2018.
- [11]. Schmidhuber, J : Deep Learning in Neural Networks: An Overview, 8 October 2014.