

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG



CƠ SỞ VÀ ỨNG DỤNG AI
FINAdL PROJECT

**Write an image classification program
based on deep convolutional neural networks in Python**

GVHD: TS. HUỖNH THẾ THIỆN

Sinh viên: ĐỖ THÁI DƯƠNG

MSSV: 21119058

LÂM HOÀNG PHÚC

MSSV: 21119350

TP. Hồ Chí Minh, tháng 05 năm 2024

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BẢNG NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Họ tên	MSSV	Đánh giá	Điểm
Đỗ Thái Dương	21119058		
Lâm Hoàng Phúc	21119350		

NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày... tháng... năm 2024

Giảng viên nhận xét

(Ký & ghi rõ họ tên)

Mục Lục

I. Tổng quan về Deep Learning	1
1. Deep Learning là gì ?	1
2. Cách thức hoạt động của mô hình học sâu - deep learning	1
3. Ưu nhược điểm học sâu (deep learning)	1
4. Tại sao nên sử dụng thuật toán Deep Learning?	2
5. Ứng dụng thực tiễn của Deep Learning	3
6. Khi nào cần sử dụng mô hình deep learning?	4
II. Các kỹ thuật Deep Learning	4
III. Kiến trúc mạng	11
IV. Quyết định thiết kế	13
1. Lựa chọn Kiến trúc Mạng:	13
2. Thiết kế Lớp (Layer Design):	13
3. Tối ưu hóa và huấn luyện Mô hình:	13
4. Sử dụng phần mềm:	13
V. Kết quả mô phỏng	14
VI. Kết luận	14

I. Tổng quan về Deep Learning

1. Deep Learning là gì ?

Mô hình deep learning - học sâu là một nhánh nhỏ của AI - artificial intelligence (trí tuệ nhân tạo). Nó hoạt động dựa trên mạng nơ-ron nhân tạo (neural networks) để phân tích, xử lý dữ liệu và mô phỏng bộ não của con người.

Deep Learning (học sâu) cũng có thể được coi là một lĩnh vực thuộc Machine Learning – nơi máy tính tự học và cải thiện thông qua các thuật toán. Thế nhưng, Deep Learning được xây dựng dựa trên những khái niệm phức tạp hơn.

Chủ yếu hoạt động với mạng thần kinh nhân tạo để bắt chước khả năng tư duy và suy nghĩ của bộ não con người. Thực ra các khái niệm liên quan đến mạng nơ-ron nhân tạo và Deep Learning đã được phát triển những năm 1960. Nhưng nó bị giới hạn bởi lượng dữ liệu và khả năng tính toán tại thời điểm đó.

Trong những năm gần đây, những tiến bộ trong phân tích big data đã cho phép chúng ta tận dụng tối đa khả năng của mạng lưới thần kinh nhân tạo. Mạng nơ-ron nhân tạo (neural networks) là động lực chính đằng sau sự phát triển của Deep Learning.

Mạng lưới thần kinh sâu (DNN - Deep neural networks) bao gồm nhiều lớp tế bào thần kinh khác nhau, có khả năng thực hiện các tính toán rất phức tạp. Deep Learning hiện đang phát triển rất nhanh và được coi là một trong những bước đột phá lớn nhất trong Machine Learning.

2. Cách thức hoạt động của mô hình học sâu - deep learning

Mạng thần kinh nhân tạo trong Deep Learning được xây dựng để mô phỏng khả năng tư duy của bộ não con người. Cách thức hoạt động như sau:

- Một mạng nơ-ron bao gồm nhiều lớp (layer) khác nhau, càng nhiều lớp thì mạng sẽ càng “sâu”. Trong mỗi lớp là các node (nút mạng) và được liên kết với các lớp khác bên cạnh.
- Mỗi kết nối giữa các nút sẽ có một trọng số tương ứng, trọng số càng cao thì mức độ ảnh hưởng của kết nối này đến mạng nơ-ron càng lớn.
- Mỗi nơ-ron sẽ có một chức năng kích hoạt, về cơ bản chịu trách nhiệm "chuẩn hóa" đầu ra từ nơ-ron này.
- Dữ liệu được người dùng nhập vào mạng thần kinh đi qua tất cả các lớp và trả về kết quả ở layer cuối cùng, được gọi là lớp đầu ra (output layer).
- Trong quá trình đào tạo mô hình mạng nơ-ron, các trọng số sẽ được thay đổi và nhiệm vụ của mô hình là tìm tập giá trị của các trọng số sao cho phán đoán đúng nhất.

Các hệ thống Deep Learning yêu cầu phần cứng rất mạnh để có thể xử lý lượng lớn dữ liệu và thực hiện các phép tính phức tạp. Nhiều mô hình Deep Learning có thể mất hàng tuần hoặc thậm chí hàng tháng để triển khai trên phần cứng tiên tiến nhất hiện nay.

3. Ưu nhược điểm học sâu (deep learning)

Deep Learning là một bước ngoặt lớn trong lĩnh vực trí tuệ nhân tạo - artificial intelligence. Nó cho phép các nhà khoa học dữ liệu khác xây dựng nhiều mô hình có độ chính xác cao trong các lĩnh vực nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, data...

Một số ưu điểm nổi bật của Deep Learning bao gồm:

- Cấu trúc neural networks linh hoạt, dễ dàng thay đổi để phù hợp với nhiều thuật toán khác nhau.
- Có khả năng giải nhiều vấn đề phức tạp với độ chính xác rất cao.

- Khả năng tự động hóa cao, tự điều chỉnh và tự tối ưu hóa.
 - Có khả năng thực hiện tính toán song song, hiệu năng tốt, xử lý lượng dữ liệu lớn.
- Bên cạnh ưu điểm, Deep Learning vẫn còn tồn tại nhiều hạn chế:
- Cần khối lượng dữ liệu khổng lồ để khai thác tối đa khả năng của Deep Learning.
 - Chi phí tính toán cao vì phải xử lý nhiều mô hình phức tạp.
 - Không có nền tảng lý thuyết vững chắc để chọn các công cụ tối ưu cho Deep Learning.

4. Tại sao nên sử dụng thuật toán Deep Learning?

Dưới đây là một lợi thế khi sử dụng các thuật toán Deep Learning thay vì phương pháp machine learning truyền thống:

Tự động hóa các tính năng

Một trong những ưu điểm của việc sử dụng deep learning là khả năng tự động hóa các tính năng. Cụ thể các thuật toán deep learning tạo ra các tính năng mới từ một số tính năng hạn chế ngay trong dữ liệu đào tạo mà không cần phải thực hiện bởi con người.

Điều này có nghĩa là deep learning có thể thực hiện các tác vụ phức tạp, đòi hỏi kỹ thuật cao. Đối với các doanh nghiệp, lợi thế của việc tự động hóa các tính năng học sâu cho phép triển khai ứng dụng hoặc công nghệ nhanh hơn, ổn định và độ chính xác cao hơn.

Tối ưu sự tương thích với các dữ liệu phi cấu trúc

Deep learning có khả năng hoạt động tốt với dữ liệu phi cấu trúc. Điều này đặc biệt phù hợp với bối cảnh hiện tại. Khi các dữ liệu được sử dụng phổ biến ở các doanh nghiệp đa phần là hình ảnh, văn bản và giọng nói...đều là dạng dữ liệu không có cấu trúc nhất định.

Nếu tiếp tục sử dụng các thuật toán machine learning bị hạn chế về khả năng phân tích dữ liệu phi cấu trúc, điều đó có nghĩa là lượng thông tin này sẽ không được khai thác triệt để. Ảnh hưởng đến bán hàng, marketing và doanh thu.

Khả năng tự học tốt hơn

Các lớp thần kinh trong học sâu cho phép các mô hình vận hành hiệu quả hơn. Đặc biệt để thực hiện các tác vụ phức tạp, chuyên sâu về tính toán, có thể thực hiện đồng thời nhiều thao tác phức tạp.

Deep learning thể hiện rõ trong các nhiệm vụ nhận thức của máy, còn được gọi là khả năng hiểu các đầu vào như hình ảnh, âm thanh và video giống như con người. Học sâu cũng hỗ trợ xác minh độ chính xác, dự đoán/đầu ra cụ thể và thực hiện các điều chỉnh khi cần thiết.

Ưu điểm về thuật toán phân tán và song song

Một mạng thần kinh hoặc mô hình học sâu phải mất nhiều ngày để tìm hiểu các tham số xác định mô hình. Các thuật toán song song và phân tán giải quyết vấn đề này bằng cách cho phép đào tạo các mô hình học sâu nhanh hơn nhiều.

Học sâu sẽ được đào tạo thông qua đào tạo cục bộ, GPU hoặc kết hợp cả hai. Tuy nhiên, không thể lưu trữ một lượng lớn bộ dữ liệu đào tạo có liên quan trên cùng một máy, dẫn đến việc tạo ra dữ liệu song song.

Tiết kiệm chi phí

Mặc dù việc đào tạo các mô hình deep learning có thể tốn kém nhưng nếu được đào tạo bài bản, deep learning sẽ giúp doanh nghiệp tối ưu hóa các khoản chi tiêu không

cần thiết.

Trong các ngành như sản xuất, tư vấn hoặc thậm chí bán lẻ, chi phí cho một dự đoán sai hoặc lỗi sản phẩm là rất lớn. Nó thường lớn hơn chi phí đào tạo một mô hình deep learning.

Phân tích nâng cao

Khi được áp dụng deep learning vào khoa học dữ liệu, có thể tạo ra các mô hình xử lý hiệu quả hơn. Nó có thể giám sát giúp thúc đẩy cải tiến liên tục, mang lại kết quả và độ chính xác cao.

Đồng thời cung cấp cho các nhà khoa học dữ liệu các kết quả phân tích ngắn gọn và đáng tin cậy hơn. Công nghệ được các doanh nghiệp ứng dụng để hỗ trợ các phần mềm từ tiếp thị đến bán hàng, nhân sự, kế toán,...

Khả năng mở rộng

Deep learning có khả năng mở rộng cao nhờ khả năng xử lý lượng dữ liệu lớn và thực hiện nhiều tính toán cho kết quả tốt nhất, tối ưu chi phí và thời gian. Điều này ảnh hưởng trực tiếp đến năng suất.

5. Ứng dụng thực tiễn của Deep Learning

Deep Learning được ứng dụng trong các công việc đòi hỏi khả năng tính toán cao, xử lý dữ liệu lớn và độ phức tạp lớn. Cùng tìm hiểu 5 ứng dụng phổ biến nhất của Deep Learning trên thực tế:

Hệ thống xe tự lái

Một trong những công nghệ mới và thú vị nhất hiện nay là hệ thống lái xe tự động, được xây dựng trên các mạng thần kinh cấp cao. Nói một cách đơn giản, các mô hình Deep Learning sẽ nhận diện các vật thể trong môi trường xung quanh xe.

Tiếp đến mô hình sẽ tiến hành tính toán khoảng cách giữa xe và các phương tiện khác, xác định tín hiệu đèn giao thông, làn đường quy định... Từ đó đưa ra các quyết định tối ưu nhất và nhanh nhất.

Phân tích cảm xúc

Đây là lĩnh vực phân tích cảm xúc của con người thông qua xử lý ngôn ngữ tự nhiên, phân tích văn bản và thống kê. Doanh nghiệp có thể ứng dụng Deep Learning để hiểu và phán đoán cảm xúc của khách hàng dựa trên các đánh giá, bình luận, feedback... từ đó đưa ra các chiến lược kinh doanh, marketing phù hợp với từng phân khúc khách hàng.

Mạng xã hội

Một số nền tảng mạng xã hội lớn như Twitter cũng áp dụng thuật toán Deep Learning để cải thiện dịch vụ của họ. Cụ thể, các trang web này sẽ phân tích lượng lớn dữ liệu thông qua mạng thần kinh nhân tạo để tìm hiểu về sở thích và xu hướng hiện tại của người dùng.

Ngoài ra, Instagram hay facebook cũng sử dụng Deep Learning để tránh các hành vi bạo lực trên không gian mạng. Chặn các bình luận xúc phạm, vi phạm tiêu chuẩn cộng đồng...

Trợ lý ảo - virtual assistant

Trợ lý ảo đang được ứng dụng rất nhiều trong cuộc sống hàng ngày. Trong đó phổ biến phải kể đến chatbot, Google Assistant, Cortana. Siri,... Các trợ lý này được xây dựng dựa trên mô hình Deep Learning với các thuật toán giúp nhận dạng và xử lý dữ liệu như giọng nói, văn bản...

Lĩnh vực chăm sóc sức khỏe

Deep Learning cũng có đóng góp đáng kể cho lĩnh vực y tế, trong đó các mô hình phổ biến bao gồm mô hình dự đoán bệnh, chẩn đoán ung thư, phân tích kết quả chụp MRI, X-quang...

6. Khi nào cần sử dụng mô hình deep learning?

Mặc dù có hiệu suất và độ chính xác vượt trội nhờ nhiều mô hình phức tạp và nguồn dữ liệu khổng lồ. Thế nhưng Deep Learning không hẳn là lựa chọn duy nhất hay tối ưu nhất trong nhiều trường hợp.

Việc quyết định có áp dụng Deep Learning hay cho machine learning hay không phụ thuộc phần lớn vào mục tiêu và chiến lược kinh doanh cụ thể. Cũng như lượng dữ liệu, tài nguyên của dự án...

Vì vậy, hãy cân nhắc kỹ lưỡng các yếu tố dưới đây trước khi lựa chọn ứng dụng mô hình học sâu deep learning:

Mức độ phức tạp của dự án

Deep Learning là khả năng giải quyết các vấn đề phức tạp, phân tích và tìm hiểu các mối quan hệ ẩn trong dữ liệu. Nếu dự án cần xử lý nhiều dữ liệu phi cấu trúc, thì đây deep learning sẽ vô cùng phù hợp.

Chẳng hạn như phân loại hình ảnh, xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói,... Mặt khác, đối với những bài toán có độ phức tạp vừa phải, không đòi hỏi tính toán rắc rối....thì thuật toán Machine Learning sẽ là lựa chọn phù hợp hơn.

Tài nguyên

Với cơ sở dữ liệu lớn như hiện nay đã giúp việc xây dựng các mô hình Deep Learning dễ dàng hơn. Tuy nhiên, do phải xử lý lượng dữ liệu cực lớn nên các mô hình Deep Learning thường rất đòi hỏi cao về mặt tài nguyên và GPU để đạt hiệu suất tốt nhất.

Mặt khác, các thuật toán Machine Learning chỉ cần một CPU và phần cứng vừa phải, nhanh hơn và có thể dễ dàng thử nghiệm nhiều kỹ thuật và mô hình khác nhau mà không phải lo lắng về tài nguyên.

Số lượng dữ liệu

Các thuật toán Deep Learning có thể tìm thấy các mối quan hệ sâu sắc trong tập dữ liệu. Tuy nhiên, điều này cũng có nghĩa là lượng dữ liệu đầu vào (dữ liệu được gán nhãn) phải lớn hơn nhiều so với các thuật toán Machine Learning.

Việc gán nhãn dữ liệu cũng đòi hỏi nguồn lực và thời gian lớn, đặc biệt trong lĩnh vực y tế đòi hỏi chuyên môn cao để có thể gán nhãn dữ liệu một cách chính xác. Trong những trường hợp này, người ta có thể nghĩ đến việc sử dụng thuật toán Machine Learning cổ điển thay vì Deep Learning.

Trong bài viết này chúng ta đã cũng tìm hiểu về khái niệm *Deep Learning là gì* cũng như ứng dụng thực tiễn của mô hình học sâu trong cuộc sống. Mong rằng qua những nội dung Tanca đã chia sẻ sẽ giúp bạn hiểu thêm về cách thức hoạt động, biết khi nào nên sử dụng Deep Learning và nhiều kiến thức phổ biến khác về khoa học máy tính.

II. Các kỹ thuật Deep Learning

Có rất nhiều kỹ thuật và thuật toán Deep Learning khác nhau, từ đơn giản đến vô cùng phức tạp, có thể ứng dụng vào hầu hết các bài toán trong lĩnh vực trí tuệ nhân tạo hiện nay. Ở phần cuối của bài viết, ta sẽ tìm hiểu một số kỹ thuật Deep Learning phổ biến nhất hiện nay.

Mạng nơ-ron cổ điển

Kiến trúc cổ điển của mạng nơ-ron là mạng kết nối đầy đủ, thường được xác định bằng các perceptron đa lớp. (Perceptron là một thuật toán đơn giản, cho phép tìm một ranh giới siêu phẳng cho các bài toán phân lớp nhị phân). Mạng nơ-ron cổ điển được thiết kế bởi Fran Rosenblatt vào năm 1958, chủ yếu được sử dụng cho các bài toán phân lớp nhị phân. Có ba loại hàm thường được sử dụng trong mô hình này là:

- Hàm tuyến tính.
- Hàm phi tuyến: gồm có hàm sigmoid, hàm tanh và hàm ReLU (Rectified Linear Unit).

Kiến trúc mạng nơ-ron cổ điển tương đối đơn giản, phù hợp nhất với các bộ dữ liệu có dạng bảng hoặc những bài toán phân loại, hồi quy có đầu vào là giá trị thực.

Mạng nơ-ron tích chập (CNN)

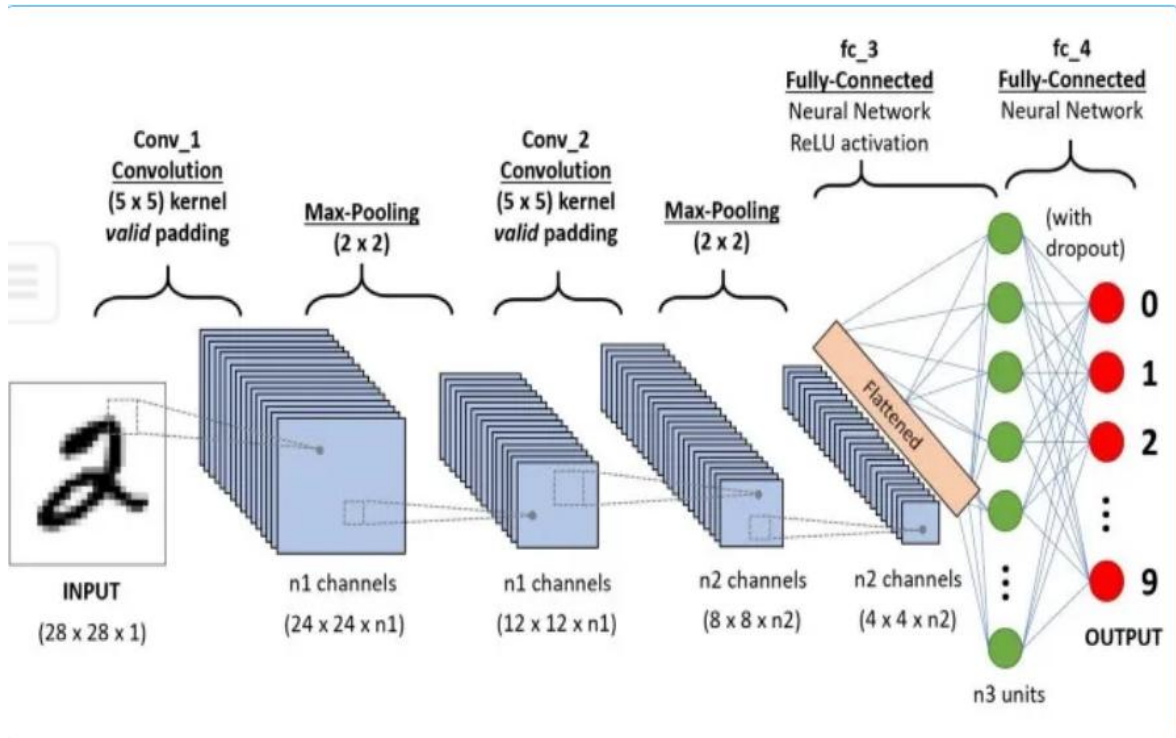
Mạng nơ-ron tích chập (Convolutional Neural Network – CNN) là một kiến trúc Neural Network nhân tạo nâng cao, được xây dựng để giải quyết các bài toán phức tạp, đặc biệt là liên quan đến xử lý hình ảnh.

Tích chập là một khái niệm trong xử lý tín hiệu số nhằm biến đổi thông tin đầu vào qua một phép tích chập với bộ lọc, nhằm trả về đầu ra là một tín hiệu mới. Tín hiệu này sẽ giảm bớt những đặc trưng mà bộ lọc không quan tâm, giữ lại những đặc trưng chính và quan trọng nhất.

Bên cạnh input layer và output layer, mô hình CNN còn có thêm một sampling layer để giới hạn số lượng nơ-ron tham gia vào các layer tương ứng. Việc xây dựng mô hình trải qua ba giai đoạn chính:

- Quá trình tích chập (convolution): Thông qua các tích chập giữa ma trận đầu vào với bộ lọc để tạo thành các đơn vị trong một tầng mới. Quá trình này có thể diễn ra liên tục ở phần đầu của mạng và thường sử dụng kèm với hàm kích hoạt ReLU. Mục tiêu của tầng này là trích xuất đặc trưng hai chiều.
- Quá trình tổng hợp (max pooling): Giảm kích thước khối ma trận đầu vào thông qua việc tìm ra 1 giá trị đại diện cho mỗi một vùng không gian mà bộ lọc đi qua sẽ không làm thay đổi các đường nét chính của bức ảnh nhưng lại giảm được kích thước của ảnh.
- Quá trình kết nối hoàn toàn (fully connected): Sau khi đã giảm kích thước đến một mức độ hợp lý, ma trận cần được trải phẳng (flatten) thành một vector và sử dụng các kết nối hoàn toàn giữa các tầng. Tầng kết nối hoàn toàn cuối cùng (fully connected layer) sẽ có số lượng đơn vị bằng với số lớp.

Dựa vào những đặc điểm của mình, các ứng dụng phổ biến nhất của mạng CNN gồm có: Nhận diện, phân tích và phân khúc hình ảnh, phân tích video, xử lý ngôn ngữ tự nhiên,...



Mạng nơ-ron tích chập (CNN)

Mạng nơ-ron hồi quy (RNN)

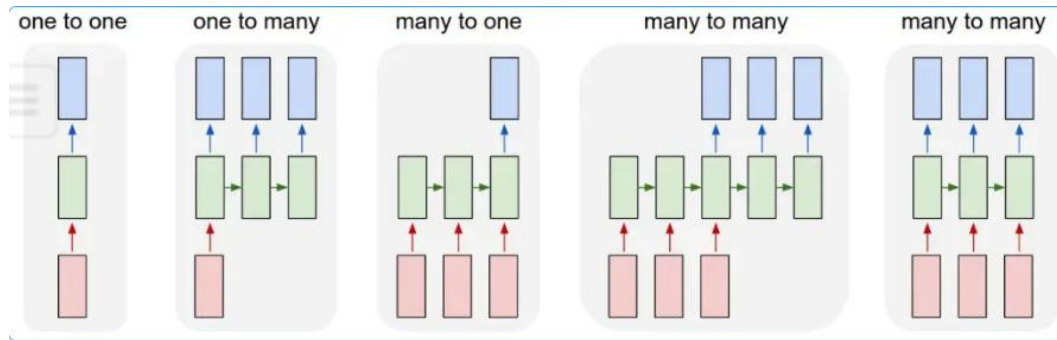
Recurrent Neural Network (RNN) là một thuật toán nổi tiếng trong lĩnh vực xử lý ngôn ngữ tự nhiên. Trong các mô hình mạng nơ-ron truyền thống, đầu vào và đầu ra độc lập với nhau, tuy nhiên RNN thực hiện cùng một tác vụ cho tất cả phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Vì vậy mạng RNN có khả năng nhớ các thông tin được tính toán trước đó.

Có hai thiết kế chính của RNN:

- LSTM (Long Short-Term Memory): Được dùng để dự đoán dữ liệu dạng chuỗi thời gian, có khả năng bỏ đi hoặc thêm các thông tin cần thiết, được điều chỉnh bởi các nhóm được gọi là cổng (gate): Input, Output và Forget.
- Gated RNN: Cũng là một thiết kế phổ biến trong lĩnh vực dự đoán dữ liệu của chuỗi thời gian, có hai cổng là Update và Reset.

Các dạng bài toán RNN:

- One to one: Chỉ có một input kết nối với một output duy nhất, chẳng hạn như các bài toán phân loại hình ảnh.
- One to many: Một input liên kết với nhiều chuỗi output, phổ biến là các bài toán đặt caption cho ảnh.
- Many to One: Nhiều input nhưng chỉ có output, ví dụ phổ biến là bài toán phân loại cảm xúc.
- Many to many: Nhiều input và nhiều output, chẳng hạn như phân loại video.

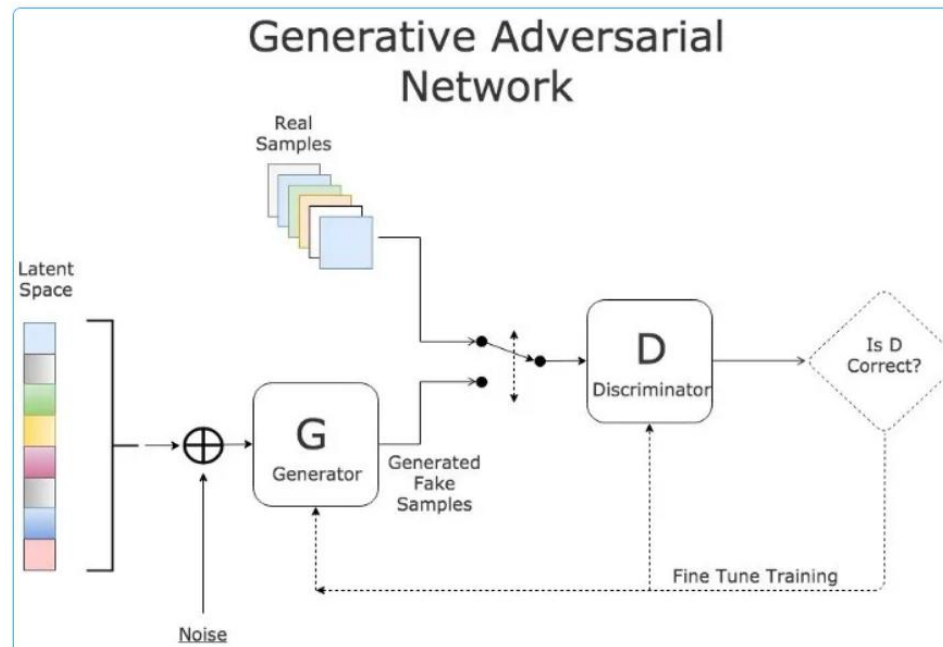


Mạng nơ-ron hồi quy (RNN)

Mạng sinh đối nghịch (GAN)

Generative Adversarial Networks (GAN) là lớp mô hình có mục tiêu tạo ra dữ liệu giả giống với thật, tên của mạng được dựa trên kiến trúc gồm hai mạng có mục tiêu đối nghịch nhau: Generator và Discriminator. Trong đó Generator học cách sinh dữ liệu giả để lừa mô hình Discriminator, còn Discriminator lại học cách phân biệt giữa dữ liệu giả và dữ liệu thật. Thông qua quá trình huấn luyện thì cả hai mô hình này đều cùng cải thiện được khả năng của mình.

Một số ứng dụng phổ biến của GAN là: Tạo khuôn mặt người, thay đổi độ tuổi khuôn mặt, sinh ảnh vật thể, tạo nhân vật hoạt hình,...



Mạng sinh đối nghịch (GAN)

Boltzmann machine

Đây là một mô hình mạng không có hướng xác định, vì vậy các node của mạng này được liên kết với nhau thành một hình tròn. Dựa vào kiến trúc này, máy Boltzmann (Boltzmann machine) thường được sử dụng để tạo ra các tham số cho mô hình. Các ứng dụng phổ biến nhất của mô hình là: giám sát hệ thống, xây dựng hệ thống khuyến nghị nhị phân,...

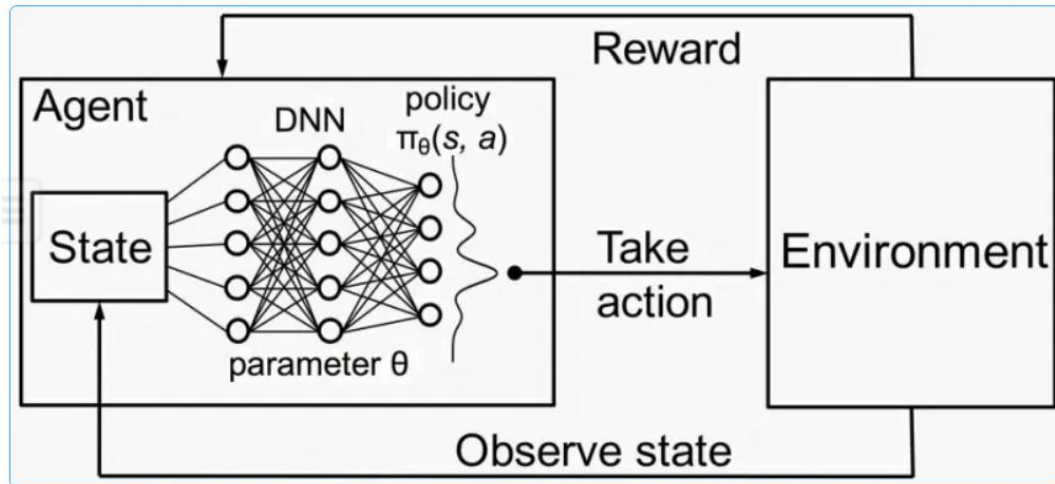
Học tăng cường sâu

Deep Reinforcement Learning là quá trình mà các tác tử (agent) tương tác với môi trường để thay đổi trạng thái của chính nó. Các tác tử có thể quan sát và thực hiện những

hành động phù hợp, từ đó giúp mạng đạt được mục tiêu.

Mô hình mạng này gồm một input layer, output layer và nhiều hidden layer khác, trong đó trạng thái của môi trường chính là input layer. Mô hình sẽ huấn luyện liên tục để dự đoán điểm đạt được sau mỗi hành động được thực hiện trong từng trạng thái nhất định.

Mô hình học tăng cường sâu được ứng dụng chủ yếu trong các game cờ vua, poker, xe tự lái, robot,...



Học tăng cường sâu

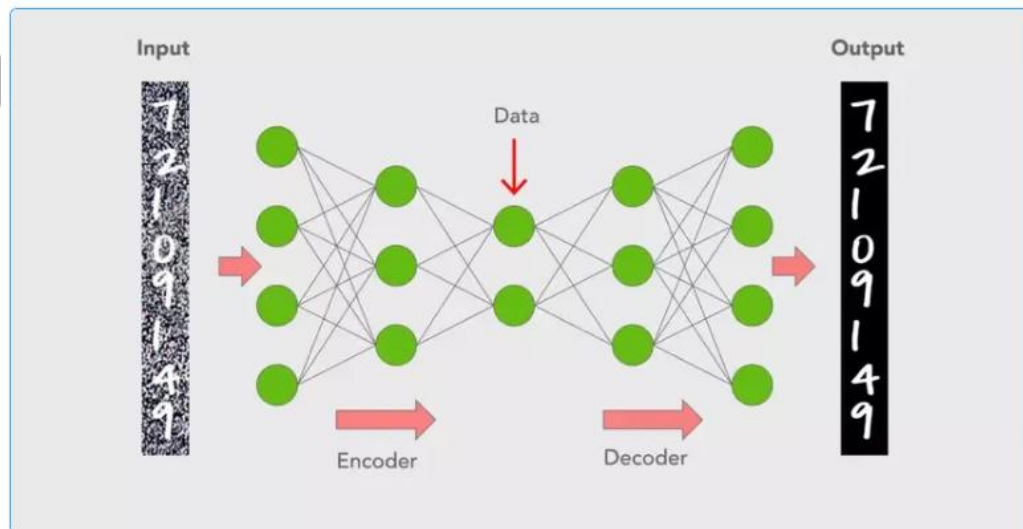
Autoencoder

Autoencoder là một trong những kỹ thuật Deep Learning phổ biến nhất hiện nay, có khả năng học các biểu diễn của dữ liệu đầu vào mà không cần nhãn, hay nói cách khác thì mạng này có khả năng học không giám sát (unsupervised learning).

Một số loại Autoencoder chính gồm có:

- **Sparse (thưa):** Số lượng hidden layer lớn hơn số lượng input layer nhằm hạn chế hiện tượng quá khớp (overfitting). Phương pháp này giới hạn hàm mất mát và ngăn không cho Autoencoder lạm dụng tất cả các node có trong mạng.
- **Denoising (lọc nhiễu):** Một phiên bản input được chuyển thành 0 ngẫu nhiên.
- **Contractive:** Bổ sung hệ số phạt vào hàm mất mát để hạn chế overfitting trong trường hợp số lượng hidden layer lớn hơn input layer.
- **Stacked:** Xếp chồng nhiều hidden layer lên nhau để tạo thành một mạng Autoencoder.

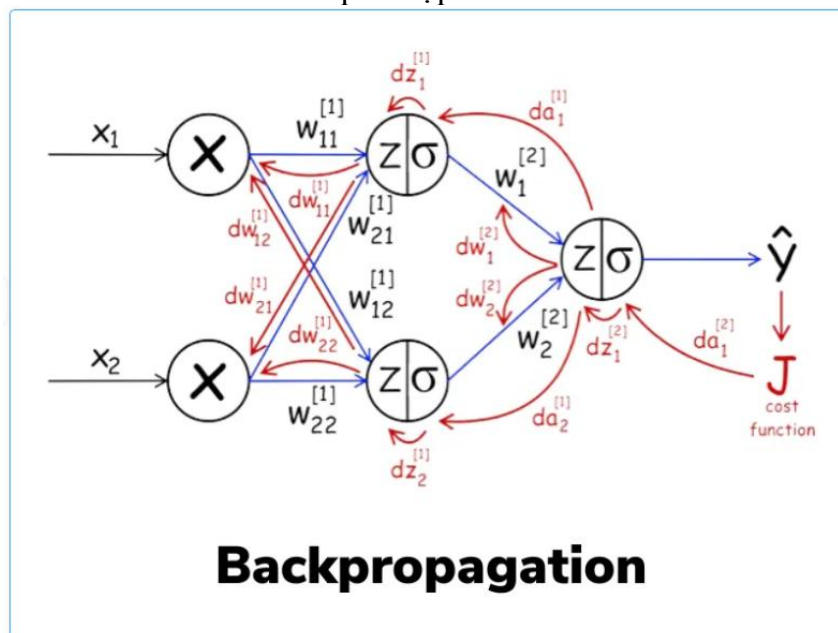
Các ứng dụng phổ biến: Phát hiện đặc trưng, xây dựng hệ thống khuyến nghị, bổ sung đặc trưng cho tập dữ liệu,...



Autoencoder

Back-propagation

Lan truyền ngược (back-propagation) là một trong những kỹ thuật quan trọng nhất của mạng nơ-ron. Về cơ bản thì đây là phương pháp giúp tính gradient ngược từ layer cuối cùng đến layer đầu tiên của mạng. Trước hết, mạng sẽ phân tích các tham số rồi điều chỉnh thông qua hàm mất mát. Tiếp theo, giá trị lỗi được tính toán sẽ lan truyền ngược lại để điều chỉnh các tham số cho phù hợp.



Back-propagation

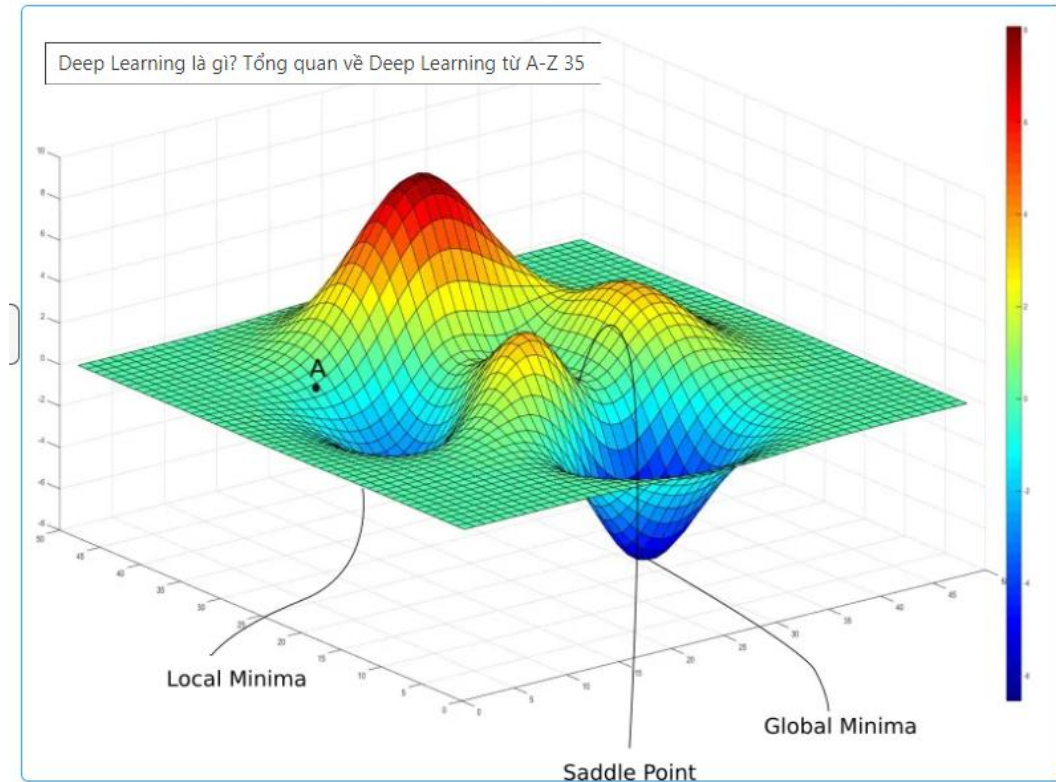
Gradient Descent

Trong Deep Learning và tối ưu hoá, ta thường phải tìm giá trị nhỏ nhất (hoặc lớn nhất) của một hàm số nào đó. Tuy nhiên việc tìm các điểm tối ưu toàn cục của hàm mất mát thường rất phức tạp, đôi khi là bất khả thi. Do đó ta có thể cố gắng tìm những điểm cực tiểu địa phương và có thể xem là nghiệm cần tìm của bài toán.

Các điểm cực tiểu địa phương về mặt toán học là nghiệm học phương trình đạo hàm bằng 0, tuy nhiên việc giải phương trình đạo hàm bằng 0 gần như là không thể trong

Machine Learning hay Deep Learning. Một cách tiếp cận phổ biến là xuất phát từ một điểm mà ta coi là gần với nghiệm của bài toán, sau đó dùng một phép lặp để tiến dần đến điểm cần tìm. Phương pháp này được gọi là hạ gradient và được sử dụng vô cùng phổ biến trong tối ưu.

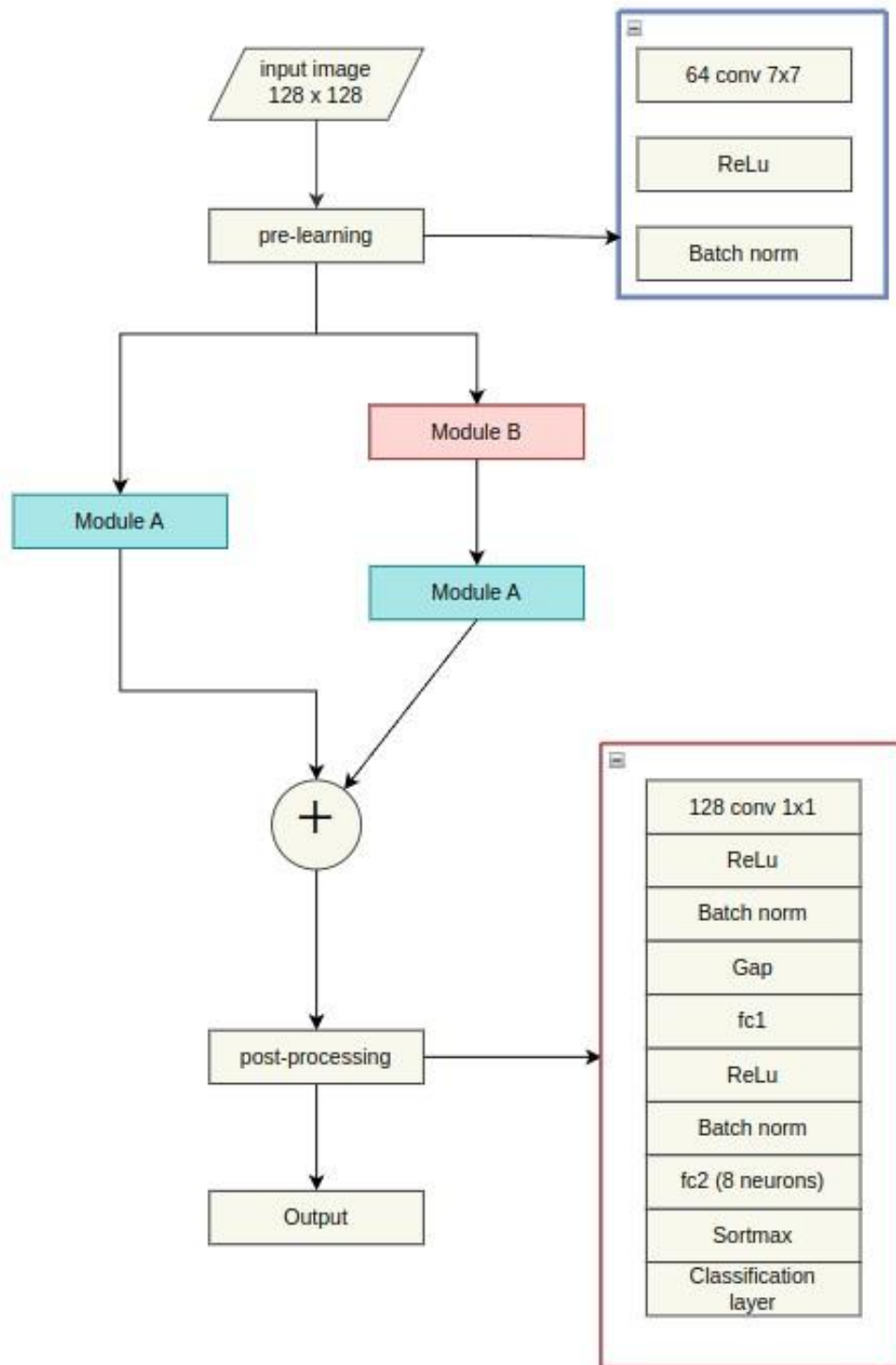
Với các mạng nơ-ron hiện đại, nhờ vào thuật toán lan truyền ngược mà gradient descent có thể nhanh hơn hàng triệu lần so với cách truyền thống.

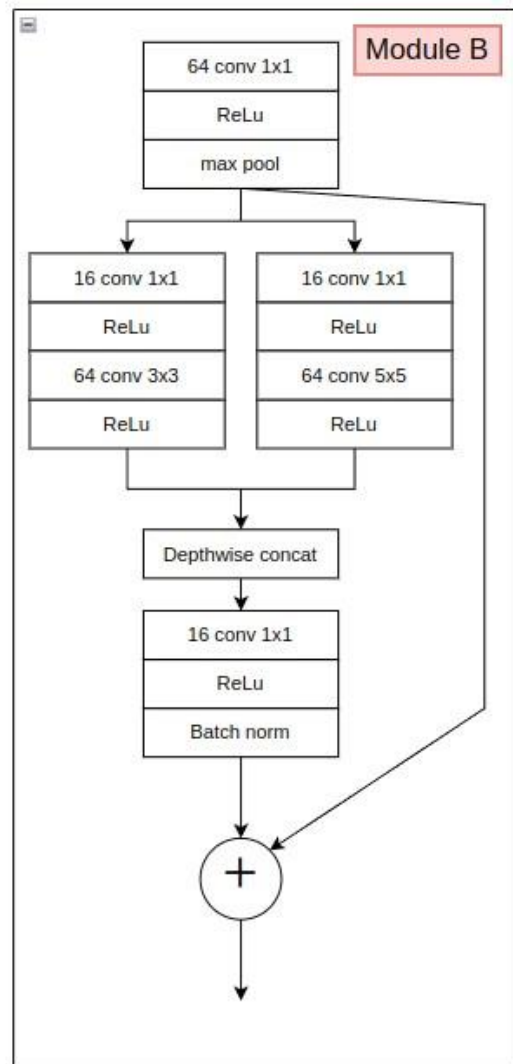
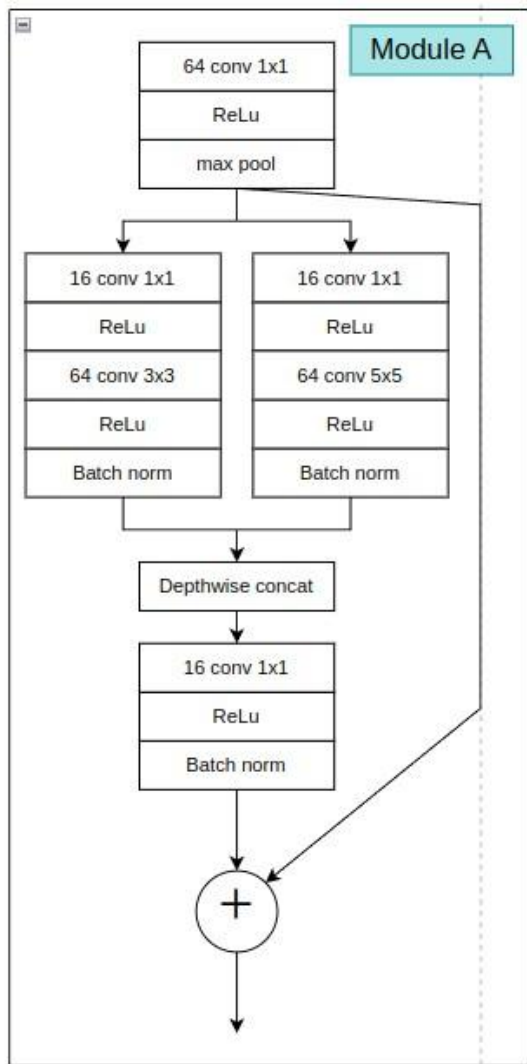


Gradient Descent

III. Kiến trúc mạng

Xây dựng theo mô hình mạng CNN, Model của bài được xây dựng như sau:





IV. Quyết định thiết kế

Dưới đây là các quyết định thiết kế chính mà nhóm đã thực hiện:

1. Lựa chọn Kiến trúc Mạng:

- Mạng nơ-ron tích chập (CNN): Chọn kiến trúc CNN do khả năng hiệu quả trong việc xử lý và phân tích hình ảnh. CNN giúp nhận diện các đặc trưng từ dữ liệu ảnh một cách hiệu quả, phù hợp với yêu cầu của dự án trong việc phân tích và nhận diện hình ảnh tự động.

2. Thiết kế Lớp (Layer Design):

- Thiết kế các lớp tích chập để trích xuất đặc trưng, theo sau là các lớp pooling để giảm kích thước không gian của đặc trưng, cuối cùng là các lớp fully connected để phân loại.
- Sử dụng các hàm kích hoạt ReLU cho các lớp tích chập và Softmax cho lớp output để phân loại.

3. Tối ưu hóa và huấn luyện Mô hình:

- Sử dụng thuật toán hàm Optimized để tối ưu hóa trọng số trong mạng, giúp giảm thiểu hàm mất mát và tăng độ chính xác của mô hình.
- Thiết lập quá trình huấn luyện với số Epoch và Batch size phù hợp để đạt hiệu quả cao trong quá trình học.

4. Sử dụng phần mềm:

- Sử dụng MATLAB và Kaggle để thực hiện dự án, từ việc phát triển mô hình đến thử nghiệm và đánh giá hiệu suất của mô hình.

V. Kết quả mô phỏng

Chạy lần 1:

```
Epoch 39/40
199/200 ----- 0s 40ms/step - accuracy: 0.9167 - loss: 0.1707
Epoch 39: ReduceLROnPlateau reducing learning rate to 1.9531249506599124e-07.
200/200 ----- 10s 52ms/step - accuracy: 0.9167 - loss: 0.1706 - val_accuracy: 0.7794 - val_loss: 0.8492 - learning_rate: 3.9062e-07
Epoch 40/40
200/200 ----- 10s 51ms/step - accuracy: 0.9168 - loss: 0.1668 - val_accuracy: 0.7794 - val_loss: 0.8407 - learning_rate: 1.9531e-07
150/150 ----- 2s 15ms/step - accuracy: 0.7603 - loss: 0.9270
```

Chạy lần 2:

```
Epoch 39/40
200/200 ----- 9s 43ms/step - accuracy: 0.9136 - loss: 0.1844 - val_accuracy: 0.7854 - val_loss: 0.6400 - learning_rate: 9.7656e-08
Epoch 40/40
199/200 ----- 0s 31ms/step - accuracy: 0.9177 - loss: 0.1829
Epoch 40: ReduceLROnPlateau reducing learning rate to 4.882812376649781e-08.
200/200 ----- 9s 44ms/step - accuracy: 0.9177 - loss: 0.1828 - val_accuracy: 0.7852 - val_loss: 0.6406 - learning_rate: 9.7656e-08
150/150 ----- 3s 17ms/step - accuracy: 0.7725 - loss: 0.6977
```

Chạy lần 3:

```
Epoch 39/40
199/200 ----- 0s 32ms/step - accuracy: 0.9187 - loss: 0.1770
Epoch 39: ReduceLROnPlateau reducing learning rate to 1.9531249506599124e-07.
200/200 ----- 9s 43ms/step - accuracy: 0.9187 - loss: 0.1770 - val_accuracy: 0.7754 - val_loss: 0.7936 - learning_rate: 3.9062e-07
Epoch 40/40
200/200 ----- 9s 43ms/step - accuracy: 0.9202 - loss: 0.1741 - val_accuracy: 0.7765 - val_loss: 0.7929 - learning_rate: 1.9531e-07
150/150 ----- 2s 15ms/step - accuracy: 0.7593 - loss: 0.8517
```

Số lượng parameters: 256,284.

Tính trung bình cả 3 lần chạy:

Trên tập train:

$$TB = ((0.9168 + 0.9177 + 0.9202) \times 100\%) / 3 = 91.8\%$$

Trên tập test:

$$TB = ((0.7794 + 0.7852 + 0.7765) \times 100\%) / 3 = 78\%$$

VI. Kết luận

Từ kết quả quá trình huấn luyện trên, mô hình đã đạt được một độ chính xác khá cao trên tập huấn luyện là 91.8%, nhưng lại không đạt được kết quả cao trên tập test là 78%. Điều này cho thấy mô hình vẫn chưa nhận dạng tốt các dữ liệu được cho.

Tuy nhiên, vẫn còn một số hạn chế trong nghiên cứu hiện tại, ví dụ: kích thước tập dữ liệu nhỏ, dữ liệu chưa đa dạng,....

Khả năng cải thiện:

Để cải thiện hiệu suất của mô hình, chúng tôi có thể thử nghiệm với các kiến trúc mạng khác nhau, tinh chỉnh siêu tham số, hoặc sử dụng các phương pháp tăng cường dữ liệu như lật, xoay hình ảnh.

Hướng phát triển tiếp theo:

Trong tương lai, chúng tôi dự định sẽ tinh chỉnh mô hình neural network của mình, đồng thời khảo sát và thử nghiệm các model khác như ResNet-50, Inception, VGG, hoặc là các mô hình khác dùng cho classification image như YoLov5,... để cải thiện độ chính xác và tổng quát hóa của mô hình.