HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN I

-----□□□□□-----

ĐỒ ÁN TỐT NGHIỆP

Đề tài: Nhận dạng hoạt động của người bằng học thích nghi

Giáo viên hướng dẫn:	PGS. TS. Phạm Văn Cường
Sinh viên:	Trần Khánh Hưng
Mã sinh viên:	B19DCCN331
Lớp:	D19HTTT3
Niên khóa:	2019 - 2024
Hệ đào tạo:	Đại học chính quy

Hà Nội, tháng 12 năm 2023

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

(Của giảng viên phản biện)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:.....(bằng chữ.....)

Đồng ý/ Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

Hà Nội, ngày ... tháng ... năm ...

Giảng viên phản biện

(Ký, ghi rõ họ tên)

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

(Của giảng viên hướng dẫn)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:.....(bằng chữ.....)

Đồng ý/ Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

Hà Nội, ngày ... tháng ... năm ...

Giảng viên hướng dẫn

(Ký, ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn đến các thầy, cô trong Khoa Công nghệ thông tin 1 tại Học viện Công nghệ Bưu Chính Viễn thông, những người trong hơn 4 năm vừa qua đã truyền đạt cho em rất nhiều kiến thức, kinh nghiệm quý báu, những hành trang cho em hướng đến tương lai.

Em xin chân thành cảm ơn thầy - PGS.TS Phạm Văn Cường, phó trưởng khoa Công nghệ thông tin I, trưởng bộ môn Khoa học máy tính, đã tận tình chỉ dạy và hướng dẫn em trong việc lựa chọn đề tài, thực hiện đề tài và viết báo cáo đồ án, giúp cho em có thể hoàn thành tốt đồ án này.

Em xin chân thành cảm ơn thầy – TS Trần Tiến Công, bộ môn Khoa học máy tính, Khoa Công nghệ thông tin I đã giúp đỡ em trong suốt quá trình thực hiện đề tài, đưa ra những lời khuyên giúp em hoàn thiện được đồ án một cách tốt nhất.

Em xin chân thành cảm ơn anh Nguyễn Duy Đức – cựu sinh viên khoá D18, An – sinh viên lớp E21CQCN04-B đã giúp đỡ em rất nhiều trong việc thu thập, xử lý dữ liệu và thử nghiệm mô hình để em có thể hoàn thành được đồ án này.

Em xin chân thành cảm ơn các anh chị, các bạn, các em đã và đang hoạt động tại câu lạc bộ ITPTIT, cảm ơn tất cả mọi người đã tạo ra một môi trường học tập chuyên nghiệp, sáng tạo để em có thể phát triển được như ngày hôm nay.

Cuối cùng em xin cảm ơn gia đình, bạn bè, những người đã luôn bên cạnh động viên em những lúc khó khăn và giúp đỡ em trong suốt thời gian học tập và nghiên cứu, tạo mọi điều kiện tốt nhất cho em để có thể hoàn thành tốt đồ án của mình.

Hà Nội, ngày 30 tháng 12 năm 2023

Tác giả

Trần Khánh Hưng

MỤC LỤC

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

MỞ ĐẦU

Bài toán ước lượng tư thế người là một trong các bài toán quan trọng và là nền tảng đẻ tiếp tục phát triển các bài toán sau đó như nhận dạng hành vi con người dựa trên tư thế, phát hiện hành vi bất thường, v.v. Ngày nay, với sự phát triển nhanh chóng của công nghệ thông tin và đặc biệt là sự phát triển của trí tuệ nhân tạo, bài toán ước lượng tư thế người ngày càng phổ biến và được nghiên cứu và phát triển, ứng dụng rỗng rãi vào nhiều các hệ thống thông mình khác nhau như camera an ninh, thiết bị điện tử đeo tay, kính thực tế ảo, v.v. Thông thường, các mô hình học máy được xây dựng để giải quyết nhiệm vụ này sử dụng hình ảnh trên các hệ thống lớn đạt hiệu quả khá cao. Tuy nhiên, vấn đề bảo mật và chi phí cao lại là một thách thức lớn đối với bài toán ước lượng tư thế người sử dụng dạng dữ liệu này. Trong khi đó, với dữ liệu sóng radar, độ bảo mật thông tin được giải quyết dễ dàng hơn rất nhiều do tín hiệu sóng radar không có tính trừu tượng cao và chi phí cài đặt hệ thống radar cũng rẻ hơn nhiều so với cài đặt hệ thống sử dụng hình ảnh. Tuy nhiên, ước lượng tư thế người sử dụng tín hiệu radar vẫn đang phải đối mặt với ba thách thức lớn. Một là, sự thiết hụt các bộ dữ liệu có gán nhãn chất lượng cao khi chỉ có một số ít các nghiên cứu đống góp dữ liệu nhưng lại không được công bố công khai. Hai là, ít nghiên cứu về ước lượng tư thế người sử dụng radar được công bố và có tính xác thực cao. Hiện nay, phần lớn các bài toán ước lượng tư thế người sử dụng dữ liệu hình ảnh với số lượng lớn các tham số, mô hình năng và yêu càu tài nguyên tính toán lớn để có thể hoạt động tốt. Vì vậy, việc xây dựng một mô hình gọn nhẹ về mặt kích thước và tối ưu về thời gian chạy là một nhiệm vụ cần thiết để dễ dàng cài đặt, có khả năng hoạt động trên các thiết bị phần cứng hiệu suất thấp. Ba là, dữ liệu tín hiệu radar chứa rất nhiều nhiễu môi trường xung quanh. Điều này đòi hỏi quá trình tiền xử lý dữ liệu cầu kỳ và công phu.

Với tất cả những động lực trên, trong phạm vi kiến thức, đồ án sẽ trình bày thử nghiệm các mô hình học sâu với số lượng tham số ít sử dụng tín hiệu song radar cho nhiệm vụ ước lượng tư thế người.

Nội dung chi tiết của đồ án trình bày trong các chương sau, bao gồm:

- Chương 1: Tổng quan về các tác vụ cho bài toán ước lượng tư thế người sử dụng tín hiệu radar

  Nội dung chương 1 khái quát các về bài toán ước lượng tư thế người, mô tả tín hiệu radar và tính thực tiễn bài toán và nêu ra động lực, mục tiêu và phạm vi đồn án.

- Chương 2: Ước lượng tư thế người sử dụng tín hiệu radar bằng học sâu

Nội dung của chương 2 giới thiệu kiến thức cơ bản về trí tuệ nhân tạo và các mạng thần kinh học sâu cũng như các bước xây dựng mô hình ước lượng tư thế người sử dụng dữ liệu radar.

- Chương 3: Thực nghiệm và kết quả

  Nội dung của chương 3 trình bày quá trình về bộ dữ liệu, quá trình thu thập, mô tả phương pháp thực nhiệm và đánh giá các mô hình ước lượng tư thế người và trình bày các kết quả của quá trình thực nghiệm

CHƯƠNG 1: TỔNG QUAN VỀ CÁC TÁC VỤ CHO BÀI ƯỚC LƯỢNG TƯ THẾ NGƯỜI SỬ DỤNG TÍN HIỆU RADAR

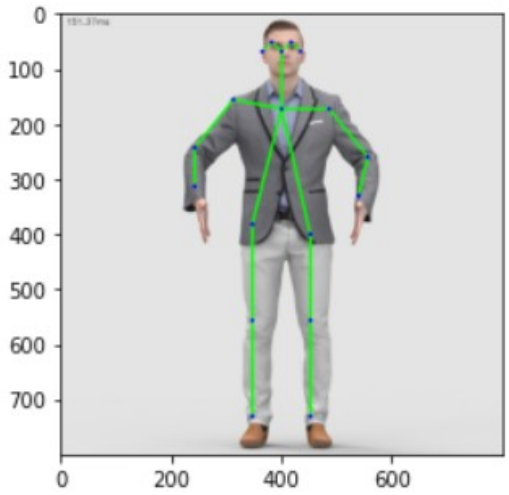
Trong chương 1, đồ án sẽ trình bày tổng quan về bài toán ước lượng tư thế người, đưa ra ý tưởng và hướng nghiên cứu cho đề tài, bao gồm các phần:

- Giới thiệu bài toán ước lượng tư thế người
- Phát biểu bài toán ước lượng tư thế người sử dụng tín hiệu radar
- Động lực và mục tiêu đồ án

1.1 Bài toán ước lượng tư thế người

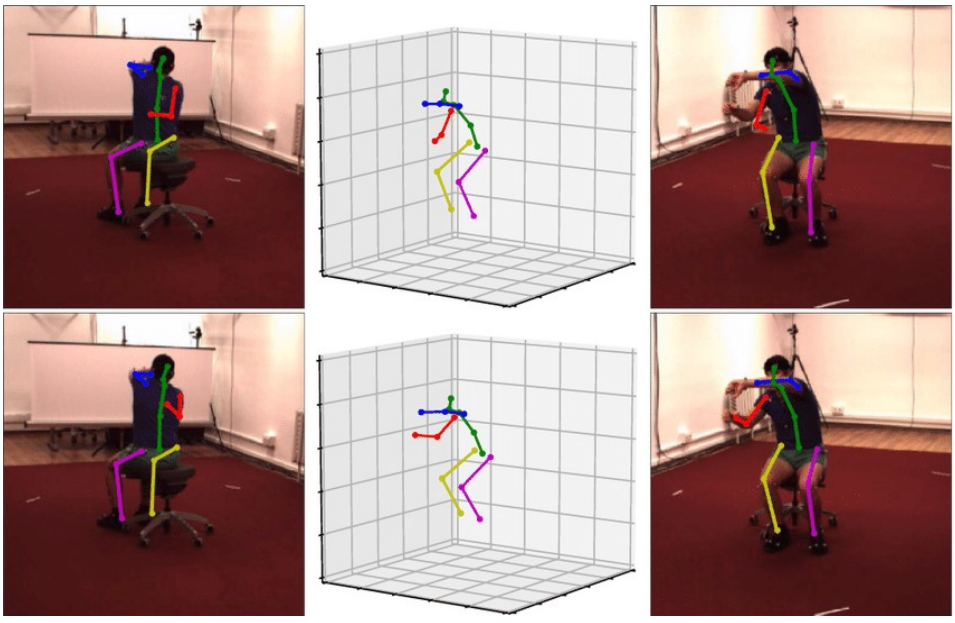
Bài toán ước lượng tư thế người (Human Pose Estimation) là một trong những thách thức quan trọng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Bài toán cũng là nền tảng quan trọng để phát triển các phát triển các giải pháp cho các bài toán phổ biến khác như phân loại hành vi con người dựa trên tư thế người, phát hiện hành động bất thường dựa trên tư thế người, v.v. Mục tiêu của bài toán này là xác định vị trí của các điểm quan trọng cụ thể trên cơ thể người thông qua dữ liệu hình ảnh, video, các giá trị của các cảm biến hay sóng phản xạ lại từ người. Các điểm này thường được gọi là các điểm khớp và thường bao gồm các vị trí quan trọng của đầu, vai, cổ, cổ tay, cổ chân, và các khớp khác trên cơ thể.

Bài toán ước lượng tư thế người được chia ra làm 2 loại: ước lượng tư thế người 2D và 3D. Đối với ước lượng tư thế người 2D, phương pháp tập trung chủ yếu vào không gian hình ảnh 2D cho phép xác định vị trí các điểm khớp trên mặt phẳng hình ảnh 2D (x,y). Phương pháp này giúp dễ triển khai và tính toán nhanh, thích hợp cho các ứng dụng đơn giản, không yêu cầu cao về tính 3D.



Hình 1.1 Hình ảnh ước lượng tư thế người 2D (Nguồn: analyticsvidhya.com).

Ngược lại, ước lượng tư thế người 3D bổ sung thêm thông tin chiều sâu vào quá trình ước lượng cho phép cung cấp vị trí của các điểm khớp trong không gian 3 chiều (x, y, z). Điều này giúp mô hình hiểu rõ hơn về vị trí thực tế của cơ thể người trên một hệ trục toạ độ nhất định nào đó và làm tăng độ chính xác, đặc biệt là trong các ứng dụng yêu cầu thông tin về chiều sâu, như thực tế ảo và nhận diện đối tượng 3D. Tuy nhiên, sự gia tăng về độ chính xác và thông tin về chiều sâu cũng sẽ làm độ phức tạp tính toán cao hơn.

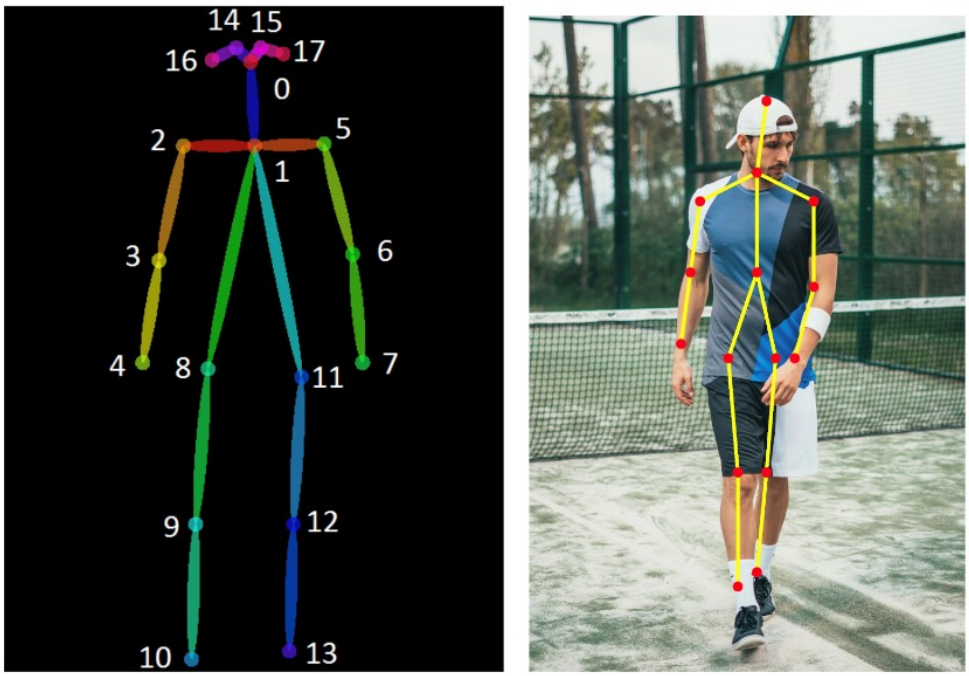


Hình 1.2 Hình ảnh ước lượng tư thế người 3D (Nguồn: [1]).

Dữ liệu ảnh và video, cùng với thông tin từ cảm biến Inertial Measurement Unit (IMU) và radar, đóng vai trò quan trọng trong việc ước lượng tư thế người. Mỗi loại dữ liệu mang đến những ưu điểm và thách thức riêng, tạo nên sự đa dạng và linh hoạt

trong các phương pháp tiếp cận bài toán. Dữ liệu từ ảnh và video cung cấp cái nhìn trực quan về tư thế người nhưng cũng đồng thời đặt ra những thách thức về xử lý hình ảnh và tính toán. Cùng lúc đó, dữ liệu từ cảm biến IMU, với khả năng đo lường chuyển động mạnh mẽ, mở ra khả năng theo dõi chính xác tư thế người trong không gian 3D. Trong khi đó, dữ liệu từ radar, mặc dù có độ phân giải không gian thấp hơn, nhưng lại đảm bảo hoạt động đáng tin cậy trong mọi điều kiện thời tiết. Các ưu và nhược điểm của từng loại dữ liệu sẽ được phân tích cụ thể như sau:

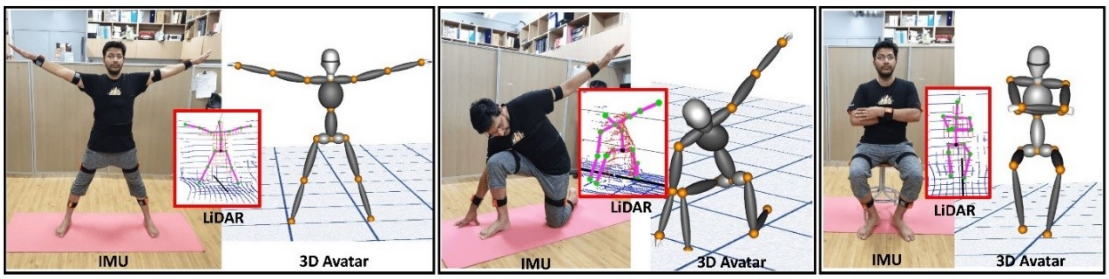
- Dữ liệu ảnh và video là một nguồn thông tin quan trọng trong ước lượng tư thế người, đây cũng là loại dữ liệu phổ biến nhất với đa dạng bộ dữ liệu có sẵn và các bài nghiên cứu khoa học như [4, 5, 6]. Bằng cách sử dụng hình ảnh hoặc video từ các camera, chúng ta có thể thu được thông tin chi tiết về tư thế và động tác của người trong một không gian 2 chiều hoặc 3 chiều. Các hệ thống sử dụng dữ liệu này thường đòi hỏi quá trình xử lý ảnh và video để nhận diện vị trí các điểm khớp trên cơ thể người và theo dõi chuyển động của người, mang lại khả năng hiểu biết và tương tác giữa người và máy hiệu quả. Tuy nhiên, dạng dữ liệu này phải đối mặt với thách thức từ ánh sáng, che mờ và độ phức tạp tính toán.



Hình 1.3 : Hình ảnh ước lượng tư thế người sử dụng dữ liệu hình ảnh (Nguồn: analyticsvidhya.com).

- Dữ liệu từ cảm biến Inertial Measurement Unit (IMU) cung cấp thông tin về gia tốc và tốc độ góc, giúp ước lượng chuyển động và tư thế của người một cách chính xác. Dữ liệu thường được sử dụng tích hợp với các loại dữ liệu khác để cải thiện độ chính xác của ước lượng tư thế người, đặc biệt là trong môi trường

không gian 3D. Tuy nhiên, dạng dữ liệu này cũng phải đối mặt với thách thức của sai số tích tụ theo thời gian và thiếu thông tin chiều sâu, khiến cho nó thường được kết hợp với dữ liệu từ nguồn khác để đạt được kết quả tốt nhất.



Hình 1.3: Hình ảnh ước lượng tư thế người sử dụng tín hiệu cảm biến (Nguồn: [2]).

- Cuối cùng, dữ liệu từ cảm biến radar cung cấp thông tin về vị trí và chuyển động của các đối tượng mục tiêu, bao gồm cả người. Mặc dù dữ liệu radar có độ phân giải không gian thấp hơn so với ảnh và video, nhưng nó có thể hoạt động hiệu quả mà không bị ảnh hưởng nhiều bởi các điều kiện thời tiết tiêu cực như mưa, tuyết, ánh sáng yếu và có khả năng xuyên qua các bề mặt có độ dày ít. Sự kết hợp của dữ liệu radar với các nguồn thông tin khác như ảnh và video có thể mang lại thông tin đa dạng và đầy đủ, đặc biệt là trong các ứng dụng đòi hỏi độ chính xác và tin cậy cao về ước lượng tư thế người.

Hình 1.4 Hình ảnh ước lượng tư thế người sử dụng tín hiệu radar (Nguồn: [3])

Quá trình ước lượng tư thế người đòi hỏi hệ thống phải hiểu được cấu trúc cơ bản của cơ thể người và tìm ra vị trí chính xác của các điểm khớp trong một bức ảnh. Điều này có ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

- Nhận dạng đối tượng: Trong hệ thống giám sát an ninh, ước lượng tư thế người có thể được sử dụng để nhận dạng và theo dõi người. Ví dụ bằng cách xác định vị

trí chính xác các điểm khớp trên cơ thể người, như đầu, vai, cổ, v.v,, giúp hệ thống nhận diện hành vi đe doạ tượng dễ dàng hơn, cải thiện độ chính xác.

- Thể thao và giáo dục: Trong lĩnh vực thể thao, ước lượng tư thế người cho phép theo dõi hoạt động và dáng thực hiện hành động của vận động viên. Từ đó cho phép đánh giá kỹ thuật thực hiện các bài tập, vận động tốt hơn.
- Ứng dụng y tế: Ước lượng tư thế người có thể được sử dụng để theo dõi tình trạng sức khỏe của bệnh nhân, đánh giá sự phục hồi của bệnh nhân sau chấn thương hoặc phẫu thuật. Bởi việc sử dụng hệ thống ước lượng tư thế người, các bác sĩ sẽ có thêm các góc nhìn để đánh giá độ phục hồi của bệnh nhân thông qua các hoạt động thường ngày.
- Tương tác giữa người và máy: Ví dụ trong lĩnh vực thực tế ảo, ước lượng tư thế người cho phép đưa ra chính xác tư thế của người dùng trong không gian 3D. Điều này giúp hệ thống hiểu được cách người dùng di chuyển, tương tác trong môi trường ảo giúp trải nghiệm thực tế hơn.

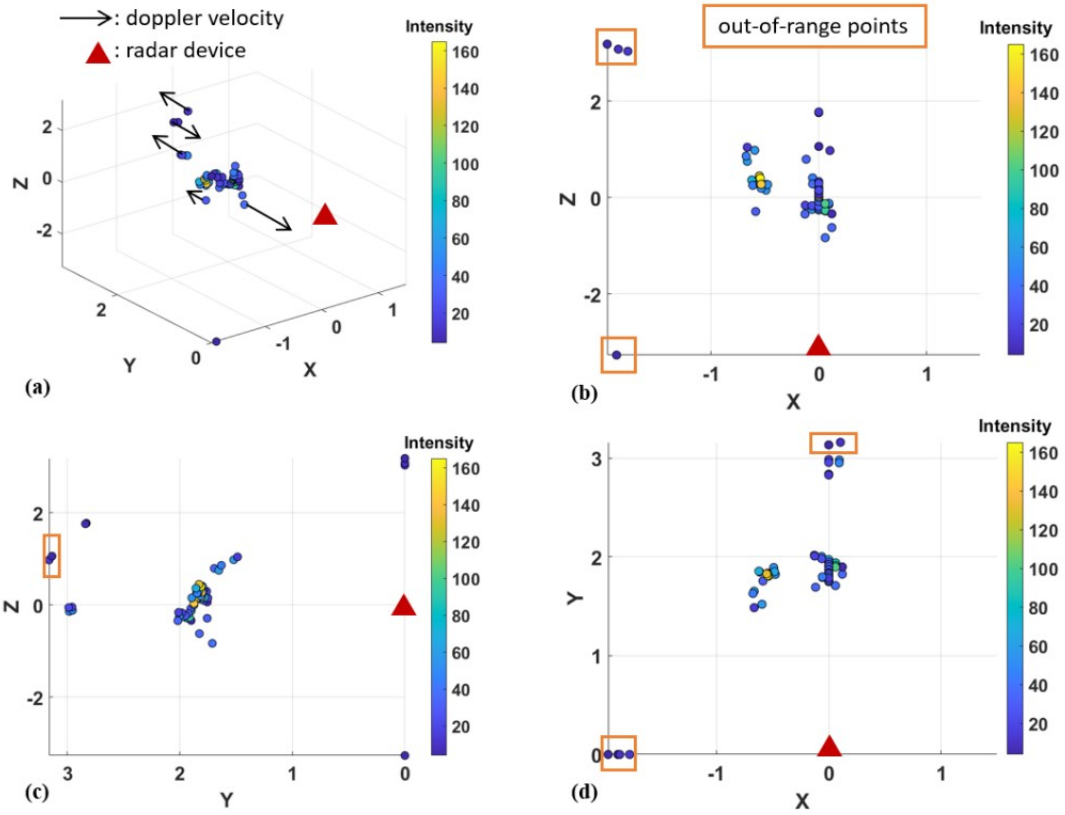
Mặc dù đã có sự tiến triển đáng kể trong lĩnh vực này, nhưng vẫn còn nhiều thách thức đối với human pose estimation, bao gồm độ chính xác, độ ổn định khi đối mặt với biến đổi hình thái và sự che mờ, cũng như khả năng ứng dụng thực tế.

1.2 Phát biểu bài toán ước lượng tư thế người sử dụng tín hiệu radar

Đầu vào bài toán là dữ liệu dạng point-cloud chứa thông tin về người trong 1 vùng không gian. Sau đó, hệ thống sẽ cần phải ước lượng chính xác tư thế người thông qua dạng dữ liệu point-cloud trên.

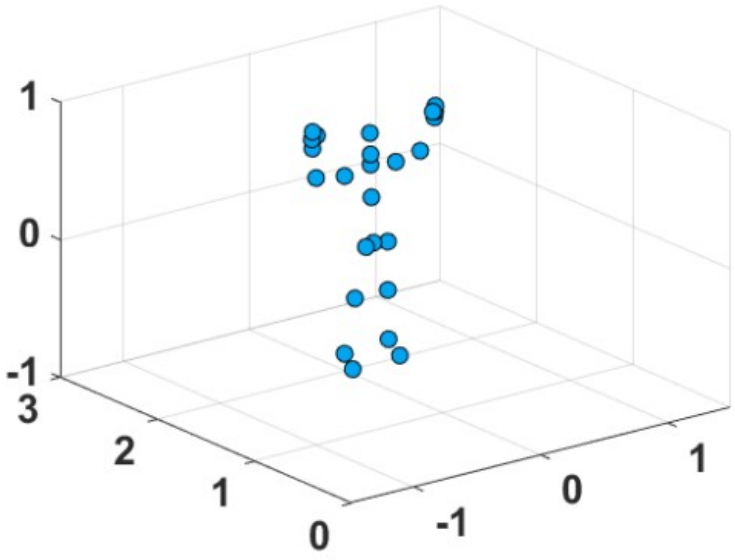
Ví dụ về quá trình xử lý trong bài toán:

- Đầu vào: Một tập hợp các điểm point-cloud chứa thông tin của người trong vùng không gian quan tâm



Hình 1.5 Hình ảnh mô phỏng các điểm point-cloud dữ liệu đầu vào trong không gian 3 chiều (Nguồn: [7])

- Đầu ra: Tập hợp các điểm khớp chứa thông tin tư thế người



Hình 1.6 Hình ảnh mô phỏng các điểm khớp chứa thông tin tư thế người trong không gian 3 chiều (Nguồn: [7])

1.3 Động lực và mục tiêu nghiên cứu

Động lực nghiên cứu

Với sự phát triển vượt bậc của công nghệ thông tin nói chung và lĩnh vực trí tuệ nhân tạo nói riêng, đã có rất nhiều nghiên cứu ứng dụng trí tuệ nhân tạo vào ước lượng tư thế người đã tiến hành. Tuy nhiên, với tính chất đặc thù về dữ liệu hình ảnh và video dễ dàng thu thập, có tính trừu tượng cao, các nghiên cứu trên tập trung chủ yếu xử lý tác vụ dựa trên loại dữ liệu này. Với việc đa dạng bộ dữ liệu hình ảnh và video, các nghiên cứu trên đã xử lý khá tốt tác vụ ước lượng tư thế người sử dụng hình ảnh hay video trong không gian 2D hay 3D cho kết quả cao. Ví dụ như mô hình Vertex heatmap autoencoder [8] cho ra kết quả khá ấn tượng với kết quả đánh giá cho phương pháp MPJPE là 15,6 mm và phương pháp PA-MPJPE là 21.9 mm. Tuy nhiên, với bản chất các mô hình trên cần rất nhiều dữ liệu có sẵn, và đây cũng là rào cản khi phải thu thập lượng lớn dữ liệu hình ảnh với lượng lưu trữ dữ liệu khổng lồ. Ngoài ra, cũng có thể dễ thấy rằng đa phần các bộ dữ liệu hình ảnh được công bố đều được ghi lại trong điều kiện tốt nhất định nào đó như đủ điều kiện ánh sáng, không bị ảnh hưởng bởi các yếu tố môi trường như mưa, bão, hay tuyết. Vậy nên, để thực thi được tác vụ ước lượng tư thế người thực tế đòi hỏi quá trình tiền xử lý dữ liệu tốt và cầu kỳ. Thêm vào đó, việc lắp đặt các thiết bị camera cung cấp hình ảnh sắc nét cũng đòi hỏi chi phí cao.

Chính vì vậy, đề tài này sẽ đưa ra một hướng tiếp cận cho bài toán ước lượng tư thế người sử dụng tín hiệu radar thay vì sử dụng dữ liệu hình ảnh hay video. Với mục tiêu giải quyết các vấn đề tồn đọng trong việc sử dụng hình ảnh và video. Về chi phí cài đặt thực tế, các hệ thống radar có chi phí lắp đặt thấp hơn so với việc sử dụng camera. Ngoài ra, tín hiệu camera cũng cho phép hệ thống hoạt động tốt trong các điều kiện môi trường khác nhau như có ánh sáng, không có ánh sáng hay trong môi trường điều kiện thời tiết xấu như mưa, bão, có tuyết. Thêm vào đó, tín hiệu radar cũng được xử lý dưới dạng point-cloud cho phép lưu lại thông tin có giá trị và chiếm ít không gian bộ nhớ.

Mặt khác, với mong muốn ước lượng tư thế người sử dụng tín hiệu radar cho kết quả hiệu năng tốt, đề tài được kỳ vọng sẽ là nền tảng để phát triển cho các ứng dụng thực tiễn về sau như phát hiện người ngã trong bệnh viện, hay phát hiện người xâm nhập trái phép vào khu vực cấm trong mọi loại điều kiện môi trường. Ngoài ra, ước lượng tư thế người sử dụng tín hiệu radar cũng cung cấp cái nhìn trực quan hơn về khả năng liên kết giữa tín hiệu radar thu được và tư thế người để từ đó có các nghiên cứu chuyên sâu hơn.

Mục tiêu nghiên cứu

Đồ án này có hai mục tiêu chính

- Đưa ra hệ thống hoàn chỉnh nhận dữ liệu radar qua quá trình tiền xử lý để ước lượng tư thế người.
- Đề ra các phương pháp sử dụng học sâu cho bài toán ước lượng tư thế người sử dụng tín hiệu radar.

1.4 Phạm vi đồ án

Nội dung đồ án sẽ tập trung vào việc xây dựng xây dựng một quá trình tiền xử lý dữ liệu radar hoàn chỉnh. Tinh chỉnh và huấn luyện các mô hình mạng trí tuệ nhân tạo để ước lượng tư thế người. Từ đó, đưa ra đánh giá mô hình dựa trên các phương pháp đánh giá MAE, MPJPE, PA-MPJPE và đưa ra những hình ảnh kết quả của mô hình.

1.5 Tổng kết chương 1

Trong chương 1, đồ án đã trình bày tổng quan về bài toán ước lượng tư thế người, dạng dữ liệu sẽ được sử dụng cho bài toán, động lực và mục tiêu của đồ án, phạm vi đồ án.

Trong chương tiếp theo, đồ án sẽ trình bày về phương pháp ước lượng tư thế người, tổng quan về học sâu, chi tiết về mô hình mạng nơ-ron tích chập 2 chiều, mô hình mạng theo thời gian, mô hình mã hoá tự động cho bài toán ước lượng tư thế người. Điểm mạnh và điểm yếu của các phương pháp.

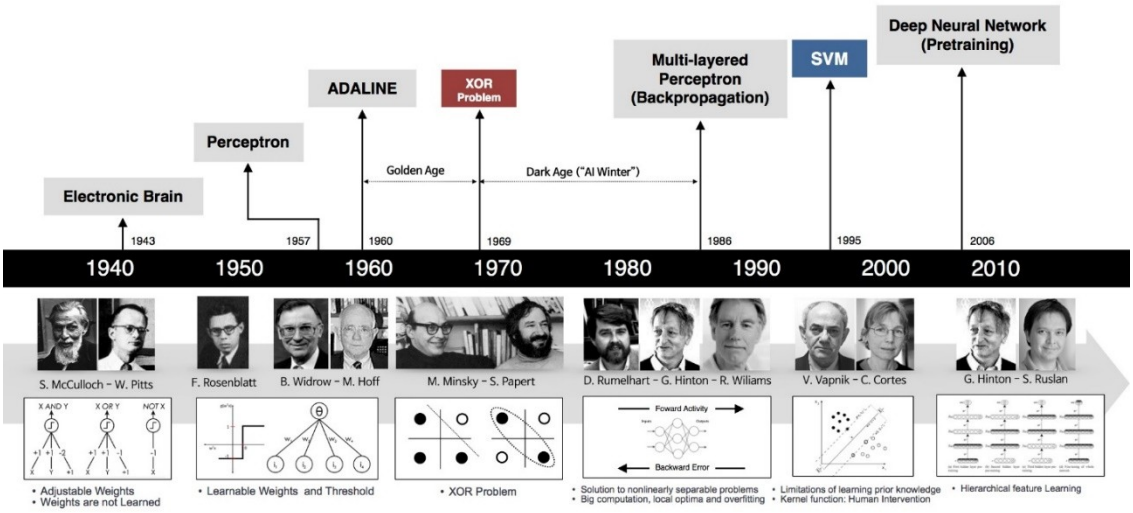
CHƯƠNG 2: ƯỚC LƯỢNG TƯ THẾ NGƯỜI SỬ DỤNG TÍN HIỆU RADAR BẰNG HỌC SÂU

Trong chương này, đồ án sẽ trình bày nội dung chi tiết về cách tiếp cận bài toán sử dựng phương pháp cổ điển và cách áp dụng mô hình học sâu vào bài toán ước lượng tư thế người qua các phần:

- Giới thiệu về trí tuệ nhân tạo
- Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN)
- Mạng nơ-ron tích chập (Convolution Neural Network - CNN)
- Mạng nơ-ron trí nhớ ngắn hạn dài (Long Short-Term Memory - LSTM)
- Đề xuất mô hình hình học sâu cho bài toán

2.1 Giới thiệu về trí tuệ nhân tạo

2.1.1 Trí tuệ nhân tạo



Hình 2.7 Sơ đồ tổng quan về trí tuệ nhân tạo (Nguồn: machinlearningcoban).

Trí tuệ nhân tạo (AI) [9] là một lĩnh vực nghiên cứu trong ngành công nghiệp công nghệ máy tính với mục tiêu tạo ra các hệ thống máy tính có khả năng thực hiện các nhiệm vụ đòi hỏi sự hiểu biết, học hỏi, và ra quyết định mà trước đây chỉ có con người mới có khả năng thực hiện được. Trí tuệ nhân tạo mô phỏng khả năng "tư duy" của con người thông qua việc phát triển và triển khai các thuật toán máy học, học sâu, và các phương pháp khác.

Lịch sử của trí tuệ nhân tạo bắt đầu từ những ý tưởng sơ khai từ thập kỷ 1940, nhưng nó đã phát triển mạnh mẽ trong những năm gần đây nhờ vào sự tiến bộ đáng kể

trong lĩnh vực máy học và nguồn dữ liệu lớn có sẵn. Một số thành tựu nổi bật của trí tuệ nhân tạo bao gồm việc nhận diện giọng nói, nhận diện hình ảnh, tự động lái xe, chơi cờ vua ở mức độ chuyên nghiệp, v.v.

Các ứng dụng của trí tuệ nhân tạo ngày càng mở rộng trên hầu hết các lĩnh vực như y tế, tài chính, giáo dục, và dự báo thời tiết, v.v. Nó đã và đang thay đổi cách chúng ta tương tác với công nghệ, cung cấp giải pháp thông minh cho các vấn đề phức tạp, và tạo ra những tiện ích hàng ngày như trợ lý ảo, hệ thống tự động, và nhiều ứng dụng khác.

2.1.2 Học máy

Học máy (Machine Learning - ML) [10] là một lĩnh vực trong trí tuệ nhân tạo (AI) mà mục tiêu là phát triển các mô hình và thuật toán có khả năng học hỏi từ dữ liệu mà không cần phải được lập trình một cách cụ thể. Ý tưởng cơ bản của học máy là xây dựng các mô hình có khả năng tự điều chỉnh và cải thiện hiệu suất thông qua trải nghiệm. Arthur Samuel đã mô tả học máy là một "lĩnh vực nghiên cứu mang lại cho máy tính khả năng học hỏi cách giải quyết vấn đề dựa trên dữ liệu đưa vào mà không cần được lập trình rõ ràng" [11]. Còn Tom Mitchell lại đưa ra một định nghĩa mang tính rõ ràng hơn về học máy [12]: "Một chương trình máy tính được cho là học từ kinh nghiệm E đối với một số loại nhiệm vụ T và thước đo hiệu suất P, nếu hiệu suất của nó ở các nhiệm vụ T, được đo bằng P, cải thiện theo kinh nghiệm E".

Cơ bản, ý tưởng của học máy là tạo ra các mô hình máy tính có khả năng tự điều chỉnh dựa trên kinh nghiệm và dữ liệu đã được cung cấp. Một điểm quan trọng là học máy không đơn thuần là việc áp dụng một bộ quy tắc lập trình, mà là quá trình tìm kiếm và ánh xạ các mối liên kết, quy luật, và biểu diễn từ dữ liệu để đưa ra dự đoán và quyết định cho dữ liệu mới.

Có bốn loại chính của học máy:

1. Học máy giám sát (Supervised Learning): Trong loại này, mô hình được huấn luyện trên một tập dữ liệu có nhãn, nghĩa là mỗi mẫu dữ liệu đã được gán nhãn với đầu ra mong muốn. Mục tiêu là học cách ánh xạ từ dữ liệu đầu vào đến các nhãn đầu ra để sau đó có thể tự động dự đoán đầu ra cho các mẫu mới mà không cần nhãn.
2. Học máy không giám sát (Unsupervised Learning): Trong loại này, mô hình được huấn luyện trên một tập dữ liệu không có nhãn. Mục tiêu chính của học máy không giám sát là tìm ra cấu trúc, mối quan hệ, hoặc phân nhóm dữ liệu trên toàn bộ tập dữ liệu mà không cần biết đầu ra mong muốn trước.

3. Học máy bán giám sát (Semi-supervised Learning): Là một hình thức trung gian giữa học máy giám sát và học máy không giám sát, loại này sử dụng một tập dữ liệu lớn không có nhãn và một số ít dữ liệu có nhãn để huấn luyện mô hình.
4. Học tăng cường (Reinforment Learning): Là một hình thức mà máy tính được thiết kế để tìm hiểu và cải thiện hiệu suất của mô hình thông qua trải nghiệm và tương tác với môi trường. Mục tiêu của mô hình là tối ưu hoá giá trị nhận được (reward) tại mỗi hành động mà mô hình thực hiện trong một môi trường nhất định nào đó. Khác với học máy truyền thống, học tăng cường không chỉ tập trung vào việc giải quyết các vấn đề cụ thể, mà còn đề cao khả năng học và ra quyết định tối ưu trong môi trường động và không chắc chắn.

Trong suốt thời gian phát triển, học máy đã đạt được nhiều thành công trong nhiều ứng dụng thực tế như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, dự đoán chuỗi thời gian, v.v. Nói cách khác, học máy đã làm thay đổi cách chúng ta xử lý thông tin và ra quyết định, mang lại sự tự động hóa và hiệu suất đáng kể. Tuy nhiên, logic tuyến tính giữa đầu vào và đầu ra trong học máy là cản trở để phương pháp này hoạt động tốt với các loại dữ liệu phức tạp, đòi hỏi tính phi tuyến.

2.1.3 Học sâu

Học sâu [13] (Deep Learning - DL) là một tập con của học máy trong lĩnh vực trí tuệ nhân tạo. Học sâu tập trung vào phát triển và huấn luyện các mô hình máy học sâu có khả năng tự động học và thực hiện các nhiệm vụ phức tạp có tính chất phi tuyến. Các mô hình này, hay còn được gọi là mạng nơ-ron nhân tạo, được thiết kế với nhiều lớp thần kinh như bộ não con người. Qua quá trình huấn luyện, mạng nơ-ron dần dần đạt được kinh nghiệm thông qua hàm đánh giá, có khả năng lưu giữ kinh nghiệm học được và sử dụng để thực hiện dự đoán cho các dữ liệu chưa biết sau này.

2.2 Mạng nơ-ron nhân tạo (Artificial Neural Netowork - ANN)

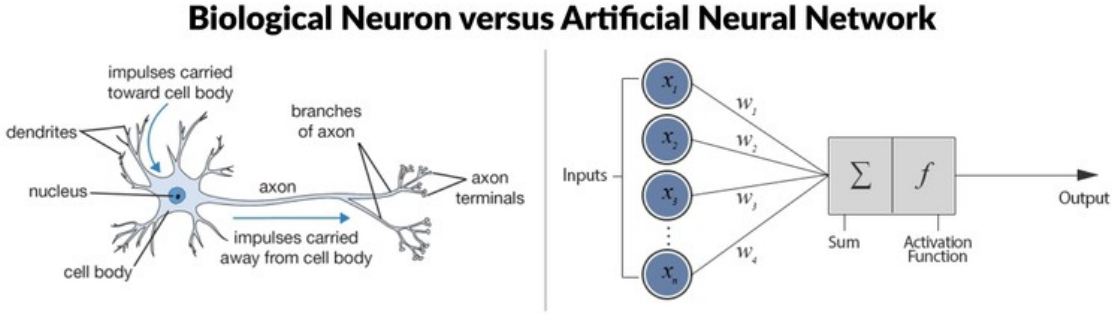
Mạng nơ-ron nhân tạo, hay Artificial Neural Network [14] (ANN), là khái niệm quan trọng và là nền tảng cho các kiến trúc mạng sau này, được thiết kế để mô phỏng cách hoạt động của não người. Với sự linh hoạt và khả năng tự động hóa của ANN làm cho chúng trở thành công cụ quan trọng trong giải quyết những vấn đề phức tạp và đưa ra dự đoán chính xác trong nhiều bài toán hiện nay. ANN có khả năng học từ kinh nghiệm và dữ liệu (thông qua quá trình huấn luyện), có thể áp dụng cho nhiều ứng dụng trong các lĩnh vực khác nhau có thể kể đến như:

- Nhận diện hình ảnh: Nhận dạng chữ viết tay, nhận dạng khuôn mặt, phân biệt giống chó mèo, v.v.

- Xử lý ngôn ngữ tự nhiên: Nhận diện giọng nói, thực hiện thao tác lệnh từ giọng nói, v.v.
- Dự đoán: Dự đoán giá nhà, dự đoán giá cổ phiếu, .v.v.
- Tự động hoá: Điều khiển xe tự lái, máy bay không người lái, v.v.
- Phát hiện dị thường: Phát hiện giao dịch giả mạo trong lĩnh vực ngân hàng, v.v.

2.2.1 Tổng quan mạng nơ-ron nhân tạo

Mạng nơ-ron nhân tạo mô phỏng lại cấu trúc của bộ não con người bao gồm các lớp nơ-ron được liên kết với nhau thông qua các cạnh, trong mỗi lớp nơ-ron gồm có các nơ-ron nhân tạo đại diện cho lớp đó. Số lượng lớp nơ-ron và số lượng nơ-ron trong mỗi lớp phụ thuộc vào cách thiết kế với từng nhiệm vụ cụ thể nào đó.



Hình 2.8 Mô tả một nơ-ron sinh học và một nơ-ron nhân tạo (Nguồn: linkedin.com)

Để làm rõ hơn, tương tự như cách hoạt động của 1 mạng nơ-ron sinh học, mạng nơ-ron nhân tạo cũng nhận dữ liệu đầu vào (input) bao gồm các giá trị $x_1, x_2, x_3, \ldots, x_n$, xử lý tín hiệu bằng cách nhân dữ liệu này với trọng số tương ứng $w_1, w_2, w_3, \ldots, w_n$, còn được gọi là weight, tính tổng thông qua hàm sum sau đó đi qua 1 hàm kích hoạt (activation function) để phá vỡ liên kết tuyến tính và thu được kết quả đầu ra (output).

Ở quá trình trên, giá trị trọng số weight đóng 1 vai trò quan trọng. Nó quyết định mức độ ảnh hưởng ít hay nhiều của 1 giá trị đầu vào x nào đó, thể hiện mức độ quan trọng của điểm dữ liệu đó với kết quả đầu ra. Ngoài ra, hàm kích hoạt cũng đóng một vai trò quan trọng không kém cái mà sẽ quyết định cách thức mô hình hoá phi tuyến giữa dữ liệu đầu vào và kết quả đầu ra.

2.2.2 Cấu trúc của một mạng nơ-ron nhân tạo đơn giản

Một mô hình mạng nơ-ron nhân tạo gồm nhiều lớp nơ-ron liên kết với nhau, mỗi lớp nơ-ron bao gồm nhiều nơ-ron. Mỗi nơ-ron tại một lớp bất kì đều liên kết đến tất cả nơ-ron ở lớp trước và lớp sau nó. Một mạng nơ-ron đơn giản gồm ba lớp chinh: lớp đầu vào, lớp đầu ra và lớp ẩn. Lớp đầu vào chính là lớp đầu tiên của mạng nơ-ron, với

số nơ-ron bằng số đặc trưng của dữ liệu, lớp đầu ra là số nơ-ron đầu ra mong muốn với từng loại tác vụ. Các lớp nằm giữa hai lớp trên được gọi chung là lớp ẩn. Logistic Regression [15] là một ví dụ điển hình cho một mạng nơ-ron nhân tạo đơn giản với một lớp đầu vào, một lớp đầu ra và không có lớp ẩn. Hàm mục tiêu của mô hình Logistic Regression là hàm bậc một nhiều biến:

$$\hat{y}=\sigma(w_0+w_1x_1+w_2x_2+\cdots+w_nx_n)$$

Hàm mục tiêu được tính toán như sau:

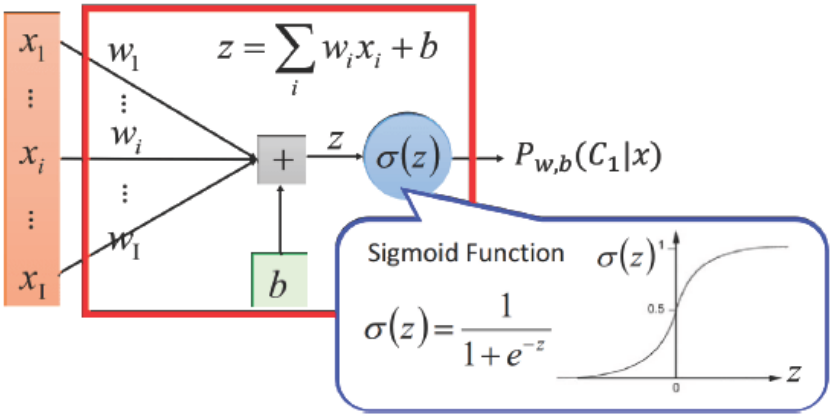
1. Tính tổng tuyến tính

$$z=w_0+w_1x_1+w_2x_2+\cdots+w_nx_n$$

2. Áp dùng hàm kích hoạt – Sigmoid

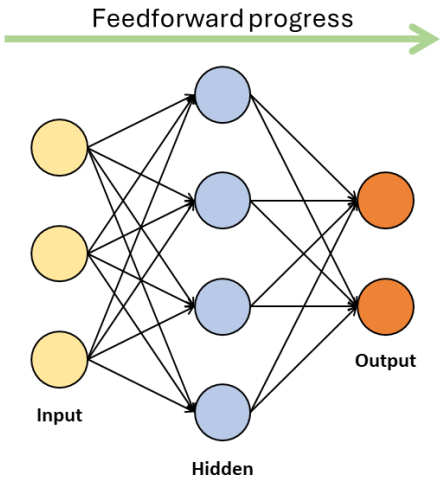
$$\hat{y}=\sigma(z), \text{ với } \sigma(z)=\frac{1}{1+e^{-z}}$$

Phương trình ở trên có thêm w_0 được gọi là hệ số tự do. Do phương trình $w_1x_1+w_2x_2+\cdots+w_nx_n=0$ sẽ luôn đi qua gốc toạ độ gây ra mô hình thiếu đi sự tổng quát, dẫn đến nhiều trường hợp phương trình không tìm được hàm mục tiêu mong muốn. Việc thêm hệ số tự do (w_0) cho phép mô hình linh hoạt hơn trong việc tìm kiếm hàm mục tiêu. Mô hình Logistic Regression được mình hoạ như hình 2.3 dưới đây.



Hình 2.9 Minh hoạ kiến trúc của mạng nơ-ron đơn giản Logistic regression (Nguồn: Medium.com)

2.2.3 Lan truyền thẳng (Feedforward)



Hình 2.10 Minh hoạ cách thức lan truyền thẳng giữa các nơ-ron trong mạng trí tuệ nhân tạo đơn giản.

Lan truyền thẳng (Feedforward) trong mạng trí tuệ nhân tạo là quá trình dữ liệu đầu vào được truyền qua các lớp nơ-ron bên trong mạng, qua quá trình tính toán, xử lý tại mỗi nơ-ron tại từng lớp và đưa ra kết quả cuối cùng tại lớp nơ-ron cuối cùng mà không có sự giám sát hoặc điều chỉnh bên ngoài tác động vào. Đây là quá trình quan trọng của một mạng nơ-ron để tạo ra đầu ra hoặc dự đoán dựa trên dữ liệu đầu vào một cách tự động.

Quá trình lan truyền thẳng bắt đầu từ lớp đầu vào, trong đó mỗi nút (nơ-ron) đại diện cho một thuộc tính của dữ liệu đầu vào. Dữ liệu được nhân với trọng số, cộng với hệ số tự do, tính tổng lại và cho qua một hàm kích hoạt với mục đích loại bỏ tính chất tuyến tính của dữ liệu để tạo ra dữ liệu đầu ra tại mỗi nút (nơ-ron) cho lớp tiếp theo. Quá trình này được lặp lại qua từng lớp nơ-ron cho đến lớp nơ-ron đầu ra cuối cùng.

Về mặt toán học, ta kí hiệu i là lớp nơ-ron thứ i trong mạng nơ-ron nhân tạo, tại mỗi lớp nơ-ron thứ i có l nút nơ-ron. Vậy nên ra có l^i là lớp nơ-ron thứ i với l nút nơ-ron. $W^k \in R^{l^{k-1} \times l^k}$ là ma trận trọng số giữa lớp thứ $k-1$ và lớp thứ k, trong đó $w_{i.j}^k$ là trọng số kết nối từ nút thứ i của lớp nơ-ron thứ $k-1$ đến nút thứ j của lớp nơ-ron thứ k. Vector $b^k \in R^{l^k}$ là vector chứa hệ số tự do của các lớp trong lớp thứ k, trong đó b_i^k là hệ số tự do của đơn vị thứ i trong lớp thứ k.

Xét giá trị tại nút thứ i trong lớp l như sau:

- Nhân với trọng số, cộng với hệ số tự do và tính tổng tuyến tính:

$$z_i^l = \sum_{j=1}^{l^{l-1}} a_j^{l-1} w_{ji}^l + b_j^l$$

- Áp dụng hàm kích hoạt:

$$a_i^l = \sigma(z_i^l)$$

Trong đó $a_i^0 = x_i$

Xét giá trị tại lớp thứ i biểu diễn dưới dạng ma trận và véc-tơ:

- Nhân với trọng số, cộng với hệ số tự do và tính tổng tuyến tính:

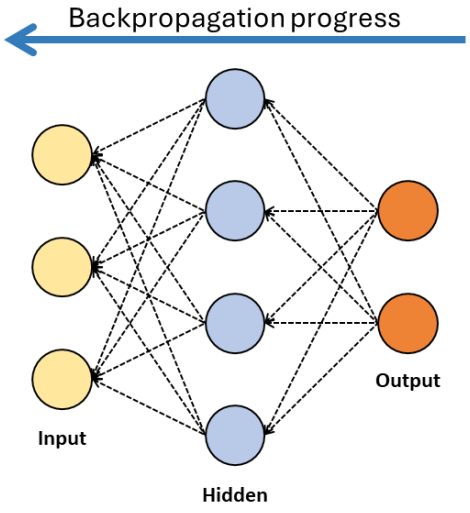
$$z^i = a^{i-1} W^i + b^i$$

- Áp dụng hàm kích hoạt:

$$a^i = \sigma(z^i)$$

Trong đó a^i là véc-tơ đầu ra của lớp thứ i với $a^i = x$

2.2.3. Lan truyền ngược và đạo hàm



Hình 2.11 Minh hoạ cách thức lan truyền ngược giữa các nơ-ron trong mạng trí tuệ nhân tạo đơn giản.

Không giống như lan truyền thẳng – quá trình tính toán dữ liệu đầu vào để đưa ra kết quả đầu ra hoặc dự đoán, lan truyền ngược là quá trình tinh giúp mạng nơ-ron nhân tạo tự động tính chỉnh lại các giá trị trọng số và các hệ số tự do thông qua quá trình

huấn luyện và giá trị trả về của hàm mất mát. Sau khi nhận được giá trị dự đoán thông qua quá trình lan truyền thẳng, các kết quả dự đoán được so sánh với giá trị mong muốn thực tế để từ đó tính ra sai số thông qua một hàm mất mát được đĩnh nghĩa sẵn. Sau đó, hàm mất mát được lan truyền ngược từ lớp nơ-ron đầu ra về lớp nơ-ron đầu vào để tính toán đạo hàm theo từng trọng số và hệ số tự do trong mạng nơ-ron. Các giá trị trọng số và hệ số tự do được cập nhật thông qua phương pháp tối ưu gradient descent để kích thích các giá trị đạo hàm dần dần về hướng hội tụ.

Với bài toán Logistic Regression, hàm mất mát 2.8 được sử dụng:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

Với N là số điểm dữ liệu trong tập dữ liệu, $\hat{y}$ là véc-tơ đầu ra của mạng nơ-ron nhân tạo.

2.2.4 Hàm kích hoạt

Hàm kích hoạt là một yếu tố quan trọng quyết định sự linh hoạt và hiệu suất của mô hình. Khái niệm này không chỉ dừng lại ở việc mô phỏng lại các tính chất của não người mà còn đóng vai trò quan trọng trong việc đưa ra dự đoán (lan truyền xuôi) và quá trình tự học (lan truyền ngược). Số lượng hàm kích hoạt rất đa dạng, có thể là hàm tuyến tính hoặc phi tuyến, điều này có ảnh hưởng quan trọng đến việc mô hình có thể học được các đặc trưng phức tạp từ dữ liệu hay không.

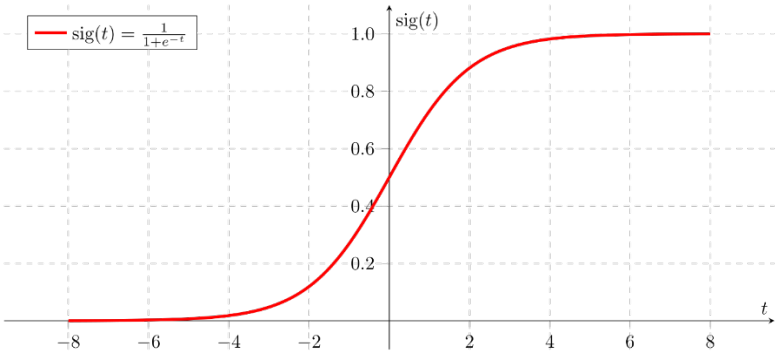
Trong bài toán Logistic Regression, tác giả sử dụng hàm kích hoạt Sigmoid với mục đích hạn chế giá trị đầu ra của nơ-ron trong khoảng $[0,1]$. Điều này rất quan trọng vì đầu vào cho hàm kích hoạt là đầu ra của hàm tổng $w \times x + b$, giá trị này nếu không bị giới hạn có thể tăng rất cao, thậm chí tiến tới ∞ trong một số trường hợp. Điều này sẽ gây ra các vấn đề tính toán như bùng nổ đạo hàm.

Đặc biệt, khi đề cập đến hàm kích hoạt, tính năng quan trọng nhất của nó là khả năng phi tuyến tính hoá mặng nơ-ron nhân tạo. Nếu không có tính chất này, mạng nơ-ron nhân tạo lúc này sẽ trở thành một hàm hay tập hợp các hàm tuyến tính đơn giản và không có khả năng học các bộ dữ liệu phức tạp như hình ảnh, âm thanh, v.v. Việc lựa chọn hàm kích hoạt phụ thuộc vào từng bài toán và kinh nghiệm của người thiết kế mạng nơ-ron. Dưới đây là một số hàm kích hoạt phổ biến thường được sử dụng trong các mạng nơ-ron nhân tạo:

Sigmoid

Công thức toán học:

$$sigmoid(z)=\frac{1}{1+e^{-t}}$$



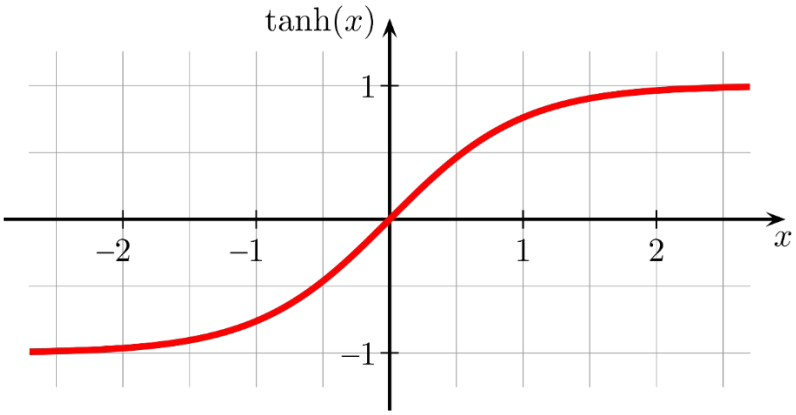
Hình 2.12 Công thức và đồ thị hàm Sigmoid (Nguồn: Cloud2data.com).

Đây là hàm phi tuyến giới hạn đầu ra trong khoảng giá trị $[0,1]$. Hàm này ít khi được sử dụng trong các mô hình thực tế do sự tốn kém về mặt tính toán, cũng như dễ gây ra các vấn đề về đạo hàm tiến gần tới 0. Hàm kích hoạt này thường được sử dụng cho các bài toán phân loại nhị phân và hay được sử dụng làm hàm kích hoạt mẫu cho người mới tiếp cận trí tuệ nhân tạo.

Tanh

Công thức toán học:

$$\tanh(z)=\frac{e^{z}-e^{-z}}{e^{z}+e^{-1}}$$



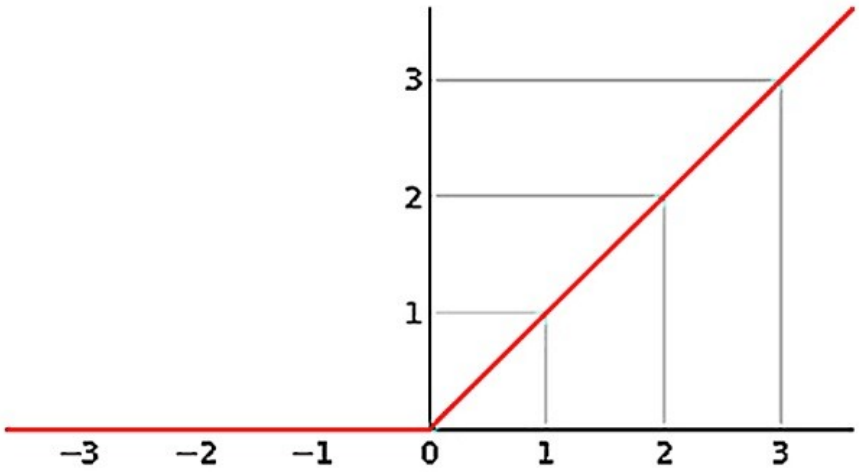
Hình 2.13 Đồ thị hàm Tanh (Nguồn: Wikipedia).

Đây là hàm phi tuyến giới hạn đầu ra trong khoảng giá trị $[-1,1]$. Hàm kích hoạt tanh cũng có nhược điểm tương tự với hàm sigmoid() khi giá trị đạo hàm là rất nhỏ với các đầu vào có trị tuyệt đối lớn, gây ra tốn kém về mặt tính toán.

ReLU

Công thúc toán học:

$$ReLU(z)=max(0,z)$$



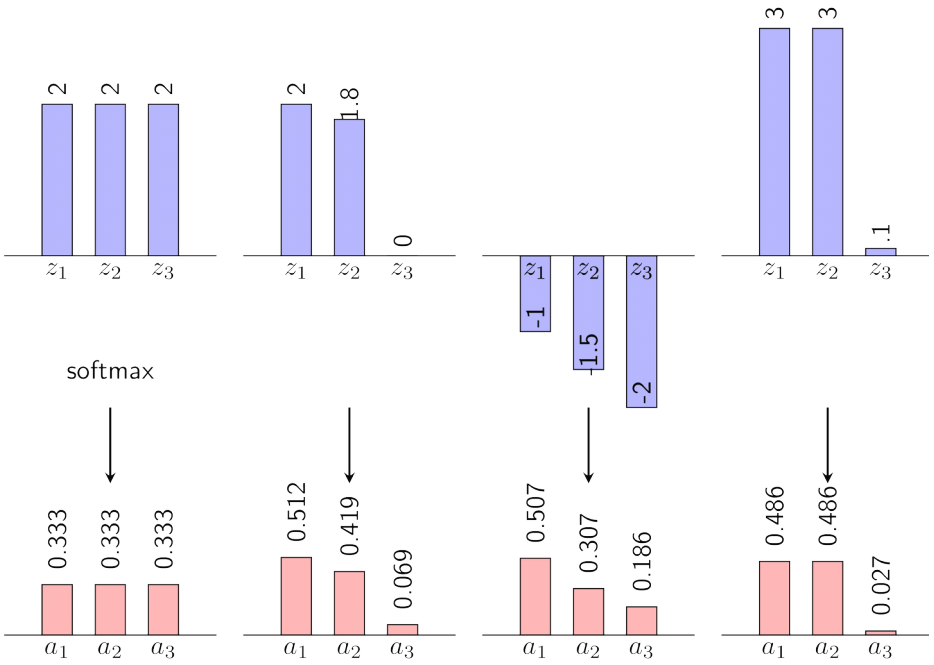
Hình 2.14 Đồ thị hàm ReLU (Nguồn: Researchgate.net).

Đây là hàm phi tuyến giới hạn đầu ra trong khoảng giá trị $[0,+\infty]$. Hàm kích hoạt ReLU (Rectified Linear Unit) đóng vai trò quan trọng trong lĩnh vực mạng trí tuệ nhân tạo, với tính chất đơn giản nhưng mạnh mẽ. Hàm trả về giá trị 0 nếu đầu vào là âm và giữ nguyên giá trị dương nếu đầu vào là dương. Tính chất phi tuyến tính của ReLU giúp mô hình có khả năng học được các biểu diễn phức tạp từ dữ liệu và giảm vấn đề "vanishing gradient" trong quá trình lan truyền ngược. Điều này giúp tăng cường khả năng hiệu quả và tốc độ học của mô hình. Mặc dù ReLU có nhược điểm là có thể dẫn đến "dead neurons" khi các trọng số được cập nhật sao cho đầu vào của neuron trở nên âm gây ra hiện tượng giá trị đầu ra bằng 0, và nó chỉ hoạt động cho các giá trị dương, nhưng tính đơn giản khiến ReLU trở thành lựa chọn ưa thích trong nhiều kiến trúc mô hình hiện đại. Ngoài ra, để giải quyết nhược điểm của ReLU, một số biến thể khác của nó đã được tận dụng như Leaky ReLU hay Parametric ReLU.

Softmax

Công thức toán học:

$$softmax(x_i)=\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$



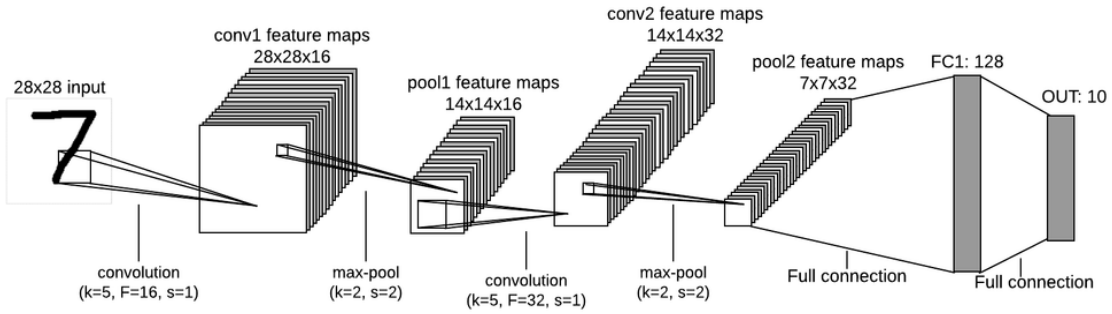
Hình 2.15 Minh hoạ kết quả sau khi đi qua hàm Softmax (Nguồn: Machinelearningcoban)

Hàm softmax là một dạng tổng hợp hơn của sigmoid, thường được sử dụng trong các bài toán phân loại nhiều lớp. Tương tự như hàm sigmoid, mỗi giá trị đầu ra của hàm softmax đều nằm trong khoảng $[0,1]$ với tổng các giá trị đầu ra luôn luôn bằng 1. Điều này giúp tạo ra một phân phối xác suất diễn giải được, làm cho mô hình không quá chắc chắn khi đưa ra dự đoán, tránh trường mô hình bị hiện tượng overfit vào một nhãn phân loại nào đó.

2.3. Mạng nơ-ron tích chập (Convolution Neural Network - CNN)

Mạng nơ-ron tích chập [13] (CNN) thể hiện sự tương đồng với mạng nơ-ron nhân tạo thông thường, bao gồm nơ-ron có khả năng học trọng số và hệ số tự do. Mỗi nơ-ron trong CNN nhận đầu vào, thực hiện phép tích vô hướng và có thể được kích hoạt bằng hàm phi tuyến tùy chọn, với hàm Softmax thường được sử dụng ở lớp cuối cùng. Cả hai loại mạng chia sẻ các kỹ thuật và thủ thuật trong quá trình huấn luyện. Tuy nhiên, điểm đặc biệt là CNN tiếp cận dữ liệu ảnh hoặc mảng nhiều chiều mà không cần chuyển đổi thành véc-tơ một chiều như mạng nơ-ron truyền thống. Điều này cho phép CNN trích xuất thuộc tính đặc trưng, đặc biệt là các thuộc tính địa phương, mà không làm mất cấu trúc dữ liệu. So với mạng nơ-ron truyền thống, CNN giúp giảm đáng kể số lượng tham số tính toán, tối ưu hóa tài nguyên tính toán và giảm nguy cơ overfitting. Thiết kế của CNN nhằm khắc phục nhược điểm của mạng nơ-ron truyền thống trong xử lý ảnh, mang lại hiệu suất chuyển tiếp tốt hơn và mô hình với số lượng tham số giảm đáng kể.

2.3.1. Kiến trúc của mạng nơ-ron tích chập



Hình 2.16 Mô hình cấu tạo các lớp của một mạng CNN (Nguồn: Easy-tensorflow).

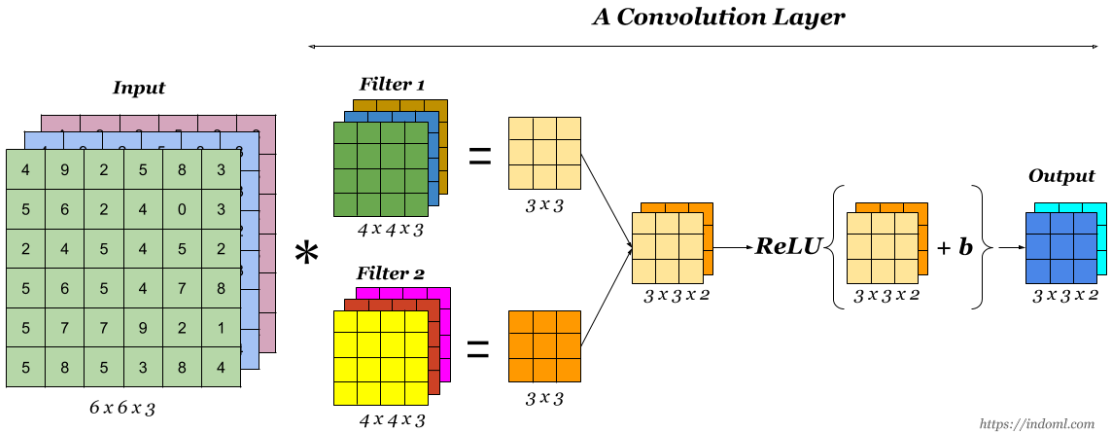
Tương tự như mạng nơ-ron nhân tạo truyền thống, một mặng CNN đơn giản bao gồm chuỗi các lớp nơ-ron được xếp liên tiếp với nhau, mỗi lớp có hàm tính toán và hàm kích hoạt riêng với mục đích khác nhau nhằm trích xuất ra các đặc trưng bậc cao qua từng lớp. Hình 2.9 mô tả cấu trúc của 1 mạng CNN đơn giản, bao gồm:

- Lớp tích chập (Convolutional Layer)
- Lớp Pooling
- Lớp kết nối đầu đủ (Fully Connected Layer)

Thông thường, các lớp tích chập sẽ được sắp xếp đứng cạnh nhau ngay đầu mỗi mạng CNN để trích xuất, tổng hợp lại các đặc trưng bậc cao của dữ liệu, theo sau mỗi lớp tích chập là lớp pooling giúp giảm độ phức tạp của dữ liệu và giảm độ phức tạp tính toán. Sau đó, các đặc trưng bậc cao tẻen sẽ được đưa vào lớp kết nối đầy đủ cùng với hàm kích hoạt softmax để đưa ra dự đoán.

2.3.2. Lớp tích chập

Trong kiến trúc mạng nơ-ron truyền thống, dữ liệu di chuyển tuần tự qua lớp đầu vào, các lớp ẩn và lớp đầu ra. Ngược lại, mạng nơ-ron tích chập (CNN) sử dụng lớp tích chập như một tập các ma trận đặc trưng, mỗi ma trận này là một bản scan của dữ liệu ban đầu, nhưng được trích xuất để tạo ra các đặc trưng cụ thể. Quá trình này sử dụng bộ lọc tích chập (kernel) quét qua ma trận dữ liệu đầu vào theo thứ tự từ trái qua phải, từ trên xuống dưới, và nhân tương ứng từng giá trị đầu vào với giá trị tương ứng trên bộ lọc tích chập và tính tổng. Kết quả được đưa qua hàm kích hoạt, tạo ra đầu ra của lớp tích chập. Hàm ReLU được sử dụng phổ biến để làm hàm kích hoạt trong quá trình này.

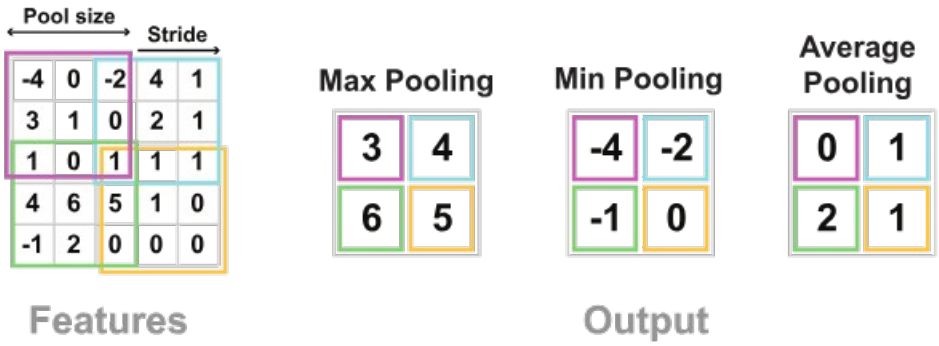


Hình 2.17 Minh hoạ phép tích chập, hay còn gọi là nhân chập trong mạng CNN (Nguồn: micro.medium.com).

Kết quả của lớp tích chập là một ma trận đặc trưng, kích thước của nó được kiểm soát bởi ba siêu thông số: độ sâu (depth), sải bước (stride), và đệm lót (padding). Độ sâu của đầu ra tương ứng với số lượng bộ lọc, mỗi bộ lọc trích xuất các đặc trưng khác nhau từ đầu vào. Điều này có thể được diễn giải bằng ý tưởng đặc trưng trích xuất từ các bộ lọc khác nhau đại diện cho 1 tính chất khác nhau nào đó của bức ảnh. Ví dụ như đầu vào là ảnh một con chó, bộ lọc đầu tiên có thể đại diện cho mắt, bộ lọc thứ hai có thể đại diện cho tai, v.v. Sải bước là giá trị khoảng cách mỗi lần trượt bộ lọc, có thể là 1 để di chuyển mỗi pixel hoặc lớn hơn để nhảy qua nhiều pixel trong một lần thực hiện trượt. Thông thường, giá trị này được chọn bằng 1 để trách việc mất mát thông tin nhưng đôi lúc, để giảm lượng tínhh toán, chúng ta cũng có thể thiết lập sải bước với giá trị lớn hơn như 2 hay 3. Thêm vào đó, đệm lót được sử dụng để trách mất mát thông tin ở biên bằng cách thêm các giá trị vào xung quang biên của ảnh. Số lượng giá trị được thêm vào phụ thuộc bởi tham số đệm lót. Thông thường, các giá trị 0 được thêm vào xung quanh đệm lót, nhưng cũng có trường hợp, các giá trị như giá trị trung bình bức ảnh, giá trị thấp nhất hay giá trị cao nhất được thêm vào thay vì giá trị 0. Điều này phụ thuộc hoàn toàn vào mục đích của người thiết kế mạng tích chập CNN.

Như vậy, mạng CNN không chỉ trích xuất các đặc trưng địa phương đặc biệt mà còn giảm kích thước ảnh đầu vào giúp tối ưu hóa quá trình tính toán và duy trì thông tin chất lượng ở biên ảnh.

2.3.3. Lớp pooling

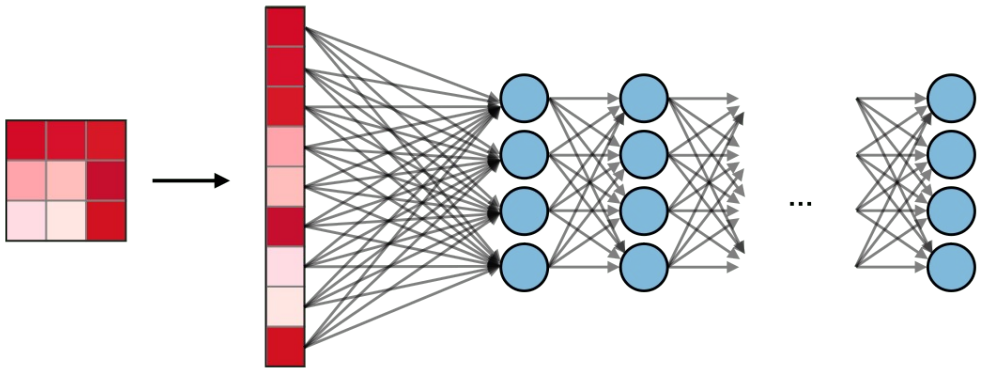


Hình 2.18 Minh hoạ 3 phương pháp pooling được sử dụng rộng rãi (Nguồn: epynn.net)

Như có đề cập ở trên, lớp Pooling thường được đặt sau lớp tích chập nhằm giảm kích thước dữ liệu giúp giảm số lượng tham số và số phép toán cần thực hiện trong mạng, từ đó có thể làm giảm thời gian huấn luyện và giải quyết được vấn đề overfitting. Lớp Pooling được hoạt động theo phương pháp trượt trên dữ liệu đầu vào tương tự như lớp tích chập. Lớp Pooling thường được sử dụng có kích thước 2×2, với bước nhảy là 2 làm giảm đi 75% nơ-ron ở lớp trước đó. Ngoài ra cũng có lớp Pooling với kích thước 3×3, 4×4, hay 5×5 nhưng ít được bắt gặp hơn. Có ba phương pháp Pooling để tính giá trị đầu ra được sử dụng rộng rãi và phổ biến, đó là:

- Bộ lọc lấy giá trị trung bình (Average-Pooling): Bộ lọc này thay thế một vùng nhỏ dữ liệu bằng giá trị trung bình của vùng đó. Với mong muốn giữ lại giá trị thông tin tổng thể trung bình để đại diện cho vùng thông tin đó.
- Bộ lọc lấy giá trị lớn nhất (Max-Pooling): Bộ lọc này thay thế một vùng nhỏ dữ liệu bằng giá trị lớn nhất của vùng đó. Với mong muốn giữ lại giá trị thông tin nổi bật nhất để đại diện cho vùng thông tin đó.
- Bộ lọc lấy giá trị nhỏ nhất (Min-Pooling): Bộ lọc này thay thế một vùng nhỏ dữ liệu bằng giá trị lớn nhất của vùng đó. Với mong muốn giữ lại giá trị thông tin ít nổi bật nhất để đại diện cho vùng thông tin đó. Tuy nhiên, bộ lọc này ít được sử dụng hơn so với hai bộ lọc trên.

2.3.4. Lớp kết nối đầu đủ



Hình 2.19 Mô tả kết nối giữa lớp tích chập CNN và mạng nơ-ron đơn giản (Nguồn: standford.edu)

Đây là lớp nơ-ron nhân tạo cuối cùng của mạng CNN với mục đích đưa ra dự đoán dựa trên các đặc trưng mà lớp tích chập và Pooling trước đó học được. Lớp kết nối đầy đủ này cũng có thể được coi là một mạng nơ-ron nhân tạo đơn giản. Các ma trận đặc trưng sau khi được trích xuất thông qua các lớp tích chập và Pooling sẽ được làm phẳng dưới dạng véc-tơ. Véc-tơ này sau đó được đi qua lớp đầu vào, các lớp ẩn và cuối cùng là lớp đầu ra theo sau bởi hàm kích hoạt softmax để đưa ra các dự đoán mong muốn.

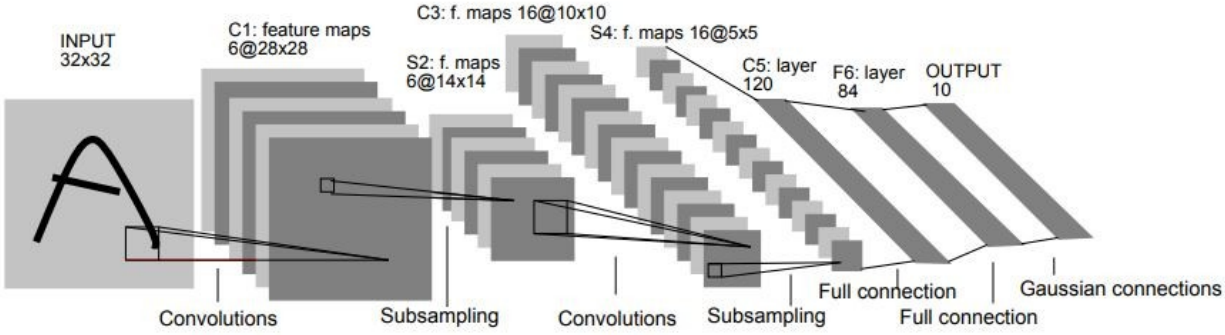
2.3.5. Những kiến trúc mạng CNN nổi tiếng

Hầu hết các mạng CNN đều được thiết kế theo nguyên tắc chung sau:

- Sử dụng nhiều lớp tích chập chồng lên nhau
- Giảm dần kích thước đầu ra của mỗi lớp
- Tăng dần số lượng ma trận đặc trưng
- Đưa ra dự đoán bởi lớp kết nối đầy đủ cuối cùng

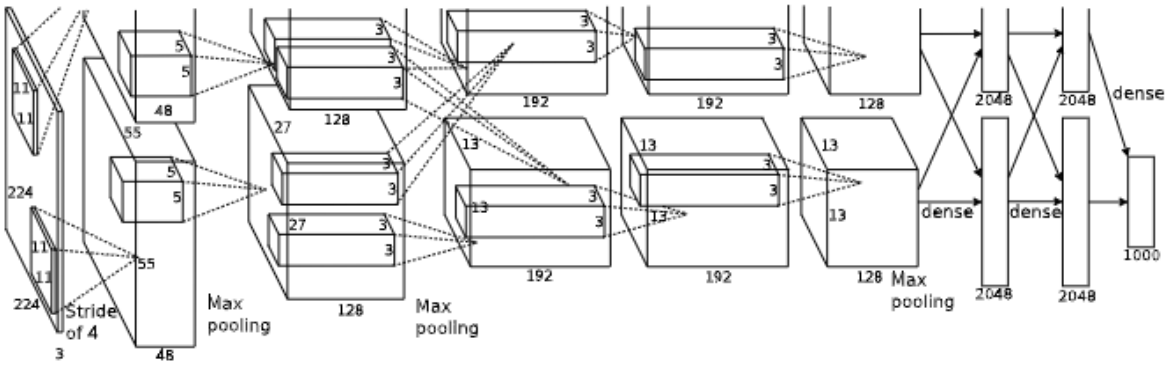
Các cách thiết kế của mạng CNN cũng được đa dạng hoá theo thời gian. Với các thiết kế ban đầu, mạng CNN đơn thuần chỉ bao gồm các lớp tích chập được kết tuần tự nhau và kết thúc bởi lớp kết nối đầy đủ. Sau này, các ý tưởng sáng tạo như mở rộng lớp tích chập theo chiều sâu, theo chiều ngang, hay bỏ qua kết nối được áp dụng vào mạng CNN cho kết quả hiệu quả hơn. Có rất nhiều các mạng CNN nổi tiếng được công bố và sử dụng rỗng rãi như:

- LeNet-5 [16]: Được giới thiệu bởi Yann Lecun năm 1998 cho bài toán nhận dạng chữ viết tay. Mô hình này được coi là nền tảng để thiết kế các mạng CNN sau này.



Hình 2.20 Kiến trúc mạng LeNet-5 [16].

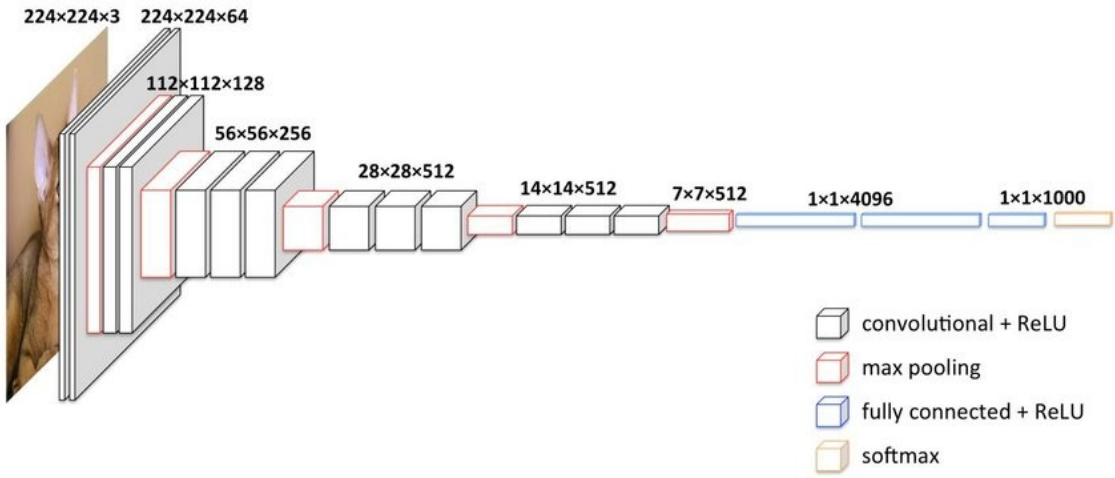
- AlexNet [17]: Alexnet, được phát triển bởi Alex Krizhevsky vào năm 2012 trong khuôn khổ của cuộc thi ImageNet 2012, giới thiệu một kiến trúc đột phá và tương tự với LeNet-5. Mô hình này được thiết kế để tham gia vào một trong những cuộc thi thị giác máy tính quan trọng nhất trên thế giới, đó là ImageNet ILSVRC. Kết quả của AlexNet gây ấn tượng mạnh, đặc biệt là với tỷ lệ lỗi trên tập dữ liệu kiểm tra chỉ là 16%. Điều này đánh dấu một cột mốc quan trọng tại thời điểm đó, khi mà mô hình học sâu đầu tiên thể hiện khả năng xuất sắc trong việc hiểu và phân loại hình ảnh. Thành công của AlexNet không chỉ là một chiến thắng trong cuộc thi mà còn là nguồn động viên quan trọng cho cộng đồng nghiên cứu thị giác máy tính. Mô hình này đã thuyết phục nhiều nhà nghiên cứu và chuyên gia về sự tiềm năng của học sâu trong việc giải quyết các nhiệm vụ phức tạp liên quan đến thị giác máy tính. Điều này đã mở ra một thời đại mới trong lĩnh vực nghiên cứu, khám phá và áp dụng mô hình học sâu cho các vấn đề thị giác máy tính hiện đại.



Hình 2.21 Kiến trúc mạng AlexNet (Nguồn: ResearchGate).

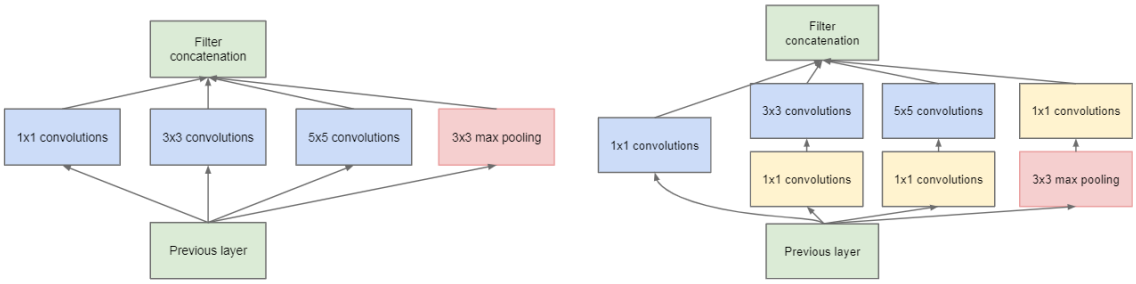
- VGG-16 [18]: Được ra đời vào năm 2014, VGG-16 đại diện cho một biến thể có độ sâu lớn hơn, tuy nhiên, lại giữ được sự đơn giản hóa so với cấu trúc tích chập thấy trong các mạng CNN. Nhìn vào hình 2.15, ta có thể nhận thấy rằng mặc dù VGG-16 giữ các tầng ở bậc cao so với LeNet và AlexNet, nhưng nó lại có sự đa

dạng và số lượng tầng lớn hơn đáng kể. Điểm đặc biệt của VGG-16 là việc giữ nguyên sự đơn giản trong cấu trúc, với các lớp convolution có kích thước nhỏ là 3×3 và các lớp Pooling có kích thước là 3×3. Sự tăng cường độ sâu của mô hình này mang lại khả năng học đặc trưng phức tạp từ dữ liệu, đồng thời cũng làm tăng độ chính xác và khả năng hiểu biết của mạng trong quá trình học. VGG-16, với sự cân nhắc giữa độ sâu và đơn giản, đã trở thành một trong những mô hình quan trọng trong lĩnh vực thị giác máy tính và nền tảng cho các nghiên cứu và ứng dụng hiện đại.



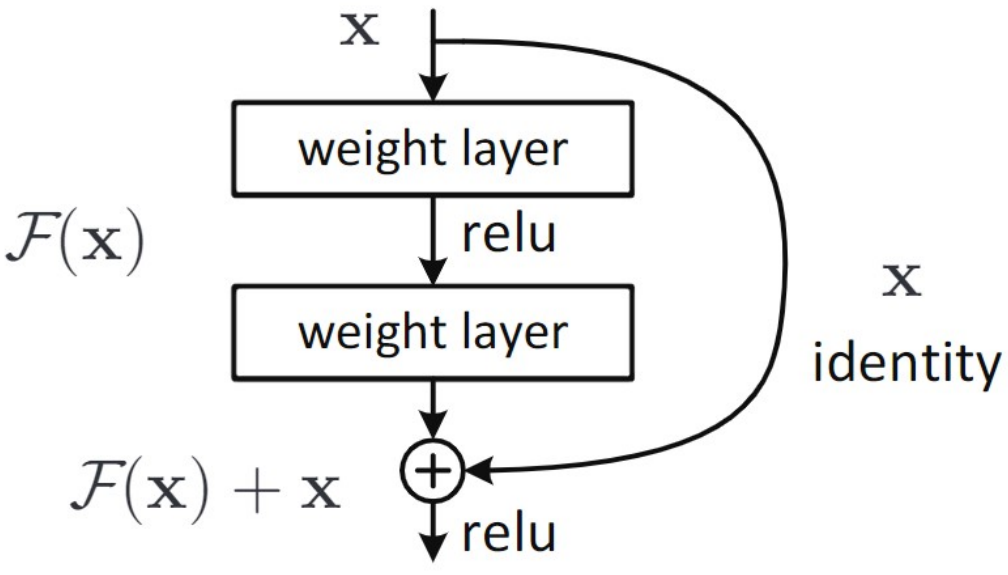
Hình 2.22 Kiến trúc mạng VGG-16 (Nguồn: ResearchGate).

- Inception (GoogleLeNet) [19]: Vào năm 2014, nhóm nghiên cứu của Google đã giới thiệu mạng Inception trong khuôn khổ của cuộc thi ImageNet 2014. Điều đặc biệt của mô hình này là không theo kiến trúc truyền thống với các lớp nối tiếp nhau, như các mạng CNN đã được giới thiệu trước đó. Thay vào đó, Inception sử dụng các đơn vị được gọi là "inception cell", thực hiện phép tích chập đầu vào với nhiều bộ lọc khác nhau và tổng hợp kết quả trên nhiều nhánh. Để tối ưu hóa tính toán, các kích thước 1×1 được áp dụng để giảm chiều sâu của kênh đầu vào. Mỗi "cell" trong Inception sử dụng các bộ lọc có kích thước 1×1, 3×3, 5×5 để trích xuất đặc trưng từ đầu vào. Trong quá trình nghiên cứu, các nhà khoa học đã thử nghiệm thay thế bộ lọc 3×3 và 5×5 bằng các bộ lọc lớn hơn như 7×7 hoặc 11×11 và phát hiện rằng các bộ lọc lớn này hiệu quả hơn trong việc trích xuất đặc trưng. Tuy nhiên, thời gian tính toán cho các bộ lọc lớn này tăng đáng kể. Inception đã mở ra một loạt các biến thể, trong đó Inception-v4 là một trong những phiên bản nổi bật nhất.



Hình 2.23 Kiến trúc mạng Inception (GoogleLeNet) [19]

- ResNet [20]: ResNet, hay còn được biết đến với tên gọi đầy đủ là Residual Network, là một sáng tạo thú vị và đáng chú ý của Kaiming He và nhóm nghiên cứu, đã giành vị trí quán quân tại cuộc thi ImageNet ILSRC vào năm 2015. Kiến trúc của ResNet đặc trưng bởi việc phân chia các lớp nhân chập thành các khối nhân chập, nơi mà mỗi khối chứa các đường kết nối trực tiếp giữa các lớp mà không theo trình tự tuần tự như trước đây. Một đặc điểm quan trọng của ResNet là việc xây dựng các đường kết nối trực tiếp, hay còn gọi là "đường dư," giữa các lớp trong mỗi khối. Điều này giúp giải quyết vấn đề biến mất đạo hàm trong quá trình huấn luyện mô hình sâu. Thêm vào đó, ResNet còn tích hợp các lớp chuẩn hóa dữ liệu để cải thiện sự ổn định và hiệu suất của mô hình. Kiến trúc của ResNet được phát triển với nhiều cấu hình khác nhau, điển hình như ResNet-18, ResNet-34, ResNet-50 và ResNet-101, với mức độ phức tạp tăng dần.



Hình 2.24 Kiến trúc mạng 1 Block của Resnet-34 và Resnet-50 [20].

2.4. Mạng nơ-ron trí nhớ ngắn hạn dài (Long Short-Term Memory - LSTM)

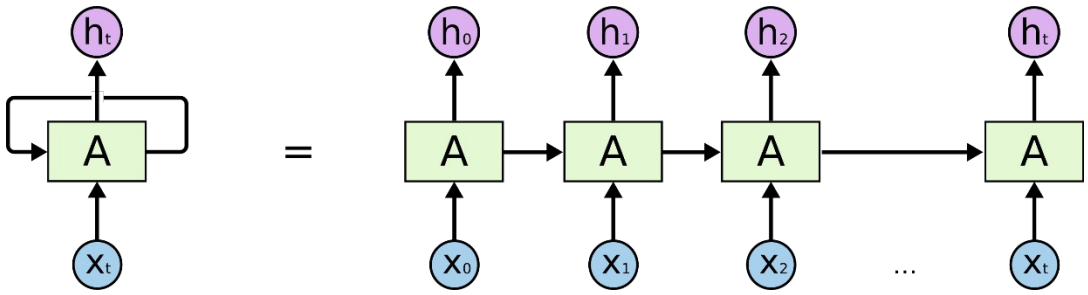
Mạng LSTM [21] là một loại mạng nơ-ron đặc biệt trong họ mạng hồi quy (Recurrent Neural Network – RNN [22]), được thiết kế để giải quyết nhược điểm của các RNN truyền thống liên quan đến vấn đề biến mất đạo hàm. Được giới thiệu lần đầu bởi Hochreiter và Schmidhuber vào năm 1997, LSTM đã trở thành một công cụ quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên, dự đoán chuỗi thời gian và nhiều ứng dụng khác.

Điểm mạnh chính của LSTM nằm ở khả năng duy trì và chọn lọc thông tin quan trọng từ quá khứ. Điều này được thực hiện thông qua việc sử dụng các cổng (gates) như cổng quên, cổng đầu vào và cổng đầu ra, giúp kiểm soát quá trình truyền thông tin trong mỗi cell của mạng. Quan trọng hơn, khả năng này giúp LSTM duy trì thông tin lâu dài và xử lý các chuỗi dữ liệu có chiều dài biến đổi một cách hiệu quả, làm cho nó trở thành công cụ ưa thích trong việc mô hình hóa và dự đoán các sự kiện có tính chất chuỗi và dài hạn.

2.4.1. Mạng hồi quy (Recurrent Neural Network – RNN)

Không thể phủ nhận ưu điểm của mạng nơ-ron đơn giản và mạng tích chập được giải thích bên trên. Nhưng phần nào đó, các dữ liệu truyền vào hai mạng này khá độc lập qua các lần truyền hay qua thời gian. Cả hai mạng nơ-ron đơn giản và mạng nơ-ron tích chập đều đưa ra quyết định độc lập ra mỗi lần nhận được dữ liệu. Điều này là chưa tốt nếu như so với não bộ con người, con người không bắt đầu suy nghĩ tại mọi thời điểm mà chúng ta luôn luôn gợi nhớ tới các kiến thức đã biết trước đó để tiếp tục suy nghĩ về vấn đề hiện tại. Ví dụ như khi đọc hai câu văn ngắn, chúng ta luôn luôn có suy nghĩ liên kết nội dung hai câu văn lại với nhau. Có thể nói rằng, não bộ con người sau khi tiếp nhận thông tin luôn luôn lưu giữ phần thông tin đã đọc ở đâu đó hay nói theo cách máy móc chính là não bộ có một bộ nhớ tạm để lưu lại những gì đã diễn ra trước đó. Tuy nhiên, với hai mạng đã nêu trên thì không thể làm điều này.

Vì vậy, mạng nơ-ron hồi quy (RNN) [22] sinh ra để giải quyết vấn đề trên. Mạng RNN chứa một vòng lặp tại mỗi nút cho phép thông tin khi sau khi xử lý được lưu lại cho lần sau.



Hình 2.25 Kiến trúc mạng nơ-ron hồi quy (RNN) (Nguồn: stackexchange).

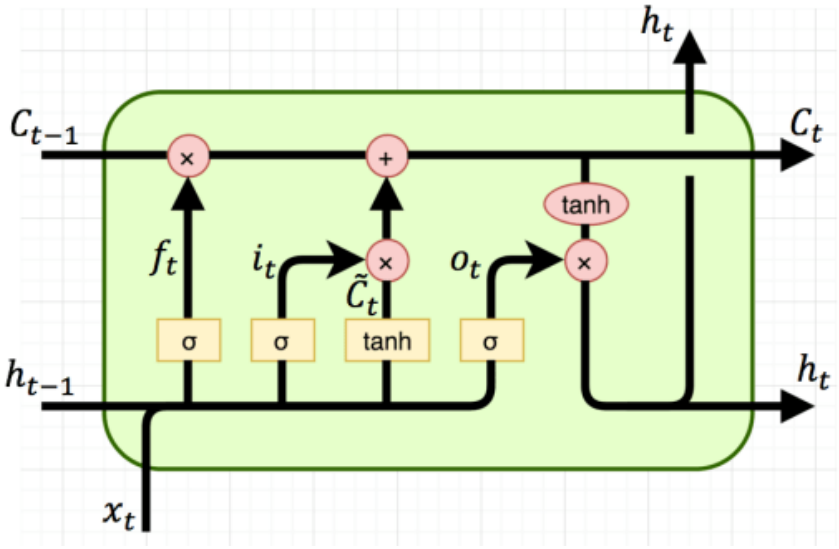
Hình 2.19 mô tả mạng RNN với đầu vào x_t và đầu ra h_t. Vòng lặp cho phép thông tin sau khi xử lý có thể được lưu lại để tái sử dụng cùng với thông tin đầu vào tiếp theo để từ đó đưa ra kết quả đầu ra. Một cách đơn giản hơn, một mạng nơ-ron hồi quy có thể được coi là nhiều bản sao chép của cùng một mạng, trong đó, đầu ra của 1 mạng này chính là đầu vào của một mạng sao chép khác. Bởi tính chất hồi quy thông tin này, mạng RNN đã có những kết quả vượt trội trong lĩnh vực nhận dạng giọng nói, dịch máy, mô tả ảnh, v.v. Tuy nhiên, RNN vẫn là một mạng nơ-ron chưa hoàn chỉnh và cần cải thiện khuyết điểm. Một trong các khuyết điểm đó là "Phụ thuộc từ xa".

2.4.2. Vấn đề phụ thuộc từ xa

Không thể phủ nhận RNN là một mạng nơ-ron với thiết kế độc đáo, giải quyết vấn đề truy hồi thông tin đã xử lý để cùng với thông tin hiện tại đưa ra dự đoán. Nhưng vấn đề xảy ra khi lượng thông tin cần lưu trữ qua mỗi vòng lặp tăng lên nhưng khả năng lưu trữ lại có hạn dẫn đến mất mát thông tin. Các kết quả thử nghiệm cho thấy, nếu lượng thông tin có tính liên kết với nhau có độ dài ngắn, mạng RNN cho kết quả khá thuyết phục so với các mạng nơ-ron truyền thống. Nhưng với lượng thông tin có tính liên kết với nhau dài như một câu văn dài 100 kí tự trong bài toán dịch thuật, mạng RNN sẽ gặp khó khăn và cho ra kết quả không tốt do có thể lượng thông tin ở đầu đã mất đi khi đã trải qua một lượng số vòng lặp nhất định nào đó.

Hochreiter (1991) và Bengio, et al. (1994) đã chỉ ra một vấn đề chính của mạng hồi quy RNN đó chính là biến mất đạo hàm. Khi mô hình RNN được huấn luyện thông qua thuật toán lan truyền ngược, đạo hàm của hàm mất mát có thể trở nên rất nhỏ khi thông tin phải lan truyền qua nhiều bước thời gian. Điều này dẫn đến việc mô hình không thể hiệu quả học được các mối quan hệ dài hạn trong dữ liệu.

2.4.3. Mạng trí nhớ ngắn hạn dài (Long short term memory - LSTM)



Hình 2.26 Kiến trúc mạng trí nhớ ngắn hạn dài (LSTM) (Nguồn: Deep Learning cơ bản).

Ý tưởng và cấu trúc cổng

Để giải quyết vấn đề mất mát thông tin và mất mát đạo hàm có đề cập tại phần 2.4.2, mạng LSTM đã được thiết kế và giới thiệu bởi Hocheirter và Schmidhuber năm 1997 nhằm mục đích giải quyết vấn đề biến mất đạo hàm tỏng mạng hồi quy – RNN, nơi thông tin quan trọng từ quá khứ thường bị mất khi lan truyền ngược. Ý tưởng chính của LSTM là sử dụng các cổng để kiểm soát thông tin trong quá trình lan truyền và cập nhật trạng thái ẩn. LSTM giữ lại thông tin quan trọng trong trạng thái dài hạn thông qua các cổng forget, input và output. Cơ chế này cho phép LSTM có khả năng xử lý và hiểu cấu trúc dữ liệu chuỗi dài hạn một các hiệu quả, tránh được vấn đề biến mất đạo hàm.

Mạng LSTM hoạt động dựa trên cơ chế cổng để kiểm soát và duy trì thông tin trong quá trình lan truyền. Mỗi cell trong LSTM bao gồm ba cổng:

- Cổng forget: Cổng này quyết định thông tin nào trong trạng thái trước đó nên được giữ lại và thông tin nào nên được loại bỏ.
- Cổng input: Cổng này xác định thông tin mới nào nên dược thêm vào trạng thái trước đó thông qua đầu vào hiện tại và trạng thái ẩn trước đó.
- Cổng output: Cổng này quyết định đầu ra dự đoán dự trên trạng thái hiện tại.

Các bước hoạt động

Giả sử ta có dữ liệu đầu vào từ vòng lặp trước h_{t-1} và dữ liệu mới x_t cùng với trạng thái cell C_{t-1}.

- Mạng LSTM thực hiện cổng forget để quyết định xem thông tin nào cần loại bỏ đi từ tế bào cell bởi việc tính hàm kích hoạt dựa trên hai giá trị h_{t-1} và x_t rồi đưa ra kết quả trong khoảng $[0,1]$. Giá trị 1 tương ứng giữ lại toàn bộ thông tin từ cell còn 0 có nghĩa là loại bỏ hoàn toàn.

$$f_t=\sigma(W_f.[h_{t-1},x_t]+b_f)$$

- Sau khi xác định loại bỏ thông tin tại cell, LSTM tiếp tục quyết định xem thông tin mới nào sẽ được lưu vào trạng thái cell đó thay cho các thông tin đã bị loại bỏ thông qua cổng input và hàm kích hoạt tanh.

$$i_t=\sigma(W_i.[h_{t-1},x_t]+b_i)$$

$$\widetilde{C}_t=\sigma(W_C.[h_{t-1},x_t]+b_C)$$

- Sau khi đã quyết định được lượng thông tin sẽ loại bỏ và thêm mới vào trạng thái cell, LSTM cần cập nhật trạng thái cell C_{t-1} thành trạng thái mới C_t.

$$C_t=(f_t*C_{t-1}+i_t*\widetilde{C}_t)$$

- Cuối cùng, LSTM sử dụng cổng output để quyết định đầu ra từ trạng thái cell C_t và 2 giá trị h_{t-1}, x_t.

$$o_t=\sigma(W_o.[h_{t-1},x_t]+b_o)$$

$$h_t=o_t*\tanh(C_t)$$

2.5. Đề xuất mô hình hình học sâu cho bài toán

Ở phần trước, đồ án đã trình bày các mạng nơ-ron tiềm năng để có thể xây dựng mô hình mạng học sâu cho nhiệm vụ ước lương tư thế người sử dụng tín hiệu radar. Trong phần này, đồ án sẽ trình bày các kiến trúc mô hình mạng học sâu được đề xuất để giải quyết tác vụ ước lượng tư thế người sử dụng tín hiệu radar.

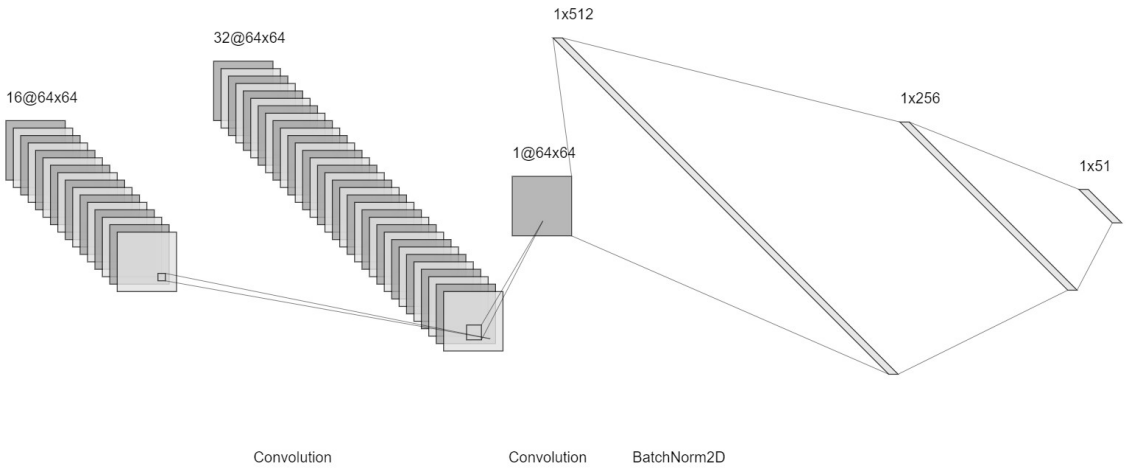
2.5.1. Mô hình mạng nơ-ron tích chập đơn giản – MAR-CNN

Ý tưởng và mong muốn

Dựa trên [7], đồ án xin phép dựng lại mô hình mạng nơ-ron tích chập đơn giản được sử dụng trong bài báo để làm thước đo so sánh cho các mạng nơ-ron được đề xuất sau này. Với ý tưởng dữ liệu radar sau khi xử lý được đưa về dạng các điểm point-cloud trên không gian ba chiều x, y, z, hai tác giả An và Ogras

đã đề xuất mạng nơ-ron tích chập với thiết kế cấu trúc mạng không quá lớn, đủ phù hợp với dữ liệu có số đặc trưng bé cùng với mong muốn khai thác điểm mạnh về khả năng học các đặc trưng cục bộ mà CNN có thể đạt được.

Kiến trúc mạng

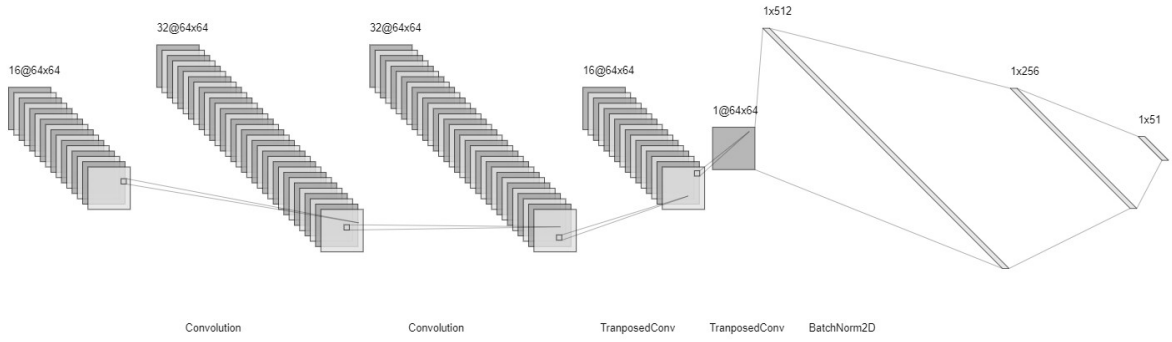


Hình 2.27 Kiến trúc mạng MAR-CNN.

2.5.2. Mạng Advanced-MAR-CNN

Ý tưởng và mong muốn

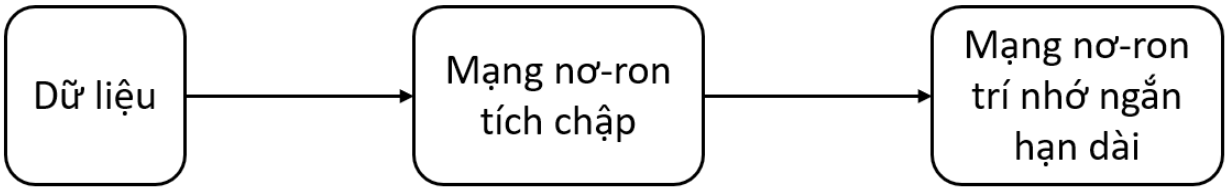
Dựa trên ý tưởng mạng MAR-CNN [7] cùng với cấu trúc mạng AutoEncoder [23], đồ án thiết kế 1 mạng AutoEncoder với 4 lớp mạng nơ-ron tích chập đơn giản dựa trên số lớp nơ-ron, kích thước mỗi lớp nơ-ron và các tham số liên quan của mạng MAR-CNN. Đồ án mong muốn tại phần trích xuất đặc trưng, mạng Advanced-MAR-CNN sẽ trích xuất và học được nhiều thông tin hơn so với mạng MAR-CNN đơn thuần của hai tác giả AN và Ogras đưa ra.



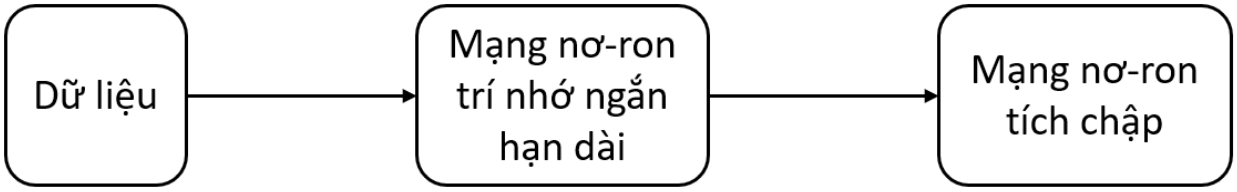
Hình 2.28 Kiến trúc mạng Advanced-MAR-CNN.

2.5.3. Mạng nơ-ron tích chập kết hợp mạng trí nhớ ngắn hạn dài

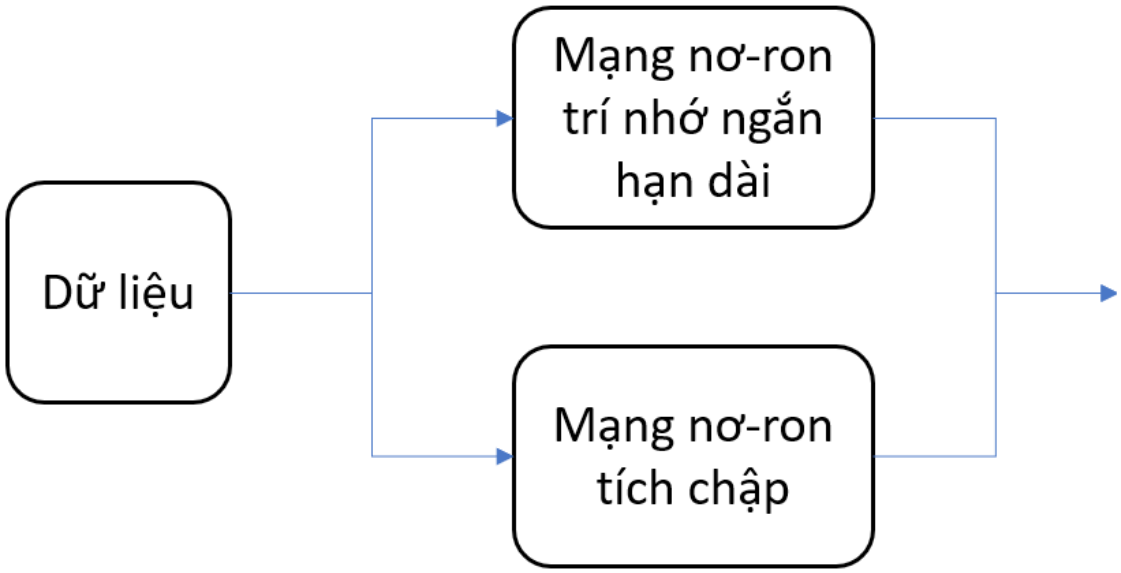
Ở phần này, đồ án xin đề xuất 3 mô hình mạng lần lượt là CNN-LSTM, LSTM-CNN và Parallel-CNN-LSTM. Nhận thấy trong bài toán ước lượng tư thế người, các tư thế theo thời gian có mối quan hệ chặt chẽ với nhau. Đặc biệt với tốc độ chụp của thiết bị radar ở khoảng 10 khung hình trên 1 giây, nên khoảng thời gian giữa 2 khung hình là rất bé. Vậy nên, có thể nói tư thế người trong 2 khung hình không có sự chênh lệch nhiều về toạ độ. Vậy nên, yếu tố các trí nhớ ngắn hạn dài được cân nhắc là một đặc trưng quan trọng giúp cải thiện chất lượng mô hình. Đồ án kết hợp tính chất trí nhớ ngắn hạn dài, cái mà cung cấp kiến thức về tư thế người ở các khung hình trước đó và mạng nơ-ron tích chập, cái mà cung cấp kiến thức về các giá trị điểm point-cloud cục bộ gần nhau. Mạng CNN-LSTM có nghĩa là lớp mạng nơ-ron tích chập được sắp xếp trước sau đó đến lớp mạng nơ-ron trí nhớ ngắn hạn dài. Với mạng này, đồ án mong muốn mạng trích xuất đặc trưng cục bộ của các vùng điểm point-cloud trước khi kết nối chúng theo thời gian. Ngược lại, mạng LSTM-CNN được sắp xếp theo thứ tự mạng nơ-ron trí nhớ ngắn hạn dài, mạng nơ-ron tích chập. Với sự sắp xếp này, đồ án mong muốn mạng có góc nhìn tổng quan về thời gian giữa các điểm dữ liệu point-cloud đầu vào, sau đó mới tích hợp lại và học các đặc trưng cục bộ. Và cuối cùng, mạng Parallel-CNN-LSTM được thiết kế với 2 mạng nơ-ron tích chập và mạng trí nhớ ngắn hạn dài song song, điều này giúp mạng học trực tiếp thông tin vùng cục bộ và thông tin theo thời gian.



Hình 2.29 Mô tả kiến trúc mạng CNN-LSTM.



Hình 2.30 Mô tả kiến trúc mạng LSTM-CNN.



Hình 2.31 Mô tả kiến trúc mạng Parallel-CNN-LSTM.

2.6. Tổng kết chương 2

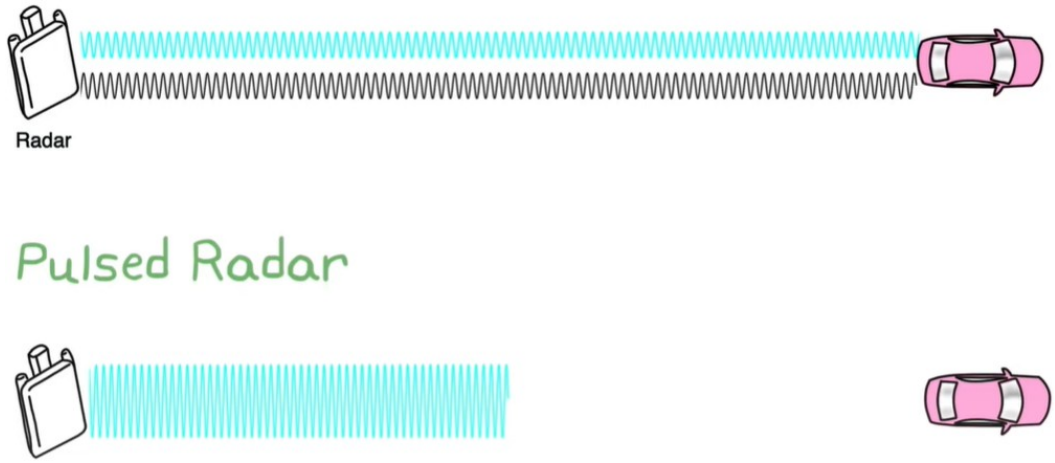
Trong chương 2, đồ án đã giới thiệu về phương pháp học sâu, các mô hình mạng nơ-ron đơn giản đến các mô hình mạng nơ-ron phổ biến. Đồng thời, đồ án cũng trình vày chi tiết phươngán đề xuất áp dụng học sâu để xây dựng 5 mô hình mạng cho bài toán ước lượng tư thế người

CHƯƠNG 3: THỰC NGHIỆM VÀ KẾT QUẢ

3.1. Dữ liệu

3.1.1. Tổng quan về sóng radar liên tục

Tín hiệu radar được chia làm hai loại: sóng liên tục (continuous wave) và radar xung (pulsed radar).



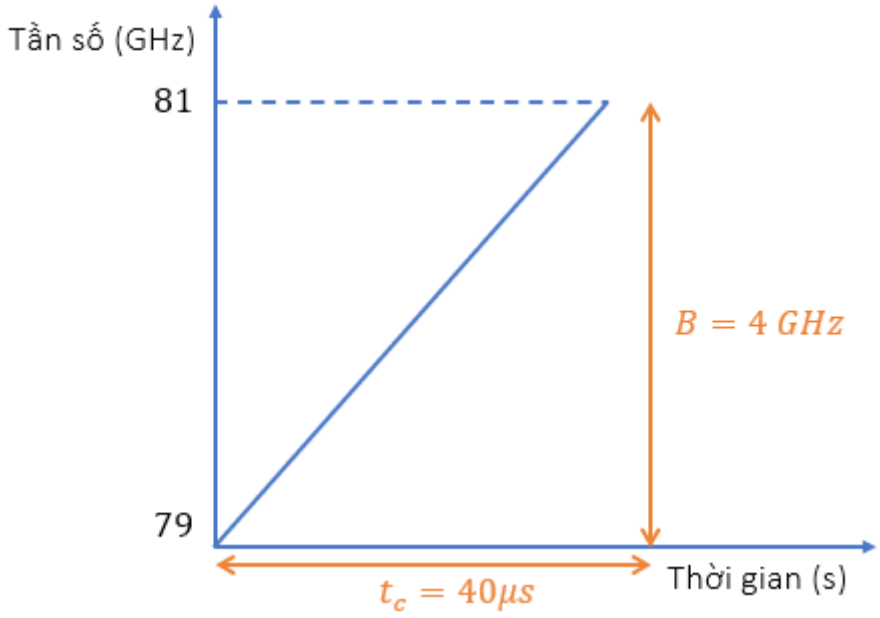
Hình 3.32 Mô tả hình ảnh sóng liên tục (phía trên) và sóng radar dạng xung (phía dưới) (Nguồn: Matlab)

Với dạng sóng pulsed radar, sóng được phát ra theo xung và thu nhận lại khi sóng va chạm vào vật và phản xạ về. Quãng thời gian Δt giữa phát và thu sẽ cho ta biết chính xác khoảng cách từ thiết bị radar tới vật phản xạ theo công thức

$$khoảng\ cách = \frac{vận\ tốc\ ánh\ sáng \times \Delta t}{2}$$

Tuy nhiên, thiết bị phát dạng sóng này khá đắt đỏ và kích thước lớn hơn nhiều so với thiết bị phát sóng liên tục. Vậy nên, thiết bị radar phát sóng liên tục phù hợp hơn với bài toán ước lượng tư thế người sử dụng tín sử dụng tín hiệu radar.

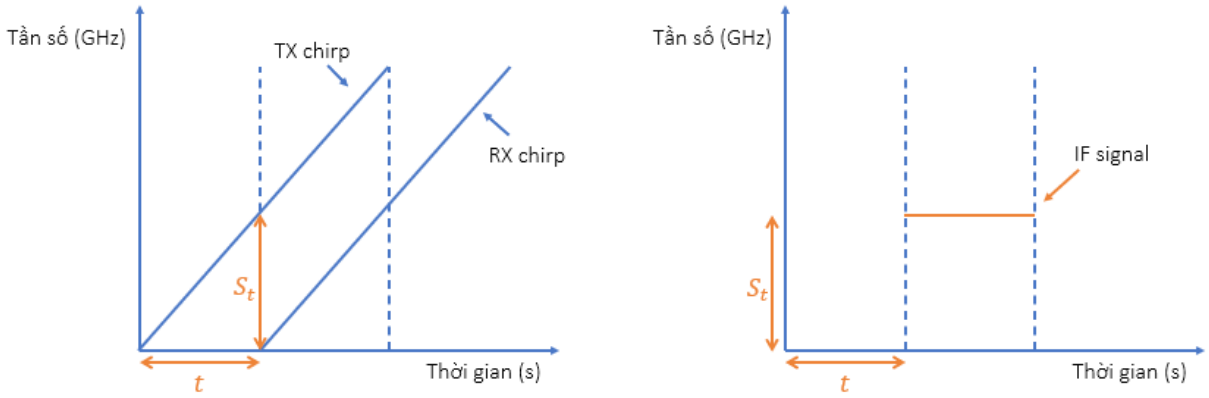
Một thiết bị radar sóng điều chế tần số liên tục (FMCW radar – Frequency Modulated Continuous Wave radar) phát ra một tín hiệu được gọi là 1 chirp. Chip là 1 sóng hình sin với tần số tăng dần theo thời gian.



Hình 3.33 Mô tả 1 chirp với biểu đồ tần số - thời gian với tần số bắt đầu (start frequency - fc), băng thông (bandwidth- B) và khoảng thời gian t_c).

Ví dụ một hệ thống radar FMCW với một bộ phát và một bộ thu tín hiểu radar.

Bộ trộn là một thiết bị gồm có ba phần: hai đầu vào và một đầu ra. Mục đích chính của bộ trộn là tạo ra một sóng hình sin, được gọi là tín hiệu tần số tức thời (IF signal - instantaneous frequency signal). Tần số tức thời của sóng đầu ra chính bằng sự khác biệt giữa tần số tức thời của hai sóng đầu vào, pha bằng với sự khác biệt pha của hai sóng đầu vào. Hình ảnh 3.3 dưới đây mô tả hai sóng đầu vào và sóng IF được sinh ra thông qua bộ trộn.



Hình 3.34 Mô tả 2 sóng đầu vào bộ trộn (hình bên trái) và sóng IF được sinh ra (hình bên phải).

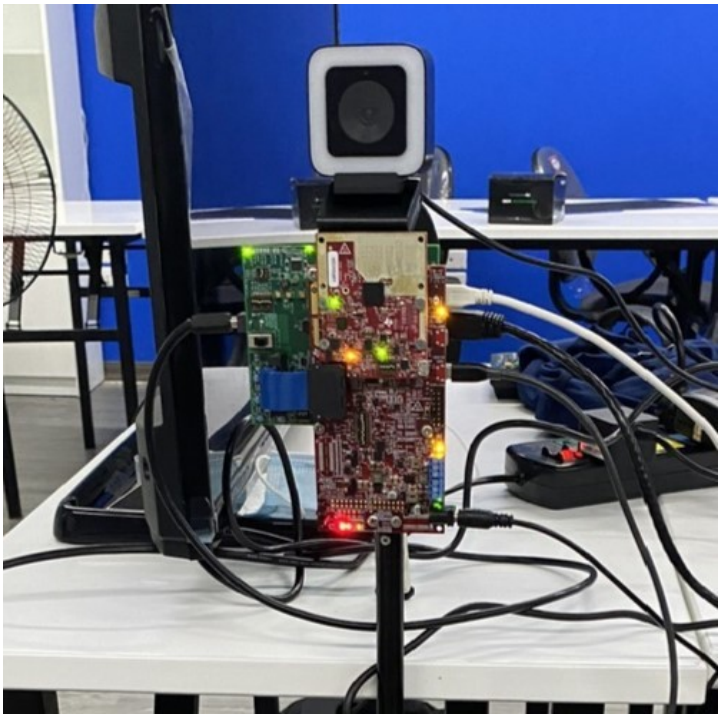
3.1.2. Bộ dữ liệu cho bài toán ước lương tư thế người sử dụng tín hiệu radar

Như đã đề cập ở phần mở đầu, dữ liệu là một vấn đề lớn trong bài toán ước lượng tư thế người sử dụng tín hiệu radar. Do bộ dữ liệu không chỉ đòi hỏi dữ liệu thu thập từ thiết bị radar, mà còn cần có dữ liệu hình ảnh, được thu thập đồng bộ song song để lấy làm chuẩn cho dữ liệu radar học theo. Vậy nên hiện nay, có rất ít bộ dữ liệu tiêu chuẩn gồm có dữ liệu radar, dữ liệu ảnh đồng bộ được công bố công khai. Vậy nên, đồ án lần này, mong muốn sử dụng bộ dữ liệu tiêu chuẩn đã được công bố và giới thiệu thêm một bộ dữ liệu mới mang tên Radim5 PTIT (viết tắt là RP). Đây là bộ dữ liệu được thu thập và dành riêng cho bài toán ước lượng tư thế người sử dụng tín hiệu radar.

3.1.3. Thu thập dữ liệu

Quá trình xây dựng tập dữ liệu RP bao gồm ba giai đoạn. Giai đoạn đầu tiên là nghiên cứu và xây dựng hướng dẫn, các mẫu hành động có trong tập dữ liệu, giai đoạn thứ hai là hướng dẫn và thu thập, giai đoạn thứ ba là tổng hợp và chỉnh sửa dữ liệu.

Nghiên cứu và xây dựng hướng dẫn, mẫu hành động



Hình 3.35 Minh hoạ cách thiết lập thiết bị radar và camera để thu thập dữ liệu.

Để xây dựng tập dữ liệu đồng bộ, nhất quán và giảm thiệu độ lệch trục toạ độ, đồ án thiết kế một hệ thống bao gồm một mmWave TI radar, một camera. Thông số kỹ thuật chi tiết và tính năng của tất cả các cảm biến được biểu diễn ở bảng 3.1. Thiết bị mmWave TI radar và camera được đặt trên một cái giá ba chân có độ cao 1m.

Bảng 3.1 So sánh thông số giữa các thiết bị thu thập dữ liệu sử dụng bao gồm: số lượng thiết bị, tần số thu dữ liệu, cách kết nối, nguồn năng lượng cung cấp và dạng dữ liệu đầu ra.

Thiết bị	Số lượng	Tần số	Kết nối	Năng lượng	Đầu ra
mmWave TI radar	1	10 Hz	Có dây	1.5W	IQ data
Camera	1	30 Hz	Có dây	30W	RGB khung hình

Tiếp theo, đồ án đưa ra hướng dẫn thu thập dữ liệu gồm có số lượng hành động cho bộ dữ liệu RP và thời gian thực hiện hành động, môi trường xung quanh khi thu dữ liệu. Dựa trên bản hướng dẫn này, một vài mẫu thử nghiệp sẽ được ghi lại bởi một sinh viên. Sau đó, tài liệu hướng dẫn sẽ được chỉnh sửa để dữ liệu thu được là phù hợp với các thiết bị hiện có và là tốt nhất.

3.1.4. Tiền xử lý dữ liệu

Như đã đề cập ở trên, một FMCW radar tổng hợp một chuỗi tín hiệu chirp để hình thành một khung hình. Thiết bị radar truyền khung hình gồm N chirp sử dụng ăng-ten TX. Nếu có bất kì vật vào trong phạm vị, nó sẽ phản xạ lại khung hình chirp. Sau đó, FMCW radar nhận tín hiệu phản xạ tại ăng-ten nhận RX. Tín hiệu phát ra tại ăng-ten TX và tín hiệu nhận được tại ăng-ten RX có khác tần số và pha. Bộ trộn tiến hành xử lý hai loại tín hiệu này và tạo ra tín hiệu IF.

Giả sử chỉ có một vật trong phạm vi sóng phát và phản xạ lại khung hình chirp tại khoảng cách d từ thiết bị FCMW radar. Độ trễ tín hiệu nhận được so với tín hiệu phát có thể được tính bởi công thức

$$t = \frac{2d}{c}$$

Với $c = 3 \times 10^8$

Thật vậy, do chỉ có một vật phản xạ nên sóng IF chỉ có một giai điệu. Trường hợp khung hình các chirp được phản xạ từ nhiều vật hoặc từ nhiều bộ phận khác nhau trên cơ thể, bộ trộn sẽ tạo ra một sóng IF với nhiều giai điệu khác nhau. Các chíp radar FMCW trích xuất các giai điệu sóng IF bằng cách tính phổ tần số sử dụng thuật toán biến đổi miền tần số nhanh Fourier (fast Fourier Transform – FFT). Với trường hợp một vật phản xạ, tần số của giai điệu sóng tỉ lệ thuận với khoảng cách tương ứng giữa

vật và thiết bị radar. Sóng IF được xử lý trong miền kỹ thuật số để ánh xạ giai điệu vào các vùng khoảng cách sử dụng thuật toán biến đổi miền tần số nhanh Fourier khoảng cách (range FFT). Công thức liên hệ giữa phạm vi phân giải (range resolution) và băng thông của chirp:

$$d_{res} = \frac{c}{2B}$$

Với $c = 3 \times 10^8$, B: băng thông

Ngoài thông số khoảng cách, vận tốc di chuyển của vật phản xạ cũng đóng vai trò quan trọng. Radar FMCW tính vận tốc sử dụng sự thay đổi pha trong một tín hiệu IF thông qua các chirp trong nó. Quá trình này chuyển đổi những dịch chuyển nhỏ của vật thành độ lệch pha trong tín hiệu IF. Như trong trường hợp có nhiều vật phản xạ, phát hiện vận tốc sẽ cho phép khả năng phân biệt các vật có cùng một khoảng cách tới radar nhưng có khác vận tốc di chuyển. Để tính toán ra vận tốc của các vật khác nhau, thuật toán biến đổi miền tần số Doppler (Doppler-FFT) được sử dụng. Công thức liên hệ giữa độ phân giải vận tốc và thời gian của một khung hình:

$$v_{res} = \frac{\lambda}{2NT_c}$$

Với λ: bước sóng, N: số lượng chirp, T_c: khoảng thời gian giữa 2 chirp

Tín hiệu sau khi thực hiện 2D-FFT sẽ được lọc nhiễu bằng phương pháp tỉ lệ báo động giả mạo không đổi (Constant false alarm rate- CFAR). Thông số góc tới sẽ được ước lượng cuối cùng, hay còn được gọi là Angle of Arrival (AOA). Góc tới được định nghĩa là góc mà sóng phản xạ từ vật quay trở về radar nằm trên mặt phẳng nằm ngang. Để đo được góc tới, thiết bị radar sẽ cần yêu cầu có ít nhất hai ăng-ten thu và được tính toán bởi công thức:

$$\theta_{res} = \frac{\lambda}{N_{TX} N_{RX} d \cos(\theta)}$$

Với λ: bước sóng, N_{TX}: số lượng ăng-ten phát sóng, N_{RX} là số lượng ăng-ten thu sóng, d là khoảng cách giữa 2 ăng-ten thu liên tiếp nhau, θ là góc giữa 2 ăng-ten thu liên tiếp nhau.

Các thiết bị radar thường được thiết kế đảm bảo $d = \frac{\lambda}{2}$ và $\theta = 0$. Vậy nên, độ phân giải góc tới có thể được tính lại như sau:

$$\theta_{res} = \frac{2}{N_{TX} N_{RX}} (radians)$$

Thiết bị radar ước lượng góc tới bởi sự khác nhau giữa đỉnh pha sau khi dữ liệu được xử lý qua Doppler-FFT bởi sự chênh lệch khoảng cách của vật tới mỗi ăng-ten thu. Góc tới này chính bằng góc tới phương năm ngang chia cho góc tới phương thẳng đứng.

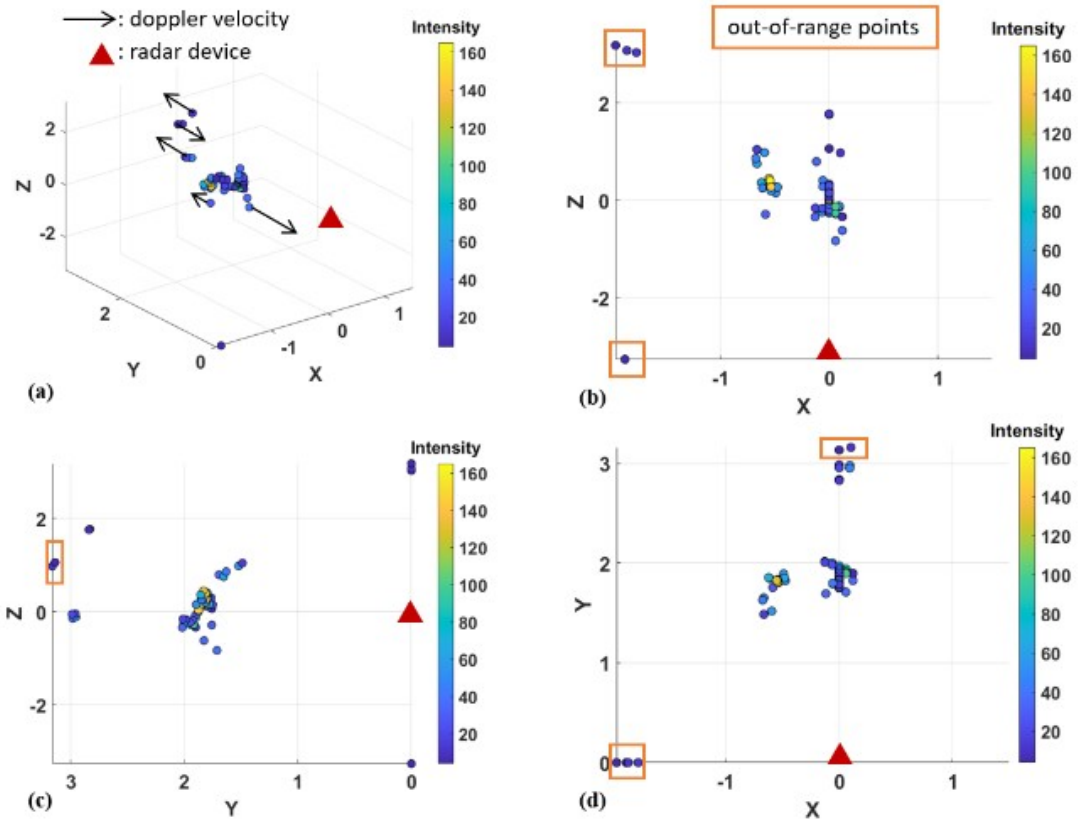
Thực tế, với mỗi một sóng phản xạ ngược lại từ vật, thiết bị radar sẽ nhận tại N_{RX} ăng-ten thu với mỗi ăng-ten thu nhận N_{TX} sóng truyền đi. Vậy nên, với mỗi phần của vật được phản xạ lại sẽ nhận được tương ứng $N_{TX} N_{RX}$ tín hiệu sóng. Các tín hiệu này sau khi được tính toán các bước đề cập trên sẽ được biến đổi dưới dạng điểm p_i trong khoang gian 3 chiều. Thêm vào đó, để tăng tính chính xác và tính thể hiện của phần phản xạ, giá trị độ đo vận tốc của phần phản xạ và cường độ sẽ được thêm vào tương ứng với mỗi điểm. Các phần phản xạ được biểu diễn dưới dạng điểm trong không gian 3D được định dạng như sau:

$$p_i = \{x_i, y_i, z_i, D_i, I_i\}, i \in [0, N_p]$$

Với các giá trị x, y, z thể hiện toạ đọ trong không gian 3 chiều, D là giá trị vận tốc, I là giá trị cường độ tín hiệu, N_p là tổ số lượng điểm có trong một khung hình.

Bảng 3.2 Danh sách các tham số và giá trị tương ứng của chúng.

Ký hiệu	Mô tả	Giá trị	Ký hiệu	Mô tả	Giá trị
f_c	Tần số bắt đầu	$60\,Hz$	v_{max}	Vận tốc tối đa	128
T_c	Thời lượng 1 chirp	$1.3\,\mu s$	v_{res}	Độ phân giải vận tốc	128
B	Băng thông	$3.25\,GHz$	N_{TX}	Số lượng ăng-ten phát	3
S	Góc nghiêng của chirp	$50\,MHz/\mu s$	θ_{res}	Độ phân giải góc	128
N	Số lượng chirp trong 1 khung hình	128	N_{RX}	Số lượng ăng-ten thu	4
d_{res}	Độ phân giải khoảng cách	128	N_P	Số điểm tối ra có thể phát hiện được mỗi khung hình	64



Hình 3.36 Mô tả dữ liệu point-cloud sau khi tiền xử lý trên không gian 3 chiều (a). Hình (b), (c),và (d) lần lượt mô tả dữ liệu nhìn từ phía trước, phía bên cạnh và từ trên cao xuống.

3.1.5. Thống kê dữ liệu

Dữ liệu được sử dụng trong bài bao gồm 2 bộ dữ liệu. Một là bộ dữ liệu MAR, hai là bộ dữ liệu tự thu Radim5 PTIT bao gồm các hành động: duỗi chi trên bên trái, duỗi chi trên bên phải, mở rộng cả hai chi tay trên, lunge phía trước bên trái, lunge phía trước bên phải, squad, lunge bên trái, lunge bên phải, mở rộng chi trái, mở rộng chi phải. Cả hai bộ dữ liệu sẽ được chia thành các tập huấn luyện/ kiểm tra (train/test) theo giao thức 2 cài đặt 2 (protocol 2 setting 2) như nhóm tác giả đã đề cập [7]. Thống kê về tập dữ liệu được trình bài trong bảng 3.3.

Bảng 3.3 Mô tả số lượng hành động, số lượng người thu dữ liệu và tổng số mẫu thu được.

	Tập huấn luyện	Giá trị	Tập kiểm tra	Giá trị
Tập dữ liệu MAR	Số lượng hành động	10	Số lượng hành động	10
	Số lượng người thu dữ liệu	16	Số lượng người thu dữ liệu	4
	Số lượng mẫu	320	Số lượng mẫu	80
	Thời gian trung bình thu mỗi người (giây)	210	Thời gian trung bình thu mỗi người (giây)	210
Tập dữ liệu Radim5 PTIT	Số lượng hành động	5	Số lượng hành động	5
	Số lượng người thu dữ liệu	4	Số lượng người thu dữ liệu	2
	Số lượng mẫu	60	Số lượng mẫu	30
	Thời gian trung bình thu mỗi người (giây)	30	Thời gian trung bình thu mỗi người (giây)	30

3.2. Cài đặt và thực nghiệm

Đồ án tiến hành thực nghiệm với hai bộ dữ liệu MAR và Radim5 PTIT với mục đích đánh giá hiệu quả của các mô hình mạng học sâu với tác vụ ước lượng tư thế người sử dụng tín hiệu radar. Thêm vào đó, đồ án cũng so sánh độ phức tạp và hiệu quả về mặt tính toán giữa các mô hình đề xuất [2.5.1]. Các mô hình thử nghiệm bao gồm AutoEncoder [2.5.2], CNN-LSTM, LSTM-CNN, và Parallel-CNN-LSTM [2.5.3].

Trong khi hai mạng học sâu CNN, AutoEncoder đơn thuần hoạt động và đưa ra dự đoán ước lượng tư thế người dựa trên các điểm dữ liệu gần nhau trong một khung hình thông qua cấu trúc mạng nơ-ron tích chập đơn thuần, thì CNN-LSTM, LSTM-CNN và Parallel-CNN-LSTM được bổ trợ thêm mạng nơ-ron trí nhớ ngắn hạn dài nhằm kết nối các khung hình theo thời gian với nhau. Việc đưa ra dữ đoán lúc này sẽ dựa trên cả các điểm dữ liệu gần nhau và các khung hình chứa tư thế người đã được dữ đoán trước đó.

Với việc xây dựng mô hình học sau, đồ án sử dụng framework Pytorch [24]. Quá trình thực nghiệm thực hiện trên máy tính với cấu hình:

- Vi xử lý: 13th Gen Intel@ Core i9-13900k
- Ram: 64GB
- Ổ cứng: 2,5TB
- Card đồ hoạ: NVIDIA Corporation GA 102GL [RTX A5000]
- Hệ điều hành: Ubuntu 20.04.2 LTS

3.3. Phương pháp đánh giá

Để đánh giá độ chính xác của các mô hình trong tác vụ ước lượng tư thé người, đồ án sử dụng phương pháp đánh giá MAE, MPJPE và PA-MPJPE.

Phương pháp đánh giá MAE

Phương pháp đánh giá MAE được sử dụng để đo lường chênh lệch trung bình giữa các giá trị dữ đoán của mô hình và giá trị thực tế. Phương pháp đánh giá được tính bằng cách lấy tổng giá trị tuyệt đối của sự chênh lệch giữa dự đoán và thực tế, sau đó chia cho số lượng mẫu. Công thức MAE được biểu diễn dưới công thức toán học sau:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J}\left|y_{ij} - \hat{y}_{ij}\right|$$

Với n là tổng số lượng mẫu dữ liệu, J là tổng số lượng các điểm khớp, y_{ij} và $\hat{y}_{ij}$ lần lượt là vị trí khớp dự đoán và vị trí khớp thực tế.

Phương pháp đánh giá MPJPE

Phương pháp MPJPE được sử dụng để đánh giá độ chính xác của mô hình nhận diện và theo dõi các khớp cơ thể trong hình ảnh hay video. Phương pháp này đo lường sự chênh lệch trung bình giữa vị trí dự đoán của mô hình và vị trí thực tế của các khớp cơ thể. MPJPE lấy trung bình sai số Euclidean giữa các điểm dự đoán và thực tế trên mỗi khớp cơ thể. Công thức MPJPE được biểu diễn dưới công thức toán học sau:

$$MPJPE = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J}\left\|y_{ij} - \hat{y}_{ij}\right\|$$

Với n là tổng số lượng mẫu dữ liệu, J là tổng số lượng các điểm khớp, y_{ij} và $\hat{y}_{ij}$ lần lượt là vị trí khớp dự đoán và vị trí khớp thực tế.

Phương pháp đánh giá MPJPE

Phương pháp đánh giá PA-MPJPE là một phương pháp đánh giá hiệu suất của các phương pháp ước tính tư thế 3D của con người. Phương pháp này dựa trên phép biến đổi Procrustes để thu được một bản sao của tư thế ước tính được căn chỉnh với tư thế tham chiếu. Sau đó, sai số trung bình trên mỗi khớp (MPJPE) được tính giữa hai tư thế đã được căn chỉnh này. Công thức PA-MPJPE được biểu diễn dưới công thức toán học sau:

$$PA-MPJPE = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J}\left\|q_{ij} - \hat{q}_{ij}\right\|$$

Với n là tổng số lượng mẫu dữ liệu, J là tổng số lượng các điểm khớp, q_{ij} và $\hat{q}_{ij}$ lần lượt là vị trí khớp dự đoán và vị trí khớp thực tế sau khi thực hiện phép biến đổi Procrustes.

3.4. Kết quả thực nghiệm

Bảng 3.4 Kết quả thực nghiệm của các mô hình trên 2 bộ dữ liệu MAR và Radim5 PTIT.

Mạng nơ-ron	Tập dữ liệu		Thời gian (giây)	MAE (mm)	MPJPE (mm)	PA-MPJPE (mm)
MAR-CNN	MAR	Tập huấn luyện	4.17	91.1	185.5	117.5
		Tập kiểm tra	0.88	145.7	298.5	131.7
	Radim5 PTIT	Tập huấn luyện	1.47	128.7	267.4	121.6
		Tập kiểm tra	0.25	130.2	269.3	122.7
Advanced-MAR-CNN	MAR	Tập huấn luyện	4.21	90.1	183.1	113.6
		Tập kiểm tra	0.89	139.2	295.4	128.2
	Radim5 PTIT	Tập huấn luyện	1.64	130.7	277.4	130.3
		Tập kiểm tra	0.28	131.2	289.9	143.7
CNN-LSTM	MAR	Tập huấn luyện	3.95	96.6	199.6	99.5
		Tập kiểm tra	0.92	132.6	284.9	111.4
	Radim5 PTIT	Tập huấn luyện	1.11	125.7	243.7	124.3
		Tập kiểm tra	0.27	127.1	273.2	133.3
LSTM-CNN	MAR	Tập huấn luyện	4.34	89.3	181.7	112.9
		Tập kiểm tra	0.94	132.8	285.3	111.2
	Radim5 PTIT	Tập huấn luyện	1.15	119.3	222.7	121.7
		Tập kiểm tra	0.29	124.5	234.7	126.1
Parallel-CNN-LSTM	MAR	Tập huấn luyện	4.02	89.9	183.0	115.6
		Tập kiểm tra	0.93	132.8	285.3	111.5
	Radim5 PTIT	Tập huấn luyện	1.11	126.4	247.1	129.6
		Tập kiểm tra	0.27	129.1	283.1	131.1

Nhìn chung, kết quả huấn luyện 5 mô hình với 2 tập dữ liệu MAR và Radim5PTIT khá tốt, kết quả gần xấp xỉ bằng kết quả của 2 tác giả An và Orgas trong bài báo [7]. Đánh giá chung, với lượng dữ liệu lớn hơn khoảng 5 lần, các mô hình được huấn luyện ở dữ liệu MAR cho kết quả tốt và ổn định hơn so với các mô hình được huấn luyện bởi dữ liệu Radim5PTIT. Cùng với đó, các mô hình tích chập kết hợp với trí nhớ ngắn hạn cho kết quả tốt hơn so với 2 mô hình MAR-CNN và Advanced-MAR-CNN sử dụng mạng nơ-ron tích chập thuần.

So sánh kết quả giữa 2 mô hình MAR-CNN và Advanced-MAR-CNN cho thấy, mô hình Advanced-MAR-CNN với việc được thiết kế thêm các lớp tích chập đằng sau, tạo mô hình AutoEncoder nhỏ nhẹ cho kết quả tốt hơn. Với lần lượt 90.1mm, 183.1mm và 113.6mm trên các độ đo MAE, MPJPE và PA-MPJPE cho bộ dữ liệu MAR khi huấn luyện. Kết quả này thấp hơn 1-4mm với từng độ đo cho mô hình MAR-CNN. Còn đối với tập kiểm tra, cả 2 mô hình đều cho kết quả tại các độ đo cả hơn 30mm so với tập huấn luyện nhưng nhìn chung, mô hình Advanced-MAR-CNN vẫn cho kết quả tốt hơn. Còn đối với dữ liệu Radim5-PTIT, mô hình MAR-CNN lại cho kết quả tốt hơn ở cả tập huấn luyện và tập kiểm tra. Điều này có thể lí giải bởi mô hình mạng MAR-CNN nhẹ và ít tham số hơn, cùng với đó dữ liệu tập Radim5-PTIT với số lượng ít hơn nên có thể mô hình MAR-CNN ngẫu nhiên cho kết quả tốt. Trong khi đó, mô hình Advanced-MAR-CNN với cấu trúc phức tạp hơn, cần nhiều dữ liệu hơn để học vậy nên có kết quả chưa tốt tại tập dữ liệu này. Thêm vào đó, tại khía cạnh thời gian thực thi, với tính chất nhẹ và ít tham số, mô hình MAR-CNN có lượng thời gian thực thi ít hơn.

Với các mô hình sử dụng thêm kiến trúc trí nhớ ngắn hạn dài, kết quả nhận được nhìn chung tốt hơn so với 2 mô hình kể trên (MAR-CNN và Advanced-MAR-CNN). Tại tập dữ liệu MAR, cả 3 mô hình CNN-LSTM, LSTM-CNN và Parallel-CNN-LSTM cho kết quả tương tự nhau, với chênh lệch tại mỗi độ đo là không đáng kể. Với mô hình CNN-LSTM cho kết quả tốt nhất tại tập huấn luyện với độ đo PA-MPJPE = 99.5mm và tập kiểm tra với độ đo MAE = 132.6mm. Trong khi đó, mô hình LSTM-CNN lại cho kết quả tốt ở tập huấn luyện với độ đo MAE = 119.3mm và độ đo PA-MPJPE = 111.2 tại tập kiểm tra. Tại tập dữ liệu Radim5-PTIT, mô hình LSTM-CNN cho kết quả tốt nhất ở cả 2 tập huấn luyện và kiểm tra với lần lượt giá trị các độ đo cho tập huấn luyện là 119.3mm, 222.7mm, 121.7mm và cho tập kiểm tra là 124.5mm, 234.7mm, 126.1mm tại các độ đo MAE, MPJPE và PA-MPJPE. Về thời gian thực thi, có thể dễ dàng thấy rằng mô hình LSTM-CNN có thời gian thực thi lâu hơn so với 2 mô hình còn lại. Điều này có thể lí giải bởi dữ liệu sau khi tiền xử lý được tiếp đi qua mạng nơ-ron trí nhớ ngắn hạn dài mà không được giảm kích thước so với các mô hình mạng khác khi đã được đi qua lớp mạng tích chập để giảm kích thước dữ liệu và cô đọng lại các đặc trưng quan trọng. Ngoài ra, với việc thiết kế mạng nơ-ron tích chập đứng trước cũng cho phép làm giảm số lượng tham số trong mạng. Tuy nhiên, với thời gian thực thi lâu hơn, mô hình LSTM-CNN cho kết quả khả quan nhất trong các mô hình được thử nghiệm ở trên. Sự đánh đổi về mặt thời gian này nhìn chung có thể chấp nhận được.

3.5. Tổng kết chương 3

Trong chương 3, đồ án đã trình bày quá trình việc thu thập dữ liệu, các quá trình xử lý dữ liệu và tiến hành thực nghiệm, đưa ra kết quả và nhận xét thực nghiệm trên 2 bộ dữ liệu MAR và Radim5 PTIT.

KẾT LUẬN

Trên cơ sở tìm hiểu và phân tích phương pháp cho bài toán ước lượng tư thế người, đồ án đã đạt được một số kết quả:

- Giới thiệu về bài toán ước lượng tư thế người, các dạng dữ liệu có thể sử dụng cho bài toán và các ứng dụng sau đó của bài toán ước lượng tư thế người.
- Trình bày khái niệm, tính chất và đặc điểm của trí tuệ nhân tạo, học máy và học sâu. Mô tả chi tiết các thành phần mạng nơ-ron đơn giản, các mạng học sâu phổ biến như mạng nơ-ron tích chập, mạng nơ-ron trí nhớ ngắn hạn dài. Đi sâu vào nghiên cứu và ứng dụng mạng nơ-ron nhân tạo để phục vụ cho bài toán ước lượng tư thế người.
- Trình bày quá trình thu thập, xử lý và xây dựng bộ dữ liệu radar cho tác vụ ước lượng tư thế người bao gồm 6 người và 5 hành động.
- Thực hiện các thực nghiệm với các mô hình đề xuất cho ra kết quả tốt. Trình bày các phương pháp đánh giá được sử dụng phổ biến cho bài toán ước lượng tư thế người. Các kết quả thực nghiệm cho thấy các mô hình đề xuấ cho kết quả tốt hơn so với mô hình ban đầu được đề xuất bởi nhóm tác giả An và Orgas.

Hướng phát triển trong tương lai

- Tuy mô hình đề xuất cho kết quả tốt nhưng để so sánh với dạng dữ liệu hình ảnh thì còn khá khiêm tốn. Trong tương lại, cần cải thiện phần mô hình mạnh mẽ hơn nữa, có thể kết hợp thêm với dạng dữ liệu sensor cảm biến từ các thiết bị đồng hộ, điện thoại hoặc dữ liệu hình ảnh để cải thiện chất lượng mô hình hơn nữa.
- Thử nghiệm xây dựng mô hình luân chuyển kiến thức giữa các miền dữ liệu khác nhau để tận dụng các mô hình đã cho ra kết quả tốt trên miền dữ liệu hình ảnh.
- Xây dựng bộ dữ liệu đa dạng và bao quát các hoạt động thường ngày của con người.
- Xây dựng ứng dụng ước lượng tư thế người sử dụng tín hiệu radar hoàn chỉnh và đưa vào ứng dụng thực tế.

TÀI LIỆU THAM KHẢO

[1] Luvizon, D.C., Picard, D., & Tabia, H. (2019). Consensus-Based Optimization for 3D Human Pose Estimation in Camera Coordinates. International Journal of Computer Vision, 130, 869 - 882.

[2] Patil, A.K., Balasubramanyam, A., Ryu, J.Y., B. N., P.K., Chakravarthi, B., & Chai, Y.H. (2020). Fusion of Multiple Lidars and Inertial Sensors for the Real-Time Pose Tracking of Human Motion. Sensors (Basel, Switzerland), 20.

[3] Zhao, M., Li, T., Alsheikh, M.A., Tian, Y., Zhao, H., Torralba, A., & Katabi, D. (2018). Through-Wall Human Pose Estimation Using Radio Signals. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7356-7365.

[4] Yang, Z., Zeng, A., Yuan, C., & Li, Y. (2023). Effective Whole-body Pose Estimation with Two-stages Distillation. ArXiv, abs/2307.15880.

[5] Xu, L., Jin, S., Liu, W., Qian, C., Ouyang, W., Luo, P., & Wang, X. (2022). ZoomNAS: Searching for Whole-Body Human Pose Estimation in the Wild. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45, 5296-5313.

[6] Jiang, T., Lu, P., Zhang, L., Ma, N., Han, R., Lyu, C., Li, Y., & Chen, K. (2023). RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose. ArXiv, abs/2303.07399.

[7] An, S., & Ogras, U.Y. (2021). MARS: mmWave-based Assistive Rehabilitation System for Smart Healthcare. ACM Transactions on Embedded Computing Systems (TECS), 20, 1 - 22.

[8] Chun, S., Park, S., & Chang, J.Y. (2023). Representation Learning of Vertex Heatmaps for 3D Human Mesh Reconstruction From Multi-View Images. 2023 IEEE International Conference on Image Processing (ICIP), 670-674.

[9] Russell, S., & Norvig, P. (1995). Artificial intelligence - a modern

approach: the intelligent agent book. Prentice Hall series in artificial intelligence.

[10] Nathani, N., & Singh, A. (2021). Foundations of Machine Learning. Introduction to AI Techniques for Renewable Energy Systems.

[11] Samuel, A.L. (1967). Some Studies in Machine Learning Using the Game of Checkers. IBM J. Res. Dev., 44, 206-227.

[12] Mitchell, T.M. (1997). Machine learning, International Edition. McGraw-Hill Series in Computer Science.

[13] LeCun, Y., Bengio, Y., & Hinton, G.E. (2015). Deep Learning. Nature, 521, 436-444.

[14] Hassoun, M.H., Intrator, N., Mckay, S., & Christian, W. (1996). Fundamentals of Artificial Neural Networks. Proceedings of the IEEE, 84, 906-.

[15] Hilbe, J.M. (2009). Logistic Regression Models.

[16] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proc. IEEE, 86, 2278-2324.

[17] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60, 84 - 90.

[18] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.

[19] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1-9.

[20] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern

Recognition (CVPR), 770-778.

[21] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9, 1735-1780.

[22] Mikolov, Tomas, Martin Karafiát, Lukáš Burget, Jan Honza Černocký and Sanjeev Khudanpur. “Recurrent neural network based language model.” Interspeech (2010).

[23] Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., Han, S., Luo, P., Zeng, G., & Wang, J. (2022). Context Autoencoder for Self-Supervised Representation Learning. ArXiv, abs/2202.03026.

[24] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Neural Information Processing Systems.