

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC VĂN LANG
—oo—



BÁO CÁO TỐT NGHIỆP

ỨNG DỤNG TRÍ TUỆ NHÂN TẠO TRONG
CẢNH BÁO SỚM BỆNH RỒI LOẠN NHỊP TIM

Sinh viên thực hiện: Phan Nhật Anh

Sinh viên thực hiện: Châu Bảo Nghi

Khoa: Khoa Kỹ Thuật Cơ - Điện Và Máy Tính

Giảng viên hướng dẫn: TS. Trương Quốc Trí

—oo—
TP. HỒ CHÍ MINH, 2024

Tóm tắt nội dung

Hiện nay, bệnh tim mạch là một trong những nguyên nhân hàng đầu gây nguy hiểm cho sức khỏe con người, với tỷ lệ người mắc bệnh ngày càng gia tăng. Một trong những công cụ quan trọng trong việc phát hiện và chẩn đoán các bệnh tim mạch là điện tâm đồ (ECG). Tuy nhiên, tín hiệu ECG có đặc tính chu kỳ và thay đổi theo thời gian, do đó đòi hỏi bác sĩ phải phân tích cẩn thận để phát hiện các bất thường, đặc biệt là rối loạn nhịp tim. Trong bối cảnh này, việc ứng dụng trí tuệ nhân tạo (AI) vào phân tích tín hiệu ECG trở thành một giải pháp tiềm năng, giúp hỗ trợ chẩn đoán chính xác hơn và phát hiện sớm các bệnh lý tim mạch.

Mô hình Long Short-Term Memory đã được chứng minh có khả năng tái lập tín hiệu ECG khá hiệu quả nhờ vào khả năng học các phụ thuộc dài hạn trong chuỗi dữ liệu. Tuy nhiên, mô hình LSTM vẫn gặp phải một số hạn chế trong việc xử lý các pha chuyển tiếp và các thay đổi đột ngột trong tín hiệu ECG, đặc biệt khi dữ liệu huấn luyện có độ đa dạng, phức tạp cao. Để khắc phục hạn chế này, nghiên cứu đề xuất sử dụng mô hình kết hợp giữa 1D-CNN và Bi-LSTM, tận dụng khả năng trích xuất đặc trưng không gian của 1D-CNN và học phụ thuộc thời gian của Bi-LSTM để nâng cao độ chính xác trong tái lập tín hiệu ECG.

Nghiên cứu hướng tới phát triển một hệ thống tự động tái lập tín hiệu ECG bằng các mô hình học sâu kết hợp, nhằm cải thiện chất lượng tín hiệu tái tạo và hỗ trợ phân tích y khoa. Việc ứng dụng các mô hình học sâu kết hợp có thể mang lại những bước tiến quan trọng trong việc nâng cao độ chính xác và hiệu quả của việc chẩn đoán bệnh tim mạch, đặc biệt là trong các thiết bị y tế hiện đại, việc này không chỉ giúp giảm tải cho bác sĩ trong quá trình phân tích tín hiệu ECG mà còn góp phần đáng kể vào việc cải thiện chất lượng chăm sóc sức khỏe tim mạch, đồng thời giảm thiểu rủi ro và chi phí y tế cho bệnh nhân.

Từ khóa: Trí tuệ nhân tạo (AI), Long Short-Term Memory (LSTM), 1D-Convolution Neural Network (1D-CNN), Bidirectional Long Short-Term Memory (Bi-LSTM), Electrocardiogram (ECG), Deep Learning (DL).

Mục lục

1 GIỚI THIỆU	1
1.1 Giới thiệu đề tài	1
1.2 Mục tiêu của đề tài	2
1.3 Phương pháp nghiên cứu	2
1.4 Cấu trúc bài báo cáo	3
2 TỔNG QUAN	4
2.1 Tổng quan tình hình nghiên cứu	4
2.2 Nội dung nghiên cứu của đề tài	6
2.3 Cách tiếp cận - tính sáng tạo của đề tài	8
2.3.1 Cách tiếp cận	8
2.3.2 Tính mới, tính độc đáo, tính sáng tạo	9
3 CƠ SỞ LÝ THUYẾT	10
3.1 Trí tuệ nhân tạo	10
3.2 Machine Learning	12
3.2.1 Supervised Learning	13
3.2.2 Unsupervised Learning	14
3.2.3 Reinforcement learning	15
3.3 Deep Learning	17
3.3.1 Tổng quan và phân loại mạng nơ-ron nhân tạo	18
3.3.2 Perceptron và Multi-Layer Perceptron (MLP)	21
3.4 Convolutional Neural Network - CNN	23
3.4.1 One Dimension Convolution Neural Network (CNN 1D) .	26
3.4.2 Convolution layer	27
3.4.3 Batch normalization layer	29
3.4.4 ReLU Layer	30
3.4.5 Pooling layer	31
3.4.6 Fully Connected Layer	32
3.4.7 Residual Neural Network	33
3.5 Recurrent Neural Network - RNN	35
3.5.1 Gated Recurrent Unit (GRU)	36

3.5.2	Long Short-Term Memory (LSTM)	37
3.5.3	Bidirectional Long Short- Term Memory (Bi-LSTM) . . .	40
3.6	Xây dựng và huấn luyện mô hình AI	42
4	KẾT QUẢ HUẤN LUYỆN MÔ HÌNH	45
4.1	Tín hiệu điện tâm đồ và tập dữ liệu MIT-BIH Arrhythmia	45
4.1.1	Tín hiệu điện tâm đồ (Electrocardiogram – ECG)	45
4.1.2	Tập dữ liệu MIT-BIH Arrhythmia	46
4.2	Mô hình 1D CNN & Bi-LSTM - Tái lập tín hiệu ECG	47
4.2.1	Dữ liệu đầu vào	47
4.2.2	Kiến trúc mô hình CNN-BiLSTM	53
4.2.3	Khởi tạo tham số	55
4.2.4	Huấn luyện mô hình	56
4.2.5	Kết quả và đánh giá	58
4.3	Kết quả huấn luyện	60
4.4	Phân tích và so sánh	65
4.5	Thảo luận	67
5	KẾT LUẬN	69
5.1	Kết quả đạt được	69
5.2	Hướng phát triển trong tương lai	69
TÀI LIỆU THAM KHẢO		70

Danh mục các chữ viết tắt

Chữ viết tắt	Giải thích
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
IoT	Internet of Thing
SVM	Support Vevtor Machine
KNN	K-Nearest Neighbors
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
ECG	Electrocardiogram
GRU	Gated recurrent units
Bi - LSTM	Bidirectional – Long Short Term Memory
RC	Reservoir Computing

Danh mục bảng

Bảng 4.1 Tham số khởi tạo cho mô hình CNN-BiLSTM	56
Bảng 4.2 Thông số chi tiết của các lớp trong mô hình CNN-BiLSTM	57
Bảng 4.3 Chỉ số hiệu suất của hai mô hình trên tập thử nghiệm	62
Bảng 4.4 Chỉ số hiệu suất của hai mô hình trên bản ghi 214	66

Danh mục hình vẽ

Hình 3.1 Mối quan hệ và các giai đoạn phát triển của Trí tuệ nhân tạo, Máy học và Học sâu (Nguồn: NVIDIA)	12
Hình 3.2 Ba loại học máy: Supervised learning, Unsupervised learning, Reinforcement learning (Nguồn: <i>mathworks.com</i>) . . .	13
Hình 3.3 Supervised learning (Nguồn: <i>geeksforgeeks.org</i>)	14
Hình 3.4 Unsupervised learning (Nguồn: https://medium.com) . . .	15
Hình 3.5 Reinforcement learning (Nguồn: https://becominghuman.ai)	16
Hình 3.6 Mạng Deep Learning (Nguồn: towardsdatascience.com) .	17
Hình 3.7 Cấu trúc nơ-ron thần kinh (Nguồn: Stanford University) .	19

Hình 3.8	Mạng nơ-ron nhân tạo đơn giản gồm Lớp đầu vào, Lớp ẩn và Lớp đầu ra (Nguồn: Stanford University)	20
Hình 3.9	Một đoạn của Mạng hồi quy (Nguồn: <i>colah.github.io</i>) . . .	21
Hình 3.10	Mạng hồi quy được trải ra thành nhiều nơ-ron với nhiều đầu vào và đầu ra (Nguồn: <i>colah.github.io</i>)	21
Hình 3.11	Neural Network with Perceptron	22
Hình 3.12	Mạng lan truyền tiến có hai lớp ẩn và một lớp đầu ra . . .	23
Hình 3.13	Kiến trúc mạng Convolution Neural Network cơ bản (Nguồn : Zilliz.com)	24
Hình 3.14	Cấu trúc của mạng CNN 1D (Nguồn: [1])	26
Hình 3.15	Convolution Layer (Nguồn: www.baeldung.com)	28
Hình 3.16	Batch Normalization layer (Nguồn: www.pinecone.io) . .	29
Hình 3.17	Biểu đồ hàm kích hoạt ReLU (Nguồn: <i>www.mygreatlearning.com</i>)	30
Hình 3.18	Pooling layer (Nguồn :wprany.github.io)	31
Hình 3.19	Cấu trúc của lớp Fully Connected	33
Hình 3.20	Sơ đồ của Residual Block (Nguồn : <i>www.researchgate.net</i>)	34
Hình 3.21	Cấu trúc tế bào của mạng RNN (Nguồn: <i>codingninjas.com</i>)	36
Hình 3.22	Kiến trúc GRU(Nguồn: <i>https://appliedsingularity.com</i>) . .	37
Hình 3.23	Kiến trúc Node của LSTM	38
Hình 3.24	Forget Gate (Nguồn: <i>colah.github.io</i>)	39
Hình 3.25	Input Gate (Nguồn: <i>colah.github.io</i>)	39
Hình 3.26	Output Gate (Nguồn: <i>colah.github.io</i>)	39
Hình 3.27	Mạng RNN tiêu chuẩn (Nguồn: <i>colah.github.io</i>)	40
Hình 3.28	Kiến trúc mạng LSTM bốn lớp (Nguồn: <i>colah.github.io</i>) .	40
Hình 3.29	Kiến trúc Bidirectional Long Short- Term Memory (Bi-LSTM) (Nguồn: <i>www.sciencedirect.com</i>)	41
Hình 4.1	Tín hiệu ECG điển hình (Nguồn: <i>www.dieutri.vn</i>)	45
Hình 4.2	Tín hiệu ECG gồm 12 chuyển đạo điện tâm đồ	46
Hình 4.3	Tín hiệu ECG của bản ghi 102 trong tập MIT-BIH Arrhythmia Database, bao gồm kênh V5 và V2	47
Hình 4.4	Tín hiệu ECG thô của bản ghi 102 (100.000 điểm dữ liệu)	48
Hình 4.5	Tín hiệu sau khi tăng cường dữ liệu - không có tính nhất quán giữa các chu kỳ	48
Hình 4.6	Tín hiệu sau khi tăng cường dữ liệu - có tính nhất quán giữa các chu kỳ	49
Hình 4.7	Tập dữ liệu tùy chỉnh sau khi cắt ghép	49
Hình 4.8	Tín hiệu ECG bản ghi 102 sau khi chuẩn hóa	50
Hình 4.9	Tín hiệu ECG gốc của bản ghi 102	50
Hình 4.10	Quá trình làm sạch nhiều tín hiệu ECG từ dữ liệu gốc . .	51
Hình 4.11	Tín hiệu ECG bản ghi 102 sau khi lọc nhiễu	51

Hình 4.12	Đoạn tín hiệu ECG từ 0 đến 300 điểm	52
Hình 4.13	Kiến trúc mô hình CNN-BiLSTM	53
Hình 4.14	Tín hiệu ECG thực tế và tái lập trên tập test gốc bởi mô hình huấn luyện trên tập dữ liệu gốc	61
Hình 4.15	Tín hiệu ECG thực tế và tái lập trên tập test gốc bởi mô hình huấn luyện trên tập dữ liệu tùy chỉnh	61
Hình 4.16	Hàm mất mát qua các epoch trên tập dữ liệu gốc	63
Hình 4.17	Hàm mất mát qua các epoch trên tập dữ liệu tùy chỉnh . . .	63
Hình 4.18	RMSE qua các epoch trên tập dữ liệu gốc	63
Hình 4.19	RMSE qua các epoch trên tập dữ liệu tùy chỉnh	64
Hình 4.20	MAE qua các epoch trên tập dữ liệu gốc	64
Hình 4.21	MAE qua các epoch trên tập dữ liệu tùy chỉnh	64
Hình 4.22	Tín hiệu ECG thực tế và tái lập trên bản ghi 214 (mô hình huấn luyện trên tập gốc)	66
Hình 4.23	Tín hiệu ECG thực tế và tái lập trên bản ghi 214 (mô hình huấn luyện trên tập tùy chỉnh)	66

CHƯƠNG 1 GIỚI THIỆU

1.1 Giới thiệu đề tài

Theo Tổ chức Y tế Thế giới (WHO), các bệnh tim mạch, bao gồm nhồi máu cơ tim và rối loạn nhịp tim, là nguyên nhân hàng đầu gây tử vong trên toàn cầu, chiếm khoảng 31% tổng số ca tử vong [2]. Việc phát hiện và chẩn đoán sớm các vấn đề liên quan đến nhịp tim là vô cùng quan trọng, giúp giảm nguy cơ đột quy, suy tim hoặc tử vong đột ngột. Để giảm thiểu số ca tử vong, nhiều nhà nghiên cứu đã phát triển các công cụ và hệ thống nhằm chẩn đoán nhanh chóng các bệnh rối loạn nhịp tim, mang lại ý nghĩa quyết định đối với việc cải thiện chất lượng chăm sóc sức khỏe tim mạch.

Trong những năm gần đây, trí tuệ nhân tạo (AI) đã tạo nên những bước tiến vượt bậc trong nhiều lĩnh vực, đặc biệt là y học. Với khả năng tự học từ dữ liệu, nhận diện các mẫu phức tạp, và đưa ra dự đoán mà không cần can thiệp thủ công, AI đang trở thành một công cụ đắc lực trong phân tích tín hiệu điện tim đồ (ECG). Các phương pháp học sâu như mạng nơ-ron tích chập (CNN) hay mạng nơ-ron hồi tiếp (RNN), điển hình là Long Short-Term Memory (LSTM), đã chứng minh hiệu quả trong việc tái tạo tín hiệu ECG và cảnh báo sớm các bất thường về nhịp tim.

Nhiều nghiên cứu gần đây đã chỉ ra rằng việc kết hợp các mô hình học sâu như CNN và LSTM có thể cải thiện đáng kể độ chính xác trong việc phân tích tín hiệu ECG và phát hiện các bất thường về nhịp tim. Các mô hình này có thể học và nhận diện các đặc trưng quan trọng từ tín hiệu ECG mà không cần quá nhiều sự can thiệp từ con người, giảm bớt khối lượng công việc cho bác sĩ và nâng cao hiệu quả chẩn đoán. Chẳng hạn, nghiên cứu của Hammad và đồng nghiệp đã sử dụng 2 mô hình là Support Vector Machine (SVM) và K-Nearest Neighbors (KNN) để phân loại các đặc trưng và hình ảnh của nhịp tim với độ chính xác trên 90% [3]. Tuy nhiên, trong quá trình ứng dụng thực tế, các mô hình hình học máy như SVM thường gặp khó khăn khi xử lý dữ liệu ECG với đặc tính phi tuyến性和 độ phức tạp cao, đòi hỏi phần cứng tính toán mạnh mẽ và dễ bị hạn chế trong việc trích xuất đặc trưng so với các phương pháp học sâu. Điều này mở ra cơ hội để các mô hình học sâu tiên tiến hơn được khai thác nhằm vượt qua những giới hạn đó.

Xuất phát từ thực tế trên, nghiên cứu này đề xuất một phương pháp kết hợp mạng nơ-ron tích chập một chiều (1D-CNN) và mạng nơ-ron hồi tiếp hai chiều dài ngắn hạn (Bi-LSTM) để phân tích tín hiệu ECG. Trong đó, 1D-CNN đảm nhiệm việc trích xuất các đặc trưng không gian từ tín hiệu, giúp nhận diện rõ ràng các thành phần sóng quan trọng như P, QRS, và T trong mỗi chu kỳ nhịp tim. Đồng thời, Bi-LSTM, với khả năng xử lý chuỗi thời gian theo cả hai hướng (quá khứ và tương lai), cho phép mô hình nắm bắt các mối quan hệ thời gian phức tạp, chẳng hạn như các biến đổi giữa các chu kỳ bình thường và bất thường. Sự kết hợp này không chỉ cải thiện khả năng tái lập tín hiệu mà còn tăng cường hiệu quả phát hiện các rối loạn nhịp tim như rung nhĩ, nhồi máu cơ tim hay các dạng loạn nhịp khác.

Điểm nổi bật của phương pháp đề xuất nằm ở sự tận dụng đồng thời ưu thế của 1D-CNN trong phân tích không gian và Bi-LSTM trong xử lý phụ thuộc thời gian. Điều này cho phép mô hình theo dõi liên tục các thay đổi trong nhịp tim, từ đó cung cấp dự đoán chính xác hơn về các nguy cơ tim mạch tiềm ẩn. Nghiên cứu hướng đến việc xây dựng một tập dữ liệu thực nghiệm từ cơ sở dữ liệu MIT-BIH Arrhythmia, kết hợp với kỹ thuật tăng cường dữ liệu để mở rộng số lượng mẫu, nhằm đảm bảo mô hình hoạt động hiệu quả ngay cả với các trường hợp hiếm gặp.

Trong bối cảnh y tế hiện đại, việc ứng dụng trí tuệ nhân tạo vào phân tích tín hiệu ECG sẽ hỗ trợ bác sĩ và các chuyên gia tim mạch đưa ra quyết định nhanh chóng và chính xác hơn, từ đó nâng cao hiệu quả chẩn đoán và điều trị. Vì vậy, nghiên cứu này tập trung phát triển một mô hình học sâu kết hợp 1D-CNN và Bi-LSTM, với mục tiêu xây dựng bộ dữ liệu thực nghiệm điện tâm đồ và mô hình tái lập điện tim ở người. Kết quả không chỉ góp phần nâng cao hiệu quả chẩn đoán mà còn mở ra triển vọng cải thiện chất lượng chăm sóc sức khỏe tim mạch, giúp phát hiện bệnh sớm các bệnh về tim.

1.2 Mục tiêu của đề tài

- Tái lập tín hiệu điện tim từ dữ liệu thực của bệnh nhân.
- Xây dựng mô hình nhằm đưa ra cảnh báo sớm các bệnh rối loạn nhịp tim.

1.3 Phương pháp nghiên cứu

- Thu thập, tìm hiểu và phân tích các thông tin liên quan đến nghiên cứu, đưa ra những vấn đề cần giải quyết và hướng nghiên cứu mới để cải thiện mô hình.

- Tiến hành phân tích chi tiết, xây dựng giải pháp cho bài toán bằng các bước như tiền xử lý dữ liệu, trích đoạn đặc trưng, huấn luyện mô hình dữ liệu, và hậu xử lý.
- Xây dựng và kiểm thử mô hình. Sự triển khai của mô hình được thực hiện hoàn toàn bằng ngôn ngữ lập trình Python, sử dụng các thư viện như TensorFlow và PyTorch để xây dựng và huấn luyện mô hình.

1.4 Cấu trúc bài báo cáo

- Chương 1 - Giới thiệu: Giới thiệu đề tài, mục tiêu và vấn đề được đặt ra.
- Chương 2 – Tổng quan: Tình hình nghiên cứu trong nước và ngoài nước. Đặt ra mục tiêu và nội dung cần nghiên cứu sau đó đưa ra hướng tiếp cận và phương pháp nghiên cứu.
- Chương 3 – Cơ sở lý thuyết
- Chương 4 – Huấn luyện và xây dựng mô hình, trình bày kết quả thực nghiệm của mô hình. Sau đó, thảo luận về tiềm năng và đề xuất ý tưởng từ nền tảng đề tài này cho các nghiên cứu tương lai.
- Chương 5 - Kết Luận: Tổng kết những việc đã được thực hiện và đánh giá mô hình. Đồng thời, mở rộng tính khả thi của mô hình và đưa ra hướng nghiên cứu mới trong tương lai.

CHƯƠNG 2 TỔNG QUAN

2.1 Tổng quan tình hình nghiên cứu

Điện tâm đồ (Electrocardiogram – ECG) là một công cụ quan trọng để phân tích hoạt động tim mạch [4, 5, 6]. Ứng dụng trí tuệ nhân tạo (AI) trong việc phát hiện các dấu hiệu bất thường liên quan đến bệnh tim – như rối loạn nhịp, ngắt đột ngột, hoặc đột quy – đã thu hút sự chú ý lớn từ cộng đồng khoa học [7, 8, 9, 10, 11, 12]. Nhiều nghiên cứu đã tận dụng AI để phân loại tín hiệu ECG và nhận diện nhịp tim, đạt được những kết quả đáng chú ý.

Một trong những nghiên cứu đáng chú ý là của Martinez et al. [13] ông cùng đồng nghiệp đã sử dụng mô hình CNN trong lĩnh vực deep learning nhằm phân loại và nhận dạng tín hiệu sinh lý như: điện não (EEG), điện cơ,v.v. Với độ chính xác dao động từ 70% đến 75%. Dù không trực tiếp áp dụng cho ECG, công trình này đã đặt nền móng cho việc sử dụng CNN trong phân tích tín hiệu sinh học, mở đường cho nhóm nghiên cứu khác phát triển trong lĩnh vực nhận dạng và phân loại các tín hiệu sinh lý.

Tiếp đến, nghiên cứu của Hammad và cộng sự (2018) [3] đã áp dụng hai mô hình học máy truyền thống – Support Vector Machine (SVM) và K-Nearest Neighbors (KNN) - để phân loại các đặc trưng và hình ảnh về rối loạn nhịp tim từ tín hiệu ECG, đạt độ chính xác trên 90%. Tuy nhiên, các mô hình này gặp hạn chế:

- Mô hình SVM, KNN đều thuộc lĩnh vực Machine Learning truyền thống, chúng hoạt động dựa trên các thuật toán tính toán đơn giản, thiếu khả năng học sâu từ dữ liệu phức tạp như các mô hình học sâu (Deep Learning).
- Tín hiệu ECG có tính phi tuyến tính, độ phức tạp cao, khiến các mô hình Machine Learning có thể gặp khó khăn trong việc trích xuất các đặc trưng và phân loại so với Deep Learning.
- Khả năng xử lý dữ liệu lớn của các mô hình Machine Learning bị hạn chế. Do đó, các vấn đề về mức độ phân tích phức tạp, cần nhiều dữ liệu.v.v, là vấn đề mà mô hình Machine Learning truyền thống đang gặp phải.

Để vượt qua những giới hạn này, nhiều nhà nghiên cứu đã chuyển sang sử dụng các mô hình học sâu như nghiên cứu của Denun và đồng sự (2019) [7] đã

phát triển mô hình CNN phân loại tín hiệu ECG từ 12 loại rối loạn nhịp tim. Mô hình này sử dụng một lượng dữ liệu lớn, bao gồm 91.323 tín hiệu ECG của 53.549 bệnh nhân và đạt được độ chính xác trên 90%. Điều này chứng tỏ rằng các mô hình học sâu có thể vượt trội trong việc phân loại các tín hiệu ECG phức tạp, đặc biệt là khi phải xử lý một lượng lớn dữ liệu. Dù kết quả cao, phương pháp này sử dụng tín hiệu ECG dạng hình ảnh, dẫn đến một số nhược điểm:

- Việc chuyển tín hiệu ECG từ dạng chuỗi thời gian thành hình ảnh có thể làm mất đi những thông tin quan trọng, như độ phân giải thời gian và các dao động nhỏ giữa các nhịp tim ảnh hưởng đến độ chính xác của mô hình.
- Tín hiệu ECG là chuỗi một chiều và liên tục theo thời gian, việc chuyển đổi thành hình ảnh có thể làm mất các chi tiết nhỏ trong tín hiệu gốc, làm giảm độ phân giải thời gian và làm cho việc phân tích trở nên khó khăn.
- Để thu thập hình ảnh ECG, người bệnh phải đeo thiết bị ECG trong thời gian dài, điều này có thể gây bất tiện và khó khăn cho việc giám sát liên tục.

Để giải quyết vấn đề hạn chế của hình ảnh ECG, Ozal Yildirm và Ru San Tan [14] đã phát triển mô hình 1D-CNN, sử dụng tín hiệu số thay vì hình ảnh. Tín hiệu ECG được thu thập từ máy Holter Monitor và lưu trữ dưới dạng chuỗi số, sau đó mô hình này xử lý các giá trị số mà không cần phải chuyển đổi thành hình ảnh. Mô hình 1D-CNN đạt độ chính xác lên đến 91.33% khi phân loại 17 loại rối loạn nhịp tim, giúp giảm bớt sự bất tiện trong việc thu thập dữ liệu và duy trì được độ chính xác cao khi phân tích tín hiệu. Đây là một giải pháp lý tưởng cho việc triển khai thực tế, đặc biệt trong các thiết bị cầm tay hoặc các hệ thống cảnh báo sớm dựa trên AI. Tuy nhiên, việc sử dụng chuỗi ECG dài hạn vẫn còn một số hạn chế:

- Khi sử dụng chuỗi ECG dài hạn, tín hiệu có thể bị nhiễu do nhiều yếu tố như cử động cơ thể, nhiễu từ thiết bị. Nhieu này làm giảm chất lượng tín hiệu và ảnh hưởng đến độ chính xác của mô hình phân tích.
- Rối loạn nhịp tim chỉ kéo dài trong thời gian ngắn. Việc sử dụng chuỗi ECG dài có thể dẫn đến việc mất đi tính nhất quán và độ chính xác trong việc nhận diện các đặc điểm quan trọng của tín hiệu, làm giảm hiệu quả phân tích.

Để giải quyết những hạn chế trong việc phân tích tín hiệu ECG dài hạn, một hướng tiếp cận mới là sử dụng sự kết hợp giữa mô hình 1D Convolutional Neural Networks (CNN) và Bi-LSTM (Bidirectional Long Short-Term Memory). Mô hình 1D-CNN giúp trích xuất các đặc trưng không gian từ tín hiệu ECG, đặc biệt hiệu quả trong việc xử lý dữ liệu chuỗi thời gian một chiều. Tuy nhiên, tín

hiệu ECG có tính chất phụ thuộc vào các mối quan hệ dài hạn, điều này khiến các mô hình truyền thống khó nhận diện được các biến đổi nhỏ nhưng quan trọng trong chuỗi tín hiệu. Lúc này, Bi-LSTM trở thành một giải pháp hiệu quả vì khả năng học và sử dụng thông tin từ cả hai hướng của chuỗi thời gian, giúp mô hình nhận diện các mẫu tín hiệu ECG phức tạp và rối loạn nhịp tim. Mô hình 1D-CNN giúp trích xuất đặc trưng không gian của tín hiệu ECG, trong khi Bi-LSTM bổ sung khả năng nhận diện các mối quan hệ dài hạn trong chuỗi tín hiệu, từ đó cải thiện khả năng phân loại và phát hiện các rối loạn nhịp tim. Sự kết hợp này tận dụng ưu thế của cả hai: 1D-CNN trích xuất đặc trưng không gian, trong khi Bi-LSTM bổ sung khả năng nhận diện các mối quan hệ thời gian, từ đó cải thiện khả năng tái lập tín hiệu ECG chính xác hơn, đặc biệt trong các đoạn tín hiệu có biến thiên mạnh như rối loạn nhịp tim.

Hướng tiếp cận này không chỉ mang lại tiềm năng nâng cao độ chính xác trong chẩn đoán các bệnh lý tim mạch mà còn mở ra cơ hội áp dụng vào các hệ thống giám sát y tế thời gian thực. Việc phát triển mô hình kết hợp 1D-CNN và Bi-LSTM sẽ có ý nghĩa lớn trong việc hỗ trợ các thiết bị y tế di động, cung cấp cảnh báo sớm và hỗ trợ công tác chăm sóc sức khỏe từ xa, góp phần nâng cao chất lượng cuộc sống cho bệnh nhân.

2.2 Nội dung nghiên cứu của đề tài

Nghiên cứu này tập trung vào việc phát triển các phương pháp tiếp cận mới nhằm nâng cao khả năng tái lập và phân tích tín hiệu điện tâm đồ (ECG) thông qua việc ứng dụng các kỹ thuật học sâu tiên tiến. Các hướng tiếp cận chính được nhóm đề xuất bao gồm sự kết hợp giữa các mô hình học sâu, tối ưu hóa dữ liệu huấn luyện, và cải thiện chất lượng tín hiệu đầu vào. Dưới đây là các nội dung nghiên cứu cụ thể:

- **Phát triển mô hình kết hợp 1D-CNN và Bi-LSTM:**

Nhóm đề xuất xây dựng một mô hình học sâu tích hợp giữa mạng nơ-ron tích chập một chiều (1D-CNN) và các mô hình hồi tiếp, cụ thể là Long Short-Term Memory (LSTM) và Bidirectional LSTM (Bi-LSTM). Trong mô hình này, 1D-CNN đóng vai trò trích xuất các đặc trưng không gian từ tín hiệu ECG, tập trung vào việc nhận diện các thành phần sóng quan trọng như P, QRS, và T – những yếu tố mang thông tin cốt lõi về hoạt động của tim.

Sau đó, các đặc trưng không gian này được chuyển tiếp đến các mô hình hồi tiếp để xử lý theo trực thời gian. Đặc biệt, Bi-LSTM, với khả năng học dữ liệu theo cả hai chiều, được ưu tiên sử dụng để khai thác các mối

quan hệ phụ thuộc dài hạn (long-term dependencies) trong chuỗi tín hiệu ECG.

Điều này cho phép mô hình không chỉ tái lập chính xác các chu kỳ tín hiệu mà còn phát hiện hiệu quả các pha chuyển tiếp giữa các trạng thái bình thường và bệnh lý, chẳng hạn như sự thay đổi đột ngột trong nhịp tim hoặc các mẫu bất thường hiếm gặp. Việc kết hợp này giúp cải thiện đáng kể độ chính xác trong cả hai nhiệm vụ: tái lập tín hiệu và nhận diện rối loạn nhịp tim, vượt trội so với các phương pháp chỉ sử dụng một loại mô hình đơn lẻ.

- **Huấn luyện mô hình với tập dữ liệu quy mô lớn:**

Một hướng tiếp cận quan trọng khác là sử dụng tập dữ liệu huấn luyện quy mô lớn, với 100.000 điểm dữ liệu ECG được thu thập và tiền xử lý cẩn thận. So với các nghiên cứu trước đây, chẳng hạn như Kiranyaz et al. (2016) [15], vốn thường sử dụng các tập dữ liệu nhỏ hơn từ MIT-BIH Arrhythmia (khoảng vài nghìn mẫu mỗi bản ghi), quy mô dữ liệu lớn trong nghiên cứu này mang lại lợi thế đáng kể. Sự gia tăng số lượng dữ liệu giúp tăng cường độ đa dạng của các mẫu tín hiệu, bao gồm cả tín hiệu bình thường và bất thường, từ đó cho phép mô hình học được các đặc trưng phong phú và phức tạp hơn. Đồng thời, điều này góp phần giảm thiểu hiện tượng quá khớp (overfitting) – một vấn đề phổ biến khi huấn luyện trên các tập dữ liệu hạn chế.

Kết quả là, mô hình đạt được khả năng tổng quát hóa tốt hơn, cho phép áp dụng hiệu quả trên các bộ dữ liệu mới hoặc trong các điều kiện thực tế đa dạng. Ngoài ra, nhóm cũng cân nhắc đến việc cân bằng tỷ lệ giữa các mẫu bình thường và bất thường trong tập dữ liệu để đảm bảo mô hình không bị thiên lệch trong quá trình học.

- **Tăng cường chất lượng tín hiệu thông qua tiền xử lý nhiễu:**

Trong thực tế, các tín hiệu ECG thường bị ảnh hưởng bởi nhiều loại nhiễu khác nhau, chẳng hạn như nhiễu từ cơ thể (muscle artifacts), nhiễu điện lưới (power line interference), hoặc nhiễu do chuyển động của bệnh nhân. Những yếu tố này có thể làm sai lệch đặc trưng của tín hiệu, gây khó khăn cho mô hình trong việc học và tái lập chính xác.

Do đó, nhóm đề xuất áp dụng các phương pháp xử lý nhiễu hiệu quả trong giai đoạn tiền xử lý dữ liệu. Kỹ thuật được sử dụng là lọc thông thấp (low-pass filtering) để loại bỏ nhiễu tần số cao và chuẩn hóa giá trị tín hiệu để đảm bảo tính đồng nhất. Quá trình này không chỉ làm sạch dữ liệu mà còn tăng cường chất lượng đầu vào cho mô hình, giúp các lớp 1D-CNN và Bi-LSTM tập trung vào việc học các đặc trưng thực sự có ý nghĩa thay

vì bị ảnh hưởng bởi các thành phần nhiễu. Kết quả là, hiệu suất tái lập tín hiệu được cải thiện đáng kể, đặc biệt trong các điều kiện thực tế nơi tín hiệu ECG thường không lý tưởng.

Những hướng tiếp cận trên không chỉ thể hiện sự đổi mới trong cách kết hợp các mô hình học sâu mà còn giải quyết các thách thức thực tiễn trong xử lý tín hiệu ECG. Bằng cách tối ưu hóa cả kiến trúc mô hình, quy mô dữ liệu, và chất lượng tín hiệu, nghiên cứu này đặt nền tảng cho việc phát triển các giải pháp AI hiệu quả hơn trong việc hỗ trợ chẩn đoán và giám sát bệnh lý tim mạch.

2.3 Cách tiếp cận - tính sáng tạo của đề tài

2.3.1 Cách tiếp cận

Trong nghiên cứu này, nhóm đã lựa chọn một cách tiếp cận có hệ thống nhằm phát triển mô hình tái lập tín hiệu điện tâm đồ (ECG) với độ chính xác cao và khả năng ứng dụng thực tiễn. Cụ thể, quá trình thực hiện được tiến hành như sau:

- **Thu thập dữ liệu:** Nhóm đã khai thác tập dữ liệu thực nghiệm từ cơ sở dữ liệu MIT-BIH Arrhythmia, một nguồn tài nguyên uy tín và được sử dụng rộng rãi trong các nghiên cứu về tín hiệu ECG. Tập dữ liệu này bao gồm 100.000 mẫu tín hiệu ECG thu thập từ các bệnh nhân, với quy mô lớn hơn đáng kể so với các nghiên cứu trước đây, tạo điều kiện thuận lợi cho việc huấn luyện mô hình học sâu. Sự gia tăng về số lượng mẫu không chỉ giúp cải thiện độ đa dạng của dữ liệu mà còn tăng khả năng học các đặc trưng phức tạp trong tín hiệu ECG.
- **Tiền xử lý tín hiệu:** Để đảm bảo chất lượng dữ liệu đầu vào cho mô hình, tín hiệu ECG được tiền xử lý bằng bộ lọc thông thấp (low-pass filter) nhằm loại bỏ nhiễu tần số cao (như nhiễu điện lưới 50/60 Hz). Sau đó, tín hiệu được chuẩn hóa về khoảng giá trị cố định. Quá trình này giúp làm sạch dữ liệu, hỗ trợ mô hình tập trung vào việc học các đặc trưng quan trọng như sóng P, QRS, và T của tín hiệu ECG.
- **Huấn luyện và tối ưu hóa mô hình:** Sau khi dữ liệu được tiền xử lý, mô hình được huấn luyện và kiểm tra thông qua các chỉ số đánh giá hiệu suất như Mean Absolute Error (MAE), Mean Squared Error (MSE), và Root Mean Squared Error (RMSE). Các chỉ số này được sử dụng để đo lường mức độ sai lệch giữa tín hiệu dự đoán và tín hiệu thực tế, từ đó đánh giá khả năng tái lập của mô hình. Quá trình tối ưu hóa được thực hiện bằng cách tinh chỉnh các siêu tham số (hyperparameters) như tốc độ học

(learning rate), kích thước batch (batch size), và số lượng epoch, nhằm nâng cao hiệu suất và khả năng tổng quát hóa của mô hình trên dữ liệu mới.

Cách tiếp cận này không chỉ đảm bảo tính khoa học trong việc xây dựng mô hình mà còn tạo nền tảng vững chắc cho việc đánh giá hiệu quả ứng dụng trong thực tế, đặc biệt trong lĩnh vực phát hiện sớm rối loạn nhịp tim.

2.3.2 Tính mới, tính độc đáo, tính sáng tạo

Nghiên cứu này mang lại những đóng góp mới mẻ và sáng tạo trong việc ứng dụng trí tuệ nhân tạo (AI) vào xử lý tín hiệu ECG, thông qua việc kết hợp các phương pháp học sâu tiên tiến và cách tiếp cận đa lớp. Các điểm nổi bật về tính mới và sáng tạo được trình bày như sau:

- **Phương pháp tiếp cận đa lớp:** Khác với các nghiên cứu truyền thống chỉ dùng một mô hình duy nhất, chúng tôi kết hợp mạng nơ-ron tích chập một chiều (1D-CNN) và Bidirectional LSTM (Bi-LSTM) để đồng thời tái lập tín hiệu ECG và phát hiện bất thường. Dữ liệu ECG được tiền xử lý kỹ lưỡng (lọc nhiễu, chuẩn hóa) trước khi đưa vào 1D-CNN để trích xuất đặc trưng không gian như sóng P, QRS, T. Sau đó, Bi-LSTM xử lý các đặc trưng theo trực thời gian, học các mối quan hệ phụ thuộc lâu dài để tái hiện chính xác chu kỳ tín hiệu.
- **Khai thác dữ liệu hai chiều với Bi-LSTM:** Điểm sáng tạo nằm ở việc sử dụng Bi-LSTM thay vì LSTM một chiều. Bi-LSTM học dữ liệu theo cả hai hướng (quá khứ và tương lai), giúp nhận diện tốt hơn các pha chuyển tiếp và biến thiên phức tạp trong tín hiệu ECG, chẳng hạn như nhịp nhanh thất hoặc ngoại tâm thu, vốn khó phát hiện nếu chỉ phân tích một chiều.
- **Ứng dụng thực tiễn trong y học:** Sự kết hợp 1D-CNN và Bi-LSTM không chỉ nâng cao độ chính xác trong tái lập và phát hiện bất thường về nhịp tim mà còn hỗ trợ chẩn đoán sớm các bệnh tim mạch. Phương pháp này có tiềm năng ứng dụng trong hệ thống giám sát sức khỏe từ xa, giúp bác sĩ đưa ra quyết định nhanh chóng và cải thiện chăm sóc bệnh nhân.

Những điểm mới mẻ và sáng tạo này không chỉ khẳng định giá trị khoa học của nghiên cứu mà còn đặt nền tảng cho các ứng dụng thực tiễn trong tương lai, đặc biệt trong bối cảnh nhu cầu về các giải pháp y tế thông minh ngày càng gia tăng.

CHƯƠNG 3 CƠ SỞ LÝ THUYẾT

3.1 Trí tuệ nhân tạo

Trong thập kỷ gần đây, Trí tuệ nhân tạo (Artificial Intelligence - AI) đã trở thành một lĩnh vực nghiên cứu và ứng dụng có sức ảnh hưởng mạnh mẽ trên nhiều mặt của cuộc sống con người. Thuật ngữ "Trí tuệ nhân tạo" được đưa ra vào năm 1956 trong một hội nghị chuyên ngành về Khoa học máy tính và từ đó, nó đã trở thành một phần không thể tách rời trong cuộc sống hiện đại. Trên toàn thế giới, cũng như ở Việt Nam, Trí tuệ nhân tạo đóng vai trò quan trọng trong nhiều lĩnh vực như Khoa học Dữ liệu, Kỹ thuật phần mềm, Robotic và điều khiển thông minh. Đây là một lĩnh vực nghiên cứu và ứng dụng mang tính đột phá, tập trung vào việc phát triển máy tính có khả năng thực hiện bởi con người. Điểm mấu chốt của AI là khả năng của máy tính “tư duy” và “học” từ dữ liệu, từ đó tự động hoặc điều chỉnh để thực hiện nhiệm vụ cụ thể mà không cần sự can thiệp của con người. AI được xem là một trong những mảnh ghép quan trọng của cách mạng công nghiệp 4.0, Trí tuệ nhân tạo đã và đang thay đổi cách chúng ta sống, làm việc và tương tác với thế giới xung quanh.

Sự phát triển nhanh chóng của các thuật toán Máy học (Machine Learning) và Học sâu (Deep Learning) đã làm cho Trí tuệ nhân tạo trở nên ngày càng mạnh mẽ và linh hoạt hơn trong việc ứng dụng vào các lĩnh vực khác nhau trong khoa học và cuộc sống hàng ngày. Các thuật toán này cho phép máy tính học từ dữ liệu và trải nghiệm để nâng cao hiệu suất và độ chính xác của các nhiệm vụ. Các ứng dụng thông minh như Siri trên thiết bị của Apple, Google Assistant trên nền tảng Google, hoặc thiết bị nhận diện và xử lý giọng nói Alexa của Amazon đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta. Bên cạnh đó, các tính năng dịch thuật tự động của ứng dụng Google Translate hay kiểm tra ngữ pháp tiếng Anh trên phần mềm Grammarly cũng là những minh chứng rõ ràng cho sự phát triển và ứng dụng của Trí tuệ nhân tạo trong lĩnh vực xử lý ngôn ngữ tự nhiên.

Ngoài ra, Trí tuệ nhân tạo cũng được áp dụng rộng rãi trong lĩnh vực xử lý hình ảnh, nơi mà các thiết bị như máy ATM tích hợp camera thông minh có thể nhận dạng chính xác khách hàng thông qua đặc trưng về khuôn mặt, mống mắt. Thậm chí, các công ty công nghệ hàng đầu như Tesla, Waymo, hay Uber đang phát triển và thương mại hóa xe tự hành, sử dụng các ứng dụng của Trí tuệ nhân

tạo như nhận dạng làn đường, biển báo, con người và vật cản. Để thực thi các nhiệm vụ cụ thể như trên, các ứng dụng Trí tuệ nhân tạo thường được xây dựng dựa trên các mô hình tính toán, trong đó các thuật toán Máy học đóng vai trò quan trọng. Những mô hình này được lập trình để khai thác và sử dụng nguồn dữ liệu lớn đã được thu thập trước. Các thuật toán này, đặc biệt là thuật toán Deep Learning, đã mở ra nhiều cơ hội mới trong việc tạo ra các ứng dụng Trí tuệ nhân tạo phức tạp và hiệu quả hơn. Và đặc biệt với sự kết hợp giữa AI và Internet of Thing(IoT) cũng đang mở ra những triển vọng mới trong việc tạo ra các hệ thống thông minh và tự động hóa các quy trình.

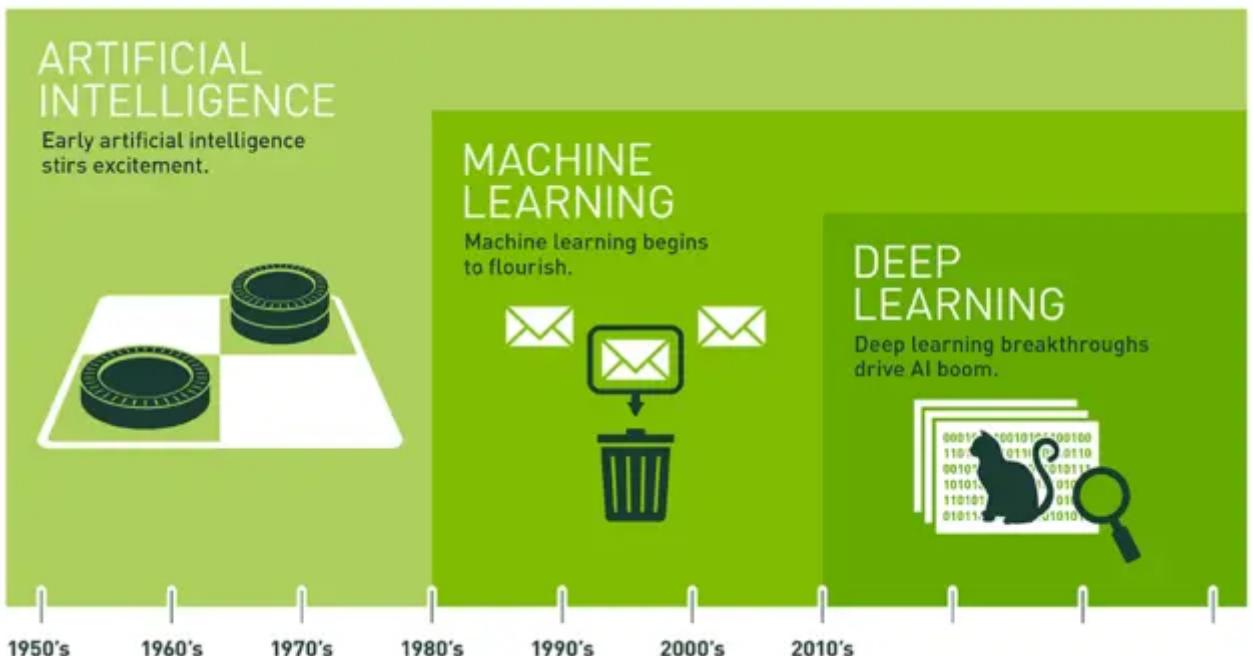
Tóm lại, Trí tuệ nhân tạo đang trải qua một giai đoạn phát triển đáng kinh ngạc, với sự tiến bộ không ngừng của các mô hình Deep Learning và sự sẵn có của dữ liệu lớn. Những tiến bộ này đã mở ra cánh cửa cho nhiều ứng dụng thông minh mới, đóng góp tích cực vào cuộc sống và công nghiệp của chúng ta.

Để hiểu sâu hơn về cách các ứng dụng AI hoạt động và thực thi công việc cụ thể, chúng ta cần nhìn vào hai phân nhánh quan trọng: Machine Learning và Deep Learning.

Hình 3.1 Mô tả mối quan hệ và các giai đoạn phát triển của Trí tuệ nhân tạo, Máy học và Học sâu. Một trong những lĩnh vực chính của AI là Machine Learning, nơi máy tính được lập trình để học từ dữ liệu và trải qua trải nghiệm để cải thiện hiệu suất. Thuật ngữ "Machine Learning" bao gồm một loạt các phương pháp và kỹ thuật để phân tích và học từ dữ liệu, từ đó tự động hóa quá trình ra quyết định và dự đoán. Từ những năm 1980, Machine Learning đã phát triển nhanh chóng nhằm mục đích làm phong phú các ứng dụng mô phỏng trí thông minh của con người. Một trong những yếu tố chính góp phần vào sự phát triển của máy học là sự tăng trưởng vượt bậc của dữ liệu, đặc biệt là dữ liệu lớn(Big Data). Dữ liệu được tạo ra từ hoạt động của xã hội và doanh nghiệp ngày càng nhiều và phức tạp, tạo thành các tập dữ liệu lớn, cung cấp nguồn cung cấp chính cho các thuật toán Machine Learning .

Những năm 2010 là thời điểm bùng nổ của các mô hình học sâu (Deep Learning), Deep Learning là một phân nhánh của Machine Learning, tập trung vào việc sử dụng mạng nơ-ron nhân tạo để thực hiện hóa trí tuệ con người trong các tác vụ cụ thể. Các mô hình Deep Learning có khả năng sử dụng tập dữ liệu lớn và mô phỏng các công việc phức tạp của con người. Một ví dụ điển hình là ứng dụng Alphago của DeepMind, đã đánh bại kỳ thủ cờ vây hàng đầu thế giới, Lee Sedol vào năm 2016. Sự thành công của Alphago đã chứng minh khả năng của Deep Learning trong việc giải quyết các nhiệm vụ phức tạp và đòi hỏi trí tuệ nhân tạo. Sự tiến bộ trong Machine Learning và Deep Learning đang mở ra

những triển vọng mới và tiềm năng không giới hạn cho tương lai của AI. Với khả năng sử dụng Big Data và sự phát triển không ngừng của công nghệ, chúng ta có thể kỳ vọng vào những ứng dụng AI ngày càng phong phú và tiên tiến, từ giải quyết vấn đề toàn cầu cho đến cải thiện cuộc sống hàng ngày của mỗi cá nhân.



Hình 3.1: Mối quan hệ và các giai đoạn phát triển của Trí tuệ nhân tạo, Máy học và Học sâu (Nguồn: NVIDIA)

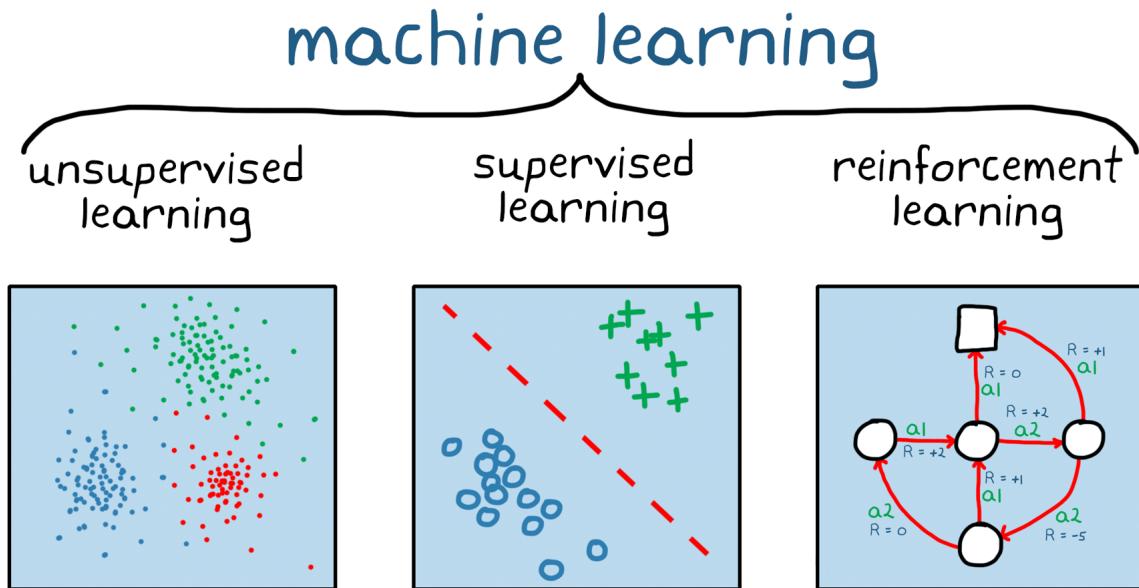
3.2 Machine Learning

Machine Learning (ML) là một lĩnh vực đang phát triển mạnh mẽ trong lĩnh vực trí tuệ nhân tạo (AI), với tiềm năng lớn trong việc giải quyết các vấn đề phức tạp và tạo ra các ứng dụng thông minh trong nhiều lĩnh vực khác nhau. Tính chất chính của Machine Learning là khả năng học hỏi từ dữ liệu mà không cần phải được lập trình cụ thể cho từng trường hợp, điều này mang lại khả năng tự động cải thiện hiệu suất của các hệ thống thông qua trải nghiệm với dữ liệu. Một trong những yếu tố quan trọng nhất của Machine Learning là dữ liệu. Dữ liệu đóng vai trò là nguồn cung cấp thông tin cho mô hình và ảnh hưởng lớn đến khả năng học của nó. Việc thu thập, xử lý và chuẩn bị dữ liệu đòi hỏi sự chăm chỉ và kỹ lưỡng để đảm bảo chất lượng và tính đáng tin cậy của dữ liệu đầu vào![16].

Các thuật toán Machine Learning đa dạng, từ học có giám sát, học không giám sát đến học củng cố, mỗi loại thuật toán được thiết kế để giải quyết các

loại vấn đề khác nhau. Bằng cách sử dụng các phương pháp như hồi quy tuyến tính, phân loại, mạng nơ-ron, và các thuật toán khác, các nhà nghiên cứu và nhà phát triển có thể tạo ra các mô hình có khả năng dự đoán, phân tích và phát hiện ra các mẫu ẩn trong dữ liệu. Quá trình huấn luyện và kiểm định mô hình là bước quan trọng trong Machine Learning, trong đó mô hình được đào tạo trên tập dữ liệu huấn luyện và sau đó được kiểm tra trên tập dữ liệu kiểm định để đánh giá hiệu suất của nó. Qua các vòng lặp của quá trình này, mô hình có thể được điều chỉnh và cải thiện để đạt được hiệu suất tốt nhất.

Cuối cùng, sau khi mô hình đã được huấn luyện và kiểm định, nó có thể được triển khai vào môi trường thực tế để thực hiện các dự đoán và tác động đến quyết định. Machine Learning đã và đang được áp dụng rộng rãi trong nhiều lĩnh vực như công nghệ, y học, tài chính và marketing, mang lại những tiện ích và giải pháp thông minh cho xã hội và doanh nghiệp. Với sự phát triển không ngừng của dữ liệu và công nghệ, Machine Learning sẽ tiếp tục đóng vai trò quan trọng trong việc giải quyết các vấn đề phức tạp và tạo ra những ứng dụng mới tiềm năng.

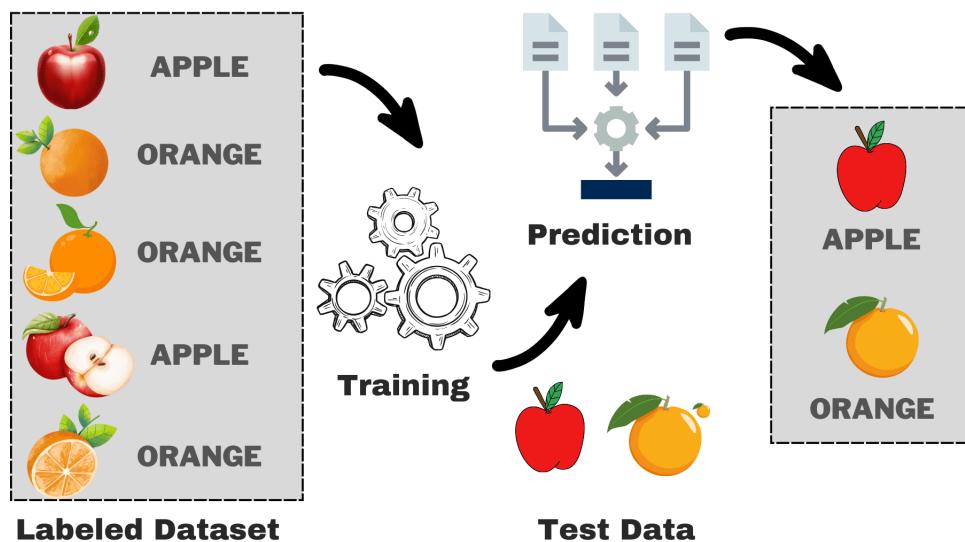


Hình 3.2: Ba loại học máy: Supervised learning, Unsupervised learning, Reinforcement learning (Nguồn: mathworks.com)

3.2.1 Supervised Learning

Học có giám sát là một trong những phương pháp quan trọng nhất trong lĩnh vực Machine Learning, nơi mà mô hình học từ các tập dữ liệu được gắn nhãn để dự đoán đầu ra cho các dữ liệu mới. Quá trình này bắt đầu bằng việc thu thập và

chuẩn bị dữ liệu, trong đó dữ liệu phải được xử lý và làm sạch để loại bỏ nhiễu và chuẩn hóa [17]. Sau đó, một mô hình Machine Learning được chọn và huấn luyện trên tập dữ liệu huấn luyện, trong đó mô hình cố gắng tìm ra các mẫu và quy luật trong dữ liệu. Một phần quan trọng trong quá trình này là đánh giá mô hình, thông qua việc kiểm tra hiệu suất của nó trên một tập dữ liệu kiểm định hoặc thử nghiệm. Dựa trên kết quả của việc đánh giá, mô hình có thể được tinh chỉnh để cải thiện hiệu suất. Cuối cùng, mô hình được triển khai vào môi trường thực tế để thực hiện các dự đoán hoặc tác động đến quyết định.



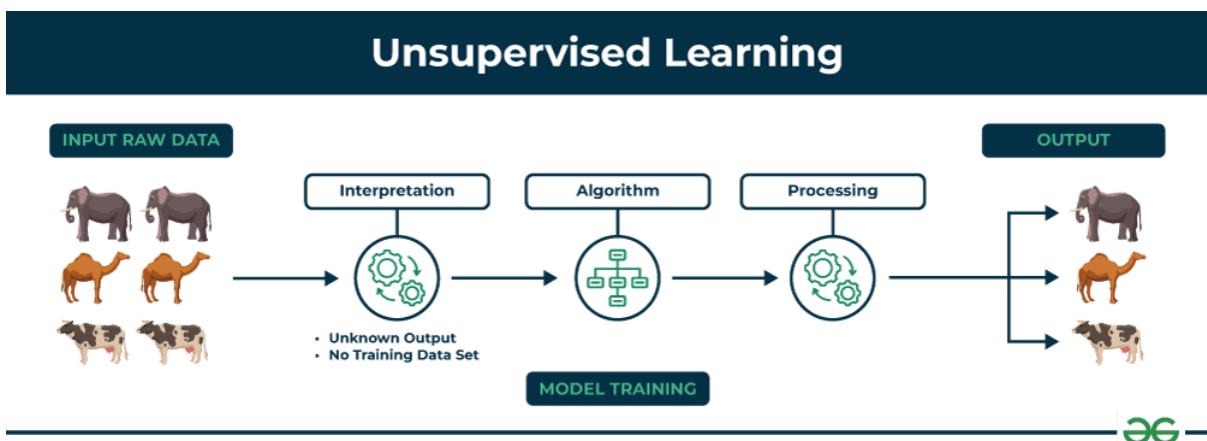
Hình 3.3: Supervised learning (Nguồn: [geeksforgeeks.org](https://www.geeksforgeeks.org/))

Học có giám sát có ứng dụng rộng rãi trong nhiều lĩnh vực như phân loại email, dự đoán thị trường tài chính, nhận dạng khuôn mặt, và nhiều lĩnh vực khác. Đây là một công cụ quan trọng trong việc giải quyết các vấn đề phức tạp và tạo ra các ứng dụng thông minh và tự động hóa. Tuy nhiên, việc thành công trong học có giám sát đòi hỏi sự cẩn trọng trong việc xử lý dữ liệu, lựa chọn mô hình phù hợp, và đánh giá hiệu suất mô hình một cách kỹ lưỡng. Với sự phát triển của dữ liệu và công nghệ, học có giám sát sẽ tiếp tục đóng vai trò quan trọng trong việc tạo ra các giải pháp thông minh và hiệu quả cho các vấn đề thực tế.

3.2.2 Unsupervised Learning

Học không giám sát cũng là một trong những phương pháp quan trọng trong Machine Learning, nơi mà mô hình học từ dữ liệu không có nhãn mục tiêu để tìm ra các mẫu, cấu trúc hoặc mối quan hệ ẩn trong dữ liệu một cách tự động

[18]. Quá trình này bắt đầu từ việc thu thập và chuẩn bị dữ liệu, trong đó dữ liệu được xử lý và làm sạch để loại bỏ nhiễu và chuẩn hóa. Sau đó, một loạt các mô hình Machine Learning có thể được chọn và huấn luyện trên tập dữ liệu không có nhãn, với mục tiêu là tìm ra cấu trúc ẩn trong dữ liệu mà không cần có sự can thiệp từ con người. Trong quá trình này, mô hình cố gắng tự động phát hiện ra các mẫu và quy luật trong dữ liệu một cách không giám sát, không cần sự hướng dẫn từ dữ liệu được gắn nhãn. Một số mô hình phổ biến trong học không giám sát bao gồm phân cụm (clustering), giảm chiều dữ liệu (dimensionality reduction), và phát hiện bất thường (anomaly detection).



Hình 3.4: Unsupervised learning (Nguồn: <https://medium.com>)

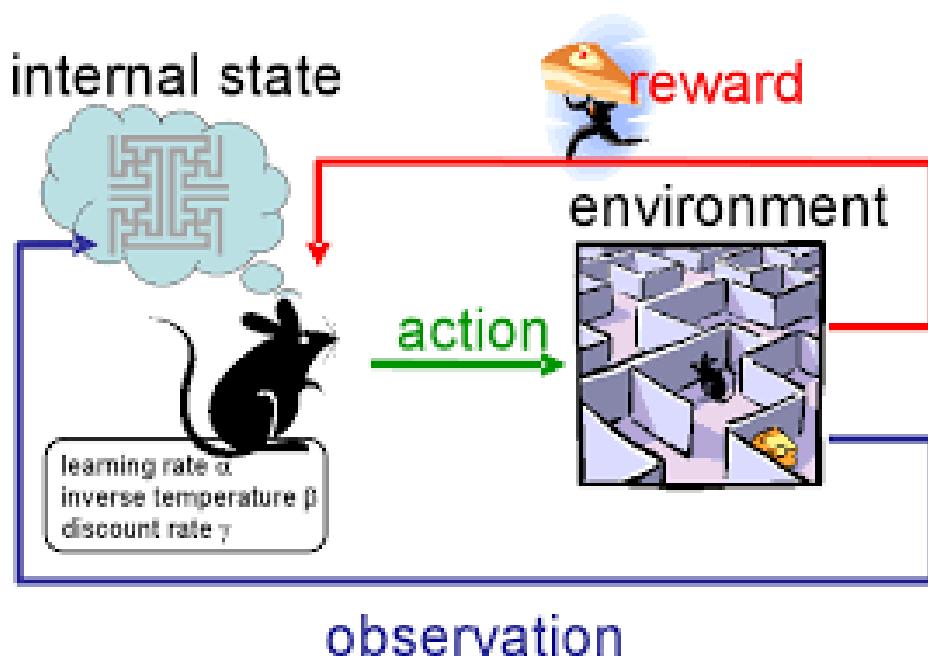
Đánh giá hiệu suất của mô hình trong học không giám sát thường phức tạp hơn so với học có giám sát, vì không có dữ liệu gắn nhãn để so sánh. Tuy nhiên, các phương pháp như kiểm tra giả thuyết, kiểm tra ổn định, và sự nhận biết bằng mắt của con người thường được sử dụng để đánh giá mức độ hiệu quả của mô hình.

Học không giám sát có nhiều ứng dụng trong thực tế, bao gồm phân tích dữ liệu, nhận dạng mẫu, xử lý ngôn ngữ tự nhiên, và nhiều lĩnh vực khác. Với sự phát triển của dữ liệu và công nghệ, học không giám sát sẽ tiếp tục đóng vai trò quan trọng trong việc khám phá và tìm hiểu cấu trúc của dữ liệu một cách tự động và hiệu quả.

3.2.3 Reinforcement learning

Học tăng cường, nơi mà agent(tác nhân) học từ kinh nghiệm tương tác với môi trường để đạt được một mục tiêu cụ thể [19]. Trong học tăng cường, agent thực hiện các hành động trong một môi trường và nhận được phản hồi từ

môi trường dưới dạng phần thưởng hoặc hình phạt. Mục tiêu của agent là tối đa hóa tổng số phần thưởng nhận được trong thời gian dài. Các yếu tố chính của học tăng cường bao gồm agent, môi trường, hành động và phần thưởng. Agent là thực thể hoặc mô hình được huấn luyện để thực hiện hành động trong môi trường. Môi trường là nơi mà agent tương tác và nhận phản hồi từ. Hành động là các quyết định mà agent có thể thực hiện, trong khi phần thưởng là tín hiệu phản hồi từ môi trường để thông báo về chất lượng của các hành động. Các thuật toán phổ biến trong học tăng cường bao gồm Q-Learning, Deep Q-Networks (DQN), và Policy Gradient Methods. Những phương pháp này đều cố gắng tìm ra chính sách tối ưu để agent đạt được mục tiêu của mình.



Hình 3.5: Reinforcement learning (Nguồn: <https://becominghuman.ai>)

Học tăng cường có rất nhiều ứng dụng thực tế, từ trò chơi điện tử đến điều khiển robot, tự động lái xe và quản lý tài nguyên. Các kết quả nghiên cứu trong lĩnh vực này không chỉ giúp cải thiện hiệu suất của các hệ thống tự động mà còn mở ra những triển vọng mới trong việc giải quyết các vấn đề phức tạp và thách thức của thế giới thực. Với sự phát triển của công nghệ và nhu cầu ngày càng tăng trong các lĩnh vực như trí tuệ nhân tạo và robot, học tăng cường sẽ tiếp tục đóng vai trò quan trọng trong việc tiến xa hơn trong việc xây dựng các hệ thống thông minh và tự động.

3.3 Deep Learning

Đối với việc nghiên cứu và áp dụng trong thực tế, Deep Learning đang trở thành một trong những lĩnh vực nổi bật của trí tuệ nhân tạo (AI). Deep Learning là một phần con của Machine Learning, một nhánh của trí tuệ nhân tạo tập trung vào việc huấn luyện các thuật toán để học từ dữ liệu và đưa ra dự đoán hoặc quyết định. Điểm đặc biệt của Deep Learning so với các thuật toán Machine Learning truyền thống là việc sử dụng mạng neural sâu, được lấy cảm hứng từ cấu trúc và chức năng của não người [20].

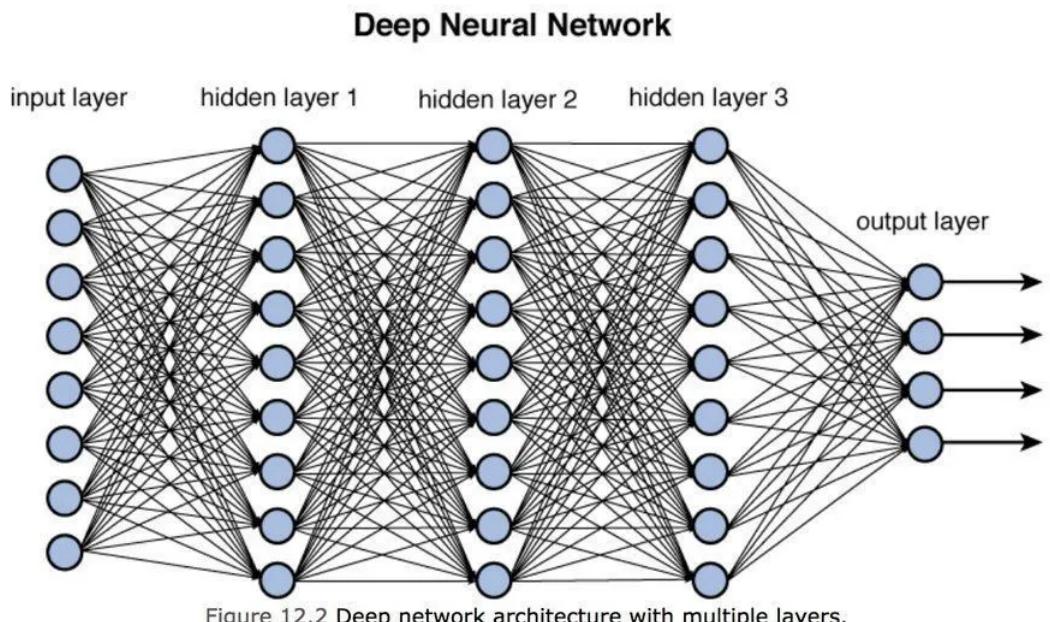


Figure 12.2 Deep network architecture with multiple layers.

Hình 3.6: Mạng Deep Learning (Nguồn: towardsdatascience.com)

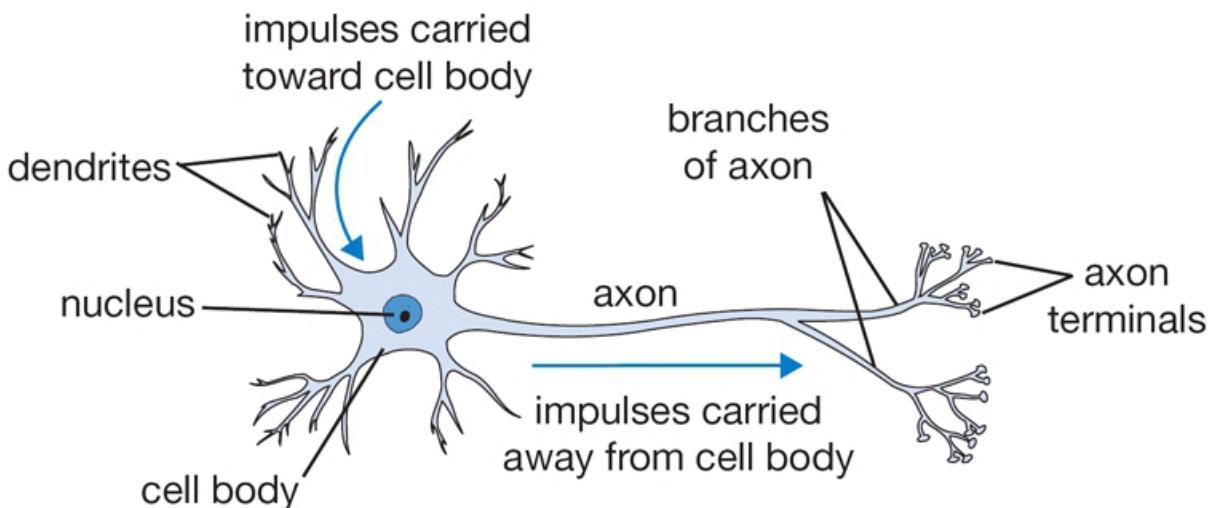
Mạng neural sâu bao gồm các lớp liên kết với nhau, mỗi lớp chứa các đơn vị tính toán gọi là neuron, mô phỏng cấu trúc của não người. Các kiến trúc Deep Learning phổ biến bao gồm Mạng Neural Tổng Hợp (FNN), Mạng Neural Tích Chập (CNN), Mạng Neural Hồi Quy (RNN), và nhiều kiến trúc khác. Quá trình huấn luyện của Deep Learning bao gồm lan truyền tiến (forward propagation), tính toán lỗi (loss calculation), và lan truyền ngược (backpropagation) để điều chỉnh trọng số và tham số mô hình. Các ứng dụng của Deep Learning rất đa dạng, từ xử lý ảnh và video, xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, đến dự đoán thị trường tài chính và phát hiện bệnh tật trong lĩnh vực y tế. Tuy nhiên, việc huấn luyện mô hình Deep Learning đòi hỏi lượng dữ liệu lớn và tài nguyên tính toán cao, đặc biệt là sử dụng GPU hoặc TPU.

Mặc dù Deep Learning mang lại những tiềm năng lớn trong nhiều lĩnh vực, nhưng nó cũng đặt ra những thách thức như tính diển giải của mô hình và độ tin

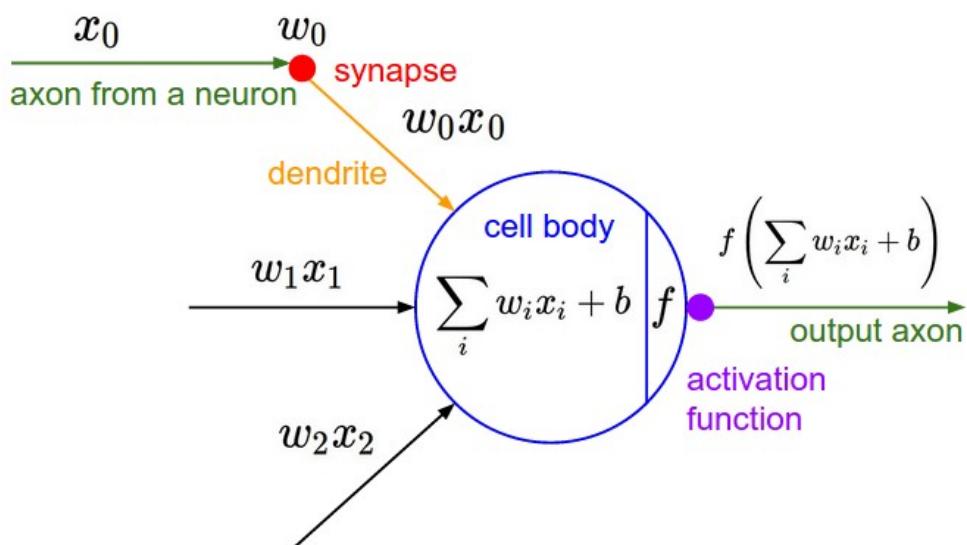
cây của dữ liệu. Tuy nhiên, với sự phát triển của công nghệ và nghiên cứu, Deep Learning đang không ngừng được cải tiến để mang lại những giải pháp thông minh và hiệu quả cho xã hội.

3.3.1 Tổng quan và phân loại mạng nơ-ron nhân tạo

Mạng nơ-ron nhân tạo là một mô hình toán học được xây dựng dựa trên cấu trúc của bộ não sinh học với thành phần cơ bản là nơ-ron thần kinh. Hình 3.7a mô tả cấu trúc của một nơ-ron thần kinh sinh học với đầy đủ các thành phần. Mỗi nơ-ron nhận kích thích tại đầu vào (synapse) thông qua các sợi nhánh (dendrite). Tín hiệu được xử lý trong thân tế bào (cell body) và thông tin được gửi đến các nơ-ron khác ở đầu ra tại các đầu cuối sợi trục (axon terminal). Hình 3.7b mô tả một mạng nơ-ron nhân tạo. Trong đó, x_0, x_1, x_2 đại diện cho các tín hiệu đến từ các nơ-ron khác thông qua sợi trục (axon). Các tín hiệu này sẽ kết hợp với w_0, w_1, w_2 tại đầu vào và tạo thành w_0x_0, w_1x_1, w_2x_2 . Các giá trị w_0, w_1, w_2 được gọi là trọng số và tất cả trọng số của mạng nơ-ron sẽ được tinh chỉnh để nhận được kết quả đầu ra phù hợp từng vấn đề cụ thể. Thông qua sợi nhánh các giá trị kết hợp được đưa vào xử lý bên trong thân tế bào. Các hàm xử lý tín hiệu được định nghĩa khác nhau tuỳ thuộc vào chức năng của mạng nơ-ron. Tín hiệu này được truyền đến các nơ-ron khác qua sợi trục. Lưu ý rằng tín hiệu sau khi xử lý tại thân tế bào thường phải đi qua hàm kích hoạt (activation function) nhằm tạo ra các khoảng giá trị đầu ra khác nhau tuỳ thuộc vào đặc trưng của từng mô hình Máy học.



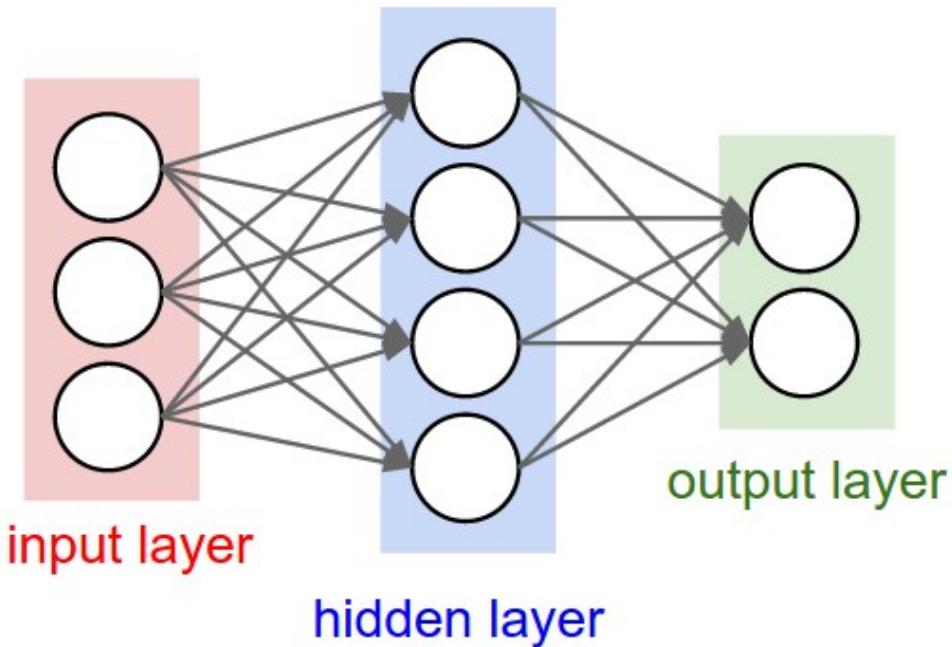
(a) Cấu trúc của một nơ-ron thần kinh sinh học



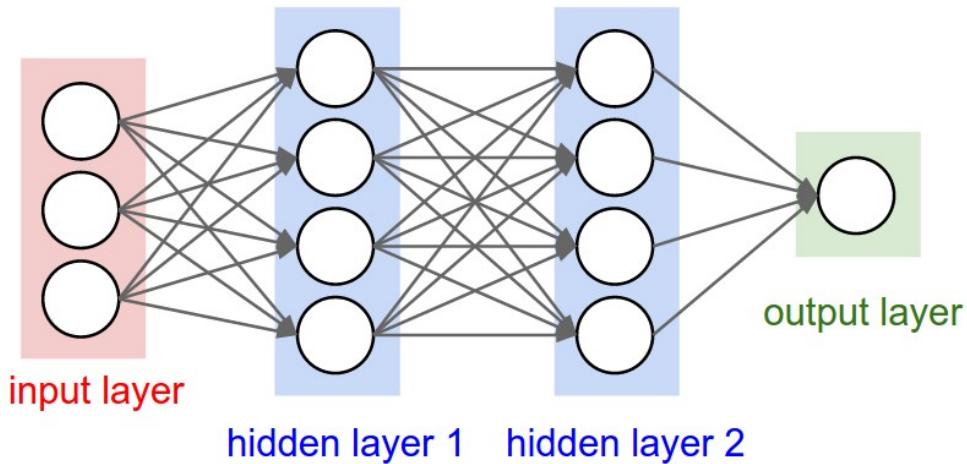
(b) Cấu trúc một nơ-ron thần kinh nhân tạo

Hình 3.7: Cấu trúc nơ-ron thần kinh (Nguồn: Stanford University)

Như mô tả trong hình 3.7 một nơ-ron điển hình được cấu tạo từ ba thành phần chính: Đầu vào, Thân tế bào giúp xử lý tín hiệu và Đầu ra. Nhiều nơ-ron kết hợp với nhau theo những quy luật được định nghĩa trước tạo thành một mạng lưới nhiều nơ-ron gọi là Mạng nơ-ron nhân tạo hay Mạng nơ-ron. Hình 3.12 mô tả một mạng nơ-ron đơn giản gồm 3 lớp: Lớp đầu vào, Lớp ẩn và Lớp đầu ra. Tại Lớp đầu vào, mỗi hình tròn được gọi là node đại diện cho một tín hiệu đưa vào mạng. Trong các mô hình Học sâu số lượng node có thể đến vài nghìn hoặc vài triệu. Tiếp theo, tín hiệu được xử lý trong Lớp ẩn. Số lượng lớp ẩn và số node trong lớp ẩn có thể khác nhau tùy vào độ phức tạp của mạng nơ-ron. Hình 3.8a, 3.8b lần lượt giới thiệu mạng nơ-ron với 1 lớp ẩn, 2 lớp ẩn.



(a) Mạng nơ-ron nhân tạo với 1 lớp ẩn



(b) Mạng nơ-ron nhân tạo với 2 lớp ẩn

Hình 3.8: Mạng nơ-ron nhân tạo đơn giản gồm Lớp đầu vào, Lớp ẩn và Lớp đầu ra (Nguồn: Stanford University)

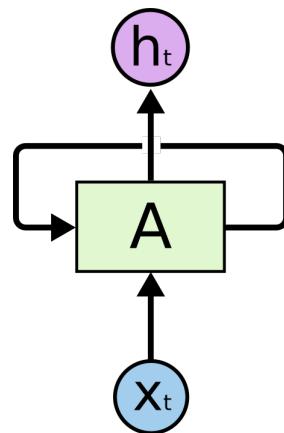
Phân loại mạng nơ-ron thần kinh

Tùy thuộc vào số lượng các Lớp (Layer), Số lượng Nút (Node), Sự tương tác giữa các lớp sẽ hình thành các mạng khác nhau. Mạng nơ-ron thường chia thành 2 dạng chính là: Mạng truyền thẳng (Feed-Forward Neural Network) và Mạng hồi quy (Recurrent Neural Network).

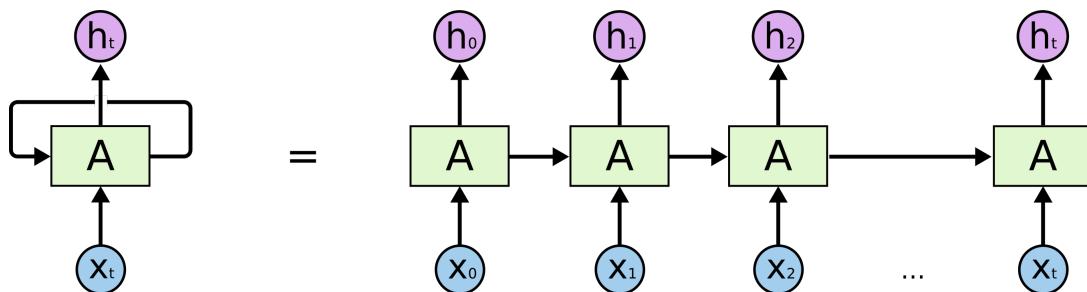
- Mạng truyền thẳng: là dạng đơn giản nhất trong mạng nơ-ron. Mạng xử lý dữ liệu theo một hướng từ đầu vào đến đầu ra như mô tả trong hình 3.12.

Các nút trong từng lớp được kết nối với tất cả các nút ở lớp tiếp theo. Tuy nhiên, tại các liên kết đầu ra sẽ không tương tác lại với các nút đầu vào trong cùng một lớp hay các lớp trước.

- **Mạng hồi quy:** sử dụng các vòng lặp để liên kết giá trị đầu ra và giá trị đầu vào của cùng một lớp hoặc nhiều lớp trước trong mạng (hình 3.9). Mạng hồi quy khác biệt với các mạng khác ở thuộc tính "bộ nhớ". "Bộ nhớ" sử dụng các thông tin ở các đầu vào trước đó để tác động vào đầu vào hoặc đầu ra hiện tại. Hình 3.10 mô tả một mạng hồi quy được trải ra thành nhiều nơ-ron với nhiều đầu vào và đầu ra. Mạng hồi quy được sử dụng nhiều trong các ứng dụng về dịch thuật, xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói.



Hình 3.9: Một đoạn của Mạng hồi quy (Nguồn: colah.github.io)

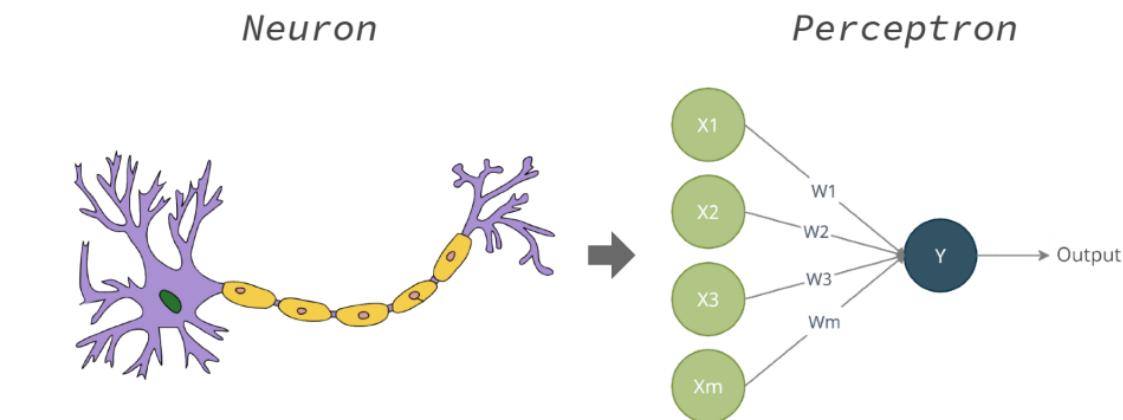


Hình 3.10: Mạng hồi quy được trải ra thành nhiều nơ-ron với nhiều đầu vào và đầu ra (Nguồn: colah.github.io)

3.3.2 Perceptron và Multi-Layer Perceptron (MLP)

Perceptron Learning Algorithm, gọi ngắn là Perceptron, là một thuật toán thực hiện công việc phân loại. Thuật toán này được Frank Rosenblatt đề xuất

dựa trên ý tưởng của Neural thần kinh, nó nhanh chóng tạo nên tiếng vang lớn trong lĩnh vực AI. Perceptron đọc từng dữ liệu và điều chỉnh biên giới sao cho tất cả các điểm nằm cùng một phía của biên giới có nhãn giống nhau.



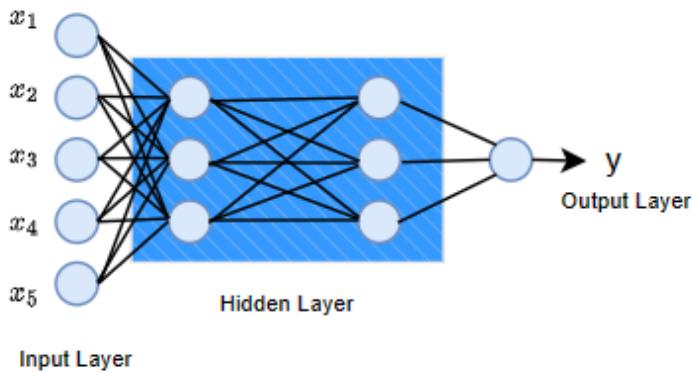
Neural Networks: Part 1

Hình 3.11: Neural Network with Perceptron

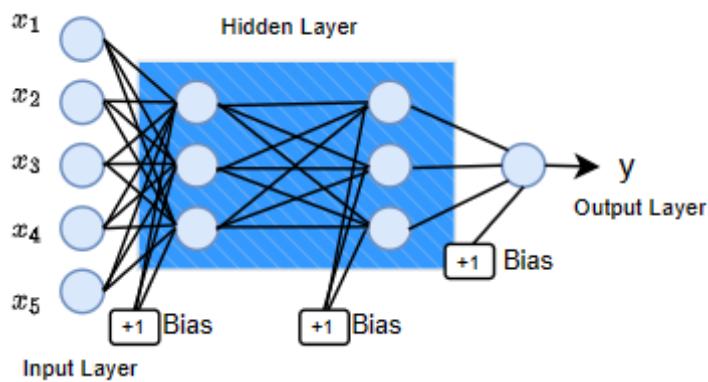
Multi-layer Perceptron là một loại mạng neural có nhiều lớp thực hiện các phép tính. Một mạng neural đơn giản nhất là perceptron, chỉ có hai lớp: lớp đầu vào và lớp đầu ra. Lớp đầu vào nhận các giá trị đặc trưng của dữ liệu, và lớp đầu ra tính toán nhãn lớp dựa trên các trọng số của các đặc trưng. Mạng neural đa lớp có thêm các lớp ẩn giữa lớp đầu vào và lớp đầu ra. Các lớp ẩn giúp mạng neural học được các đặc trưng phức tạp hơn từ dữ liệu.

Mạng neural đa lớp có kiến trúc lan truyền tiến, nghĩa là dữ liệu được truyền từ lớp này sang lớp khác theo một hướng duy nhất. Mỗi lớp được kết nối với lớp kế tiếp bằng các ma trận trọng số. Để xác định kiến trúc của mạng neural, chúng ta cần biết số lượng lớp, số lượng nút trong mỗi lớp, và hàm mất mát để tối ưu hóa. Thuật toán perceptron sử dụng hàm mất mát perceptron, nhưng chúng ta có thể chọn các hàm mất mát khác tùy theo bài toán. Hai hàm mất mát phổ biến là entropy chéo với đầu ra softmax cho bài toán phân loại nhiều lớp, và bình phương sai số với đầu ra tuyến tính cho bài toán hồi quy.

Ngoài ra, chúng ta cũng có thể thêm các nút bias vào các lớp ẩn và lớp đầu ra. Các nút bias là các nút luôn có giá trị bằng 1, và có trọng số là bias. Các nút bias giúp điều chỉnh độ lệch của đầu ra so với đầu vào. Các ví dụ về các mạng



(a) Neural Network không có Bias



(b) Neural Network có Bias

Hình 3.12: Mạng lan truyền tiến có hai lớp ẩn và một lớp đầu ra

neural đa lớp có hoặc không có các nút bias được hiển thị trong [Hình 3.12b] và [Hình 3.12a], tương ứng. Trong mỗi trường hợp, mạng neural có ba lớp. Lớp đầu vào không được tính là một lớp, vì nó chỉ truyền dữ liệu và không có phép tính nào. Nếu một lớp có p nút, thì đầu ra của lớp đó là một vector có kích thước p . Chúng ta có thể coi các nút trong cùng một lớp là một nút vector, và các ma trận trọng số là các kết nối giữa các nút vector.

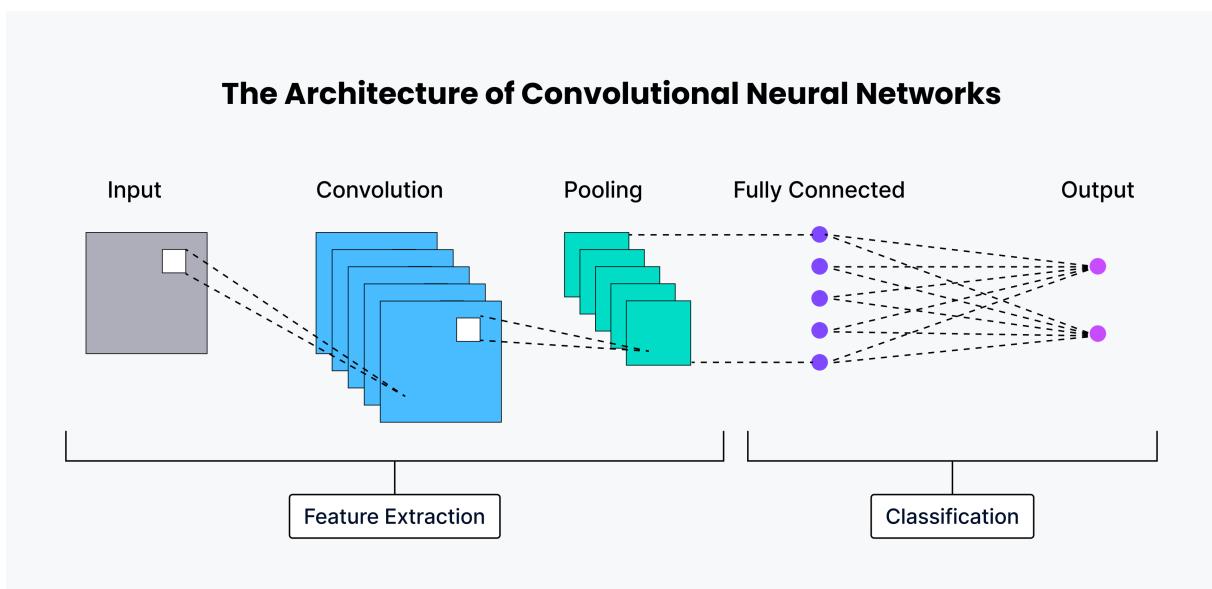
3.4 Convolutional Neural Network - CNN

Mạng Nơ-ron tích chập (CNN) là một công cụ quan trọng trong lĩnh vực học sâu (DL), được ứng dụng rộng rãi trong xử lý ảnh và xử lý ngôn ngữ tự nhiên. Khả năng của CNN đến từ cấu trúc đặc biệt của nó, đặc biệt là lớp tích chập.

Lớp tích chập trong CNN hoạt động như một bộ lọc, chúng khám phá các đặc trưng cục bộ trong ảnh và tạo ra các bản đồ đặc trưng. Các lớp tích chập sau đó được xếp chồng lên nhau, mỗi lớp có thể nhận diện các đặc trưng phức tạp hơn so với lớp trước đó. Điều này cho phép mạng Nơ-ron học được các điểm đặc trưng ngữ cảnh và biểu diễn hình ảnh ở mức độ trừu tượng cao.

Một điểm đặc biệt của CNN là khả năng học các đặc trưng từ dữ liệu mà không cần sự can thiệp của con người trong việc xác định các đặc trưng quan trọng. Điều này giúp CNN trở thành một công cụ mạnh mẽ trong việc tự động hóa quy trình phân tích hình ảnh và xử lý ngôn ngữ tự nhiên.

Kiến trúc chung của mạng Convolution Neural Network cơ bản gồm ba loại: lớp Convolution Layer, lớp Pooling Layer, cuối cùng là lớp Fully Connected Layer. Trong đó các lớp có ý nghĩa như sau:



Hình 3.13: Kiến trúc mạng Convolution Neural Network cơ bản (Nguồn : Zilliz.com)

Convolution layer: Các lớp tích chập hoạt động bằng cách di chuyển qua dữ liệu đầu vào một tập hợp các 'filter' hoặc 'kernel'. Mỗi filter được thiết kế để nhận biết một đặc điểm hoặc mẫu cụ thể, như các cạnh, góc, hoặc các hình dạng phức tạp hơn trong các deeper layer. Khi các filter di chuyển qua ảnh, chúng tạo ra một bản đồ biểu thị các vị trí mà các đặc điểm đó được phát hiện. Kết quả của lớp tích chập là một bản đồ đặc trưng, là một biểu diễn của ảnh đầu vào sau khi áp dụng các filter. Các lớp tích chập có thể được xếp chồng lên nhau để tạo ra các mô hình phức tạp hơn, có khả năng học được các đặc điểm phức tạp hơn

từ hình ảnh. Nói một cách đơn giản, các lớp tích chập đóng vai trò quan trọng trong việc trích xuất các đặc điểm từ các hình ảnh đầu vào, bao gồm các cạnh, góc, textures, và các mẫu phức tạp khác.

Pooling layer : Các lớp gộp tiếp theo sau các lớp tích chập và có nhiệm vụ giảm kích thước không gian của dữ liệu đầu vào, từ đó giúp đơn giản hóa quá trình xử lý và tiết kiệm bộ nhớ. Trong ngữ cảnh của hình ảnh, khi nói về "chiều không gian", chúng ta đề cập đến chiều rộng và chiều cao của hình ảnh, tức là số hàng và cột của các pixel. Bằng cách giảm chiều không gian, các lớp gộp giúp giảm số lượng tham số hoặc trọng số trong mạng, từ đó giảm nguy cơ overfitting và tăng tốc quá trình huấn luyện mô hình.

Có 3 kỹ thuật phổ biến được sử dụng trong lớp Pooling bao gồm: Max pooling, Average pooling, và Sum pooling

- Gộp tối đa (Max pooling) là phương pháp phổ biến trong việc giảm kích thước, nó chọn ra giá trị lớn nhất từ mỗi vùng cửa sổ gộp. Ví dụ, nếu chúng ta áp dụng gộp tối đa với cửa sổ kích thước 2×2 , chúng ta sẽ chọn pixel có giá trị cao nhất trong mỗi vùng 2×2 và bỏ qua các giá trị khác. Điều này giúp giữ lại các đặc điểm nổi bật và quan trọng nhất trong hình ảnh.
- Trong khi đó, gộp trung bình (Average pooling) tính toán giá trị trung bình của tất cả các pixel trong mỗi cửa sổ gộp. Phương pháp này tạo ra một biểu diễn mượt mà và trung bình của các đặc điểm trong cửa sổ.
- Sum pooling là một phương pháp gộp khác, trong đó giá trị của mỗi pixel trong mỗi cửa sổ gộp được tính tổng lại thay vì lấy giá trị lớn nhất hoặc giá trị trung bình. Quá trình này tạo ra một biểu diễn tổng quát của các đặc điểm trong cửa sổ, thay vì chỉ tập trung vào các đặc điểm nổi bật nhất hoặc giá trị trung bình. Tuy nhiên, sum pooling thường ít được sử dụng trong thực tế hơn so với max pooling và average pooling, vì nó có thể dẫn đến mất mát thông tin quan trọng và làm giảm hiệu suất của mô hình.

Fully Connected Layer: Các lớp kết nối đầy đủ trong mạng nơ-ron tích chập (CNN) là một phần cơ bản nhưng quan trọng, trong đó mỗi neuron được kết nối với mọi neuron trong lớp trước đó. Thông thường, chúng được sử dụng ở phần cuối của CNN, khi mục tiêu là sử dụng các đặc điểm đã học được từ các lớp tích chập và gộp tối đa để đưa ra dự đoán như phân loại đầu vào thành các nhãn. Ví dụ, nếu CNN được sử dụng để phân loại hình ảnh của các loài động vật, lớp kết nối đầy đủ cuối cùng có thể sử dụng các đặc điểm đã học được từ các lớp trước đó để phân loại hình ảnh là chúa chó, mèo, chim, v.v.

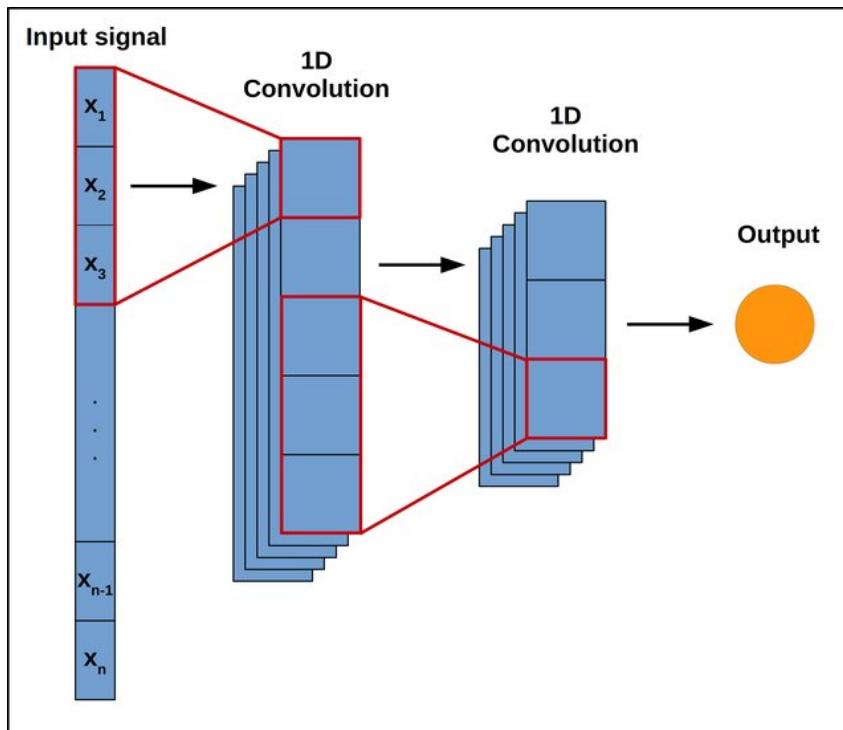
Các lớp kết nối đầy đủ chuyển đầu ra từ các lớp tích chập và gộp tối đa trước đó thành một vector một chiều, giúp mạng tích hợp tất cả các đặc điểm từ toàn bộ hình ảnh, không chỉ xem xét các đặc điểm cục bộ. Chúng chịu trách nhiệm ánh xạ các đặc điểm này đến đầu ra mong muốn, như nhãn lớp trong các nhiệm vụ phân loại. Bằng cách này, chúng là phần quyết định cuối cùng của mạng, xác định ý nghĩa của các đặc điểm trong bối cảnh cụ thể của vấn đề, chẳng hạn dạng một con mèo hoặc một con chó.

Bằng cách kết hợp các lớp tích chập và gộp tối đa, một cấu trúc phân cấp các đặc điểm được tạo ra, trong đó các lớp đầu tiên phát hiện các mẫu đơn giản và các lớp sau xây dựng trên chúng để phát hiện các mẫu phức tạp hơn.

3.4.1 One Dimension Convolution Neural Network (CNN 1D)

Mạng nơ-ron tích chập một chiều (CNN 1D) có cấu trúc tương tự CNN 2D, nhưng được thiết kế để xử lý trực tiếp dữ liệu thô một chiều, chẳng hạn như tín hiệu ECG, mà không cần trích xuất đặc trưng thủ công. Dữ liệu đầu vào được đưa qua các lớp tích chập, mỗi lớp học và tạo ra các biểu diễn phân cấp, giúp mạng hiểu được cấu trúc ẩn của tín hiệu.

Hình 3.14 minh họa cấu trúc của một mạng CNN 1D, bao gồm các lớp tích chập, lớp gộp và lớp kết nối đầy đủ, thể hiện quá trình trích xuất đặc trưng và dự đoán.



Hình 3.14: Cấu trúc của mạng CNN 1D (Nguồn: [1])

Đầu vào của CNN 1D thường là một mảng tín hiệu X , và mạng được huấn luyện để học tập tham số Θ nhằm ánh xạ X sang đầu ra dự đoán T theo công thức:

$$T = F(X | \Theta) = f_L(\dots f_2(f_1(X | \Theta_1) | \Theta_2) \dots | \Theta_L),$$

trong đó L là số lớp ẩn trong mạng.

Ở mỗi lớp tích chập thứ l , phép toán được biểu diễn như sau:

$$T_l = f_l(X_l | \Theta_l) = h(W \otimes X_l + b), \quad \Theta_l = [W, b],$$

với các thành phần:

- \otimes : Phép tích chập, trích xuất đặc trưng từ dữ liệu đầu vào.
- X_l : Ma trận đầu vào hai chiều, chứa N bản đồ đặc trưng, kích thước (N, d) .
- W : Tập hợp N bộ lọc một chiều, kích thước (N, m) , dùng để tạo đặc trưng mới.
- b : Vector độ lệch, điều chỉnh đầu ra của phép tích chập.
- h : Hàm kích hoạt, giới hạn giá trị đầu ra.
- T_l : Đầu ra của lớp tích chập, kích thước $(N, d - m + 1)$.

Các lớp gộp (pooling) được áp dụng giữa các lớp tích chập để giảm kích thước bản đồ đặc trưng và tăng trưởng thụ cảm cho các lớp tiếp theo. Đầu ra của lớp tích chập cuối được làm phẳng và đưa vào các lớp kết nối đầy đủ (fully connected), với phép toán:

$$T_l = f_l(X_l | \Theta_l) = h(WX_l + b), \quad \Theta_l = [W, b].$$

Trong phân loại đa lớp, lớp đầu ra có số nơ-ron bằng số lớp của mô hình, sử dụng hàm kích hoạt softmax để biểu thị xác suất thuộc về mỗi lớp cho mẫu đầu vào.

3.4.2 Convolution layer

Lớp tích chập thực hiện phép tích chập giữa dữ liệu đầu vào là một bộ lọc(kernel), tạo ra một bản đồ đặc trưng (feature map) làm đầu ra. Quá trình này giúp nhận diện các đặc trưng cụ thể trong dữ liệu đầu vào như cạnh, góc hoặc các điểm phức tạp hơn. Ví dụ, khi áp dụng lớp tích chập cho một hình ảnh màu, đầu vào sẽ là một ma trận pixel có ba chiều - chiều cao, chiều rộng và chiều sâu, tương ứng với kênh màu RGB.

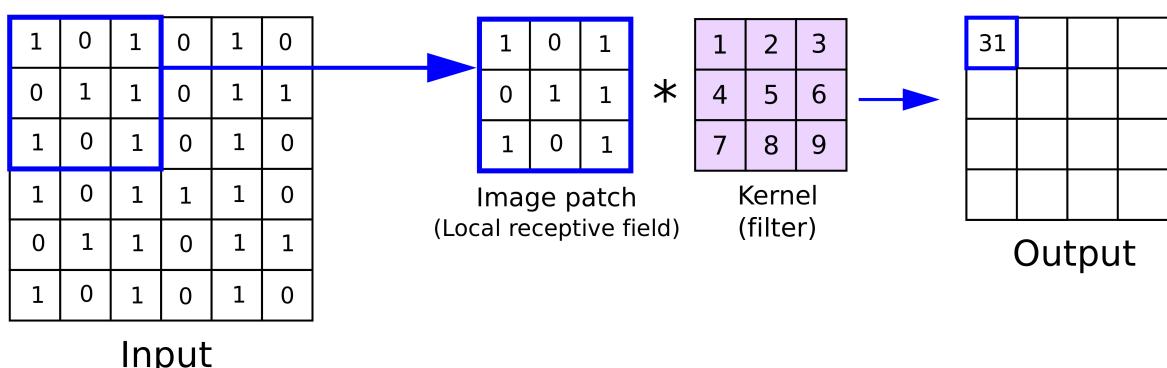
Bộ lọc, còn được gọi là kernel hoặc filter, là một ma trận hai chiều của trọng số. Kích thước của bộ lọc thường là một ma trận 3×3 , và nó di chuyển qua hình ảnh để kiểm tra sự hiện diện của các đặc trưng. Quá trình này được gọi là phép tích chập, trong đó tính toán sản phẩm chấm giữa các điểm ảnh đầu vào và bộ lọc. Kết quả là một bản đồ đặc trưng, được biết đến là feature map, activation map hoặc convolved feature.

Trọng số trong bộ lọc không thay đổi khi nó di chuyển qua hình ảnh, mà được chia sẻ trên toàn bộ đầu vào. Một số tham số khác, như các giá trị trọng số, điều chỉnh trong quá trình huấn luyện bằng cách sử dụng backpropagation và gradient descent.

Trước khi huấn luyện mạng nơ-ron, ba siêu tham số cần được xác định để ảnh hưởng đến kích thước của đầu ra:

- Số lượng bộ lọc ảnh hưởng đến chiều sâu của đầu ra. Ví dụ, nếu có ba bộ lọc, sẽ tạo ra ba feature maps, tạo ra một chiều sâu là ba.
- Stride là khoảng cách mà bộ lọc di chuyển qua đầu vào. Giá trị stride càng lớn, đầu ra càng nhỏ.
- Padding được sử dụng khi kích thước của bộ lọc không phù hợp với kích thước của hình ảnh đầu vào. Có ba loại padding: "valid", "same", và "full", ảnh hưởng đến kích thước của đầu ra.

Sau mỗi phép tích chập, một hàm phi tuyến ReLU thường được áp dụng lên feature map để tạo ra một biểu diễn phi tuyến tính.



Hình 3.15: Convolution Layer (Nguồn: www.baeldung.com)

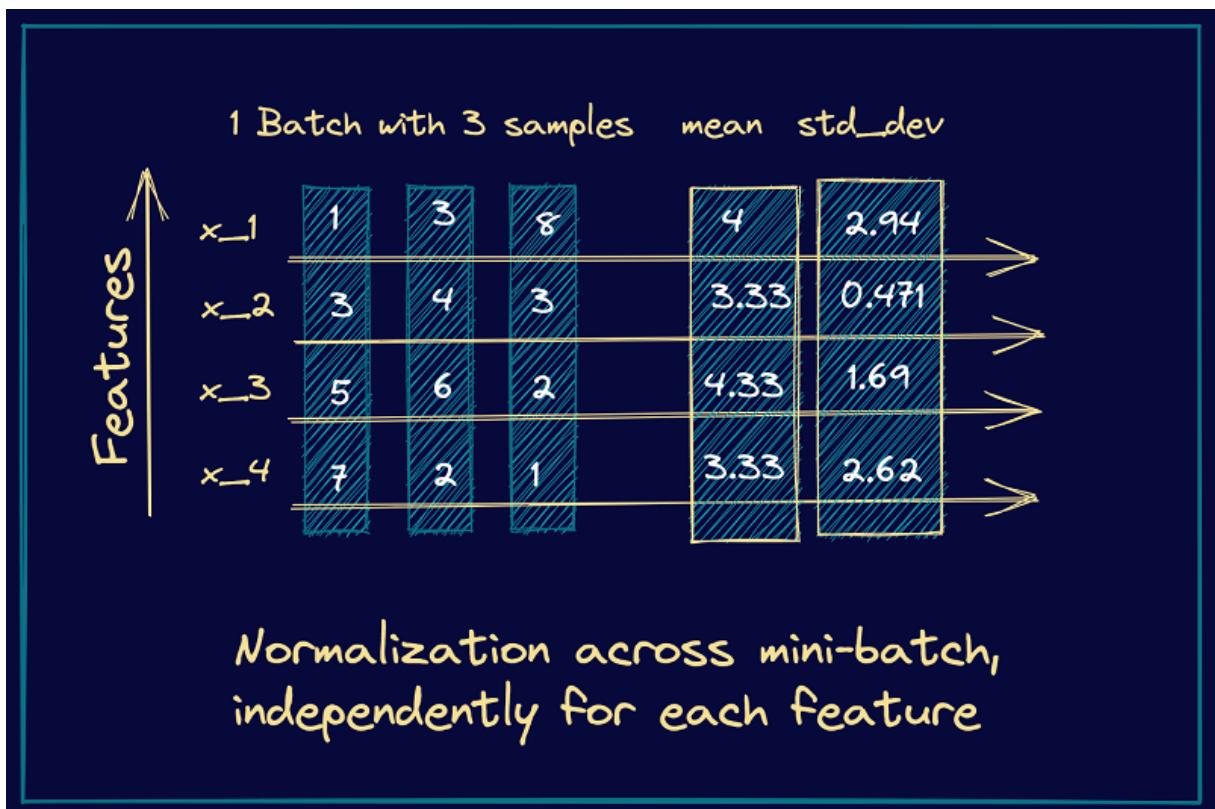
3.4.3 Batch normalization layer

Lớp Batch Normalization hoạt động bằng cách tính toán giá trị trung bình và phương sai của mỗi kênh trên toàn bộ mini-batch. Sau đó, đầu vào của mỗi kênh được chuẩn hóa theo công thức:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

Ở đây:

- x_i là giá trị đầu vào.
- μ_B là giá trị trung bình của mini-batch.
- σ_B^2 là phương sai của mini-batch.
- ϵ là một hằng số nhỏ được thêm vào để tránh chia cho 0.



Hình 3.16: Batch Normalization layer (Nguồn: www.pinecone.io)

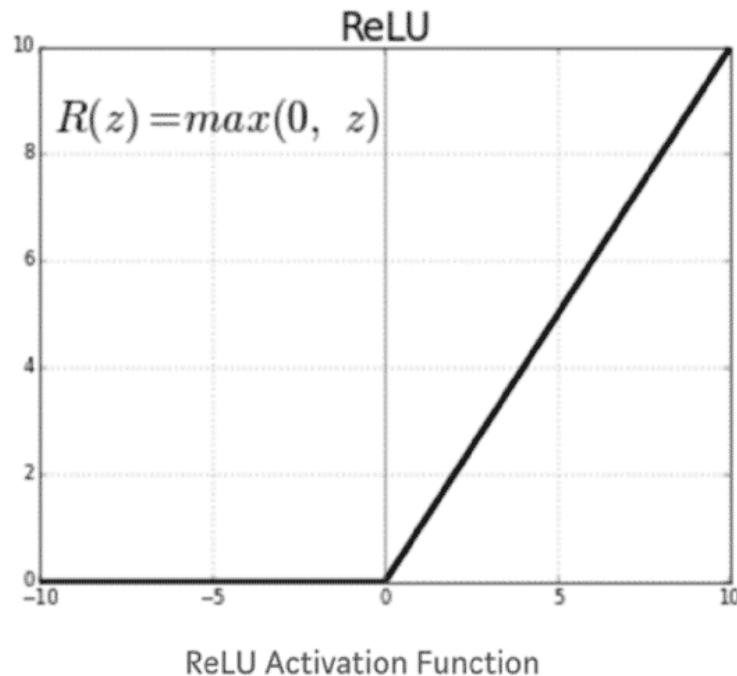
Sau khi chuẩn hóa, lớp Batch Normalization nhân đầu vào với một hệ số tỷ lệ γ có thể học được và dịch chuyển nó bằng một độ lệch β có thể học được. Quá trình này giúp giảm thiểu sự thay đổi của phân bố đầu vào của các lớp trong mạng neural, từ đó giúp tăng tốc quá trình huấn luyện, giảm sự phụ thuộc vào việc khởi tạo, và ngăn chặn hiện tượng overfitting.

3.4.4 ReLU Layer

Lớp ReLU (Rectified Linear Unit) áp dụng hàm kích hoạt để đưa tính phi tuyến vào mô hình, giúp mạng nơ-ron học các mối quan hệ phức tạp trong dữ liệu. Hàm ReLU hoạt động bằng cách đặt ngưỡng: các giá trị nhỏ hơn hoặc bằng 0 được đặt thành 0, còn các giá trị lớn hơn 0 giữ nguyên, theo công thức:

$$f(x) = \begin{cases} x, & \text{nếu } x \geq 0 \\ 0, & \text{nếu } x < 0 \end{cases}$$

Hình 3.17 minh họa biểu đồ của hàm ReLU, cho thấy đầu ra tuyến tính với các giá trị dương và bằng 0 với các giá trị âm.



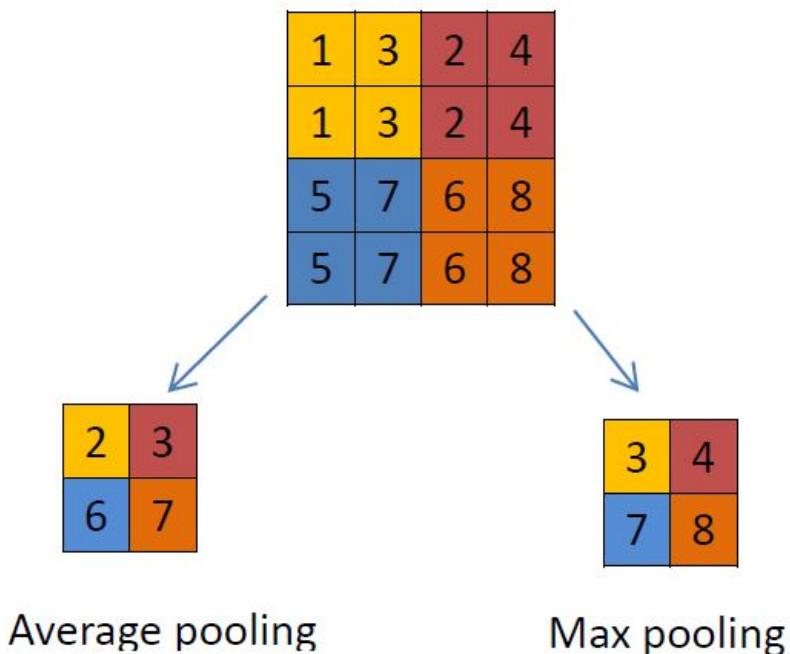
Hình 3.17: Biểu đồ hàm kích hoạt ReLU (Nguồn: www.mygreatlearning.com)

Hàm ReLU mang lại nhiều lợi ích trong học sâu. Thứ nhất, nó tạo ra tính thưa (sparsity) bằng cách đặt các giá trị âm thành 0, giúp giảm số lượng nơ-ron hoạt động và tăng hiệu quả tính toán. Thứ hai, ReLU giảm thiểu vấn đề biến mất đạo hàm (vanishing gradient) thường gặp ở các hàm kích hoạt như sigmoid hoặc tanh, do đạo hàm của ReLU là 1 với $x > 0$ và 0 với $x < 0$, giúp gradient lan truyền tốt hơn qua các lớp. Trong bài toán tái lập tín hiệu ECG, lớp ReLU hỗ trợ mô hình CNN 1D-BiLSTM học các đặc trưng phi tuyến từ tín hiệu, góp phần cải thiện độ chính xác của dự đoán.

3.4.5 Pooling layer

Lớp Pooling trong mạng nơ-ron tích chập (CNN) có mục đích là giảm kích thước của biểu diễn bằng cách giảm độ phân giải của các feature maps biểu thị đặc trưng cho lớp trước đó. Bằng cách này, lớp Pooling giúp giảm số lượng tham số và độ phức tạp tính toán của mô hình.

Lớp Pooling hoạt động bằng cách sử dụng các đơn vị kết hợp với đầu vào từ một vùng nhỏ có kích thước $n \times n$. Cửa sổ gộp này có thể có kích thước tùy ý và có thể chồng lấn lên nhau. Có hai kỹ thuật phổ biến của lớp Pooling là Average Pooling và Max Pooling.



Hình 3.18: Pooling layer (Nguồn :wprany.github.io)

- **Average Pooling (Gộp trung bình):** Công thức tính điểm đầu ra của mỗi phần tử trong feature map được tính bằng cách lấy trung bình trên các giá trị đầu vào trong vùng gộp, nhân nó với một hệ số β có thể học được, cộng thêm một độ lệch b có thể học được, và truyền kết quả qua một hàm phi tuyến. Điều này giúp tạo ra một bản đồ đặc trưng với độ phân giải thấp hơn.

$$a_j = \tanh \left(\beta \sum_{n \times n} a_{n \times n}^i + b \right)$$

- **Max Pooling (Gộp tối đa):** Công thức tính điểm đầu ra của mỗi phần tử trong feature map được thực hiện bằng cách áp dụng một hàm cửa sổ $u(x, y)$ lên vùng đầu vào, và tính giá trị lớn nhất trong vùng lân cận. Kết quả là một bản đồ đặc trưng với độ phân giải thấp hơn, nhưng thường giữ lại các đặc trưng quan trọng nhất.

$$a_j = \max_{n \times n} \left(a_{n \times n}^j u(n, n) \right)$$

Cả hai kỹ thuật này giúp giảm kích thước của feature map và tạo ra một biểu diễn càng nghĩa là các đặc trưng quan trọng được tập trung vào, đồng thời giảm thiểu độ phức tạp của mô hình.

3.4.6 Fully Connected Layer

Lớp Fully Connected (FC), hay còn gọi là Dense Layer, là một thành phần quan trọng trong mạng nơ-ron, thường được đặt ở giai đoạn cuối của các mô hình như CNN 1D hoặc Multilayer Perceptron (MLP). Đặc điểm nổi bật của lớp này là mỗi nơ-ron trong lớp FC kết nối với tất cả các nơ-ron của lớp trước đó, tạo thành một mạng liên kết chặt chẽ để học các mối quan hệ phức tạp giữa các đặc trưng.

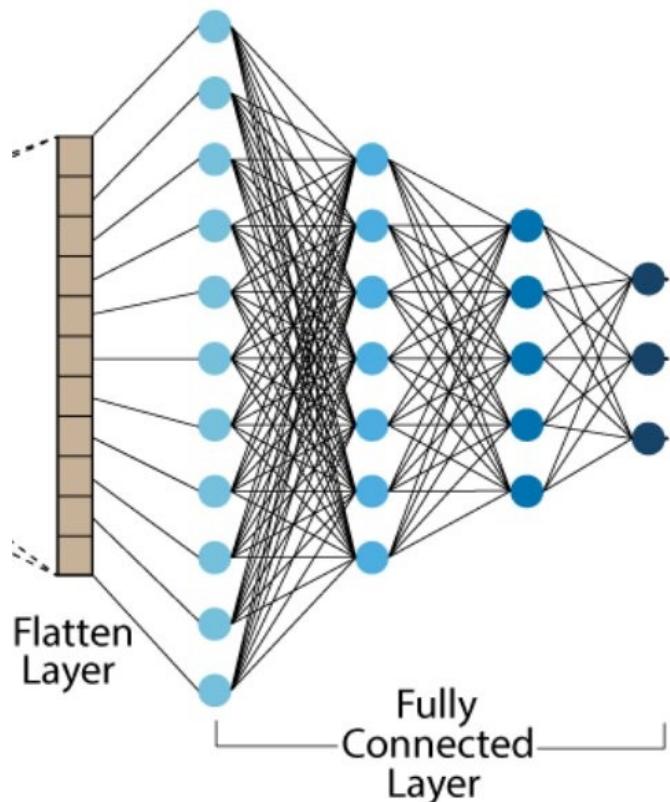
Hình 3.19 minh họa cấu trúc của một lớp Fully Connected, trong đó các nơ-ron từ lớp trước (thường là đặc trưng đã được làm phẳng) được kết nối đầy đủ với các nơ-ron trong lớp FC, dẫn đến đầu ra dự đoán.

Mỗi nơ-ron trong lớp FC được gán một bộ trọng số riêng, kết hợp với một độ lệch (bias) và hàm kích hoạt (thường là ReLU hoặc softmax). Trong quá trình huấn luyện, các trọng số này được điều chỉnh để tối ưu hóa hiệu suất mô hình. Cụ thể, đầu vào của lớp FC là một vector đặc trưng đã được làm phẳng từ các lớp trước (ví dụ: sau lớp tích chập và gộp trong CNN 1D), và đầu ra được tính theo công thức:

$$y = h(Wx + b),$$

trong đó x là vector đầu vào, W là ma trận trọng số, b là vector độ lệch, và h là hàm kích hoạt.

Trong bài toán tái lập tín hiệu ECG, lớp FC đóng vai trò ánh xạ các đặc trưng phức tạp đã học được từ các lớp tích chập và gộp thành đầu ra dự đoán, chẳng hạn như tín hiệu tái tạo hoặc phân loại bất thường. Nhờ khả năng học các mối quan hệ phi tuyến, lớp FC giúp mô hình đưa ra dự đoán chính xác hơn dựa trên thông tin tổng hợp từ toàn bộ mạng.



Hình 3.19: Cấu trúc của lớp Fully Connected

3.4.7 Residual Neural Network

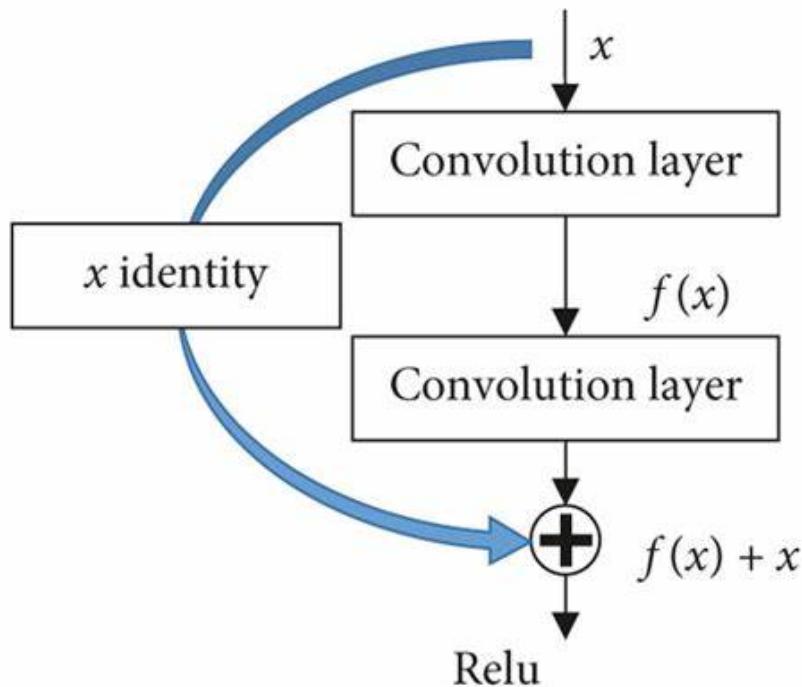
Residual Networks (ResNets) là một trong những kiến trúc mạng nơ-ron sâu rất thành công, đặc biệt trong các nhiệm vụ nhận dạng hình ảnh. ResNet được giới thiệu trong bài báo "Deep Residual Learning for Image Recognition" [21] và đã đạt được thành công vượt trội trong các cuộc thi nhận dạng ảnh như ImageNet.

Một trong những vấn đề lớn trong việc huấn luyện mạng nơ-ron sâu là vanishing gradient, tức là gradient trở nên rất nhỏ và không đủ mạnh để cập nhật các tham số của các lớp sâu. ResNet giải quyết vấn đề này bằng cách sử dụng Residual Blocks với skip connections để cho phép gradient lan truyền tốt hơn qua các lớp.

Residual Block

Residual Block là một thành phần trong kiến trúc **Residual Networks**, được thiết kế để giải quyết vấn đề **vanishing gradient** và giúp huấn luyện mạng nơ-ron sâu hiệu quả hơn. Mỗi residual block bao gồm một **kết nối nhảy (skip connection)**, cho phép thông tin đầu vào được cộng trực tiếp vào đầu ra của block, thay vì chỉ thông qua các lớp convolution. Điều này giúp duy trì dòng

chảy thông tin, cho phép gradient dễ dàng lan truyền qua các lớp trong quá trình huấn luyện.



Hình 3.20: Sơ đồ của Residual Block (Nguồn :www.researchgate.net)

Cấu trúc của một Residual Block:

- **Convolution Layer:** Thực hiện các phép lọc trên dữ liệu đầu vào.
- **Batch Normalization:** Chuẩn hóa các đầu vào để tăng tốc quá trình huấn luyện.
- **ReLU Activation:** Áp dụng hàm kích hoạt phi tuyến.
- **Convolution Layer 2:** Tiếp tục trích xuất đặc trưng từ đầu vào.
- **Batch Normalization:** Tiếp tục chuẩn hóa đầu ra.
- **Skip Connection:** Thêm đầu vào x vào đầu ra của các lớp convolution, tạo thành $F(x) + x$.

Công thức:

$$y = F(x) + x$$

Trong đó:

- $F(x)$ là kết quả tính toán từ các lớp convolution.
- x là đầu vào của block.
- y là đầu ra của block sau khi cộng với x .

Skip Connections

- Skip connection hay còn gọi là residual connection là một khái niệm quan trọng của ResNet. Thay vì chỉ truyền thông tin qua các lớp liên tiếp trong mạng, skip connection cho phép đầu vào của một lớp đi thẳng qua một số lớp và cộng trực tiếp vào đầu ra của lớp đó.
- Skip connection có tác dụng giúp gradient không bị suy giảm quá nhanh trong quá trình lan truyền ngược (backpropagation). Điều này giúp huấn luyện mạng nơ-ron sâu dễ dàng hơn và cho phép mạng học được những đặc trưng phức tạp mà không bị mất thông tin trong quá trình học.

Residual Learning

Thay vì học một ánh xạ phức tạp từ đầu vào x đến đầu ra $H(x)$, ResNet học một hàm dư $F(x)$, và kết quả là cộng $F(x)$ với đầu vào x .

Residual Learning có thể giúp mạng học hiệu quả hơn khi độ sâu của mạng tăng lên. Điều này có nghĩa là thay vì học trực tiếp ánh xạ $H(x)$, mạng học sự khác biệt giữa $H(x)$ và x , tức là $F(x)$.

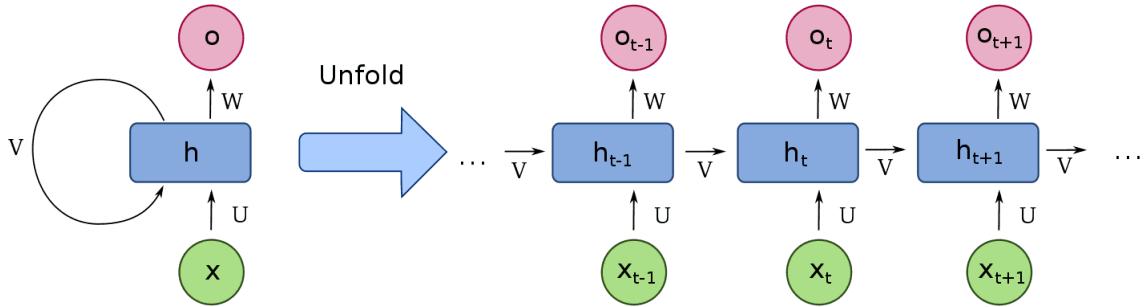
Phương trình tổng quát là:

$$H(x) = F(x) + x$$

Điều này cho phép mạng học được sự thay đổi giữa đầu vào và đầu ra, từ đó dễ dàng học các đặc trưng cần thiết mà không làm mất đi thông tin.

3.5 Recurrent Neural Network - RNN

Recurrent neural network (RNN) là một loại mạng thần kinh có khả năng xử lý dữ liệu tuần tự theo thời gian [Hình 3.10], thích hợp cho việc mô hình hóa các loại dữ liệu như chuỗi thời gian, video và âm thanh. Mạng này có khả năng ghi nhớ thông tin từ thời điểm trước đó và áp dụng cho dữ liệu đầu vào hiện tại. Bên trong cấu trúc mạng RNN [Hình 3.21] bao gồm các nút hoặc lớp kết nối tuần hoàn, mỗi nút có hai vector x_t và trạng thái ẩn trước đó là h_{t-1} làm đầu vào, đầu ra là o_t với một trạng thái ẩn mới là h_t .



Hình 3.21: Cấu trúc tế bào của mạng RNN (Nguồn: codingninjas.com)

Trong đó:

- V : là vector trọng số được sử dụng để tính toán cổng quên (Forget Gate). Cổng quên sẽ quyết định thông tin từ trạng thái ẩn trước đó là h_{t-1} sẽ được giữ lại trong trạng thái h_t .
- U : là vector trọng số được sử dụng để tính toán cổng đầu vào (Input Gate). Cổng đầu vào quyết định lượng thông tin từ đầu vào x_t sẽ được thêm vào trạng thái ẩn mới là h_t .

3.5.1 Gated Recurrent Unit (GRU)

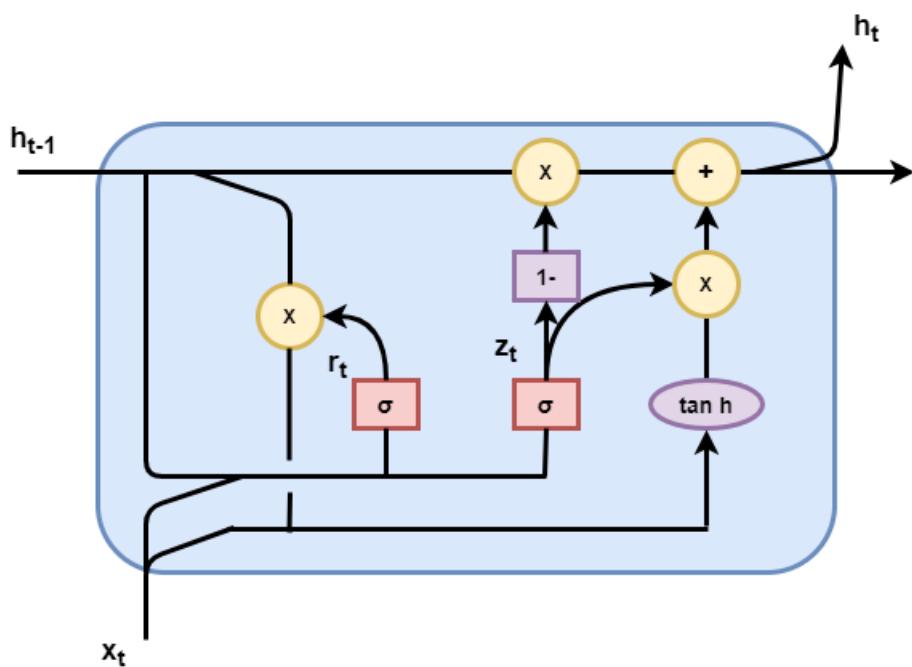
Gated Recurrent Unit (GRU) là một loại mạng nơ-ron hồi quy (RNN) được thiết kế để xử lý các chuỗi dữ liệu tuần tự. GRU được giới thiệu vào năm 2014 bởi Kyunghyun Cho, Bahdanau, Chorowski và Bengio.

GRU được thiết kế để khắc phục một số hạn chế của các mô hình RNN truyền thống, chẳng hạn như vấn đề mất độ dốc và khả năng học các phụ thuộc dài hạn. GRU sử dụng một cấu trúc "cổng" đặc biệt để điều khiển luồng thông tin qua mạng, cho phép nó học các phụ thuộc dài hạn hiệu quả hơn.

Gated Recurrent Unit (GRU) rất giống với Long Short - Term Memory (LSTM). GRU sử dụng các cổng (Gate) để kiểm soát luồng thông tin. Cấu trúc của một phần tử GRU bao gồm hai cổng:

- Update Gate: Quyết định lượng thông tin từ trạng thái ẩn trước đó sẽ được giữ lại.
- Reset Gate: Quyết định lượng thông tin từ trạng thái ẩn trước đó sẽ bị loại bỏ.

GRU cũng có một trạng thái ẩn, lưu trữ thông tin về các phần tử đầu vào trước đó. Trạng thái ẩn được cập nhật tại mỗi bước thời gian bằng cách kết hợp



Hình 3.22: Kiến trúc GRU(Nguồn: <https://appliedsingularity.com>)

thông tin từ trạng thái ẩn trước đó và đầu vào hiện tại, được điều khiển bởi các cổng cập nhật (Update Gate) và đặt lại (Reset Gate).

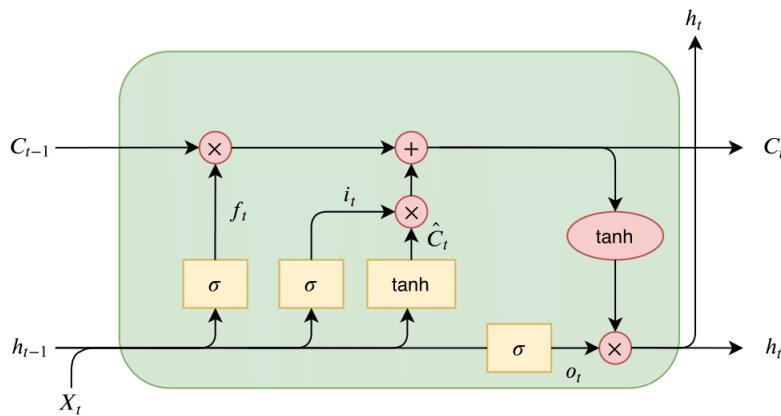
Tuy nhiên, khi áp dụng GRU vào tác vụ tái lập tín hiệu điện tâm đồ (ECG), một số hạn chế của GRU:

- **Khả năng học các phụ thuộc rất dài hạn bị hạn chế:** Mặc dù GRU có khả năng học các phụ thuộc dài hạn tốt hơn các mô hình RNN truyền thống, nhưng khả năng này vẫn bị hạn chế. Đối với các tín hiệu phi tuyến tính như ECG, các phụ thuộc dài hạn giữa các nhịp tim, vượt quá khả năng học của GRU.
- **Độ nhạy với siêu tham số:** Hiệu suất của GRU phụ thuộc rất nhiều vào các siêu tham số, như tỷ lệ học và kích thước ẩn. Việc điều chỉnh các siêu tham số này tốn nhiều thời gian.

3.5.2 Long Short-Term Memory (LSTM)

Mô hình Long Short-Term Memory (LSTM) là một loại mạng nơ-ron hồi quy (RNN) đặc biệt được thiết kế để xử lý vấn đề biến đổi dài hạn, phù hợp với nhiều ứng dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên, dự báo chuỗi thời gian, và cả trong lĩnh vực y học. Trong nghiên cứu của chúng tôi, chúng tôi tiếp cận vấn đề tái lập tín hiệu ECG bằng cách sử dụng mô hình LSTM để hiểu và tái tạo các đặc trưng phức tạp của tín hiệu này.

Để giải quyết các bối cảnh phụ thuộc lâu dài trong dữ liệu học tập. RNN được thay thế bởi các nút thuộc LSTM, nút này chứa các khối bộ nhớ với các ô nhớ được gọi là “cổng” trong lớp ẩn lặp lại trong [Hình 3.27]. Các cổng trên các ô nhớ kiểm soát các trạng thái thông tin mới, cập nhật và quên đi những trạng thái ẩn trước đó, đồng thời xác định đầu ra. Các thành phần chính của mô hình LSTM như sau:

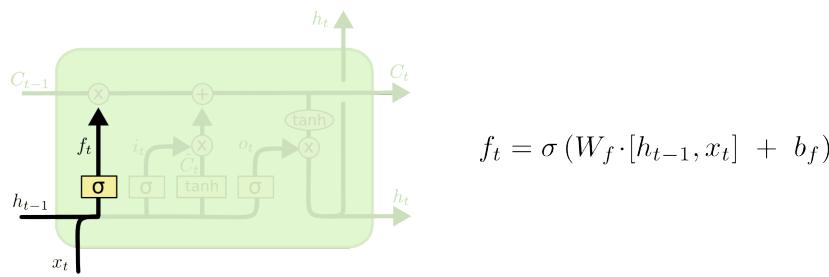


Hình 3.23: Kiến trúc Node của LSTM

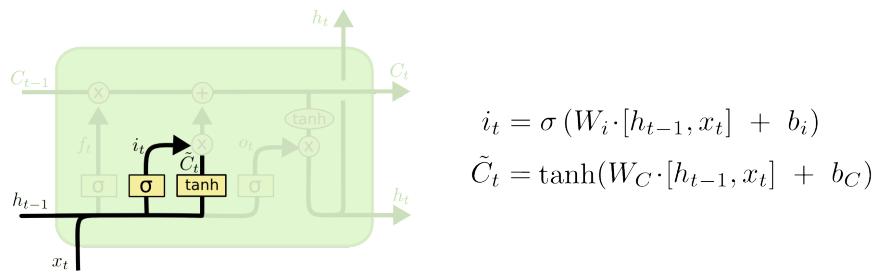
- Output: c_t, h_t . Cell State (c) là bảo toàn và truyền tải thông tin dài hạn qua các bước thời gian. Hidden State (h) là đầu ra của mô hình tại mỗi bước thời gian và cũng được sử dụng làm đầu vào cho bước tiếp theo.
- Input: c_{t-1}, h_{t-1} . Trong đó x_t là input ở state thứ t của model. c_{t-1}, h_{t-1} là output của layer trước đó và c là điểm mới của LSTM.

Trong kiến trúc mạng LSTM còn có 3 cổng đó là: Forget Gate, Input Gate, Output Gate.

- Forget Gate: Quyết định thông tin nào nên được giữ lại và thông tin nào bị quên từ trạng thái tế bào trước đó.
- Input Gate: Quyết định thông tin mới nào được thêm vào trạng thái tế bào. Ở đây có 2 phần, đầu tiên là lớp *sigmoid* quyết định những giá trị nào sẽ được cập nhật. Tiếp theo là lớp *tanh*, sẽ tạo ra một vector chứa các giá trị mới, sau đó ta nhân 2 lớp này, tạo ra một ma trận chứa tham số cập nhật.
- Output Gate: Quyết định trạng thái tế bào mới nào sẽ làm ảnh hưởng đến đầu ra của mô hình. Đầu tiên, lớp *sigmoid* quyết định những giá trị sẽ xuất

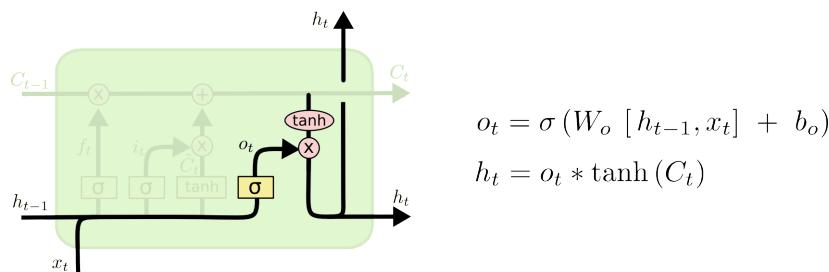


Hình 3.24: Forget Gate (Nguồn: colah.github.io)



Hình 3.25: Input Gate (Nguồn: colah.github.io)

ra. Sau đó, các giá trị sẽ qua $tanh$ (đẩy các giá trị vào khoảng từ -1 và 1) và nhân với đầu ra của cổng $sigmoid$, để xuất ra các phần quyết định.



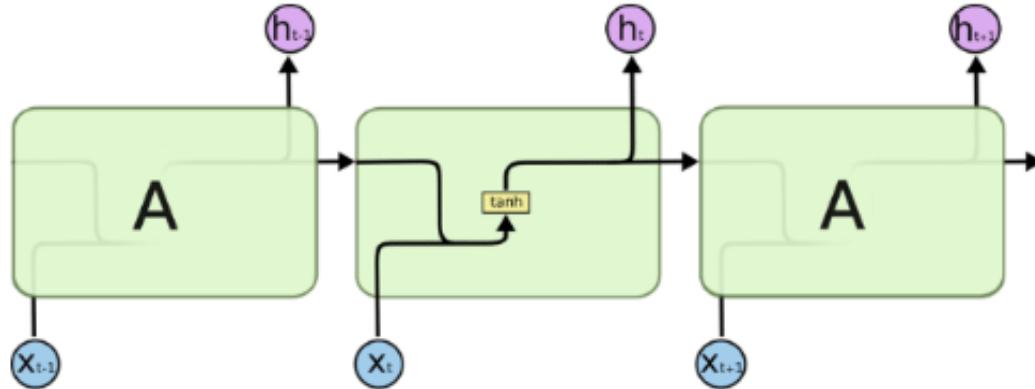
Hình 3.26: Output Gate (Nguồn: colah.github.io)

- Trong đó, các kí hiệu σ , W là ma trận trọng số và b là các vector sai lệch. Khi LSTM – RNN huấn luyện một tập dữ liệu để học, nó tập trung vào việc học các tham số b , σ và W của các cổng.

Kiến trúc mạng Long Short - Term Memory

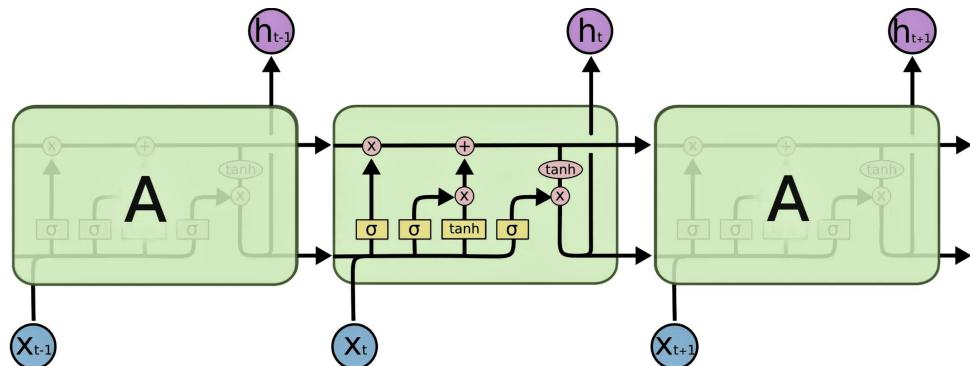
Tất cả mạng lưới thần kinh hồi quy đều có dạng là một chuỗi các mô-đun mạng lưới thần kinh lặp lại. Trong các Recurrent Neural Network (RNNs) tiêu

chuẩn, mô-đun lặp lại sẽ có cấu trúc rất đơn giản. Ví dụ, có một lớp tanh duy nhất (Hình 3.27).



Hình 3.27: Mạng RNN tiêu chuẩn (Nguồn: colah.github.io)

Mạng Long Short-Term Memory cũng có cấu trúc giống như chuỗi trên (Hình 3.28), nhưng mô-đun lặp lại của LSTM có cấu trúc khác. Thay vì có duy nhất một lớp mạng lưới thần kinh duy nhất, thì mạng LSTM có bốn lớp tương tác theo một cách đặc biệt.



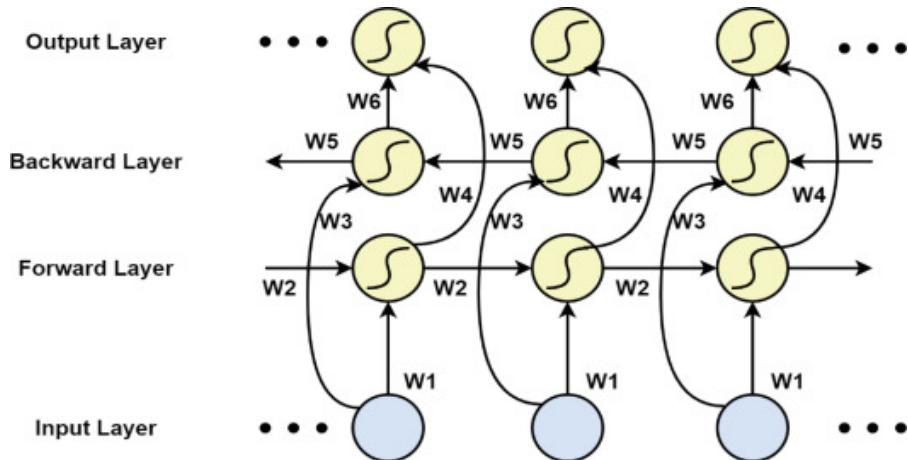
Hình 3.28: Kiến trúc mạng LSTM bốn lớp (Nguồn: colah.github.io)

Trong sơ đồ trên (Hình 3.28), có các hộp màu vàng còn được gọi là các lớp mạng thần kinh đã học dữ liệu. Ở kiến trúc này có bốn lớp (layers) trong đó gồm, Forget Gate, Input Gate, Output Gate và một lớp được bổ sung là Cell State Candidate.

3.5.3 Bidirectional Long Short-Term Memory (Bi-LSTM)

Bidirectional Long Short-Term Memory (BiLSTM) là một sự mở rộng của mạng LSTM truyền thống. Khác với các LSTM thông thường chỉ xử lý chuỗi

dữ liệu theo một hướng duy nhất, BiLSTM cho phép thông tin lưu chuyển theo cả hai hướng, từ trước đến sau và ngược lại, giúp mạng có thể nắm bắt thông tin đầy đủ hơn. Điều này giúp BiLSTM đặc biệt hiệu quả cho các tác vụ mà việc hiểu biết về cả ngữ cảnh quá khứ và tương lai là rất quan trọng.



Hình 3.29: Kiến trúc Bidirectional Long Short-Term Memory (Bi-LSTM)
(Nguồn: www.sciencedirect.com)

Bi-LSTM gồm hai lớp LSTM riêng biệt:

- **Forward LSTM:** Xử lý chuỗi từ đầu đến cuối (từ trái sang phải).
- **Backward LSTM:** Xử lý chuỗi từ cuối đến đầu (từ phải sang trái).

Đầu ra của cả hai LSTM này sẽ được kết hợp lại để tạo ra đầu ra cuối cùng. Đầu ra cuối cùng tại thời điểm t được tính bằng:

$$h_t = \vec{h}_t + \overleftarrow{h}_t$$

Trong đó:

- h_t : Vectơ xác suất cuối cùng của mạng tại thời điểm t .
- \vec{h}_t : Vectơ xác suất từ mạng LSTM tiến (Forward LSTM) tại thời điểm t .
- \overleftarrow{h}_t : Vectơ xác suất từ mạng LSTM lùi (Backward LSTM) tại thời điểm t .

Quy trình hoạt động của Bi-LSTM

Forward Long Short- Term Memory

LSTM truyền thống, hay Forward LSTM, sẽ xử lý chuỗi dữ liệu từ bước thời gian đầu tiên $t = 1$ đến bước cuối cùng $t = T$. Với mỗi bước thời gian t , Forward LSTM sẽ tính toán đầu ra $\vec{h}_t^{\text{forward}}$ dựa trên thông tin từ bước trước $\vec{h}_{t-1}^{\text{forward}}$ và đầu vào tại bước t .

Công thức tính toán trong LSTM (Forward):

$$\vec{h}_t = \text{LSTM}(\vec{x}_t, \vec{h}_{t-1})$$

Trong đó:

- \vec{h}_t : Đầu ra của Forward LSTM tại bước t .
- \vec{x}_t : Đầu vào tại bước t .
- \vec{h}_{t-1} : Đầu ra của bước trước.

Backward Long Short- Term Memory

Ngược lại, Backward LSTM sẽ xử lý chuỗi dữ liệu từ bước thời gian cuối cùng $t = T$ về bước đầu tiên $t = 1$. Tương tự như Forward LSTM, tại mỗi bước t , Backward LSTM sẽ tính toán đầu ra $\overleftarrow{h}_t^{\text{backward}}$ dựa trên thông tin từ bước tiếp theo $\overleftarrow{h}_{t+1}^{\text{backward}}$ và đầu vào tại bước t .

Công thức tính toán trong LSTM (Backward):

$$\overleftarrow{h}_t = \text{LSTM}(\overleftarrow{x}_t, \overleftarrow{h}_{t+1})$$

Trong đó:

- \overleftarrow{h}_t : Đầu ra của Backward LSTM tại bước t .
- \overleftarrow{x}_t : Đầu vào tại bước t .
- \overleftarrow{h}_{t+1} : Đầu ra của bước tiếp theo.

3.6 Xây dựng và huấn luyện mô hình AI

Quá trình xây dựng và huấn luyện mô hình trí tuệ nhân tạo nhằm ứng dụng trong cảnh báo sớm bệnh rối loạn nhịp tim. Việc huấn luyện mô hình được thực hiện dựa trên các kỹ thuật học sâu (Deep Learning), cụ thể là sự kết hợp giữa Convolutional Neural Network (CNN) và Bidirectional Long Short-Term Memory (Bi-LSTM), để khai thác hiệu quả tín hiệu điện tâm đồ (ECG). Các bước

thực hiện được mô tả cụ thể, từ việc chuẩn bị dữ liệu, huấn luyện, đánh giá, đến điều chỉnh siêu tham số, nhằm đảm bảo mô hình đạt được độ chính xác và hiệu suất tối ưu.

Quá trình huấn luyện mô hình trong bài toán tái lập tín hiệu điện tâm đồ (ECG) được thực hiện theo một quy trình có hệ thống, nhằm xây dựng mô hình học sâu có khả năng dự đoán chính xác tín hiệu ECG dựa trên dữ liệu đầu vào. Đây là bài toán hồi quy (regression), tập trung vào việc giảm thiểu sai số giữa giá trị dự đoán và giá trị thực tế của tín hiệu. Các bước lý thuyết cơ bản bao gồm như sau:

- Bước 1: Tiền xử lý dữ liệu
 - Chuẩn hóa tín hiệu ECG để loại bỏ nhiễu và đảm bảo tính đồng nhất của dữ liệu, giúp mô hình học được các đặc trưng quan trọng của tín hiệu.
 - Phân chia dữ liệu thành các tập con: tập huấn luyện (training set) để học mô hình, tập kiểm tra (validation set) để theo dõi hiệu suất, và tập thử nghiệm (test set) để đánh giá cuối cùng, thường theo tỷ lệ chuẩn như 70-15-15.
 - Phân đoạn tín hiệu ECG thành các chuỗi thời gian có độ dài cố định, đảm bảo dữ liệu đầu vào phù hợp với mục tiêu tái lập tín hiệu liên tục.
- Bước 2: Thiết kế kiến trúc mô hình
 - Xây dựng mô hình học sâu kết hợp Convolutional Neural Network (CNN) để trích xuất đặc trưng không gian từ tín hiệu ECG và Bidirectional Long Short-Term Memory (Bi-LSTM) để phân tích mối quan hệ thời gian, nhằm tái tạo chính xác các mẫu tín hiệu theo cả hai chiều.
 - Tích hợp các cơ chế như residual connections trong CNN để tăng khả năng học các đặc trưng phức tạp, đồng thời sử dụng kỹ thuật tổng hợp thông tin (ví dụ: pooling toàn cục) để cải thiện chất lượng dự đoán.
 - Thiết kế lớp đầu ra của mô hình dưới dạng hồi quy, trả về giá trị liên tục đại diện cho tín hiệu ECG tái lập.
- Bước 3: Huấn luyện mô hình
 - Sử dụng hàm mất mát phù hợp cho bài toán hồi quy, chẳng hạn Mean Squared Error (MSE), để đo lường độ lệch giữa tín hiệu ECG dự đoán và tín hiệu thực tế, nhằm tối ưu hóa quá trình học.

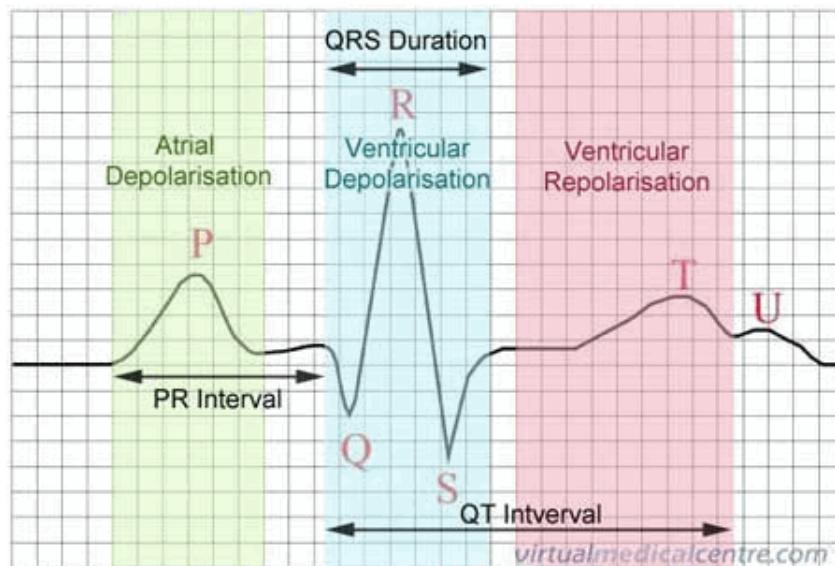
- Áp dụng các thuật toán tối ưu hóa như Gradient Descent hoặc các biến thể nâng cao (ví dụ: Adam) để điều chỉnh trọng số mô hình, giảm thiểu sai số dự đoán.
 - Tích hợp các kỹ thuật chống quá khớp như Dropout (loại bỏ ngẫu nhiên một tỷ lệ nơ-ron trong quá trình huấn luyện) và regularization (kiểm soát độ lớn trọng số) để đảm bảo mô hình có khả năng tổng quát hóa trên dữ liệu mới.
 - Theo dõi hiệu suất trong quá trình huấn luyện bằng các chỉ số như Mean Absolute Error (MAE), Mean Squared Error (MSE) và Root Mean Squared Error (RMSE), phản ánh mức độ chính xác của tín hiệu tái lập so với giá trị thực tế.
- Bước 4: Lưu trữ và chuẩn bị đánh giá
 - Lưu trữ cấu hình và trọng số của mô hình sau khi huấn luyện, nhằm sử dụng trong giai đoạn đánh giá hoặc tái lập tín hiệu trên các tập dữ liệu khác.
 - Chuẩn bị mô hình để đánh giá lý thuyết, tập trung vào việc so sánh tín hiệu dự đoán với tín hiệu thực tế, sử dụng các chỉ số MAE, MSE và RMSE để đo lường hiệu quả tái lập.

CHƯƠNG 4 KẾT QUẢ HUẤN LUYỆN MÔ HÌNH

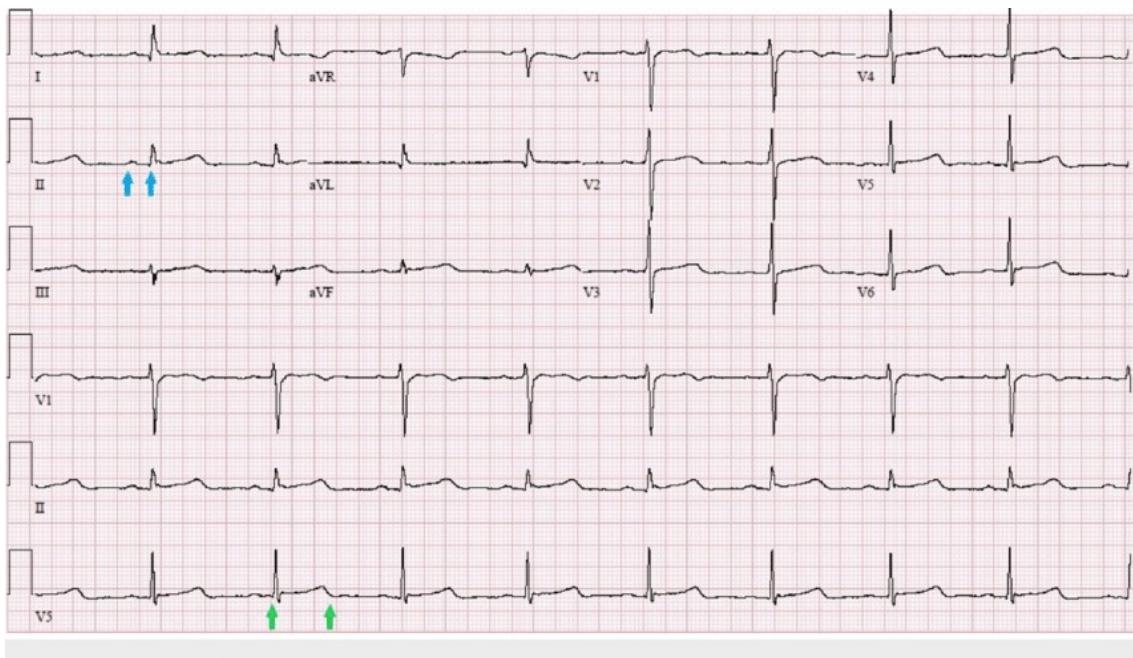
4.1 Tín hiệu điện tâm đồ và tập dữ liệu MIT-BIH Arrhythmia

4.1.1 Tín hiệu điện tâm đồ (Electrocardiogram – ECG)

Theo thống kê của Tổ chức Y tế Thế giới (WHO), mỗi năm có khoảng 17,9 triệu người tử vong do các bệnh tim mạch, chẳng hạn như rung tâm thất, ngất đột ngột, hoặc đột quy [2]. Việc phát hiện sớm các dấu hiệu bất thường trong hoạt động tim không chỉ hỗ trợ bác sĩ chẩn đoán chính xác mà còn cung cấp cảnh báo kịp thời cho bệnh nhân, góp phần giảm tỷ lệ tử vong. Để đánh giá nguy cơ bệnh lý tim mạch, các bác sĩ chuyên khoa thường sử dụng điện tâm đồ (Electrocardiogram – ECG) – một phương pháp ghi lại hoạt động điện của tim thông qua các điện cực đặt trên cơ thể. Tín hiệu ECG điển hình được thể hiện trong Hình 4.1. Về bản chất, tín hiệu này là các xung điện do tim tạo ra, được đo bằng các điện cực đặt trên các chi và ngực theo quy trình chuẩn, tạo thành 12 chuyển đạo điện tâm đồ. Những chuyển đạo này cung cấp thông tin toàn diện về hoạt động điện tim của bệnh nhân, như minh họa trong Hình 4.2.



Hình 4.1: Tín hiệu ECG điển hình (Nguồn: www.dieutri.vn)



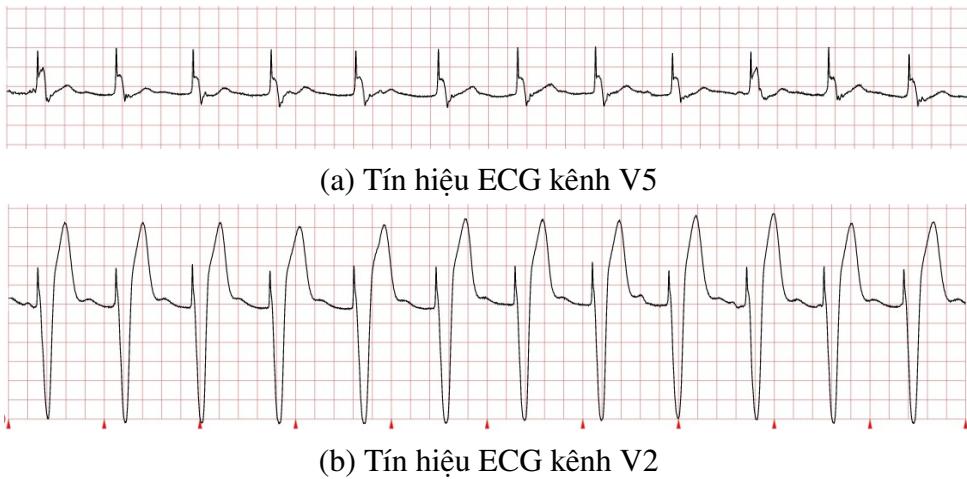
Hình 4.2: Tín hiệu ECG gồm 12 chuyển đạo điện tâm đồ

4.1.2 Tập dữ liệu MIT-BIH Arrhythmia

Trong khóa luận này, chúng tôi sử dụng tập dữ liệu MIT-BIH Arrhythmia Database [22] để phân tích và tái lập tín hiệu ECG. Tập dữ liệu này bao gồm 47 bản ghi tín hiệu ECG từ người trưởng thành, cả bệnh nhân lẫn người khỏe mạnh ở các độ tuổi khác nhau. Mỗi bản ghi kéo dài 48 giờ và được lấy mẫu với tần số 360 mẫu mỗi giây. Tập dữ liệu tập trung vào hai kênh chính trong 12 chuyển đạo điện tâm đồ là V5 và V2. Hình 4.3 minh họa tín hiệu ECG của bản ghi số 102, với các kênh V5 và V2 được hiển thị chi tiết trong Hình 4.3a và Hình 4.3b. Mỗi ô trên đồ thị tương ứng với 2 ms.

Với tần số lấy mẫu 360 mẫu/giây và thời gian ghi 48 giờ, mỗi bản ghi trong tập MIT-BIH Arrhythmia chứa khoảng 62.208.000 giá trị rời rạc, phản ánh chi tiết các xung điện tim. Để huấn luyện mô hình, chúng tôi đã tiền xử lý dữ liệu để loại bỏ nhiễu (như nhiễu cơ thể hoặc nhiễu điện lưới) và chuẩn hóa tín hiệu về khoảng [-1, 1] nhằm phù hợp với yêu cầu của các thuật toán học sâu. Trong thí nghiệm, nhóm sử dụng 100.000 tín hiệu ECG từ tập dữ liệu này để huấn luyện mô hình kết hợp 1D-CNN và Bi-LSTM, nhằm tái lập tín hiệu và phát hiện các bất thường về nhịp tim.

Tín hiệu ECG thực tế là những tín hiệu xung điện được đo bằng mV . Do vậy, để phù hợp trong huấn luyện mô hình Trí tuệ nhân tạo chúng tôi sẽ chuẩn hóa dữ liệu về khoảng [-1, 1].



Hình 4.3: Tín hiệu ECG của bản ghi 102 trong tập MIT-BIH Arrhythmia Database, bao gồm kênh V5 và V2

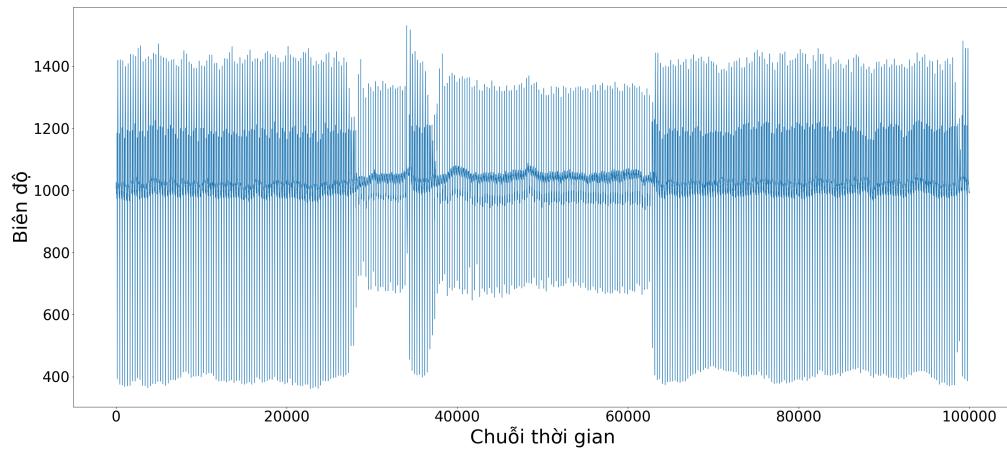
4.2 Mô hình 1D CNN & Bi-LSTM - Tái lập tín hiệu ECG

Mô hình kết hợp giữa Convolutional Neural Network một chiều (1D CNN) và Bidirectional Long Short-Term Memory (Bi-LSTM), được thiết kế để tái lập tín hiệu điện tâm đồ (ECG). Mô hình được xây dựng với các khối residual trong CNN để tăng khả năng trích xuất đặc trưng, kết hợp với Bi-LSTM để khai thác mối quan hệ thời gian, và cuối cùng tổng hợp thông tin để dự đoán tín hiệu liên tục. Quá trình huấn luyện và đánh giá được thực hiện nhằm đảm bảo hiệu quả tái lập tín hiệu so với giá trị thực tế.

4.2.1 Dữ liệu đầu vào

Dữ liệu đầu vào của mô hình bao gồm các đoạn tín hiệu ECG từ hai nguồn: tập dữ liệu gốc MIT-BIH Arrhythmia Database và một tập dữ liệu tùy chỉnh. Mỗi đoạn có kích thước cố định là (300, 1) tương ứng 300 điểm thời gian trên một kênh. Các tín hiệu đã qua tiền xử lý để loại bỏ nhiễu và chuẩn hóa, đảm bảo tính đồng nhất và phù hợp với kiến trúc mô hình 1D-CNN và Bi-LSTM dùng để tái lập tín hiệu ECG.

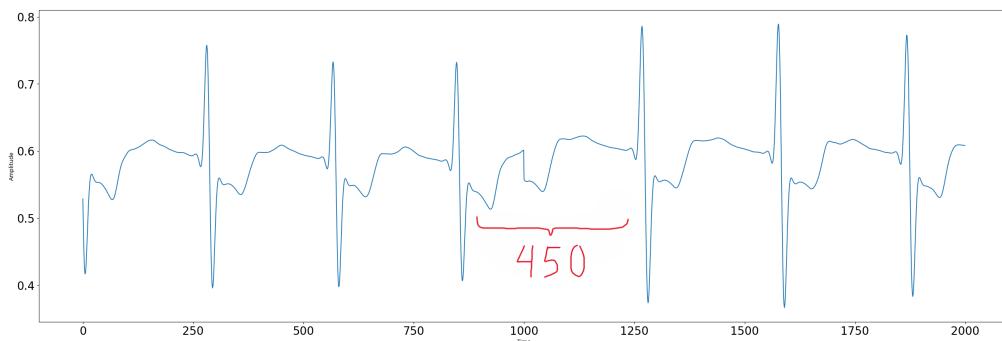
- Tập dữ liệu gốc:** Dữ liệu được trích xuất từ bản ghi 102 của MIT-BIH Arrhythmia Database [22], gồm 100.000 điểm tín hiệu ECG thu từ một bệnh nhân. Bản ghi này được chọn do tín hiệu có chu kỳ ổn định, phù hợp để tái lập tín hiệu ECG. Hình 4.4 minh họa tín hiệu gốc của bản ghi 102.



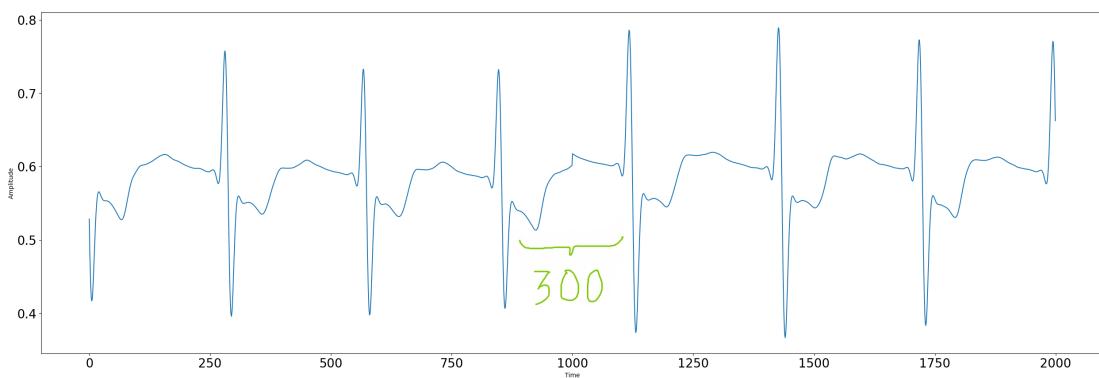
Hình 4.4: Tín hiệu ECG thô của bản ghi 102 (100.000 điểm dữ liệu)

- Tập dữ liệu tùy chỉnh:** Để tăng cường khả năng phát hiện rối loạn nhịp tim, tập dữ liệu tùy chỉnh được tạo từ bản ghi 102 của MIT-BIH Arrhythmia Database bằng kỹ thuật cắt ghép tín hiệu. Các đoạn chứa rối loạn nhịp tim được trích xuất từ tín hiệu gốc, sau đó nhân đôi các đoạn này và ghép lại với nhau, giữ tổng cộng 100.000 điểm dữ liệu. Quá trình này làm tăng tỷ lệ mẫu bất thường, giúp mô hình 1D-CNN và Bi-LSTM học tốt hơn các đặc trưng phức tạp.

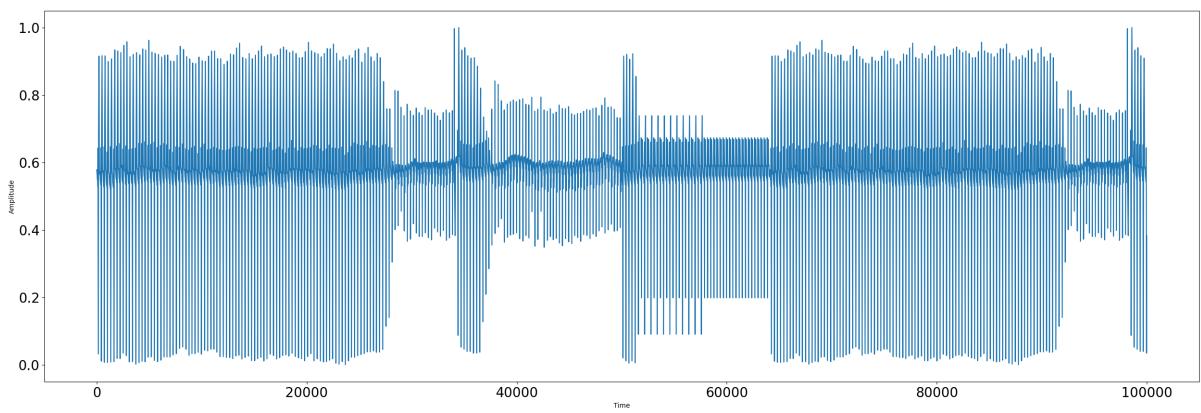
Trong quá trình tăng cường dữ liệu, việc cắt ghép được thực hiện cẩn thận để đảm bảo tính nhất quán giữa các chu kỳ tín hiệu ECG. Nếu không giữ được tính liên tục, tín hiệu sau khi ghép có thể bị đứt gãy, làm mất đặc trưng sinh lý của nhịp tim, như minh họa trong Hình 4.5. Để khắc phục, các đoạn được điều chỉnh sao cho chuỗi thời gian giữa các chu kỳ khớp nhau, duy trì tính nhất quán, như thể hiện trong Hình 4.6. Sau khi hoàn tất, tập dữ liệu tùy chỉnh được kiểm tra để đảm bảo tất cả các đoạn tăng cường giữ được đặc trưng tự nhiên của ECG. Hình 4.7 cho thấy tập dữ liệu tùy chỉnh hoàn chỉnh sau khi cắt ghép.



Hình 4.5: Tín hiệu sau khi tăng cường dữ liệu - không có tính nhất quán giữa các chu kỳ

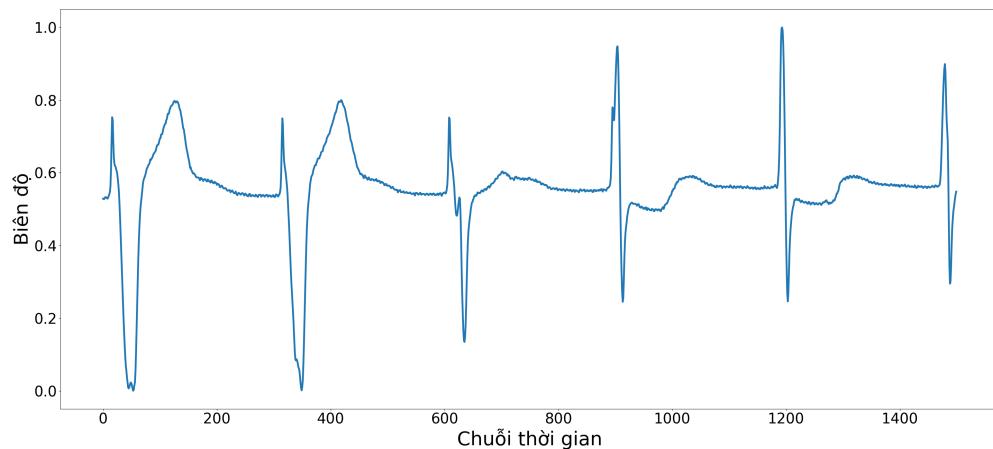


Hình 4.6: Tín hiệu sau khi tăng cường dữ liệu - có tính nhất quán giữa các chu kỳ



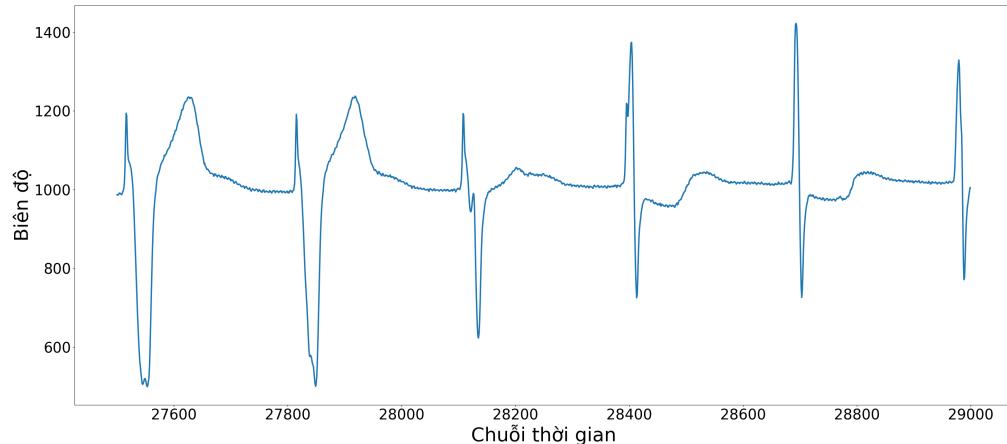
Hình 4.7: Tập dữ liệu tùy chỉnh sau khi cắt ghép

- **Tiền xử lý dữ liệu:** Tín hiệu từ cả hai tập được xử lý như sau:
 - **Chuẩn hóa dữ liệu:** Giá trị điện áp của tín hiệu ECG được chuẩn hóa về khoảng $[-1, 1]$ để đảm bảo mô hình học hiệu quả hơn và tránh ảnh hưởng từ sự khác biệt về thang đo. Hình 4.8 minh họa tín hiệu sau khi chuẩn hóa.

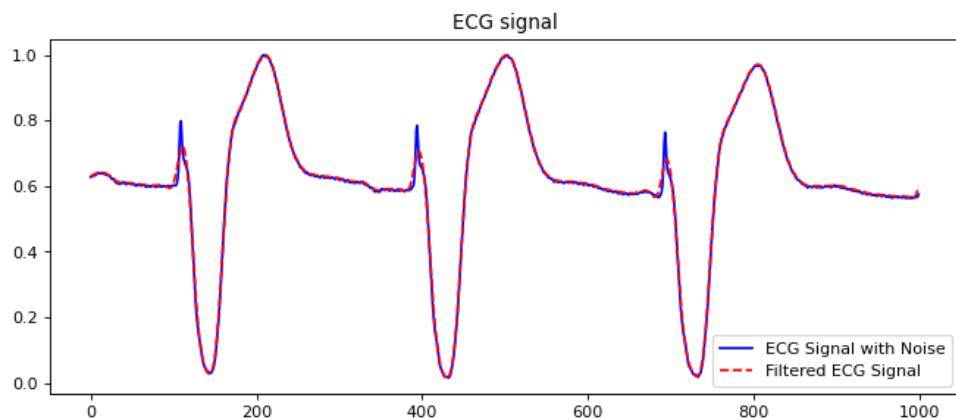


Hình 4.8: Tín hiệu ECG bản ghi 102 sau khi chuẩn hóa

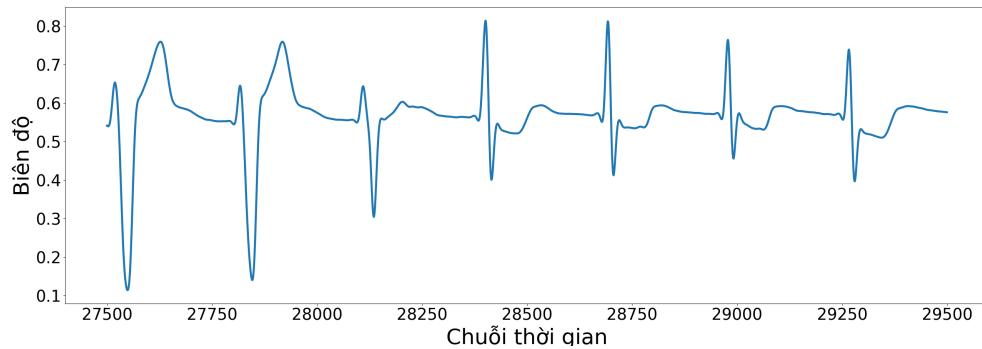
- **Lọc nhiễu:** Tín hiệu sau khi chuẩn hóa được xử lý để loại bỏ nhiễu không mong muốn (như nhiễu từ thiết bị hoặc môi trường). Hình 4.9 cho thấy tín hiệu gốc của bản ghi 102, Hình 4.10 minh họa quá trình làm sạch nhiễu từ dữ liệu gốc, và Hình 4.11 thể hiện tín hiệu sau khi hoàn tất quá trình lọc nhiễu.



Hình 4.9: Tín hiệu ECG gốc của bản ghi 102

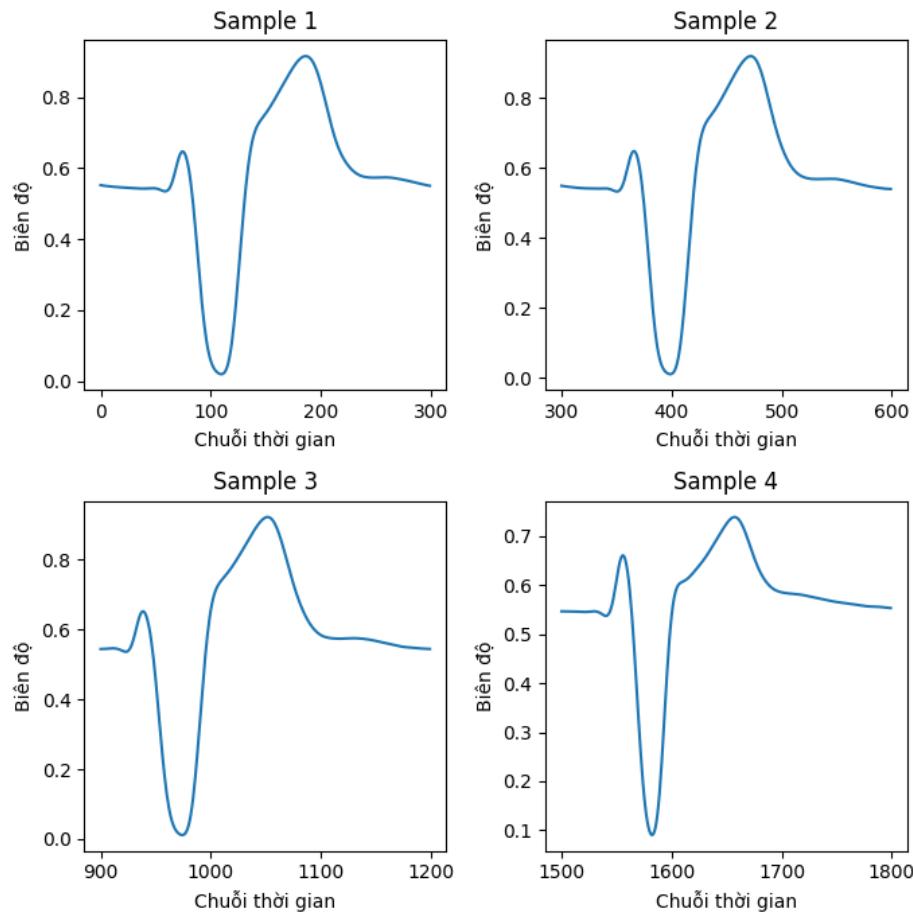


Hình 4.10: Quá trình làm sạch nhiễu tín hiệu ECG từ dữ liệu gốc



Hình 4.11: Tín hiệu ECG bản ghi 102 sau khi lọc nhiễu

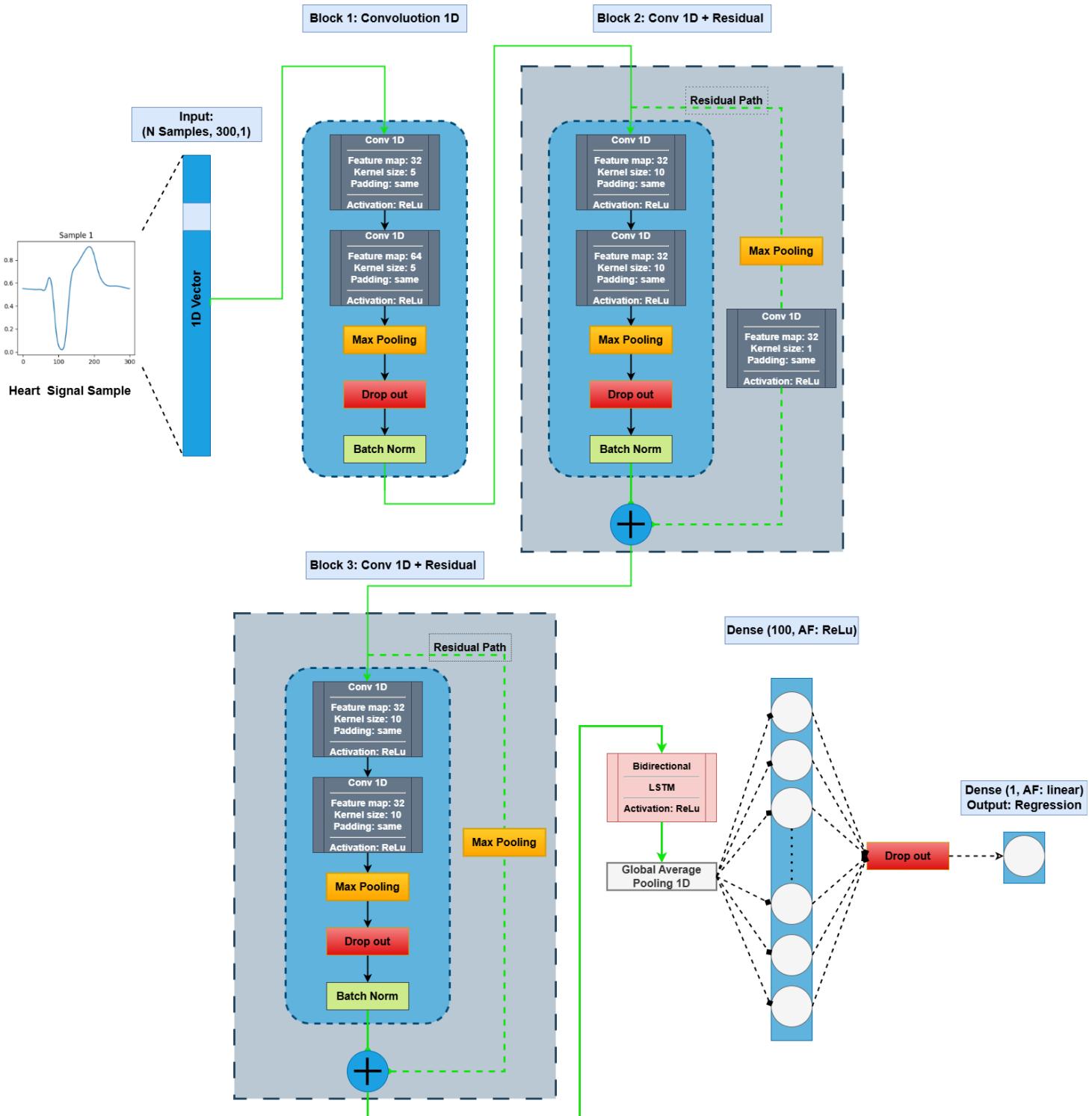
- **Phân đoạn dữ liệu:** Tín hiệu từ cả hai tập được chia thành các đoạn 300 điểm thời gian, tạo đầu vào cho mô hình 1D-CNN và Bi-LSTM.
Hình 4.12 thể hiện các đoạn tín hiệu từ 0 đến 300 điểm.



Hình 4.12: Đoạn tín hiệu ECG từ 0 đến 300 điểm

- **Chia tập dữ liệu:** Dữ liệu sau tiền xử lý được chia thành các tập huấn luyện, xác thực, và kiểm tra để huấn luyện và đánh giá mô hình. (Phần này sẽ được trình bày chi tiết trong các mục sau.)

4.2.2 Kiến trúc mô hình CNN-BiLSTM



Hình 4.13: Kiến trúc mô hình CNN-BiLSTM

- **Khối CNN ban đầu:**

- Hai lớp Conv1D liên tiếp: Lớp đầu tiên với 32 bộ lọc (kernel size 5), lớp thứ hai với 64 bộ lọc (kernel size 5), đều sử dụng hàm kích hoạt ReLU và padding 'same' để giữ nguyên kích thước không gian.
- Lớp MaxPooling1D (pool size 2) giảm kích thước không gian từ 300 xuống 150, giữ lại các đặc trưng nổi bật.
- Dropout (0.2) và BatchNormalization được thêm vào để chống quá khớp và ổn định quá trình học.

- **Khối Residual CNN thứ nhất:**

- Đầu vào khối có shape (150, 64). Hai lớp Conv1D với 32 bộ lọc (kernel size 10, ReLU, padding 'same') trích xuất đặc trưng sâu hơn.
- MaxPooling1D (pool size 2) giảm kích thước xuống (75, 32).
- Đường residual: Đầu vào được điều chỉnh kích thước bằng MaxPooling1D (pool size 2) và Conv1D (32 bộ lọc, kernel size 1) để khớp với đầu ra, sau đó kết hợp qua phép cộng (Add).
- Dropout (0.2) và BatchNormalization đảm bảo tính ổn định và tổng quát hóa.

- **Khối Residual CNN thứ hai:**

- Đầu vào khối có shape (75, 32). Tương tự khối trước, hai lớp Conv1D (32 bộ lọc, kernel size 10, ReLU, padding 'same') được áp dụng, theo sau là MaxPooling1D (pool size 2) giảm kích thước xuống (37, 32).
- Đường residual: Đầu vào được giảm kích thước bằng MaxPooling1D (pool size 2), sau đó kết hợp trực tiếp với đầu ra qua phép cộng (Add) do số kenh đã khớp.
- Dropout (0.2) và BatchNormalization tiếp tục được sử dụng.

- **Lớp Bi-LSTM:**

- Bidirectional LSTM với 64 đơn vị ẩn, return_sequences=True, xử lý dữ liệu có shape (37, 32) theo cả hai chiều thời gian.
- Áp dụng recurrent dropout (0.2) và L2 regularization (0.001) để giảm quá khớp và kiểm soát trọng số.

- **Tổng hợp và đầu ra:**

- GlobalAveragePooling1D tổng hợp thông tin từ Bi-LSTM thành một vector duy nhất.
- Lớp Dense (100 đơn vị, ReLU) với L2 regularization (0.001) và Dropout (0.3) xử lý đặc trưng cấp cao.
- Lớp Dense cuối cùng (1 đơn vị, linear) dự đoán giá trị tín hiệu ECG liên tục.

4.2.3 Khởi tạo tham số

Để tái lập tín hiệu điện tâm đồ (ECG) bằng mô hình kết hợp 1D CNN và Bi-LSTM, việc khởi tạo các tham số đóng vai trò quan trọng trong việc đảm bảo hiệu quả huấn luyện và khả năng tổng quát hóa của mô hình. Các tham số được lựa chọn dựa trên đặc điểm của bài toán hồi quy và cấu trúc mạng phức tạp với residual blocks. Dưới đây là mô tả về các tham số chính cần khởi tạo:

- **Input length:** Đây là độ dài của chuỗi tín hiệu ECG đầu vào, xác định số lượng điểm thời gian được sử dụng để tái lập tín hiệu. Trong nghiên cứu này, giá trị được chọn là 300, tương ứng với kích thước đầu vào (300, 1), nhằm bao quát đầy đủ các đặc trưng thời gian quan trọng trong dữ liệu MIT-BIH Arrhythmia.
- **Dropout:** Tham số này quy định tỷ lệ đơn vị bị loại bỏ ngẫu nhiên trong quá trình huấn luyện để giảm hiện tượng quá khớp (overfitting). Mô hình sử dụng các mức Dropout khác nhau: 0.2 cho các lớp CNN và Bi-LSTM, và 0.3 cho lớp Dense cuối cùng, nhằm cân bằng giữa khả năng học và tổng quát hóa.
- **Epochs:** Số lần lặp qua toàn bộ tập dữ liệu trong quá trình huấn luyện. Giá trị này được khởi tạo dựa trên thử nghiệm ban đầu, thường nằm trong khoảng 50 đến 100, và sẽ được điều chỉnh dựa trên hiệu suất trên tập kiểm tra.
- **Batch Size:** Số lượng mẫu dữ liệu được xử lý trong mỗi lần cập nhật trọng số. Batch Size ảnh hưởng đến tốc độ huấn luyện và độ ổn định của gradient, trong nghiên cứu này batch size đưa vào sẽ là 64 samples.
- **L2 Regularization:** Tham số kiểm soát độ lớn của trọng số trong mô hình bằng cách thêm một khoản phạt vào hàm mất mát. Giá trị 0.001 được áp dụng cho các lớp Bi-LSTM và Dense để hạn chế sự phức tạp và cải thiện khả năng tổng quát hóa.
- **Learning Rate for Adam Optimizer:** Tốc độ học của thuật toán tối ưu hóa Adam, quyết định bước cập nhật trọng số trong mỗi iteration. Giá trị

khởi tạo là 0.0001, đủ nhỏ để đảm bảo hội tụ ổn định trong bài toán tái lập tín hiệu.

Bảng 4.1: Tham số khởi tạo cho mô hình CNN-BiLSTM

Tham số	Giá trị khởi tạo
Input length	300
Dropout	0.2 (CNN, Bi-LSTM); 0.3 (Dense)
Epochs	50
Batch Size	64
Hàm tối ưu hóa	Adam
Learning Rate	0.0001
L2 Regularization	0.001

Các giá trị khởi tạo trên được lựa chọn dựa trên đặc trưng của mô hình và bài toán tái lập tín hiệu ECG. Trong quá trình huấn luyện, các tham số như Epochs và Batch Size sẽ được tinh chỉnh thông qua thử nghiệm trên tập dữ liệu, nhằm đạt được hiệu suất tối ưu về các chỉ số Mean Absolute Error (MAE), Mean Squared Error (MSE) và Root Mean Squared Error (RMSE).

4.2.4 Huấn luyện mô hình

Huấn luyện mô hình là quá trình tối ưu hóa các tham số của mạng học sâu nhằm tái lập tín hiệu điện tâm đồ (ECG) một cách chính xác trên dữ liệu mới. Quá trình này bao gồm việc chuẩn bị dữ liệu, thiết lập kiến trúc mô hình, lựa chọn hàm mất mát và thuật toán tối ưu hóa, huấn luyện trên tập dữ liệu, và tinh chỉnh các tham số để đạt hiệu suất mong muốn. Với mô hình kết hợp 1D-CNN và Bi-LSTM có residual blocks, mục tiêu là giảm thiểu sai số giữa tín hiệu dự đoán và tín hiệu thực tế, được đo bằng các chỉ số như Mean Squared Error (MSE), Mean Absolute Error (MAE) và Root Mean Squared Error (RMSE).

Mô hình CNN-BiLSTM được huấn luyện trên tập dữ liệu MIT-BIH Arrhythmia đã qua tiền xử lý, với các tham số như kích thước đầu vào, số đơn vị trong lớp Bi-LSTM, và tốc độ học được tối ưu hóa để đảm bảo khả năng tái lập tín hiệu ECG hiệu quả. Quá trình huấn luyện tận dụng residual blocks để tăng độ sâu của mạng mà không làm mất thông tin, kết hợp với Bi-LSTM để khai thác đặc trưng thời gian hai chiều.

Bảng 4.2: Thông số chi tiết của các lớp trong mô hình CNN-BiLSTM

Layer (Type)	Kích thước đầu ra	Tham số
input_layer_1 (InputLayer)	(None, 300, 1)	0
conv1d_7 (Conv1D)	(None, 300, 32)	192
conv1d_8 (Conv1D)	(None, 300, 64)	10,304
max_pooling1d_5 (MaxPooling1D)	(None, 150, 64)	0
dropout_4 (Dropout)	(None, 150, 64)	0
batch_normalization_3 (BatchNormalization)	(None, 150, 64)	256
conv1d_9 (Conv1D)	(None, 150, 32)	20,512
conv1d_10 (Conv1D)	(None, 150, 32)	10,272
max_pooling1d_6 (MaxPooling1D)	(None, 75, 32)	0
dropout_5 (Dropout)	(None, 75, 32)	0
max_pooling1d_7 (MaxPooling1D)	(None, 75, 64)	0
batch_normalization_4 (BatchNormalization)	(None, 75, 32)	128
conv1d_11 (Conv1D)	(None, 75, 32)	2,080
add_2 (Add)	(None, 75, 32)	0
conv1d_12 (Conv1D)	(None, 75, 32)	10,272
conv1d_13 (Conv1D)	(None, 75, 32)	10,272
max_pooling1d_8 (MaxPooling1D)	(None, 37, 32)	0
dropout_6 (Dropout)	(None, 37, 32)	0
batch_normalization_5 (BatchNormalization)	(None, 37, 32)	128
max_pooling1d_9 (MaxPooling1D)	(None, 37, 32)	0
add_3 (Add)	(None, 37, 32)	0
bidirectional_1 (Bidirectional)	(None, 37, 128)	49,664
global_average_pooling1d_1 (GlobalAveragePooling1D)	(None, 128)	0
dense_2 (Dense)	(None, 100)	12,900
dropout_7 (Dropout)	(None, 100)	0
dense_3 (Dense)	(None, 1)	101
Total params: 127,081 (496.41 KB)		
Trainable params: 126,825 (495.41 KB)		
Non-trainable params: 256 (1.00 KB)		

Quá trình huấn luyện được thực hiện như sau:

- **Layer CNN ban đầu:** Hai lớp Conv1D (32 và 64 bộ lọc, kernel size 5) trích xuất đặc trưng không gian từ tín hiệu ECG đầu vào có kích thước (300,1). Lớp MaxPooling1D giảm kích thước xuống (150,64), kết hợp Dropout (0.2) và BatchNormalization để ổn định và chống quá khớp.
- **Residual Blocks:** Hai khối residual lặp lượt xử lý dữ liệu từ (150,64) xuống (75,32) và (37,32), mỗi khối gồm hai lớp Conv1D (32 bộ lọc, kernel size 10) và kết nối tắt qua phép cộng (Add). Các lớp Dropout (0.2) và BatchNormalization tiếp tục được sử dụng để tăng hiệu quả học.
- **Lớp Bi-LSTM:** Bidirectional LSTM với 128 đơn vị (64 mỗi chiều) nhận đầu vào (37,32), trích xuất đặc trưng thời gian hai chiều với 49,664 tham số, tích hợp recurrent dropout (0.2) và L2 regularization (0.001).

- **Tổng hợp và đầu ra:** GlobalAveragePooling1D chuyển đổi đầu ra Bi-LSTM thành vector (128), sau đó qua lớp Dense (100 đơn vị, ReLU) với Dropout (0.3) và lớp Dense cuối (1 đơn vị, linear) để dự đoán tín hiệu ECG liên tục.
- **Tổng số tham số:** Mô hình có 127,081 tham số, trong đó 126,825 tham số có thể huấn luyện và 256 tham số cố định từ các lớp BatchNormalization.

Mô hình được huấn luyện với hàm mất mát Mean Squared Error (MSE) và optimizer Adam (learning rate 0.0001), sử dụng batch size là 64 và số epoch từ 50–100, tùy chỉnh dựa trên hiệu suất trên tập kiểm tra. Các kỹ thuật như Dropout, L2 Regularization và BatchNormalization được áp dụng để tăng tốc độ hội tụ, ngăn chặn quá khớp và cải thiện khả năng tái lập tín hiệu ECG.

4.2.5 Kết quả và đánh giá

Mục tiêu đánh giá:

Phần này nhằm trình bày kết quả huấn luyện mô hình CNN-BiLSTM trong việc tái lập tín hiệu điện tâm đồ (ECG) và đánh giá hiệu suất của mô hình trên hai tập dữ liệu đã đề cập ở mục 4.2.5 – tập gốc từ MIT-BIH Arrhythmia và tập tùy chỉnh.

Mục tiêu chính là đo lường khả năng tái lập tín hiệu ECG của mô hình thông qua các chỉ số hiệu suất như Mean Squared Error (MSE), Mean Absolute Error (MAE), và Root Mean Squared Error (RMSE), đồng thời so sánh tác động của việc tăng cường dữ liệu trong tập tùy chỉnh đến độ chính xác và khả năng phát hiện rối loạn nhịp tim.

Kết quả huấn luyện sẽ bao gồm các giá trị định lượng của các chỉ số trên, cùng với phân tích sự khác biệt giữa hai tập dữ liệu để làm rõ hiệu quả của phương pháp đề xuất.

Phương pháp đánh giá:

Mô hình CNN-BiLSTM được huấn luyện riêng biệt trên hai tập dữ liệu – tập gốc và tập tùy chỉnh – đã được trình bày ở mục 4.2.5. Cấu hình mô hình gồm các khối residual CNN để trích xuất đặc trưng không gian, lớp Bi-LSTM để học các phụ thuộc thời gian, và lớp Dense để tái lập tín hiệu đầu ra. Các tham số huấn luyện, được liệt kê chi tiết trong bảng 4.1, bao gồm learning rate 0.0001, batch size 64, và số epoch là 50, được giữ nhất quán cho cả hai tập dữ liệu nhằm đảm bảo tính công bằng trong so sánh.

Để đánh giá hiệu suất, một tập thử nghiệm độc lập được trích xuất từ dữ liệu MIT-BIH Arrhythmia gốc, bao gồm các đoạn tín hiệu ECG không trùng với dữ liệu huấn luyện. Tập thử nghiệm này được tiền xử lý tương tự như dữ liệu huấn luyện (chuẩn hóa và lọc nhiễu), đảm bảo tính đồng nhất. Hiệu suất của mô hình được đo lường bằng các chỉ số sau:

- *Mean Squared Error (MSE)*: Đo lường sai số bình phương trung bình giữa tín hiệu ECG dự đoán và tín hiệu thực tế, giúp đánh giá mức độ sai lệch tổng thể của mô hình trong quá trình tái lập.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.1)$$

Trong đó:

- y_i là giá trị thực tế của tín hiệu ECG tại điểm i
- \hat{y}_i là giá trị dự đoán của mô hình
- n là số lượng điểm dữ liệu.

- *Mean Absolute Error (MAE)*: Tính toán độ lệch trung bình tuyệt đối, cung cấp cái nhìn trực quan về sai số trung bình mà không bị ảnh hưởng bởi các giá trị ngoại lai lớn.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.2)$$

Trong đó:

- $|y_i - \hat{y}_i|$ biểu thị độ lệch tuyệt đối giữa giá trị thực tế và giá trị dự đoán.

- *Root Mean Squared Error (RMSE)*: Đại diện cho sai số trung bình ở cùng đơn vị với tín hiệu gốc, giúp dễ dàng so sánh mức độ chính xác với thang đo của tín hiệu ECG thực tế.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4.3)$$

Trong đó, RMSE là căn bậc hai của MSE, cung cấp sai số trung bình ở cùng đơn vị với tín hiệu gốc.

- *Hệ số xác định (R^2)*: Đo lường mức độ mà mô hình giải thích được sự biến thiên của dữ liệu, với giá trị gần 1 thể hiện mô hình dự đoán tốt.

Hệ số xác định R^2 được tính như sau:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4.4)$$

trong đó:

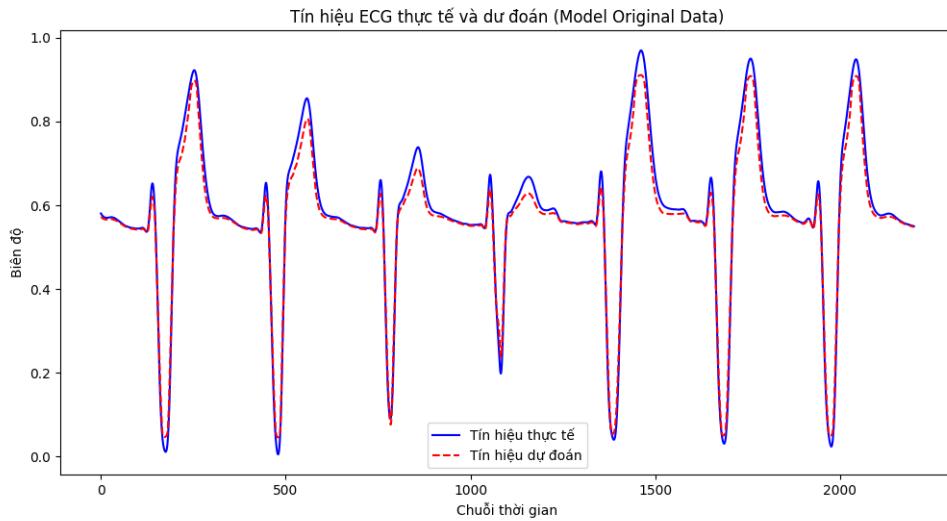
- y_i là giá trị thực tế,
- \hat{y}_i là giá trị dự đoán,
- \bar{y} là giá trị trung bình của các y_i ,
- n là số điểm dữ liệu.

Ngoài ra, để minh họa khả năng tái lập, các biểu đồ so sánh tín hiệu dự đoán và thực tế sẽ được trình bày ở phần kết quả, tập trung vào các đoạn tín hiệu bình thường và bất thường để đánh giá sự khác biệt về hiệu suất.

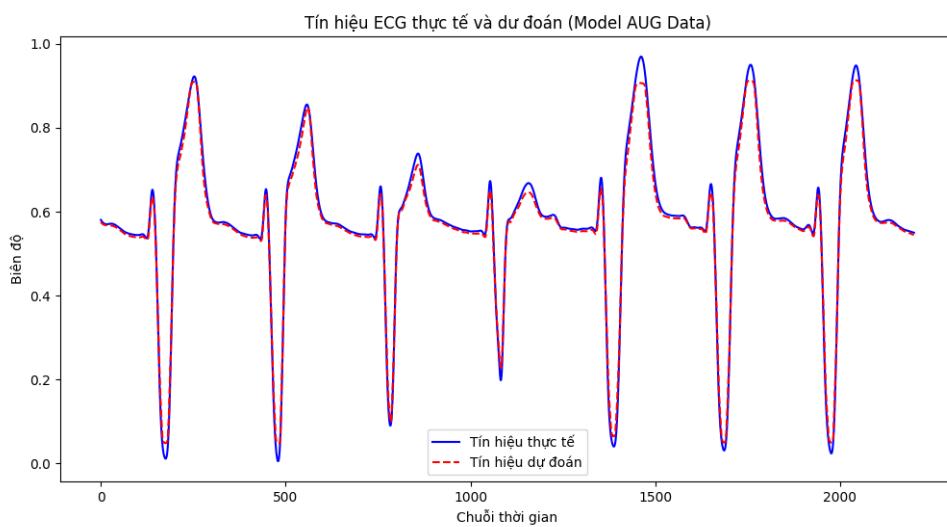
4.3 Kết quả huấn luyện

Kết quả huấn luyện mô hình CNN-BiLSTM được trình bày qua tín hiệu tái lập, chỉ số hiệu suất, và quá trình huấn luyện trên hai tập dữ liệu: gốc và tùy chỉnh.

Đầu tiên, Hình 4.14 và Hình 4.15 so sánh tín hiệu ECG thực tế với tín hiệu tái lập trên **tập test gốc ECG-102**. Cụ thể, Hình 4.14 thể hiện kết quả của mô hình được huấn luyện trên tập dữ liệu gốc, trong khi Hình 4.15 thể hiện kết quả của mô hình được huấn luyện trên tập dữ liệu tùy chỉnh. Các hình này minh họa khả năng tái lập các đoạn tín hiệu ECG, bao gồm cả nhịp bình thường và bất thường.



Hình 4.14: Tín hiệu ECG thực tế và tái lập trên tập test gốc bởi mô hình huấn luyện trên tập dữ liệu gốc



Hình 4.15: Tín hiệu ECG thực tế và tái lập trên tập test gốc bởi mô hình huấn luyện trên tập dữ liệu tùy chỉnh

Sau khi huấn luyện mô hình trên hai tập dữ liệu – tập gốc và tập tùy chỉnh – các chỉ số hiệu suất bao gồm MSE, MAE, RMSE và R^2 được tính toán và trình bày trong Bảng 4.3 dưới đây. Các giá trị này được đánh giá trên tập thử nghiệm độc lập để đảm bảo tính khách quan.

Chỉ số	Mô hình tập gốc	Mô hình tập tùy chỉnh
MSE	0.0007	0.0002
MAE	0.0186	0.0108
RMSE	0.0256	0.0140
R ²	0.9839	0.9952

Bảng 4.3: Chỉ số hiệu suất của hai mô hình trên tập thử nghiệm

Phân tích và so sánh hiệu suất: Dựa trên các chỉ số hiệu suất trong Bảng 4.3, có thể rút ra các nhận xét sau về sự khác biệt giữa hai mô hình:

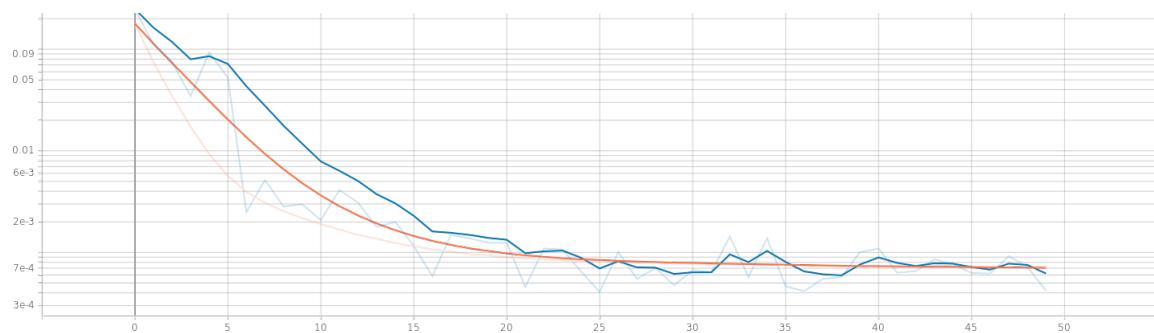
- **MSE (Mean Squared Error):** Mô hình huấn luyện trên tập tùy chỉnh đạt giá trị MSE là 0.0002, thấp hơn đáng kể so với 0.0007 của mô hình huấn luyện trên tập gốc. Điều này cho thấy sai số bình phương trung bình của mô hình trên tập tùy chỉnh nhỏ hơn, phản ánh khả năng dự đoán chính xác cao hơn.
- **MAE (Mean Absolute Error):** Giá trị MAE của mô hình huấn luyện trên tập tùy chỉnh là 0.0108, thấp hơn so với 0.0186 của mô hình huấn luyện trên tập gốc. Kết quả này chỉ ra rằng độ lệch tuyệt đối trung bình giữa giá trị dự đoán và giá trị thực tế của mô hình trên tập tùy chỉnh được giảm thiểu đáng kể.
- **RMSE (Root Mean Squared Error):** RMSE của mô hình huấn luyện trên tập tùy chỉnh là 0.0140, thấp hơn so với 0.0256 của mô hình huấn luyện trên tập gốc. Vì RMSE được tính ở cùng đơn vị với dữ liệu gốc, giá trị này cho thấy mô hình trên tập tùy chỉnh tái lập tín hiệu chính xác hơn.
- **R² (Hệ số xác định):** Mô hình huấn luyện trên tập tùy chỉnh đạt R² là 0.9952, cao hơn so với 0.9839 của mô hình trên tập gốc. Giá trị R² gần 1 hơn cho thấy mô hình trên tập tùy chỉnh giải thích tốt hơn sự biến thiên của dữ liệu, thể hiện khả năng dự đoán vượt trội.

Kết luận: Các chỉ số hiệu suất cho thấy mô hình huấn luyện trên tập dữ liệu tùy chỉnh vượt trội hơn so với mô hình huấn luyện trên tập dữ liệu gốc ở tất cả các khía cạnh: sai số thấp hơn (MSE, MAE, RMSE) và khả năng giải thích dữ liệu tốt hơn (R²). Sự cải thiện này có thể được lý giải bởi việc tập dữ liệu tùy chỉnh đã được tối ưu hóa, có thể bao gồm việc bổ sung các mẫu dữ liệu phức tạp hoặc bất thường, giúp mô hình học được các đặc trưng đa dạng và nâng cao hiệu suất dự đoán. Kết quả này khẳng định tính hiệu quả của quá trình tùy chỉnh dữ liệu trong việc cải thiện chất lượng mô hình.

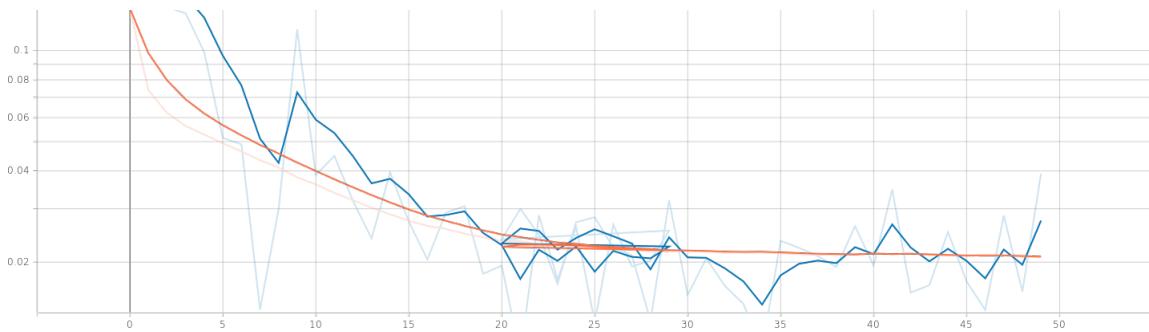
Biểu đồ huấn luyện: Để đánh giá quá trình huấn luyện của hai mô hình, chúng tôi sử dụng các biểu đồ từ TensorBoard. Hình 4.16 và Hình 4.17 thể hiện sự giảm của hàm mất mát qua các epoch trên tập dữ liệu gốc và tập tùy chỉnh. Tương tự, Hình 4.18 và Hình 4.19 cho thấy sự cải thiện của RMSE, trong khi Hình 4.20 và Hình 4.21 minh họa sự giảm của MAE qua các epoch trên hai tập dữ liệu này. Trong các biểu đồ, đường màu cam biểu thị giá trị trên tập huấn luyện (train), còn đường màu xanh biểu thị giá trị trên tập kiểm tra (validation), giúp so sánh hiệu suất giữa hai tập dữ liệu trong quá trình học.



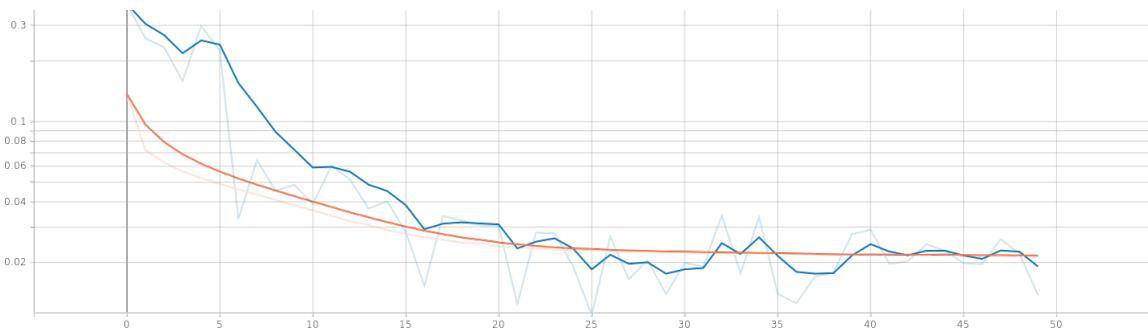
Hình 4.16: Hàm mất mát qua các epoch trên tập dữ liệu gốc



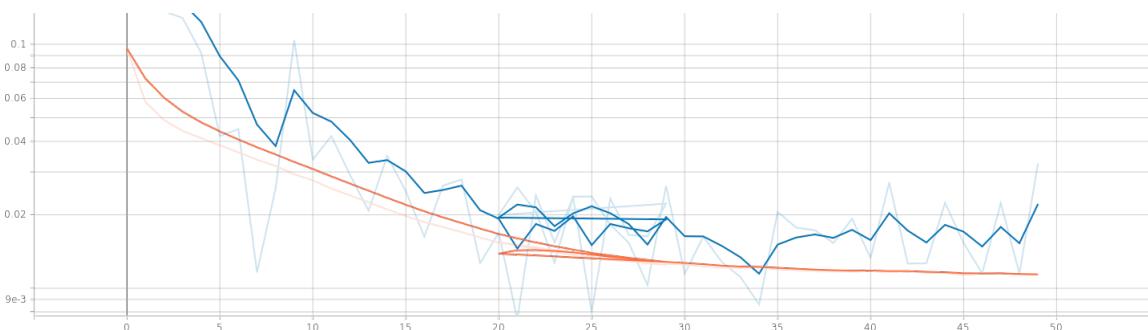
Hình 4.17: Hàm mất mát qua các epoch trên tập dữ liệu tùy chỉnh



Hình 4.18: RMSE qua các epoch trên tập dữ liệu gốc



Hình 4.19: RMSE qua các epoch trên tập dữ liệu tùy chỉnh



Hình 4.20: MAE qua các epoch trên tập dữ liệu gốc



Hình 4.21: MAE qua các epoch trên tập dữ liệu tùy chỉnh

Cụ thể, Hình 4.16 cho thấy hàm mất mát trên tập dữ liệu gốc giảm từ 0.09 xuống khoảng 0.005 sau 50 epoch trên tập train, trong khi tập validation dao động mạnh trong 20 epoch đầu (từ 0.02 đến 0.05) trước khi ổn định quanh 0.005. Điều này cho thấy mô hình trên tập gốc gặp khó khăn trong việc tổng quát hóa. Ngược lại, Hình 4.17 thể hiện sự hội tụ nhanh hơn trên tập tùy chỉnh, với cả hai đường train và validation giảm đều và ổn định hơn, phản ánh lợi ích của dữ liệu huấn luyện đa dạng. Tương tự, các biểu đồ RMSE và MAE (Hình 4.18, Hình 4.19, Hình 4.20, Hình 4.21) cho thấy xu hướng giảm trên cả hai tập dữ liệu, nhưng mô hình trên tập tùy chỉnh đạt giá trị thấp hơn và ít dao động hơn trên tập validation, minh họa khả năng học các đặc trưng phức tạp tốt hơn.

Các biểu đồ trên minh họa quá trình học của mô hình trên cả hai tập dữ liệu. Hàm mất mát, RMSE và MAE đều giảm dần qua các epoch, cho thấy mô hình đã học được các đặc trưng quan trọng của tín hiệu ECG. Đặc biệt, trên tập dữ liệu tùy chỉnh, sự hội tụ diễn ra nhanh hơn với các giá trị thấp hơn đáng kể, phản ánh hiệu quả của việc tăng cường dữ liệu bất thường trong việc cải thiện khả năng tái lập tín hiệu.

4.4 Phân tích và so sánh

Phân tích kết quả chính: Phân tích kết quả trên tập thử nghiệm chính: Dựa trên Bảng 4.3, mô hình huấn luyện trên tập tùy chỉnh vượt trội hơn mô hình trên tập gốc:

- **MSE:** Giá trị MSE giảm từ 0.0007 (tập gốc) xuống 0.0002 (tập tùy chỉnh), cho thấy sai số bình phương trung bình giảm đáng kể, phản ánh khả năng dự đoán chính xác cao hơn.
- **MAE:** MAE giảm từ 0.0186 xuống 0.0108, chỉ ra độ lệch tuyệt đối trung bình giữa giá trị dự đoán và thực tế được giảm thiểu.
- **RMSE:** RMSE giảm từ 0.0256 xuống 0.0140, cho thấy mô hình trên tập tùy chỉnh tái lập tín hiệu chính xác hơn ở cùng đơn vị với dữ liệu gốc.
- **R²:** R² tăng từ 0.9839 lên 0.9952, chứng minh mô hình trên tập tùy chỉnh giải thích tốt hơn sự biến thiên của dữ liệu.

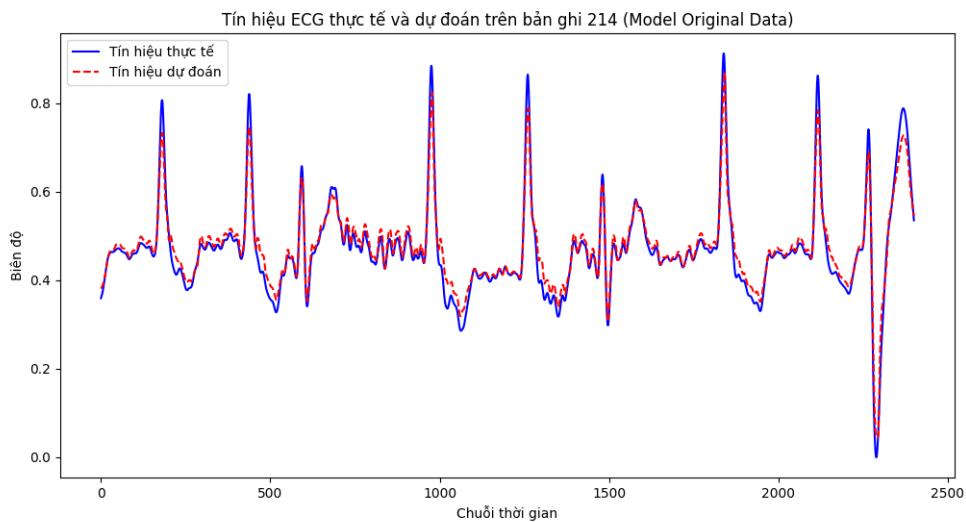
Hình ?? cũng cho thấy tín hiệu tái lập từ mô hình trên tập tùy chỉnh sát với tín hiệu thực tế hơn, đặc biệt ở các đoạn tín hiệu bất thường.

Phân tích bổ sung trên bản ghi 214: Để kiểm tra khả năng khái quát hóa của hai mô hình trên dữ liệu mới, chúng tôi đánh giá hiệu suất trên bản ghi 214 từ cơ sở dữ liệu MIT-BIH Arrhythmia. Bản ghi này gồm 30,000 điểm dữ liệu tín hiệu ECG, đã được lọc nhiễu và chuẩn hóa tương tự dữ liệu huấn luyện, và không được sử dụng trong quá trình huấn luyện, đảm bảo tính độc lập.

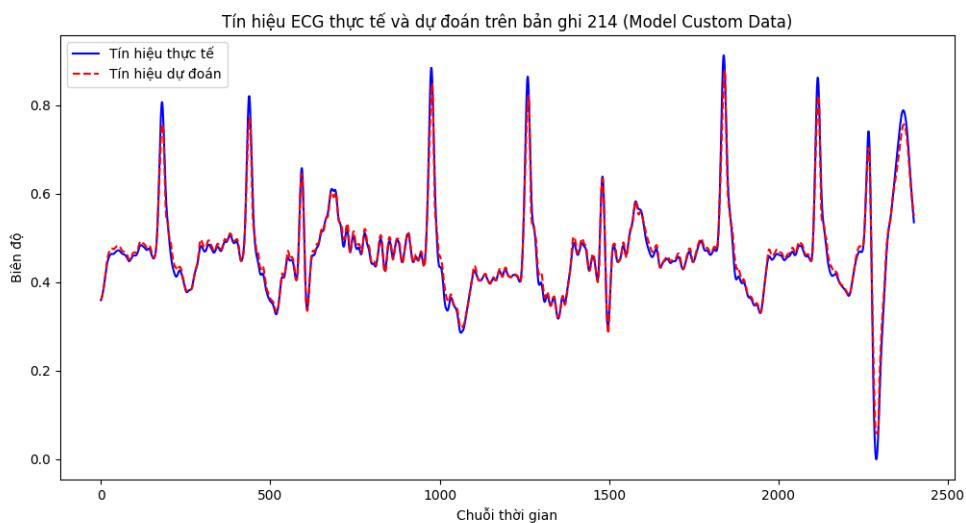
Kết quả được trình bày trong Bảng 4.4 và minh họa qua Hình 4.22 và Hình 4.23.

Chỉ số	Mô hình trên tập gốc	Mô hình trên tập tùy chỉnh
MSE	0.0005	0.0002
MAE	0.0146	0.0086
RMSE	0.0220	0.0146
R ²	0.9568	0.9810

Bảng 4.4: Chỉ số hiệu suất của hai mô hình trên bản ghi 214



Hình 4.22: Tín hiệu ECG thực tế và tái lập trên bản ghi 214 (mô hình huấn luyện trên tập gốc)



Hình 4.23: Tín hiệu ECG thực tế và tái lập trên bản ghi 214 (mô hình huấn luyện trên tập tùy chỉnh)

Phân tích chi tiết:

- **Chỉ số hiệu suất:** Mô hình trên tập tùy chỉnh đạt MSE 0.0002, thấp hơn so với 0.0005 của mô hình trên tập gốc, cho thấy sai số tổng quát giảm đáng kể. MAE giảm từ 0.0146 xuống 0.0086 và RMSE từ 0.0220 xuống 0.0146, chứng minh độ chính xác trung bình được cải thiện. R^2 tăng từ 0.9568 lên 0.9810, phản ánh khả năng tái lập sát với dữ liệu thực tế hơn.
- **So sánh trực quan:** Hình 4.22 cho thấy tín hiệu tái lập từ mô hình trên tập gốc có sai lệch ở các vùng biến thiên mạnh, chẳng hạn như đỉnh sóng hoặc đoạn tín hiệu bất thường. Ngược lại, Hình 4.23 thể hiện tín hiệu tái lập từ mô hình trên tập tùy chỉnh bám sát tín hiệu thực tế, đặc biệt ở các đoạn phức tạp, minh họa khả năng tái hiện chi tiết tốt hơn.

Kết luận: Kết quả trên bản ghi 214 cũng có nhận định rằng mô hình huấn luyện trên tập tùy chỉnh có khả năng khái quát hóa vượt trội so với mô hình trên tập gốc. Sự cải thiện này xuất phát từ việc tập dữ liệu tùy chỉnh đã tăng cường các mẫu tín hiệu bất thường, giúp mô hình học được các đặc trưng đa dạng và phức tạp hơn, từ đó nâng cao hiệu quả tái lập tín hiệu ECG trên dữ liệu mới.

4.5 Thảo luận

Ý nghĩa của kết quả: Kết quả huấn luyện và đánh giá cho thấy sự kết hợp giữa 1D-CNN và Bi-LSTM là một hướng tiếp cận hiệu quả để tái lập tín hiệu ECG, đặc biệt khi sử dụng tập dữ liệu tùy chỉnh. Trên tập thử nghiệm chính, mô hình huấn luyện trên tập tùy chỉnh đạt MSE 0.0002, MAE 0.0108, RMSE 0.0140 và R^2 0.9952, vượt trội so với mô hình trên tập gốc (MSE 0.0007, MAE 0.0186, RMSE 0.0256, R^2 0.9839). Phân tích bổ sung trên bản ghi 214 cũng cung cấp nhận định này, với MSE giảm từ 0.0005 xuống 0.0002, MAE từ 0.0146 xuống 0.0086, RMSE từ 0.0220 xuống 0.0146 và R^2 tăng từ 0.9568 lên 0.9810. Các biểu đồ tái lập (Hình 4.15 và Hình 4.23) minh họa rõ ràng tín hiệu dự đoán từ mô hình trên tập tùy chỉnh bám sát tín hiệu thực tế hơn, đặc biệt ở các đoạn tín hiệu bất thường.

Sự cải thiện này chủ yếu xuất phát từ việc tập dữ liệu tùy chỉnh đã tăng cường tỷ lệ mẫu bất thường (như ngoại tâm thu hoặc nhịp nhanh), cho phép mô hình học được các đặc trưng phức tạp mà tập dữ liệu gốc MIT-BIH Arrhythmia thiếu hụt. Điều này phù hợp với nhận định rằng tín hiệu ECG dài hạn chứa nhiều biến thiên đòi hỏi khả năng trích xuất đặc trưng không gian (qua 1D-CNN) và phụ thuộc thời gian (qua Bi-LSTM). Kết quả không chỉ khẳng định hiệu quả của phương pháp đề xuất mà còn gợi mở tiềm năng ứng dụng trong việc tái tạo tín

hiệu ECG từ dữ liệu không đầy đủ hoặc hỗ trợ phân tích tín hiệu y khoa.

Hạn chế của nghiên cứu: Tuy nhiên, nghiên cứu vẫn tồn tại một số hạn chế. Quá trình tiền xử lý tín hiệu, bao gồm chuẩn hóa và lọc nhiễu, chưa được tối ưu hóa triệt để, có thể bỏ sót một số đặc trưng quan trọng của tín hiệu, đặc biệt ở các đoạn biến thiên mạnh. Ngoài ra, tập dữ liệu tùy chỉnh chủ yếu dựa trên việc cắt ghép từ bản ghi 102, dẫn đến khả năng thiếu sự đa dạng so với dữ liệu thực tế từ nhiều bệnh nhân. Cuối cùng, việc đánh giá mô hình chủ yếu dựa trên cơ sở dữ liệu MIT-BIH Arrhythmia, chưa được kiểm chứng trên dữ liệu thực tế từ thiết bị y tế hiện đại, làm giảm tính thuyết phục trong các ứng dụng lâm sàng.

Những hạn chế này không làm giảm giá trị của kết quả đạt được, nhưng cần được xem xét để đánh giá đầy đủ hiệu quả của mô hình và định hướng cải tiến trong tương lai.

CHƯƠNG 5 KẾT LUẬN

5.1 Kết quả đạt được

Nghiên cứu đã phát triển và đánh giá thành công mô hình kết hợp giữa 1D Convolutional Neural Networks (1D-CNN) và Bidirectional Long Short-Term Memory (Bi-LSTM) trong việc tái lập tín hiệu điện tâm đồ (ECG). Mô hình được huấn luyện trên hai tập dữ liệu: tập gốc từ cơ sở dữ liệu MIT-BIH Arrhythmia và tập tùy chỉnh được tối ưu hóa với tỷ lệ mẫu bất thường cao hơn. Kết quả cho thấy mô hình đạt hiệu suất vượt trội khi sử dụng tập dữ liệu tùy chỉnh so với tập dữ liệu gốc.

Trên tập thử nghiệm chính, mô hình huấn luyện trên tập tùy chỉnh ghi nhận các chỉ số hiệu suất ấn tượng: MSE 0.0002, MAE 0.0108, RMSE 0.0140 và R^2 0.9952, cải thiện đáng kể so với mô hình trên tập gốc (MSE 0.0007, MAE 0.0186, RMSE 0.0256, R^2 0.9839). Phân tích bổ sung trên bản ghi 214 (30,000 điểm dữ liệu từ MIT-BIH Arrhythmia) cũng khẳng định ưu thế này, với MSE giảm từ 0.0005 xuống 0.0002, MAE từ 0.0146 xuống 0.0086, RMSE từ 0.0220 xuống 0.0146 và R^2 tăng từ 0.9568 lên 0.9810. Các biểu đồ tái lập tín hiệu (Hình 4.15 và Hình 4.23) cho thấy mô hình trên tập tùy chỉnh tái hiện tín hiệu thực tế chính xác hơn, đặc biệt ở các đoạn tín hiệu bất thường.

Những kết quả này chứng minh rằng việc tăng cường dữ liệu bất thường trong tập tùy chỉnh không chỉ nâng cao độ chính xác của mô hình mà còn cải thiện khả năng khai quát hóa trên dữ liệu mới. Nghiên cứu đã đạt được mục tiêu đề ra: xây dựng một mô hình hiệu quả để tái lập tín hiệu ECG, mở ra tiềm năng ứng dụng trong phân tích tín hiệu y khoa.

5.2 Hướng phát triển trong tương lai

Dựa trên các kết quả và hạn chế của nghiên cứu, một số hướng phát triển tiếp theo được đề xuất. Thứ nhất, cần cải tiến quá trình tiền xử lý tín hiệu bằng các kỹ thuật tiên tiến như wavelet transform để giữ lại nhiều đặc trưng quan trọng hơn, thay vì chỉ dựa vào lọc nhiễu cơ bản. Thứ hai, mở rộng tập dữ liệu huấn luyện bằng cách thu thập thêm tín hiệu ECG từ nhiều nguồn hoặc áp dụng các phương pháp tăng cường dữ liệu tự động (data augmentation) nhằm tăng tính đa dạng và đại diện. Thứ ba, kiểm tra mô hình trên dữ liệu thực tế từ thiết bị y tế hiện đại hoặc tích hợp vào hệ thống giám sát thời gian thực để đánh giá hiệu suất trong môi trường lâm sàng. Cuối cùng, tối ưu hóa kiến trúc mô hình (ví dụ:

giảm số lượng tham số hoặc thử nghiệm các biến thể khác của CNN và LSTM) có thể nâng cao hiệu quả tính toán và khả năng triển khai thực tế.

Những hướng phát triển này hứa hẹn sẽ cải thiện độ chính xác và tính ứng dụng của mô hình, góp phần hỗ trợ các ứng dụng y khoa trong tương lai.

TÀI LIỆU THAM KHẢO

- [1] M. J. Hasan and J.-M. Kim, “A novel deep learning model for the detection and identification of rolling element-bearing faults,” *Sensors*, vol. 20, no. 18, p. 5112, 2020.
- [2] M. A. Ouali, M. Ghnai, and K. Chafaa, “Tlbo optimization algorithm based-type2 fuzzy adaptive filter for ecg signals denoising.,” *Traitement du Signal*, vol. 37, no. 4, 2020.
- [3] M. Hammad, A. Maher, K. Wang, F. Jiang, and M. Amrani, “Detection of abnormal heart conditions based on characteristics of ECG signals,” *Measurement*, vol. 125, pp. 634–644, 9 2018.
- [4] M. Ajam, A. A. Abu-Heija, M. Shokr, F. Ajam, and G. Saydain, “Sinus bradycardia and qt interval prolongation in west nile virus encephalitis: a case report,” *Cureus*, vol. 11, no. 1, 2019.
- [5] G. B. Moody and R. G. Mark, “The mit-bih arrhythmia database on cd-rom and software for use with it,” in *[1990] Proceedings Computers in Cardiology*, pp. 185–188, IEEE, 1990.
- [6] S. Mahapatra, D. Mohanta, P. Mohanty, S. kumar Nayak, and P. kumar Behari, “A neuro-fuzzy based model for analysis of an ecg signal using wavelet packet tree,” *Procedia Computer Science*, vol. 92, pp. 175–180, 2016.
- [7] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network,” *Nature medicine*, vol. 25, no. 1, pp. 65–69, 2019.
- [8] B. Hu, Y. Chen, and E. Keogh, “Time series classification under more realistic assumptions,” in *Proceedings of the 2013 SIAM international conference on data mining*, pp. 578–586, SIAM, 2013.
- [9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

- [10] M. Engin, “Ecg beat classification using neuro-fuzzy network,” *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1715–1722, 2004.
- [11] Q.-U.-A. Mastoi, T. Y. Wah, and R. Gopal Raj, “Reservoir computing based echo state networks for ventricular heart beat classification,” *Applied Sciences*, vol. 9, no. 4, p. 702, 2019.
- [12] F. Hadaeghi, “Reservoir computing models for patient-adaptable ecg monitoring in wearable devices,” *arXiv preprint arXiv:1907.09504*, 2019.
- [13] H. P. Martinez, Y. Bengio, and G. N. Yannakakis, “Learning deep physiological models of affect,” *IEEE computational intelligence magazine*, vol. 8, pp. 20–33, 5 2013.
- [14] Ö. Yıldırım, P. Pławiak, R.-S. Tan, and U. R. Acharya, “Arrhythmia detection using deep convolutional neural network with long duration ecg signals,” *Computers in biology and medicine*, vol. 102, pp. 411–420, 2018.
- [15] S. Kiranyaz, T. Ince, and M. Gabbouj, “Real-time patient-specific ecg classification by 1-d convolutional neural networks,” *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 2, pp. 664–675, 2016.
- [16] E. Alpaydin, *Machine learning*. MIT press, 2021.
- [17] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in *Proceedings of the 23rd international conference on Machine learning*, pp. 161–168, 2006.
- [18] M. E. Celebi and K. Aydin, *Unsupervised learning algorithms*, vol. 9. Springer, 2016.
- [19] X. Xu, L. Zuo, and Z. Huang, “Reinforcement learning algorithms with function approximation: Recent advances and applications,” *Information sciences*, vol. 261, pp. 1–31, 3 2014.
- [20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [22] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, J. Mietus, G. Moody, C. Peng, and H. Stanley, “Components of a new research resource for complex physiologic signals,” *PhysioBank, PhysioToolkit, and Physionet*, 2000.