

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CUỐI KÌ MÔN
DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

KIẾN TRÚC HỌC SÂU CHO BÀI TOÁN PHÂN LOẠI CHUỖI THỜI GIAN

Người hướng dẫn: ThS. Đỗ Thục Đoan

Người thực hiện: **ĐẶNG THANH TÚ – 51603354**

Lớp : **16050302**

Khoá : **20**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN
DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**KIẾN TRÚC HỌC SÂU
CHO BÀI TOÁN PHÂN LOẠI
CHUỖI THỜI GIAN**

Người hướng dẫn: **ThS. Đỗ Thục Đoan**

Người thực hiện: **ĐẶNG THANH TÚ – 51603354**

Lớp : 16050302

Khoá : 20

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

LỜI CẢM ƠN

Em xin chân thành cảm ơn cô Đỗ Thực Đoan trong suốt thời gian qua đã hướng dẫn em hoàn thành đồ án môn học này.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của ThS Đỗ Thục Đoan. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Đặng Thanh Tú

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Đồ án là phần tìm hiểu về cơ sở lý thuyết và thực nghiệm so sánh kết quả với bài báo gốc về hiệu suất của các thuật toán học sâu trong vấn đề phân loại chuỗi thời gian. Về cơ sở lý thuyết, báo cáo trình bày những khái niệm về mạng nơ ron sâu, các thành phần đặc trưng của mạng nơ ron sâu, những đặc điểm riêng của mạng nơ ron sâu dành cho việc phân loại chuỗi thời gian; kiến trúc học sâu hiệu quả để giải quyết bài toán phân loại chuỗi thời gian. Về phân thực nghiệm so sánh, báo cáo trình bày kết quả và kết luận về việc sử dụng kiến trúc Multilayer Perceptron, Fully Connected Neural Network và Residual Network để phân loại một số tập dữ liệu chuỗi thời gian phổ biến trong bộ dữ liệu chuỗi thời gian đơn biến UCR (2018), qua đó thấy được hiệu quả của các phương pháp này trong nhiệm vụ phân loại chuỗi thời gian.

MỤC LỤC

LỜI CẢM ƠN	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	iii
TÓM TẮT	iv
MỤC LỤC.....	1
DANH MỤC CÁC BẢNG.....	6
CHƯƠNG 1 – GIỚI THIỆU	7
1.1 Tổng quan	7
1.1.1 Dữ liệu chuỗi thời gian	7
1.1.2 Bài toán phân loại chuỗi thời gian	7
1.2 Cấu trúc báo cáo.....	9
CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT VÀ CÁC CÔNG TRÌNH LIÊN QUAN	10
2.1 Các mạng nơ ron sâu (Deep Neural Networks)	10
2.1.1 Mạng nơ ron tích chập (Convolution Neural Network)	10
2.1.1.1 Các trường tiếp nhận cục bộ	10
2.1.1.2 Chia sẻ trọng số và bias	12
2.1.1.3 Lớp tổng hợp.....	14
2.1.1.4 Cái nhìn tổng quát về mạng nơ ron tích chập	15
2.1.2 Một số mạng nơ ron sâu khác	16
2.1.2.1 Mạng thần kinh hồi quy (Recurrent Neural Network – RNNs)	16
2.1.2.2 Các đơn vị bộ nhớ ngắn/dài hạn (Long Short Term Memory - LSTMs).....	17
2.1.2.3 Mạng niềm tin sâu (Deep Belief Net – DBN), mô hình sản sinh và máy Boltzmann.....	17
2.2 Các thành phần học sâu sử dụng cho bài toán phân loại chuỗi thời gian	18
2.2.1 Tầng tích chập 1D (1D-Convolution Layer).....	18

2.2.2 Drop out	19
2.2.3 Batch-Normalize (BN).....	21
2.2.4 Pooling	22
2.2.4.1 Max Pooling.....	22
2.2.4.2 Global Average Pooling.....	22
2.2.5 Hàm kích hoạt	23
2.2.5.1 Hàm ReLU	23
2.2.5.2 Hàm Leaky ReLU	23
2.2.5.3 Hàm Softmax.....	24
2.3 Các hướng tiếp cận bài toán phân loại chuỗi thời gian.....	25
2.3.1 Dựa vào việc phát triển các hàm khoảng cách để đo độ tương tự ..	25
2.3.2 Dựa vào việc trích xuất các đặc trưng có ý nghĩa từ chuỗi thời gian	25
2.3.3 Các phương pháp tiếp cận bằng học sâu.....	26
2.3.3.1 Mô hình sản sinh (generative models).....	27
2.3.3.2 Mô hình phân biệt (discriminative)	28
2.4 Ba kiến trúc học sâu hiệu quả cho bài toán phân loại chuỗi thời gian.....	36
2.4.1 Multilayer Perceptron (MLP) ^[2]	36
2.4.2 Fully Convolutional Neural Networks (FCN) ^[2]	37
2.4.3 Residual Network (ResNet) ^[2]	38
CHƯƠNG 3 – THỰC NGHIỆM	39
1.1 Dữ liệu.....	39
1.2 Phương pháp đánh giá.....	45
2.3 Kết quả	46
2.3 Trực quan hóa và đánh giá kết quả	67
CHƯƠNG 4 – KẾT LUẬN.....	70
4.1 Tổng kết	70

4.2 Hướng phát triển	70
----------------------------	----

DANH MỤC CÁC HÌNH VẼ

Hình 1. Các nơ ron tương ứng với các pixel.....	10
Hình 2. Các nơ ron đầu vào kết nối với nơ ron ở tầng ẩn.....	11
Hình 3. Các nơ ron đầu vào kết nối với nơ ron ở tầng ẩn đầu tiên	11
Hình 4. Di chuyển trường tiếp nhận cục bộ	12
Hình 5. Bản đồ đặc trưng	13
Hình 6. Tập hợp nhiều bản đồ đặc trưng.....	13
Hình 7. Lớp tổng hợp	15
Hình 8. Lớp tổng hợp theo sau lớp tích chập	15
Hình 9. Mạng nơ ron tích chập	15
Hình 10. Tầng tích chập 1 chiều	18
Hình 11. Mạng nơ ron trước khi Drop out.....	19
Hình 12. Mạng nơ ron sau khi Drop out	20
Hình 13. Max – Pooling	22
Hình 14. Global Average Pooling	23
Hình 15. Hàm ReLU	23
Hình 16. Hàm Leaky ReLU	24
Hình 17. Học sâu áp dụng cho bài toán phân loại chuỗi thời gian	27
Hình 18. Encoder	30
Hình 19. Multi-Scale Convolutional Neural Networks	31
Hình 20. Time-LeNet.....	32
Hình 21. Multi-Channels Deep Convolutional Nerual Network	33
Hình 22. Time Convolutional Neural Network	34
Hình 23. Time Warping Invariant Echo State Network.....	36
Hình 24. Kiến trúc của MLP	36
Hình 25. Kiến trúc của FCN	37
Hình 26. Kiến trúc của ResNet	38

Hình 27. CAM của tập Coffee lớp 0.	69
Hình 28. CAM của tập Coffee lớp 1.	69

DANH MỤC CÁC BẢNG

Bảng 1. Mô tả thuộc tính dữ liệu thực nghiệm	45
Bảng 2. Mô tả các lớp mẫu dữ liệu thực nghiệm	45
Bảng 3. Mô tả kết quả thực nghiệm	47

CHƯƠNG 1 – GIỚI THIỆU

1.1 Tổng quan

1.1.1 Dữ liệu chuỗi thời gian

Chuỗi thời gian (Time Series) trong thống kê, xử lý tín hiệu, kinh tế lượng và toán tài chính là một chuỗi các điểm dữ liệu, được đo theo từng khoảng khắc thời gian liên nhau theo một tần suất thời gian thống nhất. Ví dụ như bản tin thời tiết, ghi chép tài chính, tín hiệu sinh lý hay các quan sát công nghiệp. Chuỗi thời gian thường được vẽ theo các đồ thị.

Một chuỗi thời gian đơn biến $X = [x_1; x_2; \dots; x_t]$ là một tập có thứ tự của các giá trị thực. Chiều dài của X bằng với số các giá trị thực t .

Một chuỗi thời gian đa chiều với M -chiều, $X = [X^1; X^2; \dots; X^M]$ gồm M chuỗi thời gian đơn biến khác nhau với $X^i \in \mathbf{R}^T$.

Một tập dữ liệu $D = \{(X^1; Y^1); (X^2; Y^2); \dots; (X^N; Y^N)\}$ là một tập hợp các cặp $(X^i; Y^i)$, trong đó X^i có thể là một chuỗi thời gian đơn biến hoặc đa biến, Y^i tương ứng với một vector nhãn của X^i .

Đối với một tập dữ liệu chứa các lớp K , vector nhãn one-hot Y^i là một vector có độ dài K mà với mỗi phần tử $j \in [1; K]$ bằng 1 nếu lớp của X^i là j và 0 nếu ngược lại.

1.1.2 Bài toán phân loại chuỗi thời gian

Phân tích chuỗi thời gian bao gồm các phương pháp để phân tích dữ liệu chuỗi thời gian, để từ đó trích xuất ra được các thuộc tính thống kê có ý nghĩa và các đặc điểm của dữ liệu.

Nhiệm vụ của một phân loại chuỗi thời gian (Time Series Classification - TSC) là huấn luyện một phân loại trên một tập dữ liệu D để ánh xạ chuỗi thời gian từ không gian đầu vào ra một phân bố xác suất trên các giá trị biến của các lớp đầu ra (nhãn).

Phân loại chuỗi thời gian là một bộ phận của phân tích chuỗi thời gian, được áp dụng rộng rãi trong các lĩnh vực khác nhau như để phân loại độ sáng của một ngôi sao trong thiên văn học, để chẩn đoán các rối loạn tim trong khoa học y tế hoặc để nhận ra các hoạt động của con người và để nhận dạng giọng nói trong khoa học máy tính. Tuy nhiên, bản chất của dữ liệu chuỗi thời gian, bao gồm kích thước lớn của dữ liệu, tính đa chiều và sự cập nhật liên tục của chuỗi thời gian, khiến việc phân loại chuỗi thời gian trở thành một nhiệm vụ khó khăn hơn.

Một trong những phương pháp tiếp cận truyền thống nhất là việc sử dụng một phân loại 1 - Nearest Neighbor (1 - NN) cùng với một hàm khoảng cách Dynamic Time Warping (DTW) mà ý tưởng chính là đo độ tương tự giữa chuỗi cần phân loại với những chuỗi đã biết trước đầu ra trong tập dữ liệu và trả về nhãn của chuỗi gần giống nhất^[8].

Về sau, ngày càng nhiều phương pháp tốt hơn 1-NN-DTW ra đời. Hầu hết các phương pháp đó có một đặc điểm chung, là giai đoạn chuyển đổi dữ liệu mà chuỗi thời gian được chuyển thành một không gian đặc trưng mới trước khi tiến hành phân loại. Giải pháp này bao gồm một tập hợp 35 bộ phân loại được đặt tên là COTE (Collective of Transformation-based Ensembles)^[9]. Phương pháp này đã được chứng minh rằng nếu tập hợp một cách đơn giản tất cả các phân loại trên vào một nhóm thì sẽ cải thiện độ chính xác đáng kể so với bất kỳ phân loại thành phần nào của nó và bất kỳ thuật toán TSC nào được công bố trước năm 2017. Không những thế, COTE còn được kết hợp mở rộng với một hệ thống phân cấp VOTE (một cấu trúc phân cấp mới có quyền biểu quyết xác suất) để trở thành HIVE-COTE, đã được chứng minh đạt được một sự cải thiện đáng kể so với COTE. HIVE-COTE hiện được coi là thuật toán đỉnh cao để phân loại chuỗi thời gian khi đánh giá qua 85 bộ dữ liệu từ kho lưu trữ UCR/UEA.

Tuy nhiên, để đạt được độ chính xác cao, HIVE-COTE trở nên cực kỳ tốn kém khi tính toán và không khả thi khi chạy trên một vấn đề khai thác dữ liệu thực tế lớn. Trong khi đó, học sâu (Deep Learning) đã cách mạng hóa lĩnh vực thị giác máy tính

(Computer Vision) khi đạt hiệu suất mức cạnh tranh với con người trong việc nhận dạng hình ảnh, cùng với đó là giải quyết tốt các vấn đề của xử lý ngôn ngữ tự nhiên (Natural Language Processing) như dịch máy, phân loại tài liệu, ... và trong lĩnh vực nhận dạng giọng nói, ... Để ý rằng, có sự tương đồng về khía cạnh liên tục dữ liệu giữa xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói với việc phân loại chuỗi thời gian. Tận dụng đặc điểm đó, Deep Learning hoàn toàn có thể ứng dụng vào giải quyết các bài toán phân loại chuỗi thời gian. Một trong những kiến trúc học sâu đạt được hiệu quả cao trong việc phân loại chuỗi thời gian đó là kiến trúc Residual Network (ResNet).

1.2 Cấu trúc báo cáo

Phần còn lại của báo cáo bao gồm 3 Chương:

- Chương 2: Cơ sở lý thuyết và các công trình liên quan. Chương này đề cập đến các kiến thức cơ bản của học máy và học sâu, các thành phần kiến trúc học sâu được sử dụng, các cách tiếp cận để giải quyết bài toán phân loại chuỗi thời gian cũng như những công trình liên quan đến giải quyết bài toán trên.
- Chương 3: Thực nghiệm. Các thực nghiệm được tiến hành để quan sát và đánh giá hiệu suất của ba kiến trúc học sâu được cho là hiệu quả nhất để giải quyết bài toán phân loại chuỗi thời gian dựa trên bộ các tập dữ liệu chuỗi thời gian đơn biến phổ biến nhất.
- Chương 4: Kết luận. Chương này đưa ra kết luận tổng quát về các cách tiếp cận để giải quyết bài toán phân loại chuỗi thời gian cũng như đề xuất một số hướng đi mới cải thiện hiệu suất của các kiến trúc hiện tại để giải quyết tốt hơn bài toán trên.

CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT VÀ CÁC CÔNG TRÌNH LIÊN QUAN

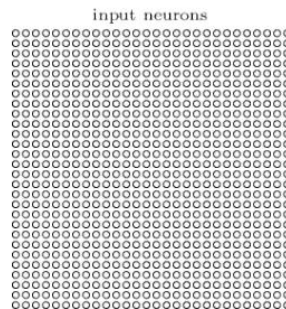
2.1 Các mạng nơ ron sâu (Deep Neural Networks)

2.1.1 Mạng nơ ron tích chập (Convolution Neural Network)

Các mạng nơ ron tích chập sử dụng một kiến trúc giúp chúng ta huấn luyện với mạng nhiều lớp, thời gian huấn luyện nhanh hơn, đặc biệt rất thích hợp với phân loại hình ảnh. Ngày nay, mạng nơ ron tích chập hoặc một số biến thể gần đây được sử dụng trong hầu hết các mạng thần kinh để nhận dạng hình ảnh. Mạng nơ-ron tích chập sử dụng ba ý tưởng cơ bản: các trường tiếp nhận cục bộ, các trọng số chia sẻ, và các phép tổng hợp.

2.1.1.1 Các trường tiếp nhận cục bộ

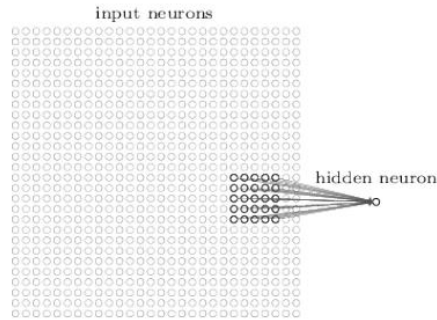
Trong các lớp kết nối đầy đủ trước đây, đầu vào được mô tả như một đường thẳng đứng của các nơ ron. Trong một mạng nơ ron tích chập, đầu vào lại là một hình vuông của các nơ ron. Ví dụ: hình vuông có kích thước 28×28 của các nơ ron sẽ có giá trị tương ứng với 28×28 pixel:



Hình 1. Các nơ ron tương ứng với các pixel

Như thường lệ, chúng ta sẽ kết nối các điểm ảnh đầu vào với một lớp nơ ron ẩn (hidden layer). Nhưng chúng ta sẽ không kết nối mỗi điểm ảnh đầu vào với mỗi neuron ẩn. Thay vào đó, chúng ta chỉ kết nối mỗi neuron ẩn với khu vực nhỏ của ảnh đầu vào.

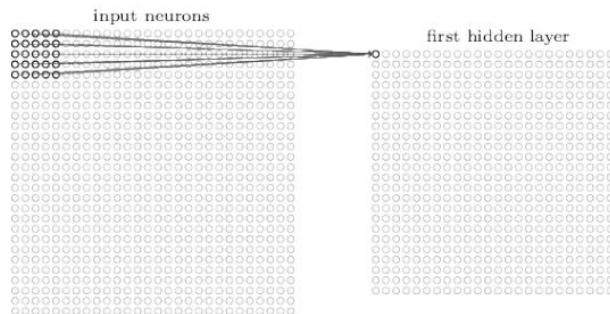
Ví dụ, với một khu vực 5×5 , tương ứng với 25 pixel đầu vào, với một nơ ron ẩn, chúng ta có thể có các kết nối trông như sau:



Hình 2. Các nơ ron đầu vào kết nối với nơ ron ở tầng ẩn

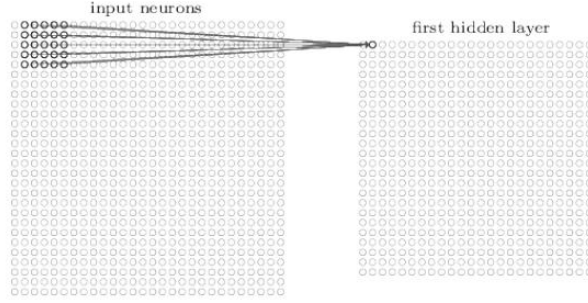
Khu vực kết nối này được gọi là trường tiếp nhận cục bộ cho các neuron ẩn. Nó như một cửa sổ nhỏ mà trên đó là một số điểm ảnh đầu vào. Mỗi kết nối học một trọng số (weight). Và neuron ẩn có một thành kiến (bias) tổng quát cho mỗi trường tiếp nhận. Chúng ta có thể hiểu rằng neuron ẩn đang học tập để phân tích các trường tiếp nhận cục bộ này.

Để kết nối cho những nơ ron ẩn tiếp theo, chúng ta trượt các trường tiếp nhận cục bộ trên toàn bộ hình ảnh đầu vào. Cụ thể đối với mỗi các trường tiếp nhận cục bộ, có một neuron ẩn khác nhau trong lớp đầu tiên ẩn:



Hình 3. Các nơ ron đầu vào kết nối với nơ ron ở tầng ẩn đầu tiên

Sau đó, chúng ta trượt trường tiếp nhận cục bộ trên qua bên phải một pixel, để kết nối cho một nơ ron ẩn thứ hai:



Hình 4. Di chuyển trường tiếp nhận cục bộ

Và cứ như vậy, chúng ta xây dựng nên lớp đầu tiên ẩn. Chúng ta chỉ dừng di chuyển trước khi va chạm với biên ảnh phía bên tay phải (hoặc dưới) của hình ảnh đầu vào. Trong ví dụ trên, trường tiếp nhận cục bộ được di chuyển một pixel cùng một lúc. Trong thực tế, đôi khi chiều dài bước trượt được lấy giá trị khác (2,3,4,...).

2.1.1.2 Chia sẻ trọng số và bias

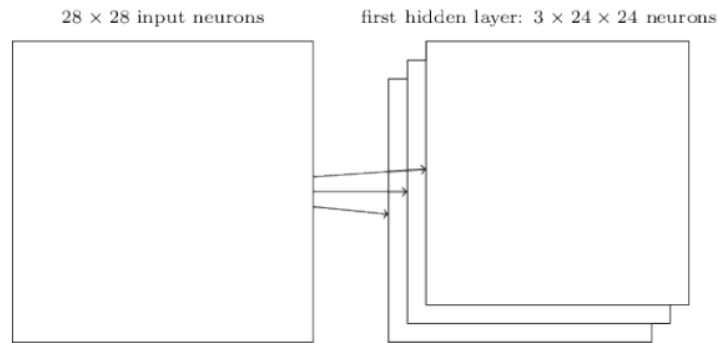
Chúng ta sẽ sử dụng chung các trọng số và bias cho mỗi kết nối từ trường tiếp nhận cục bộ đến một nơon ẩn. Nói cách khác, đối với những neuron ẩn thứ j, k , đầu ra sẽ là:

$$\sigma(b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l,k+m}) \quad (2.1)$$

Trong đó, σ là hàm kích hoạt của nơ ron, b là giá trị chung của bias. $w_{l,m}$ là một mảng 5×5 của trọng số được chia sẻ và $a_{x,y}$ biểu thị đầu vào tại vị trí x, y .

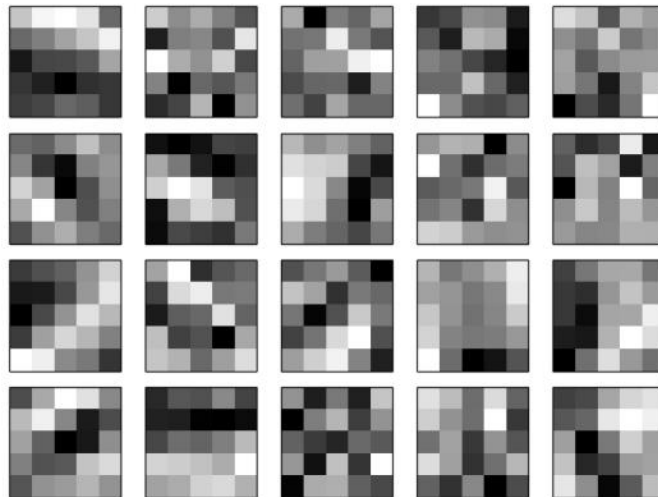
Điều này có nghĩa rằng tất cả các nơ ron trong lớp ẩn đầu tiên sẽ phát hiện chính xác các đặc trưng giống nhau, tại các vị trí khác nhau trong hình ảnh đầu vào. Đặc trưng đó có thể là một cạnh, một góc,... Chúng ta đôi khi gọi bản đồ từ lớp đầu vào cho lớp ẩn là bản đồ đặc trưng. Chúng ta gọi các trọng số dùng để xác định cho các bản đồ đặc trưng là các trọng số chia sẻ. Và tương tự các bias là các bias chia sẻ. Các trọng số và bias được chia sẻ thường được gọi là một kernel hoặc một bộ lọc.

Một bản đồ đặc trưng chỉ có thể phát hiện một loại đặc trưng cục bộ. Để nhận dạng hình ảnh, chúng ta sẽ cần nhiều hơn một bản đồ đặc trưng. Và do đó, chúng ta cần một lớp tích chập hoàn chỉnh, bao gồm các bản đồ đặc trưng khác nhau giống như sau:



Hình 5. Bản đồ đặc trưng

Với 3 bản đồ đặc trưng như trên, mạng có thể phát hiện 3 loại đặc trưng khác nhau, với mỗi đặc trưng có thể được phát hiện trên toàn bộ hình ảnh. Trong thực tế mạng tích chập có thể sử dụng số lượng bản đồ đặc trưng nhiều hơn nữa. Chúng ta hãy xem nhanh một lớp tích chập với 20 bản đồ đặc trưng.



Hình 6. Tập hợp nhiều bản đồ đặc trưng

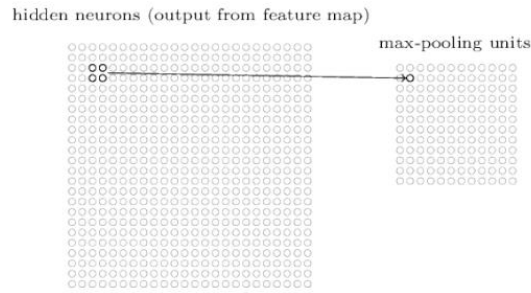
Trong đó, các ô trắng có nghĩa là nó chứa một trọng số nhỏ hơn, do kernel phản ứng ít với pixel đầu vào tương ứng, các ô đen có nghĩa là nó chứa một trọng số lớn hơn, do kernel phản ứng nhiều hơn với pixel đầu vào tương ứng. Nói tóm lại, những hình ảnh trên cho thấy các loại đặc trưng mà lớp tích chập đã phát hiện (phản ứng) ra.

Một lợi thế lớn của trọng lượng chia sẻ và những bias là nó làm giảm đáng kể số lượng các tham số liên quan đến một mạng lưới tích chập. Ví dụ, đối với trường hợp trên chúng ta cần 26 tham số ($5 \times 5 = 25$ trọng số chia sẻ, cộng với 1 bias chia sẻ duy nhất), và với 20 bản đồ đặc trưng tổng cộng chỉ cần $20 \times 26 = 520$ tham số cho lớp tích chập. Trong khi với một lớp kết nối đầy đủ đầu tiên, có $28 \times 28 = 784$ nơon đầu vào, và 30 nơon ẩn, nên sẽ có 784×30 trọng số, cộng thêm 30 bias, tổng cộng có tất cả 23.550 tham số. Nói cách khác, lớp kết nối đầy đủ sẽ có nhiều hơn 40 lần so với số tham số của lớp tích chập. Dễ thấy rằng, hiệu suất tương tự như nhau, các lớp tích chập sẽ cho kết quả huấn luyện nhanh hơn so với các lớp kết nối đầy đủ.

Thực tế, tên lớp tích chập xuất phát từ việc hàm kích hoạt của nơon thuộc lớp này có thể viết dưới dạng một phép toán tích chập.

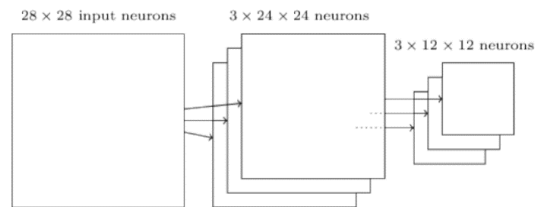
2.1.1.3 Lớp tổng hợp

Ngoài các lớp tích chập, mạng nơon tích chập cũng chứa các lớp tổng hợp. Lớp tổng hợp thường được sử dụng ngay sau lớp tích chập. Nhiệm vụ của lớp tổng hợp là đơn giản hóa các thông tin trong đầu ra của lớp tích chập. Cụ thể, một lớp tổng hợp chuyển đổi mỗi bản đồ đặc trưng đầu ra từ lớp tích chập thành một bản đồ đặc trưng tinh gọn hơn. Ví dụ, mỗi đơn vị trong lớp tổng hợp sau đây có thể thu gọn một vùng 2×2 trong lớp trước đó thành giá trị lớn nhất của vùng đó. Lớp tổng hợp này được gọi là max-pooling:



Hình 7. Lớp tổng hợp

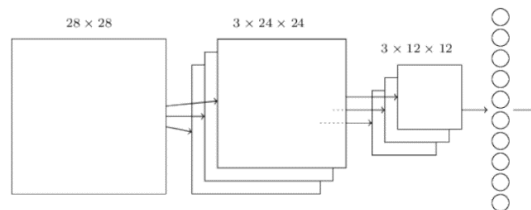
Và nếu chúng ta có 3 bản đồ đặc trưng, các lớp tích chập và lớp tổng hợp sẽ trông như sau:



Hình 8. Lớp tổng hợp theo sau lớp tích chập

Lớp tổng hợp như là một cách để xem một đặc trưng nhất định có nằm trong một khu vực của hình ảnh hay không. Một lợi ích lớn của lớp tổng hợp là có nhiều đặc trưng gộp ít hơn, và vì vậy sẽ giúp giảm số lượng các tham số cần thiết trong các lớp sau. Hàm Max không phải là kỹ thuật duy nhất được sử dụng cho lớp tổng hợp. Một phương pháp phổ biến được gọi là tổng hợp $L2$, thay vì dùng hàm Max, nó lấy căn bậc hai tổng các bình phương của kích hoạt ở trong khu vực 2×2 đang xét. Trên thực tế, cả hai kỹ thuật này đều được sử dụng rộng rãi.

2.1.1.4 Cái nhìn tổng quát về mạng nơ ron tích chập



Hình 9. Mạng nơ ron tích chập

Trong mạng nơ ron trên, chúng ta bắt đầu với 28×28 nơron đầu vào, được sử dụng để mã hóa các cường độ pixel cho một hình ảnh. Sau đó là một lớp tích chập sử dụng trường tiếp nhận cục bộ có kích thước 5×5 và 3 bản đồ đặc trưng. Đầu ra của lớp tích chập là một lớp $3 \times 24 \times 24$ nơron đặc trưng ẩn. Tiếp theo là một lớp max-pooling, áp dụng cho mỗi khu vực 2×2 trên một trong 3 bản đồ đặc trưng. Đầu ra của lớp tổng hợp này là một lớp $3 \times 12 \times 12$ nơron đặc trưng ẩn. Lớp cuối cùng của các kết nối trong mạng là một lớp kết nối đầy đủ. Lớp này kết nối tất cả các nơ ron từ đầu ra của lớp max-pooling với mỗi một nơ ron trong 10 nơron đầu ra.

Cũng giống như với mạng nơ ron thông thường, chúng ta sẽ huấn luyện mạng nơ ron tích chập bằng stochastic gradient descent và lan truyền ngược (back propagation). Tuy nhiên, chúng ta cần phải thực hiện một vài sửa đổi thủ tục lan truyền ngược.

2.1.2 Một số mạng nơ ron sâu khác

2.1.2.1 Mạng thần kinh hồi quy (Recurrent Neural Network – RNNs)

Trong các mạng lưới mà chúng ta đã đề cập, chỉ có một đầu vào duy nhất dùng để xác định kích hoạt cho tất cả các nơ ron trong mạng thông qua các lớp còn lại. Tất cả mọi thứ trong mạng là cố định. Nhưng giả sử chúng ta cho phép các yếu tố trong mạng thay đổi theo một cách năng động. Ví dụ, có thể hành vi của nơ ron tiềm ẩn không chỉ được quyết định bằng kích hoạt trong lớp ẩn trước, mà còn bằng kích hoạt vào những thời điểm trước đó. Trên thực tế, kích hoạt của nơ ron có thể được xác định một phần bằng kích hoạt riêng của nó vào một thời điểm trước đó. Hoặc có thể kích hoạt các nơ ron ẩn và đầu ra sẽ không được xác định chỉ bằng đầu vào hiện nay vào mạng, mà còn bởi nguyên liệu đầu vào trước đó.

Mạng nơ-ron có một loại hành vi về thời gian khác nhau được gọi là mạng thần kinh hồi quy hoặc RNNs. Nó đặc biệt hữu ích trong việc phân tích dữ liệu hoặc các quá trình thay đổi theo thời gian. Ví dụ, trong các vấn đề về dữ liệu và quy trình phát sinh tự nhiên như bài phát biểu hoặc ngôn ngữ tự nhiên.

RNNs cũng đã được sử dụng trong những năm gần đây để giải quyết nhiều vấn đề khác. Nó đã được đặc biệt hữu ích trong việc nhận dạng giọng nói. Phương pháp tiếp cận dựa trên RNNs đã có ghi nhận về tính chính xác trong nhận dạng âm thanh (Sound Recognition). Nó cũng đã được sử dụng để phát triển mô hình ngôn ngữ cải tiến mà chúng ta sử dụng trong khi nói chuyện. Mô hình ngôn ngữ tốt hơn giúp định hướng lời nói chính xác kể cả khi âm thanh của chúng giống nhau. RNNs đã thiết lập nhiều kỷ lục mới cho các tiêu chuẩn ngôn ngữ nhất định.

2.1.2.2 Các đơn vị bộ nhớ ngắn/dài hạn (Long Short Term Memory - LSTMs)

Một thách thức ảnh hưởng đến RNNs là mô hình rất khó khăn để huấn luyện, khó hơn cả mạng chuyển tiếp sâu (Deep Feedforward Networks). Nguyên do là bởi vấn đề đạo hàm không ổn định, với các biểu hiện thường thấy là đạo hàm ngày càng nhỏ hơn khi nó được truyền lại qua các lớp. Điều này làm cho việc học ở các lớp đầu cực kỳ chậm. Vấn đề thực sự trở nên tồi tệ hơn trong RNNs, vì đạo hàm không chỉ truyền ngược qua lớp mà còn truyền ngược qua thời gian. Nếu mạng chạy trong một thời gian dài có thể làm cho đạo hàm cực kỳ không ổn định và khó có thể học. May mắn là nó có thể kết hợp một ý tưởng được gọi là các đơn vị bộ nhớ ngắn/dài hạn (LSTMs) vào RNNs. Các đơn vị giúp giải quyết vấn đề đạo hàm không ổn định. LSTMs giúp cho việc huấn luyện RNNs có được kết quả tốt trở nên dễ dàng hơn.

2.1.2.3 Mạng niềm tin sâu (Deep Belief Net – DBN), mô hình sản sinh và máy Boltzmann

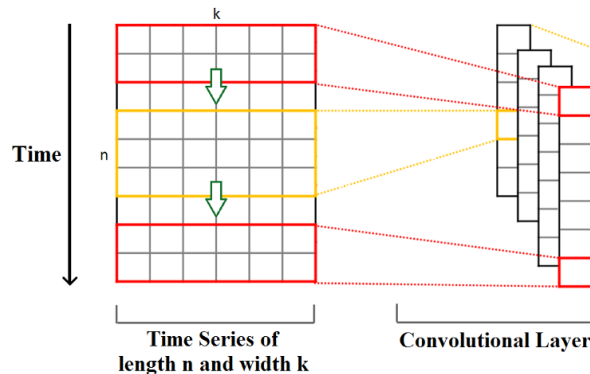
DBNs đã có ảnh hưởng trong nhiều năm, nhưng nó bắt đầu giảm bớt phổ biến từ khi các mô hình như các mạng chuyển tiếp sâu và mạng nơ ron hồi quy trở nên thịnh hành. Mặc dù vậy, DBNs vẫn có một vài đặc trưng thú vị. Một trong số đó là mô hình sản sinh. Trong một mạng chuyển tiếp sâu, chúng ta xác định kích hoạt đầu vào, và mạng xác định kích hoạt các đặc trưng của các nơ ron phía sau trong mạng. Một mô hình sản sinh như DBN có thể được sử dụng theo cách tương tự, đồng thời nó cũng có thể “chạy ngược vào trong mạng”, tạo ra giá trị cho các kích hoạt đầu vào. Ví dụ, một

DBN được huấn luyện cho phân loại hình ảnh của các chữ số viết tay cũng có thể được sử dụng để tạo ra hình ảnh trông giống như chữ số viết tay. Nói cách khác, các DBN có thể sẽ được học để viết. Khi đó, một mô hình sản sinh thực sự giống với bộ não của con người: không chỉ nó có thể đọc chữ số, nó cũng có thể viết chữ số. Lý do thứ hai là do nó có thể học không giám sát (Unsupervised Learning) và học bán giám sát (Semi-Supervised Learning). Ví dụ, khi được đào tạo với các dữ liệu hình ảnh, DBNs có thể học được các đặc trưng hữu ích cho cả những hình ảnh khác, ngay cả khi những hình ảnh huấn luyện này không được dán nhãn.

2.2 Các thành phần học sâu sử dụng cho bài toán phân loại chuỗi thời gian

2.2.1 Tầng tích chập 1D (1D-Convolution Layer)

Cũng như những tầng tích chập khác trong các mạng nơ ron dùng cho dữ liệu là hình ảnh, tầng tích chập trong mạng ResNet dùng để rút trích các đặc trưng của dữ liệu đầu vào và làm giảm đáng kể số lượng tham số mà mạng cần phải học. Tuy nhiên, điểm khác biệt là thay vì dùng các bộ lọc 2D hoặc 3D cho dữ liệu hình ảnh, mạng ResNet sử dụng các bộ lọc 1D cho dữ liệu là chuỗi thời gian.



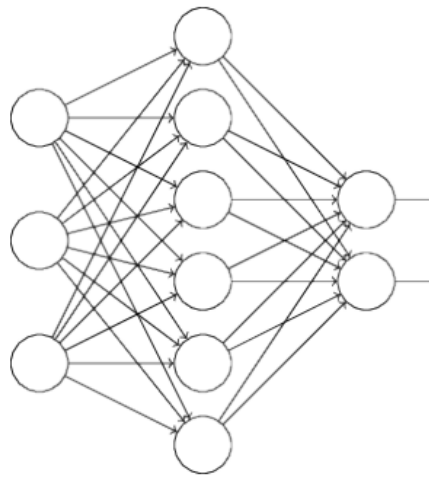
Hình 10. Tầng tích chập 1 chiều

Trên là ví dụ một chuỗi thời gian có chiều dài n và chiều rộng k . Chiều dài n là số lượng bước thời gian và chiều rộng k là số lượng biến (nếu là chuỗi thời gian đa biến). Các kernel tích chập luôn có cùng chiều rộng với chuỗi thời gian, trong khi chiều dài của chúng có thể thay đổi. Theo cách này, kernel di chuyển theo một hướng từ đầu

đến cuối chuỗi thời gian để thực hiện phép chập. Nó không di chuyển sang trái hoặc phải như khi thực hiện phép tích chập 2 chiều thông thường cho hình ảnh. Trong quá trình thực hiện tích chập, các phần tử của kernel được nhân với các phần tử tương ứng của chuỗi thời gian mà chúng bao phủ tại một điểm nhất định. Sau đó, kết quả của phép nhân được cộng với nhau và áp dụng hàm kích hoạt phi tuyến cho giá trị nhận được. Kết quả của hàm kích hoạt trở thành một thành phần của chuỗi thời gian đơn biến mới (đã được lọc bởi bộ lọc trước đó) và sau đó kernel tiếp tục di chuyển dọc theo chuỗi thời gian để tạo ra giá trị tiếp theo. Tùy thuộc vào độ dài của kernel, các khía cạnh, thuộc tính khác nhau, các đặc trưng của chuỗi thời gian ban đầu được ghi lại trong mỗi chuỗi thời gian mới đã được lọc.

2.2.2 Drop out

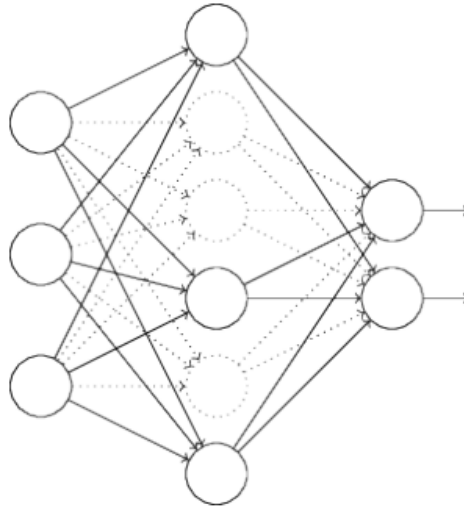
Dropout là một kỹ thuật hoàn toàn khác so với chính quy hóa. Khác với chính quy hóa L_1 và L_2 , dropout không dựa vào sửa đổi hàm chi phí. Thay vào đó, dropout tự sửa đổi các mạng riêng của nó.



Hình 11. Mạng nơ ron trước khi Drop out

Bình thường, chúng ta có một huấn luyện đầu vào x và tương ứng với mong muốn đầu ra y . Chúng ta huấn luyện lan truyền tiến x xuyên qua mạng, và sau đó lan truyền ngược để xác định sự đóng góp vào đạo hàm của các trọng số. Với dropout, quá

trình này được sửa đổi. Chúng ta bắt đầu bằng cách xóa (ngẫu nhiên và tạm thời) một nửa các nơron ẩn trong mạng, ngoại trừ nơron đầu vào và đầu ra. Điều này tạo ra một mạng đã được sửa đổi tạm thời mà chúng ta sẽ sử dụng để huấn luyện



Hình 12. Mạng nơ ron sau khi Drop out

Chúng ta thực hiện lan truyền tiến với đầu vào x thông qua mạng lưới đã được sửa đổi tạm thời, và lan truyền ngược với kết quả nhận được, cũng thông qua mạng lưới sửa đổi này. Sau khi thực hiện trên một mini-batch, chúng ta cập nhật các trọng số và các bias thích hợp. Sau đó chúng ta lặp lại quá trình này, đầu tiên khôi phục lại các nơ ron được xóa tạm thời trước đây, sau đó chọn một tập hợp ngẫu nhiên mới nơ ron ẩn khác để xóa, tính đạo hàm trên một mini-batch khác, và lại cập nhật các trọng số và các bias trong mạng. Bằng cách lặp đi lặp lại quá trình này, mạng của chúng ta sẽ được học một bộ trọng số và bias.

Khi chúng ta dropout mạng thành những bộ nơ ron khác nhau, nó khá giống với việc chúng ta đang huấn luyện những mạng lưới nơ ron khác nhau. Và như vậy dropout cũng giống như việc ta lấy trung bình ảnh hưởng của một số lượng rất lớn của các mạng khác nhau. Các mạng này sẽ overfit theo những cách khác nhau. Do đó, hy vọng với dropout chúng ta sẽ giảm được hiện tượng quá khớp.

2.2.3 Batch-Normalize (BN)

Khi huấn luyện mạng nơ ron nhân tạo, chúng ta thường gặp hiện tượng là dữ liệu không phân bố quanh giá trị 0, mà phần nhiều dữ liệu có giá trị lớn hơn không, hoặc nhỏ hơn không. Đây gọi là hiện tượng trung bình khác không (non-zero mean). Kết hợp với vấn đề độ lệch cao (high variance) khiến dữ liệu trở nên có nhiều thành phần rất lớn hoặc rất nhỏ. Vấn đề này rất phổ biến khi huấn luyện các mạng nơ ron với số tầng sâu. Việc các đặc trưng đầu vào không phân phối trong những khoảng ổn định (giá trị to nhỏ thất thường) sẽ có ảnh hưởng đến quá trình tối ưu của mạng.

Batch Normalization là một phương pháp hiệu quả khi huấn luyện một mô hình mạng nơ ron. Mục tiêu của phương pháp này chính là việc muốn chuẩn hóa các đặc trưng (feature, cũng là đầu ra của mỗi tầng sau khi đi qua các kích hoạt) về trạng thái zero-mean với độ lệch chuẩn (standard deviation) bằng 1^[3]. Điều này giúp chúng ta tránh hiện tượng non-zero đã đề cập ở trên.

Với đầu vào là x thuộc một mini-batch $\mathcal{B} = \{x_{1...m}\}$ và các tham số học γ, β thì đầu ra chuẩn hóa $\{y_i = BN_{\gamma, \beta}(x_i)\}$ được tính tuần tự theo các công thức sau:

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad (2.2)$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad (2.3)$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad (2.4)$$

$$y_i = \gamma \hat{x}_i + \beta \equiv BN_{\gamma, \beta}(x_i) \quad (2.5)$$

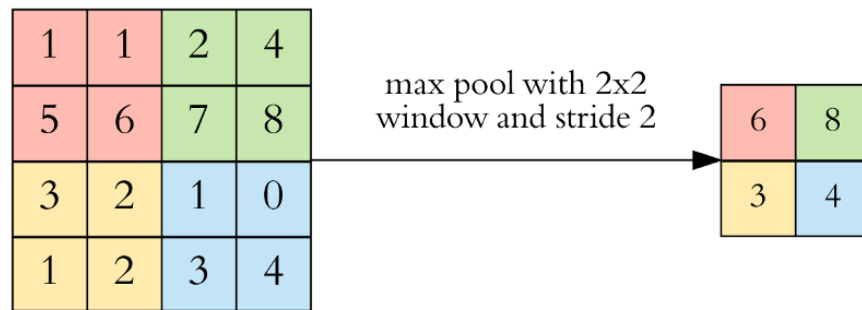
Batch normalization còn có thể giúp chúng ta tránh được hiện tượng giá trị của x rơi vào khoảng bão hòa sau khi đi qua các hàm kích hoạt phi tuyến. Vậy nên nó đảm bảo rằng không có sự kích hoạt nào bị vượt quá cao hoặc quá thấp. Điều này giúp cho các weights mà khi không dùng BN có thể sẽ không bao giờ được học thì nay lại được học bình thường, giúp chúng ta làm giảm đi sự phụ thuộc vào giá trị khởi tạo của các tham số.

Batch Normalization còn có vai trò như một dạng của regularization giúp cho việc giảm thiểu hiện tượng quá khớp (overfitting). Sử dụng batch normalization, chúng ta sẽ không cần phải sử dụng quá nhiều dropout, giúp tránh mất quá nhiều thông tin khi dropout tham số của mạng.

2.2.4 Pooling

2.2.4.1 Max Pooling

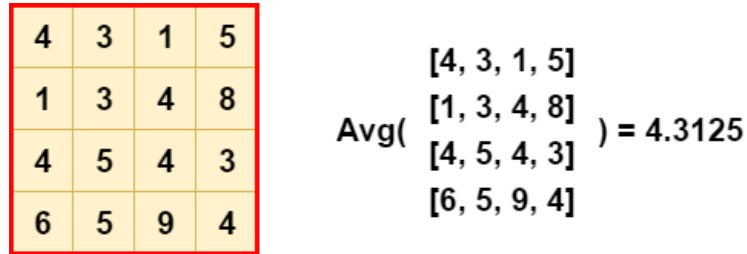
Max Pooling là một phép tính dùng để tính đầu ra lớn nhất của từng bản đồ đặc trưng (feature map) trong lớp trước. Nó làm giảm đáng kể dữ liệu và chuẩn bị cho lớp phân loại cuối cùng.



Hình 13. Max – Pooling

2.2.4.2 Global Average Pooling

Global Average Pooling (GAP) là một phép tính dùng để tính đầu ra trung bình của từng bản đồ đặc trưng (feature map) trong lớp trước^[4]. Nó cũng làm giảm đáng kể dữ liệu và chuẩn bị cho lớp phân loại cuối cùng. Tương tự như Max Pooling, nó cũng không có thông số huấn luyện. Tuy nhiên thay vì lấy giá trị lớn nhất, GAP tính toán và trả về giá trị trung bình trên một bản đồ đặc trưng.



Hình 14. Global Average Pooling

2.2.5 Hàm kích hoạt

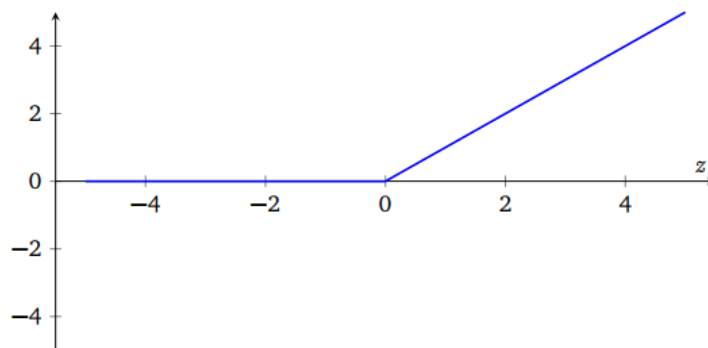
2.2.5.1 Hàm ReLU

Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng neuron. ReLU đơn giản lọc các giá trị < 0 . Nhìn vào công thức chúng ta dễ dàng hiểu được cách hoạt động của nó. Một số ưu điểm khá vượt trội của nó so với Sigmoid và Tanh:

Tốc độ hội tụ nhanh hơn hẳn. ReLU có tốc độ hội tụ nhanh gấp 6 lần hàm Tanh. Điều này có thể do ReLU không bị bão hoà ở 2 đầu như Sigmoid và Tanh.

Tính toán nhanh hơn. Do ReLU có công thức tính đơn giản hơn Tanh và Sigmoid (sử dụng hàm mũ và công thức phức tạp hơn)^[5].

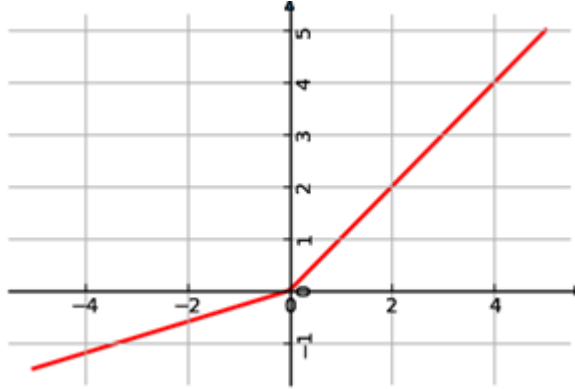
$$\max(0, w \cdot x + b)$$



Hình 15. Hàm ReLU

2.2.5.2 Hàm Leaky ReLU

$$f(x) = 1(x < 0) (\alpha x) + 1(x \geq 0) (x) \text{ với } \alpha \text{ là hằng số nhỏ.}$$



Hình 16. Hàm Leaky ReLU

Leaky ReLU là một cố gắng trong việc loại bỏ “dying ReLU”. Thay vì trả về giá trị 0 với các đầu vào < 0 thì Leaky ReLU tạo ra một đường xiên có độ dốc nhỏ (như đồ thị trên). Có nhiều báo cáo về việc hiệu Leaky ReLU có hiệu quả tốt hơn ReLU, nhưng hiệu quả này vẫn chưa rõ ràng và nhất quán.

Ngoài Leaky ReLU có một biến thể cũng khá nổi tiếng của ReLU là PReLU. PReLU tương tự Leaky ReLU nhưng cho phép nơ ron tự động chọn hệ số α tốt nhất.

2.2.5.3 Hàm Softmax

Để biến đổi kết quả đầu ra từ các tầng trước thành kết quả không âm và có tổng là 1 ở tầng đầu ra cuối cùng, trong khi vẫn giữ tính chất khả vi, đầu tiên ta cần lấy hàm mũ logarit cho từng kích hoạt ở tầng trước đó (để chắc chắn chúng không âm) và sau đó chia cho tổng của chúng (để chắc rằng tổng của chúng luôn bằng 1). Đó là những gì hàm softmax thực hiện để đảm bảo đầu ra cuối cùng là một phân phối xác suất trên các nhãn phân loại. Với a_i là các đầu ra tương ứng của tầng trước đó, xác suất cho mỗi nhãn phân loại được tính như sau:

$$f_i(\vec{a}) = \frac{e^{a_i}}{\sum_k e^{a_k}} \quad (2.6)$$

2.3 Các hướng tiếp cận bài toán phân loại chuỗi thời gian

Căn cứ vào nhu cầu phân loại chính xác dữ liệu chuỗi thời gian, các nhà nghiên cứu đã đề xuất hàng trăm phương pháp để giải quyết nhiệm vụ này. Tuy nhiên, chúng có thể được chia làm 3 nhóm sau:

2.3.1 Dựa vào việc phát triển các hàm khoảng cách để đo độ tương tự

Một trong những phương pháp tiếp cận truyền thống nhất là việc sử dụng một phân loại 1 - Nearest Neighbor (1 - NN) cùng với một hàm khoảng cách Dynamic Time Warping (DTW) mà ý tưởng chính là đo độ tương tự giữa chuỗi cần phân loại với những chuỗi đã biết trước đầu ra trong tập dữ liệu và trả về nhãn của chuỗi gần giống nhất.

2.3.2 Dựa vào việc trích xuất các đặc trưng có ý nghĩa từ chuỗi thời gian

Về sau, ngày càng nhiều phương pháp tốt hơn 1-NN-DTW ra đời. Hầu hết các phương pháp đó có một đặc điểm chung, là giai đoạn chuyển đổi dữ liệu mà chuỗi thời gian được chuyển thành một không gian đặc trưng mới trước khi tiến hành phân loại. Giải pháp này bao gồm một tập hợp 35 bộ phân loại được đặt tên là COTE (Collective of Transformation-based Ensembles). Các bộ phân loại này được hình thành dựa trên các phép biến đổi dữ liệu khác nhau trên các miền khác nhau nhằm giúp cải thiện độ chính xác của phân loại chuỗi thời gian. Các bộ phân loại này được xây dựng trong các miền thời gian (Time Domain), miền tần số (Frequency Domain), miền thay đổi (Change Domain) và miền biến đổi hình học (Shapelet Transformation Domain). Đối với miền thời gian, bộ phân loại này bao gồm các bộ phân loại thuộc tập hợp linh hoạt (Elastic Ensemble) như 1-NN sử dụng khoảng cách Euclidean (1-NN with Euclidean Distance), DTW đầy đủ, DTW với kích thước cửa sổ được thiết lập thông qua xác thực chéo (Dynamic Time Warping with window size set through Cross Validation), DTW phát sinh với cửa sổ đầy đủ (Derivative DTW with full window), DTW phát sinh với cửa sổ được thiết lập thông qua xác thực chéo (Derivative DTW with window set through Cross Validation), DTW có trọng số (weighted DTW), DTW có trọng số phát

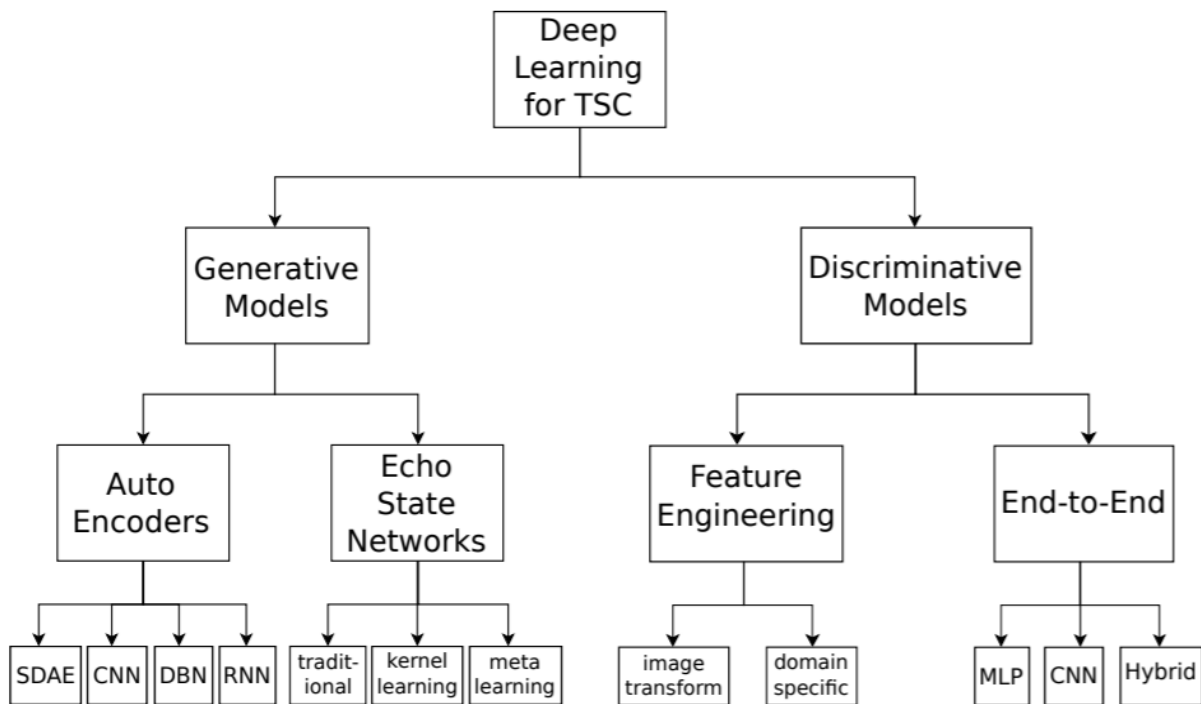
sinh (Derivative Weighted DTW), chuỗi chung dài nhất (Longest Common Subsequence), khoảng cách chỉnh sửa với hình phạt thực (Edit Distance with Real Penalty), khoảng cách chỉnh sửa với chênh lệch thời gian (Time Warp Edit Distance), và số liệu khoảng cách Move-Split-Merge (the Move-Split-Merge Distance Metric). Đối với các miền khác, bộ phân loại này bao gồm các bộ phân loại thuộc tập hợp không đồng nhất (Heterogeneous Ensemble) như k-NN với k được thiết lập thông qua xác thực chéo (k - Nearest Neighbour where k is set through cross validation), Naive Bayes, cây quyết định C4.5 (C4.5 Decision Tree), máy vector hỗ trợ với kernel hàm tuyến tính (Support Vector Machines with linear function kernels), máy vector hỗ trợ với kernel hàm bậc hai (Support Vector Machines with quadratic basis function kernels), Rừng ngẫu nhiên với 100 cây (Random Forest with 100 trees), Rừng luân chuyển với 10 cây (Rotation Forest with 10 trees) và mạng Bayes (Bayesian Network). Phương pháp này đã được chứng minh rằng nếu tập hợp một cách đơn giản tất cả các phân loại trên vào một nhóm thì sẽ cải thiện độ chính xác đáng kể so với bất kỳ phân loại thành phần nào của nó và bất kỳ thuật toán TSC nào được công bố trước năm 2017. Không những thế, COTE còn được kết hợp mở rộng với một hệ thống phân cấp VOTE (một cấu trúc phân cấp mới có quyền biểu quyết xác suất) để trở thành HIVE-COTE, đã được chứng minh đạt được một sự cải thiện đáng kể so với COTE. HIVE-COTE hiện được coi là thuật toán đỉnh cao để phân loại chuỗi thời gian khi đánh giá qua 85 bộ dữ liệu từ kho lưu trữ UCR/UEA.

2.3.3 Các phương pháp tiếp cận bằng học sâu

Tuy nhiên, để đạt được độ chính xác cao, HIVE-COTE trở nên cực kỳ tốn kém khi tính toán và không khả thi khi chạy trên một vấn đề khai thác dữ liệu thực tế lớn. Trong khi đó, học sâu (Deep Learning) đã cách mạng hóa lĩnh vực thị giác máy tính (Computer Vision) khi đạt hiệu suất mức cạnh tranh với con người trong việc nhận dạng hình ảnh, cùng với đó là giải quyết tốt các vấn đề của xử lý ngôn ngữ tự nhiên (Natural Language Processing) như dịch máy, phân loại tài liệu, ... và trong lĩnh vực

nhận dạng giọng nói, ... Để ý rằng, có sự tương đồng về khía cạnh liên tục dữ liệu giữa xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói với việc phân loại chuỗi thời gian. Tận dụng đặc điểm đó, Deep Learning hoàn toàn có thể ứng dụng vào giải quyết các bài toán phân loại chuỗi thời gian. Một trong những kiến trúc học sâu đạt được hiệu quả cao trong việc phân loại chuỗi thời gian đó là kiến trúc Residual Network (ResNet).

Đối với TSC bằng học sâu, có thể được tách ra thành hai loại mô hình chính: mô hình sản sinh (generative models) và các mô hình phân biệt (discriminative models).



Hình 17. Học sâu áp dụng cho bài toán phân loại chuỗi thời gian

2.3.3.1 Mô hình sản sinh (generative models)

Các mô hình sản sinh là các mô hình thường có một bước huấn luyện không có giám sát trước giai đoạn huấn luyện các phân loại, mục tiêu là để tìm một đại diện tốt hơn cho chuỗi thời gian trước khi huấn luyện phân loại. Ví dụ như, ta có mô hình bao gồm các bộ auto-encoder loại bỏ nhiễu xếp chồng lên nhau (SDAEs) ^[11]. Auto-encoder vẫn là một dạng neural network, gồm 3 phần chính là encoder, latent representation và

decoder. Khối encoder sẽ nhận đầu vào là vector biểu diễn dữ liệu n chiều, đầu ra là vector trong không gian ẩn m chiều (latent representation). Thông thường thì $m < n$, do đó đây cũng là một phương pháp để giảm số chiều của dữ liệu. Decoder nhận đầu vào là vector m chiều từ bên trên (đầu ra của encoder) và cố gắng khôi phục lại dữ liệu ban đầu (vector n chiều đầu vào của encoder). Có thể hình dung việc này giống như chúng ta nén một tín hiệu và khôi phục lại tín hiệu đó.

Một mô hình sản sinh dựa trên mạng nơ ron tích chập (Convolutional Neural Networks – CNN) với một giai đoạn giải tích chập (deconvolutional operation) cùng với việc sử dụng kỹ thuật lấy mẫu (sampling technique) giúp xây dựng lại một chuỗi thời gian đa biến trước khi đưa vào huấn luyện. Tương tự, ta cũng đã có các auto-encoder dựa trên (RNN) có khả năng sinh ra các chuỗi thời gian mới, huấn luyện để tìm đại diện tiềm ẩn, rồi sau đó sử dụng các đại diện đó để huấn luyện một phân loại (chẳng hạn như Support Vector Machine hoặc Random Forest), hay các DBNs cũng được sử dụng để mô hình hóa các đặc trưng tiềm ẩn một cách không có giám sát sau đó được dùng để phân loại đơn biến và đa biến chuỗi thời gian. Ngoài ra, còn có một số mô hình khác dựa trên Echo States Network (ESN) như kernel-learning, meta-learning. Bản chất của ESN là một mạng RNN với các hidden layer liên kết “lông lẻo” với nhau. Lớp này được gọi là “Reservoir”. Trong mô hình ESN, các đại diện tiềm ẩn được học trong không gian của Reservoir và chúng ta chỉ cần huấn luyện lại trọng số của lớp đầu ra, việc này giúp chúng ta rút ngắn thời gian huấn luyện mô hình.

2.3.3.2 Mô hình phân biệt (discriminative)

Một mô hình phân biệt là một phân loại (hoặc một hồi quy) trực tiếp ánh xạ từ các đầu vào thô của một chuỗi thời gian (hoặc các đặc trưng được thiết kế bằng tay) đến các đầu ra là một phân bố xác suất trên các phân lớp trong tập dữ liệu. Mô hình phân biệt có thể được tiếp tục phân chia thành hai nhóm: mô hình học sâu các đặc trưng được thiết kế bằng tay (hand engineered features) và mô hình đầu-cuối (end-to-end).

Cách thường gặp nhất và là phương pháp khai thác đặc trưng lấy cảm hứng từ Computer Vision là biến đổi chuỗi thời gian thành những hình ảnh sử dụng phương pháp tạo hình đặc biệt, sau đó làm đầu vào cho một bộ phân loại. Đó được gọi là hướng tiếp cận các đặc trưng được thiết kế bằng tay.

Ngược lại với các đặc trưng được thiết kế bằng tay, mục đích của mô hình đầu-cuối (end-to-end) là kết hợp các quá trình học các đặc trưng bằng việc tinh chỉnh các phân loại phân biệt. Loại hình này không bao gồm bất kỳ bước tiền xử lý nào. Ta có thể tách loại mô hình này ra thành các kiến trúc khác nhau dựa trên kiến trúc mạng nơ-ron của chúng.

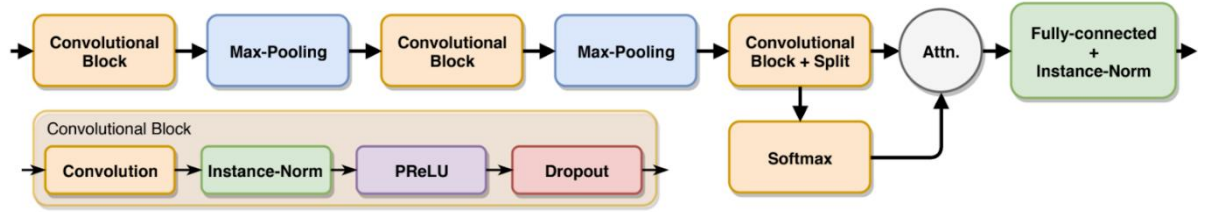
Ta sẽ không tập trung vào các mô hình sản sinh. Nguyên nhân là do nó thường kém chính xác hơn, do nó có thêm bước lắp một bộ sinh chuỗi thời gian - được coi như một rào cản với hầu hết các phương pháp vì các thành phần này thường ít được công bố công khai, và do thực hiện các mô hình này thường phức tạp hơn so với các mô hình phân biệt. Thay vào đó, ta sẽ tập trung vào các mô hình phân biệt, đặc biệt là các mô hình đầu-cuối, bởi vì mục tiêu chính của cách tiếp cận bằng học sâu mà chúng ta đang đề cập là để loại bỏ các sai lệch do các đặc trưng được thiết kế bằng tay, do đó cho phép mạng học được hầu hết các đặc trưng phân biệt hữu ích cho nhiệm vụ phân loại.

❖ Encoder ^[12]

Kiến trúc này là mạng tích chập tiêu chuẩn, được hình thành bởi ba khối chập với hai tầng max-pooling ở giữa xen kẽ. Một khối chập được hình thành bởi 1 tầng tích chập 1 chiều (các bộ lọc có kích thước lần lượt là 128, 256 và 512; với các kernel 5, 11 và 21 có bước trượt bằng 1; và kích thước đệm lần lượt là 3, 5 và 10), với 1 tầng chuẩn hóa instance, hàm kích hoạt PReLU và 1 tầng Drop out (tỉ lệ 0,2 trong tất cả các khối). Sau phần đầu tiên của mạng, một nửa trong số các bộ lọc được đưa vào hàm kích hoạt softmax, một nửa bộ lọc còn lại được áp dụng cơ chế “tập trung” (attention mechanism). Đối với mỗi bộ lọc, ta có:

$$h = h \cdot a \quad (2.7)$$

Trong đó \cdot biểu thị một phép tích chập, h là kết quả của một bộ lọc tích chập 1 chiều trên tín hiệu theo thời gian và a là vector “tập trung” theo thời gian (độc lập cho mỗi bộ lọc). Cuối cùng, kết quả của cơ chế “tập trung” trên tất cả các bộ lọc được chuyển qua một tầng nơ ron kết nối đầy đủ và một tầng chuẩn hóa.



Hình 18. Encoder

❖ Multi-Scale Convolutional Neural Networks (MCNN) ^[13]

MCNN có ba giai đoạn liên tiếp: giai đoạn biến đổi, giai đoạn tích chập cục bộ và giai đoạn tích chập đầy đủ.

- Giai đoạn biến đổi: Áp dụng các biến đổi khác nhau trên chuỗi thời gian đầu vào, bao gồm ánh xạ định danh (Identity Mapping), biến đổi lấy mẫu xuống (down-sampling) trong miền thời gian và biến đổi quang phổ (Spectral Transformation) trong miền tần số. Mỗi phần được gọi là một nhánh, và là một nhánh đầu vào của mạng nơ ron tích chập.

- Lấy mẫu xuống: Giả sử chúng ta có chuỗi thời gian $T = \{t_1, t_2, \dots, t_n\}$ và tốc độ lấy mẫu xuống là k , thì chúng ta sẽ chỉ giữ mọi điểm dữ liệu thứ k trong chuỗi thời gian mới:

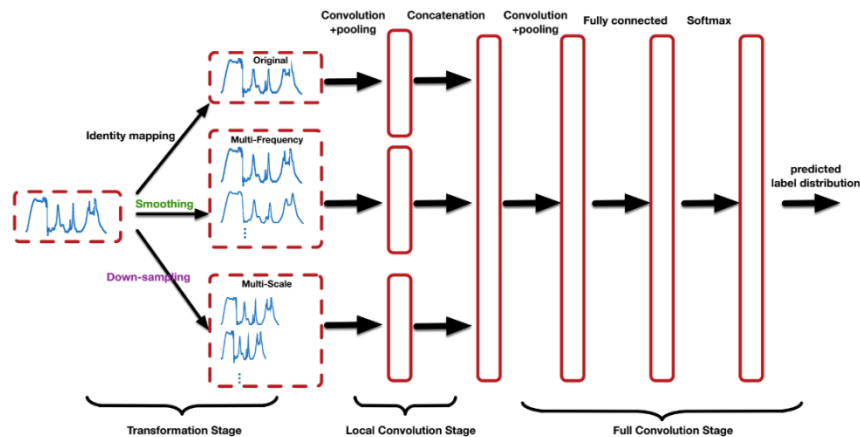
$$T^k = \{t_{1+k*i}\}, i = 0, 1, \dots, [\frac{n-1}{k}] \quad (2.8)$$

- Biến đổi quang phổ: Giả sử chuỗi thời gian ban đầu là $T = \{t_1, t_2, \dots, t_n\}$, biến đổi quang phổ hoạt động bằng cách chuyển đổi chuỗi thời gian ban đầu này thành chuỗi thời gian mới:

$$T^l = \frac{x_i + x_{i+1} + \dots + x_{i+l-1}}{l} \quad (2.9)$$

Trong đó l là kích thước cửa sổ tần số biến đổi và $i = 0, 1, \dots, n - 1 + 1$. Với các l khác nhau, MCNN tạo ra nhiều chuỗi có tần số khác nhau, tất cả sẽ được đưa vào tầng chập cục bộ tương ứng với nhánh biến đổi này.

- Giai đoạn tích chập cục bộ: chúng ta sử dụng một số tầng chập để trích xuất các đặc trưng cho từng nhánh. Trong giai đoạn này, các phép tích chập cho các nhánh khác nhau là độc lập với nhau. Tất cả các đầu ra sẽ đi qua một tầng max-pooling với nhiều kích cỡ khác nhau.
- Giai đoạn tích chập đầy đủ: chúng ta ghép tất cả các đặc trưng được trích xuất và áp dụng thêm một số tầng chập (sau mỗi tầng là một max-pooling), các tầng nơ ron được kết nối đầy đủ và một tầng softmax để tạo đầu ra cuối cùng. Đây hoàn toàn là một hệ thống đầu - cuối (end-to-end) và tất cả các tham số được đào tạo chung thông qua lan truyền ngược.

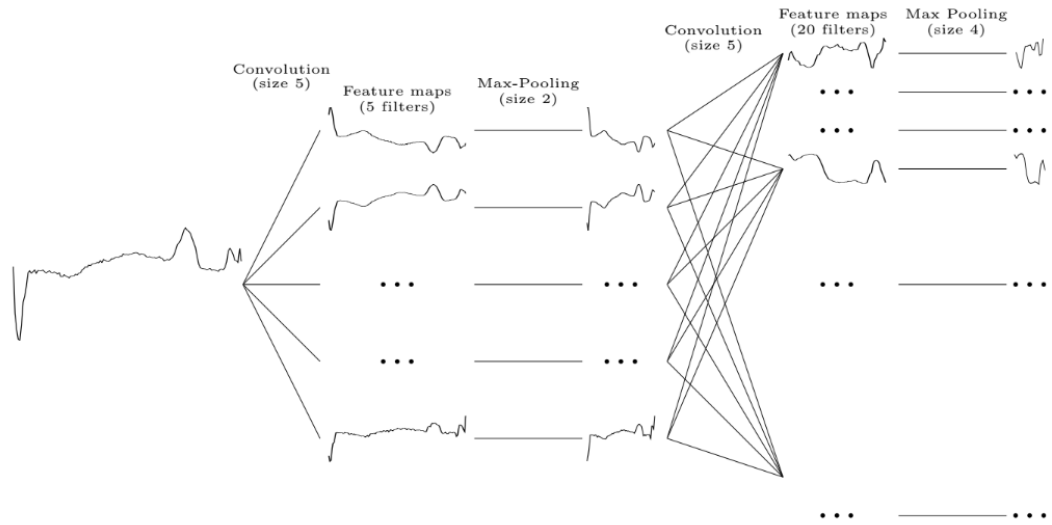


Hình 19. Multi-Scale Convolutional Neural Networks

❖ Time-LeNet (t-leNet) ^[14]

t-leNet là phiên bản cụ thể theo chuỗi thời gian của mô hình leNet. leNet đã chứng minh thành công cho phân loại hình ảnh. Nó được tạo thành từ hai tầng tích chập (tầng tích chập đầu tiên có 5 bộ lọc có bước trượt tạm thời bằng 5, tầng chập thứ hai được tạo thành từ 20 bộ lọc với bước trượt vẫn là 5), sau mỗi tầng tích chập là các

bước lấy mẫu phụ (sub-sampling) qua các tầng max-pooling (max-pooling đầu tiên có kích thước bằng 2, và max-pooling cuối cùng có kích thước là 4). Cuối cùng là các tầng nơ ron được kết nối đầy đủ cho phép ánh xạ các đặc trưng được trích xuất với các nhãn lớp dự đoán.



Hình 20. Time-LeNet

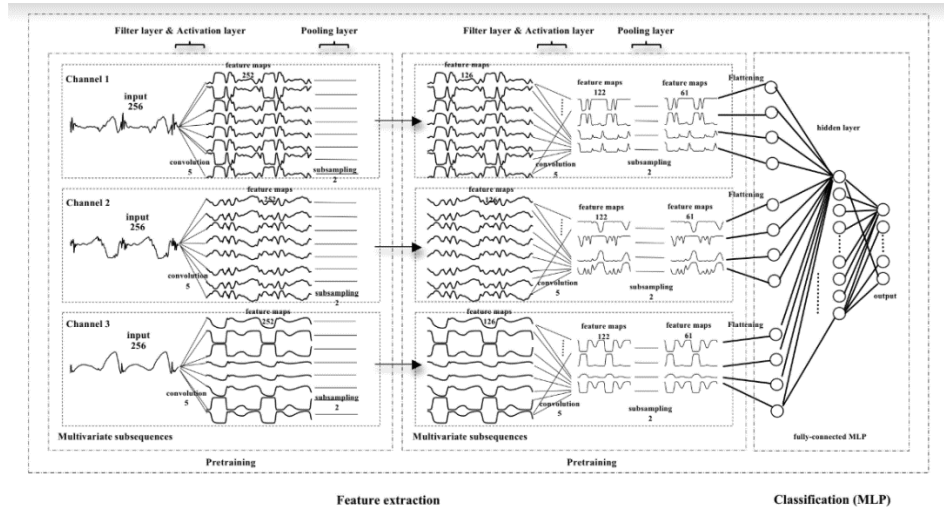
Tăng dữ liệu. Kỹ thuật này nhằm mục đích xây dựng một bộ dữ liệu tổng hợp bằng cách chuyển đổi các mẫu đã được phân loại hiện có để giúp mô hình tìm hiểu phạm vi của các biến đổi trong lớp mà người ta có thể quan sát. Phương pháp đầu tiên bao gồm việc trích xuất các lát cắt từ chuỗi thời gian và thực hiện phân loại theo các lát cắt. Cụ thể khi huấn luyện, mỗi lát được trích xuất từ một chuỗi thời gian của lớp y được gán cùng một lớp và trình phân loại được huấn luyện bằng cách học sử dụng các lát đó. Kích thước của lát cắt là một tham số của phương pháp này. Tại thời điểm kiểm thử, mỗi lát cắt từ chuỗi thời gian thử nghiệm được phân loại bằng cách sử dụng phân loại đã học và thực hiện bỏ phiếu đa số để quyết định lớp dự đoán. Phương pháp này được gọi là cắt cửa sổ (Window Slicing). Phương pháp thứ 2 có tác dụng làm rõ hơn chuỗi thời gian bằng cách làm cong vênh một lát cắt được chọn ngẫu nhiên của một

chuỗi thời gian (tăng tốc độ lên hoặc giảm tốc độ xuống, với các tỷ lệ cong vênh bằng $\frac{1}{2}$ hoặc bằng 2). Phương pháp này sẽ được gọi là cong vênh cửa sổ (Wrapping Window).

Trộn dữ liệu. Để giảm hiện tượng gradient biến mất khi lan truyền ngược, chúng ta đào tạo trước mô hình một cách không giám sát bằng cách sử dụng chuỗi thời gian từ các bộ dữ liệu khác nhau. Sau khi bước này được thực hiện, các bộ lọc tích chập được giữ và phần được giám sát của mô hình sẽ được huấn luyện riêng cho từng tập dữ liệu. Phương pháp này được gọi là trộn dữ liệu (Data Mixing).

❖ Multi-Channels Deep Convolutional Neural Network (MDCNN) ^[15]

Mạng lưới thần kinh chuyển đổi sâu đa kênh (MDCNN) bao gồm hai phần. Một là trình trích xuất đặc trưng, tự động học các đặc trưng từ dữ liệu thô. Cái còn lại là MLP được kết nối đầy đủ có thể huấn luyện, thực hiện phân loại dựa trên các đặc trưng đã học từ phần trước. Nói chung, trình trích xuất đặc trưng bao gồm nhiều giai đoạn tương tự nhau và mỗi giai đoạn được tạo thành từ ba tầng: tầng lọc (sử dụng kích thước bằng 256), tầng kích hoạt (sử dụng hàm ReLU) và tầng tổng hợp (sử dụng max-pooling). Đầu vào và đầu ra của mỗi tầng là các bản đồ đặc trưng.

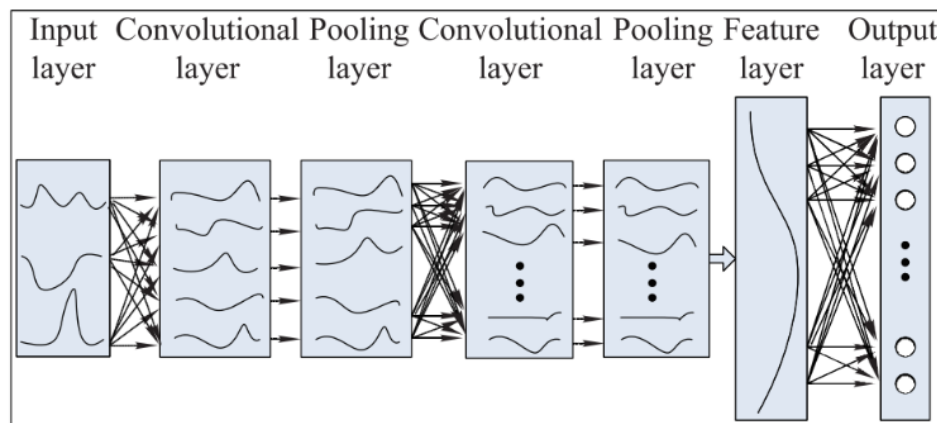


Hình 21. Multi-Channels Deep Convolutional Neural Network

Chúng ta tách chuỗi thời gian đa biến thành các chuỗi đơn biến và thực hiện việc học đặc trưng trên từng chuỗi riêng lẻ, và sau đó MLP truyền thống được kết hợp vào cuối quá trình học đặc trưng được sử dụng để phân loại.

❖ Time Convolutional Neural Network (Time-CNN) ^[16]

Với MCDCNN trước đây, để phân loại chuỗi thời gian đa biến, chúng ta đề xuất một mạng nơ ron tích chập sâu đa kênh. Chúng ta tách chuỗi thời gian đa biến thành các biến thời gian đơn lẻ để học đặc trưng riêng lẻ và đạt được hiệu suất phân loại tốt trong các thử nghiệm. Tuy nhiên, có một hạn chế lớn là phương pháp này không thể khai thác mối quan hệ giữa các chuỗi thời gian đơn biến khác nhau. Để khắc phục khuyết điểm này, chúng ta sửa đổi thuật toán theo cách: thay vì học đặc trưng riêng lẻ, chuỗi thời gian đa biến được đào tạo chung để trích xuất đặc trưng.



Hình 22. Time Convolutional Neural Network

Trong đó:

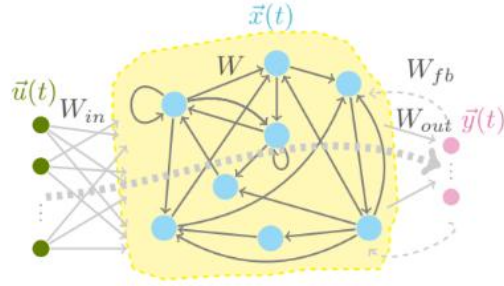
- Tầng đầu vào: Tầng này có $N \times k$ nơ ron, trong đó k biểu thị số lượng biến thiên của chuỗi thời gian đầu vào và N biểu thị độ dài của mỗi chuỗi đơn biến.
- Tầng tích chập: Thực hiện các phép tích chập trên chuỗi thời gian của tầng trước với các bộ lọc tích chập. Trong bước này có các tham số như số bộ lọc m , bước trượt s và kích thước của bộ lọc $k \times l$, trong đó k biểu thị số biến

thiên của chuỗi thời gian trong lớp trước và l biểu thị độ dài của bộ lọc. Hàm kích hoạt phi tuyến f cũng cần được xác định trong tầng này.

- Tầng tổng hợp: Bản đồ đặc trưng được chia thành N phân đoạn có độ dài bằng nhau và sau đó mọi phân đoạn được thể hiện bằng giá trị trung bình hoặc tối đa của nó. Ưu điểm của tầng này lấy mẫu xuống (down-sampling) các dải đầu ra tích chập, do đó làm giảm sự biến đổi trong các kích hoạt ẩn.
- Tầng đặc trưng: Sau một số phép tích chập và gộp, chuỗi thời gian ban đầu được thể hiện bằng một loạt các bản đồ đặc trưng. Chỉ cần kết nối tất cả các bản đồ đặc trưng để tạo ra một chuỗi thời gian dài mới như là đại diện cuối cùng của đầu vào ban đầu cho tầng đặc trưng này.
- Tầng đầu ra: Có n nơ ron, tương ứng với n lớp chuỗi thời gian. Nó được kết nối đầy đủ với tầng đặc trưng. Phương thức phổ biến nhất là lấy nơ ron đầu ra lớn nhất làm lớp của chuỗi thời gian đầu vào.

❖ Time Warping Invariant Echo State Network (TWIESN) ^[17]

TWIESN là kiến trúc biến thể của ESNs, sử dụng trực tiếp đầu vào thô và dự đoán một phân bố xác suất trên các lớp phân loại. Trong thực tế, đối với mỗi thành phần trong một chuỗi thời gian đầu vào, không gian “reservoir” được sử dụng để phóng đại những thành phần này vào một không gian nhiều chiều hơn. Như vậy, đối với một chuỗi thời gian đơn biến, các thành phần của nó được phóng vào một không gian có kích thước được suy ra từ kích thước của “reservoir”. Sau đó, một phân loại Ridge được huấn luyện để dự đoán lớp các thành phần của chuỗi thời gian. Trong thời gian thử nghiệm, với mỗi thành phần của chuỗi thời gian đầu vào, phân loại Ridge sẽ cho ra một phân bố xác suất trên các lớp trong tập dữ liệu. Sau đó, xác suất cho mỗi lớp được lấy trung bình trên tất cả các thành phần của chuỗi thời gian, rồi cuối cùng gán cho chuỗi thời gian đầu vào nhãn mà có xác suất trung bình lớn nhất.



Hình 23. Time Warping Invariant Echo State Network

2.4 Ba kiến trúc học sâu hiệu quả cho bài toán phân loại chuỗi thời gian

2.4.1 Multilayer Perceptron (MLP)^[2]

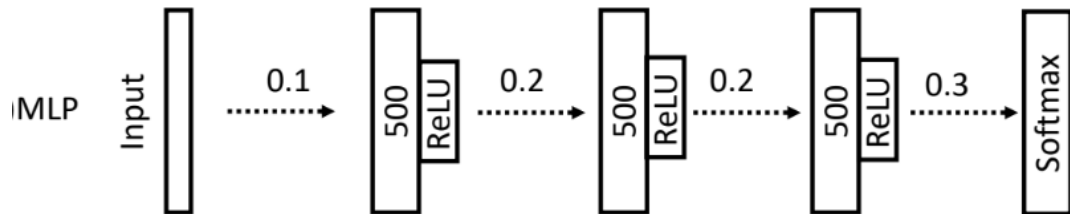
MLP là kiến trúc đơn giản được tạo bằng cách xếp chồng ba lớp nơ-ron được kết nối đầy đủ. Mỗi lớp có 500 nơ-ron tuân theo hai quy tắc: sử dụng cơ chế Dropout tại mỗi lớp đầu vào để cải thiện khả năng khái quát (tỷ lệ Drop out ở lớp đầu vào, lớp ẩn và lớp softmax lần lượt là $\{0,1; 0,2; 0,3\}$) và sử dụng hàm kích hoạt ReLU để ngăn chặn sự bão hòa của đạo hàm khi mạng trở nên sâu hơn. Mạng kết thúc với một lớp softmax. Một khối lớp cơ bản được mô tả như sau:

$$\tilde{x} = f_{drop\ out, p}(x) \quad (2.10)$$

$$y = \mathbf{W} \cdot \tilde{x} + \mathbf{b} \quad (2.11)$$

$$h = ReLU(y) \quad (2.12)$$

Với việc sử dụng kỹ thuật Drop out cho mỗi tầng trong kiến trúc của mình, MLP hứa hẹn có khả năng tránh overfitting và khái quát tốt hơn trên nhiều tập dữ liệu.



Hình 24. Kiến trúc của MLP

2.4.2 Fully Convolutional Neural Networks (FCN)^[2]

FCN hoạt động hiệu quả đối với phân đoạn trên hình ảnh. Trong bài toán phân loại chuỗi thời gian, FCN được thực hiện dưới dạng trích xuất đặc trưng. Đầu ra cuối cùng của nó vẫn đến từ lớp softmax. Một khối chập cơ bản trong FCN bao gồm một tầng chập theo sau là tầng chuẩn hóa hàng loạt (được áp dụng để tăng tốc độ hội tụ và giúp cải thiện tổng quát hóa) và tầng kích hoạt ReLU. Phép tích chập được thực hiện bằng ba kernel 1-D với các kích thước $\{8, 5, 3\}$ mà không cần bước trượt. Một khối chập cơ bản được mô tả như sau:

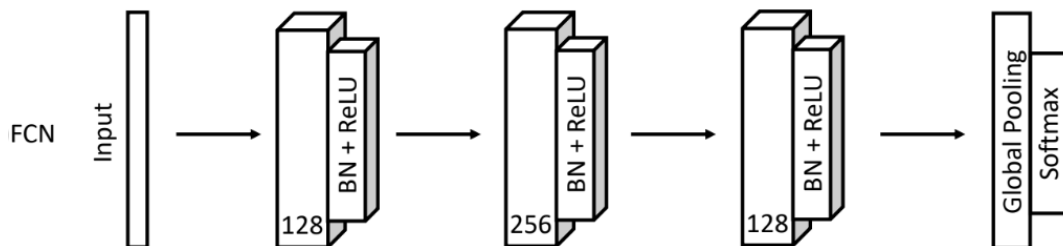
$$y = \mathbf{W} \oplus x + b \quad (2.13)$$

$$s = BN(y) \quad (2.14)$$

$$h = ReLU(s) \quad (2.15)$$

Mạng FCN được xây dựng bằng cách xếp chồng ba khối chập cơ bản như trên với các kích thước bộ lọc lần lượt là $\{128, 256, 128\}$ trong mỗi khối. Sau các khối chập, các đặc trưng được đưa vào một lớp tổng hợp trung bình thay vì một tầng pooling được kết nối đầy đủ, điều này làm giảm phần lớn số lượng trọng số. Nhãn cuối cùng được tạo bởi một tầng softmax.

Như đã nói ở phần trước, Batch normalization còn có vai trò như một dạng của regularization giúp cho việc giảm thiểu overfitting. Vì vậy, sử dụng Batch normalization cho FCN hứa hẹn sẽ đạt hiệu suất tốt hơn các phương pháp không sử dụng kỹ thuật này.



Hình 25. Kiến trúc của FCN

2.4.3 Residual Network (ResNet)^[2]

Các mạng ResNet được mở rộng thành một cấu trúc rất sâu bằng cách thêm kết nối tắt trong mỗi khối “đư” (residual) để cho phép tính toán gradient trực tiếp qua các tầng dưới cùng của một khối “đư” trong mạng. Nó đạt được hiệu suất tối ưu trong phát hiện đối tượng và các nhiệm vụ liên quan đến thị giác máy tính khác. Trong bài toán phân loại chuỗi thời gian, chúng ta xây dựng từng khối “đư” như sau:

$$h_1 = \text{Block}_{k_1}(x) \quad (2.16)$$

$$h_2 = \text{Block}_{k_2}(h_1) \quad (2.17)$$

$$h_3 = \text{Block}_{k_3}(h_2) \quad (2.18)$$

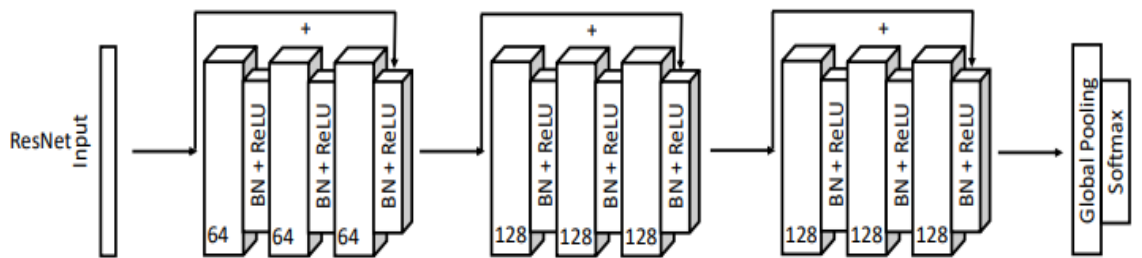
$$y = h_3 + x \quad (2.19)$$

$$\hat{h} = \text{ReLU}(y) \quad (2.20)$$

Trong đó, *Block k* là khối tích chập có số lượng bộ lọc *k*. Số lượng bộ lọc của các *Block k* lần lượt là {64, 128, 128}. ResNet được xây dựng bằng cách xếp chồng ba khối “đư” với một tầng tổng hợp trung bình và một tầng softmax.

Ngoài các thành phần cơ bản như đã đề cập ở phần trên, ResNet được mở rộng sâu hơn bằng cách thêm kết nối tắt trong mỗi khối đư để cho phép dòng gradient chảy trực tiếp qua các lớp dưới cùng. Nó đạt được hiệu suất cao trong phát hiện đối tượng và các nhiệm vụ liên quan đến nhận dạng khác.

Xếp chồng 3 khối đư cần thiết, mỗi khối có số bộ lọc cho mỗi lớp tích chập lần lượt là 64, 128, 128 cùng với một Global Average Pooling và cuối cùng là một phân loại Softmax, ta có mạng ResNet như hình sau:



Hình 26. Kiến trúc của ResNet

CHƯƠNG 3 – THỰC NGHIỆM

1.1 Dữ liệu

Thực nghiệm sử dụng 113 tập dữ liệu chuỗi thời gian đơn biến của kho dữ liệu UCR 2018. Đây là những tập dữ liệu đầy đủ và mới nhất để tiến hành các thực nghiệm liên quan tới phân loại chuỗi thời gian hiện nay.

Tên tập dữ liệu	Số mẫu huấn luyện	Số mẫu kiểm tra	Độ dài chuỗi	Số lớp phân loại	Dạng dữ liệu
ACSF1	100	100	1460	10	DEVICE
Adiac	390	391	176	37	IMAGE
ArrowHead	36	175	251	3	IMAGE
Beef	30	30	470	5	SPECTRO
BeetleFly	20	20	512	2	IMAGE
BirdChicken	20	20	512	2	IMAGE
BME	30	150	128	3	SIMULATED
Car	60	60	577	4	SENSOR
CBF	30	900	128	3	SIMULATED
Chinatown	20	345	24	2	Traffic
Chlorine Concentration	467	3840	166	3	SIMULATED
CinCECGTorso	40	1380	1639	4	ECG
Coffee	28	28	286	2	SPECTRO
Computers	250	250	720	2	DEVICE
CricketX	390	390	300	12	MOTION

CricketY	390	390	300	12	MOTION
CricketZ	390	390	300	12	MOTION
Crop	7200	16800	46	24	IMAGE
DiatomSize Reduction	16	306	345	4	IMAGE
DistalPhalanx OutlineAgeGroup	400	139	80	3	IMAGE
DistalPhalanx OutlineCorrect	600	276	80	2	IMAGE
DistalPhalanxTW	400	139	80	6	IMAGE
Earthquakes	322	139	512	2	SENSOR
ECG200	100	100	96	2	ECG
ECG5000	500	4500	140	5	ECG
ECGFiveDays	23	861	136	2	ECG
ElectricDevices	8926	7711	96	7	DEVICE
EOGHorizontal Signal	362	362	1250	12	EOG
EOGVertical Signal	362	362	1250	12	EOG
Ethanol Concentration	261	263	1751	4	OTHER
EthanolLevel	504	500	1751	4	SPECTRO
FaceAll	560	1690	131	14	IMAGE
FaceFour	24	88	350	4	IMAGE

FacesUCR	200	2050	131	14	IMAGE
FiftyWords	450	455	270	50	IMAGE
Fish	175	175	463	7	IMAGE
FordA	3601	1320	500	2	SENSOR
FordB	3636	810	500	2	SENSOR
FreezerRegular Train	150	2850	301	2	SENSOR
FreezerSmall Train	28	2850	301	2	SENSOR
Fungi	18	186	201	18	OTHER
GunPoint	50	150	150	2	MOTION
GunPointAge Span	135	316	150	2	MOTION
GunPointMale VersusFemale	135	316	150	2	MOTION
GunPointOld VersusYoung	135	316	150	2	MOTION
Ham	109	105	431	2	SPECTRO
HandOutlines	1000	370	2709	2	IMAGE
Haptics	155	308	1092	5	MOTION
Herring	64	64	512	2	IMAGE
HouseTwenty	34	101	3000	2	DEVICE
InlineSkate	100	550	1882	7	MOTION
InsectEPG	62	249	601	3	EPG

RegularTrain					
InsectEPG SmallTrain	17	249	601	3	EPG
InsectWingbeat Sound	220	1980	256	11	SENSOR
ItalyPower Demand	67	1029	24	2	SENSOR
LargeKitchen Appliances	375	375	720	3	DEVICE
Lightning2	60	61	637	2	SENSOR
Lightning7	70	73	319	7	SENSOR
Mallat	55	2345	1024	8	SIMULATED
Meat	60	60	448	3	SPECTRO
MedicalImages	381	760	99	10	IMAGE
MiddlePhalanx OutlineAgeGroup	400	154	80	3	IMAGE
MiddlePhalanx OutlineCorrect	600	291	80	2	IMAGE
MiddlePhalanx TW	399	154	80	6	IMAGE
MixedShapes	500	2425	1024	5	IMAGE
MixedShapes SmallTrain	100	2425	1024	5	IMAGE
MoteStrain	20	1252	84	2	SENSOR
NonInvasiveFetal	1800	1965	750	42	ECG

ECGThorax1					
NonInvasiveFetal ECGThorax2	1800	1965	750	42	ECG
OliveOil	30	30	570	4	SPECTRO
OSULeaf	200	242	427	6	IMAGE
Phalanges OutlinesCorrect	1800	858	80	2	IMAGE
Phoneme	214	1896	1024	39	SOUND
PigAirway Pressure	104	208	2000	52	HEMODYNA MICS
PigArt Pressure	104	208	2000	52	HEMODYNA MICS
PigCVP	104	208	2000	52	HEMODYNA MICS
Plane	105	105	144	7	SENSOR
PowerCons	180	180	144	2	DEVICE
ProximalPhalanx OutlineAgeGrou p	400	205	80	3	IMAGE
ProximalPhalanx OutlineCorrect	600	291	80	2	IMAGE
ProximalPhalanx TW	400	205	80	6	IMAGE
Refrigeration Devices	375	375	720	3	DEVICE
Rock	20	50	2844	4	SPECTRO

ScreenType	375	375	720	3	DEVICE
SemgHand GenderCh2	300	600	1500	2	SPECTRO
SemgHand MovementCh2	450	450	1500	6	SPECTRO
SemgHand SubjectCh2	450	450	1500	5	SPECTRO
ShapeletSim	20	180	500	2	SIMULATED
ShapesAll	600	600	512	60	IMAGE
SmallKitchen Appliances	375	375	720	3	DEVICE
SmoothSubspace	150	150	15	3	SIMULATED
SonyAIBO RobotSurface1	20	601	70	2	SENSOR
SonyAIBO RobotSurface2	27	953	65	2	SENSOR
StarlightCurves	1000	8236	1024	3	SENSOR
Strawberry	613	370	235	2	SPECTRO
SwedishLeaf	500	625	128	15	IMAGE
Symbols	25	995	398	6	IMAGE
SyntheticControl	300	300	60	6	SIMULATED
ToeSegmentation 1	40	228	277	2	MOTION
ToeSegmentation 2	36	130	343	2	MOTION

Trace	100	100	275	4	SENSOR
TwoLeadECG	23	1139	82	2	ECG
TwoPatterns	1000	4000	128	4	SIMULATED
UMD	36	144	150	3	SIMULATED
UWaveGesture LibraryAll	896	3582	945	8	MOTION
UWaveGesture LibraryX	896	3582	315	8	MOTION
UWaveGesture LibraryY	896	3582	315	8	MOTION
UWaveGesture LibraryZ	896	3582	315	8	MOTION
Wafer	1000	6164	152	2	SENSOR
Wine	57	54	234	2	SPECTRO
WordSynonyms	267	638	270	25	IMAGE
Worms	181	77	900	5	MOTION
WormsTwoClass	181	77	900	2	MOTION
Yoga	300	3000	426	2	IMAGE

Bảng 1. Mô tả thuộc tính dữ liệu thực nghiệm

1.2 Phương pháp đánh giá

Dự đoán	Dự đoán mẫu là dương tính	Dự đoán mẫu là âm tính
Thực tế		
Mẫu thực sự là dương tính	True Positive (TP)	False Negative (FN)
Mẫu thực sự là âm tính	False Positive (FP)	True Negative (TN)

Bảng 2. Mô tả các lớp mẫu dữ liệu thực nghiệm

Chúng ta đánh giá thực nghiệm dựa trên so sánh các chỉ số sau với các kết quả tương ứng từ tiền thực nghiệm của Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar và Pierre-Alain Muller:

- **Accuracy (Acc):** Tỷ lệ giữa kết quả dự đoán chính xác (True) và tổng của tất cả các dự đoán.

$$\frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

- **Precision (Pre):** Tỷ lệ giữa tất cả các dự đoán True Positive chia cho tất cả các dự đoán Positive.

$$\frac{TP}{TP+FP} \quad (3.2)$$

- **Recall (Rec):** Tỷ lệ giữa True Positive chia cho tất cả các Positive thực sự.

$$\frac{TP}{TP+FN} \quad (3.3)$$

- **Train time (TrT):** Thời gian huấn luyện mạng trên tập huấn luyện.

2.3 Kết quả

Thực nghiệm so sánh được tiến hành với giải thuật gốc của Zhiguang Wang, Weizhong Yan và Tim Oates chạy trên GPU Tesla K80 của Google cho kết quả như sau:

Bảng 3. Mô tả kết quả thực nghiệm

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
MLP	ACSF1	0.6243	0.57	0.57	488.3	0.517	0.49	0.49	433.3		
	Adiac	0.4003	0.3632	0.3667	690.2	0.1874	0.2199	0.2204	816.1		
	ArrowHead	0.7821	0.7771	0.781	920.5	0.7013	0.68	0.674	925.8		
	Beef	0.7367	0.6667	0.6667	286.7	0.78	0.7333	0.7333	365.9		
	BeetleFly	0.9167	0.9	0.9	284.5	0.8846	0.85	0.85	368		
	BirdChicken	0.8125	0.8	0.8	284.7	0.6	0.6	0.6	368.5		
	BME	0.9399	0.9267	0.9267	822.4	0.8981	0.8533	0.8533	647.4		
	Car	0.7892	0.7667	0.768	297.4	0.7622	0.75	0.7689	379.4		
	CBF	0.8855	0.8689	0.8696	1601.1	0.8744	0.8567	0.8573	2580.9		
	Chinatown	0.8565	0.8921	0.9091	1756.7	0.9608	0.9767	0.9839	2122.8		
MLP	ChlorineConcentration	0.7892	0.8063	0.7601	3040.2	0.617	0.5482	0.3965	2410.1		
	CinCECGTorso	0.8654	0.842	0.8429	3412.9	0.8624	0.8304	0.8316	3078.4		
	Coffee	0.9688	0.9643	0.9615	380.8	0.9643	0.9643	0.9667	456.4		
	Computers	0.5487	0.548	0.548	716.8	0.5627	0.56	0.56	649.2		
	CricketX	0.6114	0.5872	0.5957	1092	0.5722	0.5564	0.5683	861.2		
	CricketY	0.6424	0.5949	0.5954	1110.5	0.6156	0.5897	0.5911	861.9		
	CricketZ	0.6333	0.6308	0.622	1100.4	0.5817	0.5538	0.5518	858.8		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
MLP	Crop	0.6561	0.5915	0.5915	25613.9	0.6977	0.7067	0.7067	13037.3		
	DiatomSizeReduction	0.9511	0.9379	0.8708	1818	0.6645	0.8824	0.742	2556.7		
	DistalPhalanxOutlineAgeGroup	0.6106	0.6403	0.6606	963.7	0.7505	0.7626	0.7898	610.5		
	DistalPhalanxOutlineCorrect	0.7449	0.7319	0.6994	932.8	0.7559	0.7101	0.6634	965.9		
	DistalPhalanxTW	0.4826	0.6115	0.4996	609.9	0.5713	0.7194	0.5486	625		
	Earthquakes	0.5798	0.7338	0.5283	865.4	0.5769	0.741	0.5141	639.7		
	ECG200	0.9044	0.91	0.8993	443.6	0.8787	0.88	0.8576	382.5		
	ECG5000	0.6461	0.9313	0.5678	1999.3	0.7158	0.9407	0.5547	3556.5		
	ECGFiveDays	0.9791	0.9791	0.9791	2216.7	0.9111	0.9106	0.9107	4658.2		
	ElectricDevices	0.5601	0.5822	0.4913	23645	0.535	0.5811	0.5099	12977.8		
MLP	EOGHorizontalSignal	0.4776	0.4337	0.4326	1040.1	0.4751	0.4613	0.4614	839.9		
	EOGVerticalSignal	0.3711	0.4006	0.4	1104.9	0.3684	0.4033	0.4024	852.1		
	EthanolLevel	0.0854	0.248	0.25	1459.3	0.4169	0.388	0.3868	1111.2		
	FaceAll	0.7937	0.7982	0.8642	1260.7	0.7788	0.8166	0.8506	1739.1		
	FaceFour	0.8471	0.8409	0.8519	519	0.8158	0.8068	0.8213	654.7		
	FacesUCR	0.8313	0.8366	0.8029	910.6	0.7548	0.7493	0.6965	1756		
	FiftyWords	0.5868	0.7099	0.5879	851.2	0.5779	0.6857	0.5536	995.6		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả				Thực nghiệm của bản thân			
		tiền thực nghiệm							
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT
	Fish	0.8658	0.84	0.8424	508.6	0.7372	0.7143	0.7175	492
	FordA	0.8265	0.8265	0.8268	8536.4	0.8511	0.85	0.851	4522.1
	FordB	0.7189	0.7185	0.7183	8355	0.7183	0.7074	0.7063	4211.5
	FreezerRegularTrain	0.9156	0.9081	0.9081	2061	0.8205	0.82	0.82	1712.5
	FreezerSmallTrain	0.692	0.6919	0.6919	11935.6	0.6853	0.6853	0.6853	10605.8
	Fungi	0.891	0.8387	0.8451	2104.9	0.8636	0.8172	0.8332	1622.4
	GunPoint	0.9401	0.94	0.9399	611.1	0.8136	0.8067	0.8057	357
	GunPointAgeSpan	0.9338	0.9335	0.9337	647.3	0.8928	0.8861	0.8853	411.3
	GunPointMaleVersusFemale	0.9808	0.981	0.9819	602.3	0.9807	0.981	0.9816	408.2
	GunPointOldVersusYoung	0.9348	0.9333	0.9321	643.2	0.9206	0.9206	0.9203	409.4
MLP	Ham	0.7152	0.7143	0.713	315.4	0.7152	0.7143	0.713	317.8
	HandOutlines	0.9142	0.9081	0.8854	2581.8	0.9262	0.9297	0.9204	933.4
	Haptics	0.4226	0.4318	0.4318	625.2	0.435	0.4545	0.4541	395.7
	Herring	0.4591	0.5	0.4636	317.1	0.625	0.5938	0.6215	307.5
	HouseTwenty	0.7365	0.7227	0.7388	652.9	0.7753	0.7731	0.7823	431.7
	InlineSkate	0.3804	0.3345	0.3496	858.7	0.362	0.3182	0.3311	526.7
	InsectEPGRegularTrain	0.577	0.6426	0.5879	799.8	0.5722	0.6426	0.587	530.3

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
	InsectEPGSmallTrain	0.5929	0.6426	0.6004	2711.6	0.6121	0.6506	0.6103	1570		
	InsectWingbeatsSound	0.6147	0.6116	0.6116	932.2	0.6519	0.6414	0.6414	1105.9		
	ItalyPowerDemand	0.9511	0.9504	0.9505	1069.5	0.9614	0.9611	0.9612	1372.2		
	LargeKitchenAppliances	0.4569	0.4613	0.4613	1075.2	0.4924	0.496	0.496	542.2		
	Lightning2	0.6872	0.6885	0.6824	486.2	0.7399	0.7377	0.7305	292.3		
	Lightning7	0.6123	0.6438	0.6532	451.1	0.6117	0.6712	0.6531	296.1		
	Mallat	0.9337	0.9164	0.917	4231.8	0.9332	0.9113	0.912	2768.6		
MLP	Meat	0.9231	0.9	0.9	289	0.4921	0.6333	0.6333	255.7		
	MedicalImages	0.6954	0.7276	0.6867	787.2	0.6323	0.6684	0.6348	713.5		
	MiddlePhalanxOutlineAgeGroup	0.4489	0.5325	0.422	1019.1	0.5607	0.6039	0.4818	414.1		
	MiddlePhalanxOutlineCorrect	0.7661	0.7663	0.7715	931.7	0.76	0.7148	0.6779	680.5		
	MiddlePhalanxTW	0.3727	0.539	0.3771	623.1	0.4315	0.6169	0.4452	655.6		
	MixedShapesRegularTrain	0.9039	0.906	0.9114	2477.2	0.9062	0.9068	0.912	2471.8		
	MixedShapesSmallTrain	0.8449	0.8429	0.8533	2556.1	0.8365	0.8297	0.8425	3070.4		
	MoteStrain	0.856	0.857	0.8567	5256.5	0.8688	0.8698	0.8697	4860.1		
	NonInvasiveFetalECGThorax1	0.9127	0.9125	0.9126	5022.9	0.8927	0.888	0.8872	3513.6		
	NonInvasiveFetalECGThorax2	0.9191	0.9196	0.9169	5354.9	0.9108	0.9094	0.9082	3524.8		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
	OliveOil	0.7375	0.7	0.5639	290.4	0.1	0.4	0.25	371.2		
	OSULeaf	0.5642	0.5537	0.5554	387.8	0.5626	0.5455	0.5544	572.7		
	PhalangesOutlinesCorrect	0.7283	0.7413	0.7329	4393	0.7294	0.7145	0.6544	2385		
	Phoneme	0.048	0.0997	0.0446	1539.1	0.0473	0.0934	0.0397	1859.2		
	PigAirwayPressure	0.0673	0.0673	0.0673	586.7	0.0432	0.0481	0.0481	650.1		
	PigArtPressure	0.1161	0.101	0.101	585.4	0.0955	0.1058	0.1058	652.1		
	PigCVP	0.0805	0.0721	0.0721	590.2	0.1047	0.0865	0.0865	656.5		
	Plane	0.9832	0.981	0.9821	300.6	0.9609	0.9619	0.9637	463.9		
	PowerCons	0.9787	0.9778	0.9778	532.2	0.9737	0.9722	0.9722	490.1		
	MLP										
	ProximalPhalanxOutlineAgeGroup	0.7764	0.8585	0.7347	625.2	0.7807	0.8634	0.818	644		
	ProximalPhalanxOutlineCorrect	0.6302	0.6942	0.5923	932.8	0.7394	0.756	0.655	935.3		
	ProximalPhalanxTW	0.4615	0.7756	0.4669	602.1	0.3876	0.7951	0.4468	643.9		
	RefrigerationDevices	0.3926	0.3893	0.3893	1102.9	0.3612	0.36	0.36	863.3		
	Rock	0.9149	0.88	0.9048	581.9	0.9197	0.9	0.9167	561.9		
	ScreenType	0.4095	0.4053	0.4053	1055.8	0.3991	0.3947	0.3947	861.8		
	SemgHandGenderCh2	0.8144	0.83	0.8374	1033.4	0.81	0.82	0.8385	974.6		
	SemgHandMovementCh2	0.4615	0.4156	0.4156	1379.2	0.4591	0.4333	0.4333	1062.9		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
	SemgHandSubjectCh2	0.8443	0.8422	0.8422	1335.9	0.8152	0.8089	0.8089	1066.9		
	ShapeletSim	0.5351	0.5333	0.5333	677.5	0.5407	0.5389	0.5389	993.3		
	ShapesAll	0.7861	0.78	0.78	1785.4	0.748	0.73	0.73	1230.8		
	SmallKitchenAppliances	0.3903	0.384	0.384	1078.5	0.432	0.432	0.432	552.8		
	SmoothSubspace	0.9678	0.9667	0.9667	443.9	0.7723	0.7733	0.7733	269.8		
	SonyAIBORobotSurface1	0.7866	0.7121	0.7454	2769	0.7773	0.6922	0.7279	2183.8		
	SonyAIBORobotSurface2	0.8246	0.8342	0.8246	4242.3	0.8383	0.8468	0.8369	3404.6		
	StarLightCurves	0.9335	0.9553	0.939	7081.4	0.8141	0.834	0.891	4643.2		
MLP	Strawberry	0.9509	0.9595	0.9634	1581.3	0.8008	0.7784	0.8227	731.2		
	SwedishLeaf	0.8513	0.848	0.851	885.1	0.8046	0.7856	0.79	781.4		
	Symbols	0.8547	0.8462	0.8421	2646.5	0.8818	0.8663	0.8619	3680.7		
	SyntheticControl	0.972	0.97	0.97	834.9	0.9661	0.9633	0.9633	477.5		
	ToeSegmentation1	0.5682	0.5702	0.5676	496.1	0.5911	0.5921	0.5912	537.7		
	ToeSegmentation2	0.6546	0.7154	0.7449	488.4	0.6697	0.7538	0.7524	565.1		
	Trace	0.8087	0.8	0.8097	282	0.5901	0.58	0.5814	280		
	TwoLeadECG	0.7895	0.7471	0.7473	3011.1	0.6858	0.6471	0.6473	6008		
	TwoPatterns	0.9519	0.952	0.9522	4260.5	0.8871	0.887	0.8871	2922.3		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
	UMD	0.9379	0.9375	0.9375	822.1	0.8941	0.8889	0.8889	678.7		
	UWaveGestureLibraryAll	0.9529	0.9517	0.9518	3883.8	0.9546	0.9534	0.9534	3713.2		
	UWaveGestureLibraryX	0.7521	0.7688	0.7665	4040.5	0.7345	0.761	0.7583	3284.9		
	UWaveGestureLibraryY	0.699	0.7038	0.7043	3810.4	0.6846	0.6879	0.6888	3272.9		
	UWaveGestureLibraryZ	0.6829	0.6985	0.7003	3817.3	0.6914	0.7063	0.7081	3305.5		
MLP	Wafer	0.9885	0.9959	0.9905	5362.4	0.9861	0.9951	0.9887	5062.1		
	Wine	0.6615	0.6296	0.6296	327.3	0.25	0.5	0.5	489.5		
	WordSynonyms	0.539	0.5956	0.4498	951.3	0.4661	0.5643	0.4161	915.5		
	Worms	0.4644	0.4416	0.4031	513	0.604	0.5714	0.5339	464.1		
	WormsTwoClass	0.6166	0.6234	0.6174	482.8	0.6264	0.6234	0.6288	466.3		
FCN	Yoga	0.8574	0.8567	0.8543	2268.4	0.8017	0.8003	0.7965	2491.3		
	ACSF1	0.9136	0.91	0.91	382.5	0.906	0.9	0.9	538.5		
	Adiac	0.8763	0.8517	0.8582	881	0.8677	0.8465	0.8504	505.8		
	ArrowHead	0.8634	0.8514	0.8642	678.1	0.8718	0.8571	0.8676	377.3		
	Beef	0.6891	0.6667	0.6667	338.4	0.72	0.7	0.7	160.6		
	BeetleFly	1	1	1	337.5	0.9167	0.9	0.9	145.7		
	BirdChicken	1	1	1	246.7	1	1	1	143.5		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
FCN	BME	0.8318	0.8333	0.8333	577.9	0.8662	0.8667	0.8667	263.5		
	Car	0.9186	0.9167	0.9174	360	0.8975	0.9	0.8981	225.2		
	CBF	0.9946	0.9944	0.9944	2162.5	0.9945	0.9944	0.9945	1588		
	Chinatown	0.9676	0.9796	0.9826	1337.5	0.9676	0.9796	0.9826	1234.5		
	ChlorineConcentration	0.8128	0.8148	0.7653	2376.7	0.8108	0.8221	0.7798	1897.6		
	CinCECGTorso	0.8373	0.8348	0.8346	2311.1	0.8183	0.8145	0.8145	3686.2		
	Coffee	1	1	1	468.7	1	1	1	307.8		
	Computers	0.8202	0.82	0.82	599.6	0.8282	0.828	0.828	1092.8		
	CricketX	0.7919	0.7846	0.7876	834.3	0.807	0.7949	0.7993	963.3		
	CricketY	0.7763	0.7795	0.7837	877.8	0.7995	0.7872	0.7906	969.8		
CricketZ	0.801	0.8077	0.7996	840.1	0.7927	0.8	0.7885	963.4			
Crop	0.7275	0.736	0.736	18013.5	0.7317	0.7386	0.7386	8323			
DiatomSizeReduction	0.0774	0.3007	0.25	2240.1	0.0754	0.3007	0.25	1527.2			
DistalPhalanxOutlineAgeGroup	0.6969	0.7266	0.7434	539.1	0.6826	0.705	0.7451	447.8			
DistalPhalanxOutlineCorrect	0.7447	0.7428	0.7186	1185.9	0.7658	0.7681	0.7516	678.1			
DistalPhalanxTW	0.6523	0.7122	0.5884	550.4	0.5628	0.6547	0.5312	451			
Earthquakes	0.5446	0.7194	0.5187	627.9	0.5866	0.7194	0.5566	925.8			

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
FCN	ECCG200	0.8957	0.9	0.8854	232.3	0.8824	0.89	0.8776	262.7		
	ECCG5000	0.7137	0.9393	0.5507	2482.9	0.7184	0.9409	0.5364	2526		
	ECCGFiveDays	0.9873	0.9872	0.9872	2850.6	0.9861	0.9861	0.9861	2879.4		
	ElectricDevices	0.6725	0.6987	0.6034	17377.4	0.6848	0.7079	0.6171	10204.6		
	EOGHorizontalSignal	0.6266	0.5442	0.5452	1027.3	0.6468	0.5691	0.5697	2433.8		
	EOGVerticalSignal	0.4684	0.4448	0.4458	1041.3	0.4734	0.4503	0.4518	2436.3		
	EthanolLevel	0.3593	0.268	0.2661	1579.5	0.6498	0.632	0.6315	4419.9		
	FaceAll	0.9197	0.9509	0.9632	1548.3	0.9185	0.9107	0.9481	1339.3		
	FaceFour	0.9335	0.9205	0.9257	412.4	0.932	0.9205	0.9274	423.9		
	FacesUCR	0.9465	0.9454	0.9089	1071	0.9446	0.9468	0.9156	1233.6		
FCN	FiftyWords	0.5363	0.6396	0.4897	1169.2	0.5692	0.6571	0.5025	1030.8		
	Fish	0.9548	0.9543	0.9546	401.7	0.9544	0.9543	0.956	600.6		
	FordA	0.9174	0.9174	0.9178	6798.3	0.9173	0.9174	0.9178	6051.8		
	FordB	0.7741	0.7741	0.774	6415.7	0.7778	0.7778	0.7777	5833.8		
	FreezerRegularTrain	0.9969	0.9968	0.9968	1489.4	0.9962	0.9961	0.9961	1063.6		
	FreezerSmallTrain	0.7258	0.7053	0.7053	8668.2	0.8629	0.8547	0.8547	4138		
	Fungi	0.0021	0.0054	0.0079	1809.3	0.0081	0.0645	0.0635	757.3		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
FCN	GunPoint	1	1	1	460.2	1	1	1	349.4		
	GunPointAgeSpan	0.9969	0.9968	0.9968	454.7	0.9937	0.9937	0.9937	427.6		
	GunPointMaleVersusFemale	0.9967	0.9968	0.997	452.8	0.9967	0.9968	0.997	425.8		
	GunPointOldVersusYoung	0.9911	0.9905	0.99	481.3	0.9911	0.9905	0.99	424.7		
	Ham	0.6979	0.6952	0.6966	369.7	0.7388	0.7333	0.7353	426.4		
	HandOutlines	0.7664	0.7027	0.7646	3905.6	0.9145	0.9054	0.88	10782		
	Haptics	0.4974	0.487	0.4859	529.6	0.4904	0.4903	0.4885	1190.9		
	Herring	0.6603	0.6563	0.6012	381	0.6151	0.625	0.6174	368.5		
	HouseTwenty	0.9859	0.9832	0.98	467.9	0.9859	0.9832	0.98	797.9		
	InlineSkate	0.3443	0.3382	0.3481	853.3	0.3268	0.3273	0.3453	1992.3		
	InsectEPGRegularTrain	1	1	1	559.6	0.9972	0.996	0.9921	644.3		
	InsectEPGSmallTrain	0.1347	0.245	0.1723	1962.4	0.0296	0.0643	0.127	1329.9		
	InsectWingbeatSound	0.4121	0.4066	0.4066	1536.2	0.3934	0.3914	0.3914	1450.7		
	ItalyPowerDemand	0.9613	0.9611	0.9611	808.9	0.9622	0.9621	0.9621	1168		
	LargeKitchenAppliances	0.9077	0.904	0.904	844.6	0.9042	0.9013	0.9013	1528.4		
Lightning2	0.7211	0.7213	0.7154	368	0.7368	0.7377	0.7332	363.7			
Lightning7	0.8531	0.8493	0.8524	347	0.8486	0.8219	0.8223	360.1			

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
FCN	Mallat	0.9719	0.9667	0.9668	3035	0.9703	0.9642	0.9643	4035.7		
	Meat	0.9333	0.9167	0.9167	328.2	0.9175	0.9	0.9	282.8		
	MedicalImages	0.7724	0.775	0.7575	762.3	0.7769	0.7816	0.7563	684		
	MiddlePhalanxOutlineAgeGroup	0.4961	0.5455	0.4683	547.5	0.4503	0.5195	0.4427	438.2		
	MiddlePhalanxOutlineCorrect	0.7831	0.7869	0.7807	840.1	0.7945	0.7973	0.7897	693.3		
	MiddlePhalanxTW	0.3617	0.5	0.3688	581.5	0.3926	0.5	0.3741	480.3		
	MixedShapesRegularTrain	0.9548	0.9555	0.9558	2307.2	0.9541	0.9542	0.9541	5062.3		
	MixedShapesSmallTrain	0.9041	0.8981	0.8937	1954.7	0.0296	0.0643	0.127	1329.9		
	MoteStrain	0.9384	0.9385	0.9378	2353.5	0.9325	0.9321	0.9309	2549		
	NonInvasiveFetalECGThorax1	0.9562	0.9567	0.9557	4191.7	0.9534	0.9537	0.9525	7433.2		
NonInvasiveFetalECGThorax2	0.951	0.9542	0.9505	4026.7	0.9492	0.9537	0.9496	7455.2			
OliveOil	0.8205	0.7667	0.6639	361	0.497	0.7	0.5514	261.8			
OSULeaf	0.9874	0.9835	0.985	729.7	0.9874	0.9835	0.9814	668			
PhalangesOutlinesCorrect	0.822	0.8263	0.8073	3295.7	0.8212	0.8275	0.8115	1862			
Phoneme	0.2481	0.3339	0.2264	1516	0.2534	0.3286	0.222	3347.7			
PigAirwayPressure	0.2738	0.2356	0.2356	612.2	0.1852	0.1923	0.1923	530			
PigArtPressure	0.9879	0.9808	0.9808	586.3	0.9879	0.9808	0.9808	522.5			

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
FCN	PigCVP	0.8716	0.8365	0.8365	580.6	0.8793	0.8365	0.8365	525		
	Plane	1	1	1	278.5	1	1	1	259.2		
	PowerCons	0.8511	0.85	0.85	425	0.846	0.8444	0.8444	277.5		
	ProximalPhalanxOutlineAgeGroup	0.7282	0.8195	0.7523	561.3	0.7432	0.8293	0.7757	330.2		
	ProximalPhalanxOutlineCorrect	0.8886	0.8969	0.8691	852.5	0.9137	0.9175	0.8929	585.1		
	ProximalPhalanxTW	0.4408	0.7366	0.4803	563	0.5471	0.761	0.5114	328.4		
	RefrigerationDevices	0.5192	0.5147	0.5147	849.2	0.5196	0.52	0.52	642.2		
	Rock	0.5691	0.62	0.5675	434	0.5591	0.54	0.5536	263.6		
	ScreenType	0.6023	0.6027	0.6027	839.6	0.638	0.6347	0.6347	643.6		
	SemgHandGenderCh2	0.8325	0.825	0.7753	1167.4	0.8117	0.8267	0.8029	968.3		
SemgHandMovementCh2	0.5593	0.4378	0.4378	1341	0.532	0.4911	0.4911	3485			
SemgHandsSubjectCh2	0.7655	0.72	0.72	1334.7	0.7872	0.7667	0.7667	3496.5			
ShapeletSim	0.8261	0.7333	0.7333	819.2	0.8535	0.85	0.85	566.1			
ShapesAll	0.9042	0.8967	0.8967	1350.5	0.901	0.8933	0.8933	1993.8			
SmallKitchenAppliances	0.776	0.776	0.776	849.9	0.7608	0.7547	0.7547	580.5			
SmoothSubspace	0.9811	0.98	0.98	339.1	0.9742	0.9733	0.9733	130.7			
SonyAIBORobotSurfaceI	0.9433	0.9451	0.9519	2212.8	0.9574	0.9601	0.965	1167.7			

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
	SonyAIBORobotsSurface2	0.9725	0.9748	0.9744	3037.4	0.9766	0.9801	0.9818	1792		
	StarlightCurves	0.9586	0.9666	0.9449	6469.2	0	0	0	0		
	Strawberry	0.9691	0.9757	0.9794	1184	0.9743	0.9757	0.9727	521.6		
	SwedishLeaf	0.9662	0.9648	0.9653	925.7	0.9702	0.9696	0.9698	479.7		
	Symbols	0.9706	0.9698	0.9693	2392.2	0.969	0.9688	0.9685	2232.4		
	SyntheticControl	0.9867	0.9867	0.9867	673.9	0.9867	0.9867	0.9867	289.9		
	ToeSegmentation1	0.9695	0.9693	0.969	463	0.9603	0.9605	0.9616	277.5		
	ToeSegmentation2	0.7917	0.8769	0.8439	430.1	0.8128	0.8923	0.8856	299.1		
	Trace	1	1	1	341.1	1	1	1	151.2		
	TwoLeadECG	1	1	1	2164	0.9991	0.9991	0.9991	2377.1		
FCN	TwoPatterns	0.872	0.8715	0.8719	3143.7	0.8701	0.8695	0.87	1338.9		
	UMD	0.9863	0.9861	0.9861	645.1	0.9932	0.9931	0.9931	429.5		
	UWaveGestureLibraryAll	0.8208	0.8205	0.8212	3585.9	0.8153	0.8143	0.815	8072.6		
	UWaveGestureLibraryX	0.7454	0.7521	0.75	2804.4	0.751	0.7566	0.7542	3644.7		
	UWaveGestureLibraryY	0.6373	0.636	0.6372	2791.4	0.6367	0.6374	0.6385	1706		
	UWaveGestureLibraryZ	0.7226	0.7256	0.7273	2960.9	0.7281	0.7312	0.7328	1691.1		
	Wafer	0.9985	0.9974	0.988	3881	0.9974	0.9966	0.9849	1905.2		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
FCN	Wine	0.25	0.5	0.5	280.8	0.6607	0.6111	0.6111	191.1		
	WordsSynonyms	0.5144	0.5596	0.418	694.3	0.5002	0.5455	0.3962	411.5		
	Worms	0.8094	0.8052	0.7828	408.7	0.8316	0.7922	0.7581	407.9		
	WormsTwoClass	0.7483	0.7532	0.75	424.5	0.7304	0.7273	0.7348	408.8		
	Yoga	0.8397	0.8407	0.8406	1725	0.8449	0.844	0.8413	1358.1		
	ACSF1	0.9306	0.93	0.93	560.6	0.9207	0.92	0.92	459.2		
	Adiac	0.8571	0.8517	0.8559	1492.4	0.8666	0.8465	0.8509	831.3		
	ArrowHead	0.8629	0.8629	0.8666	867.3	0.8471	0.84	0.8472	447.2		
	Beef	0.7417	0.7333	0.7333	536.4	0.785	0.7333	0.7333	185.3		
	BeetleFly	0.8846	0.85	0.85	427.1	0.9167	0.9	0.9	181.1		
ResNet	BirdChicken	0.9545	0.95	0.95	501.5	0.8535	0.85	0.85	336.3		
	BME	1	1	1	704.8	1	1	1	521.8		
	Car	0.9107	0.9	0.8968	690.9	0.9208	0.9167	0.916	217.2		
	CBF	0.9923	0.9922	0.9923	2666.9	0.9945	0.9944	0.9945	1631.2		
	Chinatown	0.9629	0.9767	0.9806	2486.2	0.97	0.9825	0.988	1782.4		
	ChlorineConcentration	0.8277	0.8458	0.8183	2614.4	0.8294	0.844	0.8154	915		
	CinCECGTorso	0.8217	0.8152	0.8152	3076.5	0.8349	0.8261	0.8259	5549.4		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
	Coffee	1	1	1	704.6	1	1	1	340.2		
	Computers	0.7947	0.792	0.792	1158.7	0.829	0.828	0.828	1561.2		
	CricketX	0.8167	0.8103	0.8187	1058.2	0.8079	0.7692	0.782	1136.8		
	CricketY	0.8226	0.8231	0.8256	1097.7	0.6922	0.6128	0.605	1128.8		
	CricketZ	0.7983	0.8	0.7928	1103	0.786	0.7949	0.7841	1145.2		
	Crop	0.7353	0.7427	0.7427	23504.2	0.7361	0.7423	0.7423	6593.1		
	DiatomSizeReduction	0.0752	0.3007	0.25	3136.6	0.0752	0.3007	0.25	2214.7		
	DistalPhalanxOutlineAgeGroup	0.7564	0.7626	0.7898	1085.7	0.719	0.7554	0.7815	480.3		
	DistalPhalanxOutlineCorrect	0.7668	0.7717	0.7596	1652.9	0.7715	0.7717	0.7534	589.5		
	DistalPhalanxTW	0.5001	0.6547	0.511	1112.4	0.5221	0.6835	0.5327	474.1		
ResNet	Earthquakes	0.4534	0.6906	0.4805	811.3	0.5831	0.7266	0.5424	1306.5		
	ECG200	0.8936	0.89	0.8655	699.7	0.8097	0.82	0.7925	306		
	ECG5000	0.6766	0.9336	0.5751	4326	0.6644	0.9391	0.5421	791.8		
	ECGFiveDays	0.9931	0.993	0.9931	3405.4	0.9931	0.993	0.9931	3405.4		
	ElectricDevices	0.7029	0.7314	0.6419	22626.9	0.6924	0.7201	0.628	8957		
	EOGHorizontalSignal	0.6587	0.6133	0.6147	1433.4	0.6194	0.5718	0.573	1048.4		
	EOGVerticalSignal	0.4434	0.4392	0.4408	1429.6	0.4316	0.4309	0.432	3497.7		

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm				Thực nghiệm của bản thân			
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT
ResNet	EthanolLevel	0.7974	0.79	0.7896	2389.4	0.7042	0.678	0.678	1732.9
	FaceAll	0.8711	0.8284	0.9159	2594.4	0.8732	0.8367	0.9198	1281.2
	FaceFour	0.9667	0.9545	0.958	847	0.9667	0.9545	0.958	315.3
	FacesUCR	0.9476	0.9556	0.93	1931.1	0.9502	0.9517	0.9295	760.2
	FiftyWords	0.6489	0.756	0.6369	1947	0.5842	0.7231	0.5731	421.6
	Fish	0.9764	0.9771	0.9783	531.7	0.9821	0.9829	0.9834	372
	FordA	0.9309	0.9311	0.9313	8473.7	0.9408	0.9409	0.9411	3389
	FordB	0.8173	0.8173	0.8172	8686.3	0.8126	0.8123	0.8122	3300.9
	FreezerRegularTrain	0.9989	0.9989	0.9989	1743.1	0.9986	0.9986	0.9986	2598
	FreezerSmallTrain	0.7815	0.7126	0.7126	10251.7	0.8877	0.8775	0.8775	4851.7
ResNet	Fungi	0.1166	0.2849	0.2435	2075.9	0.0883	0.2151	0.2222	1572.3
	GunPoint	0.9935	0.9933	0.9932	929.1	0.9935	0.9933	0.9932	363.1
	GunPointAgeSpan	0.9968	0.9968	0.9969	590.2	0.9937	0.9937	0.9937	345.3
	GunPointMaleVersusFemale	0.9903	0.9905	0.9906	582.5	0.9934	0.9937	0.994	339.5
	GunPointOldVersusYoung	0.9876	0.9873	0.987	928.7	0.9911	0.9905	0.99	342.6
	Ham	0.7525	0.7524	0.7527	804.9	0.7144	0.7143	0.7146	302
	HandOutlines	0.9419	0.9432	0.9343	5623.2	0.9332	0.9324	0.9192	4151.4

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
	Haptics	0.5443	0.5292	0.5288	758.9	0.4976	0.4903	0.4882	656.1		
	Herring	0.6239	0.6406	0.6002	671.3	0.6584	0.6719	0.6447	281.5		
	HouseTwenty	0.9859	0.9832	0.98	660.9	0.9859	0.9832	0.98	1266.8		
	InlineSkate	0.394	0.3727	0.3795	1263.7	0.385	0.3709	0.3803	3300		
	InsectEPGRegularTrain	1	1	1	687.4	1	1	1	1019.9		
	InsectEPGSmallTrain	0.523	0.4699	0.3815	2482.1	0.158	0.4739	0.3333	2010.2		
	InsectWingbeatSound	0.5039	0.5071	0.5071	1402.6	0.5026	0.5035	0.5035	604.9		
	ItalyPowerDemand	0.9662	0.966	0.966	1750.7	0.968	0.9679	0.9679	1752.5		
	LargeKitchenAppliances	0.8955	0.8933	0.8933	1182.4	0.9014	0.8987	0.8987	2129.7		
	Lightning2	0.7211	0.7213	0.7154	456.9	0.7689	0.7705	0.7689	252.3		
ResNet	Lightning7	0.8832	0.8493	0.8403	431	0.8115	0.7945	0.7917	214.9		
	Mallat	0.9755	0.9727	0.9727	4012.9	0.9729	0.9701	0.9702	2138.4		
	Meat	1	1	1	419.8	0.9841	0.9833	0.9833	441.5		
	MedicalImages	0.7806	0.7842	0.7776	1223	0.7495	0.7553	0.7706	321		
	MiddlePhalanxOutlineAgeGroup	0.5012	0.5649	0.4849	1124	0.5348	0.5974	0.5064	473		
	MiddlePhalanxOutlineCorrect	0.8333	0.8351	0.8287	1430.9	0.8506	0.8488	0.8398	250.9		
	MiddlePhalanxTW	0.3966	0.5195	0.3809	1111.6	0.3323	0.4675	0.3374	263		

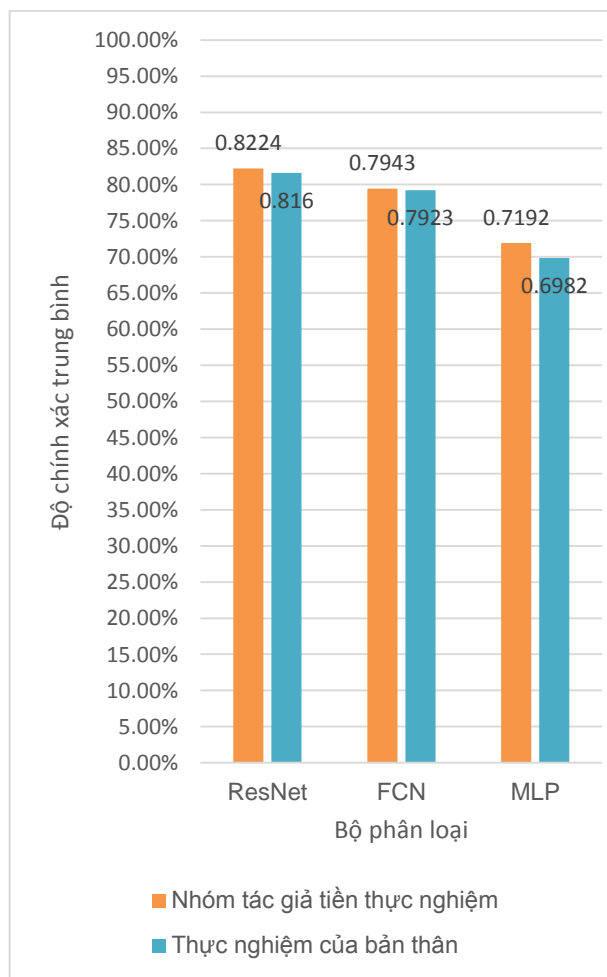
Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
ResNet	MixedShapesRegularTrain	0.9704	0.9707	0.9704	4105.2	0.9737	0.974	0.9735	1949.2		
	MixedShapesSmallTrain	0.9188	0.9204	0.9195	3496.1	0.9116	0.9142	0.9143	2250		
	MotesTrain	0.9282	0.9257	0.9231	4772.8	0.9279	0.9257	0.9232	2355.8		
	NonInvasiveFetalECGThorax1	0.9398	0.941	0.9402	5374.6	0.9546	0.9557	0.9547	2825.2		
	NonInvasiveFetalECGThorax2	0.9402	0.9435	0.94	5567.2	0.952	0.9552	0.9515	2826.4		
	OliveOil	0.899	0.8333	0.7417	535	0.7625	0.8333	0.7639	192.8		
	OSULeaf	0.9861	0.9793	0.9745	1144.5	0.985	0.9793	0.9779	365.3		
	PhalangesOutlinesCorrect	0.846	0.845	0.8236	4373.7	0.8334	0.8345	0.8134	647		
	Phoneme	0.243	0.3397	0.2273	2242.2	0.2451	0.3207	0.2252	5110		
	PigAirwayPressure	0.5094	0.4327	0.4327	879.6	0.4413	0.3894	0.3894	2256		
PigArtPressure	0.9923	0.9904	0.9904	840.6	0.9923	0.9904	0.9904	703.4			
PigCVP	0.9279	0.9087	0.9087	841.1	0.95	0.9375	0.9375	706			
Plane	1	1	1	566.3	1	1	1	379.5			
PowerCons	0.8845	0.8833	0.8833	499.7	0.8938	0.8889	0.8889	206.7			
ProximalPhalanxOutlineAgeGroup	0.769	0.8488	0.8369	1120.4	0.765	0.8488	0.8062	222.7			
ProximalPhalanxOutlineCorrect	0.9168	0.9244	0.9067	1727.3	0.9082	0.9107	0.8821	240.5			
ProximalPhalanxTW	0.4769	0.7659	0.502	1155.7	0.5492	0.7902	0.5366	221.4			

Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiên thực nghiệm				Thực nghiệm của bản thân			
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT
	RefrigerationDevices	0.5383	0.5307	0.5307	1150.4	0.5314	0.528	0.528	692.8
	Rock	0.6125	0.58	0.5516	557.6	0.6278	0.6	0.6032	323.4
	ScreenType	0.6128	0.608	0.608	1139.8	0.6347	0.6347	0.6347	692.2
	SengHandGenderCh2	0.804	0.8183	0.7899	1678.4	0.7497	0.765	0.7698	1113.1
	SengHandMovementCh2	0.4732	0.4622	0.4622	2099.6	0.4888	0.4844	0.4844	1322.2
	SengHandSubjectCh2	0.7499	0.7378	0.7378	1957.8	0.7264	0.7133	0.7133	1325.9
	ShapeletSim	0.9612	0.9611	0.9611	1202.4	0.8167	0.8111	0.8111	447.8
	ShapesAll	0.9373	0.93	0.93	1733.9	0.9181	0.915	0.915	740.4
	SmallKitchenAppliances	0.7869	0.7867	0.7867	1132.7	0.7701	0.7707	0.7707	698.5
	SmoothSubspace	0.9811	0.98	0.98	443.3	0.9804	0.98	0.98	186.9
ResNet	SonyAIBORobotsSurface1	0.9526	0.9551	0.9606	2349	0.9403	0.9418	0.949	1537.9
	SonyAIBORobotsSurface2	0.972	0.9664	0.9577	3637.3	0.9779	0.9759	0.9711	2361.3
	StarLightCurves	0.9626	0.9707	0.9516	9649.3	0.9677	0.9728	0.9521	5049.6
	Strawberry	0.9702	0.9757	0.9777	2495.5	0.9714	0.9757	0.976	477.5
	SwedishLeaf	0.957	0.9552	0.9559	1805.4	0.9703	0.9696	0.9704	317.6
	Symbols	0.8777	0.8623	0.8623	4616.7	0.8322	0.791	0.7958	2547.4
	SyntheticControl	0.9967	0.9967	0.9967	784.9	0.9936	0.9933	0.9933	265.4

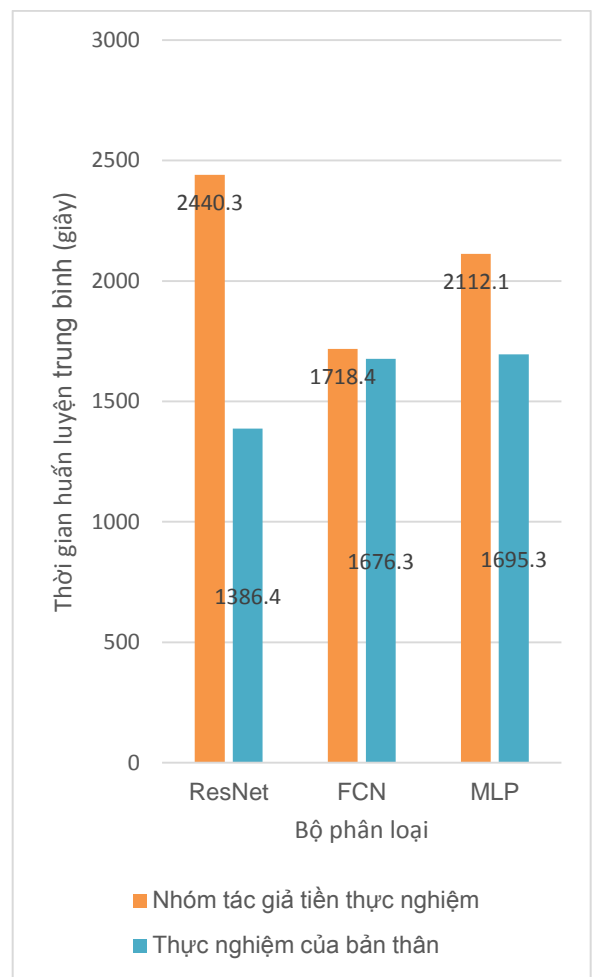
Tên phân loại	Tên tập dữ liệu	Nhóm tác giả tiền thực nghiệm					Thực nghiệm của bản thân				
		Pre	Acc	Rec	TrT	Pre	Acc	Rec	TrT		
ResNet	ToeSegmentation1	0.9602	0.9605	0.9611	781.2	0.9603	0.9605	0.9616	595.4		
	ToeSegmentation2	0.8038	0.8846	0.897	711.8	0.7711	0.8538	0.8781	635.7		
	Trace	1	1	1	426.7	1	1	1	248.8		
	TwoLeadECG	1	1	1	4518.8	1	1	1	3363.9		
	TwoPatterns	0.9997	0.9998	0.9998	3752.2	0.9774	0.9773	0.9774	1020.6		
	UMD	0.9932	0.9931	0.9931	761.4	0.9863	0.9861	0.9861	512.8		
	UWaveGestureLibraryAll	0.8623	0.8618	0.8623	5597.2	0.8502	0.8498	0.8504	4839.4		
	UWaveGestureLibraryX	0.7781	0.7845	0.7825	5454.1	0.7616	0.7677	0.7658	1841.6		
	UWaveGestureLibraryY	0.657	0.6561	0.6571	3709.5	0.6559	0.6569	0.6578	1820.1		
	UWaveGestureLibraryZ	0.7427	0.7479	0.7498	3589.4	0.7522	0.7566	0.7586	1834.6		
ResNet	Wafer	0.9992	0.9985	0.9932	4941.1	0.9989	0.9981	0.991	1412.5		
	Wine	0.8701	0.8519	0.8519	584.3	0.8375	0.8333	0.8333	299.5		
	WordsSynonyms	0.5591	0.6301	0.4739	948.1	0.5592	0.6332	0.4882	503.1		
	Worms	0.7191	0.7013	0.664	550.5	0.7847	0.7922	0.7627	641.6		
	WormsTwoClass	0.7357	0.7273	0.7386	562.8	0.7357	0.7403	0.7386	649.7		
	Yoga	0.8656	0.8643	0.8617	2221.6	0.8832	0.8837	0.8828	1037.7		

2.3 Trực quan hóa và đánh giá kết quả

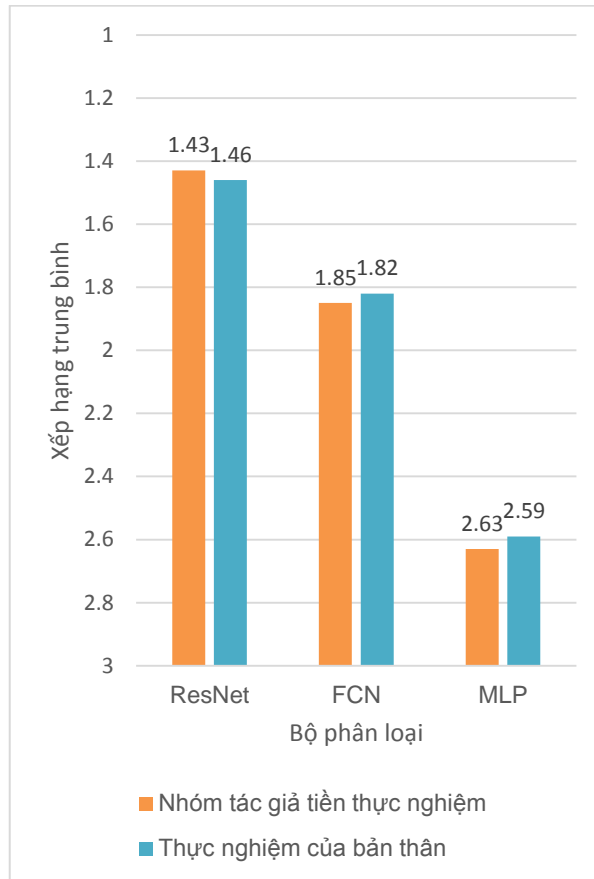
Dựa vào các biểu đồ sau, nhìn chung ta thấy kết quả thực nghiệm lại so với kết quả tiền thực nghiệm không có quá nhiều khác biệt (sai số trung bình ở độ chính xác là $9.8 \cdot 10^{-3}$). ResNet vẫn là kiến trúc có hiệu suất cao nhất, theo sau đó là FCN và MLP. Bên cạnh đó, kết quả cũng có sự khác biệt tương đối về thời gian huấn luyện trung bình giữa hai lần thực nghiệm với mạng ResNet (thời gian huấn luyện tiền thực nghiệm lâu hơn nhiều so với thực nghiệm sau, khoảng 0.5 giờ).



Biểu đồ 1. Biểu đồ thể hiện độ chính xác trung bình của ba kiến trúc MLP, FCN, ResNet trên tất cả các tập dữ liệu thực nghiệm



Biểu đồ 2. Biểu đồ thể hiện thời gian huấn luyện trung bình của ba kiến trúc MLP, FCN, ResNet trên tất cả các tập dữ liệu thực nghiệm.

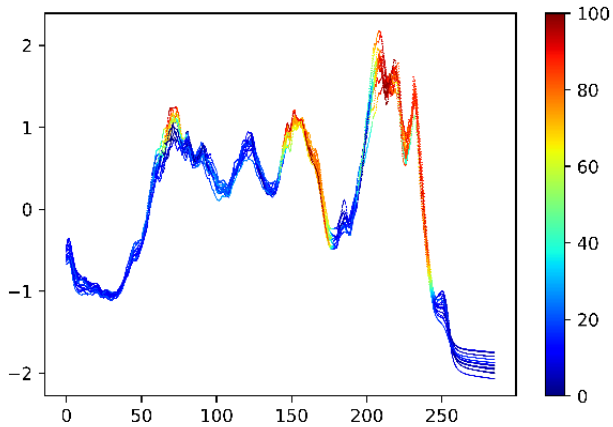


Biểu đồ 3. Biểu đồ thể hiện xếp hạng trung bình dựa trên độ chính xác của ba kiến trúc MLP, FCN, ResNet trên tất cả các tập dữ liệu thực nghiệm.

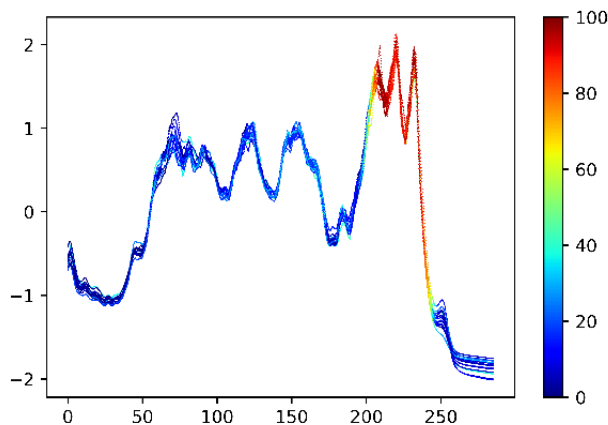
Nguyên nhân chủ yếu là do sử dụng phần cứng chưa hoàn toàn giống nhau (tiền thực nghiệm sử dụng nhiều loại GPU khác nhau như GTX 1080 Ti, Tesla K20, K40 và K80, trong khi thực nghiệm lại chỉ sử dụng một loại GPU duy nhất đó là Tesla K80).

Ngoài ra, chúng ta cũng sử dụng Class Activation Map (CAM) để làm nổi bật các thành phần đóng góp nhiều nhất cho một nhận dạng lớp nhất định trên các mẫu dữ liệu thuộc lớp đó. Ví dụ như trong hai hình dưới, chúng ta có thể thấy rõ những vị trí được mô tả bởi các phần màu xanh là các khu vực không phân biệt về đạo hàm trong chuỗi thời gian mà mạng ResNet không tập trung vào khi đưa ra quyết định. Các khu

vực có tính phân biệt cao (khu vực màu đỏ và màu vàng) là nơi mà mạng ResNet sẽ tập trung vào khi đưa ra quyết định.



Hình 27. CAM của tập Coffee lớp 0.



Hình 28. CAM của tập Coffee lớp 1.

Qua thực nghiệm trên, các mô hình học sâu nói chung và mô hình ResNet nói riêng có hiệu suất tương đối tốt, thậm chí có thể cạnh tranh với cả phương pháp phân loại đỉnh cao hiện nay là HIVE – COTE. Đặc biệt, các mô hình học sâu trong đó có ResNet còn có ưu điểm vượt trội hơn, đó là thời gian huấn luyện và đưa ra dự đoán của mạng nhanh hơn so với HIVE – COTE. Nếu tập trung nghiên cứu phát triển các mô hình học sâu nhằm cải thiện hiệu suất trong vấn đề phân loại chuỗi thời gian có thể hứa hẹn sẽ có nhiều bước tiến vượt bậc.

CHƯƠNG 4 – KẾT LUẬN

4.1 Tổng kết

Trong báo cáo này, chúng ta đã trình bày những khái niệm cơ bản nhất về mạng nơ ron sâu, các thành phần đặc trưng của mạng nơ ron sâu, những đặc điểm riêng của mạng nơ ron sâu dành cho việc phân loại chuỗi thời gian; các kiến trúc học sâu hiệu quả để giải quyết bài toán phân loại chuỗi thời gian. Chúng ta cũng đã tiến hành thực nghiệm so sánh kết quả với tiền thực nghiệm của các tác giả đi trước và có đã có nhận xét về việc sử dụng kiến trúc MLP, FCN và ResNet để phân loại một số tập dữ liệu chuỗi thời gian phổ biến trong dữ liệu chuỗi thời gian đơn biến UCR, qua đó thấy được hiệu quả của các phương pháp này trong nhiệm vụ phân loại chuỗi thời gian.

4.2 Hướng phát triển

Mặc dù chúng ta đã tiến hành đánh giá thử nghiệm sâu rộng, nhưng việc ứng dụng học sâu để phân loại chuỗi thời gian vẫn còn thiếu nhiều nghiên cứu kỹ lưỡng. Hơn nữa, ảnh hưởng của chuẩn hóa z (và các phương pháp chuẩn hóa khác) đối với khả năng học tập của DNN cũng cần được tìm hiểu kỹ. Tương lai, chúng ta sẽ làm rõ vấn đề trên bằng cách thực hiện các thử nghiệm sâu rộng hơn, đặc biệt là trên các bộ dữ liệu chuỗi thời gian đa biến. Bên cạnh đó, một thách thức quan trọng đối với bài toán TSC là cần một bộ dữ liệu chung có nhãn lớn và đầy đủ để thực hiện các thực nghiệm chính xác hơn và đầy đủ hơn.

Tóm lại, với việc chúng ta các kho khai thác dữ liệu cập nhật trở nên thường xuyên hơn, tận dụng các kiến trúc sâu hơn như đã trình bày, chúng ta có thể khiến học sâu trở thành một cách tiếp cận để giải quyết vấn đề một cách lôi cuốn hơn.

TÀI LIỆU THAM KHẢO

Tiếng Anh

- [1] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller, *Deep learning for time series classification: A review*, 2019
- [2] Zhiguang Wang, Weizhong Yan, Tim Oates, *Time series classification from scratch with deep neural networks: A strong baseline*, 2017
- [3] Sergey Ioffe, Christian Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, 2015
- [4] Min Lin, Qiang Chen, Shuicheng Yan, *Network in Network*, 2014
- [5] Abien Fred Agarap, *Deep Learning using Rectified Linear Units (ReLU)*, 2019
- [6] Wikipedia, *Softmax function*,
https://en.wikipedia.org/wiki/Softmax_function
- [7] Margarita Granat, *How to Use Convolutional Neural Networks for Time Series Classification*, 2019
<https://towardsdatascience.com/how-to-use-convolutional-neural-networks-for-time-series-classification-56b1b0a07a57>
- [8] Amaia Abanda, Usue Mori, Jose A. Lozano, *A review on distance based time series classification*, 2018
- [9] Lines J, Taylor S, Bagnall A, *HIVE-COTE: The hierarchical vote collective of transformation-based ensembles for time series classification*, 2016
- [10] Michael Nielsen, *Neural Networks and Deep Learning*, 2015
- [11] Pierre Baldi, *Autoencoders, Unsupervised Learning, and Deep Architectures*, 2012
- [12] Joan Serra, Santiago Pascual và Alexandros Karatzoglou, *Towards a Universal Neural Network Encoder for Time Series*, 2018

- [13] Cui Z, Chen W, Chen Y, *Multi-scale convolutional neural networks for time series classification*, 2016
- [14] Le Guennec A, Malinowski S, Tavenard R, *Data augmentation for time series classification using convolutional neural networks*, 2016
- [15] Zheng Y, Liu Q, Chen E, Ge Y, Zhao JL, *Time series classification using multi-channels deep convolutional neural networks*, 2014
- [16] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu, *Convolutional neural networks for time series classification*, 2017
- [17] Pattreeya Tanisaro and Gunther Heidemann, *Time Series Classification Using Time Warping Invariant Echo State Networks*, 2016