**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Phương pháp dự báo ảnh mưa kết hợp kiến trúc UNet và PredRNN cho ảnh radar vệ tinh chuỗi thời gian**

**Đỗ Ngọc Tuyền**

Tuyen.dn164442@sis.hust.edu.vn

**Trường Công nghệ Thông tin và Truyền thông**

**Khoa Khoa học máy tính**

**Ngành Hệ thống thông tin**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | PGS. TS. Phạm Văn Hải  Chữ ký của GVHD |
| **Khoa:** | Khoa học máy tính |
| **Ngành:** | Hệ thống thông tin – Kỹ Sư |
| **HÀ NỘI, 2/2022** | |

**Lời cảm ơn**

Đối với bản thân em, quãng thời gian 5 năm sinh viên ở Đại học Bách Khoa Hà Nội thực sự là một khoảng thời gian đáng nhớ trong cuộc đời. Tại đây, em đã được học tập, làm việc ở một trong những môi trường tốt nhất, có được những người bạn mới, và hơn hết là được làm việc dưới sự chỉ bảo tận tình của các thầy cô thuộc Viện Công nghệ thông tin và truyền thông. Em xin gửi lời cảm ơn trân trọng đến các thầy, các cô. Em xin chúc các thầy, các cô luôn luôn mạnh khỏe, ngày một thành công trong sự nghiệp nghiên cứu và giảng dạy của mình.

Em xin gửi lời cảm ơn sâu sắc tới PGS. TS. Phạm Văn Hải, cảm ơn thầy đã hướng dẫn em tận tình trong suốt thời gian qua. Dưới sự chỉ bảo của thầy, em đã học hỏi thêm được nhiều kiến thức hữu ích cũng như các kinh nghiệm làm nghiên cứu. Em cũng xin gửi lời cảm ơn tới tập thể phòng nghiên cứu của thầy PGS. TS. Lê Hoàng Sơn tại Trường Đại học Công nghệ - Đại học Quốc Gia Hà Nội. Tại đây, em có được những điều kiện tốt nhất để phát triển bản thân, được tìm hiểu, nghiên cứu các thuật toán trong học máy, có những người thầy, người anh cùng chung chí hướng và tài năng, và đặc biệt là có được sự hướng dẫn tận tình của PGS. TS. Lê Hoàng Sơn và TS. Trần Mạnh Tuấn. Các thầy không chỉ truyền đạt cho em những kiến thức quý báu, mà còn là những người chia sẻ, giúp đỡ em những lúc em gặp khó khăn trong quá trình làm việc. Em xin gửi lời cảm ơn chân thành đến hai thầy. Bên cạnh đó, em cũng đặc biệt cảm ơn ThS. Cù Nguyên Giáp, TS. Phạm Huy Thông, TS. Nguyễn Thọ Thông, ThS. Lê Minh Tuấn, TS. Lương Hồng Lan đã giúp đỡ em trong thời gian sinh hoạt tại phòng nghiên cứu. Em cũng xin cảm ơn các anh chị tại Viện nghiên cứu và ứng dụng CMC – CIST, cảm ơn chị Quỳnh đã là người nâng đỡ em trong thời gian em ở Viện giúp em có một khởi đầu, bước đệm tuyệt vời trong tương lai. Tại đây em cũng cảm ơn thầy TS. Đặng Minh Tuấn Vietkey và Phó trưởng viện Lê Chí Dũng đã là người đi trước, chia sẻ và dạy em nhiều điều bổ ích. Cảm ơn anh Đồng, anh Trường đã chia sẻ giúp em học hỏi từ những kinh nghiệm đi trước của các anh rất nhiều.

Cuối cùng, con muốn được gửi lời cảm ơn đến bố, mẹ, gia đình luôn bên cạnh, ủng hộ con trong suốt những tháng năm qua. Cảm ơn hai người em thân thiết đã ở cạnh, đồng hành Tú, Hưng, Khang và những người bạn tuyệt vời Kiều, Bình, Hoàng, Hiếu và Hà đã luôn sát cánh bên mình. Cảm ơn các anh hem đồng hội cấp 2, cấp 3, đại học đã cùng tôi trưởng thành.

Sinh viên thực hiện  
ký và ghi rõ họ tên

**Tóm tắt nội dung đồ án**

Dự báo ảnh mưa là một trong những nhiệm vụ chính của dự báo thời tiết nhằm dự báo các sự kiện mưa một cách chính xác ngay cả ở những vùng có lượng mưa thấp. Dự báo ảnh mưa có thể thực hiện bằng nhiều cách sử dụng các công nghệ hiện đại như cảm biến, trạm quan sát. Hiện nay, một trong những cách tiếp cận phổ biến nhất được áp dụng đó chính là sử dụng các mô hình Học máy và Học sâu. Bằng cách áp dụng các mô hình thị giác máy tính, các tập dữ liệu ảnh radar chuỗi thời gian được xử lý và dự báo cho hình ảnh trực quan nhất về vùng có mưa tại thời điểm quan sát. Dự báo chính xác vùng mưa có ý nghĩa quan trọng trong dự báo thời tiết giúp cho người dân có hình dung rõ ràng nhất về hiện tượng mưa, giúp họ có một lịch trình phù hợp (trong giao thông, nông nghiệp, nghỉ dưỡng, lịch trình, …). Sau khi nghiên cứu các kết quả liên quan, hiện nay có ít nghiên cứu giải quyết các vấn đề liên quan đến việc dự đoán hình ảnh tiếng vọng radar thời tiết chuỗi thời gian sử dụng công nghệ học sâu trong thời gian hợp lý. Đặc biệt, các nghiên cứu cũ không tận dụng được các đặc trưng thời gian và không gian dài hạn, và các đặc trưng ngắn hạn của hình ảnh chuỗi thời gian. Các nghiên cứu chủ yếu tập trung vào các đặc trưng của cảm biến, có ít các mô hình có thể tận dụng được đặc trưng không gian và thời gian của bộ dữ liệu.

Đồ án này đề xuất một mô hình tiếp cận mới có tên là RainPredRNN, mô hình là sự kết hợp của mô hình Học sâu phân đoạn UNet và mô hình xử lý ảnh chuỗi thời gian PredRNN\_v2 cho dự báo ảnh mưa với dữ liệu hình ảnh tiếng vọng của Radar thời tiết chuỗi thời gian. Hình ảnh đầu vào là một chuỗi ảnh liên tiếp trong quá khứ, mỗi ảnh sẽ tương ứng với một mốc thời gian trong quá khứ, mô hình sẽ xử lý và cho ra chuỗi ảnh dự báo liên tiếp trong tương lai. Bằng cách tận dụng các ưu điểm của mô hình UNet (phần mã hóa và giải mã), số lượng các phép toán phải thực hiện của mô hình đề xuất RainPredRNN được giảm đi đáng kể. Điều này có nghĩa là thời gian xử lý khi huấn luyện và dự báo cũng được giảm đi. Các hình ảnh đầu vào được mã hóa, giảm chiều không gian giúp cho số lượng phép tính trong phần chính của mô hình được giảm đi đáng kể. Sau khi được xử lý và cho ra ảnh dự báo, ảnh này sẽ được giải mã bằng phần giải mã của mạng UNet nhằm cho ra kết quả có kích thước ban đầu. Điều này làm giảm thời gian xử lý của mô hình tổng thể trong khi vẫn duy trì sai số hợp lý trong các hình ảnh dự đoán. Để kiểm chứng các mô hình, thực nghiệm sẽ được thực hiện trên bộ dữ liệu thực thu thập được tại trạm radar thời tiết Pha Đin, đặt tại tỉnh Điện Biên, Việt Nam. Thử nghiệm sẽ sử dụng các độ đo đánh giá chất lượng như Sai số tuyệt đối trung bình (MAE), Chỉ số tương đồng về cấu trúc (SSIM) và Critical Success Index (CSI). Thực nghiệm đã cho thấy mô hình đề xuất đã cho ra kết quả tốt, các độ đo MAE, SSIM, CSI có giá trị lần lượt vào khoảng 0,43, 0,95, 0,94. Thời gian huấn luyện của mô hình đề xuất giảm còn 30% thời gian huấn luyện so với các mô hình cũ. Thời gian tới, mô hình sẽ được tiếp tục hiệu chỉnh và đánh giá để có thể cải thiện được kết quả hiện tại.

**Phiếu giao nhiệm vụ đồ án tốt nghiệp**

**Thông tin sinh viên**

* **Họ và tên**: Đỗ Ngọc Tuyền
* **Email**: [tuyen.dn164442@sis.hust.edu.vn](mailto:tuyen.dn164442@sis.hust.edu.vn)
* **Điện thoại liên lạc**: 0339503366
* **Lớp**: CNTT2.01
* **Hệ đào tạo**: Đại học chính quy
* **Đồ án tốt nghiệp được thực hiện tại**: Bộ môn hệ thống thông tin - Viện công nghệ thông tin và truyền thông.
* **Thời gian làm đồ án tốt nghiệp**: Từ ngày 10/2021 đến 2/2022.

**Mục đích nội dung của đồ án tốt nghiệp**

Cải tiến thời gian huấn luyện và xử lý của mô hình học sâu PredRNN\_v2 cho bài toán dự báo ảnh mưa tiếng vọng radar.

**Các nhiệm vụ cụ thể của đồ án tốt nghiệp**

1. Nghiên cứu bài toán dự đoán ảnh chuỗi thời gian cho ảnh tiếng vọng radar.
2. Tìm hiểu và áp dụng cơ mã hóa và giải mã của Unet vào mạng PredRNN\_v2.
3. Thực nghiệm và đánh giá kết quả.

**Lời cam đoan của sinh viên**

Tôi – Đỗ Ngọc Tuyền - cam kết đồ án tốt nghiệp là sản phẩm của bản thân tôi dưới sự hướng dẫn của PGS. TS. Phạm Văn Hải.

Các kết quả trong đồ án là trung thực, không phải sao chép toàn văn của bất kì công trình nào khác.

Hà Nội, ngày 15 tháng 02 năm 2022

Sinh viên

Đỗ Ngọc Tuy ền

Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành và cho phép bảo vệ:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Hà Nội, ngày 15 tháng 02 năm 2022

Giảng viên hướng dẫn

PGS. TS. Phạm Văn Hải

**Mục lục**

[CHƯƠNG 1. BÀI TOÁN DỰ BÁO ẢNH MƯA 11](#_Toc95672433)

[1.1 Đặt vấn đề 11](#_Toc95672434)

[1.2 Ý nghĩa 12](#_Toc95672435)

[1.3 Tổng quan các nghiên cứu liên quan 12](#_Toc95672436)

[1.4 Nhận xét về các nghiên cứu liên quan 14](#_Toc95672437)

[1.5 Mục tiêu nghiên cứu 14](#_Toc95672438)

[1.6 Phương pháp nghiên cứu 14](#_Toc95672439)

[1.7 Phạm vi đồ án 14](#_Toc95672440)

[1.8 Dữ liệu nghiên cứu 14](#_Toc95672441)

[1.9 Tiêu chí đánh giá 16](#_Toc95672442)

[1.10 Bố cục đồ án 17](#_Toc95672443)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 18](#_Toc95672444)

[2.1 Mạng nơ-ron 18](#_Toc95672445)

[2.1.1 Hàm kích hoạt (Activation Function) 18](#_Toc95672446)

[2.1.2 Hàm mất mát Mean Squared Error (MSE) 20](#_Toc95672447)

[2.1.3 Lan truyền tiến (Feed Forward) 20](#_Toc95672448)

[2.1.4 Lan truyền ngược (Backpropagation) 21](#_Toc95672449)

[2.2 Mạng nơ-ron tích chập (Convolution Neural Network) 21](#_Toc95672450)

[2.3 Mạng nơ-ron hồi quy (Recurrent Neural Network) 23](#_Toc95672451)

[2.4 Long Short-Term Memory (LSTM) 25](#_Toc95672452)

[2.5 Convolution LSTM (Conv-LSTM) 27](#_Toc95672453)

[2.6 Spatiotemporal LSTM (ST-LSTM) với Memory Flow 28](#_Toc95672454)

[2.7 Spatiotemporal LSTM với Memory Decoupling 29](#_Toc95672455)

[CHƯƠNG 3. MÔ HÌNH ĐỀ XUẤT 31](#_Toc95672456)

[3.1 Ý tưởng 31](#_Toc95672457)

[3.2 Kiến trúc mạng UNet 31](#_Toc95672458)

[3.3 Kiến trúc tổng thể mô hình đề xuất 32](#_Toc95672459)

[3.4 Vai trò của phần mã hóa và giải mã trong mạng UNet 33](#_Toc95672460)

[3.5 Vai trò của PredRNN 34](#_Toc95672461)

[3.6 Thuật toán học 34](#_Toc95672462)

[CHƯƠNG 4. KẾT QUẢ THỬ NGIỆM VÀ ĐÁNH GIÁ 38](#_Toc95672463)

[4.1 Dữ liệu đánh giá 38](#_Toc95672464)

[4.2 Công cụ cài đặt 38](#_Toc95672465)

[4.2.1 Python 38](#_Toc95672466)

[4.2.2 PyTorch 39](#_Toc95672467)

[4.3 Cấu hình máy tính 41](#_Toc95672468)

[4.4 Cài đặt thử nghiệm và lựa chọn siêu tham số 41](#_Toc95672469)

[4.5 Kết quả và đánh giá 41](#_Toc95672470)

[4.6 Hình ảnh kết quả 42](#_Toc95672471)

[4.7 Demo ứng dụng dự báo ảnh mưa 44](#_Toc95672472)

[4.8 Nhận xét chung 45](#_Toc95672473)

[CHƯƠNG 5. KẾT LUẬN 46](#_Toc95672474)

[TÀI LIỆU THAM KHẢO 47](#_Toc95672475)

**Danh mục hình vẽ**

[Hình 1.1: Ứng dụng của Học sâu trong lĩnh vực dự báo thời tiết. 11](#_Toc95819315)

[Hình 1.2: Vị trí địa lý của trạm Pha Đin 15](#_Toc95819316)

[Hình 1.3: Mẫu hình ảnh phản xạ radar được ghi lại trong khoảng thời gian từ 17h50 đến 19h00 ngày 23 tháng 6. 16](#_Toc95819317)

[Hình 2.1: Đồ thị hàm sigmoid [32]. 19](#_Toc95819318)

[Hình 2.2: Đồ thị hàm Tanh [32]. 20](#_Toc95819319)

[Hình 2.3: Đồ thị hàm ReLU [32]. 20](#_Toc95819320)

[Hình 2.4: Mô hình tổng quát mạng nơ-ron tích chập [33]. 23](#_Toc95819321)

[Hình 2.5: Tầng tích chập hai chiều [33]. 23](#_Toc95819322)

[Hình 2.6: Tầng Max pooling (bên trái) và Average pooling (bên phải) [33]. 24](#_Toc95819323)

[Hình 2.7: Minh họa độ trượt [33]. 24](#_Toc95819324)

[Hình 2.8: Minh họa padding [33]. 24](#_Toc95819325)

[Hình 2.9: Một mạng nơ-ron hồi quy không có đầu ra đại diện cho phương trình 2.9. 25](#_Toc95819326)

[Hình 2.10: Kiến trúc của mạng nơ-ron hồi quy trong đó x, h, o, L, y lần lượt thể hiện đầu vào, trạng thái ẩn, đầu ra, hàm mất mát và giá trị đâu ra. 25](#_Toc95819327)

[Hình 2.11: Cấu trúc cơ bản của LSTM. 27](#_Toc95819328)

[Hình 2.12: Cấu trúc cơ bản của LSTM gồm các cổng. 27](#_Toc95819329)

[Hình 2.13: Cấu trúc lan truyền của mạng ConvLSTM [36] 30](#_Toc95819330)

[Hình 2.14: Cấu trúc của ST-LSTM [37]. 31](#_Toc95819331)

[Hình 2.15: Sự lan truyền thông tin theo hướng zigzag [37]. 31](#_Toc95819332)

[Hình 2.16: Thành phần mất mát tách bộ nhớ [24]. 32](#_Toc95819333)

[Hình 3.1: Kiến trúc mạng UNet [27]. 33](#_Toc95819334)

[Hình 3.2: Mô hình đề xuất RainPredRNN. 35](#_Toc95819335)

[Hình 3.3: Mô tả thuật toán gradient descent. 37](#_Toc95819336)

[Hình 4.1: Giá trị mất mát trong quá trình huấn luyện của các mô hình. 44](#_Toc95819337)

[Hình 4.2: Dữ liệu đầu vào thử nghiệm 6 ảnh đầu vào liên tiếp nhau. 44](#_Toc95819338)

[Hình 4.3: Dữ liệu đầu ra kiểm chứng 6 ảnh đầu ra liên tiếp nhau. 45](#_Toc95819339)

[Hình 4.4: Kết quả ảnh dự báo của mô hình PredRNN. 45](#_Toc95819340)

[Hình 4.5: Kết quả ảnh dự báo của mô hình PredRNN\_v2. 46](#_Toc95819341)

[Hình 4.6: Kết quả ảnh dự báo của mô hình RainPredRNN. 46](#_Toc95819342)

[Hình 4.7: Demo ứng dụng dự báo ảnh mưa của mô hình đề xuất PredRNN. 47](#_Toc95819343)

**Danh mục bảng biểu**

[Bảng 1.1: Ma trận nhầm lẫn 12](#_Toc92717877)

[Bảng 4.1: Kích thước các tập dữ liệu 31](#_Toc92717878)

[Bảng 4.2: Các độ đo giá mô hình đề xuất với những mô hình khác. 32](#_Toc92717879)

# BÀI TOÁN DỰ BÁO ẢNH MƯA

## Đặt vấn đề

Dự báo ảnh mưa từ dữ liệu radar có độ phân giải cao là điều cần thiết trong nhiều ngành như quản lý nước, nông nghiệp, hàng không, lập kế hoạch khẩn cấp, v.v. Điều này nhằm mục đích đưa ra các dự đoán chi tiết và hợp lý về ảnh mưa radar trong thời gian tương lai gần dựa trên ảnh radar trong quá khứ với thông tin về thời gian và vị trí của sự kiện mưa. Bài toán này có ý nghĩa quan trọng để dự báo diễn biến mưa trong 1-3-6-9-12 giờ tới với áp thấp nhiệt đới đi theo hướng cho trước, đi từ khu vực này sang khu vực khác. Trong trường hợp lượng mưa lớn trong những nhiều ngày và những ngày tới sẽ tiếp tục gia tăng, dự báo sẽ giúp các địa phương đảm bảo an toàn hồ đập và các tuyến đê xung yếu, tránh xả lũ bất ngờ gây ngập úng vùng hạ lưu. Mưa có tác động xấu đến nhu cầu đi lại, thời gian đi lại, tai nạn giao thông ở các khu vực đô thị trên toàn thế giới [1, 2, 3].

Vì vậy giải quyết bài toán dự báo ảnh mưa dựa trên ảnh radar với thời gian xử lý nhanh, cho kết quả tốt là điều cần thiết và quan trọng. Bên cạnh đó, hiện nay Học sâu cũng đã chứng minh sức mạnh rõ rệt trong các lĩnh vực liên quan đến dự báo thời tiết. Có thể kết tới những ví dụ dưới đây về ứng dụng của Học sâu. Dưới đây là một số ứng dụng của trí tuệ nhân tạo đang được sử dụng trong ngành dự báo thời tiết:

Diagram

Description automatically generated

Hình 1.1: Ứng dụng của Học sâu trong lĩnh vực dự báo thời tiết.

* Ứng dụng trong cháy rừng: dự báo diễn biến, mức độ của vùng cháy rừng. Dự đoán những vùng có khả năng xảy ra cháy rừng cao.
* Ứng dụng trong dự báo đường đi của bão: dự báo diễn biến đường đi, mức độ của bão trong thời gian ngắn.
* Ứng dụng trong dự báo ảnh mưa: dự báo mức độ và diễn biến của vùng mưa trong thời gian ngắn.
* Ứng dụng trong dự báo mây: dự báo mức độ và diễn biến của vùng mây trong thời gian ngắn.
* Ứng dụng trong nghiên cứu khoa học: đưa ra các bài toán cấp thiết có thể giải quyết trong nghiên cứu khoa học.
* Ứng dụng trong giáo dục: giới thiệu, diễn giải, phân tích các bài toán trong lĩnh vực dự báo thời tiết liên quan.

Đồ án này được thiết kế và thực hiện nhằm mục đích đưa ra một mô hình Học sâu nhằm giải quyết bài toán đã đặt ra. Mô hình là một minh chứng cho ứng dụng của trí tuệ nhân tạo nói chung và Học sâu nói riêng trong lĩnh vực dự báo thời tiết, cụ thể là dự báo ảnh mưa. Chi tiết cơ sở lý thuyết, dữ liệu nghiên cứu, thiết kế mô hình và các phần liên quan sẽ được trình bày chi tiết trong các phần sau của đồ án.

## Ý nghĩa

Dự báo ảnh mưa với tập dữ liệu chuỗi thời gian trong thời gian ngắn sẽ có ý nghĩa quan trọng trong nhiều lĩnh vực như: giao thông, nông nghiệp, lịch trình, du lịch, … Hiện nay, Học sâu đã thể hiện dược sức mạnh ứng dụng trong nhiều lĩnh vực, trong đó có cả dự báo thời tiết. Như đã đặt ra bài toán ở trên, việc đưa ra một mô hình Học sâu phù hợp để giải quyết bài toán này sẽ rất quan trọng có ý nghĩa trực tiếp đến ngành dự báo thời tiết, đặt nền móng cho các nghiên cứu sau này. Đây cũng là bước đầu trong việc ứng dụng trực tiếp mô hình Học sâu vào thực tế dự báo.

## Tổng quan các nghiên cứu liên quan

Trong những năm gần đây, Học sâu đã được ứng dụng vào nhiều lĩnh vực [4, 5, 6]. Các biến thể khác nhau của kiến trúc Mạng nơ-ron tích chập (CNN) và Mạng nơ-ron hồi quy (RNN) đã được sửa đổi và áp dụng trong nhiều lĩnh vực khác nhau để tạo ra các phiên bản phù hợp và giải quyết các vấn đề cụ thể [7, 8, 9, 10]. Một số nghiên cứu về ứng dụng của Học sâu cho các bài toán dữ liệu chuỗi thời gian được xem xét ngắn gọn như sau.

Khiali và cộng sự [11] vào năm 2018 đã đề xuất một cách tiếp cận mới là sự kết hợp của các kỹ thuật dựa trên đồ thị để thiết kế một khung mẫu phân cụm mới cho các ảnh chuỗi thời gian vệ tinh. Các đặc trưng không gian trước hết được trích xuất, sau đó được biểu diễn trong các chuyển động của chúng trong đồ thị. Dựa trên các đặc điểm giống nhau của các đặc trưng, các cụm không gian dữ liệu được sinh ra.

Vào năm 2020, Fahim [12] đã đưa ra một mạng nơ-ron tích chập có tên là self-attention convolutional neural network (SAT-CNN). Mạng SAT-CNN sử dụng các đặc trưng trích xuất hình ảnh chuỗi thời gian để phát hiện và phân loại lỗi. Bằng cách thêm một kỹ thuật tiền xử lý đó là biến đổi wavelet rời rạc (discrete wavelet transform - DWT), mô hình đề xuất cho thấy sự vượt trội về hiệu suất so với các mô hình khác.

Vì lý do các phương pháp khai thác hình ảnh chuỗi thời gian truyền thống đã bị con người can thiệp vào quá trình trích xuất các đặc trưng, Li và cộng sự [13] đã giới thiệu một cách tiếp cận mới sử dụng các thuật toán thị giác máy tính khác nhau để tự động trích xuất các đặc trưng từ hình ảnh chuỗi thời gian sau khi hình ảnh được chuyển thành các biểu đồ lặp lại. Phương pháp cho thấy hiệu suất đáng kể trong hai bộ dữ liệu trong hai cuộc thi: cuộc thi dự báo M4 và cuộc thi dự báo du lịch.

Dự báo ảnh mưa đã thu hút nhiều nhà nghiên cứu tham gia [14, 15, 16]. Trong những năm gần đây, thị giác máy tính với sự nổi lên của Học sâu đã đem đến nhiều hứa hẹn. Agrawal và cộng sự. [17] vào năm 2019 đã áp dụng một trong những mô hình phân cụm phổ biến nhất được gọi là UNet để dự ảnh mưa và tạo ra các kết quả rất tốt. Vào năm 2021, Fernández và Mehrkanoon [18] đã sử dụng mô hình Học sâu để dự báo thời tiết bằng cách đưa ra một mô hình kiến ​​trúc mới dựa trên UNet có tên là Broad-UNet. Để trích xuất các đặc trưng trừu tượng phức tạp hơn của hình ảnh đầu vào, mô hình sẽ thay đổi các lớp tích chập và các lớp tổng hợp bằng các tích chập song song không đối xứng và Tập hợp kim tự tháp không gian Atrous (Atrous Spatial Pyramid Pooling - ASPP) tương ứng. Vì vậy, mô hình Broad-UNet thể hiện hiệu suất tuyệt vời so với các mô hình khác. Để hỗ trợ các nhà khí tượng học hiện nay dự báo thời tiết ngắn hạn với khối lượng lớn hình ảnh vệ tinh và radar, Ionescu [19] vào năm 2021 đã giới thiệu họ kiến ​​trúc mạng nơ-ron tích chập được gọi là DeePS. Bằng cách sử dụng năm công cụ vệ tinh để thu thập dữ liệu ảnh vệ tinh, mô hình được phân tích và so sánh với các mô hình dựa trên CNN khác và đạt 3,84% điểm MAE cho toàn bộ tập dữ liệu.

Bằng cách áp dụng các phương pháp Học sâu trong việc hỗ trợ các nhà khí tượng học dự đoán thời tiết cực đoan trong thời gian tương lai gần, Zhang [20] vào năm 2021, đã đề xuất mô hình có hiệu suất cao để dự đoán những thay đổi của hình ảnh tiếng vọng radar thời tiết, dựa trên sự kết hợp của mạng CNN truyền thống và mạng Bộ nhớ ngắn hạn dài (LSTM). Trên thực tế, mô hình của họ tạo ra kết quả rất tốt trong các độ đo đánh giá khác nhau như Chỉ số Thành công Quan trọng (Critical Success Index - CSI) và Heidke Skill Score so với các mô hình ConvLSTM và TrajGRU. Vì các đặc tính vật lý của khí quyển đã đi đầu các lĩnh vực dự báo thời tiết trong những năm qua, Kevin Trebing đã nhận thấy rằng các đặc tính này thiếu thông tin để dự đoán ngắn hạn. Trebing và cộng sự [21] đã giới thiệu một ứng dụng của Học sâu trong mạng nơ-ron mới có hiệu suất có tên là Small Attention Unet (SmaAt-Unet), mạng mới này chỉ sử dụng 25% các tham số của mạng gốc.

Bên cạnh đó, năm 2019 Le Xuan Hien [22] lần đầu tiên áp dụng Mạng LSTM để dự báo lũ trên sông Đà, được đánh giá bằng điểm hiệu quả Nash – Sutcliffe (NSE) trong các trường hợp dự báo khác nhau và tạo ra hiệu suất cao đáng kể (khoảng 90% NSE). Vào năm 2021, Le Xuan Hien [23] cũng so sánh các mô hình Học sâu khác nhau để dự báo dòng chảy của sông. Nhiều mô hình hiện đại khác nhau đã được xem xét và đánh giá, chẳng hạn như StackedLSTM và BiLSTM.

Những năm gần đây, Wang và cộng sự [24] đã giới thiệu một mô hình Học sâu, đã được chứng minh là mạnh mẽ trong việc xử lý các tập dữ liệu hình ảnh chuỗi thời gian. Mô hình của Wang là một cách tiếp cận mới, mô hình là sự kết hợp của các lớp spatiotemporal LSTM (ST-LSTM) và kỹ thuật lưu thông bộ nhớ không gian đã cho ra các kết quả hứa hẹn với các bộ dữ liệu chuẩn.

## Nhận xét về các nghiên cứu liên quan

Như đã đề cập ở trên, Học sâu có thể ứng dụng trong lĩnh vực dự báo hiện nay bằng cách áp dụng các mô hình học sâu hiện đại, nhưng có rất ít bài báo có thể tận dụng các đặc trưng thời gian và không gian dài hạn, và các đặc trưng ngắn hạn của hình ảnh chuỗi thời gian. Đặc biệt là trong dự báo ảnh mưa, vẫn chưa có đủ các nghiên cứu áp dụng cho hình ảnh chuỗi thời gian để dự đoán các hình ảnh trong tương lai [25, 26]. Mặc dù mô hình ban đầu PredRNN\_v2 hoạt động tốt trong hầu hết các trường hợp, nhưng mô hình này tốn nhiều thời gian cho các giai đoạn huấn luyện và dự báo với tập dữ liệu (đặc biệt là về tập dữ liệu radar), và đặc biệt nếu chúng ta muốn huấn luyện lại mô hình với tập dữ liệu lớn hơn.

## Mục tiêu nghiên cứu

Mục tiêu đồ án: thiết kế một mô hình Học sâu mới dựa trên mô hình PredRNN\_v2 [24] và mạng phân cụm UNet để tạo ra một mô hình dự đoán cho bộ dữ liệu radar chuỗi thời gian khắc phục những hạn chế về thời gian xử lý (thời gian xử lý dài).

## Phương pháp nghiên cứu

Từ khi được công cố, các phần mã hóa và giải mã của UNet đã chứng minh rằng nó có vai trò quan trọng trong việc giảm kích thước đầu vào trong khi vẫn nắm bắt được các đặc trưng cấp cao của hình ảnh gốc. Bên cạnh đó, mô hình PredRNN\_v2 có sự mạnh mẽ của trong việc xử lý cả thông tin không gian và thời gian của hình ảnh trong chuỗi thời gian. Vì vậy trong đồ án này, một mô hình mới sẽ được đề xuất là sự kết hợp sức mạnh của kiến trúc UNet [27] và mô hình Học sâu PredRNN\_v2 [24] với mục đích giảm thời gian huấn luyện và dự báo trong khi vẫn bảo toàn các đặc trưng không gian phức tạp của dữ liệu radar. Với sự hỗ trợ mạnh mẽ của công nghệ hiện đại, mô hình đề xuất đã được cài đặt và thử nghiệm thành công và cho ra kết quả tốt, cạnh tranh.

## Phạm vi đồ án

Đồ án được thực hiện với phạm vi đưa ra một mô hình Học sâu cải tiến về thời gian xử lý hình ảnh. Bên cạnh đó vẫn giữ được hiệu suất mô hình không đổi so với phiên bản trước đó. Mô hình đề xuất được thử nghiệm với dữ liệu được thu thập tại khu vực trạm radar thời tiết Pha Đin được đặt tại tỉnh Điện Biên. Tập dữ liệu được thu thập bằng cách phát và nhận các xung sóng vô tuyến trong khoảng thời gian từ tháng 6 đến tháng 7 năm 2020.

## Dữ liệu nghiên cứu

Đồ án này sử dụng bộ dữ liệu phản xạ radar, được lấy từ trạm radar thời tiết Pha Đin, đặt tại tỉnh Điện Biên, Việt Nam. Trạm Pha Đin - nằm ở 21,58° Bắc và 103,52° Nam - chịu sự quản lý trực tiếp của Trung tâm khí tượng thủy văn khu vực Tây Bắc và có nhiệm vụ chính là cung cấp thông tin dự báo ngắn hạn về khí tượng và khí hậu cho các tỉnh trong khu vực này. Trạm chính thức đi vào hoạt động từ tháng 3 năm 2019, đây là một trạm radar thời tiết doppler và có khả năng truyền và nhận các xung sóng vô tuyến theo cả hướng thẳng đứng và hướng ngang. Nhờ đó, trạm có thể cung cấp khả năng quan sát thời tiết với độ phân giải siêu cao và bao phủ một khu vực rộng lớn với bán kính quét hiệu quả lên đến 300km.

Diagram

Description automatically generated

Hình 1.2: Vị trí địa lý của trạm Pha Đin

Đối với vấn đề dự báo ảnh mưa dựa trên dữ liệu radar thời tiết, dữ liệu thu thập được ở đây được hiểu là hình ảnh phản xạ tổng hợp của các xung vô tuyến. Trong đó, những hình ảnh này là hình ảnh xám, và mỗi hình ảnh đại diện cho việc truyền và nhận tín hiệu radar thời tiết. Với vùng phủ sóng 300km×300km (tương đương với phạm vi hiệu quả của radar), những hình ảnh phản xạ này có độ phân giải không gian 150×150 pixel và độ phân giải thời gian tương ứng là 10 phút. Tổng cộng 2.429 ảnh phản xạ tổng hợp radar thời tiết được thu thập trong các đợt mưa diễn ra từ tháng 6 đến tháng 7 năm 2020 (vào mùa mưa ở Việt Nam). Một số hình ảnh radar thời tiết được minh họa trong Hình 1.3.

Timeline

Description automatically generated

Hình 1.3: Mẫu hình ảnh phản xạ radar được ghi lại trong khoảng thời gian từ 17h50 đến 19h00 ngày 23 tháng 6.

Các điểm ảnh có giá trị cao (màu trắng) biểu thị các khu vực mưa. Ngược lại, các pixel có giá trị thấp không phải là vùng mưa. Trong phần thực nghiệm, dữ liệu này sẽ được sử dụng và làm mẫu so sánh của mô hình đề xuất với các mô hình khác.

## Tiêu chí đánh giá

Trong ngữ cảnh của thị giác máy tính, Mean Absolute Error (MAE) [28] là độ đo sự khác biệt giữa mọi pixel của hình ảnh được dự đoán và ảnh nhãn (giá trị thực) của hình ảnh đó. MAE lấy tổng số lỗi tuyệt đối trong toàn bộ tập dữ liệu thử nghiệm sẽ được chia cho số lần quan sát. Độ đo MAE được mô tả toán học dưới đây như sau:

|  |  |  |
| --- | --- | --- |
|  |  | PT 1.1 |

Trong đó và lần lượt là hình ảnh dự đoán thứ và giá trị thực thứ trong tập thử nghiệm và phép toán trừ là một phép toán element wise.

Một độ đo khác cũng có tác động đáng kể đến việc đánh giá hoạt động của mô hình là Chỉ số đo lường sự tương đồng về cấu trúc (Structure Similarity Index Measure - SSIM) [29]. Chỉ số SSIM tính toán sự suy giảm chất lượng hình ảnh sau một số giai đoạn xử lý, đặc biệt là lan truyền thông qua mô hình Học sâu. Công thức dưới đây giải thích phép đo SSIM theo toán học:

|  |  |  |
| --- | --- | --- |
|  |  | PT 1.2 |

Trong đó và lần lượt là trung bình và phương sai của nhãn và dự đoán , là hiệp phương sai của hai ảnh ( và ). và là hai biến chịu trách nhiệm ổn định sự phân chia và được trình bày như sau:

|  |  |  |
| --- | --- | --- |
|  |  | PT 1.3 |

Trong đó , được đặt theo mặc định và là phạm vi động của giá trị pixel của hình ảnh.

Thứ ba, độ đo Critical Success Index (CSI) [30] hay còn được coi là Threat Score sẽ được sử dụng để đánh giá mức độ hoạt động của mô hình RainPredRNN so với những mô hình trước đây. Giả sử rằng chúng ta đang sử dụng bốn đại lượng của Ma trận nhầm lẫn [31] được mô tả trong Bảng 1.1 dưới đây:

Bảng 1.1: Ma trận nhầm lẫn

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Nhãn** | |
| **Mưa** | **Không mưa** |
| **Dự đoán** | **Mưa** | TP | FP |
| **Không Mưa** | FN | TN |

Trong đó Dương tính thật (True Positive - TP) là số lượng pixel mưa trên nhãn (Mưa) đã được dự đoán chính xác. Dương tính giả (False Positive - FP) tương ứng với số lượng pixel không mưa trên nãn (Không Mưa) đã được dự đoán không chính xác. Âm tính giả (False Negative - FN) là số lượng pixel mưa của nhãn không được dự đoán. Âm tính thật (True Negative - TN) tương ứng với số lượng pixel không mưa trên nhãn đã được dự đoán chính xác là không mưa. Điểm CSI được biểu diễn như sau trong phương trình dưới đây:

|  |  |  |
| --- | --- | --- |
|  |  | PT 1.4 |

Trong đồ án này, vì những cải tiến của mô hình vào việc giảm thời gian xử lý của mô hình Học sâu, nên thời gian đào tạo và thử nghiệm cũng được đánh giá là một yếu tố quan trọng để đánh giá hiệu suất của các mô hình. Tiêu chí cuối cùng trong giai đoạn đánh giá là các phép toán nhân tích lũy (Multiply–Accumulate Operation - MACs - Một MAC có một phép toán nhân và một phép toán cộng). Kết quả sẽ được trình bày chi tiết rõ ràng trong phần tiếp theo.

## Bố cục đồ án

Nhằm mục đích thể hiện rõ ràng và chính xác nhất mô hình đề xuất, đồ án này được chia thành 5 chương. Trong đó:

* CHƯƠNG 1 trình bày tổng quan về bài toán, ý nghĩa của bài toán dự báo ảnh lượng mưa cho bộ dữ liệu chuỗi thời gian radar cũng như mô tả khái quát về phương án đề xuất, phạm vi nghiên cứu, phương pháp nghiên cứu và các nghiên cứu liên quan.
* CHƯƠNG 2 là các cơ sở lý thuyết như các kiến thức cơ bản trong học máy, mạng nơ-ron, mạng tích chập, mạng hồi quy, thuật toán gradient descent và các hàm mất mát.
* CHƯƠNG 3 trình bày ý tưởng, kiến trúc của mô hình sử dụng và mô hình đề xuất trong đồ án.
* CHƯƠNG 4 là các thử nghiệm cũng như là đánh giá chất lượng mô hình đề xuất và so sánh với các mô hình cũ bằng dữ liệu được mô tả trong CHƯƠNG 1.
* CHƯƠNG 5 là kết luận, hạn chế của mô hình và định hướng trong tương lai.

# CƠ SỞ LÝ THUYẾT

## Mạng nơ-ron

Chương này trình bày những kiến thức cơ bản về mạng nơ-ron, và được viết dựa theo Blog Machine learning cơ bản của Vũ Hữu Tiệp [32] và tài liệu bài giảng của khóa học CS230 - Deep Learning [33] của đại học Stanford.

Mạng nơ-ron nhân tạo được lấy ý tưởng từ mô hình mạng nơ-ron thần kinh trong não người. Nó đang ngày càng trở nên mãnh mẽ và là một công cụ hiệu quả trong việc xây dựng các hệ thống phân loại, nhận diện tiếng nói, hay thị giác máy tính, v.v.

### Hàm kích hoạt (Activation Function)

Trong các mô hình mạng nơ-ron thông thường, một nơ-ron sau khi tổng hợp các thông tin từ các nơ-ron phía trước, phải cho qua một hàm kích hoạt để có được đầu ra. Một số hàm kích hoạt thường dùng trong mạng nơ-ron là sigmoid [32], tanh [32], hay ReLU [32], hoặc một số biến thể của các hàm trên.

**Hàm kích hoạt sigmoid**

Hàm sigmoid:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.1 |

Chart, histogram

Description automatically generated

Hình 2.1: Đồ thị hàm sigmoid [32].

Tính chất của hàm kích hoạt sigmoid:

* khi
* khi

Do , vì thế khi thì , điều này đồng nghĩa với việc các tham số tại các nơ-ron này sẽ không được cập nhật. Đây chính là hiện tượng biến mất đạo hàm (vanishing gradient).

**Hàm kích hoạt tanh**

Hàm tanh:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.2 |

Chart, line chart

Description automatically generated

Hình 2.2: Đồ thị hàm Tanh [32].

Một số tính chất hàm tanh:

* . Dễ thấy nếu hoặc thì , do đó , nên hàm cũng gặp vấn đề biến mất đạo hàm tương tự như với hàm .

**Hàm kích hoạt ReLU**

Hàm ReLU:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.3 |

Chart, line chart

Description automatically generated

Hình 2.3: Đồ thị hàm ReLU [32].

ReLU được sử dụng rất rộng rãi trong các mô hình mạng nơ-ron hiện tại. Hàm ReLU chỉ giữ lại các giá trị dương, và bỏ qua các giá trị nhỏ hơn 0. Hàm ReLU có nhiều ưu điểm vượt trội so với hàm sigmoid và hàm tanh như:

* Do có dạng hàm cũng như có đạo hàm rất đơn giản, ReLU có tốc độ nhanh hơn hẳn so với sigmoid và tanh. ReLU có tốc độ hội tụ nhanh gấp 6 lần tanh. Điều này có thể do ReLU không bị bão hoà ở hai đầu như sigmoid và tanh.
* ReLU cũng có nhược điểm: Với các node nhỏ hơn 0, khi đi qua ReLU sẽ biến thành 0. Hiện tượng đó gọi là Dying ReLU. Nếu các nơ-ron bị chuyển thành 0 thì sẽ không có ý nghĩa khi truyền tới các lớp tiếp theo và các trọng số tương ứng từ nơ-ron đấy cũng không được cập nhật với gradient descent. Từ đó, Leaky ReLU ra đời.

### Hàm mất mát Mean Squared Error (MSE)

Sai số toàn phương trung bình (MSE) [33] được sử dụng trong các nhiệm vụ hồi quy trong đó chúng ta đang cố gắng giảm thiểu giá trị kỳ vọng của một số hàm trên dữ liệu huấn luyện, còn được gọi là “hàm mất mát”.

Sai số toàn phương trung bình (MSE) được tính bằng cách lấy chênh lệch giữa nhãn và dự đoán của mạng , sau đó lấy bình phương các giá trị đó. Chúng ta lấy bình phương của chênh lệch này, cộng tất cả chúng lại với nhau để có giá trị cuối cùng, cuối cùng lại chia số này cho tổng số giá trị nhãn. Đây sẽ là kết quả cuối cùng. Công thức tính toán sai số toàn phương trung bình như sau:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.4 |

Điều này sẽ cung cấp cho chúng ta một giá trị mất mát từ 0 đến vô cùng với các giá trị lớn hơn cho biết lỗi bình phương trung bình.

### Lan truyền tiến (Feed Forward)

Thuật ngữ Feed Forward [33] được hiểu là lan truyền tiến, tức truyền thông tin theo một đường thẳng, và không "quay lại". Mạng nơ-ron về bản chất, chính là một hàm số dùng để mô tả quan hệ của dữ liệu huấn luyện , và cho phép dự đoán nhãn cho dữ liệu đầu vào ở tập kiểm thử: .

Xét một mạng nơ-ron nhiều tầng (MLP). Một mạng MLP chính là một mạng nơ-ron gồm một hay nhiều tầng ẩn, luồng dữ liệu đi qua mạng là lan truyền tiến.

Xét một MLP có tầng, khi đó . Trong đó tầng 1 đóng vai trò , tầng 2 đóng vai trò . Trong một mạng nơ-ron thông thường, hàm thường được sử dụng có dạng:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.5 |

Trong đó chỉ hàm kích hoạt. Hàm kích hoạt trong mạng nơ-ron phải là hàm phi tuyến. Bởi nếu ta chọn hàm kích hoạt là hàm tuyến tính thì sẽ không còn ý nghĩa nữa, bởi hàm tuyến tính của một hàm tuyến tính vẫn sẽ là một hàm tuyến tính. là ma trận chuyển vị của ma trận . Đầu ra của tầng này sẽ là đầu vào của tầng kế tiếp nó. Tổng quát, ta có công thức lan truyền tiến cho mạng MLP với tầng, đầu vào như sau:

|  |  |  |
| --- | --- | --- |
|  | , với là hàm kích hoạt ở tầng thứ | PT 2.6 |

chính là đầu ra dự đoán của mô hình với đầu vào . Trong thực tế, ta luôn mong muốn và là giống nhau. Với và , ta sẽ thiết lập các hàm mất mát sao cho phù hợp với từng bài toán.

### Lan truyền ngược (Backpropagation)

Khi huấn luyện một mô hình, điều tiên quyết cần phải làm được là tính được đạo hàm cho các tham số của mô hình đó. Đối với một mạng nơ-ron, phương pháp phổ biến nhất được sử dụng hiện nay chính là lan truyền ngược [33].

Xét hai hàm số là hai hàm số liên tục và khả vi trên . Gọi , hay .Khi đó . Công thức này có ý nghĩa rằng, nếu muốn tính đạo của hàm hợp với biến , ta có thể thực hiện qua hai bước: đầu tiên ta tính đạo hàm của theo , rồi lấy kết quả nhân với đạo hàm của theo . Hay nói cách khác đạo hàm được tính ngược từ cuối, đó chính là lan truyền ngược.

Xét bài toán phân loại dữ liệu với nhãn , có thể nhận một trong giá trị. Trong bài toán này, ta sử dụng một mô hình mạng nơ-ron MLP với chỉ một tầng ẩn, với số nơ-ron trong tầng ẩn là . Hàm kích hoạt tại tầng này được sử dụng là hàm ReLU. Hàm mất mát được sử dụng trong bài toán này là hàm mất mát Cross Entropy. Tham số cần học của bài toán bao gồm , trong đó .

Với một đầu vào , quá trình lan truyền tiến sẽ được thực hiện như sau:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.7 |

Ta sẽ tính đạo hàm ngược từ tầng thứ 2 trước. Xét một cặp điểm dữ liệu . Chú ý rằng sẽ có dạng là một véc-tơ one-hot. Mục tiêu của chúng ta sẽ là phải tính toán được các giá trị với hàm mất mát:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.8 |

Bằng cách lan truyền ngược lại đạo hàm từ tầng cuối cùng mà mọi tham số của mạng nơ-ron đều có thể tính đạo hàm được.

## Mạng nơ-ron tích chập (Convolution Neural Network)

Phần này sẽ trình bày mạng nơ-ron tích chập, nội dung và các hình ảnh trong phần này được tham khảo từ [33]. Mạng nơ-ron tích chập (Convolutional neural networks), còn được biết đến với tên CNN, là một dạng mạng nơ-ron được cấu thành bởi các tầng sau:

Diagram

Description automatically generated

Hình 2.4: Mô hình tổng quát mạng nơ-ron tích chập [33].

**Tầng tích chập**

Tầng tích chập (Convolution) sử dụng các bộ lọc để thực hiện phép tích chập khi chúng đi qua đầu vào ***I*** theo chiều của nó. Các siêu tham số của bộ lọc này bao gồm kích thước bộ lọc **F** và độ trượt (stride) **S**. Kết quả đầu ra ***O*** được gọi là lớp đặc trưng hay activation map. Trong Hình 2.7, đầu vào ***I*** hình bên trái, bộ lọc hình giữa, đầu ra ***O*** hình bên phải.

Diagram

Description automatically generated

Hình 2.5: Tầng tích chập hai chiều [33].

**Tầng pooling**

Tầng pooling là một phép giảm chiều (downsampling), thường được sử dụng sau tầng tích chập, giúp tăng tính bất biến không gian. Cụ thể, max pooling và average pooling là những dạng pooling đặc biệt, trong đó giá trị lớn nhất và giá trị trung bình được lấy ra.

* Max pooling: từng phép max pooling chọn giá trị lớn nhất trong khu vực mà nó đang được áp dụng. Max pooling giúp bảo toàn các đặc trưng đã phát hiện.
* Average pooling: từng phép pooling tính trung bình các giá trị trong khu vực mà nó đang được áp dụng. Average pooling giúp giảm kích thước lớp đặc trưng.

Chart

Description automatically generated with low confidence

Hình 2.6: Tầng Max pooling (bên trái) và Average pooling (bên phải) [33].

**Các siêu tham số của bộ lọc**

**Các chiều của bộ lọc**: Một bộ lọc có kích thước áp dụng lên đầu vào chứa eq kênh (channels) thì có kích thước tổng thể là thực hiện phép tích chập trên đầu vào có kích thước và cho ra một lớp đặc trưng có kích thước . Lưu ý: việc áp dụng bộ lọc có kích thước cho ra một lớp đặc trưng có kích thước .

**Độ trượt**

Đối với phép tích chập hoặc phép pooling, độ trượt ký hiệu số điểm ảnh mà cửa sổ sẽ di chuyển sau mỗi lần thực hiện phép tính.

Chart

Description automatically generated with medium confidence

Hình 2.7: Minh họa độ trượt [33].

**Padding**

Padding là tên gọi của quá trình thêm số vào các biên của đầu vào. Tùy thuộc vào kích thước của lớp đặc trưng đầu ra mong muốn mà ta có thể lựa chọn các giá trị phù hợp.

A picture containing text, crossword puzzle

Description automatically generated

Hình 2.8: Minh họa padding [33].

## Mạng nơ-ron hồi quy (Recurrent Neural Network)

Mạng nơ-ron hồi quy (RNN) [34] là một phần của họ mạng nơ-ron được sử dụng để xử lý dữ liệu tuần tự. Phương trình của mạng RNN:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.9 |

Diagram

Description automatically generated

Hình 2.9: Một mạng nơ-ron hồi quy không có đầu ra đại diện cho phương trình 2.9.

Mạng RNN lấy làm đầu vào và kết hợp nó vào trạng thái , còn được gọi là trạng thái ẩn được chuyển tiếp. Hình vuông màu đen biểu thị độ trễ trong một bước thời gian. Mạng nơ-ron hồi quy chia sẻ cùng một trọng số huấn luyện qua mỗi bước thời gian, đây là một thuộc tính quan trọng của RNN.

Diagram

Description automatically generated

Hình 2.10: Kiến trúc của mạng nơ-ron hồi quy trong đó x, h, o, L, y lần lượt thể hiện đầu vào, trạng thái ẩn, đầu ra, hàm mất mát và giá trị đâu ra.

Mạng nơ-ron hồi quy ánh xạ một chuỗi giá trị đầu vào với một chuỗi giá trị đầu ra tương ứng. Hàm giá trị mất mát đo lường sự khác biệt giữa nhãn thực tế và kết quả dự đoán . Mạng RNN cũng có đầu vào cho kết nối ẩn được tham số hóa bởi ma trận trọng số , trạng thái ẩn được kết nối đến trạng thái ẩn được tham số hóa bởi ma trận trọng số và kết nối ẩn đến đầu ra được tham số hóa bởi ma trận trọng số . Sau đó từ bước thời gian đến được áp dụng phương trình sau:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.10 |

Các phương trình trên còn được gọi là lan truyền tiến RNN trong đó và là các vectơ độ chệch và tanh và softmax là các hàm kích hoạt. Để cập nhật ma trận trọng số , , , chúng ta tính toán gradient của hàm mất mát cho mỗi ma trận trọng lượng tức là , , , và cập nhật từng ma trận trọng số với sự trợ giúp của một thuật toán lan truyền ngược. Khi một thuật toán lan truyền ngược được áp dụng cho RNN, nó đôi khi còn được gọi là BPTT (backpropagation through time), tức là lan truyền ngược theo thời gian. Tính toán gradient yêu cầu lan truyền tiến và lan truyền ngược của mạng có nghĩa là thời gian chạy của cả hai quá trình truyền là với n là độ dài của đầu vào. Thời gian chạy của thuật toán không thể giảm hơn nữa vì thiết kế của mạng vốn là tuần tự.

Tùy thuộc vào mục tiêu mà chúng ta có thể chọn bất kỳ hàm mất mát nào. Tổng số hàm mất mát cho một chuỗi các giá trị nhất định là tổng của tất cả các tổn thất tại một bước thời gian riêng lẻ.

## Long Short-Term Memory (LSTM)

Về cơ bản, LSTM [35] hoạt động rất giống với RNN [34] (đã nhắc đến ở phần 2.3). LSTM bao gồm ba phần, như thể hiện trong hình dưới đây và mỗi phần thực hiện một chức năng riêng lẻ.

Diagram

Description automatically generated

Hình 2.11: Cấu trúc cơ bản của LSTM.

Phần đầu tiên chọn xem thông tin đến từ mốc thời gian trước đó sẽ được ghi nhớ nếu liên quan hay có thể bị lãng quên nếu không liên quan. Trong phần thứ hai, LSTM cố gắng tìm hiểu thông tin mới từ đầu vào cho ô này. Cuối cùng, trong phần thứ ba, ô chuyển thông tin cập nhật từ mốc thời gian hiện tại sang mốc thời gian tiếp theo. Ba phần này của lớp LSTM được gọi là cổng. Phần đầu tiên được gọi là cổng quên (forget gate), phần thứ hai được gọi là cổng đầu vào (input gate) và phần cuối cùng là cổng đầu ra (đầu ra).

Cũng giống như RNN đơn giản, LSTM cũng có trạng thái ẩn trong đó đại diện cho trạng thái ẩn của mốc thời gian trước đó và là trạng thái ẩn của mốc thời gian hiện tại. Ngoài ra, LSTM cũng có một trạng thái ô được đại diện bởi và cho mốc thời gian trước đó và hiện tại tương ứng.

Ở đây trạng thái ẩn được gọi là bộ nhớ ngắn hạn (short-term) và trạng thái ô được gọi là bộ nhớ dài hạn (long-term). Tham khảo Hình 2.12 sau đây.

Diagram

Description automatically generated

Hình 2.12: Cấu trúc cơ bản của LSTM gồm các cổng.

**Forget Gate**

Trong một ô của mạng LSTM, bước đầu tiên là quyết định xem chúng ta nên giữ thông tin từ dấu thời gian trước hay quên nó. Đây là phương trình cho cổng quên:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.11 |

Trong đó:

* : đầu vào cho mốc thời gian hiện tại.
* : trọng số liên quan đến đầu vào
* : Trạng thái ẩn của mốc thời gian trước đó
* : Là ma trận trọng số liên quan đến trạng thái ẩn

Sau đó, một hàm sigmoid được áp dụng. Điều đó sẽ làm cho giá trị hàm trở thành một số từ 0 đến 1. Giá trị hàm này sau đó được nhân với trạng thái ô của mốc thời gian trước đó như được hiển thị bên dưới.

|  |  |  |
| --- | --- | --- |
|  | nếu thì quên mọi thứ  nếu thì không quên | PT 2.12 |

Nếu bằng 0 thì mạng sẽ quên mọi thứ và nếu giá trị của bằng 1 thì nó sẽ không quên gì cả.

**Input Gate**

Cổng đầu vào được sử dụng để định lượng tầm quan trọng của thông tin mới do đầu vào mang theo. Đây là phương trình của cổng đầu vào:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.13 |

Trong đó:

* : Đầu vào tại mốc thời gian hiện tại t
* : ma trận trọng số của đầu vào
* : Trạng thái ẩn ở mốc thời gian trước
* : Ma trận trọng số của đầu vào được liên kết với trạng thái ẩn

Một lần nữa, LSTM đã áp dụng hàm kích hoạt sigmoid. Kết quả là, giá trị của tại dấu thời gian sẽ nằm trong khoảng từ 0 đến 1.

**Thông tin mới**

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.14 |

Bây giờ thông tin mới cần được chuyển đến trạng thái ô là một hàm của trạng thái ẩn ở mốc thời gian trước đó và đầu vào ở dấu thời gian . Hàm kích hoạt ở đây là tanh. Do hàm tanh, giá trị của thông tin mới sẽ từ -1 đến 1. Nếu giá trị của là âm, thông tin sẽ bị trừ khỏi trạng thái ô và nếu giá trị là dương, thông tin sẽ được thêm vào trạng thái ô tại mốc thời gian hiện tại.

Tuy nhiên, sẽ không được thêm trực tiếp vào trạng thái ô. Đây là phương trình cập nhật:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.15 |

Ở đây, là trạng thái ô tại mốc thời gian hiện tại và các trạng thái khác là các giá trị mà chúng ta đã tính toán trước đó.

**Output Gate**

Đây là phương trình của cổng ra, khá giống với hai cổng trước:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.16 |

Giá trị của nó cũng sẽ nằm trong khoảng từ 0 đến 1 vì hàm sigmoid. Bây giờ để tính toán trạng thái ẩn hiện tại, chúng ta sẽ sử dụng và hàm tanh của trạng thái ô được cập nhật:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.17 |

Nó chỉ ra rằng trạng thái ẩn là một hàm của bộ nhớ dài hạn () và đầu ra hiện tại. Nếu ta cần lấy đầu ra của dấu thời gian hiện tại, chỉ cần áp dụng kích hoạt Softmax ở trạng thái ẩn :

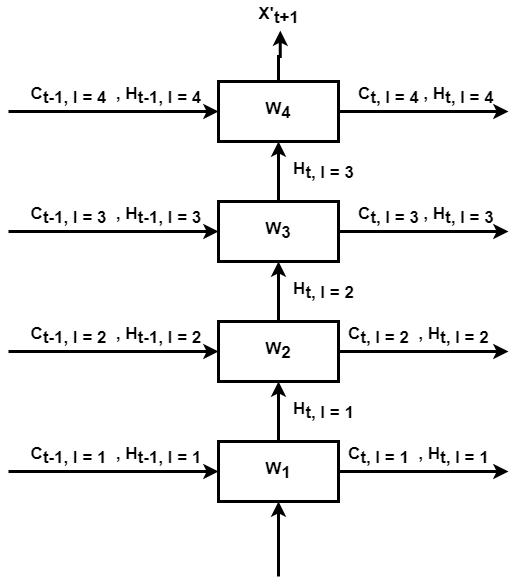
|  |  |  |
| --- | --- | --- |
|  |  | PT 2.18 |

## Convolution LSTM (Conv-LSTM)

Vì các lớp LSTM [35] truyền thống, là một kiến trúc RNN [34] đặc biệt, có một nhược điểm đáng kể trong việc mô hình hóa đồng thời thông tin không gian của đầu vào, trạng thái ẩn và ô nhớ đầu ra, ConvLSTM [36] với nhiều cải tiến khác nhau có thể giải quyết vấn đề phiên bản cũ (FC-LSTM). Đầu tiên, để mã hóa thông tin cấu trúc không gian, tất cả các đầu vào , các ô đầu ra và các trạng thái ẩn là các tensor 3D () trong đó M và N là các hàng và cột thể hiện các kích thước chiều không gian. Thứ hai, vì ‘’ và ‘’ biểu thị toán tử tích chập (convolution) và tích Hadamard, tất cả các cổng cũng là các tensor 3D chịu trách nhiệm biến đổi thông tin trong các điều kiện khác nhau. Các phương trình của ConvLSTM được mô tả như sau:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.19 |

Vì hai kích thước cuối cùng của FC-LSTM truyền thống bằng 1, FC-LSTM có thể được coi là trường hợp đặc biệt của ConvLSTM. Mặc dù ConvLSTM có vai trò quan trọng mở đường cho việc xử lý tập dữ liệu hình ảnh chuỗi thời gian cho nhiều vấn đề trong cuộc sống thực, nhưng có một số điểm có thể được cải thiện hơn nữa trong kiến trúc này. Đầu tiên, các trạng thái bộ nhớ chỉ đơn thuần phụ thuộc vào biểu diễn thứ bậc đặc trưng của các lớp khác do các trạng thái được cập nhật theo chiều ngang của các lớp tương ứng.



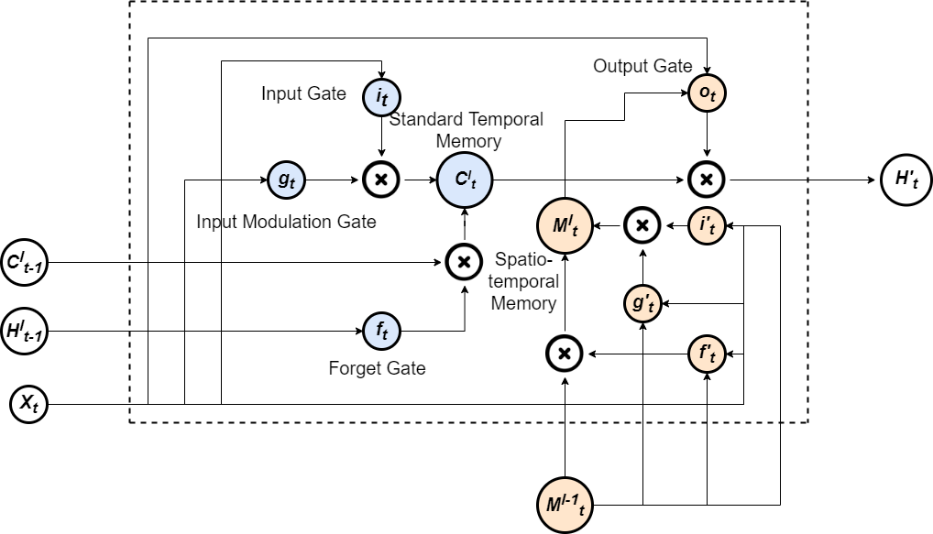
Hình 2.13: Cấu trúc lan truyền của mạng ConvLSTM [36]

Điều này có nghĩa là toán tử trong lớp đầu tiên của mốc thời gian hiện tại sẽ không có thông tin những đặc trưng nào được ghi nhớ trong lớp trên cùng trước đó của mốc thời gian . Thứ hai, vì các trạng thái ẩn là đầu ra của hai cổng và có nghĩa là sẽ chứa cả thông tin dài hạn và ngắn hạn, hiệu suất của mô hình sẽ bị hạn chế đáng kể bởi các biến thể không gian này.

## Spatiotemporal LSTM (ST-LSTM) với Memory Flow

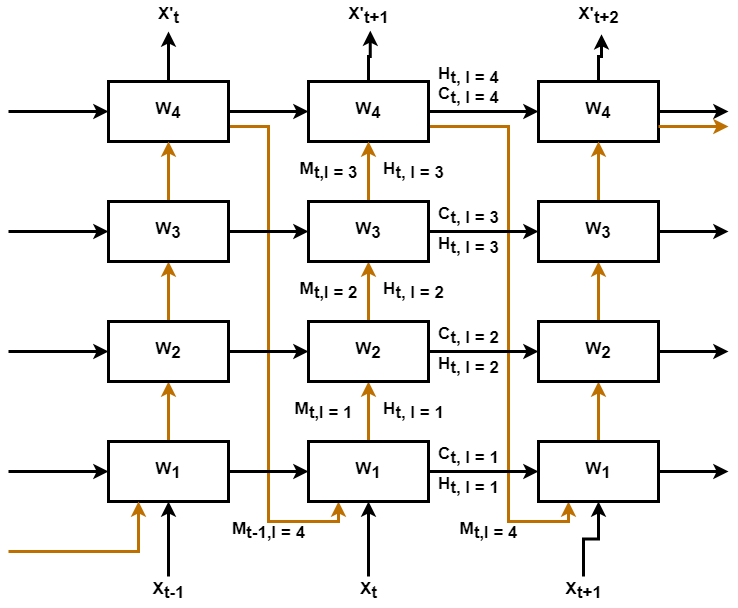
Bằng cách kết hợp bộ nhớ thông tin không gian ngắn hạn và dài hạn (spatiotemporal long short-term memory - ST-LSTM) như là các khối xây dựng cơ bản và kiến trúc lưu thông bộ nhớ không gian, Wang và cộng sự [37] đã giới thiệu mạng neural hồi quy dự đoán (predictive recurrent neural network – PredRNN), mạng đã vượt qua các giới hạn của phiên bản cũ ConvLSTM. Các phương trình của ST-LSTM được biểu diễn như sau [37]:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.20 |



Hình 2.14: Cấu trúc của ST-LSTM [37].

Có hai cải tiến đáng kể được đề xuất bởi PredRNN: ô nhớ không gian và cách các ô này được cập nhật theo hướng zigzag. Có hai ô nhớ chứa thông tin thời gian và không gian: ô đầu ra tiêu chuẩn được truyền theo chiều ngang từ lớp tương ứng trước đó tại mốc thời gian đến bước thời gian hiện tại và ô mới được phân phối theo chiều dọc từ phía dưới lớp . Trong lần cải tiến đầu tiên, bằng cách trình bày cấu trúc cổng cho , đầu ra ẩn cuối cùng được hưởng lợi từ việc chứa thông tin của cả hai cổng và . Thứ hai, ô bộ nhớ không gian được phân phối theo kiểu zigzag (thông tin được truyền tải lên trên trước rồi chuyển tiếp giữa các lớp), có nghĩa là từ lớp đầu tiên, (có stack ST-LSTM lớp). Cơ chế tạo ra động lực dài hạn và ngắn hạn dẫn đến đầu ra ẩn bằng cách kết hợp các cặp trạng thái bộ nhớ (theo chiều ngang và chiều dọc).



Hình 2.15: Sự lan truyền thông tin theo hướng zigzag [37].

## Spatiotemporal LSTM với Memory Decoupling

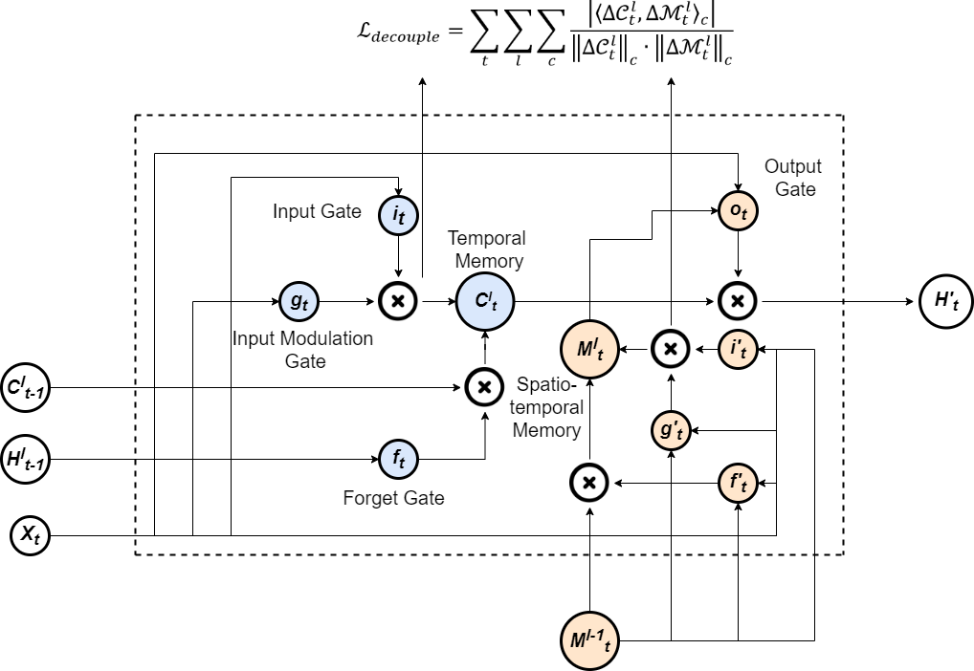
Trên thực tế, bằng cách sử dụng t-SNE [38] để trực quan hóa dữ liệu bộ nhớ tại mọi dấu thời gian, tác giả [24] nhận thấy rằng các trạng thái bộ nhớ không được phân biệt với nhau một cách tự động và được tách ra một cách tự nhiên. Bằng cách dựa trên PredRNN, tác giả đã thiết lập một hàm mất mát mới là sự kết hợp của hàm mất mát Meam Square Error (MSE) và mất mát tách mới:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.21 |

Trong đó là hàm mất mát MSE truyền thống của phiên bản trước PredRMM và là hàm mất mát chính quy tách 1bộ nhớ được miêu tả bằng phương trình toán học như sau:

|  |  |  |
| --- | --- | --- |
|  |  | PT 2.22 |

Trong đó là tham số của lớp tích chập được thêm vào sau ô bộ nhớ và tại mỗi bước thời gian. Bằng cách này, hai trạng thái nhớ được tách biệt và huấn luyện trên các khía cạnh của thông tin không gian và thời gian. Hơn nữa, lớp tích chập mới sẽ được loại bỏ ở pha dự đoán làm cho kích thước của toàn bộ mô hình không thay đổi. Từ đây tạo ra phiên bản mới của PredRNN có tên là PredRNN\_v2.



Hình 2.16: Thành phần mất mát tách bộ nhớ [24].

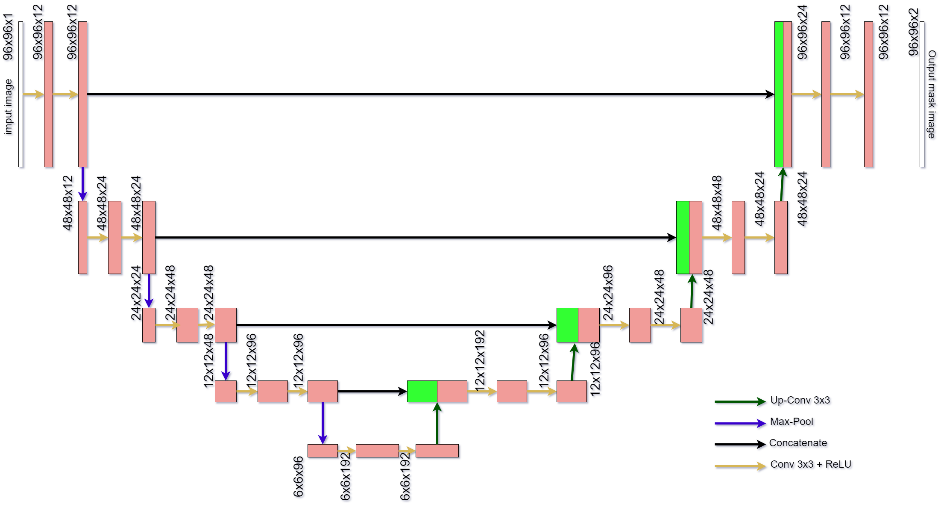
# MÔ HÌNH ĐỀ XUẤT

## Ý tưởng

Trong chương này, bằng cách sử dụng thế mạnh của mô hình PredRNN\_v2, mô hình mới có tên **RainPredRNN** sẽ được đề xuất có thể phù hợp với các vấn đề xử lý hình ảnh radar trong chuỗi thời gian để dự đoán hình ảnh trong bước thời gian sau. Mô hình đề xuất sử dụng phần mã hóa và giải mã của mô hình UNet làm bộ mã hóa và bộ giải mã trước và sau khi chuyển tiếp đầu vào đến các lớp ST-LSTM, điều này sẽ giảm thiểu một số lượng lớn các phép tính phải thực hiện. Chi tiết mô hình được mô tả ở các phần tiếp theo.

## Kiến trúc mạng UNet

Kiến trúc UNet [27], được xuất bản lần đầu tiên vào năm 2015, đã là một cuộc cách mạng trong lĩnh vực học sâu. Kiến trúc đã giành chiến thắng trong Hội nghị chuyên đề quốc tế về thách thức theo dõi tế bào Y sinh (ISBI) năm 2015 ở nhiều hạng mục với tỷ lệ chênh lệch lớn. Một số công trình của họ bao gồm phân đoạn cấu trúc tế bào thần kinh trong các ngăn xếp hiển vi điện tử và hình ảnh hiển vi ánh sáng truyền qua.



Hình 3.1: Kiến trúc mạng UNet [27].

Qua kiến ​​trúc được hiển thị trong Hình 3.1, chúng ta có thể nhận thấy tại sao mạng được gọi là kiến ​​trúc U-Net. Hình dạng của kiến ​​trúc được hình thành có dạng chữ 'U' và do đó tên U-Net bắt ngồn từ đây. Nhìn vào cấu trúc và vô số yếu tố liên quan đến quá trình xây dựng kiến ​​trúc này, có thể hiểu rằng mạng được xây dựng là một mạng phức hợp đầy đủ. Họ đã không sử dụng bất kỳ lớp nào khác như dense hoặc làm phẳng (flatten) hoặc các lớp tương tự khác. Biểu diễn trực quan cho thấy U-Net có một phần mã hóa (encoder) theo sau là một phần giải mã (decoder).

Kiến trúc cho thấy rằng một hình ảnh đầu vào được xử lý qua mô hình và sau đó nó được theo sau bởi một vài lớp tích chập với hàm kích hoạt ReLU. Kích thước hình ảnh đang giảm từ 572×572 xuống 570×570 và cuối cùng là 568×568. Lý do cho việc giảm này là do mạng đã sử dụng các chập với padding bằng 0, dẫn đến giảm kích thước tổng thể. Ngoài các khối tích chập, mạng U-Net có một khối mã hóa ở phía bên trái, tiếp theo là khối bộ giải mã ở phía bên phải.

Khối mã hóa giảm kích thước hình ảnh liên tục với sự trợ giúp của các lớp tổng hợp tối đa (Maxpooling). Mạng có các lớp tích chập lặp lại với số lượng bộ lọc (filter) ngày càng tăng trong kiến ​​trúc bộ mã hóa. Khi dữ liệu xử lý gần cuối bộ giải mã, số lượng bộ lọc (filter) trong các lớp tích chập bắt đầu giảm cùng với việc lấy mẫu (usampling) với kích thước không gian tăng dần ở các lớp sau lên đến lớp đầu ra. Việc sử dụng các kết nối bỏ qua (skip-connection) kết nối các đầu ra trước đó với các lớp trong khối bộ giải mã là giảm sự mất mát của gradient trong quá trình huấn luyện (lan truyền ngược).

Kết nối bỏ qua này là một khái niệm quan trọng để bảo toàn sự mất mát thông tin từ các lớp trước để chúng phản ánh mạnh mẽ hơn về các giá trị tổng thể. Chúng cũng được khoa học chứng minh là tạo ra kết quả tốt hơn và dẫn đến sự hội tụ mô hình nhanh hơn. Trong khối tích chập cuối cùng, chúng ta có một vài lớp tích chập theo sau là lớp tích chập cuối cùng. Lớp này có bộ lọc 2 với chức năng thích hợp để hiển thị kết quả đầu ra. Lớp cuối cùng này có thể được thay đổi tùy theo mục đích mong muốn của bài toán đang thực hiện.

## Kiến trúc tổng thể mô hình đề xuất

Để tận dụng sự mạnh mẽ của mô hình UNet, mô hình đễ xuất sẽ lấy các đặc điểm chính của kiến ​​trúc: bộ mã hóa (encoder), bộ giải mã (decoder) và kỹ thuật kết nối (concatenation).

Đầu tiên, mọi hình ảnh đầu vào gốc được truyền qua bộ mã hóa với một lớp tổng hợp tối đa (Maxpooling) nằm giữa bốn lớp tích chập 3×3. Bằng cách làm này, các ngữ cảnh có giá trị cấp cao của các đầu vào sẽ được ghi lại và lưu trữ trong các bản đồ đặc trưng trước khi xử lý bởi các lớp ST-LSTM [37].

Vì các thuật toán thay đổi kích thước hình ảnh truyền thống sẻ làm mất thông tin hình ảnh đáng kể và chuyển đổi hình ảnh không đúng cách, nên bộ mã hóa giữ lại nhiều ngữ cảnh nhất có thể và vẫn làm giảm kích thước không gian của hình ảnh đầu vào gốc.

Vì ST-LSTM được thiết kế với nhiều cổng và số lượng lớn các phép toán dấu phẩy động (FLOPs), đầu vào càng lớn thì càng phải tính toán nhiều phép tính. Sau khi mã hóa, các đầu vào ban đầu được giảm một nửa chiều rộng và chiều cao và có nhiều thông tin không gian hơn.

Phần giải mã sẽ được thêm vào ngay sau các lớp ST-LSTM và lấy đầu ra của các lớp này làm đầu vào. Tại thời điểm này, kỹ thuật kết nối (concatenation) sẽ được áp dụng thêm thông tin của dữ liệu tương ứng trong bộ mã hóa để nhận được nhiều thông tin hơn và tránh vấn đề biến mất gradient (vanishing gradient).

Do đó, thời gian tính toán của ST-LSTM sẽ được giảm đáng kể trong cả quá trình lan truyền xuôi và ngược. Hình ảnh trực quan về mô hình được thể hiện trong Hình 3.2 như sau:

Chart, box and whisker chart

Description automatically generated

Hình 3.2: Mô hình đề xuất RainPredRNN.

Các hộp có chữ “ST-LSTM” biểu thị lớp ST-LSTM thông thường, các hộp màu xám biểu thị hình ảnh ở các mức xử lý khác nhau của mô hình và các hộp màu nâu là bản sao của bản đồ đặc trưng. Trong khi bộ mã hóa có vai trò giảm kích thước không gian của đầu vào để giảm thời gian tính toán cho các lớp ST-LSTM xếp chồng, bộ giải mã xử lý đầu ra của các lớp ST-LSTM xếp chồng để khôi phục trở lại kích thước ban đầu của đầu vào.

Bằng cách sử dụng một lớp upsampling, chúng ta sẽ có được kích thước không gian ban đầu của các hình ảnh gốc.

Hướng tiếp cận mới của mô hình nằm ở kỹ thuật mã hóa và giải mã dữ liệu trước và sau khi ra khỏi phần xử lý chính. Các kỹ thuật biển đổi và nén ảnh thông thường sẽ làm cho dữ liệu mất mát thông tin, từ đó mô hình sẽ cho ra kết quả với chất lượng không tốt. Bằng việc kết hợp kỹ thuật này, các thông tin về không gian được dữ lại nhưng vẫn làm giảm chiều không gian và thời gian. Mô hình là một kiến trúc tổng thể của sự kết hợp nêu trên thay vì rời rạc với các kỹ thuật tiền xử lý biến đổi. Việc sửa đổi này đã giảm đáng kể thời gian huấn luyện và thử nghiệm trong khi mô hình vẫn tạo ra kết quả được đánh giá giống như phiên bản cũ. Kết quả sẽ trình bày chi tiết các kết quả thử nghiệm trong CHƯƠNG 4.

## Vai trò của phần mã hóa và giải mã trong mạng UNet

Phần này sẽ trình bày về vai trò của các phần mã hóa và giải mã của UNet trong kiến trúc tổng thể của mạng đề xuất.

**Phần mã hóa**: hai lớp tích chập 3×3 liên tiếp để trích rút ngữ cảnh của hình ảnh gốc, theo sau là một hàm kích hoạt ReLU để làm cho mô hình trở nên phi tuyến và chuẩn hóa theo lô – batch normalization (chính quy). Để giảm kích thước dữ liệu sau khi xử lý, một lớp tổng hợp là lớp tổng hợp tối đa (Maxpooling) được áp dụng ngay sau các lớp tích chập này. Sau mỗi bước xử lý trên, các đầu vào ban đầu được cắt giảm một nửa kích thước không gian và tăng gấp đôi số kênh đặc trưng để tạo bản đồ đặc trưng cấp cao.

**Phần giải mã**: Đầu tiên, mô hình phải khuếch đại bản đồ đặc trưng do phần mã hóa tạo ra để dần dần trở lại hình dạng kích thước ban đầu. Thứ hai, sau mỗi toán tử khuếch đại, số lượng kênh đặc trưng sẽ bị cắt đi một nửa bởi một lớp tích chập chuyển vị 2×2. Ngoài ra, một kỹ thuật nối (concatenation) sẽ được sử dụng từ bản đồ đặc trưng tương ứng trong phần mã hóa để tránh làm biến mất các gradient (gradient vanish) khi huấn luyện. Thứ ba, hai lớp tích chập 3×3 với hàm kích hoạt ReLU và chuẩn hóa theo lô được áp dụng. Ở lớp cuối cùng, lớp tích chập 1×1 được sử dụng để ánh xạ mọi pixel thành số lớp mong muốn.

## Vai trò của PredRNN

Các lớp ST-LSTM trong PredRNN có vai trò xử lý hình ảnh chuỗi thời gian với đầu vào là ảnh trong bước thời gian trước và đầu ra là ảnh của bước thời gian tiếp theo. ST-LSTM với các cổng đặc trưng của LSTM: cổng đầu vào, cổng quên và cổng trạng thái ẩn với các phép toán nhân thông thường được thay bằng phép tích chập. Điều này sẽ giúp cho mạng trích xuất được thông tin không gian của ảnh đầu vào. Cùng với cổng quên, mạng cũng sẽ có khả năng giữ hoặc quên các thông tin không gian và thời gian trong dài hạn. Một đặc trưng nữa của PredRNN đó là phương thức lan truyền thông tin giữa các lớp ST-LSTM và các bước thời gian. Bằng việc thiết kế ra một luồng lan truyền thông tin theo hướng zigzag, mạng đã thành công trong việc lan truyền được các đặc trưng trừu tượng của ảnh tại bước thời gian trước tới bước thời gian hiện tại. Cuối cùng, ta có thể thấy PredRNN rất thành công trong việc xử lý ảnh chuỗi thời gian.

## Thuật toán học

Chương này sẽ trình bày thuật toán học nhằm ứng dụng huấn luyện mô hình học sâu RainPredRNN. Cùng với đó, thuật toán này sẽ được áp dụng đồng thời huấn luyện các mô hình còn lại. Ý tưởng, cơ sở lý thuyết, biến thể và chiến thuật huấn luyện sẽ được trình bày.

Trong các bài toán học máy, từ mô hình đề xuất ta luôn luôn phải thiết lập một hàm mất mát và phải giải quyết bài toán tối thiểu hàm mất mát đó để tìm được giá trị tối ưu cho các tham số mô hình. Gradient Descent chính là một phương pháp phổ biến và được ứng dụng nhiều nhất trong việc giải quyết bài toán tối ưu trên.

Xét bài toán tìm

Xét điểm bất kì thuộc D, khi đó, theo công thức khai triển Taylor của tại điểm , ta có:

với thuộc lân cận ,

Khi đó ta có thể xấp xỉ

Thay bởi , ta được:

Tức và ngược dấu. Vì thế ta có thể viết , với được gọi là tốc độ học. Trong thực tế, thường được chọn trước bởi người lập trình.

Diagram

Description automatically generated

Hình 3.3: Mô tả thuật toán gradient descent.

Hình 3.3 mô tả một cách trực quan quá trình thực hiện của thuật toán gradient descent. Bắt đầu từ một điểm khởi tạo đủ gần, ta thực hiện dịch chuyển nó theo ngược chiều đạo hàm để tìm được tối ưu. Ngoài ra, đối với việc sử dụng gradient, ta chỉ có thể tìm được điểm tối ưu cục bộ của hàm số (điểm cực tiểu) nếu hàm số không phải là hàm số lồi trên tập xác định, ta không thể tìm được làm cho giá trị hàm số là nhỏ nhất. Tuy nhiên, trong thực tế, điểm cực tiểu tìm được cũng đã cho ta những kết quả rất tốt. **Thuật toán 1** mô tả thuật toán gradient descent tổng quát.

|  |  |
| --- | --- |
| **Thuật toán 1**: Thuật toán Gradient Descent tổng quát | |
| Đầu vào: Tập dữ liệu , hàm mất mát , learning rate | |
| Đầu ra: | |
| Bắt đầu: Khởi tạo giá trị | |
| While ( chưa đủ tốt) | |
|  | Chọn ngẫu nhiên điểm dữ liệu |
|  | Tính toán đạo hàm |
|  | Cập nhật |
|  | Kiểm tra điều kiện dừng. Nếu đủ tốt, dừng vòng while. Trả về |
| Kết thúc. | |

**Adam Optimization**

Adaptive moment estimation (Adam), là một biến thể của gradient decent, là một phương pháp tối ưu dựa trên gradient khác nhằm adaptive learning rate cho các tham số trong quá trình học.

Ngoài việc tính toán trung bình có trọng số của bình phương gradient.

|  |  |  |
| --- | --- | --- |
|  |  | PT 3.1 |

Tác giả còn đề xuất sử dụng thêm trung bình có trọng số của gradient

|  |  |  |
| --- | --- | --- |
|  |  | PT 3.2 |

Trong đó, , được chọn bởi người lập trình và được chọn khá gần 1, , được khởi tạo bởi véc-tơ 0.

Tiếp đó, ta sẽ phải tính bias correct và ước tính thời điểm (moment estimation)

|  |  |  |
| --- | --- | --- |
|  |  | PT 3.3 |

Công thức cập nhật cuối cùng sẽ là

|  |  |  |
| --- | --- | --- |
|  |  | PT 3.4 |

**Minibatch Gradient Descent**

Minibatch Gradient Descent cũng sử dụng dữ liệu trong mỗi lần cập nhật epoch, , nhưng vẫn nhỏ hơn rất nhiều. Số này được gọi là batch size. Không có khái niệm minibatch-size. Như vậy sẽ có lần cập nhật tham số khi thực hiện một epoch. Đây là phương pháp được sử dụng nhiều và rộng rãi nhất trong việc training các mô hình học máy và học sâu. Số epoch và batch size là các *siêu tham số* (hyperparameter) cần phải chọn trước bởi người lập trình. Thông thường, batch size thường được chọn không quá lớn, và là lũy thừa cơ số 2: 16, 64, 256, v.v. **Thuật toán 2** mô tả các bước thực hiện của minibatch gradient descent.

|  |  |  |
| --- | --- | --- |
| **Thuật toán 2**: Thuật toán MiniBatch Gradient Descent | | |
| Đầu vào: Tập dữ liệu , hàm mất mát , learning rate , số epoch , batch size | | |
| Đầu ra: | | |
| Bắt đầu: Khởi tạo giá trị | | |
| For epoch | | |
|  | Số batch | |
|  | For | |
|  |  | Chọn ngẫu nhiên điểm dữ liệu gồm điểm dữ liệu |
|  |  | Tính toán đạo hàm |
|  |  | Cập nhật |
|  |  | Kiểm tra điều kiện dừng. Nếu đủ tốt, dừng vòng for. |
|  | Trả về | |
| Kết thúc. | | |

Một số điều kiện dừng thường được sử dụng trong quá trình huấn luyện một mô hình sử dùng thuật toán Gradient Descent là

* Sử dụng cố định số epoch. Chạy hết số epoch thì dừng việc huấn luyện.
* Dừng quá trình huấn luyện sớm bằng cách:
  + Đặt một *ngưỡng* (threshold) cố định đủ nhỏ. Nếu giá trị hàm mất mát đo được trên tập train giữa hai epoch nhỏ hơn giá trị đó thì dừng quá trình huấn luyện.
  + Đặt một *ngưỡng* cố định đủ nhỏ, nếu khoảng cách Euclid của các tham số học được ở hai epoch nhỏ hơn giá trị đó thì dừng quá trình huấn luyện.
* Chia tập huấn luyện thành hai phần nhỏ hơn. Ta gọi hai phần này là *tập huấn luyện* và *tập thẩm định* (validation set). Có rất nhiều cách để tạo ra hai tập như trên, cách đơn giản nhất thường được sử dụng là xáo trộn toàn bộ tập huấn luyện rồi chia tập huấn luyện và tập thẩm định theo một tỉ lệ chọn trước. Sử dụng tập huấn luyện để huấn luyện mô hình, trong quá trình huấn luyện, ta quan sát mất mát của mô hình trên tập thẩm định, nếu như mất mát nhỏ hơn mất mát ở epoch trước đó, ta tiến hành dừng quá trình huấn luyện. Cách làm này thường được dùng trong việc tránh hiện tượng *quá khớp* (overfitting) cho mô hình.

CHƯƠNG 4 sẽ trình bày chi tiết thử nghiệm mô hình đề xuất và các mô hình còn lại bằng thuật toán tối ưu Adam kết hợp với thuật toán MiniBatch Gradient Descent.

# KẾT QUẢ THỬ NGIỆM VÀ ĐÁNH GIÁ

## Dữ liệu đánh giá

Để huấn luyện mô hình đề xuất một cách hiệu quả và tạo ra hiệu suất mô hình cao, tập dữ liệu thu thập sẽ được chia được thành ba phần với tỷ lệ 80:10:10. Điều này có nghĩa là trong 2.429 hình ảnh của tập dữ liệu, tập huấn luyện, kiểm chứng và thử nghiệm sẽ chứa chính xác 1947, 242 và 242 hình ảnh tương ứng với tỷ lệ trên. Chi tiết về sự phân chia tập dữ liệu được trình bày trong Bảng 4.1 như sau:

Bảng 4.1: Kích thước các tập dữ liệu

|  |  |  |
| --- | --- | --- |
| **Tập dữ liệu** | **Số lượng** | **Kích thước** |
| Tập huấn luyện | 1947 | 150×150 |
| Tập kiểm chứng | 242 | 150×150 |
| Tập thử nghiệm | 22 | 150×150 |

Để tạo các hình ảnh làm đầu vào cho mô hình, tất cả hình ảnh sẽ được xếp chồng thành một mảng và lấy cửa sổ trượt liên tục của ngăn xếp tuần tự cho đến khi chỉ mục đạt đến ảnh cuối. Đối với mỗi phần liên tiếp, cần xác định một số khung của phần đầu là đầu vào và phần còn lại là đầu ra. Ví dụ, chọn các hình ảnh liên tiếp trong mảng có 10 khung hình cho mỗi cửa sổ, 5 khung cho đầu vào và 5 cho đầu ra.

## Công cụ cài đặt

### Python

Python [39] là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao (high-level) ngữ nghĩa động (dynamic semantics). Python hỗ trợ các module và gói (packages), khuyến khích chương trình module hóa và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sẵn dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối tự do.

Sau đây là các đặc điểm của Python:

* Ngữ pháp đơn giản, dễ đọc.
* Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented)
* Hỗ trợ module và hỗ trợ gói (package)
* Xử lý lỗi bằng ngoại lệ (Exception)
* Kiểu dữ liệu động ở mức cao.
* Có các bộ thư viện chuẩn và các module ngoài, đáp ứng tất cả các nhu cầu lập trình.
* Có khả năng tương tác với các module khác viết trên C/C++ (Hoặc Java cho Jython, hoặc .Net cho IronPython).
* Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

Python dễ dàng kết nối với các thành phần khác:

* Python có thể kết nối với các đối tượng COM, .NET (Ironpython, Python for .net), và CORBA, Java… Python cũng được hỗ trợ bởi Internet Communications Engine (ICE) và nhiều công nghệ kết nối khác.
* Có thể viết các thư viện trên C/C++ để nhúng vào Python và ngược lại.

Python là ngôn ngữ có khả năng chạy trên nhiều nền tảng.

* Python có cho mọi hệ điều hành: Windows, Linux/Unix, OS/2, Mac, Amiga, và những hệ điều hành khác. Thậm chí có cả những phiên bản chạy trên .NET, máy ảo Java, và điện thoại di động (Nokia Series 60). Với cùng một mã nguồn sẽ chạy giống nhau trên mọi nền tảng.

Python rất đơn giản và dễ học

* Python có cộng đồng lập trình rất lớn, hệ thống thư viện chuẩn, và cả các thư viện mã nguồn mở được chia sẻ trên mạng.

Python là ngôn ngữ mã nguồn mở

* Cài đặt Python dùng giấy phép nguồn mở nên được sử dụng và phân tối tự do, ngay cả trong việc thương mại. Giấy phép Python được quản lý bởi Python Software Foundation.

### PyTorch

PyTorch [40] được phát triển bởi Facebook và lần đầu tiên được phát hành công khai vào năm 2016. PyTorch được định nghĩa là một thư viện học máy mã nguồn mở cho Python. Nó được sử dụng cho các ứng dụng như xử lý ngôn ngữ tự nhiên. Ban đầu nó được phát triển bởi nhóm nghiên cứu trí tuệ nhân tạo của Facebook và phần mềm Uber’s Pyro để lập trình xác suất được xây dựng trên nó.

PyTorch thiết kế lại và triển khai Torch bằng Python trong khi chia sẻ cùng các thư viện C lõi cho mã back-end. Các nhà phát triển PyTorch đã điều chỉnh mã back-end này để chạy Python một cách hiệu quả. Họ cũng giữ nguyên khả năng tăng tốc phần cứng dựa trên GPU cũng như các tính năng mở rộng đã tạo nên Torch dựa trên Lua.

PyTorch nổi tiếng là được sử dụng rộng rãi trong nghiên cứu hơn là trong production. Tuy nhiên, kể từ khi được phát hành vào năm sau TensorFlow, PyTorch đã chứng kiến sự gia tăng mạnh mẽ về việc sử dụng của các nhà phát triển chuyên nghiệp.

Vì các lập trình viên Python thấy việc sử dụng rất tự nhiên nên PyTorch nhanh chóng thu hút được người dùng, tạo cảm hứng cho nhóm TensorFlow áp dụng nhiều tính năng phổ biến nhất của PyTorch trong TensorFlow 2.0.

Danh sách Khảo sát dành cho nhà phát triển Stack Overflow năm 2020 về “Các khung, Thư viện và Công cụ khác” phổ biến nhất báo cáo rằng 10,4% các nhà phát triển chuyên nghiệp chọn TensorFlow và 4,1% chọn PyTorch. Vào năm 2018, tỷ lệ phần trăm là 7,6 phần trăm cho TensorFlow và chỉ 1,6 phần trăm cho PyTorch. Đối với nghiên cứu, PyTorch là một lựa chọn phổ biến và các chương trình khoa học máy tính như Stanford’s hiện sử dụng nó để dạy học sâu.

Các tính năng chính của PyTorch được đề cập dưới đây:

* Giao diện thân thiệt - PyTorch cung cấp API dễ sử dụng; do đó nó được coi là rất đơn giản để vận hành và chạy trên Python. Việc thực thi mã trong khuôn mẫu này khá dễ dàng.
* Sử dụng Python - Thư viện này được coi là Pythonic tích hợp trơn tru với ngăn xếp khoa học dữ liệu Python. Do đó, nó có thể tận dụng tất cả các dịch vụ và chức năng được cung cấp bởi môi trường Python.
* Đồ thị tính toán - PyTorch cung cấp một nền tảng tuyệt vời cung cấp đồ thị tính toán động. Do đó người dùng có thể thay đổi chúng trong thời gian chạy. Điều này rất hữu ích khi một nhà phát triển không biết cần bao nhiêu bộ nhớ để tạo mô hình mạng nơ-ron.

PyTorch được biết đến với ba cấp độ trừu tượng như được đưa ra dưới đây:

* Tensor - Mảng n-chiều bắt buộc chạy trên GPU.
* Biến - Nút trong đồ thị tính toán. Điều này lưu trữ dữ liệu và gradient.

Mô-đun - Lớp mạng nơ-ron sẽ lưu trữ trạng thái hoặc trọng số có thể học được.

Ưu điểm của PyTorch:

* rất dễ dàng để gỡ lỗi và hiểu mã.
* Bao gồm nhiều lớp như Torch.
* Bao gồm rất nhiều hàm mất mát.
* Có thể coi đây là phần mở rộng NumPy cho GPU.
* Cho phép xây dựng các mạng có cấu trúc phụ thuộc vào chính tính toán.

## Cấu hình máy tính

Mã nguồn của mô hình sẽ được lập trình và thử nghiệm trên một máy trạm mạnh mẽ chạy trên hệ điều hành Windows 10: 64-bit. Máy được trang bị một card đồ họa GPU 12Gb GeForce RTX 2080Ti. Để chạy mô hình Học sâu đề xuất RainPredRNN trên GPU, máy trạm này cũng đã cài đặt CUDA driver phiên bản 10.1 tương thích có thể được tích hợp với card NVIDIA.

## Cài đặt thử nghiệm và lựa chọn siêu tham số

Trong phần này, để có thể dự đoán sáu khung hình tiếp theo (trước 1 giờ), tất cả các mô hình sẽ được thiết lập một cách phù hợp. Trên thực tế, sau khi tiến hành các thử nghiệm khác nhau, các mô hình sẽ được cài đặt và thử nghiệm trên các tham số sau:

**Phần mã hóa và giải mã**

Để làm rõ việc triển khai thử nghiệm một cách chi tiết, các siêu tham số của mô hình thực tế sẽ được mô tả phù hợp nhất với tập dữ liệu thu thập được. Mô hình đề xuất RainPredRNN bao gồm các đặc điểm quan trọng của kiến ​​trúc U-Net đã trình bày trong phần 3.5, trong đó kích thước nhân của các lớp tích chập đều được đặt thành 3×3, và cả stride và padding đều bằng 1.

**Phần PredRNN**

Dữ liệu thu thập với đặc điểm các vùng mưa di chuyển chạm và kích thước không quá lớn. Vì vậy, trong phần chính của mô hình, hai lớp ST-LSTM sẽ được đạt liên tiếp với nhau và được thiết lập với 64 trạng thái ẩn mỗi lớp và một bộ lọc 3×3 của các lớp tích chập bên trong. Ngoài ra, tổng độ dài đầu vào được cố định thành 12 khung hình, với sáu hình ảnh đầu liên tiếp đầu tiên cho đầu vào và sáu hình ảnh cuối cùng cho nhãn.

**Siêu tham số (Hyperparameters)**

Trong quá trình huấn luyện thử nghiệm, kết quả cho thấy các mô hình đã hội tụ trong khoảng 100 epoch. Do đó, để so sánh hiệu suất của tất cả các mô hình, tất cả mô hình sẽ được huấn luyện trong 100 epoch trong đó batch size được đặt bằng 4 và tốc độ học (learning rate) được đặt thành 0,001 trong toàn bộ giai đoạn huấn luyện. Tất cả các mô hình sẽ được đánh giá với các tiêu chí nêu trên. Ba mô hình: PredRNN, PredRNN\_v2, RainPredRNN - được triển khai và so sánh chi tiết trong phần 4.4 và 4.5.

## Kết quả và đánh giá

Trong quá trình thực nghiệm, kết quả đã cho thấy rằng tất cả các mô hình đều được hội tụ về cùng một điểm tại quá trình huấn luyện và kiểm chứng. Các giá trị này không quá sai khác nhau về độ lớn. Tại epoch cuối cùng, giá trị của hàm mất mát vào khoảng trong quá trình huấn luyện và trong quá trình kiểm chứng. Điều này chứng minh rằng, các mô hình đã được huấn luyện tốt và phù hợp với bộ dữ liệu. Để làm rõ sự phù hợp, các mô hình sau khi huấn luyện sẽ được thử nghiệm với ba độ đo đã nêu ở mục 1.9.

Chi tiết về các độ đo tiêu chuẩn đánh giá được liệt kê trong Bảng 4.2, trong đó MAE [28], CSI [30] và SSIM [29] được xác định với tập thử nghiệm. Từ bảng này, phép đo SSIM của tất cả các mô hình không có sự khác biệt đáng kể (khoảng ), có nghĩa là chất lượng của hình ảnh không bị suy giảm sau khi qua các mô hình, hình ảnh dự đoán sẽ có độ phân giải tương tự như nhãn. Từ đây có thể kết luận rằng sự hiệu quả của phần mã hóa và giải mã trong mô hình đề xuất. Việc thêm hai phần mã hóa và giải mã này đã làm giảm chiều không gian của dữ liệu đầu vào và không làm mất mát thông tin.

Bảng 4.2: Các độ đo giá mô hình đề xuất với những mô hình khác.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mô hình** | **MAE** | **CSI** | **SSIM** | **Thời gian huấn luyện (giờ)** | **MACs (G)** |
| PredRNN | 0.4535 | 0.9455 | 0.9397 | 15.1 | 101.469 |
| PredRNN\_v2 | 0.4157 | 0.9420 | 0.9430 | 15.53 | 103.885 |
| RainPredRNN | 0.4301 | 0.9590 | 0.9412 | **4.46** | **54.705** |

Chart, histogram

Description automatically generated

Hình 4.1: Giá trị mất mát trong quá trình huấn luyện của các mô hình.

Từ Bảng 4.2 ta thấy, mô hình RainPredRNN chỉ chiếm dưới 30% thời gian huấn luyện so với PredRNN và PredRNN\_v2. Điều này sẽ có ý nghĩa quan trọng trong tương lai nếu dữ liệu huấn luyện mới sinh ra và chúng ta muốn huấn luyện một phiên bản mới của mô hình. Giá trị MACs của RainPredRNN bằng một phần hai so với các giá trị của mô hình khác, vào khoảng 54 tỷ phép tính, vì vậy có thể kết luận rằng mô hình đề xuất đã làm giảm đáng kể các phép toán cần phải tính lại khi huấn luyện và thử nghiệm. Ngoài ra, mô hình đề xuất vẫn có hiệu suất tốt so với các phiên bản trước. Từ đó, các mô hình này chắc chắn sẽ tạo ra hình ảnh dự đoán với chất lượng và độ phân giải cao.

## Hình ảnh kết quả

Trong phần này, các mô hình so sánh sẽ được kiểm chứng với hình ảnh trong tập dữ liệu thử nghiệm. Dữ liệu đầu vào và đầu ra được mô tả lần lượt trong hình Hình 4.2 và Hình 4.3.

Graphical user interface, application, PowerPoint

Description automatically generated

Hình 4.2: Dữ liệu đầu vào thử nghiệm 6 ảnh đầu vào liên tiếp nhau.

Graphical user interface, application

Description automatically generated

Hình 4.3: Dữ liệu đầu ra kiểm chứng 6 ảnh đầu ra liên tiếp nhau.

Hình ảnh dự đoán của mô hình PredRNN, PredRNN\_v2 và RainPredRNN được mô tả lần lượt trong Hình 4.4, Hình 4.5 và Hình 4.6. Chất lượng của mô hình đề xuất có xu hướng có độ phân giải cao và chính xác hơn chất lượng của các mô hình PredRNN và PredRNN\_v2.

Graphical user interface, application

Description automatically generated

Hình 4.4: Kết quả ảnh dự báo của mô hình PredRNN.

Graphical user interface, application

Description automatically generated

Hình 4.5: Kết quả ảnh dự báo của mô hình PredRNN\_v2.

Graphical user interface

Description automatically generated with medium confidence

Hình 4.6: Kết quả ảnh dự báo của mô hình RainPredRNN.

Kết quả cho thấy họ mô hình PredRNN phù hợp với vấn đề Dự báo ảnh mưa và mô hình đề xuất có thể giúp giảm đáng kể việc huấn luyện và thử nghiệm và tạo ra hình ảnh tương lai chất lượng cao trong thời gian ngắn.

## Demo ứng dụng dự báo ảnh mưa

Nhằm hiện thực hóa ứng dụng của mô hình đề xuất RainPredRNN, một ứng dụng đơn giản trên nền tảng website được thiết kế và cài đặt. Ứng dụng sẽ là một chức năng quan trọng trong hệ thống dự báo lớn của các trạm quan sát hay trung tâm dự báo. Với dự mạnh mẽ đã được chứng minh, mô hình đề xuất sẽ cho ra kết quả có độ phân giải cao trong thời gian ngắn. Kết quả dự báo của mô hình sẽ được ứng dụng cho dự báo thời tiết, cảnh báo lịch trình, … tại vùng quan sát của các trạm thu ảnh radar. Ứng dụng sử dụng các công nghệ lập trình web hiện đại như:

* Phần Backend: Python, Khung mẫu Django.
* Phần Frontend: Bootstrap, HTML 5, Jquery, Javascript.
* Phần Cơ sở dữ liệu: SQLite 3.

Lý do khung mẫu Django được sử dụng là Django được viết bằng ngôn ngữ Python, điều này sẽ tạo điều kiện thuận lợi cho việc cài đặt và thích hợp mô hình học sâu vào ứng dụng. Phần chính của ứng dụng là mô hình Học sâu RainPredRNN. Ứng dụng là một website nhỏ với các thành phần hoạt đột được mô tả như sau:

* Đầu vào: 6 ảnh radar có bước thời gian liên tiếp nhau: t-5, t-4, …, t.
* Phần xử lý: mô hình đã huấn luyện RainPredRNN
* Đầu ra: 6 ảnh radar dự báo có bước thời gian liên tiếp nha: t+1, t+2, …, t+6.

Hình 4.9 dưới đây mô tả trực quan dứng dụng. Các thức hoạt động: người dùng sẽ tải lên 6 ảnh theo thứ tự mốc thời gian từ máy tính và gửi lên ứng dụng. Ứng dụng sẽ sử dụng mô hình học sâu được huấn luyện từ trước và cho ra kết quả dự báo. Kết quả sẽ được trả lại và hiển thị trên màn hình.

Graphical user interface

Description automatically generated

Hình 4.7: Demo ứng dụng dự báo ảnh mưa của mô hình đề xuất PredRNN.

## Nhận xét chung

Nhìn chung, các mô hình cho kết quả tốt với các cấu hình thực nghiệm. Xong nhìn vào kết quả cho thấy thấy được cải tiến rõ ràng về mặt thời gian xử lý. Mô hình RainPredRNN đã cho kết quả tốt, có thể so sánh được với các mô hình tiền nhiệm. Bên cạnh đó, số lượng phép tính phải thực hiện đã giảm đi đáng kể, từ đó cho ra thời gian huấn luyện và dự báo giảm đi một lượng lớn. Với việc cho ra kết quả tốt trong thời gian ngắn, mô hình sẽ đóng góp đáng kể vào bài toán thực tế của các lĩnh vực liên quan cho các dự báo ngắn hạn như: lịch trình, nông nghiệp, du lịch, …

# KẾT LUẬN

## Kết luận

Đồ án đã đề xuất một mô hình Học sâu mới có tên là RainPredRNN để dự báo ảnh mưa bằng hình ảnh tiếng vọng radar thời tiết. Đồ án đã đưa ra một mô hình đồng nhất là sự kết hợp của U-Net và PredRNN\_v2 với mục đích giảm thời gian huấn luyện và thử nghiệm trong khi vẫn bảo toàn các đặc trưng không gian phức tạp của dữ liệu radar. RainPredRNN quản lý cả thông tin không gian và thời gian của hình ảnh trong chuỗi thời gian. Bên cạnh đó, lớp mã hóa và giải mã của UNet có vai trò quan trọng trong việc giảm kích thước đầu vào trong khi nó vẫn nắm bắt được các đặc trưng cấp cao của hình ảnh gốc. Các thử nghiệm trên dữ liệu thực từ trạm radar thời tiết Pha Đin, đặt tại tỉnh Điện Biên, Việt Nam đã khẳng định rằng RainPredRNN giảm đáng kể việc huấn luyện và thử nghiệm, đồng thời tạo ra hình ảnh chuỗi thời gian tuong lai chất lượng cao trong thời gian ngắn.

## Định hướng phát triển

Tuy nhiên, vẫn còn một số hạn chế của mô hình như chưa cải tiến được hiệu suất trên các chỉ số so sánh. Trong tương lai, mô hình sẽ xem xét phần lõi xử lý ST-LSTM và sẽ được cải tiến nhằm tăng độ chính xác và các độ đo đánh giá của mô hình để dự báo ảnh mưa. Bên cạnh đó, mô hình cũng sẽ được cải tiến nhằm giải quyết các bài toán liên quan đến ảnh vệ tinh

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | E. Chung, O. Ohtani, H. Warita, M. Kuwahara and H. Morita, "Effect of rain on travel demand and traffic accidents," *In Proceedings of IEEE Intelligent Transportation Systems,* pp. 1080-1083, 2005. |
| [2] | K. Keay and I. Simmonds, "Road accidents and rainfall in a large Australian city," *Accident Analysis & Prevention 2006,* p. 38, 445-454. |
| [3] | Q. Sun, C. Miao, Q. Duan, H. Ashouri, S. Sorooshian and K.-L. Hsu, "A Review of Global Precipitation Data Sets: Data Sources, Estimation, and Intercomparisons," *Rev. Geophys,* pp. 79-107, 2018. |
| [4] | Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature ,* pp. 436-444, 2015. |
| [5] | M. Sit, B. Demiray, Z. Xiang, G. Ewing, Y. Sermet and I. Demir, "A Comprehensive Review of Deep Learning Applications in Hydrology and Water Resources," *ArXiv,* 2020. |
| [6] | Z. Gao, X. Shi, H. Wang, D.-Y. Yeung, W.-c. Woo and W.-K. Wong, "Deep Learning and the Weather Forecasting Problem: Precipitation Nowcasting," *In Deep Learning for the Earth Sciences,* 2021. |
| [7] | K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ArXiv,* 2014. |
| [8] | Z.-Q. Zhao, P. Zheng, S.-t. Xu and X. Wu, "Object Detection with Deep Learning: A Review," *ArXiv ,* 2019. |
| [9] | X. Le, G. Lee, K. Jung, H.-u. An, S. Lee and Y. Jung, "Application of Convolutional Neural Network for Spatiotemporal Bias Correction of Daily Satellite-Based Precipitation," *Remote Sens,* 2020. |
| [10] | G. Ayzel, T. Scheffer and M. Heistermann, "RainNet v1.0: a convolutional neural network for radar-based precipitation nowcasting," *Geosci,* pp. 2631-2644, 2020. |
| [11] | L. Khiali, D. Ienco and M. Teisseire, "Object-oriented satellite image time series analysis using a graph-based representation," *Ecol. Inform,* pp. 52-64, 2018. |
| [12] | S. Fahim, Y. Sarker, S. Sarker, M. Sheikh and S. Das, "Self attention convolutional neural network with time series imaging based feature extraction for transmission line fault detection and classification," *Electric Power Systems Research,* 2020. |
| [13] | X. Li, Y. Kang and F. Li, "Forecasting with time series imaging," *Expert Systems with Applications,* 2020. |
| [14] | S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem and S. e. a. Madge, "Skilful precipitation nowcasting using deep generative models of radar," *Nature,* pp. 672-677, 2021. |
| [15] | D. Li, Y. Liu and C. Chen, "MSDM v1.0: A machine learning model for precipitation nowcasting over eastern China using multisource data," *Geosci. Model,* pp. 4019-4034, 2021. |
| [16] | L. Chen, Y. Cao, L. Ma and J. Zhang, "A Deep Learning-Based Methodology for Precipitation Nowcasting With Radar," *Earth and Space Science,* 2020. |
| [17] | S. Agrawal, L. Barrington, C. Bromberg, J. Burge, C. Gazen and J. Hickey, "Machine Learning for Precipitation Nowcasting from Radar Images," *ArXiv,* 2019. |
| [18] | J. Fernández and S. Mehrkanoon, "Broad-UNet: Multi-scale feature learning for nowcasting tasks," *Neural Networks,* pp. 419-427, 2021. |
| [19] | V.-S. Ionescu, G. Czibula and E. Mihuleţ, "DeePS at: A deep learning model for prediction of satellite images for nowcasting purposes," *Procedia Computer Sci.,* pp. 622-631, 2021. |
| [20] | L. Zhang, Z. Huang, W. Liu, Z. Guo and Z. Zhang, "Weather radar echo prediction method based on convolution neural network and Long Short-Term memory networks for sustainable e-agriculture," *Journal of Cleaner Production,* 2021. |
| [21] | K. Trebing, T. Staǹczyk and S. Mehrkanoon, "SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture," *Pattern Recognition Letters,* pp. 178-186, 2021. |
| [22] | X. Le, H. Ho, G. Lee and S. Jung, "Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting," *Water ,* 2019. |
| [23] | X. Le, D. Nguyen, S. Jung, M. Yeon and G. Lee, "Comparison of Deep Learning Techniques for River Streamflow Forecasting," *IEEE Access,* 2021. |
| [24] | Y. Wang, H. Wu, J. Zhang, Z. Gao, J. Wang, P. Yu and M. Long, "PredRNN: A Recurrent Neural Network for Spatiotemporal Predictive Learning," *ArXiv ,* 2021. |
| [25] | J. Kreklow, B. Tetzlaff, B. Burkhard and G. Kuhnt, "Radar-Based Precipitation Climatology in Germany—Developments, Uncertainties and Potentials," *Atmosphere 2020,* p. 11, 2020. |
| [26] | S. Otsuka, S. Kotsuki, M. Ohhigashi and T. Miyoshi, "GSMaP RIKEN Nowcast: Global Precipitation Nowcasting with Data Assimilation," *Journal of the Meteorological Society of Japan,* pp. 1099-1117, 2019. |
| [27] | O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *ArXiv ,* 2015. |
| [28] | T. Chai and R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," *Geosci. Model Dev,* pp. 1247-1250, 2014. |
| [29] | Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process,* pp. 600-612, 2004. |
| [30] | J. Schaefer, "The Critical Success Index as an Indicator of Warning Skill," *Weather and Forecasting 1990,* pp. 570-575, 1990. |
| [31] | J. Townsend, "Theoretical analysis of an alphabetic confusion matrix," *Perception & Psychophysics 1971,* pp. 40-50, 1971. |
| [32] | V. H. Tiệp, "Multi-layer Perceptron và Backpropagation," [Online]. Available: https://machinelearningcoban.com/2017/02/24/mlp/.. |
| [33] | A. A. a. A. Shervine, "Mạng neural tích chập cheatsheet," [Online]. Available: https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-convolutional-neuralnetworksl.. |
| [34] | K. V. M. B. G. C. B. D. B. F. S. H. &. B. Y. Cho, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv ,* 2014. |
| [35] | S. a. J. S. Hochreiter, "Long short-term memory," *Neural computation 9,* p. 1735–1780, 1997. |
| [36] | X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong and W.-c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *ArXiv,* 2015. |
| [37] | Y. Wang, M. Long, J. Wang, Z. Gao and P. Yu, "PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs," *arXiv,* 2017. |
| [38] | L. Maaten and G. Hinton, "Visualizing Data using t-SNE," *J. Mach. Learn. Res. ,* pp. 2579-2605, 2008. |
| [39] | "Python," [Online]. Available: https://www.python.org/. |
| [40] | Facebook, "FROM RESEARCH TO PRODUCTION," Facebook, [Online]. Available: https://pytorch.org/. |