TUYỂN TẬP BÁO CÁO HỘI NGHỊ TOÀN QUỐC

KHOA HỌC TRÁI ĐẤT VÀ TÀI NGUYÊN VỚI PHÁT TRIỂN BỀN VỮNG

Hà Nội, 12 - 11 - 2020

ERSD 2020

NHÀ XUẤT BẢN GIAO THÔNG VẬN TẢI

EARTH SCIENCES AND
NATURAL RESOURCES FOR
SUSTAINABLE DEVELOPMENT

TUYỂN TẬP BÁO CÁO HỘI NGHỊ TOÀN QUỐC
KHOA HỌC TRÁI ĐẤT VÀ TÀI NGUYÊN VỚI PHÁT TRIỂN BỀN VỮNG

TIỂU BAN
PHÂN TÍCH DỮ LIỆU VÀ HỌC MÁY

ĐƠN VỊ TỔ CHỨC

Trường Đại học Mỏ - Địa chất (HUMG)

CÁC ĐƠN VỊ PHỐI HỢP TỔ CHỨC

Tập đoàn Công nghiệp Than - Khoáng sản Việt Nam
Tập đoàn Dầu khí Việt Nam
Tổng cục Địa chất và Khoáng sản Việt Nam
Tổng hội Địa chất Việt Nam
Cục Đo đạc, Bản đồ và Thông tin địa lý Việt Nam
Hội Khoa học Công nghệ Mỏ Việt Nam
Hội Công trình ngầm Việt Nam
Hội Địa chất Thủy văn Việt Nam
Hội Địa chất Công trình và Môi trường Việt Nam
Hội Kỹ thuật Nổ mìn Việt Nam
Hội Khoa học Kỹ thuật Địa vật lý Việt Nam
Hội Trắc địa - Bản đồ - Viễn thám Việt Nam
Viện Địa chất và Địa vật lý biển
Viện Khoa học Địa chất và Khoáng sản
Trường Đại học Công nghệ Đồng Nai
Trường Đại học Đông Á
Trường Đại học Thủ Dầu Một

BAN TỔ CHỨC

Trưởng ban
GS.TS Trần Thanh Hải, *Trường Đại học Mỏ Địa - chất*

Phó Trưởng ban
GS.TS Bùi Xuân Nam, *Trường Đại học Mỏ - Địa chất*
PGS.TS Triệu Hùng Trường, *Trường Đại học Mỏ - Địa chất*

Ủy viên
GS.TS Võ Chí Mỹ, *Hội Trắc địa - Bản đồ - Viễn thám Việt Nam*
GS.TS Nguyễn Quang Phích, *Hội Công trình ngầm Việt Nam*
PGS.TS Trần Tuấn Anh, *Viện Địa chất, Viện HLKH&CN Việt Nam*
PGS.TS Đoàn Văn Cánh, *Hội Địa chất Thủy văn Việt Nam*
PGS.TS Tạ Đức Thịnh, *Hội Địa chất Công trình và Môi trường Việt Nam*
PGS.TS Nguyễn Như Trung, *Viện Địa chất và Địa vật lý biển, Hội Khoa học kỹ thuật Địa vật lý Việt Nam*
TS Nguyễn Đại Đồng, *Cục Đo đạc, Bản đồ và Thông tin địa lý Việt Nam*
TS Trần Xuân Hòa, *Hội Khoa học và Công nghệ Mỏ Việt Nam*
TS Hoàng Văn Khoa, *Tổng hội Địa chất Việt Nam*
TS Đỗ Hồng Nguyên, *Tập đoàn Công nghiệp Than - Khoáng sản Việt Nam*
TS Nguyễn Văn Nguyên, *Tổng cục Địa chất và Khoáng sản Việt Nam*
TS Lê Văn Quyền, *Hội Kỹ thuật Nổ mìn Việt Nam*
TS Trịnh Hải Sơn, *Viện Khoa học Địa chất và Khoáng sản, Bộ Tài nguyên và Môi trường*
TS Nguyễn Quốc Thập, *Tập đoàn Dầu khí quốc gia Việt Nam*
TS Đặng Kim Triết, *Trường Đại học Công nghệ Đồng Nai*
TS Trần Văn Trung, *Trường Đại học Thủ Dầu Một*
TS Đỗ Trọng Tuấn, *Trường Đại học Đông Á*
TS Nguyễn Thanh Tùng, *Viện Dầu khí Việt Nam*

BAN KHOA HỌC

Trưởng ban

GS.TS Bùi Xuân Nam, *Trường Đại học Mỏ - Địa chất*

Phó trưởng ban

PGS.TS. Đỗ Ngọc Anh, *Trường Đại học Mỏ - Địa chất*

Ủy viên

GS.TSKH Hoàng Ngọc Hà, *Trường Đại học Mỏ - Địa chất*
GS.TS Võ Trọng Hùng, *Trường Đại học Mỏ - Địa chất*
GS.TS Trương Xuân Luận, *Trường Đại học Mỏ - Địa chất*
GS.TS Đỗ Như Tráng, *Trường Đại học Công nghệ GTVT*
PGS.TS Bùi Hoàng Bắc, *Trường Đại học Mỏ - Địa chất*
PGS.TS Đỗ Văn Bình, *Trường Đại học Mỏ - Địa chất*
PGS.TS Phùng Mạnh Đắc, *Hội KHCN Mỏ Việt Nam*
PGS.TSKH Hà Minh Hòa, *Viện Khoa học Đo đạc và Bản đồ*
PGS.TS Phạm Văn Hòa, *Trường Đại học Mỏ - Địa chất*
PGS.TS Lê Văn Hưng, *Trường Đại học Mỏ - Địa chất*
PGS.TS Hoàng Văn Long, *Viện Dầu khí Việt Nam*
PGS.TS Phạm Văn Luận, *Trường Đại học Mỏ - Địa chất*
PGS.TS Nguyễn Quang Minh, *Trường Đại học Mỏ - Địa chất*
PGS.TS Phạm Xuân Núi, *Trường Đại học Mỏ - Địa chất*
PGS.TS Khổng Cao Phong, *Trường Đại học Mỏ - Địa chất*
PGS.TS Nguyễn Văn Sáng, *Trường Đại học Mỏ - Địa chất*
PGS.TS Ngô Xuân Thành, *Trường Đại học Mỏ - Địa chất*
PGS.TS Đặng Trung Thành, *Trường Đại học Mỏ - Địa chất*
PGS.TS Tạ Đức Thịnh, *Hội Địa chất Công trình và Môi trường Việt Nam*
PGS.TS Nguyễn Thế Vinh, *Trường Đại học Mỏ - Địa chất*
TS Lê Hồng Anh, *Trường Đại học Mỏ - Địa chất*
TS Trần Quốc Cường, *Viện Địa chất, Viện HLKH&CN Việt Nam*
TS Công Tiến Dũng, *Trường Đại học Mỏ - Địa chất*
TS Trần Tuấn Dũng, *Viện Địa chất và Địa vật lý biển, Viện HL KH&CN Việt Nam*
TS Nguyễn Đại Đồng, *Cục Đo đạc, Bản đồ và Thông tin địa lý Việt Nam*
TS Nguyễn Mạnh Hùng, *Trường Đại học Mỏ - Địa chất*
TS Nguyễn Quốc Phi, *Trường Đại học Mỏ - Địa chất*
TS Bùi Thị Thu Thủy, *Trường Đại học Mỏ - Địa chất*
TS Nguyễn Thế Truyện, *Viện NC Điện tử, Tin học, Tự động hóa*
TS Nguyễn Văn Xô, *Trường Đại học Mỏ - Địa chất*

BAN BIÊN TẬP

Trưởng ban

TS Nguyễn Viết Nghĩa, *Trường Đại học Mỏ - Địa chất*

Phó Trưởng ban

TS Nguyễn Thạc Khánh, *Trường Đại học Mỏ - Địa chất*

Ủy viên

PGS.TS Bùi Hoàng Bắc, *Trường Đại học Mỏ - Địa chất*
PGS.TS Phạm Văn Luận, *Trường Đại học Mỏ - Địa chất*
PGS.TS Trần Tuấn Minh, *Trường Đại học Mỏ - Địa chất*
PGS.TS Bùi Ngọc Quý, *Trường Đại học Mỏ - Địa chất*
PGS.TS Đỗ Như Ý, *Trường Đại học Mỏ - Địa chất*
TS Nguyễn Thị Mai Dung, *Trường Đại học Mỏ - Địa chất*
TS Nguyễn Mạnh Hùng, *Trường Đại học Mỏ - Địa chất*
TS Phạm Trung Kiên, *Trường Đại học Mỏ - Địa chất*
TS Nguyễn Quốc Phi, *Trường Đại học Mỏ - Địa chất*

BAN THƯ KÝ

Trưởng ban

PGS.TS Đỗ Ngọc Anh, *Trường Đại học Mỏ - Địa chất*

Phó Trưởng ban

TS Nguyễn Thạc Khánh, *Trường Đại học Mỏ - Địa chất*

Ủy viên

PGS.TS Phạm Văn Luận, *Trường Đại học Mỏ - Địa chất*
PGS.TS Nguyễn Văn Sáng, *Trường Đại học Mỏ - Địa chất*
TS Tô Xuân Bản, *Trường Đại học Mỏ - Địa chất*
TS Nguyễn Trọng Dũng, *Trường Đại học Mỏ - Địa chất*
TS Lê Quang Duyến, *Trường Đại học Mỏ - Địa chất*
TS Nguyễn Duy Huy, *Trường Đại học Mỏ - Địa chất*
TS Nguyễn Quốc Phi, *Trường Đại học Mỏ - Địa chất*
TS Ngô Thanh Tuấn, *Trường Đại học Mỏ - Địa chất*
TS Nguyễn Mạnh Hùng, *Trường Đại học Mỏ - Địa chất*
ThS Nguyễn Ngọc Dung, *Trường Đại học Mỏ - Địa chất*
ThS Hoàng Thu Hằng, *Trường Đại học Mỏ - Địa chất*
ThS Nguyễn Thanh Hải, *Trường Đại học Mỏ - Địa chất*
ThS Phạm Đức Nghiệp, *Trường Đại học Mỏ - Địa chất*

LỜI NÓI ĐẦU

Hội nghị Toàn quốc Khoa học Trái đất và Tài nguyên với Phát triển bền vững - ERSD được Trường Đại học Mỏ - Địa chất (HUMG) và các đối tác tổ chức 2 năm một lần để các nhà chuyên môn trong và ngoài nước tụ hội, giới thiệu những kết quả và hướng nghiên cứu khoa học mới, thảo luận về các xu thế phát triển, thách thức và cơ hội mới đối với nhiều lĩnh khác nhau của Khoa học Trái đất, Tài nguyên và các ngành khác có liên quan.

Tiếp nối thành công của Hội nghị lần thứ nhất năm 2018 (ERSD 2018) và được sự cho phép của Bộ Giáo dục và Đào tạo, Hội nghị Toàn quốc Khoa học Trái đất và Tài nguyên với Phát triển bền vững lần thứ hai (ERSD 2020) được Trường Đại học Mỏ - Địa chất (HUMG) đăng cai tổ chức với sự phối hợp đồng tổ chức của nhiều đơn vị quản lý, nghiên cứu khoa học, đào tạo và sản xuất có uy tín trong nước gồm Tập đoàn Công nghiệp Than - Khoáng sản Việt Nam, Tập đoàn Dầu khí Quốc gia Việt Nam, Tổng cục Địa chất và Khoáng sản Việt Nam, Cục Đo đạc, Bản đồ và Thông tin địa lý Việt Nam, Viện Địa chất và Địa vật lý biển, Viện Khoa học Địa chất và Khoáng sản, Trường Đại học Công nghệ Đồng Nai, Trường Đại học Đông Á, Trường Đại học Thủ Dầu Một, Tổng hội Địa chất Việt Nam, Hội Khoa học Công nghệ Mỏ Việt Nam, Hội Công trình ngầm Việt Nam, Hội Địa chất Thủy văn Việt Nam, Hội Địa chất Công trình và Môi trường Việt Nam, Hội Kỹ thuật Nổ mìn Việt Nam, Hội Khoa học Kỹ thuật Địa vật lý Việt Nam, Hội Trắc địa - Bản đồ - Viễn thám Việt Nam, và với sự tham gia của nhiều tổ chức và cá nhân khác.

Các chủ đề chính của Hội nghị lần này tập trung vào thảo luận các kết quả khoa học công nghệ và hướng nghiên cứu mới của Khoa học Trái đất và Tài nguyên thiên nhiên, Khai thác và sử dụng tài nguyên địa chất, Môi trường và các lĩnh vực khoa học khác có liên quan như Cơ - Điện, Công nghệ Thông tin, Xây dựng, … cũng như việc ứng dụng chúng vào phát triển bền vững đối với nhiều lĩnh vực khác nhau của khoa học công nghệ, kinh tế và xã hội.

Trong quá trình tổ chức Hội nghị, Ban Tổ chức đã nhận được sự quan tâm của đông đảo các nhà khoa học, chuyên môn và quản lý trong và ngoài nước, trong đó có hơn 300 báo cáo khoa học liên quan tới các chủ đề của Hội nghị đã được gửi tới Ban biên tập. Trên cơ sở đó, 255 báo cáo có chất lượng đã được lựa chọn và xuất bản trong Tuyển tập tóm tắt các báo cáo và Tuyển tập các báo cáo toàn văn của Hội nghị. Báo cáo toàn văn được tập hợp thành 16 tập, mỗi tập ứng với một chủ đề khoa học sau:

1. Địa chất khu vực
2. Địa chất công trình - Địa chất thủy văn
3. Tài nguyên địa chất và phát triển bền vững
4. Môi trường trong khai thác tài nguyên và phát triển bền vững
5. An toàn mỏ
6. Công nghệ và thiết bị khai thác
7. Thu hồi và chế biến khoáng sản
8. Công trình ngầm và Địa kỹ thuật
9. Vật liệu và kết cấu
10. Kỹ thuật dầu khí tích hợp
11. Trắc địa
12. Bản đồ, Viễn thám và Hệ thống thông tin địa lý
13. Khoa học Cơ bản trong lĩnh vực Khoa học Trái đất và Môi trường
14. Cơ khí, điện và Tự động hóa
15. Công nghệ thông tin
16. Phân tích dữ liệu và học máy

Toàn bộ thông tin khoa học về hội nghị, trong đó có Tuyển tập các báo cáo toàn văn, được đưa lên trang Website chính thức của Hội nghị tại địa chỉ: http://ersd2020.humg.edu.vn/.

Ban tổ chức xin trân trọng cảm ơn Trường Đại học Mỏ - Địa chất, với tư cách là đơn vị đăng cai tổ chức Hội nghị, cùng các đơn vị đồng tổ chức đã hợp tác và góp phần quan trọng vào sự thành công của Hội nghị này. Cảm ơn các nhà khoa học đã đóng góp các công bố khoa học có giá trị cho Hội nghị. Ban tổ chức cũng đánh giá cao sự nỗ lực của Ban biên tập và các chuyên gia biên tập để nâng cao chất lượng của các báo cáo khoa học cũng như sự cố gắng lớn của Ban thư ký trong việc chuẩn bị và tổ chức hội nghị này.

Ban tổ chức mong muốn tiếp tục nhận được sự hợp tác chặt chẽ và góp ý chân thành của các đơn vị và cá nhân đối với việc chuẩn bị, tổ chức, biên tập, và xuất bản các báo cáo khoa học, nhằm nâng cao chất lượng của các hội nghị tiếp theo, góp phần thúc đẩy sự phát triển bền vững của các hoạt động nghiên cứu khoa học, chuyển giao công nghệ thuộc các lĩnh vực Khoa học Trái đất và Tài nguyên và các lĩnh vực khoa học khác có liên quan.

TRƯỞNG BAN TỔ CHỨC

GS.TS Trần Thanh Hải

MỤC LỤC

TIỂU BAN
PHÂN TÍCH DỮ LIỆU VÀ HỌC MÁY

Nghiên cứu sử dụng giải thuật di truyền để lựa chọn các tham số cho học sâu
Bùi Thị Vân Anh, Đặng Hữu Nghị 01

Đánh giá độ chính xác của một số phương pháp xử lý dữ liệu thiếu cho dữ liệu chuỗi thời gian (time series data)
Nguyễn Thị Phương Bắc, Đặng Văn Nam 08

Integrated flow shop and vehicle routing problem
Ta Quang Chieu, Ta Xuan Giang, Ha Thi Thu Hien, Tran Thi Hieu, Pham Duc Hau 18

Bảo mật dữ liệu trong Điện toán đám mây
Đỗ Như Hải 24

Nghiên cứu mô hình học máy Naïve Bayes trong phân lớp văn bản; Ứng dụng phân lớp cho tập dữ liệu các nhận xét trên Twitter
Đặng Văn Nam 29

Xây dựng chương trình minh họa hỗ trợ giảng dạy và học tập môn Trí tuệ nhân tạo
Đặng Hữu Nghị, Bùi Thị Vân Anh, Phạm Đức Hậu, Đặng Quốc Trung 37

Nghiên cứu mạng học sâu sử dụng Keras trong nhận dạng hình ảnh
Phạm Đình Tân, Trần Thị Thu Thúy, Diêm Công Hoàng 43

Phát triển thuật toán nâng cao chất lượng tiếng nói thời gian thực
Dương Thị Hiền Thanh, Trần Thanh Huân 48

HỘI NGHỊ TOÀN QUỐC KHOA HỌC TRÁI ĐẤT
VÀ TÀI NGUYÊN VỚI PHÁT TRIỂN BỀN VỮNG (ERSD 2020)

Nghiên cứu sử dụng giải thuật di truyền để lựa chọn các tham số cho học sâu

Bùi Thị Vân Anh[1,*], Đặng Hữu Nghị[1]
[1]*Trường Đại học Mỏ - Địa chất*

TÓM TẮT

Học máy (machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Học sâu (Deep learning) là một chi của học máy, xây dựng mô hình bằng cách sử dụng nhiều lớp nơ ron với cấu trúc biến đổi phi tuyến. Để có thể đạt được kết quả tốt, khi thiết kế các mạng học sâu, chúng ta phải tối ưu hóa cấu trúc của mạng bằng cách lựa chọn các tham số phù hợp. Trong báo cáo này, chúng tôi đề xuất giải pháp sử dụng giải thuật di truyền để lựa chọn các tham số cho học sâu.

Từ khóa: Học máy; học sâu; trí tuệ nhân tạo; dữ liệu; tối ưu hóa cấu trúc; giải thuật di truyền.

1. Mở đầu

Mạng tích chập (Convolution neural network - CNN) đã đạt được một thành công đáng kể trong các bài toán phân loại hình ảnh trong những năm gần đây. Hiệu suất của CNN phụ thuộc rất nhiều vào kiến trúc của nó. Mạng tích chập (CNN) rất nhạy cảm với việc lựa chọn các tham số của chúng. Tuy nhiên, rất khó xác định kiến trúc mạng lý tưởng. Không có quy tắc rõ ràng về số lượng nơ ron ở các lớp trung gian, số lượng lớp trung gian và cách kết nối giữa các nơ ron.

Các phương pháp chọn tham số truyền thống bao gồm tìm kiếm thủ công, tìm kiếm lưới hoặc tìm kiếm ngẫu nhiên, nhưng những phương pháp này đòi hỏi kiến thức chuyên môn hoặc nặng nề về tính toán.

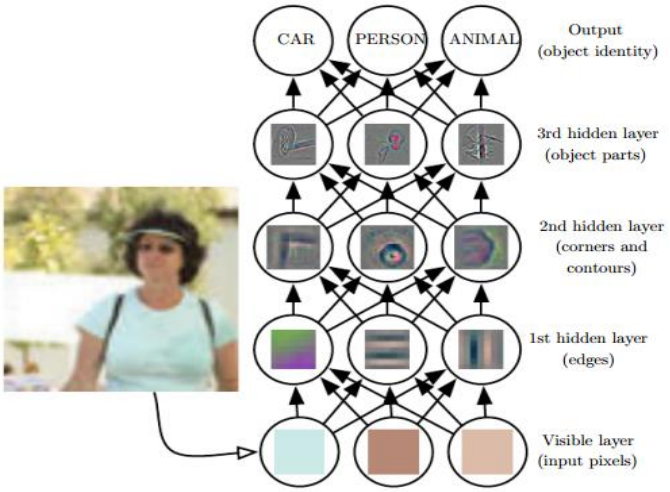
Trong bài báo này, chúng tôi đề xuất một phương pháp thiết kế kiến trúc CNN bằng cách sử dụng giải thuật di truyền để lựa chọn các tham số tốt cho CNN, để nâng cao hiệu quả cho các bài toán phân loại ảnh.

2. Phương pháp học sâu (deep learning)

2.1. Giới thiệu về deep learning

Học sâu là một nhánh đặc biệt của ngành học máy, trở nên phổ biến trong thập kỷ gần đây do các nhà khoa học đã có thể tận dụng khả năng tính toán mạnh mẽ của các máy tính hiện đại cũng như khối lượng dữ liệu khổng lồ (hình ảnh, âm thanh, văn bản,…) trên Internet (Adit Deshpande, 2016; Andrej Karpathy).

Các mạng huấn luyện theo phương pháp Học sâu còn được gọi với cái tên khác là mạng nơ-ron sâu Deep Neural Network) do cách thức hoạt động của chúng. Về cơ bản, các mạng này bao gồm rất nhiều lớp khác nhau, mỗi lớp sẽ phân tích dữ liệu đầu vào theo các khía cạnh khác nhau và theo mức độ trừu tượng nâng cao dần.



Hình 1. Mức độ trừu tượng tăng dần qua các tầng học của Học sâu

* *Tác giả liên hệ*
Email: buithivananh@humg.edu.vn

Cụ thể, với một mạng Học sâu cho nhận dạng ảnh, các lớp đầu tiên trong mạng chỉ làm nhiệm vụ rất đơn giản là tìm kiếm các đường thẳng, đường cong, hoặc đốm màu trong ảnh đầu vào. Các thông tin này sẽ được sử dụng làm đầu vào cho các lớp tiếp theo, với nhiệm vụ khó hơn là từ các đường, các cạnh đó tìm ra các thành phần của vật thể trong ảnh. Cuối cùng, các lớp cao nhất trong mạng huấn luyện sẽ nhận nhiệm vụ phát hiện ra vật thể trong ảnh.

Với cách thức học thông tin từ ảnh lần lượt qua rất nhiều lớp, nhiều tầng khác nhau như vậy, các phương pháp này có thể giúp cho máy tính hiểu được những dữ liệu phức tạp bằng nhiều lớp thông tin đơn giản qua từng bước phân tích. Đó cũng là lý do chúng được gọi là các phương pháp Học sâu.

Tuy có nhiều điểm ưu việt trong khả năng huấn luyện máy tính cho các bài toán phức tạp, Học sâu vẫn còn rất nhiều giới hạn khiến nó chưa thể được áp dụng vào giải quyết mọi vấn đề. Điểm hạn chế lớn nhất của phương pháp này là yêu cầu về kích thước dữ liệu huấn luyện, mô hình huấn luyện. Học sâu đòi hỏi phải có một lượng khổng lồ dữ liệu đầu vào để có thể thực hiện việc học qua nhiều lớp với một số lượng lớn nơ-ron và tham số. Đồng thời, việc tính toán trên quy mô dữ liệu và tham số lớn như vậy cũng yêu cầu đến sức mạnh xử lý của các máy tính server cỡ lớn. Quy trình chọn lọc dữ liệu cũng như huấn luyện mô hình đều tốn nhiều thời gian và công sức, dẫn đến việc thử nghiệm các tham số mới cho mô hình là công việc xa xỉ, khó thực hiện. Tuy nhiên, nhờ các phương pháp Học tập chuyển giao, hiện nay điểm hạn chế lớn nhất này đã không còn là vấn đề quá nghiêm trọng như trước.

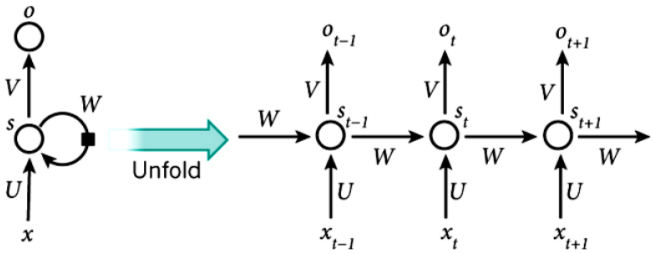
Ngoài hạn chế về kích thước dữ liệu đầu vào, Học sâu còn chưa đủ thông minh để nhận biết và hiểu được các logic phức tạp như con người, các tác vụ do chúng thực hiện vẫn tương đối máy móc và cần cải thiện để "thông minh" hơn nữa.

2.2. Một số thuật toán deep learning

2.1.1. Recurrent Neural Network (RNN - Mạng nơron tái phát)

Recurrent Neural Network (RNN) là một trong những mô hình Deep learning được đánh giá có nhiều ưu điểm trong các tác vụ xử lý ngôn ngữ tự nhiên (NLP).

RNN là một mô hình có trí nhớ (memory), có khả năng nhớ được thông tin đã tính toán trước đó. Không như các mô hình nơron truyền thống, đó là thông tin đầu vào (input) hoàn toàn độc lập với thông tin đầu ra (output). Về lý thuyết, RNN có thể nhớ được thông tin của chuỗi có chiều dài bất kì, nhưng trong thực tế mô hình này chỉ nhớ được thông tin ở vài bước trước đó.



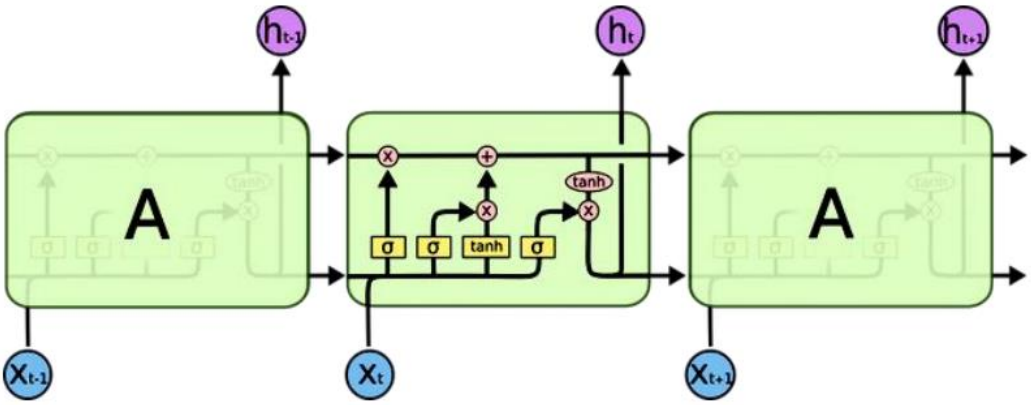
Hình 2. Mạng nơron tái phát

RNN tạo ra các mạng vòng lặp bên trong chúng, cho phép thông tin được lưu trữ lại cho các lần phân tích tiếp theo.

Huấn luyện RNN tương tự như huấn luyện mạng nơ ron truyền thống. Ta cũng sử dụng đến thuật toán backpropagation (lan truyền ngược) nhưng cần một chút tinh chỉnh. Gradient tại mỗi output không chỉ phụ thuộc vào kết quả tính toán của bước hiện tại mà còn phụ thuộc vào kết quả tính toán của các bước trước đó.

2.2.2. Long short-term memory (LSTM)

Long Short Term Memory networks - thường được gọi là "LSTM", là trường hợp đặc biệt của RNN, có khả năng học long-term dependencies. Mô hình này được giới thiệu bởi Hochreiter & Schmidhuber (1997), và được cải tiến lại. Sau đó, mô hình này dần trở nên phổ biến nhờ vào các công trình nghiên cứu gần đây. Mô hình này có khả năng tương thích với nhiều bài toán nên được sử dụng rộng rãi ở các ngành liên quan. LSTM được thiết kế nhằm loại bỏ vấn đề long-term dependency.



Hình 3. Mạng nơron LSTM

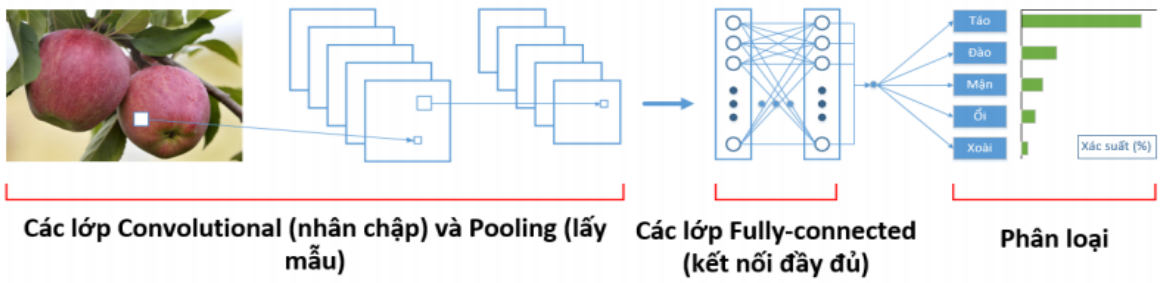
Mô hình này có cấu trúc tương tự như RNN nhưng có cách tính toán khác đối với các hidden state Memory trong LSTM được gọi là hạt nhân (cells). Ta có thể xem đây là một hộp đen nhận thông tin đầu vào gồm hidden state st-1 và giá trị xt. Bên trong các hạt nhân này, chúng sẽ quyết định thông tin nào cần lưu lại và thông tin nào cần chuyển đi, nhờ vậy mà mô hình này có thể lưu trữ được thông tin dài hạn.

2.2.3. Mạng nơ-ron tích chập (Convolution neural network - CNN)

Mạng nơ-ron tích chập (CNN - Convolutional Neural Network) là một trong những mô hình mạng Học sâu phổ biến nhất hiện nay, có khả năng nhận dạng và phân loại hình ảnh với độ chính xác rất cao, thậm chí còn tốt hơn con người trong nhiều trường hợp. Mô hình này đã và đang được phát triển, ứng dụng vào các hệ thống xử lý ảnh lớn của Facebook, Google hay Amazon… cho các mục đích khác nhau như các thuật toán tagging tự động, tìm kiếm ảnh hoặc gợi ý sản phẩm cho người tiêu dùng.

Sự ra đời của mạng CNN là dựa trên ý tưởng cải tiến cách thức các mạng nơ-ron nhân tạo truyền thống học thông tin trong ảnh. Do sử dụng các liên kết đầy đủ giữa các điểm ảnh vào node, các mạng nơ-ron nhân tạo truyền thẳng (Feedforward Neural Network) bị hạn chế rất nhiều bởi kích thước của ảnh, ảnh càng lớn thì số lượng liên kết càng tăng nhanh và kéo theo sự bùng nổ khối lượng tính toán. Ngoài ra sự liên kết đầy đủ này cũng là sự dư thừa khi với mỗi bức ảnh, các thông tin chủ yếu thể hiện qua sự phụ thuộc giữa các điểm ảnh với những điểm xung quanh nó mà không quan tâm nhiều đến các điểm ảnh ở cách xa nhau. Mạng CNN ra đời với kiến trúc thay đổi, có khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến node trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng nơ-ron truyền thẳng.

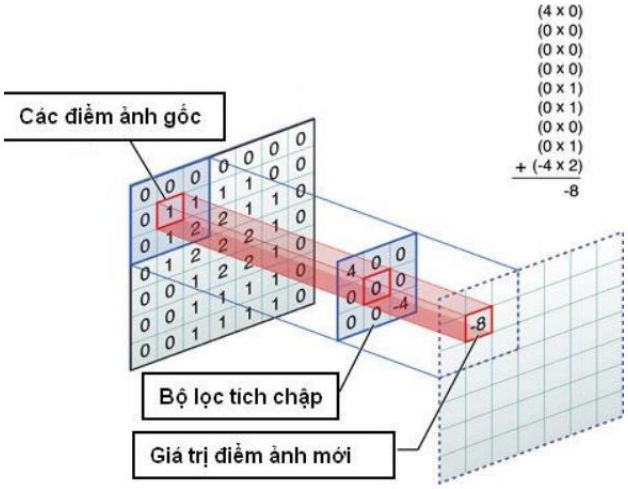
Các lớp cơ bản trong một mạng CNN bao gồm: Lớp tích chập (Convolutional), Lớp kích hoạt phi tuyến ReLU (Rectified Linear Unit), Lớp lấy mẫu (Pooling) và Lớp kết nối đầy đủ (Fully-connected), được thay đổi về số lượng và cách sắp xếp để tạo ra các mô hình huấn luyện phù hợp cho từng bài toán khác nhau.



Hình 4. Kiến trúc cơ bản của một mạng tích chập

- Lớp tích chập:

Đây là thành phần quan trọng nhất trong mạng CNN, cũng là nơi thể hiện tư tưởng xây dựng sự liên kết cục bộ thay vì kết nối toàn bộ các điểm ảnh. Các liên kết cục bộ này được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc - filters - có kích thước nhỏ.



Hình 5. Ví dụ bộ lọc tích chập được sử dụng trên ma trận điểm ảnh

Trong ví dụ ở Hình 5, ta thấy bộ lọc được sử dụng là một ma trận có kích thước 3x3. Bộ lọc này được dịch chuyển lần lượt qua từng vùng ảnh đến khi hoàn thành quét toàn bộ bức ảnh, tạo ra một bức ảnh mới có kích thước nhỏ hơn hoặc bằng với kích thước ảnh đầu vào.

Như vậy, sau khi đưa một bức ảnh đầu vào cho lớp Tích chập ta nhận được kết quả đầu ra là một loạt ảnh tương ứng với các bộ lọc đã được sử dụng để thực hiện phép tích chập.

- Lớp kích hoạt phi tuyến ReLU:

Lớp này được xây dựng với ý nghĩa đảm bảo tính phi tuyến của mô hình huấn luyện sau khi đã thực hiện một loạt các phép tính toán tuyến tính qua các lớp Tích chập.

Lớp Kích hoạt phi tuyến nói chung sử dụng các hàm kích hoạt phi tuyến như ReLU hoặc sigmoid, tanh… để giới hạn phạm vi biên độ cho phép của giá trị đầu ra. Trong số các hàm kích hoạt này, hàm ReLU được chọn do cài đặt đơn giản, tốc độ xử lý nhanh mà vẫn đảm bảo được tính toán hiệu quả. Cụ thể, phép tính toán của hàm ReLU chỉ đơn giản là chuyển tất cả các giá trị âm thành giá trị 0.

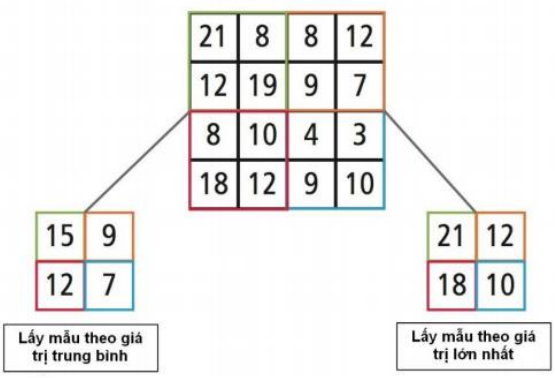
$$(x) = \max(0, x)$$

Thông thường, lớp ReLU được áp dụng ngay phía sau lớp Tích chập, với đầu ra là một ảnh mới có kích thước giống với ảnh đầu vào, các giá trị điểm ảnh cũng hoàn toàn tương tự trừ các giá trị âm đã bị loại bỏ.

- Lớp lấy mẫu:

Một thành phần tính toán chính khác trong mạng CNN là lấy mẫu (Pooling), thường được đặt sau lớp Tích chập và lớp ReLU để làm giảm kích thước kích thước ảnh đầu ra trong khi vẫn giữ được các thông tin quan trọng của ảnh đầu vào. Việc giảm kích thước dữ liệu có tác dụng làm giảm được số lượng tham số cũng như tăng hiệu quả tính toán. Lớp lấy mẫu cũng sử dụng một cửa sổ trượt để quét toàn bộ các vùng trong ảnh tương tự như lớp Tích chập, và thực hiện phép lấy mẫu thay vì phép tích chập - tức là ta sẽ chọn lưu lại một giá trị duy nhất đại diện cho toàn bộ thông tin của vùng ảnh đó.

Hình 6 thể hiện các phương thức lấy mẫu thường được sử dụng nhất hiện nay, đó là Max Pooling (lấy giá trị điểm ảnh lớn nhất) và Avarage Pooling (lấy giá trị trung bình của các điểm ảnh trong vùng ảnh cục bộ)



Hình 6. Phương thức Avarage Pooling và Max Pooling

Như vậy, với mỗi ảnh đầu vào được đưa qua lấy mẫu ta thu được một ảnh đầu ra tương ứng, có kích thước giảm xuống đáng kể nhưng vẫn giữ được các đặc trưng cần thiết cho quá trình tính toán sau này.

- Lớp kết nối đầy đủ:

Lớp kết nối đầy đủ này được thiết kế hoàn toàn tương tự như trong mạng nơ-ron truyền thống, tức là tất cả các điểm ảnh được kết nối đầy đủ với node trong lớp tiếp theo. So với mạng nơ-ron truyền thống, các ảnh đầu vào của lớp này đã có kích thước được giảm bớt rất nhiều, đồng thời vẫn đảm bảo các thông tin quan trọng cho việc nhận dạng. Do vậy, việc tính toán nhận dạng sử dụng mô hình truyền thẳng đã không còn phức tạp và tốn nhiều thời gian như trong mạng nơ ron truyền thống.

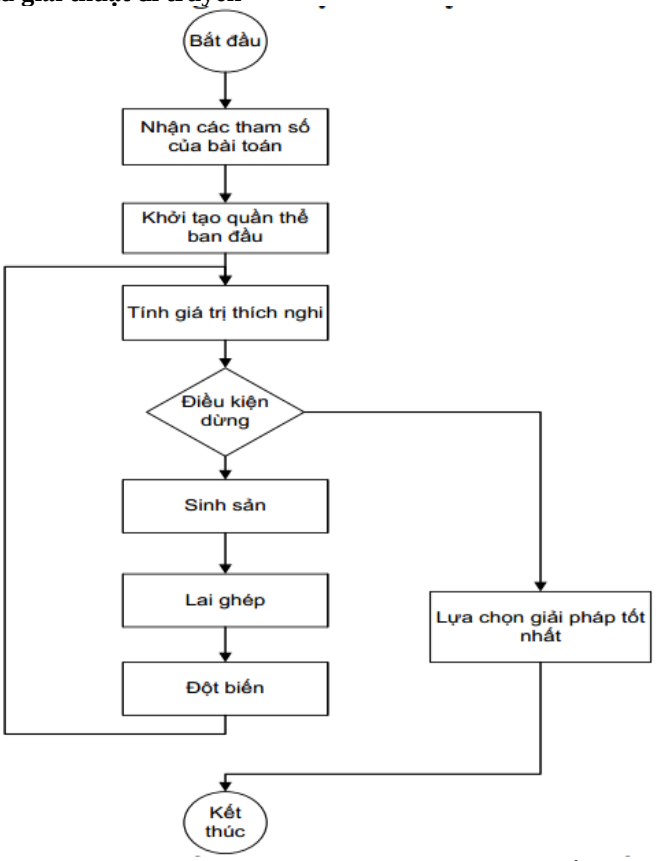
3. Giải thuật di truyền (GAs)

GAs là một kỹ thuật của khoa học máy tính nhằm tìm kiếm giải pháp thích hợp cho các bài toán tối ưu tổ hợp (combinatorial optimization), là một phân ngành của giải thuật tiến hóa, vận dụng các nguyên lý của tiến hóa như: di truyền, đột biến, chọn lọc tự nhiên, và trao đổi chéo. Nó sử dụng ngôn ngữ máy tính để mô phỏng quá trình tiến hoá của một tập hợp những đại diện trừu tượng (gọi là những nhiễm sắc thể), của các giải pháp có thể (gọi là những cá thể) cho bài toán tối ưu hóa vấn đề. Tập hợp này sẽ tiến triển theo hướng chọn lọc những giải pháp tốt hơn. GAs cũng như các thuật toán tiến hoá, đều được hình thành dựa trên một quan niệm được coi là một tiên đề phù hợp với thực tế khách quan. Đó là quan niệm "Quá trình tiến hoá tự nhiên là quá trình hoàn hảo nhất, hợp lý nhất và tự nó đã mang tính tối ưu". Quá trình tiến hoá thể hiện tính tối ưu ở chỗ thế hệ sau bao giờ cũng tốt hơn thế hệ trước (S.N.Sivanandam).

Ngày nay, GAs càng trở nên quan trọng, đặc biệt là trong lĩnh vực tối ưu hoá, một lĩnh vực có nhiều bài toán thú vị, được ứng dụng nhiều trong thực tiễn nhưng thường khó và chưa có phương pháp hiệu quả để giải quyết.

GAs là kỹ thuật chung, giúp giải quyết vấn đề bằng cách mô phỏng sự tiến hóa của con người hay của sinh vật nói chung (dựa trên thuyết tiến hóa muôn loài của Darwin), trong điều kiện qui định sẵn của môi trường. Mục tiêu của GAs không nhằm đưa ra lời giải chính xác tối ưu mà là đưa ra lời giải tương đối tối ưu. Một cá thể trong GAs sẽ biểu diễn một giải pháp của bài toán. Tuy nhiên, không giống với trong tự nhiên là một cá thể có nhiều nhiễm sắc thể (NST) mà để giới hạn trong GAs, ta quan niệm một cá thể có một NST. Do đó, khái niệm cá thể và NST trong GAs coi như là tương đương. Một NST được tạo thành từ nhiều gen, mỗi gen có thể có các giá trị khác nhau để quy định một tình trạng nào đó. Trong GAs, một gen được coi như một phần tử trong chuỗi NST. Một tập hợp các cá thể có cùng một số đặc điểm nào đấy được gọi là quần thể. Trong thuật giải di truyền, ta quan niệm quần thể là một tập các lời giải của một bài toán.

Các bước cơ bản của giải thuật di truyền



Hình 7. Các bước cơ bản của giải thuật di truyền

4. Áp dụng giải thuật di truyền để lựa chọn các tham số cho mạng nơ ron tích chập

Trước khi đào tạo một mạng nơron sâu, cần phải xác định kiến trúc của mạng đó được thực hiện lần lượt bằng cách chọn các siêu tham số được liên kết với mỗi lớp của mạng lưới. Thông thường, các siêu tham số được xác định bởi trực giác của con người, kinh nghiệm hoặc thử nghiệm.

Chúng tôi thử nghiệm mô hình CNN trên tập dữ liệu MNIST, tập dữ liệu này thường dùng để đánh giá hiệu quả của giải thuật nhận dạng ký tự số viết tay. Tập dữ liệu MNIST có nguồn gốc từ tập NIST do tổ chức National Institute of Standards and Technology (NIST) cung cấp, sau đó được LeCun cập nhật và chia thành 2 tập riêng biệt:

- Tập học (huấn luyện) gồm có 60.000 ảnh kích thước 28 x 28, của chữ số viết tay được dung việc huấn luyện mô hình máy học tự động. Tất cả các ảnh trong tập học đều được canh chỉnh và biến đổi thành dữ liệu dạng điểm gồm 60.000 phần tử (ký tự số) có 784 chiều là giá trị mức xám của các điểm, 10 lớp (từ 0 đến 9).
- Tập kiểm tra gồm có 10.000 ảnh của kí số viết tay được dùng cho việc kiểm thử, tương tự các ảnh trong tập thử cũng được biến đổi và canh chỉnh thành dữ liệu điểm gồm 10000 phần tử trong 784 chiều, 10 lớp (từ 0 đến 9).

Cấu trúc mạng CNN gồm:

1. Lớp nhân chập có 32 cửa sổ trượt với kích thước 5x5 và hàm kích hoạt X
2. Lớp lấy mẫu theo giá trị lớn nhất trong cửa sổ 2 x 2
3. Lớp nhân chập có 32 cửa sổ trượt với kích thước 5x5 và hàm kích hoạt X
4. Lớp lấy mẫu lấy tối đa trong cửa sổ 2 x 2
5. Lớp làm phẳng
6. Lớp kết nối đầy đủ có Y nơron và hàm kích hoạt X
7. Lớp đầu ra

Các tham số được đưa vào lựa chọn là các hàm kích hoạt X, phương pháp tối ưu và số nơron Y của lớp kết nối đầy đủ. Trong đó:

- Các hàm kích hoạt là: tanh, relu và sigmoid.
- Các phương pháp tối ưu là: adam, rmsprop, sgd.
- Số nơron của lớp kết nối đầy đủ là từ 32 đến 256.

Như vậy mỗi cá thể sẽ gồm 3 thành phần đó là hàm kích hoạt, phương pháp tối ưu và số nơron của lớp kết nối đầy đủ.

Ứng dụng giải thuật di truyền trong bài toán lựa chọn tham số cho CNN như sau:

[Khởi tạo] Sinh ngẫu nhiên một quần thể gồm n cá thể (ở đây chúng tôi chọn n = 20)

[Quần thể mới] Tạo quần thể mới bằng cách lặp lại các bước sau cho đến khi quần thể mới hoàn thành

[Thích nghi] Thực hiện CNN với bộ tham số lưu trong mỗi cá thể. Độ thích nghi chính là kết quả độ chính xác khi phân loại được đánh giá bởi phương pháp cross validation.

[Kiểm tra] Kiểm tra điều kiện kết thúc giải thuật. Ở đây điều kiện kết thúc là sau 1 số lần lặp n nào đó hoặc sự chênh lệnh về độ chính xác của 2 cá thể tốt nhất trong 2 thế hệ nhỏ hơn β ($0.001 < \beta < 0.1$)

[Chọn lọc] Chọn hai cá thể bố mẹ từ quần thể cũ theo độ thích nghi của chúng (cá thể có độ thích nghi càng cao thì càng có nhiều khả năng được chọn)

[Lai ghép] Với một xác suất lai ghép được chọn, lai ghép hai cá thể bố mẹ để tạo ra một cá thể mới.

[Đột biến] Với một xác suất đột biến được chọn, biến đổi cá thể mới

[Chọn kết quả] Nếu điều kiện dừng được thỏa mãn thì thuật toán kết thúc và trả về lời giải tốt nhất trong quần thể hiện tại

Sau đây là một số kết quả minh họa:

hàm kích hoạt	phương pháp tối ưu	Số nơron	Độ chính xác
Relu	Adam	78	0.9589
Sigmoid	Sgd	50	0.5889
Sigmoid	Sgd	107	0.6208
Sigmoid	Adam	169	0.9115

Nếu sử dụng phương pháp tìm kiếm lưới để lựa chọn bộ tham số tốt nhất ta cần phải thực hiện 225 x 3 x 3 lần thực hiện mạng CNN. Trong khi đó với giải thuật di truyền chỉ vài trăm lần thực hiện CNN ta đã nhận được cấu trúc mạng khá tốt.

5. Kết luận

Trong bài báo này chúng tôi vừa trình bày việc ứng dụng giải thuật di truyền để lựa chọn các tham số cho mạng nơ ron tích chập CNN để nhận dạng chính xác ký tự số viết tay. Kết quả thử nghiệm trên tập dữ liệu thực MNIST cho thấy rằng việc ứng dụng giải thuật di truyền để lựa chọn các tham số phương pháp là khá hiệu quả, nó giúp giảm đáng kể thời gian tính toán so với các phương pháp xác định tham số truyền

thống như tìm kiếm lưới. Trong thời gian tới, chúng tôi sẽ áp dụng phương pháp này để xác định cho các giải thuật khác của phương pháp học sâu.

Lời cảm ơn

Bài báo này là kết quả của Đề tài mã số CT.2019.01.02.

Tài liệu tham khảo

Adit Deshpande (July 20, 2016). A Beginner's Guide To Understanding Convolutional Neural Networks. https://adeshpande3.github.io/A-Beginner'sGuide-To-Understanding-Convolutional-Neural-Networks/

Andrej Karpathy. CS231n Convolutional Neural Networks for Visual Recognition - Image Classification. http://cs231n.github.io/classification/

Michael A. Nielsen (2013), Neural Networks And Deep Learning, Determination Press.

Ian Goodfellow, Y oshua Bengio, Aaron Courville, Deep Learning, An MIT Press book. http://www.deeplearningbook.org/

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541-551, Winter 1989. yann.lecun.com/exdb/mnist/

Y. LeCun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard. Handwritten digit recognition: Applications of neural net chips and automatic learning. *IEEE Communication*, pages 41-46, November 1989. invited paper.

S.N.Sivanandam · S.N.Deepa, Introduction to Genetic Algorithms, Springer-Verlag Berlin Heidelberg, 2008

ABSTRACT

Using genetic algorithms to select parameters for deep learning

Bui Thi Van Anh[1], Dang Huu Nghi[1]
[1]*Hanoi University of Mining and Geology*

Machine learning is a field of artificial intelligence involved in the research and construction of techniques that allow systems to "learn" automatically from data to solve specific problems. Deep learning is a branch of machine learning that models highly abstracted data using multiple processing layers with complex structures and nonlinear transformations. To achieve good results, when designing a deep neural network we have to optimize its structure by selecting appropriate parameters. In this paper, we propose to use genetic algorithms to select parameters for deep learning models.

Keywords: Machine learning; deep learning; artificial intelligence; data; structure optimization; genetic algorithm.

Đánh giá độ chính xác của một số phương pháp xử lý dữ liệu thiếu cho dữ liệu chuỗi thời gian (time series data)

Nguyễn Thị Phương Bắc[1*], Đặng Văn Nam
[1]*Trường Đại học Mỏ - Địa chất*

TÓM TẮT

Xử lý dữ liệu thiếu (missing values) là một trong những công đoạn không thể thiếu trong quá trình làm sạch dữ liệu (Data cleansing). Có hai nhóm phương pháp xử lý dữ liệu thiếu đó là loại bỏ các giá trị thiếu và/hoặc thay thế dữ liệu thiếu bằng một giá trị mới. Việc lựa chọn sử dụng nhóm phương pháp nào hoặc đồng thời cả hai phụ thuộc vào từng bài toán, từng loại và kiểu dữ liệu cụ thể…Dữ liệu chuỗi thời gian (time series data) là một chuỗi các điểm dữ liệu được đo theo từng khoảng thời gian liền nhau theo một tần suất thời gian thống nhất. Việc xử lý giá trị thiếu trong dữ liệu chuỗi thời gian cũng có những khác biệt lớn so với việc xử lý giá trị thiếu của các dạng dữ liệu khác. Trong nội dung của bài báo này, chúng tôi sẽ giới thiệu 4 phương pháp chính được sử dụng để xử lý giá trị thiếu đối với dữ liệu chuỗi thời gian bao gồm: LOCF, NOCB, nội suy tuyến tính, nội suy Spline; Cũng trong bài báo, chúng tôi sẽ trình bày việc áp dụng các phương pháp này và đánh giá độ chính xác của mỗi phương pháp sử dụng độ đo MAE, RMSE cho dữ liệu nhiệt độ trong năm 2019 tại 5 trạm khu vực phía Bắc.

Từ khóa: Dữ liệu thiếu; chuỗi thời gian; mae; rmse.

1. Mở đầu

Chuẩn bị dữ liệu (Data preparation) là công đoạn bắt buộc, là khâu chiếm nhiều thời gian, công sức và nguồn lực nhất của bất kỳ một dự án khoa học dữ liệu nào. Các kết quả nghiên cứu cho thấy 80% thời gian, công sức và nguồn lực của một dự án khoa học dữ liệu là cho việc này. Chuẩn bị dữ liệu bao gồm rất nhiều thao tác, nghiệp vụ, kỹ thuật và yêu cầu khác nhau, phụ thuộc vào từng loại dữ liệu và từng dự án cụ thể. Tuy nhiên, chúng ta có thể tổng hợp vào ba nhóm thao tác chính: Làm sạch dữ liệu (Data cleansing); Chuyển đổi dữ liệu (Data transformation) và Tích hợp dữ liệu (Combining data). (Davy Cielen and et al., 2016)

Như vậy, làm sạch dữ liệu là bước đầu tiên cần phải thực hiện sau khi thu thập dữ liệu, các dữ liệu ban đầu khi thu thập được từ các nguồn khác nhau được gọi là dữ liệu thô (raw data). Dữ liệu này thường chứa rất nhiều "nhiễu", các dữ liệu không liên quan, chưa theo một chuẩn chung thống nhất, hoặc dữ liệu không đầy đủ, bị mất mát, thiếu thông tin…đây là những vấn đề luôn hiện hữu trong mọi bộ dữ liệu.

Một trong những thao tác quan trọng trong quá trình làm sạch dữ liệu đó là xử lý các giá trị thiếu, mất mát (missing values). Có nhiều phương pháp khác nhau để xử lý giá trị thiếu, mỗi một phương pháp lại phù hợp với một bài toán, một lĩnh vực và một loại dữ liệu cụ thể. Không có một phương pháp xử lý nào là tốt cho tất cả mọi trường hợp.

Những dữ liệu quan sát liên tục cho một hiện tượng (vật lý, kinh tế ...) trong một khoảng thời gian sẽ tạo nên một chuỗi thời gian như: Doanh số của công ty trong 20 năm gần đây, hoặc nhiệt độ ghi nhận tại một trạm quan trắc khí tượng, hoặc công suất điện năng tiêu thụ trong một nhà máy, đó là các ví dụ điển hình cho một chuỗi thời gian. Dữ liệu chuỗi thời gian (time series data) là một trong những loại dữ liệu đang rất phổ biến và quan trọng ngày nay. Được ứng dụng vào trong rất nhiều lĩnh vực khác nhau, đặc biệt cho các bài toán dự đoán, dự báo. Cũng giống như các loại dữ liệu khác, dữ liệu chuỗi thời gian hoàn toàn có thể bị thiếu, mất mát. Tuy nhiên, việc xử lý dữ liệu thiếu trong chuỗi thời gian lại có những phương pháp và cách thức riêng.

Trong nội dung của bài báo này, nhóm tác giả sẽ nghiên cứu về dữ liệu chuỗi thời gian, các phương pháp cơ bản để xử lý dữ liệu thiếu trong chuỗi thời gian bao gồm: Thay thế giá trị thiếu bằng giá trị liền trước (LOCF); Thay thế giá trị thiếu bằng giá trị liền sau (NOCB); Nội suy tuyến tính (Linear), Nội suy Spline. Nhóm tác giả cũng tiến hành thực nghiệm các phương pháp này trên bộ dữ liệu nhiệt độ tại 5 trạm quan trắc: 48805 - Hà Giang, 48806 - Sơn La, 48825 - Hà Đông, 48838 - Móng Cái và 48839 - Bạch Long Vĩ

[*] *Tác giả liên hệ:*
Email: nguyenthiphuongbac@humg.edu.vn

trong năm 2019, sau đó sử dụng độ đo MAE và RMSE để đánh giá độ chính xác của 4 phương pháp xử lý dữ liệu thiếu nêu trên.

2. Một số phương pháp xử lý giá trị thiếu cho dữ liệu chuỗi thời gian

2.1 .Dữ liệu chuỗi thời gian (time series data)

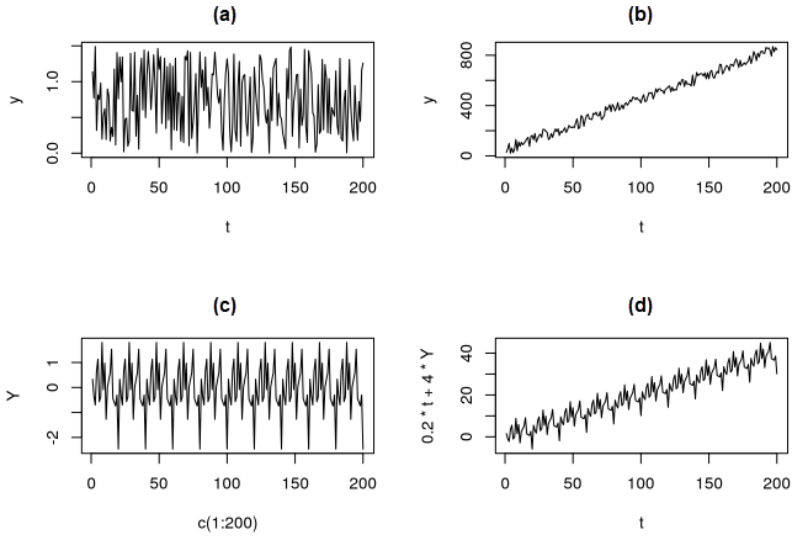
Chuỗi thời gian (time series) trong thống kê, xử lý tín hiệu, kinh tế lượng và toán tài chính là một chuỗi các điểm dữ liệu, được đo theo từng khoảng thời gian liền nhau và theo một tần suất thời gian thống nhất. Dữ liệu chuỗi thời gian thường bao gồm các phép đo liên tiếp thực hiện từ cùng một nguồn trong một khoảng thời gian. Phân tích chuỗi thời gian có mục đích nhận dạng và tập hợp lại các yếu tố, những biến đổi theo thời gian mà nó ảnh hưởng tới giá trị của biến quan sát.

Nếu phân tích dữ liệu chuỗi thời gian theo mô hình hồi quy tuyến tính, giá trị tại thời điểm t là y(t) được xác định như sau:

$$y(t) = m(t) + s(t) + \varepsilon(t) \quad (1)$$

Trong đó: m(t) là xu hướng (trend), s(t) là tính thời vụ (seasonality) và ε(t) là biến ngẫu nhiên. Dựa trên phương trình 1, có thể chia dữ liệu chuỗi thời gian thành 4 loại:

- Dữ liệu chuỗi thời gian không có xu hướng và mùa vụ (còn gọi là nhiễu trắng - white nosie) - Hình 1(a)
- Dữ liệu chuỗi thời gian có tính xu hướng nhưng không có tính mùa vụ - Hình 1(b)
- Dữ liệu chuỗi thời gian không có tính xu hướng nhưng có tính mùa vụ - Hình 1(c)
- Dữ liệu chuỗi thời gian có cả tính xu hướng và tính mùa vụ - Hình 1(d)



Hình 1. Các dạng dữ liệu chuỗi thời gian

Dữ liệu chuỗi thời gian được ứng dụng rất rộng rãi trong nhiều lĩnh vực như: Dự báo kinh tế; Phân tích thời gian thực; Tính toán doanh số bán hàng; Phân tích thị trường; Dự báo thời tiết…. (Shumway and et al., 2017). Quá trình thu thập dữ liệu chuỗi thời gian có thể do nhiều nguyên nhân chủ quan, khách quan một số thời điểm trong chuỗi bị thiếu, mất mát. Do đó việc xử lý giá trị thiếu là vấn đề bắt buộc trước khi có thể sử dụng được dữ liệu này cho bất kỳ mục đích nào (Xi Wang, Chen Wang, 2019).

2.2 . Một số phương pháp xử lý giá trị thiếu

Xử lý giá trị thiếu (missing values) luôn là một bước quan trọng và bắt buộc trong quá trình làm sạch dữ liệu. Có rất nhiều phương pháp để xử lý dữ liệu thiếu, có thể gom các phương pháp này vào 2 nhóm chính đó là: Loại bỏ các dòng hoặc các cột dữ liệu chứa giá trị thiếu ra khỏi tập dữ liệu; Thay thế các điểm dữ liệu thiếu bằng một giá trị mới theo từng thuật toán cụ thể (Choi J and et al., 2018). Trong thực tế, dữ liệu thiếu được chia thành 3 dạng, bao gồm:

- Dữ liệu thiếu hoàn toàn ngẫu nhiên (Missing Completely at Random - MCAR): Sự mất mát dữ liệu tại các điểm quan sát là hoàn toàn ngẫu nhiên, và không có bất kỳ một mối quan hệ hay sự liên hệ nào giữa dữ liệu thiếu với các dữ liệu quan sát khác.

- Dữ liệu thiếu ngẫu nhiên (Missing at Random - MAR): Sự mất mát dữ liệu ở đây là ngẫu nhiên, tuy nhiên vẫn có mối quan hệ hệ thống giữa dữ liệu bị mất và dữ liệu được quan sát.
- Dữ liệu thiếu không ngẫu nhiên (Missing Not at Random - MNAR): Sự mất mát dữ liệu không phải là ngẫu nhiên mà có một mối quan hệ xu hướng giữa giá trị bị thiếu và giá trị không bị thiếu trong một biến.

Với dữ liệu chuỗi thời gian ta không thể xóa bỏ các dòng dữ liệu thiếu mà chỉ có thể sử dụng nhóm phương pháp thứ 2 là thay thế bằng một giá trị mới. Với các dữ liệu dạng chuỗi thời gian, các điểm dữ liệu sẽ có mối quan hệ với các điểm phía trước và phía sau nó, cũng như tuân theo xu hướng và mùa vụ (Carl Bonander, Ulf Stromberg, 2018). Có 4 giải pháp đơn giản nhưng hiệu quả để xử lý dữ liệu thiếu cho chuỗi thời gian bao gồm:

a) Thay thế giá trị thiếu bằng giá trị liền trước (LOCF-Last observation carried forward):

Phương pháp này đơn giản là thay thế các điểm dữ liệu thiếu bằng các dữ liệu của điểm liền trước nó. Hình 1 minh họa việc xử lý giá trị thiếu bằng phương pháp LOCF. Hình 1(a) thể hiện một đoạn dữ liệu chuỗi thời gian theo ngày, trong đó có 3 ngày dữ liệu bị thiếu là các ngày 05/01, 07/01 và 08/01 (ký hiệu N/A). Hình 1(b) thể hiện kết quả khi sử dụng phương pháp LOCF để xử lý giá trị thiếu. Dữ liệu thiếu ngày 05/01 sẽ được thay thế bằng dữ liệu của ngày liền trước đó là ngày 04/01.

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	155	87%
7	7-Jan	N/A	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%

(a)

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	90	86%
6	6-Jan	155	87%
7	7-Jan	155	89%
8	8-Jan	155	90%
9	9-Jan	180	92%

(b)

Hình 2(a). Dữ liệu gốc ban đầu chứa các dữ liệu thiếu (N/A)
(b). Kết quả thay thế giá trị thiếu bằng phương pháp LOCF

b) Thay thế giá trị thiếu bằng giá trị liền sau (NOCB-Next observation carried backward):

Phương pháp này thực hiện tương tự như với phương pháp LOCF chỉ khác biệt là sử dụng các giá liền sau để thay thế cho giá trị thiếu. Hình 2 minh họa việc sử dụng phương pháp NOCB để xử lý giá trị thiếu. Dữ liệu thiếu ngày 05/01 sẽ được thay thế bằng dữ liệu của ngày liền sau đó là ngày 06/01.

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	155	87%
7	7-Jan	N/A	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%

(a)

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	155	86%
6	6-Jan	155	87%
7	7-Jan	180	89%
8	8-Jan	180	90%
9	9-Jan	180	92%

(b)

Hình 3(a). Dữ liệu gốc ban đầu chứa các dữ liệu thiếu (N/A)
(b). Kết quả thay thế giá trị thiếu bằng phương pháp NOCB

c) Thay thế giá trị thiếu bằng phương pháp nội suy tuyến tính (Linear interpolation):

Nội suy là phương pháp ước tính giá trị của các điểm dữ liệu chưa biết trong phạm vi của một tập hợp rời rạc chứa một số điểm dữ liệu đã biết. Nội suy tuyến tính là nội suy đơn giản nhất, được thực hiện bằng cách lấy giá trị trung bình giữa giá trị của điểm dữ liệu trước và sau điểm dữ liệu bị thiếu. Hình 3 minh họa phương pháp xử lý giá trị thiếu bằng phương pháp nội suy tuyến tính. Dữ liệu thiếu ngày 05/01 được xác định dựa trên dữ liệu ngày trước đó 04/01 và ngày sau đó 06/01. Bản chất của nội suy tuyến tính là xây

dựng một đường thẳng đi qua 2 điểm trước và sau của điểm dữ liệu bị thiếu để sau đó xác định giá trị của điểm dữ liệu thiếu này.

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%

(a)

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	120	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	170	90%
9	9-Jan	180	92%

(90+150)/2 = 120

(160+180)/2 = 170

(b)

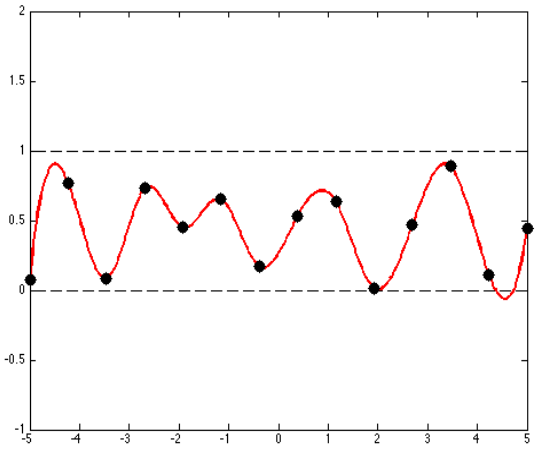
Hình 4(a). Dữ liệu gốc ban đầu chứa các dữ liệu thiếu (N/A)
(b) Kết quả thay thế giá trị thiếu bằng phương pháp nội suy tuyến tính

d) Thay thế giá trị thiếu bằng phương pháp nội suy spline (Spline interpolation):

Nội suy Spline là phương pháp xây dựng các đường cong trơn đi qua n + 1 điểm dữ liệu đã biết (x_0, y_0),..., (x_n, y_n). Thực thế là đi tìm một hàm f(x) sao sao $f(x_i) = y_i$ với mọi i. Chúng ta sẽ xác định n đa thức bậc p_0,…., p_{n-1} sao cho $f(x) = p_i(x)$ với mọi x trong khoảng $[x_i, x_{i+1}]$ (Erdogan KAYA, 2014). Trong thực tế nhóm tác giả sử dụng nội suy spline với đa thức bậc 3 khi đó $p_i(x)$ được định nghĩa như sau:

$$p_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad \text{(Ajao and et al., 2012)}$$

Hình 4 minh họa việc xây dựng các đường cong bậc 3 (đường màu đỏ) đi qua 14 điểm đã biết (điểm chấm đen). Dựa vào các đường cong đi qua các điểm đã biết này để xác định giá trị của các điểm dữ liệu bị thiếu.



Hình 5. Nội suy Spline bậc 3 qua 14 điểm đã biết

Các phương pháp này áp dụng cho dữ liệu chuỗi thời gian với giả thiết là dữ liệu liên tục, các điểm dữ liệu có mối quan hệ tương quan với nhau. Dữ liệu nhiệt độ tại các thời điểm trong ngày là dữ liệu dạng này. Dữ liệu tại một thời điểm sẽ có mối quan hệ với dữ liệu tại điểm trước và sau đó. Hình 5 minh họa phương pháp xử lý giá trị thiếu bằng phương pháp nội suy spline.

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	N/A	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	N/A	90%
9	9-Jan	180	92%

Mobile ID	Date	Download Speed	Data Limit Usage
1	1-Jan	157	80%
2	2-Jan	99	81%
3	3-Jan	167	83%
4	4-Jan	90	84%
5	5-Jan	94	86%
6	6-Jan	150	87%
7	7-Jan	160	89%
8	8-Jan	151	90%
9	9-Jan	180	92%

Hình 6(a). Dữ liệu gốc ban đầu chứa các dữ liệu thiếu (N/A)
(b). Kết quả thay thế giá trị thiếu bằng phương pháp nội suy Spkine

Vì vậy trong phần thực nghiệm khi đánh giá độ chính xác của các phương pháp thay thế giá trị thiếu cho dữ liệu nhiệt độ tại 5 trạm khu vực phía bắc, nhóm tác giả sẽ thực hiện cả 4 phương pháp đã trình bày ở trên.

3. Chỉ số xác định sai số

Để đánh giá độ chính xác của dữ liệu thay thế với giá trị quan sát, trong thống kê người ta thường sử dụng 4 chỉ số bao gồm: Sai số trung bình (ME - Mean Error); Sai số tuyệt đối trung bình (MAE - Mean Absolute Error); Sai số bình phương trung bình (MSE - Mean Square Error) và sai số bình phương trung bình quân phương (RMSE - Root Mean Square Error). Các chỉ số đánh giá này được xác định như sau:

a) Sai số trung bình (ME):

$$ME = \frac{1}{N}\sum_{i=1}^{N}(F_i - O_i)$$

Trong đó:

- F_i là giá trị thay thế (trong phần thực nghiệm F_i là giá trị thay thế có được bằng một trong các phương như LOCF, NOCB, nội suy Linear, nội suy Spline).
- O_i là giá trị thật của biến (trong phần thực nghiệm O_i là giá trị nhiệt độ quan trắc).
- N là tổng số các điểm xác định sai số

Các thông số F_i, O_i và N tương tự cho MAE, MSE và RMSE

Sai số trung bình có giá trị nằm trong khoảng (-∞, +∞). ME cho biết xu hướng lệch trung bình của giá trị thay thế so với giá trị quan trắc, nhưng không phản ánh độ lớn của sai số. ME dương cho biết giá trị thay thế vượt quá giá trị quan trắc và ngược lại. Nếu ME = 0 được xem là "hoàn hảo" khi đó dữ liệu thay thế trùng với dữ liệu quan trắc.

b) Sai số tuyệt đối trung bình (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|F_i - O_i|$$

Sai số tuyệt đối trung bình nằm trong khoảng (0, +∞). MAE biểu thị biên độ trung bình của sai số mô hình nhưng không nói lên xu hướng lệch của giá trị thay thế và giá trị quan trắc. Khi MAE = 0, các giá trị thay thế hoàn toàn trùng khớp với các giá trị quan trắc, khi đó phương pháp xử lý giá trị thiếu được xem là "lý tưởng". Thông thường, MAE được sử dụng cùng với ME để đánh giá độ tin cậy.

c) Sai số bình phương trung bình (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(F_i - O_i)^2$$

Sai số bình phương trung bình nằm trong khoảng (0, +∞), MSE phản ánh mức độ dao động giữa giá trị thay thế với giá trị quan trắc.

d) Sai số bình phương trung bình quân phương (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(F_i - O_i)^2}$$

Sai số bình phương trung bình là một trong những đại lượng cơ bản và thường được sử dụng phổ biến cho việc đánh giá kết quả giữa giá trị thay thế và giá trị quan trắc. Người ta thường hay sử dụng RMSE biểu thị độ lớn trung bình của sai số. Đặc biệt RMSE rất nhạy với những giá trị sai số lớn. Giống như MAE, RMSE không chỉ ra độ lệch giữa giá trị dự báo và giá trị quan trắc. Giá trị của RMSE nằm trong khoảng (0,+ ∞)

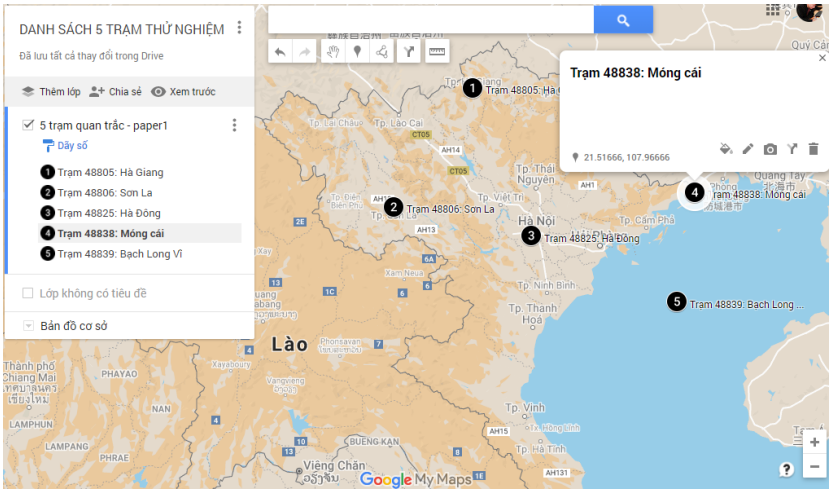
Trong phần thực nghiệm, nhóm tác giả sử dụng 2 chỉ số là MAE và RMSE để xác định sai số giữa giá trị thay thế và giá trị quan trắc thực tế, qua đó xác định phương pháp thay thế nào cho độ chính xác cao hơn (nghĩa là có sai số nhỏ hơn).

4. Thực nghiệm trên dữ liệu nhiệt độ tại 5 trạm quan trắc

4.1. Mô tả dữ liệu

Để đánh giá độ chính xác của các phương pháp xử lý dữ liệu thiếu đã trình bày trong phần 2, nhóm tác giả sử dụng tập dữ liệu nhiệt độ thu thập được từ 5 trạm quan trắc bao gồm: Trạm 48805 - Hà Giang; Trạm 48806 - Sơn La; Trạm 48825 - Hà Đông; Trạm 48838 - Móng Cái và Trạm 48839 - Bạch Long Vĩ. Vị trí của các trạm trên Google Map như Hình 5. Cả 5 trạm này đều là các trạm quan trắc 3h, nghĩa là sẽ thực hiện thu thập dữ liệu khí tượng 8 lần mỗi ngày, mỗi lần cách nhau 3 giờ tại các thời điểm 00h, 03h, 06h,

09h, 12h, 15h, 18h, 21h theo giờ GMT, tương ứng với 01h, 04h, 07h, 10h, 13h, 16h, 19h, 22h giờ Việt Nam. Đây là các dữ liệu thực tế được cung cấp bởi Trung tâm thông tin và dữ liệu khí tượng thủy văn; Trong phần thực nghiệm này, nhóm tác giả sử dụng dữ liệu nhiệt độ quan trắc trong năm 2019 từ thời điểm 01h ngày 01 tháng 01 đến 22h ngày 31 tháng 12. Dữ liệu được lưu trữ trong file .CSV bao gồm các thông tin về thời điểm quan trắc và giá trị nhiệt độ (ºC) của 5 trạm này. Hình 6 mô tả chi tiết file dữ liệu, trong đó dòng đầu tiên là tiêu đề của file dữ liệu, bao gồm 6 cột. Cột đầu tiên cho biết thời điểm quan trắc, 5 cột còn lại tương ứng với giá trị nhiệt độ tại 5 trạm lấy theo mã trạm.



Hình 7. Vị trí 5 trạm quan trắc trên Google Maps

	A	B	C	D	E	F
1	TimeVN	48805	48806	48825	48838	48839
2	2019-01-01 1:00	12.4	9.4	11.8	9.3	12.2
3	2019-01-01 4:00	12.2	9.3	11.4	9.3	12.1
4	2019-01-01 7:00	12.2	9.1	11	9.4	12.2
5	2019-01-01 10:00	13.2	11.1	11.6	11.2	12
6	2019-01-01 13:00	14.8	13.1	12.2	13	12.6
7	2019-01-01 16:00	14.6	13.7	13	12	12.8
8	2019-01-01 19:00	13.2	12	12.3	10.4	13
9	2019-01-01 22:00	12.2	11.2	12	10.4	12.8
10	2019-01-02 1:00	12.1	10.6	12	10.6	13
11	2019-01-02 4:00	12.2	10.4	12	10.4	13
12	2019-01-02 7:00	11.8	10.4	11.9	10.4	12.8
13	2019-01-02 10:00	14	12	13.8	12.6	13.1
14	2019-01-02 13:00	16.2	13.4	16.1	16.2	14.1
15	2019-01-02 16:00	16.1	13.1	16.2	15	14.2
16	2019-01-02 19:00	14.6	12.1	15.3	13.4	14.3
17	2019-01-02 22:00	14	11.9	15.2	13	14.1
18	2019-01-03 1:00	13.6	11.6	14.6	12.8	13.2
19	2019-01-03 4:00	13.4	11.2	13.6	11.8	12.9

Data_Original

Hình 8. File dữ liệu của 5 trạm thực nghiệm

Bảng 1 dưới đây tổng hợp các thông số và một số đặc trưng thống kê liên quan đến dữ liệu của 5 trạm này, qua đó thấy được cái nhìn tổng quan về tập dữ liệu. Dữ liệu của 5 trạm này trong năm 2019 là dữ liệu đầy đủ, không bị thiếu giá trị nào. Đây là file dữ liệu chuẩn sẽ được sử dụng để đánh giá độ chính xác của các phương pháp xử lý giá trị thiếu.

Bảng 1. Thông số thống kê cho tập dữ liệu ban đầu (Data_Original)

Thông số	Trạm quan trắc				
	48805	48806	48825	48838	48839
Tổng số điểm dữ liệu quan trắc	2920	2920	2920	2920	2920
Số điểm có dữ liệu	2920	2920	2920	2920	2920
Số điểm dữ liệu thiếu	0	0	0	0	0
Nhiệt độ thấp nhất (ºC)	6.9	2.9	10.3	7.4	12

Nhiệt độ cao nhất (ºC)	38.5	36.9	39.7	35.5	34.4
Nhiệt độ trung bình (ºC)	24.405	22.939	25.482	24.133	25.341
Độ lệch chuẩn	5.237	5.329	5.435	5.424	4.502

Để đánh giá độ chính xác của các phương pháp xử lý dữ liệu thiếu, nhóm tác giả sẽ thực hiện loại bỏ đi một số giá trị trong tập dữ liệu gốc với cả 3 dạng mất mát dữ liệu khác nhau bao gồm MAR, MCAR và MNAR cho dữ liệu nhiệt độ của 5 trạm này. Tổng số điểm dữ liệu bị loại bỏ tại mỗi trạm là 35 điểm trong tổng số 2920 điểm, tương ứng với 1.2% tổng số điểm dữ liệu. Mỗi một trạm sẽ có một kiểu mất mát dữ liệu khác nhau cụ thể như sau:

- Trạm 48805: Các điểm mất mát dữ liệu là hoàn toàn ngẫu nhiên; Dữ liệu gốc bị loại bỏ từng điểm quan trắc riêng rẽ hoàn toàn ngẫu nhiên.
- Trạm 48806: Các điểm mất mát dữ liệu là hoàn toàn ngẫu nhiên; Dữ liệu gốc bị loại bỏ một số điểm liền nhau có thể là 2 điểm, 3 điểm hoặc một ngày liên tục hoàn toàn ngẫu nhiên.
- Trạm 48825: Dữ liệu mất mát không ngẫu nhiên; Dữ liệu gốc bị loại bỏ tại thời điểm 07h (thời điểm nhiệt độ trung bình thấp nhất trong ngày) của 35 ngày liên tiếp.
- Trạm 48838: Dữ liệu mất mát không ngẫu nhiên; Dữ liệu gốc bị loại bỏ tại thời điểm 13h (thời điểm nhiệt độ trung bình cao nhất trong ngày) của 35 ngày liên tiếp.
- Trạm 48839: Dữ liệu mất mát không ngẫu nhiên; Dữ liệu gốc bị loại bỏ tại các thời điểm 10h và 19h các ngày 01 và 15 hàng tháng.

Bảng 2 mô tả thông số của tập dữ liệu sau khi đã loại bỏ đi một số giá trị để trở thành tập dữ liệu thiếu theo các nguyên tắc ở trên.

Bảng 2. Thông số thông kế cho tập dữ liệu chứa giá trị thiếu (Data_Missing)

Thông số	Trạm quan trắc				
	48805	48806	48825	48838	48839
Tổng số điểm dữ liệu quan trắc	2920	2920	2920	2920	2920
Số điểm có dữ liệu	2885	2885	2885	2885	2885
Số điểm dữ liệu thiếu	35	35	35	35	35
Nhiệt độ thấp nhất (ºC)	6.9	2.9	10.3	7.4	12
Nhiệt độ cao nhất (ºC)	38.5	36.9	39.7	35.5	34.4
Nhiệt độ trung bình (ºC)	24.397	22.899	25.450	24.040	25.329
Độ lệch chuẩn	5.245	5.337	5.458	5.385	4.508

Như vậy, chúng ta đã có 2 tập dữ liệu; Tập dữ liệu Data_Original là tập dữ liệu gốc, chứa dữ liệu nhiệt độ quan trắc tại 5 trạm trong năm 2019 với 2920 thời điểm, mỗi thời điểm cách nhau 3h. Đây là tập dữ liệu đầy đủ, không bị thiếu giá trị nào; Tập dữ liệu Data_Missing là tập dữ liệu lấy từ tập Data_Original nhưng đã bị loại bỏ đi 35 điểm quan trắc cho mỗi trạm (dữ liệu thiếu chiếm ~1.2% tổng số điểm dữ liệu), Dữ liệu trong mỗi trạm sẽ bị loại bỏ đi theo những dạng mất mát khác nhau bao gồm MCAR và MNAR như đã trình bày ở trên.

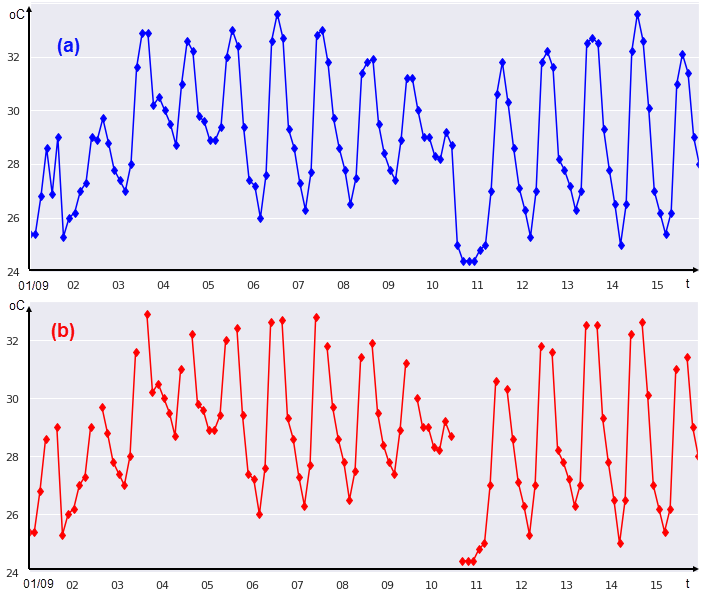
Hình 9 minh họa dữ liệu trạm 48838 trong khoảng thời gian từ 01/09 đến 15/09/2019. Hình 9(a) thể hiện dữ liệu trong tập Data_Origninal, dữ liệu gốc thu tập được không bị mất mát. Hình 9(b) thể hiện dữ liệu trong tập Data_Missing của trạm 48838, Dữ liệu này đã bị loại bỏ tại thời điểm lúc 13h trong 35 ngày liên tiếp.

4.2. Áp dụng các phương pháp xử lý giá trị thiếu

Để xử lý giá trị thiếu theo các phương pháp đã trình bày trong phần 2, nhóm tác giả sử dụng ngôn ngữ lập trình Python, thư viện hỗ trợ phân tích và xử lý số liệu Pandas. Pandas là một trong những thư viện rất mạnh mẽ trong việc xử lý dữ liệu, đặc biệt là dữ liệu chuỗi thời gian.

Nhóm tác giả sẽ tiến hành lần lượt việc thay thế các điểm dữ liệu thiếu theo 4 phương pháp LOCF, NOCF, nội suy tuyến tính (linear) và nội suy Spline bậc 3. Bảng 3 thể hiện kết quả thay thế các giá trị thiếu bằng 4 phương pháp cho trạm 48806 thời điểm ngày 10-08-2019. Cột đầu tiên thể hiện thời điểm quan trắc dữ liệu, với tần suất 3h một lần, trong một ngày sẽ thu thập dữ liệu tại 8 thời điểm. Cột 48806_mis là dữ liệu lấy từ tập Data_Missing, như vậy trong ngày 10/08/2019 có 5 điểm dữ liệu quan trắc liền nhau (4h, 7h, 10h, 13h và 16h) bị mất mát (trong Pandas dữ liệu thiếu ký hiệu NaN). Cột 48806 ori là dữ liệu lấy từ tập Data_Original, đây là các giá trị quan trắc chính xác tại các thời điểm tương ứng, các giá trị đúng này sẽ làm cơ sở để đánh giá sai số cho các phương pháp xử lý dữ liệu thiếu được trình bày trong phần 4.3 dưới đây. Cột LOCF - kết quả thay thế các giá trị thiếu bằng phương pháp thay thế giá trị tại thời điểm liền trước.

Cột NOCB - kết quả thay thế giá trị thiếu bằng phương pháp thay thế giá trị tại thời điểm liền sau. Cột Linear - kết quả thay thế giá trị thiếu bằng phương pháp nội suy tuyến tính. Cột Spline - kết quả thay thế giá trị thiếu bằng phương pháp nội suy spline bậc 3.

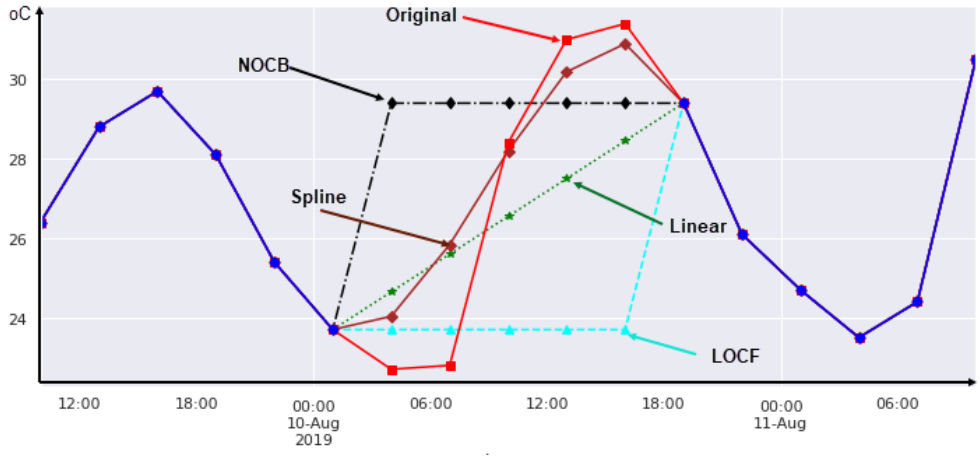


Hình 9(a). Dữ liệu đầy đủ của trạm 48838 từ 01/09 đến 15/09
(b). Dữ liệu đã bị loại bỏ các thời điểm 13h trong khoảng thời gian tương ứng

Bảng 3. Kết quả xử lý giá trị thiếu theo các phương pháp khác nhau

Thời điểm	48806_mis	48806_ori	LOCF	NOCB	Linear	Spline
2019-08-10 01:00:00	23.7	23.7	23.7	23.7	23.7	23.7
2019-08-10 04:00:00	NaN	22.7	23.7	29.4	24.7	24.0
2019-08-10 07:00:00	NaN	22.8	23.7	29.4	25.6	25.8
2019-08-10 10:00:00	NaN	28.4	23.7	29.4	26.6	28.2
2019-08-10 13:00:00	NaN	31.0	23.7	29.4	27.5	30.2
2019-08-10 16:00:00	NaN	31.4	23.7	29.4	28.5	30.9
2019-08-10 19:00:00	29.4	29.4	29.4	29.4	29.4	29.4
2019-08-10 22:00:00	26.1	26.1	26.1	26.1	26.1	26.1

Hình 9 là đồ thị thể hiện trực quan giữa dữ liệu quan trắc thực (các điểm màu đỏ) và dữ liệu tại các điểm thay thế sử dụng các phương pháp xử lý giá trị thiếu khác nhau.



Hình 10. Trực quan hóa dữ liệu thực và các giá trị thay thế dữ liệu thiếu theo 4 phương pháp

4.3.Đánh giá độ chính xác của các phương pháp

Để đánh giá phương pháp xử lý giá trị thiếu nào trong 4 phương pháp LOCF, NOCB, Linear và Spline cho kết quả gần với giá trị quan trắc nhất (Sai số giữa giá trị thay thế và giá trị quan trắc nhỏ nhất) nhóm tác giả sử dụng 4 độ đo bao gồm: ME, MAE, MSE, RMSE. Hình 10 là mã nguồn các hàm tính toán sai số theo các công thức đã được trình bày trong phần 3 của bài báo.

```python
#Xây dựng các hàm tính sai số MA, MAE, MSE, RMSE
import numpy as np
def me(actual,predicted):
    """ Mean Error """
    error = np.mean(actual -  predicted)
    return round(error,4)

def mae(actual, predicted):
    """ Mean Absolute Error """
    error = np.mean(abs(actual -  predicted))
    return round(error,4)

def mse(actual, predicted):
    """ Mean Squared Error """
    error = np.mean(np.square(actual -  predicted))
    return round(error,4)

def rmse(actual, predicted):
    """ Root Mean Squared Error """
    error = np.sqrt(np.mean(np.square(actual -  predicted)))
    return round(error,4)
```

Hình 10. Xây dựng các hàm tính sai số ME, MAE, MSE, RMSE

Các Bảng 4,5 thể hiện kết quả sai số MAE và RMSE giữa giá trị thay thế với giá trị quan trắc tại 4 trạm ứng với 4 phương pháp xử lý giá trị thiếu.

Bảng 4. Tổng hợp sai số MAE của 4 phương pháp xử lý giá trị thiếu

Trạm	MAE_LOCF	MAE_NOCB	MAE_Linear	MAE_Spline
48805	0.0201	0.0188	0.0105	**0.0078**
48806	0.0344	0.0471	0.0301	**0.0287**
48825	**0.0076**	0.0316	0.0135	0.0086
48838	0.0175	0.0156	0.0157	**0.0123**
48839	0.0221	**0.0100**	0.0106	0.0106

Bảng 5. Tổng hợp sai số RMSE của 4 phương pháp xử lý giá trị thiếu

Trạm	RMSE_LOCF	RMSE_NOCB	RMSE_Linear	RMSE_Spline
48805	0.2712	0.2138	0.1275	**0.1034**
48806	0.3902	0.5156	**0.3416**	0.4086
48825	**0.0823**	0.3167	0.1359	0.0977
48838	0.1967	0.2124	0.1790	**0.1436**
48839	0.2523	**0.1068**	0.1229	0.1242

Qua kết quả tổng hợp sai số giữa giá trị thay thế và giá trị quan trắc có thể nhận thấy không có một phương pháp xử lý giá trị thiếu nào là tốt cho mọi loại mất mát.

- Nếu mất mát là hoàn toàn ngẫu nhiên nhưng chỉ bị mất ứng với từng thời điểm riêng lẻ (như của trạm 48805), hoặc mất mát không ngẫu nhiên, thời điểm mất là đơn lẻ, tại thời điểm nhiệt độ cao nhất trong ngày (như của trạm 48838) thì phương pháp sử dụng nội suy Spline bậc 3 cho độ chính xác cao hơn (sai số nhỏ nhất).
- Nếu mất mát không ngẫu nhiên, thời điểm mất mát là đơn lẻ, mất mát tại thời điểm nhiệt độ thấp nhất trong ngày 07h (như của trạm 48825) thì phương pháp LOCF cho độ chính xác cao hơn.

Tuy nhiên về tổng thể, với 5 dạng mất mát khác nhau ứng với dữ liệu thiếu tại 5 trạm, phương pháp nội suy Spline bậc 3 cho sai số thấp hơn, nghĩa là độ chính xác của dữ liệu được thay thế gần với giá trị quan trắc hơn.

5. Kết luận

Chuẩn bị dữ liệu là giai đoạn bắt buộc trong bất kỳ một dự án khoa học dữ liệu nào, dữ liệu được chuẩn bị tốt sẽ giúp cho việc xây dựng các mô hình dự đoán, dự báo được chính xác. Chuẩn bị dữ liệu bao gồm rất nhiều công đoạn và yêu cầu khác nhau, trong đó xử lý giá trị thiếu (giá trị mất mát - missing values) là yêu cầu bắt buộc; Có nhiều phương pháp khác nhau để xử lý giá trị thiếu, việc lựa chọn phương pháp nào hay kết hợp nhiều phương pháp phụ thuộc bào từng bài toán, từng loại dữ liệu cụ thể. Với dữ liệu chuỗi thời gian (time series) chúng ta không thể sử dụng phương pháp loại bỏ các điểm giá trị thiếu, mà chỉ có thể thay thế các điểm dữ liệu thiếu bằng một giá trị khác phù hợp. Có rất nhiều phương pháp thay thế giá trị thiếu, tuy nhiên không có một phương pháp pháp thay thế nào là tối ưu cho mọi loại dữ liệu, mọi bài toán. Trong nội dung của bài báo này, nhóm tác giả đã giới thiệu 4 phương pháp thay thế giá trị thiếu thường được áp dụng cho dữ liệu chuỗi thời gian mà ở đó giá trị tại mỗi điểm có mối quan hệ tương quan với các điểm phía trước và phía sau nó. Nhóm tác giả cũng thực nghiệm 4 phương pháp thay thế giá trị thiếu LOCF, NOCB, nội suy Linear và nội suy Spline trên tập dữ liệu nhiệt độ tại 5 trạm quan trắc 48805, 48806, 48825, 48838 và 48839 với các dạng mất mát MCAR và MNAR. Các kết quả của bài báo làm cơ sở cho nhóm tác giả lựa chọn được phương pháp xử lý giá trị thiếu phù hợp với loại dữ liệu nhiệt độ, và áp dụng phương pháp xử lý đó cho dữ liệu tại các trạm quan trắc khác.

Tài liệu tham khảo

Davy Cielen, Arno D. B., Meysman, Mohamed Ali, (2016). Introducing Data Science, Manning Publications Co.

Shumway, R.H., Stoffer, D.S. (2017), Time Series Analysis and Its Applications: With R Examples. Cham, Switzerland: Springer, 562 p.

Xi Wang, Chen Wang, Time Series Data Cleaning: A Survey, IEEE Access (12/2019), page 1866-1881

Choi J, Dekkers OM, le Cessie S. A comparison of different methods to handle missing data in the context of propensity score analysis. Eur J Epidemiol, 2018.

Carl Bonander, Ulf Stromberg. Methods to handle missing values and missing individuals. European Journal of Epidemiology, 2018.

Erdogan KAYA. Spline Interpolation Techniques. Journal of Technical Science and Technologies, ISSN 2298-0032, 2014.

Ajao, I.O., Ibraheem, A.G., Ayoola, F.J. Cubic spline interpolation: A robust method of disaggregating annual data to quarterly series. Journal of Physical Sciens and Environmental Safety, Volume 2, Number 1,2012.

ABSTRACT

Evaluate the accuracy of some missing value processing methods for time series data

Nguyen Thi Phuong Bac[1], Dang Van Nam[1]

[1]Hanoi University of Mining and Geology

Dealing with missing data (missing values) is one of the important steps in the data cleansing process. There are two ways to deal with missing data: delete missing values and/or replace missing data with new values. The choice of which set of methods or both to use depends on each question, specific data type and data type... Time series data is a series of data points, with a uniform time frequency for each continuous time. The treatment of missing values in time series data is also very different from the treatment of missing values of other data types. In this article, we will introduce 4 main methods for dealing with missing values of time series data, including: LOCF, NOCB, linear interpolation, spline interpolation... The results of using apply these methods to different missing data types MAE and RMSE measurements to evaluate the accuracy of each method in the 2019 temperature observation data of 5 stations in northern Vietnam.

Keywords: Missing values; time series data; MAE; RMSE.

Integrated flow shop and vehicle routing problem

Ta Quang Chieu[1,*], Ta Xuan Giang[2], Ha Thi Thu Hien[3], Tran Thi Hiep[3], Pham Đuc Hau[1]
[1]*Hanoi University of Mining and Geology*
[2]*Viet Hung Industrial University*
[3]*Viet Tri Industrial University*

ABSTRACT

We consider in this paper a *m*-machine permutation flow shop scheduling problem and vehicle routing problem (VRP) integrated. The manufacturing workshop is a flow shop and there is only one vehicle available. We are interested in the minimization of the total tardiness, related to the delivery completion times. Therefore, we are faced to three interdependent problems: scheduling the jobs in the flow shop environment, batching the completed jobs, and routing the batches. Then, we present the resolution method based on Tabu search and the results that have been obtained. Computational experiments are performed on random data sets and show the efficiency of the methods. Finally, a conclusion and some future research directions are proposed.

Keywords: Flowshop problem; vehicle routing problem; metaheuristic algorithm; tabu search.

1. Introduction and notations

We consider that the jobs have to be delivered to the customers after their production by using a single vehicle. The processing time of each job on each machine and the due date of delivery for each job are known, and a matrix of travel times is given. The jobs have to be scheduled in a *m*-machine flow shop environment, then batches have to be defined (one batch corresponds to one trip of the vehicle) and a route has to be determined for each batch, so that the total tardiness of delivery is minimized.

The vehicle routing problem consists in defining a route starting from the production site, visiting the customers associated to the jobs in the batch, and finishing at the production site. Each customer requiring goods is visited by the vehicle (see *Figure 1*).

The aim of this paper is to propose an algorithm for scheduling the jobs on the *m*-machines flow shop, for constituting batches of jobs and for determining the vehicle routing for each batch, so that the total tardiness of delivery is minimized. This problem is clearly an *NP*-hard problem (Lenstra et al., 1981).

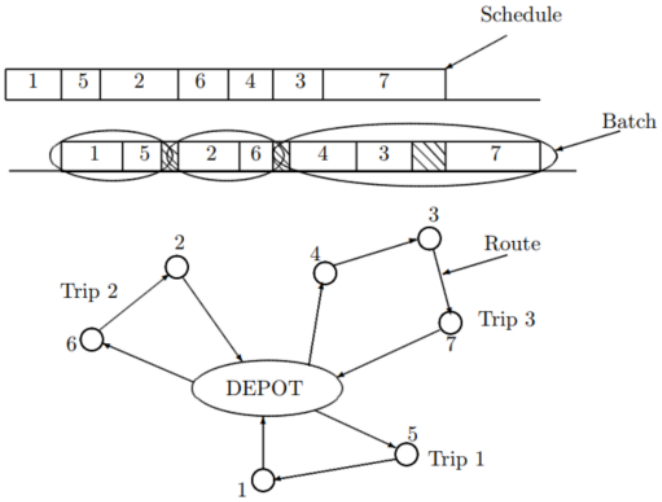


Figure 1. Illustration of the integrated production scheduling and vehicle routing problems

* *Tác giả liên hệ*
Email: taquangchieu@humg.edu.vn

Notations

The specific notations that are used in this chapter are the following:

- To each job J_j is associated a customer location number j, the location of the production facility has an index 0,
- The delivery time between each pair of locations i and j is denoted by $l_{i,j}$,
- The number of vehicles is equal to 1.

2. Tabu search algorithm

2.1. Coding of a solution

We use an array of $3n$ elements to represent a complete solution. The first n elements represent the sequence of jobs (i.e. the schedule), the next $2n$ elements give for each trip the number of jobs in the trip and the list of jobs (the routing of these jobs is implicitely the order of the jobs in this list). In the following, the first part of the coding of a solution S (first n elements) is denoted by S^0 and the second part ($2n$ elements) is denoted by S^T. The second part is decomposed into several trips. Each trip λ of S^T is denoted by S^T_λ and is composed by the number of visits k_λ and the list of visits ($S^T_{\lambda[1]}$,…, $S^T_{\lambda[k_\lambda]}$). The coding is illustrated in *Figure 2*.

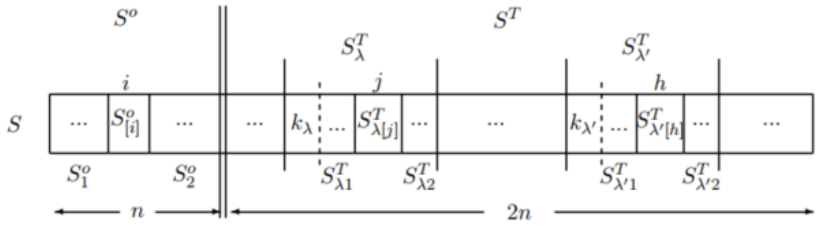


Figure 2. Illustration of the coding of a solution and notations

Example: In the example presented in *Figure 3*, the schedule is $\{J_1, J_5, J_2, J_6, J_4, J_3, J_7\}$, the number jobs in each trip is (2, 2, 3, 0, 0, 0, 0), i.e. two jobs in the first two trips and three jobs in the third trip, the remaining trips are empty.

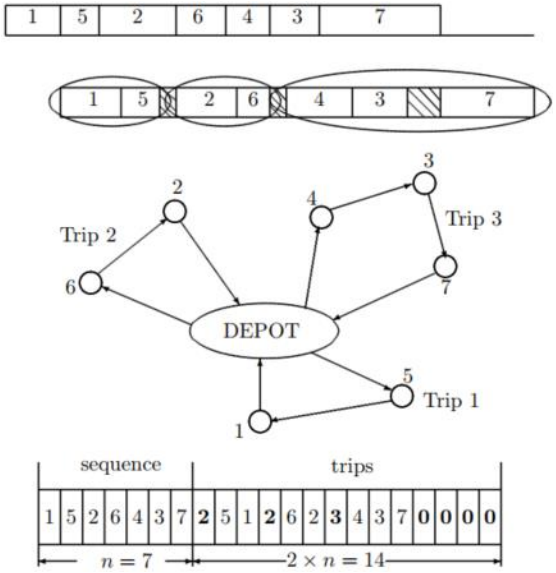


Figure 3. Example of the coding of a solution

2.2. Initial solution

EDD algorithm is used for giving an initial solution for the scheduling problem. For the trips, each trip contains only one job. The customers are visited in the same order as in the sequence.

Example: An example of an initial solution is given in *Figure 4*.

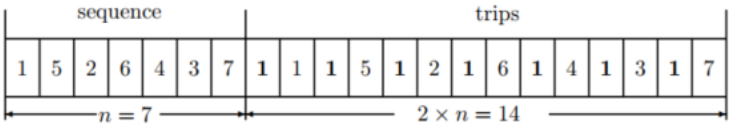


Figure 4. Example of initial solution

2.3. Neighborhood definitions

We assume that $S = S^0//S^T$ is the current complete solution with sequence S^0 and trips S^T, where:

- $S^0 = S_1^0/S_{[i]}^0/S_2^0/S_{[j]}^0/S_3^0$ with $S_{[i]}^0$ and $S_{[j]}^0$ the jobs in positions i and j ($i < j$) in S^0 and S_1^0, S_2^0 and S_3^0 three partial sequences.

$S^T = k_1, S_1^T/.../k_{\lambda-1}, S_{\lambda-1}^T/k_\lambda, S_\lambda^T/k_{\lambda+1}, S_{\lambda+1}^T/k_n, S_n^T$ with k_λ the number of jobs in trip λ, S_λ^T is the sequence of jobs in trip λ, $\forall\lambda \in \{1, 2, \ldots, n\}$.

If $k_\lambda = 0$ then S_λ^T is empty. Notice also that $\sum_{\lambda=1}^{n} k_\lambda$=n.

Neighborhood based on the sequence of jobs

We use $SWAP^0$ operator for creating the neighbors of sequence S^0. This operator is the same as the one described (Ta Quang Chieu et al. 2013).

$SWAP^0$: A neighbor of S is created by interchanging the jobs in position i and j in sequence S^0. The corresponding jobs are also swapped in S^T. See *Figure 5* for an illustration of this operator.

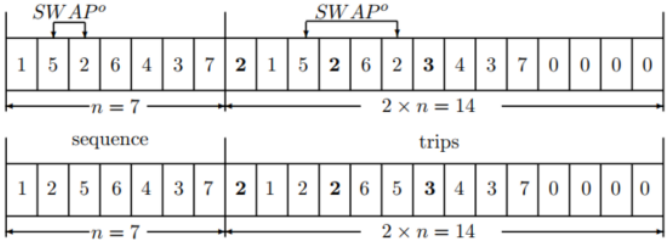


Figure 5. Illustration of $SWAP^0$ operator

Neighborhood based on a single trip

We use $SWAP^T$, $EBSR^T$, $ESFR^T$ and $Inversion^T$ operators for creating the neighbors of a trip S_λ^T. These operators are the same as those described (Ta Quang Chieu et al. 2013).

Given a sequence of visits S_λ^T of trip λ, two random positions i and j in S_λ^T (i < j and$\leq$ k_λ): $S_\lambda^T = S_{\lambda 1}^T$, $S_{\lambda[i]}^T$, $S_{\lambda 2}^T$, $S_{\lambda[j]}^T$, $S_{\lambda 3}^T$.

- $SWAP^T$: A neighbor of S_λ^T is created by interchanging the jobs in position i and j, leading to sequence $S_\lambda^{T\prime} = S_{\lambda 1}^T$, $S_{\lambda[j]}^T$, $S_{\lambda 2}^T$, $S_{\lambda[i]}^T$, $S_{\lambda 3}^T$. See *Figure 6* for example.

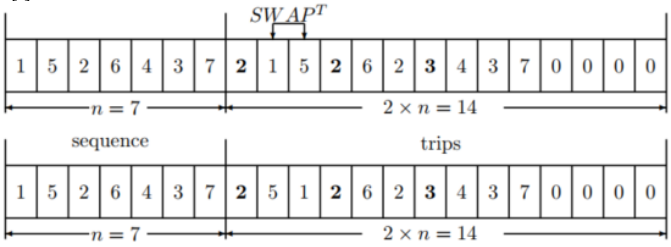


Figure 6. Illustration of $SWAP^T$ operator for a trip

- $EBSR^T$: A neighbor of S_λ^T is created by extracting the visit in position j and reinserting this visit backward just before the visit in position i, leading to sequence $S_\lambda^{T\prime} = S_{\lambda 1}^T$, $S_{\lambda[j]}^T$, $S_{\lambda[i]}^T$, $S_{\lambda 2}^T$, $S_{\lambda 3}^T$. See *Figure 7* for example.

- $ESFR^T$: A neighbor of S_λ^T is created by extracting the visit in position i and reinserting it forward immediately after the visit in position j, leading to a sequence $S_\lambda^{T\prime} = S_{\lambda 1}^T$, $S_{\lambda 2}^T$, $S_{\lambda[j]}^T$, $S_{\lambda[i]}^T$, $S_{\lambda 3}^T$. See *Figure 8* for example.

- *Inversion*^{T}: A neighbor of S_λ^T is created by inserting $S_{\lambda[i]}^T$, $S_{\lambda 2}^T$, $S_{\lambda[j]}^T$ in the inverse order between $S_{\lambda 1}^T$ and $S_{\lambda 3}^T$. See *Figure 9* for example.

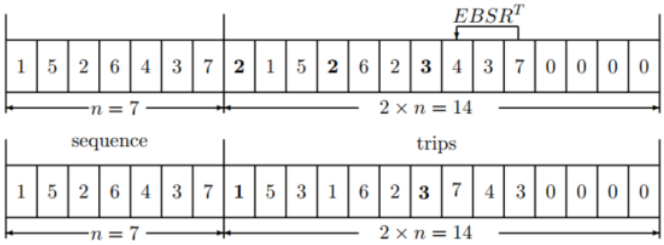


Figure 7. Illustration of EBSR^T operator for a trip

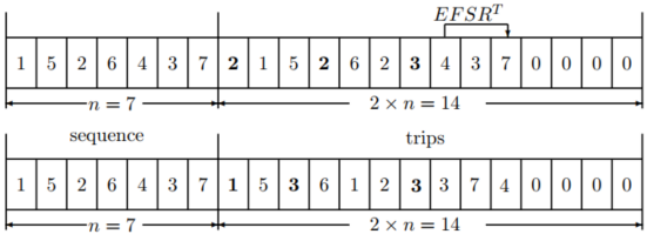


Figure 8. Illustration of EFSR^T operator for a trip

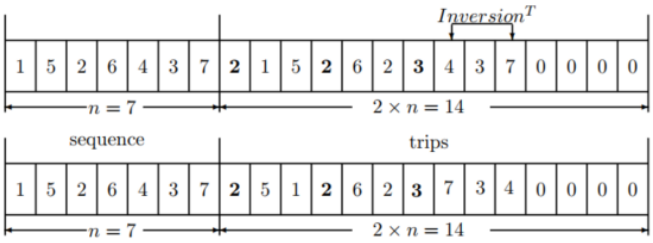


Figure 9. Illustration of Inversion^T operator for a trip

Moves and selection of the best neighbor

The aim of the problem is to minimize the total tardiness. The best neighbor in the candidate item that is non-tabu and with the smallest total tardiness. For managing the tabu list, we use the first-in-first-out (FIFO) strategy. Old attributes are deleted as new attributes are inserted.

Tabu list

We define three tabu lists and the size of the tabu lists is fixed. An element of the tabu list is defined by (λ, μ, J_i, J_j), λ, μ are the indexes of two trips, J_i and J_j are two jobs:

- For a swap of two jobs J_i and J_j in the sequence, we insert in the Tabu list the element (0, 0, J_i, J_j). For example, see *Figure 5*, the element inserted in the Tabu list is (0, 0, J_5, J_2). In this case $\lambda = \mu = 0$.
- For a move in two trips λ and μ, we insert in the Tabu list the element (λ, μ, J_i, J_j) where i and j are the positions that are concerned in trips λ and μ respectively. For a move in a single trip, we have $\lambda = \mu$. For example, see *Figure 6*, where the tabu list is (1, 1, J_1, J_5).

Stopping condition

The algorithm is stopped when the time limit has been reached. This time limit is denoted by *TimeLimitTS = n(m=2)t ms* (Vallada et al., 2008), where $t = 90$.

Detailed algorithm

The detailed TS algorithm is given in *Algorithm 1*

3. Computational experiments

We present in this section the generation of data, and we discuss the results.

Generation data

We have tested the algorithms on a PC Intel coreTMi5 CPU 2.4GHz. Data sets have been randomly generated (notice that there is no benchmark instance for the m-machine flowshop and vehicle routing problem integrated). The processing times $p_{i,j}$ have been generated in [1,100], the due dates dj have been

generated in [50, 50n], the position of "custom" j is given by its coordinates (x_j, y_j) generated in [1, 70]. The delivery time $l_{i,j}$ is the classical euclidian distance $l_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$. Ten instances are used for each combination of n and m, with $n \in \{20, 30, 50, 70, 100, 150, 200\}$ and $m \in \{2, 4\}$. For the TS algorithm, some preliminary experiments have conducted to the following parameters settings: *TimeLimitTS* = 10 seconds, *Tabu list* = {10, 40, 60, 120} elements.

Initialization
S_0 = initial solution, S = current solution
$S' = S_0$ // best solution of $N(S)$
$S^* = S_0$ // best solution of $N(S)$ and *non-tabu*
$f^* = f(S_0)$ // f^* value of S^* and $f(S_0)$ value of S_0
$T = \emptyset$ // T is the tabu list
while (CPU $\leq$ *TimeLimitTS*) **do**
  $f(S') = \infty$,
  //Selecting neighbor in sequence
  for $i = 0$ **to** $n - 1$ **do**
    for $j = i + 1$ **to** n **do**
      Test($SWAP^o$)
    end for
  end for
  //Selecting neighbor of a trip
  for $\lambda = 0$ **to** $n - 1$ **do**
    for $i = 0$ **to** $nbjoboftrip[\lambda] - 2$ **do**
      //$nbjoboftrip[\lambda]$ is the number jobs of trip λ
      for $j = i + 1$ **to** $nbjoboftrip[\lambda] - 1$ **do**
        Test($Opera^T$)
      end for
    end for
  end for
  //Selecting neighbor in two trips
  for $\lambda = 0$ **to** $n - 2$ **do**
    for $\mu = 0$ **to** $n - 1$ **do**
      for $i = 0$ **to** $nbjoboftrip[\lambda] - 1$ **do**
        //$nbjoboftrip[\lambda]$, $nbjoboftrip[\mu]$ are the number jobs of trip λ, μ
        for $j = 0$ **to** $nbjoboftrip[\mu] - 1$ **do**
          Test($Opera_2^T$)
        end for
      end for
    end for
  end for
  if ($f(S') < f^*$) **then** $S^* = S'$, $f^* = f(S)$, **end if**
  if (SizeTabu $\geq$ TabuMax) **then** Del(T) **end if**
  Add($T, (\lambda, \mu, i, j)$)
end while

Algorithm 1. Tabu search algorithm for scheduling and vehicle routing

Results

In Table 1, column 'Best' for TS_X ($X \in \{10, 40, 60, 120\}$) indicates the number of times this method strictly outperforms other method. Column 'Δ_{TSX}' indicates the average deviation between TS_X and the best method between TS_{10}, TS_{40}, TS_{60} TS_{120}.

We can see that the TS_{40} (with tabu list is 40) leads to the best results with a number of best solutions equal to 54. On average, the deviation between the solutions returned by this method and the best solutions is 10,46%. This value is around and 14,98% for TS_{10}, 10,46% for TS_{40}, 8,49% for TS_{60} and 9,94% for TS_{120}.

4. Conclusions and discussions

We approach a problem where a m-machine permutation flow shop scheduling problem and a vehicle routing problem are integrated to minimize the total tardiness. To our knowledge, this is the first time that this problem is approached in the literature. We present a direct coding for a complete solution and a neighborhood method for finding a sequence and trips. We propose a tabu search algorithm for this

problem, the first results show that the TS greatly improves the initial solution given by EDD and where each trip serves only one job at a time.

In the future, the first research directions are about the evaluation of the method. We will develop genetic algorithms and other methods in order to see the real quality of the tabu search. Then, we will combine mathematical programming and local search (matheuristic), in order to see if matheuristic are performing methods for this problem.

n x m	TS_{10}		TS_{40}		TS_{60}		TS_{120}	
	Best	Δ_{TS10}	**Best**	Δ_{TS40}	**Best**	Δ_{TS60}	**Best**	Δ_{TS120}
20 × 2	3	17,58%	7	4,16%	3	17,58%	0	24,81%
30 × 2	1	11,62%	7	13,34%	3	10,97%	0	30,19%
50 × 2	0	17,40%	1	21,81%	2	12,49%	7	3,23%
70 × 2	0	16,12%	0	21,88%	2	10,99%	8	2,50%
100 × 2	1	19,24%	1	25,61%	0	14,93%	8	1,65%
150 × 2	0	23,42%	0	15,27%	4	5,29%	6	5,95%
200 × 2	0	26,35%	4	5,60%	3	5,67%	3	8,19%
20 × 4	0	9,51%	9	0,00%	1	9,51%	0	12,04%
30 × 4	0	9,28%	10	0,00%	0	9,28%	0	17,63%
50 × 4	0	8,67%	5	3,19%	2	3,27%	1	4,60%
70 × 4	1	7,34%	2	9,54%	3	5,38%	5	4,47%
100 × 4	0	17,60%	3	9,84%	2	9,23%	5	4,75%
150 × 4	0	19,08%	2	2,84%	4	3,08%	4	4,10%
200 × 4	0	18,52%	3	2,60%	5	4,16%	2	10,73%
	6	14,98%	54	10,46%	34	8,49%	49	9,94%

References

Lenstra, J. K. and Rinnooy Kan, A. H. G, 1981. Complexity of vehicle routing and scheduling problems. Networks, 11, 221-227.

Ta, Q. C., Billaut, J. C., and Bouquard, J. L, 2015. Minimizing total tardiness in the m-machine flow-shop by genetic and matheuristic algorithms. Journal of Intelligent Manufacturing, 29, 617- 628.

Ta, Q. C., Billaut, J. C., and Bouquard, J. L, Morin, P-E., 2014. Minimisation de la somme des retards pour un problème d'ordonnancement de type flow-shop à deux machines et un problème de livraison intégrés. 15ème congrès de la Société de la Société Française de Recherche Opérationnelle et d'Aide à la Décision Roadef'2014, Bordeaux, France.

Ta, Q. C., Billaut, J. C., and Bouquard, J. L, 2013. An hybrid metaheuristic, an hybrid lower bound and a tabu search for the two-machine flowshop total tardiness problem. Proceedings of the 10th IEEE RIVF International Conference on Computing and Communication Technologies (RIVF'13), Hanoi, Vietnam.

Vallada, E., Ruiz, R., and Minella, G, 2008. Minimising total tardiness in the m- machine flowshop problem: A review and evaluation of heuristics and metaheuristics. Computers & Operations Research, 35, 1350-1373.

TÓM TẮT

Sự kết hợp bài toán flowshop và bài toán định tuyến xe

Tạ Quang Chiểu[1], Tạ Xuân Giang[3], Hà Thị Thu Hiền[3], Trần Thị Hiệp[3], Phạm Đức Hậu[1]

[1]Trường Đại học Mỏ - Địa chất

[2]Trường Đại học Công nghiệp Việt Hung

[3]Trường Đại học Công nghiệp Việt Trì

Trong báo báo này chúng tôi xem xét sự kết hợp bài toán flowshop và bài toán định tuyến xe (Vehicle Routing Problem - VRP). chúng tôi đề xuất một thuật toán Metaheuristic - Tabu search để giải bài toán này và đưa ra các kết quả đạt được. Cuối cùng là kết luận và một vài hướng nghiên cứu trong tương lai được đề xuất.

Từ khóa: Flowshop problem; vehicle routing problem; metaheuristic algorithm; tabu search.

Bảo mật dữ liệu trong Điện toán đám mây

Đỗ Như Hải[1,*]
[1]*Trường Đại học Mỏ - Địa chất*

TÓM TẮT

Điện toán đám mây là một công nghệ phát triển nhanh chóng ngày nay với số lượng người dùng di động tăng lên đáng kể. Quyền riêng tư và bảo mật dữ liệu cho người sử dụng thiết bị di động là một khía cạnh đáng quan tâm. Nghiên cứu hướng đến mục đích đề xuất một giải pháp kết hợp các thuật toán đã có nhằm tăng cường bảo mật cho điện toán đám mây. Bảo mật cho điện toán đám mây liên quan đến nhiều vấn đề như bảo mật mạng, bảo mật web, truy cập dữ liệu, xác thực, ủy quyền, bảo mật dữ liệu… Ở nghiên cứu này sử dụng phương pháp phân tích và tổng hợp, với phạm vi đưa ra giải pháp để tăng cường bảo mật cho điện toán đám mây với các thiết bị di động nhằm đảm bảo tính xác thực và chống rò rỉ dữ liệu . Trong đó có thể sử dụng biến thể RSA để đảm bảo khả năng bảo mật cao.

Từ khóa: Điện toán đám mây; thuật toán mã hóa khối; thuật toán mã hóa hóa công khai; bảo mật điện toán đám mây.

1. Đặt vấn đề

Điện toán đám mây là một mô hình nở rộ và phát triển nhanh chóng, với các tính năng mới được cập nhật thường xuyên. Nó tích hợp các công nghệ khác nhau để cung cấp dịch vụ, nền tảng và cơ sở hạ tầng cho nhiều người dùng và các tổ chức kinh doanh. Điện toán đám mây di động kết hợp thêm điện toán đám mây với các thiết bị đầu cuối di động và công nghệ không dây được tích hợp để cho phép kết nối liên tục và linh hoạt. Các doanh nghiệp hiện nay có xu hướng sử dụng điện toán đám mây để giảm thiểu chi phí vận hành hệ thống so với tự quản trị và duy trì trung tâm dữ liệu của riêng mình. Vấn đề khó khăn nhất là làm thế nào để đảm bảo dữ liệu của họ và đặc biệt là dữ liệu cá nhân nhạy cảm của khách hàng như số điện thoại, thông tin thẻ tín dụng, số tài khoản ngân hàng, số chứng minh thư, số thẻ căn cước, số định danh cá nhân… được an toàn khi lưu trữ và xử lý trên môi trường này.

Việc bảo mật cho dữ liệu của người dùng trên đám mây khác khá nhiều so với cách lưu trữ truyền thống, và bảo mật đối với dữ liệu người dùng kết nối đám mây trên các thiết bị di động có tài nguyên và năng lực xử lý giới hạn cũng khó khăn hơn so với trên máy tính thông thường. Cách thức truyền thống có thể là cài đặt phần mềm bảo mật trên thiết bị, nhưng ở đây sẽ đưa ra một giải pháp cho phép bảo vệ dữ liệu cá nhân khi giao dịch trên mạng. Việc bảo vệ thông tin của các hệ thống đám mây cũng dựa trên các nguyên tắc đã biết về bảo mật, tính sẵn sàng và tính toàn vẹn, nhưng được áp dụng cho các kiến trúc phân tán, ảo hóa và mềm dẻo.

2. Cơ sở lý thuyết

2.1 Bảo mật dữ liệu

Bảo mật dữ liệu nói chung liên quan đến nhiều khía cạnh như bảo mật khi dữ liệu được lưu trữ hoặc khi dữ liệu được truyền trên mạng hay trong quá trình dữ liệu được xử lý và chống truy cập trái phép.

Đối với điện toán đám mây thì dữ liệu cũng cần được đảm bảo các vấn đề nêu trên. Nhà cung cấp dịch vụ đám mây cần thiết kế hệ thống của riêng họ để bảo vệ khỏi các cuộc tấn công bên ngoài, sử dụng các giao thức mã hóa đủ mạnh để bảo vệ khi thông tin được truyền tải, lưu trữ và chia sẻ. Hơn nữa, kiến trúc của hệ thống phải được thiết kế để có khả năng chịu tải, chịu lỗi và chống lại các cuộc tấn công nhằm làm quá tải hệ thống. Tuy vậy mô hình này có một lợi điểm đáng kể là đảm bảo an toàn và độ sẵn sàng cao của dữ liệu, do cách thức lưu trữ là phân tán nên tránh được rủi ro mất mát so với mô hình lưu trữ tập trung truyền thống.

Thế nhưng điện toán đám mây vẫn còn khá yếu trong khía cạnh xác thực nguồn gốc dữ liệu và chống giả mạo cá nhân hay đối tác giao dịch. Do vậy, nghiên cứu này đưa ra một cách thức xác thực người dùng có sử dụng thiết bị di động và dữ liệu được lưu trữ trên đám mây.

[*] *Tác giả liên hệ*
Email: donhuhai@humg.edu.vn

2.2 Thuật toán mã hóa RSA

Để bảo mật dữ liệu có thể sử dụng thuật toán mã hóa khóa đối xứng như 3DES hoặc AES, tuy nhiên những thuật toán này phù hợp và hiệu quả hơn với yêu cầu trao đổi dữ liệu lớn, việc cài đặt chúng phức tạp hơn RSA cũng như cần có cơ chế để thỏa thuận khóa giữa hai đối tác truyền tin trước khi sử dụng.

RSA là hệ mật mã khóa công khai hay mã hóa khóa bất đối xứng được biết đến nhiều và sử dụng rộng rãi nhất hiện nay. Trong đó mỗi người dùng có một cặp khóa gồm một khóa bí mật (private key) dùng để giải mã và một khóa công khai (public key) được đối tác truyền tin dùng để mã hóa dữ liệu. Ngoài ra RSA cũng được sử dụng trong sơ đồ chữ ký số nhằm đảm bảo tính toàn vẹn của dữ liệu và xác thực đối tác truyền tin.

RSA dựa trên các phép toán lũy thừa trong trường hữu hạn các số nguyên theo modulo nguyên tố. Cụ thể, mã hoá hay giải mã là các phép toán luỹ thừa theo modulo số rất lớn, có thể là 1024 bit (1024 bit tương đương 10^{350}) hoặc 2048 và 4096 bit cho các yêu cầu cần độ bảo mật cao. Việc thám mã hay phá mã, tức là tìm khoá riêng khi biết khoá công khai, dựa trên bài toán khó là phân tích một số rất lớn đó ra thừa số nguyên tố. Nếu không có thông tin gì, thì ta phải lần lượt kiểm tra tính chia hết của số đó cho tất cả các số nguyên tố nhỏ hơn căn của nó. Đây là việc làm tốn năng lực tính toán và cần rất nhiều thời gian, cho nên không khả thi trong thực tế.

Mô tả thuật toán RSA và các bước sử dụng:

- *Tạo khoá:*

Mỗi đầu cần tạo một khoá công khai và một khóa riêng tương ứng theo các bước sau:

(1) Tạo 2 số nguyên tố lớn ngẫu nhiên và khác nhau p và q; p và q có độ lớn xấp xỉ nhau.

(2) Tính n = p.q và $\Phi(n) = (p-1)(q-1)$.

(3) Chọn một số nguyên ngẫu nhiên e thỏa mãn $1 < e < \Phi(n)$ và $\gcd(e,\Phi(n)) = 1$, trong đó gcd là hàm tìm ước chung lớn nhất của 2 số.

(4) Giải phương trình sau để tìm khoá giải mã d: $e.d \equiv 1 \bmod \Phi(n)$ với $0 \le d \le \Phi(n)$.

(5) Khoá công khai là cặp số (n,e). Khoá riêng bí mật là bộ (d, p, q).

- *Mã hoá dùng khóa công khai:*

B mã hoá một thông báo m để gửi cho A bản mã.

B phải thực hiện các bước:

(1) Thu nhận khoá công khai (n,e) của A.

(2) Biểu diễn bản tin dưới dạng một số nguyên m trong khoảng [0, n-1]

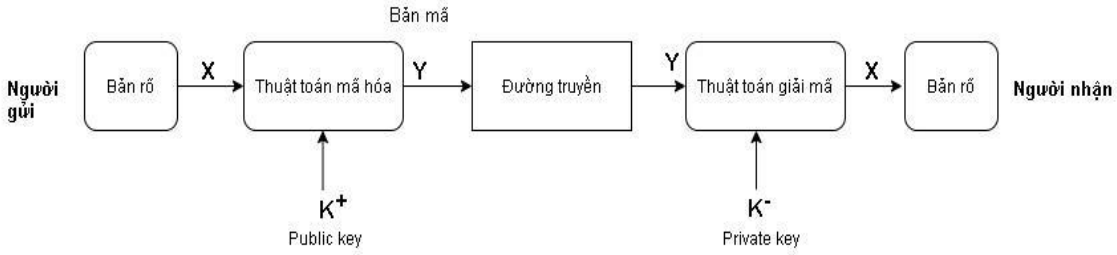
(3) Tính $c = m^e \bmod n$, trong đó $0 \le m < n$

(4) Gửi bản mã c cho A.

- *Giải mã dùng khóa bí mật:*

Để khôi phục bản rõ m từ bản mã c, A phải thực hiện phép tính sau bằng cách dùng khoá riêng: $m = c^d \bmod n$.

Dưới đây là sơ đồ mô tả hệ mật RSA.

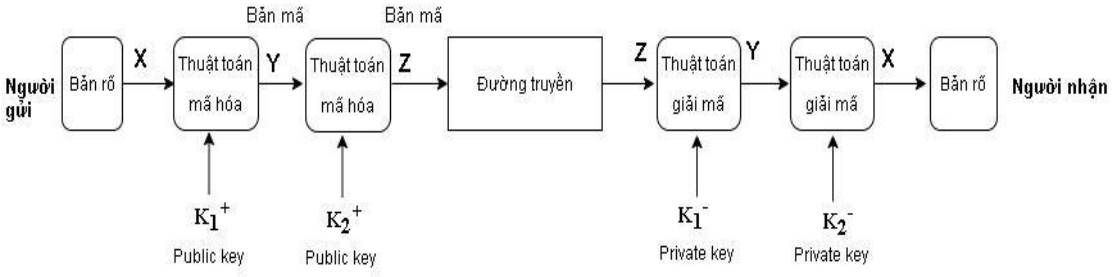


Hình 1. Hệ mật mã khóa công khai RSA

Hệ mật RSA hiện tại dùng khóa 1024 bít hoặc 2048 bít và vẫn được coi là an toàn vì chưa bị phá vỡ trên thực tế. Tuy vậy năng lực tính toán của máy tính ngày càng tăng lên và các nhà nghiên cứu cho rằng RSA 1024 bít có thể sớm bị phá vỡ, khuyến nghị dùng 2048 bít. Đối với những ứng dụng cần độ bảo mật cao thì có thể xem xét dùng khóa 4096 bít. Tuy nhiên ta biết rằng tốc độ thực hiện của thuật toán RSA trên các số lớn như vậy lầ rất chậm vì thời gian tính toán tăng theo hàm mũ, khó áp dụng vào thực tế.

Một giải pháp được đưa ra nhằm cân bằng giữa yêu cầu cần độ bảo mật cao hơn và thời gian thực hiện thuật toán mã hóa không tăng lên đáng kể là áp dụng hai lần thuật toán RSA (2-RSA) với khóa 2048 bit. Ở đây tiến hành mã hóa bản rõ X với khóa công khai $K1^+$ qua thuật toán lần thứ nhất thu được bản mã Y, sau đó Y là đầu vào để thuật toán mã hóa lần thứ hai cho ra bản mã Z với khóa công khai $K2^+$. Quá trình giải

mã tiến hành ngược lại với khóa bí mật $K1^-$ để thu được Y trước rồi giải mã với khóa bí mật $K2^-$ để thu được bản rõ X ban đầu. Như vậy so sánh 2-RSA với RSA ta thấy thời gian mã hóa hầu như không thay đổi nhiều, chỉ gấp 2 lần so với RSA. Nhưng độ an toàn được nâng lên đáng kể, thời gian phá mã 2-RSA bằng vét cạn sẽ tăng lên gấp nhiều lần so với RSA. Bởi vì để phá mã thì phải thực hiện thám mã Z (giả sử phải thực hiện m phép thử) để có được Y rồi từ đó tiếp tục thám mã Y (giả sử phải thực hiện n phép thử) để có được X. Như vậy, mỗi một khả năng có thể có của khóa của Z thì phải thử tất cả các khả năng có thể có của khóa của Y, khi đó để tìm ra được X phải thử tất cả các khả năng kết hợp khóa của Z và khóa của Y là m.n phép thử.

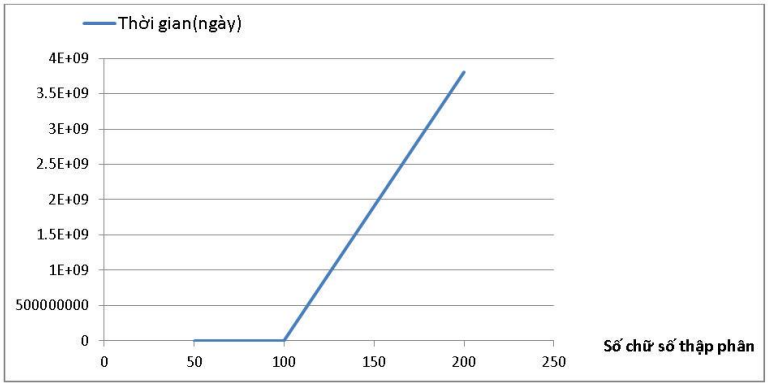


Hình 2. Hệ mật mã khóa công khai 2-RSA

Trong 2-RSA nếu khóa có độ dài 512 bit tức là có thể lưu được số nguyên lớn có 155 chữ số thập phân, tương tự với khóa 1024 bit là 309 chữ số thập phân và khóa có độ dài 2048 bit là 617 chữ số thập phân. Dựa trên thực nghiệm đã tính được thời gian cần thiết để phá mã 2-RSA với số chữ số thập phân ít hơn 100 và dùng mô phỏng để tính toán ước lượng cho các trường hợp lớn hơn 100 chữ số như Bảng 1 dưới đây.

Bảng 1. Khoảng thời gian ước lượng cần thiết phá mã 2-RSA

Số chữ số thập phân	Số phép tính bít	Thời gian
50	$1,4.10^{10}$	3,9 giờ
75	9.10^{12}	104 ngày
100	$2,3.10^{15}$	74 năm
200	$1,2.10^{23}$	$3,8.10^{9}$ năm
300	$1,5.10^{29}$	$4,9.10^{15}$ năm
500	$1,3.10^{39}$	$4,2.10^{25}$ năm



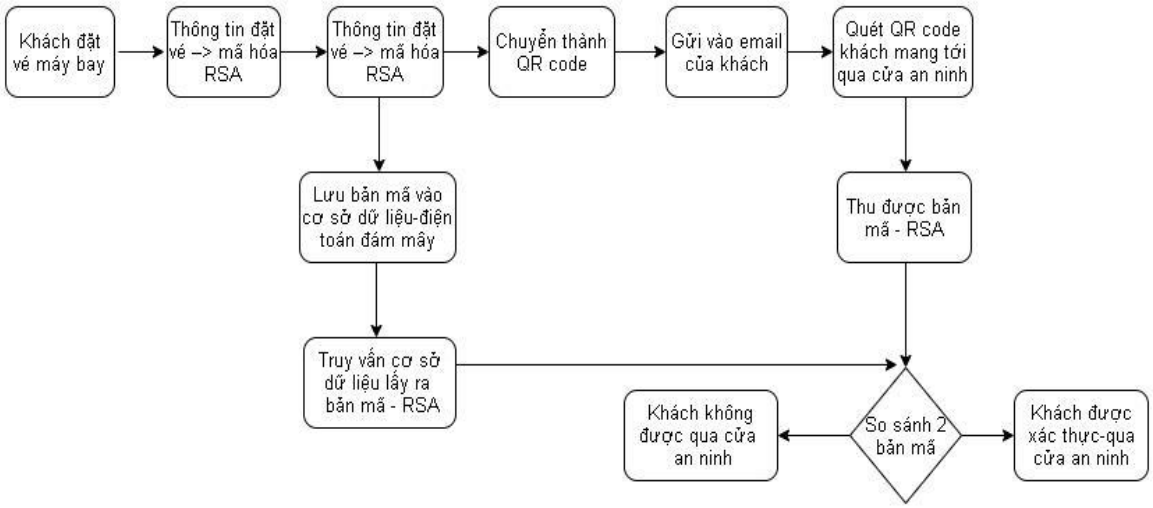
Hình 3. Minh họa thời gian phá mã 2-RSA

3. Giải pháp tăng cường bảo mật dữ liệu người dùng trên điện toán đám mây

Ở đây đưa ra một mô hình lưu trữ và xử lý dữ liệu cá nhân của hành khách đi đường hàng không được bảo mật và thuận tiện trong khâu kiểm tra an ninh tại sân bay. Việc thông tin hành khách đi máy bay bị lộ và các hãng taxi có được đã từng xảy ra. Do vậy việc giữ bí mật dữ liệu hành khách là cần thiết để bảo vệ quyền riêng tư cá nhân của họ.

Trong mô hình này, khi khách đặt vé online xong thông tin đặt vé sẽ được hệ thống mã hóa bằng thuật toán RSA 2 lần. Sau đó thông tin ở dạng bản mã RSA (đảm bảo tính bí mật) được lưu vào cơ sở dữ liệu

trên điện toán đám mây, mặt khác nó được chuyển thành mã QR và gửi cho khách qua email. Đến thời điểm khách làm thủ tục qua cửa an ninh sân bay, họ đưa mã QR đã được gửi qua email trên thiết bị di động của mình cho nhân viên an ninh kiểm tra, mã này được máy quét QR code chuyển thành bản mã RSA ban đầu. Đồng thời thông tin của khách dưới dạng mã RSA được truy vấn từ cơ sở dữ liệu đám mây, hệ thống sẽ so khớp 2 mã RSA này, nếu khớp nhau thì hành khách được qua cửa an ninh và ngược lại họ sẽ không được qua. Việc dùng mã QR nhằm giúp nhân viên an ninh hàng không sử dụng máy quét cầm tay để nhanh chóng lấy được thông tin đặt vé khách mang đến qua thiết bị di động, rút ngắn thời gian kiểm tra cho từng hành khách và bảo mật thông tin của khách.



Hình 4. Xác thực bằng mã QR và 2-RSA

Đồng thời mô hình này lưu trữ dữ liệu dựa trên điện toán đám mây để đảm bảo độ sẵn sàng cao, có tính linh hoạt và sử dụng cơ chế mã mật bảo vệ dữ liệu cá nhân của hành khách, chống lại việc khai thác bất hợp pháp từ hacker hoặc chính nhân viên ngành hàng không. Dữ liệu được mã hóa bằng hệ mật mã RSA cho nên việc bị tấn công vét cạn là rất khó xảy ra trong thời gian thực tế và cũng không cần phải thiết lập cơ chế trao đổi khóa như hệ mật mã khóa đối xứng, vì đã tận dụng được ưu điểm của hệ mật khóa công khai là có sẵn cơ chế trao đổi khóa. Tuy nhiên nhược điểm của mô hình này là không thích hợp nếu lượng dữ liệu cần mã hóa lớn, do thuật toán RSA có thời gian thực thi chậm, hơn nữa mã QR cũng chỉ thích hợp với lượng thông tin nhỏ.

Tóm lại, đối với điện toán đám mây có sử dụng các thiết bị di động thì các vấn đề bảo mật chính gồm: thiết bị đầu cuối di động, mạng di động và đám mây di động. Trong thiết bị đầu cuối di động có thể dùng các ứng dụng Anti virus để ngăn chặn các phần mềm độc hại. Ngoài ra cũng cần thường xuyên cài đặt các bản vá lỗi hệ thống cũng như khuyến nghị điều chỉnh hành vi của người dùng để giảm thiểu việc bị tấn công. Đối với mạng di động, có thể sử dụng các giao thức bảo mật và thuật toán mã hóa dữ liệu mạnh để tránh rò rỉ thông tin và đảm bảo toàn vẹn dữ liệu. Trong đám mây di động, có các vấn đề như độ tin cậy của nền tảng, bảo vệ dữ liệu và quyền riêng tư của người dùng có thể được ngăn chặn bằng cách tích hợp các công nghệ bảo mật hiện tại có cải tiến nhằm tăng cường việc quản lý khóa, mã hóa dữ liệu, xác thực và kiểm soát truy cập như mô hình đã được đề xuất ở trên.

4. Kết luận

Nghiên cứu đưa ra mô hình bảo vệ dữ liệu bằng cách thực hiện hai lần thuật toán RSA kết hợp sử dụng mã QR để bảo mật tốt hơn cho dữ liệu cá nhân người dùng được lưu trữ trên điện toán đám mây. Việc sử dụng 2-RSA với độ dài khóa 2048 bit sẽ nâng cao độ an toàn hơn nhiều so với RSA đơn thuần để chống lại các hình thức tấn công mạng phổ biến, đồng thời tốc độ chạy thuật toán vẫn đảm bảo khả thi trong thực tế. Mô hình cũng đảm bảo tính toàn vẹn của dữ liệu và xác thực người dùng, ứng dụng trong môi trường đám mây có sử dụng thiết bị di động.

Tài liệu tham khảo

Atul Kahate. (2011). Cryptography and Network Security Second Edition. Thirteenth reprint. Tata McGraw Hill Education Private Limited.

Harish Sharma, Dr. shiv kumar & Kavita Sharma. (2014). Dual mode public key cryptosystem. IJARCSSE Volume 8.

H. Wu, Y. Ding, C. Winer, L. Yao. (2010). Network security for virtual machine in cloud computing, 5th International Conference on Computer Sciences and Convergence Information Technology, pp. 18-21.

J. Wei, X. Zhang, G. Ammons, V. Bala, P. Ning. (2009). Managing security of virtual machine images in a cloud environment, Proceedings of the 2009 ACM Workshop on Cloud Computing Security, in: CCSW '09, New York, NY, USA, ISBN: 978-1-60558-784-4, pp. 91-96.

Klymash, Mykhailo, Olga Shpur, Orest Lavriv, and Nazar Peleh. (2019). "Information Security in Virtualized Data Center Network." In 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), pp. 419-422. IEEE.

Mykola karpinskyy & Yaroslav Kinakh. (2003). Reliability of RSA Algorithm and its Computational Complexity. IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.

M. Christodorescu, R. Sailer, D.L. Schales, D. Sgandurra, D. Zamboni. (2009). Cloud security is not (just) virtualization security: A short paper, Proceedings of the 2009 ACM Workshop on Cloud Computing Security, in: CCSW '09, ACM, New York, NY, USA, ISBN: 978-1-60558-784-4, pp. 97-102.

Wang, Juan, Chengyang Fan, Jie Wang, Yueqiang Cheng, Yinqian Zhang, Wenhui Zhang, Peng Liu, and Hongxin Hu. (2019). "SvTPM: A Secure and Efficient vTPM in the Cloud." arXiv preprint arXiv:1905.08493.

ABSTRACT

Data security in Cloud computing

Đỗ Như Hải[1]

[1]*Hanoi University of Mining and Geology*

Cloud computing technology develops rapidly with the number of mobile users increasing significantly. Privacy and data security for mobile device users is an interesting aspect. The research aims to propose a solution that combines existing algorithms to enhance security for cloud computing. Security for cloud computing involves many issues such as network security, web security, data access, authentication, authorization, data security, etc. In this study, analytical and aggregated methods are used, with a range of solutions to enhance the security of cloud computing with mobile devices to ensure data integrity and security. Which can use the RSA variant to ensure high security capabilities.

Keywords: Cloud computing; block cipher algorithm; public key encryption algorithm; cloud computing security.

Nghiên cứu mô hình học máy Naïve Bayes trong phân lớp văn bản; Ứng dụng phân lớp cho tập dữ liệu các nhận xét trên Twitter

Đặng Văn Nam[1,*]
[1]*Trường Đại học Mỏ - Địa chất*

TÓM TẮT

Trí tuệ nhân tạo (AI - Artificial Intelligence), Dữ liệu lớn (Big Data), Internet kết nối vạn vật (IoT - Internet of Things)…là những từ khóa của cuộc cách mạng công nghiệp 4. Các ứng dụng của trí tuệ nhân tạo nói chung và của Học máy (Machine Learning) nói riêng đã đem lại hiệu quả to lớn trong nhiều lĩnh vực của đời sống con người. Những năm gần đây, Học máy có vai trò quan trọng trong xử lý ngôn ngữ tự nhiên (NLP - Natural Language Processing). Một trong những bài toán rất quan trọng và phổ biến trong NLP đó là phân lớp văn bản (Text Classification). Có rất nhiều thuật toán học máy được sử dụng để giải quyết cho bài toán này. Tuy nhiên, Naïve Bayes là thuật toán có thời gian chạy nhanh và độ chính xác cao nên thường được sử dụng cho các bài toán phân lớp văn bản. Bài báo trình bày cụ thể việc xây dựng một mô hình học máy với thuật toán Naïve Bayes sử dụng đặc trưng TF-IDF (Term Frequency - Inverse Document Frequency) trong phân lớp văn bản. Tập dữ liệu thực nghiệm bao gồm 56 700 nhận xét (comments) trên Twitter và văn bản được phân thành 2 lớp bao gồm: lớp 0 - Non Toxic, lớp 1 - Toxic. Đồng thời, trong nội dung bài báo tác giả cũng thực hiện so sánh độ chính xác của Naïve Bayes với thuật toán SVN (Support Vector Machine) và KNN (K- Nearest Naighbors) trên cùng tập dữ liệu thực nghiệm.

Từ khóa: Học máy; naïve bayes; xử lý ngôn ngữ tự nhiên; phân lớp văn bản, tf-idf.

1. Mở đầu

Xử lý ngôn ngữ tự nhiên là một trong những lĩnh vực quan trọng của trí tuệ nhân tạo. Rất nhiều ứng dụng của NLP đã, đang và ngày càng có ảnh hưởng sâu rộng tới mọi mặt của đời sống con người. Có thể hiểu NLP là một lĩnh vực khoa học máy tính, kỹ thuật thông tin và trí tuệ nhân tạo tập trung vào nghiên cứu các tương tác về mặt ngôn ngữ giữa máy tính và con người, cụ thể hơn là làm thế nào để lập trình cho máy tính xử lý và phân tích một lượng lớn dữ liệu ngôn ngữ tự nhiên. Nói cách khác, NLP quan tâm tới việc làm thế nào để máy tính có thể hiểu và vận dụng được các tập dữ liệu sẵn có dưới dạng ngôn ngữ tự nhiên. Một số ứng dụng quan trọng của NLP có thể chỉ ra như: Các hệ thống máy dịch, ví dụ Google translation mà chúng ta đã quá quen thuộc; Ứng dụng trong xử lý văn bản và ngôn ngữ; Ứng dụng trong tóm tắt và phân loại văn bản, Hệ thống chatbot, Tổng đài tự động (ACC), …

Những năm gần đây, Học máy đang trở thành một phần không thể thiếu trong quá trình xử lý ngôn ngữ tự nhiên. Từ việc xây dựng các tập qui tắc bằng tay đòi hỏi rất nhiều thời gian, công sức, các nghiên cứu đang hướng đến việc sử dụng cơ sở dữ liệu lớn để tự động (hoặc bán tự động) sinh ra các quy tắc đó. Phương pháp này đã cho những kết quả khả quan trong nhiều lĩnh vực khác nhau của NLP.

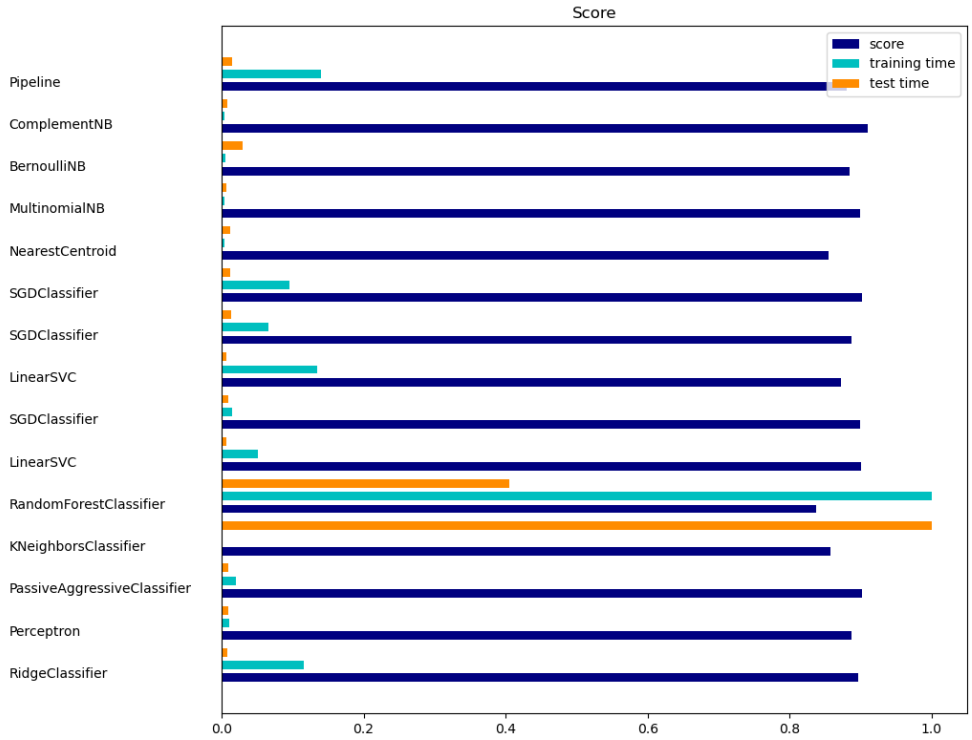
Trong các ứng dụng của NLP, bài toán phân lớp văn bản là một trong những bài toán quan trọng và kinh điển nhất. Mục tiêu của một hệ thống phân lớp văn bản là nó có thể tự động phân lớp một văn bản cho trước, để xác định xem văn bản đó thuộc lớp nào. Các ứng dụng của phân lớp văn bản rất đa dạng như: Hiểu được ý nghĩa, đánh giá, bình luận của người dùng; Lọc email rác; Phân tích cảm xúc; Phân lớp tin tức, các bài báo điện tử…Bài toán phân lớp văn bản là một bài toán học có giám sát (supervised learning) trong học máy, tập văn bản huấn luyện đã được gán nhãn và được sử dụng để thực hiện phân lớp, việc thực hiện gán các nhãn lên một văn bản mới sẽ dựa trên trên mức độ tương tự của văn bản đó so với các văn bản đã được gán nhãn trong tập huấn luyện. Để giải quyết một bài toán phân lớp văn bản, thường trải qua 4 giai đoạn:

- Chuẩn bị dữ liệu (Data Preparation)
- Trích chọn đặc trưng (Feature Engineering)
- Xây dựng mô hình phân lớp (Buil Model)
- Tinh chỉnh mô hình và cải thiện hiệu năng (Improve Performance)

[*] *Tác giả liên hệ*
Email: dangvannam@humg.edu.vn

Như đã trình bày ở trên, Phân lớp văn bản là một bài toán học có giám sát, có rất nhiều thuật toán học có giám sát như: Support Vector Machines, Decision Trees, Naïve Bayes, K-Nearest Neighbors, Random Forest…Tuy nhiên, Naïve Bayes Classifiers là một trong những thuật toán đem lại hiệu quả cho những bài toán phân lớp trong NLP nhờ độ chính xác cao, thời gian huấn luyện mô hình nhỏ.

Hình 1 là kết quả so sánh các thuật toán học máy khác nhau trong việc phân lớp văn bản trên cùng một tập dữ liệu bao gồm 18000 văn bản được gán nhãn vào 20 lớp, sử dụng cùng một phương pháp trích chọn đặc trưng và cùng một cấu hình máy tính như nhau. Các kết quả chỉ ra bao gồm: Độ chính xác (score) của thuật toán trên tập kiểm thử, thời gian thuật toán sử dụng để huấn luyện trên tập huấn luyện (Training time) và thời gian thuật toán sử dụng để kiểm thử mô hình trên tập kiểm thử (Test time). Kết quả cho thấy thuật toán Naïve Bayes với ba biến thể khác nhau (ComplementNB, BernoulliNB và MultinomialNB) đều cho Score cao và Training time, Test time nhỏ hơn các thuật toán khác.



Hình 1. So sánh các thuật toán học máy khác nhau trong phân lớp văn bản

Nội dung của bài báo này tác giả sẽ nghiên cứu về thuật toán Naïve Bayes, và ứng dụng của Naïve Bayes trong bài toán phân lớp văn bản trên tập dữ liệu cụ thể là những nhận xét (comments) thu được từ mạng xã hội Twitter, các nhận xét được phân thành 2 lớp, gán nhãn 0 - Non Toxic (nhận xét không chứa các từ độc hại, không mang ý nghĩa tục tĩu, kích động….); gán nhãn 1 - Toxic (nhận xét có chứa những từ độc hại, mang ý nghĩa tục tĩu, kích động).

Như đã biết, độ chính xác của một mô hình học máy không những phụ thuộc vào thuật toán và thiết lập các tham số tương ứng của thuật toán đó, mà một trong những yếu tố rất quan trọng ảnh hưởng lớn tới độ chính xác đó là phương pháp trích chọn đặc trưng, chuyển đổi dữ liệu dạng văn bản đã được xử lý về dạng vector thuộc tính dạng số học. Có nhiều cách khác nhau để đưa dữ liệu dạng văn bản về dữ liệu dạng số như: Bag of words (BoW), TF-IDF, Word Embeddings, n-grams model, skip-Gram model…Bài báo này cũng trình bày về phương pháp trích chọn đặc trưng cơ bản được sử dụng rộng rãi là TF-IDF, và áp dụng kỹ thuật này cho tập dữ liệu thử nghiệm.

2. Thuật toán phân lớp Naïve Bayes

Naive Bayes Classifiers (NBC) là một trong những thuật toán tiêu biểu cho bài toán phân lớp dựa trên lý thuyết xác suất áp dụng định lý Bayes.

Định lý Bayes cho phép chúng ta có thể tính toán một xác suất chưa biết dựa vào các xác suất có điều kiện khác. Với công thức tổng quát tính xác suất của biến cố A với điều kiện biến cố B_k xảy ra trước (hay được gọi là xác suất hậu nghiệm):

Với P(A) > 0 và $\{B_1, B_2, \dots, B_n\}$ là một hệ đầy đủ các biến cố thỏa mãn tổng xác suất của hệ bằng 1 ($\sum_{k=1}^{n} P(B_k) = 1$) và từng đôi một xung khắc ($P(B_i \cap B_j) = 0$). Khi đó ta có:

$$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{P(A)} = \frac{P(A|B_k)P(B_k)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)} \quad (1)$$

Bộ phân lớp Naive bayes hoạt động như sau:

- Gọi D là tập dữ liệu huấn luyện, trong đó mỗi phần tử dữ liệu A chứa n giá trị thuộc tính $B_1, B_2,...,B_n$ được biểu diễn bằng một vector n thành phần $\{x_1,x_2,...,x_n\}$
- Giả sử có m lớp $C_1, C_2,..,C_m$. Cho một phần tử dữ liệu A, bộ phân lớp sẽ gán nhãn cho A là lớp có xác suất hậu nghiệm lớn nhất. Cụ thể, bộ phân lớp Bayes sẽ dự đoán A thuộc vào lớp C_i nếu và chỉ nếu:

$$P(C_i|A) > P(C_j|A) \quad (i \le i, j \le m \; i \ne j) \quad (2)$$

Giá trị này sẽ tính dựa trên định lý Bayes.

- Để tìm xác suất lớn nhất, ta nhận thấy các giá trị P(A) là giống nhau với mọi lớp nên không cần tính. Do đó ta chỉ cần tìm giá trị lớn nhất của $P(A|C_i) * P(C_i)$. Chú ý rằng $P(C_i)$ được ước lượng bằng $|D_i|/|D|$, trong đó D_i là tập các phần tử dữ liệu thuộc lớp C_i. Nếu xác suất tiền nghiệm $P(C_i)$ cũng không xác định được thì ta coi chúng bằng nhau $P(C_1) = P(C_2) = ... = P(C_m)$, khi đó ta chỉ cần tìm giá trị $P(A|C_i)$ lớn nhất.
- Khi số lượng các thuộc tính mô tả dữ liệu là lớn thì chi phí tính toán $P(A|C_i)$ là rất lớn, do đó có thể giảm độ phức tạp của thuật toán Naive Bayes nếu giả thiết các thuộc tính độc lập nhau. Khi đó ta có thể tính:

$$P(A|C_i) = P(x_1|C_i) \ldots\ldots P(x_n|C_i) \quad (3)$$

NBC có thể hoạt động với các vector đặc trưng mà một phần là liên tục (sử dụng Gaussian Naive Bayes), phần còn lại ở dạng rời rạc (sử dụng Multinomial hoặc Bernoulli). Trong phần thực nghiệm, tác giả sử dụng MultinomialNB để xây dựng mô hình. Mỗi văn bản được biểu diễn bởi một vector có độ dài d chính là số từ trong từ điển. Giá trị của thành phần thứ i trong mỗi vector chính là số lần từ thứ i xuất hiện trong văn bản đó.

Khi đó, P(xi|Cj) tỉ lệ với tần suất từ thứ i (hay thuộc tính thứ i cho trường hợp tổng quát) xuất hiện trong các văn bản của lớp Cj. Giá trị này có thể được tính bằng cách

$$P(x_i|C_j) = \frac{N_{ci}}{N_c} \quad (4)$$

Trong đó:

- N_{ci} là tổng số lần từ thứ i xuất hiện trong các văn bản của lớp C_i, nó được tính bằng tổng của tất cả các thành phần thứ i của các vector thuộc tính ứng với lớp C_j
- N_c là tổng số từ (kể cả lặp) xuất hiện trong lớp C_j. Nói cách khác, nó bằng tổng độ dài của toàn bộ các văn bản thuộc vào lớp C_j.

Cách tính này có một hạn chế là nếu có một từ mới chưa bao giờ xuất hiện trong lớp Cj thì biểu thức (1) sẽ bằng 0, điều này dẫn P(A|Ci) = 0 bất kể các giá trị còn lại có lớn thế nào. Việc này sẽ dẫn đến kết quả không chính xác. Để giải quyết việc này, một kỹ thuật được gọi là Laplace smoothing được áp dụng như trong biểu thức (5):

$$P(x_i|C_j) = \frac{N_{ci}+\alpha}{N_c+d\alpha} \quad (5)$$

Với α là một số dương, thường bằng 1, để tránh trường hợp tử số bằng 0. Mẫu số được cộng với dα để đảm bảo tổng xác suất $\sum_{i=1}^{d} P(x_i|C_j) = 1$ (Vũ Hữu Tiệp, 2018; Haiyi Zhang, DiLi, 2007; Wei Zhang, Feng Gao, 2011)

3. Trích chọn đặc trưng TF-IDF

Thuật ngữ TF-IDF (Term Frequency - Inverse Document Frequency) là một phương thức thống kê được biết đến rộng rãi để xác định độ quan trọng của một từ trong đoạn văn bản trong một tập nhiều đoạn văn bản khác nhau.

TF-IDF xác định trọng số của một từ trong văn bản thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản. Giá trị TF-IDF cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu. Một vài biến thể của TF-IDF thường được sử dụng trong các hệ thống tìm kiếm như một công cụ chính để đánh giá và sắp xếp văn bản dựa vào truy vấn của người dùng. TF-IDF cũng được sử dụng để lọc những từ stopwords trong các bài toán như tóm tắt văn bản và phân lớp văn bản.

TF (Term Frequency) - Tần suất xuất hiện của từ là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Như vậy, TF thường được chia cho độ dài văn bản (tổng số từ trong một văn bản).

$$TF(t,d) = \frac{f(t,d)}{\max\{f(w,d):w \in d\}} \quad (6)$$

Trong đó:

- TF(t, d) - Tần suất xuất hiện của từ t trong văn bản d.
- f(t,d) - Số lần xuất hiện của từ trong văn bản d.
- max{f(w,d): w ∈ d} - Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d.

IDF (Inverse Document Frequency) - Nghịch đảo tần suất của văn bản, giúp đánh giá tầm quan trọng của một từ. Khi tính tần số xuất hiện TF thì các từ đều được coi là quan trọng như nhau. Tuy nhiên có một số từ thường được được sử dụng nhiều nhưng không quan trọng để thể hiện ý nghĩa của đoạn văn. Vì vậy ta cần giảm đi mức độ quan trọng của những từ đó bằng cách sử dụng IDF:

$$IDF(t, D) = log \frac{|D|}{|\{d \in D : t \in d\}|} \quad (7)$$

Trong đó:

- IDF(t, D) - Giá trị IDF của từ t trong tập văn bản D.
- |D| - Tổng số văn bản trong tập D.
- |{d ∈ D : t ∈ d}| - Thể hiện số văn bản trong tập D có chứa từ t.

Cơ số logarit trong công thức này không làm thay đổi giá trị IDF của từ mà chỉ thu hẹp khoảng giá trị của từ đó. Việc sử dụng logarit nhằm giúp giá trị TF-IDF của một từ nhỏ hơn, do công thức tính TF-IDF của một từ trong một văn bản là tích của TF và IDF của từ đó. Công thức tính TF_IDF được xác định như sau:

$$TF - IDF(t, d, D) = TF(t, d) x\ IDF(t, D) \quad (8)$$

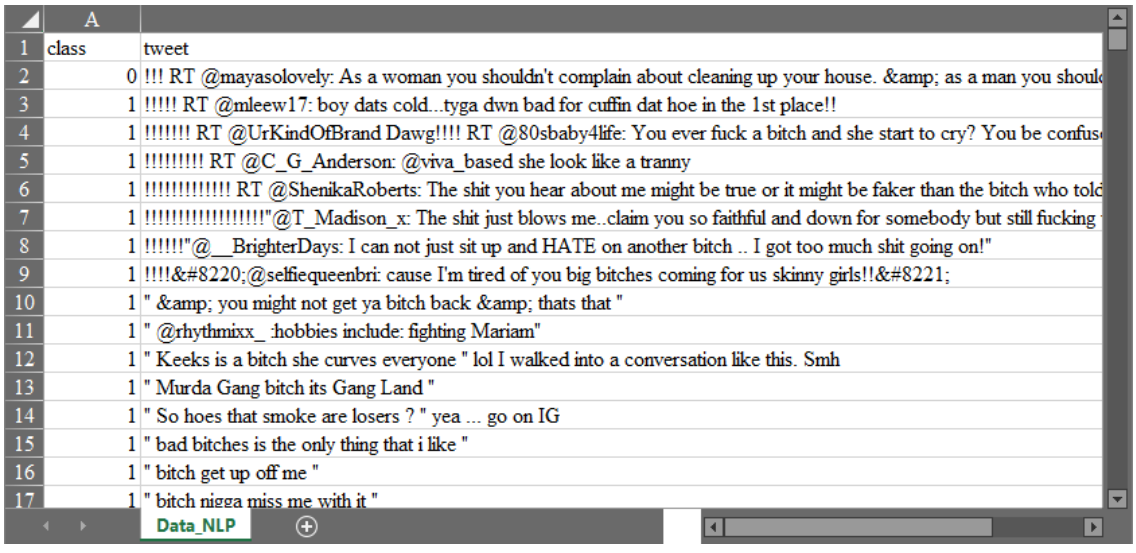
Từ (8) cho thấy những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao chính là các từ khóa của văn bản đó.

4. Ứng dụng thuật toán Naive Bayes trên tập dữ liệu Twitter

4.1. Chuẩn bị dữ liệu

Tập dữ liệu các nhận xét (comments) được thu thập từ Twitter và lưu trữ trong tệp Data_NLP.csv như minh họa trong Hình 2, chứa 56 700 dòng dữ liệu; với 2 cột bao gồm:

- Cột class: cho biết lớp mà comments được gán nhãn; các comments được gán vào 2 lớp: 0 - Non Toxic (33 847 comments) | 1 - Toxic (22 853 comments).
- Cột tweet: cho biết nội dung của đoạn văn bản comments. Đây là đoạn văn bản gốc được lấy về từ Twitter nên có thể thấy chứa rất nhiều nhiễu (noise)



	A	
1	class	tweet
2	0	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. & as a man you shoul
3	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
4	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confus
5	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
6	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told
7	1	!!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking
8	1	!!!!!!"@__BrighterDays: I can not just sit up and HATE on another bitch .. I got too much shit going on!"
9	1	!!!!“@selfiequeenbri: cause I'm tired of you big bitches coming for us skinny girls!!”
10	1	" & you might not get ya bitch back & thats that "
11	1	" @rhythmixx_ :hobbies include: fighting Mariam"
12	1	" Keeks is a bitch she curves everyone " lol I walked into a conversation like this. Smh
13	1	" Murda Gang bitch its Gang Land "
14	1	" So hoes that smoke are losers ? " yea ... go on IG
15	1	" bad bitches is the only thing that i like "
16	1	" bitch get up off me "
17	1	" bitch nigga miss me with it "

Data_NLP

Hình 2. Tập dữ liệu thô đã được gán nhãn ban đầu Data_NLP.csv

Có thể thấy các câu comments là dữ liệu thô ban đầu (raw data) được lấy về từ Twitter chứa rất nhiều nhiễu như: các ký hiệu đặc biệt, các thẻ HTML, JavaScript, các từ viết tắt, các liên kết, email, con số, tag ....Do đó cần phải được chuẩn bị trước khi sử dụng cho các mục đích tiếp theo. Như trong phần mở đầu đã chỉ ra, chuẩn bị dữ liệu (Data Preparation) là giai đoạn thực hiện đầu tiên trong quá trình xây dựng một mô hình học máy cho bài toán phân lớp văn bản.

Các phương pháp mà nhóm tác giả sử dụng để chuẩn bị dữ liệu áp dụng cho tập dữ liệu Data_NLP bao

gồm:

• Loại bỏ nhiễu trong các comments bao gồm: Loại bỏ các đường link, địa chỉ email; Loại bỏ các ký tự đặc biệt, ký tự số.

• Loại bỏ các từ StopWords: StopWords là những từ xuất hiện nhiều trong ngôn ngữ tự nhiên, tuy nhiên lại không mang nhiều ý nghĩa. Để loại bỏ StopWords có 2 cách chính là dùng từ điển (corpus) hoặc dựa theo tần suất xuất hiện của từ. Với tiếng anh, những từ StopWords bao gồm: the, this, that, these, is, are, was, were….đã được tổng hợp vào trong một corpus trong thư viện NLTK.

• Chuẩn hóa từ: Trong tiếng Anh, mỗi một từ có thể có nhiều biến thể khác nhau. Điều này làm cho việc so sánh giữa các từ là không thể mặc dù về mặt ý nghĩa cơ bản là như nhau. Ví dụ như các từ "walks", "walking", "walked" đều là biến thể của từ "walk". Để biến đổi các từ này về dạng gốc 2 kỹ thuật thường được sử dụng là Stremming và Lemmatiaztion.

Nhóm tác giả sử dụng ngôn ngữ lập trình Python, các thư viện hỗ trợ bao gồm: Pandas, NumPy, NLTK, RE, mã nguồn được viết trên hệ thống Google Colab. Kết quả sau khi chạy các hàm chuẩn bị dữ liệu được thể hiện như trong Bảng 1 với 5 comments trước và sau khi xử lý.

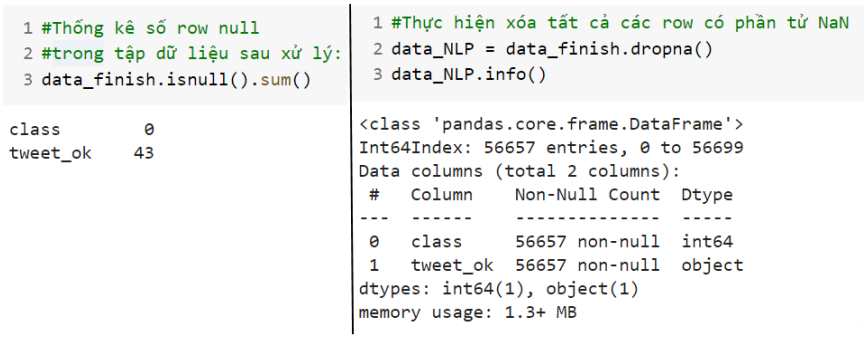
Bảng 1. Minh họa 5 comments trước và sau khi thực hiện xử lý

STT	Comments ban đầu	Comments đã xử lý
1	!!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking with hoes! 😂😂😂"	shit blow faithful somebody still fuck hoe
2	" got ya bitch tip toeing on my hardwood floors " 😂 http://t.co/cOU2WQ5L4q	get ya bitch tip toe hardwood floor
3	"@Dunderball: I'm an early bird and I'm a night owl, so I'm wise and have worms."	early bird night owl wise worm
4	"@Tmacc_GFG: “@VoiceOfDStreetz: "@Tmacc_GFG: “@tizzimarie: No slushes 😥”hoes nasty anyway fam"😴”them hoes taste like meds"🍇🍇🍼	slush hoe nasty anyway fam hoe taste like meds
5	beÃ°ÂŸÂ’Â“ Ã°ÂŸÂ’Â“@user @user @user @user @user @user @user @user @user @user	

4.2. Trích chọn đặc trưng

Như đã trình bày trong phần mở đầu, giai đoạn thứ 2 trong xây dựng mô hình học máy cho bài toán NLP đó là trích chọn đặc trưng (Feature Engineering), có rất nhiều phương pháp để thực hiện trích trọn đặc trưng chuyển đổi văn bản sang vector dữ liệu số. Tác giả sử dụng phương pháp tính TF-IDF để thực hiện việc này.

Sau khi thực hiện chuẩn bị dữ liệu như trong phần 4.1, một số comments sau xử lý chỉ còn là chuỗi rỗng. Thống kê cho thấy có 43 chuỗi comments sau khi xử lý chỉ còn lại là chuỗi rỗng (NaN). Do đó, trước khi thực hiện trích chọn đặc trưng cho toàn bộ tập comments này, các chuỗi rỗng cần phải được xử lý. Tác giả thực hiện loại bỏ các dòng comments chứa các chuỗi rỗng này ra khỏi tập dữ liệu bằng phương thức dropna() trong thư viện Pandas (Hình 3). Như vậy, tập dữ liệu ban đầu bao gồm 57 000 comments, sau khi loại bỏ thu được tập dữ liệu mới còn lại 56 657 comments.



Hình 3. Thống kê và xử lý commnets rỗng trong Pandas

Thực hiện tách tập dữ liệu thu được sau khi loại bỏ các comments rỗng thành 2 tập con sử dụng cho việc huấn luyện và kiểm thử mô hình theo tỷ lệ 80% cho tập huấn luyện và 20% cho tập kiểm thử. Tác giả sử

dụng thư viện sklearn để tự động hóa việc tách này. Kết quả sau khi thực hiện việc tách tập dữ liệu data_NLP sẽ thu được tập huấn luyện chứa 45 325 comments (chiếm 80%), tập kiểm thử chứa 11 332 comments (chiếm 20%) như Hình 4.

```
#THỰC HIỆN TÁCH TẬP DỮ LIỆU THÀNH TẬP TRAIN VÀ TEST
from sklearn import model_selection
#Tách tập dữ liệu thành Train - Test (tỷ lệ: 0.8 - 0.2)
train_x,test_x,train_y,test_y=model_selection.train_test_split(
                                                data_NLP['tweet_ok'],
                                                data_NLP['class'],
                                                test_size=0.2)

print('Tập Train (80%): ', train_x.shape)
print('Tập Test (20%): ', test_x.shape)
```

```
Tập Train (80%):  (45325,)
Tập Test (20%):  (11332,)
```

Hình 4. Phân tách tập dữ liệu thành tập Train (80%) - tập Test (20%)

Để thực hiện trích chọn đặc trưng theo phương pháp TF-IDF đã được mô tả trong phần 3, tác giả sử dụng module TfidfVectorizer trong thư viện Sklearn. Kết quả sau quá trình này sẽ thực hiện chuyển đổi dữ liệu văn bản sang vector số (Hình 5); Dữ liệu số này sẽ được sử dụng là đầu vào cho mô hình học máy Naive Bayes để thực hiện phân lớp các comments.

```
# Tính TF-IDF cho tập dữ liệu
from sklearn.feature_extraction.text import TfidfVectorizer
#Convert a collection of raw documents to a matrix of TF-IDF features.
vector = TfidfVectorizer(analyzer='word',
                         max_features=20000,
                         stop_words = 'english')

vector.fit(data_NLP['tweet_ok'])
xtrain_tfidf = vector.transform(train_x)
xtest_tfidf = vector.transform(test_x)
print('Kết quả Vector hóa tập Train sang dạng số:')
print(xtrain_tfidf.data)
print(xtrain_tfidf.shape)
print('Kết quả Vector hóa tập Test sang dạng số:')
print(xtest_tfidf.data)
print(xtest_tfidf.shape)
```

```
Kết quả Vector hóa tập Train sang dạng số:
[0.54907788 0.53160745 0.64490852 ... 0.33670852 0.13308591 0.26849291]
(45325, 20000)
Kết quả Vector hóa tập Test sang dạng số:
[0.73386444 0.51887036 0.43842506 ... 0.58960476 0.39331306 0.52415607]
(11332, 20000)
```

Hình 5. Kết quả vector hóa văn bản sang dạng số sử dụng TF-IDF

4.3. Xây dựng mô hình

Để sử dụng thuật toán phân lớp Naive Bayes, tác giả sử dụng thư viện học máy Sklearn; Sklearn cung cấp 5 giải thuật của Naive Bayes bao gồm: Gaussian Naive Bayes; Multinomial Naive Bayes; Complement Naive Bayes; Bernoulli Naive Bayes; Categorical Naive Bayes. Như so sánh thể hiện trong Hình 1, giải thuật Multinomial Naive Bayes cho bài toán phân lớp văn bản có hiệu quả cả về Score, Training time và Test time. Vì vậy, tác giả sử dụng Multinomial Naive Bayes áp dụng cho tập dữ liệu của mình. Muiltinomial Naive Bayes có 2 tham số đầu vào chính bao gồm:

- alpla: Tham số dùng để làm mịn (Laplace smoothing); giá trị thiết lập là một số thực nằm trong đoạn [0,1]; Thiết lập mặc định bằng 1.
- fit_prior: Tham số dùng để xác định có sử dụng xác suất các phần tử trước đó cho việc huấn luyện hay không; Giá trị thiết lập kiểu boolean (True|Flase), Thiết lập mặc định bằng True.

Quá trình chạy mô hình với tham số alpha =0.79, fit_prior=True cho kết quả tốt nhất. Hình 6 thể hiện việc xây dựng mô hình MultinomialNB, huấn luyện trên tập Train, và kiểm thử trên tập Test. Độ chính xác

trên khi huấn luyện mô hình đạt 93.83%, độ chính xác của mô hình khi chạy trên tập Test đạt 91.63 %.

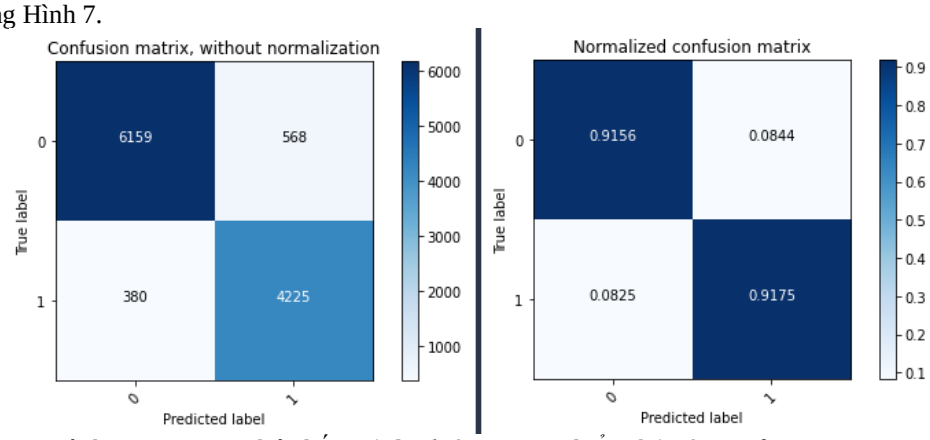
```
#Sử dụng mô hình Naive Bayes với TF-IDF
import time
from sklearn import naive_bayes as nb
from sklearn.metrics import accuracy_score

#Sử dụng thuật toán MultinomialNB
MultiNB = nb.MultinomialNB(alpha=0.79, fit_prior=True)
#Huấn luyện mô hình với tập huấn luyện Train
MultiNB.fit(xtrain_tfidf,train_y)
train_score = MultiNB.score(xtrain_tfidf, train_y)
print('Độ chính xác của mô hình trên tập Train: ', round(train_score*100,2),'%')

#Sử dụng mô hình huấn luyện dự đoán trên tập TEST
y_pred = MultiNB.predict(xtest_tfidf)
#Đánh giá độ chính xác của mô hình trên tập Test
test_sore = round(accuracy_score(test_y,y_pred)*100,2)
print('Độ chính xác của mô hình trên tập Test: ', test_sore, '%')
```

```
Độ chính xác của mô hình trên tập Train:  93.83 %
Độ chính xác của mô hình trên tập Test:  91.63 %
```

Hình 6. Xây dựng mô hình phân lớp văn bản sử dụng MultinomialNB

4.4. Đánh giá kết quả

Như ở trên cho thấy độ chính xác của mô hình phân lớp MultinomialNB trên tập dữ liệu Test bao gồm 11 332 Comments đạt 91.63%; Thời gian để huấn luyện mô hình hết 0.013969 giây, thời gian để chạy mô hình trên tập dữ liệu kiểm thử hết 0.002822 giây. Để có cái nhìn tổng quan hơn về kết quả đánh giá độ chính xác của mô hình trên tập kiểm thử, tác giả sử dụng Confusion matrix để trực quan hóa kết quả thể hiện trong Hình 7.



Hình 7. Trực quan hóa kết quả dự đoán trên tập kiểm thử với Confusion matrix

Có tất cả 10 384 comments trên tổng số 11 332 comments trong tập kiểm thử được dự đoán chính xác. Trong đó:

- 6 159 comments lớp 0 và được dự đoán đúng vào lớp 0 (91.56%);
- 4 225 comments lớp 1 và được dự đoán đúng vào lớp 1 (91.75%);
- 568 comments lớp 0 nhưng mô hình dự đoán sai sang lớp 1 (8.44%)
- 380 comments lớp 1 nhưng mô hình dự đoán sai sang lớp 0 (8.25%)

Bên cạnh đó, tác giả cũng thực hiện trích chọn đặc trưng theo phương pháp Bag of Words (BoW) và cài đặt bộ phân lớp với 2 thuật toán học máy khác để đánh giá là Support Vector Machines (SVM) và K-Nearest Neighbors (KNN).

Bảng 2. Độ chính xác của các thuật toán phân lớp trên tập kiểm thử

Phương pháp trích chọn đặc trưng	Thuật toán phân lớp		
	MultinomialNB (alpha=0.79)	SVM (C=0.5)	KNN (n_neughbors=10)
TF-IDF	91.63%	59.66%	67.94%
BoW	71.83%	59.29%	42.95%

Kết quả thể hiện độ chính xác của các mô hình trên tập kiểm thử được thống kê trong Bảng 2. Có thể thấy rằng Naïve Bayes là thuật toán tốt cho mô hình phân lớp văn bản với độ chính xác cao. Ngoài ra phương pháp trích chọn đặc trưng cũng ảnh hưởng rất lớn tới sự chính xác của mô hình.

5. Kết luận

Học máy đã được nghiên cứu, ứng dụng trong rất nhiều lĩnh vực, trong đó có lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và thu được nhiều kết quả tích cực. Trong các bài toán NLP, phân lớp văn bản là bài toán cơ bản và rất quan trọng. Trong nội dung của bài báo này, tác giả đã nghiên cứu và tìm hiểu về một trong những thuật toán rất hiệu quả cho phân lớp văn bản đó là Naive Bayes. Các kết quả trong phần thực nghiệm phân lớp văn bản với tập 56 700 comments trên Twitter thành 2 lớp 0 - Non Toxic và 1 - Toxic đã cho thấy hiệu quả của thuật toán Naive Bayes so với các thuật toán học máy khác. Kết quả thực nghiệm cũng chỉ ra rằng, độ chính xác của một mô hình phân lớp không những phụ thuộc vào thuật toán, tham số thiết lập cho thuật toán đó mà còn phụ thuộc rất lớn vào phương pháp trích chọn đặc trưng của tập dữ liệu đầu vào. Tuy nhiên, có thể khẳng định Naive Bayes Classifiers (NBC) là một lựa chọn tốt cho các bài toán phân lớp văn bản (Text classification), NBC có thời gian huấn luyện, kiểm thử mô hình rất nhanh và độ chính xác cao.

Tài liệu tham khảo

Vũ Hữu Tiệp, 2018, Bải giảng Học Máy (Machine Learning), Nhà xuất bản Khoa học kỹ thuật.145-147

Haiyi Zhang, DiLi; 2007; Naïve Bayes Text Classifier, Cranular Computing - IEEE international, GrC2007.

Wei Zhang, Feng Gao, 2011; An Improvement to Naïve Bayes for Text Classification, Procedia Engineering. 15 (2011), 2160 - 2164.

ABSTRACT

Research on Naïve Bayes classifiers to text classification; Application for Twitter's comments data set

Dang Van Nam[1]

[1]*Hanoi University of Mining and Geology*

The content of this article will research on Naïve Bayes classifiers to text classification and application for Twitter's comments data set. Naïve Bayes classifiers which are widely used for text classification in machine learning are based on the conditional probability of features belonging to a class, in which the features are selected by feature selection methods. The paper specifically presents the construction of a machine learning model with the Naïve Bayes algorithm using TF-IDF features in text classification. The experimental data set includes 56 700 comments on Twitter and that is divided into 2 classes: class 0 - Non Toxic, class 1 - Toxic. At the same time, in the content of the article, the author also compared the accuracy of Naïve Bayes with SVN algorithm and KNN on the same experimental data set.

Keywords: Machine Learning; naïve bayes; nlp; text classification; tf-idf.

Xây dựng chương trình minh họa hỗ trợ giảng dạy và học tập môn Trí tuệ nhân tạo

Đặng Hữu Nghị[1,*], Bùi Thị Vân Anh[1], Phạm Đức Hậu[1], Đặng Quốc Trung[1]
[1]*Trường Đại học Mỏ - Địa chất*

TÓM TẮT

Trí tuệ nhân tạo hay trí thông minh nhân tạo (Artificial Intelligence - AI) là một nhánh thuộc lĩnh vực khoa học máy tính. Đây là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người. Trí tuệ nhân tạo ngày càng đang tiếp cận vào đời sống của con người trong mọi lĩnh vực.

Hầu hết các trường đào tạo về Công nghệ Thông tin đều đưa môn học Trí tuệ nhân tạo vào giảng dạy. Để giúp cho việc giảng dạy và học tập môn học Trí tuệ nhân tạo được hiệu quả, chúng tôi đã xây dựng một chương trình minh họa cho một số thuật toán cơ bản trong Trí tuệ nhân tạo. Chương trình giúp cho sinh viên có cái nhìn trực quan hơn về ứng dụng của các thuật toán được học, từ đó giúp sinh viên dễ hiểu bài và nắm chắc các kiến thức hơn.

Từ khóa: Trí tuệ nhân tạo; giải thuật tìm kiếm mù; giải thuật tìm kiếm heuristic; giải thuật có đối thủ.

1. Mở đầu

Với sự phát triển vượt bậc của công nghệ, máy tính cùng các thiết bị điện tử ngày càng đóng vai trò quan trọng trong mọi hoạt động của con người. Không chỉ hỗ trợ trong công việc, các máy tính ngày nay còn là một phương tiện hữu ích phục vụ cho các sinh hoạt hàng ngày của con người như hỗ trợ trong đi lại, cập nhật tin tức, giải trí... Tuy nhiên, để máy tính thật sự trở thành một người bạn thân thiết của con người, một số trở ngại cơ bản cần phải được vượt qua. Các máy tính ngày nay có sức mạnh vượt trội so với trước, khả năng xử lý và lưu trữ khổng lồ, nhưng nó vẫn chưa giải quyết được nhiều bài toán trên thực tế như: làm sao để có thể tự suy nghĩ và hành động trong những môi trường thế giới thực phức tạp, làm sao để học và rút trích các thông tin, tri thức từ môi trường xung quanh phục vụ cho hoạt động của mình... Giải quyết các vấn đề trên không chỉ đòi hỏi sức mạnh công nghệ mà còn cần đến những hướng tiếp cận, những thuật toán thông minh hơn.

Nhiều bài toán đã được đưa vào máy tính để giải quyết, với mong muốn tận dụng được sức mạnh của máy tính để tìm lời giải nhanh và chính xác hơn. Đến nay, có nhiều phương pháp để giải một bài toán trên máy tính. Một trong các phương pháp đó là phương pháp tìm kiếm (Lê Hoài Bắc và Tô Hoài Việt, 2014).

2. Các phương pháp tìm kiếm trong trí tuệ nhân tạo

Vấn đề tìm kiếm, một cách tổng quát, có thể hiểu là tìm một đối tượng thỏa mãn một số đòi hỏi nào đó, trong một tập hợp rộng lớn các đối tượng. Chúng ta có thể kể ra rất nhiều vấn đề mà việc giải quyết nó được quy về vấn đề tìm kiếm.

Các trò chơi, chẳng hạn cờ vua, cờ carô có thể xem như vấn đề tìm kiếm. Trong số rất nhiều nước đi được phép thực hiện, ta phải tìm ra các nước đi dẫn tới tình thế kết cuộc mà ta là người thắng.

Có các kỹ thuật tìm kiếm sau (Trần Ngân Bình, Wolfgang Ertel, 2017):

- Các kỹ thuật tìm kiếm mù, trong đó chúng ta không có hiểu biết gì về các đối tượng để định hướng tìm kiếm mà chỉ đơn thuần là xem xét theo một hệ thống nào đó tất cả các đối tượng để phát hiện ra đối tượng cần tìm.
  - Tìm kiếm theo bề rộng (breadth-first search)
  - Tìm kiếm theo độ sâu (depth-first search)
- Các kỹ thuật tìm kiếm theo kinh nghiệm (tìm kiếm heuristic) trong đó chúng ta dựa vào kinh nghiệm và sự hiểu biết của chúng ta về vấn đề cần giải quyết để xây dựng nên hàm đánh giá hướng dẫn sự tìm kiếm. Các chiến lược tìm kiếm kinh nghiệm quan trọng nhất là:
  - Tìm kiếm tốt nhất - đầu tiên (best-first search)

[*] *Tác giả liên hệ*
Email: danghuunghi@humg.edu.vn

- Tìm kiếm leo đồi (hill-climbing search)

- Các kỹ thuật tìm kiếm tối ưu gồm các kỹ thuật tìm đường đi ngắn nhất trong không gian trạng thái như:
  - Thuật toán A*
  - Thuật toán nhánh và cận.
  - …
- Các phương pháp tìm kiếm có đối thủ, tức là các chiến lược tìm kiếm nước đi trong các trò chơi hai người, chẳng hạn cờ vua, cờ tướng, cờ carô. Các phương pháp tìm kiếm có đối thủ sử dụng chiến lược tìm kiếm nước đi Minimax.

Một khi chúng ta muốn giải quyết một vấn đề nào đó bằng tìm kiếm, đầu tiên ta phải xác định không gian tìm kiếm. Không gian tìm kiếm bao gồm tất cả các đối tượng mà ta cần quan tâm tìm kiếm. Nó có thể là không gian liên tục, chẳng hạn không gian các véctơ thực n chiều; nó cũng có thể là không gian các đối tượng rời rạc.

Ví dụ, trong trò chơi cờ vua, mỗi cách bố trí các quân trên bàn cờ là một trạng thái. Trạng thái ban đầu là sự sắp xếp các quân lúc bắt đầu cuộc chơi. Mỗi nước đi hợp lệ là một toán tử, nó biến đổi một trạng thái trên bàn cờ thành một trạng thái khác.

Khi giải quyết một bài toán bằng phương pháp tìm kiếm (gọi tắt là bài toán tìm kiếm) trước hết phải biểu diễn được bài toán đó. Một bài toán thường gồm có vấn đề (yêu cầu) và lời giải cho bài toán. Phương pháp tìm kiếm được sử dụng để tìm ra lời giải trong không gian tìm kiếm.

Biểu diễn một bài toán tìm kiếm thường bao gồm năm thành phần:

- Không gian trạng thái: Cho biết tất cả các trạng thái có thể có của bài toán.
- Trạng thái ban đầu. Nơi bắt đầu việc tìm kiếm.
- Hàm trạng thái con. Phát sinh trạng thái kế tiếp từ một trạng thái cụ thể x. Hàm trạng thái con, cùng với trạng thái ban đầu, tạo nên không gian trạng thái của bài toán. Không gian này bao gồm các trạng thái mà ta sẽ xem xét đến trong quá trình tìm kiếm.
- Một tập các trạng thái đích (kết thúc). Các trạng thái đích này đều thuộc không gian trạng thái. Tùy vào bài toán mà có một hay nhiều trạng thái đích khác nhau.
- Hàm chi phí đường đi. Dùng để tính chi phí cho lời giải của bài toán. Với cách biểu diễn này ta có thể chuyển bài toán tìm lời giải về bài toán tìm đường đi từ trạng thái ban đầu đến trạng thái đích.

Giải quyết bài toán bằng phương pháp tìm kiếm được áp dụng trên nhiều lĩnh vực khác nhau. Ở đây, ta xét một bài toán tìm kiếm thông dụng, đó là bài toán trò chơi.

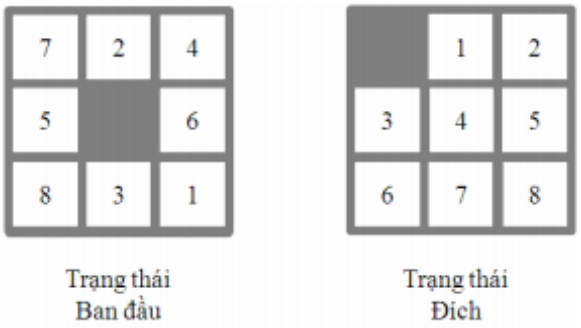
Bài toán trò chơi minh họa cho những phương pháp giải quyết vấn đề khác nhau. Thông thường loại bài toán này luôn có mô tả chính xác, cụ thể. Có điểm chung thống nhất như vậy giúp dễ dàng so sánh hiệu quả của các thuật toán khác nhau.

3. Xây dựng một số ví dụ minh họa

Để giúp cho việc giảng dạy và học tập môn học Trí tuệ nhân tạo được hiệu quả, chúng tôi đã xây dựng một chương trình minh họa một số thuật toán cơ bản trong Trí tuệ nhân tạo thông qua các trò chơi sau

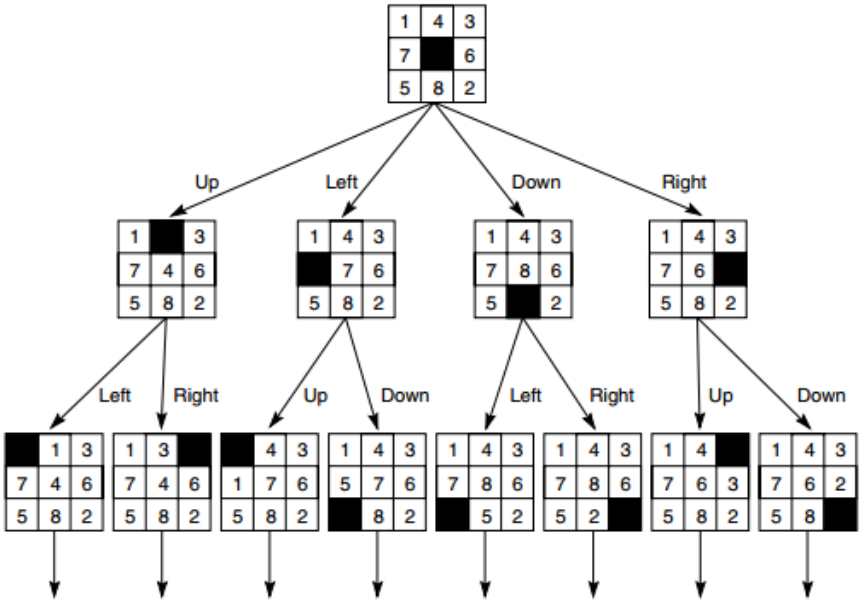
3.1 Bài toán trò chơi 8-puzzle

Bài toán 8-puzzle (còn gọi là Taci) thường được sử dụng để thử nghiệm các thuật toán tìm kiếm. Trong bài toán này, chúng ta có một bàn cờ kích thước 3×3 trong đó 8 quân cờ được đánh số từ 1 đến 8 và đặt ở 8 ô vuông, ô còn lại để trống. Các quân cờ có thể được đẩy vào ô trống kế cận nó. Mục tiêu của trò chơi này là đẩy các quân cờ sao cho từ cách sắp xếp ban đầu (trạng thái đầu) đưa về được cách sắp xếp mong muốn (trạng thái đích) như trong Hình 1.



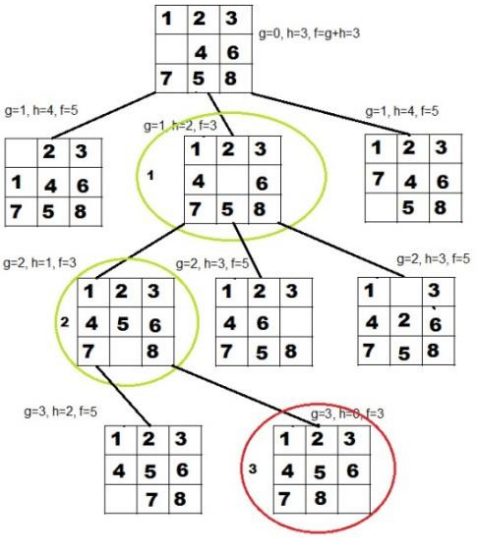
Hình 1. Một dạng của bài toán 8-puzzle

Không gian trạng thái của bài toán như sau:



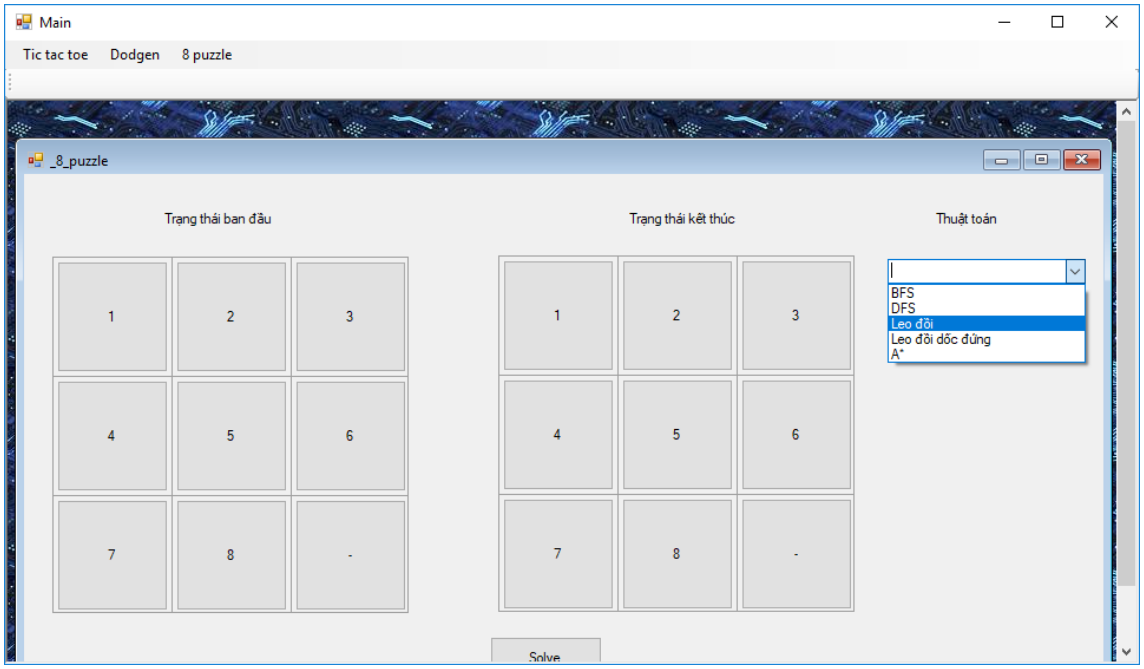
Hình 2. Không gian trạng thái của trò chơi 8-puzzle

Chúng tôi sử dụng thuật toán A* cho trò chơi 8-puzzle.Trong thuật toán A* hàm độ tốt dùng để ước lượng quãng đường từ nút khởi đầu đến nút đích qua nút hiện tại f = g+h. Trong đó g là chiều sâu từ nút gốc đến nút hiện tại và h là ước lượng quãng đường từ nút hiện tại đến nút đích còn gọi là hàm tính chi phí. Việc xác hàm tính chi phí *h* là một trong các yếu tố quan trọng. Hàm tính chi phí *h* của bài toán 8-puzzle là số lượng các ô sai vị trí (Khoảng cách Manhattan). Ví dụ như hình sau:



Hình 3. Ví dụ về hàm tính độ tốt trong bài toán 8-puzzle

Kết quả cài đặt trò chơi 8-puzzle như sau:



Hình 4. Giao diện game 8-puzzle

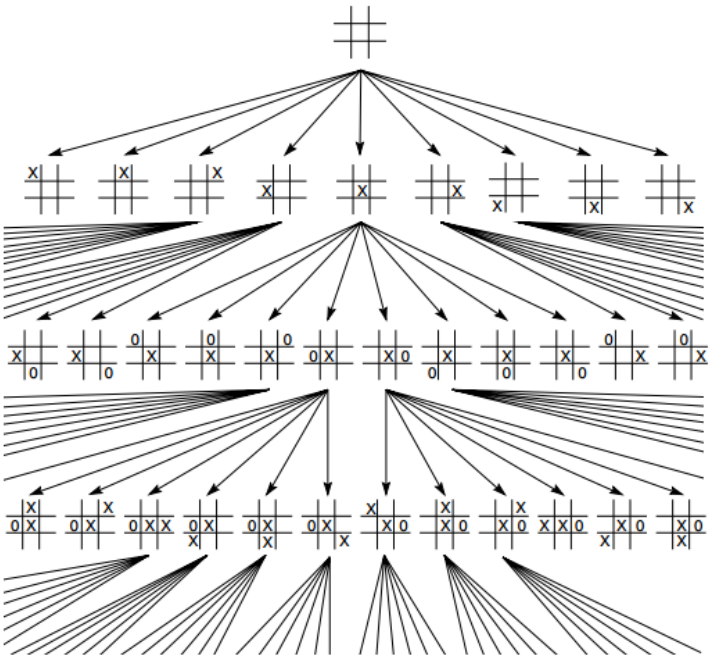
3.2 Trò chơi Tic-Tac-Toe

Chúng tôi cũng xây dựng trò chơi tictactoe và dodgem để minh họa việc áp dụng thuật toán tìm kiếm có đối thủ minimax.

Bắt đầu ván đấu bằng bàn cờ trống, đấu thủ thứ nhất có thể đặt ký hiệu nước đi X vào bất cứ ô nào trong 9 ô trống của bàn cờ.

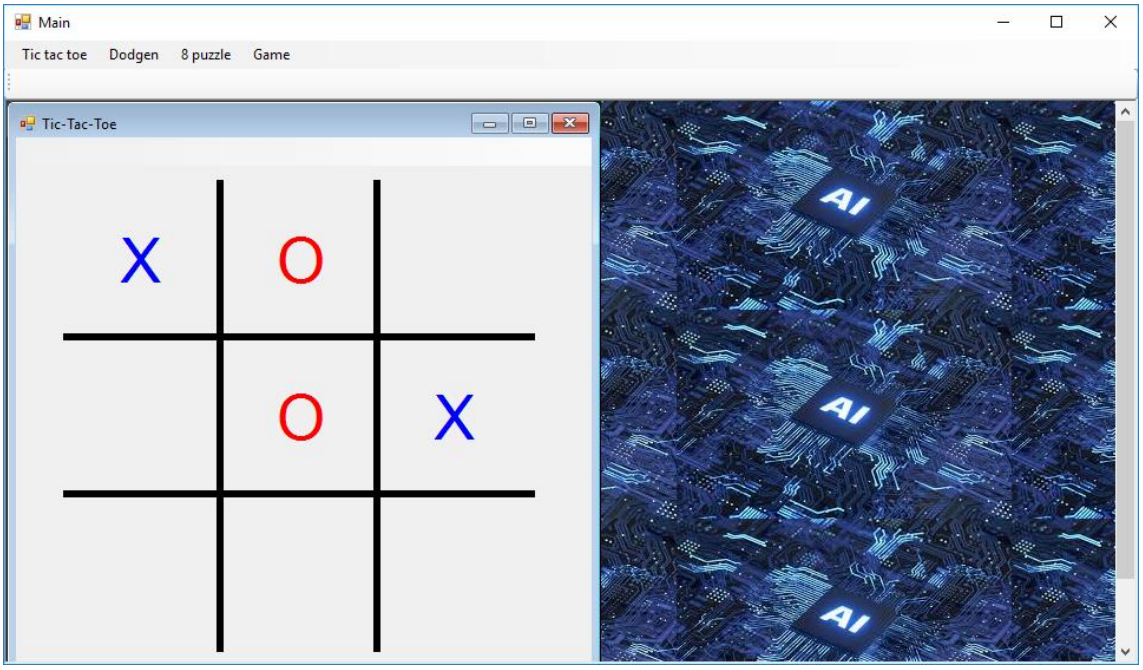
Mỗi trong số những nước đi này tạo ra một bàn cờ khác cho phép đấu thủ thứ hai đến lượt mình đi sẽ có thể chọn 8 cách đặt ký hiệu nước đi O của mình vào... Và sẽ cứ luân phiên như thế.

Không gian trạng thái của bài toán như sau:

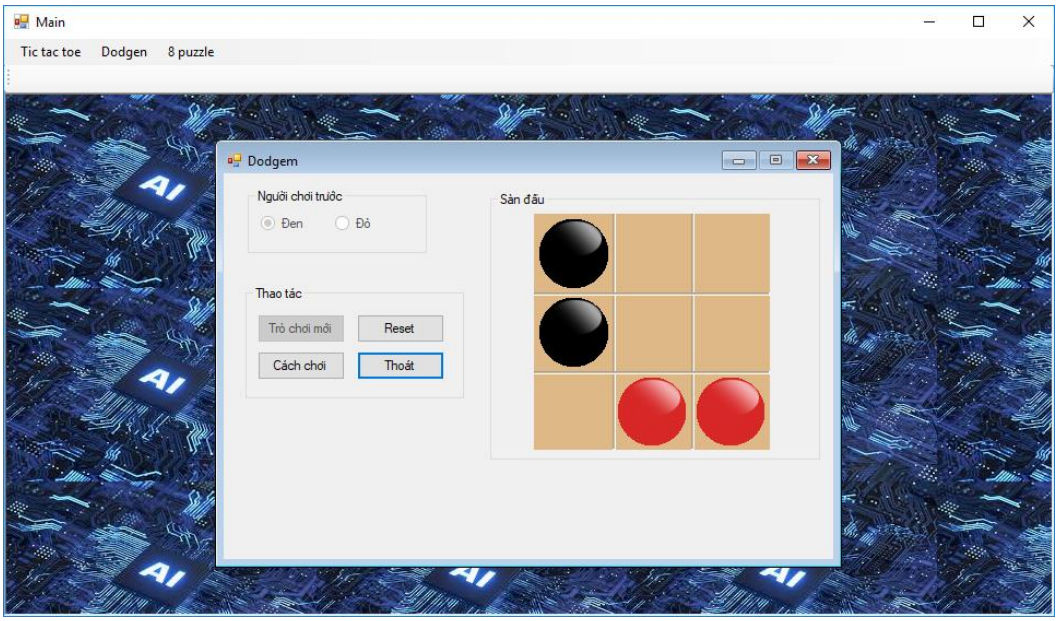


Hình 5. Không gian trạng thái của trò chơi Tic-Tac-Toe

Kết quả cài đặt các trò chơi tictactoe và dodgem như sau:



Hình 6. Giao diện game Tic-Tac-Toe



Hình 7. Giao diện game Dodgen

4. Kết luận

Trí tuệ nhân tạo và học máy là những lĩnh vực rất quan trọng của hiện tại và tương lai. Muốn thực sự nắm được các kỹ năng kiến thức của trí tuệ nhân tạo để có thể dùng trong thực tế, thì không có cách nào khác là phải thực hành. Những ứng dụng thực tế cần được đưa vào giảng dạy và minh hoạ để sinh viên dễ dàng hình dung và cọ xát thực tế.

Trong bài báo này chúng tôi đã xây dựng một số ứng dụng minh họa cho các giải thuật trí tuệ nhân tạo cổ điển. Trong thời gian tới chúng tôi sẽ tiếp tục phát triển để có nhiều ví dụ minh họa hơn cho các giải thuật về máy học

Lời cảm ơn

Bài báo này là kết quả của Đề tài mã số CT.2019.01.02.

Tài liệu tham khảo

Lê Hoài Bắc - Tô Hoài Việt, cơ sở trí tuệ nhân tạo, nhà xuất bản khoa học và kỹ thuật, 2014.

Trần Ngân Bình, Trí tuệ nhân tạo, Phiên bản trực tuyến: https://voer.edu.vn/c/tri-tue-nhan-tao/764b3239

Wolfgang Ertel, Introduction to Artificial Intelligence, Springer, 2017

ABSTRACT

Programming a program to support teaching and learning Artificial Intelligence

Dang Huu Nghi[1], Bui Thi Van Anh[1], Pham Duc Hau[1], Dang Quoc Trung[1]

[1]*HaNoi University of Mining and Geology*

Artificial Intelligence (AI) is a branch of computer science. This is a human-programmed intelligence with the goal of helping computers automate intelligent behaviors like humans. Artificial intelligence is increasingly employed in the human life in all fields.

Most universities teaching Information Technology have a subject called Artificial Intelligence. In order to make the teaching and learning of the subjecteffectively, we have built a program that illustrates some of the basic algorithms in Artificial Intelligence. The program helps students to have a more intuitive view of the application of the algorithms, thereby making it easier for students to understand the lessons and gain the knowledge.

Keyword : Artificial intelligence; blind search algorithms; heuristic search algorithms; minimax algorithm.

Nghiên cứu mạng học sâu sử dụng Keras trong nhận dạng hình ảnh

Phạm Đình Tân[1,*], Trần Thị Thu Thúy[1], Diêm Công Hoàng[1]
[1]*Trường Đại học Mỏ - Địa chất*

TÓM TẮT

Lĩnh vực thị giác máy tính nghiên cứu các kỹ thuật giúp máy tính tự động nhận dạng và đọc hiểu thông tin từ hình ảnh. Thị giác máy tính có rất nhiều ứng dụng trong thực tế như nghiên cứu giải quyết bài toán xe tự lái, quản lý chất lượng sản phẩm dựa trên hình ảnh cũng như các lĩnh vực gaming, thực tại ảo. Trong những năm gần đây, lĩnh vực thị giác máy tính đã có những bước phát triển mạnh mẽ dựa trên sự bùng nổ của kỹ thuật học sâu. Kỹ thuật học sâu sử dụng kiến trúc mạng nơ-ron nhiều lớp để trích chọn đặc trưng một cách tự động dựa trên dữ liệu. Mô hình mạng nơ-ron nhiều lớp (được xây dựng dựa trên mô hình mạng nơ-ron trong não bộ của con người) là tập hợp các nút mạng được kết nối và thông tin được lan truyền từ nút mạng này sang nút mạng khác. Rất nhiều kiến trúc mạng học sâu khác nhau đã được đề xuất cho bài toán nhận dạng hình ảnh. Mạng học sâu DenseNet là kiến trúc mạng có hiệu năng cao trong nhận dạng hình ảnh. Keras là thư viện mã nguồn mở dùng để xây dựng và huấn luyện các mạng học sâu trên ngôn ngữ lập trình Python. Trong bài báo này, nhóm tác giả nghiên cứu sử dụng thư viện Keras để xây dựng và huấn luyện mạng học sâu sử dụng kiến trúc DenseNet nhằm giải quyết bài toán phân lớp hình ảnh. Hiệu năng của mạng học sâu được đánh giá trên bộ dữ liệu hình ảnh CIFAR-10.

Từ khóa: Mạng nơ-ron; thị giác máy tính; kỹ thuật học sâu; keras; densenet.

1. Đặt vấn đề

Trong những năm gần đây, kỹ thuật học sâu (Deep Learning) đã trở thành một công cụ hiệu quả trong lĩnh vực thị giác máy tính. Những tiến bộ về tốc độ tính toán của phần cứng máy tính như bộ xử lý đồ họa (GPU) và các kiến trúc mạng nhiều lớp đã cho phép huấn luyện các mạng học sâu với hàng triệu tham số cùng hàng trăm lớp ẩn. Lý thuyết về mạng nơ-ron tích chập (CNN) được phát triển từ lâu nhưng gần đây mới được triển khai rộng rãi nhờ các tiến bộ về tốc độ xử lý song song của GPU, vốn được phát triển cho các ứng dụng gaming, đồ họa. Triển khai các thuật toán học sâu trên GPU giúp rút ngắn thời gian huấn luyện/ kiểm thử mô hình.

Rất nhiều kiến trúc CNN đã được đề xuất, một trong những kiến trúc CNN nổi bật là DenseNet (Huang và nnk, 2017]. Trong kiến trúc DenseNet, một kiến trúc mạng học sâu trong đó tất cả các tầng đều có kết nối trực tiếp đến các tầng khác được sử dụng để đảm bảo luồng thông tin tối đa giữa các tầng trong mạng. Để duy trì bản chất chuyển tiếp, mỗi lớp có được các đầu vào bổ sung từ tất cả các lớp trước đó và truyền tất cả các bản đồ đặc trưng đến các tầng phía sau. Kiến trúc mạng này yêu cầu ít tham số hơn các mạng tích chập truyền thống, vì không cần phải tìm lại các bản đồ đặc trưng dư thừa. Kiến trúc mạng này có thể được xem như là thuật toán có trạng thái được truyền từ tầng này sang tầng khác. Mỗi tầng đọc trạng thái từ tầng trước của nó và ghi vào tầng tiếp theo. Nó thay đổi trạng thái nhưng cũng truyền thông tin cần được duy trì. Bên cạnh việc sử dụng ít tham số, kiến trúc mạng này còn giúp truyền hiệu quả luồng dữ liệu và gradient qua các tầng mạng, giúp cho việc huấn luyện mạng trở nên dễ dàng hơn. Mỗi tầng đều có quyền truy cập trực tiếp vào các gradient từ các hàm mục tiêu và các dữ liệu đầu vào giúp huấn luyện sâu hơn kiến trúc mạng. Hơn nữa, các kết nối dày đặc cũng có chức năng dàn đều, tránh bị overfit khi sử dụng các tập dữ liệu huấn luyện nhỏ.

Keras là một thư viện học sâu được sử dụng rộng rãi bởi các hãng công nghệ lớn như Google, Netflix và NVIDIA. TensorFlow là thư viện mã nguồn mở rất thông dụng của Google vì TensorFlow cung cấp nhiều công cụ cho việc triển khai và bảo trì ứng dụng, tìm lỗi và biểu diễn trực quan, cũng như chạy các mô hình trên các trình duyệt và các thiết bị nhúng. Google công bố chọn Keras làm API bậc cao chính thức cho thư viện TensorFlow vì Keras rất dễ sử dụng và có thể triển khai nhanh các mô hình học sâu. Các bước triển khai bao gồm huấn luyện (train), kiểm thử (validation) và đánh giá (test). Việc huấn luyện mạng trở nên đơn giản với Keras khi các đầu vào yêu cầu sẽ là dữ liệu, số epoch và hàm mục tiêu. Kiến trúc chung

* *Tác giả liên hệ*
Email: phamdinhtan@humg.edu.vn

của mạng học sâu là kiến trúc mạng có nhiều lớp, trong đó mỗi lớp là một phép toán ma trận có chức năng riêng. Mỗi lớp thực hiện tính toán dựa trên dữ liệu đầu vào từ lớp trước và chuyển kết quả đến lớp kế tiếp. Trong bài báo này, nhóm tác giả nghiên cứu triển khai kiến trúc mạng học sâu DenseNet sử dụng Keras. Hiệu năng mạng được đánh giá sử dụng bộ dữ liệu CIFAR-10.

2. Các nghiên cứu liên quan

Một cấu trúc mạng nơ-ron tương tự có kiến trúc mạng lưới dày đặc đã được đề xuất từ những năm 1980 (He và nnk., 2016). Công việc tiên phong của họ tập trung vào mạng perceptron nhiều lớp trong đó các lớp được huấn luyện tuần tự. Gần đây, mạng nơ-ron nhiều lớp được kết nối đầy đủ được huấn luyện theo kiểu từng lô. Mặc kiến trúc này hiệu quả trên các tập dữ liệu nhỏ, kiến trúc này chỉ phù hợp cho các mạng có vài trăm tham số. Theo Lin và nnk (2013), kiến trúc mạng nơ-ron Network-in-Network (NiN) được đề xuất trong đó sử dụng perceptron nhiều lớp trong các bộ lọc của lớp tích chập nhằm trích chọn các đặc trưng phức tạp. Các kiến trúc CNN thông thường sử dụng các tầng tích chập (bao gồm hàm lọc tuyến tính và hàm kích hoạt phi tuyến) và tầng gộp (pooling). Kết quả đầu ra được gọi là các bản đồ đặc trưng. Trong NiN, các hàm vi mô được sử dụng để xấp xỉ các hàm phi tuyến. Thay vì sử dụng các lớp được kết nối đầy đủ truyền thống để phân loại trong CNN, kiến trúc NiN trực tiếp xuất ra mức trung bình trong không gian của các bản đồ đối tượng từ lớp cuối cùng dưới dạng độ tin cậy của danh mục thông qua lớp tổng hợp trung bình toàn cục và sau đó vectơ kết quả được đưa vào lớp softmax. Trong CNN truyền thống, rất khó để diễn giải cách thông tin cấp danh mục từ lớp chi phí mục tiêu được chuyển trở lại lớp tích chập trước đó do kết nối đầy đủ các lớp hoạt động như một hộp đen ở giữa. Ngược lại, tổng hợp trung bình toàn cầu có ý nghĩa hơn và có thể hiểu được vì nó thực thi sự tương ứng giữa các bản đồ đối tượng và các danh mục, được tạo ra có thể bằng mô hình cục bộ mạnh hơn sử dụng kiến trúc mạng vi mô. Hơn nữa, kết nối đầy đủ các lớp dễ bị trang bị quá mức và phụ thuộc nhiều vào dropout, trong khi lớp gộp giúp ngăn chặn việc overfitting cho toàn bộ kiến trúc. Kiến trúc chung của các mạng CNN là các lớp chập, hàm gộp cực đại và các lớp kết nối đầy đủ. Trong mỗi lớp này đều có thể có các hàm kích hoạt phi tuyến. Mối mạng thường có số lượng tham số lớn và sử dụng hàm phổ quát Dropout. Theo Springenberg và nnk (2014), một kiến trúc mạng chỉ sử dụng các lớp tích chập được đề xuất, trong đó lớp gộp cực đại được thay thế bởi một lớp tích chập. Theo Lee và nnk (2015), kỹ thuật giám sát mức sâu được sử dụng trong đó mỗi lớp ẩn đều được gắn kèm một bộ phân lớp, giúp cho các lớp trung gian có thể học được các đặc trưng riêng biệt. Mỗi lớp sẽ học có giám sát từ hàm mục tiêu sử dụng các kết nối trực tiếp đến gradient từ hàm mục tiêu. Các bản đồ đặc trưng có độ phân biệt cao sẽ giúp đạt được kết quả phân lớp tốt hơn. Bằng cách sử dụng phản hồi về chất lượng đặc trưng ở mỗi tầng sẽ giúp cập nhật các tham số để đạt được các bản đồ đặc trưng có độ phân tách cao hơn. Đây là cách thức học có giám sát cho từng lớp mạng, cho kết quả hội tụ nhanh hơn so với kỹ thuật lan truyền ngược từ tầng cuối cùng. Các kiến trúc mạng học sâu sử dụng việc nối tầng các biến đổi phi tuyến làm hạn chế quá trình truyền gradient. Theo Srivastava và nnk (2015), một kiến trúc mạng được đề xuất sử dụng cơ chế ngưỡng thích nghi giúp cho thông tin lưu thông qua nhiều lớp mà không bị suy hao. Khi số lớp trong các kiến trúc CNN ngày một tăng, một vấn đề mới phát sinh là khi thông tin về đầu vào hoặc gradient đi qua nhiều lớp, nó có thể biến mất trước khi đến lớp cuối cùng. Nhiều nghiên cứu gần đây đề cập đến điều này. Một số mạng sử dụng cách nối tắt từ lớp này sang lớp kế tiếp. Mạng ResNets được đề xuất (He và nnk, 2016) sử dụng kỹ thuật loại bỏ bớt số tầng một cách ngẫu nhiên trong quá trình huấn luyện để dữ liệu và gradient được lưu thông tốt hơn. Mặc dù những cách tiếp cận khác nhau khác nhau trong cấu trúc liên kết mạng và quy trình huấn luyện, tất cả đều có chung một đặc điểm chính: họ tạo các đường dẫn ngắn từ các lớp đầu đến các lớp sau. Các mạng CNN ra đời sau thường có nhiều tầng hơn các mạng trước. Tuy nhiên, khi dữ liệu hoặc gradient truyền qua quá nhiều tầng thì sẽ có thể bị biến mất mà không thể truyền được đến tầng cuối cùng. Một số nghiên cứu gần đây đã tập trung giải quyết vấn đề này. Ví dụ như mạng ResNet giải quyết vấn đề này bằng cách thêm một nối tắt từ mỗi tầng đến tầng tiếp theo.

Để đảm bảo luồng thông tin tối đa giữa các lớp trong mạng, kiến trúc DenseNet được đề xuất bởi Huang và nnk (2017) với các lớp được kết nối trực tiếp với nhau. Để duy trì tính chất chuyển tiếp một chiều, đầu vào mỗi tầng sẽ là từ tất cả các tầng phía trước và đầu ra sẽ được chuyển đến tất cả các tầng phía sau. Trái ngược với ResNets, DenseNet không thực hiện phép tính tổng mà thực hiện ghép nối các đặc trưng dữ liệu. Kiến trúc DenseNet sử dụng ít tham số hơn so với các mạng nơ-ron tích chập truyền thống, vì không cần phải học lại các bản đồ đặc trưng dư thừa. Các kiến trúc chuyển tiếp truyền thống có thể được xem như các thuật toán với trạng thái, được truyền từ lớp này sang lớp khác. Mỗi tầng sẽ nhận thông tin từ tầng trước và truyền thông tin sang tầng tiếp theo. ResNets duy trì thông tin bằng cách thêm vào một lối tắt. Các biến thể gần đây của ResNets cho thấy nhiều tầng có rất ít ảnh hưởng và có thể loại bỏ một cách ngẫu nhiên trong quá trình huấn luyện. Điều này làm cho số lượng tham số của ResNets lớn hơn đáng kể vì mỗi lớp có các trọng số riêng. Kiến trúc DenseNet phân biệt rõ ràng giữa thông tin được thêm vào mạng và thông tin được duy trì. Các lớp DenseNet rất hẹp chỉ thêm một tập hợp nhỏ các bản đồ đặc trưng vào mạng và giữ nguyên

các bản đồ đặc trưng còn lại. Bộ phân loại sẽ đưa ra quyết định cuối cùng dựa trên tất cả các bản đồ đặc trưng trong mạng. Ngoài việc sử dụng ít tham số, một lợi thế lớn của DenseNet là luồng thông tin và gradient có thể lưu thông dễ dàng trên toàn mạng giúp quá trình huấn luyện dễ dàng hơn. Mỗi tầng có quyền truy cập trực tiếp vào các gradient từ hàm mục tiêu và dữ liệu đầu vào, giúp cho quá trình học có giám sát hiệu quả hơn. Ngoài ra các kết nối dày trong DenseNet cũng có tác dụng phổ quát, giúp tránh bị overfit trên các bộ dữ liệu cỡ nhỏ. Thay vì tập trung khai thác việc biểu diễn dựa trên các kiến trúc mạng nhiều tầng, DenseNet khai thác tiềm năng của mạng thông qua việc tái sử dụng đặc trưng, mang lại các mô hình rút gọn, dễ huấn luyện và không sử dụng quá nhiều tham số. Việc nối tầng các bản đồ đặc trưng được học bởi các tầng khác nhau làm tăng sự thay đổi trong đầu vào của các lớp tiếp theo và nâng cao hiệu quả.

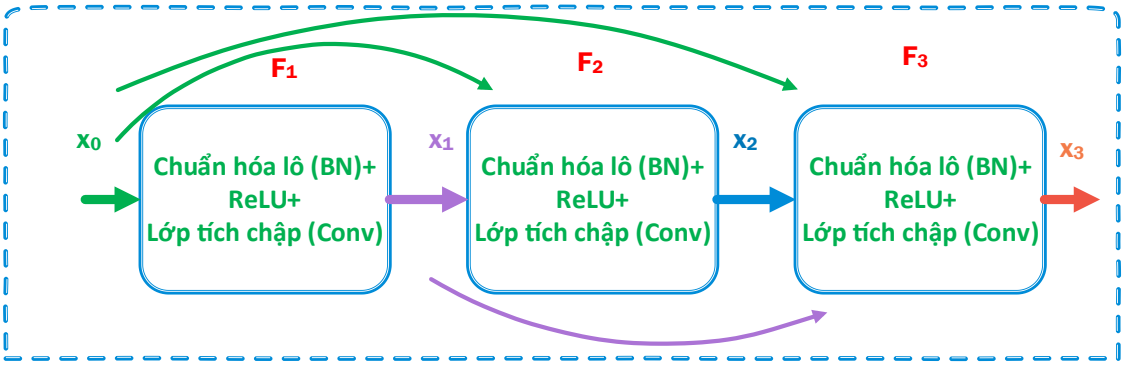
3. Kiến trúc mạng DenseNet

Trong phạm vi bài báo này, nhóm tác giả nghiên cứu triển khai kiến trúc mạng DenseNet (Huang. và nnk, 2017) sử dụng thư viện Keras. Hiệu năng của kiến trúc mạng DenseNet được đánh giá sử dụng bộ dữ liệu hình ảnh CIFAR-10.

Trong kiến trúc mạng DenseNet, gọi F_i là phép toán phi tuyến tổng hợp của các phép toán chuẩn hóa nhóm (BN), phép tính phi tuyến ReLU và phép tính tích chập (Conv) ở lớp thứ i. Đầu ra của lớp thứ i sẽ được biểu diễn bởi công thức:

$$x_i = F_i([x_0, x_1, \cdots, x_{i-1}]) \quad (1)$$

Điều này nghĩa là khối Dense luôn duy trì kết nối trực tiếp từ đầu ra đến tất cả các lớp phía sau như minh họa trong Hình 1.

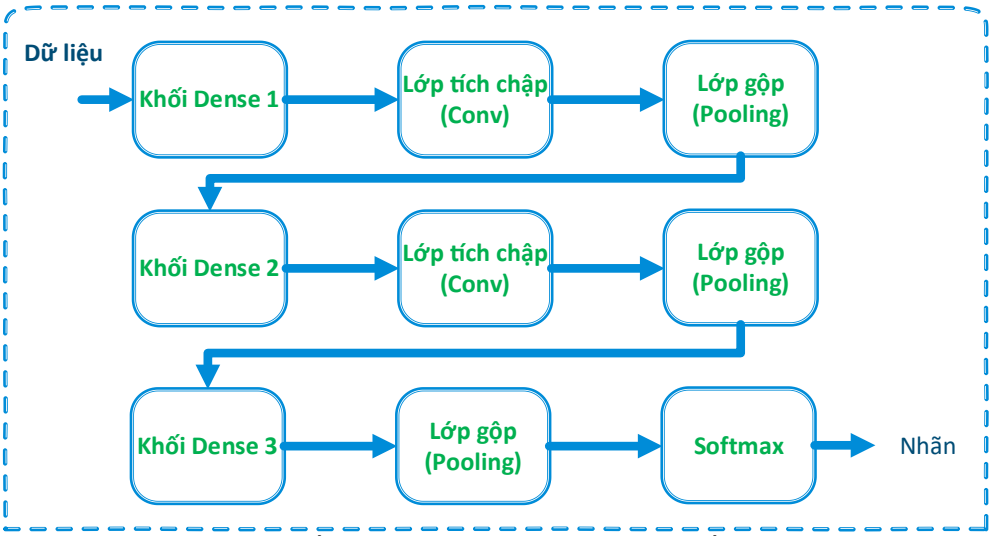


Hình 1. Sơ đồ khối Dense

Kiến trúc mạng DenseNet được mô tả như trong Hình 2 sử dụng 3 khối Dense. Giữa các khối Dense là các lớp chuyển tiếp. Các lớp chuyển tiếp bao gồm lớp tích chập và lớp gộp (pooling) nhằm giảm số mẫu để giữ nguyên kích thước của bản đồ đặc trưng. Các khối chức năng chính bao gồm: khối tích chập (Conv), khối chuẩn hóa lô (BN), hàm kích hoạt ReLU và khối gộp (pooling).

Lớp gộp (pooling): Các mạng CNN thường thực hiện tích chập ở các lớp đầu tiên của mạng. Trong bước phân lớp, các bản đồ đặc trưng của lớp cuối cùng được biến đổi thành dạng vector hóa và được đưa vào lớp kết nối đầy đủ rồi đi qua lớp softmax. Cấu trúc này kết nối các cấu trúc tính tích chập với các bộ phân lớp. Các lớp tích chập được sử dụng cho mục đích trích chọn đặc trưng, rồi đầu ra được đưa vào bộ phân lớp. Các lớp kết nối đầy đủ rất dễ bị overfit, nghĩa là không có khả năng phổ quát hóa đối với các dữ liệu mới. Hàm phổ quát Dropout có thể được sử dụng bằng cách xóa ngẫu nhiên một phần hàm kích hoạt về không. Điều này giúp tăng tính tổng quát của mạng với dữ liệu mới và giảm bớt overfit.

Hàm kích hoạt (activation function) giúp mạng có thể học các hàm phi tuyến. Hàm kích hoạt sẽ xác định việc perceptron được kích hoạt hay không. Trong quá trình huấn luyện, các hàm kích hoạt rất quan trọng trong việc điều chỉnh gradient. Hầu hết các hàm kích hoạt là các hàm số liên tục và khả vi, ngoại trừ hàm RELU. Tuy nhiên, hàm RELU hoạt động như một bộ lọc chỉ cho các giá trị dương đi qua nên rất đơn giản và thông dụng.

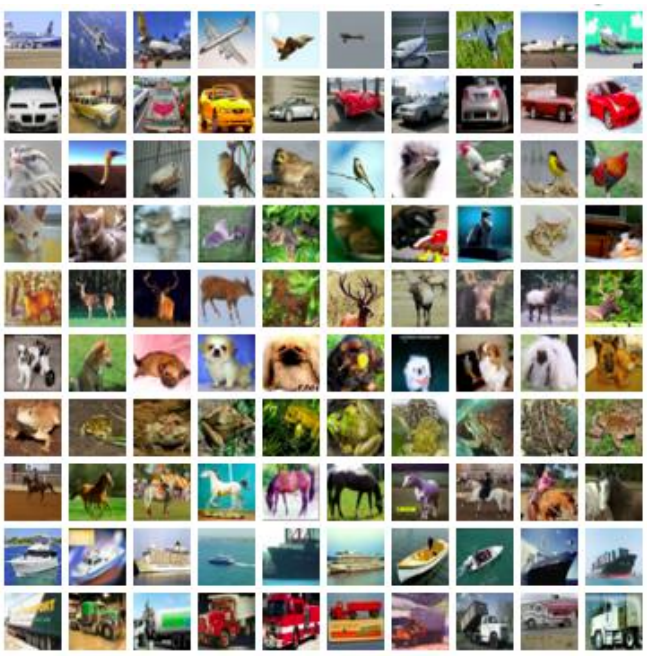


Hình 2. Kiến trúc mạng DenseNet sử dụng 3 khối Dense

4. Kết quả

4.1. Bộ dữ liệu hình ảnh CIFAR-10

CIFAR-10 (Krizhevsky và Hinton, 2009) là cơ sở dữ liệu ảnh màu có kích thước mỗi ảnh là 32x32 điểm ảnh được sử dụng rộng rãi trong việc đánh giá hiệu năng thuật toán trong lĩnh vực thị giác máy tính. CIFAR là từ viết tắt của cụm từ Canadian Institute for Advanced Research (Viện nghiên cứu cao cấp của Canada). Dữ liệu ảnh CIFAR-10 được chia thành 10 lớp bao gồm máy bay, ô tô, chim gà, mèo, hươu, chó, ếch, ngựa, tàu thuyền và xe tải. Tập huấn luyện (training set) gồm 50000 ảnh và tập kiểm tra (test set) gồm 10000 ảnh. Trong số 50000 ảnh của tập huấn luyện, 5000 ảnh được lấy ra để làm tập kiểm thử (validation set).



Hình 3. Ảnh minh họa các lớp đối tượng trong bộ dữ liệu CIFAR-10

4.2. Quá trình huấn luyện mạng

Tất cả các mạng được huấn luyện bằng cách sử dụng thuật toán gradient ngẫu nhiên (SGD). Việc huấn luyện được thực hiện theo lô 64 ảnh, được chia thành 300 epoch. Giá trị tốc độ học được thiết lập là 0,1. Quá trình huấn luyện mạng được thực hiện trên máy chủ sử dụng CPU Intel i7-8700, 32GB RAM và GPU Nvidia GeForce GTX 1080Ti. Độ chính xác của kiến trúc mạng DenseNet được đánh giá trên cơ sở dữ liệu

ảnh CIFAR-10 là 93,66% như trong Bảng 1. Kiến trúc mạng DenseNet thể hiện hiệu năng vượt trội trong phân lớp ảnh so với các kỹ thuật học sâu khác như ResNet và All-CNN.

Bảng 1. Độ chính xác nhận dạng trên bộ dữ liệu hình ảnh CIFAR-10

TT	Kỹ thuật học sâu	Độ chính xác phân lớp (%)
1	NiN (Lin và nnk, 2013)	91,19
2	All-CNN (Springenberg và nnk, 2014)	92,75
3	Deeply Supervised Net (Lee và nnk, 2015)	91,78
4	Highway Network (Srivastava và nnk, 2015)	92,4
5	ResNet (He và nnk, 2016)	93,39
6	DenseNet sử dụng Keras	93,66

5. Kết luận

Bài báo đã nghiên cứu về triển khai mạng học sâu sử dụng thư viện Keras để giải quyết bài toán nhận dạng hình ảnh. Keras giúp đơn giản hóa việc triển khai các kiến trúc mạng học sâu phức tạp khi sử dụng kết hợp với thư viện TensorFlow của Google. Kiến trúc mạng học sâu được sử dụng là kiến trúc DenseNet, với đầu ra mỗi lớp luôn tồn tại một kết nối trực tiếp đến các lớp phía sau. Kiến trúc DenseNet cho độ chính xác nhận dạng trên bộ dữ liệu hình ảnh CIFAR-10 lên đến 93,66%.

Tài liệu tham khảo

Nguyễn Thanh Tuấn. 2020. Deep Learning cơ bản. E-book, tái bản lần thứ 2.

Atienza, R. 2020. Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more. Packt Publishing Ltd.

He, K., Zhang, X., Ren, S., & Sun, J. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. 2016. Deep networks with stochastic depth. In European conference on computer vision (pp. 646-661). Springer, Cham.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 4700-4708).

Krizhevsky, A., & Hinton, G. 2009. Learning multiple layers of features from tiny images.

Larsson, G., Maire, M., & Shakhnarovich, G. 2016. Fractalnet: Ultra-deep neural networks without residuals. arXiv preprint arXiv:1605.07648.

Lee, C. Y., Xie, S., Gallagher, P., Zhang, Z., & Tu, Z. 2015. Deeply-supervised nets. In Artificial Intelligence and Statistics (pp. 562-570).

Lin, M., Chen, Q., & Yan, S. 2013. Network in network. arXiv preprint arXiv:1312.4400.

Shanmugamani, R. 2018. Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras. Packt Publishing Ltd.

Srivastava, R. K., Greff, K., & Schmidhuber, J. 2015. Training very deep networks. In Advances in neural information processing systems (pp. 2377-2385).

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. 2014. Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806.

ABSTRACT

A Study on Deep Learning for Image Classification Using Keras

Pham Dinh Tan[1], Tran Thi Thu Thuy[1], Diem Cong Hoang[1]

[1]*Faculty of Information Technology, Hanoi University of Mining and Geology*

Computer vision is a field of research on techniques that help computers automatically recognize, read and understand information from images. There is a diverse range of applications using computer vision such as vehical self-driving, image-based product quality control as well as in gaming and virtual reality. In recent years, computer vision observes a lot of advances with the booming of deep learning. Deep learning uses a multi-layered neural network architecture to extract features automatically based on data. Neural network (modeled to neural network in human brain) is a set of connected network nodes and information

is transmitted from one node to another. Many network architectures are proposed for deep learning. DenseNet is a network architecture with high performance in image recognition. Keras is an open-source library for implement and training deep learning networks in Python. In this article, Keras library is used to implement and train DenseNet for image classification. Deep learning network performance is evaluated using on CIFAR-10 image dataset.

Keywords: Neural network; computer vision; deep learning; keras; densenet.

Phát triển thuật toán nâng cao chất lượng tiếng nói thời gian thực

Dương Thị Hiền Thanh[1,*], Trần Thanh Huân[2]
[1]*Trường Đại học Mỏ - Địa chất*
[2]*Trường Đại học Công nghiệp Hà Nội*

TÓM TẮT

Nâng cao chất lượng tiếng nói (speech enhancement) là một trong những bài toán quan trọng trong lĩnh vực nghiên cứu về xử lý tiếng nói (speech processing) hiện nay. Nâng cao chất lượng tiếng nói được sử dụng trực tiếp trong hệ thống nhận dạng tiếng nói tự động để tăng độ chính xác nhận dạng trong môi trường nhiễu, được tích hợp trong các thiết bị viễn thông, điện thoại di động giúp loại bỏ các âm thanh nhiễu và nâng cao chất lượng đàm thoại,... Trong trường hợp ứng dụng cho hệ thống đàm thoại trực tiếp, yêu cầu đặt ra đối với thuật toán nâng cao chất lượng tiếng nói là phải có tốc độ tính toán thời gian thực và đáp ứng với cấu hình phần cứng thấp (như cấu hình điện thoại di động). Bài báo trình bầy thuật toán loại nhiễu và nâng cao chất lượng tiếng nói cho tín hiệu đơn kênh, đáp ứng các yêu cầu đối với hệ thống đàm thoại trực tiếp. Thí nghiệm được thực hiện với bộ dữ liệu test gồm tiếng nói tiếng Anh và tiếng Việt cho thấy hiệu quả của thuật toán.

Từ khóa: Nâng cao chất lượng tiếng nói; loại nhiễu; phổ âm thanh; thời gian thực.

1. Đặt vấn đề

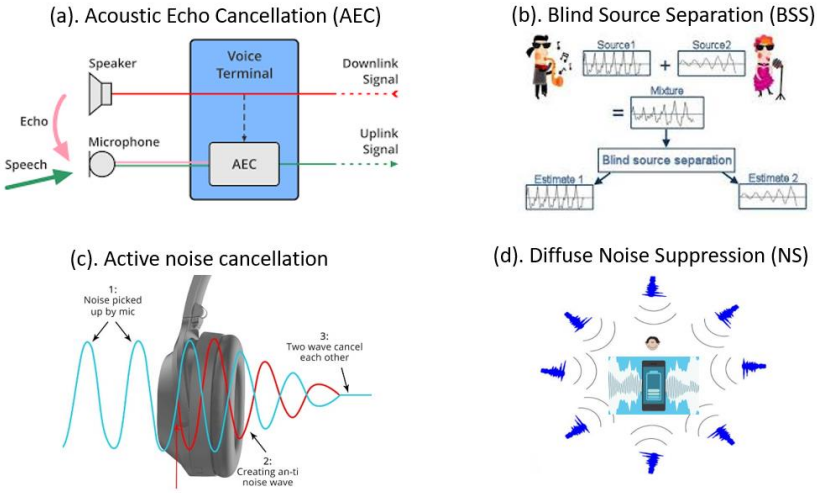
Nâng cao chất lượng tiếng nói là kỹ thuật nhằm loại bỏ tiếng ồn môi trường và những âm thanh không mong muốn khác (gọi chung là nhiễu) có trong tín hiệu thu âm và làm cho tiếng nói mong muốn (desired speech) trở nên "tốt" hơn (Benesty và nnk, 2005). Có nhiều kỹ thuật khác nhau đã được các nhóm nghiên cứu phát triển và áp dụng cùng nhằm mục tiêu giảm âm thanh nhiễu và nâng cao chất lượng tiếng nói mong muốn như: loại bỏ tiếng vọng (Acoustic Echo Cancellation - AEC, Hình 1(a)) (Reuven và nnk, 2007), loại bỏ âm thanh nhiễu (Active noise cancellation, Hình 1(c)) (Priyanga và nnk, 2013), tách nguồn âm thanh (Blind source separation, Hình 1(b)) (Duong và nnk, 2015; Dương và nnk, 2019; Emmanuel Vincent và nnk, 2014), hay loại bỏ tiếng ồn khuếch tán (Diffuse noise suppression, Hình 1(d)) (Murase và nnk, 2015). Các kỹ thuật kể trên có thể được sử dụng riêng biệt hoặc kết hợp với nhau tùy theo tình huống áp dụng và đặc điểm dữ liệu của bài toán. Tuy nhiên chúng có điểm chung là quá trình xử lý đều được thực hiện trên miền tần số, do đó cần có bước biến đổi dữ liệu âm thanh thu được ở miền thời gian sang miền tần số, và quá trình biến đổi ngược lại được thực hiện sau khi hoàn thành các phép xử lý tín hiệu.

Nâng cao chất lượng tiếng nói là một trong số những vấn đề rất quan trọng trong lĩnh vực xử lý âm thanh và xử lý tiếng nói và được áp dụng trực tiếp trong nhiều hệ thống thực tế. Hệ thống nâng cao chất lượng tiếng nói được tích hợp trong máy trợ thính giúp người khiếm thính dễ dàng hơn trong việc nhận biết âm thanh (Goehring và nnk, 2016; Khaleelur Rahiman và nnk, 2020; Revit & Schulein, 2013; Sunohara và nnk, 2017). Các hệ thống nhận dạng tiếng nói tự động cần có pha loại nhiễu và nâng cao chất lượng tiếng nói để kết quả nhận dạng đạt được độ chính xác cao hơn (Moore và nnk, 2017; Ochiai và nnk, 2017; Sainath và nnk, 2017). Đặc biệt trong lĩnh vực viễn thông mà điển hình là tình huống giao tiếp qua điện thoại di động, việc nâng cao chất lượng tiếng nói của người gọi là rất cần thiết bởi người dùng thường xuyên thực hiện cuộc gọi ở những nơi công cộng có rất nhiều tiếng ồn như ngoài đường, trong nhà ga, trong nhà hàng,… (Panahi và nnk, 2016). Trong trường hợp này, thuật toán nâng cao chất lượng tiếng nói không chỉ cần đáp ứng mục tiêu loại bỏ những âm thanh nhiễu và làm nổi rõ tiếng nói mong muốn, mà còn cần đáp ứng các yêu cầu khác như có tốc độ xử lý nhanh theo thời gian thực (real time) và chi phí tài nguyên thấp phù hợp với cấu hình hạn chế của điện thoại di động.

Chúng tôi tập trung nghiên cứu và cài đặt thử nghiệm thuật toán loại nhiễu dựa trên phương pháp ước lượng cực tiểu hóa trung bình bình phương sai số (Minimum Mean Square Error) của biên độ phổ tín hiệu âm thanh theo khung thời gian ngắn. Thuật toán có tốc độ xử lý theo thời gian thực và yêu cầu cấu hình

[*] *Tác giả liên hệ*
Email: duongthihienthanh@humg.edu.vn

phần cứng thấp có thể tích hợp trực tiếp cho điện thoại di động nhằm nâng cao chất lượng tiếng nói của người dùng. Thuật toán xử lý tín hiệu thu âm đơn kênh nên có thể áp dụng được cho cả điện thoại chỉ sử dụng một microphone và điện thoại nhiều microphone.



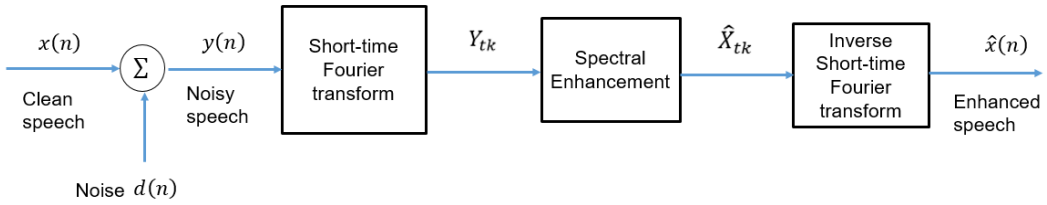
Hình 1. Các kỹ thuật thuật loại nhiễu và nâng cao chất lượng tiếng nói

2. Thuật toán loại nhiễu bằng phương pháp xử lý ước lượng phổ của tín hiệu

2.1. Mô hình tổng quát của thuật toán

Gọi $y(n)$ là tín hiệu tiếng nói thu âm trong môi trường thực tế biểu diễn trên miền thời gian. Khi đó $y(n)$ sẽ là tín hiệu trộn lẫn của tiếng nói $x(n)$ và nhiễu nền, tiếng vọng và các âm thanh không mong muốn khác, gọi chung là âm thanh nhiễu $d(n)$. Công thức biểu diễn tín hiệu thu âm đầu vào là:

$$y(n) = x(n) + d(n) \quad (1)$$



Hình 2. Sơ đồ thuật toán nâng cao chất lượng tiếng nói tổng quát

Khi chuyển sang miền tần số bằng phép biến đổi Fourier, công thức (1) được biến đổi thành công thức (2), với Y_{tk}, X_{tk}, D_{tk} lần lượt là ký hiệu của tín hiệu thu âm đầu vào, tiếng nói, và nhiễu trong miền tần số.

$$Y_{tk} = X_{tk} + D_{tk} \quad (2)$$

Sơ đồ khối của thuật toán nâng cao chất lượng tiếng nói bằng phương pháp ước lượng phổ âm thanh được mô tả như Hình 2. Từ tín hiệu đầu Y_{tk}, thuật toán thực hiện các phép tính toán trên ma trận phổ của tín hiệu để ước lượng tín hiệu tiếng nói $\hat{X}_{tk}$ trên miền tần số, với mục tiêu sai số giữa tiếng nói ước lượng được $\hat{X}_{tk}$ và tín hiệu tiếng nói sạch X_{tk} càng nhỏ càng tốt. Cuối cùng, tín hiệu tiếng nói sau khi ước lượng được chuyển sang miền thời gian bằng phép biến đổi Fourier ngược.

Giả sử tín hiệu tiếng nói và nhiễu trong miền tần số là các biến ngẫu nhiên độc lập, có quy luật phân bố Gaussian trung bình bằng 0 (zero-mean statistically independent Gaussian) và được biểu diễn dưới dạng công thức (3)

$$X_{tk} \sim \mathcal{N}\left(0, \hat{\lambda}_{tk}\right), \; D_{tk} \sim \mathcal{N}\left(0, \sigma_{tk}^2\right) \quad (3)$$

Sử dụng phương pháp cực tiểu hóa trung bình bình phương sai số (Minimum Mean Square Error) của biên độ phổ tín hiệu âm thanh theo khung thời gian, gọi tắt là MSE (Minimum mean square error short-time Spectral amplitude Estimation) (Ephraim & Malah, 1984), thuật toán thực hiện ước lượng hàm tín hiệu tiếng nói $\hat{X}_{tk}$ bằng bước lặp và cự tiểu hóa hàm loss sau:

$$\min_{\hat{X}_{tk}} E\left\{d(X_{tk}, \widehat{X_{tk}}) \middle| \hat{p}_{tk}, \hat{\lambda}_{tk}, \widehat{\sigma_{tk}^2}, Y_{tk}\right\}, \quad (4)$$

trong đó:

$\hat{p}_{tk}$ là giá trị ước lượng xác suất hiện diện của tín hiệu tiếng nói,

$\hat{\lambda}_{tk}$ là giá trị ước lượng phương sai phổ của tiếng nói X_{tk} (variance of speech spectral coefficient),

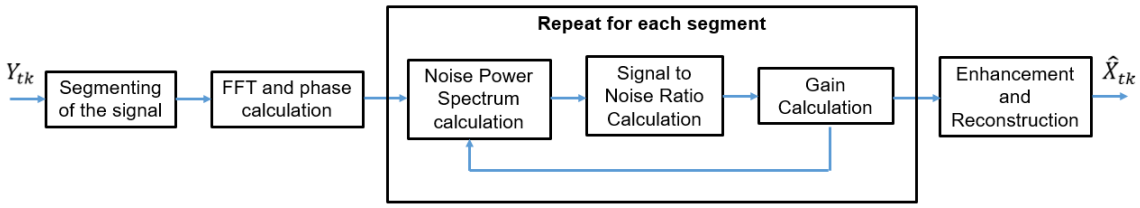
$\widehat{\sigma_{tk}^2}$ là giá trị ước lượng phương sai phổ của âm thanh nhiễu D_{tk} (variance of noise spectral coefficient),

Y_{tk} là tín hiệu âm thanh đầu vào đã xác định,

$d\left(X_{tk}, \widehat{X}_{tk}\right) = \left|g(X_{tk}) - g(\hat{X}_{tk})\right|^2$ với $g(X)$ có thể lựa chọn là một trong số các hàm $X, |X|, \log(X), \ldots$

2.2. Cải tiến thuật toán cho xử lý real time

Từ thuật toán mô tả trong phần 2.1, chúng tôi cải tiến thuật toán cho phép ước lượng tín hiệu tiếng nói theo từng đoạn âm thanh kích thước nhỏ (chẳng hạn 1 giây). Tín hiệu đầu vào được cắt thành các đoạn ngắn để đưa vào thuật toán, bước ước lượng tín hiệu tiếng nói theo công thức (4) được thực hiện theo từng khung (frame by frame) chứ không phụ thuộc vào toàn bộ file dữ liệu đầu vào như thuật toán ban đầu. Sơ đồ thuật toán cải tiến được thể hiện trong Hình 3.



Hình 3. Sơ đồ thuật toán nâng cao chất lượng tiếng nói cải tiến

Với mỗi đoạn ngắn âm thanh đầu vào, thuật toán thực hiện xử lý trên từng khung kích thước 0.02 giây, với bước dịch chuyển 0.01 giây. Như vậy các khung kế tiếp sẽ có sự chồng lấp 50%, điều này để đảm bảo có thể khôi phục tín hiệu sau khi phân tách và xử lý. Trước tiên, thuật toán ước lượng phương sai phổ của tín hiệu từ 6 khung đầu tiên và giả định đó là nhiễu σ_{tk}^2. Với mỗi khung tiếp theo, thuật toán tính toán và cập nhật phương sai nhiễu σ_{tk}^2 nếu xác định khung đó không chứa tiếng nói, hoặc cập nhật phương sai tiếng nói $\hat{\lambda}_{tk}$ nếu xác định được khung đó có chứa tín hiệu tiếng nói, từ đó tính SNR (Signal to Noise Ratio) theo công thức $\xi_{tk} = \frac{\lambda_{tk}}{\sigma_{tk}^2}$, đồng thời hàm Gain được xác định theo công thức (5)

$$G_{MSE}(\xi) = \frac{\xi}{1+\xi} \quad (5)$$

Cuối cùng tiếng nói được ước lượng bằng cách nhân hàm Gain với khung tín hiệu đầu vào:

$$\hat{X}_{tk} = G_{MSE}(\xi_{tk}) * Y_{tk} \quad (6)$$

Bằng cách xử lý theo từng khung, tốc độ tính toán của thuật toán đã được cải thiện đáng kể. Hơn nữa, thuật toán có thể xử lý nâng cao chất lượng tiếng nói cho từng đoạn ngắn âm thanh đầu vào nhằm phù hợp với mục tiêu có thể tích hợp sử dụng thuật toán trong hệ thống đàm thoại trực tiếp.

3. Kết quả thử nghiệm và thảo luận

3.1. Mô tả dữ liệu thử nghiệm

Để đánh giá hiệu quả loại nhiễu và nâng cao chất lượng tiếng nói của thuật toán, chúng tôi thiết kế hai tập dữ liệu thử nghiệm với hai ngôn ngữ tiếng Anh và tiếng Việt như sau:

Tập D1: Là tập dữ liệu chuẩn được thiết kế và sử dụng trong SISEC[†] campaign (International Signal Separation and Evaluation Campaign) từ năm 2010 cho đến nay. D1 gồm 9 file âm thanh đơn kênh là tín hiệu trộn lẫn giữa tiếng nói tiếng Anh và âm thanh nhiễu từ môi trường thu âm. Có 3 loại nhiễu môi trường khác nhau là quảng trường (square), nhà hàng (cafeteria), và ga tàu điện ngầm (subway). Các file dữ liệu có độ dài 10 giây và được lấy mẫu ở tần số 16 kHz [‡].

Tập D2: Là tập dữ liệu do nhóm tự xây dựng, gồm 6 file âm thanh trộn lẫn giữa tiếng nói tiếng Việt (của cả giọng nam và giọng nữ) và các loại nhiễu môi trường khác nhau (gồm tiếng ồn trong quán ăn, tiếng gió, tiếng mưa, tiếng ô tô và các phương tiện giao thông ngoài đường phố, tiếng sóng biển). Các file được tạo với tỷ lệ tín hiệu tiếng nói trên nhiễu SNR thay đổi ngẫu nhiên từ 0 dB đến 10 dB. Các file có độ dài thay đổi từ 5 đến 10 giây và tần số lấy mẫu là 16 kHz.

3.2 Phương pháp đánh giá kết quả

[†] https://sisec.inria.fr/sisec-2016/bgn-2016/

[‡] http://sisec.wiki.irisa.fr/tiki-index79f3.html?page=Two-channel+mixtures+of+speech+and+real-world+background+noise

Kết quả thử nghiệm được đánh giá dựa trên 2 tiêu chí:

- Đánh giá mức độ giảm nhiễu của thuật toán: Mức độ giảm nhiễu được đánh giá dựa trên hai độ đo theo tiêu chuẩn quốc tế được sử dụng rộng rãi trong cộng đồng xử lý âm thanh là SNR và SDR (Signal-to-Distortion Ratio). SNR đo mức độ triệt tiêu nhiễu của thuật toán bằng cách tính toán tỷ lệ hai loại tín hiệu tiếng nói và nhiễu. Độ đo SDR đánh giá mức độ triệt tiêu nhiễu của các thuật toán nhưng có tính đến độ méo của tín hiệu tiếng nói sau khi xử lý. Trong hai độ đo trên thì SDR được đánh giá là quan trọng hơn vì có tính đến cảm nhận của người nghe với chất lượng tiếng nói nói chung. Hai độ đo này được tính toán bằng phần mềm BSSEval[§], đơn vị tính là dB và có giá trị càng lớn càng tốt (Vincent và nnk, 2006).

- Đánh giá hiệu năng tính toán: chúng tôi so sánh tương quan giữa thời gian xử lý của thuật toán so với kích thước dữ liệu đầu vào. Thuật toán được đánh giá đạt tốc độ tính toán theo thời gian thực nếu thời gian tính toán của thuật toán trên máy tính thông thường nhỏ hơn hoặc bằng độ dài của file âm thanh đầu vào.

3.3 Kết quả và thảo luận

Kết quả đánh giá định lượng bằng các độ SNR, SDR được tổng hợp trong labtop 1 và Bảng 2. Kết quả được tính cho từng file âm thanh trong các tập D1 và D2, và giá trị trung bình cho tất cả các file. Cột "Unprocessed" là kết quả tính toán của tín hiệu đầu vào chưa qua xử lí.

Bảng 1. Kết quả của tập dữ liệu tiếng Anh D1

Method	Criteria	Ca1_Ce_A	Ca1_Ce_B	Ca1_Co_A	Sq1_Ce_A	Sq1_Ce_B	Sq1_Co_A	Sq1_Co_B	Su1_Ce_A	Su1_Ce_B	AVERAGE	INCREASE
Unprocessed	SNR	4.71	5.13	4.49	-3.26	-3.55	-5.12	-1.85	-13.73	-6.96	**-2.24**	
	SDR	4.42	4.98	4.36	-3.89	-4.18	-5.40	-2.13	-13.87	-7.10	**-2.54**	
MSE	SNR	13.42	16.67	18.55	13.51	16.29	17.28	23.16	3.66	9.38	**14.66**	**16.90**
	SDR	7.00	10.43	12.01	3.31	4.61	4.37	10.76	-1.40	2.91	**6.00**	**8.54**

Bảng 2. Kết quả của tập dữ liệu tiếng Việt D2

Method	Criteria	fvn_cafe	fvn_station	mvn_car	mvn_ocean	fvn_rain	mvn_wind	AVERAGE	INCREASE
Unprocessed	SNR	10.04	5.00	0.04	10.02	5.02	0.12	5.04	
	SDR	10.04	5.00	0.04	10.02	5.02	0.12	5.04	
MSE	SNR	24.65	22.33	27.64	22.75	18.84	12.71	**21.49**	**16.45**
	SDR	13.61	10.62	16.27	8.58	6.38	-5.82	**8.27**	**3.23**

Theo Bảng 1 kết quả thử nghiệm của tập dữ liệu SiSEC, độ đo SNR tăng 16,9 dB, và SDR tăng 8,54 dB sau khi áp dụng thuật toán xử lí tín hiệu MSE. Kết quả trên tập dữ liệu tiếng Việt D2 trong Bảng 2 cũng cho thấy các độ đo SNR và SDR tăng lần lượt là 16,45 dB và 3,23 dB sau khi xử lý loại nhiễu. Những kết quả đó khẳng định khả năng giảm nhiễu và nâng cao chất lượng tiếng nói của thuật toán MSE.

Về hiệu năng tính toán, để xử lý đoạn âm thanh đầu vào dài 1 giây, thuật toán cần tiêu tốn thời gian là 0.9 giây khi chạy trên máy laptop thông thường (Processor Intel(R) Core(TM) i5-6300U CPU @2.40GHz, RAM 16 GB) và hết khoảng 0,2 giây trên máy laptop đời mới. Dung lượng bộ nhớ thuật toán chiếm dụng khi chạy là khoảng 68 MB, chỉ hết gần 0,5% dung lượng 16Gb RAM của điện thoại di động thông thường hiện nay. Với những thông số đó, thuật toán hoàn toàn đáp ứng được khi tích hợp trên điện thoại di động có cấu hình trung bình.

4. Kết luận

Tín hiệu thu âm trong môi trường thực thường là sự trộn lẫn của nhiều nguồn âm thanh khác nhau, nhiễu nền, và tiếng vọng,... do đó việc phát triển các thuật toán loại nhiễu và nâng cao chất lượng tiếng nói mong muốn là rất cần thiết cho các hệ thống ứng dụng, trong đó có hệ thống đàm thoại trực tiếp qua điện thoại di động. Chúng tôi đã phát triển một thuật toán nâng cao chất lượng tiếng nói có tốc độ tính toán thời gian thực, đáp ứng với điều kiện cấu hình thấp của điện thoại di động. Thuật toán sử dụng phương pháp ước lượng cực tiểu hóa trung bình bình phương sai số biên độ phổ tín hiệu âm thanh theo từng khung thời gian ngắn, các phép tính toán xử lý được thực hiện theo từng khung độc lập. Thuật toán đã được cài đặt thử nghiệm trên hai tập dữ liệu tiếng Anh và tiếng Việt, được thiết kế mô phỏng theo đúng điều kiện thu âm thực tế với những môi trường ồn ào và mức độ tiếng ồn khác nhau. Kết quả thử nghiệm đã cho thấy khả năng cải thiện chất lượng tiếng nói của thuật toán cũng như xác thực thời gian tính toán và chi phí tài nguyên của thuật toán.

Trong tương lai, chúng tôi sẽ tiếp tục cải tiến thuật toán bằng cách thay đổi hàm tối ưu trong bước ước lượng phổ tín hiệu, tích hợp xử lý tiếng vọng, hoặc kết hợp MSE với mô hình mạng học sâu cũng là một

[§] http://bass-db.gforge.inria.fr/bss_eval/

hướng phát triển tiềm năng.

Tài liệu tham khảo

Benesty, J., Makino, S., & Chen, J. (2005). *Speech enhancement*. Springer.

Duong, H.-T. T., Nguyen, Q.-C., Nguyen, C.-P., Tran, T.-H., & Duong, N. Q. K. (2015). Speech enhancement based on nonnegative matrix factorization with mixed group sparsity constraint. *Proceedings of the Sixth International Symposium on Information and Communication Technology*, 247-251.

Duong, T. T. H., Duong, N. Q. K., Nguyen, P. C., & Nguyen, C. Q. (2019). Gaussian Modeling-Based Multichannel Audio Source Separation Exploiting Generic Source Spectral Model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *27*(1), 32-43.

Ephraim, Y., & Malah, D. (1984). Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *32*(6), 1109-1121.

Goehring, T., Yang, X., Monaghan, J. J. M., & Bleeck, S. (2016). Speech enhancement for hearing-impaired listeners using deep neural networks with auditory-model based features. *2016 24th European Signal Processing Conference (EUSIPCO)*, 2300-2304.

G.U.Priyanga, Mr.B.Prasad, T.Sangeetha, & P.Saranya. (2013). Active Noise Cancellation System Using DSP Prosessor. *International Journal of Scientific & Engineering Research*, *4*(4).

Khaleelur Rahiman, P., Jayanthi, V., & Jayanthi, A. (2020). RETRACTED: Speech enhancement method using deep learning approach for hearing-impaired listeners. *Health Informatics Journal*, 146045821989385.

Moore, A. H., Parada, P. P., & Naylor, P. A. (2017). Speech enhancement for robust automatic speech recognition: Evaluation using a baseline system and instrumental measures. *Computer Speech & Language*, *46*, 574-584.

Murase, Y., Chiba, H., Ono, N., Miyabe, S., Yamada, T., & Makino, S. (2015). Diffuse noise suppression with asynchronous microphone array based on amplitude additivity model. *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 599-603.

Ochiai, T., Watanabe, S., & Katagiri, S. (2017). Does speech enhancement work with end-to-end ASR objectives?: Experimental analysis of multichannel end-to-end ASR. *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1-6.

Panahi, I., Kehtarnavaz, N., & Thibodeau, L. (2016). Smartphone-based noise adaptive speech enhancement for hearing aid applications. *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 85-88.

Reuven, G., Gannot, S., & Cohen, I. (2007). Multichannel acoustic echo cancellation and noise reduction in reverberant environments using the transfer-function GSC. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference On*, *1*, I-81-I-84.

Revit, L. J., & Schulein, R. B. (2013). Sound reproduction method and apparatus for assessing real-world performance of hearing and hearing aids. *The Journal of the Acoustical Society of America*, *133*(2), 1196.

Sainath, T. N., Weiss, R. J., Wilson, K. W., Li, B., Narayanan, A., Variani, E., Bacchiani, M., Shafran, I., Senior, A., Chin, K., Misra, A., & Kim, C. (2017). Multichannel Signal Processing With Deep Neural Networks for Automatic Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(5), 965-979.

Sunohara, M., Haruta, C., & Ono, N. (2017). *Low-latency real-time blind source separation for hearing aids based on time-domain implementation of online independent vector analysis with truncation of non-causal components*. 216-220.

Vincent, E., Gribonval, R., & Fevotte, C. (2006). Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech and Language Processing*, *14*(4), 1462-1469.

Vincent, Emmanuel, Bertin, N., Gribonval, R., & Bimbot, F. (2014). From Blind to Guided Audio Source Separation: How models and side information can improve the separation of sound. *IEEE Signal Processing Magazine*, *31*(3), 107-115.

ABSTRACT

Real-time speech enhancement using a Minimum Mean Square Error Short-Time Spectral Amplitude Estimation method

Duong Thi Hien Thanh[1], Tran Thanh Huan[2]
[1] *Hanoi University of Mining and Geology*
[2] *Hanoi University of Industry*

Speech enhancement is a very important problem in the audio processing field. The speech enhancement algorithms are used directly in the automatic speech recognition (ASR) system to increase recognition accuracy in noisy environments, they are integrated into mobile telecommunications equipment, eliminating noise and improving the conversation quality, etc. In the case of applying to the live conversation system, the requirements for the speech enhancement algorithm are to have a real-time calculation speed, and responsive to low hardware configurations (like mobile phone configurations). The paper presents a speech enhancement algorithm for single-channel signals, which meet the requirements for direct communication systems. The experiment is done with two datasets including English and Vietnamese language to show the effectiveness of the algorithm.

Keywords: Speech enhancement; denoising; noise cancellation; spectral; real-time.

KHOA HỌC TRÁI ĐẤT VÀ TÀI NGUYÊN VỚI PHÁT TRIỂN BỀN VỮNG

ISBN 978-604762277-1