TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH
Tập 18, Số 3 (2021): 463-476
ISSN: 1859-3100
HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE
Vol. 18, No. 3 (2021): 463-476
Website: *http://journal.hcmue.edu.vn*

Bài báo nghiên cứu

ĐẢM BẢO TÍNH RIÊNG TƯ DỮ LIỆU VỚI HỌC LIÊN KẾT CẢI TIẾN

Nguyễn Thị Hường[1*], Bùi Huy Toàn[2], Lê Tấn Phong[2], Nguyễn Đình Thúc[2]

[1]Smartnet HCMC, Việt Nam

[2]Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh, Việt Nam

[*]Tác giả liên hệ: Nguyễn Thị Hường – Email: nguyenhuongk07@gmail.com

Ngày nhận bài: 02-3-2021; ngày nhận bài sửa: 18-3-2021; ngày duyệt đăng: 20-03-2021

TÓM TẮT

Mô hình hóa dữ liệu là bài toán quan trọng trong phân tích dữ liệu. Học máy là phương pháp được sử dụng rộng rãi để giải quyết bài toán mô hình hóa này. Hầu hết các mô hình học là cục bộ theo nghĩa dữ liệu huấn luyện mô hình được tập trung nơi máy chủ, do đó, không tận dụng được dữ liệu chia sẻ đa dạng từ nhiều nguồn. Kết quả là tính tổng quát hóa của mô hình thu được có thể bị hạn chế. Học liên kết là phương pháp học với dữ liệu huấn luyện từ nhiều nguồn, và vì thế, nó có nhiều ưu điểm so với các mô hình học khác. Mô hình học liên kết có thể được áp dụng cho nhiều dạng dữ liệu và nhiều thuật toán máy học khác nhau. Bên cạnh độ tổng quát hóa cao, mô hình học liên kết còn đảm bảo tính riêng tư cho tập dữ liệu huấn luyện. Bài báo này, đề xuất mô hình học liên kết cải tiến đảm bảo tính riêng tư dựa trên mô hình học liên kết. Kết quả thử nghiệm cho thấy tính khả thi có thể áp dụng vào các bài toán sử dụng học máy trong thực tế, đồng thời cũng mở ra những những thách thức tiếp tục nghiên cứu, cải tiến.

Từ khóa: đảm bảo tính riêng tư dữ liệu; mô hình liên kết; phân tích dữ liệu đảm bảo tính riêng tư; đảm bảo tính riêng tư với mô hình liên kết

1. Giới thiệu

Khai thác dữ liệu, học máy và học sâu đang ngày càng phát triển nhờ nguồn dữ liệu phong phú, khổng lồ. Cốt lõi của việc học máy là dữ liệu. Theo cách truyền thống, ta phải thu thập và lưu trữ rất nhiều dữ liệu tại một máy tính (máy chủ), rồi sử dụng thuật toán huấn luyện để học trên tập dữ liệu đó – học cục bộ. Trong các mô hình học cục bộ, do trách nhiệm bảo mật thuộc phía máy chủ, nơi thu thập và quản lí tất cả dữ liệu nên chất lượng về tính chính xác của mô hình học là quan trọng hơn tính riêng tư dữ liệu. Trong ngữ cảnh học từ nhiều nguồn dữ liệu được chia sẻ chung – học liên kết hay học phân tán, học phi tập trung, ở đó, các đối tác (local server) hợp tác cùng xây dựng mô hình từ dữ liệu (local data) và chỉ chia sẻ kết quả tính toán của mình trong quá trình học. Trong ngữ cảnh này, bên cạnh việc

Cite this article as: Nguyen Thi Huong, Bui Huy Toan, Le Tan Phong, & Nguyen Dinh Thuc (2021). Data privacy-preserving via improved federated learning model. *Ho Chi Minh City University of Education Journal of Science, 18*(3), 463-476.

yêu cầu về tính đúng đắn của mô hình thì vấn đề riêng tư của dữ liệu cũng là yêu cầu quan trọng không kém, đặc biệt là trong các lĩnh vực dữ liệu có tính cá nhân, nhạy cảm như dữ liệu tài chính, y tế, sinh học… Trong các lĩnh vực nhạy cảm như vậy, ngoài mục tiêu quan trọng là độ chính xác thì các mô hình thuật toán học hay phân tích dữ liệu cũng cần phải chú ý đến tính riêng tư của dữ liệu, đây tính chất đặc biệt quan trọng khi các bộ luật về bảo vệ dữ liệu cá nhân đang được nhiều nước trên thế giới áp dụng.

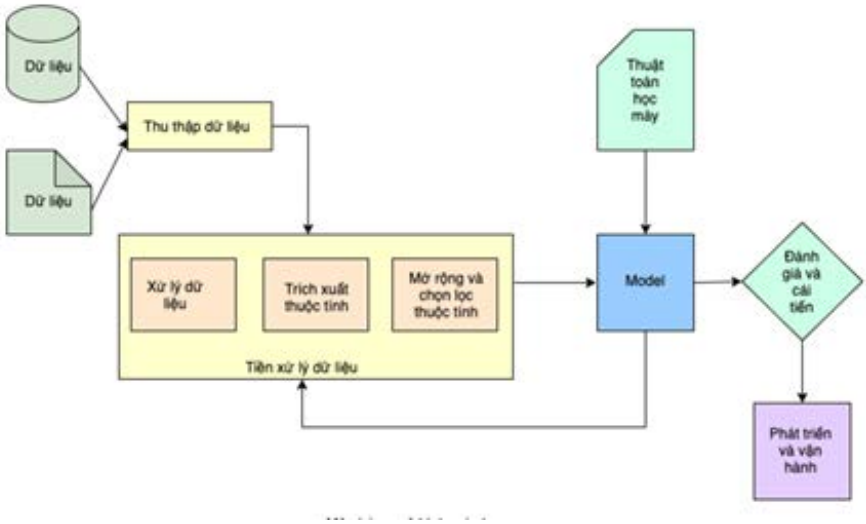
Có nhiều kĩ thuật hỗ trợ việc bảo vệ tính riêng tư cho dữ liệu. Trong đó, học phân tán là kĩ thuật đảm bảo tính riêng tư ngày càng trở nên phổ biến như hiện nay. Kĩ thuật này tổng quát, áp dụng được cho mọi dạng dữ liệu và các thuật toán học máy đa dạng.

2. Cơ sở lí thuyết

Phần này sẽ giới thiệu lại các khái niệm quan trọng sẽ được dùng cho các phần sau.

2.1. Học cục bộ

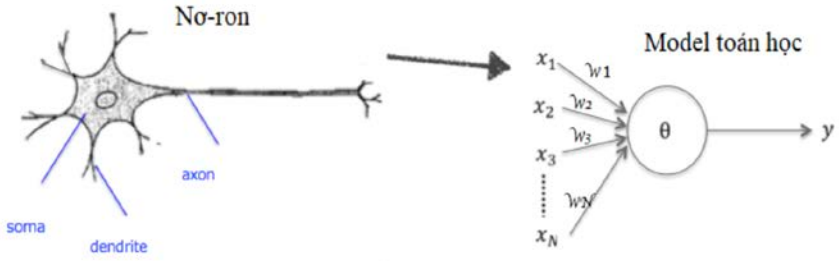
Mô hình học cục bộ không có định nghĩa rõ ràng, cụ thể mà chỉ là khái niệm để chỉ mô hình học tập trung cũ, trước khi có sự ra đời của mô hình học liên kết hay học phi tập trung. Mô hình học cục bộ (Hình 1) là cách tiếp cận truyền thống và được sử dụng rộng rãi trong lĩnh vực học máy. Tiến trình huấn luyện được tiến hành dựa trên một tập dữ liệu đã được thu thập và một thuật toán được thực thi tại cùng vị trí vật lí (cùng máy chủ lưu trữ và tính toán).



Hình 1. *Mô hình học cục bộ*

2.2. Mạng nơ-ron nhân tạo

Mạng nơ-ron nhân tạo (ANN – Artificial Neural Network), thường gọi tắt là mạng nơ-ron, là một mô hình toán học được dựa trên các mô hình nơ-ron sinh học. ANN gồm một nhóm các nơ-ron nhân tạo (nút) liên kết với nhau, thông tin được xử lí bằng cách truyền theo các liên kết và giá trị mới tại các nơ-ron được tính lại (Hình 2).



Hình 2. *Mạng nơ-ron và mô hình mạng nơ-ron*

Một nút sẽ nhận một hoặc nhiều tín hiệu đầu vào x và cho ra kết quả o ở dạng nhị phân. Các đầu vào ảnh hưởng nhiều hay ít vào nút đầu ra qua các tham số quan trọng tương ứng w của nó:

$$o = \begin{cases} 0 \; if \; w^T x \; \leq 0 \\ 1 \; if \; w^T x \; > 0 \end{cases}$$

Với chỉ những phép tính đơn thuần như vậy, trên thực tế mạng nơ-ron sẽ không thể phát hiện ra những quan hệ phức tạp của dữ liệu (ví dụ như: dự đoán nợ xấu, tài chính, các bài toán xử lí ảnh hay các bài toán rút trích ngữ nghĩa của văn bản) khả năng dự đoán của mạng nơ-ron sẽ bị giới hạn và giảm đi rất nhiều. Từ đó, xuất hiện khái niệm hàm kích hoạt (Activation function) là hàm có thể trả về các giá trị thực bị chặn. Sự kết hợp của các hàm kích hoạt nhằm giúp các mô hình có thể học được các quan hệ phi tuyến phức tạp tiềm ẩn trong dữ liệu. Một số hàm kích hoạt cơ bản như[1]:

Hàm sigmoid: $\delta(z) = \frac{1}{1+e^{-z}}$

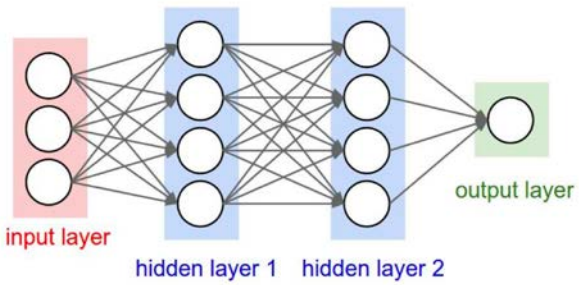
Hàm Tanh: $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Hàm ReLU: $f(x) = \max(0, x)$

Hàm Leaky ReLU: $f(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x)$ (với α là hằng số nhỏ)

2.3. Kiến trúc mạng nơ-ron

Một mạng nơron đơn giản được cấu thành bởi các nơ-ron đơn lẻ, được gọi là perceptron. Mạng nơ-ron tổng quát là sự kết hợp của các tầng perceptron.



Hình 3. *Mạng nhiều tầng*

[1] Xem liên kết Activation functions in neural networks(2020) (có sẵn trên mạng ngày 25 tháng 2 năm 2021).

Một mạng nơ-ron có ba kiểu tầng (Hình 3):

- Tầng vào (input layer): là tầng bên trái cùng của mạng biểu diễn các đầu vào của mạng;
- Tầng ra (output layer): là tầng bên phải cùng của mạng, biểu diễn các đầu ra của mạng;
- Tầng ẩn (hidden layer): là (các) tầng nằm giữa tầng vào và tầng ra biểu diễn cho việc suy luận logic và tính toán trung gian của mạng;

Một mạng nơ-ron luôn chỉ có 1 tầng vào và 1 tầng ra, nhưng có thể không có hoặc có nhiều tầng ẩn. Ở mỗi tầng, số lượng các nút nơ-ron có thể khác nhau, tùy thuộc vào bài toán và cách giải quyết.

2.4. Mạng nơ-ron tích chập

Tích chập là một khái niệm trong xử lí tín hiệu số nhằm biến đổi thông tin đầu vào thông qua một phép tích chập với bộ lọc để trả về đầu ra là một tín hiệu mới. Tín hiệu này sẽ làm giảm những đặc trưng mà bộ lọc không quan tâm và chỉ giữ những đặc trưng chính.

Cũng giống như mạng nơ-ron truyền thống, mạng tích chập (Hình 4) hoạt động theo phương thức nhận thông số đầu vào là các điểm tín hiệu và biến đổi tín hiệu đó thông qua các tầng mạng. Tuy nhiên, điểm khác biệt nằm ở cấu trúc của đầu vào và cấu trúc bên trong các tầng của mạng tích chập.

Lấy cảm hứng từ xử lí ảnh nên đầu vào của mạng tích chập có cấu trúc ma-trận như một bức ảnh chứ không có dạng vector như mạng nơ-ron nhân tạo thông thường. Cụ thể, một bức ảnh sau khi số hóa có dạng ngang-dọc-sâu (ngang: số lượng điểm ảnh trên chiều rộng, dọc: số lượng điểm ảnh trên chiều cao, sâu: số lượng kênh như RGB có 3 kênh đại diện cho mức độ của 3 màu đỏ, lục, lam) nên đầu vào của mạng tích chập là một ma trận 3 chiều.

Mạng nơ-ron tích chập gồm các tầng tích chập, đệm và bước nhảy, tầng gộp dùng để kết hợp thông tin qua các vùng không gian kề nhau, qua việc sử dụng đa kênh hay bộ lọc ở mỗi tầng.

Tầng tích chập là lớp quan trọng nhất và cũng là lớp đầu tiên của mô hình mạng tích chập. Lớp này có chức năng chính là phát hiện các đặc trưng không gian hiệu quả. Tầng tích chập nhận đầu vào là ma trận 3 chiều và một bộ lọc cần phải học. Bộ lọc này sẽ trượt qua từng vị trí trên bức ảnh để tính tích chập giữa bộ lọc và phần tương ứng trên bức ảnh. Ma trận kết quả của quá trình này được gọi là ma trận đặc trưng.

Bước nhảy là số lượng điểm ảnh dịch chuyển ma trận đầu vào hay dùng để dịch chuyển bộ lọc theo mỗi bước xác định.

Đệm là thêm các điểm ảnh vào xung quanh bức ảnh để giữ nguyên kích cỡ ma trận đặc trưng ban đầu. Mục đích là sau mỗi lần sử dụng các bộ lọc để quét ảnh, kích thước của ảnh sẽ nhỏ hơn và sẽ không giữ nguyên kích thước ban đầu của ảnh nên sẽ không thể khai thác được ảnh nữa, do đó cần thêm một số điểm ảnh bên ngoài vào hình ảnh.

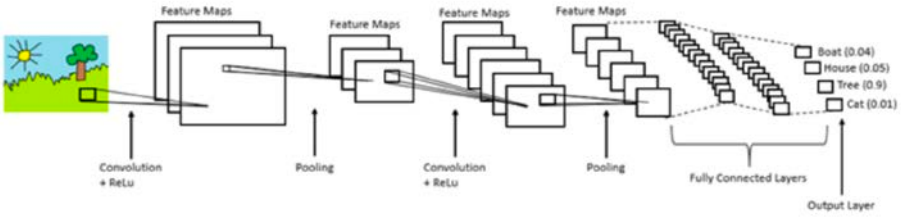
Tầng gộp thường được sử dụng ngay sau bước phi tuyến của tầng tích chập để đơn giản hóa thông tin đầu ra giảm bớt số lượng nơ-ron. Quá trình gộp phổ biến là gộp cực đại (max-pooling), thủ tục này chọn giá trị lớn nhất trong vùng đầu vào.

Ý tưởng đằng sau tầng gộp là vị trí tuyệt đối của những đặc trưng trong không gian ảnh không còn cần thiết, thay vào đó là vị trí tương đối giữ các đặc trưng đã đủ để phân loại đối tượng. Hơn nữa giảm tầng gộp có giúp giảm đi số chiều, làm hạn chế việc quá khớp (over fitting) và giảm thời gian huấn luyện.

ReLU (Rectified Linear Units, $y = f(x) = \max(0, x)$) là hàm kích hoạt phổ biến nhất cho mạng nơ-ron tích chập tại thời điểm hiện tại. Trước khi hàm ReLU được áp dụng thì hàm mức, hàm Sigmoid hay Tanh, là những hàm phổ biến. Hàm ReLU được ưa chuộng vì tính toán đơn giản, giúp hạn chế tình trạng tiêu biến gradient (đạo hàm xấp xỉ 0) và cũng cho kết quả tốt hơn.

Tầng cuối cùng của mô hình mạng nơ-ron tích chập trong bài toán phân loại ảnh là tầng kết nối đầy đủ. Tầng này có chức năng chuyển ma trận đặc trưng ở tầng trước thành vector chứa xác suất của các đối tượng cần được dự đoán.

Và cuối cùng, quá trình huấn luyện mô hình CNN cho bài toán phân loại ảnh cũng tương tự như huấn luyện các mô hình khác. Cần định nghĩa hàm lỗi để tính sai số giữa dự đoán của mô hình và nhãn chính xác, cũng như sử dụng thuật toán lan truyền ngược cho quá trình cập nhật trọng số.



Hình 4. *Cấu trúc CNN*

2.5. Thuật toán tối ưu Gradient Descent

Như ta đã biết, để tìm cực trị của một hàm số liên tục $y = f(x)$ sẽ phải giải phương trình đạo hàm của hàm đó: $f'(x) = 0$.

Nhưng phương trình trên không phải lúc nào cũng giải được dễ dàng, có những trường hợp việc giải phương trình trên là bất khả thi. Gradient Descent là cách thức tìm các điểm cực tiểu cục bộ này một cách xấp xỉ sau một số vòng lặp. Trong thực tế, các giá trị dữ liệu thường không đúng 100% mà đôi khi chúng ta cần những con số gần đúng nên những cách tính toán xấp xỉ, gần đúng là giải pháp phù hợp nhất.

Gradient Descent là một thuật toán lặp tối ưu được sử dụng trong các bài toán học máy và nhất là mạng học sâu, thường là các bài toán tối ưu lồi với mục tiêu là tìm một tập các biến nội tại cho việc tối ưu mô hình máy học. Ý tưởng của Gradient Descent thực hiện, tại

mỗi điểm của hàm lỗi hay hàm mất mát, nó sẽ xác định độ dốc sau đó đi ngược lại với hướng của độ dốc đến khi nào độ dốc tại chỗ đó gần hoặc bằng 0 (cực tiểu).

Gradient Descent có nhiều dạng khác nhau như Stochastic Gradient Descent (SGD), Mini-batch Gradient Descent (MGD). Về cơ bản thì các dạng Gradient Descent đều được thực thi như sau:

a. Khởi tạo biến nội tại
b. Đánh giá model dựa vào biến nội tại và hàm mất mát (loss function)
c. Cập nhật các biến nội tại theo hướng tối ưu hàm mất mát
d. Lặp lại bước đánh giá và cập nhật (bước b và c) cho tới khi thỏa điều kiện dừng.

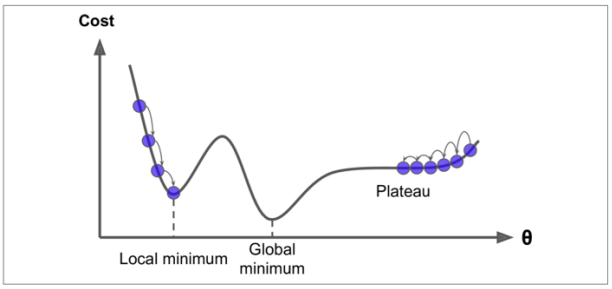
Công thức cập nhật cho Gradient Descent có thể viết là:

$\theta \longleftarrow \theta - \eta \nabla_\theta f(\theta)$

θ: tập các biến cần cập nhật, η: tốc độ học (learning rate), $\nabla_\theta f(\theta)$: gradient của hàm mất mát f theo tập θ.

Tối ưu hàm mất mát là việc tìm các điểm tối ưu mà ở đó hàm mất mát đạt cực đại hoặc cực tiểu. Nếu hàm mất mát không phải là hàm lồi thì sẽ có các cực đại cục bộ hoặc cực tiểu cục bộ bên cạnh các cực đại cực tiểu. Tuy nhiên, trong bài toán tối ưu lồi áp dụng Gradient Descent thì các cực tiểu cục bộ của hàm mất mát cũng chính là cực tiểu toàn cục của nó.

Có 1 tham số quan trọng trong Gradient Descent đó là giá trị độ lớn của mỗi lần di chuyển. Tham số này được gọi là tốc độ học hay hệ số học (learning rate). Nếu tốc độ học quá nhỏ, thuật toán sẽ phải thực hiện nhiều bước để hội tụ và sẽ mất nhiều thời gian. Nhưng nếu tốc độ học quá lớn sẽ khiến thuật toán đi qua cực tiểu, và vượt hẳn ra ngoài khiến thuật toán không thể hội tụ (xem Hình 5).



Hình 5. *Minh họa tham số tốc độ học*

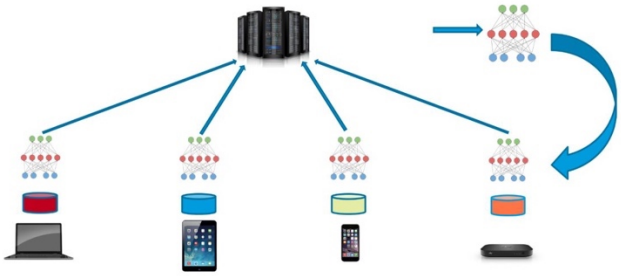
Như Hình 5 minh họa, điểm xuất phát có thể ở bên trái hoặc bên phải, nếu xuất phát từ bên trái, thuật toán sẽ hội tụ ở điểm cực tiểu cục bộ mà không đi đến được cực tiểu toàn cục. Nếu điểm xuất phát từ bên phải sẽ phải mất nhiều thời gian để vượt qua điểm lồi để đến được điểm cực tiểu toàn cục và nếu kết thúc thuật toán quá sớm sẽ không đến được điểm cực tiểu toàn cục.

Các bài toán trong thực tế áp dụng Batch Gradient Descent thường khó tìm được các cực tiểu toàn cục, đa phần rơi vào các cực tiểu cục bộ, tuy nhiên, chúng ta vẫn có thể chấp nhận các kết quả Gradient Descent trả về khi mô hình đã đủ tốt.

2.6. Mô hình học liên kết

Mô hình học liên kết (federate learning) là kĩ thuật trong học máy, được dùng để huấn luyện mô hình một cách phi tập trung. Khác với cách tiếp cận truyền thống – học cục bộ, mô hình học liên kết triển khai trên hệ thống dữ liệu phi tập trung thay vì phải thu thập tất cả các dữ liệu về máy chủ. Do đó mà các thiết bị tham gia vào mô hình này sẽ được hưởng lợi từ việc mô hình huấn luyện được học từ nhiều nguồn dữ liệu từ khác nhau, giúp đưa ra kết quả, dự đoán chính xác hơn, tổng quát hơn so với chỉ học trên tập dữ liệu máy cục bộ.

Mô hình học liên kết cho phép nhiều điểm (thiết bị) tham gia mà không cần chia sẻ dữ liệu, tài nguyên; thay vào đó các thiết bị - máy con chỉ trao đổi bộ tham số của mô hình huấn luyện. Vì thế, máy chủ không hề biết chi tiết dữ liệu của máy con. Điều này giúp giải quyết các vấn đề bảo mật thông tin, an toàn dữ liệu và quyền truy cập cơ sở dữ liệu.



Hình 6. *Mô hình học liên kết*

2.7. Cấu trúc mô hình học liên kết

Mô hình học liên kết được chia làm hai phần chính là (i) các thiết bị tham gia hay còn gọi là máy con và (ii) máy chủ. Các máy con không cần kết nối với nhau mà chỉ cần kết nối với máy chủ. Nhìn chung, nó khá giống với cấu trúc Client – server được áp dụng phổ biến trong lĩnh vực mạng máy tính, điển hình là các trang web.

Trong một hệ thống, máy chủ được xem là trái tim của cả hệ thống, giữ vai trò quan trọng trong việc vận hành hệ thống, nếu máy chủ không hoạt động đồng nghĩa với cả hệ thống sẽ dừng hoạt động. Trong mô hình học liên kết cũng tương tự, máy chủ giữ vai trò quan trọng trong việc quản lí các máy con, duy trì sự hoạt động của mô hình. Máy chủ có các chức năng chính:

- Quản lí, điều khiển các máy con trong mô hình học;
- Thực hiện tổng hợp thông số của mô hình huấn luyện;
- Lưu trữ các mô hình huấn luyện đã được tổng hợp.

Bên cạnh đó, máy chủ còn có bộ dữ liệu của riêng mình để kiểm thử mô hình huấn luyện nhằm đánh giá mô hình huấn luyện, từ đó đưa ra quyết định sử dụng mô hình huấn luyện nào tốt nhất. Nếu trường hợp máy chủ không có các dữ liệu kiểm thử thì sẽ đánh giá kết quả của mô hình huấn luyện thông qua các máy con tham số.

Máy con hay các thiết bị tham gia vào mô hình học liên kết giữ vai trò quan trọng không kém trong hệ thống. Mỗi máy con được xem như một mô hình học truyền thống cục bộ. Máy con có các chức năng chính:

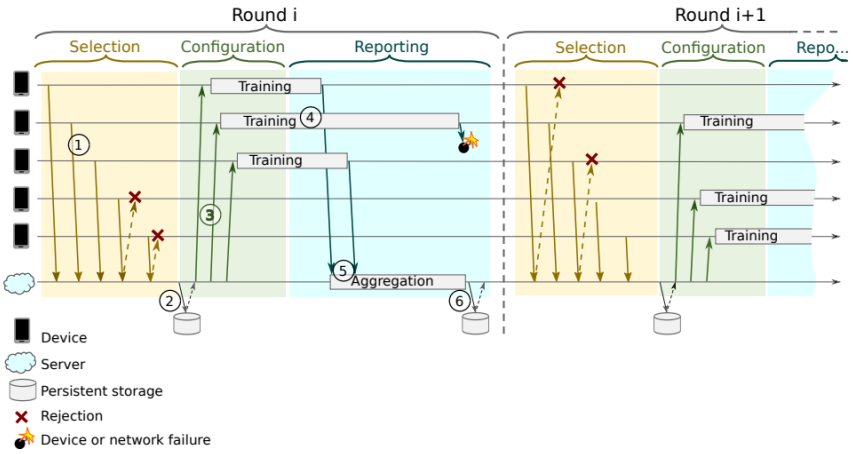
- Huấn luyện mô hình huấn luyện với tập dữ liệu riêng;
- Cập nhật bộ tham số của mô hình huấn luyện lên máy chủ;
- Cập nhật bộ tham số của mô hình huấn luyện từ máy chủ về.

Máy con sở hữu tập dữ liệu riêng (local data) nhưng mô hình huấn luyện phải đồng nhất với mô hình huấn luyện của cả hệ thống mô hình học liên kết, đồng thời cũng cần đồng nhất giữa các tham số trong quá trình học của các máy con. Điểm khác là thay vì liên tục cho mô hình huấn luyện học dữ liệu, các máy con sau một khoảng thời gian học nhất định phải gửi bộ tham số của mô hình học lên máy chủ, sau đó chờ máy chủ tổng hợp ra bộ tham số mới, nhận bộ tham số mới đó và tiếp tục quá trình học cục bộ của mình. Như vậy, mô hình huấn luyện của máy con đã bị chi phối bởi máy chủ, không còn mang tính cá nhân và đã chuyển sang trạng thái tổng quát trên toàn bộ các máy con khác của mô hình học liên kết.

2.8. Hoạt động của học liên kết mạng học sâu

2.8.1. Nguyên lí hoạt động

Nguyên lí hoạt động của mô hình học liên kết tập trung vào cách thức giao tiếp giữa máy chủ và máy con. Về nguyên lí hoạt động ở máy con sẽ gần tương tự như mô hình học cục bộ (xem Hình 7).



Hình 7. *Mô hình học liên kết*

Mô hình học liên kết gồm:

Devices: 1 hay nhiều máy con

Server: 1 máy chủ

Persistent storage: nơi lưu trữ mô hình huấn luyện (thường được lưu trữ trên máy chủ)

Rejection: tín hiệu bị từ chối

Devide or network failure: tín hiệu lỗi từ máy con hoặc mạng kết nối.

Khá giống với mô hình học cục bộ ở bước học mô hình huấn luyện, quy trình học của mô hình học liên kết cũng được chia ra nhiều vòng tổng hợp. Mỗi vòng tổng hợp là một quy trình nhỏ, là sự trao đổi và tổng hợp bộ tham số của mô hình huấn luyện giữa máy con và máy chủ. Chi tiết hơn, mỗi vòng tổng hợp gồm có 3 pha: pha lựa chọn, pha cấu hình và pha báo cáo

Pha lựa chọn là pha đầu tiên trong quy trình, nó giúp máy chủ chọn ra những ứng cử viên tham gia vào vòng tổng hợp này.

Đầu tiên, bước (1), máy chủ sẽ phát tín hiệu thông báo pha lựa chọn bắt đầu, điều này cũng đồng nghĩa với việc bắt đầu một vòng tổng hợp mới. Máy con nào nhận được tín hiệu và đang ở trong trạng thái sẵn sàng sẽ gửi thông báo sẵn sàng cho máy chủ. Những máy con đang còn làm công việc dang dở sẽ tiếp tục làm và đợi đến thông báo cho pha lựa chọn sau. Máy chủ sau một khoảng thời gian chờ nhất định sẽ thu thập được một tập hợp máy con có tín hiệu sẵn sàng, sau đó thực hiện chọn ngẫu nhiên ra các máy con sẵn sàng đó với số lượng nhất định và gửi thông báo chấp nhận pha lựa chọn cho các máy con. Máy con nhận được thông báo chấp nhận pha lựa chọn sẽ chuyển từ trạng thái sẵn sàng sang trạng thái đang hoạt động và chờ nhận thống báo từ pha tiếp theo. Còn các máy con không được chọn thì máy chủ sẽ gửi thông báo từ chối, các máy con nhận được thông báo từ chối sẽ tiếp tục ở trạng thái sẵn sàng và đợi đến pha lựa chọn tiếp theo. Trong trường hợp hết thời gian chờ của pha lựa chọn mà máy chủ không nhận đủ số lượng tín hiệu từ máy con ở trạng thái sẵn sàng thì sẽ hủy pha lựa chọn này, gửi thông báo từ chối tới tất cả máy con mà máy chủ đã gửi thông báo chấp nhận pha lựa chọn trước đó để các máy con trở lại trạng thái sẵn sàng, đợi một khoảng thời gian nhất định và bắt đầu lại pha lựa chọn mới

Bắt đầu pha cấu hình, bước (2), máy chủ sẽ đọc bộ tham số mô hình huấn luyện mới nhất từ nơi lưu trữ mô hình huấn luyện. Sau đó, bước (3), máy chủ gửi bộ tham số mô hình huấn luyện cho tất cả các máy con được chọn từ pha lựa chọn trước đó. Bước (4), các máy con sau khi nhận được bộ tham số mô hình huấn luyện từ máy chủ sẽ áp nó vô mô hình huấn luyện của mình và tiến hành huấn luyện mô hình huấn luyện mới được cập nhật bằng tập dữ liệu của mình. Ở máy con, không nhất thiết là cứ xong một vòng lặp học cục bộ thì máy con gửi bộ tham số mà máy con có thể thực hiện một số lượng vòng lặp học cục bộ nhất định rồi mới gửi bộ tham số cho máy chủ.

Pha báo cáo được bắt đầu sau khi máy chủ thực hiện gửi bộ tham số mô hình huấn luyện từ pha cấu hình cho tất cả các máy con hoàn tất. Ở pha này, máy chủ sẽ đợi các máy con được chấp nhận huấn luyện mô hình huấn luyện xong và gửi bộ tham số của mô hình huấn luyện mới lên. Máy con gửi bộ tham số mô hình huấn luyện xong sẽ chuyển trạng thái sang sẵn sàng và chờ vòng tổng hợp tiếp theo. Bước (5), sau khi nhận đủ tất cả bộ tham số mô hình huấn luyện, máy chủ tiến hành tổng hợp tất cả bộ tham số mô hình huấn luyện thành một bộ tham số mô hình huấn luyện mới và bước (6), lưu trữ lại tại nơi lưu trữ mô hình huấn luyện. Máy con có thể gửi tín hiệu lỗi máy con hoặc lỗi kết nối đến máy chủ khi gặp sự cố

về quá trình huấn luyện mô hình huấn luyện hoặc về kết nối, máy chủ nhận được tín hiệu lỗi sẽ bỏ qua máy con đó. Máy con gửi tín hiệu lỗi dừng mọi hoạt động lại và chuyển sang trạng thái sẵn sàng để chờ vòng tổng hợp mới. Trong trường hợp hết thời gian chờ của pha báo cáo mà máy chủ vẫn chưa nhận đủ bộ tham số mô hình huấn luyện của các máy con, máy chủ sẽ gửi tín hiệu hết thời gian thông báo cho các máy con chậm trễ, sau đó tính tổng các máy con đã gửi bộ tham số mô hình huấn luyện nếu đủ số lượng thì sẽ tiến hành tổng hợp như bình thường, còn không sẽ hủy pha báo cáo này và dừng vòng tổng hợp lại, bắt đầu vòng tổng hợp mới

2.8.2. Tổng hợp bộ tham số mô hình huấn luyện

Trong máy học, các thuật toán mạng nơ-ron có thể biểu diễn dưới dạng hàm $f(x)$ với bộ tham số là tập hợp $w = \{w_1, w_2, \dots, w_k\}$ với k là số lượng tham số có trong hàm $f(x)$. Đây cũng chính là bộ tham số mà máy chủ và máy con truyền nhận cho nhau và tại máy chủ sự tổng hợp cũng diễn ra trên bộ tham số này.

$$w_{tb} = \left\{ w_1, w_2, \dots, w_k | w_i = \frac{1}{n} \sum_{j=0}^{n} w_{ji} \; với \; i \; \in [1, k] \right\}$$

trong đó:

w_{tb}: bộ tham số mô hình huấn luyện tổng hợp

k: số lượng tham số trong bộ tham số mô hình huấn luyện tổng hợp

n: số lượng bộ tham số của máy con cần được tổng hợp

w_{ji}: tham số tại vị trí i thuộc bộ tham số của máy con j.

2.8.3. Thuật toán

1. **procedure:** sumWeight_FL()
2. **Input:** w_{con} ← mảng hai chiều có n bộ tham số mô hình huấn luyện con, mỗi bộ tham số mô hình huấn luyện con có k tham số
3. $for\ i = 0 \rightarrow k$:
4. $tong = 0$
5. $for\ j = 0 \rightarrow n$:
6. $tong = tong + w_{con}[j][i]$
7. $w_{tb}[i] = tong/n$
8. **return** w_{tb} ← mảng bộ tham số tổng hợp có k tham số

3. Đề xuất thuật toán cải tiến về bộ nén tham số mô hình huấn luyện

Việc truyền tải bộ tham số mô hình huấn luyện qua lại giữa các máy con với máy chủ xảy ra liên tục trong quá trình học, tạo sức ép lớn lên mạng kết nối cũng như ảnh hưởng tới thời gian đợi của máy chủ trong pha báo cáo. Nếu mô hình huấn luyện lớn, kéo theo dung lượng bộ tham số cũng lớn theo, có thể lên tới vài trăm MB. Điều này gây khó dễ cho các máy con có đường truyền mạng kém, không ổn định. Đồng thời, cũng gây khó khăn cho máy chủ khi phải tiếp nhận và xử lí bộ tham số dung lượng lớn của nhiều máy con cùng lúc.

Vì vậy, chúng tôi tìm cách giảm dung lượng của bộ tham số mô hình huấn luyện, giúp tăng tốc độ cũng như giảm áp lực về sự truyền tải tham số qua mạng.

Ý tưởng cho việc giảm dung lượng bộ tham số mô hình huấn luyện là bỏ đi ngẫu nhiên một tập con của các tham số trong bộ tham số đó theo một ngưỡng tối thiểu cho trước, rồi mới gửi bộ tham số đó lên máy chủ. Khi máy chủ thực hiện tổng hợp các bộ tham số, nếu tham số nào rỗng sẽ bỏ qua, chỉ lấy trung bình cộng những tham số khác rỗng.

Cơ sở cho ý tưởng nén này là về mặt lí thuyết, nếu các máy con có lượng dữ liệu đủ lớn và mô hình huấn luyện phù hợp thì mỗi mô hình huấn luyện của từng máy con đều hướng tới cùng một mô hình huấn luyện chung tốt, bộ tham số sẽ gần giống nhau. Nên khi ta ngẫu nhiên bỏ đi tham số trong bộ tham số của máy con này thì theo cách ngẫu nhiên sẽ có máy con khác không bỏ đi tham số đó. Như vậy, các tham số sẽ được xen kẽ bổ sung cho nhau ở các máy con. Ý tưởng này sẽ không ảnh hưởng quá nhiều tới kết quả chung.

Thuật toán nén bộ tham số

1. **procedure:** scaleWeight_FL()
2. **Input:** w_{con} ← mảng bộ tham số ở máy con có k tham số
3. $dungluong(x)$ ← hàm tính dung lượng của một biến x
4. Xáo trộn ngẫu nhiên phần tử trong w_{con}
5. $tongdungluong = 0$
6. $for\ i = 0\ \rightarrow k$:
7. $w_{con}[i] = \emptyset$
8. $tongdungluong = tongdungluong + dungluong(w_{con}[i])$
9. $if\ \frac{tongdungluong}{dungluong(w_{con})} > \emptyset$:
10. break

Thuật toán tổng hợp bộ tham số

1. **procedure:** sumScaleWeight_FL()
2. **Input:** w_{con} ← mảng hai chiều có n bộ tham số mô hình huấn luyện con, mỗi bộ tham số mô hình huấn luyện con có k tham số
3. $for\ i = 0\ \rightarrow k$:
4. $tong = 0$
5. $soluong = 0$
6. $for\ j = 0 \rightarrow n$:
7. $if\ w_{con}[j][i] \neq \emptyset$:
8. $tong = tong + w_{con}[j][i]$
9. $soluong += 1$
10. $if\ soluong \neq \emptyset$:
11. $w_{tb}[i] = tong/soluong$
12. **return** w_{tb} ← mảng bộ tham số tổng hợp có k tham số

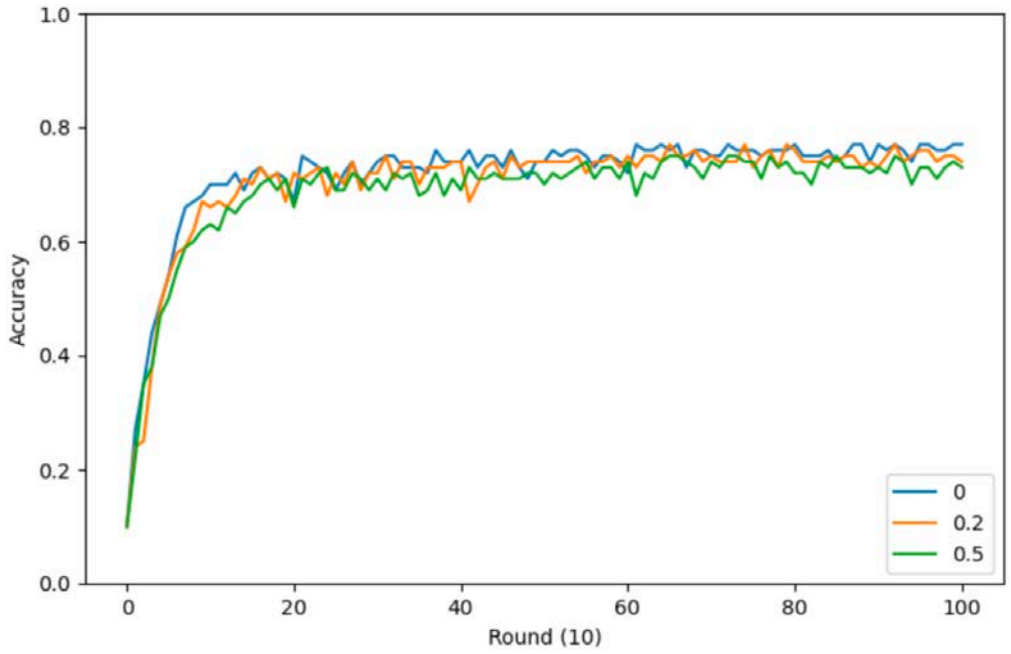
4. Kết quả

Để đánh giá được phương pháp, chúng tôi dùng tập dữ liệu huấn luyện CIFAR10 gồm 60.000 hình được chia làm 2 tập, 50.000 hình là tập huấn luyện và 10.000 hình là tập kiểm tra. Mỗi hình có kích thước 32×32 là hình màu gồm có 10 nhãn là: máy bay, xe hơi, chim, mèo, nai, chó, ếch, ngựa, tàu và xe tải.

Để giải quyết bài toán phân loại ảnh, chúng tôi sử dụng mạng nơ-ron tích chập CNN và mô hình học liên kết rút gọn. Chúng tôi thực hiện khảo sát số lượng máy con từ 1 đến 100, với số lượng máy con bằng 1 thì được xem là mô hình học cục bộ. Các yếu tố còn lại:

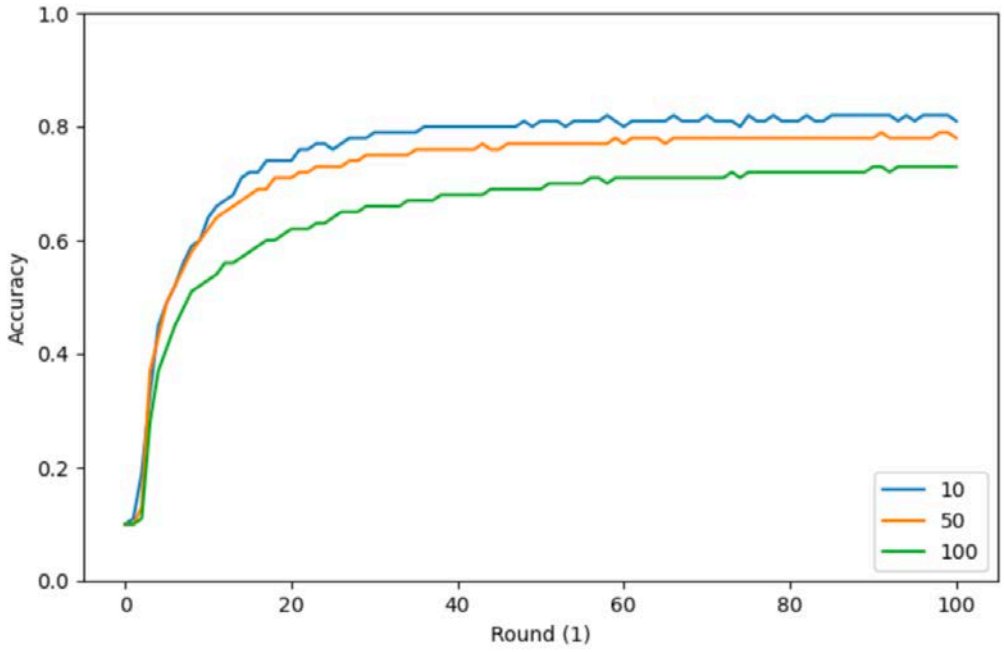
- Độ lớn batch bằng 100;
- Dữ liệu được chia đều ngẫu nhiên cho từng máy con. Số lượng dữ liệu ở máy con bằng số lượng tập huấn luyện chia số lượng máy con.

Kết quả sử dụng nén bộ tham số:



Hình 8. *Minh họa 1 vòng trong Học liên kết cải tiến*

Kết quả không nén bộ tham số



Hình 9. *Minh họa vòng 1 trong Học liên kết gốc*

Nén bộ tham số mô hình huấn luyện vừa dễ dàng hiểu về mặt lí thuyết và dễ dàng triển khai về mặt thực tế, đem lại tốc độ hội tụ và độ chính xác không thua kém so với không nén bộ tham số.

Về mặt hiệu quả, mô hình học liên kết cho tốc độ học nhanh hơn và có độ chính xác tổng quát cao hơn khi ta hiểu và cài đặt tham số hợp lí với từng bài toán.

Về mặt ứng dụng, mô hình học liên kết có thể phát triển thành hệ thống trí tuệ nhân tạo cho các tập đoàn, công ti lớn, để tăng tốc độ học khai thác tối đa tài nguyên của máy tính.

Về mặt bảo mật dữ liệu, đây là một phương pháp cực kì hữu ích cho bài toán tạo một mô hình học cần bảo mật dữ liệu.

❖ ***Tuyên bố về quyền lợi:*** *Các tác giả xác nhận hoàn toàn không có xung đột về quyền lợi.*

❖ ***Lời cảm ơn:*** *Nghiên cứu được tài trợ bởi Đại học Quốc gia Thành phố Hồ Chí Minh (ĐHQG-HCM) trong dự án NCM2019-18-01.*

TÀI LIỆU THAM KHẢO

Cenk Bircanoğlu, & Nafiz Arıca (2018). *A comparison of activation functions in artificial neural networks,* Bahcesehir Universitesi, Istanbul, TR.

Fanglin Li, Bin Wu, Liutong Xu, Chuan Shi, & Jing Shi (2014). *A fast distributed stochastic Gradient Descent Algorithm for Matrix Factorization,* Beijing Key Lab of Intelligent Telecommunication Software and Multimedia.

Jakub Konecny, H. Brendan McMahan, & Daniel Ramage (2016). *Federated Optimization: Distributed Machine Learning for On-Device Intelligence,* University of Edinburgh.

McMahan, Daniel Ramage (2017). *Federated Learning: Collaborative Machine Learning without Centralized Training Data.*

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Aguera y Arcas (2017). *Communication-efficient learning of deep networks from decentralized data,* Google, Inc., 651 N 34th St., Seattle, WA 98103 USA.

Saad Albawi, Tareq Abed Mohammed, & Saad Al-Zawi (2017). *Understanding of a convolutional neural network,* Department of Computer Engineering, Istanbul Kemerburgaz University, Istanbul, Turkey.

Siddharth Sharma, & Simone Sharma (2020). *Activation functions in neural networks,* Dept. of Computer Science and Engineering, Global Institute of Technology, Jaipur.

DATA PRIVACY-PRESERVING VIA IMPROVED FEDERATED LEARNING MODEL

Nguyen Thi Huong[1*], Bui Huy Toan[2], Le Tan Phong[2], Nguyen Dinh Thuc[2]

[1]Smartnet HCMC, Vietnam

[2]University of Science, Vietnam National University Ho Chi Minh City, Vietnam

[*]Corresponding author: Nguyen Thi Huong – Email: nguyenhuongk07@gmail.com

Received: March 02, 2021; Revised: March 18, 2021; Accepted: March 20, 2021

ABSTRACT

Data modeling is an important problem in data analysis. Machine learning is the most popular method to solve this modeling problem. All most of machine learning schemes are local learning schemes in which the training dataset is stored at a server, therefore it can't take advantage of the diversity of data shared from multiple sources. As a result, the generalization of the obtained model is limited. The federated learning is a learning from multi-source of data so it has many advantages compared to other methods. Federated learning model applies to a variety of data types and machine learning algorithms. Besides accuracy, this model also ensures privacy for the training data set. This paper proposes an improvement of the federated learning model to ensure privacy protection based on an federated-learning model. The experimental results show the feasibility which can be applied to problems using machine learning in practice and also open up challenges to improve research and innovation.

Keywords: differential privacy; federated learning; privacy-preserving data analysis; privacy-preserving with federated learning