SỬ DỤNG GRAPH NEURAL NETWORK ĐỂ PHÂN LOẠI VĂN BẢN TIẾNG ANH

Vũ Phú Lộc [1,*], Huỳnh Thị Cẩm Dung [1]
TS. Trần Khải Thiện[2], TS Nguyễn Thị Bích Ngân[3], TS. Phạm Nguyễn Huy Phương[4]
[1]Lớp cao học CNTT Khóa 01, Trường Đại học Công Thương Thành phố Hồ Chí Minh
[2]Khoa CNTT, Trường Đại học Ngoại Ngữ - Tin Học Thành phố Hồ Chí Minh
[3]Khoa Công nghệ Thông tin, Trường Đại học Công Thương Thành phố Hồ Chí Minh
[4]Phòng Quản lý Sau Đại Học, Trường Đại học Công Thương Thành phố Hồ Chí Minh
*Email: [1]vplghost@gmail.com, [1]huynhthicamdung2906@gmail.com,
[2]Thientk@huflit.edu.vn,[3]nganntb@huit.edu.vn, [4]Phuongpnh@huit.edu.vn
Ngày nhận bài … ; Ngày chấp nhận đăng: ../../2024

TÓM TẮT

Hiện nay có nhiều bài toán trong thực tế mà dữ liệu khi ánh xa qua đồ thị có cấu trúc phức tạp, không thuộc không gian Euclide, cũng không có hình thức cố định với kích thước và số chiều lớn. Chẳng hạn như: phát hiện thuốc giả, hệ thống thương mại điện tử cần phát hiện người bán, người mua sản phẩm độc hại, giải mã bộ gen hoặc khoa học vật liệu. Chính vì vậy việc nghiên cứu và xây dựng các thuật toán xử lý dữ liệu đồ thị trở thành một chủ đề quan trọng trong phương pháp học sâu và học máy. Trong bài báo này, chúng tôi nghiên cứu mô hình Graph neural networks (GNN) trong học sâu (deep learning) áp dụng trên dữ liệu đồ thị để phân loại văn bản Tiếng Anh ứng dụng trong bài toán phân lớp băn bản. Kết quả thực nghiệm trên bộ dữ liệu Movie Reviews[1] đã đạt đến độ chính xác lên đến 76,468% so với một số mô hình học sâu khác trong bài toán phân lớp văn bản được trình bày trong nghiên cứu.

Từ khoá: Học sâu, Máy học, Phân lại văn bản, Xử lý ngôn ngữ tự nhiên, Mạng nơ-ron, Deep Learning, Machine Learning, Natural language processing, Graph neural network, Graph convolutional network, Text classification

1. GIỚI THIỆU

Hiện nay dữ liệu đồ thị trở thành một phần không thể thiếu và áp dụng nhiều trong các lĩnh vực như: thị giác máy tính, nhận dạng mẫu, phân tích hình ảnh, xử lý ngôn ngữ tự nhiên, … Gần đây dữ liệu đồ thị còn được sử dụng để dự đoán cấu trúc phân tử protein bậc 3 và 4. Việc này có ý nghĩa quan trọng trong việc tìm ra các chất mới có khả năng tương tác sinh học ở mức độ phân tử và tế bào. Sự đặc biệt của đồ thị là không giới hạn về kích thước hay số chiều nên việc khai thác tiềm năng của deep learning trên dữ liệu đồ thị trở thành xu hướng mà các nhà nghiên cứu hướng tới. Tương tự trong các bài toán phân loại văn bản, có nhiều phương pháp phân loại phổ biến được sử dụng trong lĩnh vực học máy và xử lý ngôn ngữ tự nhiên (Natural language processing - NLP) như: Cây quyết định, Support Vector Machines (SVM), Logistic Regression, và Neural Networks trong đó có mô hình Graph Neural Networks,…. Cây quyết định, Support Vector Machines (SVM), và Logistic Regression là những mô hình phân loại truyền thống trong học máy, trong khi Neural Networks (NN) là một mô hình phân loại dựa trên mạng nơ-ron. Sự khác biệt giữa các mô hình này dẫn đến hiệu suất khác nhau trong việc phân loại dữ liệu. Đối với với mạng nơ-ron thì có khả năng học và trích xuất các đặc trưng từ dữ liệu một cách tự động thông qua việc học sâu, trong khi các mô hình truyền thống thường yêu cầu việc trích xuất đặc trưng thủ công. Ngoài ra mạng nơ-ron có thể được xây dựng và tinh chỉnh dễ dàng để phù hợp với nhiều loại dữ liệu và bài toán khác nhau. Nó có thể kết hợp với các lớp và kiến trúc khác nhau để cải thiện hiệu suất.

[1] https://www.cs.cornell.edu/people/pabo/movie-review-data

Graph Neural Networks (GNNs) là mô hình mạng nơ-ron có giám sát. Mỗi nơ-ron trong GNN tương ứng với một nút trong đồ thị, các nơ-ron được kết nối với nhau theo cách kết nối của các nút trong đồ thị. Nên GNN hiệu quả trong việc xử lý và phân tích cấu trúc phức tạp và mối quan hệ trong dữ liệu đồ thị, bởi vì nó có khả năng cập nhật trạng thái của mỗi nút dựa trên trạng thái của các nút liên kết với nó. Chính vì vậy trong bài này chúng tôi tiếp cận việc tìm hiểu, giới thiệu các khía cạnh của GNN, nghiên cứu về sức mạnh biểu đạt và tính linh hoạt của GNN cũng như khám phá cách chúng có thể thay đổi cách chúng ta hiểu và xử lý dữ liệu đồ thị trong tương lai, đề xuất và xây dựng ứng dụng phân loại văn bản tiếng Anh áp dụng với mạng neuron đồ thị [1].

2. PHƯƠNG PHÁP ĐỀ XUẤT

2.1 Tổng quan phương pháp

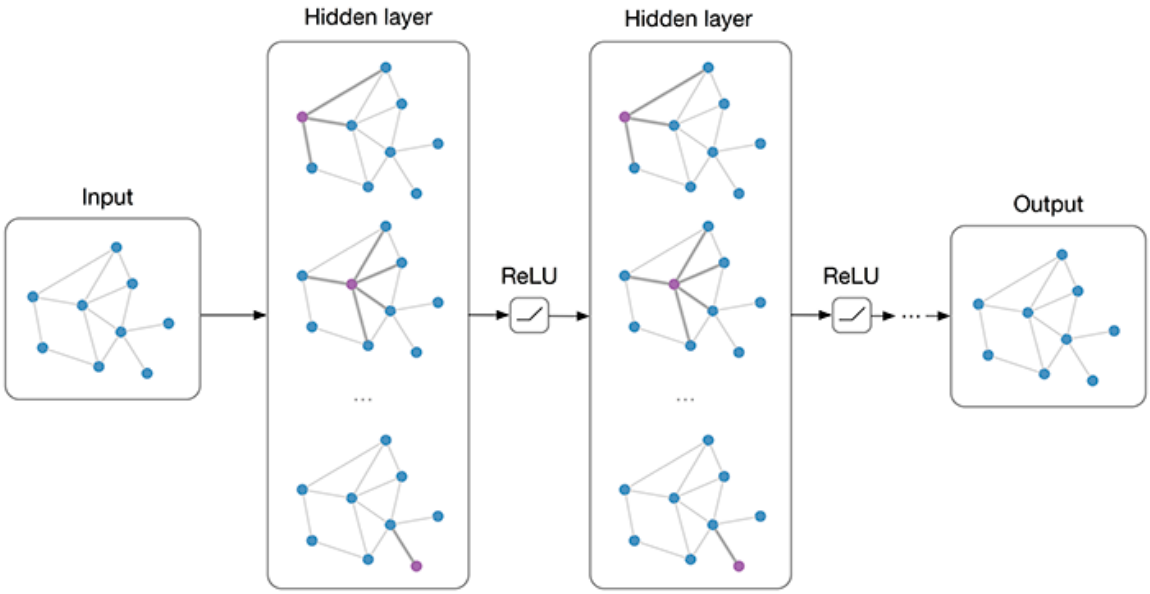
Trong bài viết nghiên cứu chúng tôi sẽ sử dụng mô hình Graph Neural Network để phân loại cảm xúc trong tập dữ liệu Movie Reviews (Pang và Lee 2005) [2] là một tập dữ liệu thường được sử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên và học máy. Đầu vào của cả hai mô hình là dữ liệu văn bản. Tập dữ liệu này chứa các đánh giá và bình luận từ người dùng về các bộ phim, và kết quả đầu ra được chia thành hai loại cảm xúc chính là tích cực (Positive) và tiêu cực (Negative). Đồng thời chúng tôi sẽ so sánh kết quả thực nghiệm GNN với các phương pháp phân loại văn bản phổ biến khác và rút ra kết luận.

2.2 Cấu trúc mô hình mạng

2.2.1. Graph convolutional network(GCN)

Mạng GCN (Graph Convolutional Network) là một dạng của mạng nơ-ron sử dụng để xử lý dữ liệu đồ thị, cho phép nó hiểu và khai thác thông tin phức tạp và mối quan hệ trong dữ liệu đồ thị, từ đó nâng cao hiệu suất và độ chính xác của các ứng dụng liên quan đến học sâu và trí tuệ nhân tạo. GCN được thiết kế để học cấu trúc và thông tin trong dữ liệu đồ thị bằng cách áp dụng phép tích chập (convolution) trên đồ thị. Đầu vào của GCN là một đặc trưng đỉnh (node feature matrix) ký hiệu là X với kích thước N x D, với N là số lượng đỉnh trong đồ thị và D là số chiều của đặc trưng và một ma trận kề (adjacency matrix) với kích thước N x N, biểu diễn mối quan hệ kề giữa các đỉnh, ký hiệu là A. Cấu trúc cơ bản của mô hình GCN [3] [4] [5] gồm có các lớp và được biểu diễn ở hình 1:

- Lớp tích chập đồ thị (Graph Convolution Layer): cho phép nó truyền thông tin từ các đỉnh láng giềng đến các đỉnh khác trong đồ thị
  - Tính toán đặc trưng mới cho mỗi đỉnh bằng cách tính trung bình có trọng số của đặc trưng của đỉnh láng giềng và đặc trưng của chính đỉnh đó.
  - Công thức tính toán: $H = \sigma(\hat{H} * W)$, với H là ma trận đặc trưng đỉnh mới, $\hat{H}$ là ma trận đặc trưng đỉnh mở rộng (được tính bằng cách nối thêm một cột của giá trị 1 vào X), W là ma trận trọng số và σ là hàm kích hoạt phi tuyến (ví dụ: ReLU).
- Kích hoạt phi tuyến (Non-linear Activation)
  - Áp dụng hàm kích hoạt phi tuyến (ví dụ: ReLU) cho các phần tử của ma trận đặc trưng đỉnh mới H.
- Lặp lại các lớp tích chập đồ thị
  - Quá trình tính toán lớp tích chập đồ thị và kích hoạt phi tuyến được lặp lại cho một số lớp để lấy thông tin từ các cấp độ xa hơn trong đồ thị.
- Đầu ra
  - Ma trận đặc trưng nút cuối cùng, thường được sử dụng cho các tác vụ như phân loại, dự đoán hoặc nhận dạng.

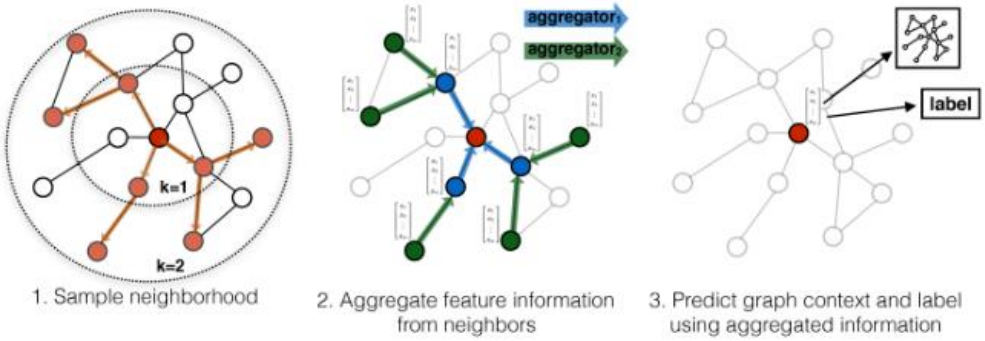


Hình 1 Cấu trúc của mô hình GCN [6]

2.2.2. GraphSage

Điểm đặc biệt của GraphSage (Graph Sample and Aggregated) so với các phương pháp truyền thống khác là khả năng học từ các đồ thị mới mà mô hình chưa từng gặp, đây là tính chất gọi là "inductive learning". GraphSage sử dụng một quá trình lấy mẫu chỉ lấy ngẫu nhiên tập hàng xóm có kích thước cố định chứ không phải lấy toàn bộ hàng xóm và tổng hợp thông tin từ các đỉnh hàng xóm của các đỉnh và sau đó cập nhật đặc trưng cho mỗi đỉnh. Có thể hiểu mô hình GraphSage được xây dựng cũng dựa trên ý tưởng là tổng hợp thông tin từ các nút lân cận có thể biểu diễn đồ thị tổng thể và khả năng xử lý các đồ thị lớn. Chi tiết mô hình được minh họa ở hình 2 như sau [7]:

Tuy nhiên, với dữ liệu đồ thị, việc sử dụng quá nhiều aggre function có thể làm giảm hiệu suất của mô hình trên một số trường hợp.

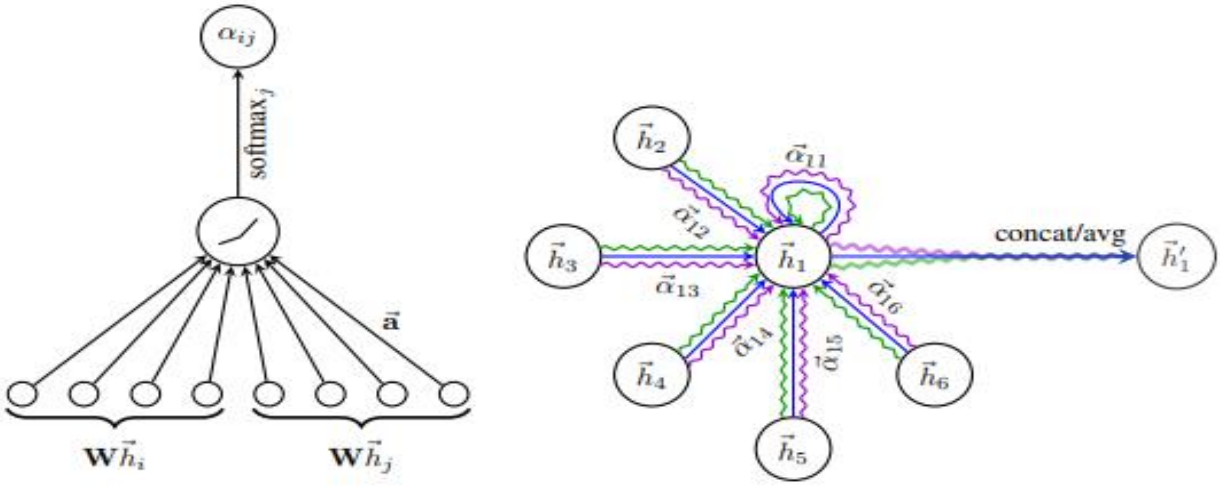


Hình 2 Minh họa trực quan GraphSage và cách tổng hợp thông tin từ các nút lân cận [7]

2.2.3. Graph Attention Network (GAT)

Việc sử dụng mô hình GCN sẽ làm giảm độ chính xác ở một số bài toán có dữ liệu mà trong đó các nút có mức độ ảnh hưởng khác nhau tới các node còn lại và mô hình, và Graph Attention Network (GAT) sẽ giải quyết vấn đề này khi xét tới mức độ quan trọng các các nút hàng xóm. Graph Attention Network (GAT) là một kiến trúc mạng thần kinh sâu được thiết kế để học biểu diễn vector cho các đỉnh trong đồ thị. Điểm đặc biệt của GAT là khả năng tự chú ý (self-attention) để xác định mức độ quan trọng của các nút láng giềng trong quá trình tính toán. Cho phép nó tập trung vào các đỉnh quan trọng nhất và bỏ qua những nút ít quan trọng hơn trong quá trình tính toán. Vì sử dụng trọng số cố định cho tất cả các nút láng giềng, GAT tự động học trọng số chú ý cho mỗi nút láng giềng dựa trên đặc trưng của nút hiện tại và nút láng giềng đó. GAT cho phép mô hình hóa các mối quan hệ phi tuyến và phức tạp giữa các nút trong đồ thị. Nó đã được chứng minh là hiệu

quả trong nhiều nhiệm vụ xử lý đồ thị, bao gồm phân loại đồ thị, dự đoán liên kết, phân cụm đồ thị và nhiều ứng dụng khác và mô tả GAT được biểu diễn ở hình 3.



Hình 3 Mô tả GAT

2.2.4. Hàm mất mát (Loss function)

Hàm mất mát (Loss function) là một thành phần quan trọng trong quá trình huấn luyện của mô hình máy học và deep learning, đóng vai trò đánh giá đánh giá mức độ sai lệch giữa giá trị dự đoán và giá trị thực tế của mô hình với một bộ trọng số tương ứng. Mục đích của quá trình huấn luyện là tìm ra bộ số để độ lớn hàm mất mát (loss function) là nhỏ nhất (cực tiểu). Trong bài báo này sử dụng hàm CrossEntropyLoss làm hàm mất mát cho mô hình các mô hình GNN.

2.2.5. Optimizer

Chúng tôi sử dụng thuật toán Adam như một phương pháp tối ưu hóa để cực tiểu hóa hàm mất mát. Adam là một thuật toán tối ưu hóa mạnh mẽ và hiệu quả, được rất nhiều trong cộng đồng học sâu sử dụng, đặc biệt trong bài toán phân loại cảm xúc văn bản.

2.3 Các mô hình phân lớp văn bản phổ biến được so sánh trong nghiên cứu

2.3.1. Logistic Regression

Logistic Regression là một thuật toán phân loại, học có giám sát trong Machine Learning. Mục tiêu của Logistic Regression là dự đoán xác suất của một mẫu thuộc về một lớp cụ thể, thường được sử dụng trong các tình huống mà đầu ra mong muốn là một biến phân loại như: "Có/Không", "Đúng/Sai", "Pass/Fail",…

Regression được sử dụng khi biến phụ thuộc là một biến rời rạc hoặc nhị phân (binary) và các biến độc lập có thể là các biến liên tục hoặc rời rạc [8].

Thuật toán Logistic Regression sử dụng hàm sigmoid (hay hàm logistic) để chuyển đổi đầu vào thành một giá trị xác suất nằm trong khoảng từ 0 đến 1 thể hiện xác suất dự đoán cho một lớp. Giá trị đầu ra của hàm sigmoid thường được coi như xác suất của một quan sát thuộc về lớp tích cực (ví dụ: 1) so với lớp tiêu cực (ví dụ: 0).

2.3.2. LinearSVC

LinearSVC là thuật toán Support Vector Machines (SVM) với hạt nhân tuyến tính (linear kernel). LinearSVC được sử dụng để giải quyết bài toán phân loại nhị phân và đa lớp. Mô hình này phù hợp cho các tình huống mà việc tìm kiếm một siêu phẳng tuyến tính trong không gian đặc trưng có thể phân tách một cách rõ ràng giữa các lớp dữ liệu. Nó xây dựng một mô hình phân loại

dựa trên việc tìm một siêu phẳng (hyperplane) trong không gian đặc trưng, để phân tách các mẫu thuộc về các lớp khác nhau [9].

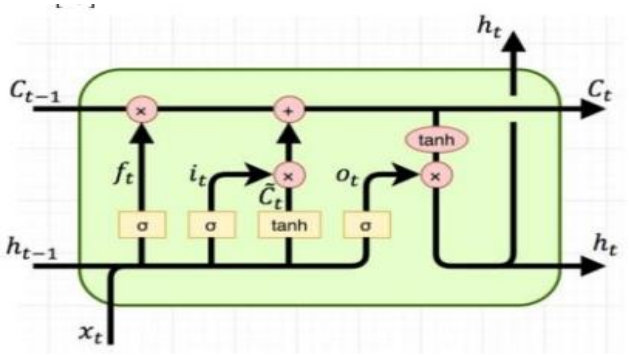
2.3.3. Cây quyết định

Cây quyết định là một cấu trúc dữ liệu có hình dạng cây dùng để phân lớp các đối tượng dựa vào các dãy luật, trong đó mỗi nút đại diện cho một câu hỏi hoặc một quyết định, và mỗi nhánh đại diện cho một kết quả khả thi. Và cây quyết định là một thuật toán học máy giám sát được sử dụng cho cả bài toán phân loại và hồi quy. Thuật toán DecisionTreeClassifier sử dụng cây quyết định để xây dựng một mô hình phân loại dựa trên các quyết định tại các nút của cây [10].

2.3.4. Long Short Term Memory (LSTM)

Mạng Long Short Term Memory (LSTM) làm một kiến trúc đặc biệt của RNN (Recurrent Neural Network) có khả năng học được sự phục thuộc trong dài hạn (long-term dependencies) được giới thiệu bởi Hochreiter & Schmidhuber (1997). LSTM có các kết nối phản hồi, có thể xử lý dữ liệu dạng hình ảnh và toàn bộ chuỗi dữ liệu ví dụ như giọng nói hoặc video. LSTM đã khắc phục được rất nhiều những hạn chế của RNN trước đây về mất đạo hàm (vanishing gradient).

LSTM có cấu trúc dạng chuỗi các nút mạng như RNN, nhưng cấu trúc bên trong thì lại phức tạp hơn, bao gồm 4 tầng tương tác với nhau (Hình 4). Điểm đặc biệt của mạng LSTM nằm ở trạng thái ô C (cell state), nơi lưu trữ các trọng số dài hạn của mô hình. Các thông số trạng thái ô C được kiểm soát bởi hàm sigmoid (trong một tầng gọi là tầng ẩn h (hidden state), đầu vào tại thời điểm t là x(t) được đưa vào nút mạng. Sau khi được xử lý qua các hàm kích hoạt sigmoid σ, và các phép toán véc–tơ, kết quả đầu ra là trạng thái ô C và trạng thái ẩn h tại thời điểm t sẽ được sử dụng cho nút mạng t+1 tiếp theo [11]. Khác với mang RNN tiêu chuẩn, LSTM không chỉ có một tầng đóng mà có tới 4 tầng ẩn (3 sigmoid và 1 tanh) tương tác với nhau như hình 4:



Hình 4 Cấu trúc một nút mạng trong mạng LSTM.

2.4 Các bước thực hiện

Chúng tôi đã thực nghiệm các mô hình GNN như GCN, GAT và GraphSage, đồng thời thực nghiệm các mô hình khác như LSTM, Logistic Regression, LinearSVC, Cây quyết định để tiến hành phân lớp văn bản trên tập dữ liệu MR và tiến hành so sánh kết quả giữa các mô hình GNN với các mô hình phân loại văn bản phổ biến khác và rút ra kết luận.

Quá trình thực hiện gồm 5 bước chính như hình 5:

1. Chuẩn bị và tiền xử lý dữ liệu
2. Chuẩn hóa dữ liệu
3. Tạo model, optimize, metric
4. Đào tạo mô hình
5. Đánh giá mô hình

Chi tiết các bước thực hiện như sau:

Bước 1: Chuẩn bị và tiền xử lý dữ liệu

- Input: Tập dữ liệu văn bản MR chưa được xử lý.

- Output: Dữ liệu văn bản đã được tiền xử lý, làm sạch và chuẩn hóa sẵn sàng cho các bước tiếp theo.

Hình 5 Các bước thực hiện thực nghiệm

Bước 2: Chuẩn hóa dữ liệu

Input: Dữ liệu văn bản sau khi tiền xử lý.

Output: Dữ liệu văn bản đã được chuẩn hóa, có thể là vector hoặc dạng ma trận để đưa vào các mô hình máy học.

Bước 3: Tạo model, optimize, metric

Input: Dữ liệu văn bản đã được chuẩn hóa.

Output: Các mô hình máy học đã được khởi tạo, được tối ưu hóa bằng các thuật toán tối ưu hóa như Adam và sử dụng các chỉ số (metrics) như độ chính xác, F1-score để đánh giá hiệu suất.

Bước 4: Đào tạo mô hình

Input: Mô hình máy học đã được khởi tạo và dữ liệu huấn luyện.

Output: Các mô hình máy học đã được huấn luyện trên dữ liệu và thu được các tham số tối ưu.

Bước 5: Đánh giá mô hình

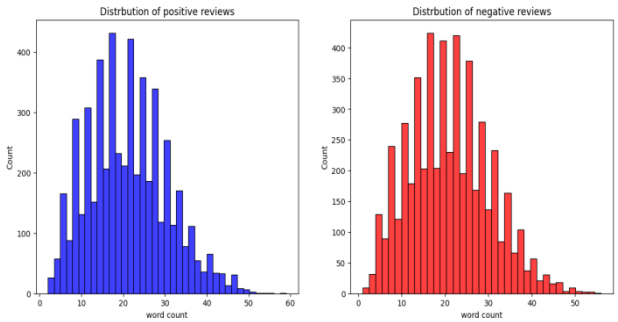
Input: Các mô hình máy học đã được huấn luyện.

Output: Kết quả đánh giá và so sánh hiệu suất của các mô hình với nhau dựa trên các chỉ số như độ chính xác, F1-score để rút ra kết luận về hiệu suất của từng mô hình và so sánh chúng với nhau.

3. THỰC NGHIỆM

3.1 Tập dữ liệu

Bài báo tiến hành thực nghiệm và đánh giá trên tập dữ liệu Movie Review (MR). Bộ dữ liệu chứa các đánh giá về phim để phân loại cảm xúc, trong đó mỗi đánh giá chỉ chứa một câu (Pang và Lee 2005)[2]. Dữ liệu gồm có 5.331 đánh giá tích cực và 5.331 đánh giá tiêu cực. Nghiên cứu đã sử dụng phân tách huấn luyện / kiểm thư trong github của bài báo PTE: Nhúng văn bản dự đoán thông qua mạng văn bản không đồng nhất quy mô lớn (ng, Qu và Mei 2015)[3] biểu diễn như hình 5 bên dưới.



Hình 5 Phân phối 2 lớp trong dữ liệu MR

3.2 Môi trường thực nghiệm

[2] https://www.cs.cornell.edu/people/pabo/movie-review-data/

[3] https://github.com/mnqu/PTE/tree/master/da/mr

Trong bài báo này chúng tôi thực nghiệm dựa trên các môi trường và thư viện sau:

- Google Colab: làm môi trường để chạy thực nghiệm cho các mô hình [13]
- Framework Pytorch: cung cấp các module và lớp để xây dựng và huấn luyện thực nghiệm cho mô hình GNN [14].
- Thư viện DGL: cung cấp các công cụ cho việc giải quyết các bài toán liên quan đến đồ thị như phân loại đồ thị, dự báo đồ thị và nhúng đồ thị [15]
- Thư viện Sklearn: xây dựng và huấn luyện mô hình, các mô hình phân lớp truyền thống khác như LinearSVC, Logistic Regession… để tiến hành so sánh với phương pháp áp dụng mạng nơ ron đồ thị [16].

3.3 Chi tiết thực nghiệm

3.3.1.Text-GCN

Với Text GCN, kích thước của nhúng trong lớp chập ban đầu được thiết lập ở mức 200 và window-size được cài đặt là 20. Learning-rate của mô hình định ở 0,02 và drop-out ở 0,5 [3]. Mô hình Text GCN được huấn luyện trong khoảng tối đa 200 epochs, với việc sử dụng thuật toán tối ưu Adam. Một tiêu chí "early stopping" được kích hoạt khi mất mát không giảm suốt 10 epochs liên tiếp. Đối với các mô hình cơ bản, việc áp dụng nhúng từ đã được huấn luyện trước với 300 chiều từ bộ nhúng GloVe [12].

Kết quả:

Bảng 2. Kết quả thực nghiệm đối với Tex-GCN

	precision	recall	f1-score
Positive	0.7629	0.7642	0.7632
Negative	0.7638	0.7642	0.7632
accuracy			0.7634
macro avg	0.7634	0.7634	0.7634
weighted avg	0.7634	0.7634	0.7634

3.3.2. GraphSage

Giống như Text-GCN và GraphSage, trong mô hình này, kích thước nhúng của lớp chập ban đầu được thiết lập là 200 và window-size được cấu hình là 20. Learning-rate được định nghĩa ở mức 0,02 và drop-out ở mức 0,5 [3]. Mô hình được huấn luyện trong tối đa 200 epochs, với việc sử dụng thuật toán tối ưu Adam. Một tiêu chí "early stopping" được kích hoạt nếu mất mát (loss) không giảm trong 10 epochs liên tiếp. Đối với các mô hình cơ bản, việc sử dụng nhúng từ đã được huấn luyện trước với 300 chiều từ bộ nhúng GloVe [12].

Kết quả:

Bảng 2. Kết quả thực nghiệm đối với GraphSage

	precision	recall	f1-score
Positive	0.7567	0.7648	0.7607
Negative	0.7622	0.7541	0.7581
accuracy			0.7594
macro avg	0.7594	0.7594	0.7594
weighted avg	0.7594	0.7594	0.7594

3.3.3. GAT

Tương tự với Text-GCN, GraphSage, mô hình đặt kích thước nhúng của lớp chập đầu tiên là 200 và đặt window-size là 20. Learning-rate là 0,02, drop-out là 0,5 [3],mô hình đào tạo cho tối đa 200 kỷ nguyên, optimize Adam. Mô hình áp dụng chỉ số early stopping nếu loss không giảm trong

10 epoch liên tiếp. Đối với đường cơ sở các mô hình sử dụng nhúng từ được đào tạo trước đã sử dụng 300- chiều nhúng từ GloVe [12]

Kết quả:

Bảng 3. Kết quả thực nghiệm đối với GAT

	precision	recall	f1-score
Positive	0.7630	0.7681	0.7656
Negative	0.7666	0.7614	0.7640
accuracy			0.7648
macro avg	0.7648	0.7648	0.7648
weighted avg	0.7648	0.7648	0.7648

3.3.4. LSTM

Kết quả:

Bảng 4. Kết quả thực nghiệm đối với LSTM

	precision	recall	f1-score
Positive	0.7253	0.7724	0.7481
Negative	0.7440	0.6934	0.7178
accuracy			0.7338
macro avg	0.7347	0.7329	0.733
weighted avg	0.7344	0.7338	0.7333

3.3.5. Logistic Regression

Chúng tôi đào tạo mô hình Logistic Regression sử dụng thư viện sklearn:

lr = LogisticRegression()

lr.fit(x_train, y_train)

Kết quả:

Bảng 5. Kết quả thực nghiệm đối với Logistic Regression

	precision	recall	f1-score
Positive	0.7426	0.7608	0.7516
Negative	0.7618	0.7436	0.7526
accuracy			0.7521
macro avg	0.7522	0.7522	0.7521
weighted avg	0.7523	0.7521	0.7521

3.3.6. LinearSVC

Chúng tôi đào mô hình LinearSVC sử dụng thư viện sklearn:

lsvm = LinearSVC()

lsvm.fit(x_train, y_train)

Kết quả:

Bảng 6. Kết quả thực nghiệm đối với LinearSVC

	precision	recall	f1-score
Positive	0.7438	0.7523	0.748
Negative	0.7565	0.7481	0.7522
accuracy			0.7501
macro avg	0.7501	0.7502	0.7501
weighted avg	0.7502	0.7501	0.7502

3.3.7. Cây quyết định

Chúng tôi đào tạo mô hình DecisionTreeClassifier sử dụng thư viện sklearn:

dt = DecisionTreeClassifier()

dt.fit(x_train, y_train)

Kết quả:

Bảng 7. Kết quả thực nghiệm đối với Cây quyết định

	precision	recall	f1-score
Positive	0.645	0.6336	0.6392
Negative	0.6498	0.6609	0.6553
accuracy			0.6474
macro avg	0.6474	0.6472	0.6473
weighted avg	0.6474	0.6474	0.6474

3.3.8.Nhận xét

Model	class	precision	recall	f1-score	text_acc
Text- GCN	positive	0.7629	0.7642	0.7636	0.7634
	negative	0.7638	0.7625	0.7632	
GAT	**positive**	**0.7630**	**0.7681**	**0.7656**	**0.7648**
	negative	**0.7666**	**0.7614**	**0.7640**	
GATSage	positive	0.7567	0.7648	0.7607	0.7594
	negative	0.7622	0.7541	0.7581	
LSTM	positive	0.7253	0.7724	0.7481	0.7338
	negative	0.7440	0.6934	0.7178	
Logistic Regression	positive	0.7426	0.7608	0.7516	0.7521
	negative	0.7618	0.7436	0.7526	
LinearSVC	positive	0.7438	0.7523	0.748	0.7501
	negative	0.7565	0.7481	0.7522	
DecisionTreeClassifier	positive	0.645	0.6336	0.6392	0.6474
	negative	0.6498	0.6609	0.6553	

Bảng 8. Kết quả thực nghiệm 8 model trên tập dữ liệu MR

Kết quả đánh giá hiệu suất của các mô hình trong nghiên cứu này cho thấy cả hai mô hình đã được áp dụng đều cho kết quả ấn tượng. Mô hình Graph Attention Network (GAT) đã đạt đến độ chính xác lên đến 76,468% trên tập test, đồng thời f1-score của mô hình này cũng cao trên cả hai lớp 'Positive' và 'Negative'. Các mô hình mạng GNN khác như Text-GCN và GATSage cũng cho thấy hiệu suất ổn định, với độ chính xác vượt qua ngưỡng 70%. Ngoài ra, các mô hình Logistic Regression và SVC cũng thể hiện sự hiệu quả với độ chính xác lần lượt trên 75%.

4. KẾT LUẬN

Mạng nơ-ron đồ thị (Graph Neural Network - GNN) là một lĩnh vực nghiên cứu đang phát triển mạnh mẽ trong lĩnh vực học máy và khai thác dữ liệu đồ thị. GNN có khả năng mô hình hóa thông tin đồ thị và đạt được hiệu suất tốt trong nhiều bài toán phân loại, dự đoán và xử lý trên đồ

thị. Trong nghiên cứu này chúng tôi đã thực nghiệm và so sánh 8 mô hình trong bài toán phân lớp cảm xúc trong văn bản trong bộ dữ liệu MR. Qua thực nghiệm, ta có thể thấy rằng mô hình mạng nơ ron đồ thị hoạt động mang lại kết quả khá tốt với mô hình Graph Attention Network(GAT) với tỷ lệ chính xác 76,5% - cao nhất trong 8 mô hình.

Trong thời gian tới, chúng tôi sẽ tập trung vào việc nâng cao chất lượng mô hình bằng các phương pháp mới, những tập dataset lớn hơn, nghiên cứu sâu hơn ứng dụng GNN vào phân lớp văn bản tiếng Việt. Đồng thời tìm hiểu một số phương pháp phát triển kiến trúc GNN để GNN có thể xử lý ở độ sâu dữ liệu lớn hơn. Điều này có thể bao gồm việc tạo ra các kiến trúc GNN có khả năng mô hình mô hình hóa thông tin đồ thị trên nhiều tầng, cũng như các phương pháp chuyển đổi dữ liệu để giảm hiện tượng mất mát thông tin khi truyền qua nhiều tầng.

TÀI LIỆU THAM KHẢO

[1] Giới thiệu về Graph Neural Networks (GNNs) https://viblo.asia/p/gioi-thieu-ve-graph-neural-networks-gnns-yZjJYG7MVOE, truy cập 03/04/2024.

[2] Movie Review Data https://www.cs.cornell.edu/people/pabo/movie-review-data/, truy cập 01/04/2024

[3] M. W. Thomas N. Kipf, "Semi-Supervised Classification with Graph Convolutional Networks," ICLR 2017, 2017.

[4] B. Q. Manh, "Tản mạn về Graph Convolution Networks" https://viblo.asia/p/tan-man-ve-graph-convolution-networks-phan-1-6J3Zga8A5mB, truy cập 03/04/2024.

[5] M. W. Thomas N. Kipf, "Semi-Supervised Classification with Graph Convolutional Networks," in ICLR 2017, 2016.

[6] [Deep Learning] Graph Neural Network - A literature review and applications https://viblo.asia/p/deep-learning-graph-neural-network-a-literature-review-and-applications-6J3ZgP0qlmB, truy cập 03/04/2024.

[7] R. Y. J. L. William L. Hamilton, "Inductive Representation Learning on Large Graphs," NIPS 2017, 2017.

[8] V. H. Tiep, "Logistic Regression," https://machinelearningcoban.com/2017/01/27/logisticregression/, truy cập 02/04/2024

[9] "sklearn.svm.LinearSVC," https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html, truy cập 01/04/2024

[10] Cây Quyết Định (Decision Tree), https://trituenhantao.io/kien-thuc/decision-tree/, truy cập 31/03/2024

[11] F. C. S. F. J. B. D. W. J. T. F. G. M. G. G. Felix Gers, "Understanding LSTM Networks," 27 08 2015, https://colah.github.io/posts/2015-08-Understanding-LSTMs/, truy cập 02/03/2024

[12] R. S. C. D. M. Jeffrey Pennington, "Glove: Global Vectors for Word Representation," Computer Science Department, Stanford University, Stanford, CA 94305, Stanford, 2014.

[13] Google Colab https://colab.research.google.com, truy cập 03/03/2024

[14] Pytorch https://pytorch.org, truy cập 03/03/2024

[15] Thư viện DGL https://www.dgl.ai, truy cập 03/03/2024

[16] Thư viện Sklearn Thư viện Sklearn, truy cập 03/03/2024

ABSTRACT

USE GRAPH NEURAL NETWORK TO CLASSIFY ENGLISH TEXTS

Vũ Phú Lộc [1,*], Huỳnh Thị Cẩm Dung [1]
TS. Trần Khải Thiện[2], TS Nguyễn Thị Bích Ngân[3], TS. Phạm Nguyễn Huy Phương[4]
[1]Master's degree IT course 01, HCMC University of Industry and Trade
[2]Information Technology, HCMC University of Foreign Languages – Information Technology
[3]Information Technology, HCMC University of Industry and Trade
[4]Department of Graduate Management, HCMC University of Industry and Trade
*Email: [1]vplghost@gmail.com, [1]huynhthicamdung2906@gmail.com, [2]Thientk@huflit.edu.vn, [3]nganntb@huit.edu.vn, [4]Phuongpnh@huit.edu.vn

ABSTRACT: Currently, there are many real-life problems where data when reflected through graphs has a complex structure. These structures do not belong to the Euclidean space, nor do they have fixed forms with large dimensions. Examples include detecting counterfeit drugs, e-commerce systems needing to identify harmful sellers or buyers, decoding genomes, or material science. Therefore, researching and developing algorithms to process graph data has become an important topic in deep learning and machine learning methodologies. In this paper, we investigate the Graph Neural Networks (GNN) model in deep learning applied to graph data to classify English texts, applied to the text classification problem. Experimental results on the Movie Reviews dataset achieved an accuracy of up to 76.468% compared to some other deep learning models in the text classification problem presented in the study.

Keywords: Deep Learning, Machine Learning, Natural language processing, Graph neural network, Graph convolutional network, Text classification

Tôi xin cam kết bài báo này chưa được đăng và gửi đăng ở bất kỳ tạp chí nào khác.

Tác giả:

Họ và tên: Huỳnh Thị Cẩm Dung

Số điện thoại liên hệ: 0966601459