Week 4 - Deep Learning for NLP

Ngày 20 tháng 9 năm 2025

Trong lĩnh vực Xử lý ngôn ngữ tự nhiên (NLP), Deep Learning đã tạo ra những bước ngoặc quan trọng. Trước đây, NLP chủ yếu dựa vào các phương pháp trích xuất đặc trưng thủ công từ ngôn ngữ hoặc các mô hình thống kê truyền thống. Tuy nhiên, sự ra đời của Deep Learning với các kiến trúc như **RNN, LSTM**, hay sau này là **Transformer** đã giúp máy tính học được những ngữ nghĩa phức tạp hơn từ nguồn dữ liệu khổng lồ. Nhờ đó, các ứng dụng NLP hiện đại như Machine Translation, Text Summarization, Chatbot, etc.. đã đạt được hiệu suất vượt trội so với trước đây. Đặc biệt, với sự ra đời của các Mô hình Ngôn ngữ lớn (LLMs) đã đưa NLP trở thành nền tảng cốt lõi trong nhiều lĩnh vực của đời sống và công nghệ ngày nay.

1 Hidden Layer

Hãy tưởng tượng Hidden Layer giống như nhà bếp ở trong nhà hàng:

- Đầu vào (Input Layer) sẽ là các nguyên liệu
- Đầu bếp (Hidden Layer) sẽ sơ chế, nấu nướng theo nhiều công đoạn (nhưng ta không nhìn thấy hết bên trong)
- Đầu ra (Output Layer) sẽ là món ăn hoàn chỉnh được bày biện.

Lớp ẩn (Hidden Layer) là nơi mạng nơ-ron học cách tự động trích xuất các đặc trưng và mẫu từ dữ liệu. Số lượng các lớp ẩn và số nơ-ron trong mỗi lớp ảnh hưởng đến khả năng học và độ phức tạp của mô hình. Càng nhiều lớp ẩn, mạng càng có thể giải quyết các bài toán phức tạp hơn, nhưng cũng có nguy cơ bị quá khớp (overfitting).

Liên hệ với các khái niệm khác:

- Gắn liền với Perceptron vì Hidden Layer được tạo từ nhiều Perceptron
- Các trọng số của các unit trong hidden layer có thể được được cập nhật bằng Gradient Descent.

2 Perceptron

Perceptron là một đơn vị xử lý thông tin rất nhỏ, giống như một tế bào não. Nó nhận thông tin từ nhiều tế bào não khác, mỗi thông tin có một *mức độ quan trọng* riêng gọi là trọng số. Từ đó, nó sẽ tổng hợp tất cả thông tin lại và đưa ra quyết định, ví dụ: "có" hoặc "không"

Trọng số: Tưởng tượng bạn đang quyết định xem có nên đi xem phim hay không. "Phim được đánh giá cao" có thể là một thông tin quan trọng, trong khi "Thời tiết đẹp" có thể có trọng số nhỏ hơn. Trọng số cho biết thông tin nào quan trọng để đưa ra quyết định cuối cùng để đưa ra kết quả "có đi" hoặc "không đi".

Liên hệ với các khái niệm khác:

- Là thành phần cơ bản tạo nên Hidden Layer và toàn bộ mạng nơ-ron.
- Khái niệm Perceptron đặt nền móng cho Multi-Layer Perceptron (MLP), một dạng mạng nơ-ron phổ biến.
- Liên quan chặt chẽ đến Activation Function (ReLU, Sigmoid, Tanh) để giúp mô hình học được các quan hệ phi tuyến.

3 Memory-based Learning

Tưởng tượng bạn đang học cách nhận biết các loài động vật. Thay vì học các đặc điểm của chúng trông như thế nào, bạn chỉ cần nhớ lại tất cả các con vật mà bạn từng thấy. Khi bạn gặp con vật mới, bạn chỉ cần so sánh nó với tất cả con vật đã nhớ để xem nó giống với con vật nào nhất và gọi tên nó.

Ngược lại, trong **Model-based Learning**, bạn sẽ xây dựng một mô hình hoặc bộ quy tắc tổng quát từ dữ liệu đã quan sát.

Liên hệ với các khái niệm khác:

- Có liên hệ với thuật toán K-Nearest Neighbors (KNN), vì cũng dựa trên so sánh với dữ liệu đã quan sát.
- Trái ngược với Model-based Learning, vốn xây dựng mô hình tổng quát thay vì chỉ ghi nhớ.
- Có thể kết hợp với Distance Metrics (Euclidean, Cosine) để đo độ giống nhau giữa các mẫu.

4 Gradient Descent

Hãy tưởng tượng ban đang ở một đỉnh núi với sương mù dày đặc, bây giờ bạn muốn đi xuống đáy thung lũng. Mặc dù bạn không thể nhìn thấy đường đi nhưng bạn có thể cảm nhận được độ dốc dưới chân, chứng tỏ và đang đi đúng hướng. Gradient Descent giống như việc bạn đi từng bước nhỏ và đi theo hướng dốc xuống. Cứ như vậy, bạn sẽ dần đến được điểm thấp nhất của thung lũng.

- Gradient Descent được sử dụng để tối ưu Loss Function.
- Là cơ chế học chính trong các mô hình Model-based Learning như mạng nơ-ron.
- Có nhiều biến thể: Stochastic Gradient Descent (SGD), Mini-batch GD, Adam, giúp cải thiện tốc độ hội tụ.

5 Loss Function

Hãy tưởng tượng bạn đang chơi trò đoán số. Quản trò sẽ chọn 1 số bí mật từ 1 đến 100, và bạn sẽ cố gắng đoán.

- Loss Function ở đây giống như việc bạn đo lường mức độ sai. Ví dụ, nếu số bí mật là 50 và bạn đoán 70, bạn có thể tính mức độ “sai"là 20. Nếu bạn của bạn đoán 55, mức độ "sai"chỉ là 5.
- Mục tiêu của bạn của bạn là làm sao cho mức độ "sai"này (giá trị của Loss Function) càng nhỏ càng tốt

Trong học máy, Loss Function hoạt động tương tự. Nó đo lường sự khác biệt giữa kết quả dự đoán của mô hình và kết quả thực tế. Giá trị này sẽ được sử dụng để điều chỉnh mô hình trong quá trình học.

Liên hệ với các khái niệm khác:

- Đóng vai trò then chốt trong việc cập nhật tham số bằng Gradient Descent.
- Liên quan đến Evaluation Metrics (Accuracy, F1-score) vì cả hai đều đo lường hiệu suất nhưng ở các mức độ khác nhau.
- Các bài toán khác nhau cần Loss Function khác nhau: Cross-Entropy (classification), MSE.

6 Bài tập áp dụng

6.1 Minh họa Gradient Descent

Xét hàm số:

$$f(x) = 3x^2 + 5x - 7.$$

Mục tiêu là tìm điểm cực tiểu của hàm $f(x)$ bằng thuật toán Gradient Descent.

Trong thuật toán Gradient Descent, ta khởi tạo một giá trị x_0, sau đó cập nhật theo quy tắc:

$$x_{k+1} = x_k - \eta \cdot f'(x_k),$$

trong đó η là *learning rate* (tốc độ học), và $f'(x)$ là đạo hàm của hàm mục tiêu.

Để đảm bảo thuật toán dừng đúng lúc, ta sử dụng hai tiêu chí:

- **Điều kiện hội tụ**: thuật toán dừng khi $|f'(x_k)| < \epsilon$, tức là gradient đủ nhỏ (gần 0), coi như nghiệm đã hội tụ về cực tiểu. Ở đây $\epsilon > 0$ là ngưỡng rất nhỏ do người dùng lựa chọn.
- **Giới hạn số vòng lặp**: nếu sau một số lần lặp nhất định (*max iterations*) mà thuật toán vẫn chưa đạt điều kiện trên, thì dừng lại để tránh lặp vô hạn.

Như vậy, Gradient Descent kết thúc khi *một trong hai* tiêu chí trên được thỏa mãn, và giá trị x lúc này gần với cực tiểu của hàm $f(x)$.

Step	θ	$f(\theta)$	Gradient
1	10.000000	343.000000	65.000000
2	6.750000	163.437500	45.500000
3	4.475000	75.451875	31.850000
4	2.882500	32.338919	22.295000
5	1.767750	11.213570	15.606500
6	0.987425	0.862149	10.924550
7	0.441197	-4.210047	7.647185
8	0.058838	-6.695423	5.353029
9	-0.208813	-7.913257	3.747121
10	-0.396169	-8.509996	2.622984
⋮	⋮	⋮	⋮
43	-0.833330	-9.083333	0.000020
44	-0.833331	-9.083333	0.000014
45	-0.833332	-9.083333	0.000010

Bảng 1: Minh họa quá trình tìm cực tiểu của $f(x) = 3x^2 + 5x - 7$.

Vì đây là 1 hàm bậc 2 đơn giản nên ta có thể dễ dàng tính được cực tiểu của hàm số là -5/6, xấp xỉ với kết quả tìm được bằng thuật toán Gradient Descent

6.2 Thực hiện Text Classification bằng Neural Network

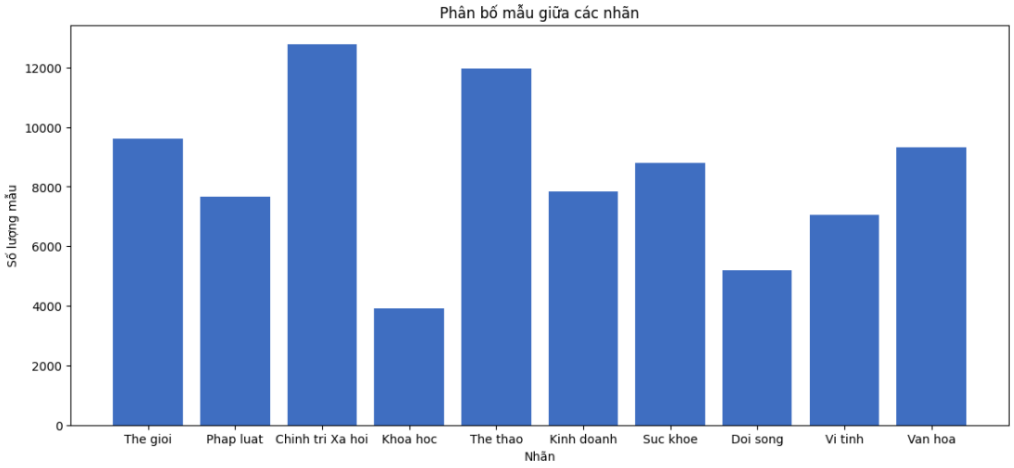
6.2.1 Dữ liệu sử dụng

Dataset sử dụng: https://github.com/duyvuleo/VNTC. Dataset gồm khoảng 50k samples ở tập train và 33k sample ở tập test. Em quyết định gộp train-test lại và chia lại theo tỷ lệ 8 train : 2 test để giống các phân bố thông thường được sử dụng với số lượng samples không quá lớn.

Qua biểu đồ số lượng mẫu, ta thấy hai class Đời sống và Khoa học có số lượng mẫu ít hơn đáng kể so với các class khác. Do vậy, mô hình Neural Network có xu hướng dự đoán kém chính xác hơn cho hai class này, dẫn đến Precision, Recall và F1-score thấp hơn so với các class còn lại

6.2.2 Thực nghiệm

Trong thực nghiệm, em sử dụng vocabulary size là 10,000, embedding dimension là 200, 5 epoch và learning rate là 1e-3. Hiện tại các giá trị này được chọn dựa trên suy đoán, nhưng trong tương lai, có nhiều thời gian hơn em sẽ áp



Hình 1: Phân bố nhãn trong dataset

dụng các phương pháp tuning siêu tham số để tìm ra bộ tham số tối ưu, nhằm cải thiện hiệu suất của mô hình.

Classification Report:

	precision	recall	f1-score	support
Kinh doanh	0.88	0.92	0.90	783
The thao	0.98	0.98	0.98	1197
Doi song	0.78	0.77	0.78	519
Khoa hoc	0.81	0.83	0.82	392
Chinh tri Xa hoi	0.85	0.87	0.86	1279
Suc khoe	0.91	0.92	0.92	880
Vi tinh	0.93	0.92	0.92	704
The gioi	0.95	0.92	0.94	961
Van hoa	0.93	0.92	0.92	933
Phap luat	0.92	0.87	0.90	766
accuracy			0.90	8414
macro avg	0.89	0.89	0.89	8414
weighted avg	0.90	0.90	0.90	8414

(a) Không tiền xử lý văn bản

Classification Report:

	precision	recall	f1-score	support
Kinh doanh	0.89	0.90	0.90	783
The thao	0.99	0.97	0.98	1197
Doi song	0.75	0.77	0.76	519
Khoa hoc	0.82	0.79	0.81	392
Chinh tri Xa hoi	0.81	0.90	0.85	1279
Suc khoe	0.92	0.90	0.91	880
Vi tinh	0.94	0.92	0.93	704
The gioi	0.97	0.90	0.93	961
Van hoa	0.92	0.94	0.93	933
Phap luat	0.92	0.87	0.89	766
accuracy			0.90	8414
macro avg	0.89	0.89	0.89	8414
weighted avg	0.90	0.90	0.90	8414

(b) Có áp dụng tiền xử lý văn bản

Hình 2: So sánh hiệu suất

Kết quả so sánh giữa việc không tiền xử lý và có tiền xử lý văn bản cho thấy hiệu suất của mô hình gần như tương đương. Điều này chứng tỏ, trong trường hợp này, việc tiền xử lý không ảnh hưởng đáng kể đến hiệu suất của Neural Network.

Như dự đoán, hai class có số lượng mẫu ít là Đời sống và Khoa học có Precision, Recall và F1-score thấp hơn so với các class còn lại, phản ánh ảnh hưởng của sự mất cân bằng dữ liệu đến hiệu quả dự đoán.