XÂY DỰNG HÀM THẤT THOÁT CHO HIỆU SUẤT TỐI ƯU TRONG HỌC MÁY VI PHÂN

Lê Bích Phượng
Trường Đại học Mỏ - Địa chất
Email: lebichphuong@humg.edu.vn

TÓM TẮT

Trong học máy vi phân (differential machine learning), phương pháp luồng gradient ngẫu nhiên thường được sử dụng để tìm điểm "gần như cực tiểu" của hàm thất thoát (loss function), điều này tương ứng với việc tối ưu hóa thuật toán phân loại. Mặc dù hàm thất thoát đóng vai trò rất quan trọng trong quá trình này, nhưng cho đến nay, cơ sở lí thuyết cho các hàm thất thoát vẫn chưa được phát triển đầy đủ. Bài báo này nhằm đóng góp vào việc xây dựng cơ sở lí thuyết cho hàm thất thoát, cung cấp một khung lí thuyết chi tiết và có hệ thống hơn để hỗ trợ việc phát triển các phương pháp tối ưu hóa và phân loại hiệu quả hơn. Tác giả cũng trình bày các phân tích về cách hàm thất thoát ảnh hưởng đến hiệu suất của mô hình và đề xuất một số cải tiến trong việc thiết kế và sử dụng hàm thất thoát sao cho đạt được hiệu suất tối ưu. Những nghiên cứu này không chỉ giúp hiểu rõ hơn về bản chất của hàm thất thoát mà còn mở ra hướng đi mới cho việc ứng dụng học máy vi phân trong các bài toán thực tiễn. Qua đó, bài báo mong muốn góp phần nâng cao chất lượng và hiệu quả của các mô hình học máy hiện nay.

Từ khoá: *hàm thất thoát, học máy vi phân, luồng gradient.*

1. ĐẶT VẤN ĐỀ

Trong bài báo này, để cho việc trình bày được rõ ràng và dễ hiểu, chúng ta sẽ chủ yếu xét bài toán phân loại nhị phân, hầu hết các bài toán phân loại tổng quát khác đều có thể được xử lí một cách tương tự. Hàm thất thoát đóng một vai trò vô cùng quan trọng trong học máy. Tuy nhiên trước năm 2000, việc nghiên cứu hàm thất thoát ít được để tâm vì các nhà khoa học cho rằng đây chỉ là vấn đề tính toán và không ảnh hưởng đến kết quả cuối cùng của mô hình học máy (xem [1, 2, 3]). Hiện nay các nhà khoa học đã chú ý nhiều hơn đến các tính chất của hàm thất thoát và ảnh hưởng của chúng đến sự hội tụ của các thông số trong phương pháp luồng gradient ngẫu nhiên (stochastic gradient flow) [4, 5, 6, 7, 8, 9, 10, 11, 12]. Mục đích của bài báo là góp phần phát triển một lí thuyết về hàm thất thoát. Sau khi đề cập đến những kiến thức chung về học máy vi phân, tác giả sẽ chứng minh một số kết quả lí thuyết và thực nghiệm sau đây:

- Mô hình tổng quan của học máy vi phân
- Độ nhiễu ngẫu nhiên ngăn cản sự hội tụ của phương pháp gradient đến điểm cực tiểu.
- Các hàm thất thoát không đối xứng tốt hơn các hàm thất thoát đối xứng, đặc biệt là đối với các vấn đề có sự mất cân bằng nghiêm trọng về dữ liệu giữa các lớp khác nhau, lớp này có quá ít dữ liệu so với lớp kia.

2. PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Mô hình tổng quan về học máy vi phân

Ta sử dụng các ký hiệu sau: Ω là không gian đầu vào, bao gồm tất cả các tình huống có thể xuất hiện trong bài toán, cùng với độ đo xác suất P (phụ thuộc vào từng bối cảnh, tình huống). Ví dụ Ω là tập các ảnh trong bài toán phân loại nhị phân phát hiện con hổ: ảnh có con hổ và ảnh không có con hổ trong đó.

$y_{true}: \Omega \to \{0,1\}$ là ground truth (giá trị thật) của hàm phân loại nhị phân. Ví dụ $y_{true}(x) = 1$ nếu và chỉ nếu ảnh là ảnh có con hổ trong đó.

Một mô hình học máy là một ánh xạ:

$$M:\Omega\times\Theta\to\{0,1\}, \quad (1)$$

trong đó Θ biểu thị không gian các tham số có thể học của mô hình học máy này. Đối với mỗi lựa chọn các tham số $\theta\in\Theta$ thì mô hình M cho ta đầu ra là một hàm dự đoán:

$$y_{predic}=M_\theta:\Omega\to\{0,1\}, \quad (2)$$

Trong quá trình học vi phân, người ta thường thay thế hàm có giá trị rời rạc $y_{predict}$ bởi một hàm trơn liên tục hầu khắp:

$$y=DM_\theta:\Omega\to[0,1] \quad (3)$$

có thể được hiểu là "xác suất", "khả năng" hoặc "mức độ tin cậy" trong một dự đoán nhị phân: ta đặt $y_{predict}=1$ khi $y>0.5$ (hoặc một ngưỡng khác nào đó), và y càng gần 1 thì ta càng tin tưởng vào dự đoán này.

Ví dụ, trong bài toán phát hiện con hổ, nếu $y_{predict}=0.99$ thì máy nói rằng ảnh có con hổ với độ tin tưởng rất cao, nếu $y_{predict}=0.65$ thì máy coi là ảnh có vẻ có hổ nhưng không chắc chắn lắm, còn nếu $y_{predict}=0.03$ thì có thể coi là máy không nhận ra có hổ trong ảnh.

Số chiều của không gian Θ chính là số các tham số mà mô hình máy có thể học. Số chiều này có thể là nhỏ (vài đơn vị hoặc vài chục, vài trăm, vài nghìn) trong các mô hình học máy đơn giản (ví dụ như trong một mô số bài toán hồi quy ước lượng giá trị), và cũng có thể rất lớn, đến hàng chục triệu hoặc hàng trăm triệu, đặc biệt là trong các mô hình học sâu (deep learning) sử dụng CNN (convolutional neural network – mạng thần kinh dùng tích chập trên các tensor).

Quá trình học trên mô hình M là một hệ động lực (ngẫu nhiên, rời rạc, thời gian hữu hạn) trên không gian tham số Θ:

$$\theta_0\mapsto\theta_1\mapsto\theta_2\mapsto\ldots\mapsto\theta_n\mapsto\ldots \quad (4)$$

sao cho với n nào đó thì ta đạt được M_{θ_n} là một xấp xỉ tốt nhất có thể của y_{true} (hi vọng như vậy).

Hàm đo độ chính xác nhị phân (binary accuracy function):

$$S(M_\theta,y_{true})=P\{x\in\Omega\,|\,M_\theta(x)=y_{true}(x)\} \quad (5)$$

và các hàm tương tự, như là sensitivity (tỉ lệ dương đúng (true positive): tổng số các trường hợp thực sự dương tính được máy xác định là dương tính chia cho tổng số các trường hợp thực sự dương tính) và hàm specificity (tỉ lệ âm đúng (true nagative): tổng số các trường hợp thật sự âm tính mà được máy xác định là âm tính chia cho tổng số các trường hợp thật sự âm tính), được sử dụng để đo độ chính xác của mô hình.

Trong thực hành, $S(M_\theta,y_{true})$ được tính bằng thực nghiệm, dựa trên việc kiểm tra kết quả trên một tập ngẫu nhiên N trường hợp $x_i\in\Omega,i=1,\ldots,N$ không được sử dụng trong quá trình học (điều này cũng tương tự như là việc ra đề bài kiểm tra đối với học sinh: các bài kiểm tra phải có nét tương tự nhưng khác so với các bài đã được chữa trên lớp, để kiểm tra xem học sinh có khả năng áp dụng kiến thức không hay chỉ học vẹt thôi):

$$S(M_\theta,y_{true})=\frac{\left|\{k=1,\ldots,N;M_\theta(x_k)=y_{true}(x_k)\}\right|}{N} \quad (6)$$

Tập các x không dùng để học mà dùng để kiểm tra như vậy được gọi là tập xác nhận (validation set) hoặc tập kiểm tra (test set) (tùy theo ai là người thực hiện việc kiểm tra này: người tạo ra mô hình học máy, hay là người sử dụng mô hình học máy).

Trong quá trình học vi phân (differential learning), người ta thay thế tỉ lệ lỗi $1-S(M_\theta,y_{true})$ bởi một hàm gọi là hàm thất thoát (loss function)

$$L:\Theta\to\mathbb{R} \quad (7)$$

có tính khả vi liên tục hầu khắp. Nói một cách trực giác, hàm thất thoát L phải được chọn sao cho giá trị của L càng thấp thì ứng với độ chính xác của máy càng cao.

Hàm thất thoát L được tính bởi công thức tích phân (lấy trung bình):

$$L(\theta)=\int_{x\in\Omega}\ell\left(DM_\theta(x),y_{true}(x)\right)dP_\Omega, \qquad (8)$$

trong đó ℓ là một hàm thất thoát tính cho từng điểm, và có tính chất khả vi liên tục hầu khắp.

Khi có hàm thất thoát L, người ta sử dụng phương pháp giảm giá trị theo luồng gradient (gradient descent) để tìm giá trị các tham số θ_n có tính chất "hầu như làm cực tiểu hóa (almost minimizes)" hàm L.

Một cách đơn giản, quá trình học vi phân được xác định như sau. Bắt đầu với một bộ tham số $\theta_0\in\Theta$ (hoặc là một giá trị ngẫu nhiên, hoặc một giá trị "đã được học từ trước, bây giờ sẽ học tiếp"). Ở bước i trong quá trình học, ta đặt:

$$\theta_i\mapsto\theta_{i+1}=\theta_i-\alpha\nabla L(\theta_i)+m(\theta_i-\theta_{i-1}) \qquad (9)$$

trong đó $\alpha>0$ được chọn là số dương nhỏ, gọi là tỉ lệ học (learning rate), ∇ kí hiệu cho gradient, và $m(\theta_i-\theta_{i-1})$ là một "momemtum" nhỏ (dư âm từ bước trước) được thêm vào công thức.

Nói chung không thể tính chính xác giá trị của gradient $\nabla L(\theta)$. Người ta chỉ có thể tính toán nó theo phương pháp thống kê lấy trung bình, sử dụng một mẫu dữ liệu tương đối nhỏ (đủ nhỏ để có thể cho vào bộ nhớ hoạt động của bộ vi xử lí của máy tính) gọi là batch ở mỗi bước, và do đó luồng gradient được gọi là luồng gradient ngẫu nhiên.

Luồng gradient thực sự của một hàm số thì không nhất thiết tiến tới điểm cực tiểu toàn cục, mà hay bị mắc kẹt tại những điểm cực tiểu địa phương (có gía trị cao hơn so với cực tiểu toàn cục) và tại những điểm "yên ngựa" (không phải cực tiểu nhưng có đạo hàm toàn phần bằng 0). Để tránh các tình huống như vậy và để cho quá trình học có nhiều khả năng tiến đến những điểm gần đạt giá trị cực tiểu toàn cục hơn, người ta thêm vào các thành phần "momentum" kiểu như $m(\theta_i-\theta_{i-1})$ được viết ở phía trên vào công thức. Như vậy, trên thực tế, người ta sử dụng luồng gradient có tính ngẫu nhiên và có mommentum, thay vì luồng gradient thuần túy. Trên thực tế, người ta có thể dùng nhiều lựa chọn công thức khác nhau cùng có dáng điệu chung kiểu gradient cho quá trình học, không nhất thiết phải là công thức đúng như trên.

2.2. Hiện tượng mất cân bằng dữ liệu

Mô tả mô hình: Trong mô hình này, không gian đầu vào Ω chỉ là một khoảng. Hàm phân loại nhị phân là một hàm hằng số theo từng khúc:

$$\Omega=\cup_{i=0}^{n}\left[a_i,a_{i+1}\right[\qquad (10)$$

với $a=a_0<a_1<\ldots<a_{n+1}=b$ và sự thật là: $y_{true}=1$ trên $\Omega_+=\cup\left[a_{2i},a_{2i+1}\right[$ và $y_{true}=-1$ trên $\Omega_-=\cup\left]a_{2i+1},a_{2i+2}\right[$. (Thay vì lấy giá trị 0 và 1 cho phân loại nhị phân, ở đây ta lấy 1 và -1 để đặc trưng cho dương tính và âm tính).

Mỗi bộ tham số trong mô hình này gồm n tham số $\theta=(\theta_1,\ldots,\theta_n)$. $M_\theta(x)=-1$ nếu $x\in\cup\left]a_{2i+1},a_{2i+2}\right[$. Rõ ràng, mô hình đạt độ chính xác tối đa 100% khi các tham số $\theta=(\theta_1,\ldots,\theta_n)$ trùng khớp với các hệ số $a=(a_1,\ldots,a_n)$ của hàm sự thật.

Vì y_θ không khả vi nên trong mô hình học máy vi phân này ta chọn một hàm khả vi g đại diện cho nó và phụ thuộc vào n tham số $(\theta_1,\ldots,\theta_n)$, có dạng như sau:

$$g(\theta_1,\ldots,\theta_n,x)=\prod_i\left(-\phi(x-\theta_i)\right) \qquad (11)$$

trong đó $\phi(x)$ là một hàm lẻ $(\phi(-x)=-\phi(x))$ đơn điệu tăng trên $\mathbb{R}$, và lõm (có đạo hàm bậc 2 là hàm âm) trên $\mathbb{R}_+$, thỏa mãn $\phi(0)=0$ và $\lim_{x\to\pm\infty}\phi(x)=\pm 1$.

Ví dụ, ta lấy $\phi(x)=\frac{2}{\pi}\arctan(x)$, hoặc $\phi(x)=\frac{x}{\sqrt{x^2+\epsilon}}$ với ϵ là một số thực dương bất kì. Chúng ta sẽ không băn khoăn nhiều về công thức chính xác của hàm $\phi(x)$.

Hàm số $g(\theta_1,....,\theta_n,x)$ có giá trị nằm trong khoảng $]-1,1[$; dương hoặc bằng 0 trên tập $\cup[a_{2i},a_{2i+1}]$ và âm trên tập $\cup]a_{2i+1},a_{2i+2}[$. Bởi vậy, hàm dự đoán của mô hình là:

$$M_\theta(\omega)=1 \text{ nếu } g(\theta,\omega)\geq 0 \qquad (12)$$

và

$$M_\theta(\omega)=0 \text{ nếu } g(\theta,\omega)<0 \qquad (13)$$

Chúng ta không biết các giá trị của $(a_1,....,a_n)$, và muốn tìm chúng bằng cách sử dụng luồng gradient ngẫu nhiên của hàm thất thoát sau:

$$L(\theta)=\int_a^b \ell(\theta,x)dx, \qquad (14)$$

trong đó

$$\ell(\theta,x)=1-y_{true}(x)g(\theta,x) \qquad (15)$$

Ý nghĩa của hàm thất thoát theo điểm $\ell(\theta,x)$ định nghĩa phía trên như sau: giá trị của thất thoát tại mỗi điểm nằm trong khoảng từ 0 đến 2; nếu đoán đúng lớp, tức là $g(\theta,x)$ và $y_{true}(x)$ có cùng dấu, thì thất thoát nhỏ hơn 1, còn nếu đoán sai lớp thì thất thoát lớn hơn 1. Thất thoát tại điểm $x\in\Omega=[a,b]$ mà càng nhỏ (càng gần 0) thì có nghĩa là $g(\theta,x)$ phải càng gần ± 1 và có dấu trùng với $y_{true}(x)$, tức là máy đoán đúng một cách "càng chắc chắn" về giá trị của $y(x)$ cho điểm x.

Thay vì xét hàm thất thoát, ta có thể xét hàm thu thập (gain function):

$$G(\theta)=\int_a^b y_{true}(x)g(\theta,x)dx \qquad (16)$$

Hàm thu thập $G(\theta)$ không đạt cực đại tại điểm $\theta=(a_1,....,a_n)$ trong không gian tham số, là tham số cho phép máy M_θ dự đoán chính xác hoàn toàn, mà là tại một điểm tham số khác. Nói cách khác, nói chung, phương pháp học vi phân với hàm thất thoát như trên (hoặc với bất kì hàm thất thoát nào khác cho vấn đề dự đoán nhị phân ở đây) sẽ không cho chúng ta mô hình dự đoán với độ chính xác 100%, ngay cả khi mô hình đó tồn tại. Thực tế này có thể thấy rõ trong trường hợp chỉ một tham số có thể học $(n=1)$.

3. KẾT QUẢ VÀ THẢO LUẬN

3.1. Trường hợp với một tham số

Mệnh đề: Với các ký hiệu trên, trong trường hợp $n=1$, $g(\theta,x)=-\phi(x-\theta)$, ta có:

1 - Trường hợp cân bằng. Nếu $b-a_1=a_1-a$, tức là $a_1=(a+b)/2$, thì a_1 là điểm cực đại của hàm thu thập $G(\theta)$.

2 - Trường hợp chênh lệch. Nếu $b-a_1>a_1-a$ nhưng $|b+a-2a_i|$ đủ nhỏ thì G đạt cực đại không phải tại điểm $\theta=a_1$ mà tại một điểm lân cận trong khoảng $[a,a_1[$.

3 - Trường hợp quá chênh lệch. Nếu $a_1-a<b-a_1$ và nhỏ đến mức

$\phi(b-a_1)>3\phi(a_1-a)$, thì khi đó đạo hàm của G là hàm âm trên đoạn $[a,b]$ và điểm cực đại của G trên đoạn $[a,b]$ là điểm a.

Trong mệnh đề trên, a_1-a hiểu là độ lớn của tập dương tính trong mô hình, còn $b-a_1$ là độ lớn của tập âm tính. Trường hợp 1) là trường hợp cân bằng, khi hai độ lớn này bằng nhau; trường hợp 2) là trường hợp có nhiều dữ liệu âm tính hơn dương tính, còn trường hợp 3) là trường hợp có quá ít dữ liệu dương tính so với dữ liệu âm tính.

Chứng minh. Trong trường hợp $n=1$ thì $g(\theta,x)=-\phi(x-\theta)$ và $y_{true}(x)=sign(a_1-x)$, do đó:

$$\begin{aligned} G(\theta) &= -\int_{a}^{a_1}\phi(x-\theta)\,dx+\int_{a_1}^{b}\phi(x-\theta)\,dx \\ &= \int_{a-\theta}^{a_1-\theta}\phi(x)\,dx+\int_{a_1-\theta}^{b-\theta}\phi(x)\,dx \end{aligned} \tag{17}$$

Đạo hàm của G bằng:

$$\begin{aligned} G'(\theta) &:= \frac{dG(\theta)}{d\theta} \\ &= -\phi(a-\theta)-\phi(b-\theta)+2\phi(a_1-\theta) \end{aligned} \tag{18}$$

Do $\phi(x)$ được chọn là hàm lẻ và dương khi x dương nên công thức trên còn có thể viết như sau:

$$\begin{aligned} G'(\theta) &= \phi(\theta-a)-\phi(b-\theta)-2\phi(\theta-a_1) \\ &= \phi(\theta-a)+2\phi(a_1-\theta)-\phi(b-\theta) \end{aligned} \tag{19}$$

trong đó $\phi(\theta-a),\phi(b-\theta)\geq 0$, còn dấu của $\phi(\theta-a_1)$ trùng với dấu của $\theta-a_1$.

Trong trường hợp 3) thì đạo hàm $G'(\theta)$ luôn âm với mọi $\theta\in[a,b]$. Thật vậy, nếu $a\leq\theta\leq a_1$ thì $\begin{aligned}&\phi(\theta-a)+2\phi(a_1-\theta)\leq 3\phi(a_1-a)\\&<\phi(b-a_1)\leq\phi(b-\theta)\end{aligned}$, từ đó suy ra $G'(\theta)<0$. Nếu $a_1<\theta<(a+b)/2$ thì $\phi(\theta-a)<\phi(b-\theta)$ và $\phi(\theta-a_1)>0$, vậy ta cũng có $G'(\theta)<0$. Nếu $\theta>(a+b)/2$ thì do tính lõm của hàm ϕ trên tập số dương và do $\phi(0)=0$ nên ta có:

$$\begin{aligned}&\phi(\theta-a)+\phi(a_1-a)<\phi(\theta-a_1)+\phi(b-a_1)\\&<\phi(\theta-a_1)+\phi(b-\theta)+\phi(\theta-a_1)\end{aligned}$$

suy ra $G'(\theta)<0$. Như vậy, trong trường hợp 3) hàm $G'(\theta)$ là hàm âm trên $[a,b]$, suy ra điểm cực đại của hàm G chính là điểm a trong trường hợp này.

Trong trường hợp 1) khi $a_1=(a+b)/2$ nằm giữa a và b, dễ thấy $G'(\theta)=0$ tại chính điểm $\theta=a_1$.

Tại các điểm $\theta>a_1$ thì:

$$\begin{aligned}\theta-a &= (b-\theta)+2\left[\theta-(a+b)/2\right] \\ &= (b-\theta)+2(\theta-a_1)\end{aligned},$$

Suy ra $\phi(\theta-a)<\phi(b-\theta)+2\phi(\theta-a_1)$ do tính chất lõm của ϕ trên $\mathbb{R}_+$ và do $\phi(0)=0$, suy ra $G'(\theta)<0$. Tương tự như vậy, với mọi $a\leq\theta\leq a_1$ ta cũng có $G'(\theta)<0$, và do đó cực đại của G đạt tại chính điểm a_1 trong trường hợp đối xứng này.

Trong trường hợp 2), lí luận tương tự như trong trường hợp 3), ta cũng có với mọi $\theta\geq a_1$. Mặt khác, $G'(a)=2\phi(a_1-a)-\phi(b-a)>0$ nếu a_1 đủ gần $(a+b)/2$, bởi như lúc trước ta đã thấy $2\phi((b+a)/2-a)-\phi(b-a)>0$ do tính lõm của ϕ trên tập $\mathbb{R}_+$. Do vậy, điểm cực

đại của G nằm trong đoạn $[a, a_1]$ trong trường hợp này, và do tính liên tục nên khi a_1 càng gần $(a+b)/2$ (trường hợp cân bằng) thì điểm cực đại này cũng sẽ càng gần a_1.

3.2. Các kết luận suy ra từ mệnh đề

Mệnh đề trên cho ta thấy:

Sự cân bằng dữ liệu là cần thiết để có sự trùng khớp giữa các điểm cực tiểu của hàm thất thoát và điểm tốt nhất của mô hình học máy. Khi không có cân bằng dữ liệu thì kể cả nếu luồng gradient ngẫu nhiên hội tụ được về điểm cực tiểu của hàm thất thoát, ta cũng không chắc sẽ tìm được điểm tốt nhất cho mô hình học máy.

Khi có chênh lệch về dữ liệu giữa các lớp, thì lớp có càng nhiều dữ liệu lại càng dễ được máy học kiểu vi phân thiên vị, lớp có càng ít dữ liệu lại càng dễ bị kì thị. Khi mà một lớp quá nhỏ so với lớp khác, thì có nguy cơ là máy học kiểu vi phân sẽ bỏ qua lớp đó, coi như nó không tồn tại, trong quá trình làm tối thiểu hóa thất thoát.

Ở đây tuy không đi vào chi tiết trường hợp nhiều tham số nhưng về mặt trực giác ta thấy, khi có nhiều tham số, việc mất cân bằng dữ liệu cũng khiến cho điểm cực tiểu của hàm thất thoát có thể khác xa điểm tối ưu của máy, và điều này ảnh hưởng xấu đến kết quả của việc học máy vi phân.

Khắc phục ảnh hưởng của bất cân đối dữ liệu:

Để cải thiện khả năng học của các mô hình học máy vi phân trong trường hợp dữ liệu mất cân đối, ta có thể áp dụng một số phương pháp khác nhau và kết hợp chúng với nhau, như:

Cân bằng lại dữ liệu (data rebalancing):

Ta khuếch đại một cách nhân tạo lớp thiểu số trong quá trình cho dữ liệu vào máy học, để sao cho lượng dữ liệu của lớp thiểu số đưa vào máy cũng tương đương với lượng dữ liệu của lớp đa số. Về mặt xác suất, việc đó có nghĩa là ta thay đổi phân bố xác suất trên không gian các dữ liệu, tạo ra một phân bố xác suất mới sử dụng cho việc học máy, mà trong đó lớp thiểu số và đa số có xác suất ngang nhau.

Có nhiều cách khuếch đại dữ liệu khác nhau. Ví dụ, ta có thể làm các phép biến đổi bảo toàn y_{true}, không chỉ riêng cho lớp thiểu số, mà cho cả lớp đa số, để tập hợp các dữ liệu dùng cho máy học được phong phú và bao phủ đầy đủ các tình huống hơn. Chẳng hạn, đối với một ảnh con hổ, ta có thể làm các phép biến đổi như cắt bớt nó, viết đè lên nó, xoay nghiêng nó, v.v…, mà vẫn nhận được là có con hổ trong đó. Các phép biến đổi như vậy thường được gọi là phép tăng cường dữ liệu (data augmentation). Ngoài tăng cường dữ liệu, ta cũng có thể tạo ra dữ liệu tổng hợp nhân tạo (synthetic data) từ những dữ liệu thực tế ban đầu.

Hàm thất thoát không đối xứng (asymmetric loss functions):

Hầu hết các hàm thất thoát thông dụng có sẵn là hàm đối xứng theo các lớp, tức là giá trị của nó không thay đổi nếu ta hoán vị giá trị của các lớp với nhau, các lớp đều quan trọng như nhau trong công thức của hàm thất thoát.

Tuy nhiên, ta có thể xây dựng các hàm thất thoát không đối xứng (asymmetric loss functions), mà ở đó các lớp khác nhau được tính theo các trọng số khác nhau. Đặc biệt, các lớp thiểu số thì được tính theo trọng số cao hơn trong hàm thất thoát, để cho vị thế của chúng trong quá trình học được tăng cường lên ngang bằng với vị thế của lớp đa số.

Hàm thất thoát có độ sắc cao (sharp loss functions):

Trong mô hình ở trên, họ các hàm số:

$$\phi_\epsilon(x) = \frac{x}{\sqrt{x^2+\epsilon}} \tag{20}$$

dùng để xây dựng hàm thất thoát có chứa một tham số ϵ. Tham số này phản ánh độ sắc (sharpness) của hàm thất thoát: ϵ càng nhỏ thì hàm càng sắc, theo nghĩa là lượng điểm có mức thất thoát lưng chừng càng ít đi, và độ thất thoát của các điểm đoán đúng (đoán sai) càng gần mức nhỏ nhất (cao nhất). Khi máy M_θ cố định và θ tiến tới 0 thì độ thất thoát của các điểm đoán đúng tiến tới 0 và của các điểm đoán sai tiến tới một trong hai trong mô hình nói trên.

Khi ϵ tiến tới 0 thì "độ kì thị" đối với lớp thiểu số trong trường hợp 2) của Mệnh đề (ở đây là độ chênh lệch giữa điểm cực tiểu của hàm thất thoát và điểm tối ưu $\theta = a_1$ của mô hình) cũng tiến tới 0. Như vậy, hàm thất thoát càng sắc thì điểm cực tiểu của nó càng gần tham số tối ưu thực sự của mô hình hơn và càng bớt kì thị lớp thiểu số hơn. Vì lí do này, ta có thể muốn tăng độ sắc của hàm thất thoát dùng trong quá trình học máy vi phân.

Tuy nhiên, có một giá phải trả cho độ sắc của hàm thất thoát, khiến cho ta không thể dùng hàm thất thoát sắc quá một mức nào đó. Cụ thể là, khi hàm thất thoát càng sắc, thì ảnh hưởng của tính bất định tạo bởi các yếu tố ngẫu nhiên trong quá trình học càng cao, khiến cho luồng gradient ngẫu nhiên càng khó tiến tới điểm cực tiểu mà trái lại càng giao động ở "mức năng lượng" cao hơn. Nói cách khác, khi điểm cực tiểu của hàm thất thoát càng gần điểm tham số tối ưu thực sự của máy (do hàm thất thoát càng sắc) thì những điểm mà ta tìm ra được bằng phương pháp luồng gradient lại càng xa điểm cực tiểu của hàm thất thoát, và như vậy chúng vẫn xa điểm tham số tối ưu của máy dù hàm thất thoát có sắc đến mấy. Và tất nhiên, khi độ sắc tiến tới vô cùng, thì hàm thất thoát trở thành hàm nhảy giữa một số giá trị rời rạc, mất tính khả vi và không còn dùng được trong học máy vi phân nữa.

Trong mỗi vấn đề học máy vi phân cần có một độ sắc tối ưu cho hàm thất thoát, để trung hòa tốt nhất giữa độ lớn của hai sai số (để cho cả hai đều nhỏ): sai số giữa điểm tối ưu của máy và điểm cực tiểu của hàm thất thoát; sai số giữa điểm cực tiểu của hàm thất thoát và điểm tìm được bằng phương pháp luồng gradient ngẫu nhiên.

4. KẾT LUẬN VÀ KIẾN NGHỊ

4.1. Kết luận

Bài báo này đã nhấn mạnh tầm quan trọng của việc cân bằng dữ liệu và xây dựng hàm thất thoát phù hợp trong học máy vi phân để đảm bảo sự trùng khớp giữa các điểm cực tiểu của hàm thất thoát và điểm tối ưu của mô hình. Các phát hiện chính bao gồm:

1) Cân bằng dữ liệu là yếu tố then chốt để đảm bảo rằng điểm cực tiểu của hàm thất thoát tương ứng với điểm tối ưu của mô hình học máy. Thiếu cân bằng dữ liệu có thể khiến mô hình không đạt được hiệu suất tốt nhất, ngay cả khi luồng gradient ngẫu nhiên hội tụ.

2) Chênh lệch dữ liệu giữa các lớp có thể dẫn đến sự thiên vị của mô hình đối với lớp có nhiều dữ liệu hơn và sự kì thị đối với lớp có ít dữ liệu hơn. Nếu một lớp quá nhỏ so với lớp khác, có nguy cơ mô hình sẽ bỏ qua lớp nhỏ này trong quá trình tối ưu hóa.

3) Mất cân bằng dữ liệu và nhiều tham số cũng có thể dẫn đến điểm cực tiểu của hàm thất thoát khác xa điểm tối ưu của mô hình, ảnh hưởng xấu đến kết quả của học máy vi phân.

4) Xây dựng hàm thất thoát phù hợp là một yếu tố quan trọng để nâng cao hiệu suất của mô hình. Một hàm thất thoát được thiết kế tốt sẽ giúp mô hình đạt được hiệu suất cao hơn và giảm thiểu sai số, đặc biệt trong các tình huống dữ liệu không đồng đều.

4.2. Kiến nghị

Để khắc phục những ảnh hưởng của mất cân bằng dữ liệu và cải thiện khả năng học của các mô hình học máy vi phân, các chiến lược sau đây nên được xem xét mở rộng, nghiên cứu và ứng dụng:

1) Nghiên cứu sâu hơn về cân bằng dữ liệu (Data Rebalancing):

Nghiên cứu và phát triển các phương pháp khuếch đại dữ liệu thiểu số (Data Augmentation) mới, đảm bảo rằng các phép biến đổi không chỉ bảo toàn đặc tính mà còn tăng cường khả năng bao phủ của tập dữ liệu.

Phát triển các kỹ thuật tạo dữ liệu tổng hợp (Synthetic Data) tiên tiến hơn, đặc biệt là sử dụng mô hình Generative Adversarial Networks (GANs) để tạo ra dữ liệu đa dạng và thực tế hơn.

2) Xây dựng và thử nghiệm các hàm thất thoát không đối xứng (Asymmetric Loss Functions):

Tiến hành các nghiên cứu thực nghiệm để xác định các trọng số tối ưu cho các lớp thiểu số

trong hàm thất thoát, nhằm đảm bảo cân bằng giữa các lớp trong quá trình học.

Phát triển các hàm thất thoát động (Dynamic Loss Functions) có khả năng điều chỉnh trọng số dựa trên sự thay đổi của phân bố dữ liệu trong quá trình huấn luyện.

3) Phát triển hàm thất thoát có độ sắc cao (Sharp Loss Functions):

Nghiên cứu mối quan hệ giữa độ sắc của hàm thất thoát và hiệu suất của mô hình để xác định mức độ sắc tối ưu.

Phát triển các thuật toán học máy mới để giảm thiểu ảnh hưởng của tính bất định do các yếu tố ngẫu nhiên trong quá trình học khi sử dụng các hàm thất thoát sắc.

4) Ứng dụng trong các lĩnh vực cụ thể:

Áp dụng các phương pháp trên vào các lĩnh vực như nhận dạng hình ảnh, xử lí ngôn ngữ tự nhiên, và phát hiện gian lận để kiểm chứng tính hiệu quả và thực tiễn.

Phát triển các hệ thống học máy vi phân chuyên dụng cho các ứng dụng đòi hỏi độ chính xác cao và sự công bằng giữa các lớp dữ liệu.

TÀI LIỆU THAM KHẢO

1. Cristianini, N. and Shawe Taylor, J. (2000). An Introduction to Support Vector Machines. Cambridge University Press, Cambridge, UK.
2. Nguyen Thanh Thien, Nguyen Tien Zung, Reduction and Integrability of Stochastic Dynamical Systems, Journal of Mathematical Sciences, 2017, 225 (4),pp.681-706.
3. V. Vapnik, Statistical Learning Theory. Wiley, New York (1998).
4. Nabila Abraham, Naimul Mefraz Khan, A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation, arXiv:1810.07842 (2018)
5. Gareth M. James, Variance and Bias for General Loss Functions, Machine Learning, May 2003, Volume 51, Issue 2, pp 115–135.
6. Hoel Kervadec, Jihene Bouchtiba, Christian Desrosiers, Eric Granger Jose Dolz, Ismail Ben Ayed, Bound[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár, Focal Loss for Dense Object Detection, arXiv:1708.02002 (2017).
7. Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana and Alessandro Verri, Are Loss Functions All the Same? Neural Computation, Volume 16, Issue 5, May 2004, p.1063-1076
8. Chen Shen, Holger R. Roth, Hirohisa Oda, Masahiro Oda, Yuichiro Hayashi, Kazunari Misawa, Kensaku Mori, On the influence of Dice loss function in multi-class organ segmentation of abdominal CT using 3D fully convolutional networks, preprint arXiv:1801.05912v1, 2018.
9. Carole H Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, M. Jorge Cardoso, Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations, arXiv:1707.03237v3 (2017)
10. Jurgen¨ Braun, On Kolmogorov's Superposition Theorem and Its Applica-tions, SVH Verlag, 2010, 192 pp.
11. Lijun Wu, Fei Tian, Yingce Xia, Yang Fan, Tao Qin, Jianhuang Lai, Tie-Yan Liu, Learning to Teach with Dynamic Loss Functions, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.
12. Hang Zhao, Orazio Gallo, Iuri Frosio, Jan Kautz, Loss Functions for Im-age Restoration With Neural Networks, IEEE Transactions on Computational Imaging, Volume 3 , Issue 1, March 2017, pages 47 – 57.

Thông tin của tác giả:

TS. Lê Bích Phượng
Khoa Khoa học Cơ bản, Trường Đại học Mỏ - Địa chất
Điện thoại: +(84).988.782.112 Email: lebichphuong@humg.edu.vn

CONSTRUCTING LOSS FUNCTIONS FOR OPTIMAL PERFORMANCE IN DIFFERENTIAL MACHINE LEARNING

Information about authors:

Le Bich Phuong, Ph.D., Faculty of Basic Science, Hanoi University of Mining and Geology.
Email: lebichphuong@humg.edu.vn

ABSTRACT:

In differential machine learning, the stochastic gradient flow method is often used to find the "near-minimum" point of the loss function, which corresponds to optimizing the classification algorithm. Despite the crucial role of the loss function in this process, its theoretical foundation has not been fully developed. This paper aims to contribute to the theoretical foundation of loss functions, providing a more detailed and systematic framework to support the development of more effective optimization and classification methods. We also present analyses on how the loss function impacts model performance and propose several improvements in the design and use of loss functions to achieve optimal performance. These studies not only help to better understand the nature of loss functions but also pave the way for new applications of differential machine learning in practical problems. Through this, the paper hopes to enhance the quality and efficiency of current machine learning models.

Keywords: *loss function, differential machine learning, gradient flow.*

REFERENCES

1. Cristianini, N. and Shawe Taylor, J. (2000). An Introduction to Support Vector Machines. Cambridge University Press, Cambridge, UK.
2. Nguyen Thanh Thien, Nguyen Tien Zung, Reduction and Integrability of Stochastic Dynamical Systems, Journal of Mathematical Sciences, 2017, 225 (4),pp.681-706.
3. V. Vapnik, Statistical Learning Theory. Wiley, New York (1998).
4. Nabila Abraham, Naimul Mefraz Khan, A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation, arXiv:1810.07842 (2018)
5. Gareth M. James, Variance and Bias for General Loss Functions, Machine Learning, May 2003, Volume 51, Issue 2, pp 115–135.
6. Hoel Kervadec, Jihene Bouchtiba, Christian Desrosiers, Eric Granger Jose Dolz, Ismail Ben Ayed, Bound[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár, Focal Loss for Dense Object Detection, arXiv:1708.02002 (2017).
7. Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana and Alessandro Verri, Are Loss Functions All the Same? Neural Computation, Volume 16, Issue 5, May 2004, p.1063-1076
8. Chen Shen, Holger R. Roth, Hirohisa Oda, Masahiro Oda, Yuichiro Hayashi, Kazunari Misawa,

Kensaku Mori, On the influence of Dice loss function in multi-class organ segmentation of abdominal CT using 3D fully convolutional networks, preprint arXiv:1801.05912v1, 2018.

9. Carole H Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, M. Jorge Cardoso, Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations, arXiv:1707.03237v3 (2017)
10. Jurgen¨ Braun, On Kolmogorov's Superposition Theorem and Its Applica-tions, SVH Verlag, 2010, 192 pp.
11. Lijun Wu, Fei Tian, Yingce Xia, Yang Fan, Tao Qin, Jianhuang Lai, Tie-Yan Liu, Learning to Teach with Dynamic Loss Functions, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.
12. Hang Zhao, Orazio Gallo, Iuri Frosio, Jan Kautz, Loss Functions for Im-age Restoration With Neural Networks, IEEE Transactions on Computational Imaging, Volume 3 , Issue 1, March 2017, pages 47 – 57.

Ngày nhận bài: 20/5/2024;
Ngày gửi phản biện: 21/5/2024;
Ngày nhận phản biện: 10/6/2024;
Ngày chấp nhận đăng: 14/6/2024.