ERSD
EARTH SCIENCES AND NATURAL RESOURCES FOR SUSTAINABLE DEVELOPMENT

HỘI NGHỊ TOÀN QUỐC KHOA HỌC TRÁI ĐẤT VÀ TÀI NGUYÊN VỚI PHÁT TRIỂN BỀN VỮNG (ERSD2020 – 12/11/2020)

CÁC HÀM THẤT THOÁT TRONG VẤN ĐỀ PHÂN LOẠI

Lê Bích Phượng, Phạm Tuấn Cường, Tạ Quang Chiểu

Trường Đại học Mỏ - Địa chất

Mở đầu

- Hàm thất thoát chiếm một vai trò vô cùng quan trọng trong học máy.
- Quan điểm cũ: hàm thất thoát chỉ là vấn đề tính toán và không ảnh hưởng đến kết quả cuối cùng của mô hình học máy .
- 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. Tuy nhiên, vẫn còn thiếu một lý thuyết đầy đủ về hàm thất thoát.
- Đề tài này góp phần phát triển lý thuyết về hàm thất thoát.
- Các nội dung chính:

  Nội dung 1: Mô hình tổng quan về học máy vi phân
  Nội dung 2: Một số vấn đề ảnh hưởng đến độ chính xác
  Nội dung 3: Thiết kế hàm thất thoát
  Nội dung 4: Một số kết quả thực hành

- Ω kí hiệu cho không gian các dữ liệu đầu vào.
- $y_{true} : \Omega \to \{0, 1\}$ là *ground truth* của hàm phân loại nhị phân.

Học máy thông thường: Mô hình

Mô hình học máy là một ánh xạ

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

Với mỗi lựa chọn các tham số $\theta \in \Theta$ thì mô hình M cho ta một hàm dự đoán:

$$y_{predict} = M_\theta : \Omega \to \{0, 1\}.$$

Học máy vi phân: Mô hình

Mô hình là một hàm khả vi liên tục hầu khắp,

$$DM : \Omega \times \Theta \to [0, 1]$$

Với mỗi $\theta \in \Theta$ thì hàm $y = DM_\theta : \Omega \to [0, 1]$ dùng để thay thế hàm dự đoán. Lúc đó ta có thể đặt

$y_{predict} = 1$ khi $y > 0.5$ (hoặc một ngưỡng khác)

Ở đây Θ biểu thị không gian các tham số có thể học của mô hình. Số chiều của không gian Θ (số các tham số của mô hình) có thể là nhỏ trong các mô hình học máy đơn giản, và cũng có thể rất lớn, đến hàng trăm triệu, ví dụ trong các mô hình học sâu (deep learning).

Học máy thông thường: **Đánh giá kết quả**

- Hàm đo *độ chính xác (accuracy)*

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

- Giá trị $S(M_\theta, y_{true})$ được xấp xỉ bằng thực nghiệm, dựa trên việc kiểm tra kết quả trên một tập ngẫu nhiên $T = \{x_k \in \Omega, k = 1, \ldots, N\}$

$$\hat{S}(M_\theta, y_{true}) = \frac{|\{k : M_\theta(x_k) = y_{true}(x_k)\}|}{N}$$

ở đây $x_k \in T$ thường chưa được sử dụng trong quá trình học. Tập T được gọi là *tập xác nhận (validation set)* hoặc *tập kiểm tra (test set)*.

- có thể dùng các hàm khác như *tỉ lệ dương đúng* (*sensitivity*) hoặc *tỉ lệ âm đúng* (*specificity*).

Học máy vi phân: **Đánh giá kết quả**

- Tỉ lệ lỗi $1 - S(M_\theta, y_{true})$ được thay thế bởi *hàm thất thoát* (*loss function*)

$$L : \Theta \to \mathbb{R} \qquad (1)$$

có tính chất khả vi liên tục hầu khắp nơi (continuously differentiable almost everywhere).

- Hàm thất thoát L được tính từ mô hình vi phân DM, bởi công thức tích phân

$$L(\theta) = \int_{x \in \Omega} \ell(DM_\theta(x), y_{true}(x)) dP_\Omega, \quad (2)$$

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

Nội dung của đề tài 1. Mô hình tổng quan về học máy vi phân

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

$$\theta_0 \mapsto \theta_1 \mapsto \theta_2 \mapsto \ldots \mapsto \theta_n \mapsto \ldots \tag{3}$$

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

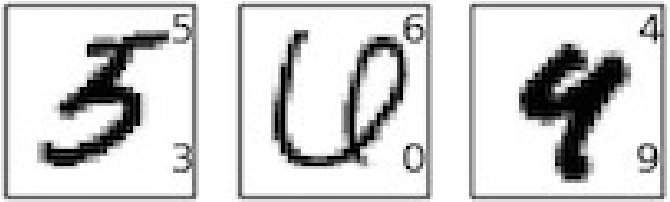
- Quá trình học vi phân được bắt đầu bằng việc chọn bộ tham số $\theta_0 \in \Theta$ (ngẫu nhiên hoặc một giá trị đã được học từ trước, nay học tiếp)
- Bước chuyển tiếp từ θ_i sang θ_{i+1} được thực hiện theo công thức

$$\theta_i \mapsto \theta_{i+1} = \theta_i - \alpha \nabla L(\theta_i) + m(\theta_i - \theta_{i-1}) \tag{4}$$

trong đó $\alpha > 0$ được chọn là số dương nhỏ (ví dụ $\alpha = 0.001$), gọi là *tỉ lệ học (learning rate)*, ∇ kí hiệu cho gradient, và $m(\theta_i - \theta_{i-1})$ là một "momentum" nhỏ (dư âm từ bước trước) được thêm vào công thức.

• Nói chung không thể tính chính xác giá trị của gradient $\nabla L(\theta)$. Ta thường tính giá trị trung bình trên một mẫu dữ liệu tương đối nhỏ (đủ nhỏ để có thể cho vào bộ nhớ của máy tính) gọi là *batch* ở mỗi bước, và do đó luồng gradient được gọi là *luồng gradient ngẫu nhiên*.

2.1. Những trường hợp ranh giới



Hình: Một số trường hợp ranh giới trong bộ dữ liệu MNIST

2.2. Dữ liệu không cân đối

Việc mất cân đối của dữ liệu khiến cho việc học trở nên khó khăn: luồng gradient ngẫu nhiên không hội tụ đến các giá trị mong muốn của các tham số, bởi vì các tham số đưa ra dự đoán chính xác nhất lại thường không phải là các tham số làm cho hàm thất thoát đạt cực tiểu.

2.3. Nhiễu bất định

• Về mặt toán học, các luồng gradient ngẫu nhiên có momentum dùng trong học máy vi phân có thể được coi như các *hệ động lực Hamilton ngẫu nhiên có suy giảm* (*damped stochastic Hamiltonian flows*).

• Nhiễu ngẫu nhiên khiến cho hệ động lực Hamilton có suy giảm không thể tiến tới điểm cực tiểu của hàm thất thoát (ở đây hiểu như là hàm năng lượng của hệ Hamilton). Trái lại, nó sẽ giao động xung quanh một mức năng lượng nào đó cao hơn mức thấp nhất, và do vậy dự đoán của máy sẽ có phần bất định.

Việc lựa chọn hàm thất thoát

Trong nhiều trường hợp, ta có thể cải thiện mô hình máy cho tăng độ chính xác đáng kể, bằng cách sử dụng một hàm thất thoát tốt hơn, phù hợp hơn với vấn đề, thay vì dùng một hàm thất thoát cổ truyền phổ biết như hàm BCE (binary cross entropy).

Đây là nội dung chính của đề tài

Hàm thất thoát tổng quát

Một hàm thất thoát tổng quát cho bài toán phân loại nhị phân có thể viết dưới dạng sau:

$$\ell(c, y_{true}, y_{pred}) = (1 - y_{true}) f(y_{pred}) + c y_{true} f(1 - y_{pred}) \tag{5}$$

trong đó:

- y_{true} là ground truth, y_{pred} là giá trị đoán ra, y_{true} chỉ nhận hai giá trị 0 và 1, còn y_{pred} có thể nhận mọi giá trị trong đoạn $[0, 1]$.
- $f : [0, 1] \to \mathbb{R}$ là một hàm đơn điệu tăng. Để đơn giản, ta sẽ coi $f(0) = 0$, tuy điều kiện này không quan trọng.
- c là "hệ số bất đối xứng". Nếu ta không quan tâm tới vấn đề mất cân bằng (bất đối xứng) dữ liệu thì đặt $c = 1$.

Khi $y_{true} = 0$ thì hàm thất thoát chính bằng $f(y_{pred})$, còn khi $y_{true} = 1$ thì hàm thất thoát bằng $c \cdot f(1 - y_{pred})$

Hàm thất thoát chặt đuôi

• Để tăng độ tập trung của hàm thất thoát, ta có thể thiết kết nó sao cho khi y gần 0 (máy đã đoán đúng với độ tin tưởng cao) thì đạo hàm của nó cũng bằng 0 (không cần phải làm cho y tiến về 0 thêm nữa trong trường hợp đó)

• Một ví dụ về hàm thất thoát đa thức chặt đuôi là

$$f_{cutoff}(y) = (\max(y - 0.2, 0))^2 \tag{6}$$

• Hàm thất thoát chặt đuôi (cutoff) trên có tính chất gãy khúc tại điểm $y = 0.2$, tức là điểm ở đó khúc đuôi của nó bị chặt.

• Trong thực hành, chúng tôi thấy hàm thất thoát có chặt đuôi nhiều khi cho ra kết quả tốt hơn là hàm thất thoát trơn không có chặt đuôi.

Hàm thất thoát gãy khúc

• Ngoài hàm thất thoát chặt đuôi, chúng tôi nghiên cứu các hàm thất thoát có gãy khúc (không trơn) tại các điểm lưng chừng khác, đặc biệt là tại điểm ngưỡng $y = 0.5$.

• Một số ví dụ của hàm thất thoát gãy khúc giữa chừng:

$$f_{brokenloss1}(y) = \min(y, 0.5)^2 + \max(y - 0.25, 0) \tag{7}$$

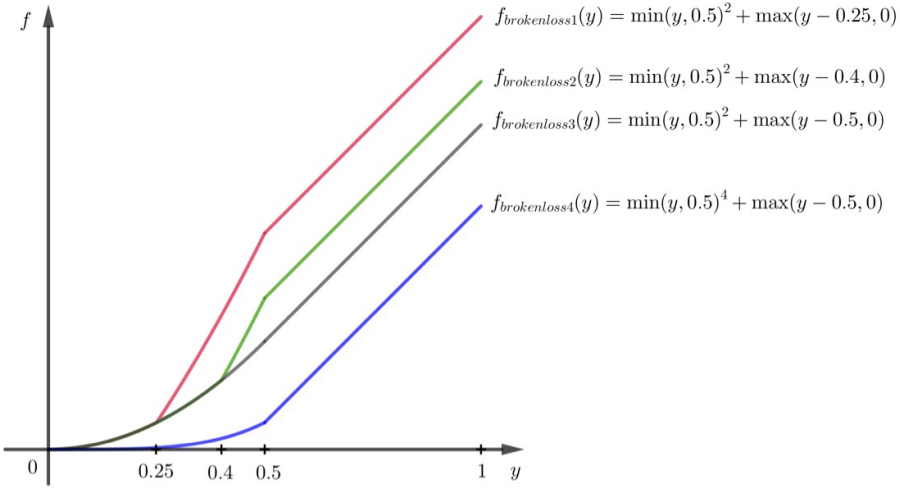
$$f_{brokenloss2}(y) = \min(y, 0.5)^2 + \max(y - 0.4, 0) \tag{8}$$

$$f_{brokenloss3}(y) = \min(y, 0.5)^2 + \max(y - 0.5, 0) \tag{9}$$

$$f_{brokenloss4}(y) = \min(y, 0.5)^4 + \max(y - 0.5, 0) \tag{10}$$

• Xét theo dáng điệu đạo hàm thì hàm $f_{brokenloss4}$ có tính tập trung tốt nhất trong 4 hàm trên. (Xem đồ thị trên trang sau).

• Khi thực hành thì hàm $f_{brokenloss4}$ cũng cho kết quả khá tốt còn các hàm kia cho kết quả tồi, thậm chí máy có thể bị rơi vào "bẫy" không thoát ra để học được.



Hình: Một số hàm thất thoát gãy khúc

Nội dung 4. Một số kết quả thực hành

- Tập dữ liệu dùng để thực hành: CIFAR-10 (https://en.wikipedia.org/wiki/CIFAR-10)
  - 60.000 ảnh được chia thành 50.000 học (training) và 10.000 để kiểm tra(test)
  - 10 loại (class): airplanes (máy bay), cars (ô-tô con), birds (chim), cats (mèo), deer (hươu), dogs (chó), frogs (ếch), horses (ngựa), ships (thuyền), trucks (xe tải)
- Thí nghiệm được thực hiện trên nền tảng Keras/Tensorflow của Google, dùng mạng Nơ-ron tích chập (CNN) có tên VGG16 do Oxford Visual Geometry Group tạo ra (`http://www.robots.ox.ac.uk/~vgg/practicals/cnn/index.html`).
- Cho mỗi hàm thất thoát ở trên, thí nghiệm được tiến hành như sau:
  - Phân loại nhị phân: vd. "đây có phải là thuyền hay không?". 1/10 dữ liệu thuộc lớp "có", và 9/10 dữ liệu thuộc lớp "không".
  - Lặp lại 30 lượt (epochs) trên bộ ảnh
  - Các ảnh được biến đổi tăng cường (augmentation), ví dụ như là xoay ảnh, lật ngược, v.v. trước khi đưa vào máy học.

Nội dung 4. Kết quả cho lớp "Airplane"

c	r	brokenloss1	brokenloss2	brokenloss3	brokenloss4	cutoffloss1
1	1	831 , 868 , 850	910 , 850 , 880	864 , 908 , 886	851 , 924 , 888	871 , 908 , **890**
	3	956 , 644 , 800	920 , 785 , 852	934 , 842 , 888	948 , 837 , 892	942 , 876 , **909**
	5	972 , 656 , 814	1000 , 000 , 500	981 , 703 , 842	895 , 905 , **900**	916 , 878 , 897
	7	1000 , 000 , 500	990 , 617 , 803	963 , 771 , 867	908 , 900 , **904**	946 , 834 , 890
	9	1000 , 000 , 500	1000 , 000 , 500	949 , 820 , 884	932 , 879 , **905**	936 , 871 , 903
2	1	945 , 720 , 833	897 , 858 , 877	881 , 893 , **887**	820 , 940 , 880	758 , 963 , 860
	3	952 , 713 , 833	950 , 815 , 882	947 , 805 , 876	948 , 813 , 880	927 , 866 , **896**
	5	1000 , 000 , 500	964 , 711 , 838	987 , 706 , 846	911 , 883 , **897**	953 , 833 , 893
	7	1000 , 000 , 500	953 , 789 , 871	966 , 770 , 868	950 , 854 , 902	929 , 881 , **905**
	9	1000 , 000 , 500	1000, 000, 500	1000, 000, 500	964, 803, 884	946 , 853 , **900**
3	1	929 , 713 , 821	922 , 853 , 887	890 , 893 , 892	862 , 921 , 892	889 , 917 , **903**
	3	963 , 651 , 807	976 , 690 , 833	952 , 834 , 893	912 , 864 , 888	924 , 884 , **904**
	5	1000 , 000 , 500	918 , 824 , 871	931 , 853 , 892	942 , 896 , **919**	922 , 873 , 898
	7	1000 , 000 , 500	963 , 846 , 854	1000 , 000 , 500	928 , 879 , **904**	917 , 863 , 890
	9	1000 , 000 , 500	1000 , 000 , 500	983 , 720 , 851	937 , 855 , 896	947 , 862 , **904**
4	1	911 , 780 , 846	905 , 859 , 882	867 , 924 , 895	799 , 955 , 877	938 , 850 , **894**
	3	972 , 530 , 751	938 , 786 , 862	935 , 821 , 878	926 , 870 , 898	927 , 868 , **904**
	5	966 , 618 , 792	958 , 772 , 865	971 , 752 , 861	916 , 899 , **907**	871 , 920 , 897
	7	1000 , 000 , 500	949 , 780 , 864	962 , 793 , 878	958 , 794 , 876	932 , 854 , **896**
	9	1000 , 000 , 500	984 , 642 , 813	949 , 846 , 898	912 , 907 , **910**	961 , 807 , 893
5	1	949 , 625 , 787	828 , 896 , 862	829 , 926 , 878	883 , 914 , **899**	883 , 907 , 895
	3	978 , 510 , 744	972 , 741 , 856	901 , 886 , 893	946 , 824 , 885	908 , 894 , **901**
	5	973 , 634 , 804	963 , 721 , 842	913 , 875 , 893	932 , 878 , **905**	931 , 871 , 901
	7	1000 , 000 , 500	973 , 700 , 836	960 , 790 , 875	939 , 875 , **907**	934 , 846 , 890
	9	979 , 550 , 765	1000 , 000 , 500	961 , 788 , 875	928 , 877 , **902**	956 , 818 , 887

Nội dung của đề tài 4. Kết quả cho lớp "Airplane"

• Bảng phía trên là kết quả thực hành cho bài toán phân loại nhị phân cho lớp "airplane".
• Hệ số r là hệ số tăng cường dữ liệu cho lớp "airplane": mỗi ảnh airplane được cho vào học r lần trong khi mỗi ảnh khác chỉ được cho vào học 1 lần trong mỗi epoch. (Mục đích: giảm chênh lệch giữa có và không).
• Hệ số c là hệ số bất đối xứng trong hàm thất thoát, mục đích cũng là để xử lý vấn đề data imbalance.
• Ở mỗi ô kết quả có 3 số: sensitivity (tỷ lệ dương đúng, tính theo phần nghìn), specificity (tỷ lệ âm đúng), và balanced accuracy (trung bình cộng của hai tỷ lệ này).
• Trong thí nghiệm này, ta thấy hàm thất thoát chặt đuôi (cutofff1) cho độ chính xác cao nhất, sau đó đến hàm brokenloss4 (cũng là hàm có độ tập trung cao). Còn các hàm thất thoát gãy khúc khác cho kết quả tồi hơn, và hay bị "kẹt bẫy" không học được gì. (Những ô có 3 số 1000, 000, 500 là máy đoán tất cả các ảnh đều là máy bay).

Nội dung 4. Kết quả cho lớp "Cat"

c	r	brokenloss1	entropyloss	brokenloss3	entropylossR	cutoffloss1
1	1	779 , 770 , 775	1000 , 000 , 500	761 , 860 , **810**	434, 955, 695	664, 887, 776
	3	841 , 660 , 750	1000 , 000 , 500	796 , 842 , **819**	686 , 902 , 794	758 , 873 , 816
	5	1000 , 000 , 500	1000 , 000 , 500	819 , 817 , 818	689 , 904 , 797	874 , 792 , **833**
	7	1000 , 000 , 500	1000 , 000 , 500	895 , 695 , 795	732 , 864 , 798	894 , 786 , **840**
	9	1000 , 000 , 500	1000 , 000 , 500	928 , 630 , 779	821 , 834 , **828**	882 , 753 , 817
2	1	797 , 733 , 765	1000 , 000 , 500	657 , 907 , 782	581 , 908 , 745	684 , 881 , **782**
	3	1000 , 000 , 500	1000 , 000, 500	777 , 836 , 806	696 , 903 , 800	711 , 878 , **794**
	5	856 , 650 , 753	1000 , 000 , 500	818 , 804 , 811	793 , 863 , 828	793 , 851 , **822**
	7	1000 , 000 , 500	1000 , 000 , 500	868 , 731 , 780	737 , 872 , 805	847 , 835 , **841**
	9	1000 , 000 , 500	1000 , 000 , 500	917 , 637 , 777	739 , 862 , 801	879 , 739 , **809**
3	1	740 , 761 , 751	1000 , 000 , 500	681 , 872 , **777**	399 , 973, 686	526 , 941 , 733
	3	898 , 563 , 730	1000 , 000 , 500	835 , 814 , **825**	714 , 872 , 793	646 , 912 , 779
	5	1000 , 000 , 500	1000 , 000 , 500	729 , 854 , 792	723 , 885 , 804	731 , 879 , **805**
	7	1000 , 000 , 500	1000 , 000 , 500	1000 , 000 , 500	838 , 820 , **829**	808 , 825 , 816
	9	1000 , 000 , 500	1000 , 000 , 500	1000 , 000 , 500	729 , 883 , 806	834 , 804 , **819**
4	1	870 , 651 , 760	1000 , 000 , 500	730 , 860 , **795**	425 , 957 , 691	1000 , 000 , 500
	3	910 , 604 , 757	1000 , 000 , 500	777 , 850 , **814**	730 , 878 , 804	707 , 904 , 806
	5	924 , 536 , 730	1000 , 000 , 500	779 , 819 , 799	803 , 838 ,**820**	804 , 829 , 817
	7	1000 , 000 , 500	1000 , 000 , 500	1000 , 000 , 500	863 , 725 , 794	809 , 781 , **795**
	9	1000 , 000 , 500	1000 , 000 , 500	1000 , 000 , 500	738 , 851 , 795	854 , 785 , **819**
5	1	658 , 780, 718	1000 , 000 , 500	719 , 868 , **793**	664 , 910, 787	615 , 896 , 756
	3	966 , 403 , 685	1000 , 000 , 500	821 , 797 , **809**	670 , 902 , 786	740 , 868 , 804
	5	927 , 526 , 727	1000 , 000 , 500	877 , 760 , 819	693 , 897 , 795	830 , 815 , **822**
	7	1000 , 000 , 500	1000 , 000 , 500	859 , 776 , **817**	791 , 839 , 815	859 , 759 , 809
	9	1000 , 000 , 500	1000 , 000 , 500	873 , 735 , 803	741 , 885 , 813	904 , 750 , **827**

Nội dung của đề tài 4. Kết quả cho lớp "Cat"

- Bảng phía trên là kết quả thực hành cho bài toán phân loại nhị phân cho lớp "cat".
- Trong thí nghiệm này, hàm thất thoát BCE (entropylossR, R có nghĩa "regularized", $f_{entropyR}(y) = -\ln(1 - y + \epsilon)$) nói chung có kế quả không tốt bằng cutoffloss1 và brokenloss3.
- Hàm thất thoát entropyloss mà không có thêm epsilon ($f_{entropy}(y) = -\ln(1 - y)$) thì khiến máy bị rơi vào NaN nên không học được.
- Hàm thất thoát brokenloss1 hay khiến máy rơi vào bẫy không học được.

CHÂN THÀNH CẢM ƠN!