Học máy: Ghi chú bài giảng

Kyunghyun ChoĐại học New York & Genentech

8 Tháng Năm, 2025

LỜI TỰA

Tôi đã chuẩn bị bài giảng này để dạy DS-GA 1003 "Machine Learn-ing" tại Trung tâm Khoa học Dữ liệu của Đại học New York. Đây là khóa học đầu tiên về học máy dành cho sinh viên thạc sĩ và tiến sĩ về khoa học dữ liệu, và mục tiêu của tôi là cung cấp cho họ một nền tảng vững chắc để họ có thể tiếp tục học các chủ đề tiên tiến và hiện đại hơn về học máy, khoa học dữ liệu cũng như trí tuệ nhân tạo rộng hơn. Vì mục tiêu này, bài giảng này có khá nhiều dẫn xuất toán học của các khái niệm khác nhau trong học máy. Điều này không ngăn cản sinh viên đọc qua ghi chú bài giảng này, vì tôi đã xen kẽ những dẫn xuất này với những giải thích dễ tiếp cận về trực giác và hiểu biết đằng sau những dẫn xuất này. Tất nhiên, khi tôi đang chuẩn bị ghi chú này, nó chỉ trở nên rõ ràng nền tảng của tôi trong học máy nông cạn như thế nào. Nhưng, tôi đã cố gắng.

Khi chuẩn bị bài giảng này, tôi đã cố gắng hết sức để liên tục nhắc nhở bản thân về "Bài học cay đắng" của Richard Sutton [Sutton, 2019]. Tôi buộc bản thân phải trình bày các thuật toán, mô hình và lý thuyết khác nhau theo những cách hỗ trợ triển khai có thể mở rộng, cả cho tính toán và dữ liệu. Do đó, tất cả các thuật toán học máy trong bài giảng này được trình bày để hoạt động với độ dốc ngẫu nhiên và các biến thể của nó. Tất nhiên, có những khía cạnh khác của khả năng mở rộng, chẳng hạn như điện toán phân tán, nhưng tôi hy vọng và hy vọng rằng các khóa học tiếp theo nâng cao hơn sẽ dạy sinh viên các chủ đề nâng cao này dựa trên nền tảng mà khóa học này đã trang bị cho những sinh viên đó.

Mặc dù tôi có ý định bao gồm càng nhiều chủ đề cơ bản càng tốt trong khóa học này, nhưng rõ ràng là một khóa học không đủ dài để đào sâu hơn vào tất cả các chủ đề này. Tôi đã phải đưa ra một quyết định khó khăn là bỏ qua một số chủ đề mà tôi thấy nền tảng, thú vị và thú vị, chẳng hạn như học trực tuyến, phương pháp hạt nhân và cách xử lý các giá trị còn thiếu. Có những chủ đề đẹp khác mà tôi cố tình bỏ qua, mặc dù tôi tin rằng chúng cũng là nền tảng, bởi vì chúng được đề cập rộng rãi trong nhiều khóa học khác, chẳng hạn như mô hình hóa trình tự (hoặc mô hình ngôn ngữ quy mô lớn). Hơn nữa, tôi đã kiềm chế thảo luận về bất kỳ ứng dụng cụ thể nào, hy vọng rằng có các khóa học tiếp theo khác tập trung vào các lĩnh vực ứng dụng riêng lẻ, chẳng hạn như thị giác máy tính, sinh học tính toán và xử lý ngôn ngữ tự nhiên.

Có một vài chủ đề hiện đại hơn mà tôi hy vọng mình có thể đề cập nhưng không thể do thời gian. Để liệt kê một vài trong số đó, chúng bao gồm các mô hình tổng quát dựa trên phương trình vi phân thông thường (ODE) và học tương phản cho cả học đại diện và học số liệu. Có lẽ trong tương lai, tôi có thể tạo một loạt hai khóa học về học máy và thêm các tài liệu bổ sung này. Cho đến lúc đó, học sinh sẽ phải tìm kiếm các tài liệu khác để tìm hiểu về các chủ đề này.

Ghi chú bài giảng này không nhằm mục đích làm sách tham khảo mà được tạo ra để làm tài liệu giảng dạy. Đây là cách tôi xin lỗi trước rằng tôi đã không cẩn thận chút nào khi trích dẫn rộng rãi và đầy đủ tất cả các tài liệu trong quá khứ có liên quan. Tôi hy vọng sẽ thêm các trích dẫn kỹ lưỡng hơn vào lần tới khi tôi dạy cùng một khóa học này, mặc dù không có kế hoạch ngay lập tức để làm như vậy sớm.

Nội dung

Chương 1

Một hàm năng lượng

Một cách thông thường để dạy học máy là trải qua các thiết lập vấn đề khác nhau. Nó thường bắt đầu với phân loại nhị phân, khi perceptron, regres-sion hậu cần và các máy vectơ hỗ trợ được giới thiệu, và tiếp tục với phân loại đa lớp. Tại thời điểm này, người ta thường giới thiệu hồi quy như một phiên bản phân loại liên tục. Thông thường, tại thời điểm này, người ta sẽ tìm hiểu về các phương pháp hạt nhân và mạng nơ-ron, tập trung vào sự lan truyền ngược (một sự phát triển gần đây hơn về mặt giảng dạy học máy.) Đây cũng là một điểm mà người ta sẽ đi đường vòng bằng cách học máy học xác suất, với mục tiêu cuối cùng là giới thiệu cách tiếp cận Bayes đối với học máy, tức là gạt ra ngoài lề hơn tối ưu hóa. Tuy nhiên, nửa sau của khóa học sẽ gần giống với nội dung cho đến nay trong một môi trường không giám sát, nơi chúng ta học được rằng học máy có thể hữu ích ngay cả khi các quan sát không liên quan đến kết quả (nhãn). Người ta sẽ tìm hiểu về nhiều kỹ thuật phân tích ma trận, phân cụm cũng như mô hình tạo xác suất. Nếu giảng viên có tham vọng, họ sẽ lẻn vào một hoặc hai bài giảng về học tăng cường ở phút cuối.

Một vấn đề chính của việc dạy học máy theo cách thông thường như vậy là học sinh cực kỳ bất tiện khi nhìn thấy một nền tảng chung làm nền tảng cho tất cả các kỹ thuật và mô hình khác nhau này. Học sinh thường gặp khó khăn khi thấy việc học có giám sát và không giám sát kết nối với nhau như thế nào. Học sinh thậm chí còn khó khăn hơn khi tìm ra rằng phân loại và phân cụm chỉ đơn giản là hai mặt của cùng một đồng xu. Theo tôi, đơn giản là không thể làm cho đa số sinh viên thấy nền tảng thống nhất đằng sau tất cả các kỹ thuật và mô hình khác nhau này nếu chúng ta gắn bó với việc liệt kê tất cả các mô hình và kỹ thuật này. Do đó, trong khóa học này, tôi cố gắng thực hiện một cách tiếp cận mới để giảng dạy học máy, phần lớn dựa trên và lấy cảm hứng từ một bài hướng dẫn trước đó do Yann LeCun và các đồng nghiệp của ông viết [LeCun et al., 2006]. Ngoài bài hướng dẫn này, cách tiếp cận này vẫn chưa tồn tại và sẽ có hình dạng Tôi tiếp tục viết ghi chú bài giảng này khi khóa học tiếp tục.

Để bắt đầu hành trình này, chúng ta bắt đầu bằng cách xác định một hàm năng lượng, hoặc điểm tương thích âm. Hàm năng lượng e này gán một giá trị thực cho một

cặp của một cá thể quan sát được và một cá thể tiềm ẩn (x, z) và được tham số hóa bởi một vectơ đa chiều θ.

$$e: X \times Z \times \Theta \rightarrow R. \tag{1.1}$$

X là tập hợp tất cả các trường hợp có thể quan sát được, Z là tập hợp tất cả các thực thể tiềm ẩn có thể và Θ là tập hợp tất cả các cấu hình tham số có thể có.

Khi hàm năng lượng thấp (nghĩa là khả năng tương thích cao), chúng ta nói rằng một cặp nhất định (x, z) được ưu tiên cao cho θ. Khi hàm năng lượng cao, không có gì ngạc nhiên khi chúng tôi nói rằng cặp nhất định không được ưu tiên.

Quan sát tiềm ẩn z, như tên cho thấy, không được quan sát trực tiếp. Tuy nhiên, nó đóng một vai trò quan trọng trong việc nắm bắt sự không chắc chắn. Khi chúng ta chỉ quan sát x, chứ không quan sát z, chúng ta không thể xác định đầy đủ x thích hợp như thế nào. Với một tập hợp các giá trị nhất định của z, năng lượng có thể thấp, trong khi nó có thể cao với các giá trị khác của z. Điều này cho chúng ta cảm giác về sự không chắc chắn. Ví dụ, chúng ta có thể tính cả giá trị trung bình và phương sai của năng lượng của một cá thể quan sát được xby

$$e\mu(x, \theta) = E[e(x, z, \theta)] = \sum_{z \in Z} p(z)e(x, z, \theta), \tag{1.2}$$

$$ev(x, \theta) = E[(e(x, z, \theta) - e\mu(x, \theta))2]. \tag{1.3}$$

Với một hàm năng lượng e và tham số θ, chúng ta có thể rút ra nhiều mô hình khác nhau trong học máy bằng cách giảm thiểu hàm năng lượng với việc xem lại các biến khác nhau. Ví dụ, hãy để quan sát được phân chia thành hai phần; đầu vào và đầu ra và giả định rằng không có biến tiềm ẩn, tức là e([x, y], ∅, θ). Với một đầu vào mới x′, chúng ta có thể giải quyết vấn đề học tập được giám sát bằng cách

$$\hat{y} = \arg\min_{y \in Y} e([x', y], \emptyset, \theta), \tag{1.4}$$

trong đó Y là tập hợp tất cả các kết quả có thể xảy ra y. Khi Y bao gồm các mục rời rạc, chúng ta gọi nó là phân loại. Nếu y là một biến liên tục, chúng ta gọi nó là hồi quy.

Khi Z là một tập hợp hữu hạn của các mục rời rạc, một hàm năng lượng nhất định e xác định gán cụm của một quan sát x, dẫn đến phân cụm:

$$\hat{z} = \arg\min_{z \in Z} e(x, z, \theta). \tag{1.5}$$

Nếu z là một biến liên tục, chúng ta sẽ giải quyết cùng một bài toán nhưng gọi nó là học tập phản đối.

Tất cả các mô hình khác nhau này tương ứng hiệu quả với việc giải quyết một vấn đề giảm thiểu đối với một số tập hợp con của đầu vào cho hàm năng lượng e. Nói cách khác, với một đầu vào được quan sát một phần, chúng ta suy ra phần không quan sát được giảm thiểu hàm năng lượng. Đây thường là lý do tại sao mọi người đề cập đến việc sử dụng bất kỳ mô hình học máy nào sau khi đào tạo là suy luận.

Không phải là tầm thường để giải quyết một vấn đề giảm thiểu như vậy. Mức độ khó phụ thuộc vào nhiều yếu tố, bao gồm cả cách xác định hàm năng lượng,

chiều của các biến quan sát cũng như tiềm ẩn cũng như chính các tham số. Trong suốt khóa học, chúng tôi sẽ xem xét các thiết lập khác nhau trong đó các thuật toán tối ưu hóa hiệu quả và hiệu quả được biết đến và được sử dụng để suy luận.

Như cái tên 'học máy' cho thấy, phần lớn học máy đang ước tính θ. Dựa trên những gì chúng ta đã thấy ở trên, có thể bị cám dỗ để nghĩ rằng học tập không là gì ngoài

$$\min_{\theta \in \Theta} E_{x \sim pdata} [e(x, \emptyset, \theta)] , \tag{1.6}$$

khi không có biến tiềm ẩn. Thật không may, việc học không dễ dàng, vì chúng ta phải đảm bảo rằng năng lượng được gán cho quan sát không mong muốn, tức là pdata(x) $\downarrow$, phải tương đối cao. Nói cách khác, chúng ta phải giới thiệu một thuật ngữ bổ sung để quy định hóa việc học:

$$\min_{\theta \in \Theta} E_{x \sim pdata} [e(x, \emptyset, \theta) - R(\theta)] . \tag{1.7}$$

Việc lựa chọn R phải được thực hiện phù hợp với từng vấn đề chúng ta giải quyết và trong suốt khóa học, chúng ta sẽ học cách thiết kế các chuẩn hóa thích hợp để đảm bảo học tập đúng cách.

Tất nhiên nó thậm chí còn trở nên phức tạp hơn khi có các biến tiềm ẩn (không quan sát) z, vì nó đòi hỏi chúng ta cũng phải giải quyết vấn đề suy luận đồng thời. Điều này xảy ra đối với các vấn đề như phân cụm trong đó không xác định cụm của mỗi quan sát và phân tích yếu tố trong đó các yếu tố tiềm ẩn chưa được biết trước. Chúng ta sẽ học cách giải thích các biến tiềm ẩn và thuật toán cho phép chúng ta ước tính θ trong trường hợp không có biến tiềm ẩn.

Tóm lại, có ba khía cạnh đối với mọi vấn đề học máy; (1) xác định một hàm năng lượng E (tham số hóa), (2) ước tính các tham sốθ từ dữ liệu (học tập), và (3) suy ra một phần bị thiếu cho một quan sát một phần (suy luận). Trong ba bước này có một hàm năng lượng và một khi chúng ta có được hàm năng lượng e, chúng ta có thể dễ dàng kết hợp các bước này từ các mô hình học máy khác nhau.

Chương 2

Ý tưởng cơ bản trong MachineLearning với Phân loại

2.1 Phân loại

Trong bài toán phân loại, một quan sát x có thể được chia thành đầu vào và đầu ra; [x, y]. Đầu ra y lấy một trong số lượng hữu hạn của các phạm trù trong Y.Hiện tại, chúng ta giả định rằng không có biến tiềm ẩn, tức là Z = ∅. Suy luận khá tầm thường trong trường hợp này, vì tất cả những gì chúng ta cần làm là chọn phạm trù có năng lượng thấp nhất, sau khi tính năng lượng cho tất cả các phạm vi có thể có một lần:

$$\hat{y}(x) = \arg\min_{y \in Y} e([x, y], \emptyset, \theta). \tag{2.1}$$

Tất nhiên, điều này có thể tốn kém về mặt tính toán nếuY | lớn hoặc x có chiều cao. Chúng ta có thể khắc phục vấn đề này bằng cách khéo léo tham số hóa hàm năng lượng, chẳng hạn như

$$e([x, y], \emptyset, \theta) = 1(y)^{\top} f(x, \theta), \tag{2.2}$$

trong đó 1(y) = [0, . . . , 0, 1, 0, . . . , 0] là một vectơ một nóng. Vectơ một nóng này là tất cả các số không ngoại trừ phần tử thứ y được đặt thành 1.

$f : X \times \Theta \rightarrow R^{|Y|}$ là một trình trích xuất tính năng trả về nhiều giá trị thực như có các danh mục. Với tham số hóa này, chúng ta có thể tính song song các giá trị năng lượng của tất cả các loại. Một ví dụ tương đối đơn giản của f là một hàm tuyến tính, được định nghĩa là

$$f(x, \theta) = W x + b, \tag{2.3}$$

trong đó $\theta = (W, b)$ với $W \in R^{|Y| \times |x|}$ và $b \in R^{|Y|}$. Khi một trình trích xuất tính năng tuyến tính như vậy được sử dụng, chúng tôi gọi nó là bộ phân loại tuyến tính.

Một câu hỏi tự nhiên tiếp theo là làm thế nào chúng ta có thể học các tham số θ (ví dụ: W và b). Chúng tôi tiếp cận việc học từ góc độ tối ưu hóa. Đó là, chúng tôi thiết lập

trước tiên là một hàm tổn thất và tìm ra cách giảm thiểu hàm tổn thất được tính trung bình trên một tập huấn luyện D, trong đó tập huấn luyện D được giả định là bao gồm các quan sát được lấy mẫu Nđộc lập từ phân phối giống hệt nhau (i.i.d.):

$$D = \{[x_n, \text{trong}]\}_{n=1}^{N} . \tag{2.4}$$

Có lẽ hàm mất mát rõ ràng nhất mà chúng ta có thể tưởng tượng là cái gọi là tổn thất zero-one (0-1):

$$L_{0-1}([x; y], \theta) = 1(y \neq \hat{y}(x)), \tag{2.5}$$

đâu

$$\hat{y}(x) = \arg\min_{y' \in Y} e([x, y'], \emptyset, \theta), \tag{2.6}$$

như đã mô tả trước đó (được sao chép ở đây để nhấn mạnh.) 1 (a) là một hàm chỉ báo được định nghĩa là

$$1_{(\text{một})} = \begin{cases} 1, & \text{nếu a là đúng.} \\ 0, & \text{nếu không.} \end{cases} \tag{2.7}$$

Với hàm tổn thất zero-one này, mục tiêu tổng thể của việc học sau đó là

$$\min_{\theta} \frac{1}{\text{là}} \sum_{n=1}^{N} L_{0-1}([x_n, y_n], \theta). \tag{2.8}$$

Bài toán tối ưu hóa này rất khó khăn, bởi vì hầu như không có tín hiệu về cách chúng ta có thể thay đổi dần θ để giảm dần hàm tổn thất. Tổn thất zero-one là một hàm hằng số theo từng phần đối với θ. Nó là 0 hoặc 1, và bất kỳ thay đổi vô cùng nhỏ nào đối với θ không có khả năng thay đổi giá trị tổn thất. Nói cách khác, cách duy nhất để giải quyết vấn đề này là quét qua nhiều (nếu không phải tất cả) giá trị có thể có của θ và xác định giá trị có tổn thất tổng thể thấp nhất. Một cách tiếp cận như vậy được gọi là tối ưu hóa hộp đen, và được biết là nổi tiếng là khó khăn.

2.1.1 Chức năng Perceptron và tổn thất ký quỹ

Thay vào đó, chúng ta có thể đưa ra một proxy cho hàm zero-one loss này, dễ dàng hơn để tối ưu hóa. Chúng ta làm như vậy bằng cách giả định rằng hàm năng lượng có thể phân biệt đối với θ, nghĩa là, ∇θe tồn tại và dễ dàng tính toán.1 Sau đó, chúng ta chỉ cần đảm bảo rằng hàm tổn thất không phải là hằng số theo từng mảnh đối với chính hàm năng lượng.

Chúng ta bắt đầu bằng cách nhận thấy rằng tổn thất zero-one được giảm thiểu (= 0) khi y liên quan đến năng lượng thấp nhất (= $\hat{y}$) trùng với y từ dữ liệu đào tạo. Nói cách khác, tổn thất zero-one được giảm thiểu khi năng lượng liên quan đến

1Chúng ta sẽ sớm thấy tại sao nó được tìm thấy để giả định rằng nó có thể dễ dàng tính toán.

Kết quả thực sự y, tức là e([x, y], ∅, θ), thấp hơn năng lượng liên quan đến bất kỳ y nào khác ′/ = y. Mục tiêu này sau đó có thể được viết ra là thỏa mãn các bất bình đẳng sau:

$$e([x, y], \emptyset, \theta) \leq e([x, \hat{}y'], \emptyset, \theta) - m, \tag{2.9}$$

trong đó m > 0 và

$$\hat{}y' = \underset{\text{arg}}{\text{miny}'\in Y\backslash\{y\}} e([x, y'], \emptyset, \theta). \tag{2.10}$$

Bằng cách sắp xếp lại các thuật ngữ trong sự bất bình đẳng này, chúng ta có được

$$m + e([x, y], \emptyset, \theta) - e([x, \hat{}y'], \emptyset, \theta) \leq 0. \tag{2.11}$$

Để thỏa mãn sự bất đẳng thức này, chúng ta cần giảm thiểu phía bên trái (l.h.s.) cho đến khi nó chạm 0. Chúng ta không cần phải giảm thiểu thêm l.h.s. sau khi đạt 0, vì bất đẳng thức đã được thỏa mãn. Điều này có nghĩa là cái gọi là tổn thất ký quỹ (hoặc tổn thất bản lề):

$$\text{Lmargin}([x, y], \theta) = \max(0, m + e([x, y], \emptyset, \theta) - e([x, \hat{}y'], \emptyset, \theta)). \tag{2.12}$$

Tổn thất này được gọi là tổn thất ký quỹ, bởi vì nó đảm bảo rằng có ít nhất biên độ m giữa các giá trị năng lượng của kết quả chính xác y và kết quả tốt thứ hai ˆy′. Mất biên là trung tâm của ma-chines véc tơ hỗ trợ [Cortes, 1995].

Hãy xem xét trường hợp trong đó m = 0:

$$\text{Lperceptron}([x, y], \theta) = \max(0, e([x, y], \emptyset, \theta) - e([x, \hat{}y'], \emptyset, \theta)). \tag{2.13}$$

Nếu y = ˆy (không phải ˆy′), tổn thất đã được giảm thiểu ở 0, vì

$$e([x, y], \emptyset, \theta) < e([x, \hat{}y'], \emptyset, \theta). \tag{2.14}$$

Nói cách khác, nếu một ví dụ nhất định [x, y] đã được giải chính xác, chúng ta không cần phải thay đổi θ cho ví dụ này. Chúng tôi chỉ cập nhật θ khi y ≠ ˆy. Sự mất mát này được gọi là mất perceptron và có từ những năm 1950 [Rosenblatt, 1958].

2.1.2 Softmax và mất entropy chéo

Thường thuận tiện khi dựa vào khung xác suất, vì nó cho phép chúng ta sử dụng một bộ công cụ lớn được phát triển cho các kỹ thuật thống kê và suy luận xác suất. Như một ví dụ về việc làm như vậy, bây giờ chúng ta sẽ suy ra một classi-fier xác suất từ hàm năng lượng e([x, y], ∅, θ). Bước đầu tiên là biến hàm năng lượng này thành một phân phối phân loại trên Y cho đầu vào x. Giả sử pθ(y|x) là xác suất phân loại của y cho x. Có hai ràng buộc chính phải được đáp ứng:

1. Không tiêu cực: pθ(y|x) ≥ 0 cho tất cả y ∈ Y.

2. Chuẩn hóa: $\sum_{y' \in Y} p_\theta(y'|x) = 1$.

Tất nhiên, có thể có nhiều (nếu không muốn nói là vô hạn) cách khác nhau để mape([x, y], ∅, θ) thành $p_\theta(y|x)$, trong khi thỏa mãn hai điều kiện này [Peters et al., 2019]. Do đó, chúng ta cần áp đặt một ràng buộc hơn nữa để thu hẹp một ánh xạ cụ thể từ hàm năng lượng đến xác suất phân loại. Sự ràng buộc như vậy là tiêu chí entropy tối đa một cách tự nhiên.

Entropy (Shannon) được định nghĩa là

$$H(y|x; \theta) = -\sum_{và \in Y} p_\theta(y|x) \log p_\theta(y|x). \tag{2.15}$$

Entropy lớn nếu có mức độ không chắc chắn lớn. Để đối phó với vấn đề nhật ký 0, chúng tôi giả định rằng

$$H(y|x; \theta) = 0, \text{ nếu } p_\theta(y|x) = \begin{cases} 0 \\ 1 \end{cases}. \tag{2.16}$$

Tại sao điều này là tự nhiên? Bởi vì, đó là cách chúng ta thừa nhận rõ ràng rằng chúng ta không nhận thức đầy đủ về thế giới và có thể có điều gì đó không được biết đến, dẫn đến một số không chắc chắn về sự lựa chọn tiềm năng của chúng ta. Điều này thường được gọi là nguyên lý entropy tối đa [Jaynes, 1957].

Sau đó, chúng ta có thể chuyển đổi các giá trị năng lượng {a1 = e([x, y = 1], ∅, θ), . . . , ad = e([x, y = d], ∅, θ)}được gán cho các lớp kết quả khác nhau Y = {1, 2, . . . , d} thành khả năng xác suất phân loại {p1, . . . , pd} bằng cách giải bài toán tối ưu hóa hạn chế sau:

$$\max_{p1,\ldots,pd} -\sum_{i=1}^{d} \text{AIPI} - \sum_{i=1}^{d} p_i \log p_i \tag{2.17}$$

tuân theo

$$p_i \geq 0, \text{ cho tất cả } i = 1, \ldots, d \tag{2.18}$$

$$\sum_{i=1}^{d} p_i = 1. \tag{2.19}$$

Chúng ta có thể giải quyết bài toán tối ưu hóa này bằng phương pháp Lagrangianmultipliers. Đầu tiên, chúng ta viết hàm mục tiêu không bị ràng buộc:

$$J(p_1, \ldots, p_d, \lambda_1, \ldots, \lambda_d, \gamma) = -\sum_{i=1}^{d} \text{AIPI} - \sum_{i=1}^{d} p_i \log p_i + \sum_{i=1}^{d} \lambda i(p_i - s2i) + \gamma\left(\sum_{i=1}^{d} p_i - 1\right), \tag{2.20}$$

trong đó λ1, . . . , λd và γ là hệ số nhân Lagragian, và s1, . . . , sd là các biến thiên chùng xuống.

Trước tiên, chúng ta hãy tính đạo hàm từng phần của J đối với pi và đặt nó thành 0:

$$\frac{\partial J}{\partial pi} = -ai - \log pi - 1 + \lambda i + \gamma = 0 \tag{2.21}$$

$$\Leftrightarrow \log pi = -ai + \lambda i - 1 + \gamma \tag{2.22}$$

$$\Leftrightarrow pi = \exp(-ai + \lambda i - 1 + \gamma) > 0. \tag{2.23}$$

Chúng ta nhận thấy rằng pi đã lớn hơn 0 tại điểm cực trị này, có nghĩa là ràng buộc đầu tiên pi ≥ 0 đã được thỏa mãn. Chúng ta chỉ có thể đặt λi thành bất kỳ giá trị tùy ý nào và chúng ta sẽ chọn 0, tức là λi = 0 cho tất cả i = 1, . . . , d. Điều này dẫn đến

$$pi = \exp(-ai) \exp(-1 + \gamma). \tag{2.24}$$

Bây giờ chúng ta hãy cắm nó vào ràng buộc thứ hai và giải γ:

$$EXP(-1 + c) \sum_{i=1}^{d} Exp(-ai) = 1 \tag{2.25}$$

$$\Leftrightarrow -1 + c + \log \sum_{i=1}^{d} \exp(-ai) = 0 \tag{2.26}$$

$$\Leftrightarrow \gamma = 1 - \text{nhật ký} \sum_{i=1}^{d} \exp(-ai). \tag{2.27}$$

Bằng cách cắm nó vào pi ở trên, chúng ta nhận được

$$pi = \exp(-ai) \exp(-1 + 1 - \text{nhật ký} \sum_{j=1}^{d} \exp(-aj)) \tag{2.28}$$

$$= \frac{\exp(-ai)}{\sum_{j=1}^{d} \exp(-aj)}. \tag{2.29}$$

Công thức này thường được gọi là softmax [Bridle, 1990].
Bây giờ, chúng ta có xác suất phân loại pi = pθ(y = i|x). Sau đó, chúng ta có thể xác định một hàm khách quan trong khuôn khổ xác suất, như

$$Lce([x, y]; \theta) = -\log p\theta(y|x) = e([x, y], \emptyset, \theta) + \log \sum_{\text{và } 'єY} \exp(-e([x, y'], \emptyset, \theta)). \tag{2.30}$$

Chúng ta thường gọi đây là tổn thất entropy chéo, hoặc khả năng log âm tương đương.

Không giống như tổn thất ký quỹ và perceptron từ trên, nó cung cấp nhiều thông tin hơn

Để xem xét độ dốc của tổn thất entropy chéo:

$$\nabla\theta Lce([x, y], \emptyset, \theta) = \nabla\theta e([x, y], \emptyset, \theta) - X_{\text{và } 'y' \in Y} \frac{\exp(-e([x, y'], \emptyset, \theta))Py'' \in Y \exp(-e([x, y''], \emptyset, \theta))}{\{z\}=p\theta(y'|x)} \nabla\theta e([x, y'], \emptyset, \theta)) \tag{2.31}$$

$$= \nabla\theta e([x, y], \emptyset, \theta) - Ey|x;\theta [\nabla\theta e([x, y'], \emptyset, \theta))] . \tag{2.32}$$

Độ dốc này, hoặc một quy tắc cập nhật vì chúng tôi cập nhật θ theo hướng này, được gọi là học máy Boltzmann [Ackley và cộng sự, 1985].

Có hai thuật ngữ trong quy tắc cập nhật này; (a) các điều khoản tích cực và (b) tiêu cực. Thuật ngữ tích cực tương ứng với việc tăng giá trị năng lượng liên quan đến kết quả thực sự y.2 Thuật ngữ tiêu cực tương ứng với việc giảm các giá trị năng lượng liên quan đến tất cả các kết quả có thể xảy ra, nhưng chúng được tính trọng số tùy theo khả năng chúng nằm dưới các thông số hiện tại.

Chúng ta hãy xem xét thuật ngữ phủ định cẩn thận hơn một chút:

$$- X_{\text{và } 'y' \in Y} \frac{\exp(-\beta e([x, y'], \emptyset, \theta))Py'' \in Y \exp(-\beta e([x, y''], \emptyset, \theta)) \nabla\theta e([x, y'], \emptyset, \theta)).}{} \tag{2.33}$$

β đã được thêm vào để giúp phân tích của chúng tôi dễ dàng hơn. Chúng ta thường gọi β là nhiệt độ nghịch đảo.β mặc định là 1, nhưng bằng cách thay đổi β, chúng ta có thể hiểu rõ hơn về thuật ngữ tiêu cực.

Hãy xem xét trường hợp trong đó β = 0, số hạng âm giảm xuống

$$- \frac{1}{|Y|} \sum_{y' \in Y} \nabla\theta e([x, y'], \emptyset, \theta)). \tag{2.34}$$

Điều này sẽ tương ứng với việc tăng năng lượng liên quan đến mỗi kết quả như nhau.

Còn khi β → ∞ thì sao? Trong trường hợp đó, số hạng âm giảm xuống

$$-\nabla\theta e([x, \hat{y}], \emptyset, \theta), \tag{2.35}$$

đâu

$$\hat{y} = \arg\min_{y \in Y} e([x, y], \emptyset, \theta). \tag{2.36}$$

Khi β → ∞, chúng ta kết thúc với hai trường hợp. Đầu tiên, bộ phân loại đưa ra dự đoán chính xác; $\hat{y} = y$. Trong trường hợp này, các số hạng dương và âm triệt tiêu lẫn nhau và không có gradient. Do đó, không có cập nhật cho tham số. Điều này nhắc nhở chúng ta về sự mất mát perceptron từ phần trước. Mặt khác, nếu $\hat{y} \neq y$, nó sẽ cố gắng giảm giá trị năng lượng liên quan đến

2Hãy nhớ lại rằng đây là một tổn thất được giảm thiểu.

kết quả chính xác y trong khi tăng giá trị năng lượng liên quan đến dự đoán hiện tại ˆy. Điều này tiếp tục cho đến khi dự đoán khớp với kết quả chính xác.

Hai trường hợp cực đoan này cho chúng ta biết điều gì xảy ra với sự mất entropy chéo. Tuy nhiên, nó nhẹ nhàng điều chỉnh các giá trị năng lượng liên quan đến tất cả các kết quả có thể xảy ra dựa trên khả năng chúng là dự đoán. Sự mất entropy chéo đã trở thành tiêu chuẩn trên thực tế ít nhiều khi nói đến việc đào tạo một mạng nơ-ron trong những năm gần đây.

2.2 Lan truyền ngược

Khi bạn quyết định hàm thua lỗ, đã đến lúc chúng ta đào tạo một bộ phân loại để giảm thiểu tổn thất trung bình. Khi làm như vậy, một trong những cách tiếp cận hiệu quả nhất là đi xuống dốc beenstochastic, hoặc biến thể của nó. Độ dốc ngẫu nhiên, mà chúng ta sẽ thảo luận sâu hơn sau, lấy một tập hợp con của các phiên bản đào tạo từ D, tính toán và tính trung bình độ dốc của sự mất mát của từng trường hợp trong tập hợp con này và cập nhật các tham số theo hướng âm của gradient ngẫu nhiên này. Điều này làm cho nó vừa thú vị vừa quan trọng đối với chúng ta để nghĩ về cách tính toán độ dốc của một hàm mất mát.

Chúng ta hãy xem xét cả tổn thất ký quỹ và mất entropy chéo, vì không có gradient có ý nghĩa của hàm tổn thất zero-one và tổn thất perceptron là một trường hợp đặc biệt của tổn thất ký quỹ:

$$\nabla_\theta \text{Lmargin}([x, y], \theta) = \begin{cases} (\nabla_\theta e([x, y], \emptyset, \theta) - \nabla_\theta e([x, \hat{y}'], \emptyset, \theta), & \text{nếu Lmargin}([x, y], \theta) > 0. \\ 0, & \text{Khác.} \end{cases} \quad (2.37)$$

$$\nabla_\theta \text{Lce}([x, y], \theta) = \nabla_\theta e([x, y], \emptyset, \theta) - E_{y|x;\theta}\left[\nabla_\theta e([x, y'], \emptyset, \theta))\right]. \quad (2.38)$$

Trong cả hai trường hợp, gradient của hàm năng lượng hiển thị: $\nabla_\theta e([x, y], \emptyset, \theta))$. Do đó, chúng tôi tập trung vào độ dốc của hàm năng lượng trong trường hợp này.

2.2.1A Hàm năng lượng tuyến tính

Chúng ta hãy bắt đầu với một trường hợp rất đơn giản mà chúng ta đã xem xét trước đó. Chúng ta giả định rằng x là vectơ có giá trị thực của kích thước d, tức là $x \in R^d$. Chúng ta sẽ giả định thêm rằng y lấy một trong K giá trị tiềm năng, tức là $y \in \{1, 2, \ldots, K\}$. Các tham số θ bao gồm

1. Ma trận trọng lượng $W = \begin{bmatrix} w_1 \\ w_2 \\ \ldots \\ w_K \end{bmatrix} \in R^{K \times d}$

2. Vectơ thiên vị $b = \begin{bmatrix} B1 \\ B2 \\ \ldots \\ Bk \end{bmatrix} \in R^K$

Bây giờ chúng ta có thể định nghĩa hàm năng lượng là

$$e([x, y], \emptyset, \theta) = -w^{\top}y\ x - bằng. \tag{2.39}$$

Độ dốc của hàm năng lượng đối với vectơ trọng lượng liên quan wy sau đó là

$$\nabla E = -x. \tag{2.40}$$

Tương tự, đối với thiên vị:

$$\frac{\partial e}{\partial by} = -1. \tag{2.41}$$

Cái đầu tiên (gradient w.r.t. wy) nói rằng để năng lượng được giảm xuống cho sự kết hợp cụ thể này (x, y), chúng ta nên thêm đầu vào x vào vectơ trọng lượng wy. Cái thứ hai (gradient wrt by) làm giảm năng lượng cho kết quả y bất kể đầu vào.

Chúng ta hãy xem xét tổn thất perceptron hoặc tổn thất ký quỹ với ký quỹ bằng không. Gradient kỳ đầu tiên, $\nabla\theta e([x, y], \emptyset, \theta)$, cập nhật vectơ trọng số và giá trị thiên vị liên quan đến kết quả chính xác. Với tốc độ học $\eta > 0$, năng lượng cập nhật liên quan đến kết quả chính xác, nơi chúng ta tuân theogradient âm,3 sau đó nhỏ hơn hàm năng lượng ban đầu:

$$-(wy + \eta x)^{\top}x - (bằng + \eta) = -\ w^{\top}y\ x - bằng - \eta(\|x\|2 + 1) \tag{2.42}$$

$$= e([x, y], \emptyset, \theta) - \eta(\|x\|2 + 1) \tag{2.43}$$

$$< e([x, y], \emptyset, \theta). \tag{2.44}$$

Đây chính xác là những gì chúng tôi dự định, vì chúng tôi muốn giá trị năng lượng thấp hơn với sự kết hợp tốt giữa đầu vào và kết quả.

Tuy nhiên, điều này thôi là không đủ như một quy tắc học tập đầy đủ. Ngay cả khi giá trị năng lượng liên quan đến sự kết hợp phù hợp được hạ xuống, nó có thể không được hạ thấp đủ, do đó kết quả chính xác được chọn khi đầu vào được trình bày lại. Gradient thuật ngữ thứ hai khen ngợi điều này bằng cách có dấu hiệu ngược lại phía trước nó. Bằng cách đi theo gradient âm của năng lượng âm liên quan đến đầu vào và kết quả dự đoán $\hat{y}$, chúng tôi đảm bảo rằng giá trị năng lượng cụ thể này được tăng lên:

$$-(w\hat{y} - \eta x)^{\top}x - (b\hat{y} - \eta) = e([x, \hat{y}], \emptyset, \theta) + \eta(\|x\|2 + 1) \tag{2.45}$$

$$> e([x, \hat{y}], \emptyset, \theta). \tag{2.46}$$

Vì vậy, quy tắc học tập này sẽ làm giảm giá trị năng lượng liên quan đến kết quả chính xác và tăng giá trị năng lượng liên quan đến kết quả dự đoán không chính xác, cho đến khi kết quả có năng lượng thấp nhất trùng với kết quả chính xác. Khi điều đó xảy ra, sự mất mát là liên tục, và không có học hỏi nào xảy ra, bởi vì $y = \hat{y}$.

3Chúng ta sẽ sớm thảo luận lý do tại sao chúng ta làm như vậy ở phần sau của chương này.

Tại thời điểm này, chúng ta bắt đầu thấy rằng đạo hàm và đối số ở trên áp dụng như nhau cho x, đầu vào. Thay vì độ dốc của năng lượng w.r.t. vectơ trọng lượng wy, nhưng chúng ta cũng có thể tính toán w.r.t. đó là x đầu vào:

$$\nabla xe = -wy,$$

Giả sử rằng x là liên tục và hàm năng lượng có thể vi phân được W.R.T. x.By theo gradient (ngược lại) trong không gian đầu vào, chúng ta có thể thay đổi hàm mất, thay vì sửa đổi các vectơ trọng số và sai lệch.

Tất nhiên điều này hoàn toàn trái ngược với những gì chúng ta đang cố gắng làm ở đây, vì mục tiêu chính là tìm một bộ phân loại phân loại một đầu vào nhất định x vào đúng danh mục y. Tuy nhiên, quan điểm này tự nhiên dẫn chúng ta đến ý tưởng về lan truyền ngược [Rumelhart et al., 1986].

2.2.2A Hàm năng lượng phi tuyến

Thay vì điều chỉnh vectơ trọng lượng W và vectơ thiên vị b, chúng ta có thể điều chỉnh đầu vào x trực tiếp để sửa đổi giá trị năng lượng liên quan. Cụ thể hơn, với tổn thất perceptron, đó là tổn thất ký quỹ với biên độ bằng không, khi dự đoán không chính xác, tức là $y / = \hat{}y$, độ dốc của tổn thất perceptron đối với đầu vào x là4

$$\nabla xLperceptron([x, y], \theta) = \nabla xe([x, y], \theta) - \nabla xe([x, \hat{}y], \theta) = -wy + w\hat{}y. \quad (2.47)$$

Tương tự như ma trận trọng lượng và vectơ thiên vị ở trên, nếu chúng ta cập nhật đầu vàox theo hướng ngược lại với hướng này, chúng ta có thể tăng giá trị năng lượng liên quan đến kết quả chính xác y trong khi giảm giá trị năng lượng với kết quả được dự đoán không chính xác $\hat{}y$. Mặc dù điều này nói chung là vô dụng với hàm năng lượng tuyến tính, như chúng ta vừa thảo luận, đây là một thí nghiệm tư duy thú vị, vì điều này cho chúng ta biết rằng chúng ta có thể giải quyết vấn đề bằng cách điều chỉnh các tham số, tức là ma trận trọng số và vectơ thiên vị, hoặc bằng cách điều chỉnh chính các điểm dữ liệu đầu vào. Cái sau nghe có vẻ như là một sự thay thế hấp dẫn, bởi vì nó sẽ giúp chúng ta thoát khỏi bị ràng buộc bởi tính tuyến tính của hàm năng lượng.

Tuy nhiên, có một vấn đề lớn với lựa chọn thứ hai. Nghĩa là, chúng ta không biết làm thế nào để thay đổi đầu vào mới trong tương lai (không được bao gồm trong tập huấn luyện), vì một đầu vào mới như vậy có thể không đi kèm với kết quả chính xác liên quan. Do đó, chúng ta cần xây dựng một hệ thống dự đoán đầu vào bị thay đổi sẽ được cung cấp một đầu vào mới trong tương lai.

Để khắc phục vấn đề này, chúng ta bắt đầu bằng cách sử dụng một số phép biến đổi h của đầu vào x, với các tham số riêng của nó θ′, thay vì đầu vào ban đầu x Đó là, $h = F(x, \theta')$. Tương tự, chúng tôi đề cập đến đầu vào mới được cập nhật bởi $\hat{}h$. Chúng ta thu được $\hat{}h$ bằng cách đi theo hướng gradient từ Phương trình (2.47). Bây giờ chúng ta định nghĩa một hàm năng lượng mới e′ sao cho tổ hợp $(h, \hat{}h)$ được gán một năng lượng thấp hơn các tổ hợp khác nếu h và $\hat{}h$ gần nhau. Theo hàm năng lượng này, năng lượng thấp nếu phép biến đổi này của đầu vào $h = F(x, \theta')$ là

4Khi rõ ràng rằng không có biến z tiềm ẩn (không quan sát), tôi sẽ bỏ qua φ để ngắn gọn.

tương tự như đầu vào ˆh được cập nhật. Điều này có ý nghĩa một cách trực quan, vì ˆh là phép biến đổi mong muốn của x đầu vào, vì nó làm giảm hàm tổn thất tổng thể ở trên.

Một ví dụ điển hình của một hàm năng lượng như vậy sẽ là

$$e'([h, \hat{h}], \theta') = \frac{1}{2} \| \sigma(U^T x + c) - \hat{h} \|_2, \quad (2.48)$$

trong đó u và c lần lượt là ma trận trọng lượng và vectơ thiên vị, và σ là hàm phi tuyến tùy ý. h = σ(U Tx + c) sẽ là một số biến đổi của x đầu vào, như đã mô tả ở trên.

Hàm mất trong trường hợp này có thể đơn giản là chính hàm năng lượng:

$$L_{I2}([h, \hat{h}], \theta') = e'([h, \hat{h}], \theta'). \quad (2.49)$$

Độ dốc của hàm mất w.r.t. ma trận biến đổi U sau đó là:

$$\nabla U = x (h - \hat{h}) \odot h'^T \quad (2.50)$$

đâu

$$h' = \sigma'(U^T x + c) \quad (2.51)$$

với $\sigma'(a) = \frac{\partial \sigma}{\partial a}(a)$, theo quy tắc chuỗi của các công cụ phái sinh. ⊙ biểu thị phép nhân theo nguyên tố. Tương tự, gradient w.r.t. vectơ thiên vị c là

$$\nabla c = (h - \hat{h}) \odot h'. \quad (2.52)$$

Trước khi tiếp tục thêm, chúng ta hãy xem xét các gradient này. Nếu chúng ta nhìn vào ∇c, số hạng đầu tiên, hoặc phủ định của nó, vì chúng ta muốn giảm thiểu năng lượng, nói rằng chúng ta nên thay đổi c về phía ˆh khỏi h. Nếu h xa ˆh hơn, chúng ta cần thay đổi c nhiều hơn. Số hạng thứ hai h′ được nhân với (h − ˆh). Thuật ngữ h ′ này là độ dốc của hàm kích hoạt phi tuyến σ ở đầu vào hiện tại U Tx + c. Nếu độ dốc dương, chúng ta nên cập nhật c theo dấu ˆh−h , như thường lệ. Nhưng, nếu độ dốc là âm, chúng ta nên lật hướng cập nhật của c, vì tăng c sẽ dẫn đến giảm ˆh − h.

Để phân tích gradient wrt U , chúng ta hãy xem xét gradient wrt một phần tử cụ thể của U , tức là uij. UIJ có thể được coi là trọng số giữa kích thước thứ i của đầu vào, xi và chiều thứ j của phép biến đổi, hj. Gradient này được viết ra dưới dạng

$$\frac{\partial}{\partial u_{ij}} = x_i(h_j - \hat{h}_j)h'_j = (x_i h_j - x_i \hat{h}_j)h.j. \quad (2.53)$$

Chúng ta đã biết h′j làm gì: nó quyết định độ dốc là dương hay âm, và do đó liệu hướng cập nhật có nên lật ngược hay không. Bởi vì chúng ta đi theo hướng ngược lại (vì chúng ta muốn giảm năng lượng), số hạng đầu tiên

XIHJ được trừ khỏi UIJ. Thuật ngữ này cho chúng ta biết giá trị của xi được phản ánh mạnh mẽ như thế nào trên giá trị của hj. Vì hj hiện đang được cập nhật, ảnh hưởng của xi trên chiều thứ j của phép biến đổi thông qua uij phải được giảm bớt. Mặt khác, thuật ngữ thứ hai xiˆhj làm ngược lại. Nó nói rằng ảnh hưởng của xi đối với chiều thứ j của phép biến đổi, theo giá trị mới cập nhật ˆhj, phải được phản ánh nhiều hơn trên uij. Nếu giá trị mới của chiều thứ j có cùng dấu với xi, uij sẽ có xu hướng về giá trị dương. Nếu không, nó sẽ có xu hướng hướng tới giá trị âm.

Bây giờ chúng ta có thể tưởng tượng một quy trình mà chúng ta xen kẽ giữa tính toán ˆh và cập nhật u và c để khớp ˆh. Tất nhiên, quy trình này có thể không tối ưu, vì không có gì đảm bảo (hoặc khó có được bất kỳ đảm bảo nào) rằng việc liên tục cập nhật u và c theo độ dốc của hàm năng lượng thứ hai dẫn đến cải thiện tổn thất tổng thể khi h = σ (U ⊤x + c) được sử dụng thay cho mục tiêu ˆh. Khi hàm năng lượng thứ hai thực sự được giảm thiểu sao cho σ(U ′⊤x + c′) trùng với ˆh, tổn thất sẽ nhỏ hơn so với ban đầuh = σ(U ⊤x + c). Tuy nhiên, không rõ liệu tổn thất có nhỏ hơn cho đến khi đạt được mức tối thiểu này hay không.

Thay vào đó, chúng ta có thể nghĩ đến một quy trình trong đó chúng ta cập nhật trực tiếp u và c mà không tạo ra ˆh như một đại lượng trung gian. Giả sử chúng ta chỉ cần một bước đơn vị để cập nhật ˆh:

$$\hat{h} = h + (wy - w\hat{y}) \tag{2.54}$$

$$\Longleftrightarrow \hat{h} - h = -\nabla_h L(h). \tag{2.55}$$

Nghĩa là, chúng ta sử dụng tốc độ học tập (hoặc kích thước bước) là 1.

Sau đó

$$\nabla U = x\,(\nabla_h L(h) \odot h')^\top \tag{2.56}$$

$$\nabla c = \nabla_h L(h) \odot h'. \tag{2.57}$$

Nói cách khác, chúng ta có thể bỏ qua tính toán ˆh và trực tiếp tính toán các gradient của tổn thất w.r.t. u và c bằng cách sử dụng gradient w.r.t. h.

Cũng giống như những gì chúng ta đã làm với h (hoặc ban đầu x), chúng ta có thể kiểm tra cách chúng ta sẽ thay đổi x mới này để giảm thiểu hàm năng lượng thứ hai e′. Điều này được thực hiện bằng cách tính toán độ dốc của e′ wrt x:

$$\nabla x = U\,(h - \hat{h}) \odot h', \tag{2.58}$$

tương tự như gradient wrt U . Nếu chúng ta thay thế (h − ˆh) bằng ∇hL(h), chúng ta

$$\nabla x = U\,(\nabla_h L(h) \odot h'). \tag{2.59}$$

Đây là lần thứ ba chúng tôi thảo luận về nó, nhưng vâng, chúng tôi biết h′ làm gì ở đây: nó quyết định dấu hiệu của bản cập nhật. Nếu chúng ta bỏ qua h′ bằng cách đơn giản giả định rằng σ

ví dụ như một bản đồ nhận dạng (có nghĩa là h′ = 1), chúng ta nhận ra rằng ∇x là phép biến đổi tuyến tính của ∇h 5, như

$$\nabla x = U \nabla h. \tag{2.60}$$

Đối chiếu nó với thuật ngữ màu đỏ bên dưới:

$$h = \sigma (U \top x + c) \tag{2.61}$$

Thuật ngữ màu đỏ ở trên có thể được coi là truyền tín hiệu đầu vào xvia U ⊤ đến h. Ngược lại, U ∇h có thể được coi là truyền ngược tín hiệu lỗi ∇h qua U đến đầu vào x.
Bạn phải xem chúng ta đang hướng đến đâu bây giờ. Chúng ta hãy thay thế x một lần nữa, lần này, bằng z. Nói cách khác,

$$h = \sigma(U \top z + c)$$

và

$$z = \sigma (V \top x + s).$$

Tương tự chúng ta có thể giới thiệu một hàm năng lượng khác e ″ được định nghĩa là

$$e''([z, \hat{z}], \theta'') = \underline{1}2 \|z - \hat{z}\|2, \tag{2.62}$$

đâu

$$\hat{z} = z - \nabla z \tag{2.63}$$

$$= z - U \nabla h. \tag{2.64}$$

Theo các bước dẫn xuất chính xác từ trên, chúng ta kết thúc với

$$\nabla V = x (\nabla z \odot z')\top \tag{2.65}$$

$$\nabla s = \nabla z \odot z', \tag{2.66}$$

đâu

$$\nabla z = U \nabla h. \tag{2.67}$$

Trong một lần quét duy nhất, chúng ta có thể lan truyền ngược tín hiệu lỗi từ chức năng mất mát trở lại x và tính toán độ dốc của hàm mất w.r.t.tất cả các tham số, W, b, U, c, V và s. Tất nhiên, khi làm như vậy, chúng tôi phải lưu trữ cái gọi là vectơ kích hoạt chuyển tiếp, x, z và h, thường được gọi là sổ sách.

Quá trình tính toán độ dốc của chức năng tổn thất với tất cả các pa-rameter từ nhiều giai đoạn biến đổi phi tuyến của đầu vào được gọi là

5Bất cứ khi nào rõ ràng, chúng tôi sẽ bỏ một số điều ngắn gọn và làm rõ. Trong trường hợp này, ∇h là ∇hL(h).

hỗ trợ ngược. Điều này có thể được khái quát hóa cho bất kỳ biểu đồ tính toán nào mà không có bất kỳ vòng lặp nào (mặc dù, các vòng lặp có thể được mở ra cho một số chu kỳ hữu hạn trong thực tế) và là một trường hợp đặc biệt của vi phân tự động [Baydin và cộng sự, 2018], được gọi là vi phân tự động chế độ ngược.

Bởi vì sự phân biệt tự động ở chế độ đảo ngược hiệu quả cả về tính toán và bộ nhớ (cả tuyến tính), nó được sử dụng phổ biến để tính toán gradient và được thực hiện tốt trong nhiều công cụ học sâu được sử dụng rộng rãi, chẳng hạn như PyTorch và Jax. Tính phổ quát này ngụ ý rằng một khi chúng ta quyết định về hàm tổn thất và một hàm năng lượng sao cho hàm tổn thất có thể phân biệt đượcw.r.t. các tham số của hàm năng lượng, chúng ta có thể đơn giản giả định độ dốc sẽ có sẵn.

2.3 Độ dốc ngẫu nhiên

Khi chúng ta đã xác định một hàm năng lượng và một hàm tổn thất liên quan, chúng ta có thể tính toán độ dốc của hàm mất mát này với các tham số. Với gradient, chúng ta có thể cập nhật các tham số nhiều lần để chúng ta có thể giảm thiểuhàm mất. Điều quan trọng là phải quan sát rằng chúng ta đã xác định hàm mất cho từng ví dụ đào tạo riêng lẻ và cuối cùng mục tiêu của chúng ta trở thành giảm thiểu mức trung bình của sự mất mát của tất cả các ví dụ đào tạo. Vì một lý do ngẫu nhiên, chúng ta sẽ sử dụng fi(θ) để biểu thị hàm tổn thất của ví dụ thứ i tại θ, và do đó tổn thất tổng thể là

$$f(\theta) = \frac{1}{N}\sum_{i=1}^{N} fi(i). \tag{2.68}$$

Khi tổn thất tổng thể là trung bình (hoặc tổng) của các hàm tổn thất riêng lẻ, chúng ta nói rằng tổn thất có thể phân hủy.
Chúng ta có thể xem một hàm tổn thất tổng thể như tính toán hàm tổn thất độc lập dự kiến:

$$f(\theta) = Ei\,[fi(\theta)]\,, \tag{2.69}$$

trong đó i ∼ U(1, . . . , N). Tất nhiên, chúng ta có thể thay thế phân phối đồng nhất này bằng một phân phối dữ liệu tùy ý và viết điều này như

$$f(\theta) = Ex{\sim}pdata\,[f(x;\theta)]\,, \tag{2.70}$$

mặc dù bây giờ chúng ta sẽ gắn bó với chỉ mục thống nhất trên tập huấn luyện.

Với phương trình (2.69), chúng ta cũng nhận được

$$\nabla f = Không\,[\nabla fi]\,, \tag{2.71}$$

bởi vì kỳ vọng về một biến ngẫu nhiên hữu hạn, rời rạc có thể được ghi lại bằng cách sử dụng tổng hữu hạn.

Có hai hằng số chúng ta nên xem xét khi quyết định làm thế nào để giảm thiểu f wrt θ. Đó là số lượng ví dụ đào tạo N và

số lượng tham số dim(θ) (nếu không khó hiểu, chúng ta sẽ sử dụng dim(θ) và |θ|thay thế cho nhau.) Chúng ta hãy bắt đầu với cái sau |θ|. Nếu số lượng tham số lớn, chúng ta không thể mong đợi tính toán bất kỳ thông tin đạo hàm bậc cao nào của hàm f ngoài đạo hàm bậc nhất, đó là độ dốc của nó. Nếu không có quyền truy cập vào đạo hàm bậc cao hơn, chúng ta không thể hưởng lợi từ thuật toán tối ưu hóa nâng cao, chẳng hạn như thuật toán của Newton. Thật không may, trong học máy hiện đại,|θ| có thể lớn tới hàng chục tỷ và chúng ta thường bị mắc kẹt với các thuật toán tối ưu hóa bậc nhất.

Nếu N lớn, nó ngày càng trở nên nặng nề để tính toán f không phải ∇f trực tiếp mỗi lần cập nhật. Nói cách khác, chúng ta chỉ có thể mong đợi sử dụng gradient thực của f chỉ khi chỉ có ít ví dụ đào tạo, tức là N nhỏ. Trong máy học hiện đại, chúng ta thường phải đối mặt với hàng trăm nghìn, nếu không muốn nói là hàng triệu hoặc hàng tỷ, các ví dụ đào tạo và chúng ta thường không thể tính toán chính xác tổn thất tổng thể. Nói tóm lại, chúng ta đang ở trong tình huống mà chúng ta thậm chí không thể sử dụng thông tin gradient đầy đủ, đúng để cập nhật các tham số.

Để đối phó với N lớn và |θ|, chúng ta thường sử dụng ước tính độ dốc ngẫu nhiên thay vì gradient đầy đủ, trong đó gradient ngẫu nhiên được định nghĩa là

$$git = \nabla \text{phù hợp } (\theta t), \tag{2.72}$$

nơi nó được rút ra từ phân bố đồng đều trên {1, . . . , N }. Sau đó, chúng tôi cập nhật các tham số bằng cách sử dụng ước tính gradient ngẫu nhiên này bằng cách

$$\theta t+1 = \theta t - \alpha t git. \tag{2.73}$$

Khi làm như vậy, một thông lệ là duy trì một tập hợp cái gọi là trạm kiểm soát và chọn một tập hợp tốt nhất trong bộ trạm kiểm soát này. Chúng ta sẽ thảo luận về cách chúng ta chọn điểm kiểm tra tốt nhất theo tiêu chí nào trong phần tiếp theo chi tiết hơn, vì đây là nơi tối ưu hóa và học tập khác nhau.

Bây giờ, chúng ta hãy gắn bó với tối ưu hóa và đặc biệt là tối ưu hóa lặp đi lặp lại. Khi nghĩ về tối ưu hóa, có hai khái niệm riêng biệt quan trọng như nhau. Đầu tiên là hội tụ. Với sự hội tụ, chúng tôi có nghĩa là liệu tối ưu hóa lặp lại có dần dần di chuyển θt lặp lại đến một giải pháp mong muốn. Một giải pháp mong muốn có thể là tối thiểu toàn cục (nếu có), bất kỳ localminimum nào hoặc bất kỳ cực trị nào (trong đó gradient bằng không.) Điều quan trọng là phải biết liệu lặp lại có hội tụ đến một giải pháp mong muốn như vậy hay không và nếu có, ở tốc độ nào. Khái niệm quan trọng thứ hai là thuộc tính đi xuống. Một thuật toán tối ưu hóa lặp đi lặp lại là hạ thấp nếu nó luôn tiến bộ, tức là f (θt + 1) ≤ f (θt) cho allt.

Như chúng ta sẽ tìm hiểu về nó ngay trong phần tiếp theo, giải pháp mong muốn không được xác định với hàm tổn thất tổng thể f . Thay vào đó, giải pháp mong muốn forus được định nghĩa bằng cách sử dụng một hàm khác f ∗. Hàm f ∗ này tương tự như f hầu như ở mọi nơi trên θ nhưng hai hàm này khác nhau. Do đó, chúng ta nên liệt kê một loạt θt với các giá trị f (θt) nhỏ và cuối cùng chọn một bằng cách sử dụng f ∗(θt). Nói cách khác, nó không phải là sự hội tụ mà là thuộc tính hậu duệ.

2.3.1Hạ nguồn Lemma

Chúng tôi bắt đầu bằng cách nêu và chứng minh một trong những kết quả cơ bản nhất trong việc tối ưu hóa, được gọi là lemma đi xuống. Theo lemma đi xuống, bất đẳng thức tiếp theo có hiệu lực khi ∇f là một hàm liên tục L-Lipschitz, tức là ‖∇f (x) − ∇f (y)‖ ≤ L‖x − y‖:

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|2. \tag{2.74}$$

Bất đẳng thức này cho phép chúng ta giới hạn trên giá trị của một hàm tại một điểm cho giá trị cũng như gradient tại một điểm x khác.

Giả sử g(t) = f (x + t(y − x)) sao cho g(0) = f (x) và g(1) = f (y). Sau đó

$$f(y) - f(x) = g(1) - g(0) = \int_0^1 g'(u)\,du = \int_0^1 \nabla f(x + t(y - x))^\top (y - x)dt. \tag{2.75}$$

Bằng cách trừ ∇f (x)⊤(y − x) từ cả hai bên, chúng ta nhận được

$$f(y) - f(x) - \nabla f(x)^\top (y - x) = \int_0^1 (\nabla f(x + t(y - x)) - \nabla f(x))^\top (y - x)dt. \tag{2.76}$$

Chúng ta có thể giới hạn nó bằng cách sử dụng bất đẳng thức Cauchy-Schwarz, tức là a⊤b ≤ ‖a‖‖b‖:

$$f(y) - f(x) - \nabla f(x)^\top (y - x) \leq \int_0^1 \|\nabla f(x + t(y - x)) - \nabla f(x)\|\|y - x\|dt. \tag{2.77}$$

Chúng ta có thể sử dụng giả định ở trên rằng ∇f là một hàm L-Lipschitz để đơn giản hóa nó thành

$$f(y) - f(x) - \nabla f(x)^\top (y - x) \leq \int_0^1 Lt\|y - x\|2dt = \frac{L}{2}\|y - x\|2, \tag{2.78}$$

lần lượt

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|2. \tag{2.79}$$

Nếu chúng ta giả định rằng N không quá lớn, chúng ta có thể tính toán chính xác gradient và cập nhật các tham số theo hướng gradient âm:

$$\theta_{t+1} = \theta_t - \alpha_t \nabla f(\theta_t) \tag{2.80}$$

$$\Longleftrightarrow \theta_{t+1} - \theta_t = -\alpha_t \nabla f(\theta_t) \tag{2.81}$$

Chúng ta hãy cắm (θt, θt+1) vào (x, y) trong lemma đi xuống:

$$f(\theta_{t+1}) \leq f(\theta_t) - \alpha_t\|\nabla f(\theta_t)\|2 + \frac{\alpha 2_t L}{2}\|\nabla f(\theta_t)\|2 \tag{2.82}$$

$$= f(\theta_t) - (\alpha_t - \frac{L}{2}\alpha 2_t)\|\nabla f(\theta_t)\|2. \tag{2.83}$$

Vì ‖∇f (θt)‖2 ≥ 0, chúng ta muốn tìm αt tối đa hóa − L2 α2t + αt. Chúng ta chỉ cần tính toán đạo hàm của biểu thức wrt αt này và đặt nó về không:

$$-L\alpha t + 1 = 0 \Longleftrightarrow \alpha t = 1L . \quad (2.84)$$

Nói cách khác, nếu chúng ta đặt tốc độ học tập thành 1/L (nghĩa là tỷ lệ nghịch với tốc độ thay đổi hàm số), chúng ta có thể đạt được tiến bộ nhiều nhất mỗi lần. Tất nhiên, điều này không áp dụng trực tiếp cho trường hợp ngẫu nhiên, vì dòng chảy không áp dụng cho ước tính độ dốc ngẫu nhiên như nó vốn có.

2.3.2Độ dốc ngẫu nhiên

Tiếp tục từ lemma đi xuống ở trên, chúng ta sẽ sử dụng quy tắc stochastic gradientupdate từ Eq. (2.73). Hãy trình bày lại quy tắc gradient ngẫu nhiên:

$$\theta t+1 = \theta t - \alpha t g i t \Longleftrightarrow \theta t+1 - \theta t = -\alpha t g i t. \quad (2.85)$$

Cắm (θt, θt+1) vào lemma đi xuống, chúng ta nhận được

$$f(\theta t+1) \le f(\theta t) - \alpha t \nabla f(\theta t)^{\top} g i t + \alpha 2t \, L2 \, \|g i t\|2. \quad (2.86)$$

Chúng tôi quan tâm đến sự tiến bộ dự kiến ở đây ∼ U(1, . . . , N):

$$E[f(\theta t+1)] \le f(\theta t) - \alpha t \nabla f(\theta t)^{\top} E[g i t] + \alpha 2t \, L2 \, E\|g i t\|2 \quad (2.87)$$

$$= f(\theta t) - \alpha t \|\nabla f(\theta t)\|2 + \alpha 2t \, L2 \, E\|g i t\|2 , \quad (2.88)$$

bởi vì ∇f (θt) = Eit [git].

Có hai thuật ngữ đều tích cực nhưng có dấu hiệu đối lập. Thuật ngữ đầu tiên (a) là tin tốt. Nó nói rằng theo kỳ vọng, chúng tôi sẽ đạt được một tiến bộ tích cực, tức là giảm giá trị kỳ vọng sau một bước gradient ngẫu nhiên. Vì số hạng này được nhân với αt, chúng ta có thể bị cám dỗ chỉ đơn giản là đặt αt thành một giá trị lớn để cải thiện kỳ vọng. Thật không may, điều này không xảy ra vì nhiệm kỳ thứ hai (b).

Mặc dù gradient ngẫu nhiên là một ước tính không thiên vị của gradient đầy đủ, nhưng nó vẫn là một ước tính nhiễu. Thuật ngữ thứ hai (b) phản ánh tiếng ồn này. Hãy tưởng tượng chúng ta gần với / a tối thiểu của f sao cho ∇f (θt) = 0. Thuật ngữ thứ hai (b) sau đó là dấu vết của hiệp phương sai của gradient ngẫu nhiên. Bởi vì nó không bằng không (tức là nhiễu), độ dốc ngẫu nhiên sẽ không làm giảm chức năng khách quan về kỳ vọng nhưng có thể làm tăng nó.

Để kiểm soát số hạng thứ hai (b), chúng ta phải đảm bảo rằng αt đủ nhỏ để αt ≫ α2t , hoặc phải giả định thêm các ràng buộc về f . Nếu chúng ta giảm αt trên t , độ dốc ngẫu nhiên sẽ tiến bộ theo kỳ vọng (tức là đi xuống) và cuối cùng vượt qua / a tối thiểu f . Thông tin chi tiết về

(Các) tốc độ hội tụ của độ dốc ngẫu nhiên nằm ngoài phạm vi của khóa học này.

Tóm lại, chúng tôi sử dụng độ dốc ngẫu nhiên trong học máy hiện đại và với tốc độ học tập nhỏ, độ dốc ngẫu nhiên thể hiện thuộc tính hạ thấp theo kỳ vọng. Do đó, chúng tôi sẽ bớt lo lắng hơn và dựa vào sự xuống dốc ngẫu nhiên trong suốt khóa học.

2.3.3Phương pháp tỷ lệ học tập thích ứng

Mặc dù chúng tôi đã tiếp cận vấn đề tối ưu hóa ngẫu nhiên bằng cách tuyên bố rằng chúng tôi tuân theo ước tính độ dốc ngẫu nhiên (âm) tại mỗi lần cập nhật, nhưng đó không nhất thiết phải là cách duy nhất để xem xét vấn đề này. Thay vào đó, chúng ta có thể xem vấn đề học tập là tối ưu hóa trực tuyến. Trong tối ưu hóa trực tuyến, hoặc học trực tuyến, chúng ta chơi một trò chơi trong đó tại mỗi lượt t, chúng ta nhận được ước tính stochasticgradient gt = ∇θfit(θt−1) và sử dụng nó để cập nhật ước tính của chúng ta về các tham số, θt−1, gt → θt. Chúng ta nhận được hình phạt là sự khác biệt giữa ước tính ngẫu nhiên của tổn thất tại tham số cập nhật và ước tính tại cấu hình tham số tối ưu,6 tức là phù hợp (θt) − phù hợp (θ∗). Chúng tôi gọi hình phạt này là hối tiếc, vì điều này định lượng chúng tôi có thể làm tốt hơn bao nhiêu khi nhận thức muộn (nghĩa là hối tiếc). Mục tiêu là để giảm thiểu sự hối tiếc theo thời gian:

$$R(T) = \sum_{t=1}^{T} \left[\text{phù hợp}(\theta_t) - \text{phù hợp}(\theta_*) \right]. \tag{2.89}$$

Sự hối tiếc phải phát triển theo tuyến tính, tức là R (T) = o (T), vì tăng trưởng tuyến tính, tức là R (T) = O (T), ngụ ý rằng thuật toán học tập không hội tụ về phía giải pháp tối ưu (hoặc giá trị tối thiểu liên quan của nó.)

Chúng tôi (cố gắng) đạt được mục tiêu này bằng cách tìm ra một quy tắc cập nhật thích hợp ánh xạ θt−1 và gt thành θt. Khi làm như vậy, tương đối đơn giản để nghĩ đến khung đơn giản sau đây, khái quát hóa sự xuống dốc ngẫu nhiên:

$$\theta_t \leftarrow \theta_{t-1} + \eta_t \odot g_t, \tag{2.90}$$

trong đó ηt là một tập hợp các tốc độ học tập cho tất cả các tham số.7 Bằng cách điều chỉnh ηt một cách thích hợp, chúng ta có thể đạt được sự hối tiếc cận tuyến. Trong SGD ở trên, ηt thường là phi hướng, tức là ηit = ηjt cho tất cả i /= j. SGD trên thực tế đạt được sự hối tiếc cận tuyến, O(√T) với ηt =1√T, nhưng hóa ra chúng ta có thể làm tốt hơn hoặc tiệm cận hoặc

6Tính tối ưu trong bối cảnh thích ứng trực tuyến này được định nghĩa là giải pháp cuối cùng đạt được bởi quy trình tối ưu hóa trực tuyến. Nếu chúng ta đi theo hướng tương quan vớigradient, chúng ta biết rằng chúng ta đang tiến bộ trung bình hướng tới cấu hình cực đoan cục bộ do lemma tốt ở trên. Do đó, chúng ta biết rằng về mặt tiệm cận giải pháp tối ưu ở đây θ∗ sẽ có tổn thất thấp hơn bất kỳ điểm trung gian nào khác. Điều này làm cho góc độ học trực tuyến khác với góc độ tối ưu hóa tử trên.

7Có thể sử dụng ma trận ηt thay vì vectơ ηt, và có thể có một cơ hội tốt là chúng ta sẽ đạt được một giới hạn hối tiếc tốt hơn. Thật không may, điều này có thể làm tăng đáng kể độ phức tạp tính toán cho mỗi lần cập nhật, từ O(|θ|) đến O(|θ|2), có thể bị cấm trong nhiều ứng dụng hiện đại.

thực tế bằng cách tính đến bối cảnh hàm thua lỗ, tức là cách mất mát thay đổi wrt các tham số, cẩn thận hơn.

Adagrad [Duchi và cộng sự, 2011]. Đối với mỗi tham số θi, độ lớn của Đạo hàm một phần của tổn thất, (git)2, cho chúng ta biết giá trị tổn thất nhạy cảm như thế nào đối với sự thay đổi trong θi. Hoặc, một cách khác là xem nó như tác động của sự thay đổi inθi đối với tổn thất. Bằng cách tích lũy điều này theo thời gian, qPtt′=1(git′)2, chúng ta có thể đo lường tác động tổng thể của θi đối với tổn thất. Sau đó, chúng ta có thể chuẩn hóa mỗi bản cập nhật theo tỷ lệ nghịch đảo để đảm bảo mỗi và mọi tham số có tác động ít nhiều như nhau đối với tổn thất. Đó là

$$\theta t \leftarrow \theta t-1 + \frac{\square \sqrt{\text{Ptt}'=1(\text{g1t}'} \square}{)2... \square\square\square\square \text{GT}}{1\text{qPtt}'=1(\text{g}|\theta|\text{t}')2} \tag{2.91}$$

Sự hối tiếc của Adagrad là O(√T), giống như SGD, giả sử ‖gt‖ ≪ ∞.Tuy nhiên, nó thường giảm nhanh hơn, đặc biệt là khi nhiều tham số không có chỉ số (thưa thớt) và/hoặc học nhanh (bởi vì độ lớn tích lũy tăng nhanh và nghịch đảo của nó hội tụ về 0 nhanh chóng.)

Adam [Kingma và Ba, 2014]. Một nhược điểm lớn của Adagrad ở trên là tốc độ học trên mỗi tham số suy giảm một cách đơn điệu, thường dẫn đến việc chấm dứt sớm. Điều này đặc biệt có vấn đề với một vấn đề tối ưu hóa không lồi, chẳng hạn như những vấn đề trong đào tạo mạng nơ-ron sâu, vì nó có thể yêu cầu nhiều cập nhật để tối ưu hóa đủ gần với một giải pháp tốt trong không gian tham số. Chúng ta có thể giải quyết vấn đề này bằng cách không tích lũy độ lớn của gradient trong toàn bộ thời lượng mà sử dụng làm mịn theo cấp số nhân:

$$\text{vt} \leftarrow \beta\text{vvt}-1 + (1 - \beta\text{v})\text{g2t}\ , \tag{2.92}$$

trong đó βv ∈ [0, 1]. Sau đó, chúng ta sử dụng √vt làm tốc độ học tập thay thế, dẫn đến

$$\theta\text{it} \leftarrow \theta\text{it}-1 + \frac{\text{gitpvit} +}{\varepsilon\ ,} \tag{2.93}$$

trong đó ε > 0 là một vô hướng nhỏ để ngăn chặn trường hợp thoái hóa.
Adam cũng sử dụng làm mịn theo cấp số nhân để giảm phương sai của ước tính gradient:

$$\text{MT} \leftarrow \beta\text{mmt}-1 + (1 - \beta\text{m})\text{gt}. \tag{2.94}$$

Điều này dẫn đến quy tắc cập nhật cuối cùng sau:

$$\theta\text{it} \leftarrow \theta\text{it}-1 + \frac{\text{mitpvit} +}{\varepsilon\ ,} \tag{2.95}$$

trong đó α ∈ (0, 1] là kích thước bước mặc định. Adam cũng có sự hối tiếc của O(√T) và thể hiện một hành vi tiệm cận tổng thể tương tự như Adagrad. Tuy nhiên, Adam thường được ưa chuộng hơn Adagrad, bởi vì tỷ lệ học trên mỗi thông số không còn giảm một cách đơn điệu nữa. Kể từ khi nó được đề xuất trước đó, đã có một số cải tiến đối với Adam, mặc dù chúng nằm ngoài phạm vi của khóa học này.

Nhìn chung, bất cứ khi nào chúng tôi đề cập đến độ dốc ngẫu nhiên trong phần còn lại của khóa học, chúng tôi thường đề cập đến Adam hoặc các biến thể của nó cập nhật tốc độ học của từng tham số một cách thích ứng. Mặc dù đó chỉ là một sự khôn ngoan dân gian, khá nhiều nhà nghiên cứu, bao gồm cả tôi, cho rằng những thành công đáng kinh ngạc được quan sát thấy gần đây của nhiều thuật toán học máy thông thường với tối ưu hóa gradient cho các thuật toán tốc độ học thích ứng này.

2.4 Tổng quát hóa và lựa chọn mô hình

2.4.1Rủi ro dự kiến so với rủi ro thực nghiệm: giới hạn khái quát hóa

Rủi ro là một từ khác mà chúng tôi sử dụng để chỉ thua lỗ. Trong phần này, chúng tôi sẽ sử dụng rủi rothay vì thua lỗ, vì cái trước thường được sử dụng trong bối cảnh cụ thể này. Nếu bạn bối rối với thuật ngữ "rủi ro", chỉ cần đọc to nó là "thua lỗ" bất cứ khi nào bạn gặp phải nó.

Đối với mỗi ví dụ (x, y), bây giờ chúng ta biết cách xây dựng một hàm năng lượng và cũng là một hàm tổn thất liên kết L([x, y], θ). Giả sử pdata(x, y) là một số phân phối chưa biết mà từ đó chúng ta rút ra một ví dụ (x, y). Chúng ta không biết phân phối này là gì, nhưng chúng ta giả định rằng đây là sự phân phối mà từ đó các ví dụ đào tạo đã được rút ra và bất kỳ trường hợp nào trong tương lai cũng sẽ được rút ra.8 Sau đó, mục tiêu của chúng ta phải là giảm thiểu

$$R(\theta) = E_{data}\left[L([x, y], \theta)\right] . \tag{2.96}$$

Thật không may, rủi ro dự kiến này không thể tính toán được và chúng ta chỉ có quyền truy cập vào một proxy dựa trên mẫu cho rủi ro dự kiến, được gọi là rủi ro thực nghiệm:

$$\hat{R}(\theta) = \frac{1}{N} \sum_{n=1}^{N} L([x_n, trong], \theta). \tag{2.97}$$

Để ngắn gọn và rõ ràng, hãy cho Sn = Pnk=1 L([xk, yk], θ). Sau đó, chúng ta có thể diễn đạt những rủi ro này như

$$R(\theta) = \underset{\text{Dữ liệu}}{\text{Dữ liệu}\times\cdots\times\text{dữ liệu}} \frac{1}{N} S_N \square, \text{ và } \hat{R}(\theta) = \frac{1}{N} S_N . \tag{2.98}$$

Cái trước có hiệu lực vì mỗi trường hợp (x, y) được vẽ độc lập từ cùng một phân phối dữ liệu.

8Điều này chắc chắn không đúng trong thực tế, nhưng là một điểm khởi đầu hợp lý. Chúng ta sẽ thảo luận về phần sau của khóa học những gì chúng ta có thể làm nếu giả định này không đúng, hy vọng nếu thời gian cho phép.

Giả sử rằng một trận thua riêng lẻ được giới hạn trong khoảng từ 0 đến 1 (trường hợp này sẽ là trường hợp của trận thua 0-1.) Sau đó, chúng ta có thể sử dụng bất đẳng thức của Hoeffding để có được

$$p(|R(\theta) - \hat{R}(\theta)| \geq (e) \leq 2 \exp - \frac{2 (N \varepsilon) 2PNn}{= 1 (1 - 0) 2}! = 2 \text{ kinh nghiệm } -2N \varepsilon 2 . \quad (2.99)$$

Sự bất bình đẳng này cho chúng ta biết rằng khoảng cách giữa rủi ro dự kiến và rủi ro thực nghiệm thu hẹp theo cấp số nhân với N, số lượng ví dụ đào tạo mà chúng ta sử dụng để tính toán rủi ro thực nghiệm. Bất đẳng thức này áp dụng cho bất kỳ θ nào, ngụ ý rằng sự hội tụ của rủi ro thực nghiệm đối với rủi ro dự kiến là đồng nhất trên không gian tham số (hoặc không gian phân loại tương ứng.) Sự hội thảo đồng nhất như vậy rất tốt ở chỗ chúng ta không phải lo lắng về việc học tập hoạt động tốt như thế nào (nghĩa là loại giải pháp mà chúng ta kết thúc sau khi tối ưu hóa), để xác định mức độ lệch mà chúng ta sẽ dự đoán giữa rủi ro thực nghiệm (rủi ro chúng ta có thể tính toán) và rủi ro dự kiến ở bất kỳ θ nào. Mặt khác, có một câu hỏi lớn là liệu chúng ta có thực sự quan tâm đến hầu hết không gian tham số hay không; Có khả năng chúng tôi không làm vậy và chúng tôi chỉ quan tâm đến một tập hợp con nhỏ của không gian tham số mà tối ưu hóa lặp lại, chẳng hạn như độ dốc ngẫu nhiên, khám phá. Chúng ta sẽ thảo luận thêm một chút về điều này sau, nhưng bây giờ, hãy giả sử rằng chúng ta hài lòng với sự hội tụ đồng nhất này.9

Hãy tưởng tượng rằng ai đó (hoặc một thuật toán học tập nào đó) đã cho tôi θ được cho là tốt với một rủi ro thực nghiệm cụ thể $\hat{R}(\theta)$. Có cách nào để tôi kiểm tra xem rủi ro dự kiến R(θ) sẽ tồi tệ hơn bao nhiêu, dựa trên sự bất đẳng thức của Hoeffding ở trên không? Tất nhiên, một tuyên bố như vậy sẽ phải là xác suất, vì chúng ta đang làm việc với các biến ngẫu nhiên, R(θ) và $\hat{R}(\theta)$.

Bất bình đẳng ở trên cho phép chúng ta thể hiện điều đó

$$|R(\theta) - \hat{R}(\theta)| < e \quad (2.100)$$

với một số xác suất ít nhất là 1−δ. Hãy lưu ý rằng hướng đi của bất bình đẳng đã đảo ngược.

Nếu $|R(\theta) - \hat{R}(\theta)| < \varepsilon$, chúng ta biết rằng $R(\theta) < \hat{R}(\theta) + \varepsilon$. Chúng tôi quan tâm đến bất đẳng thức thứ hai này, bởi vì chúng tôi muốn giới hạn cao hơn rủi ro (thực sự) dự kiến. Nếu rủi ro thực sự thấp hơn rủi ro thực nghiệm, chúng ta hạnh phúc và không quan tâm đến nó. Chúng ta muốn biết nếu chúng ta không hạnh phúc (nghĩa là, rủi ro dự kiến lớn hơn rủi ro thực nghiệm), chúng ta sẽ không hạnh phúc như thế nào trong trường hợp xấu nhất.

Bởi vì chúng ta muốn đưa ra một tuyên bố như vậy với xác suất ít nhất là

9Trong thực tế, chúng ta không phải vậy.

1 − δ, chúng ta đánh đồng phía bên phải ở trên với δ:

$$2 \text{ kinh nghiệm } (-2N\,\varepsilon 2) = \delta \tag{2.101}$$

$$\Leftrightarrow\Rightarrow -2N\,\varepsilon 2 = \log \delta 2 \tag{2.102}$$

$$\Leftrightarrow\Rightarrow e2 = \frac{1}{2N}\log \frac{2}{d} \tag{2.103}$$

$$\Leftrightarrow\Rightarrow \epsilon = \sqrt{\frac{1}{2\delta}\log \frac{2}{.}} \tag{2.104}$$

Kết hợp hai điều này với nhau, bây giờ chúng ta có thể nói rằng với xác suất ít nhất là 1 − δ, chúng ta có

$$R(\theta) < \hat{R}(\theta) + \sqrt{\frac{1}{2\delta}\log \frac{2}{.}} \tag{2.105}$$

cho tham số mô hình θ.

Giới hạn khái quát hóa này có ý nghĩa. Nếu chúng ta muốn có được một sự đảm bảo mạnh mẽ, tức là (1 − δ) → 1 (tương đương δ → 0), chúng ta sẽ có một giới hạn thua cuộc hơn nhiều, vì giới hạn là O(qlog−1δ). Chúng ta có thể chống lại điều này bằng cách thu thập nhiều ví dụ đào tạo hơn, tức là N → ∞, vì giới hạn co lại nhanh chóng khi N phát triển: O(N − 12).

Giới hạn này có vẻ hợp lý, nhưng có một điểm mấu chốt. Điểm mấu chốt là điều này dựa trên một mô hình θ nhất định. Nói cách khác, ràng buộc này quá lạc quan, vì trong thực tế, chúng ta thường cần phải chọn θ bản thân trong số nhiều lựa chọn thay thế bằng quá trình học tập. Khi làm như vậy, chúng ta cần xem xét khả năng bằng cách nào đó chúng ta đã chọn một trong những người có khoảng cách khái quát hóa tồi tệ nhất |R(θ) − ˆR(θ)|. Nói cách khác, chúng ta cần xem xét giới hạn khái quát hóa của tất cả các tham số mô hình có thể có.

Để đơn giản, chúng tôi giả định rằng θ ∈ Θ trong đó Θ là một tập hợp hữu hạn có kích thước K.Học sau đó là một quá trình chọn một trong K cấu hình tham số có thể dựa trên dữ liệu. Chúng tôi sử dụng ý tưởng về cái gọi là liên kết ràng buộc từ lý thuyết xác suất cơ bản, trong đó nói rằng

$$p(e1 \cup e2 \cup \cdots \cup eN) \le \sum_{i=1}^{N} p(\text{không}). \tag{2.106}$$

Điều này hơi rõ ràng, bởi vì một cặp (ei, ej) có thể không loại trừ lẫn nhau. Hãy nghĩ về một sơ đồ Venn. Với điều này, chúng ta muốn tính toán

$$p(\cup_{\in\Theta}|R(\theta) - \hat{R}(\theta)| \ge (e) \le \sum_{i\in T} p(|R(\theta) - \hat{R}(\theta)| \ge e) \le 2||\text{ kinh nghiệm } -2N\,\varepsilon 2 \cdot |\{z\}=2\sqrt{\log|\Theta|}\exp(-2N\,\varepsilon 2) \tag{2.107}$$

Chúng ta có thể làm theo logic chính xác ở trên:

$$2\exp(\log|\Theta| - 2N\,\varepsilon 2) = \delta \tag{2.108}$$

$$\Leftrightarrow\Rightarrow \epsilon = \frac{\sqrt{\log|\Theta| - \text{nhật ký } 2d}}{2 \text{ lần}}. \tag{2.109}$$

Điều này có ý nghĩa, vì giới hạn khái quát hóa bây giờ phụ thuộc vào kích thước của Θ, không gian giả thuyết của chúng ta. Nếu tập hợp giả thuyết lớn, có nhiều khả năng chúng ta tìm ra một giải pháp tốt về mặt kinh nghiệm $\hat{R}(\theta) \downarrow$ nhưng dựa trên kỳ vọng $R(\theta) \uparrow$ rất tệ. Điều này cũng ngụ ý rằng chúng ta cần N (số lượng ví dụ đào tạo) để tăng theo cấp số nhân so với kích thước của không gian giả thuyết Θ.

Giới hạn này chỉ hoạt động với một tập giả thuyết kích thước hữu hạn Θ mà không ưu tiên bất kỳ cấu hình tham số cụ thể nào. Để làm việc với một tập hợp giả thuyết lớn vô hạn, chúng ta phải đưa ra các cách tiếp cận khác nhau. Ví dụ, chiều Vapnik-Chervonenkis (VC) có thể được sử dụng để giới hạn độ phức tạp của tập hợp giả thuyết lớn vô hạn [Vapnik và Chervonenkis, 1971]. Hoặc, chúng ta có thể sử dụng giới hạn PAC-Bayes, trong đó một phân phối trước trên tập hợp giả thuyết (có khả năng lớn vô hạn) được giới thiệu [McAllester, 1999]. Đây là tất cả những điều nằm ngoài phạm vi của khóa học này, nhưng chúng tôi đề cập ngắn gọn đến ý tưởng về PAC-Bayesbound ở đây trước khi kết thúc phần này.

Giới hạn PAC-Bayesian. Kết quả ban đầu của PAC-Bayes nói rằng

$$D_{KL}(B(\hat{R}(Q)) \| B(R(Q))) \leq \frac{1}{N} D_{KL}(Q \| P) + \frac{\log N}{} \square + 1\delta \tag{2.110}$$

với xác suất ít nhất là 1 − δ. Mặc dù sự bất bình đẳng này trông khá dày đặc, nhưng những thuật ngữ này cực kỳ mô tả, một khi chúng ta định nghĩa và học cách đọc chúng.

Đầu tiên, R(Q) và $\hat{R}(Q)$ được định nghĩa tương tự như R(θ) và $\hat{R}(\theta)$, ngoại trừ việc chúng ta gạt ra ngoài θ bằng cách sử dụng cái gọi là phân phối hậu Q(θ). Đó là

$$R(Q) = E_Q[R(\theta)] \tag{2.111}$$

$$\hat{R}(Q) = E_Q\langle \hat{R}(\theta) \rangle . \tag{2.112}$$

Q có thể là bất kỳ phân phối nào và có thể phụ thuộc vào dữ liệu D bao gồm N ví dụ. Bởi vì chúng ta tiếp tục giả định rằng chúng ta làm việc với một tổn thất có giới hạn, chúng ta có thể giả định rằng $R(Q) \in [0, 1]$ và $\hat{R}(Q) \in [0, 1]$. Sau đó, chúng ta có thể xác định phân phối Bernoullisử dụng hai giá trị này làm phương tiện. Chúng tôi biểu thị các phân phối này là $B(R(Q))$ và $B(\hat{R}(Q))$ tương ứng. Bạn có thể nghĩ về các phân phối này như các rủi ro dự kiến và thực nghiệm thay đổi như thế nào khi θ theo phân phối Q. Sau đó, chúng ta có thể đo lường sự khác biệt giữa hai đại lượng này, theo định nghĩa là khoảng cách tổng quát, bằng cách sử dụng phân kỳ KL. Đây là phía bên trái của bất bình đẳng ở trên.

Phía bên phải sau đó là giới hạn về mức độ khác biệt giữa rủi ro thực nghiệm và rủi ro dự kiến có thể có trung bình với Q. Có hai thuật ngữ ở đây. Thuật ngữ đầu tiên là phân kỳ KL giữa Q sau và cái gọi là P trước, trong đó P bị hạn chế để độc lập với dataD. Bạn có thể coi P là niềm tin trước của chúng ta về tham số θ nào sẽ tốt. Mặt khác, Q là niềm tin của chúng ta sau khi quan sát dữ liệu D. Do đó, thuật ngữ đầu tiên nói rằng sự khác biệt sẽ lớn hơn nếu niềm tin trước đó của chúng ta là không đúng, nghĩa là niềm tin của chúng ta sau khi quan sát dữ liệu thay đổi đáng kể từ

niềm tin trước đó. Tuy nhiên, hiệu ứng này sẽ biến mất nhanh chóng khi số lượng ví dụ đào tạo tăng lên do 1N .
Chúng ta có thể đọc hai điều từ số hạng thứ hai 1N log N + 1δ . Bởi vì δ nằm trong mẫu số, chúng ta biết rằng chúng ta có khả năng nhận được sự khác biệt lớn hơn nếu chúng ta muốn có được một sự đảm bảo mạnh mẽ hơn, tức là δ → 0. log (N + 1) Nbiến mất về phía 0 khi kích thước dữ liệu tăng lên, tức là N → ∞. Tuy nhiên, tốc độ hội tụ này khá chậm, tức là cận tuyến.

Tương tự như những gì chúng ta đã làm trước đó, chúng ta có thể biến bất đẳng thức này trong Phương trình (2.110) thành một giới hạn khái quát hóa. Đặc biệt, chúng tôi sử dụng bất đẳng thức của Pinsker. Trong trường hợp của chúng ta với các biến ngẫu nhiên Bernoulli, chúng ta nhận được

$$\hat{R}(Q) - R(Q)2 \le 12 DKL(B(\hat{R}(Q)))\|B(R(Q)))). \tag{2.113}$$

Sau đó

$$| \hat{R}(Q) - R(Q)| \le \sqrt{\frac{1}{2 \text{ lần}} \square DKL (Q \| P) + \log \frac{N + \square}{1\delta}}. \tag{2.114}$$

Chúng ta kết thúc với giới hạn khái quát hóa sau:

$$R (Q) \le \hat{R} (Q) + \sqrt{\frac{1}{2 \text{ lần}} \square DKL (Q \| P) + \log \frac{N + \square}{1\delta}}. \tag{2.115}$$

Không giống như giới hạn khái quát hóa trước đó và các biến thể của nó, PAC-Bayesianbound này cung cấp cho chúng ta những hiểu biết hữu ích hơn. Đầu tiên, chúng tôi muốn phân phối sau Q tốt ở chỗ nó dẫn đến rủi ro thực nghiệm trung bình thấp hơn. Nghe có vẻ hiển nhiên, nhưng giới hạn khái quát hóa trước đó được thiết kế để hoạt động với bất kỳ cấu hình tham số nào (hội tụ đồng nhất) và không cho chúng ta biết ý nghĩa của việc chọn một cấu hình tham số tốt. Với PAC-Bayesianbound, chúng ta đã biết rằng chúng ta muốn chọn cấu hình tham số sao cho rủi ro thực nghiệm trung bình thấp. Nói cách khác, chúng ta nên sử dụng một thuật toán học tốt.

Tuy nhiên, phân phối sau Q không thể quá xa so với nơi chúng ta bắt đầu. Vì giới hạn là một hàm của sự khác biệt giữa Q và niềm tin trước đó của chúng ta P . Lật đồng xu xung quanh, nó cũng nói rằng chúng ta phải chọn P trước đó của chúng ta để nó đặt xác suất cao trên các cấu hình tham số có khả năng xảy ra theo phân phối hậu Q. Nói cách khác, chúng ta muốn đảm bảo rằng chúng ta cần một lượng công việc tối thiểu để đi từ P đến Q, để giảm thiểu giới hạn khái quát.

Tóm lại, giới hạn PAC-Bayes cho chúng ta biết rằng chúng ta nên có một số kiến thức trước về vấn đề và chúng ta không nên đào tạo một mô hình dự đoán quá nhiều, do đó đảm bảo rằng phân phối sau Q ở gần với phân phối trước P . Điều này sẽ đảm bảo rằng rủi ro dự kiến không đi quá nhiều so với rủi ro thực nghiệm.

2.4.2Thiên vị, Phương sai và Không chắc chắn

Một cách khác để viết tổn thất 0-1 là dựa vào chênh lệch bình phương giữa nhãn thực và nhãn dự đoán, trong trường hợp phân loại nhị phân, trong đó chỉ có hai loại mà đầu vào có thể thuộc về. Chúng ta hãy sử dụng ∈ {−1, 1} để biểu thị hai lớp. Sau đó

$$L([x, y], \theta) = \frac{1}{4}(y - \hat{y}(x, \theta))2, \tag{2.116}$$

trong đó $\hat{y}(x, \theta) = \arg\min_{c\in\{-1,1\}} e([x, c], \theta)$. Nếu y và $\hat{y}$ giống nhau, tổn thất này bằng không. Nếu không, nó là

Như chúng ta đã thảo luận trước đó, một trường hợp (x, y) được rút ra từ một phân phối dữ liệu cơ bản pdata(x, y) có thể được ghi lại dưới dạng

$$\text{pdata}(x, y) = \text{pdata}(x)\text{pdata}(y|x) \tag{2.117}$$

theo định nghĩa về xác suất có điều kiện.

Hơn nữa, chúng ta cũng có thể tưởng tượng một phân phối trên θ: q(θ). Sự phân phối này có thể xuất hiện từ hư không. Tuy nhiên, việc có một phân phối trên θ thay vì một giá trị duy nhất của θ là điều tự nhiên nếu chúng ta nhận ra rằng việc học luôn phụ thuộc vào một số tính ngẫu nhiên, do tối ưu hóa phá vỡ đối xứng tùy ý, lấy mẫu ngẫu nhiên các ví dụ đào tạo hoặc đôi khi thiếu khả năng kỹ thuật trong việc giảm nhiễu trong hệ thống của chúng ta. Chúng ta sẽ thảo luận sâu về sự không chắc chắn này trong các tham số mô hình sau, và bây giờ, chúng ta giả định rằng q (θ) này được đưa ra cho chúng ta.

Sau đó, chúng ta có thể viết ra tổn thất 0-1 dự kiến cho phân loại nhị phân dưới phân phối dữ liệu này (không xác định) trên các tham số mô hình như sau:

$$\frac{1}{4}E_{x,y,\theta}(y - \hat{y}(x, \theta))2 \propto E_x E_{y|x}(y - \mu y)2 + \tag{2.118}$$

$$\mu 2y \tag{2.119}$$

$$+ E_\theta(\hat{y}(x, \theta) - \hat{\mu}y)2 + \hat{\mu}2y \tag{2.120}$$

$$-2E_{y|x}E_\theta[(y - \mu y)(\hat{y}(x, \theta) - \hat{\mu}y)] - 2\mu y\,\hat{\mu}y$$

$$= \text{Ví dụ} \quad \underbrace{E_{y|x}(y - \mu y)2}_{=\{z\}} + \underbrace{E_\theta(\hat{y}(x, \theta) - \hat{\mu}y)2}_{=\{z\}} \tag{2.121}$$

$$-2\,\underbrace{E_{y|x}E_\theta[(y - \mu y)(\hat{y}(x, \theta) - \hat{\mu}y)]}_{=0}\Big|_{\{z\}} + \underbrace{(\mu y - \hat{\mu}y)2}_{=\{z\}}, \tag{2.122}$$

đâu

$$\mu y = E_{y|x}[y], \tag{2.123}$$

$$\hat{\mu}y = E_\theta[\hat{y}(x, \theta)]. \tag{2.124}$$

Có rất nhiều thuật ngữ chúng ta cần xem xét trong phương trình này, nhưng chúng ta sẽ xem xét chúng từng thuật ngữ một, từ phía sau. Đầu tiên, chúng ta hãy bắt đầu với (µy − ˆµy)2.Thuật ngữ (c) này cho chúng ta biết người học của chúng ta nắm bắt được giá trị trung bình của đầu ra thực y tốt như thế nào. Thuật ngữ này không quan tâm đến việc có bao nhiêu phương sai dưới phân phối dữ liệu pdata(y|x) cũng như dưới phân phối mô hình q(θ). Nó chỉ nói về việc có được kết quả chính xác trung bình. Thuật ngữ này được gọi là thiên vị. Khi thuật ngữ (c) này bằng không, chúng ta gọi công cụ dự đoán của chúng ta là không thiên vị.

Số hạng thứ hai từ phía sau, bằng không, là hiệp phương sai (âm) giữa kết quả thực y và kết quả dự đoán ˆy(x, θ), cả hai đều là biến ngẫu nhiên. Bởi vì chúng ta không giả định bất cứ điều gì về q(θ), nói chung chúng ta không thể giả định θ có tương quan với y|x, ngụ ý rằng không nên có bất kỳ hiệp phương sai nào. Chúng ta có thể bỏ qua thuật ngữ này.

Chúng ta hãy tiếp tục với hai thuật ngữ còn lại, (a) và (b). Thuật ngữ đầu tiên (a) là phương sai của kết quả thực sự y. Điều này phản ánh sự không chắc chắn vốn có trong kết quả thực sự cho một đầu vào x. Sự không chắc chắn vốn có này không thể giảm bớt, vì nó không phải là những gì chúng ta kiểm soát mà được trao cho chúng ta bởi bản chất của vấn đề mà chúng ta đang giải quyết. Nếu số lượng này lớn, chúng ta chỉ có thể làm được rất nhiều. Chúng ta thường gọi điều này là sự không chắc chắn hoặc sự không chắc chắn không thể rút gọn.

Thuật ngữ thứ hai (b) cũng là sự không chắc chắn, vì nó đo lường phương sai phát sinh từ sự không chắc chắn trong các tham số mô hình. Tuy nhiên, sự không chắc chắn này có thể điều khiển được và do đó có thể giảm bớt bằng những nỗ lực, vì nó phát sinh từ sự không chắc chắn q(θ) của chúng ta trong việc chọn các tham số θ. Khi mô hình đơn giản hơn, chúng ta có xu hướng nắm bắt tốt hơn trong việc học và có thể giảm rất nhiều sự không chắc chắn có thể rút gọn (hoặc nhận thức) này. Khi mô hình phức tạp và do đó thể hiện nhiều đối xứng phải bị phá vỡ một cách tùy tiện, rất khó (nếu không muốn nói là không thể) để giảm bớt nhiều sự không chắc chắn của biểu tượng này. Thuật ngữ này thường được gọi là phương sai.

Cần khá rõ ràng tại thời điểm này rằng phải có một số sự đánh đổi cố hữu giữa sai lệch (c) và phương sai (b). Một bộ phân loại càng phức tạp thì phương sai chúng ta càng cao, nhưng do độ phức tạp của nó, nó sẽ có thể khớp dữ liệu tốt, dẫn đến độ lệch thấp hơn. Khi một bộ phân loại đơn giản, phương sai sẽ thấp hơn, nhưng độ lệch sẽ cao hơn. Do đó, việc học có thể được coi là tìm ra sự cân bằng tốt giữa hai đại lượng cạnh tranh này.

Lời giải thích ở trên hơi khác so với cách thông thường trong đó mô tả sự đánh đổi giữa thiên vị-phương sai [Người đóng góp Wikipedia, 2023]. Đặc biệt, chúng tôi đang xem xét một phân phối chung q (θ) có thể liên quan trực tiếp hoặc không liên quan trực tiếp đến bất kỳ bộ dữ liệu đào tạo cụ thể nào, khi cách tiếp cận thông thường thường gắn bó với sự phụ thuộc mạnh mẽ vào tập dữ liệu đào tạo và phân phối trên tập huấn luyện. Đây là một sự khác biệt nhỏ, nhưng điều này có thể hữu ích khi chúng ta bắt đầu suy nghĩ về những cách kỳ lạ hơn mà chúng ta sử dụng q(θ). Nếu thời gian và không gian cho phép sau này trong khóa học, chúng ta có thể học một hoặc hai kỹ thuật liên quan đến các kỹ thuật kỳ lạ như vậy, chẳng hạn như học chuyển giao và học đa nhiệm.

10Tôi phải nhấn mạnh ở đây rằng độ phức tạp của một bộ phân loại không dễ định lượng. Khi tôi nói về 'phức tạp' hoặc 'đơn giản' ở đây, tôi đang đề cập đến thước đo huyền thoại này về độ phức tạp của bộ phân loại và không có nghĩa là chúng ta có thể tính toán nó một cách dễ dàng.

2.4.3Sự không chắc chắn về tỷ lệ lỗi

Trước tiên chúng ta cần nói về các biến ngẫu nhiên. Trong các khóa học xác suất mà bạn có thể đã tham gia trước đó, bạn phải học về sự phân biệt nghiêm ngặt giữa các biến ngẫu nhiên và biến không ngẫu nhiên. Trên thực tế, một biến ngẫu nhiên không lấy bất kỳ giá trị cụ thể nào mà mang theo một phân phối xác suất trên tất cả các giá trị có thể có mà nó có thể thực hiện. Một khi chúng ta rút ra một mẫu từ sự phân phối này, giá trị này không còn ngẫu nhiên nữa mà là xác định.

Thật không may, nó trở nên dễ dàng rườm rà để phân biệt rõ ràng các biến ngẫu nhiên giữa hai người và các mẫu được rút ra từ các phân phối của chúng. Đó là một trong những lý do tại sao chúng tôi đã không nêu rõ liệu bất kỳ biến cụ thể nào là ngẫu nhiên hay không. Một lý do khác, có lẽ quan trọng hơn, là hầu hết mọi biến trong học máy đều là ngẫu nhiên, bởi vì hầu hết mọi biến đều phụ thuộc vào một tập hợp các mẫu được rút ra từ một phân phối cơ bản chưa xác định. Ví dụ, các tham số θ là ngẫu nhiên, bởi vì chúng được khởi tạo bằng cách vẽ một mẫu từ cái gọi là phân phối trước, hoặc bởi vì chúng được cập nhật bằng cách sử dụng ước tính gradient ngẫu nhiên là một hàm của aset của các mẫu được rút ra từ phân phối dữ liệu. Từ quan điểm này, trên thực tế, dự đoán ˆy chúng tôi thực hiện bằng cách sử dụng một mô hình tham số hóa với θ cũng là một biến ngẫu nhiên. Do đó, tổn thất, hoặc rủi ro, là một biến ngẫu nhiên, như chúng ta đã thấy trong§2.4.1.

Khoảng tin cậy: nắm bắt sự thay đổi của bộ kiểm thử. Chúng ta hãy gắn bó với zero-one loss (mặc dù điều này không thực sự cần thiết, nhưng nó làm cho lập luận sau đây dễ theo dõi hơn.) Tổn thất l là một hàm của (1) một quan sát cụ thể [x, y] được rút ra từ phân phối dữ liệu pdata và (2) các tham số θ. Cả hai đều là nguồn của tính ngẫu nhiên, nhưng bây giờ, hãy giả sử rằng θ được cho chúng ta như một giá trị cố định, chứ không phải là một biến ngẫu nhiên với một phân phối gắn liền với nó. Nếu chúng ta giả sử có quyền truy cập vào N ví dụ kiểm tra, được rút ra độc lập từ pdata phân phối giống hệt nhau, chúng ta có

$$(l1, l2, \ldots, lN), \tag{2.125}$$

trong đó mỗi ln tự nó là một biến ngẫu nhiên. Mỗi và mọi biến N ran-dom này đều tuân theo cùng một phân phối. Bởi vì tất cả chúng đều tuân theo cùng một phân phối, chúng cũng chia sẻ giá trị trung bình và phương sai:

$$\mu = E[l] \text{ và } \sigma 2 = V[l] < \infty, \tag{2.126}$$

trong đó chúng ta giả định một cách an toàn rằng phương sai là hữu hạn.

Theo định lý giới hạn trung tâm, chúng ta sau đó biết rằng

$$\sqrt{N}(1N - \mu) \rightarrow d\ N(0, \sigma 2), \tag{2.127}$$

trong đó →d đề cập đến sự hội tụ trong phân phối, và

$$1N = \frac{1}{N} \sum_{n=1}^{N} Ln. \tag{2.128}$$

$\bar{l}_N$ là một biến ngẫu nhiên đề cập đến tổn thất trung bình được tính trên các Nexamples. Nói cách khác, với N lớn hơn, chúng ta mong đợi rằng độ chính xác trung bình mà chúng ta nhận được từ việc xem xét N ví dụ tập trung vào μ trung bình thực với phương sai $\frac{\sigma^2}{N}$. Vì vậy, càng nhiều N , chúng ta càng tin tưởng rằng mức trung bình mẫu không lệch quá nhiều so với mức trung bình thực. Tuy nhiên, với smallN, chúng ta không thể tự tin rằng độ chính xác trung bình mẫu của chúng ta đủ gần với mức trung bình thực và sự thiếu tin cậy này tỷ lệ thuận với phương sai thực cơ bản của độ chính xác. Thật không may, chúng ta không có quyền truy cập vào phương sai thực sự của độ chính xác nhưng thường có thể hiểu sơ bộ về nó bằng cách xem xét phương sai mẫu.

Nếu N lớn, chúng ta có thể tính khoảng tin cậy[11] và sử dụng nó để so sánh với một bộ phân loại khác hoặc kỳ vọng trước đó của bạn về độ chính xác. Ví dụ, bởi vì ước tính độ chính xác hội tụ đến phân phối chuẩn, chúng ta có thể sử dụng cái gọi là kiểm tra t, vì sự khác biệt giữa giá trị trung bình thực và giá trị trung bình của ước tính hội tụ về phân phối t của Học sinh. Trong trường hợp đó, khoảng tin cậy cho độ chính xác nhị phân (đơn giản là $1 - l^\star$, trong đó $l^\star$ là tổn thất thực của bộ phân loại) được cho bởi

$$\text{CHÚNG TÔI} \approx \left[(1 - \bar{l}_N) - Z \sqrt{\frac{\bar{l}_N (1 - \bar{l}_N)}{N}}, (1 - \bar{l}_N) + Z \sqrt{\frac{\bar{l}_N (1 - \bar{l}_N)}{N}} \right], \qquad (2.129)$$

trong đó Z được xác định dựa trên mức độ tin cậy mục tiêu γ. Nếu γ = 0,99, Z sẽ xấp xỉ 2,576.

Giả sử l_0 là độ chính xác của bộ phân loại hiện có. Chúng tôi sẽ giả định đây là đại lượng chính xác vì chúng tôi đã chạy bộ phân loại này trong một thời gian rất dài. Chúng ta có thể sử dụng khoảng tin cậy này để biết liệu chúng ta có muốn thay thế bộ phân loại hiện có bằng bộ phân loại mới này hay không. Nếu l_0 nằm thoải mái bên ngoài khoảng tin cậy này, chúng ta sẽ cảm thấy thoải mái hơn khi xem xét tùy chọn này.

Cách tiếp cận này tập trung vào việc ước tính tỷ lệ lỗi và xác nhận liên quan, cho một bộ phân loại θ. Nói cách khác, tính ngẫu nhiên mà chúng ta đang xem xét bắt nguồn từ việc lựa chọn tập thử nghiệm D. Nếu chúng tôi liên tục thu được các bộ kiểm tra mới và tính toán các khoảng tin cậy liên quan, chúng tôi dự đoán độ chính xác thực sẽ được đưa vào khoảng tin cậy khoảng γ lần. Tuy nhiên, điều này chỉ cho chúng ta biết một khía cạnh của câu chuyện. Chúng ta hãy xem xét hai khía cạnh bổ sung.

Khoảng thời gian đáng tin cậy: nắm bắt các biến thể mô hình. Có khá nhiều Các yếu tố làm cho thuật toán học tập của chúng ta ngẫu nhiên. Thứ nhất, hàm mục tiêu của chúng ta có xu hướng có nhiều cực nhỏ cục bộ, phát sinh từ các lý do như các đặc điểm đồng tuyến tính và bất biến tỷ lệ. Ví dụ, nếu chúng ta sử dụng tổn thất zero-one, các bộ phân loại sau đây đều tương đương:

$$\hat{y} = \arg\max_{y=1,\ldots,|y|} \alpha W^\top x + \text{bằng}, \quad \text{một} > 0, \qquad (2.130)$$

---

[11] Khoảng tin cậy cho một đại lượng có mức tin cậy γ có nghĩa là nếu chúng ta lặp lại quá trình suy luận đại lượng mục tiêu và đo khoảng tin cậy, đại lượng mục tiêu thực sự sẽ được đưa vào khoảng tin cậy tỷ lệ thuận với γ.

trong đó (·)j đề cập đến phần tử thứ j của vectơ, bởi vì tổn thất zero-one bất biến với tỷ lệ nhân của giá trị năng lượng. Một cặp tính năng đồng tuyến tính được xác định là có mối quan hệ tuyến tính với kết quả mục tiêu. Hãy tưởng tượng điều đó

$$x_j = \alpha x_i, \tag{2.131}$$

khi y = c. Sau đó, chúng ta nói rằng (xi, xj) là đồng tuyến tính cho y = c. Trong trường hợp này, hai hàm năng lượng sau đây là tương đương:

$$e([x, c], \theta) = -\begin{cases} w_{c,1}, \ldots, w_{c,i}, \ldots, w_{c,|x|} & \text{x – trước Công nguyên,} \\ 0 & \text{otherwise} \end{cases} \tag{2.132}$$

$$e'([x, c], \theta') = -\begin{cases} w_{c,1}, \ldots, \frac{1}{\alpha} w_{c,i}, \ldots, w_{c,|x|} & \text{x – trước Công nguyên,} \\ 0 & \text{otherwise} \end{cases} \tag{2.133}$$

cho bất kỳ α nào / = 0. Chúng ta không thể thực sự phân biệt hai hàm năng lượng này. Có nhiều hơn trong số này, mà chúng tôi sẽ đề cập trong phần còn lại của khóa học, và tất cả chúng đều dẫn đến vấn đề rằng người học của chúng tôi sẽ chọn một trong các giải pháp tương đương (hoặc gần như tương đương) một cách ngẫu nhiên. Tính ngẫu nhiên như vậy phát sinh từ nhiều yếu tố, bao gồm khởi tạo ngẫu nhiên, xây dựng ngẫu nhiên của các lô nhỏ trong độ dốc ngẫu nhiên và thậm chí không xác định trong việc triển khai các kiến trúc điện toán cơ bản. Nghĩa là, học tập không thực sự là một quá trình xác định mà là một quá trình ngẫu nhiên, dẫn đến một ˆθ ngẫu nhiên. Nói cách khác, mỗi khi chúng ta đào tạo một mô hình, chúng ta đang lấy mẫu ˆθ một cách hiệu quả từ phân phối vô điều kiện trên một biến ngẫu nhiên θ cho tập huấn luyện D, tức là, ˆθ ∼ p(θ|D). Phân phối này thường được gọi là phân phối hậu và nếu thời gian cho phép, chúng ta sẽ tìm hiểu về phân phối này cẩn thận hơn trong bối cảnh học máy Bayes sau.

Chúng tôi coi1N, độ chính xác của bộ thử nghiệm, trong Phương trình (2.128) là một biến ngẫu nhiên có tính ngẫu nhiên phát sinh từ việc lựa chọn tập thử nghiệm. Tuy nhiên, ở đây chúng tôi coi nó là một biến ngẫu nhiên có tính ngẫu nhiên được tạo ra bởi sự lựa chọn các tham số θ chứ không phải tập thử nghiệm D′. Điều này có thể hiểu được bây giờ vì θ là một biến ngẫu nhiên chứ không phải là một biến xác định nhất định như trước đây. Sau đó, chúng ta có thể viết xác suất của1N dưới dạng

$$p(\bar{1}N \mid D, D') = \int p(\bar{1}N \mid \theta, D') p(\theta \mid D) d\theta, \tag{2.134}$$

trong đó chúng ta giả định an toàn θ độc lập với tập thử nghiệm D′.

Có thể gây nhầm lẫn khi nhìn thấy p (1N |θ, D ′), vì chúng ta thường nhận được một độ chính xác kiểm tra (tổn thất) khi chúng ta có một mô hình và một bộ kiểm tra cố định. Tuy nhiên, điều này không đúng nói chung, vì chạy một mô hình, đang thực hiện arg max trên hàm năng lượng, thường tự ồn ào hoặc tính toán khó giải quyết nên chúng ta phải sử dụng một số loại ngẫu nhiên.

Sau đó, chúng ta có thể suy ra cái gọi là khoảng đáng tin cậy của độ chính xác của tập thử nghiệm, sao cho độ chính xác của tập thử nghiệm thực sự sẽ được chứa trong khoảng này với

xác suất γ. Hãy để γ = 1−α để thuận tiện. Sau đó, chúng ta đang tìm kiếm một khoảng [l, u]:

$$p(\text{ı}N \le l|D, D') = \alpha 2 \quad \text{và } p(\text{ı}N \ge u|D, D') = \alpha 2 . \tag{2.135}$$

Khoảng đáng tin cậy này là hợp lý khi p(ıN |D, D′) là đơn phương thức, nhưng điều này có thể không đúng. Mật độ xác suất có thể tập trung ở hai tiểu vùng tách biệt tốt, trong trường hợp đó khoảng đáng tin cậy này sẽ không cần thiết rộng và không có thông tin.

Trong trường hợp đó, chúng ta có thể cố gắng xác định một vùng C đáng tin cậy, có thể không liền kề. Khu vực đáng tin cậy được xác định để đáp ứng

$$\text{Với} \int_{\text{ı}N \in C} p(\bar{\text{ı}}N|D, D')d\bar{\text{ı}}N = \gamma, \tag{2.136}$$

$$p(\text{ı}N|D, D') \ge p(\text{ı}'N|D, D') \text{ cho tất cả } \text{ı}N \in C \wedge \text{ı}'N /\in C. \tag{2.137}$$

Điều kiện thứ hai thường được gọi là sự thống trị mật độ. Trên thực tế, vùng đáng tin cậy bao gồm một hoặc nhiều tiểu vùng liền kề sao cho không có điểm nào trong các tiểu vùng này có mật độ thấp hơn bất kỳ điểm nào khác bên ngoài các vùng này. Bằng cách kiểm tra khu vực đáng tin cậy này, chúng ta có thể hiểu rõ tỷ lệ chính xác (hoặc lỗi) thực sự sẽ như thế nào với xác suất γ.

Trong thực tế, chúng ta thường không thể tính toán chính xác bất kỳ đại lượng nào trong số đó, bởi vì phân phối hậu θ|D có thể điều khiển được và thậm chí không được biết đến. Thay vào đó, chúng tôi sử dụng xấp xỉ Monte Carlo bằng cách đào tạo các mô hình nhiều lần, hưởng lợi từ tính ngẫu nhiên trong học tập. Giả sử {θ1, . . . , θM } là một tập hợp các mô hình kết quả. Foreach θm, we draw a sample of the test loss $\bar{\text{ı}}mN$, resulting in $\square\bar{\text{ı}}1N , \ldots , \bar{\text{ı}}MN$. Sau đó, chúng ta có thể sử dụng các mẫu này để mô tả đặc điểm, hiểu và phân tích độ chính xác của kiểm tra thực sự sẽ như thế nào với thuật toán học tập được đưa ra các tập huấn luyện và thử nghiệm, D và D ′.

Nắm bắt các biến thể của bộ huấn luyện. Ngoài sự ngẫu nhiên phát sinh Từ việc xây dựng bộ thử nghiệm cũng như bản thân quá trình học tập, có một nguồn ngẫu nhiên khác mà chúng ta muốn tính đến. Nguồn ngẫu nhiên này phát sinh từ việc xây dựng tập huấn luyện D. Nếu chúng ta tiếp tục từ khu vực đáng tin cậy ở trên, chúng ta không muốn p(ıN |D, D′) mà đúng hơn là

$$p(\text{ı}) = \int_D \int_{D'} p(\text{ı}|D'||D, D')p(D)p(D')dDdD'. \tag{2.138}$$

Nói một cách khác, chúng tôi muốn kiểm tra sự thay đổi của độ chính xác của bộ kiểm traı sau khi loại bỏ cả tập huấn luyện và tập thử nghiệm. Thật không may, chúng tôi thường không có quyền truy cập vào phân phối trên tập dữ liệu. Thay vào đó, chúng ta chỉ được cung cấp một bộ dữ liệu duy nhất được chia thành hai tập dữ liệu; một để đào tạo và một để đánh giá.

Trong trường hợp này, chúng ta có thể sử dụng ý tưởng về cái gọi là lấy mẫu lại bootstrap. Ý tưởng rất đơn giản: (1) chúng tôi lấy mẫu lại N ví dụ từ tập hợp N huấn luyện ban đầu

ví dụ với sự thay thế, (2) tính toán số liệu thống kê mẫu quan tâm và (3) lặp lại (1-2) triệu lần. Trong bước (2), chúng ta có thể chia tập hợp được lấy mẫu lại thành tập huấn luyện được lấy mẫu và tập thử nghiệm được lấy mẫu lại. Chúng tôi sử dụng tập huấn luyện được lấy mẫu lại để đào tạo một mô hình và sau đó là tập thử nghiệm được lấy mẫu lại để đánh giá mô hình được đào tạo để thu được ˆm|D′|. Sau M lần lặp lại như vậy, chúng ta kết thúc với n ¯ ˆ(m)NoMm=1.Các số liệu thống kê được lấy mẫu này sau đó đóng vai trò như một tập hợp các mẫu được rút ra từ p(ˆ), cho phép chúng ta hiểu rõ về cách thuật toán học được đề xuất hoạt động trên vấn đề cụ thể này (không phải một tập dữ liệu cụ thể).

Có nhiều cách để mô tả sự không chắc chắn trong việc đánh giá mức độ hoạt động của bất kỳ thuật toán học nào. Mặc dù chúng ta đã xem xét một số khía cạnh của sự không chắc chắn mà chúng ta nên xem xét trong phần này, nhưng có nhiều cách khác để suy nghĩ về vấn đề này. Ví dụ, nếu chúng ta muốn so sánh hai thuật toán học tập, chúng ta nên tính đến sự không chắc chắn như thế nào? Nếu có sự không chắc chắn trong thuật toán mylearning, có cách nào tốt hơn để hưởng lợi từ sự không chắc chắn này không? Chúng tôi sẽ đề cập đến một số câu hỏi này trong phần còn lại của khóa học.

2.5 Điều chỉnh siêu tham số: Lựa chọn mô hình

Chúng ta thường sử dụng thuật ngữ 'siêu tham số' để chỉ bất cứ điều gì mà chúng ta có thể điều khiển để ảnh hưởng đến việc học. Ví dụ: trong trường hợp gradient ngẫu nhiên, tốc độ học tập α (hoặc bất kỳ núm nào trong bộ lập lịch tốc độ học tập) là siêu tham số ma-jor. Có rất nhiều siêu tham số trong học máy. Ví dụ, các tham số của phân phối mà người ta sử dụng để khởi tạo các tham số mô hình là siêu tham số. Sự lựa chọn / tham số hóa một chức năng năng lượng là một siêu tham số khác rất phức tạp. Chúng ta sẽ sử dụng λ để tham khảo tập hợp của tất cả các siêu tham số.

Chúng ta đã học được cho đến nay rằng các tham số mô hình θ nên được ước tính từ dữ liệu D. Sau đó, chúng ta nên ước tính các siêu tham số λ như thế nào? Chúng ta bắt đầu bằng cách nhận ra rằng học tập tương ứng với

$$\text{Học}(D; \lambda, \varepsilon) = \arg\min_{\theta} \hat{R}(\theta; D). \tag{2.139}$$

Nói cách khác, học tập là quá trình giảm thiểu rủi ro thực nghiệm. Tuy nhiên, quá trình học này không chỉ là một hàm của dữ liệu D mà còn của các siêu tham số λ và ε nhiễu.

Bây giờ chúng ta cần tìm đúng tập hợp siêu tham số. Chức năng khách quan ở đây nên là gì? Chúng ta có thể sử dụng một tập dữ liệu riêng biệt $D_{val} \cap D = \emptyset$, được gọi là tập hợp xác nhận, để đo lường mức độ tốt của mỗi tập siêu tham số:

$$\text{Tune}(D_{val}, D; \varepsilon') = \arg\min_{\lambda} \; \mathop{E}_{\varepsilon} \hat{R}(\text{Học}(D; \lambda, \varepsilon); D_{val}). \tag{2.140}$$

Quá trình điều chỉnh siêu tham số này là một chức năng của cả bộ đào tạo và bộ định hình cũng như một số nguồn nhiễu ε'.

Sau đó, chúng ta có thể có được mô hình cuối cùng bằng cách

$$\hat{\theta} = \text{Học}(D; \text{Giai điệu}(D_{val}, D; \varepsilon'), \varepsilon), \tag{2.141}$$

hoặc

$$\hat{\theta} = \text{Học}\,(D \cup \text{Dval};\ \text{Tune}(\text{Dval}, D;\ \varepsilon'),\ \varepsilon). \tag{2.142}$$

Hơn nữa, chúng ta có thể thu được một số mô hình như vậy bằng cách lấy mẫu lặp đi lặp lại ε.12 Chúng ta sẽ tìm hiểu về những gì chúng ta có thể làm với trường hợp có nhiều mô hình như vậy và ý nghĩa của việc có chúng sau này khi chúng ta nói về máy học Bayes (nếu thời gian cho phép) trong §6.2.

Câu hỏi đặt ra là làm thế nào để thực hiện và thực hiện tối ưu hóa siêu tham số trong Phương trình (2.140). Người ta cũng có thể bị cám dỗ để sử dụng tối ưu hóa dựa trên gradient ở đây, đây là phản ứng đầu tiên hoàn toàn đúng. Tuy nhiên, có một vấn đề lớn. Chúng ta đã thấy vấn đề này trước đó và phải đưa ra sự suy giảm gradient stochas-tic, và vấn đề này là chi phí tính toán của việc tính toán gradient, vì gradient yêu cầu chúng ta phải tính toán

$$\text{JacλLearn}(D;\ l,\ e). \tag{2.143}$$

Có nhiều cách khác nhau để ước tính đại lượng này, chẳng hạn như vi phân tự động chế độ chuyển tiếp cũng như định lý hàm ngầm. Không bao giờ ít hơn, số lượng này cuối cùng là một số lượng khá đắt để tính toán do nhiều yếu tố bao gồm kích thước tập dữ liệu ngày càng tăng |D| và do đó chi phí tối ưu hóa học tập ngày càng tăng.

Do đó, thông thường hơn là coi tối ưu hóa siêu tham số như một bài toán tối ưu hóa hộp đen, nơi chúng ta có thể đánh giá kết quả (nghĩa là tổn thất được tính trên tập hợp xác thực) của một tổ hợp siêu tham số cụ thểnhưng không thể truy cập bất kỳ thứ gì khác của quá trình học tập này.

Tìm kiếm ngẫu nhiên là một trong những phương pháp dựa trên tối ưu hóa hộp đen được sử dụng rộng rãi nhất để tối ưu hóa siêu tham số. Trong tìm kiếm ngẫu nhiên, chúng ta bắt đầu bằng cách xác định một phân phối trước p (λ) trên các siêu tham số λ. Chúng tôi rút ra K sam-ples từ phân phối trước này, {λ1, . . . , λK}, và song song đánh giá chúng bằng cách đào tạo một mô hình bằng cách sử dụng từng siêu tham số được lấy mẫu này. Sau đó, chúng tôi chọn siêu tham số tốt nhất dựa trên rủi ro xác thực, rk = $\hat{R}$ (Learn(D; λk, ε); Dval).

Thay vì chỉ đơn giản là chọn cái tốt nhất, người ta có thể cập nhật trước đósiêu tham số dựa trên

$$\{(\lambda 1, r1), \ldots, (\lambda K, rK)\}, \tag{2.144}$$

sao cho xác suất tập trung vào vùng lân cận của các cấu hình hy-perparameter rủi ro thấp. Toàn bộ quá trình sau đó có thể được lặp lại bằng cách sử dụng phân phối cập nhật này như trước trên các siêu tham số. Ap-proach lặp đi lặp lại này tương tự như phương pháp được sử dụng rộng rãi được gọi là phương pháp entropy chéo [Ru-binstein và Kroese, 2004].

12Chúng ta cũng có thể lặp đi lặp lại lấy mẫu ε′ để thu được nhiều hơn một bộ siêu tham số tốt, nhưng quá trình này có xu hướng quá tốn kém về mặt tính toán để thực tế, vì chúng ta cần phải đào tạo nhiều mô hình mới liên tục cho mục đích tối ưu hóa.

2.5.1Tối ưu hóa dựa trên mô hình tuần tự để điều chỉnh hyperpa-rameter

Thay vì vẽ các cấu hình siêu tham số độc lập, chúng ta có thể nghĩ đến việc vẽ một loạt các cấu hình siêu tham số tương quan. Giả sử Dn−1 =((λ1, r1), . . . , (λn−1, rn−1)) là một loạt các cấu hình siêu tham số và các rủi ro xác nhận liên quan của chúng, được lựa chọn và thử nghiệm cho đến nay. Tại thời điểm n, chúng ta cần quyết định siêu tham số nào để kiểm tra tiếp theo. Quyết định này đòi hỏi chúng tôi phải hỏi chúng tôi muốn cấu hình siêu tham số tiếp theo đáp ứng tiêu chí nào. Có nhiều tiêu chí khả thi, nhưng một tiêu chí cụ thể dễ hiểu là cải thiện dự kiến.

Sự cải thiện dự kiến theo nghĩa đen tính toán mức độ cải thiện mà chúng ta sẽ thấy trong rủi ro dựa trên kỳ vọng. Kỳ vọng này được tính trên phân phối sau, tương tự như Phương trình (2.134):

$$p(r|\lambda, D_{n-1}) = \text{Với} p(r|\lambda, \theta)p(\theta|D_{n-1})d\theta. \quad (2.145)$$

p(r|λ, θ) là một mô hình dự đoán đầu ra R cho hình siêu tham số λ, sử dụng các tham số θ. Xem Phương trình (6.64) và thảo luận xung quanh về cách tạo ra một mô hình như vậy. Sự cải thiện dự kiến của một cấu hình siêu tham số λ sau đó được định nghĩa là

$$EI(\lambda) = E_{r|\lambda,D_{n-1}} [\text{tối đa } (0, \hat{r}_{n-1} - r)] , \quad (2.146)$$

đâu

$$\hat{r}_{n-1} = \min_{i=1,...,n-1} r_i. \quad (2.147)$$

Điều này thường có thể được ước tính bằng cách sử dụng các mẫu:

$$EI (L) \approx \frac{1}{M} \sum_{m=1}^{M} \max(0, \hat{r}_{n-1} - r_m), \quad (2.148)$$

trong đó $r_m \sim r|\lambda, D_{n-1}$.
Sau đó, chúng ta muốn vẽ cấu hình siêu tham số tiếp theo từ phân phối sau:

$$Q (L |D_{n-1}) \propto \exp (\beta EI(\lambda)) , \quad (2.149)$$

trong đó β ≥ 0. Khi β = 0, chúng tôi khôi phục tìm kiếm ngẫu nhiên và khi β → ∞, chúng tôi luôn chọn cấu hình siêu tham số với sự cải tiến mong đợi tốt nhất. Tuy nhiên, thường khó tìm kiếm cấu hình siêu tham số tốt nhất mỗi lần để tối đa hóa sự cải thiện mong đợi và chúng tôi chỉ lấy mẫu cấu hình siêu tham số tiếp theo tỷ lệ thuận với cải tiến dự kiến.

Khi số lượng siêu tham số lớn, tức là $|\lambda| \gg 1$, có thể khó lấy mẫu chính xác từ phân phối này. Trong trường hợp đó, nó có ý nghĩa

để thu hẹp không gian bằng cách làm cho mật độ tập trung cục bộ xung quanh siêu tham số tốt nhất cho đến nay:

$$Q\ (L\ |Dn-1) \propto \exp(\beta EI(\lambda) - \alpha D(\lambda, \hat{}\lambda n-1), \tag{2.150}$$

trong đó ˆλn−1 là cấu hình siêu tham số tốt nhất cho đến nay và D là một số liệu khoảng cách cụ thể cho vấn đề. Sau đó, chúng ta có thể dễ dàng lấy mẫu từ phân phối này bằng cách trước tiên vẽ một tập hợp các mẫu ngẫu nhiên trong vùng lân cận của cấu hình siêu tham số tốt nhất cho đến nay và chọn một trong số chúng theo tỷ lệ thuận với cải tiến dự kiến. Sự thay đổi này giống như tối ưu hóa lặp đi lặp lại, chẳng hạn như sự xuống dốc của asstochastic.

Nhìn chung, cách tiếp cận này, thường được gọi là tối ưu hóa dựa trên mô hình tuần tự [Joneset al., 1998], bao gồm lặp lại ba bước; (1) phù hợp với một pre-dictor nhận thức được sự không chắc chắn về rủi ro được đưa ra một cấu hình siêu tham số, (2) vẽ cấu hình siêu tham số tiếp theo để tối đa hóa sự cải thiện dự kiến theo bộ dự đoán được đào tạo và (3) kiểm tra cấu hình siêu tham số mới được chọn. Tất nhiên, có thể dễ dàng nhận thấy rằng chúng ta không phải kiểm tra chỉ mộtcấu hình siêu tham số tại một thời điểm. Thay vào đó, chúng ta có thể rút ra nhiều mẫu từ phân phối đề xuất q, kiểm tra tất cả chúng (bằng cách đào tạo nhiều mô hình và đánh giá chúng trên tập hợp xác thực) và cập nhật yếu tố dự đoán nhận thức độ không chắc chắn trên tất cả các cặp tích lũy của cấu hình siêu tham số và rủi ro xác thực liên kết. Cách tiếp cận này đã trở thành tiêu chuẩn trên thực tế khi đào tạo một mạng nơ-ron sâu mới với nhiều siêu tham số [Bergstra et al., 2011].

2.5.2Chúng tôi vẫn cần báo cáo độ chính xác của bộ thử nghiệm một cách riêng biệt

Thuật toán tối ưu hóa siêu tham số ở trên có thể được coi là triển khai Tune in

$$\hat{}\theta = \text{Học}\ (D;\ \text{Giai điệu}\ (\text{Dval}, D;\ \varepsilon'),\ \varepsilon), \tag{2.151}$$

Khi chúng tôi tìm thấy cấu hình siêu tham số tốt nhất, chúng tôi đào tạo mô hình cuối cùng trên tập huấn luyện D để có được tham số mô hình cuối cùng ˆθ của chúng tôi. Nó sẽ hoạt động tốt như thế nào?

Thật không may, chúng tôi không thể sử dụng rủi ro xác nhận, vì đó là mục tiêu mà ˆθ đã được chọn. Trong khi đó, khi mô hình này được triển khai trong tự nhiên, thế giới sẽ không tử tế như vậy và một tập hợp các ví dụ được ném vào mô hình này sẽ không hoàn hảo cho mô hình. Do đó, chúng ta cần một tập hợp khác, được gọi là tập thử nghiệm, Dtest để kiểm tra độ chính xác của thử nghiệm. Tập hợp này phải tách biệt với cả tập huấn luyện và xác nhận, và chúng ta có thể báo cáo rủi ro trên tập hợp này như hiện tại, hoặc chúng ta có thể báo cáo thêm số liệu thống kê, như chúng ta đã thảo luận trước đó trong §2.4.3.

Chương 3

Các khối xây dựng của mạng lưới thần kinh

Trước đó trong §2.2.2, chúng ta đã nói về cách biến đổi chung F (x; θ) có thể như thế nào. Như một ví dụ hồi đó, chúng tôi đã xem xét

$$\text{F tuyến tính } (x; \theta) = \sigma (U^T x + c), \tag{3.1}$$

trong đó σ là một phi tuyến tính theo điểm như một đơn vị tuyến tính chỉnh lưu:

$$\sigma (a) = \text{tối đa } (0, a). \tag{3.2}$$

Bằng cách xếp chồng khối này nhiều lần, chúng ta có thể tạo ra một biến đổi phi tuyến ngày càng nhiều, đó là ý tưởng cơ bản đằng sau các bộ nhận thức nhiều lớp [Rumel-hart và cộng sự, 1986]. Chúng ta thường gọi một hàm biến đổi phi tuyến như vậy bao gồm một ngăn xếp các lớp phi tuyến như vậy là mạng nơ-ron sâu.

Lớp tuyến tính1 này không phải là lựa chọn duy nhất, mặc dù điều này được sử dụng rộng rãi do không có sai lệch quy nạp. Nghĩa là, nếu chúng ta không có bất kỳ kiến thức cụ thể nào về đầu vào x, thì có thể an toàn khi coi nó như một vectơ chiều hữu hạn phẳng và cung cấp nó qua một ngăn xếp các lớp tuyến tính này. Tuy nhiên, chúng ta thường biết về các cấu trúc cơ bản của một quan sát. Ví dụ, nếu chúng ta đang xử lý một tập hợp các mục như một quan sát, chúng ta muốn phép biến đổi của chúng ta trở thành hoán vị cân bằng hoặc bất biến, vì không có trật tự cố hữu giữa các mục trong một tập hợp.

Trong chương (ngắn) này, chúng tôi sẽ giới thiệu thêm một số khối xây dựng cơ bản này để xây dựng một mạng nơ-ron sâu. Ngoài các khối này, chúng tôi có thể sáng tạo nhất có thể miễn là các khối mới được thiết kế của bạn có thể phân biệt được cả thông số và đầu vào của riêng chúng. Một số khối có thể thiếu bất kỳ thông số nào và điều đó hoàn toàn ổn. Ví dụ, tôi có thể có một khối chỉ đơn giản đảo ngược thứ tự của các mục trong một đầu vào theo cách xác định.

1Mặc dù khối này không phải là tuyến tính, nhưng chúng ta thường gọi khối này là khối tuyến tính hoặc lớp tuyến tính.

3.1 Chuẩn hóa

Chúng ta hãy xem xét hàm năng lượng bình phương đơn giản từ phương trình (6.64), với tính phi tuyến tính bất danh:

$$e'([x, y], (u, c)) = \frac{1}{2} (u^{T}x + c - y)2. \tag{3.3}$$

Chúng ta sẽ giả định thêm rằng y là một vô hướng và do đó u là một vectơ chứ không phải là ma trận.

Tổn thất tổng thể sau đó là

$$J(\theta) = \frac{1}{N} \sum_{n=1}^{N} e'([xn, trong], (u, c)) = \frac{1}{2N} \sum_{n=1}^{N} (u^{T}xn + c - trong)2. \tag{3.4}$$

Độ dốc của tổn thất w.r.t. u sau đó là

$$\nabla u = \frac{1}{N} \sum_{n=1}^{N} (u^{T}xn + c - yn)x^{T}n , \tag{3.5}$$

$$\nabla c = \frac{1}{N} \sum_{n=1}^{N} (u^{T}xn + c - trong). \tag{3.6}$$

Cho đến nay, không có gì khác biệt so với các bài tập trước đây của chúng tôi. Bây giờ chúng ta xem xét Hessian của sự mất mát:

$$H = \begin{bmatrix} \frac{1}{N} \sum_{n=1} xnx^{T}n & \frac{1}{la} \sum_{n=1} xn \\ \frac{1}{la} \sum_{n=1} xn & 1 \end{bmatrix}. \tag{3.7}$$

Ma trận Hessian cho chúng ta biết về độ cong của hàm mục tiêu và liên quan trực tiếp đến độ khó của việc tối ưu hóa bằng cách tiếp cận dựa trên gradient. Đặc biệt, tối ưu hóa dựa trên gradient khó khăn hơn khi số điều kiện lớn hơn, trong đó số điều kiện được định nghĩa là

$$k = \frac{|\text{maxi li}(H)|}{|\text{nhỏ}\ \lambda i (H)|} \geq 1, \tag{3.8}$$

trong đó λi(H) là giá trị riêng thứ i của H.
Không nằm ngoài phạm vi của khóa học này để thảo luận sâu về lý do tại sao số điều kiện lại quan trọng để tối ưu hóa. Ở cấp độ cao, bạn có thể nghĩ về các giá trị riêng củaHessian của hàm mục tiêu như định lượng mức độ kéo dài hàm này dọc theo các hướng vectơ riêng liên quan. Nghĩa là, nếu giá trị riêng của aneigenvector lớn, điều đó có nghĩa là giá trị hàm thay đổi đáng kể hơn theo hướng ve-toơ riêng này. Khi giá trị mục tiêu thay đổi rất khác nhau trên tất cả các hướng này (hướng trực giao, vì chúng là hướng vectơ riêng của ma trận đối xứng), sự xuống dốc ngẫu nhiên bị ảnh hưởng, vì nó sẽ dễ dàng dao động dọc theo các hướng với những thay đổi dốc trong khi nó sẽ không đạt được nhiều tiến bộ dọc theo các hướng chỉ với những thay đổi nhỏ. Do đó, chúng tôi muốn

các giá trị riêng của Hessian tương tự nhau, để một thuật toán op-timization lặp đi lặp lại như vậy hoạt động tốt. Để thảo luận chặt chẽ hơn, hãy tham khảo cuốn sách tối ưu hóa lồi yêu thích của bạn [Nocedal và Wright, 2006].

Dựa trên định nghĩa này, ma trận nhận dạng có điều kiện tối thiểu số-ber. Nói cách khác, chúng ta có thể chuyển đổi ma trận Hessian thành ma trận nhận dạng, để tạo điều kiện tối ưu hóa dựa trên gradient [LeCun et al., 1998]. Trong trường hợp cụ thể này, vì ma trận Hessian không phụ thuộc vào θ mà chỉ phụ thuộc vào các quan sát xn, chúng ta có thể chỉ cần chuyển đổi đầu vào trước bằng cách

$$(1)\ \text{xn} \leftarrow \text{xn} - 1\underline{\text{N}} \overset{\text{N}}{\text{X}} \text{xn}' \qquad \text{(định tâm)} \qquad (3.9)$$

$$(2)\ \text{xn} \leftarrow \frac{1}{\text{lâ}} \overset{\text{N}}{\text{X}} \text{xn}'\text{xTn}' \qquad \text{xn} \qquad \text{(làm trắng)} \qquad (3.10)$$

Điều này sẽ dẫn đến ma trận Hessian nhận dạng, cải thiện sự hội tụ của tối ưu hóa dựa trên gradient.

Chuẩn hóa như vậy là chìa khóa thành công trong tối ưu hóa, nhưng nó rất khó khăn để áp dụng nó trong thực tế một cách chính xác, vì ma trận Hessian thường không đứng yên khi chúng ta đào tạo một mạng nơ-ron sâu. Ma trận Hessian thay đổi khi chúng ta cập nhật các tham số mô hình và không có cách nào có thể xử lý được để biến ma trận Hessian thành ma trận nhận dạng. Hơn nữa, việc đảo ngược ma trận Hessian này thậm chí còn khó khăn hơn. Tuy nhiên, hóa ra việc chuẩn hóa (như một phiên bản làm trắng yếu hơn) đầu vào cho mỗi khối giúp ích cho việc học. Chuẩn hóa như vậy cũng có thể được coi là một khối xây dựng, và chúng ta hãy xem xét một vài cái được sử dụng rộng rãi ở đây.

Chuẩn hóa hàng loạt [Ioffe và Szegedy, 2015]. Đây là một trong những bản dựng-ing đã châm ngòi cho cuộc cách mạng trong học sâu, tạo điều kiện thuận lợi rất nhiều cho việc học:

$$\text{Fbatch−norm}(x;\ \theta = (m, s)) = m + \exp(s) \cdot ((x - m) \oslash s)\ , \qquad (3.11)$$

trong đó μ và σ2 là phương sai trung bình và đường chéo của đầu vào cho khối này. Bởi vì nghịch đảo của ma trận hiệp phương sai đầy đủ, thường tương tự như ma trận Hessian cho đến một số hạng cộng, rất tốn kém, chúng ta chỉ xem xét đường chéo của ma trận hiệp phương sai, dễ dàng đảo ngược. Thay vì sử dụng bộ đào tạo đầy đủ để ước tính μ và σ2 , điều này sẽ rất tốn kém, chúng tôi sử dụng minibatch ở mỗi bản cập nhật trong quá trình đào tạo để có được ước tính ngẫu nhiên của hai đại lượng này. Thực hành này hoàn toàn ổn trong quá trình đào tạo nhưng nó trở thành vấn đề khi mô hình được triển khai, vì mô hình sẽ nhận được một ví dụ tại một thời điểm. Với một ví dụ duy nhất, chúng ta không thể ước tính μ và σ2, hoặc nếu chúng ta làm vậy, nó sẽ chỉ đơn giản là trừ đi toàn bộ đầu vào của nó. Thay vào đó, thông thường là ước tính lại đầy đủ μ và σ2 bằng cách sử dụng bộ đào tạo đầy đủ sau khi đào tạo kết thúc hoặc giữ ước tính chạy của μ và σ2 trong quá trình đào tạo và sử dụng chúng sau khi đào tạo kết thúc.

Chuẩn hóa lớp [Ba và cộng sự, 2016]. Thay vì chuẩn hóa các giá trị trên ví dụ, có thể chuẩn hóa các giá trị trong mỗi ví dụ trên các kích thước. Khi chúng ta làm như vậy, chúng ta gọi nó là chuẩn hóa lớp:

$$\text{Flayer-norm}(x;\theta=(m,s)) = m + \frac{\exp(s)\sqrt{\frac{1}{|x|}\sum_{i=1}^{|x|}(x_i-\mu)^2+\epsilon}}{} \left(\frac{x - \frac{1}{|x|}\sum_{i=1}^{|x|} x_i}{}\right), \tag{3.12}$$

trong đó chúng ta giả định x là một vectơ hữu hạn. Chúng ta chắc chắn có thể sửa đổi nó để đối phó tốt hơn với các loại đầu vào khác, nhưng điều đó nằm ngoài phạm vi của khóa học này. Không rõ tại sao điều này lại giúp tối ưu hóa, nhưng nó đã được phát hiện là tạo điều kiện thuận lợi cho việc học trong nhiều thí nghiệm quy mô lớn.

Không giống như chuẩn hóa hàng loạt, người ta phải cẩn thận khi sử dụng chuẩn hóa lớp, vì nó có thể dễ dàng phá vỡ mối quan hệ giữa các ví dụ khác nhau. Ví dụ, hãy tưởng tượng một bài toán phân loại nhị phân đơn giản, trong đó lớp dương bao gồm tất cả các vectơ đầu vào có chuẩn Euclid nhỏ hơn 1 và lớp phủ định của tất cả các vectơ đầu vào có chuẩn Euclid lớn hơn hoặc bằng 1. Giả sử x+ = [0.9, 0] và x− = [2, 0]. Sau khi chuẩn hóa lớp, chúng được chuyển thành ˆx+ = [0,5, −0,5] và ˆx− = [0,5, −0,5], tương ứng. Đột nhiên, hai đầu vào này, thuộc về hai lớp riêng biệt, không thể phân biệt được với nhau. Điều này xảy ra, không giống như chuẩn hóa hàng loạt, vì chuẩn hóa được áp dụng khác nhau cho các phiên bản khác nhau, trong khi chuẩn hóa hàng loạt áp dụng cùng một chuẩn hóa cho tất cả các phiên bản cùng một lúc.

3.2 Các khối tích chập

Nhiều vấn đề trong học máy tập trung vào việc phát hiện các mẫu trong đầu vào xuất hiện nhiều lần trong tập dữ liệu đào tạo. Ví dụ, hãy xem xét một thuật toán phát hiện đối tượng. Ban đầu chúng ta không biết loại mẫu nào được coi là đại diện cho từng đối tượng. Do đó, việc học phải tìm ra các mẫu nào xuất hiện liên tục bất cứ khi nào đầu vào được liên kết với một nhãn đối tượng cụ thể. Tuy nhiên, các mẫu này không phải là toàn cầu mà được bản địa hóa, vì đối tượng có thể xuất hiện ở bất kỳ đâu trong hình ảnh đầu vào. Nó có thể xuất hiện ở trung tâmnhưng cũng xuất hiện ở bất kỳ góc nào của hình ảnh, không ảnh hưởng đến đối tượngnhận dạng. Nói cách khác, một máy dò đối tượng phải là bất biến tịnh tiến.2

2Chúng ta nói F là biến đổi tương đương với một phép biến đổi cụ thể T khi

$$F(T(X)) = T(F(X)). \tag{3.13}$$

Chúng ta nói F là invairant đối với một biến đổi T cụ thể khi

$$F(T(X)) = F(X). \tag{3.14}$$

Bất kỳ bất biến nào cũng có thể được thực hiện dưới dạng một ngăn xếp các khối cân bằng được thực hiện bởi một toán tử kiêm, chẳng hạn như tính tổng. Do đó, chúng ta cần thực hiện một khối cân bằng dịch thuật. Trong phần này, chúng tôi xem xét cái gọi là khối tích chập, hoặc chính xác hơn là khối tương quan.

Chúng ta bắt đầu bằng cách xem xét một chuỗi thời gian rời rạc dài vô hạn, x = [. . . , xt−1, xt, xt+1, . . .]với |x| → ∞, như một đầu vào cho khối này. Mỗi mục xt là một vectơ thực tế hữu hạn. Tham số của khối này là một tập hợp các dãy bộ lọc có độ dài hữu hạn, f k = [f k1 , f k2 , . . . , f k2M + 1] với M ≪ ∞ và k = 1, . . . , K. Tương tự như xt, eachf kt cũng là một vectơ thực hữu hạn chiều với |f kt | = |xt|. Sau đó, khối tích chập trả về một chuỗi thời gian dài vô hạn, h = [. . . , ht−1, ht, ht+1, . . .], trong đó |ht| = K.

Giả sử hkt là phần tử thứ k của ht. Sau đó, chúng ta tính toán nó như

$$HKT = \sum_{m'=-M}^{m'=M} x_{t+m'} f_{km'+M+1}. \tag{3.15}$$

Nói cách khác, chúng tôi áp dụng bộ lọc thứ k f k ở mỗi vị trí t để kiểm tra mức độ giống nhau (theo nghĩa là tích chấm) tín hiệu tập trung tại t với bộ lọc f k.

Một cách khác để viết nó ra là

$$ht = \sum_{m'=-M}^{m'=M} Fm' + M + 1xt + m', \tag{3.16}$$

đâu

$$Fm = \begin{matrix} \square & \square \\ \square\square\square & \square\square\square\square \\ \square f & \text{Đường} \times K \\ 1mf & \\ 2m... & \end{matrix} \tag{3.17}$$

với d = |xt|. Các tham số đầy đủ của khối tích chập 1 chiều này có thể được tổng hợp dưới dạng tensor 3 chiều có kích thước d × K × (2M + 1).

Khá đơn giản để thấy rằng phép toán này là dịch tương đương. Nếu chúng ta dịch chuyển mỗi xt theo δ, ht kết quả sẽ dịch chuyển δ mà không ảnh hưởng đến giá trị tính toán của nó. Thật không may, trong thực tế, điều này không hoàn hảo, vì chúng ta không làm việc với một chuỗi dài vô hạn. Chúng ta phải quyết định làm thế nào để xử lý các ranh giới của dãy với một dãy có độ dài hữu hạn, và sự lựa chọn này sẽ ảnh hưởng đến mức độ đẳng phương sai tịnh tiến gần các ranh giới. Tuy nhiên, thảo luận chi tiết về cách chúng ta xử lý ranh giới nằm ngoài phạm vi của khóa học này.

Bây giờ chúng ta có thể dễ dàng mở rộng tích chập 1-D này thành tích chập N-D. Ví dụ, tích chập 2D sẽ hoạt động trên một hình ảnh lớn vô hạn và tích chập 3D trên một video dài và lớn vô hạn. Hơn nữa, chúng tôi có thể mở rộng nó bằng cách giới thiệu các tính năng khác nhau, chẳng hạn như sải chân. Những điều này cũng nằm ngoài phạm vi của khóa học này, nhưng tôi khuyên bạn nên đọc nửa đầu của tác phẩm kinh điển của LeCun et al.[1998].

3.3 Các khối lặp đi lặp lại

Thông thường, phương sai đều hoặc bất biến mạnh có xu hướng quá nghiêm ngặt. Có lẽ chúng ta chỉ muốn phương sai đẳng trong một bối cảnh cụ thể chứ không phải trong bối cảnh khác. Tuy nhiên, rất khó để thực hiện nó theo nghĩa chặt chẽ. Chúng ta có thể tiến thêm một bước trong mức độ trừu tượng hóa và làm việc để áp dụng cùng một toán tử lặp đi lặp lại trên đầu vào. Đây là ý tưởng cốt lõi đằng sau một khối lặp lại.

Một khối lặp lại hoạt động trên một chuỗi các mục đầu vào (x1, x2, . . . , xT), giống như khối tích chập 1-D ở trên. Khối này bao gồm một mạng nơ-ron được áp dụng lặp lại cho xt tuần tự (nghĩa là từng mạng một.) Mạng thần kinh này lấy làm đầu vào sự nối của xt và bộ nhớ (hoặc trạng thái ẩn) ht−1 và trả về một bộ nhớ được cập nhật ht:

$$ht = F([xt, ht-1]; \theta r), \tag{3.18}$$

trong đó θr là các tham số của hàm lặp lại này F . θr bao gồm trạng thái InitalHidden H0. Khi chúng ta quét dãy đầu vào bằng F, khối lặp lại trả về dãy có cùng độ dài bằng cách nối tất cả các ht: (h1, h2, . . . , hT).

Ưu điểm của một khối lặp lại như vậy so với ví dụ như tích chập 1-D ở trên là nó có kích thước ngữ cảnh không giới hạn. Trong tích chập 1-D, bất kỳ đầu ra nào tại thời điểm t chỉ phụ thuộc vào 2M + 1 vectơ đầu vào tập trung tại t. Mặt khác, khối lặp lại tính đến tất cả các đầu vào lên đến t để tính toán trạng thái ẩn ht tại thời điểm t. Hơn nữa, bằng cách chỉ cần xếp chồng hai khối lặp lại với sự đảo ngược trình tự ở giữa, chúng ta có thể làm cho mỗi vectơ đầu ra phụ thuộc vào toàn bộ trình tự một cách dễ dàng.

Một ví dụ điển hình (và đơn giản) về các khối lặp lại được sử dụng rộng rãi (và dễ sử dụng) là một đơn vị lặp lại có cổng [GRU; Cho và cộng sự, 2014] được định nghĩa là

$$FGRU = ut \odot ht-1 + (1 - ut) \odot \tilde{}ht, \tag{3.19}$$

đâu

$$rt = \sigma(Wrxt + Urht-1 + br) \quad \text{(Đặt lại cổng)} \tag{3.20}$$

$$ut = \sigma(Wuxt + Uuht-1 + bu) \quad \text{(Cổng cập nhật)} \tag{3.21}$$

$$\tilde{}ht = \tanh(Whxt + Uh(rt \odot ht-1) + bh) \quad \text{(Tiểu bang ứng cử viên)} \tag{3.22}$$

Sự kết hợp tuyến tính (có trọng số) này đã được chứng minh là giải quyết hiệu quả vấn đề gradient biến mất [Bengio et al., 1994], và đã trở thành một thực hành tiêu chuẩn trong học máy trong thập kỷ qua hoặc lâu hơn [He et al., 2016].

3.4 Phương sai đẳng hoán vị: chú ý

Chúng ta thường phải đối mặt với tình huống đầu vào cho một khối là một tập hợp các vectors X = {xi}Ni=1. Chúng ta muốn biến đổi mỗi mục xk trong ngữ cảnh của tất cả các mục khác trong tập hợp này, kết quả là một tập hợp vectơ khác H = {hi}Ni=1. Chúng tôi muốn điều này

lớp có biến đổi đều với hoán vị sao cho F ((xσ(i))Ni=1) = (hσ(i))Ni=1, trong đó σ : {1, . . . , N } → {1, . . . , N } là một toán tử hoán vị. Hãy xem xét một cách chuẩn để xây dựng một khối cân đẳng hoán vị như vậy.

Khối này bắt đầu với ba khối tuyến tính:

$$k_i = \text{Tuyến tính } (x_i; \theta_k), \tag{3.23}$$

$$q_i = \text{Tuyến tính } (x_i; \theta_q), \tag{3.24}$$

$$v_i = \text{Tuyến tính } (x_i; \theta_v). \tag{3.25}$$

Chúng ta được gọi là vectơ khóa, truy vấn và giá trị của mục thứ i xi.
Đối với mỗi mục thứ j xj, chúng tôi kiểm tra mức độ tương thích của nó với mục thứ i hiện tạixi:

$$A_{ji} = \frac{\exp(q_i^T k_j)}{\sum_{j'=1}^{N} \exp(q_i^T k_{j'})}. \tag{3.26}$$

Chúng tôi chuẩn hóa nó thành một bằng cách sử dụng softmax.
Bây giờ, chúng ta sử dụng các trọng số quan trọng này để tính trung bình có trọng số của các giá trị:

$$\hat{v}_i = \sum_{j=1}^{N} \text{Chương VI.} \tag{3.27}$$

Thông thường lặp lại quá trình này K lần để sản xuất

$$\hat{v}_i \leftarrow \begin{matrix} \hat{v}_1 \\ \dots \\ \hat{v}_K \end{matrix}_i . \tag{3.28}$$

Khi chúng ta làm điều này, mỗi K quá trình như vậy được gọi là đầu chú ý.
Tại thời điểm này, ˆvi là một hàm tuyến tính của đầu vào X = {x1, . . . , xN }.
Chúng tôi muốn giới thiệu một số tính phi tuyến tính ở đây bằng cách giới thiệu lớp tuyến tính cuối cùng cùng với một kết nối dư:

$$h_i = \text{Flayer}-\text{norm}(F_{\sigma\text{linear}}(\hat{v}_i; \theta_h); \theta_l) + \text{Flinear}(x_i; \theta_r), \tag{3.29}$$

trong đó việc thiếu chỉ số trên trong nhiệm kỳ thứ hai có nghĩa là không có tính trực tuyến. Nếu |hi| = |xi|, theo thông lệ, người ta sẽ cố định θr = (I, 0). Thông thường thêm khối chuẩn hóa alayer sau ˆvi hoặc tại hi, để tạo điều kiện tối ưu hóa.
Khi được thực hiện trong một khối duy nhất, khối này thường được gọi là khối chú ý (nhiều đầu) [Bahdanau và cộng sự, 2015, Vaswani và cộng sự, 2017].

Mã hóa vị trí. Một cách khác để nghĩ về khối chú ý ở trên là để xem nó như một cách để xử lý đầu vào có kích thước thay đổi. Bất kể kích thước của bộ đầu vào, khối chú ý này có thể hoạt động với đầu vào. Do đó, thật hấp dẫn khi sử dụng khối chú ý cho một chuỗi có độ dài thay đổi, đó là động lực ban đầu đằng sau khối chú ý. Có một rào cản phải là

vượt qua trong trường hợp đó. Nghĩa là, chúng ta phải đảm bảo rằng mỗi mục trong một trình tự được đánh dấu với vị trí của nó.

Có hai cách tiếp cận chính cho điều này. Cách tiếp cận đầu tiên dựa trên đánh dấu phụ gia. Đối với mỗi vị trí i, hãy cho ei là một vectơ có kích thước |x| và đại diện cho vị trí thứ i. Có nhiều cách để xây dựng vectơ này, và đôi khi thậm chí có thể học vectơ này từ dữ liệu, mặc dù chúng ta chỉ có thể xử lý độ dài được nhìn thấy trong quá trình đào tạo trong trường hợp thứ hai. Một cách tiếp cận cụ thể là sử dụng các hàm hình sin để mỗi chiều của ei nắm bắt các tốc độ khác nhau tại đó vị trí thay đổi. Chẳng hạn

$$\text{edi} = \begin{cases} \text{(không có thân thể)} \dfrac{iLi/|x|}{} \square, & \text{Nếu } \mod 2 = 0 \\ \dfrac{iL(i-1)/|x|}{}, & \text{nếu tôi } \mod 2 = 1 \end{cases} \tag{3.30}$$

trong đó L là một siêu tham số và thường được đặt thành 10000. Sau đó, vectơ này được thêm vào mỗi mục đầu vào, tức là xi + ei trước khi được đưa vào khối chú ý.

Cách tiếp cận đầu tiên, cách tiếp cận cộng thêm, giúp khối chú ý dễ dàng nắm bắt vị trí của mỗi vectơ, bởi vì các vectơ neary có xu hướng có các nhúng vị trí tương tự và nắm bắt pat-tern dựa trên vị trí tuyệt đối, vì mỗi vị trí tuyệt đối được biểu diễn bằng vectơ nhúng vị trí duy nhất của nó. Tuy nhiên, rất khó đối với khối chú ý để nắm bắt các mô hình dựa trên các vị trí tương đối vượt ra ngoài địa phương đơn giản.

Đặc biệt, hãy xem xét cách cái gọi là trọng số chú ý trên mục thứ j cho mục thứ i được tính trong Phương trình (3.26). Trọng số tỷ lệ thuận với tích chấm giữa vectơ truy vấn thứ i và vectơ khóa thứ j:

$$\mathrm{q}^{\mathrm{T}}\mathrm{i}\,\mathrm{kj} = (\mathrm{Wq}(\mathrm{xi} + \mathrm{ei}))^{\mathrm{T}}(\mathrm{Wk}(\mathrm{xj} + \mathrm{ej})) \tag{3.31}$$

$$= \mathrm{x}^{\mathrm{T}}\mathrm{i}\ \mathrm{W}\ ^{\mathrm{T}}\mathrm{q}\ \mathrm{Wkxj} + \mathrm{e}^{\mathrm{T}}\mathrm{i}\ \mathrm{W}\ ^{\mathrm{T}}\mathrm{q}\ \mathrm{Wkxj} + \mathrm{x}^{\mathrm{T}}\mathrm{i}\ \mathrm{W}\ ^{\mathrm{T}}\mathrm{q}\ \mathrm{Wkek} + \mathrm{e}^{\mathrm{T}}\mathrm{i}\ \mathrm{W}\ ^{\mathrm{T}}\mathrm{q}\ \mathrm{Wkek}, \tag{3.32}$$

trong đó chúng tôi giả định không có vectơ thiên vị cho cả vectơ truy vấn và vectơ khóa. Từ thuật ngữ đầu tiên trong biểu thức mở rộng, chúng ta nhận thấy rằng mối quan hệ dựa trên nội dung giữa đầu vào thứ i và đầu vào thứ j phần lớn độc lập với vị trí của chúng. Nói cách khác, mối quan hệ ngữ nghĩa giữa hai đầu vào này là đứng yên trên các vị trí tương đối của chúng, điều này có thể hạn chế trong nhiều ứng dụng xuôi dòng.

Tập trung vào thuật ngữ đầu tiên ở trên, chúng ta có thể nghĩ ra một cách để đảm bảo rằng mối quan hệ ngữ nghĩa theo cặp này phụ thuộc vào vị trí. Cụ thể hơn, chúng tôi muốn nó phụ thuộc vào vị trí tương đối giữa xi và xj:

$$\langle \mathrm{qi}, \mathrm{kj} \rangle_{j-i} = \mathrm{q}^{\mathrm{T}}\mathrm{i}\ \mathrm{R}^{\mathrm{T}}\mathrm{i}\ \mathrm{Rjxj}, \tag{3.33}$$

trong đó Rm là một ma trận trực giao được tham số hóa bởi một vị trí vô hướng mand thay đổi trơn tru w.r.t. m. Một cách để xây dựng một ma trận trực giao như vậy là xây dựng một ma trận chéo khối trong đó ma trận xoay 2-D được lặp lại dọc theo

Đường chéo:

$$R_m = \begin{bmatrix} R_{2,1}(m) & 0 & \cdots & 0 \\ 0 & R_{2,2}(m) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{2|x|/2}(m) \end{bmatrix} \quad (3.34)$$

trong đó $R_{2k}(m)$ là ma trận quay 2 chiều xoay một vectơ thực 2 chiều và được định nghĩa là

$$R_{2k}(m) = \begin{bmatrix} \cos(mLk/|x|) & -\sin(mLk/|x|) \\ \sin(mLk/|x|) & \cos(mLk/|x|) \end{bmatrix}. \quad (3.35)$$

Nói cách khác, chúng ta xoay mọi cặp phần tử của vectơ truy vấn/khóa dựa trên vị trí của nó trước khi tính tích chấm giữa hai vectơ này. Vì vòng quay này phụ thuộc vào vị trí tương đối giữa truy vấn và vectơ khóa, cách tiếp cận này có thể nắm bắt mối quan hệ ngữ nghĩa phụ thuộc vào vị trí giữa đầu vào thứ i và đầu vào thứ j. Ý tưởng này đã trở thành một trong những cách tiếp cận tiêu chuẩn để kết hợp thông tin vị trí trong khối chú ý trong những năm gần đây [Su et al., 2021].

Chương 4

MachineLearning xác suất và Học không giám sát

4.1 Giải thích xác suất của chức năng năng lượng

Mặc dù chúng ta đã học về cách biến một hàm năng lượng thành một hàm xác suất trong §2.1.2, chúng ta sẽ đi sâu hơn một chút trong phần này, vì nó sẽ giúp chúng ta rút ra một loạt các thuật toán học máy trong chương này.

Hàm năng lượng được xác định với quan sát x, biến không quan sát (la-tent) z và các tham số mô hình θ: e(x, z, θ). Bây giờ, chúng ta sẽ giả định rằng θ không phải là một biến ngẫu nhiên, không giống như x và z. Sau đó, chúng ta có thể tính toán phân phối chung trên x và z như

$$p(x, z; \theta) = \frac{\exp(-e(x, z, \theta))RR}{\exp(-e(x', z', \theta))dx'dz'}. \tag{4.1}$$

Tất nhiên, thường (nếu không phải hầu như luôn luôn) khó tính toán hằng số chuẩn hóa (hoặc hàm phân vùng) trong mẫu số. Một thách thức như vậy gợi ý về một cách tiếp cận khác cho cùng một vấn đề. Thay vì định nghĩa một hàm en-ergy trước và sau đó suy ra hàm xác suất, tại sao không trực tiếp định nghĩa hàm xác suất? Rốt cuộc, chúng ta có thể khôi phục hàm năng lượng cơ bản cho một hàm xác suất lên đến một hằng số:

$$e(x, z, \theta) = -\log p(x, z; \theta) + \log Z(\theta). \tag{4.2}$$

Trên thực tế, thậm chí có thể dễ dàng hơn để phân tách hàm xác suất chung p(x, z)

Hơn nữa,1 Sử dụng quy tắc xác suất chuỗi:

$$p(x, z) = p(z)p(x|z). \tag{4.3}$$

Sự phân hủy như vậy mang lại cho chúng ta một cách thú vị để giải thích mô hình xác suất này. z là một biến tiềm ẩn xác định các tính chất nội tại của quan sát x. Do đó, trước tiên chúng ta vẽ một cấu hình thuộc tính nội tại ztừ phân phối trước p(z). Với cấu hình thuộc tính nội tại z này, chúng ta vẽ quan sát thực tế x.

Ví dụ, bạn có thể tưởng tượng rằng z đề cập đến một phạm trù đối tượng (một, một con mèo, một chiếc xe hơi, v.v.) Đầu tiên chúng ta vẽ một phạm trù của một đối tượng mà chúng ta muốn vẽ bằng cách chọnz, theo phân phối trước p(z). Sự phân bố trước này phản ánh tần số của các loại đối tượng này trên thế giới. Với đối tượng categoryz, bây giờ chúng ta có thể vẽ đối tượng bằng cách vẽ x từ p(x|z). Phân phối có điều kiện này gói gọn tất cả các biến thể của đối tượng z ở dạng trực quan của nó, chẳng hạn như điều kiện sét, nền, kết cấu, v.v.

Một phân phối khác, hoặc hàm xác suất, mà chúng ta quan tâm là phân phối vị trí trên z cho quan sát x. Tiếp tục từ ví dụ trên, chúng ta có thể nghĩ đến việc cố gắng suy ra đối tượng z nào trong một bức tranh nhất định x de-picts. Suy luận như vậy thường không hoàn hảo và dẫn đến sự phân phối trên các danh mục đối tượng thay vì chọn một danh mục chính xác. Chúng ta có thể suy ra phân phối này bằng cách sử dụng quy tắc Bayes:

$$p(z|x) = \frac{p(x|z)p(z)R\ p(x|z')p(z')dz' =}{p(x|z)p(z)p(x)}. \tag{4.4}$$

Cũng giống như trước đó khi chúng ta cố gắng biến hàm năng lượng thành một hàm xác suất, suy luận hậu thường khó giải quyết về mặt tính toán do hằng số chuẩn hóa trong mẫu số: R p(x|z′)p(z′)dz′.

Với các hàm xác suất này trong tay, bây giờ chúng ta có thể định nghĩa một hàm genericloss:

$$\text{Lll}(x, \theta) = -\log \overset{\text{Với}}{p}(x|z; \theta)p(z; \theta)dz. \tag{4.5}$$

Chúng ta thường gọi hàm mất mát này là khả năng log âm hoặc xác suất log. Nếu bạn không thoải mái với việc có một quan sát một biến x, chúng ta có thể viết điều này theo cặp đầu vào-kết quả (x, y):

$$\text{Lll}([x, y], \theta) = -\log \overset{\text{Với}}{p}(y|x, z; \theta)p(x)p(z; \theta)dz \tag{4.6}$$

$$= -\text{nhật ký} \overset{\text{Với}}{p}(y|x, z; \theta)p(z; \theta)dz + \text{const.}, \tag{4.7}$$

trong đó chúng ta giả định rằng p(x) được cho đơn giản và không được tối ưu hóa với các tham số riêng của nó. Khi z không tồn tại, nó giảm xuống tổn thất entropy chéo từEq. (2.30):

$$\text{Lll}([x, y], \theta) = -\log p(y|x; \theta). \tag{4.8}$$

1Như thường lệ, chúng ta sẽ bỏ qua θ nếu sự tồn tại của nó, hoặc thiếu nó, rõ ràng từ ngữ cảnh.

Trong phần còn lại của chương này, chúng ta tập trung vào trường hợp chúng ta có các quan sát chỉ đầu vào. Chúng ta thường gọi một thiết lập như vậy là học tập không giám sát.

4.2 Suy luận biến thiên và mô hình hỗn hợp Gaussian

Chúng ta sẽ rút ra một cái gì đó kỳ diệu trong phần này, mặc dù nó sẽ không có vẻ kỳ diệu chút nào khi nhìn lại vào cuối phần này. Để làm như vậy, trước tiên chúng ta hãy nói lại rằng thật khó để suy ra phân phối hậu p(z|x) trên biến tiềm ẩn cho một quan sát, trừ khi chúng ta đặt ra những ràng buộc nghiêm trọng rõ ràng trên các dạng của p(x|z) và p(z).2 Thay vì tính toán trực tiếp p(z|x), có lẽ chúng ta có thể tìm thấy một proxy q(z; φ(x)) cho phân phối hậu chính xác này, được gọi là gần đúng. Hàm xác suất hậu gần đúng này được tham số hóa bởiφ(x), trong đó chúng ta sử dụng (x) để biểu thị rằng các tham số này là cụ thể cho x. Khi nó không gây nhầm lẫn, chúng ta sẽ bỏ (x) ở đây và ở đó để vừa ngắn gọn vừa rõ ràng.

Bây giờ chúng ta đang phải đối mặt với một nhiệm vụ để làm cho proxy q(z; φ(x)) trở thành một xấp xỉ tốt với hậu thực p(z|x). Chúng tôi sẽ làm như vậy bằng cách giảm thiểu phân kỳ Kullback-Leibler (KL) được định nghĩa là

$$D_{KL}(q\|p) = -\int_{\text{nhật ký } q(z;\varphi(x))} \frac{p(z|x)q(z;}{\varphi(x))}\,dz \tag{4.9}$$

$$= -E_{z\sim q}[\log p(z|x)] - H(q) \geq 0 \tag{4.10}$$

trong đó H(q) là entropy của q được định nghĩa là

$$H(q) = -\int q(z)\log q(z)dz. \tag{4.11}$$

Điều quan trọng là phải lưu ý sự bất đẳng thức ở trên, nghĩa là phân kỳ KL theo định nghĩa là không âm.

Chúng ta hãy tiếp tục từ phân kỳ KL:

$$D_{KL}(q\|p) = -\int_{\text{nhật ký } q(z;\varphi(x))} \frac{p(z|x)q(z;}{\varphi(x))}\,dz \tag{4.12}$$

$$= -\int_{\text{nhật ký } q(z;\varphi(x))} \frac{p(x|z)p(z)p(x)q(}{z;\varphi(x))}\,dz \tag{4.13}$$

$$= \text{nhật ký } p(x) - \int q(z;\varphi(x))\log p(x|z)dz - \int_{\text{nhật ký } q(z;\varphi(x))} \frac{p(z)q(z;}{\varphi(x))}\,dz \tag{4.14}$$

$$= \log p(x) - E_{z\sim q}[\log p(x|z)] + D_{KL}(q(z;\varphi(x))\|p(z)). \tag{4.15}$$

2Tóm lại, p(z) phải là cái gọi là liên hợp trước khả năng p(x|z), do đó p(z|x) sau theo cùng họ phân phối như p(x|z).

Để tìm q (hoặc các tham số của nó φ(x)), chúng ta thu nhỏ số hạng thứ hai và thứ ba ở trên, vì số hạng đầu tiên log p(x) không phải là hàm của q . Nói cách khác,

$$\hat{\varphi}(x) = \arg\min_{\varphi(x)} [I]q\,[\log p(x|z)] + DKLE(q(z;\varphi(x)\|p(z)) \tag{4.16}$$

$$= \arg\text{tối đa}_{\varphi(x)}\ Ez{\sim}q\,[\log p(x|z)] - DKL(q(z;\varphi(x))\|p(z)) \quad \{z\} = J(\varphi(x)) \tag{4.17}$$

Nếu chúng ta thiết kế q(z; φ(x)), thường có thể tính toán độ dốc (ngẫu nhiên) của hàm khách quan J wrt φ(x) này và sử dụng sự đi xuống gradient ngẫu nhiên để cập nhật φ(x) lặp đi lặp lại để tìm q là đại diện tốt hơn cho phân phối thực sự so với lúc đầu.

Trong một bước ngoặt thú vị, hàm mục tiêu J này là giới hạn dưới của log p(x), bởi vì phân kỳ KL lớn hơn hoặc bằng 0:

$$\log p(x;\theta) \geq Ez{\sim}q\,[\log p(x|z;\theta)] - DKL(q(z;\varphi(x))\|p(z)) \quad \{z\} = J(x) \tag{4.18}$$

Điều này có nghĩa là chúng ta có thể gián tiếp tối đa hóa xác suất log được mô hình gán cho quan sát x bằng cách tối đa hóa giới hạn dưới của nó. Tối đa hóa giới hạn dưới không đảm bảo rằng số lượng thực tế tăng lên, nhưng nó đảm bảo rằng số lượng thực tế cao hơn giới hạn dưới tối đa đạt được. Chất lượng của việc làm như vậy được xác định bởi khoảng cách giữa giới hạn dưới và đại lượng thực tế, và khoảng cách này hóa ra chính xác là phân kỳ KL giữa hậu gần đúng và hậu thực sự, DKL(q‖p). Nói cách khác, xấp xỉ iquad với hậu là tốt, chúng ta có giới hạn dưới chặt chẽ hơn và do đó có thể tối đa hóa số lượng mục tiêu thực sự tốt hơn.

Vì cùng một hàm mục tiêu J được sử dụng cho cả việc giảm thiểu phân kỳ KL để tìm ra hậu gần đúng tốt hơn (4.16) và tối đa hóa giới hạn dưới thành đại lượng thực (4.18), chúng ta có thể thực hiện đồng thời cả hai tối ưu hóa [Neal và Hinton, 1998]:

$$\max_{\varphi(x1),\ldots,\varphi(xN),\,\theta} \frac{1}{\text{là}\ n} \sum_{n=1}^{N} Ez{\sim}q(z;\phi(xn))\,[\log p(xn|z;\theta)] - DKL(q(z;\phi(xn))\|p(z)), \tag{4.19}$$

trong đó x1, . . . , xN là các ví dụ đào tạo. Công thức này cũng cho phép chúng ta sử dụng gradient ngẫu nhiên từ §2.3.2. Quy trình này thường được gọi là suy luận và học tập biến thiên ngẫu nhiên. Suy luận đề cập đến ước tínhφ(xn) và học tập đề cập đến ước tính θ.

4.2.1 Mô hình hỗn hợp Gaussian biến thiên

Chúng ta hãy xem xét một trường hợp sử dụng thực tế của suy luận biến thiên ngẫu nhiên và học ở trên. Chúng tôi bắt đầu bằng cách định nghĩa một hỗn hợp Gaussian. Một câu chuyện tổng quát đằng sau một hỗn hợp Gaussian (hoặc tương đương với một mô hình hỗn hợp Gaussian) là

rằng có một số lượng hữu hạn các phân phối Gaussian, được gọi là "thành phần", và một biến tiềm ẩn z chọn một trong các thành phần này. Khi thành phần được chọn, một quan sát x được rút ra từ phân phối Gaussian tương ứng.

Để ánh xạ câu chuyện này vào các hàm xác suất, chúng ta bắt đầu với một phân phối ưu tiên trên các thành phần:

$$p(z) = \frac{1}{\text{triệu}}, \tag{4.20}$$

trong đó M là số thành phần Gaussian. Điều này nói rằng mỗi và mọi thành phần đều có khả năng được chọn như nhau. Điều này có thể được nới lỏng, nhưng chúng tôi sẽ gắn bó với điều này ngay bây giờ. z có thể lấy bất kỳ một trong {1, . . . , M }.

Khi thành phần được chọn, chúng ta vẽ một quan sát x từ

$$p(x|z) = N(x|\mu z, \Sigma z), \tag{4.21}$$

trong đó μz và Σz là giá trị trung bình và hiệp phương sai của thành phần Gaussian thứ z. Để đơn giản, chúng ta hãy giả sử rằng Σz = I, nghĩa là, hiệp phương sai là một ma trận đồng dạng. Trong trường hợp như vậy, chúng tôi nói rằng thành phần là Gaussian hình cầu.

Chúng tôi giới thiệu một hậu gần đúng cho mỗi ví dụ đào tạo xn. Hậu gần đúng này là

$$q(z = k; \varphi n) = \alpha nk, \tag{4.22}$$

trong đó αnz ≥ 0 và PMk=1 αnk = 1 cho tất cả n = 1, . . . , N .

Bây giờ chúng ta có thể viết ra mục tiêu J:

$$J(\alpha 1, \ldots, \alpha N, \mu 1, \ldots, \mu M) = \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha nm - \frac{12}{2\pi} \|xn - \mu m\|2 - d2 \log \tag{4.23}$$

$$+ \sum_{m=1}^{M} \alpha nm \log M - \sum_{m=1}^{M} \alpha nm \log \alpha nm, \tag{4.24}$$

trong đó d = dim(xn).

Hãy tính toán độ dốc của J wrt μk:

$$\nabla \mu k = \frac{1}{N} \sum_{n} (\alpha nk (xn - \mu k)) = \frac{1}{N} \sum_{n} \alpha nk\, xn - \mu k \sum_{n} \text{Châ} = 0 \tag{4.25}$$

$$\Longleftrightarrow \mu k = \sum_{n=1}^{N} \frac{\alpha nk PNn'=}{1\ \alpha n'k} xn. \tag{4.26}$$

Chúng ta có thể tính toán nghiệm chính xác cho μk một cách phân tích, tối đa hóa J.

Hãy làm tương tự cho αnk :

$$\nabla \alpha nk = -\frac{1}{2}\|xn - \mu k\|2 - d2 \log 2\pi - \log M - \log \alpha nk - 1 = 0 \tag{4.27}$$

$$\Longleftrightarrow \log 2 = -\frac{1}{2}\|xn - \mu k\|2 - \frac{d}{2} \log 2\pi - \log M - 1 \tag{4.28}$$

$$\Longleftrightarrow ank = \frac{\exp -\frac{1}{2}\|xn - \mu k\|2 - d2 \log 2\pi - \log M \; PKk'=1}{\exp -\frac{1}{2}\|xn - \mu k'\|2 - d2 \log 2\pi - \log M\,,} \tag{4.29}$$

$$\Longleftrightarrow ank = \frac{\exp -\frac{1}{2}\|xn - \mu k\|2 PKk'=1}{\exp -\frac{1}{2}\|xn - \mu k'\|2\,,} \tag{4.30}$$

bởi vì PKk=1 αnk = 1.

Đối với hậu gần đúng, chúng ta có thể giải nó một cách phân tích và chính xác. Trên thực tế, nếu chúng ta phân tích giải pháp trên cẩn thận hơn, chúng ta nhận ra rằng nó giống hệt với hậu thực sự:

$$\text{nhậ}\atop\text{t}\ \underset{|xn)}{\overset{\{z\}p(z=k}{ANK|}} = \log N\ (xn|\mu k,\ I)\ \underset{|\{z\}=p(xn|z=k;}{+\ \text{nhật}}\ (i)\ \underset{\{z\}=p(}{\overset{1\ \text{triệu}}{|}}\ \underset{z=k)}{} - \text{nhật ký } Z. \tag{4.31}$$

trong đó Z là hằng số chuẩn hóa. Nói cách khác, phân kỳ KL giữa q(z; φ(x)) và p(z|x) bằng không. Nó cũng ngụ ý rằng không có khoảng cách giữa giới hạn dưới biến thiên và log bằng chứng thực p(x).

Các mô hình hỗn hợp Gaussian đặc biệt ở chỗ giới hạn dưới biến thiên chặt chẽ, tức là không có khoảng trống. Chúng cũng đặc biệt ở chỗ chúng ta có thể tìm thấy giải pháp phân tích cho suy luận hậu và tối đa hóa khả năng một cách tương đối đơn giản. Ngay cả trong trường hợp này, người ta nên lưu ý rằng các giải pháp để đặt các gradient này về không là đồng phụ thuộc. Do đó, chúng ta cần cập nhật các đại lượng này lặp đi lặp lại nhiều lần cho đến khi đạt được một số loại hội tụ. Quá trình này được gọi là tối đa hóa kỳ vọng (EM), hay nói chung hơn là thuật toán tọa độ-đi lên. Mỗi bước trong số hai bước này (cập nhật hậu và cập nhật các tham số) được đảm bảo cải thiện giới hạn dưới biến thiên và quy trình xen kẽ này cuối cùng sẽ tìm thấy mức tối đa cục bộ.

Mặc dù có một giải pháp phân tích cho các tham số ở mỗi lần lặp lại E-M, nhưng có thể không mong muốn sử dụng giải pháp phân tích này, bởi vì nó yêu cầu chúng ta sử dụng các phương tiện sau của tất cả các ví dụ đào tạo N. Khi N lớn, bước này, cần được lặp lại, có thể rất tốn kém. Thay vào đó, chúng ta có thể sử dụng gradient ngẫu nhiên bằng cách tính toán các phương tiện hậu của chỉ một tập hợp con nhỏ của tập huấn luyện (có thể được thực hiện chính xác như chúng ta đã suy ra ear-lier) và chỉ cập nhật một chút các tham số theo gradient ngẫu nhiên được tính toán chỉ bằng minibatch này. Mỗi lần lặp lại EM không được đảm bảo sẽ cải thiện giới hạn dưới biến thiên tổng thể, nhưng trung bình với kích thước bước đủ nhỏ, độ dốc ngẫu nhiên sẽ tiến bộ. Đây sẽ là một cách tiếp cận tốt để thực hiện mô hình hỗn hợp Gaussian trên một bộ dữ liệu rất lớn.

Sau khi học xong, chúng ta có thể sử dụng mô hình hỗn hợp Gaussian phù hợp với (a)

Vẽ nhiều mẫu hơn và (b) suy ra sự phân bố hậu trên các thành phần được đưa ra một quan sát mới.

4.2.2K-có nghĩa là phân cụm

Chúng ta hãy giới thiệu nhiệt độ β ≥ 0 dưới dạng siêu tham số trong Phương trình (4.30):

$$\text{ANK} = \frac{\exp - 12\beta \|xn - \mu k\|2PKk'=1}{\exp - 12\beta \|xn - \mu k'\|2 .} \tag{4.32}$$

Khi nhiệt độ cao, tức là β → ∞, phân bố sau gần với phân bố đồng đều hơn. Điều này có thể hiểu được nếu bạn nghĩ về nhiệt động lực học tĩnh. Khi nhiệt độ cao, không có cấu hình cụ thể nào có nhiều khả năng hơn những cấu hình khác, vì tất cả các phân tử đều nảy không ngừng với năng lượng cao. Mặt khác, khi nhiệt độ tiếp cận 0, nhiệt độ sau hội tụ về phía một trong các góc của đơn giản chiều (K − 1), có nghĩa là chỉ có một trong các thành phần có thể xảy ra và tất cả các thành phần khác hoàn toàn không. Đây sẽ là một trường hợp cực đoan khiến chúng tôi quan tâm.

Khi β → 0, chúng ta có thể viết lại lời giải cho suy luận hậu như

$$\text{ANK} = \begin{cases} 1, \text{ nếu } \|xn - \mu k\|2 = \text{chồn } ' = 1,...,K \|xn - \mu k'\|2 \\ 0, \text{ nếu không.} \end{cases} \tag{4.33}$$

Trong trường hợp này, chúng ta có thể tiết kiệm hơn bằng cách lưu trữ

$$\hat{}zn = \arg \max k = 1,... K \text{ ank} , \tag{4.34}$$

thay vì giá trị K cho mỗi ví dụ đào tạo thứ n. Nói cách khác, chúng ta chỉ cần bit ⌈log2 K⌉ trái ngược với bit K × B trong đó B là số bit để biểu thị một giá trị thực trong hệ thống của một người.

Trong trường hợp này, quy tắc cập nhật cho giá trị trung bình của mỗi thành phần trong Phương trình (4.26) cũng có thể được đơn giản hóa:

$$\mu k = \sum_{n=1}^{N} \frac{\alpha nkPNn'=}{1\ \alpha n'k} xn \tag{4.35}$$

$$= \sum_{n=1}^{N} 1(\hat{}zn = k)xn. \tag{4.36}$$

Nghĩa là, chúng tôi thu thập tất cả các ví dụ đào tạo thuộc thành phần thứ k và tính toán trung bình của các ví dụ đào tạo này. Điều này tiếp tục tiết kiệm một lượng điện toán đáng kể, vì chúng ta chỉ đi qua các ví dụ đào tạo N / K trung bình cho mỗi thành phần để tính vectơ trung bình của nó.

Bởi vì chúng ta đang đưa ra lựa chọn khó khăn một cách hiệu quả về thành phần mà mỗi ví dụ đào tạo thuộc về (4.34), chúng ta thường đề cập đến trường hợp đặc biệt này là

tối đa hóa kỳ vọng khó (EM). Hơn nữa, bởi vì chúng ta đang nhóm hiệu quả các ví dụ đào tạo thành các cụm K và mỗi cụm được biểu diễn bằng giá trị trung bình của nó, thuật toán này cũng được gọi là phân cụm K-means. Đây là một trong những thuật toán được sử dụng rộng rãi nhất trong học tập không giám sát và phân tích dữ liệu, và cách tiếp cận dựa trên suy luận biến thiên mà chúng tôi bắt đầu cho phép chúng tôi mở rộng thuật toán này linh hoạt hơn để hoạt động với các phân phối không tầm thường hơn.

4.3 Mô hình biến tiềm ẩn liên tục

Hãy trình bày lại hàm khách quan bắt nguồn từ nguyên tắc suy luận biến thiên trước đó trong Phương trình (4.19):

$$\max_{\varphi(x1),\ldots,\varphi(xN),\,\theta} \frac{1}{la\,n} \sum_{n=1}^{N} Ez{\sim}q(z;\,\phi(xn))\,[\log p(xn|z;\,\theta)] - DKL(q(z;\,\phi(xn))\|p(z)). \tag{4.37}$$

Nhìn vào công thức này, hoàn toàn không có lý do gì để chúng ta giả định rằng z là một biến rời rạc, như chúng ta đã làm với hỗn hợp Gauss ở trên. z có thể là một vectơ có giá trị thực liên tục.

Chúng ta hãy thử một trường hợp đơn giản ở đây bằng cách giả định rằng

$$p(z) = N(z;\,0,\,\sigma 2I|z|) \tag{4.38}$$

$$p(x|z;\,\theta) = N\,(x;\,W\,z + b,\,I|x|), \tag{4.39}$$

trong đó θ = (W, b) với W ∈ R|x|×|z| và b ∈ R|z|. σ2 là một siêu tham số và kiểm soát cường độ của chính quy hóa. Chúng ta sẽ thảo luận thêm về ý nghĩa của điều này. Chúng tôi tiếp tục sử dụng một hậu gần đúng đơn giản cho mỗi ví dụ xn:

$$q(z;\,\varphi(xn)) = N(z;\,\mu n,\,I|z|), \tag{4.40}$$

trong đó φ(xn) = (μn).

Sau đó, mục tiêu cho mỗi ví dụ đào tạo xn trở thành

$$J_n = \text{này} \sim q_n \Big[-\frac{1}{2}\|x_n - W z - b\|^2 - \frac{|x|}{2}\log 2\pi - \frac{1}{2} * K + \|\mu_n\|^2 \text{ giây } 2 \Big] - K + 2K \ln(\sigma) \tag{4.41}$$

$$= -\text{Điều này} \frac{1}{2}\|x_n\|^2 + \frac{1}{2}\|W z + b\|^2 - x_n^T (W z + b) - \frac{1}{2}\sigma^2 \|\mu_n\|^2 \tag{4.42}$$

$$= -\text{Điều này} \frac{1}{2} z^T W^T W z + \frac{1}{2}\|b\|^2 + b^T W z - x_n^T W z - x_n^T b - \frac{1}{2}\sigma^2 \|\mu_n\|^2 + \text{hằng số}. \tag{4.43}$$

$$= -\frac{1}{2}\operatorname{tr}(W \operatorname{E}_{z}[(z-\mu_n)(z-\mu_n)^T] W^T) - \frac{1}{2}\operatorname{tr}(W \mu_n \operatorname{E}_{z}[z]^T W^T) - \frac{1}{2}\operatorname{tr}(W \operatorname{E}_{z}[z] \mu_n^T W^T) + \frac{1}{2}\operatorname{tr}(W \mu_n \mu_n^T W^T) \tag{4.44}$$

$$- \frac{1}{2}\|b\|^2 - b^T W \mu_n + x_n^T W \mu_n + x_n^T b - \frac{1}{2}\sigma^2 \|\mu_n\|^2 + \text{const.} \tag{4.45}$$

$$= -\frac{1}{2}\operatorname{tr}(W W^T) - \frac{1}{2}\mu_n^T W^T W \mu_n - \frac{1}{2}\|b\|^2 - b^T W \mu_n + x_n^T W \mu_n + x_n^T b - \frac{1}{2}\sigma^2 \|\mu_n\|^2 + \text{hằng số}. \tag{4.46}$$

trong đó const. đề cập đến các thuật ngữ không phụ thuộc vào φ(xn) cũng như θ.

Trước tiên, hãy thực hiện suy luận hậu bằng cách tính toán độ dốc của $J = \frac{1}{n}\sum_n J_n$ wrt φ(xn):

$$\nabla \mu_n = -W^T W \mu_n + W^T (x_n - b) - \frac{1}{\sigma^2}\mu_n = 0 \tag{4.47}$$

$$-(W^T W + \sigma^{-2} I)\mu_n + W^T (x_n - b) = 0 \tag{4.48}$$

$$\mu_n = (W^T W + \sigma^{-2} I)^{-1} W^T (x_n - b). \tag{4.49}$$

Cũng giống như với MoG ở trên, chúng ta nhận được một giải pháp phân tích, rõ ràng cho mỗi μn.Bởi vì chúng ta cần tính nghịch đảo của $W^T W + I \in \mathrm{R}^{K \times K}$, điều này có thể hơi tốn kém, nhưng chúng ta cần tính toán nó một lần và sử dụng nó cho tất cả N μn. Chúng ta hãy xem vai trò của σ2 từ p(z) trước đó trong ngữ cảnh này. Khi $\sigma^2 \to \infty$, biểu thức ở trên đơn giản hóa thành

$$\mu_n = \underbrace{(W^T W)^{-1} W^T}_{=\{z} (x_n - b), \tag{4.50}$$

trong đó (a) là nghịch đảo giả của W . Khi W là một ma trận vuông và có thể đảo ngược, điều này tương ứng với W^{-1}. Trong trường hợp đó, chúng ta có thể coi μn là lời giải cho

$$\mu_n = W^{-1}(x_n - b), \tag{4.51}$$

tương đương với

$$x_n = W \mu_n + b. \tag{4.52}$$

Biểu thức này là giá trị trung bình của p(x|z; θ) từ trên.
Nếu không có thông tin trước cho z, tức là σ2 → ∞, dự đoán tốt nhất của chúng ta cấu hình tiềm ẩn nào dẫn đến xn (nghĩa là suy luận sau) là mul-tiply xn (sau khi trừ đi độ lệch b) với nghịch đảo của ma trận chuyển tiếpW . Nói cách khác, kiến thức trước đó mà chúng ta có (trong trường hợp này, rằng các cấu hình tiềm ẩn có nhiều khả năng hơn nếu chúng gần với nguồn gốc hơn) sẽ ảnh hưởng đến suy luận phía sau. Đây là những gì chúng ta muốn nói trước đó về chính quy hóa và σ2kiểm soát cường độ của chính quy hóa.

Bây giờ chúng ta tính toán độ dốc của J wrt W và b cho µn. Hãy bắt đầu với b:

$$\nabla b = \frac{1}{N}\sum_{n=1}^{N}(-b - W\mu n + xn) = 0 \tag{4.53}$$

$$\Longleftrightarrow b = \frac{1}{N}\sum_{n=1}^{N}(W\mu n - xn)\,. \tag{4.54}$$

Biểu hiện này có ý nghĩa trực quan. b, độ sai, là độ lệch trung bình giữa những gì chúng ta nhận được với cấu hình tiềm ẩn và những gì chúng ta thực sự quan sát.

Hãy để chúng tôi tiếp tục với W:

$$\nabla W = \frac{1}{N}\sum_{n=1}^{N} -W - W\,\mu n\mu\mathsf{T}n - b\mu\mathsf{T}n + xn\mu\mathsf{T}n \tag{4.55}$$

$$= -W\left(I + \frac{1}{N}\sum_{n=1}^{N}\mu n\mu\mathsf{T}n\right) + \frac{1}{\text{lần}}\sum_{n=1}^{N}(xn - b)\mu\mathsf{T}n\,. \tag{4.56}$$

Sau đó

$$W = \left(\frac{1}{\text{lần}}\sum_{n=1}^{N}(xn - b)\mu\mathsf{T}n\right)\left(I + \frac{1}{N}\sum_{n=1}^{N}\mu n\mu\mathsf{T}n\right)^{-1}\,. \tag{4.57}$$

Thuật ngữ đầu tiên trong sản phẩm ở phía bên phải có thể được coi là sự hoàn thiện cái gọi là quy tắc học tập của Hebbian: "các tế bào thần kinh hoạt động cùng nhau, kết nối với nhau" [Hebb, 1949]. Nếu chiều thứ i của quan sát xi cháy (nghĩa là vượt quá độ thiên vị bi) và chiều thứ j của biến tiềm ẩn µj cùng nhau hoạt động (trong đó 'lửa' được định nghĩa là bất kỳ lệch nào ra khỏi độ lệch hoặc không), thì cường độ của giá trị trọng lượng wij giữa chúng phải lớn. Điều này đã hiển thị là số hạng thứ hai trong gradient wrt W ở trên.

Số hạng thứ hai ở phía bên phải (tương ứng với số hạng đầu tiên trong gradient) phức tạp hơn. Điều này hoạt động như làm trắng µn bên trong thuật ngữ đầu tiên. Nghĩa là, nó làm cho µn được phân phối sao cho hiệp phương sai gần với đồng dạng. Điều này hoạt động như làm cho W nắm bắt được hiệp phương sai giữa các quan sát được trừ theo chủ đề (xn − b) và các cấu hình tiềm ẩn được làm trắngµn I + 1N PNn = 1 µnµTn −1. Bằng cách đó, trong lần lặp tiếp theo của EM này

thủ tục, μn sẽ được phân phối sao cho hiệp phương sai tập thể của chúng sẽ gần hơn với đồng dạng, đó là những gì chúng tôi áp đặt bằng cách nói rằng tiền trị trên biến tiềm ẩn phải là một phân phối Gaussian hình cầu. Nói cách khác, đây cũng là ảnh hưởng của chính quy hóa do phân phối trước.

Bằng cách xoay vòng qua các bước cập nhật μn, W và b này, giới hạn thấp hơn biến thiên sẽ cải thiện dần dần cho đến khi hội tụ. Trong giới hạn σ2 → ∞và với các ràng buộc rằng W là trực giao, tức là W W ⊤ = I và thatb = 1N PNn = 1 xn, chúng tôi khôi phục phân tích thành phần chính [Hotelling, 1933]. Dẫn xuất của chúng tôi ở đây là một trường hợp đặc biệt của một phiên bản tổng quát hơn được gọi là phân tích thành phần chính xác suất-tic [Tipping and Bishop, 1999]. Đặc biệt, chúng tôi tuân theo cách tiếp cận suy luận biến thiên [Ilin và Raiko, 2010].

Cách tiếp cận dựa trên giới hạn dưới biến thiên này một lần nữa cho phép chúng tôi sử dụng gradient stochas-tic có khả năng mở rộng hơn nhiều so với quy trình EM chính xác. Tại mỗi lần lặp, chúng tôi chọn một loạt các ví dụ đào tạo, suy ra các phương tiện hậu (gần đúng) và sử dụng chúng để tính toán độ dốc của biến thiên bên dưới wrt W và b. Thay vì tính toán các giá trị tối ưu cho minibatch này, chúng tôi chỉ cần cập nhật chúng một chút theo hướng gradient ngẫu nhiên.

4.3.1Bộ mã hóa tự động biến thể

Một câu hỏi tự nhiên, dựa trên những gì chúng ta đã thấy trong §2.2.2, là liệu chúng ta có thể sử dụng một phép biến đổi phi tuyến tính cho p(x|z) thay vì phép biến đổi tuyến tính trong Eq hay không. (4.38). Điều này hoàn toàn có thể thực hiện được với

$$p(x|z; \theta) = N (x|F(z; \theta), l|x|), \tag{4.58}$$

trong đó F là một hàm phi tuyến tùy ý, được tham số hóa bởi θ, ánh xạ z tox và có thể vi phân được wrt θ. Nói cách khác, chúng ta ổn với bất kỳ loại tham số hóa nào miễn là chúng ta có thể tính toán[3]

$$\text{JacθF}(z; \theta) = \frac{\partial F}{\partial \theta}(z; \theta). \tag{4.59}$$

Sự thay đổi nhỏ này có một hậu quả lớn về sức mạnh mô hình hóa của mô hình biến tiềm ẩn liên tục. Điều này là do các tính chất đặc biệt (và đáng kinh ngạc) của các phân phối chuẩn. Chúng ta hãy xem lại trường hợp tuyến tính ở trên (4.38):

$$p(z) = N(z; 0, \sigma 2l|z|) \tag{4.60}$$

$$p(x|z; \theta) = N (x; W z + b, l|x|), \tag{4.61}$$

[3]Tôi sẽ sử dụng $\frac{\partial}{\partial x}$ ký hiệu để chỉ ma trận Jacobian trừ khi nó gây nhầm lẫn.

Sau đó, xác suất chung có thể được viết ra dưới dạng

$$\log p(x, z; \theta) = \log p(z) + \log p(x|z; \theta) = -\frac{1}{2\sigma^2}\|z\|^2 - \frac{1}{2}\|x - Wz - b\|^2 + \text{const.} \tag{4.62}$$

$$= -\frac{1}{2} z^\top(\sigma^2 I)z + (x - b)^\top I(x - b) + z^\top W^\top W z - (x - b)^\top W z - z W^\top (x - b) + \text{hằng số.} \tag{4.63}$$

$$= -\frac{1}{2} (x - b)^\top I(x - b) + z^\top(W^\top W + \sigma^2 I)z - (x - b)^\top W z - z^\top W^\top z + \text{hằng số.} \tag{4.64}$$

Cho $v = [x, z]^\top$ và $\mu = [b, 0|z|]^\top$. Sau đó,

$$\log p(x, z; \theta) - \log Z(\theta) = -\frac{1}{2}(v - \mu)^\top \begin{bmatrix} \text{Tôi} & -W \\ -W^\top & W^\top W + \sigma^2 I \end{bmatrix} (v - \mu) + \text{định vị.} \tag{4.65}$$

Điều này cho thấy sự phân bố chung trên [x, z] cũng là Gaussian với meanμ. Mặc dù chúng ta chỉ cần chỉ ra điều này cho lập luận tiếp theo của chúng ta, chúng ta cũng hãy kiểm tra ma trận hiệp phương sai của phân phối chung.

Có một công thức kỳ diệu được gọi là lemma đảo ngược ma trận khối:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}. \tag{4.66}$$

Chúng ta có thể sử dụng điều này để viết ra hiệp phương sai của phân phối chung p(x, z; θ):

$$S = \begin{bmatrix} I + W(W^\top W + \sigma^2 I - WW^\top)^{-1}W^\top & W(W^\top W + \sigma^2 I - WW^\top)^{-1} \\ (W^\top W + \sigma^2 I - WW^\top)^{-1}W^\top & (W^\top W + \sigma^2 I - WW^\top)^{-1} \end{bmatrix} \tag{4.67}$$

$$= \begin{bmatrix} I + \sigma^2 W W^\top & \sigma^2 W \\ \sigma^2 W^\top & \sigma^2 I \end{bmatrix}. \tag{4.68}$$

Bởi vì phân phối cận biên trên bất kỳ tập hợp con của các chiều của một biến ngẫu nhiên chuẩn cũng là bình thường, chúng ta biết rằng phân phối cận biên trên x, tức là $p(x) = \int p(x|z)p(z)dz$, cũng là bình thường. Nói cách khác, mối quan hệ tuyến tính giữa x và z trong phân tích thành phần chính xác suất (PCA) ở trên chỉ có thể đại diện cho phân phối Gaussian trên x. Đây là một hạn chế quan trọng. Một giới hạn như vậy không còn tồn tại nữa nếu chúng ta sử dụng một hàm phi tuyến để mô hình hóa mối quan hệ giữa x và z. Trong trường hợp đó, phân phối chungp(x, z) sẽ không phải là Gauss nói chung, bởi vì cấu trúc hiệp phương sai sẽ

không đứng yên mà thay đổi động tùy thuộc vào X và Z. Chúng ta muốn có một hỗn hợp Gaussian với vô số thành phần:

$$p(x) = \overset{\text{Với}}{p(x|z)p(z)dz} = \overset{\text{Với}}{p(z)N}(x|F(z;\theta), I)dz. \tag{4.69}$$

Hãy xem xét giới hạn dưới biến thiên với công thức phi tuyến này cho một ví dụ cụ thể xn:

$$\text{Jn} = \text{này}\sim\text{qn}\ -\frac{1}{2}\|\text{xn} - F(z;\theta)\|2 - |x|2\log 2\pi - \frac{1}{2}\,\underline{K + \quad\quad} - K + 2K\ln(\sigma)\ \square.$$

(4.70)

Gradient của Jn wrt μn sau đó là

$$\nabla\mu\text{n} = -\frac{Z\exp -\frac{1}{2}\|z-\mu\text{n}\|2}{(2p)|z|/2}\,\frac{1}{2}(z-\mu\text{n})\|\text{xn} - F(z;\theta)\|2 - \frac{}{\mu\text{n}\sigma 2} \tag{4.71}$$

$$= -\frac{1}{2}\,\text{Ez}\,(z-\mu\text{n})\|\text{xn} - F(z;\theta)\|2 - \mu\underline{\text{n}}\sigma 2 \tag{4.72}$$

$$= -\frac{1}{2}\,2\text{xnEz}\,[zF(z;\theta)] - 2\text{xn}\mu\text{nEz}\,[F(z;\theta)] + \text{Ez}\,z\|F(z;\theta)\|2 - \mu\text{nEz}\,\|F(z;\theta)\|2 - \mu\text{n}\sigma\underline{2}\,. \tag{4.73}$$

Rõ ràng là nếu không biết dạng của F, thì không thể đưa ra một giải pháp phân tích cho μn nói chung. Tệ hơn nữa, không rõ làm thế nào để tính toán gradient một cách phân tích, do những kỳ vọng đầy thách thức phải được tính toán. Tuy nhiên, chúng ta có thể sử dụng xấp xỉ Monte Carlo dựa trên mẫu, vì chúng ta có thể chọn q sau gần đúng để có thể dễ dàng khuếch đại:

$$\nabla\mu\text{n} \approx \tilde{} \nabla\mu\text{n} = -\frac{1}{2}\,(\tilde{}z - \mu\text{n})\|\text{xn} - F(\tilde{}z;\theta)\|2 - \mu\text{n}\sigma 2\,. \tag{4.74}$$

Trong trường hợp cụ thể này của hậu Gaussian, chúng ta có thể vẽ một mẫu bằng cách sử dụng thủ thuật tham quan hóa:[4]

$$\tilde{}z = \mu\text{n} + \text{trong}, \tag{4.76}$$

trong đó $\varepsilon \sim N(0, I|z|)$. Cắm cái này vào, chúng ta nhận được

$$\tilde{}\nabla\mu\text{n} = -\frac{\sigma 2}{}\,\text{e}\|\text{xn} - F(\mu\text{n} + \text{in? }\theta)\|2 - \mu\text{n}\sigma 2\,. \tag{4.77}$$

4Thủ thuật so sánh đề cập đến việc xây dựng quá trình lấy mẫu từ một phân phối phân bố dạng nhiễu biến đổi phi tuyến tính và xác định được rút ra từ một phân phối khác:

$$z = g(\varepsilon;\varphi), \qquad \varepsilon \sim p(\varepsilon). \tag{4.75}$$

Điều này cho phép chúng ta tính toán đạo hàm của mẫu z wrt các tham số của hàm định nghĩa này, tức là $\partial g\partial\varphi$ (z). Đây là một thủ thuật tiện dụng, vì lấy mẫu thường được coi là toán tử không phân biệt. Mặc dù hữu ích của nó, nhưng không phải lúc nào cũng có thể đưa ra sự khắc phục như vậy.

Sau đó, chúng ta có thể sử dụng ước tính gradient ngẫu nhiên này để tìm nghiệm cho μn. Tuy nhiên, nhìn vào gradient ở trên, chúng ta có thể thấy hướng gradient này chỉ ra điều gì. Đặc biệt, thuật ngữ đầu tiên xem xét các hướng tương tự như μn trung bình nhưng có một số nhiễu. Sau đó, chúng ta cân từng hướng như vậy (hoặc sự khác biệt giữa hướng này và ước tính hiện tại của μn) theo chất lượng của chúng, trong đó chất lượng được định nghĩa là mức độ tương tự của quan sát được giải mã, F (z; θ), với quan sát thực tế xn (lưu ý dấu âm biến khoảng cách này thành chất lượng.) Nói cách khác, chúng tôi tìm kiếm sự thay đổi thành μn làm cho quan sát được giải mã gần hơn với quan sát thực tế. Điều này hoàn toàn hợp lý, từ quan điểm của p(y|z). Thuật ngữ thứ hai chỉ đơn giản mang μn về phía nguồn gốc với tỷ lệ nghịch với phương sai trước tỷ lệ nghịch với cường độ chính quy hóa.

Việc thiếu một giải pháp phân tích cho μn là có vấn đề, bởi vì chúng ta phải giữ μn cho tất cả các ví dụ đào tạo N trong các lần lặp lại EM, ngay cả với sự xuống dốc stochas-tic. Tại mỗi lần lặp, chúng tôi sẽ chọn một số lượng nhỏ M các ví dụ đào tạo, truy xuất các quan sát liên quan {xm}Mm=1 cũng như ước tính hiện tại liên quan của các phương tiện sau {μm}Mm=1, cập nhật các phương tiện vị trước một chút theo hướng gradient, cập nhật các tham số theo hướng gradient ngẫu nhiên và cuối cùng lưu trữ lại ước tính cập nhật của các phương tiện sau. Điều này không gây quá nhiều áp lực cho tính toán nhưng nó gây áp lực rất lớn lên bộ lưu trữ và I/O, vì chúng ta cần các bit O(b × |z| × N) để lưu trữ các phương tiện sau.

Suy luận khấu hao. Thay vì lưu trữ các phương tiện sau gần đúng cho tất cả các ví dụ đào tạo, chúng ta có thể nén chúng thành một mạng lưới thần kinh sâu mạnh mẽ. Let G : X → R|z| là một mạng suy luận, vì chúng ta sẽ yêu cầu G suy ra gần hơn biến tiềm ẩn cho một đầu vào x. This G cũng được tham số hóa bởi bộ tham số θG của riêng nó. Mạng suy luận này hoạt động hiệu quả như một phiên bản nén của bảng chứa {μm}Mm=1, vì chúng ta có thể truy xuất μm bằng cách

$$\mu m = G(xm; \theta G). \tag{4.78}$$

Trên thực tế, mạng suy luận này thậm chí còn cho phép chúng ta truy xuất một phân phối poste-rior gần đúng với một đầu vào mới x′ /∈ D nhờ khả năng khái quát hóa của nó.

Bây giờ chúng ta hãy cắm mạng suy luận G này vào hàm mục tiêu mỗi phiên bản từ Phương trình (4.70):

$$Jn = Ez{\sim}qn(z;\, G(xn;\, \theta G),\, tr2)\, \frac{-\,12}{2\pi} \|xn - F\,(z;\, \theta)\|2 - |x|2 \log \tag{4.79}$$

$$\frac{-\,12\,K + \|G(xn;}{\theta G)\|2\sigma 2} - K + 2K \ln\,(\sigma)\ \square. \tag{4.80}$$

Vì kỳ vọng rất khó đánh giá, chúng tôi sẽ xem xét một-

dự toán mẫu của Jn:

$$\tilde{J}n = -\frac{1}{2} \|xn - F(G(xn; \theta G) + in; \theta)\|2 - \frac{1}{2}\sigma 2 \|G(xn; \theta G)\|2 + \text{hằng số}, \quad (4.81)$$

trong đó ε ∼ N (0, I).

Có hai thuật ngữ không hằng số trong mục tiêu gần đúng này. Thuật ngữ đầu tiên là lỗi tái thiết. Đầu vào xn được xử lý bởi mạng suy luận G trước, và sau đó phiên bản nhiễu của đầu ra của G sau đó được xử lý bởi F để xây dựng lại đầu vào. Mục tiêu được tối đa hóa khi sự khác biệt giữa đầu vào ban đầu và đầu vào được tái tạo được giảm thiểu (xem thenegation trước định mức L2.) Quá trình này thường được gọi là tự động mã hóa và đây là lý do tại sao toàn bộ khung này được gọi là bộ mã hóa tự động biến thiên [Kingmaand Welling, 2013].

Thuật ngữ thứ hai là một bộ chính quy đẩy định mức L2 của đầu ra từ mạng suy luận trở nên nhỏ. Điều này đảm bảo rằng tất cả các đầu vào {x1, . . . , xM } được ánh xạ đến không gian tiềm tờ, tức là không gian của biến tiềm ẩn z, càng chặt chẽ càng tốt. Nếu không có thuật ngữ này, định mức đầu ra của G có thể phát triển vô thời hạn, đẩy các hậu suy luận của tất cả các đầu vào càng xa càng tốt, vì điều này sẽ đảm bảo rằng F có thể tái tạo đầu vào ban đầu một cách hoàn hảo ngay cả với nhiễu được tiêm. Tuy nhiên, điều này sẽ khiến F không thể đối phó với bất kỳ z nào được lấy mẫu từ trước đó hoặc nằm giữa bất kỳ phân phối sau suy luận của bất kỳ cặp đầu vào nào, dẫn đến một mô hình tổng quát tồi tệ.

Nhờ thủ thuật tái chuẩn hóa, chúng ta có thể tính toán độ dốc của $\tilde{J}$nw.r.t. tất cả các tham số, bao gồm cả F và G. Nói cách khác, chúng ta có thể sử dụng lan truyền ngược để đào tạo cả mạng suy luận và thế hệ, G và F, tương ứng. Điều này cho phép chúng tôi đào tạo mạng suy luận cực kỳ hiệu quả mà không cần phải duy trì toàn bộ cơ sở dữ liệu về các tham số sau gần nhất của phiên bản cụ thể. Hơn nữa, như đã thảo luận trước đó, mạng suy luận này có thể được sử dụng với một đầu vào mới, làm cho nó hữu ích để phân tích một tập hợp các đầu vào không có trong quá trình đào tạo. Kết hợp sau gần đúng được tính toán bởi mạng suy luận có thể được tinh chỉnh thêm bằng cách sử dụng độ dốc để phù hợp với hậu thực tốt hơn [Hjelm và cộng sự, 2016].

Có lẽ quan trọng hơn, điều này ngụ ý rằng việc học từ đầu đến cuối như vậy của mạng lưới suy luận và thế hệ là có thể thực hiện được với sự lan truyền ngược và descent stochasticgradient miễn là chúng ta có thể sử dụng thủ thuật so sánh để lấy mẫu từ một hậu gần đúng mà không phá vỡ khả năng vi sai. Điều này mở ra một cánh cửa mở rộng cho một loạt các cơ hội hoàn toàn mới để mở rộng các mod-el xác suất khác nhau mà trước đây rất rườm rà để rút ra và sử dụng, mặc dù những điều này nằm ngoài phạm vi của khóa học này.

4.3.2 Lấy mẫu tầm quan trọng và phương sai của nó.

Trước khi kết thúc phần này, chúng ta hãy nghĩ về cách chúng ta nên tính xác suất cận biên log của một quan sát x từ phương trình (4.69):

$$p(x) = Ez{\sim}p(z) [p(x|z)] . \quad (4.82)$$

Không giống như trong thời gian đào tạo, chúng ta ít chịu áp lực về thời gian, và do đó cách tiếp cận không tự nhiên sẽ là một xấp xỉ Monte-Carlo ngây thơ:

$$p(x) \approx 1\underline{M} \overset{M}{\underset{m=1}{X}} p(x|zm), \tag{4.83}$$

trong đó zm ∼ pz(z).
Thật không may, cách tiếp cận ngây thơ này có thể có một sự khác biệt lớn.
Để ngắn gọn, letf (z) = p(x|z) và p(z) = pz(z). Bởi vì chúng ta đã biết rằng nó không thiên vị, sau đó chúng ta có thể viết phương sai như

$$V \left[\frac{1}{M} \overset{M}{\underset{m=1}{X}} F(zm) \right] = \frac{1}{M^2} V \left[\overset{M}{\underset{m=1}{X}} F(zm) \right] = \frac{1}{\text{phút } 2} \overset{M}{\underset{m=1}{X}} V[f(zm)] = \frac{1}{M^2} M V[f(z)] = \frac{V[f(z)]}{M}, \tag{4.84}$$

Bởi vì ZM được phân phối giống hệt nhau theo p(x).
Hóa ra chúng ta có thể giảm phương sai này bằng cách tránh lấy mẫu từ pztrực tiếp nhưng từ một phân phối khác qz. Kỹ thuật này được gọi là lấy mẫu tầm quan trọng:

$$I'm\ pz[f(z)] = I'm / qz \left[\frac{pz(z) qz(z)}{f(z)} \right] \approx \frac{1}{M} \overset{M}{\underset{m=1}{X}} \frac{PZ(ZM) QZ(ZM)}{F(ZM)}. \tag{4.85}$$

Sau đó, chúng ta có thể kiểm soát phương sai của công cụ ước tính này bằng cách chọn qz một cách cẩn thận. Để hiểu cách chúng ta có thể chọn qz một cách cẩn thận, hãy xem xét phương sai của công cụ ước lượng này:

$$V \left[\frac{1}{M} \overset{M}{\underset{m=1}{X}} \frac{PZ(ZM) QZ(ZM)}{F(ZM)} \right] = \frac{1}{M} V \left[\frac{pz(zm)}{(zm) f(zm)} qz \right]. \tag{4.86}$$

Hãy cắm lại p(x|z) và pz(z):

$$\frac{1}{\text{triệu}} \frac{PZ(ZM) QZ(ZM)}{F(ZM)} = \frac{1}{M} V \left[\frac{p(x|z) p(z)}{(z) q(z)} \right]. \tag{4.87}$$

Bằng cách sử dụng V[X] = E[X2] − E[X]2, chúng ta nhận được

$$V \left[\frac{p(x|z) p(z)}{Q(Z)} \right] = Z \frac{p(x|z)^2 pz(z|)^2}{Q(Z)} dz - \text{định vị}. \tag{4.88}$$

Thuật ngữ thứ hai là hằng số wrt q, bởi vì đó không là gì ngoài đại lượng ban đầu mà chúng ta đang cố gắng ước tính.

Hãy nhớ lại định nghĩa sau đây về bất đẳng thức Cauchy-Schwarz:

$$|\langle , v\rangle|^2 \le \langle , \text{ rằng } \rangle \langle v, v\rangle, \tag{4.89}$$

trong đó ⟨···⟩ là một sản phẩm bên trong khái quát hóa một sản phẩm chấm tiêu chuẩn. Chúng ta có thể định nghĩa một tích bên trong trên các hàm tích phân bình phương5 như

$$\langle f, g\rangle = \int f(x)\, g(x)\, dx \tag{4.91}$$

trên miền của X. Sau đó, chúng ta có thể viết bất đẳng thức Cauchy-Schwarz là

$$\int f(x)\, g(x)\, dx \le \sqrt{\int f^2(x)\, dx \int g^2(x)\, dx}. \tag{4.92}$$

Bởi vì $\int q(z)dz = 1$ theo định nghĩa, chúng ta quan sát thấy rằng

$$\left[\int \frac{p(x|z)p(z)}{\sqrt{Q(z)}} dz\right]^2 = \left[\int \sqrt{Q(z)dz} \frac{p(x|z)p(z)}{\sqrt{Q(z)}}\right]^2 \le \int \sqrt{Q(z)dz}^2 \int \left[\frac{p(x|z)p(z)}{\sqrt{Q(z)}}\right]^2 = \int p(x|z)p(z)dz. \tag{4.93}$$

Xem xét cả hai bên một cách cẩn thận, chúng ta thấy rằng họ bình đẳng khi

$$Cq(z) = p(x|z)p(z). \tag{4.94}$$

Điều này có thể dễ dàng kiểm tra bằng cách cắm nó vào phía bên trái của bất đẳng thức ở trên:

$$\left[\int \frac{Cq(z)}{\sqrt{q(z)}} dz\right]^2 = \left[\int \sqrt{Q(z)dz}\right]^2 = C2, \tag{4.95}$$

và sau đó vào phía bên phải của bất bình đẳng:

$$\int \sqrt{Cq(z)dz}^2 = C2. \tag{4.96}$$

Bởi vì $\int q(z)dz = 1$,

$$C = \int p(x|z)p(z)dz. \tag{4.97}$$

Đặt tất cả chúng lại với nhau, chúng ta nhận được câu hỏi tối ưu sau:

$$q*(z) = \frac{p(x|z)p(z)R}{p(x|z')p(z')dz'}, \tag{4.98}$$

5A f có thể tích phân bình phương khi

$$\int f(x)dx < \infty. \tag{4.90}$$

hóa ra chính xác là phân phối hậu trên z cho x. Nói cách khác, nếu chúng ta lấy mẫu từ phân phối hậu thay vì phân phối trước đó và cân lại p(x|z) theo tỷ lệ của chúngp(z)p(z|x), xấp xỉ của chúng ta vừa không thiên vị vừa có phương sai tối thiểu.

Tuy nhiên, đây không phải là con đường đúng đắn về phía trước, vì xác suất hậu có mẫu số riêng của nó là tích phân khó giải quyết. Thay vào đó, điều này nói rằng cái gọi là phân phối đề xuất q phải gần với phân phối hậu thực sựp(z|x), trên thực tế chính xác là tiêu chí mà chúng ta đã sử dụng để suy ra giới hạn thấp hơn biến thiên trước đó trong §4.2. Khi giới hạn dưới biến đổi, đóng vai trò là hàm khách quan cho các mô hình biến tiềm ẩn, được tối đa hóa, sự phân kỳ KL giữa q (hậu gần đúng) và p (hậu thực sự) thu hẹp. Nói cách khác, chúng ta có thể chỉ cần sử dụng mạng suy luận q được đào tạo làm phân phối đề xuất để xấp xỉ xác suất cận biên log của một quan sát sau khi đào tạo để có được một ước lượng không thiên vị, phương sai thấp của đại lượng.6 Hóa ra việc tối đa hóa suy luận biến thiên có một lợi thế khác.

6Giới hạn dưới biến thiên cũng có thể được sử dụng như một đại diện cho xác suất cận biên log. Đây thực sự là một thực hành chuẩn mực trong quá trình đào tạo, để theo dõi tiến độ học tập. Tuy nhiên, đại lượng này là một ước tính sai lệch về xác suất cận biên log, và điều quan trọng là sử dụng lấy mẫu tầm quan trọng để kiểm tra xác suất biên-log thực sự.

Chương 5

Mô hình tạo không định hướng

Chúng tôi đã nghiên cứu một số cách tiếp cận khác nhau để mô hình hóa tổng quát trong chương trước. Những cách tiếp cận này có thể được coi là phù hợp với một mô hình đồ thị có hướng trong đó có hai biến, quan sát x và z. Trong mô hình này, chúng tôi đã xác định hai phân phối tương đối đơn giản, hoặc có lẽ chính xác hơn là dễ mô tả một cách tương đối, p(z) và p(x|z) nhưng có thể mô hình hóa phân bố phức tạp qua quan sát bằng quá trình gạt ra bên lề, p(x) = R p(z)p(x|z)dz. Bây giờ, chúng ta phải hỏi liệu có cách nào khác để làm điều tương tự không.

5.1 Máy Boltzmann bị hạn chế: Sản phẩm của các chuyên gia

Chúng ta bắt đầu với một ý tưởng khá cũ được gọi là máy Boltzmann hạn chế [RBM; Smolensky, 1986]. RBM xác định biểu đồ lưỡng phần với các cạnh không định hướng giữa hai nhóm; x và z. Mỗi phân vùng bao gồm các kích thước củaquan sát x hoặc z tiềm ẩn . Các phân vùng này được kết nối hoàn toàn với nhau, nhưng không có cạnh nào trong mỗi phân vùng. Mỗi cạnh có một giá trị trọng số, dẫn đến một ma trận W ∈ R|x|×|z| . Mỗi nút cũng có thiên vị vô hướng riêng, dẫn đến hai vectơ b ∈ R|x| và c ∈ R|z|. Sau đó, chúng ta định nghĩa một hàm năng lượng là

$$e(x, z, \theta = (W, b, c)) = -x^{\top}W\,c - x^{\top}b - z^{\top}c \tag{5.1}$$

$$= -\sum_{i=1}^{|x|}\sum_{j=1}^{|với|} wijxizj - \sum_{i=1}^{|x|} xibi - \sum_{j=1}^{|với|} zjcj. \tag{5.2}$$

Mặc dù nó không cần thiết cho x, nhưng chúng ta giới hạn z là một vectơ nhị phân: $z \in \{0, 1\}^{|z|}$.

Như chúng ta đã làm đi làm lại cho đến nay, chúng ta có thể biến hàm năng lượng này thành hàm xác suất chung:

$$\log p(x, z; \theta) = -e(x, z, \theta) - \log \text{Với}_{x' \in X} \quad X_{z' \in \{0,1\}^{|z|}} \quad \exp(-e(x', z', \theta))\, dx', \tag{5.3}$$

trong đó X là tập hợp tất cả các giá trị có thể được x có thể nhận. Nếu X là một tập hợp hữu hạn, chúng ta thay thế R bằng P.

Chúng ta hãy tập trung vào hằng số chuẩn hóa, $P_{z' \in \{0,1\}^{|z|}} \exp(-e(x, z', \theta))$. Từ

$$\exp(a + b) = \exp(a) \exp(b), \tag{5.4}$$

chúng ta có thể viết lại nó thành

$$\exp(-e(x, z, \theta)) = \prod_{i=1}^{|x|} \text{Và} \prod_{j=1}^{\text{với}|} \text{Và} \text{EXP}(W_{ij} x_i z_j) \prod_{i=1}^{|x|} \text{Và} \text{EXP}(x_i b_i) \prod_{j=1}^{\text{với}|} \text{Và} \text{EXP}(z_j c_j) \tag{5.5}$$

$$= \prod_{i=1}^{|x|} \text{Và} \text{EXP}(x_i b_i) \prod_{j=1}^{\text{với}|} \text{Và} \exp(w_{ij} x_i z_j + z_j c_j). \tag{5.6}$$

Bây giờ, tôi muốn gạt z ra khỏi cách diễn đạt này. Trong hầu hết các trường hợp, điều này sẽ khó giải quyết, vì có $2^{|z|}$ các giá trị có thể z có thể thực hiện. Tuy nhiên, cấu trúc lưỡng bên này hóa ra là một phước lành mà chúng ta có thể dựa vào.

Hãy xem xét trường hợp đơn giản sau:

$$\sum_{z \in \{0,1\}^2} \prod_{j=1}^{2} \text{Và} f_j(z_j) = f_1(0) f_2(0) + f_1(0) f_2(1) + f_1(1) f_2(0) + f_1(1) f_2(1) \tag{5.7}$$

$$= f_1(0)(f_2(0) + f_2(1)) + f_1(1)(f_2(0) + f_2(1)) \tag{5.8}$$

$$= (f_1(0) + f_1(1))(f_2(0) + f_2(1)) \tag{5.9}$$

$$= \prod_{j=1}^{2} \text{Và}(f_j(0) + f_j(1)). \tag{5.10}$$

Thay vì tổng nhiều số hạng theo cấp số nhân, chúng ta có thể nhân |z| Chỉ điều khoản:

$$\sum_{z\in\{0,1\}^{|z|}} \exp(-e(x, z, \theta)) = \sum_{z\in\{0,1\}^{|z|}} \prod_{i=1}^{|x|} \exp(x_i b_i) \prod_{j=1}^{|\text{với}|} \exp(w_{ij} x_i z_j + z_j c_j) \tag{5.11}$$

$$= \prod_{i=1}^{|x|} \exp(x_i b_i) \sum_{z\in\{0,1\}^{|z|}} \prod_{j=1}^{|\text{với}|} \exp(w_{ij} x_i z_j + z_j c_j) \tag{5.12}$$

$$\overset{\text{kinh nghiệm}}{=} \prod_{i=1}^{|x|} \exp(x_i b_i) \prod_{j=1}^{|\text{với}|} (1 + \exp(w_{ij} x_i + c_j)) \tag{5.13}$$

$$\tag{5.14}$$

Bạn có thể nghĩ về phía bên trái của đạo hàm này là hàm xác suất không chuẩn hóa $\tilde{p}(x; \theta)$ của x, vì hằng số chuẩn hóa của $p(x, z; \theta)$ không phải là hàm của x cũng như z. Trong trường hợp đó, chúng ta có thể viết nó ra như

$$\tilde{p}(x; \theta) \propto \varphi_0(x) \prod_{j=1}^{|\text{với}|} \varphi_j(x), \tag{5.15}$$

đâu

$$\log \varphi_0(x) = x^\top b, \tag{5.16}$$

$$\log \varphi_j(x) = \log(1 + \exp(w_{\cdot j}^\top x + c_j)). \tag{5.17}$$

Chúng tôi gọi mỗi φ_k là một chuyên gia, và đây là một công thức điển hình của một sản phẩm củacác chuyên gia [PoE; Hinton, 2002].

PoE không giống như hỗn hợp các chuyên gia (MoE), chẳng hạn như hỗn hợp Gaus-sians từ §4.2.1. MoE có một lợi thế đáng kể so với PoE ở chỗ chúng dễ dàng được chuẩn hóa miễn là mỗi chuyên gia đều được chuẩn hóa tốt. Tuy nhiên, PoE có thể mô hình hóa một phân phối sắc nét hơn nhiều, không giống như của MoE. Entropy của aMoE luôn được giới hạn thấp hơn bởi entropy của một thành phần riêng lẻ. Đây không phải là trường hợp của PoE, bởi vì điểm số từ các chuyên gia được nhân lên thay vì tính trung bình. Bất kỳ chuyên gia nào cũng có thể đơn giản phủ quyết bằng cách đưa ra một giá trị gần bằng 0, trong khi điều này sẽ không ảnh hưởng đến kết quả tổng thể trong trường hợp của MoE.

Chúng tôi sử dụng mục tiêu khả năng log, tính trung bình trên toàn bộ tập huấn luyện, để đào tạo RBM này:

$$\text{Lll}(x, \theta) = e(x, \theta) + \log \int_{\text{Với}_i} \exp(-e(x', \theta))\, dx'. \tag{5.18}$$

Cũng giống như trước đó, chúng ta sử dụng độ dốc ngẫu nhiên và để làm như vậy, chúng ta cần có khả năng tính toán độ dốc của sự mất mát trên mỗi ví dụ này với năng lượng e.

Khi chúng ta có thể tính toán nó, chúng ta có thể sử dụng quy tắc chuỗi của các đạo hàm để tính toán gradient w.r.t. mỗi tham số. Như vậy

$$\nabla_\theta L_{II} = \nabla_\theta e(x, \theta) - \sum_{\text{Với } i} \frac{\exp(-e(x', \theta))}{\exp(-e(x'', \theta))dx''} \nabla_\theta e(x', \theta)dx' \quad (5.19)$$

$$= \underbrace{\nabla_\theta e(x, \theta)}_{=\{z\}} - \underbrace{E_{x'; \theta} \nabla_\theta e(x', \theta)}_{=\{z\}} . \quad (5.20)$$

Có hai thuật ngữ trong gradient này. Thuật ngữ đầu tiên (a) được gọi là giai đoạn tích cực, vì nó chủ động làm giảm (nhớ lại rằng chúng ta đang đi theo hướng tiêu cực) năng lượng của ví dụ tích cực, trong đó ví dụ tích cực đề cập đến một trong các ví dụ đào tạo x từ tập huấn luyện. Thuật ngữ thứ hai (b) được gọi là pha âm, trong đó nó chủ động làm tăng năng lượng của acấu hình x′ rất có thể xảy ra theo mô hình hiện tại, tức là p(x′; θ) ↑.Đây chính xác là những gì chúng ta đã thấy trước đó khi chúng ta tìm hiểu về tổn thất entropy chéo để phân loại trong §2.1.2.

Không giống như entropy chéo với softmax trước đó, chúng ta đang ở trong một tình huống tồi tệ hơn ở đây, bởi vì số lượng giá trị có thể x có thể lấy lớn hơn nhiều. Trên thực tế, nó lớn hơn theo cấp số nhân, vì chúng ta thường sử dụng RBM hoặc bất kỳ mô hình tạo nào trong số này để mô hình hóa phân phối trên một không gian chiều cao. Nói cách khác, chúng ta không thể tính toán pha âm (b) chính xác trong một thời gian có thể xử lý được, hoặc đôi khi chúng ta không biết làm thế nào để tính toán nó.

Trong phần còn lại của phần này, chúng tôi nghiên cứu cách chúng tôi có thể vẽ các mẫu âm tính này một cách hiệu quả và sử dụng chúng để học.

5.1.1Lấy mẫu Markov Chain Monte Carlo (MCMC)

Hãy tưởng tượng rằng chúng ta muốn vẽ một tập hợp các mẫu từ một phân phối mục tiêu phức tạp $p_*(x)$. Sẽ thật tuyệt nếu chúng ta có thể vẽ các mẫu độc lập song song, nhưng điều này thường là không thể. Thay vào đó, chúng ta cần phải đưa ra một cách để vẽ một loạt các mẫu sao cho chúng tạo thành một tập hợp các mẫu độc lập từ phân phối đích. Chúng ta sẽ làm điều này như thế nào?

Chúng ta làm như vậy bằng cách xác định một chuỗi Markov (X, p0, T), trong đó X là tập hợp tất cả các quan sát có thể (tức là không gian trạng thái), p0 là phân phối ban đầu trên X và T là một toán tử chuyển tiếp. Toán tử chuyển tiếp thực sự không là gì ngoài phân phối vô điều kiện trên X cho một mẫu từ X, tức là T (x|x′). Chúng ta có thể rút ra một loạt các quan sát (x1, x2, . . .) bằng cách lặp đi lặp lại lấy mẫu $x_t \sim T(x|x_{t-1})$ với $x_0 \sim p_0(x)$. Cuối cùng, đó là, phần sau của chuỗi lấy mẫu lặp lại này, chúng ta muốn những mẫu đó được rút ra từ phân phối mục tiêu $p_*(x)$. Nói cách khác, chúng ta muốn một phân phối tĩnh p_∞, là số lượt truy cập tích lũy chuẩn hóa cho tất cả các trạng thái và thỏa mãn

$$p_\infty = T\, p_\infty, \quad (5.21)$$

để phù hợp với p_*. Khi chúng ta hội tụ đến phân phối tĩnh, khớp với phân phối mục tiêu, chúng ta có thể chỉ cần áp dụng toán tử chuyển tiếp lặp đi lặp lại

và tin rằng một loạt các mẫu được thu thập tạo thành một tập hợp các mẫu từ phân phối mục tiêu.

Ngoài điều kiện này ($p^\infty = p*$), chúng ta cần phải đáp ứng thêm một điều kiện. Nghĩa là, phân phối tĩnh này phải là duy nhất. Nếu có các phân phối tĩnh khác, chúng ta có thể không thể nói rằng ngay cả sau khi chạy toán tử chuyển tiếp này vô thời hạn rằng chúng ta đang thu thập các mẫu từ truedistribution. Để làm như vậy, chúng tôi tiếp tục đặt ra một ràng buộc rằng chuỗi Markov này isergodic. Trong chuỗi Markov ergodic, bất kỳ trạng thái nào (hoặc một vùng của không gian trạng thái, trong trường hợp X lớn vô hạn) đều có thể tiếp cận được từ bất kỳ trạng thái nào khác trong một số bước chuyển tiếp hữu hạn. Tính công thái học này đảm bảo rằng chỉ có một phân phối tĩnh và các ứng dụng lặp đi lặp lại của toán tử chuyển tiếp cuối cùng sẽ hội tụ về phân phối tĩnh duy nhất này. Lấy mẫu từ một phân phối mục tiêu phức tạp $p*$ sau đó rút gọn để hủy ký một toán tử chuyển tiếp T sao cho chuỗi Markov kết quả có một phân phối tĩnh duy nhất. Câu hỏi tiếp theo là làm thế nào chúng ta có thể đảm bảo rằng tồn tại một phân phối tĩnh, vì tính công thái học cho chúng ta biết rằng có một phân bố tĩnh duy nhất nếu có một phân bố tĩnh dưới chuỗi Markov này. Có nhiều hơn một cách để làm như vậy, và một cách tương đối nổi tiếng là nguyên tắc cân bằng chi tiết. Số dư chi tiết trong Markovchain được định nghĩa là có toán tử chuyển tiếp T thỏa mãn

$$T(x'|x)p^\infty(x) = T(x|x')p^\infty(x'). \tag{5.22}$$

Như khá rõ ràng từ phương trình, nó nói rằng bất cứ thứ gì chảy từ trạng thái này sang trạng thái khác đều phải chảy trở lại. Điều này mạnh hơn so với việc có phân phối tĩnh, vì phân phối tĩnh p^∞ có thể không thỏa mãn điều này. Khi cân bằng chi tiết được thỏa mãn, chúng ta thường gọi một chuỗi Markov như vậy là một chuỗi Markov có thể đảo ngược, vì chúng ta sẽ không thể biết hướng của thời gian một khi nó hội tụ.

Mục tiêu của chúng tôi sau đó là thiết kế một toán tử chuyển tiếp T sao cho chuỗi Markov kết quả là ergodic và thỏa mãn sự cân bằng chi tiết.1 Chúng tôi tham khảo quy trình lấy mẫu bằng cách thu thập một loạt các trạng thái đã truy cập từ chuỗi Markov như vậy bằng các phương pháp Chuỗi Markov Monte Carlo (MCMC).

Một trong những thuật toán MCMC phổ biến và được sử dụng rộng rãi nhất là thuật toán Metropolis-Hastings (MH) [Hastings, 1970]. Thuật toán MH giả định rằng chúng ta có quyền truy cập vào xác suất chưa chuẩn hóa $\tilde{p}*(x)$ của phân phối đích:

$$p*(x) = \frac{\tilde{p}*(x)}{\int \tilde{p}*(x)dx}. \tag{5.23}$$

Giả định này làm cho thuật toán MH đặc biệt phù hợp với nhiều mô hình dựa trên năng lượng, chẳng hạn như máy Boltzmann hạn chế (RBM), vì chúng ta có thể dễ dàng thường xuyên xác suất không chuẩn hóa nhưng không thể tính toán hằng số chuẩn hóa một cách dễ dàng.

1Tuyên bố này không loại trừ khả năng thiết kế một chuỗi Markov cho phép chúng ta lấy mẫu từ một phân phối mục tiêu ngay cả khi nó không thỏa mãn số dư chi tiết. Fur-thermore, tuyên bố này không loại trừ khả năng mở rộng không gian trạng thái bằng cách tăng thêm x với một biến phụ. Nó đã được chứng minh rằng điều này có thể có lợi vớicái gọi là phương pháp Hamiltonian Monte Carlos [Neal, 1993].

Đầu tiên chúng ta giả định rằng chúng ta được cho (hoặc có thể tạo ra) một phân phối đề xuất q(x|x′) thường tập trung ở x′ và khối lượng xác suất của nó chủ yếu tập trung trong vùng lân cận của x′. Q phải là ergodic, nghĩa là, nếu chúng ta lặp đi lặp lại lấy mẫu từ Q(x|x'), chúng ta sẽ có thể đạt được bất kỳ trạng thái nào (hoặc một vùng của không gian trạng thái) trong một số bước hữu hạn. Sau đó, chúng ta xác định một xác suất chấp nhận α(x|x′) sao cho

$$\alpha(x|x') = \text{phút } 1, \tilde{} \frac{p*(x)q(x'|x)\tilde{}}{\tilde{}p*(x')q(x|x')} \square \tag{5.24}$$

Sau đó, toán tử chuyển tiếp là

$$T(x|x') = \alpha(x|x')q(x|x') + (1 - \alpha(x|x')\delta x'(x), \tag{5.25}$$

đâu

$$\delta x'(x) = \begin{cases} (\infty, \text{ nếu } x = x' \\ 0, \quad \text{Khác} \end{cases} \tag{5.26}$$

và

$$\text{Với} \int \delta x'(x)dx = 1. \tag{5.27}$$

Chúng ta có thể lấy mẫu từ toán tử chuyển tiếp này cho mẫu trước đây x ′ bằng cách

$$(1)\tilde{} x \sim q(x|x') \qquad \text{(Thế hệ ứng viên)} \tag{5.28}$$

$$(2)\tilde{} u \sim U[0, 1] \qquad \text{(Rút thăm ngẫu nhiên)} \tag{5.29}$$

$$(3)\ x = \begin{cases} (\tilde{} x, \text{ nếu } \tilde{} u \le \alpha (\tilde{} \\ \tilde{}x|x')x', \text{ nếu không} \end{cases} \qquad \text{(Chấp nhận)} \tag{5.30}$$

Toán tử chuyển tiếp này đáp ứng cả tính công thái học và sự cân bằng chi tiết, và rất nhiều thuật toán MCMC có thể được xem là các biến thể của thuật toán MH với các lựa chọn cụ thể của phân phối đề xuất q.

Lấy mẫu Gibbs. Giả sử rằng x là một vectơ hữu hạn. Sau đó, chúng ta có thể xác định xác suất có điều kiện trên một chiều cụ thể d cho tất cả các chiều khác /= d như

$$pd(xd|x'1,...,d-1,d+1,...,|x|) = \frac{p([x'1, \ldots, x'd-1, x, x'd+1, \ldots, x|x|])R}{p([x'1, \ldots, x'd-1, \tilde{} x, x'd+1, \ldots, x|x|])d\tilde{} x}. \tag{5.31}$$

Giả sử d tuân theo một phân bố đồng đều, tức là d ∼ U {1, 2, . . . , |x|} và chúng ta bắt đầu từ x′ = [x′1, . . . , x′|x|]. Bây giờ chúng ta thay thế chiều thứ d của x bằng cách lấy mẫu từ phân phối có điều kiện pd, kết quả là x = [x′1, . . . , x′d−1,˜xd, x′d+1, . . . , x′|x|].

Để tính xác suất chấp nhận, chúng ta phải tính toán

$$\frac{\tilde{p}*(x)pd(x'|x)\tilde{p}*(x')pd(x|x') = \tilde{p}*([x'1, \ldots, x'd-1, xd, x'd+1, \ldots, x'|x|])pd(x'd|[\ x'1, \ldots, x'd-1, x'd+1, \ldots, x'|x|])p*([x'1, \ldots, x'|x|])pd(xd|[\ x'1, \ldots, x'd-1, x'd+1, \ldots, x'|x|])} \tag{5.32}$$

$$= \frac{\tilde{p}*([x'1, \ldots, x'd-1, xd, x'd+1, \ldots, x'|x|])\ \tilde{p}*([x'1, \ldots, x'd-1, x'd, x'd+1, \ldots, x'|x|])\ C([x'1, \ldots, x'd-1, x'd+1, \ldots, x'|}{\tilde{p}*([x'1, \ldots, x'd-1, xd, x'd+1, \ldots, x'|x|])\ C([x'1, \ldots, x'd-1, x'd+1, \ldots, x'|x|])}$$

$$= 1 \tag{5.34}$$

Nói cách khác, xác suất chấp nhận là 1 và chúng tôi luôn chấp nhận mẫu mới này khác với mẫu trước chỉ trong một chiều d.

Quy trình này được gọi là lấy mẫu Gibbs. Chúng ta chọn một tọa độ, lấy mẫu từ phân phối có điều kiện của tọa độ cụ thể đó, thay thế nó bằng giá trị tọa độ mới được lấy mẫu và lặp lại nó. Quy trình này thường có thể thực hiện được ngay cả khi chúng ta chỉ có quyền truy cập vào xác suất không chuẩn hóa, vì xác suất có điều kiện thường có thể xử lý được trong trường hợp đó. Hơn nữa, bởi vì mọi mẫu đều được tự động chấp nhận, hầu như không có thêm chi phí không triển khai, điều này làm cho nó trở thành một lựa chọn thuật toán hấp dẫn.

Suy luận biến thiên sẽ không hoạt động. Dựa trên những gì chúng ta đã học được trong §4.2, người ta có thể tự hỏi liệu chúng ta có thể sử dụng suy luận biến thiên thay vì lấy mẫu MCMC hay không. Câu trả lời rất tiếc là không. Ý tưởng cốt lõi của suy luận biến thiên là xấp xỉ một phân bố mục tiêu phức tạp (phân bố sau trong §4.2, và ở đây là phân phối mục tiêu $p*$) với một phân phối đơn giản hơnq bằng cách giảm thiểu

$$KL(q\|p*) = -Ex\sim q\ [\log \tilde{p}*(x)] + \log \underbrace{\tilde{Z}}_{\text{Const. w.R.T.}} + H\ (q). \tag{5.35}$$

Nếu chúng ta tập trung vào số hạng đầu tiên của phân kỳ KL, chúng ta quan sát thấy rằng chúng ta chỉ quan tâm đến vùng của không gian quan sát nơi q là cao. Nghĩa là, KLdivergence chỉ quan tâm đến các vùng có khả năng cao dưới q và bỏ qua bất kỳ vùng nào khác có khả năng cao dưới $p*$ nhưng không quan tâm đến q. Nói cách khác, các mẫu chúng tôi lấy từ q sau khi giảm thiểu phân kỳ KL ở trên sẽ không đại diện cho $p*$, bởi vì chúng phần lớn sẽ bỏ lỡ các vùng có thể xảy ra cao dưới $*$.

Vấn đề này biến mất khi độ phức tạp của q tăng lên và tiến gần đến mức độ phức tạp của $p*$. Tuy nhiên, điều này đi kèm với chính vấn đề mà chúng tôi muốn giải quyết; nghĩa là, chúng ta phải lấy mẫu từ q phức tạp như nhau này để tính toán xấp xỉ và giảm thiểu phân kỳ KL ở trên. Sau đó trong chương này, chúng ta xem xét việc xây dựng trực tiếp một bộ lấy mẫu để một q được xác định ngầm vừa đủ phức tạp vừa xấp xỉ giảm thiểu phân kỳ KL ở trên.

5.1.2(Dai dẳng) Phân kỳ tương phản

Chúng ta cần lấy mẫu từ p(x; θ), để đào tạo RBM. Một cách để tạo ra một tập hợp các mẫu từ p(x; θ) là vẽ một tập hợp các mẫu (x, z) từ p(x, z; θ) và loại bỏ z từ mỗi cặp. Khi làm như vậy, chúng tôi muốn sử dụng lấy mẫu Gibbs. Trước tiên, chúng ta hãy thử viết ra xác suất có điều kiện của z cho x:

$$\log p(z|x;\theta) = \sum_{j=1}^{|\text{với}|} zj \sum_{i=1}^{|x|} Wijxi + CJ + \text{định số.} \tag{5.36}$$

Điều này ngụ ý rằng z1, . . . , z|z| là độc lập có điều kiện cho x, như

$$p(z|x;\theta) = \prod_{j=1}^{|\text{với}|} \text{Và} \frac{\text{Exp } zj\, w^{T}\cdot j\, x + cj\exp(0)}{+ \exp w^{T}\cdot j\, x + cj\,.} \tag{5.37}$$

Do đó, chúng ta có thể xem xét từng chiều của z một cách riêng biệt:

$$p(zj = 1|x;\theta) = \frac{\exp(w^{T}\cdot j\, x + cj)1 +}{\exp(w^{T}\cdot j\, x + cj) =} \quad \frac{11 + \exp(-w^{T}\cdot j\, x - cj) = \sigma(w^{T}\cdot j\, x + cj),}{} \tag{5.38}$$

trong đó σ là một hàm sigmoid mà chúng ta đã thấy trước đó:

$$\sigma(a) = \frac{11 + \exp}{(-a)\,.} \tag{5.39}$$

Lấy mẫu tất cả |z| các chiều có thể song song một cách đáng xấu hổ, vì chúng độc lập có điều kiện. Giả sử chúng ta đã lấy mẫu một z mới. Bây giờ chúng ta cần lấy mẫu một x mới cho z. Sau một dẫn xuất tương tự, chúng ta kết thúc với

$$p(x|z;\theta) = \prod_{i=1}^{|x|} \text{Và}\, p(xi|z;\theta), \tag{5.40}$$

đâu

$$p(xi = 1|z;\theta) = \sigma(w^{T}i\cdot z + bi). \tag{5.41}$$

Nói cách khác, chúng ta cũng có thể lấy mẫu song song tất cả các chiều của x. Sau đó, chúng ta có thể xen kẽ giữa lấy mẫu x và z nhiều lần để thu thập một loạt các cặp (x, z) tạo thành một tập hợp các mẫu được rút ra từ p(x, z; θ). Tất nhiên, chúng tôi có thể muốn loại bỏ khá nhiều cặp từ giai đoạn đầu của việc lấy mẫu, vì chúng có thể đã được thu thập trước khi Markovchain hội tụ. Hơn nữa, để tránh tốc độ trộn chậm của chuỗi Markov, chúng ta có thể chỉ muốn sử dụng mọi mẫu thứ k. Chiến lược này thường được gọi là làm mỏng.

Tất nhiên, điều này không thực sự giúp chúng tôi quá nhiều, vì chúng tôi phải chạy một chuỗi lấy mẫu Gibbs để thu thập đủ các mẫu độc lập.

Nếu chúng ta chạy quá ngắn, ước tính gradient ngẫu nhiên của chúng ta có thể sẽ không chính xác, dẫn đến một kết quả thảm khốc.

Thay vào đó, hóa ra chúng ta có thể chỉ cần bắt đầu chuỗi lấy mẫu Gibbs từ một ví dụ tích cực, chỉ chạy một số bước nhỏ (ít nhất là chỉ một) và sử dụng mẫu kết quả làm ví dụ tiêu cực. Đó là

$$\nabla_\theta L_k(\theta; x) = \nabla_\theta e(x, \theta) \underbrace{|_{\{z\}=\text{tích cực}}}_{} - \frac{1}{S} \sum_{s=1}^{S} \nabla e(x's, \theta) \underbrace{|_{\{z\}=\text{tiêu cực}}}_{}, \tag{5.42}$$

trong đó x's là một trong các mẫu S được rút ra sau khi chạy k bước của lấy mẫu Gibbs bắt đầu từ x. Thông thường đặt S thành 1. Trong giới hạn $k \to \infty$, điều này là chính xác, vì mẫu âm x′ sẽ là từ phân bố tĩnh trùng với phân bố thực p(x; θ). Tuy nhiên, nó không phải như vậy với một k hữu hạn, và thậm chí không có sự đảm bảo rằng một k lớn hơn dẫn đến một xấp xỉ tốt hơn, khi k là nhỏ. Tuy nhiên, chiến lược này dẫn đến một RBM được đào tạo hợp lý và thường được gọi là phân kỳ tương phản.

Hóa ra chúng ta có thể duy trì độ phức tạp tính toán với chi phí tối thiểu về độ phức tạp của bộ nhớ bằng cách duy trì các mẫu S trên các bước gradient ngẫu nhiên trong khi đảm bảo rằng việc học hội tụ đến giải pháp chính xác một cách tiệm cận. Chúng tôi làm như vậy bằng cách chạy chuỗi S của lấy mẫu Gibbs song song với độ dốc ngẫu nhiên. Giữa các bước liên tiếp của SGD, chúng tôi chạy chuỗi S của lấy mẫu Gibbs cho T ≈ mỗi bước 1 bước để cập nhật một tập hợp các mẫu S có nhiều khả năng được rút ra từ mô hình mới nhất. Sau đó, chúng tôi sử dụng các mẫu mới được cập nhật này để tính toán ước tính độ dốc ngẫu nhiên, để cập nhật các tham số mô hình.

Khi việc học tiếp tục, sự thay đổi đối với các tham số mô hình chậm lại (vì chúng ta ngày càng tiến gần hơn đến mức tối thiểu cục bộ), và do đó các chuỗi sam-pling Gibbs trong nền ngày càng tiến gần hơn đến sự phân phối cố định của mô hình cuối cùng. Điều này làm cho giai đoạn đầu của việc học không chính xác nhưng có phương sai thấp (vì chúng ta không làm xáo trộn các ví dụ tiêu cực quá nhiều) nhưng giai đoạn sau là chính xác vì các tham số mô hình thay đổi rất chậm. Chiến lược này được gọi là phân kỳ tương phản dai dẳng.

5.2 Mạng lưới đối nghịch sinh sản dựa trên năng lượng

Thật khó để lấy mẫu từ một phân phối phức tạp, chiều caongay cả với thuật toán MCMC tiên tiến. Thay vào đó, chúng ta có thể muốn xem xét đào tạo một mạng nơ-ron để lấy mẫu từ một phân phối như vậy. Bất kỳ mạng nơ-ron nào như vậy có thể được mô tả là

$$x = g(e; \theta g), \tag{5.43}$$

trong đó ε ∼ P (ε) và P (ε) là một số phân phối dễ lấy mẫu mà chúng tôi lựa chọn. Bộ lấy mẫu này được tham số hóa bởi θg.

Chúng ta có thể đào tạo bộ lấy mẫu này bằng cách giảm thiểu hàm mất sau:

$$\text{Lrkl}(\theta g) = \text{KL}(pg \| pe) = -\text{Ex}\sim pg\,[\log pe(x) - \log pg(x)] \tag{5.44}$$

$$= -\,\text{Ex}\sim pg\,[\log pe(x)] \quad - H \quad , \tag{5.45}$$

trong đó pg là phân phối cơ bản của bộ lấy mẫu g và pe là phân phối được xác định từ hàm năng lượng e bằng cách sử dụng công thức Boltzmann. Chúng tôi sẽ xem xét hai thuật ngữ trong hàm mất mát này một cách riêng biệt.

Số hạng đầu tiên (a) là năng lượng dự kiến âm của x cộng với một số hằng số:

$$(a) = \text{Ex}\sim pg\,[\log pe(x)] = E^{\square -e(x) - \text{nhật ký}^{\text{Với}} \exp(-e(x'))dx'} \tag{5.46}$$

$$= E\,[-e(x)] + \text{hằng số}. \tag{5.47}$$

Mặc dù chúng ta không có pg, nhưng chúng ta có thể lấy mẫu từ phân phối này với g. Do đó, chúng ta có thể tính toán độ dốc ngẫu nhiên của (a):

$$\nabla a\theta g \approx -\frac{1}{M} \sum_{m=1}^{M} \nabla\theta g\, e(g(\varepsilon m)), \tag{5.48}$$

trong đó εm ∼ p(ε). Miễn là g có thể vi sai được wrt θg và e có thể vi saiw.r.t. đầu vào, chúng ta có thể tính toán gradient ngẫu nhiên này bằng cách sử dụng backpropagation. Bằng cách đi theo hướng ngược lại với gradient ngẫu nhiên này, chúng ta có thể giảm thiểu hiệu quả số hạng đầu tiên (a).

Thật không may, (b) ít tầm thường hơn để tính toán, vì chúng ta không có quyền truy cập topg. Thay vì tối đa hóa entropy (xem dấu âm trước (b),), chúng ta có thể cố gắng làm cho pg gần hơn với một phân phối khác có khả năng có entropy cao hơn. Giả sử rằng X là một không gian thực đa chiều, tức là Rd, phân phối chuẩn là phân phối entropy tối đa cho một trung bình và một ma trận hiệp phương sai. Do đó, chúng ta có thể lấy nhiều mẫu từ pg bằng cách sử dụng g, ước tính trung bình μgand hiệp phương sai Σg từ các mẫu này và sau đó sử dụng phân phối chuẩn với μg và αΣg làm giá trị trung bình và hiệp phương sai, tương ứng, làm phân phối mục tiêu có entropy cao hơn pg, với α > 1.

Khi chúng ta có hai tập hợp mẫu được rút ra từ hai phân phối, chúng ta có thể sử dụng sự chênh lệch trung bình tối đa nhân (MMD) để đo lường sự tương đồng giữa hai phân phối này. Thật không may, nó chắc chắn nằm ngoài phạm vi của khóa học để thảo luận về MMD và công cụ ước tính hạt nhân của nó [Gretton et al., 2012]. Thay vào đó, chúng tôi sẽ tin tưởng rằng những điều sau đây đo lường sự khác biệt giữa hai

khi chúng ta chỉ có hai bộ mẫu:

$$\mathrm{MMD2}(D, D') = \frac{1}{(1)} \frac{|D|(|D| - \quad X \quad X \quad k(x, x')}{| \quad \{z \quad \}} \tag{5.49}$$

$$+ \frac{1}{(1)} \frac{|D'|(|D'| - \quad X \quad X \quad k(x, x')}{| \quad \{z \quad \}} \tag{5.50}$$

$$- \frac{2|D||D'|}{|} \quad X \quad X \quad k(x, x'), \tag{5.51}$$

trong đó $k(\cdot, \cdot)$ là một hàm nhân. Chúng ta sẽ không thảo luận về các hàm hạt nhân là gì, nhưng bạn có thể coi hàm hạt nhân như một số loại số liệu khoảng cách, sao cho bất kỳ hàm hạt nhân nào k (a, b) thỏa mãn hai thuộc tính. Đầu tiên, nó đối xứng:

$$k(a, b) = k(b, a). \tag{5.52}$$

Thứ hai, nó là xác định bán dương:

$$x^{\top} K x \geq 0, \text{ cho tất cả } x \in \mathbb{R}^n, \tag{5.53}$$

trong đó K là ma trận n × n với mỗi mục nhập Kij = k(vi, vj) cho bất kỳ tập hợp nào {vi}ni=1.Đối với các vectơ thực, một lựa chọn thông thường là nhân Gaussian được định nghĩa là

$$k(a, b) = \text{kinh nghiệm} \quad \square - \frac{1}{\sigma 2} \|a - b\|2 \quad . \tag{5.54}$$

Bởi vì MMD hạt nhân ở trên có thể phân biệt được với các mẫu, miễn là hàm hạt nhân được chọn để có thể vi sai, chúng ta có thể tính toán độ dốc của MMD w.r.t. các tham số của bộ lấy mẫu g và sử dụng nó thay cho độ dốc của (b) từ phương trình (5.55).

Mặc dù chúng tôi sẽ không đi vào bất kỳ chi tiết kỹ thuật nào đằng sau MMD hạt nhân này, nhưng việc kiểm tra nó ở mức độ trực quan là rất hữu ích. Chúng ta hãy bắt đầu từ phía sau. Thuật ngữ thứ ba (c) là chính xác về mặt trực giác, vì nó tính toán khoảng cách trung bình theo cặp giữa tất cả các cặp mẫu có thể có từ hai phân phối. Nếu khoảng cách theo cặp trung bình lớn hơn, sự khác biệt giữa hai phân phối cơ bản cũng phải cao.

Giả sử |D| = |D ′ | (nghĩa là, chúng ta có cùng một số mẫu từ mỗi phân phối.) Sau đó, khoảng cách theo cặp tối thiểu này có thể được xác định bởi khoảng cách theo cặp trung bình trong mỗi tập hợp, vì tất cả các mẫu này sẽ được đặt trên các mẫu từ phân phối khác. Hơn nữa, khi điều này xảy ra, hai số hạng đầu tiên, (a) và (b), sẽ trùng với nhau. Xem xét rằng hai số hạng đầu tiên và thuật ngữ cuối cùng có các dấu đối lập, chúng sẽ triệt tiêu lẫn nhau, kết quả là 0, như

Mong muốn. Nói cách khác, (c) xác định sự khác biệt tổng thể giữa hai phân phối, trong khi (a) và (b) ở đó để tính đến sự khác biệt nhỏ giữa hai phân phối phần lớn bị giới hạn từ bên dưới bởi sự phân tán trong phân phối.

Bằng cách giảm thiểu tổn thất sau đây, chúng ta có thể đào tạo một mạng lấy mẫu g chuyển đổi một mẫu từ một phân phối đơn giản p (ε) thành một mẫu từ phân phối mục tiêu được xác định từ hàm năng lượng e:

$$Jg(\theta g; e) = -\frac{1}{M}\sum_{m=1}^{N} e(g(\varepsilon m)) - \lambda \underbrace{MMD2\,\square\{sn\}Nn=1, \{g(\varepsilon m)\}Mm=1\square}_{=R(\theta g)}, \quad (5.55)$$

đâu

$$sn \sim N\left(\mu = \frac{1}{\text{triệu}}\sum_{m=1}^{M} g(em), aS\right) \quad (5.56)$$

với

$$\Sigma = \frac{1}{M}\sum_{m=1}^{M} (g(\varepsilon m) - \mu)\,(g(\varepsilon m) - \mu)^{T}. \quad (5.57)$$

Điều quan trọng là phải coi sn là hằng số hơn là các hàm của εm. λ > 0kiểm soát sự cân bằng giữa hai số hạng này.

Bây giờ, chúng ta có thể sử dụng bộ lấy mẫu g này thay vì sử dụng bộ lấy mẫu MCMC tốn kém để lấy mẫu từ một hàm năng lượng. Nói cách khác, chúng ta có thể tính gradient cho hàm năng lượng từ phương trình (5.19) bằng cách rút mẫu từ bộ lấy mẫu g:

$$\tilde{\nabla}\theta = \nabla\theta e(x, \theta) - \frac{1}{M}\sum_{m=1}^{M} \nabla\theta e(xm, \theta), \quad (5.58)$$

trong đó xm = g(εm; θg) với εm ∼ p(ε).

Nếu bạn nhìn vào số hạng đầu tiên từ phương trình (5.55) (hàm khách mục để được tối đa hóa để đào tạo g) và số hạng thứ hai ở trên, có thể dễ dàng thấy rằng chúng giống hệt nhau. Sau đó, chúng ta có thể đặt hai điều này lại với nhau thành một hàm mục tiêu duy nhất và sau đó thấy rằng chúng ta có thể đào tạo cả hàm năng lượng và bộ lấy mẫu cùng nhau bằng cách giải một bài toán minimax:

$$\min\theta \max\theta g\; Ex\sim D\,[e(x, \theta)] - E\varepsilon\sim p(\varepsilon)\,[e(g(\varepsilon; \theta g), \theta)] - \lambda R(\theta g). \quad (5.59)$$

Nói một cách khác, chúng tôi cố gắng điều chỉnh θ để đảm bảo các trường hợp đào tạo được gán các giá trị năng lượng thấp hơn, trong khi các mẫu được lấy từ pg được gán các giá trị năng lượng cao hơn. Trong khi đó, chúng tôi đảm bảo rằng bộ lấy mẫu g rút ra các mẫu được chỉ định giá trị năng lượng thấp hơn và sự phân bố ngầm của entropy pg được tối đa hóa.

Bởi vì chúng ta không phụ thuộc vào việc lấy mẫu Gibbs, chúng ta có thể lỏng lẻo hơn nhiều về cách thiết kế một hàm năng lượng, không giống như RBM ở trên. Một

Lựa chọn tự nhiên là một bộ mã hóa tự động xác định tương tự như bộ mã hóa tự động biến thể từ §4.3.1 tuy nhiên không có bất kỳ nhiễu nào ở giữa. Với bộ mã hóa tự động xác định, hàm năng lượng được định nghĩa là

$$e(x; \theta) = \|F(G(x; \theta G); \theta F) - x\|2, \tag{5.60}$$

trong đó $\theta = \theta G \cup \theta F$. Giá trị năng lượng thấp hơn nếu x có thể được tái tạo tốt hơn. Người ta có thể xem đây là hàm năng lượng e và bộ lấy mẫu g đang chơi một trò chơi đối nghịch. Công việc của hàm năng lượng là đảm bảo rằng các mẫu của bộ lấy mẫu ít có khả năng hơn các đầu vào thực, trong khi công việc của bộ lấy mẫu là đảm bảo rằng các mẫu được tạo ra có khả năng là đầu vào thực theo hàm năng lượng. Cách tiếp cận này được tiên phong bởi Goodfellow et al. [2014], và cách đặc biệt này để mô tả cách tiếp cận này bằng cách sử dụng hàm năng lượng đã được Zhao et al. [2016] khám phá ngay sau đó. Sau khi đào tạo kết thúc, người ta có thể sử dụng bộ lấy mẫu nguyên trạng hoặc có thể sử dụng bộ lấy mẫu làm khởi tạo để lấy mẫu từ hàm năng lượng được đào tạo.

5.3 Mô hình tự hồi quy

Cho đến nay, chúng tôi đã xem xét một họ các mô hình tổng hợp, được gọi là mô hình biến tiềm ẩn. Bất kể các phụ thuộc xác suất được mô tả bằng cách sử dụng các cạnh có hướng hay không có hướng, chúng tôi đã sử dụng các biến không quan sát được, hoặc các biến thể tiềm ẩn, để nắm bắt các phân phối phức tạp. Đối với mỗi cấu hình biến tiềm ẩn, chúng ta xác định một phân phối tương đối đơn giản trên quan sát. Chúng ta gọimột phân phối đơn giản khi phân phối này có một số lượng nhỏ tham số và nếu chúng ta có thể xây dựng một mạng nơ-ron có thể vi phân ánh hình dung biến tiềm ẩn với các tham số này của phân phối. Bằng cách loại bỏ các biến tiềm ẩn này, chúng ta kết thúc với một mô hình có thể nắm bắt một phân phối phức tạp. Sau đó, có lựa chọn thay thế nào không?

Một phân phối đơn giản như vậy thường không đủ để nắm bắt tất cả các biến thể của quan sát đầy đủ X hầu như luôn bao gồm các thành phần đơn giản hơn (chiều thấp hơn), tức là $X = \{x1, \ldots, xd\}$. Tuy nhiên, một phân phối đơn giản như vậy thường đủ để nắm bắt sự phân phối có điều kiện trên một thành phần riêng lẻ, thường có chiều thấp hơn đáng kể. Ví dụ: nếu xi là một biến phân loại với các thể loại C, chúng ta có thể dễ dàng sử dụng softmax với các tham số C để nắm bắt phân phối này. Tuy nhiên, X có thể lấy Cd nhiều giá trị có thể xảy ra, và điều này sẽ không dễ dàng để nắm bắt với tham số hóa dựa trên softmax đơn giản. Sau đó, thật hấp dẫn để tưởng tượng mô hình hóa các thành phần d này của X một cách riêng biệt và kết hợp chúng để xây dựng một mô hình của X.

Nhớ lại quy tắc chuỗi xác suất:

$$p(X) = p(x\Pi(1))p(x\Pi(2)|x\Pi(1))p(x\Pi(3)|x\Pi(1), x\Pi(2)) \cdots \tag{5.61}$$

$$= \prod_{i=1}^{d} p(x\Pi(i)|x\Pi(1), \ldots, x\Pi(i-1))|\{z\}=(a), \tag{5.62}$$

trong đó Π là hoán vị tùy ý của (1, 2, . . . , d). Quy tắc chuỗi này nói rằng xác suất của bất kỳ cấu hình nào của X có thể được tính toán dưới dạng tích của xác suất của các thành phần d, được điều chỉnh một cách thích hợp trên một tập hợp con của các thành phần. Không mất tính tổng quát, chúng ta giả định Π(i) = i.

Mục tiêu của chúng tôi là xây dựng một mạng nơ-ron mô hình hóa (a) ở trên và do đó mô hình hóa hàm xác suất chung p (X). Có hai điều cần xem xét. Đầu tiên, chúng tôi không muốn có d các mạng nơ-ron riêng biệt để nắm bắt các phân phối xác suất có điều kiện d. Thay vào đó, chúng ta muốn có một mạng nơ-ron duy nhất có thể mô hình hóa mối quan hệ giữa bất kỳ cặp nào của chiều mục tiêu xivà các chiều ngữ cảnh x<i = (x1, . . . , xi−1). Điều này cho phép công cụ dự đoán hưởng lợi từ các mô hình được chia sẻ trên các cặp này. Ví dụ, nếu xi là i-thpixel trong một hình ảnh, chúng ta biết rằng giá trị điểm ảnh của xi phải hơi giống với xi−1, bất kể i, do tính cục bộ của các giá trị pixel. Kiến thức này sẽ dễ dàng nắm bắt hơn nếu một công cụ dự đoán duy nhất được sử dụng cho tất cả i.

Thứ hai, số lượng tham số không được tăng w.r.t. d, tức là |θ| = o(d). Trên thực tế, nó là mong muốn để có |θ| = O(1), bằng cách hoàn toàn không phụ thuộc vào d.Điều này cho phép chúng ta xây dựng một mô hình không giám sát có thể hoạt động trên một quan sát có kích thước thay đổi, điều này cực kỳ quan trọng khi xử lý các tùy chọn có độ dài thay đổi, chẳng hạn như văn bản ngôn ngữ tự nhiên và video.

Kết hợp hai cân nhắc này, bây giờ chúng ta có thể viết cách tiếp cận này dưới dạng

$$xi \sim G(F((x1, x2, \ldots, xi-1); \theta), \varepsilon), \tag{5.63}$$

trong đó ε là tiếng ồn. Điều này nhắc nhở chúng ta về mô hình tự hồi quy trong xử lý tín hiệu,2 và do đó chúng ta đề cập đến một cách tiếp cận như vậy là mô hình tự hồi quy, vì điều này tương tự như một mô hình tự hồi quy phi tuyến tính với bối cảnh không giới hạn ($p \to \infty$). Hai khối xây dựng từ §3 đặc biệt thích hợp để thực hiện F; arecurrent và khối chú ý (với mã hóa định vị.) Trong trường hợp một khối lặp lại, chúng ta không cần bất kỳ sửa đổi nào, nhưng có thể chỉ cần nạp theo toàn bộ trình tự (x0, x1, x2, . . . , xd) và đọc ra (p(x1), p(x2|x1), . . . , p(xd|x<d)). Cụ thể hơn, nếu chúng ta sử dụng các đơn vị lặp lại có cổng,

$$hi = FGRU([xi, hi-1]; \theta r) \tag{5.65}$$

$$p(xi+1|x\le i) = \frac{\exp(u^\top xi+1hi + cxi+1)}{\sum_{x\in C} \exp(u^\top x\, hi + cx)}, \tag{5.66}$$

trong đó h0 là một phần của tham số và x0 là vectơ giữ chỗ. Sau đó, chúng ta có thể đào tạo mạng lặp lại này để giảm thiểu mất nhật ký trung bình:

$$\min_{\theta r, U, c} -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{Dn} \log p(xni|xn<i; \theta r, U, c), \tag{5.67}$$

2Một mô hình tự hồi quy điển hình của bậc p là xử lý tín hiệu được định nghĩa là

$$xi = \sum_{k=1}^{p} \theta k xi-k + \varepsilon i. \tag{5.64}$$

nơi chúng ta đang rõ ràng về khả năng của các quan sát có kích thước thay đổi bằng cách viết dn.

Tuy nhiên, khối chú ý yêu cầu một sửa đổi nhỏ. Việc sửa đổi này là cần thiết, vì chúng ta phải đảm bảo rằng hi chỉ được tính bằng cách sử dụng (x0, x1, . . . , xi). Điều này có thể được thực hiện bằng cách che đi các trọng số chú ý từ Phương trình (3.26) như

$$\text{Aji} = \frac{\exp(q^{\top} i\ kj - mij)\text{PNj}'=1}{\exp(q^{\top} i\ kj' - mij')}, \tag{5.68}$$

đâu

$$\text{tôi} = \begin{cases} 0, \text{ nếu } j < i\infty, \\ \text{nếu } j \geq i \end{cases} \tag{5.69}$$

Điều này sẽ đảm bảo rằng đầu ra ˆvi từ khối chú ý không được tính toán bằng bất kỳ vectơ đầu vào nào (xi, xi + 1, . . . , xd). Một số người gọi điều này là che giấu nhân quả bằng cách vay mượn từ khái niệm về một hệ thống nhân quả trong xử lý tín hiệu.

Chúng ta phải cẩn thận khi chúng ta đang đối phó với xi liên tục. Chúng ta sẽ thảo luận tại sao lại như vậy, và làm thế nào chúng ta có thể giải quyết nó một cách đúng đắn trong §6.4, nếu thời gian cho phép.

Một lợi thế chính của phương pháp mô hình tự hồi quy này là chúng ta có thể tính toán xác suất log của bất kỳ quan sát nào một cách chính xác. Chúng ta chỉ cần tính toán xác suất log có điều kiện và tổng chúng để có được xác suất log của quan sát. Điều này không giống như bất kỳ cách tiếp cận biến tiềm ẩn nào mà chúng tôi đã đề cập ở trên. Trong trường hợp của một bộ mã hóa tự động biến thể, chúng ta phải giải quyết vấn đề gạt ra bên lề khó giải quyết, và trong trường hợp của RBM, chúng ta phải tính toán hàm phân vùng log khó giải quyết, hoặc hằng số log-normaliztaion. Hơn nữa, chúng tôi có thể dễ dàng vẽ các mẫu độc lập một cách dễ dàng với mô hình hồi quy này, đây là một lợi thế lớn cho RBM yêu cầu lấy mẫu MCMC tốn kém và đầy thách thức.

Mô hình mô hình tự hồi quy này đã trở thành tiêu chuẩn trên thực tế xây dựng các tác nhân đàm thoại trong những năm gần đây kể từ khi Brown et al. [2020] và Ouyang et al. [2022] chứng minh thành công. Để tìm hiểu thêm về các nguyên tắc cơ bản đằng sau mô hình ngôn ngữ và các ý tưởng liên quan, hãy xem ghi chú bài giảng có phần lỗi thời này [Cho, 2015]. Chúng tôi không đi sâu vào bất kỳ chi tiết nào, vì các chủ đề này nằm ngoài phạm vi của khóa học này.

Chương 6

Các chủ đề khác

6.1 Học tăng cường

Học tăng cường một bước. Chúng ta đang ở trong một tình huống mà chúng ta phải đào tạo một bộ phân loại nhưng chúng ta không được cung cấp các cặp đầu vào-đầu ra, mà là một hộp đen lấy làm đầu vào một trong các đầu ra và trả về phần thưởng vô hướng, tức là R : {1, . . . , C} → R. Đây thực sự là một hộp đen, không giống như học từ §2.5 khi học là một hộp đen do tính khó xử lý của nó. Chúng ta muốn đào tạo một bộ phân loại để chúng ta tối đa hóa phần thưởng bằng hộp đen trên kỳ vọng:

$$\max_\theta \; \mathrm{E}_{x\mathrm{E}y|x;\theta}\,[R(y)]\,. \tag{6.1}$$

Câu hỏi đầu tiên chúng ta thường cần hỏi là liệu chúng ta có thể tính toán độ dốc ngẫu nhiên của mục tiêu này với các tham số θ hay không. Chúng ta hãy tự mình thử điều đó ở đây:

$$\nabla_\theta \int_x p(x) \sum_{y=1}^{C} p(y|x;\theta) R(y)\,dx = \int_x p(x) \underbrace{\nabla_\theta \sum_{y=1}^{C} p(y|x;\theta) R(y)}_{=\nabla \mathrm{E}_{y|x;\theta} R(y)}\,\mathrm{D}x. \tag{6.2}$$

Chúng tôi tiếp tục với $\nabla \mathrm{E}_{y|x;\theta} R(y)$:

$$\nabla_\theta \sum_{y=1}^{C} p(y|x;\theta) R(y) = \sum_{y=1}^{C} \nabla_\theta p(y|x;\theta) R(y) \tag{6.3}$$

$$= \sum_{y=1}^{C} p(y|x;\theta) R(y) \nabla \log p(y|x;\theta) \tag{6.4}$$

$$= \mathrm{E}_{y|x;\theta}[R(y) \nabla \log p(y|x;\theta)], \tag{6.5}$$

nơi chúng ta sử dụng cái gọi là thủ thuật dẫn hàm log.[1]

[1] $$f' = f \cdot (\text{nhật ký } f)', \tag{6.6}$$

Nói cách khác, gradient ngẫu nhiên của phần thưởng dự kiến được đưa ra cho đầu vào x là tổng trọng số của gradient ngẫu nhiên của xác suất log được gán cho mỗi đầu ra có thể, trong đó trọng số là phần thưởng liên quan và đầu ra được vẽ theo phân phối đầu ra của bộ phân loại. Điều này có ý nghĩa trực giác. Chúng tôi muốn tuân theo hướng gradient sẽ khuyến khích bộ phân loại đặt xác suất cao hơn cho đầu ra có liên quan đến phần thưởng cao hơn, nhiều hơn so với các hướng khác. Bởi vì nó thường tốn kém (hoặc thậm chí không thể) để chạy hộp đen này, nó là một thực tế thông thường để sử dụng một dranw mẫu duy nhất từ y|x; θ để xấp xỉ gradient ngẫu nhiên này:

$$\nabla\theta E_{y|x;\theta}[R(y)] \approx R(\tilde{y})\nabla\theta \log p(\tilde{y}|x;\theta) = \hat{g}, \tag{6.8}$$

trong đó $\tilde{y} \sim y|x;\theta$.
Trước khi tuyên bố chiến thắng, chúng ta hãy tính toán phương sai của công cụ ước lượng stochasticgradient này:

$$V[\hat{g}] = E[\hat{g}2] - E[\hat{g}]2. \tag{6.9}$$

Mặc dù chúng ta biết rằng đây là một cộng cụ ước tính không thiên vị vì chúng ta đã loại bỏ nó hoàn toàn cho đến khi chúng ta sử dụng xấp xỉ Monte Carlos mẫu đơn (tự nó không thiên vị), trước tiên chúng ta hãy tính toán $E[\hat{g}]$:

$$E[\hat{g}] = E_{y|x;\theta}[R(y)\nabla\theta \log p(y|x;\theta)] \tag{6.10}$$

$$= \sum_{\text{và}} p(y|x;\theta)R(y)\nabla\theta \log p(y|x;\theta) \tag{6.11}$$

$$= \sum_{\text{và}} R(y)\nabla\theta p(y|x;\theta) \tag{6.12}$$

$$= \nabla\theta \sum_{\text{và}} R(y)p(y|x;\theta) \tag{6.13}$$

$$= \nabla\theta E_{y|x;\theta}[R(y)] \tag{6.14}$$

Sau đó, chúng ta cần tính số hạng đầu tiên của phương sai ở trên:

$$E\ \hat{g}2 = E_{y|x;\theta}\, R2(y)\|\nabla\theta \log p(y|x;\theta)\|2 \tag{6.15}$$

Kết hợp chúng lại với nhau, chúng ta nhận được

$$V\ [\hat{g}] = E\ R2(y)\|\nabla \log p(y|x;\theta)\|2 - \|\nabla\theta E[R(y)]\|2. \tag{6.16}$$

Nhìn vào số hạng đầu tiên của phương sai, chúng ta nhận thấy rằng có hai điều ảnh hưởng rất nhiều đến phương sai. Yếu tố đầu tiên là mức độ của phần thưởng. Nếu phần thưởng có độ lớn cao, nó dẫn đến sự gia tăng phương sai của

---

vì

$$(\log f)' = f'f \tag{6.7}$$

ước tính độ dốc ngẫu nhiên. Điều này cho thấy rằng điều quan trọng đối với chúng ta là kiểm soát mức độ của phần thưởng, mặc dù điều này là không thể nếu chúng ta đang làm việc với hộp đen R thực sự. Yếu tố thứ hai là định mức của gradient của xác suất log-của hành động đã chọn với các tham số θ. Nói cách khác, phương sai sẽ lớn hơn nếu xác suất dự đoán được tính toán bởi mô hình nhạy cảm với sự thay đổi trong các thông số. Điều này gợi ý một cách rất rõ ràng để chính quy hóa việc học bằng cách giảm thiểu đại lượng này trực tiếp, để ổn định việc học. Kỹ thuật này thường được gọi là hình phạt gradient.

Tại thời điểm này, chúng ta bắt đầu tự hỏi liệu có một công cụ ước tính ngẫu nhiên khác không thiên vị nhưng có khả năng có phương sai thấp hơn hay không. Hãy xem xét công cụ ước tính sau đây, thường được gọi là công cụ ước tính gradient chính sách:

$$\nabla\theta E_{y|x;\theta}\,[R(y)] \approx (R(\tilde{y}) - b(x))\nabla\theta \log p(\tilde{y}|x;\theta), \tag{6.17}$$

trong đó b có thể là một hàm của x nhưng độc lập với y. Nếu chúng ta xem xét giá trị kỳ vọng của phía bên trái, chúng ta nhận thấy rằng

$$E\,[R(\tilde{y})\nabla\theta \log p(y|x;\theta)] - b(x)\,E\,[\nabla\theta \log p(y|x;\theta)] \quad . \tag{6.18}$$

Chúng ta hãy đào sâu hơn vào (a) ở trên:2

$$\sum_{y=1}^{C} p(y)\nabla \log p(y) = \sum_{y=1}^{C} p(y)\frac{1}{p(y)}\nabla p(y) = \nabla \sum_{y=1}^{C} p(y) = 0. \tag{6.19}$$

Nói cách khác, công cụ ước lượng trong phương trình (6.17) là một công cụ ước tính không thiên vị. Mặc dù công cụ ước lượng này giống hệt với công cụ ước tính ban đầu về độ lệch, nhưng phép trừ thêm b(x) từ R(˜ y) này có một hệ quả quan trọng đối với phương sai. Chúng ta hãy xem xét số hạng đầu tiên của phương sai bằng cách sử dụng ước lượng mới trong Phương trình (6.16). Chúng tôi đặc biệt quan tâm đến việc tìm b(x) giảm thiểu thuật ngữ này:

$$\nabla b\, \tfrac{1}{2}\, E_{y|x;\theta}\,(R(y) - b)^2\|s(y)\|^2 = -E\,(R(y) - b)\|s(y)\|^2 = 0 \tag{6.20}$$

$$\iff b E\|s(y)\|^2 - E\,R(y)\|s(y)\|^2 = 0 \tag{6.21}$$

$$\iff b_* = \frac{E\,R(y)\|s(y)\|^2}{E\|s(y)\|^2}. \tag{6.22}$$

trong đó s(y) = ∇ log p(y|x; θ). Thật không may, đường cơ sở tối ưu này không thể tính toán được, vì nó yêu cầu chúng ta truy vấn hộp đen R cho từng và mọi kết quả có thể có cho x đầu vào.

Khá nhiều thông tin hơn khi xem xét giới hạn trên của số hạng đầu tiên của phương sai. Hãy để $c_{max} = \max_{y=1,\ldots,C} \|s(y)\|^2 \ll \infty$, mà chúng ta có thể khuyến khích bằng kỹ thuật phạt gradient. Sau đó

$$E\,(R(y) - b(x))^2\|s(y)\|^2 \le c_{max} E\,(R(y) - b(x))^2 . \tag{6.23}$$

2Ta sẽ bỏ |x; θ cho sự ngắn gọn mà không mất đi tính tổng quát.

Đường cơ sở tối ưu để giảm thiểu giới hạn trên ở phía bên tay phải là

$$\nabla b \, \overline{cmax2} \, E \, (R \, (y) - b) \, 2 = -cmax \, (ER \, (y) - b) = 0 \tag{6.24}$$

$$\Longleftrightarrow b* = Ey|x;\theta \, [R(y)] \, . \tag{6.25}$$

Nói cách khác, đường cơ sở tối ưu là phần thưởng kỳ vọng mà chúng ta dự đoán với đầu vào x.

Tất nhiên, đại lượng này một lần nữa khó giải quyết hoặc không thể tính toán chính xác. Tuy nhiên, bây giờ chúng ta có thể phù hợp với một yếu tố dự đoán của $b*$ cho x bằng cách sử dụng tất cả các quan sát trong quá khứ của$(x, R(\tilde{} \, y))$, bởi vì mỗi $R(\tilde{} \, y)$ là một xấp xỉ mẫu đơn với $Ey|x;\theta \, [R(y)]$.3Bởi vì chúng ta cập nhật θ trong quá trình, nhiều mẫu trong quá khứ sẽ không hợp lệ dưới θ hiện tại. Tuy nhiên, nếu chúng ta giả định rằng θ được cập nhật chậm và yếu tố dự đoán được điều chỉnh nhanh chóng, thì tiệm cận đây là một quy trình chính xác, giống như sự phân kỳ tương phản dai dẳng từ §5.1.2.

Sau đó, chúng ta cần duy trì hai yếu tố dự đoán. Một yếu tố dự đoán thường được gọi là mạng chính sách ánh xạ đầu vào hiện tại, hoặc trạng thái, x với sự phân phối trên các đầu ra hoặc hành động có thể xảy ra. Công cụ dự đoán khác thường được gọi là mạng giá trị ánh xạ trạng thái hiện tại x với phần thưởng dự kiến. Sau này được gọi là mạng lưới giá trị, bởi vì nó dự đoán giá trị của trạng thái hiện tại, bất kể hành động được thực hiện bởi chính sách. Các mạng này được đào tạo song song.

Trường hợp phần thưởng ồn ào: một phương pháp phê bình diễn viên. Hãy tưởng tượng rằng phần thưởng R phụ thuộc vào cả x và y và nó cũng là ngẫu nhiên. Nghĩa là, chúng ta chỉ quan sát thấy một ước tính nhiễu của phần thưởng tại x với lựa chọn đầu ra y.Chúng ta có thể muốn sau đó tối đa hóa phần thưởng mong đợi:

$$\max\theta \; Ey|x;\theta E\varepsilon \, [R(y, x; \varepsilon)] \, , \tag{6.28}$$

nơi chúng ta sử dụng ε để gọi chung là bất kỳ loại không chắc chắn nào trong phần thưởngR. Sau đó, chúng ta phải xấp xỉ thêm gradient chính sách với một samplereward$\tilde{} \, R(y, x)$:

$$\nabla\theta Ey|x;\theta E\varepsilon \, [R(y, x; \varepsilon)] \approx (E\varepsilon[R(\tilde{} \, y, x; \varepsilon)] - b(x))\nabla\theta \log p(\tilde{} \, y|x; \theta) \tag{6.29}$$

$$\approx (\tilde{} \, R(\tilde{} \, y, x) - \hat{} \, b(x; \theta b))\nabla\theta \log p(\tilde{} \, y|x; \theta), \tag{6.30}$$

3Đặc biệt, chúng ta nên sử dụng sai số bình phương trung bình làm hàm tổn thất khi lắp apredictive để ước tính phần thưởng mong đợi. Điều này xuất phát từ thực tế là giải pháp tối ưu để giảm thiểu sai số bình phương trung bình tương ứng với việc tính toán trung bình, như dễ thấy bên dưới:

$$\nabla\mu \, \overline{12N} \, \overline{\sum_{n=1}^{N}} \, (\mu - xn)2 = 1N \, \overline{\sum_{n=1}^{N}} \, \overline{n = 1 \, (\mu - xn) = \mu - 1N \sum_{n=1}^{N}} \, xn = 0 \tag{6.26}$$

$$\Longleftrightarrow \mu = 1N \, \overline{\sum_{n=1}^{N}} \, xn. \tag{6.27}$$

trong đó ˆb(x) đề cập đến đường cơ sở dự đoán, ví dụ: giá trị của x. θb là các tham số của hàm giá trị này.

Thật không may, công cụ ước tính này sẽ có thêm một phương sai do phần thưởng ồn ào. Tương tự như những gì chúng tôi đã làm với đường cơ sở ở trên, chúng tôi có thể giảm phương sai bằng cách dự đoán phần thưởng dự kiến tại (x, ˜ y) bằng cách sử dụng một công cụ dự đoán được đào tạo trên các mẫu. Đó là

$$\nabla\theta Ey|x;\theta E\varepsilon\,[R(y,x;\varepsilon)] \approx (\,\hat{}R(\tilde{}y,x;\theta r) - \hat{}b(x;\theta b)\,)\nabla\theta \log p(\tilde{}y|x;\theta), \qquad (6.31)$$

trong đó ˆR là công cụ dự đoán phần thưởng, được tham số hóa bởi θr. Một dự đoán phần thưởng như vậy thường được gọi là giá trị Q của cặp trạng thái-hành động (x, y).4 Sự khác biệt (a) này giữa giá trị Q ˆR và giá trị ˆb được gọi là lợi thế, vì điều này cho chúng ta biết về lợi thế của việc chọn ˜ y so với các đầu ra/hành động khác.

Một quan sát thú vị ở đây là nếu chúng ta có ˆR(y, x; θr) và nếu C, số của tất cả các giá trị y có thể, là nhỏ, chúng ta có thể thay thế mạng giá trị ˆb bằng

$$\hat{}b(x) = Ey|x;\theta\ h\ \hat{}R(y,x;\theta r)i\ , \qquad (6.32)$$

điều này có thể giúp giảm phương sai từ việc phải đào tạo hai pre-dictor phân cách. Với một C hợp lý, điều này có thể được thực hiện khá hiệu quả bằng cách có

ˆR để xuất ra một vectơ giá trị thực chiều C, nhân đầu ra với đầu ra từ dự đoán y (thường được gọi là chính sách) và tổng các giá trị này. Đôi khi chúng ta gọi ˆR này là nhà phê bình và p(y|x; θ) là diễn viên. Do đó, cách tiếp cận này được gọi là thuật toán diễn viên-phê bình.

Học tăng cường nhiều bước Giả sử có C-many tồn tại |X | × |X | ma trận chuyển tiếp ngẫu nhiên Σ(y) sao cho

$$\Sigma ij(y) \geq 0 \quad \text{và} \quad \sum_{i=1}^{C} \Sigma i\cdot(y) = 1, \qquad (6.33)$$

cho y ∈ {1, 2, . . . , C}. Ma trận chuyển tiếp này cung cấp cho chúng ta sự phân phối trên trạng thái tiếp theo cho trạng thái hiện tại xt−1 và hành động đã chọn yt, như

$$q(x = k|xt-1, yt) = \Sigma xt-1,k(yt), \qquad (6.34)$$

trong đó chúng ta giả định X là một tập hợp hữu hạn, mặc dù rất dễ dàng để mở rộng nó đến một không gian trạng thái liên tục X .

Khi định nghĩa toán tử chuyển tiếp q này, chúng ta đã đưa ra một giả định quan trọng được gọi là giả định Markov. Nghĩa là, tại thời điểm t − 1, nơi chúng ta sẽ kết thúc tại thời điểm do sự lựa chọn của tôi về yt là độc lập với các trạng thái trong quá khứ (x1, . . . , xt−2) tôi đã ghé thăm cho đến nay cũng như các lựa chọn hành động (y1, . . . , yt−1) mà tôi đã thực hiện cho đến nay. Chúng tôi tiếp tục

4Mặc dù hầu như không có bài báo nào đề cập rõ ràng đến chữ 'Q' là viết tắt của gì, nhưng người ta thừa nhận rộng rãi rằng nó đại diện cho chất lượng.

giả sử rằng một hàm phần thưởng S⋆ được xác định trên mỗi trạng thái và trả về ascalar, tức là S⋆ : X → R. Mỗi lần chúng ta chuyển từ xt−1 sang xt do yt, chúng ta sẽ nhận được phần thưởng S⋆(xt).

Cùng với một chính sách π(y|x; θ), nó xác định sự phân phối trên quỹ đạo, hoặc thường được gọi là tập. Sau đó, chúng ta có thể lấy mẫu một chuỗi (có khả năng dài vô hạn) của các bộ của trạng thái trước đó xt−1, hành động được chọn yt, trạng thái tiếp theo xt và nhận phần thưởng st = S⋆(xt). Tất nhiên, các bộ này có mối tương quan cao với nhau, vì chúng được thu thập từ một quỹ đạo duy nhất được xác định bởi một tập hợp các phân phối chung, chính sách, quá trình chuyển đổi và phần thưởng. Tuy nhiên, bây giờ chúng tôi sẽ bỏ qua điều này bằng cách nói rằng chúng tôi đang xem xét một bước thời gian cụ thể t từ nhiều quỹ đạo độc lập.

Chúng ta hãy sử dụng n để chỉ từng quỹ đạo này. Để áp dụng policygradient, hoặc thuật toán diễn viên-phê bình, từ trên, chúng ta phải bắt đầu với mạng Q

$\hat{}$Q(xnt−1, ynt). Mạng Q này xấp xỉ chất lượng dự kiến của (xnt−1, ynt).

Chúng tôi xác định chất lượng mong đợi bằng cách trước tiên xác định chất lượng của (xnt−1, ynt) từ quỹ đạo thứ n là

$$\tilde{}Q(xnt-1, ynt) = snt + \sum_{t' = t+1}^{Tn} \gamma t'-tsnt', \tag{6.35}$$

trong đó γ ∈ [0, 1] là cái gọi là hệ số chiết khấu và Tn là độ dài của quỹ đạo thứ n.

Công thức này cho chúng ta biết rằng chất lượng của bất kỳ cặp trạng thái-hành động cụ thể nào được xác định bởi phần thưởng tích lũy từ đó trong suốt quỹ đạo đầy đủ. Bởi vì chúng ta giả định thuộc tính Markov, nên chúng ta hoàn toàn tốt nếu bỏ qua cách chúng ta đạt được (xt−1, yt). Với γ < 0, chúng tôi chỉ rõ rằng chúng tôi không muốn tính đến những gì xảy ra quá xa trong tương lai. Đây thường là một chiến lược tốt để tạo điều kiện học tập trong trường hợp các tập có độ dài hữu hạn, i.e.Tn < ∞ và cần thiết để xác định chất lượng hữu hạn với các tập dài vô hạn, tức là Tn → ∞.[5]

Chất lượng đặc biệt này từ quỹ đạo thứ n có thể được coi là một mẫu từ một biến ngẫu nhiên Q(xnt−1, ynt) được định nghĩa là

$$Q(xnt-1, ynt) = snt + Eq(xt|xt-1,ynt) [\tag{6.36}$$

$$\gamma E\pi(yt+1|xt)q(xt+1|xt,yt+1) [s\star(xt+1)+ \tag{6.37}$$

$$\gamma E\pi(yt+2|xt+1)q(xt+2|xt+1,yt+2) [s\star(xt+2) + \cdots]\square\square . \tag{6.38}$$

Nói cách khác, chất lượng mong đợi là tổng có trọng số của tất cả các phần thưởng trên mỗi bước trong tương lai sau khi gạt ra ngoài lề tất cả các quỹ đạo có thể có trong tương lai theo mô hình chuyển đổi và chính sách.

Khi chúng ta đang làm việc với quỹ đạo có chiều dài hữu hạn, chúng ta có thể dễ dàng đào tạomạng Q để giảm thiểu số lượng sau:

$$\min_{\theta r} \frac{1}{\text{lâ}} \sum_{n=1}^{N} \sum_{t=2}^{Ln} \frac{1}{2} \hat{}R(xnt-1, ynt) - \tilde{}Q(xnt-1, ynt)2 , \tag{6.39}$$

n

[5]Trừ γ < 1, chất lượng dễ dàng phân kỳ, giả sử st > 0 ngay cả khi |st| < ∞.

bởi vì ˆQ là một mẫu không thiên vị được rút ra từ sự phân bố thực sự của chất lượng được xác định ngay ở trên.

Thật không may, điều này là không thể, nếu chúng ta đang làm việc với một tập phim dài vô hạn. Một tập phim dài vô hạn như vậy không phổ biến trong các thiết lập hiện tại, nhưng đó là điều mà chúng tôi mong muốn làm việc trong tương lai, nơi chúng tôi sẽ dự đoán một hệ thống dựa trên học tập sẽ được triển khai trong các tình huống thực tế và tự điều chỉnh một cách nhanh chóng. Tất nhiên, trong trường hợp này, chúng ta phải cập nhật mạng Qcũng nhanh chóng. Thật không may, không thể có được dù chỉ một mẫu˜Q, vì chúng ta không bao giờ thấy phần cuối của bất kỳ tập nào.

Chúng ta hãy sắp xếp lại các thuật ngữ trong Phương trình (6.35):

$$\tilde{}Q(x_{nt-1}, y_{nt}) = s_{nt} + \sum_{t'=t+1}^{T_n} \gamma^{t'-t} s_{nt'} \tag{6.40}$$

$$= s_{nt} + \gamma \left(s_{nt+1} + \sum_{t'=t+2}^{T_n} \gamma^{t'-t} s_{nt'} \right) \tag{6.41}$$

$$= s_{nt} + \gamma \tilde{}Q(x_{nt}, y_{nt+1}).$$

Chúng ta thấy rằng chất lượng được xác định đệ quy:

$$\tilde{}Q(x_{nt-1}, y_{nt}) = s_{nt} + \gamma \tilde{}Q(x_{nt}, y_{nt+1}). \tag{6.42}$$

Điều này cho phép chúng ta viết một hàm mất mát để đào tạo mạng Q mà không cần đợi toàn bộ tập kết thúc (hoặc không bao giờ kết thúc) bằng cách xem xét sự khác biệt thời gian tại thời điểm t:

$$\min_{\theta_r} \frac{1}{n} \sum_{n=1}^{N} \left[\hat{}R(x_{nt-1}, y_{nt}; \theta_r) - \gamma s_{nt} + \hat{}R(x_{nt}, y_{nt+1}; \tilde{}\theta_r) \right]^2 \tag{6.43}$$

$$\Longleftrightarrow \text{phút}_{\theta_r} \frac{c^2}{N} \sum_{n=1}^{N} \left[\frac{1}{C} \hat{}R(x_{nt-1}, y_{nt}; \theta_r) - \hat{}R(x_{nt}, y_{nt+1}; \tilde{}\theta_r) - s_{nt} \right]^2, \tag{6.44}$$

trong đố˜θr là ước tính trước đó của θr. Chúng tôi khởi động từ một số chức năng Q ngẫu nhiên (hoặc ước tính của nó) và lặp đi lặp lại cải thiện ước tính của chúng tôi về hàm Q bằng cách học để dự đoán sự khác biệt về thời gian. Không có gì ngạc nhiên khi chúng tôi gọi loại học tập này là phương pháp khác biệt thời gian [Sutton, 1988].

Hóa ra các phương pháp khác biệt thời gian như vậy có hiệu quả ngay cả khi chúng ta đang đối phó với các giai đoạn có độ dài hữu hạn, khi những giai đoạn này dài. Tuy nhiên, nhìn chung là một thách thức để đào tạo mạng Q với phương pháp chênh lệch thời gian như vậy do nhiều yếu tố. Ví dụ, hàm khách quan ở trên cho chúng ta biết một cách hiệu quả rằng bản thân hàm mục tiêu là một hàm của ước tính trước đó của chúng ta˜θr, có nghĩa là một tối thiểu mà một người tìm thấy bây giờ sẽ không tiếp tục là tối thiểu một khi bạn cắm ước tính mới ˆθr vào˜ ˜θr. Hơn nữa, sẽ mất nhiều thời gian để ước tính chất lượng chụp trong dài hạn

phụ thuộc của việc lựa chọn một hành động cụ thể ynt tại xnt−1 trên nhiều bước sau đó, vì sự khác biệt thời gian ngây thơ chỉ xem xét độ lệch một bước tại một thời điểm. Đã có nhiều cải tiến được đề xuất kể từ công việc ban đầu, nhưng nó nằm ngoài phạm vi của khóa học này để bao gồm những cải tiến đó.

Với mạng Q này (hoặc mạng phê bình, như chúng ta đã học cách gọi nó ở trên),chúng ta có thể dựa vào gradient chính sách để cập nhật chính sách (hoặc mạng tác nhân) từ Eq. (6.31). Tất nhiên có nhiều cách khác nhau để cải thiện actorupdate, chẳng hạn như hạn chế bản cập nhật có phần hạn chế. Một lần nữa, những điều này ít nhiều nằm ngoài phạm vi của khóa học này.

6.2 Phương pháp tổng hợp

Đóng bao. Như chúng ta đã thảo luận nhiều lần trong suốt khóa học (xem ví dụ: §2.4.3,) chúng ta thường ở trong tình huống mà chúng ta không chỉ có một yếu tố dự đoán nhưng có quyền truy cập vào nhiều yếu tố dự đoán khác nhau. Các yếu tố dự đoán này có thể được coi là các mẫu được rút ra từ một số phân phối trên tất cả các yếu tố dự đoán có thể:

$$\tilde{\theta}_n \sim q(\theta). \tag{6.45}$$

Chúng ta sẽ thảo luận về nguồn gốc của một phân phối như vậy sau, nhưng bây giờ, chúng ta sẽ giả định rằng nó tồn tại một cách kỳ diệu và chúng ta có thể dễ dàng rút ra N bộ phân loại từ phân phối q này.

Chúng ta đã xem xét trường hợp có q trước đó trong §2.4.2 khi chúng ta xem xét sự phân rã phương sai-sai lệch sau đây từ Phương trình (2.118):

Ví dụ,y,θ(y − ˆy(x, θ))2 ∝ □□Ey|x (y − µy)2 + Eθ (ˆy(x, θ) − ˆµy)2 + (µy − ˆµy)2 □
Ví dụ | =(z) | =(σ) | =(ε) □
(6.46)

đâu

$$\mu_y = E_{y|x}[y], \tag{6.47}$$
$$\hat{\mu}_y = E_\theta[\hat{y}(x, \theta)]. \tag{6.48}$$

Sự phân hủy này được thực hiện trên tổn thất trung bình trên các yếu tố dự đoán được rút ra từ phân phối hậu q. Thay vào đó, chúng ta có thể xem xét tổn thất được tính toán bằng cách sử dụng dự đoán trung bình từ các yếu tố dự đoán được rút ra từ phân phối sau. Đó là, dự đoán của chúng tôi là

$$\hat{y}(x) = E_\theta[\hat{y}(x, \theta)]. \tag{6.49}$$

Sau đó

$$\text{Ví dụ, } y\,(y - \hat{}y(x))2 \propto \underbrace{Ey|x\,(y - \mu y)2}_{=(a)} \underbrace{-2\,\hat{}y(x)\,Ey|x\,[y]}_{\{z=\hat{}\mu y\}} + \underbrace{\hat{}y2(x)}_{=\hat{}\mu 2y} \quad (6.50)$$

$$\text{Ví dụ} = \underbrace{Ey|x\,(y - \mu y)2}_{=(a')} \underbrace{-\mu 2y + (\hat{}\mu y - \mu y)2}_{|\{z\}=(b')} \quad (6.51)$$

Bây giờ, chúng ta hãy xem xét sự khác biệt giữa hai hàm mất mát này. Vì (a) và (a') tương đương và (c) và (c') tương đương, chúng ta chỉ cần xem xét (b) và (b'):

$$E\theta\,(\hat{}y(x;\theta) - \hat{}\mu y)2 + \mu 2y = E\theta\,\hat{}y2(x;\theta) - 2\hat{}\mu y\,\underbrace{E\theta\,[\hat{}y(x;\theta)]}_{=(a)} + \hat{}\mu 2y + \hat{}\mu 2y = E\theta\,\hat{}y2(x;\theta) \geq 0. \quad (6.52)$$

Nói cách khác, điều này cho chúng ta biết rằng tổn thất trung bình trên các yếu tố dự đoán luôn lớn hơn hoặc bằng tổn thất dự đoán trung bình của các yếu tố dự đoán. Điều này thúc đẩy ý tưởng đóng bao [Breiman, 1996].

Miễn là chúng ta có q, hoặc một bộ lấy mẫu rút ra các yếu tố dự đoán, hoặc các tham số tương ứng, từ phân phối q này, việc đóng bao cho chúng ta biết rằng không bao giờ là một ý tưởng tồi khi sử dụng nhiều yếu tố dự đoán được lấy mẫu đó và tính trung bình các dự đoán của chúng, thay vì sử dụng bất kỳ một trong số chúng một cách trung bình. Hóa ra có nhiều cách khác nhau làm cho yếu tố dự đoán θ của chúng ta trở nên ngẫu nhiên hơn là xác định. Chúng tôi đã đề cập đến hầu hết chúng trước đó trong khóa học, nhưng hãy để chúng tôi xem qua chúng ở đây một lần nữa.

Trong học máy hiện đại, một nguồn chính của tính ngẫu nhiên là việc sử dụng gradient ngẫu nhiên trên hàm mất mát không lồi. Hàm mất mát không lồi với các tham số, vì chúng tôi xếp chồng các khối phi tuyến tính cao để xây dựng một công cụ dự đoán dựa trên mạng nơ-ron sâu, và khi làm như vậy, chúng tôi giới thiệu một mức độ lớn các dư thừa (hoặc mơ hồ). Những sự mơ hồ này ít nhiều được giải quyết tùy ý bởi tính ngẫu nhiên trong độ dốc ngẫu nhiên. Ví dụ, sự lựa chọn của chúng ta về các giá trị ban đầu của các tham số ảnh hưởng đến một không gian con mà trên đó gradient ngẫu nhiên có thể khám phá và tìm ra mức tối thiểu cục bộ. Ngoài khởi tạo, còn có các loại ngẫu nhiên khác trong gradient ngẫu nhiên, tức là cách chúng ta xây dựng các lô nhỏ bằng cách chọn các tập con ngẫu nhiên của tập huấn luyện. Hơn nữa, khá nhiều khối xây dựng vốn dĩ là ngẫu nhiên. Nhớ lại bộ mã hóa tự động biến thiên từ §4.3.1, nơi chúng tôi đưa nhiễu vào xử lý từng trường hợp trong quá trình đào tạo. Nói cách khác, chúng ta có thể nghĩ về giải pháp kết quả bằng cách chạy gradient ngẫu nhiên như một mẫu được rút ra từ một số phân phối được xác định ngầm bởi quá trình học này.

Tất nhiên, một nguồn ngẫu nhiên chính khác là sự lựa chọn của tập huấn luyện. Như chúng ta đã thảo luận trước đó trong §2.4.3, chúng ta có thể bắt chước tính ngẫu nhiên trong việc thu thập dữ liệu ngay cả khi chúng ta có một tập hợp các điểm dữ liệu duy nhất được rút ra từ

phân phối cơ bản theo quá trình lấy mẫu lại bootstrap. Thay vì sử dụng bộ đào tạo như hiện tại, chúng ta có thể lấy mẫu lại nó để phù hợp với kích thước ban đầu của nó bằng cách lấy mẫu lại các ví dụ đào tạo bằng cách thay thế. Mỗi lần, chúng ta sử dụng một tập huấn luyện được lấy mẫu lại khác nhau, chúng ta kết thúc với một giải pháp hơi khác có thể được coi là một mẫu được rút ra từ phân phối một lần nữa được xác định ngầm bởi quá trình xây dựng tập huấn luyện.

Tóm lại, chúng ta nên nắm bắt tính ngẫu nhiên vốn có trong học tập và thu thập dữ liệu để tạo ra một tập hợp các yếu tố dự đoán riêng biệt và tính trung bình các dự đoán của chúng cho mỗi đầu vào. Trung bình, điều này sẽ cung cấp cho chúng ta một công cụ dự đoán tổn thất thấp, nhờ lý thuyết đóng túi, ở trên.

Học máy Bayes. Cuộc thảo luận của chúng ta cho đến nay đã tiến triển với giả định rằng chúng ta được cung cấp phân phối này q(θ). Khi q(θ) được đưa ra, bao cho chúng ta biết rằng chúng ta muốn sử dụng dự đoán trung bình từ nhiều yếu tố dự đoán được lấy mẫu từ q để xây dựng một dự đoán tổn thất thấp hơn trung bình. Tuy nhiên, điều này không cho tôi biết bất cứ điều gì về cách chúng ta có thể tự tạo ra sự phân phối này, hoặc cách phân phối này là gì.

Hóa ra chúng ta có thể dựa vào xác suất để hướng dẫn chúng ta thiết kế cũng như hiểu phân phối q(θ) này. Điều này sẽ giống với những gì chúng tôi đã làm trong §4, và nếu bạn không gặp nhiều khó khăn khi theo dõi phần đó, bạn sẽ không thấy nó khó hiểu. Chúng ta hãy cố gắng suy ra phân phối q này bằng cách trước tiên coi giá trị tổn thất trên tập hợp đào tạo của một yếu tố dự đoán θ duy nhất là hàm năng lượng:

$$e(\theta; D) = \sum_{x \in D} L(x; \theta), \tag{6.53}$$

với một hàm mất mát rất chung L.

Chúng ta có thể giải thích hàm năng lượng này giống như bất kỳ hàm năng lượng nào khác mà chúng ta đã xác định và sử dụng trong suốt học kỳ. Chúng ta muốn phép dự đoán được phân tích bởi θ được gán một giá trị năng lượng thấp khi nó tốt. Độ tốt của yếu tố dự đoán được định nghĩa là hàm tổn thất mà bộ dự đoán này đạt được thấp như thế nào trên tập huấn luyện D.

Bây giờ chúng ta có thể biến hàm năng lượng này thành hàm xác suất bằng cách sử dụng

Công thức Boltzmann, như chúng ta đã làm đi làm lại cho đến bây giờ:

$$\text{Hỏi}(\text{I}|\text{D}, \text{b}) = \frac{\exp -\beta \, \text{P}_{x\in D} \, \text{L}(x; \theta) \text{R}\Theta \exp}{-\beta \, \text{P}_{x\in D} \, \text{L}(x, \theta') \, d\theta'} \tag{6.54}$$

$$= \text{Y}_{x\in D} \frac{\exp(-\beta \text{L}(x; \theta)) \text{R}\Theta \exp -\beta}{\text{P}_{x\in D} \, \text{L}(x, \theta') \, d\theta'} \tag{6.55}$$

$$= \text{Y}_{x\in D} \frac{\exp(-\beta \text{L}(x; \theta)) \text{RX} \exp}{(-\beta \text{L}(x', \theta)) \, dx'} \frac{\text{RX} \exp(-\beta \text{L}(x'; \theta)) \, dx' \text{R}\Theta \exp -\beta}{\text{P}_{x\in D} \, \text{L}(x; \theta') \, d\theta'} \tag{6.56}$$

$$= \text{Y}_{x\in D} \, p(x|\theta, b) \frac{\text{RX} \exp(-\beta \text{L}(x'; \theta)) \, dx' \text{R}}{\text{Tôi} \, \text{RX} \exp(-\beta \text{L}(x'; \theta')) \, dx' d\theta'} \frac{\text{RX} \exp(-\beta \text{L}(x'; \theta')) \, dx' d\theta' \text{R}\Theta}{\exp -\beta \, \text{P}_{x\in D} \, \text{L}(x; \theta') \, d\theta'} \tag{6.57}$$

$$= \text{Y}_{x\in D} \, p(x|\theta, b) \frac{p(\theta) \text{Q}x'\in D}{p(x'|\beta)} . \tag{6.58}$$

Đây chính xác là phân phối hậu trên θ, trong đó chúng ta coi θ là một biến ngẫu nhiên. Nó nói rằng niềm tin của chúng ta (xác suất) về một cấu hình tham số cụ thể θ tỷ lệ thuận với tích của khả năng p(D|θ, β) =Qx∈D p(x|θ, β) và niềm tin trước của θ.

Với niềm tin cập nhật (nghĩa là hậu kỳ) của chúng tôi về θ này, chúng tôi có thể muốn gạt θ ra ngoài lề khi chúng tôi đưa ra dự đoán về một trường hợp mới x′ /∈ D:

$$p(x' \, |\text{D}, \text{b}) = \text{Z}_{\text{Tôi}} \, p(x'|\theta, \beta) q(\theta|\text{D}, \text{b}) \, d\theta. \tag{6.59}$$

Công thức này cho chúng ta biết rằng chúng ta nên lấy mẫu nhiều yếu tố dự đoán theo q(θ|D, β) và trung bình các dự đoán của họ, giống như đóng gói ở trên:

$$p(x' \, |\text{D}, \text{b}) \approx \frac{1}{\text{M}} \sum_{m=1}^{\text{M}} p(x'|\theta\text{m}, \beta), \tag{6.60}$$

trong đó θm ∼ q(θ|D, β). Nói cách khác, nếu chúng ta tuân theo quy tắc Bayes và nghĩ về hàm tổn thất như một hàm năng lượng của tham số θ cho một cá thể riêng lẻ, chúng ta đi đến kết luận rằng chúng ta nên rút ra các phép dự đoán từ phân phối sau q(θ|D, β). Đây là một thuộc tính tuyệt vời, vì bây giờ chúng ta có một hướng dẫn tốt về những gì chúng ta nên làm, mặc dù việc đưa β vào đây là hoàn toàn cố ý, vì nó nói rằng chúng ta vẫn cần một số loại tìm kiếm siêu tham số ngay cả trong cái gọi là học máy Bayes.

Bây giờ chúng ta hãy kết nối phân phối hậu (log-) này với những gì chúng ta đã học được cho đến nay bằng cách viết nó như

$$\text{Nhật ký} \, p(\theta|\text{D}, \text{b}) = \sum_{x\in D} \log p(x|\theta, \beta) + \log p(\theta) - \log \text{Z}(\text{D}, \beta) \tag{6.61}$$

$$= -\beta \sum_{x\in D} \text{L}(x; \theta) + \log p(\theta) - \log \text{Z}'(\text{D}, \beta), \tag{6.62}$$

nơi chúng tôi thu thập tất cả các điều khoản không đổi wrt β vào nhật ký Z′.
Bằng cách đặt β = a |D| , chúng ta kết thúc với− log p(θ|D) = α|D|Xx∈DL(x; θ) − log p(θ)

+hằng số. (6.63)

=− log p(θ|D,a)

Nếu chúng ta giảm thiểu điều này, nghĩa là, nếu chúng ta tối đa hóa log-posterior, đây chính xác là những gì chúng ta đã làm trong suốt thời gian qua. Chúng ta tìm kiếm tham số configu-ration θ để giảm thiểu tổn thất trung bình nhưng sử dụng bộ chính quy để đảm bảo rằng chúng ta kết thúc với một tham số khái quát hóa. Sự cân bằng giữa hai điều này được xác định bởi α không đổi.

Vì chúng ta có thể tính toán chính xác xác suất hậu không chuẩn hóa, chúng ta có thể nghĩ đến việc sử dụng một kỹ thuật lấy mẫu tiên tiến, dựa trên các phương pháp Markov ChainMonte Carlo, từ §5.1.1 [Neal, 1996]. Thật không may, điều này thường quá tốn kém về mặt tính toán, bởi vì chúng ta phải đánh giá tổn thất trên toàn bộ tập D mỗi khi chúng ta đánh giá xác suất chấp nhận. Rốt cuộc, toàn bộ lý do tại sao chúng tôi giới thiệu độ dốc ngẫu nhiên trước đó chính xác là vì quá tốn kém để đánh giá tổn thất trên toàn bộ bộ tập luyện.

May mắn thay, hoặc rõ ràng là khi nhìn lại, các nhà nghiên cứu đã nhận ra rằng sự suy giảm gradient stochas-tic, với một số điều chỉnh hoặc đôi khi không có nhiều quảng cáo, lấy mẫu từ phân bố hậu cụ thể này [xem, ví dụ, Welling và Teh, 2011]. Một ý tưởng chung đằng sau các thuật toán gần đây này, hoặc các phát hiện, là nếu chúng ta không cố gắng giảm ảnh hưởng của nhiễu, tức là (b) trong phương trình (2.87), độ dốc ngẫu nhiên sẽ có xu hướng hướng tới mức tối thiểu cục bộ nhưng sẽ không có xu hướng ở mức tối thiểu cục bộ và nhảy ra hướng tới một mức tối thiểu cục bộ khác. Các cực nhỏ cục bộ này tương ứng với các phương thức phân bố sau. Bằng cách tổng hợp tất cả các cấu hình tham số được truy cập bởi độ dốc ngẫu nhiên hoặc một số tập hợp con của chúng thông qua việc làm mỏng, chúng tôi gọi các mẫu tham số gần đúng theo phân bố sau.

Quan điểm về sự đi xuống gradient ngẫu nhiên như một bộ lấy mẫu sau cho chúng ta biết một giải pháp thay thế nữa để tạo ra một tập hợp các yếu tố dự đoán để đóng bao. Nghĩa là, chúng tôi chỉ đơn giản chạy giảm dốc ngẫu nhiên, mà không ủ tốc độ học về không hoặc trong khi thêm nhiễu bổ sung một cách rõ ràng và thỉnh thoảng thu thập một bộ dự đoán, để tạo thành một tập hợp các yếu tố dự đoán để đóng túi. Cách tiếp cận này giải thích tại sao nó đã thành công trong việc xây dựng một túi mạng nơ-ron sâu để xây dựng một bộ phân loại tổng hợp [Krizhevsky và cộng sự, 2012], bởi vì đó là những mẫu gần đúng từ phân phối sau.

Tăng cường độ dốc. Hãy xem xét một bài toán hồi quy trong đó mục tiêu nằm ∈ Rd và hàm năng lượng được định nghĩa là

$$e([x; y], \theta) = \frac{1}{2} \|y - f(x; \theta)\|2. \tag{6.64}$$

Chúng ta hãy tưởng tượng rằng chúng ta đã có một công cụ dự đoán được đào tạo f (x; θ) không hoàn hảo. Chúng ta muốn phù hợp với một yếu tố dự đoán khác g(x; θ′) để đảm bảo rằng chúng ta có thể

Đưa ra dự đoán tốt hơn trên X. Chúng ta có thể tiếp cận điều này bằng cách trước tiên định nghĩa một dự đoán tổng hợp là

$$h(x; \{tôi, tôi '\}) = f(x; \theta) + \alpha g(x; \theta), \tag{6.65}$$

trong đó α > 0. Sau đó, chúng ta có thể viết hàm năng lượng bao gồm h dưới dạng

$$e'([x; y], \{\theta, \theta'\}) = \|y - h(x; \{tôi, tôi '\})\|2 \tag{6.66}$$

$$= \| (y - f (x; \theta)) \quad -\alpha g(x; \theta')\|2. \tag{6.67}$$

Chúng ta có thể giảm thiểu hàm năng lượng này w.r.t. α và θ′ , dẫn đến g bổ sung cho yếu tố dự đoán hiện có f để giảm thiểu bất kỳ sai số nào còn lại bởi f . Ý tưởng này thường được gọi là tăng cường [Schapire, 1990], vì nó tăng sức mạnh phản đối của các yếu tố dự đoán yếu bằng cách kết hợp hai yếu tố dự đoán yếu, ở đây f và g, để tạo thành một yếu tố dự đoán mạnh hơn. Quy trình này có thể được lặp lại bằng cách coi h là f và đưa một yếu tố dự đoán yếu g khác vào hỗn hợp, cho đến khi đạt được mức độ tổn thất thấp thỏa mãn. Mặc dù chúng ta đã suy ra nó trong ngữ cảnh của một ví dụ duy nhất (x, y), nó nên dễ dàng được mở rộng thành nhiều cặp ví dụ.

Bằng cách kiểm tra cẩn thận (a), chúng ta nhận ra rằng thuật ngữ này là gradient âm của phương trình (6.64) wrt f (x; θ):

$$\partial e \partial f (x; \theta) = -(y - f (x; \theta)). \tag{6.68}$$

Thay vì e, tương đương với tổn thất, vì nó được công thức bằng cách sử dụng khoảng cách L2, chúng ta có thể sử dụng một tổn thất chung chung hơn l(θ; [x, y]). Sau đó, chúng ta có thể viết lại e′ như

$$e'([x; y], \{\theta, \theta'\}) \propto \| - \nabla \hat{y} l(\theta; [x, y]) - \alpha g(x; \theta')\|2, \tag{6.69}$$

trong đó $\hat{y} = f (x; \theta)$. Bằng cách giảm thiểu e ′ wrt θ ′ và α, chúng tôi để g nắm bắt hiệu quả độ dốc âm (chia tỷ lệ) của tổn thất wrt $\hat{y}$.

Như chúng ta đã học được nhiều lần trong suốt khóa học cho đến nay, độ dốc chỉ có ý nghĩa ở một khu vực lân cận nhỏ nào đó. Nói cách khác, thực hiện toàn bộ bước theo hướng gradient âm có thể không nhất thiết làm giảm tổn thất tổng thể và chúng ta phải chia tỷ lệ gradient cho phù hợp. Do đó, chúng tôi tìm kiếm kích thước bước phù hợp bằng cách giải

$$\min \gamma \geq 0 \; l(\{\theta, \theta'\}; [x, y]), \tag{6.70}$$

trong đó tổn thất l được tính bằng cách so sánh y và

$$\hat{y} = f (x; \theta) + \gamma g(x; \theta'). \tag{6.71}$$

Quy trình này giống với quá trình giảm độ dốc từ §2.3.2, và do đó được gọi là tăng cường độ dốc [Friedman, 2001].

Tăng cường không chỉ định cách ước tính β và g (hoặc θ ′ tương đương) ở mỗi lần lặp lại và tùy thuộc vào người học để quyết định người học nào (yếu) g họ sử dụng và họ chọn chức năng mất mát nào. Các lựa chọn phổ biến bao gồm cây quyết định và máy vectơ hỗ trợ dựa trên hạt nhân. Theo nghĩa này, đây không phải là một thuật toán học tập mà là một meta-heuristics.

6.3 Meta-Learning

Trong phần §6.2 trước, chúng ta đã học được rằng sẽ là một ý tưởng hay để tính trung bình các dự đoán từ nhiều mô hình nếu chúng ta có một phân phối q (θ) trên các mô hình (hoặc các yếu tố dự đoán) thay vì một yếu tố dự đoán duy nhất. Sau đó, chúng ta biết được rằng học máy Bayes cho chúng ta biết rằng sự phân phối này nên được điều chỉnh trên tập huấn luyện, kết quả là q(θ|D), và chúng ta có thể có được sự phân bố hậu này theo quy tắc Bayes:

$$\text{Hỏi}(I|D) \propto p\ (\theta)\ Y_{x\in D}\ p(x|\theta). \tag{6.72}$$

Tại thời điểm này, một câu hỏi công bằng là liệu chúng ta có phải tuân theo quy tắc cụ thể này dựa trên quy tắc của Bayes hay không. Có lẽ có một cách tốt hơn để ánh xạ tập huấn luyện D với phân phối hậu trên θ.

Let us assume that we have not one but multiple training set □D1, D2, . . . , DM ,corresponding to the M prediction tasks. Đối với mỗi tập huấn luyện, chúng ta có thể định nghĩa cái gọi là tổn thất xác nhận chéo K-fold là

$$\text{LKCV}(\varphi; Dm) = -\frac{1}{K}\sum_{k=1}^{K} \sum_{x\in Dm\sigma k(1):\sigma k(\lceil \frac{1}{K}|DM|\rceil)} \text{nhật}^{V\acute{\sigma}} p(x|\theta)q\ \theta|DMSK(\lceil \frac{1}{K}|Dm|\rceil+1):\sigma k(|dm|);\ \Phi\ D\theta, \tag{6.73}$$

trong đó σk là hoán vị thứ k của các chỉ số từ 1 đến |DM|. Để tính toán tổn thất này, chúng tôi thường phân vùng dữ liệu Dm thành các phân vùng K. Đối với mỗi phân vùng, chúng tôi sử dụng phần còn lại của các phân vùng để đào tạo một dự đoán (hoặc một tập hợp các yếu tố dự đoán) và sử dụng công cụ dự đoán này để tính toán tổn thất. Chúng tôi tính trung bình các giá trị tổn thất K này và sử dụng nó như một đại diện cho tổn thất tổng quát [Kohavi, 1995].

Trong trường hợp cụ thể ở trên, tổn thất xác nhận chéo này là một hàm φ tham số hóa phân phối hậu q trên các tham số θ. Tham số này biến bài toán suy luận hậu trong máy học Bayes thành việc xây dựng một công cụ dự đoán ánh xạ một tập hợp các điểm dữ liệu đào tạo thành phân phối trên các tham số, trong đó công cụ dự đoán này được tham số hóa bằng cách sử dụng θ. Nói cách khác, chúng tôi đào tạo một công cụ dự đoán giải quyết vấn đề suy luận hậu bằng cách giải quyết

$$\text{tối thiểu}_{\varphi}\ \frac{1}{\text{tri ệu}}\sum_{m=1}^{M} \text{LKCV}(\varphi; Dm). \tag{6.74}$$

In this case, we would call □D1, . . . , DM một bộ đào tạo meta.
Cũng giống như những gì chúng ta đã thấy trước đó trong §4, tổn thất xác nhận chéo gấp K này không dễ tính toán cũng như giảm thiểu. Thay vào đó, chúng ta có thể sử dụng kỹ thuật tương tự từ suy luận biến thiên từ trước đó để giảm thiểu giới hạn trên thành LKCV:

$$\text{LKCV}(\varphi; Dm) \leq -\frac{1}{K}\sum_{k=1}^{K} \sum_{x \in Dm\sigma k(1):\sigma k(\lceil \frac{1}{K}|DM| \rceil)} \frac{1}{B}\sum_{b=1}^{B} \text{nhật ký } p(x|\theta b), \quad (6.75)$$

trong đó θb ∼ q θ|Dmσk(⌈ 1K |Dm|⌉+1):σk(|dm|); ϕ□. Vì θ thường liên tục, chúng ta có thể tính toán độ dốc của LKCV w.r.t. φ với thủ thuật reparametrization, miễn là q có thể vi phân w.r.t. φ.

Điều này rất thú vị, vì chúng ta có thể linh hoạt về cách chúng ta tham số q và q này được tối ưu hóa trực tiếp để dẫn đến một phân phối trên θ hoặc một tập hợp của θ mà theo đó tổn thất dự đoán là tối thiểu. Nói cách khác, q là một thuật toán học tập và chúng ta đang đào tạo một thuật toán học tập bằng cách giảm thiểu hàm siêu mục tiêu trong Eq. (6.74) wrt q.

Ví dụ, chúng ta có thể định nghĩa q một cách ngầm bằng cách vẽ một mẫu của parame-ters θ từ q chỉ bằng cách sử dụng một vài bước N của gradient ngẫu nhiên, trái ngược với việc chạy nó cho đến khi hội tụ như từ §2.3.2. Khi làm như vậy, chúng ta có thể xem xétkhởi tạo θ0 của các tham số là φ. Bằng cách giảm thiểu chức năng siêu mục tiêu wrt θ0, chúng tôi đang tìm kiếm sự khởi tạo của các tham số tối ưu với N bước SGD. Nếu tập huấn luyện mới sau meta-learning như vậy tương tự như các tập meta-training, chúng ta hy vọng rằng N bước SGD sẽ đủ nếu không muốn nói là tối ưu để có được dự đoán tốt nhất. Cách tiếp cận này ban đầu được đề xuất bởi [Finn et al., 2017] và được gọi là meta-learning bất khả tri mô hình.

Tất nhiên, chúng ta hoàn toàn có thể từ bỏ bất kỳ tối ưu hóa lặp đi lặp lại nào khi hủy ký q và xây dựng một công cụ dự đoán ánh xạ trực tiếp một tập hợp các điểm dữ liệu đào tạoD với dự đoán trên một quan sát mới x′. Khi làm như vậy, điều quan trọng là phải nhận ra rằng công cụ dự đoán này không thể chỉ đơn giản lấy làm đầu vào D mà cần mô hình hóa nhiễu trong chính việc học. Điều này tự nhiên đòi hỏi phải bao gồm các biến tiềm ẩn z vào bộ dự đoán này, giống như cách chúng ta đã làm trước đó với các mô hình tổng quát trong §4. Trong trường hợp này, phân phối hậu q (θ) là ngầm và chúng ta trực tiếp dự đoán xác suất dự đoán bằng cách

$$p(x|D; \varphi) = \int_{\text{Với}}^{\text{Với}} p(x|z; \varphi x) p_z(z|D; \varphi z)\, dz, \quad (6.76)$$

trong đó pz là trước trên z và chúng tôi gạt ra ngoài lề z. Cách tiếp cận này thường được gọi là các quá trình thần kinh [Garnelo và cộng sự, 2018]. Bởi vì sự gạt ra ngoài lề này thường khó giải quyết, nên việc tiếp cận nó từ suy luận và học tập biến thiên mà chúng ta đã học được trong §4.3.1.

Nhìn chung, các phương pháp này được gọi là meta-learning, vì một quy trình như vậy dẫn đến một công cụ dự đoán biết cách học cách giải quyết một vấn đề với các ví dụ mới. Sau đó, meta-learning có thể được sử dụng để giải quyết không chỉ các vấn đề học tập mà còn bất kỳ loại vấn đề tập hợp nào, chẳng hạn như khám phá nhân quả và các vấn đề suy luận thống kê. Đây là một lĩnh vực nghiên cứu thú vị và tích cực.

6.4 Hồi quy: Mạng mật độ hỗn hợp

Giả sử e([x, y], θ) là hàm năng lượng trong đó y không được phân loại với một số lượng nhỏ các loại.6 Không mất tính tổng quát, hãy để y ∈ Rd. Chúng ta có thể biến điều này thành một hàm mật độ xác suất bằng cách

$$p(y|x;\theta) = \frac{\exp\left(-e([x,y];\theta)\right)}{\int_{R^d} \exp\left(-e([x,y'];\theta)\right) dy'}. \tag{6.77}$$

Không giống như bài toán phân loại mà chúng ta đã thấy trong Phương trình (2.30), nói chung, việc tính toán hằng số chuẩn hóa trong trường hợp này với y không phân loại là cực kỳ khó khăn. Trên thực tế, vấn đề này giống hệt với các mô hình đồ họa không định hướng, chẳng hạn như các máy Boltzmann bị hạn chế từ §5.1, đòi hỏi MCMCsampling tốn kém [Boulanger-Lewandowski et al., 2012].

Do đó, việc xem xét tham số hóa của hàm năng lượng sao cho hằng số chuẩn hóa tự động là 1. Chúng tôi đã xem xét một cách tiếp cận cụ thể theo mô hình này trước đó trong §4.3.1. Với một biến tiềm ẩn z (một biến không quan sát), chúng ta có thể làm cho nó dễ dàng chuẩn hóa:

$$p(y|x;\theta) = \underset{\text{Với}}{\text{Với}}\, p(z) \frac{\exp(-e([x,y],z,\theta))}{\exp(-e([x,y'],z,\theta)dy'\,dz.} \tag{6.78}$$

Nếu chúng ta chọn tham số hóa sau đây của hàm năng lượng, chúng ta biết cách tính toán chính xác hằng số chuẩn hóa, bởi vì chúng ta kết thúc với phân phối Gaussian trên y cho x và z:

$$e([x,y],z,\theta) = \frac{1}{2} \|y - \mu(x,z;\theta)\|2. \tag{6.79}$$

Thật không may, cách tiếp cận này cũng không tầm thường, vì chúng ta phải gạt ra bên lề biến tiềm ẩn z. Vấn đề gạt ra bên lề này nói chung không dễ giải quyết và chúng ta thường cần phải sử dụng một cách tiếp cận gần đúng, chẳng hạn như suy luận biến thiên [Chung và cộng sự, 2015].

Bởi vì y thường có chiều thấp hơn x , có một giải pháp thay thế dễ giải quyết cho hai cách tiếp cận khó giải quyết này. Cách tiếp cận này hạn chế cách tiếp cận biến tiềm ẩn ở trên để |Z | ≪ ∞ , nghĩa là, z có thể lấy một trong một số giá trị có thể, tức là Z = {1, 2, . . . , K}. Trong trường hợp đó, chúng ta có thể giải quyết chính xác vấn đề gạt ra bên lề và đi đến

$$p(y|x;\theta) = \sum_{z=1}^{K} \frac{1}{K} N\, y;\, \mu z(x),\, \sigma 2z(x)I, \tag{6.80}$$

Giả

$$e([x,y],z,\theta) = \frac{12\sigma 2z\,(F\,(x;\theta F\,);\, \theta\sigma)\, \|y - \mu z(F\,(x;\theta F);}{\theta\mu)\|2\,.} \tag{6.81}$$

Biến phân loại lấy một giá trị từ một số lượng nhỏ các giá trị có thể được xác định trước, giống như phân loại.

Đây chính xác là một hỗn hợp của Gaussian, tuy nhiên có điều kiện trên x. F (x; θF) là một bộ trích xuất tính năng của đầu vào x, và bộ trích xuất này được chia sẻ giữa μ và σ2. Chúng tôi tiếp tục giả định rằng trước đó trên các thành phần hỗn hợp là đồng nhất, tức là p (z) = 1K . Điều này không hoàn toàn cần thiết, nhưng chỉ đơn giản là làm cho việc học dễ dàng hơn, vì chúng tôi loại bỏ bất kỳ tham số bổ sung nào để tính toán trước khỏi x đầu vào. Miễn là μz và σ2z có thể phân biệt được w.r.t. θμ, θσ và θF (nói chung, bao gồm θ,) chúng ta có thể đào tạo tất cả bộ dự đoán này cùng nhau mà không cần phải dựa vào một số gạt gần đúng bằng cách

$$\min_{\theta F,\theta\mu,\theta\sigma} -\frac{1}{N}\sum_{n=1}^{N}\log\sum_{z=1}^{K}\frac{1}{K}\,\mathcal{N}(y_n;\mu_z(x_n),\sigma^2_z(x_n)). \tag{6.82}$$

Một yếu tố dự đoán như vậy được gọi là mạng lưới mật độ hỗn hợp và vượt xa tất cả các cách tiếp cận khác ở trên [Bishop, 1994].

Một trường hợp đặc biệt chính của mạng mật độ hỗn hợp này là khi chỉ có một thành phần hỗn hợp, tức là K = 1. Trong trường hợp đó, điều này giảm thành hồi quy tuyến tính gia đình hơn với hàm mất sai số bình phương trung bình, giả địnhphương sai không đổi, tức là σ2z (x) = c. Mặc dù đây là một cách tiếp cận thông thường và cũng là những gì chúng tôi đã làm trước đó khi chúng tôi suy ra sự lan truyền ngược trong §2.2.2, cách tiếp cận này của một thành phần hỗn hợp đơn lẻ có một nhược điểm lớn là chỉ có thể có một chế độ duy nhất trong phân phối dự đoán. Điều này đặc biệt có vấn đề khi phân phối thực cơ bản có nhiều chế độ cục bộ, vì việc học với tiêu chí trên sẽ làm cho phân phối dự đoán này được phân tán để bao phủ tất cả các phương thức của phân phối thực, dẫn đến dự đoán không chắc chắn không cần thiết với khối lượng xác suất tập trung vào một vùng của không gian đầu ra có liên quan đến vị trí của các chế độ thực. Bằng cách tăng K vượt quá 1, chúng ta tăng cơ hội nắm bắt được sự không chắc chắn vốn có trong hồi quy.

Mặc dù việc đào tạo có thể được thực hiện chính xác, nhưng điều này không có nghĩa là chúng ta có thể thực hiện dự đoán một cách dễ dàng với mạng mật độ hỗn hợp. Trừ khi K = 1, không có giải pháp phân tích cho

$$\hat{y}(x) = \arg\max_{y\in\mathbb{R}^d} \log\sum_{z=1}^{K}\mathcal{N}(y;\mu_z(x),\sigma^2_z(x)) - \log K. \tag{6.83}$$

Chúng ta có thể giải quyết vấn đề này bằng cách đi xuống gradient sẽ tìm thấy một trong nhiều Kmode của phân bố phức này hoặc tìm một điểm yên ngựa.

Tuy nhiên, việc trả về một ước tính điểm duy nhất của giải pháp là không thỏa đáng, khi chúng ta đào tạo công cụ dự đoán của mình để nắm bắt sự phân phối đầy đủ trên không gian đầu ra. Thay vào đó, có thể mong muốn trả về một tập hợp các giá trị có thể có của kết quả y nằm trong một vùng đáng tin cậy, theo quy trình từ §2.4.3.Điều này đặc biệt mong muốn, vì chúng ta có thể dễ dàng rút ra càng nhiều mẫu độc lập từ hỗn hợp Gaussian. Khi các mẫu {y1, . . . , yM } được vẽ, chúng tôi chấm điểm mỗi mẫu với mạng mật độ hỗn hợp, một lần nữa là tầm thường, dẫn đến {p1, . . . , pM }. Sau đó, chúng ta có thể phù hợp với một hàm mật độ tích lũy trên các điểm số này và chỉ chọn những điểm trên ngưỡng được xác định trước. Những đầu ra được chọn này có thể được coi là một tập hợp đầu ra đáng tin cậy cho x.

6.5 Mối quan hệ nhân quả

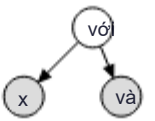
Một hạn chế lớn của tất cả các phương pháp trong ghi chú bài giảng này, có lẽ ngoại trừ việc học tập củng cố trong §6.1, là tất cả chúng đều dựa gần như hoàn toàn vào sự liên kết, hoặc tương quan. Tất cả các thuật toán này đều tìm kiếm các mẫu nào xuất hiện cùng với các mẫu khác thường xuyên trong một tập dữ liệu nhất định. Ngay trong §2.2.2, điều này đã được thể hiện rõ ràng. Ví dụ, hãy nhớ lại quy tắc cập nhật sau đây cho một khối tuyến tính trong Phương trình (2.53):

$$\frac{\partial}{\partial} uij = xihj - xi\hat{}hj, \tag{6.84}$$

trong đó chúng ta giả định không có tính phi tuyến, tức là h'j = 1. Thuật ngữ đầu tiên làm giảm giá trị của uij về phía nguồn gốc 0 nếu xi và giá trị cũ, không mong muốn của tế bào thần kinh ẩn thứ j có cùng dấu hiệu.7 Mặt khác, số hạng thứ hai làm tăng giá trị của uij ra khỏi nguồn gốc nếu xi và ˆhj mới, mong muốn có cùng dấu hiệu. Nói cách khác, uij, một trong nhiều tham số của bộ dự đoán này, mã hóa mức độ tương quan giữa chiều thứ i của quan sát và chiều thứ j của biến ẩn với nhau.

Điều này hoàn toàn ổn, nếu mục tiêu là nắm bắt các mối tương quan như vậy và sử dụng chúng để gán các giá trị bị thiếu, chẳng hạn như đầu ra liên quan đến các quan sát thời gian thử nghiệm. Tuy nhiên, điều này là không đủ nếu chúng ta muốn suy ra mối quan hệ nhân quả giữa các biến, bởi vì như chúng ta thường nói một cách ngẫu nhiên, "tương quan không ngụ ý nhân quả". 8

Chúng ta hãy đào sâu hơn một chút vào tuyên bố này và xem xét một vài trường hợp trong đó mối tương quan tồn tại nhưng nhân quả thì không. Trường hợp đầu tiên là khi tồn tại một nhiễu không quan sát được, trong đó nhiễu z được định nghĩa để ảnh hưởng đến cả đầu vào x và kết quả y, sao cho



Cả x và y đều do nhiễu z không quan sát được này trong sơ đồ này và chúng ta có thể viết ra phân phối cận biên trên (x, y) như

$$p(x, y) = \text{Với}_{i} p(x|z)p(y|z)p(z)dz. \tag{6.85}$$

Tương đối đơn giản để thấy rằng điều này sẽ không được phân tích thành tích của p(x) và p(z), tức là.

$$\text{Với}_{i} p(x|z)p(y|z)p(z)dz \neq p(x)p(z), \tag{6.86}$$

7Chúng ta đang đi theo hướng ngược lại với hướng gradient.8 Khi chúng ta nói điều này, chúng ta đang đề cập đến sự phụ thuộc bằng tương quan, nhưng trừ khi nó gây nhầm lẫn về mặt kỹ thuật, tôi sẽ sử dụng tương quan và phụ thuộc thay thế cho nhau trong phần này.

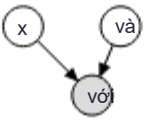
trừ khi

$$\text{Với} \int p(x|z)p(y|z)p(z)dz = p(x) \quad \text{Với} \int p(y|z)p(z)dz, \tag{6.87}$$

Điều này ngụ ý rằng không có cạnh nào đi từ z đến x ngay từ đầu.
Việc chúng ta không thể tính p(x, y) vào tích của các biên của x andy ngụ ý rằng x và y phụ thuộc vào nhau. Tương tự, chúng ta có thể nói rằng x và y tương quan với nhau (có khả năng phi tuyến.) Tuy nhiên, chúng không liên quan đến nhau về mặt nhân quả, vì can thiệp vào x sẽ không gây ra bất kỳ thay đổi nào trong y và ngược lại.

Một ví dụ về trường hợp gây nhiễu không quan sát được này có thể được tìm thấy trong việc lái xe. Nếu một người không biết cách lái xe hoạt động và chỉ nhìn vào bảng điều khiển của acar,9 có thể dễ dàng nhận thấy rằng đèn báo rẽ và góc vô lăng có mối tương quan cao với nhau, điều này có thể dẫn đến kết luận nhân quả không chính xác rằng đèn báo rẽ khiến vô lăng quay hoặc ngược lại. Điều này thiếu một kẻ gây nhiễu lớn là một người lái xe và ý định của họ để quay xe.

Trường hợp thứ hai là cái mà chúng ta thường gọi là thiên kiến xác nhận. Hãy xem xét mô hình nhân quả sau:



Trong trường hợp này, x và y độc lập với nhau một tiên nghiệm. Rõ ràng là chúng không liên quan đến nhau về mặt nhân quả, vì việc đặt thủ công một trong những điều này thành một giá trị cụ thể sẽ không thay đổi giá trị của biến kia. Tuy nhiên, thật thú vị khi quan sát thấy rằng hai biến này, x và y, đột nhiên phụ thuộc vào nhau, một khi chúng ta quan sát z. Nghĩa là, theo phân phối sau, x và y không độc lập:

$$p(x, y|z) = \frac{p(x)p(y)p(z|x, y)R}{p(x')p(y')p(z|x', y')dx'dy'}. \tag{6.88}$$

Do p(z|x, y), chúng ta không thể phân tích p(x, y|z) vào tích của hai hạng, mỗi số hạng chỉ phụ thuộc vào x hoặc y. Nếu chúng ta có thể, điều đó có nghĩa là z được gây ra bởi một trong x hoặc y (hoặc không phải cả hai.) Đầu vào và kết quả có tương quan trong trường hợp này, bởi vì chúng ta chỉ xem xét một tập hợp con của các cặp (x, y) được liên kết với một giá trị cụ thể của z. Do đó, điều này còn được gọi là thiên kiến chọn lọc.

Chúng ta hãy xem xét một ví dụ, trong đó x tương ứng với một vụ trộm và y tương ứng với một trận động đất. z là báo động nhà. Báo động nhà kêu (z = 1) khi có trộm (x = 1) hoặc có động đất (y = 1). Nó khá an toàn cho

9Hãy tưởng tượng bạn đang thu thập dữ liệu từ xe để xây dựng một mô hình tự lái.

bây giờ để giả định rằng khả năng trộm cắp và động đất là khá độc lập với nhau. Tuy nhiên, nếu bạn nghe thấy báo thức của bạn đã kêu, nghĩa là, nếu bạn điều kiện z = 1, trộm cắp và động đất không còn độc lập nữa, vì tôi sẽ có thể giải thích khả năng trộm cắp nếu bản thân tôi cảm thấy động đất. Đó là, khả năng động đất và trộm cắp xảy ra cùng nhau và kích hoạt báo động là bao nhiêu. Mặc dù không có mối liên hệ nhân quả giữa trận động đất và trộm cắp, nhưng chúng hiện có mối tương quan tiêu cực với nhau vì chúng ta có điều kiện báo động.

Những trường hợp này nhấn mạnh sự khác biệt giữa liên kết (tương quan) và nhân quả. Để nắm bắt các mối quan hệ nhân quả giữa các biến và sử dụng chúng để kiểm soát hệ thống cơ bản, chúng ta phải sử dụng thêm một tập hợp các giả định và công cụ để loại trừ các liên kết không nhân quả, hoặc cái gọi là tương quan giả. Một khi chúng ta được trang bị các công cụ như vậy, chúng ta có thể làm cho máy học mạnh mẽ hơn và các kịch bản thực tế hơn, ví dụ như sự phân bố mà từ đó các quan sát được rút ra thay đổi giữa thời gian đào tạo và kiểm tra. Đây là một chủ đề hấp dẫn trong học máy và rộng hơn là trí tuệ nhân tạo, nhưng nằm ngoài phạm vi của khóa học này. Tôi khuyên bạn nên xem ghi chú bài giảng của tôi "Giới thiệu ngắn gọn về suy luận nhân quả trong học máy" [Cho, 2024] và sau đó chuyển sang các tài liệu chuyên sâu hơn về suy luận nhân quả, khám phá nhân quả và học biểu diễn nhân quả.

Thư mục

DH Ackley, GE Hinton và TJ Sejnowski. Thuật toán học tập cho máy boltzmann. Khoa học nhận thức, 9(1):147–169, 1985.

JL Ba, JR Kiros và GE Hinton. Chuẩn hóa lớp. Bản in trước arXiv arXiv:1607.06450, 2016.

D. Bahdanau, K. Cho và Y. Bengio. Dịch máy thần kinh bằng cách cùng học cách sắp xếp và dịch. Trong Hội nghị Quốc tế về Đại diện Học tập, 2015.

AG Baydin, BA Pearlmutter, AA Radul và JM Siskind. Tự động Sự khác biệt trong học máy: Một cuộc khảo sát. Tạp chí Nghiên cứu Học máy, 18(153):1–43, 2018. URL http://jmlr.org/papers/v18/17-468.html.

Y. Bengio, P. Simard và P. Frasconi. Học các phụ thuộc lâu dài với Xuống dốc rất khó. Giao dịch IEEE trên mạng nơ-ron, 5(2):157–166, 1994.

J. Bergstra, R. Bardenet, Y. Bengio và B. K'egl. Thuật toán cho siêu - tối ưu hóa tham số. Trong J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, và K. Weinberger, biên tập viên, Những tiến bộ trong hệ thống xử lý thông tin thần kinh, tập 24. Curran Associates, Inc., 2011.URL https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

CM Giám mục. Mạng mật độ hỗn hợp. Báo cáo kỹ thuật, Điện toán thần kinh Nhóm nghiên cứu, Đại học Aston, Birmingham, Vương quốc Anh, 1994.

N. Boulanger-Lewandowski, Y. Bengio và P. Vincent. Mô hình hóa thời gian dependencies trong chuỗi chiều cao: Ứng dụng cho âm nhạc đa âm. Trong Kỷ yếu của Hội nghị Quốc tế lần thứ 29 về Học máy (ICML-12), trang 1159–1166. ICML, 2012.

L. Breiman. Các yếu tố dự đoán đóng túi. Học máy, 24(2):123–140, 1996.

JS Dây cương. Giải thích xác suất của mạng phân loại feedforward đầu ra, với mối quan hệ với nhận dạng mẫu thống kê. Trong Neurocom-putting: Thuật toán, kiến trúc và ứng dụng, trang 227–236. Springer 1990.

TB Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Nee-lakantan, P. Shyam, G. Sastry, A. Askell, và cộng sự. Các mô hình ngôn ngữ là những người học ít bắn. Những tiến bộ trong hệ thống xử lý thông tin thần kinh, 33: 1877–1901, 2020.

K. Cho. Hiểu ngôn ngữ tự nhiên với biểu diễn phân tán. arXiv bản in trước arXiv: 1511.07916, 2015.

K. Cho. Giới thiệu ngắn gọn về suy luận nhân quả trong học máy. arXiv bản in trước arXiv: 2405.08793, 2024.

K. Cho, B. van Merri¨enboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, và Y. Bengio. Học các biểu diễn cụm từ bằng cách sử dụng RNNencoder-decoder để dịch máy thống kê. Trong Kỷ yếu của Hội nghị năm 2014 về các phương pháp thực nghiệm trong xử lý ngôn ngữ tự nhiên (EMNLP), trang 1724–1734, Doha, Qatar, 2014. Hiệp hội ngôn ngữ học tính toán. URL https://aclanthology.org/D14-1179.

J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville và Y. Bengio. Tái phát mô hình biến tiềm ẩn cho dữ liệu tuần tự. Trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, trang 2980–2988, 2015.

C. Cortes. Mạng vectơ hỗ trợ. Học máy, 1995.

J. Duchi, E. Hazan và Y. Singer. Phương pháp gradient phụ thích ứng cho trực tuyến Học tập và tối ưu hóa ngẫu nhiên. Tạp chí nghiên cứu học máy, 12(7), 2011.

C. Finn, P. Abbeel và S. Levine. Meta-learning bất khả tri mô hình để thích ứng nhanh chóng-Tion. Hội nghị quốc tế về Học máy, trang 1126–1135, 2017.

J. H. Friedman. Xấp xỉ hàm tham lam: Một gradient tăng cường ma-Chine. Biên niên sử thống kê, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451. URL https://projecteuclid.org/euclid.aos/1013203451.

M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, DJ Rezende, S. Eslami, và Y. W. Teh. Các quá trình thần kinh. bản in sẵn arXiv arXiv: 1807.01622, 2018.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, và Y. Bengio. Mạng lưới đối nghịch sinh sản. Trong Nâng cao hệ thống xử lý thông tin thần kinh, trang 2672–2680, 2014.

A. Gretton, KM Borgwardt, MJ Rasch, B. Sch¨olkopf và A. Smola. Một thử nghiệm hai mẫu hạt nhân. Tạp chí Nghiên cứu Học máy, 13 (Mar): 723–773, 2012.

W. K. Hastings. Phương pháp lấy mẫu Monte carlo sử dụng chuỗi Markov và Ứng dụng. Sinh học, 57(1):97–109, 1970. doi: 10.1093/biomet/57.1.97.

K. He, X. Zhang, S. Ren và J. Sun. Học dự sâu để nhận dạng hình ảnh-Tion. Trong Kỷ yếu của hội nghị IEEE về thị giác máy tính và nhận dạng mẫu, trang 770–778, 2016.

D. O. Hebb. Tổ chức hành vi: Lý thuyết tâm lý thần kinh. Wiley & Con trai, New York, 1949.

GE Hinton. Đào tạo sản phẩm của các chuyên gia bằng cách giảm thiểu thợ lặn tương quan-gence. Tính toán thần kinh, 14(8):1771–1800, 2002.

D. Hjelm, RR Salakhutdinov, K. Cho, N. Jojic, V. Calhoun và J. Chung. Tinh chỉnh lặp đi lặp lại của hậu gần đúng cho các mạng niềm tin có định hướng. Những tiến bộ trong hệ thống xử lý thông tin thần kinh, 29, 2016.

H. Khách sạn. Phân tích một phức biến thống kê thành kết quả chính ponents. Tạp chí tâm lý giáo dục, 24(6):417, 1933.

A. Ilin và T. Raiko. Các phương pháp tiếp cận thực tế để phân tích thành phần chính trong sự hiện diện của các giá trị bị thiếu. Tạp chí Nghiên cứu Học máy, 11: 1957–2000, 2010.

S. Ioffe và C. Szegedy. Chuẩn hóa hàng loạt: Tăng tốc đào tạo mạng sâu bằng cách giảm sự dịch chuyển hiệp biến bên trong. Kỷ yếu của Hội nghị Quốc tế lần thứ 32 về Học máy, 37: 448–456, 2015.

ET Jaynes. Lý thuyết thông tin và cơ học thống kê. Đánh giá vật lý, 106(4):620, 1957.

DR Jones, M. Schonlau và WJ Welch. Tối ưu hóa toàn cầu hiệu quả các chức năng hộp đen đắt tiền. Tạp chí Tối ưu hóa Toàn cầu, 13(4):455–492, 1998. doi: 10.1023 / A: 1008352424803.

DP Kingma và J. Ba. Adam: Một phương pháp tối ưu hóa ngẫu nhiên. arXiv bản in trước arXiv: 1412.6980, 2014.

DP Kingma và M. Welling. Tự động mã hóa các baye biến thể. Bản in trước arXiv arXiv:1312.6114, 2013.

R. Kohavi. Nghiên cứu về xác nhận chéo và bootstrap để ước tính độ chính xác và lựa chọn mô hình. Ijcai, 14(2):1137–1145, 1995.

A. Krizhevsky, I. Sutskever và GE Hinton. Phân loại Imagenet với chiều sâu mạng nơ-ron tích chập. Trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, trang 1097–1105, 2012.

Y. LeCun, L. Bottou, Y. Bengio và P. Haffner. Ứng dụng học tập dựa trên gradient-plied đến nhận dạng tài liệu. Kỷ yếu của IEEE, 86(11):2278–2324, 1998.

Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. Huang, và cộng sự. Hướng dẫn về học tập dựa trên năng lượng. Dự đoán dữ liệu có cấu trúc, 1(0), 2006.

DA McAllester. Một số định lý PAC-Bayes. Trong Kỷ yếu của Thứ Mười Hai Hội nghị thường niên về Lý thuyết Học tập Tính toán, trang 230–234. ACM 1999.

RM Neal. Lai monte carlo. Báo cáo kỹ thuật CRG-TR-93-1, Bộ phận Khoa học Máy tính, Đại học Toronto, 1993.

RM Neal. Học Bayes cho mạng nơ-ron, tập 118 của Ghi chú bài giảng trong Thống kê. Springer Science & Business Media, New York, 1996.

RM Neal và GE Hinton. Một quan điểm về thuật toán em biện minh cho trong-cremental, thưa thớt và các biến thể khác. Trong Học tập trong các mô hình đồ họa, trang 355–368. Springer, 1998.

J. Nocedal và SJ Wright. Tối ưu hóa số. Khoa học Springer & Truyền thông kinh doanh, ấn bản thứ 2, 2006. ISBN 978-0-387-30303-1.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, và cộng sự. Đào tạo các mô hình ngôn ngữ để tuân theo hướng dẫn với phản hồi của con người. Những tiến bộ trong hệ thống xử lý thông tin thần kinh, 35: 27730–27744, 2022.

B. Peters, V. Niculae và AF Martins. Các mô hình trình tự đến trình tự thưa thớt. bản in trước arXiv arXiv: 1905.05702, 2019.

F. Rosenblatt. Perceptron: một mô hình xác suất để lưu trữ thông tin và tổ chức trong não. Đánh giá tâm lý, 65(6):386, 1958.

R. Y. Rubinstein và DP Kroese. Phương pháp entropy chéo: một ứng dụng thống nhất-đề cập đến tối ưu hóa tổ hợp, mô phỏng Monte-Carlo và học máy. Truyền thông Khoa học & Kinh doanh Springer, 2004.

DE Rumelhart, GE Hinton và RJ Williams. Các đại diện học tập bằng cách truyền ngược lỗi. Thiên nhiên, 323 (6088): 533–536, 1986.

R. E. Schapire. Sức mạnh của khả năng học hỏi yếu. Học máy, 5(2): 197–227, 1990. doi: 10.1007/BF00116037.

P. Smolensky. Xử lý thông tin trong hệ thống động: Nền tảng của Lý thuyết hài hòa. Trong DE Rumelhart và JL McClelland, biên tập viên, Xử lý phân tán Paral-lel: Khám phá trong cấu trúc vi mô của nhận thức, Tập 1: Nền tảng, trang 194–281. Nhà xuất bản MIT, Cambridge, MA, 1986.

J. Su, Y. Lu, S. Pan, B. Murtadha, S. Wen và Y. Liu. Roformer: Nâng cao máy biến áp với vị trí quay nhúng. Trong Kỷ yếu của Hội nghị quốc tế năm 2021 về Đại diện Học tập, 2021.

R. Sutton. Bài học cay đắng. Ý tưởng chưa hoàn chỉnh (blog), 13(1):38, 2019.

RS Sutton. Học cách dự đoán bằng các phương pháp khác biệt về thời gian. Mẹ-
Học Trung Quốc, 3:9–44, 1988.

ME Tipping và CM Bishop. Phân tích thành phần chính xác suất.
Tạp chí của Hiệp hội Thống kê Hoàng gia: Series B (Phương pháp thống kê), 61 (3): 611–622, 1999.

V. N. Vapnik và A. Y. Chervonenkis. Về sự hội tụ đồng nhất của rela-
tần số của các sự kiện đến xác suất của chúng. Lý thuyết xác suất và ứng dụng của nó, 16(2):264–280, 1971.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez,
L. Kaiser, và I. Polosukhin. Chú ý là tất cả những gì bạn cần. Trong Nâng cao hệ thống xử lý thông tin thần kinh, trang 5998–6008, 2017.

M. Welling và Y. W. Teh. Học Bayes thông qua langevin gradient ngẫu nhiên
động lực học. Kỷ yếu của hội nghị quốc tế lần thứ 28 về học máy (ICML-11), trang 681–688, 2011.

Những người đóng góp Wikipedia. Sự đánh đổi thiên vị-phương sai — Wikipedia, en- miễn phí
bách khoa toàn thư, 2023. URL https://en.wikipedia.org/w/index.php?
title=Bias%E2%80%93variance_tradeoff&oldid=1178072263. [Trực tuyến;
truy cập ngày 16 tháng 10 năm 2023].

J. Zhao, M. Mathieu và Y. LeCun. Mạng đối nghịch sinh sản dựa trên năng lượng-
làm. bản in trước arXiv arXiv: 1609.03126, 2016.