Học máy và Thị giác máy tính

SangDV, BKAI

Jan 10, 2023

Mục lục

Chương 1

Hàm mục tiêu

1.1 Mối liên quan giữa MSE và CE qua góc nhìn MLE

Đối với các mô hình học máy có giám sát, Cross Entropy (CE) thường được sử dụng cho tác vụ phân lớp, còn hàm trung bình bình phương độ lỗi (Mean Square Error - MSE) lại thường được sử dụng cho tác vụ hồi quy. Câu hỏi đặt ra là tại sao lại như vậy?

Trước hết, CE và MSE về bản chất chỉ là trường hợp riêng của MLE (Maximum Likelihood Estimation) [xem Goodfellow et al., 2016, chap 5].

Xét bài toán học có giám sát với tập học $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ được lấy mẫu độc lập và tuân theo phân phối giống nhau (*i.i.d.*) từ phân phối dữ liệu $\Omega(\mathcal{X}, \mathcal{Y})$, trong đó $\mathcal{X}$ là không gian dữ liệu và $\mathcal{Y}$ là không gian nhãn. Ký hiệu $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ và $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$. Xét mô hình học máy giám sát ước lượng xác suất có điều kiện $Pr(y|\mathbf{x}, \boldsymbol{\theta})$ để dự đoán đầu ra y từ dữ liệu vào $\mathbf{x}$. Khi đó ước lượng MLE của tham số mô hình có dạng:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{MLE} &= \arg\max_{\boldsymbol{\theta}} Pr(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{n} Pr(y_i|\mathbf{x}_i, \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log Pr(y_i|\mathbf{x}_i, \boldsymbol{\theta}) \end{aligned} \tag{1.1}$$

Dẫn xuất MSE từ MLE trong bài toán hồi quy

Đối với bài toán hồi quy, thông thường ánh xạ $\mathbf{x} \to y$ không phải là hàm số, nghĩa là tồn tại những mẫu học cùng một dữ liệu đầu vào $\mathbf{x}$ nhưng có các nhãn y tương ứng khác nhau. Chẳng hạn, khi xét bài toán hồi quy cân nặng, chiều cao, các chỉ số sinh hoá, giá trị nhãn y có thể bị sai lệch do thiết bị đo đạc và thiết bị xét nghiệm... Trong bài toán hồi quy giá nhà, cùng một $\mathbf{x}$ (cùng diện tích, cùng tầng, cùng số phòng...) nhưng có thể có nhiều mức giá y khác nhau tuỳ thời điểm và các yếu tố khách quan khác. Vì vậy, người ta thường giả định với cùng một dữ liệu vào $\mathbf{x}$ thì nhãn y tuân theo phân phối tự nhiên:

$$y = \mathcal{N}(f(\mathbf{x}), \beta^2) = f(\mathbf{x}) + \epsilon \tag{1.2}$$

trong đó $\epsilon = \mathcal{N}(0, \beta^2)$ là nhiễu Gauss với kỳ vọng bằng 0; $f(\mathbf{x})$ là ánh xạ đúng biểu diễn quan hệ giữa đầu vào và đầu ra bài toán.

Do ánh xạ $f(\mathbf{x})$ không biết trước nên ta cần xấp xỉ nó bằng một mô hình học máy $h(\mathbf{x}, \boldsymbol{\theta})$:

$$y = h(\mathbf{x}, \boldsymbol{\theta}) + \epsilon \tag{1.3}$$

Theo định nghĩa phân phối Gauss, ta có:

$$Pr(y|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\beta^2}} e^{-\frac{(y-h(\mathbf{x},\boldsymbol{\theta}))^2}{2\beta^2}} \tag{1.4}$$

Thay Công thức (1.4) vào Công thức (1.1) ta thu được:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{MLE} &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \Big(-\frac{1}{2} log(2\pi) - log(\beta) - \frac{(y_i - h(\mathbf{x}_i, \boldsymbol{\theta}))^2}{2\beta^2} \Big) \\
&= \arg\max_{\boldsymbol{\theta}} \Big[\underbrace{-\frac{n}{2} log(2\pi) - nlog(\beta)}_{const} - \sum_{i=1}^{n} \Big(\frac{(y_i - h(\mathbf{x}_i, \boldsymbol{\theta}))^2}{2\beta^2} \Big) \Big] \\
&= \arg\min_{\boldsymbol{\theta}} \underbrace{\sum_{i=1}^{n} (y_i - h(\mathbf{x}_i, \boldsymbol{\theta}))^2}_{MSE\ loss}
\end{aligned} \tag{1.5}$$

Dẫn xuất CE từ MLE trong bài toán phân lớp

Tiếp theo, ta xét bài toán phân loại với K lớp sử dụng softmax. Đầu ra của lớp softmax $\mathbf{p} = h(\mathbf{x}, \boldsymbol{\theta})$ là $\mathbf{p} = \{p(1), p(2), \ldots, p(K)\}$ tương ứng với xác suất thuộc về K lớp được xác định như sau:

$$p(c) = \frac{e^{v_c}}{\sum_{j=1}^{K} e^{v_j}}, c = 1, 2, \ldots, K \tag{1.6}$$

trong đó $\{v_1, v_2, ..., v_K\}$ là đầu vào của lớp softmax.

Giả sử nhãn $y = c$ được mã hoá dưới dạng one-hot $\{y(1), y(2), \ldots, y(K)\} = \{0, \ldots, 0, \underbrace{1}_{c}, 0, \ldots, 0\}$.

Khi đó nhãn y có thể biểu diễn tuân theo phân phối Bernoulli tổng quát:

$$Pr(y = c|\mathbf{x}, \boldsymbol{\theta}) = p_c = \prod_{j=1}^{K} p(j)^{y(j)} \tag{1.7}$$

Thay Công thức (1.7) vào Công thức (1.1) ta thu được:

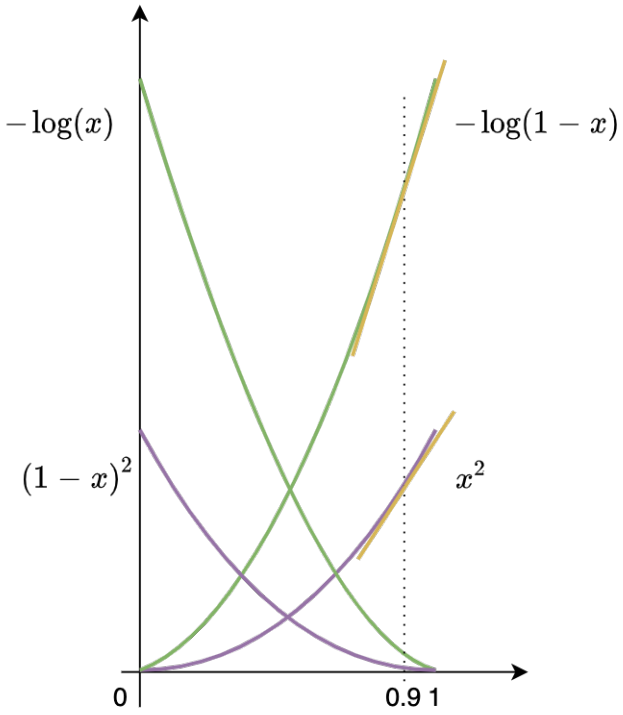
$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{MLE} &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \sum_{j=1}^{K} y_i(j) log(p_i(j)) \\
&= \arg\min_{\boldsymbol{\theta}} \underbrace{-\sum_{i=1}^{n} \sum_{j=1}^{K} y_i(j) log(p_i(j))}_{CE\ loss}
\end{aligned} \tag{1.8}$$

Như vậy hàm mất mát MSE và CE đều suy dẫn từ MLE ứng với hai giả định phân phối nhãn tương ứng là Gauss (đối với hồi quy) và Bernoulli (đối với phân lớp). Mặt khác Likelihood trong Công thức (1.1) cũng có thể hiểu là CE giữa phân phối dữ liệu và phân phối ước lượng mô hình [xem Goodfellow et al., 2016, chap 5, p 130]. Do đó có thể nói MSE về bản chất chính là CE.

1.2 Tại sao MSE thông thường không tốt cho bài toán phân lớp?

Như phân tích ở Phần 1.1, MSE là dẫn xuất của MLE khi ta giả định nhãn dữ liệu tuân theo phân phối Gauss. Rõ ràng giả định này không phù hợp với bài toán phân lớp.

Cụ thể hơn, MSE không ưu tiên nhiều vào nhãn đúng và phạt quá ít khi dự đoán bị lệch ra khỏi nhãn đúng. Đối với bài toán hồi quy, việc dự đoán lệch nhãn đúng một chút không phải quá nghiêm trọng. Nhưng đối với bài toán phân lớp, đầu ra mô hình học máy thường là phân phối xác suất trong khoảng [0, 1], một sự thay đổi nhỏ về dự đoán phân phối xác suất là dẫn tới ngay sự khác biệt về kết quả phân lớp, chẳng hạn thay đổi từ 0.49 thành 0.51 đối với phân lớp nhị phân. Xét một ví dụ đơn giản với phân lớp nhị phân, nhãn đúng là 0, mô hình dự đoán là 0.9. Khi đó MSE phạt $0.9^2 = 0.81$, nhưng CE phạt $-log(1 - 0.9) = 2.3$. Đạo hàm của MSE là $2 \times 0.9 = 0.18$, bé hơn nhiều lần so với đạo hàm của CE là $\frac{1}{1-0.9} = 10$. Do đó tín hiệu gradient của CE mạnh hơn so với MSE, cho phép các tham số cập nhật nhanh hơn và hội tụ nhanh hơn. Đây chính là lý do tại sao CE thường cho kết quả tốt hơn MSE thông thường trên các tác vụ phân lớp.



Hình 1.1: Hàm MSE phạt ít hơn CE khi dự đoán sai. Đạo hàm của MSE cũng bé hơn đạo hàm của CE nhiều lần.

1.3 MSE hiệu chỉnh cho bài toán phân lớp

Hiểu được yếu điểm của MSE thông thường khi chú ý quá ít vào nhãn đúng, ta có thể hiệu chỉnh giúp nó làm việc hiệu quả.

Với bài toán phân lớp, hàm mất mát MSE thông thường cho một mẫu dữ liệu $\mathbf{x}$ có nhãn $y = c$ được mã hoá one-hot có thể biểu diễn như sau:

$$MSE = \frac{1}{K}\Big((p(c) - 1)^2 + \sum_{j=1, j\neq c} p(c)^2\Big) \tag{1.9}$$

Hui and Belkin [2021] đã đề xuất hàm mất mát Rescaled MSE để tập trung hơn vào nhãn đúng:

$$MSE = \frac{1}{K}\Big(\lambda * (p(c) - M)^2 + \sum_{j=1, j\neq c} p(c)^2\Big) \tag{1.10}$$

trong đó tham số λ để tăng trọng số của loss vào nhãn đúng và tham số M để khuếch đại giá trị của mã hoá one-hot. Nếu $\lambda = M = 1$ ta có hàm mất mát MSE thông thường như Công thức (1.9).

Hui and Belkin [2021] chỉ ra rằng dùng hàm mất mát MSE hiệu chỉnh trong Công thức (1.10) để huấn luyện các mạng nơ-ron cho kết quả tương đương hoặc thậm chí tốt hơn hàm mất mát CE trong nhiều tác vụ phân lớp.

Một ưu điểm khác của MSE là tính đơn giản về mặt giải tích, cho phép đơn giản hoá việc nghiên cứu tính chất lý thuyết của mô hình học máy phức tạp như mạng nơ-ron. Ví dụ Liu et al. [2022] đã chỉ ra rằng hàm mất mát MSE với các mạng nơ-ron tham số hoá quá mức (overparameterized) thoả mãn tính chất μ-PL* hầu như mọi nơi. Điều này đảm bảo các giải thuật (S)GD với tốc độ học đủ nhỏ luôn hội tụ tới cực tiểu toàn cục với tốc độ hàm mũ.

Bên cạnh đó, Papyan et al. [2020] và Han et al. [2021] cũng chỉ ra rằng cả MSE và CE đều dẫn tới hiện tượng Neural Collapse ở các bước cuối khi huấn luyện các mô hình học máy trong giai đoạn TPT (Terminal Phase of Training), tức là giai đoạn khi mô hình đã nội suy tập học (lỗi tập học bằng 0 và hàm mất mát trên tập học tiến đến 0). Tại trạng thái Neural Collapse, (NC1) phương sai véc-tơ đặc trưng tại layer cuối của các mẫu học trong từng lớp tiến tới 0, nghĩa là véc-tơ đặc trưng của các mẫu học gần như trùng với véc-tơ đặc trưng trung bình (class-means) của cả lớp; (NC2) Các véc-tơ đặc trưng trung bình của các lớp có độ dài bằng nhau, góc giữa hai véc-tơ đôi một bằng nhau và chúng phân tán xa nhau nhất có thể. Cấu trúc này gọi là Simplex Equiangular Tight Frame (ETF); (NC3) Các véc-tơ đặc trưng trung bình hội tụ về véc-tơ tham số tương ứng của lớp đó (tham số W của lớp linear trước softmax); (NC4) Quá trình suy diễn đơn giản trở thành là 1-NN trên tập các véc-tơ trung bình class-means.

Tuy nhiên cũng cần lưu ý rằng hiện tượng Neural Collapse chỉ xảy ra trên tập học [Hui et al., 2022], do đó nó chỉ là một hiện tượng trong quá trình tối ưu chứ không có mối liên quan chặt chẽ với khả năng tổng quát hoá của mô hình.

Chương 2

Bias và Variance

2.1 Rủi ro kỳ vọng (expected risk)

Xét bài toán học có giám sát với tập học $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ được lấy mẫu độc lập và tuân theo phân phối giống nhau (*i.i.d.*) từ phân phối dữ liệu $\Omega(\mathcal{X}, \mathcal{Y})$, trong đó $\mathcal{X}$ là không gian dữ liệu và $\mathcal{Y}$ là không gian nhãn.

Giả sử $f : \mathcal{X} \to \mathcal{Y}$ là ánh xạ đúng (*true function*) biểu diễn quan hệ từ một dữ liệu đầu vào $\mathbf{x} \in \mathcal{X}$ sang một nhãn đầu ra $y \in \mathcal{Y}$:

$$y = f(\mathbf{x}) + \epsilon \tag{2.1}$$

trong đó $\epsilon = \mathcal{N}(0, \beta^2)$ là nhiễu Gauss với kỳ vọng bằng 0.

Như đã thảo luận ở Phần 1.1, một dữ liệu đầu vào $\mathbf{x}$ có thể tương ứng với nhiều nhãn y khác nhau. Kỳ vọng của nhãn y thu được ứng với dữ liệu đầu vào $\mathbf{x}$ chính là $f(\mathbf{x})$:

$$f(\mathbf{x}) = \mathbb{E}\big[y|\mathbf{x}\big] = \int_y yPr(y|\mathbf{x})\partial y \tag{2.2}$$

Giả sử ta xấp xỉ $f(\mathbf{x})$ bằng giả thuyết $h(\mathbf{x}, \mathbf{D})$ là một mô hình có tham số được huấn luyện trên tập học D. Quá trình huấn luyện h thường được quy về bài toán cực tiểu hóa rủi ro thực nghiệm (*emprirical risk* hoặc *training error*):

$$\hat{R}_D(h) \triangleq \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(h(\mathbf{x}_i, D), y_i) \tag{2.3}$$

trong đó $\mathcal{L}(h(\mathbf{x}_i, D), y_i)$ là hàm chi phí được định nghĩa trước.

Tuy nhiên, hiệu năng của mô hình phải được đánh giá trên toàn bộ phân phối dữ liệu $\mathcal{D}$, bao gồm cả những dữ liệu chưa nhìn thấy (dữ liệu test), sử dụng độ đo gọi là rủi ro kỳ vọng (*expected risk*):

$$R_D(h) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\Omega}\big[\mathcal{L}(h(\mathbf{x}, D), y)\big] = \int_{\mathbf{x}}\int_y \mathcal{L}(h(\mathbf{x}, D), y)Pr(\mathbf{x}, y)\partial y\partial\mathbf{x} \tag{2.4}$$

Giả thuyết h khi huấn luyện trên các tập học D khác nhau sẽ thu được các mô hình $h(\mathbf{x}, D)$ khác nhau ứng với các bộ tham số khác nhau. Do đó khi đánh giá giả thuyết h, ta quan tâm tới rủi ro kỳ vọng ứng với mọi trường hợp khác nhau của tập học D:

$$\mathbb{E}_D\big[R_D(h)\big] = \int_D R_D(h)Pr(D)\partial D = \int_D \int_{\mathbf{x}} \int_y \mathcal{L}(h(\mathbf{x}, D), y)Pr(\mathbf{x}, y)Pr(D)\partial y \partial \mathbf{x} \partial D$$
$$= \mathbb{E}_{\mathbf{x},y,D}\big[\mathcal{L}(h(\mathbf{x}, D), y)\big] \tag{2.5}$$

Mô hình kỳ vọng $\bar{h}(\mathbf{x})$ là trung bình tất cả các mô hình $h(\mathbf{x}, D)$ được huấn luyện trên tất cả các tập học D khác nhau:

$$\bar{h}(\mathbf{x}) = \mathbb{E}_D\big[h(\mathbf{x}, D)\big] = \int_D h(\mathbf{x}, D)Pr(D)\partial D \tag{2.6}$$

2.2 Phân rã bias và variance

Xét trường hợp hồi quy tuyến tính với hàm mất mát bình phương $\mathcal{L}(h(\mathbf{x}, D), y) = (h(\mathbf{x}, D) - y)^2$. Khi đó, rủi ro kỳ vọng có thể khai triển như sau [xem Bishop and Nasrabadi, 2006, chap 3, pp. 147-152]:

$$\begin{aligned}
\mathbb{E}_{\mathbf{x},y,D}\big[\mathcal{L}(h(\mathbf{x}, D), y)\big] &= \mathbb{E}_{\mathbf{x},y,D}\big[\big(h(\mathbf{x}, D) - y\big)^2\big] = \mathbb{E}_{\mathbf{x},y,D}\big[\big((h(\mathbf{x}, D) - \bar{h}(\mathbf{x})) + (\bar{h}(\mathbf{x}) - y)\big)^2\big] \\
&= \mathbb{E}_{\mathbf{x},y,D}\big[\big(h(\mathbf{x}, D) - \bar{h}(\mathbf{x})\big)^2\big] + \mathbb{E}_{\mathbf{x},y,D}\big[\big(\bar{h}(\mathbf{x}) - y\big)^2\big] \\
&+ 2\mathbb{E}_{\mathbf{x},y,D}\big[\big(h(\mathbf{x}, D) - \bar{h}(\mathbf{x})\big)\big(\bar{h}(\mathbf{x}) - y\big)\big] \\
&= \mathbb{E}_{\mathbf{x},D}\big[\big(h(\mathbf{x}, D) - \bar{h}(\mathbf{x})\big)^2\big] + \mathbb{E}_{\mathbf{x},y}\big[\big(\bar{h}(\mathbf{x}) - y\big)^2\big] \\
&+ 2\mathbb{E}_{\mathbf{x},y,D}\big[\big(h(\mathbf{x}, D) - \bar{h}(\mathbf{x})\big)\big(\bar{h}(\mathbf{x}) - y\big)\big]
\end{aligned} \tag{2.7}$$

Xét số hạng thứ 3 trong Công thức (2.7), do $\bar{h}(\mathbf{x}) - y$ không phụ thuộc vào D nên có thể đặt làm thừa số chung:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x},y,D}\Big[\big(h(\mathbf{x}, D) - \bar{h}(\mathbf{x})\big)\big(\bar{h}(\mathbf{x}) - y\big)\Big] = \mathbb{E}_{\mathbf{x},y}\Big[\big(\bar{h}(\mathbf{x}) - y\big)\mathbb{E}_D\big[h(\mathbf{x}, D) - \bar{h}(\mathbf{x})\big]\Big] \\
&= \mathbb{E}_{\mathbf{x},y}\Big[\big(\bar{h}(\mathbf{x}) - y\big)\Big(\mathbb{E}_D\big[h(\mathbf{x}, D)\big] - \mathbb{E}_D\big[\bar{h}(\mathbf{x})\big]\Big)\Big] \\
&= \mathbb{E}_{\mathbf{x},y}\Big[\big(\bar{h}(\mathbf{x}) - y\big)\Big(\bar{h}(\mathbf{x}) - \bar{h}(\mathbf{x})\Big)\Big] \\
&= \mathbb{E}_{\mathbf{x},y}\Big[0\Big] \\
&= 0
\end{aligned} \tag{2.8}$$

Xét số hạng thứ 2 của Công thức (2.7), ta có:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x},y}\Big[\big(\bar{h}(\mathbf{x}) - y\big)^2\Big] = \mathbb{E}_{\mathbf{x},y}\Big[\big((\bar{h}(\mathbf{x}) - f(\mathbf{x})) + (f(\mathbf{x}) - y)\big)^2\Big] \\
&= \mathbb{E}_{\mathbf{x},y}\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)^2\Big] + \mathbb{E}_{\mathbf{x},y}\Big[(f(\mathbf{x}) - y)^2\Big] + 2\mathbb{E}_{\mathbf{x},y}\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)\big(f(\mathbf{x}) - y\big)\Big] \\
&= \mathbb{E}_{\mathbf{x}}\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)^2\Big] + \mathbb{E}_{\mathbf{x},y}\Big[(f(\mathbf{x}) - y)^2\Big] + 2\mathbb{E}_{\mathbf{x},y}\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)\big(f(\mathbf{x}) - y\big)\Big]
\end{aligned} \tag{2.9}$$

Xét số hạng thứ 3 trong Công thức (2.9), do $\bar{h}(\mathbf{x}) - f(\mathbf{x})$ không phụ thuộc vào y nên có thể đặt làm thừa số chung như sau:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x},y}\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)\big(f(\mathbf{x}) - y\big)\Big] = \mathbb{E}_x\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)\mathbb{E}_{y|x}\big[f(\mathbf{x}) - y\big]\Big] \\
&= \mathbb{E}_x\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)\big(\mathbb{E}_{y|x}\big[f(\mathbf{x})\big] - \mathbb{E}_{y|x}\big[y\big]\big)\Big] \\
&= \mathbb{E}_x\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)\big(f(\mathbf{x}) - f(\mathbf{x})\big)\Big] \\
&= 0
\end{aligned} \tag{2.10}$$

Kết hợp các Công thức (2.7), (2.8), (2.9), (2.10), ta có phân rã cuối cùng:

$$\underbrace{\mathbb{E}_{\mathbf{x},y,D}\big[\big(h(\mathbf{x},D) - y\big)^2\big]}_{expected\ risk} = \underbrace{\mathbb{E}_{\mathbf{x},D}\big[\big(h(\mathbf{x},D) - \bar{h}(\mathbf{x})\big)^2\big]}_{variance} + \underbrace{\mathbb{E}_{\mathbf{x}}\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)^2\Big]}_{bias^2} + \underbrace{\mathbb{E}_{\mathbf{x},y}\Big[\big(f(\mathbf{x}) - y\big)^2\Big]}_{noise} \tag{2.11}$$

Như vậy ta có các khái niệm sau:

Bias của giả thuyết $h(\mathbf{x})$ tại một điểm dữ liệu vào $\mathbf{x}$ thể hiện sự chênh lệch giữa giá trị nhãn kỳ vọng $f(\mathbf{x})$ so với kỳ vọng $\bar{h}(\mathbf{x})$ của các phán đoán $h(\mathbf{x}, D)$ do các mô hình được huấn luyện trên các tập học D khác nhau đưa ra.

$$bias\big[h(\mathbf{x})\big] = \bar{h}(\mathbf{x}) - f(\mathbf{x}) = \mathbb{E}_D\big[h(\mathbf{x},D)\big] - f(\mathbf{x}) = \int_D h(\mathbf{x},D)Pr(D)\partial D - f(\mathbf{x}) \tag{2.12}$$

Nếu xét trên toàn bộ không gian dữ liệu $\mathcal{X}$, ta có khái niệm bias bình phương (*squared bias*) của giả thuyết h:

$$bias^2\big[h\big] = \mathbb{E}_{\mathbf{x}}\Big[\big(\bar{h}(\mathbf{x}) - f(\mathbf{x})\big)^2\Big] = \int_{\mathbf{x}}\left(\int_D h(\mathbf{x},D)Pr(D)\partial D - f(\mathbf{x})\right)^2 Pr(\mathbf{x})\partial\mathbf{x} \tag{2.13}$$

Variance của giả thuyết h tại một điểm dữ liệu vào $\mathbf{x}$ thể hiện sự phân tán của các phán đoán khác nhau $h(\mathbf{x}, D)$ do các mô hình được huấn luyện trên các tập học D khác nhau đưa ra.

$$variance\big[h(\mathbf{x})\big] = \mathbb{E}_D\Big[\big(h(\mathbf{x},D) - \bar{h}(\mathbf{x})\big)^2\Big] = \mathbb{E}_D\left[\Big(h(\mathbf{x},D) - \mathbb{E}_D\big[h(\mathbf{x},D)\big]\Big)^2\right] \tag{2.14}$$

Nếu xét trên toàn bộ không gian dữ liệu $\mathcal{X}$, ta có khái niệm variance của giả thuyết h như sau:

$$\begin{aligned}
variance\big[h\big] &= \mathbb{E}_{\mathbf{x},D}\big[\big(h(\mathbf{x},D) - \bar{h}(\mathbf{x})\big)^2\big] \\
&= \int_{\mathbf{x}}\int_D\left(h(\mathbf{x},D) - \int_D h(\mathbf{x},D)Pr(D)\partial D\right)^2 Pr(D)Pr(\mathbf{x})\partial D\partial\mathbf{x}
\end{aligned} \tag{2.15}$$

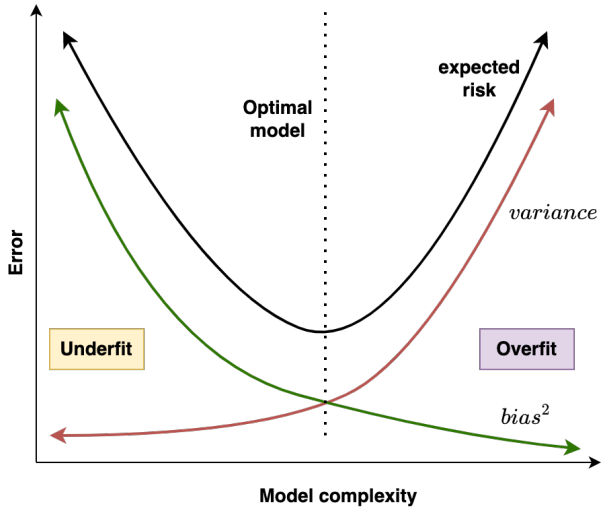
Noise là thành phần nhiễu không thể loại bỏ được (*irreducible error*). Nhiễu này không phụ thuộc vào giả thuyết nên dù mô hình hoá bài toán tốt như thế nào đi nữa cũng không

giảm được nó. Đây là thành phần ta bắt buộc phải "sống chung" do các yếu tố khách quan của dữ liệu.

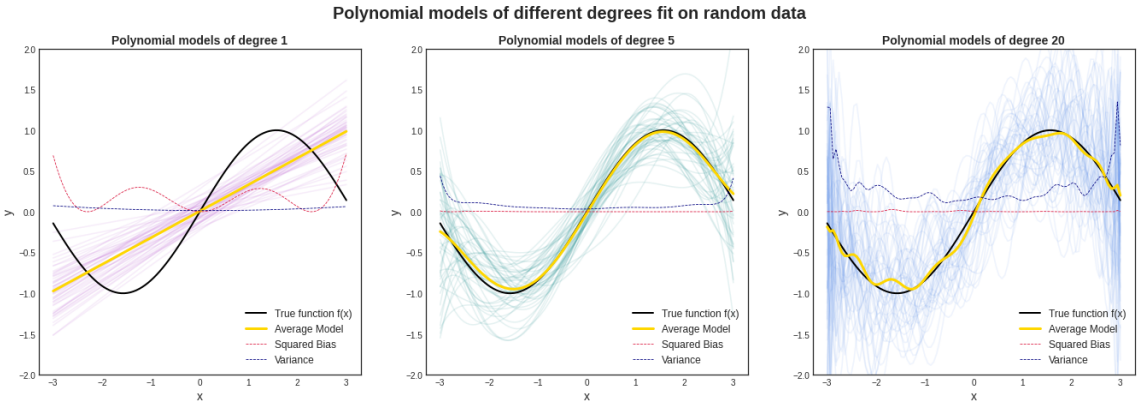
$$noise = \mathbb{E}_{\mathbf{x},y}\Big[\big(f(\mathbf{x}) - y\big)^2\Big] = \int_{\mathbf{x}}\int_{y}\big(f(\mathbf{x}) - y\big)^2 Pr(\mathbf{x}, y)\partial y \partial \mathbf{x} \tag{2.16}$$

2.3 Lý thuyết cổ điển về đánh đổi bias-variance

Trong lý thuyết cổ điển về học máy, bias và variance là hai đại lượng đối nghịch nhau. Khi giả thuyết h có bias thấp thì variance sẽ cao và ngược lại. Người ta gọi đó là đánh đổi (*tradeoff*) bias-variance.



Hình 2.1: Giả thuyết chữ U cổ điển về rủi ro kỳ vọng.



Hình 2.2: Minh hoạ hiện tượng underfit và overfit.

Underfitting là hiện tượng khi mô hình khớp dữ liệu học không tốt dẫn đến sai số cao trên cả tập học lẫn tập test. Điều này xảy ra khi mô hình quá đơn giản do có ít tham số, dẫn tới khả năng biểu diễn kém và không đủ khả năng để xấp xỉ ánh xạ đúng $f(\mathbf{x})$. Một

mô hình bị underfit thường có bias cao và variance thấp. Nghĩa là dù thay đổi nhiều tập học khác nhau thì sai số vẫn cao ổn định.

Overfitting là hiện tượng khi mô hình có sai số thấp trên tập học nhưng sai số trên tập test cao hơn đáng kể. Mô hình bị overfit khi khả năng tổng quát hoá (*generalization*) của mô hình kém. Lý thuyết học máy cổ điển cho rằng điều này xảy ra khi mô hình quá phức tạp so với dữ liệu do chứa quá nhiều tham số hơn mức cần thiết. Khi đó thay vì xấp xỉ ánh xạ đúng $f(\mathbf{x})$, mô hình tìm cách ghi nhớ (*memorize*) nhãn của các mẫu học trong quá trình huấn luyện mà không hiểu được quan hệ vốn có giữa dữ liệu và nhãn. Mô hình bị overfit thường có bias thấp và variance cao. Nghĩa là phán đoán của mô hình dao động rất lớn khi được huấn luyện trên nhiều tập học khác nhau, nhưng giá trị trung bình các phán đoán lại rất gần với giá trị nhãn đúng.

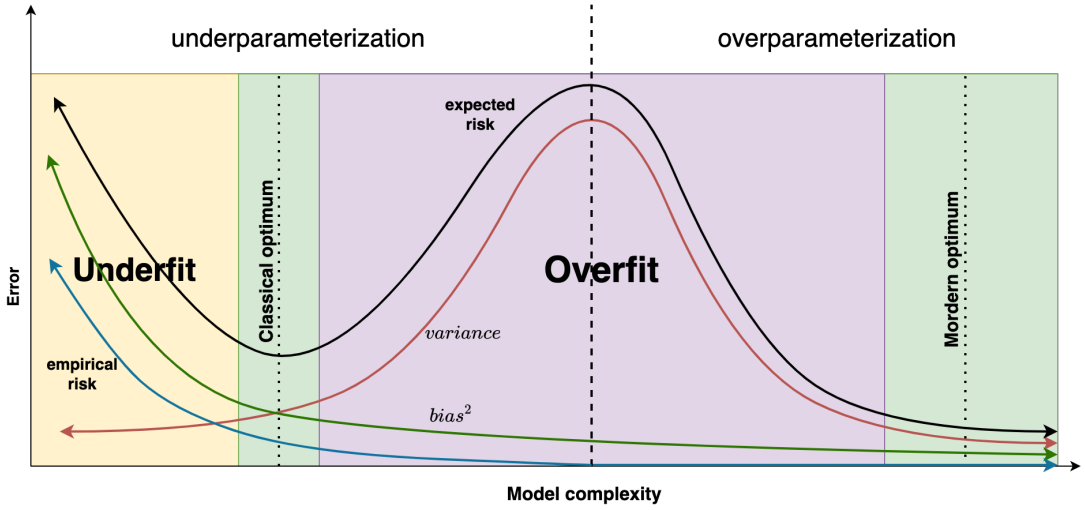
Giả thuyết chữ U về rủi ro kỳ vọng (*U-shaped risk curve*) trong lý thuyết học máy cổ điển cho rằng khi tăng dần độ phức tạp của mô hình, sai số kiểm tra kỳ vọng sẽ giảm dần khi mô hình chuyển từ trạng thái underfit sang trạng thái tối ưu (*classical optimum*) với rủi ro kỳ vọng thấp nhất. Sau đó nếu tiếp tục tăng độ phức tạp, mô hình sẽ dần mất trạng thái tối ưu và chuyển sang trạng thái overfit. Khi độ phức tạo càng tăng, trạng thái overfit càng nghiêm trọng, nghĩa là sai số kiểm tra kỳ vọng càng cao. Các nghiên cứu gần đầy chỉ ra rằng giả thuyết này chỉ phù hợp khi mô hình ở chế độ tham số hoá dưới mức (*underparameterization*). Khi mô hình có số tham số lớn hơn nhiều so với kích thước tập học, tức tham số hoá quá mức (*overparameterization*), giả thuyết chữ U về rủi ro kỳ vọng như Hình 2.2 không còn đúng nữa.

2.4 Double descent và lý thuyết học máy hiện đại

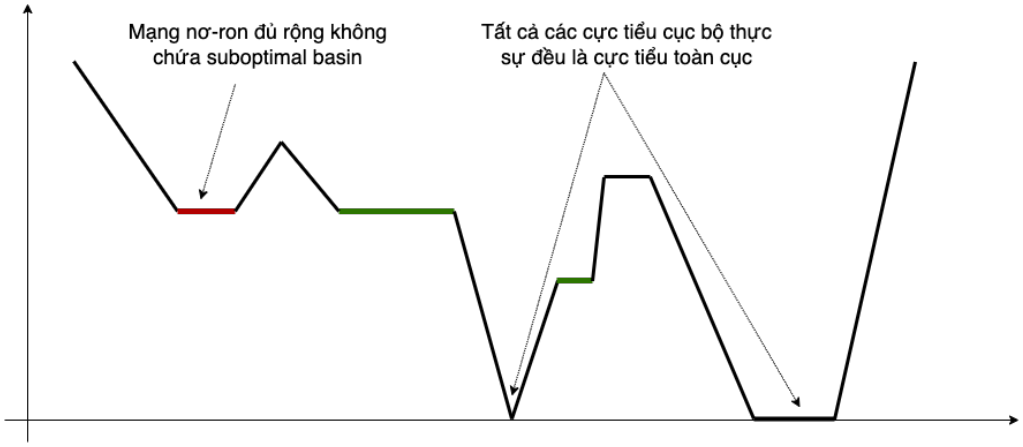
Hiện tượng hạ kép (*double descent*) lần đầu tiên được phát hiện bởi Belkin và cộng sự trong [Belkin et al., 2019]. Lý thuyết học máy cổ điển cho rằng mô hình càng lớn càng dễ bị quá khớp và có khả năng tổng quát hóa kém, nhưng Belkin và cộng sự đã chỉ ra rằng các mô hình càng lớn càng tốt hơn. [Belkin et al., 2019] phát hiện ra rằng bức tranh cổ điển hình chữ U về sự đánh đổi bias-variance sẽ bị phá vỡ tại thời điểm khi sai số huấn luyện xấp xỉ 0 mà họ gọi là ngưỡng nội suy (*interpolation threshold*). Trước ngưỡng nội suy, sự đánh đổi bias-variance được đảm bảo, và độ phức tạp của mô hình càng tăng sẽ dẫn đến hiện tượng học quá khớp, làm tăng rủi ro kỳ vọng. Tuy nhiên, sau ngưỡng nội suy, họ nhận thấy rằng lỗi kiểm tra bắt đầu giảm xuống khi tiếp tục tăng độ phức tạp của mô hình. Hiện tượng hạ kép được quan sát thấy không chỉ với các mạng nơ-ron mà cũng xảy ra với các mô hình học máy cổ điển dạng ensemble và boosting nói chung như Random Forest hay AdaBoost. Với các mô hình học máy cổ điển đơn giản hơn như mô hình hồi quy tuyến tính, hồi quy với đặc trưng ngẫu nhiên hiện tượng hạ kép có thể chứng minh bằng lý thuyết.

Trong chế độ nội suy khi mô hình bị tham số hoá quá mức (số tham số mô hình nhiều hơn số mẫu học), Rocks and Mehta [2022] đã chỉ ra bằng thực nghiệm rằng bias và variance đơn điệu giảm khi tăng độ phức tạp của mô hình (xem Hình 2.3). Khi số tham số đủ lớn, mô hình sẽ thoát khỏi overfitting, chuyển sang trạng thái "**benign overfit**" và cho khả năng tổng quát hoá tốt trên tập kiểm tra, thậm chí tốt hơn cả mô hình tối ưu cổ điển tham số hoá dưới mức (*classical optimum*).

Với mạng nơ-ron, khi mô hình đủ rộng (số nơ-ron ở mỗi lớp đủ lớn) và được huấn luyện bằng các giải thuật (S)GD, mô hình qua biến đổi hàm nhân NTK (*Neural Tangent Kernel*) sẽ hoạt động như các mô hình tuyến tính (*kernel machine*). [Li et al., 2018] đã chứng minh rằng khi mạng nơ-ron đủ rộng thì tất cả hố cực tiểu địa phương ngặt (set-wise strict local minimum) đều chứa cực tiểu toàn cục. Nghĩa là ngoại trừ các điểm cực tiểu cục bộ thuộc



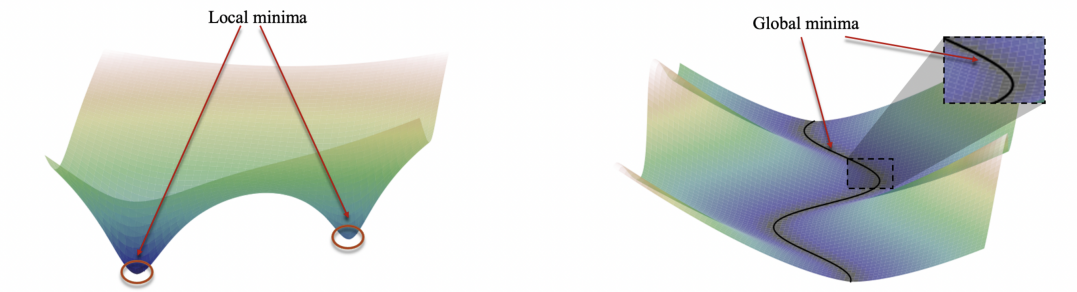
Hình 2.3: Hiện tượng hạ kép trong mạng nơ-ron và nhiều mô hình học máy cổ điển khác: Nửa trái là bức tranh cổ điển về sự đánh đổi bias-variance ở chế độ thiếu tham số hoá; Nửa phải là bức tranh hiện đại khi các mô hình được tham số hóa quá mức.



Hình 2.4: Hàm mất mát của các mạng nơ-ron đủ rộng có tính chất weak global minimum, không chứa những điểm cực tiểu không toàn cục như màu đỏ. Tất cả các điểm cực tiểu khác, ngoại trừ các điểm như màu xanh lá cây (không thuộc set-wise strict local minimum), đều là cực tiểu toàn cục.

vùng phẳng nằm ngang (plateau) lưng chừng giữa dốc đi tới cực tiểu toàn cục, tất cả các điểm cực tiểu cục bộ còn lại đều là cực tiểu toàn cục (xem Hình 2.4). Ở chế độ nội suy khi số tham số của mô hình vượt qua số mẫu học, Liu et al. [2022] khẳng định rằng tất cả các cực tiểu toàn cục đều có giá trị hàm mất mát bằng 0 và nằm liên tục cạnh nhau cùng nhiều điểm cực tiểu toàn cục khác tạo thành các manifold cực tiểu toàn cục (xem Hình 2.5). Liu et al. [2022] cũng chỉ ra rằng hàm mất mát của các mạng nơ-ron được tham số hoá quá mức thoả mãn tính chất μ-PL* hầu như mọi nơi, ngoại trừ những điểm mà NTK suy biến. Điều này đảm bảo giải thuật (S)GD cùng tốc độ học đủ nhỏ xuất phát từ điểm khởi tạo

w_0 bất kỳ, với w_0 là tâm một siêu cầu $B(w_0, R)$ bán kính cỡ $R = O(\frac{1}{\mu})$ mà trong siêu cầu đó đảm bảo tính chất μ-PL* cho hàm mất mát, đều hội tụ tới cực tiểu toàn cục với tốc độ hội tụ hàm mũ.



(a) Loss landscape of under-parameterized models

(b) Loss landscape of over-parameterized models

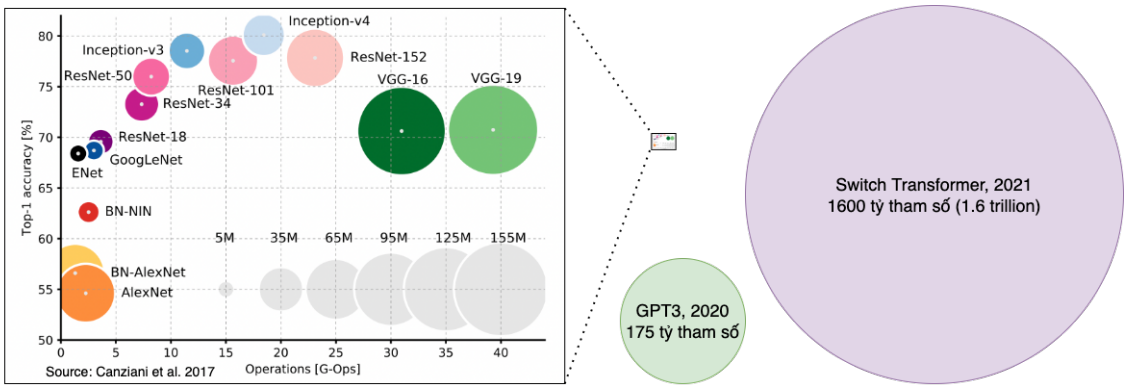
Hình 2.5: Bề mặt hàm mất mát của các mô hình học máy: a) Các mô hình ít tham số có nhiều cực tiểu địa phương độc lập. Hàm mất mát thường lồi cục bộ (locally convex) tại các điểm cực tiểu địa phương này; b) Các mô hình nhiều tham số chỉ có cực tiểu toàn cục và chúng nằm cạnh nhau tạo thành các manifold cực tiểu toàn cục. Hàm mất mát không lồi cục bộ (locally non-convex) tại bất kỳ điểm cực tiểu toàn cục nào trừ khi manifold đi qua điểm đó là tuyến tính cục bộ [Liu et al., 2022]

Như vậy đối với các mô hình tham số hoá quá mức, xuất suất rất cao ta sẽ tìm được cực tiểu toàn cục khi sử dụng giải thuật (S)GD xuất phát từ một điểm khởi tạo bất kỳ. Bản thân giải thuật (S)GD được chứng minh có tính tự hiệu chỉnh ngầm (*implicit regularization*) cho phép nó tự tìm đến các cực tiểu toàn cục rộng và phẳng (*flat wide minimum*), có độ mượt cao và có norm thấp (*small norm solutions*). Vì vậy nếu chiếu theo nguyên lý Occam's razor, mô hình học được bởi (S)GD "đơn giản hơn" các lời giải toàn cục khác và do vậy có tính tổng quát hoá cao hơn. Khi mô hình càng có nhiều tham số, không gian lời giải càng cao và (S)GD càng có nhiều cơ hội để lựa chọn các cực tiểu toàn cục với norm càng thấp, và các mô hình học được này (modern optimum) có tính tổng quát hoá càng cao như thể hiện trên phần bên phải Hình 2.3.

2.5 Nội suy tập học là cách tốt nhất để huấn luyện các mô hình học máy?

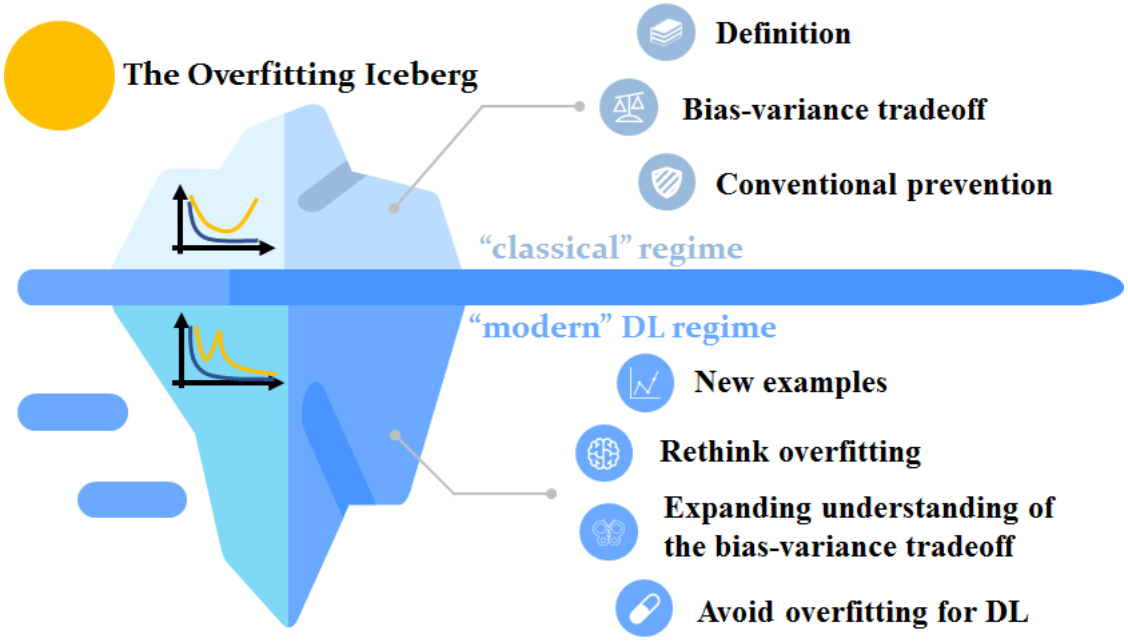
Bức tranh hiện đại về học máy cho ta thấy lý thuyết cổ điển về đánh đổi bias-variance với các kỹ thuật hiệu chỉnh chống overfitting thường thấy khi mô hình tham số hoá dưới mức chỉ là phần nổi của tảng băng chìm. Bức tranh lớn xảy ra lúc các mô hình ở chế độ nội suy khi được tham số hoá quá mức (xem Hình 2.7).

Ruslan Salakhudikov trong tutorial của mình về Deep Learning tại Simons Institution năm 2017 đã nói rằng: "the best way to solve the problam from practical standpoint is you **build a very big system** ... basically you want to **hit the zero training error**." Nghĩa là để giải quyết một bài toán học máy, ta hãy xây dựng một mô hình thật to với thật nhiều tham số để đảm bảo nó nội suy tập học. Việc tiếp tục tinh chỉnh (tuning) mô hình là cần thiết để được các kết quả SoTA nhưng chỉ cần nội suy dữ liệu học như vậy là đã có kết quả đủ tốt rồi.



Hình 2.6: Kích thước các mạng nơ-ron SoTA gần đây ngày càng lớn khủng khiếp so với kích thước tập học.

Thực tế các mô hình SoTA gần đây như GPT3 hay Switch Transformer đều có kích thước khủng khiếp lên tới 175 tỷ đến 1600 tỷ tham số, vượt xa kích thước tập học. Đây là minh chứng cho sự thành công khi huấn luyện mô hình học máy trong chế độ nội suy.



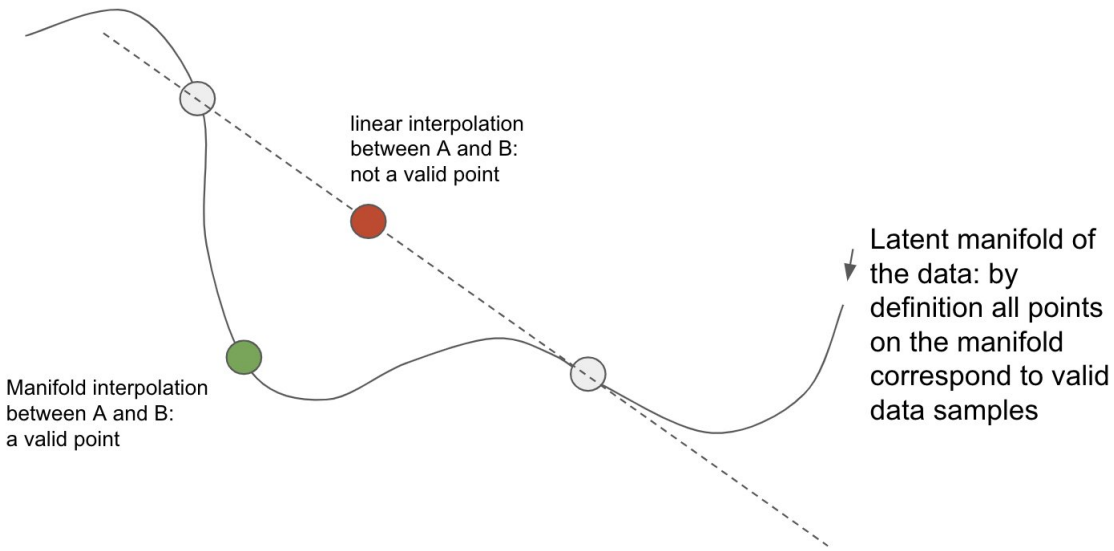
Hình 2.7: Lý thuyết học máy cổ điển về hình dạng chữ U của rủi ro kỳ vọng và các kỹ thuật hiệu chỉnh chống overfitting thường gặp chỉ là phần nổi của tảng băng chìm[1]. "**Hãy khớp dữ liệu đừng sợ hãi - Fit without fear** [Belkin, 2021]" để khám phá phần chìm của bức tranh trong chế độ nội suy tập học với đầy ắp những lời giải tốt.

[1]Source: https://blog.ml.cmu.edu/2020/08/31/4-overfitting/

2.6 Mô hình cần bao nhiêu tham số và dữ liệu bao nhiêu là đủ để giải quyết tốt bài toán?

Trong phần 2.5, ta đã thảo luận rằng tham số hoá quá mức vượt qua ngưỡng nội suy là phương pháp tốt nhất để giải quyết bài toán. Nhưng bao nhiêu tham số là đủ? Ngưỡng nội suy cho mô hình học máy giải quyết bài toán có m đầu ra trên một tập học có n mẫu là $m \times n$ [Belkin et al., 2019]. Ví dụ tập ImageNet-1K có khoảng 10^6 ảnh và 1000 lớp phân loại đầu ra, ngưỡng nội suy là cỡ $10^6 \times 1000 = 10^9$ tham số, lớn hơn rất nhiều so với các mô hình SoTA hiện tại trên tập này. Vì vậy có thể nói các kết quả SoTA hiện nay trên tập ImageNet-1K đều bị tham số hoá dưới mức và vẫn đang nằm trong phạm vi giải thích của học máy cổ điển.

Tuy nhiên, nội suy tập học là chưa đủ. Chúng ta cần một tập học đủ tốt để bao phủ khắp không gian dữ liệu. Vậy kích thước tập học bao nhiêu là đủ?



Hình 2.8: Nội suy đa tạp (*manifold interpolation*) khác với nội suy tuyến tính thông thường.

Nhiều người cho rằng quá trình suy diễn tập test của các mạng tham số hoá quá mức đơn giản chỉ là quá trình nội suy. Nhưng Balestriero et al. [2021] đã chứng minh rằng xác suất một mẫu kiểm tra nằm trong bao lồi (*convex hull*) của tập huấn luyện nhiều chiều là vô cùng bé, chính vì vậy đa số quá trình suy diễn của mạng nơ-ron hiện nay là ngoại suy. Để duy trì xác suất cao một tập dữ liệu kiểm tra ngẫu nhiên luôn nằm trong bao lồi của tập học, kích thước tập học N phải tăng theo hàm số mũ theo số chiều nội tại d^* (*intrinsic dimention*) của toàn bộ không gian dữ liệu $N > d^* \times 2^{d^*}$. Ở đây, số chiều nội tại của dữ liệu hiểu là số chiều không gian bé nhất mà ta có thể nén dữ liệu gốc xuống nhưng vẫn bảo toàn phần lớn thông tin của dữ liệu, ví dụ kích thước nội tại của bộ MNIST khoảng 7-12, CIFAR10 cỡ 13-26, ImageNet cỡ 26-43. Pope et al. [2021] chỉ ra bằng thực nghiệm rằng với cùng kích thước tập học, khả năng tổng quát hoá của mô hình chỉ phụ thuộc vào số chiều nội tại của dữ liệu mà không phụ thuộc vào các chiều còn lại (*extrinsic dimension*). Trong lý thuyết học đa tạp, Narayanan and Mitter [2010] cũng chỉ ra rằng để học được đường

biên giới phân tách tốt trong phân lớp nhị phân cần số lượng mẫu tỉ lệ hàm mũ với kích thước nội tại của dữ liệu.

Tuy nhiên, François Chollet trong lúc tranh luận với Yan Lecun đã đưa ra quan điểm khác về vấn đề này. François Chollet cho rằng không nên hiểu khái niệm dữ liệu nằm trong hay nằm ngoài bao lồi tập học dưới quan điểm tuyến tính của hình học thông thường. Cần nhìn nhận vấn đề nội suy trong không gian đa tạp (*manifold*) như thể hiện ở Hình 2.8. Giả sử dữ liệu test là điểm màu xanh, dữ liệu học là hai điểm màu xám. Mặc dù dưới góc nhìn tuyến tính trực tiếp từ không gian gốc, điểm màu xanh nằm ngoài khoảng nội suy giữa hai điểm màu xám, nhưng dưới góc nhìn manifold điểm màu xanh vẫn là điểm nằm giữa hai điểm màu xám. Vì vậy khi nói tới nội suy của các mạng nơ-ron ta cần hiểu là nội suy trong không gian manifold. Các lớp của mạng nơ-ron về bản chất là học ra các manifold của dữ liệu gốc ban đầu, lớp càng sâu thì manifold học được có số chiều càng bé và không gian dữ liệu học được càng cô đặc. Do đó để duy trì xác suất cao một dữ liệu test bất kỳ rơi vào miền nội suy manifold của tập học, không nhất thiết phải có số mẫu đủ lớn cỡ 2^{d^*} như trong [Balestriero et al., 2021] đề cập.

Việc trả lời chính xác cho câu hỏi kích thước dữ liệu lớn bao nhiêu là đủ vẫn là câu hỏi khó. Nhưng chúng ta có thể nói rằng kích thước dữ liệu cần phải tỉ lệ (cỡ hàm mũ?) với độ khó của dữ liệu (intrinsic dimension) và tỉ lệ thuận với số lượng đầu ra của mô hình.

Tài liệu tham khảo

R. Balestriero, J. Pesenti, and Y. LeCun. Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*, 2021.

M. Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

X. Han, V. Papyan, and D. L. Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

L. Hui and M. Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

L. Hui, M. Belkin, and P. Nakkiran. Limitations of neural collapse for understanding generalization in deep learning. *arXiv preprint arXiv:2202.08384*, 2022.

D. Li, T. Ding, and R. Sun. On the benefit of width for neural networks: Disappearance of bad basins. *arXiv preprint arXiv:1812.11039*, 2018.

C. Liu, L. Zhu, and M. Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

H. Narayanan and S. Mitter. Sample complexity of testing the manifold hypothesis. *Advances in neural information processing systems*, 23, 2010.

V. Papyan, X. Han, and D. L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117 (40):24652–24663, 2020.

P. Pope, C. Zhu, A. Abdelkader, M. Goldblum, and T. Goldstein. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*, 2021.

J. W. Rocks and P. Mehta. Memorizing without overfitting: Bias, variance, and interpolation in overparameterized models. *Physical Review Research*, 4(1):013201, 2022.