S A N J E E V A R O R A

其他贡献者：拉马纳罗拉，若昂布鲁纳，纳达夫科恩，西蒙杜，荣

G E , S U R I Y A G U N A S E K A R , E L A D H A Z A N , C H I J I N , J A S O N L E E , T E N G Y U M A , B E H N A M N E Y S H A B U R ，

赵松

深度学习理论

Contents

List of Figures

15.1 各种生成方法的文本困惑度。随机和贪婪方法相当糟糕。使用 $p = 0.95$ 的核采样最接近人类。（我们没有描述束搜索，所以请忽略那些行。）179 15.2 性能曲线：左图有 $\theta = 0.1$ 和变化 $k = 2$，4，8，16。k 的更高值大大提高了性能（对于 $k = 2$ 个有效的 α，β 不存在）。右图有 $k = 8$ 和 $\theta = 0.05$，0.1，0.2。第 ?? 节阐明，它还描述了模型对于 t-技能组的性能曲线，对于 $\theta = 0.05$ 和 $t = 1$，2，4 分别（例如，蓝色曲线为 4 元组）。188 18.1 *Flying pigs?* (A) 是一头猪的图像，而（B）是它的略微扰动版本。一个正常训练的 ResNet50 分类器将（B）标记为“飞机”。这两个图像之间的差异很小；在（C）中，您可以看到一个图像，其像素级差异是（A）和（B）的 50 倍。如果没有 50x 缩放，则（C）将包含接近 0 的值的像素（即，空白图像）。*Source: Kolter-Madry Tutorial.* 201 18.2 一个对抗性的 3D 对象！这个贴有少量贴纸的停车标志可靠地欺骗了图像识别分类器，将其分类为 45 英里/小时的限速。201 18.3 对输入 x_0 的 PGD 攻击。红色箭头对应于基于梯度的更新。当它们产生 Ball(x_0, r) 之外的点时，投影操作（用绿色箭头表示）找到球内的最近点。202 18.4 ℓ_∞-扰动范围内的对抗示例的概念说明。在标准分类器中，大多数数据点都接近决策边界，如 ℓ_∞ 距离所测。红色星号是对抗示例。经过对抗训练后，数据点周围的小 ℓ_∞-球不再与决策边界相交。信用：Madry 等人 2018。203 18.5 以 x (蓝色一个) 和 x' (红色一个) 为中心的单变量高斯分布，以及 $E(z)$ 从 1 到 0 转换的点。205

List of Tables

19.1 为了简化，我们忽略了 O。ℓ_∞/ℓ_2 是最强可能的保证，其次是 ℓ_2/ℓ_2，然后是 ℓ_2/ℓ_1，而恰好 k-sparse 是较弱的。我们还注意到，所有 [RV08, CGV13, Bou14, HR16] 都获得了对限制等距性质（Restricted Isometry property）的改进分析；算法在 [BD08] 中被建议并分析（模 RIP 性质）。[HIKP12] 中的工作没有明确说明扩展到 d-维的情况，但可以从论证中轻易推断出来。[HIKP12, IK14, Kap16, KVZ19] 在每个维度中宇宙大小为 2 的幂时工作。211

Introduction

版本日期：2022年3月28日

本专著讨论了深度学习的兴起理论。它起源于2019年秋季在普林斯顿大学教授的研讨班上的讲师笔记，并与高级研究所的优化、统计学和机器学习特别年联合举办。Sanjeev Arora在2021年春季和2022年春季课程的两次后续提供中清理并扩展了本书。

这更像是讲义而不是书籍。可能有很多错误和错别字。

1

Basic Setup and some math notions

本章介绍了基本命名法。训练/测试错误、泛化误差等。
≪Tengyu notes: Todos: Illustrate with plots: a typical training
曲线和测试曲线
提及一些流行的架构（前馈、卷积、池化、resnet、densenet），每项简要介绍。≫

我们回顾统计学习理论中的基本概念。

- 可能的数据点空间 $\mathcal{X}$。
- 一个可能的标签空间 $\mathcal{Y}$。
- 一个在 $\mathcal{X} \times \mathcal{Y}$ 上的联合概率分布 $\mathcal{D}$。我们假设我们的训练数据由 n 个数据点组成

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \overset{\text{i.i.d.}}{\sim} \mathcal{D},$$

每个独立地从 $\mathcal{D}$ 中抽取。

- 假设空间：$\mathcal{H}$ 是一组假设，或一组预测器。例如，$\mathcal{H}$ 可以是所有具有固定架构：$\mathcal{H} = \{h_\theta\}$ 的神经网络的集合：其中 h_θ 是由参数 θ 参数化的神经网络。

- 损失函数：$\ell : (\mathcal{X} \times \mathcal{Y}) \times \mathcal{H} \to \mathbb{R}$。

  例如，在二分类中，其中 $\mathcal{Y} = \{-1, +1\}$，假设我们有一个假设 $h_\theta(x)$，那么在数据点 (x, y) 上假设 h_θ 的逻辑损失函数是

$$\ell((x, y), \theta) = \frac{1}{1 + \exp(-y h_\theta(x))}.$$

- 预期损失：

$$L(h) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\ell((x, y), h)\right].$$

  回忆 $\mathcal{D}$ 是 $\mathcal{X} \times \mathcal{Y}$ 上的数据分布。

- 训练损失（也称为经验风险）：

$$\widehat{L}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(\left(x^{(i)}, y^{(i)}\right), h\right),$$

  在 $\left(x^{(1)},\ y^{(1)}\right)$，$\left(x^{(2)},\ y^{(2)}\right)$，...，$\left(x^{(n)},\ y^{(n)}\right)$ 是从 $\mathcal{D}$ 中独立同分布抽取的 n 训练样本。

- 经验风险最小化器（ERM）：$\widehat{h} \in \arg\min_{h\in\mathcal{H}} \widehat{L}(h)$。

- 正则化：假设我们有一个正则化器 $R(h)$，那么正则化损失是

$$\widehat{L}_\lambda(h) = \widehat{L}(h) + \lambda R(h)$$

  .

≪Suriya 注意：杂项符号：梯度、海森矩阵、范数{v*}

1.1 *List of useful math facts*

现在我们列出一些有用的数学事实。

1.1.1 *Probability tools*

本节介绍了我们在证明中使用的概率工具。引理1.1.4、1.1.5和1.1.6是关于随机标量变量的尾界。引理1.1.7是关于高斯分布的累积分布函数。最后，引理1.1.8是关于随机矩阵的集中结果。

引理1.1.1（马尔可夫不等式）。*If x is a nonnegative random variable and $t > 0$, then the probability that x is at least t is at most the expectation of x divided by t:*

$$\Pr[x \geq t] \leq \mathbb{E}[x]/t.$$

引理1.1.2（切比雪夫不等式）。*Let x denote a nonnegative random variable and $t > 0$, then*

$$\Pr[|x - \mathbb{E}[x]| \geq t] \leq \mathrm{Var}[x]/t^2.$$

接下来，我们介绍一些关于独立随机变量和的集中不等式。集中不等式背后的经验规则是 *Central Limit Theorem*。

定理1.1.3（中心极限定理，非正式）。*If X_1，X_2，...，X_n are independent random variables of mean μ_1，μ_2，...，μ_n and variances σ_1^2，σ_2^2，...，σ_n^2 then as n gets larger, $\sum_i X_i$ behaves like the normal distribution $\mathcal{N}(\sum_i \mu_i,\ \sum_i \sigma_i^2)$.*

浓度界是这种的定量版本，在p_i和σ_i可能依赖于n（变量总数）的设置中也能工作。但在许多情况下，中心极限定理是一个很好的经验法则。

引理1.1.4（切诺夫不等式 [Che52]）。*Let* $X = \sum_{i=1}^{n} X_i$, *where* $X_i = 1$ *with probability* p_i *and* $X_i = 0$ *with probability* $1 - p_i$, *and all* X_i *are independent. Let* $\mu = \mathbb{E}[X] = \sum_{i=1}^{n} p_i$. *Then*
1. $\Pr[X \geq (1+\delta)\mu] \leq \exp(-\delta^2\mu/3), \forall \delta > 0$; 2. $\Pr[X \leq (1-\delta)\mu] \leq \exp(-\delta^2\mu/2), \forall 0 < \delta < 1$.

引理1.1.5（Hoeffding界 [Hoe63]）。*Let* $X_1, \cdots, X_n$ *denote n independent bounded variables in* $[a_i, b_i]$. *Let* $X = \sum_{i=1}^{n} X_i$, *then we have*

$$\Pr[|X - \mathbb{E}[X]| \geq t] \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

引理1.1.6（伯恩斯坦不等式 [Ber24]）。*Let* $X_1, \cdots, X_n$ *be independent zero-mean random variables. Suppose that* $|X_i| \leq M$ *almost surely, for all i. Then, for all positive t,*

$$\Pr\left[\sum_{i=1}^{n} X_i > t\right] \leq \exp\left(-\frac{t^2/2}{\sum_{j=1}^{n}\mathbb{E}[X_j^2] + Mt/3}\right).$$

引理1.1.7（高斯分布的反集中性）。*Let* $X \sim N(0, \sigma^2)$, *that is, the probability density function of X is given by* $\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$. *Then*

$$\Pr[|X| \leq t] \in \left(\frac{2}{3}\frac{t}{\sigma}, \frac{4}{5}\frac{t}{\sigma}\right).$$

引理1.1.8（矩阵Bernstein，[Tro15]中的定理6.1.1）。*Consider a finite sequence* $\{X_1, \cdots, X_m\} \subset \mathbb{R}^{n_1 \times n_2}$ *of independent, random matrices with common dimension* $n_1 \times n_2$. *Assume that*

$$\mathbb{E}[X_i] = 0, \forall i \in [m] \quad \text{and} \quad \|X_i\| \leq M, \forall i \in [m].$$

Let $Z = \sum_{i=}^{m} 1\, X_i$. *Let* $\mathrm{Var}[Z]$ *be the matrix variance statistic of sum:*

$$\mathrm{Var}[Z] = \max\left\{\left\|\sum_{i=1}^{m}\mathbb{E}[X_i X_i^\top]\right\|, \left\|\sum_{i=1}^{m}\mathbb{E}[X_i^\top X_i]\right\|\right\}.$$

Then

$$\mathbb{E}[\|Z\|] \leq (2\mathrm{Var}[Z] \cdot \log(n_1 + n_2))^{1/2} + M \cdot \log(n_1 + n_2)/3.$$

Furthermore, for all $t \geq 0$,

$$\Pr[\|Z\| \geq t] \leq (n_1 + n_2) \cdot \exp\left(-\frac{t^2/2}{\mathrm{Var}[Z] + Mt/3}\right).$$

explain these in a para

一个有用的缩写如下：如果 y_1，y_2，... ，y_m 是具有均值为 0 且取值在 $[-1, 1]$ 中的独立随机变量，那么它们的平均值 $\frac{1}{m}\sum_i y_i$ 的行为类似于均值为零且方差最多为 $1/m$ 的高斯变量。换句话说，这个平均值绝对值至少为 ϵ 的概率至多为 $\exp(-\epsilon^2 m)$。

1.1.2 *Singular Value Decomposition*

待定。

2
Basics of Optimization

本章建立了基于梯度的优化算法的基本分析框架，并讨论了它如何应用于深度学习。这些算法在实践中效果良好；理论上的问题是分析它们并为实践提供建议。这证明更具挑战性，近年来越来越明显的是，关于优化的传统思维方式可能无法很好地与深度学习中遇到的现象相匹配。

优化中的基本概念框架建立在损失函数的简单泰勒近似（公式2.1）之上，因此依赖于损失函数的导数（各种阶数）。

≪Suriya notes: To ground optimization to our case, we can also mention that f is often of the either the ERM or stochastic optimization form $L(w) = \sum l(w; x, y)$ - it might also be useful to mention that outside of this chapter, we typically use f as an alternative for h to denote a function computed≫

2.1 Gradient descent (GD)

假设我们希望最小化一个在 $\mathbb{R}^d$ 上的连续函数 $f(w)$。

$$\min_{w \in \mathbb{R}^d} f(w) .$$

梯度下降（GD）算法是

$$\begin{aligned} w_0 &= \text{initialization} \\ w_{t+1} &= w_t - \eta \nabla f(w_t) \end{aligned}$$

在 η 被称为 *step size* 或 *learning rate.* 的地方，η 的选择很重要，并且是本章剩余部分的主要内容。

GD的一个动机或理由是，更新方向$-\nabla f(w_t)$在局部上是下降最快的方向。考虑

泰勒展开式在点 w_t 处

$$f(w) = f(w_t) + \underbrace{\langle \nabla f(w_t), w - w_t \rangle}_{\text{linear in } w} + \underbrace{\frac{1}{2}(w - w_t)^T \nabla^2 f(w_t)(w - w_t)}_{\text{quadratic in } w} + \cdots \tag{2.1}$$

这里 $\nabla^2(f)$ 是二阶导数的矩阵，称为 *Hessian*。它的 (i, j) 项是 $\partial^2 f / \partial w_i \partial w_j$。注意，它是一个对称矩阵。

假设我们丢弃高阶项，仅在一个以 w_t 为中心的邻域内优化一阶近似

$$\begin{aligned} \arg\min_{w \in \mathbb{R}^d} \quad & f(w_t) + \langle \nabla f(w_t), w - w_t \rangle \\ \text{s.t.} \quad & \|w - w_t\|_2 \le \epsilon \end{aligned}$$

问题 2.1.1. *Show that the optimizer of the program above is equal to* $w + \delta$ *where* $\delta = -\alpha \nabla f(w_t)$ *for some positive scalar* α.

换句话说，为了局部最小化 $f(\cdot)$ 在 w_t 附近的首次近似，我们应该朝着 $-\nabla f(w_t)$ 的方向移动。1

梯度下降不保证能找到一般损失函数的最优解。例如，复杂度理论表明，对于在n变量中总次数最多为6的4次多项式 $p(w_1, w_2, ..., w_n)$，确定它对于某些变量的赋值是否为0是NP难的。这可以通过使用3SAT问题的NP完备性来简单地证明。

经典反向传播算法（第4章）用于高效计算损失梯度。请注意，今天的深度网络通常使用非线性激活函数，例如ReLU，这使得网络的计算函数不可微分。然而，这种可微性是轻微的，并且在实践中似乎没有问题。2

最近，用可微分的激活函数如SWISH或GeLU替换ReLU激活，发现并不会降低性能。

2.1.1 *Upperbound on the Taylor Expansion via Smoothness*

GD的训练速度的基本分析涉及损失函数的平滑性。

定义 2.1.2 (L-平滑)。*A function* f *is* L*-smooth in a domain if for every* w *in the domain all eigenvalues of* $\nabla^2 f(w)$ *lie in the interval* $[-L, L]$.

问题 2.1.3. *Prove that if* f *is* L*-smooth then*

$$f(w) \le f(w_t) + \langle \nabla f(w_t), w - w_t \rangle + \frac{L}{2}\|w - w_t\|_2^2 \tag{2.2}$$

2.1.2 *Descent lemma for gradient descent*

以下说明，在梯度下降和足够小的学习率下，除非迭代点的梯度为零，否则函数值总是减少。（梯度为零的点称为*stationary points*。）

引理2.1.4（下降引理）。*Suppose* f *is* L*-smooth. Then, if* $\eta < 1/L$*, we have*

$$f(w_{t+1}) \le f(w_t) - \frac{\eta}{2} \cdot \|\nabla f(w_t)\|_2^2$$

证明使用了泰勒展开。主要思想是即使使用方程（2.2）提供的上界也足够了。3

3 证明还表明，只要 η <^2/L，就会发生下降。

Proof. 我们有以下结果

$$\begin{aligned} f(w_{t+1}) &= f(w_t - \eta \nabla f(w_t)) \\ &\leq f(w_t) + \langle \nabla f(w_t), -\eta \nabla f(w_t) \rangle + \frac{L}{2} \|\eta^2 \nabla f(w_t)\|_2^2 \\ &= f(w_t) - (\eta - \eta^2 L/2) \|\nabla f(w_t)\|_2^2 \\ &\leq f(w_t) - \frac{\eta}{2} \cdot \|\nabla f(w_t)\|_2^2, \end{aligned}$$

在第二步遵循公式（2.2），最后一步遵循 $\eta \leq 1/L$。

□

我们已经表明，当梯度 ∇ 变为零时，GD 停止取得进展。这够好吗?

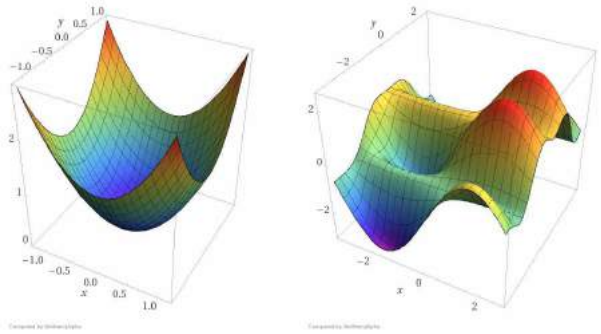


图2.1：二维中的凸和非凸函数。对于非凸函数，GD将到达一个梯度为零的驻点。（图来自kdnuggets.org）

深度学习的损失函数在具有多于一层网络时是非凸的。因此，梯度下降法并不保证产生全局最优解。然而，它在实践中找到的解在目标函数的值上相当低——甚至接近零。（回想一下，损失函数通常是非负的。）本书的其他地方解释了GD的这一性质在一些具体设置中的应用。

定义 2.1.5. *We say w is a* 静态点 *of f if $\nabla(f(w)) = 0$.*
If in addition $\nabla^2(f)$ is positive-semidefinite at w then w is called a local minimum.

2.2 *Stochastic gradient descent (SGD)*

SGD 是大型数据集梯度下降的一个非常实用的变体。回想一下

$$\widehat{L}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell \left((x^{(i)}, y^{(i)}), h \right).$$

计算梯度 $\nabla\widehat{L}(h)$ 在 n 中线性缩放，即训练数据集的大小。随机梯度下降（SGD）通过采样少量训练数据点并计算平均值来估计梯度。根据通常的采样定理，随着样本大小的增加，梯度估计趋近于真实梯度。

更新：我们为了便于阐述，稍微简化了符号。我们考虑优化函数

$$\frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

因此，在统计学习设置中，f_i 对应于 $\ell((x^i、y^{(i)})、h)$。在每次迭代 t 中，SGD 算法首先从 $[n]$ 中均匀采样 i_1、...、i_B，然后使用这些样本计算估计梯度：

$$g_S(w) = \frac{1}{B}\sum_{k=1}^{B} \nabla f_{i_k}(w_t)$$

这里 S 是 $\{i_1，...，i_B\}$ 的简称。SGD 算法通过以下方式更新迭代：{v*}

$$w_{t+1} = w_t - \eta \cdot g_S(w_t).$$

注意，如果学习率 η 非常小，那么参数在一系列更新中不会变化太多，因此 SGD 会趋向于与（完整）GD 相似。然而，当 η 不太小（并且批量大小较小）时，从批次中得到的梯度估计是真实梯度的噪声估计。人们可能会想象这会使 SGD 比GD更差。实际上，SGD 往往优于 GD。首先，由于随机梯度估计非常高效，SGD 在相同的计算预算下允许进行更多的迭代。其次，SGD 似乎对泛化有有益的影响，这意味着它找到的解往往比 GD 找到的解具有更好的测试误差。后面的章节将介绍试图解释 SGD 在泛化方面优于 GD 的理论的章节。

2.3 *Accelerated Gradient Descent*

加速梯度下降算法的基本版本被称为重球算法。它具有以下更新规则：

$$w_{t+1} = w_t - \eta\nabla f(w_t) + \beta(w_t - w_{t-1})$$

这里 $\beta(w_{t+1} - w_t)$ 是所谓的动量项。该算法的动机和名称的来源在于它可以

视为二阶常微分方程的离散化：

$$\ddot{w} + a\dot{w} + b\nabla f(w) = 0$$

另一种编写算法的等效方式是

$$\begin{aligned} u_t &= -\nabla f(w_t) + \beta u_{t-1} \\ w_{t+1} &= w_t + \eta u_t \end{aligned}$$

练习：验证算法的两种形式确实等价。重型球算法的另一种变体归功于Nesterov

$$\begin{aligned} u_t &= -\nabla f(w_t + \beta \cdot (u_t - u_{t-1})) + \beta \cdot u_{t-1}, \\ w_{t+1} &= w_t + \eta \cdot u_t. \end{aligned}$$

可以观察到 u_t 存储了所有过去梯度的加权总和。实际上，w_t 的更新依赖于所有过去的梯度。这是加速梯度下降算法的另一种解释。

Nesterov梯度下降在训练深度神经网络方面与重球算法在经验上相似。对于凸损失函数，它具有更强的最坏情况保证优势。加速GD的两种版本都可以与随机梯度一起使用，但关于随机加速梯度下降的理论保证知之甚少。

2.4 *Running time: Learning Rates and Update Directions*

当GD的迭代接近局部最小值时，梯度下降的行为更清晰，因为函数可以由一个二次函数局部近似。在本节中，为了简单起见，我们假设我们正在优化一个凸二次函数，并了解函数的曲率如何影响算法的收敛。

我们使用梯度下降进行优化

$$\min_w \frac{1}{2} w^\top A w$$

在 $A \in \mathbb{R}^{d\times d}$ 是一个正定矩阵，并且 $w \in \mathbb{R}^d$。注：w.l.o.g，我们可以假设 A 是一个对角矩阵。对角化是线性代数中的一个基本思想。假设 A 有奇异向量分解 $A = U\Sigma U^\top$，其中 Σ 是一个对角矩阵。我们可以验证 $w^\top A w = \widehat{w}^\top \Sigma \widehat{w}$ 与 $\widehat{w} = U^\top w$。换句话说，在一个由 U 定义的差分坐标系中，我们正在处理一个以对角矩阵 Σ 为系数的二次型。注意这里的对角化技术仅用于分析。

因此，我们假设 $A = \text{diag}(\lambda_1, \ldots, \lambda_d)$，其中 $\lambda_1 \geq \cdots \geq \lambda_d$。该函数可以简化为

$$f(w) = \frac{1}{2}\sum_{i=1}^{d} \lambda_i w_i^2$$

梯度下降更新可以表示为

$$x \leftarrow w - \eta \nabla f(w) = w - \eta \Sigma w$$

这里我们省略了时间步的下标 t，并使用坐标的下标。等价地，我们可以写出每个坐标的更新规则

$$w_i \leftarrow w_i - \eta \lambda_i w_i = (1 - \lambda_i \eta_i) w_i$$

现在我们看到，如果对于某些 i，有 $\eta >$^2/λ_i，那么 w_i 的绝对值将呈指数级爆炸并导致不稳定行为。因此，我们需要 $\eta \lesssim \frac{1}{\max \lambda_i}$。注意，$\max \lambda_i$ 对应于 f 的平滑度参数，因为 λ_1 是 $\nabla^2 f = A$ 的最大特征值。这与引理2.1.4中的条件一致，即 η 需要很小。

假设为了简化，我们设 η = = 1/(2λ_1)，那么我们可以看到 w_1 坐标的收敛非常快——坐标 w_1 在每次迭代中减半。然而，坐标 w_d 的收敛较慢，因为它每次只减少一个因子 $(1 - \lambda_d/(2\lambda_1))$。因此，需要 $O(\lambda_1/\lambda_d \cdot \log(1/\epsilon))$ 次迭代才能收敛到一个误差 ϵ。这里的分析可以扩展到一般的凸函数，这也反映了以下原则：

条件数定义为 $\kappa = \sigma_{\max}(A)/\sigma_{\min}(A) = \lambda_1/\lambda_d$。它控制了GD的收敛速度。

≪Tengyu 笔记：添加图 ≫

2.4.1 *Pre-conditioners*

从上面的玩具二次示例中，我们可以看到，如果我们能为不同的坐标使用不同的学习率，那将是非常理想的。换句话说，如果我们为每个坐标引入一个学习率 η_i = $1/\lambda_i$，那么我们可以实现更快的收敛。在更一般的设置中，其中 A 不是对角线，我们事先不知道坐标系，该算法对应于

$$w \leftarrow w - A^{-1} \nabla f(w)$$

在更加一般的设置中，其中 f 不是二次的，这对应于牛顿算法

$$w \leftarrow w - \nabla^2 f(w)^{-1} \nabla f(w)$$

计算Hessian $\nabla^2 f(w)$ 可能计算困难，因为它在 d (中呈二次增长，实践中可能超过 100 万)。因此，使用Hessian及其逆的近似：

$$w \leftarrow w - \eta Q(w)\nabla f(w)$$

在 $Q(w)$ 应该是一个好的 $\nabla^2 f(w)$ 近似的情况下，有时被称为预处理器。在实践中，人们通常首先用对角矩阵来近似 $\nabla^2 f(w)$，然后取其逆。例如，在 Adagrad 中，使用 diag($\nabla f(w)\nabla f(w)^\top$) 的最近值的加权平均来近似 Hessian，然后使用对角矩阵的逆作为预处理器（见第 2.5.4 节）。

2.5 *Convergence rates under smoothness conditions*

如第2章所述，基于梯度的方法通常无法找到低次多项式等简单函数的最优值。但我们确实注意到，如果函数可导且光滑，那么只要梯度不为零，损失就会单调递减，学习率足够小。换句话说，最终会得到一个 *stationary point,*，其中 $\nabla = 0$。本章建立了接近平衡点的上界。有关收敛速度到更强类型解（局部最优解）的分析，请参阅第7章。

通常，目标/损失函数表示为 $f(w)$，其中 $w \in \Re^d$。该过程有 T 次迭代，这些迭代中的参数向量分别表示为 w_1，...，w_T。我们假设 *boundedness*：即存在一个已知的 M，使得对于所有 $t = 1$，...，T，有 $|f(w_t)| \leq \frac{M}{2}$。我们还假设 f 是 β-平滑的，即

$$f(w) \leq f(w') + \nabla f(w')(w - w') + \frac{\beta}{2}\|w - w'\|^2. \tag{2.3}$$

在整个章节中，∇_t是$\nabla f(w_t)$的简称。
这是 β 与本章其他地方 L 相同。

2.5.1 *Lower bounds, and the need for smoothness*

在约束非凸优化中，最小化梯度带来了困难的计算挑战。一般来说，即使目标函数是有界的，局部信息也可能无法提供关于驻点位置的信息。

例如，考虑图2.2中绘制的函数。在这个在$\mathbb{R}^n$超立方体上定义的构造中，存在一个唯一的点

消失梯度的局部极小值是一个隐藏的谷地，而这个谷地外的梯度都是相同的。显然，从信息论的角度来看，要有效地找到这个点是无望的：要发现这个谷地，这个函数的值或梯度评估的数量必须是exp($\Omega(n)$)。

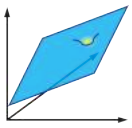


图2.2：非凸优化的一个困难的“针插麦堆”案例。一个具有隐藏谷的函数，黄色显示小梯度。

为了绕过这种固有的困难和退化的情况，我们需要目标函数是光滑的。正如我们将看到的，这允许找到具有小梯度的点的有效算法。

2.5.2 *Convergence rates for GD*

本节分析给定精确梯度的梯度下降。下一节分析随机GD。

算法 1 梯度下降

1: 输入：f，T，初始点 $w_1 \in$ K，步长序列 $\{\eta_t\}$ 2: for $t = 1$ 到 T do 3: 令 $w_{t+1} = w_t - \eta_t \nabla f(w_t)$ 4: end for 5: 返回 w_τ，$\tau \in [T$] 使得 ∇_τ 在欧几里得范数下最小。

定理2.5.1. *For unconstrained minimization of β-smooth functions and $\eta_t = \frac{1}{\beta}$, Algorithm 1 satisfies*

$$\|\nabla_\tau\|^2 \leq \frac{1}{T}\sum_t \|\nabla_t\|^2 \leq \frac{4M\beta}{T}.$$

Proof. 表示 $h_t = f(w_t) - f(w^*)$。下降引理在以下简单方程中给出，

$$\begin{aligned}
h_{t+1} - h_t &= f(w_{t+1}) - f(w_t) \\
&\leq \nabla_t^\top (w_{t+1} - w_t) + \frac{\beta}{2}\|w_{t+1} - w_t\|^2 && \beta\text{-smoothness} \\
&= -\eta_t \|\nabla_t\|^2 + \frac{\beta}{2}\eta_t^2 \|\nabla_t\|^2 && \text{algorithm defn.} \\
&= -\frac{1}{2\beta}\|\nabla_t\|^2 && \text{choice of } \eta_t = \frac{1}{\beta}
\end{aligned}$$

因此，对 T 次迭代求和，我们得到

$$\frac{1}{2\beta}\sum_{t=1}^{T}\|\nabla_t\|^2 \le \sum_t (h_t - h_{t+1}) = h_1 - h_{T+1} \le 2M$$

□

2.5.3 *Stochastic gradient descent*

在机器学习的优化中，目标函数 f 的形式为

$$f(w) = \frac{1}{m}\sum_i \ell(w, z_i),$$

在 z_i，$i \in [m]$ 是训练集示例，且 ℓ 是应用于参数 w 和数据点 z_i 的某个损失函数的情况下，随机梯度下降的关键思想是可以使用具有相同期望的随机变量来代替梯度。这个随机变量简单地是来自训练集的小 *batch* 个示例的平均梯度。下面的分析甚至允许批量大小为 1（参见问题 2.5.3）。

我们用 $\widehat{\nabla}_t$ 表示一个随机变量，满足 $\mathbb{E}[\widehat{\nabla}_t] = \nabla f(w_t) = \nabla_t$ (，其中期望是在梯度估计中使用的随机性上求的，并且对随机变量的二阶矩有一个界限：

$$\mathbb{E}[\|\widehat{\nabla}_t\|^2] = \sigma^2. \tag{2.4}$$

算法 2 随机梯度下降

1: 输入：f，T，初始点 $w_1 \in$ K，步长序列 $\{\eta_t\}$ 2: for $t = 1$ 到 T do 3: 令 $w_{t+1} = w_t - \eta_t\widehat{\nabla}_t$ 4: end for 5: 返回 w_τ，$\tau \in [T]$ 使得 ∇_τ 在欧几里得范数下最小。

定理2.5.2. *For unconstrained minimization of β-smooth functions and $\eta_t = \eta = \sqrt{\frac{M}{\beta\sigma^2 T}}$, Algorithm 2 satisfies*

$$\mathbb{E}[\|\nabla_\tau\|^2] \le \mathbb{E}\left[\frac{1}{T}\sum_t \|\nabla_t\|^2\right] \le 2\sqrt{\frac{M\beta\sigma^2}{T}}.$$

Proof. 用 ∇_t 表示 $\nabla f(w_t)$ 的简称，以及 $h_t = f(w_t) - f(w^*)$。随机下降引理如下公式给出：

翻译文本：tion,

$$\begin{aligned}\mathbb{E}[h_{t+1} - h_t] &= \mathbb{E}[f(w_{t+1}) - f(w_t)] \\ &\leq \mathbb{E}[\nabla_t^\top (w_{t+1} - w_t) + \frac{\beta}{2}\|w_{t+1} - w_t\|^2] && \beta\text{-smoothness} \\ &= -\mathbb{E}[\eta \nabla_t^\top \widetilde{\nabla}_t] + \frac{\beta}{2}\eta^2 \mathbb{E}\,\|\widetilde{\nabla}_t\|^2 && \text{algorithm defn.} \\ &= -\eta\|\nabla_t\|^2 + \frac{\beta}{2}\eta^2\sigma^2 && \text{variance bound.}\end{aligned}$$

因此，对 T 次迭代求和，我们得到 $\eta = \sqrt{\frac{M}{\beta\sigma^2 T}}$，

$$\begin{aligned}\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla_t\|^2\right] &\leq \tfrac{1}{T\eta}\textstyle\sum_t \mathbb{E}\,[h_t - h_{t+1}] + \eta\tfrac{\beta}{2}\sigma^2 \leq \tfrac{M}{T\eta} + \eta\tfrac{\beta}{2}\sigma^2 \\ &= \sqrt{\tfrac{M\beta\sigma^2}{T}} + \tfrac{1}{2}\sqrt{\tfrac{M\beta\sigma^2}{T}} \leq 2\sqrt{\tfrac{M\beta\sigma^2}{T}}.\end{aligned}$$

□

因此，我们得出结论，需要 $O(\frac{1}{\epsilon^4})$ 次迭代才能找到一个具有 $\|\nabla f(w)\| \leq \epsilon$ 的点。然而，每次迭代只需要梯度的随机估计，因此在实践中，随机梯度下降（SGD）要快得多。

问题 2.5.3. *Suppose the gradient is estimated using a random sample of B datapoints. (a) Let $\widetilde{\nabla}_t^{(B)}$ be the stochastic gradient at time t when the batchsize is B. Suppose the variance of $\widetilde{\nabla}_t^{(1)}$ (defined as $\mathbb{E}\left[\left\|\widetilde{\nabla}_t^{(1)} - \nabla_t\right\|^2\right]$ is bounded by γ_1^2. Show that there exists an upper bound γ_B^2 on the variance of $\widetilde{\nabla}_t^{(B)}$ that scales with 1/B. (b) Compute the asymptotic size of T to find a point with $\|\nabla f(w)\| \leq \epsilon$ depending on B and ϵ. For simplicity, you only need to consider the case when $\eta \leq \frac{1}{\beta}$.*

2.5.4 *Adaptive Algorithms and AdaGrad*

自适应方法保留一些来自过去更新的信息，并使用这些信息来修改基本梯度步骤。一个简单的例子，*momentum*，在第2章中简要讨论过。自适应方法需要更多空间来存储它们的参数，通常每个梯度中的d坐标需要2或3个参数。但它们可以更快地收敛，以及在深度学习设置中具有其他神秘的属性，这些属性在数学上尚未理解。

待定：描述rmsprop、adam

一些这些算法即使对于凸损失也不能保证收敛。我们分析了AdaGrad [4]，它是以下算法的先驱：

现代自适应算法确实有一个收敛性的证明。

4 J. Duchi, E. Hazan, 和 Y. Singer. 自适应子梯度方法在在线学习和随机优化中的应用。*Journal of Machine Learning Research*, 2011

算法 3 AdaGrad

对于 $t = 1$ 到 T 执行

输入：矩阵/标量 P_t，如下

集合

$$w_{t+1} = w_t - P_t \widehat{\nabla}_t$$

结束for

返回 w_τ，$\tau \in [T]$，使得 ∇_τ 在欧几里得范数下最小。

我们首先对自适应步长进行简单分析，如下定理所示。在本节中，除了上述符号外，我们还使用缩写符号 $\nabla_{1:t} = \sum_{i=1}^{t} \nabla_i$，并令 $G \geq \|\nabla_t\|$ 为梯度范数的上界。

定理2.5.4. *For unconstrained minimization of β-smooth functions and* $P_t = \|\widehat{\nabla}_{1:t}^2\|^{-1} \cdot I$*, Algorithm 3 satisfies*

$$\mathbb{E}[\|\nabla_\tau\|^2] \leq \mathbb{E}\left[\frac{1}{T}\sum_t \|\nabla_t\|^2\right] \leq \frac{(\beta + M\log GT)\cdot\|\widehat{\nabla}_{1:t-1}^2\|}{T}.$$

Proof. 从下降引理：

$$\begin{aligned} -M &\leq f(w_{T+1}) - f(w_1) \\ &= \textstyle\sum_t (f(w_{t+1}) - f(w_t)) \\ &\leq \textstyle\sum_t (\nabla_t^\top (w_{t+1} - w_t) + \frac{\beta}{2}\|w_t - w_{t+1}\|^2) \quad \text{smoothness} \\ &\leq \textstyle\sum_t (-\nabla_t^\top P_t \widehat{\nabla}_t + \frac{\beta}{2}\widehat{\nabla}_t^\top P_t^2 \widehat{\nabla}_t) \end{aligned}$$

让 $P_t = \|\widehat{\nabla}_{1:t}^2\|^{-1}$，并让 $\sigma^2 \geq \|\widehat{\nabla}_t\|^2$ 为随机梯度的二阶矩的上界。然后注意到，通过调和级数，

$$\sum_t \widehat{\nabla}_t^\top P_t^2 \widehat{\nabla}_t = \sum_t \frac{\|\widehat{\nabla}_t\|^2}{\|\widehat{\nabla}_{1:t}^2\|^2} = \sum_t \frac{\|\widehat{\nabla}_t\|^2}{\sum_{i=1}^t \|\widehat{\nabla}_i^2\|^2} \leq \log GT$$

使用此不等式在之前的推导中，我们得到：{v*}

$$\sum_t \nabla_t^\top P_t \widehat{\nabla}_t \leq M + \frac{\beta}{2}\log GT.$$

取最小值LHS，我们得到

$$\nabla_\tau^\top \widehat{\nabla}_\tau \cdot P_{\tau-1} \leq \nabla_\tau^\top \widehat{\nabla}_\tau \cdot P_\tau \leq \frac{(\beta + M\log GT)}{T}.$$

对无偏梯度估计器的期望进行取值，并移项，我们得到

$$\|\nabla_\tau\|^2 \leq \frac{(\beta + M\log GT)\cdot\|\widehat{\nabla}_{1:t-1}^2\|}{T}.$$

□

2.5.5 *Adagrad Convergence: Diagonal Matrix case*

定理2.5.5. *For unconstrained minimization of β-smooth functions and* $P_t = diag(\sum_{i=1}^{t-1} \widehat{\nabla}_i \widehat{\nabla}_i^\top + \sigma^2 I)^{-1/2}$*, Algorithm 3 satisfies*

$$\mathbb{E}[\|\nabla_\tau\|^2] \le \mathbb{E}\left[\frac{1}{T}\sum_t \|\nabla_t\|^2\right] \le (M + \beta \log GT) \cdot \frac{\sum_j \sqrt{\widehat{\nabla}_{1:t}^2(j)}}{T}.$$

Proof. 从下降引理:

$$\begin{aligned} M &\ge f(w_1) - f(w_{T+1}) \\ &= \textstyle\sum_t (f(w_t) - f(w_{t+1})) \\ &\ge \textstyle\sum_t (\nabla_t^\top (w_t - w_{t+1}) - \frac{\beta}{2}\|w_t - w_{t+1}\|^2) \quad \text{smoothness} \\ &= \textstyle\sum_t (\nabla_t^\top P_t \widehat{\nabla}_t - \frac{\beta}{2}\widehat{\nabla}_t^\top P_t^2 \widehat{\nabla}_t). \end{aligned}$$

考虑条件期望，以及 P_t 的定义，它是与 $\widehat{\nabla}_t$ 条件独立的，我们得到

$$\begin{aligned} M &\ge \textstyle\sum_{i=1}^d \left\{\sum_t (\nabla_t^2(i) P_t(i) - \frac{\beta}{2}\widehat{\nabla}_t^2(i) P_t^2(i))\right\} \\ &\ge \textstyle\sum_{i=1}^d \left\{\sum_t (\nabla_t^2(i) P_t(i) - \frac{\beta}{2}\log \sigma^2 T)\right\} \\ &\ge \textstyle\max_{i=1}^d \sum_t \nabla_t^2(i) P_t(i) - \frac{\beta}{2}\log \sigma^2 T, \end{aligned}$$

在第二个不等式是由于调和级数,

$$\sum_t \widehat{\nabla}_t^2(i) P_t^2(i) = \sum_t \frac{\widehat{\nabla}_t^2(i)}{\widehat{\nabla}_{1:t-1}^2(i) + \sigma^2} \le \sum_t \frac{\widehat{\nabla}_t^2(i)}{\widehat{\nabla}_{1:t}^2(i)} \le \log \sigma^2 T.$$

我们得出结论，任何 j,

$$\sum_t \nabla_t^2(j) P_t(j) \le \max_i \sum_t \nabla_t^2(i) P_t(i) \le M + \frac{\beta}{2}\log \sigma^2 T.$$

设 c_j 为一个随机变量，其等于 $\nabla_t^2(j)$ 的概率为 $\frac{1}{T}$。那么上述公式意味着

$$\mathbb{E}[c_j] \le \frac{M + \frac{\beta}{2}\log \sigma^2 T}{T P_t(j)} = (M + \beta \log GT) \cdot \frac{\sqrt{\widehat{\nabla}_{1:t}^2(j)}}{T}.$$

因此，对坐标 j 求和，我们得到

$$\mathbb{E}\,\|\nabla_\tau^2\|^2 = \mathbb{E}[\sum_j c_j] \le (M + \beta \log GT) \cdot \frac{\sum_j \sqrt{\widehat{\nabla}_{1:t}^2(j)}}{T}.$$

□

2.6 *Correspondence of theory with practice*

现在我们描述上述理论与现实之间的比较。结果发现，关于固定且已知的平滑度的假设（在第六章、第七章以及各种其他地方也被使用）在今天的深度学习环境中是有问题的。

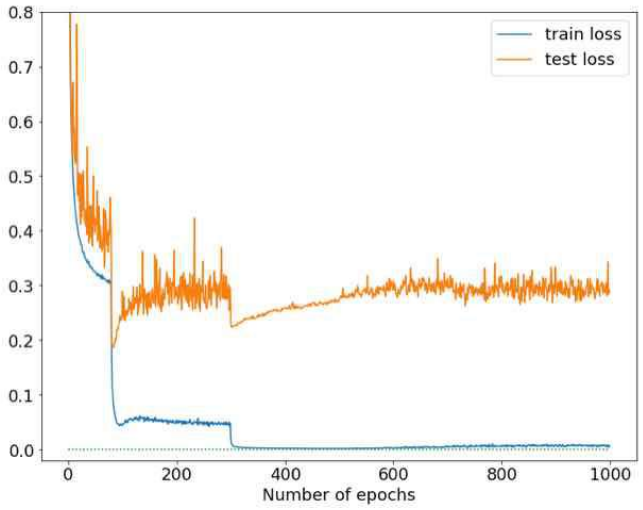


图2.3：在CIFAR10上对PreResNet 32进行SGD训练和测试损失的行为（50k数据点）。测试损失通过一个固定的保留数据集在各个步骤进行估计。初始学习率为1，在80个和300个epoch时减少了10倍。（一个epoch是对训练数据集的完整遍历，其中数据集已被随机分成批次用于SGD–在这种情况下，批次大小为128。）注意训练损失和测试损失之间的复杂关系，特别是，即使训练损失平稳或下降，测试损失也可能略有上升。

Setting learning rate 各种算法在本章中使用平滑度参数设置学习率。不幸的是，对于参数向量空间 *entire* 上的损失函数的平滑度参数进行估计并不容易。然而，对于特定的参数向量 w，可以估计 Hessian 矩阵 $H = \nabla^2(f)$ 在 w 处的最大特征值。这使用了 *power method*，它从高斯单位向量 u 开始，并重复计算 $u \leftarrow Hu/\|Hu\|_2$。(每个迭代都是通过第 4 章中描述的 Hessian-向量积计算有效地实现的。)这种方法被称为幂方法，因为它实际上相当于计算 $H^t x/\|H^t x\|_2$，这可以很容易地检查收敛到一个向量，该向量是对应于 H 的最大特征值（绝对值）的特征向量的组合。特别是，如果 v 是最终向量，$v^T Hv$ 将是对平滑度的良好近似。

当然，这只能得到特定参数向量 w 的平滑度参数。当 w 变化时，平滑度也可能发生变化。频繁地重新计算平滑度将非常耗费计算资源。在实践中，学习率被启发式地设置为某个值。如果GD在几次迭代后没有降低损失，则学习率会通过一个小的因子（如2或5）降低。在后面的章节中，我们将重新探讨学习率的问题。

Edge of stability phenomenon. 上述关于学习率的阐述在经典优化理论中是标准的——它将平滑度 L 作为已知条件，并描述了如何设置学习率小于 $2/L$ 以确保损失持续下降。最近的一篇论文 5 给出

证据表明，在深度网络中，车在马之前，因此

[5] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *ICLR*, 2021

说。换句话说，如果我们把学习率设为某个小的 η，则平滑度 *adjusts* 会迅速降至约 $2/\eta$。这个“稳定性边缘”阶段似乎对良好的最终性能很重要。

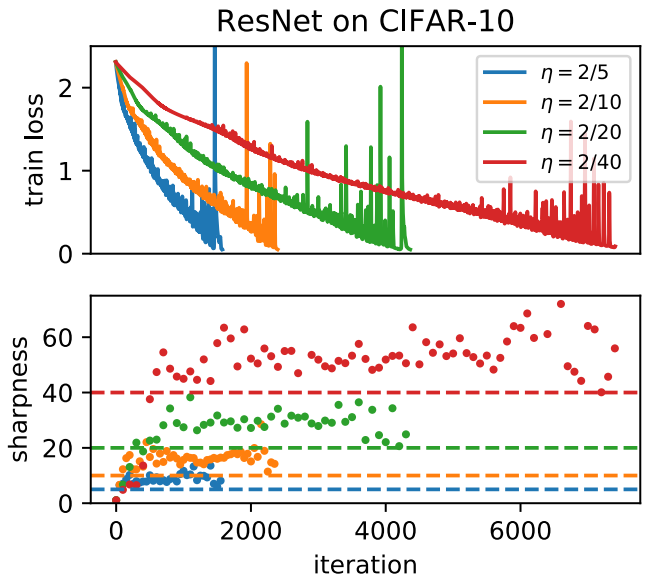


图2.4：稳定性边缘现象。Cohen等人2021年的第5图显示了在5k个示例上训练的ResNet。在进行GD（与SGD相反）时，使用小的学习率η，观察到平滑度上升到$2/\eta$并略有超出（右图）。在此之后，可以看到损失在迭代中上下波动，呈现长期下降趋势。目前尚无理论解释。作者使用“尖锐度”代替平滑度，这实际上有些道理，因为更高的L对应于更不均匀的地形。

3
Note on overparametrized linear regression and kernel regression

本简短部分分析了非常经典模型的梯度下降：最小二乘线性回归。问题是凸的，优化效果良好。我们主要关注欠确定版本，其中存在无限多个零损失解，有趣的问题是：梯度下降找到了什么？我们使用伪逆找到了一个优雅的精确分析。分析也扩展到核最小二乘回归。

尽管是经典的，但这些分析是理解过度参数化的深度网络（在第9章和第8章中描述）努力的起点，这些网络也有大量的低成本解决方案，我们希望了解哪些是通过梯度下降和相关算法找到的。

3.1 Overparametrized least squares linear regression

如第1章所述，我们假设我们的训练数据由 n 数据点组成，每个数据点独立地从分布 $\mathcal{D}$ 中抽取，

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}.$$

这里 $x^{(i)} \in \Re^d$ 和 $y^{(i)} \in \Re$。写 ℓ_2 损失将很方便

$$\widehat{L}(w) = \frac{1}{2} \sum_i (x^{(i)} \cdot w - y^{(i)})^2,$$

使用矩阵表示法，如 $\widehat{L}(w) = \frac{1}{2}\|Xw - y\|^2$，其中 X 是行由 $x^{(i)}$ 组成的矩阵，w 是列向量，y 是列向量，其 i 个元素是 $y^{(i)}$。我们感兴趣的是 $x^{(i)}$ 独立且 $d > n$ 的情况。在这种情况下，损失函数有无限多个最小值，所有这些最小值都达到零训练损失。那么梯度是什么?

下降找到?

1 您可以通过注意到在将 w 中的任意子集 $d - n$ 坐标设置为零后存在一个可行解来验证这一点。

为了简单起见，将梯度下降的起始点初始化为 $w_0 = 0$。任何 w 处的梯度是 $\nabla\widehat{L}(w) = X^T(Xw - y)$。因此，学习率为 η 的梯度下降给出以下轨迹

$$w_{t+1} = w_t - \eta X^\top (Xw_t - y) \tag{3.1}$$
$$= (I_{d\times d} - \eta X^\top X)w_t + \eta X^\top y \tag{3.2}$$
$$= (\sum_{j=0}^{t}(I_{d\times d} - \eta X^\top X))^j \eta X^\top y \tag{3.3}$$

假设通过试错法使 η 足够小，特别是 $\eta < 1/\lambda_{max}(X^TX)$，上述方程中的无穷级数 $t \to \infty$ 收敛于 2

$$\lim_{t\to\infty} w_t = (X^\top X)^\dagger X^\top y \tag{3.4}$$
$$= X^\top (XX^\top)^{-1} y \tag{3.5}$$

2 我们使用 $\sum_{i\geq 0} A^i = (I - A)^\dagger$，当正半定矩阵 A 的最大特征值小于 1，且 $Z^\dagger$ 表示 Z 的伪逆。此外，方程 (3.5) 的第二个等式可以通过使用 X 的奇异值分解来验证。

当然，这是著名的 *pseudo-inverse* 解法，用于过定线性方程组。此解法也是拟合数据的使 ℓ_2-范数最小化的解：$\arg\min_{w\in\Re^d} \|w\|_2$ s.t. $Xw = y$ 。

3.1.1 SVD and Matrix pseudo-inverse

矩阵的逆仅定义于满秩的方阵。上述例子说明了我们还需要为非方阵以及秩不足的方阵定义逆的概念。*Moore-Penrose*伪逆是在20世纪基于这些动机定义的。为了简单起见，我们描述这个理论对于实$m \times n$矩阵，这些矩阵已知具有如下形式的*singular value decomposition* (SVD):

$$M = \sum_{i=1}^{k} \sigma_i u_i v_i^T, \tag{3.6}$$

k 是 M 的秩，σ_i 是奇异值，$u_i \in \Re^m$，$v_i \in \Re^n$ 是列向量。

pseudo-inverse 是按如下定义的 $n \times m$ 矩阵 $M^\dagger$，其中符号假定所有 σ_i 都不为零:

$$M^\dagger = \sum_{i=1}^{k} \frac{1}{\sigma_i} v_i u_i^T \tag{3.7}$$

一个特殊情况是当 M 对称 $m \times m$ 时，在这种情况下，SVD 有 $v_i = u_i$ 并称为 *spectral decomposition*。

问题 3.1.1. *Show that* $MM^\dagger M = M$ *and* $M^\dagger MM^\dagger = M$ †.

问题 3.1.2. *Prove the properties mentioned in (3.5).* 3

3 提示：如果 M 是对称的，则 M^j 的特征值是 M 的特征值的 j 次幂。此外，特征向量构成一个正交归一基。

3.2 *Kernel least-squares regression*

内核模型涉及数据空间 $\mathcal{X}$ 的一种新表示，该表示将数据点 $x \in \mathcal{X}$ 表示为 $\phi(x)$，其中 ϕ 是从 $\Re^d$ 到称为“再生”的适当希尔伯特空间[4] 的映射

希尔伯特空间是向量空间向无限维的推广，内积被良好定义。

内核希尔伯特空间”或RKHS。这意味着对于每个x，$y \in \mathcal{X}$，内积 $\phi(x) \cdot \phi(y)$都是良好定义的。通常将内积表示为$K(x, y)$，这被称为核函数。感兴趣的函数类是在变换特征$h_w(x) = \phi(x) \cdot w$上的线性模型。数学和数据科学中已知有大量的有用核函数。

示例3.2.1. *The Kernel* $K(x, y) = (1 + x \cdot y)^2$ *corresponds to representing* $x = (x_1, x_2, \ldots, x_d)$ *by*

$$\phi(x) = (1, \sqrt{2}x_1, \ldots, \sqrt{2}x_d, \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_1x_d, \sqrt{2}x_2x_3, \ldots, \sqrt{2}x_2x_d, \ldots, \sqrt{2}x_{d-1}x_d, x_1^2, x_2^2, \ldots, x_d^2).$$

You can verify that $\phi(x) \cdot \phi(y) = K(x, y)$。

在数据科学中，所需的关键属性是内积 $K(x_1、x_2)$ 可以高效计算。实际上，在实践研究中，研究人员通过从这种高效可计算性属性开始来设计核，而从不考虑底层表示 $\phi(\cdot)$，因为那在训练中不起作用。

问题 3.2.2. *Suppose datapoints are unit vectors in* $\Re^d$*. Find an infinite dimensional representation* $\varphi()$ *that realizes the following kernels. (a) (polynomial kernel)* $K(x_1, x_2) = (1 + (x \cdot y)^d)$*. (b) (Gaussian Kernel)* $K(x_1, x_2) = \exp(-\|x - y\|^2)$*. (c) (Laplace Kernel)* $K(x_1, x_2) = \exp(\|x_1 - x_2\|_2) = \exp(\sqrt{1 - 2x_1 \cdot x_2})$*. (Hint for (b) and (c): look at the Taylor expansion of* $K()$*.)*

现在让我们看看如何高效地解决以下核回归问题。

$$\ell(w) = \frac{1}{2}\sum_i (\phi(x)^{(i)} \cdot w - y^{(i)})^2.$$

当 $\phi(x)$ 是无限维的，我们很快意识到早期对过参数化回归的分析适用。计算数据矩阵 $XX^\top$ 只需要内积，这在 RKHS 中是良好定义的，因此数据矩阵变成了一个 $n \times n$ *gram matrix* G，其中 $G_{ij} = \phi(x^{(i)}) \cdot \phi(x^{(j)}) = K(x^{(i)}, x^{(j)})$ 实际上计算 G 允许在不显式计算 $\phi(x^{(i)})$ 的情况下进行梯度下降。表达式 (3.5) 显示梯度下降以一个分类器 $h(x) = \lim_{t\to\infty} w_t.\phi(x)$ 结束，它将一个输入点 $x \in \mathcal{X}$ 映射到

$$h(x) = z^T G^{-1} y \tag{3.8}$$

在 z 是其 i 个坐标为 $K(x, x_i)$ 的列向量的情况下，或者表示为 $\alpha = G^{-1}y \in \Re^n$，方程 (3.8) 中的解可以另视为在训练点上的核评估的加权组合，

$$h(x) = \sum_i \alpha_i K(x, x_i). \tag{3.9}$$

表达式（3.9）等价于在拟合核回归目标的同时最小化w的ℓ_2范数，这在函数空间中对应于关于核K的最小RKHS范数解。

4
Note on Backpropagation and its Variants

全书我们依赖于计算损失相对于模型参数的梯度。对于深度网络，这种计算是通过反向传播来完成的，这是一个使用微积分链式法则的简单算法。为了方便，我们更普遍地描述它为计算神经网络输出对其所有参数的敏感性，即$\partial f/\partial w_i$，其中f是输出，w_i是第i个参数。在这里*parameters*可以是与网络节点或边相关的边权重或偏差。这种基本算法的版本显然在1960年代到1980年代期间在几个领域被独立地重新发现了几次。本章介绍了这个算法以及一些涉及不仅仅是梯度，还包括海森矩阵的先进变体。

在本书的大部分内容中，感兴趣的量是训练损失的梯度。但上述表述——计算输出相对于输入的梯度——是完全通用的，因为可以简单地向网络添加一个新输出节点，该节点从旧输出计算训练损失。然后，感兴趣的量确实是这个新输出相对于网络参数的梯度。

反向传播的重要性源于其效率。假设节点操作需要单位时间，运行时间是 *linear*，具体来说，是 $O(\text{网络大小}) = O(V + E)$，其中 V 是网络中的节点数，E 是边的数量。正如计算机科学中的许多其他设置——例如，排序数字——朴素算法将需要平方时间，这对于今天的大型网络来说将非常低效，甚至不可行。

4.1 Problem Setup

反向传播仅适用于具有有向边的无环网络。（它可以像后面所描述的那样启发式地应用于具有环的网络。）不失一般性，无环网络可以是

以分层结构可视化，第 $t+$ 层的节点从 t 层及之前的节点输出中获取所有输入。我们用 $f \in \mathbb{R}$ 表示网络的输出。在我们所有的图中，网络的输入位于底部，输出位于顶部。

我们的阐述使用了概念 $\partial f / \partial u$，其中 f 是输出，u 是网络中的一个节点。这意味着以下内容：假设我们切断节点 u 的所有入边，并固定/夹紧所有网络参数的当前值。现在想象将 u 从其当前值改变。这种变化可能会影响连接到 u 的较高级别节点的值，最终输出 f 就是这样的一个节点。然后 $\partial f / \partial u$ 表示当我们改变 u 时，f 将如何变化的速率。（旁白：熟悉常规反向传播阐述的读者应注意到，在那里 f 是训练误差，而 $\partial f / \partial u$ 结果恰好是传播到节点 u 的“误差”。）

主张4.1.1. *To compute the desired gradient with respect to the parameters, it suffices to compute $\partial f / \partial u$ for every node u.*

Proof. 从链式法则的直接应用中得出，我们通过图片证明它，即图4.1。假设节点u是节点z_1、...、z_m (的加权求和，之后将通过一个非线性激活σ和)。也就是说，我们有$u = w_1 z_1 + \cdots + w_n z_n$。根据链式法则，我们有

$$\frac{\partial f}{\partial w_1} = \frac{\partial f}{\partial u} \cdot \frac{\partial u}{\partial w_1} = \frac{\partial f}{\partial u} \cdot z_1.$$

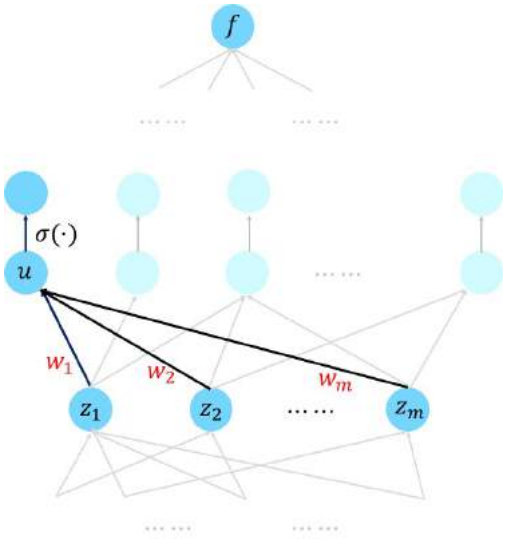


图4.1：为什么只需要计算关于节点的导数就足够了。

因此，我们看到，在计算了 $\partial f / \partial u$ 之后，我们可以计算

$\partial f/\partial w_1$，并且此外这可以通过居住在 w_1 的边端点本地完成。

□

4.1.1 *Multivariate Chain Rule*

关于计算节点导数，我们首先回顾多元链式法则，它方便地描述了这些偏导数之间的关系（取决于图结构）。

假设一个变量 f 是变量 u_1，...，u_n 的函数，而这些变量又依赖于变量 z。那么，多元链式法则表明

$$\frac{\partial f}{\partial z} = \sum_{j=1}^{n} \frac{\partial f}{\partial u_j} \cdot \frac{\partial u_j}{\partial z} .$$

为了说明，我们在图4.2中将它应用于之前使用的相同示例，但具有不同的焦点和节点编号。

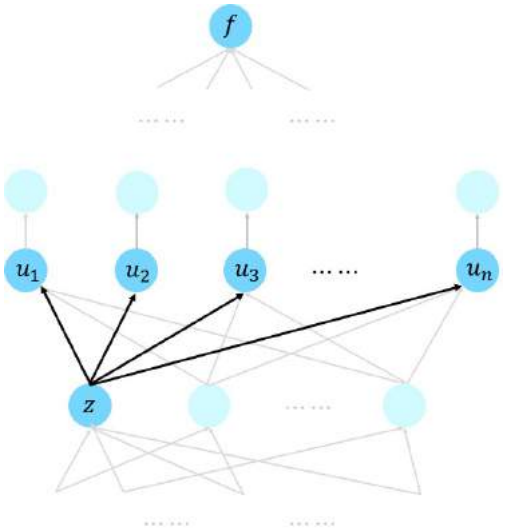


图4.2：多元链式法则：对节点 z 的导数可以通过对所有 z 输入节点的导数的加权求和来计算。

我们注意到，在计算了相对于所有位于节点 z 上方的节点的导数之后，我们可以通过加权求和来计算相对于节点 z 的导数，其中权重涉及通常容易计算的局部导数 $\partial u_j/\partial z$。

这让我们来到了如何衡量运行时间的问题。为了记账，我们假设

Basic assumption: 如果 u 是第 $t+1$ 级的节点，并且 z 是任何输出为 u 输入的 $\leq t$ 级的节点，那么在我们的计算机上计算 $\frac{\partial u}{\partial z}$ 需要单位时间。

4.1.2 *Naive feedforward algorithm (not efficient!)*

首先指出由链式法则隐含的朴素二次时间算法是有用的。大多数作者跳过了这个平凡的版本，我们认为这类似于只使用快速排序来教授排序，而忽略了效率较低的冒泡排序。

朴素算法是计算每个节点对 $\partial u_i/\partial u_j$，其中 u_i 位于比 u_j 更高的层级。当然，在这些 V^2 值（其中 V 是节点数）中，也有所有 $\partial f/\partial u_i$ 的期望值，因为 f 本身就是输出节点的值。

此计算可以以前馈方式完成。如果已经为直到并包括级别 t 的每个 u_j 获得了这样的值，那么可以通过检查多元链式法则来表示某些 u_ℓ 在级别 $t+1$ 的值 $\partial u_\ell/\partial u_j$ 作为每个直接输入到 u_ℓ 的 u_i 的值 $\partial u_i/\partial u_j$ 的加权组合。这种描述表明，对于固定的 j，计算量与边的数量 E 成正比。这种工作量对所有 $j \in V$ 都发生，使我们得出结论，算法中的总工作量是 $O(VE)$。

4.2 *Backpropagation (Linear Time)*

更高效的反向传播，正如其名所示，按反向方向计算偏导数。消息以一波的形式从编号较高的层向后传递到编号较低的层。（一些算法的描述将其描述为动态规划。）

算法 4 反向传播

节点 u 从该边的另一端的节点接收沿着每条出边的信息。它将这些信息相加以获得一个数字 S (如果 u 是整个网络输出，那么定义 $S = 1$) 然后它将以下信息发送到任何比它低一级的相邻节点 z:

$$S \cdot \frac{\partial u}{\partial z}$$

显然，每个节点所做的功与其度成正比，因此总功是节点度的总和。将所有节点度相加会导致每条边被重复计算一次，因此总功是 O(网络大小)。

为了证明正确性，我们证明以下内容:

引理 4.2.1. *At each node z, the value S is exactly $\partial f/\partial z$.*

Proof. 从深度简单归纳得出。

Base Case: 在输出层这是正确的，因为 $\partial f/\partial f = 1$. 假设对于 $t+1$ 层及更高层，该命题是正确的，并且 u 位于 t 层，其出边指向某些位于 $t+1$ 层或更高层的节点 u_1，u_2，...，u_m。通过归纳假设，节点 z 确实从每个 u_j 接收 $\frac{\partial f}{\partial u_j} \times \frac{\partial u_j}{\partial z}$。因此，根据链式法则，

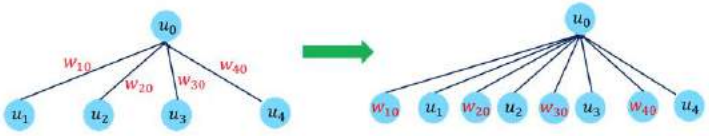
$$S = \sum_{i=1}^{m} \frac{\partial f}{\partial u_i} \frac{\partial u_i}{\partial z} = \frac{\partial f}{\partial z}.$$

这完成了归纳并证明了主要命题。 □

4.3 *Auto-differentiation*

自上述阐述几乎未涉及网络和节点执行的操作的细节，它扩展到可以组织为无环图且每个节点计算其入边邻居的可微函数的每个计算。这一观察结果是许多在深度学习环境中发现的自动微分包的基础：它们允许计算此类计算的输出相对于网络参数的梯度。

我们首先观察到，在非常一般的设置中，命题4.1.1仍然成立。这不会失去一般性，因为我们可以将与边相关的参数视为也位于节点上（实际上，是叶节点）。这可以通过对网络进行简单的转换来实现；对于单个节点，如图下所示；并且需要在从下面进入u_1、u_2等其余网络的其余部分继续进行这种转换。



然后，我们可以使用消息协议来计算关于节点的导数，只要局部偏导数可以有效地计算。我们注意到该算法可以以相当模块化的方式实现：对于每个节点 u，只需指定（a）它如何依赖于入节点，例如，z_1，...，z_n，以及（b）如何计算偏导数乘以 S，即 $S \cdot \frac{\partial u}{\partial z_j}$。

扩展到向量消息：事实上（b）可以在更一般的设置中高效完成，在这些设置中，我们允许网络中每个节点的输出是一个向量（甚至矩阵/张量），而不仅仅是实数。在这里，我们需要将$\frac{\partial u}{\partial z_j} \cdot S$替换为$\frac{\partial u}{\partial z_j}[S]$，它表示对$S$应用算子$\frac{\partial u}{\partial z_j}$的结果。我们注意到，为了与通常反向传播的常规表述中的约定保持一致，当$y \in \mathbb{R}^p$是$x \in \mathbb{R}^q$的函数时，我们使用$\frac{\partial y}{\partial x}$来表示具有$\partial y_j/\partial x_i$作为（$i$，$j$）-th条目的$q \times p$维矩阵。读者可能会注意到，这与数学中定义的通常的雅可比矩阵的转置相同。因此$\frac{\partial y}{\partial x}$是一个将$\mathbb{R}^p$映射到$\mathbb{R}^q$的算子，并且我们可以验证$S$与$u$和$\frac{\partial u}{\partial z_j}[S]$具有相同的维度。

例如，如下所示，假设节点 $U \in \mathbb{R}^{d_1 \times d_3}$ 是两个矩阵 $W \in \mathbb{R}^{d_2 \times d_3}$ 和 $Z \in \mathbb{R}^{d_1 \times d_2}$ 的乘积。那么我们有 $\partial U/\partial Z$ 是一个将 $\mathbb{R}^{d_2 \times d_3}$ 映射到 $\mathbb{R}^{d_1 \times d_3}$ 的线性算子，这直观上需要一个维度为 $d_2 d_3 \times d_1 d_3$ 的矩阵表示。然而，计算 (b) 可以高效地进行，因为

$$\frac{\partial U}{\partial Z}[S] = W^\top S.$$

此类向量运算也可以使用今天的GPU高效实现。

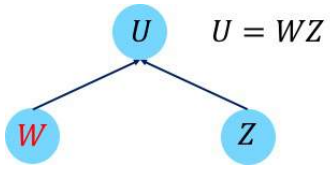


图4.3：上述向量化版本

4.4 *Notable Extensions*

Allowing weight tying: 在许多神经网络架构中，设计者希望强制许多网络单元（如边或节点）共享相同的参数。例如，在包括无处不在的卷积网中，必须在整个图像上应用相同的过滤器，这意味着在网络的两个层之间的大量边之间重用相同的参数。

为了简单起见，假设两个参数 a 和 b 应该具有相同的值。这相当于添加一个新节点 u，并将 u 通过操作 $a = u$ 和 $b = u$ 连接到 a 和 b。因此，根据链式法则，

$$\frac{\partial f}{\partial u} = \frac{\partial f}{\partial a} \cdot \frac{\partial a}{\partial u} + \frac{\partial f}{\partial b} \cdot \frac{\partial b}{\partial u} = \frac{\partial f}{\partial a} + \frac{\partial f}{\partial b}.$$

因此，等价地，关于共享参数的梯度是关于单个出现的梯度的总和。

Backpropagation on networks with loops. 上述阐述假设网络是无环的。许多尖端应用，如机器翻译和语言理解，使用具有有向循环的网络（例如，循环神经网络）。这些架构——以下是“可微计算”范例的所有例子——可能会变得复杂，可能涉及在单独的内存上执行操作，以及将注意力转移到数据的不同部分和内存中的机制。

网络中存在环路的网络也使用梯度下降进行训练，使用 *back-propagation through time*，它通过有限的时间步将网络扩展为无环图，并复制相同的网络。这些副本共享权重（权重绑定！），以便可以计算梯度。在实践中，可能会出现与 *exploding or vanishing gradients,* 相关的问题，这会影响收敛。这些问题可以通过剪切梯度或重新参数化技术（如 *long short-term memory* ）等手段在实践中进行仔细处理。最近的研究表明，仔细初始化参数可以缓解一些梯度消失问题。

该梯度可以高效地计算此类具有循环的通用网络的事实，激发了具有记忆或甚至数据结构的神经网络模型（例如，参见*neural Turing machines*和*differentiable neural computer*）。使用梯度下降，可以优化具有循环的参数化网络族，以找到解决特定计算任务（在训练示例上）的最佳网络。这些想法的局限性仍在探索中。

4.4.1 Hessian-vector product in linear time: Werbos-Pearlmutter trick

它可以将反向传播推广到使用二阶导数，特别是与Hessian H 相关，这是一个对称矩阵，其 (i, j) 项是 $\partial^2 f / \partial w_i \partial w_j$。有时 H 也表示为 $\nabla^2 f$。仅写下这个矩阵就需要二次时间和内存，这对于今天的深度网络来说是不切实际的。令人惊讶的是，使用反向传播可以在线性时间内计算任何向量 x 的矩阵-向量乘积 Hx。这个技巧由Pearlmutter描述，他将它归功于Werbos的早期工作 [1]。

主张4.4.1. *Suppose an acyclic network with V nodes and E edges has output f and leaves z_1, ..., z_m. Then there exists a network of size $O(V +$*

1 P. J. Werbos. 反向传播：过去与未来。在 *IEEE InternationalConfer-ence on Neural Networks*，第 343–35 3 页，1988；以及 Barak Pearlmutter. 通过Hessian的快速精确乘法。*Neural Computation*，199 4

E) that has $z_1, \ldots, z_m$ *as input nodes and* $\frac{\partial f}{\partial z_1}, \ldots, \frac{\partial f}{\partial z_m}$ *as output nodes.*

证明该命题的方法直接从将消息传递协议实现为一个无环电路中得出。

接下来，我们展示如何计算$\nabla^2 f(z) \cdot v$，其中v是一个给定的固定向量。令$g(z) = \langle \nabla f(z), v \rangle$为$\mathbb{R}^d \to \mathbb{R}$的函数。然后根据上述断言，$g(z)$可以通过大小为$O(V + E)$的网络来计算。现在再次将断言应用于$g(z)$，我们得到$\nabla g(z)$也可以通过大小为$O(V + E)$的网络来计算。

请注意，通过构造，

$$\nabla g(z) = \nabla^2 f(z) \cdot v.$$

因此，我们在网络大小时间内计算了Hessian向量积。

5
Basics of generalization theory

从第1章回忆起第1章的实证风险最小化语言。一个用于分类的数据点 x (实际上是一个向量和标签) 的配对，它们来自分布 $\mathcal{D}$，而 S 表示训练样本。假设 h 在数据点 x 上的损失是 $\ell(x,\ h)$。(由于深度学习中的假设由参数向量 w 给出，我们也可以表示为 $\ell(x,\ w)$。) 在泛化理论中，我们感兴趣的是理解测试损失和训练损失（分别）之间的关系：

$$L_{\mathcal{D}}(h) = \mathop{\mathbb{E}}_{x\in\mathcal{D}}[\ell(x,h)] \quad \text{and} \quad \widehat{L}_S(h) = \mathop{\mathbb{E}}_{x\in S}[\ell(x,h)]\,. \tag{5.1}$$

(这里 $\widehat{\ }$ 指的是“经验”。如果 $L_S(h)$ 很小，并且 *generalization error* $\Delta_S(h) = L_{\mathcal{D}}(h) - \widehat{L}_S(h)$ 也很小，则认为训练成功。

泛化理论给出了保证低泛化误差所需训练样本数量的估计。本章描述的经典思想为深度学习提供了非常宽松（即微不足道)）的估计。我们概述了提供更紧估计的尝试。

泛化理论从一种古老的哲学原则中汲取灵感，称为*Occam's razor*：在简单科学理论和更复杂的理论之间做出选择，两者都能解释一些经验观察，我们应该相信更简单的那个。例如，哥白尼的日心说在科学中受到青睐，因为它比古代亚里士多德的理论更简单地解释了已知事实。虽然这从直觉上讲是有道理的，但奥卡姆剃刀法则有点模糊且含糊不清。什么使一个理论“更简单”或“更好”？

5.1 Occam's razor formalized for ML

以下是将上述直观概念映射到机器学习概念的过程。（为了简化，我们在此仅关注监督学习，并在后续章节中考虑其他设置。）

Observations/evidence $\leftrightarrow$ 训练数据集 S 假设 h
All possible theories $\leftrightarrow$ 假设类别 $\mathcal{H}$ *Finding theory to fit observations* $\leftrightarrow$ 最小化训练损失
以找到 $h \in \mathcal{H}$ *Theory is good (good predictions in new settings)* $\leftrightarrow$ h 具有低测试损失
Simpler theory $\leftrightarrow$ h 描述更短

“简短描述”的概念将通过多种方式形式化，使用一个 *complexity measure* 对类 $\mathcal{H}$ 进行表示，记为 $C(\mathcal{H})$，并利用它来上界泛化误差。

让 S 成为 m 数据点的样本。经验风险最小化（ERM）范式（见第1章）涉及找到 $\widehat{h} = \arg\min \widehat{L_S}(h)$。当然，在深度学习中，我们可能找不到绝对最优 h，但在实践中，训练损失变得非常小，接近零。直观上，如果泛化误差很大，那么假设在训练样本 S 上的性能并不能准确反映在所有示例的全分布上的性能，我们称之为 *overfitted* 到样本 S。

一般泛化误差的上界[1]表明
至少以概率$1-\delta$ 在训练数据的选择上，以下

$$\Delta_S(h) \leq \sqrt{\frac{C(\mathcal{H}) + O(\log(1/\delta))}{m}} \tag{5.2}$$

因此，为了降低泛化误差，只需使 m 显著大于“复杂度度量”。因此，具有较低复杂度的类别需要更少的训练样本，这与奥卡姆的直觉一致。

1 这是一般化界限的典型格式！一般来说，本章侧重于基本思想的清晰阐述，而对常数则有些马虎。

5.1.1 Motivation for generalization theory

泛化界限试图使用训练模型 h 和训练数据集 S 来估计泛化误差。学生可能会想知道，如果实验者已经决定了架构、训练算法等，这个界限是否有任何用处。实际上，那么实验可以继续进行训练，并使用保留的数据集来估计泛化误差。

希望发展泛化理论能够提供如何设计架构和算法的见解，从而使训练网络具有“低复杂性”，从而实现良好的泛化。显然，沿着这样的原则性理解会更加理想。

5.1.2 *Warmup: Classical polynomial interpolation*

假设我们给出了 n 个点 $(x^{(1)}, y^{(1)})$，...，$(x^{(n)}, y^{(n)})$，这些点是根据以下概率过程选择的：$x^{(i)}$ 从 $[0, 1]$ 上的均匀分布中选择，$y^{(i)} = p(x^{(i)}) + \eta$ 其中 $p()$ 是一个未知的 d 次多项式，η 是从噪声分布 $\mathcal{N}(0, \sigma^2)$ 中抽取的样本。由于 $\mathbb{E}[\eta] = 0$ 和 $\mathbb{E}[\eta^2] = \sigma^2$，找到 p 的明显方法是最小化最小二乘拟合以找到多项式的系数 θ_0，θ_1，...，θ_d：

$$\ell(\vec{\theta}) = \sum_{i=1}^{n} (y^{(i)} - \sum_{j=0} \theta_j (x^{(i)}))^2.$$

这是隐式地使用新的数据表示进行线性回归，其中点 $x \in \Re$ 使用向量（1，x，x^2，...，x^d）表示。

但是如果我们不知道 d 并尝试拟合一个在 $N \gg d$ 的 N 次多项式，在什么条件下最小化上述损失会给我们一个与 $p()$ 大致相同的多项式？一个实际的想法——注意到上述损失对于真实的多项式 $p()$ 也是 $n\sigma^2$——是在上述损失中添加一些较大的 $\lambda > 0$ 正则化器 $\lambda \|\theta\|_2^2$。这向梯度下降发出信号，即减少最小二乘损失到零并不重要，而应该找到低范数的解 θ。更普遍地，可以使用其他比欧几里得范数平方的“复杂性”度量。

这个例子直观且可以更严格地分析，但需要关于自然分布上[0, 1]的正交多项式理论。

5.2 *Some simple upper bounds on generalization error*

回忆初等概率中的 *union bound*：每个事件集 A_1，A_2，... 满足 $\Pr[\cup_i A_i] \leq \sum_i \Pr[A_i]$。许多关于泛化的分析都利用了这个简单事实。

第一个例子几乎在一种平凡的设置中说明了这一点。稍后我们将看到同样的想法也存在于其他泛化界限[2]的核心，尽管这通常隐藏在证明中。这个界限表明

如果假设类最多包含 N 个不同的假设，那么 $\log N$ (，即表示这个类中假设索引所需的位数，) 是 (5.2) 中的有效复杂度度量。

2 联合界也被称为许多书中的均匀收敛框架。通常假设类是无限的，但证明将其离散化，如定理5.2.7中所示。

定理5.2.1（有限假设类）。*If the loss function takes values in* $[0, 1]$ *and hypothesis class* $\mathcal{H}$ *contains* N *distinct hypotheses then*

with probability at least $1-\delta$*, every* $h \in \mathcal{H}$ *satisfies*

$$\Delta_S(h) \leq 2\sqrt{(\log N + \log(1/\delta))/m}.$$

Proof. 对于任何 *fixed* 假设 g 想象绘制一个大小为 m 的训练样本。然后 $\widehat{L_S}(g)$ 是独立同分布变量的平均值，其期望为 $L_{\mathcal{D}}(g)$。集中不等式意味着 $L_{\mathcal{D}}(g) - \widehat{L_S}(g)$ 至少具有与一元高斯 $\mathcal{N}(0, 1/m)$ 相当强的集中性质。前述陈述对于类中的所有假设 g 都成立，因此并集不等式意味着这个量超过 ϵ 的概率至多为 $N\exp(-\epsilon^2 m/4)$，对于类中的 *some* 假设。由于 h 是 ERM 的解，我们得出结论，当 $\delta \leq N\exp(-\epsilon^2 m/4)$ 时，则 $\Delta_S(h) \leq \epsilon$。简化并消除 ϵ，我们得到定理。

□

初看之下，联合界似乎对深度网络无用，因为如果网络有 k 个实值参数，即使我们固定了架构和参数数量，假设集——也由 $\mathbb{R}^k$ 中的所有向量组成，一个不可数集！

示例5.2.2（可能的解决方案？）。*As we saw in Chapter 2, the end result of gradient descent on the loss is special. For instance it must be a stationary point (i.e., where* $\nabla = 0$*) of the training loss. One can similarly imagine other conditions on Hessian* $\nabla^2()$ *and so forth. One could hope that the set of points in the loss landscape with such properties could be small and finite. This line of investigation hasn't yet worked out because current nets are so overparametrized (i.e., number of parameters far exceeding the number of training data points) that the set of such solution points in the landscape is also in general a continuous set (i.e., uncountable). The next hope is to take into account the training algorithm, because not all solution points may be accessible via standard training algorithms. These are some ideas for restricting attention to a finite set of solutions, though they haven't yet worked out.*

存在一种更明显的方法将可能的深度网络集合转化为有限集合：*discretization!* 假设我们假设参数向量的 ℓ_2 范数不超过 1，这意味着所有深度网络集合已经与单位球 Ball(0, 1) 相对应。（此处 Ball(w, r) 指的是 $\mathbb{R}^k$ 中距离 r 内的所有点的集合。）

假设损失在参数上是Lipschitz连续的：对于每个数据点 {v*} 和参数向量 w_1、w_2 $\|\ell(x、w_1) - \ell(x、w_2)\| \leq、 C\|w_1 - w_2\|_2$，对于某个显式常数 C。下面的论证只需要局部Lipschitz连续性，即对于某个显式常数 ρ，条件 $\|w_1 - w_2\|_2 \leq \rho$ 成立。此外，它只需要训练集上平均损失的Lipschitz连续性，而不是单个数据点的损失。

假设 $\rho > 0$ 是这样的，如果 w_1，$w_2 \in \mathbb{R}^k$ 满足 $\|w_1 - w_2\|_2 \leq \rho$，那么这两个参数向量的网络在每一个输入上具有本质上相同的损失，意味着损失最多相差 γ 对于某些 $\gamma > 0$。(注意：这相当于一个 *local* 拉普拉斯常数，并且它最多是 γ/ρ。) 对于每个 $\gamma > 0$，这样的 ρ 必须存在是有直觉意义的，因为当我们让 $\rho \to 0$ 时，两个网络变得相等[3]。

3 我们忽略的问题是 ρ、γ 可能依赖于训练集的大小。不幸的是，这似乎是现实生活中的深度学习情况，而这篇分析却忽略了这一点。

问题 5.2.3. *Compute Lipschitz constant of the ℓ_2 regression loss: the loss on datapoint (x, y) is $(w \cdot x - y)^2$.*

问题 5.2.4. *Compute Lipschitz constant of ℓ_2 loss for a two layer deep net with ReLU gates (zero bias) on the middle layer. Assume the two parameter vectors are infinitesimally close.*

定义5.2.5 (ρ-覆盖). *A set of points $w_1, w_2, \ldots \in \mathbb{R}^k$ is a* ρ-覆盖 *if for every $w \in$ 球$(0, 1)$ there is some w_i such that $w \in$ 球(w_i, ρ).*

引理5.2.6（ρ-覆盖的存在性）。*There exists a ρ-cover of size at most $((2 + \rho)/2\rho)^k$.*

Proof. 证明简单但巧妙。让我们在单位球（0，1）中任意选择 w_1。对于 $i = 2, 3, \cdots$，执行以下操作：在单位球（0，1）中任意选择不在 $\cup_{j \leq i}$Ball(w_j, ρ) 内的任意一点，并将其指定为 w_{i+1}。

A priori 不清楚这个过程是否永远终止。我们现在证明它最多在$(2/\rho)^k$步之后终止。为了证明这一点，只需注意对于所有$i < j$，有Ball$(w_i, \rho/2) \cap$ Ball$(w_j, \rho/2) = \emptyset$。因为如果不是这样，那么$w_j \in$ Ball(w_i, ρ)，这意味着在上面的过程中w_j不可能被选中。因此，我们得出结论，这个过程最多在以下步骤后停止：

$$\text{volume}(\text{Ball}(0, 1 + \rho/2))/\text{volume}(\text{Ball}(0, \rho/2))$$

迭代[4]，最多为$((2 + \rho)/2\rho)^k$，因为球体积在 $\mathbb{R}^k$ 与半径的k次方成正比。

4 分子中$1 + \rho/2$ 的原因是，如果一个 w_i 位于球（0，1）的表面上，那么围绕它的半径为 $\rho/2$ 的球体位于围绕原点的半径为 $1 + \rho/2$ 的球体内部

最后，末尾的 w_i 序列必须是一个 ρ-覆盖，因为只有在找不到 $\cup_j$Ball(w_j, ρ) 外的点时，过程才会停止。

□

定理5.2.7（赋范空间的泛化界）。*If (i) hypotheses are unit vectors in $\mathbb{R}^k$ and (ii) every two hypotheses h_1, h_2 with $\|h_1 - h_2\|_2 \leq \rho$ differ in terms of loss on every datapoint by at most γ then* [5]

$$\Delta_S(h) \leq \gamma + 2\sqrt{k \log(2/\rho)/m}.$$

5 即使忽略对Lipschitz常数的依赖，这个界限也要求数据点的数量与网络中可训练参数的数量线性增长。这甚至不能开始解释现实生活中的深度学习令人惊讶的有效性，其中参数的数量远远超过训练数据点的数量。

Proof. 应用 ρ-覆盖的并集界。其他任何网络的最大损失不能超过 γ，比 ρ-覆盖中的网络高。 □

5.3 *Data dependent complexity measures*

到目前为止，我们考虑了假设类复杂度度量作为量化其“复杂度”的方法：假设类的大小（假设它是有限的）以及其中 {v*} 覆盖的大小。当然，由此得到的样本复杂度界限仍然很宽松。

但是这些简单的界限适用于每个数据分布 $\mathcal{D}$。在实践中，似乎很明显，深度网络——或者任何学习方法——通过能够利用输入分布的性质（例如，卷积结构利用了所有图像子块可以非常相似地处理的事实）来工作。因此，应该尝试证明一些依赖于数据分布的复杂度度量。

5.3.1 *Rademacher Complexity*

拉代马赫复杂度是一种依赖于数据分布的复杂度度量。如通常所述，我们的描述假设损失函数的值在[0, 1]区间内。

定义涉及以下思想实验。回想一下，分布 $\mathcal{D}$ 在标记数据点上 (x, y)。为了简单起见，我们用 z 表示标记数据点。

现在 *Rademacher Complexity* [6] 个假设类 $\mathcal{H}$ 在一个分布-$\mathcal{D}$ 定义如下，其中 $l(z, h)$ 是在标记数据点 z 上对假设 h 的损失。

$$\mathcal{R}_{m,D}(\mathcal{H}) = \mathop{\mathbb{E}}_{S_1,S_2}\left[\frac{1}{2m}\sup_{h\in\mathcal{H}}\left|\sum_{z\in S_1} l(z,h) - \sum_{z\in S_2} l(z,h)\right|\right], \qquad (5.3)$$

在期望是关于 S_1 的情况下，S_2 是两个独立同分布的样本集（即多集），每个大小为 m，来自数据分布 $\mathcal{D}$。请注意，此定义涉及选择 S_1、S_2 并选择一个在这些上训练误差尽可能不同的分类器的思维实验。以下定理将此与训练假设的泛化误差联系起来。

6 标准账户通常会让学生感到困惑，或者至少会以一个复杂的定理5.3.1的证明来错误地给他们留下印象，这个证明隐藏了下面的简单想法。我们的定义简化了一些：在标准定义中，为每个$2m$数据点选择一个符号± 1（或*Rademacher*随机变量），并观察由这个符号加权的损失。我们的定义得到的价值在标准定义中的值附近$\pm O(1/\sqrt{m})$。

定理5.3.1. *If h is the hypothesis trained via ERM using a training set S_2 of size m, then the probability (over S_2) is $> 1-\delta$, that*

$$\Delta_{S_2}(h) \le 2\mathcal{R}_{m,D}(\mathcal{H}) + O((\log(1/\delta))/\sqrt{m}).$$

Proof. 泛化误差 $\Delta_{S_2}(h) = L_{\mathcal{D}}(h) - \widehat{L_{S_2}}(h)$，ERM 保证了一个 h，该 h 最大化了这一点。想象我们从这个分布 $\mathcal{D}$ 中选择另一个独立同分布的 m 样本，以获得另一个（多重）集合 S_1。然后，至少以概率 $1-\delta$，这些样本上的损失接近 $L_{\mathcal{D}}(h)$：

$$\Delta_{S_2}(h) \le \widehat{L_{S_1}}(h) - \widehat{L_{S_2}}(h) + O((\log(1/\delta))/\sqrt{m}).$$

现在我们注意到所绘制的 S_1、S_2 与 (5.3) 7 (5.3) 中的思想实验中绘制的集合以及此处的最大化值 h 完全相同

表达式定义为 $\mathcal{R}_{m,D}$。因此，右侧至多为

$$2\mathcal{R}_{m,D}(\mathcal{H}) + O((\log(1/\delta))/\sqrt{m}).$$

□

7 在这里，假设 h 可以依赖于 S_2 但不依赖于 S_1。在思想实验中，上确界是在 h 上取的，它可以同时依赖于这两个变量。这只会帮助不等式，因为后者 h 可以达到更大的值。注意，因子 2 是由于 (5.3) 中 $2m$ 的缩放引起的。

问题 5.3.2. *Show that the Rademacher complexity of the set of linear classifiers (unit norm vectors* $U = \{w | w \in \mathbb{R}^d, \ \|w\|_2 = 1\}$*), on a given sample* $S = \{x_1, \ x_2, \ \cdots, \ x_m\}$ *(each* $x_i \in \mathbb{R}^d$*) is* $\leq \max_{i\in[m]} \|x_i\|_2/\sqrt{m}$.

问题 5.3.3. *Consider the kernel classifier of the form* $h(x) = z^\top G^{-1} y$ *we studied in Section 3.2 where* G *is the* $n \times n$ *kernel matrix,* y *is the labels and* z *is the column vector whose* i*-th coordinate is* $K(x, x_i)$*. Show that the Rademacher complexity upper is* $\sqrt{2y^\top Gy \cdot \mathrm{Tr}(G)}/n$*. (We will use this result in Chapter 9 to prove certain over-parameterized student nets can learn simple two-layer teacher nets.)*

5.3.2 *Alternative Interpretation: Ability to correlate with random labels*

教师们更直观地解释Rademacher复杂性为 *ability of classifiers in* $\mathcal{H}$ *to correlate with random labelings of the data.* 这在二元分类（即标签是0/1）中理解最好，损失函数也是二元的（正确标签的损失为0，错误标签的损失为1）。现在考虑以下实验：选择 S_1，S_2 如Rademacher复杂性的定义中所述，并想象翻转 S_1 的标签。现在 S_2 上的平均损失为$1 - \widehat{L_{S_2}}(h)$。因此，选择 h 以最大化（5.3）右侧就像寻找一个 h，它在标签被翻转的 $S_1 \cup S_2$ 上的损失较低。换句话说，h 能够在标签被翻转的某些随机选择的训练点集合上的数据集上实现低损失。

当损失不是二进制时，类似的陈述在定性上仍然成立。

5.4 *Understanding limitations of the union-bound approach*

该联合界方法及相关方法所捕捉的现象也被称为 *uniform convergence*。如果我们已经识别了一个有限假设集 $\mathcal{H}$ 和数据点样本 S 足够大，那么在至少 $1 - \delta$ 的概率下，关于 S 的选择是

$$\|L_{\mathcal{D}}(h) - \widehat{L}_S(h)\| \leq \epsilon \quad \forall h \in \mathcal{H}. \tag{5.4}$$

这里需要注意的是，一个固定的样本集 S 可以用于对 *every* 分类器 h 的一般化误差进行良好估计。

在8.当然，使用γ-覆盖这种结论可以是
显示对于连续集合的类 $\mathcal{H}$ 也适用，例如具有有界 ℓ_2 范数的假设。现在我们描述一个来自 Nagarajan 和 Kolter 9 的简单且有趣的例子，该例子指出了为什么这个框架可能
在现代设置中难以应用，尤其是在深度学习中。关键在于感兴趣的假设类是通过优化算法（例如，梯度下降）隐式定义的，而这个类可能不允许通过并集界进行清晰的分析。

请注意，这些分类器中的大多数可能在 S 上有巨大的损失；并集界只保证泛化误差很小。

9 V Nagarajan 和 Zico Kolter。统一收敛可能无法解释深度学习中的泛化。*NeurIPS*，2019

5.4.1 *An illustrative example that mixes optimization and generalization*

假设点在 $\Re^{D+K}$ 中，标签在 ± 1。存在一个固定向量 $u \in \Re^k$，使得带标签的数据点 (x, y) 来自以下分布 $\mathcal{D}$：首先在 $\{\pm 1\}$ 中均匀选择第一个标签 y，然后将 x 的第一个 K 坐标——我们为了方便将其表示为 x_1 ——设置为向量 $y \cdot u$。其余的 D 坐标，表示为 x_2，由一个随机向量 $\Re^D$ 组成，其每个坐标独立地从 $\mathcal{N}(0, 1/D)$ 中抽取。集中度量的含义是 x_2 在本质上类似于在 $\Re^D$ 中的随机单位向量。

分类可以清楚地使用线性分类器 $x \to \text{sgn}(w^* \cdot x)$ 解决，其中第一个 K 坐标包含 $w_1^* = u/\|u\|_2^2$，最后一个 D 坐标包含 $w_2^* = 0$。

让我们考虑一个简单的训练目标：找到最大化 $y \cdot h(x)$ 的线性分类器 $h(x)$。粗略地说，这忽略了 $h(x)$ 的大小，并试图使 $h(x)$ 和 y 的符号一致。使用学习率 $\eta = 1$ 和 m 数据点的样本 S (x^i, y^i) 为 $i = 1, ..., m$ 梯度下降产生具有 $w_S = (w_1, w_2)$ 的分类器，其中

$$w_{S,1} = m \cdot u, \qquad w_{S,2} = \sum_i y^i x_2^i. \tag{5.5}$$

注意，$w_{S,2}$ 是 m 个随机单位向量的和，这意味着它的范数在 $\sqrt{m}$ 附近相当集中。换句话说，与我们的理想分类器 w^* 不同，学习到的分类器在最后的 D 坐标中有很多与分类无关的冗余信息。

现在我们描述如何设置各种参数。通常，m表示训练集大小。我们设置

$$m\sqrt{(\log 1/\delta)} \approx D \tag{5.6}$$

$$\|u\|_2^2 = \frac{1}{m} \tag{5.7}$$

学习到的分类器 w_S 的 ℓ_2 范数约为 $\sqrt{m^2\|u\|_2^2 + m}$，因此 (5.7) 表明这个范数是 $\sqrt{2m}$。

让我们检查垃圾坐标不会干扰随机选择的数据点m的分类——换句话说，具有良好的测试误差。给定一个新数据点$x = (y \cdot u, x_2)$，其中x_2是一个随机单位向量，学习到的分类器产生答案$w_S \cdot x = my\|u\|_2^2 + x_2 \cdot (\sum_i y^i x_2^i)$。由于固定向量与随机高斯向量$\mathcal{N}(0, 1/D)$的内积是一个标准差为$1/\sqrt{D}$倍固定向量范数的一元高斯分布，我们看到这个符号是正确的，即y，只要概率为$1 - \delta$

$$m\|u\|_2^2 > \sqrt{\frac{m \log 1/\delta}{D}},$$

该公式从（5.6）和（5.7）成立。因此，学习到的分类器在随机测试数据点上工作良好。

但是，现在想象一下我们试图通过一个上界论证来解释学习的成功。让我们用 $\mathcal{H}_0$ 表示从 m 数据点的训练集中通过GD可能产生的此类分类器的集合。论证必须证明，对于所有分类器 $h \in \mathcal{H}_0$，$\Delta_S(h)$ 以高概率很小。下一个结果表明这并不正确。

5.4.1. *For a random sample set S, whp there is a classifier w_{flip} whose generalization error is large (specifically, whose loss on full distribution $\mathcal{D}$ is small but whose loss on S is large.)*

Proof. 我们令 w_{flip} 为在集合 S_{flip} 上训练的分类器，该集合通过取 S 并翻转 x_2 部分的符号得到。换句话说，S 中的数据点 $z = (y^i u、x_2^i)$、y^i 转变为 S_{flip} 中的 $z_{\text{flip}} = (y^i u、-x_2^i)、y^i$。请注意，$S_{\text{flip}}$ 与 S 具有相同的概率。根据我们之前的分析，w_S 和 w_{flip} 在前 K 个坐标上达成一致，但最后 D 个坐标的符号相反。因此，$(w_S - w_{\text{flip}}) \cdot z$ 的绝对值至少为 $2x_2^i \cdot x_2^i = 2$。因此，我们已经证明了 $w_S \cdot z$ 和 $w_{\text{flip}} \cdot z$ 的符号不同。

□

让我们考虑我们已经展示的内容。分类器 w_S 和 $w_{\mathbf{flip}}$ 都有优秀的测试误差。然而，在用于生成 w_S 的训练集 S 上，分类器 $w_{\mathbf{flip}}$ 有着糟糕的泛化误差。这表明在训练数据集 S 上使用朴素并集界来证明 w_S 良好的泛化存在一个障碍。

注意，上述限制不成立，如果我们被允许修改/剪枝训练结束时获得的分类器。可以想象通过某种简单测试识别出学习到的分类器中的非影响力坐标，并意识到最后 D 坐标可以置零而不会严重影响准确性。然后所有学习到的分类器都成为理想分类器 w^* 的标量倍数。换句话说，这里所示的限制不适用于我们在下一节中描述的方法。

5.5 A Compression-based framework

现在我们描述了一种基于压缩的简单技术10来自Arora等人11将一个非常简单的想法形式化。假设火车-数据集 S 包含 m 个样本，h 是一个来自复杂类别（例如，具有超过 m 个参数的深度网络）的分类器，它产生了非常低的经验损失。我们试图通过观察 h 和 S 来了解 h 的一般化程度。现在假设我们可以计算一个具有离散可训练参数且远少于 m 的分类器 g，它在训练数据上产生的损失与 h 相似。我们称这种为 *approximator* 的 h。然后如果 g 的描述长度足够低，其泛化可以通过简单的并集界限论证来得出。12

compression

10 请不要将此与基于 *data* 的一般化理论中的另一种较老且无关的技术混淆，该技术不适用于深度学习。

11 Sanjeev Arora, Rong Ge, Behnam Neyshabur 和 Yi Zhang. 通过压缩方法获得深度网络的更强泛化界限。在 *Proc. ICML*, 第 254–263 页，2018

2018

此框架具有保持直观参数计数和避免显式处理包含 h (（见定理5.5.3之后的注释）) 的假设类的好处。请注意，从 f 到 g 的映射只需 *exist,* 不需要高效可计算。但在我们所有的例子中，映射将是明确的并且相当高效的。现在我们正式化这些概念。证明通过集中界限是基本的，并出现在附录中。

12 这种场景与网络剪枝的实证工作非常相似，其中训练好的深度网络通过长长一串方法之一进行压缩，剪掉大量参数并重新训练剩余部分。如果剪枝后的网络足够紧凑，就可以证明剪枝网络的泛化界限。参见

定义5.5.1 ((γ,S)-可压缩). *Let f be a classifier and $G_{\mathcal{A}} = \{g_A | A \in \mathcal{A}\}$ be a class of classifiers. We say f is (γ, S)-compressible via $G_{\mathcal{A}}$ if there exists $A \in \mathcal{A}$ such that for any $x \in S$, we have for all y*

$$|f(x)[y] - g_A(x)[y]| \leq \gamma.$$

我们还考虑了一种不同的设置，其中压缩算法允许一个“辅助字符串”s，它是任意的但必须在查看训练样本之前固定。通常s将包含随机数。13

13 一个简单的例子是让 s 成为用于训练深度网络的随机初始化。然后可以在最终权重和 *difference* 之间进行压缩；这可以给出更好的泛化界限。

定义5.5.2 ((γ,S)-可压缩，使用辅助字符串 s）。*Suppose $G_{\mathcal{A},s} = \{g_{A,s} | A \in \mathcal{A}\}$ is a class of classifiers indexed by trainable parameters A and fixed strings s. A classifier f is (γ, S)-compressible with respect to $G_{\mathcal{A},s}$ using helper string s if there exists $A \in \mathcal{A}$ such that for any $x \in S$, we have for all y*

$$|f(x)[y] - g_{A,s}(x)[y]| \leq \gamma.$$

以下定理是上述并集界方法的一个简单应用。

定理5.5.3. *Suppose $G_{\mathcal{A},s} = \{g_{A,s} | A \in \mathcal{A}\}$ where A is a set of q parameters each of which can have at most r discrete values and s is a helper string. Let S be a training set with m samples. If the trained classifier f is (γ, S)-compressible via $G_{\mathcal{A},s}$ with helper string s, then there exists $A \in \mathcal{A}$*

with high probability over the training set,

$$L(g_A) \le \widehat{L}_\gamma(f) + O\left(\sqrt{\frac{q \log r}{m}}\right),$$

where $L(f) = \mathbb{E}_{(x,\ y)\in\mathcal{D}}[f(x)[y] \le \max_{j\neq y} f(x)[j]]$ *is the expected error and* $\widehat{L}_\gamma(f)$ *is the proportion of data* (x, y) *satisfying* $f(x)[y] \le \max_{j\neq y} f(x)[j]$ *in the training set* S.

Remarks: (1) 框架证明了泛化不是f，而是其压缩g_A。（一个例外是，如果两个在域的每个点上显示出相似的损失，而不仅仅是训练集。这是定理5.5.6的情况。）

(2) 前一项突出了与我们之前所说的联合界（定理5.2.1）的区别。在那里，需要固定训练集的一个假设类 *independent*。相比之下，我们没有假设类，只有一个具有某些特定属性的 *single* 神经网络，该属性在 *single* 有限训练集上。但如果我们可以将这个特定神经网络压缩到一个参数更少的简单神经网络，那么我们可以使用覆盖数论据来获得压缩网络的泛化。

(3) 问题（1）也存在于研究人员经常如何应用标准的PAC-Bayes框架于深度网络（第5.6节）。

5.5.1 Example 1: Linear classifiers with margin

为了说明上述压缩方法，我们使用具有高边界的线性分类器。考虑一个简单的线性分类器族，由单位向量 $c \in \mathbb{R}^d$ 组成，其 ± 1 输出在输入 x 上由 $\text{sgn}(c \cdot x)$ 给出（即与数据点的内积的符号）。假设所有数据点也都是单位向量。假设 c 有 *margin* γ，如果对于所有训练对 (x, y)，我们有 $y(c^\top x) \ge \gamma$。

我们展示了如何将具有边距 γ 的此类分类器压缩为只有 $O(1/\gamma^2)$ 个非零条目的分类器。首先，假设所有 c_i 的绝对值小于 $\gamma^2/8$。

对于每个坐标 i，掷一个概率为 $\Pr[heads] = p_i = 8c_i^2/\gamma^2$ 的硬币，如果出现正面，则将坐标设置为等于 $c_i/p_i = \gamma^2/8c_i$。这产生一个向量 $\widehat{c}$。$\widehat{c}$ 中非零条目的期望数量是 $\sum_{i=1}^d p_i = 8/\gamma^2$。根据切诺夫不等式，我们知道非零条目的数量以高概率不超过 $O(1/\gamma^2)$。

此外，i的方差为$2p_i(1-p_i)\frac{c_i^2}{p_i^2} \le \frac{2c_i^2}{p_i} \le \gamma^2/4$。因此，对于任何与$\widehat{c}$选择无关的单位向量$u$，我们有$\mathbb{E}[\widehat{c}^\top u] = c^\top u$。现在我们估计随机变量$\widehat{c}^\top u$的方差。它是$\le \gamma^2/4 \cdot \|u\|^2 \le \gamma^2/4$。根据切比雪夫不等式，我们知道$\Pr[|\widehat{c}^\top u - c^\top u| \ge \gamma] \le 1/4$，因此$\widehat{c}$和$c$将使

对于所有满足 $|c^\top u| \geq \gamma$ 的 u 都有相同的预测。然后我们可以对 $\widehat{c}$ (的离散化版本通过平凡的舍入) 应用定理 5.5.3，以表明稀疏分类器具有 $O(\log d/\gamma^2)$ 个样本的良好泛化能力。

问题 5.5.4. *Redo the proof above when some coordinates have absolute value more than $\gamma^2/8$.*

此压缩分类器对于固定输入 x 以恒定概率正确工作，但不是高概率。为了解决这个问题，可以求助于“固定字符串压缩”模型。固定字符串是一个随机线性变换。当应用于单位向量 c 时，它倾向于使所有坐标相等，并且保证 $|\widehat{c}^\top u - c^\top u| < \gamma$ 可以以高概率成立。这个随机线性变换可以在看到训练数据之前固定。

问题 5.5.5. *Prove the above property of random linear transformations. That is, let M be a random matrix of size $O(1/\gamma^2) \times d$, drawn from a suitable distribution you choose before seeing the unit vector c and the training data. Then, show that the following holds for fixed unit vectors c and u with high probability*

$$\|Mc\|_\infty = O(1), \qquad |\langle Mc, Mu\rangle - \langle c, u\rangle| < \gamma.$$

This means we can compress a unit vector c to $\widehat{c} = M^\top Mc$. Finally, Apply Theorem 5.5.3 on a discretized version of $\widehat{c}$ to show a good generalization bound with $\widetilde{O}(1/\gamma^2)$ samples, where $\widetilde{O}$ can hide polylog factors of d and $1/\gamma$.

5.5.2 *Example 2: Generalization bounds for deep nets using low rank approximations*

一些早期关于全连接网络的泛化界限使用了层矩阵通常是低秩的事实。（或者可能是最终矩阵减去初始化。）我们给出了这样一个结果的一个简单证明。

实现一个秩为 r 的 $h \times h$ 矩阵实际上有 $2hr$ 个参数，尽管有 h^2 个条目。我们回忆，对于一个方阵 A，谱范数（即最大的奇异值）表示为 $\|A\|_2$，奇异值的平方和表示为 $\|A\|_F^2$，其中 $\|\cdot\|_F$ 也被称为 *Frobenius norm*。比值 $\|A\|_F^2/\|A\|_2^2$ 被称为 *stable rank*，并且它显然被秩所上界。通常，训练好的网络的层具有低稳定的秩，尽管本身的秩很高。

定理5.5.6. *([14]) For a depth-d ReLU net with hidden layers of equal width h and single coordinate output, let $A^1, A^2, \ldots A^d$ be weight matrices*

[14] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *ICLR*, 2018

and γ be the output margin on a training set S of size m. Then the generalization error can be bounded by

$$\widetilde{O}\left(\sqrt{\frac{hd^2 \max_{x\in S}\|x\| \prod_{i=1}^d \|A^i\|_2^2 \sum_{i=1}^d \frac{\|A^i\|_F^2}{\|A^i\|_2^2}}{\gamma^2 m}}\right).$$

第二部分（$\sum_{i=1}^d \frac{\|A^i\|_F^2}{\|A^i\|_2^2}$）是层的稳定秩之和，这是它们真实参数数量的自然度量。第一部分（$\prod_{i=1}^d \|A^i\|_2^2$）与网络的Lipschitz常数相关，即如果输入是单位向量，它能够产生的向量的最大范数。矩阵算子B的Lipschitz常数就是它的谱范数$\|B\|_2$。由于网络应用了一系列矩阵操作，其中穿插了ReLU，而ReLU是1-Lipschitz，我们得出结论，整个网络的Lipschitz常数至多为$\prod_{i=1}^d \|A^i\|_2$。

为了证明定理5.5.6，我们使用以下引理将每层的矩阵压缩为秩更小的矩阵。由于秩为r的矩阵可以表示为两个内维为r的矩阵的乘积，它有$2hr$个参数（而不是平凡的h^2）。此外，参数可以通过平凡的舍入进行离散化，以获得符合定义5.5.1所需的离散参数的压缩。

引理5.5.7. *For any matrix $A \in \mathbb{R}^{m\times n}$, let $\widehat{A}$ be the truncated version of A where singular values that are smaller than $\delta\|A\|_2$ are removed. Then $\|\widehat{A} - A\|_2 \le \delta\|A\|_2$ and $\widehat{A}$ has rank at most $\|A\|_F^2/(\delta^2\|A\|_2^2)$.*

Proof. 设 r 为$\widehat{A}$ 的秩。根据构造，$\widehat{A} - A$ 的最大奇异值至多为 $\delta\|A\|_2$。由于剩余的奇异值至少为 $\delta\|A\|_2$，因此我们有 $\|A\|_F \ge \|\widehat{A}\|_F \ge \sqrt{r}\delta\|A\|_2$。 □

对于每个 i，使用上述引理将其层 i 替换为其压缩，其中 $\delta = \gamma(3d\|x\|\prod_{i=1}^d \|A^i\|_2)^{-1}$。这会在每一层引入多少误差，以及它会对通过中间层（并被它们的Lipschitz常数放大）后的输出产生多大影响？由于 $A - \widehat{A}^i$ 的谱范数（即Lipschitz常数）最多为 $\delta\|A^i\|_2$，因此单独改变层 i 在输出处引入的误差最多为 $\prod_{j=i+1}^d \|A^j\|_2 \cdot \delta\|A^i\|_2 \cdot \prod_{j=1}^{i-1} \|A^j\|_2 \cdot \|x\| \le \gamma/3d$。其余的证明留给读者，泛化界限立即从定理5.5.3得出。

问题 5.5.8. *Complete the above proof using a simple induction (see [15] if needed) to show the total error incurred in all layers is strictly bounded by γ. That is, for an input x, the change in the deep net output is smaller than γ after replacing every weight matrix A^i with its truncated version $\widehat{A}^i$.*

[15] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *ICLR*, 2018

5.6 PAC-Bayes bounds

这些界限由McAllester（1999）[McA99]提出，在原则上是最紧的，意味着本章中之前的界限是其子情况。它们源自一个古老的哲学传统，即考虑信念系统的逻辑基础，这通常使用贝叶斯定理。例如，在18世纪，拉普拉斯试图给“*What is the probability that the sun will rise tomorrow?*”等问题赋予意义。这个问题的答案取决于个人的先验信念（例如，科学知识的程度）以及他们观察到的太阳在他们一生中每天升起的事实。这种哲学联系有时有助于学生提高他们对泛化的理解。

在机器学习（ML）的上下文中，PAC-Bayes 界限假设实验者（即，机器学习专家）对假设 $\mathcal{H}$ 有一些先验分布 P。如果被要求在没有看到任何具体训练数据的情况下进行分类，实验者会根据 P (（表示为 $h \sim P$)）选择一个假设 h 并使用它进行分类 h。在看到训练数据并运行计算后，实验者的分布变为后验 Q，这意味着现在如果被要求进行分类，他们会选择 $h \sim Q$ 并使用它。因此，期望测试损失是

$$\mathop{\mathbb{E}}_{h\sim Q}[L_{\mathcal{D}}(h)].$$

该理论要求 Q 是相对于 P 的 *valid posterior*，这意味着在 P 下得到零概率的每个假设 h 也必须在 Q 下具有零概率。以下形式的 PAC-贝叶斯界限来自 16。

16 约翰·兰福德。*Quantitatively tight sample complexity bounds*。博士论文，2002年

定理5.6.1（PAC-Bayes界）。*Let $\mathcal{D}$ be the data distribution and P be a prior distribution over hypothesis class $\mathcal{H}$ and $\delta > 0$. If S is a set of i.i.d. samples of size m from $\mathcal{D}$ and Q is any valid posterior (possibly depending arbitrarily on S) then $\Delta_S(Q) = \mathbb{E}_{h\sim Q}[L_{\mathcal{D}}(h) - \widehat{L}_S(h)]$ satisfies the following bound with probability $1-\delta$,*

$$\Delta_S(Q) \leq \sqrt{\frac{D(Q||P) + \ln(2m/\delta)}{2(m-1)}},$$

where $D(Q||P) = \mathbb{E}_{h\sim Q}[\ln(Q(h)/P(h))]$ is the so-called KL-diver gence[17].

换句话说，泛化误差可以使用分布的KL散度（平方根）的上界来表示，再加上一些由集中界限引起的项。

17 这是一种衡量分布之间距离的度量，当 P 在 Q 中占主导地位时具有意义，即每个在 Q 中具有非零概率的 h 也具有非零概率在 P 中。请注意，在这个定义中，$0 \ln 0$ 被解释为 0。

示例5.6.2. *P could be the standard normal distribution, which assigns nonzero probability to every vector. For any sample set S, we could let Q be*

the distribution on parameter vectors obtained by vanilla deep learning using S: that is, initialize parameters using random Gaaussian, and train with SGD with a predetermined learning rate schedule. Since SGD is a stochastic process (due to randomness of batches) it leads to a natural distribution Q on trained classifiers at the end of training. Notice, Q is a valid posterior of P because P assigns nonzero probability to every classifier h. As this example emphasizes, one can consider various P and Q for the same classification setup (e.g., by changing some aspect of training) and the generalization bound will hold for every fixed choice.

示例5.6.3. *Suppose h is any classifier and P, Q are the distribution that assigns probability 1 to h and zero to all other hypotheses. Then $D(Q||P) = 0$, and by Hoeffding bound we have $\Delta_S(Q) = \Delta_S(h) \leq \sqrt{\frac{\log(1/\delta)}{2m}}$. The inequality in PAC-Bayes bound is satisfied.*

问题 5.6.4. *Derive the union bound Theorem 5.2.1 using PAC-Bayes.*

现在我们准备证明定理5.6.1。为了阐述方便，我们证明一个定性相似但并不完全正确的较弱命题：

$$\Delta_S(Q) \leq \sqrt{\frac{2(D(Q||P) + \ln(2/\delta))}{m}} \tag{5.8}$$

由于对数量 $z = \sqrt{m}\Delta_S(h)$ 的简化假设不正确，其中 h 是一个固定的分类器，S 是 m 样本的一个随机子集。由于 $\Delta_S(h)$ 是取值在 $[-1, 1]$ 内且均值为 0 的 m 个独立同分布变量的平均值，我们假设 z 的行为 *exactly* 类似于正态分布 $\mathcal{N}(0, 1)$。当然，在极限 $m \to \infty$ 中，z 在分布上被 $\mathcal{N}(0, 1)$ 所支配。可以通过使用 Hoeffding 界的更定量论证来去除这个假设。这个假设允许我们假设当 x 从 $\mathcal{N}(0, 1)$ 中抽取，且 ϵ 是一个任意小的常数时，$e^{z^2/(2+\epsilon)}$ 的期望值趋近于 $\sqrt{2}$。为了简单起见，我们将假设 $e^{z^2/2} = \sqrt{2}$。可以使用集中界限来修复证明。

Proof. (定理5.6.1，较弱版本（5.8)）重新排列定理陈述中的表达式，我们看到它给出了$(m/2)\,\mathbb{E}_{h\sim Q}[\Delta_S(h)]^2 - D(Q||P)$上$\ln(2/\delta)$的上界。根据Jensen不等式 [18]

应用于平方函数 {v*}，此表达式至多为 $(m/2)\,\mathbb{E}_{h\sim Q}[\Delta_S(h)^2] - D(Q||P)$。我们证明这是上界于 $\ln(2/\delta)$。步骤如下：

$$\begin{aligned}
&= \mathop{\mathbb{E}}_{h\sim Q}\left[(m/2)\Delta_S(h)^2 - \ln(Q(h)/P(h))\right] \\
&= \mathop{\mathbb{E}}_{h\sim Q}\left[\ln\left(\exp((m/2)\Delta_S(h)^2)\cdot P(h)/Q(h)\right)\right] \\
&\leq \ln\left(\mathop{\mathbb{E}}_{h\sim Q}\left[\exp((m/2)\Delta_S(h)^2)\cdot P(h)/Q(h)\right]\right),
\end{aligned}$$

18 Jensen不等式：对于凹函数f和随机变量 X，$\mathbb{E}[f(X)] \leq f(\mathbb{E}[X])$。对于凸函数，不等式方向相反。

在最后一个不等式中使用了Jensen不等式以及ln的凹性。此外，由于对 $h \sim Q$取期望实际上类似于一个加权求和，其中h的项由$Q(h)$加权，我们有19

19 经常在机器学习中看到KL散度时，你会看到这个技巧被用来切换取期望的分布！

$$\ln \mathop{\mathbb{E}}_{h\sim Q}\left[\exp((m/2)\Delta(h)^2)\cdot P(h)/Q(h)\right] = \ln \mathop{\mathbb{E}}_{h\sim P}\left[\exp((m/2)\Delta(h)^2)\right].$$

总结来说，我们因此展示了以下针对固定数据集 S 的内容：

$$(m/2)\mathop{\mathbb{E}}_{h\sim Q}[\Delta_S(h)]^2 - D(Q||P) \le \ln\left(\mathop{\mathbb{E}}_{h\sim P}\left[e^{(m/2)\Delta_S(h)^2}\right]\right) \tag{5.9}$$

注意，右侧没有对后验 Q 的依赖。使用 S 是大小为 m 的随机样本的事实，以及先验信念 P 在看到 S (之前是固定的，即独立于 S)：

$$\mathop{\mathbb{E}}_{S}\left[\mathop{\mathbb{E}}_{h\sim P}\left[\, e^{(m/2)\Delta_S(h)^2}\right]\right] = \mathop{\mathbb{E}}_{h\sim P}\left[\mathop{\mathbb{E}}_{S}\left[e^{(m/2)\Delta_S(h)^2}\right]\right] = \sqrt{2} \le 2.$$

简单平均意味着在概率1下 $-\delta$ 在 S 上，

$$\mathop{\mathbb{E}}_{h\sim P}\left[e^{(m/2)\Delta_S(h)^2}\right] \le 2/\delta \tag{5.10}$$

现在通过对两边取对数，证明完成。

□

5.7 *Exercises*

1. 假设损失函数 ℓ 是 1-Lipschitz。考虑我们在第 3.2 节中研究的核分类器形式 $h(x) = z^\top G^{-1} y$，其中 G 是 $n \times n$ 核矩阵，y 是标签，z 是列向量，其 i-th 坐标是 $K(x, x_i)$。证明其 Rademacher 复杂度是有上界的 $\sqrt{2y^\top G y \cdot \mathrm{Tr}(G)/n}$。（提示：将核分类器视为 RKHS 中的线性分类器）

6
Tractable Landscapes for Nonconvex Optimization

深度学习依赖于优化复杂、非凸损失函数。在最坏情况下，寻找非凸目标的全局最小值是NP难的。然而，在深度学习中，简单的算法如随机梯度下降通常在结束时将目标值降至零或接近零。本章重点研究由非凸目标定义的*optimization landscape*，并确定了这些景观的性质，这些性质允许简单的优化算法找到全局最小值（或接近最小值）。迄今为止，这些性质适用于比深度学习更简单的非凸问题，而如何用这种景观分析来分析深度学习仍然是开放的。

Warm-up: Convex Optimization 为了理解优化景观，可以先看看如何优化一个凸函数。如果一个函数$f(w)$是凸函数，那么它满足许多很好的性质，包括

$$\forall \alpha \in [0,1], w, w', f(\alpha w + (1-\alpha)w') \leq \alpha f(w) + (1-\alpha) f(w'). \quad (6.1)$$

$$\forall w, w', f(w') \geq f(w) + \langle \nabla f(w), w' - w \rangle. \quad (6.2)$$

这些方程描述了目标函数$f(w)$的重要几何属性。特别是，方程（6.1）表明$f(w)$的所有全局最小值必须相互连接，因为如果w和w'都是全球最优的，那么$\alpha w + (1-\alpha)w'$段上的任何东西也必须是最优的。这些属性很重要，因为它给出了所有全局最小值的特征。方程（6.2）表明，所有具有$\nabla f(w) = 0$的点都必须是全局最小值，因为对于每个w'，我们都有$f(w') \geq f(w) + \langle \nabla f(w), w' - w \rangle \geq f(w)$。这些属性很重要，因为它将一个局部属性（梯度为0）与全局最优性联系起来。

通常，优化景观寻找表征其局部/全局最优点的目标函数的性质（例如方程（6.1））或连接局部性质与全局最优性（例如方程（6.2））。

6.1 Preliminaries and challenges in nonconvex landscapes

我们一直在非正式地讨论全局/局部最小值，这里我们首先给出一个精确的定义：

定义6.1.1（全局/局部最小值）。*For an objective function* $f(w)$: $\mathbb{R}^d \to \mathbb{R}$*, a point* w^* *is a* 全局最小值 *if for every* w *we have* $f(w^*) \leq f(w)$*. A point* w *is a* 局部最小/最大值 *if there exists a radius* $\epsilon > 0$ *such that for every* $\|w' - w\|_2 \leq \epsilon$*, we have* $f(w) \leq f(w')$ $(f(w) \geq f(w')$ *for local maximum). A point* w *with* $\nabla f(w) = 0$ *is called a* 临界点*, for smooth functions all local minimum/maximum are critical points.*

在整个章节中，我们将始终处理全局最小值存在的函数，并使用 $f(w^*)$ 表示函数的最优值1。为了简单起见，我们专注于优化

尽管可能存在多个全局最小值 w^*，但根据定义，$f(w^*)$ 的值是唯一的。

问题没有任何约束($w \in \mathbb{R}^d$)。可以将本章中的所有内容扩展到具有非退化的等式约束的优化，这将需要相对于流形的梯度和Hessian的定义，但这超出了本书的范围。

Spurious local minimum 第一个非凸优化的障碍是一个 *spurious local minimum*。

定义6.1.2（虚假局部最小值）。*For an objective function* $f(w)$: $\mathbb{R}^d \to \mathbb{R}$*, a point* w *is a spurious local minimum if it is a local minimum, but* $f(w) > f(w^*)$*.*

许多简单的优化算法基于局部搜索的思想，因此无法逃离虚假的局部最小值。正如我们稍后将会看到的，许多非凸目标没有虚假的局部最小值。

Saddle points 非凸优化的第二个障碍是一个 *saddle point*。鞍点的最简单例子是在点 $w = (0, 0)$ 的 $f(w) = w_1^2 - w_2^2$。在这种情况下，如果 w 沿着方向 $(\pm 1, 0)$ 移动，函数值会增加；如果 w 沿着方向 $(0, \pm 1)$ 移动，函数值会减少。

定义 6.1.3（鞍点）。*For an objective function* $f(w) : \mathbb{R}^d \to \mathbb{R}$*, a point* w *is a saddle point if* $\nabla f(w) = 0$*, and for every radius* $\epsilon > 0$*, there exists* w^+*,* w^- *within distance* ϵ *of* w *such that* $f(w^-) < f(w) < f(w^+)$*.*

这个定义涵盖了所有情况，但很难验证一个点是否是鞍点。在大多数情况下，可以判断

基于其Hessian矩阵，判断一个点是否为鞍点、局部最小值或局部最大值。

主张6.1.4. *For an objective function* $f(w) : \mathbb{R}^d \to \mathbb{R}$ *and a critical point* w ($\nabla f(w) = 0$)*, we know*

- *If* $\nabla 2 f(w) \succ 0$*,* w *is a local minimum.*
- *If* $\nabla 2 f(w) \prec 0$*,* w *is a local maximum.*
- *If* $\nabla^2 f(w)$ *has both a positive and a negative eigenvalue,* w *is a saddle point.*

这些标准被称为优化中的二阶充分条件。直观上，可以通过观察二阶泰勒展开来证明这个命题。命题中的三个情况并没有涵盖所有可能的Hessian矩阵。剩余的情况被认为是退化的，可以是局部最小值、局部最大值或鞍点2。

2 可以考虑函数 $w = 0$ 的点 w^4，$-w^4$，w^3，它分别是局部极小值、极大值和鞍点。

Flat regions 即使一个函数没有任何虚假的局部极小值或鞍点，它仍然可能是非凸的，见图6.1。在高维情况下，这类函数仍然可能非常难以优化。这里的困难主要在于，即使范数$\|\nabla f(w)\|_2$很小，与凸函数不同，不能得出$f(w)$接近$f(w^*)$的结论。然而，在许多情况下，人们可以希望函数$f(w)$满足某种放宽的凸性概念，并据此设计有效的算法。我们在第6.2节讨论了这种情况之一。

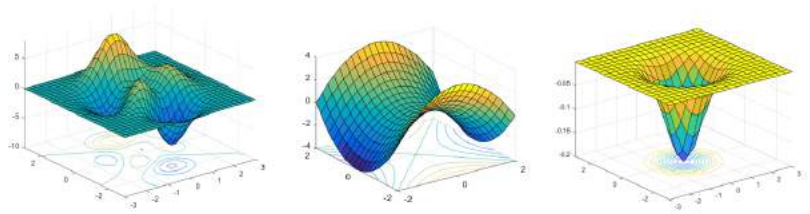


图6.1：非凸优化的障碍。从左到右：局部最小值、鞍点和平坦区域。

6.2 *Cases with a unique global minimum*

我们首先考虑与凸目标最相似的案例。在本节中，我们考虑的目标函数没有虚假的局部极小值或鞍点。事实上，在我们的例子中，目标函数将只有一个唯一的全局最小值。优化这些函数的唯一障碍是梯度小的点可能不是最优的。

这里的主要思想是识别目标函数的性质以及一个*potential function*，使得当我们运行如梯度下降等简单优化算法时，势函数保持递减。前人文献中使用了许多性质，包括

定义 6.2.1. *Let $f(w)$ be an objective function with a unique global minimum w^*, then*

Polyak-Lojasiewicz *f satisfies Polyak-Lojasiewicz if there exists a value $\mu > 0$ such that for every w, $\|\nabla f(x)\|_2^2 \geq \mu(f(w) - f(w^*))$.*

weakly-quasi-convex *f is weakly-quasi-convex if there exists a value $\tau > 0$ such that for every w, $\langle \nabla f(w), \ w - w^* \rangle \geq \mu(f(w) - f(w^*))$.*

Restricted Secant Inequality (RSI) *f satisfies RSI if there exists a value τ such that for every w, $\langle \nabla f(w), w - w^* \rangle \geq \mu \|w - w^*\|_2^2$.*

任何这三个属性中的任何一个都可以意味着快速收敛，同时伴随着f的某些平滑性。

主张6.2.2. *If an objective function f satisfies one of Polyak-Lojasiewicz, weakly-quasi-convex or RSI, and f is smooth*[3]*, then gradient descent converges to global minimum with a geometric rate*[4].

直观上，Polyak-Lojasiewicz条件要求对于任何不是全局最小点的点，梯度必须非零，因此可以始终跟随梯度并进一步降低函数值。此条件在某些全局最小点不唯一的情况下也可以工作。弱拟凸和RSI在以下意义上相似，即它们都要求（负）梯度与正确的方向相关联——从当前点w到全局最小点w^*的方向。

3 波利亚克-洛贾西耶维奇和RSI需要与方程（2.2）中相同的标准光滑性定义，弱拟凸性需要详细在[HMR18]中描述的特殊光滑性属性。

4 聚类-洛加西耶维奇和弱拟凸的势函数是函数值f；RSI 的势函数是平方距离$\|w -$

在这个部分，我们将使用广义线性模型作为例子，展示如何使用这些属性。

6.2.1 Generalized linear model

在广义线性模型（也称为等调回归）[KS09, KKSK11]中，输入由从分布$\mathcal{D}$中抽取的样本$\{x^{(i)}$、$y^{(i)}\}$组成，其中$(x, \ y) \sim \mathcal{D}$满足

$$y = \sigma(w_*^\top x) + \epsilon. \tag{6.3}$$

这里 $\sigma : \mathbb{R} \to \mathbb{R}$ 是一个已知的单调函数，ϵ 是满足 $\mathbb{E}[\epsilon|x] = 0$ 的噪声，而 w_* 是我们试图学习的未知参数。

在这种情况下，考虑以下预期损失是自然的{v*}

$$L(w) = \frac{1}{2} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[(y - \sigma(w^\top x)^2 \right]. \tag{6.4}$$

当然，在实践中，人们只能访问训练损失，它是观察到的 $\{x^{(i)}$、$y^{(i)}\}$ 对的平均值。为了简单起见，我们在这里使用期望损失。两个损失之间的差异可以使用第 ?? 章中的技术进行界定。

广义线性模型可以看作是学习一个单神经元，其中 σ 是其非线性。

我们将给出如何证明广义线性模型中弱拟凸或RSI等性质的高级思路。首先，我们将目标函数重写为：

$$\begin{aligned} L(w) &= \frac{1}{2} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[(y - \sigma(w^\top x)^2 \right] \\ &= \frac{1}{2} \mathop{\mathbb{E}}_{(x,\epsilon)} \left[(\epsilon + \sigma(w_*^\top x) - \sigma(w^\top x))^2 \right]. \\ &= \frac{1}{2} \mathop{\mathbb{E}}_{\epsilon} \left[\epsilon^2 \right] + \frac{1}{2} \mathop{\mathbb{E}}_{x} \left[(\sigma(w_*^\top x) - \sigma(w^\top x)^2 \right]. \end{aligned}$$

这里第二个等式使用了模型的定义（方程（6.3））， 第三个等式使用了事实 $\mathbb{E}[\epsilon|x] = 0$（因此没有交叉项）。这种分解很有帮助，因为第一个项 $\frac{1}{2}\mathbb{E}_\epsilon\left[\epsilon^2\right]$ 现在只是一个常数。

现在我们可以对目标函数求导：

$$\nabla L(w) = \mathop{\mathbb{E}}_{x} \left[(\sigma(w^\top x) - \sigma(w_*^\top x))\sigma'(w^\top x)x \right].$$

注意，弱拟凸和RSI都需要目标函数与$w - w_*$相关联，因此我们计算

$$\langle \nabla L(w), w - w^* \rangle = \mathop{\mathbb{E}}_{x} \left[(\sigma(w^\top x) - \sigma(w_*^\top x))\sigma'(w^\top x)(w^\top x - w_*^\top x) \right].$$

目标是要证明RHS大于0。一个简单的方法是使用介值定理：$\sigma(w^\top x) - \sigma(w_*^\top x) = \sigma'(\xi)(w^\top x - w_*^\top x)$，其中$\xi$是$w^\top x$和$w_*^\top x$之间的一个值。然后我们有

$$\langle \nabla L(w), w - w^* \rangle = \mathop{\mathbb{E}}_{x} \left[\sigma'(\xi)\sigma'(w^\top x)(w^\top x - w_*^\top x)^2 \right].$$

在RHS的期望中，由于σ是单调的，因此两个导数($\sigma'(\xi)$, $\sigma'(w^\top x)$)都是正的，并且$(w^\top x - w_*^\top x)^2$显然是非负的。通过在$\sigma$和$x$的分布上做出更多假设，可以将RHS下界降低到弱拟凸或RSI所需的形式。我们将此作为练习。

6.2.2 *Alternative objective for generalized linear model*

存在另一种方法来找到广义线性模型中的 w_*，这种方法更具体于这种设置。在此方法中，人们估计一个不同的

“梯度”对于广义线性模型：

$$\nabla g(w) = \mathop{\mathbb{E}}_{x,y}\left[(\sigma(w^\top x) - y)x\right] = \mathop{\mathbb{E}}_{x}\left[(\sigma(w^\top x) - \sigma(w_*^\top x))x\right]. \quad (6.5)$$

第一个方程给出了一种估计这个“梯度”的方法。这里的主要区别在于，在等号右边，我们不再像在 $\nabla L(w)$ 中那样有一个因子 $\sigma'(w^\top x)$。当然，不清楚为什么这个公式是某个函数 g 的梯度，但我们可以以下方式构造函数 g：

设 $\tau(x)$ 为 $\sigma(x)$ 的积分：$\tau(x) = \int_0^x \sigma(x')\mathrm{d}x'$。定义 $g(w)$: $= \mathbb{E}_x\left[\tau(w^\top x) - \sigma(w_*^\top x)w^\top x\right]$。可以验证 $\nabla g(w)$ 确实是方程（6.5）中的函数。非常令人惊讶的是 $g(w)$ 实际上是一个具有 $\nabla g(w_*) = 0$ 的凸函数！这意味着 w_* 是 g 的全局最小值，我们只需要遵循 $\nabla g(w)$ 就可以找到它。这里不需要非凸优化。

当然，这项技术相当特殊，在广义线性模型中使用了大量结构。然而，类似的想法也被用于5中学习单个神经元。一般来说，当一个目标难以

[5]

分析，寻找一个具有相同全局最小值但更容易优化的替代目标可能更容易。

6.3 *Symmetry, saddle points and locally optimizable functions*

在上一节中，我们看到了一些条件，这些条件允许非凸目标函数被有效地优化。然而，这些条件通常不适用于神经网络，或者更一般地说，任何具有某些对称性质的功能。

更具体地说，考虑一个两层神经网络 $h_\theta(x)$：$\mathbb{R}^d \to \mathbb{R}$。参数 θ 是 $(w_1, w_2, ..., w_k)$，其中 $w_i \in \mathbb{R}^d$ 表示第 i 个神经元的权重向量。函数可以评估为 $h_\theta(x) = \sum_{i=1}^k \sigma(\langle w_i,\ x\rangle)$，其中 σ 是一个非线性激活函数。给定数据集 $(x^{(1)}, y^{(1)})..., (x^{(n)}, y^{(n)}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$，可以定义训练损失和期望损失，如第 1 章所述。现在，这个神经网络 $f(\theta) = L(h_\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell((x, y), h_\theta)]$ 的目标有 *permutation symmetry*。也就是说，对于任何交换神经元权重的排列 $\pi(\theta)$，我们知道 $f(\theta) = f(\pi(\theta))$。

对称性有许多含义。首先，如果全局最小值 θ^* 是一个不是所有神经元都具有相同权重向量的点（这很可能成立），那么对于每个排列 π 必须存在等效的全局最小值 $f(\pi(\theta^*))$。具有这种对称性的目标也必须是非凸的，因为如果它是凸的，那么 π 在所有排列) 上的和的点 $\bar{\theta} = \frac{1}{k!}\sum_\pi \pi(\theta^*)$ (是全局最小值的凸组合，因此它也必须是全局最小值。然而，对于 $\bar{\theta}$，神经元的权重向量都是

等于 $\frac{1}{k}\sum_{i=1}^{k} w_i$ (其中 w_i 是 i-th 神经元在 θ^*) 中的权重，因此 $h_{\bar{\theta}}(x) = k\sigma(\langle \frac{1}{k}\sum_{i=1}^{k} w_i,\ x\rangle)$ 等价于一个具有单个神经元的神经网络。在大多数情况下，单个神经元网络不应达到全局最小值，因此通过反证法我们知道 f 不应该是凸的。

它还可能表明具有对称性的函数必须有鞍点6。因此，为了优化此类函数，算法需要能够避免或逃离鞍点。更具体地说，人们希望找到 *second order stationary point*。

6 除了某些退化情况，例如常函数。

定义6.3.1（二阶驻点（SOSP））。*For an objec- tive function* $f(w)$: $\mathbb{R}^d \to \mathbb{R}$*, a point* w *is a second order stationary point if* $\nabla f(w) = 0$ *and* $\nabla^2 f(w) \succeq 0$*.*

二阶驻点条件被称为局部最小值二阶必要条件。当然，一般来说，优化算法无法找到精确的二阶驻点（就像在章节 ?? 中我们只展示了梯度下降找到具有小梯度的点，但不是0梯度）。优化算法可以用来找到近似二阶驻点:

定义6.3.2（近似二阶驻点）。*For an objective function* $f(w)$: $\mathbb{R}^d \to \mathbb{R}$*, a point* w *is a* $(\epsilon,\ \gamma)$*-second order stationary point (later abbreviated as* $(\epsilon,$ $\gamma)$*-SOSP) if* $\|\nabla f(w)\|_2 \le \epsilon$ *and* $\lambda_{\min}(\nabla^2 f(w)) \ge -\gamma$*.*

稍后在第??章中，我们将展示简单变种的梯度下降实际上可以高效地找到 (ϵ, γ)-SOSPs。

现在我们准备定义一类可以高效优化并允许对称和鞍点的函数。

定义6.3.3（局部可优化函数）。*An objective function* $f(w)$ *is locally optimizable, if for every* $\tau > 0$*, there exists* $\epsilon,\ \gamma = poly(\tau)$ *such that every* $(\epsilon,\ \gamma)$*-SOSP* w *of* f *satisfies* $f(w) \le f(w_*) + \tau$*.*

大致来说，一个目标函数在局部可优化，如果函数的每一个局部最小值也是全局最小值，并且每一个鞍点的Hessian矩阵具有负特征值。在某些先前结果中，这类函数被称为“严格鞍点”或“可骑乘”函数。许多非凸目标函数，包括矩阵感知[BNS16a, PKCS17, GJZ17a]、矩阵补全[GLM16a, GJZ17a]、字典学习[SQW16a]、相位恢复[SQW18]、张量分解[GHJY15a]、同步问题[BBV16]和某些双层神经网络的优化目标[GLM18]，已知是局部可优化的。

6.4 Case study: top eigenvector of a matrix

在这个部分，我们来看一个局部可优化函数的简单例子。给定一个对称的PSD矩阵 $M \in \mathbb{R}^{d\times d}$，我们的目标是找到它的最大特征向量（对应于最大特征值的特征向量）。更确切地说，使用奇异值分解（SVD），我们可以将 M 写作

$$M = \sum_{i=1}^{d} \lambda_i v_i v_i^\top.$$

这里 v_i 是 M 的正交归一特征向量，λ_i 是特征值。为了简化，我们假设 $\lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_d \geq 0$[7]。

存在许多目标函数，其全局最优解给出最大特征向量。例如，使用谱范数的基本定义，我们知道对于PSD矩阵M，其全局最优解为

$$\max_{\|x\|_2=1} x^\top M x$$

是 M 的最大特征向量。然而，这个公式需要约束。我们使用一个无约束版本，其正确性由 Eckhart-Young 定理得出：

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{4}\|M - xx^\top\|_F^2. \tag{6.6}$$

请注意，这个函数确实具有对称性，即$f(x) = f(-x)$。根据我们的假设，这个函数的唯一全局极小值是 $x = \pm\sqrt{\lambda_1}v_1$。我们将证明这些也是唯一的二阶平稳点。我们将给出两种常用的证明局部可优化性质的证明策略。

[7] 注意，这里唯一的真实假设是 $\lambda_1 > \lambda_2$，因此最大的特征向量是唯一的。其他不等式在不妨碍一般性的前提下成立。

6.4.1 Characterizing all critical points

第一个想法很简单——我们只需尝试求解方程 $\nabla f(x) = 0$ 以获得所有临界点的位置；然后对于不是期望的全局最小值的临界点，尝试证明它们是局部最大值或鞍点。

Computing gradient and Hessian 在解决方程 $\nabla f(x) = 0$ 对于在方程 (6.6) 中定义的目标函数$f(x)$ 之前，我们首先给出一种计算梯度和国赫森的方法。我们首先将$f(x+\delta)$ (展开，其中 δ 应该被视为一个小的扰动-

bation):

$$\begin{aligned} f(x+\delta) &= \frac{1}{4}\|M-(x+\delta)(x+\delta)^\top\|_F^2 \\ &= \frac{1}{4}\|M-xx^\top-(x\delta^\top+\delta x^\top)-\delta\delta^\top\|_F^2 \\ &= \frac{1}{4}\|M-xx^\top\|_F^2-\frac{1}{2}\langle M-xx^\top, x\delta+\delta x^\top\rangle \\ &\quad+\left[\frac{1}{4}\|x\delta^\top+\delta x^\top\|_F^2-\frac{1}{2}\langle M-xx^\top,\delta\delta^\top\rangle\right]+o(\|\delta\|_2^2). \end{aligned}$$

注意，在最后一步，我们根据 δ 的度数收集了项，并忽略了所有小于 $o(\|\delta\|_2^2)$ 的项。现在我们可以将这个表达式与 $f(x+\delta)$ 的泰勒展开式进行比较：

$$f(x+\delta)=f(x)+\langle\nabla f(x),\delta\rangle+\frac{1}{2}\delta^\top[\nabla^2 f(x)]\delta+o(\|\delta\|_2^2).$$

通过匹配术语，我们立即得到

$$\begin{aligned} \langle\nabla f(x),\delta\rangle &= -\frac{1}{2}\langle M-xx^\top, x\delta^\top+\delta x^\top\rangle, \\ \delta^\top[\nabla^2 f(x)]\delta &= \frac{1}{2}\|x\delta^\top+\delta x^\top\|_F^2-\langle M-xx^\top,\delta\delta^\top\rangle. \end{aligned}$$

这些可以简化为给出实际的梯度和Hessian[8]

实际上，在下一小节中，我们将看到通常只需要知道如何计算 $\langle\nabla f(x)、\delta\rangle$ 和 $\delta^\top[\nabla^2 f(x)]\delta$ 就足够了。

$$\nabla f(x)=(xx^\top-M)x,\quad \nabla^2 f(x)=\|x\|_2^2 I+2xx^\top-M. \tag{6.7}$$

Characterizing critical points 现在我们可以执行原始计划。首先设置 $\nabla f(x)=$ 为 0，我们有

$$Mx=xx^\top x=\|x\|_2^2 x.$$

幸运的是，这是一个研究得很好的方程，因为我们知道$Mx = \lambda x$的唯一解是当λ是特征值且x是对应特征向量的（缩放版本）时。因此，我们知道$x = \pm\sqrt{\lambda_i}v_i$或$x = 0$。这些都是目标函数$f(x)$的唯一临界点。

在这些关键点中，$x = \pm\sqrt{\lambda_1}v_1$ 是我们的解决方案。接下来，我们需要证明对于每个其他关键点，其Hessian有一个负的特征方向。我们将为 $x = \pm\sqrt{\lambda_i}v_i(i>1)$ 做这件事。根据定义，只需证明存在一个 δ，使得 $\delta^\top[\nabla^2 f(x)]\delta<0$。证明的主要步骤涉及猜测这个方向是什么 δ。在这种情况下，我们将选择 $\delta = v_1$ (，我们将在下一小节中给出更多关于如何选择这样一个方向的本能。

当 $x = \pm\sqrt{\lambda_i}v_i$ 和 $\delta = v_1$ 时，我们有

$$\delta^\top[\nabla^2 f(x)]\delta = v_1^\top[\|\sqrt{\lambda_i}v_i\|_2^2 I + 2\lambda_i v_i v_i^\top - M]v_1 = \lambda_i - \lambda_1 < 0.$$

这里最后一步使用了 v_i 是正交归一向量的事实和 $v_1^\top M v_1 = \lambda_1$。$x = 0$ 的证明非常相似。结合上述所有步骤，我们证明了以下命题：

断言6.4.1（临界点的性质）。*The only critical points of $f(x)$ are of the form $x = \pm\sqrt{\lambda_i}v_i$ or $x = 0$. For all critical points except $x = \pm\sqrt{\lambda_1}v_1$, $\nabla^2 f(x)$ has a negative eigenvalue.*

这个断言直接意味着唯一的二阶平稳点是 $x = \pm\sqrt{\lambda_1}v_1$，因此所有二阶平稳点也都是全局最小值。

6.4.2 Finding directions of improvements

第6.4.1节中的方法很简单。然而，在更复杂的问题中，通常无法枚举所有$\nabla f(x) = 0$的解。我们在第6.4.1节中证明的也不足以证明$f(x)$是局部可优化的，因为我们只证明了每个精确SOSP都是全局最小值，而局部可优化函数要求每个近似SOSP都接近全局最小值。现在我们将给出一种更灵活、更稳健的替代方法。

对于每个不是全局最小值的点 x，我们定义其改进方向如下：

定义6.4.2（改进方向）。*For an objective function f and a point x, we say δ is a direction of improvement (of f at x) if $|\langle\nabla f(x), \delta\rangle| > 0$ or $\delta^\top[\nabla^2 f(x)]\delta < 0$. We say δ is an $(epsilon, \gamma)$ direc- tion of improvement (of f at x) if $|\langle\nabla f(x), \delta\rangle| > \epsilon\|\delta\|_2$ or $\delta^\top[\nabla^2 f(x)]\delta < -\gamma\|\delta\|_2^2$.*

直观上，如果 δ 是 f 在 x 处的改进方向，那么沿着 δ 或 $-\delta$ 中的一个以足够小的步长移动可以减小目标函数。事实上，如果一个点 x 有改进方向，它不能是二阶驻点；如果一个点 x 有 $(epsilon, \gamma)$ 改进方向，那么它不能是 (ϵ, γ)-SOSP。

现在我们可以看看在局部可优化函数定义中我们试图证明的逆命题：如果每个点 x 与 $f(x) > f(x^*) + \tau$ 都有一个 (ϵ, γ) 改进方向，那么每个 (ϵ, γ)-二阶驻点必须满足 $f(x) \leq f(x^*) + \delta$。因此，我们在这部分的目标是找到每个非全局最优点的改进方向。

为了简单起见，我们将研究更简单的顶阶特征向量问题的一个版本。特别是，我们考虑$M = zz^\top$是一个秩为1的矩阵，而z是一个单位向量的情况。在这种情况下，我们在方程（6.6）中定义的目标函数变为

$$\min_x f(x) = \frac{1}{4}\|zz^\top - xx^\top\|_F^2. \tag{6.8}$$

目标全局最优解是 $x = \pm z$。这个问题通常被称为矩阵分解问题，因为我们给定一个矩阵 $M = zz^\top$[9]，目标是找到一个分解 $M = xx^\top$。

[9] Note that we only observe M, not z.

我们应该朝哪个方向移动以减小目标函数？在这个问题中，我们只有最优方向 z 和当前方向 x，所以自然的猜测会是 z、x 或 $z - x$。实际上，这些方向就足够了：

引理 6.4.3. *For objective function* (6.8)*, there exists a universal constant* $c > 0$ *such that for any* $\tau < 1$*, if neither* x *or* z *is an* $(c\tau, \quad , \quad 1/4$ *)-direction of improvement for the point* x*, then* $f(x) \le \tau$*.*

这个引理的证明涉及一些详细的计算。为了获得一些直观感受，我们首先可以思考如果 neither x nor z 是改进方向会发生什么。

引理 6.4.4. *For objective function* (6.8)*, if neither* x *or* z *is a direction of improvement of* f *at* x*, then* $f(x) = 0$*.*

Proof. 我们将使用与方程（6.7）相同的梯度和国赫森计算方法，除了现在的 M 变为 $zz^\top$。首先，由于 x 不是一个改进的方向，我们必须有

$$\langle \nabla f(x), x\rangle = 0 \implies \|x\|_2^4 = \langle x, z\rangle^2. \tag{6.9}$$

如果 z 不是一个改进方向，我们知道 $z^\top[\nabla^2 f(x)]z \ge 0$，这意味着

$$\|x\|^2 + 2\langle x, z\rangle^2 - 1 \ge 0 \implies \|x\|^2 \ge 1/3.$$

这里我们使用了 $\langle x,\ z\rangle^2 \le \|x\|_2^2\|z\|_2^2 = \|x\|_2^2$ 的事实。结合方程（6.9），我们知道 $\langle x,\ z\rangle^2 = \|x\|_2^4 \ge 1/9$。

最后，由于 z 不是一个改进的方向，我们知道 $\langle \nabla f(x),\ z\rangle = 0$，这意味着 $\langle x,\ z\rangle(\|x\|_2^2 - 1) = 0$。我们已经证明了 $\langle x,\ z\rangle^2 \ge 1/9 > 0$，因此 $\|x\|_2^2 = 1$。再次结合方程（6.9），我们知道 $\langle x,\ z\rangle^2 = \|x\|_2^4 = 1$。唯一两个具有 $\langle x, z\rangle^2 = 1$ 和 $\|x\|_2^2 = 1$ 的向量是 $x = \pm z$。

□

证明引理6.4.3的方法与引理6.4.4非常相似，除了我们需要允许在每个方程和不等式中使用松弛项。拥有更稳健的引理6.4.3的额外好处是，我们可以在...

证明即使在无法访问确切目标的情况下也具有鲁棒性 - 在仅有一组坐标 $zz^\top$[10] 的子集的设置中，人们仍然可以证明目标函数是局部可优化的，因此通过非凸优化找到 z。

[10] 此设置被称为 *matrix completion* 并已广泛应用于推荐系统。

引理6.4.4和引理6.4.3都使用了方向x和z。当$\langle x, z\rangle \geq 0$（以及当$\langle x, z\rangle < 0$时使用$x + z$）时，也可以使用方向$x - z$。这两个想法都可以推广到处理$M = ZZ^\top$的情况，其中$Z \in \mathbb{R}^{d\times r}$，因此$M$是一个秩-$r$矩阵。

7
Escaping Saddle Points

梯度下降（GD）和随机梯度下降（SGD）是大规模机器学习的动力源泉。虽然经典理论侧重于分析这些方法在 *convex* 优化问题中的性能，但机器学习中最显著的成功涉及 *nonconvex* 优化，理论与实践之间出现了差距。

确实，GD和SGD的传统分析表明，这两种算法都有效地收敛到平稳点。但这些分析没有考虑到收敛到鞍点的可能性。受第十章中几何特性的启发，解决许多非凸机器学习问题的核心困难在于逃离鞍点。

在这一章中，我们将讨论一种简单的扰动形式的梯度下降，它能够非常有效地逃离鞍点。特别是，在收敛速度和维度依赖性方面，几乎就像鞍点根本不存在一样！

7.1 Preliminaries

≪Chi notes: 许多定义在早期章节中出现过。可能需要协调。≫

在这一章中，我们关注解决如下形式的通用无约束优化问题：{v*}

$$\min_{x\in\mathbb{R}^d} f(x),$$

在 f 是一个可以非凸的光滑函数的情况下。特别是，我们假设 f 具有Lipschitz梯度和李普希茨Hessian，这确保了梯度和Hessian不会变化得太快。

DeFi定义7.1.1. *A differentiable function f is ℓ-**gradient Lipschitz** if:*

$$\|\nabla f(x_1) - \nabla f(x_2)\| \le \ell \|x_1 - x_2\| \quad \forall\, x_1, x_2.$$

定义7.1.2. *A twice-differentiable function f is* ***ρ-Hessian Lipschitz*** *if:*

$$\left\|\nabla^2 f(x_1) - \nabla^2 f(x_2)\right\| \leq \rho \left\|x_1 - x_2\right\| \quad \forall\, x_1, x_2.$$

对于最小化问题，鞍点和局部极大值都是明显不希望的。我们的重点将是“鞍点”，尽管我们的结果也直接适用于局部极大值。不幸的是，对于光滑函数，区分鞍点和局部极小值在一般情况下仍然是NP难的[Nes 00]。为了避免这些困难的结果，我们关注鞍点的一个子类。

定义7.1.3（严格鞍点）。*For a twice-differentiable function f, x is a* ***strict saddle point*** *if x is a stationary point and $\lambda_{\min}(\nabla^2 f(x)) < 0$.*

一个通用的鞍点必须满足 $\lambda_{\min}(\nabla^2 f(\mathrm{x})) \leq 0$。简单地“严格”排除 $\lambda_{\min}(\nabla^2 f(\mathrm{x})) = 0$ 的情况。我们将我们的目标重新表述为寻找不是严格鞍点的驻点。

定义7.1.4（SOSP）。*For twice-differentiable function $f(\cdot)$, x is a* ***second-order stationary point*** *if*

$$\nabla f(x) = 0, \quad \textit{and} \quad \nabla^2 f(x) \succeq 0.$$

定义7.1.5 (ϵ-SOSP). *For a ρ-Hessian Lipschitz function $f(\cdot)$, x is an* ***ϵ-second-order stationary point*** *if:*

$$\|\nabla f(x)\| \leq \epsilon \quad \textit{and} \quad \nabla^2 f(x) \succeq -\sqrt{\rho\epsilon}\cdot I.$$

定义7.1.5描述了SOSP的一个ϵ-近似版本，以便我们可以讨论速率。定义7.1.5中对Hessian的条件使用Hessian Lipschitz参数ρ来保留一个精度参数，并使梯度和Hessian的单位相匹配1，

遵循[NP06]的惯例。

通过匹配“单位”，我们可以使优化结果对简单缩放 $g(x) = af(bx)$ 对于标量 a、$b > 0$ 无关。在这里，ρ 具有三阶导数的缩放，ϵ 具有一阶导数的缩放，因此 $\sqrt{\rho\epsilon}$ 具有二阶导数的缩放。

7.2 *Perturbed Gradient Descent*

根据更新方程，梯度下降（GD）仅在梯度非零时进行非零步长，因此在非凸设置中，如果初始化在鞍点，它将陷入鞍点。因此，我们考虑GD的一个简单变体，该变体在每一步迭代中添加扰动。

$$\mathrm{x}_{t+1} \leftarrow \mathrm{x}_t - \eta(\nabla f(\mathrm{x}_t) + \xi_t), \qquad \xi_t \sim \mathcal{N}(0, (r^2/d)\mathrm{I})$$

在每次迭代中，扰动梯度下降（PGD）几乎与梯度下降相同，除了它添加了一个小的各向同性随机

高斯扰动梯度。扰动 ξ_t 从均值为零的高斯分布中采样，协方差为 $(r^2/d)\mathrm{I}$，因此 $\mathbb{E}\,\|\xi_t\|^2 = r^2$。参数 r 控制扰动的有效半径，通常选择非常小。[JNG+19] 证明了这种简单的 PGD 形式能够逃离严格鞍点并有效地找到 SOSP。

在这一章中，为了展示PGD背后的见解，我们转向算法的另一种变体，其形式略复杂，但分析更容易。我们在此考虑的变体在每个迭代中执行以下两个步骤：

1. 如果 $\|\nabla f(\mathrm{x}_t)\| \le \epsilon$ 且在最后 $\mathcal{T}$步中没有添加扰动，则添加小的扰动 $\mathrm{x}_t \leftarrow \mathrm{x}_t - \eta\xi_t$ 其中 $\xi_t \sim$ 均匀($\mathbb{B}_0(r)$)。

2. $\mathrm{x}_{t+1} \leftarrow \mathrm{x}_t - \eta\nabla f(\mathrm{x}_t)$。

$\mathbb{B}_0(r)$ 是以 0 为中心、半径为 r 的欧几里得球。这种 PGD 变体仅在梯度较小时添加扰动，并且在最后 $\mathcal{T}$步中没有添加扰动，因此与原始 PGD 形式相比，向算法中添加了更少的随机性。在以下定理中，我们提供了对这种 PGD 变体的理论保证如下：

定理7.2.1. *Assume f is ℓ-gradient Lipschitz, and ρ-Hessian Lipschitz. For any ϵ, $\delta > 0$, if we choose $\eta = 1/\ell$, $r = \widetilde{\Theta}(\epsilon)$, $\mathcal{T} = \widetilde{\Theta}(\ell/\sqrt{\rho\epsilon})$, and run PGD for more than $\widetilde{O}(\ell(f(x_0) - f^\star)/\epsilon^2)$ iterations, then with probability at least $1-\delta$, at least one of the iterates will be ϵ-SOSP.*

这里$\widetilde{O}(\cdot)$，$\widetilde{\Theta}(\cdot)$ 在 d，ℓ，ρ，ϵ，δ 和$f(x_0) - f^\star$ 中隐藏了绝对常量和多项式对数依赖性。我们的选择 $\mathcal{T}$在对数因子上，是梯度Lipschitz参数 ℓ 和 ϵ-SOSP 中的Hessian精度容忍度的比率 — $\sqrt{\rho\epsilon}$。

我们指出，在经典优化文献中，已知GD在$O(\ell(f(\mathrm{x}_0) - f^\star)/\epsilon^2)$次迭代中找到一个$\epsilon$一阶驻点（满足$\|\nabla f(\mathrm{x})\| \le \epsilon$的点）[Nes98]。定理7.2.1表明，PGD在几乎与GD找到一阶驻点相同的时间内找到二阶驻点，仅相差对数因子。特别是，尽管在鞍点中可能在d维空间中只有一个逃逸方向，但PGD的维度依赖性仅为多项式对数。这在高维设置中非常重要，例如训练深度神经网络。这提供了一个令人信服的解释，为什么严格鞍点对于一阶梯度方法在计算上是良性的。

整体证明策略如下。根据定义7.1.5，如果一个迭代点x_t不是ϵ二阶平稳点，那么x_t

必须具有大的梯度或是一个近似鞍点。我们证明以下两个命题:

1. 大梯度($\|\nabla f(x_t)\| > \epsilon$)，然后函数值在一步中显著下降：$f(x_{t+1}) - f(x_t) \leq -\Omega(\epsilon^2/\ell)$。

2. 近似鞍点（$\|\nabla f(x_t)\| \leq \epsilon$ 和 $\lambda_{\min}(\nabla^2 f(x_t)) < -\sqrt{\rho\epsilon}$），然后，以高概率，函数值在 $\mathcal{T}$ 步中显著下降：$f(x_{t+\mathcal{T}}) - f(x_t) \leq -\widetilde{\Omega}(\mathcal{T} \cdot \epsilon^2/\ell)$。

这意味着，在两种情况下，函数值在每个步骤中平均将减少 $\widetilde{\Omega}(\epsilon^2/\ell)$。由于函数值不能低于最优值 $f^\star$，我们知道在 $\widetilde{O}(\ell(f(x_0) - f^\star)/\epsilon^2)$ 步骤后，至少有一个迭代点必须是 ϵ-二阶平稳点。第一个断言直接由下降引理（引理2.1.4）得出。在下一节中，我们将展示如何证明第二个断言。

7.3 *Saddle Points Escaping Lemma*

在这一节中，我们正式证明，如果起始点具有Hessian的严格负特征值，则添加扰动并随后进行梯度下降将在$\mathcal{T}$次迭代中导致函数值显著下降。

引理7.3.1（鞍点逃逸引理）。*Under the setting of Theorem 7.2.1, if $\widetilde{x}$ satisfies $\|\nabla f(\widetilde{x})\| \leq \epsilon$, and $\lambda_{\min}(\nabla^2 f(\widetilde{x})) \leq -\sqrt{\rho\epsilon}$, then let $x_0 = \widetilde{x} + \eta\xi$ ($\xi \sim$ Uniform$(B_0(r))$), and run gradient descent starting from x_0. With probability at least $1-\delta$, we have*

$$f(x_{\mathcal{T}}) - f(\widetilde{x}) \leq -\widetilde{\Omega}(\mathcal{T} \cdot \epsilon^2/\ell)$$

where $x_{\mathcal{T}}$ is the $\mathcal{T}^{th}$ gradient descent iterate starting from x_0.

回忆 $\mathcal{T} = \widetilde{\Theta}(\ell/\sqrt{\rho\epsilon})$。这表明鞍点逃逸时间和函数减少的量仅依赖于维度 d 的多项式对数。为了证明这个引理，我们将展示以下内容:

- (*Improve or Localize*) 如果梯度下降在一定的迭代次数内持续取得微小进展，那么那些迭代内的所有迭代点都必须卡在一个小的欧几里得球内。
- (*Stuck probability is small around saddle points*) 如果Hessian具有显著的负特征值，那么在随机扰动后，以高概率，梯度下降不会长时间陷入一个小欧几里得球中。

结合上述两个陈述，我们得出结论，第二个陈述中的GD必须在一定次数的迭代后取得显著进展，这证明了引理7.3.1。

7.3.1 Improve or Localize

我们首先证明以下引理，该引理表明，如果函数值在 t 次迭代中下降不多，那么所有迭代 $\{x_\tau\}_{\tau=0}^{t}$ 将保持在 x_0 的小邻域内。

引理7.3.2（改进或局部化）。*Assume function f is ℓ-gradient Lipschitz, and run GD with $\eta \le 1/\ell$, then for any $t \ge \tau > 0$, we have:*

$$\|x_\tau - x_0\| \le \sqrt{2\eta t(f(x_0) - f(x_t))}.$$

Proof. 给定梯度更新，$x_{t+1} = x_t - \eta \nabla f(x_t)$，我们有对于任意的 $\tau \le t$ ：

$$\begin{aligned}\|x_\tau - x_0\| \le \sum_{\tau=1}^{t} \|x_\tau - x_{\tau-1}\| &\overset{(1)}{\le} [t \sum_{\tau=1}^{t} \|x_\tau - x_{\tau-1}\|^2]^{\frac{1}{2}} \\ &= [\eta^2 t \sum_{\tau=1}^{t} \|\nabla f(x_{\tau-1})\|^2]^{\frac{1}{2}} \overset{(2)}{\le} \sqrt{2\eta t(f(x_0) - f(x_t))},\end{aligned}$$

在步骤（1）中使用了柯西-施瓦茨不等式，步骤（2）归因于下降引理（引理2.1.4）。 □

引理7.3.2立即意味着，如果 $f(x_{\mathcal{T}}) - f(x_0) \ge -\widetilde{O}(\mathcal{T} \cdot \epsilon^2/\ell)$，即GD在扰动后$\mathcal{T}$步内没有取得足够的进展，那么我们立即有 $\|x_t - x_0\| \le \widetilde{O}(\epsilon\mathcal{T}/\ell)$ 对所有 $t \in [\mathcal{T}]$ 成立。

7.3.2 Bounding the Width of the Stuck Region

其次，我们表明，如果以鞍点附近的点进行随机扰动初始化，GD序列陷入的概率很小。回顾第7.3.1引理，x_0 ~服从$\mathbb{B}_{\tilde{x}}(\eta r)$均匀分布。我们将$\mathbb{B}_{\tilde{x}}(\eta r)$称为*perturbation ball*，并定义扰动球内的*stuck region*为从这些点开始，GD在$\mathcal{T}$步内进展甚微的点集：

$$\begin{aligned}\mathcal{X}_{\text{stuck}} := \{x \in B_{\tilde{x}}(\eta r) \mid \{x_t\}_{t=0}^{\mathcal{T}} &\text{ is a GD sequence with} \\ x_0 = x, &\text{and } \forall t \in [\mathcal{T}], \|x_t - x_0\| \le \widetilde{O}(\epsilon\mathcal{T}/\ell)\}.\end{aligned}$$

查看图7.1以获取说明。由于x_0从该扰动球中均匀采样，GD在扰动后陷入的概率等于陷入区域体积与扰动球体积的比率。因此，我们想要证明陷入区域体积很小。

通常，卡住区域的形状可能非常复杂，因此直接计算其体积非常困难。这里的一个关键观察是，尽管我们不知道卡住区域的形状

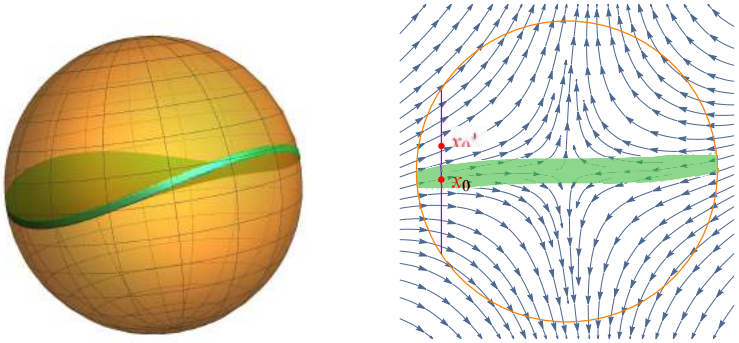


图7.1：左：3D中的扰动球和“薄饼”形状的卡住区域。右：在梯度流下的2D扰动球和“窄带”卡住区域

区域，我们可以证明$\mathcal{X}_{\text{stuck}}$沿$\nabla^2 f(\tilde{x})$的最小特征值方向的宽度很小。事实上，如果宽度不超过$\eta\omega$，那么我们有$\text{Vol}(\mathcal{X}_{\text{stuck}}) \le \text{Vol}(\mathbb{B}_0^{d-1}(\eta r))\eta\omega$，因此，

$$\begin{aligned}\Pr(x_0 \in \mathcal{X}_{\text{stuck}}) = \frac{\text{Vol}(\mathcal{X}_{\text{stuck}})}{\text{Vol}(\mathbb{B}_{\tilde{x}}^d(\eta r))} \le & \frac{\eta\omega \times \text{Vol}(\mathbb{B}_0^{d-1}(\eta r))}{\text{Vol}(\mathbb{B}_0^d(\eta r))} \\ = & \frac{\omega}{r\sqrt{\pi}} \frac{\Gamma(\frac{d}{2}+1)}{\Gamma(\frac{d}{2}+\frac{1}{2})} \le \frac{\omega}{r} \cdot \sqrt{\frac{d}{\pi}}\end{aligned}$$

要实现最大失败概率不超过 δ，我们希望 $\omega \le O(\delta r/\sqrt{d})$。我们通过耦合序列的新技术对卡住区域的宽度 $\mathcal{X}_{\text{stuck}}$ 进行限制——考虑两个 GD 序列 $\{x_t\}_{t=0}^{\mathcal{T}}$，$\{x_t'\}_{t=0}^{\mathcal{T}}$，它们满足：(1) $\max\{\|x_0 - \tilde{x}\|, \|x_0' - \tilde{x}\|\} \le \eta r$；以及(2) $x_0 - x_0' = \eta\omega e_1$，其中 e_1 是 $\nabla^2 f(\tilde{x})$ 的最小特征向量，$\omega \ge \omega_0$ 对于某个阈值 ω_0。

引理7.3.3. *For any $\omega_0 \in (0, \epsilon]$, under the setting of Lemma 7.3.1, if $\{x_t\}_{t=0}^{\mathcal{T}}$, $\{x_t'\}_{t=0}^{\mathcal{T}}$ are coupling sequences as specified above, then for $\mathcal{T} \ge \Omega(\kappa \cdot \log(\epsilon\kappa/\omega_0))$ where $\kappa := \ell/\sqrt{\rho\epsilon}$, we have,)*

$$\exists t \in [\mathcal{T}], \quad \max\{\|x_t - x_0\|, \|x_t' - x_0'\|\} \ge \tilde{\Omega}(\epsilon\mathcal{T}/\ell)$$

引理7.3.3声称，对于任何一对差值在e_1方向上，长度大于或等于$\eta\omega_0$的x_0和x_0'，至少有一个x_0或x_0'在$\mathcal{X}_{\text{stuck}}$之外。这直接意味着$\mathcal{X}_{\text{stuck}}$在$e_1$方向上的宽度是$\eta\omega_0$。引理7.3.3进一步声称，通过仅支付$\mathcal{T}$选择的对数因子，宽度$\eta\omega_0$可以被任意缩小。

Proof. 我们通过反证法证明。假设引理7.3.3的否定为真——$\max\{\|x_t - x_0\|, \|x_t' - x_0'\|\} \le \tilde{O}(\epsilon\mathcal{T}/\ell)$ 对于所有 $t \in [\mathcal{T}]$成立，即GD序列在 $\mathcal{T}$ 步内都卡在一个小的欧几里得球内。

我们可以写出该差分的更新方程式

一对序列 $\hat{\mathrm{x}}_t := \mathrm{x}_t - \mathrm{x}'_t$ 如下：

$$\begin{aligned}\hat{\mathrm{x}}_{t+1} =& \hat{\mathrm{x}}_t - \eta[\nabla f(\mathrm{x}_t) - \nabla f(\mathrm{x}'_t)] = (\mathrm{I} - \eta\mathcal{H})\hat{\mathrm{x}}_t - \eta\Delta_t\hat{\mathrm{x}}_t \\ =& \underbrace{(\mathrm{I} - \eta\mathcal{H})^{t+1}\hat{\mathrm{x}}_0}_{\mathrm{p}(t+1)} - \underbrace{\eta\sum_{\tau=0}^{t}(\mathrm{I} - \eta\mathcal{H})^{t-\tau}\Delta_\tau\hat{\mathrm{x}}_\tau}_{\mathrm{q}(t+1)},\end{aligned}$$

在 $\mathcal{H} = \nabla^2 f(\tilde{\mathrm{x}})$ 和 $\Delta_t = \int_0^1 [\nabla^2 f(\mathrm{x}'_t + \theta(\mathrm{x}_t - \mathrm{x}'_t) - \mathcal{H}]\mathrm{d}\theta$。我们注意到，如果函数 f 在 $\tilde{\mathrm{x}}$ 附近是二次的，则项 $\mathrm{p}(t))$ 是 $\hat{\mathrm{x}}_t$ 的公式，而 $\mathrm{q}(t$) 是由函数 f 非二次引起的近似误差项。

我们将后来证明二次项是主导项，即在所有 $t \in [\mathcal{T}]$ 的意义上，$\|\mathrm{q}($ $t)\| \leq \|\mathrm{p}(t)\|/2$。鉴于这是真的，因为 $\hat{\mathrm{x}}_0 = \eta\omega\mathrm{e}_1$，所以我们有

$$\|\mathrm{p}(t)\| \geq (1 + \sqrt{\epsilon\rho}/\ell)^t \cdot \eta\omega_0$$

此术语呈指数增长。因此，通过选择 $\mathcal{T} \geq \Omega(\kappa \cdot \log(\epsilon\kappa/\omega_0))$，我们得到 $\|\hat{\mathrm{x}}_t\| \geq \|p(t)\|/2 \geq \widetilde{\Omega}(\epsilon\mathcal{T}/\ell)$，这与假设GD序列在半径为$\widetilde{O}(\epsilon\mathcal{T}/$ $\ell)$ 的小欧几里得球中停留 $\mathcal{T}$ 步相矛盾（我们注意到 $\|\mathrm{x}_0 - \mathrm{x}'_0\| \leq 2$ $\eta r \ll \widetilde{O}(\epsilon\mathcal{T}/\ell)$）。这证明了引理7.3.3。

对于证明的剩余部分，我们只需要通过归纳法验证 $\|\mathrm{q}(t)\| \leq \|\mathrm{p}(t)$ $\|/2$ 对所有 $t \in [\mathcal{T}]$ 成立。对于基本情况 $t = 0$，该命题成立，因为 $\|$ $\mathrm{q}(0)\| = 0 \leq \|\hat{\mathrm{x}}_0\|/2 = \|\mathrm{p}(0)\|/2$。现在假设归纳命题对 t 成立。记 $\lambda_{\min}(\mathcal{H}) = -\gamma$。注意 $\hat{\mathrm{x}}_0$ 位于 $\mathcal{H}$ 的最小特征向量的方向上。因此，对于任何 $\tau \leq t$，我们有：

$$\|\hat{\mathrm{x}}_\tau\| \leq \|\mathrm{p}(\tau)\| + \|\mathrm{q}(\tau)\| \leq 2\|\mathrm{p}(\tau)\| = 2(1 + \eta\gamma)^\tau\eta\omega.$$

通过Hessian Lipschitz性质，我们进一步有

$$\|\Delta_t\| \leq \rho\max\{\|\mathrm{x}_t - \tilde{\mathrm{x}}\|, \|\mathrm{x}'_t - \tilde{\mathrm{x}}\|\} \leq \widetilde{O}(\rho\epsilon\mathcal{T}/\ell) = \widetilde{O}(\sqrt{\rho\epsilon})$$

因此：

$$\begin{aligned}&\|\mathrm{q}(t+1)\| = \left\|\eta\sum_{\tau=0}^{t}(\mathrm{I} - \eta\mathcal{H})^{t-\tau}\Delta_\tau\hat{\mathrm{x}}_\tau\right\| \\ \leq&\eta\sum_{\tau=0}^{t}\|\Delta_t\|\left\|(\mathrm{I} - \eta\mathcal{H})^{t-\tau}\right\|\|\hat{\mathrm{x}}_\tau\| \leq \widetilde{O}(\eta\sqrt{\rho\epsilon})\sum_{\tau=0}^{t}(1 + \eta\gamma)^t\eta\omega \\ \leq&\widetilde{O}(1)(1 + \eta\gamma)^t\eta\omega \leq \widetilde{O}(1)\|\mathrm{p}(t+1)\|,\end{aligned}$$

在第二个不等式中使用了 $t + 1 \leq \mathcal{T}$，以及$\widetilde{O}(\eta\mathcal{T}\sqrt{\rho\epsilon}) = \widetilde{O}(1)$。最后，通过仔细处理常数和对数因子，我们实际上可以使这个$\widetilde{O}(1)$ 项小于或等于 1/2（此处省略细节）。这完成了归纳证明。

□

8
Algorithmic Regularization

大规模实际应用的神经网络高度过参数化，与训练样本数量相比，具有更多的可训练模型参数。因此，学习这种高容量模型的优化目标有许多适合训练数据的全局最小值。然而，使用特定的优化算法最小化训练损失不仅将我们带到任何全局最小值，而是某些特殊的全局最小值——从这个意义上说，优化算法的选择在学习过程中引入了一种隐式的归纳偏差，这有助于泛化。

在过参数化的模型中，特别是在深度神经网络中，所学习模型的大部分（如果不是全部）归纳偏差都来自优化算法的这种隐式正则化。例如，关于这个主题的早期实证工作（参考[NTS15a, NSS15, HS97, KMN+16, ZBH+16a, CCS+16, DPBB17, ADG+16, Ney17, WRS+17, HHS17, Smi18]）表明，深度模型即使在仅通过最小化训练误差进行训练而没有任何显式正则化的情况下也能很好地泛化，甚至在网络高度过参数化到能够拟合随机标签的程度时也是如此。因此，存在许多零训练误差解，它们都是训练目标的全局最小值，其中大部分泛化效果极差。尽管如此，我们选择的优化算法，通常是梯度下降的变体，似乎更喜欢那些确实能很好地泛化的解。这种泛化能力不能通过显式指定的模型类（即在所选架构中可表示的函数）的容量来解释。相反，优化算法倾向于“简单”模型，最小化某些隐式“正则化度量”，例如$R(\mathrm{w})$，这对于泛化至关重要。因此，通过表征$R(\mathrm{w})$来理解隐式归纳偏差*e.g.*对于理解模型如何以及学习什么至关重要。例如，在线性回归中，可以证明使用梯度下降最小化一个欠定模型（参数多于样本）会产生最小ℓ_2范数

解决方案（见命题8.1.1），对于在可分数据上训练的线性逻辑回归，梯度下降收敛到硬间隔支持向量机解决方案的方向（定理8.3.2），即使优化问题中未明确指定范数或间隔。实际上，这种分析表明，从优化算法中隐含的归纳偏差导致泛化并非新现象。在提升算法的背景下，（作者？）[EHJT04]和（作者？）[Tel13]分别建立了梯度提升算法（坐标下降）与ℓ_1范数最小化、ℓ_1间隔最大化的联系。观察到最小化。这种最小范数或最大间隔的解决方案当然在所有适合训练数据的解决方案或分离器中非常特殊，并且特别可以确保泛化[BM03, KST09]。

在这一章中，我们主要介绍了在回归和分类问题中，针对各种简单和复杂模型类别的未正则化训练损失最小化时， vanilla梯度下降的算法正则化结果。我们简要讨论了最速下降和镜像下降等通用算法族。

Meanings of "implicit regularization due to training algorithm."

当前章节的结果往往表明，通过在 *Objective 1* 上应用训练算法 *A* 并使其基本收敛（例如，收敛到梯度下降的驻点），所获得的解也满足某些其他 *Objective 2* 的 KKT 局部最优性条件。在许多结果中，目标 2 只是目标 1 加上一个正则化项，通常涉及解的某个范数。因此，我们可以将训练算法视为 *implicitly regularizing* 目标。

虽然这些结果对训练的影响提供了很好的洞察，但仍有一些注意事项，尤其是如果我们寻求深度学习的启示。首先，尽管找到的解决方案恰好是目标2的KKT点，但在实际上对目标2进行标准训练时，它可能永远不会（或几乎永远不会）被观察到。其次，在

本章通常被表述为进行无限时间的训练，这也可能限制它们在实际生活中的适用性。

1 回想一下，在非凸景观中，训练结束时获得的解受到初始化的极大影响，而在深度学习中，初始化非常特殊。

在后续章节中，我们将看到不同类型的分析，该分析分析了解决方案在训练过程中随时间演变的轨迹。这种 *dynamic* 训练视图很快就会变得复杂（与理解驻点时采用的更静态视图相反），并且尚未在现实中的深度网络中实现。

8.1 Linear models in regression: squared loss

suriya: 请查看第3章并根据需要修改说明。我们首先在一个简单的

线性回归设置，其中预测函数由输入的线性函数指定：$f_w(x) = w^\top x$，我们具有以下经验风险最小化目标。

$$L(w) = \sum_{i=1}^{n} \left(w^\top x^{(i)} - y^{(i)} \right)^2. \tag{8.1}$$

这样的简单模式是构建扩展到复杂模型的分析工具的自然起点，这样的结果为理解和改进神经网络中的经验实践提供了直觉。尽管本节的结果是针对平方损失指定的，但结果和证明技术适用于任何光滑损失，具有独特的有限根：在预测 $\widehat{y}$ 和标签 y 之间，$\ell(\widehat{y}、y)$ 在 $\widehat{y}$ 的唯一有限值处最小化 [GLSS18a]。

我们对$n < d$和观测可实现的情况特别感兴趣，即$\min_w L(w) = 0$。在这些条件下，方程（8.1）中的优化问题是不确定的，并且有多个全局最小值，用$\mathcal{G} = \{w\colon\ \forall i,\ w^\top x^{(i)} = y^{(i)}\}$表示。在本文及所有后续问题中，我们的目标是回答：
Which specific global minima do different optimization algorithms reach when minimizing $L(w)$?

以下命题是算法正则化现象的最简单说明。

命题8.1.1. *Consider gradient descent updates w_t for the loss in eq.* (8.1) *starting with initialization w_0. For any step size schedule that minimizes the loss $L(w)$, the algorithm returns a special global minimizer that implicitly also minimzes the Euclidean distance to the initialization: $w_t \to \underset{w \in \mathcal{G}}{argmin} \|w - w_0\|_2$.*

Proof. 关键思想在于注意到损失函数的梯度具有特殊结构。对于方程（8.1）中的线性回归损失 $\forall w$，$\nabla L(w) = \sum_i (w^\top x^{(i)} - y^{(i)}) x^{(i)} \in \text{span}(\{x^{(i)}\})$ —— 即梯度被限制在一个与 w 无关的 n 维子空间中。因此，从初始化 $w_t - w_0 = \sum_{t'<t} \eta w_{t'}$ 开始的梯度下降更新，线性累积梯度，再次被限制在 n 维子空间中。现在很容易检查，存在一个唯一的全局最小值，它既适合数据 ($w \in \mathcal{G}$)，又可以通过梯度下降 ($w \in w_0 + \text{span}(\{x^{(i)}\})$) 达到。通过检查KKT条件，可以验证这个唯一的最小值由 $\text{argmin}_{w \in \mathcal{G}} \|w - w_0\|_2^2$ 给出。

□

在一般情况下，过参数化优化问题中，对隐含偏差或算法正则化的描述通常并不如此优雅或简单。对于同一模型类，改变算法或改变相关超参数（如步长）

大小和初始化），甚至改变模型类的特定参数化也可以改变隐含偏差。例如，（作者？）[WRS+17]表明，对于某些标准的深度学习架构，具有不同动量和自适应梯度更新（AdaGrad和Adam）的SGD算法变体表现出不同的偏差，因此具有不同的泛化性能；（作者？）[KMN+16]，（作者？）[HHS17]和（作者？）[Smi18]研究了SGD中使用的迷你批大小如何影响泛化；（作者？）[NSS15]比较了path-SGD（相对于尺度不变路径范数的最速下降）的偏差与标准SGD。

全面理解所有算法选择如何影响隐含偏见超出了本章的范围（以及当前的研究状态）。然而，在本章的背景下，我们特别想强调由优化算法和特定参数化引起的*geometry*的作用，以下将简要讨论。

8.1.1 Geometry induced by updates of local search algorithms

梯度下降与最小化到初始化的欧几里得距离的隐含偏差之间的关系暗示了算法正则化与局部搜索方法中更新几何之间的联系。特别是，梯度下降迭代可以用以下方程来替代，其中 $t+$ 的第 1 次迭代是通过最小化损失的局部（一阶泰勒）近似，同时约束欧几里得范数中的步长来得到的。

$$w_{t+1} = \operatorname*{argmin}_{w} \langle w, \nabla L(w_t) \rangle + \frac{1}{2\eta} \|w - w_t\|_2^2 . \tag{8.2}$$

受上述联系启发，我们可以研究其他在非欧几里得几何下工作的算法族。两个方便的族是关于势 R 的镜像下降 [BT03, NY83] 和关于一般范数的最速下降 [BV04]。

Mirror descent w.r.t. potential R　镜下降更新定义为任何强凸且可微的势 R，如下所示

$$\begin{aligned} w_{t+1} &= \arg\min_{w} \eta \langle w, \nabla L(w_t) \rangle + D_R(w, w_t), \\ \implies \nabla R(w_{t+1}) &= \nabla R(w_t) - \eta \nabla L(w_t) \end{aligned} \tag{8.3}$$

在 $D_R(w、w') = R(w) - R(w') - \langle \nabla R(w')、w - w' \rangle$ 是关于 *Bregman divergence* [Bre67] 的 R。这个家族捕捉了通过 Bregman 距离 D_R 指定的几何形状的更新。镜面下降的势函数 R 的例子包括平方 ℓ_2 范数

$R(w) = 1/\{v^*\}$，这导致梯度下降；熵势 $R(w) = \sum_i w[i] \log w[i] - w[i]$; 矩阵值 w 的谱熵，其中 $R(w)$ 是 w 的奇异值上的熵势；任何正定矩阵 D 的一般二次势 $R(w) = 1/2\|w\|_D^2 = 1/2\, w^\top D w$；以及 $p \in (1, 2$ 的平方 ℓ_p 范数。

从等式（8.3)中，我们看到，而不是w_t (称为原始迭代)，而是 $\nabla R(w_t)$ (称为对偶迭代)被限制在低维数据流形$\nabla R(w_0) + \text{span}(\{x^{(i)}\})$上。现在可以将梯度下降的论点推广，得到以下结果。

定理8.1.2. *For any realizable dataset $\{x^{(i)}, y^{(i)}\}_{n=1}^N$, and any strongly convex potential R, consider the mirror descent iterates w_t from eq.* (8.3) *for minimizing the empirical loss $L(w)$ in eq.* (8.1)*. For any initializations w_0 and any step-size schedule , if w_t converges to some zero-loss solution w^*, then it holds that*

$$w^* = \underset{w:\forall i, w^\top x^{(i)} = y^{(i)}}{\arg\min} D_R(w, w_0). \tag{8.4}$$

特别地，如果我们从 $w_0 =$ 开始，使得 $_w R(w)$ (最小化，那么 $\nabla R(w_0) = 0$)，然后我们得到 $\arg\min_{w \in \mathcal{G}} R(w)$。2

2 定理8.1.2和命题8.1.1的分析也适用于用实例级随机梯度代替$\nabla L(w_t)$的情况。

Proof of Theorem 8.1.2. 从方程（8.3）中，我们得到 $\nabla R(w_t) = \nabla R(w_0) + \sum_{i=1}^n \lambda_i x_i$ 对于某些 λ_1，...，$\lambda_n \in \mathbb{R}$。因此 w^* 是损失函数 $w \mapsto D_R(w, w_0) - \sum_{i=1}^n \lambda_i x_i$ 的驻点。由于这个函数是凸函数，w^* 是全局最小值，因此是所有插值解中损失函数的约束全局最小值。

□

Steepest descent w.r.t. general norms 梯度下降也是关于通用范数 $\|.\|$ 的最速下降（SD）的特殊情况 [BV04]，其更新由以下公式给出，

$$w_{t+1} = w_t + \eta_t \Delta w_t, \text{ where } \Delta w_t = \arg\min_v \langle \nabla L(w_t), v \rangle + \frac{1}{2}\|v\|^2. \tag{8.5}$$

斜率下降的例子包括梯度下降，它是相对于 ℓ_2 范数的斜率下降，以及坐标下降，它是相对于 ℓ_1 范数的斜率下降。一般来说，方程（8.5）中的更新 Δw_t 并没有唯一定义，可能存在多个使方程（8.5）最小化的方向 Δw_t。在这种情况下，方程（8.5）的任何最小化器都是有效的斜率下降更新。

推广梯度下降和镜像下降，我们可能期望最速下降迭代在对应范数下收敛到初始化点最近的解，即arg min{v*}。当等式8.5成立时，这确实是在二次范数{v*}下的情况。

等同于具有 $R(w) = 1/2\|w\|_D^2$ 的镜像下降。不幸的是，正如以下结果所示，这并不适用于一般范数。

示例1. 在坐标下降的情况下，这是相对于ℓ_1范数的最速下降的特殊情况，（作者？）[EHJT04]在梯度提升的背景下研究了这一现象：观察到有时但*not always*，由$\Delta w_{t+1} \in \text{conv}\left\{-\eta_t \frac{\partial L(w_t)}{\partial w}[j_t]\, e_{j_t} : j_t = \underset{j}{\text{argmax}} \left|\frac{\partial L(w_r)}{\partial w}[j]\right|\right\}$ 给出的坐标下降的优化路径与ℓ_1 *regularization path*给出的一致，$\hat{w}(\lambda) = \arg\min_w L(w) + \lambda\|w\|_1$。当更新平均所有最优坐标和步长是无穷小时，具体的坐标下降路径等同于前向阶段选择，即ϵ-boostin g [Fri01]。当ℓ_1正则化路径$\hat{w}(\lambda)$在每个坐标上都是单调的，它就等同于这个阶段选择路径，即坐标下降优化路径（以及相关的LARS路径）[E HJT04]。在这种情况下，当$\lambda \to 0$和$t \to \infty$的极限时，优化和正则化路径都收敛到最小ℓ_1范数解。然而，当正则化路径$\hat{w}(\lambda)$不是单调的，这可能会发生，优化和正则化路径就会发散，前向阶段选择可以收敛到具有次优ℓ_1范数的解。

示例 2. 以下示例表明，即使在 ℓ_p 范数中，$\|.\|_p^2$ 是平滑且强凸的，最速下降法返回的全局最小值也取决于步长。考虑使用最速下降法对 $L(w)$ 进行最小化，数据集 $\{(x^{(1)} = [1, 1, 1],\ y^{(1)} = 1),\ (x^{(2)} = [1, 2, 0],\ y^{(2)} = 10)\}$。关于 $\ell_{4/3}$ 范数的更新。图 8.1 中该问题的经验结果清楚地表明，最速下降法收敛到一个全局最小值，该最小值取决于步长，甚至在 $\eta \to 0$ 的连续步长极限 w_t *does not* 中，$\min_{w \in \mathcal{G}} \|w - w_0\|$ 收敛到预期的解。

总结来说，对于平方损失，我们用势函数和初始化来描述了通用镜像下降算法的隐含偏差。然而，即使在简单的线性回归中，对于具有一般范数的最速下降法，我们也无法得到有用的描述。相比之下，在第8.3.2节中，我们研究了用于分类的类似于逻辑函数的严格单调损失，其中我们*can*得到了最速下降法的描述。

8.1.2 Geometry induced by parameterization of model class

在许多学习问题中，同一个模型类可以用多种方式进行参数化。给定一个参数空间 $\mathbb{R}^d$ 和一个参数化的模型 f_w，该模型将输入 x 映射到输出 $f_w(x)$，我们考虑一个新的参数化方法。

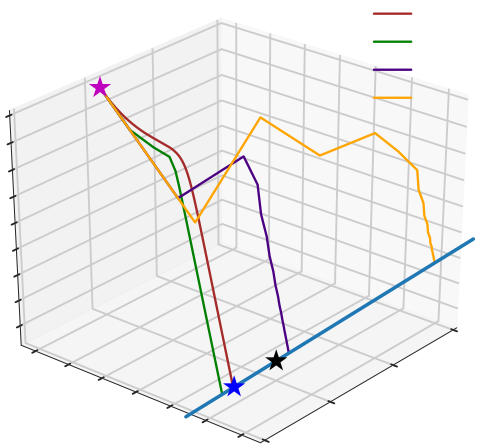


图8.1：关于$\|.\|_{4/3}$的梯度下降：梯度下降收敛到的全局最小值取决于η。在这里，$w_0 = [0, 0, 0]$，$w^*_{\|.\|} = \arg\min_{R\in G} \|w\|_{4/3}$表示最小范数全局最小值，$w^\infty_{\eta\to 0}$表示具有$\eta \to 0$的无限小SD的解。注意，即使$\eta \to 0$，预期的特征也不成立，即$w^\infty_{\eta\to 0} \neq w^*_{\|.\|}$。

参数空间 {v*}，其中 $D \geq d$ 和 a *parametrization*，这是一个从 $\mathbb{R}^D$ 到 $\mathbb{R}^d$ 的满射映射 G：$\theta \mapsto G(\theta)$。参数化 G 诱导了一个新的参数化模型 $\widetilde{f}_\theta(x) \triangleq f_{G(\theta)}(x)$，对所有输入 x。例如，$\mathbb{R}^d$ 中的线性函数可以以规范的方式参数化为 $w \in \mathbb{R}^d$，使用 $f_w(x) = w^\top x$，但也可以等价地通过 $\theta = (u, v)$，其中 $u, v \in \mathbb{R}^d$ 与 $\widetilde{f}_\theta(x) = f_{u\odot v}(x) = (u \odot v)^\top x$ 或 $\widetilde{f}_\theta(x) = f_{u^{\odot 2}-v^{\odot 2}}(x) = (u^{\odot 2} - v^{\odot 2})^\top x$。所有这样的等价参数化都导致等价模型类，然而，在过参数化模型中，使用不同参数化的普通梯度下降会导致原始参数空间 $\mathbb{R}^d$ 中的不同轨迹 $\{G(\theta_t)\}_{t\in\mathbb{N}}$，从而在函数空间中产生不同的诱导偏差。这种现象背后的原因是，尽管普通梯度下降找到相对于 ℓ_2-范数的最陡下降方向，但参数化 G 并不一定保持 ℓ_2 距离，从而扭曲局部下降的几何形状。例如，（作者？）[GWB+17，GLSS18b] 在矩阵分解和线性卷积网络中展示了这一现象，在这些参数化中，这些参数化被证明会引入最小化核范数和 ℓ_p（对于 $p = 2$/深度）范数在傅里叶域中的有趣和异常偏差。一般来说，这些结果暗示了架构选择在不同神经网络模型中的作用，并展示了即使在使用相同的梯度下降算法的情况下，不同的参数化如何诱导函数空间中的不同几何形状。

8.1.3 *Equivalence between geometry inducded by local search algorithms and reparametrization*

在上一节中，我们看到了模型类参数化和优化算法的局部搜索方法都可以诱导出优化景观的不同几何形状。在本节中，我们表明这两种几何形状可以是等价的，也就是说，对于连续的梯度/镜像下降，损失 L、w_t 和 $G(\theta_t)$ 的两个优化轨迹是相同的，其中 x_t 遵循镜像流方程（8.6）

$$\frac{d\nabla R(w_t)}{dt} = -\nabla L(w_t), \tag{8.6}$$

并且 θ_t 遵循重参数化梯度流方程（8.7）

$$\frac{d\theta_t}{dt} = -\nabla(L \circ G)(\theta_t) \tag{8.7}$$

我们强调，尽管优化几何依赖于 L，但本节将展示的主要等价结果是参数化 G、势 R 以及可能的初始化 θ_0、$w_0 = G(\theta_0)$ 的属性。特别是，当等价成立时，它同时对所有可微损失 L 成立。

以下我们展示等价性的直觉。镜像流方程（8.6）可以改写为：

$$\frac{dw_t}{dt} = -(\nabla^2 R(w_t))^{-1}\nabla L(w_t). \tag{8.8}$$

并且重新参数化的梯度流方程（8.7）在 w-空间中得到以下轨迹，$\mathbb{R}^d$：

$$\frac{dG(\theta_t)}{dt} = -\partial G(\theta_t)\partial G(\theta_t)^\top \nabla L(G(\theta_t)). \tag{8.9}$$

因此，为了使两个轨迹 $G(\theta_t)$ 和 w_t 相同，只需要求：

$$\partial G(\theta_t)\partial G(\theta_t)^\top = (\nabla^2 R(G(\theta_t)))^{-1}, \quad \forall t \geq 0. \tag{8.10}$$

如果方程（8.10）成立，那么 $G(\theta_t)$ 和 w_t 都满足相同的微分方程，因此它们是相同的。一个更强（但同时也更方便）的充分条件如下：

$$\partial G(\theta)\partial G(\theta)^\top = (\nabla^2 R(G(\theta)))^{-1}, \quad \forall \theta. \tag{8.11}$$

方程（8.11）更容易检查，因为它不需要理解优化轨迹 θ_t。以下是两个等价性成立的例子，因为方程（8.11）得到满足：

示例8.1.3（二次镜映射和线性重参数化）。
The geometry induced by mirror descent with quadratic potential $R(w) = w^\top \Sigma^{-1} w/2$ *is equivalent to the geometry induced by gradient descent with reparametrization* $G(\theta) = \sqrt{\Sigma}\theta$ *for any positive definite matrix* $\Sigma \in \mathbb{R}^{d\times d}$.

示例8.1.4（熵镜像图和二次重参数化）。
The geometry induced by mirror descent with entropy potential $R(w) = {}^1/_4 \sum_i w[i] \log w[i] - w[i]$ *is equivalent to the geometry induced by gradient descent with reparametrization* $G(\theta) = \theta^{\odot 2}$.

然而，等式（8.11）对于等价性的成立并非必要。事实上，存在大量例子，其中等式（8.11）不成立，但等价性仍然成立。这样的例子通常具有非可逆参数化 G，这进一步允许两个不同的参数 θ_1 和 θ_2 在 w -空间中具有值，*i.e.*，$G(\theta_1) = G(\theta_2)$。参见例8.1.5以获取一个具体例子。

示例8.1.5. *Let* $D = 2d$ *and* $\theta = (\theta^+, \theta^-) \in \mathbb{R}^D$. *Consider the following parametrization* $G(\theta) = (\theta^+)^{\odot 2} - (\theta^-)^{\odot 2}$.

To see why Equation (8.11) *fails, it suffices to take* $d = 1$ *and consider* $\theta_1 = (1, 0)$ *and* $\theta_2 = (\sqrt{2}, 1)$. *We can see* $G(\theta_1) = G(\theta_2) = 1$ *but* $4 = \partial G(\theta_1)\partial G(\theta_1)^\top \neq \partial G(\theta_2)\partial G(\theta_2)^\top = 12$.

以下定理定理8.1.6表明，对于在例8.1.5中定义的参数化，对于某种形式的每个初始化θ_0，存在一个依赖于初始化θ_0的镜像映射，满足方程（8.10）。因此，镜像下降与

并且梯度下降适用于这种参数化。

3 镜像映射对初始化的依赖是必要的。否则，如果方程（8.10）对所有不同的初始化都成立，我们又可以再次应用示例8.1.5中的构造，这会产生矛盾。

定理8.1.6（作者? [WGL+20]）。*Under setting for Example 8.1.5, for any* $\alpha > 0$ *and* $\theta_0 = (\alpha 1, -\alpha 1)$[4], *where* $1 \in \mathbb{R}^d$ *is the all-one vector, the gradient flow trajectory* θ_t *of* $L \circ G$ *(Equation* (8.7)*) is equivalent to the mirror flow trajectory* w_t *of* $R_\alpha(w) \triangleq {}^{\alpha^2}/_4 \cdot \sum_{i=1}^n q\left(\frac{w[i]}{\alpha^2}\right)$ *(Equation* (8.6)*), where*

$$q(z) = 2 - \sqrt{4 + z^2} + z \cdot \mathit{arcsinh}\left(\frac{z}{2}\right). \tag{8.12}$$

4 此定理可以推广到任何具有可能更复杂的镜像映射的 $\theta_0 \in \mathbb{R}^{2d}$

这里的高层次证明思路是，我们可以将 $\partial G(\theta_t)\partial G(\theta_t)^\top$ 写成 $G(\theta_t)$ 和 θ_0 的函数，而这两个函数又可以写成相对于 $G(\theta_t)$ 的镜像映射函数 Hessian 的逆。镜像映射依赖于 θ_0。一个关键的观察是时间上的守恒定律，方程（8.13）。

Proof of Theorem 8.1.6. 首先我们注意到，通过链式法则，

$$d(\theta_t^+ \odot \theta_t^-)/dt = 0 \tag{8.13}$$

并且因此 $\theta_t^+ \odot \theta_t^- \equiv \theta_0^+ \odot \theta_0^- = \alpha^2 1$ 随时间保持不变 t。这进一步意味着

$$\partial G(\theta_t)\partial G(\theta_t)^\top = 8 \cdot \frac{(\theta_t^+)^{\odot 2} + (\theta_t^-)^{\odot 2}}{2} \tag{8.14}$$

$$= 8\left(\left(\frac{(\theta_t^+)^{\odot 2} - (\theta_t^-)^{\odot 2}}{2}\right)^{\odot 2} + \mathbf{1}\alpha^4\right)^{1/2} \tag{8.15}$$

$$= 8\left(\left(\frac{G(\theta_t)}{\odot 1/2}\, 2\right)^{\odot 2} + 1\right. \tag{8.16}$$

$$\left.\alpha^4\right) = 4\left((G(\theta_t))^{\odot 2} + 4 \cdot 1\alpha^4\right)^{\odot 1/2}。 \tag{8.17}$$

最后我们注意到，对于任何 i，$j \in [d]$ 和 $w[i]$，$w[j] \in \mathbb{R}$，

$$\frac{\partial R_\alpha(w)}{\partial w[i]} = \frac{\alpha^2 q(w[i]/\alpha^2)}{dw[i]} = \frac{1}{4}\text{arcsinh}\left(\frac{w[i]}{2\alpha^2}\right), \tag{8.18}$$

并且，

$$\frac{\partial^2 R_\alpha(w)}{\partial w[i]\partial w[j]} = \frac{\partial^2 R_\alpha(w)}{\partial w[i]\partial w[i]}\mathbf{1}_{i=j} = \frac{1}{4}\frac{d\text{arcsinh}\left(\frac{w[i]}{2\alpha^2}\right)}{dw[i]}\mathbf{1}_{i=j} \tag{8.19}$$

$$= \frac{1}{4\sqrt{w[i]^2 + 4\alpha^4}}\mathbf{1}_{i=j} \tag{8.20}$$

This 完成了证明，因为现在根据公式（8.9），我们有

$$\frac{dG(\theta_t)}{dt} = -(\nabla^2 R_\alpha(G(\theta_t)))^{-1}\nabla L(G(\theta_t)), \tag{8.21}$$

显示 $G(\theta_t)$ 等于 w_t，因为它们都是方程（8.6）的唯一解。

□

结合上述定理与定理8.1.2，我们得到以下定理：

定理8.1.7（（作者？）[WGL+20]中的定理1）。*Under the setting of Theorem 8.1.6, additionaly assume that*

1. $L(w) = \frac{1}{n}(w^\top x^{(i)} - y^{(i)})^2$ *where* $(x^{(i)}, y^{(i)})_{i=1}^n$ *are training datasets;*
2. *the reparametrized gradient flow Equation* (8.7) *with initial point* $\theta_0 = (\alpha 1, -\alpha 1)$ *converges to a o-loss solution* θ_∞.

Then $w_\infty = G(\theta_\infty)$ *satisfies that*

$$w_\infty = \arg\min_{w \in \mathbb{R}^d} R_\alpha(w) \tag{8.22}$$

$$s.t. L(w) = 0. \tag{8.23}$$

[是否在镜映射中添加一些基本注释？]

8.1.4 *Equivalence Between Commuting Parametrization and Mirror Descent*

在这个小节中，我们关注以下两个问题：

1. 对于什么镜像映射 R，存在一个参数化 G 使得等价性成立？2. 对于什么参数化 G，存在一个镜像映射 R 使得等价性成立?

证明定理8.1.6的方法并没有给我们太多关于如何决定给定参数化G是否存在这样的镜像映射R的见解。它依赖于守恒定律方程（8.13），这似乎是例8.1.5中参数化G的一个特殊性质。

在这个子节中，我们将介绍一个更通用的框架来攻击（作者？）[LWLA22]中提出的上述两个问题。结果，第一个问题的答案是肯定的。当满足参数化G的某些条件时，第二个问题的答案是肯定的，*e.g.*，以下*commuting*条件：

定义8.1.8（交换参数化）。*We say a parametriza- tion* $G : \mathbb{R}^D \to \mathbb{R}^d$ *is* 交换 *iff* $[\nabla G_i, \nabla G_j](\theta) = \nabla^2 G_i(\theta)\nabla G_j(\theta) - \nabla^2 G_j(\theta)\nabla G_i(\theta) = 0$ *for all* $\theta \in \mathbb{R}^D$，i，$j \in [D]$.[5]

5 在这里 $[\cdot, \cdot]$ 表示李括号。形式上，对于任意两个向量场 X，Y，李括号 $[X, Y]$ 定义为 $[X, Y](\theta) = \partial X \cdot Y - \partial Y \cdot X$ 对于任意 θ。

可以轻松验证，8.1.3到8.1.5中的参数化都是交换参数化。（作者？）[LWLA22]还表明，为了诱导与某些镜像映射等价的几何形状，需要稍微放宽的交换参数化概念。

[资源：待定]

8.2 *Matrix factorization*

≪Suriya notes: I would like to include this section here but can also move to a separate chapter. Ideally, summarize our 2017 paper, Tengyu's 2018 paper and Nadav's 2019 paper. May be we can discuss this after Nadav's lecture?≫

8.3 *Linear Models in Classification*

我们现在转向研究具有逻辑或交叉熵类型损失的分类问题。我们专注于二元分类问题，其中 $y^{(i)} \in \{-1, 1\}$。包括逻辑、交叉熵和对数损失在内的许多0-1损失的连续代理是严格单调损失函数 ℓ 的例子，其中隐含偏好的行为在根本上是不同的，隐含偏好的特征也可以这样描述。

我们研究适合训练数据 $\{x^{(i)}, y^{(i)}\}_i$ 的线性决策边界 $f(x) = w^\top x$，其决策规则由 $\widehat{y}(x) = \text{sign}(f(x))$ 给出。在证明的许多实例中，我们还假设不失一般性地 $y^{(i)} = 1$ 对于所有 i，因为对于线性模型，$y^{(i)}$ 的符号可以等价地吸收到 $x^{(i)}$ 中。我们再次考虑形式如式(8.1)的无正则化经验风险最小化目标，但现在具有严格单调损失。当训练数据 $\{x^{(i)}, y^{(i)}\}_n$ 不可线性分离时，经验目标 $L(w)$ 可以有一个有限的全局最小值。然而，如果数据集是线性可分的，即 $\exists w : \forall i,\ y^{(i)} w^\top y^{(i)} > 0$，经验损失 $L(w)$ 再次是不良设定的，并且更重要的是 $L(w)$ 没有任何有限的极小值，即 $L(w) \to 0$ 仅当 $\|w\| \to \infty$。因此，对于任何序列 $\{w_t\}_{t=0}^{\infty}$，如果 $L(w_t) \to 0$，那么 w_t 必然发散到无穷大而不是收敛，因此我们无法谈论 $\lim_{t\to\infty} w_t$。相反，我们考虑极限方向 $\bar{w}_\infty = \lim_{t\to\infty} \frac{w_t}{\|w_t\|}$，当极限存在时。我们将这个极限的存在称为方向收敛。请注意，极限方向完全指定了我们关心的分类器的决策规则。

在本章剩余部分，我们关注以下指数损失 $\ell(u, y) = \exp(-uy)$。然而，我们的渐近结果可以扩展到具有紧密指数尾部的损失函数，包括逻辑和S型损失，类似于（作者？）[SHS17]和（作者？）[Tel13]。

$$L(w) = \sum_{i=1}^{n} \exp(-y^{(i)} w^\top x^{(i)}). \tag{8.24}$$

8.3.1 Gradient Descent

(作者？)[SHS17]表明，对于几乎所有线性可分的数据集，具有 *any initialization and any bounded step-size*的梯度下降在方向上收敛到具有单位ℓ_2范数的最大间隔分离器，即硬间隔支持向量机分类器。

此对隐含偏差的描述与步长以及初始化均无关。我们已看到，与具有唯一有限根的损失函数的梯度下降的隐含偏差存在根本性的差异（第 ?? 节），其中描述依赖于初始化。上述结果作为更一般结果的一部分在定理 8.3.2 中得到了严格的证明。下面是一个更简单的陈述，并附有启发式证明草图，旨在传达此类结果的感觉。

定理8.3.1. *For almost all dataset which is linearly separable, consider gradient descent updates with any initialization w_0 and any step size that minimizes the exponential loss in eq.* (8.24)*, i.e., $L(w_t) \to 0$. The gradient*

descnet iterates then converge in direction to the ℓ_2 max-margin vector, i.e., $\lim_{t\to\infty} \frac{w_t}{\|w_t\|_2} = \frac{\widehat{w}}{\|\widehat{w}\|}$, *where*

$$\widehat{w} = \underset{w}{\operatorname{argmin}} \|w\|^2 \text{ s.t. } \forall i,\ w^\top x^{(i)} y^{(i)} \geq 1. \tag{8.25}$$

不考虑一般性，假设$\forall i$、$y^{(i)} = 1$作为线性模型的符号可以吸收到$x^{(i)}$中。

Proof Sketch 我们首先直观地理解为什么损失指数尾端导致渐近收敛到最大边缘向量：考察当指数损失最小化时梯度下降的渐近状态，正如我们之前所论证的，这需要 $\forall i : w^\top x^{(i)} \to \infty$。假设 $w_t / \|w_t\|_2$ 收敛到某个极限 w_∞，因此我们可以写出 $w_t = g(t) w_\infty + \rho(t)$ 使得 $g(t) \to \infty$，$\forall i$，$w_\infty^\top x^{(i)} > 0$，并且 $\lim_{t\to\infty} \rho(t)/g(t) = 0$。在 w_t 处的梯度由以下公式给出：

$$\begin{aligned} -\nabla \mathcal{L}(w) &= \sum_{i=1}^{n} \exp\left(-w^\top x^{(i)}\right) x^{(i)} \\ &= \sum_{i=1}^{n} \exp\left(-g(t) \mathrm{w}_\infty^\top x^{(i)}\right) \exp\left(-\rho(t)^\top x^{(i)}\right) \mathbf{x}_n \,. \end{aligned} \tag{8.26}$$

随着$g(t) \to \infty$和指数变得更加负，只有那些具有最大（*i.e.*，最负）指数的样本将对梯度做出贡献。这些正是具有最小边界的$\operatorname{argmin}_i w_\infty^\top x^{(i)}$的样本，即所谓的“支持向量”。负梯度的累积，从而是w_t，将随后由支持向量的非负线性组合所主导。这正是SVM问题的KKT条件（方程8.25）。将这些直觉严格化构成了（作者？）[SHS17]中证明的大部分内容，该证明使用了一种与以下章节（第8.3.2节）中不同的证明技术。

8.3.2 Steepest Descent

. 回想一下，梯度下降是关于通用范数 $\|\cdot\|$ 的最速下降（SD）的特殊情况，其更新由公式（8.5）给出。公式（8.5）中 Δw_t 的最优条件要求

$$\langle \Delta w_t, -\nabla L(w_t) \rangle = \|\Delta w_t\|^2 = \|\nabla L(w_t)\|_\star^2, \tag{8.27}$$

$\|x\|_\star = \wedge_{\|y\|\leq 1} x^\top y$ 是 $\|\cdot\|$ 的双范数。最速下降的例子包括梯度下降，它是关于 ℓ_2 范数的最速下降，以及贪婪坐标下降（高斯-索思韦尔选择规则），它是关于 ℓ_1 范数的最速下降。一般来说，方程（8.5）中的更新 Δw_t 并没有唯一定义，并且可能有多个方向 Δw_t 可以使方程（8.5）最小化。在这种情况下，任何

最小化方程（8.5）是一个有效的最速下降更新，并满足方程（8.27）
。

在定理8.3.1的初步结果中，我们证明了指数损失梯度流的极限方向是ℓ_2最大间隔解。在以下定理中，我们展示了这一结果的天然扩展到所有最速下降算法。

定理8.3.2. *For any separable dataset* $\{x_i,\ y_i\}_{i=1}^n$ *and any norm* $\|\cdot\|$*, consider the steepest descent updates from eq.* (8.27) *for minimizing* $L(w)$ *in eq.* (8.24) *with the exponential loss* $\ell(u,\ y) = \exp(-uy)$ *. For all initial- izations* w_0*, and all bounded step-sizes satisfying* $\eta_t \leq \min\{\eta_+, \frac{1}{B^2 L(w_t)}\}$*, where* $B := \max_n \|x_n\|_\star$ *and* $\eta_+ < \infty$ *is any finite number, the iterates* w_t *satisfy the following,*

$$\lim_{t\to\infty} \min_n \frac{y_i \langle w_t, y_i\rangle}{\|w_t\|} = \max_{w:\|w\|\leq 1} \min_n y_i \langle w, x_i\rangle =: \gamma.$$

In particular, if there is a unique maximum-$\|\cdot\|$ *margin solution* $w^* = \arg\max_{\|w\|\leq 1} \min_i y_i \langle w,\ x_i\rangle$*, then the limit direction satisfies* $\lim_{t\to\infty} \frac{w_t}{\|w_t\|} = w^*$*.*

一个定理8.3.2的特殊情况是关于ℓ_1范数的最速下降，正如我们之前所看到的，这对应于贪婪坐标下降。更具体地说，具有精确线搜索的指数损失坐标下降等价于AdaBoost [SF12]，其中每个坐标代表一个“弱学习器”的输出。实际上，最初神秘的提升泛化性质已经通过隐式ℓ_1正则化[SF12]来理解，后来AdaBoost（步长足够小）被证明收敛到最大ℓ_1边缘解的方向，正如定理8.3.2所保证的。事实上，（作者？）[Tel13]将结果推广到更丰富的指数尾损失函数，包括逻辑损失，以及一类非常数的步长规则。有趣的是，具有精确线搜索的坐标下降（AdaBoost）可能导致无限步长，导致迭代收敛到一个不同的方向，而不是最大ℓ_1-边缘方向[RDS04]，因此定理8.3.2中的有界步长规则。

定理8.3.2是关于其他范数下最陡下降结果的推广，我们的证明遵循了（作者？）相同的策略。我们首先证明了（作者？）[SSS10]的对偶结果的一个推广：如果存在一个单位范数的线性分离器，它实现了边缘γ，那么对于所有w，有$\|\nabla L(w)\|_\star \geq \gamma L(w)$。通过使用这个关于梯度对偶范数的下界，我们能够证明损失比迭代范数的增加更快，从而在最大化边缘的方向上建立收敛性。

在本文本剩余部分，我们讨论定理8.3.2的证明。证明分为三个步骤：

1. 梯度主导条件：对于所有范数和任意的 w，$\|\nabla L(w)\|_\star \geq \gamma L(w)$ 2. 最速下降法的优化性质，如损失函数的减少和梯度在双范数下收敛到零。3. 建立相对于 $L(w_t)$ 的增长足够快的收敛性，以证明定理。

命题8.3.3. *Gradient domination condition (Lemma 10 of [GLSS18a]) Let* $\gamma = \max_{\|w\|\leq 1} \min_i y_i x_i^\top w$. *For all* w,

$$\|\nabla L(w)\|_\star \geq \gamma L(w).$$

接下来，我们建立最速下降算法的一些优化性质，包括梯度范数和损失值的收敛性。

命题8.3.4. *(Lemma 11 and 12 of* (作者？ *) [GLSS18a]) Consider the steepest descent iterates* w_t *on* (8.24) *with stepsize* $\eta \leq \frac{1}{B^2 L(w_0)}$, *where* $B = \max_i \|x_i\|_\star$. *The following holds:*

1. $L(w_{t+1}) \leq L(w_t)$.
2. $\sum_{t=0}^{\infty} \|\nabla L(w_t)\|^2 < \infty$ *and hence* $\|\nabla L(w_t)\|_\star \to 0$.
3. $L(w_t) \to 0$ *and hence* $w_t^\top x_i \to \infty$.
4. $\sum_{t=0}^{\infty} \|\nabla L(w_t)\|_\star = \infty$.

给定这两个命题，证明分为两步进行。我们首先证明损失足够快地收敛到零，从而可以降低未归一化边缘的下界 $\min_i w_t^\top x_i$。接下来，我们上界 $\|w_t\|$。通过将第一步中的下界除以第二步中的上界，我们可以降低归一化边缘的下界，从而完成证明。

Proof of Theorem 8.3.2. 第一步：降低未归一化边缘的下限。首先，我们建立损失收敛足够快。定义

$\gamma_t = \|\nabla L(w_t)\|_\star$从泰勒定理,

$$
\begin{aligned}
&L(w_{t+1}) \leq \\
&L(w_t) + \eta_t \langle \nabla L(w_t), \Delta w_t \rangle + \sup_{\beta \in (0,1)} \frac{\eta_t^2}{2} \Delta w_t{}^\top \nabla^2 L\left(w_t + \beta \eta_t \Delta w_t\right) \Delta w_t \\
&\overset{(a)}{\leq} L(w_t) - \eta_t \|\nabla L(w_t)\|_\star^2 + \frac{\eta_t^2 B^2}{2} \sup_{\beta \in (0,1)} L\left(w_t + \beta \eta_t \Delta w_t\right) \|\Delta w_t\|^2 \\
&\overset{(b)}{\leq} L(w_t) - \eta_t \|\nabla L(w_t)\|_\star^2 + \frac{\eta_t^2 B^2}{2} L(w_t) \|\Delta w_t\|^2
\end{aligned}
\tag{8.28}
$$

在(a)中使用 $v^\top \nabla^2 L(w) v \leq L(w) B^2 \|v\|^2$，(b)使用命题8.3.4部分1和凸性来展示 $\sup_{\beta \in (0,1)} L\left(w_t + \beta \eta_t \Delta w_t\right) \leq L(w_t)$。

从方程式8.28，使用$\gamma_t = \|\nabla L(w_t)\|_\star = \|\Delta w_t\|$，我们有

$$
\begin{aligned}
L(w_{t+1}) &\leq L(w_t) - \eta \gamma_t^2 + \frac{\eta^2 B^2 L(w_t) \gamma_t^2}{2} \\
&= L(w_t) \left[1 - \frac{\eta \gamma_t^2}{L(w_t)} + \frac{\eta^2 B^2 \gamma_t^2}{2}\right] \\
&\overset{(a)}{\leq} L(w_t) \exp\left(-\frac{\eta \gamma_t^2}{L(w_t)} + \frac{\eta^2 B^2 \gamma_t^2}{2}\right) \\
&\overset{(b)}{\leq} L(w_0) \exp\left(-\sum_{u \leq t} \frac{\eta_u \gamma_u^2}{L(w_u)} + \sum_{u \leq t} \frac{\eta^2 B^2 \gamma_u^2}{2}\right),
\end{aligned}
\tag{8.29}
$$

我们在使用 $(1 + x) \leq \exp(x)$ 得到 (a)，并通过递归参数得到 (b)。

接下来，我们降低未归一化边界的下界。从公式（8.29）中，我们有,

$$
\begin{aligned}
\max_{n \in [N]} \exp(-\langle w_{t+1}, x_n \rangle) &\leq L(w_{(t+1)}) \\
&\leq L(w_0) \exp\left(-\sum_{u \leq t} \frac{\eta \gamma_u^2}{L(w_u)} + \sum_{u \leq t} \frac{\eta^2 B^2 \gamma_u^2}{2}\right)
\end{aligned}
\tag{8.30}
$$

通过应用 $-$ 对数,

$$
\min_{n \in [N]} \langle w_{t+1}, x_n \rangle \geq \sum_{u \leq t} \frac{\eta \gamma_u^2}{L(w_u)} - \sum_{u \leq t} \frac{\eta^2 B^2 \gamma_u^2}{2} - \log L(w_0). \tag{8.31}
$$

步骤 2：上界 $\|w_{t+1}\|$。使用 $\|\Delta w_u\| = \|\nabla L(w_u)\|_\star = \gamma_u$，我们有
,

$$
\|w_{t+1}\| \leq \|w_0\| + \sum_{u \leq t} \eta \|\Delta w_u\| \leq \|w_0\| + \sum_{u \leq t} \eta \gamma_u. \tag{8.32}
$$

为了完成证明，我们只需将方程（8.31）和（8.32）结合起来，以降低归一化边界的下界。

$$\frac{\langle w_{t+1}, x_n\rangle}{\|w_{t+1}\|} \geq \frac{\sum_{u\leq t}\frac{\eta\gamma_u^2}{L(w_u)}}{\sum_{u\leq t}\eta\gamma_u + \|w_0\|} - \left(\frac{\sum_{u\leq t}\frac{\eta^2B^2\gamma_u^2}{2} + \log L(w_0)}{\|w_{t+1}\|}\right).$$
$$:= (I) \qquad +(II). \tag{8.33}$$

对于术语（I），根据命题8.3.3，我们有 $\gamma_u = \|\nabla L(w_u)\|_\star \geq \gamma L(w_u)$。因此，分子下界为 $\sum_{u\leq t}\frac{\eta\gamma_u^2}{L(w_u)} \geq \gamma\sum_{u\leq t}\eta\gamma_u$。我们有

$$\frac{\sum_{u\leq t}\frac{\eta\gamma_u^2}{L(w_u)}}{\sum_{u\leq t}\eta\gamma_u + \|w_0\|} \geq \gamma\frac{\sum_{u\leq t}\eta\gamma_u}{\sum_{u\leq t}\eta\gamma_u + \|w_0\|} \to \gamma, \tag{8.34}$$

使用命题8.3.4中的 $\sum_{u\leq t}\eta\gamma_u \to \infty$ 和 $\|w_0\| < \infty$。

对于项（II），使用命题8.3.3对$L(w_0) < \infty$和$\sum_{u\leq t}\frac{\eta^2B^2\gamma_u^2}{2} < \infty$取对数。因此（$II$）$\to$=0。

使用上述内容代入方程（8.33），我们得到

$$\lim_{t\to\infty}\frac{w_{t+1}^\top x_i}{\|w_{t+1}\|} \geq \gamma := \max_{\|w\|\leq 1}\min_i \frac{w^\top x_i}{\|w\|}.$$

□

8.4 *Homogeneous Models with Exponential Tailed Loss*

≪Suriya 注意到：Jason：我认为我们应该在这里给出 Kaifengs 的证明。它更普遍且与当前工作同步.≫ 在本节中，我们考虑当预测函数在参数中是齐次时的梯度下降的渐近行为。考虑损失

$$L(w) = \sum_{i=1}^n \exp(-y_i f_i(w)), \tag{8.35}$$

在 $f_i(cw) = c^\alpha f_i(w)$ 是 α-齐次的条件下。通常，$f_i(w)$ 是预测函数（如深度网络）的输出。类似于第5.5.1节中的线性情况，存在一个相关的最大边缘问题。定义最优边缘为 $\gamma = \max_{\|w\|_2=1}\min_i y_i f_i(w)$。相关的非线性边缘最大化由以下非凸约束优化给出：

$$\min \|w\|^2 \text{ st } y_i f_i(w) \geq \gamma. \tag{Max-Margin}$$

类似于第5.5.1节，我们预计对公式（8.35）的梯度下降会收敛到最大边缘问题（Max-Margin）的最优解。然而，最大边缘问题本身是一个有约束的非凸问题，因此我们无法期望达到全局

最优。相反，我们表明梯度下降迭代收敛到最大间隔问题的第一阶驻点。

定义8.4.1（一阶驻点）。*The first-order optimality conditions of Max-Margin are:*

1. $\forall i,\ y_i f_i(w) \geq \gamma$
2. *There exists Lagrange multipliers* $\lambda \in \mathbb{R}_+^N$ *such that* $w = \sum_n \lambda_n \nabla f_n(w)$ *and* $\lambda_n = 0$ *for* $n \not\in S_m(w) := \{i : y_i f_i(w) = \gamma\}$, *where* $S_m(w)$ *is the set of support vectors .*

We denote by $\mathcal{W}^\star$ *the set of first-order stationary points.*

设 w_t 为梯度流的迭代（步长趋于零的梯度下降）。定义 $\ell_{it} =$ 为 $\exp(-f_i(w_t))$，$\boldsymbol{\ell}_t$ 为具有条目 $\ell_i(t)$ 的向量。以下两个假设假设极限方向 $\frac{w_t}{\|w_t\|}$ 存在，以及损失的极限方向。

$\frac{\boldsymbol{\ell}_t}{\|\boldsymbol{\ell}_t\|}_1$ 存在。这样的假设在最大边缘问题背景下是自然的，因为我们想论证 w_t 收敛到一个最大边缘方向，并且损失 $\boldsymbol{\ell}_t/\|\boldsymbol{\ell}_t\|_1$ 收敛到支持向量的指示向量。尽管这一点在 6 中得到了证明，但我们将直接假设这些极限存在。 6

假设8.4.2（平滑性）。*We assume* $f_i(w)$ *is a* C^2 *function.*

假设8.4.3（渐近公式）。*Assume that* $L(w_t) \to 0$, *that is we converge to a global minimizer. Further assume that* 极限 $_{t\to\infty} \frac{w_t}{\|w_t\|_2}$ *and* 极限 $_{t\to\infty} \frac{\boldsymbol{\ell}_t}{\|\boldsymbol{\ell}_t\|_1}$ *exist. Equivalently,*

$$\ell_{nt} = h_t a_n + h_t \epsilon_{nt} \tag{8.36}$$

$$w_t = g_t \bar{w} + g_t \delta_t, \tag{8.37}$$

with $\|\boldsymbol{a}\|_1 = 1$, $\|\bar{w}\|_2 = 1$, 极限 $_{t\to\infty} h(t) = 0$, 极限 $_{t\to\infty} \epsilon_{nt} = 0$, *and* 极限 $_{t\to\infty} \delta_t t = 0$.

假设8.4.4（线性无关约束充分性）。
Let w *be a unit vector. LICQ holds at* w *if the vectors* $\{\nabla f_i(w)\}_{i\in S_m(w)}$ *are linearly independent.*

约束资格使得定义8.4.1的一阶最优性条件成为最优性的必要条件。没有约束资格，即使是全局最优也可能不满足最优性条件。

例如，在线性SVM中，如果支持向量 x_i 线性无关，则LICQ得到保证。对于从绝对连续分布采样的数据，线性SVM的解将始终具有线性无关的支持向量。

定理8.4.5. *Define* $\bar{w} = \lim_{t\to\infty} \frac{w_t}{\|w_t\|}$. *Under Assumptions 8.4.2, 8.4.3, and 8.4.4,* $\bar{w} \in \mathcal{W}$ *is a first-order stationary point of* (最大边缘).

Proof. 定义 $S = \{i : f_i(\bar{w}) = \gamma\}$，其中 γ 是单位范数 w 可达的最优边界。

引理8.4.6. *Under the setting of Theorem 8.4.5,*

$$\nabla f_i(w_t) = \nabla f_i(g_t\bar{w}) + O(Bg_t^{\alpha-1}\|\delta_t\|). \tag{8.38}$$

For $i \in S$ *, the second term is asymptotically negligible as a function of* t*,*

$$\nabla f_i(w_t) = \nabla f_i(g_t\bar{w}) + o(\nabla f_i(g_t\bar{w})).$$

引理8.4.7. *Under the conditions of Theorem 8.4.5,* $a_i = 0$ *for* $i \notin S$.

从梯度流动力学,

$$\begin{aligned}\dot{w}(t) &= \sum_i \exp(-f_i(w_t))\nabla f_i(w_t) \\ &= \sum_i (h_t a_i + h_t\epsilon_{it})(\nabla f_i(g_t\bar{w}) + \Delta_{it},\end{aligned}$$

在 $\Delta_i(t) = \int_{s=0}^{s=1} \nabla^2 f_i(g_t\bar{w} + sg_t\delta_t)g_t\delta_t ds$。通过展开和使用 $a_i = 0$ 对于 $n \notin S$ (引理 8.4.7) ,

$$\begin{aligned}\dot{w}_t = &\underbrace{\sum_{i\in S} h_t a_i \nabla f_i(g_t w)}_{I} \\ &+ \underbrace{h_t \sum_{i\in S} a_i\Delta_{it}}_{II} + \underbrace{h_t \sum_i \epsilon_{it}\nabla f_i(g_t\bar{w})}_{III} + \underbrace{\sum_i h_t\epsilon_{it}\Delta_{it}}_{IV}\end{aligned}$$

通过假设8.4.4，项 $I = \Omega(g_t^{\alpha-1}h_t)$ 以及从引理8.4.6，$II = o(I)$。使用这些，第一个项 I 是最大的，因此标准化后，

$$\frac{\dot{w}_t}{\|\dot{w}_t\|} = \sum_{i\in S} a_i \nabla f_i(g_t\bar{w}) + o(1). \tag{8.39}$$

由于 \(\lim_t \frac{w_t}{\|w_t\|} =\) \(\lim_t \frac{\dot{w}_t}{\|w_t\|}\) [GLSS18a]，因此

$$\lim_{t\to\infty} \frac{w_t}{\|w_t\|} = \sum_{i\in S} \nabla f_i(g_t\bar{w}). \tag{8.40}$$

因此，我们已经证明 w 满足定义 8.4.1 的第一阶最优性条件。

□

8.5 Induced bias in function space

≪Suriya notes: Jason: can you introduce the idea of induced biases and give special results for linear convnets, any relevant results from yours+tengyu's margin paper, and infinite width 2 layer ReLU network?≫

9
Ultra-wide Neural Networks and Neural Tangent Kernels

本章涉及一个乍一看似乎荒谬的模型：内部层有无限多个节点。为了理解为什么它实际上很有趣，让我们回顾一下我们试图解决的神秘问题。

训练神经网络是一个非凸优化问题，在最坏的情况下，它是NP难的[BR89]。另一方面，从经验上看，像随机梯度下降这样的简单梯度算法通常可以实现零训练损失，即简单算法可以找到一个适合所有训练数据的神经网络。此外，当原始标签被随机标签替换时，对于无意义的数据，人们仍然可以观察到这种现象[ZBH+16b]。

Key role of overparametrization. 网络可以完美地拟合无意义数据的事实并不令人惊讶，因为网络参数过多。例如，Wide ResNet在ImageNet上训练时，参数数量是训练数据点的100倍。回想第三章，在这种条件下，即使是线性回归（通过梯度下降求解）也可以完美地拟合训练数据。但这并不能解释现实生活中的神经网络，因为我们仍然需要证明：(a)从*random initialization* (b)开始，可以通过*gradient-based training*获得低损失；(b)当在适当的数据上训练时，训练好的网络具有良好的泛化能力。许多传统的泛化界限提供的是空洞的保证，如第五章所述。

Teacher/Student Nets. 一个在处理这个研究议程时出现的困难是，显然在某个点上，理论应该考虑数据的属性，而现实生活中的数据（例如图像）没有好的描述。研究人员采取的一条途径是假设训练数据的标签是通过某种方式计算得出的

地真实网络有时被称为 *teacher net*。因此，正在训练的网络被视为一个 *student net*，而良好泛化的目标是能够产生与教师网络广泛一致的标签。

考虑到过参数化在现实生活中的重要性，允许学生网络比教师网络更深或更宽是自然的。[1]

1 确实，在实验中发现，如果通过将输入从分布传递到教师网络来生成合成的标记数据，那么如果允许新网络显著更大，那么从零开始教授新网络来模仿教师网络在实践上要容易得多。

Infinite nets and NTKs: 现在我们解释无限神经网络模型。想法是让内部层的宽度非常大，本质上趋于无穷大。例如，想象一个标准的网络如AlexNet被允许将其全连接层扩展到无限宽度，卷积层具有无限数量的通道。这种极度膨胀的AlexNet架构仍然使用之前相同的输入，但其训练和泛化行为可能从通常版本有很大的变化。研究人员研究了这些架构，并迅速意识到至少有一种初始化/训练方式会导致网络变成一个核分类器，称为*Neural Tangent Kernel* (NTK) [2]。本章是无限宽网络的介绍并且 NTKs。我们将看到 NTKs 的行为可以通过计算 NTK 的核内积的高效算法来计算。NTKs 确实表现出良好的优化（即收敛到低训练损失）和合理的泛化行为，但并不如它们的有限对应物那么好。例如，对应于 AlexNet 的 NTK 在图像数据上泛化表现合理，但准确率远低于 AlexNet。我们将讨论 NTK 在小数据任务中的一些实际应用。

2 亚瑟·雅科特，弗兰克·加布里埃尔，克莱门特·洪勒。神经切线核：神经网络中的收敛性和泛化。*Advances in neural information processing systems*，第 8571–8580 页，2018 年

9.1 *Evolution equation for net parameters*

本节推导了在最小二乘损失下网络在训练过程中的演化。它适用于任何网络，简单的表达式将在NTK理论中发挥关键作用。

我们用 $f(w, x) \in \mathbb{R}$ 表示神经网络的输出，其中 $w \in \mathbb{R}^N$ 是网络中的所有参数，$x \in \mathbb{R}^d$ 是输入。给定训练数据集 $\{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$，考虑通过最小化训练数据上的平方损失来训练神经网络：

$$\ell(w) = \frac{1}{2} \sum_{i=1}^{n} \left(f(w, x_i) - y_i\right)^2 .$$

为了简单起见，在本章中，我们研究梯度流，即以无穷小学习率的梯度下降。在这种情况下，dy-

namics 可以用常微分方程（ODE）描述：

$$\frac{dw(t)}{dt} = -\nabla \ell(w(t)).$$

使用此参数动力学的描述，下一个引理描述了训练数据点的预测动力学的变化。

引理 9.1.1. *Let $u(t) = (f(w(t), x_i))_{i\in[n]} \in \mathbb{R}^n$ be the network outputs on all x_i's at time t, and $y = (y_i)_{i\in[n]}$ be the labels. Then $u(t)$ follows the following evolution, where $H(t)$ is an $n \times n$ positive semidefinite matrix whose (i, j)-th entry is $\left\langle \frac{\partial f(w(t),x_i)}{\partial w}, \frac{\partial f(w(t),x_j)}{\partial w} \right\rangle$:*

$$\frac{du(t)}{dt} = -H(t) \cdot (u(t) - y). \tag{9.1}$$

Proof of Lemma 9.1.1. 参数 w 根据微分方程演变

$$\frac{dw(t)}{dt} = -\nabla \ell(w(t)) = -\sum_{i=1}^{n} (f(w(t), x_i) - y_i) \frac{\partial f(w(t), x_i)}{\partial w}, \tag{9.2}$$

在 $t \geq 0$ 是连续时间索引的情况下。根据方程（9.2），网络输出 $f(w(t), x_i)$ 的演化可以表示为

$$\frac{df(w(t), x_i)}{dt} = -\sum_{j=1}^{n} (f(w(t), x_j) - y_j) \left\langle \frac{\partial f(w(t), x_i)}{\partial w}, \frac{\partial f(w(t), x_j)}{\partial w} \right\rangle. \tag{9.3}$$

自 $u(t) = (f(w(t)$ 以来，$x_i))_{i\in[n]} \in \mathbb{R}^n$ 是在时间 t 所有 x_i 的网络输出，而 $y = (y_i)_{i\in[n]}$ 是期望输出，因此方程（9.3）可以更紧凑地写为

$$\frac{du(t)}{dt} = -H(t) \cdot (u(t) - y), \tag{9.4}$$

在 $H(t) \in \mathbb{R}^{n\times n}$ 是一个定义为 $[H(t)]_{i,j} = \left\langle \frac{\partial f(w(t),x_i)}{\partial w}, \frac{\partial f(w(t),x_j)}{\partial w} \right\rangle$ $(\forall i, j \in [n])$ 的核矩阵，其中 {v*} 保持不变。 □

9.1.1 Behavior in the infinite limit

回忆一下，我们在训练集固定、宽度趋于无穷大以及对于合适的初始化尺度（这取决于宽度）的情况下，对深度网络训练的极限感兴趣。在相当一般的条件下，可以证明矩阵 $H(t)$ 在训练过程中保持粗糙 *con- stant*，即大致等于 $H(0)$。此外，矩阵 $H(0)$，其定义涉及初始化时使用的随机权重，以概率收敛到该

训练数据集（见第3章）针对某些核，称为*Neural Tangent Kernel*。然后方程（9.1）变为

$$\frac{du(t)}{dt} = -H^* \cdot (u(t) - y). \tag{9.5}$$

换句话说，第3章中描述的最小二乘核回归，但具有无限小的学习率。回想一下，最终的分类器描述为

$$f^*(x) = (k(x, x_1), \ldots, k(x, x_n)) \cdot (H^*)^{-1} y. \tag{9.6}$$

9.2 NTK: Simple 2-layer example

本节中，我们以以下形式的简单两层神经网络为背景，发展了该理论：

$$f(a, W, x) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma\left(w_r^\top x\right) \tag{9.7}$$

$\sigma(\cdot)$ 是激活函数。这里我们假设 $|\dot{\sigma}(z)|$ 和 $|\ddot{\sigma}(z)|$ 在所有 $z \in \mathbb{R}$ 上都受限于 1。例如，软加激活函数，$\sigma(z) = \log(1 + \exp(z))$，满足这个假设。[3] 我们还假设输入 x 是一个单位向量，即 $\|x\|_2 = 1$。

缩放$1/\sqrt{m}$将在证明$H(t)$在固定核下保持接近Gram矩阵H^*时发挥重要作用。在整个章节中，为了衡量两个矩阵A和B的接近程度，我们使用算子范数$\|\cdot\|_2$。我们将使用以下事实：如果对应项接近，那么它们的谱性质A、B也接近。这源于坐标间平方差的和$\|A - B\|_F^2$也是$\|A - B\|_2^2$的上界。

3 注意修正线性单元（ReLU）激活函数不满足此假设。然而，可以通过对ReLU的专门分析来证明 $H(t) \approx H^*$ [DZPS18]。

我们使用随机初始化 $w_r(0) \sim N(0, I)$ 和 $a_r \sim \text{Unif}[\{-1, 1\}]$. 为了简单起见，我们只优化第一层，即 $W = [w_1, \ldots, w_m]$. 注意这仍然是一个非凸优化问题。

我们可以首先计算 $H(0)$ 并表示为 $m \to \infty$，$H(0)$ 收敛到一个固定矩阵 H^*。注意 $\frac{\partial f(a,W,x_i)}{\partial w_r} = \frac{1}{\sqrt{m}} a_r x_i \sigma'\left(w_r^\top x_i\right)$。因此，$H(0)$ 的每个元素都满足以下公式

$$\begin{aligned}
[H(0)]_{ij} &= \sum_{r=1}^{m} \left\langle \frac{\partial f(a, W(0), x_i)}{\partial w_r(0)}, \frac{\partial f(a, W(0), x_j)}{\partial w_r(0)} \right\rangle \\
&= \sum_{r=1}^{m} \left\langle \frac{1}{\sqrt{m}} a_r x_i \dot{\sigma}\left(w_r(0)^\top x_i\right), \frac{1}{\sqrt{m}} a_r x_j \sigma'\left(w_r(0)^\top x_i\right) \right\rangle \\
&= x_i^\top x_j \cdot \frac{\sum_{r=1}^{m} \sigma'\left(w_r(0)^\top x_i\right) \sigma'\left(w_r(0)^\top x_j\right)}{m}
\end{aligned}$$

这里我们使用了 $a_r^2 = 1$ 对于所有 $r = 1, ..., m$，因为我们初始化 $a_r \sim \text{Unif}[\{-1, 1\}]$。回忆每个 $w_r(0)$ 都是从中独立同分布采样的

一个标准高斯分布。因此，可以认为 $[H(0)]_{ij}$
as the average of m i.i.d. random variables. 如果 m 很大，那么根据大数定律，我们知道这个平均值接近随机变量的期望。这里的期望是在 x_i 和 x_j 上评估的 NTK：

$$H_{ij}^* \triangleq x_i^\top x_j \cdot \mathop{\mathbb{E}}_{w\sim N(0,I)} \left[\sigma'\left(w^\top x_i\right)\sigma'\left(w^\top x_j\right)\right]$$

问题 9.2.1. *If the activation σ is ReLU then (noting that it is differentiable everywhere except at one point) then show that*

$$H_{ij}^* = \mathbb{E}_{w\sim\mathcal{N}(0,I)}\left[\dot{\sigma}w^\top x\dot{\sigma}w^\top x'\right] = \frac{\pi - \arccos\left(\frac{x^\top x'}{\|x\|_2\|x'\|_2}\right)}{2\pi}. \tag{9.8}$$

使用Hoeffding不等式和并集不等式，可以轻松获得以下界限，该界限描述了m以及$H(0)$和H^*的接近程度。

引理9.2.2（初始化扰动，[DZPS19, SY19]）。*Fix some $\epsilon > 0$. If $m = \Omega\left(\epsilon^{-2}n^2\log\left(n/\delta\right)\right)$, then with probability at least $1-\delta$ over $w_1(0)$, ..., $w_m(0)$, we have*

$$\|H(0) - H^*\|_2 \le \epsilon.$$

Proof of Lemma 9.2.2. 我们首先固定了一个条目（i，j）。注意

$$\left|x_i^\top x_j\sigma'\left(w_t(0)^\top x_i\right)\sigma'\left(w_r(0)^\top x_j\right)\right| \le 1.$$

应用Hoeffding不等式，我们有概率$1-\frac{\delta}{n^2}$，

$$|[H(0)]_{i,j} - H_{i,j}^*| \le \left(\frac{2}{m}\log(2n^2/\delta)\right)^{1/2} \le 4(\frac{\log(n/\delta)}{m})^{1/2} \le \frac{\epsilon}{n}.$$

接下来，对所有对$(i,j)\in[n]\times[n]$应用并集界，对于所有(i,j)，我们有$\left|[H(0)]_{i,j} - H_{i,j}^*\right| \le \frac{\epsilon}{n^2}$。为了建立算子范数界，我们只需使用以下不等式链

$$\begin{aligned}\|H(0) - H^*\|_2 &\le \|H(0) - H^*\|_F \\ &= \left(\sum_{ij}|\,[H(0)]_{i,j} - H_{i,j}^*|^2\right)^{1/2} \\ &\le (n^2\cdot\frac{\epsilon^2}{n^2})^{1/2} = \epsilon.\end{aligned}$$

□

现在我们继续展示在训练过程中，$H(t)$接近$H(0)$。形式上，我们证明以下引理。

引理 9.2.3. *Assume $y_i = O(1)$ for all $i = 1, \ldots, n$. Given $t > 0$, suppose that for all $0 \leq \tau \leq t$, $u_i(\tau) = O(1)$ for all $i = 1, \ldots, n$. If $m = \Omega\left(\frac{n^6 t^2}{\epsilon^2}\right)$, we have*

$$\|H(t) - H(0)\|_2 \leq \epsilon.$$

Proof of Lemma 9.2.3. 第一个关键思想是展示 *every weight vector only moves little if m is large.* 为了展示这一点，让我们计算单个权重向量 w_r 的移动。

$$\begin{aligned}
\|w_r(t) - w_r(0)\|_2 =& \left\| \int_0^t \frac{dw_r(\tau)}{d\tau} d\tau \right\|_2 \\
=& \left\| \int_0^t \frac{1}{\sqrt{m}} \sum_{i=1}^n \left(u_i(\tau) - y_i\right) a_r x_i \dot{\sigma}\left(w_r(\tau)^\top x_i\right) d\tau \right\|_2 \\
\leq& \frac{1}{\sqrt{m}} \int \left\| \sum_{i=1}^n \left(u_i(\tau) - y_i\right) a_r x_i \dot{\sigma}\left(w_r(\tau)^\top x_i\right) \right\|_2 d\tau \\
\leq& \frac{1}{\sqrt{m}} \sum_{i=1}^n \int_0^t \left\| u_i(\tau) - y_i a_r x_i \dot{\sigma}\left(w_r(\tau)^\top x_i\right) \right\|_2 d\tau \\
\leq& \frac{1}{\sqrt{m}} \sum_{i=1}^n \int_0^t O(1) d\tau \\
=& O\left(\frac{tn}{\sqrt{m}}\right).
\end{aligned}$$

此计算表明，在任意给定的时间 t，只要 m 足够大，$w_r(t)$ 就接近 $w_r(0)$。接下来，我们展示这表明核矩阵 $H(t)$ 接近 $H(0)$。我们计算单个

条目。

$$
\begin{aligned}
&[H(t)]_{ij} - [H(0)]_{ij} \\
=&\left| \frac{1}{m} \sum_{r=1}^{m} \left(\dot{\sigma}\left(w_r(t)^\top x_i\right) \dot{\sigma}\left(w_r(t)^\top x_j\right) - \dot{\sigma}\left(w_r(0)^\top x_i\right) \dot{\sigma}\left(w_r(0)^\top x_j\right) \right) \right| \\
\leq& \frac{1}{m} \sum_{r=1}^{m} \left| \dot{\sigma}\left(w_r(t)^\top x_i\right) \left(\dot{\sigma}\left(w_r(t)^\top x_j\right) - \dot{\sigma}\left(w_r(0)^\top x_j\right) \right) \right| \\
&+ \frac{1}{m} \sum_{r=1}^{m} \left| \dot{\sigma}\left(w_r(0)^\top x_j\right) \left(\dot{\sigma}\left(w_r(t)^\top x_j\right) - \dot{\sigma}\left(w_r(0)^\top x_i\right) \right) \right| \\
\leq& \frac{1}{m} \sum_{r=1}^{m} \left| \max_r \dot{\sigma}\left(w_r(t)^\top x_i\right) \|x_i\|_2 \|w_r(t) - w_r(0)\|_2 \right| \\
&+ \frac{1}{m} \sum_{r=1}^{m} \left| \max_r \dot{\sigma}\left(w_r(t)^\top x_i\right) \|x_i\|_2 \|w_r(t) - w_r(0)\|_2 \right| \\
=& \frac{1}{m} \sum_{r=1}^{m} O\left(\frac{tn}{\sqrt{m}}\right) \\
=& O\left(\frac{tn}{\sqrt{m}}\right).
\end{aligned}
$$

因此，使用与引理9.2.2相同的论据，我们有

$$
\|H(t) - H(0)\|_2 \leq \sum_{i,j} \left| [H(t)]_{ij} - [H(0)]_{ij} \right| = O\left(\frac{tn^3}{\sqrt{m}}\right).
$$

将我们对 m 的假设代入，我们完成了证明。 □

一些备注随后。

注意1：假设 $y_i = O(1)$ 是一个温和的假设，因为在实践中，大多数标签都被一个绝对常数所限制。

注意2：对于所有 $\tau \leq t$ 和 m 对 t 的依赖性，以及对 $u_i(\tau) = O(1)$ 的假设可以放宽。这需要更精细的分析。参见 [DZPS19]。

备注3：可以将多层神经网络的证明进行推广。参见[ADH+19b]获取更多细节。

备注4：虽然我们只证明了连续时间极限，但通过小学习率（离散时间）梯度下降，$H(t)$接近H^*，这并不难证明。参见[DZPS19]。

9.3 *Explaining Optimization and Generalization of Ultra-wide Neural Networks via NTK*

现在我们已建立了以下近似

$$
\frac{du(t)}{dt} \approx -H^* \cdot (u(t) - y) \tag{9.9}
$$

在 H^* 是 NTK 矩阵的情况下。现在我们使用这个近似来分析超宽神经网络的优化和泛化行为。

Understanding Optimization 动态的 $u(t)$ 如下

$$\frac{du(t)}{dt} = -H^* \cdot (u(t) - y)$$

实际上是一个线性动力系统。对于这种动力学，有一个标准分析。我们记H^*的特征值分解为

$$H^* = \sum_{i=1}^{n} \lambda_i v_i v_i^\top$$

在 $\lambda_1 \geq ... \geq \lambda_n \geq 0$ 是特征值且 $v_1 ... v_n$ 是特征向量的情况下。通过这种分解，我们考虑 $u(t)$ 在每个特征向量 *separately* 上的动力学。形式上，固定一个特征向量 v_i 并将两边乘以 v_i，我们得到

$$\begin{aligned}\frac{dv_i^\top u(t)}{dt} &= -v_i^\top H^* \cdot (u(t) - y) \\ &= -\lambda_i \left(v_i^\top (u(t) - y)\right).\end{aligned}$$

观察发现，$v_i^\top u(t)$ 的动力学仅依赖于自身和 λ_i，因此这实际上是一个一维常微分方程。此外，这个常微分方程有解析解

$$v_i^\top (u(t) - y) = \exp(-\lambda_i t) \left(v_i^\top (u(0) - y)\right). \tag{9.10}$$

现在我们使用方程（9.10）来解释为什么我们可以找到一个零训练误差解。我们需要假设对于所有 $i = 1, ..., n$，即这个核矩阵的所有特征值都是严格正的，$\lambda_i > 0$。这在相当一般的条件下可以证明。参见 [DZPS19, DLL+18]。

观察$(u(t) - y)$是时间t预测值与训练标签之间的差异，并且算法找到0训练误差解意味着$t \to \infty$，我们有$u(t) - y \to 0$。方程(9.10)表明，这个差异的每个分量，即$v_i^\top (u(t) - y)$，由于$\exp(-\lambda_i t)$项，以指数速度收敛到0。此外，注意$\{v_1, ..., v_n\}$构成了$\mathbb{R}^n$的正交归一基，因此$(u(t) - y) = \sum_{i=1}^n v_i^\top (u(t) - y)$。由于我们知道每个$v_i^\top (u_i(t) - y) \to 0$，我们可以得出结论，$(u(t) - y) \to 0$。

方程（9.10）实际上给我们提供了更多关于收敛的信息。注意每个分量 $v_i^\top (u(t) - y)$ 以不同的速率收敛到0。对应较大 λ_i 的分量比对应较小 λ_i 的分量更快地收敛到0。对于一组标签，为了实现更快的收敛，我们希望投影

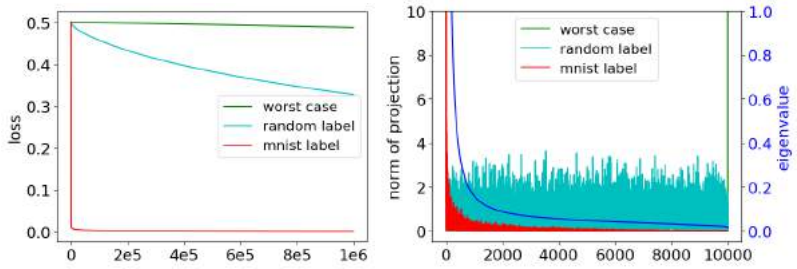


图9.1：收敛速度与投影到核矩阵特征向量的关系。

y映射到顶部特征向量上要大。4 因此，我们得到
以下直观规则以定性方式比较收敛速度（对于固定的 $\|y\|_2$）：

4 这里为了简化，我们忽略了 $u(0)$ 的影响。参见 [ADH+19a] 了解如何减轻对 $u(0)$ 的影响。

- 对于一个标签集 y，如果它们与最大的特征向量对齐，即对于大的 λ_i，$(v_i^\top y)$ 很大，那么梯度下降会快速收敛。
- 对于一个标签集，如果投影到特征向量 $\{(v_i^\top y)\}_{i=1}^n$ 上是均匀的，或者标签与特征向量在较小的特征值上对齐，那么梯度下降收敛速度较慢。

我们可以通过实验验证这一现象。在图9.1中，我们比较了使用原始标签、随机标签和最坏情况标签（对应于λ_n的H^*的归一化特征向量）进行梯度下降时的收敛速度。我们使用方程（9.7）中定义的神经网络架构，并使用ReLU激活函数，仅训练第一层。在右图中，我们绘制了H^*的特征值以及真实、随机和最坏情况标签在不同H^*特征向量上的投影。实验使用来自MNIST两个类别的数据使用梯度下降。这些图表明，原始标签与最高特征向量有更好的对齐，因此收敛速度更快。

9.3.1 *Understanding Generalization in 2-layer setting*

方程（9.9）中的近似意味着超宽神经网络的最终预测函数大约等于方程（9.6）中定义的核预测函数。因此，我们可以仅使用核的泛化理论来分析超宽神经网络的泛化行为。对于方程（9.6）中定义的核预测函数，我们可以使用Rademacher复杂度界来推导以下1-Lipschitz损失函数的泛化界（它是分类误差的上界）：

$$\frac{\sqrt{2y^\top (H^*)^{-1} y \cdot \operatorname{tr}(H^*)}}{n}. \tag{9.11}$$

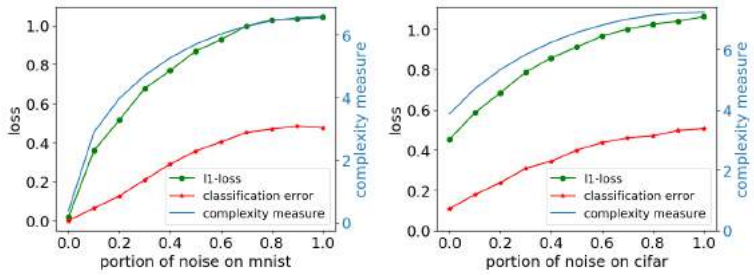


图9.2：泛化误差与复杂度度量。

这是一个 *data-dependent* 复杂度度量，它上界了泛化误差。

我们可以通过经验来检查这个复杂度度量。在图9.2中，我们比较了泛化误差（ℓ_1损失和分类误差）与这个复杂度度量。我们改变数据集中随机标签的部分，以观察泛化误差和复杂度度量如何变化。我们使用方程（9.7）中定义的神经网络架构，并仅训练第一层。左图使用MNIST的两个类别的数据，右图使用CIFAR的两个类别。当随机标签的部分增加时，这个复杂度度量几乎与泛化误差的趋势相匹配。

Learning from simple teacher nets. 现在我们解释为什么NTK可以学习一些可以用两层网络表示的函数。因为我们知道对于任何标签，优化误差都会趋近于0，所以只需要研究教师网络能够使泛化误差保持较小的原因。具体来说，我们给出了一些两层教师网络，使得泛化界限（9.11）趋近于0，即$n \to \infty$。

线性函数：我们从简单的例子开始。假设某个向量 β 的标签为 $y = \beta^\top x$。然后可以证明 (9.11) 被上界 $O\left(\frac{\|\beta\|_2}{\sqrt{n}}\right)$ 限制。因此，NTK 可以学习具有有界系数的线性函数。

两层具有多项式激活的网：考虑$y = \sum_{j=1}^k \alpha_j \left(\beta_j^\top x\right)^p$，即一个激活函数为$z^p$且$p$为偶数的两层网络。类似于线性函数，可以证明(9.11)的上界为 $O\left(\frac{p\sum_{j=1}^k |\alpha_j| \|\beta_j\|_2^p}{\sqrt{n}}\right)$。因此，我们可以论证NTK可以学习具有有界系数的两层多项式网络。

余弦激活除了多项式之外，还可以证明NTK可以学习一些奇特的功能。例如，如果$y = \sum_{j=1}^k \alpha_j \left(\cos\left(\beta_j^\top x_2\right) - 1\right)$，那么我们可以将(9.11)的界限定义为 $O\left(\frac{p\sum_{j=1}^k |\alpha_j| \|\beta_j\|_2 \sinh(\|\beta\|_2)}{\sqrt{n}}\right)$。

所有这些例子都可以通过一种基于一般技术的方法来证明

on Taylor expansion of NTK. See [ADH+19a].

9.4 NTK formula for Multilayer Fully-connected Neural Network

在这个部分，我们展示了全连接神经网络的NTK公式。我们首先正式定义一个全连接神经网络。令$x \in \mathbb{R}^d$为输入，并为了方便表示，用$g^{(0)}(x) = x$和$d_0 = d$表示。我们递归地定义一个具有L隐藏层的全连接神经网络，对于$h = 1, 2, \ldots, L$:

$$f^{(h)}(x) = W^{(h)} g^{(h-1)}(x) \in \mathbb{R}^{d_h}, g^{(h)}(x) = \sqrt{\frac{c_\sigma}{d_h}} \sigma\left(f^{(h)}(x)\right) \in \mathbb{R}^{d_h} \tag{9.12}$$

$W^{(h)} \in \mathbb{R}^{d_h \times d_{h-1}}$ 是第 h 层的权重矩阵 ($h \in [L]$)，$\sigma : \mathbb{R} \to \mathbb{R}$ 是逐坐标激活函数，$c_\sigma = \left(\mathbb{E}_{z \sim \mathcal{N}(0,1)}\left[\sigma z^2\right]\right)^{-1}$。神经网络的最后一层是

$$\begin{aligned} f(w, x) = f^{(L+1)}(x) &= W^{(L+1)} \cdot g^{(L)}(x) \\ &= W^{(L+1)} \cdot \sqrt{\frac{c_\sigma}{d_L}} \sigma W^{(L)} \cdot \sqrt{\frac{c_\sigma}{d_{L-1}}} \sigma W^{(L-1)} \cdots \sqrt{\frac{c_\sigma}{d_1}} \sigma W^{(1)} x, \end{aligned}$$

$W^{(L+1)} \in \mathbb{R}^{1 \times d_L}$是最终层的权重，而$w = \left(W^{(1)}, \ldots, W^{(L+1)}\right)$表示网络中的所有参数。

我们初始化所有权重为独立同分布的 $\mathcal{N}(0, 1)$ 随机变量[5]，并考虑大隐藏宽度极限：$d_1, d_2, \ldots, d_L \to \infty$。方程（9.12）中的缩放因子 $\sqrt{c_\sigma / d_h}$确保每个 $h \in [L]$ 的 $g^{(h)}(x)$ 范数在初始化时大致保持不变（参见 [DLL+18]）。特别是对于 ReLU 激活，我们有

$$\mathbb{E}\left[\left\|g^{(h)}(x)\right\|_2^2\right] = \|x\|_2^2 \ (\forall h \in [L]).$$

从引理9.1.1中回忆，我们需要计算在随机初始化下，$\left\langle \frac{\partial f(w,x)}{\partial w}, \frac{\partial f(w,x')}{\partial w} \right\rangle$在无限宽度极限下收敛到的值。我们可以将关于特定权重矩阵$W^{(h)}$的偏导数写成紧凑形式:

$$\frac{\partial f(w, x)}{\partial W^{(h)}} = b^{(h)}(x) \cdot \left(g^{(h-1)}(x)\right)^\top, \qquad h = 1, 2, \ldots, L+1,$$

哪里

$$b^{(h)}(x) = \begin{cases} 1 \in \mathbb{R}, & h = L+1, \\ \sqrt{\frac{c_\sigma}{d_h}} D^{(h)}(x) \left(W^{(h+1)}\right)^\top b^{(h+1)}(x) \in \mathbb{R}^{d_h}, & h = 1, \ldots, L, \end{cases} \tag{9.13}$$

$$D^{(h)}(x) = \operatorname{diag}\left(\dot{\sigma}\left(f^{(h)}(x)\right)\right) \in \mathbb{R}^{d_h \times d_h}, \qquad h = 1, \ldots, L. \tag{9.14}$$

5 此缩放是NTK行为的关键。以远小的方差初始化会导致非常不同的行为。

然后，对于任意两个输入 x 和 x'，以及任意 $h \in [L+1]$，我们可以计算

$$\begin{aligned}
&\left\langle \frac{\partial f(w,x)}{\partial W^{(h)}}, \frac{\partial f(w,x')}{\partial W^{(h)}} \right\rangle \\
=&\left\langle b^{(h)}(x) \cdot \left(g^{(h-1)}(x)\right)^\top, b^{(h)}(x') \cdot \left(g^{(h-1)}(x')\right)^\top \right\rangle \\
=&\left\langle g^{(h-1)}(x), g^{(h-1)}(x') \right\rangle \cdot \left\langle b^{(h)}(x), b^{(h)}(x') \right\rangle .
\end{aligned}$$

注意第一个项 $\left\langle g^{(h-1)}(x),\ g^{(h-1)}(x') \right\rangle$ 是 x 和 x' 在第 h 层的协方差。当宽度趋于无穷大时，$\left\langle g^{(h-1)}(x),\ g^{(h-1)}(x') \right\rangle$ 将收敛到一个固定数，我们将其表示为 $\Sigma^{(h-1)}(x,\ x')$。这个协方差具有一个递归公式，对于 $h \in [L]$,

$$\begin{aligned}
&\Sigma^{(0)}(x,x') = x^\top x', \\
&\Lambda^{(h)}(x,x') = \begin{pmatrix} \Sigma^{(h-1)}(x,x) & \Sigma^{(h-1)}(x,x') \\ \Sigma^{(h-1)}(x',x) & \Sigma^{(h-1)}(x',x') \end{pmatrix} \in \mathbb{R}^{2\times 2}, \\
&\Sigma^{(h)}(x,x') = c_\sigma \mathbb{E}_{(u,v)\sim\mathcal{N}\left(0,\Lambda^{(h)}\right)} \left[\sigma(u)\,\sigma(v)\right].
\end{aligned} \tag{9.15}$$

现在我们推导这个公式。直觉是 $\left[f^{(h+1)}(x)\right]_i = \sum_{j=1}^{d_h} \left[W^{(h+1)}\right]_{i,j} \left[g^{(h)}(x)\right]_j$ 是一个居中的高斯过程取决于 $f^{(h)}$ $(\forall i \in [d_{h+1}])$，具有协方差

$$\begin{aligned}
&\mathbb{E}\left[\left[f^{(h+1)}(x)\right]_i \cdot \left[f^{(h+1)}(x')\right]_i \middle| f^{(h)}\right] \\
&= \left\langle g^{(h)}(x), g^{(h)}(x') \right\rangle \\
&= \frac{c_\sigma}{d_h} \sum_{j=1}^{d_h} \sigma\left(\left[f^{(h)}(x)\right]_j\right) \sigma\left(\left[f^{(h)}(x')\right]_j\right),
\end{aligned} \tag{9.16}$$

哪个收敛到 $\Sigma^{(h)}(x,\ x')$，当且仅当每个 $\left[f^{(h)}\right]_j$ 是居中对齐的高斯过程，协方差为 $\Sigma^{(h-1)}$。这导致了方程（9.15）的归纳定义。

接下来我们处理第二个项 $\left\langle b^{(h)}(x), b^{(h)}(x') \right\rangle$ 从方程（9.13）我们得到

$$\begin{aligned}
&\left\langle b^{(h)}(x), b^{(h)}(x') \right\rangle \\
=&\left\langle \sqrt{\frac{c_\sigma}{d_h}} D^{(h)}(x) \left(W^{(h+1)}\right)^\top b^{(h+1)}(x), \sqrt{\frac{c_\sigma}{d_h}} D^{(h)}(x') \left(W^{(h+1)}\right)^\top b^{(h+1)}(x') \right\rangle .
\end{aligned} \tag{9.17}$$

尽管 $W^{(h+1)}$ 和 $b_{h+1}(x)$ 互相关联，但 $W^{(h+1)}$ 的高斯初始化使我们能够用全新的 $W^{(h+1)}$ 替换

样本$\widetilde{W}^{(h+1)}$不改变其极限：(参见 [?] 以获取精确证明。)

$$\begin{aligned}
&\left\langle \sqrt{\frac{c_\sigma}{d_h}} D^{(h)}(x) \left(W^{(h+1)}\right)^\top b^{(h+1)}(x), \sqrt{\frac{c_\sigma}{d_h}} D^{(h)}(x') \left(W^{(h+1)}\right)^\top b^{(h+1)}(x') \right\rangle \\
\approx &\left\langle \sqrt{\frac{c_\sigma}{d_h}} D^{(h)}(x) \left(\widetilde{W}^{(h+1)}\right)^\top b^{(h+1)}(x), \sqrt{\frac{c_\sigma}{d_h}} D^{(h)}(x') \left(\widetilde{W}^{(h+1)}\right)^\top b^{(h+1)}(x') \right\rangle \\
\to &\frac{c_\sigma}{d_h} \mathrm{tr} D^{(h)}(x) D^{(h)}(x') \left\langle b^{(h+1)}(x), b^{(h+1)}(x') \right\rangle \\
\to &\dot{\Sigma}^{(h)}(x, x') \left\langle b^{(h+1)}(x), b^{(h+1)}(x') \right\rangle .
\end{aligned}$$

应用此近似到方程（9.17）中，我们得到

$$\left\langle b^{(h)}(x), b^{(h)}(x') \right\rangle \to \prod_{h'=h}^{L} \dot{\Sigma}^{(h')}(x, x').$$

最后，由于$\left\langle \frac{\partial f(w,x)}{\partial w}, \frac{\partial f(w,x')}{\partial w} \right\rangle = \sum_{h=1}^{L+1} \left\langle \frac{\partial f(w,x)}{\partial W^{(h)}}, \frac{\partial f(w,x')}{\partial W^{(h)}} \right\rangle$，我们得到了全连接神经网络的最终NTK表达式：

$$\Theta^{(L)}(x, x') = \sum_{h=1}^{L+1} \left(\Sigma^{(h-1)}(x, x') \cdot \prod_{h'=h}^{L+1} \dot{\Sigma}^{(h')}(x, x') \right).$$

9.5 NTK in Practice

目前我们已经展示了一个具有特定初始化方案并由梯度流训练的超宽神经网络，对应于具有特定核函数的核。一个自然的问题是：*why don't we use this kernel classifier directly?*

最近的一系列工作表明，NTKs在经验上是有用的，尤其是在小到中等规模的数据集上。Arora等人[ADL+19]在UCI数据库中测试了90个小到中等规模的数据集上的NTK分类器。他们发现NTK可以击败神经网络，其他

[6] https://archive.ics.uci.edu/ml/datasets.php

核函数如高斯函数和最佳先前分类器，包括平均排名、平均准确度等在内的各种指标下的随机森林。这表明NTK分类器应该属于任何现成机器学习方法的列表中。

对于每个神经网络架构，都可以推导出一个相应的核函数。Du 等人 [DHS+19] 为图分类任务推导了图神经核函数（GNTK）。在各个社交网络和生物信息学数据集上，GNTK 可以优于图神经网络。

类似地，Arora等人[?]推导了与卷积神经网络相对应的卷积NTK（CNTK）公式。对于图像分类任务，在小规模数据和低样本设置中，CNTKs可以非常强大[ADL+19]。然而，对于大规模数据，Arora

等人[?]发现CNTK和CNN之间仍然存在性能差距。解释这一现象的理论是一个开放性问题。这可能需要超越NTK框架。

9.6 Exercises

1. NTK 公式对于 ReLU 激活函数：证明

$$\mathbb{E}_{w\sim\mathcal{N}(0,I)}\left[(\dot{\sigma}(w^\top x)\dot{\sigma}(w^\top x')\right]=\frac{\pi-\arccos\left(\frac{x^\top x'}{\|x\|_2\|x'\|_2}\right)}{2\pi}.$$

2. 证明方程（9.11）。（提示：泛化界限取决于最终 W 矩阵与初始 W 之间的差的范数，您可以使用类似于第3章中核回归分析的求和/积分来对其进行上界估计。）

3. 为什么NTK可以学习线性函数：在这个问题中，你被要求证明NTK可以学习线性函数（这里的技术也可以用来证明NTK可以学习第9.3.1节中列出的其他函数）。在这个问题中，我们假设我们有一些数据点 n，$\{(x_i,\ y_i)\}_{i=1}^n$，其中每个输入 x_i 的范数为1，以及 $y_i=\beta^\top x_i$ 对于某个 β。由两层ReLU神经网络) 诱导的神经切线核矩阵 H^* (的每个条目是 $H^*_{ij}=\frac{\pi-\arccos\left(\frac{x^\top x'}{\|x\|_2\|x'\|_2}\right)}{2\pi}$。

(a) 泰勒展开：使用 arccos(z) 的泰勒展开 $=\frac{\pi}{2}-z-\sum_{\ell=1}^{\infty}\frac{(2\ell-3)!!z}{(2\ell-2)!!}$ 来证明 H^* 允许以下形式

$$H^*=\frac{K}{4}+\sum_{\ell=1}^{\infty}\frac{1}{2\pi}\frac{(2\ell-3)!!}{(2\ell-2)!!}\cdot\frac{K^{\circ 2\ell}}{2\ell-1}$$

在 $K_{ij}=x_i^\top x_j$ 和 $K^\ell_{ij}=\left(x_i^\top x_j\right)^\ell$。

(b) 假设 H^* 和 K 是可逆的。证明 $(H^*)^{-1}\preceq 4K^{-1}$。

(c) 显示 tr $(H^*)\leq n$。

(д) 使用假设 $y_i=x_i^\top\beta$ 来证明

$$\frac{\sqrt{2y^\top(H^*)^{-1}y\cdot\mathrm{tr}(H^*)}}{n}\leq\frac{2\sqrt{2}\,\|\beta\|_2}{\sqrt{n}}.$$

10
Interpreting output of Deep Nets: Credit Attribution

本章考虑了试图理解的方法：*Why did the model give the answer it did?* 基本概念相当古老，但正确且高效地应用于深度学习环境相对较新。从数学上讲，所需的是对系统的各个组成部分，包括训练数据，进行 *credit attribution* 以做出最终决策。

我们考虑两种类型的解释。*Influence functions*试图理解单个数据点如何影响模型在测试数据点上的答案。*Saliency methods*试图从相同数据点的内容来理解模型在测试数据点上的答案，通常以热图（也称为 *saliency maps*）的形式呈现，描述单个坐标的重要性。在这些设置中，经常出现的一个优雅的想法是*Shapley values*。

10.1 Influence Functions

对于固定的训练数据集 S，*influence function* 捕获了从训练集 S 中添加或删除数据点 x 对测试数据点 z 上的答案（或损失）的影响。Cook 和 Weisberg 的文本 1

这是该主题的标准参考。计算影响函数的朴素方法是 *leave-one-out retraining*：对于每个 x，在 $S \setminus \{x\}$ 上重新计算模型。但也可以使用在 S 上训练的模型 θ^* 进行更直接的计算，该模型使用更连续的影响概念。

1 R D Cook 和 S Weisberg. 回归中的残差和影响。1982

设 $\ell()$ 为一个具有 $\ell(x、\theta)$ 的二阶可微损失函数，其中 $\ell(x、\theta)$ 表示模型参数 θ 在数据点 x 上的 () 损失。为了简洁，我们让 $\ell(S、\theta)$ 表示数据点集合 S 上的平均损失。影响 $I(x、z)$ 的正式定义涉及2思想实验。

修改训练点 x 的权重从 $1/|S|$ 变为 $1/|S| + \epsilon$ 的过程。对于固定的 S，令 θ^* 为 $\ell(S$ 的最小化值，$\theta)$ 和 $\theta^*_{x,\epsilon}$ 为扰动后的最小化值。

[2] Better notation might be $I_S(x,z)$, to clarify dependence on S.

定义 10.1.1. $I(x,\ z) = \frac{\partial}{\partial \epsilon}\ell(z,\ \theta^*_{x,\epsilon})|_{\epsilon=0}$.

影响函数是为了凸模型（如最小二乘线性回归）而发明的，因此理论假设 θ^* 满足 $\nabla_\theta(S,\ \theta^*) =$ 为 0 以及 $\nabla^2_\theta(S,\ \theta^*)$ 是正半定。3

3 在深度学习环境中，人们希望达到一个 *stationary point*，即 $\nabla_\theta(S,\ \theta^*)$ 为零（或者更现实地，接近零），此外它还是一个 *local optimum*，即 $\nabla^2_\theta(S,\ \theta^*)$ 是正半定的。在实践中，这两个条件都不完全成立，但 $\nabla^2_\theta(S,\ \theta^*)$ 通常不具有大的负特征值。因此，实际中的影响函数计算使用 $(H_{\theta^*} + \lambda I)^{-1}$，对于某个小的 $\lambda > 0$。

定理10.1.2. $I(x, z) = -\nabla_\theta(\ell(z, \theta^*))^T H^{-1}_{\theta^*}\nabla_\theta\ell(x, \theta^*)$.

Proof. 通过最优性，$\ell(S,\ \theta^* + \Delta\theta) \approx \ell(S,\ \theta^*) + \frac{1}{2}(\Delta\theta)^T H_{\theta^*}(\Delta\theta)$ 对于小的扰动 $\Delta\theta$。将数据点 x 的权重从 $1/\{S\}$ 改变为 $1/\{S\} + \epsilon$ 并重新优化给出 $\theta^*_{x,\epsilon} = \theta^* + \Delta\theta$，其中 $\Delta\theta$ 是最小化者

$$\epsilon\ell(x, \theta^* + \Delta\theta) + \frac{1}{2}(\Delta\theta)^T H_{\theta^*}(\Delta\theta).$$

自 $\ell(x,\ \theta^* + \Delta\theta) \approx \ell(x,\ \theta^*) + \nabla\ell(x,\ \theta^*) \cdot \Delta\theta$以来，我们得到的是一个二次表达式，并且它通过$\Delta\theta = -\epsilon H^{-1}_{\theta^*}\nabla_\theta(\ell(x,\ \theta^*))$最小化。由于参数从$\theta^*$到$\theta^* + \Delta\theta$的变化导致测试点$z$的损失变化了$(\Delta\theta) \times \nabla_\theta(\ell(z,\ \theta^*))$，现在定理成立。

□

10.1.1 *Computing Influence Functions*

首先看来，由于逆海森矩阵的计算，计算影响函数似乎很困难，这在大致上具有参数数量的立方复杂度。Koh和Liang 4 设计了更快的方法

4 P W Koh 和 P Liang. 通过影响函数理解黑盒预测。在 *Proc. ICML*，2017

方法。关键思想是以下问题中的简单识别。

问题 10.1.3. *If A is any positive definite matrix with full rank and maximum eigenvalue less than 1 then show that*[5] $A^{-1} = \sum_{i=0}^{\infty}(I - A)^i$ 。

5 提示：请注意 A 是可对角化的。A^i 的特征向量和特征值如何与 A 的特征向量和特征值相关联?

Agarwal等人6指出如何使用这个恒等式进行快速（但近似）的imate) 海森矩阵-向量计算。

6 Agarwal N, Bullins B, 和 Hazan E. 线性时间内用于机器学习的二阶随机优化。2017

问题 10.1.4. *If S_r denotes the truncation of the series to its first r terms, then show that $\lambda_{max}(A^{-1} - S_r) \leq$??.*

定理10.1.2表明，我们需要计算某些向量v的$H^{-1}v$，但问题10.1.4允许我们将它近似为某些合理小的r的$\sum_{i=0}^{r}(I - H)^i v$。由于$(I - H)v = v - Hv$，我们发现只需要进行$r$个Hessian-向量计算，每个计算时间与深度网络的规模成线性关系。（见第4.4.1节）。

10.2 *Shapley Values*

Shapley 值 7 是合作博弈论中的一个概念
使用以下设置。存在一个由N名玩家组成的群体（为简便起见，用$[N]$表示），他们愿意为了某个目标进行合作

7 Lloyd Shapley. *"Notes on the n-Person Game – II: The Value of an n-Person Game"*. RAND Corporation

目标。一个效用函数 {v*} 规定了每个玩家子集的奖励/效用：如果玩家子集 S 最终合作，他们将作为一个群体获得效用 $U(S)$。如果他们中的所有 N 都最终合作，如何恰当地和公平地分配效用 $U([N])$？在合理条件下，会出现唯一的支付 s_1，s_2，...，s_N，称为 *Shapley values*，使得 $\sum_i s_i = U([N])$ (定理 10.2.3)。

在机器学习中，以下两种设置代表了Shapley值的典型应用：(a) *Pricing of datapoints:*“玩家”可以是持有可用于训练机器学习模型的数据的个人，Shapley值可以被视为使用他们数据的报酬。(b) 当我们试图从单个（测试）数据点的角度，以单个坐标对深度网络决策的贡献（即 *saliency*）来理解深度网络的输出时。

Shapley 值定义为第 i 个玩家的值，使用玩家以随机顺序将自己添加到联盟中的思维实验，并观察当第 i 个玩家加入联盟时预期效用的增加。

定义 10.2.1. Shapley 值 *of player i, denoted s_i, is defined as the following expected value, where π is a random permutation of $\{1, 2, \ldots, N\}$ and $\pi_{<i}$ is shorthand for the subset of players that appear before i in the permutation:*

$$s_i = E_\pi \left[U(\pi_{\leq i}) - U(\pi_{<i}) \right]. \tag{10.1}$$

s_i的定义有点自然，尽管有人可能会对玩家随机加入联盟的略微人为假设提出异议。这里有一个避免随机顺序的等价定义。

问题 10.2.2. *Show that the following definition of Shapley value is equivalent:*

$$\begin{aligned} s_i &= \sum_{S \subseteq [N] \setminus \{i\}} \frac{\{S\}!(N-\{S\}-1)!)}{N!} (U(S \cup \{i\}) - U(S)) \\ &= \frac{1}{N} \sum_{S \subseteq [N] \setminus \{i\}} \frac{U(S \cup \{i\}) - U(S)}{\binom{N-1}{|S|}}. \end{aligned}$$

自$\binom{N-1}{k}$是大小为$N-$的集合中大小为k的子集的数量，问题10.2.2实际上重新定义了Shapley值s_i为以下随机过程的期望：从$[0, N-1]$中随机选择一个整数k，然后选择一个不包含i的大小为k的随机子集S，并测量在S中添加i时的效用变化。

虽然这似乎更自然，但仍然是一个好主意，了解我们是否遗漏了一些关于如何分配信用的根本不同的定义。让我们尝试为任何信用归属方法形式化自然属性。以下公理对于定义 s_i 的任何系统来说似乎是自然的。

Efficiency: 玩家价值之和为 $U([N])$。

Symmetry: 如果对于所有不包含 i 的 S，$U(S\cup\{i\}) = U(S\cup\{j\})$，则它们的付款相同。8

Linearity: 如果 U_1，U_2 是任意两个效用函数，那么 $U_1 + U_2$ 的支付是 U_1 的支付和 U_2 的支付之和。

Null Player: 如果对于所有不包含 i 的 S，则 i 的付款为零。

您可以迅速说服自己Shapley值满足公理。

定理10.2.3. *The payment scheme in Definition 10.2.1 is the only one that satisfies the previous axioms.*

问题 10.2.4. *Prove Theorem 10.2.3.* 9

10.2.1 *Algorithms to approximate Shapley values*

给定效用函数 U (的简短描述，例如作为一个电路)，在一般情况下计算 Shapley 值是 NP-hard 的，即意味着
(假设 $P \neq NP$) 运行时间将比 N 的任何多项式增长得更快，以及 U 的描述。然而，如果效用是有界的，则可以近似计算它们。具体来说，我们假设对于所有 S、i，$U(S\cup\{i\}) - U(S)$ 的绝对值有一个上界 R。

朴素近似：选择 $O(R^2N$ 个 $\log N/\epsilon^2)$ 随机排列，并使用它们来估计 (10.1) 中的期望。（注意 $U(\cdot)$ 的计算次数是 $O(R^2N^2\log N/\epsilon^2)$。这是计算成本。）然后，集中界限表明期望的估计 $\widehat{s_i}$ 落在 $[s_i - \epsilon/\sqrt{N}$, $s_i + \epsilon/\sqrt{N}]$ 内。这意味着所有 Shapley 值的向量估计在 ℓ_2 范数 ϵ 内，即 $\|s - \widehat{s}\|_2 \leq \epsilon$。

更好的近似：我们给出一种使用 $O(R^2N\log N)$ 次评估 $U(\cdot)$ 的方法。它使用了关于 Shapley 值的以下事实。

定理10.2.5. *Differences of Shapley values satisfy the following:*

$$s_i - s_j = \frac{1}{N-1}\sum_{S\subseteq[N]\setminus\{i,j\}} \frac{U(S\cup\{i\}) - U(S\cup\{j\})}{\binom{N-2}{|S|}}.$$

问题 10.2.6. *Prove Theorem 10.2.5 from Problem 10.2.2.*

现在我们可以描述该方法：(1) 使用朴素近似来逼近 s_1，其加性误差为 $\epsilon/2\sqrt{N}$。(2) 使用定理10.2.5中的特性，适当地采样集合 S 以估计所有类型 $s_1 - s_j$ 的差异，误差为 $\epsilon/2\sqrt{N}$。

8 有时这被描述为“支付应取决于他们的贡献，而不是他们的名字。”

9 提示：取两种支付方案之差。

10 X邓和C Papadimitriou. 关于合作解概念的复杂性。
Math of Operations Research，1994

问题 10.2.7. *Figure out how to do step (2). (Hint: You pick S with a certain probability $p(|S|)$. This uses the observation that the same S can be used to estimate $s_1 - s_j$ for many j's.)*

10.3 Data Models

这是一个同时为单个训练数据点分配信用，但采用线性回归方法的方法11。通过训练许多
模型在训练集子集的 p 分数数据点中，作者表明一些有趣的测试误差度量（使用 logit 值定义）表现如下：度量 $f(x)$ 可以很好地近似为一个（稀疏）线性表达式 $\theta_0 + \sum_i \theta_i x_i$，其中 x 是一个表示 p 分数训练数据点样本的二进制向量，$x_i = 1$ 表示第 i 个训练点的存在，$x_i = -1$ 表示不存在。系数 θ_i 通过 lasso 回归估计。这里的惊喜是 $f(x)$——这是在数据集 x 上深度学习的结果——被很好地近似为 $\theta_0 + \sum_i \theta_i x_i$。作者指出，$\theta_i$ 可以被视为第 i 个数据点离散影响的启发式估计。请注意，估计的 θ_i 依赖于上述过程中 p 的值。

11 安德鲁·伊利亚斯，宋敏·帕克，洛根·恩斯特罗姆，纪尧姆·莱克勒，亚历山大·马德里。数据模型：从训练数据预测预测。*arXiv preprint arXiv:2202.00622*，2022

为什么数据模型有效？原因与模型正在对训练集的随机子集的 p 分数进行训练有关。可以使用有趣但基本的谐波分析来理解它 12。

12 布拉夫曼·桑杰夫·阿鲁尔·尼昆什，阿鲁希·古普塔。通过谐波分析理解影响函数和数据模型。*ICLR*，2023

10.4 Saliency Maps

Saliency methods 尝试从与测试数据点内容相同的角度理解模型在该数据点上的答案，通常以热图（也称为*saliency maps*）的形式呈现，显示各个坐标的重要性。例如，如果深度网络通过标签对图像进行分类，那么对被标记为“狗”的图像的显著性图可能涉及突出显示确定此标签的像素。

待编写。Shapley值可以用来定义测试数据点中每个坐标对最终输出的“贡献”。还有其他方法。

也基于梯度的方法。

11

Inductive Biases due to Algorithmic Regularization

许多基于深度神经网络的现代机器学习系统过度参数化，即参数数量通常远大于样本大小。换句话说，存在（无限）多个经验风险的近似最小化者，其中许多在未见数据上泛化效果不佳。因此，为了使学习成功，关键是通过权衡经验损失与确保经验风险和总体风险接近的某个复杂度项，将学习算法偏向“更简单”的假设。实践中已使用几种显式正则化策略来帮助这些系统泛化，包括参数的ℓ_1和ℓ_2正则化[NH92]。

除了显式正则化技术外，从业者还使用了一系列算法方法来提高过参数化模型的一般化能力。这包括反向传播的早期停止[CLG01]、批量归一化[IS15b]、dropout[SHK+14]等更多方法1。尽管这些启发式方法已经取得了巨大的成功-

1 我們參考[KGC17]以獲得對這些50多個建議的出色闡述。

在训练深度网络方面取得了成功，但这些启发式方法如何在深度学习中提供正则化的理论理解仍然相对有限。

在这一章中，我们研究了由于Dropout引起的正则化，这是[SHK+14]最近提出的一种算法启发式。使用dropout训练神经网络的基本思想是，在正向传播过程中，我们根据伯努利分布独立且相同地随机丢弃神经网络中的神经元。具体来说，在反向传播算法的每一轮中，对于每个神经元，独立地，以概率p“丢弃”该神经元，因此它不会参与对给定数据点的预测，以概率$1-p$保留该神经元。

深度学习是一个关键创新由从业者推动的领域，其中许多技术是通过从其他领域汲取见解而激发的。例如，Dropout被引入作为一种打破神经元之间“共适应”的方法，从而汲取

[2] The parameter p is treated as a hyper-parameter which we typically tune for based on a validation set.

洞察高级生物进化中性生殖模型的成功。另一个被[SHK+14]引用的动机是关于“平衡网络”。尽管有针对解释Dropout 3的几个理论工作，但它仍然不清楚

3

Dropout提供了什么样的正则化，或者Dropout更喜欢哪些类型的网络，以及这如何有助于泛化。在本章中，我们通过实例化Dropout引起的正则化的显式形式，以及它们如何在各种机器学习（包括线性回归（第11.4节）、矩阵感知（第11.1.1节）、矩阵补全（第11.1.2节）和深度学习（第11.2节）中提供容量控制，来努力实现这一目标。

11.1 Matrix Sensing

我们从理解矩阵感知中的dropout开始，这是一个有大量应用且从理论角度来看已经得到充分理解的矩阵学习问题的重要实例。以下是问题设置。

设 $M_* \in \mathbb{R}^{d_2 \times d_0}$ 为一个秩为 r_* 的矩阵：$= \mathrm{Rank}(M_*)$。设 $A^{(1)}$，...，$A^{(n)}$ 为与 M_* 大小相同的测量矩阵集。矩阵感知的目标是从 n 个观察值 $y_i = \langle M_*, A^{(i)} \rangle$ 中恢复矩阵 M_*，使得 $n \ll d_2 d_0$。我们考虑的学习算法是经验风险最小化，并选择用其因子 $U \in \mathbb{R}^{d_2 \times d_1}$，$V \in \mathbb{R}^{d_0 \times d_1}$ 的乘积来表示参数矩阵 $M \in \mathbb{R}^{d_2 \times d_0}$：

$$\min_{U \in \mathbb{R}^{d_2 \times d_1}, V \in \mathbb{R}^{d_0 \times d_1}} \widehat{L}(U, V) := \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle UV^\top, A^{(i)} \rangle)^2. \tag{11.1}$$

当 $d_1 \gg r_*$ 时，存在许多“坏”的实证最小化器，即那些具有较大真实风险的。然而，最近，[LMZ18] 表明在限制等距性质下，尽管存在这样的较差的 ERM 解，但适当的初始化的梯度下降是 *implicitly* 偏向于找到具有最小核范数的解——这是一个重要的结果，它最初被 [GWB+17] 猜测并经验验证。

我们提出使用由于dropout的算法正则化来解决ERM问题（11.1），在训练时间，U和V的对应列根据伯努利随机变量独立且相同地被丢弃。与梯度下降的*implicit*效应相反，这种dropout启发式*explicitly*正则化了经验目标。然后，在矩阵感知的情况下，自然要问dropout是否也会使ERM偏向某些低范数解。为了回答这个问题，我们首先观察到dropout可以被视为以下

目标：

$$\widehat{L}_{\text{drop}}(\mathrm{U},\mathrm{V}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathrm{B}}(y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)}\rangle)^2, \tag{11.2}$$

在$\mathrm{B} \in \mathbb{R}^{d_1 \times d_1}$是一个对角矩阵，其对角元素是服从$\mathrm{B}_{jj} \sim \frac{1}{1-p}\mathrm{Ber}(1-p)$分布的伯努利随机变量，对于$j \in [d_1]$。在这种情况下，我们可以证明对于任何$p \in [0,1)$：

$$\widehat{L}_{\text{drop}}(\mathrm{U},\mathrm{V}) = \widehat{L}(\mathrm{U},\mathrm{V}) + \frac{p}{1-p}\widehat{R}(\mathrm{U},\mathrm{V}), \tag{11.3}$$

哪里

$$\widehat{R}(\mathrm{U},\mathrm{V}) = \sum_{j=1}^{d_1}\frac{1}{n}\sum_{i=1}^{n}(\mathrm{u}_j^\top \mathrm{A}^{(i)}\mathrm{v}_j)^2 \tag{11.4}$$

这是一个依赖于数据的项，它捕捉了由于dropout而引入的 *explicit* 正则化器。

Proof. 考虑方程11.2中的其中一个加项，然后我们可以写出

$$\begin{aligned}\mathbb{E}_{\mathrm{B}}[(y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)}\rangle)^2] &= \left(\mathbb{E}_{\mathrm{B}}[y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)}\rangle]\right)^2 \\ &\quad + \mathrm{Var}(y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)}\rangle)\end{aligned} \tag{11.5}$$

对于伯努利随机变量B_{jj}，我们有$\mathbb{E}[\mathrm{B}_{jj}] = 1$且$\mathrm{Var}(\mathrm{B}_{jj}) = \frac{p}{1-p}$。因此，右侧的第一个项等于$(y_i - \langle \mathrm{UV}^\top, \mathrm{A}^{(i)}\rangle)^2$。对于第二个项，我们有

$$\begin{aligned}\mathrm{Var}(y_i - \langle \mathrm{UBV}^\top, \mathrm{A}^{(i)}\rangle) &= \mathrm{Var}(\langle \mathrm{UBV}^\top, \mathrm{A}^{(i)}\rangle) \\ &= \mathrm{Var}(\langle \mathrm{B}, \mathrm{U}^\top \mathrm{A}^{(i)}\mathrm{V}\rangle) \\ &= \mathrm{Var}(\sum_{j=1}^{d_1}\mathrm{B}_{jj}\mathrm{u}_j^\top \mathrm{A}^{(i)}\mathrm{v}_j) \\ &= \sum_{j=1}^{d_1}(\mathrm{u}_j^\top \mathrm{A}^{(i)}\mathrm{v}_j)^2\,\mathrm{Var}(\mathrm{B}_{jj}) \\ &= \frac{p}{1-p}\sum_{j=1}^{d_1}(\mathrm{u}_j^\top \mathrm{A}^{(i)}\mathrm{v}_j)^2\end{aligned}$$

使用上述事实在方程（11.2）中，我们得到

$$\begin{aligned}\widehat{L}_{\text{drop}}(\mathrm{U},\mathrm{V}) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \langle \mathrm{UV}^\top, \mathrm{A}^{(i)}\rangle)^2 + \frac{1}{n}\sum_{i=1}^{n}\frac{p}{1-p}\sum_{j=1}^{d_1}(\mathrm{u}_j^\top \mathrm{A}^{(i)}\mathrm{v}_j)^2 \\ &= \widehat{L}(\mathrm{U},\mathrm{V}) + \frac{p}{1-p}\widehat{R}(\mathrm{U},\mathrm{V}).\end{aligned}$$

这完成了证明。 □

假设样本大小 n 足够大，给定样本上的 *explicit* 正则化器在底层数据分布方面的行为与其期望值非常相似 4。此外，鉴于我们

4 在温和的假设下，我们可以形式上证明dropout正则化器很好地集中在其均值周围

寻找 $\widehat{L}_{\text{drop}}$ 的最小值，只需考虑在所有产生相同经验损失的因子中，具有最小正则化器值的因子。这促使研究以下与分布相关的 *induced* 正则化器：

$$\Theta(\mathrm{M}) := \min_{\mathrm{UV}^\top = \mathrm{M}} R(\mathrm{U},\mathrm{V}), \quad \text{where} \quad R(\mathrm{U},\mathrm{V}) := \mathbb{E}_{\mathrm{A}}[\widehat{R}(\mathrm{U},\mathrm{V})].$$

接下来，我们考虑两个重要的随机感知矩阵的例子。

11.1.1 Gaussian Sensing Matrices

我们假设传感矩阵的元素独立同分布于标准高斯分布，即 $\mathrm{A}(i)_{k\ell} \sim \mathcal{N}(0,1)$。对于高斯传感矩阵，我们表明由于Dropout引起的正则化提供了核范数正则化。形式上，我们表明

$$\Theta(\mathrm{M}) = \frac{1}{d_1}\|\mathrm{M}\|_*^2. \tag{11.6}$$

Proof. 我们回忆方程11.4中矩阵感知问题的dropout正则化的一般形式，并对感知矩阵上的分布取期望。然后，对于任何一对因子（U，V），期望正则化如下。

$$\begin{aligned}
R(\mathrm{U},\mathrm{V}) &= \sum_{j=1}^{d_1} \mathbb{E}(\mathrm{u}_j^\top \mathrm{A}\mathrm{v}_j)^2 \\
&= \sum_{j=1}^{d_1} \mathbb{E}\Big(\sum_{k=1}^{d_2}\sum_{\ell=1}^{d_0} \mathrm{U}_{kj}\mathrm{A}_{k\ell}\mathrm{V}_{\ell j}\Big)^2 \\
&= \sum_{j=1}^{d_1}\sum_{k,k'=1}^{d_2}\sum_{\ell,\ell'=1}^{d_0} \mathrm{U}_{kj}\mathrm{U}_{k'j}\mathrm{V}_{\ell j}\mathrm{V}_{\ell' j}\,\mathbb{E}[\mathrm{A}_{k\ell}\mathrm{A}_{k'\ell'}] \\
&= \sum_{j=1}^{d_1}\sum_{k=1}^{d_2}\sum_{\ell=1}^{d_0} \mathrm{U}_{kj}^2\mathrm{V}_{\ell j}^2\,\mathbb{E}[\mathrm{A}_{k\ell}^2] \\
&= \sum_{j=1}^{d_1}\sum_{k=1}^{d_2}\sum_{\ell=1}^{d_0} \mathrm{U}_{kj}^2\mathrm{V}_{\ell j}^2 \\
&= \sum_{j=1}^{d_1} \|\mathrm{u}_j\|^2\|\mathrm{v}_j\|^2
\end{aligned}$$

现在，使用柯西-施瓦茨不等式，我们可以将期望正则化器的界限表示为

$$\begin{aligned} R(\mathrm{U},\mathrm{V}) &\geq \frac{1}{d_1}\left(\sum_{i=1}^{d_1}\|\mathrm{u}_i\|\|\mathrm{v}_i\|\right)^2 \\ &= \frac{1}{d_1}\left(\sum_{i=1}^{d_1}\|\mathrm{u}_i\mathrm{v}_i^\top\|_*\right)^2 \\ &\geq \frac{1}{d_1}\left(\|\sum_{i=1}^{d_1}\mathrm{u}_i\mathrm{v}_i^\top\|_*\right)^2 = \frac{1}{d_1}\|\mathrm{U}\mathrm{V}^\top\|_*^2 \end{aligned}$$

在任意一对向量 a, b 中，等式成立，因为 $\|\mathrm{a}\mathrm{b}^\top\|_* = \|\mathrm{a}\mathrm{b}^\top\|_F = \|\mathrm{a}\|\|\mathrm{b}\|$，最后一个不等式是由于三角不等式。

接下来，我们需要从[MAV18]中得到以下关键结果。

定理11.1.1. 对于任意一对矩阵$\mathrm{U} \in \mathbb{R}^{d_2\times d_1}$，$\mathrm{V} \in \mathbb{R}^{d_0\times d_1}$，存在一个旋转矩阵$\mathrm{Q} \in \mathrm{SO}(d_1)$，使得矩阵$\widetilde{\ }\mathrm{U} := \mathrm{U}\mathrm{Q}$，$\widetilde{\ }\mathrm{V} := \mathrm{V}\mathrm{Q}$满足$\|\widetilde{\ }\mathrm{u}_i\|\|\widetilde{\ }\mathrm{v}_i\| = \frac{1}{d_1}\|\mathrm{U}\mathrm{V}^\top\|_*$，对于所有$i \in [d_1]$。

使用定理11.1.1在(U, V)上，UQ，VQ处的期望dropout正则化器如下

$$\begin{aligned} R(\mathrm{U}\mathrm{Q},\mathrm{V}\mathrm{Q}) &= \sum_{i=1}^{d_1}\|\mathrm{U}\mathrm{q}_i\|^2\|\mathrm{V}\mathrm{q}_i\|^2 \\ &= \sum_{i=1}^{d_1}\frac{1}{d_1^2}\|\mathrm{U}\mathrm{V}^\top\|_*^2 \\ &= \frac{1}{d_1}\|\mathrm{U}\mathrm{V}^\top\|_*^2 \\ &\leq \Theta(\mathrm{U}\mathrm{V}^\top) \end{aligned}$$

这完成了证明。 □

为了完整性，我们提供了定理11.1.1的证明。

Proof. 定义 $\mathrm{M} := \mathrm{U}\mathrm{V}^\top$。令 $\mathrm{M} = \mathrm{W}\Sigma\mathrm{Y}^\top$ 为 M 的紧凑奇异值分解。定义 $\widehat{\ }\mathrm{U} := \mathrm{W}\Sigma^{1/2}$ 和 $\widehat{\ }\mathrm{V} := \mathrm{Y}\Sigma^{1/2}$。令 $\mathrm{G}_\mathrm{U} = \widehat{\ }\mathrm{U}^\top\widehat{\ }\mathrm{U}$ 和 $\mathrm{G}_\mathrm{V} = \widehat{\ }\mathrm{V}^\top\widehat{\ }\mathrm{V}$ 分别为各自的格拉姆矩阵。观察 $\mathrm{G}_\mathrm{U} = \mathrm{G}_\mathrm{V} = \Sigma$。我们将证明存在一个旋转 Q，使得对于 $\widetilde{\ }\mathrm{U} = \widehat{\ }\mathrm{U}\mathrm{Q}$，$\widetilde{\ }\mathrm{V} = \widehat{\ }\mathrm{V}\mathrm{Q}$，有

$$\|\widetilde{\mathrm{u}}_j\|^2 = \frac{1}{d_1}\|\widetilde{\mathrm{U}}\|_F^2 = \frac{1}{d_1}\mathrm{Tr}(\widetilde{\mathrm{U}}^\top\widetilde{\mathrm{U}}) = \frac{1}{d_1}\mathrm{Tr}(\Sigma) = \frac{1}{d_1}\|\mathrm{M}\|_*$$

和

$$\|\widetilde{\mathrm{v}}_j\|^2 = \frac{1}{d_1}\|\widetilde{\mathrm{V}}\|_F^2 = \frac{1}{d_1}\mathrm{Tr}(\widetilde{\mathrm{V}}^\top\widetilde{\mathrm{V}}) = \frac{1}{d_1}\mathrm{Tr}(\Sigma) = \frac{1}{d_1}\|\mathrm{M}\|_*$$

因此，有 $\|\tilde{u}_i\|\|\tilde{v}_i\| = \frac{1}{d_1}\|M\|_*$。

所有剩下的就是给出矩阵Q的构造。我们注意到，如果且仅如果$Q^\top G_U Q$的所有对角元素都相等[5]，并且等于 $\frac{\mathrm{Tr}\, G_U}{d_1}$. 关键思想是对于迹为零的矩阵 $G_1 := G_U - \frac{\mathrm{Tr}\, G_U}{d_1} I_{d_1}$，如果 $G_1 = \sum_{i=1}^r \lambda_i e_i e_i^\top$ 是 G_1 的特征分解，那么对于特征向量的平均值，即对于 $w_{11} = \frac{1}{\sqrt{r}} \sum_{i=1}^r e_i$，有 $w_{11}^\top G_1 w_{11} = 0$。我们递归地使用这个性质来展示一个正交变换 Q，使得 $Q^\top G_1 Q$ 在其对角线上为零。

[5] since $(Q^\top G_U Q)_{jj} = \|\tilde{u}_j\|^2$

为了验证这个说法，首先注意到 w_{11} 是单位范数

$$\|w_{11}\|^2 = \|\frac{1}{\sqrt{r}} \sum_{i=1}^r e_i\|^2 = \frac{1}{r} \sum_{i=1}^r \|e_i\|^2 = 1.$$

此外，很容易看出

$$w_{11}^\top G w_{11} = \frac{1}{r} \sum_{i,j=1}^r e_i G e_j = \frac{1}{r} \sum_{i,j=1}^r \lambda_j e_i^\top e_j = \frac{1}{r} \sum_{i=1}^r \lambda_i = 0.$$

完整 $W_1 := [w_{11}, w_{12}, \cdots, w_{1d}]$ 应满足 $W_1^\top W_1 = W_1 W_1^\top = I_d$。观察 $W_1^\top G_1 W_1$ 的第一个对角元素为零

$$W_1^\top G_1 W_1 = \begin{bmatrix} 0 & b_1^\top \\ b_1 & G_2 \end{bmatrix}$$

主要子矩阵 G_2 也具有零迹。用类似的方法，设 $w_{22} \in \mathbb{R}^{d-1}$ 使得 $\|w_{22}\| = 1$ 和 $w_{22}^\top G_2 w_{22} = 0$，并定义 $W_2 = \begin{bmatrix} 1 & 0 & 0 \cdots 0 & 0 \\ 0 & w_{22} & w_{23} \cdots w_{2d} \end{bmatrix} \in \mathbb{R}^{d \times d}$ 使得 $W_2^\top W_2 = W_2 W_2^\top = I_d$，并观察到

$$(W_1 W_2)^\top G_1 (W_1 W_2) = \begin{bmatrix} 0 & \cdot & \cdots \\ \cdot & 0 & \cdots \\ \vdots & \vdots & G_3 \end{bmatrix}.$$

此过程可以递归应用，因此对于矩阵$Q = W_1 W_2 \cdots W_d$，我们有

$$Q^\top G_1 Q = \begin{bmatrix} 0 & \cdot & \cdots & \cdot \\ \cdot & 0 & \cdots & \cdot \\ \vdots & \vdots & \ddots & \vdots \\ \cdot & \cdot & \cdot & 0 \end{bmatrix},$$

因此 $\mathrm{Tr}(\tilde{U}\tilde{U}^\top) = \mathrm{Tr}(Q^\top G_U Q) = \mathrm{Tr}(\Sigma) = \mathrm{Tr}(Q^\top G_V Q) = \mathrm{Tr}(\tilde{V}^\top \tilde{V})$。

□

11.1.2 *Matrix Completion*

接下来，我们考虑矩阵补全问题，它可以被表述为具有随机指示矩阵的矩阵感知的特殊情况。形式上，我们假设对于所有 $j \in [n]$，令 $A^{(j)}$ 为一个指示矩阵，其 (i,k)-th 元素以概率 $p(i)q(k)$ 随机选择，其中 $p(i)$ 和 $q(k)$ 分别表示选择 i-th 行和 j-th 列的概率。

我们将展示，在这个设置中，Dropout 诱导了 [SS10] 和 [FSSS11] 研究的 *weighted trace-norm*。形式上，我们证明

$$\Theta(M) = \frac{1}{d_1}\|\mathrm{diag}(\sqrt{p})UV^\top \mathrm{diag}(\sqrt{q})\|_*^2. \tag{11.7}$$

Proof. 对于任何一对因子（U，V），都成立{v*}

$$\begin{aligned}
R(U,V) &= \sum_{j=1}^{d_1} \mathbb{E}(u_j^\top A v_j)^2 \\
&= \sum_{j=1}^{d_1}\sum_{k=1}^{d_2}\sum_{\ell=1}^{d_0} p(k)q(\ell)(u_j^\top e_k e_\ell^\top v_j)^2 \\
&= \sum_{j=1}^{d_1}\sum_{k=1}^{d_2}\sum_{\ell=1}^{d_0} p(k)q(\ell)U(k,j)^2 V(\ell,j)^2 \\
&= \sum_{j=1}^{d_1} \|\sqrt{\mathrm{diag}(p)}u_j\|^2 \|\sqrt{\mathrm{diag}(q)}v_j\|^2 \\
&\geq \frac{1}{d_1}\left(\sum_{j=1}^{d_1} \|\sqrt{\mathrm{diag}(p)}u_j\|\|\sqrt{\mathrm{diag}(q)}v_j\|\right)^2 \\
&= \frac{1}{d_1}\left(\sum_{j=1}^{d_1} \|\sqrt{\mathrm{diag}(p)}u_j v_j^\top \sqrt{\mathrm{diag}(q)}\|_*\right)^2 \\
&\geq \frac{1}{d_1}\left(\|\sqrt{\mathrm{diag}(p)}\sum_{j=1}^{d_1} u_j v_j^\top \sqrt{\mathrm{diag}(q)}\|_*\right)^2 \\
&= \frac{1}{d_1}\|\sqrt{\mathrm{diag}(p)}UV^\top\sqrt{\mathrm{diag}(q)}\|_*^2
\end{aligned}$$

第一不等式由柯西-施瓦茨不等式得出，第二个不等式由三角不等式得出。第一不等式右侧的等式由以下事实得出：对于任意两个向量a，b，$\|ab^\top\|_* = \|ab^\top\|_F = \|a\|\|b\|$。由于不等式对任何U，V都成立，因此它意味着

$$\Theta(UV^\top) \geq \frac{1}{d_1}\|\sqrt{\mathrm{diag}(p)}UV^\top\sqrt{\mathrm{diag}(q)}\|_*^2.$$

应用定理11.1.1于$(\sqrt{\mathrm{diag}(p)}U, \sqrt{\mathrm{diag}(q)}V)$，其中

存在一个旋转矩阵 Q，使得

$$\|\sqrt{\mathrm{diag}(\mathrm{p})}\mathrm{U}\mathrm{q}_j\|\|\sqrt{\mathrm{diag}(\mathrm{q})}\mathrm{V}\mathrm{q}_j\| = \frac{1}{d_1}\|\sqrt{\mathrm{diag}(\mathrm{p})}\mathrm{U}\mathrm{V}^\top\sqrt{\mathrm{diag}(\mathrm{q})}\|_*.$$

我们在UQ，VQ评估期望的dropout正则化器 :

$$\begin{aligned} R(\mathrm{UQ},\mathrm{VQ}) &= \sum_{j=1}^{d_1}\|\sqrt{\mathrm{diag}(\mathrm{p})}\mathrm{U}\mathrm{q}_j\|^2\|\sqrt{\mathrm{diag}(\mathrm{q})}\mathrm{V}\mathrm{q}_j\|^2 \\ &= \sum_{j=1}^{d_1}\frac{1}{d_1^2}\|\sqrt{\mathrm{diag}(\mathrm{p})}\mathrm{U}\mathrm{V}^\top\sqrt{\mathrm{diag}(\mathrm{q})}\|_*^2 \\ &= \frac{1}{d_1}\|\sqrt{\mathrm{diag}(\mathrm{p})}\mathrm{U}\mathrm{V}^\top\sqrt{\mathrm{diag}(\mathrm{q})}\|_*^2 \\ &\le \Theta(\mathrm{U}\mathrm{V}^\top) \end{aligned}$$

这完成了证明。 □

以上结果很有趣，因为它们将深度学习中的算法启发式方法Dropout与经验上有效且理论上理解良好的强复杂度度量联系起来。为了说明，这里我们给出了矩阵补全中带有dropout的泛化界限，以经验问题的最小值处的*explicit*正则化器的值为依据。

定理11.1.2.不失一般性，假设$d_2 \ge d_0$和$\|\mathrm{M}_*\| \le 1$。此外，假设$\min_{i,j} p(i)q(j) \ge \frac{\log(d_2)}{n\sqrt{d_2 d_0}}$。令(U, V)为方程(11.2)中dropout ERM目标的最小化者，并假设$\max_i \|\mathrm{U}(i,:)\|^2 \le \gamma$，$\max_i \|\mathrm{V}(i,:)\|^2 \le \gamma$。令$\alpha$满足$\widehat{R}(\mathrm{U},\mathrm{V}) \le \alpha/d_1$。那么，对于任何$\delta \in (0,1)$，以下泛化界限至少以概率$1-2\delta$在大小为$n$的样本中成立：

$$L(\mathrm{U},\mathrm{V}) \le \widehat{L}(\mathrm{U},\mathrm{V}) + C(1+\gamma)\sqrt{\frac{\alpha d_2\log(d_2)}{n}} + C'(1+\gamma^2)\sqrt{\frac{\log(2/\delta)}{2n}}$$

只要 $n = \Omega\left((d_1\gamma^2/\alpha)^2\log(2/\delta)\right)$，其中 C、C' 是一些绝对常数。

定理11.1.2的证明基于ℓ_2损失的标准化泛化界限[MRT18]，该界限基于具有加权迹范数有界于$\sqrt{\alpha}$的函数类的Rademacher复杂度[BM02]，即$\mathcal{M}_\alpha := \{\mathrm{M}: \|\mathrm{diag}(\sqrt{\mathrm{p}})\mathrm{M}\mathrm{diag}(\sqrt{\mathrm{q}})\|_*^2 \le \alpha\}$。该类Rademacher复杂度的界限由[FSSS11]建立。这里的技巧包括证明显式正则化器在其期望值周围很好地集中，以及推导预测的上界。还有一些注意事项。

我们要求采样分布是非退化的，如条件 $\min_{i,j} p(i)q(j) \geq \frac{\log(d_2)}{n\sqrt{d_2 d_0}}$ 所指定。这是自然的要求，用于界定 $\mathcal{M}_\alpha$ 的 Rademacher 复杂度，如 [FSSS11] 中所讨论的。

我们注意到，对于足够大的样本量，$\widehat{R}(\mathrm{U}, \mathrm{V}) \approx R(\mathrm{U}, \mathrm{V}) \approx \Theta(\mathrm{U}\mathrm{V}^\top) = \frac{1}{d_1}\|\mathrm{diag}(\sqrt{p})\mathrm{U}\mathrm{V}^\top \mathrm{diag}(\sqrt{q})\|_*^2$，其中第二个近似是由于(U, V)对是极小值的事实。也就是说，与加权的迹范数相比，在极小值处的显式正则化器的值大致按$1/d_1$的比例缩放。因此，推论陈述中的假设$\widehat{R}(\mathrm{U}, \mathrm{V}) \leq \alpha/d_1$。

实际上，对于使用dropout训练的模型，训练误差 $\widehat{L}(\mathrm{U}, \mathrm{V})$ 可以忽略不计。此外，鉴于样本量足够大，第三项可以任意小。因此，第二项，即$\widetilde{O}(\gamma\sqrt{\alpha d_2/n})$，主导了定理11.1.2中泛化误差界限的右侧。

The assumption $\max_i \|\mathrm{U}(i,:)\|^2 \leq \gamma, \max_i \|\mathrm{V}(i,:)\|^2 \leq \gamma$ is motivated by the practice of deep learning; such *max-norm* constraints are typically used with dropout, where the norm of the vector of incoming weights at each hidden unit is constrained to be bound by a constant [SHK+14]. In this case, if a dropout update violates this constraint, the weights of the hidden unit are projected back to the constraint norm ball. In proofs, we need this assumption to give a concentration bound for the empirical explicit regularizer, as well as bound the supremum deviation between the predictions and the true values. We remark that the value of γ also determines the complexity of the function class. On one hand, the generalization gap explicitly depends on and increases with γ. However, when γ is large, the constraints on U, V are milder, so that $\widehat{L}(\mathrm{U}, \mathrm{V})$ can be made smaller.

最后，所需的样本大小严重依赖于最优值处的显式正则化器的值（α/d_1），因此也依赖于dropout率p。特别是，增加dropout率会增加正则化参数$\lambda := \frac{p}{1-p}$，从而加剧显式正则化器带来的惩罚。直观上，较大的dropout率p会导致较小的α，从而可以保证更紧密的泛化间隙。我们通过实验表明，在实践中确实如此。

11.2 *Deep neural networks*

接下来，我们关注具有多个隐藏层的神经网络。令 $\mathcal{X} \subseteq \mathbb{R}^{d_0}$ 和 $\mathcal{Y} \subseteq \mathbb{R}^{d_k}$ 分别表示输入和输出空间。令 $\mathcal{D}$ 表示在 $\mathcal{X} \times \mathcal{Y}$ 上的联合概率分布。给定从联合分布中独立同分布抽取的 n 个示例 $\{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^n \sim \mathcal{D}^n$，以及损失函数 $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$，学习的目标是

是要找到一个假设 $f_{\mathrm{w}} : \mathcal{X} \to \mathcal{Y}$，由 w 参数化，其具有小的 *population risk* $L(\mathrm{w}) := \mathbb{E}_{\mathcal{D}}[\ell(f_{\mathrm{w}}(\mathrm{x}), \mathrm{y})]$。

我们关注平方 ℓ_2 损失，即 $\ell(\mathrm{y}, \mathrm{y}') = \|\mathrm{y} - \mathrm{y}'\|^2$，并研究 dropout 算法的泛化性质，以最小化 *empirical risk* $\widehat{L}(\mathrm{w}) := \frac{1}{n}\sum_{i=1}^{n}[\|\mathrm{y}_i - f_{\mathrm{w}}(\mathrm{x}_i)\|^2]$。我们考虑与具有 k 层的前馈神经网络相关的假设类，即形式为 $f_{\mathrm{w}}(\mathrm{x}) = \mathrm{W}_k\sigma(\mathrm{W}_{k-1}\sigma(\cdots \mathrm{W}_2\sigma(\mathrm{W}_1\mathrm{x}$ $)\cdots))$ 的函数，其中 $\mathrm{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$，对于 $i \in [k]$，是第 i 层的权重矩阵。参数 w 是权重矩阵集合 $\{\mathrm{W}_k, \mathrm{W}_{k-1}, \ldots, \mathrm{W}_1\}$ 和 $\sigma : \mathbb{R} \to \mathbb{R}$ 是应用于输入向量的逐项激活函数。

在现代机器学习系统中，我们与其讨论特定的网络拓扑，不如从层拓扑的角度思考，其中每一层都可能具有不同的特性——例如，全连接、局部连接或卷积。在卷积神经网络中，通常的做法是将dropout仅应用于全连接层，而不是卷积层。此外，在深度回归中，观察到仅对隐藏层之一应用dropout最为有效[LMAPH19]。在我们的研究中，dropout应用于学习到的表示或*features*之上，即顶层隐藏层的输出。在这种情况下，dr opout更新可以被视为*dropout objective*上的随机梯度下降迭代：

$$\widehat{L}_{\mathrm{drop}}(\mathrm{w}) := \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\mathrm{B}}\,\|\mathrm{y}_i - \mathrm{W}_k\mathrm{B}\sigma(\mathrm{W}_{k-1}\sigma(\cdots \mathrm{W}_2\sigma(\mathrm{W}_1\mathrm{x}_i)\cdots))\|^2 \tag{11.8}$$

在B是一个对角随机矩阵，其对角元素以相同且独立的方式分布为B $_{ii} \sim \frac{1}{1-p}\,\mathrm{Bern}(1-p)$，$i \in [d_{k-1}]$，对于某个 *dropout rate* p。我们试图理解由于dropout产生的 *explicit* 正则化器：

$$\widehat{R}(\mathrm{w}) := \widehat{L}_{\mathrm{drop}}(\mathrm{w}) - \widehat{L}(\mathrm{w}) \qquad \text{(显式正则化器)}$$

我们用 i-th 隐藏层上的输入向量 x 的 j-th 隐藏节点的输出表示为 $a_{i,j}(\mathrm{x}) \in \mathbb{R}$；例如，$a_{1,2}(\mathrm{x}) = \sigma(\mathrm{W}_2(1,:)^\top\sigma(\mathrm{W}_1\mathrm{x}))$。同样，向量 $\mathrm{a}_j(\mathrm{x}$ $) \in \mathbb{R}^{d_j}$ 表示输入 x 在 j-th 层上的激活。使用这种表示法，我们可以方便地将 Dropout 目标（见方程 11.8）重写为 $\widehat{L}_{\mathrm{drop}}(\mathrm{w})$: $= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\mathrm{B}}\,\|\mathrm{y}_i - \mathrm{W}_k\mathrm{B}\mathrm{a}_{k-1}(\mathrm{x}_i)\|^2$。然后很容易证明，由于 dropout 而产生的显式正则化器如下所示。

命题11.2.1（深度回归中的Dropout正则化器）。

$$\widehat{L}_{\mathrm{drop}}(\mathrm{w}) = \widehat{L}(\mathrm{w}) + \widehat{R}(\mathrm{w}), \text{ where } \widehat{R}(\mathrm{w}) = \lambda\sum_{j=1}^{d_{k-1}}\|\mathrm{W}_k(:,j)\|^2\widehat{a}_j^2.$$

在 $\widehat{a}_j = \sqrt{\frac{1}{n}\sum_{i=1}^n a_{j,k-1}(\mathrm{x}_i)^2}$ 和 $\lambda = \frac{p}{1-p}$ 是正则化参数。

Proof. 回忆起 $\mathbb{E}[\mathrm{B}_{ii}] = 1$ 和 $\mathrm{Var}(\mathrm{B}_{ii}) = \frac{p}{1-p}$。在当前小批量中的 x，y 条件下，我们有

$$\mathbb{E}_{\mathrm{B}}[\|\mathrm{y} - \mathrm{W}_k \mathrm{B} a_{k-1}(\mathrm{x})\|^2] = \sum_{i=1}^{d_k} \mathbb{E}_{\mathrm{B}}(y_i - \mathrm{W}_k(i,:)^\top \mathrm{B} a_{k-1}(\mathrm{x}))^2. \quad (11.9)$$

以下对上述每个加数均成立：

$$\begin{aligned}\mathbb{E}_{\mathrm{B}}(y_i - \mathrm{W}_k(i,:)^\top \mathrm{B} a_{k-1}(\mathrm{x}))^2 &= \Big(\mathbb{E}_{\mathrm{B}}[y_i - \mathrm{W}_k(i,:)^\top \mathrm{B} a_{k-1}(\mathrm{x})]\Big)^2 \\ &\quad + \mathrm{Var}(y_i - \mathrm{W}_k(i,:)^\top \mathrm{B} a_{k-1}(\mathrm{x})).\end{aligned}$$

自 $\mathbb{E}[\mathrm{B}] = 我$，右侧的第一个项等于 $(y_i - \mathrm{W}_k(:, \ i)^\top \mathrm{a}_{k-1}(\mathrm{x}))^2$。对于第二个项，我们有

$$\begin{aligned}\mathrm{Var}(y_i - \mathrm{W}_k(i,:)^\top \mathrm{B} a_{k-1}(\mathrm{x})) &= \mathrm{Var}(\mathrm{W}_k(i,:)^\top \mathrm{B} a_{k-1}(\mathrm{x})) \\ &= \mathrm{Var}(\sum_{j=1}^{d_{k-1}} \mathrm{W}_k(i,j)\mathrm{B}_{jj} a_{j,k-1}(\mathrm{x})) \\ &= \sum_{j=1}^{d_{k-1}} (\mathrm{W}_k(i,j) a_{j,k-1}(\mathrm{x}))^2 \mathrm{Var}(\mathrm{B}_{jj}) \\ &= \frac{p}{1-p} \sum_{j=1}^{d_{k-1}} \mathrm{W}_k(i,j)^2 a_{j,k-1}(\mathrm{x})^2\end{aligned}$$

将上述内容代入方程（11.9）

$$\begin{aligned}\mathbb{E}_{\mathrm{B}}[\|\mathrm{y} - \mathrm{W}_k \mathrm{B} a_{k-1}(\mathrm{x})\|^2] &= \|\mathrm{y} - \mathrm{W}_k a_{k-1}(\mathrm{x})\|^2 \\ &\quad + \frac{p}{1-p} \sum_{j=1}^{d_{k-1}} \|\mathrm{W}_k(:,j)\|^2 a_{j,k-1}(\mathrm{x})^2\end{aligned}$$

现在对x，y取经验平均值，我们得到

$$\widehat{L}_{\mathrm{drop}}(\mathrm{w}) = \widehat{L}(\mathrm{w}) + \frac{p}{1-p} \sum_{j=1}^{d_{k-1}} \|\mathrm{W}_k(:,j)\|^2 \widehat{a}_j^2 = \widehat{L}(\mathrm{w}) + \widehat{R}(\mathrm{w})$$

这完成了证明。 □

显式正则化器 $\widehat{R}(\mathrm{w})$ 是对隐藏节点的求和，为对应神经元输出的经验二阶矩与输出权重平方范数的乘积。对于具有 ReLU 的两层神经网络，当输入分布是对称和各向同性的，期望正则化器等于网络 [NTS15b] 的平方 ℓ_2 路径范数。这种连接已在深度线性网络 [MAV18, MA19] 中建立；这里我们将该结果扩展到单隐藏层 ReLU 网络。

命题11.2.2. 考虑一个具有ReLU激活函数的隐藏层的两层神经网络 $f_{\mathrm{W}}(\cdot)$。此外，假设边际输入分布$\mathbb{P}_{\mathcal{X}}(\mathrm{x})$是对称和各向同性的，即 $\mathbb{P}_{\mathcal{X}}(\mathrm{x}) = \mathbb{P}_{\mathcal{X}}(-\mathrm{x})$和$\mathbb{E}[\mathrm{x}\mathrm{x}^\top] = \mathrm{I}$。那么由于dropout产生的期望显式正则化器如下：

$$R(\mathrm{w}) := \mathbb{E}[\widehat{R}(\mathrm{w})] = \frac{\lambda}{2} \sum_{i_0,i_1,i_2=1}^{d_0,d_1,d_2} \mathrm{W}_2(i_2,i_1)^2 \mathrm{W}_1(i_1,i_0)^2, \qquad (11.10)$$

Proof of Proposition 11.2.2. 使用命题11.2.1，我们已有 ve that:

$$R(\mathrm{w}) = \mathbb{E}[\widehat{R}(\mathrm{w})] = \lambda \sum_{j=1}^{d_1} \|\mathrm{W}_2(:,j)\|^2 \, \mathbb{E}[\sigma(\mathrm{W}_1(j,:)^\top \mathrm{x})^2]$$

它需要计算数量 $\mathbb{E}_{\mathrm{x}}[\sigma(\mathrm{W}_1(j,:)^\top \mathrm{x})^2]$。根据对称假设，我们有 $\mathbb{P}_{\mathcal{X}}(\mathrm{x}) = \mathbb{P}_{\mathcal{X}}(-\mathrm{x})$。因此，对于任何 $\mathrm{v} \in \mathbb{R}^{d_0}$，我们也有 $\mathbb{P}(\mathrm{v}^\top \mathrm{x}) = \mathbb{P}(-\mathrm{v}^\top \mathrm{x})$。也就是说，随机变量 $z_j := \mathrm{W}_1(j,:)^\top \mathrm{x}$ 也关于原点对称。很容易看出 $\mathbb{E}_z[\sigma(z)^2] = \frac{1}{2}\mathbb{E}_z[z^2]$。

$$\begin{aligned} \mathop{\mathbb{E}}_{z}[\sigma(z)^2] &= \int_{-\infty}^{\infty} \sigma(z)^2 d\mu(z) \\ &= \int_{0}^{\infty} \sigma(z)^2 d\mu(z) = \int_{0}^{\infty} z^2 d\mu(z) \\ &= \frac{1}{2}\int_{-\infty}^{\infty} z^2 d\mu(z) = \frac{1}{2}\mathop{\mathbb{E}}_{z}[z^2]. \end{aligned}$$

将上述恒等式代入 $R(\mathrm{w})$ 的表达式中，我们得到

$$R(\mathrm{w}) = \frac{\lambda}{2}\sum_{j=1}^{d_1} \|\mathrm{W}_2(:,j)\|^2 \, \mathbb{E}[(\mathrm{W}_1(j,:)^\top \mathrm{x})^2] = \frac{\lambda}{2}\sum_{j=1}^{d_1} \|\mathrm{W}_2(:,j)\|^2 \|\mathrm{W}_1(j,:)\|^2$$

在第二个等式成立是因为假设分布是各向同性的。

□

11.3 *Landscape of the Optimization Problem*

虽然第11.2节的重点在于从结果正则化学习问题的全局最优解的角度理解dropout的隐式偏差，但在这里我们关注dropout作为优化过程的计算方面。由于dropout是一阶方法，且Dropout目标函数的景观（例如，问题11.11）高度非凸，我们可能只能希望找到一个*local*最小值，前提是问题没有退化鞍点[LSJR16, GHJY15b]。因此，在本节中，我们提出以下问题：*What is the implicit bias of dropout in terms of*

local minima? Do local minima share anything with global minima structurally or in terms of the objective? Can dropout find a local optimum?

为了分析的简便性，我们关注具有相同权重的单隐藏层 *linear* 自动编码器的情况，即 $U = V$。我们假设输入分布是对称的，即 $C_x = I$。在这种情况下，总体风险简化为

$$\begin{aligned}\mathbb{E}[\|y - UU^\top x\|^2] &= \mathrm{Tr}\,(C_y) - 2\langle C_{yx}, UU^\top\rangle + \|UU^\top\|_F^2 \\ &= \|M - UU^\top\|_F^2 + \mathrm{Tr}\,(C_y) - \|C_{yx}\|_F^2\end{aligned}$$

在 $M = \frac{C_{yx}+C_{xy}}{2}$. 忽略与权重矩阵 U 无关的项，目标是使 $L(U) = \|M - UU^\top\|_F^2$ 最小化。使用 Dropout 等于解决以下问题:

$$\min_{U\in\mathbb{R}^{d_0\times d_1}} L_\theta(U) := \|M - UU^\top\|_F^2 + \lambda \underbrace{\sum_{i=1}^{d_1}\|u_i\|^4}_{R(U)} \tag{11.11}$$

我们可以将上述问题的全局最优解描述如下。

定理11.3.1。对于任意的 $j \in [r]$，令 $\kappa_j := \frac{1}{j}\sum_{i=1}^{j}\lambda_i(C_{yx})$。此外，定义 $\rho := \max\{j \in [r] : \lambda_j(C_{yx}) > \frac{\lambda j \kappa_j}{r+\lambda j}\}$。然后，如果 U_* 是问题11.11的全局最优解，它满足 $U_*U_*^\top = \mathcal{S}_{\frac{\lambda\rho\kappa_\rho}{r+\lambda\rho}}(C_{yx})$。

接下来，很容易看出问题11.11的目标函数的梯度由以下公式给出

$$\nabla L_\theta(U) = 4(UU^\top - M)U + 4\lambda U\mathrm{diag}(U^\top U).$$

我们还在问题11.11的关键点做出了以下重要观察。引理11.3.2允许我们界定关键点的不同范数，这将在后面的证明中看到。

引理11.3.2。如果U是问题11.11的临界点，则成立$U^\top \preceq M$。

Proof of Lemma 11.3.2. 由于 $\nabla L_\theta(U) = 0$，因此我们有

$$(M - UU^\top)U = \lambda U\mathrm{diag}(U^\top U)$$

乘以两边的右侧 $U^\top$ 并重新排列得到

$$MUU^\top = UU^\top UU^\top + \lambda U\mathrm{diag}(U^\top U)U^\top \tag{11.12}$$

注意，右侧是对称的，这意味着左侧也必须是对称的，即

$$MUU^\top = (MUU^\top)^\top = UU^\top M,$$

因此，M 和 $UU^\top$ 互易。注意，在方程（11.12）中，$U\mathrm{diag}(U^\top U)U^\top \succeq 0$。因此，$MUU^\top \succeq UU^\top UU^\top$。设 $UU^\top = W\Gamma W^\top$ 为 $UU^\top$ 的紧特征分解。我们得到

$$MUU^\top = MW\Gamma W^\top \succeq UU^\top UU^\top = W\Gamma^2 W^\top.$$

从右向左分别乘以 $W\Gamma^{-1}$ 和 $W^\top$，我们得到 $W^\top MW \succeq \Gamma$，从而完成证明。 □

我们在第11.3.1节中表明，（a）问题11.11的局部最小值继承了全局最优解相同的隐含偏差，即所有局部最小值都是相等的。然后，在第11.3.2节中，我们表明对于足够小的正则化参数，（b）不存在虚假的局部最小值，即所有局部最小值都是全局的，并且（c）所有鞍点都是非退化的（参见定义11.3.4）。

11.3.1 Implicit bias in local optima

回忆起人口风险 $L(U)$ 是旋转不变的，即对于任何旋转矩阵 Q，有 $L(UQ) = L(U)$。现在，如果权重矩阵 U 不相等化，那么存在索引 i，$j \in [r]$，使得 $\|u_i\| > \|u_j\|$。我们表明，设计一个旋转矩阵（除了列 i 和 j 外处处等于单位矩阵）将质量从 u_i 移动到 u_j 是容易的，这样 UQ 对应列的范数之差严格减小，而其他列的范数保持不变。换句话说，这种旋转严格减少了正则化器，从而减少了目标。形式上，这暗示以下结果。

引理11.3.3. 问题11.11的所有局部极小值都相等，即如果U是一个局部最优解，那么$\|u_i\| = \|u_j\|\ \forall i,\ j \in [r]$。

引理11.3.3揭示了dropout的基本性质。一旦我们在隐藏层中执行dropout – *no matter how small the dropout rate* – 所有局部最小值都变得相等。我们使用图11.1中的玩具问题来说明这一点。

Proof of Lemma 11.3.3. 我们表明，如果 U 没有归一化，那么 U 的任何 ϵ-邻域都包含一个具有严格小于 $L_\theta(U)$ 的 dropout 目标的点。更正式地说，对于任何 $\epsilon > 0$，我们展示了一个旋转 Q_ϵ，使得 $\|U - UQ_\epsilon\|_F \leq \epsilon$ 和 $L_\theta(UQ_\epsilon) < L_\theta(U)$。设 U 是问题 11.11 的一个临界点，它没有归一化，即 U 存在两列具有不同的范数。不失一般性，设 $\|u_1\| > \|u_2\|$。我们设计一个旋转矩阵 Q，使其几乎是一个等距变换，但它将质量从 u_1 移动到 u_2。因此，新的因子变得“更不归一化”，并实现了更小的

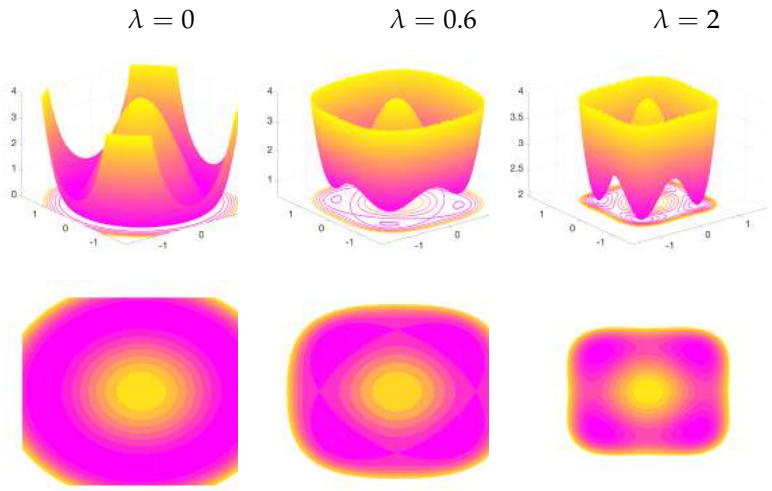


图11.1：单层线性自动编码器网络（输入和输出为一维，隐藏层宽度为r =^2，带有dropout）的优化景观（顶部）和等高线图（底部）对于不同的正则化参数λ的值。左侧：对于λ ==0，问题简化为平方损失最小化，如水平集所示，具有旋转不变性。中间：对于λ >=0，全局最优解向原点收缩。所有局部最小值都是全局的，并且相等，即权重与向量$(\pm 1, \pm 1)$平行。右侧：随着λ的增加，全局最优解进一步收缩。

正则化器，同时保留损失值。为此，定义

$$Q_\delta := \begin{bmatrix} \sqrt{1-\delta^2} & -\delta & 0 \\ \delta & \sqrt{1-\delta^2} & 0 \\ 0 & 0 & I_{r-2} \end{bmatrix}$$

并且让 $\widehat{U} := UQ_\delta$。容易验证 Q_ϵ 确实是一个旋转。首先，我们证明对于任何 ϵ，只要 $\delta^2 \leq \frac{\epsilon^2}{2\,\mathrm{Tr}(M)}$，我们就得到 $\widehat{U} \in \mathcal{B}_\epsilon(U)$：

$$\begin{aligned} \|U - \widehat{U}\|_F^2 &= \sum_{i=1}^{r} \|u_i - \widehat{u}_i\|^2 \\ &= \|u_1 - \sqrt{1-\delta^2}u_1 - \delta u_2\|^2 + \|u_2 - \sqrt{1-\delta^2}u_2 + \delta u_1\|^2 \\ &= 2(1-\sqrt{1-\delta^2})(\|u_1\|^2 + \|u_2\|^2) \\ &\leq 2\delta^2\,\mathrm{Tr}(M) \leq \epsilon^2 \end{aligned}$$

在第二个不等式之前的不等式由引理11.3.2得出，因为$\|u_1\|^2 + \|u_2\|^2 \leq \|U\|_F^2 = \mathrm{Tr}(UU^\top) \leq \mathrm{Tr}(M)$，以及$1 - \sqrt{1-\delta^2} = \frac{1-1+\delta^2}{1+\sqrt{1-\delta^2}} \leq \delta^2$这一事实。

接下来，我们证明对于足够小的 δ，L_θ 在 $\widehat{U}$ 的值严格小于 U 的值。观察如下：

$$\begin{aligned} \|\widehat{u}_1\|^2 &= (1-\delta^2)\|u_1\|^2 + \delta^2\|u_2\|^2 + 2\delta\sqrt{1-\delta^2}u_1^\top u_2 \\ \|\widehat{u}_2\|^2 &= (1-\delta^2)\|u_2\|^2 + \delta^2\|u_1\|^2 - 2\delta\sqrt{1-\delta^2}u_1^\top u_2 \end{aligned}$$

并且剩余的列不会改变，即对于 $i = 3, \cdots, r$，$\widehat{u}_i = u_i$。结合 Q_δ 保持范数的性质，即 $\|U\|_F = \|UQ_\delta\|_F$，我们得到

$$\|\widehat{u}_1\|^2 + \|\widehat{u}_2\|^2 = \|u_1\|^2 + \|u_2\|^2. \tag{11.13}$$

让 $\delta = -c \cdot \operatorname{sgn}(u_1^\top u_2)$ 对于足够小的 $c > 0$，使得 $\|u_2\| < \|\widehat{u}_2\| \leq \|\widehat{u}_1\| < \|u_1\|$。使用方程 (11.13)，这表明 $\|\widehat{u}_1\|^4 + \|\widehat{u}_2\|^4 < \|u_1\|^4 + \|u_2\|^4$，进而得到 $R(\widehat{U}) < R(U)$，因此 $L_\theta(\widehat{U}) < L_\theta(U)$。因此，非均衡临界点不能是局部最小值，从而证明了引理的第一个命题。 □

11.3.2 Landscape properties

接下来，我们描述dropout收敛到的解的特征。我们通过理解问题11.11的优化景观来实现这一点。我们分析的核心是以下关于*strict saddle property*的概念。

定义11.3.4（严格鞍点/性质）。设$f : \mathcal{U} \to \mathbb{R}$是一个二阶可微函数，设$U \in \mathcal{U}$是$f$的一个临界点。那么，如果$f$在U处的Hessian至少有一个负特征值，即$\lambda_{\min}(\nabla^2 f(U)) < 0$，则U是$f$的*strict saddle point*。此外，如果f的所有鞍点都是严格鞍点，则f满足*strict saddle property*。

严格鞍点性质确保对于任何不是局部最优的临界点U，Hessian矩阵具有显著的负特征值，这使得梯度下降（GD）和随机梯度下降（SGD）等一阶方法能够逃离鞍点并收敛到局部最小值[LSJR16, GHJY15b]。在此基础上，关于不同机器学习问题景观的研究工作如雨后春笋般涌现，包括低秩矩阵恢复[BNS16b]、广义相干重建问题[SQW16b]、矩阵补全[GLM16b]、深度线性网络[Kaw16]、矩阵感知和鲁棒PCA[GJZ17b]以及张量分解[GHJY15b]，为第一阶方法的全局最优性提供了依据。

对于没有正则化的特殊情况（即 $\lambda = 0$；等价地，没有dropout），问题11.11简化为标准的平方损失最小化，这已被证明没有虚假的局部最小值并满足严格鞍点性质（参见，例如[BH89, JGN+17]）。然而，由dropout引入的正则化可能会引入新的虚假局部最小值以及退化的鞍点。我们的下一个结果确立，至少当dropout率足够小的时候，情况并非如此。

定理11.3.5。设$r := \operatorname{Rank}(M)$。假设$d_1 \leq d_0$且正则化参数满足 $\lambda < \frac{r\lambda_r(M)}{(\sum_{i=1}^r \lambda_i(M)) - r\lambda_r(M)}$。那么对于问题11.11，有

1. 所有局部极小值都是全局的，

2. 所有鞍点都是严格鞍点。

一些评论是恰当的。首先，假设 $d_1 \leq d_0$ 决非限制性，因为网络图 $UU^\top \in \mathbb{R}^{d_0 \times d_0}$ 的秩至少为

大多数 d_0，并且让 $d_1 > d_0$ 不增加由网络表示的函数类的表达能力。其次，定理11.3.5保证任何不是全局最优的临界点U是一个严格的鞍点，即 $\nabla^2 L(\mathrm{U}, \mathrm{U})$ 有一个负特征值。这个性质允许使用一阶方法，如dropout，逃离这样的鞍点。第三，注意定理11.3.5中的保证在正则化参数 λ 足够小的情况下成立。这类假设在文献中很常见（例如，参见[GJZ17b]）。虽然这是定理11.3.5结果的一个 *sufficient* 条件，但尚不清楚它是否是 *necessary*。

Proof of Theorem 11.3.5. 这里我们概述了定理11.3.5证明的主要步骤。

1. 在引理11.3.3中，我们证明了非归一化临界点集合不包含任何局部最优解。此外，引理11.3.6表明所有这些点都是严格鞍点。2. 在引理11.3.7中，我们给出了所有归一化临界点的闭式特征，这些特征是关于M的特征分解。然后我们证明，如果选择λ适当，则所有不是全局最优的临界点都是严格鞍点。3. 由第1项和第2项可知，如果选择λ适当，则所有不是全局最优的临界点都是严格鞍点。

□

引理11.3.6. *All critical points of Problem 11.11 that are not equalized, are strict saddle points.*

Proof of Lemma 11.3.6. 通过引理11.3.3，非等化临界点集不包含任何局部最优解。我们证明所有这样的点都是严格鞍点。设U为一个非等化的临界点。为了证明U是一个严格鞍点，只需证明Hessian矩阵有一个负特征值。在这里，我们展示了一条曲线，其沿该曲线的二阶方向导数为负。假设，不失一般性，$\|\mathrm{u}_1\| > \|\mathrm{u}_2\|$，并考虑以下曲线

$$\Delta(t) := [(\sqrt{1-t^2}-1)\mathrm{u}_1 + t\mathrm{u}_2, (\sqrt{1-t^2}-1)\mathrm{u}_2 - t\mathrm{u}_1, \mathrm{o}_{d,r-2}]$$

对于任何 $t \in \mathbb{R}$，由于 $\mathrm{U} + \Delta(t)$ 实质上是在 U 上的旋转，并且 L 在旋转下是不变的，因此很容易检查 $L(\mathrm{U} + \Delta(t)) = L(\mathrm{U})$。

观察发现

$$
\begin{aligned}
&g(t) := L_\theta(\mathrm{U} + \Delta(t)) \\
&= L_\theta(\mathrm{U}) + \|\sqrt{1-t^2}\mathrm{u}_1 + t\mathrm{u}_2\|^4 - \|\mathrm{u}_1\|^4 + \|\sqrt{1-t^2}\mathrm{u}_2 - t\mathrm{u}_1\|^4 - \|\mathrm{u}_2\|^4 \\
&= L_\theta(\mathrm{U}) - 2t^2(\|\mathrm{u}_1\|^4 + \|\mathrm{u}_2\|^4) + 8t^2(\mathrm{u}_1\mathrm{u}_2)^2 + 4t^2\|\mathrm{u}_1\|^2\|\mathrm{u}_2\|^2 \\
&\qquad + 4t\sqrt{1-t^2}\mathrm{u}_1^\top\mathrm{u}_2(\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2) + O(t^3).
\end{aligned}
$$

g 的导数随后给出为

$$
\begin{aligned}
g'(t) = &-4t(\|\mathrm{u}_1\|^4 + \|\mathrm{u}_2\|^4) + 16t(\mathrm{u}_1\mathrm{u}_2)^2 + 8t\|\mathrm{u}_1\|^2\|\mathrm{u}_2\|^2 \\
&+ 4(\sqrt{1-t^2} - \frac{t^2}{\sqrt{1-t^2}})(\mathrm{u}_1^\top\mathrm{u}_2)(\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2) + O(t^2).
\end{aligned}
$$

由于 U 是一个临界点且 L_θ 是连续可微的，因此应该满足

$$
g'(0) = 4(\mathrm{u}_1^\top\mathrm{u}_2)(\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2) = 0.
$$

由于根据假设 $\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2 > 0$，因此应该有 $\mathrm{u}_1^\top\,\mathrm{u}_2 = 0$。我们现在考虑二阶方向导数：

$$
\begin{aligned}
g''(0) &= -4(\|\mathrm{u}_1\|^4 + \|\mathrm{u}_2\|^4) + 16(\mathrm{u}_1\mathrm{u}_2)^2 + 8\|\mathrm{u}_1\|^2\|\mathrm{u}_2\|^2 \\
&= -4(\|\mathrm{u}_1\|^2 - \|\mathrm{u}_2\|^2)^2 < 0
\end{aligned}
$$

这完成了证明。 □

我们现在关注于被归一化的关键点，即满足 $\nabla L_\theta(\mathrm{U}) = 0$ 且 $\mathrm{diag}(\mathrm{U}^\top\mathrm{U}) = \frac{\|\mathrm{U}\|_F^2}{d_1}\mathrm{I}$ 的点 U。

引理11.3.7. *Let* $r := \mathrm{Rank}(M)$. *Assume that* $d_1 \le d_0$ *and* $\lambda < \frac{r\lambda_r}{\sum_{i=1}^r(\lambda_i - \lambda)}$. *Then all equalized local minima are global. All other equalized critical points are strict saddle points.*

Proof of Lemma 11.3.7. 设 U 为一个等化的临界点。此外，设 r' 为 U 的秩，且 $\mathrm{U} = \mathrm{W}\Sigma\mathrm{V}^\top$ 是其秩-r' SVD，即 $\mathrm{W} \in \mathbb{R}^{d_0\times r'}$，$\mathrm{V} \in \mathbb{R}^{d_1\times r'}$ 满足 $\mathrm{U}^\top\mathrm{U} = \mathrm{V}^\top\mathrm{V} = \mathrm{I}_{r'}$，并且 $\Sigma \in \mathbb{R}^{r'\times r'}$ 是一个正定对角矩阵，其对角元素按降序排列。我们有：

$$
\begin{aligned}
&\nabla L_\theta(\mathrm{U}) = 4(\mathrm{U}\mathrm{U}^\top - \mathrm{M})\mathrm{U} + 4\lambda\mathrm{U}\mathrm{diag}(\mathrm{U}^\top\mathrm{U}) = \mathrm{o} \\
\Longrightarrow\ &\mathrm{U}\mathrm{U}^\top\mathrm{U} + \frac{\lambda\|\mathrm{U}\|_F^2}{d_1}\mathrm{U} = \mathrm{M}\mathrm{U} \\
\Longrightarrow\ &\mathrm{W}\Sigma^3\mathrm{V}^\top + \frac{\lambda\|\Sigma\|_F^2}{d_1}\mathrm{W}\Sigma\mathrm{V}^\top = \mathrm{M}\mathrm{W}\Sigma\mathrm{V}^\top \\
\Longrightarrow\ &\Sigma^2 + \frac{\lambda\|\Sigma\|_F^2}{d_1}\mathrm{I} = \mathrm{W}^\top\mathrm{M}\mathrm{W}
\end{aligned}
$$

由于上述等式的左边是对角线，这意味着 $W \in \mathbb{R}^{d_0 \times r'}$ 对应于 M 的某些 r' 特征向量。令 $\mathcal{E} \subseteq [d_0]$，$|\mathcal{E}| = r'$ 表示存在于 W 中的 M 的特征向量集合。上述等式等价于以下线性方程组：

$$(I + \frac{\lambda}{d_1} 11^\top)\mathrm{diag}(\Sigma^2) = \vec{\lambda},$$

在 $\vec{\lambda} = \mathrm{diag}(W^\top MW)$ 的地方。上述线性方程组的解由以下给出

$$\mathrm{diag}(\Sigma^2) = (I - \frac{\lambda}{d_1 + \lambda r'})\vec{\lambda} = \vec{\lambda} - \frac{\lambda \sum_{i=1}^{r'} \lambda_i}{d_1 + \lambda r'} 1_{r'}. \tag{11.14}$$

因此，集合 $\mathcal{E}$ 属于以下类别之一：

0. $\mathcal{E} = [r']$, $r' > \rho$

1. $\mathcal{E} = [r']$, $r' = \rho$

2. $\mathcal{E} = [r']$, $r' < \rho$

3. $\mathcal{E} \neq [r']$

我们在此对上述分区进行逐案分析。
案例0。[$\mathcal{E} = [r'], r' > \rho$]。我们证明$\mathcal{E}$不能属于此类，即当$\mathcal{E} = [r']$时，应满足$r' \leq \rho$。为了证明这一点，考虑方程(11.14)中的第$r'$个线性方程：

$$\sigma_{r'}^2 = \lambda_{r'} - \frac{\lambda \sum_{i=1}^{r'} \lambda_i}{d_1 + \lambda r'}.$$

自从秩U {v*}，因此 $\sigma_{r'} > 0$，这又反过来意味着

$$\lambda_{r'} > \frac{\lambda \sum_{i=1}^{r'} \lambda_i}{d_1 + \lambda r'} = \frac{\lambda r' \kappa_{r'}}{d_1 + \lambda r'}.$$

它来自定理11.3.1中ρ的最大性，得出$r' \leq$。 ρ.
案例1. [$\mathcal{E} = [r'], r' = \rho$] 当W对应于M的前-$\rho$特征向量时，我们检索到由定理11.3.1描述的全局最优解。因此，所有这些临界点都是全局最小值。
案例2. [$\mathcal{E} = [r'], r' < \rho$] 设$W_{d_0} := [W, W_\perp]$ 是M的对应于M的降序特征值的完备特征基，其中$W_\perp \in \mathbb{R}^{d_0 \times d_0 - r'}$构成W的正交子空间的基。对于秩亏的M，$W_\perp$包含M的零空间，因此对应于M的零特征值的特征向量。同样，设$V_\perp \in \mathbb{R}^{d_1 \times d_1 - r'}$张成V的正交子空间，使得$V_{d_1} := [V, V_\perp]$ 形成基$\mathbb{R}^{d_1}$的正交基。注意，由于$r' \leq \min\{d_0, d_1\}$，$W_\perp$和$V_\perp$都是良好定义的。定义

$\mathrm{U}(t) = \mathrm{W}_{d_0}\Sigma'\mathrm{V}_{d_1}^\top$ 其中 $\Sigma' \in \mathbb{R}^{d_0 \times d_1}$ 是对角线且对角线元素非零，给定为 $\sigma_i' = \sqrt{\sigma_i^2 + t^2}$ 对于 $i \le d_1$。观察可知

$$\mathrm{U}(t)^\top \mathrm{U}(t) = \mathrm{V}\Sigma^2\mathrm{V}^\top + t^2\mathrm{V}_{d_1}^\top\mathrm{V}_{d_1} = \mathrm{U}^\top\mathrm{U} + t^2\mathrm{I}_{d_1}.$$

因此，参数曲线 $\mathrm{U}(t)$ 对所有 t 进行了均衡。在 $\mathrm{U}(t)$ 处的群体风险等于：

$$\begin{aligned} L(\mathrm{U}(t)) &= \sum_{i=1}^{d_1}(\lambda_i - \sigma_i^2 - t^2)^2 + \sum_{i=d_1+1}^{d_0}\lambda_i^2 \\ &= L(\mathrm{U}) + d_1 t^4 - 2t^2\sum_{i=1}^{d_1}(\lambda_i - \sigma_i^2). \end{aligned}$$

此外，由于 $\mathrm{U}(t)$ 被归一化，我们得到正则化器的以下形式：

$$\begin{aligned} R(\mathrm{U}(t)) &= \frac{\lambda}{d_1}\|\mathrm{U}(t)\|_F^4 = \frac{\lambda}{d_1}\left(\|\mathrm{U}\|_F^2 + d_1 t^2\right)^2 \\ &= R(\mathrm{U}) + \lambda d_1 t^4 + 2\lambda t^2\|\mathrm{U}\|_F^2. \end{aligned}$$

定义 $g(t) := L(\mathrm{U}(t)) + R(\mathrm{U}(t))$。我们有

$$g(t) = L(\mathrm{U}) + R(\mathrm{U}) + d_1 t^4 - 2t^2\sum_{i=1}^{d_1}(\lambda_i - \sigma_i^2) + \lambda d_1 t^4 + 2\lambda t^2\|\mathrm{U}\|_F^2.$$

它很容易验证 $g'(0) = 0$。此外，g 在 $t = 0$ 处的二阶导数给出如下：

$$g''(0) = -4\sum_{i=1}^{d_1}(\lambda_i - \sigma_i^2) + 4\lambda\|\mathrm{U}\|_F^2 = -4\sum_{i=1}^{d_1}\lambda_i + 4(1+\lambda)\|\mathrm{U}\|_F^2 \tag{11.15}$$

我们使用 $\|\mathrm{U}\|_F^2 = \sum_{i=1}^{r'}\sigma_i^2$ 和方程（11.14）得出

$$\|\mathrm{U}\|_F^2 = \mathrm{tr}\Sigma^2 = \sum_{i=1}^{r'}(\lambda_i - \frac{\lambda\sum_{j=1}^{r'}\lambda_j}{d_1 + \lambda r'}) = (\sum_{i=1}^{r'}\lambda_i)(1 - \frac{\lambda r'}{d_1 + \lambda r'}) = \frac{d_1\sum_{i=1}^{r'}\lambda_i}{d_1 + \lambda r'}$$

将上述等式（11.15）代入，我们获取

$$g''(0) = -4\sum_{i=1}^{d_1}\lambda_i + 4\frac{d_1 + d_1\lambda}{d_1 + \lambda r'}\sum_{i=1}^{r'}\lambda_i = -4\sum_{i=r'+1}^{d_1}\lambda_i + 4\frac{(d_1 - r')\lambda}{d_1 + \lambda r'}\sum_{i=1}^{r'}\lambda_i$$

要得到U为严格鞍点的充分条件，它足够

fices that $g''(t)$ be negative at $t = 0$, 即

$$\begin{aligned} g''(0) < 0 &\implies \frac{(d_1 - r')\lambda}{d_1 + \lambda r'} \sum_{i=1}^{r'} \lambda_i < \sum_{i=r'+1}^{d_1} \lambda_i \\ &\implies \lambda < \frac{(d_1 + \lambda r') \sum_{i=r'+1}^{r} \lambda_i}{(d_1 - r') \sum_{i=1}^{r'} \lambda_i} \\ &\implies \lambda (1 - \frac{r' \sum_{i=r'+1}^{d_1} \lambda_i}{(d_1 - r') \sum_{i=1}^{r'} \lambda_i}) < \frac{d_1 \sum_{i=r'+1}^{d_1} \lambda_i}{(d_1 - r') \sum_{i=1}^{r'} \lambda_i} \\ &\implies \lambda < \frac{d_1 \sum_{i=r'+1}^{d_1} \lambda_i}{(d_1 - r') \sum_{i=1}^{r'} \lambda_i - r' \sum_{i=r'+1}^{d_1} \lambda_i} \\ &\implies \lambda < \frac{d_1 h(r')}{\sum_{i=1}^{r'} (\lambda_i - h(r'))} \end{aligned}$$

在 $h(r') := \frac{\sum_{i=r'+1}^{d_1} \lambda_i}{d_1 - r'}$ 是尾特征值 $\lambda_{r'+1}, \ldots, \lambda_{d_1}$ 的平均值的情况下。容易看出，由于 $h(r')$ 随着 r' 单调递减，因此右手边随着 r' 单调递减。因此，只需确保对于 $r' = r - 1$ 的选择，λ 小于右手边就足够了，其中 $r := \mathrm{Rank}(\mathrm{M})$。也就是说，$\lambda < \frac{r\lambda_r}{\sum_{i=1}^{r}(\lambda_i - \lambda_r)}$。

案例3。[$\mathcal{E} \neq [r']$] 我们表明所有这样的临界点都是严格的鞍点。设w'是W中缺失的顶r'个特征向量之一。设$j \in \mathcal{E}$使得w_j不属于M的前r'个特征向量。对于任意的$t \in [0, 1]$，设$\mathrm{W}(t)$在所有列中都与W相同，但在第j^{th}列中，其中$\mathrm{w}_j(t) = \sqrt{1 - t^2}\mathrm{w}_j + t\mathrm{w}'$。注意，对于所有$t$的值，$\mathrm{W}(t)$仍然是一个正交矩阵。定义参数化曲线$\mathrm{U}(t) := \mathrm{W}(t)\Sigma\mathrm{V}^\top$对于$t \in [0, 1]$并观察：

$$\begin{aligned} \|\mathrm{U} - \mathrm{U}(t)\|_F^2 &= \sigma_j^2 \|\mathrm{w}_j - \mathrm{w}_j(t)\|^2 \\ &= 2\sigma_j^2 (1 - \sqrt{1 - t^2}) \leq t^2 \operatorname{Tr} \mathrm{M} \end{aligned}$$

这意味着对于任何 $\epsilon > 0$，存在一个 $t > 0$，使得 $\mathrm{U}(t)$ 属于 U 的 ϵ-球内。我们证明 $L_\theta(\mathrm{U}(t))$ 严格小于 $L_\theta(\mathrm{U})$，这意味着 U 不能是局部极小值。注意，这种 $\mathrm{U}(t)$ 的构造保证了 $R(\mathrm{U}') = R(\mathrm{U})$。特别是，很容易看出 $\mathrm{U}(t)^\top \mathrm{U}(t) = \mathrm{U}^\top \mathrm{U}$，因此 $\mathrm{U}(t)$ 对于所有 t 的值都保持相等。此外，我们还有

$$\begin{aligned} &L_\theta(\mathrm{U}(t)) - L_\theta(\mathrm{U}) = \|\mathrm{M} - \mathrm{U}(t)\mathrm{U}(t)^\top\|_F^2 - \|\mathrm{M} - \mathrm{U}\mathrm{U}^\top\|_F^2 \\ &= -2 \operatorname{Tr}(\Sigma^2 \mathrm{W}(t)^\top \mathrm{M} \mathrm{W}(t)) + 2 \operatorname{Tr}(\Sigma^2 \mathrm{W}^\top \mathrm{M} \mathrm{W}) \\ &= -2\sigma_j^2 t^2 (\mathrm{w}_j(t)^\top \mathrm{M} \mathrm{w}_j(t) - \mathrm{w}_j^\top \mathrm{M} \mathrm{w}_j) < 0, \end{aligned}$$

在最后一个不等式成立，因为根据构造，$\mathrm{w}_j(t)^\top \mathrm{M} \mathrm{w}_j(t) > \mathrm{w}_j^\top \mathrm{M} \mathrm{w}_j$。定义 $g(t) := L_\theta(\mathrm{U}(t)) = L(\mathrm{U}(t)) + R(\mathrm{U}(t))$。为了证明这样的鞍点是非退化的，只需证明 $g''(0) < 0$。

它很容易验证在原点处的二阶方向导数由以下公式给出

$$g''(0) = -4\sigma_j^2(\mathrm{w}_j(t)^\top \mathrm{M}\mathrm{w}_j(t) - \mathrm{w}_j^\top \mathrm{M}\mathrm{w}_j) < 0,$$

这完成了证明。 □

11.4 Role of Parametrization

对于最小二乘线性回归（即，对于问题11.8中的 $k = 1$ 和 $\mathrm{u} = \mathrm{W}_1^\top \in \mathbb{R}^{d_0}$ ），我们可以证明使用dropout相当于解决以下正则化问题：

$$\min_{\mathrm{u}\in\mathbb{R}^{d_0}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathrm{u}^\top \mathrm{x}_i)^2 + \lambda \mathrm{u}^\top \widehat{\mathrm{C}}\mathrm{u}.$$

所有上述问题的极小化器都是以下线性方程组的解（$(1+\lambda)\,\mathrm{X}^\top\mathrm{X}\mathrm{u} = \mathrm{X}^\top\mathrm{y}$，其中$\mathrm{X} = [\mathrm{x}_1, \cdots, \mathrm{x}_n]^\top \in \mathbb{R}^{n\times d_0}$，$\mathrm{y} = [y_1, \cdots, y_n]^\top \in \mathbb{R}^{n\times 1}$ 分别是设计矩阵和响应向量。与Tikhonov正则化不同，它产生线性方程组（$\mathrm{X}^\top\mathrm{X} + \lambda\mathrm{I})\,\mathrm{u} = \mathrm{X}^\top\mathrm{y}$（一个有用的先验，即使它们表现出良好的可区分性，也会丢弃解释数据中小方差的方向），dropout正则化仅表现为参数的缩放。这表明参数化在确定结果正则化的性质中起着重要作用。然而，对于深度线性网络[MA19]，已经证明了由于dropout引起的数据相关正则化仅导致参数的缩放。同时，在矩阵感知的情况下，我们看到更丰富的正则化类。一个可能的解释是，在线性网络的情况下，我们需要网络中的卷积结构以产生丰富的归纳偏差。例如，矩阵感知可以写成以下卷积形式的两层网络：

$$\langle \mathrm{U}\mathrm{V}^\top, \mathrm{A}\rangle = \langle \mathrm{U}^\top, \mathrm{V}^\top\mathrm{A}^\top\rangle = \langle \mathrm{U}^\top, (\mathrm{I}\otimes\mathrm{V}^\top)\mathrm{A}^\top\rangle.$$

11.4.1 Related Work

12

SDE approximation of SGD and its implications

在第二章中，梯度下降收敛性的分析假设学习率 η 被设置得足够小，以至于损失在每个迭代中都会减少。因此，如果将 η 取为零，优化仍然可以很好地工作。当 $\eta \to$ 为 0 时，轨迹变得连续，这个过程被称为 *gradient flow* (GF)，由以下微分方程给出

$$\frac{dx}{dt} = -\nabla f(x) \quad \textit{(Gradient Flow)}.$$

如果我们试图理解模型参数在训练过程中的演变，梯度流是最自然的过程来分析，就像我们在第8章所做的那样。在第2章，我们也研究了随机梯度下降法（SGD），它可能更有效，因为它可以使用随机小批次的梯度噪声估计。但是，如果SGD中的$\eta \to$为0，那么更新再次是无穷小，因此随机梯度平均后等于真实梯度。所以当$\eta \to$为0时，GD和SGD都简化为GF。

但是，正如我们在后续章节中讨论的那样，这三种算法在泛化方面可能会有非常不同的行为。换句话说，轨迹很重要。然而，在极限 $\eta \to 0$ 时，它们变得相同！本章提出了一种将 SGD 模型为具有无穷小步骤的过程的方法：*stochastic differential equa- tions, or SDEs*。这些方程使用来自随机过程的噪声进行增强。我们将看到，它们为大型批量训练提供了所谓的缩放规则；理解 SGD 可能如何快速逃离鞍点1以及归一化网络的范数动力学。

works2. 近期，这些技术已被应用于研究
自适应优化算法，例如RMSprop和Adam，并为它们推导缩放规则。

在这个部分，我们使用 x 作为参数向量，而不是我们通常的 w，因为在SDE文献中它更标准。

1 胡文清，李军驰，李磊，刘建国. 关于非凸随机梯度下降的扩散近似. *Annals of Mathematical*, 4(1), 2019

Sciences and Applications

2 李志远，刘开丰，和Sanjeev Arora。将现代深度学习与传统优化分析相协调：内在学习率。在Hugo Larochelle，Marc' Aurelio Ranzato，Raia Hadsell，Maria-Florina Balcan，和Hsuan-Tien Lin，编辑，*Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*

12.1 Understanding gradient noise in SGD

假设有 m 个训练点，并令 $f_i(x)$ 表示在示例 i 上的损失函数，当 x 表示模型参数时。在 SGD 中，我们采样一个大小为 B 的批次来计算 f_1，... ，f_B，并使用批次梯度 $\nabla f^{(B)}(x) = \frac{1}{B}\sum_{i=1}^{B}\nabla f_i(x)$ 来更新参数。因此，

实际上，在每次迭代中，训练数据被随机划分为批次，而不是为每次梯度估计抽取一个全新的随机批次这种统计上正确的方法。

噪声在此估计中为 $z^{(B)} = \nabla f^{(B)}(x) - \nabla f(x) = \frac{1}{B}\sum_{i=1}^{B} z_i$。显然，它具有零均值，这意味着随机梯度的期望是真实梯度 $\overline{\nabla} = \frac{1}{m}\nabla f_i(x)$。

第2.5.3节中SGD的收敛性分析还假设了方差的幅度的上界是已知的。但如前所述，除了噪声的幅度外，噪声的性质对于良好的泛化4似乎也很重要，我们现在

例如，计算完整的梯度并向其中添加均匀高斯噪声并不像SGD中的噪声那样具有相同的益处。

尝试更好地理解它。如果一个向量随机变量 z 的均值为零，其 *covariance* 矩阵是期望 $\mathbb{E}[zz^T]$。这是其分布“形状”的一个度量。

斌石，魏杰·苏，和迈克尔·乔丹。关于学习率和薛定谔算子。*arXiv preprint*，*arXiv:2004.06977*，2020；以及斯蒂芬·曼特，马修·D·霍夫曼，和大卫·M·布莱。随机梯度下降作为近似贝叶斯推理。*The Journal of Machine Learning Research*，18(1): 4873–4907，2017

问题 12.1.1. *Show that the covariance matrix of* $z^{(B)}(x)$ *is*

$$\Sigma^{(B)}(x) = \frac{1}{B}\mathop{\mathbb{E}}_{i}(\nabla f_i(x) - \overline{\nabla})(\nabla f_i(x) - \overline{\nabla})^T.$$

噪声 $z^{(B)}$ 是零均值的，但其协方差 $\Sigma^{(B)}(x)$ 依赖于当前参数，并且与 B 成反比。为了强调这一点，我们使用 $\Sigma(x)$ 表示与 $B = 1$ 的协方差，从而得到 $\Sigma^{(B)}(x) = \frac{1}{B}\Sigma(x)$。我们抽象出这个梯度公式，它隐含地假设了一个损失函数，并突出了噪声的形状和尺度的重要性。

定义12.1.2（具有缩放的噪声梯度算子）。*A noisy gradient oracle with scale parameter* $\sigma > 0$ *(NGOS) takes a parameter* x *and returns a stochastic gradient* $g = \nabla f(x) + \sigma z$*, where* z *is drawn from a mean-zero distribution* $\mathcal{Z}_\sigma(x)$ *with covariance* $\Sigma(x)$*.* $\mathcal{Z}_\sigma(x)$ *can change with* σ*, but* $\Sigma(x)$ *must remain fixed.*

多GPU并行性的改进鼓励从业者尝试较大的批量大小：如果架构可以处理批量大小 B，那么训练时间（保持epoch数量不变）按 $1/B$ 的比例缩放。这是一个吸引人的想法，但会遇到泛化误差随着 B 增加的问题。显然，噪声协方差的幅度在泛化中起着重要作用，就像形状一样。以下在5中使用，以提高有效尺度

5 Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, 和 Kaiming He. 准确、大批量的SGD：1小时内训练ImageNet。*arXiv preprint*，2017. *arXiv:1706.02677*

噪声6并且结果证明它相当保留了泛化误差大批量（尽管当批量太大时也会失败）。稍后我们将从数学上证明它。

定义12.1.3（线性缩放规则）。*When running SGD with batch size* $B' = \kappa B$*, use learning rate* $\eta' = \kappa\eta$*.*

6 线性缩放规则似乎仅适用于SGD，不适用于自适应梯度方法等。它也适用于极大规模的批处理大小，这已经在数学上得到了研究。

12.1.1 Motivating example: Loss with Fixed Gradient

我们现在考虑一个简单的情况，其中梯度在每个时间步长都固定为$\bar{g}$，NGOS 具有各向同性噪声：$g_k \sim \mathcal{N}(\bar{g}, \sigma^2 I)$。然后，经过 k 步的 SGD，$x_k \sim \mathcal{N}(-\eta\bar{g}k, \eta^2\sigma^2 k)$。如果 SGD 轨迹存在连续近似，那么 x_k 必须能够通过固定时间 t 的连续轨迹值来近似，独立于 σ (，即，无论批大小)。为了防止 x_k 的分布随 σ 变化，我们必须调整 η 和 k，使得 $\eta \sim 1/\sigma^2$ 和 $k \sim 1/\eta$。第一个观察结果得出线性缩放规则，这可以通过注意到 $\sigma \sim 1/\sqrt{B}$ 来看出。后一个观察结果促使对 $t \sim k\eta$ 进行连续时间缩放。然而，我们注意到，缩放规则和连续时间尺度只能通过正式展示相应 SDE 公式的质量来严格证明，如定理 12.3.4 所示。为了更严格地进行，我们必须现在精确描述连续轨迹近似离散轨迹的含义。

12.2 Stochastic processes: Informal Treatment

随机过程可以被视为随机游走的连续时间版本。

定义12.2.1（莱维过程）。*A stochastic process $W = \{W_t : t \geq 0\}$ is a Lévy process if it satisfies the following properties.* [7]

1. $W_0 = 0$ *almost surely*
2. *Continuity: For any $\epsilon > 0$ and $t \geq 0$, $\lim_{h\to 0} \Pr[|W_{t+h} - W_t| > \epsilon] = 0$.*
3. *Stationary increments: For $s < t$, the distribution of $W_t - W_s$ is equal to the distribution of W_{t-s}.*
4. *Independent increments: for every $t > 0$, future increments $W_{t+\delta} - W_t$ for $\delta \geq 0$ are independent of past values W_s for $s \leq t$.*
5. *W_t is continuous in t.*[8]

定义12.2.2（维纳过程）。[9] *A Wiener process in $\Re^d$ is a Lévy process that has Gaussian increments: $W_{t+\delta} - W_t \sim \mathcal{N}(0, \delta I_{d\times d})$ for $\delta \geq 0$.*

7 将 t 视为时间。

8 即，W_t的轨迹没有间断。

9 威纳过程也称为 *Brownian motion*，爱因斯坦用它来解释悬浮在流体中的微小粒子的观察到的运动。整体轨迹是由分子子碰撞引起的非常微小的随机方向的运动所导致的。

请注意，由这些过程的一次运行定义的轨迹在正常意义上是不可微分的。*Ito Calculus*提供了一种定义某种导数的方法，表示为dW_t，其从时间0到T的积分是W_T。我们省略了细节，因为下面将不需要它们。

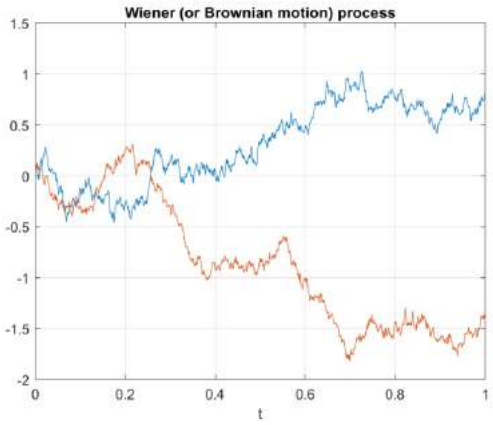


图12.1：从同一起始点生成的两个一维维纳过程轨迹。它们进行独立的随机移动并迅速分离。

一个 *Stochastic Differential Equation* 的形式为

$$dx_t = \mu(x_t, t)dt + \sigma(x_t, t)dW_t, \quad (SDE) \tag{12.1}$$

dW_t表示维纳过程，$\mu()$和$\sigma()$同时依赖于当前时间和位置x_t。启发式解释如下：

在一个小时间间隔 $[t, t+\delta]$ 内，过程根据均值为 $\mu(x_t, t)\delta$ 和协方差矩阵 $\sigma(x_t, t)^2\delta I$ 的高斯分布随机移动。

示例12.2.3（时间变化）。*Suppose we change the scale of time, so that the new time τ is t/a. How does this change the equation? Using the above intuition, it becomes*

$$dx_\tau = a\mu(x_\tau, \tau)d\tau + \sqrt{a}\sigma(x_\tau, \tau)dW_\tau.$$

The fundamental reason is that the Weiner process (being a geometric random walk) only goes a distance proportional to $\sqrt{\delta}$ in time δ.

12.2.1 SDEs and SGD

最简单的用于在损失 $f()$ 上建模 SGD 的 SDE 是

$$dx_t = -\eta\nabla f(x_t) + \eta\sigma dW_t \tag{12.2}$$

在 dW_t 是标准维纳过程，η、σ 是固定常数的情况下。这是将批梯度中的噪声建模为均匀高斯10。定义12.1.2中NGOS的更现实的SDE是

$$dx_t = -\eta\nabla f(x) + \eta\sigma\Sigma(x)^{1/2}dW_t \tag{12.3}$$

通过12.2.3例子的推理，这相当于

10 本诗，魏杰苏，迈克尔·乔丹。关于学习率和薛定谔算子。*arXiv preprint, arXiv:2007.xxxxx*，2020。

以及斯蒂芬·曼特，马修·D·霍夫曼，和大卫·M·布莱。随机梯度下降作为近似贝叶斯推理。*The Journal of Machine Learning Research*，18(1): 4873–4907，2017

$$dx_t = -\nabla f(x) + \sqrt{\eta}\sigma\Sigma(x)^{1/2}dW_t \quad \text{(SGD的规范SDE)} \tag{12.4}$$

这稍微好一些（尽管一开始有点令人不安），因为学习率 η 并不在梯度之前出现；仅在噪声项中。

问题12.2.4. *Define the loss function $f(x) = x^2 + 9$, and let $f_1(x) = (x-3)^2$, $f_2(x) = (x+3)^2$, and $f_3(x) = x^2 + 9$. Each iteration of SGD will uniformly sample f_1, f_2, or f_3 and use the corresponding gradient to compute the next iterate with learning rate η: $x_{k+1} = x_k - \eta\nabla f_i(x)$, where $i \sim \mathcal{U}(\{1, 2, 3\})$. Write down the precise SDE approximation for the trajectory SGD takes in this setting.*

注意，我们已通过启发式方法定义了标准的SDE近似；我们没有证明这种连续近似实际上跟踪相应的SGD。尽管如此，我们现在可以给出LSR的非正式理由，我们将在定理12.3.5中稍后尝试使其更加严谨。

非正式的线性缩放规则的理由：[11] *Canonical SDE captures generalization properties. If we simultaneously scale the batch size B and learning rate η in SGD by a factor κ then the noise scale σ changes by $1/\sqrt{\kappa}$ and thus the Canonical SDE does not change.*

11 S Jastrz ębski, Z Kenton, D Arpit, N Ballas, A Fischer, Y Bengio, 和 A Storkey. 影响SGD中最小值的三种因素 *ICANN*，2018

12.3 Notion of closeness between stochastic processes

SGD和(12.4)中相应的SDE都是随机过程，一个是离散的，另一个是连续的。每个都诱导参数空间轨迹上的一个分布。我们将使用k表示离散时间步长，t表示连续时间。根据我们上面的讨论，如果SDE轨迹演化时间为T，则这对应于SGD中的$K = \lfloor T/\eta_e \rfloor$个离散步长。12 对应轨迹涉及参数向量的序列，用 $\{x_k^{\eta_e}\}_{k=0}^K$ 表示 SGD，用 $\{X\}_T$ 表示 SDE13。

尽管对于SGD有 $\eta_e = \eta$，我们使用 η_e 而不是 η，因为不同的优化算法可能需要不同的连续时间缩放（例如，第12.4节）。

两个随机过程接近意味着什么？我们需要在它们诱导的参数向量上的分布之间给出一个 *distance* 的公式（如第14章和第16章所述）。衡量接近程度的一种常见方法是比较两个分布上某些 *test functions* 的期望差异14。对于例-

充足的自然测试函数在我们的上下文中是训练网络的 *test error*。

13 集合表示法捕捉到我们正在讨论由随机种子驱动的、对于固定的 η_e 选择所决定的随机轨迹族，但我们可以在不失一般性的情况下，交替讨论这个族和单个轨迹。

第14节 第16章中的 *discriminator net* 是一个测试函数的例子，我们发现在那里，测试函数的类别可以在分布空间上定义运输度量。

定义12.3.1（分布之间的距离）。*For a function class F, define the distance between distributions $\{\widetilde{X}_k\}$ and $\{x_k\}$ as*

$$d_F(\{x_k\}, \{\widetilde{X}_k\}) = \sup_{f\in F} \left| \mathop{\mathbb{E}}_{X\sim\{\widetilde{X}_k\}} f(X) - \mathop{\mathbb{E}}_{x\sim\{x_k\}} f(x) \right|$$

完全通用的测试函数在这种情况下没有意义，因为它们可以将分布中的微小差异放大为期望中的较大差异。为了避免这种情况，我们将函数类限制为最多多项式增长15。类G的

连续函数$\mathbb{R}^d \to \mathbb{R}$在$\forall g \in G$存在正整数$\kappa_1, \kappa_2 > 0$的条件下具有*polynomial growth*。对于$\alpha \in \mathbb{N}^+$，我们用G^α表示α次连续可微的函数g的集合，其中所有形式为$\frac{\partial^{\vec{\alpha}} g}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}$的偏导数也属于$G$。我们可以通过取位置距离的最大值来扩展标准定义的位置分布之间的距离到整个轨迹分布。

15 深度学习相关的测试函数，如泛化误差，在整个参数向量空间上可能不是多项式增长。但当我们应用定义来衡量SDE和SGD的接近程度时，我们只对出现在训练轨迹上的参数向量感兴趣，在这些参数向量上，泛化误差可能表现得更好。

定义12.3.2（轨迹距离）。*The trajectory distance over a finite number of steps $K > 0$ between a discrete trajectory $\{x\}_K$ and the corresponding rescaled continuous trajectory $\{\widetilde{X}\}_K$ under a class of test functions G is*

$$D_G(\{x\}_K, \{\widetilde{X}\}_K, K) = \max_{k=0,\ldots,K} d_G(\{x_k\}, \{\widetilde{X}_k\})$$

我们可以因此定义一个关于*weak approximation*的概念，它保证了对于一类至多多项式增长的测试函数类，最大区分期望的上界。

定义12.3.3（阶-α弱逼近）。*We say $\{X\}_t$ and $\{x\}_k$ are order-α weak approximations*[16] *of each other if for every test function $g \in G2$, there exists a constant $C > 0$ independent of η_e such that*

$$D_G(\{x\}_k, \{\widetilde{X}\}_k, T) \leq C\eta_e^\alpha$$

16 钱晓莉，程泰，和 E 魏南。随机修正方程和随机梯度算法的动力学 i：数学基础。*J. Mach. Learn. Res.*，20:40–1，2019

12.3.1 Formal Approximation

现在我们解释在什么意义上

定理12.3.4（SDE是SGD的一阶弱近似）。
Assume the NGOS and loss function f satisfy

1. *$\nabla f(x)$ is Lipschitz and $\mathcal{C}^\infty$-smooth*
2. *All partial derivatives of ∇f and $\Sigma^{1/2}$ up to and including the 4th order have polynomial growth*
3. *Low skewness: there exists a function $K(x)$ of polynomial growth independent of σ such that $|\mathbb{E}_{z\sim\mathcal{Z}_\sigma(x)}[z^{\otimes 3}]| \leq K(x)/\sigma$ for all $x \in \mathbb{R}^d$ and all noise scales σ.*

4. *Bounded moments: for all integers $m \geq 1$ and all noise scales σ, there exists a constant C_{2m} independent of σ such that $\mathbb{E}_{z\sim\mathcal{Z}_\sigma(x)}[\|z\|_2^{2m}]^{\frac{1}{2m}} \leq C_{2m}(1+\|x\|_2)$ for all $x \in \mathbb{R}^d$.*

Let $\{x\}_K$ be a family of discrete SGD trajectories with learning rate η and $\{X\}_T$ be the corresponding family of SDE trajectories given by Equation (12.4). Then, $\{x\}_K$ and $\{X\}_T$ are order-1 weak approximations (Definition 12.3.3) of each other for any $T > 0$ and with $\eta_e = \eta$.[17]

17 钱晓莉，程泰，和魏文安。随机修正方程与随机梯度算法的动力学 i：数学基础。*J. Mach. Learn. Res.*，20:40–1，2019

标准化网络可能违反梯度的Lipschitz条件，因为导数是无界的，但如果轨迹远离原点和无穷大，则条件仍然满足。低偏度条件要求NGOs具有小的三阶矩，而有界矩条件确保NGOs不是重尾分布。这两个条件共同允许将随机噪声建模为维纳过程。上述定理可以扩展以证明LSR的有效性。

定理12.3.5（线性缩放规则的有效性）。*Let $\{x\}_K^{(B)}$ be a family of discrete SGD trajectories with batch size B and learning rate η and $\{x\}_{\lfloor K/\kappa \rfloor}^{(\kappa B)}$ be the family with batch size κB and learning rate $\kappa\eta$. Furthermore, define the time-rescaled discrete trajectory $\{\widetilde{x}\}_K^{(\kappa B)}$ where $\{\widetilde{x}_k\}^{(\kappa B)} = \{x_{\lfloor k/\kappa \rfloor}\}^{(\kappa B)}$. Then, if $\{x\}_K^{(B)}$ and $\{\widetilde{x}\}_K^{(\kappa B)}$ have the same initial condition, for any g with at most polynomial growth and any number of time steps $K > 0$,*

$$M_g(\{x\}_K^{(B)}, \{\widetilde{x}\}_K^{(\kappa B)}, T) = C(1+\kappa)\eta$$

Proof. 协方差的线性意味着通过 κ 缩放批次大小仅通过缩放 σ 的 $1/\sqrt{\kappa}$ 来修改 NGOS。因此，根据 LSR 修改超参数时，SDE 保持不变。SDE 对 SGD 的弱近似以 η 为条件，由于 η 在 LSR 中通过 κ 缩放，因此相同的方法给出了 $C\kappa\eta$ 的上界。我们还考虑了 $\kappa < 1$ 的情况，从而得到了 $C(1+\kappa)\eta$ 的界限。

□

通过证明机制，我们看到当SDE近似成立时，线性缩放规则成立。当伊藤SDE近似失败时（例如，当噪声分布违反高斯类似假设18时），LSR也可能成立。如果$(1+\kappa)$是

被视为常数时，我们恢复与之前相同的弱近似。当 κ 变得很大时，界限变得宽松，实际上，我们观察到对于非常大的批次，线性缩放规则会失效。19

18 李志远，马莎迪卡·马拉迪，和桑杰夫·阿罗拉。关于使用随机微分方程（SDEs）建模SGD的有效性。*Advances in Neural Information Processing Systems*，34，2021

19 Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 准确、大批量的SGD：1小时内训练ImageNet。*arXiv preprint*，2017 *arXiv:1706.02677*，2017及Zhiyuan Li，Sadhika Malladi和Sanjeev Arora。关于用随机微分方程（SDEs）建模SGD的有效性。*Advances in Neural Information Processing Systems*，34，2021

12.3.2 *Proof Sketch*

为简化符号，我们省略了描述两条轨迹位置分布的集合符号，改用 x_k 和 $\widetilde{X}_k$。我们遵循两个主要步骤来证明定理12.3.4中的弱近似成立。首先，我们将一个 T 长度的时间区间划分为一系列单步区间。我们定义 $x_k(x、k_0)$ 为时间 k 处的离散轨迹值，初始条件为 $x_{k_0} = x$，并对 X_t 和 $\widetilde{X}_k$ 做同样的处理。在此符号下，$\widetilde{X}_k(x_k、k) = x_k、$（即为经过 k 步后的 SGD，以及 $)、0) = \widetilde{X}_k = X_{k\eta_e}$（即为经过 $k\eta_e$ 持续时间后的 SDE。我们从 SGD 轨迹开始，按顺序用从相应初始条件开始的 SDE 轨迹替换每个区间，如图12.2所示。

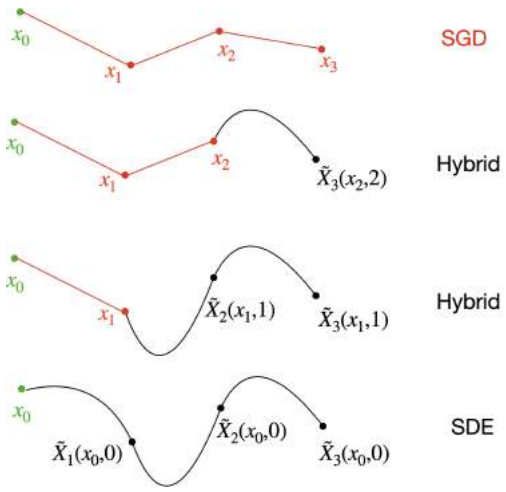


图12.2：示例在SGD和SDE之间插值的混合轨迹。

这些混合轨迹对任何$1 \leq k \leq K$产生以下误差分解。

$$|\mathbb{E}[g(x_k)] - \mathbb{E}[g(\widetilde{X}_k)]| \leq \sum_{j=0}^{k-1} \left|\mathbb{E}[g(\widetilde{X}_k(x_{j+1}, j+1))] - \mathbb{E}[g(\widetilde{X}_k(x_j, j))]\right|$$

问题 12.3.6. *Show that the above error decomposition holds.*

因此，整个时间间隔内的误差与单步误差的总和有关。

第二步是通过泰勒展开来展示近似单步误差足够小。定义离散轨迹的初始条件为 x 的单步移动为 $\Delta(x) = x_1 - x$，连续轨迹为 $\widetilde{\Delta}(x) = \widetilde{X}_1 - x$。离散轨迹的泰勒展开是直接的：

$$\mathbb{E}[g(x+\Delta)] = g(x) + \langle \mathbb{E}[\Delta], \nabla g(x)\rangle + \mathbb{E}\left[\left\langle \frac{\Delta\Delta^\top}{2}, \nabla^2 g(x)\right\rangle\right] + \dots$$

泰勒展开 $g(x+\widetilde{\Delta})$ 时，我们引入了来自随机微积分的一个关键技术工具 *Itô's Lemma*，它也被称为标准链式法则的随机对应物。

定义12.3.7（伊藤引理）。*For a general Itô SDE* $\mathrm{d}X_t = b(X_t)\mathrm{d}t + \sigma(X_t)\mathrm{d}W_t$*, where* W_t *is a Wiener process, and a twice differentiable function* $h : \mathbb{R}^d \to \mathbb{R}$*,*

$$\mathrm{d}h(X_t) = \langle \nabla h(X_t), b(X_t)\rangle \mathrm{d}t + \langle \nabla h(X_t), \sigma(X_t)\mathrm{d}W_t\rangle + \frac{1}{2}\operatorname{Tr}[\nabla^2 h(X_t)\sigma^\top(X_t)\sigma(X_t)]\mathrm{d}t$$

前两项是标准的微积分链式法则，最后一项是一个校正项，用于考虑随机性。我们省略了随机泰勒展开的完整计算，但请注意，在这一步中利用了NGOS条件，以表明在两个泰勒展开中的高阶项都很小。

现在，我们可以比较泰勒展开式，以显示单步近似误差为 $O(\eta_e^2)$，由于有 T/η_e 个区间，因此在一个有限时间区间 T 上的近似误差为 $O(\eta_e)$，正如所期望的那样。

12.4 Stochastic Variance Amplified Gradient (SVAG)

定义12.3.3中的近似误差界限依赖于一个小的学习率。然而，现实中的深度网络通常使用较大的学习率进行训练，尤其是在遵循LSR并使用大批量时，因此不清楚SDE近似是否适用于实际设置。直接模拟SDE（例如，通过标准离散化方法，如Euler-Maruyama）在计算上是不可行的，因为它需要反复计算梯度协方差 $\Sigma(X_t)$ 和完整梯度 $\nabla f(X_t)$ 以获得细粒度间隔。在本节中，我们讨论了一种计算高效的SDE模拟方法：SVAG.[20]

[20] 李志远，马莎迪卡·马拉迪，和桑杰夫·阿罗拉。关于使用随机微分方程（SDEs）建模SGD的有效性。*Advances in Neural Information Processing Systems*，34，2021

定义12.4.1（SVAG算法）。*For a given NGOS* g*, learning rate* η*, and a chosen hyperparameter* $\ell > 0$*, the SVAG algorithm computes the stochastic gradient as*

$$\widehat{g} = \frac{1+\sqrt{2\ell-1}}{2} g_{\gamma_1} + \frac{1-\sqrt{2\ell-1}}{2} g_{\gamma_2}$$

where g_γ *indicates a stochastic gradient sampled with* γ *as the random seed, and uses learning rate* η/ℓ *with the same update rule as SGD.*

SVAG算法可以通过采样两个批次并计算如上所示的加权平均损失来实现。因为学习率按$1/\ell$缩放，我们必须运行SVAG ℓ 步以近似 η 连续时间（即，

SDE value corresponding to a single discrete step of SGD). We can formal ly show that the SVAG trajectory is a weak approximation of the SDE for SGD with $\eta_e = \eta/\ell$. 对SGD单个离散步骤的SDE值）。我们可以形式上证明SVAG轨迹是SGD的SDE的弱近似，其中包含$\eta_e = \eta/\ell$。

定理12.4.2（SVAG算法逼近SDE）。*Let $\{X\}_T$ be the SDE for SGD with hyperparameter η, and let $\{x\}_K$ be the analogous SVAG trajectory with hyperparameter ℓ. Furthermore, define $\{\widetilde{X}\}_K$ such that $\{\widetilde{X}_k\} = \{X_{k\eta/\ell}\}$ and set $\eta_e = \eta/\ell$. Assume conditions 1, 2, and 4 hold from Theorem 12.3.4. Then, for any test function $g \in G^4$ and finite time interval $T > 0$:*

$$D_G(\{x\}_K, \{\widetilde{X}\}_K, T) \leq C\eta/\ell$$

Proof. 证明依赖于展示如果定理12.3.4中的条件1、2和4成立，则应用SVAG算法将导致满足条件3的NGOS。在所有条件都满足的情况下，可以将SVAG算法视为具有较小学习率的SGD，并保证NGOS满足噪声分布的类似高斯和非重尾假设。因此，我们可以直接应用SGD与相应SDE之间的标准近似定理（即定理12.3.4），从而得出SVAG是SDE对SGD的一阶弱近似。

□

我们在此处指出，该界限为 η/ℓ，因此可以通过增加 ℓ 来使其对于固定的 η 变得很小。增加 ℓ 需要采取更多的梯度步来模拟SDE，但发现SVAG轨迹似乎收敛，并且对于可计算的 ℓ 值，通常与SDE相匹配。[21]

问题 12.4.3. *Let $\widehat{\mathcal{Z}}_{\ell\sigma}(x)$ be the distribution of*

$$\widehat{z} = \frac{1}{\ell}\left(\frac{1+\sqrt{2\ell-1}}{2}z_1 + \frac{1-\sqrt{2\ell-1}}{2}z_2\right)$$

Let $\widehat{g}$ be defined as in Definition 12.4.1. Show that $\widehat{g}$ has the same distribution as $\nabla f(x) + \ell\sigma\widehat{z}$.

21 李志远，马莎迪卡·马拉达，和桑杰夫·阿罗拉。关于使用随机微分方程（SDEs）建模SGD的有效性。*Advances in Neural Information Processing Systems*，34，2021

13
Effect of Normalization in Deep Learning

大约在2014年，由于无法使网络更深，建立在新深度学习成功（如AlexNet）之上的努力受到了阻碍。训练过于挑剔，往往无法大幅降低损失。Ioffe和Szegedy 1引入了*Batch Normalization*，一种归一化的形式

层级参数，它们发现可以使训练速度提高10倍，并且也提高了泛化能力。从那时起，已经发明了其他相关方法，包括*Layer Normalization* [2] 和*Group Normalization* [3]. 今天，大多数深度架构都利用某种形式的归一化。

在这一章中，您会很快注意到——例如，定理13.3.1——归一化导致现代训练与传统分析优化不兼容，这些分析在第二章和其他章节中已进行概述。本章介绍了考虑归一化的新分析。关键的新数学概念是 *scale invariance*。

1 S Ioffe 和 C Szegedy. 批标准化：通过减少内部协变量偏移来加速深度网络训练。*ICML*，2015

2 J Ba, J R Kiros 和 G E Hinton. 层归一化。*NeurIPS*，2016

3 Y Wu 和 K He. 群归一化。*ECCV*，2017

13.1 Warmup Example: How Normalization Helps Optimization

由于深度网络被认为对于它们所执行的任务过度参数化，网络原则上可以通过多种方式实现所需的输入输出行为。让我们考虑一个简单的场景，具有一维不可分数据集 $\{(x_i, y_i): i = 1, ..., n\}$，其中 $x_i \in \Re$，$y_i \in \{+1, -1\}$。标准逻辑损失 $\widehat{\ell}(W)$ 将是 $\sum_i \log(1 + \exp(-W x_i y_i))$。假设我们过度参数化它，允许 k 变量 $(w_1, ..., w_k)$ 和

$$\ell(w_1, \ldots, w_k) = \widehat{l}\left(\prod_{i=1}^{k} w_i\right) = \sum_i \log(1 + \exp(-x_i y_i \prod_{i=1}^{k} w_i)).$$

这与标准损失在逻辑上是等价的，但在梯度下降的行为上并不等价。GD使用固定的学习率快速优化原始损失，而这里发生的是以下情况。

问题 13.1.1. *Suppose at initialization, all w_i's are the same, $\prod_i w_i = W_0$ and k is even. Show that if learning rate $\eta > \frac{2}{|\nabla \widehat{l}(W_0)|}|W_0|^{1/k-1}$ and $W_0 > W^*$ where $W^* > 0$ is the minimizer of $\widehat{l}$, then $\prod_i w_i$ will monotonically increase (i.e., explode).*

此示例（也可参见4中的更不平凡的示例1.2）说明了深度网络可能导致梯度产生大数值，从而复杂化训练过程。

4 Z Li, S Bhojanapalli, M Zaheer, Reddi S, 和 Kumar S. 使用尺度不变架构的神经网络的鲁棒训练。*arxiv*，2022

定义13.1.2（同质性度）。*A function $f : \Re^d \to \Re^{d'}$ has* 同质性度 *k if for all $c > 0$ and all x, $f(c \cdot x) = c^k f(x)$.*
We also call such functions k-homogeneous.

问题 13.1.3. *Show that if any mapping from parameters to outputs with inputs fixed, $f : \Re^d \to \Re^{d'}$, is computed by a feed-forward net with depth k and only ReLU gates with zero bias* [5] *then it has degree of homogeneity k.*

[5] In other words, ReLU(z) $= \max\{0, z\}$.

在一个高k的k-同质网络中，参数的微小变化可能导致梯度和对Hessian的巨大波动。归一化方案可以减少这种影响。

13.2 Normalization schemes and scale invariance

归一化可以以多种方式进行，以下变体可能是最容易理解的。

Layer normalization: Let a_i 表示在通常的前馈深度网络（可能包含卷积）中某个层的 i 个输入坐标，该网络具有固定的训练数据点。层归一化将首先计算 $\mu = \frac{1}{H}\sum_i a_i$ 和 $\sigma^2 = \frac{1}{H}\sum_i (a_i - \mu)^2$，从而改变这种架构。层归一化层的 i 个输出坐标定义为

$$\text{LayerNorm}(a)_i = \gamma_i \cdot \frac{a_i - \mu}{\sigma} + \beta_i,$$

在 γ_i、β_i 是与该层归一化层相关的可学习参数。

Group normalization 这是一种对层归一化的推广，其中统计量 μ 和 σ 只允许为层的子组计算。*Batch normalization* 与层归一化类似，除了平均 μ 和方差 σ^2 是相对于当前训练批次中的所有数据点在节点上计算的。

通常不清楚如何在网络包含归一化后分析优化。Arora等人6的论文提出了一种
通过识别一个称为 *scale invariance* 的属性来找到前进的道路。

[6] S Arora, Z Li, and K Lyu. Theoretical analysis of auto rate-tuning by batch normalization. *ICLR*, 2019

定义13.2.1（尺度不变性）。*A function* $f: \Re^d \to \Re^{d'}$ *is* 尺度不变 *if* $f(c \cdot x) = f(x)$ *for all* $c > 0$. [7]

[7] 这意味着它的同质性程度为零。

注意，如果 h_1、h_2 是 k-齐次的，那么 $h_1 + h_2$ 和 ReLU(h_1) 也是，而 h_1/h_2 是尺度不变的。

引理13.2.2. *In the above description of layer normalization, if* w *denotes the parameter vector for the entire network, then for each layer* $l \geq 1$*, the output of ReLU,* $x^{(l)}$ *has degree of homogeneity* 1 *with respect to* w*, where* L *is the total number of layers,* $x^{(0)}$ *is the input of the network and* $x^{(l)} := ReLU(LayerNorm(W^{(l-1)}x^{(l-1)}))$ *for* $1 \leq l \leq L$*.*

If the parameters after the last normalization, $\gamma_i^{(L-1)}$, $\beta_i^{(L-1)}$, $W^{(L)}$ *are fixed during training then the function computed is scale-invariant.*

Proof. 证明是通过层高的归纳进行的，l。回想一下，层是从计算前一层输出的线性函数开始的。因此，在上面的描述中，每个$x_i^{(l)}$所表示的函数的齐次度为1（除了$x_i^{(0)}$，它是0齐次的），$\mu^{(l)}$和$\sigma^{(l)}$也是如此。归一化值$(x_i^{(l)} - \mu^{(l)})/\sigma^{(l)}$随后是尺度不变的。然而，$\gamma_i^{(l)}(x_i^{(l)} - \mu^{(l)})/\sigma^{(l)} + \beta_i^{(l)}$再次是1齐次的，并且通过ReLU后仍然保持1齐次。这完成了归纳。我们得出结论，如果前一层输出是1齐次的，那么下一层的输出也是1齐次的。

□

一个简单的修复使网络具有尺度不变性：在最后归一化后，随机修复训练开始时的参数 $\gamma_i^{(L-1)}$，$\beta_i^{(L-1)}$，$W^{(L)}$。然后按常规训练。根据上述引理，训练损失相对于网络参数具有尺度不变性。8中的实验表明，修复顶层不会损害分类

[8] S Arora, Z Li, 和 K Lyu. 批标准化自动速率调整的理论分析。*ICLR*，2019

阳离子精度等。虽然上面我们关注的是简单的架构，但基本思想可以修改以展示包括ResNets和语言模型在内的大多数已知深度架构的尺度不变性；参见9，10。

[9] Z Li 和 S Arora. 深度学习的指数学习率调度。*ICLR*，2019

[10] Z Li，S Bhojanapalli，M Zaheer，Reddi S，和Kumar S. 使用尺度不变架构进行神经网络鲁棒训练。*arxiv*，2022

在本章的其余部分，我们在证明优化收敛速度的同时假设尺度不变性。

引理13.2.3（尺度不变函数的性质）。*If* L *is scale-invariant then the following hold:*

1. $\langle w, \nabla L(w) \rangle = 0$.
2. $\nabla L(c \cdot w) = \frac{1}{c}\nabla L(w)$.
3. $\nabla^2 L(c \cdot w) = \frac{1}{c^2}\nabla^2 L(w)$.

Proof. 1) 随后通过对 $L(c \cdot w) = L(w)$ 关于 c 求导并设置 $c = 1$。2) 随后对 $L(c \cdot w) = L(w)$ 关于 w 求梯度，(3) 随后进行两次求导。

□

13.3 Exponential learning rate schedules

通常训练深度网络涉及仔细的学习率调整，学习率在训练过程中逐渐降低。然而，在过去的几年中，一些异类的学习率计划，如 *cosine*，已被成功使用。这看起来最初像是一个谜。然而，可以证明，某些在经典观点中无意义的特定学习计划在归一化网络中变得有效。我们描述了11号的一个结果，即使提高学习率

11 Z Li 和 S Arora. 深度学习的指数学习率调度。*ICLR*，2019

指数速率（即，在每次迭代中将 η 乘以 $(1+c)$ 对于某些 $c>0$）至少与常规训练一样强大。

这发生是因为在实践中，归一化通常与 *weight decay (WD)* 和 *momentum* 一起使用。为了简化，我们忽略了动量（参见上述论文以获取完整分析）。使用学习率（LR）η 和 WD参数 λ 的基本更新如下，其中 $\nabla L_t(\cdot)$ 表示在第 t 次迭代中使用的迷你批次的梯度：

$$w_{t+1} \leftarrow (1-\eta\lambda)w_t - \eta\nabla L_t(w_t). \quad \text{(GD + WD)} \tag{13.1}$$

现在我们展示了一个基于GD的替代算法，它可以达到相同的效果，但其LR在每个迭代中增加一个乘法因子（$1+\alpha$）。这表明指数增长的LR至少与标准的GD+WD训练一样有效。

定理13.3.1. *If training loss is scale-invariant, the effect of (13.1) for T steps can also be obtained by the following alternative protocol:*

$$\widehat{w}_{t+1} \leftarrow \widehat{w}_t - \eta_t\nabla L_t(\widehat{w}_t). \tag{13.2}$$

with learning rate at step t being $\eta_t = (1-\eta\lambda)^{-(2t+1)}$.

Proof. Let w_t 表示GD + WD经过 t 步后的参数向量，$\widehat{w}_t$ 表示我们替代协议经过 t 步后的参数向量。

我们通过归纳法证明 $\widehat{w}_t = w_t/(1-\eta\lambda)^t$ 成立。[12] 这对于 $t=0$ 由设计决定。假设它对 t 成立，(13.2) 给出

12 回想一下，损失对参数向量的缩放是不变的。

$$\widehat{w}_{t+1} \leftarrow \frac{w_t}{(1-\eta\lambda)^t} - \frac{\eta}{(1-\eta\lambda)^{2t+1}}\nabla L_t\left(\frac{w_t}{(1-\eta\lambda)^t}\right).$$

该命题通过引理13.2.3部分2简化为 $(1-\eta\lambda)^{-(t+1)}w_{t+1} \leftarrow (1-\eta\lambda)^{-(t+1)}\left((1-\eta\lambda)w_t - \eta\nabla L_t(w_t)\right)$，从而完成归纳。

□

13.4 Convergence analysis for GD on Scale-Invariant Loss

现在我们分析GD +WD（13.1）在尺度不变损失 $L(\cdot)$ 上的收敛速度。基本迭代是

$$w_{t+1} = (1-\eta\lambda)w_t - \eta\nabla L(w_t). \tag{13.3}$$

这里我们将单位范数向量 $w/\|w\|_2$ 记为 $\overline{w}$。

标准收敛分析如第2.5.2节定理2.5.1中所述，由于几个原因而不可行。首先，引理13.2.3的第2部分表明，使梯度范数变小并不一定意味着低损失或甚至接近局部最优：增加参数向量的尺度会减小梯度但不会影响损失。其次，由于平滑性（即Hessian矩阵的最大特征值）在参数向量向原点移动时变得无界，因此不能使用平滑性的倒数（即LR）来设置。我们提出了固定LR（取自第13章）的第一个收敛分析。

在此设置中，这还有一个额外的好处，即表明初始化的规模并不重要——正如人们直观上在尺度不变设置中预期的那样。

13 Z Li, S Bhojanapalli, M Zaheer, Reddi S , 和 Kumar S. 使用尺度不变架构进行神经网络鲁棒训练。*arxiv*，2022

定义13.4.1. $\rho = \max_{w:\|w\|_2=1} \|\nabla^2 L(w)\|_2 = \max_w \|\nabla^2 L(\overline{w})\|_2$。

定理13.4.2（主要）。对于$\eta\lambda < \frac{1}{2}$，存在$t \leq \frac{1}{2\lambda\eta}\left(\left|\ln \frac{\|w_0\|_2^2}{\rho\pi^2\eta}\right| + 3\right)$使得$\|\nabla L(\overline{w}_t)\|_2^2 \leq 8\pi^4\rho^2\lambda\eta$。14

14 迭代次数仅与 $\|w_0\|$ 成对数关系，突显了归一化如何使优化对初始化的规模具有相当强的鲁棒性。

为了理解上述定理保证的范上界是否有意义，我们试图理解各种量的规模。

引理13.4.3. *1.* $\|\nabla L(w)\| \leq \pi\rho$ *for all* w *of unit* ℓ_2 *norm.*

2. $L(w) - \min_w L(w) \leq \pi^2\rho/2$ *for all* $w \neq 0$.

Proof. 第1部分：设w^*是单位球上L的任意局部极小值。设γ：[0, 1] $\to \Re^d$为单位球上的测地线曲线，其端点为$\gamma(0) = w^*$和$\gamma(1) = w$。我们知道s的长度至多为π，因此

$$\|\nabla L(\gamma(1))\|_2 = \|\int_{t=0}^{1} \nabla^2(L(\gamma(t)))\frac{d\gamma}{dt}dt\|_2 \leq \int_{t=0}^{1} \|\nabla^2(L(\gamma(t))\|_2\|\frac{d\gamma}{dt}\|_2 dt \leq \pi\rho.$$

第二部分类似，留作练习。

□

问题 13.4.4. *Prove part 2 of Lemma 13.4.3.*

定理13.4.2保证算法可以快速找到一个解，其中$\|\nabla L(\overline{w})\|_2$至多是单位球上最大可能值的$O(\sqrt{\lambda\eta})$因子。这在实践中是有意义的，因为$\lambda\eta$非常小，就像$10\wedge{-4} \sim 10\wedge{-6}$。

引理13.4.5. *A twice-differential scale-invariant function* $L : \Re^d \to \Re$ *with* $\rho = \max_{\|x\|_2=1} \|\nabla^2 L(x)\|_2$ *satisfies for every pair of orthogonal vectors* x, v

$$L(x+v) - L(x) \leq \langle v, \nabla L(x)\rangle + \frac{\rho\|v\|_2^2}{2\|x\|_2^2}.$$

Proof. 定义一个函数 $\gamma : [0, 1] \to \Re^d$ 为 $\gamma(s) = x + s \cdot v$ 和 $F(s) := L(\gamma(s))$。然后 $L(\gamma(0)) = L(x)$ 和 $L(\gamma(1)) = L(x + v)$。通过泰勒展开和介值定理 $F(1) = F(0) + F'(0) + F"(s^*)/2$ 对于某个 $s^* \in [0, 1]$。此外，$F'(0) = \langle \nabla L(x), v \rangle$ 和尺度不变性意味着：

$$F''(s^*) = \gamma'(s^*)\nabla^2(\gamma(s^*))\gamma'(s^*) \leq \frac{\rho}{\|\gamma(s^*)\|_2^2}\|\gamma'(s^*)\|_2^2.$$

现在通过注意到 $\gamma'(s^*) = v$ 和 $\|\gamma(s^*)\|_2 \geq \|x\|_2^2$ 由于正交性而得出这个词元。 □

下一定理使用梯度的范数的平方来界定损失的变动，并且与第2.5.2节中更简单设置中的类似界限类似。

定理13.4.6. *If w_{t+1}, w_t are as in (13.3) and $\eta\lambda \leq 1/2$ then*

$$L(w_t) - L(w_{t+1}) \geq \eta(1 - \frac{2\rho\eta}{\|w_t\|_2^2})\|\nabla L(w_t)\|_2^2.$$

问题 13.4.7. *Prove Theorem 13.4.6 from Lemma 13.4.5. (Hint: Use* $(1 - \eta\lambda)w_t$*) as x and $-\eta\nabla L(w_t)$ as v.)*

如前所述，当 w 接近零向量时，$\nabla^2 L(w)$ 可能会爆炸。因此，分析必须根据 $\|w_0\|_2$ 分离出两种情况。首先，我们展示如果初始范数太小，那么它会迅速变得足够大，使得引理13.4.9中的论点适用。

引理13.4.8. *In any sequence of $\frac{1}{6\lambda\eta}$ successive iterations there must exist some step T where $\|w_T\|_2^2 \geq \pi^2\rho\eta$ or $\|\nabla L(\overline{w}_T)\|_2^2 \leq 8\pi^4\rho^2\lambda\eta$. Furthermore, $\|w_T\|_2^2 \leq \frac{2(\pi^2\rho\eta)^2}{\|w_0\|_2^2}$.*

Proof. 只要 $\|w_t\|_2^2 \leq \pi^2\rho\eta$ 和 $\|\nabla L(\overline{w}_t)\|_2^2 \geq 8\pi^4\rho^2\lambda\eta$，那么利用勾股定理以及 $\nabla L(w_t)$ 与 w_t 垂直的事实，可以得出结论

$$\|w_{t+1}\|_2^2 - (1 - \eta\lambda)^2\|w_t\|_2^2 = \eta^2\|\nabla L(w_t)\|_2^2. \tag{13.4}$$

这导致

$$\|w_{t+1}\|_2^2 - \|w_t\|_2^2 \geq \eta^2\|\nabla L(\overline{w}_t)\|_2^2/\|w_t\|_2^2 - 2\eta\lambda\|w_t\|_2^2 \geq 6\pi^2\rho\lambda\eta^2.$$

这些不等式在 t 上的求和表明，左边是 $\|w_t\|^2 - \|w_0\|^2$，它最多是 $\pi^2\rho\eta$。另一方面，右边与 t 线性相关，即 $6t\pi^2\rho\lambda\eta^2$。我们得出结论 t 不能超过 $1/(6\lambda\eta)$。因此，在这个之前必须有一个第一个 T。

点 $\|w_T\|_2^2 > \pi^2\rho\eta \geq \|w_{T-1}\|_2^2$。再次应用勾股定理，我们有

$$\begin{aligned}\|w_T\|_2^2 \leq & \|w_{T-1}\|_2^2 + \eta^2\|\nabla L(\overline{w}_{T-1})\|_2^2/\|w_{T-1}\|_2^2 \\ \leq & \pi^2\rho\eta + \eta^2\|\nabla L(\overline{w}_{T-1})\|_2^2/\|w_0\|_2^2 \\ \leq & \frac{2(\pi^2\rho\eta)^2}{\|w_0\|_2^2},\end{aligned}$$

这给出了对 $\|w_T\|_2^2$ 的所需上界。 □

利用前一个引理，我们可以关注初始范数足够大的情况。

引理13.4.9. *For $\eta\lambda < \frac{1}{2}$, if $\|w_0\|_2^2 > \pi^2\rho\eta$ and $T_0 = \frac{1}{2\eta\lambda}\ln\frac{2\|w_0\|_2^2}{\rho\pi^2\eta}$ then some $t < T_0$ must satisfy*

$$\|\nabla L(\overline{w}_t)\|_2^2 \leq 8\pi^4\rho^2\lambda\eta. \tag{13.5}$$

Proof. 首先，我们证明存在某个 $T \leq T_0$ 使得 $\|w_T\|_2^2 \leq \pi^2\rho\eta$。否则，使用（13.4）进行简单归纳给出

$$\begin{aligned}\|w_{T_0}\|_2^2 - (1-\eta\lambda)^{2T_0}\|w_0\|_2^2 = & \sum_{t=0}^{T_0-1}\eta^2(1-\eta\lambda)^{2(T_0-t)}\|\nabla L(w_t)\|_2^2 \\ \leq & \sum_{t=0}^{T_0-1}\frac{\eta^2}{2}\|\nabla L(w_t)\|_2^2.\end{aligned}$$

求和定理13.4.6中证明的基本不等式在$t = 0$到$T_0 - 1$上，以及假设$\|w_t\| \geq \pi^2\rho\eta$一起表明右侧被上界$\eta(L(w_0) - L(w_{T_0}))$所限制，$\eta(L(w_0) - L(w_{T_0}))$至多为$\pi^2\eta\rho/2$。最后，通过选择$T_0$，我们有$(1 - \eta\lambda)^{2T_0}\|w_0\|_2^2 < \pi^2\eta\rho/2$。将此代入$T_0$的表达式中，我们得出$\|w_{T_0}\|_2^2 \leq \pi^2\eta\rho$。矛盾！因此，必须有一个第一步$T \leq T_0$，其中$\|w_T\|_2^2 \leq \pi^2\eta\rho < \|w_{T-1}\|_2^2$。（注意：$T \geq 0$，因为初始化时范数超过$\pi^2\eta\rho$。）由于$\|w_T\|_2 \geq (1-\eta\lambda)\|w_{T-1}\|_2$，使用（13.4）我们得出

$$\begin{aligned}\|\nabla L(\overline{w}_{T-1})\|_2^2 \leq & \eta^{-2}\left(\|w_T\|_2^2 - (1-\eta\lambda)^2\|w_{T-1}\|_2^2\right)\|w_{T-1}\|_2^2 \\ \leq & \eta^{-2}\cdot 2\lambda\eta\|w_T\|_2^2\cdot\frac{\|w_T\|_2^2}{(1-\eta\lambda)^2} \\ \leq & 8\pi^4\rho^2\lambda\eta.\end{aligned}$$

这表示对 $\|\nabla L(\overline{w}_{T-1})\|_2$ 所需的上界。 □

主定理，定理13.4.2，通过直接结合引理13.4.8和13.4.9得到证明。{v*}

14
Unsupervised learning: Distribution Learning

本书至今主要关注监督学习——即训练数据集由数据点和表示它们所属类别的标签组成，模型必须学习在给定输入的情况下产生正确的标签。本章是关于无监督学习的介绍，其中随机采样了数据点但没有标签或类别。我们概述了这种学习形式的可能目标，然后重点介绍 *distribution learning*，它解决了许多这些目标。

14.1 Possible goals of unsupervised learning

Learn hidden/latent structure of data. 一个例子是 *Principal Component Analysis (PCA)*，关注于在数据中找到最重要的方向。结构学习的其他例子可以包括稀疏编码（也称为字典学习）或非负矩阵分解（NMF）。

Learn the distribution of the data. 一个经典例子是皮尔逊在1893年通过对马耳他岛蟹类种群数据的研究，对进化理论的贡献。生物学家在野外采样了1000只螃蟹，并为每只测量了23个属性（例如，长度、重量等）。假设这些数据点应该表现出高斯分布，但皮尔逊无法找到一个与高斯分布的良好拟合。然而，他能够证明分布实际上是两个高斯分布的 *mixture*。因此，该种群由两个不同的物种组成，这两个物种在进化术语中并未分离太久。

通常，在密度估计中，假设未标记的数据集由来自固定分布的独立同分布样本组成，模型 θ 学习某些分布 $p_\theta(\cdot)$ 的表示，该分布将概率 $p_\theta(x)$ 分配给数据点 x。这是 *density estimation.* 的一般问题

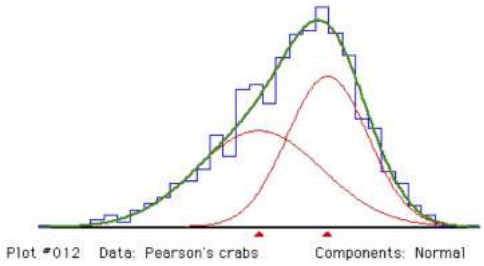


图14.1：Pearson的蟹数据作为两个高斯分布混合的可视化。（来源：麦马斯特大学MIX主页。）

一种密度估计的形式是学习一个 *generative model*，其中学习到的分布具有形式 $p_\theta(h, x)$，其中 x 是可观测的（即数据点）和 h 由一个隐藏变量向量组成，通常称为 *latent* 变量。然后 x 的密度分布为 $\int p_\theta(h, x)dh$。在蟹的例子中，分布是高斯混合 $\mathcal{N}(\mu_1, \Sigma_1)$，$\mathcal{N}(\mu_2, \Sigma_2)$，其中第一个贡献 ρ_1 的样本分数，其他贡献 $1 - \rho_1$ 的分数。然后 θ 向量由两个高斯参数以及 ρ_1 组成。可见部分 x 由蟹的属性向量组成。隐藏向量 h 由一个位组成，指示这个 x 是由哪个高斯生成的，以及生成 x 的高斯随机变量的值。

Learning good representation/featurization of data 例如，图像的像素表示可能在其他任务中不太有用，人们可能希望有一个更“高级”的表示，以便以数据高效的方式解决下游任务。人们希望使用未标记的数据学习这样的特征化。

在某些设置中，特征化是通过生成模型学习的：假设一个数据分布 $p_\theta(h、x)$ 如上所述，并且假设可见样本点 x 的特征化是用于生成它的隐藏变量 h。更确切地说，隐藏变量是条件分布 $p(h|x)$ 的一个样本。这种表示学习的观点在后面描述的 *autoencoders* 中被使用。

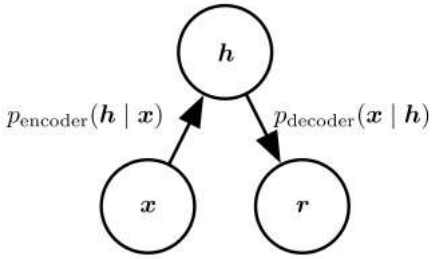


图14.2：使用密度分布 $p(h, x)$ 定义的自动编码器，其中 h 是对应于可见向量 x 的潜在特征向量。给定 x 计算出 h 的过程称为“编码”，其逆过程称为“解码”。通常，在 x 上应用编码器然后解码器不会再次给出 x，因为组合变换是从一个分布中抽取的样本。

例如，主题模型是文本的简单概率模型

生成，其中 x 是一些文本，h 是特定主题（“体育”、“政治”等）的比例。然后可以想象 h 是 x 的某种简短且更高级的描述符。

许多密度估计技术——如后面描述的变分方法——也给出了一种表示的概念：学习分布的方法通常也伴随着一个候选分布 $p(h|x)$。这就是为什么学生有时会将表示学习与密度估计混淆。但今天许多表示学习的方法并不归结为

14.2 *Training Objective for Learning Distributions: Log Likelihood*

我们希望根据从分布中获得的独立同分布样本集 S（“证据”）推断出最佳的 θ。衡量“最佳”的一个标准方法是根据 *maximum likelihood principle* 选择 θ，它指出最佳模型是分配最高概率给训练数据集的模型。1

1 似然原理是一种哲学立场，而不是某些数学分析的结果。

$$\max_{\theta} \prod_{x^{(i)} \in S} p_\theta(x^{(i)}) \tag{14.1}$$

因为对数是单调的，这也等价于最小化 *log likelihood*，这是一个对训练样本的求和，因此其形式与书中迄今为止看到的训练目标相似：

$$\max_{\theta} \sum_{x^{(i)} \in S} \log p_\theta(x^{(i)}) \quad \textit{(log likelihood)} \tag{14.2}$$

经常使用每个数据点的平均对数似然，这意味着将（14.2）除以$\|S\|$。

与监督学习一样，除了泛化之外，还需要跟踪训练对数似然，并在最大化它的模型中进行选择。通常，即使在相当简单的设置中，这种优化也是计算上不可行的，因此在实践中使用梯度下降的变体。

当然，更重要的问题是训练好的模型学习数据分布的效果如何。显然，我们需要一个类似于监督学习中的 *generalization* 的“良好性”概念来描述无监督学习。

14.2.1 *Notion of goodness for distribution learning*

最明显的泛化概念源于对数似然目标。与泛化概念最相似的

在监督学习中，一个任务是评估 *held-out* 数据上的对数似然目标：保留一些数据用于测试，并将模型在训练数据上的平均对数似然与在测试数据上的平均对数似然进行比较。

示例14.2.1. *The log likelihood objective makes sense for fitting any parametric model to the training data. For example, it is always possible to fit a simple Gaussian distribution* $\mathcal{N}(\mu, \sigma^2 I)$ *to the training data in* $\Re^d$*. The log-likehood objective is*

$$\sum_i \frac{|x_i - \mu|^2}{\sigma^2},$$

which is minimized by setting μ *to* $\frac{1}{m}\sum_i x_i$ *and* σ^2 *to* $\sum_i \frac{1}{n}|x_i - \mu|^2$*.*

Suppose we carry this out for the distribution of real-life images. What do we learn? The mean μ *will be the vector of average pixel values, and* σ^2 *will correspond to the average variance per pixel. Thus a random sample from the learn distribution will look like some noisy version of the average pixel.*

This example also shows that matching average log-likelihood for training and held-out data is insufficient for actually learning the distribution. The gaussian model only has $d + 1$ *parameters and simple* ϵ*-cover arguments as in Chapter 5 show under fairly general conditions (such as coordinates of* x_i*'s being bounded) that if the number of training samples is moderately high then the log-likelihood on the average test sample is similar to that on the average training sample. However, the learned distribution may be nothing like the true distribution.*

This is reminiscent of the situation in supervised learning whereby a nonsensical model —e.g., one that outputs random labels—has excellent generalization as well because it has similar loss on training as well as test data.

但我们如何知道对数似然目标在原则上能够学习分布？以下定理展示了这一点。

定理14.2.2. *Given enough training data, the* θ *maximizing (14.2) minimizes the KL divergence* $KL(P||Q)$ *where* P *is the true distribution and* Q *is the learnt distribution.*

Proof. 这可以从以下得出

$$\begin{aligned} KL(P||Q) &= \mathop{\mathbb{E}}_{x\sim P}[\log \frac{P(x)}{Q(x)}] \\ &= \mathop{\mathbb{E}}_{x\sim P}[\log P(x)] - \mathop{\mathbb{E}}_{x\sim P}[\log Q(x)]. \end{aligned}$$

注意，$\mathbb{E}_{x\sim P}[\log P(x)]$ 是一个仅依赖于数据分布的常数，并且使用来自 P 的独立同分布样本计算对数似然就像估计第二项一样。我们得出结论，

给定足够的样本，最小化 $KL(P||Q)$ 等同于最大化对数似然（加上一个加性常数）。 □

请注意，除了低维设置外，前述定理并未给出对所需训练数据点数量的任何有意义的界限。

14.3 *Variational method*

如上图所示，我们假设了一个真实情况下的生成模型 $p(x、h)$，并假设我们通过根据真实情况生成 $(x、h)$ 对并丢弃 h 部分，获得了 x 的样本。*variational method* 尝试从这些样本中学习 $p(x)$，其中标题中的“变分”指的是变分法。它利用了 *duality*，这是数学中一个广泛的原则。想法是维持一个分布 $q(h|x)$，作为尝试模拟 $p(h|x)$ 并提高 $p(x)$ 的某个下界的尝试。关键事实如下。[2]

2 请参阅Arora和Risteski在offconvex.org上的博客文章，了解算法如何尝试使用某种形式的梯度下降或局部改进来提高 $q(h|x)$。

引理14.3.1（ELBO界限）。*For any distribution* $q(h|x)$

$$\log p(x) \geq \mathbb{E}_{q(h|x)}[\log(p(x,h))] + H[q(h|x)], \tag{14.3}$$

where H is the Shannon Entropy. (Note: equality is attained when $q(h|x) = p(h|x)$*.)*

Proof. 由于

$$KL[q(h|x) \,||\, p(h|x)] = \mathbb{E}_{q(h|x)}\left[\log \frac{q(h|x)}{p(h|x)}\right] \tag{14.4}$$

并且 $p(x)p(h|x) = p(x,\ h)$ (贝叶斯定理)，我们有：

$$KL[q(h|x)|p(h|x)] = \mathbb{E}_{q(h|x)}[\log \frac{q(h|x)}{p(x,h)} \cdot p(x)] \tag{14.5}$$

$$= \underbrace{\mathbb{E}_{q(h|x)}[\log(q(h|x))]}_{-H(q(h|x))} - \mathbb{E}_{q(h|x)}[\log(p(x,h))] + \mathbb{E}_{q(h|x)}[\log p(x)] \tag{14.6}$$

但是，由于KL散度总是非负的，所以我们得到：

$$\mathbb{E}_{q(h|x)}[\log(p(x))] - \mathbb{E}_{q(h|x)}[\log(p(x,h))] - H(q(h|x)) \geq 0 \tag{14.7}$$

这导致所需的不等式，因为 $\log(p(x))$ 在 $q(h|x)$ 上是常数，因此 $\mathbb{E}_{q(h|x)}[\log(p(x))] = p(x)$。

□

14.4 *Autoencoders and Variational Autoencoder (VAEs)*

自编码器找到数据点 x 的压缩潜在表示 h，使得可以从 h 中近似恢复 x 。它们可以通过改变“近似恢复”含义的形式化来以多种方式定义。

在这一节中，我们使用潜在变量生成模型对它们进行形式化。在深度学习中，这种实现的一个流行实例是 *Variational Autoencoder (VAE)* [3] 。正如其名所示，两个核心经典思想在于

[3] VAEs的设计背后：自编码器——原始数据$x \in \mathbb{R}^n$被映射到一个低维（希望是有意义的）流形上的高级描述符$z \in \mathbb{R}^d$；变分推理——要最大化的目标是似然对数的下界，而不是似然对数本身。

回忆在密度估计中，我们给定一个数据样本 x_1，...，x_m 和一个参数模型 $p_\theta(x)$，我们的目标是最大化数据的对数似然：$\max_\theta \sum_{i=1}^m \log p_\theta(x_i)$。作为变分方法，VAEs 使用证据下界（ELBO）作为训练目标。对于任何在 (x, z) 上的分布 p 和 q 在 $z|x$ 上的分布，ELBO 是从 $KL(q(z|x) \,||\, p(z|x)) \geq 0$ 这一事实推导出来的。

$$\log p(x) \geq \mathbb{E}_{q(z|x)}[\log p(x,z)] - \mathbb{E}_{q(z|x)}[\log q(z|x)] = ELBO \tag{14.8}$$

在等号成立的情况下，当且仅当 $q(z|x) \equiv p(z|x)$。在 VAE 设置中，分布 $q(z|x)$ 作为编码器，将给定的数据点 x 映射到高级描述符的分布，而 $p(x、z) =、p(z)p(x|z)$ 作为解码器，根据随机种子 $z \sim p(z)$ 重建数据 x 上的分布。在构建上述编码器 q 和解码器 p 时，深度学习在 VAEs 中发挥作用。特别是，

$$q(z|x) = \mathcal{N}(z; \mu_x, \sigma_x^2 I_d), \quad \mu_x, \sigma_x = E_\phi(x) \tag{14.9}$$

$$p(x|z) = \mathcal{N}(x; \mu_z, \sigma_z^2 I_n), \quad \mu_z, \sigma_z = D_\theta(z), \quad p(z) = \mathcal{N}(z; 0, I_d) \tag{14.10}$$

在 E_ϕ 和 D_θ 分别由 ϕ 和 θ 参数化的编码器和解码器神经网络中，μ_x 和 μ_z 是相应维度的向量，σ_x 和 σ_z 是（非负）标量。高斯分布的选择本身并不是模型所必需的，可以替换为任何其他相关分布。然而，正如通常情况，高斯分布提供了计算上的便利和直观的支持。使用高斯分布的直观论证是，在轻微的正则性条件下，每个分布都可以通过高斯分布的混合（在分布上）来逼近。这源于通过将分布的累积分布函数（CDF）近似为阶梯函数，可以得到分布上的近似混合。

常数的，即具有 ≈ 0 方差的高斯混合。另一方面，计算简便性在 VAEs 的训练过程中更为明显。

14.4.1 Training VAEs

如前所述，变分自编码器的训练涉及在由（14.9），（14.10）描述的模型下，最大化（14.8）的右侧，即ELBO，关于参数ϕ，θ。鉴于参数模型基于两个神经网络E_ϕ，D_θ，目标优化是通过基于梯度的方法进行的。由于目标涉及对$q(z|x)$的期望，计算其精确估计及其梯度是不可行的，因此我们求助于（无偏）梯度估计器，并最终使用基于随机梯度的优化方法（例如SGD）。

在这个部分，使用符号 $\mu_\phi(x)$、$\sigma_\phi(x) = E_\phi(x)$ 和 $\mu_\theta(z)$、$\sigma_\theta(z) = D_\theta(z)$ 来强调对参数 ϕ、θ 的依赖。给定训练数据 x_1、...、$x_m \in \mathbb{R}^n$，考虑一个任意数据点 x_i、$i \in [m]$ 并将其通过编码器神经网络 E_ϕ 转换为 $\mu_\phi(x_i)$、$\sigma_\phi(x_i)$ 。接下来，从分布 $q(z|x = x_i) = \mathcal{N}(z$ 中采样 s 个点 z_{i1}、...、z_{is}，其中 s 是批量大小；通过重参数化技巧 [4] 采样 ϵ_1、...、$\epsilon_s \sim \mathcal{N}(0$、$I_d)$ [4] 从标准高斯分布，并使用变换 $z_{ij} = \mu_\phi(x_i) + \sigma_\phi(x_i) \cdot \epsilon_j$。重参数化技巧背后的原因是，关于一般分布 q_ϕ 上期望的无偏估计的参数 ϕ 的梯度不一定是期望梯度的无偏估计。然而，当分布 q_ϕ 可以将参数 ϕ 从分布中的随机性中分离出来时，这种情况就成立了，即它是一个依赖于参数无分布的 ϕ 的确定性变换。通过从 $q(z|x = x_i)$ 中获得的独立同分布样本 s，我们得到目标 ELBO 的无偏估计

$$\sum_{j=1}^{s} \log p(x_i, z_{ij}) - \sum_{j=1}^{s} \log q(z_{ij}|x_i) = \sum_{j=1}^{s} [\log p(x_i|z_{ij}) + \log p(z_{ij}) - \log q(z_{ij}|x_i)] \tag{14.11}$$

这里批大小 s 表示在估计中的计算效率和准确度之间的基本权衡。由于 (14.11) 中的求和项都是高斯分布，我们可以将 ELBO 估计明确地用参数相关的 $\mu_\phi(x_i)$、$\sigma_\phi(x_i)$、$\mu_\theta(z_{ij})$、$\sigma_\theta(z_{ij})$ (来表示，同时跳过一些常数)。对于 $j \in [s]$ 的单个项给出如下

$$-\frac{1}{2}\left[\frac{||x_i - \mu_\theta(z_{ij})||^2}{\sigma_\theta(z_{ij})^2} + n \log \sigma_\theta(z_{ij})^2 + ||z_{ij}||^2 - \frac{||z_{ij} - \mu_\phi(x_i)||^2}{\sigma_\phi(x_i)^2} - d \log \sigma_\phi(x_i)^2\right] \tag{14.12}$$

注意，(14.12) 对所有分量 $\mu_\phi(x_i)$、$\sigma_\phi(x_i)$、$\mu_\theta(z_{ij})$、$\sigma_\theta(z_{ij})$ 都是可微的，而每个这些分量，作为一个具有参数 ϕ 或 θ 的神经网络的输出，对参数 ϕ 或 θ 是可微的。因此，批量求和 (14.11) 对 ϕ (或 θ) 的可处理梯度是，*due to the reparame- terization trick*，这是 $\nabla_\phi ELBO$ (或 $\nabla_\theta ELBO$) 的无偏估计，可用于任何基于随机梯度的优化算法以最大化目标 ELBO 并训练 VAE。

14.5 Normalizing Flows

VAE的局限性在于，它不是直接优化对数似然，而是优化了对数似然的下界。理想情况下，我们希望在保持具有复杂表示能力的深度模型的同时，克服这一局限性。（本章开头描述的简单高斯拟合也直接优化对数似然，但它无法表示复杂的分布。）*Normalizing flows*可以做到这一点，

The idea in *Normalizing Flows* (Rezende and Mohamed 2015) is to make the deep net *invertible*. Specifically, it computes a function $f_\theta : \Re^d \rightarrow \Re^d$ tha t is parametrized by trainable parameter vector θ and maps image x to its repr esentation $h = f_\theta(x)$ (note: both have the same dimension). Importantly, f is an invertible map (i.e., one-to-one and onto) and differentiable (or almost eve rywhere differentiable). The advantage of such a transformation is that it give s a clear connection between the probability densities of x and h. In generativ e models h is assumed to have some prescribed probability density $\mu(h)$, usua lly uniform gaussian. Via the invertible map, this translates to a density $\rho(\cdot)$ o n x given by

$$\rho(x) = \mu(f(x))|\det(J_f)| \tag{14.13}$$

在 J_f 是 f 的雅可比矩阵，即其 (i, j) 项为 $\partial f(x)_i/\partial x_j$，$\det(\cdot)$ 表示矩阵的行列式。这个关于训练数据点似然的确切表达式允许进行基于梯度的常规训练。

这引发了一个问题：如何约束网络使其可逆？请注意，只需约束单个层使其可逆即可，因为整体雅可比矩阵是层雅可比矩阵的复合。[5] 为了使层可逆，通常使用一种变体

以下技巧来自模型NICE [6]和Real NVP [7]。如果z^l是输入到层 l 和 z^{l+1} 的输出，然后在 z^l 和 z^{l+1} 中识别一组特殊的坐标 A 并施加限制（其中 z_A 表示

[5] Since $\det(AB) = \det(A)\mathbf{det}(B)$ the determinant of the net Jacobian is the product of the determinants of the layers.

[6] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Nonlinear Independent Component Analysis. *Proc. ICLR*, 2015

[7] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density Eestimation using Real NVP. *Proc. ICLR*, 2017

图14.3：顶行中的面孔是通过基于VAE的方法生成的，而第二行中的面孔是通过使用归一化流的真实NVP生成的。VAE因其生成模糊图像而闻名。RealNVP的输出要好得多，但仍存在可见的伪影。

z在由A和B给出的坐标中的部分，其中B是A的简称）。

$$z_A^{l+1} = z_A^l \tag{14.14}$$
$$z_B^{l+1} = z_B^l \odot h_\theta(z_A^l) + s_\theta(z_A^l) \tag{14.15}$$

$\odot$表示分量积，$h_\theta()$是一个每个输出都是非负的函数，一个方便的选择是将它设为某个其他函数$r_\theta()$的$\exp(r_\theta(z_A^l))$。

这一层是可逆的，因为给定 z^{l+1}，可以按照以下方式恢复 z^l:

$$z_A^l = z_A^{l+1} \tag{14.16}$$
$$z_B^l = (z_B^{l+1} - s_\theta(z_A^l)) \odot h_\theta(z_A^l) \tag{14.17}$$

请注意，A、B的选择可能会从一层到另一层发生变化，因此所有坐标在通过多层时都可能被更新。

此外，用 $z|_A$ 表示层向量在坐标 A 上的部分，层映射的雅可比矩阵是下三角矩阵。因此，行列式是主对角线元素的乘积。

$$\frac{\partial}{\partial z^l} z^{l+1} = \begin{pmatrix} I_{|A|\times|A|} & 0 \\ \frac{\partial}{z^l|_A} z^{l+1}|_B & \text{diag}(h_\theta(z_A^l)) \end{pmatrix}$$

正态流可以通过限制卷积为1×1来扩展到卷积网络。然后卷积滤波器仅涉及通道值的缩放，相应的雅可比矩阵是对角非零矩阵。此外，坐标的A和B分割也可以在通道内完成。这是GLOW模型8中的一个想法，它可以生成比其更好的图像。

8 Diederik P. Kingma 和 Prafulla Dhariwal. GLOW：具有可逆1×1卷积的生成流。*Proc.*，2019 *Neurips*

前驱

更近期的自回归模型，如PixelCNN，能够从随机种子生成非常逼真的图像。然而，它们不符合上述描述的分布学习范式，因此我们在此不讨论它们。它们涉及生成

图像像素逐个（粗略地说）因此并行化不好。

问题14.5.1. *Let* (z_1, z_2, z_3, z_4) *be distributed as a standard Gaussian* $\mathcal{N}(0, I)$ *in* $\mathbb{R}^4$. *Let* $f\colon \mathbb{R}^4 \to \mathbb{R}^4$ *be an invertible function which maps* (z_1, z_2, z_3, z_4) *to* $(z_1, z_2, e^{a_0} z_3 + a_1 z_1^2 + a_2 z_2^2, e^{b_0} z_4 + b_1 z_1^2 + b_2 z_2^2)$ *for some coefficients* $a_0, a_1, a_2, b_0, b_1, b_2 \in \mathbb{R}$. *Compute the probability density function of* $f(z_1, z_2, z_3, z_4)$.

14.6 Stable Diffusion

您可能见过一些AI模型，根据文本提示如“教皇方济各穿着羽绒服散步”生成人工图像。这些是由*diffusion models* [9]制作的，我们将在本节中对其进行描述，

尽管没有文本提示。

扩散模型让人联想到正态流和自编码器，因为它们定义了一个映射 f，将所有图像的集合转换到集合 $\mathcal{N}(0、I)$，以及一个逆映射 f^{-1}，将高斯向量映射到图像。不同之处在于 f 非常简单；只是一系列噪声步骤。此外，f^{-1} 只是在对 f 的输出进行去噪方面进行了定制训练。

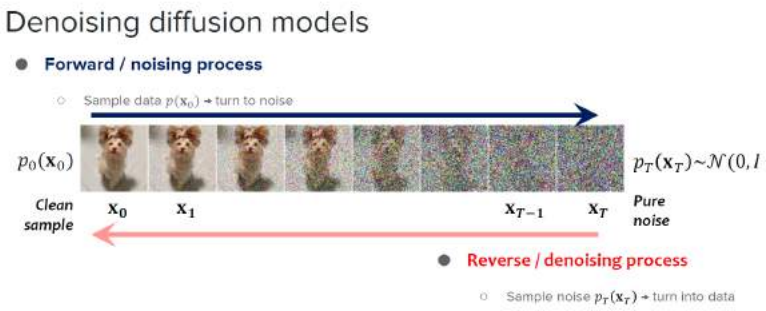


图14.4：使用扩散模型对图像进行噪声化和去噪的示例。（来源：Binxu Wang）

噪声层以以下方式在 T 步骤中将图像 x_0 到 x_T 噪声化，其中 $z_t \sim \mathcal{N}(0、I)$ 以及每个 $\alpha_t \in (0、1)$

$$x_{t+1} = \sqrt{\alpha_t} x_t + \sqrt{1 - \alpha_t} z_t, \qquad t = 0, \ldots, T-1 \tag{14.18}$$

一个简单的归纳表明这等价于以下公式，其中 $\alpha_t = \prod_{i=1}^{t} \alpha_i$：

$$x_{t+1} = \sqrt{\overline{\alpha_t}} x_t + \sqrt{1 - \overline{\alpha_t}} z. \tag{14.19}$$

上述计算使用以下内容。

问题 14.6.1. *If* z_1, z_2 *are independent samples from* $\mathcal{N}(\mu_1, \sigma_1^2 I)$ *and* $\mathcal{N}(\mu_2, \sigma_2^2 I)$ *respectively then* $z_1 + z_2$ *is distributed as*

$$\mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2 I).$$

15
Language Models (LMs)

大约从2020年开始，语言模型（LMs）突然成为了人工智能研究最显眼的代表。本章介绍了基本概念和结果。其核心

一个关键概念是，LMs 假设人类产生的文本是从概率分布中采样的，其中 $\Pr(w_1\ w_2\ ...\ w_i)$ 是与单词序列 $w_1\ w_2\ ...\ w_i$ 相关的概率。然后贝叶斯定理意味着一个分解：

$$\Pr[w_1\ w_2\ \ldots w_{i+1}] = \Pr[w_1\ w_2\ \ldots w_i]\Pr[w_{i+1}\ |w_1 w_2 \ldots w_i] \quad (15.1)$$

这表明，为了对*generate*一系列单词进行建模，我们只需要计算$\Pr[w_1]$的能力，然后根据前面的i个单词生成$i+1$个单词的方法。简而言之，这就是语言模型所做的事情.[1]最简单的模型可以追溯到20世纪50年代，当时计算机

普林斯顿先驱克劳德·香农提出了对（15.1）的多种简单近似。最简单的是称为 *unigram* 模型的，通过测量足够大的语料库中单词 w 的经验频率来计算概率 $\Pr[w]$ 的估计值，并使用近似 $\Pr[w_{i+1}\ |w_1 w_2 \ldots w_i] = \Pr[w_{i+1}]$，这相当于说 $\Pr[w_1\ w_2 \ldots w_{i+1}] = \prod_i \Pr[w_i]$。*bigram* 模型做类似的事情，但假设 $\Pr[w_{i+1}\ |w_1 w_2 \ldots w_i] = \Pr[w_{i+1} \mid w_i]$。也就是说，下一个单词的概率取决于前一个单词，但不取决于任何更早的单词。然后只需要对所有单词对 w，w' 的 $\Pr[w \mid w']$ 进行经验估计，这可以使用适度的语料库大小来完成。

1 此阐述忽略了用于计算文本片段语义嵌入的其他类型语言模型。著名的包括BERT、ERNIE、RoBERTA等。

在最近几十年中，随着神经网络应用于语言建模，一个关键思想出现了：上下文语义向量。这个想法是，为了预测下一个词 w_{i+1}，使用前一个词的序列 $w_1 w_2 ... w_i$，你计算 $w_1 w_2 ... w_i$ 的“意义”的嵌入 $c_i \in \Re^D$。你还有一个每个词 w 的语义嵌入 $v_w \in \Re^D$。然后下一个词的分布描述为：

$$\Pr[w_{i+1}\ |w_1 w_2 \ldots w_i] \propto \exp(v_{w_{i+1}} \cdot c_i). \quad (15.2)$$

自从这个想法在语言建模中变得无处不在，让我们了解一个叫做 *softmax* 的概念。[2] 描述分布通过（15.2）相当于说，下一个词的分布是$\Re^V$中向量的softmax，其$i+$的第1个坐标是$v_{w_{i+1}} \cdot c_i$。在这里，V表示词汇量大小（即不同单词的数量）。语义嵌入和softmax也用于定义今天LMs中的“下一个词”分布。与现代LMs的关键变化是用于计算嵌入的transformer架构。

2 回想一下，softmax 是从 $\Re^k$ 到单纯形的映射，即 $\{(p_1, p_2, ..., p_k): p_i \geq 0, \sum_i p_i = 1\}$。对于 $(s_1, s_2, ..., s_k) \in \Re^k$，定义其 softmax 为第 i 个坐标为 $\exp(s_i)/(\sum_{j=1}^k \exp(s_j))$ 的向量。

语言模型的良好性通过其 *cross entropy loss* 计算，对于一个单词序列 $w_1 w_2 ... w_t$ 是：

$$\ell(M) = \sum_i \log \frac{1}{\Pr_M[w_{i+1} \mid w_1 w_2 \dots w_i]} \qquad \text{(Cross Entropy)} \quad (15.3)$$

模型通过在文本语料库上最小化（通过模型参数的梯度下降）这种训练损失来训练，它们的良好性通过它们的*test loss*来计算——在相同语料库中保留的文本上评估相同的损失表达式。通常，训练语料库非常大，模型只在每篇文本上训练一次（或几次），到结束时，保留文本上的测试损失几乎与训练损失相同。因此，泛化误差相当小。

自上述损失的最小化等同于最大化

$$\prod_i \Pr_M[w_{i+1} \mid w_1 w_2 \dots w_i],$$

有时，语言模型的目标被描述为尝试创建一个模型，该模型将最大的可能概率分配给训练语料库。这实际上对学生来说有点误导。第15.2节解释说，真正的目标是让模型学习人类分布。

语言任务：自然语言处理领域的数十年的研究已经确定了数千个任务。一些例子：*Implication:* 给定两段文本，判断第一段是否暗示了第二段。*Sentiment* 给定一段文本，判断它是否具有 *positive, negative or neutral* 情感。*Question-answering:* 给定一段文本，回答与之相关的问题。*Translation:* 给定一种语言的文本，将其翻译成另一种语言。

多年来，此类任务很难。如今，正如您所知，这些以及更多更困难的任务通常使用大型语言模型来解决。本章不会讨论架构和训练细节，因为互联网上已经存在很好的文章。相反，它讨论了交叉熵损失的概念基础（第15.2节），以及如何训练语言模型生成在人类交互中有用或有意义文本的方法（第15.6节，？）。然后我们关注

15.1 Transformer Architecture

关于2017年之前的15年左右，在LMs领域有很多创新，许多神经架构在这个过程中被设计出来。该架构的目标是计算词嵌入和上下文嵌入，以便预测下一个词。

2017年开始，这些架构在大多数环境中已被Transformer架构有效取代，现在也用于图像、声音、基因组和其他类型的数据。我们建议阅读Lilian Weng的博客上关于Transformer的优秀介绍。3以下问题邀请您测试您的

3 "*Tranformer Family: Version 2*. https://lilianweng.github.io/. 2023

理解。

问题 15.1.1. *Consider a N-layer single-head transformer with input section length L and hidden state dimension d. For each layer l, let the input for the layer be X_l and the output be X_{l+1}, we have*

$$X_{l+1} = V_l \cdot softmax\left(\frac{Q_l^\top K_l}{\sqrt{d}}\right),$$

where $Q_l = W_l^q X_l$, $K_l = W_l^k X_l$, $V_l = W_l^v X_l$, and $X_l \in \mathbb{R}^{d\times L}$, W_l^q, W_l^k, $W_l^v \in \mathbb{R}^{d\times d}$. If the size of dataset is M (total number of tokens), find the asymptotic training time of each epoch in terms of M, N, L, d.
Note that here we consider the input data X_l as a collection of column vectors (each column is a data point), so the parameter matrices W's are multipled on the left of X's. In some other literatures such as the Lilian Weng's blog, X_l is a collection of row vectors (each row is a datapoint). The 2 definitions are sometimes used interchangeably.

15.2 Explanation of Cross-Entropy Loss

现在我们试图理解交叉熵损失（15.3）背后的概念框架。如前所述，存在一个用于生成下一个单词的基真（即人类）分布，该分布将概率 $p_i(w \mid w_1 w_2 \dots w_i)$ 分配给事件：在给定前缀单词 $w_1 w_2 \dots w_i$ 的情况下，$(i+1)$th 个单词是 w。为了简洁的符号，我们将 $p_i(w \mid w_1 w_2 \dots w_i)$ 简写为 $p_i(w)$，因此 $(i+1)$th 个单词的 *entropy* 是

$$\sum_w p_i(w) \log \frac{1}{p_i(w)} \quad \text{(ENTROPY)} \tag{15.4}$$

这个熵是语言的固有属性，源于人类作者在下一个词的选择中做出的许多选择。给定序列 $w_1 w_2 \dots w_i$，模型对下一个词 w 有概率分布 $q(w|w_1 w_2 \dots w_i)$。扩展我们的紧凑表示法，我们使用 $q_i(w)$

作为一个缩写，该模型在$(i+1)$词上的交叉熵损失是$\log \frac{1}{q(w_{i+1})}$，这应被视为一个经验估计

$$E_{w\sim p_i()}[\log \frac{1}{q(w)}] \quad \text{(EXPECTED C-E LOSS)} \tag{15.5}$$

KL divergence, 也被称为 *excess entropy*，是非负的，定义为

$$KL(p_i||q_i) = E_{w\sim p_i()}[\log \frac{p_i(w)}{q_i(w)}] \quad \text{EXCESS ENTROPY} \tag{15.6}$$

因此，按单词计算，我们有：

$$\text{EXPECTED C-E LOSS} = \text{ENTROPY} + \text{EXCESS ENTROPY} \tag{15.7}$$

对整个保留的语料库进行求和，可以得到整个语料库的相似估计。可以做出轻微的假设，即条件概率 $p_i()$、$q_i()$ 只依赖于（比如说）前 10^3 个词，而保留的语料库大小 M 要大得多，例如 $M \gg 10^8$。因此，语料库由某种随机游走组成，大约每 10^4 个词就会切换到语言分布的不同部分。在这样的假设下，上述关系在期望层面上在词级别上成立，在语料库级别上应该相当精确地成立。

总结来说，由于文本的熵是一个常数，我们可以将((15.7))解释如下。

语言模型目标：*The goal of language modeling is to minimize KL-divergence of the human's next-word distribution to the model's distribution.*

请注意，这适用于本章中所述的普通建模。在实践中，训练方法和甚至损失函数都偏离了上述简单图景。

15.3 Scaling Laws and Emergence

深度学习中一个古老的野心是能够简单地扩大网络规模，用更多数据进行训练，并继续更好地解决问题。虽然这个希望在此之前运作得很好，但在此之后，得益于transformers，它得到了涡轮增压。关键发现是所谓的*scaling laws*。

这些是经验推导的表达式，描述了在保留数据上的测试交叉熵损失（在实验中）如何随着模型参数数量（N）和数据集大小（D）[4] [5]，[6] 成比例变化。对于Chinchilla 模型 [7] 法律如下：

$$L(N,D) = A + \frac{B}{N^{0.34}} + \frac{C}{D^{0.28}} \qquad A = 1.61 \quad B = 406.4 \quad C = 410.7. \tag{15.8}$$

[4] [5]

[6] Yasaman Bahri，Ethan Dyer，Jared Kaplan，Jaehoon Lee和Utkarsh Sharma。解释神经缩放定律。
arXiv preprint arXiv:2102.06701，2021

[7] 约翰·霍夫曼，塞巴斯蒂安·博热奥，阿图尔·门什，叶莲娜·布查茨卡娅，特雷弗·蔡，伊丽莎白·拉瑟福德，迭戈·德·拉斯·卡萨斯，丽莎·安妮·亨德里克斯，约翰内斯·韦尔布，艾登·克拉克，等人。训练计算最优的大型语言模型。
arXiv preprint arXiv:2203.15556，2022

这里15.8中的常数A、B、C仅适用于特定的架构和训练策略——甚至常数A也取决于分词。这种使用两个基本参数描述宏观行为——类似于热力学第二定律——将帮助我们绕过对训练机制理解的必要性。我们的理论将仅依赖于方程的一般形式，特别是，N、D的依赖是逆多项式。因此，它适用于其他训练框架（例如，过拟合模型8）

在何处也发现了缩放定律。

8 尼克拉斯·穆尼霍夫，亚历山大·M拉什，博阿兹·巴拉克，特文·勒·斯高，亚历山德拉·皮克图斯，努瓦曼·塔齐，萨姆波·皮耶萨洛，托马斯·沃尔夫，以及科林·拉夫尔。扩展数据约束的语言模型，2023

Emergence: 出现是指一个有趣的实证现象，当 D、N 同时增加时，模型在 *broad range* 语言任务上的性能（零样本或少样本）以相关的方式提高。当 D、N 在对数尺度上绘制时（这通常是情况），改进可能表现为快速过渡，但现在普遍认为，对于大多数任务，当 D、N 扩大时，性能会逐渐提高。因此，术语 *slow emergence* 更为准确。此外，已知对于不同的任务，出现的发生速率不同，并且对于文本与训练数据中找到的文本可能最接近的任务，出现通常最快。

许多任务目前模型难以处理，并且它们通常与在常规文本语料库中找到的内容大相径庭。参见[WTB+22?，SMK23]以了解关于广泛语言任务出现率的实验结果。因此，可以合理地假设，技能的出现源于在训练数据集中解决下一个词预测时隐式解决的关联任务上的训练。这确实是我们研究的起点。

9 姜伟，泰伊，里希·博马萨尼，科林·拉费尔，巴雷特·佐普，塞巴斯蒂安·博尔热奥，达尼·约加塔玛，马滕·博斯马，周登尼，唐纳德·梅茨勒，等人。大型语言模型的涌现能力。*arXiv preprint*，2022。*arXiv:2206.07682*

15.4 (Mis)understanding, Excess entropy, and Cloze Questions

关于涌现和尺度定律的思考，可能会产生以下混淆：
"When we increase D from 10^{11} *to* 10^{12} *then according to (15.8) this changes cross-entropy by a tiny amount. Why does it lead to big changes in macroscopic behavior?"* 本节解释了这种推理中的缺陷。

缺陷在于，大部分损失仅仅捕捉到了语言的固有熵（即（15.8）中的 A 项）。我们现在认为，模型在下游任务上的错误（即其误解）被 *excess* 熵所捕捉，正如15.2节所述，每次模型以十倍的比例放大时，该熵都会以一个常数因子减少。

我们用一个经典的例子来说明，这个例子后来启发了 the *Winograd Schema Challenge(WSC)* [11]：

10 特里·温格罗德。作为计算机程序中理解自然语言数据的表示过程。技术报告，麻省理工学院剑桥分校，MAC项目，1971年

11 赫克托·莱维斯克，欧内斯特·戴维斯，以及莱奥拉·莫根施特恩。Winograd框架挑战。在 *Thirteenth inter-national conference on the principles of knowledge representation and reasoning*，2012

`The city councilmen refused the demonstrators a permit because they feared violence.`

这里代词 `they` 是模糊的——语法规则允许它指代 `demonstrators` 或 `city councilmen`。Winograd 指出，消除这种歧义（即指代消解）需要文本本身不可用的世界知识，即演示可能会变得暴力，市政委员会成员不喜欢暴力。

设计语言理解测试平台（如WSC）的关键思想是Cloze程序[12,13]，也常用于测试儿童语言发展。为了测试模型对这句话中 `they` 的理解，我们可以在其后添加一个 *prompt*：`Q. Who feared violence?`。这之后可能是一个空白，或者多个答案的选择：`A. city councilmen.  B. demonstrators.` 对于 WSC 示例，尽管人类会百分之百确定答案，但截至 2016 年的语言模型在这两个选项之间大约是 50/50 的困惑。

[12] 13 填空题是多项选择题，这允许测试大多数语言技能。它们不适合像理解讽刺这样的技能，因为其中一个多项选择已经解释了这个笑话。

在上述示例中，人类对答案有100%的确定性，这意味着他们的熵在这里是log 1，即0。然而，如果模型在两个选项之间以50-50的比例分割，这表明它具有交叉熵log 2，所有这些都是*excess entropy*！鉴于模糊代词在通常的英语中的频率，可以得出结论，一个没有学习到代词消歧的模型将在周围文本的许多地方显示出巨大的过剩熵。因此，过剩熵的减少（由于缩放而自然发生）往往会挤出这些错误。接下来的分析试图使这种直觉数学上精确。

当然，文本语料库通常不包含这种人工的完形填空题。但可以想象，模型对上述类型的误解往往会导致相邻文本中的预测错误。我们在第15.8节的理论将假设完形填空题可以紧密捕捉模型的误解。

15.5 How to generate text from an LM

这个部分的标题可能感觉荒谬，因为（15.1）中LM的定义涉及根据前面的词预测下一个词的能力。这似乎提供了一种生成好文本的明显方法：使用第一个词的模型分布来采样特定的w_1，然后从分布$\Pr[w_2|w_1]$中采样以生成第二个词，依此类推。所描述的程序称为随机生成或简单地采样，但实际上它并不产生好的文本[14]。

下一个自然想法是贪婪的：在生成 $w_1 w_2 \ldots w_i$ 后，使用概率最高的词生成 w_{i+1}。

[14] The quality of **sampling** gets better as models scale up and thus closer to the language distribution.

下一位。乍一看这似乎很有吸引力，因为对于语言模型（LMs）的训练目标隐式地训练它们最大化分配给它们的训练文本的概率。所以当生成文本时，为什么不尝试生成模型赋予最高可能概率的文本片段呢？15 原因是，如解释所述

在15.2节中，语言模型的真实目标是使KL距离最小化到人类分布。

15 实际上贪婪算法并不完全最大化概率，但它是朝着这个方向的一个尝试。

贪婪文本看起来平淡无趣。例如，*Thanks for the dessert, it was* . . . 的贪婪延续可能是 *great*，但可能有更多有趣且概率较低的选项，例如 *exquisite*，*life-changing* 等。人类的交流常常偏向于低概率词汇。确实，贪婪方法生成的文本的困惑度远不及人类生成的文本！关于这个问题，16 中有很好的讨论，其中图-

16 A Holtzman, J Buys, L Du, M Forbes, 和 Y Choi. 神经文本退化的奇特案例。*ICLR*，2020

图15.1已被取用。

Method	Perplexity
Human	12.38
Greedy	1.50
Beam, b=16	1.48
Stochastic Beam, b=16	19.20
Pure Sampling	22.73
Sampling, t=0.9	10.25
Top-k=40	6.88
Top-k=640	13.82
Top-k=40, t=0.7	3.48
Nucleus p=0.95	**13.13**

自台数选择的主要结果

图15.1：各种生成方法的文本困惑度。随机和贪婪方法相当糟糕。使用$p = 0.95$进行核采样最接近人类。（我们没有描述束搜索，所以请忽略那些行。）

实际上，最佳方法实际上通过一些贪婪式的选择来*reshape*分布。

- Top-{v*}: 确定前 k 个选择作为第一个单词，并限制自己只从这些单词中选择第一个单词（即不允许在这个位置选择其他任何单词）。如果它们的联合概率是 p_k，则通过将这 k 个单词的概率按 $1/p_k$ 缩放并使所有其他单词的概率为零来实现，然后从这个分布中选择。选择了第一个单词后，继续以类似的方式选择其余单词。您通过试错法设置 k 以最佳匹配人类文本的困惑度。

- 核采样（又称“top p”）：这是上述方法的一种较软的变体。不是对第一个词的可能选择数量（即，k）做出硬决策，而是决定允许 k 变化，但随后对第一个位置允许的所有选择的总体概率施加硬约束，即 p。对其他词位置也以相同的方式继续进行。

“概率预算。”您通过试错法设置 p 以最佳匹配困惑度与人类文本。

15.6 Instruction tuning

如上所述，LLMs可能对语言有很好的理解，但可能不太理解参与人类对话。例如，如果人类询问*Can you write a haiku that fits in a tweet?*，LLM可能只是回答“是”，因为它可能不理解它被期望提供这样的俳句。指令调整包括在(x, y)对的数据集上进行训练，其中x是人类指令，y是模型答案。机器被喂食这样的(x, y)对，并训练以最小化y中的交叉熵。

在本文末尾，它能够为人类问题提供良好的答案。让我们称这个分布为 $\pi_{SFT}(y|x)$，其中 SFT 代表“监督微调”。

15.7 Aligning LLMs with human preferences

尽管指令调整赋予了LLM回答问题的能力，但在提示有问题的情况下，其答案可能不符合我们对正确性、道德等观念的认识。例如，当被要求帮助策划犯罪时，指令调整的LLM会轻易提供详细的指示。或者当询问它未见过数据的事件时，它可能会*hallucinate*关于这一事件的某些事实，因为语言模型对所有类型的文本都有非零概率。设$\mathcal{D}$为这类问题提示的分布。将此视为根据提示的“棘手”程度对其进行加权。

训练数据集由包含成对17 $(x, y_1) \succ (x, y_2)$ 的人类偏好数据组成，其中 x 是一个有问题的提示被抽取根据，y_1，y_2 是对它的两个不同答案，而 y_1 更受青睐于 y_2。

17 人类往往觉得在两种选择之间做出偏好比在更大的一组选择中产生排名要容易得多。

著名的布拉德利-特里模型18用于人类偏好表示这样的成对排名对应于以下分布的随机抽样：

18 拉尔夫·A·布拉德利和米尔顿·E·特里。不完整区组设计的秩分析：I. 配对比较法。*Biometrika*，1952

$$p^*(y_1 \succ y_2 \mid x) = \sigma(r^*(y_1|x) - r^*(y_2|x)) \tag{15.9}$$

where sig 是 *sigmoid* 函数 19 和 r^* 是所谓的 *reward function* 该映射将一个（提示，响应）对（x，y）映射到$\Re$。假设人类在脑海中具有这样的奖励函数。如果你给一个人两个针对提示x的响应y_1，y_2，那么他们会根据上述函数更喜欢y_1而不是y_2。[20]我们表达奖励

19 Sigmoid函数 $\sigma(t) = \frac{1}{1+e^{-t}}$ 将 $(-\infty, \infty)$ 映射到 $(0, 1)$。$= \frac{e^t}{1+e^t}$

$r^*(y_1 \mid x)$ 使用条件表示法，因为奖励函数仅适用于比较对同一提示 x 的响应。它不能

例如，如果提示是一个请求帮助犯下可怕罪行的提问，那么大多数人会强烈反对一个提供犯罪建议的回应 $y|x$。这可以解释为这个回应具有极低的 r^* 价值。

用于比较对两个不同提示 x 和 x' 的响应的奖励。我们如何在人类大脑中学习这个（未知的）奖励函数 $r*()$ 呢？诀窍在于

问题15.7.1（学习奖励模型）。*Suppose we have a dataset $\mathcal{D}$ consisting of $(x,\ y_1,\ y_2)$ triples where x is a question, y_1, y_2 are two answers and human raters have indicated $(x,\ y_1) \succ (x, y_2)$. Show that the max- likelihood fit for a reward function r_ϕ is*

$$\mathcal{L}(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x,y_1,y_2)\sim\mathcal{D}}\left[\log \sigma(r_\phi(x, y_1) - r_\phi(x, y_2))\right] \tag{15.10}$$

我们希望训练一个参数模型 $\pi_\theta(y|x)$，该模型根据学习到的奖励函数 $r^*()$ 行动。我们可以通过以下思想实验来解决这个问题：想象一个提示 x 从 $\mathcal{D}$ 中采样，并且模型根据 $\pi_\theta(y \mid x)$ 生成一个响应。然后机器获得一个 *reward* $r^*(y \mid x)$。因此，一个好的模型应该优化

$$\max \mathop{\mathbb{E}}_{x\sim\mathcal{D}, y\sim\pi(y|x)}[r^*(y \mid x)]. \tag{15.11}$$

然而，此目标仅涉及有利于“狡猾”提示的分布 $\mathcal{D}$。为了继续成为一个优秀的语言模型，$\pi_\theta()$ 必须不偏离 $\pi_{SFT}()$ 过远。因此，我们必须添加一个正则化项，以防止分布不必要地偏离 $\pi_{SFT}()$。

$$\max \mathop{\mathbb{E}}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[r^*(y \mid x)] - \beta \cdot KL(\pi_\theta(y \mid x)||\pi_{SFT}(y \mid x)||). \tag{15.12}$$

这是 *Reinforcement Learning with Human Feedback* (RLHF)[21] 的精髓，在数年内一直是对齐的首选方法
语言模型具有人类价值观。我们不会详细讨论确切的优化技术，而是转向下面的更好替代方案。

[21] P. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, 和 D. Amodei. 从人类偏好中进行深度强化学习。*NeurIPS*，2017

问题 15.7.2. *Show that if the parameterization θ for the policy has unlimited size then the optimum solution satisfies*

$$\pi_\theta(y \mid x) = \frac{1}{Z(x)}\pi_{SFT}(y \mid x)\exp\left(\frac{1}{\beta}r^*(y \mid x)\right), \tag{15.13}$$

where $Z(x) = \sum_y \pi_{SFT}(y \mid x)\exp\left(\frac{1}{\beta}r^*(y \mid x)\right)$ is the partition function.

15.7.1 Direct Reward Optimization

上述RLHF方法由于优化问题通常难以实现。最近，一种称为 *Direct Preference Optimization (DPO)* [22]的方法因其利用了
与更简单的优化相同的Bradley-Terry偏好框架。

[22] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, 和 Chelsea Finn. 直接偏好优化：你的语言模型其实是一个奖励模型。*arxiv*，2023

主要思想是跳过奖励函数，而是直接使用（15.7.2）通过模型 $\pi_\theta(y \mid x)$ 表达奖励函数，并直接优化人类偏好对 $(x, y_1) \succ (x, y_2)$ 。具体来说，（15.7.2）意味着最优模型 $\mathrm{p}^*(y \mid x)$ 对所有 x 满足

$$\Pr[(y_1 \mid x) \succ (y_2 \mid x)] = \frac{1}{1+\exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{SFT}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{SFT}(y_1|x)}\right)} \tag{15.14}$$

问题15.7.3. *(DPO) Use the above to show that the following objective captures the search for the best parametric model $\pi_\theta(y|x)$ given preferences (x, y_1, y_2)*

$$\max_\theta \mathbb{E}_{(x,y_1,y_2)} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_1 \mid x)}{\pi_{SFT}(y_1 \mid x)} - \beta \log \frac{\pi_\theta(y_2 \mid x)}{\pi_{SFT}(y_2 \mid x)}\right)\right]. \tag{15.15}$$

DPO 实际上像 RLHF 一样，将 KL 惩罚附加到这个目标上，并对其进行优化。请参阅论文以获取实现细节和实验。

15.8 Mathematical Framework for Skills and Emergence

我们现在给出本章的主要理论组成部分，这是一个关于技能及其可能与语言理解任务以及出现现象相关联的新数学框架。这来自一篇近期论文23。

23 Sanjeev Arora 和 Anirudh Goyal. 语言模型中复杂技能出现的理论。*arXiv preprint*，2023 *arXiv:2307.15936*

首先，假设语言理解涉及一系列技能，尽管理论不需要知道一个精确的列表。（学者们已经发现并命名了数千种技能。经过良好训练的转换器无疑发现了更多尚未命名的技能。）接下来，理论将假设缩放定律，如（15.8），因此不需要对训练和泛化进行推理。相反，它可以直接对模型在测试分布上的行为进行推理，即从训练数据中抽取的分布。我们假设这个测试分布是一个长的无序列表，其中每个文本片段都与一个相关度量24相关。

24 文本片段应被视为大小介于一段到几页之间，从较长的语料库中提取。为了使模型能够进行良好的预测，文本片段可以包括在较长的语料库中先于它的辅助文本。模型不需要对辅助文本中的单词进行预测，但可以使用它来对文本片段进行预测。

交叉熵损失使用此相关度量进行平均。

定义15.8.1（文本片段）。*The test corpus for the model is viewed as being divided into* 文本片段*, each consisting of C_{test} tokens. There is also a measure $\mu_2()$ on these text-pieces, with $\mu_2(t)$ denoting the measure of text-piece t. The usual cross-entropy loss is computed by weighting text-pieces with respect to this measure.*

现在我们做一些假设。我们假设模型对文本片段的“理解”可以通过合适的完形填空题进行测试。

与第15.4节中的Winograd示例类似。具体来说，我们假设在测试时已经使用了一个（未知的）cloze过程来向文本片段添加这样的cloze问题。这些是用简单英语清晰标记的多项选择题，模型必须回答这些问题。请注意，训练语料库不包含此类cloze问题，因此在测试时这是一种简单的分布偏移形式。cloze问题的预测损失不需要预测cloze问题的位置或内容——它只需要选择多项选择题的正确答案。

我们允许过程填空来调整问题以适应正在测试的模型。因此下一个假设是合理的。

假设15.8.2. *[Cloze Sufficiency Assumption:]* 预训练模型在Cloze问题上的平均（多类别）预测损失——这里的平均是对文本片段的分布进行计算——与模型在经典下一个单词预测上的交叉熵过剩（在1.1这样的小乘数因子内）紧密相关。

注意：如第15.4节所述，如果假设完形填空问题可以被人类完美解答，那么模型给出的任何错误答案都可以被解释为类似过度的交叉熵。我们的假设相当于说，完形填空问题上的错误可以紧密捕捉到模型在（15.3）中定义的过度熵。下一个定理表明，存在一个（尽管相当人为的）完形填空问题集合，其中模型答案的过度交叉熵与下一个词预测的整体过度交叉熵相匹配。

定理15.8.3. *If a model's excess entropy at the ith place in text is ϵ then there is a cloze question with binary answer such that the probability that the model answers it incorrectly is at most $\sqrt{2\epsilon}$.*

Proof. 证明涉及皮恩斯克不等式（维基百科版本），它将变分距离和KL散度联系起来。如第15.4节所述，令$p_i()$为人类对文本片段中$i+$第1个词的概率分布，$q_i()$为模型的分布。人类和模型给出不同答案的概率是两个分布之间的变分距离，即所有单词子集A中的$\sum_{w\in A}(p_i(w)-q_i(w))$的最大值。令$A_{i+1}$表示使前述表达式最大化的子集。完形填空题由将文本中的词w_{i+1}替换为问题：
*Is the next word among the words listed in option (a) or in option (b)*组成，其中选项(a)列出A_{i+1}中的单词，(b)列出A_{i+1}中的单词。现在定理可从皮恩斯克不等式得出。

——— □

15.8.1 Skills: A Statistical View

语言假定有一个潜在的集合 S，其中包含 *skills*。每个文本片段 t 都有一个与之相关的技能集，这些技能对于理解它是必需的。该理论允许这个技能集相当大——它只需要比分布中的文本片段数量（一个巨大的数字）小（相当多）。

定义15.8.4（技能图）。*A*技能图*is a bipartite graph* (S, T, E) *where nodes in S correspond to skills, nodes in T correspond to text-pieces, and* (s, t) *is in the edge set E if "comprehending" text-piece t (i.e., answering its associated cloze questions) requires using skill s. (See Figure* ??*)*

重要的是要认识到我们感兴趣的是量化模型在技能上的*competence*。例如，虽然上述定义假设文本片段的分布包括那些需要技能“代词消解”才能理解的部分，但语言模型（甚至人类个体！）通常无法在所有文本片段中正确应用这种技能。因此，“代词消解能力”不是0/1——相反，它被量化为与这种技能相关的文本片段中，模型正确回答的完形填空题的比例。量化这种（换句话说，模型的能力）的成功率是本文其余部分的目标。

我们的理论最终要素是技能图具有随机边，这在定义15.8.5中得到了精确描述。为了理解为什么这有意义，我们回忆Winograd的例子：
The city councilmen
refused the demonstrators a permit because they feared violence。Winograd隐含地假设这里最棘手的技能是代词/照应解析，但当然，在这个上下文中应用这种技能需要其他技能：对因果关系的理解（即“因为”的解释）以及关于“市政委员”、“许可”、“示威者”等的世界知识。这个例子突出了这样一个事实，如果我们观察需要代词消歧的随机文本片段，我们会遇到随机现实场景，其理解需要非常不同的技能组合。此外，这些场景（以及相关的技能）在语料库中出现的概率可能不同。

为了简单起见，我们假设每个文本片段恰好需要一些 k 的 k 技能，并且这个集合是从技能集合上的一个潜在测度中通过 iid 抽样得到的。（将 k 视为随机变量是自然的，但在此处不予考虑。）下一个定义将上述框架以 *skill cluster* 的形式形式化。

定义15.8.5（度-k技能簇）。*This is a skill graph* (S, T, E)

where the collection of text pieces is generated by "nature" by applying the following process: pick a subset of k skills via iid sampling from an underlying measure μ_1 on skills, and then use a procedure 代 *to create a text-piece t whose comprehension requires these skills, as well as a measure $\mu_2(t)$ associated*[25] *with this text piece t. Then nature uses process* 填空 *to add cloze prompts to test comprehension on t. The* 预测损失 *on the text-piece is the cross-entropy loss on predicting the answers to the cloze questions in it. The average prediction loss over all text-pieces is computed with respect to the measure $\mu_2()$. We call the skill-graph thus produced a* 度-*k* 技能簇*.*

25 注意，对文本片段的度量必须具有正确的边缘，例如，包含技能 s 的所有文本片段的 μ_2-度量是 $\mu_1(s)$。由于文本片段的数量远远大于技能的数量，因此有许多度量满足这个弱条件。

现在我们正式化一个关于全文语料库外观的简单模型。更复杂的框架扩展（例如，考虑语料库之间的层次结构）留待未来工作。

定义15.8.6. *(Text corpus) The text corpus consists of many skill clusters (e.g., math, newspapers, science, coding, etc.)* (S, T_1, E_1), (S, T_2, E_2), ... *which share the same underlying set of skills S but have disjoint sets of text-pieces T_1, T_2, ... that are generated as in Definition 15.8.5.*

定义15.8.5允许我们在更熟悉的统计学习理论背景下定义“对技能的胜任能力”，具体来说，是通过让我们将一个统计任务与之关联。该任务涉及预测包含该技能的文本片段子分布中的完形填空问题的答案。我们的涌现理论将适用于下一个定义的任务家族。

定义15.8.7（技能能力）。*In the setting of Definition 15.8.5, for each skill cluster and each skill $s \in S$*与*s*和此簇 *is defined as follows. The learner is given a text-piece created by sampling s_1, ..., s_{k-1} via iid sampling $(k-1)$ times from measure μ_1, and applying* gen *and* cloze *to the skill-tuple $(s, s_1, ..., s_{k-1})$ to convert it into a text piece t with an associated measure $\mu_2(t)$ (but the measure is re-scaled so that the total measure of the inputs to this task τ_s is 1). The* 错误率 *of the model at the statistical tasks is the expected prediction loss on text-pieces drawn from the above distribution. Since error rate is between 0 and 1, the* 能力 *refers to (1 − error rate).*

For every k′-tuple of skills $(s_1, s_2, ..., s_{k'})$ (where $k' \leq k$) the statistical task $\tau_{s_1,s_2,...,s_{k'}}$ corresponding to that k′-tuple is similarly defined. The inputs to the task are generated by completing the k′-tuple to a k-tuple $\vec{s}$ by iid sampling of $k - k'$ additional skills from μ_1 and then using gen *and* cloze *to convert it into a text-piece.*

Competence on the k′-tuple is defined just as above. [26]

注意：该定义涉及从μ_1中通过独立同分布抽样选择*k*-元组，原则上允许选择两次技能。

26 因此，如果一个文本片段涉及5项技能，那么该文本片段将出现在与单个技能对应的5个统计任务中，与技能对对应的$\binom{5}{2}$个任务中，依此类推。然而，我们在这些统计任务上衡量损失的方法隐含地假设，如果模型错误地回答了这个问题（即，它将显著的概率分配给了错误答案），那么这种损失就发生在*all*这些统计任务中。这是对技能能力的保守估计。

然而，选择相同技能两次的概率按 $O(1/|S|)$ 缩放。由于假设技能集 S 很大，分布几乎与采样不同的 k-技能元组相同。两种方法之间的微小差异 $O(1/|S|)$ 不会影响任何随机图论计算。

15.9 *Analysis of Emergence (uniform cluster)*

我们已到达关于涌现的核心数学问题：*As the model's excess cross entropy goes down (due to scaling), this improves the model's performance on cloze tasks inserted in the test stream (Assumption 15.8.2). How does this improve competence on the skills as well as on tuples of skills –in other words, performance on the associated cloze questions?*

本节分析了一个简单的设置，其中测试流由一个单度-k技能簇组成，技能均匀分布，文本片段也是如此——换句话说，定义15.8.5中的分布μ_1和μ_2是均匀的。第??节将分析扩展到一般设置。下面的计算只需要技能的总数远小于文本分布的支持大小——换句话说，技能集可以非常大。

假设该文本片段上的模型 *makes a mistake*，如果该文本片段的所有填空题[27]的总预测损失至少 1/2. 随着模型规模的扩大，存在两个不同的阶段。

Phase 1: In every text-piece, the error on its cloze questions exceeds 1/2 . 因此，该模型在任何技能上都没有发展出能力，因为它在每一篇文本片段中都犯错误。

Phase 2: On the average text-piece, the total error on all cloze questions is less than 1/2. 现在，人们开始在一些技能上看到非平凡的竞争力。以下分析将开始。

如果文本片段的平均交叉熵损失为 δ，我们得出结论 Y 至多包含 2δ 的文本片段比例。以下结果保证，对应于大多数技能的统计任务不会给 Y 中的文本片段分配显著的概率——换句话说，该模型在这些技能相关的统计任务上表现良好。

定理15.9.1（基本）。*Let* α，β，$\theta > 0$，$\beta > 1$，$\alpha\beta < 1$，$\theta < 1$ *satisfy*

$$H(\theta) + k\theta\left(H(\beta\alpha) - \beta\alpha \log\frac{1}{\alpha} - (1-\beta\alpha)\log(\frac{1}{1-\alpha})\right) < 0 \quad (15.16)$$

and the distribution on skills and text pieces be uniform in the skill-cluster. Then irrespective of the details of 生成 *and* 填空 *processes, the following*

27 这是在即使只有一个填空题被以明显的概率选择错误答案时产生的错误量。

property holds for every subset Y of text pieces that contains at least θ fraction of text pieces: at least $1-\alpha$ fraction of skills have at most $\beta\theta k|T|/|S|$ edges to Y (in other words, at most β times the number of edges a skill would be 预期 *to have to text-pieces in Y).*

注意：由于技能节点 s 和集合 Y 之间的边对应于统计任务 τ_s 中的错误，定理15.9.1给出了在（$1-\alpha$）技能分数上的能力下界：它至少为 $1-\beta\theta$。

15.9.1 Proof of Theorem 15.9.1.

证明使用了著名的 *Probabilistic Method* [28]。在这里，人们试图在某个概率空间中，不存在 *bad* 个结果。用 W 表示一个整数随机变量，表示坏结果的数量，如果我们证明 $\mathbb{E}[W] \approx 0$，那么至少以概率 $1-\mathbb{E}[W]$，$W=0$。具体来说，在证明中，W 将是违反引理的某个大小的“坏”集合对 (Y, Z) 的数量。

28 N Alon 和 J Spencer. *The Probabilistic Method (4th Ed)*. Wiley, 2016

Proof. 对于 $Y \subseteq V_1$、$|Y| = \theta|T|$ 和 $Z \subseteq S$，我们说 (Y, Z) 是 *bad*，如果 Z 至少有 $\alpha\beta\theta k|T|$ 条边到 Y。用 W 表示这样的 Z 的数量。期望值被上界限制为

$$|S||T|\binom{|S|}{\alpha|S|} \times \binom{|T|}{\theta|T|} \times \binom{k\theta|T|}{\beta\alpha k\theta|T|} \times \alpha^{\beta\alpha\theta k|T|} \times (1-\alpha)^{(1-\beta\alpha)\theta k|T|} \tag{15.17}$$

对于(15.17)要成为$\ll 1$，只需其对数是负数。根据Stirling近似 $\binom{N}{tN} \leq 2^{(H(t)+\epsilon_N)N}$ 其中$H(t) = -t\log t - (1-t)\log(1-t)$是二进制熵函数，$\epsilon_N$随着$N \to \infty$迅速趋近于零。将此应用于(15.17)并取对数，并假设$|S| \ll |T|$，我们得到大$|T|$的条件(15.16)。

□

15.9.2 Competence on skills and tuples of skills: Performance Curves

定义15.9.2（性能曲线）。*The contour plot (i.e., the boundary) of the region of α, β combinations satisfying Theorem 15.9.1 is called a* 性能曲线*and denoted $C_{(k,\theta)}$. A performance curve C is*更好，*than another curve C' if for every α β on C there is a corresponding point β') on C' for $\beta' > \beta$.*

我们通过在水平轴上绘制（$1-\alpha$）和在垂直轴上绘制 $\beta\theta$ 来绘制性能曲线，因此曲线上（0.8，0.16）的点表示至少 0.8 的技能最多有 0.16 的边在“错误集”Y (中，因此它们的边有 0.84 的部分在错误集之外)。出现曲线明显下移（即，

暗示随着 k 的增加会出现更多技能）的趋势。下一个引理表明这一趋势始终成立。

引理15.9.3（单调性）。*If $\theta' < \theta$ then the performance curve for θ', k lies below that for θ, k.*

If $k' > k$ then the performance curve of θ, k' lies below that for k, θ.

Proof. 从以下事实中得出：$H(\theta)/\theta$ 在区间 $(0, 1)$ 上是一个递减函数。 □

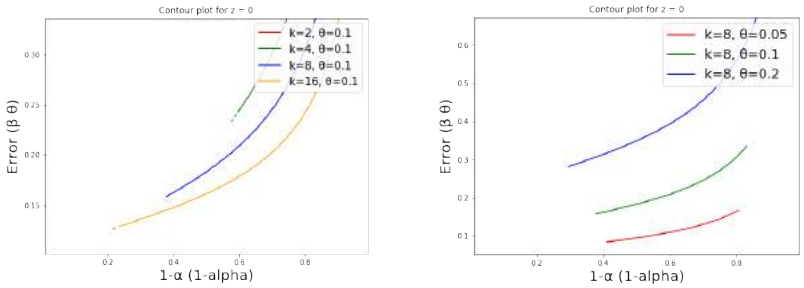


图15.2：性能曲线：左侧图具有 $\theta = 0.1$，$k =$ 变化 2，4，8，16。k 的更高值大大提高了性能（对于 $k = 2$ 个有效的 α，β 不存在）。右侧图具有 $k = 8$ 和 $\theta = 0.05$，0.1，0.2。第 ?? 节阐明，它还描述了模型对于 t-技能元组的性能曲线，对于 $\theta = 0.05$ 和 $t = 1$，2，4 分别（例如，蓝色曲线为 4 元组）。

15.9.3 The tensorization argument

当上述方法产生性能曲线时，可以通过张量论论证得到更好的曲线。考虑以下 *k'-wise recombination* 操作在测试流上。首先将测试流随机划分为大小为 k' 的子集，然后将每个子集中的 k' 文本片段连接起来，创建一个更大的文本片段，我们称之为“k'-片段”，其度量是组成测试片段的度量的总和。保留所有旧测试片段的完形填空题，不插入任何新完形填空题。显然，如果模型每个平均文本片段的错误为 δ，那么每个平均 b-片段的错误为 $k'\delta$。然而，每个 k'-片段现在正在使用一个随机的 $k'k$-元组技能。重要的是，这个 $k'k$ 技能集是由从技能分布中抽取的独立同分布样本组成的，这也可以看作是一个 k-元组 k'-元组。因此，将技能的 k'-元组视为“复杂技能”，我们可以将这些复杂技能作为定理15.9.1中的设置中的技能集，这给我们提供了一个容易的推论，量化了与技能 k'-元组相对应的任务的性能。

引理15.9.4（技能{v*}元组的涌现）。*Consider the skill- graph (S', T', E) where S' consists of all k'-tuples of skills, T' consists of k'-pieces, and E consists of (s', t') where s' is a k'-tuple of skills and t' is a k'-piece where this tuple of skills is used. Let Y consist of θ fraction of k'-pieces. Then for any α, $\beta > 0$, $\beta > 1$, $\alpha\beta < 1$ satisfying (15.17) there are at least $1 - \alpha$ fraction of k'-tuples of skills that have at most $\alpha\beta\theta\theta N_1$ $\beta\theta$ fraction of their edges connected to Y.*

下一个问题要求你推导出一个多少有些令人惊讶的普遍原理，这个原理也在图15.2的标题中有所暗示。为了简单起见，假设一个类似长尾兔的缩放定律，即10倍放大导致过剩熵减少2倍。如果一个模型在当前缩放下被认为在个别技能上表现合理，那么在进一步放大10倍后，将在技能对上看到类似合理的表现，在那之后再次放大10倍将产生在技能4元组上类似合理的表现，等等。请注意，这些都是关于性能提升的*provable lower bounds*——实际提升可能更高。图15.2说明了这一现象。

问题15.9.5. *Show that when a model M_1 with loss δ is scaled up (e.g., as per equation (15.8)) so that the new model M_2 has loss δ/k', then the performance curve inferred by our method for k'-tuples of skills using M_2 is identical to the curve inferred for individual skills on model M_1.*

16
Generative Adversarial Nets

第14章描述了一些生成模型的经典方法，这些方法通常使用对数似然方法进行训练。我们还看到，它们通常不足以用于学习复杂分布，如真实图像分布的高保真学习。*Generative Adversarial Nets (GANs)*是一种生成更真实样本的方法。它依赖于深度网络在判别任务上的能力。为了方便起见，在本章中我们假设感兴趣的数据是图像，我们将它们视为某些d中的点。该模型试图生成逼真的图像。

在开发GANs理论之前，我们调查了各种测试两个分布相似性的概念，因为这一讨论直接影响到GANs中使用的设置。这很重要，因为示例14.2.1说明了从监督学习中得到的*generalization*这一概念在推理分布学习正确性时可能会让我们误入歧途。

16.1 Distance between Distributions

如何衡量两个分布 P 和 Q 之间的差异？如果我们能访问计算每个分布密度的公式，那么可以计算任何合适的 f：$\Re \to \Re$ 的 f-散度，该散度是凸的。

$$D_f(P||Q) = \int f(\frac{P(x)}{Q(x)})Q(x)dx \tag{16.1}$$

问题16.1.1. *(i) Show that f-divergence is nonnegative. (ii) Show that the f-divergence for $f(t) = t \log t$ coincides with $KL(P||Q)$. (iii) Show that the f-divergence for $f = \frac{1}{2}|t-1|$ coincides with total variation (or ℓ_1) distance: $|P-Q|_1 = \int |P(x) - Q(x)|dx$.*

然而，在实践中，人们没有概率密度函数的公式，必须使用来自 P 的样本来估计距离

并且 Q。一个自然的想法是将两类 *test* 函数在两个分布上的期望进行比较。

示例16.1.2. *The expectation $\mathbb{E}_P[x]$ is the mean of the distribution P. Similarly expectation of monomials of form $x_{i1}x_{i2}\cdots x_{ik}$ constitutes the kth moment. Moments can be estimated from samples (under fairly general conditions) and the difference of moments of distributions P, Q can be seen as some measure of their difference.*

很遗憾，对于多元分布的所有高阶矩的准确估计在计算时间和样本复杂度方面都会变得昂贵。这促使在 *transportation metrics* 中出现了一种距离的概念。1 如果 $\mathcal{F}$ 是一个函数类，那么

1 请在网上查找Wasserstein度量法和地球迁移距离。

定义 P 和 Q 之间的距离，使用在 $\mathcal{F}$ 中函数可达到的最高期望差异。

$$d(P,Q) = \sup_{f\in\mathcal{F}} |\mathop{\mathbb{E}}_{x\sim P}[f(x)] - \mathop{\mathbb{E}}_{x\sim Q}[f(x)]| \tag{16.2}$$

例如 $\mathcal{F}$ 可以是次数最多为的多项式 k.

问题 16.1.3. *Show that the distance defined above satisfies triangle inequality.*

16.2 Introducing GANs

生成对抗网络（Goodfellow等人2）是一个框架，用于通过(16.2)的定义学习生成模型。具体来说，试图训练一个生成模型G，该模型（如第14章所述）使用随机向量u生成一个图像$G(u)$。这产生了一个图像分布，我们通过使用一个具有固定大小和架构的深度网络类$\mathcal{F}$来检查这个分布的质量，并通过尝试找到一个最大化该表达式的网络来估计(16.2)中的距离。现在我们详细说明GANs的主要组成部分。

2 I Goodfellow, J Pouget-Abadie, M M irza, B Xu, D Warde-Farley, S Ozair, A Courville, and Y Bengio. 生成对抗网络。*NeurIPS*, 2014

Idea 1: Since deep nets are good at recognizing images —e.g., distinguishing pictures of people from pictures of cats—why not let a deep net be the judge of the outputs of $G()$?

更具体地说，假设图像被表示为 $\Re^d$ 中的向量，并令 P_{real} 为真实图像的分布。生成器 G 已经学会了从新的分布 P_{synth} （（即当 h 是一个随机种子) 时的 $G(h)$ 分布）生成合成图像。我们可以尝试训练一个将图像映射到 [0, 1] 范围内数字的判别器深度网络 D，并试图以下述方式区分这些分布。当 x 从 P_{real} 中抽取时，其期望输出 $E_x[D(x)]$ 尽可能高，而当 x 从 P_{synth} 中抽取时，其期望输出尽可能低。判别器可以被训练

使用标准监督学习（例如，回归）具有两个标签。如果 $P_{synth} = P_{real}$，那么当然没有任何分类器能够在这个预期输出中实现差距，因此训练将失败。另一方面，如果我们能够训练一个优秀的判别器深度网络——其平均输出在真实样本和合成样本之间有明显的差异——那么这充分证明了这两个分布是不同的。3

存在一种中间情况，其中分布不同，但判别器网络没有检测到差异。这种情况在故事中很快就会变得很重要。

Idea 2: If a good discriminator net has been trained, use it to provide "gradient feedback"that improves the generative model.
自然目标对于生成器是使$E_h[D(G(h))]$尽可能高，因为这意味着它更能欺骗判别器D。给定一个固定的D，提高G的自然方法是选择几个随机种子h，并略微调整G的可训练参数以增加这个目标。请注意，这个梯度计算涉及到通过组合网络$D(G(\cdot))$的反向传播)。4

4 这些对 G 的更新假设 D 已修复，反之亦然。许多论文提出了替代的更新方法，这些方法预测 D 对此更新的响应，并展示了这使实际训练更稳定的证据。参见问题 16.2.1。

Idea 3: Turn the training of the generative model and the accompanying discriminator net into a game of many moves (i.e., rounds of parameter updates).
每次判别器的移动都包括从 P_{real} 和 P_{synth} 中取一些样本，并提高其区分它们的能力。每次生成器的移动都包括从 P_{synth} 中生成一些样本，并更新其参数，以便 $E_u[D(G(h))]$ 稍微上升一点。

注意，判别器始终将生成器视为一个黑盒——即从不检查其内部参数——而生成器需要判别器的参数来计算其梯度方向。具体来说，G的梯度是通过反向传播通过D来计算的。此外，生成器在计算过程中从不使用来自P_{real}的真实图像。（尽管当然它间接依赖于真实图像，因为判别器是使用它们进行训练的。）

一个人可以用多种方式填充上述框架。最明显的是，生成器可以尝试最大化 $E_u[f(D(G(h)))]$，其中 f 是某个递增函数。（我们称之为*测量函数*。）具体来说，如果 D、G 是具有指定架构的深度网络，并且其参数数量由算法设计者在事先固定，那么训练目标就是：

$$\min_G \max_D \quad E_{x\sim P_{real}}[f(D(x))] + E_h[f(1 - D(G(h)))]. \tag{16.3}$$

问题16.2.1. *(1) Write an expression for updates for G and D for the loss in (16.3) when f is the identity map (i.e. $f(x) = x$). (2) Write an expression where G and D anticipate the effect of the other's immediate response to their update. (This can be done in more than one way.)*

f的影响：测量函数具有对不同样本赋予不同重要性的作用。Goodfellow 等人最初使用了 $f(x) = \log(x)$，由于 $\log x$ 的导数是 $1/x$，这隐含地赋予了合成数据 $G(u)$ 更多的重视，其中判别器输出非常低的值 $D(G(h))$。换句话说，使用 $f(x) = \log x$ 使得训练对判别器认为很糟糕的实例比对判别器认为一般般的实例更敏感。相比之下，$f(x) = x$ 对所有样本赋予相同的重要性，并导致 *Wasserstein GAN*。

问题16.2.2. *Show that if the discriminator has unbounded capacity (i.e., able to compute any function) then for* $f(x) = \log x$ *the optimum value of the expression in (16.3) is the following quantity (called* Jensen-Shannon 散度*) where* $\mu = P_{real}$，$\nu = P_{synth}$ *and KL was defined in Section 5.6:* [5]

$$KL(\mu \| \frac{\mu + \nu}{2}) + KL(\nu \| \frac{\mu + \nu}{2}).$$

5 提示：最优 D 将会过于强大：给定输入 x，其输出取决于概率 $P_{real}(x)$、$P_{synth}(x)$。

16.2.1 Game-theoretic interpretation and implications for training

一个实施上述训练的严重实际困难是它可能会出现振荡，这意味着上述目标会上下波动。这与通常的深度网络训练不同，在通常的深度网络训练中（至少在它起作用的情况下），训练会稳步提高目标。原因是隐式地，判别器和生成器在进行一个两人零和博弈[6]，他们的“动作”是两个电路

6 请在网上了解零和博弈，包括著名的关于均衡的Min-Max定理。

D, G 以及生成器（最小化器）到判别器（最大化器）的收益是损失。因此，生成器正在选择动作以最小化以下收益

$$\max_D \quad E_{x \sim P_{real}}[f(D(x))] + E_h[f(1 - D(G(h)))]$$

而判别器正在最大化

$$\min_G \quad E_{x \sim P_{real}}[f(D(x))] + E_h[f(1 - D(G(h)))].$$

此类游戏并不总是有一个 *equilibrium,*，即双方玩家都最优地反应对方，因此缺乏改变的动力。（这与优化中的鞍点类似。）

尽管从两人博弈的角度看可能不存在均衡，当然在训练过程中，两位玩家都受到训练算法的控制。因此，对基于梯度的训练进行适当的修改，理论上可以使收敛到某些解，即使它不是一个均衡。（例如，即使 D 不是对当前 G 的最优反应，梯度可能不允许改进当前 D 的方法。）大量论文提出了这样的想法；一些例子包括：此处需要一些参考文献

16.3 "Generalization" for GANs vs Mode Collapse

第14.2.1节讨论了当我们试图从有限样本中学习分布时出现的复杂性。在GAN设置中，训练目标（16.3）使用完整分布进行了描述，但当然在实践中，判别器 D 是在两个分布的有限样本之间进行区分。

通常在机器学习中，良好的泛化意味着测试数据集上的平均损失函数与训练数据集上的相似。然而，我们发现对于像对数似然这样的常用损失函数，这种泛化概念并不一定意味着已经很好地学习了分布。在引入GAN之后，对GAN方法能否绕过这些问题进行了广泛的研究，并且在这个努力中尝试了大量的训练目标和算法。有人指出，它们对分布的学习相当不完美7，但尚不清楚这是否会

离开更大的训练数据集或不同的目标。

collapse

7 一个代表性问题是 *mode*：学习到的分布似乎不具有与以下相同的多样性

示例16.3.1. *Since the objective (16.3) allows maximization over all neural nets D of the allowed architecture, even two different samples from the same distribution can look very different to a neural net. For example if we take two finite samples from the d-dimensional Gaussian $\mathcal{N}(0, I)$ then even if the samples have size say d^3, they are distinguishable by a deep net that is somewhat larger than d^3. The reason is that the samples are two discrete sets in which all pairs of points are almost orthogonal (with high probaility). While this is an artificial example, it is the case that in real-life GAN training, the objective does not usually drop to zero —the net is indeed able to somewhat distinguish the samples from the two distributions.*

我们现在描述了来自8的理论分析，表明
学习分布的质量本质上受判别器 *represen- tational capacity* 的 *regardless of how large we make the the training dataset* 限制。

8 Sanjeev Arora, Rong Ge, Yingyu Liang , Tengyu Ma 和 Yi Zhang. 生成对抗网络（GANs）中的泛化和均衡。*Proc. ICML*，2017

通常在讨论泛化时，我们假设网络具有某种有限大小，比如说 N ，并且分布的样本大小足够大，以便 N 大小的网络能够泛化。以下两个问题是从以下内容中提取的

问题16.3.2. *Suppose the loss is C-Lipschitz and the parameter vector is in $\Re^d$ and has ℓ_2-norm at most L. Suppose this discriminator trained on a training set of size N achieved training loss at most ϵ_1. Show that if $N > \Omega(\frac{L}{\epsilon_2^2}\log(C/\epsilon_2))$ then it has test loss at most $\epsilon_1 + \epsilon_2$ on the full distribution.* [9]

9 提示：使用第5章的结果，例如定理5.2.7。

这难道意味着 GANs 确实学会了分布？不，就像在示例 14.2.1 中一样：泛化只意味着训练和测试损失接近，并不意味着分布接近。

问题16.3.3. *Under the same conditions as Problem 16.3.2 show that if the learned distribution P_{synth} has finite support and is a uniform distribution on $\Omega(\frac{L}{\epsilon_2^2}\log(C/\epsilon_2))$ random iid samples from P_{real} then no discriminator can achieve test loss more than $\epsilon_1+\epsilon_2$ when comparing the distributions P_{synth} and P_{real}.*

在先前的问题中，让我们将 P_{real} 视为所有真实图像上的分布。然后 P_{synth} 与 P_{real} 非常不同：它是某些小集合上真实图像的均匀分布。尽管如此，没有任何判别器（其大小、范数和Lipschitz常数都适当地有上界）能区分这两种分布。正如我们将看到的，这种 P_{synth} 在实践中确实被观察到。此外，对 GAN 架构的更改（例如编码器-解码器 GAN）不会影响这个基本结果 10。

10 GANs是否学习分布？一些理论与实证

以下练习给出了一个（可能非常宽松的）生成器大小的上界，该生成器可以产生这样的 P_{synth}。

问题 16.3.4. *Show that the uniform distribution on M images where each image is in $\Re^d$ an be generated using a net with $O(Md^2)$ parameters.* 11

11 更确切地说，分布将是围绕 M 图像的具有微小方差的高斯混合分布。

将前三个问题结合起来，我们得出以下结论：在以下场景中，GAN 目标不足以防止验证器学习一个支持在有限小图像集上的分布（“模式崩溃”）：生成器的“容量”略大于判别器。12

12 这是一个开放性问题，即当判别器很大时是否可以避免模式坍塌，尽管在这种情况下，通常难以降低训练损失，原因在示例16.3.1中已探讨。

16.3.1 *Experimental verification of Mode Collapse: Birthday Paradox Test*

上述理论表明，使用具有某种“容量”（在泛化理论意义上）的判别器训练的GAN具有低训练和测试误差的解，其中合成分布P_{synth}支持在少量图像上，因此与P_{real}相当不同。GAN最终以少量图像组成的合成分布结束的现象被称为*mode collapse*，之前被认为是由训练失败或使用过小的真实图像训练集引起的。上述结果表明，对于容量较低的生成器和判别器（与P_{real}中不同模式的数量成比例的容量相反），模式坍塌并不是一个令人惊讶的结果。

这引发了一个问题，即我们是否能够在现实生活中的GANs中检测到这种模型崩溃。换句话说，估计它能够产生多少“独特”的图像。乍一看，这样的估计似乎非常困难。毕竟，图像相似度的自动化/启发式度量很容易被欺骗，而我们人类肯定也不够

时间来处理数百万或数十亿张图片，对吧?

幸运的是，可以使用简单的生日悖论进行粗略估计，这是大学本科离散数学的基础。

问题 16.3.5（生日悖论）。[13] *Consider a uniform distribution on a set of size N. Show that a random sample of size $2\sqrt{N}$ contains a duplicate probability at least $1 - 1/e$. (The name for this paradox comes from its implication that if you put 23 random people in a room, then the odds are good that two of them have the same birthday is significant.)*

[13] GANs是否学习分布？一些理论与实证

让我们实现 GANs 的含义。为了辩论的目的，假设 P_{real} 是人脸图片的分布。这个分布中有多少个模式？至少是不同人脸的数量吗？这感觉像是一个相当大的集合，因为我们都知道成千上万的面孔（包括新闻中遇到的）并且没有看到任何无关的 *doppelgangers*；只有双胞胎。更精确地说，生日悖论表明，如果不同人脸的数量是 N，那么在看到 $\sqrt{N}$ 张面孔之后，我们预计会看到双胞胎。

在 GAN 设置中，分布是连续的，而不是离散的。因此，我们提出的 GAN 生日悖论测试 [14] 如下。

(a) 从生成的分布中选取大小为 s 的样本。 (b) 使用自动化的图像相似度度量来标记样本中20（例如）最相似的成对。 (c) 视觉检查标记的成对，并检查人类认为接近重复的图像。 (d) 重复。

[14] Sanjeev Arora 和 Yi Zhang. 编码器-解码器 GAN 架构的理论限制。*Proc. ICLR*，2018

如果这个测试表明大小为 s 的样本有很高的概率存在重复图像，那么怀疑分布的支持大小约为 s^2。

16.3.2 Other notes on GANs and mode collapse

尽管最近的 GAN（如渐进式 GAN）使用非常大的判别器和生成器来产生更高质量的图像（根据人类判断），但它们仍然受到模式崩溃的影响，这与上述理论一致。

另一方面，Florian等人[15]认为上述分析采取假设训练中的静态近平衡，而现实生活中的训练永远不会达到平衡，由非平衡产生的训练动力学可以作为GAN行为的强大塑造者。

[15] F Schaefer, H Zheng, 和 A Anand-kumar. GANs中的隐式竞争正则化。*ICML*，2020

最近的一篇论文[16]表明，尽管GANs存在模式问题崩溃，它们可以用来预测泛化。换句话说，给定一个数据集 S 和一个在该数据集上训练的判别模型，使用以下泛化误差预测器：训练一个条件 GAN

[16] Y Zhang, A Gupta, N Saunshi, and S Arora. On predicting generalization using gans. *ICLR*, 2022

使用 S 并使用训练好的 GAN 的随机样本代替保留数据来预测泛化。这被证明比其他泛化误差预测器效果更好。

GANs的基本思想已被扩展到其他设置，最显著的是学习良好的图像到图像映射（例如，将照片转换为绘画，或虚拟试穿衣服在人的图像上）。这效果非常好，没有模式崩溃结果的类似物。

17
Self-supervised Learning

语义表示（即 *semantic embeddings*）在复杂数据类型（例如图像、文本、视频）中已成为机器学习的核心，也出现在机器翻译、语言模型、GANs、领域迁移等中。这些涉及学习一个表示函数 f，使得 $f(x)$ 是数据点 x 的紧凑且“高级”表示——意味着它保留了语义信息，同时丢弃了低级细节——例如，图像中单个像素的颜色。一个好的表示的测试是，它应该通过允许通过线性分类器（或其他低复杂度分类器）使用少量标记数据来解决新分类任务，从而极大地简化解决新分类任务。

研究人员最感兴趣的是使用未标记数据的无监督表示学习。一个流行的早期例子是 *word em- beddings*，它使用了简单的线性代数[1]，并在
信息检索几十年。更近期的词嵌入如word2vec[2]成为了语义的灵感来源
嵌入各种数据类型的表示，如分子、社交网络、图像、文本等。

[1] LSI paper

[2] word2vec 论文；查看word2vec的维基百科页面以获取其他类似算法的链接

在这一章中，我们遇到了*self-supervised*学习，这是一组用于学习良好表示的方法。在处理未标记数据时，学习器定义了一个寻找良好表示函数的学习目标。与书中其他地方研究的学习范式的一个重要区别是，训练和测试任务是不同的，因此泛化的概念并没有捕捉到学习的最终目标。这是一个*training on task A to later do well on task B*的例子，人们想象这实际上是智能行为的一个重要方面。

18
Adversarial Examples and efforts to combat them

尽管现代深度网络在解决图像分类任务时表现出超越人类的准确性，但它们有一个令人惊讶的阿基里斯之踵，这首先在1：中被报道，对于大多数正确分类的图像 x

存在一个小的扰动向量 δ，使得 $x+\delta$ 被深度网络错误分类，而对于人类来说 $x+\delta$ 与 x 看起来相当相似。这些被称为 *adversarial examples*；请注意 δ 是特别的

1 C Szegedy, W Zaremba, I Sutskever, J Bruna, D Erhan, I Goodfellow 和 R Fergus. 神经网络的有趣性质。在 *ICLR*，2014

图18.1：*Flying pigs?* (A)是猪的图像，而（B）是它的略微扰动版本。一个正常训练的ResNet50分类器将（B）标记为“飞机”。这两张图像之间的差异很小；在（C）中，您可以看到一张图像，其像素级差异是（A）和（B）的50倍。如果没有50x缩放，（C）将包含接近0的像素值（即空白图像）。
Source: Kolter-Madry Tutorial.

构建了给定的 x 使用优化技术，因此 $x+\delta$ 并非来自分类器训练的常规输入分布。尽管如此，$x+\delta$ 被错误分类仍然令人惊讶，尽管在我们人类看来它看起来像是一张正常图像。

对抗样本在各种数据集和神经网络架构中得到了广泛记录。已经发现了强大的方法（基于优化）来寻找这样的样本。本章的概念和定义与Kolter和Madry的在线教程2紧密相关。因为这种现象暗示

2 Kolter Z 和 Madry M. 对抗鲁棒性 – 理论与实践

当前基于机器学习的系统脆弱性，已经尝试了无数方法通过改变训练协议来缓解这个问题，尽管进展缓慢。

图18.2：一个对抗性的3D物体！这个贴了几张贴纸的停车标志可靠地欺骗了图像识别分类器，将其分类为45英里/小时的速度限制。

18.1 Basic Definitions

分类器 $f:\mathbb{R}^d \to \mathcal{Y}$ 将输入映射到有限集合 $\mathcal{Y}$ 中的标签。存在一个允许的集合 *perturbations* $\Delta \subseteq \Re^d$。在本章中，我们假设 Δ 是向量集合 ℓ_p 的范数不超过 ϵ 的向量，其中

p 是1，2，∞之一。*adversary*必须找到一个$\delta \in \Delta$，使得$f(x+\delta) \neq f(x)$。

targeted adversary 被赋予一个特定的目标标签 $y' \neq f(x)$ 并必须找到一个 $\delta \in \Delta$ 使得 $f(x+\delta) = y'$。*Black box attacks* 涉及一个不知道 f 内部参数向量的对手；对手只能向 f 提供输入并看到答案。*White box attacks* 允许对手访问内部参数向量。3

我们将关注白盒攻击。我们假设存在一个自然损失函数 {v*}，它给出了分类器 w 在输入 x 和标签 y 上的损失。

现在我们描述基本攻击和防御方法。

18.1.1 Attack method: PGD

一个代表性（且流行）的攻击方法使用 *Projected Gradient Descent (PGD)* 4，其中 $\text{Proj}_{x_0+\Delta}(x)$ 是类型 $x_0+\delta$ 的最近点到 x，其中扰动 δ 的 ℓ_p 范数不超过 Δ。此外，尽管 x 有标签 y，但分类器分配给 $x_0+\delta$ 的标签是 y'。

注意，如果范数为 ℓ_2，则投影 $_{x_0+\Delta}(x)$ 简单地是 $x_0 + \Delta\frac{(x-x_0)}{|x-x_0|_2}$。

问题 18.1.1. *Give methods to compute* $Proj_{x_0+\Delta}(x)$ *for norms* ℓ_1 *and* ℓ_∞.

现在我们可以描述寻找对抗样本最流行的方法。

PGD方法：*Given input* x_0*, do the following iteration k times:* $x \leftarrow x + \eta\nabla_x\ell(w, x, y)$*, followed by* $x \leftarrow Proj_{x_0+\Delta}(x)$。

注意梯度是关于输入 x 的，而不是参数向量 w！它可以通过对反向传播的简单修改来计算。5

针对针对性攻击，可以使用 $\nabla_x\ell(w、x、y) - \nabla_x\ell(w、x、y')$。 *ent*

深度网络以常规方式训练时非常容易受到此类攻击。对于大多数输入 x，上述算法等可以找到一个附近的点 $x+\delta$，分类器输出不同的标签。

18.1.2 Adversarial Defense

为了使网络对上述攻击具有一定的抵抗力，必须以不同的方式对其进行训练。具体来说，它使用对抗性示例进行训练，并教授其正确分类它们。使用一批输入，敌手（如上述敌手）用于生成对抗性示例。现在调整参数，使分类器在这些示例上输出正确的标签。然后，使用敌手为新的调整参数生成一组新的对抗性示例。这种来回重复进行一定次数。

3 虽然黑盒攻击看起来毫无希望，但在实践中它们确实存在。对手使用针对他们自己基于同一数据集训练的深度网络的白盒攻击来生成对抗图像。这些图像能够欺骗具有未知架构和参数的其他网络。黑盒攻击的成功表明，不同的架构在分类行为上相当相似。

4 A Madry, A Makelov, L Schmidt, D Tsipras, 和 A Vladu. 面向对对抗攻击具有抵抗力的深度学习模型。*ICLR*，2018

5 常见攻击使用所谓的 *sign gradi-*，将所有正坐标四舍五入到 +1，将所有负坐标四舍五入到 −1。

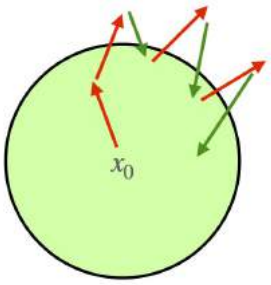


图18.3：对输入 x_0 的PGD攻击。红色箭头对应基于梯度的更新。当它们产生球面 Ball (x_0, r) 外的点时，投影操作（用绿色箭头表示）找到球面内的最近点。

时间。本质上，分类器正在被训练以将决策边界远离数据点；见图18.4。

上述协议有许多变体，但最终结果是效用/鲁棒性权衡：攻击者找到对抗性样本的能力大大降低，因此，它只能对大约40%的输入（例如）进行操作，而不是几乎所有的输入x。缺点是，在这个过程中，分类器在原始数据集（即通常的输入）上的整体准确率也大幅下降（例如，从95%下降到80%）。因此，鲁棒分类器在非对抗性设置中的效用显著降低。

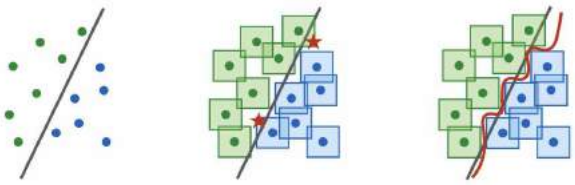


图18.4：ℓ_∞-范数扰动对抗样本的概念图示。在原始分类器中，大多数数据点距离决策边界很近，如ℓ_∞距离所测。红色星号是对抗样本。经过对抗训练后，数据点周围的小ℓ_∞-球体不再与决策边界相交。来源：Madry等人2018年。

实现这一点，即在训练结束时，分类器只有逃避特定对手（即训练时使用的攻击算法）生成的对抗样本的能力。通常使用 *differ- ent* 攻击算法可以生成新的对抗样本。这已经被证明过很多次，确实是一个非常令人沮丧的状态。

18.1.3 Other defense ideas

大量能量被用于通过在分类器内部构建某种形式的混淆来尝试避免对抗性攻击，通常是通过在分类器网络内部执行非可微变换。动机是使上述基于梯度的攻击难以实施。然而，大多数此类防御都因一些巧妙的方法而被破解；有关介绍，请参阅6。

6

18.2 Provable defense via randomized smoothing

由于许多防御实际上在几个月内就被破解了，因此似乎有必要采用更原则性的防御，并有一些关于安全性的数学证明。我们现在描述 *randomized smoothing*，已知这类防御中最好的一个类别——尽管它会导致分类精度的显著降低。为了简单起见，我们描述了这个想法

对于 ℓ_2-攻击。

定义18.2.1. *If $\mathcal{D}$ is a distribution on (input, label) pairs, where inputs are in $\Re^d$, then classifier g is γ-*在(x, y)上*if $f(x') = y$ for all x' such that $\|x' - x_0\|_2 \leq \gamma$.*鲁棒

7 我们没有调查基于混合整数规划的定理证明的可证明防御方法，这种方法也进行了广泛的研究，但到目前为止还没有达到随机平滑提供的保证。

随机平滑中的想法是通过取另一个分类器的局部空间平均输出来尝试产生一个鲁棒的分类器。下面，为了简单起见，我们将 ℓ_2-ball of radius $\beta\sqrt{d}$ around x 等同于高斯分布 $\mathcal{N}(x, \beta^2 I)$，因为这两个分布非常接近。

定义18.2.2. *If $\mathcal{D}$ is a distribution on (input, label) pairs and g is a classifier, then* 的 β-平滑 *g, denoted g^β, is the classifier that, given x, outputs the probabilistic answer $g(x+\delta)$ where δ is a random vector from $\mathcal{N}(0,$ $\beta^2 I)$. The classifier g^β_{smooth} is a classifier that on input x gives the* 多个标签*, namely, the label that is given highest probability by g^β. (It breaks ties among labels arbitrarily.)*

当定义 g^β 时涉及对连续分布的平均，而对于特定标签 y 的 $g^\beta(x) = y$ 可以通过采样以任意加性精度进行估计。分类器 g^β_{smooth} 的输出可以通过概率接近 1.8 的采样来确定。目标将是证明 g^β_{smooth} 是

γ- 对某些小的 γ 具有鲁棒性。

实际上，这需要在最流行标签的概率和第二流行标签的概率之间留出一点差距。

定理18.2.3. *If $g^\beta(x)$ outputs label y with probability p_a then $g^\beta(x')$ outputs y with probability at least*

$$\Phi(\Phi^{-1}(p_a) - \frac{1}{\beta}|x - x'|_2)$$

where $\Phi()$ is the cumulative distribution function[9] *of the univariate Gaussian $\mathcal{N}(0,1)$.*

[9] This means $\Phi(t) = \Pr_{z\sim\mathcal{N}(0,I)}[z \le t]$.

该定理在接下来的几段中得到证明。首先我们注意到以下简单推论。

推论18.2.4. *If $p_a = \Pr[g^\beta(x) = y]$ and all other labels are given probability at most p_0, then y is the label given highest probability by $g^\beta(x+\delta)$ provided $|\delta|_2 \le \frac{\beta}{2}(\Phi^{-1}(p_a) - \Phi^{-1}(p_0))$。*

让我们了解如何使用推论来训练一个对ℓ_2扰动鲁棒的分类器g。通常，x是训练集中带有标签y的输入，然后在正常训练中，你通过更新梯度来减少将标签y分配给x相关的损失总和。为了确保对ℓ_2扰动的鲁棒性，你将训练改为也将相同的标签y分配给随机样本的噪声输入$x+\delta$，其中$\delta \sim \mathcal{N}(0, \beta^2 I)$。

当训练结束时，使用保留数据估计保留图像中 ρ 的比例 x，其中 (a) g^β 的多数标签是正确的，并且 (b) 其概率 p_a 与下一个最有可能的标签的概率 p_0 之间存在明显的差距。然后，引理18.2.4表明，对于这样的点 x，分类器 g^β_{smooth} 对所有 x' 输出相同的标签 y，其中 $|x - x'|_2 \le \frac{\beta}{2}(\Phi^{-1}(p_a) - \Phi^{-1}(p_0))$。

Proof. (定理18.2.3) 设x'为x邻域内的任意一点，我们试图上界Pr[$g^\beta(x) = y$]和Pr[$g^\beta(x') = y$]之间的差异。然后从$\mathcal{N}(z$中采样一个随机u，$\beta^2 I_{d\times d})$可以等价地看作首先根据一元高斯$\mathcal{N}(z,\ \beta^2)$沿$x' - x$选择这个向量的投影，然后根据$d-1$维高斯$\mathcal{N}(z,\ \beta^2 I_{d-1\times d-1})$选择垂直于$x - x'$的向量剩余部分。假设$z$是穿过$x$，$x'$的无穷线的投影。定义$E(z) = \int_u 1_{g(u)=y} du$，其中$\int_u$在这样一个超平面z上的$d-1$维分布$\mathcal{N}(0$，$\beta^2 I)$上积分。那么我们有

$$\Pr[g^\beta(x) = y] = \int_z E(z)dz \tag{18.1}$$

在 $\int_z$ 对单变量高斯密度 $\mathcal{N}(x,\ \beta^2)$ 进行积分的地方，对于 Pr[$g^\beta(x') = y$]，有相应的表达式，其中 $\int_z$ 对以 x' 为中心的单变量高斯密度 $\mathcal{N}(x',\ \beta^2)$ 进行积分。假设 x 在此行的 x' 的左侧，这两个之间的最大差异是多少?

直观上看，似乎很清楚，当 $E(z)$ 取其可能的最大值，即接近 x 的点上的 1 时，$\Pr[g^\beta(x) = y] - \Pr[g^\beta(x') = y]$ 达到最大值，然后开始切换到更接近 x' 的较低值。假设这种直觉是正确的[10]，让我们看看它在 x' 处能降低到多低。

最坏的情况必须是在 $E(z)$ 为 1 时，对于 $z = x$ 到 $z < x + \beta\Phi^{-1}(p_a)$，而在其右侧为零。然后由于 $g^\beta(x') = g^\beta(x + x' - x)$，相同的 $E(z)$ 以不同的权重进入 x' 的平均值，对应于标准正态分布中 $|x - x'|/\beta$ 个标准差的位移。因此，在这种最坏的情况下，$\Pr[g^\beta(x') = y]$ 等于 $\Phi(\Phi^{-1}(p_a) - |x - x'|/\beta)$，并且通常至少等于这个值。

□

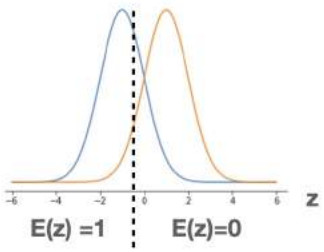


图18.5：以 {v*} 为中心的单变量高斯分布，蓝色一个) 和红色一个 x' (，以及 $E(z)$ 从1切换到0的点。

10 这直觉的正确性由统计学的 *Neyman Pearson lemma* 所暗示。

上述分析可以稍微收紧一些；参见11。虽然上述参数依赖于高斯分布与 ℓ_2 距离的紧密关系，它可以扩展到使用 ℓ_p 距离的适当类似物进行 ℓ_p 有界攻击。

11 H Salman, G Yang, J Li, P Zhang, H Zhang, I Razenshteyn, 和 S Bubeck. 通过对抗训练的平滑分类器实现可证明鲁棒的深度学习。*NeurIPS*, 2019

问题 18.2.5. *If g is a function mapping data points in $\Re^d$ to $\Re$ and $g(x) \leq 1$ for all x, then show that there is a constant C (independent of d) such that the gradient of g^1_{smooth} has norm at most C.*

19
Examples of Theorems, Proofs, Algorithms, Tables, Figures

在这一章中，赵提供了许多事物的例子，如定理、引理、算法、表格和图。如果有任何疑问，请直接联系赵。

19.1 Example of Theorems and Lemmas

我们提供一些示例

定理19.1.1 (*d*-维稀疏傅里叶变换). *There is an algorithm (procedure* FourierSparseRecovery *in Algorithm 5) that runs in ??? times and outputs ??? such that ???.*

请注意，通常，如果我们提供定理/引理的算法。定理应尝试引用相应的算法。对于定理/引理/推论等的名称，我们只需将首字母大写，

引理19.1.2（梯度的上界）。*Blah blah.*

问题 19.1.3. *This is how you put in a problem. It inherits chapter and section numbers.*

定理19.1.4（主要结果）。

19.2 Example of Long Equation Proofs

我们可以按以下方式重写 $\|Ax' - b\|_2^2$,

$$\begin{aligned}\|Ax' - b\|_2^2 &= \|Ax' - Ax^* + AA^\dagger b - b\|_2^2 \\ &= \|Ax^* - Ax'\|_2^2 + \|Ax^* - b\|_2^2 \\ &= \|Ax^* - Ax'\|_2^2 + \mathrm{OPT}^2\end{aligned}$$

在第一步由 $x^* = A^\dagger b$ 推出，第二步由勾股定理推出，最后一步由 OPT $:= \|Ax^* - b\|_2$ 推出。

19.3 *Example of Algorithms*

这里是一个算法的例子。通常算法应该引用某些定理/引理，相应的定理/引理也应该反向引用。这将更容易验证正确性。

Algorithm 5 Fourier Sparse Recovery Algorithm

procedure FOURIERSPARSERECOVERY(x, n, k, μ, R^*) ▷ Theorem 19.1.1
  Require that $\mu = \frac{1}{\sqrt{k}}\|\widehat{x}_{-k}\|_2$ and $R^* \geq \|\widehat{x}\|_\infty / \mu$
  $H \leftarrow 5, \nu \leftarrow \mu R^*/2,\ y \leftarrow \vec{0}$
  Let $\mathcal{T} = \{\mathcal{T}^{(1)}, \cdots, \mathcal{T}^{(H)}\}$ where each $\mathcal{T}^{(h)}$ is a list of i.i.d. uniform samples in $[p]^d$
  while true do
    $\nu' \leftarrow 2^{1-H}\nu$
    $z \leftarrow$ LINFINITYREDUCE($\{x_t\}_{t\in\mathcal{T}}$)
    if $\nu' \leq \mu$ **then return** $y + z$ ▷ We found the solution
    $y' \leftarrow \vec{0}$
    for $f \in \mathrm{supp}(y + z)$ **do**
      $y'_f \leftarrow \Pi_{0.6\nu}(y_f + z_f)$ ▷ We want $\|\widehat{x} - y'\|_\infty \leq \nu$ and the dependence between y' and $\mathcal{T}$ is under control
    end for
    $y \leftarrow y', \nu \leftarrow \nu/2$
  end while
end procedure

19.4 *Example of Figures*

我们应该确保所有图片都由相同的软件绘制。目前，每个人都可以自由地包含他们自己的图片。赵最终将使用tikz重新绘制图片。

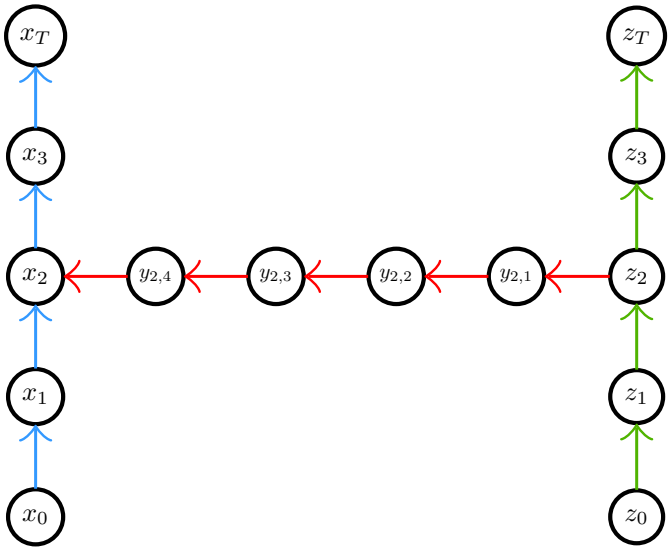


图19.1：追逐序列

19.5 Example of Tables

Reference	Samples	Time	Filter	RIP
[GMS05]	$k\log^{O(d)} n$	$k\log^{O(d)} n$	Yes	No
[CT06]	$k\log^6 n$	$\mathrm{poly}(n)$	No	Yes
[RV08]	$k\log^2 k\log(k\log n)\log n$	$\widetilde{O}(n)$	No	Yes
[HIKP12]	$k\log^d n\log(n/k)$	$k\log^d n\log(n/k)$	Yes	No
[CGV13]	$k\log^3 k\log n$	$\widetilde{O}(n)$	No	Yes
[IK14]	$2^{d\log d}k\log n$	$\widetilde{O}(n)$	Yes	No
[Bou14]	$k\log k\log^2 n$	$\widetilde{O}(n)$	No	Yes
[HR16]	$k\log^2 k\log n$	$\widetilde{O}(n)$	No	Yes
[Kap16]	$2^{d^2}k\log n$	$2^{d^2}k\log^{d+O(1)} n$	Yes	No
[KVZ19]	$k^3\log^2 k\log^2 n$	$k^3\log^2 k\log^2 n$	Yes	Yes
[NSW19]	$k\log k\log n$	$\widetilde{O}(n)$	No	No

表19.1：为了简化，我们忽略O。ℓ_∞/ℓ_2是最强有力的保证，其次是ℓ_2/ℓ_2，然后是ℓ_2/ℓ_1，而恰好k-稀疏是最弱的。我们还注意到，所有[RV08, CGV13, Bou14, HR16]都获得了对限制等距性质的改进分析；算法在[BD08]中被建议并分析（模RIP性质）。[HIKP12]中的工作没有明确指出对d-维情况的外延，但可以从论证中轻易推断出来。[HIKP12, IK14, Kap16, KVZ19]在当每个维度的宇宙大小是2的幂时工作。

19.6 Exercise

本节提供了几个练习的示例。

Exercises

Exercise 19.6-1: 解以下方程以求解 $x \in C$，其中 C 是复数集：

$$5x^2 - 3x = 5 \tag{19.1}$$

Exercise 19.6-2: 解以下方程以求解 $x \in C$，其中 C 是复数集：

$$7x^3 - 2x = 1 \tag{19.2}$$

Bibliography

[ADG+16] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, и Nando de Freitas. 通过梯度下降学习梯度下降。在 *Advances in Neural Information Processing Systems*，2016。

[ADH+19a] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li 和 Ruosong Wang. 超参数化两层神经网络的优化和泛化细粒度分析。在 *International Conference on Machine Learning*，第 322–332 页，2019。

[ADH+19b] Sanjeev Arora，Simon S Du，Wei Hu，Zhiyuan Li，Ruslan S alakhutdinov，和 Ruosong Wang。关于无限宽神经网络的精确计算。*arXiv preprint arXiv:1904.11955*，2019。

[ADL+19] Sanjeev Arora, Simon S Du, Zhiyuan Li, Ruslan Salakhutdinov, Ruosong Wang, 和 Dingli Yu. 在小数据任务中利用无限宽深度网络的强大功能。*arXiv preprint arXiv:1910.01663*，2019。

[AG23] Sanjeev Arora 和 Anirudh Goyal. 语言模型中复杂技能出现的理论。*arXiv preprint arXiv:2307.15936*，2023。

[AGL+17] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma 和 Yi Zha ng. 生成对抗网络（GANs）中的泛化和均衡。*Proc. ICML*，2017。

[AGNZ18] Sanjeev Arora, Rong Ge, Behnam Neyshabur 和 Yi Zhang. 通过压缩方法获得深度网络的更强泛化界限。在 *Proc. ICML 2018*，第 254–263 页，2018。

[ALL19] S Arora, Z Li, 和 K Lyu. 批标准化自动速率调整的理论分析。*ICLR*，2019。

[aro] GANs学习分布吗？一些理论与实证。[AS16] N Alon和J Spencer。*The Probabilistic Method (4th Ed)*。Wiley，2016。[AZ18] Sanjeev Arora和Yi Zhang。编码器-解码器GAN架构的理论限制。*Proc. ICLR*，2018。[BBV16] Afonso S Bandeira，Nicolas Boumal和Vladislav Voroninski。关于同步和社区检测中出现的半定规划的低秩方法。在*Conference on learning theory*中，第361–382页，2016。[BD08] Thomas Blumensath和Mike E Davies。稀疏逼近的迭代阈值法。*Journal of Fourier analysis and Applications*，14(5-6):629–654，2008。[BDK+21] Yasaman Bahri，Ethan Dyer，Jared Kaplan，Jaehoon Lee和Utkarsh Sharma。解释神经缩放定律。*arXiv preprint arXiv:2102.06701*，2021。[Ber24] Sergei Bernstein。关于Chebyshev不等式和Laplace误差公式的修改。*Ann. Sci. Inst. Sav. Ukraine, Sect. Math*，1(4):38–49，1924。[BH89] Pierre Baldi和Kurt Hornik。神经网络与主成分分析：无局部最小值地从例子中学习。*Neural networks*，2(1):53–58，1989。[BKH16] J Ba，J R Kiros和G E Hinton。层归一化。*NeurIPS*，2016。[BM02] Peter L Bartlett和Shahar Mendelson。Rademacher和Gaussian复杂性：风险界限和结构结果。*Journal of Machine Learning Research*，3(Nov):463–482，2002。

[BM03] P. L. Bartlett 和 S. Mendelson. Rademacher 和高斯复杂度：风险界限和结构结果。*Journal of Machine Learning Research*，2003。

[BNS16a] Srinadh Bhojanapalli，Behnam Neyshabur和Nati Srebro。低秩矩阵恢复的局部搜索的全局最优性。在*Advances in Neural Information Pro- cessing Systems*，第3873-3881页，2016年。

[BNS16b] Srinadh Bhojanapalli，Behnam Neyshabur和Nati Srebro。低秩矩阵恢复的局部搜索的全局最优性。在*Advances in Neural Information Pro- cessing Systems (NIPS)*，第3873–3881页，2016年。[Bou14] Jean Bourgain。在限制等距问题中的改进估计。在*Geometric Aspects of Functional Analysis*，第65–70页。Springer，2014年。[BR89] Avrim Blum和Ronald L Rivest。训练3节点神经网络是NP完全的。在*Advances in neural information processing systems*，第494–501页，1989年。[Bre67] L. M. Bregman。寻找凸集公共点的松弛方法及其在凸规划问题中的应用。*USSR computational mathematics and mathematical physics*，1967年。[BT52] Ralph A. Bradley和Milton E. Terry。不完整区组设计的秩分析：I.配对比较法。*Biometrika*，1952年。[BT03] A. Beck和M. Teboulle。镜面下降和非线性投影子梯度方法在凸优化中的应用。*Operations Research Letters*，2003年。[BV04] S. Boyd和L. Vandenberghe。*Convex optimization*。剑桥大学出版社，2004年。[CCS+16] Pratik Chaudhari，Anna Choromanska，Stefano Soatto，Yann LeCun，Carlo Baldassi，Christian Borgs，Jennifer Chayes，Levent Sagun和Riccardo Zecchina。熵-sgd：将梯度下降引入宽山谷。*arXiv preprint arXiv:1611.01838*，2016年。[CGV13] Mahdi Cheraghchi，Venkatesan Guruswami和Ameya Velingker。傅里叶矩阵的限制等距性和随机线性码的列表可解码性。*SIAM Journal on Computing*，42(5)：1888–1914，2013年。[Che52] Herman Chernoff。基于观察和的假设检验的渐近效率度量。*The Annals of Mathematical Statistics*，第493–507页，1952年。[CKL+21] Jeremy Cohen，Simran Kaur，Yuanzhi Li，J Zico Kolter和Ameet Talwalkar。神经网络上的梯度下降通常发生在稳定性的边缘。*ICLR*，2021年。

[CLB+17] P. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, 和 D. A modei. 从人类偏好中进行深度强化学习。*NeurIPS*，2017。[CLG01] R ich Caruana，Steve Lawrence，和 C Lee Giles。神经网络中的过拟合：反向传播、共轭梯度法和早期停止。在 *Advances in Neural Information Processing Systems (NIPS)*，第 402–408 页，2001。[CT06] Emma nuel J Candes 和 Terence Tao。从随机投影中恢复近最优信号：通用编码策略？*IEEE transactions on information theory*，52(12)：5406–5425，2006。[CW82] R D Cook 和 S Weisberg。回归中的残差和影响。1982。[DHS11] J. Duchi，E. Hazan，和 Y. Singer。自适应子梯度方法用于在线学习和随机优化。*Journal of Machine Learning Research*，2011。[DHS+19] Simon S Du，Kangcheng Hou，Russ R Salakhutdinov，Barnabas Poczos，Ruosong Wang，和 Keyulu Xu。图神经切线核：融合图神经网络和图核。在 *Advances in Neural Information Processing Systems*，第 5724–5734 页，2019。[DKB15] Laurent Dinh，David Krueger，和 Yoshua Bengio。NICE：非线性独立成分分析。*Proc. ICLR*，2015。[DLL+18] Simon S Du，Jason D Lee，Haochuan Li，Liwei Wang，和 Xiyu Zhai。梯度下降找到深度神经网络的全球最小值。*arXiv preprint arXiv:1811.03804*，2018。

[DP94] X Deng 和 C Papadimitriou. 关于合作解概念的复杂性。*Math of Operations Research*，1994。

[DPBB17] Laurent Dinh, Razvan Pascanu, Samy Bengio 和 Yoshua Bengio. 锐利最小值可以推广到深度网络。在 *International Conference on Machine Learning*，2017。

[DSDB17] Laurent Dinh, Jascha Sohl-Dickstein 和 Samy Bengio. 使用真实 NVP 进行密度估计。*Proc. ICLR*，2017。

[DZPS18] Simon S Du, Xiyu Zhai, Barnabas Poczos 和 Aarti Singh. 可证明梯度下降优化过参数化神经网络。*arXiv preprint arXiv:1810.02054*，2018。

[DZPS19] Simon S Du, Xiyu Zhai, Barnabas Poczos 和 Aarti Singh. 梯度下降可证明优化过参数化的神经网络。*arXiv preprint arXiv:1810.02054*，2019。

[EHJT04] B. Efron, T. Hastie, I. Johnstone 和 R. Tibshirani. 最小角度回归。*The Annals of statistics*，2004。

[Fri01] Jerome H Friedman. 贪婪函数逼近：梯度提升机。*Annals of statistics*，2001。

[FSSS11] Rina Foygel、Olad Shamir、Nati Srebro和Ruslan R Salakhutdinov。在任意抽样分布下使用加权迹范学习。在 *Advances in Neural Information Processing Systems*，第2133–2141页，2011年。

[GDG+17] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter No-ordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, 和 Kaiming He. 准确、大批量的SGD：1小时内训练ImageNet。*arXiv preprint arXiv:1706.02677*，2017。

[GHJY15a] 龙鸽，黄芙蓉，金驰，杨源。逃离鞍点——张量分解的在线随机梯度。在 *Conference on Learning Theory*，第 797–842 页，2015。

[GHJY15b] 龙鸽，黄芙蓉，金驰，杨源。逃离鞍点——张量分解的在线随机梯度。在 *Conf. Learning Theory (COLT)*，2015。

[GJZ17a] Rong Ge，Chi Jin，和 Yi Zheng。非凸低秩问题中无虚假局部极小值：统一几何分析。在 *Proceedings of the 34th International Conference on Machine Learning-Volume 70*，第 1233–1242 页。JMLR.org，2017。

[GJZ17b] 龙鸽，迟金，郑毅. 非凸低秩问题中无虚假局部极小值：统一几何分析。*arXiv preprint arXiv:1704.00708*，2017。

[GLM16a] Rong Ge，Jason D Lee，和 Tengyu Ma. 矩阵补全没有虚假局部最小值。在 *Advances in Neural Information Processing Systems*，第 2973–2981 页，2016 年。[GLM16b] Rong Ge，Jason D Lee，和 Tengyu Ma. 矩阵补全没有虚假局部最小值。在 *Advances in Neural Information Processing Systems (NIPS)*，2016 年。

[GLM18] 龙鸽，李杰森，马腾宇。通过景观设计学习单隐藏层神经网络。在 *ICLR*。arXiv 预印本 arXiv:1711.00501，2018。

[GLSS18a] Suriya Gunasekar，Jason Lee，Daniel Soudry和Nathan Srebro。从优化几何的角度描述隐含偏见。*arXiv preprint arXiv:1802.08246*，2018。

[GLSS18b] Suriya Gunasekar，Jason D Lee，Daniel Soudry和Nati Srebro。线性卷积网络的梯度下降的隐式偏差。在S. Bengio，H. Wallach，H. Larochelle，K. Grauman，N. Cesa-Bianchi和R. Garnett编辑的 *Advances in Neural Information Pro- cessing Systems*，第31卷。Curran Associates，Inc.，2018。[GMS05] Anna C Gilbert，S Muthukrishnan和Martin Strauss。近最优稀疏傅里叶表示的时间界限改进。在 *Optics & Photonics 2005*，第59141A–59141A页。国际光学和光子学学会，2005。[GPAM+14] I Goodfellow，J Pouget-Abadie，M Mirza，B Xu，D Warde-Farley，S Ozair，A Courville和Y Bengio。生成对抗网络。*NeurIPS*，2014。

[GWB+17] Suriya Gunasekar，Blake E Woodworth，Srinadh Bho-janapalli，Behnam Neyshabur和Nati Srebro。矩阵分解中的隐式正则化。在 *Advances in Neural Information Processing Systems*，第6151–6159页，2017。

[HBD+20] A Holtzman, J Buys, L Du, M Forbes, 和 Y Choi. 神经文本退化的奇特案例。*ICLR*，2020。

[HBM+22] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl,

Aidan Clark等人。训练计算最优的大型语言模型。*arXiv preprint arXiv:2203.15556*，2022。[HHS17] Elad Hoffer，Itay Hubara和Daniel Soudry。训练更长，泛化更好：关闭神经网络大批量训练的泛化差距。在*Advances in Neural Information Processing Systems*，2017。[HIKP12] Haitham Hassanieh，Piotr Indyk，Dina Katabi和Eric Price。几乎最优的稀疏傅里叶变换。在*Proceedings of the forty-fourth annual ACM symposium on Theory of computing*，第563–578页。ACM，2012。[HLLL19] Wenqing Hu，Chris Junchi Li，Lei Li和Jian-Guo Liu。关于非凸随机梯度下降的扩散近似。*Annals of Mathematical Sciences and Applications*，4(1)，2019。[HMR18] Moritz Hardt，Tengyu Ma和Benjamin Recht。梯度下降学习线性动力系统。在*JLMR*。arXiv预印本arXiv:1609.05191，2018。[Hoe63] Wassily Hoeffding。有界随机变量的和的概率不等式。*Journal of the American Statistical Association*，58(301)：13–30，1963。[HR16] Ishay Haviv和Oded Regev。子采样傅里叶矩阵的受限等距性质。在*SODA*，第288–297页。arXiv预印本arXiv:1507.01768，2016。[HS97] Sepp Hochreiter和Jürgen Schmidhuber。平坦最小值。*Neural Computation*，1997。[IK14] Piotr Indyk和Michael Kapralov。任何常数维度的样本最优傅里叶采样。在*Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*，第514–523页。IEEE，2014。[IPE+22] Andrew Ilyas，Sung Min Park，Logan Engstrom，Guillaume Leclerc和Aleksander Madry。Datamodels：从训练数据预测预测。*arXiv preprint arXiv:2202.00622*，2022。[IS15a] S Ioffe和C Szegedy。批量归一化：通过减少内部协变量偏移来加速深度网络训练。*ICML*，2015。[IS15b] Sergey Ioffe和Christian Szegedy。批量归一化：通过减少

内部协变量偏移。在 *International Conference on Machine Learning (ICML)*，第 448–456 页，2015 年。

[JGH18] Arthur Jacot, Franck Gabriel 和 Clément Hongler. 神经切线核：神经网络的收敛性和泛化性。在 *Advances in neural information processing systems*，第 8571–8580 页，2018。

[JGN+17] 池金，荣格，普拉内什·内特拉帕利，沙姆·M·卡卡德，迈克尔·I·乔丹。如何高效地逃离鞍点。*arXiv preprint arXiv:1703.00887*，2017。

[JKA+18] S Jastrz ębski, Z Kenton, D Arpit, N Ballas, A Fischer, Y B engio, 和 A Storkey. 影响SGD中最小值的三种因素。*ICANN*，2018。

[JNG+19] 池金，普拉内特·内特拉帕利，荣格，沙姆·M·卡卡德，以及迈克尔·I·乔丹。关于机器学习的非凸优化：梯度、随机性和鞍点。*arXiv preprint arXiv:1902.04811*，2019。

[Kap16] 米哈伊尔·卡普拉洛夫。任何常维度的稀疏傅里叶变换在亚线性时间内具有近最优的样本复杂度。在 *Symposium on Theory of Computing Conference, STOC'16, Cambridge, MA, USA, June 19-21, 2016*，2016。

[Kaw16] 川口健二. 无劣局部最小值的深度学习。在 *Adv in Neural Information Proc. Systems (NIPS)*，2016。

[KD19] Diederik P. Kingma 和 Prafulla Dhariwal. GLOW: 具有可逆 1×1 卷积的生成流。*Proc. Neurips*，2019。

[KGC17] 简库卡维卡，弗拉基米尔·戈尔科夫，丹尼尔·克雷默斯。深度学习的正则化：一个分类法。*arXiv preprint arXiv:1710.10686*，2017。

[KKSK11] Sham M Kakade, Varun Kanade, Ohad Shamir 和 Adam Kala i. 使用同形回归高效学习广义线性模型和单指标模型。在 *Advances in Neural Information Processing Systems*，第 927–935 页，2011 年。

[KL17] P W Koh 和 P Liang. 通过影响函数理解黑盒预测。在 *Proc. ICML*，2017。

[KMN+16] Nitish Shirish Keskar，Dheevatsa Mudigere，Jorge No-ceda l，Mikhail Smelyanskiy，和Ping Tak Peter Tang。关于深度学习的大批量训练：泛化差距和尖锐极小值。在*International Conference on Learning Representations*，2016。

[KS09] Adam Tauman Kalai 和 Ravi Sastry. 等距算法：高维等距回归。在 *COLT*。Citeseer，2009。

[KST09] Sham M Kakade, Karthik Sridharan 和 Ambuj Tewari. 关于线性预测的复杂性：风险界限、边界界限和正则化。在 *Advances in neural information processing systems*，2009。

[KVZ19] 米哈伊尔·卡普拉洛夫，阿梅亚·维林克尔，和阿米尔·赞迪耶。独立于维度的稀疏傅里叶变换。在*Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*，第 2709–2728 页。SIAM，2019。

[LA19] 李志和S·阿罗拉。深度学习的指数学习率调度。*ICLR*，2019。

[Lan02] 约翰·兰福德. *Quantitatively tight sample complexity bounds*. 卡内基梅隆大学博士论文，2002年。

[LBZ+22] 李Z，博霍贾帕利S B，扎希尔M，雷迪S，和库马尔S。使用尺度不变架构的神经网络的鲁棒训练。*arxiv*，2022。

[LDM12] Hector Levesque，Ernest Davis 和 Leora Morgenstern。Wino grad 模式挑战。在 *Thirteenth international conference on the principles of knowledge representation and reasoning*，2012。

[LLA20] 李志远，刘凯峰，和Sanjeev Arora。将现代深度学习与传统优化分析相协调：内在学习率。在Hugo Larochelle，Marc’Aure lio Ranzato，Raia Hadsell，Maria-Florina Balcan和Hsuan-Tien Lin编辑的*Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*，2020。

[LMA21] 李志远，马莉迪，和阿拉拉·桑杰夫。关于将随机微分方程（SDEs）建模SGD的有效性。*Advances in Neural Information Processing Systems*，34，2021。

[LMAPH19] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda- Pineda, 和 Radu Horaud. 深度回归的全面分析。
IEEE transactions on pattern analysis and machine intelligence，2019。[LMZ18] Yuanzhi Li，Tengyu Ma，和 Hongyang Zhang。在过参数化矩阵感知和具有二次激活的神经网络中的算法正则化。在
Conference On Learning Theory，第 2–47 页，2018。[LSJR16] Jason D Lee，Max Simchowitz，Michael I Jordan，和 Benjamin Recht。梯度下降收敛到最小值。*arXiv preprint arXiv:1602.04915*，2016。[LTW19] Qianxiao Li，Cheng Tai，和 E Weinan。随机修改方程和随机梯度算法的动力学：数学基础。*J. Mach. Learn. Res.*，20:40–1，2019。[LWLA22] Zhiyuan Li，Tianhao Wang，Jason D Lee，和 Sanjeev Arora。重参数化模型上梯度下降的隐含偏差：关于与镜像下降等价。
Advances in Neural Information Processing Systems，35:34626–34640，2022。[MA19] Poorya Mianjy 和 Raman Arora。关于 dropout 和核范数正则化。在 *International Conference on Machine Learning*，2019。[MAV18] Poorya Mianjy，Raman Arora，和 Rene Vidal。关于 dropout 的隐含偏差。在 *International Conference on Machine Learning*，第 3537–3545 页，2018。[McA99] David A McAllester。一些 pac-bayesian 定理。
Machine Learning，37(3):355–363，1999。[MHB17] Stephan Mandt，Matthew D Hoffman，和 David M Blei。随机梯度下降作为近似贝叶斯推理。*The Journal of Machine Learning Research*，18(1):4873–4907，2017。[MMS+18] A Madry，A Makelov，L Schmidt，D Tsipras，和 A Vladu。朝着对对抗攻击具有抵抗力的深度学习模型。*ICLR*，2018。[MRB+23] Niklas Muennighoff，Alexander M. Rush，Boaz Barak，Teven Le Scao，Aleksandra Piktus，Nouamane Tazi，Sampo Pyysalo，Thomas Wolf，和 Colin Raffel。扩展数据约束语言模型，2023。

[MRT18] Mehryar Mohri, Afshin Rostamizadeh 和 Ameet Talwalkar. *Foundations of machine learning*. MIT 压力，2018。[NBE17] Agar wal N，Bullins B 和 Hazan E。线性时间机器学习的二阶随机优化。2017。

[NBMS18] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, 和 Nathan Srebro. 关于神经网络谱归一化边界的不动点贝叶斯方法。*ICLR*，2018。[Nes98] Yurii Nesterov. *Introductory Lectures on Convex Program- ming Volume I: Basic Course*。Springer，1998。[Nes00] Yurii Nesterov. 平方泛函系统和优化问题。在 *High performance optimization*，第 405–440 页。Springer，2000。[Ney17] Behnam Neyshabur. 深度学习中隐式正则化。*arXiv preprint arXiv:1709.01953*，2017。[NH92] Steven J Nowlan 和 G eoffrey E Hinton。通过软权重共享简化神经网络。*Neural computa- tion*，4(4):473–493，1992。[NK19] V Nagarajan 和 Zico Kolter。均匀收敛可能无法解释深度学习中的泛化。*NeurIPS*，2019。[NP06] Yurii Nesterov 和 Boris T Polyak。牛顿方法的立方正则化和其全局性能。*Mathematical Programming*，108(1):177–205，2006。[NS23] Mark Br averman Sanjeev Aror Nikunj Saunshi，Arushi Gupta。通过谐波分析理解影响函数和数据模型。*ICLR*，2023。[NSS15] Behnam Neyshabur, Ruslan R Salakhutdinov，和 Nati Srebro。Path-sgd：深度神经网络中的路径归一化优化。在 *Advances in Neural Information Processing Systems*，第 2422–2430 页，2015。[NSW19] Vasileios Nak os，Zhao Song，和 Zhengyu Wang。任何维度的（几乎）样本最优稀疏傅里叶变换；RIPless 和 Filterless。在 *FOCS*。`https://arxiv.org/pdf/1909.11123.pdf`，2019。

[NTS15a] Behnam Neyshabur, Ryota Tomioka, 和 Nathan Sre-bro. 在寻找真实的归纳偏置：关于隐式正则化在深度学习中的作用。在 *International Conference on Learning Representations*，2015。

[NTS15b] Behnam Neyshabur、Ryota Tomioka 和 Nathan Srebro. 基于范数的神经网络容量控制。在 *Conference on Learning Theory*，第 1376–1401 页，2015 年。

[NY83] A. Nemirovskii 和 D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.

[Pea94] Barak Pearlmutter. 通过Hessian的快速精确乘法。*Neural Computation*，1994。

[PKCS17] , 斯塔 , 尼斯, . Burer-Monteiro . *AISTATS* . arXiv arXiv:1609.03240, 2017.

[RDS04] Cynthia Rudin，Ingrid Daubechies，和Robert E Schapire。Adaboost的动力学：边缘的循环行为和收敛。*Journal of Machine Learning Research*，5(12月)：1557–1595，2004。

[RSM+23] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, 和 Chelsea Finn. 直接偏好优化：你的语言模型其实是一个奖励模型。*arxiv*，2023。

[RV08] 马克·鲁德森和罗曼·弗谢宁。关于从傅里叶和高斯测量中进行稀疏重建。*Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*，61(8)：1025–1045，2008。

[SF12] 罗伯特·E沙皮雷和约阿夫·弗里德曼。*Boosting: Foundations and algorithms*。麻省理工学院出版社，2012年。

[Sha] 洛伊德·沙普利。*"Notes on the n-Person Game – II: The Value of an n-Person Game"*。兰德公司。

[SHK+14] Nitish Srivastava，Geoffrey E Hinton，Alex Krizhevsky，Ilya Sutskever和Ruslan Salakhutdinov。Dropout：一种防止神经网络过拟合的简单方法。*Journal of Machine Learning Research (JMLR)*，15(1)，2014。

[SHS17] Daniel Soudry, Elad Hoffer, 和 Nathan Srebro. 分离数据上梯度下降的隐式偏差。*arXiv preprint arXiv:1710.10345*，2017。[Smi18] Le Smith, Kindermans. 不要衰减学习率，增加批量大小。在 *ICLR*，2018。[SMK23] Ryan Schaeffer, Brando Miranda, 和 Sanmi Koyejo. 大型语言模型的涌现能力是幻想吗？*ArXiv e-prints*，2023年4月。[SQW16a] Ju Sun, Qing Qu, 和 John Wright. 球面上的完整字典恢复：概述和几何图像。*IEEE Transactions on Information Theory*，63(2):853–884，2016。[SQW16b] Ju Sun, Qing Qu, 和 John Wright. 相位恢复的几何分析。在 *IEEE International Symposium on Information Theory (ISIT)*，第2379–2383页，2016。[SQW18] Ju Sun, Qing Qu, 和 John Wright. 相位恢复的几何分析。*Foundations of Computational Mathematics*，18(5):1131–1198，2018。[SS10] Nathan Srebro 和 Ruslan R Salakhutdinov. 非均匀世界中的协同过滤：加权迹范学习。在 *Advances in Neural Information Processing Systems*，第2056–2064页，2010。[SSJ20] Bin Shi, Weijie J Su, 和 Michael I Jordan. 关于学习率和薛定谔算子。*arXiv preprint arXiv:2004.06977*，2020。[SSS10] Shai Shalev-Shwartz 和 Yoram Singer. 弱可学习和线性可分等价：新的松弛和有效的提升算法。*Machine learning*，2010。[SY19] Zhao Song 和 Xin Yang. 通过矩阵切诺夫界对过参数化的二次充分性。*arXiv preprint arXiv:1906.03593*，2019。[SYL+19] H Salman, G Yang, J Li, P Zhang, H Zhang, I Razenshteyn, 和 S Bubeck. 通过对抗性训练的平滑分类器实现可证明的鲁棒深度学习。*NeurIPS*，2019。[SZA20] F Schaefer, H Zheng, 和 A Anandkumar. GAN中的隐式竞争正则化。*ICML*，2020。

[SZS+14] C Szegedy, W Zaremba, I Sutskever, J Bruna, D Erhan, I Goodfellow, 和 R Fergus. 神经网络的有趣性质。在 *ICLR*，2014。[Tel13] Matus Telgarsky. 边界，收缩和提升。*arXiv preprint arXiv:1303.4172*，2013。[Tro15] Joel A Tropp. 矩阵集中不等式导论。*Foundations and Trends® in Machine Learning*，8(1-2):1–230，2015。[Wer88] P. J. Werbos. 反向传播：过去和未来。在 *IEEE InternationalConference on Neural Networks*，第 343–353 页，1988。[WGL+20] Blake Woodworth，Suriya Gunasekar，Jason D Lee，Edward Moroshko，Pedro Savarese，Itay Golan，Daniel Soudry，和 Nathan Srebro。在过参数化模型中的核和丰富机制。在 *Conference on Learning Theory*，第 3635–3673 页，2020。[WH17] Y Wu 和 K He。组归一化。*ECCV*，2017。[Win71] Terry Winograd。在理解自然语言的计算机程序中将程序作为数据表示。技术报告，麻省理工学院剑桥项目 MAC，1971。[WRS+17] Ashia C Wilson，Rebecca Roelofs，Mitchell Stern，Nati Srebro，和 Benjamin Recht。机器学习中自适应梯度方法的边际价值。在 *Advances in Neural Information Processing Systems*，2017。[WTB+22] Jason Wei，Yi Tay，Rishi Bommasani，Colin Raffel，Barret Zoph，Sebastian Borgeaud，Dani Yogatama，Maarten Bosma，Denny Zhou，Donald Metzler，等。大型语言模型的涌现能力。*arXiv preprint arXiv:2206.07682*，2022。[ZBH+16a] Chiyuan Zhang，Samy Bengio，Moritz Hardt，Benjamin Recht，和 Oriol Vinyals。理解深度学习需要重新思考泛化。*arXiv preprint arXiv:1611.03530*，2016。[ZBH+16b] Chiyuan Zhang，Samy Bengio，Moritz Hardt，Benjamin Recht，和 Oriol Vinyals。理解深度学习需要重新思考泛化。*arXiv preprint arXiv:1611.03530*，2016。

[ZGSA22] 张Y，古普塔A，萨恩希N，阿罗拉S. 关于使用生成对抗网络进行泛化预测。*ICLR*，2022。

[ZM] Kolter Z 和 Madry M. 对抗鲁棒性——理论与实践。

[ZY+05] 钟张，余斌，等人。提前停止的Boosting：收敛性和一致性。*The Annals of Statistics*，2005。