几个数学建模问题与机器学习理论进展

王铭泽

北京大学　数学科学学院

2023 年 7 月 4 日

个人简介

王铭泽

- 北京大学 数学科学学院
  - 二年级直博生 (*2021.9 - 现在*)
  - 专业：计算数学
  - 导师：鄂维南老师
  - 研究方向：机器学习/深度学习理论与算法、凸/非凸优化理论与算法等

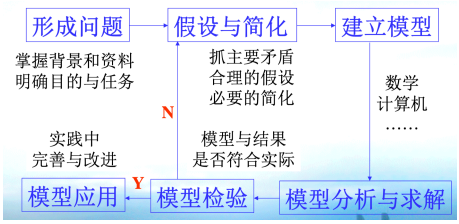
- 浙江大学 数学科学学院
  - 本科学位 (*2017.9 - 2021.6*)
  - 专业：数学与应用数学
  - 排名：1/111
  - 研究兴趣：数学建模，连续/组合优化等

Contents

第一部分. 几个数学建模问题

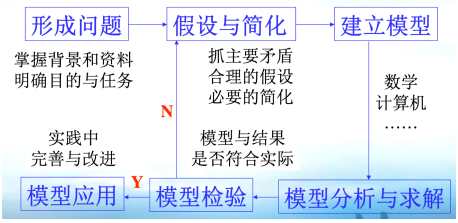
数学建模简介

- 数学建模概述：针对实际应用问题，建立合理的数学模型，并使用数学/计算机求解。
- 主要步骤



数学建模简介

- 数学建模概述：针对实际应用问题，建立合理的数学模型，并使用数学/计算机求解。
- 主要步骤



- 主要数学方法
  - 数学基础：微积分，线性代数，概率论与数理统计
  - 微分方程：常微分方程，偏微分方程
  - 运筹优化：连续优化，组合优化，动态最优化，博弈论
  - 随机方法：随机过程、排队论

高压油管的压力控制

高压油管的压力控制

“高压油管的压力控制”——2019 年高教社杯全国大学生数学建模竞赛 A 题

王铭泽，林浩通，何怡君（前 2.5%）

- 问题背景：燃油发动机的工作基础是燃油进入和喷出高压油管，分别由高压油泵与喷油嘴来完成，两者往往周期性间歇工作，这会导致油管压力发生变化，从而影响喷出的燃油量，影响发动机的工作效率。因此，为了使高压油管的气压在一定时间内稳定在特定水平，需要设计精细的进油与出油控制方案。

高压油管的压力控制

“高压油管的压力控制”——2019 年高教社杯全国大学生数学建模竞赛 A 题

王铭泽，林浩通，何怡君（前 2.5%）

- 问题背景：燃油发动机的工作基础是燃油进入和喷出高压油管，分别由高压油泵与喷油嘴来完成，两者往往周期性间歇工作，这会导致油管压力发生变化，从而影响喷出的燃油量，影响发动机的工作效率。因此，为了使高压油管的气压在一定时间内稳定在特定水平，需要设计精细的进油与出油控制方案。

- 问题概述：
  1. 给定某型号高压油管的尺寸信息、高压油泵每开启一次之后的关闭时间、喷油嘴的工作周期以及一个周期内的喷油速率，求出能将高压油管内的压力稳定在 100MPa 的供油时间。
  2. 在问题一的基础上，对喷油与进油的机制做出如下改变：
     - 喷油机制：给定周期内的喷油速率 $\rightarrow$ 给定喷油嘴的工作原理及相关几何参数，针阀上升则喷油，针阀下降至最低端则喷油嘴关闭；
     - 进油机制：稳压进油 $\rightarrow$ 凸轮转动进油。

     求解凸轮运动的角速度，使得高压油管内的压力稳定在 100MPa。
  3. 在问题二的基础上，增加一个同规格喷油嘴后和一个单向减压油阀，设计高压油泵和减压阀的控制方案。

高压油管的压力控制

通用数学模型

三个问题的目标都是在一定条件下，制定最优策略，使高压油管内的压力 $p(t)$ 尽可能稳定在 100MPa 附近。

高压油管的压力控制

通用数学模型

三个问题的目标都是在一定条件下，制定最优策略，使高压油管内的压力 $p(t)$ 尽可能稳定在 100MPa 附近。

- **Step I. 寻找问题机理**：质量守恒。油泵流入质量与喷油嘴流出质量的差异导致了油管内质量变化进而引起密度变化，从引起压力变化。由质量守恒定律，知 t 时刻高压油管中的燃油密度 $\rho_{管}(t)$ 满足积分方程：

$$\rho_{管}(t)\,V-\rho_{管}(0)\,V=\int_0^t \rho_{泵}(t)\,Q_{进管}(t)\mathrm{d}t-\int_0^t \rho_{管}(t)\,Q_{出管}(t)\mathrm{d}t.$$

高压油管的压力控制

通用数学模型

三个问题的目标都是在一定条件下，制定最优策略，使高压油管内的压力 $p(t)$ 尽可能稳定在 100MPa 附近。

- **Step I. 寻找问题机理**：质量守恒。油泵流入质量与喷油嘴流出质量的差异导致了油管内质量变化进而引起密度变化，从引起压力变化。由质量守恒定律，知 t 时刻高压油管中的燃油密度 $\rho_{管}(t)$ 满足积分方程：

$$\rho_{管}(t)\,V-\rho_{管}(0)\,V=\int_0^t \rho_{泵}(t)\,Q_{进管}(t)\mathrm{d}t-\int_0^t \rho_{管}(t)\,Q_{出管}(t)\mathrm{d}t.$$

- **Step II. 燃油密度 $\rho_{管}(t)$ 与压力 $p(t)$ 关系**。首先根据数据，拟合弹性模量 $E(p)$；然后求解可分离变量方程 $\mathrm{d}p=\frac{E(p)}{\rho}\mathrm{d}\rho$ 得到燃油压力 p 与密度 $\rho_{管}$ 的严格单调的恒等关系式 $\rho_{管}=\phi(p)$。

高压油管的压力控制

通用数学模型

三个问题的目标都是在一定条件下，制定最优策略，使高压油管内的压力 $p(t)$ 尽可能稳定在 100MPa 附近。

- **Step I. 寻找问题机理**：质量守恒。油泵流入质量与喷油嘴流出质量的差异导致了油管内质量变化进而引起密度变化，从引起压力变化。由质量守恒定律，知 t 时刻高压油管中的燃油密度 $\rho_{管}(t)$ 满足积分方程：

$$\rho_{管}(t)\,V-\rho_{管}(0)\,V=\int_0^t \rho_{泵}(t)\,Q_{进管}(t)\mathrm{d}t-\int_0^t \rho_{管}(t)\,Q_{出管}(t)\mathrm{d}t.$$

- **Step II. 燃油密度 $\rho_{管}(t)$ 与压力 $p(t)$ 关系**。首先根据数据，拟合弹性模量 $E(p)$；然后求解可分离变量方程 $\mathrm{d}p=\frac{E(p)}{\rho}\mathrm{d}\rho$ 得到燃油压力 p 与密度 $\rho_{管}$ 的严格单调的恒等关系式 $\rho_{管}=\phi(p)$。
- **Step III. 量化稳定性指标**：平均偏移量 $\alpha(p):=\frac{1}{T}\left\|p(t)-100\right\|_{L^1[0,T]}$；最大偏移量 $\beta(p):=\left\|p(t)-100\right\|_{L^\infty[0,T]}$.

高压油管的压力控制

通用数学模型

三个问题的目标都是在一定条件下，制定最优策略，使高压油管内的压力 $p(t)$ 尽可能稳定在 100MPa 附近。

- **Step I. 寻找问题机理**：质量守恒。油泵流入质量与喷油嘴流出质量的差异导致了油管内质量变化进而引起密度变化，从引起压力变化。由质量守恒定律，知 t 时刻高压油管中的燃油密度 $\rho_{管}(t)$ 满足积分方程：

$$\rho_{管}(t)\,V-\rho_{管}(0)\,V=\int_0^t \rho_{泵}(t)\,Q_{进管}(t)\mathrm{d}t-\int_0^t \rho_{管}(t)\,Q_{出管}(t)\mathrm{d}t.$$

- **Step II. 燃油密度 $\rho_{管}(t)$ 与压力 $p(t)$ 关系**。首先根据数据，拟合弹性模量 $E(p)$；然后求解可分离变量方程 $\mathrm{d}p=\frac{E(p)}{\rho}\mathrm{d}\rho$ 得到燃油压力 p 与密度 $\rho_{管}$ 的严格单调的恒等关系式 $\rho_{管}=\phi(p)$。
- **Step III. 量化稳定性指标**：平均偏移量 $\alpha(p):=\frac{1}{T}\left\|p(t)-100\right\|_{L^1[0,T]}$；最大偏移量 $\beta(p):=\left\|p(t)-100\right\|_{L^\infty[0,T]}$.
- **Step IV. 建立通用最优控制模型**：

$$\begin{aligned}&\min_{\pi(t)}:\ \alpha(p)+\lambda\beta(p)\\&\quad\text{s.t. }\rho_{管}=\phi(p),\\&\qquad\rho_{管}(t)\text{ 满足在控制策略 }\pi(t)\text{ 下的质量守恒方程 (4).}\end{aligned}$$

高压油管的压力控制

不同情形下的策略求解

针对三个问题，只需研究$\rho_{管}(t)$ 满足在控制策略下的质量守恒方程，然后数值求解最优控制问题即可。

$$\rho_{管}(t)\,V-\rho_{管}(0)\,V=\int_0^t \rho_{泵}(t)\,Q_{进管}(t)\mathrm{d}t-\int_0^t \rho_{管}(t)\,Q_{出管}(t)\mathrm{d}t.$$

1. 在问题一中，
   - 设 $t_1\in\mathbb{R}_+$ 为单向阀每次开启的时间，$t_0=10ms$ 为单向阀每次关闭的时间，形成周期性控制系统。
   - $\rho_{泵}(t)\equiv\rho_{泵}$；$Q_{进管}$ 和 $Q_{出管}$ 分别满足：

$$Q_{进管}(t)=\begin{cases} CA\sqrt{\dfrac{2(p_{泵}-p_{管}(t))}{\rho_{泵}(t)}} & 0\le\dfrac{t}{T_1}-\left\lfloor\dfrac{t}{T_1}\right\rfloor<\dfrac{t_1}{T_1}\\ 0 & 其他\end{cases}$$

$$Q_{出管}(t)=\begin{cases} k\left(t-T_2\left\lfloor\dfrac{t}{T_2}\right\rfloor\right) & 0\le\dfrac{t}{T_2}-\left\lfloor\dfrac{t}{T_2}\right\rfloor<\dfrac{0.2}{T_2}\\ 20 & \dfrac{0.2}{T_2}\le\dfrac{t}{T_2}-\left\lfloor\dfrac{t}{T_2}\right\rfloor<\dfrac{2.2}{T_2}\\ 20-k\left(t-2.2-T_2\left\lfloor\dfrac{t}{T_2}\right\rfloor\right) & \dfrac{2.2}{T_2}\le\dfrac{t}{T_2}-\left\lfloor\dfrac{t}{T_2}\right\rfloor<\dfrac{2.4}{T_2}\\ 0 & 其他\end{cases}$$

   - 最后转化为求解关于 $t_1\in\mathbb{R}_+$ 的 1 维连续优化问题。

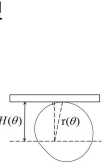
高压油管的压力控制

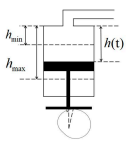
不同情形下的策略求解

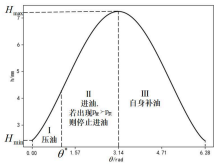
$$\rho_{管}(t)\,V-\rho_{管}(0)\,V=\int_0^t \rho_{泵}(t)\,Q_{进管}(t)\mathrm{d}t-\int_0^t \rho_{管}(t)\,Q_{出管}(t)\mathrm{d}t.$$

❷ 在问题二中，设 $\omega\in\mathbb{R}_+$ 为凸轮转动角速度，形成周期性控制系统。

- 进油模型







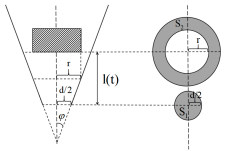
$$Q_{进管}(t)=\begin{cases} CA\sqrt{\dfrac{2(p_{泵}(t)-p_{管}(t))}{\rho_{泵}(t)}} & 0\le\dfrac{t}{T_1}-\left\lfloor\dfrac{t}{T_1}\right\rfloor<\dfrac{1}{2},\text{且}p_{管}(t)\le p_{泵}(t)\\ 0 & \text{其他}\end{cases}\tag{13}$$

为方便后续微分方程形式的统一及求解，我们将式 (12) 积分方程改写成等价周期性微分方程及初值

$$\begin{cases}\dfrac{\mathrm{d}\rho_{泵}}{\mathrm{d}t}=-\dfrac{\rho_{泵}(t)Q_{进管}(t)+\rho_{泵}(t)h'(t)}{h(t)}\\ \rho_{泵}\left(T_1\left\lfloor\dfrac{t}{T_1}\right\rfloor\right)=\rho_{泵}(0)\end{cases}\tag{14}$$

其中，$T_1\left\lfloor\dfrac{t}{T_1}\right\rfloor\le t<T_1\left(\left\lfloor\dfrac{t}{T_1}\right\rfloor+1\right)$

- 出油模型



$$Q_{出管}(t)=\begin{cases} CA(t)\sqrt{\dfrac{2(p_{管}(t)-p_0)}{\rho_{管}(t)}} & p_{管}(t)\ge p_0\\ 0 & \text{其他}\end{cases}\tag{15}$$

其中，$A(t)=\min\{S_1,S_2(t)\}$

$$S_1=\frac{\pi d^2}{4}\tag{16}$$

$S_2(t)=\pi\left(l(t)tan\phi+r_0\right)^2-\pi r_0^2$，$\phi$为密封座半角

- 最后转化为求解关于 $\omega\in\mathbb{R}_+$ 的 1 维连续优化问题。

高压油管的压力控制

不同情形下的策略求解

3. 问题三求解与问题二类似。
4. 在问题解决后，
   - 进行模型的稳定性分析：定义解的 Lyapunouv 周期稳定性。
   - 进行模型的灵敏度分析：探究不同参数扰动对结果影响的大小。

高压油管的压力控制

不同情形下的策略求解

3. 问题三求解与问题二类似。
4. 在问题解决后，
   - 进行模型的稳定性分析：定义解的 Lyapunouv 周期稳定性。
   - 进行模型的灵敏度分析：探究不同参数扰动对结果影响的大小。

小结：典型的基于机理的建模问题

数学工具：动态最优化（最优控制），连续优化，常微分方程，周期性常微分方程，平面几何等

The Longest Lasting Sandcastles

The Longest Lasting Sandcastles

“The Longest Lasting Sandcastles”——2020 年美国大学生数学建模竞赛 B 题

“Build a Sandcastle to ‘Live in’ ”——王铭泽，林浩通，何怡君（前 8%）

- **Background.** People enjoy building spectacular sandcastles on the beach for fun. It arouses our interest that even if experiencing roughly the same erosion from waves, tides and rains, some sandcastles lasts longer than others.

The Longest Lasting Sandcastles

“The Longest Lasting Sandcastles”——2020 年美国大学生数学建模竞赛 B 题
“Build a Sandcastle to ‘Live in’ ”——王铭泽，林浩通，何怡君（前 8%）

- **Background.** People enjoy building spectacular sandcastles on the beach for fun. It arouses our interest that even if experiencing roughly the same erosion from waves, tides and rains, some sandcastles lasts longer than others.
- **Problems.**
  1. Construct a mathematical model to identify the best 3-dimensional geometric shape to use as a sandcastle foundation that will last the longest period of time on a seashore that experiences waves and tides under the following conditions: (i) at the same distance from the water on the same beach, (ii) using the same amount of sand and water-to-sand proportion.
  2. Using your model, determine an optimal sand-to-water mixture proportion for the castle foundation.
  3. Adjust your model as needed to determine how the best 3-dimensional sandcastle foundation you identified in the first problem is affected by rain, and whether it remains the best 3-dimensional geometric shape to be used as a castle foundation when it is raining.

The Longest Lasting Sandcastles

Problem 1. Static Model for Scouring from Waves and Tides

- **Step I. Simplification.** We neglect the deformation of the foundation. Therefore, the shape is the best if and only if **it minimizes the erosion on the foundation from waves and tides in one cycle**. Let the shape be $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, and $u \in \mathcal{C}(\bar{\Omega}) \cap \mathcal{C}^k(\Omega - \{\mathbf{0}\})$.

The Longest Lasting Sandcastles

Problem 1. Static Model for Scouring from Waves and Tides

- **Step I. Simplification.** We neglect the deformation of the foundation. Therefore, the shape is the best if and only if **it minimizes the erosion on the foundation from waves and tides in one cycle**. Let the shape be $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, and $u \in \mathcal{C}(\bar{\Omega}) \cap \mathcal{C}^k(\Omega - \{\mathbf{0}\})$.
- **Step II. Quantify Erosion.**
  - **Part I. Horizontal Scouring by Waves.** The steeper, the more unstable; the scouring of waves decreases slower with the increase of height. So we consider $I_1(u) := \int_\Omega e^{-u(\boldsymbol{x})} \|\nabla u(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}$ (weighted TV-norm).
  - **Part II. Complete Soaking in Tides.** The larger surface tends to accelerate the exchange between water and sand, causing greater destruction to the foundation. So we consider $I_2(u) := \int_\Omega \sqrt{1 + \|\nabla u(\boldsymbol{x})\|^2} \mathrm{d}\boldsymbol{x}$.

The Longest Lasting Sandcastles

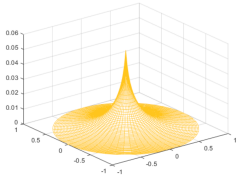
Problem 1. Static Model for Scouring from Waves and Tides

- **Step I. Simplification.** We neglect the deformation of the foundation. Therefore, the shape is the best if and only if **it minimizes the erosion on the foundation from waves and tides in one cycle**. Let the shape be $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, and $u \in \mathcal{C}(\bar{\Omega}) \cap \mathcal{C}^k(\Omega - \{\mathbf{0}\})$.
- **Step II. Quantify Erosion.**
  - **Part I. Horizontal Scouring by Waves.** The steeper, the more unstable; the scouring of waves decreases slower with the increase of height. So we consider $I_1(u) := \int_\Omega e^{-u(\boldsymbol{x})} \|\nabla u(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}$ (weighted TV-norm).
  - **Part II. Complete Soaking in Tides.** The larger surface tends to accelerate the exchange between water and sand, causing greater destruction to the foundation. So we consider $I_2(u) := \int_\Omega \sqrt{1 + \|\nabla u(\boldsymbol{x})\|^2} \mathrm{d}\boldsymbol{x}$.
- **Step III. The constrained dynamic optimization problem.**

$$\min_{u,\Omega} : I(u) = \int_\Omega \left(\alpha e^{-u} \|\nabla u\| + \beta \sqrt{1 + \|\nabla u(\boldsymbol{x})\|^2} \right) \mathrm{d}\boldsymbol{x}$$

$$\text{s.t.} \int_\Omega u(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = V, \qquad u|_{\partial\Omega} \equiv 0.$$

- **Step IV. Solve** the OPT problem in Step III by **Variational method**.



  - **Simplification.** Symmetry and Polar Decomposition.
  - **The 2d problem.**

$$\min_{u,R} \; I(u(r),R) = 2\pi \int_0^R \left(\alpha e^{-u}|u'| + \beta\sqrt{1+|u'|^2}\right) r\,dr = 2\pi \int_0^R F(r,u,u')\,dr$$
$$s.t. \begin{cases} \int_0^R u(r) r dr = \frac{V}{2\pi} \\ u|_{r=R} = 0 = \phi(R) \end{cases} \tag{3}$$

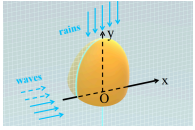
  - **Analytic Solution** through Variational Method. Combining the Constrained Conditions, its Euler-Lagrange Equation, and its Transversality Condition, we can obtain the final ODE:

$$u'' = \begin{cases} -\frac{u'(1+u'^2)}{r} + \frac{\alpha}{\beta r} e^{-u}(1+u'^2)^{3/2}, u' \le 0 \\ -\frac{u'(1+u'^2)}{r} - \frac{\alpha}{\beta r} e^{-u}(1+u'^2)^{3/2}, u' > 0 \end{cases} \tag{6}$$

The Longest Lasting Sandcastles

Problem 3. Diffusion-based Model for Erosion by Seawater and Rain

- **Step I. Simplification.** Symmetry, 3d→2d.
- **Step II. Diffusion PDE.**
  - Let $S(x, y, t)$ be the concentration of the sand at the point (x, y) on the vertical plane at time t.
  - Diffusion Law: $\boldsymbol{Q} = -\boldsymbol{D}\nabla S$, where $\boldsymbol{Q}$ is the flow velocity; $\boldsymbol{D}$ is the diffusion coefficient.
  - Establish the Equation of Transfer of Sand.



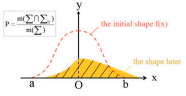
$$\left(S|_{t+\Delta t} - S|_t\right)\Delta x\Delta y = \left(\boldsymbol{q}|_x - \boldsymbol{q}|_{x+\Delta x}\right)\Delta y\Delta t + \left(\boldsymbol{q}|_y - \boldsymbol{q}|_{y+\Delta y}\right)\Delta x\Delta t$$

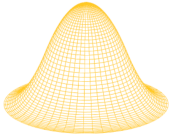
After substituting and rearranging the equation, we have the general equ

$$\frac{\partial S}{\partial t} = -\frac{\partial}{\partial x}\left(v_x S\right) + \frac{\partial}{\partial x}\left(D\frac{\partial S}{\partial x}\right) + \frac{\partial}{\partial y}\left((v_2 - \sigma)S\right)$$

$$\begin{cases} -w_x S - \frac{K\gamma}{f(0)-y}\frac{\partial S}{\partial n} = 0, (x, y) \in \partial\Sigma_+ \\ -w_x S - \frac{K\gamma}{f(0)-y}\frac{\partial S}{\partial n} = F, (x, y) \in \partial\Sigma \cap \{y = 0\} \end{cases} \quad (14)$$

- **Step III. Area-Based Evaluation Index** $P := \frac{m(\Sigma \cap \Sigma_T)}{m(\Sigma)}$.



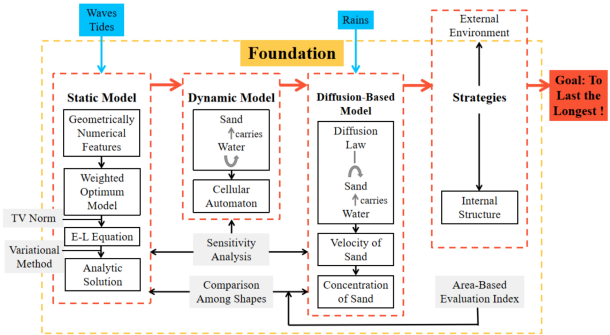


- **Step IV. Compare some shapes by** P.

The Longest Lasting Sandcastles

小结：典型的基于机理的建模问题

数学工具：动态最优化（变分法），偏微分方程等



原料分装问题

原料分装问题

“原料分装问题”——2020 年浙江大学数学建模竞赛 B 题
“基于整数规划的原料分装问题新探”——王铭泽（前 1%）

- 问题背景：一大型化工企业的材料科负责生产所用各种精细辅料向单位的发放。多数情况下生产单位所需的辅料数量远小于其规格，故而材料科需将原装辅料重新分装为不同规格的小包辅料。由于分装有严格的质控要求，小包辅料的规格为整数，且种类不宜过多，通常在 $2-8$ 种之间。接获申请时，至多将两包相同或不同规格的小包交付生产单位。在满足交付规格总和与实际所需规格之间的误差不超过 r 的情况下，考虑怎样分装能满足尽可能多种类的生产需求。

原料分装问题

“原料分装问题”——2020 年浙江大学数学建模竞赛 B 题
“基于整数规划的原料分装问题新探”——王铭泽（前 1%）

- 问题背景：一大型化工企业的材料科负责生产所用各种精细辅料向单位的发放。多数情况下生产单位所需的辅料数量远小于其规格，故而材料科需将原装辅料重新分装为不同规格的小包辅料。由于分装有严格的质控要求，小包辅料的规格为整数，且种类不宜过多，通常在 $2-8$ 种之间。接获申请时，至多将两包相同或不同规格的小包交付生产单位。在满足交付规格总和与实际所需规格之间的误差不超过 r 的情况下，考虑怎样分装能满足尽可能多种类的生产需求。

- 问题概述：
  1. 给定辅料的原装规格 P，小包辅料最小规格 q，给定小包规格种类数 k，求解小包规格以满足最多种类的需求。
  2. 在某月具体生产需求情况下，考虑采用怎样的小包规格设置可在小包规格种类数尽可能少的情况下，满足尽可能多的生产需求，同时尽量避免一种规格小包的月用量过小。附件给出的两种辅料需求情况，规定 A 辅料分装规格为 5 的整数倍，B 辅料分装规格为原装规格的一较简分数。
  3. 若材料科每月末提前分装下月生产所需辅料，需提供策略以使其达到更好的效果。

原料分装问题

问题一分析与建模

问题需求

- 接获申请时，至多将两包相同或不同规格的小包交付生产单位。
- 给定辅料的原装规格 P，小包辅料最小规格 q，给定小包规格种类数 k。
- 在满足交付与实际所需数量之间的误差不超过 r 的情况下，求解小包规格（整数）以满足最多种类的需求。

分析建模

- 设小包规格为正整数 $q_1, \cdots, q_k$，不妨设 $q \le q_1 < \cdots < q_k \le P$.
- 为描述小包规格的所有可能组合方式，补充定义 $q_0 = 0$ 后引入整数变量 q_{ij}，满足：$q_{ij} = q_i + q_j$, $1 \le i \le j \le k$ 但 $(i,j) \ne (0,0)$。记指标集 $\mathbf{\Omega}^k = \{(i,j) : 1 \le i \le j \le k\} - \{(0,0)\}$。
- 设生产需求 $x \in \mathbb{R}_+$，则 x 可被满足 $\iff \exists (i,j) \in \mathbf{\Omega}^k$, s.t. $x \in [\frac{q_{ij}}{1+r}, \frac{q_{ij}}{1-r}]$
- 最大化区间覆盖 $\max : card\left([1, \frac{2P}{1-r}] \cap \mathbb{Z} \cap \left(\cup_{(i,j)\in\mathbf{\Omega}^k}[\frac{q_{ij}}{1+r}, \frac{q_{ij}}{1-r}]\right)\right)$.

原料分装问题

问题一分析与建模

可以直接建立下列组合优化问题（整数非线性规划）

$$\max : card\left([q, \lfloor \frac{2P}{1-r} \rfloor] \bigcap \mathbb{Z} \bigcap \left(\bigcup_{(i,j)\in \mathbf{\Omega}^k} [\frac{1}{1+r} q_{ij}, \frac{1}{1-r} q_{ij}] \right) \right) \tag{3}$$

$$s.t. \quad \begin{cases} q_{ij} = q_i + q_j & ,(i,j) \in \mathbf{\Omega}^k. \\ q \le q_1 \le \cdots \le q_k \le P. \end{cases} \tag{4}$$

$$\text{其中} \quad \begin{cases} q_0 = 0. \\ q_i \in \mathbb{Z} \cap [q, P] & ,1 \le i \le k. \\ q_{ij} \in \mathbb{Z} \cap [q, \lfloor \frac{2P}{1-r} \rfloor] & ,(i,j) \in \mathbf{\Omega}^k. \end{cases} \tag{5}$$

原料分装问题

问题一分析与建模

可以直接建立下列组合优化问题（整数非线性规划）

$$\max : card\left([q, \lfloor \frac{2P}{1-r} \rfloor] \bigcap \mathbb{Z} \bigcap \left(\bigcup_{(i,j)\in \mathbf{\Omega}^k} [\frac{1}{1+r} q_{ij}, \frac{1}{1-r} q_{ij}] \right) \right) \tag{3}$$

$$s.t. \quad \begin{cases} q_{ij} = q_i + q_j & ,(i,j) \in \mathbf{\Omega}^k. \\ q \le q_1 \le \cdots \le q_k \le P. \end{cases} \tag{4}$$

$$\text{其中} \quad \begin{cases} q_0 = 0. \\ q_i \in \mathbb{Z} \cap [q, P] & ,1 \le i \le k. \\ q_{ij} \in \mathbb{Z} \cap [q, \lfloor \frac{2P}{1-r} \rfloor] & ,(i,j) \in \mathbf{\Omega}^k. \end{cases} \tag{5}$$

- 整数规划，但非线性规划。
- 求解：如模拟退火算法，但维数灾难明显。

是否可以通过等价数学转换降低问题难度？

原料分装问题

分析与建模

是否可以降低问题难度？思路：**非线性转为线性**。主要技巧：

- **先引入决策变量**：$x_{ijk} \in \{0,1\}; z_k \in \{0,1\}.$
- **再等价数学转换**：

定理 1. 设M为正整数，x取值为$[0,M]$中的整数，y为$0-1$变量,则:非线性约束$y=\boldsymbol{\chi}_{[1,M]}(x)$可等价转化为线性约束$M(y-1)+1 \le x < My+1$.（其中$\boldsymbol{\chi}_{[1,M]}(x)$表示$[1,M]$的特征函数）

定理 2. 设x取值为整数，且有上界M，且$\frac{11}{10}z < M$，且x,z都是非负整数变量，y是$0-1$变量,则：非线性约束$y=\boldsymbol{\chi}_{[\frac{10}{11}z,\frac{11}{10}z]}(x) \Leftrightarrow$线性约束

$$\begin{cases} y \le y_1,\ y \le y_2,\ y_1+y_2-1 \le y \\ M(y_1-1)+\frac{10}{11}z \le x < My_1+\frac{10}{11}z \\ \frac{11}{10}z - My_2 < x \le M(1-y_2)+\frac{11}{10}z \\ \text{其中} y_i=0,1 \quad ,i=1,2. \end{cases}.$$

原料分装问题

分析与建模

是否可以降低问题难度？思路：**非线性转为线性**。主要技巧：

- 先引入决策变量：$x_{ijk} \in \{0,1\}; z_k \in \{0,1\}$.
- 再等价数学转换：

定理 1. 设M为正整数，x取值为$[0,M]$中的整数，y为$0-1$变量,则:非线性约束$y=\boldsymbol{\chi}_{[1,M]}(x)$可等价转化为线性约束$M(y-1)+1 \le x < My+1$.（其中$\boldsymbol{\chi}_{[1,M]}(x)$表示$[1,M]$的特征函数）

定理 2. 设x取值为整数，且有上界M，且$\frac{11}{10}z < M$，且x,z都是非负整数变量，y是$0-1$变量,则：非线性约束$y=\boldsymbol{\chi}_{[\frac{10}{11}z,\frac{11}{10}z]}(x)$ $\Leftrightarrow$线性约束

$$\begin{cases} y \le y_1,\ y \le y_2,\ y_1+y_2-1 \le y \\ M(y_1-1)+\frac{10}{11}z \le x < My_1+\frac{10}{11}z \\ \frac{11}{10}z - My_2 < x \le M(1-y_2)+\frac{11}{10}z \\ \text{其中} y_i = 0,1 \quad ,i=1,2. \end{cases}.$$

将**整数非线性规划问题**等价转换为**整数线性规划问题**。

- 求解：分支定界法，割平面法

$$\max: \sum_{t=q}^{\lfloor \frac{2P}{1-r} \rfloor} z_t$$

$$s.t. \begin{cases} 4P(z_t-1)+1 \le \sum\limits_{(i,j)\in\mathbf{\Omega}^k} x_{ijt} < 4Pz_t+1. \\ x_{ijt} \le x_{ijtl}. \\ \sum\limits_{l=1}^{2} x_{ijtl} - 1 \le x_{ijt}. \\ 4P(x_{ijt1}-1)+\frac{1}{1+r}q_{ij} \le t < 4Px_{ijt1}+\frac{1}{1+r}q_{ij}. \\ \frac{1}{1-r}q_{ij} - 4Px_{ijt2} < t \le M(1-x_{ijt2})+\frac{1}{1-r}q_{ij}. \\ q_{ij} = q_i + q_j. \\ q \le q_1 \le \cdots \le q_k \le P. \end{cases} \tag{11}$$

$$\text{其中} \begin{cases} q_0 = 0. \\ q_i \in \mathbb{Z} \cap [q, P] \quad ,1 \le i \le k. \\ q_{ij} \in \mathbb{Z} \cap [q, \lfloor \frac{2P}{1-r} \rfloor] \quad ,(i,j) \in \mathbf{\Omega}^k. \\ x_{ijt} = 0,1 \quad ,(i,j) \in \mathbf{\Omega}^k \ \ t = q, q+1, \cdots, \lfloor \frac{2P}{1-r} \rfloor. \\ x_{ijtl} = 0,1 \quad ,(i,j) \in \mathbf{\Omega}^k \ \ t = q, q+1, \cdots, \lfloor \frac{2P}{1-r} \rfloor \ \ l = 1,2. \\ z_t = 0,1 \quad ,t = q, q+1, \cdots, \lfloor \frac{2P}{1-r} \rfloor. \end{cases}$$

原料分装问题

问题一求解结果

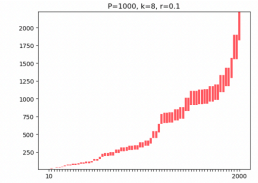


图 2: 纵轴表示生产需求数量，横轴表示小包及小包组合规格，将所有线段向纵轴投影可观测出生产需求满足情况。$P = 1000, k = 8, r = 10\%$时最优解的生产需求满足率为 **99.32%**。

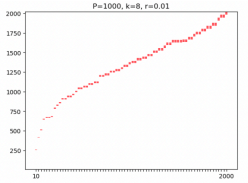


图 3: 纵轴表示生产需求数量，横轴表示小包及小包组合规格，将所有线段向纵轴投影可观测出生产需求满足情况。$P = 1000, k = 8, r = 1\%$时最优解的生产需求满足率为 **50.27%**。

- **问题 $P = 1000, k = 8, r = 10\%$ 的生产需求可能取值共有 2198 种，最优解所的生产需求满足率达到 $\mathbf{99.32}\%$，几乎覆盖了生产需求取值集合。**
- **问题 $P = 1000, k = 8, r = 1\%$ 的生产需求可能取值共有 2002 种，最优解所的生产需求满足率仅有 $\mathbf{50.27}\%$，大约覆盖了生产需求取值集合的一半。**
- ……

原料分装问题

- 问题二：只需将单目标优化转为**多目标优化**。
- 问题三：引入适当假设，使用 Markov 链及时间序列模型，将问题一中优化目标改为**概率优化**问题即可。

小结：组合优化建模问题

数学方法：整数非线性规划，整数线性规划，多目标规划，Markov 链

Section 2. Some Progress on Machine Learning Theory

Introduction to Supervised Learning

- In **supervised learning**, we are given n training data $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ drawn i.i.d. from unknown distribution $\mathcal{D}$.

Introduction to Supervised Learning

- In **supervised learning**, we are given n training data $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ drawn i.i.d. from unknown distribution $\mathcal{D}$.
- We aim to minimize the **population risk (PRM)**:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} : \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathcal{D}}[\ell(y; f(\boldsymbol{x}; \boldsymbol{\theta}))],$$

  - $\ell(\cdot; \cdot)$ is loss function, such as square loss $\ell(y_1, y_2) = (y_1 - y_2)^2/2$, logistic loss $\ell(y_1, y_2) = \log(1 + \exp(-y_1 y_2))$, and cross-entropy loss.
  - $f(\cdot; \boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}$ is the prediction model parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$, such as linear model $f(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{x}$ and deep fully-connected neural networks (FCN) $f(\boldsymbol{x}; \boldsymbol{\theta}) = \sigma(\boldsymbol{W}^L \sigma(\cdots(\sigma(\boldsymbol{W}^1 \boldsymbol{x}))))$.

Introduction to Supervised Learning

- In **supervised learning**, we are given n training data $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ drawn i.i.d. from unknown distribution $\mathcal{D}$.
- We aim to minimize the **population risk (PRM)**:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} : \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathcal{D}}[\ell(y; f(\boldsymbol{x}; \boldsymbol{\theta}))],$$

  - $\ell(\cdot; \cdot)$ is loss function, such as square loss $\ell(y_1, y_2) = (y_1 - y_2)^2/2$, logistic loss $\ell(y_1, y_2) = \log(1 + \exp(-y_1 y_2))$, and cross-entropy loss.
  - $f(\cdot; \boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}$ is the prediction model parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$, such as linear model $f(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{x}$ and deep fully-connected neural networks (FCN) $f(\boldsymbol{x}; \boldsymbol{\theta}) = \sigma(\boldsymbol{W}^L \sigma(\cdots(\sigma(\boldsymbol{W}^1 \boldsymbol{x}))))$.
- However, we can only minimize the **empirical risk (ERM)** on the training dataset $\mathcal{S}$:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} : \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \ell(y_i; f(\boldsymbol{x}_i; \boldsymbol{\theta}))$$

Introduction to Supervised Learning

- In **supervised learning**, we are given n training data $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ drawn i.i.d. from unknown distribution $\mathcal{D}$.
- We aim to minimize the **population risk (PRM)**:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} : \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathcal{D}}[\ell(y; f(\boldsymbol{x}; \boldsymbol{\theta}))],$$

  - $\ell(\cdot; \cdot)$ is loss function, such as square loss $\ell(y_1, y_2) = (y_1 - y_2)^2/2$, logistic loss $\ell(y_1, y_2) = \log(1 + \exp(-y_1 y_2))$, and cross-entropy loss.
  - $f(\cdot; \boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}$ is the prediction model parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$, such as linear model $f(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{x}$ and deep fully-connected neural networks (FCN) $f(\boldsymbol{x}; \boldsymbol{\theta}) = \sigma(\boldsymbol{W}^L \sigma(\cdots(\sigma(\boldsymbol{W}^1 \boldsymbol{x}))))$.
- However, we can only minimize the **empirical risk (ERM)** on the training dataset $\mathcal{S}$:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} : \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \ell(y_i; f(\boldsymbol{x}_i; \boldsymbol{\theta}))$$

  through some optimization algorithms, such as (Stochastic) Gradient Descent with random initialization:

$$\textbf{GD:} \quad \boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta_t \nabla \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}(t)),$$

$$\textbf{SGD:} \quad \boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \frac{\eta_t}{B} \sum_{i \in \mathcal{B}_t} \nabla \ell(f(\boldsymbol{x}_i; \boldsymbol{\theta}(t)), \boldsymbol{y}_i),$$

  where $\mathcal{B}_t = \{\gamma_{t,1}, \cdots, \gamma_{t,B}\}$ is a batch, and $\gamma_{t,1}, \cdots, \gamma_{t,B} \overset{\text{i.i.d.}}{\sim} \mathbb{U}([n])$ and are independent with $\boldsymbol{\theta}(t)$.

Two Main Problems in Supervised Learning.

- **Optimization.** Can we use some Optimization Algorithms such as Gradient Descent to find the global minimum $\boldsymbol{\theta}^*$ of empirical risk $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$? i.e. find $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$.

Two Main Problems in Supervised Learning.

- **Optimization.** Can we use some Optimization Algorithms such as Gradient Descent to find the global minimum $\boldsymbol{\theta}^*$ of empirical risk $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$? i.e. find $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$.
- **Generalization.** Does the global minimum $\boldsymbol{\theta}^*$ of $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$ (found by some optimization algorithms) have small population risk? i.e. small $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*)$

Introduction to Machine Learning Theory

Two Main Problems in Supervised Learning.

- **Optimization.** Can we use some Optimization Algorithms such as Gradient Descent to find the global minimum $\boldsymbol{\theta}^*$ of empirical risk $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$? i.e. find $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$.
- **Generalization.** Does the global minimum $\boldsymbol{\theta}^*$ of $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$ (found by some optimization algorithms) have small population risk? i.e. small $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*)$

Machine Learning Theory = Approximation + **Optimization** + **Generalization**.

We mainly focus on **Deep Learning Theory**, which means the prediction models $f(\cdot;\boldsymbol{\theta})$ are Neural Networks.

Optimization Dynamics of Training Neural Networks

1. 几个数学建模问题
   - 高压油管的压力控制
   - The Longest Lasting Sandcastles
   - 原料分装问题

2. **Some Progress on Machine Learning Theory**
   - **Optimization Dynamics of Training Neural Networks**
   - Implicit Bias/Regularization of Stochastic Gradient Descent
   - Designing Algorithms Inspired by Theory

Theoretical Challenge in Optimization

Theoretical Challenge in Optimization. In this part, we denote $\mathcal{L}(\boldsymbol{\theta}) := \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$.

- **Fact:** Training Neural Networks is **Non-convex Non-smooth** Optimization problem!
- **Theoretical Side:** Even finding a local minimum is **NP-hard!**
- **Practical observation:** Gradient-based optimization methods often find high quality solutions.

Theoretical Challenge in Optimization

Theoretical Challenge in Optimization. In this part, we denote $\mathcal{L}(\boldsymbol{\theta}) := \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$.

- **Fact:** Training Neural Networks is **Non-convex Non-smooth** Optimization problem!
- **Theoretical Side:** Even finding a local minimum is **NP-hard!**
- **Practical observation:** Gradient-based optimization methods often find high quality solutions.

Previous Theoretical Results: Highly over-parameterized regime

Theorem (Neual Tanget Kernel Theory (Du et. al, 2018))

Let the parameters of NNs $\boldsymbol{\theta}(t)$ be trained by Gradient Descent started with random initialization. If the network width is large enough $m \geq \text{poly}(n, 1/\lambda_0 \cdots)$, then with high probability, $\mathcal{L}(\boldsymbol{\theta}(t)) \leq (1 - \eta\lambda_0)^t \mathcal{L}(\boldsymbol{\theta}(0))$.

- **Good News.** Global exponential convergence.
- **Bad News.** In this regime, (1) the network width is not practical; (2) the network is close to a kernel method (linear model); (3) the optimization problem is nearly convex; (4) converged solution is no better than that of kernel method.

Gap between highly over-parameterized and practical-size NNs

Highly over-parameterized NNs $\approx$ Kernel methods $\neq$ Practical-size NNs.

- Highly over-parameterized NNs keep close to kernel methods (lazy training)

$$f(\boldsymbol{x};\boldsymbol{\theta}(t)) \approx f(\boldsymbol{x};\boldsymbol{\theta}(0)) + \langle \nabla f(\boldsymbol{x};\boldsymbol{\theta}(0)), \boldsymbol{\theta}(t) - \boldsymbol{\theta}(0) \rangle .$$

- However, NNs have obvious superiorities to kernel methods to learn even a single ReLU neuron

$$\min_{\boldsymbol{\theta}} : \mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)} \Big[\Big(f(\boldsymbol{x};\boldsymbol{\theta}) - \mathrm{ReLU}(\boldsymbol{w}^{*\top}\boldsymbol{x} + b^*) \Big)^2 \Big].$$

Theorem (Shamir et. al, 2018, 2019)

1. **NNs with even one neuron** can learn single neuron **efficiently** at a linear rate.
2. **Random features** suffer from the **"curse of dimensionality"**, i.e. they fail unless the network size is **exponentially large** $\Omega(e^d)$ with respect to the input dimension d.

Optimization theory for practical-size NNs[1]

General Non-convex Optimization is NP-hard. However,

Phenomenon. Fast decreasing of the loss value always happens, at least in the early stage of training.

- It is common that the loss experiences a drastic decreasing **at the beginning of the training**.
- In many cases, this decrease of loss even continues until **the loss achieves** 0, i.e. the optimization algorithm fully converges.

[1] Mingze Wang and Chao Ma. "Early Stage Convergence and Global Convergence of Training Mildly Parameterized Neural Networks". In: *NeurIPS* (2022).

Optimization theory for practical-size NNs[1]

General Non-convex Optimization is NP-hard. However,

Phenomenon. Fast decreasing of the loss value always happens, at least in the early stage of training.

- It is common that the loss experiences a drastic decreasing **at the beginning of the training**.
- In many cases, this decrease of loss even continues until **the loss achieves** 0, i.e. the optimization algorithm fully converges.

Theoretical Problems. When we train **practical-size (mildly parameterized) NNs** by GD or SGD,

1. Does the **fast convergence in the early stage** of the training provably exist? If so, how long will the phenomenon last and how much will loss descend in the early stage?
2. Can the **global convergence** be proved under some special conditions on loss function and training data?

[1] Mingze Wang and Chao Ma. "Early Stage Convergence and Global Convergence of Training Mildly Parameterized Neural Networks". In: *NeurIPS* (2022).

- **Loss:** a large number of loss functions, such as quadratic loss, exponential-type loss and hinge-loss.
- **Data:** $\|\boldsymbol{x}\| \leq 1$ and there exists $s > -1$ s.t. $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \geq s$ for any $i, j \in [n]$. It holds for normalized image datasets such as MNIST and CIFAR-10.

Theorem (Early Stage Convergence of Training Mildly Parameterized NNs)

Let $\boldsymbol{\theta}(t)$ be the parameters of two-layer ReLU NNs trained by GD or SGD. If the width $m = \Omega(\log n)$ **(mildly parameterized)**, the input dimension $d = \Omega(\log m)$, and the learning rate $\eta_t = \eta \lesssim 1$, then with high probability, **the loss will descend** $\Omega(1)$ **in** $T = \Theta(\frac{1}{\eta})$ **iterations**.

- **Loss:** Exponential-type loss, such as exponential, logistic and cross-entropy loss.
- **Data:** n is even. $y_i = 1$ for $i \in [n/2]$; $y_i = -1$ for $i \in [n] - [n/2]$. $\boldsymbol{x}_i^\top \boldsymbol{x}_j \geq 0$ for i, j in the same class; $\boldsymbol{x}_i^\top \boldsymbol{x}_j \leq 0$ for i, j in different classes.

Theorem (Global Convergence of Training Mildly Parameterized NNs)

1. Let $\boldsymbol{\theta}(t)$ be the parameters of two-layer ReLU NNs trained by GD starting from random initialization. Let the width $m = \Omega(\log n)$ and the input dimension $d = \Omega(\log m)$. Then, with adaptive learning rate, with high probability, GD will converge at **arbitrary polynomial rate** $r > 0$: $\mathcal{L}(\boldsymbol{\theta}(t)) = \mathcal{O}\big(1/t^r\big)$.
2. If only the first layer be trained, GD will converge at **exponential rate**: $\mathcal{L}(\boldsymbol{\theta}(t)) \leq \left(1 - \frac{Vc}{2}\right)^{t-1} \log 2$.

Optimization Dynamics and Comparison

Optimization Dynamics. The loss landscape may be complicated near the random initialization. However,

- **Stage I (Feature Learning)**. Neurons will adjust directions rapidly, and the iterator will enter a good region which contains neither spurious local minima nor saddle points.
- **Stage II (Lazy Training)**. Then, neurons will keep going towards the right directions for a period of time, during which process the loss will descend fast and significantly.

We provide detailed **theorems**, **explanations**, and **proofs** (73 pages). Please refer to our paper[2] for details.

Comparison with highly over-parameterized NNs.

	mildly parameterized NNs	highly over-parameterized NNs
network width m	$\Omega(\log n)$ (practical-size)	$\Omega(\mathrm{poly}(n, 1/\lambda_0))$
model	highly non-linear	nearly linear
convexity	highly non-convex	nearly convex
training dynamics	feature learning	lazy training

[2] Mingze Wang and Chao Ma. "Early Stage Convergence and Global Convergence of Training Mildly Parameterized Neural Networks". In: *NeurIPS* (2022).

Optimization dynamics of training DNNs

A crucial theoretical topic: understanding the optimization dynamics of training DNNs.

- **Most previous works** focus on:
  - either local analysis, like the initial/end of training;
  - or approximate linear models, like Neural Tangent Kernel.

Optimization dynamics of training DNNs

A crucial theoretical topic: understanding the optimization dynamics of training DNNs.

- **Most previous works** focus on:
  - either local analysis, like the initial/end of training;
  - or approximate linear models, like Neural Tangent Kernel.
- **However**, the training of practical networks can exhibit plenty of **nonlinear behaviors**:

Optimization dynamics of training DNNs

A crucial theoretical topic: understanding the optimization dynamics of training DNNs.

- **Most previous works** focus on:
  - either local analysis, like the initial/end of training;
  - or approximate linear models, like Neural Tangent Kernel.
- **However**, the training of practical networks can exhibit plenty of **nonlinear behaviors**:
  - **Initial training**: **initial condensation**, i.e., neurons condense onto a few isolated orientations.
  - **End of training**: for exp-tailed loss (classification), NNs **directionally converge to KKT points** of some constrained problem. However, determining which KKT point (not unique) GD converges to is challenging.
  - Nonlinear training behaviors **besides initial and terminating stages** of optimization are also numerous:
    - **Saddle-to-saddle dynamics**: for square loss, GD traverses a sequence of saddles during training. But it is unclear whether similar behavior can occur for classification tasks using exp-tailed loss.
    - **Changes of activation patterns**. For ReLU nets, most activation patterns $\mathbb{I}\{\boldsymbol{w}^\top \boldsymbol{x} > 0\}$ do not change during training in lazy regime, it remains uncertain how patterns evolve beyond lazy regime.
    - **Learning of increasing complexity**, also known as simplifying-to-complicating or frequency-principle has yet to be proven.

Optimization Dynamics of Training Neural Networks

Our first work (Wang and Ma (NeurIPS 2022)) explores the complete dynamics on classifying orthogonally separable data. However,

- this data is easy to learn, and all the features can be learned rapidly (accuracy=100%) in initial training, followed by lazy training (activation patterns do not change).
- Unfortunately, this simplicity does not hold true for actual tasks on much more complex data, and NNs can only learn some features in initial training, which complicates the overall learning process.

[3] Mingze Wang and Chao Ma. "Understanding Multi-phase Optimization Dynamics and Rich Nonlinear Behaviors of ReLU Networks". In: *arXiv preprint arXiv:2305.12467* (2023).

Optimization Dynamics of Training Neural Networks

Our first work (Wang and Ma (NeurIPS 2022)) explores the complete dynamics on classifying orthogonally separable data. However,

- this data is easy to learn, and all the features can be learned rapidly (accuracy=100%) in initial training, followed by lazy training (activation patterns do not change).
- Unfortunately, this simplicity does not hold true for actual tasks on much more complex data, and NNs can only learn some features in initial training, which complicates the overall learning process.

In this work (Wang and Ma (2023))[3], we make an attempt to theoretically describe the whole neural network training dynamics beyond the linear regime, in a setting that many nonlinear behaviors manifest.

- We analyze the training process of a two-layer ReLU net trained by GF on a linearly separable data.
- Our analysis captures **the whole optimization process** starting from random initialization to final convergence.
- Despite the relatively simple model and data that we studied, we reveal **multiple phases** in training process, and show a general **simplifying-to-complicating learning trend** by detailed analysis of each phase.

[3] Mingze Wang and Chao Ma. "Understanding Multi-phase Optimization Dynamics and Rich Nonlinear Behaviors of ReLU Networks". In: *arXiv preprint arXiv:2305.12467* (2023).

Optimization Dynamics of Training Neural Networks

Specifically, in this work (Wang and Ma (2023)), by our meticulous theoretical analysis of the whole training process, we precisely identify **four different phases** that exhibit **numerous nonlinear behaviors**.

- In Phase I, **initial condensation and simplification** occur as living neurons rapidly condense in two different directions. Meanwhile, GF **escapes from the saddle** around initialization.
- In Phase II, GF **gets stuck into the plateau** of training accuracy for a long time, then **escapes**. The first two phases exhibit a **saddle-to-plateau dynamics**.
- In Phase III, a significant number of neurons are **deactivated**, leading to **self-simplification** of the network, then GF tries to learn using the almost simplest network.
- In Phase IV, a considerable number of neurons are **reactivated**, causing **self-complication** of the network. Finally, GF **converges towards an initialization-dependent direction**.
- Overall, the whole training process exhibits a remarkable **simplifying-to-complicating** learning trend.

A Brief Overview of four-phase Optimization

In this work, we present a meticulously detailed and comprehensive depiction of the whole optimization dynamics and nonlinear behaviors. First, we display the timeline of our dynamics and some nonlinear behaviors.

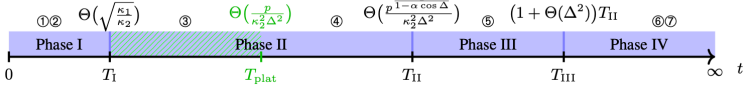
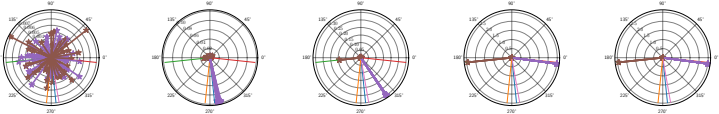


Fig: Timeline of the four-phase optimization dynamics, containing some key time points T_{I}, T_{II}, T_{III}, T_{plat} and their theoretical estimates, and some basic nonlinear behaviors: ① initial condensation, ② saddle escape, ③ getting stuck in plateau, ④ plateau escape, ⑤ neuron deactivation, ⑥ neuron reactivation, ⑦ initialization-dependent directional convergence. Notice ①∼⑦ are only some basic nonlinear behaviors. Moreover, ②+③ is saddle-to-plateau, ①+⑤+⑥ is simplifying-to-complicating.



We provide detailed **theorems**, **explanations**, and **proofs** (88 pages). Please refer to our paper[4] for details.

[4] Mingze Wang and Chao Ma. "Understanding Multi-phase Optimization Dynamics and Rich Nonlinear Behaviors of ReLU Networks". In: *arXiv preprint arXiv:2305.12467* (2023).

Implicit Bias of Stochastic Gradient Descent

1. 几个数学建模问题
   - 高压油管的压力控制
   - The Longest Lasting Sandcastles
   - 原料分装问题

2. **Some Progress on Machine Learning Theory**
   - Optimization Dynamics of Training Neural Networks
   - **Implicit Bias/Regularization of Stochastic Gradient Descent**
   - Designing Algorithms Inspired by Theory

Introduction to Implicit Bias/Regularization

- Modern neural networks are usually **over-parameterized**, i.e. $p \gg n$, where p is the dimension of $\boldsymbol{\theta}$ and n is the sample size.

Introduction to Implicit Bias/Regularization

- Modern neural networks are usually **over-parameterized**, i.e. $p \gg n$, where p is the dimension of $\boldsymbol{\theta}$ and n is the sample size.
- There are plenty of global minima $\boldsymbol{\theta}^*$ of $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$, all of which have zero training loss ($\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}^*) = 0$) but their test performance can be **significantly different** (different $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*)$).

Introduction to Implicit Bias/Regularization

- Modern neural networks are usually **over-parameterized**, i.e. $p \gg n$, where p is the dimension of $\boldsymbol{\theta}$ and n is the sample size.
- There are plenty of global minima $\boldsymbol{\theta}^*$ of $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$, all of which have zero training loss ($\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}^*) = 0$) but their test performance can be **significantly different** (different $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*)$).
- To get good generalization, one may think that we must rely on some **explicit regularization tricks**, such as Weight Decay, Data Augmentation, Dropout, Batch Normalization, etc.

Introduction to Implicit Bias/Regularization

- Modern neural networks are usually **over-parameterized**, i.e. $p \gg n$, where p is the dimension of $\boldsymbol{\theta}$ and n is the sample size.
- There are plenty of global minima $\boldsymbol{\theta}^*$ of $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$, all of which have zero training loss ($\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}^*) = 0$) but their test performance can be **significantly different** (different $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*)$).
- To get good generalization, one may think that we must rely on some **explicit regularization tricks**, such as Weight Decay, Data Augmentation, Dropout, Batch Normalization, etc.
- **Surprisingly**, practitioners often find that optimizers (such as SGD) can find good solutions ($\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*)$ is small) **without** the need of any explicit regularization.

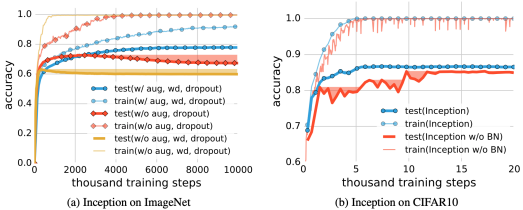
Introduction to Implicit Bias/Regularization

- Modern neural networks are usually **over-parameterized**, i.e. $p \gg n$, where p is the dimension of $\boldsymbol{\theta}$ and n is the sample size.
- There are plenty of global minima $\boldsymbol{\theta}^*$ of $\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$, all of which have zero training loss ($\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}^*) = 0$) but their test performance can be **significantly different** (different $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*)$).
- To get good generalization, one may think that we must rely on some **explicit regularization tricks**, such as Weight Decay, Data Augmentation, Dropout, Batch Normalization, etc.
- **Surprisingly**, practitioners often find that optimizers (such as SGD) can find good solutions ($\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*)$ is small) **without** the need of any explicit regularization.

Implicit Bias:

Without any explicit regularization tricks, optimizers converges to generalizable global minima!

Introduction to Implicit Bias/Regularization



(a) Inception on ImageNet (b) Inception on CIFAR10

Fig: Effect of implicit regularization (in Zhang et al, ICLR 2017)

- For CIFAR10 (Fig (b)), the implicit regularizations account for $\mathbf{85}\%+$ accuracy. Explicit regularizations only improve less than $\mathbf{5}\%$ accuracy.
- In ImageNet (Fig (a)), explicit regularizations are more important, but still not as crucial as the implicit bias.

Introduction to Implicit Bias/Regularization

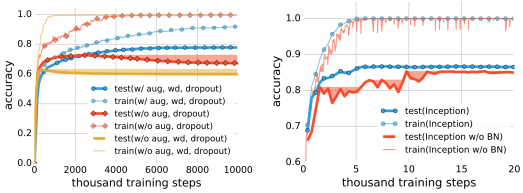


Fig: Effect of implicit regularization (in Zhang et al, ICLR 2017)

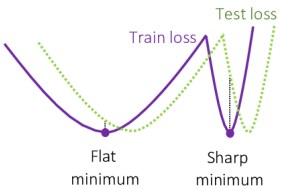
- For CIFAR10 (Fig (b)), the implicit regularizations account for $\mathbf{85}\%+$ accuracy. Explicit regularizations only improve less than $\mathbf{5}\%$ accuracy.
- In ImageNet (Fig (a)), explicit regularizations are more important, but still not as crucial as the implicit bias.

Implicit Bias: Without any explicit regularization tricks, optimizers converges to generalizable global minima!

- We focus on the implicit bias of **Stochastic Gradient Descent (SGD)**.
- There are some **main factors** that effect the implicit bias: network structure; initialization scale; learning rate η and batch size B; **structure of SGD noise**; direction of gradient.

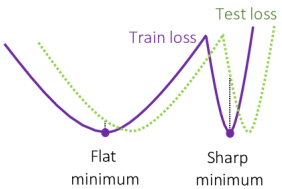
What Solutions Generalize Well

Famous Flatness Hypothesis: Flatter minima generalize better. (1997)



What Solutions Generalize Well

Famous Flatness Hypothesis: Flatter minima generalize better. (1997)



SGD → flat minima → generalize better

1. **Why SGD finds flat minima?**
2. Why flat minima generalize better? Flatness-based generalization error bounds.

Introduction to SGD Noise

- Consider the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$. Let $f(\cdot; \boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}$ be the model parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$. Let $\ell_i(\boldsymbol{\theta}) = \frac{1}{2} \left(f(\boldsymbol{x}_i; \boldsymbol{\theta}) - y_i\right)^2$ be the squared loss at the i-th sample and $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \ell_i(\boldsymbol{\theta})$ be the empirical risk.

Introduction to SGD Noise

- Consider the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$. Let $f(\cdot;\boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}$ be the model parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$. Let $\ell_i(\boldsymbol{\theta}) = \frac{1}{2}\left(f(\boldsymbol{x}_i;\boldsymbol{\theta}) - y_i\right)^2$ be the squared loss at the i-th sample and $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \ell_i(\boldsymbol{\theta})$ be the empirical risk.
- SGD can be rewritten as GD+noise:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \frac{\eta}{B}\sum_{i\in\mathcal{B}_t} \nabla \ell_i(\boldsymbol{\theta}(t)) = \boldsymbol{\theta}(t) - \eta\left(\nabla \mathcal{L}(\boldsymbol{\theta}(t)) + \boldsymbol{\xi}(t)\right),$$

where $\boldsymbol{\xi}_t$ is the noise, satisfying $\mathbb{E}[\boldsymbol{\xi}_t] = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{\xi}_t \boldsymbol{\xi}_t^\top] = \Sigma(\boldsymbol{\theta}_t)/B$. Here the **noise covariance**

$$\Sigma(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \nabla \ell_i(\boldsymbol{\theta}) \nabla \ell_i(\boldsymbol{\theta})^\top - \nabla \mathcal{L}(\boldsymbol{\theta}) \nabla \mathcal{L}(\boldsymbol{\theta})^\top.$$

Introduction to SGD Noise

- Consider the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$. Let $f(\cdot;\boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}$ be the model parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$. Let $\ell_i(\boldsymbol{\theta}) = \frac{1}{2}\left(f(\boldsymbol{x}_i;\boldsymbol{\theta}) - y_i\right)^2$ be the squared loss at the i-th sample and $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \ell_i(\boldsymbol{\theta})$ be the empirical risk.
- SGD can be rewritten as GD+noise:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \frac{\eta}{B}\sum_{i\in\mathcal{B}_t} \nabla \ell_i(\boldsymbol{\theta}(t)) = \boldsymbol{\theta}(t) - \eta\left(\nabla \mathcal{L}(\boldsymbol{\theta}(t)) + \boldsymbol{\xi}(t)\right),$$

where $\boldsymbol{\xi}_t$ is the noise, satisfying $\mathbb{E}[\boldsymbol{\xi}_t] = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{\xi}_t\boldsymbol{\xi}_t^\top] = \Sigma(\boldsymbol{\theta}_t)/B$. Here the **noise covariance**

$$\Sigma(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \nabla \ell_i(\boldsymbol{\theta}) \nabla \ell_i(\boldsymbol{\theta})^\top - \nabla \mathcal{L}(\boldsymbol{\theta}) \nabla \mathcal{L}(\boldsymbol{\theta})^\top.$$

- The **Hessian and Gram** matrix of the loss landscape can be written as

$$H(\boldsymbol{\theta}) = G(\boldsymbol{\theta}) + \frac{1}{n}\sum_{i=1}^n \left(f(\boldsymbol{x}_i;\boldsymbol{\theta}) - y_i\right) \nabla^2 f(\boldsymbol{x}_i;\boldsymbol{\theta}), \quad G(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \nabla f(\boldsymbol{x}_i;\boldsymbol{\theta}) \nabla f(\boldsymbol{x}_i;\boldsymbol{\theta})^\top.$$

When the fit errors are small, we have $G(\boldsymbol{\theta}) \approx H(\boldsymbol{\theta})$ and in particular, if $\mathcal{L}(\boldsymbol{\theta}^*) = 0$, then $H(\boldsymbol{\theta}^*) = G(\boldsymbol{\theta}^*)$. Additionally, for linear regression $f(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{x}$, $H \equiv G \equiv \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^\top$.

Noise Geometry

- **Decoupling approximation.** An intuitive approximation:

$$\Sigma(\boldsymbol{\theta}) \approx \frac{1}{n}\sum_{i=1}^{n}(f_i(\boldsymbol{\theta}) - y_i)^2 \nabla f_i(\boldsymbol{\theta}) \nabla f_i(\boldsymbol{\theta})^\top \approx 2\mathcal{L}(\boldsymbol{\theta}) G(\boldsymbol{\theta}).$$

This approximation cannot be true in general but tells us two critical properties of SGD noise:

- The noise magnitude is proportional to the loss value.
- The noise covariance aligns with the Gram matrix.

[5] Lei Wu, Mingze Wang, and Weijie Su. "The alignment property of SGD noise and how it helps select flat minima: A stability analysis". In: *NeurIPS* (2022).

[6] Mingze Wang and Lei Wu. "The Noise Geometry of Stochastic Gradient Descent: A Quantitative and Analytical Characterization". In: *under review* (2023).

Noise Geometry

- **Decoupling approximation.** An intuitive approximation:

$$\Sigma(\boldsymbol{\theta}) \approx \frac{1}{n}\sum_{i=1}^{n}(f_i(\boldsymbol{\theta}) - y_i)^2 \nabla f_i(\boldsymbol{\theta}) \nabla f_i(\boldsymbol{\theta})^\top \approx 2\mathcal{L}(\boldsymbol{\theta}) G(\boldsymbol{\theta}).$$

  This approximation cannot be true in general but tells us two critical properties of SGD noise:

  - The noise magnitude is proportional to the loss value.
  - The noise covariance aligns with the Gram matrix.

In our works[5][6], we provide theoretical explanations and quantitative characterizations of how SGD noise aligns with local loss landscape. Moreover, we apply our noise geometry results to investigate how SGD escapes from minima.

- **Weak Alignment**: The noise covariance Σ aligns with the local landscape G in an average sense.
- **Strong Alignment** (directional alignment): the component of noise energy along any direction is proportional to that direction's sharpness.

[5] Lei Wu, Mingze Wang, and Weijie Su. "The alignment property of SGD noise and how it helps select flat minima: A stability analysis". In: *NeurIPS* (2022).

[6] Mingze Wang and Lei Wu. "The Noise Geometry of Stochastic Gradient Descent: A Quantitative and Analytical Characterization". In: *under review* (2023).

Flat Minima Selection

Q: What is the role of this alignment structure of SGD noise?

A: It helps SGD select flat minima!

Flat Minima Selection

Q: What is the role of this alignment structure of SGD noise?

A: It helps SGD select flat minima!

By establishing **Linear Stability Analysis**, we prove that:

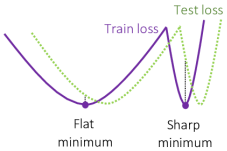
- Size-independent Flatness Bound of SGD's solutions: $\left\|\nabla^2 \mathcal{L}(\boldsymbol{\theta}^*)\right\| \leq \frac{1}{\eta}\sqrt{\frac{B}{\mu_0}}$, where B is the batch size, η is the learning rate, and μ_0 reflects how SGD noise aligns with the loss landscape.
- SGD escapes from sharp minima exponentially fast.

Designing Algorithms Inspired by Theory

Previous Algorithms inspired by Theory

Regression Task.

- **Flatness.** Flatter minima generalize better.
- **Theory.** Flatness-based generalization error bounds.
- **Algorithms.** Such as Sharpness-aware Minimization (SAM).

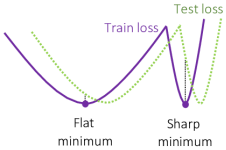


$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} : \max_{\|\boldsymbol{\epsilon}\|\leq\delta} \mathcal{L}(\boldsymbol{\theta}+\boldsymbol{\epsilon}) \approx \mathcal{L}(\boldsymbol{\theta}) + \nabla\mathcal{L}(\boldsymbol{\theta})^\top\boldsymbol{\epsilon} + \frac{1}{2}\boldsymbol{\epsilon}^\top\nabla^2\mathcal{L}(\boldsymbol{\theta})\boldsymbol{\epsilon}.$$

Previous Algorithms inspired by Theory

Regression Task.

- **Flatness.** Flatter minima generalize better.
- **Theory.** Flatness-based generalization error bounds.
- **Algorithms.** Such as Sharpness-aware Minimization (SAM).

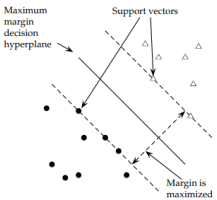


$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} : \max_{\|\boldsymbol{\epsilon}\|\leq\delta} \mathcal{L}(\boldsymbol{\theta}+\boldsymbol{\epsilon}) \approx \mathcal{L}(\boldsymbol{\theta}) + \nabla\mathcal{L}(\boldsymbol{\theta})^\top\boldsymbol{\epsilon} + \frac{1}{2}\boldsymbol{\epsilon}^\top\nabla^2\mathcal{L}(\boldsymbol{\theta})\boldsymbol{\epsilon}.$$

Classification Task.

- **Margin.** $\gamma(\boldsymbol{\theta}) := q_{\min}(\frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|})$, where $q_{\min}(\boldsymbol{\theta}) = \min_{i\in[n]} y_i f(\boldsymbol{x}_i;\boldsymbol{\theta})$ (binary classification $y_i \in \{\pm 1\}$).
- **Theory.** Margin-based generalization error bounds.
- **Algorithms.** Large-margin Learning, such as

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} : \mathcal{L}(\boldsymbol{\theta}) + \frac{\lambda}{\gamma(\boldsymbol{\theta})}.$$



Faster Margin Maximization for Logistic Regression

Problem Settings

- Given a dataset $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}_{i=1}^n \subset \mathbb{R}^d \times \{\pm 1\}$. WLOG, we assume $\|\boldsymbol{x}_i\|_2 \leq 1$ for any i.
- **Linearly Separable.** The margin of the dataset $\gamma^* = \max_{\boldsymbol{w} \in \mathbb{S}^{d-1}} \min_{i \in [n]} y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0$.
- **Logistic Regression.**

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^n \log\left(1 + \exp\left(-y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle\right)\right).$$

Faster Margin Maximization for Logistic Regression

Problem Settings

- Given a dataset $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}_{i=1}^n \subset \mathbb{R}^d \times \{\pm 1\}$. WLOG, we assume $\|\boldsymbol{x}_i\|_2 \leq 1$ for any i.
- **Linearly Separable.** The margin of the dataset $\gamma^* = \max_{\boldsymbol{w} \in \mathbb{S}^{d-1}} \min_{i \in [n]} y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0$.
- **Logistic Regression.**

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^n \log\left(1 + \exp\left(-y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle\right)\right).$$

- Vanilla Gradient Desecnt (GD) and Normalized Gradient Descent (NGD)

$$\textbf{GD:} \qquad \boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \eta \nabla \mathcal{L}(\boldsymbol{w}(t)),$$

$$\textbf{NGD:} \qquad \boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \eta \frac{\nabla \mathcal{L}(\boldsymbol{w}(t))}{\mathcal{L}(\boldsymbol{w}(t))}.$$

- **Margin** of $\boldsymbol{w}$. $\gamma(\boldsymbol{w}) := \min_{i \in [n]} y_i \left\langle \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}, \boldsymbol{x}_i \right\rangle$.

Faster Margin Maximization for Logistic Regression

Theory

Theorem (Soudry, 2018)

For GD starting form any $\boldsymbol{w}(0) \in \mathbb{R}^d$ with constant η, we have $\gamma^* - \gamma(\boldsymbol{w}(t)) = \tilde{\mathcal{O}}\left(\frac{1}{\log t}\right)$.

[7] Mingze Wang. "Maximize Margin Nearly Exponentially Fast by First-order Optimization Method". In: *preparing* (2023).

Faster Margin Maximization for Logistic Regression

Theory

Theorem (Soudry, 2018)

For GD starting form any $\boldsymbol{w}(0) \in \mathbb{R}^d$ with constant η, we have $\gamma^* - \gamma(\boldsymbol{w}(t)) = \tilde{\mathcal{O}}\left(\frac{1}{\log t}\right)$.

Work	Algorithm	Convergence Rate of Margin Maximization
(Soudry, 2018)	GD	$\tilde{\mathcal{O}}\left(\frac{1}{\log t}\right)$
(Nacson, 2019)	NGD	$\tilde{\mathcal{O}}\left(\frac{1}{\sqrt{t}}\right)$
(Ji, 2021)	NGD	$\tilde{\mathcal{O}}\left(\frac{1}{t}\right)$
(Ji, 2022)	Dual Momentum GD	$\tilde{\mathcal{O}}\left(\frac{1}{t^2}\right)$ **(SOTA)**

[7] Mingze Wang. "Maximize Margin Nearly Exponentially Fast by First-order Optimization Method". In: *preparing* (2023).

Faster Margin Maximization for Logistic Regression

Theory

Theorem (Soudry, 2018)

For GD starting form any $\boldsymbol{w}(0) \in \mathbb{R}^d$ with constant η, we have $\gamma^* - \gamma(\boldsymbol{w}(t)) = \tilde{\mathcal{O}}\left(\frac{1}{\log t}\right)$.

Work	Algorithm	Convergence Rate of Margin Maximization
(Soudry, 2018)	GD	$\tilde{\mathcal{O}}\left(\frac{1}{\log t}\right)$
(Nacson, 2019)	NGD	$\tilde{\mathcal{O}}\left(\frac{1}{\sqrt{t}}\right)$
(Ji, 2021)	NGD	$\tilde{\mathcal{O}}\left(\frac{1}{t}\right)$
(Ji, 2022)	Dual Momentum GD	$\tilde{\mathcal{O}}\left(\frac{1}{t^2}\right)$ **(SOTA)**

Can we achieve margin maximization beyond polynomial rate by first-order methods?

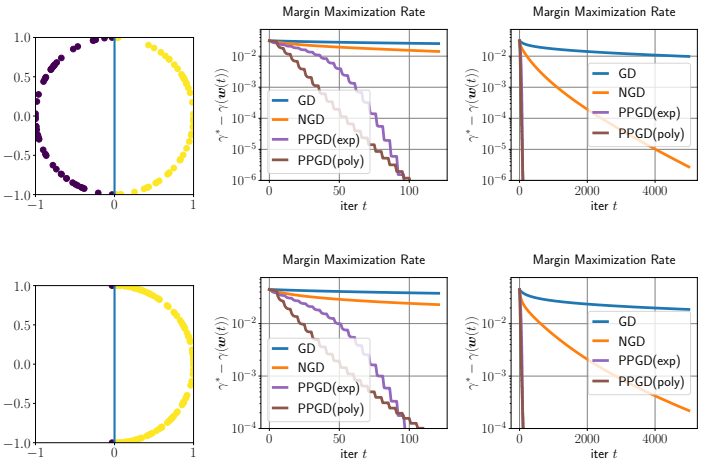
Work	Algorithm	Convergence Rate of Margin Maximization
This Work[7]	**Progressive Projected GD**	$\tilde{\mathcal{O}}\left(\frac{1}{C^t}\right)$??

[7] Mingze Wang. "Maximize Margin Nearly Exponentially Fast by First-order Optimization Method". In: *preparing* (2023).

Faster Margin Maximization for Logistic Regression

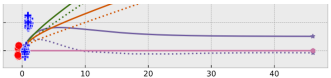
Experiments

Choose the same $\eta \equiv 1$ in these algorithms.



Faster Margin Maximization for Logistic Regression

Key Observation.



- Homogeneity: $\boldsymbol{w}$ and $c\boldsymbol{w}$ $(c > 0)$ have the same margin $\gamma(c\boldsymbol{w}) = \gamma(\boldsymbol{w})$.
- The convexity of the problem is stronger where $\|\boldsymbol{w}\|$ is small.
- The directional OPT at $100\boldsymbol{w}$ is more efficient than $\boldsymbol{w}$: smaller $\arg \langle -\nabla \mathcal{L}(\cdot), \boldsymbol{w}^* \rangle$.
- Regularized path $\boldsymbol{w}^*_{\text{reg}}(B)$ with larger B is closer to the max-margin direction.
- Regularized path $\boldsymbol{w}^*_{\text{reg}}(B) := \underset{\|\boldsymbol{w}\|_2 \leq B}{\arg\min}\, \mathcal{L}(\boldsymbol{w})$ is closer to $\boldsymbol{w}^*$ than NGD path in Figure in (Ji, 2020)

The intuition of acceleration.

- $\|\boldsymbol{w}\|$ should be stretched to ∞ to ensure (i) correct directional convergence to $\boldsymbol{w}^*$, (ii) more efficient directional opt.
- With small $\|\boldsymbol{w}\|$, the local opt is faster due to stronger convexity;
- Small-Large norm trade-off.

Acknowledgement and Main References

Mentor: **Weinan E**, Professor in Peking University and Princeton University

Collaborators: **Chao Ma**, Szegö Assistant Professor in Stanford University

Lei Wu, Assistant Professor in Peking University

Weijie J. Su, Associate Professor in Wharton School of the University of Pennsylvania

Acknowledgement and Main References

Mentor: **Weinan E**, Professor in Peking University and Princeton University

Collaborators: **Chao Ma**, Szegö Assistant Professor in Stanford University

Lei Wu, Assistant Professor in Peking University

Weijie J. Su, Associate Professor in Wharton School of the University of Pennsylvania

Mingze Wang, Chao Ma. "Early Stage Convergence and Global Convergence of Training Mildly Parameterized Neural Networks", NeurIPS 2022.

Lei Wu, **Mingze Wang**, Weijie J. Su. "The alignment property of SGD noise and how it helps select flat minima: A stability analysis", NeurIPS 2022.

Mingze Wang, Chao Ma. "Understanding Multi-phase Optimization Dynamics and Rich Nonlinear Behaviors of ReLU Networks", arXiv 2023.05, under review.

Mingze Wang, Lei Wu. "The Noise Geometry of Stochastic Gradient Descent: A Quantitative and Analytical Characterization", 2023.05, under review.

Mingze Wang. "Maximize Margin Nearly Exponentially Fast by First-order Optimization Method", preparing.

Thanks!

Q&A

WeChat: wmz931303659wmz

Email: mingzewang@stu.pku.edu.cn

Homepage: https://wmz9.github.io/