# SGD 类优化算法下批处理大小与学习率之间的缩放规律

朱梦

初稿于 2025 年 6 月 28 日，修改于 2025-06-29

## 1.　问题定义及其分析

当批处理大小（Batch Size）增大时，各种超参数尤其是学习率（Learning Rate）应该如何调整，才能保持原本的训练效果并最大化训练效率呢？这称为 Batch Size 和学习率之间的缩放规律（Scaling Law）。

## 2.　问题解决

### 2.1　单调有界

参考 OpenAI 的经典工作《An Empirical Model for Large-Batch Training》，它通过损失函数的二阶近似来分析 SGD 的最优学习率，得出“学习率随着批处理大小的增加二单调递增但有上界”的结论。整个推导过程最关键的思想是将学习率也视作优化参数：设待优化函数为 $f(\boldsymbol{\theta})$，当前批处理大小的梯度为 $\boldsymbol{g}_{\mathcal{R}}$，那么 SGD 后的损失函数为 $f(\boldsymbol{\theta}-\eta\boldsymbol{g}_{\mathcal{R}})$，将最优学习率的求解视为优化问题：

$$\eta^* = \arg\min_{\eta} \mathbb{E}[f(\boldsymbol{\theta}-\eta\boldsymbol{g}_{\mathcal{R}})] \tag{1}$$

式(1)表示，选择学习率使得平均而言损失值下降最快。由泰勒级数得：

$$\begin{aligned} f(\boldsymbol{\theta}-\eta\boldsymbol{g}_{\mathcal{R}}) &\approx f(\boldsymbol{\theta}) - \eta < \boldsymbol{g}_{\mathcal{R}}, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) > + \frac{1}{2}\eta^2 < \boldsymbol{g}_{\mathcal{R}}, \boldsymbol{\mathcal{H}}\boldsymbol{g}_{\mathcal{R}} > \\ &\triangleq f(\boldsymbol{\theta}) - \eta < \boldsymbol{g}_{\mathcal{R}}, \boldsymbol{g} > + \frac{1}{2}\eta^2 < \boldsymbol{g}_{\mathcal{R}}, \boldsymbol{\mathcal{H}}\boldsymbol{g}_{\mathcal{R}} > \end{aligned} \tag{2}$$

其中，$\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$ 表示函数 $f(.)$ 在点 $\boldsymbol{\theta}$ 处的梯度，即一阶偏导数向量。$\boldsymbol{\mathcal{H}}$ 表示函数 $f(.)$ 在点 $\boldsymbol{\theta}$ 处的 Hessian 矩阵，即二阶偏导数矩阵。接着求期望：

$$\begin{aligned} \mathbb{E}[f(\boldsymbol{\theta}-\eta\boldsymbol{g}_{\mathcal{R}})] &\approx \mathbb{E}[f(\boldsymbol{\theta}) - \eta < \boldsymbol{g}_{\mathcal{R}}, \boldsymbol{g} > + \frac{1}{2}\eta^2 < \boldsymbol{g}_{\mathcal{R}}, \boldsymbol{\mathcal{H}}\boldsymbol{g}_{\mathcal{R}} >] \\ &= \mathbb{E}[f(\boldsymbol{\theta})] - \eta\mathbb{E}[< \boldsymbol{g}_{\mathcal{R}}, \boldsymbol{g} >] + \frac{1}{2}\eta^2\mathbb{E}[< \boldsymbol{g}_{\mathcal{R}}, \boldsymbol{\mathcal{H}}\boldsymbol{g}_{\mathcal{R}} >] \\ &= f(\boldsymbol{\theta}) - \eta||\boldsymbol{g}||_2^2 + < \boldsymbol{g}, \boldsymbol{\mathcal{H}}\boldsymbol{g} > + \frac{\mathrm{Tr}(\boldsymbol{\Sigma}\boldsymbol{\mathcal{H}})}{B} \end{aligned} \tag{3}$$

**假设 1** $\mathcal{H}$ 为正定方阵。

基于假设1，那么问题就变成了二次函数的最小值，解得：

$$
\begin{aligned}
\eta^* &\approx \frac{||\boldsymbol{g}||_2^2}{<\boldsymbol{g}, \boldsymbol{\mathcal{H}}\boldsymbol{g}> + \frac{\mathrm{Tr}(\boldsymbol{\Sigma}\boldsymbol{\mathcal{H}})}{B}} = \frac{\eta_{\max}}{1 + \mathcal{B}_{\text{noise}/B}}, \\
\eta_{\max} &= \frac{||\boldsymbol{g}||_2^2}{<\boldsymbol{g}, \boldsymbol{\mathcal{H}}\boldsymbol{g}>},\ \mathcal{B}_{\text{noise}} = \frac{\mathrm{Tr}(\boldsymbol{\Sigma}\boldsymbol{\mathcal{H}})}{<\boldsymbol{g}, \boldsymbol{\mathcal{H}}\boldsymbol{g}>}
\end{aligned}
\tag{4}
$$

当 $B \ll \mathcal{B}_{\text{noise}}$ 时，$1 + \frac{\mathcal{B}_{\text{noise}}}{B} \approx \frac{\mathcal{B}_{\text{noise}}}{B}$，所以 $\eta^* \approx \frac{\eta_{\max} B}{\mathcal{B}_{\text{noise}}} \propto B$，即线性缩放；当 $B > \mathcal{B}_{\text{noise}}$ 时，$\eta* \to \eta_{\max}$，这意味批处理大小增加的成本远大于训练效率的提升。所以，$\mathcal{B}_{\text{noise}}$ 相当于一个分水岭，当批处理大小超过数值时，就没有继续投入算力增大批处理大小了。

## 2.2 实践分析

**假设 2** $\mathcal{H}$ 近似单位方阵的若干倍。

基于假设(2)，有：

$$
\mathcal{B}_{\text{noise}} \approx \frac{\mathrm{Tr}(\boldsymbol{\Sigma})}{||\boldsymbol{g}||_2^2} \triangleq \mathcal{B}_{\text{simple}} \tag{5}
$$

$\mathcal{B}_{\text{simple}}$ 在计算更为可行，且实验发现也为 $\mathcal{B}_{\text{noise}}$ 的一个良好近似，因此选择估计 $\mathcal{B}_{\text{simple}}$ 而不是 $\mathcal{B}_{\text{noise}}$。算法1展示了如何估算 $\mathcal{B}_{\text{simple}}$。需要注意的是每一轮 epoch 得到的 $\mathcal{B}_{\text{simple}}$ 是变化的，所以如果希望得到一个静态的规律，需要持续训练几轮 epoch，等到模型的训练进入“正轨”后计算的 $\mathcal{B}_{\text{simple}}$ 才可靠的，或者也可以在训练过程中持续监控 $\mathcal{B}_{\text{simple}}$，从而辅助更好地估算最优 $\mathcal{B}_{\text{simple}}$。

算法 1 $\mathcal{B}_{\text{simple}}$ 的估算算法

**输入:** 训练样本集 $\boldsymbol{\mathcal{D}}$，缓存梯度 $\bar{\boldsymbol{g}}_l$，缓存范数值 $\bar{\mathrm{L}}_{2,l}$。
**输出:** 估计的 $\mathcal{B}_{\text{simple}}$。

固定模型参数：选择一个固定的模型参数点 $\boldsymbol{\theta}_l$;
**对于** $t = 1$ **到** $T$ **执行**
  随机抽取训练样本子集：$\boldsymbol{\mathcal{R}}_t \subseteq \boldsymbol{\mathcal{D}}$;
  梯度计算：$\boldsymbol{g}_{l,t} = \nabla_{\boldsymbol{\theta}_{l,t-1}} \mathcal{L}$;
  梯度累积：$\bar{\boldsymbol{g}}_{l,t} = \bar{\boldsymbol{g}}_{l,t-1} + \frac{\boldsymbol{g}_{l,t}}{T}$;
  范数累积：$\bar{\mathrm{L}}_{2,l,t} = \bar{\mathrm{L}}_{2,l,t-1} + \frac{||\boldsymbol{g}_{l,t}||_2^2}{T-1}$;
计算估计的 $\mathrm{Tr}(\boldsymbol{\Sigma})$：$\mathrm{Tr}(\boldsymbol{\Sigma}) \approx \bar{\mathrm{L}}_{2,l,t} - \frac{T}{T-1}||\bar{\boldsymbol{g}}_{l,t}||_2^2$;
计算估计的 $||\boldsymbol{g}||_2^2$：$||\boldsymbol{g}||_2^2 \approx ||\bar{\boldsymbol{g}}_{l,t}||_2^2$;
计算估计的 $\mathcal{B}_{\text{simple}}$。

### 2.3 数据效率

从上述结果出发，还可以推导训练数据量和训练步数之间的渐进关系。将式(4)代入待优化函数中可以算得，在最优学习率下每一步迭代带来的损失值减少量为：

$$
\begin{aligned}
\Delta\mathcal{L} = f(\boldsymbol{\theta}) - \mathbb{E}[f(\boldsymbol{\theta} - \eta^* \boldsymbol{g}_{\mathcal{R}})] \approx \frac{(\Delta\mathcal{L})_{\text{max}}}{1 + \mathcal{B}_{\text{noise}}/B}, \\
(\Delta\mathcal{L})_{\text{max}} = \frac{||\boldsymbol{g}||_2^2}{2 < \boldsymbol{g}, \boldsymbol{\mathcal{H}}\boldsymbol{g} >}
\end{aligned}
\tag{6}
$$

当 $B \to \infty$ 也就是全量 SGD，每一步损失值减少量达到了最大值 $(\Delta\mathcal{L})_{\text{max}}$，此时可以用最少的训练步数（记为 $S_{\text{min}}$）到达目标点。当 $B$ 有限时，每一步的损失值下降量平均只有 $\Delta\mathcal{L}$，这意味着需要 $1 + \mathcal{B}_{\text{noise}}/B$ 步才能到达全量 SGD 单步的下降量，所以训练的总步数大约为 $S = (1 + \mathcal{B}_{\text{noise}}/B)S_{\text{min}}$。

由于批处理大小为 $B$ 时，训练过程消耗的样本总数为 $N = BS = (B + \mathcal{B}_{\text{noise}})S_{\text{min}}$，这是 $B$ 的增函数。当 $B \to 0$ 时，使得 $N_{\text{min}} = \mathcal{B}_{\text{noise}}S_{\text{min}}$，这表明只要足够小的批处理大小去训练模型，那么所需的训练样本总数 $N$ 也会相应地减少，代价是训练步数 $S$ 增多。综合起来，这些符号的关系为：

$$
\left(\frac{S}{S_{\text{min}}} - 1\right)\left(\frac{N}{N_{\text{min}}} - 1\right) = 1 \tag{7}
$$

式(7)为训练数据量和训练步数之间的缩放规律，表示数据量越小，训练步数更多，即缩小批处理大小，才能更大可能达到最优解。

## 3. 实验结果及其分析

## 4. 结论及其反思