

8 数值稳定性，激活函数和硬件

概要

- 数值稳定性
 - 梯度爆炸
 - 梯度消失
- 稳定模型训练
 - 权重初始化
- 激活函数

数值稳定性

神经网络的梯度

➤ 考虑具有d层的神经网络

$$\mathbf{h}^t = f_t(\mathbf{h}^{t-1})$$

$$y = \ell \circ f_d \circ \cdots \circ f_1(\mathbf{x})$$

➤ 计算损失 ℓ 的梯度 $\mathbf{W}_t$

$$\frac{\partial \ell}{\partial \mathbf{W}^t} = \frac{\partial \ell}{\partial \mathbf{h}^d} \frac{\partial \mathbf{h}^d}{\partial \mathbf{h}^{d-1}} \cdots \frac{\partial \mathbf{h}^{t+1}}{\partial \mathbf{h}^t} \frac{\partial \mathbf{h}^t}{\partial \mathbf{W}^t}$$

$d-t$ 矩阵相乘

深度神经网络的两个问题

➤ 本质上：连乘的数值计算，带来的累计误差。可能很小，也可能很大

梯度爆炸



$$1.5^{100} \approx 4 \times 10^{17}$$

梯度消失



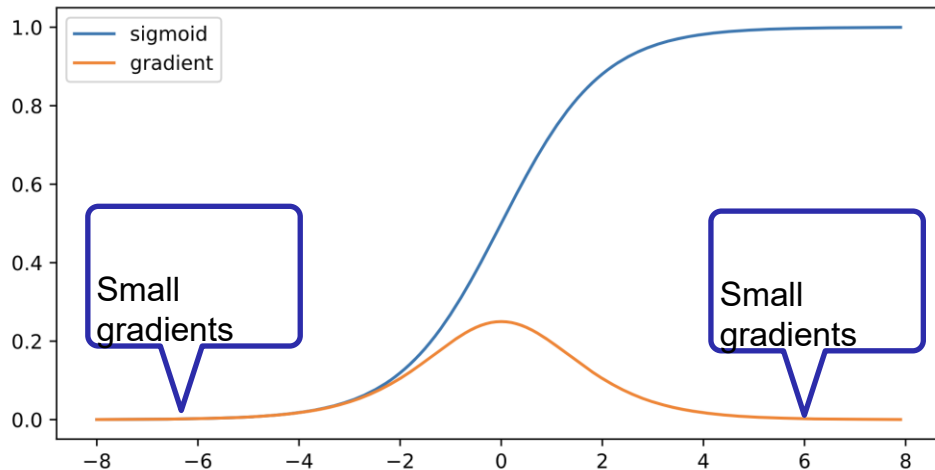
$$0.8^{100} \approx 2 \times 10^{-10}$$

$$\prod_{i=t}^{d-1} \frac{\partial \mathbf{h}^{i+1}}{\partial \mathbf{h}^i}$$

梯度消失

➤ 使用sigmoid作为激活函数

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad \sigma'(x) = \sigma(x)(1 - \sigma(x))$$



梯度消失

- 使用sigmoid作为激活函数

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \sigma'(x) = \sigma(x)(1 - \sigma(x))$$

- $\prod_{i=t}^{d-1} \frac{\partial \mathbf{h}^{i+1}}{\partial \mathbf{h}^i} = \prod_{i=t}^{d-1} \text{diag}(\sigma'(\mathbf{W}^i \mathbf{h}^{i-1}))(W^i)^T$ 是 t 个较小值的乘积

- 例如 $0.8^{100} \approx 2 \times 10^{-10}$

梯度消失的问题

- 梯度值趋近为0的渐变
 - 16位浮点（梯度值小于 $2^{-24} \approx 5.96 \times 10^{-8}$ 即为0）
- 训练没有进展
 - 无论如何选择学习率（LR）
- 底层训练基本无效
 - 只有顶层训练有效
 - 使网络更深可能并没有更好

梯度爆炸

➤ 例如，使用 ReLU 作为激活函数

$$\sigma(x) = \max(0, x) \quad \rightarrow \quad \sigma'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

➤ $\prod_{i=t}^{d-1} (W^i)^T$ 为来自 $\prod_{i=t}^{d-1} \frac{\partial \mathbf{h}^{i+1}}{\partial \mathbf{h}^i} = \prod_{i=t}^{d-1} \text{diag}(\sigma'(\mathbf{W}^i \mathbf{h}^{i-1})) (W^i)^T$ 的元素

➤ 当 d_t 很大时，可产生很大的值 $1.5^{100} \approx 4 \times 10^{17}$

梯度爆炸的问题

- 梯度值超出范围：无穷大值
 - 严重的使用 16 位浮点
 - 范围：[6e-5, 6e4]
- 对学习率（LR）敏感
 - 不够小的LR -> 更大的权重 -> 更大的梯度
 - 太小的LR -> 模型训练没有进展
 - 可能需要在训练期间大幅改变LR

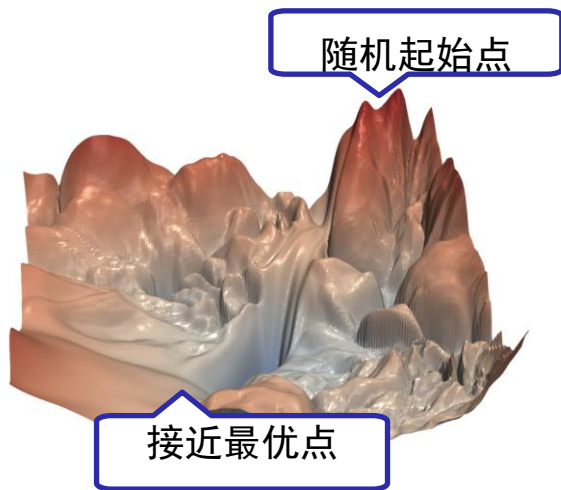
稳定模型训练

稳定模型训练

- 目标：确保渐变值在适当的范围内
 - 例如 在 $[1e-6, 1e3]$ 之间
- 改变神经网络框架结构（连续的乘法 -> 加）
 - ResNet, LSTM
- 归一化
 - 批量归一化，渐变修剪
- 适当的权重初始化和激活函数

权重初始化

- 使用适当范围内的随机值初始化权重
- 训练的开始容易受到数值不稳定性的影响
 - 远离最优点的表面可能很复杂
 - 接近最优点的表面可能更平坦
- 权重根据 $\mathcal{N}(0, 0.01)$ 初始化，对小网络很有效，但不保证对深度神经网络也有效



每层神经网络的常数方差

- 将每层的输出和梯度看成随机变量
- 使每层的输出的均值和方差相同，类似于梯度

正向

$$\begin{aligned}\mathbb{E}[h_i^t] &= 0 \\ \text{Var}[h_i^t] &= a\end{aligned}$$

反向

$$\mathbb{E}\left[\frac{\partial \ell}{\partial h_i^t}\right] = 0 \quad \text{Var}\left[\frac{\partial \ell}{\partial h_i^t}\right] = b \quad \forall i, t$$

a 和 b 都是常数

例子：MLP

➤ 假设：

➤ i.i.d $w_{i,j}^t, \mathbb{E}[w_{i,j}^t] = 0, \text{Var}[w_{i,j}^t] = \gamma_t$

➤ h_i^{t-1} 与 $w_{i,j}^t$ 是独立的

➤ 激活：用 $\mathbf{h}^t = \mathbf{W}^t \mathbf{h}^{t-1}$, $\mathbf{W}^t \in \mathbb{R}^{n_t \times n_{t-1}}$

$$\mathbb{E}[h_i^t] = \mathbb{E} \left[\sum_j w_{i,j}^t h_j^{t-1} \right]_0^0 = \sum_j \mathbb{E}[w_{i,j}^t] \mathbb{E}[h_j^{t-1}] = 0$$

正向方差

$$\begin{aligned}\text{Var}[h_i^t] &= \mathbb{E}[(h_i^t)^2] - \mathbb{E}[h_i^t]^2 = \mathbb{E}[(\sum_j w_{i,j}^t h_j^{t-1})^2] \\ &= \mathbb{E}[\sum_j (w_{i,j}^t)^2 (h_j^{t-1})^2 + \sum_{j \neq k} w_{i,j}^t w_{i,k}^t h_j^{t-1} h_k^{t-1}] \\ &= \sum_j \mathbb{E}[(w_{i,j}^t)^2] \mathbb{E}[(h_j^{t-1})^2] \quad \Rightarrow \quad \eta_{t-1} \gamma_t = 1 \\ &= \sum_j \text{Var}[w_{i,j}^t] \text{Var}[h_j^{t-1}] = n_{t-1} \gamma_t \text{Var}[h_j^{t-1}]\end{aligned}$$

反向均值和方差

➤ 同正向分析：

$$\frac{\partial \ell}{\partial \mathbf{h}^{t-1}} = \frac{\partial \ell}{\partial \mathbf{h}^t} \mathbf{W}^t \quad \Rightarrow \quad \left(\frac{\partial \ell}{\partial \mathbf{h}^{t-1}} \right)^T = (W^t)^T \left(\frac{\partial \ell}{\partial \mathbf{h}^t} \right)^T$$

$$\mathbb{E} \left[\frac{\partial \ell}{\partial h_i^{t-1}} \right] = 0$$

$$\text{Var} \left[\frac{\partial \ell}{\partial h_i^{t-1}} \right] = n_t \gamma_t \text{Var} \left[\frac{\partial \ell}{\partial h_j^t} \right] \quad \Rightarrow \quad n_t \gamma_t = 1$$

Xavier 初始化

- Xavier 初始化（作者 Xavier Glorot），其核心思想是：
 - 保持每一层激活值的方差和反向传播梯度的方差，在层与层之间保持一致。
- 权重的初始化值，根据该层的输入神经元数量 (fan_in) 和输出神经元数量 (fan_out) 来动态调
 - a) Xavier 均匀分布 (Xavier Uniform)
 - 从一个均匀分布 $U[-limit, limit]$ 中采样，其中 $limit$ 的计算公式为：
 - $limit = \sqrt{6 / (fan_in + fan_out)}$
 - b) Xavier 正态分布 (Xavier Normal)
 - 从一个均值为 0，标准差为 std 的正态分布中采样，其中 std 的计算公式为：
 - $std = \sqrt{2 / (fan_in + fan_out)}$
 - 这里的关键参数：
 - fan_in : 权重矩阵的输入维度，即上一层的神经元数量。
 - fan_out : 权重矩阵的输出维度，即当前层的神经元数量。
 - 层的输入和输出神经元越多，初始化的权重值就应该越小，以防止信号的方差被放大

He 初始化

➤ Xavier 初始化

- 推导过程有一个重要假设：激活函数是线性的，并且关于原点对称。
- 最常用的激活函数是 ReLU (Rectified Linear Unit) 及其变体。ReLU 函数 $f(x) = \max(0, x)$ 并不是线性的，它会将所有负的输入都置为0。
- 这个“置零”操作会导致输出方差大约减半。Xavier 初始化导致信号的方差会逐层递减，最终消失。

➤ Kaiming Initialization

- 考虑到 ReLU 会使一半的输入变为0，为了保持方差不变，初始化的权重值应该相应地放大一些。
- 公式（正态分布）： $\text{std} = \sqrt{2 / \text{fan_in}}$ （注意：这里只考虑了 fan_in，因为实践中发现它效果更好且更简单）
- He 初始化的方差是 Xavier 的两倍左右（当 $\text{fan_in} \approx \text{fan_out}$ 时），正好弥补了 ReLU 带来的方差损失。

激活函数

简单的线性激活函数

➤ 假设 $\sigma(x) = \alpha x + \beta$ $\mathbf{h}' = \mathbf{W}^t \mathbf{h}^{t-1}$ and $\mathbf{h}^t = \sigma(\mathbf{h}')$

➤ $\mathbb{E}[h_i^t] = \mathbb{E}[\alpha h_i' + \beta] = \beta \implies \beta = 0$

$$\begin{aligned} \text{Var}[h_i^t] &= \mathbb{E}[(h_i^t)^2] - \mathbb{E}[h_i^t]^2 \\ &= \mathbb{E}[(\alpha h_i' + \beta)^2] - \beta^2 \\ &= \mathbb{E}[\alpha^2 (h_i')^2 + 2\alpha\beta h_i' + \beta^2] - \beta^2 \\ &= \alpha^2 \text{Var}[h_i'] \end{aligned} \implies \alpha = 1$$

反向

➤ 假设

$$\sigma(x) = \alpha x + \beta, \quad \frac{\partial \ell}{\partial \mathbf{h}'} = \frac{\partial \ell}{\partial \mathbf{h}^t} (W^t)^T \quad \text{and} \quad \frac{\partial \ell}{\partial \mathbf{h}^{t-1}} = \alpha \frac{\partial \ell}{\partial \mathbf{h}'}$$

$$\text{➤ } \mathbb{E} \left[\frac{\partial \ell}{\partial h_i^{t-1}} \right] = 0 \implies \beta = 0$$

$$\text{➤ } \text{Var} \left[\frac{\partial \ell}{\partial h_i^{t-1}} \right] = \alpha^2 \text{Var} \left[\frac{\partial \ell}{\partial h'_j} \right] \implies \alpha = 1$$

总结

- 数值稳定性
 - 梯度爆炸
 - 梯度消失
- 稳定模型训练
 - 权重初始化
- 激活函数