

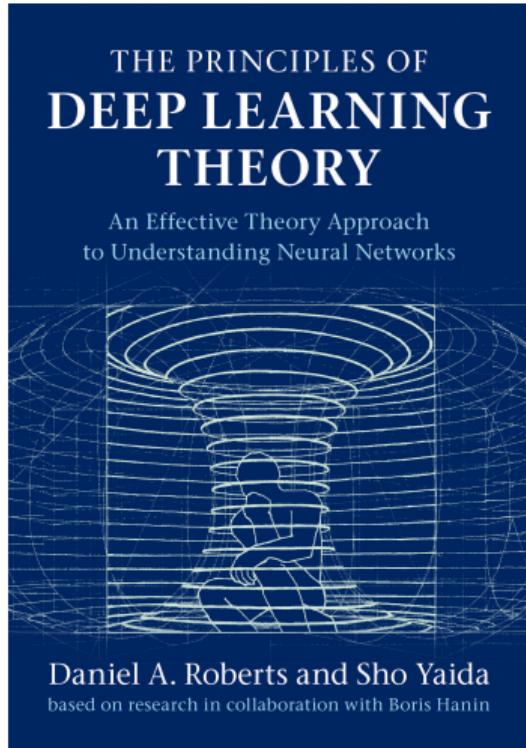
无限宽神经网络：初始化，激活函数与优化

The Principles of Deep Learning Theory

陈千

qianchen02@hust.edu.cn

简介



- 随着芯片技术的发展，我们可以训练越来越大的深度学习模型，其性能表现也越来越好。然而，深度学习仍然基于实验，能够像牛顿运动定律一样指导实践的理论并没有被发现。
- 这本书从第一性原理出发，运用数学和物理的工具，对多层感知机进行了详尽的分析，并且这种分析可以扩展到 CNN, ResNet, Transformer 等。
- 我们聚焦在无限宽多层感知机的情况，分析其特点。
 - ▶ 对前向传播，求解每一层神经元服从的概率分布，并分析激活函数的影响和后验概率（即 Bayes 学习）。
 - ▶ 对反向传播，求解 NTK，并分析多层感知机的优化过程与特点，并求出最优解。

多层感知机

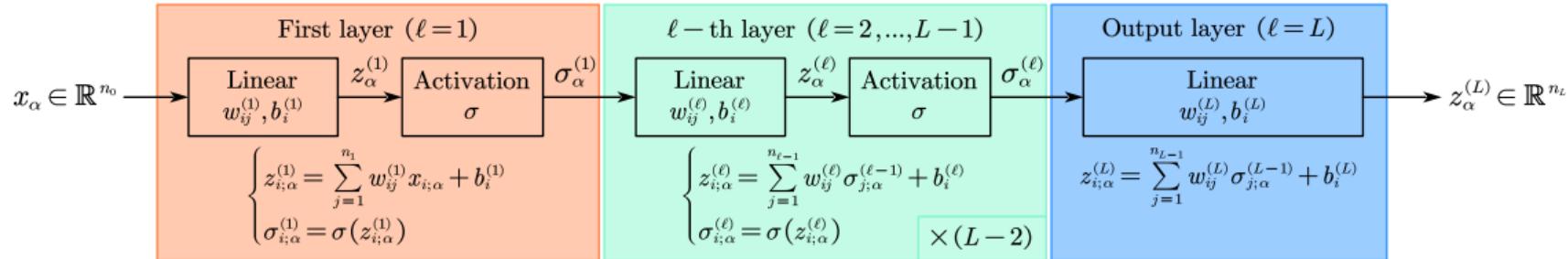


图: 多层感知机的结构。无限宽指 $n_1, n_2, \dots, n_{L-1} \rightarrow \infty$ 。

- 网络结构: n_ℓ 表示第 ℓ 层的宽度, 激活函数记作 σ 。网络的输入 $x_\alpha \in \mathbb{R}^{n_0}$, 对应的第 ℓ 线性层的输出为 $z_\alpha^{(\ell)} \in \mathbb{R}^{n_\ell}$, 经过激活函数后为 $\sigma_\alpha^{(\ell)} \in \mathbb{R}^{n_\ell}$ 。
- 数据集: $\mathcal{D} = \{(x_\alpha, y_\alpha)\}_{\alpha=1}^{N_{\mathcal{D}}}$, 其中样本 $x_\alpha \in \mathbb{R}^{n_0}$, 标签 $y_\alpha \in \mathbb{R}^{n_L}$ 。
- 初始化: $w_{ij}^{(\ell)} \sim \text{i.i.d.} \mathcal{N}\left(0, \frac{C_w^{(\ell)}}{n_{\ell-1}}\right)$, $b_i^{(\ell)} \sim \text{i.i.d.} \mathcal{N}\left(0, C_b^{(\ell)}\right)$ 。

前向传播

考虑到多层感知机的参数是随机初始化的，为了分析前向传播的过程，必须求出任意层的输出服从的概率分布，即 $p(z_{i_1;\alpha_1}^{(\ell)}, z_{i_2;\alpha_2}^{(\ell)}, \dots, z_{i_m;\alpha_m}^{(\ell)})$ 。对此，我们有如下定理。

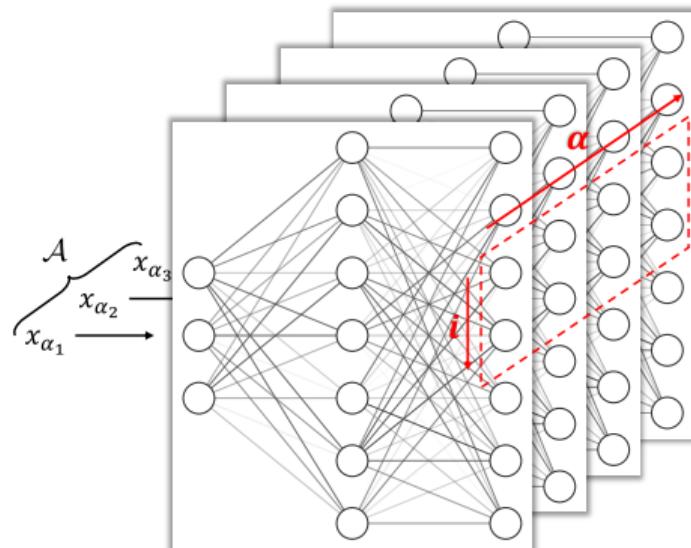


图: 前向传播

定理

对无限宽多层感知机，设 $I \subset \{1, 2, \dots, n_\ell\}$ 是第 ℓ 层神经元的一个有限子集， $\mathcal{A} \subset \{1, 2, \dots, N_D\}$ 是输入的子集，那么

$$z_{I;\mathcal{A}}^{(\ell)} \sim \mathcal{N}(0, \delta_{ij} K_{\alpha\beta}^{(\ell)}) \quad (1)$$

其中 $K_{\alpha\beta}^{(\ell)} = \mathbb{E} z_{i;\alpha}^{(\ell)} z_{i;\beta}^{(\ell)}$ 。

- 不同神经元的输出独立。
- 不同样本的同一神经元输出相关。

激活函数

通过分析 $K_{\alpha\beta}^{(\ell+1)}$, 可以得到递推公式:

$$K_{\alpha\beta}^{(\ell+1)} = C_w^{(\ell+1)} \langle \sigma(z_\alpha) \sigma(z_\beta) \rangle_{K_{\alpha\beta}^{(\ell)}} + C_b^{(\ell+1)}, \ell = 1, 2, \dots, L-1 \quad (2)$$

$$K_{\alpha\beta}^{(1)} = C_w^{(1)} \left(\frac{1}{n_0} \sum_{i=1}^{n_0} x_{i;\alpha} x_{i;\beta} \right) + C_b^{(1)} \quad (3)$$

要将输入有效的传递到更深的层中, 必须保证 $K_{\alpha\beta}^{(\ell)}$ 不会以指数速度收敛或发散, 并称满足这一条件的初始化方式为**临界初始化**。设 $K^{(\ell)} \rightarrow K^*, \ell \rightarrow \infty$ 。

- 尺度不变类 (ReLU, LeakyReLU): $(C_b, C_w) = \left(0, \frac{2}{[\sigma'_+(0)]^2 + [\sigma'_-(0)]^2}\right)$ (He Initialization)
- $K^* = 0$ ($K_{\alpha\beta}^{(\ell)} = \mathcal{O}(1/\ell)$) 普适类 (tanh): $(C_b, C_w) = \left(0, \frac{1}{[\sigma'(0)]^2}\right)$ (Xavier Initialization)
- 半稳定普适类 (SWISH, GELU): 存在不稳定的不动点。
- 非临界类 (sigmoid, softplus): 无临界初始化方式。

Bayes 学习

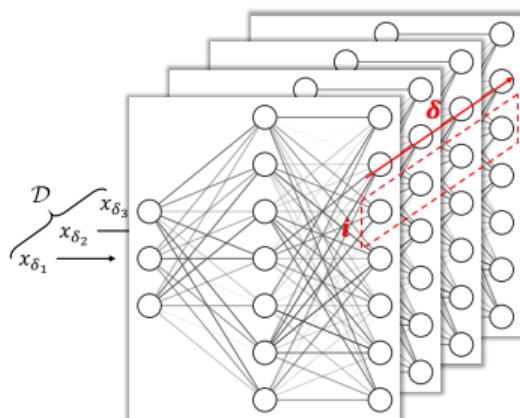
Bayes 学习考虑在已知训练集输出正确，即 $z_{i;\mathcal{A}}^{(L)} = y_{i;\mathcal{A}}$ 的条件下，神经网络在测试集上的输出 $z_{i;\mathcal{B}}^{(L)}$ 服从的条件分布。设训练集 \mathcal{A} 和测试集 \mathcal{B} 是数据集 \mathcal{D} 的一个划分。由于输出层神经元 $z_{i;\mathcal{D}}^{(L)} \sim \text{i.i.d.} \mathcal{N}(0, K_{\alpha\beta}^{(\ell)})$ ，因而待求分布为高斯分布的后验，即：

$$p(z_{i;\mathcal{B}}^{(L)} \mid z_{i;\mathcal{A}}^{(L)} = y_{i;\mathcal{A}}) = \mathcal{N}(z_{i;\mathcal{B}}^{(L)}; m_{i;\beta}, \mathbb{K}_{\beta_1\beta_2}) \quad (4)$$

$$m_{i;\beta} = \sum_{\alpha_1, \alpha_2 \in \mathcal{A}} K_{\beta\alpha_1}^{(L)} \tilde{K}_{(L)}^{\alpha_1\alpha_2} y_{i;\alpha_2} \quad (5)$$

$$\mathbb{K}_{\beta_1\beta_2} = K_{\beta_1\beta_2}^{(L)} - \sum_{\alpha_1, \alpha_2 \in \mathcal{A}} K_{\beta_1\alpha_1}^{(L)} \tilde{K}_{(L)}^{\alpha_1\alpha_2} K_{\alpha_2\beta_2}^{(L)} \quad (6)$$

无限宽多层感知机可以看作高斯过程，其 Bayes 学习等同于高斯过程回归。



图：输出层服从独立的正态分布。

Bayes 学习

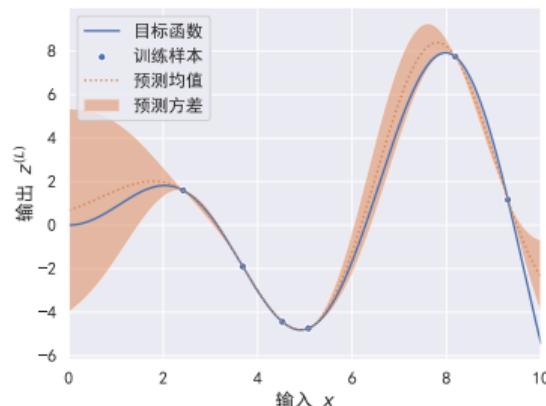
Bayes 学习考虑在已知训练集输出正确，即 $z_{i;\mathcal{A}}^{(L)} = y_{i;\mathcal{A}}$ 的条件下，神经网络在测试集上的输出 $z_{i;\mathcal{B}}^{(L)}$ 服从的条件分布。设训练集 \mathcal{A} 和测试集 \mathcal{B} 是数据集 \mathcal{D} 的一个划分。由于输出层神经元 $z_{i;\mathcal{D}}^{(L)} \sim \text{i.i.d.} \mathcal{N}(0, K_{\alpha\beta}^{(\ell)})$ ，因而待求分布为高斯分布的后验，即：

$$p(z_{i;\mathcal{B}}^{(L)} \mid z_{i;\mathcal{A}}^{(L)} = y_{i;\mathcal{A}}) = \mathcal{N}(z_{i;\mathcal{B}}^{(L)}; m_{i;\beta}, \mathbb{K}_{\beta_1\beta_2}) \quad (4)$$

$$m_{i;\beta} = \sum_{\alpha_1, \alpha_2 \in \mathcal{A}} K_{\beta\alpha_1}^{(L)} \tilde{K}_{(L)}^{\alpha_1\alpha_2} y_{i;\alpha_2} \quad (5)$$

$$\mathbb{K}_{\beta_1\beta_2} = K_{\beta_1\beta_2}^{(L)} - \sum_{\alpha_1, \alpha_2 \in \mathcal{A}} K_{\beta_1\alpha_1}^{(L)} \tilde{K}_{(L)}^{\alpha_1\alpha_2} K_{\alpha_2\beta_2}^{(L)} \quad (6)$$

无限宽多层感知机可以看作高斯过程，其 Bayes 学习等同于高斯过程回归。



图：无限宽多层感知机是高斯过程。

基于梯度的学习

通常我们使用基于梯度下降的方法优化参数 $\{\theta_\mu\}_{\mu=1}^P$, 为了分析学习率的影响, 我们引入学习率张量 $\lambda_{\mu\nu}$, 设全局学习率为 η , 损失函数为 $\mathcal{L}(z_A^{(L)}, y_A)$, 则梯度下降的参数更新为:

$$\Delta \theta_\mu = -\eta \sum_\nu \lambda_{\mu\nu} \frac{\partial \mathcal{L}}{\partial \theta_\nu} \quad (7)$$

由于参数发生变化, 输出层同样发生变化:

$$\begin{aligned} \Delta z_{i_1; \alpha_1}^{(L)} &= -\eta \sum_{\mu, \nu} \lambda_{\mu\nu} \frac{\partial z_{i_1; \alpha_1}^{(L)}}{\partial \theta_\mu} \frac{\partial \mathcal{L}}{\partial \theta_\nu} + \mathcal{O}(\eta^2) \\ &= -\eta \sum_{i_2, \alpha_2} \underbrace{\sum_{\mu, \nu} \lambda_{\mu\nu} \frac{\partial z_{i_1; \alpha_1}^{(L)}}{\partial \theta_\mu} \frac{\partial z_{i_2; \alpha_2}^{(L)}}{\partial \theta_\nu}}_{\text{NTK: } H_{i_1 i_2; \alpha_1 \alpha_2}} \frac{\partial \mathcal{L}}{\partial z_{i_2; \alpha_2}^{(L)}} + \mathcal{O}(\eta^2) \end{aligned} \quad (8)$$

核学习

实践中，我们对不同的参数取不同的学习率，我们设置第 ℓ 层的权重 $w_{ij}^{(\ell)}$ 的学习率为 $\frac{\lambda_w^{(\ell)}}{n_{\ell-1}}$ ，偏置 $b_i^{(\ell)}$ 的学习率为 $\lambda_b^{(\ell)}$ 。此时，对于无限宽多层感知机：

- $H_{i_1 i_2; \alpha_1 \alpha_2} \rightarrow \delta_{i_1 i_2} \Theta_{\alpha_1 \alpha_2}$
- 余项 $\mathcal{O}(\eta^2) \rightarrow 0$

利用以上性质，我们可以构造一个特殊的损失函数，使得一步梯度下降就可以到达最优点。具体来说，取全局学习率 $\eta = 1$ ，取训练集 \mathcal{A} 上的特殊的损失函数

$$\mathcal{L}_{\mathcal{A}} = \frac{1}{2} \sum_{i=1}^{n_L} \sum_{\alpha_1, \alpha_2 \in \mathcal{A}} \tilde{\Theta}^{\alpha_1 \alpha_2} \left(z_{i; \alpha_1}^{(L)} - y_{i; \alpha_1} \right) \left(z_{i; \alpha_2}^{(L)} - y_{i; \alpha_2} \right) \quad (9)$$

仅需一步梯度下降，可满足训练集 $z_{i; \alpha_1}^{(L)} (t=1) = y_{i; \alpha_1}$ 的最优条件：

$$\Delta z_{i; \delta}^{(L)} = - \sum_{\alpha_1, \alpha_2 \in \mathcal{A}} \Theta_{\delta \alpha_1} \underbrace{\tilde{\Theta}^{\alpha_1 \alpha_2} \left(z_{i; \alpha_2}^{(L)} - y_{i; \alpha_2} \right)}_{\partial \mathcal{L}_{\mathcal{A}} / \partial z_{i; \alpha_2}^{(L)}} \triangleq \mathcal{A}^* \left(z_{i; \mathcal{A}}^{(L)} - y_{i; \mathcal{A}} \right) \quad (10)$$

核学习

- 性质 1: $\mathcal{A}^* : \mathbb{R}^{n_L \times N_D} \rightarrow \mathbb{R}^{n_L \times N_D}$ 是线性变换，但只与偏差在训练集上的分量有关。
- 性质 2: 使用任意学习率 η 和任意损失函数 \mathcal{L} 执行梯度下降后，设输出的改变量设为 Δz ，则带入计算可知， $\mathcal{A}^*(\Delta z) = -\Delta z$ 。

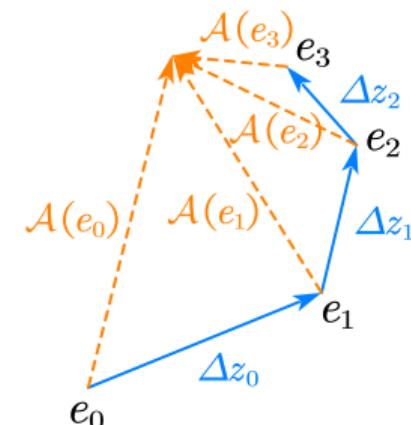
设 $e_0 = z_D^{(L)} - y_D$ 是初始的偏差， Δz 同上，那么经过这一步梯度下降后再执行 \mathcal{A}^* 所代表的梯度下降，得到的输出为

$$(e_0 + \Delta z) + \mathcal{A}^*(e_0 + \Delta z) = e_0 + \cancel{\Delta z} + \mathcal{A}^*(e_0) + \cancel{\mathcal{A}^*(\Delta z)} \quad (11)$$

等号的右边为在初始情况下执行 \mathcal{A}^* 所代表的梯度下降得到的输出。因此，无限宽多层感知机完全训练后的输出与每一步选择全局的学习率 η 和损失函数 \mathcal{L} 之间无关。进一步，最终输出为

$$z_{i;\delta}^{(L)}(t=T) = z_{i;\delta}^{(L)} - \sum_{\alpha_1, \alpha_2} \Theta_{\delta \alpha_1} \tilde{\Theta}^{\alpha_1 \alpha_2} \left(z_{i;\alpha_2}^{(L)} - y_{i;\alpha_2} \right) \quad (12)$$

与 Bayes 学习类似，最终输出 $z_{i;\delta}^{(L)}(t=T)$ 服从正态分布。

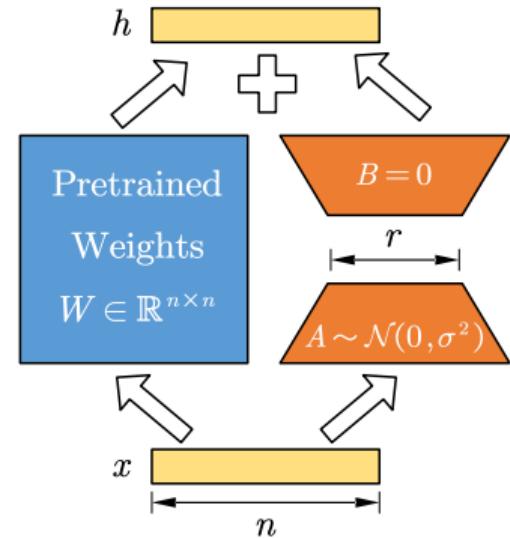


图：无限宽多层感知机最终输出与优化过程无关。

研究展望

本书对无限宽的多层感知机的初始化，激活函数与优化过程都做出了细致分析。同时，这种分析可以扩展到 CNN，ResNet [1]，Transformer [2] 等结构上。

- 本书中提到的初始化方式和学习率设置与一些研究一致。
例如，在 LoRA 提出的 $h = Wx + BAx$ 的相关改进中：
 1. rsLoRA [3]: $h = Wx + BAx/\sqrt{r}$, 相当于对 A 和 B 的初始化方差进行调整，与前述的参数初始化方式一致。
 2. LoRA+ [4]: $\eta_A = \Theta(1/n)$, $\eta_B = \Theta(1)$ 。这与前述的学习率设置方式一致。
- 研究价值：
 1. 研究其他网络结构如 UNet 等的参数初始化方式和学习率设置，指导超参数选择，从而提高性能。
 2. 研究输出层的概率分布与网络效果的关系，例如：Diffusion 模型预测添加的噪声时的效果更好 [5]，是否与神经网络初始化导致的输出层先验分布有关。



图：LoRA

谢谢！

参考文献

- [1] Daniel A. Roberts, Sho Yaida, and Boris Hanin. *The Principles of Deep Learning Theory*. Vol. 46. Cambridge University Press Cambridge, MA, USA, 2022.
- [2] Emily Dinan, Sho Yaida, and Susan Zhang. *Effective Theory of Transformers at Initialization*. Apr. 2023. arXiv: 2304.02034 [hep-th, stat].
- [3] Damjan Kalajdzievski. *A Rank Stabilization Scaling Factor for Fine-Tuning with LoRA*. Nov. 2023. arXiv: 2312.03732 [cs].
- [4] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. *LoRA+: Efficient Low Rank Adaptation of Large Models*. Feb. 2024. arXiv: 2402.12354 [cs, stat].
- [5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.