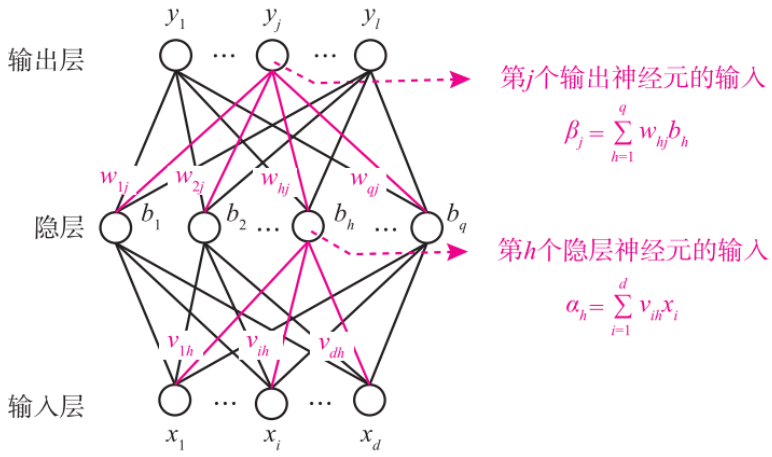
《人工智能导论》2025

智能科学与技术学院

第三次作业-神经网络与深度学习

1. 神经网络(Neural Network) [45pt]

给定训练集 $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$。其中，$x_i \in \mathbb{R}^d, y_i \in \mathbb{R}^l$，下图给出了一个有 d 个输入、l 个输出、q 个隐层神经元的多层神经网络,其中输出层第 j 个神经元的阈值用 θ_j 表示，隐层第 h 个神经元的阈值用 γ_{h} 表示。输入层第 i 个神经元与隐层第 h 个神经元之间的连接权为 $v_{i\mathrm{h}}$，隐层第 h 个神经元与输出层第 j 个神经元之间的连接权为 $w_{\mathrm{h}j}$。记隐层第 h 个神经元接收到的输入为 $\alpha_{\mathrm{h}}=\sum_{i=1}^{d} v_{i\mathrm{h}} x_i$，输出层第 j 个神经元接收到的输入为 $\beta_j=\sum_{\mathrm{h}=1}^{q} w_{\mathrm{h}j} b_{\mathrm{h}}$，其中 b_{h} 为隐层第 h 个神经元的输出。



不同任务中神经网络的输出层往往使用不同的激活函数和损失函数，本题介绍几种常见的激活函数和损失函数,并对其梯度进行推导，请给出具体的推导过程，而不是直接给一个结果。

（1）[10pt] 在二分类问题 $(l = 1)$ 中,标记 $y \in \{0,1\}$, 一般使用 Sigmoid 函数作为激活函数, 使输出值在 $[0,1]$ 范围内, 使模型预测结果可直接作为概率输出。 Sigmoid 函数的输出一般配合二元交叉熵损失函数使用, 对于一个训练样本 $(\boldsymbol{x}, y)$ 有

$$\ell(y, \hat{y}_1) = -[y\log(\hat{y}_1) + (1-y)\log(1-\hat{y}_1)].$$

记 $\hat{y}_1$ 为模型将样本判断为正例的预测概率, 请计算 $\frac{\partial \ell(y,\hat{y}_1)}{\partial \beta_1}$。

（2）[20pt] 当 $l > 1$ 时, 网络的预测结果为 $\hat{\boldsymbol{y}} \in \mathbb{R}^l$, 其中 $\hat{y}_i$ 表示输入被预测为 第 i 类的概率. 对于第 i 类的样本, 其标记 $\boldsymbol{y} \in \{0,1\}^l$, 有 $y_i = 1, y_j = 0, j \neq i$。对于一个训练样本 $(\boldsymbol{x}, \boldsymbol{y})$, 交叉熵失函数 $\ell(\boldsymbol{y}, \hat{\boldsymbol{y}})$ 的定义如下：

$$\ell(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{j=1}^{l} y_j \log \hat{y}_j$$

多分类问题一般使用 SoftMax 函数作为输出层激活函数, 计算公式为：

$$\hat{y}_j = \frac{e^{\beta_j}}{\sum_{k=1}^{l} e^{\beta_k}}.$$

易见 SoftMax 函数输出的 $\hat{\boldsymbol{y}}$ 符合 $\sum_{j=1}^{l} \hat{y}_j = 1$, 所以可以直接视为每个类别的概率。SoftMax 函数的输出一般配合交叉熵损失函数使用, 请计算 $\frac{\partial \ell(\boldsymbol{y},\hat{\boldsymbol{y}})}{\partial \beta}$。

（3）[5pt] 分析在二分类问题中使用的 SoftMax 激活函数和 Sigmoid 激活函数的联系与区别。

（4）[10pt] KL 散度定义了两个分布之间的差异性, 对于两个离散分布 $Q(x)$ 和 $P(x)$, 其定义为：

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right).$$

其中 $\mathcal{X}$ 为 x 的取值空间。试分析交叉熵损失函数和 KL 散度的关系。

2. 链式法则的矩阵形式 [30pt]

假设有两个隐层的神经网络(隐层神经元数目分别为 q_1 和 q_2)，记网络输入为 $X \in \mathbb{R}^{m \times d}$, 其中 m 为样本数, d 为特征维度. 样本标记为 $Y \in \mathbb{R}^{m \times l}$, 表示 m 个样本的标记维度为 l。 第 i 层的权重矩阵为 $W_i \in \mathbb{R}^{q_i \times q_{i+1}}$, 其中 $q_0 = d$， 偏移为 $b_i \in \mathbb{R}^{1 \times q_{i+1}}$。隐层激活函数为 $f(x) = \sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$，输出层无激活函数 (即使用线性激活函数)。第 i 层的输出表示为 X_i, 网络的输入为第 0 层 $X_0 = X$。使用均方误差作为损失函数， 即 $\ell = \frac{1}{2ml} \|X_3 - Y\|_{\mathrm{F}}^2$。

（1）[10pt] 请使用向量化的形式描述多层神经网络的前向传播过程, 即从输入 X 到计算出损失的过程。

（2）[20pt] 使用反向传播的方式计算各层参数的导数 $\frac{\partial \ell}{\partial W_i}, \frac{\partial \ell}{\partial b_{\mathrm{i}}}$, 并写出其矩阵形式。

3. 神经网络的初始化[15pt]

在课堂上讲深度神经网络的技巧的时候，我们提到不同的初始化会影响模型的性能，比如不能做全 0 初始化，并且介绍了两种常用的初始化方法，即 Xavier 初始化与 He 初始化

（1）[5pt] 试分析全 0 初始化的问题是什么？

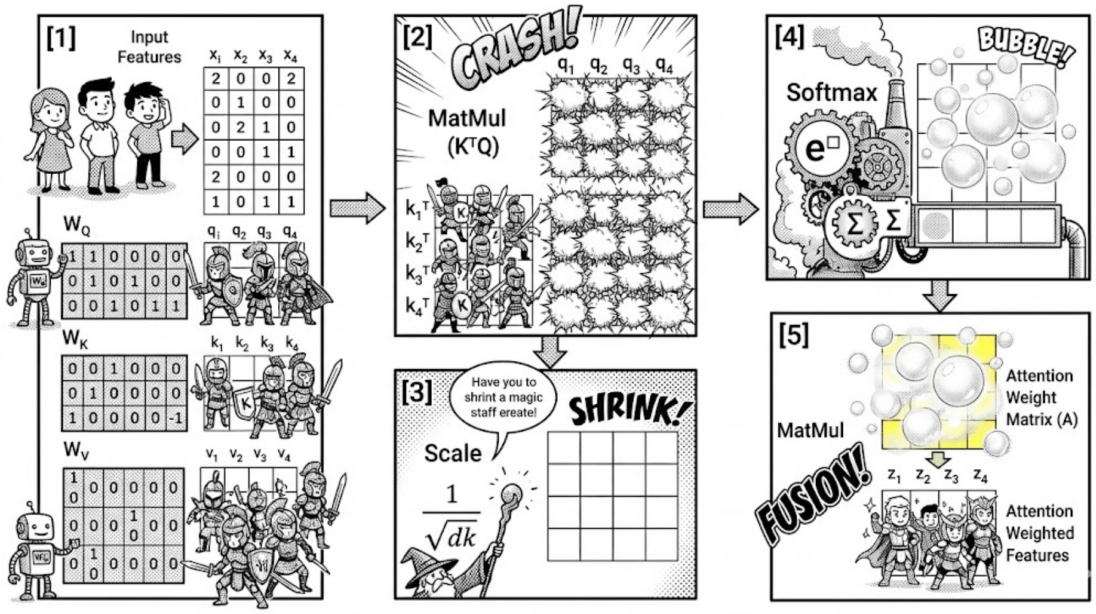
（2）[10pt] 试以一个 2 层 MLP 网络为例，推导分析 Xavier 初始化的好处，以及为什么更适用于 Sigmoid、Tanh 未激活函数

的 MLP 网络而不适用于 ReLU 激活函数（提醒：如果直接把这个问题复制给大模型，可能得到的结果并不能拿满分，建议答案中的每一步推导，自己都要能够完全理解其含义）

4. Self-Attention [10pt]

为了防止直接用大模型得到答案，我生成了一张图片来模拟一个具体的 Self-Attention 的计算过程，

（1）[5pt] 请以矩阵运算的形式，写出每一步的计算公式和计算结果（由于图片生成的原因，W_v 矩阵中的两行“10”为数字十，假设 Scale 中根号下 dk 数值为 2，为了简化计算，e当做 3 来处理）



（2）[5pt] 请分析，为什么要做第 3 步 Scale，如果不做会有什么问题？