3.4 softmax回归重点摘录与练习解答

（1）独热编码

但一般的分类问题并不与类别之间的自然顺序有关。统计学家很早以前就发明了一种表示分类数据的简单方法：独热编码（one-hot encoding）。独热编码是一个向量，它的分量和类别一样多。类别对应的分量设置为 1，其他所有分量设置为 0。在我们的例子中，标签 $\boldsymbol{y}$ 将是一个三维向量，其中 $(1,0,0)$ 对应于“猫”、$(0,1,0)$ 对应于“鸡”、$(0,0,1)$ 对应于“狗”：

$$\boldsymbol{y} \in \{(1,0,0),(0,1,0),(0,0,1)\}.$$

（2）网络构架

与线性回归一样，softmax 回归也是一个单层神经网络，由于计算每个输出 o_1、o_2 和 o_3 取决于所有输入 $\boldsymbol{x}_1$、$\boldsymbol{x}_2$、$\boldsymbol{x}_3$ 和 $\boldsymbol{x}_4$，所以softmax回归的输出层也是全连接层。

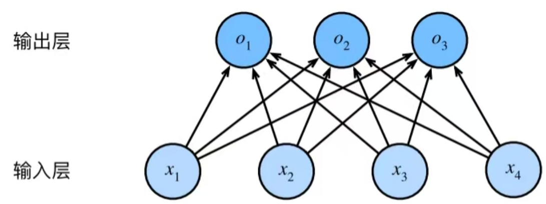

图 1: 单层神经网络

（3）Softmax 运算

现在我们将优化参数以最大化观测数据的概率。为了得到预测结果，我们将设置一个阈值，如选择具有最大概率的标签。

我们希望模型的输出 $\hat{\boldsymbol{y}}_j$ 可以视为属于类 j 的概率，然后选择具有最大输出值的类别 $\arg\max_j \hat{\boldsymbol{y}}_j$ 作为我们的预测。例如，如果 $\hat{\boldsymbol{y}}_1$、$\hat{\boldsymbol{y}}_2$ 和 $\hat{\boldsymbol{y}}_3$ 分别为 0.1、0.8 和 0.1，那么我们预测的类别是 2，在我们的例子中代表“鸡”。

然而我们不能直接将未规范化的预测 o 直接视作我们感兴趣的输出，因为在通常情况下，

$$\sum_i o_i \neq 1,$$

并且由于输入的不同，可能存在 $o_i < 0$ 的情况，直接作为概率是不适合的。

因此使用 softmax 函数对概率进行校正：softmax 函数能够将未规范化的预测变换为非负数并且总和为1，同时让模型保持可导的性质。

$$\hat{\boldsymbol{y}} = softmax(\boldsymbol{o}) \quad 其中 \quad \hat{\boldsymbol{y}}_j = \frac{\exp(o_j)}{\sum_k \exp(o_k)}$$

这里，对于所有的 j 总有 $0 \leq \hat{\boldsymbol{y}}_j \leq 1$。因此，$\hat{\boldsymbol{y}}$ 可以视为一个正确的概率分布。softmax运算不会改变未规范化的预测 $\boldsymbol{o}$ 之间的大小次序，只会确定分配给每个类别的概率。因此，在预测过程中，

我们仍然可以用下式来选择最有可能的类别。

$$\arg\max_j \hat{\boldsymbol{y}}_j = \arg\max_j o_j.$$

尽管 softmax 函数是一个非线性函数，但 softmax 回归的输出仍然由输入特征的仿射变换决定，故 softmax 回归是一个线性模型（linear model）。

（4）小批量样本的矢量化

假设我们读取了一个批量的样本 $\boldsymbol{X}$，其中特征维度（输入数量）为 d，批量大小为 n。此外，假设我们在输出中有 q 个类别。那么小批量样本的特征为 $\boldsymbol{X} \in \mathbb{R}^{n\times d}$，权重为 $\boldsymbol{W} \in \mathbb{R}^{d\times q}$，偏置为 $\boldsymbol{b} \in \mathbb{R}^{1\times q}$。softmax 回归的矢量计算表达式为：

$$\boldsymbol{O} = \boldsymbol{X}\boldsymbol{W} + \boldsymbol{b},$$

$$\hat{\boldsymbol{Y}} = softmax(\boldsymbol{O}).$$

（5）损失函数的相关推导

softmax函数给出了一个向量 $\hat{\boldsymbol{y}}$，我们可以将其视为“对给定任意输入 $\boldsymbol{x}$ 的每个类的条件概率”。假设整个数据集 $\boldsymbol{X}, \boldsymbol{Y}$ 具有 n 个样本，其中索引 i 的样本由特征向量 $\boldsymbol{x}^{(i)}$ 和独热标签向量 $\boldsymbol{y}^{(i)}$ 组成。根据独立性写出似然函数如下：

$$L(\boldsymbol{Y} \mid \boldsymbol{X}) = \prod_{i=1}^{n} P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}).$$

根据 MLE ，我们最大化 $L(\boldsymbol{Y} \mid \boldsymbol{X})$，相当于最小化负对数似然：

$$\arg\min\left(-\log L(\boldsymbol{Y} \mid \boldsymbol{X})\right) = \arg\min\left(-\sum_{i=1}^{n} \log P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)})\right) = \arg\min \sum_{i=1}^{n} l(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)}),$$

其中，对于任何标签 $\boldsymbol{y}$ 和模型预测 $\hat{\boldsymbol{y}}$，使用交叉熵（cross-entropy）损失函数：

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

由于 $\boldsymbol{y}$ 是一个长度为 q 的独热编码向量，则：

$$y_i = \begin{cases} 1, \ i = j, \\ 0, \ i \neq j. \end{cases}$$

由于所有 $\hat{y}_j$ 都是预测的概率，所以它们的对数永远不会大于 0。

下面是对于交叉熵损失函数的推导：对于第 i 行，$l(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)}) = -\log P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)})$，其中，$P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)})$ 意为给定 $\boldsymbol{x}^{(i)}$ 时，$\boldsymbol{y}^{(i)}$ 的条件概率，且 $\boldsymbol{y}^{(i)}$ 的 q 个项中，

$$y_j = 1, \ and \ y_i = 0, \forall i \neq j.$$

因此有：

$$\sum_{j=1}^{q} \boldsymbol{y}_j \log \hat{\boldsymbol{y}}_j = 0 \times \log \hat{\boldsymbol{y}}_1 + \cdots + 1 \times \log \hat{\boldsymbol{y}}_j + \cdots + 0 \times \log \hat{\boldsymbol{y}}_q = \log \hat{\boldsymbol{y}}_j$$

根据条件概率的性质可知有：$P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}) = \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} = \hat{\boldsymbol{y}}_j$. 因此：

$$l(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)}) = -\log P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}) = -\log \hat{\boldsymbol{y}}_j = -\sum_{j=1}^{q} \boldsymbol{y}_j \log \hat{\boldsymbol{y}}_j$$

（6）softmax 及其导数

由于softmax和相关的损失函数很常见，因此我们需要更好地理解它的计算方式。利用softmax的定义，$\sum_{j=1}^{q} y_j = 1$我们得到：

$$\begin{aligned} l(\mathbf{y}, \hat{\mathbf{y}}) &= -\sum_{j=1}^{q} y_j \log \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} \\ &= \sum_{j=1}^{q} y_j \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j \\ &= \left(\log \sum_{k=1}^{q} \exp(o_k)\right) \cdot \left(\sum_{j=1}^{q} y_j\right) - \sum_{j=1}^{q} y_j o_j \\ &= \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j. \end{aligned}$$

考虑相对于任何未规范化的预测 o_j 的导数，我们得到：

$$\partial_{o_j} l(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} - y_j = \mathrm{softmax}(\mathbf{o})_j - y_j.$$

换句话说，导数是我们softmax模型分配的概率与实际发生的情况（由独热标签向量表示）之间的差异。从这个意义上讲，这与我们在回归中看到的非常相似，其中梯度是观测值 $\mathbf{y}$ 和估计值 $\hat{\mathbf{y}}$ 之间的差。

（7）问题解答

1、我们可以更深入地探讨指数族与softmax之间的联系

解： 计算softmax交叉熵损失 $l(\mathbf{y},\hat{\mathbf{y}})$ 的二阶导数：

$$\partial_{o_j} l(\mathbf{y},\hat{\mathbf{y}}) = \frac{\exp(o_j)}{\sum_{k=1}^q \exp(o_k)} - y_j = \mathrm{softmax}(\boldsymbol{o})_j - y_j$$

$$\begin{aligned}\partial^2_{o_j} l(\mathbf{y},\hat{\mathbf{y}}) &= \frac{\exp(o_j)\sum_{k=1}^q \exp(o_k) - \exp(o_j)^2}{\left(\sum_{k=1}^q \exp(o_k)\right)^2} \\ &= softmax(\boldsymbol{o})_j - softmax(\boldsymbol{o})_j^2 \\ &= softmax(\boldsymbol{o})_j(1 - softmax(\boldsymbol{o})_j)\end{aligned}$$

计算 softmax$(\boldsymbol{o})$ 给出的分布方差，并与上面计算的二阶导数匹配，期望计算如下：

$$\begin{aligned}E[softmax(\boldsymbol{o})_j] &= \frac{1}{q}\sum_{j=1}^q 1\cdot softmax(\boldsymbol{o})_j \\ &= \frac{1}{q}\sum_{j=1}^q \frac{\exp(o_j)}{\sum_k \exp(o_k)} \\ &= \frac{1}{q}.\end{aligned}$$

方差计算如下：

$$\begin{aligned}Var[\boldsymbol{o}] &= E[(softmax(\boldsymbol{o})_j - E[softmax(\boldsymbol{o})_j])^2] \\ &= \frac{1}{q}\sum_{j=1}^q (softmax(\boldsymbol{o})_j - \frac{1}{q})^2 \\ &= \frac{1}{q}\left[\sum_{j=1}^q softmax^2(\boldsymbol{o})_j - \frac{2}{q}\sum_{j=1}^q softmax(\boldsymbol{o})_j + \frac{1}{q}\right] \\ &= -\frac{1}{q}\left[\sum_{j=1}^q softmax(\boldsymbol{o})_j - \sum_{j=1}^q softmax^2(\boldsymbol{o})_j + \frac{2}{q} - 1 - \frac{1}{q}\right] \\ &= \frac{q-1}{q^2} - \frac{1}{q}\sum_{j=1}^q (softmax(\boldsymbol{o})_j - softmax^2(\boldsymbol{o})_j) \\ &= \frac{q-1}{q^2} - \frac{1}{q}\sum_{j=1}^q \partial^2_{o_j} l(\mathbf{y},\hat{\mathbf{y}}).\end{aligned}$$

3、softmax 是对上面介绍的映射的误称（虽然深度学习领域中很多人都使用这个名字）。真正的 **softmax** 被定义为 $RealSoftMax(a,b) = \log(e^a + e^b)$。

解：首先证明 $\text{RealSoftMax}(a,b) > \max(a,b)$。

不妨假设假设 $\max(a,b) = a$，则

$$\begin{aligned} RealSoftMax(a,b) &= \log(e^a + e^b) \\ &> \log(e^a) \\ &= a = \max(a,b). \end{aligned}$$

然后证明 $\frac{1}{\lambda} RealSoftMax(\lambda a, \lambda b) > \max(a,b)$ 成立，前提是 $\lambda > 0$，

则当 $\lambda > 0$ 时，假设 $\max(a,b) = a$

$$\begin{aligned} \frac{1}{\lambda}\text{RealSoftMax}(\lambda a, \lambda b) &= \frac{1}{\lambda}\log(e^{\lambda a} + e^{\lambda b}) \\ &> \frac{1}{\lambda}\log(e^{\lambda a}) \\ &= a = \max(a,b). \end{aligned}$$

然后证明对于 $\lambda \to \infty$，有 $\lim\limits_{\lambda\to\infty} \frac{1}{\lambda} RealSoftMax(\lambda a, \lambda b) = \max(a,b)$；

不妨设 $\max(a,b) = a$，则有：

$$e^{\lambda a} < e^{\lambda a} + e^{\lambda b} < 2 \cdot e^{\lambda a}$$

因此有

$$\begin{aligned} a &= \lim_{\lambda\to\infty} \frac{\log(e^{\lambda a})}{\lambda} \\ &\leqslant \lim_{\lambda\to\infty} \frac{\log(e^{\lambda a} + e^{\lambda b})}{\lambda} \\ &= \lim_{\lambda\to\infty} \frac{1}{\lambda} RealSoftMax(\lambda a, \lambda b) \\ &\leqslant \lim_{\lambda\to\infty} \frac{\log(2 \cdot e^{\lambda a})}{\lambda} \\ &= \lim_{\lambda\to\infty} \frac{\log 2}{\lambda} + \lim_{\lambda\to\infty} \frac{\log(e^{\lambda a})}{\lambda} \\ &= a. \end{aligned}$$

由夹逼原理可知结论成立。

而soft-min表达式如下：

$$\text{softmin}(\boldsymbol{o})_j = \frac{\exp(-o_j)}{\sum_k \exp(-o_k)}$$