4_人工神经网络（全连接人工神经网络、前馈神经网络、多层感知机模型）

4_1_人工神经网络结构及前向传播

训练数据集

$$T=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_i,y_i),\cdots,(\mathbf{x}_N,y_N)\}$$

其中，$\mathbf{x}_i$为第i个特征向量（实例），$\mathbf{x}_i=\left(x_i^{(1)},x_i^{(2)},\ldots,x_i^{(j)},\ldots,x_i^{(n)}\right)^T\in\mathcal{X}\subseteq\mathbb{R}^n$；$y_i$为$\mathbf{x}_i$的类别标记，常将类别标记表示为类别位置为1，其余位置为0的类别向量（one-hot编码）。

人工神经网络输入层（层 1）

$$\begin{aligned}\mathbf{a}^1&=\left(a_1^1,a_2^1,\ldots,a_j^1,\ldots,a_n^1\right)^T\\ a_j^1&=x_i^{(j)}\quad(j=1,2,\ldots,n)\end{aligned}$$

人工神经网络隐藏层（层 2）

$$\begin{aligned}\mathbf{a}^2&=\left(a_1^2,a_2^2,\ldots,a_j^2,\ldots,a_p^2\right)^T\\ a_j^2&=\sigma\left(z_j^2\right)\\ z_j^2&=\sum_k w_{jk}^2\cdot a_k^1+b_j^2\quad(j=1,2,\ldots,p)\\ \mathbf{z}^2&=\left(z_1^2,z_2^2,\ldots,z_j^2,\ldots,z_p^2\right)^T\end{aligned}$$

人工神经网络输出层（层 3）

$$\begin{aligned}\mathbf{a}^3&=\left(a_1^3,a_2^3,\ldots,a_j^3,\ldots,a_m^3\right)^T\\ a_j^3&=\sigma\left(z_j^3\right)\\ z_j^3&=\sum_k w_{jk}^3\cdot a_k^2+b_j^3\quad(j=1,2,\ldots,m)\\ \mathbf{z}^2&=\left(z_1^2,z_2^2,\ldots,z_j^2,\ldots,z_m^2\right)^T\end{aligned}$$

预测输出

$$\begin{aligned}\hat{\mathbf{y}}&=\left(\hat{y}_1,\hat{y}_2,\ldots,\hat{y}_j,\ldots,\hat{y}_m\right)^T\\ \hat{y}_j&=a_j^3\quad(j=1,2,\ldots,m)\end{aligned}$$

实际输出

$$\mathbf{y}=\left(y_1,y_2,\ldots,y_j,\ldots,y_m\right)^T\quad(j=1,2,\ldots,m)$$

其中激活函数$\sigma(z)$为sigmoid函数：

$$\sigma(z)=sigmoid(z)=\frac{1}{1+\exp(-z)}$$

4_2_损失函数及经验损失

单个实例$\mathbf{x}$的损失函数$L_{\mathbf{x}}\left(\mathbf{y}, \hat{\mathbf{y}}\right)$为平方损失函数

$$C_{\mathbf{x}} = \frac{1}{2}\|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \frac{1}{2}\sum_j \left(y_j - \hat{y}_j\right)^2$$

目标函数（经验风险）

$$C = \frac{1}{N}\sum_{\mathbf{x}} C_{\mathbf{x}}$$

4_3_误差反向传播

定义第l层的第j个神经元上的误差

$$\delta_j^l \equiv \frac{\partial C_{\mathbf{x}}}{\partial z_j^l} \quad (l = 2, 3)$$

输出层误差

$$\begin{aligned}
\delta_j^3 &= \frac{\partial C_{\mathbf{x}}}{\partial z_j^3} \\
&= \frac{\partial C_{\mathbf{x}}}{\partial a_j^3} \cdot \frac{\partial a_j^3}{\partial z_j^3} \\
&= \frac{\partial C_{\mathbf{x}}}{\partial a_j^3} \cdot \sigma'\left(z_j^3\right) \\
&= \frac{\partial\left(\frac{1}{2}\sum_j \left(y_j - \hat{y}_j\right)^2\right)}{\partial a_j^3} \cdot \sigma'\left(z_j^3\right) \\
&= \left(a_j^3 - y_j\right) \cdot \sigma'\left(z_j^3\right) \quad (j = 1, 2, \ldots, m)
\end{aligned}$$

隐藏层误差

$$\begin{aligned}
\delta_j^2 &= \frac{\partial C_{\mathbf{x}}}{\partial z_j^2} \\
&= \sum_k \frac{\partial C_{\mathbf{x}}}{\partial z_k^3} \cdot \frac{\partial z_k^3}{\partial z_j^2} \\
&= \sum_k \frac{\partial z_k^3}{\partial z_j^2} \cdot \delta_k^3 \\
&= \sum_k \frac{\partial \left(\sum_j w_{kj}^3 \cdot a_j^2 + b_k^3 \right)}{\partial z_j^2} \cdot \delta_k^3 \\
&= \sum_k \frac{\partial \left(\sum_j w_{kj}^3 \cdot \sigma\left(z_j^2\right) + b_k^3 \right)}{\partial z_j^2} \cdot \delta_k^3 \\
&= \sum_k w_{kj}^3 \cdot \sigma'\left(z_j^2\right) \cdot \delta_k^3 \\
&= \sigma'\left(z_j^2\right) \cdot \sum_k w_{kj}^3 \delta_k^3 \quad (j = 1, 2, \ldots, p) \quad (k = 1, 2, \ldots, m)
\end{aligned}$$

损失函数在隐藏层（层2）/输出层（层3）关于偏置的梯度

$$\frac{\partial C_{\mathbf{x}}}{\partial b_j^l} = \frac{\partial C_{\mathbf{x}}}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial b_j^l} = \delta_j^l \cdot \frac{\partial \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right)}{\partial b_j^l} = \delta_j^l \quad (l = 2, 3)$$

损失函数在隐藏层（层2）/输出层（层3）关于权值的梯度

$$\frac{\partial C_{\mathbf{x}}}{\partial w_{jk}^l} = \frac{\partial C_{\mathbf{x}}}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial w_{jk}^l} = \delta_j^l \cdot \frac{\partial \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right)}{\partial w_{jk}^l} = \delta_j^l \cdot a_k^{l-1} \quad (l = 2, 3)$$

误差反向传播算法：

1. 输入实例$\mathbf{x}$: 为输入层设置对应的激活值$\mathbf{a}^1$；
2. 前向传播：对每个l（$l = 2, 3$）计算
$$\begin{aligned}
a_j^l &= \sigma\left(z_j^l\right) \\
z_j^l &= \sum_k w_{jk}^l \cdot a_k^{l-1} + b_j^l
\end{aligned}$$
3. 输出层误差δ_j^3；
4. 反向误差传播：隐藏层误差δ_j^2；
5. 输出：损失函数在隐藏层（层2）/输出层（层3）关于偏置及权值的梯度$\frac{\partial C_{\mathbf{x}}}{\partial b_j^l}$和$\frac{\partial C_{\mathbf{x}}}{\partial w_{jk}^l}$。

4_4_梯度下降算法

梯度下降算法：

1. 输入训练实例的集合；
2. 对每个训练实例$\mathbf{x}$：设置对应的输入激活$\mathbf{a}^{\mathbf{x},1}$，并执行以下步骤：
    - 前向传播：计算$\mathbf{z}^{\mathbf{x},l}=\mathbf{w}^l\mathbf{a}^{\mathbf{x},l-1}+\mathbf{b}^l$及$\mathbf{a}^{\mathbf{x},l}=\sigma\left(\mathbf{z}^{\mathbf{x},l}\right)$，其中$l=2,3,\cdots,L$。
    - 输出层误差：$\delta^{\mathbf{x},L}=\nabla_{\mathbf{a}}C_{\mathbf{x}}\odot\sigma'\left(\mathbf{z}^{\mathbf{x},L}\right)$
    - 误差反向传播：对每个$l=L-1,L-2,\cdots,2$，计算$\delta^{\mathbf{x},l}=\left(\left(\mathbf{w}^{l+1}\right)^{\top}\delta^{\mathbf{x},l+1}\right)\odot\sigma'\left(\mathbf{z}^{\mathbf{x},l}\right)$。
3. 梯度下降：对每个$l=L-1,L-2,\cdots,2$，根据$\mathbf{w}^l\rightarrow\mathbf{w}^l-\frac{\eta}{m}\sum_{\mathbf{x}}\delta^{\mathbf{x},l}\left(\mathbf{a}^{\mathbf{x},l-1}\right)^{\top}$和$\mathbf{b}^l\rightarrow\mathbf{b}^l-\frac{\eta}{m}\sum_{\mathbf{x}}\delta^{\mathbf{x},l}$更新权重和偏置。

4_5_人工神经网络的改进

1. 交叉熵损失函数：
   单个实例$\mathbf{x}$的损失函数$L_{\mathbf{x}}\left(\mathbf{y},\hat{\mathbf{y}}\right)$为交叉熵损失函数

$$\begin{aligned}C_{\mathbf{x}}&=-\left(\mathbf{y}\ln\hat{\mathbf{y}}+(1-\mathbf{y})\ln\left(1-\hat{\mathbf{y}}\right)\right)\\&=-\sum_j\left[y_j\ln\hat{y}_j+\left(1-y_j\right)\ln\left(1-\hat{y}_j\right)\right]\end{aligned}$$

1. 目标函数正则化（结构风险）：
   L2正则化：$C=\frac{1}{N}\sum_{\mathbf{x}}C_{\mathbf{x}}+\frac{\lambda}{2N}\sum_w w^2$
   L1正则化：$C=\frac{1}{N}\sum_{\mathbf{x}}C_{\mathbf{x}}+\frac{\lambda}{N}\sum_w|w|$

1. 激活函数：
   双曲正切激活函数

$$\sigma(z)=\tanh(z)=\frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$$

   输出层柔性最大值激活函数

$$\begin{aligned}\sigma\left(z^L\right)&=softmax\left(z^L\right)\\&=\frac{\exp\left(z^L\right)}{\sum\exp\left(z^L\right)}\end{aligned}$$

1. 权重初始化：设神经元有n_{in} 个输入权重，可使权重

$$w\backsim N\left(0,\left(\frac{1}{\sqrt{n_{in}}}\right)^2\right)$$