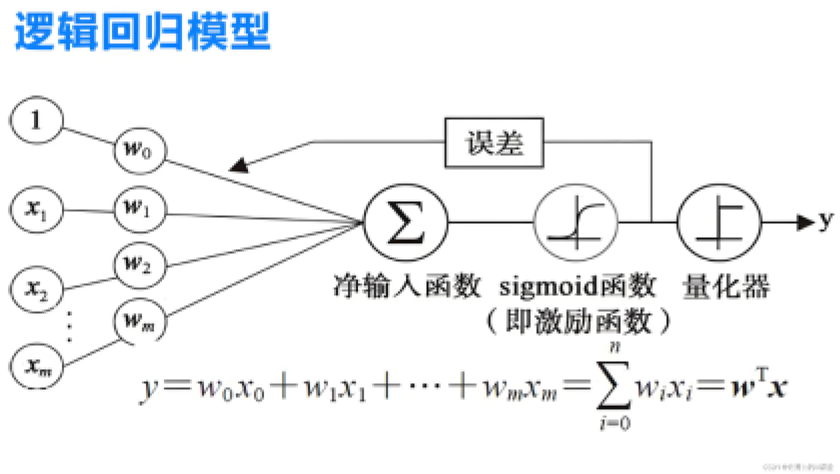
神经网络算法推导全过程

一. 二元 logistic 回归

假设现在有 m 个样本(x_i, y_i)，每个样板的维度为 n。



1.1 前向传播推导过程

(1) $X=\begin{bmatrix} x_1^1 & \dots & x_m^1 \\ \dots & \dots & \dots \\ x_1^n & \dots & x_m^n \end{bmatrix}$, $Y=[y_1,\dots,y_m]$

(2) $z=w^T x+b$, $w=\begin{bmatrix} w_1 \\ \dots \\ w_n \end{bmatrix}$, $b = w_0$

(3) $\hat{y}=\frac{1}{1+e^{-z}}$，$\hat{y}$为预测值，$Loss(w,b) = -(y\log(\hat{y}) + (1-y)\log(1-\hat{y})$

(4) 代价函数 $J(w,b)=-\frac{1}{m}\sum_{i=1}^{m} y_i\log(\hat{y}_i)+(1-y_i)\log(1-\hat{y}_i)$

1.2 反向传播过程推导

利用微积分原理，初始化 w, b。然后利用导数和随机梯度下降算法不断求取最优解。

（1）
$$
\begin{aligned}
&\mathrm{w} = w - \alpha \frac{\vartheta L(w,b)}{\vartheta w} \\
&b = b - \alpha \frac{\vartheta L(w,b)}{\vartheta b}
\end{aligned}
$$

$$
\begin{aligned}
&\mathrm{w} = w - \alpha dw, dw = (\hat{y}_i - y_i)x_i \\
&b = b - \alpha db, db = (\hat{y}_i - y_i)
\end{aligned}
$$

（2） 考虑有m个样本

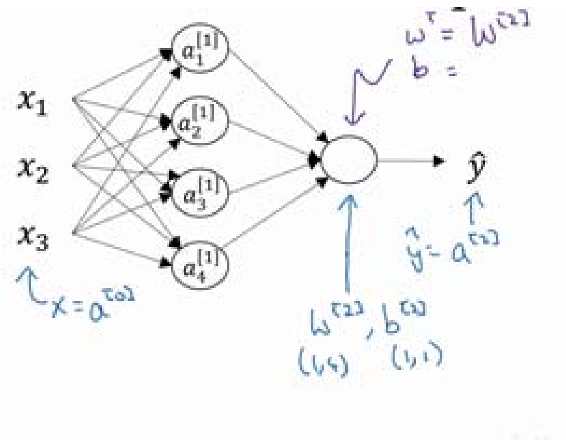
$$
w = w - \alpha \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)x_i
$$

$$
b = b - \alpha \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)
$$

（3）最后，通过循环多次，不断更新 w 和 b 求取最优解。

二. 人工神经网络

假设现在有 m 个样本（x_i, y_i），每个样板的维度为 n。



2.1 人工神经网络前向传播

（1） $\mathrm{X} = \begin{bmatrix} x_1^1 & \dots & x_m^1 \\ \dots & \dots & \dots \\ x_1^n & \dots & x_m^n \end{bmatrix}$ ， $\mathrm{Y} = [y_1, \dots, y_m]$

（2）

隐藏层前向传播过程

隐藏层激活函数$:\theta^{[1]}(\mathrm{z})=\tanh(\mathrm{z})=\dfrac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$

$\mathrm{z}_1{}^{[1]}=w_1{}^{[1]}x+b_1{}^{[1]},a_1{}^{[1]}=\theta^{[1]}(\mathrm{z}_1{}^{[1]}),$

$\mathrm{z}_2{}^{[1]}=w_2{}^{[1]}x+b_2{}^{[1]},a_2{}^{[1]}=\theta^{[1]}(\mathrm{z}_2{}^{[1]})$

...

$z_l{}^{[1]}=w_l{}^{[1]}x+b_l{}^{[1]},a_l{}^{[1]}=\theta^{[1]}(z_l{}^{[1]})$

向量化表示$:z^{[1]}=w^{[1]}x+b^{[1]},a^{[1]}=\theta^{[1]}(\mathrm{z}^{[1]})$

$w_{(l,n)},z_{(l,m)},b_{(l,1)},a_{(l,m)}$

输出层前向传播过程

输出层激活函数$:\mathrm{f}(z)=sigmoid(z)=\dfrac{1}{1+e^{-z}}$

$z^{[2]}=w^{[2]}a^{[1]}+b^{[2]},a^{[2]}=sigmoid(z^{[2]})$

$w^{[2]}{}_{(1,l)},z^{[2]}$

$\widehat{\mathrm{y}}=a^{[2]}$

整理如下

$z^{[1]}=w^{[1]}x+b^{[1]}$

$a^{[1]}=\theta^{[1]}(z^{[1]})$

$z^{[2]}=w^{[2]}a^{[1]}+b^{[2]}$

$a^{[2]}=\theta^{[2]}(z^{[2]})$

(3) 损失函数

$$\mathrm{Cost}(w,b)=-\frac{1}{m}\sum_{i=1}^{m}y_i\log(\widehat{y}_i)+(1-y_i)\log(1-\widehat{y}_i)$$

2.2 人工神经网络反向传播

利用微积分原理，初始化 w, b。然后利用链式求导法和随机梯度下降算法不断求取最优解。

(1)输出层通过误差反向传播更新权重

$$\mathrm{dw}^{[2]}=\frac{\partial J(w^{[2]},b^{[2]},w^{[1]},b^{[1]})}{\partial w^{[2]}},\mathrm{db}^{[2]}=\frac{\partial J(w^{[2]},b^{[2]},w^{[1]},b^{[1]})}{\partial b^{[2]}}$$

$$\mathrm{w}^{[2]}=w^{[2]}-\alpha dw^{[2]},dw^{[2]}=(\hat{y}-y)*z^{[2]}$$

$$b^{[2]}=b^{[2]}-\alpha db^{[2]},db[2]=(\hat{y}-y)$$

考虑m个样本

$$\mathrm{w}^{[2]}=\mathrm{w}^{[2]}-a\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i-y_i)*a_i^{[1]}$$

$$b^{[2]}=b^{[2]}-a\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i-y_i)$$

(2) 隐藏层通过误差反向传播更新权重

$$\mathrm{z}^{[1]}=w^{[1]}x+b^{[1]},z^{[1]}\text{维度}:(l,m),w^{[1]}\text{维度}:(l,n)$$

$$a^{[1]}=\theta^{[1]}(z^{[1]}),\theta^{[1]}=\frac{e^z-e^{-z}}{e^z+e^{-z}},a^{[1]}\text{维度}:(l,m)$$

$$\mathrm{z}^{[2]}=w^{[2]}a^{[1]}+b^{[2]},\mathrm{z}^{[2]}\text{维度}:(1,1),w^{[2]}\text{维度}:(1,l)$$

$$a^{[2]}=\theta^{[2]}(z^{[2]}),\theta^{[2]}=\frac{1}{1+e^{-z}},\text{维度}:(1,1)$$

$$\hat{y}=a^{[2]}$$

$$J(w^{[2]},b^{[2]},w^{[1]},b^{[1]})=-\frac{1}{m}\sum_{i=1}^{m}y_i\log\hat{y}_i+(1-y_i)\log(1-\hat{y}_i)$$

$$\frac{\partial J}{\partial w^{[1]}}=dw^{[1]}=\frac{\hat{y}_i-y_i}{(1-\hat{y}_i)\hat{y}_i}*(1-\hat{y}_i)\hat{y}_i*w^{[2]}*\frac{\partial\theta^{[1]}}{\partial z_i^{[1]}}*x_i=(\hat{y}_i-y_i)w^{[2]^T}*\frac{\partial\theta^{[1]}}{\partial z}*x_i^T$$

$$\frac{\partial J}{\partial b^{[1]}}=db^{[1]}=\frac{\hat{y}_i-y_i}{(1-\hat{y}_i)\hat{y}_i}*(1-\hat{y}_i)\hat{y}_i*w^{[2]}*\frac{\partial\theta^{[1]}}{\partial z_i^{[1]}}=(\hat{y}_i-y_i)w^{[2]^T}*\frac{\partial\theta^{[1]}}{\partial z}$$

考虑n个样本

$$w^{[1]}=w^{[1]}-a\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i-y_i)w^{[2]^T}*\frac{\partial\theta^{[1]}}{\partial z_i^{[1]}}*x_i^T$$

$$b^{[1]}=b^{[1]}-a\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i-y_i)w^{[2]^T}*\frac{\partial\theta^{[1]}}{\partial z_i^{[1]}}$$