BP神经网络

梯度下降回归

求解无约束凸函数的极小值

$$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$$

$$\theta^*=\arg\min_\theta J(\theta)$$

线性回归回顾

$$y^{(i)}=\theta^T x^{(i)}+\varepsilon^{(i)}$$

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$$

$$\theta=\theta-\alpha\bullet\frac{\partial J(\theta)}{\partial\theta}$$

1.求解无约束凸函数最小值问题

2.定义损失函数J和梯度下降更新参数的方法

Logistic回归回顾

$$p=h_\theta(x)=g\left(\theta^T x\right)=\frac{1}{1+e^{-\theta^T x}}$$

$$g(z)=\frac{1}{1+e^{-z}}$$

$$loss=-\ell(\theta)=-\sum_{i=1}^{m}\left(y^{(i)}\ln h_\theta\left(x^{(i)}\right)+\left(1-y^{(i)}\right)\ln\left(1-h_\theta\left(x^{(i)}\right)\right)\right)$$

$$=\sum_{i=1}^{m}\left[-y^{(i)}\ln\left(h_\theta\left(x^{(i)}\right)\right)-\left(1-y^{(i)}\right)\ln\left(1-h_\theta\left(x^{(i)}\right)\right)\right]$$

$$g'(z)=\left(\frac{1}{1+e^{-z}}\right)'=\frac{e^{-z}}{\left(1+e^{-z}\right)^2}$$

$$=\frac{1}{1+e^{-z}}\cdot\frac{e^{-z}}{1+e^{-z}}=\frac{1}{1+e^{-z}}\cdot\left(1-\frac{1}{1+e^{-z}}\right)$$

$$=g(z)\cdot(1-g(z))$$

$$\theta_j=\theta_j+\alpha\sum_{i=1}^{m}\left(y^{(i)}-h_\theta\left(x^{(i)}\right)\right)x_j^{(i)}$$

将线性回归的输出通过一个逻辑函数（通常是Sigmoid函数）映射到0和1之间，得到一个概率值

p为样本属于正类的该率

神经网络之BP算法

求解w的算法，根据误差值修改每一层的权重，继续迭代

神经网络之SGD

输出层误差

$$E=\frac{1}{2}(d-O)^2=\frac{1}{2}\sum_{k=1}^{\ell}\left(d_k-O_k\right)^2$$

隐层的误差

$$E=\frac{1}{2}\sum_{k=1}^{\ell}\left(d_k-f(net_k)\right)^2=\frac{1}{2}\sum_{k=1}^{\ell}\left(d_k-f\left(\sum_{j=1}^{m}w_{jk}y_j\right)\right)^2$$

输入层误差

$$E=\frac{1}{2}\sum_{k=1}^{\ell}\left(d_k-f\left[\sum_{j=0}^{m}w_{jk}f\left(net_j\right)\right]\right)^2=\frac{1}{2}\sum_{k=1}^{\ell}\left(d_k-f\left[\sum_{j=0}^{m}w_{jk}f\left(\sum_{i=1}^{n}\upsilon_{ij}x_i\right)\right]\right)^2$$

SGD的核心思想是在每次迭代中随机选择一个样本（或一小批样本）来估计梯度，而不是使用整个数据集。这样做的优点是计算效率高，尤其是当数据集很大时。此外，SGD也能够逃离局部最小值，因为随机性引入了一定的噪声，有助于模型探索更多的参数空间

BP算法例子

FP过程

BP过程

多次迭代效果

$$\frac{\partial E_{total}}{\partial w_5}=-\left(\text{target}-out_{o1}\right)*out_{o1}*\left(1-out_{o1}\right)*out_{h1}$$

$$\frac{\partial E_{total}}{\partial w_1}=\frac{\partial E_{total}}{\partial out_{h1}}*\frac{\partial out_{h1}}{\partial net_{h1}}*\frac{\partial net_{h1}}{\partial w_1}=\left(\frac{\partial E_{o1}}{\partial out_{h1}}+\frac{\partial E_{o2}}{\partial out_{h1}}\right)*\frac{\partial out_{h1}}{\partial net_{h1}}*\frac{\partial net_{h1}}{\partial w_1}$$