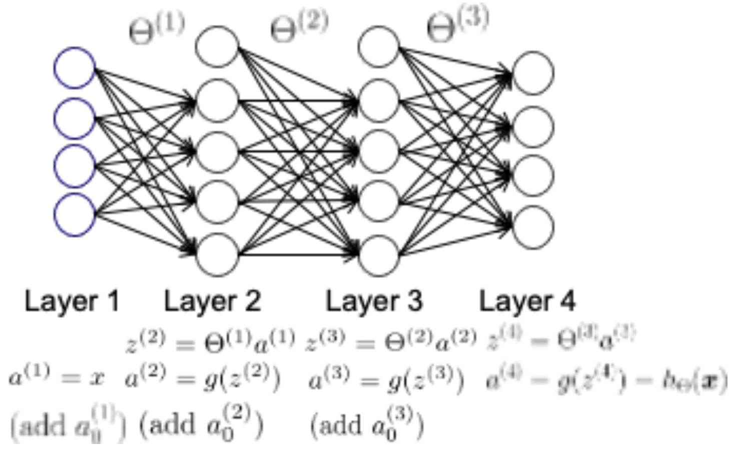
神经网络的BP算法

---

1. 使用神经网络可进行K类分类任务。



第1层：$\boldsymbol{a}^{(1)} = \boldsymbol{x}$，增加$\boldsymbol{a}_0^{(1)} = 1$
第2层：$\boldsymbol{z}^{(2)} = \Theta^{(1)}\boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)} = g(\boldsymbol{z}^{(2)})$，增加$\boldsymbol{a}_0^{(2)} = 1$
第3层：$\boldsymbol{z}^{(3)} = \Theta^{(2)}\boldsymbol{a}^{(2)}, \boldsymbol{a}^{(3)} = g(\boldsymbol{z}^{(3)})$，增加$\boldsymbol{a}_0^{(3)} = 1$
第4层：$\boldsymbol{z}^{(4)} = \Theta^{(3)}\boldsymbol{a}^{(3)}, \boldsymbol{a}^{(4)} = g(\boldsymbol{z}^{(4)}) = h_\Theta(\boldsymbol{x})$，增加$\boldsymbol{a}_0^{(4)} = 1$

2. 代价函数：采用交叉熵来度量 y 和$h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})$之间的差异。

$$\begin{aligned} J(\boldsymbol{\theta}) &= -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)}\log_2(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))_k + (1-y_k^{(i)})\log_2(1-h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))_k\right] \quad (1) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)}\log_2(\boldsymbol{a}^{(3)}(\boldsymbol{x}^{(i)}))_k + (1-y_k^{(i)})\log_2(1-\boldsymbol{a}^{(3)}(\boldsymbol{x}^{(i)}))_k\right] \quad (2) \end{aligned}$$

或使用均方误差：

$$\begin{aligned} J(\boldsymbol{\theta}) &= -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)} - (h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))_k\right]^2 \quad (3) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)} - \boldsymbol{a}^{(3)}\right]^2 \quad (4) \end{aligned}$$

3. **BP**算法总体也是使用梯度下降法进行优化。只是因为多层神经网络导致计算代价函数相对于模型参数的导数中存在多级嵌套函数，因此，需要使用链式求导法。尤其是隐含层节点的梯度计算需要先计算分解到当前隐含层节点的误差。

   基于交叉熵代价函数的误差(输出层)

$$\begin{aligned}
\delta_{CE}^{(3)} &= \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{z}^{(3)}} = \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{a}^{(3)}} \frac{\partial \boldsymbol{a}^{(3)}}{\partial \boldsymbol{z}^{(3)}} && (5) \\
&= \frac{a^{(3)} - y}{a^{(3)}(1 - a^{(3)})} \left[a^{(3)}(1 - a^{(3)})\right] && (6) \\
&= a^{(3)} - y && (7)
\end{aligned}$$

基于均方误差和代价函数的误差（输出层）

$$\begin{aligned}
\delta_{MSE}^{(3)} &= \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{z}^{(3)}} = \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{a}^{(3)}} \frac{\partial \boldsymbol{a}^{(3)}}{\partial \boldsymbol{z}^{(3)}} && (8) \\
&= \frac{1}{2} \times 2(a^{(3)} - y) \times \frac{\partial \boldsymbol{a}^{(3)}}{\partial \boldsymbol{z}^{(3)}} && (9) \\
&= (a^{(3)} - y)a^{(3)}(1 - a^{(3)}) && (10)
\end{aligned}$$

面向隐含层节点的误差（隐含层）

$$\begin{aligned}
\delta^{(2)} &= \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{a}^{(3)}} \frac{\partial \boldsymbol{a}^{(3)}}{\partial \boldsymbol{z}^{(3)}} \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} && (11) \\
&= \delta^{(3)} \cdot \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} && (12) \\
&= \delta^{(3)} \cdot \boldsymbol{\theta}^{(2)} \cdot \boldsymbol{a}^{(2)}(1 - \boldsymbol{a}^{(2)}) && (13)
\end{aligned}$$

基于感知器的输出误差计算输入参数的梯度

$$\begin{aligned}
\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{(2)}} &= \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{a}^{(3)}} \frac{\partial \boldsymbol{a}^{(3)}}{\partial \boldsymbol{z}^{(3)}} \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{\theta}^{(2)}} = \delta^{(3)} \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{\theta}^{(2)}} && (14) \\
&= \delta^{(3)} \boldsymbol{a}^{(2)} && (15)
\end{aligned}$$

类似地，

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{(1)}} = \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{a}^{(3)}} \frac{\partial \boldsymbol{a}^{(3)}}{\partial \boldsymbol{z}^{(3)}} \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \frac{\partial \boldsymbol{z}^{(2)}}{\partial \boldsymbol{\theta}^{(1)}} \tag{16}$$

$$= \delta^{(3)} \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \frac{\partial \boldsymbol{z}^{(2)}}{\partial \boldsymbol{\theta}^{(1)}} \tag{17}$$

$$= \delta^{(2)} \frac{\partial \boldsymbol{z}^{(2)}}{\partial \boldsymbol{\theta}^{(1)}} = \delta^{(2)} \boldsymbol{a}^{(1)} \tag{18}$$

神经网络的误差反向传递算法也是梯度下降算法，其中更新链接权重的表达式如下：

$$\Theta_{ij}^{(l)}(t+1) = \Theta_{ij}^{(l)}(t) - \alpha \frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}} \tag{19}$$

【3层神经网络（或MLP）的BP算法】：

1、输入与前向传播

将m个样本$\boldsymbol{x}$赋值给$\boldsymbol{a}^{(1)}$；之后，增加$\boldsymbol{a}_0^{(1)} = 1$；然后，接着计算$\boldsymbol{z}^{(2)}$、$\boldsymbol{a}^{(2)}$、$\boldsymbol{z}^{(3)}$、$\boldsymbol{a}^{(3)}$。

2、计算输出层的误差

$\delta^{(3)} = \boldsymbol{a}^{(3)} - y$

3、计算隐含层（$l = 2$）的误差（输出层误差$\delta^{(3)}$反向传递至隐含层）

$\delta^{(2)} = (\Theta^{(2)})^T \delta^{(3)} .* g'(\boldsymbol{z}^{(2)})$

4、面向每条连接上的权重，根据$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{(2)}} = \delta^{(3)} \boldsymbol{a}^{(2)}$和$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{(1)}} = \delta^{(2)} \boldsymbol{a}^{(1)}$计算梯度

① 为每条连接累积每条训练样本上的梯度，$\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$

② 计算$\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}} = \frac{1}{m} \Delta_{ij}^{(l)}$。如果代价函数加入正则化项，要加上$\frac{\lambda}{m} \theta_{ij}^{(l)}$（$j \neq 0$）。

5、使用$\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}}$和梯度下降法优化模型参数。

【算法实现的细节】

1、，此时，$\boldsymbol{a}^{(1)}$尺寸为$(K^{(1)}+1)\times m$，其中，$K^{(1)}=n$；尺寸分别为$K^{(2)}\times m$、$(K^{(2)}+1)\times m$、$K^{(3)}\times m$、$(K^{(3)}+1)\times m$。

2、其中，$\delta^{(3)}$的尺寸为：$K^{(3)}\times m$

3、其中，$\Theta^{(2)}$尺寸为$K^{(3)}\times(K^{(2)}+1)$，$\boldsymbol{z}^{(2)}$尺寸为$K^{(2)}\times m$。此步不使用$\Theta_0^{(2)}$。