http://deeplearning.stanford.edu/wiki/index.php/UFLDL%E6%95%99%E7%A8%8B

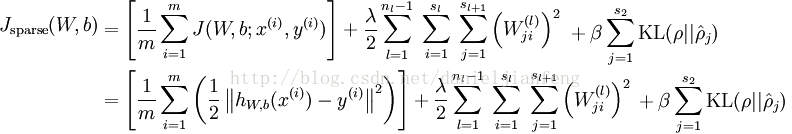
反向传播算法可表示为以下几个步骤：

1. 进行前馈传导计算，利用前向传导公式，得到 \textstyle L_2, L_3, \ldots直到输出层 \textstyle L_{n_l} 的激活值。

前向传播：

 \begin{align}
z^{(l+1)} &= W^{(l)} a^{(l)} + b^{(l)}   \\
a^{(l+1)} &= f(z^{(l+1)})
\end{align}其中f是激励函数

稀疏自动编码的损失函数的lose function：



以上关于\textstyle J(W,b)定义中的第一项是一个均方差项average sum-of-squares error。

其中对于单个样例 \textstyle (x,y)，其代价函数为：


\begin{align}
J(W,b; x,y) = \frac{1}{2} \left\| h_{W,b}(x) - y \right\|^2.
\end{align}

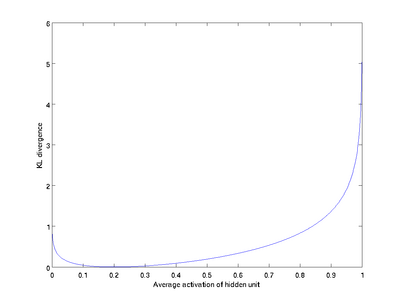

第二项是一个规则化项regularization term（也叫**权重衰减项**），其目的是减小权重的幅度，防止过度拟合。

第三项 相对熵（KL divergence）:

\textstyle {\rm KL}(\rho || \hat\rho_j)
 = \rho \log \frac{\rho}{\hat\rho_j} + (1-\rho) \log \frac{1-\rho}{1-\hat\rho_j}

\textstyle \rho 是**稀疏性参数**，sparsityParam通常是一个接近于0的较小的值

\begin{align}
\hat\rho_j = \frac{1}{m} \sum_{i=1}^m \left[ a^{(2)}_j(x^{(i)}) \right]
\end{align}表示隐藏神经元 \textstyle j 的平均活跃度



所以，最小化这一惩罚因子具有使得 \textstyle \hat\rho_j 靠近 \textstyle \rho 的效果。

1. 对输出层（第 \textstyle n_l 层），计算：

 \begin{align}
\delta^{(n_l)}
= - (y - a^{(n_l)}) \bullet f'(z^{(n_l)})
\end{align}

**如果选择 \textstyle f(z) = 1/(1+\exp(-z)) ，也就是sigmoid函数，那么它的导数就是 \textstyle f'(z) = f(z) (1-f(z)) （如果选择tanh函数，那它的导数就是 \textstyle f'(z) = 1- (f(z))^2**

1. 对于 \textstyle l = n_l-1, n_l-2, n_l-3, \ldots, 2 的各层，计算：

 \begin{align}
\delta^{(l)} = \left((W^{(l)})^T \delta^{(l+1)}\right) \bullet f'(z^{(l)})
\end{align}

为了对相对熵进行导数计算，我们可以使用一个易于实现的技巧，这只需要在你的程序中稍作改动即可。具体来说，前面在后向传播算法中计算第二层（ \textstyle l=2 ）更新的时候我们已经计算了

\begin{align}
\delta^{(2)}_i = \left( \sum_{j=1}^{s_{2}} W^{(2)}_{ji} \delta^{(3)}_j \right) f'(z^{(2)}_i),
\end{align}

现在我们将其换成

\begin{align}
\delta^{(2)}_i =
  \left( \left( \sum_{j=1}^{s_{2}} W^{(2)}_{ji} \delta^{(3)}_j \right)
+ \beta \left( - \frac{\rho}{\hat\rho_i} + \frac{1-\rho}{1-\hat\rho_i} \right) \right) f'(z^{(2)}_i) .
\end{align}

就可以了。

1. 计算最终需要的偏导数值：

 \begin{align}
\nabla_{W^{(l)}} J(W,b;x,y) &= \delta^{(l+1)} (a^{(l)})^T, \\
\nabla_{b^{(l)}} J(W,b;x,y) &= \delta^{(l+1)}.
\end{align}