

# 自动编码器AE

## 基本思想

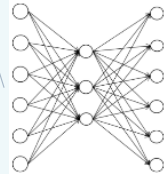
利用神经网络，直接将一个样本映射成一组特征向量。就是输入一组数据，然后利用神经网络的变换输出

这有一个问题，这组特征是什么，我们并不能指定，指定就加入了人工干扰

## 编码器结构

编码器的结构如图所示，输入层和隐含层是编码器，隐含层和输出层是解码器。

借助于PCA的思想，将一个向量输入之后再将它还原回来，即输入一个 $x$ 输出一个 $x'$ 。然后计算 $x$ 和 $x'$ 的差异，将两者的差异最小化，然后去掉输出层就得到了想要的编码器。



## 损失函数

$$\min \frac{1}{2l} \sum_i \|x_i - g_{\theta'}(h_{\theta}(x_i))\|_2^2$$

$h$ 是编码器， $g$ 是解码器

$$\min \frac{1}{2l} \sum_i \|x_i - x'_i\|_2^2$$

就好像这样子

## 去噪自动编码器

去噪自动编码器就是在某个样本的属性中选取一定量的属性然后将这些属性设置成0，然后将其作为自动编码器的输入，以原有的样本作为对比的样本计算重构误差。这样就可以训练出一个可以消除噪声的编码器

$$\min \frac{1}{l} \sum_{i=1}^l L(x_i, g_{\theta'}(h_{\theta}(\tilde{x}_i)))$$

## 稀疏自动编码器

这个编码器的作用是使编码器的隐含层的输出值尽可能接近0，即造成了稀疏输出。

如何造成稀疏是引入了稀疏惩罚项

$$\sum_{i=1}^n (\rho \ln \frac{\rho}{\hat{\rho}_i} + (1 - \rho) \ln \frac{1 - \rho}{1 - \hat{\rho}_i})$$

$\rho_i$ 是第 $i$ 个神经元的平均激活度。就是一共有 $L$ 个样本，每个样本都算了一遍，在这第 $i$ 个神经元上都有激活函数值，然后将这 $L$ 个值取个平均，就得到了第 $i$ 个神经元的平均激活度。 $\rho$ 是人工设定的活跃度，是一个接近于0的值。 $n$ 个激活函数

损失函数

$$\min \frac{1}{2l} \sum_i \|x_i - g_{\theta'}(h_{\theta}(x_i))\|_2^2 + \beta \sum_{i=1}^n (\rho \ln \frac{\rho}{\hat{\rho}_i} + (1 - \rho) \ln \frac{1 - \rho}{1 - \hat{\rho}_i})$$

## 收缩自动编码器

在损失函数中加上正则化项，使得编码器函数的导数尽可能小

损失函数

$$L_{CAE}(\theta) = \sum_{x \in D_n} (L(x, g(h(x))) + \lambda \|L_h(x)\|^2)$$

$$\|L_h(x)\|^2 = \sum_{i,j} (\frac{\partial h_j(x)}{\partial x_i})^2$$

## 多层自动编码器

将多个编码器级联起来。逐层提取输入数据的特征，在此过程中逐层降低输入数据的维度。上一级的输出作为下一级的输入。

训练方法：训练是逐层进行的，首先训练第一层自动编码器。再将第一层编码器的输出作为第二层编码器的输入。

训练应该是使用编码器加解码器完整的结构。在第一层将原本数据先降维再还原，跟原有数据比较优化参数。然后去掉解码层，将第一层的输出作为第二层的输入再用完整的结构训练