

# 自编码、自编码的稀疏性、自我学习

2015年7月27日 星期一

下午10:29

## autoencoder and sparsity

### 一、简介

就是一个input和output都是训练样本的网络，目的是为了学中间的隐层

### 二、原理

相当于自编码无监督的压缩，类似PCA的作用，他利用天然的input本身作为监督学习的目标，从而在数据角度是一个无监督学习，原理是是一个监督学习，学习的过程相当于先压缩，在解压，学习的目标就是压缩再解压之后，output和之前的input也很像

### 三、拓展

当然隐层dim也可以>input的dim，这也可能得到一些有意思的结构

这里面还有些稀疏化的内容，理论基础可能和正则化类似，但这里没提，只是说nerual network的激活hidden unit，大多数应该是inactive的状态，这就是

### 四、进一步拓展：稀疏性

稀疏性，为了保持这个稀疏性，设了一个稀疏化系数

$\rho$

is a sparsity parameter.

做稀疏化的过程是：计算平均活跃度，训练集所有样本算出的激活值乘样本加起来

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m [a_j^{(2)}(x^{(i)})]$$

希望有

$$\hat{\rho}_j = \rho,$$

所以优化目标加上了这两个的KL距离

$$\sum_{j=1}^{s_2} \text{KL}(\rho || \hat{\rho}_j),$$

s2是隐层dim

加入损失函数，求导就行

但是有个很大的弊端：

加入稀疏性限制之后，必须提前求得整个训练集上的平均激活度，相当于需要两次前馈，所以不实用

问题：如果不稀疏了会怎么样？

丧失了nerual network的特性，so what?

## 五、应用上的拓展：自我学习

详见 <http://ufldl.stanford.edu/wiki/index.php/%E8%87%AA%E6%88%91%E5%AD%A6%E4%B9%A0>

可以通过autoencoder学出隐层(无监督的), 然后把隐层和输入 $x_i$ 并在一起作为 $x_i$ 的输入, 传入后面的阶段, 后面可以是正常的有监督学习或者其他的。

这个过程仅仅是给输入数据加了一个自己学出的特征(隐层)

(当然, 输入也可以直接用隐层而不用 $x_i$ 了)

$m_l$

的已标注训练集

$$\{(x_l^{(1)}, y^{(1)}), (x_l^{(2)}, y^{(2)}), \dots (x_l^{(m_l)}, y^{(m_l)})\}$$

(下标

$l$

表示“带类标”)

训练集就变成

$$\{(a_l^{(1)}, y^{(1)}), (a_l^{(2)}, y^{(2)}), \dots (a_l^{(m_l)}, y^{(m_l)})\}$$

或者是

$$\{((x_l^{(1)}, a_l^{(1)}), y^{(1)}), ((x_l^{(2)}, a_l^{(1)}), y^{(2)}), \dots, ((x_l^{(m_l)}, a_l^{(1)}), y^{(m_l)})\}$$

(

参考资料:

[http://ufldl.stanford.edu/wiki/index.php/Autoencoders\\_and\\_Sparsity](http://ufldl.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity)

<http://ufldl.stanford.edu/wiki/index.php/%E8%87%AA%E7%BC%96%E7%A0%81%E7%AE%97%E6%B3%95%E4%B8%8E%E7%A8%80%E7%96%8F%E6%80%A7>

<http://ufldl.stanford.edu/wiki/index.php/%E8%87%AA%E6%88%91%E5%AD%A6%E4%B9%A0>