

EvanJames的专栏

Be Real. Be Yourself.

目录视图

个人资料



积分：3924
等级：BLOG > 5
排名：第6946名

原创：227篇 转载：100篇
译文：0篇 评论：9条

文章搜索

文章分类

- Android (47)
- LeetCode (147)
- Java (65)
- 数据结构 (13)
- deep learning (7)
- Python (18)
- C语言 (1)
- matlab (16)
- css (5)
- 资料参考 (5)
- MySQL (1)

文章存档

- 2016年11月 (30)
- 2016年10月 (31)
- 2016年03月 (1)
- 2016年01月 (2)
- 2015年12月 (2)

展开

阅读排行

- matlab二维三维图形绘制和坐... (8452)
- Matlab自带排序函数sort用法 (5001)
- IGARSS 会议介绍 (3745)

【公告】关于开启用户注册及登录手机短信验证的通知 CSDN日报20170415 —— 《谈谈程序员的焦虑感》 博客

Sparse Autoencoder(稀疏自编码器)

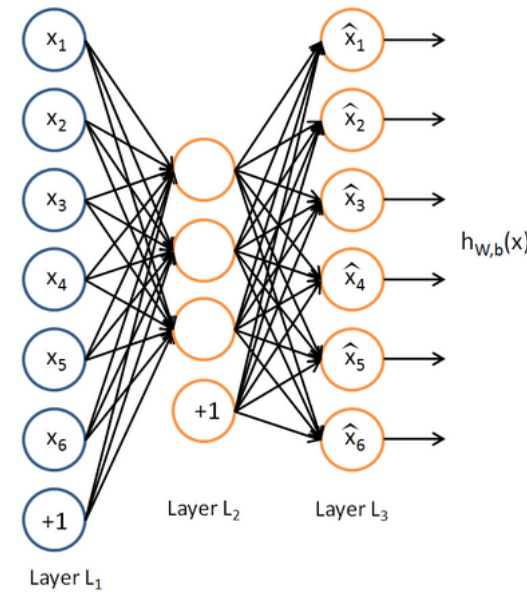
标签：SAE deep learning matlab

2014-10-16 15:46 838人阅读 评

分类：deep learning (6)

前言：

这节课来学习下Deep learning领域比较出名的一类算法——sparse autoencoder，即稀疏模式的自动编码。我们知道，deep le learning，所以这里的sparse autoencoder也应是无监督的。按照前面的博文：[Deep learning：一\(基础知识_1\)](#)，[Deep learning](#)是有监督的学习的话，在神经网络中，我们只需要确定神经网络的结构就可以求出损失函数的表达式了（当然，该表达式需对网络的参数不要太大），同时也能求出损失函数偏导函数的表达式，然后利用优化算法求出网络最优的参数。应该清楚的是，损失函数的表达式中，那么这里的sparse autoencoder为什么能够无监督学习呢？难道它的损失函数的表达式中不需要标注的样本值（即通常所说的y值）么？值也是需要的，只不过它的输出理论值是本身输入的特征值x，其实这里的标注值y=x。这样做的好处是，网络的隐含层能够很好的代替较准确的还原出那些输入特征值。Sparse autoencoder的一个网络结构图如下所示：



损失函数的求法：

无稀疏约束时网络的损失函数表达式如下：

$$J(W, b) = \left[\frac{1}{m} \sum_{i=1}^m J(W, b; x^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2$$
$$= \left[\frac{1}{m} \sum_{i=1}^m \left(\frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2$$

稀疏编码是对网络的隐含层的输出有了约束，即隐含层节点输出的平均值应尽量为0，这样的话，大部分的隐含层节点都处于非activ sparse autoencoder损失函数表达式为：

$$J_{\text{sparse}}(W, b) = J(W, b) + \beta \sum_{j=1}^{s_2} \text{KL}(\rho || \hat{\rho}_j),$$

libsvm ——SVM中参数 c和q...	(3478)
神经网络工具箱-matlab 详细...	(3048)
java中startsWith与endsWith...	(2878)
CNN卷积神经网络反向推导	(2452)
LinkedHashMap源码分析与L...	(2146)
Android 动画	(2081)
java简单的字符串大小比较—...	(1900)

评论排行

Android开发常用功能	(2)
Android 动画	(2)
matlab灰度共生矩阵提取纹理...	(1)
Efficient Learning of Sparse ...	(1)
matlab二维三维图形绘制和...	(1)



你应该知道吧 Android 数据库(系列)来啦

* HDF5副本放置节点选择的优化

* CSDN日报20170416 —— 《为什么程序员活少钱多死得早?》

* 凡人视角C++之string(上)

最新评论

Matlab自带排序函数sort用法	小奇葩 : 索引 那块的那个逻辑 你好像不小心写错了
Android开发常用功能	geanwen : 第一个计时器的那个, 还可以用属性动画实现吧, 是不是能稍微好一点
Android开发常用功能	ganshenml : 东西很实用! 赞一个
Android 动画	戎码人生 : 博主加油!!!
Android 动画	戎码人生 : 博主加油!!!
android 完美获取音乐文件中的专辑图片...	失落无意义 : 的确能获取, 但是只有很少一部分能显示出来, 有些明明有图, 但不能显示, 不知道是怎么回事
matlab二维三维图形绘制和坐标轴范围...	miangmiang咩 : 怎样设置mesh的纵坐标的范围
Efficient Learning of Sparse Representa...	校草的舍友 : 大神可以举个例子吗
matlab灰度共生矩阵提取纹理特征	Beidou973 : 谢谢

后面那项为KL距离, 其表达式如下:

$$KL(\rho||\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$$

隐含层节点输出平均值求法如下:

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

其中的参数一般取很小, 比如说0.05, 也就是小概率发生事件的概率。这说明要求隐含层的每一个节点的输出均值接近0.05 (其实activite函数为sigmoid函数), 这样就达到稀疏的目的了。KL距离在这里表示的是两个向量之间的差异值。从约束函数表达式中可以看出因此最终的隐含层节点的输出会接近0.05。

损失函数的偏导数的求法:

如果不加入稀疏规则, 则正常情况下由损失函数求损失函数偏导数的过程如下:

1. Perform a feedforward pass, computing the activations for layers L_2 , L_3 , and so on up to the output layer L_{n_L} .

2. For each output unit i in layer n_J (the output layer), set

$$\delta_i^{(n_i)} = \frac{\partial}{\partial z_i^{(n_i)}} \frac{1}{2} \|y - h_{W,b}(x)\|^2 = -(y_i - a_i^{(n_i)}) \cdot f'(z_i^{(n_i)})$$

3. For $l = n_l - 1, n_l - 2, n_l - 3, \dots, 2$
For each node i in layer l , set

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) f'(z_i^{(l)})$$

4. Compute the desired partial derivatives, which are given as:

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W; b; x, y) = a_j^{(l)} \delta_i^{(l+1)}$$

$$\frac{\partial}{\partial b_i^{(l)}} J(W; b; x, y) = \delta_i^{(l+1)}.$$

而加入了稀疏性后, 神经元节点的误差表达式由公式:

$$\delta_i^{(2)} = \left(\sum_{j=1}^{s_2} W_{ji}^{(2)} \delta_j^{(3)} \right) f'(z_i^{(2)}),$$

变成公式:

$$\delta_i^{(2)} = \left(\left(\sum_{j=1}^{s_2} W_{ji}^{(2)} \delta_j^{(3)} \right) + \beta \left(-\frac{\rho}{\hat{\rho}_i} + \frac{1-\rho}{1-\hat{\rho}_i} \right) \right) f'(z_i^{(2)}).$$

梯度下降法求解:

有了损失函数及其偏导数后就可以采用梯度下降法来求网络最优化的参数了, 整个流程如下所示:

1. Set $\Delta W^{(l)} := 0$, $\Delta b^{(l)} := 0$ (matrix/vector of zeros) for all l .

2. For $i = 1$ to m ,

a. Use backpropagation to compute $\nabla_{W^{(l)}} J(W; b; x, y)$ and $\nabla_{b^{(l)}} J(W; b; x, y)$.

b. Set $\Delta W^{(l)} := \Delta W^{(l)} + \nabla_{W^{(l)}} J(W; b; x, y)$.

c. Set $\Delta b^{(l)} := \Delta b^{(l)} + \nabla_{b^{(l)}} J(W; b; x, y)$.

3. Update the parameters:

$$W^{(l)} = W^{(l)} - \alpha \left[\left(\frac{1}{m} \Delta W^{(l)} \right) + \lambda W^{(l)} \right]$$

$$b^{(l)} = b^{(l)} - \alpha \left[\frac{1}{m} \Delta b^{(l)} \right]$$

从上面的公式可以看出, 损失函数的偏导其实是个累加过程, 每来一个样本数据就累加一次。这是因为损失函数本身就是由每个训练照加法的求导法则, 损失函数的偏导也应该是由各个训练样本所损失的偏导叠加而成。从这里可以看出, 训练样本输入网络的顺序并不重的操作是等价的, 后面样本的输入所产生的结果并不依靠前一次输入结果 (只是简单的累加而已, 而这里的累加是顺序无关的)。

参考资料:

- Deep learning : 一(基础知识_1)
- Deep learning : 七(基础知识_2)
- http://deeplearning.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity

顶

0

踩

0

- [上一篇](#) CNN卷积神经网络反向推导
- [下一篇](#) Table of results for Caltech 101 dataset

我的同类文章

deep learning (6)

• Deep learning加速之Theano的安装和G..

2015-09-29

阅读 766

• 深度学习 deep learning 部分工具箱和...

2014-10-20

阅读 1014

• Table of results for Caltech 101 datas...

2014-10-16

阅读 749

• SLIC算法 超像素 superpixel

2015-04-

• Efficient Learning of Sparse Represen...

2014-10-

• CNN卷积神经网络反向推导

2014-10-



参考知识库



PHP知识库
6509 关注 | 1053 收录



算法与数据结构知识库
15549 关注 | 2320 收录

猜你在找

- [数据结构与算法在实战项目中的应...](#)
- [稀疏自动编码Sparse Autoencoder](#)
- [使用决策树算法对测试数据进行分...](#)
- [sparse Autoencoder3---自编码算...](#)
- [数据结构和算法](#)
- [UFLDL练习Sparse Autoencoder](#)
- [数据结构基础系列\(1\)：数据结构...](#)
- [sparse-autoencoder](#)
- [以性别预测为例，谈谈数据挖掘中...](#)
- [UFLDL Tutorial_Sparse Autoenc...](#)



查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题	Hadoop	AWS	移动游戏	Java	Android	iOS	Swift	智能硬件	Docker	OpenStack	VPN	
IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP	jQuery	BI	HTML5	Spring	Apache
HTML	SDK	IIS	Fedora	XML	LBS	Unity	Splashtop	UML	components	Windows Mobile	Rails	
Cassandra	CloudStack	FTC	coremail	OPhone	CouchBase	云计算	iOS6	Rackspace	Web App	S		

02

