

原 Deep Learning (深度学习) 学习笔记整理系列之 (五)

分类: Deep Learning 机器学习 机器视觉

2013-04-09 12:53 830人阅读

Deep Learning (深度学习) 学习笔记整理系列

zouxy09@qq.com

<http://blog.csdn.net/zouxy09>

作者: **Zouxy**

version 1.0 2013-04-08

声明:

- 1) 该Deep Learning的学习系列是整理自网上很大牛和机器学习专家所无私奉献的资料的。具体引用的资料请看参考文献。具体的版本声明也参考原文献。
- 2) 本文仅供学术交流, 非商用。所以每一部分具体的参考资料并没有详细对应。如果某部分不小心侵犯了大家的利益, 还望海涵, 并联系博主删除。
- 3) 本人才疏学浅, 整理总结的时候难免出错, 还望各位前辈不吝指正, 谢谢。
- 4) 阅读本文需要机器学习、计算机视觉、神经网络等等基础(如果没有也没关系了, 没有就看看, 能不能看懂, 呵呵)。
- 5) 此属于第一版本, 若有错误, 还需继续修正与增删。还望大家多多指点。大家都共享一点点, 一起为祖国科研的推进添砖加瓦(呵呵, 好高尚的目标啊)。请联系:
zouxy09@qq.com

目录:

一、概述

二、背景

三、人脑视觉机理

四、关于特征

4.1、特征表示的粒度

4.2、初级(浅层)特征表示

4.3、结构性特征表示

4.4、需要有多少个特征？

五、Deep Learning的基本思想

六、浅层学习（Shallow Learning）和深度学习（Deep Learning）

七、Deep learning与Neural Network

八、Deep learning训练过程

8.1、传统神经网络的训练方法

8.2、deep learning训练过程

九、Deep Learning的常用模型或者方法

9.1、AutoEncoder自动编码器

9.2、Sparse Coding稀疏编码

9.3、Restricted Boltzmann Machine(RBM)限制波尔兹曼机

9.4、Deep Belief Networks深信度网络

9.5、Convolutional Neural Networks卷积神经网络

十、总结与展望

十一、参考文献和Deep Learning学习资源

接上

9.2、Sparse Coding稀疏编码

如果我们把输出必须和输入相等的限制放松，同时利用线性代数中基的概念，即 $O = a_1 * \Phi_1 + a_2 * \Phi_2 + \dots + a_n * \Phi_n$ ， Φ_i 是基， a_i 是系数，我们可以得到这样一个优化问题：

$$\text{Min } |I - O|, \text{ 其中 } I \text{ 表示输入， } O \text{ 表示输出。}$$

通过求解这个最优化式子，我们可以求得系数 a_i 和基 Φ_i ，这些系数和基就是输入的另外一种近似表达。

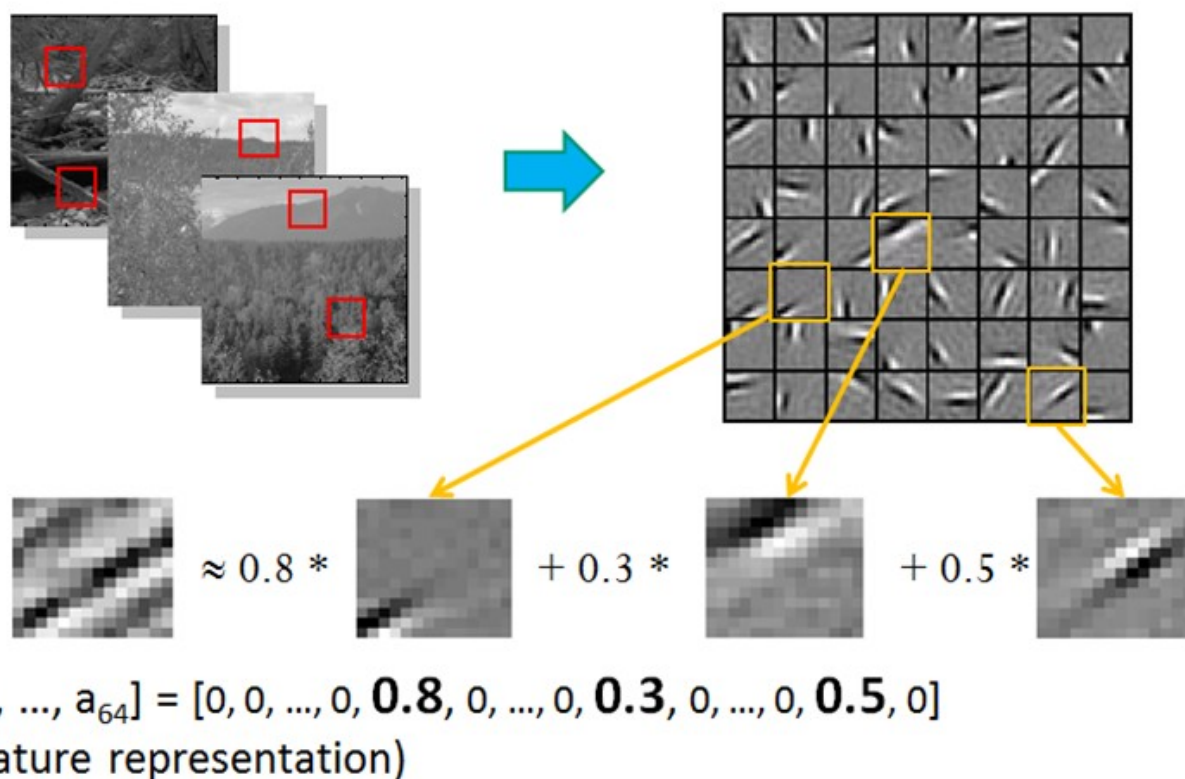
$$\mathbf{x} = \sum_{i=1}^k a_i \phi_i$$

因此，它们可以用来表达输入 \mathbf{l} ，这个过程也是自动学习得到的。如果我们在上述式子上加上L1的Regularity限制，得到：

$$\text{Min } \|\mathbf{l} - \mathbf{O}\| + u^*(\|\Phi_1\| + \|\Phi_2\| + \dots + \|\Phi_n\|)$$

这种方法被称为**Sparse Coding**。通俗的说，就是将一个信号表示为一组基的线性组合，而且要求只需要较少的几个基就可以将信号表示出来。“稀疏性”定义为：只有很少的几个非零元素或只有很少的几个远大于零的元素。要求系数 \mathbf{a}_i 是稀疏的意思就是说：对于一组输入向量，我们只想有尽可能少的几个系数远大于零。选择使用具有稀疏性的分量来表示我们的输入数据是有原因的，因为绝大多数的感官数据，比如自然图像，可以被表示成少量基本元素的叠加，在图像中这些基本元素可以是面或者线。同时，比如与初级视觉皮层的类比过程也因此得到了提升（人脑有大量的神经元，但对于某些图像或者边缘只有很少的神经元兴奋，其他都处于抑制状态）。

稀疏编码算法是一种无监督学习方法，它用来寻找一组“超完备”基向量来更高效地表示样本数据。虽然形如主成分分析技术（PCA）能使我们方便地找到一组“完备”基向量，但是这里我们想要做的是找到一组“超完备”基向量来表示输入向量（也就是说，基向量的个数比输入向量的维数要大）。超完备基的好处是它们能更有效地找出隐含在输入数据内部的结构与模式。然而，对于超完备基来说，系数 \mathbf{a}_i 不再由输入向量唯一确定。因此，在稀疏编码算法中，我们另加了一个评判标准“稀疏性”来解决因超完备而导致的退化（**degeneracy**）问题。（详细过程请参考：[UFLDL Tutorial](#)稀疏编码）



比如在图像的Feature Extraction的最底层要做Edge Detector的生成，那么这里的工作就是从Natural Images中randomly选取一些小patch，通过这些patch生成能够描述他们的“基”，也就是右边的 $8 \times 8 = 64$ 个basis组成的basis，然后给定一个test patch，我们可以按照上面的式子通过basis的线性组合得到，而sparse matrix就是a，下图中的a中有64个维度，其中非零项只有3个，故称“sparse”。

这里可能大家会有疑问，为什么把底层作为Edge Detector呢？上层又是什么呢？这里做个简单解释大家就会明白，之所以是Edge Detector是因为不同方向的Edge就能够描述出整幅图像，所以不同方向的Edge自然就是图像的basis了……而上一层的basis组合的结果，上上层又是上一层的组合basis……（就是上面第四部分的时候咱们说的那样）

Sparse coding分为两个部分：

1) **Training**阶段：给定一系列的样本图片 $[x_1, x_2, \dots]$ ，我们需要学习得到一组基 $[\Phi_1, \Phi_2, \dots]$ ，也就是字典。

稀疏编码是k-means算法的变体，其训练过程也差不多（EM算法的思想：如果要优化的目标函数包含两个变量，如 $L(W, B)$ ，那么我们可以先固定W，调整B使得L最小，然后再固定B，调整W使L最小，这样迭代交替，不断将L推向最小值。EM算法可见我的博客：“从最大似然到EM算法浅解”）。

训练过程就是一个重复迭代的过程，按上面所说，我们交替的更改a和 Φ 使得下面这个目标函数最小。

$$\min_{a, \phi} \sum_{i=1}^m \left\| x_i - \sum_{j=1}^k a_{i,j} \phi_j \right\|^2 + \lambda \sum_{i=1}^m \sum_{j=1}^k |a_{i,j}|$$

每次迭代分两步：

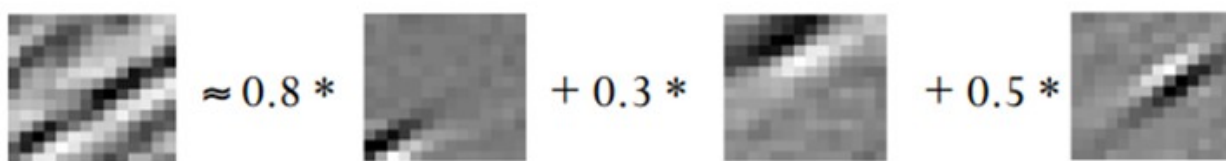
- 固定字典 $\Phi[k]$ ，然后调整 $a[k]$ ，使得上式，即目标函数最小（即解LASSO问题）。
- 然后固定住 $a[k]$ ，调整 $\Phi[k]$ ，使得上式，即目标函数最小（即解凸QP问题）。

不断迭代，直至收敛。这样就可以得到一组可以良好表示这一系列 x 的基，也就是字典。

2) **Coding**阶段：给定一个新的图片 x ，由上面得到的字典，通过解一个LASSO问题得到稀疏向量 a 。这个稀疏向量就是这个输入向量 x 的一个稀疏表达了。

$$\min_a \sum_{i=1}^m \left\| x_i - \sum_{j=1}^k a_{i,j} \phi_j \right\|^2 + \lambda \sum_{i=1}^m \sum_{j=1}^k |a_{i,j}|$$

例如：



$$\text{Image} \approx 0.8 * \text{Basis 1} + 0.3 * \text{Basis 2} + 0.5 * \text{Basis 3}$$

Represent x_i as: $a_i = [0, 0, \dots, 0, \mathbf{0.8}, 0, \dots, 0, \mathbf{0.3}, 0, \dots, 0, \mathbf{0.5}, \dots]$