

自编码神经网络与稀疏性

叶慧敏, 11531016

College of Information Science and Electronic Engineering

Zhejiang University

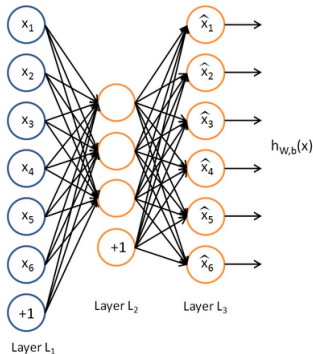
Spring, 2016

Outline

- 1 自编码算法
- 2 稀疏性
- 3 学习方式
- 4 多层网络结构
- 5 应用
- 6 参考文献

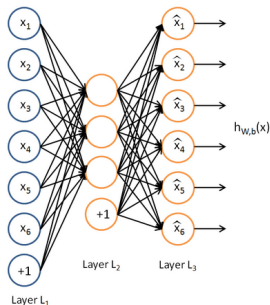
自编码神经网络

假设我们只有一个没有带类别标签的样本集合 $x^{(1)}, x^{(2)}, x^{(3)}, \dots$ ，其中 $x^{(i)} \in \mathbf{R}^n$ 。自编码神经网络采用无监督学习方式，让输出值尽量靠近输入值，即 $y^{(i)} = x^{(i)}$ 。下图是一个自编码神经网络



自编码神经网络尝试学习一个 $h_{W,b}(x) \approx x$ 的函数。换句话说，它尝试逼近一个恒等函数，从而使得输出 \hat{x} 接近于输入 x 。

自编码神经网络



限定隐藏神经元的数量，我们就可以从输入数据中发现一些有趣的结构。举例来说，假设输入 x 是一张 10×10 图像(100个像素)的像素值，于是 $n = 100$ ，其隐藏层 L_2 中有50个隐藏神经元。注意，输出也是100维的 $y \in \mathbf{R}^{100}$ 。由于只有50个隐藏神经元，迫使自编码神经网络去学习输入数据的一种压缩表示，也就是说，它必须从50维的隐藏神经元激活向量 $a^{(2)} \in \mathbf{R}^{50}$ 中重构出100维的像素灰度值输入 x 。

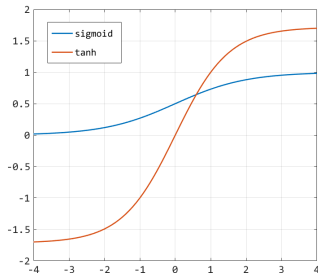
如果输入数据中隐含着一些特定的结构，比如某些输入特征是彼此相关的，那么这一算法就可以发现输入数据中的这些相关性。

Outline

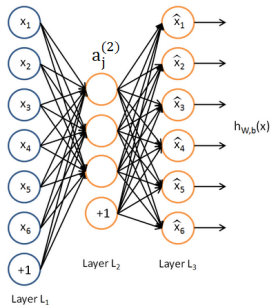
- 1 自编码算法
- 2 稀疏性
- 3 学习方式
- 4 多层网络结构
- 5 应用
- 6 参考文献

稀疏性表示

稀疏性可以解释如下：假设激活函数是sigmoid函数（或tanh函数），当神经元的输出接近于 1 的时候我们认为它被激活，而输出接近于0（或-1）的时候认为它被抑制，那么使得神经元大部分的情况下都是被抑制（输出接近于0或-1）的限制被称作稀疏性限制。



稀疏性表示



$a_j^{(2)}$ 表示隐藏神经元 j 的激活度，但是这一表示并未明确指出哪一个输入样本 x 的激活度。所以我们将使用 $a_j^{(2)}(x)$ 来表示在给定输入为 x 情况下，自编码神经网络隐藏神经元 j 的激活度。而用如下

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

表示隐藏神经元 j 的平均活跃度（在 m 个训练训练样本上取平均）。

稀疏性表示

我们可以再加入一条限制

$$\hat{\rho}_j = \rho, \quad (2)$$

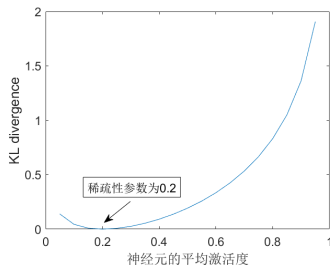
其中, ρ 是稀疏性参数, 通常是一个接近于 0 的较小的值 (比如 $\rho = 0.05$)。也就是, 我们想要让隐藏神经元 j 的平均活跃度接近 0.05, 也就是大部分的隐藏神经元的活跃度必须接近于 0 (sigmoid 函数)。

为了实现稀疏性限制, 我们将会优化的目标函数中加入一个额外的惩罚因子, 相对熵 (KL divergence):

$$\sum_{j=1}^{s_2} \text{KL}(\rho || \hat{\rho}_j) = \sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}. \quad (3)$$

这里, s_2 是隐藏层中神经元的数量, 而 j 依次代表隐藏层中的每一个神经元。

稀疏性表示



它将惩罚那些 $\hat{\rho}_j$ 和 ρ 相差很大的情况从而使得隐藏神经元的平均活跃度保持在较小值附近。

$$\sum_{j=1}^{s_2} \text{KL}(\rho || \hat{\rho}_j) = \sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}. \quad (4)$$

这里， s_2 是隐藏层中神经元的数量，而 j 依次代表隐藏层中的每一个神经元。

Outline

- 1 自编码算法
- 2 稀疏性
- 3 学习方式
- 4 多层网络结构
- 5 应用
- 6 参考文献

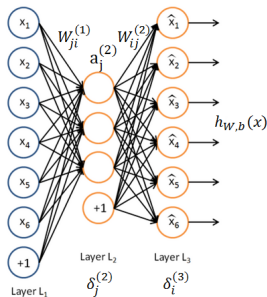
目标函数

整体代价函数为:

$$\begin{aligned}
 J(W, b) = & \left[\frac{1}{m} \sum_{t=1}^m \frac{1}{2} \| h_{W,b}(x^{(t)}) - x^{(t)} \|^2 \right] \\
 & + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \left(W_{ji}^{(l)} \right)^2 \\
 & + \beta \sum_{j=1}^{s_2} \text{KL}(\rho \| \hat{\rho}_j).
 \end{aligned} \tag{5}$$

- 第一项 $J(W, b)$ 是一个误差平方项（所有样本取平均）；
- 第二项是权重衰减项，其目的是减小权重的幅度，减缓过拟合；
- $\hat{\rho}_j$ 则也（间接地）取决于 W, b ，因为它是隐藏神经元 j 的平均激活度，由前向传播计算得到 $a = f(Wx + b)$ ；
- λ, β 权衡惩罚项的比例。

反向传播计算



$$\delta_i^{(3)} = (h_{W,b}(x^{(i)}) - x^{(i)}) f'(z_i^{(3)}), \quad (6)$$

$$\delta_j^{(2)} = \left(\left(\sum_{i=1}^{s_2} W_{ij}^{(2)} \delta_i^{(3)} \right) + \beta \left(-\frac{\rho}{\hat{\rho}_j} + \frac{1 - \rho}{1 - \hat{\rho}_j} \right) \right) f'(z_j^{(2)}). \quad (7)$$

$$\frac{\partial J}{\partial W_{ij}^{(2)}} = \delta_i^{(3)} a_j^{(2)} + \lambda W_{ij}^{(2)}, \quad \frac{\partial J}{\partial b_i^{(2)}} = \delta_i^{(3)}. \quad (8)$$

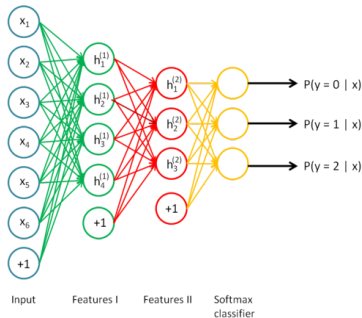
$$\frac{\partial J}{\partial W_{ji}^{(1)}} = \delta_j^{(2)} x_i + \lambda W_{ji}^{(1)}, \quad \frac{\partial J}{\partial b_j^{(1)}} = \delta_j^{(2)}. \quad (9)$$

Outline

- 1 自编码算法
- 2 稀疏性
- 3 学习方式
- 4 多层网络结构**
- 5 应用
- 6 参考文献

深度神经网络

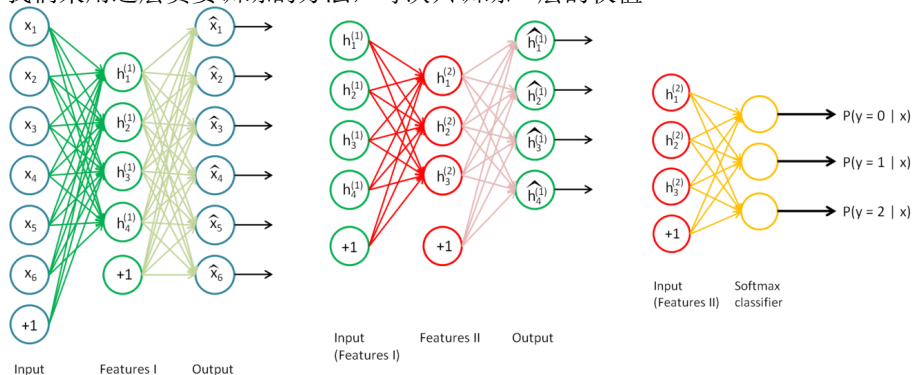
我们可以用多个自编码神经网络构建深度神经网络，用于图像分类。



举个例子，如果网络的输入数据是图像，网络的第一层会学习如何去识别边，第二层一般会学习如何去组合边，从而构成轮廓、角等。更高层会学习如何去组合更形象且有意义的特征。例如，如果输入数据集包含人脸图像，更高层会学习如何识别或组合眼睛、鼻子、嘴等人脸器官。

逐层贪婪训练

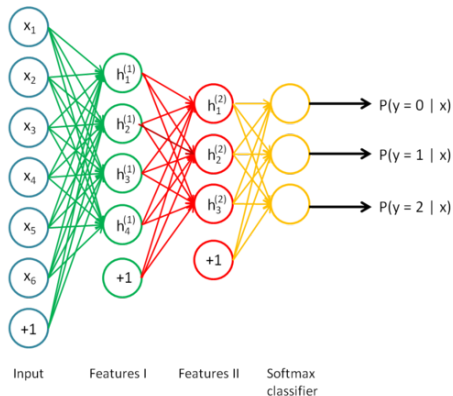
我们采用逐层贪婪训练的方法，每次只训练一层的权值



首先是（非监督）训练第一层，然后保存权值（绿色）和隐藏层的输出，舍弃之后的部分；第二层（红色）的权值也类似；最后一层是（监督）训练softmax分类器的权值(黄色)。

深度神经网络

最终，我们可以将这三层组合起来构建一个包含两个隐藏层和一个最终softmax分类器的栈式自编码网络，这个网络能够进行图像分类。

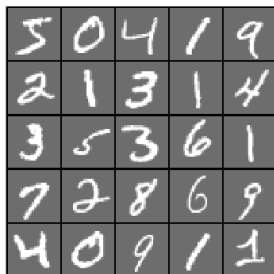


Outline

- 1 自编码算法
- 2 稀疏性
- 3 学习方式
- 4 多层网络结构
- 5 应用**
- 6 参考文献

基于公开数据集的测试

MNIST手写数据集¹由Yann LeCun et al. 维护，是一个手写数字0-9的灰度图像集合，一共含有60000张训练样本，10000张测试样本。图片的大小都统一，且数字置于图片中心位置。已经被广泛用于测试图像分类算法的性能。LeCun et al. 1998 完成准确率是99.3%，而且Ciresan et al. 2012 已经达到目前最高的准确率是99.77%，已经能和人眼的识别率相匹配。



¹<http://yann.lecun.com/exdb/mnist/>

基于MNIST数据集的分类测试

图像输入(28×28) \Rightarrow 隐藏层1(28×28) \Rightarrow 隐藏层2(28×28) \Rightarrow softmax分类



Figure 1: 输入图像

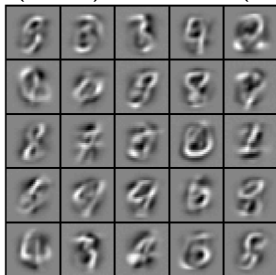


Figure 2: 第一层权值

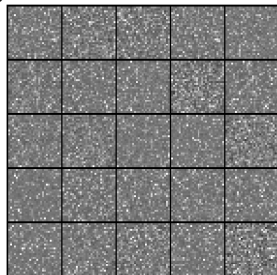


Figure 3: 第二层权值

准确率有89.34%，其实自编码网络的结构权值参数过多，所以性能并不好，不能与CNN相比拟。

Outline

- 1 自编码算法
- 2 稀疏性
- 3 学习方式
- 4 多层网络结构
- 5 应用
- 6 参考文献

Bibliography I

内容来自Andrew Ng, et al. UFLDL教程 © Stanford University.
http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial

End

谢谢！