PRML（Pattern Recognition And Machine Learning）读书会

第五章　Neural Networks

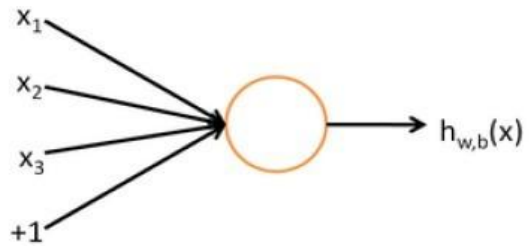
主讲人 网神

（新浪微博：@豆角茄子麻酱凉面）

QQ 群 177217565

读书会微信公众平台请扫描下面的二维码

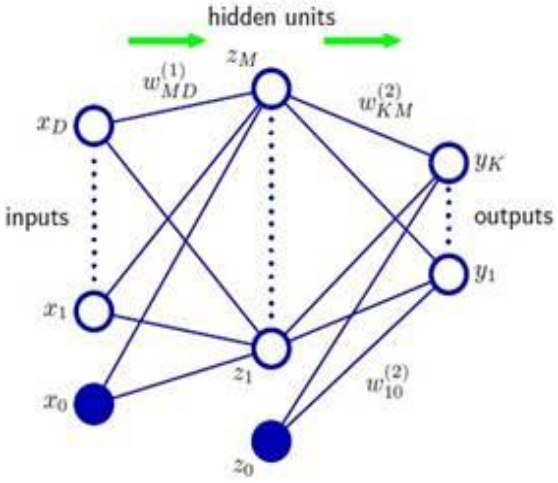
网神(66707180) 18:55:06
那我们开始了啊，前面第 3,4 章讲了回归和分类问题，他们应用的主要限制是维度灾难问题。今天的第 5 章神经网络的内容：
1. 神经网络的定义
2. 训练方法：error 函数，梯度下降，后向传导
3. 正则化：几种主要方法，重点讲卷积网络
书上提到的这些内容今天先不讲了，以后有时间再讲：BP 在 Jacobian 和 Hessian 矩阵中求导的应用；混合密度网络；贝叶斯解释神经网络。
首先是神经网络的定义，先看一个最简单的神经网络，只有一个神经元：



这个神经元是一个以 x1,x2,x3 和截距 1 为输入的运算单元，其输出是：

$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$$

其中函数 f 成为"激活函数"，activation function.激活函数根据实际应用确定，经常选择 sigmoid 函数.
如果是 sigmoid 函数，这个神经元的输入-输出的映射就是一个 logistic 回归问题。
神经网络就是将许多个神经元连接在一起组成的网络，如图：



神经网络的图跟后面第八章图模型不同， 图模型中，每个节点都是一个随机变量，符合某个分布。神经网络中的节点是确定的值，由与其相连的节点唯一确定。
上图这个神经网络包含三层，左边是输入层，x1...xd 节点是输入节点， x0 是截距项。最右边是输出层，y1,...,yk 是输出节点. 中间是隐藏层 hidden level, z1,...,zM 是隐藏节点，因为不能从训练样本中观察到他们的值，所以叫隐藏层。
可以把神经网络描述为一系列的函数转换。首先，构造 M 个输入节点的线性组合：

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}$$

上式中，j=1,...,M，对应 M 个隐藏节点. 上标(1)表示这是第一层的参数。wji 是权重 weight,wj0 是偏置. aj 叫做 activation. 中文叫激活吧，感觉有点别扭。

把 activation aj 输入到一个非线性激活函数 h(.) $z_j = h(a_j).$ 就得到了隐藏节点的值 zj。

在这个从输入层到隐藏层的转换中，这个激活函数不能是线性的，接下来，将隐藏节点的值线性组合得到输出节点的 activation：

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}$$

每个 ak 输入一个输出层激活函数 $y_k = \sigma(a_k)$ ，就得到了输出值。

这个从隐藏层到输出层的激活函数σ,根据不同应用，有不同的选择，例如回归问题的激活函数是 identity 函数,既 y = a.分类问题的激活函数选择 sigmoid 函数，multiclass 分类选择是 softmax 函数.
把上面各阶段的计算连起来，神经网络整个的计算过程就是：

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(2)} \right)$$

上面计算过程是以 2 层神经网络为例。实际使用的 NN 可能有多层，记得听一个报告现在很火的 deep learning 的隐藏层有 5-9 层.这个计算过程 forward propagation，从前向后计算。与此相反，训练时，会用 back propagation 从后向前来计算偏导数。
神经网络有很强的拟合能力，可以拟合很多种的曲线，这个图是书上的一个例子，对四种曲线的拟合：

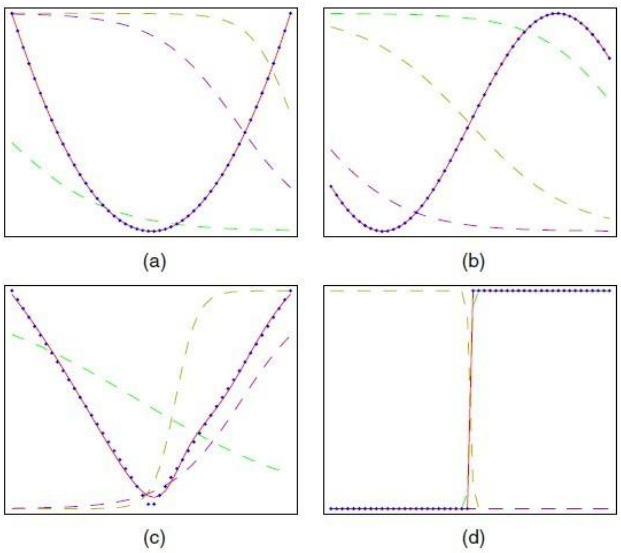
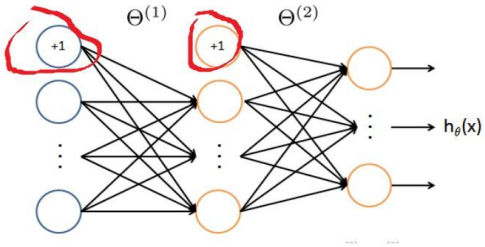


Figure 5.3 Illustration of the capability of a multilayer perceptron to approximate four different functions comprising (a) $f(x) = x^2$, (b) $f(x) = \sin(x)$, (c), $f(x) = |x|$, and (d) $f(x) = H(x)$ where $H(x)$ is the Heaviside step function. In each case, $N = 50$ data points, shown as blue dots, have been sampled uniformly in x over the interval $(-1, 1)$ and the corresponding values of $f(x)$ evaluated. These data points are then used to train a two-layer network having 3 hidden units with 'tanh' activation functions and linear output units. The resulting network functions are shown by the red curves, and the outputs of the three hidden units are shown by the three dashed curves.

第一部分神经网络的定义就这么多，大家有没有什么问题啊？
==========讨论==========
阳阳(236995728) 19:20:43
输入层到隐藏层使用的激活函数与隐藏层到输出层的激活函数是否要一致？
网神(66707180) 19:21:11
两层激活函数不一定一致，输入层到隐藏层 经常是 sigmoid 函数。 而隐藏层到输出层，根据应用不同，选择不同.
牧云(1106207961) 19:22:30
一般的算法，比如神经网络，都有分类和拟合的功能，分类和拟合有共同点吗？为什么能拟合，这个问题我没有找到地方清晰的解释。
独孤圣者(303957511) 19:23:47

拟合和分类，在我看来实际上是一样的，分类无非是只有 2 个或多个值的拟合而已。2 个值的拟合，被称为二值分类
ant/wxony(824976094) 19:24:30
不是吧，svm 做二值分类，就不是以拟合为目标。二值拟合可以用做分类。
独孤圣者(303957511) 19:29:42
可以作为概率来理解吧，我个人认为啊，sigmoid 函数，是将分类结果做一个概率的计算。
阳阳(236995728) 19:24:07
输入层到隐藏层 经常是 sigmoid 函数 那么也就是特征值变成了【0-1】之间的数了？
网神(66707180) 19:25:19
是的，我认为是.机器学习经常需要做特征归一化，也是把特征归一化到[0，1]范围。当然这里不只是归一化。
阳阳(236995728) 19:26:56
这里应该不是归一化的作用@网神
网神(66707180) 19:27:33
嗯，不是归一化，我觉得神经网络的这些做法，没有一个科学的解释。不像 SVM 每一步都有严谨的理论。
罗杰兔(38900407) 19:29:00
参数 W,b 是算出来的，还是有些技巧得到的。因为常听人说，神经网络难就难在调试参数上。
阳阳(236995728) 19:29:11
我觉得神经网络也是在学习新的特征 而这些特征比较抽象难于理解@牧云 @网神
网神(66707180) 19:29:48
对，我理解隐藏层的目的就是学习特征
阳阳(236995728) 19:30:24
是的 但是这些特征难于理解较抽象@网神
网神(66707180) 19:30:47
最后一个隐藏层到输出层之间，就是做分类或回归。前面的不同隐藏层，则是提取特征。
独孤圣者(303957511) 19:30:52
隐藏层的目的就是学习特征，这个表示赞同
网神(66707180) 19:31:33
后面讲到卷积网络时，可以很明显感受到，隐藏层的目的就是学习特征。
阳阳(236995728) 19:31:59
但是我不理解为什么隐藏层学习到的特征值介于【0-1】之间是合理的？是归一化的作用？
牧云(1106207961) 19:32:47
归一化映射
Wilbur_中博(1954123) 19:33:59
要归一化也是对每一个输入变量做吧。。各个输入变量都组合在一起了，归一化干嘛。我理解就是一个非线性变换，让神经网络具有非线性能力。sigmoid 之外，其他非线性变换用的也蛮多的。和归一化没什么关系。不一定要在[0, 1]之间。
独孤圣者(303957511) 19:39:50
这个解释我很同意，tanh 函数也经常替换 sigmoid 函数，在神经网络中普遍使用。
网神(66707180) 19:39:59
嗯，让 NN 具有非线性能力是主要原因之一。
牧云(1106207961) 19:40:03



这个 1 要的干嘛

网神(66707180) 19:40:18

这个是偏置项，y = wx+b， 那个+1 就是为了把 b 合入 w,变成 y = w x

罗杰兔(38900407) 19:41:36

Ng 好象讲 b 是截距

网神(66707180) 19:43:16

截距和偏执项是一个意思。 放在坐标系上，b 就是截距

阳阳(236995728) 19:40:33

$$y_k(\mathbf{x},\mathbf{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right)$$ 它可以逼近任意的非线性函数吗@网神

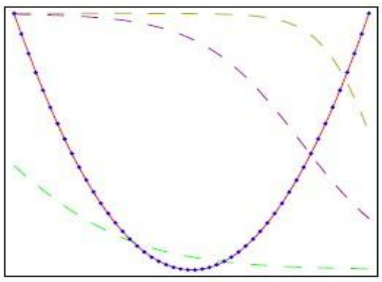
网神(66707180) 19:42:13

书上没说可以逼近任意曲线，说的是逼近很多曲线

阳阳(236995728) 19:42:58

它到底是怎么逼近的啊@网神 我想看看多个函数叠加的？

网神(66707180) 19:44:30



就是多个函数叠加。这个图上那个抛物线就是最终的曲线，由另外三条曲线叠加而成，另外那三条，每条是一个隐藏节点生成的曲线。

阳阳(236995728) 19:45:05

这三条曲线用的函数形式是不是一样的

zeno(117127143) 19:45:31

理解 nn 怎么逼近 xor 函数，就知道怎么非线性的了

HX(458728037) 19:45:56

问一句，你现在讲的是 BP 网络吧？

网神(66707180) 19:46:07

是 BP 网络

==========讨论结束==========

网神(66707180) 19:46:59

接下来是神经网络的训练。神经网络训练的套路与第三，四章相同：确定目标变量的分布函数和似然函数，取负对数，得到 error 函数，用梯度下降法求使 error 函数最小的参数 w 和 b。

NN 的应用有：回归、binary 分类、K 个独立 binary 分类、multiclass 分类。不同的应用决定了不同的激活函数、不同的目标变量的条件分布，不同的 error 函数.
首先，对于回归问题，输出变量 t 的条件概率符合高斯分布：

$$p(t|\mathrm{x},\mathrm{w}) = \mathcal{N}\left(t|y(\mathrm{x},\mathrm{w}),\beta^{-1}\right)$$

因此，给定一个训练样本集合，其似然函数是：

$$p(\mathbf{t}|\mathbf{X},\mathrm{w},\beta) = \prod_{n=1}^{N} p(t_n|\mathrm{x}_n,\mathrm{w},\beta).$$

error 函数是对似然函数取 负对数，得到：

$$\frac{\beta}{2}\sum_{n=1}^{N}\{y(\mathrm{x}_n,\mathrm{w}) - t_n\}^2 - \frac{N}{2}\ln\beta + \frac{N}{2}\ln(2\pi)$$

最小化这个 error 函数，只需要最小化第一项，所以回归问题的 errro 函数定义是：

$$E(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(\mathrm{x}_n,\mathrm{w}) - t_n\}^2$$

回归问题的隐藏层到输出层的激活函数是 identity 函数，也就是 $y_k = a_k$

ak 就是对隐藏层节点的线性组合：

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}$$

这样有了 error 函数，就可以同梯度下降法求极值点.
再说说 binary 分类和 multiclass 分类的 error 函数：
binary 分类的目标变量条件分布式伯努利分布： $p(t|\mathrm{x},\mathrm{w}) = y(\mathrm{x},\mathrm{w})^t\left\{1 - y(\mathrm{x},\mathrm{w})\right\}^{1-t}$
这样，通过对似然函数取负对数，得到 error 函数如下：

$$E(\mathrm{w}) = -\sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}$$

另外 binary 分类，隐藏层到输出层的激活函数是 sigmoid 函数。对 multiclass 问题，也就是结果是 K 个互斥的类别中的一个.类似思路可以得到其 error 函数.这里就不说了。书上都有.
知道了 error 函数，和激活函数，这里先做一个计算，为后面的 BP 做准备，将 E(w)对激活 ak 求导，可得：

$$\frac{\partial E}{\partial a_k} = y_k - t_k$$

这里有意思的是，不管是回归、binary 分类还是 multiclass 分类，尽管其 error 函数不同，这个求导得出的结果都是 yk-tk
回归问题这个求导很明显，下式中，y = a, 所以对 a 求导就是 y - t.

$$E(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(\mathrm{x}_n,\mathrm{w}) - t_n\}^2$$

其他错误函数，需要推导一下，这里就不推了。有了 error 函数，用梯度下降法求其极小值点。

因为输入层到隐藏层的激活函数是非线性函数，所以 error 函数是非凸函数，所以可能存在很多个局部极值点。

两栖动物(9219642) 20:07:40

你这里说的回归问题是用神经网络拟合随机变量？

网神(66707180) 20:07:56

是的，就是做线性回归要做的事.只是用神经网络的方法，可以拟合更复杂的曲线。这是我的理解。

梯度下降法的思路就是对 error 函数求导，然后按下式对参数作调整：

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)})$$

一直到导数为 0，就找到了极值点.

这是基本思路，实际上很多算法，如共轭梯度法，拟牛顿法等，可以更有效的寻找极值点。因为梯度下降是机器学习中求极值点的常用方法，这些算法不是本章的内容，就不讲了，有专门的章节讲这些算法。

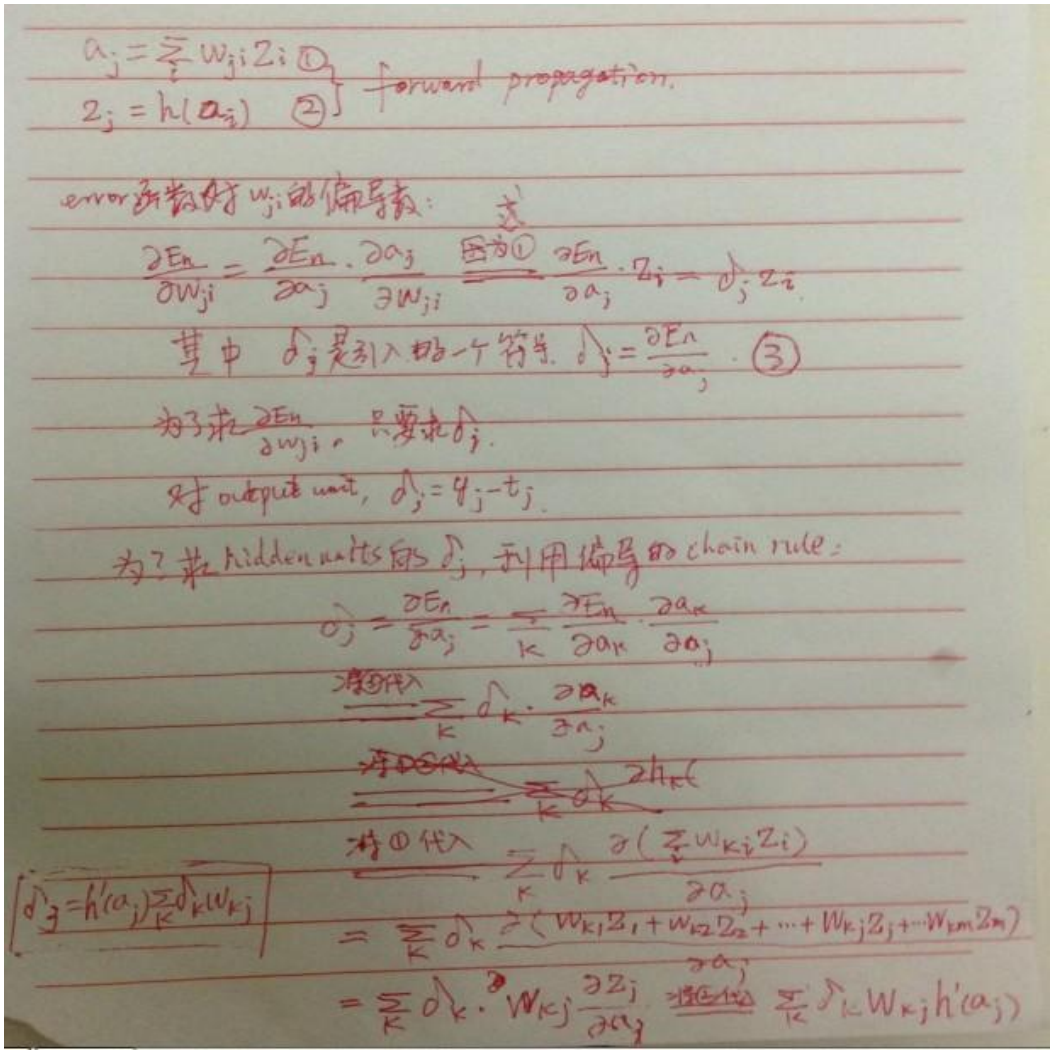
这里要讲的是 error 函数对参数 w 的求导方法，也就是 back propagation 后向传导法。

可以高效的计算这个导数。

这个导数的计算利用的 求导数的链式法则，

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}}.$$

把 E 对 w 的求导，转换成 E 对 activation a 的求导 和 a 对 w 的求导. 这个地方因为涉及的式子比较多，我在纸上写了一下，我整个贴上吧：



大家看看，可以讨论一下
阳阳(236995728) 20:21:30
error function 是负对数似然函数吧
网神(66707180) 20:21:38
是的
天上的月亮(785011830) 20:22:35
更新 hidden 层的 w，为什么利用偏导的 chain rule 呢？
网神(66707180) 20:22:49
最主要的结论就是：

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}}.$$

对于输出层的节点，

$$\frac{\partial E_n}{\partial a_j} = y_k - t_k$$

对于隐藏层的节点，

$$\frac{\partial E_n}{\partial a_j} = h'(a_j)\sum_k w_{kj}\delta_k$$

E 对隐藏层节点 w 的导数，通过上式，就由输出层节点的导数 求得了.
有了上面求偏导数的方法，整个训练过程就是：
1.输入一个样本 Xn,根据 forward propagation 得到每个 hidden 单元和 output 单元的激活值 a.

2.计算每个 output 单元的偏导数 δ_k

3.根据反向传导公式，计算每个 hidden 单元的 δ_j

4.计算偏导数。
5.用一定的步长更新参数 w。
循环往复，一直找到满意的 w。
独孤圣者(303957511) 20:32:09
BP 时，是不是每个样本都要反馈一次？
网神(66707180) 20:33:32
应该是每个样本都反馈一次，batch 也是.batch 形式的每个样本反馈一次，将每个样本的偏导数求和，如下式，得到所有样本的偏导数，然后做一次参数 w 的调整.

$$\frac{\partial E}{\partial w_{ji}} = \sum_n \frac{\partial E_n}{\partial w_{ji}}.$$

接下来就讲正则化，神经网络参数多，训练复杂，所以正则化方法也被研究的很多，书上讲了很多种的正则化方法. 第一个，就是在 error 函数上，加上正则项。
第三章回归的正则项如下：

$$\widetilde{E}(\mathbf{w}) = E(\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}. \qquad (5.112)$$

后面那一项对过大的 w 做惩罚。

但是对神经网络，正则项需要满足 一个缩放不变性，所以正则项的形式如下：

$$\widetilde{E}(\mathbf{w}) = E(\mathbf{w}) + \frac{\lambda_1}{2}\sum_{w\in\mathcal{W}_1} w^2 + \frac{\lambda_2}{2}\sum_{w\in\mathcal{W}_2} w^2$$

这个形式怎么推导出来的，大家看书吧，这里不说了。

书上讲到的第二种正则化方法是 early stop，我理解的思路就是 在训练集上每调整一次 w, 得到新的参数，就在验证集上验证一下。 开始在验证集上的 error 是逐渐下降的，后来就不下降了，会上升。那么在开始上升之前，就停止训练，那个参数就是最佳情况。

忘了说一个正则化注意对象了，就是隐藏节点的数量 M。这个 M 是预先人为确定的，他的大小决定了整个模型的参数多少。所以是影响 欠拟合还是过拟合的 关键因素。

monica(909117539) 20:42:38

开始上升是出现了过拟合么？节点数量和层数都是怎样选择的呀？

网神(66707180) 20:42:56

开始上升就是过拟合了，是的，在训练集上，error 还是下降的，但验证集上已经上升了。节点数量 M 是人为确定，我想这个数怎么定，是 NN 调参的重点。

monica(909117539) 20:44:36

：）

网神(66707180) 20:44:38

ML 牛的人，叫做 调得一手好参，我觉得这里最能体现。

这个图是不同的 M 值对拟合情况的影响：

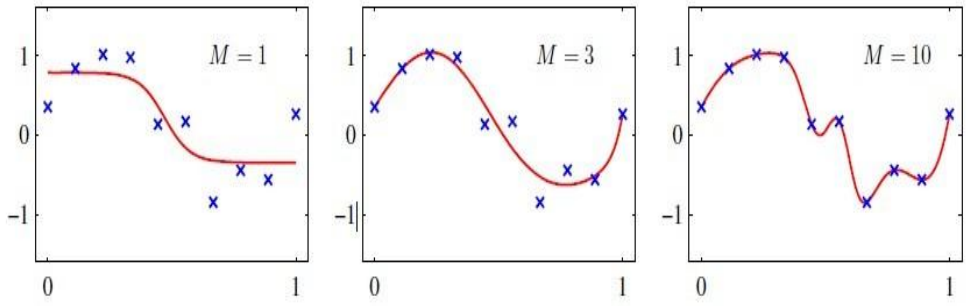


Figure 5.9 Examples of two-layer networks trained on 10 data points drawn from the sinusoidal data set. The graphs show the result of fitting networks having $M = 1, 3$ and 10 hidden units, respectively, by minimizing a sum-of-squares error function using a scaled conjugate-gradient algorithm.

另外，因为 NN 又不是凸函数，导致 train 更麻烦

monica(909117539) 20:47:01

嗯，这个能理解，前面模型太简单，欠拟合了，后面模型太复杂，就过拟合了。

网神(66707180) 20:47:08

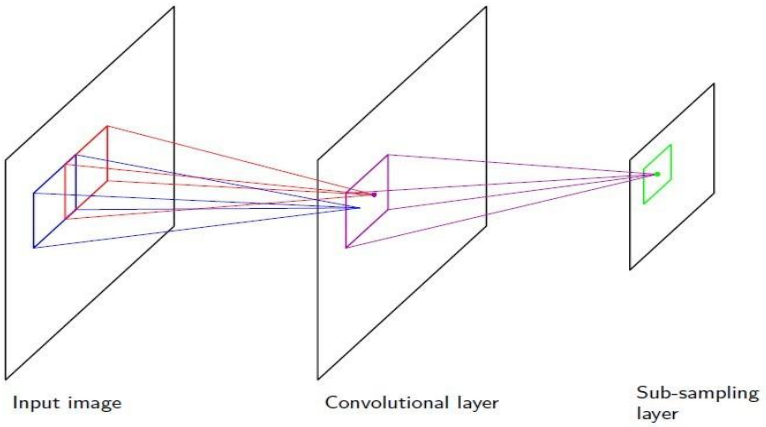
需要尝试不同的 M 值，对于每一个 M 值，又需要 train 多次，每次随即初始化 w 的值，然后尝试找到不同的局部极值点。

上面说了三种正则化方法。接下来，因为神经网络在图像识别中，应用广泛，所以图像识别对正则化有些特殊的需求，就是 平移、缩放、旋转不变性。不知道这个需求在其他应用中，是否存在，例如 NLP 中，反正书上都是以图像处理为例讲的.

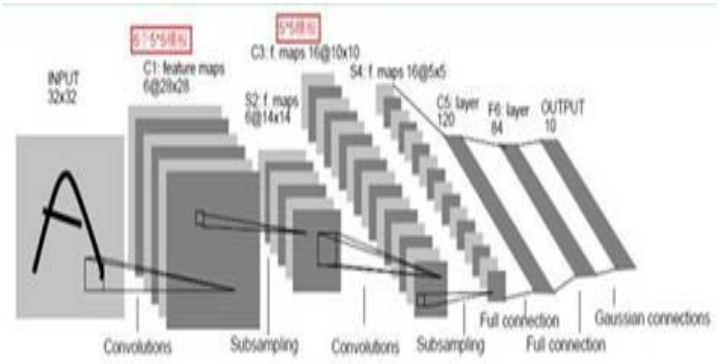
这些不变形，就是图像中的物体位置移动、大小变化、或者一定程度的旋转， 对 ouput 层的结果应该没有影响。

这里主要讲卷积神经网络。这个东东好像很有用，我看在 Ng 的 deep learning 教程中也重点讲了，当图像是大图时，如 100x100 像素以上，卷积网络可以大大减少参数 w 的个数.

卷积网络的 input 层的 unit 是图像的像素值.卷积网络的 hidden 层由卷积层和子采样层组成，如图：



一个卷及网络可能包括多个卷积层和子采样层，如下图有两对卷积/子采样层：



卷积层的 units 被划分成多个 planes，每个 plane 是一个 feature map，一个 feature map 里的一个 unit 只连接 input 层的一个小区域，一个 feature map 里的所有 unit 使用相同的 weight 值。

卷积层的作用可以认为是提取特征。每个 plane 就是一个特征滤波器，通过卷积的方式提取图像的一种特征，过滤掉不是本特征的信息。比如提取 100 种特征，就会有 100 个 plane，上图中，提取 6 个特征，第二层有 6 个平面。

一个 plane 里面的一个 unit 与图像的一个子区域相连，unit 的值表示从该区域提取的一个特征值。共有多少个子区域，plane 里就有多少个 unit。

以上图为例，输入图像是 32x32 像素，所以 input 层有 32x32 个 unit. 取子区域的大小为 4x4，，那么共有 28x28 个子区域。每个子区域提取得到 6 个特征，对应第二层有 6 个 plane， 每个 plane 有 28x28 个 unit.

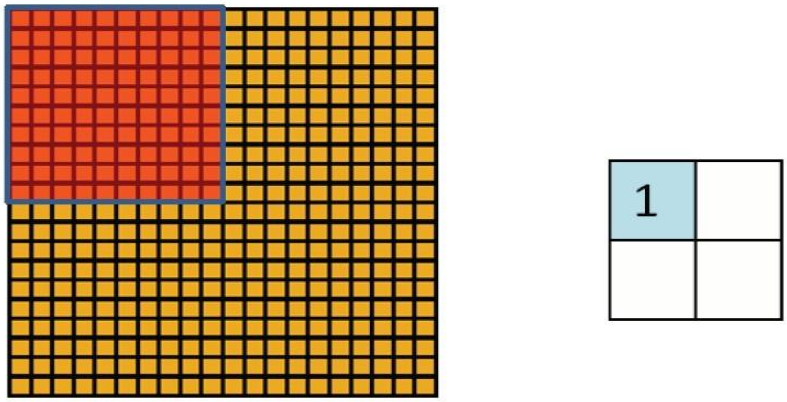
这样一个 plane 里的一个 unit 与输入层的 4x4 个 unit 相连，有 16 个权重 weight 值，这个 plane 里的其他 unit 都公用 16 个 weight 值.

这样，6 个平面，共有 16x6=96 个 weight 参数和 6 个 bias 参数.

下面说子采样层，上面说道，6 个平面，每个平面有 28x28 个 unit，所以第二层共有 6x28x28 个 unit. 这是从图像中提取出来的特征，作为后一层的输入，用于分类或回归。

但实际中，6 个特征远远不够，图像也不一定只有 32x32 像素。假如从 96x96 像素中提取 400 个特征，子区域还是 4x4，则第二层的 unit 会有 400 x (96-4) x (96-4) = 300 多万。 超过 300 多万输入的分类器很难训练，也容易过拟合。

这就通过子采样来减少特征。子采样就是把卷积层的一个平面划分成几部分，每个部分取最大值或平均值，作为子采样层的一个 unit 值.如下图：



左边表示卷积层的一个 plane，右边是这个 plane 的子采样结果。一个格子表示一个 unit。
把左边整个 plane 分成 4 部分，互相不重合，每个部分取最大值或平均值，作为右边子采样层的一个 unit。这样子采样层一个 plane 只有 4 个 unit。
通过卷积和子采样：得到的特征作为下一层分类或回归的输入。
主要好处：
a. 这样输入层的各种变形在后面就会变得不敏感。如果有多对 卷积/子采样，每一对就将不变形更深入一层。
b. 减少参数的个数。就这么多了，大家慢慢看。
阳阳(236995728) 21:01:30
关于卷积神经网络的材料有吗？@网神
阿邦(1549614810) 21:01:44
问个问题，cnn 怎么做的 normalization？
网神(66707180) 21:03:26
卷积网络的材料，我看了书上的，还有 Andrew Ng 的 deep learning 里讲的，另外网上搜了一篇文章这里我贴下链接。
罗杰兔(38900407) 21:04:19
不大明白为什么可以采样，怎么保证采样得到的信息正好是我们需要的
阿邦(1549614810) 21:05:01
采样是 max pooling，用于不变性
网神(66707180) 21:05:02
http://ibillxia.github.io/blog/2013/04/06/Convolutional-Neural-Networks/
牧云(1106207961) 21:05:02
采样是随机的吗？
网神(66707180) 21:06:40
传了个 deep learning 教程。这是 Andrew Ng 上课的 notes, 网上一些人众包翻译的 ,我觉得翻译的不错。里面讲了神经网络、卷积网络，和其他深度网络的东西。
采样不是随机的。
牧云(1106207961) 21:08:04
卷积提取的特征具有稳定性吗
阿邦(1549614810) 21:08:56
据说要数据量很大才可以
网神(66707180) 21:09:03
卷积层和子采样层主要是 解决 不变性问题 和减少参数 w 数量问题，所以可以起到泛化作用.