deepshare.net
深度之眼

Deep Learning
深度学习中的正则化

导师：johnson

deepshare.net
深度之眼

深度学习中的正则化

Regularization in deep learning

深度学习中的正则化

Regularization in deep learning

deepshare.net
深度之眼

1. 参数范数惩罚
2. 作为约束的范数惩罚
3. 正则化和欠约束问题
4. 数据集增强
5. 噪声鲁棒性
6. 半监督学习
7. 多任务学习
8. 提前终止
9. 参数绑定和参数共享
10. 稀疏表示
11. Bagging和其他集成方法
12. Dropout

深度学习中的正则化

Regularization in deep learning

deepshare.net
深度之眼

正则化定义为“**对学习算法的修改——旨在减少泛化误差而不是训练误差**”。

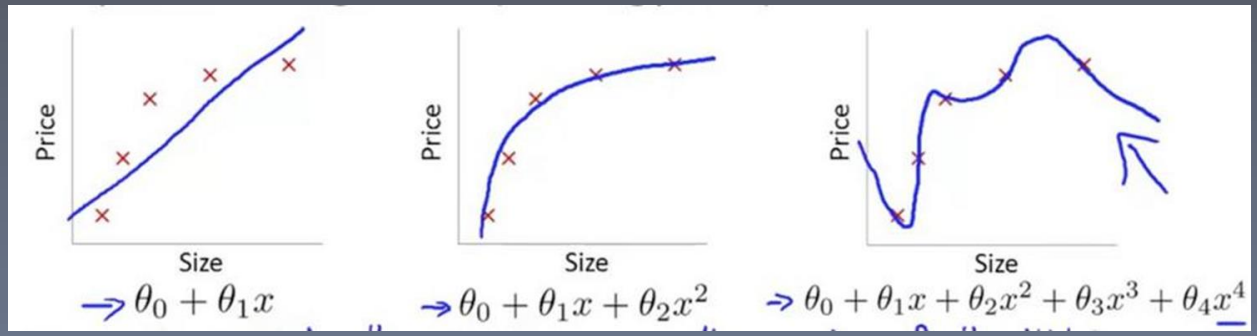
有些策略向机器学习模型添加限制参数值的额外约束。(参数惩罚)

有些策略向目标函数增加额外项来对参数值进行软约束。（约束惩罚）

在本章中我们将更详细地介绍正则化，重点介绍深度模型的正则化策略，
包括参数范数惩罚、约束惩罚、提前终止、Dropout等等。

深度学习中的正则化

Regularization in deep learning



几种训练情形：

（1）不管真实数据的生成过程---欠拟合，偏差大

（2）匹配真实数据的生成过程---刚刚好

（3）不止真实数据的生成过程，还包含其他生成过程---过拟合，方差大

正则的目标：

从（3）--->（2），偏差换方差，提升泛化能力

注：

永远不知道训练出来的模型是否包含数据生成过程（因为很难知道数据生成的物理规律，比如人脸识别，目前还不知道人脑的机制）！

深度学习应用领域极为复杂，图像、语音、文本等，生成过程难以琢磨

事实上，最好的模型总是适当正则化的大型模型

参数范数惩罚

Parameters norm punishment

通常只惩罚权重W，不管b——b是单变量，且容易过拟合

θ=(W; b)≈(W)

$$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta})$$

α是惩罚力度，Ω是正则项。

最常见参数范数惩罚:

- L2参数正则化
- L1参数正则化

L2参数正则化

L2 parameter regularization

L2参数正则化（也称为岭回归、Tikhonov正则）通常被称为权重衰减（weight decay），是通过向目标函数添加一个正则项 $\Omega(\theta) = \frac{1}{2}||\omega||_2^2$ 使权重更加接近原点

目标函数：

$$\tilde{J}(\omega; X, y) = J(\omega; X, y) + \frac{\alpha}{2}\omega^T\omega$$

计算梯度

$$\nabla_\omega \tilde{J}(\omega; X, y) = \nabla_\omega J(\omega; X, y) + \alpha\omega$$

更新权重

$$\omega \leftarrow \omega - \epsilon(\alpha\omega + \nabla_\omega J(\omega; X, y)) = (1 - \epsilon\alpha)\omega - \epsilon\nabla_\omega J(\omega; X, y)$$

从上式可以看出，加入权重衰减后会导致学习规则的修改，即在每步执行梯度更新前先收缩权重（乘以 $(1 - \epsilon\alpha)$）。

L2参数正则化

L2 parameter regularization

L2 参数正则化效应示意图:

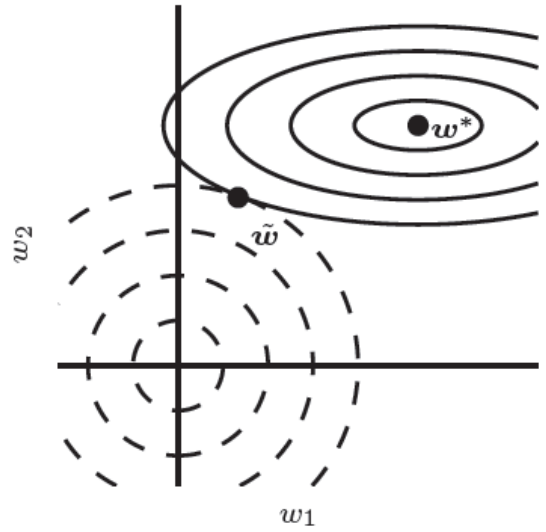


图 7.1: L^2（或权重衰减）正则化对最佳 $\boldsymbol{w}$ 值的影响。实线椭圆表示没有正则化目标的等值线。虚线圆圈表示 L^2 正则化项的等值线。在 $\tilde{\boldsymbol{w}}$ 点，这两个竞争目标达到平衡。目标函数 J 的 Hessian 的第一维特征值很小。当从 $\boldsymbol{w}^*$ 水平移动时，目标函数不会增加得太多。因为目标函数对这个方向没有强烈的偏好，所以正则化项对该轴具有强烈的影响。正则化项将 w_1 拉向零。而目标函数对沿着第二维远离 $\boldsymbol{w}^*$ 的移动非常敏感。对应的特征值较大，表示高曲率。因此，权重衰减对 w_2 的位置影响相对较小。

说明： 只有在显著减小目标函数方向上的参数会保留得相对完好。 在无助于目标函数减小的方向上改变参数不会显著增加梯度。 这种不重要方向对应的分量会在训练过程中因正则化而衰减掉。

L1参数正则化

L1 parameter regularization

形式地，对模型参数w的L1正则化被定义为：

$$\Omega(\theta) = ||w||_1 = \sum_i |w_i|,$$

目标函数：

$$\tilde{J}(\omega; X, y) = J(\omega; X, y) + \alpha\|\omega\|_1$$

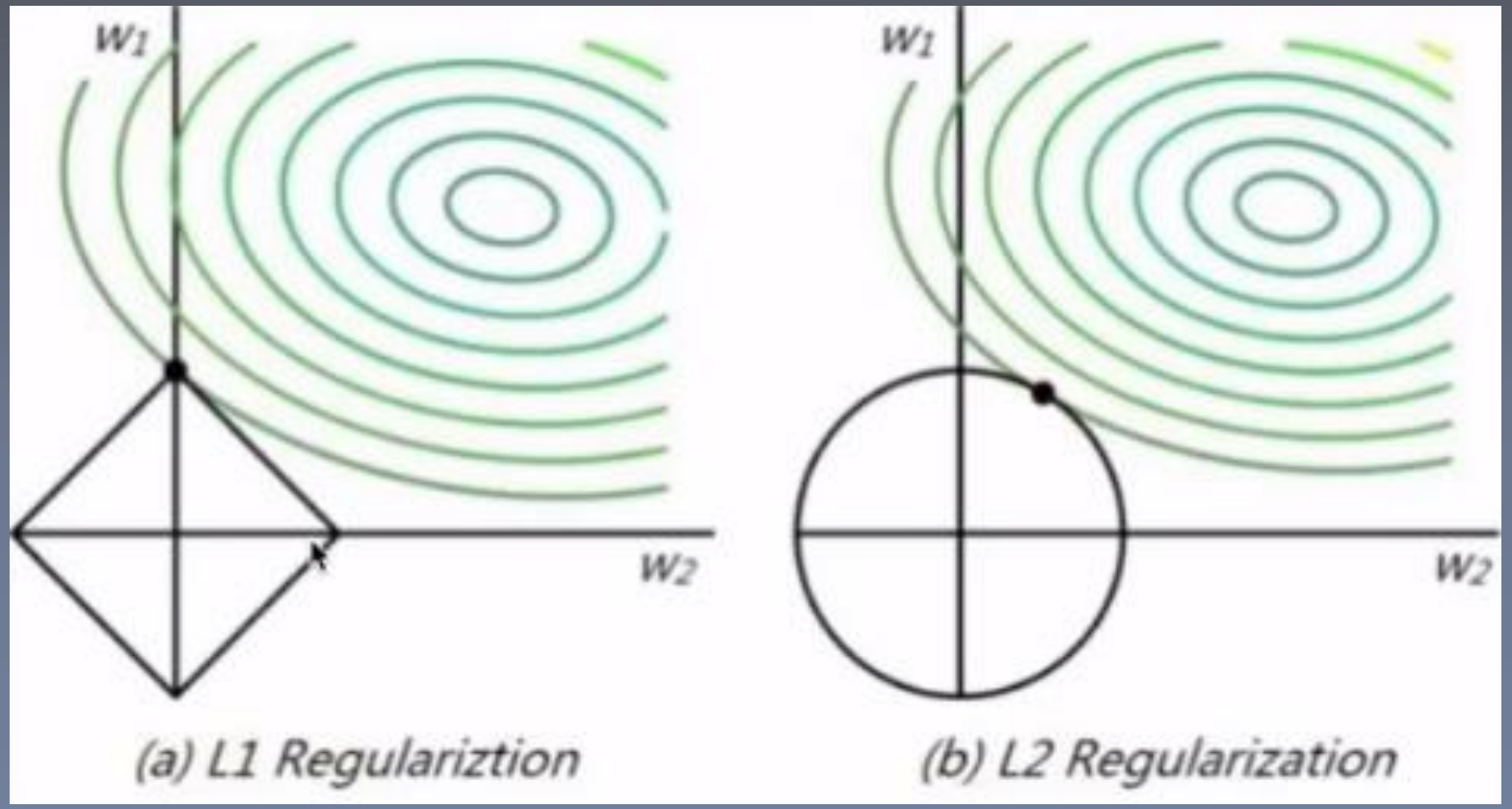
计算梯度

$$\nabla_\omega \tilde{J}(\omega; X, y) = \nabla_\omega J(\omega; X, y) + \alpha sgn(\omega)$$

L2正则化 vs. L1正则化

L2 vs. L1

总结 L2 与 L1 正则化



作为约束的范数惩罚

Norm punishment as constraint

deepshare.net
深度之眼

考虑经过参数范数正则化的代价函数：

$$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta}). \tag{7.25}$$

回顾第 4.4 节我们可以构造一个广义 Lagrange 函数来最小化带约束的函数，即在原始目标函数上添加一系列惩罚项。每个惩罚是一个被称为 **Karush–Kuhn–Tucker**（Karush–Kuhn–Tucker）乘子的系数以及一个表示约束是否满足的函数之间的乘积。如果我们想约束 $\Omega(\boldsymbol{\theta})$ 小于某个常数 k，我们可以构建广义 Lagrange 函数

$$\mathcal{L}(\boldsymbol{\theta}, \alpha; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \alpha(\Omega(\boldsymbol{\theta}) - k). \tag{7.26}$$

这个约束问题的解由下式给出

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \max_{\alpha, \alpha \geq 0} \mathcal{L}(\boldsymbol{\theta}, \alpha). \tag{7.27}$$

正则化和欠约束问题

Regularization and under constrained problems

机器学习中许多线性模型，如线性回归和PCA，都依赖与矩阵 X^TX 求逆，如果 X^TX 不可逆，这些方法就会失效。这种情况下，正则化的许多形式对应求逆 $X^TX+\alpha I$ ，这个正则化矩阵是可逆的。大多数正则化方法能够保证应用于欠定问题的迭代方法收敛。

X^TX不一定可逆（奇异），导致无法求逆（PCA）

解决：加正则，X^TX--->$X^TX+\alpha I$（一定可逆），

说明：α--阿尔法，I--大写的i，即单位阵。

大多数正则化能保证欠定（不可逆）问题的迭代方法收敛

注：伪逆 $\boldsymbol{X}^{+}=\lim_{\alpha \searrow 0}\left(\boldsymbol{X}^{\top} \boldsymbol{X}+\alpha \boldsymbol{I}\right)^{-1} \boldsymbol{X}^{\top}$

数据集增强

Data Augmentation

让机器学习模型泛化得更好的最好办法是使用更多的数据进行训练。 当然，在实践中，我们拥有的数据量是很有限的。 解决这个问题的一种方法是创建假数据并添加到训练集中。方式包括加入特定噪声（如高斯噪声），做一定的几何变换等等。

让机器学习模型泛化得更好的最好办法是使用更多的数据进行训练，因此需要在有限的数据中创建假数据并添加到训练集中。数据集增强在对象识别领域是特别有效的方法。

- 数据集的各种变换，如对图像的平移、旋转和缩放。
- 在输入层注入噪声，也可以看作数据集增强的一种方法（如去噪自编码器）。通过将随机噪声添加到输入再进行训练能够大大改善神经网络的健壮性。

噪声鲁棒性

Noise-robustness

大多数数据集的y标签都有一定错误。 错误的y不利于最大化logp(y | x)。 避免这种情况的一种方法是显式地对**标签上的噪声进行建模**。

例如，我们可以假设，对于一些小常数ϵ，训练集标记y是正确的概率是1−ϵ，（以ϵ的概率）任何其他可能的标签也可能是正确的。 这个假设很容易就能解析地与代价函数结合，而不用显式地抽取噪声样本。

例如，标签平滑（label smoothing）通过把确切分类目标从0和1替换成ϵ/k−1和1−ϵ，正则化具有k个输出的softmax函数的模型。 标准交叉熵损失可以用在这些非确切目标的输出上。 使用softmax函数和明确目标的最大似然学习可能永远不会收敛——softmax函数永远无法真正预测0概率或1概率，因此它会继续学习越来越大的权重，使预测更极端。 使用如权重衰减等其他正则化策略能够防止这种情况。 标签平滑的优势是能够防止模型追求确切概率而不影响模型学习正确分类。

半监督学习

Semi-supervised learning

在半监督学习的框架下，P(x)产生的未标记样本和P(x,y)中的标记样本都用于估计P(y | x)或者根据x预测y。

在深度学习的背景下，半监督学习通常指的是学习一个表示 h=f(x)。学习表示的目的是使相同类中的样本有类似的表示。 无监督学习可以为如何在表示空间聚集样本提供有用线索。 在输入空间紧密聚集的样本应该被映射到类似的表示。 在许多情况下，新空间上的线性分类器可以达到较好的泛化。 这种方法的一个经典变种是使用主成分分析作为分类前（在投影后的数据上分类）的预处理步骤。

我们可以构建这样一个模型，其中生成模型P(x)或P(x,y)与判别模型P(y | x)共享参数，而不用分离无监督和监督部分。 我们权衡监督模型准则 −logP(y | x) 和无监督或生成模型准则（如−logP(x) 或−logP(x,y) ）。生成模型准则表达了对监督学习问题解的特殊形式的先验知识，即P(x)的结构通过某种共享参数的方式连接到P(y | x)。 通过控制在总准则中的生成准则，我们可以获得比纯生成或纯判别训练准则更好的权衡。

多任务学习

Multi-task learning

多任务学习是通过合并几个任务中的样例(可以视为对参数施加的软约束)来提高泛化的一种方式。当模型的一部分被多个额外的任务共享时，这部分将被约束为良好的值，通常会带来更好的泛化能力。

右图展示了多任务学习的一种普遍形式。不同的监督任务共享相同的输入 x 和中间表示层 $h^{(shared)}$ ，能学习共同的因素池。

从深度学习的观点看，底层的先验知识如下：能解释数据变化（在与之相关联的不同任务中观察到）的因素中，某些因素是跨两个或更多任务共享的

如：人脸识别和性别分类的任务

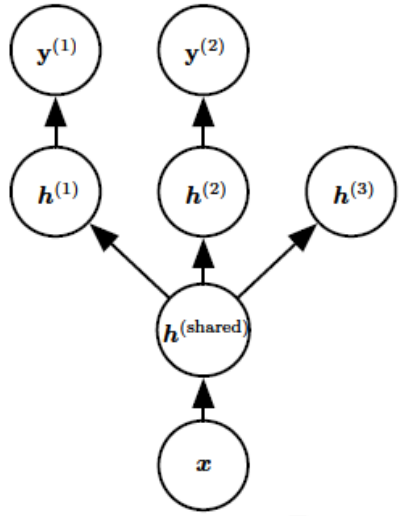


图 7.2: 多任务学习在深度学习框架中可以以多种方式进行，该图说明了任务共享相同输入但涉及不同目标随机变量的常见情况。深度网络的较低层（无论是监督前馈的，还是包括向下箭头的生成组件）可以跨这样的任务共享，而任务特定的参数（分别与从 $\boldsymbol{h}^{(1)}$ 和 $\boldsymbol{h}^{(2)}$ 进入和发出的权重）可以在共享表示 $\boldsymbol{h}^{(\text{shared})}$ 之上学习。这里的基本假设是存在解释输入 $\mathbf{x}$ 变化的共同因素池，而每个任务与这些因素的子集相关联。在该示例中，额外假设顶层隐藏单元 $\boldsymbol{h}^{(1)}$ 和 $\boldsymbol{h}^{(2)}$ 专用于每个任务（分别预测 $\mathbf{y}^{(1)}$ 和 $\mathbf{y}^{(2)}$），而一些中间层表示 $\boldsymbol{h}^{(\text{shared})}$ 在所有任务之间共享。在无监督学习情况下，一些顶层因素不与输出任务 ($\boldsymbol{h}^{(3)}$) 的任意一个关联是有意义的：这些因素可以解释一些输入变化但与预测 $\mathbf{y}^{(1)}$ 或 $\mathbf{y}^{(2)}$ 不相关。

提前终止

Early Stopping

deepshare.net
深度之眼

当训练有足够的表示能力甚至会过拟合的大模型时，我们经常观察到，训练误差会随着时间的推移逐渐降低但验证集的误差会再次上升。

这意味着我们只要返回使验证集误差最低的参数设置，就可以获得验证集误差更低的模型（并且因此有希望获得更好的测试误差）。 在每次验证集误差有所改善后，我们存储模型参数的副本。 当训练算法终止时，我们返回这些参数而不是最新的参数。 当验证集上的误差在事先指定的循环次数内没有进一步改善时，算法就会终止。

参数绑定和参数共享

Parameter Binding and Parameter Sharing

参数范数惩罚或约束是相对于固定区域或点，如 L2 正则化是对参数偏离 0 固定值进行惩罚。但有时我们需要对模型参数之间的相关型进行惩罚，使模型参数尽量接近或者相等。

参数共享：强迫模型某些参数相等

主要应用：卷积神经网络（CNN）

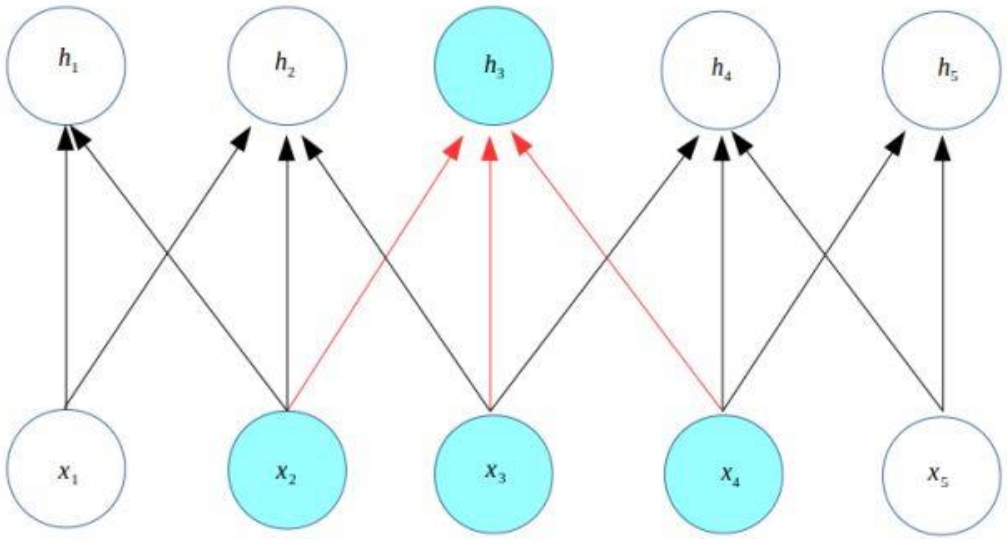
优点：显著降低了CNN模型的参数数量（CNN模型参数数量经常是千万量级以上），减少模型所占用的内存，并且显著提高了网络大小而不需要相应的增加训练数据。

稀疏表示

Sparse Representation

稀疏表示也是卷积神经网络经常用到的正则化方法。 L1 正则化会诱导稀疏的参数，使得许多参数为0；而稀疏表示是惩罚神经网络的激活单元，稀疏化激活单元。换言之，稀疏表示是使得每个神经元的输入单元变得稀疏，很多输入是0。

例如右图， h3 只依赖于上一层的3个神经元输入 x2 、 x3 、 x4 ， 而其他神经元到 h3 的输入都是0。



稀疏表示

Bagging和其他集成方法

Bagging and other integration approaches

deepshare.net
深度之眼

Bagging(bootstrap aggregating)是通过结合几个模型降低泛化误差的技术。主要想法是分别训练几个不同的模型，然后让所有模型表决测试样例的输出。这是机器学习中常规策略的一个例子,被称为模型平均(model averaging)。采用这种策略的技术被称为集成方法。

Bagging是一种允许重复多次使用同一种模型、训练算法和目标函数的方法。具体来说,Bagging涉及构造 k 个不同的数据集。每个数据集从原始数据集中重复采样构成,和原始数据集具有相同数量的样例。

Bagging和其他集成方法

Bagging and other integration approaches

模型平均是一个减少泛化误差的非常强大可靠的方法。例如我们假设有k个回归模型，每个模型误差是ϵ_i，误差服从零均值、方差为 v 、协方差为 c 的多维正态分布。则模型平均预测的误差为

$$\mathbb{E}\left[(\frac{1}{k}\sum_i \epsilon_i)^2\right] = \frac{1}{k^2}\mathbb{E}\left[\sum_i(\epsilon_i^2 + \sum_{i\neq j}\epsilon_i\epsilon_j)\right] = \frac{1}{k}v + \frac{k-1}{k}c$$

在误差完全相关即 c=v 的情况下，均方误差为 v ，模型平均没有帮助。在误差完全不相关即 c=0 时，模型平均的均方误差的期望仅为 1/kv 。这说明集成平方误差的期望随集成规模的增大而线性减少。

其他集成方法，如Boosting，通过向集成逐步添加神经网络，可以构建比单个模型容量更高的集成模型。

Bagging和其他集成方法

Bagging and other integration approaches

例子：

其他集成方法，如Boosting，通过向集成逐步添加神经网络，可以构建比单个模型容量更高的集成模型。

Dropout

deepshare.net
深度之眼

Dropout

Dropout可以被认为是集成大量深层神经网络的实用Bagging方法。但是Bagging方法涉及训练多个模型，并且在每个测试样本上评估多个模型。当每个模型都是一个大型神经网络时，Bagging方法会耗费很多的时间和内存。而Dropout则提供了一种廉价的Bagging集成近似，能够训练和评估指数级数量的神经网络。

Dropout：

- 集成大量深层网络的bagging方法
- 施加到隐含层的掩掩码噪声

Dropout

Dropout

示例：

2个输入，1个输出，2个隐含层

一共2^4=16种情形

问题：大部分没有输入，输入到输出的路径

网络越宽，这种问题概率越来越小

注：

不同于bagging，模型独立

dropout所有模型共享参数

推断：对所有成员累计投票做预测

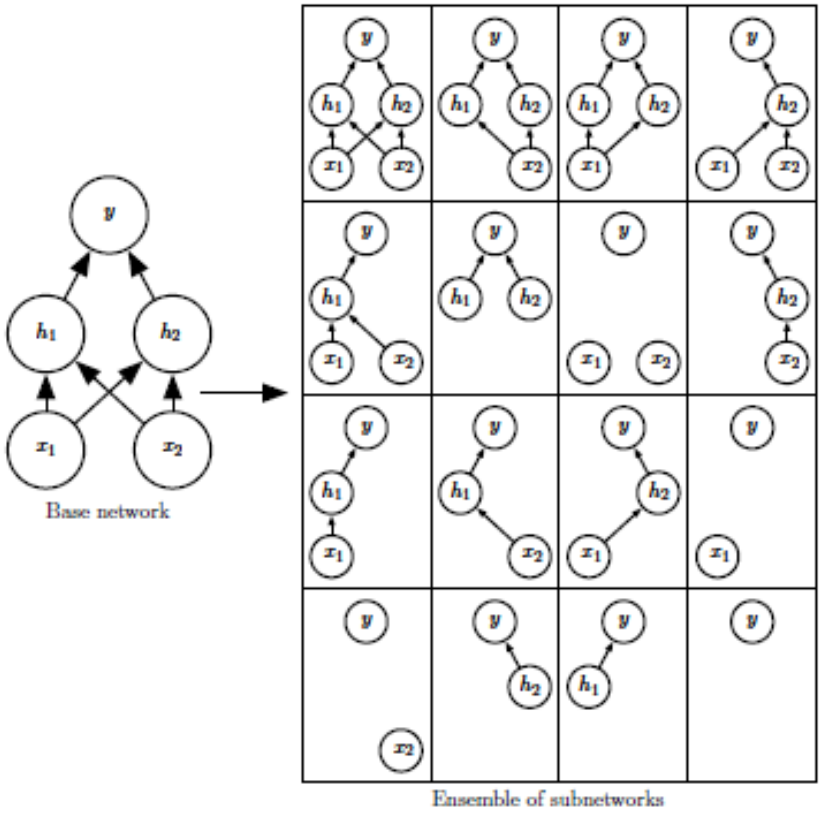


图 7.6: Dropout训练由所有子网络组成的集成，其中子网络通过从基本网络中删除非输出单元构建。我们从具有两个可见单元和两个隐藏单元的基本网络开始。这四个单元有十六个可能的子集。右图展示了从原始网络中丢弃不同的单元子集而形成的所有十六个子网络。在这个小例子中，所得到的大部分网络没有输入单元或没有从输入连接到输出的路径。当层较宽时，丢弃所有从输入到输出的可能路径的概率变小，所以这个问题不太可能在出现层较宽的网络中。

Dropout

deepshare.net
深度之眼

Dropout

效果:

Dropout比其他标准正则化方法更有效

权重衰减、过滤器范数约束、稀疏激活

可以跟其他形式正则一起使用

优点:

计算量小

不限制模型和训练过程

参考资料

References

星小环的AI读书会—深度学习系列05深度学习中的正则化
https://zhuanlan.zhihu.com/p/29487189

Deep Learning读书笔记3---深度学习中的正则化
https://blog.csdn.net/u012554092/article/details/77987797

深度学习Bible学习笔记：第七章 深度学习中的正则化

https://www.cnblogs.com/ariel-dreamland/p/9077793.html

深度学习入门基础——算法工程师带你读AI圣经《Deep Learning》

https://ke.qq.com/course/277276

deepshare.net

深度之眼

联系我们:

电话: 18001992849

邮箱: service@deepshare.net

Q Q: 2677693114

公众号

客服微信