卷积神经网络(CNN)防止过拟合的方法

因为数据量的限制以及训练参数的增多，几乎所有大型卷积神经网络都面临着过拟合的问题，目前常用的防止过拟合的方法有下面几种：

1、data augmentation

这点不需要解释太多，所有的过拟合无非就是训练样本的缺乏和训练参数的增加。一般要想获得更好的模型，需要大量的训练参数，这也是为什么 CNN 网络越来越深的原因之一，而如果训练样本缺乏多样性，那再多的训练参数也毫无意义，因为这造成了过拟合，训练的模型泛化能力相应也会很差。大量数据带来的特征多样性有助于充分利用所有的训练参数。data augmentation 的手段一般有：

1）收集更多数据

2）对已有数据进行 crop，flip，加光照等操作

3）利用生成模型（比如 GAN）生成一些数据。

2、weight decay(权重衰减)

常用的 weight decay 有 L1 和 L2 正则化，L1 较 L2 能够获得更稀疏的参数，但 L1 零点不可导。在损失函数中，weight decay 是放在正则项（regularization）前面的一个系数，正则项一般指示模型的复杂度，所以 weight decay 的作用是调节模型复杂度对损失函数的影响，若 weight decay 很大，则复杂的模型损失函数的值也就大。

2.1 L1 正则化

在原有代价函数基础上加一项：

$$\lambda|w|$$

假设原始代价函数是 C0，则代价函数公式变成：

$$C = C_0 + \frac{1}{n}\sum_w \lambda w .$$

梯度公式变成：

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n}\mathrm{sgn}(w).$$

在优化时使 weight 变得 sparse，最后只使用输入重要部分的 sparse 子集，对“噪声”具有不变性。即产生一个 sparse 模型，可以用于特征选择。

当一个特定的权重绝对值|w|很大时，L1 规范化权重缩小比 L2 要小；当一个特定的权重绝对值|w|很小时，L1 规范化权重缩小量比 L2 要大。最终的结果就是：L1 规范化趋向于将网络的权重在相对少量的高重要度连接上，而其他权值就会向 0 接近。

2.2 L2 正则化

在原有代价函数基础上加一项：

$$\frac{1}{2}\lambda w^2$$

假设原始代价函数是 C0，则代价函数公式变成：

$$C = C_0 + \frac{1}{2n}\sum_w \lambda w^2$$

2.2.1 为什么可以对权重进行衰减

我们对加入 L2 正则化后的代价函数进行推导，梯度公式变成：

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w$$
$$\frac{\partial C}{\partial b} = \frac{\partial C_0}{\partial b}.$$

可以发现 L2 正则化项对 b 的更新没有影响，但是对于 w 的更新有影响：

$$w \rightarrow w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta \lambda}{n} w$$
$$= \left(1 - \frac{\eta \lambda}{n}\right) w - \eta \frac{\partial C_0}{\partial w}.$$

在不使用 L2 正则化时，求导结果中 w 前系数为 1，现在 w 前面系数为 1-ηλ/n，因为 η、λ、n 都是正的，所以 1-ηλ/n 小于 1，它的效果是减小 w，**这也就是权重衰减（weight decay）的由来**。当然考虑到后面的导数项，w 最终的值可能增大也可能减小。

另外，需要提一下，对于基于 mini-batch 的随机梯度下降，w 和 b 更新的公式跟上面给出的有点不同。

$$w \rightarrow \left(1 - \frac{\eta \lambda}{n}\right) w - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial w}$$

$$b \rightarrow b - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial b}$$

对比上面 w 的更新公式，可以发现后面那一项变了，变成所有导数加和，乘以η再除以 m，m 是一个 mini-batch 中样本的个数。

2.2.2 权重衰减（L2 正则化）的作用

作用：L2 正则化使权值更加分散，更加小，尽量使用所有的输入（权值和所有输入相乘）而不是只用到一部分输入，模型 rubost 更强。可以防止模型过拟合。

思考：L2 正则化项有让 w 变小的效果，但是为什么 w 变小可以防止过拟合呢？

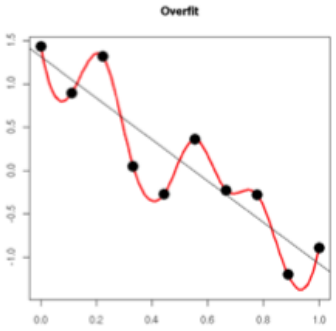
原理：如果参数分散不均匀，有的参数很大有的很小会使得几个输入特征就严重影响结果，几个特征占据了主要部分。比如，线性回归中，如果参数有的参数很大，只要数据偏移一点点，就会对结果很大影响。

（1）从模型的复杂度上解释：小的权重在某种程度上，意味着更低复杂性，也就对数据给出了一种更简单却更强大的解释。即数据的拟合更好（这个法则也叫做奥卡姆剃刀），而在实际应用中，也验证了这一点，L2 正则化的效果往往好于未经正则化的效果。更小的权重意味着网络的行为不会因为我们的随便改变一个输入而改变太大。这会让规范化的网络学习局部噪声的影响更加困难。对比看，大的权重的网络困难会因为输入的微小改变而产生比较大的行为改变。所以一个无规范化的网络可以使用大的权重来学习包含训练数据中的噪声的大量信息的复杂模型。简言之，规范化网络受限于根据训练数据中常见的模式来构造相对简单的模型，而能够抵抗训练数据中的噪声的特性影响。

关于更多可以参考:机器学习中的范数规则化之（一）L0、L1 与 L2 范数

（2）从数学方面的解释：过拟合的时候，拟合函数的系数往往非常大，为什么？如下图所示，过拟合，就是拟合函数需要顾忌每一个点，最终形成的拟合函数波动很大。在某些很小的区间里，函数值的变化很剧烈。这就意味着函数在某些小区间里的导数值（绝对值）非常大，由于自变量值可大可小，所以只有系数足够大，才能保证导数值很大。而正则化是通过约束参数的范数使其不要太大，所以可以在一定程度上减少过拟合情况。

内容来自：正则化方法：L1 和 L2 regularization、数据集扩增、dropout



3、权重约束

在深度学习中，批量归一化（batch normalization）以及对损失函数加一些正则项这两类方法，一般可以提升模型的性能。这两类方法基本上都属于权重约束，用于减少深度学习神经网络模型对训练数据的过拟合，并改善模型对新数据的性能。

在本教程中，使用 Keras API，用于向深度学习神经网络模型添加权重约束以减少过拟合。Keras API 支持权重约束，且约束可以按每层指定。

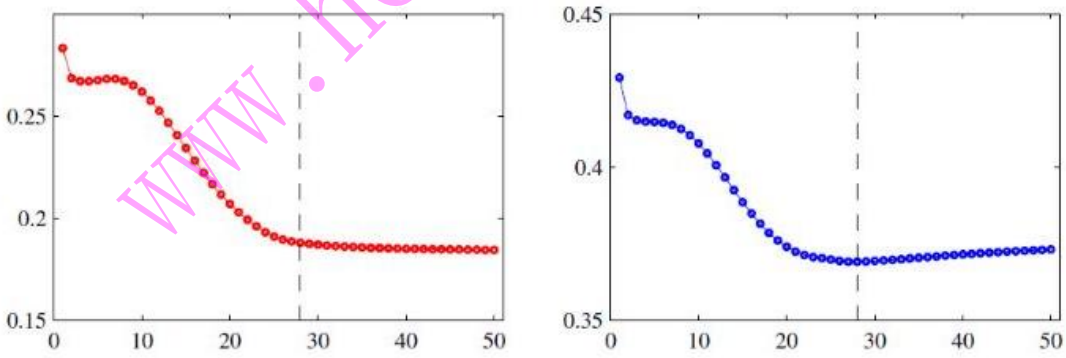
使用约束通常涉及在图层上为输入权重设置 `kernel_constraint` 参数，偏差权重设置为 `bias_constraint`。通常，权重约束方法不涉及偏差权重。

一组不同的向量规范在 `keras.constraints` 模块可以用作约束：

- 最大范数（max_norm）：强制权重等于或低于给定限制;
- 非负规范（non_neg）：强制权重为正数；
- 单位范数（unit_norm）：强制权重为 1.0；
- Min-Max 范数（min_max_norm）：强制权重在一个范围之间；

4、提前终止

提前停止其实是另一种正则化方法，就是在训练集和验证集上，一次迭代之后计算各自的错误率，当在验证集上的错误率最小，在没开始增大之前停止训练，因为如果接着训练，训练集上的错误率一般是会继续减小的，但验证集上的错误率会上升，这就说明模型的泛化能力开始变差了，出现过拟合问题，及时停止能获得泛化更好的模型。如下图（左边是训练集错误率，右图是验证集错误率，在虚线处提前结束训练）：



提前终止是一种很常用的缓解过拟合的方法,如在决策树的先剪枝的算法,提前终止算法,使得树的深度降低,防止其过拟合.

5、dropout

CNN 训练过程中使用 dropout 是在每次训练过程中随机将部分神经元的权重置为 0，即让一些神经元失效，这样可以缩减参数量，避免过拟合，关于 dropout 为什么有效，有两种观点：1）每次迭代随机使部分神经元失效使得模型的多样性增强，获得了类似多个模型 ensemble 的效果，避免过拟合 2）dropout 其实也是一个 data augmentation 的过程，它导致了稀疏性，使得局部数据簇差异性更加明显，这也是其能够防止过拟合的原因。关于

dropout 的解释可参考这篇博客。

6、Batch Normalization

在 Google Inception V2 中所采用,是一种非常有用的正则化方法,可以让大型的卷积网络训练速度加快很多倍,同事收敛后分类的准确率也可以大幅度的提高.

BN 在训练某层时,会对每一个 mini-batch 数据进行标准化(normalization)处理,使输出规范到 N(0,1)的正太分布,减少了 Internal convariate shift(内部神经元分布的改变),传统的深度神经网络在训练是,每一层的输入的分布都在改变,因此训练困难,只能选择用一个很小的学习速率,但是每一层用了 BN 后,可以有效的解决这个问题,学习速率可以增大很多倍.

6、参考链接

1. https://blog.csdn.net/leo_xu06/article/details/71320727(卷积网络防止过拟合的方法)
2. https://blog.csdn.net/qq_27248897/article/details/76933986（l1,l2 正则化）
3. https://www.cnblogs.com/bonelee/p/8993812.html
4. https://blog.csdn.net/program_developer/article/details/80867468 （权重衰减）
5. https://www.cnblogs.com/yunqishequ/p/10043298.html （权重约束）