# 深度学习训练流程和几种提升神经网络性能的方法

## 1953729 吴浩泽

## 深度学习训练流程：

一个成熟合格的深度学习训练流程至少具备以下功能：在训练集上进行训练；在验证集上进行验证；模型可以保存最优的权重，并读取权重；记录下训练集和验证集的精度，便于调参。简称为构建数据验证集、模型训练、模型加载和模型调参四个部分。

### 构建验证集

在机器学习模型（特别是深度学习模型）的训练过程中，模型是非常容易过拟合的。深度学习模型在不断的训练过程中训练误差会逐渐降低，但测试误差的走势则不一定。在模型的训练过程中，模型只能利用训练数据来进行训练，并不能接触到测试集上的样本，故需要构建验证数据集对模型进行验证。

通过曲线可以观察到拟合，过拟合，欠拟合的情况。其中防止过拟合的方法有：正则化，数据增强，重新冲洗数据，提前终止法，丢弃法。防止欠拟合的方法有：添加其他特征项，添加多项式特征，减少正则化参数。

### 模型训练和验证

1. 好好检查数据

训练神经网络的第一步是完全不接触任何神经网络代码，而是从彻底检查数据开始。此步骤至关重要。花时间去检查数据是一件比较重要的工作。因为数据中往往可能存在异常值，而且了解它们的分布可以有利于我们找到一个更好的模型。

2. 评估框架并得到一个并不完美的baseline

固定随机种子：始终使用固定的随机种子来确保两次运行代码时您将获得相同的结果。

验证损失：验证损失是否从正确的损失值开始。

设定一个好的初始化

3. 过度拟合

找到一个好的模型的方法有两个阶段：

首先获得一个足够大的模型以使其可以过度拟合（即专注于训练损失），然后适当地对其进行正则化（放弃一些训练损失以提高验证损失）。

选择模型：为了减少训练损失，您需要为数据选择合适的体系结构。

不要相信学习率衰减的默认值：如果您要重新使用其他领域的代码，请务必小心学习率。

4. 正则化

获取更多数据

数据扩充

使用预训练网络

坚持监督学习

减小输入维数

减小模型尺寸

减小批量大小

Dropout

5. 微调

随机网格搜索

超参数优化

6. 进一步提高精确率

模型集成

### 模型调参

深度学习原理少但实践性非常强，基本上很多的模型的验证只能通过训练来完成。同时深度学习有众多的网络结构和超参数，因此需要反复尝试。训练深度学习模型需要GPU的硬件支持，也需要较多的训练时间，如何有效的训练深度学习模型逐渐成为了一门学问。

深度学习有众多的训练技巧。与传统的机器学习模型不同，深度学习模型的精度与模型的复杂度、数据量、正则化、数据扩增等因素直接相关。所以当深度学习模型处于不同的阶段（欠拟合、过拟合和完美拟合）的情况下，可以知道可以什么角度来继续优化模型。

初步构建简单的CNN模型，不用特别复杂，跑通训练、验证和预测的流程。

简单CNN模型的损失会比较大，尝试增加模型复杂度，并观察验证集精度。

在增加模型复杂度的同时增加数据扩增方法，直至验证集精度不变。

## 几种提升神经网络性能的方法：

1. 数据增广

在不改变图像类别的情况下，增加数据量，能提高模型的泛化能力。

自然图像的数据增广方式包括很多，如常用的水平翻转（horizontally flipping），一定程度的位移或者裁剪和颜色抖动（color jittering）。此外还可以尝试多种操作的组合， 例如同时做旋转和随机尺度变换，此外还可以把每个patch中所有像素在HSV颜色空间中的饱和度和明度提升0.25-4次幂方，乘以0.7-1.4之间的一个因子，再加一个-0.1-0.1之间的值。同样可以在色调通道（H）对每张图片或patch的所有像素增加一个-0.1~0.1之间的值。

2.图像预处理

零均值化：数据有过大的均值可能导致参数的梯度过大，如果有后续的处理，可能要求数据零均值，比如PCA。零均值化并没有消除像素之间的相对差异，人们对图像信息的摄取通常来自于像素之间的相对色差，而不是像素值的高低。

白化：相当于在零均值化和归一化操作之间插入一个旋转操作，将数据投影到主轴上。一张图片经过白化后，可以认为每个像素之间是统计独立的。然而白化很少在卷积神经网络中使用，可能原因是图像信息本来就是依靠像素之间的相对差异来体现的，白化让像素间去相关，让这种差异变得不确定，损失了信息。

3.网络初始化

不要将参数全部初始化为0

用小的随机数初始化

方差归一化

4.不同正则化方法

L1正则化

这里，对于网络中的每一个权重w ，我们都会加上一个项λ|w|到目标函数中。L1正则化有一个非常有趣的属性，那就是它会使得权重向量w在优化期间变得稀疏（例如非常接近零向量）。 带有L1正则化项结尾的神经网络仅仅使用它的最重要的并且接近常量的噪声的输入的一个稀疏的子集。相比之下，最终的权重向量从L2正则化通常是分散的、小数字。在实践中，如果不关心明确的特征选择，可以预计L2正则化在L1的性能优越。

L2正则化

它可以通过将模型中所有的参数的平方级作为惩罚项加入到目标函数（objective）中来实现。也就是说，对网络中的每一个权重w ，我们将其项 12λw2 加入到目标函数中，其中λ是正则化的强度参数。在惩罚项公式的前面加上12是很常见的，这样做的原因是因为优化函数12λw2 求导的时候不至于前面产生一个常数项因子2，而只是λw 这样简单的形式。对L2正则化的直观的解释是，L2正则化对尖峰向量的惩罚很强，并且倾向于分散权重的向量。

最大范数约束

正规化的另一种形式是实施绝对上限的大小在每个神经元的权向量中，利用投影梯度下降来强制约束。在实践中，这对应于执行参数正常更新，然后执行夹紧约束的vec { w }每个神经元的权向量满足平行vec { w } parallel\_2 < c。典型的c值是3或4的订单。

5.集成多个深度网络

在机器学习中，在训练多个学习器并将它们进行组合来使用是一种前沿的学习方法。众所周知，集成方法通常在得到更高的精确性的时候相比于单个学习器是至关重要的。并且，集成方法已经在现实任务中取得了伟大的成功。在实际应用中，尤其是挑战和竞赛中，几乎所有的第一和第二名获胜者都使用集成。

集成技巧：

相同的模型，不同的初始化

使用交叉验证决定最优超参数，然后根据最好的超参数集训练多个方法，但是使用不同的随机初始化。这种方法的危险是模型的多样性仅仅取决于初始化。

交叉验证阶段的最优模型的发现

使用交叉验证决定最优超参数，然后选择少量几个效果最好的模型进行集成。这样改善了集成的多样性，但是他也有风险：例如局部最优。在实践中,这可以更容易执行，因为它不需要额外的培训交叉验证后的模型。事实上，可以直接选择几个最先进的深度模型从Caffe Model Zoo执行集成。

单个模型的不同检查点

如果训练的代价很高，有些人取得了有限的成功在不同的检查点的单一网络随时间(例如在每个阶段)和使用这些形成了一个整体。显然,这受制于某些缺乏多样性，但是在实践中仍然可以工作的很好。这种方法的优点是，非常简便。

### 互评人：2053512 孙博闻 互评成绩：8分