第十四章 超参数调整

> Markdown Revision 1;
> Date: 2018/10/25
> Editor: 乔成磊-同济大学
> Contact: qchl0318@163.com
> Updater: sjsdfg，王超锋

14.1 调试处理

关于训练深度最难的事情之一是你要处理的参数的数量，从学习速率到 Momentum（动量梯度下降法）的参数。如果使用 Momentum 或 Adam 优化算法的参数，也许你还得选择层数，也许你还得选择不同层中隐藏单元的数量，也许你还想使用学习率衰减。所以，你使用的不是单一的学习率。接着，当然你可能还需要选择 mini-batch 的大小。

结果证实一些超参数比其它的更为重要，我认为，最为广泛的学习应用是，学习速率是需要调试的最重要的超参数。

除了，还有一些参数需要调试，例如 Momentum 参数，0.9 就是个很好的默认值。我还会调试 mini-batch 的大小，以确保最优算法运行有效。我还会经常调试隐藏单元，我用橙色圈住的这些，这三个是我觉得其次比较重要的，相对于而言。重要性排第三位的是其他因素，层数有时会产生很大的影响，学习率衰减也是如此。当应用 Adam 算法时，事实上，我从不调试，和，我总是选定其分别为 0.9，0.999 和，如果你想的话也可以调试它们。

但希望你粗略了解到哪些超参数较为重要，无疑是最重要的，接下来是我用橙色圈住的那些，然后是我用紫色圈住的那些，但这不是严格且快速的标准，我认为，其它深度学习的研究者可能会很不同意我的观点或有着不同的直觉。

14.2 神经网络中一般包含哪些超参数

一般可以将超参数分为三类：网络参数、优化参数、正则化参数。

网络参数：可指网络层与层之间的交互方式（相加、相乘或者串接等）、卷积核数量和卷积核尺寸、网络层数（也称深度）和激活函数等。

优化参数：一般指学习率（learning rate）、批样本数量（batch size）、不同优化器的参数以及部分损失函数的可调参数。

正则化：权重衰减系数，丢弃法比率（dropout）

14.3 模型优化寻找最优解和正则项之间的关系

网络模型优化调整的目的是为了寻找到全局最优解（或者相比更好的局部最优解），而正则项又希望模型尽量拟合到最优。两者通常情况下，存在一定的对立，但两者的目标是一致的，即最小化期望风险。模型优化希望最小化经验风险，而容易陷入过拟合，正则项用来约束模型复杂度。所以如何平衡两者之间的关系，得的最优或者较优的解就是超参数调整优化的目的。

14.4 超参数的重要性顺序

首先，学习率，损失函数上的可调参数。在网络参数、优化参数、正则化参数中最重要的超参数可能就是学习率了。学习率直接控制着训练中网络梯度更新的量级，直接影响着模型的**有效容限能力**；损失函数上的可调参数，这些参数通常情况下需要结合实际的损失函数来调整，大部分情况下这些参数也能很直接的影响到模型的的有效容限能力。这些损失一般可分成三类，第一类辅助损失结合常见的损失函数，起到辅助优化特征表达的作用。例如度量学习中的Center loss，通常结合交叉熵损失伴随一个权重完成一些特定的任务。这种情况下一般建议辅助损失值不高于或者不低于交叉熵损失值的两个数量级；第二类，多任务模型的多个损失函数，每个损失函数之间或独立或相关，用于各自任务，这种情况取决于任务之间本身的相关性，目前笔者并没有一个普适的经验由于提供参考；第三类，独立损失函数，这类损失通常会在特定的任务有显著性的效果。例如RetinaNet中的focal loss，其中的参数 γ，α，对最终的效果会产生较大的影响。这类损失通常论文中会给出特定的建议值。

其次，批样本数量，动量优化器（Gradient Descent with Momentum）的动量参数β。批样本决定了数量梯度下降的方向。过小的批数量，极端情况下，例如batch size为1，即每个样本都去修正一次梯度方向，样本之间的差异越大越难以收敛。若网络中存在批归一化（batchnorm），batch size过小则更难以收敛，甚至垮掉。这是因为数据样本越少，统计量越不具有代表性，噪声也相应的增加。而过大的batch size，会使得梯度方向基本稳定，容易陷入局部最优解，降低精度。一般参考范围会取在(1:1024]之间，当然这个不是绝对的，需要结合具体场景和样本情况；动量衰减参数 β是计算梯度的指数加权平均数，并利用该值来更新参数，设置为 0.9 是一个常见且效果不错的选择；

最后，Adam优化器的超参数、权重衰减系数、丢弃法比率（dropout）和网络参数。在这里说明下，这些参数重要性放在最后**并不等价于这些参数不重要。**而是表示这些参数在大部分实践中**不建议过多尝试**，例如Adam优化器中的$\beta1，\beta2，\epsilon$，常设为 0.9、0.999、10−8就会有不错的表现。权重衰减系数通常会有个建议值，例如0.0005，使用建议值即可，不必过多尝试。dropout通常会在全连接层之间使用防止过拟合，建议比率控制在[0.2,0.5]之间。使用dropout时需要特别注意两点：一、在RNN中，如果直接放在memory cell中,循环会放大噪声，扰乱学习。一般会建议放在输入和输出层；二、不建议dropout后直接跟上batchnorm，dropout很可能影响batchnorm计算统计量，导致方差偏移，这种情况下会使得推理阶段出现模型完全垮掉的极端情况；网络参数通常也属于超参数的范围内，通常情况下增加网络层数能增加模型的容限能力，但模型真正有效的容限能力还和样本数量和质量、层之间的关系等有关，所以一般情况下会选择先固定网络层数，调优到一定阶段或者有大量的硬件资源支持可以在网络深度上进行进一步调整。

14.5 如何选择调试值?

14.6 为超参数选择合适的范围

14.7 如何搜索超参数?

最后，关于如何搜索超参数的问题，我见过大概两种重要的思想流派或人们通常采用的两种重要但不同的方式。

一种是你照看一个模型，通常是有庞大的数据组，但没有许多计算资源或足够的 CPU 和 GPU 的前提下，基本而言，你只可以一次负担起试验一个模型或一小批模型，在这种情况下，即使当它在试验时，你也可以逐渐改良。比如，第 0 天，你将随机参数初始化，然后开始试验，然后你逐渐观察自己的学习曲线，也许是损失函数 J，或者数据设置误差或其它的东西，在第 1 天内逐渐减少，那这一天末的时候，你可能会说，看，它学习得真不错。我试着增加一点学习速率，看看它会怎样，也许结果证明它做得更好，那是你第二天的表现。两天后，你会说，它依旧做得不错，也许我现在可以填充下 Momentum 或减少变量。然后进入第三天，每天，你都会观察它，不断调整你的参数。也许有一天，你会发现你的学习率太大了，所以你可能又回归之前的模型，像这样，但你可以说是在每天花时间照看此模型，即使是它在许多天或许多星期的试验过程中。所以这是一个人们照料一个模型的方法，观察它的表现，耐心地调试学习率，但那通常是因为你没有足够的计算能力，不能在同一时间试验大量模型时才采取的办法。

另一种方法则是同时试验多种模型，你设置了一些超参数，尽管让它自己运行，或者是一天甚至多天，然后你会获得像这样的学习曲线，这可以是损失函数 J 或实验误差或损失或数据误差的损失，但都是你曲线轨迹的度量。同时你可以开始一个有着不同超参数设定的不同模型，所以，你的第二个模型会生成一个不同的学习曲线，也许是像这样的一条（紫色曲线），我会说这条看起来更好些。与此同时，你可以试验第三种模型，其可能产生一条像这样的学习曲线（红色曲线），还有另一条（绿色曲线），也许这条有所偏离，像这样，等等。或者你可以同时平行试验许多不同的模型，橙色的线就是不同的模型。用这种方式你可以试验许多不同的参数设定，然后只是最后快速选择工作效果最好的那个。在这个例子中，也许这条看起来是最好的（下方绿色曲线）。

所以这两种方式的选择，是由你拥有的计算资源决定的，如果你拥有足够的计算机去平行试验许多模型，那绝对采用鱼子酱方式，尝试许多不同的超参数，看效果怎么样。但在一些应用领域，比如在线广告设置和计算机视觉应用领域，那里的数据太多了，你需要试验大量的模型，所以同时试验大量的模型是很困难的，它的确是依赖于应用的过程。但我看到那些应用熊猫方式多一些的组织，那里，你会像对婴儿一样照看一个模型，调试参数，试着让它工作运转。尽管，当然，甚至是在熊猫方式中，试验一个模型，观察它工作与否，也许第二或第三个星期后，也许我应该建立一个不同的模型（绿色曲线），像熊猫那样照料它，我猜，这样一生中可以培育几个孩子，即使它们一次只有一个孩子或孩子的数量很少。