调参流程

机器学习组论文分享-第3期

常坤亮

2019.4.16

问题?

- 代码永远不会崩溃、引发异常，甚至变慢。
- 网络持续训练，损失仍在不断减少。
- 几个小时后会收敛，但结果却很糟糕。

How to?

- 简单处开始
- 确认你的模型损失(loss)
- 检查中间输出和连接
- 诊断参数
- 跟踪你的工作

从简单开始

- 首先构建一个更简单的模型
  - 作为起点，构建一个具有单个隐藏层的小型网络，并验证一切正常，然后逐渐添加模型复杂性，同时检查模型结构的每个方面（层、参数等）是否有效。
- 在单个数据点上训练模型
  - 作为一种快速检查，可以使用一组或两组训练数据点来确认模型是否会产生过拟合。

神经网络应立即过拟合，训练精度为100%，验证准确度与您随机猜测的模型相当。如果模型不能在那些数据点上过拟合，那么要么数据集太小，要么有错误。

确认模型损失

- 损失适合于当前任务
- 损失函数都以正确的比例因子进行度量。
- 注意初始损失也很重要。

使用分类交叉熵损失进行多分类问题或使用焦点损失来解决类别不平衡问题）
如果您在网络中使用多种类型的损失，例如MSE、对抗式、L1、特性丢失，那么请确保所有损失都以相同的比例正确缩放。
如果您的模型是通过随机猜测开始的，请检查初始损失是否接近您的预期损失。

出于性能考虑寻找正确的损失。使用小参数初始化时，请确保获得预期的损失。最好先只检查数据损失（因此将正则化强度设置为零）。
例如，对于具有Softmax分类器的CIFAR-10，我们预期初始损失为2.302，因为我们期望每个类别的扩散概率为0.1（因为有10个类别），而Softmax损失是正确的类别的负的对数概率，所以：-ln(0.1) = 2.302

检查中间输出和连接

- 梯度更新的算式不正确
  - 如果梯度值是零，这可能意味着优化器中的学习率太小
- 未应用权重更新
- 消失或爆炸的梯度
  - 使用梯度检查，通过使用数值方法逼近梯度来检查这些错误。如果它接近计算的梯度，则反向传播实现是正确的

除了查看梯度更新的绝对值之外，还要确保监视每个层的激活幅度、权重和更新。例如，参数更新的幅度（权重和偏差）应为1-e3。
存在一种称为“死亡ReLU”或“消失梯度问题”的现象，其中ReLU神经元在学习其权重的大负偏差项后将输出零。那些神经元永远不会再在任何数据点上激活。

诊断参数

- 神经网络具有大量彼此相互作用的参数，使得优化变得困难。
- 批量大小（mini-batch） -希望批量大到足以准确估计误差梯度，但小到足以使mini-batch随机梯度下降(SGD)可以正则化网络。
  - 论文"On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima
- 学习率 - 学习率太低会导致收敛缓慢或陷入局部最小值的风险，而学习率太大会导致优化发散，因为存在跳过损失函数更深、但更窄部分的风险。
- 梯度下降方法
- http://ruder.io/optimizing-gradient-descent/

当使用较大批次时，通过泛化的能力衡量的模型的质量会降低。
众所周知，局部的最小值导致较差的泛化。相比之下，小批量方法始终如一地收敛到平面最小化，

诊断参数

- 批量标准化
- 正则化
  - 要注意的一个危险是正则化损失可能会压倒数据损失
- Dropout
  - 同BN一起使用需注意，BN必须在Dropout之后

在这种情况下，梯度将主要来自正则化（通常具有更简单的梯度表达式）。这可以掩盖数据损失梯度的错误实现
从理论上讲，我们发现网络从训练状态转移到测试时，Dropout会改变特定神经元的方差。
但是，BN将在测试阶段保持其整个学习过程累积的统计方差。
该方差的不一致性（我们将该方案命名为“方差偏移”）导致推理中不稳定的数值行为，
当在BN之前应用Dropout时，最终导致更多错误的预测。
批量规范化必须在Dropout之后，否则您将通过规范化统计传递信息。“

跟踪工作

- 记录实验的重要性!
- 模型的关键信息，如超参数、模型性能指标和环境详细信息