P52

第四章 数值计算

优化是指改变x大小以最小化或最大化某个函数f的任务。

导数是在x点处的斜率，换句话说，它表明如何缩放输入的小变化才能在输出获得相应的变化。因此导数对于最小化一个函数很有用，因为它告诉我们如何更改x来略微改善y。

沿着梯度相反的方向进行下降就是梯度下降

反向传播的梯度计算

不是采用函数定义的数值求导，而是反向传播中的批量求导

需要计算输入和输出都为向量的函数的所有偏导数，包含所有这样的偏导数的矩阵被称为Jacobian矩阵。

二阶导数确定函数的曲率，对于负曲率，代价函数实际上比梯度预测下降的更快，没有曲线时，梯度正确预测下降值。对于正曲率，代价函数比预期下降得更慢，并且最终会开始增加，因此太大的步骤，实际上可能会无意的增加函数值。

当我们的函数具有多维输入时，二阶导数也有很多，我们将这些导数合并成一个矩阵，称为Hessian。可以通过Hessian的特征值判断函数的点是不是全局最优点。

使用牛顿法去找最小点

仅使用梯度信息的优化算法称为一阶优化算法，如梯度下降。使用Hessian矩阵的优化算法称为二阶最优化算法，如牛顿法。

深度学习中的大部分问题都难以表示成凸优化的形式，凸优化的分析思路对于深度学习算法的收敛性非常有用。

P169 深度学习中的优化

这一类特定的优化问题：寻找神经网络上的一组参数ϴ，它能显著的降低代价函数j(ϴ),该代价函数通常包括整个训练集上的性能评估和额外的正则化项。

1. 机器学习任务中，作为训练算法使用的优化与纯优化有什么不同。
2. 导致神经网络优化困难的几个具体挑战？
3. 实用算法，包括优化算法本身和初始化策略。更高级的包括在训练中自适应的调整学习率，或者使用代价函数二阶导数包含的信息。

将机器学习问题转化为一个优化问题的最简单的方法是最小化训练集上的期望损失。基于最小化这种平均训练误差的训练过程称为经验风险最小化。

代理损失函数和提前终止

我们真正关心的函数并不能真正的被高效优化，因此采用代理损失函数，例如负对数似然可以作为替代，在某种情况下，代理损失函数比原函数学到的更多，负对数似然允许模型估计给定样本的类别的条件概率，如果该模型效果好，那么它能够输出期望最小分类误差所对应的类别。

在某些情况下，代理损失函数比原函数学到的很多。例如，使用对数似然替代函数时，在训练集上的0-1损失达到0之后，测试集上的0-1损失还能持续下降很长一段时间。这是因为即使0-1损失期望为0时，我们还能拉开不同类别的距离以改进分类器的鲁棒性，获得一个更强壮的、更值得信赖的分类器，从而，相当于简单的最小化训练集上的平均0-1损失，它能够从训练数据中抽取更多的信息。

一般的优化和我们用于训练算法的优化有一个重要的不同，训练算法通常不会停止在局部极小点。反之，机器学习通常优化代理损失函数，但是在基于提前终止的收敛条件满足时停止。通常，提前终止使用真实潜在损失函数，如验证集上的0-1损失，并设计为在过拟合发生之前终止。与纯优化不同的是，提前终止时代理损失函数仍然有较大的导数，而纯优化终止时导数较小。

9