第九章 前向神经网络

1.多层感知机与布尔函数

2.深度神经网络中的激活函数

常用激活函数?

Sigmoid

Tanh

ReLu

为什么Sigmoid和tanh激活函数会导致梯度消失的现象?

Sigmoid的导数为f'(z) = f(z)(1-f(z))在z很大或很小的时候都会趋近于0，造成梯度消失的现象。

tanh的导数为f'(z) = 1-(f(z))²，在z很大或很小时都会趋近于0，同样会出现梯度消失的问题。实际上,tanh激活函数相当于sigmoid的平移。

ReLu系列的激活函数相对于sigmoid和tanh激活函数优点是什么？它们有什么局限性以及如何改进?

ReLu的优点:

（1）ReLu只需一个阈值即可得到激活值，而sigmoid和tanh激活函数的值需要计算指数。

（2）ReLu的非饱和性可以有效地解决梯度消失的问题，提供相对宽的激活边界（0, +inf）。

（3）ReLu的单侧抑制提供了网络的稀疏表达能力。

ReLu的局限性:

（1）导致神经元死亡的问题，这是由于函数f(z) = max(0, z)导致负梯度在经过该ReLu单元时被置为0，且在之后也不被任何数据激活，即流经该神经元的梯度永远为0，不对任何数据产生响应。

（2）为了解决神经元坏死的问题，提出了leaky ReLu f(z) = max(αx, x)，其中α为一个很小的正常数，这样既实现了单侧抑制，又保留了部分负梯度信息以致不完全丢失。但α需要较强的人工先验或多次重复训练以确定适合的值。

（3）基于此，参数化的PReLu也产生了，与Leaky ReLu的区别是将负轴部分斜率α作为网络中一个可学习的参数，进行反向传播训练。还存在Random ReLu，斜率α作为一个满足某种分布的随机采样，在一定程度上能起到正则化的作用。

3.多层感知机的反向传播算法

平凡误差损失函数和交叉熵损失函数分别适合什么场景?

一般来说，平方损失函数更适合输出为连续，并且最后一层不含sigmoid或softmax激活函数的神经网络；交叉熵损失则更适合二分类或多分类的场景。详细解释参考P.214

4.神经网络训练技巧

解决过拟合的方法：

（1）数据集增广。

（2）参数范数惩罚/ 正则化。

（3）模型集成。

（4）Dropout是模型集成方法中最搞笑与常用的。

（5）批量归一化有效避免复杂参数对网络训练产生的影响，在加速训练的同时也提升了网络的泛化能力。

神经网络训练时是否可以将全部参数初始化为0?

不可以。如果将全部参数初始化为同样的值，那么无论前向传播还是反向传播的取值都是完全相同的。学习过程将永远无法打破这种对称性，最终同一网络层中的各个参数仍然是相同的。因此需要随机地初始化神经网络的参数，以打破这种随机性。

为什么dropout可以抑制过拟合？它的工作原理和实现?

Dropout作用于每份**小批量训练数据**，由于其随机丢弃部分神经元的机制，相当于**每次迭代**都在训练不同的神经网络。类比于Bagging，dropout可被认为是一种实用的大规模深度神经网络集成方法。

dropout的具体实现中，要求某个神经元节点激活值以一定的概率p被丢弃，即该神经元暂时停止工作。因此对于包含N个神经元节点的网络，在dropout的作用下，可以看作2^n个模型的集成（每个神经元有2种状态）。这2^n个模型可以认为是原始网络的子网络，它们共享部分权值，并且具有相同的网络层数，而模型整体的参数数目不变，这就大大简化了运算。

在神经网络中应用dropout包括训练和预测两个阶段。在训练阶段，每个神经元需要增加一个概率系数，具体公式参考p.219。测试阶段是前向传播的过程。在前向传播的计算时，每个神经元的参数要预先乘以概率系数p，以恢复在训练中该神经元只有p的概率被用于整个神经网络的前向传播计算。

批量归一化的基本的基本动机和原理是什么？在CNN中如何使用?

随着神经网络训练恶进行，每个隐藏层的参数变化使得后一层的输入发生变化，从而每一批训练数据的分布也随之改变，致使网络在每次迭代中都需要拟合不同的分布，增大训练的复杂度以及过拟合的风险。
批量归一化方法是针对每一批数据，在网络的每一层输入之前增加归一化处理（均值为0，标准差为1），将所有批数据强制在统一的数据分布下。公式如下：
对任意一个神经元（假设为第k维），x^k = (x^k - E(x^k)) / √var(x^k)， 其中x^k为该层第k个神经元的原始输入数据，E(x^k)为这一批输入数据在第k个神经元的均值， √var(x^k)为这一批数据再第k个神经元的标准差。
但是批量归一化也降低了模型的拟合能力，归一化之后的输入分布被强制为0均值和1标准差。以sigmoid激活函数为例，批量归一化之后数据集整体处于函数的非饱和区，只包含线性变化，破坏了之前学习到的特征分布。因此，在具体实现中引入了变换重构以及可学习参数γ和β（参考书P.221）。

5.深度卷积神经网络

其特点是每层的神经元只响应前一层局部区域范围内的神经元（全连接网络中每个神经元节点相应前一层的全部节点）。相较于其他网络模型，卷积操作的参数共享特性使得需要优化的参数数目大大缩减，提高了模型的训练效率及可扩展性。

卷积操作的本质特性包括稀疏交互和参数共享，具体解释这两种特性及其作用。

稀疏交互：

对于全连接网络，任意一对输入与输出神经元之间都产生交互，形成稠密的连接结构。而在卷积神经网络中，卷积核尺度远小于输入的维度，这样每个输出神经元与前一层特定局部区域内的神经元存在连接权重（即产生交互），这种特性称为稀疏交互。
稀疏交互的物理意义是，通常图像、文本、语音等现实世界中的数据具有局部的特征结构，我们可以先学习局部的特征，再将局部的特征组合起来形成更复杂和抽象的特征（例子参考书p.224）。

参数共享：

参数共享是指在同一个模型的不同模块中使用相同的参数，它是卷积运算的固有属性。全连接网络中，计算每层的输出时，权值参数矩阵中的每个元素只作用于某个输入元素一次；而在卷积神经网络中，卷积核中的每一个元素将作用于每一次局部输入的特征位置上。根据参数共享的思想，我们只需要学习一组参数集合，而不学要针对每个位置的每个参数都进行优化，从而大大降低了模型的存储需求。参数共享的物理意义是使得卷积层具有平移等变性。假如图像中有一只猫，那么无论它出现在图像中的任何位置，我们都应该将它识别为猫，也就是说神经网络的输出对于平移变换来说应当是等变的，

常用的池化操作有哪些？池化的作用是什么？

常用的池化操作只要针对非重叠区域，包括均值池化、最大池化等。

池化操作除了能显著降低参数量外，还能够保持对平移、伸缩、旋转操作的不变性。

6.深度残差网络

ResNet的提出背景是解决或缓解深层的神经网络训练中的**梯度消失**问题。试验表明，56层的神经网络比20层的神经网络误差要大，这很大程度归结于深度神经网络的梯度消失问题。