

1.为什么必须在神经网络中引入非线性?

答: 否则, 我们将获得一个由多个线性函数组成的线性函数, 那么就成了线性模型。线性模型的参数数量非常少, 因此建模的复杂性也会非常有限。

3 Relu在零点不可导, 那么在反向传播中怎么处理?

答:

caffe源码~/caffe/src/caffe/layers/relu_layer.cpp倒数第十行代码:

```
bottom_diff[i] = top_diff[i] * ((bottom_data[i] > 0)+ negative_slope * (bottom_data[i]
<= 0));
```

这句话就是说间断点的求导按左导数来计算。也就是默认情况下 (negative_slope=0) ,间断点处的导数认为是0.

5 Relu激活函数的优缺点?

答: 优点包括:

1. 解决了梯度消失、爆炸的问题
2. 计算方便, 计算速度快, 求导方便
3. 加速网络训练

缺点包括:

1. 由于负数部分恒为0, 会导致一些神经元无法激活
2. 输出不是以0为中心

BN解决了什么问题

BN的实现流程

请参考:

https://blog.csdn.net/weixin_42137700/article/details/91488889

附加题:

CV:

3.目标检测里有什么情况是SSD、YOLOv3、Faster R-CNN等所不能解决的，

假设网络拟合能力无限强

若网络拟合能力无限强，

- 剩下最大得问题数据完备性。
- 目标是3维的，显示在图像是2维的，二维数据库在目标大小，以及方向上。无法满足SSD,YOLOv3,Faster R-CNN得需求。

NLP:

MLP: 因为句子长度不固定，所以一般是用Bag-of-Word-Vectors 简单将词向量相加，然后使用MLP，这个方法比较简单，训练速度比较快，得到的结果也不是很差。只是没有利用到上下文信息。

RNN: 将句子或文本作为序列输入比较自然，可以利用到历史信息，将词的顺序也考虑进去。此外，RNN的使用有很多变化，最简单的是使用最基础的RNN将最后一个词的隐含层输出进行softmax，但这样越往后的词会显得越重要（因为从前向后，最后一个输入的信息会更多地留下来），或者将每个隐含层的输出的和和均值进行softmax。也可以将基础的RNN变化为使用LSTM（很容易过拟合）或GRU单元，或使用bi-RNN获取更多的上下文信息。最后结果上并没有比MLP的结果好很多，但训练速度上慢了很多。（PS: 我这里的RNN指Recurrent Neural Network）

CNN: 这个方法看起来对处理这种序列输入不是很自然，所以一般是对句子的所有词的词向量使用不同大小的窗口（能获得一定前后文信息，有点像隐式的n-gram）进行一维的卷积，然后用最大化获得最重要的影响因子（我个人觉得这个比较讲得通，因为文本中不是所有词对分类结果都有贡献）和定长输出。CNN的方法也训练起来也比较简单，现阶段最后实验的效果也是最好。但有一些窗口大小上选取的经验问题，对文本长程依赖上的问题也并不是很好解决，因为窗口大小一般不会选很大。

BI:

MLR的原理是什么？ 做了哪些优化？

MLR可以看做是对LR的一个自然推广，它采用分而治之的思路，用分片线性的模式来拟合高维空间的非线性分类面，其形式化表达如下：

这里面超参数分片数m可以较好地平衡模型的拟合与推广能力。

MLR算法适合于工业级的大规模稀疏数据场景问题，如广告CTR预估。

优势体现在两个方面：

1. 端到端的非线性学习：从模型端自动挖掘数据中蕴藏的非线性模式，省去了大量的人工特征设计，这 使得MLR算法可以端到端地完成训练，在不同场景中的迁移和应用非常轻松。
2. 稀疏性：MLR在建模时引入了L1和L2, 1范数正则，可以使得最终训练出来的模型具有较高的稀疏度， 模型的学习和在线预测性能更好。

2.3 MLR算法高级特性

1. 结构先验。基于领域知识先验，灵活地设定空间划分与线性拟合使用的不同特征结构。例如精准定向 广告中验证有效的先验为：以user特征空间划分、以ad特征为线性拟合。
2. 线性偏置。这个特性提供了一个较好的方法解决CTR预估问题中的bias特征，如位置、资源位等。

3. 模型级联。MLR支持与LR模型的级联式联合训练，这有点类似于wide&deep learning。在我们的实践经验中，一些强feature配置成级联模式有助于提高模型的收敛性。

1. 增量训练。实践证明，MLR通过结构先验进行pretrain，然后再增量进行全空间参数寻优训练，会获得进一步的效果提升。同时增量训练模式下模型达到收敛的步数更小，收敛更为稳定。

MLR中使用的优化算法是从OWLQN改进过来的，主要有三个地方的变化：

MLR使用方向导数来优化目标函数，而不是OWLQN的次梯度

MLR对更新方向 p 进行了象限约束：非正定时直接用方向导数作为搜索方向，否则要进行象限约束在方向导数所在象限内。

线性搜索的象限约束不同，当MLR参数不在零点时，line search保持在参数所在象限内搜索，在零点时，参数在方向导数约束的象限内进行line search，给定更新方向，MLR 使用了 backtracking line search方法找到合适的步长 α