DL_learning_report

> 对于task1 task2 的完成过程以及这一阶段的学习情况进行总结

完成情况

- 代码地址: https://github.com/k15201363625/hands_on_deeplearing
- task1 : 基于numpy 实现基本神经网络层次结构 并在mnist数据集上测试实现的层次
  - **Fullyconnectedlayer**
  - **Convlayer**
  - **MaxPoolinglayer**
  - **AveragePoolinglayer**
  - **SoftmaxLosslayer**
  - **BasicRNNCell**
  - **LSTMCell**(仍有问题存在　未解决)
  - **多种Activator**
  - **多种Optimizer**
  - **DropoutLayer**
  - **BatchNormLayer**
- task2 : 优化改进已有层次　并重新组织代码　实现了改进版本的CNN　并用于cifar100数据集上
  - 改进 1 :改变网络结构的组织
    - 从原来的单层次实现相互独立到基于统一的基类Operator Variable实现
    - 从原先的操作与数据耦合　到操作与数据解耦
    - 从原来的各个层次单独调用　到有了图结构后通过递归实现自动前向反向传播
    - 通过图结构对于变量以及操作更好的管理　在更新参数上也更方便
  - 改进 2 : 加上了深度神经网络的基本优化策略
    - 激活函数从sigmoid -> relu 等在深度神经网络更快速收敛的激活函数　在CNN上表现很好
    - 训练过程优化器改进　从gd->sgd->momentum->rga(基于二阶)->Adam(adadelta+rmsprop)
    - 加上了BN(batch normalization层次)
    - 加上了Dropout层次　在解决过拟合方面很有意义
  - 问题：
    - 在20分类的cifar100数据集上　神经网络的表现不好　尽管对于结构进行了调整　但是由于CPU计算缓慢　深度网络的训练耗时　目前表现没有达到baseline (相反mnist很轻松达到90以上)

学习情况

- 从以前的纯理论学习公式推导　+　高级封装调用(tensorflow pytorch)，到现在基于numpy 手动实现自己的神经网络层次对我而言是一个很大的挑战，从前的学习中由于现有框架几乎完美的封装，所谓的实践就是调包然后获取成就感，对于公式没有细致的挖掘，对于很多细节问题没有重视，很多推导过程的问题以及实现过程的难点都有意无意的忽略掉了。但通过这次手动实现自己的层次结构，并且通过gradient_check检验结果，并构建自己的神经网络用于实战的学习，我对于很多基础数学知识的掌握更加熟练并学到很多新的有关向量张量运算的知识以，并且对于基本的神经网络结构层次的原理有了更深层次的理解，还在手动实现中对于numpy的基本使用更加熟练对于python处理矩阵运算，面对对象程序设计，以及神经网络训练流程的代码编写更加得心应手。

- 从简单的全连接层->卷积层->循环神经网络中的RNNCell LSTMCell 前向传播,反向传播的推导中包含大量数学基本知识的应用,在这个过程中我学到了很多(*很多资料博客看了很多遍,然后手动推导了几次,之后在手动实现前还需要温习一下,并且在实现过程中还参考了已有代码的实现*)
- 完成进度由于循环神经网络的学习受阻，并且在为了该进程图结构而重构代码上花费了不少时间，在version1版本代码训练表现很差的调试修正问题上耗费了不少时间，综上完成时间远超出预计。

难点与解决

- FC层次对于x看做行向量/列向量的不同处理
- conv层次的快速实现(矩阵乘法形式)
- conv层次反向传播中矩阵翻转之后进行反卷积的推导
- Pooling层次索引对照关系
- 多种Activator 对于网络效果的分析比较
- 多种训练过程优化器从直观上以及原理上的分析比较
- BasicRNNCell 反向传播　BPTT过程推导
- 较为复杂的层次结构LSTM的反向传播的推导
- 更为复杂的Recursive NN的基于结构的反向传播的推导以及应用(目前还未实现)
- 对于维度对应问题的调试
- 对于梯度计算结果的检查(有用的Gradient_check 的快速实现)
- 对于复杂网络结构以及大量代码的调试
- 对于已有实现的问题:数据操作耦合　网络结构组织混乱需要手动调用各个层次问题的解决(基于graph结构　借鉴tensorflow中的实现对于代码进行重构　实现数据操作解耦　并且自动前向传播　反向传播　更新参数)
- 训练流程的控制，以及过程信息的记录
- 层次结构具体代码实现的一些细节问题

> 以上问题都已得到解决(可能我比较笨，很多问题上花费了很多时间)
> 如下是**余留问题**

余留问题

- LSTM的gradient check 过程中发现对于输出们的三个参数(woh,wox,bo)的梯度计算都是正确的，但是对于其他门的参数计算出现错误，整整调试了一天，并且参考了不同的文章之后手动重新推到了公式，并且参考了一些其他的代码实现但是仍然没有解决这个问题
- 循环神经网络Embedding 的推导以及RecursiveNN的实现部分还没有来得及看　导致还没有进行NLP相关任务(稀疏编码效率太低问题颇多)

参考(reference)

> 包含原理讲解以及代码实现的参考(以下排名有一定先后顺序)
>

> - http://ruder.io/optimizing-gradient-descent/index.html
> - https://zybuluo.com/hanbingtao/note/541458
> - https://zhuanlan.zhihu.com/c_162633442
> - https://www.jianshu.com/p/b6b9a1b0aabf
> - https://github.com/wuziheng/CNN-Numpy
> - https://github.com/hanbt/learn_dl/blob/master/
> - https://github.com/hanbt/learn_dl
> - http://fanding.xyz/
> - https://zhuanlan.zhihu.com/p/34209938
> - https://segmentfault.com/a/1190000016775420
> - https://ilewseu.github.io/
> - https://zhuanlan.zhihu.com/p/28054589

- https://blog.csdn.net/qq_16234613/article/details/79476763
- https://blog.csdn.net/zhaojc1995/article/details/80572098
- http://nicodjimenez.github.io/2014/08/08/lstm.html
- http://www.wildml.com/deep-learning-glossary/
- https://blog.csdn.net/qq_29762941/article/details/89294252
- https://zhuanlan.zhihu.com/p/26892413
- https://zhuanlan.zhihu.com/p/24709748