使用库

- TensorFlow>=1.10 and <2.0
- matplotlib
- numpy

推荐TensorFlow安装方式

> conda install tensorflow1.14.0
> pip install tensorflow1.14.0

课程说明

- 课程流程以课程大纲为主
- 本节课主要学习内容为
  - 神经网络模型
  - 激活函数
  - 优化问题
  - 多层神经网络模型
  - 交叉熵损失函数
  - TensorFlow相关API使用
    - 学习求导
- 补充内容，仅作为了解内容
  - 使用NumPy实现多层神经网络推断过程
  - 样本均衡问题
  - 梯度消失问题

参考材料

- 官方文档
  - tensorflow.google.cn
  - tensorflow.org
  - numpy
- 《线性代数》
- 《深度学习》第五章

机器学习库的使用

- 机器学习库和工具
  - 实现过程中需要较强的数学基础
  - sklean
    - 常用的机器学习库，算法已经封装好，不需要太多调整。
    - 学习目标，是将sklearn中的算法进行了解和实现。
  - Excel:包含部分机器学习算法
  - spss:包含数据分析与统计功能
  - weka:开源包含机器学习算法的库
- 深度学习库
  - 今后可能需要自行实现
  - TensorFlow
    - 矩阵运算
    - 自动差分（求导）
    - 工程中使用较多
  - PyTorch and TF 2.0
    - 现在研究领域使用较多
  - caffe
  - paddlepaddle
- 可视化
  - matplotlib
    - seaborn
  - vispy
  - mayavi
- 学习过程中
  - 一定要看书
  - 并且找项目练手
  - flyai.com；天池数据大赛；kaggle
  - 学习过程中先懂得算法的使用，再了解算法原理。

内容回顾

- 建议1：把原理写完整
- 建议2：尝试不同的参数完成任务
- 小建议3：直接保存矢量图，不要截图。直接发word或pdf版，不要压缩。
- 了解数据
  - data：[1000, 1], 代表有1000个样本，每个样本1个属性。
    - 大部分机器学习算法是处理浮点类型数据的
    - 通常使用浮点型的向量表征样本、特征

        - 作业例子中样本向量长度是1
    - label：[1000, 1], 代表标签向量长度为1。
    - 可视化：
        - 绘制图形，进行分析
        - 可以辅助建模
- 建立模型
    - 根据可视化结果和样本shape建立模型为：
        - Y=XW+b, Y模型输出，X样本，Wb为可训练参数
        - X[1000, 1], Y[1000, 1]，所以W矩阵形状是W[1, 1], b[1]
        - 训练过程中随机给W即可，b取0。
    - 给定损失函数（loss,L）为均方误差（MSE）
        - 进行训练即可
        - 头疼的问题：L关于w和b的导数如何求
        - TensorFlow等机器学习库就可以自动求导
        - 有自动求导就可以完成优化了
        - 人们只需要关心建模即可
- 模型优化
    - 问题是？太简单拟合效果不好
    - 所以需要做特征工程
        - 做x^2的特征
        - 做特征后，数据特征就增多了
        - 此时X[1000, 2], W[2, 1]
        - 特征工程首先是根据数据图看

学习问题

- 求导难求

注意点

- 学习过程中，要用于尝试。
    - 对于新的想法要不断尝试
- 如何拟合复杂的曲面（曲线）
    - 选择1：做特征工程+简单模型
        - 问题是：什么特征合适?
        - 尝试不同的特征并进行训练
        - 让模型自行对特征进行选择

    - 选择2：建立比较复杂的模型
        - 问题是：有很多机器学习方法可以选择：SVM，Tree，Adaboost
        - 本节课使用多层神经网络（多层感知器），是一个复杂度较高的模型。
        - 让模型自动做特征
        - 但是需要付出更多计算代价

基础分类问题（Basic文件夹）

- API使用
    - tf.constant:定义常数
    - tf.Variable:定义变量
    - tf.placeholder:定义placeholder
- 多层神经网络
    - X -> H = f(XW+b) -> Y=HW+b
    - H称为隐藏层，其是非线性特征
        - H向量长度也称为隐藏层神经元数量
    - f为非线性函数，也称为激活函数
    - 万能近似定理：有一个隐藏层的神经网络可以拟合任意复杂的函数
- 多分类问题
- 复杂曲面的拟合问题
    - 思路1：层数少，但是隐藏层神经元数量多 -> 广度神经网络
    - 思路2：层数多，隐藏层神经元数量适中 -> 深度神经网络

神经网络进行手写数字识别（MNIST文件夹）

- 数据是28×28的灰度图
- 使用长度为784的向量来进行保存
    - 此时数据X，格式为[N, 784]
    - 其他文本数据可能20w维
- 标签：手写数字
    - 从0~9的10个数字，但是机器学习应当处理浮点向量
    - OneHot，将整形数字转换为浮点型的向量。
    - 此时标签向量长度为10
    - 推广：很多分类问题标签都可以使用类似向量化方式
- 明确输入
    - 图像：X[N, 784]
    - 标签：D[N, 10]
- 此时需要建模

    - Y=XW+b
    - W[784, 10]
    - b[10]
    - 起名非常多
        - 线性模型
        - 线性回归
        - 单层神经网络
        - 一个全连接层
- 训练过程
    - 初始值可能会影响迭代收敛速度
    - Xavier、He初始化 （了解）
- 新的分类问题损失函数：交叉熵
    - $loss = \int_R -p(x) \log q(x) dx$
    - OneHot标签可以看成是p
    - 神经网络输出也要转换为概率
    - 通过Softmax，将神经网络输出转换为概率q。
    - 交叉熵当p和q分布相同时loss取得极小值。
    - 对于分类问题，通常都使用交叉熵。
- 图像数据也可以做特征工程
    - 其包含卷积等复杂变换
    - 任意一本图像、信号处理书籍中均有
- 名词
    - 全连接层：矩阵乘法+激活函数
    - 多个全连接层：多层感知器
- 重点
    - 了解多层神经网络的构建
    - 了解模型的保存与恢复
    - 了解交叉熵作为损失函数
    - 学习TensorFlow基础API的使用

实例2：鸢尾花分类（IRIS文件夹）

- 数据类型：X[150, 4]
- 标签3类：D[150, 3]
- 模型：多层神经网络
- 梯度消失问题
    - 多层神经网络容易梯度消失问题
    - 梯度消失问题表现形式

- 梯度消失问题解决
  - relu优势
  - BN作为了解

补充实例：贷款欺诈（Cheat文件夹）

- 数据类型：X[N, 28]
- 标签2类：D[N, 2]
- 模型：多层神经网络
- 样本均衡问题
  - 多数情况数据集均是有偏的，不同类别数量不同
  - 正负样本数量不同的时候，需要进行样本均衡
  - 对正样本进行降采样
  - 对负样本进行超采样
  - 对样本进行加权
  - 平衡数据集：不同类别样本数量相同

TensorFlow的使用总结

- TensorFlow 1.x
  - 特点：延迟执行
    - 先使用Python描述计算过程（计算图）
    - 再将描述的计算过程放到C/C++写的高速核心中执行
    - 可以获得高效的计算
- 矩阵的定义和使用
- 矩阵的运算

建模流程总结

- 用向量串联起流程
- 深度学习模型可解释比较弱。
- 核心目标
  - 精度
    - 精度要高
    - 根据具体场景确定精度
  - 速度
    - 速度要快

        - 需要根据具体硬件进行优化
    - 精度和速度通常不可兼得
        - 需要具备根据具体场景权衡精度和速度的能力。
- 数据输入是什么
    - 作业：数据是X和D
- 模型是什么
    - 作业：Y=XW+b
- 模型的可训练参数是什么
    - 作业：Wb
- loss函数是什么
    - 作业：MSE
- 选择1：做特征工程，并建立一个较为简单的机器学习模型
- 选择2：让机器学习模型自动做特征
- 二维数据实例
    - 数据假设有两个属性x1,x2 -X[N, 2]
    - 样本有两类分别用-1,1来表示 -D[N, 1]
    - 建立模型Y=XW+b，符合这种形式均为线性模型。
    - 如果数据线性不可分：一条直线无法分开
        - 需要考虑非线性特征
        - 建立非线性模型
- 建模关键性问题
    - 机器学习中，数据大多是高维（n）数据
    - 此时用长度为n的向量v来表征数据
    - 如果进行计算h=vW[n, n2]->h长度为n2，称将数据v变到了n2维
        - 此时v和h是线性关系
        - 激活函数：
            - relu=max(0, x)
            - sigmoid = 1/(1+e^{-x})
        - 有一个隐藏层的神经网络
            - 可以以任意精度拟合任意复杂函数（万能近似定理）
        - 神经网络越复杂，拟合能力越强
            - 名词1：拟合能力，能形成多复杂的曲面，能拟合复杂数据
            - 名词2：隐藏层，计算过程中的向量，比如h
            - 名词3：隐藏层神经元数量，向量长度
            - 名词4：多层感知器，由多个隐藏层构成的神经网络
            - 广度神经网络
                - 隐藏层神经元数量较多
            - 深度神经网络
                - 隐藏层较多

        - 深度、广度神经网络均可以拟合复杂数据

API总结

- x.shape 矩阵形状
- tf.Variable()#定义TF变量
- tf.placeholder()#从外面接收样本
- tf.Session()#会话，执行所需操作
- 回去复习
    - 矩阵分块计算

运算补充

- $A \cdot B = \sum_k a_{ik} b_{kj}$
    - Numpy:
        - np.dot(A,B)
        - A.dot(B)
        - A @ B
    - TensorFlow
        - tf.matmul(A, B)
        - A @ B
- $A \odot B = a_{ij} b_{ij}$
    - A * B

练习

- 证明：交叉熵损失函数，为啥分布一样，得到最小值?
    - 通过使得偏导数等于0可证明
    - 泛函分析
- 回去想详细了解线性模型
    - 看深度学习16-20章

学习重点

- 学会TensorFLow的使用
    - 使用TF复现作业 y=ax+b
- 学习了解多层神经网络建模过程

    - 了解手写数字识别，以及精度的提升
- 课外练习：Cheat文件夹下，贷款欺诈预测，使用TF做训练和预测。
- 深度学习第二部分可以参考
- TensorFlow的API可以参考

其他

DarkNet:C语言写的深度学习框架。

课上问题

- tf1的静态计算图，如果是tf自动在计算前 自动加入 ‘会话’和‘执行计算图’的步骤可以么，和动态计算图有什么区别呢？如果不可以，会有什么问题？
- 从静态图到动态图有什么优势，感觉动态图不是很利于优化（tf.function）？感觉spark里面的DAG也是静态图+惰性执行，为什么tensorflow会改用动态图？
    - 易用性
- 是否可以认为一个session就是一个函数？tf的session和python的function有什么区别？
- 现在工业上用的tensorflow主要是tensorflow1还是tensorflow2？
- session.run执行的时候，类似代码中train_op 和 loss 这种图上有顺序的，是不是run的时候也要按照顺序写？
- 定义GradientDescentOptimazer()之后为什么还要定义compute gradient？
- 老师，现在tensorflow都2.多了，很多语法和库都变了，现在还学习1.多会不会out了
- NaN不是Not a Number的意思吗在这里表无穷的意思嘛 无穷不是Inf嘛
- tf里面有可以直接求梯度的功能，那是不是sympy这种求导的工具也就可以省略使用了
    - sympy属于符号计算，速度非常慢
    - tf求导属于数值计算，速度很快
- b = tf.get_variable("b", [1]) 这中间等号左边的b是计算步骤的名字，等号右边的“b”是变量的名字么？ 这两个‘b’有什么不同不太理解，后面引用‘b’的时候是指的哪个？
- W1 = tf.get_variable("w1", [1, 100])在jupyter里运行两次就会报错，tf.get_variable()与tf.Variable有什么区别，什么时候用前者，什么时候用后者？
- 激活函数的设计是不是求导方便和极值只有一个，两个准则，为什么relu在神经网络中比较好？
- 神经网络的最后输出如果是分类，可以不用softmax回归吗？ 就直接改成sigmoid，哪个类输出大就认为是哪个类，既然sigmoid本身就是输出（0,1），那么做归一化，使用softmax的好处是什么？
    - 简单可行
    - 基于能量的模型
- softmax为什么要弄成指数项，如果是归一为什么不直接输出求个比值y(k)/Σ(yi)？ 也能归一到0-1区间

    - 概率均是大于0的
- 神经网络的时间复杂度还有空间复杂度要怎么计算?
- 为什么特征编码时数据使用浮点数而不是整型数？整型数的计算速度不是更快吗?
- 全连接网络的缺陷或者上限主要在哪里？为什么精度无法达到更高?
    - 模型庞大
    - 计算速度较慢
    - 需要一定的特征工程
- 如何把每一个隐藏层的内容可视化?
    - 目前为止难以进行可视化
- 模型收敛的速度具体取决于哪些因素?
    - 模型大小
    - 数据
    - 优化
- 在一个网络里定义了这么多同名的变量如net，那在run()里fetch这个变量名出来的会是哪一个?
- Relu计算导数的时候把输出小于零的神经元全部消亡了，一下子干掉一半为什么Relu效果更好呢? 是不是梯度消失其实指的是神经网络大面积的神经元反向传播的时候，梯度全部消失？ 而少量的神经元参数梯度消失，并不会影响网络的效果?
    - 稀疏性
- 如何设计激活函数，激活函数是有非线性成分，求导简单就行了吗?
    - 参考文献
- Batch Norm后，对应的Normalization方法和参数也要保存在模型内，在预测的时候要用，是么?
- batch normal能否加快
  梯度下降收敛 但是batch Normal应该影响的正向传播 为啥会改善梯度消失 按说梯度消失应该是反向传播时候激活函数的导数去影响的?
- 解决正负样本不平衡问题可以通过修改损失函数的方式来实现吗？如果可以的话，有没有一些比较好的策略?
- 超采样：多次采用完全相同的样本会不会造成对这一类样本的过拟合？会不会有人工给这类样本加一定的扰动后生成新的样本的办法来避免过拟合?
    - 扩充训练集
        - 加入噪声
        - 避免过拟合问题
- 机器学习：人力+体力
    - 需要人工标注大量数据
    - 保护数据成本很高
    - MINIST数据集，如果人都识别不了这个数字是多少，那么机器识别出来了，该如何评判？能不能算是过拟合了?
        - 数据决定了模型的上限
        - 除非是信噪比较低的数据（比如超级夜景）