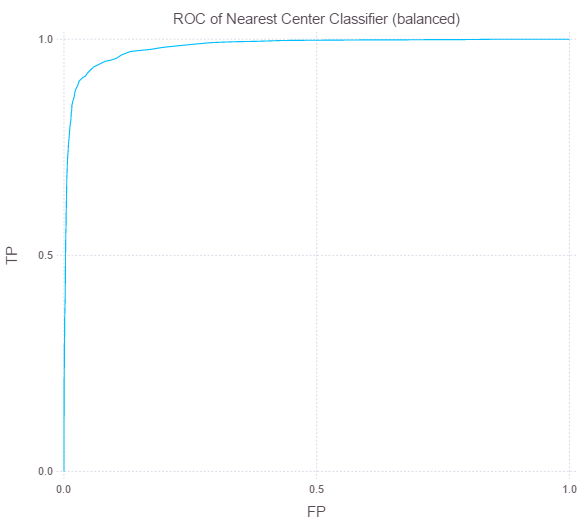
作业3.2 – 1133730117 - 张实唯

## 实验结果

由于不平衡的数据分类效果太差(auc接近0.5)，所以下面只放了平衡后数据的图。

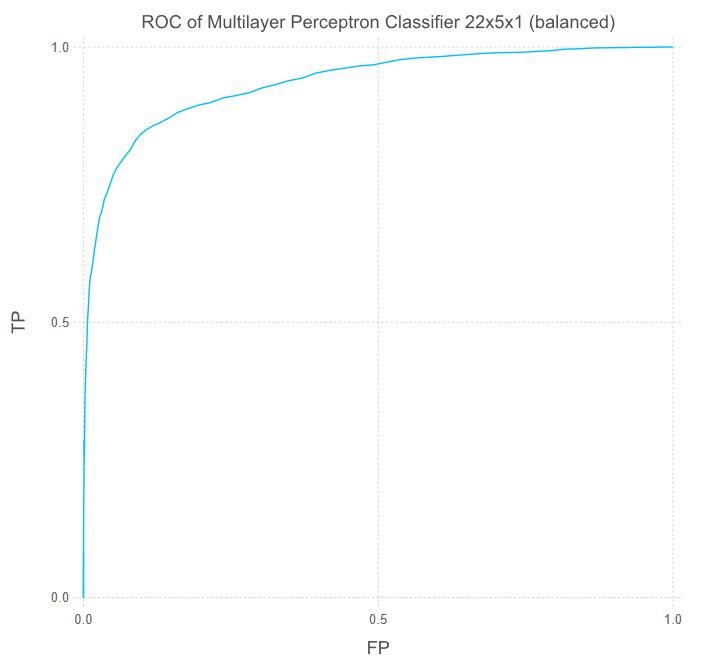
从分类结果上看对这个数据集而言22x15x1的MLP效果最好(0.9867)，但还是比不上单层感知机(0.9885), 而随着中间层数和节点数量的提升，分类效果先是有所上升，而后急剧下降。在隐含层增加到3层的时候，可以看到分类效果已经比不上随机分类了。

1. **NearestCenterClassifier**



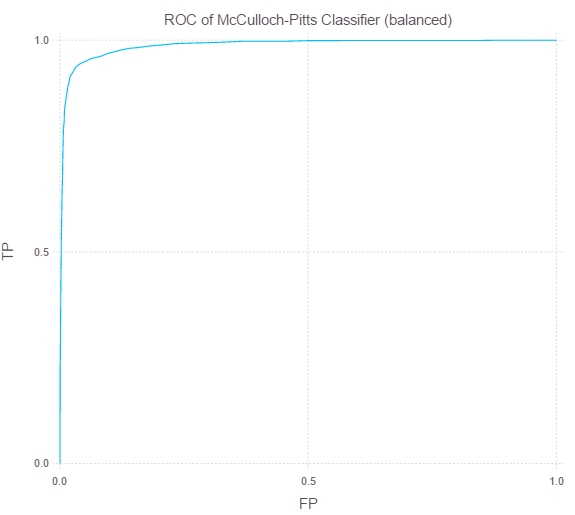
auc: 0.9830

1. **MultilayerPerceptronClassifier(22x5x1)**



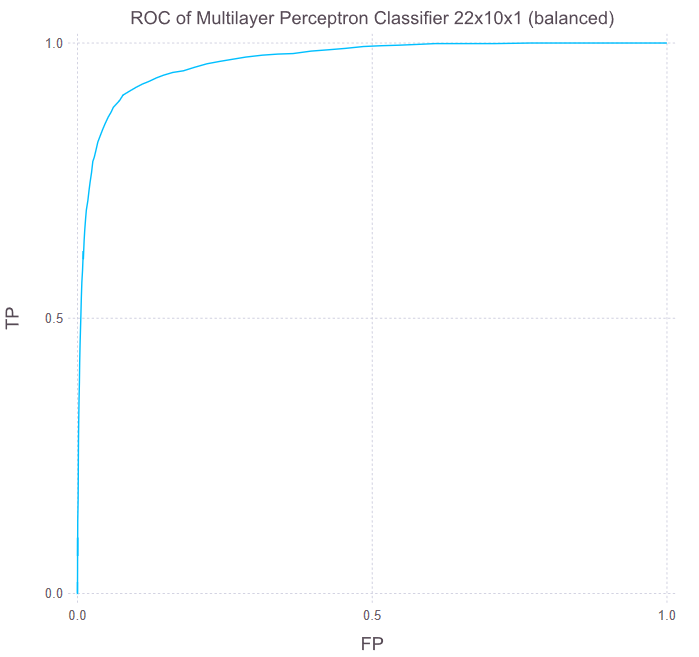
auc: 0.9356

1. **McCullochPittsClassifier**



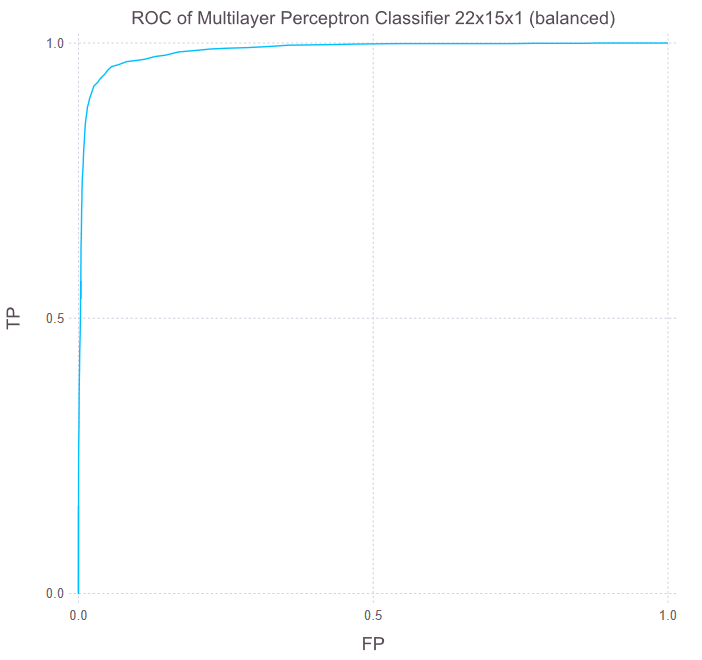
auc: 0.9885

1. **MultilayerPerceptronClassifier(22x10x1)**



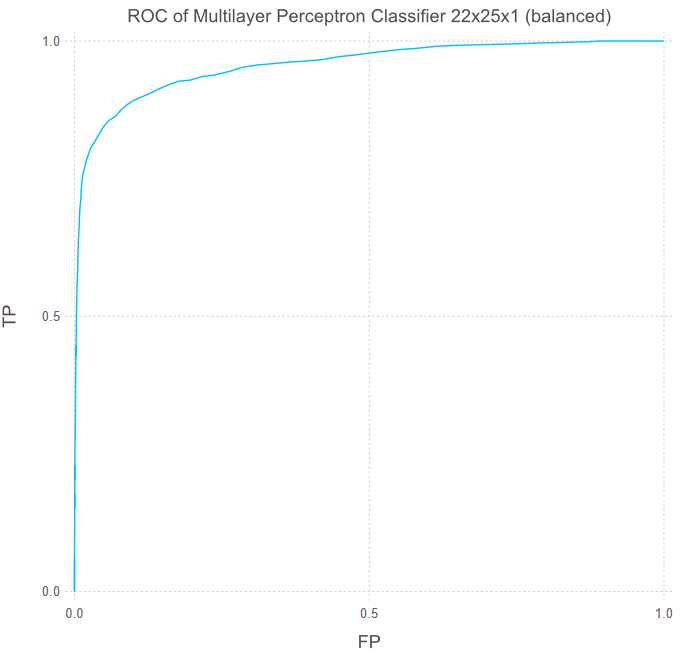
auc: 0.9679

1. **MultilayerPerceptronClassifier(22x15x1)**



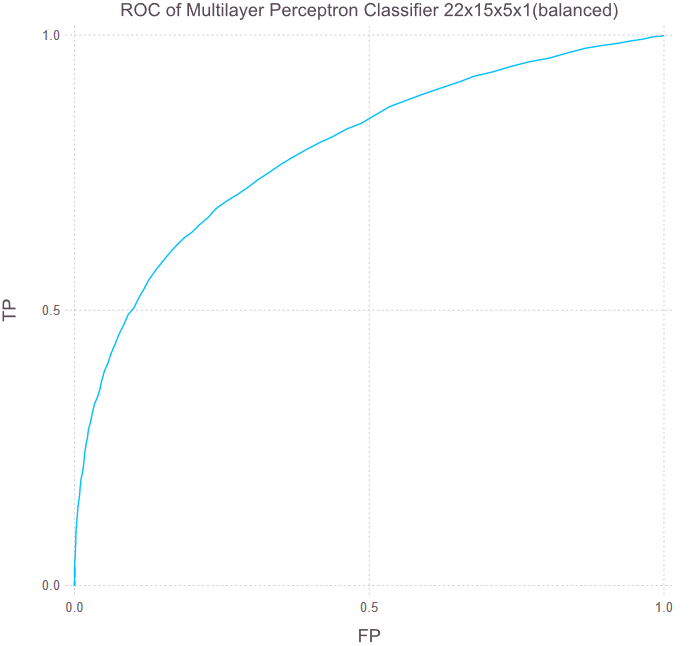
auc: 0.9867

1. **MultilayerPerceptronClassifier(22x25x1)**



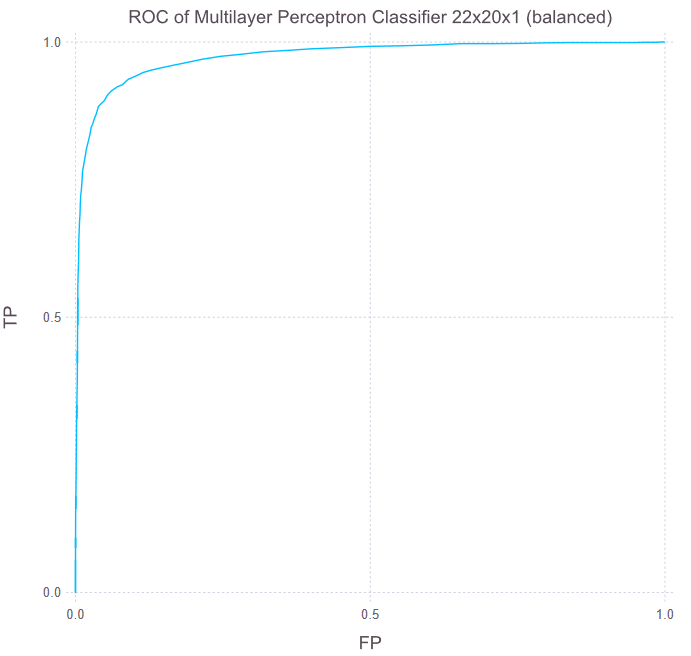
auc: 0.9565

1. **MultilayerPerceptronClassifier(22x15x5x1)**



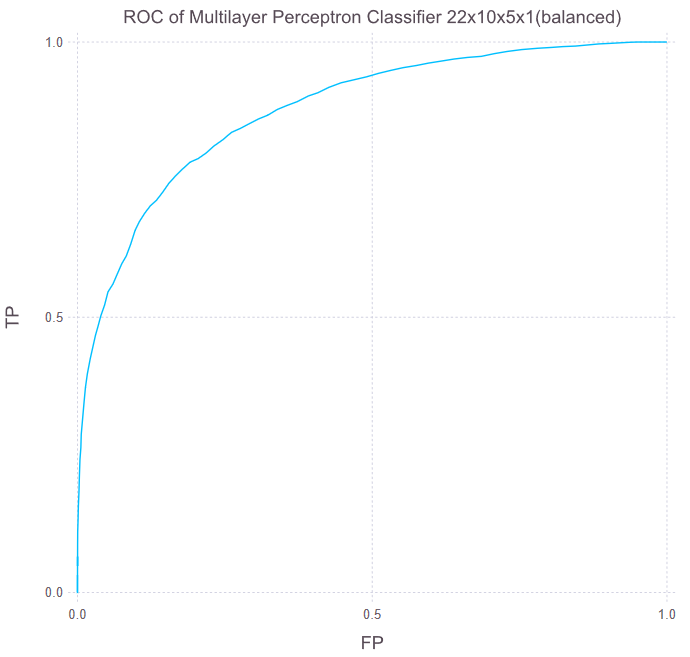
auc: 0.7911

1. **MultilayerPerceptronClassifier(22x20x1)**



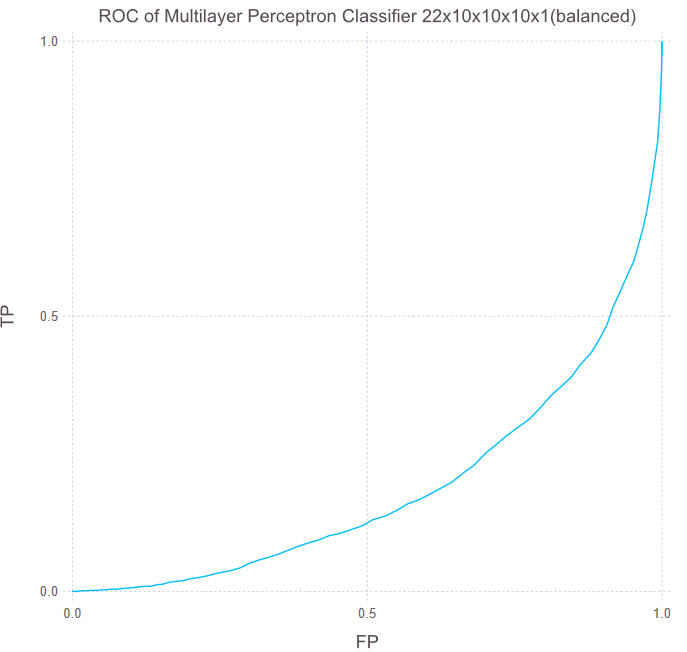
auc: 0.9736

1. **MultilayerPerceptronClassifier(22x10x5x1)**



auc: 0.8768

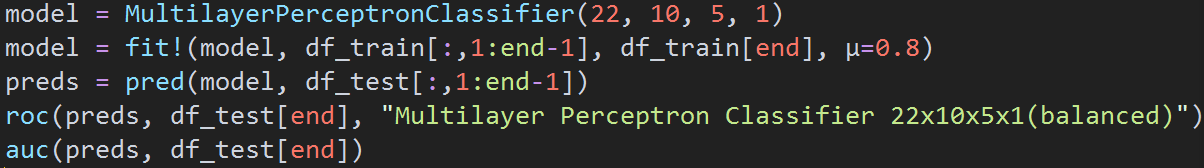
1. **MultilayerPerceptronClassifier(22x10x10x10x1)**



auc: 0.1898

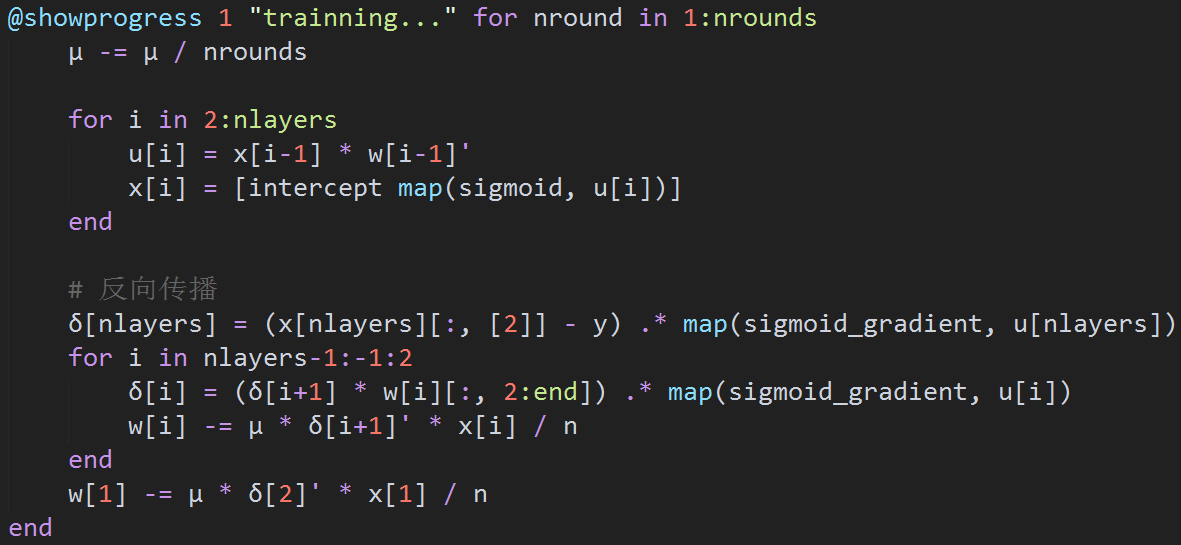
## 代码解释

1. **程序接口和基本特性：**



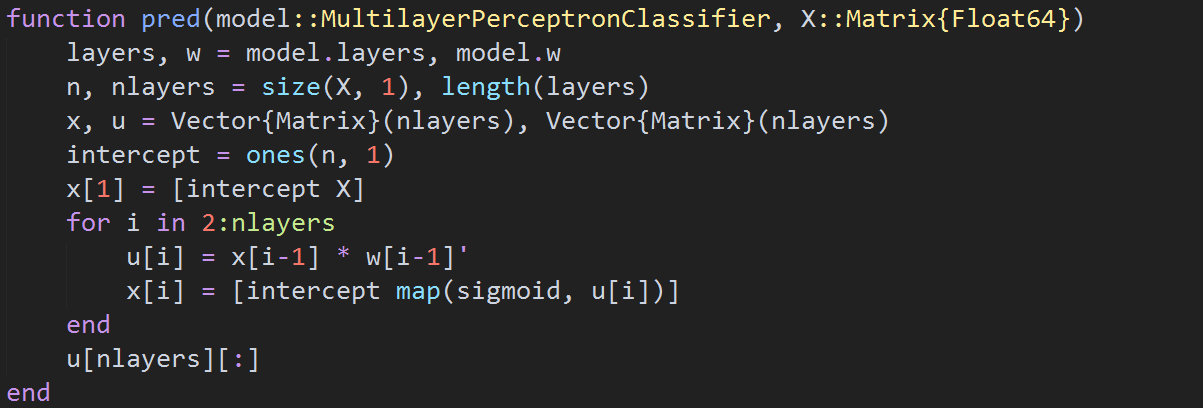
* 程序接口和上次作业的两个模型完全一样，只需替换模型名即可比对不同模型。
* 支持任意隐藏层层数和节点数量

1. **训练模型：**



* 代码使用向量化的写法，将所有输入作为一个矩阵一次性进行计算。
* μ是学习率，每轮学习开始时适当降低以减小即将coverge时的振荡。
* 第一个内层for循环是正向预测过程，先用上一层的输出结果与系数矩阵相乘得到当前层的原始输入，再调用激活函数并在每一行开头加上1来得到当前层的结果。
* 第二个内层for循环是误差反向传播过程，先用预测值和真实值相减获得误差大小，再乘以激活函数的导数作为最后一层的敏感信号，之后套公式向前逐层计算敏感信号值并更新对应层的系数矩阵。

1. **预测结果：**

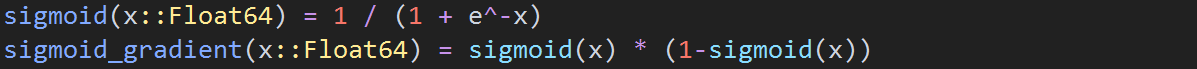


* 类似训练过程的前半部分，逐层将输入乘以系数矩阵，再调用激活函数，最后在每条记录前面加上固定值1。

1. **绘制ROC曲线、计算AUC值、训练集平衡方法：**

详见上次作业。

1. **激活函数及其导数：**



* 采用sigmoid函数作为激活函数