机器学习导论
作业三

参考答案

2018 年 5 月 22 日

1 [15pts] Decision Tree I

(1) **[5pts]** 假设一个包含三个布尔属性X, Y, Z的空间，并且目标函数是$f(x, y, z) = x$ **XOR** z，其中**XOR**为异或运算符。令H为基于这三个属性的决策树，请问：目标函数f可实现吗？如果可实现，画出相应的决策树以证明；如果不可实现，请论证原因。

(2) **[10pts]** 现有如表 1所示数据集：

Table 1: 样例表

X	Y	Z	f
1	0	1	1
1	1	0	0
0	0	0	0
0	1	1	1
1	0	1	1
0	0	1	0
0	1	1	1
1	1	1	0

请画出由该数据集生成的决策树。划分属性时要求以信息增益(information gain)为准则。当信息增益(information gain) 相同时，依据字母顺序选择属性即可。

Solution.

*(1) **[5pts]*** 可以构建一颗实现$f(x, y, z) = x$ **XOR** z的决策树，树如图 *1*所示：

*(2) **[10pts]*** 记现有数据集为D，属性集合为$A = \{X, Y, Z\}$。

第一次划分时，首先计算出$Ent(D) = -\sum_{f=0}^{1} p_f \log_2 p_f = 1$，

对于属性X，计算信息增益：$Gain(D, X) = Ent(D) - \sum_{x=0}^{1} \frac{|D^x|}{|D|} Ent(D^x) = 0$；

对于属性Y，计算信息增益：$Gain(D, Y) = Ent(D) - \sum_{y=0}^{1} \frac{|D^y|}{|D|} Ent(D^y) = 0$；

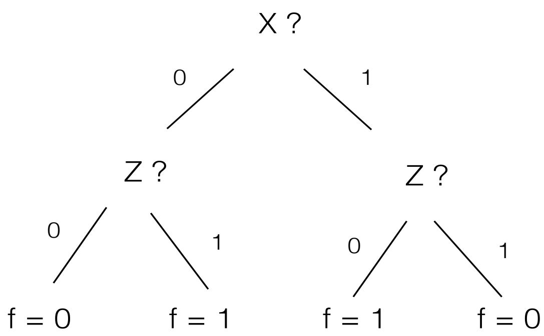

Figure 1: $f(x, y, z) = x \quad \textbf{XOR} \quad z$

对于属性Z，计算信息增益：$Gain(D,Z) = Ent(D) - \sum_{z=0}^{1} \frac{|D^z|}{|D|} Ent(D^z) = 1.5 - 0.75 \log_2 3$；

因此第一次应对属性Z进行划分。进而，在$Z = 0$的分支，不难发现该分支上的样本取值相同，因此$Z = 0$分支不必再进行划分，得到图 *2*：

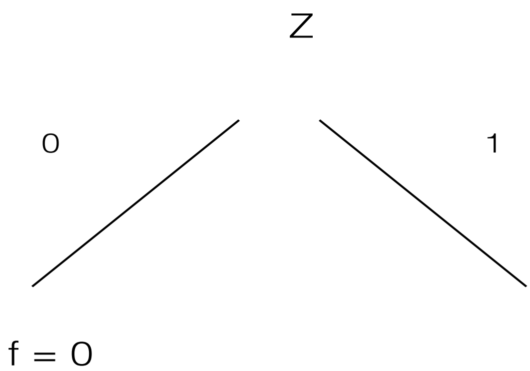

Figure 2: 第一次划分

接下来，对$Z = 1$进行划分，此时对数据集进行更新$D \leftarrow D \backslash \{(1,1,0,0),(0,0,0,0)\}$，对属性集合进行更新$A \leftarrow A \backslash \{Z\}$。

对于属性X，计算信息增益：$Gain(D,X) = Ent(D) - \sum_{x=0}^{1} \frac{|D^x|}{|D|} Ent(D^x) = 0$；

对于属性Y，计算信息增益：$Gain(D,Y) = Ent(D) - \sum_{y=0}^{1} \frac{|D^y|}{|D|} Ent(D^y) = 0$；

按字母顺序，选择属性X进行划分得到树如图 *3*：

此时，我们仅剩下属性Y，并且数据集在左右分支并没有得到相同的标签，因此，左右分支均以属性Y进行划分即可，得到最终如图 *4*所示树。

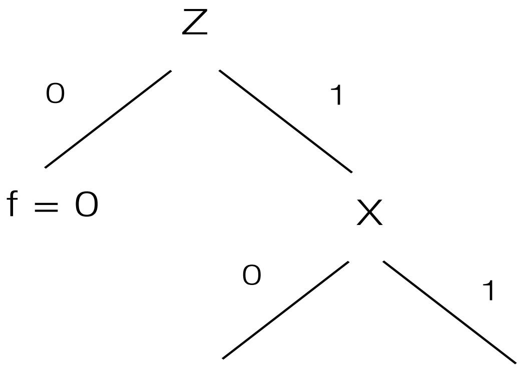

Figure 3: 第二次划分

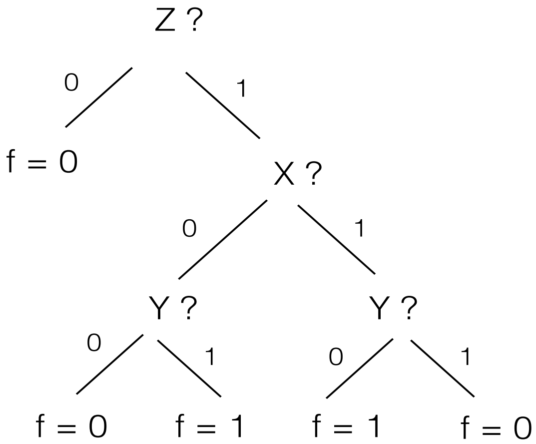

Figure 4: 最终树

2 [20pts] Decision Tree II

考虑如下矩阵：

$$\begin{bmatrix} 4 & 6 & 9 & 1 & 7 & 5 \\ 1 & 6 & 5 & 2 & 3 & 4 \end{bmatrix}^T$$

该矩阵代表了6个样本数据，每个样本都包含2个特征f_1和f_2。这6个样本数据对应的标签如下：

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T$$

在这个问题中，我们要构造一个深度为2的树进行分类任务。

(1) **[5pts]** 请计算根结点(root) 的熵值(entropy)。

(2) **[10pts]** 请给出第一次划分的规则，例如$f_1 \geq 4, f_2 \geq 3$；对于第一次划分后产生的两个结点，请给出下一次划分的规则。

提示：可以直观判断，不必计算熵。

(3) **[5pts]** 现在回到根结点(root)，并且假设我们是建树的新手。是否存在一种划分使得根结点(root) 的信息增益(information gain) 为0?

Solution. 此处用于写证明*(中英文均可)*

(1) ***[5pts]*** $-0.5\log_2 0.5 - 0.5\log_2 0.5 = 1$

(2) ***[10pts]*** 如果依据f_1划分，那么特征与标签的对应关系如下：

$$\begin{bmatrix} 1 & 4 & 5 & 6 & 7 & 9 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

如果依据f_2划分，那么特征与标签的对应关系则为：

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

显然，第一次划分的最佳规则是$f_1 \geq 7$。

当第二次划分时，标签为$(1,1)$的那一支不必再划分；对于标签为$(0,1,0,0)$的一支，如果我们依据f_1划分，则得到特征与标签的对应关系则为：

$$\begin{bmatrix} 1 & 4 & 5 & 6 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

如果依据f_2划分，得到特征与标签的对应关系则为：

$$\begin{bmatrix} 1 & 2 & 4 & 6 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

因此，第二次应该选择$f_2 \geq 2$。

(3) ***[5pts]*** $f_1 \geq 5, f_2 \geq 3, f_2 \geq 5$

3 [25pts] Universal Approximator

已知$f: [-1,1]^n \mapsto [-1,1]$是一个$\rho - Lipschiz$函数. 给定一个$\epsilon > 0$，要求构造一个激活函数为$sgn(x)$的神经网络$N: [-1,1]^n \mapsto [-1,1]$，使得对于任意的$x \in [-1,1]^n$，有$|f(x) - N(x)| \leq \epsilon$. ($Lipschiz$条件为：$\forall x, y \in [-1,1]^n, \exists \rho > 0, s.t. |f(x) - f(y)| \leq \rho ||x-y||_2$,且此处$sgn(x)$为书本P98的输出为0,1的函数.)

(1) **[5pts]** 要求有构造的神经网络N的示意图.

(2) **[10pts]** 要求对构造的神经网络有简要的说明(写清每一层的线性组合形式，也就是nodes间的连接方式和对应的weights).

(3) **[10pts]**证明自己构造的神经网络的拟合误差满足要求.

Solution. 此处用于写解答*(中英文均可)*

由于f是一个$\rho - Lipschiz$函数，所以$\forall x, y \in [-1,1]^n, \exists \rho > 0, s.t. |f(x) - f(y)| \leq \rho ||x-y||_2$. 所以当把函数的定义域$[-1,1]^n$的每一维平均分为$2m$份，则可以切成$(2m)^n$个$blocks$，每一个$block$中

的$||x-y||_2 \le \frac{\sqrt{n}}{m}$.

因此，当$m > \frac{\sqrt{n}\rho}{\epsilon}$时，$|f(x)-f(y)| < \epsilon$，也就是说我们构造的神经网络只要能在前几层判断$x$在哪一个$block$中，然后在最后一层输出这一$block$中的$f(x)$的均值就可以满足误差要求.

构造的神经网络如下图所示：第一层和第二层的作用是在判断每一维$feature$的输入在哪一个$[\frac{k}{m}, \frac{k+1}{m}]$中；而第三层的作用就是判断输入的x是在哪一个$block$中；第四层输出那个$block$中的$f(x)$的均值作为神经网络的输出.

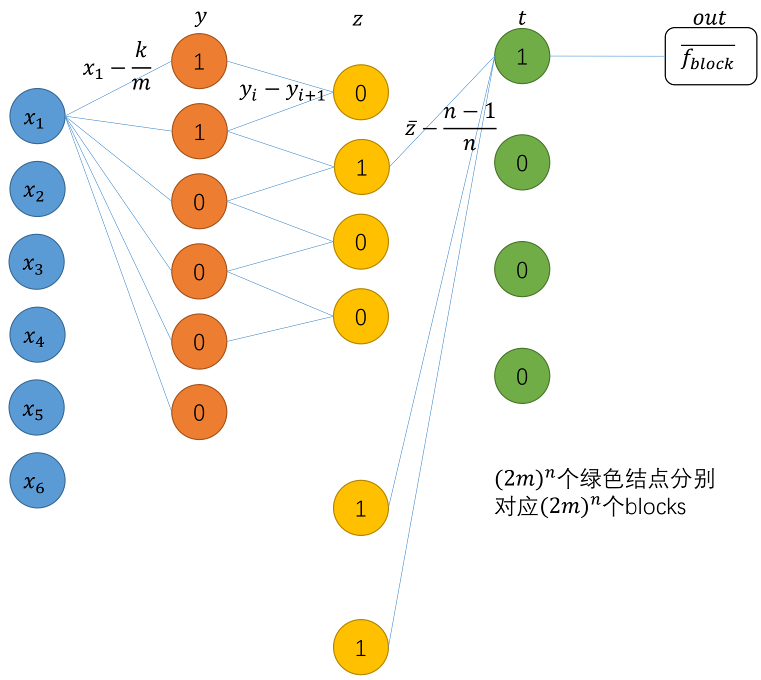

Figure 5: 神经网络示意图

4 [40pts] Neural Network in Practice

学习了课本第5章之后相信大家对神经网络有了初步的了解，复杂结构神经网络，即深度学习，在某些现实机器学习问题，如图像、自然语言处理等表现优异。本次作业旨在引导大家使用一种深度学习工具，快速搭建、训练深度神经网络，完成分类任务。

我们选取PyTorch为本次实验的深度学习工具，有了基础的深度学习工具，我们就能如同搭积木一样构建深度网络。PyTorch是Facebook开发的一种开源深度学习框架，有安装方便、文档齐全、构架方便、训练效率高等特点。所以本次作业的首要任务就是安装PyTorch，目前PyTorch仅支持Linux和MacOS操作系统，所以Window用户需要装一个Linux虚拟机或者直接安装Linux系统。PyTorch安装很方便，只需要在其主页中的Get Start一栏选择对应的环境设置，便能够一键安装，有GPU的同学也可以尝试安装GPU版本的PyTorch。为保证此次作业的公平性，只要求使用CPU进行网络训练，当然有条件的同学也可以尝试使用GPU进行训练。在助教批改作业时，助教会提供Python 2.7、3.5、3.6三种环境进行实验验证。

有了深度学习工具后，我们选取CIFAR10作为本次作业的训练任务。CIFAR10是一个经典的

图片分类数据集，数据集中总共有60000张32×32的彩色图片，总共有10类，每类6000张图片，其中50000张图片构成训练集，10000张图片构成测试集。幸运的是PyTorch中通过torchvision给用户提供了获取CIFAR10的方法，详细信息可见PyTorch的tutorial。此外关于CIFAR10分类准确率排行可见此链接。

有了工具了问题，接下来就是使用工具解决问题啦。本次作业的要求如下：

(1) **[15pts]** 首先我们跟随PyTorch的tutorial，用一个简单的卷积神经网络（Convolutional Neural Network, CNN），完成CIFAR10上的分类任务，具体要求如下：

- **[7pts]** 在代码实现之前，大家可能需要对CNN网络进行一定的了解，请大家自行查阅资料（PyTorch的tutorial中也有一点），并在实验报告中给出CNN的简介，主要回答什么是卷积层，什么是Pooling层，两者的作用分别是什么；
- **[8pts]** 接下来就是具体的代码实现和训练，tutorial中会手把手教你完成一次训练过程，tutorial中是使用SGD作为优化方法，请同学们自行调整epoch的大小和学习率，完成此次训练，请在实验报告中给出必要的参数设置，训练结果如最终的loss、在测试集上的准确率等；

(2) **[20pts]** 显然，这样一个简单的网络在CIFAR10上并不能取得令人满意的结果，我们需要选取一个更为复杂的网络来提升训练效果，在此小题中我们选取了CIFAR10准确率排行榜上排名第二的结构，此处有相关论文链接，为了方便大家实现，我们直接给出了网络结构如图6所示，请大家搭建完成此网络结构，并选择Adam为优化器，自行调整相关参数完成训练和预测，实验结果报告内容同第（1）小题；

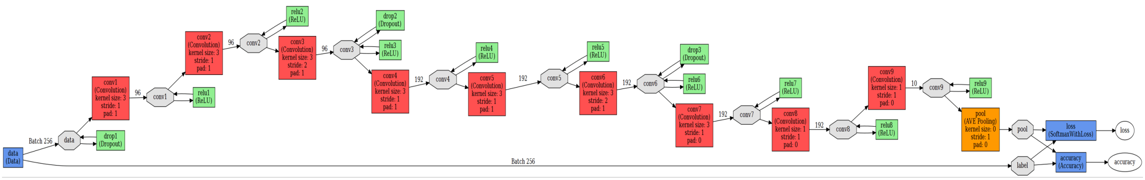

Figure 6: 待实现网络结构。

(3) **[5pts]** 通过上一题实验我们可能发现，即使使用别人提出的结构也不一定能达到和别人相同的训练效果，请大家试分析其中的原因是什么，并谈谈本次实验的感想，以及对深度学习调参的体会。

第四题相关问题.

- 很多作业可以多尝试一些参数设置，并且给出不同参数的结果对比
- 很多作业的网络结构，可以更复杂一点
- 希望同学们的实验报告尽量清晰，不要把实验结果直接完全copy上去十几页，看着很乱
- 希望同学们在实验过程中确认实验有没有收敛，并根据实验结果写一些见解，例如你认为什么时候过拟合了，等等