Introduction to Machine Learning
Homework 1

2021 年 10 月 4 日

Academic integrity

Our lesson cares much more on academic integrity. No matter who should do our utmost to handle the establishment of academic integrity standard including the host teacher and assistants of this lesson. We hope you will have the same faith with us.

(1) Discussion between students is allowing. The work named by yourself must be completed by your own hands. Any kind of Copying from existing documents will be seen as illegal.

(2) Any kind of Copying from other people's fruits of labour(Publication or Internet documents) will be accused of plagiarism. The score of plagiarists will be canceled. Please mark the authors if you cited any public documents of them;

(3) Highly resemble homework will be seen as Coping. No matter who you are, the one who copy or the one who is copied, both of your score will be canceled. Please protect your homework not to be copied by others actively.

Homework submission notes

(1) Please follow the submission methods on the website;

(2) If you are not follow the methods or your submission format are not correct. We will deduct some score of your homework;

(3) Unless some special cases, the submission over deadline will not be accepted and your score will be set as zero.

1 [30pts] Basic concepts

1.1 Probabiliy

Suppose Bob has been tested with a terrible disease. The event T and D represent a person has been tested positive for this disease and actually has this disease, respectively. According to statistics, we know:

$$\begin{cases} \Pr(\mathrm{T}|\mathrm{D}) & = 0.98 \\ \Pr(\mathrm{T}|\neg\mathrm{D}) & = 0.10 \\ \Pr(\mathrm{D}) & = 0.01 \end{cases} \tag{1.1}$$

He wants you to help him calculate the probability that he actually has the disease?

Solution: The target is Pr(D|T). According to Bayes Therom

$$\begin{aligned} \Pr(\mathrm{D}|T) = \frac{\Pr(\mathrm{T}|\mathrm{D})\Pr(\mathrm{D})}{\Pr(\mathrm{T})} &= \frac{\Pr(\mathrm{T}|\mathrm{D})Pr(\mathrm{D})}{\Pr(\mathrm{T}|\mathrm{D})\Pr(\mathrm{D}) + \Pr(\mathrm{T}|\neg\mathrm{D})\Pr(\neg\mathrm{D})} \\ &= \frac{0.98 * 0.01}{0.98 * 0.01 + 0.10 * 0.99} = 0.09007 \end{aligned} \tag{1.2}$$

1.2 Maximum likelihood estimation

We have an uneven coin, and the probability of tossing it heads up at random is p. Suppose you toss this coin 10 times, 8 of which are heads up. Please estimate p based on the existing information using MLE.

Solution: suppose that the random variable X is the times of the coin's heading up, $X \sim B(10, p)$, the log likelihood function is

$$\begin{aligned} LL(p|X=8) = \log Pr(X=8|p) &= \log \binom{10}{8} p^8 (1-p)^2 \\ &= 8\log p + 2\log(1-p) + const \end{aligned} \tag{1.3}$$

To maximize the likelihood function, let $\frac{\partial LL(p|X=8)}{\partial p} = 0$ i.e. $\frac{8}{p} - \frac{2}{1-p} = 0$, $p = 0.8$

1.3 Performance measure

We have a set of samples with binary classes (denoted as 0 and 1) and two classifers C_1 and C_2. For each sample, the classifier gives a score to measure the confidence that the classifier believes that the sample belongs to class 1. Below are the predicted results of two classifiers (C_1 and C_2) for 8 samples, their ground truth labels (y), and the scores for both classifiers (y_{C_1} and y_{C_2}).

y	1	0	1	1	1	0	0	1
y_{C_1}	0.62	0.39	0.18	0.72	0.45	0.01	0.32	0.93
y_{C_2}	0.34	0.12	0.82	0.89	0.17	0.75	0.36	0.97

(1) Calculate the area under the ROC curve (AUROC) for both classifiers C_1 and C_2.

(2) For the classifier C_1, we select a decision threshold $th_1 = 0.40$ which means that C_1 classifies a sample as class 1, if its score $y_{C_1} > th_1$, otherwise it classifies this sample as class 0. Calculate the confusion matrix and the F_1 score. Do the same thing for the classifier C_2 using a threshold value $th_2 = 0.90$.

Solution(1): 根据 RUC 曲线定义，将预测值按降序排列，先全部分类为反例（阈值为 1），然后依预测值排列顺序设定阈值，计算此时的 TPR，FPR，遍历完有限样例后得到坐标，绘制曲线。
编写程序绘制曲线如下所示：

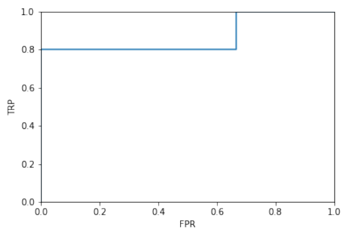


图 1: ROC1

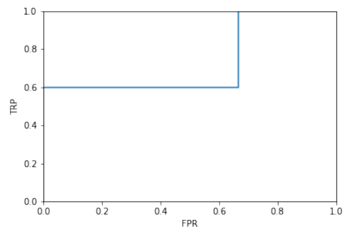


图 2: ROC2

根据所得坐标可计算得到，AUC 分别为：$\frac{13}{15}, \frac{11}{15}$。通过排序损失也可同样计算出 AUC 大小。

Solution(2): 对于分类器 C_1，设定阈值为 0.40，则 TP, FN, FP, TN 分别等于 $4, 1, 0, 3$，查准率 P，查全率 R 分别等于 $\frac{4}{4}, \frac{4}{5}$，相应的 $F_1 = \frac{2PR}{P+R} = \frac{8}{9}$

混淆矩阵: $\begin{bmatrix} 4 & 1 \\ 0 & 3 \end{bmatrix}$

对于分类器 C_2，设定阈值为 0.90，则 TP, FN, FP, TN 分别等于 $1, 4, 0, 3$，查准率 P，查全率 R 分别等于 $\frac{1}{1}, \frac{1}{5}$，相应的 $F_1 = \frac{2PR}{P+R} = \frac{1}{3}$

混淆矩阵: $\begin{bmatrix} 1 & 4 \\ 0 & 3 \end{bmatrix}$

2 [20pts] Linear model

Suppose you are given a data set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$. We want to use a regularized linear regression model to fit this data set, that is, to solve the following minimization problem:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 \tag{2.1}$$

where, $\mathbf{y} \in \mathbb{R}^m, \mathbf{X} \in R^{n\times d}$. Assume that $\mathbf{X}$ is column full-rank matrix.

1. Please give the closed-form solution for Eq.(2.1). You need to give your solution in detail.

2. The data set D is shown in the Table 1, where each sample has 3 dimensions (F_1, F_2, F_3). Please calculate the optimal solution for $\mathbf{w}$ when $\lambda = 1$.

F_1	2	9	8	8	2	8	4	1	3	5
F_2	9	3	3	8	1	4	3	8	3	3
F_3	1	1	1	1	1	1	1	1	1	1
y	290	1054	944	964	246	948	488	167	370	598

表 1: Training set for linear regression

Solution(1): 目标函数为 $\frac{1}{2}||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_2^2$, 目标函数对权重 $\mathbf{w}$ 求导得:

$$\begin{aligned} &\frac{\partial \frac{1}{2}\mathbf{y}^T\mathbf{y} + \frac{1}{2}\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - (\mathbf{y}^T\mathbf{X})\mathbf{w} + \lambda\mathbf{w}^T\mathbf{w}}{\partial \mathbf{w}} \\ &= \mathbf{X}^T\mathbf{X}\mathbf{w} - (\mathbf{X}^T\mathbf{y}) + \lambda\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda I)\mathbf{w} - (\mathbf{X}^T\mathbf{y}) \end{aligned} \tag{2.2}$$

令导数为 0, 由于 $\mathbf{X}$ 列满秩，所以 $\mathbf{X}^T\mathbf{X} + \lambda I$ 可逆.
闭式解为 $(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}(\mathbf{X}^T\mathbf{y})$
Solution(2)): 根据第一问得到 $\mathbf{w}$ 的闭式解，利用 python 程序求解得 $\mathbf{w} = [113.52, 5.66, 12.57]$，求解过程见附件。

3 [20pts] Decision tree

Suppose you are given data consisting of a training set of 5 examples and a test set of 4 examples. Each sample in the training and test set has three binary features (A, B, C) and one binary label (y).

(1) Using the training set (Table 2), construct a decision tree for the binary classification. Use the Information Gain (IG) as the decision criterion to select which attribute to split on. Show your calculations for the IG for all possible attributes for every split. (If there are multiple optimal features when splitting, please select the feature with the smallest alphabetical order.)

(2) Now evaluate the decision tree you have created on the test set (Table 3).

A	B	C	y
1	0	1	1
1	1	0	0
0	0	0	0
0	1	0	1
1	0	1	1

表 2: Training set for decision tree

A	B	C	y
0	0	0	0
0	1	1	1
1	1	1	0
1	0	0	0

表 3: Training set for decision tree

Solution: $Ent(D) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5}$

$Gain(D, A) = Ent(D) - \frac{3}{5}(-\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}) - \frac{2}{5}(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2})$

$Gain(D, B) = Ent(D) - \frac{2}{5}(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}) - \frac{3}{5}(-\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3})$

$Gain(D,C) = Ent(D) - \frac{2}{5} * 0 - \frac{3}{5}(-\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3})$

$Gain(D,C) \geq Gain(D,A) = Gain(D,B)$ 所以按照 C 的取值划分第一个节点，产生 D^1 和 D^2，其中 D^2 全部为正例，不需进一步划分。

对 D^1, $Ent(D^1) = -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}$

$Gain(D^1,A) = Ent(D^1) - \frac{1}{3} * 0 - \frac{2}{3}(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2})$

$Gain(D^1,B) = Ent(D^1) - \frac{1}{3} * 0 - \frac{2}{3}(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2})$

$Gain(D^1,A) = Gain(D^2,B)$，所以默认选取特征 A 作为节点，最后以 B 作为节点，得到的决策树模型如下：

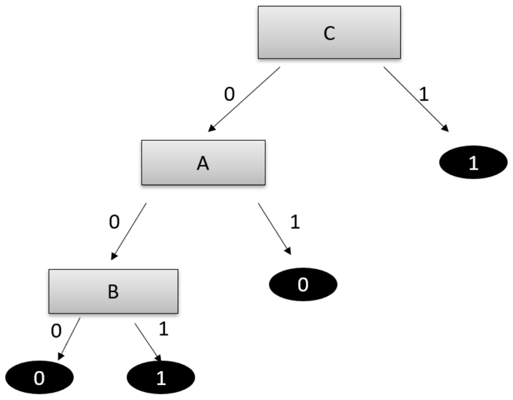


图 3: Decision Tree

(2) 在表 3 中的运行结果为：前三个测试样本预测正确，最后一个预测错误。

Precision $= \frac{1}{2}$

Recall $= 1$

$F_1 = \frac{2}{3}$

4 [30pts] Neural network

In this problem, you are asked to build a neural networks from scratch and examine performance of the network you just build on pendigits[1] data set. Here are some instructments listed below:

1. You are allow to use out-of-the-box deep learning tools (e.g., PyTorch, TensorFlow, ...) to build your model.
2. You don't have to implement deep and complex neural networks to achieve the state-of-the-art performance. However, brief performance comparisons between different architectures, different hyper-parameters, and different optimization methods are needed.
3. You need to submit your code and describe how to use them. Briefly showing your analysis, experimental results, and conclusions in this homework is also necessary.

Solution: 代码内容见附件 MLP.ipynb

分别尝试了线性模型，单层 MLP(16->32->10)，多层 MLP(16->64->32->16->10)；默认 epoch 为 100，学习率 1e-2，weight decay 1e-4，激活函数 ReLU()，每层默认做 BatchNormalization, optimizer 为 Adam

在测试集上分类预测正确率分别为 86.16%, 96.97%, 97.68%

在训练集和验证集上损失为

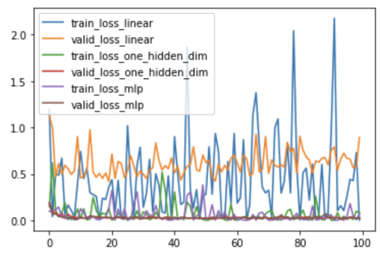
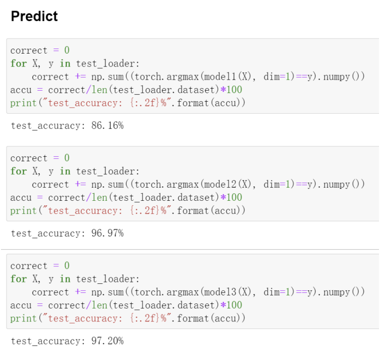


图 4: loss

[1]https://archive.ics.uci.edu/ml/machine-learning-databases/pendigits/



Predict

```
correct = 0
for X, y in test_loader:
    correct += np.sum((torch.argmax(model1(X), dim=1)==y).numpy())
accu = correct/len(test_loader.dataset)*100
print("test_accuracy: {:.2f}%".format(accu))
```

```
test_accuracy: 86.16%
```

```
correct = 0
for X, y in test_loader:
    correct += np.sum((torch.argmax(model2(X), dim=1)==y).numpy())
accu = correct/len(test_loader.dataset)*100
print("test_accuracy: {:.2f}%".format(accu))
```

```
test_accuracy: 96.97%
```

```
correct = 0
for X, y in test_loader:
    correct += np.sum((torch.argmax(model3(X), dim=1)==y).numpy())
accu = correct/len(test_loader.dataset)*100
print("test_accuracy: {:.2f}%".format(accu))
```

```
test_accuracy: 97.20%
```

图 5: result

下面以单层 MLP 为 baseline，调节超参数比较性能（测试集上预测正确率）：

学习率：

1e-1: 91.85%

1e-2: 96.14%

1e-3: 96.88%

1e-4: 96.57% (相应的，调高了 num epoch)

weight decay:

1e-1: 77.47%

1e-2: 90.31%

1e-4: 96.14%

1e-6: 97.20%

0: 97.26%

optimizer:

Adam: 96.14%

SGD: 95.28%

RMSProp: 96.31%

综上可见，该数据集比较容易训练，epoch 较小的情况，在验证集上的准确率已经很高，最终测试集上的准确率也不低。

同时，真实的函数并不复杂，仅用单层的 MLP 就已经达到很好的效果，用深层网络调节参数（最高达到 97.71%，应该可以通过调整网络架构，batch size 等超参数使预测正确率更高，但暂时并未尝试）

training loss 有一定波动，但实际波动值并不大，在验证集，测试集的预测效果都很好，说明数据的分布一致性很强。

（因为没有设定 random.seed, 所以在复现时存在一定不确定性, res 文件夹中存有单层 MLP 参数, 正确率 96.31%, 深层 MLP 参数, 正确率 97.68%, 可以通过 load_state 加载检验）