2022秋季学期人工智能课程
第二次作业
林耕宇 SA22225093

1、 关于二分类任务，请回答以下问题：

1） 损失函数是如何定义的？

使用了Logistic回归算法，这是一个二分类算法。

Logistic算法可以写为：

$$f(x)=\sigma(\sum_{i=1}^{n}w_ix_i+b)$$
$$P(\hat{y}=1|x)=f(x)$$
$$P(\hat{y}=0|x)=1-f(x)$$

使用交叉熵作为损失函数：

$$L(\omega)=\prod_{i=1}^{m}[f(x_i)]^{y_i}[1-f(x_i)]^{1-y_i}$$

对上式求导：

$$\ln L(\omega)=\sum_{i=1}^{m}(y_i\ln[f(x_i)]+(1-y_i)\ln[1-f(x_i)])$$

2） 评价指标有哪些？分别是怎么定义的？

单一指标有：

①Precision：$P=\frac{TP}{TP+FP}$

②Recall：$R=\frac{TP}{TP+FN}$

③Accuracy：$Acc=\frac{TP+TN}{TP+TN+FP+FN}$

④F1 score：是Precision和Recall的调和平均值，$\frac{2}{F_1}=\frac{1}{P}+\frac{1}{R}$

综合指标有：

①PRC（Precision Recall Curve）：

在不同的分类阈值下，以P为y轴，以R为横轴，画出不同分类阈值下的(P,R)对。

②ROC：

在不同的分类阈值下，分别计算“真正例率”TPR和“假正例率”FPR，$TPR=\frac{TP}{TP+FN}$，$FPR=\frac{FP}{TN+FP}$。以TPR为y轴，以FPR为x轴，画出不同分类阈值下的(TPR,FPR)对。

3） 混淆矩阵的行和列分别代表什么?

行：样本的真实标签

列：模型的预测标签

4） 混淆矩阵可以用于错误分析。在sk-learn中，混淆矩阵是，基于标签值和预测值，使用confusion_matrix函数计算得到的。请问这里的预测值是指的训练集的预测值，还是验证集的预测值，还是测试集的预测值？为什么?

我认为都可以，在不同的阶段，使用不同的混淆矩阵进行错误分析。

比如在确定超参数的时候，使用训练集和验证集寻找较好的超参。

比如在最终测试阶段，使用训练集和测试集的混淆函数来检查模型的泛化性。

5） Sk-learn中，通过precision_recall_curve函数可以得到precisions和recalls的值。请问，计算precisions和recalls时，precision_recall_curve函数的输入值是什么?

第一个参数是样本真实值，第二个参数是模型计算得到的预测值（是一个0~1之间的数）

1、 关于多分类任务，请回答以下问题：

1）利用分类器KNeighborsClassifier实现MNIST数据集的分类。假定该分类器的超参配置如下：“ algorithm='auto', leaf_size=30, metric='minkowski',

metric_params=None, n_jobs=None, n_neighbors=4, p=2,

weights='distance' ”

a) 请利用混淆矩阵进行错误分析，即分析该分类器在哪些图片上的泛化性能差。（可以将相关代码和运行结果的截图贴在作业中，并针对运行结果进一步分析从而得出结论）

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix

knn_clf = KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=4, p=2,
weights='distance')
knn_clf.fit(train_X, train_y)
y_train_pred = knn_clf.predict(train_X)


C1 = confusion_matrix(y_true=train_y, y_pred=y_train_pred)
print(C1.shape)
print(C1)
```

```
(10, 10)
[[5560    0    0    0    0    0    0    0    0    0]
 [   0 6277    0    0    0    0    0    0    0    0]
 [   0    0 5610    0    0    0    0    0    0    0]
 [   0    0    0 5708    0    0    0    0    0    0]
 [   0    0    0    0 5529    0    0    0    0    0]
 [   0    0    0    0    0 5040    0    0    0    0]
 [   0    0    0    0    0    0 5480    0    0    0]
 [   0    0    0    0    0    0    0 5790    0    0]
 [   0    0    0    0    0    0    0    0 5468    0]
 [   0    0    0    0    0    0    0    0    0 5538]]
```

通过训练集的混淆矩阵可以看出，模型的训练效果过分好了，存在一定的过拟合的可能。

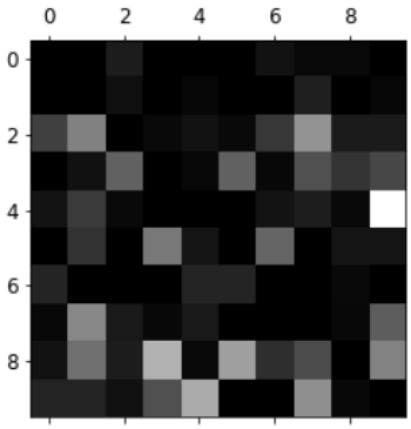
```
# 计算训练集的混淆矩阵

y_test_pred = knn_clf.predict(test_X)
C2 = confusion_matrix(y_true=test_y, y_pred=y_test_pred)
print(C2.shape)
print(C2)

# 画图：让对角线全为黑色，格子颜色亮说明分错的情况多

row_sums = C2.sum(axis=1, keepdims=True)
norm_C2 = C2 / row_sums
np.fill_diagonal(norm_C2, 0)
plt.matshow(norm_C2, cmap=plt.cm.gray)
save_fig("confusion_matrix_errors_plot", tight_layout=False)
plt.show()
```

```
(10, 10)
[[1336    0    3    0    0    0    2    1    1    0]
 [   0 1592    2    0    1    0    0    4    0    1]
 [   7   14 1327    1    2    1    6   16    3    3]
 [   0    2   11 1384    1   11    1    9    6    8]
 [   2    6    1    0 1254    0    2    3    1   26]
 [   0    5    0   12    2 1240   10    0    2    2]
 [   4    0    0    0    4    4 1383    0    1    0]
 [   1   16    3    1    3    0    0 1467    1   11]
 [   2   12    3   19    1   17    5    8 1276   14]
 [   4    4    2    9   19    0    0   16    1 1365]]
```



根据训练集和测试集的混淆矩阵，发现模型整体的泛化性能还不错。格子（4,9）最亮，说明模型容易将4误判为9，格子（9,4）、（8,3）、（8,5）、（9,7）、（2,7）都比较亮，说明模型在这些这些情况下容易出现误判、泛化效果不好。

b) 该分类器进行训练，然后计算其在测试集上的准确率。（可以将相关代码和运行结果的截图作为作业答案。）

```
from sklearn.metrics import accuracy_score

acc_score = accuracy_score(y_true=test_y, y_pred=y_test_pred)
acc_score
```

```
0.9731428571428572
```

在测试集上的准确率为0.9731.

2）多分类任务中，损失函数是什么？

多分类任务一般使用softmax作为神经网络的最后一层，然后计算交叉熵损失。

多分类的交叉熵损失可以表示为：

$$L = \sum_{i=1}^{k} y_i \log p_i$$

其中，k是分类数量；y_i是标签，如果类别是i，则$y_i = 1$，否则$y_i = 0$；p_i是神经网络的输出，也就是模型判断类别是i的概率。

3）多分类任务的常用评价指标有哪些？PRC和ROC曲线，可以用来评价多分类问题的性能吗？

多分类任务常用的评价指标有：平均AP（mAP）、Kappa系数、铰链损失、混淆矩阵。

可以使用，在多分类任务下，如果有几个类别就可以对应画出几条曲线。