MLSM 2021 - Quiz 5

12/11/2021 - 24/11/2021 at 23:59

---

(1) (1.0 pt.) In most of the implementation of the Multi-layer Perceptron algorithm, e.g. Tensorflow, Pytorch, Sklearn, the RELU function, $\max(0, x)$, is used as default activation. Can the Stochastic Gradient Descent(SGD) for MLPs be applied with some modification if the RELU is the available activation function? Which of the following statements is true?

Hint: Recall Question 2 of Quiz 3 comparing the algorithm of the Perceptron and the Logistic Regression,

(1) Since the RELU is not differentiable, it can not be applied in the SGD.

(2) The SGD can be applied, where the derivative of the activation is replaced with a step function:

$$h(u) = \begin{cases} 0 & u \leq 0, \\ 1 & u > 0. \end{cases}$$

(3) The SGD can be applied, where in place of the derivative of the activation we need to use this function:

$$H(u) = \begin{cases} 0 & u \leq 0, \\ -u & u > 0. \end{cases}$$

(2) (1.0 pt.) Assume 21 base learners with independent true risk 0.45 ($\epsilon = 0.45$), are grouped to form a strong learner via majority voting. What is the probability of having an incorrect aggregated prediction (using the formula given on the slide "Why can model combination work? A thought experiment" of Lecture about "Ensemble Learning"):

(1) 0.5

(2) 0.16

(3) 0.32

(4) 0.21

(3) (1.0 pt.)

Which option is NOT correct in AdaBoost

(1) The weight of the training examples, $D_t(i)$, can be arbitrary numbers.

(2) The sum of the weights $\sum_{i=1}^{m} D_t(i) = 1$ and $D_t > 0$ for any t, thus they can be interpreted as a probability distribution on the training examples.

(3) The weights might increase or decrease in each iteration for all examples.

(4) (2.0 pt.)

In this question the performance of three different classifiers mentioned in the lecture are compared. In the comparison this example of the Sklearn package:

`https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html`

can be used as a prototype of this kind of tasks. In our problem, we have only this data set:

- from sklearn.datasets import load_breast_cancer.

The three learners are the following ones:

- from sklearn.neural_network import MLPClassifier
- from sklearn.ensemble import AdaBoostClassifier
- from sklearn.ensemble import GradientBoostingClassifier

Those learners need to be parametrized in this way to receive comparable results:

- MLPClassifier(alpha=1, max_iter=100, random_state=1)
- AdaBoostClassifier(random_state=1)
- GradientBoostingClassifier(n_estimators=200, learning_rate=1, max_depth=1, random_state=1)

The data set is cut once into training and test following the Sklearn example code, but the test size is selected as 0.5, and the random state is set to 1. The data is scaled by the StandardScaler function of the Sklearn.

The task is to report that classifier which provides the best result at this given configuration. In comparing the performance you need to use by F1 score instead of the default one.

Hint: You could use and modify the Sklearn example, and all available functions provided by that package. Be careful, the example code contains several details that is not important to answer this question.

(1) MLPClassifier
(2) AdaBoostClassifier
(3) GradientBoostingClassifier