MLSM 2021 - Quiz 5
12/11/2021 - 24/11/2021 at 23:59

---

(1) (1.0 pt.)
Which option is NOT a valid statement relating to the Multi-layer perceptrons.

(1) If the activation functions are linear in all layers then that network can be replaced with one containing only a linear function.
(2) The function NOT(XOR(.,.)) can not be represented by perceptron of one layer.
(3) It is not NP-hard to find the parameters minimizing the empirical error for a network with one single hidden layer that contains 5 neurons.
(4) Binary classification can be realized by applying softmax-function.

(2) (1.0 pt.) Assume 25 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) 0.50
(2) 0.31
(3) 0.15
(4) 0.22

(3) (1.0 pt.)
Which option is NOT a valid statement relating to the ensemble learning.

(1) The loss function of the LogitBoost is not equal to logarithm of the loss of the AdaBoost
(2) The sum of the weights in the AdaBoost satisfies these rules: $\sum_{i=1}^{m} D_t(i) = 1$ and $D_t > 0$ for any t. Thus they can be interpreted as a probability distribution on the training examples.
(3) An ensemble of learners taken from a hypothesis class can only learn patterns which are learnable by the elements of that class.
(4) The VC-dimension of the hypothesis class corresponding to the AdaBoost can grow faster in the number of weak learners, T, than the VC-dimension of the weak learners, d.

(4) (2.0 pt.)
The performance of four different classifiers mentioned in the lectures are compared in this question. In the implementation, 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 three data sets:

```
from sklearn.datasets import load_breast_cancer
from sklearn.datasets make_moons
from sklearn.datasets make_circles
```

The dataset loading functions are parameterized in this way

```
load_breast_cancer(return_X_y=True),  ## X input, y output
make_moons(n_samples = 100, noise=0.3, random_state=1),
make_circles(n_samples = 100, noise=0.2, factor=0.5, random_state=1)
```

The four learners are the following ones:

```
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
```

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

```
MLPClassifier(hidden_layer_sizes = (100,), alpha=0.0001, max_iter=200, \
                    random_state=1),
AdaBoostClassifier(n_estimators = 100, random_state=1),
GradientBoostingClassifier(n_estimators=100, learning_rate=1, \
                                  max_depth=1, random_state=1), \
RandomForestClassifier(max_depth=4, random_state=1)
```

The data sets are cut only 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 task is to report that classifier which provides the best result on these three datasets. In the comparison, the performance of the classifiers is measured by the average of the *Area Under the Receiver Operating Characteristic Curve (ROC AUC)* computed on each datasets separately.

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

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