海大資工 AI 機器學習作業報告範本

資工 3B 00857125 葉冠昊

(一)實驗結果

系統代碼	執行環境	實驗資料	正確率
A	Weka	10-fold cross-validation	99.61%
A	Weka	訓練 vs 測試 4:1	99.7%
B	Weka	10-fold cross-validation	98.11%
B	Weka	訓練 vs 測試 4:1	98.52%
C	Weka	10-fold cross-validation	95.2%
C	Weka	訓練 vs 測試 4:1	95.73%
D	Weka	10-fold cross-validation	94.73%
D	Weka	訓練 vs 測試 4:1	94.26%
E	Weka	10-fold cross-validation	95.11%
E	Weka	訓練 vs 測試 4:1	95.29%
F	Weka	10-fold cross-validation	99.61%
F	Weka	訓練 vs 測試 4:1	99.55%
G	Weka	10-fold cross-validation	95.2%
G	Weka	訓練 vs 測試 4:1	95.73%
H	Weka	10-fold cross-validation	99.23%
H	Weka	訓練 vs 測試 4:1	98.83%
I	Scikit-learn	訓練 vs 測試 4:1	99.7%
J	Scikit-learn	訓練 vs 測試 4:1	99.55%
K	Scikit-learn	訓練 vs 測試 4:1	95.73%
L	Scikit-learn	訓練 vs 測試 4:1	98.08%
M	Scikit-learn	訓練 vs 測試 4:1	96.32%
N	Scikit-learn	訓練 vs 測試 4:1	7.79%
O	Scikit-learn	訓練 vs 測試 4:1	96.61%
P	Scikit-learn	訓練 vs 測試 4:1	92.94%
Q	Scikit-learn	訓練 vs 測試 4:1	95.58%
R	Tensorflow	訓練 vs 測試 4:1	95.15%
S	Tensorflow	訓練:驗證:測試 3:1:1	97.05%
T	Tensorflow	訓練:驗證:測試 3:1:1	98.23%

(二)系統描述

Weka

系統 A：RandomForest (tree)

系統 B：RandomTree (tree)

系統 C: DecisionStump (tree)

系統 D: MultiClassClassifier (meta)

系統 E: NaiveBayes (bayes)

系統 F: LogitBoost (meta)

系統 G: LWL (lazy)

系統 H: DecisionTable (rules)

Scikit-learn

系統 I: Decision Tree

系統 J: RandomForest

系統 K: SVM

系統 L: AdaBoost

系統 M: KNeighbors

系統 N: Naïve bayes GaussianNB

系統 O: Naïve bayes MultinomialNB

系統 P: Naïve bayes BernoulliNB

系統 Q: Naïve bayes ComplementNB

Tensorflow

系統 R: Tensorflow.estimator.DNNClassifier

Hidden_units(30, 10) step = 5000 batch_size = 256

系統 S: Dense(128, activation=tf.nn.relu) epochs = 50
Dropout(0.1) batch-size = 8
Dense(128, activation=tf.nn.relu) optimizer = Adam
Dropout(0.1)
Softmax()

系統 T: Keras Tuner 尋找最佳超參數(神經元數 dropout)
Dense(64, activation=tf.nn.relu) epochs = 50
Dropout(0.05) batch-size = 8
Dense(128, activation=tf.nn.relu) optimizer = Adam
Dropout(0.2) learning_rate = 0.01 (keras tuner 找尋)
Softmax()

(三)結論

Weka:

在上述系統中，當實驗資料是使用 10-fold cross-validation 時，集成式學習的 Random Forest 和元方法中的 LogitBoost 可以獲得最好的準確率，而同是元方法中的 MultiClassClassifier，在此實驗資料獲得了最差的準確率。

當實驗資料使用訓練 vs 測試 4:1 時，為集成式學習的 Random Forest 獲得了最高的準確率，而元方法中的 MultiClassClassifier，在此實驗資料也獲得了最差的準確率。

在此實驗資料和測試的系統中，集成式學習中使用 Bagging 的 Random Forest 獲得了最高的準確率。而比較在使用 10-fold cross-validation 和訓練 vs 測試 4:1 時，因應不同的模型，兩種方法的準確度沒有一方一定準於另一方的表現。

Scikit-learn:

在上述系統中，皆使用訓練 vs 測試 4:1 的方式，其中使用 Decision Tree 時會獲得最好的準確率，而使用 Naïve bayes GaussianNB 時則獲得了最差的準確率，相較於 SVM、貝氏分類器和 KNN，使用任何 Tree 或集成式學習來分類都可獲得較好的分類效果，而也可從不同的環境中的 Random Forest 分類器發現，即便使用的實驗資料相同，在不同環境上也會有些微的誤差，詢問老師後得出可能因為在不同環境的分類器上程式碼不同導致的。

Tensorflow:

在上述系統中，使用 Keras Tuner 自動執行神經網路的超參數調整獲得了最高的準確率，並使用 dropout 來防止可能因過度擬合，導致了模型在訓練集的表現較佳，但在測試集（即從未見過的數據）可能表現較不好的機率。而直接使用 Tensorflow.estimator 中的 DNNClassifier 則獲得了最低的準確率。

總結: 由上述全部使用的系統中，效果最好的仍為使用 Tree 來分類的系統，因此在資料集較單純的情況下，使用 Tree 來分類就可獲得較好的訓練效果了，在此資料集中使用神經網路或其他模型則不會有那麼高的準確率。