CSCI 5512: Artificial Intelligence II (Fall '19)

Homework 4

(Due Thu, Dec. 5, 11:59 pm central)

1. **(70 points)** [Programming Assignment] In this problem, we will consider classifying the following datasets: breast cancer[1], iris[2], handwritten digits[3], and wine[4]. Using the scikit-learn[5] machine learning library, learn a model for each of the following algorithms:

   - Logistic regression,
   - Perceptron,
   - Linear support vector machine (SVM),
   - k-nearest neighbor (KNN).

   For each algorithm, use 5-fold cross validation to tune the following hyperparameters (start with the recommended range but adjust as necessary):

   - Logistic regression[6]: $C \in [1e-5, 1e-4, 1e-3, 1e-2, 0.1, 1, 10, 100, 1000]$
   - Perceptron[7]: set penalty = 'l2' and $\alpha \in [11e-5, 1e-4, 1e-3, 1e-2, 0.1, 1, 10, 100, 1000]$
   - SVM[8]: $C \in [1e-5, 1e-4, 1e-3, 1e-2, 0.1, 1, 10, 100, 1000]$
   - KNN[9]: $k \in \{6x+1\}$ for $x \in \{0, 1, \ldots, 20\}$.

   For each algorithm, dataset, and hyperparameter, plot the mean classification error rate and standard deviation (as error bars) across the 5 folds. For each algorithm and dataset, choose the 'best' hyperparameter and explain your choice. Submit a single python file named `prob1.py` which takes no arguments and runs and displays plots for each algorithm and dataset.

2. **(15 points)** When training neural networks, we often use the following error function

$$Error = (correct - output)^2.$$

   Give two specific reasons (along with explanations) why the above error function is preferred over the following error function:

$$Error = |correct - output|.$$

---

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html
[5] https://scikit-learn.org/stable/
[6] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[7] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html
[8] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
[9] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

3. **(15 points)** The neural networks we discussed in class had every layer connected to only the next layer. Suppose we were to modify this to allow either: (i) layers can connect to future layers (left figure) or (ii) layers can connect back to previous layers (right figure). Which of these modifications is more difficult to adapt our learning method (gradient descent) to fit? Explain your reasoning (a general statement) and provide a concrete example to back up your thoughts (a specific example).
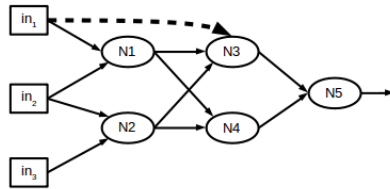
(1) to future
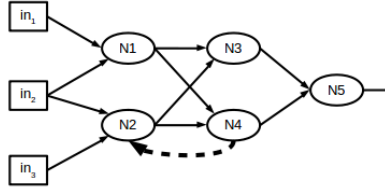
(2) to previous

Instructions

Please follow these instructions carefully. Code submitted without adhering to these instructions will not receive any credit. Note: you can implement the algorithms above but it is strongly recommended to use scikit-learn instead.

For the programming question, you have to submit the code as required by the problem and only Python 3.6 will be accepted, any other language will receive zero credit. The program must run on the CSE labs machines and will not receive credit if it fails this.