Assignment 6 (Sol.)

Introduction to Data Analytics

Prof. Nandan Sudarsanam & Prof. B. Ravindran

1. On a particular data set, we use the ensemble method approach to building a predictor and achieve state of the art performance. Is it possible for some of the individual models in this ensemble to have poor performance as measured on the training data?

   (a) no

   (b) yes

   Sol. (b)
   As was discussed in the lectures, diversity among the constituent models is desirable when building an ensemble predictor. This can lead to some of the individual models not exhibiting good performance considering all of the training data, but their good performance on critical subsets of the training data may result in good performance for the ensemble as a whole.

2. With regards to bagging and boosting, which among the following are true?

   (a) the different learners in bagging can be trained in parallel

   (b) the different learners in boosting can be trained in parallel

   (c) individual classifiers in bagging are trained with all data points in the training set

   (d) individual classifiers in boosting are trained with all data points in the training set

   Sol. (a), (d)
   In bagging, we sample with replacement, which indicates that not all the points in the training data set are available to the individual classifiers. Also, once the samples from the training data set have been created, each individual classifier can be trained on its own sample in parallel with the other classifiers. This however is not possible in boosting as each individual classifier is dependent on the previous classifier due to the re-weighting of points performed after each classifier is trained.

3. Can the boosting technique be applied on regression problems? Can bagging be applied on regression problems?

   (a) no, no

   (b) no, yes

   (c) yes, no

   (d) yes, yes

Sol. (d)
Ensemble methods are not tied to the classification problem, and can be used for regression as well.

4. In the general context of classification, re-weighting the data points (relative to an original training data set where the points are un-weighted) can lead to

   (a) change in the underlying optimisation problem that is solved

   (b) change in the positions of data points in the feature space

   (c) change in the decision surface generated by the classifier

   (d) change in the nature of the data set from being linearly separable to becoming linearly non-separable (in case the original data was linearly separable)

   Sol. (a), (c)
   Consider the case of support vector machines. Suppose a specific data point was re-weighted so that its new weight is 2 (originally all data points were un-weighted, i.e., had weight of 1). This is as good as there being two instances of the same data point, and assuming that this data point falls on the wrong side of the separating hyperplane, the error contributed by this point doubles. Hence, the underlying optimisation problem changes, with their being more of an emphasis to reduce the error on this specific point. Solving this modified optimisation problem may lead to a change in the separating hyperplane (compared to the one for the original data set). Note however, that re-weighting does not result in any change in the positions of the points in the feature space, hence if the original training data was linearly separable, the re-weighted data set remains linearly separable.

5. If one feature (compared to all others) is a very strong predictor of the class label of the output variable, then all of the trees in a random forest will have this feature as the root node.

   (a) false

   (b) true

   Sol. (a)
   In random forests, due to the random subset of features selected at each split, while constructing individual trees, many of the trees will not have this feature as the root even if it is the most important feature for predicting the label of the target variable.

6. Suppose we have 2-class training data which is linearly separable. We use the perceptron training algorithm to build a classifier. What is the number of possible solutions that can be obtained through this method?

   (a) 1

   (b) depends on the size of the margin separating the data points belonging to the two classes

   (c) infinite

   (d) depends on the size of the margin separating the data points belonging to the two classes and the learning rate parameter

   Sol. (c)
   Any linear decision boundary (of which there are infinite) that separates the data belonging to the two classes can theoretically be obtained as a result of the perceptron training algorithm.

7. By using a linear activation function in the output layer of a neural network for solving regression tasks, are we constraining the resultant model to be linear in the input features?

   (a) no

   (b) yes

   Sol. (a)
   Note that in this case, the linear functions in the output layer still take as input non-linear outputs of the hidden layer units.

8. In the backpropogation algorithm, the gradient of the error with respect to the weight vector is itself a vector. What does the direction of this vector indicate?

   (a) it points in the direction of steepest decrease in the error

   (b) it points in the direction of steepest increase in the error

   (c) it indicates that component of the weight vector that results in maximum error

   (d) it indicates that component of the weight vector that results in minimum error

   Sol. (b)
   The gradient of a function points in the direction of the greatest rate of increase of the function. In backpropogation, we are calculating the gradient of the error function. This is the reason why we take the negative of the calculated gradient when computing the updates in the backpropogation algorithm.

9. Given a multi-class data set, you choose to use an artificial neural network to build a classification model for this data. How would you determine the number of hidden layer nodes to use for this task?

   (a) same as the number of input layer nodes

   (b) same as the number of output layer nodes

   (c) through cross validation

   (d) maximum of the number of input and output layer nodes

   Sol. (c)
   In general, there is no direct way to glean from the data, the number of hidden layer nodes that would be required. Thus, we treat this as any other parameter, and use the cross validation approach to arrive at a suitable number for the task at hand.

10. In the lectures, we saw how to train a 7 layer auto encoder network. In case we wanted to perform classification on the data used for training this network, while making use of the trained network, a suitable approach would be to

    (a) add an additional eighth layer on top of the 7 layers as the output layer and train the entire network for the classification task

    (b) add an additional eighth layer on top of the 7 layers as the output layer and only modify the weights between layers 7 and 8 in training for the classification task

    (c) discard the top 3 layers, add an additional layer on top of the 4^{th} layer as the output layer and train the entire network for the classification task

(d) discard the top 3 layers, add an additional layer on top of the 4^{th} layer as the output layer and only modify the weights between layers 4 and 5 in training for the classification task

Sol. (c)
Counting from the bottom, the fourth layer of the network contains the compact representation of the input data that we used the greedy unsupervised layerwise pretraining technique to obtain. Post finetuning, we can discard the top 3 layers, and add an additional layer on top of layer 4 which will act as our output layer. Next we train the network as we would normally do using backpropogation and modifying the weights of all the layers.

Note that just modifying the weights between layers 4 and 5 may not allow us to obtain a classifier with satisfactory performance. This is because up till this stage, the previous layers have been trained only for the auto-encoder task, and may need to be modified further to allow for good performance in the classification task.