EE6550 Machine Learning, Spring 2017

Homework Assignment #5

Please submit your programs, including source codes, report (in pdf format) and user manual (in pdf format), to iLMS by 23:59 on June 19th Monday. Late submission will not be accepted unless the instructor gives a pre-approval. You are encouraged to consult or collaborate with other students while solving the problems, but you will have to turn in your own solutions and programs with your own words and work. If you find any resources in the internet to assist you, you should understand but not copy them. Copying will not be tolerated.

(15%) **Programming Problem:**

Implementation of a feedforward neural network learning algorithm by the stochastic gradient descent (SGD) algorithm described on page 26 in the slides of lecture #12.

Input:

1. A data file which contains a labeled training sample S. This labeled training sample is used to train the SGD algorithm which will return a hypothesis h_S^{SGD} represented by the weight vector $\boldsymbol{w}$.
   - In this homework assignment, it is suggested to use 5-fold cross-validation to determine the model parameters such as the depth of the feedforward neural network and the value of the regularization parameter λ.
   - In the cross-validation, the validation fold will be used to provide a stopping condition of the SGD algorithm. While the training error will continue to decrease as the number of iterations increases, the validation error may decrease in the early iterations but then starts to increase in the later iterations. You have to stop the iteration at the turning point to yield the lowest validation error and obtain the trained weights.
2. A data file which contains a labeled testing sample $\tilde{S}$. This labeled testing sample is used to evaluate the performance of the returned hypothesis h_S^{SGD} from the SGD algorithm based on the labeled training sample S.
3. The architecture $(V = \cup_{t=0}^{T} V_t, E, \sigma)$ of a feedforward neural network.
   - Except the output layer, each layer should have one constant neuron which outputs the value 1.
   - The total number of weights, which is $\sum_{t=0}^{T-1}(k_t+1)k_{t+1}$, is no greater than 1/15 of the size of the training dataset. (After the model selection by 5-fold cross-validation, it is suggested to partition the training dataset into two subsamples with the ratio: 2:1. The larger subsample is used to train the feedforward neural network with the selected model parameters, while the smaller subsample is used as the validation set to provide the stopping criterion of the SGD algorithm. Effectively, the total number of weights is no greater than 10% of the number of items used for training the neural network.)
   - It is required to consider architectures with T at least to be $3, 4, 5, 6, 7$.

4. The regularization parameter λ.
   - It is suggested to consider $\lambda \in [0, 1]$.
   - You must consider the case of $\lambda = 0$.
5. $\boldsymbol{w}_0$: suggested center of the initial weight vector. If no prior information, set $\boldsymbol{w}_0 = \mathbf{0}$.
6. N: number of iterations.

Model Selection:

1. The depth of the network and the size of each layer.
2. The regularization parameter λ which is used in the minimization of $R(\boldsymbol{w}) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2$.

Options:

- $\{\eta_i\}_{i=1}^{N}$: step size sequence. There are two popular choices:
  - $\eta_i = \eta \ \forall\ i$: fixed step size.
  - $\eta_i = \frac{\alpha}{\sqrt{i}}$: variable step size for some $\alpha > 0$.

Initialization:

- The initial weight vector $\boldsymbol{w}^{(1)}$ is chosen at random around the center $\boldsymbol{w}_0$ by an algorithm $\textsc{Random}(\boldsymbol{w}_0)$. It is a common practice to set $w_j^{(1)} = w_{0,j} + \varepsilon_j$ where ε_j is drawn i.i.d. from a probability distribution on the range $[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$ with zero mean with n the dimension of the input space. This probability distribution can be a uniform distribution on $[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$ or obtained by sampling a Gaussian distribution with zero mean and variance $1/n$ but deleting all samples out of the range $[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$.

Modules:

1. $\textsc{Sample}(S)$: an algorithm to take a labeled item $(\boldsymbol{x}^{(i)}, \boldsymbol{c}(\boldsymbol{x}^{(i)}))$ from the sample S. There are three possible ways:
   - Round-robin: first ordering the sample S at random and once the order is fixed, selecting a labeled item form S in this order and around.
   - Purely random: selecting a labeled item form S at random in each iteration.
   - Round-robin once and then purely random.
2. $\textsc{Backpropagation}(\boldsymbol{x}^{(i)}, \boldsymbol{c}(\boldsymbol{x}^{(i)}), V, E, \sigma, \boldsymbol{w}^{(i)})$: an algorithm to calculate a conditionally unbiased estimate of the gradient $\nabla R(\boldsymbol{w}^{(i)})$ of the risk R at the given weight vector $\boldsymbol{w}^{(i)}$ with a labeled item $(\boldsymbol{x}^{(i)}, \boldsymbol{c}(\boldsymbol{x}^{(i)}))$ drawn from the sample S. A pseudocode is given on pages 44-46 in the slides of lecture #12.

Output:

1. The best architecture of the feedforward neural network from depth $= 2, 3, 4, 5, 6$.
2. The best regularization parameter λ in $[0, 1]$.
3. The hypothesis h_S^{SGD} returned by the SGD algorithm with the best performing $\boldsymbol{w}^{(i)}$ on a validation set.

- As mentioned in above, after the model selection by 5-fold cross-validation, it is suggested to partition the training dataset into two subsamples with the ratio: 2:1. The larger subsample is used to train the feedforward neural network with the selected model parameters, while the smaller subsample is used as the validation set to provide the stopping criterion of the SGD algorithm.

4. Performance of the returned hypothesis h_S^{SGD} on the labeled testing sample $\tilde{S}$ kernel.

What to submit? You should submit the following items:

1. The source codes of your SGD learning algorithm.
   Please indicate which programming language you have used to write the programs in the title of the submission entry with one of the following formats:
   - HW5_yourstudentid_yourname_matlab (The environment you use should be compatible with the version licensed to the NTHU Computer Center.)
   - HW5_yourstudentid_yourname_python36
   - HW5_yourstudentid_yourname_cpp14
   - HW5_yourstudentid_yourname_c11

   This will facilitate the distribution of homeworks to graders for grading. Please have your code compilable/interpretable by a standard compiler/interpreter environment: Matlab (NTHU CC), Python3.x, C++14, and C11.

2. A printed report consisting of at least:
   (a) a table of the 5-fold cross-validation error $\hat{R}_{CV}(V = \cup_{t=0}^{T} V_t, E, \sigma, \lambda)$ as a function of the architecture $(V = \cup_{t=0}^{T} V_t, E, \sigma)$ and the regularization parameter λ and a discussion of how you determine the model selection from such a table;
   (b) an output file which represents the hypothesis h_S^{SGD} returned by the SGD learning algorithm;
   (c) the performance of the returned hypothesis h_S^{SGD} on the labeled testing sample $\tilde{S}$.

   Please submit your report in pdf format (please do not submit word files).

3. A user manual which should include instructions of
    (a) how to compile the source code with a standard compiler/interpreter;
    (b) how to run, i.e., execute the SGD learning algorithm, including the required formats of input parameters or data files;
    (c) how to use an output file which represents a returned hypotheses to do the testing;
    (d) what results are reported.

    These instructions should support the test scenarios in the grading session. Please submit your manual in pdf format (please do not submit word files).

Test scenarios in the grading session: the grader will test your algorithm with your source codes by the following procedure:

1. inputting a training data file, which contains a labeled training sample S. This labeled training sample is used to train the SGD learning algorithm which will return a hypothesis h_S^{SGD} after the 5-fold cross-validation.
2. inputting a testing data file, which contains a labeled testing sample $\tilde{S}$. This labeled testing sample is used to evaluate the performance of the returned hypothesis h_S^{SGD} from the SGD learning algorithm based on the labeled training sample S.
3. doing 5-fold cross-validation to determine the best architecture $(V = \cup_{t=0}^{T} V_t, E, \sigma)$ and the regularization parameter λ to minimize the cross-validation error $\hat{R}_{CV}(V = \cup_{t=0}^{T} V_t, E, \sigma, \lambda)$.
4. checking the output table of the cross-validation error $\hat{R}_{CV}(V = \cup_{t=0}^{T} V_t, E, \sigma, \lambda)$ as a function of the architecture $(V = \cup_{t=0}^{T} V_t, E, \sigma)$ and the regularization parameter λ and the obtained best model.
5. checking if there is an output file SGD_hypothesis_header.csv.
6. inputting the obtained best architecture and regularization parameter to a prepared reference SGD learning program to return a learned hypothesis h_S^{SGD}.
7. Comparing the testing performance between your codes and the prepared reference SGD learning program.
8. checking the performance of the returned hypothesis h_S^{SGD} on the labeled testing sample $\tilde{S}$ by inputting the your output file SGD_hypothesis_header.csv to a prepared program.