Mini Project for the course - Data Structures and Algorithms

In this mini-project, the student will implement the algorithm for hyperparameter optimization given in [1].

Description: Using the linear model $f(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{x}$ the aim is to solve the regression problem and tune the hyperparameters using gradient based approach mentioned in [1]. Given the data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ for the regression problem, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, the idea is to solve the following problem formulation:

$$\min \quad \frac{1}{2}\sum_{i=1}^{n}(f(\boldsymbol{x}_i;\boldsymbol{w}) - y_i)^2 + \frac{1}{2}\sum_{j=1}^{d}\lambda_j w_j^2$$

where $\lambda_1, \lambda_2, \ldots, \lambda_d$ are tunable non-negative hyperparameters and $\boldsymbol{w} = (w_1, w_2, \ldots, w_d)^T$.

The implementation involves differentiation of Cholesky Factorization as detailed in [1]. The implemented algorithm will be tested on "median price prediction problem" and the corresponding Boston housing dataset details are available at [2,3]. The data is 13 dimensional. The first 456 examples in the dataset will form the training set. The remaining examples are equally split into the validation set and the test set. It is expected to plot the training set error, validation set error and test set error (all in one graph) as a function of number of training iterations. One training iteration amounts to one forward and backward pass.

It is also proposed to extend these ideas to a one hidden layer neural network with hyperbolic tangent activations, linear output layer and squared error loss. Evaluation of this model will be done on the same dataset and similar graphs will be plotted.

References:

[1] Y. Bengio. Gradient based optimization of hyperparameters. *Neural Computation*, vol 12, pp 1889-1900, 2000.

[2] `http://lib.stat.cmu.edu/datasets/boston`

[3] `https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html`