Machine Learning

JACOBS UNIVERSITY

Spring Semester 2022
Prof. Dr. Peter Zaspel
Nitesh Khatiwada, Tin Nguyen, Diwas Pandey

Assignment Sheet 6.

Submit on **Tuesday, March 22, 2022, 10:00**.

Excercise 1. (Gradient descent on paper)

Consider the following training data:

$$\mathcal{T} = \left\{ \left((-1,0)^\top, 3\right), \left((1,3)^\top, -1\right), \left((-2,1)^\top, 0\right), \left((0,4)^\top, -2\right) \right\}.$$

We want to compute the predictor from linear regression by least squares. However, instead of using Theorem 4.1, we use batch gradient descent to compute the coefficient vector $\hat{\beta}$.
Choose the initial guess $\beta^{(0)} = (0,0,1)^\top$ and calculate the first two steps of batch gradient descent with learning rate $\eta = 0.25$.

(4 Points)

Excercise 2. (Gradient descent variations for other models)

In this task, we consider again the training data

$$\mathcal{T} = \left\{ \left((4,1)^\top, 2\right), \left((2,8)^\top, -14\right), \left((1,0)^\top, 1\right), \left((3,2)^\top, -1\right) \right\}.$$

However, this time we do not use the linear model. Instead we use a quadratic model

$$f(\mathbf{X}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \beta_4 X_1^2 + \beta_5 X_2^2$$

and stick to the L_2 loss.

a) Derive the gradient $\nabla_\beta L_2(y_i, f(\mathbf{x}_i))$ of the loss with respect to the coefficient vector.

b) Use the initial guess $\beta^{(0)} = \mathbf{0}$ and the learning rate $\eta = 0.1$ and perform the first two steps of stochastic gradient descent. Recall that in stochastic gradient descent, we at some point pick random samples from the training set. In order to get a unique solution in this task, we assume that a random selection of samples from the training set would give you the training samples in their original order.

c) Use the initial guess $\beta^{(0)} = \mathbf{0}$ and the learning rate $\eta = 0.1$ and perform the first two steps of mini-batch gradient descent with a batch size of $N_b = 2$. Use the same approach for the randomization part as in the previous sub-task.

(4+2+2 Points)

Programming Exercise 1. In this programming exercise, you will implement mini-batch gradient descent in order to train a linear model using least squares. Note that you can use an implementation of mini-batch gradient descent to get both, batch gradient descent and stochastic gradient descent.
As training data, you will use again the Energy efficiency Data Set from the UCI Machine Learning Repository, where we consider the required heating energy as output quantity.
To be more specific, you are supposed to reproduce Example 5.2 from the lecture notes, however with your own implementation of mini-batch gradient descent.
Reference solutions will only be provided in Python+Matplotlib. The submission format for Python is a Jupyter notebook. The submission format for C/C++ is standard source files. Choose an appropriate format for the Gnuplot-related submission.

(4 Points)