Linear Regression and SGD

EE599 Deep Learning

Kuan-Wen (James) Huang

Spring 2020

Linear Regression and SGD

- Setting:
  - Input: $\boldsymbol{x} \in R^D$ (features, observations, inputs)
  - Output: $\hat{y} \in R$ (targets, responses, outputs)
  - Training Data: $\left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$
  - Model: $\hat{y} = f(\boldsymbol{x}) = b + \boldsymbol{w}^T \boldsymbol{x}$

    We call $\boldsymbol{w} = [w_1, w_2, \cdots, w_D]$ weights or parameter vector and b is the bias. Sometimes, we use the augmented weights $\widetilde{\boldsymbol{w}} = [w_1, w_2, \cdots, w_D, b]^T$ with the augmented features $\widetilde{\boldsymbol{x}} = [x_1, x_2, \cdots, x_D, 1]^T$ so that $\hat{y} = \widetilde{\boldsymbol{w}}^T \widetilde{\boldsymbol{x}}$.

Linear Regression and SGD

- Objective
  - We want to minimized the sum of squared error, i.e.
$$e(\boldsymbol{w}) = \sum_{i=1}^{N} \left(y^{(i)} - \widetilde{\boldsymbol{w}}^{T} \widetilde{\boldsymbol{x}}^{(i)} \right)^2 .$$
  - It can be also written as
$$e(\boldsymbol{w}) = \left\| \boldsymbol{y} - \tilde{X} \widetilde{\boldsymbol{w}} \right\|^2$$
    where
$$\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}, \tilde{X} = \begin{bmatrix} \widetilde{\boldsymbol{x}}^{(1)T} & 1 \\ \vdots & \vdots \\ \widetilde{\boldsymbol{x}}^{(N)T} & 1 \end{bmatrix}$$

Linear Regression and SGD

- The optimal solution which minimizes $e(\boldsymbol{w})$ is
$$\boldsymbol{w}^* = \left(\tilde{X}^T\tilde{X}\right)^{-\mathbf{1}}\tilde{X}^T\boldsymbol{y}$$
if $\tilde{X}^T\tilde{X}$ is invertible.

- In general, not knowing whether the objective is convex quadratic, we solve it via stochastic gradient descent.

Linear Regression and SGD

- Let’s do some coding!