day03

January 23, 2025

0.1 HW0

use [q:q+1] to get the (1, F) shape rather than [q] to get the (F,) shape

Be sure to vectorize the numpy object , don't write nested for loops > Great performance improvements

Try to start with for loops then move to vectorize

0.2 Linear Regression

Recall: MSE (mean square error)

Will be smooth everywhere (differentiable), unlike MAE (mean absolute error)

Goal: Minimize squared error

Objective function (J) - something we want to minimize

min_(w) : means want to minimize objective w.r.t w (find the lowest w)

J'(w) = 0 (find the critical point)

Can use rise over run

Need to minimize the w by taking the derivative of the objective function w.r.t to w, and solving for w when J' is 0

When training for 1-dim no bias Linear regression, need 1 xi to be non-zero (or else zero in denominator)

Theta = [w1 .. wf b] ~xn = [xn1 .. xnF 1]

Theta^T * ~xn is just dot product (inner product)

In "always holds", we have (F+1, F+1) * (F+1, 1) = (F+1, N) * (N, 1) (dimensionality agrees)

if inverse exists: we have that Theta has at least F + 1 feature vectors that are linearly independent

Otherwise, many w, b will yield lowest possible training error