Beyond the basics of gradient descent

Poll: yes or no

yes: I want A4 to be shorter but due 10/30 → A3 size

no weekend

no: I want longer, due 11/6

A2 size = 2 · A3 size

A3: due tonight
grades by Wed.

Projects, some day next week to discuss

Stochastic gradient descent (SGD)

Goals: Step sizes, stopping criteria

Loss as an average

SGD algorithm

Batching

Really useful for NNs, large datasets

WARNING NEW NOTATION

$$\min_{\vec{w} \in \mathbb{R}^d} C(\vec{w})$$

$\vec{w}$ = model parameters

think $\vec{\beta}$

(weights)

w/ or w/o intercept

Beware: optimization uses $\min_{x} f(x)$

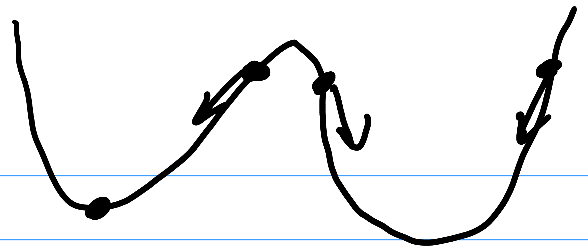
GD iteration

$$\vec{w}_{t+1} = \vec{w}_t - h \cdot \nabla C(\vec{w}_t)$$

$$= \vec{w}_t + h_t \cdot \vec{d}_t$$

h_t : step size at time t

$\vec{d}_t$: direction at time t, for GD $\vec{d}_t = -\nabla C(\vec{w}_t)$



How to pick step size:

too big
- shoot off into distance, exploding
- overcorrection overshoot
- oscillations

} unstable doesn't converge

too small
- takes forever "hangs"
- will converge

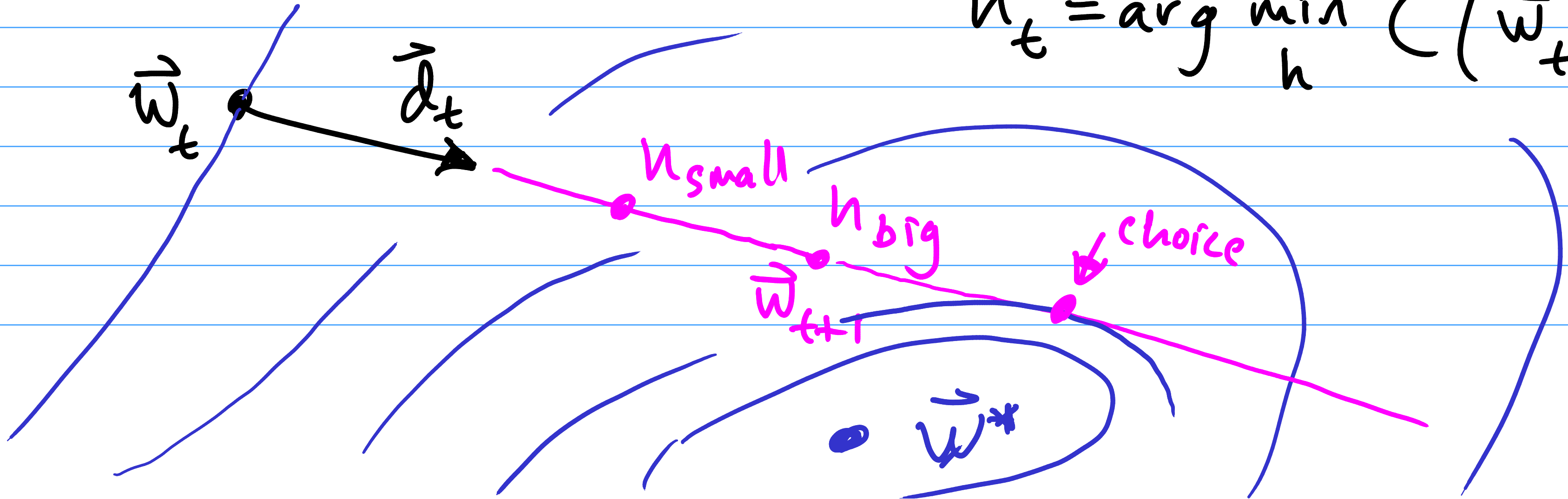
Options for picking step size

- $\eta_t = \eta$ pick constant and see what happens
  goal: to acheive convergence to a "good" minimum

  Hard to pick a priori, depends on $(X, \vec{y})$ the data, loss function, regularization

- exact line search

$$\eta_t = \arg\min_{\eta} C(\vec{w}_t - \eta \vec{d}_t)$$



- backtracking line search (Armijo rule)
  Alg 9.2
  BV book
  - start w/ $\eta = 1$
  - check if $C(\vec{w}_{t+1})$ is small enough
  - if not, shrink η

- pick $\eta_t = \begin{cases} \eta_1 & \text{for } t < 1000 \\ \eta_1/10 & \text{for } t > 1000 \end{cases}$

  common in NN's "schedule"

How do you check for convergence?

- relative decrease of cost

$$\frac{|C(\vec{w}_{t+1}) - C(\vec{w}_t)|}{|C(\vec{w}_t)|} < \text{tol} \approx 10^{-6}$$

or

10^{-8}

really $\frac{1}{\sqrt{n}}$

ML form of loss function

$$C(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{l_i(\vec{w})}_{\text{loss of } i^{\text{th}} \text{ data pt.}}$$

avg. loss

ex/ least squares $\quad l_i(\vec{w}) = L_{LS}(\hat{y}_i, y_i)$

$$= (\vec{x}_i^T\vec{w} - y_i)^2$$

$$\nabla C = \frac{1}{n}\sum_{i=1}^{n} 2\vec{x}_i(\vec{x}_i^T\vec{w} - y_i) = \frac{2}{n}(X^TX - X^T\vec{y})$$

What happens if $n = 10^8$?

$O(n)$ evaluate cost and gradient

$$C = \frac{1}{n}\sum_{i=1}^{n} l_i(\vec{w})$$

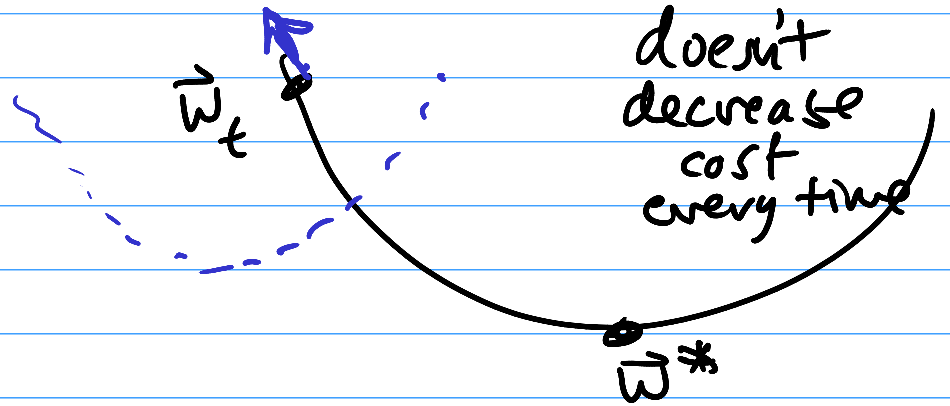
SGD

I_t = uniform random index $\{1, \dots, n\}$ (pick)

$$\vec{d}_t = -\nabla l_{I_t}(\vec{w}_t)$$

$$\vec{w}_{t+1} = \vec{w}_t + h_t \vec{d}_t$$

(skip line search)



$$\mathbb{E}[\vec{d}_t] = \sum_{i=1}^{n} \underbrace{P[I_t = i]}_{1/n} (-\nabla l_i)$$

$$= \frac{1}{n}\sum_{i=1}^{n} (-\nabla l_i) = -\nabla C(\vec{w}_t)$$

$\Rightarrow \vec{d}_t$ unbiased estimator of (-gradient)