Thursday, Oct. 10

Our goal: Let $c(w; x, y)$ be our objective for one sample

$$\min_{w \in \mathbb{R}^d} \underbrace{E[c(w; X, Y)]}_{= \int p(x,y)\, c(w; x, y)\, dx\, dy}$$

be our objective for one sample.

Empirical estimate of true obj.: $C(w) = \frac{1}{n} \sum C_i(w)$

$$C_i(w) = c(w; x_i, y_i)$$

Batch grad. descent

$$\nabla C(w) = \nabla \left(\frac{1}{n} \sum C_i(w) \right)$$
$$= \frac{1}{n} \sum \nabla C_i(w)$$

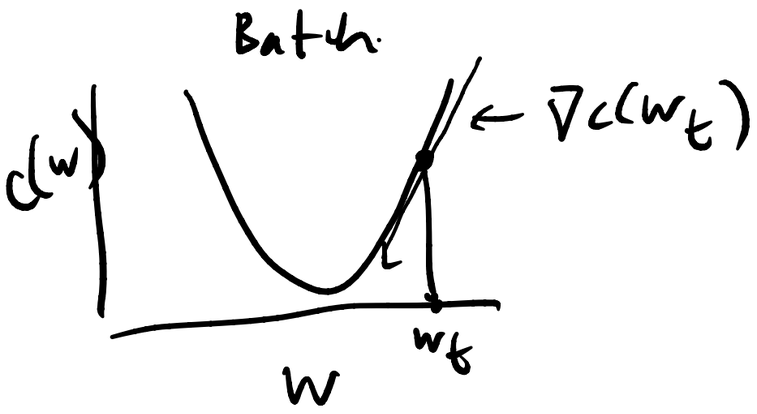
$O(dn)$ per iteration

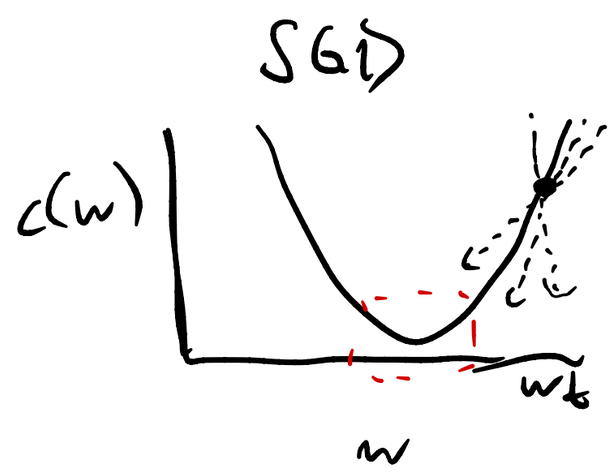
Stochastic Grad Descent

$$\nabla C(w) \overset{?}{\approx} \nabla C_i(w)$$

for random $i \in \{1, \ldots, n\}$.

$O(d)$ per iteration





$\nabla c_i(w)$

$\nabla c_j(w)$

j is randomly sample from $1, \cdots, n$

$$E[\nabla c_j(w)] = \sum_{i=1}^{n} p(i)\, \nabla c_i(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla c_i(w) = \nabla c(w)$$

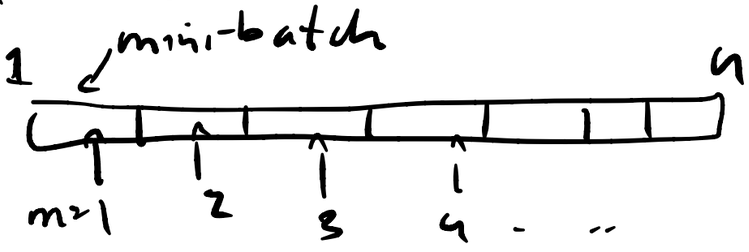
Inbetween: mini-batch SGD $\quad \nabla c(w) \approx \frac{1}{b} \sum_{j=1}^{b} \nabla c_{i_j}(w)$

Cost per iteration: $O(db)$ $\quad i_j$ uniform random in $\{1, \cdots, n\}$.

Extremes: $b = n$ or $b = 1$ (very noisy)

Usually better to be more moderate, (e.g. $b = 16$)

Implementation:



Epoch = one processing of the data

Mini-batch SGD:

for k = 1 : # of epochs (e.g 100 epochs)

→ Shuffle data $\{c_1, \ldots, c_n\}$

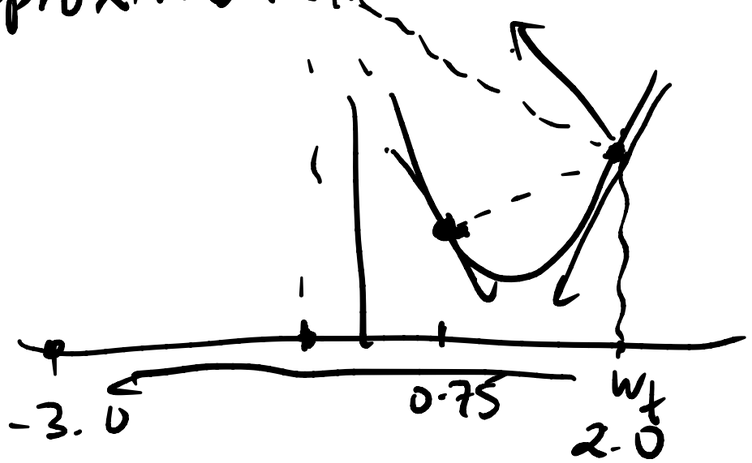
for $m = 1 : n/b$ b is mini-batch size.

$w = w - \eta$ (gradient-on-mini-batch)

Batch-grad. = First-order grad. descent

Goal for η : Find $\eta = \underset{\eta \geq 0}{\operatorname{argmin}}\ C(w_t - \eta \nabla C(w_t))$

Approximate: line search, start at $\eta = 1.0$, $0 < \tau < 1$



reduce by τ if overshoot, $\eta \in \tau\eta$

$w_t = 2.0$, $g = 5.0$, $\eta = 1.0$, $\tau = 0.5$

$\tilde{w} = w_t - 1.0 \cdot g = -3.0$ ∴ ↑C(w)

$\hat{w} = w_t - 0.5\, g = -0.5$ ∴ ↑C(w)

$\tilde{w} = w_t - 0.25 g = 0.75$ ∴ ↓C(w)

Accept

$w_{t+1} = w_t - 0.25 g$.

Should we use this for SGD: $\underset{\eta \geq 0}{\operatorname{argmin}}\ C_i(w_t - \eta \nabla C_i(w))$

not our goal

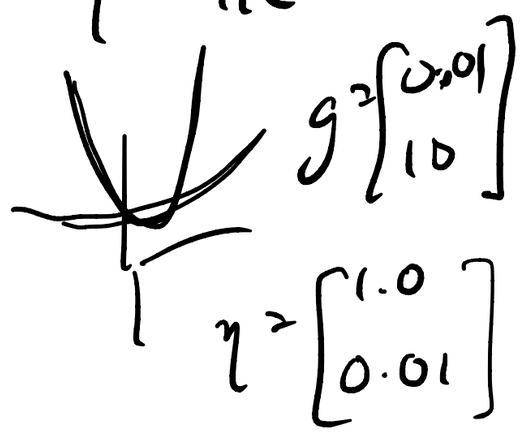
Heuristic approach for η for SGD:

e.g. Adagrad, use vector of stepsizes. $\vec{\eta} \in \mathbb{R}^d$.

$$b_t = b_t + \nabla c_i(w_t)^2 \leftarrow \text{element-wise squaring}$$

$$\eta_t = \frac{\eta_0}{\sqrt{b_t}} \leftarrow \text{element-wise}$$

$$g^2 = \begin{bmatrix} g_1^2 \\ g_2^2 \\ \vdots \\ g_d^2 \end{bmatrix}$$



statistics on magnitudes of gradients

What is the batch GD and SGD updates for l_2 regularized linear regression?

$$c(w) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(x_i^T w - y_i)^2 + \frac{\lambda}{2}\|w\|_2^2 = \frac{1}{n}\sum c_i(w)$$

$$c_i(w) = \frac{1}{2}(x_i^T w - y_i)^2 + \frac{\lambda}{2}\|w\|_2^2 \quad \rightarrow \quad \frac{1}{n}\sum_{i=1}^{n} \frac{\lambda}{2}\|w\|_2^2 \rightarrow \text{gradient} = \lambda w$$

Update: for random k

$$w_{t+1} = w_t - \eta_t \left[(x_k^T w - y_k) x_k + \lambda w \right]$$

Batch update for ℓ_1 i.e. $\|w\|_1 = \sum_{j=1}^{d} |w_j|$

$$\min_w \frac{1}{2n} \underbrace{\|Xw - y\|_2^2}_{\text{compute gradient}} + \lambda \|w\|_1$$

$$\frac{\partial \|w\|_1}{\partial w_j} = \frac{\partial \sum_{k=1}^{d} |w_k|}{\partial w_j} = \frac{\partial |w_j|}{\partial w_j}$$

$$= \begin{cases} +1 & \text{if } w_j > 0 \\ -1 & \text{if } w_j < 0 \\ 0 \text{ or } -1 \text{ or } +1 \end{cases}$$

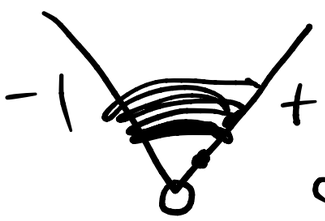
$$\frac{df(w)}{dw} = \lim_{h \to 0} \frac{f(w+h) - f(w)}{h}$$

$h = -10 \to 0$

h from left $= -1$

h from right $= +1$

$h = +10 \to 0$



< Bad, jumpy.

Nice, smooth

Limit at $w=0$: $\lim_{h \to 0} \frac{|h|}{h}$.

Alternative: descend and project

$$c(w) = \frac{1}{2n} \|Xw - y\|_2^2 \quad \text{(smooth)}$$

$$R(w) = \lambda \|w\|_1 \quad \text{(non-smooth)}$$

Descend: $\tilde{w}^{(t+1)} = w^{(t)} - \eta \nabla c(w^{(t)})$ makes error smaller

Project: (back onto space where $R(w)$ is reasonably small)

$$w^{(t+1)} = \text{prox}\left(\tilde{w}^{(t+1)}\right)$$

$$= \underset{w \in \mathbb{R}^d}{\text{argmin}} \; \underbrace{\eta\lambda \|w\|_1}_{\eta R(w)} + \frac{1}{2}\|\tilde{w}^{(t+1)} - w\|_2^2$$

big +

$$\downarrow \quad w_j^{(t+1)} = \begin{cases} \hat{w}_j^{(t+1)} - \eta\lambda & \text{if } \hat{w}_j^{(t+1)} > \eta\lambda \\ \hat{w}_j^{(t+1)} + \eta\lambda & \text{if } \hat{w}_j^{(t+1)} < -\eta\lambda \\ 0 & \text{else} \end{cases}$$

big −