Mini-Batch Gradient Descent (MBGD) Review

Example 6.18: Let $n = 8, b = 2$ then $M = 8/2 = 4$, and the dataset can be visualized as

n=9, b=2 M=floor(9/2)=4

$$\mathcal{D} = \left(\underbrace{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)}_{\text{mini-batch 1}}, \underbrace{(\mathbf{x}_3, y_3), (\mathbf{x}_4, y_4)}_{\text{mini-batch 2}}, \underbrace{(\mathbf{x}_5, y_5), (\mathbf{x}_6, y_6)}_{\text{mini-batch 3}}, \underbrace{(\mathbf{x}_7, y_7), (\mathbf{x}_8, y_8)}_{\text{mini-batch 4}} \right).$$

, $(\vec{x}_9, y_9)$

For each mini-batch $m \in \{1, \ldots, M\}$ we have a sample mean estimate of the expected loss

$$\hat{L}_m(\mathbf{w}) = \frac{1}{b} \sum_{i=(m-1)b+1}^{mb} (\mathbf{x}_i^\top \mathbf{w} - y_i)^2.$$

$$\nabla \hat{L}_m(w) = \frac{2}{b} \sum_{i=(m-1)b+1}^{mb} (\vec{x}_i^{\,T} \vec{w} - y_i) \vec{x}_i$$

During an epoch t, the MBGD learner updates the weight vector $\mathbf{w}$ for each mini-batch $m \in \{1, \ldots, M\}$ using the update rule

$$\mathbf{w}^{(t,m+1)} = \mathbf{w}^{(t,m)} - \eta^{(t)} \nabla \hat{L}_m(\mathbf{w}^{(t,m)}).$$

Once the final mini-batch M is reached, the learner updates the weight vector $\mathbf{w}$ for the next epoch $t + 1$ using the update rule

$$\mathbf{w}^{(t+1,1)} = \mathbf{w}^{(t,M)} - \eta^{(t)} \nabla \hat{L}_m(\mathbf{w}^{(t,M)}).$$

The MBGD learner $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{F}$ is defined in algorithm 3.

$$\mathcal{A}(\mathcal{D}) = \hat{f} \quad \text{where} \quad \hat{f}(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}^{(T,M)}.$$

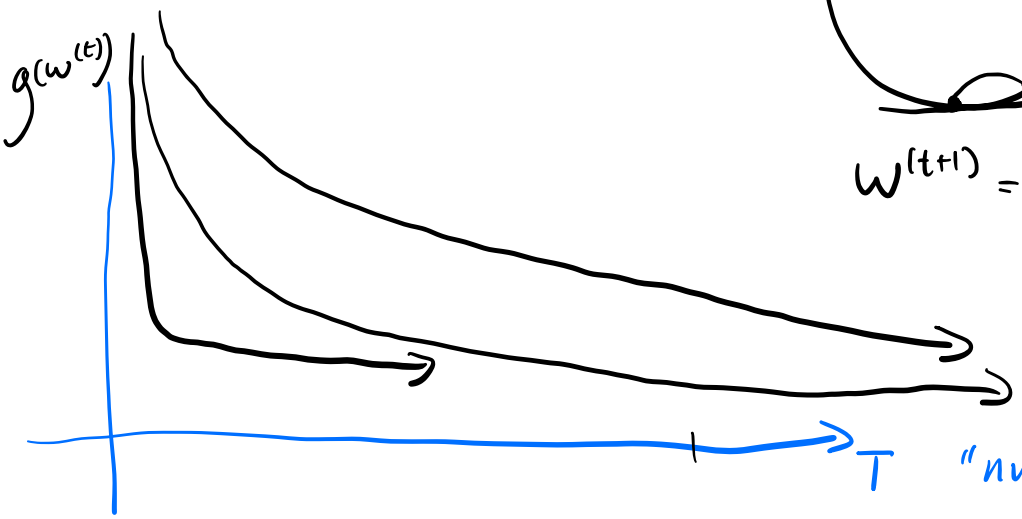
Algorithm 3: MBGD Linear Regression Learner (with a constant step size)

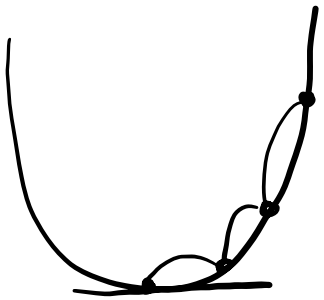
input:
$\mathcal{D} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))$, step size η, number of epochs T, mini-batch size b
$\mathbf{w} \leftarrow$ random vector in $\mathbb{R}^{d+1}$
$M \leftarrow \text{floor}(\frac{n}{b})$
for $t = 1, \ldots, T$ **do**
  randomly shuffle $\mathcal{D}$
  for $m = 1, \ldots, M$ **do**
    $\nabla \hat{L}_m(\mathbf{w}) \leftarrow \frac{2}{b} \sum_{i=(m-1)b+1}^{mb} (\mathbf{x}_i^\top \mathbf{w} - y_i)\mathbf{x}_i$
    $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla \hat{L}_m(\mathbf{w})$
return $\hat{f}(\mathbf{x}) = \mathbf{x}^T \hat{\mathbf{w}}$

TM gradient steps

Exercise 6.1: You are minimizing a function $g : \mathbb{R}^d \to \mathbb{R}$ by using gradient descent with an initial point $\mathbf{w}^{(0)}$ and step size $\eta^{(t)} = \eta$ for all $t \in \mathbb{N}$. You run gradient descent for $T = 10^6$ iterations and get the parameter $\mathbf{w}^{(T)}$. Can you be certain that $\mathbf{w}^{(T)} = \mathbf{w}^* = \text{argmin}_{\mathbf{w} \in \mathbb{R}^d} g(\mathbf{w})$? □

No. But $g(w^{(T)}) \approx g(w^*)$





$w^{(t+1)} = w^{(t)} - \eta^{(t)} \cdot 0$

Exercise 6.2: Suppose you are using an exponential decaying step size. Is there some way to set the parameters η and λ so that you are using a constant step size? □

$\eta^{(t)} = \eta \exp(-\lambda t)$ exp decay

set $\lambda = 0$

$\exp(-0 \cdot t) = 1$

$\eta^{(t)} = \eta$ const

Exercise 6.3: What would the BGD update rule be if we used the loss function $\ell(\hat{y}, y) = (\hat{y} - y)^4$? In the 1-dimensional case, with no bias term (i.e. $w \in \mathbb{R}$), is $\hat{L}(w)$ convex? □

based on ℓ

$$\vec{w}^{(t+1)} = \vec{w}^{(t)} - \eta^{(t)} \nabla \hat{L}(\vec{w}^{(t)})$$

$$\hat{L}(\vec{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\vec{x}_i^T \vec{w}, y_i) = \frac{1}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i)^4$$

$$\nabla \hat{L}(\vec{w}) = \left(\frac{\partial \hat{L}}{\partial w_0}(\vec{w}), \dots, \frac{\partial \hat{L}}{\partial w_d}(\vec{w}) \right)^T \in \mathbb{R}^{d+1}$$

$$\frac{\partial \hat{L}}{\partial w_j}(\vec{w}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial \left((\vec{x}_i^T \vec{w} - y_i)^4 \right)}{\partial w_j}(\vec{w})$$

$$= \frac{1}{n} \sum_{i=1}^{n} 4 (\vec{x}_i^T \vec{w} - y_i)^3 x_{ij}$$

$$\nabla \hat{L}(\vec{w}) = \frac{4}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i)^3 \underbrace{\left(x_{i0}, \dots, x_{id} \right)^T}_{\vec{x}_i}$$

$\in \mathbb{R}^{d+1}$

$$= \frac{4}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i)^3 \vec{x}_i \quad \in \mathbb{R}^{d+1}$$

$$\vec{w}^{(t+1)} = \vec{w}^{(t)} - \eta^{(t)} \frac{4}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i)^3 \vec{x}_i$$

1-D case $\frac{d\hat{L}}{dw}(w) = \frac{4}{n}\sum_{i=1}^{n}(x_i w - y_i)^3 x_i$

$$\frac{d^2\hat{L}}{dw^2}(w) = \frac{4}{n}\sum_{i=1}^{n}\frac{d\left((x_i w - y_i)^3 x_i\right)}{dw}(w)$$

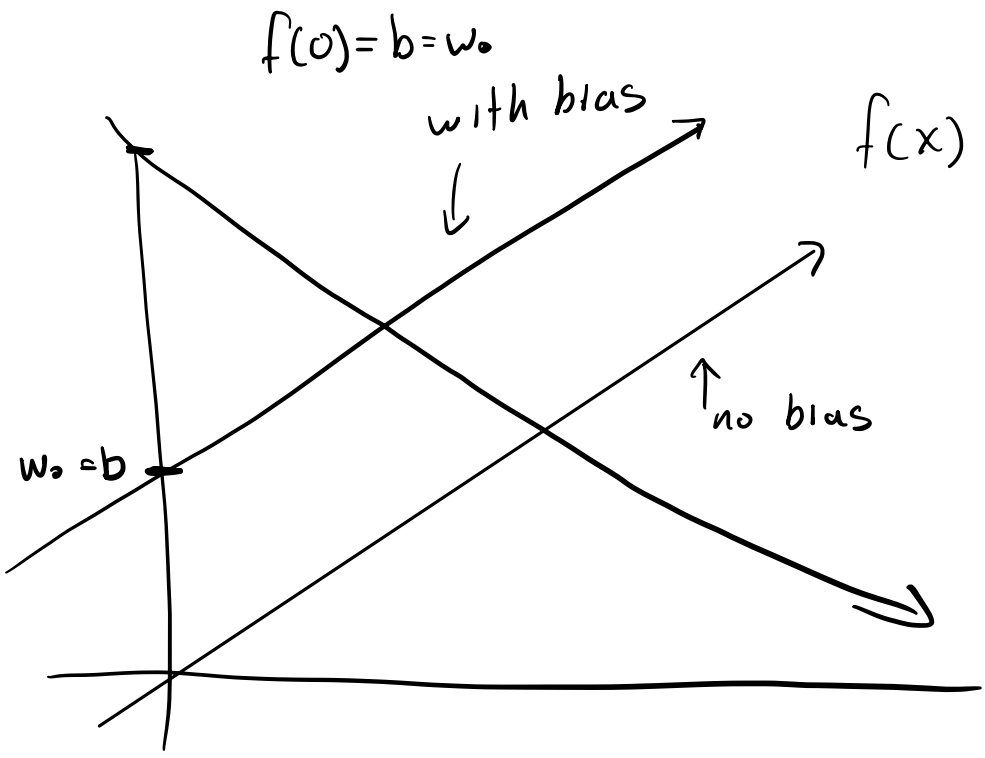
$\hat{L}$ is convex

$$= \frac{4\cdot 3}{n}\sum_{i=1}^{n}(x_i w - y_i)^2 x_i x_i$$

≥ 0 (from ≥ 0)

$$= \frac{12}{n}\sum_{i=1}^{n}(x_i w - y_i)^2 x_i^2$$

≥ 0 (≥ 0) ≥ 0



$f(x) = mx + b$

m: w_1; b: bias w_0

Exercise 6.4: Write the pseudocode for BGD with an exponential decaying step size. □

Algorithm 5: BGD Linear Regression Learner (with an exponential decaying step size)

input: $\mathcal{D} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))$, step size parameters η_0, λ, number of epochs T
$\mathbf{w} \leftarrow$ random vector in $\mathbb{R}^{d+1}$
~~**for** $t = 1, \ldots, T$ **do**~~ for $t = 0, \ldots, T-1$
  $\nabla \hat{L}(\mathbf{w}) \leftarrow \frac{2}{n} \sum_{i=1}^{n} (\mathbf{x}_i^\top \mathbf{w} - y_i)\mathbf{x}_i$
  $\eta \leftarrow \eta_0 \exp(-\lambda t)$ ← new stuff
  $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla \hat{L}(\mathbf{w})$
return $\hat{f}(\mathbf{x}) = \mathbf{x}^T \hat{\mathbf{w}}^{(T)}$

$\eta^{(t)} = \eta \exp(-\lambda t)$

step_size

Exercise 6.5: Suppose that we did not discard the last $n - Mb$ data points from the dataset, and instead used them in the last mini-batch $M+1$. What would the sample mean estimate $\hat{L}_{M+1}(\mathbf{w})$ be? □

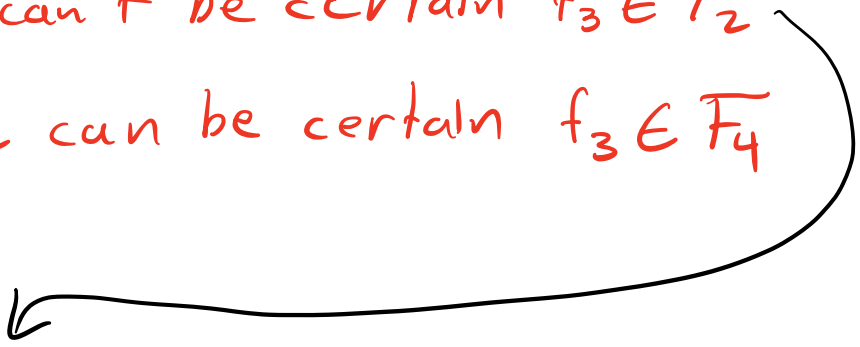
$$\hat{L}_m(\mathbf{w}) = \frac{1}{b} \sum_{i=(m-1)b+1}^{mb} (\mathbf{x}_i^\top \mathbf{w} - y_i)^2. \quad \text{if } m \in \{1, \ldots, M\}$$

$$\hat{L}_{M+1}(\vec{w}) = \frac{1}{n-Mb} \sum_{i=Mb+1}^{n} (\vec{x}_i^T \vec{w} - y_i)^2 \quad \text{if } m = M+1 \text{ and } \frac{n}{b} \text{ not int}$$

Exercise 6.6: If $f_3 \in \mathcal{F}_3$, can we be certain that $f_3 \in \mathcal{F}_2$ or that $f_3 \in \mathcal{F}_4$? □

No we can't be certain $f_3 \in \mathcal{F}_2$

Yes we can be certain $f_3 \in \mathcal{F}_4$



suppose $f_3(x) = x^3 \cdot 1$ $x^3 \notin \mathcal{F}_2$

Exercise 6.7: Let $d = 3$, and $p = 2$. What is $\bar{p}$? Write the feature map $\phi_p(\mathbf{x})$. □

$$\bar{p} = \binom{d+p}{p} = \binom{5}{2} = \frac{5 \cdot 4}{2} = 10$$

$$\vec{x} = (x_0 = 1, x_1, x_2, x_3)^T$$

$$\phi_2(\vec{x}) = (1, x_1, x_2, x_3, x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2)$$

Exercise 6.8: Suppose that

$$\bar{\mathcal{F}}_p = \{f | f : \mathbb{R}^{d+1} \to \mathbb{R}, \text{ and } f(\mathbf{x}) = \exp(\phi_p(\mathbf{x})^\top \mathbf{w}), \text{ for some } \mathbf{w} \in \mathbb{R}^{\bar{p}}\}.$$

yes

Is it true that $\bar{\mathcal{F}}_1 \subset \bar{\mathcal{F}}_2$? Is it true that $\mathcal{F}_1 \subset \bar{\mathcal{F}}_1$? Is it true that $\bar{\mathcal{F}}_1 \subset \mathcal{F}_1$? Is it true that $\min_{f \in \bar{\mathcal{F}}_1} \hat{L}(f) \leq \min_{f \in \bar{\mathcal{F}}_2} \hat{L}(f)$? □

NO

$x \in \mathcal{F}_1$

$x \notin \bar{\mathcal{F}}_1$

no

$e^x \in \bar{\mathcal{F}}_1$

$e^x \notin \mathcal{F}_1$

no

if $f_2^* = \underset{f \in \bar{\mathcal{F}}_2}{\operatorname{argmin}} \hat{L}(f) \notin \bar{\mathcal{F}}_1$

$\hat{L}(f_2^*) = \min_{f \in \bar{\mathcal{F}}_2} \hat{L}(f) \leq \min_{f \in \bar{\mathcal{F}}_1} \hat{L}(f)$

Exercise 6.9: Suppose you want to find $\hat{f}_1 = \min_{f \in \mathcal{F}_1} \hat{L}(f)$, and $\hat{f}_2 = \min_{f \in \mathcal{F}_2} \hat{L}(f)$. To do this you decide to use batch gradient descent. Let $\hat{L}_p(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (\phi_p(\mathbf{x}_i)^\top \mathbf{w} - y_i)^2$. You run BGD on the function $\hat{L}_1(\mathbf{w})$ for $T = 100$ epochs, and get $\mathbf{w}_1^{(100)}$. You run BGD on the function $\hat{L}_2(\mathbf{w})$ for $T = 1000$ epochs, and get $\mathbf{w}_2^{(1000)}$. You are told that for both cases the number of iterations T that BGD is run for is likely enough for it to reach a good approximation of the minimizer.

Can you be certain that $\hat{L}_1(w_1^{(100)}) \geq \hat{L}_2(w_2^{(1000)})$? Do you think it is likely that $\hat{L}_1(w_1^{(100)}) \geq \hat{L}_2(w_2^{(1000)})$? □

↑ yes

↑ NO

$f_1 \in \mathcal{F}_1 \quad f_2 \in \mathcal{F}_2$

$f_1(\vec{x}) = \phi_1(\vec{x})^T \vec{w}_1 = \vec{x}^T \vec{w}_1 \Rightarrow \vec{w}_1 \in \mathbb{R}^{d+1}$

$f_2(\vec{x}) = \phi_2(\vec{x})^T \vec{w}_2 \Rightarrow \vec{w}_2 \in \mathbb{R}^{\bar{p}}$

$\bar{p} = \binom{d+2}{2}$

Exercise 6.10: Write the pseudocode for the degree 3 polynomial feature map $\phi_3(\mathbf{x})$. □

Algorithm 6: Degree 3 Polynomial Feature Map

input: feature vector $\mathbf{x} = (x_0 = 1, x_1, \ldots, x_d)^\top \in \mathbb{R}^{d+1}$
$\bar{p} \leftarrow (d+1)(d+2)(d+3)/6$ ⟵ new
$\varphi \leftarrow (\varphi_0 = 0, \varphi_1 = 0, \ldots, \varphi_{\bar{p}-1} = 0)^\top \in \mathbb{R}^{\bar{p}}$
$j \leftarrow 0$
for $k = 0, \ldots, d$ **do**
  for $l = k, \ldots, d$ **do**
    for $q = l, \ldots, d$ **do** ⟵ new part
      $\varphi_j = x_k \cdot x_l \cdot x_q$ ⟵
      $j = j + 1$
return φ