Gradient Descent

$\min f(x)$

$x \in \mathbb{R}^N$

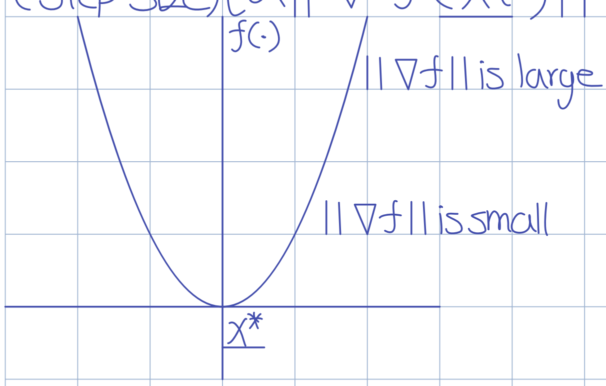
$f(x)$ is a convex function

Gradient Update

$$x_{t+1} = x_t - (\text{step size})_t \, u_t$$

$$u_t = \frac{\nabla f(x_t)}{\|\nabla f(x_t)\|}$$

$$(\text{Step Size})_t \propto \|\nabla f(x_t)\|$$



$$(\text{Step Size})_t = \epsilon_t \|\nabla f(x_t)\|$$

Learning Rate

Gradient Update

$$x_{t+1} = x_t - \epsilon_t \nabla f(x_t)$$

Learning Rate

Gradient Descent Algorithm

Initialize $\underline{x_0}$ in some arbitrary fashion

$t = 0, 1, 2, \ldots$

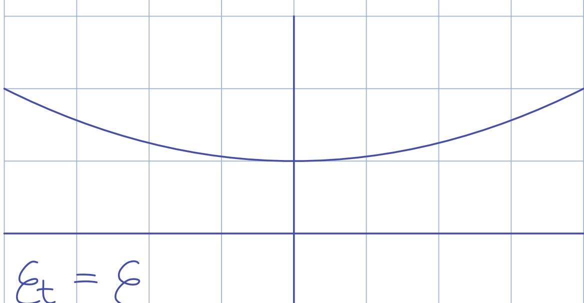
Compute $\underline{g_t} = \nabla f(\underline{x_t})$

Select Direction $\underline{u_t} = -\underline{g_t}$

Update $\underline{x_{t+1}} = \underline{x_t} + \varepsilon_t u_t$

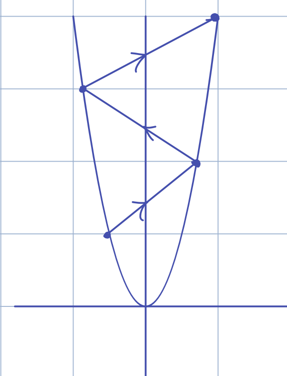
Continue until stopping condition is reached for convec function

$\|\nabla f(\underline{x_t})\| \approx 0$



$\varepsilon_t = \varepsilon$

(Very Small)



$\varepsilon_t = \varepsilon$ (Very Large)

$\varepsilon_t \approx \frac{1}{t}$ (Proportional to 1/iteration index)

Linear Regression

Loss Function

$$E_n(\underline{w}) = \frac{1}{N}\sum_{n=1}^{N} e_n(\underline{w})$$

$$e_n(\underline{w}) = (w^T x_n - y_n)^2$$

$$\underline{w_{LS}^*} = \underset{\underline{w} \in \mathbb{R}^{d+1}}{\arg\min} E_{in}(\underline{w})$$

$$= (\mathcal{X}^T \mathcal{X})^{-1} \mathcal{X}^T \underline{y}$$

Might Not Be Invertible

Use Gradient Descent To Minimize $E_{in}(\underline{w})$

Update Rule

$$\underline{w}_{t+1} = \underline{w}_t - \varepsilon_t \nabla E_{in}(\underline{w}_t)$$

$$\nabla E_{in}(\underline{w}_t) = \nabla\left(\frac{1}{N}\sum_{n=1}^{N} e_n(\underline{w})\right)$$

$$= \frac{1}{N}\sum_{n=1}^{N} \nabla e_n(\underline{w})$$

$$\nabla_{\underline{w}} e_n(\underline{w}) = \nabla_{\underline{w}} (\underline{w^T x_n} - y_n)^2$$

$$= 2(\underline{w^T x_n} - y_n)\nabla_w(\underline{w^T x_n} - y_n)$$

$$= 2(\underline{w^T x_n} - y_n)\underline{x_n}$$

$$\boxed{\underline{w}_{t+1} = \underline{w}_t - \varepsilon_t \left[\frac{2}{N}\sum_{n=1}^{N} (\underline{w^T x_n} - y_n)\underline{x_n}\right] \quad \text{Iterative Procedure}}$$

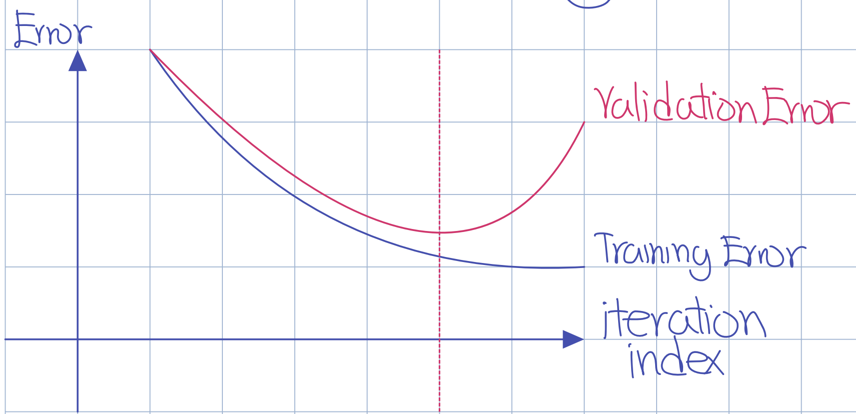
Batch Gradient Descent

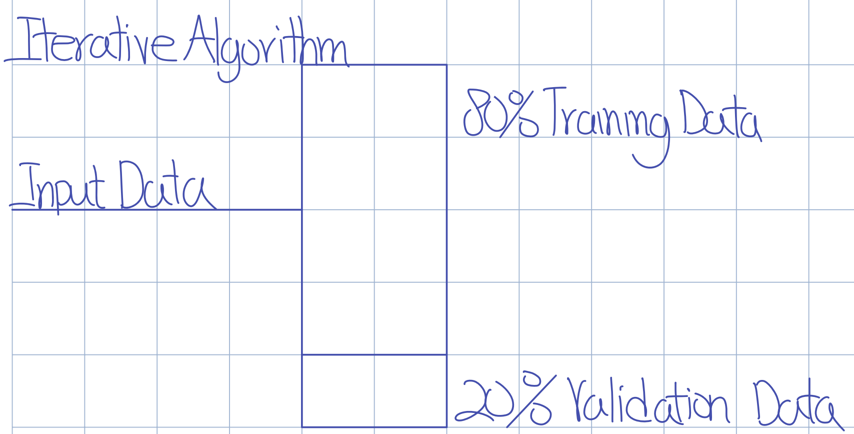
$O(Nd)$

$\underline{w_t}$ will converge to $\underline{w_{LS}^*}$

Two Concerns

1) $(\mathcal{X}^T\mathcal{X})$ could be ill conditioned
2) $(\mathcal{X}^T\mathcal{X})^{-1}$ is computationally expensive





Batch Gradient Descent

Complexity per update $O(Nd)$

Stochastic Gradient Descent

$$w_{t+1} = w_t - \varepsilon_t \nabla e_n(w_t)$$

$n \in \text{Uniform}\{1, 2, \dots, N\}$

Random Variable

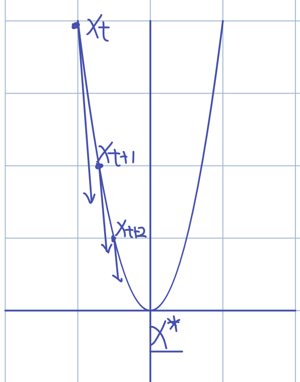
Justification of SGD

$$\mathbb{E}_n\left[\nabla e_n(\underline{w_t})\right] = \sum_{n=1}^{N} \Pr(n=1)\, \nabla e_n(\underline{w_t})$$

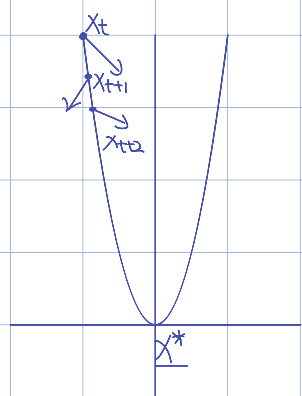
Expectation

$$= \nabla E_{in}(\underline{w_t})$$

Matches True Gradient In Expectation



Batch Gradient Descent



SGD

Mini Batch Gradient Descent

at each iteration draw M examples from the dataset at random (without replacement)

$S = \{j_1, j_2, \ldots, j_M\}$

Update Rule

$$\underline{w_{t+1}} = \underline{w_t} - \frac{\epsilon_t}{M} \sum_{n=1}^{M} \nabla e_{(j_n)}(\underline{w_t})$$

$M=N$, Batch Gradient Update

$M=1$, SGD

GD For Logistic Regression

$$E_{in}(\underline{w}) = \frac{1}{N}\sum_{n=1}^{N} e_n(\underline{w})$$

$$e_n(\underline{w}) = \log(1+e^{-y_n \underline{w}^T x_n})$$

SGD Update

$$\underline{W_{k+1}} = \underline{W_k} - \varepsilon_k \nabla e_n(\underline{W_k})$$

$$n \in \text{Uniform}\{1, 2, \ldots, N\}$$

$$\nabla_{\underline{W_k}} e_n(\underline{W_k}) = \nabla_{W_k}\left[\log(1+e^{-y_n \underline{W_k^T x_n}})\right]$$

By Chain Rule and Calculus

$$= \frac{1}{1+e^{-y_n \underline{W_k^T x_n}}} \nabla_{W_k}(1+e^{-y_n \underline{W_k^T x_n}})$$

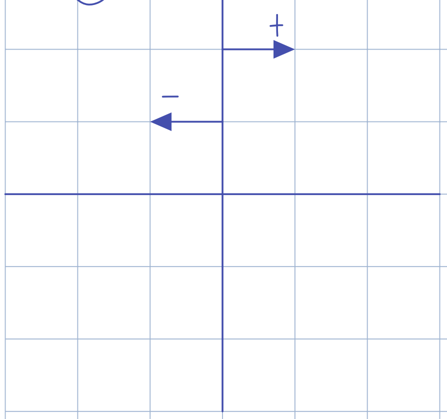
$$= \frac{e^{-y_n \underline{W_k^T x_n}}}{1+e^{-y_n \underline{W_k^T x_n}}} \nabla_{W_k}(-y_n \underline{W_k^T x_n})$$

$$= \frac{1}{1+e^{y_n \underline{W_k^T x_n}}}(-y_n \underline{x_n})$$

Update Rule

$$\underline{W_{k+1}} = \underline{W_k} + \varepsilon_k \left\{ \frac{y_n x_n}{1+e^{y_n \underline{W_k^T x_n}}} \right\}$$

"Highly Misclassified Point"



$y_n \underline{w_k}^T \underline{x_n} << 0$ (Large And Negative)

$\underline{w_{k+1}} = \underline{w_k} + \epsilon_k y_n \underline{x_n}$

Well Classified Point

$y_n \underline{w_k}^T \underline{x_n} >> 0$ (Large And Possitive)

$\underline{w_{k+1}} \approx \underline{w_k}$

Multiple Classes Logistic Regression

$y \in \{1, 2, \dots, c\}, \ c \geq 2$

Input Data

$D = \{(\underline{x_1}, y_1), \dots, (\underline{x_N}, y_N)\}$

$\underline{x_i} \in \mathbb{R}^{d+1}, \ y_i \in \{1, 2, \dots, c\}$

We will consider the case: $c = 3$

Let $\underline{w}(1), \underline{w}(2)$ and $\underline{w}(3)$ be the weight be the weight vectors for class 1, 2 and 3

$\underline{\Omega} = \{\underline{w}(1), \underline{w}(2), \underline{w}(3)\}$

Output:

$[\hat{P}_\Omega(1 \mid \underline{x}), \ \hat{P}_\Omega(2 \mid \underline{x}), \ \hat{P}_\Omega(3 \mid \underline{x})]$

$$\hat{P}_\Omega(i \mid x) = \frac{e^{w^T(i)x}}{e^{w^T(1)x} + e^{w^T(2)x} + e^{w^T(3)x}}$$

$i \in \{1, 2, 3\}$

Loss Function

Log Loss Function

Given (x_n, y_n)

$y_n \in \{1, 2, 3\}$

$$e_n(\Omega) = -\log \hat{P}_\Omega(y_n \mid x_n)$$

$$\Omega^* = \underset{\Omega = \{w(1), w(2), w(3)\}}{\mathrm{argmin}}\, E_n(\Omega) = \{w(1)^*, w(2)^*, w(3)^*\}$$

$C = 2$ (Binary Classification)

$\Omega = \{w(1), w(2)\}$

$$\hat{P}_\Omega(1 \mid x) = \frac{e^{w(1)^T x}}{e^{w(1)^T x} + e^{w(2)^T x}} = \frac{e^{(w(1) - w(2))^T x}}{1 + e^{(w(1) - w(2))^T x}}$$

$$\hat{P}_\Omega(2 \mid x) = \frac{e^{w(2)^T x}}{e^{w(1)^T x} + e^{w(2)^T x}} = \frac{1}{1 + e^{(w(1) - w(2))^T x}}$$

Previous Logistic Regression Model

$\Omega = \{w\}$

$$\hat{p}_{\Omega}(1 \mid x) = \frac{e^{w^T x}}{1 + e^{w^T x}}$$

$$\hat{p}_{\Omega}(2 \mid x) = \frac{1}{1 + e^{w^T x}}$$

Thus if $w(1) - w(2) = w$

Then the models will output same probabilities

SGD Update Rule for

Multi Class Logistic Regression

Iteration #k

Model Parameter

$$\Omega_k = \{ w_k(1), w_k(2), w_k(3) \}$$

$$w_{k+1}(1) = w_k(1) - \varepsilon_k \nabla_{w_k(1)} e_n(\Omega_k)$$

$$w_{k+1}(2) = w_k(2) - \varepsilon_k \nabla_{w_k(2)} e_n(\Omega_k)$$

$$w_{k+1}(3) = w_k(3) - \varepsilon_k \nabla_{w_k(3)} e_n(\Omega_k)$$

$$\begin{bmatrix} w_{k+1}(1) \\ w_{k+1}(2) \\ w_{k+1}(3) \\ \Omega_{k+1} \end{bmatrix} = \begin{bmatrix} w_k(1) \\ w_k(2) \\ w_k(3) \\ \Omega_k \end{bmatrix} - \varepsilon_k \nabla_{\Omega_k} e_n(\Omega_k)$$

Compute

$$\underline{\nabla_{w_k(l)}} \left[-\log \hat{P}_\Omega(y_n|x_n) \right] \quad l \in \{1,2,3\}$$

$$= \underline{\nabla_{w_k(l)}} \left[-\log \frac{e^{\underline{w_k^T(y_n)x_n}}}{e^{\underline{w^T(1)x}} + e^{\underline{w^T(2)x}} + e^{\underline{w^T(3)x}}} \right]$$

$$= \underline{\nabla_{w_k(l)}} \left[-\underline{w_k^T(y_n)x_n} + \log\left(e^{\underline{w^T(1)x}} + e^{\underline{w^T(2)x}} + e^{\underline{w^T(3)x}}\right) \right]$$

If $l = y_n$

$$= -x_n + \underline{\nabla_{w_k(y_n)}} \left[\log\left(\sum_{i=1}^{3} e^{\underline{w_k^T(i)x_n}} \right) \right]$$

$$= -x_n + \frac{1}{\sum_{i=1}^{3} e^{\underline{w_k^T(i)x_n}}} e^{\underline{w_k^T(l)x}} \underline{x_n}$$

If $c \neq y_n$

$$= \frac{1}{\sum_{i=1}^{3} e^{w_k^T(i) x_n}} e^{w_k^T(c) x} x_n$$