First TRIPODS Summer School
Stochastic Gradient Descent, AdaGrad, & Co.

Francesco Orabona, Boston University

Content:

- Stochastic Gradient Descent
- Adaptive Stepsizes
- AdaGrad

Sources:

- See papers in the text

Disclaimer

- We will only see the easy proofs, that means strong assumptions will be used
- Stronger results do exist, but the proofs are long and boring

Unconstrained Convex Optimization

Unconstrained convex optimization problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad f(\boldsymbol{w})$$

Unconstrained Convex Optimization

Unconstrained convex optimization problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad f(\boldsymbol{w})$$

Optimality condition for differentiable convex objectives

$\boldsymbol{w}$ is a global minimizer if and only if $\nabla f(\boldsymbol{w}) = \mathbf{0}$

Unconstrained convex optimization problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \quad f(\boldsymbol{w})$$

Optimality condition for differentiable convex objectives

$\boldsymbol{w}$ is a global minimizer if and only if $\nabla f(\boldsymbol{w}) = \mathbf{0}$

Unfortunately, can't always find closed-form solution to system of equations $\nabla f(\boldsymbol{w}) = \mathbf{0}$ $\Rightarrow$ Resort to iterative methods to find a solution.

Gradient descent for differentiable objectives

- Start with some initial $\boldsymbol{w}_1 \in \mathbb{R}^d$
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied
  - Compute gradient of f at $\boldsymbol{w}_t$:
  $$\boldsymbol{g}_t := \nabla f(\boldsymbol{w}_t)$$
  - Update:
  $$\boldsymbol{w}_{t+1} := \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t$$
- Output: $\boldsymbol{w}_T$

Subgradient Descent

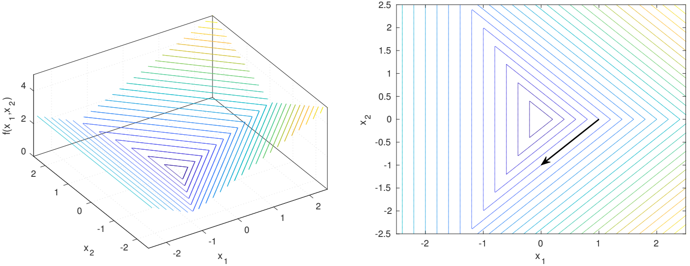
Subgradient Descent

- Start with some initial $\boldsymbol{w}_1 \in \mathbb{R}^d$
- For $t = 1, 2, \ldots, T$
  - Get subgradient $\boldsymbol{g}_t$ of f at $\boldsymbol{w}_t$, i.e. $\boldsymbol{g}_t \in \partial f(\boldsymbol{w}_t)$
  - Update:

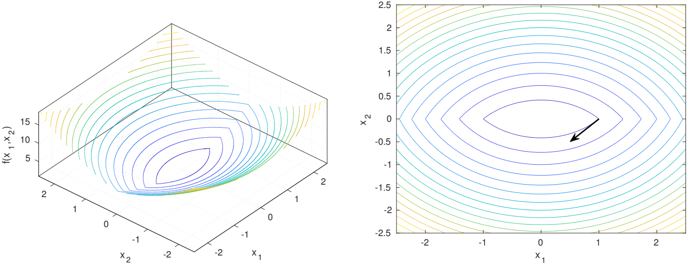
$$\boldsymbol{w}_{t+1} := \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t$$

- Output: $\boldsymbol{w}_T$ or $\bar{\boldsymbol{w}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{w}_t$

Subgradient Descent is not a Descent Method

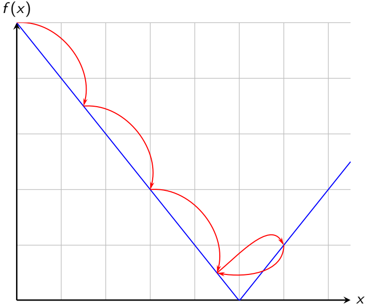


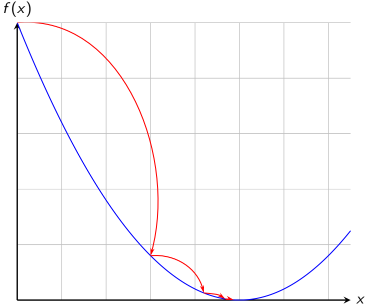
3D plot (left) and level sets (right) of $f(\boldsymbol{x}) = \max[-x_1, x_1 - x_2, x_1 + x_2]$



3D plot (left) and level sets (right) of $f(\boldsymbol{x}) = \max[x_1^2 + (x_2+1)^2, x_1^2 + (x_2-1)^2]$

Decreasing Stepsizes/Learning Rates



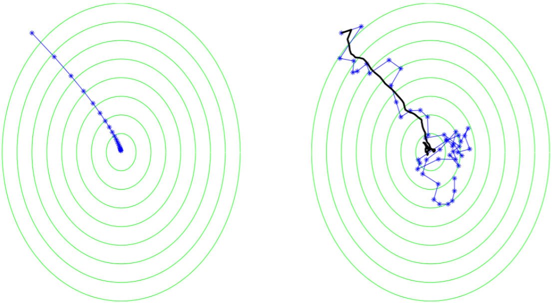


The effect of a constant stepsize on non-differentiable (left) and smooth (right) functions

Stochastic Subgradient Descent

- Start with some initial $\boldsymbol{w}_1 \in \mathbb{R}^d$
- For $t = 1, 2, \ldots, T$
  - Get stochastic subgradient $\boldsymbol{g}_t$ of f at $\boldsymbol{w}_t$ such that
    $$\mathbb{E}[\boldsymbol{g}_t] \in \partial f(\boldsymbol{w}_t)$$
  - Update:
    $$\boldsymbol{w}_{t+1} := \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t$$
- Output: $\boldsymbol{w}_T$ or $\bar{\boldsymbol{w}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{w}_t$

SGD is More "Unstable"



Analyzing SGD for Convex Functions (1)

For any sequence of $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T$, any $\eta_1, \cdots, \eta_T > 0$, and any $\boldsymbol{w}^\star$, we have

$$\eta_t \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle = \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2}{2} + \frac{\eta^2}{2}\|\boldsymbol{g}_t\|^2$$

Analyzing SGD for Convex Functions (1)

For any sequence of $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T$, any $\eta_1, \cdots, \eta_T > 0$, and any $\boldsymbol{w}^\star$, we have

$$\eta_t \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle = \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2}{2} + \frac{\eta_t^2}{2} \|\boldsymbol{g}_t\|^2$$

Considering a constant stepsize η, diving by η, and summing we have

$$\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle = \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{T+1} - \boldsymbol{w}^\star\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2$$

Analyzing SGD for Convex Functions (1)

For any sequence of $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T$, any $\eta_1, \cdots, \eta_T > 0$, and any $\boldsymbol{w}^\star$, we have

$$\eta_t \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle = \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2}{2} + \frac{\eta_t^2}{2}\|\boldsymbol{g}_t\|^2$$

Considering a constant stepsize η, diving by η, and summing we have

$$\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle = \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{T+1} - \boldsymbol{w}^\star\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2$$

Assume that $\|\boldsymbol{g}_t\| \leq L$ for all t, $\boldsymbol{w}_1 = \boldsymbol{0}$, and that $\|\boldsymbol{w}^\star\| \leq B$ we obtain

$$\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq \frac{B^2}{2\eta} + \frac{\eta L^2 T}{2}$$

Analyzing SGD for Convex Functions (1)

For any sequence of $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T$, any $\eta_1, \cdots, \eta_T > 0$, and any $\boldsymbol{w}^\star$, we have

$$\eta_t \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle = \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2}{2} + \frac{\eta_t^2}{2}\|\boldsymbol{g}_t\|^2$$

Considering a constant stepsize η, diving by η, and summing we have

$$\sum_{t=1}^T \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle = \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{T+1} - \boldsymbol{w}^\star\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|^2$$

Assume that $\|\boldsymbol{g}_t\| \leq L$ for all t, $\boldsymbol{w}_1 = \boldsymbol{0}$, and that $\|\boldsymbol{w}^\star\| \leq B$ we obtain

$$\sum_{t=1}^T \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq \frac{B^2}{2\eta} + \frac{\eta L^2 T}{2}$$

In particular, for $\eta = \sqrt{\frac{B^2}{L^2 T}}$ we get

$$\sum_{t=1}^T \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq BL\sqrt{T}$$

Analyzing SGD for Convex Functions (2)

Taking expectation of both sides w.r.t. the randomness of choosing $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T$ we obtain:

$$\mathbb{E}_{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T}\left[\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle\right] \leq BL\sqrt{T}$$

Analyzing SGD for Convex Functions (2)

Taking expectation of both sides w.r.t. the randomness of choosing $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T$ we obtain:

$$\mathbb{E}_{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T}\left[\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t\rangle\right] \leq BL\sqrt{T}$$

The law of total expectation: for every two random variables α, β, and a function h, $\mathbb{E}_\alpha[h(\alpha)] = \mathbb{E}_\beta E_\alpha[h(\alpha)|\beta]$. Therefore

$$\mathbb{E}_{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T}[\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t\rangle] = \mathbb{E}_{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_{t-1}}\mathbb{E}_{\boldsymbol{g}_t, \cdots, \boldsymbol{g}_T}[\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t\rangle | \boldsymbol{g}_1, \cdots, \boldsymbol{g}_{t-1}]$$

Analyzing SGD for Convex Functions (2)

Taking expectation of both sides w.r.t. the randomness of choosing $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T$ we obtain:

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t\rangle\right] \leq BL\sqrt{T}$$

The law of total expectation: for every two random variables α, β, and a function h, $\mathbb{E}_\alpha[h(\alpha)] = \mathbb{E}_\beta E_\alpha[h(\alpha)|\beta]$. Therefore

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}[\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t\rangle] = \mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_{t-1}}\mathbb{E}_{\boldsymbol{g}_t,\cdots,\boldsymbol{g}_T}[\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t\rangle|\boldsymbol{g}_1,\cdots,\boldsymbol{g}_{t-1}]$$

Once we know $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_{t-1}$ the value of $\boldsymbol{w}_t$ is not random, hence,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{g}_t,\cdots,\boldsymbol{g}_T}[\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t\rangle|\boldsymbol{g}_1,\cdots,\boldsymbol{g}_{t-1}] &= \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \mathbb{E}_{\boldsymbol{g}_t,\cdots,\boldsymbol{g}_T}[\boldsymbol{g}_t]\rangle \\ &= \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \mathbb{E}_{\boldsymbol{g}_t}[\boldsymbol{g}_t]\rangle \\ &= \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \nabla f(\boldsymbol{w}_t)\rangle\end{aligned}$$

We got:

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \nabla f(\boldsymbol{w}_t)]\rangle\right] \leq BL\sqrt{T}$$

Analyzing SGD for Convex Functions (3)

We got:

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \nabla f(\boldsymbol{w}_t)]\rangle\right] \leq BL\sqrt{T}$$

By the defintion of subgradient, this means

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[\sum_{t=1}^{T}(f(\boldsymbol{w}_t) - f(\boldsymbol{w}^\star))\right] \leq BL\sqrt{T}$$

Analyzing SGD for Convex Functions (3)

We got:

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \nabla f(\boldsymbol{w}_t)]\rangle\right] \leq BL\sqrt{T}$$

By the defintion of subgradient, this means

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[\sum_{t=1}^{T}(f(\boldsymbol{w}_t) - f(\boldsymbol{w}^\star))\right] \leq BL\sqrt{T}$$

Dividing by T and using Jensen's inequality,

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}_t\right)\right] \leq f(\boldsymbol{w}^\star) + \frac{BL}{\sqrt{T}}$$

Analyzing SGD for Convex Functions (3)

We got:

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \nabla f(\boldsymbol{w}_t)]\rangle\right] \leq BL\sqrt{T}$$

By the defintion of subgradient, this means

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[\sum_{t=1}^{T}(f(\boldsymbol{w}_t) - f(\boldsymbol{w}^\star))\right] \leq BL\sqrt{T}$$

Dividing by T and using Jensen's inequality,

$$\mathbb{E}_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_T}\left[f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}_t\right)\right] \leq f(\boldsymbol{w}^\star) + \frac{BL}{\sqrt{T}}$$

Very slow convergence, but very low per-step complexity!

- $$\min_{\boldsymbol{w} \in \mathcal{F}} L_S(w) \text{ where } L_S(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$$

Stochastic Gradient Descent for Machine Learning

- $$\min_{\boldsymbol{w} \in \mathcal{F}} \; L_S(w) \text{ where } L_S(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$$
- Gradient descent on the empirical risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_S(\boldsymbol{w}_t)$

- $$\min_{\boldsymbol{w}\in\mathcal{F}} L_S(w) \text{ where } L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$$
- Gradient descent on the empirical risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_S(\boldsymbol{w}_t)$
- Observe: $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$

- $$\min_{\boldsymbol{w} \in \mathcal{F}} \; L_S(w) \text{ where } L_S(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$$
- Gradient descent on the empirical risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_S(\boldsymbol{w}_t)$
- Observe: $\nabla L_S(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- It takes $O(m)$ time to calculate

- $$\min_{\boldsymbol{w} \in \mathcal{F}} L_S(w) \text{ where } L_S(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$$
- Gradient descent on the empirical risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_S(\boldsymbol{w}_t)$
- Observe: $\nabla L_S(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- It takes $O(m)$ time to calculate
- But we can estimate it by $\nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_j), y_j)$ for j a random variable uniform in $[1, m]$

- $$\min_{\boldsymbol{w} \in \mathcal{F}} L_S(w) \text{ where } L_S(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$$
- Gradient descent on the empirical risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_S(\boldsymbol{w}_t)$
- Observe: $\nabla L_S(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- It takes $O(m)$ time to calculate
- But we can estimate it by $\nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_j), y_j)$ for j a random variable uniform in $[1, m]$
- In other words, $\boldsymbol{g}_t$ is an unbiased estimate of the gradient

- $$\min_{\boldsymbol{w}\in\mathcal{F}} L_S(w) \text{ where } L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \ell(p(\boldsymbol{w},\boldsymbol{x}_i), y_i)$$
- Gradient descent on the empirical risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_S(\boldsymbol{w}_t)$
- Observe: $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w},\boldsymbol{x}_i), y_i)$
- It takes $O(m)$ time to calculate
- But we can estimate it by $\nabla \ell(p(\boldsymbol{w},\boldsymbol{x}_j), y_j)$ for j a random variable uniform in $[1, m]$
- In other words, $\boldsymbol{g}_t$ is an unbiased estimate of the gradient
- We just showed that this is good enough!

Mini-Batches

- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$

Mini-Batches

- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- We can use 1 or more samples to estimate the gradient: mini-batch

Mini-Batches

- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- We can use 1 or more samples to estimate the gradient: mini-batch
- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i) \approx \frac{1}{n}\sum_{i \in \text{mini-batch}} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$

Mini-Batches

- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- We can use 1 or more samples to estimate the gradient: mini-batch
- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i) \approx \frac{1}{n}\sum_{i \in \text{mini-batch}} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- In expectation, nothing changes for any choice of n, but...

Mini-Batches

- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- We can use 1 or more samples to estimate the gradient: mini-batch
- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i) \approx \frac{1}{n}\sum_{i \in \text{mini-batch}} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- In expectation, nothing changes for any choice of n, but...
- ...the variance decreases with n

Mini-Batches

- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- We can use 1 or more samples to estimate the gradient: mini-batch
- $\nabla L_S(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i) \approx \frac{1}{n}\sum_{i \in \text{mini-batch}} \nabla \ell(p(\boldsymbol{w}, \boldsymbol{x}_i), y_i)$
- In expectation, nothing changes for any choice of n, but...
- ...the variance decreases with n
- ...the time to calculate the approximation grows with n

SGD Guarantee for Convex ERM Problems

Corollary

Consider a convex ERM problem, L-Lipschitz and with the domain of diameter B. Then, if we run the SGD method for minimizing $L_S(\boldsymbol{w})$ with T iterations and with $\eta = \sqrt{\frac{B^2}{L^2 T}}$, then the output of SGD satisfies:

$$\mathbb{E}[L_S(\bar{\boldsymbol{w}})] \leq \min_{\boldsymbol{w} \in \mathcal{F}} L_S(\boldsymbol{w}) + \frac{BL}{\sqrt{T}}$$

In words, we minimize the empirical risk (but, we still have to hope that this will give us small true risk...)

Learning *Directly* with Stochastic Gradient Descent

- Consider a learning problem
- Recall: our goal is to (probably approximately) solve:

$$\min_{\boldsymbol{w} \in \mathcal{F}} L_{\mathcal{D}}(\boldsymbol{w}) \text{ where } L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$$

- So far, learning was based on the empirical risk, $L_S(\boldsymbol{w})$

Learning *Directly* with Stochastic Gradient Descent

- Consider a learning problem
- Recall: our goal is to (probably approximately) solve:

$$\min_{\boldsymbol{w} \in \mathcal{F}} L_{\mathcal{D}}(\boldsymbol{w}) \text{ where } L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$$

- So far, learning was based on the empirical risk, $L_S(\boldsymbol{w})$
- Can we minimize directly $L_{\mathcal{D}}(\boldsymbol{w})$?

Stochastic Gradient Descent for Learning

$$\min_{\boldsymbol{w} \in \mathcal{F}} L_{\mathcal{D}}(w) \text{ where } L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$$

- Gradient descent on the true risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$

Stochastic Gradient Descent for Learning

$$\min_{\boldsymbol{w} \in \mathcal{F}} L_{\mathcal{D}}(w) \text{ where } L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$$

- Gradient descent on the true risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$
- Observe: $\nabla L_{\mathcal{D}}(\boldsymbol{w}_t) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\nabla \ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$

Stochastic Gradient Descent for Learning

$$\min_{\boldsymbol{w} \in \mathcal{F}} L_{\mathcal{D}}(w) \text{ where } L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$$

- Gradient descent on the true risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$
- Observe: $\nabla L_{\mathcal{D}}(\boldsymbol{w}_t) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\nabla \ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$
- We cannot calculate $\nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$ because we do not know $\mathcal{D}$

Stochastic Gradient Descent for Learning

$$\min_{\boldsymbol{w} \in \mathcal{F}} L_{\mathcal{D}}(w) \text{ where } L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$$

- Gradient descent on the true risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$
- Observe: $\nabla L_{\mathcal{D}}(\boldsymbol{w}_t) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\nabla \ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$
- We cannot calculate $\nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$ because we do not know $\mathcal{D}$
- But we can estimate it by $\nabla \ell(p(\boldsymbol{w},\boldsymbol{x}),y)$ for $(\boldsymbol{x},y) \sim \mathcal{D}$

Stochastic Gradient Descent for Learning

$$\min_{\boldsymbol{w} \in \mathcal{F}} L_{\mathcal{D}}(w) \text{ where } L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$$

- Gradient descent on the true risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$
- Observe: $\nabla L_{\mathcal{D}}(\boldsymbol{w}_t) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\nabla \ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$
- We cannot calculate $\nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$ because we do not know $\mathcal{D}$
- But we can estimate it by $\nabla \ell(p(\boldsymbol{w},\boldsymbol{x}),y)$ for $(\boldsymbol{x},y) \sim \mathcal{D}$
- Again, we can use a mini-batch

Stochastic Gradient Descent for Learning

$$\min_{\boldsymbol{w} \in \mathcal{F}} L_{\mathcal{D}}(w) \text{ where } L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$$

- Gradient descent on the true risk: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$
- Observe: $\nabla L_{\mathcal{D}}(\boldsymbol{w}_t) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\nabla \ell(p(\boldsymbol{w},\boldsymbol{x}),y)]$
- We cannot calculate $\nabla L_{\mathcal{D}}(\boldsymbol{w}_t)$ because we do not know $\mathcal{D}$
- But we can estimate it by $\nabla \ell(p(\boldsymbol{w},\boldsymbol{x}),y)$ for $(\boldsymbol{x},y) \sim \mathcal{D}$
- Again, we can use a mini-batch
- Once again, we already showed that this is good enough!

Corollary

Consider a convex learning problem, L-Lipschitz and with the domain of diameter B. Then, if we run the SGD method to minimize $L_{\mathcal{D}}(\boldsymbol{w})$ with T iterations (i.e. number of examples) and with $\eta = \sqrt{\frac{B^2}{L^2 T}}$, then the output of SGD satisfies:

$$\mathbb{E}[L_{\mathcal{D}}(\bar{\boldsymbol{w}})] \leq \min_{\boldsymbol{w} \in \mathcal{F}} L_{\mathcal{D}}(\boldsymbol{w}) + \frac{BL}{\sqrt{T}}$$

In words, we minimize the true risk!

Is Good Precision Important for ML?

- In many applications of (convex) optimization, we care about solving problems to very high precision

Is Good Precision Important for ML?

- In many applications of (convex) optimization, we care about solving problems to very high precision
- For machine learning applications: optimization problem based on training data often just a means-to-an-end

Is Good Precision Important for ML?

- In many applications of (convex) optimization, we care about solving problems to very high precision

- For machine learning applications: optimization problem based on training data often just a means-to-an-end

  We only care about the true risk

Is Good Precision Important for ML?

- In many applications of (convex) optimization, we care about solving problems to very high precision
- For machine learning applications: optimization problem based on training data often just a means-to-an-end

  We only care about the true risk
- Running gradient descent to convergence not strictly necessary
  On the contrary, it may be beneficial to stop early (e.g., when validation error starts to increase significantly) [Yao et al. 2005]

Is Good Precision Important for ML?

- In many applications of (convex) optimization, we care about solving problems to very high precision
- For machine learning applications: optimization problem based on training data often just a means-to-an-end

  We only care about the true risk
- Running gradient descent to convergence not strictly necessary
  On the contrary, it may be beneficial to stop early (e.g., when validation error starts to increase significantly) [Yao et al. 2005]
- More precisely, it is questionable to have a precision bigger than $O(\frac{1}{\sqrt{T}})$ because the excess risk of your predictor is $O(\frac{1}{\sqrt{T}})$

Is Good Precision Important for ML?

- In many applications of (convex) optimization, we care about solving problems to very high precision
- For machine learning applications: optimization problem based on training data often just a means-to-an-end

  We only care about the true risk
- Running gradient descent to convergence not strictly necessary
  On the contrary, it may be beneficial to stop early (e.g., when validation error starts to increase significantly) [Yao et al. 2005]
- More precisely, it is questionable to have a precision bigger than $O(\frac{1}{\sqrt{T}})$ because the excess risk of your predictor is $O(\frac{1}{\sqrt{T}})$
- In this view, SGD is computationally optimal [Bottou&Bousquet, NIPS'08]

- Most of the time, you can avoid to take the average, the last solution is good enough

Practical Considerations for Convex Problems with Enough Data

- Most of the time, you can avoid to take the average, the last solution is good enough
- Most of the time, one pass over the data will give you close to optimal performance, if you tune the learning rate

Practical Considerations for Convex Problems with Enough Data

- Most of the time, you can avoid to take the average, the last solution is good enough
- Most of the time, one pass over the data will give you close to optimal performance, if you tune the learning rate
- Most of the time, few iterations over the training data will give you optimal performance, if you tune the learning rate

Practical Considerations for Convex Problems with Enough Data

- Most of the time, you can avoid to take the average, the last solution is good enough
- Most of the time, one pass over the data will give you close to optimal performance, if you tune the learning rate
- Most of the time, few iterations over the training data will give you optimal performance, if you tune the learning rate
- Learning rates of $\frac{a}{\sqrt{t}}$ are good for most of the convex problems

Practical Considerations for Convex Problems with Enough Data

- Most of the time, you can avoid to take the average, the last solution is good enough
- Most of the time, one pass over the data will give you close to optimal performance, if you tune the learning rate
- Most of the time, few iterations over the training data will give you optimal performance, if you tune the learning rate
- Learning rates of $\frac{a}{\sqrt{t}}$ are good for most of the convex problems
- In some cases, you can go with step-wise constant learning rates, or $\frac{a}{t}$ rates

Practical Considerations for Convex Problems with Enough Data

- Most of the time, you can avoid to take the average, the last solution is good enough
- Most of the time, one pass over the data will give you close to optimal performance, if you tune the learning rate
- Most of the time, few iterations over the training data will give you optimal performance, if you tune the learning rate
- Learning rates of $\frac{a}{\sqrt{t}}$ are good for most of the convex problems
- In some cases, you can go with step-wise constant learning rates, or $\frac{a}{t}$ rates
- Adaptive learning rates do work

Practical Considerations for Convex Problems with Enough Data

- Most of the time, you can avoid to take the average, the last solution is good enough
- Most of the time, one pass over the data will give you close to optimal performance, if you tune the learning rate
- Most of the time, few iterations over the training data will give you optimal performance, if you tune the learning rate
- Learning rates of $\frac{a}{\sqrt{t}}$ are good for most of the convex problems
- In some cases, you can go with step-wise constant learning rates, or $\frac{a}{t}$ rates
- Adaptive learning rates do work
- If you do not want learning rates at all, google "parameter-free online optimization"

Example: Soft-Margin SVM (without Offset)

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad f(\boldsymbol{w}) := \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 + \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S} [1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle]_+$$

f is the sum of convex functions, and hence is convex

Example: Soft-Margin SVM (without Offset)

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad f(\boldsymbol{w}) := \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 + \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S} [1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle]_+$$

f is the sum of convex functions, and hence is convex

Question: How do we compute *a* subgradient $\boldsymbol{g}$ of f at a given point $\boldsymbol{w} \in \mathbb{R}^d$?

Example: Soft-Margin SVM (without Offset)

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad f(\boldsymbol{w}) := \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 + \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S} [1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle]_+$$

f is the sum of convex functions, and hence is convex

Question: How do we compute *a* subgradient $\boldsymbol{g}$ of f at a given point $\boldsymbol{w} \in \mathbb{R}^d$?

$$\partial f(\boldsymbol{w}) \ni \boldsymbol{g} = \lambda \boldsymbol{w} + \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S} \begin{cases} \boldsymbol{0} & \text{if } 1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle < 0; \\ -y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle \geq 0 \end{cases}$$

Example: Soft-Margin SVM (without Offset)

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \qquad f(\boldsymbol{w}) := \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{|S|}\sum_{(\boldsymbol{x},y)\in S}[1 - y\langle\boldsymbol{w},\boldsymbol{x}\rangle]_+$$

f is the sum of convex functions, and hence is convex

Question: How do we compute *a* subgradient $\boldsymbol{g}$ of f at a given point $\boldsymbol{w}\in\mathbb{R}^d$?

$$\partial f(\boldsymbol{w}) \ni \boldsymbol{g} = \lambda\boldsymbol{w} + \frac{1}{|S|}\sum_{(\boldsymbol{x},y)\in S}\begin{cases}\boldsymbol{0} & \text{if } 1 - y\langle\boldsymbol{w},\boldsymbol{x}\rangle < 0;\\ -y\boldsymbol{x} & \text{if } 1 - y\langle\boldsymbol{w},\boldsymbol{x}\rangle \geq 0\end{cases}$$

Also fine:

$$\partial f(\boldsymbol{w}) \ni \boldsymbol{g} = \lambda\boldsymbol{w} + \frac{1}{|S|}\sum_{(\boldsymbol{x},y)\in S}\begin{cases}\boldsymbol{0} & \text{if } 1 - y\langle\boldsymbol{w},\boldsymbol{x}\rangle < 0;\\ -\frac{1}{3}y\boldsymbol{x} & \text{if } 1 - y\langle\boldsymbol{w},\boldsymbol{x}\rangle = 0;\\ -y\boldsymbol{x} & \text{if } 1 - y\langle\boldsymbol{w},\boldsymbol{x}\rangle > 0\end{cases}$$

Example: Soft-Margin SVM (without Offset)

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad f(\boldsymbol{w}) := \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 + \frac{1}{|S|} \sum_{(\boldsymbol{x}, y) \in S} [1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle]_+$$

f is the sum of convex functions, and hence is convex

Question: How do we compute *a* subgradient $\boldsymbol{g}$ of f at a given point $\boldsymbol{w} \in \mathbb{R}^d$?

$$\partial f(\boldsymbol{w}) \ni \boldsymbol{g} = \lambda \boldsymbol{w} + \frac{1}{|S|} \sum_{(\boldsymbol{x}, y) \in S} \begin{cases} \mathbf{0} & \text{if } 1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle < 0; \\ -y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle \geq 0 \end{cases}$$

Also fine:

$$\partial f(\boldsymbol{w}) \ni \boldsymbol{g} = \lambda \boldsymbol{w} + \frac{1}{|S|} \sum_{(\boldsymbol{x}, y) \in S} \begin{cases} \mathbf{0} & \text{if } 1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle < 0; \\ -\frac{1}{3} y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle = 0; \\ -y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle > 0 \end{cases}$$

In practice, usually don't have examples with $y\langle \boldsymbol{w}, \boldsymbol{x} \rangle = 1$ *exactly* anyway

Subgradient descent algorithm for soft-margin SVM:

- Start with some initial $\boldsymbol{w}_1 \in \mathbb{R}^d$.
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied

$$\boldsymbol{w}_{t+1} := \boldsymbol{w}_t - \eta_t \left(\lambda \boldsymbol{w}_t + \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S} \begin{cases} \boldsymbol{0} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle < 0; \\ -y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \geq 0 \end{cases} \right)$$

$$= (1 - \lambda\eta_t)\boldsymbol{w}_t + \eta_t \frac{1}{|S|} \sum_{\substack{(\boldsymbol{x},y) \in S: \\ y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \leq 1}} y\boldsymbol{x}$$

Example: Soft-Margin SVM (without Offset)

Subgradient descent algorithm for soft-margin SVM:

- Start with some initial $\boldsymbol{w}_1 \in \mathbb{R}^d$.
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied

$$\begin{aligned}
\boldsymbol{w}_{t+1} &:= \boldsymbol{w}_t - \eta_t \left(\lambda \boldsymbol{w}_t + \frac{1}{|S|} \sum_{(\boldsymbol{x},y)\in S} \begin{cases} \mathbf{0} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle < 0; \\ -y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \geq 0 \end{cases} \right) \\
&= (1 - \lambda\eta_t)\boldsymbol{w}_t + \eta_t \frac{1}{|S|} \sum_{\substack{(\boldsymbol{x},y)\in S: \\ y\langle \boldsymbol{w}_t, \boldsymbol{x}\rangle \leq 1}} y\boldsymbol{x}
\end{aligned}$$

Note effect of regularization term $\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$ **(whenever** $\eta_t < 1/\lambda$**)**:

Subgradient descent algorithm for soft-margin SVM:

- Start with some initial $\boldsymbol{w}_1 \in \mathbb{R}^d$.
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied

$$\boldsymbol{w}_{t+1} := \boldsymbol{w}_t - \eta_t \left(\lambda \boldsymbol{w}_t + \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S} \begin{cases} \mathbf{0} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle < 0; \\ -y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \geq 0 \end{cases} \right)$$

$$= (1 - \lambda\eta_t)\boldsymbol{w}_t + \eta_t \frac{1}{|S|} \sum_{\substack{(\boldsymbol{x},y) \in S: \\ y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \leq 1}} y\boldsymbol{x}$$

Note effect of regularization term $\frac{\lambda}{2} \|\boldsymbol{w}\|_2^2$ **(whenever** $\eta_t < 1/\lambda$**)**:

- Shrink $\boldsymbol{w}_t$ by a factor $1 - \lambda\eta_t$ before updating with subgradient of loss

Example: Soft-Margin SVM (without Offset)

Subgradient descent algorithm for soft-margin SVM:

- Start with some initial $\boldsymbol{w}_1 \in \mathbb{R}^d$.
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied

$$
\begin{aligned}
\boldsymbol{w}_{t+1} &:= \boldsymbol{w}_t - \eta_t \left(\lambda \boldsymbol{w}_t + \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S} \begin{cases} \boldsymbol{0} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle < 0; \\ -y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \geq 0 \end{cases} \right) \\
&= (1 - \lambda\eta_t)\boldsymbol{w}_t + \eta_t \frac{1}{|S|} \sum_{\substack{(\boldsymbol{x},y) \in S: \\ y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \leq 1}} y\boldsymbol{x}
\end{aligned}
$$

Note effect of regularization term $\frac{\lambda}{2} \|\boldsymbol{w}\|_2^2$ **(whenever** $\eta_t < 1/\lambda$**)**:

- Shrink $\boldsymbol{w}_t$ by a factor $1 - \lambda\eta_t$ before updating with subgradient of loss
- It tries to prevent length of $\boldsymbol{w}_t$ from becoming too large

Example: Soft-Margin SVM (without Offset)

Subgradient descent algorithm for soft-margin SVM:

- Start with some initial $\boldsymbol{w}_1 \in \mathbb{R}^d$.
- For $t = 1, 2, \dots$ until some stopping condition is satisfied

$$\boldsymbol{w}_{t+1} := \boldsymbol{w}_t - \eta_t \left(\lambda \boldsymbol{w}_t + \frac{1}{|S|} \sum_{(\boldsymbol{x},y)\in S} \begin{cases} \boldsymbol{0} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x}\rangle < 0; \\ -y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x}\rangle \geq 0 \end{cases} \right)$$

$$= (1 - \lambda\eta_t)\boldsymbol{w}_t + \eta_t \frac{1}{|S|} \sum_{\substack{(\boldsymbol{x},y)\in S: \\ y\langle \boldsymbol{w}_t, \boldsymbol{x}\rangle \leq 1}} y\boldsymbol{x}$$

Note effect of regularization term $\frac{\lambda}{2} \|\boldsymbol{w}\|_2^2$ **(whenever** $\eta_t < 1/\lambda$**)**:

- Shrink $\boldsymbol{w}_t$ by a factor $1 - \lambda\eta_t$ before updating with subgradient of loss
- It tries to prevent length of $\boldsymbol{w}_t$ from becoming too large
- In NN literature is called **Weight decay**

Example: Soft-Margin SVM (without Offset)

Subgradient descent algorithm for soft-margin SVM:

- Start with some initial $\boldsymbol{w}_1 \in \mathbb{R}^d$.
- For $t = 1, 2, \dots$ until some stopping condition is satisfied

$$\boldsymbol{w}_{t+1} := \boldsymbol{w}_t - \eta_t \left(\lambda \boldsymbol{w}_t + \frac{1}{|S|} \sum_{(\boldsymbol{x},y)\in S} \begin{cases} \boldsymbol{0} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle < 0; \\ -y\boldsymbol{x} & \text{if } 1 - y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \geq 0 \end{cases} \right)$$

$$= (1 - \lambda\eta_t)\boldsymbol{w}_t + \eta_t \frac{1}{|S|} \sum_{\substack{(\boldsymbol{x},y)\in S: \\ y\langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \leq 1}} y\boldsymbol{x}$$

Note effect of regularization term $\frac{\lambda}{2} \|\boldsymbol{w}\|_2^2$ **(whenever** $\eta_t < 1/\lambda$**)**:

- Shrink $\boldsymbol{w}_t$ by a factor $1 - \lambda\eta_t$ before updating with subgradient of loss
- It tries to prevent length of $\boldsymbol{w}_t$ from becoming too large
- In NN literature is called **Weight decay**
- Optimal stepsize $\eta_t = \frac{1}{\lambda t}$ gives the Pegasos algorithm [Shalev-Shwartz et al. ICML'07]

Adaptive Stepsizes

Adaptive Stepsizes

Nobody wants to tune the learning rates/stepsizes!

Adaptive algorithms:

- AdaGrad [Duchi et al. COLT'10]
- AdaDelta [Zeiler. ArXiv'12]
- RMSProp [Tieleman&Hinton. Coursera slide'12]
- Adam [Kingma&Ba. ICLR'15]
- ...

Let's consider again

$$f\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t\right) - f(\boldsymbol{w}^\star) \leq \frac{\frac{B^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}{T}$$

- We proposed a stepsize of the form $\eta = \frac{B}{\sqrt{T}}$

Optimal Stepsize with Knowledge of the Future

Let's consider again

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}_t\right) - f(\boldsymbol{w}^\star) \le \frac{\frac{B^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2}{T}$$

- We proposed a stepsize of the form $\eta = \frac{B}{\sqrt{T}}$
- However, the optimal stepsize is $\eta = \frac{B}{\sqrt{\sum_{i=1}^{T}\|\boldsymbol{g}_i\|^2}}$

Optimal Stepsize with Knowledge of the Future

Let's consider again

$$f\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t\right) - f(\boldsymbol{w}^\star) \leq \frac{\frac{B^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}{T}$$

- We proposed a stepsize of the form $\eta = \frac{B}{\sqrt{T}}$
- However, the optimal stepsize is $\eta = \frac{B}{\sqrt{\sum_{i=1}^{T} \|\boldsymbol{g}_i\|^2}}$
- It would result in a convergence rate of $\frac{B\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}}{T}$

Optimal Stepsize with Knowledge of the Future

Let's consider again

$$f\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t\right) - f(\boldsymbol{w}^\star) \le \frac{\frac{B^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2}{T}$$

- We proposed a stepsize of the form $\eta = \frac{B}{\sqrt{T}}$
- However, the optimal stepsize is $\eta = \frac{B}{\sqrt{\sum_{i=1}^{T}\|\boldsymbol{g}_i\|^2}}$
- It would result in a convergence rate of $\frac{B\sqrt{\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2}}{T}$
- But, we should know the future!

Approximating the Future

- Let's approximate the optimal stepsize with something we can calculate [Auer et al. 2002]
- Instead of $\eta = \frac{B}{\sqrt{\sum_{i=1}^{T} \|\boldsymbol{g}_i\|^2}}$ let's use $\eta_t = \frac{B}{\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|^2}}$
- Let's prove that it works!

Adding Projections to the Algorithm

Projected Stochastic Subgradient Descent

- Start with some initial $\boldsymbol{w}_1 \in V$, V convex set
- For $t = 1, 2, \ldots, T$
  - Get stochastic subgradient $\boldsymbol{g}_t$ of f at $\boldsymbol{w}_t$ such that
$$\mathbb{E}[\boldsymbol{g}_t] \in \partial f(\boldsymbol{w}_t)$$
  - Update:
$$\boldsymbol{w}_{t+1} := \Pi_V(\boldsymbol{w}_t - \eta_t \boldsymbol{g}_t)$$
- Output: $\boldsymbol{w}_T$ or $\bar{\boldsymbol{w}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t$

$$\Pi_V(\boldsymbol{w}) = \arg\min_{\boldsymbol{v} \in V} \|\boldsymbol{w} - \boldsymbol{v}\|_2$$

$$\eta_t \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2}{2} + \frac{\eta^2}{2} \|\boldsymbol{g}_t\|^2$$

Denote $\Delta_t = \|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2$, divide by η_t and sum

where we assumed V to have diameter B

Zinchevich's Proof [Zinchevich, ICML'03]

$$\eta_t \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \le \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2}{2} + \frac{\eta^2}{2} \|\boldsymbol{g}_t\|^2$$

Denote $\Delta_t = \|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2$, divide by η_t and sum

$$\begin{aligned} &\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \\ &\le \frac{1}{2\eta_1}\Delta_1 - \frac{1}{2\eta_T}\Delta_{T+1} + \sum_{t=1}^{T-1} \left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t} \right) \Delta_t + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\boldsymbol{g}_t\|^2 \end{aligned}$$

where we assumed V to have diameter B

Zinchevich's Proof [Zinchevich, ICML'03]

$$\eta_t \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2}{2} + \frac{\eta^2}{2}\|\boldsymbol{g}_t\|^2$$

Denote $\Delta_t = \|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2$, divide by η_t and sum

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \\
&\leq \frac{1}{2\eta_1}\Delta_1 - \frac{1}{2\eta_T}\Delta_{T+1} + \sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right)\Delta_t + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|^2 \\
&\leq \frac{1}{2\eta_1}B^2 + B^2\sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|^2
\end{aligned}$$

where we assumed V to have diameter B

Zinchevich's Proof [Zinchevich, ICML'03]

$$\eta_t \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2}{2} + \frac{\eta^2}{2}\|\boldsymbol{g}_t\|^2$$

Denote $\Delta_t = \|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2$, divide by η_t and sum

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \\
&\leq \frac{1}{2\eta_1}\Delta_1 - \frac{1}{2\eta_T}\Delta_{T+1} + \sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right)\Delta_t + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|^2 \\
&\leq \frac{1}{2\eta_1}B^2 + B^2\sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|^2 \\
&= \frac{1}{2\eta_1}B^2 + B^2\left(\frac{1}{2\eta_T} - \frac{1}{2\eta_1}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|^2
\end{aligned}$$

where we assumed V to have diameter B

Zinchevich's Proof [Zinchevich, ICML'03]

$$\eta_t \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^\star\|^2}{2} + \frac{\eta^2}{2}\|\boldsymbol{g}_t\|^2$$

Denote $\Delta_t = \|\boldsymbol{w}_t - \boldsymbol{w}^\star\|^2$, divide by η_t and sum

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \\
&\leq \frac{1}{2\eta_1}\Delta_1 - \frac{1}{2\eta_T}\Delta_{T+1} + \sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right)\Delta_t + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|^2 \\
&\leq \frac{1}{2\eta_1}B^2 + B^2\sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|^2 \\
&= \frac{1}{2\eta_1}B^2 + B^2\left(\frac{1}{2\eta_T} - \frac{1}{2\eta_1}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|^2 \\
&= \frac{B^2}{2\eta_T} + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|^2
\end{aligned}$$

where we assumed V to have diameter B

The Approximation Works

A useful lemma:

$$\sum_{t=1}^{T} \frac{a_t}{\sqrt{\sum_{i=1}^{t} a_t}} \leq 2\sqrt{\sum_{t=1}^{T} a_t}$$

The Approximation Works

A useful lemma:

$$\sum_{t=1}^{T} \frac{a_t}{\sqrt{\sum_{i=1}^{t} a_t}} \leq 2\sqrt{\sum_{t=1}^{T} a_t}$$

Set $\eta_t = \frac{\sqrt{2}B}{2\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|^2}}$, then

$$\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq \frac{B^2}{2\eta_T} + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\boldsymbol{g}_t\|^2$$

The Approximation Works

A useful lemma:

$$\sum_{t=1}^{T} \frac{a_t}{\sqrt{\sum_{i=1}^{t} a_t}} \leq 2\sqrt{\sum_{t=1}^{T} a_t}$$

Set $\eta_t = \frac{\sqrt{2}B}{2\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|^2}}$, then

$$\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq \frac{B^2}{2\eta_T} + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\boldsymbol{g}_t\|^2 \leq \sqrt{2}B\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}$$

The Approximation Works

A useful lemma:

$$\sum_{t=1}^{T} \frac{a_t}{\sqrt{\sum_{i=1}^{t} a_t}} \leq 2\sqrt{\sum_{t=1}^{T} a_t}$$

Set $\eta_t = \frac{\sqrt{2}B}{2\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|^2}}$, then

$$\sum_{t=1}^{T} \langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t \rangle \leq \frac{B^2}{2\eta_T} + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\boldsymbol{g}_t\|^2 \leq \sqrt{2}B \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}$$

Hence

$$f\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t\right) - f(\boldsymbol{w}^\star) \leq \sqrt{2}\frac{B\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}}{T}$$

- Only $\sqrt{2}$ worse than knowing the future!
- Similar proof with stochastic subgradients

Adaptation to Subgradients Implies Adaptation to Noise (1)

Let's now assume that

- ∇f is M-Lipschitz $\Rightarrow \|\nabla f(\boldsymbol{w})\|^2 \leq 2M(f(\boldsymbol{w}) - f(\boldsymbol{w}^\star))$
- $\mathbb{E}_t[\boldsymbol{g}_t] = \nabla f(\boldsymbol{w}_t)$
- $E_t[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t)\|^2] \leq \sigma^2$

Adaptation to Subgradients Implies Adaptation to Noise (1)

Let's now assume that

- ∇f is M-Lipschitz $\Rightarrow \|\nabla f(\boldsymbol{w})\|^2 \leq 2M(f(\boldsymbol{w}) - f(\boldsymbol{w}^\star))$
- $\mathbb{E}_t[\boldsymbol{g}_t] = \nabla f(\boldsymbol{w}_t)$
- $E_t[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t)\|^2] \leq \sigma^2$

$$\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) \leq \sqrt{2}B\sqrt{\sum_{t=1}^{T} \mathbb{E}[\|\boldsymbol{g}_t\|^2]}$$

Adaptation to Subgradients Implies Adaptation to Noise (1)

Let's now assume that

- ∇f is M-Lipschitz $\Rightarrow \|\nabla f(\boldsymbol{w})\|^2 \leq 2M(f(\boldsymbol{w}) - f(\boldsymbol{w}^\star))$
- $\mathbb{E}_t[\boldsymbol{g}_t] = \nabla f(\boldsymbol{w}_t)$
- $E_t[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t)\|^2] \leq \sigma^2$

$$
\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) &\leq \sqrt{2}B\sqrt{\sum_{t=1}^{T} \mathbb{E}[\|\boldsymbol{g}_t\|^2]} \\
&= \sqrt{2}B\sqrt{\sum_{t=1}^{T} \mathbb{E}[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t) + \nabla f(\boldsymbol{w}_t)\|^2]}
\end{aligned}
$$

Adaptation to Subgradients Implies Adaptation to Noise (1)

Let's now assume that

- ∇f is M-Lipschitz $\Rightarrow \|\nabla f(\boldsymbol{w})\|^2 \leq 2M(f(\boldsymbol{w}) - f(\boldsymbol{w}^\star))$
- $\mathbb{E}_t[\boldsymbol{g}_t] = \nabla f(\boldsymbol{w}_t)$
- $E_t[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t)\|^2] \leq \sigma^2$

$$
\begin{aligned}
\sum_{t=1}^T \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) &\leq \sqrt{2}B\sqrt{\sum_{t=1}^T \mathbb{E}[\|\boldsymbol{g}_t\|^2]} \\
&= \sqrt{2}B\sqrt{\sum_{t=1}^T \mathbb{E}[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t) + \nabla f(\boldsymbol{w}_t)\|^2]} \\
&= \sqrt{2}B\sqrt{\sum_{t=1}^T \mathbb{E}[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t)\|^2 + \|\nabla f(\boldsymbol{w}_t)\|^2]}
\end{aligned}
$$

Adaptation to Subgradients Implies Adaptation to Noise (1)

Let's now assume that

- ∇f is M-Lipschitz $\Rightarrow \|\nabla f(\boldsymbol{w})\|^2 \leq 2M(f(\boldsymbol{w}) - f(\boldsymbol{w}^\star))$
- $\mathbb{E}_t[\boldsymbol{g}_t] = \nabla f(\boldsymbol{w}_t)$
- $E_t[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t)\|^2] \leq \sigma^2$

$$
\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) &\leq \sqrt{2}B\sqrt{\sum_{t=1}^{T} \mathbb{E}[\|\boldsymbol{g}_t\|^2]} \\
&= \sqrt{2}B\sqrt{\sum_{t=1}^{T} \mathbb{E}[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t) + \nabla f(\boldsymbol{w}_t)\|^2]} \\
&= \sqrt{2}B\sqrt{\sum_{t=1}^{T} \mathbb{E}[\|\boldsymbol{g}_t - \nabla f(\boldsymbol{w}_t)\|^2 + \|\nabla f(\boldsymbol{w}_t)\|^2]} \\
&\leq \sqrt{2}B\sqrt{T\sigma^2 + 2M\sum_{t=1}^{T}(\mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star))}
\end{aligned}
$$

Adaptation to Subgradients Implies Adaptation to Noise (2)

$$\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) \le \sqrt{2}B\sqrt{T\sigma^2 + 2M\sum_{t=1}^{T}(\mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star))}$$

Adaptation to Subgradients Implies Adaptation to Noise (2)

$$\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) \leq \sqrt{2}B\sqrt{T\sigma^2 + 2M\sum_{t=1}^{T}(\mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star))}$$

implies

$$\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) \leq B\sigma\sqrt{T} + 4MB^2$$

Adaptation to Subgradients Implies Adaptation to Noise (2)

$$\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) \leq \sqrt{2} B \sqrt{T\sigma^2 + 2M \sum_{t=1}^{T} (\mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star))}$$

implies

$$\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) \leq B\sigma\sqrt{T} + 4MB^2$$

that gives, by Jensen's inequality,

$$\mathbb{E}\left[f\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t\right)\right] - f(\boldsymbol{w}^\star) \leq \frac{\sqrt{2}B\sigma}{\sqrt{T}} + \frac{4MB^2}{T}$$

Adaptation to Subgradients Implies Adaptation to Noise (2)

$$\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) \leq \sqrt{2}B\sqrt{T\sigma^2 + 2M\sum_{t=1}^{T}(\mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star))}$$

implies

$$\sum_{t=1}^{T} \mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^\star) \leq B\sigma\sqrt{T} + 4MB^2$$

that gives, by Jensen's inequality,

$$\mathbb{E}\left[f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}_t\right)\right] - f(\boldsymbol{w}^\star) \leq \frac{\sqrt{2}B\sigma}{\sqrt{T}} + \frac{4MB^2}{T}$$

- If $\sigma > 0$ we will converge as $O(\frac{1}{\sqrt{T}})$
- If $\sigma = 0$ we will converge as $O(\frac{1}{T})$
- The algorithm adapts to the level of noise

Folklore, e.g. [Li&Orabona, ArXiv'18]

AdaGrad

AdaGrad [Duchi et al. COLT'10]

We can think to use the previous stepsize for each single coordinate!
Proof is easy, for each coordinate j we have $\eta_{t,j} = \frac{\sqrt{2}B}{2\sqrt{\sum_{i=1}^{t} g_{i,j}^2}}$

$$\sum_{t=1}^{T}(w_{t,j} - w_j^\star)g_{t,j} \leq \sqrt{2}B_i\sqrt{\sum_{t=1}^{T} g_{t,j}^2}$$

AdaGrad [Duchi et al. COLT'10]

We can think to use the previous stepsize for each single coordinate!
Proof is easy, for each coordinate j we have $\eta_{t,j} = \frac{\sqrt{2}B}{2\sqrt{\sum_{i=1}^{t} g_{i,j}^2}}$

$$\sum_{t=1}^{T}(w_{t,j} - w_j^\star)g_{t,j} \leq \sqrt{2}B_i \sqrt{\sum_{t=1}^{T} g_{t,j}^2}$$

Summing over the coordinate we have

$$\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t\rangle = \sum_{j=1}^{d}\sum_{t=1}^{T}(w_{t,j} - w_j^\star)g_{t,j} \leq \sqrt{2}\sum_{j=1}^{d} B_i \sqrt{\sum_{t=1}^{T} g_{t,j}^2}$$

$$\leq \sqrt{2}B_\infty \sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} g_{t,j}^2}$$

AdaGrad [Duchi et al. COLT'10]

We can think to use the previous stepsize for each single coordinate!
Proof is easy, for each coordinate j we have $\eta_{t,j} = \frac{\sqrt{2}B}{2\sqrt{\sum_{i=1}^{t} g_{i,j}^2}}$

$$\sum_{t=1}^{T}(w_{t,j} - w_j^\star)g_{t,j} \leq \sqrt{2}B_i\sqrt{\sum_{t=1}^{T} g_{t,j}^2}$$

Summing over the coordinate we have

$$\sum_{t=1}^{T}\langle \boldsymbol{w}_t - \boldsymbol{w}^\star, \boldsymbol{g}_t\rangle = \sum_{j=1}^{d}\sum_{t=1}^{T}(w_{t,j} - w_j^\star)g_{t,j} \leq \sqrt{2}\sum_{j=1}^{d} B_i\sqrt{\sum_{t=1}^{T} g_{t,j}^2}$$

$$\leq \sqrt{2}B_\infty \sum_{j=1}^{d}\sqrt{\sum_{t=1}^{T} g_{t,j}^2}$$

So, as before,

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}_t\right) - f(\boldsymbol{w}^\star) \leq \frac{\sqrt{2}B_\infty \sum_{j=1}^{d}\sqrt{\sum_{t=1}^{T} g_{t,j}^2}}{T}$$

Is AdaGrad Always a Good Idea?

Adaptive bound

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}_t\right) - f(\boldsymbol{w}^\star) \leq \sqrt{2}\frac{B\sqrt{\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2}}{T}$$

AdaGrad

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}_t\right) - f(\boldsymbol{w}^\star) \leq \frac{\sqrt{2}B_\infty \sum_{j=1}^{d}\sqrt{\sum_{t=1}^{T} g_{t,j}^2}}{T}$$

- It depends on B vs B_∞ and the gradients
- Hypercubes are better for AdaGrad
- Balls are better for adaptive
- Sparse gradients are better for AdaGrad

AdaGrad is Scale-free

- Having a gradient vector whose coordinates have vastly different magnitude is a problem
  - Bad "condition number"
  - Related to vanishing gradient in DNN
- Important observation: AdaGrad does not depend on the scale of each single coordinate of the gradients!
- AdaGrad is scale-free [Orabona&Pal, ALT'15]

General Recipe for Adaptive Algorithms

- Find a tight convergence bound that depends on an unknown hyperparameter
- Find the optimal setting of that hyperparameter
- Approximate it
- Try to prove a convergence rate for the approximated version