Lecture 6: Optimization Methods (II)

Soufiane Hayou

Friday 19$^{\text{th}}$ May, 2023

Lats week we discussed,

Lats week we discussed,

- Convexity

Lats week we discussed,

- Convexity
- Gradient Descent (GD)

Lats week we discussed,

- Convexity
- Gradient Descent (GD)

$\rightarrow$ The issue with GD?

- A loss function measures the discrepancy at data point $\mathbf{z} = (\mathbf{x}, \mathbf{y})$: $F(\mathbf{w}, \mathbf{z}) = \ell(h_{\mathbf{w}}(\mathbf{x}), \mathbf{y})$
- Linear regression uses l_2 loss: $F(\mathbf{w}, \mathbf{z}) = (y - \mathbf{w}^T \mathbf{x})^2$.
- Data follows a distribution $(\mathbf{x}, \mathbf{y}) \sim \mu$. Ideally, we would like to minimize

$$f(\mathbf{w}) = \mathbb{E}_{\mathbf{z}} F(\mathbf{w}, \mathbf{z}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu} \left[\ell(h_{\mathbf{w}}(\mathbf{x}), \mathbf{y})\right]$$

- Problem: expectation can be difficult to compute

Empirical loss

- Target loss function:

$$f(\mathbf{w}) = \mathbb{E}_{\mathbf{z}} F(\mathbf{w}, \mathbf{z}) = \mathbb{E}_{\mathbf{x},\mathbf{y}} \ell(h_{\mathbf{w}}(\mathbf{x}), \mathbf{y})$$

- Empirical version, draw n data points $\mathbf{z}_i$

$$f_n(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} F(\mathbf{w}, \mathbf{z}_i) \approx f(\mathbf{w})$$

- When n is large, optimization can be expensive

$$\nabla f_n(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \nabla F(\mathbf{w}, \mathbf{z}_i)$$

Running Gradient descent, each step involves n computation.

Stochastic gradient descent

Cost of Gradient descent

- Population loss gradient: we cannot obtain $\nabla_{\mathbf{w}} f(\mathbf{w}) = \mathbb{E}\left[\nabla_{\mathbf{w}} F(\mathbf{w}, \mathbf{z})\right]$
- Empirical loss gradient: require one pass of the data

$$\nabla_{\mathbf{w}} f_n(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\mathbf{w}} F(\mathbf{w}, z_i)$$

- Computational and storage cost for each update $O(n)$
- Online learning (SGD): we only use one data point z_i to update

Stochastic gradient

- If we draw $\mathbf{z}_i$ from μ

$$\mathbb{E}\left[\nabla F(\mathbf{w}, \mathbf{z}_i)\right] = \nabla \mathbb{E}[F(\mathbf{w}, \mathbf{z}_i)] = \nabla f(\mathbf{w}).$$

- If we draw $\mathbf{z}_i$ from existing data, i is uniform from $\{1, \ldots, n\}$

$$\mathbb{E}_i[\nabla F(\mathbf{w}, \mathbf{z}_i)] = \nabla \mathbb{E}_i[F(\mathbf{w}, \mathbf{z}_i)] = \nabla f_n(\mathbf{w}).$$

- $\nabla F(\mathbf{w}, \mathbf{z}_i)$ is an unbiased estimator of $\nabla f_n(\mathbf{w})$.
- We call it a stochastic gradient.
- Cheap computation cost

$$\mathbf{w}^{k+1} = \mathbf{w}^k - h_k \nabla F(\mathbf{w}^k, \mathbf{z}_k)$$

- Mean and deviation

$$\nabla F(\mathbf{w}, z_i) = \nabla f(\mathbf{w}) + \xi_i, \quad \mathbb{E}\xi_i = 0$$

- Noise variance: $\mathbb{E}(\xi_i)^2 \leq \sigma^2$

- Mean and deviation

$$\nabla F(\mathbf{w}, z_i) = \nabla f(\mathbf{w}) + \xi_i, \quad \mathbb{E}\xi_i = 0$$

- Noise variance: $\mathbb{E}(\xi_i)^2 \leq \sigma^2$

Theorem

Suppose f is c-strongly convex, ∇f is L-Lipschitz. Then running SGD with fixed stepsize $h \leq \frac{c}{L^2}$

$$\mathbb{E}\|\mathbf{w}_n - \mathbf{w}^*\| \leq (1 - ch)^n \mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \frac{\sigma^2 h}{c}$$

There is an error term in the end. Can we remove/reduce it?

Sketch of proof

$$\begin{aligned}\mathbb{E}\|w_{k+1} - w^*\|^2 &= \mathbb{E}\|w_k - w^* - h\nabla f(w_k) + \xi_k h\|^2 \\ &\leq \mathbb{E}(1 - ch)\|w_k - w^*\|^2 + \sigma^2 h^2 \\ &\leq (1 - ch)\mathbb{E}\|w_k - w^*\| + \sigma^2 h^2\end{aligned}$$

By induction we can show our claim.

How about different h_k?

- Let

$$S_{1,n} = \sum_{k=1}^{n} h_k, \quad S_{2,n} = \sum_{k=1}^{n} h_k^2.$$

Choice of stepsize

How about different h_k?

- Let

$$S_{1,n} = \sum_{k=1}^{n} h_k, \quad S_{2,n} = \sum_{k=1}^{n} h_k^2.$$

Theorem

Suppose f is c-strongly convex, ∇f is L-Lipschitz. Then running SGD with $h_k \leq \frac{c}{L^2}$

$$\min_{k \leq n} \mathbb{E}\|\mathbf{w}_k - \mathbf{w}^*\|^2 \leq \frac{\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sigma^2 S_{2,n}}{c\, S_{1,n}}$$

Using the proof of Gradient descent

$$\begin{aligned}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2 &= \mathbb{E}\|\mathbf{w}_k - \mathbf{w}^* - h_k \nabla f(\mathbf{w}_k) + \xi_k h_k\|^2 \\ &= \mathbb{E}\|\mathbf{w}_k - \mathbf{w}^* - h_k \nabla f(\mathbf{w}_k)\| + h_k^2 \sigma^2 \\ &\leq (1 - c h_k)\mathbb{E}\|\mathbf{w}_k - \mathbf{w}^*\|^2 + h_k^2 \sigma^2\end{aligned}$$

Summing over we find that

$$cS_{1,n} \sum_{k=1}^{n} \mathbb{E}\|\mathbf{w}_k - \mathbf{w}^*\|^2 \leq \mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + S_{2,n}\sigma^2.$$

Choice of stepsize

Principled choice of h_k?

Theorem

Suppose f is c-strongly convex, ∇f is L-Lipschitz. Then running SGD with $h_k \leq \frac{c}{L^2}$

$$\min_{k \leq n} \mathbb{E}\|\mathbf{w}_k - \mathbf{w}^*\|^2 \leq \frac{\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sigma^2 S_{2,n}}{c\, S_{1,n}}$$

Principled choice of h_k?

Theorem

Suppose f is c-strongly convex, ∇f is L-Lipschitz. Then running SGD with $h_k \leq \frac{c}{L^2}$

$$\min_{k\leq n} \mathbb{E}\|\mathbf{w}_k - \mathbf{w}^*\|^2 \leq \frac{\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sigma^2 S_{2,n}}{c\, S_{1,n}}$$

$\rightarrow$ It is sufficient to choose $\lim_n S_{1,n} = \infty$ and $\lim_n S_{2,n} < \infty$ to guarantee "convergence".

- Prefixed stepsize $h_k = h_0 k^{-\alpha}, \alpha \in (\frac{1}{2}, 1]$

Variance and Mini-batch

- Mean and deviation

$$\nabla F(\mathbf{w}, z_i) = \nabla f(\mathbf{w}) + \xi_i, \quad \mathbb{E}\xi_i = 0$$

- Let the variance of $\mathbb{E}(\xi_i)^2 = \sigma_{\xi,i}^2$
- A smaller variance in general improves the final performance
- We can use more samples to reduce the variance.

Variance and Mini-batch

- Mean and deviation

$$\nabla F(\mathbf{w}, z_i) = \nabla f(\mathbf{w}) + \xi_i, \quad \mathbb{E}\xi_i = 0$$

- Let the variance of $\mathbb{E}(\xi_i)^2 = \sigma_{\xi,i}^2$
- A smaller variance in general improves the final performance
- We can use more samples to reduce the variance.
- Consider

$$\nabla F(\mathbf{w}, z_{Bi+1}, \ldots, z_{(B+1)i}) = \frac{1}{B} \sum_{j=Bi+1}^{(B+1)i} \nabla F(\mathbf{w}, z_j)$$

- The variance in the stochastic gradient is only $\frac{1}{B}$ of the original.

Momentum Gradient Descent

Momentum

(Stochastic) Gradient descent depends only on the current gradient! It does not use information from previous steps.

Momentum

(Stochastic) Gradient descent depends only on the current gradient! It does not use information from previous steps.
$\rightarrow$ Include "momentum" in the updates.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - h_k m_k$$

where $m_k = \beta \times m_{k-1} + (1 - \beta) \times \nabla f_n(\mathbf{w}_k)$.

Momentum

(Stochastic) Gradient descent depends only on the current gradient! It does not use information from previous steps.
$\rightarrow$ Include "momentum" in the updates.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - h_k m_k$$

where $m_k = \beta \times m_{k-1} + (1-\beta) \times \nabla f_n(\mathbf{w}_k)$.

- β is the momentum parameter (usually chosen 0.9)
- Variants of momentum SGD achieve state-of-the-art performance (Adam, AdaGrad, etc.)
- By accumulating "speed", momentum algorithms usually move faster.
- Momentum can also help escape bad local minima.