Stochastic Gradient Descent

Numerical Methods for Deep Learning

Review: Supervised Learning Problem

Most machine learning problems are of the following structure

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \mathbf{Y}) + R(\boldsymbol{\theta}), \quad \text{with} \quad F(\boldsymbol{\theta}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta}, \mathbf{y}_i).$$

Review: Supervised Learning Problem

Most machine learning problems are of the following structure

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \mathbf{Y}) + R(\boldsymbol{\theta}), \quad \text{with} \quad F(\boldsymbol{\theta}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta}, \mathbf{y}_i).$$

For shallow learning, problem might be convex or have a unique minimum. For deep networks, problem is usually not convex and has many local minimum

Review - Optimization Techniques

So far, we used deterministic gradient-based methods

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \mathbf{A}_k \nabla F(\boldsymbol{\theta}_k, \mathbf{Y}), \quad \nabla F(\theta, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\theta, \mathbf{y}_i)$$

Review - Optimization Techniques

So far, we used deterministic gradient-based methods

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \mathbf{A}_k \nabla F(\boldsymbol{\theta}_k, \mathbf{Y}), \quad \nabla F(\theta, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\theta, \mathbf{y}_i)$$

Examples:

- steepest descent: $\mathbf{A}_k = \mathbf{I}$
- Newton: $\mathbf{A}_k = \nabla^2 F(\theta, \mathbf{Y}) = \sum_{i=1}^{N} \nabla^2 f_i(\theta, \mathbf{y}_i)$

Review - Optimization Techniques

So far, we used deterministic gradient-based methods

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \mathbf{A}_k \nabla F(\boldsymbol{\theta}_k, \mathbf{Y}), \quad \nabla F(\theta, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\theta, \mathbf{y}_i)$$

Examples:

- steepest descent: $\mathbf{A}_k = \mathbf{I}$
- Newton: $\mathbf{A}_k = \nabla^2 F(\theta, \mathbf{Y}) = \sum_{i=1}^{N} \nabla^2 f_i(\theta, \mathbf{y}_i)$

Drawbacks:

- Evaluating gradient needs pass through the whole data set (called *epoch*).
- If data is redundant can be very expensive
- Idea: use only a part of the data to update $\boldsymbol{\theta}$

Stochastic Gradient Descent

Let $\mathcal{S}_k \subset \{1, 2, \ldots, n\}$. Define the batch objective function as

$$F_{\mathcal{S}_k}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} f_i(\boldsymbol{\theta}, \mathbf{Y}_i)$$

Then a straight forward extension is

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \mathbf{A}_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$

Questions

- Would the method converge?
- Under what conditions on $\mu_k, \mathbf{A}_k, \mathcal{S}_k$?
- How fast?

References: original method [4], recent surveys [2, 1, 3]

Stochastic Gradient Descent

Let $\mathcal{S}_k \subset \{1, 2, \ldots, n\}$. Define the batch objective function as

$$F_{\mathcal{S}_k}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} f_i(\boldsymbol{\theta}, \mathbf{Y}_i)$$

Then a straight forward extension is

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \mathbf{A}_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$

Stochastic Gradient Descent

Let $\mathcal{S}_k \subset \{1, 2, \ldots, n\}$. Define the batch objective function as

$$F_{\mathcal{S}_k}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} f_i(\boldsymbol{\theta}, \mathbf{Y}_i)$$

Then a straight forward extension is

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \mathbf{A}_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$

If $\mathbf{A}_k = \mathbf{I}$, $|\mathcal{S}_k| = 1$ and $\mu_k \to 0$ slow enough, that is

$$\sum_{k=1}^{\infty} \mu_k = \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \mu_k^2 < \infty$$

then SGD converges to stationary point

Stochastic Gradient Descent

Let $\mathcal{S}_k \subset \{1, 2, \ldots, n\}$. Define the batch objective function as

$$F_{\mathcal{S}_k}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} f_i(\boldsymbol{\theta}, \mathbf{Y}_i)$$

Then a straight forward extension is

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \mathbf{A}_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$

If $\mathbf{A}_k = \mathbf{I}$, $|\mathcal{S}_k| = 1$ and $\mu_k \to 0$ slow enough, that is

$$\sum_{k=1}^{\infty} \mu_k = \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \mu_k^2 < \infty$$

then SGD converges to stationary point (Ex: $\mu_k = k^{-1}$).

Stochastic Gradient Descent

Let $\mathcal{S}_k \subset \{1, 2, \ldots, n\}$. Define the batch objective function as

$$F_{\mathcal{S}_k}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} f_i(\boldsymbol{\theta}, \mathbf{Y}_i)$$

Then a straight forward extension is

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \mathbf{A}_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$

If $\mathbf{A}_k = \mathbf{I}$, $|\mathcal{S}_k| = 1$ and $\mu_k \to 0$ slow enough, that is

$$\sum_{k=1}^{\infty} \mu_k = \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \mu_k^2 < \infty$$

then SGD converges to stationary point (Ex: $\mu_k = k^{-1}$).

How fast? Convergence is **sublinear**

A Glimpse into the theory

Consider the iteration and $\mathbf{A}_k = \mathbf{I}$

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$

A Glimpse into the theory

Consider the iteration and $\mathbf{A}_k = \mathbf{I}$

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$

Re-write this as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \underbrace{\mu_k \nabla F(\boldsymbol{\theta}, \mathbf{Y})}_{\text{true gradient}} - \underbrace{\mu_k \left(\nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k) - \nabla F(\boldsymbol{\theta}, \mathbf{Y})\right)}_{\text{noise}}$$

A Glimpse into the theory

Consider the iteration and $\mathbf{A}_k = \mathbf{I}$

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$

Re-write this as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \underbrace{\mu_k \nabla F(\boldsymbol{\theta}, \mathbf{Y})}_{\text{true gradient}} - \underbrace{\mu_k \left(\nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k) - \nabla F(\boldsymbol{\theta}, \mathbf{Y})\right)}_{\text{noise}}$$

Note that (unbiased estimator)

$$\mathbb{E}(\nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)) = \nabla F(\boldsymbol{\theta}).$$

A Glimpse into the theory

Consider the iteration and $\mathbf{A}_k = \mathbf{I}$

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mu_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$

Re-write this as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \underbrace{\mu_k \nabla F(\boldsymbol{\theta}, \mathbf{Y})}_{\text{true gradient}} - \underbrace{\mu_k \left(\nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k) - \nabla F(\boldsymbol{\theta}, \mathbf{Y})\right)}_{\text{noise}}$$

Note that (unbiased estimator)

$$\mathbb{E}(\nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)) = \nabla F(\boldsymbol{\theta}).$$

Finally note that

$$\mathrm{Var}\left(\mu_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)\right) = \mu_k^2 \mathrm{Var}\left(\nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)\right)$$

Improvements of SGD: Momentum

Idea: Accelerate convergence by keeping gradient informations from previous batches.

$$\mathbf{S}_{k+1} = \gamma \mathbf{S}_k + \mu_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$
$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mathbf{S}_{k+1}$$

μ_k - learning rate, γ - momentum

Improvements of SGD: Momentum

Idea: Accelerate convergence by keeping gradient informations from previous batches.

$$\mathbf{S}_{k+1} = \gamma \mathbf{S}_k + \mu_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k)$$
$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mathbf{S}_{k+1}$$

μ_k - learning rate, γ - momentum

Hard to choose in practice, heuristic
γ - Start with 0.5 and increase slowly to 0.9
μ - problem dependent start small and decrease after a few epoch

Improvements of SGD: Nesterov

Idea: Predict next iterate using momentum, correct next step using gradient there.

$$
\begin{aligned}
\boldsymbol{\theta}_{k+\frac{1}{2}} &= \boldsymbol{\theta}_k - \gamma \mathbf{S}_k \\
\mathbf{S}_{k+1} &= \gamma \mathbf{S}_k + \mu_k \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_{k+\frac{1}{2}}) \\
\boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k - \mathbf{S}_{k+1}
\end{aligned}
$$

Improvements of SGD: AdaGrad

Idea: Scale step according to size of weights (relation to prior-conditioning in SGD)

Iteration:

$$\begin{aligned}
\mathbf{D}_{k+1} &= \boldsymbol{\theta}_k^2 + \mathbf{D}_k \\
\mathbf{S}_{k+1} &= \mu_k \operatorname{diag}(\mathbf{D}_{k+1})^{-1} \nabla F_{\mathcal{S}_k}(\boldsymbol{\theta}_k) \\
\boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k - \mathbf{S}_{k+1}
\end{aligned}$$

Theory and Final Comments

General Comments:

- Lots of theory for convex problems
- Recall: SGD is not the best tool for most convex problems (see example of least-squares)
- Require very careful tuning

SGD in deep learning:

- currently the main workhorse (DNN $\rightsquigarrow$ nonconvex optimization)
- why it works? mostly open but some relation to Langevin flow (we also have a few ideas)
- observed to regularize problems (theory for quadratic case)
- potentially possible to prove global optimality?

Coding: Using SGD for Classification Problem

Outline:

- Use single layer or ResNet example
- Change objective function to accept index set S_k
- Use small minibatch
- Test using peaks example

References

[1] D. P. Bertsekas. Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey. *arXiv.org*, July 2015.

[2] L. Bottou. Stochastic gradient descent tricks. *Neural networks: Tricks of the trade*, 2012.

[3] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization Methods for Large-Scale Machine Learning. *arXiv.org*, June 2016.

[4] H. Robbins and S. Monro. A Stochastic Approximation Method. *The annals of mathematical statistics*, 22(3):400–407, 1951.