Optimization using gradient descent

Huy L. Nguyễn

In the next few lectures, we will talk about optimization using gradient descent. We saw earlier in the course that when we have linear constraints and linear objective, we can optimize in polynomial time. Unfortunately generalizing to non-linear constraints/objective makes the problem much harder.

For instance, we can have a non-linear constraint that forces all variables to be either 0 or 1:

$$\sum_i x_i^2(1-x_i)^2 = 0$$

Thus, optimization with this constraint is NP-hard.

Among non-linear problems, it turns out that convex problems (to be defined) are solvable in polynomial time and non-convex problems are generally hard. Gradient descent is a popular method for both of these types of problems. It can find the global optimum for convex problems under very general conditions. For non-convex problems, it finds a local optimum.

1 Calculus review and properties of convex functions

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$. The partial derivative of f with respect to x_i is

$$\frac{\partial f(x)}{\partial x_i} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon e_i) - f(x)}{\varepsilon}$$

For example, consider $f : \mathbb{R}^2 \to \mathbb{R}$ where $f(x) = x_1^2 + 2x_1x_2$.

$$\frac{\partial f}{\partial x_1} = 2x_1 + 2x_2; \quad \frac{\partial f}{\partial x_2} = 2x_1$$

The gradient vector of f is

$$\nabla f(x) = Df(x)^T = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \cdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

The gradient points in the direction where the function increases the most rapidly.

For functions with high dimensional output, $f : \mathbb{R}^n \to \mathbb{R}^m$ and $f(x) = (f_1(x), \ldots, f_m(x))$, we have the Jacobian matrix whose entries are the partial derivatives.

$$Df(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ & \cdots & \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

The gradient gives us the best local approximation of a function $f(x)$ around x_0 using a linear function.

$$f(x) \approx f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle$$

We can also try to approximate better using higher degree polynomials. The Hessian is the matrix of second order partial derivatives of f:

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ & \cdots & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

The Hessian is a symmetric matrix. The second order approximation of f around x_0 is

$$f(x) \approx f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle + \frac{1}{2} \cdot (x - x_0)^T \nabla^2 f(x_0)(x - x_0)$$

More precisely, Taylor's theorem states that for any x, x_0, there exists x_1 on the line segment between x and x_0 such that

$$f(x) = f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle + \frac{1}{2} \cdot (x - x_0)^T \nabla^2 f(x_1)(x - x_0)$$

We often compute derivatives of functions that are compositions of many simpler functions. The following rules are often useful in such cases. Consider two function $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^m$

$$D(f(x) + g(x)) = Df(x) + Dg(x)$$
$$D(f(x)^T g(x)) = g(x)^T (Df(x)) + f(x)^T (Dg(x))$$

We also have the chain rule for composition of $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^m \to \mathbb{R}^p$ to get $h(x) = g(f(x))$.

$$Dh(x) = Dg(f) \cdot Df(x)$$

A common use-case of the chain rule is the composition with an affine function. Consider $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g(x) = f(Ax + b)$.

$$Dg(x) = Df(Ax + b) \cdot A$$

For the special case $m = 1$, we have

$$\nabla g = A^T \nabla f$$
$$\nabla^2 g = A^T \nabla^2 f A$$

2 Convex optimization

As mentioned before, general optimization is hard. However, we will focus on a special case that is tractable: minimizing a convex function over a convex domain.

A set S is convex if for any two points $x, y \in S$, we have $\theta x + (1 - \theta) y \in S \ \forall \theta \in [0, 1]$ i.e. the whole line segment connecting x, y is in S.

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if for any two points x, y in the domain and any $\theta \in [0, 1]$, we have $\theta f(x) + (1 - \theta) f(y) \geq f(\theta x + (1 - \theta) y)$. When the function is differentiable, it is convex if and only if

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle \ \forall x, y$$

When the function is twice differentiable, it is convex if and only if the Hessian is positive semidefinite i.e. $y^T H(f(x)) y \geq 0 \ \forall y$.

The class of convex functions is actually rather broad. Below we consider a few examples.

- $f(x) = \log(e^{x_1} + \cdots + e^{x^n})$ is convex in $\mathbb{R}^n$. This fact is central to the multiplicative weight updates.
- Every norm is convex. This is because it satisfies the triangle inequality.
- $f(x) = \|Ax - b\|_2^2$. This is the objective of least squares regression.

Why is minimizing a convex function over a convex set tractable? The following theorem gives one explanation.

Theorem 2.1. *Any local minimum is a global minimum.*

Proof. Suppose x^* is a local minimum and $y \neq x$ is the global minimum.

Suppose that $f(y) < f(x^*)$. For any $\theta \in [0, 1]$, we have $f(\theta x^* + (1-\theta)y) \leq \theta f(x^*) + (1-\theta)f(y) < f(x^*)$. For θ close to 1, this contradicts the fact that x^* is a local minimum. □

Thus, our goal now is to find a local minimum.

Theorem 2.2. *Consider a convex and differentiable $f : \mathbb{R}^n \to \mathbb{R}$ and a convex set S. A point x^* is a global minimum if and only if*

$$\langle \nabla f(x^*), x - x^* \rangle \geq 0 \ \forall x \in S$$

Proof. If $\langle \nabla f(x^*), x - x^* \rangle \geq 0 \ \forall x$ then

$$f(x) \geq f(x^*) + \langle \nabla f(x^*), x - x^* \rangle \geq f(x^*)$$

On the other hand, suppose that x^* is a global minimum. Assume for contradiction that for some $x \in S$, we have

$$\langle \nabla f(x^*), x - x^* \rangle < 0$$

Let $g(\theta) = f(x^* + \theta(x - x^*))$. By Taylor's theorem at 0,

$$g(\theta) = g(0) + g'(0)\theta + O(\theta^2) = f(x^*) + \langle \nabla f(x^*), x - x^* \rangle + O(\theta^2)$$

For sufficiently small θ, $g(\theta) < g(0)$, which contradicts the fact that x^* is a global minimum. □

The proof reveals that if we are at a point x and there is a point y such that $\langle \nabla f(x), y - x \rangle < 0$ then moving a bit in the direction $y - x$ will decrease the function value.

3 Gradient descent

As noted above, if we have a current solution x and there is a direction $y - x$ such that $\langle \nabla f(x), y - x \rangle < 0$ then we can improve the solution by moving a bit in that direction. Such a direction is called a *descent direction*. When there are multiple descent directions, which one should we choose? One strategy is greedy: go in the direction that descends the fastest, which is the gradient. This choice leads to the gradient descent algorithm for $\min_{x \in \mathbb{R}^n} f(x)$.

```
Initialize $x^{(0)}$
for $t$ from 1 to $T$ do
    choose step size $\eta_t$
    $x^{(t)} \leftarrow x^{(t-1)} - \eta_t \nabla f(x^{(t-1)})$
end for
```

```
return $x^{(T)}$ or $\frac{1}{T}\sum_{t=1}^{T} x^{(t)}$
```

How should we choose the step? On the one hand, we would like to improve our solution quickly and thus, pick a large step. On the other hand, as we move, the gradient also changes and we might overshoot if the step is too large. There is no universal answer to this question. One possible answer is to pick the best possible step size i.e. pick η_t to minimize $f(x^{(t-1)} - \eta_t \nabla f(x^{(t-1)}))$. The downside of this choice is that optimizing step size might be expensive. Other common choices are fixed step size: $\eta_t = \eta \ \forall t$ and time decaying step size: $\eta_t = \Theta(1/t)$.

Let's analyze the performance of the algorithm and see how to set the step sizes if we go with the simpler options above. The idea is to use a potential function to track our progress. We will use mainly two quantities to track how good our current solution is:

- Function value distance: $f(x^{(t)}) - f(x^*)$
- Distance to the optimum: $\|x^{(t)} - x^*\|^2$

Our potential is a linear combination of these two quantities:

$$\Phi_t = a_t(f(x^{(t)}) - f(x^*)) + b_t\|x^{(t)} - x^*\|^2$$

In each time step, we will bound the change in the potential i.e. showing that $\Phi_t - \Phi_{t-1} \le B_t$. At the end, we take the telescoping sum and obtain

$$\Phi_T \le \Phi_0 + \sum_{i=1}^{T} B_t$$

4 Gradient descent with bounded gradient

To be able to set the step size and analyze the performance, we need to know a bit more about the objective function. Suppose the length of the gradient of the function is always bounded by G. We will analyze our algorithm with a fixed step size $\eta_t = \eta \ \forall t$ and the potential $\Phi_t = \frac{1}{2\eta}\|x^{(t)} - x^*\|^2$.

$$\begin{aligned}
\Phi_t - \Phi_{t-1} &= \frac{1}{2\eta}(\|x^{(t)} - x^*\|^2 - \|x^{(t-1)} - x^*\|^2) \\
&= \frac{1}{2\eta}\left\langle x^{(t)} - x^{(t-1)}, x^{(t)} + x^{(t-1)} - 2x^* \right\rangle \\
&= \frac{1}{2\eta}\left\langle -\eta\nabla f(x^{(t-1)}), -\eta\nabla f(x^{(t-1)}) + 2x^{(t-1)} - 2x^* \right\rangle \\
&= \frac{1}{2\eta}\left(\eta^2\|\nabla f(x^{(t-1)})\|^2 + 2\eta\langle \nabla f(x^{(t-1)}), x^* - x^{(t-1)}\rangle\right) \\
&= \frac{\eta}{2}\|\nabla f(x^{(t-1)})\|^2 + \underbrace{\langle \nabla f(x^{(t-1)}), x^* - x^{(t-1)}\rangle}_{\le f(x^*) - f(x^{(t-1)})} \\
&\le \frac{\eta G^2}{2} + f(x^*) - f(x^{(t-1)})
\end{aligned}$$

Our telescoping sum gives us

$$\Phi_T - \Phi_0 \le \frac{T\eta G^2}{2} + \sum_{t=1}^{T}(f(x^*) - f(x^{(t-1)}))$$

By shuffling the terms around, we obtain

$$\sum_{t=1}^{T}(f(x^{(t-1)}) - f(x^*)) \leq \frac{T\eta G^2}{2} + \Phi_0 - \Phi_T \leq \frac{T\eta G^2}{2} + \Phi_0$$

By convexity,

$$\sum_{t=1}^{T} f(x^(t-1)) \geq Tf\left(\frac{1}{T}\sum_{t=1}^{T} x^{(t-1)}\right) =: f(\bar{x})$$

Therefore,

$$f\left(\frac{1}{T}\sum_{t=1}^{T} x^{(t-1)}\right) - f(x^*) \leq \frac{\eta G^2}{2} + \frac{\|x^{(0)} - x^*\|^2}{2T\eta}$$

We would like to pick η to minimize the RHS. Suppose $\|x^{(0)} - x^*\| \leq R$.Thus,

$$\frac{\eta G^2}{2} = \frac{R^2}{2T\eta} \implies \eta = \frac{R}{G\sqrt{T}}$$

With this step size, our guarantee is

$$f(\bar{x}) - f(x^*) \leq \frac{RG}{\sqrt{T}}$$

This guarantee is usually referred to as $\frac{1}{\sqrt{T}}$ convergence.

5 Smooth functions

We can obtain better results for more restrictive classes of functions. Recall that we can only set a small step because the gradient changes as we change the solution. A function is smooth if the change in the gradient is bounded by how far the solution changes. More precisely, a function is β-smooth if for any two points x, y, we have

$$\|\nabla f(x) - \nabla f(y)\| \leq \beta\|x - y\|$$

Note that this implies for any x, y

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\beta}{2}\|y - x\|^2$$

Thus, we can always bound the objective using a quadratic function. In order to find the next location for the solution, we minimize the quadratic approximation at the current point. The solution turns out to be $\eta_t = 1/\beta$. With this choice, we have

$$f(x^{(t)}) \leq f(x^{(t-1)}) - \frac{1}{2\beta}\|\nabla f(x^{(t-1)}\|^2 \tag{1}$$

We analyze this algorithm using the potential $\Phi_t = t(f(x^{(t)}) - f(x^*)) + \frac{\beta}{2}\|x^{(t)} - x^*\|^2$.

The change in the potential is

$$\Phi_t - \Phi_{t-1} = t(f(x^{(t)}) - f(x^{(t-1)})) + (f(x^{(t-1)}) - f(x^*)) + \frac{\beta}{2}(\|x^{(t)} - x^*\|^2 - \|x^{(t-1)} - x^*\|^2)$$

We can bound the first term by smoothness:

$$f(x^{(t)}) - f(x^{(t-1)}) \le \langle \nabla f(x^{(t-1)}), x^{(t)} - x^{(t-1)} \rangle + \frac{\beta}{2}\|x^{(t)} - x^{(t-1)}\|^2 \le -\frac{\|\nabla f(x^{(t-1)})\|^2}{2\beta}$$

We can bound the second term using convexity:

$$f(x^{(t-1)}) - f(x^*) \le \langle \nabla f(x^{(t-1)}), x^{(t-1)} - x^* \rangle$$

We can bound the third term using the same argument as the general case:

$$\begin{aligned} \frac{\beta}{2}\left(\|x^{(t)} - x^*\|^2 - \|x^{(t-1)} - x^*\|^2\right) &= \frac{\beta}{2}\left(\langle x^{(t)} - x^{(t-1)}, x^{(t)} + x^{(t-1)} - 2x^* \rangle\right) \\ &\le \frac{\beta}{2}\left(\eta_t^2 \|\nabla f(x^{(t-1)})\|^2 + 2\eta_t \langle \nabla f(x^{(t-1)}), x^* - x^{(t-1)} \rangle\right) \end{aligned}$$

Notice that the inner product terms cancel and we are left with

$$\Phi_t - \Phi_{t-1} \le -\frac{t-1}{2\beta}\|\nabla f(x^{(t-1)})\|^2 \le 0$$

Therefore, the telescoping sum gives us

$$T(f(x^{(T)}) - f(x^*)) \le \Phi_T \le \Phi_0 = \frac{\beta}{2}\|x^{(0)} - x^*\|^2$$

This is referred to as $1/T$ convergence.

6 Constrained optimization

We now extend our results to the constrained case $\min_{x \in S} f(x)$ for a convex set S. In the unconstrained case, we take the quadratic approximation of the function at the current solution and the next solution is the minimizer of the quadratic approximation. Notice that this step is still meaningful when we have constraints. Thus our algorithm is

$$x^{(t)} \leftarrow \operatorname*{argmin}_{z \in S} f(x^{(t-1)}) + \langle \nabla f(x^{(t-1)}), z - x^{(t-1)} \rangle + \frac{\beta}{2}\|z - x^{(t-1)}\|^2$$

Another idea is to move in the direction of the gradient, which might take us out of the feasible region, and then project back to the feasible region. In other words, our algorithm is

$$\begin{aligned} y^{(t)} &\leftarrow x^{(t-1)} - \eta_t \nabla f(x^{(t-1)}) \\ x^{(t)} &\leftarrow \operatorname*{argmin}_{x \in S} \|y^{(t)} - x\| \end{aligned}$$

It turns out that for step size $\eta_t = 1/\beta$, these two algorithms are identical. In order to analyze this algorithm, we need a property of the projection operation.

Lemma 6.1. *Given a convex set S, let $a \in S$ and $b' \in \mathbb{R}^n$. Let $b = \operatorname{argmin}_{x \in S} \frac{1}{2}\|x - b'\|^2$. Then $\langle a - b, b - b' \rangle \ge 0$ and therefore, $\|a - b\|^2 \le \|a - b'\|^2$.*

Proof. The lemma follows from the optimality of b. The gradient of $\frac{1}{2}\|x - b'\|^2$ at $x = b$ is $b - b'$. Because of the optimality of b, we have $\langle a - b, b - b' \rangle \ge 0$. □

Using this property, we obtain $\|x^{(t)} - x^*\|^2 \le \|y^{(t)} - x^*\|^2$ and we can observe that the rest of the original proof goes through in the constrained setting.

7 Stochastic gradient descent

In many situation, the objective function we would like to minimize can be decomposed into many terms: $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ where each function f_i is convex. An important example is empirical risk minimization in machine learning. The time complexity to compute $\nabla f(x)$ is high: we need to compute $\nabla f_i(x)$ for all i and sum them up. In this section, we consider a new method that offers a significant running time improvement in some situations. The idea is to use randomness and instead of computing $\nabla f(x)$ exactly, we compute a random vector with high correlation with the correct gradient (we have seen a very similar style of algorithm when we talk about hashing earlier). For example, in the decomposable setting above, one can simply sample a random index i and return $g(x) = \nabla f_i(x)$. The vector we sample $g(x)$ has the right expectation $\mathbb{E}[g(x)] = \nabla f(x)$.

We now consider the simplest setting so far: we would like to minimize f over a convex set S of diameter R and each f_i is assumed to have gradient of length at most G. Our stochastic algorithm is as follows.

```
Initialize $x^{(0)}$
for $t$ from 1 to $T$ do
    compute a stochastic $g(x^{(t-1)})$ and choose step size $\eta_t$
    $y^{(t)} \leftarrow x^{(t-1)} - \eta_t g(x^{(t-1)})$
    $x^{(t)} \leftarrow \operatorname{argmin}_{x \in S} \|y^{(t)} - x\|$
end for
return $\frac{1}{T}\sum_{t=1}^{T} x^{(t)}$
```

Using the same analysis as before with $\Phi_t = \frac{1}{2\eta}\|x^{(t)} - x^*\|^2$,

$$\begin{aligned}
\Phi_t - \Phi_{t-1} &= \frac{1}{2\eta}(\|x^{(t)} - x^*\|^2 - \|x^{(t-1)} - x^*\|^2) \\
&\le \frac{1}{2\eta}(\|y^{(t)} - x^*\|^2 - \|x^{(t-1)} - x^*\|^2) \\
&= \frac{1}{2\eta}\left\langle y^{(t)} - x^{(t-1)}, y^{(t)} + x^{(t-1)} - 2x^* \right\rangle \\
&= \frac{1}{2\eta}\left\langle -\eta g(x^{(t-1)}), -\eta g(x^{(t-1)}) + 2x^{(t-1)} - 2x^* \right\rangle \\
&= \frac{1}{2\eta}\left(\eta^2 \|g(x^{(t-1)})\|^2 + 2\eta\langle g(x^{(t-1)}), x^* - x^{(t-1)}\rangle\right)
\end{aligned}$$

Taking expectation conditioned on $x^{(t-1)}$,

$$\begin{aligned}
\mathbb{E}[\Phi_t - \Phi_{t-1} | x^{(t-1)}] &\le \frac{\eta}{2}G^2 + \underbrace{\langle \nabla f(x^{(t-1)}), x^* - x^{(t-1)}\rangle}_{\le f(x^*) - f(x^{(t-1)})} \\
&\le \frac{\eta G^2}{2} + f(x^*) - f(x^{(t-1)})
\end{aligned}$$

As before pick $\eta = \frac{R}{G\sqrt{T}}$. With this step size, our guarantee is

$$\mathbb{E}[f(\bar{x})] - f(x^*) \le \frac{RG}{\sqrt{T}}$$

Note that this is exactly the same convergence as our deterministic algorithm!