First and second order methods

(I.) First order methods : involving only the gradient information.

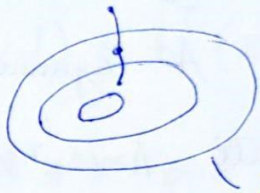
$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$$

$d^{(k)}$ ↑ descent direction

$$f(x^{(k+1)}) < f(x^{(k)})$$

(1.) Gradient descent = steepest descent

Denote $g^{(k)} = \nabla f(x^{(k)})$. Then in the method we have $d^{(k)} = -\frac{g^{(k)}}{\|g^{(k)}\|}$



- What is the actual path?

Ex.: $f(x) = x_1 x_2^2$. $\nabla f(x) = (x_2^2, 2x_1x_2)$

$x^{(k)} = (1,2) \Rightarrow \nabla f(x^{(k)}) = (2^2, 2\cdot 1\cdot 2) = (4,4)$

$\Rightarrow d^{(k)} = \frac{-1}{\sqrt{4^2+4^2}}(4,4) = \left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$.

- What is the actual path? Suppose we do the exact line search, that is at each step we solve

$$\alpha^{(k)} = \arg\min_{\alpha} f(x^{(k)} + \alpha d^{(k)})$$

$$\Rightarrow \frac{d}{d\alpha} f(x^{(k)} + \alpha d^{(k)})\Big|_{\alpha=\alpha^{(k)}} = 0 \Rightarrow \nabla f(x^{(k)} + \alpha^{(k)} d^{(k)})^T \cdot d^{(k)} = 0$$

But $d^{(k+1)} = -\frac{\nabla f(x^{(k)} + \alpha d^{(k)})}{\|\nabla f(x^{(k)} + \alpha d^{(k)})\|} \Rightarrow (d^{(k+1)})^T \cdot d^{(k)} =$

$= \langle d^{(k+1)}, d^{(k)} \rangle = 0. \Rightarrow$

$\Rightarrow d^{(k+1)} \perp d^{(k)}$.

*Note: Approximate line search doesn't satisfy this

Level lines of a function.

(*) Hence, it might take many steps to progress in a 'long valley'

(2) Conjugate gradient descent

This method overcomes (*): suppose we want to optimize a quadratic function

$$\min_x f(x) = \frac{1}{2} x^T A x + b^T x + c,$$

A symmetric, positive definite

} For non-quadratic, we apply using quadratic approximation

(recall, $A \succ 0 \Leftrightarrow x^T A x > 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\}$)

(also recall, that A pos. def. $\Leftrightarrow$  all such matrices have positive determinants)

Sylvester's criterion.

$\Rightarrow \det A > 0 \Rightarrow A$ invertible

$$\nabla f(x) = Ax + b = 0 \Rightarrow x^* = -A^{-1} b.$$

$$\nabla^2 f(x) = Hf(x) = A > 0$$

} $\Rightarrow x^*$ is global minimum.

- We can define a scalar product:

$$\langle x, y \rangle_A := x^T A y = \langle x, Ay \rangle$$

↙ since it is symmetric, order of vectors doesn't matter.

- Two vectors x, y are conjugate iff $\langle x, y \rangle_A = 0 \Leftrightarrow x^T A y = 0$.
- Take a basis in this space: $\{p_1, \dots p_n\}$

Any vector can be expressed in this basis: in particular,

$$x^* = \sum_{i=1}^{n} \beta_i p_i \Rightarrow$$

$$\Rightarrow Ax^* = \sum_{i=1}^{n} \beta_i A p_i \Rightarrow b = \sum_{i=1}^{n} \beta_i A p_i \Rightarrow$$

$$\Rightarrow \langle p_i, b \rangle = \beta_i \langle A p_i, p_i \rangle \Rightarrow \boxed{\beta_i = \frac{\langle p_i, b \rangle}{\langle p_i, p_i \rangle_A}}.$$

- We want to find it iteratively:

(1) Start with $d^{(1)} = -g^{(1)}$

(2) Compute α exactly:

Suppose $\min_{\alpha} f(x + \alpha d)$:

$$\frac{\partial f(x+\alpha d)}{\partial \alpha} = \frac{\partial}{\partial \alpha}\left[\frac{1}{2}(x+\alpha d)^T A (x+\alpha d) + b^T (x+\alpha d) + c\right]$$

$$= d^T A (x+\alpha d) + d^T b = d^T A x + \alpha d^T A d + d^T b$$

$$= d^T (Ax + b) + \alpha d^T A d = 0 \Rightarrow$$

$$\Rightarrow \boxed{\alpha = -\frac{d^T (Ax+b)}{d^T A d}} \qquad \alpha^{(k)} = -\frac{d^{(k)T} g^{(k)}}{\langle d^{(k)}, d^{(k)} \rangle_A} = -\frac{\langle d^{(k)}, g^{(k)} \rangle}{\langle d^{(k)}, d^{(k)} \rangle_A}$$

$\rightsquigarrow$ $x^{(2)} = x^{(1)} + \alpha^{(1)} d^{(1)}$. Same for $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$

Ex.: This ~~a~~ descent direction

(3) $\{d^{(k+1)} = -g^{(k+1)} + \beta^{(k)} d^{(k)}\}$, where $\beta^{(k)}$ is

($g^{(k+1)}$: gradient)

chosen so that $\langle d^{(k+1)}, d^{(k)} \rangle_A = 0$ (conjugate direction)

$$0 = d^{(k+1)T} A d^{(k)} = \left(-g^{(k+1)T} + \beta^{(k)} d^{(k)T}\right) A d^{(k)} =$$

$$= -g^{(k+1)T} A d^{(k)} + \beta^{(k)} d^{(k)T} A d^{(k)} \Rightarrow \beta^{(k)} = \frac{g^{(k+1)T} A d^{(k)}}{d^{(k)T} A d^{(k)}}$$

$$\Leftrightarrow \boxed{\beta^{(k)} = \frac{\langle g^{(k+1)}, d^{(k)} \rangle_A}{\langle d^{(k)}, d^{(k)} \rangle_A}} \qquad \underline{\text{Recall}}: \langle g^{(k+1)}, d^{(k)} \rangle = 0$$

- Fletcher - Reeves: (*)

$$\beta^{(k)} = \frac{g^{(k)T} g^{(k)}}{g^{(k-1)T} g^{(k-1)}}$$

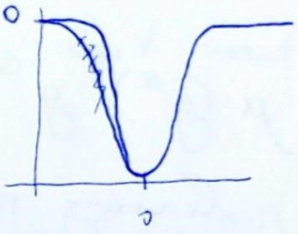
- Polak - Ribière:

$$\beta^{(k)} = \frac{g^{(k)T} \left(g^{(k)} - g^{(k-1)}\right)}{g^{(k-1)T} g^{(k-1)}}$$

Thm: For quadratic func's, the conjugate gradient descent converges in at most n (= dimension) steps.
◀ w/o proof ▶
If we don't know / want to compute Hessian, we continue as follows: (*)

(3.) Momentum:

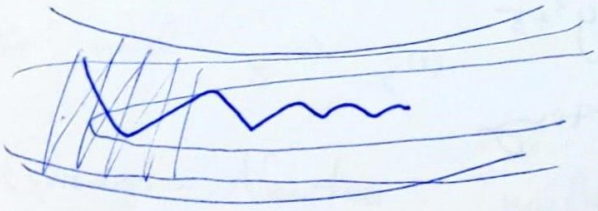
Plato's: $-e^{-x^2}$



⇒ Accumulate momentum for descent.

$$\begin{cases} v^{(k+1)} = \beta v^{(k)} - \alpha g^{(k)} \\ x^{(k+1)} = x^{(k)} + v^{(k+1)} \end{cases}$$

Cumulative momentum?



(4.) Nesterov momentum:

$$\begin{cases} v^{(k+1)} = \beta v^{(k)} - \alpha \nabla f\left(x^{(k)} + \beta v^{(k)}\right) \\ x^{(k+1)} = x^{(k)} + v^{(k)} \end{cases}$$

(future position)

(5.) Adagrad (adaptive gradient)

For sparse gradients:

$$s_i^{(k)} = \sum \left(g_i^{(j)}\right)^2$$

↑ infrequent updates.

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\alpha}{\varepsilon + \sqrt{s_i^{(k)}}} g_i^{(k)}$$

ε: 10^{-8}

$s_i^{(k)}$: non-decreasing → Learning rate drops.

⇒ accelerated methods. (RMS Prop, Adadelta, Adam)
↑ adaptive momentum.

(6.) Hypergradient descent: adapt learning rate.

⇒ derivative of learning rate.

$$\frac{\partial f(x^{(k)})}{\partial \alpha} = -(g^k)^T g^{(k-1)}$$

$$\Rightarrow \alpha^{(k+1)} = \alpha^{(k)} - \mu \frac{\partial f(x^{(k)})}{\partial \alpha} =$$

$$= \alpha^{(k)} + \mu (g^k)^T g^{(k-1)}$$ • Two step to remember

μ = hypergradient learning rate.

Can be combined w/ previous methods.

Ex: Do 2 steps of conjugate grad. descent

$f(x,y) = x^2 + xy + y^2 + 5$

$(x,y) = (1,1)$ starting

Do 2 steps of hypergrad. descent

(II) Second order methods.: Use quadratic approx.

Newton's method:

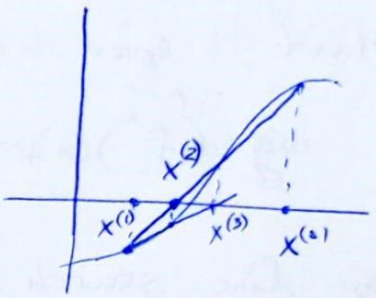
$$x^{(k+1)} = x^{(k)} - \left(H^{(k)}\right)^{-1} g^{(k)}$$

(in 1D: $x^{(k+1)} = x^{(k)} - \dfrac{f'(x^{(k)})}{f''(x^{(k)})}$) ← roots of f'.

- Works well near minimum (when $H^{(k)} > 0$)

① Secant method: don't want f'' (in 1D)

$$f''(x^{(k)}) \approx \frac{f'(x^{(k)}) - f'(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$



and $$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f'(x^{(k)}) - f'(x^{(k-1)})} f'(x^{(k)})$$

We need $x^{(0)}$, $x^{(1)}$

② Quasi-Newton: "multidim. secant" — see below

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} Q^{(k)} g^{(k)}$$

$Q^{(k)}$ ~ inverse Hessian

- Davidon-Fletcher-Powell (DFP) $Q^{(1)} = Id$

$$\gamma^{(k+1)} = g^{(k+1)} - g^{(k)}$$

$$\delta^{(k+1)} = x^{(k+1)} - x^{(k)}$$

$$Q \leftarrow Q - \frac{Q\gamma(\gamma^T Q^T)}{\gamma^T Q \gamma} + \frac{\delta\delta^T}{\delta^T\gamma}$$

1) Q remains symmetric, positive definite

2) $f(x) = \frac{1}{2} x^T A x + b^T x + c \Rightarrow Q = A^{-1}$

(2) Quasi-Newton method: 'multidim secant'

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \underbrace{Q}_{\text{inverse Hessian}} {}^{(k)}_{H}\, g^{(k)}$$

There are two methods

- Davidon – Fletcher – Powell (DFP) historically first
- Broyden – Fletcher – Goldfarb – Shanno (BFGS)

(also, L-BFGS, with limited memory)

These methods are ~~dly~~ dual to each other.

$$Q_k(x) = y^{(k)} + (g^{(k)})^T (x - x^{(k)}) + \frac{1}{2} (x - x^{(k)})^T H^{(k)} (x - x^{(k)})$$

Do the line search (for the starting function):

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} (H^{(k)})^{-1} g^{(k)}$$

This gives $x^{(k+1)}$. Now we need to find the next $H^{(k+1)} \sim H(x^{(k+1)}) = \nabla^2 f(x^{(k+1)})$.

Conditions:

$$\left.\begin{aligned} \nabla Q_{k+1}(x^{(k+1)}) &= g^{(k+1)} \\ \nabla Q_{k+1}(x^{(k)}) &= g^{(k)} \\ Q_{k+1}(x^{(k+1)}) &= y^{(k+1)} \end{aligned}\right\} \Rightarrow H^{(k+1)} \underbrace{(x^{(k+1)} - x^{(k)})}_{\& s^{(k)}} = \underbrace{g^{(k+1)} - g^{(k)}}_{y^{(k)}}$$

$$\Rightarrow \boxed{H^{(k+1)} s^{(k)} = y^{(k)}} \quad \text{secant condition}$$

It is positive definite if $(s^{(k)})^T y^{(k)} > 0$ curvature condition

(related to convexity of f; can be enforced $x^{(k+1)}$ Wolfe conditions)

To maintain symmetry & positive definitness, we choose

$$H^{(k+1)} = H^{(k)} + \alpha\, uu^T + \beta\, vv^T \quad , \quad \alpha, \beta > 0$$

(uu^T: matrix)

* Note that: $w^T(uu^T)w = \langle u, w\rangle^2 > 0$ if $w \neq 0$ and $u \not\perp w$

Choose: $u = y^{(k)}$ and $v = H^{(k)} s^{(k)}$

$$H^{(k+1)} s^{(k)} = y^{(k)} = H^{(k)} s^{(k)} + \alpha\, y^{(k)} \underbrace{(y^{(k)})^T s^{(k)}}_{} + \beta\, H^{(k)} s^{(k)} \underbrace{(s^{(k)})^T (H^{(k)})^T s^{(k)}}_{} \Rightarrow$$

$$\Rightarrow y^{(k)}\left(1 - \alpha \underbrace{(y^{(k)})^T \cdot s^{(k)}}_{>0 \text{ (curvature cond.)}}\right) = H^{(k)} s^{(k)} \left(1 + \beta \underbrace{(s^{(k)})^T (H^{(k)})^T s^{(k)}}_{>0 \text{ by induction, } H^{(k)} \text{ is pos. def.}}\right)$$

($y^{(k)}$: same vectors.)

$$\Rightarrow \alpha = \frac{1}{(y^k)^T \cdot s^{(k)}} \quad , \quad \beta = -\frac{1}{(s^{(k)})^T (H^{(k)})^T s^{(k)}}$$

Conclusion:

$$H^{(k+1)} = H^{(k)} + \frac{y^{(k)} (y^{(k)})^T}{(y^k)^T \cdot s^{(k)}} - \frac{H^{(k)} s^{(k)} (s^{(k)})^T (H^{(k)})^T}{(s^{(k)})^T (H^{(k)})^T s^{(k)}}$$

$$H^{(1)} = \mathrm{id}$$

Exercise: Do 1 step in Newton method for

$$f(x) = f(x_1, x_2) = (x_1+1)^2 + (x_2+3)^2 + 4$$

starting at $(0,0)$.

Stochastic Gradient Descent

In steepest descent,

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \underbrace{\nabla f(x^{(k)})}_{\text{we want to change only this!}}, \quad f: \mathbb{R}^n \to \mathbb{R}$$

What optimization problems we are interested in?

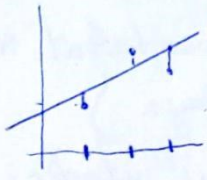
normalization $\min_x \left(\frac{1}{n}\right) \sum_{i=1}^{n} f_i(x)$ (finite sum problems)

(1) Why so? $(x_1, y_1) \dots (x_m, y_m) \in \mathbb{R}^n \times Y$ training data.

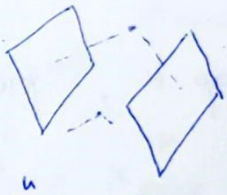
Both n and m are large!

- think of images
- think of users in social networks.
- Y: • Yes/No Classification • Numbers Regression

* $\frac{1}{n} \|Ax - b\|_2^2 = \frac{1}{n} \sum_{i=1}^{n} (a_i^T x - b_i)^2 = \frac{1}{n} \sum f_i(x)$ Least-squares



or minimize distance to hyperplanes



* $\frac{1}{n} \|Ax - b\|_2^2 + \lambda \|x\|_1 = \frac{1}{n} \sum_{i=1}^{n} (a_i^T x - b_i)^2 + \lambda \sum_{j=1}^{d} |x_j|$

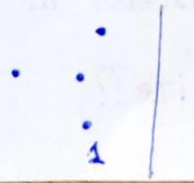
Lasso ℓ_1- least squares (we will see in constraint optimization)

least absolute shrinkage and selection operator

coming from constrained optimization

* $\frac{1}{2} \|x\|_2^2 + \frac{c}{n} \sum_{i=1}^{n} \max\{0, 1 - y_i (x^T a_i - b)\}$ SVM

$\{-1, 1\}$



* Deep Neural Networks

$$\frac{1}{n}\sum_{i=1}^{n} \text{loss}\left(y_i, \overbrace{DNN(x;a_i)}^{\text{prediction}}\right) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

loss: same function; y_i: labels; a_i: training data; x: weights; f_i: Capture N.N. architecture

training

* Maximum likelyhood estimations, etc...

$$\rightsquigarrow \boxed{\min_x \underbrace{\frac{1}{n}\sum_{i=1}^{n} f_i(x)}_{f}}$$

If we use fast gradient descent, then

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \nabla f(x^{(k)}) =$$

$$= x^{(k)} - \alpha^{(k)} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^{(k)})}_{\text{Huge sum! Computing one step is long}}$$

$\rightsquigarrow$ take only 1 component $\rightsquigarrow$

$\rightsquigarrow$ stochastic gradient descent. (the most important method nowadays)

(2.) What if at each step k, pick randomly an integer $i(k) \in \{1, 2, \dots, n\}$?

And perform the update

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \underbrace{\nabla f_{i(k)}(x^{(k)})}_{\text{STOCHASTIC GRADIENT}}$$

n times faster than using ∇f

Convergence, if $\sum_{k=1}^{\infty} \alpha^{(k)} = \infty$ and $\sum (\alpha^{(k)})^2 < \infty$

$\alpha^{(k)} \sim \frac{c}{k}$

[Robbins, Monro, 1951] ~ 12250 citations (1 citation every 2 days)

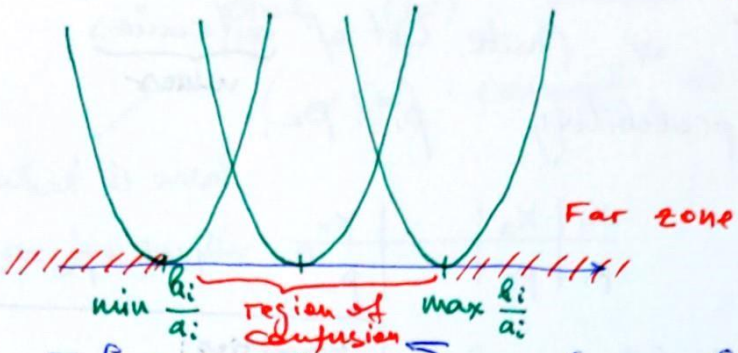
DEMO: $(1-x_1)^2 + 100(x_2 - x_1^2)^2$ It is not descent as such!

• Best step size??

(3.) Works really well at the beginning, and then start fluctuating?

1D intuition:

$$\min \quad f(x) = \frac{1}{2} \sum_{i=1}^{n} (a_i x - b_i)^2 , \qquad x \in \mathbb{R}, \quad a_i \in \mathbb{R}$$



$$\nabla f = 0 \Rightarrow \sum a_i (a_i x - b_i) = 0 \Rightarrow$$

$$\Rightarrow x^* = \frac{\sum_{i=1}^{n} a_i b_i}{\sum_{i=1}^{n} a_i^2}$$

For components, the minimum of $f_i(x) = \frac{1}{2}(a_i x - b_i)^2$ is $x_i^* = \frac{b_i}{a_i}$

Now let $R = [\underbrace{\min_i x_i^*}_{x^*_{min}}, \underbrace{\max_i x_i^*}_{x^*_{max}}]$. Therefore,

$\forall i$, $x^*_{min} \cdot a_i \cdot b_i \leq x^*_{max} \cdot a_i \Rightarrow$

$\Rightarrow x^* \leq x^*_{max} \frac{\sum a_i^2}{\sum a_i^2} = x^*_{max}$. Similarly for x_{min}.

Therefore, $\boxed{x^* \in R}$

- If x lies outside R, then

$$\nabla f_i(x) = a_i (a_i x - b_i)$$

$$\nabla f(x) = \sum_{i=1}^{n} a_i (a_i x - b_i)$$

have the same signs!

$\Rightarrow$ using S.G. $\nabla f_i(x)$ instead of $\nabla f(x)$ also gives progress!

Outside we have a component in the direction of true gradient!
Quickly do reasonable progress is sometimes only what is needed!

(4.) Stochastic gradient descent uses 'stochastic gradients' $g(x)$ such that

$$x^{(k+1)} = x^{(k)} - \alpha g^{(k)} \qquad g = g^{(k)}$$

$$\mathbb{E}[g(x)] = \nabla f(x)$$

↑ expectation (mean, average)

- Unbiased estimate of true gradient

* Digression:

For a random variable X (function) w/ finite list of outcomes (values) $x_1 \dots x_k$ that occur w/ probability $p_1 \dots p_k$,

$$\mathbb{E}[X] = x_1 p_1 + \dots + x_k p_k$$

x_1	x_2	...	x_k
p_1	p_2		p_k

, $p_1 + \dots + p_k = 1$

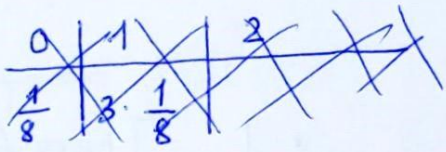
Properties:
$\mathbb{E}[\text{const}] = \text{const}$
$\mathbb{E}[aX + c'X'] = c\mathbb{E}[X] + c'\mathbb{E}[X]$
$X \leq Y \Rightarrow \mathbb{E}[X] \leq \mathbb{E}[Y]$

X = ~~we see a~~ # of dots.

$$\mathbb{E}[X] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + \dots + 6 \cdot \frac{1}{6} = 3.5$$

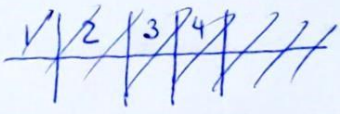
~~If we pick~~ ÷

3 ~~tosses~~ ~~X = ## of heads in row (conseq. heads)~~



If € ... you win 10 €
If : you lose 5 €
The rest: loose 1 €

X = Expected ~~on~~ profit?



10	−5	−1
1/6	3/6	3/6

$$\mathbb{E}[X] = 10 \cdot \frac{1}{6} - \frac{10}{6} - \frac{3}{6} = -\frac{1}{2}$$

For us, we choose $i(k)$ uniformly at random

$$\Leftrightarrow \quad P(i(k) = s) = \frac{1}{n} \Rightarrow$$ ← honest average

$$\mathbb{E}\left[\nabla f_{i(k)}(x)\right] = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x) = \nabla f(x).$$

Therefore, $\nabla f_{i(k)}(x)$ is a stochastic gradient.

If $\nabla f_{i(k)}$ is bounded, then $\sigma \leq C \cdot \text{bound}$

- We also need to care about variance

$$\text{Var}(X) = \mathbb{E}\left[(X - \mathbb{E}(X))^2\right]$$

exercise

(examples above)

$$\text{Var}\left[\nabla f_{i(k)}(x)\right] = \frac{1}{n} \sum (\nabla f_i - \nabla f)^2(x)$$

Should be small

(5.) Practice and theory: which of the two versions to use?

- pick $i(k)$ uniformly at random
  ($i(k)$ is independant of $i(k-1), i(k-2), \ldots$)
  ← theory
- pick $i(k)$ without replacement
  ($i(k)$ cannot be equal to $i(k-1), i(k-2), \ldots$)
  ← what is used

(k is smaller n)

Mini-batch idea:

$$x^{(k+1)} = x^{(k)} - \frac{\alpha^{(k)}}{|I_k|} \underbrace{\sum_{j \in I_k} \nabla f_j(x^{(k)})}_{\text{mini batch}}$$

→ decreases ~~induces~~ variance.
→ how to select them?
s.g.

Ex.: Suppose $|I_k| = 2$, smaller
~~show~~ show that variance is ↑ compute variance.
Assume that we pick a mini batch uniformly at random.

~~Pro~~

(6.) SGD w/ momentum:

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} g(x^{(k)}) + \beta \left(x^{(k)} - x^{(k-1)} \right)$$

↑ s.g.
↑ for stabilization.

(7.) Why does SGD work?
Let us start w/ regular gradient descent.

(a) Assume $|u^T Hf(x)\, u| \le L \|u\|^2 \quad \forall x$

By Taylor, $\exists\, \xi_k$

$$f(x^{(k+1)}) = f\left(x^{(k)} - \alpha \nabla f(x^{(k)})\right) =$$

$$= f(x^{(k)}) - \alpha \nabla f(x^{(k)})^T \cdot \nabla f(x^{(k)}) + \frac{1}{2}\left(\alpha \nabla f(x^{(k)})\right)^T \nabla^2 f(\xi_k) \cdot \left(\alpha \nabla f(x^{(k)})\right)$$

$$\le f(x^{(k)}) - \alpha \|\nabla f(x^{(k)})\|^2 + \frac{\alpha^2 L}{2} \|\nabla f(x^{(k)})\|^2 =$$

$$= f(x^{(k)}) - \alpha\left(1 - \frac{\alpha L}{2}\right) \|\nabla f(x^{(k)})\|^2 \quad \left.\right\} \text{Descent Lemma!}$$

If α small enough (e.g., $\alpha \le \frac{1}{L}$), then

$$f(x^{(k+1)}) \le f(x^{(k)}) - \frac{1}{2}\alpha \|\nabla f(x^{(k)})\|^2 \quad ! \quad (\#)$$

($f(x^{(k+1)})$ decreases.)

Note further,

$$\frac{1}{2}\alpha \sum_{k=0}^{T-1} \|\nabla f(x^{(k)})\|^2 \le \sum_{k=0}^{T-1} f(x^{(k)}) - f(x^{(k+1)}) =$$

(T iterations)

$$= f(x^{(0)}) - f(x^{(T)}) \le f(x^{(0)}) - f^* \quad (f^* = f(x^*),\ x^* \text{ solution})$$

$$\Rightarrow \quad \min \|\nabla f(x^{(k)})\|^2 \le \frac{1}{T} \sum_{k=0}^{T-1} \|\nabla f(x^{(k)})\|^2 \le$$

$$\le \frac{2\left(f(x^{(0)}) - f^*\right)}{\alpha T} \xrightarrow{T \to \infty} 0$$

$\Rightarrow$ we converge.

(b.) Let us now look at SGD.

We assume

$\|\nabla_i f_i(x)\| \le G$ (1^{st} der. bound)

$\|u^T \cdot \nabla^2 f(x) \cdot u\| \le L \cdot \|u\|^2$ (2^{nd} derivative bound)

$$f(x^{(k+1)}) \leq f(x^{(k)}) - \alpha^{(k)} \left(\nabla f_{i_k}(x^{(k)})\right)^T \cdot \nabla f(x^{(k)}) + \frac{(\alpha^{(k)})^2 L}{2} \cdot \underbrace{\|\nabla f_{i_k}(x^{(k)})\|^2}_{\text{can be of any sign!}}$$

Use bounded variance here!

$\longrightarrow$ Take Expectation:

$$\mathbb{E}[f(x^{(k+1)})] \leq \mathbb{E}[f(x^{(k)})] - \alpha^{(k)} \mathbb{E}[\nabla f_{i_k}(x^{(k)})^T \cdot \nabla f(x^{(k)})] + \frac{(\alpha^{(k)})^2 G^2 L}{2}$$

tricky!

$\Rightarrow \mathbb{E}[\nabla f_{i_k}(x^{(k)})] = \nabla f(x^{(k)})$, and hence

$$\mathbb{E}[f(x^{(k+1)})] \leq \mathbb{E}[f(x^{(k)})] - \alpha^{(k)} \mathbb{E}[\|\nabla f(x^{(k)})\|^2] + \frac{(\alpha^{(k)})^2 G^2 L}{2}$$

(Compare to (#))

$$\Rightarrow \sum_{k=0}^{T} \alpha^{(k)} \mathbb{E}[\|\nabla f(x^{(k)})\|^2] \leq f(x^{(0)}) - \mathbb{E}[f(x^{(T)})] + \frac{G^2 L}{2} \sum_{k=0}^{T-1} (\alpha^{(k)})^2$$

$$\leq f(x^{(0)}) - f^* + \frac{G^2 L}{2} \sum_{k=0}^{T-1} (\alpha^{(k)})^2 .$$

$\left(f(x^{(T)}) \geq f^* \Rightarrow \mathbb{E}[f(x^{(T)})] \geq f^* \right)$

How to make sense of this?

Run SGD for a random number of iterations τ with $P(\tau = t) = \frac{\alpha^{(t)}}{\sum_{k=0}^{T-1} \alpha^{(k)}}$ [New discrete random variable]

$$\Rightarrow \mathbb{E}[\|\nabla f(x^{(\tau)})\|^2] = \sum_{k=0}^{T-1} \mathbb{E}\|\nabla f(x^{(k)})\| \cdot P(\tau = k) =$$

$$= \frac{1}{\sum_{k=0}^{T-1} \alpha^{(k)}} \sum_{k=0}^{T-1} \alpha^{(k)} \cdot \mathbb{E}\|\nabla f(x^{(k)})\| \leq \left(\sum_{k=0}^{T-1} \alpha^{(k)}\right)^{-1} \cdot \left(f(x^{(0)}) - f^* + \frac{G^2 L}{2} \sum_{k=0}^{T-1} (\alpha^{(k)})^2\right)$$

~~Therefore better estimates for convex objectives.~~

⊙ If the learning rate is constant, then

$\alpha^{(k)} = \alpha \Rightarrow$

$$\Rightarrow \mathbb{E}\left[\|\nabla f(x^{(T)})\|^2\right] \leq \frac{f(x^{(0)}) - f^*}{\alpha T} + \underbrace{\frac{\alpha L G^2}{2}}$$

$\frac{1}{\alpha T} \to 0$

doesn't go to zero!

'noise ball' term, 'uncertainty range'

This is what we observed.

⊙ If $\sum_{k=0}^{T-1} \alpha^{(k)}$ grows faster than $\sum_{k=0}^{T-1} (\alpha^{(k)})^2$, then

For example $\mathbb{E}\left[\|\nabla f(x^{(T)})\|^2\right] \xrightarrow[T\to\infty]{} 0$.

Example: $\alpha^{(k)} = \frac{1}{\sqrt{k+1}}$

$$\sum_{k=0}^{T-1} \alpha^{(k)} = \sum_{k=0}^{T-1} \frac{1}{\sqrt{k+1}} \sim \int_0^T \frac{1}{\sqrt{x}}\,dx = 2\sqrt{T}$$

$$\sum (\alpha^{(k)})^2 = \sum_{k=0}^{T-1} \frac{1}{k+1} \sim \int_1^T \frac{1}{x}\,dx = \log T$$

$$\Rightarrow \boxed{\mathbb{E}\left[\|\nabla f(x^{(T)})\|^2\right] \lesssim O\left(\frac{1}{\sqrt{T}}\right)}$$

up to approx. of integrals

→ There are better estimates for convex objectives.

Why

$$\mathbb{E}\left[\nabla f_{i(k)}\left(x^{(k)}\right)\right] = \nabla f\left(x^{(k)}\right) \; ?$$

$$\mathbb{E}[X \mid Y=y] = \sum x \cdot \frac{P(X=x, Y=y)}{P(Y=y)}.$$

(X: random var.)

If X and Y are independant, then $P(X=x, Y=y) =$

$= P(X=x) \cdot P(Y=y) \Rightarrow \mathbb{E}[X \mid Y=y] = \mathbb{E}[X]$

Let $s_1 \dots s_{k-1}$ is the sequence of indices s.t.

$$x^{(k)} = x^{(k-1)} + \alpha^{(k-1)} \nabla f_{s_{k-1}}\left(x^{(k-1)}\right)$$

$$x^{(k-1)} = x^{(k-2)} + \alpha^{(k-2)} \nabla f_{s_{k-2}}\left(x^{(k-2)}\right)$$

$$\vdots$$

$$x^{(1)} = x^{(0)} + \alpha^{(0)} \nabla f_{s_0}\left(x^{(0)}\right)$$

These are independ.

$$P\left(i(k)=s, \left(i(k-1), i(k-2), \dots i(0)\right) = (s_{k-1}, s_{k-2}, \dots, s_0)\right)$$

$$P\left(\left(i(k-1) \dots i(0)\right) = (s_{k-1} \dots s_0)\right)$$

$$\Rightarrow \mathbb{E}\left[\nabla f_{i(k)}\left(x^{(k)}\right)\right] = \sum_{s=1}^{n} \underbrace{\nabla f_s\left(x^{(k)}\right)}_{\text{value}} \cdot$$

$$= \sum_{s=1}^{n} \nabla f_s\left(x^{(k)}\right) \cdot P\left(i(k)=s\right) =$$

$$= \frac{1}{n} \sum_{s=1}^{n} \nabla f_s\left(x^{(k)}\right) = \nabla f\left(x^{(k)}\right).$$

Let us write $\mathbb{E}_i[\nabla f_{i(k)}(x^{(k)})]$ for expectation.

Assume $\mathbb{E}[\|\nabla f_{i(k)}(x) - \nabla f(x)\|^2] \le \sigma^2$

$\Leftrightarrow \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x) - \nabla f(x)\|^2 \le \sigma^2$

$\|\nabla f_{i(k)}(x)\|^2 - 2\langle \nabla f_{i(k)}(x), \nabla f(x)\rangle + \|\nabla f(x)\|^2$

Variance is bounded. $\Rightarrow \mathbb{E}[\|\nabla f_{i(k)}(x)\|^2] \le \|\nabla f(x)\|^2 + \sigma^2$

Similarly, we get:

$$f(x^{(k+1)}) \le f(x^{(k)}) - \alpha^{(k)}\langle \nabla f_{i(k)}(x^{(k)}), \nabla f(x^{(k)})\rangle + \frac{(\alpha^{(k)})^2 L}{2}\|\nabla f_{i(k)}(x^{(k)})\|^2.$$

$$\Rightarrow \mathbb{E}[f(x^{(k+1)}) \mid x^{(k)}] \le \mathbb{E}[f(x^{(k)}) \mid x^{(k)}] \dots$$

$$= f(x^{(k)}) - \alpha^{(k)}\Big\langle \overbrace{\mathbb{E}[\nabla f_{i(k)}(x^{(k)}) \mid x^{(k)}]}^{\nabla f(x^{(k)}) \text{ as above}}, \nabla f(x^{(k)})\Big\rangle + \frac{(\alpha^{(k)})^2 L}{2}\mathbb{E}[\|\nabla f_{i(k)}(x^{(k)})\|^2 \mid x^{(k)}]$$

$$= f(x^{(k)}) - \alpha^{(k)}\|\nabla f(x^{(k)})\|^2 + \frac{(\alpha^{(k)})^2 L}{2} \dots$$

$$\Rightarrow \mathbb{E}[f(x^{(k+1)}) \mid x^{(k)}] \le f(x^{(k)}) - \frac{\alpha^{(k)}}{2}\left(1 - \frac{\alpha^{(k)} L}{2}\right)\|\nabla f(x^{(k)})\|^2 + \frac{(\alpha^{(k)})^2 \sigma^2 L}{2}$$

($\|\nabla f(x^{(k)})\|^2$: expectation)

$\left(\alpha^{(k)} \le \frac{1}{L}\right)$

$$\Rightarrow \mathbb{E}[f(x^{(k+1)})] \le \mathbb{E}[f(x^{(k)})] - \frac{\alpha^{(k)}}{2}\mathbb{E}[\|\nabla f(x^{(k)})\|^2] + \frac{(\alpha^{(k)})^2 \sigma^2 L}{2}.$$

(This is more parallel to GD method.)