Regret and SGD

- at each step, we see a new objective function
- online optimization

at time step $t$, alg sees $f_t : K \to \mathbb{R}$ and proposes $x_t$

- how to score this?

$$\text{regret} = \sum_t f_t(x_t) - \min_{x^* \in K} \sum_t f_t(x^*)$$

- time average of cost functions at chosen points

you see all alg fn's in advance - find smallest average.
The reason this is helpful is we can prove bounds, which lead to new algs.

Take g.d. and substitute

\[ x_{t+1} \leftarrow x_t - \eta \nabla f_t(x_t) \]

\[ \text{use function } \Phi_t \]

From proof of G-2, we have

\[ f_t(x_t) + \Phi_{t+1} - \Phi_t \leq f_t(x^*) + \frac{1}{2} \eta G^2 \]

(cause true for any convex).

Show thus

\[ \sum_{t=1}^{T} (f_t(x_t) - f_t(x^*)) \leq \sum_{t=1}^{T} (\Phi_t - \Phi_{t+1}) + \frac{1}{2} \eta G^2 \]

\[ \leq \Phi_1 + \frac{1}{2} \eta TG^2 \]
how plug in $T \geq \frac{\|x_t-x^*\|^2G^2}{\epsilon^2}$

\[ \eta = \frac{\|x_t-x^*\|}{G \sqrt{T}} \]

now do algebra to get:

\[ \frac{1}{T} \sum_{t=0}^{T} (f_t(x_t) - f_t(x^*)) \leq \frac{\|x_t-x^*\|G}{\sqrt{T}} \leq \epsilon \]

\[ \uparrow \]

\[ \text{regret.} \]
But pure SGD is not particularly well adapted to some cases.

For regret bound, we assumed nothing about $f_t(x)$. (Except convexity)

Now imagine we know something.

E.g.

$$f_t(x) = \frac{1}{2} [a_t^T x]^2$$

where $a_t$ is very sparse.

i'th component of $a_t$ has prob of non-zero $P_i$, iid

and (aren't we lucky!) $P$ is sorted $P_1 \gg P_2 \cdots \gg P_d$. 
Think about the sequence of gradients $\{\nabla f_t\}$:

- Few zeros
- About $(1 - \beta_t)$, many zeros

In this case, we face a serious problem:

For some components, we take a large step very occasionally. Connecting this step would be hard (slow) because gradient would be 0 at most. We would like to reduce the step using some estimate of sparsity.
Example

\[ x^i = \begin{bmatrix} x_i \\ \vdots \\ x_d \end{bmatrix} \]

\[ f_t(x) = \sum_i \delta_{i,t} x_i^2 \]

where \( \delta_{i,t} = \begin{cases} 1 \text{ with prob } p_i^- \\ 0 \text{ (1-} p_i^-) \end{cases} \)

and IID over \( t \)

Now think about component \( i \)?

\[ x_i^{(w+1)} = x_i^{(w)} - \eta g_i^{(w)} \]

\[ = \begin{cases} x_i^{(w)} \text{ with prob } p_i^- \\ 0 \text{ otherwise} \end{cases} \]

so \( x^{(w+1)} \approx (1-\eta) x^{(w)} \)

---

essentially, you see the gradient about \( p^{(w+1)} \) time 5
but this converges rather slowly

**IDEA:**

\[
x_i^{(w+1)} = (\text{shrink}) \left[ x_i^{(w)} - \eta g_i^{(w)} \right]
\]

where shrink tiers the frequency with which \(g_i^{(w)} \neq 0\)

**ADAGRAM:**

Define \(G_w = \sum_{t=1}^{w} g(t) g(t)^T\) outer product matrix of past gradients

Consider \(\text{diag } G_w = \Delta_w\)

ith diagonal component is approx \(E[g_i^2]\)
update
\[
\mathbf{x}^{(w+1)} = \left[ \mathbf{\Delta}^{(w)} \right]^{1/2} \left[ \mathbf{x}^{(w)} - \eta \left[ \mathbf{\Delta}^{(w)} \right]^{(-1/2)} \mathbf{g}^{(w)} \right]
\]

notice you could have a divide by zero issue here — actually use
\[
\left[ \mathbf{\Delta}^{(w)} + \eta I \right]^{(-1/2)}
\]

Fact: (Duchi et al, p2125)

let \( x^{(w)} \) be generated as above

and assume
\[
\max_t \| \mathbf{x}^* - \mathbf{x}_t \|_\infty \leq D_\infty
\]

use \( \eta = \frac{D_\infty}{\sqrt{2}} \)

then for \( \mathbf{x}^* \) we get.
\[ R(T) \leq \sqrt{2} D_\infty \sum_{i=1}^{d} \|g_{i:T},i\|_2 \]

Regret at \( T \)

Length of \( i \)-th component of gradient sequence (Note: Not squared length)

This is better than previous bound when:

- \( D_\infty \) is reasonable (e.g., we're in a box)
- Gradients have nice sparsity properties

More examples: Duchi et al 20125
Alternative view

\( f(x) \) is a "noisy" function rather than an arbitrary seq of convex functions (avoid defining, as do authors!)

. Consider steps.

. In each component, we would like the step to be bounded by about the same amount.

. Consider

\[
\alpha \left[ \frac{\text{mean (gradient)}}{\sqrt{\text{mean (gradient}^2)}} \right]
\]

\( \Rightarrow \) this will tend to be about 1 or -1
ISSUE

estimate means?

these estimates should be mean(gradient_t) over all possible f_t

But we can't compute that.

instead, compute moving average

\[ v_0 = 0 \]
\[ v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t \]

\[ \uparrow \text{ element wise} \]

this incorporates past gradients into est of mean(gradient_t^2)
But as an est of $\text{mean}(g_t^2)$, it is biased (Welling et al., p3).

Use

$$\frac{V_t}{(1 - \beta_2^t)}$$

(same arg for $\text{mean}(g_t)$)

We now have ADAM

Experience shows

- Fast convergence to slightly worse solutions
- SGD with good step lengths gets better solutions but much slower

Why?
Algorithm 2: AdaMax, a variant of Adam based on the infinity norm. See section 7.1 for details.
Good default settings for the tested machine learning problems are $\alpha = 0.002$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. With $\beta_1^t$ we denote $\beta_1$ to the power $t$. Here, $(\alpha/(1 - \beta_1^t))$ is the learning rate with the bias-correction term for the first moment. All operations on vectors are element-wise.

Require: $\alpha$: Step size
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates
Require: $f(\theta)$: Stochastic objective function with parameters $\theta$
Require: $\theta_0$: Initial parameter vector

$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$u_0 \leftarrow 0$ (Initialize the exponentially weighted infinity norm)
$t \leftarrow 0$ (Initialize timestep)

while $\theta_t$ not converged do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep $t$)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $u_t \leftarrow \max(\beta_2 \cdot u_{t-1}, |g_t|)$ (Update the exponentially weighted infinity norm)
    $\theta_t \leftarrow \theta_{t-1} - (\alpha/(1 - \beta_1^t)) \cdot m_t/u_t$ (Update parameters)
end while

return $\theta_t$ (Resulting parameters)