Numerical Optimization for Machine Learning

SGD with Constant Step Sizes, Growing Batches, and Over-Parameterization

Mark Schmidt

University of British Columbia

Summer 2022

Last Time: Convergence of Stochastci Gradient Descent

- We considered stochastic gradient descent (SGD),

$$w^{k+1} = w^k - \alpha_k \nabla f_{i_k}(w^k).$$

  which performs a gradient descent step using a random training example i_k.
  - This gives an unbiased gradient approximation, $\mathbb{E}[\nabla f_{i_k}(w^k)] = \nabla f(w^k)$.

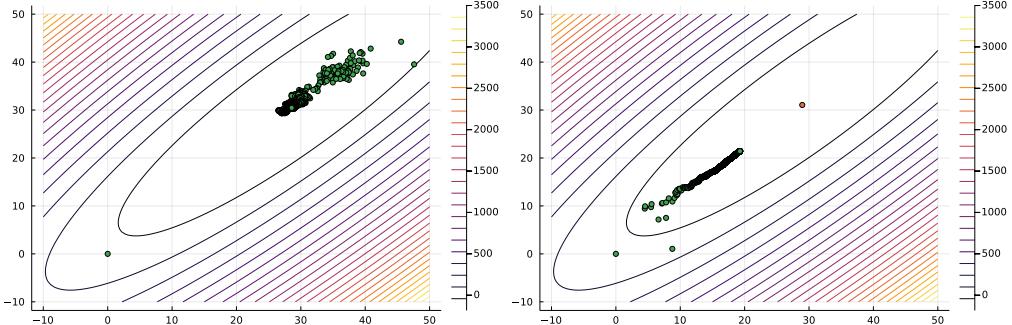
- If we assume $\mathbb{E}[\|\nabla f_{i_k}(w^k)\|^2] \leq \sigma^2$ then we can show

$$\min_{k=0,1,\ldots,t-1} \{\mathbb{E}\|\nabla f(w^k)\|^2\} \leq \frac{f(w^0) - f^*}{\sum_{k=0}^{t-1} \alpha_k} + \frac{L\sigma^2}{2} \frac{\sum_{k=0}^{t-1} \alpha_k^2}{\sum_{k=0}^{t-1} \alpha_k},$$

  where first term is like gradient descent bound and second term is effect of noise.
- Converge depends on value of $\sum_k \alpha_k^2 / \sum_k \alpha_k$.
  - $\alpha_k = \gamma/k$ converges at extremely slow $O(1/\log(k))$.
  - $\alpha_k = \gamma/\sqrt{k}$ converges at faster $\tilde{O}(1/\sqrt{k})$.
  - $\alpha_k = \gamma$ converges at faster $O(1/k)$ but only to solution accuracy $O(\gamma\sigma^2)$.

SGD with Decreasing Step Sizes

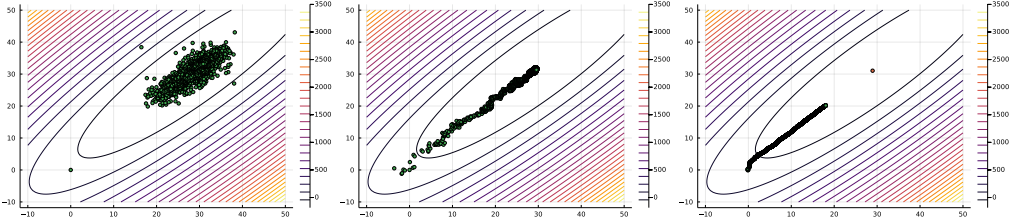
- 10000 SGD iterations with $\alpha_k = 1/\mu k$ and $\alpha_k = 1/10\mu_k$:



- This step size works well in limited situations but is not robust:
  - For strongly-convex problems, we will discuss how $\alpha_k = 1/\mu k$ has $O(1/k)$ rate.
  - But using $1/10\mu k$ leads to extremely slow convegence.
  - And using $10/\mu k$ the method explodes (no iterations would fix on plot).

SGD with Robust Decreasing Step Sizes

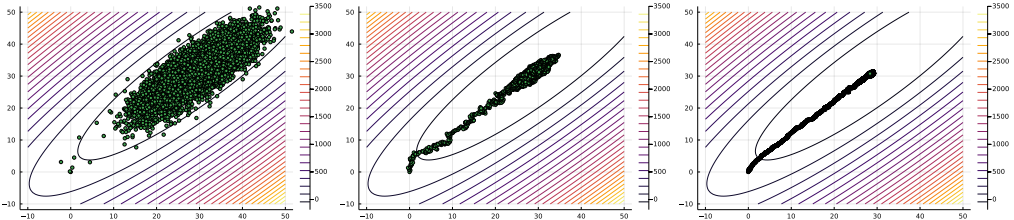
- 10000 SGD iterations with $\alpha_k = 10/L\sqrt{k}$, $\alpha_k = 1/L\sqrt{k}$, and $\alpha_k = 1/10L\sqrt{k}$:



- Step sizes proportional to square root of k are more robust
  - Works well for a range of constants, even though "best case" rate is slower.

SGD with Constant Step Sizes

- 10000 SGD iterations with $\alpha_k = 1/L$, $\alpha_k = 1/10L$, and $\alpha_k = 1/100L$:



- Constant step sizes converge quickly to neighbourhood of solution.
  - Then behave erratically within neighbourhood and do not converge to solution.

SGD as Gradient Descent with Random Error

- We can write the SGD step as a deterministic gradient descent step with error,

$$w^{k+1} = w^k - \alpha_k(\nabla f(w^k) + e^k),$$

where for SGD the $e^k = \nabla f_i(w^k) - \nabla f(w^k)$ is random.

- Since SGD is unbiased, for SGD the mean of e^k is 0:

$$\mathbb{E}[e^k] = \mathbb{E}[\nabla f_i(w^k)] - \nabla f(w^k) = 0.$$

- Progress for gradient descent with error is affect by $\|e^k\|^2$.
  - To guarantee progress, we usually want $\|e^k\|^2 \leq \|\nabla f(w^k)\|^2$.

- For SGD, expected value $\|e^k\|^2$ is a measure of the variation in the gradients,

$$\mathbb{E}[\|e^k\|^2] = \mathbb{E}[\|\nabla f_i(w^k) - \nabla f(w^k)\|^2].$$

Convergence of SGD with More-Realistic Noise Bound

- The assumption that $\mathbb{E}[\|\nabla f_{i_k}(w^k)\|^2] \leq \sigma^2$ is strong.
  - Implies gradients bounded, and cannot hold globally for PL functions.

- We can instead assume variation in gradients is bounded,

$$\mathbb{E}[\|e^k\|^2] \leq \sigma^2,$$

  which leads to a similar bound under the descent lemma (see bonus slide).
- Following similar analysis under this assumption (and $\alpha_k < 2/L$) gives

$$\min_{k=0,1,\ldots,t-1} \{\mathbb{E}\|\nabla f(w^k)\|^2\} \leq \frac{f(w^0) - f^*}{\sum_{k=0}^{t-1} \alpha_k(1 - \frac{\alpha_k L}{2})} + \frac{L\sigma^2}{2} \frac{\sum_{k=0}^{t-1} \alpha_k^2}{\sum_{k=0}^{t-1} \alpha_k(1 - \frac{\alpha_k L}{2})}.$$

- This leads to the similar conclusions regarding choosing the step size.

SGD with Random Permutations

- In practice, SGD is often implemented with random permutations.
  - A common variation is switching between 2 random permutations.
    - Yields a predictable/optimizable data access pattern.

- Bottou [2009] conjected that random permutations yields an $O(1/k^2)$ rate.
  - Based on experiments.

- A sequence of papers have worked towards resolving the rate in various settings.
  - For strongly-convex functions, we now have $O(1/nk^2)$ rate after k epochs.
    - Whereas regular SGD would have $O(1/nk)$ after same number of updates.
  - For strongly-convex quadratics, improves to $O\left(\frac{1}{(nk)^2} + \frac{1}{nk^3}\right)$.
    - Above results assume iterates stay bounded, and there are matching lower bounds.

Outline

1. SGD for PL Functions
2. Mini-Batch SGD and Growing Batches
3. SGD and Over-Parameterization
4. Faster Algorithms under Over-Parameterization

Convergence of SGD for PL Functions

- You can get faster rates for SGD if f is strongly-convex or PL:
  - Under these assumptions you can get an $O(1/k)$ rate.
  - Requires a step size of $\alpha_k = O(1/k)$, but constant matters.

- For strongly-convex f, using $\alpha_k = 1/\mu k$ gives the $O(1/k)$ rate.
  - Initial steps are huge, then it slowly converges to solution.
    - Might do worse than slower $O(1/\sqrt{k})$ step sizes after finite steps.
  - And be careful, if you over-estimate μ rate can be much worse.
  - The only problem where I have seen $\alpha_k = O(1/k)$ work effectively is binary SVMs.
    - Where $\alpha_k = 1/\mu k$ is tough to beat.

Convergence Rate of SGD with Constant Step under PL

- We showed that SGD with constant step size has rate $O(1/\alpha k) + O(\alpha\sigma^2)$.
  - For f bounded below, ∇f Lipschitz, and noise bounded by σ^2.
  - Convergence rate of gradient descent.
  - Up to accuracy proportional to step size and noise bound.

- As before, we can derive faster rates under PL: $O(\rho(\alpha)^k) + O(\alpha\sigma^2)$.
  - Linear convergence up to solution level proportional to step size and noise bound.
  - The number of $\rho(\alpha)$ will depend on the precise step-size we choose.

  - We will show this assuming $\alpha < 1/2\mu$ and $\mathbb{E}[\|\nabla f(w)\|^2] \leq \sigma^2$.
  - Bonus slides show this for $\alpha < 2/L$ and weaker $\mathbb{E}[\|e^k\|^2] \leq \sigma^2$.

- Constant step sizes adapt to problem.
  - Do not need to know if f is convex or PL.
  - Do not need to know which variation bound is satisfied.
  - This is more like gradient descent where $\alpha_k = 1/L$ works for many problems.

Convergence Rate of SGD with Constant Step under PL

- To derive the result under PL, we start with our SGD progress bound:

$$\mathbb{E}[f(w^{k+1})] \leq f(w^k) - \alpha_k \underbrace{\|\nabla f(w^k)\|^2}_{\text{good}} + \alpha_k^2 \underbrace{\frac{L}{2}\mathbb{E}[\|\nabla f_{i_k}(w^k)\|^2]}_{\text{bad}}.$$

- Bound with PL ($\|\nabla f(w^k)\|^2 \geq 2\mu(f(w^k) - f^*)$) and variation bound,

$$\mathbb{E}[f(w^{k+1})] \leq f(w^k) - \alpha_k 2\mu(f(w^k) - f^*) + \alpha_k^2 \frac{L\sigma^2}{2}.$$

- Subtract f^* from both sides and factorize,

$$\mathbb{E}[f(w^{k+1})] - f^* \leq (1 - 2\alpha_k\mu)(f(w^k) - f^*) + \alpha_k^2 \frac{L\sigma^2}{2}.$$

Convergence Rate of SGD with Constant Step under PL

- Bound from previous slide, with a constant step size $\alpha_k = \alpha$:

$$\begin{aligned}
\mathbb{E}[f(w^{k+1})] - f^* &\leq (1 - 2\alpha\mu)(f(w^k) - f^*) + \alpha^2 \frac{L\sigma^2}{2} \\
\text{(with tower prop)} &\leq (1 - 2\alpha\mu)\left((1 - 2\alpha\mu)(f(w^{k-1}) - f^*) + \alpha^2 \frac{L\sigma^2}{2}\right) + \alpha^2 \frac{L\sigma^2}{2} \\
&= (1 - 2\alpha\mu)^2(f(w^{k-1}) - f^*) + \alpha^2 \frac{L\sigma^2}{2}(1 + (1 - 2\alpha\mu)).
\end{aligned}$$

- Applying bound recursively from k down to 0 we get

$$\mathbb{E}[f(w^k)] - f^* \leq (1 - 2\alpha\mu)^k(f(w^k) - f^*) + \alpha^2 \frac{L\sigma^2}{2} \sum_{t=0}^{k} (1 - 2\alpha\mu)^t.$$

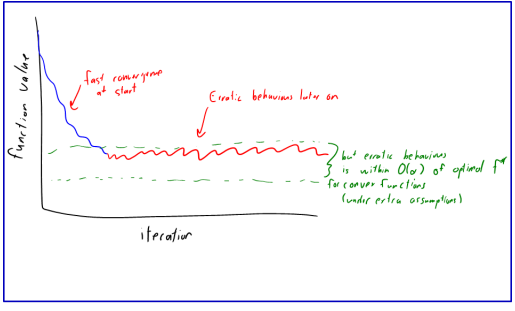
- We have $\sum_{t=0}^{k}(1 - 2\alpha\mu)^t < \sum_{t=0}^{\infty}(1 - 2\alpha\mu)^t = 1/2\alpha\mu$ (geometric series).

SGD with Constant Step Size

- Convergence rate of SGD with constant step size α for PL f:

$$\mathbb{E}[f(w^k) - f^*] \leq (1 - 2\alpha\mu)^k (f(w^0) - f(w^*)) + \alpha\sigma^2 \frac{L}{4\mu}.$$

- First term looks like linear convergence, but second term does not go to zero.



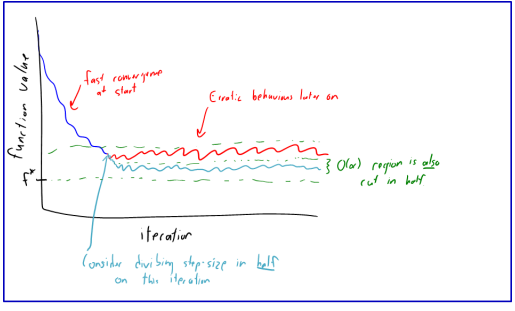
- Theory justifies "divide the step-size in half if it looks like it's stalled" heuristic.
  - Halving α divides bound on distance to f^* in half (similar for non-convex).

SGD with Constant Step Size

- Convergence rate of SGD with constant step size α for PL f:

$$\mathbb{E}[f(w^k) - f^*] \leq (1 - 2\alpha\mu)^k (f(w^0) - f(w^*)) + \alpha\sigma^2 \frac{L}{4\mu}.$$

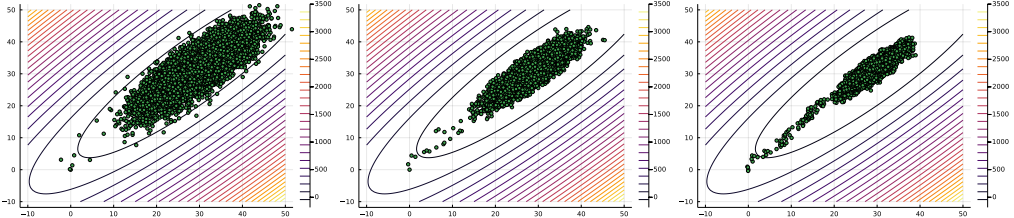
- First term looks like linear convergence, but second term does not go to zero.



- Theory justifies "divide the step-size in half if it looks like it's stalled" heuristic.
  - Halving α divides bound on distance to f^* in half (similar for non-convex).

SGD with Constant Step Sizes

- For strongly-convex, we get the same type of convergence in terms of iterates.
- 10000 SGD iterations with $\alpha_k = 1/L$, $\alpha_k = 1/2L$, and $\alpha_k = 1/4L$:



- Constant step sizes converges linearly to neighbourhood of solution.
  - Then behave erratically within neighbourhood.
  - Size of neighbourhood is propotional to α_k.

Optimization vs. Machine Learning

- Optimization: we want $\nabla f(w^k)$ to converge to 0.
  - So we need to use decreasing step sizes to guarantee continued progress.
    - But as we decrease the step size SGD will converge slower.

- Machine learning: we only need $\nabla f(w^k)$ close to 0.
  - We expect test error to be similar for all w_k "close enough" to stationary point.
    - May only care about 2 decimal places of accuracy (model is not perfect anyways).
    - So do not need 10 decimal places of optimization accuracy.

- For any "closeness", we could use a small-enough constant step size $\alpha_k = \alpha$.
  - Guarantee expected progress when $\nabla f(w_k)$ is large.
  - Adapts to the difficulty of the problem (same for PL, convex, and non-convex).
  - But in areas where gradient is small, SGD can behave erratically.

Early Stopping: A Practical Strategy for Deciding When to Stop

- How do you decide when to stop?
  - In gradient descent, we stop when gradient is close to zero.
- In SGD:
  - Individual gradients do not necessarily go to zero.
  - We cannot see full gradient, so we do not know when to stop.
- Practical trick for machine learning problems:
  - Every k iterations (for some large k), measure validation set error.
  - Stop if the validation set error "is not improving"..
    - We do not check gradient, since it takes a lot longer for gradient to get small.
    - Early stopping can also reduce overfitting (chosen iteration was validated).

Outline

1 SGD for PL Functions

2 Mini-Batch SGD and Growing Batches

3 SGD and Over-Parameterization

4 Faster Algorithms under Over-Parameterization

SGD with Mini-Batches

- Deterministic gradient descent uses all n gradients,

$$\nabla f(w^k) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w^k).$$

- Stochastic gradient descent approximates it with 1 sample,

$$\nabla f(w^k) \approx \nabla f_{i_k}(w^k).$$

- A common variant is to use m samples as a mini-batch $\mathcal{B}^k$,

$$\nabla f(w^k) \approx \frac{1}{m}\sum_{i\in\mathcal{B}^k} \nabla f_i(w^k).$$

- Mini-batches are particularly useful for vectorization/parallelization.
  - For example, with 16 cores set $m = 16$ and compute 16 gradients at once.

Unbiasedness of Mini-Batch Approximation

- Taking expectation over choice of mini-batch gives:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{m}\sum_{i\in\mathcal{B}}\nabla f_i(w)\right] &= \frac{1}{m}\mathbb{E}\left[\sum_{i\in\mathcal{B}}\nabla f_i(w)\right] && \text{(linearity of } \mathbb{E}\text{)} \\
&= \frac{1}{m}\sum_{i\in\mathcal{B}}\mathbb{E}[\nabla f_i(w)] && \text{(linearity of } \mathbb{E}\text{)} \\
&= \frac{1}{m}\sum_{i\in\mathcal{B}}\nabla f(w) && \text{(unbiased estimate)} \\
&= \frac{m}{m}\nabla f(w) && \text{(term is repeated } |\mathcal{B}| \text{ times)} \\
&= \nabla f(w),
\end{aligned}$$

so mini-batch approximation is unbiased.

Variation in Mini-Batch Approximation

- To analyze variation in gradients, we use a variance-like identity:
  - If random variable g is an unbiased approximation of vector μ, then

$$\begin{aligned}
\mathbb{E}[\|g-\mu\|^2] &= \mathbb{E}[\|g\|^2 - 2g^T\mu + \|\mu\|^2] && \text{(expand square)}\\
&= \mathbb{E}[\|g\|^2] - 2\mathbb{E}[g]^T\mu + \|\mu\|^2 && \text{(linearity of } \mathbb{E}\text{)}\\
&= \mathbb{E}[\|g\|^2] - 2\mu^T\mu + \|\mu\|^2 && \text{(unbiased)}\\
&= \mathbb{E}[\|g\|^2] - \|\mu\|^2.
\end{aligned}$$

Variation in Mini-Batch Approximation

- We also need expectation of inner product between independent samples:

$$\begin{aligned}
\mathbb{E}[\nabla f_i(w)^T \nabla f_j(w)] &= \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{n^2} \nabla f_i(w)^T \nabla f_j(w) && \text{(definition of } \mathbb{E}) \\
&= \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w)^T \left(\frac{1}{n}\sum_{j=1}^{n} \nabla f_j(w) \right) && \text{(distributive)} \\
&= \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w)^T \nabla f(w) && \text{(gradient of } f) \\
&= \left(\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) \right)^T \nabla f(w) && \text{(distributive)} \\
&= \nabla f(w)^T \nabla f(w) = \|\nabla f(w)\|^2 && \text{(gradient of } f),
\end{aligned}$$

which is squared gradient norm.

Variation Bound for Mini-Batch Approximation

- Let $g_2(w) = \frac{1}{2}(\nabla f_i(w) + \nabla f_j(w))$ be mini-batch approximation with 2 samples.

$$
\begin{aligned}
\mathbb{E}[\|g_2(w) - \nabla f(w)\|^2] &= \mathbb{E}[\|\frac{1}{2}(\nabla f_i(w) + \nabla f_j(w))\|^2] - \|\nabla f(w)\|^2 && \text{(variance identity)} \\
&= \frac{1}{4}\mathbb{E}[\|\nabla f_i(w)\|^2] + \frac{1}{2}\mathbb{E}[\nabla f_i(w)^T \nabla f_j(w)] + \frac{1}{4}\mathbb{E}[\|\nabla f_j(w)\|^2] - \|\nabla f(w)\|^2 && \text{(expand square)} \\
&= \frac{1}{2}\mathbb{E}[\|\nabla f_i(w)\|^2] + \frac{1}{2}\mathbb{E}[\nabla f_i(w)^T \nabla f_j(w)] - \|\nabla f(w)\|^2 && (\mathbb{E}[\nabla f_i] = \mathbb{E}[\nabla f_j]) \\
&= \frac{1}{2}\mathbb{E}[\|\nabla f_i(w)\|^2] + \frac{1}{2}\|\nabla f(w)\|^2 - \|\nabla f(w)\|^2 && (\mathbb{E}[\nabla f_i \nabla f_j] = \nabla f^2) \\
&= \frac{1}{2}\mathbb{E}[\|\nabla f_i(w)\|^2] - \frac{1}{2}\|\nabla f(w)\|^2 \\
&= \frac{1}{2}\left(\mathbb{E}[\|\nabla f_i(w)\|^2] - \|\nabla f(w)\|^2\right) && \text{(factor } \frac{1}{2}\text{)} \\
&= \frac{1}{2}\mathbb{E}[\|\nabla f_i(w) - \nabla f(w)\|^2] && \text{(variance identity)} \\
&= \frac{\sigma(w)^2}{2} && (\sigma^2 \text{ is 1-sample variation})
\end{aligned}
$$

- So SGD error $\mathbb{E}[\|e^k\|^2]$ is cut in half compared to using 1 sample.

Variance of Mini-Batch Approximation

- With m samples in our mini-batch we have that (see bonus)

$$\mathbb{E}[\|e^k\|^2] = \frac{\sigma(w^k)^2}{m},$$

  where $\sigma^2(w^k)$ is the variation in the individual gradients at w^k.

- "With a mini-batch size of 100, effect of noise is divided by 100".
  - Biggest gains obtained for increasing small batch sizes.

- "With a mini-batch size of 100, you can use a step size that is 100-times larger."
  - "Linear scaling rule" (but may not guarantee progress if $\alpha_k \geq 2/L$)

Batching: Growing the Batch Size

- Consider mini-batch SGD under PL with small constant step size:
  - Converges linearly to a sub-optimality of $O(\alpha\sigma^2/m)$.
    - For $\sigma(w) > \sigma$ for all w.

- You could decrease α_k to get closer to the solution.
  - But this makes SGD converge more slowly.

- Or, you can increase the batch size m.
  - Doubling batch size has same effect as halving the step size.
    - But without needing to use a smaller step size.
  - If you grow the batch size over the iterations, converges with a constant step size.
    - Effect of noise goes to 0 as the batch size increases.
    - Growing batch size methods are sometimes called batching methods.

Variance of Mini-Batch Approximation for Finite Data

- Variance of mini-batch approximation is smaller for finite datasets.

- If we sample without replacement from a set of n examples we have

$$\mathbb{E}[\|e^k\|^2] = \frac{\sigma(w^k)^2}{m}\frac{n-m}{n},$$

  where the extra term is called the finite sample correction.
  - This term is less than 1 so variance is strictly better for "with replacement" sampling.

- Finite sample correction is minor when m is small.
- But as m approaches n the finite-sample correction drives effect of noise to 0.
  - Because you sample a greater portion of the overall dataset.

Stochastic Heavy-Ball/Nesterov/Newton or Line Search?

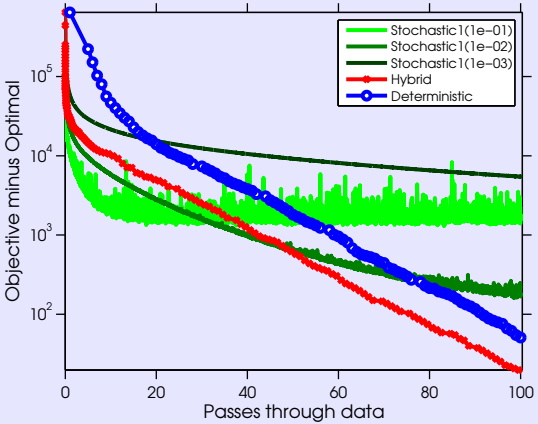
- Should we use heavy-ball/Nesterov/Newton-like stochastic methods?
  - May improve dependency on L and μ.
  - But do not improve dependency on noise σ^2.
    - So do not improve the convergence rate over SGD.
  - These can even amplify the effect of the noise.
    - Need momentum to converge to 0 for SGD+momentum to converge.

- Can get faster rates with growing batch sizes using these techniques.
  - Need to grow batch size so $\mathbb{E}[\|e^k\|^2]$ goes to 0 at the fast rate.
    - If you want linear convergence rate with constant γ, need $\mathbb{E}[\|e^k\|^2] = O(\gamma^k)$.
  - But this increases the iteration cost.

- Similar ideas hold for line-search:
  - With fixed batch size, standard line-search methods do not work.
  - With line-search converges with growing batch sizes.

Comparison of Deterministic, Stochastic, and Hybrid

- For training a conditional random field, below plot compares:
  - Deterministic: quasi-Newton method (L-BFGS) with Wolfe line-search.
  - Stochastic: SGD with the 3 best-performing step sizes (among powers of 10).
  - Hybrid: growing-batch quasi-Newton method (L-BFGS) with Armijo line-search.



Outline

1 SGD for PL Functions

2 Mini-Batch SGD and Growing Batches

3 SGD and Over-Parameterization

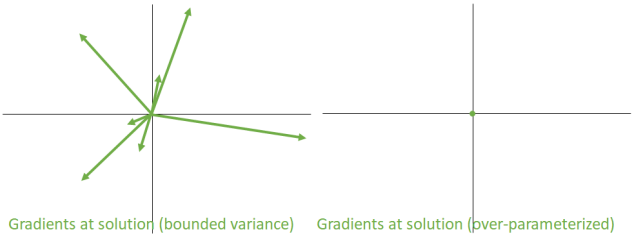
4 Faster Algorithms under Over-Parameterization

Motivation: Over-Parameterized Models in Machine Learning

- Modern machine learning practioners often do a weird thing:
  - Train (and get excellent performance) with models that are over-parameterized.
    - "The model is so complicated that you can fit the data perfectly".
    - The exact setting where we normally teach students that bad overfitting happens.
- Examples:
  - Many state-of-the-art deep computer vision models are over-parameterized.
    - Models powerful enough to fit training set with random labels [Zhang et al., 2017].
  - Linear models with sufficiently expressive features [Liang & Rakhlin, 2018].
- Many recent papers study benefits of over-parameterization in various settings:
  - Algorithms may have implicit regularization that reduces overfitting.
  - Optimizers may find global optima in problems we normally view as hard.
- Over-parameterization significantly changes the behaviour of SGD.

Effect of Over-Parameterization on SGD

- We say a model is over-parameterized if it can exactly fit all training examples.
  - Unlike usual bounded variance assumption, we have $\nabla f_i(w_*) = 0$ for all i:



- For over-parameterized models, the variance is 0 at minimizers.
  - And SGD converges with a sufficiently small constant step size.

Stochastic Convergence Rates under Over-Parameterization

- One way to characterize over-parameterization: strong growth condition (SGC),

$$\mathbb{E}[\|\nabla f_i(w)\|^2] \leq \rho \|\nabla f(w)\|^2,$$

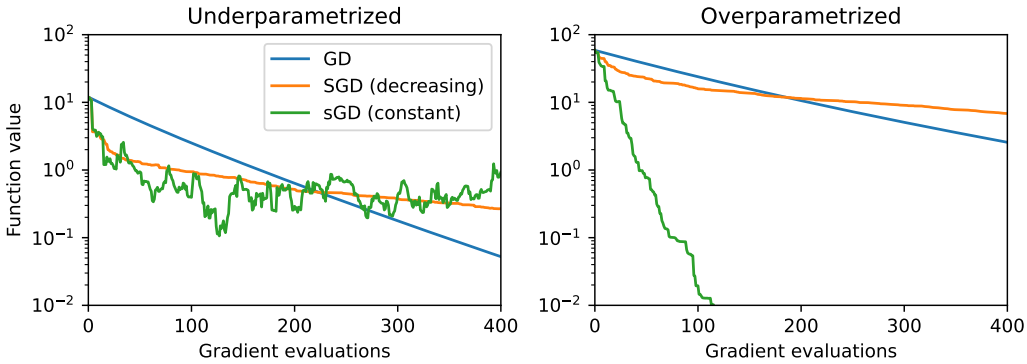
which implies the interpolation property that $\nabla f(w) = 0 \rightarrow \nabla f_i(w) = 0$ for all i.

- Under the SGC, SGD achieves the deterministic convergence rates,
  - $\mathbb{E}[f(w)] - f^* = O(\gamma^k)$ for strongly-convex and PL functions (for some $\gamma < 1$).
  - $\mathbb{E}[f(w)] - f^* = O(1/k)$ for convex functions.
  - $\mathbb{E}[\|\nabla f(w)\|^2] = O(1/k)$ for bounded-below functions (which may be non-convex).

- All of these above rates are obtained for any sufficiently small step size.
  - So SGD adapts to the difficulty of the problem.
    - The same step size works for strongly-convex and non-convex problems.
  - Partial explanation for the success of constant step sizes in practice.
    - Which do not converge in the usual setting.

Stochastic Convergence Rates under Over-Parameterization

- Comparison of least squares performance in under-/over-parameterized models:



Bound on Error under the SGC

- Under the SGC the SGD error is bounded by the full gradient size,

$$\begin{aligned}
\mathbb{E}[\|e^k\|^2] &= \mathbb{E}[\|\nabla f_i(w^k) - \nabla f(w^k)\|^2] && \text{(definition of } e^k\text{)} \\
&= \mathbb{E}[\|\nabla f_i(w^k)\|^2] - \|\nabla f_i(w^k)\|^2 && \text{(variance identity)} \\
&= \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(w^k)\|^2 - \|\nabla f_i(w^k)\|^2 && \text{(expand } \mathbb{E}\text{)} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \rho\|\nabla f(w^k)\|^2 - \|\nabla f(w^k)\|^2 && \text{(use SGC)} \\
&= (\rho - 1)\|\nabla f(w^k)\|^2 && \text{(simplify)}
\end{aligned}$$

- So under the SGC, we do not need an assumption like $\mathbb{E}[\|e^k\|] \leq \sigma^2$.

Progress Bound under the SGC

- Using SGD in descent lemma with $\alpha_k = 1/L\rho$, with SGC we obtain

$$\mathbb{E}[f(w^{k+1})] \leq f(w^k) - \frac{1}{2L\rho}\|\nabla f(w^k)\|^2,$$

  the function decrease of deterministic gradient descent up to a factor of ρ.
  - See bonus slide for the case of a general step size.
  - Any step size $\alpha < 2/L\rho$ guarantees descent.

- From this inequality you can derive the rates under the different assumptions.

Close to Over-Parameterized

- Often we are not over-parameterized but close to over-parameterized.
  - Training error can be made small but not exactly 0.

- To address this case you add a constant term to the SGC,

$$\mathbb{E}[\|\nabla f_i(w)\|^2] \leq \rho\|\nabla f(w)\|^2 + \sigma^2,$$

  combining SGC with earlier assumption on expected gradient size.
  - This condition is weaker than both of those, and allowing smaller ρ or σ.

- This is not sufficient for convergence with a constant step size.
  - But with constant step will converge quickly to region of size $O(\alpha\sigma^2)$.
    - If σ^2 is small, this may be all you need.
    - And again note that σ^2 decreases with the batch size.

SGC vs. Interpolation

- Under strong-convexity and SGC, SGD obtains a rate

$$\mathbb{E}[f(w^k)] - f(w^*) \leq \left(1 - \frac{\mu}{\rho L}\right)^k [f(w^0) - f(w^*)].$$

- But in the worst case, ρ can be as large as $L_{\max}/\mu$.
  - Where $L_{\max}$ is the maximum Lipschitz constant of the ∇f_i ($L_{\max} \geq L$).
- Assuming only the interpolation property (implied by SGC)

$$\mathbb{E}[\|\nabla f_i(w^*)\|^2] = 0,$$

we can show an alternate rate of

$$\mathbb{E}[f(w^k)] - f(w^*) \leq \left(1 - \frac{\mu}{L_{\max}}\right)^k [f(w^0) - f(w^*)],$$

which is faster for problems where $\rho L > L_{\max}$.

- Interpolation is not sufficient to get convergence for bounded-below functions.
- Bonus slides discuss weak growth condition which leads to faster rates than both.

Outline

1. SGD for PL Functions
2. Mini-Batch SGD and Growing Batches
3. SGD and Over-Parameterization
4. Faster Algorithms under Over-Parameterization

Faster SGD for Over-Parameterized Models?

- Over-parameterization leads to faster convergence rates for SGD.
- But can we exploit over-parameterization to develop faster methods than SGD?
  - Without needing to grow the batch size.
- Yes, there now exist methods that go faster in over-parameterized setting:
  - With Nesterov acceleration you can improve rate to $(1 - \sqrt{\mu/\rho L})$.
  - With non-uniform sampling proportional to L_i you can rates depending on $\bar{L}$.
  - With second-order updates and growing batch you can get faster local convergence.
    - With a much-slower growth in the batch size than without over-parameterization.
- Under over-parameterization, you can also use SGD with a line-search.

Review of Standard Methods to Automatically Set Step Size

- There are a huge number of papers on setting SGD step size as we go.
  - "Update step size based on some simple statistics".
  - "Do gradient descent on the step size".
  - "Use a line-search/trust-region based on the mini-batch".

- Most of these methods have at least one of these problems:
  - Introduces new hyper-parameter that is just as hard to tune as the step size.
  - Do not converge theoretically (and can catastrophically fail).
  - Converges theoretically, but works badly in practice.
  - Needs to assume that σ_k goes to 0 to work.

- Student recommendation when not over-parameterized: coin betting.
- If growing batch size or over-parameterized:
  - Can adapt step sizes and line searches designed for deterministic gradient descent.

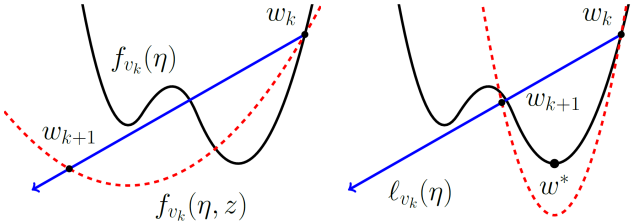
Stochastic Line Search - Theory

- An Armijo line-search on the mini-batch selects a step size satisfying

$$f_{i_k}(w_k - \alpha_k \nabla f_{i_k}) \leq f_{i_k}(w_k) - c\alpha_k \|\nabla f_{i_k}(w_k)\|^2,$$

for some constant $c > 0$.

- Without interpolation this does not work (satisfied by steps that are too large).



- With interpolation, can guarantee sufficient progress towards solution.

Stochastic Line Search - Theory

- Consider using the largest step-size satisfying Armijo condition on $[0, \alpha_{\max}]$.
  - Under interpolation and strong-convexity, $c = 1/2$ and $\alpha_{\max}$ sufficiently large gives

$$\mathbb{E}\left[\|w_k - w_*\|^2\right] = \left(1 - \frac{\mu}{L_{\max}}\right)^k \|w_0 - w_*\|^2.$$

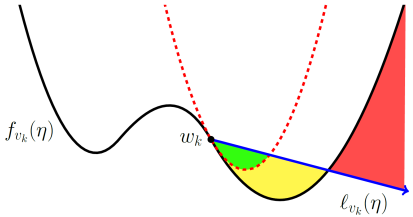
  - Same rate we achieve knowing smoothness constant under interpolation.
  - For convex objectives we obtain an $O(1/k)$ rate.
  - For non-convex objectives we obtain the $O(1/k)$ rate if $\alpha_{\max}$ is small enough.

- In practice, we can use a backtracking line search.
  - You can alternately use the stochastic Polyak step size if you know f^*.

Superiority of Line Search over Theoretical Step Sizes

- The line search guarantees same rate as when we know smoothness constant.
  - But this is in the worst case.

- We expect the line-search to converge faster in practice.



- Red dotted line is bound obtained with known smoothness for an f_i.
  - Using $\alpha_k = 1/L_{\max}$ moves to minimizer within green region.
- Armijo accepts step sizes in the yellow region (blue line is gradient of an f_i).
  - Armijo allows larger step sizes that decrease the function by a larger amount.

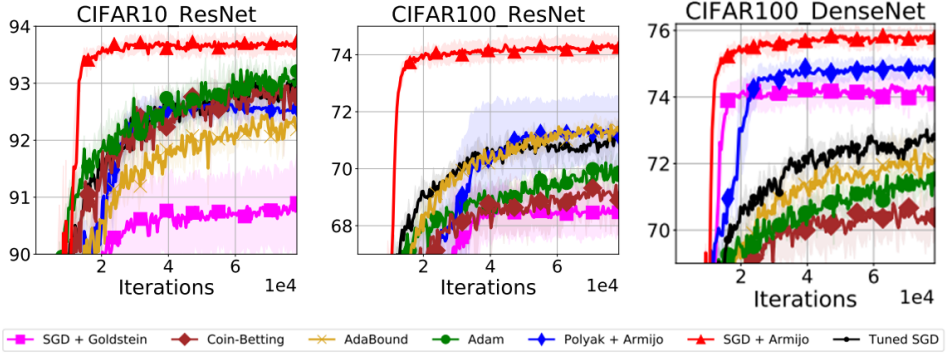
Stochastic Line Search - Practice

- In our experiments:
  - We used $c = 0.1$ in the Armijo condition.
  - We multiply the step size by 0.8 if the Armijo condition fails.
  - We increase the step size between iterations.
    - Specifically, we initialize the line search with $\max\{10, \alpha_{k-1} 2^{(\text{ratio of training data used})}\}$.

- With these choices, median number of times we test Armijo condition was 1.
  - Running this algorithm has similar cost to trying 2 fixed step sizes.

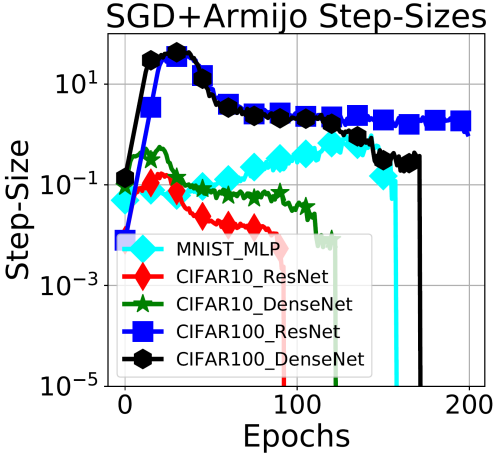
Experimental Results with Stochastic Line Search

- We did a variety of experiments, including training CNNs on standard problems.
  - Better than fixed step sizes, adaptive methods, alternate adaptive step sizes.



Experimental Results with Stochastic Line Search

- Step sizes over time under line search for different datasets.



Stochastic Line Search - Discussion

- The same line search can be used for different types of functions.
  - Strongly-convex, PL, or convex. (And bounded below under restriction of $\alpha_{\max}$.)
  - Adaptivity to problem difficulty.

- We were not the first to try line searches for SGD.
  - Or even Armijo line search for SGD applied to deep learning benchmarks.
  - But we showed why over-parameterization is key to performance.

- On synthetic experiments conrolling degree of over-parameterization.
  - With over-parameterization, the stochastic line search works great.
  - If close to over-parameterized, line search still works really well.
    - Theory can be modified to handle case of being close to over-parameterized.
  - If far from over-parameterized, line search catastrophically fails.

- Line search experiments were done with batch normalization.
  - This is not covered by the theory.
  - Armijo still seems effective but gap is not as large.

Problems with Current Over-Parameterization Optimization Theory

- Line search is not as effective for LSTMs or transformers.
  - Adam seems to have an advantage here.
  - Theoretical and practical details to be worked out.

- Some deep learning losses like in GANs do not fit over-parameterized regime.

(Chavdarova et al., 2019)

- Theory is still incomplete for non-convex functions:
  - Interpolation not sufficient for SGD to converge for non-convex.
    - Non-convex results rely on PL or SGC.
  - Line-search is not sufficient for convergence on non-convex.
    - Non-convex results require $\alpha_{\max} = O(1/L)$.

Summary

- Convergence of SGD with constant step size
  - Similar speed to gradient descent, up to accuracy proportional to step size.
  - For machine learning, this may be all you need.
- Mini-batch SGD:
  - Effect of noise is divided by the mini-batch size.
    - Effect of noise decrease faster for without-replacement sampling.
  - Growing batch sizes allow you use tricks for deterministic gradient descent.
    - Like acceleration, second-order information, and line searches.
- Over-Parameterized SGD:
  - For many problems we can exactly fit every example.
  - In this setting, SGD converges like gradient descent.
  - You can develop faster accelerated and second-order methods for this setting.
  - You can use line search or other clever step sizes in this setting.

- No lecture next week (I will be away).

Descent Lemma for Gradient Descent with Error

- Recall the descent lemma,

$$f(w^{k+1} \leq f(w^k) + \nabla f(w^k)^T(w^{k+1} - w^k) + \frac{L}{2}\|\nabla f(w^k)\|^2.$$

- Pluggin in gradient descent with error, $w^{k+1} - w^k = -\alpha_k(\nabla f(w^k) + e^k)$:

$$f(w^{k+1}) \leq f(w^k) - \alpha_k\|\nabla f(w^k)\|^2 - \alpha_k \nabla f(w^k)^T e^k + \frac{\alpha^2 L}{2}\left(\|\nabla f(w^k)\|^2 - 2\nabla f(w^k)^T e^k + \|e^k\|^2\right).$$

- If e^k is unbiased then $\nabla f(w^k)^T \mathbb{E}[e^k] = 0$ and after simplifying we get

$$\mathbb{E}[f(w^{k+1})] \leq f(w^k) - \alpha_k\left(1 - \frac{\alpha_k L}{2}\right)\|\nabla f(w^k)\|^2 + \frac{\alpha^2 L}{2}\mathbb{E}[\|e^k\|^2],$$

where the middle term on the right is negative if $\alpha_k < 2/L$.

Convergence Rate under PL and Bounded Variation

- Descent lemma for gradient descent with generic unbiased error e^k:

$$\mathbb{E}[f(w^{k+1})] \leq f(w^k) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \|\nabla f(w^k)\|^2 + \frac{\alpha^2 L}{2} \mathbb{E}[\|e^k\|^2].$$

- With $\alpha_k < 2/L$, PL ($\|\nabla f(w^k)\|^2 \geq 2\mu(f(w^k) - f^*)$), and $\mathbb{E}[\|e^k\|^2] \leq \sigma^2$ gives

$$\mathbb{E}[f(w^{k+1})] \leq f(w^k) - 2\mu\alpha_k \left(1 - \frac{\alpha_k L}{2}\right) (f(w^k) - f^*) + \frac{\alpha^2 L \sigma^2}{2}.$$

- Subtracting f^* from both sides and recursing with constant $\alpha_k = \alpha$ as before gives

$$\mathbb{E}[f(w^{k+1})] - f^* \leq \left(1 - 2\mu\alpha\left(1 - \frac{\alpha L}{2}\right)\right)^k (f(w^0) - f^*) + \alpha\sigma^2 \frac{L}{4\mu} \frac{1}{(1 - \frac{\alpha L}{2})},$$

  which is the result we had before with some extra factors.
- If $\alpha = 1/L$ RHS simplifies to $(1 - \mu/L)^k (f(w^0) - f^*) + \frac{\sigma^2}{2\mu}$.
- Or if $\alpha = \gamma/L$ for $\gamma < 2$ we get $(1 - \gamma(2 - \gamma)\mu/L)^k (f(w^0) - f^*) + \frac{\gamma\sigma^2}{2\mu(2-\gamma)}$.

Variation Bound for Mini-Batch Approximation

- Variation of mini-batch approximation with batch size of m:

$$\begin{aligned}
\mathbb{E}[\|g_m(w) - \nabla f(w)\|^2] &= \mathbb{E}[\|\frac{1}{m}\sum_{i\in\mathcal{B}} \nabla f_i(w)\|^2] - \|\nabla f(w)\|^2 && \text{(variance identity)} \\
&= \frac{1}{m^2}\sum_{i\in\mathcal{B}} \mathbb{E}[\|\nabla f_i(w)\|^2] + \frac{2}{m^2}\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B},j\neq i} \mathbb{E}[\nabla f_i(w)^T \nabla f_j(w)] - \|\nabla f(w)\|^2 && \text{(expand square)} \\
&= \frac{m}{m^2}\mathbb{E}[\|\nabla f_i(w)\|^2] + \frac{m(m-1)}{m^2}\nabla f(w)^T \nabla f(w) - \|\nabla f(w)\|^2 && \text{(repeated terms)} \\
&= \frac{1}{m}\mathbb{E}[\|\nabla f_i(w)\|^2] - \frac{1}{m}\|\nabla f(w)\|^2 && \text{(simplify)} \\
&= \frac{1}{m}\mathbb{E}[\|\nabla f_i(w) - \nabla f(w)\|^2] && \text{(variance identity)} \\
&= \frac{\sigma^2}{m} && (\sigma^2 \text{ is 1-sample variation})
\end{aligned}$$

Progress Bound under SGC with Generic Step Size

- Descent lemma for gradient descent with generic unbiased error e^k:

$$\mathbb{E}[f(w^{k+1})] \leq f(w^k) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \|\nabla f(w^k)\|^2 + \frac{\alpha^2 L}{2} \mathbb{E}[\|e^k\|^2].$$

- With SGC we have $\mathbb{E}[\|e^k\|^2] \leq (\rho - 1)\|\nabla f(w^k)\|^2$, giving

$$\begin{aligned}\mathbb{E}[f(w^{k+1})] &\leq f(w^k) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \|\nabla f(w^k)\|^2 + \frac{\alpha^2 L}{2}(\rho - 1)\|\nabla f(w^k)\|^2 \\ &= f(w^k) - \alpha_k \left(1 - \frac{\alpha L \rho}{2}\right) \|\nabla f(w^k)\|^2.\end{aligned}$$

- The second term is negative for any $\alpha_k < 2/L\rho$.
- With $\alpha_k = 1/L\rho$ we get

$$\mathbb{E}[f(w^{k+1})] \leq \left(1 - \frac{\mu}{L\rho}\right) \|\nabla f(w^k)\|^2.$$

Ways to Characterize Over-Parameterization

- First over-parameterization results are due to Solodov [1998] and Tseng [1998].
  - They considered variation on what is now called the strong growth condition (SGC),

$$\mathbb{E}[\|\nabla f_i(w)\|^2] \leq \rho \|\nabla f(w)\|^2.$$

- Bach & Moulines [2011] later analyze SGD when variance at solution is 0.
  - We call this the interpolation property (which is implied by the SGC),

$$\mathbb{E}[\|\nabla f_i(w_*)\|^2] = 0.$$

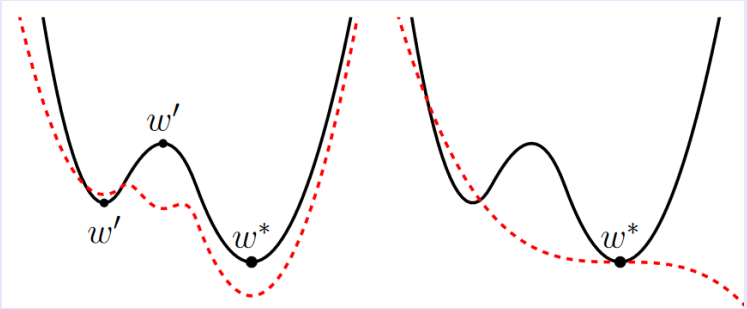
- An alternate condition was considerd by Vaswani et al. [2019].
  - The weak growth condition (WGC) for an L-smooth function is

$$\mathbb{E}[\|\nabla f_i(w)\|^2] \leq 2\rho L(f(w) - f(w_*)).$$

  - Relation between conditions for L-smooth f and $L_{\max}$-smooth f_i:
    - SGC $\rightarrow$ interpolation and WGC.
    - For invex functions: interpolation $\rightarrow$ WGC.
    - For PL functions: WGC $\rightarrow$ SGC.

Strong Growth Condition vs. Weak Growth Condition

- SGC implies each f_i is stationary when f is stationary.
- Interpolation and WGC imply each f_i is stationary at global minizers.



  - Neither condition rules out non-isolated or multiple global minimizers.
- The constant under WGC may be smaller:
  - For PL functions satisfying SGC we have $\rho \leq L_{\max}/\mu$.
  - For invex functions satisfying WGC we have $\rho \leq L_{\max}/L$.