The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Schedule for Least Squares

Rong Ge*, Sham M. Kakade**, Rahul Kidambi*** and Praneeth Netrapalli****

* Duke University, Durham NC

** University of Washington Seattle WA

*** Cornell University, Ithaca NY

**** Microsoft Research India

Paper ID: 8546, NeurIPS 2019

SGD: Theory Vs. Practice

- Stochastic Gradient Descent (SGD) [Robbins & Monro, '51]
  - Simple to implement, drives modern Machine Learning applications.

- In theory [Ruppert '88, Polyak & Juditsky '92]
  - Relies on **iterate averaging** [Rakhlin et al. 2012, Bubeck 2014].
  - Employs **polynomially** decaying stepsizes.
  - Minimax optimal predictive guarantees.

- In practice [Bottou & Bousquet 2008]
  - Implementations predominantly use the **final iterate** of SGD.
  - Employs a geometrically decaying "step decay" schedule.
  - Strong computation vs. generalization tradeoffs.

This Paper: SGD's Final Iterate For Least Squares

- Streaming Least Squares Regression – compute:

$$w^* \in \operatorname{argmin}_w f(w) = \frac{1}{2} \cdot \mathrm{E}_{(x,y)\sim D}\left[(y - w.x)^2\right],$$

- Several recent works e.g. [Bach & Moulines 2013, Jain et al. 2016, 2017].
  - Constant step size SGD + iterate averaging achieves minimax optimal rates.
- This paper: SGD's final iterate behavior.
  - Several works [Widrow et al. '60, Nagumo et al. '67, Proakis '74] etc.
  - These efforts do not achieve minimax rates.

Problem Setup

$$\text{Compute } w^* \in \operatorname{argmin}_w f(w) = \tfrac{1}{2}\cdot \mathrm{E}_{(x,y)\sim D}\left[(y - w.x)^2\right]$$

- Given samples $(x_1, y_1), \cdots, (x_N, y_N) \sim D \subseteq \mathrm{R}^d \times \mathrm{R}$.

$(x, y) \sim D, y = w^*.x + n,\ n$: noise.

- Hessian $H = \nabla^2 f(w) = E\left[xx^\top\right]$.
- Access to stochastic first order oracle (SFO), which when queried at any iterate w returns:

$$\widehat{\nabla f}(w) = -(y - w.x)\, x$$

Stochastic Gradient computed on a single example (x, y) at w.

Problem Setup (2)

- **Assumptions** (for precise statements, refer to paper):
  - A1. Strong convexity, i.e., $H \succ 0 \Rightarrow \mu = \lambda_{min}(H) > 0$.
  - A2. Bounded Inputs, i.e., $||x||^2 < \mathrm{R}^2$ almost surely.
    - Denote the condition number $\kappa := \mathrm{R}^2/\mu$.
  - A3. Bounded noise, i.e., $\forall\ (x, y) \sim D,\ |y - w^* x| < \sigma$ almost surely.
- Under A1-3, any algorithm outputting $\widehat{w}_t$ with t-calls to an SFO [Vaart 2000] has error at least:

$$F(\widehat{w}_t) - F(w^*) \geq \frac{d\sigma^2}{t}.$$

SGD Algorithm

Input: initialization w_0, # iterations N, step size $\{\gamma_t\}_{t=1}^N$

Repeat For $t = 1, \cdots, N$

$$w_{t+1} \leftarrow w_t - \gamma_t \cdot \widehat{\nabla f}(w_t) \quad \text{// Invoke SFO at } w_t$$

Return w_{N+1}

Theory: For general classes of objectives, returning $\overline{w} = \frac{1}{N}\sum_i w_i$ with poly stepsizes $\gamma_t \propto \frac{1}{t^\alpha}, \alpha \in [0.5, 1)$ achieves minimax rates.

Question 1: What happens to w_N for streaming least squares regression with polynomially decaying stepsizes?

Practice: Use the geometrically decaying "step-decay" schedule and return w_N.

Question 2: What happens to w_N for streaming least squares regression with geometrically decaying stepsizes?

Q1: SGD's Final Iterate with Polynomially Decaying Stepsizes

Theorem (informal):

For a given end time N, under assumptions A1-3, there exists a streaming least squares regression problem such that **every** step size scheme $\gamma_t = \frac{a}{b+t^\alpha}$, with $\alpha \in [0.5,1]$ suffers the error:

$$E[F(w_N)] - F(w^*) \geq \kappa \cdot \frac{d\sigma^2}{N}.$$

Comments:

[1] Final iterate + poly decay: highly sub-optimal (by a condition number).

[2] Recall, iterate averaging + poly decay: optimal behavior.

SGD With The Step-Decay Schedule

Input: initialization w_0, # iterations N, initial step size γ_0

Repeat for $e = 1, \cdots, \log N$:

$\gamma_e \leftarrow \gamma_0 / 2^{e-1}$ // Halve learning rate every epoch

Repeat for $t = 1, \cdots, N/\log N$:

$w \leftarrow w - \gamma_e \cdot \widehat{\nabla f}(w)$ // Invoke SFO at w

Return w

- **Remarks:** Final iterate w returned – similar to practice.
  - To run the algorithm: we require initial stepsize γ_0 and # iterations N.
  - Used heavily in algorithms for modern ML/AI models [Krizhevsky et al. 2012].

Q2: SGD's Final Iterate with Step-Decay Schedule

Theorem (informal):

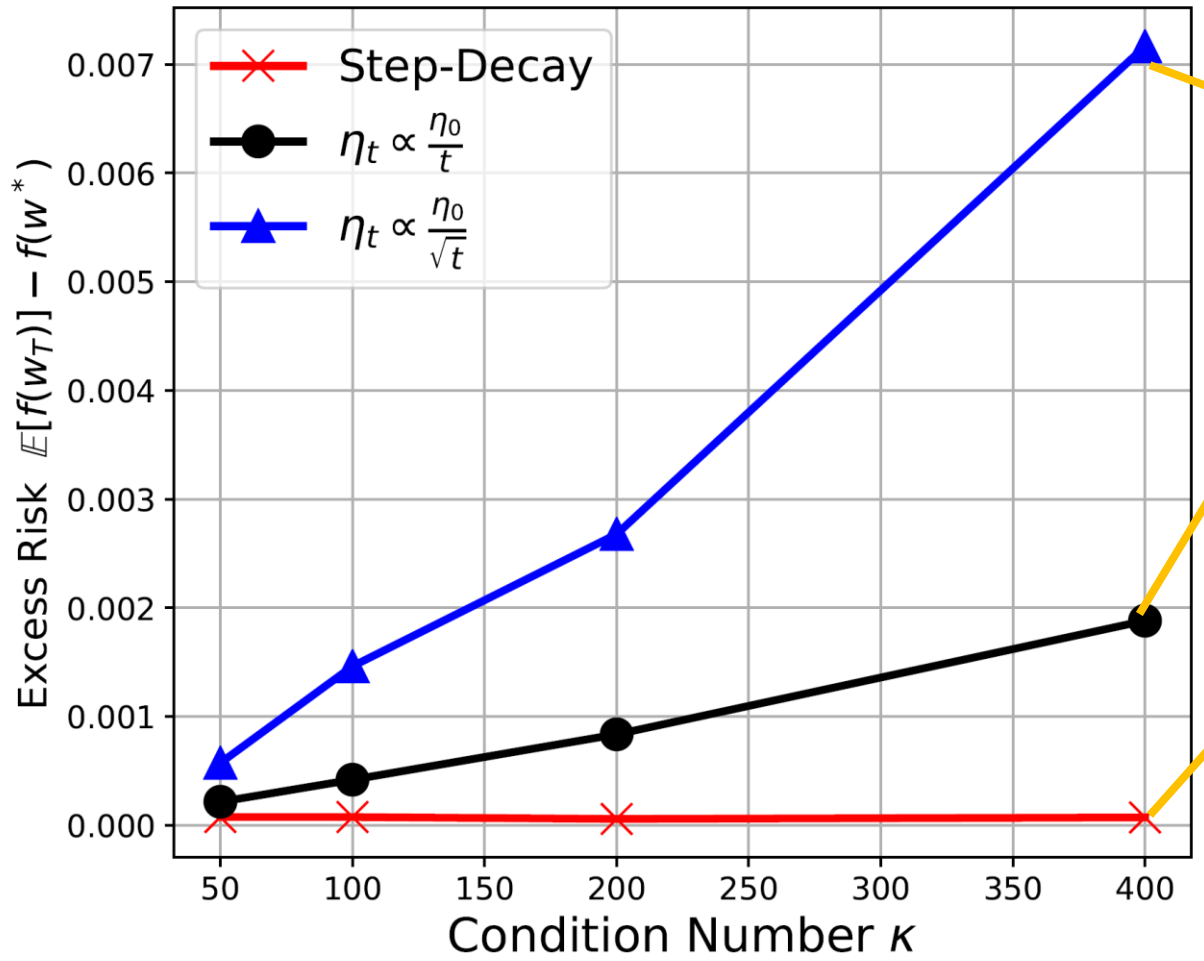
Under assumptions A1-3, SGD with an initial learning rate $\gamma = \frac{1}{R^2}$ and the step decay schedule offers the following guarantee:

$$E[F(w)] - F(w^*) \leq \exp\left(-\frac{N}{\kappa \cdot \log(N)}\right) \cdot \left(F(w_0) - F(w^*)\right) + \log N \cdot \frac{d\sigma^2}{N} \ .$$

- **Variance** term: Near-optimal rate (upto $\log(N)$) factors.
- SGD with step-decay requires knowing # iterations N in advance.
- Significant difference compared to polynomially decaying stepsizes.
- Related work: Shamir & Zhang (2012), Jain, Nagaraj & Netrapalli (2019).

Simulations: Synthetic Streaming Least Squares

- Plot of final iterate's error (y-axis) against condition number (x-axis)



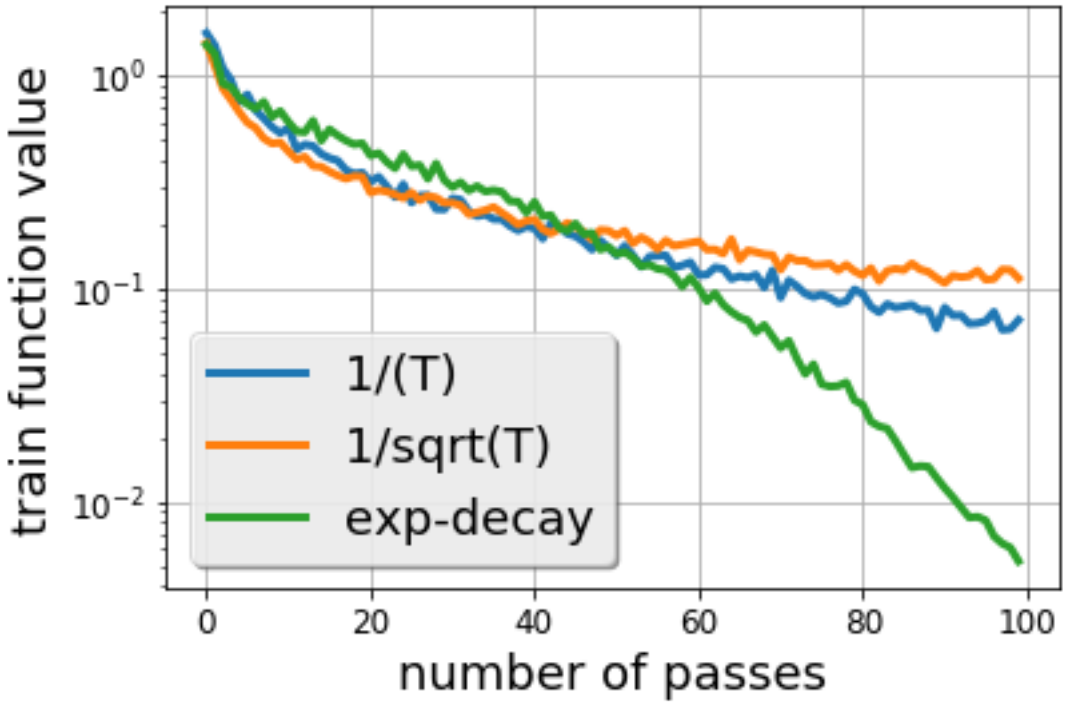
Polynomially Decaying Stepsizes: Error grows linearly wrt Condition Number.

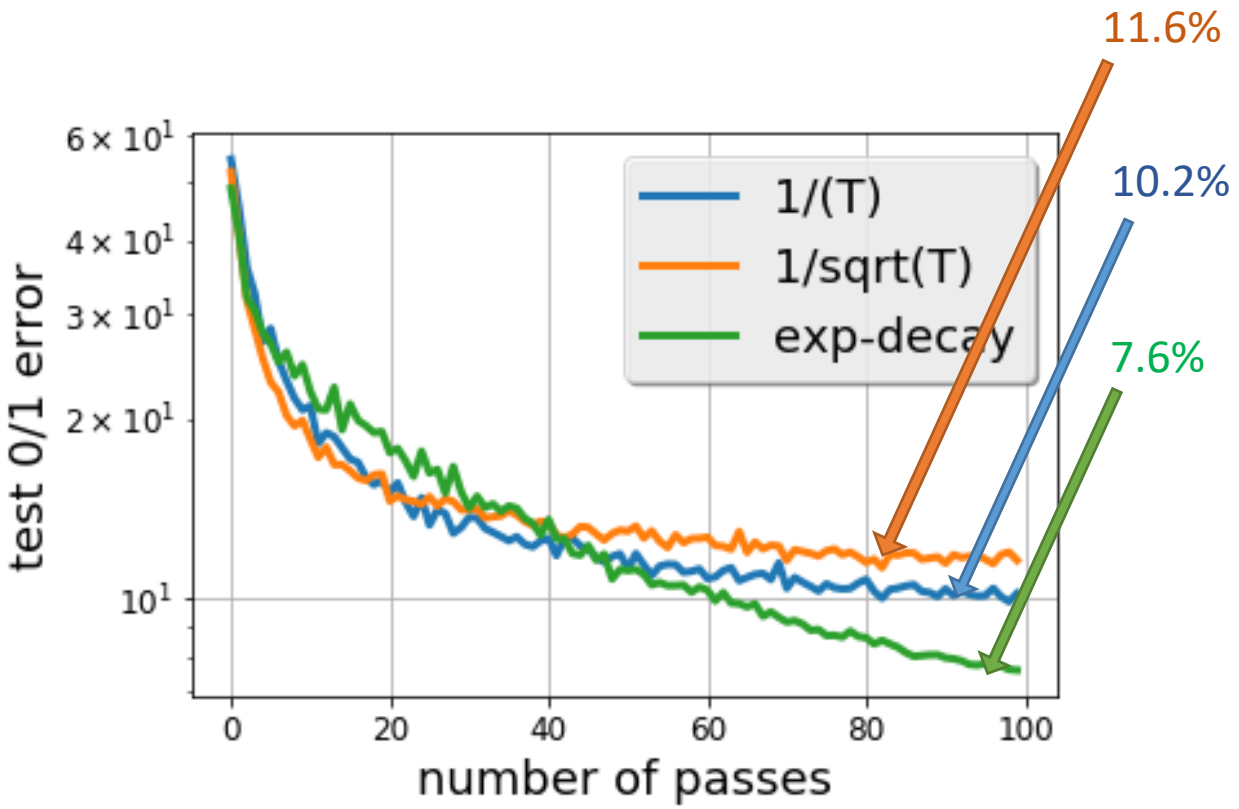
Step Decay: Error is near constant as a function of condition number (sub-optimal by $\log(T)$ factors)

Non-Convex Optimization: CIFAR-10 With ResNet-44

- Grid search three schemes (with a fixed end time):

$$\text{(a) } \eta_t = \frac{\eta_0}{1+b\cdot t}, \text{(b) } \eta_t = \frac{\eta_0}{1+b\sqrt{t}}, \text{(c) } \eta_t = \eta_0 \cdot \exp(-bt)$$





Towards Anytime Algorithms

- This paper's results: assumes the # iterations N to be fixed apriori.
- How about **anytime** algorithms?
  - Anytime ⇒ Doesn't require knowing # iterations N in advance.
  - Iterate averaging + polynomially decaying stepsizes ⇒ **anytime** optimal.
  - What about the final iterate?
- The next slide presents anytime behavior of SGD's final iterate for the streaming least squares regression problem.

Q2: SGD's Final Iterate with Step-Decay Schedule

Theorem (informal):
Under assumptions A1-3, SGD's final iterate with stepsizes $\gamma_t \leq 1/(2\,\mathrm{R}^2)$ queries highly sub-optimal iterates infinitely often. In particular,

$$\limsup_{T\to\infty} \frac{E[f(w_T)] - f(w^*)}{d\sigma^2/T} \geq C \cdot \frac{\kappa}{\log(\kappa)}$$

- **Related work:** See Harvey et al. (2019) for a similar statement in non-smooth stochastic convex optimization.
- See also the related work of Ge et al. (2019), a COLT open problem about understanding the sub-optimality of query points more generally.

Other Empirical Results on CIFAR-10 with ResNet-44

- **Suffix iterate averaging versus final iterate with polynomially decaying stepsizes**:
  - Empirical evidence indicates that suffix averaging (regardless of the suffix length) offers little advantage over the final iterate behavior for non-convex optimization involving training a ResNet-44 model on CIFAR-10 dataset.

- **Hyper-parameter optimization with truncated runs**:
  - The broader issues concerning design of anytime optimal SGD methods tends to imply hyper-parameter search methods based on truncated runs may benefit from a round of rethinking.

Conclusions

- For the streaming least squares problem, SGD's final iterate behavior:
  - With polynomially decaying stepsizes is highly sub-optimal.
  - With step-decay schedule is near-optimal (upto logarithmic factors).

- The behavior of SGD's final iterate in an anytime sense is sub-optimal in that SGD needs to query highly sub-optimal points infinitely often.

- Empirical results and ramifications towards the use of iterate averaging and for hyper-parameter optimization is shown through optimizing a ResNet-44 on the CIFAR-10 dataset.