Stochastic Gradients with Adaptive Stepsizes

Rachel Ward

University of Texas at Austin
Department of Mathematics
The Oden Institute for Computational Engineering and Sciences
Institute for Machine Learning

Texas A&M, October 2022

Acknowledgments

Joint work with Xiaoxia Wu (Microsoft), Leon Bottou (Meta AI Research), Matthew Faw (UT Austin), Isidoros Tziotis (UT Austin), Constantine Caramanis (UT Austin), Aryan Mokhtari (UT Austin), and Sanjay Shakkottai (UT Austin)

Thanks also to Francesco Orabona for helpful feedback

Funding thanks: NSF, AFOSR, DOE

Optimization in large-scale machine learning

"Finite sum" form:

$$\min_{w \in \mathbb{R}^p} F(w), \qquad F(w) = \frac{1}{n} \sum_{j=1}^{n} f_j(w)$$

- average of functions is the loss function (least squares loss for regression, cross entropy loss for classification)
- $f_j(w)$ is the loss term associated to fitting jth training data point to the model class parameterized by weights $w \in \mathbb{R}^p$.

Optimization in large-scale machine learning

"Finite sum" form:

$$\min_{w \in \mathbb{R}^p} F(w), \qquad F(w) = \frac{1}{n} \sum_{j=1}^{n} f_j(w)$$

- average of functions is the loss function (least squares loss for regression, cross entropy loss for classification)
- $f_j(w)$ is the loss term associated to fitting jth training data point to the model class parameterized by weights $w \in \mathbb{R}^p$.

Dimensions are large – for example, dimension of each training point, number of data points n, and number of weights p are on order of **billions**.

At these scales, only simple first-order optimization methods (methods which require only first-derivative/gradient computations) can be implemented practically.

Stochastic Gradient Descent (SGD)

When n is large, computing even a *single* gradient $\nabla F(w) = \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(w)$ is costly.

Significantly cheaper: Draw random index i from $\{1, \ldots, n\}$ and compute a single component gradient $\nabla f_i(w)$.

$\mathbb{E}_i \nabla f_i(w) = \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(w) = \nabla F(w)$.

Stochastic Gradient Descent (SGD)

When n is large, computing even a *single* gradient $\nabla F(w) = \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(w)$ is costly.

Significantly cheaper: Draw random index i from $\{1, \ldots, n\}$ and compute a single component gradient $\nabla f_i(w)$.

$\mathbb{E}_i \nabla f_i(w) = \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(w) = \nabla F(w)$.

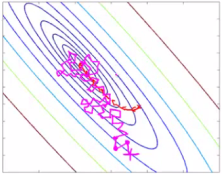
Stochastic Gradient "Descent":

- **Initialize** $w_1 \in \mathbb{R}^p$;
- **Until** convergence,

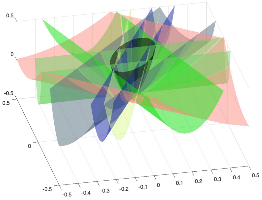
  $t + 1 \leftarrow t$

  Draw random index i_t *from*$\{1, \ldots, n\}$

  $w_{t+1} \leftarrow w_t - \eta_t \nabla f_{i_t}(w_t)$



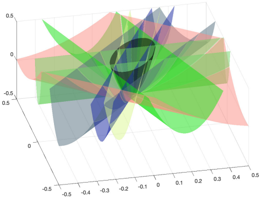
Stochastic Gradient Descent (SGD)



Example: least squares regression/interpolation:

$$F(w) = \|Aw - y\|_2^2$$
$$= \frac{1}{n} \sum_{j=1}^{n} n(\langle a_j, w \rangle - y_j)^2$$

Stochastic Gradient Descent (SGD)



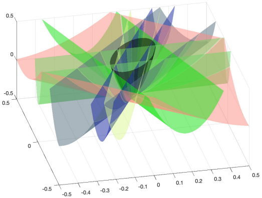
Example: least squares regression/interpolation:

$$F(w) = \|Aw - y\|_2^2$$
$$= \frac{1}{n}\sum_{j=1}^{n} n(\langle a_j, w\rangle - y_j)^2$$

SGD update: $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t) = w_t - \eta_t \left(\langle a_{i_t}, w_t\rangle - y_{i_t}\right) a_{i_t}^T$

Related: Alternating Projections onto Convex sets (von Neumann 1933), randomized Kaczmarz algorithm (Strohmer, Vershynin 2007), Stochastic approximation (Robbins, Monro 1951).

Stochastic Gradient Descent (SGD)



Example: least squares regression/interpolation:

$$F(w) = \|Aw - y\|_2^2$$
$$= \frac{1}{n}\sum_{j=1}^{n} n(\langle a_j, w\rangle - y_j)^2$$

SGD update: $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t) = w_t - \eta_t \left(\langle a_{i_t}, w_t\rangle - y_{i_t}\right) a_{i_t}^T$

Related: Alternating Projections onto Convex sets (von Neumann 1933), randomized Kaczmarz algorithm (Strohmer, Vershynin 2007), Stochastic approximation (Robbins, Monro 1951).

Affine variance bound:
$\mathbb{E}_{i_t}\|\nabla f_{i_t}(w) - \nabla F(w)\|_2^2 \leq \min_w F(w) + (\|A\|_F^2 - \|A\|^2)\|\nabla F(w)\|^2$

SGD: General framework

Stochastic Gradient Descent for solving $\min_{w \in \mathbb{R}^p} F(w)$

- **Initialize** $w_1 \in \mathbb{R}^p$;
- **Until** convergence:
  - $t + 1 \leftarrow t$
  - Generate a realization of the random variable ξ_t
  - Compute a stochastic vector $g_t = g(w_t, \xi_t)$
  - Choose a step-size $\eta_t > 0$
  - Set the new iterate as $w_{t+1} = w_t - \eta_t g_t$

SGD: General framework

Stochastic Gradient Descent for solving $\min_{w \in \mathbb{R}^p} F(w)$

- **Initialize** $w_1 \in \mathbb{R}^p$;
- **Until** convergence:
  - $t + 1 \leftarrow t$
  - Generate a realization of the random variable ξ_t
  - Compute a stochastic vector $g_t = g(w_t, \xi_t)$
  - Choose a step-size $\eta_t > 0$
  - Set the new iterate as $w_{t+1} = w_t - \eta_t g_t$

Standard Assumptions:

- $\{\xi_t\}$ is a sequence of jointly independent random variables
- $\mathbb{E}_{\xi_t} g_t = \nabla F(w_t)$
- Affine variance $\mathbb{E}_{\xi_t} \|g_t - \nabla F(w_t)\|_2^2 \leq \sigma_0^2 + \sigma_1^2 \|\nabla F(w_t)\|_2^2$
- L-Lipschitz-continuous gradient:
  $\|\nabla F(w) - \nabla F(z)\|_2 \leq L \|w - z\|_2$ for all $w, z \in \mathbb{R}^p$
- $F_{\min} := \inf_w F(w) > -\infty$

SGD: Convergence theory

Theorem (Ghadimi and Lan, 2013, Bottou et al 2018)

Under the standard assumptions, consider the SGD algorithm with fixed step-size $\eta_t = \eta$ *satisfying* $0 < \eta \leq \frac{1}{L(1+\sigma_1^2)}$. *The expected mean sum-of-squares gradients of* F *corresponding to the SGD iterates satisfy for all* $T \in \mathbb{N}$:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla F(w_t)\|_2^2\right] \leq \eta L \sigma_0^2 + \frac{2(F(w_1) - F_{min})}{\eta T}$$

SGD: Convergence theory

Theorem (Ghadimi and Lan, 2013, Bottou et al 2018)

Under the standard assumptions, consider the SGD algorithm with fixed step-size $\eta_t = \eta$ satisfying $0 < \eta \leq \frac{1}{L(1+\sigma_1^2)}$. The expected mean sum-of-squares gradients of F corresponding to the SGD iterates satisfy for all $T \in \mathbb{N}$:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla F(w_t)\|_2^2\right] \leq \eta L\sigma_0^2 + \frac{2(F(w_1) - F_{min})}{\eta T}$$

Corollary: Fix $T \in \mathbb{N}$ and $\eta = \min\left\{\frac{\sqrt{2(F(w_1)-F_{\min})}}{\sqrt{L}\sigma_0\sqrt{T}}, \frac{1}{L(1+\sigma_1^2)}\right\}$.

Then

$$\min_{1\leq t\leq T}\mathbb{E}\|\nabla F(w_t)\|_2^2 \leq \frac{C^2(1+\sigma_1^2)}{T} + \frac{C\sigma_0}{\sqrt{T}}$$

where $C = \sqrt{2L(F(w_1) - F_{\min})}$.

Proof

First, by L-smoothness of $F(\cdot)$,

$$\eta\|\nabla F(w_t)\|^2 \leq F(w_t) - F(w_{t+1}) + \eta\langle \nabla F(w_t), \nabla F(w_t) - g_t\rangle + \frac{L\eta^2}{2}\|g_t\|_2^2$$

Proof

First, by L-smoothness of $F(\cdot)$,

$$\eta\|\nabla F(w_t)\|^2 \le F(w_t) - F(w_{t+1}) + \eta\langle \nabla F(w_t), \nabla F(w_t) - g_t\rangle + \frac{L\eta^2}{2}\|g_t\|_2^2$$

Because $\mathbb{E}_{\xi_t}[g_t] = \nabla F(w_t)$,

$$\|\nabla F(w_t)\|^2 \le \frac{F(w_t) - \mathbb{E}_{\xi_t}[F(w_{t+1})]}{\eta} + \frac{L\eta}{2}\mathbb{E}_{\xi_t}\|g_t\|_2^2$$

Proof

First, by L-smoothness of $F(\cdot)$,

$$\eta\|\nabla F(w_t)\|^2 \leq F(w_t) - F(w_{t+1}) + \eta\langle \nabla F(w_t), \nabla F(w_t) - g_t\rangle + \frac{L\eta^2}{2}\|g_t\|_2^2$$

Because $\mathbb{E}_{\xi_t}[g_t] = \nabla F(w_t)$,

$$\|\nabla F(w_t)\|^2 \leq \frac{F(w_t) - \mathbb{E}_{\xi_t}[F(w_{t+1})]}{\eta} + \frac{L\eta}{2}\mathbb{E}_{\xi_t}\|g_t\|_2^2$$

$\mathbb{E}_{\xi_t}\|g_t\|_2^2 \leq \sigma_0^2 + (\sigma_1^2 + 1)\|\nabla F(w_t)\|_2^2$ by assumption, so

$$\left(1 - \frac{L\eta(\sigma_1^2 + 1)}{2}\right)\|\nabla F(w_t)\|^2 \leq \frac{F(w_t) - \mathbb{E}_{\xi_t}[F(w_{t+1})]}{\eta} + \frac{\eta L\sigma_0^2}{2}$$

Proof

First, by L-smoothness of $F(\cdot)$,

$$\eta\|\nabla F(w_t)\|^2 \leq F(w_t) - F(w_{t+1}) + \eta\langle \nabla F(w_t), \nabla F(w_t) - g_t\rangle + \frac{L\eta^2}{2}\|g_t\|_2^2$$

Because $\mathbb{E}_{\xi_t}[g_t] = \nabla F(w_t)$,

$$\|\nabla F(w_t)\|^2 \leq \frac{F(w_t) - \mathbb{E}_{\xi_t}[F(w_{t+1})]}{\eta} + \frac{L\eta}{2}\mathbb{E}_{\xi_t}\|g_t\|_2^2$$

$\mathbb{E}_{\xi_t}\|g_t\|_2^2 \leq \sigma_0^2 + (\sigma_1^2 + 1)\|\nabla F(w_t)\|_2^2$ by assumption, so

$$\left(1 - \frac{L\eta(\sigma_1^2 + 1)}{2}\right)\|\nabla F(w_t)\|^2 \leq \frac{F(w_t) - \mathbb{E}_{\xi_t}[F(w_{t+1})]}{\eta} + \frac{\eta L\sigma_0^2}{2}$$

Using $\eta \leq \frac{1}{L(1+\sigma_1^2)}$, summing over $1 \leq t \leq T$ and applying the law of total expectation,

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla F(w_t)\|_2^2\right] \leq \eta L\sigma_0^2 + \frac{2(F(w_1) - F_{\min})}{\eta T}$$

SGD: Optimality

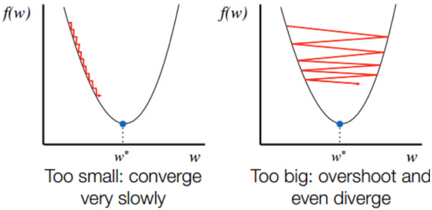
Under the smoothness assumptions here, *any* algorithm accessing a smooth function through a *stochastic first-order oracle* satisfying $\mathbb{E}_{\xi_t} g_t = \nabla F(w_t)$ and $\mathbb{E}_{\xi_t} \|g_t - \nabla F(w_t)\|_2^2 \leq \sigma_0^2$ requires

$$\Omega(L(F(w_1) - F_{\min})\sigma_0^2 \epsilon^{-4})$$

oracle queries to find a point w such that $\mathbb{E}\|\nabla F(w)\| \leq \epsilon$

Arjevani et al, *Lower Bounds for Non-Convex Stochastic Optimization*, 2019.

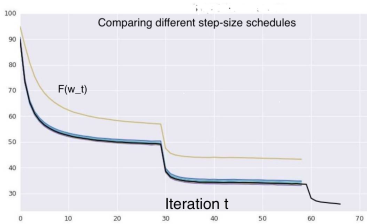
SGD: from theory to practice



A shortcoming of SGD is that the convergence is very sensitive to the choice of step-size schedule. The smoothness parameter L and noise variance parameters σ_0^2, and σ_1^2 determining a good step-size schedule in theory are not known in practice.

Line search heuristics for adaptively choosing the step-size at each iteration – which solve the problem in the setting of standard gradient descent – do not work in the presence of noise (yet)

Implementing SGD in practice



In practice, a good step-size schedule is found by manual trial and error, searching over schedules of form

$$\eta_j = \begin{cases} \alpha, & t = 1, \ldots, T_1 \\ \tau\alpha, & t = T_1 + 1, \ldots, T_2 \\ \tau^2\alpha, & t = T_2 + 1, \ldots \end{cases}$$

- Using *adaptive step-size* variations of SGD which learn a good step-size *along the way* are useful for making convergence more automatic and robust

AdaGrad: adaptive step-size updates[1]

SGD with Adagrad step-size updates

- **Initialize** $w_1 \in \mathbb{R}^p$, $\mathrm{b}_0 = \epsilon$, and scalar $\eta > 0$;
- **Until** convergence:
  - $t + 1 \leftarrow t$
  - Generate a realization of the random variable ξ_t
  - Compute a stochastic vector $g_t = g(w_t, \xi_t)$
  - Per coordinate, update $b_{t,j}^2 = b_{t-1,j}^2 + |g_{t,j}|^2$

    Coordinate step-size update $\eta_{t,j} = \frac{\eta}{b_{t,j}} = \frac{\eta}{\sqrt{\epsilon^2 + \sum_{s=1}^{t} |g_{s,j}|^2}}$
  - Update new iterate per coordinate as $w_{t+1,j} = w_{t,j} - \eta_{t,j} g_{t,j}$

[1][Duchi, Hazan, Singer 2011], [McMahan, Streeter, 2010]

AdaGrad: adaptive step-size updates[1]

SGD with Adagrad step-size updates

- **Initialize** $w_1 \in \mathbb{R}^p$, $\mathrm{b}_0 = \epsilon$, and scalar $\eta > 0$;
- **Until** convergence:
  - $t + 1 \leftarrow t$
  - Generate a realization of the random variable ξ_t
  - Compute a stochastic vector $g_t = g(w_t, \xi_t)$
  - Per coordinate, update $b_{t,j}^2 = b_{t-1,j}^2 + |g_{t,j}|^2$

    Coordinate step-size update $\eta_{t,j} = \frac{\eta}{b_{t,j}} = \frac{\eta}{\sqrt{\epsilon^2 + \sum_{s=1}^{t} |g_{s,j}|^2}}$
  - Update new iterate per coordinate as $w_{t+1,j} = w_{t,j} - \eta_{t,j} g_{t,j}$

AdaGrad became popular for always converging reasonably well without step-size tuning

[1] [Duchi, Hazan, Singer 2011], [McMahan, Streeter, 2010]

AdaGrad-Norm adaptive step-size update rule [2]

Simple starting point for analysis: SGD with scalar Adagrad step-size update

- **Initialize** $w_1 \in \mathbb{R}^p$ and scalars $\eta > 0$ and $b_0 = \epsilon > 0$;
- **Until** convergence:
  - $t + 1 \leftarrow t$
  - Generate a realization of the random variable ξ_t
  - Compute a stochastic vector $g_t = g(w_t, \xi_t)$
  - Update $b_t^2 = b_{t-1}^2 + \|g_t\|_2^2 = b_0^2 + \sum_{s=1}^t \|g_s\|_2^2$;

    Use step-size $\eta_t = \frac{\eta}{b_t} = \frac{\eta}{\sqrt{\epsilon^2 + \sum_{s=1}^t \|g_s\|_2^2}}$
  - Set the new iterate as $w_{t+1} = w_t - \eta_t g_t$

[2] [Li, Orabona 2018], [W,Wu, Bottou 2018]

SGD: General framework

Stochastic Gradient Descent for solving $\min_{w \in \mathbb{R}^p} F(w)$

- **Initialize** $w_1 \in \mathbb{R}^p$;
- **Until** convergence:
  - $t + 1 \leftarrow t$
  - Generate a realization of the random variable ξ_t
  - Compute a stochastic vector $g_t = g(w_t, \xi_t)$
  - Choose a step-size $\eta_t > 0$
  - Set the new iterate as $w_{t+1} = w_t - \eta_t g_t$

Standard Assumptions:

- $\{\xi_t\}$ is a sequence of jointly independent random variables
- $\mathbb{E}_{\xi_t} g_t = \nabla F(w_t)$
- Affine variance $\mathbb{E}_{\xi_t} \|g_t - \nabla F(w_t)\|_2^2 \leq \sigma_0^2 + \sigma_1^2 \|\nabla F(w_t)\|_2^2$
- L-Lipschitz-continuous gradient:
  $\|\nabla F(w) - \nabla F(z)\|_2 \leq L\|w - z\|_2$ for all $w, z \in \mathbb{R}^p$
- $F_{\min} := \inf_w F(w) > -\infty$

SGD with AG-Norm step-size always converges

Theorem

Under the standard assumptions, SGD with AG-Norm adaptive step-size update exhibits convergence at the rate

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla F(w_t)\|_2^2\right] \leq \frac{C_0}{\sqrt{T}} logs(T) + \frac{C_1}{T} logs(T),$$

where C_0, C_1 depend 'reasonably' on $F(w_1) - F^{min}, \sigma_0, \eta L, \sigma_1$. Moreover, $C_0 = 0$ when $\sigma_0 = \sigma_1 = 0$.

- Adagrad-Norm has order-optimal (up to log factors) convergence rate of SGD with carefully tuned step-sizes in terms of L, σ_0, σ_1.

*Faw, Tziotis, Caramanis, Mokhtari, Shakkottai, W 2022;
W, Wu, Bottou 2018 Li and Orabona 2018

Challenges of adaptive analysis

Start with the standard first step in SGD analysis of L-smooth $F(\cdot)$:

$$\eta_t \|\nabla F(w_t)\|^2 \le F(w_t) - F(w_{t+1}) + \eta_t \langle \nabla F(w_t), \nabla F(w_t) - g_t \rangle + \frac{L\eta_t^2}{2} \|g_t\|^2$$

To obtain the target $\tilde{\mathcal{O}}(1/\sqrt{T})$ rate we want

$$\frac{1}{\sqrt{T}} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla F(w_t)\|^2\right] \le F(w_1) - F_{\min} + \text{const} \log(T)$$

[3] For $a_1 \ge 1$ and $a_2, \ldots, a_n \ge 0$, $\sum_{k=1}^{n} \frac{a_k}{\sum_{j=1}^{k} a_j} \le \log(\sum_{j=1}^{n} a_j) + 1$

Challenges of adaptive analysis

Start with the standard first step in SGD analysis of L-smooth $F(\cdot)$:

$$\eta_t \|\nabla F(w_t)\|^2 \leq F(w_t) - F(w_{t+1}) + \eta_t \langle \nabla F(w_t), \nabla F(w_t) - g_t \rangle + \frac{L\eta_t^2}{2} \|g_t\|^2$$

To obtain the target $\tilde{\mathcal{O}}(1/\sqrt{T})$ rate we want

$$\frac{1}{\sqrt{T}} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla F(w_t)\|^2\right] \leq F(w_1) - F_{\min} + \text{const} \log(T)$$

In the adaptive step-size setting $\eta_t = \frac{\eta}{\sqrt{\epsilon^2 + \sum_{s=1}^{t} \|g_s\|_2^2}}$:

- Good news: $\mathbb{E}\left[\sum_{t=1}^{T} \eta_t^2 \|g_t\|^2\right] = \mathcal{O}(\log(T))$

  $\Rightarrow$ variance term is bounded[3]
- Bad news: The inner-product term is *not* mean-zero (since η_t is a random variable and depends on g_t)
  - Not clear if $\eta_t \geq 1/\sqrt{t}$, even in expectation

[3] For $a_1 \geq 1$ and $a_2, \ldots, a_n \geq 0$, $\sum_{k=1}^{n} \frac{a_k}{\sum_{j=1}^{k} a_j} \leq \log(\sum_{j=1}^{n} a_j) + 1$

Challenges of adaptivity

$$\eta_t\|\nabla F(w_t)\|^2 \leq (F(w_t) - F(w_{t+1}) + \eta_t\langle\nabla F(w_t), \nabla F(w_t) - g_t\rangle + \frac{L\eta_t^2}{2}\|g_t\|^2$$

- Bounding the biased inner-product term: introduce "surrogate" step-size $\tilde{\eta}_t = \frac{\eta}{\sqrt{b_{t-1}^2+(1+\sigma_1^2)\|\nabla F(w_t)\|^2+\sigma_0^2}}$ for analysis
- Lower bounding η_t (in expectation)
  - First prove that $\mathbb{E}\left[\sum_{t=1}^T \|\nabla F(w_t)\|^2\right] = \tilde{\mathcal{O}}(T^3)$ deterministically
  - Starting from the crude polynomial bound, recursively refine the bound

Extension to coordinate Adagrad

- Zou et al (2019) extended $\tilde{\mathcal{O}}(1/\sqrt{T})$ convergence of Adagrad-Norm to coordinate-wise Adagrad
  - *Careful:* Wilson et al (2017): *The marginal value of adaptive gradient methods in machine learning*

Adam: Adagrad + Momentum[4]

SGD with Adam step-size updates

- **Initialize** $w_1 \in \mathbb{R}^p$, $\mathrm{b}_0 = \epsilon$, $m_0 = 0$, and scalars $\eta > 0$, $0 < \beta_2 \leq 1, 0 \leq \beta_1 < \beta_2$;
  - **Until** convergence:
    - $t + 1 \leftarrow t$
    - Generate a realization of the random variable ξ_t
    - Compute a stochastic vector $g_t = g(w_t, \xi_t)$
    - Per coordinate updates:

$$m_{t,j} = \beta_1 m_{t-1,j} + g_{t,j}$$

$$b_{t,j}^2 = \beta_2 b_{t-1,j}^2 + |g_{t,j}|^2$$

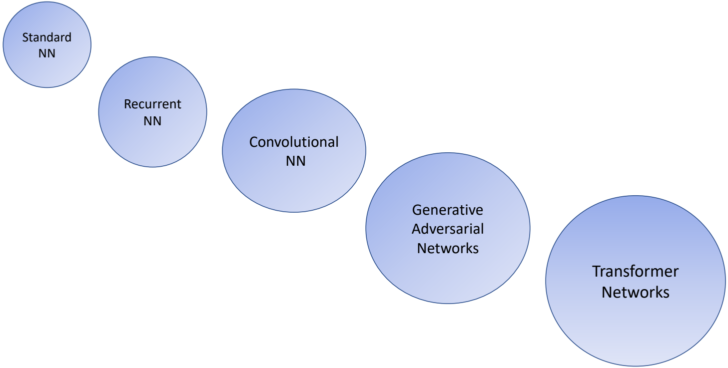
    - Update coordinate step-sizes $\eta_{t,j} = \frac{\eta}{b_{t,j}}$
      Update $w_{t+1,j} = w_{t,j} - \eta_{t,j} m_{t,j}$

$\beta_1 = 0, \beta_2 = 1$ recovers Adagrad.

[4] [Kingman, Ba 2014]

Adam: Adagrad + Momentum

Adam has remained one of the most popular optimization algorithms for deep learning, even as state-of-art architectures change



How can we understand Adam?

- Defossez et al (2020): extended $\tilde{\mathcal{O}}(1/\sqrt{T})$ convergence to a family of adaptive gradient methods, including Adam. *Careful:* momentum benefit in Adam remains to be shown.
  - We must go beyond the standard assumptions (since SGD and Adagrad achieve lower bound $\Omega(\epsilon^{-4})$ oracle queries to reach ϵ-stationary point)
  - In the case of *linear regression*, and provided the stochastic noise level is sufficiently small, Stochastic Heavy Ball Momentum converges faster than SGD

Can et al 2019, Bollapragada et al 2022

Summary

Stochastic Gradient Descent (SGD) is the workhorse algorithm for large-scale optimization problems in machine learning.

The introduced stochasticity allows SGD to scale to very large problems, but SGD algorithm comes with many hyperparameters (such as step sizes).

Variations of SGD with automatic adaptive step-size updates were popular in practice but not understood theoretically in this context.

We gave a first theoretical proof of convergence for an adaptive gradient variation of SGD, showing the order-optimal convergence rate of SGD with carefully chosen step-sizes, but without needing to know the smoothness and noise parameters in advance.

Thank You!

Questions?

References:

1. M Faw, I Tziotis, C Caramanis, A Mokhtari, S Shakkottai, R Ward. The power of adaptivity in SGD: self-tuning step sizes with affine variance. Conference on Learning Theory, 2022.
2. R Ward, X. Wu, L. Bottou. Adagrad stepsizes: sharp convergence over nonconvex landscapes. The Journal of Machine Learning Research, 2018.