Motivation

- So far we’ve told you why SGD is “safe” :) 
- ...but Robbins-Monro is just a sufficient condition 
- ...then how to choose learning rates to achieve
  - fast convergence
  - better local optimum
Motivation (cont.)

- **idea 1**: search the best learning rate schedule
  - grid search, random search, cross validation
  - Bayesian optimization
- **But I’m lazy and I don’t want to spend too much time**

**Figure**: from [Bergstra and Bengio 2012]
Motivation (cont.)

- idea 2: let the learning rates adapt themselves
  - pre-conditioning with $|\text{diag}(H)|$ [Becker and LeCun 1988]
  - adaGrad [Duchi et al. 2010]
  - adaDelta [Zeiler 2012], RMSprop [Tieleman and Hinton 2012], ADAM [Kingma and Ba 2014]
  - Equilibrated SGD [Dauphin et al. 2015] (this year’s NIPS!)
- now we’ll talk a bit about online learning
  - a good tutorial [Shalev-Shwartz 2012]
  - milestone paper [Zinkevich 2003]
- ... and some practical comparisons
Batch learning often assumes:
- We’ve got the whole dataset $\mathcal{D}$
- the cost function $C(w; \mathcal{D}) = \mathbb{E}_{x \sim \mathcal{D}}[c(w; x)]$

SGD/mini-batch learning accelerate training in real time by
- processing one/a mini-batch of datapoint each iteration
- considering gradients with data point cost functions

$$\nabla C(w; \mathcal{D}) \approx \frac{1}{M} \sum_{m=1}^{M} \nabla c(w; x_m)$$
Let’s forget the cost function on the batch for a moment:

- online gradient descent (OGD):
  - each iteration $t$ we receive a loss $f_t(w)$
  - ...and a (noisy) (sub-)gradient $g_t \in \partial f_t(w)$
  - ...then we update the weights with $w \leftarrow w - \eta g_t$

- for SGD the received gradient is defined by

$$g_t = \frac{1}{M} \sum_{m=1}^{M} \nabla c(w; x_m)$$
From batch to online learning (cont.)
Online learning assumes each iteration $t$ we receive a loss $f_t(w)$ (in batch learning context $f_t(w) = c(w; x_t)$). Then we learn $w_{t+1}$ following some rules.

Performance is measured by regret

$$R^*_T = \sum_{t=1}^{T} f_t(w_t) - \inf_{w \in S} \sum_{t=1}^{T} f_t(w)$$  \hspace{1cm} (1)

General definition $R_T(u) = \sum_{t=1}^{T} [f_t(w_t) - f_t(u)]$

SGD $\iff$ “follow the regularized leader” with $L_2$ regularization
Follow the Leader (FTL)

$$w_{t+1} = \arg\min_{w \in S} \sum_{i=1}^{t} f_i(w)$$  \hspace{1cm} (2)

Lemma (upper bound of regret)

Let $w_1, w_2, \ldots$ be the sequence produced by FTL, then $\forall u \in S$, 

$$R_T(u) = \sum_{t=1}^{T} \left[ f_t(w_t) - f_t(u) \right] \leq \sum_{t=1}^{T} \left[ f_t(w_t) - f_t(w_{t+1}) \right].$$  \hspace{1cm} (3)
A game that fools FTL

Let $w \in S = [-1, 1]$ and the loss function at time $t$

$$f_t(w) = \begin{cases} 
-0.5w, & t = 1 \\
w, & t \text{ is even} \\
-w, & t > 1 \text{ and } t \text{ is odd} 
\end{cases}$$

FTL is easily fooled!
Follow the Regularized Leader (FTRL)

\[ w_{t+1} = \arg\min_{w \in S} \sum_{i=1}^{t} f_i(w) + \varphi(w) \] (4)

Lemma (upper bound of regret)

Let \( w_1, w_2, \ldots \) be the sequence produced by FTRL, then \( \forall u \in S \),

\[
\sum_{t=1}^{T} [f_t(w_t) - f_t(u)] \leq \varphi(u) - \varphi(w_1) + \sum_{t=1}^{T} [f_t(w_t) - f_t(w_{t+1})].
\]
Let’s assume the loss $f_t$ are convex functions:

$$\forall w, u \in S : \quad f_t(u) - f_t(w) \geq \langle u - w, g(w) \rangle, \forall g(w) \in \partial f_t(w)$$

use linearisation $\tilde{f}_t(w) = f_t(w_t) + \langle w, g(w) \rangle$ as a surrogate:

$$\sum_{t=1}^{T} f_t(w_t) - f_t(u) \leq \sum_{t=1}^{T} \langle w_t, g_t \rangle - \langle u, g_t \rangle, \quad \forall g_t \in \partial f_t(w_t)$$
Online Gradient Descent (OGD)

\[ w_{t+1} = w_t - \eta g_t, \quad g_t \in \partial f_t(w_t) \]  

From FTRL to online gradient descent:

- use linearisation as surrogate loss (upper bound LHS)
- use $L_2$ regularizer $\varphi(w) = \frac{1}{2\eta} \|w - w_1\|_2^2$
- apply FTRL regret bound to the surrogate loss

\[ w_{t+1} = \arg\min_{w \in S} \sum_{i=1}^{t} \langle w, g_i \rangle + \varphi(w) \]
Online Gradient Descent (OGD)

$$w_{t+1} = w_t - \eta g_t, \quad g_t \in \partial f_t(w_t)$$  \hspace{1cm} (5)

Theorem (regret bound for online gradient descent)

Assume we run online gradient descent on convex loss functions $f_1, f_2, \ldots, f_T$ with regularizer $\phi(w) = \frac{1}{2\eta}||w - w_1||^2_2$. Then for all $u \in S$,

$$R_T(u) \leq \frac{1}{2\eta}||u - w_1||^2_2 + \eta \sum_{t=1}^{T} ||g_t||^2_2, \quad g_t \in \partial f_t(w_t).$$
Theorem (regret bound for online gradient descent)

Assume we run online gradient descent on convex loss functions $f_1, f_2, \ldots, f_T$ with regularizer $\phi(w) = \frac{1}{2\eta} \| w - w_1 \|_2^2$. Then for all $u \in S$,

$$R_T(u) \leq \frac{1}{2\eta} \| u - w_1 \|_2^2 + \eta \sum_{t=1}^{T} \| g_t \|_2^2, \quad g_t \in \partial f_t(w_t).$$

- keep in mind for later discussion of adaGrad:
  - I want $\phi$ to be strongly convex wrt. (semi-)norm $\| \cdot \|$
  - ...then the $L_2$ norm will be changed to $\| \cdot \|_*$
  - ...and we use Hölder’s inequality for proof
Advanced adaptive learning rates

**adaGrad [Duchi et al. 2010]**

Use **proximal** terms $\psi_t$!

- $\psi_t$ is a strongly-convex function and
  
  $$B_{\psi_t}(w, w_t) = \psi_t(w) - \psi_t(w_t) - \langle \nabla \psi_t(w_t), w - w_t \rangle$$

- Primal-dual sub-gradient update:
  
  $$w_{t+1} = \arg\min_w \eta \langle w, \bar{g}_t \rangle + \eta \varphi(w) + \frac{1}{t} \psi_t(w). \quad (6)$$

- Proximal gradient/composite mirror descent:
  
  $$w_{t+1} = \arg\min_w \eta \langle w, g_t \rangle + \eta \varphi(w) + B_{\psi_t}(w, w_t). \quad (7)$$
**adaGrad [Duchi et al. 2010] (cont.)**

**adaGrad with diagonal matrices**

\[
G_t = \sum_{i=1}^{t} g_t g_t^T, \quad H_t = \delta I + \text{diag}(G_t^{1/2}).
\]

\[
w_{t+1} = \arg\min_{w \in S} \eta \langle w, \bar{g}_t \rangle + \eta \varphi(w) + \frac{1}{2} w^T H_t w.
\] (8)

\[
w_{t+1} = \arg\min_{w \in S} \eta \langle w, g_t \rangle + \eta \varphi(w) + \frac{1}{2} (w - w_t)^T H_t (w - w_t).
\] (9)

- To get adaGrad from online gradient descent:
  - add a proximal term or a Bregman divergence to the objective
  - adapt the proximal term through time:

\[
\psi_t(w) = \frac{1}{2} w^T H_t w
\]
example: composite mirror descent on $\mathcal{S} = \mathbb{R}^D$, with $\varphi(w) = 0$

$$w_{t+1} = \arg\min_w \eta\langle w, g_t \rangle + \frac{1}{2}(w - w_t)^T[\delta I + \text{diag}(G_t)](w - w_t)$$

$$\Rightarrow \quad w_{t+1} = w_t - \frac{\eta g_t}{\delta + \sqrt{\sum_{i=1}^t g_i^2}}$$

(10)
WHY???

- time-varying learning rate vs. time-varying proximal term
- I want the contribution of $||g_t||^2$ induced by $\psi_t$ be upper bounded by $||g_t||_2$:
  - define $|| \cdot ||_{\psi_t} = \sqrt{\langle \cdot, H_t \cdot \rangle}$
  - $\psi_t(w) = \frac{1}{2} w^T H_t w$ is strongly convex wrt. $|| \cdot ||_{\psi_t}$
  - a little math can show that
    $\sum_{t=1}^{T} ||g_t||_{\psi_t}^2 \leq 2 \sum_{d=1}^{D} ||g_1:T,d||_2^2$
Theorem (Thm. 5 in the paper)

Assume sequence \( \{w_t\} \) is generated by adaGrad using primal-dual update (8) with \( \delta \geq \max_t \|g_t\|_{\infty} \), then for any \( u \in S \),

\[
R_T(u) \leq \frac{\delta}{\eta} \|u\|_2^2 + \frac{1}{\eta} \|u\|_{\infty}^2 \sum_{d=1}^{D} \|g_{1:T,d}\|_2 + \eta \sum_{d=1}^{D} \|g_{1:T,d}\|_2.
\]

For \( \{w_t\} \) generated using composite mirror descent update (9),

\[
R_T(u) \leq \frac{1}{2\eta} \max_{t \leq T} \|u - w_t\|_2^2 \sum_{d=1}^{D} \|g_{1:T,d}\|_2 + \eta \sum_{d=1}^{D} \|g_{1:T,d}\|_2.
\]
Improving adaGrad

possible problems of adaGrad:
- global learning rate $\eta$ still need hand tuning
- sensitive to initial conditions

possible improving directions:
- use truncated sum / running average
- use (approximate) second-order information
- add momentum
- correct the bias of running average

existing methods:
- RMSprop [Tieleman and Hinton 2012]
- adaDelta [Zeiler 2012]
- ADAM [Kingma and Ba 2014]
RMSprop

\[ G_t = \rho G_{t-1} + (1 - \rho)g_t g_t^T, \quad H_t = (\delta I + \text{diag}(G_t))^{1/2}. \]

\[ w_{t+1} = \arg\min_w \eta \langle w, g_t \rangle + \frac{1}{2} (w - w_t)^T H_t (w - w_t). \]

\[ \Rightarrow w_{t+1} = w_t - \frac{\eta g_t}{\text{RMS}[g]_t}, \quad \text{RMS}[g]_t = \text{diag}(H_t) \quad (11) \]

- possible improving directions:
  - use truncated sum / running average
  - use (approximate) second-order information
  - add momentum
  - correct the bias of running average
Improving adaGrad (cont.)

\[ \Delta w \propto H^{-1} g \propto \frac{\partial f / \partial w}{\partial^2 f / \partial w^2} \]

\[ \Rightarrow \frac{\partial^2 f / \partial w^2}{\Delta w} \propto \frac{\partial f / \partial w}{\text{RMS}[g]_t} \]

\[ \Rightarrow \text{diag}(H) \approx \frac{\text{RMS}[\Delta w]_{t-1}}{\text{RMS}[\Delta w]_{t-1}} \]

- possible improving directions:
  - use truncated sum / running average
  - use (approximate) second-order information
  - add momentum
  - correct the bias of running average
Improving adaGrad (cont.)

\[
\text{diag}(H_t) = \frac{\text{RMS}[g]_t}{\text{RMS}[\Delta w]_{t-1}}
\]

\[
\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \langle \mathbf{w}, g_t \rangle + \frac{1}{2} (\mathbf{w} - \mathbf{w}_t)^T H_t (\mathbf{w} - \mathbf{w}_t).
\]

\[
\Rightarrow \quad \mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\text{RMS}[\Delta w]_{t-1}}{\text{RMS}[g]_t} g_t
\]

- possible improving directions:
  - use truncated sum / running average
  - use (approximate) second-order information
  - add momentum
  - correct the bias of running average
Improving adaGrad (cont.)

**ADAM**

\[
\begin{align*}
\mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t, \\
\mathbf{v}_t &= \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \mathbf{g}_t^2, \\
\hat{\mathbf{m}}_t &= \frac{\mathbf{m}_t}{1 - \beta_1^t}, \\
\hat{\mathbf{v}}_t &= \frac{\mathbf{v}_t}{1 - \beta_2^t}, \\
\mathbf{w}_{t+1} &= \mathbf{w}_t - \eta \frac{\hat{\mathbf{m}}_t}{\hat{\mathbf{v}}_t + \delta}.
\end{align*}
\]

- possible improving directions:
  - use truncated sum / running average
  - use (approximate) second-order information
  - add momentum
  - correct the bias of running average
**Demo time!**

wikipedia page “test functions for optimization”
https://en.wikipedia.org/wiki/Test_functions_for_optimization

<table>
<thead>
<tr>
<th>Name</th>
<th>Plot</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ackley's function:</td>
<td><img src="image" alt="Ackley's function" /></td>
<td>$f(x, y) = -20 \exp \left(-0.2\sqrt{0.5(x^2 + y^2)}\right)$</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Ackley's function" /></td>
<td>$- \exp \left(0.5(\cos(2\pi x) + \cos(2\pi y))\right) + e + 20$</td>
</tr>
<tr>
<td>Sphere function</td>
<td><img src="image" alt="Sphere function" /></td>
<td>$f(x) = \sum_{i=1}^{n} x_i^2$</td>
</tr>
<tr>
<td>Rosenbrock function</td>
<td><img src="image" alt="Rosenbrock function" /></td>
<td>$f(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right]$</td>
</tr>
</tbody>
</table>

Yingzhen Li (University of Cambridge)
Adaptive learning rates
Nov 26, 2015 23 / 27
Learn the learning rates

- idea 3: learn the (global) learning rates as well!

\[ w_{t+1} = w_t - \eta g_t \quad \Rightarrow \quad w_t = w(t, \eta, w_1, \{g_i\}_{i \leq t-1}) \]

- \( w_t \) is a function of \( \eta \)
- learn \( \eta \) (with gradient descent) by

\[ \eta = \text{argmin} \sum_{s \leq t} f_s(w(s, \eta)) \]

- Reverse-mode differentiation to back-track the learning process [Maclaurin et al. 2015]
- stochastic gradient descent to learn \( \eta \) on the fly [Massé and Ollivier 2015]
Summary

- Recent successes of machine learning rely on stochastic approximation and optimisation methods
- Theoretical analysis guarantees good behaviour
- Adaptive learning rates provides good performance in practice
- Future work will reduce labour on tuning hyper-parameters
References


