Adagrad, Adam and Online-to-Batch

Raunak Kumar

University of British Columbia

MLRG, 2017 Summer

July 4, 2017

Overview

Adagrad

Introduction

Let's begin by motivating Adagrad from 2 different viewpoints:

- Stochastic optimization (brief).
- Online convex optimization.

Stochastic Optimization

- We usually deal with sparse feature vectors.
- Infrequently occurring features are highly informative.
- Examples
  - Blue sky vs orange sky.
  - TF-IDF: Word w is important in document d if it occurs frequently in d but not in the entire corpus.

Stochastic Optimization

- Standard stochastic gradient algorithms follow a predetermined scheme.
- Ignore the characteristics of the observed data.
- **Idea**: Use a learning rate dependent on the frequency of the features.
  - Frequent feature $\rightarrow$ low learning rate.
  - Infrequent feature $\rightarrow$ high learning rate.

FTL

- In online convex optimization, Follow The Leader (FTL) chooses the next decision as the best one in hindsight

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \sum_{s=1}^{t} f_s(x).$$

- We've seen that $\text{regret}_T = O(T)$ because FTL is "unstable".
- **Idea**: Can we stabilize it by adding a *regularizer*?

Regularization Functions

- We will consider regularization functions $R : \mathcal{K} \to \mathbb{R}$.
- R is strongly-convex, smooth and twice-differentiable.
- Thus, $\nabla^2 R(x) \succ 0$.

RFTL

Algorithm 1 Regularized Follow The Leader

1: Input: Stepsize $\eta > 0$, regularization function R and a convex, compact set decision set $\mathcal{K}$
2: Let $x_1 = \arg\min_{x \in \mathcal{K}} \{R(x)\}$.
3: **for** $t = 1$ to T **do**
4: Predict x_t, observe f_t and let $\nabla_t = \nabla f_t(x_t)$.
5: Update

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \{\sum_{s=1}^{t} \nabla_s^T x + R(x)\}.$$

RFTL

Theorem (RFTL Regret)

1. $\forall u \in \mathcal{K}.\ regret_T \leq 2\eta \sum_{t=1}^{T} \|\nabla_t\|_t^{*2} + \frac{R(u) - R(x_1)}{\eta}.$
2. *If* $\forall t.\ \|\nabla_t\|_t^* \leq G_R$, *optimize* η *to get*

$$regret_T \leq 2 D_R G_R \sqrt{2T}.$$

3. *Can also write*

$$regret_T \leq \max_{u \in \mathcal{K}} \sqrt{2 \sum_t \|\nabla_t\|_t^{*2} B_R(u \| x_t)}.$$

Online Mirror Descent

Recall OMD from 3 weeks ago:

Algorithm 2 Online Mirror Descent (Agile version)

Input: Stepsize $\eta > 0$, regularization function R
Let y_1 s.t. $\nabla R(y_1) = 0$.
Let $x_1 = \arg\min_{x \in \mathcal{K}} B_R(x \| y_1)$.
for t = 1 to T **do**
    Predict x_t, observe f_t and let $\nabla_t = \nabla f_t(x_t)$.
    Update y_{t+1} as

$$\nabla R(y_{t+1}) = \nabla R(x_t) - \eta \nabla_t.$$

    Project according to B_R

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} B_R(x \| y_{t+1}).$$

Online Mirror Descent

- Suppose we choose $R(x) = \frac{1}{2}\|x - x_0\|_2^2$ for an arbitrary $x_0 \in \mathcal{K}$. What do we get?

Online Mirror Descent

- Suppose we choose $R(x) = \frac{1}{2}\|x - x_0\|_2^2$ for an arbitrary $x_0 \in \mathcal{K}$. What do we get?
- Online Gradient Descent!
- Regret bound

$$
\begin{aligned}
regret_T &\leq \frac{1}{\eta} D_R^2 + 2\eta \sum_t \|\nabla_t\|_t^{*2} \\
&\leq 2GD\sqrt{T}
\end{aligned}
$$

Online Mirror Descent

- $R(x) = \frac{1}{2}\|x - x_0\|_2^2$.
- $\nabla R(x) = x - x_0$.
- Update

$$y_t = x_{t-1} - \eta \nabla_{t-1}$$
$$x_t = \Pi_{\mathcal{K}}(y_t)$$

- Projection with respect to B_R (exercises 2, 3)

$$\begin{aligned} x_t &= \arg\min_{x \in K} B_R(x \| y_{t+1}) \\ &= \arg\min_{x \in K} \frac{1}{2}(x - y_{t+1})^T \nabla^2 R(z)(x - y_{t+1}) \\ &= \arg\min_{x \in \mathcal{K}} \frac{1}{2}\|x - y_{t+1}\|_2^2 \end{aligned}$$

Thus, OMD is equivalent to OGD with this choice of R.

Optimal Regularization

What have we seen so far?

- Using a regularization function to make FTL stable.
- OGD is a special case of OMD with $R(x) = \frac{1}{2}\|x - x_0\|_2^2$.
- The regularization function R affects our regret bound.

Question: Which regularization function should we choose to minimize regret?

Optimal Regularization

What have we seen so far?

- Using a regularization function to make FTL stable.
- OGD is a special case of OMD with $R(x) = \frac{1}{2}\|x - x_0\|_2^2$.
- The regularization function R affects our regret bound.

Question: Which regularization function should we choose to minimize regret? Depends on $\mathcal{K}$ and the cost functions.

So we'll learn the optimal regularizer online!

Preliminaries

- Goal: Learn a regularizer that adapts to the sequence of cost functions. In some sense, an optimal regularizer to use in hindsight.
- We will consider all strongly convex regularizers with a fixed and bounded Hessian in $\mathcal{H}$

$$\forall x \in \mathcal{K}.\ \nabla^2 R(x) = \nabla^2 \in \mathcal{H}$$

where

$$\mathcal{H} = \{X \in \mathbb{R}^{n \times n} : Tr(X) \leq 1,\ X \succeq 0\}$$

Adagrad

Algorithm 3 AdaGrad: Adaptive (sub)Gradient Method (Full Matrix Version)

1: Input: Stepsize $\eta > 0$, $\delta > 0$, $x_1 \in \mathcal{K}$.
2: Initialize: $S_0 = G_0 = 0$.
3: **for** t = 1 to T **do**
4: Predict x_t, observe loss f_t and let $\nabla_t = \nabla f_t(x_t)$.
5: Update

$$S_t = S_{t-1} + \nabla_t \nabla_t^T$$

$$G_t = S_t^{\frac{1}{2}} + \delta I$$

$$y_{t+1} = x_t - \eta G_t^{-1} \nabla_t$$

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \|x - y_{t+1}\|_{G_t}^2$$

Adagrad

Algorithm 4 AdaGrad: Adaptive (sub)Gradient Method (Diagonal Version)

1: Input: Stepsize $\eta > 0$, $\delta > 0$, $x_1 \in \mathcal{K}$.
2: Initialize: $S_0 = G_0 = 0$.
3: Initialize: $S_{1:0} = 0$.
4: **for** t = 1 to T **do**
5: Predict x_t, observe loss f_t and let $\nabla_t = \nabla f_t(x_t)$.
6: Update

$$\begin{aligned} S_{1:t} &= [S_{1:t-1} \nabla_t] \\ H_{t,i} &= \|S_{1:t,i}\|_2 \\ G_t &= diag(H_t) + \delta I \\ y_{t+1} &= x_t - \eta G_t^{-1} \nabla_t \\ x_{t+1} &= \arg\min_{x \in \mathcal{K}} \|x - y_{t+1}\|_{G_t}^2 \end{aligned}$$

Regret

We will show the following regret bound

Theorem

Let $\{x_t\}$ be the sequence of decisions defined by AdaGrad.
Let $D = \max_{u \in \mathcal{K}} \|u - x_1\|_2$. Choose $\delta = \frac{1}{2nD}$, $\eta = D$.
Then, for any $x^* \in \mathcal{K}$

$$regret_T(AdaGrad) \leq 2D \sqrt{\min_{H \in \mathcal{H}} \sum_t \|\nabla_t\|_H^{*2} + 1}$$

Optmial Regularizer

Proposition

(Exercise 11) Let $A \succeq 0$. Consider the following problem

$$\begin{aligned} \textit{minimize:} \quad & Tr(X^{-1}A) \\ \textit{subject to:} \quad & X \succeq 0 \\ & Tr(X) \leq 1 \end{aligned}$$

Then, minimizer is $X = A^{\frac{1}{2}} / Tr(A^{\frac{1}{2}})$ and the minimum objective value is $Tr^2(A^{\frac{1}{2}})$.

Optmial Regularizer

Corollary

1. $\sqrt{\min_{H\in\mathcal{H}}\sum_t \|\nabla_t\|_H^{*2}} = Tr(G_T - \delta I) = Tr(G_T) - \delta n.$
2. *Optimial regularizer:* $(G_T - \delta I)/Tr(G_T - \delta I)$.

Optmial Regularizer

Proof.

$$
\begin{aligned}
\|\nabla_t\|_H^{*2} &= \nabla_t^T H^{-1} \nabla_t \\
\sum_t \|\nabla_t\|_H^{*2} &= \sum_t \nabla_t^T H^{-1} \nabla_t \\
&= \sum_t Tr(\nabla_t^T H^{-1} \nabla_t) \\
&= \sum_t Tr(H^{-1} \nabla_t \nabla_t^T) \\
&= Tr(\sum_t H^{-1} \nabla_t \nabla_t^T) \\
&= Tr(H^{-1} \sum_t \nabla_t \nabla_t^T) \\
&= Tr(H^{-1} (G_T - \delta I)^2)
\end{aligned}
$$

□

Optmial Regularizer

Proof.

Thus, we can rewrite our objective as

$$\min_{H\in\mathcal{H}} \sum_t \|\nabla_t\|_H^{*2} = \min_{H\in\mathcal{H}} Tr(H^{-1}(G_T - \delta I)^2).$$

Plugging in $A = (G_T - \delta I)^2$ in the previous proposition yields the result. $\square$

Proof of Regret

Based on the corollary, it suffices to prove

Lemma

$$regret_T \leq 2DTr(G_T) = 2D\sqrt{\min_{H \in \mathcal{H}} \sum_t \|\nabla_t\|_H^{*2}} + 2nD\delta$$

Proof of Regret

Proof.

Using first order definition of convexity,

$$f(x_t) - f(x^*) \leq \nabla_t^T (x_t - x^*)$$

Thus, we can bound the total regret as

$$\begin{aligned} regret_T &= \sum_t f_t(x_t) - f_t(x^*) \\ &\leq \sum_t \nabla_t^T (x_t - x^*) \end{aligned}$$

Goal: Split this bound further into 2 terms and bound each of them separately. □

Proof of Regret

Proof.

Using definition of y_{t+1}, we get

$$\begin{aligned}
y_{t+1} - x^* &= x_t - x^* - \eta G_t^{-1} \nabla_t \\
G_t(y_{t+1} - x^*) &= G_t(x_t - x^*) - \eta \nabla_t \\
(y_{t+1} - x^*)^T G_t (y_{t+1} - x^*) &= (x_t - x^*)^T G_t (x_t - x^*) \\
&\quad - 2\eta \nabla_t^T (x_t - x^*) + \eta^2 \nabla_t^T G_t^{-1} \nabla_t
\end{aligned}$$

Using Pythagoras' Theorem

$$\|y_{t+1} - x^*\|_{G_t}^2 \geq \|x_{t+1} - x^*\|_{G_t}^2$$

in the above, we get

$$\nabla_t^T (x_t - x^*) \leq \frac{\eta}{2} \nabla_t^T G_t^{-1} \nabla_t + \frac{1}{2\eta} (\|x_t - x^*\|_{G_t}^2 - \|x_t - x^*\|_{G_t}^2)$$

Proof of Regret

Proof.

Thus, we have

$$\sum_t f_t(x_t) - f_t(x^*) \leq \frac{\eta}{2}\sum_{t=1}^{T} \nabla_t^T G_t^{-1} \nabla_t$$
$$+ \frac{1}{2\eta}\sum_{t=1}^{T}(x_t - x^*)^T(G_t - G_{t-1}(x_t - x^*)$$
$$+ \frac{\delta}{2\eta} D^2$$

Now, we proceed by bounding the first 2 terms in the inequality to get our result. □

Proof of Regret

Lemma

$$\sum_{t=1}^{T} \nabla_t^T G_t^{-1} \nabla_t \leq 2Tr(G_T)$$

Proof.

Proof Sketch: Induction and use that for $A \succeq B \succ 0$,

$$2Tr((A-B)^{\frac{1}{2}}) + Tr(A^{\frac{1}{2}}B) \leq 2Tr(A^{\frac{1}{2}})$$

□

Proof of Regret

Lemma

$$\sum_{t=1}^{T}(x_t - x^*)^T(G_t - G_{t-1})(x_t - x^*) \leq D^2 Tr(G_T)$$

.

Proof.

Proof sketch: Use that for $A \succeq 0$, $\lambda_{\max}(A) \leq Tr(A)$ and linearity of trace. $\square$

Proof of Regret

Proof.

FINALLY(!), we get

$$\sum_t f_t(x_t) - f_t(x^*) \leq \sum_t \nabla_t^T (x_t - x^*) \tag{1}$$

$$\leq \eta Tr(G_T) + \frac{1}{2\eta} D^2 Tr(G_T) + \frac{1}{2\eta} \delta D^2 \tag{2}$$

$$\leq 2DTr(G_T) \tag{3}$$

□

Adam

Key Idea

- Adagrad accumulates the squared gradients and G_T increases monotonically.
- The learning rate eventually becomes too small.
- Idea: Use a decaying average!
- Examples: Adadelta, RMSProp, etc.

Adam

- Adam: Adaptive Moment Estimation.
- It estimates the mean (first moment) and variance (second moment) of the gradients.
- Keeps track of
  - Exponentially decaying average of past squared gradients v_t
  - Exponentially decaying average of past gradients m_t

Adam

Algorithm 5 Adam: Adaptive Moment Estimation

1: Input: Stepsize $\eta > 0$, decay rates β_1, β_2, $\epsilon > 0$, and $x_1 \in \mathcal{K}$.
2: Intialize: First moment $m_0 = 0$, second moment $v_0 = 0$.
3: **for** t = 1 to T **do**
4: Predict x_t, observe loss f_t and let $\nabla_t = \nabla f_t(x_t)$.
5: Update

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1 - \beta_1)\nabla_t \\
v_t &= \beta_2 v_{t-1} + (1 - \beta_2)\nabla_t^2 \\
\hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \\
x_{t+1} &= x_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}
\end{aligned}
$$

Bias Correction

- Let $\nabla_t = \nabla f_t(x_t)$ and assume each ∇_t is a draw from some underlying distribution.
- We estimate its second raw moment as

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\nabla_t^2 = (1-\beta_2)\sum_{i=1}^{t} \beta_2^{t-i}\nabla_i^2$$

- We want to compare estimate $E[v_t]$ with true $E[\nabla_t^2]$ and correct for the mismatch (bias correction).

Bias Correction

$$E[v_t] = E[(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}\nabla_i^2]$$
$$= E[\nabla_t^2](1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i} + c$$
$$= E[\nabla_t^2](1-\beta_2^t) + c$$

- If $E[\nabla_i^2]$ is stationary, then $c = 0$ and we divide by $1-\beta_2^t$ to correct for the bias.
- Otherwise, β_1 should be chosen to assign small weights to gradients too far in the past.

Regret

Theorem

Assume

1. f_t *has bounded gradients:* $\|f_t(x)\|_2 \leq G$ *and* $\|f_t(x)\|_\infty \leq G_\infty$
2. *Distance between* x_t *generated by Adam is bounded:* $\forall n, m \in [T].\ \|x_n - x_m\|_2 \leq D$ *and* $\|x_n - x_m\|_\infty \leq D_\infty$
3. $\beta_1, \beta_2 \in [0, 1]$ *and* $\frac{\beta_1^2}{\sqrt{\beta_2}} < 1$
4. $\alpha_t = \frac{\alpha}{\sqrt{t}}$
5. $\beta_{1,t} = \beta_1 \lambda^{t-1}$ *for* $\lambda \in (0, 1)$

Then

$$\forall T \geq 1,\ regret_T = O(\sqrt{T})$$

.

Average Regret

Corollary

Assume

1. f_t *has bounded gradients:* $\|f_t(x)\|_2 \leq G$ *and* $\|f_t(x)\|_\infty \leq G_\infty$
2. *Distance between* x_t *generated by Adam is bounded:* $\forall n, m \in [T]. \ \|x_n - x_m\|_2 \leq D$ *and* $\|x_n - x_m\|_\infty \leq D_\infty$

Then

$$\forall T \geq 1, \ \frac{regret_T}{T} = O(\frac{1}{\sqrt{T}})$$

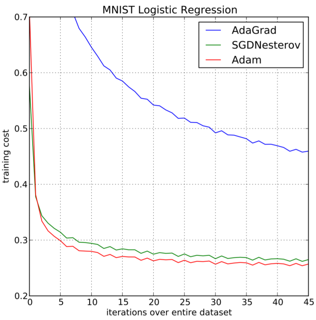
Comparison with Adagrad

- Choose $\beta_1 = 0$, $\beta_2 \to 1$ and stepsize $\eta_t = t^{\frac{-1}{2}}$.
- Then, $\lim_{\beta_2 \to 1} \hat{v}_t = \frac{1}{t} \sum_{t=1}^{T} \nabla_t^2$.
- Thus, the update is

$$x_{t+1} = x_t - \frac{\eta t^{\frac{-1}{2}}}{\sqrt{t^{-1} \sum_{t=1}^{T} \nabla_t^T}} \nabla_t$$

- Note that bias correction is important for this comparison.

Experiments



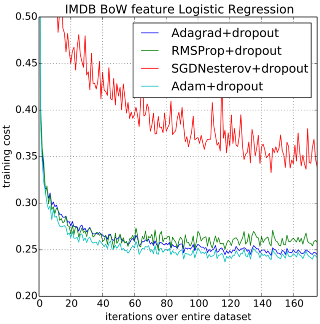


Figure 1: Logistic regression training negative log likelihood on MNIST images and IMDB movie reviews with 10,000 bag-of-words (BoW) feature vectors.

- Adagrad performs well in case sparse features and gradients.
- Adam performs well always?

Experiments

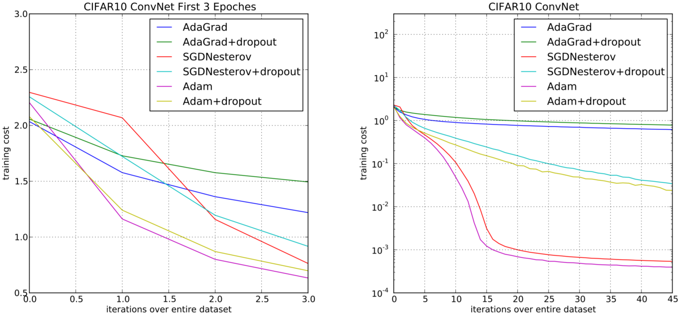


Figure 3: Convolutional neural networks training cost. (left) Training cost for the first three epochs. (right) Training cost over 45 epochs. CIFAR-10 with c64-c64-c128-1000 architecture.

- Adam reduces minibatch variance through first moment.
- This is important in the case of CNNs.

Online-To-Batch

Introduction

Given:

- Training set $S = ((x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T))$.
- Online learner $\mathcal{A}$ which constructs a sequence of models $h_1, h_2, \ldots, h_T$.

Goal: Construct a model h_S based on $\{h_t\}_{t=1}^{T}$.

Ideas

Possible conversion strategies:

1. Return the final model h_T.

Ideas

Possible conversion strategies:

1. Return the final model h_T.
2. Return the longest surviving model.

Ideas

Possible conversion strategies:

1. Return the final model h_T.
2. Return the longest surviving model.
3. Return h_t which performs best on a validation set.

Ideas

Possible conversion strategies:

1. Return the final model h_T.
2. Return the longest surviving model.
3. Return h_t which performs best on a validation set.
4. Randomly choose h_t from $\{h_t : t \in [T]\}$.

Ideas

Possible conversion strategies:

1. Return the final model h_T.
2. Return the longest surviving model.
3. Return h_t which performs best on a validation set.
4. Randomly choose h_t from $\{h_t : t \in [T]\}$.
5. Average over the models in some way.
6. Cutoff-Averaging.

References

Elad Hazan
Introduction to Online Convex Optimization
Foundations and Trends in Optimization, vol 2, no. 3 – 4, pp. 157 – 325, 2015.

Duchi, Hazan, Singer
Adaptive Subgradient Methods for Online Learning and Stochastic Optimization
Journal of Machine Learning Research, vol 12(Jul), pp. 2121 – 2159, 2011.

Sebastian Ruder
An Overview of Gradient Descent Optimization Algorithms.
arXiv 1609.04747

Kingma, Ba
Adam: A Method For Stochastic Optmization
ICLR, 2015.

Shivani Agarwal
Online to Batch Conversions
E0: 370 Statistical Learning Theory, 2013.

Shivani Agarwal
From Online to Batch Learning with Cutoff-Averaging
NIPS, 2008.

The End