The Marginal Value of Adaptive Gradient Methods in Machine Learning[1]

Jiayue Wan

Cornell University

April 11, 2019

[1]based on [Wilson et al., 2017]

Quiz

1. Which of the following is considered an adaptive gradient method? Select the first correct answer.
   A. SGD
   B. Nesterov's Accelerated Gradient method (NAG)
   C. Adam
   D. Heavy-ball method (HB)
2. Which of the following is **not** an observation of this paper?
   A. Adaptive methods have worse training performance than SGD.
   B. Adaptive methods generalize worse than SGD.
   C. Adaptive methods display faster initial progress on the training set than SGD.
   D. Performance of adaptive methods often plateaus quickly on the development set.

The Marginal Value of Adaptive Gradient Methods in Machine Learning[1]

Jiayue Wan

Cornell University

April 11, 2019

[1] based on [Wilson et al., 2017]

Overview

1. Background
   - Gradient methods
   - Adaptive gradient methods
2. Potential perils of adaptivity
   - Problem setup
   - Non-adaptive methods
   - Adaptive methods
3. Deep learning experiments
   - Hyper-parameter tuning
   - Convolutional neural network
   - Character-level language modeling
   - Constituency parsing
4. Conclusion & Discussion

Outline

Gradient Methods

- Stochastic gradient method

$$w_{k+1} = w_k - \alpha_k \tilde{\nabla} f(w_k)$$

where $\tilde{\nabla} f(w_k)$ is gradient computed on a batch of data.

Gradient Methods

- Stochastic gradient method

$$w_{k+1} = w_k - \alpha_k \tilde{\nabla} f(w_k)$$

  where $\tilde{\nabla} f(w_k)$ is gradient computed on a batch of data.
- Stochastic momentum methods

$$w_{k+1} = w_k - \alpha_k \tilde{\nabla} f(w_k + \gamma_k(w_k - w_{k-1})) + \beta_k(w_k - w_{k-1}).$$

Gradient Methods

- Stochastic gradient method

$$w_{k+1} = w_k - \alpha_k \tilde{\nabla} f(w_k)$$

  where $\tilde{\nabla} f(w_k)$ is gradient computed on a batch of data.

- Stochastic momentum methods

$$w_{k+1} = w_k - \alpha_k \tilde{\nabla} f(w_k + \gamma_k(w_k - w_{k-1})) + \beta_k(w_k - w_{k-1}).$$

- Examples include:
  - Polyak's heavy-ball method (HB): $\gamma_k = 0$;
  - Nesterov's Accelerated Gradient method (NAG): $\gamma_k = \beta_k$.

Adaptive Gradient Methods

- General form:

$$w_{k+1} = w_k - \alpha_k H_k^{-1} \tilde{\nabla} f(w_k + \gamma_k(w_k - w_{k-1})) \\ + \beta_k H_k^{-1} H_{k-1}(w_k - w_{k-1}).$$

Adaptive Gradient Methods

- General form:

$$w_{k+1} = w_k - \alpha_k H_k^{-1} \tilde{\nabla} f(w_k + \gamma_k(w_k - w_{k-1})) \\ + \beta_k H_k^{-1} H_{k-1}(w_k - w_{k-1}).$$

- The matrix H_k is usually defined as:

$$H_k = \operatorname{diag}\left(\left\{\sum_{i=1}^{k} \eta_i g_i \circ g_i\right\}^{1/2}\right)$$

where $g_k = \tilde{\nabla} f(w_k + \gamma_k(w_k - w_{k-1}))$.

Adaptive Gradient Methods

- General form:

$$w_{k+1} = w_k - \alpha_k H_k^{-1} \tilde{\nabla} f(w_k + \gamma_k(w_k - w_{k-1})) \\ + \beta_k H_k^{-1} H_{k-1}(w_k - w_{k-1}).$$

- The matrix H_k is usually defined as:

$$H_k = \operatorname{diag}\left(\left\{ \sum_{i=1}^{k} \eta_i g_i \circ g_i \right\}^{1/2} \right)$$

  where $g_k = \tilde{\nabla} f(w_k + \gamma_k(w_k - w_{k-1}))$.
- Examples:
  - AdaGrad, RMSProp: $\beta_k = \gamma_k = 0$;
  - Adam: $\gamma_k = 0$.

Adaptive Gradient Methods

Question: do adaptive methods generalize better than non-adaptive methods (SGD)?

Outline

Overview

Background

The Potential Perils of Adaptivity

Deep Learning Experiments

Conclusion

The Potential Perils of Adaptivity - Setup

Consider the binary least-squares classification problem, in which we aim to solve

$$\min_{w} R_S[w] := \frac{1}{2}\|Xw - y\|_2^2.$$

- Here, X is an $n \times d$ matrix of features and y is an n-dimensional vector of labels in $\{-1, 1\}$.

The Potential Perils of Adaptivity - Setup

Consider the binary least-squares classification problem, in which we aim to solve

$$\min_w R_S[w] := \frac{1}{2}\|Xw - y\|_2^2.$$

- Here, X is an $n \times d$ matrix of features and y is an n-dimensional vector of labels in $\{-1, 1\}$.
- When $d > n$, if there is a minimizer with loss 0 then there is an infinite number of global minimizers.

The Potential Perils of Adaptivity - Setup

Consider the binary least-squares classification problem, in which we aim to solve

$$\min_{w} R_S[w] := \frac{1}{2}\|Xw - y\|_2^2.$$

- Here, X is an $n \times d$ matrix of features and y is an n-dimensional vector of labels in $\{-1, 1\}$.
- When $d > n$, if there is a minimizer with loss 0 then there is an infinite number of global minimizers.

The Potential Perils of Adaptivity - Setup

Consider the binary least-squares classification problem, in which we aim to solve

$$\min_{w} R_S[w] := \frac{1}{2}\|Xw - y\|_2^2.$$

- Here, X is an $n \times d$ matrix of features and y is an n-dimensional vector of labels in $\{-1, 1\}$.
- When $d > n$, if there is a minimizer with loss 0 then there is an infinite number of global minimizers.

Question: what solution does an algorithm find and how well does it perform on unseen data?

Least-squares Classification: Non-adaptive Methods

Least-squares classification problem:

$$\min_{w} R_S[w] := \frac{1}{2}\|Xw - y\|_2^2.$$

- Any gradient or stochastic gradient of R_S must lie in the span of the rows of X.

Least-squares Classification: Non-adaptive Methods

Least-squares classification problem:

$$\min_{w} R_S[w] := \frac{1}{2}\|Xw - y\|_2^2.$$

- Any gradient or stochastic gradient of R_S must lie in the span of the rows of X.
- Consider any method that is initialized in the row span of X (for instance, $w = 0$).

Least-squares Classification: Non-adaptive Methods

Least-squares classification problem:

$$\min_w R_S[w] := \frac{1}{2}\|Xw - y\|_2^2.$$

- Any gradient or stochastic gradient of R_S must lie in the span of the rows of X.
- Consider any method that is initialized in the row span of X (for instance, $w = 0$).
- Linear combination of gradients, stochastic gradients, and previous iterates must also lie in the row span of X.

Least-squares Classification: Non-adaptive Methods

Least-squares classification problem:

$$\min_w R_S[w] := \frac{1}{2}\|Xw - y\|_2^2.$$

- Any gradient or stochastic gradient of R_S must lie in the span of the rows of X.
- Consider any method that is initialized in the row span of X (for instance, $w = 0$).
- Linear combination of gradients, stochastic gradients, and previous iterates must also lie in the row span of X.
- The **unique** solution that lies in the row span of X is the solution with minimum Euclidean norm.

Least-squares Classification: Non-adaptive Methods

Least-squares classification problem:

$$\min_{w} R_S[w] := \frac{1}{2}\|Xw - y\|_2^2.$$

- Any gradient or stochastic gradient of R_S must lie in the span of the rows of X.
- Consider any method that is initialized in the row span of X (for instance, $w = 0$).
- Linear combination of gradients, stochastic gradients, and previous iterates must also lie in the row span of X.
- The **unique** solution that lies in the row span of X is the solution with minimum Euclidean norm.
- Thus, we denote $w^{\text{SGD}} = X^T(XX^T)^{-1}y$.

Least-squares Classification: Adaptive Methods

Lemma

Suppose there exists a scalar c such that $X \mathit{sign}(X^T y) = cy$. Then, when initialized at $w_0 = 0$, AdaGrad, Adam and RMSProp all converge to the unique solution $w \propto \mathit{sign}(X^T y)$.

Least-squares Classification: Adaptive Methods

Lemma

Suppose there exists a scalar c such that $X \mathit{sign}(X^T y) = cy$. Then, when initialized at $w_0 = 0$, AdaGrad, Adam and RMSProp all converge to the unique solution $w \propto \mathit{sign}(X^T y)$.

Proof by induction.

- We show that $w_k = \lambda_k \text{sign}(X^T y)$ for some scalar λ_k.

Least-squares Classification: Adaptive Methods

Lemma

Suppose there exists a scalar c such that $X\mathit{sign}(X^T y) = cy$. Then, when initialized at $w_0 = 0$, AdaGrad, Adam and RMSProp all converge to the unique solution $w \propto \mathit{sign}(X^T y)$.

Proof by induction.

- We show that $w_k = \lambda_k \text{sign}(X^T y)$ for some scalar λ_k.
- Initial point $w_0 = 0$ satisfies the assertion with $\lambda_0 = 0$.

Least-squares Classification: Adaptive Methods

Lemma

Suppose there exists a scalar c such that $X \mathrm{sign}(X^T y) = cy$. Then, when initialized at $w_0 = 0$, AdaGrad, Adam and RMSProp all converge to the unique solution $w \propto \mathrm{sign}(X^T y)$.

Proof by induction.

- We show that $w_k = \lambda_k \mathrm{sign}(X^T y)$ for some scalar λ_k.
- Initial point $w_0 = 0$ satisfies the assertion with $\lambda_0 = 0$.
- Assume the assertion holds for all $t \leq k$. Observe that

$$
\begin{aligned}
&\nabla R_S(w_k + \gamma_k(w_k - w_{k-1})) \\
&= X^T(X(w_k + \gamma_k(w_k - w_{k-1})) - y) \\
&= X^T((\lambda_k + \gamma_k(\lambda_k - \lambda_{k-1}))X\mathrm{sign}(X^T y) - y) \\
&= ((\lambda_k + \gamma_k(\lambda_k - \lambda_{k-1}))c - 1)X^T y \\
&= \mu_k X^T y.
\end{aligned}
$$

Least-squares Classification: Adaptive Methods

Proof by induction (con't).

- Letting $g_k = \nabla R_S(w_k + \gamma_k(w_k - w_{k-1}))$, we have

$$H_k = \mathrm{diag}\left(\left\{\sum_{s=1}^{k} \eta_s g_s \circ g_s\right\}^{1/2}\right) = \mathrm{diag}\left(\left\{\sum_{s=1}^{k} \eta_s \mu_s^2\right\}^{1/2} |X^T y|\right)$$
$$= \nu_k \mathrm{diag}(|X^T y|).$$

$\square$

Least-squares Classification: Adaptive Methods

Proof by induction (con't).

- Letting $g_k = \nabla R_S(w_k + \gamma_k(w_k - w_{k-1}))$, we have

$$H_k = \text{diag}\left(\left\{\sum_{s=1}^{k} \eta_s g_s \circ g_s\right\}^{1/2}\right) = \text{diag}\left(\left\{\sum_{s=1}^{k} \eta_s \mu_s^2\right\}^{1/2} |X^T y|\right)$$
$$= \nu_k \text{diag}(|X^T y|).$$

- In sum, we have that

$$w_{k+1} = \left(\lambda_k - \frac{\alpha_k \mu_k}{\nu_k} + \frac{\beta_k \nu_{k-1}}{\nu_k}(\lambda_k - \lambda_{k-1})\right) \text{sign}(X^T y)$$

proving the claim.

$\square$

Least-squares Classification: Adaptive Methods

We construct a generative model where AdaGrad fails to find a solution that generalizes. For $i = 1, 2, \cdots, n$, sample the label y_i to be 1 with probability p and -1 with probability $1 - p$ for some $p > \frac{1}{2}$. Let x_i be an infinite dimensional vector with entries

$$x_{ij} = \begin{cases} y_i & j = 1 \\ 1 & j = 2, 3 \\ 1 & j = 4 + 5(i-1), \cdots, 4 + 5(i-1) + 2(1 - y_i) \\ 0 & \text{otherwise} \end{cases}.$$

If the class label is 1, there is 1 unique feature. If the class label is -1, there are 5 unique features. The only discriminative feature useful for classifying data outside the training set is the first feature.

Least-squares Classification: Adaptive Methods

Consider the AdaGrad solution for $\min_w R_S[w] := \frac{1}{2}\|Xw - y\|_2^2$. Notice that

$$\operatorname{sign}((X^T y)_j) = \begin{cases} 1 & j = 1 \\ 1 & j = 2, 3 \\ y_j & j > 3 \text{ and } x_{\lfloor \frac{j+1}{5} \rfloor, j} = 1 \\ 0 & \text{otherwise} \end{cases}.$$

Thus, we have that $\langle \operatorname{sign}(X^T y), x_i \rangle = y_i + 2 + y_i(3 - 2y_i) = 4y_i$. Hence, by Lemma the AdaGrad solution $w^{\text{ada}} \propto \operatorname{sign}(X^T y)$, i.e. w^{ada} has all of its components equal to $\pm\tau$ for some $\tau > 0$. Now, for a new data point, x^{test}, we have

$$\langle w^{\text{ada}}, x^{\text{test}} \rangle = \tau(y^{\text{test}} + 2) > 0.$$

Therefore, the AdaGrad solution will label all unseen data as a positive example!

Least-squares Classification: Adaptive Methods

Consider the minimum 2-norm solution. We know that the optimal solution has the form $w^{\text{SGD}} = X^T\alpha$ where $\alpha = K^{-1}y$ and $K = XX^T$. Note that

$$K_{ij} = \begin{cases} 4 & i = j, y_i = 0 \\ 8 & i = j, y_i = -1 \\ 3 & i \neq j, y_i y_j = 1 \\ 1 & i \neq j, y_i y_j = -1 \end{cases}.$$

Positing that $\alpha_i = \alpha_+$ if $y_i = 1$ and $\alpha_i = \alpha_-$ if $y_i = -1$, we find

$$\begin{aligned} (3n_+ + 1)\alpha_+ + n_-\alpha_- &= 1 \\ n_+\alpha_+ + (3n_- + 3)\alpha_- &= -1. \end{aligned}$$

Least-squares Classification: Adaptive Methods

Solving the system equations yields

$$\alpha_+ = \frac{4n_- + 3}{9n_+ + 3n_- + 8n_+n_- + 3}, \alpha_- = -\frac{4n_+ + 1}{9n_+ + 3n_- + 8n_+n_- + 3}.$$

For a new data point, we have

$$\langle w^{\text{SGD}}, x^{\text{test}} \rangle = y^{\text{test}}(n_+\alpha_+ - n_-\alpha_-) + 2(n_+\alpha_+ + n_-\alpha_-).$$

Whenever $n_+ > n_-/3$, the SGD solution makes no errors.

Question

Does this result generalize to other machine learning problems?

Outline

Deep Learning Experiments

Table 1: Summary of the models used for experiments.

Name	Network type	Architecture	Dataset	Framework
C1	Deep Convolutional	`cifar.torch`	CIFAR-10	Torch
L1	2-Layer LSTM	`torch-rnn`	War & Peace	Torch
L2	2-Layer LSTM + Feedforward	`span-parser`	Penn Treebank	DyNet
L3	3-Layer LSTM	`emnlp2016`	Penn Treebank	Tensorflow

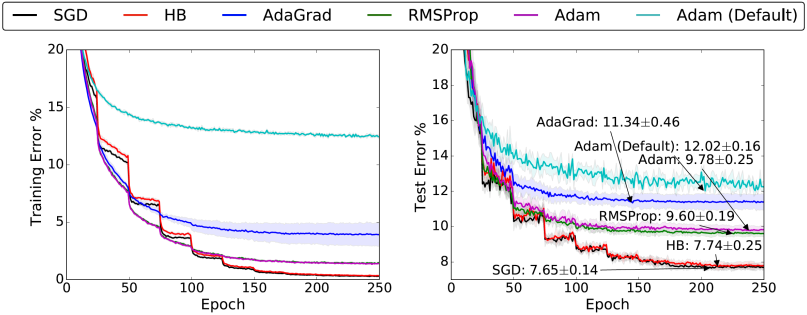
Comparison among:

- non-adaptive methods: SGD and HB
- adaptive methods: AdaGrad, RMSProp and Adam

Hyperparameter tuning:

- step size: logarithmically-space grid of step sizes
- step size decay: development-based decay scheme, fixed frequency decay scheme

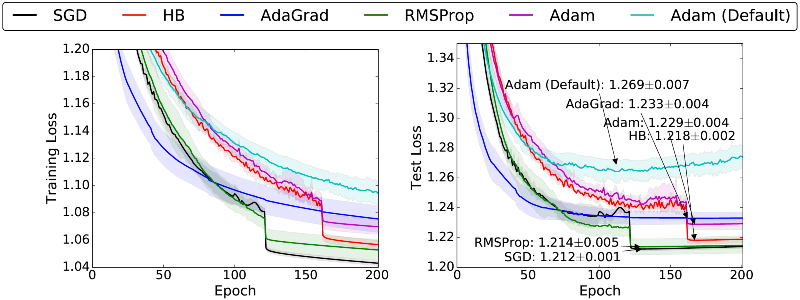
Convolutional Neural Network



(a) CIFAR-10 (Train) **(b)** CIFAR-10 (Test)

Figure 1: Training (left) and top-1 test error (right) on CIFAR-10. The annotations indicate where the best performance is attained for each method. The shading represents $\pm$ one standard deviation computed across five runs from random initial starting points. In all cases, adaptive methods are performing worse on both train and test than non-adaptive methods.

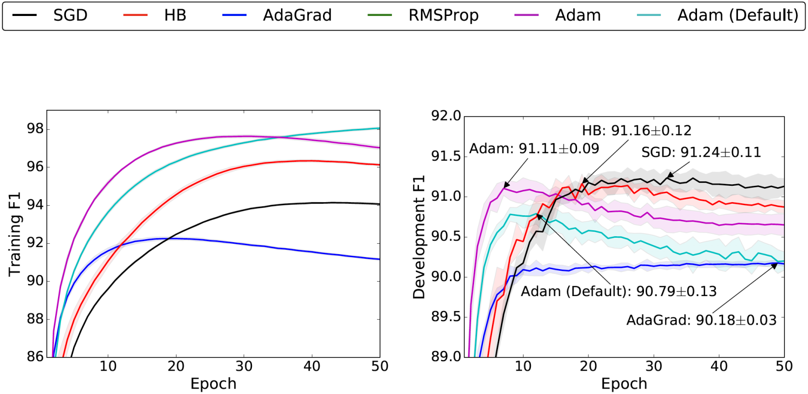
Character-level Language Modeling



(a) War and Peace (Training Set) **(b)** War and Peace (Test Set)

Figure 2: Performance curves on the training data (left) and the development/test data (right) for character-level language modeling.

Constituency Parsing - Discriminative Model



(c) Discriminative Parsing (Training Set) **(d)** Discriminative Parsing (Development Set)

Figure 3: Performance curves on the training data (left) and the development/test data (right) for discriminative model of constituency parsing.

Constituency Parsing - Generative Model

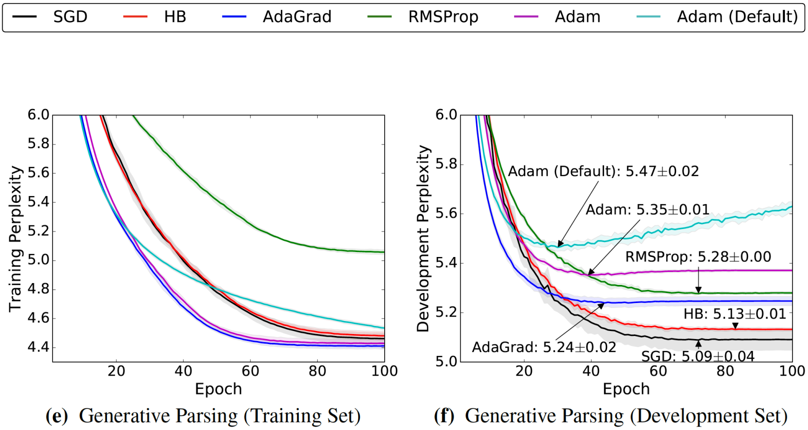


Figure 4: Performance curves on the training data (left) and the development/test data (right) for generative model of constituency parsing.

Outline

Conclusion and Discussion

Primary findings:

- Adaptive methods find solutions that generalize worse than those found by non-adaptive methods.
- Even when the adaptive methods achieve the *same training loss or lower* than non-adaptive methods, the development or test performance is worse.
- Adaptive methods often display faster initial progress on the training set, but their performance quickly plateaus on the development set.
- Though conventional wisdom suggests that Adam does not require tuning, we find that tuning the initial learning rate and decay scheme for Adam yields significant improvements over its default settings in all cases.

Conclusion and Discussion

Question

Why does Adam algorithm remain incredibly popular?

TITLE	CITED BY	YEAR
Adam: A Method for Stochastic Optimization DP Kingma, J Ba Proceedings of the 3rd International Conference on Learning Representations …	20177	2014

Thank you!