The Marginal Value of Adaptive Gradient Methods in Machine Learning

A. Wilson, R. Roelofs, M. Stern, N. Srebro, B. Recht

April 3, 2018

as understood by Kelly Zhang

Disclaimer: This presentation is made up of my understandings and intuitions about this paper - as a complete newbie to the land of optimization... Questions and feedback welcome!

Should be fun though!

1 Understanding Deep Learning Requires Rethinking Generalization

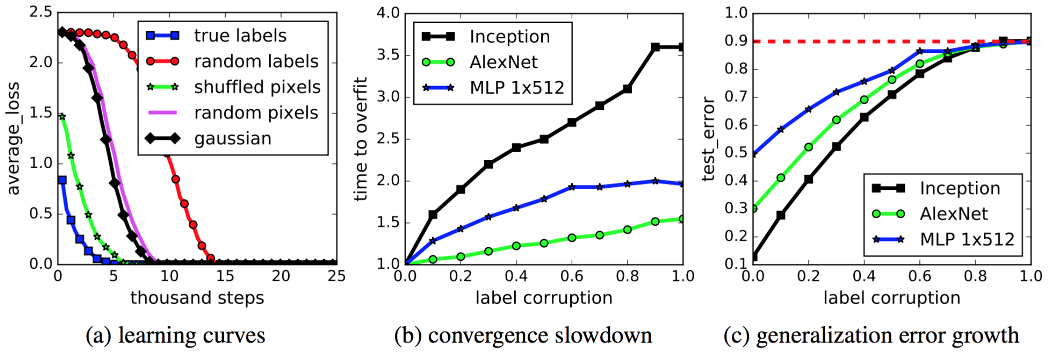


Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

Some relevant takeaways:

- The models we're training are over-parametrized, even with regularization (at least for image classification)
- Faster convergence does seem to lead to less overfitting / better generalization (similar argument as that for early-stopping as regularization)

2 Generalization in Over-Parameterized Settings

- When models are over-parametrized, there are multiple solutions that minimize training error
- Still, some minima can be better than others in terms of generalization

  In general, the simplest model that explains the data has better generalization. Thus in the over-parameterized setting, "simplicity" is favored.

  I am not exactly sure how "simplicity" is measured generally, but we will discuss this in a specific example.
- This paper

  1. Proves in a simple model and problem setting where adaptive gradient methods fail to generalize as well as regular SGD (even when both methods achieve zero training error)
  2. Empirically shows the inferiority of adaptive gradient methods in terms of generalization on several image and text tasks

3 Gradient Descent!

3.1 SGD

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

where α is the learning rate

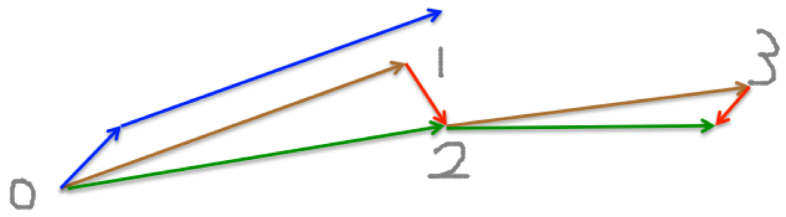
3.2 Momentum

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

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

A picture of the Nesterov method

- First make a big jump in the direction of the previous accumulated gradient.
- Then measure the gradient where you end up and make a correction.



brown vector = jump, red vector = correction, green vector = accumulated gradient

blue vectors = standard momentum

- Take step in direction of previous gradient: $\beta_k(w_k - w_{k-1})$
- Correction: $w_k - \alpha \tilde{\nabla} f(w_k + \gamma_k(w_k - w_{k-1}))$

3.3 Adaptive Gradient Methods

What are they? How do they relate to each other?

3.3.1 Newton's Method

A procedure way to numerically find the roots or zeros of a function.

Newton's Rule (single variable):

$$f(x_{t+1}) = 0 = f(x_t) + f'(x_t)(x_t - x_{t+1})$$

$$x_{t+1} = x_t - \frac{f(x_t)}{f'(x_t)}$$

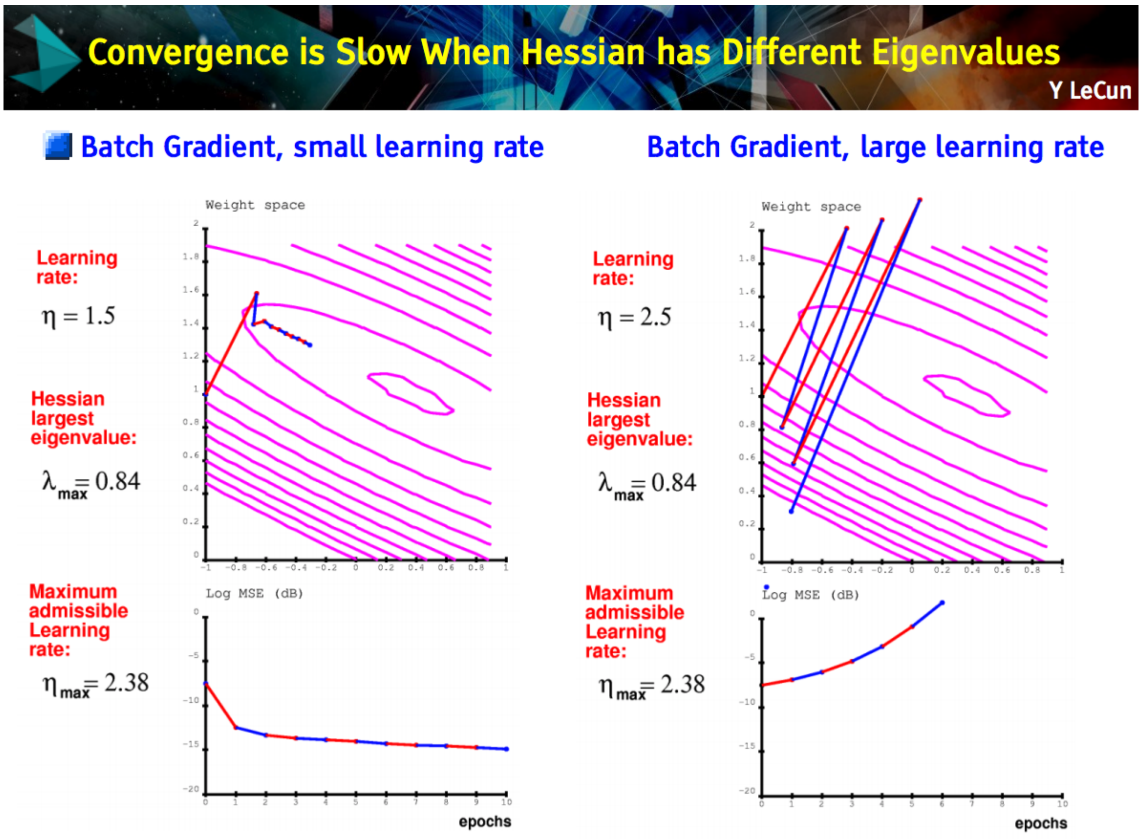
Newton's method can easily be adapted to help us find critical points of a function, g, by simply setting $f(x) = g'(x)$:

$$x_{t+1} = x_t - \frac{g'(x_t)}{g''(x_t)}$$

Multivariate Newton's method:

$$w_{k+1} := w_k - \alpha H_k^{-1} \nabla f(w_k)$$

where H_k is the Hessian of $f(w_k)$.



3.3.2 AdaGrad (by Duchi, Hazan, and Singer, 2011)

$$w_{k+1} := w_k - \alpha H_k^{-1} \tilde{\nabla} f(w_k)$$
$$G_{k+1} := G_k + D_k$$
$$(H = \sqrt{G})$$

Where D_k is a diagonal matrix (with $\tilde{\nabla} f(w_k)$ squared for each entry):

$$D_k := \mathrm{Diag}\{\tilde{\nabla} f(w_k)^2\}$$

So G is linear combinations of the square of past gradient components.

Note:
Different learning rate for each parameter. Larger updates for "infrequent" parameters (small square of gradient over updates) and small updates for "frequent" parameters (large square of gradient over updates).

This feature of adaptive gradient methods is what can lead it give undue influence to "spurious" features.

3.3.3 RMSProp (by Hinton, unpublished)

Same as AdaGrad, but with different weighting scheme:

$$w_{k+1} := w_k - \alpha H_k^{-1} \tilde{\nabla} f(w_k)$$
$$G_{k+1} := \beta_2 G_k + (1 - \beta_2) D_k$$
$$(H = \sqrt{G})$$

3.3.4 ADAM (by Kingma and Ba, 2014)

$$w_{k+1} := w_k - \alpha \hat{H}_{k+1}^{-1} \hat{g}_{k+1}$$
$$g_{k+1} := \beta_1 g_k + (1 - \beta_1) \tilde{\nabla} f(w_k)$$
$$G_{k+1} := \beta_2 G_k + (1 - \beta_2) D_k$$
$$\hat{g}_{k+1} = \frac{g_{k+1}}{1 - \beta_1^{k+1}}$$
$$\hat{G}_{k+1} = \frac{G_{k+1}}{1 - \beta_2^{k+1}}$$
$$(\hat{H} = \sqrt{\hat{G}})$$

(Note in β_1^k, k denotes β_2 to the power k.)

4 Simple Setting for Analysis: Gradient Descent for Linear Regression

$$\min_w R[w] := ||Xw - y||^2$$

with

- X is a $n \times d$ matrix (n for data points; d for features)
- y is a n-dimensional vector of labels in $\{-1, 1\}$

We assume the problem is over-parameterized / over-determined, so $d > n$ (number of features $>$ number of data points).

Thus if there is a w that achieves loss 0, then there are infinitely many global minimizers (solution not unique, since over-complete).

4.1 Calculating the Gradient

$$\begin{gathered} R = (Xw - y)^T(Xw - y) = \\ (Xw)^T(Xw - y) - y^T(Xw - y) = \\ (Xw)^T(Xw) - (Xw)^T y - y^T(Xw) + y^T y \\ w^T X^T X w - 2y^T X w + y^T y \\ w^T X^T X w - 2(X^T y)^T w + y^T y \end{gathered}$$

Using

$$\begin{gathered} \frac{d}{dw} y^T w = \frac{d}{dw} w^T y = y \\ \frac{d}{dw} w^T X w = (X + X^T) w \end{gathered}$$

We get:

$$\frac{d}{dw} R = (X^T X + X X^T) w - 2X^T y = 2X^T X w - 2X^T y$$

Thus, $\frac{d}{dw} R \in \text{span}\{\text{ rows of } X\ \}$, since for arbitrary vector a, $X^T a$ for is a linear combination of the rows of X.

4.2 Non-adaptive methods

Claim: If w is initialized in the row span of X (e.g. $w = 0$), and uses only linear combination of gradients / stochastic gradients, w_t must also lie in the row span of X, for all updates t.

Justification: If we start in the span of the rows of X, and add linear combinations of the rows of X, we will still end up in the span of X's rows.

Claim: The unique solution that lies in the row span of X also happens to be the solution with **minimum Euclidian norm**, so

$$w^{SGD} = X^T(XX^T)^{-1}y$$

Justification: The idea is that since the system is underdetermined, we can think of the problem of finding vector as w, as finding the vector in a very high dimensional space, that projects to another vector y in a lower dimensional space (the projection function is determined by X).

$$\min_w R[w] := ||Xw - y||^2$$

Important Idea! The w with the minimum norm in this case generalizes the best.

4.3 Adaptive methods

Claim: Suppose $X^T y$ has no components equal to 0, and there exists a scalar c such that $X sign(X^T y) = cy$.

Then when initialized at $w_0 = 0$, AdaGrad, Adam, and RMSProp all converge to the solution $w \propto sign(X^T y)$.

Justification: We want to show for all iterations k,

$$w_k = \lambda_k sign(X^T y)$$

We prove by induction.
Base case We know the assertion holds for $w_0 = 0$, as $\lambda_0 = 0$.
Inductive step Now, given that for some λ_{k-1},

$$\nabla R(w_{k-1}) = \lambda_{k-1} sign(X^T y)$$

We can show that for some μ_k, (details in paper)

$$\nabla R(w_k) = \mu_k X^T y$$

The problematic part - what differentiates the adaptive gradient methods - is the H_k term. Letting $g_k = \nabla R(w_k)$, then H_k is diagonal matrix such that

$$H_k = \text{diag}\left\{\sqrt{g_k^2}\right\} = \text{diag}\left\{\sqrt{\mu_k}|X^T y|\right\} =: v_k \text{diag}(|X^T y|)$$

$$\begin{aligned} w_{k+1} = w_k - \alpha_k H_k^{-1} \tilde{\nabla} f(w_k + \gamma_k(w_k - w_{k-1})) + \beta_k H_k^{-1} H_{k-1}(w_k - w_{k-1}) \\ = \lambda_k \text{sign}(X^T y) - \\ \alpha_k (v_k |X^T y|)^{-1} \mu_k X^T y + \\ \beta_k (v_k |X^T y|)^{-1} v_{k-1} |X^T y| (\lambda_k - \lambda_{k-1}) \text{sign}(X^T y) \\ = \left\{\lambda_k - \frac{\alpha_k \mu_k}{v_k} + \frac{\beta_k v_{k-1}}{v_k}(\lambda_k - \lambda_{k-1})\right\} \text{sign}(X^T y) \end{aligned}$$

4.4 Adaptivity Can Overfit

4.4.1 Problem Setting

Infinite dimensional, n examples. Labels $y^{(i)} \in \{-1, 1\}$, assigned 1 with probability $p > \frac{1}{2}$.

$$x_j^{(i)} = \begin{cases} y^{(i)}, & j = 1, \\ 1, & j = 2, 3, \\ 1, & j = 4 + 5(i-1), 4 + 5(i-1) + 1, ..., 4 + 5(i-1) + 2(1 - y^{(i)}) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

- Only the first feature is useful (class label)
- Next two features are always 1
- All other features are unique for each n

  If there label is 1, 1 unique features

  If there label is -1, 5 unique features

4.4.2 SGD Solution

SGD will find minimum norm solution –> no generalization problem.

Let P and N by the set of positive and negative examples respectively. α_+ and α_- are non-negative constants (can be solved for in closed form; see paper).

$$w^{sgd} = \sum_{i \in P} \alpha_+ x_i - \sum_{j \in N} \alpha_- x_j$$

4.4.3 AdaGrad Solution

$$u = X^T y = w$$

$$b = \sum_{i=1}^{n} y^{(-i)}$$

$$u_j = \begin{cases} n, & j = 1, \\ b, & j = 2, 3, \\ y_j, & \text{if } j > 3 \text{ and } x_j = 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

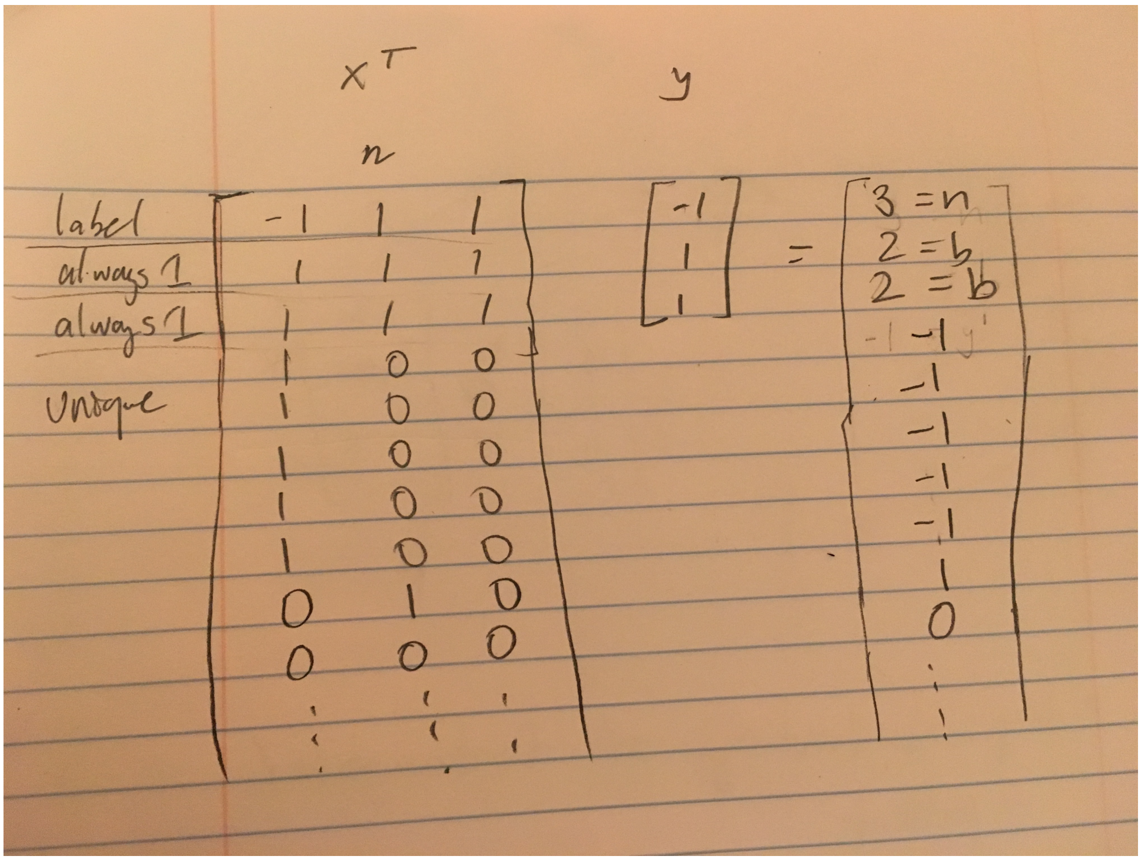
Satisfies conditions of lemma since $< u, x_i >= y_i + 2 + y_i(3 - 2y_i) = 4y_i$; so

$$w^{ada} \propto \text{sign}(u)$$

So all components of w^{ada} either zero or $\pm\tau$ for some constant τ.

For a new point though, x^{test}, the only features that are nonzero for both x^{test} and w^{ada} are the first three. Thus, the solution will label all unseen data as being in the positive class:

$$< w^{ada}, x^{test} >= \tau(y^{test} + 2) > 0$$



$$X^T \qquad\qquad y$$

$$\begin{array}{l} \\ \text{label} \\ \text{always } 1 \\ \text{always } 1 \\ \\ \text{Unique} \\ \\ \\ \\ \\ \\ \\ \end{array}
\begin{array}{c} n \\ \begin{bmatrix} -1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \end{bmatrix} \end{array}
\begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 = n \\ 2 = b \\ 2 = b \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

5 Empirical Investigations

5.1 Methodology

Strategy: Hold the model, the training objective, and epoch training budget constant, varying only the optimization strategy and measure the generalization error; 5 runs each.

5.1.1 Learning Rate Tuning

- Initial learning rate: grid search on log scale
- LR decay schedule: development-based decay, fixed frequency decay

5.1.2 Training Objective and Models

- CIFAR10

  VGG+BN+Dropout

  SGD: $7.65 \pm 0.14\%$ test error

  RMSProp: $9.60 \pm 0.19\%$ test error

- Character-Level Language Model on War and Peace

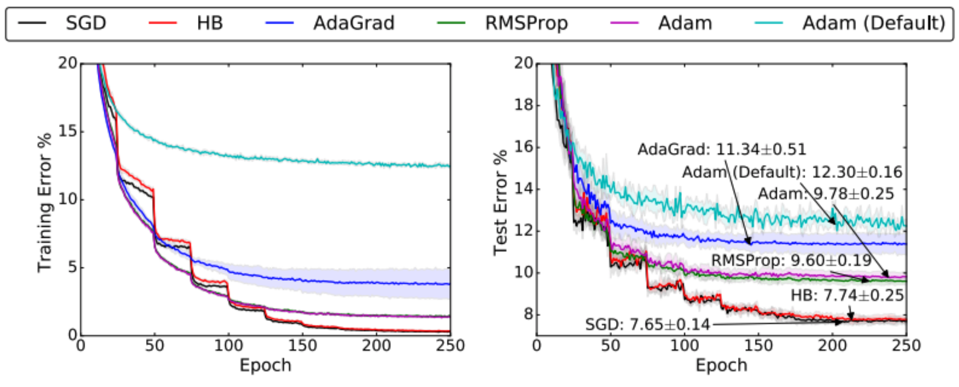
  LSTM model

  SGD: 1.212 ± 0.001 test loss

  RMSProp: test loss very close to SGD's

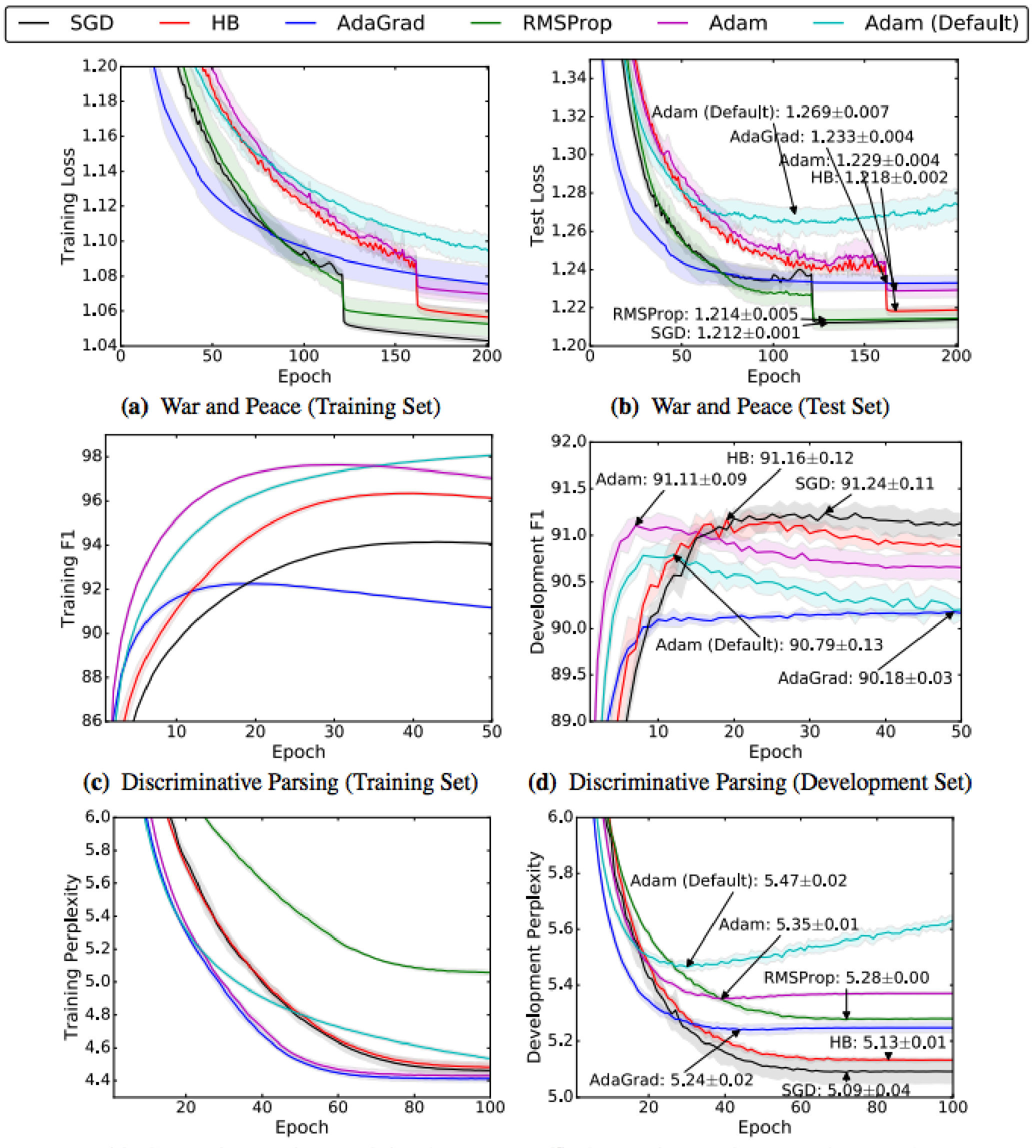
- Constituency Parsing

Often adaptive gradient methods outperform SGD early on in training, but their test error of AdaGrad's rate of improvement flatlines earliest.



(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 ± 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.



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

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

(e) Generative Parsing (Training Set) **(f)** Generative Parsing (Development Set)

Figure 2: Performance curves on the training data (left) and the development/test data (right) for three experiments on natural language tasks. The annotations indicate where the best performance is attained for each method. The shading represents one standard deviation computed across five runs from random initial starting points.

6 Take-Aways

- In over-parameterized settings, using adaptive gradient methods can give undue influence to spurious features and lead to minima that lead to worse generalization.
- Jury is out on whether adaptive gradient methods are beneficial for training GANs.
- The difference in generalization (empirically) using SGD vs. Adaptive gradient methods is kind of small, but significant.
- Exploring what generalization and regularization in the context of deep learning and over-parameterized settings is very interesting.