Adaptive Gradient Methods And Beyond

Liangchen Luo

Peking University, Beijing
luolc.witty@gmail.com

March, 2019

From SGD to Adam

- SGD (Robbins & Monro, 1951)
  - + Momentum (Qian, 1999)
  - + Nesterov (Nesterov, 1983)
- AdaGrad (Duchi et al., 2011)
- RMSprop (Tieleman & Hinton, 2012)
- Adam (Kingma & Lei Ba, 2015)

Stochastic Gradient Decent (Robbins & Monro, 1951)

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$$

H. Robbins, S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 1951.

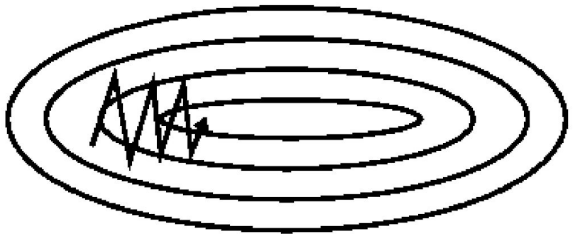
SGD with Momentum (Qian, 1999)

In the original paper:

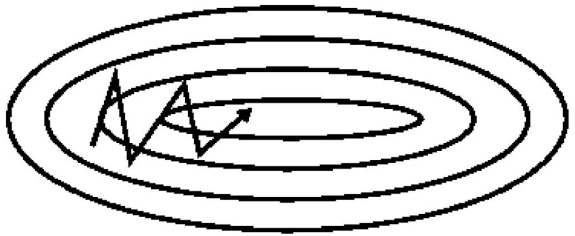
$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$
$$\theta = \theta - v_t$$

Actual implementation in PyTorch:

$$v_t = \gamma v_{t-1} + \nabla_\theta J(\theta)$$
$$\theta = \theta - \eta v_t$$



(a) SGD without momentum

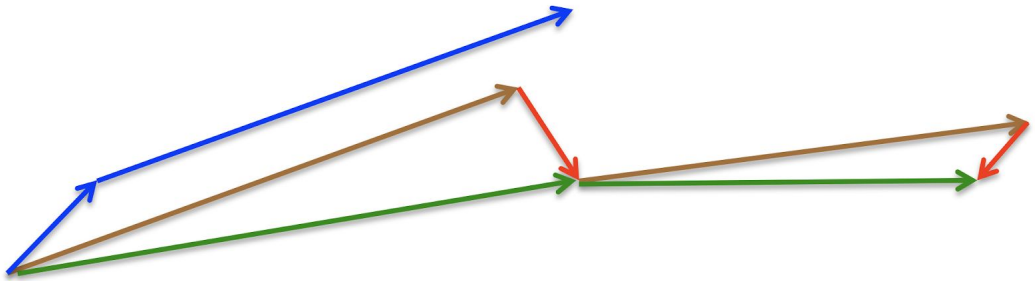


(b) SGD with momentum

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neu. Net.*, 1999.
Figure source: https://www.willamette.edu/~gorr/classes/cs449/momrate.html

Nesterov Accelerated Gradient (Nesterov, 1983)

$$v_t = \gamma\, v_{t-1} + \eta \nabla_\theta J(\theta - \gamma v_{t-1})$$
$$\theta = \theta - v_t$$



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

blue vectors = standard momentum

Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence O (1/k^2). *Doklady AN USSR*, 1983.
Figure source: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

AdaGrad (Duchi et al., 2011)

$G_t \in \mathbb{R}^{d \times d}$ is a diagonal matrix where each diagonal element $G_{t,ii}$ is the sum of the squares of the gradients w.r.t. θ_i up to time step t

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

J. Duchi, E. Hazan, Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 2011.

RMSprop (Tieleman & Hinton, 2012)

Use an exponential moving average instead of the sum used in AdaGrad.

$$\mathbb{E}\left[g^2\right]_t = \gamma \mathbb{E}\left[g^2\right]_{t-1} + (1-\gamma) g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\mathbb{E}\left[g^2\right]_t + \epsilon}} g_t$$

http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

Adam (Kingma & Lei Ba, 2015)

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\, g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\, g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

bias correction

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

D. P. Kingma, J. Lei Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

Adaptive Methods: Pros

- Faster training speed
- Smoother learning curve
- Easier to choose hyperparameters (Kingma & Lei Ba, 2015)
- Better when data are very sparse (Dean et al., 2012)

Adaptive Methods: Cons

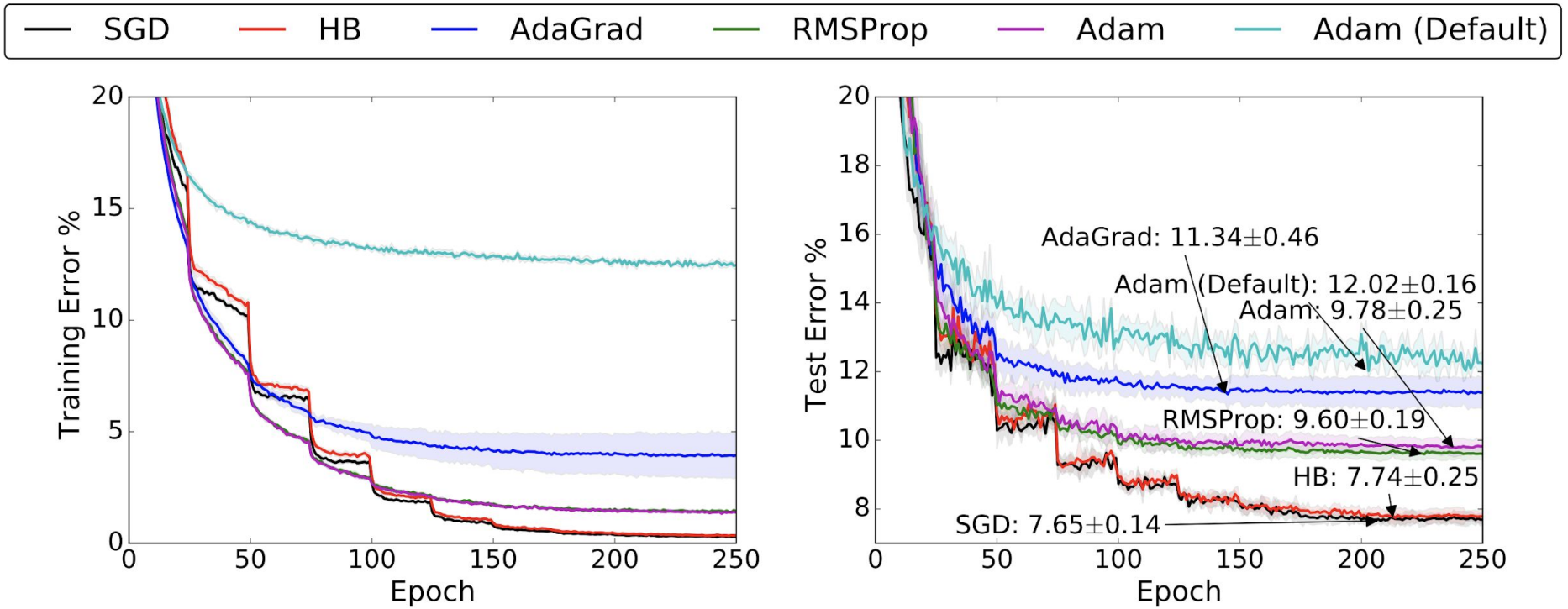
- Worse performance on unseen data (viz. dev/test set; Wilson et al., 2017)
- Convergence issue caused by non-decreasing learning rates (Reddi et al., 2018)
- Convergence issue caused by extreme learning rates (**Luo et al., 2019**)

Worse Performance on Unseen Data (Wilson et al., 2017)

The authors construct a binary classification example where different algorithms can find entirely different solutions when initialized from the same point, and particularly, *adaptive methods find a solution which has worse out-of-sample error than SGD*.

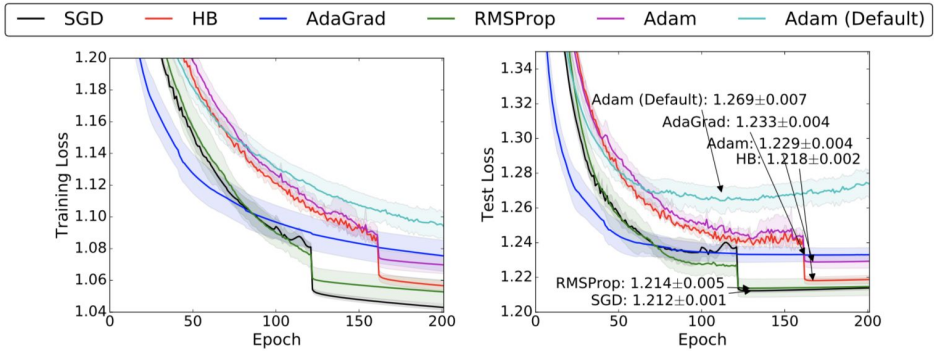
$$x_{ij} = \begin{cases} y_i & j = 1 \\ 1 & j = 2, 3 \\ 1 & j = 4 + 5(i-1), \ldots, 4 + 5(i-1) + 2(1 - y_i) \\ 0 & \text{otherwise} \end{cases}$$

A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, B. Recht. The marginal value of adaptive gradient methods in machine learning. *NeurIPS*, 2017.



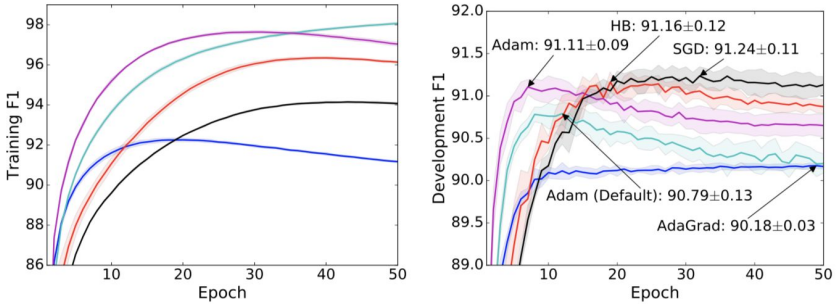
(a) CIFAR-10 (Train)

(b) CIFAR-10 (Test)



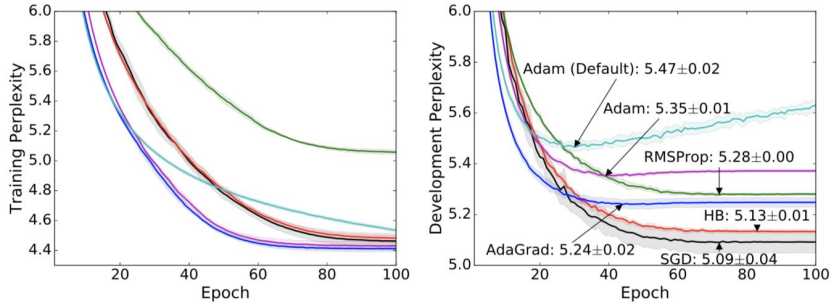
(a) War and Peace (Training Set)

(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)

Convergence Issue Caused by Non-Decreasing Learning Rates (Reddi et al., 2018)

The following quantity is always larger than or equals to zero for SGD, while not necessarily the case for Adam and RMSprop, which translates to *non-decreasing* learning rates. The authors prove that this can result in undesirable convergence behavior in certain cases.

$$\Gamma_{t+1} = \left(\frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t} \right)$$

S. J. Reddi, S. Kale, S. Kumar. On the convergence of Adam and beyond. *ICLR*, 2018.

Convergence Issue Caused by Extreme Learning Rates (**Luo et al., 2019**)

The authors demonstrate the existence of *extreme* learning rates when the model is close to convergence, and prove that this can lead to undesirable convergence behavior for Adam and RMSprop in certain cases *whatever value the initial step size is*.

Figure 1: Learning rates of sampled parameters. Each cell contains a value obtained by conducting a logarithmic operation on the learning rate. The lighter cell stands for the smaller learning rate.

Liangchen Luo, Yuanhao Xiong, Yan Liu, Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *ICLR*, 2019.

Proposals for Improvement

- **AMSGrad (Reddi et al., 2018)**
- **AdaBound (Luo et al., 2019)**
- AdaShift (Zhou et al., 2019)
- Padam (Chen & Gu, 2019)
- NosAdam (Huang et al., 2019)

AMSGrad (Reddi et al., 2018)

Algorithm 2 AMSGRAD

Input: $x_1 \in \mathcal{F}$, step size $\{\alpha_t\}_{t=1}^{T}$, $\{\beta_{1t}\}_{t=1}^{T}$, β_2
Set $m_0 = 0$, $v_0 = 0$ and $\hat{v}_0 = 0$
for $t = 1$ **to** T **do**

$$g_t = \nabla f_t(x_t)$$
$$m_t = \beta_{1t} m_{t-1} + (1 - \beta_{1t}) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$
$$\hat{v}_t = \max(\hat{v}_{t-1}, v_t) \text{ and } \hat{V}_t = \text{diag}(\hat{v}_t)$$
$$x_{t+1} = \Pi_{\mathcal{F}, \sqrt{\hat{V}_t}}(x_t - \alpha_t m_t / \sqrt{\hat{v}_t})$$

end for

gurantee a non-negative Γ_t
viz. non-increasing learning rates

S. J. Reddi, S. Kale, S. Kumar. On the convergence of Adam and beyond. *ICLR*, 2018.

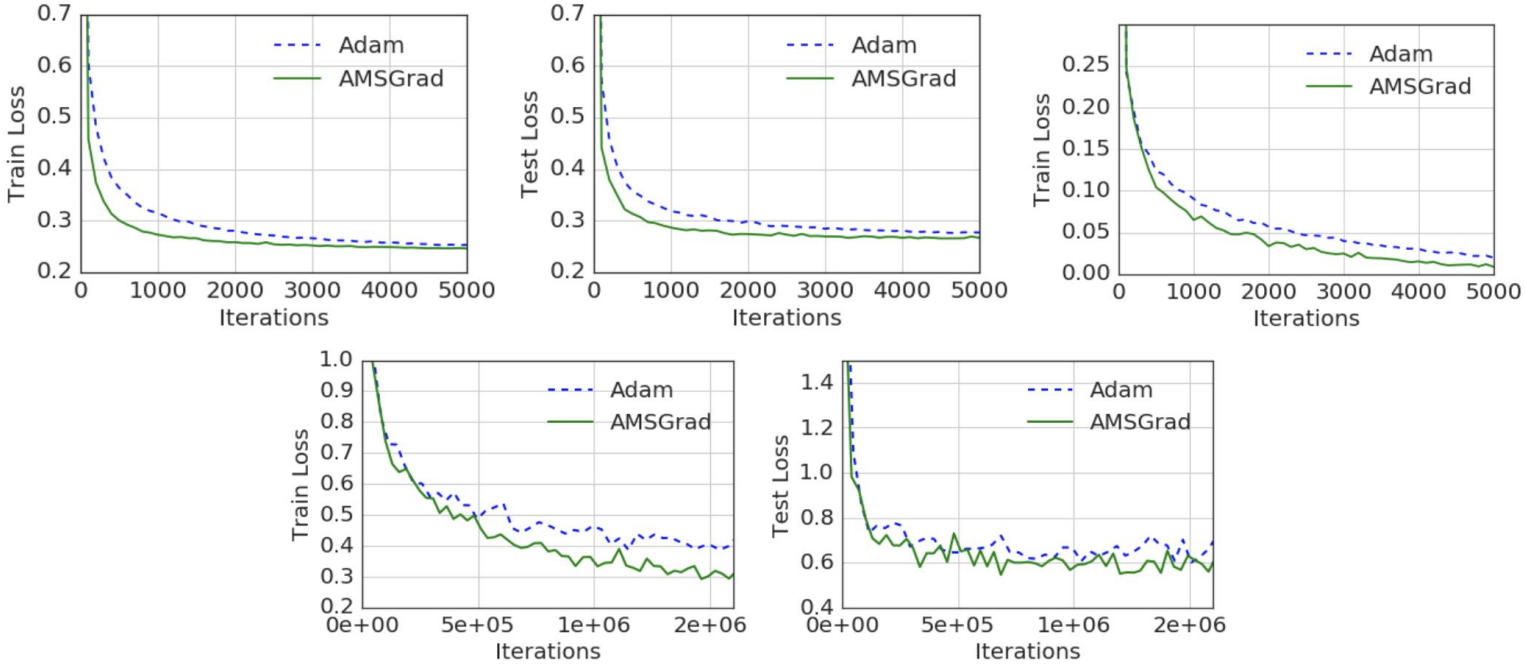


Figure 2: Performance comparison of ADAM and AMSGRAD for logistic regression, feedforward neural network and CIFARNET. The top row shows performance of ADAM and AMSGRAD on logistic regression (left and center) and 1-hidden layer feedforward neural network (right) on MNIST. In the bottom row, the two plots compare the training and test loss of ADAM and AMSGRAD with respect to the iterations for CIFARNET.

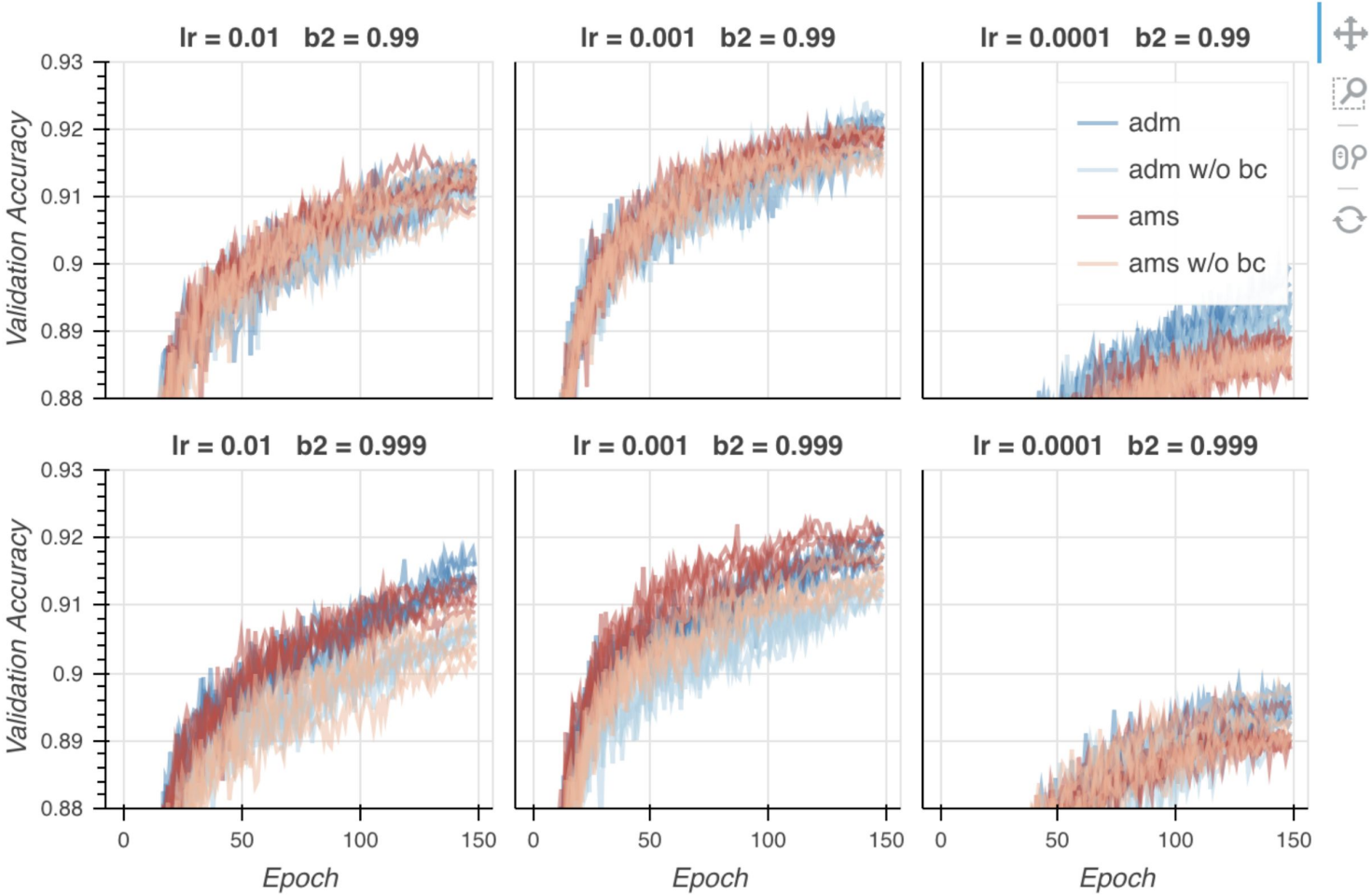


Figure source: https://fdlm.github.io/post/amsgrad/

AdaBound (**Luo et al., 2019**)

Consider applying the following operation in Adam, which clips the learning rate element-wisely such that the output is constrained to be in $[\eta_l, \eta_u]$.

$$\mathrm{Clip}(\alpha/\sqrt{V_t}, \eta_l, \eta_u)$$

It follows that SGD(M) and Adam can be considered as the following cases where:

For SGD(M): $\eta_l = \eta_u = \alpha^*$

For Adam: $\eta_l = 0, \eta_u = \infty$

Liangchen Luo, Yuanhao Xiong, Yan Liu, Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *ICLR*, 2019.

Algorithm 2 ADABOUND

Input: $x_1 \in \mathcal{F}$, initial step size α, $\{\beta_{1t}\}_{t=1}^{T}$, β_2, lower bound function η_l, upper bound function η_u

Set $m_0 = 0$, $v_0 = 0$
for $t = 1$ **to** T **do**
  $g_t = \nabla f_t(x_t)$
  $m_t = \beta_{1t} m_{t-1} + (1 - \beta_{1t}) g_t$
  $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ and $V_t = \mathrm{diag}(v_t)$
  $\hat{\eta}_t = \mathrm{Clip}(\alpha/\sqrt{V_t}, \eta_l(t), \eta_u(t))$ and $\eta_t = \hat{\eta}_t/\sqrt{t}$
  $x_{t+1} = \Pi_{\mathcal{F}, \mathrm{diag}(\eta_t^{-1})}(x_t - \eta_t \odot m_t)$
end for

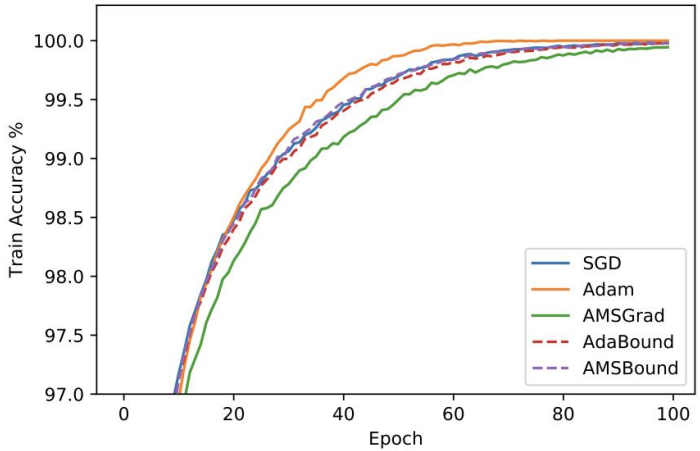
applying bound on learning rates

Employ η_l and η_u as functions of t instead of constant lower and upper bound, where

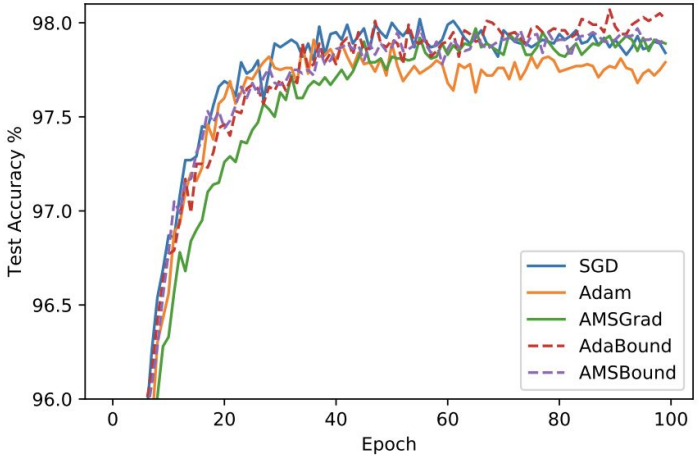
- $\eta_l(t)$ is an increasing function that starts from 0 and converges to α^* asymptotically;
- $\eta_u(t)$ is a decreasing function that starts from ∞ and converges to α^* asymptotically.

$$\eta_l(t) = (1 - \frac{1}{(1-\beta)t+1})\alpha^*,$$

$$\eta_u(t) = (1 + \frac{1}{(1-\beta)t})\alpha^*,$$

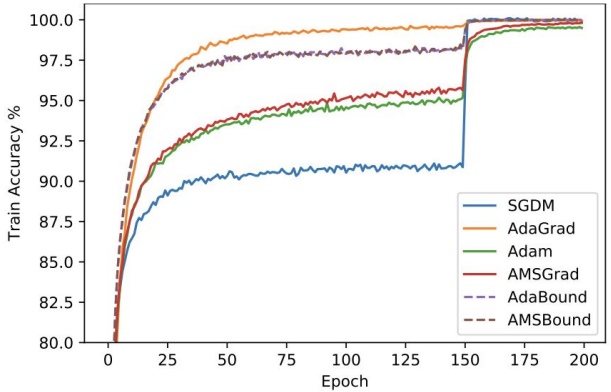


(a) Training Accuracy

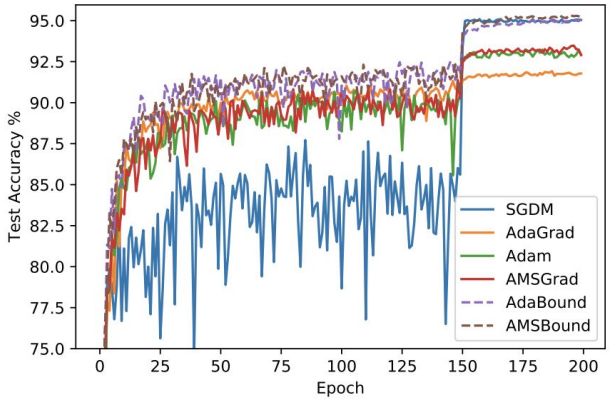


(b) Test Accuracy

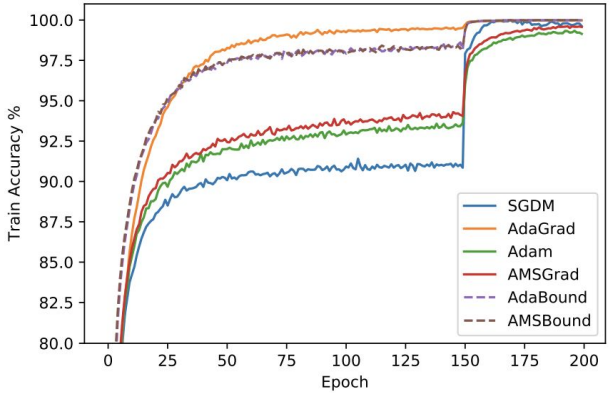
Figure 2: Training (left) and test accuracy (right) for feedforward neural network on MNIST.



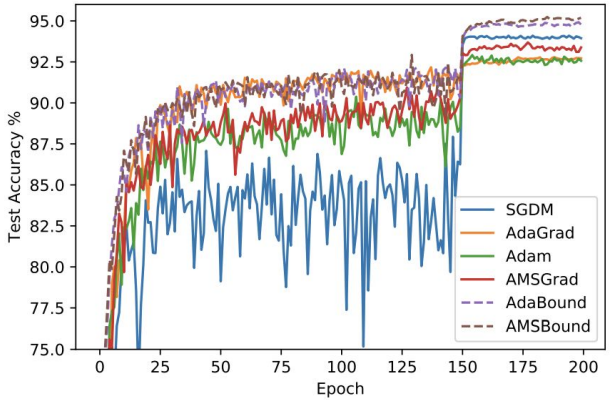
(a) Training Accuracy for DenseNet-121



(b) Test Accuracy for DenseNet-121

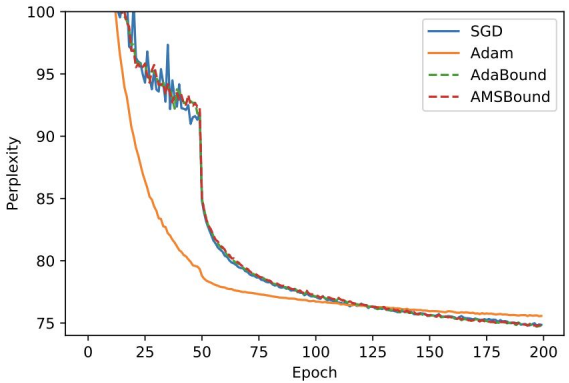


(c) Training Accuracy for ResNet-34

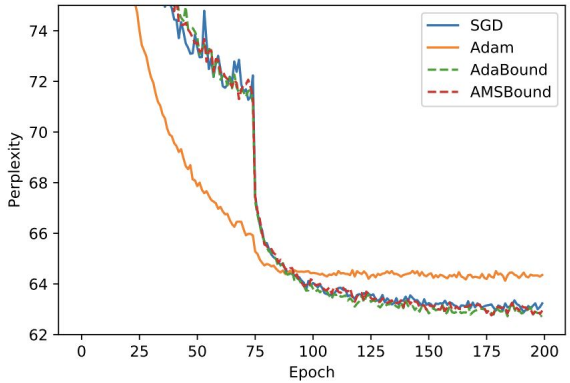


(d) Test Accuracy for ResNet-34

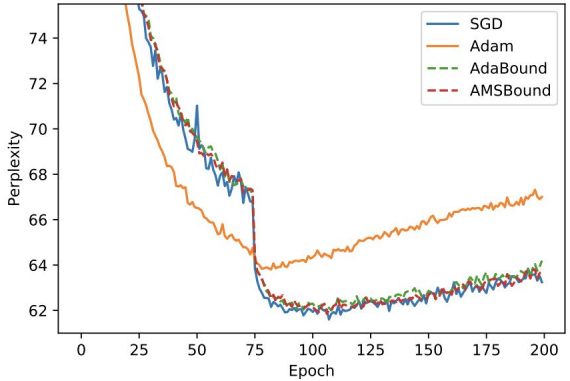
Figure 3: Training and test accuracy for DenseNet-121 and ResNet-34 on CIFAR-10.



(a) L1: 1-Layer LSTM



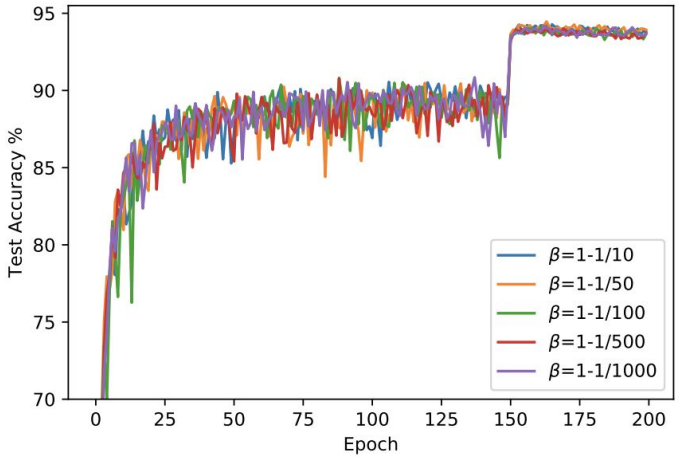
(b) L2: 2-Layer LSTM



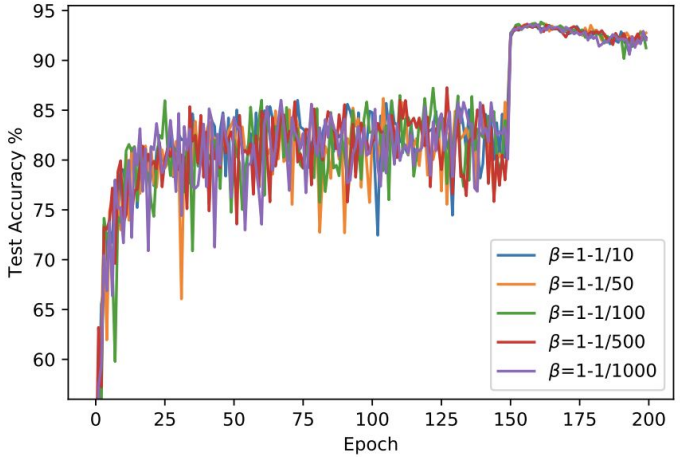
(c) L3: 3-Layer LSTM

Figure 4: Perplexity curves on the test set comparing SGD, ADAM, ADABOUND and AMSBOUND for the LSTM with different layers on Penn Treebank.

The Robustness of AdaBound

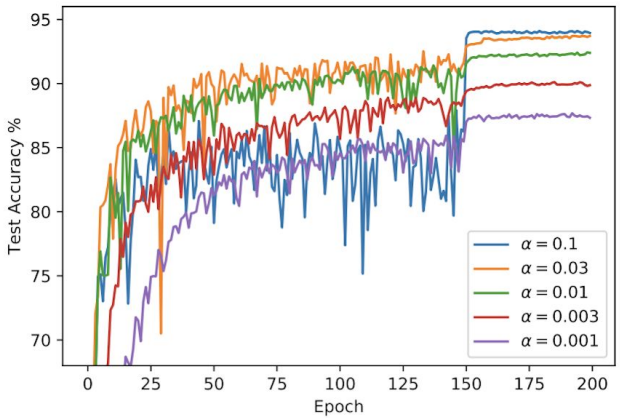


(a) ADABOUND with $\alpha^* = 0.1$

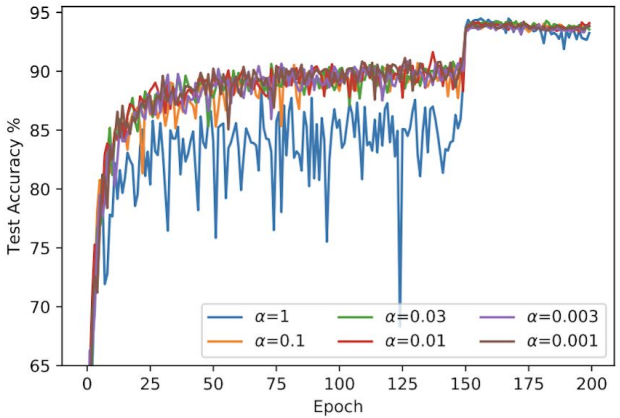


(b) ADABOUND with $\alpha^* = 1$

Figure 5: Test accuracy of ADABOUND with different β using ResNet-34 on CIFAR-10.



(a) SGDM with different α



(b) ADABOUND with different α^*

Figure 6: Test accuracy of SGDM/ADABOUND with different α/α^* using ResNet-34 on CIFAR-10. The result of SGDM with $\alpha = 1$ is not shown above as its performance is too poor (lower than 70%) to be plotted together with other results in a single figure.

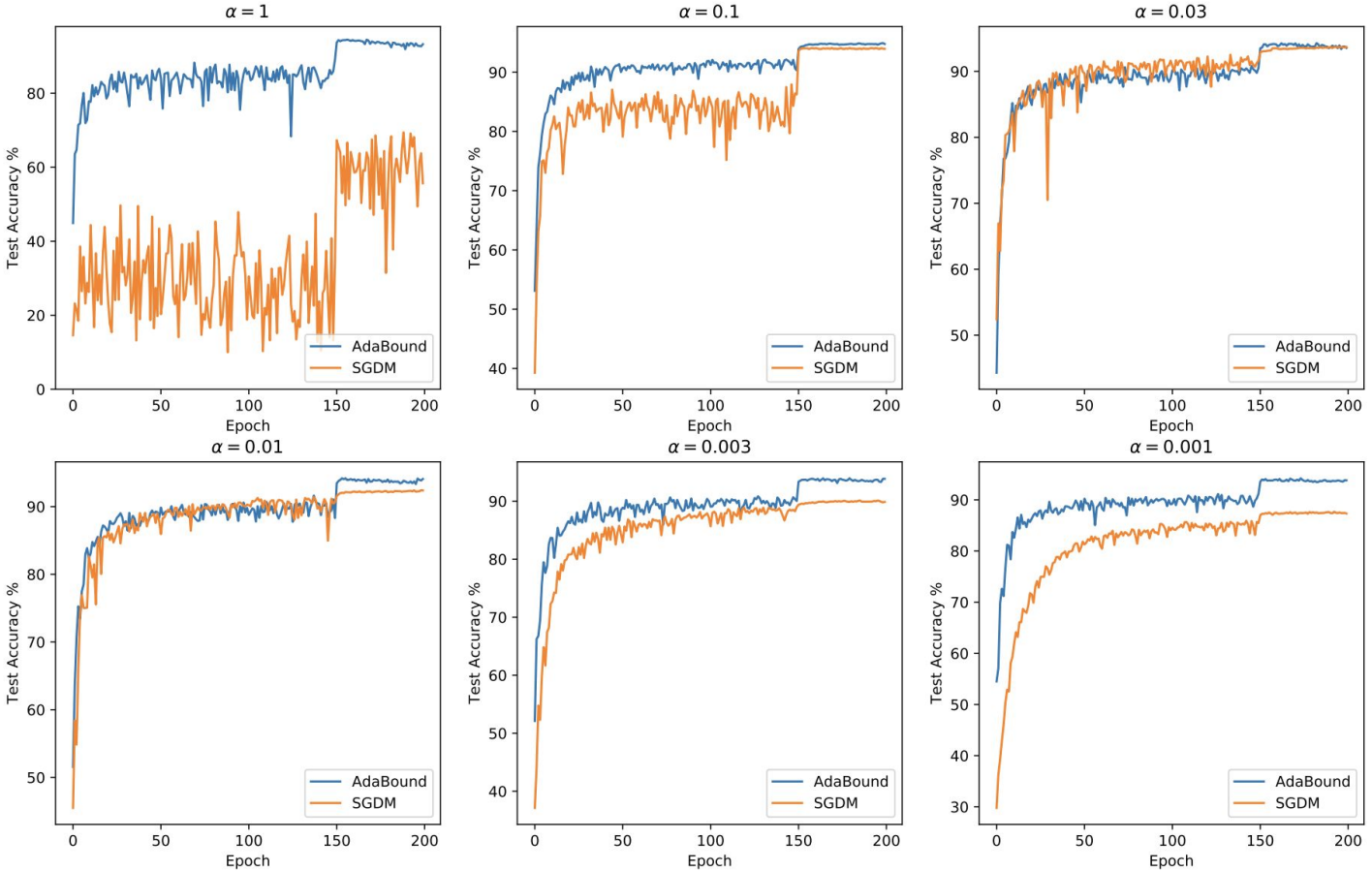


Figure 7: Comparison of test accuracy between SGDM and ADABOUND with different α/α^*.

The Limitations of AdaBound

- Based on the assumption that SGD would perform better than Adam w.r.t. the final generalization ability
- The form of bound function
  - gamma as a function of expected global step?
  - other functions
- Fixed final learning rate
  - how to determine the final learning rate automatically?

Any questions?