Prodigy: An Expeditiously Adaptive Parameter-Free Learner

SAMSUNG Research

Konstantin Mishchenko [1] Aaron Defazio [2]

[1] Samsung AI Center [2] Fundamental AI Research Team at Meta

Meta

The setting

We want to design practical adaptive methods for deep learning. How do we do that?
We find the convex framework to be useful:

$$\min_{x \in \mathbb{R}^p} f(x), \qquad (1)$$

where f is a differentiable function.
Assumption 1a (non-smooth f): For every $x \in \mathbb{R}^p$ and $g \in \partial f(x)$, it holds $\|g\| \le G$.
Assumption 1b (smooth f): For every $x, y \in \mathbb{R}^p$, it holds $\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|$.

Background: AdaGrad

Adam works well but it doesn't have good theory :(
AdaGrad is similar and it can be studied :)

$x_{t+1} = x_t - D_\infty \frac{g_t}{\sqrt{\sum_{k=0}^t g_k^2}}$, where $D_\infty = \|x_0 - x_*\|_\infty$, $g_t \in \partial f(x_t)$. For theory, it is easier to study **AdaGrad-Norm** [3],

$x_{t+1} = x_t - D \frac{g_t}{\sqrt{\sum_{k=0}^t \|g_k\|^2}}$, where $D = \|x_0 - x_*\|$.

or even **Normalized** Subgradient Descent [2]:

$$x_{t+1} = x_t - \alpha_t \frac{g_t}{\|g_t\|}, \quad \text{where} \quad \alpha_t \sim \frac{D}{\sqrt{t}},$$

Main issue: all methods require D or D_∞.

Background: D-Adaptation

Consider gradient descent, $x_{t+1} = x_t - \eta_t g_t$, then:

$$\begin{aligned} 0 &\le \sum_{t=0}^k \eta_t (f(x_t) - f(x_*)) \qquad \text{(optimality of } x_*) \\ &\le \sum_{t=0}^k \eta_t \langle g_t, x_t - x_* \rangle \qquad \text{(convexity of } f) \\ &= \sum_{t=0}^k \eta_t \langle g_t, x_0 - x_* \rangle + \sum_{t=0}^k \eta_t \langle g_t, x_t - x_0 \rangle \\ &\le \left\| \sum_{t=0}^k \eta_t g_t \right\| \|x_0 - x_*\| + \sum_{t=0}^k \eta_t \langle g_t, x_t - x_0 \rangle, \end{aligned}$$

which gives

$$D = \|x_0 - x_*\| \ge \hat{d}_k = \frac{\sum_{t=0}^k \eta_t \langle g_t, x_0 - x_t \rangle}{\| \sum_{t=0}^k \eta_t g_t \|}.$$

Using $d_k = \max(\hat{d}_k, d_{k-1})$ as the stepsize is the key idea behind **D-Adaptation** [1].

Summary

Goal: Adam-like method without learning rate.
Idea: instead of estimating just the gradient magnitude g^2, estimate the *product* of d and g (hence the name, Pro-di-gy)
Theory: Prodigy has faster convergence rates than D-Adaptation in terms of D/d_0 (price of adaptivity).
Experiments: Prodigy nearly matches hand-tuned Adam on a range of deep learning problems.

Prodigy

Algorithm 1 Prodigy (Adam-based)

Input: $x_0 \in \mathbb{R}^p$, $d_0 > 0$ (default: 10^{-6}), β_1 (default: 0.9), β_2 (default: 0.999), schedule γ_k (default: 1), $\epsilon = 10^{-8}$)

1: **for** $k = 0, 1, \dots$ **do**
2: $g_k \in \partial f(x_k)$
3: $m_{k+1} = \beta_1 m_k + (1 - \beta_1) d_k g_k$
4: $v_{k+1} = \beta_2 v_k + (1 - \beta_2) d_k^2 g_k^2$
5: $r_{k+1} = \sqrt{\beta_2} r_k + (1 - \sqrt{\beta_2}) \gamma_k d_k^2 \langle g_k, x_0 - x_k \rangle$
6: $s_{k+1} = \sqrt{\beta_2} s_k + (1 - \sqrt{\beta_2}) \gamma_k d_k^2 g_k$
7: $\hat{d}_{k+1} = \frac{r_{k+1}}{\|s_{k+1}\|_1}$
8: $d_{k+1} = \max(d_k, \hat{d}_{k+1})$
9: $x_{k+1} = x_k - \gamma_k d_k m_{k+1} / (\sqrt{v_{k+1}} + d_k \epsilon)$
10: **end for**

Algorithm 2 Prodigy (GD-based)

Input: $x_0 \in \mathbb{R}^p$, $d_0 > 0$

1: **for** $k = 0, 1, \dots$ **do**
2: $g_k \in \partial f(x_k)$
3: $v_{k+1} = v_k + d_k^2 \|g_k\|^2$
4: $\eta_k = \frac{d_k^2}{\sqrt{v_{k+1} + d_k^2 G^2}}$
5: $r_{k+1} = r_k + \eta_k \langle g_k, x_0 - x_k \rangle$
6: $\hat{d}_{k+1} = \frac{r_{k+1}}{\|x_0 - x_k\|}$
7: $d_{k+1} = \max(d_k, \hat{d}_{k+1})$
8: $x_{k+1} = x_k - \eta_k g_k$
9: **end for**

Theorem 1. On non-smooth problems, gap $f(x) - f_*$ converges as $\mathcal{O}\left(\frac{\sqrt{\log_{2+}(D/d_0)}}{\sqrt{k}}\right)$.

Theorem 2. On L-smooth problems, we set $G = 0$ and the gap $f(x) - f_*$ converges with rate $\mathcal{O}\left(\frac{\log_{2+}(D/d_0) \log_{2+}^2\left(\frac{LD}{d_0\|g_0\|}\right)}{k}\right)$.

References

[1] Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by D-adaptation. In *International Conference on Machine Learning*, pages 7449–7479. PMLR, 2023.

[2] Naum Zuselevich Shor. *Minimization methods for non-differentiable functions.* Springer Science & Business Media, 1985.

[3] Matthew Streeter and H. Brendan McMahan. Less regret via online conditioning. *arXiv preprint arXiv:1002.4862*, 2010.

Experiments

We compare Prodigy, which has no learning-rate tuning, to Adam with a manually tuned learning rate (weight decay was tuned for all methods).

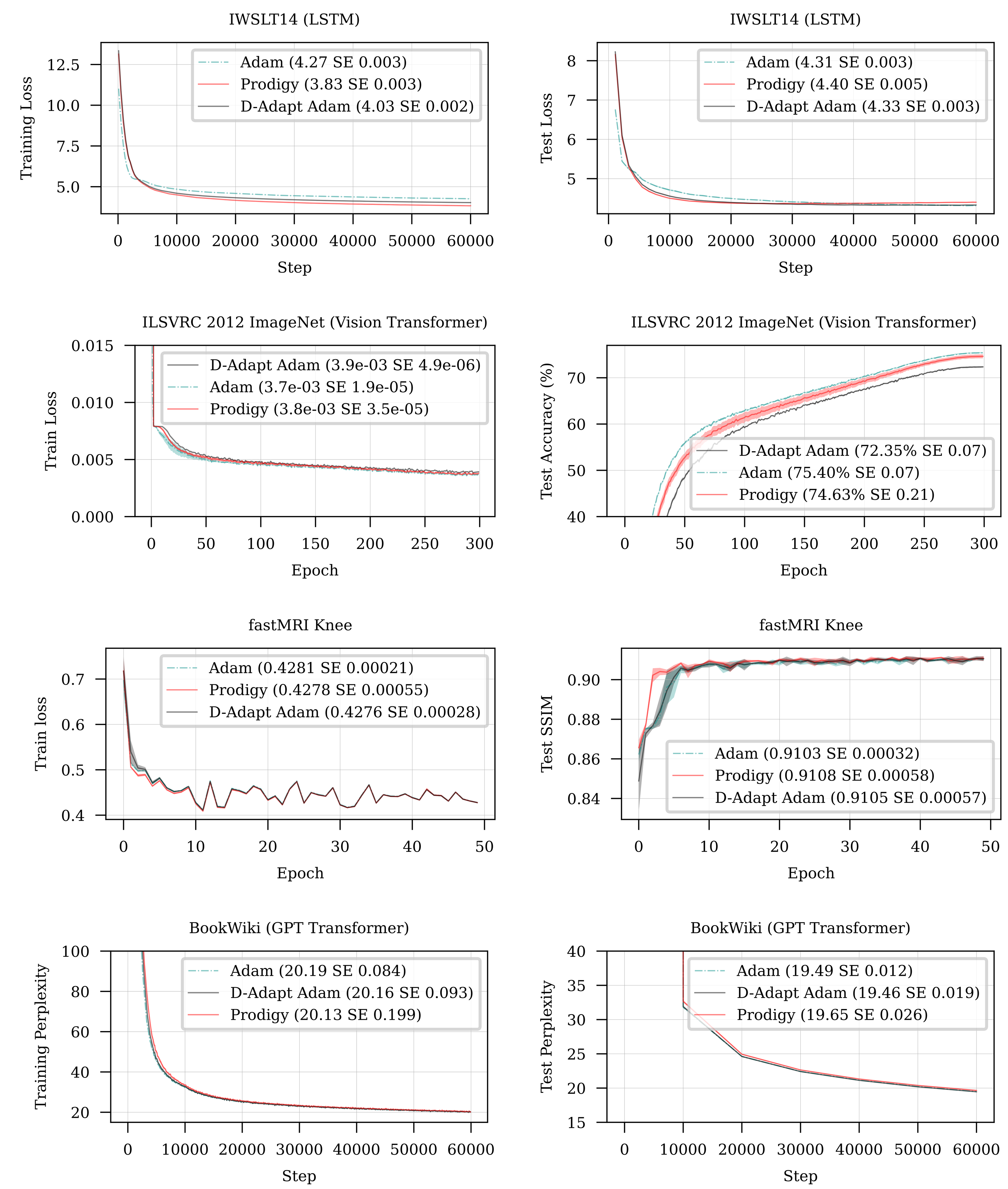

Figure 1: Left: train loss, right: validation loss.

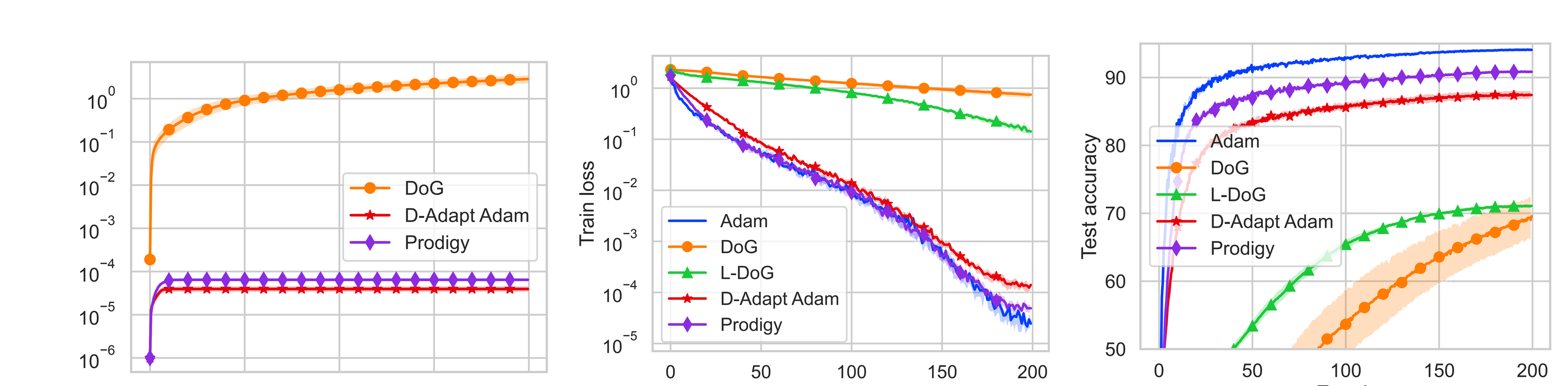

Figure 2: CIFAR10 experiment (ResNet-50). Left: learning rate, middle: train loss, right: validation accuracy.