Noise Is Not the Main Factor Behind the Gap Between SGD and Adam on Transformers. But Sign Descent Might Be.

We do not have a good explanation for **why Adam outperforms SGD by a large margin**.

Prior work suggests **Adam might be more robust to the noise induced by subsampling**.

Through experiments in full batch, we show **resilience to noise is not the main factor**.

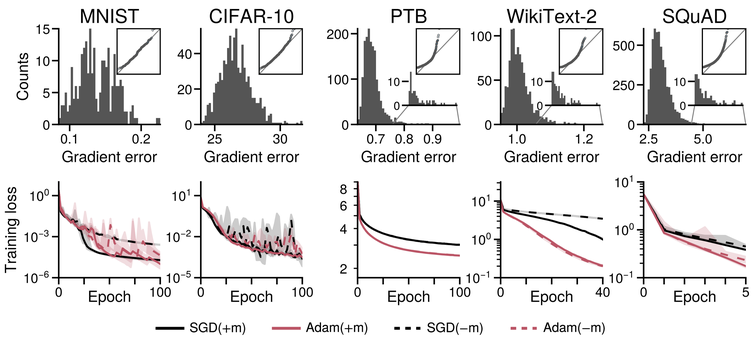
Experiments in full batch show that **the behavior of Adam is closest to Sign Descent.**

openreview.net/pdf?id=a65YK0cqH8g

Noise is not the main bottleneck for SGD

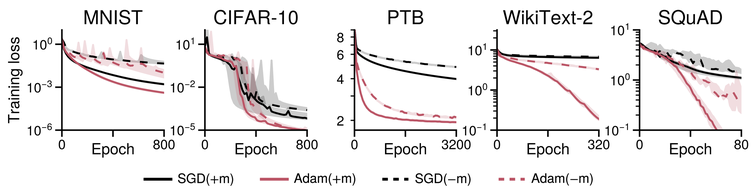
Previous work suggests that Adam outperforms SGD because it is more resilient to heavy-tailed noise in stochastic gradients. **But is this what is holding back SGD?**

Prior work suggests the gap between SGD and Adam might come from resilience to heavy-tailed noise

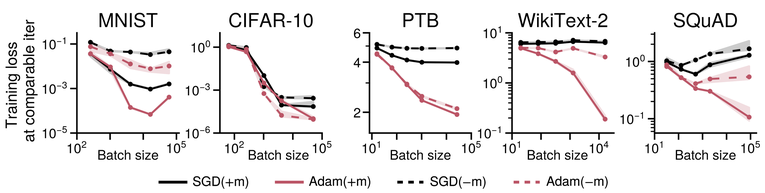


We also observe that the performance gap between SGD and Adam is larger on transformers than on CNNs, which coincides with heavier tails in stochastic gradient error.

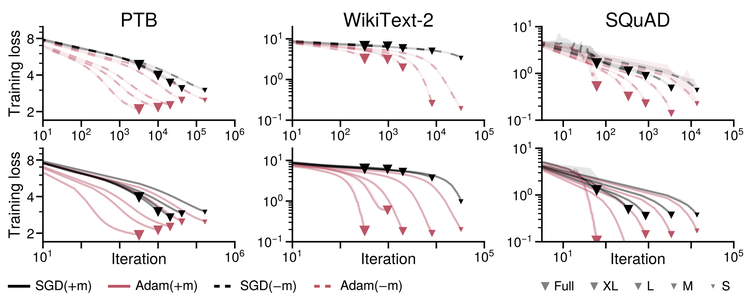
But the gap does not disappear in full batch...



... instead the gap grows with batch size on transformers



Looking at the full trajectory shows that SGD does not improve as much as Adam with batch size



For each problem, each optimizer is shown with five different batch sizes, interpolating between the small batch setting of Figure 1 and the full batch setting of Figure 2. Adam better takes advantage of the reduction in noise due to large batch sizes.

The benefit of Adam over SGD appears deterministic

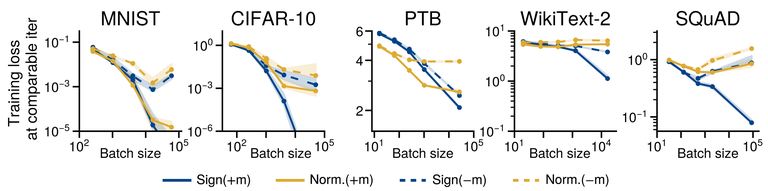
If the benefit is deterministic, which component of Adam leads to this improvement over gradient descent?

Can a simpler algorithm do it?

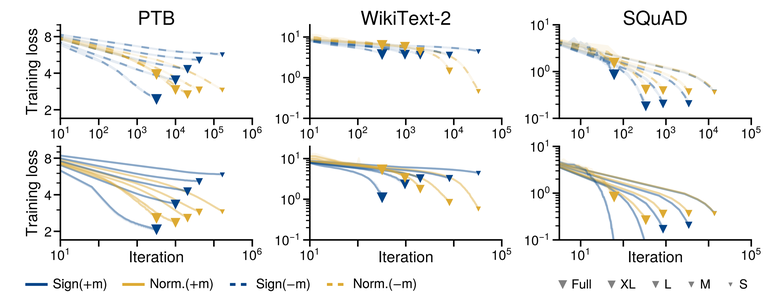
Repeat experiments with simpler methods; **Normalized GD** and **Sign descent**

$$m_{t+1} = \beta m_t + h(g_t), \quad \text{where} \quad h(g) = g/\|g\|_2 \quad \text{for normalized GD},$$
$$x_{t+1} = x_t - \alpha m_{t+1}. \qquad\qquad h(g) = \text{sign}(g) \quad \text{for sign descent}.$$

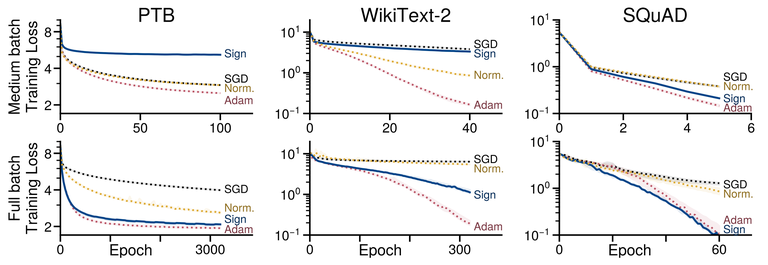
Normalization and Sign descent scale better with batch size and outperform SGD as the batch size increases



Sign descent with momentum improves most with batch size and performs similarly as Adam in full batch



Despite (very!) poor performance with small batches, Sign Descent performs similarly to Adam in full batch



Sign descent gives a simpler algorithm to try to analyze, but still missing a complete theory for why it helps

Also observed in practice, e.g. the LION optimizer is Sign descent + Momentum
(Chen et al., 2302.06675)

Possible relationship between gradient and Hessian justifying sign-like methods
(Zhang et al., 1905.11881 and Crawshaw et al., 2208.11195)

Frederik Kunstner Jacques Chen
J. Wilder Lavington Mark Schmidt

THE UNIVERSITY OF BRITISH COLUMBIA