The AdEMAMix Optimizer: Better, Faster, Older

Pierre Ablin
Apple
p_ablin@apple.com

The vast majority of large-scale machine learning models are trained using momentum-based optimizers like Adam and AdamW [1]. They rely on an exponential moving average (EMAs) of gradients to accumulate past optimization information. However, a single EMA faces a fundamental trade-off: it cannot simultaneously assign high weight to recent gradients (which are up-to-date) and non-negligible weight to older gradients (which are useful for reducing variance and accelerating convergence).

We introduce **AdEMAMix**, a simple modification of AdamW that uses a mixture of two EMAs with different decay rates. The first EMA uses a standard decay ($\beta_1 \approx 0.9$) for recent information, while the second uses a much slower decay ($\beta_3 \approx 0.9999$) to leverage gradients from tens of thousands of steps in the past. The update rule combines both EMAs with a mixing coefficient α that controls their relative contributions.

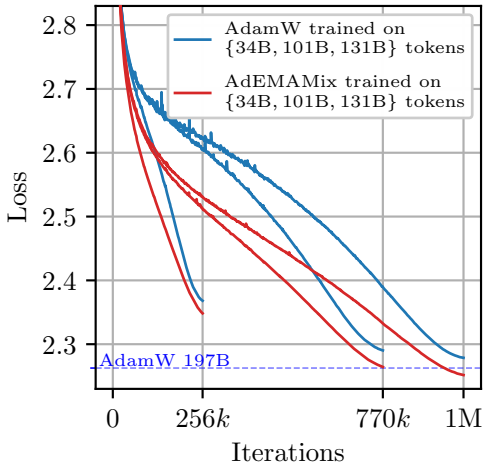

Figure 1: Validation loss for 1.3B parameter transformers trained on RedPajama. AdEMAMix converges faster than AdamW.

Experimental Results. We trained transformer language models (110M–1.3B parameters) on RedPajama, Mamba models (168M) on FineWeb, and Vision Transformers (24M–86M) on ImageNet datasets.

- **Language modeling:** For 1.3B parameters, AdEMAMix on 101B tokens matches AdamW on 197B tokens, which amounts to 95% less data (Fig. 1). We report similar gains across all sizes (110M, 330M, 1.3B).
- **Vision:** On ImageNet-21k (11M images, 37 epochs), AdEMAMix outperforms AdamW for ViTs on train/test losses.
- **Forgetting:** By tracking individual batches, AdEMAMix forgets training data significantly slower than AdamW.
- **Overhead:** Negligible slowdown ($< 3\%$) compared to AdamW, and we release drop-in code, compatible with Adam.

Joint work with: Matteo Pagliardini, David Grangier

References

[1] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.