Neon: Nuclear Norm to Beat Muon

Alexey Kravatskiy, Ivan Kozyrev, Nikolay Kozlov, Alexander Vinogradov
Optimization Class Project. MIPT

Introduction

Recent advances in optimization techniques have highlighted the benefits of leveraging the matrix structure of neural network weights during training. Optimizers like Muon (Jordan et al.) and Shampoo (Gupta, Koren and Singer) have shown promise, but at a significantly higher computational cost than traditional methods like Adam. To bridge this gap, we propose Neon, a new optimizer that builds upon the framework of Bernstein and Newhouse. By using alternative norms, such as kernel norm (Kernel-Neon) or custom $F*$ norm ($F*$-Neon), we induce low-rank update matrices that enable more efficient computation. We evaluate the performance of Neon, Muon, and Adam on MLP, CIFAR10, and NanoGPT, and demonstrate [resting results here].

Neon's update rule

Bernstein and Newhouse suggest obtaining the update step for a weight matrix W as a solution to the optimization problem:

$$\langle G, \delta W \rangle + \frac{\lambda}{2}\|\delta W\|^2 \to \min_{\delta W}, \tag{1}$$

where G is a gradient-like matrix obtained via backpropagation. Setting norm to RMS-to-RMS norm (scaled version of Spectral norm) produces Muon. We consider two different choices instead:

1. Choosing kernel norm, $\|\cdot\|_*$, produces rank-1 update, defined by

$$\delta W = -\frac{1}{\lambda} u_1 \sigma_1 v_1^T, \tag{2}$$

where σ_1 is largest singular value of G, and u_1, v_1 are corresponding singular values.

2. Choosing $F*$ norm, defined by $\|\cdot\|_{F*} = (\|\cdot\|_* + \|\cdot\|_F)/2$, produces a relatively small-rank update, defined by

$$\delta W = -\frac{1}{\lambda} U D V^T \tag{3}$$

with $D = \mathbf{diag}(d_i)$, where $d_i = [\sigma_i - \tau]_+$ and τ is given by

$$\sum_{i=1}^{n} [\sigma_i - \tau]_+ = \tau\,.$$

Efficient update computation

A couple of sentences about Lanczos algorithm and reference computation times and required number of iterations for tipical value of τ.

Algorithms

Now we can write down pseudocode for both versions of Neon.

Algorithm 1 Kernel-Neon update step for linear layer

Input: λ, gradient-like matrix G.
Output: Weight matrix update δW.
1. $U, \Sigma, V := \mathrm{Lanczos}(G, 1)$.
Return $-\frac{1}{\lambda} U \Sigma V^T$.

For $F*$-Neon it is a bit trickier (3), we need to compute τ and know number of singular values larger then τ, which we denote by r. Assuming that singular values spectrum of G changes little between iterations (which was true in our experiments) we propose the following algorithm.

Algorithm 2 $F*$-Neon update step for linear layer

Input: λ, r, gradient-like matrix G.
Output: Weight matrix update δW.
1. $U, \Sigma, V := \mathrm{Lanczos}(G, r+1)$.
2. $s := \sum_{i=1}^{r} \sigma_r$.
3. **If** $(r+1)\sigma_{r+1} > s$**:**
   4. $r := r+1$. 5. $\tau := \frac{s+\sigma_r}{r+1}$.
6. **Else if** $(r+1)\sigma_{r-1} < s$**:**
   7. $r := r-1$. 8. $\tau := \frac{s-\sigma_r}{r+1}$.
9. **Else:**
   10. $\tau := \frac{s}{r+1}$
11. $D = [\Sigma - \tau I]_+$.

Save r for the next iteration.
Return $-\frac{1}{\lambda} U D V^T$.

MLP tests

Couple of sentences and a vector(!) picture (reference picture for now).

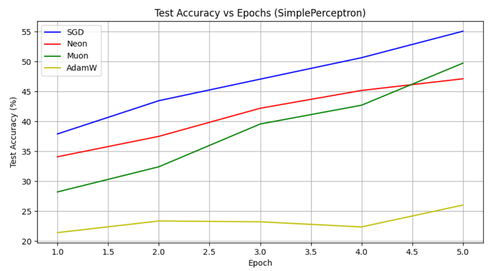

Figure 1: MLP validation loss

CIFAR10 tests

Couple of sentences and a vector(!) picture (reference picture for now).

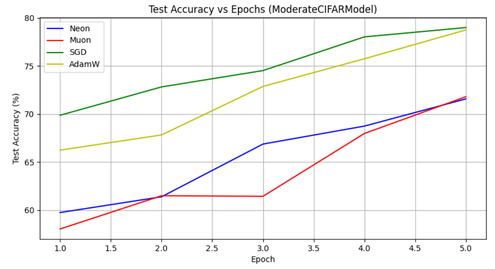

Figure 2: CIFAR10 validation loss

NanoGPT tests

Couple of sentences and a vector(!) picture (reference picture for now).

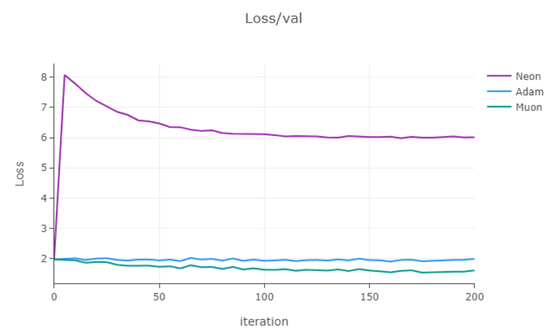

Figure 3: NanoGPT validation loss

Conclusion

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Acknowledgements

This material is based upon work supported by the X Fellowship and my mom.