Comparison Matrix

I investigated adam_epochs × sgd_epochs-sized matrices where the i, j event is $\frac{\langle \theta_i, \theta_j \rangle}{\|\theta_i\| \|\theta_j\|}$ where θ is the model's parameters updated according to either base Adam or SGD. For example, for a CNN on CIFAR-10 has comparison matrix shown in figure 1.

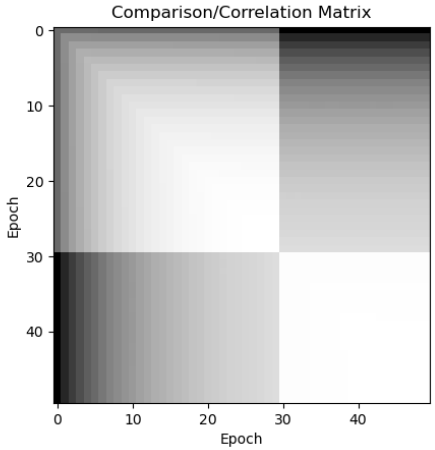

Figure 1: Comparison Matrix for a CNN on CIFAR-10

- Not fruitful! One reason is this behaves like $\langle U, V \rangle$, where U, V are uniform on the sphere (comes from normalized Gaussians), but the variance of this is like $1/d$, where d is dimension (very large).

Starting Counterexample Search

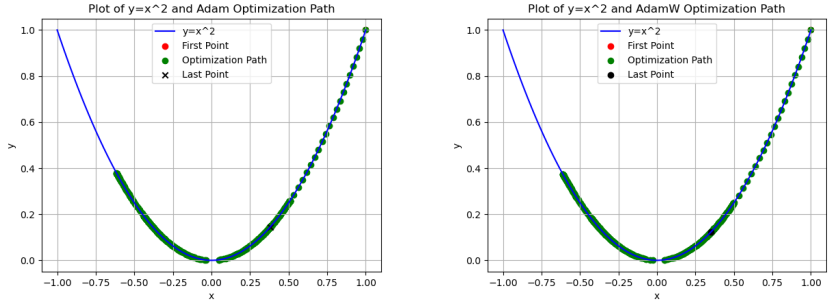

Figure 2: $\beta_1 = .999, \beta_2 = 0.001$ leads to bad convergence

I started searching for counterexamples with simple functions and bad β_1, β_2 parameters. This was easily found in figure 2. Obvious, but it gives hope. In theorem 3 of [3], the authors describe an avenue for Adam failing by variation in the gradients. In effect, Adam will have to memorize more than a single momentum parameter. My second attempt to get a counterexample was looking at quadratics of the form $\alpha x^2 + \beta y^2$ with large deviation in α, β. These are shown below in Figure 3.

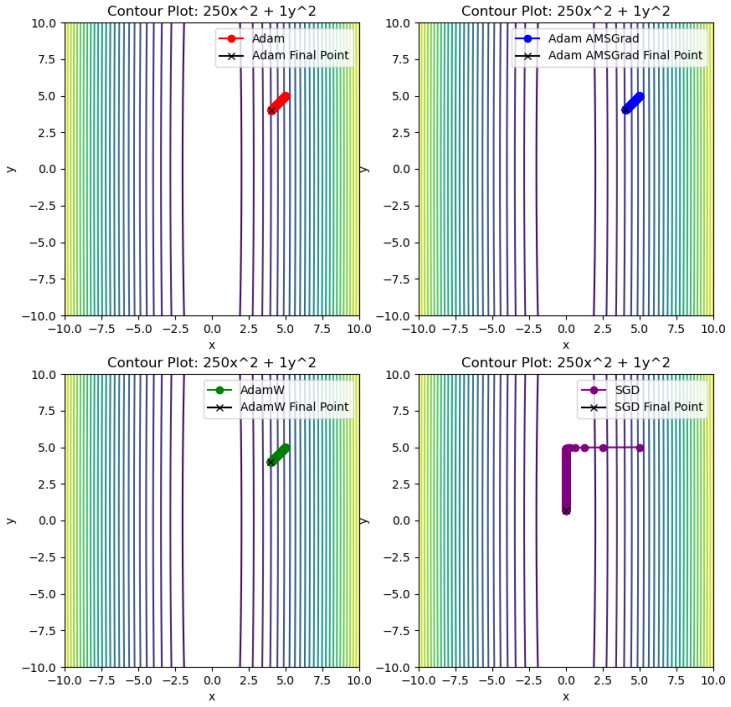

Figure 3: Optimizer Behavior with 1000 iterations and initial point (5,5)

Note that this was using default parameters, and the high variance slows the adaptive methods to a crawl (akin to walking slowly down a steep hill). This is because $\Delta\theta_t \sim \nabla f(\theta_t)$ for SGD, and the changes are slower for Adam. I could not find a general convex function where Adam fails to converge, so I had to turn to online learning methods.

Regret and Online Convex Optimization

In online convex optimization, the learner makes a prediction x_t, receives the convex loss function $f_t(\cdot)$ and incurs a loss $f_t(x_t)$. For some intuition think about

$$\min\{f(x)\} + \min\{g(x)\} \leq \min\{f(x) + g(x)\},$$

so if we could do a minibatch-wise optimization, it could be better than doing this after accumulation. The optimization problem is now given by the regret R of algorithm $\mathcal{A}$ in

$$R_T(\mathcal{A}) = \sum_{t=1}^{T} f_t(\theta_t) - \min_{\theta \in C} \sum_{t=1}^{T} f_t(\theta),$$

where C is the convex set where our weights belong. Average regret, R_T/T compares our average distance to the optimal value, so we want $R_T = o(T)$. In general, we know SGD can do online learning in $\mathcal{O}(\sqrt{T})$ ([2]) for bounded subgradients. In section 3 of [3], the authors show Adam can have constant average regret (not learn!) with any constant $C > 2$ and

$$f_t(x) = \begin{cases} Cx, & t \mod 3 = 1 \\ -x, & \text{else} \end{cases} \qquad \text{convex domain: } [-1, 1]$$

Theory

The authors show $x_t > 0$ and very often $x_t = 1$, staying away from the optimal $x = -1$. Originally, I thought AdamW would decay $x_t \downarrow 0$ in the limit and therefore provide a better solution, but I modified the proof in [3] to work for AdamW (under default weight decay, although other levels may work). In fact, for some cases (like decaying step size), the proof shows that all the iterates $x_t > 1/2$ for both Adam and AdamW, meaning we can actually create a more natural sequence of functions

$$g_t(x) = \mathbb{1}_{t \pmod 3 = 1} \begin{cases} Cx^2, & 0 \leq x \leq 1/2 \\ Cx, & 1/2 < x \leq 1 \end{cases} + \mathbb{1}_{t \pmod 3 \neq 1} \begin{cases} -x^2, & 0 \leq x \leq 1/2 \\ -x, & 1/2 < x \leq 1 \end{cases}$$

which are quadratic until it hits $1/2$, when they become linear. I did have to weaken the assumptions to $3C > 8$ instead of $C > 2$ from Reddi et. al., but this is minor. For the proof of the first case see the report pdf in the folder, but for the intuition think about one large gradient step (from C) followed by stepping up twice with positive gradient updates (from $-x$).

1 Numerical Verification

I tested the prior f_t with $C = 4$ (to satisfy my stronger condition) and $\beta_1 = 0, \beta_2 = 1/(1 + C^2)$ which satisfies the qualifications for theorem 4.1 in [1] for all iterations up to 10000. Each iteration I calculated the average regret and the guessed x_t value. Results are shown below in Figure 4.

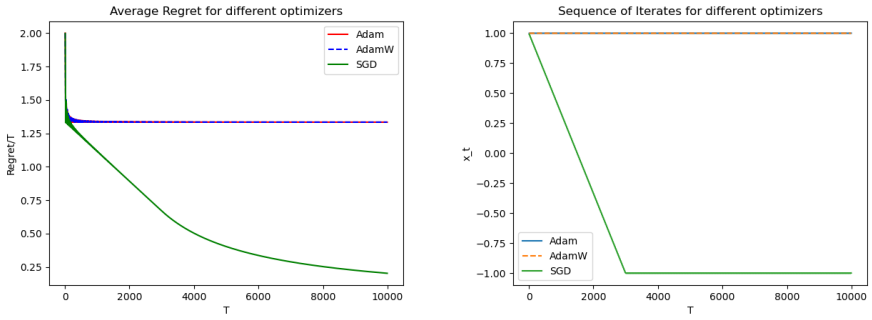

Figure 4: Average Regret and Iterates for Different Optimizers

We have the expected convergence of SGD reasonably following the bound, but we see both Adam and AdamW (default weight decay) failing to learn. The iterates also clearly stay far away from the bound of $1/2$ as predicted in the theory, with variance around 2×10^{-7} for Adam and AdamW, but SGD does indeed converge to the optimal solution. With more weight decay AdamW does indeed start to find $x = 0$, but the threshold I found numerically was around 20% weight decay (less than my guarantee) which is not only impractical, also it only works because the optimal point is in the same direction as 0 w.r.t. x_t. Since this problem is translation invariant, we could translate everything far away so AdamW could behave worse.

References

[1] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[2] John Langford, Alexander Smola, and Martin Zinkevich. *Slow Learners are Fast*. 2009. arXiv: 0911.0491 [math.OC].

[3] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. *On the Convergence of Adam and Beyond*. 2019. arXiv: 1904.09237 [cs.LG].