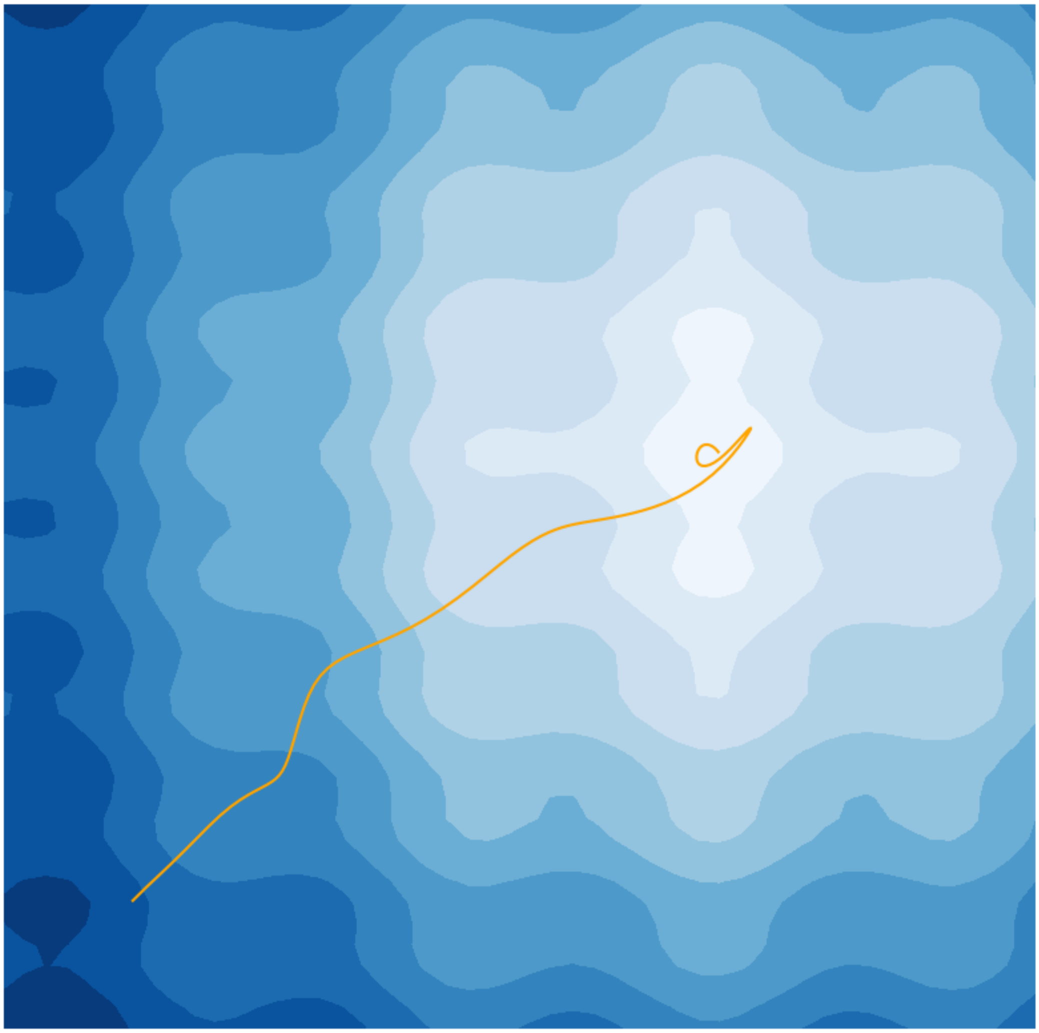
The Adam optimizer and solving large i.i.d. linear least squares systems

Axel Orrhede

November 2024



1 The Adam Optimizer

The Adam optimizer is an evolution of SGD, implementing the benefits of momentum from ADAgrad and adaptive individual learning rates for each parameter from RMSprop. Since Adam's publication in 2014, many further developments have been made. In practice, however, Adam still stands as a great optimizer [7], which is why it is widely used and why we will dive further into it in this report.

Firstly, we need to define two variables for each parameter w to introduce momentum and adaptive learning rate. [6]

$$m_w^{(t+1)} := \beta_1 m_w^{(t)} + (1-\beta_1)\nabla_w L^{(t)} \tag{1}$$

$$v_w^{(t+1)} := \beta_2 v_w^{(t)} + (1-\beta_2)(\nabla_w L^{(t)})^2 \tag{2}$$

Where β_1 and β_2 are hyperparameters for the optimization and $\nabla_w L^{(t)}$ is the gradient element of the loss function L pertaining to the parameter w. It is common to initialize $v_w^{(0)} = 0$ and $m_w^{(0)} = 0$, but this means you should probably do the following bias-correction, allowing better steps even though m and v have yet to reach their potential. [6]

$$\hat{m}_w = \frac{m_w^{(t+1)}}{1-\beta_1^t} \qquad \hat{v}_w = \frac{v_w^{(t+1)}}{1-\beta_2^t} \tag{3}$$

In the end, you update the parameters according to the following formula. [6]

$$w^{(t+1)} := w^{(t)} - \alpha\frac{\hat{m}_w}{\sqrt{\hat{v}_w}+\epsilon} \tag{4}$$

Where ϵ is used to prevent division by 0 and α is the learning rate.

If we then create a nonlinear sum of functions shown in the title page:

$$L(\mathbf{w}) = \sin(\pi w_0) + 0.5\cos(2\pi w_1) - 20\exp\left(-\frac{\|\mathbf{w} - \begin{bmatrix}1.5\\0.5\end{bmatrix}\|^2}{20}\right) \tag{5}$$

We can implement the Adam optimizer from above to find the minimum as shown in figure 1.

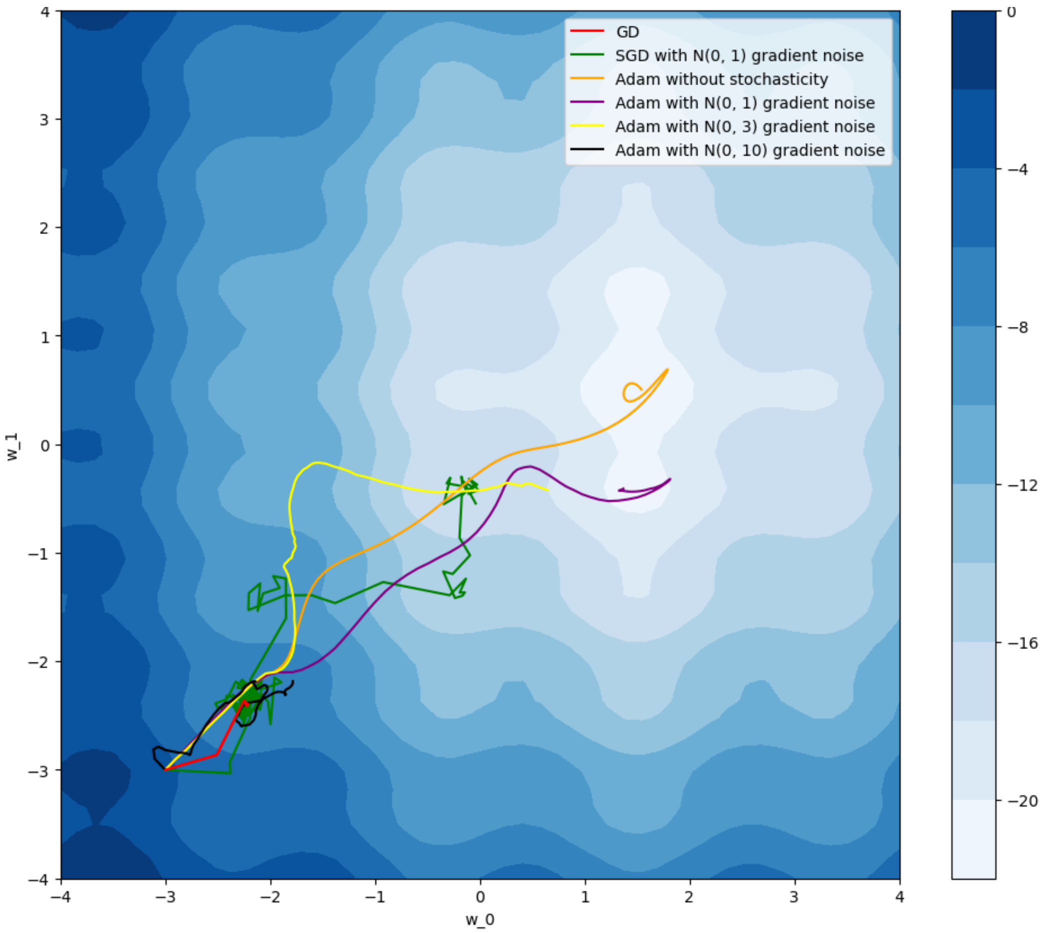


Figure 1: Optimization path over a nonlinear and nonconvex sum of functions with SGD and Adam with different levels of gradient noise. All optimizers have been initialized at $[-3, -3]$, used $\alpha = 0.1$ and taken 100 steps, the Adam optimizers have used $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

In figure 1, it is apparent that the Adam optimizers are pretty good. Finding the global minima of a nonconvex function is a difficult task. Looking at the red curve, we can see that after only 3 steps, the GD algorithm without noise gets stuck in the first local minima and refuses to move significantly for the coming 97 steps. Adding some gradient noise to model the stochastic behavior of sampling batches gets us the green curve. The "SGD" curve also finds the first minima but gets bumped out after a while and moves on to 4 more local minima. Given enough steps, the green curve might even end up in the global minima, highlighting one advantage SGD has over normal gradient descent.

Three out of the four Adam optimizers blaze past this first minima because of their momentum. The orange Adam optimizer without noise converges quickly to the global minima. It has enough momentum to pass the minimum and has to reset its momentum while making a U-turn. The purple optimizer shows a similar behavior but gets bumped into the wrong minima. The Adam optimizers with higher noise struggle a bit more. The yellow curve takes an odd path but will seemingly end up in a pretty deep minimum, while the black one gets completely thrown off by the noise and ends up in the same place as the GD optimizer.

2 Solving large i.i.d. linear least-squares problems

A linear least-squares problem with design matrix $\mathbf{X}$ with gaussian distributed elements $\mathbf{X}_{i,j} \sim \mathcal{N}(0,1)$ and N rows and M columns where $N, M \to \infty$ and $\frac{N}{M} \to \alpha \in (0, \infty)$ can be analyzed using its Marchenko–Pastur distribution [5]. We start by creating two sample covariance matrices depending on the size of α.

$$\mathbf{W}_{\alpha<1} = \frac{1}{M}(\mathbf{X}\mathbf{X}^T) \tag{6}$$

$$\mathbf{W}_{\alpha>1} = \frac{1}{N}(\mathbf{X}^T\mathbf{X}) \tag{7}$$

The Marchenko-Pastur law now states that the eigenvalues of W will fall within [1].

$$\lambda(\mathbf{W}_{\alpha<1}) \in [(1-\sqrt{\alpha})^2, (1+\sqrt{\alpha})^2] \tag{8}$$

$$\lambda(\mathbf{W}_{\alpha>1}) \in [(1-\sqrt{1/\alpha})^2, (1+\sqrt{1/\alpha})^2] \tag{9}$$

Meaning that the singular values of $\mathbf{X}$ falls within

$$\sigma(\mathbf{X}) \in \left[\frac{1-\sqrt{\alpha}}{\sqrt{M}}, \frac{1+\sqrt{\alpha}}{\sqrt{M}}\right] \qquad \textit{if } \alpha < 1 \tag{10}$$

$$\sigma(\mathbf{X}) \in \left[\frac{1-\sqrt{1/\alpha}}{\sqrt{N}}, \frac{1+\sqrt{1/\alpha}}{\sqrt{N}}\right] \qquad \textit{if } \alpha > 1 \tag{11}$$

Which allows us to calculate the condition number of $\mathbf{X}$

$$\kappa(\mathbf{X}) = \frac{\sigma_{max}(\mathbf{X})}{\sigma_{min}(\mathbf{X})} = \begin{cases} \frac{1+\sqrt{\alpha}}{1-\sqrt{\alpha}} & \text{if } \alpha < 1 \\ \frac{1+\sqrt{1/\alpha}}{1-\sqrt{1/\alpha}} & \text{if } \alpha > 1 \end{cases} \tag{12}$$

While κ diverges at $\alpha = 1$ [2]. This means we end up with the relation between α and κ shown in figure 2. The phenomenon shown here is the *double descent* in condition numbers [2].

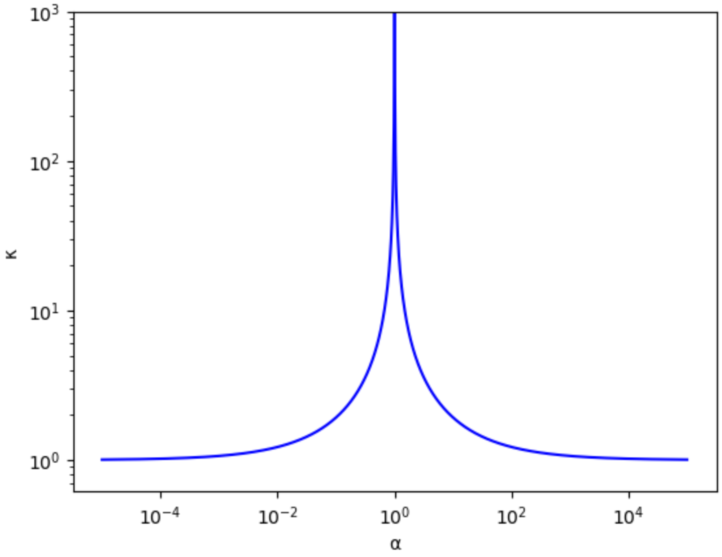


Figure 2: κ over α while $\alpha \in [10^{-5}, 10^5] \setminus \{1\}$

Knowing the time complexity for solving this infinite least squares problem

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 \Leftrightarrow \mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y} \tag{13}$$

might seem ridiculous, since even an $O(min(N, M))$ would be impossible to calculate in the limit. However, the Marchenko-Pastur law is still a good approximation for smaller matrices. In figure 3 we can see that even for matrices with only $MN = 2*10^6$ this seems to be a reasonable estimate of the condition number.

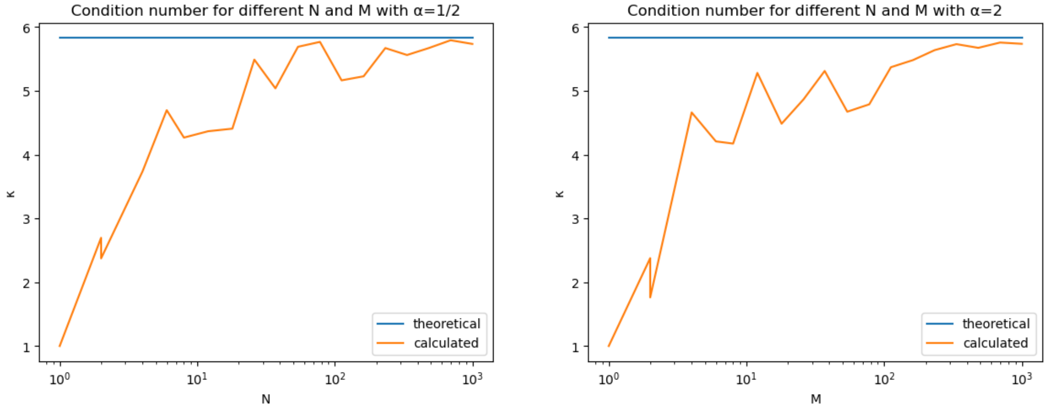


Figure 3: Calculated condition numbers for $\mathbf{X} \in \mathbb{R}^{N \times 2N}$ when $\alpha = 1/2$ or $\mathbf{X} \in \mathbb{R}^{2M \times M}$ when $\alpha = 2$, the theoretical line is given by equation 12.

Solving the linear system 13 with a dense factorization method costs $O\left(NM\min\{N,M\}\right)$ [3]. The cost of a single iteration of gradient descent or conjugate gradient is dominated by a matrix-vector multiplication, $O(NM)$. The convergence of GD with optimal step size is bounded by 14 while CG is bounded by 15 [4].

$$\frac{\|e_i\|_{\mathbf{A}}}{\|e_0\|_{\mathbf{A}}} \leq c^{-2i/\kappa(\mathbf{A})} \tag{14}$$

$$\frac{\|e_i\|_{\mathbf{A}}}{\|e_0\|_{\mathbf{A}}} \leq c^{-2i/\sqrt{\kappa(\mathbf{A})}} \tag{15}$$

Where i is the number of iterations and $\mathbf{A} = \mathbf{X}^T\mathbf{X}$ and therefore $\kappa(\mathbf{A}) = \kappa(\mathbf{X})^2$. If we allow for a relative error:

$$\epsilon = \frac{\|e_i\|_{\mathbf{A}}}{\|e_0\|_{\mathbf{A}}} \tag{16}$$

We can then reach this relative error by doing the following amounts of steps:

$$i \geq \frac{ln(1/\epsilon)\kappa(\mathbf{X})^2}{2} \qquad \text{for GD} \tag{17}$$

$$i \geq \frac{ln(1/\epsilon)\kappa(\mathbf{X})^2}{2} \qquad \text{for CG} \tag{18}$$

Meaning that the time complexity to achieve a relative error ϵ for gradient descent with optimal step size is $O(ln(1/\epsilon)\kappa(\mathbf{X})^2MN)$ and for conjugate gradient it is $O(ln(1/\epsilon)\kappa(\mathbf{X})MN)$.

References

[1] J. Bryson, R. Vershynin, and H. Zhao. Marchenko-pastur law with relaxed independence conditions, 2021.

[2] T. Poggio, G. Kur, and A. Banburski. Double descent in the condition number, 2020.

[3] R. Webber. Lecture 2 in math216a, 2024.

[4] R. Webber. Lecture 8-10 in math216a, 2024.

[5] Wikipedia contributors. Marchenko–pastur distribution — Wikipedia, the free encyclopedia, 2024. [Online; accessed 11-November-2024].

[6] Wikipedia contributors. Stochastic gradient descent — Wikipedia, the free encyclopedia, 2024. [Online; accessed 8-November-2024].

[7] Y. Zhang, C. Chen, N. Shi, R. Sun, and Z.-Q. Luo. Adam can converge without any modification on update rules, 2023.