Gradient Spread as an Intuition for the Adam Optimizer

David Eigen

May 2025

1 Introduction

The Adam optimizer [1] is a variant of stochastic gradient descent that makes use of stochasticity of samples to modulate step size. When the calculated gradient is relatively constant between samples, step size is large; when gradients vary between samples, step size decreases. The basic notion is that if steps are similar from one sample to the next — and in particular, if they keep going in the same direction — we can take large steps in that direction, since a sequence of small steps of varying magnitude all in the same direction are they same as one large step. Conversely, when gradients are different from one sample to the next — and in particular, are in different directions —, we need to slow down and take smaller steps in order not to jump over a curved region and accumulate samples for better estimates.

The Adam optimizer tracks averages

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

where g_t is the gradient at step t, and sets the parameter update step to

$$\Delta_t = m_t / \sqrt{v_t}\ , \quad \theta_t = \theta_{t-1} - \alpha \Delta_t,$$

which normalizes the momentum gradient by an estimate of the "second moment". But what is the significance of this normalization term?

2 Influence of Gradient Spread

Some limiting examples can help illustrate this. Say σ_t is the standard deviation of the gradient samples at step t, and that m_t and v_t correspond to estimates of the recent mean and second moment, so that $\sigma_t^2 = E[g^2] - E[g]^2 = v_t - m_t^2$. This is not strictly the case in practice (described below in the experiments), but we'll use it for these illustrations.

Phrased in terms of the standard deviation, the update is

$$\Delta_t = \frac{m_t}{\sqrt{v_t}} = \frac{m_t}{\sqrt{\sigma_t^2 + m_t^2}}$$

First, consider the case where every gradient sample is the same (this corresponds to a linear region in the function being optimized). Then the standard deviation between samples is $\sigma = 0$, and the update is

$$\Delta_t = \frac{m_t}{\sqrt{m_t^2}} = \pm 1$$

where $\pm 1 = sign(m_t)$. This means the sign of each element in the average gradient determines the direction of the step, but steps always have the same magnitude. If two different parameters (components) have different sized gradients, but each is constant along a recent set of updates, we'll use the same update size for each of them. And why not? If there isn't any variation to the gradient samples as we step, it makes sense to take the maximum step along each dimension, regardless of gradient magnitude.

However, a function whose gradient is always the same isn't very interesting (indeed, it must be linear if the derivative is constant). Now suppose the gradients change between samples, and that their variation is measured by $\sigma_t = \gamma_t |m_t|$. Here we phrase the standard deviation as a multiple of the mean. γ_t is sometimes known as the "spread" and describes variance factoring out the magnitude of the sample values.

In this case where $\sigma_t = \gamma_t |m_t|$, the update is

$$\Delta_t = \frac{m_t}{\sqrt{\sigma_t^2 + m_t^2}} = \frac{m_t}{\sqrt{\gamma_t^2 m_t^2 + m_t^2}} = \frac{\pm 1}{\sqrt{\gamma_t^2 + 1}}$$

again where $\pm 1 = sign(m_t)$.

The size of this step does not depend on the magnitude of the gradient — only the spread, which is determined by the variance of samples relative to the mean and not directly related to magnitudes. The only place the actual gradient value is used is in the numerator, $\pm 1 = sign(m_t)$. That is: the running mean gradient determines the *direction* of the step (i.e. its sign, positive or negative), while *the spread determines the size.*

3 Experiments

The above analysis shows a separation between the determining factors for sign and size, but it supposes ideal conditions where m_t and v_t correspond to the true mean and second moment at the current time step. In practice, these are estimated using exponential averages, which may not match the true values, particularly since they track "behind" the current timestep where the estimates are applied, and furthermore, are estimated at different rates ($\beta_1 < \beta_2$). How much does the intuition behind separation of direction and size apply in practice?

Here, we run a few experiments on small models, showing that this view of the algorithm still applies when these assumptions are relaxed, though in a softer form. In particular, the gradient mean has effect on the size of the updates (which is likely beneficial in practice), though as might be expected, not as much as in vanilla gradient descent. The relationships between step, mean and spread identified before are clearly visible.

This set of measurements uses a small CNN on CIFAR-10: only two conv layers and two fc layers with maxpooling, $conv1 \rightarrow maxpool \rightarrow conv2 \rightarrow maxpool \rightarrow fc1 \rightarrow fc2$, with batchnorm and relu between each weighted layer. We look at $\beta_1 = \beta_2 = 0.99$, so that each m_t and v_t vary at the same rate, as well as the usual values $\beta_1 = 0.9, \beta_2 = 0.999$, and compare to gradient descent with momentum equal to β_1. In each case, we measure step sizes using difference between weight values before and after each update, and calculate $\gamma_t = \sqrt{v_t - m_t^2}$ using the exponential averages for m_t and v_t. Scatter plots show one dot for each update step t. Each dot measures the mean absolute value of gradient mean or spread, first taken within each layer, then across layers. We only measure values for conv and fc weights W, excluding bias and batchnorm parameter updates.

Figure 1 shows measurements for the same-rate case $\beta_1 = \beta_2 = 0.99$. While the mean gradient corresponds directly to the update step for SGD, there is little relation between them for Adam, after an initial period at the start of training (the color of each dot indicates the step number t). Measured against spread γ_t, the step follows closely to the shape of theoretical step $1/\sqrt{\gamma_t^2 + 1}$ (dotted line) for Adam, while much more diffuse for SGD.

Figure 2 shows measurements for the more typically practical case of $\beta_1 = 0.9$, $\beta_2 = 0.999$. Here, the relation between spread and step is still visible, but weaker — the mean gradient has more influence of the step size, due to the faster estimation rate.

Another interesting to note, particularly visible in Figure 1, is that at later training steps (lighter colors), the update steps for Adam don't become consistently smaller as they do for SGD, but tend to move around, going up and down slightly as the sampling variation increases and decreases. When update steps are smaller, there is less gradient variation between successive samples due the smaller change in parameters between sample times. This lower variation causes updates to increases, which increases variation, in turn lowering updates back down in a cycle, causing oscillations.

4 Oscillation Near an Optimum

Variation (and hence spread) in the gradients can come from two sources: the change in calculated gradient due to the function weights being updated by the last optimization step, or randomization in choosing input samples or other randomization in the function. In the case where all variation comes the parameter updates themselves (e.g., full batch and no randomization in the function), what is the behavior of this around an optimum? Do updates slow down and

CIFAR-10, $\beta_1 = \beta_2 = 0.99$

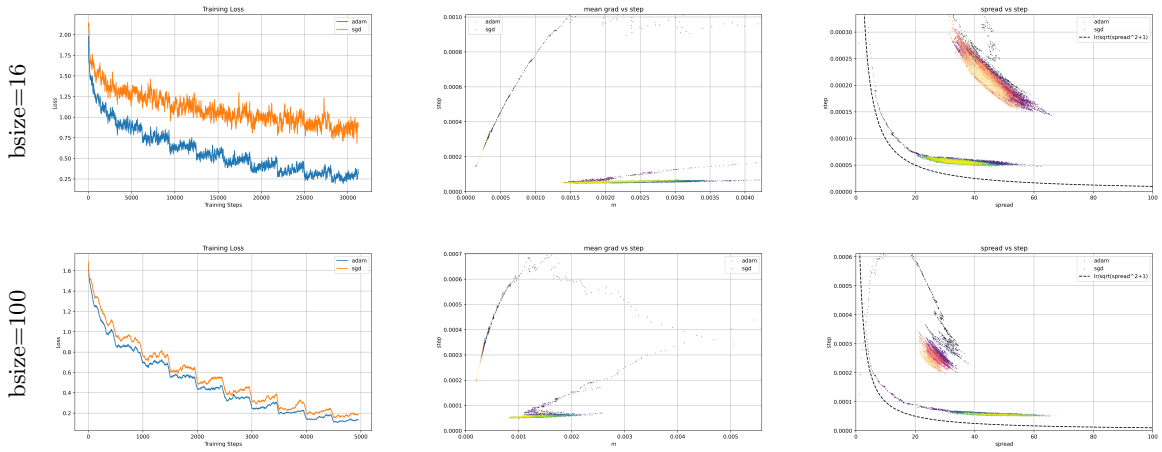

Figure 1: CIFAR-10, $\beta_1 = \beta_2 = 0.99$. Top row: batch size 16, bottom row: batch size 100. (a) Loss curve, (b) mean gradient m_t vs average update size, (c) spread of gradient γ_t vs update size. Color corresponds to step number t.

CIFAR-10, $\beta_1 = 0.9, \beta_2 = 0.999$

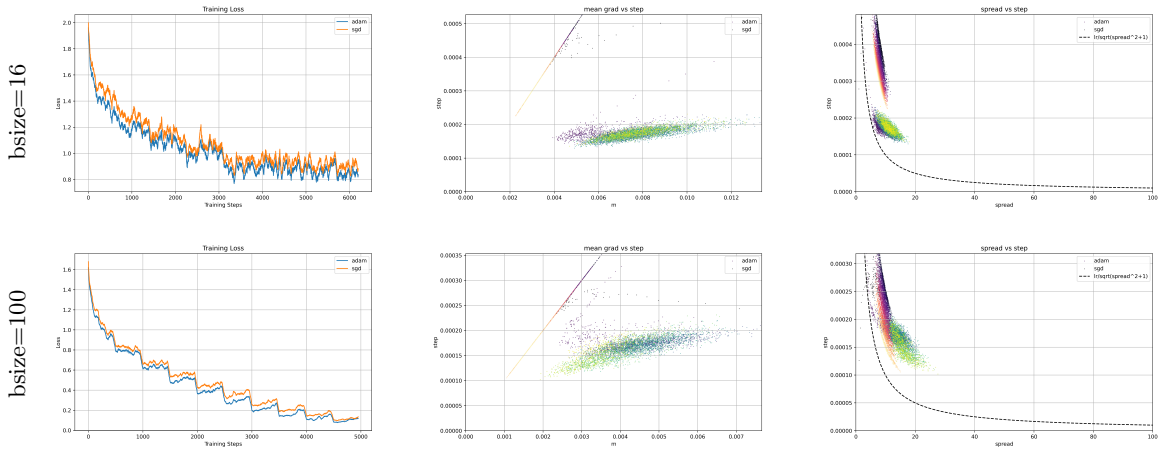

Figure 2: CIFAR-10, $\beta_1 = 0.9, \beta_2 = 0.999$. Top row: batch size 16, bottom row: batch size 100. (a) Loss curve, (b) mean gradient m_t vs average update size, (c) spread of gradient γ_t vs update size. Color corresponds to step number t.

Quadratic $f(x) = x^2$, $\beta_1 = \beta_2 = 0.99$

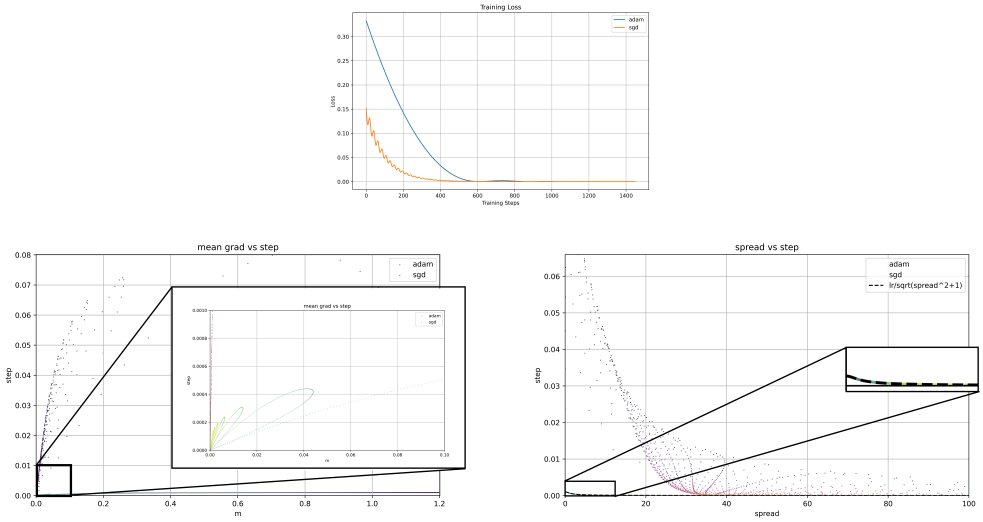

Figure 3: Optimization of the quadratic $f(x) = x^2$: (a) Loss curve, (b) mean gradient m_t vs average update size, (c) spread of gradient γ_t vs update size

converge?

In vanilla gradient descent, the update is the gradient. For a convex function and fixed learning rate that is appropriately small enough, each successive value θ_t will be closer to the optimum than the previous one. So the gradient and step $\Delta_t = g_t$ naturally goes to 0 and the process converges.

In the view of Adam optimizer where the update is $\Delta_t = \pm 1/\sqrt{\gamma_t^2 + 1}$, it isn't immediately obvious whether the updates converge, since they depend not on the value of the gradient, but the amount of variation.

This oscillation behavior, even converging towards the optimum, can be clearly seen in a toy example optimizing $f(x) = x^2$. Figure 3(b) shows the mean absolute value of the gradient df/dx versus step Δ_t. While this zips straight to 0 for vanilla gradient descent, the Adam optimizer oscillates, with mean gradient decreasing, passing by 0, and growing out again, since step sizes are determined by spread. Figure 3(c) shows relation between spread and step size — step values for gradient descent are fairly disperse those for Adam perfectly coincide with the curve $1/\sqrt{\gamma^2 + 1}$ (in fact one may need to look closely to see this).

References

[1] Diederik P Kingma. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).