- In SGD, a fixed learning rate might cause large adjustments to the parameters, ϕ, especially when associated with large gradients.
- To avoid the risk of divergence, a more conservative learning rate is often required. However, this can lead to slow progress in the optimization process.

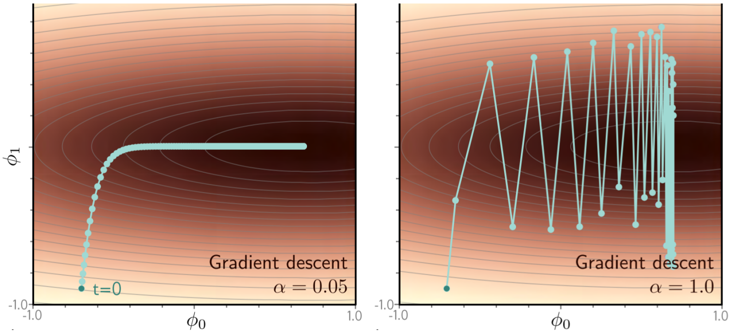

Figure: The trade-off between small and large learning rates in SGD.[1]

[1] Adopted from the book, Understanding Deep Learning

- First Idea: Move a fixed distance in each direction.
- In this context, operations like square root and division are performed element-wise on the parameters.

$$
\begin{aligned}
m_{t+1} &\leftarrow \frac{\partial L[\phi_t]}{\partial \phi} \\
v_{t+1} &\leftarrow \left(\frac{\partial L[\phi_t]}{\partial \phi} \right)^2 \\
\phi_{t+1} &\leftarrow \phi_t - \alpha \cdot \frac{m_{t+1}}{\sqrt{v_{t+1}} + \epsilon}
\end{aligned}
$$

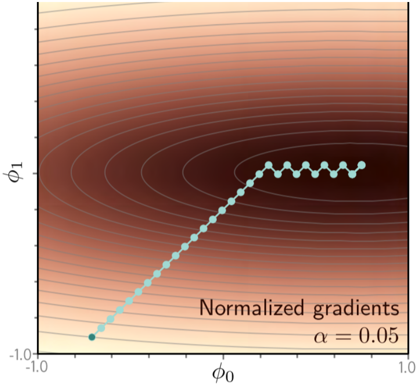

Figure: Illustration of moving a fixed distance in each epoch. [2]

[2]Adopted from the book, Understanding Deep Learning

- Second Idea: Incorporate momentum into the optimization process.

$$
\begin{aligned}
m_{t+1} &\leftarrow \beta \cdot m_t + (1-\beta)\frac{\partial L[\phi_t]}{\partial \phi} \\
v_{t+1} &\leftarrow \gamma \cdot v_t + (1-\gamma)\left(\frac{\partial L[\phi_t]}{\partial \phi}\right)^2 \\
\phi_{t+1} &\leftarrow \phi_t - \alpha \cdot \frac{m_{t+1}}{\sqrt{v_{t+1}} + \epsilon}
\end{aligned}
$$

- Let's denote $\frac{\partial L[\phi_t]}{\partial \phi}$ by g_t.
- Then, we have:

$$
\begin{aligned}
m_0 &= 0 \\
m_1 &= (1-\beta)g_0 \\
m_2 &= (1-\beta)g_1 + (1-\beta)\beta g_0 \\
m_3 &= (1-\beta)\beta^2 g_0 + (1-\beta)\beta g_1 + (1-\beta)g_2 \\
&\vdots \\
m_t &= (1-\beta)\Sigma_{i=0}^{t-1}\beta^{t-i-1}g_i
\end{aligned}
$$

- It can be shown that

$$\mathbb{E}[m_{t+1}] \approx (1-\beta^{t+1})\mathbb{E}[g_t]$$

- In the initial epochs, the momentum could be zero. This issue leads us to the third idea, which is the Adam optimization algorithm.

- Adam is used normally with the mini-batches.

$$
\begin{aligned}
m_{t+1} &\leftarrow \beta \cdot m_t + (1-\beta) \sum_{i \in B_t} \frac{\partial l_i(\phi_t)}{\partial \phi} \\
v_{t+1} &\leftarrow \gamma \cdot v_t + (1-\gamma) \sum_{i \in B_t} \left(\frac{\partial l_i(\phi_t)}{\partial \phi} \right)^2 \\
\tilde{m}_{t+1} &\leftarrow \frac{m_{t+1}}{1-\beta^{t+1}} \\
\tilde{v}_{t+1} &\leftarrow \frac{v_{t+1}}{1-\gamma^{t+1}} \\
\phi_{t+1} &\leftarrow \phi_t - \alpha \cdot \frac{\tilde{m}_{t+1}}{\sqrt{\tilde{v}_{t+1}} + \epsilon}
\end{aligned}
$$

- In practice, $\beta = 0.9$, $\gamma = 0.999$ and $\epsilon = 10^{-7}$ are good starting points.

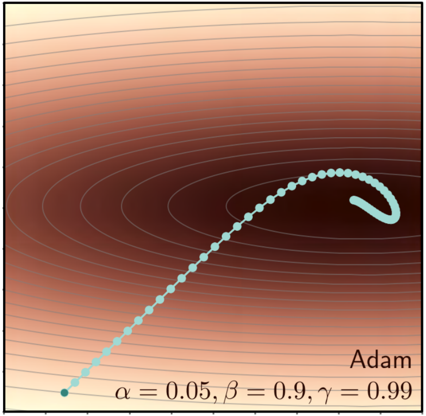

Figure: Adam creates a smoother path.[3]

[3] Adopted from the book, Understanding Deep Learning