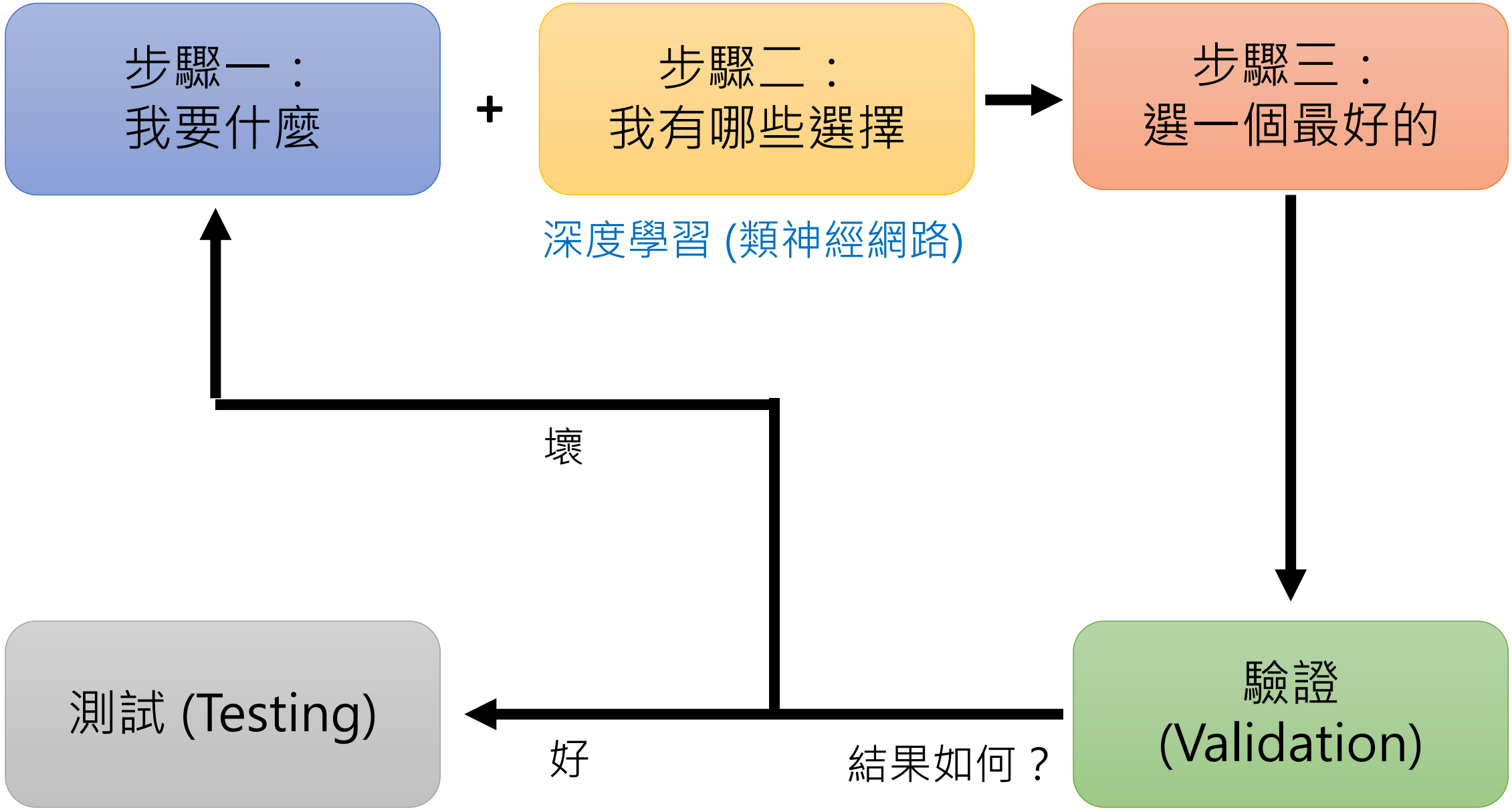
一堂課看懂訓練類神經網路的各種訣竅

李宏毅


步驟一：
我要什麼
+
步驟二：
我有哪些選擇
深度學習 (類神經網路)
步驟三：
選一個最好的
驗證
(Validation)
結果如何？
好
壞
測試 (Testing)

介紹各種訓練類神經網路常用技巧

- 聽到一個跟訓練類神經網路有關的方法時，你要這樣問自己

方法名	改了那一個步驟	帶來什麼好處
……	……	……

步驟一：
我要什麼

+

步驟二：
我有哪些選擇

→

步驟三：
選一個最好的

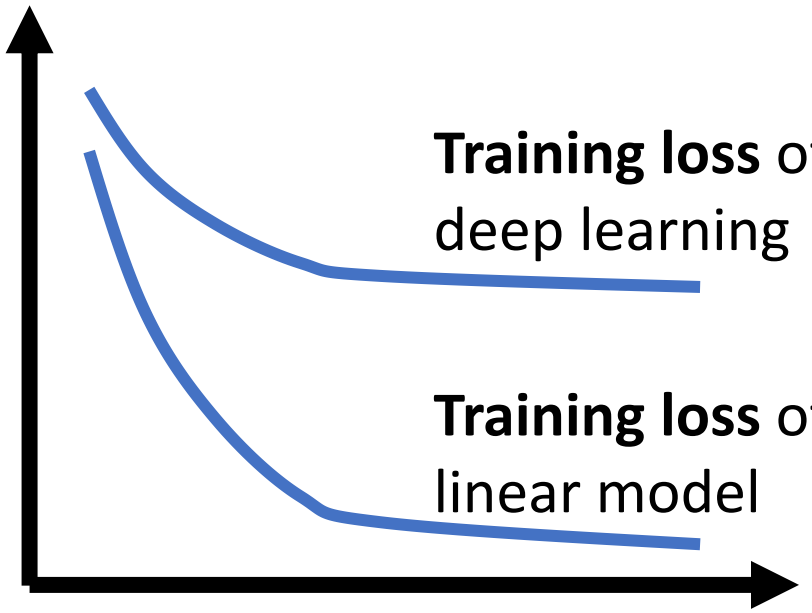
介紹各種訓練類神經網路常用技巧

- 聽到一個跟訓練類神經網路有關的方法時，你要這樣問自己

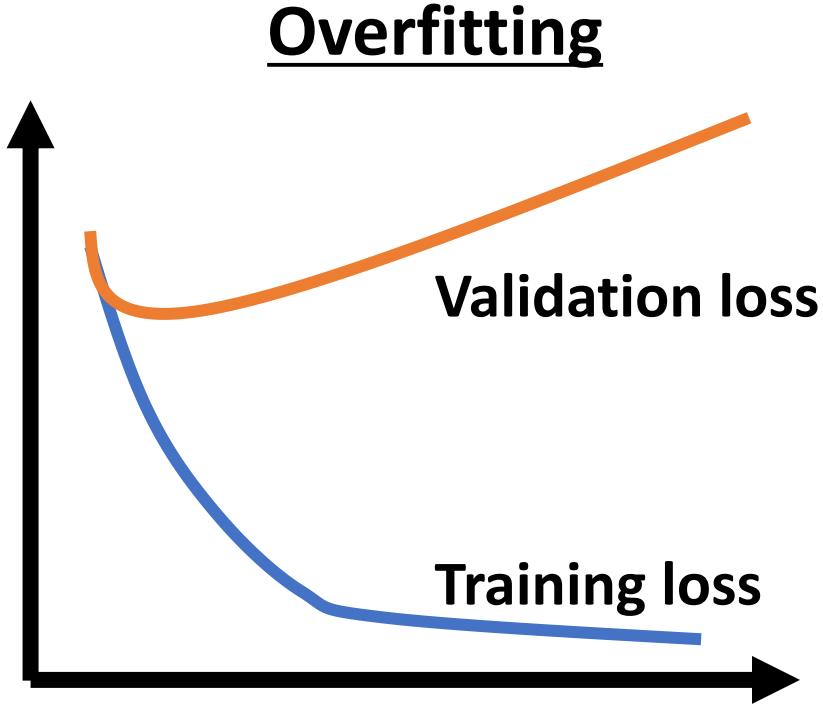
方法名	改了那一個步驟	帶來什麼好處
……	……	……

  - Better Optimization：更低的 Training Loss
  - Better Generalization：更低的 Validation Loss
  - ……

選擇合適的技巧

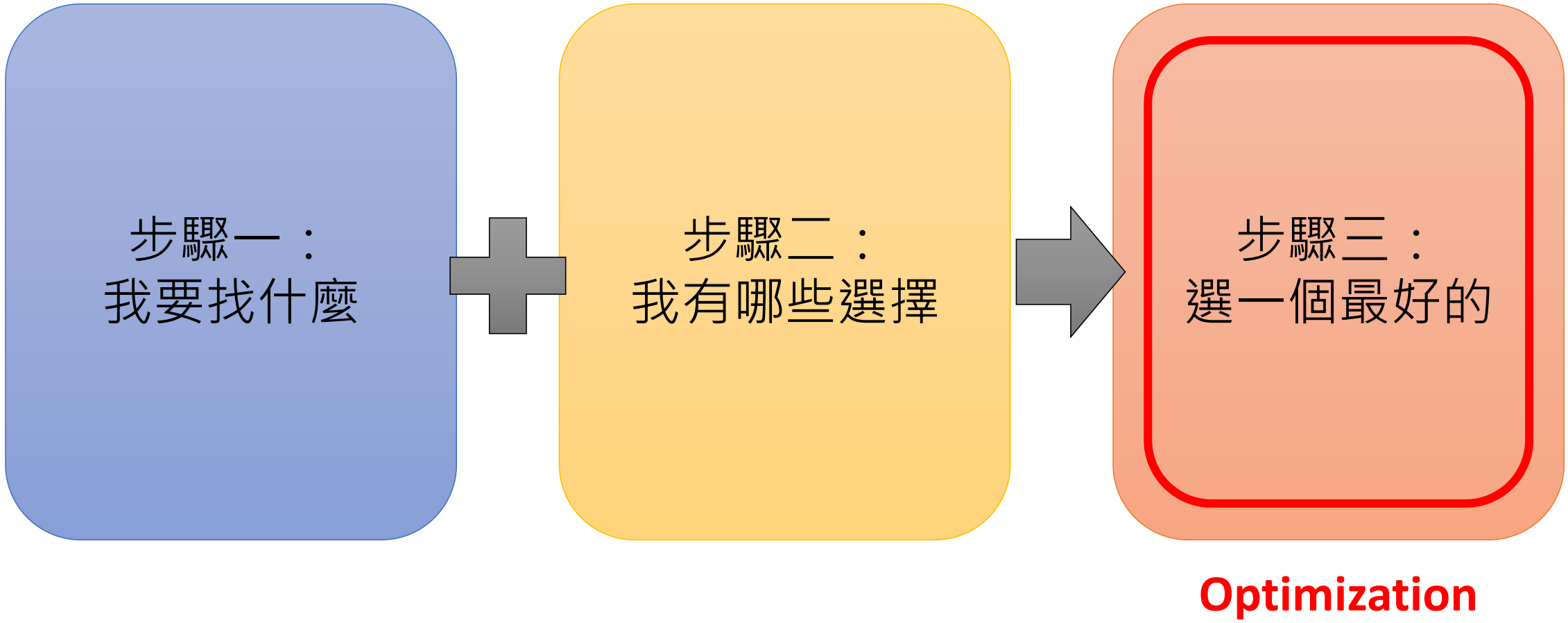


Better Optimization!



Better Generalization!

找函式步驟 3 + 1



Vanilla Gradient Descent

$$\boldsymbol{\theta}^* = arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

- (Randomly) Pick initial values $\boldsymbol{\theta^0}$
- Compute gradient $\boldsymbol{g^0} = \nabla L(\boldsymbol{\theta^0})$

$$\boldsymbol{\theta^1} \leftarrow \boldsymbol{\theta^0} - \eta \boldsymbol{g^0}$$

- Compute gradient $\boldsymbol{g^1} = \nabla L(\boldsymbol{\theta^1})$

$$\boldsymbol{\theta^2} \leftarrow \boldsymbol{\theta^1} - \eta \boldsymbol{g^1}$$

- Compute gradient $\boldsymbol{g^2} = \nabla L(\boldsymbol{\theta^2})$

$$\boldsymbol{\theta^3} \leftarrow \boldsymbol{\theta^2} - \eta \boldsymbol{g^2}$$

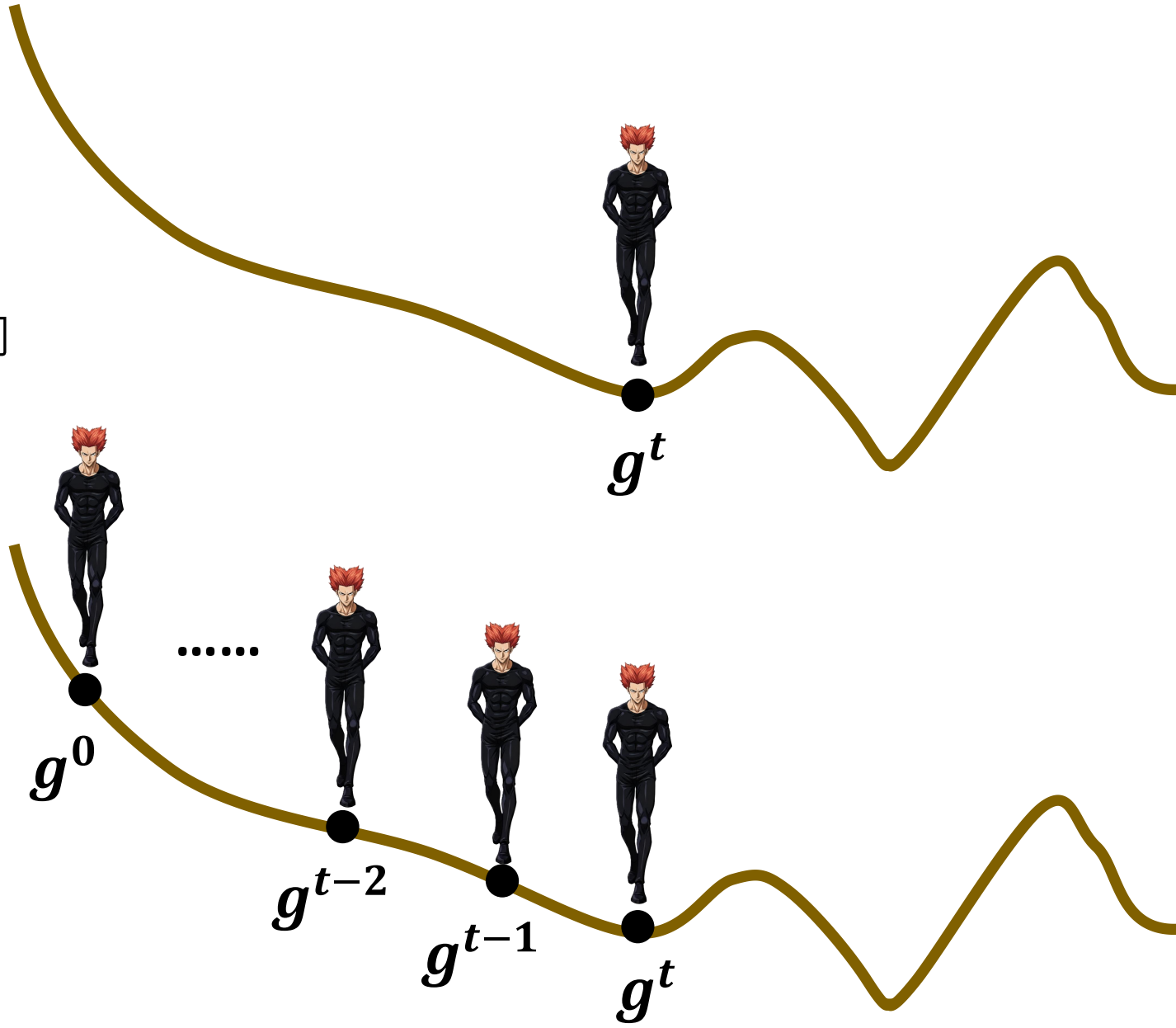
Optimizer

Vanilla Gradient Descent

根據當下算出來的 $\boldsymbol{g^t}$ 來決定方向

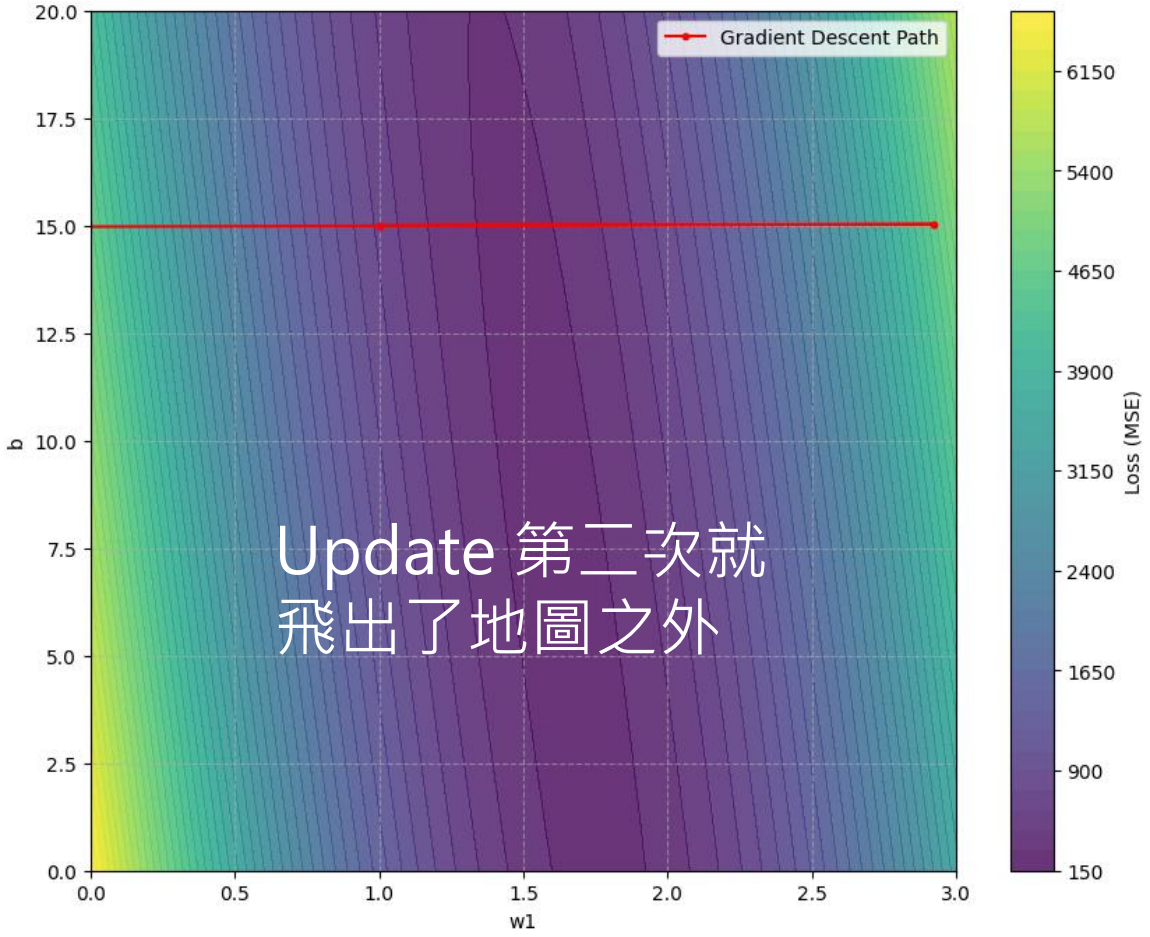
Gradient Descent + Optimizer

根據 $\boldsymbol{g^0, g^1, g^2, \ldots, g^t}$
一起來決定方向

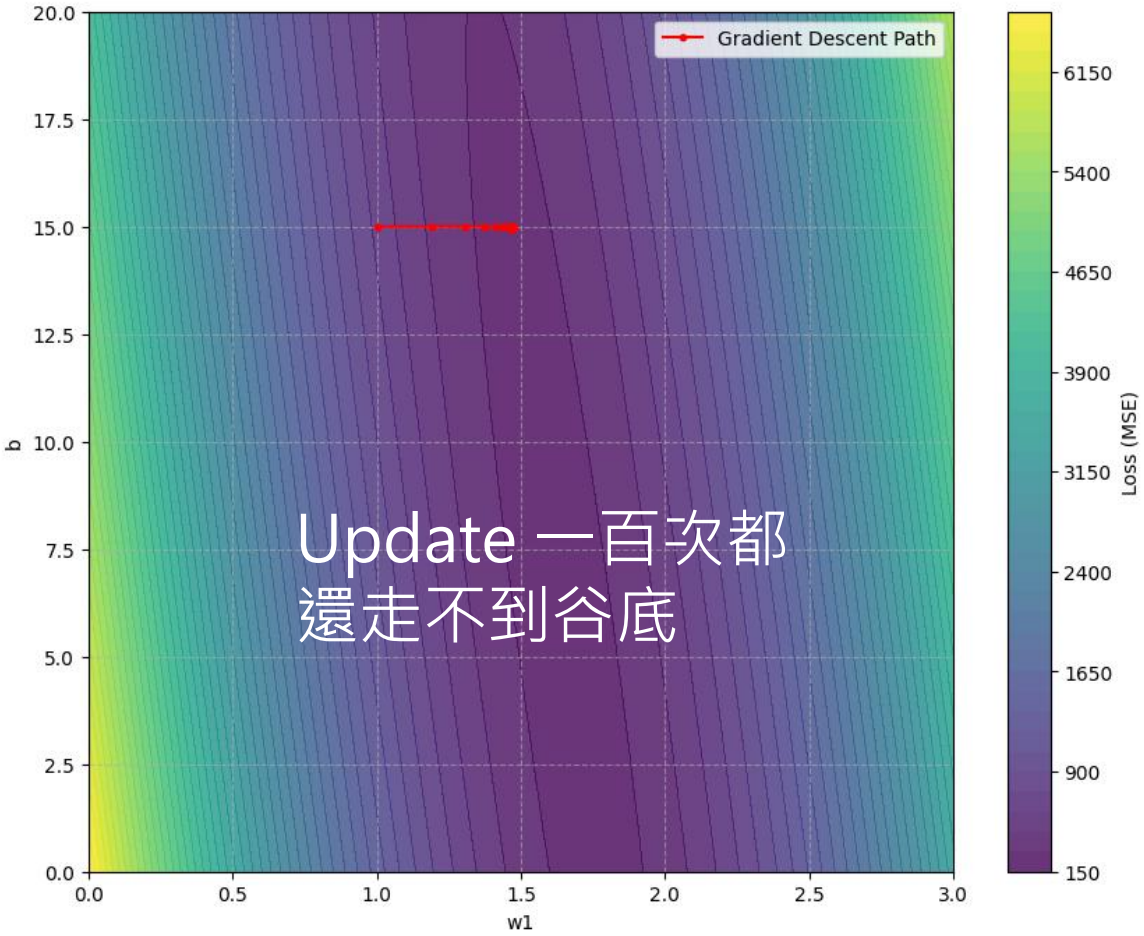


Learning Rate 實在是很難調 ...

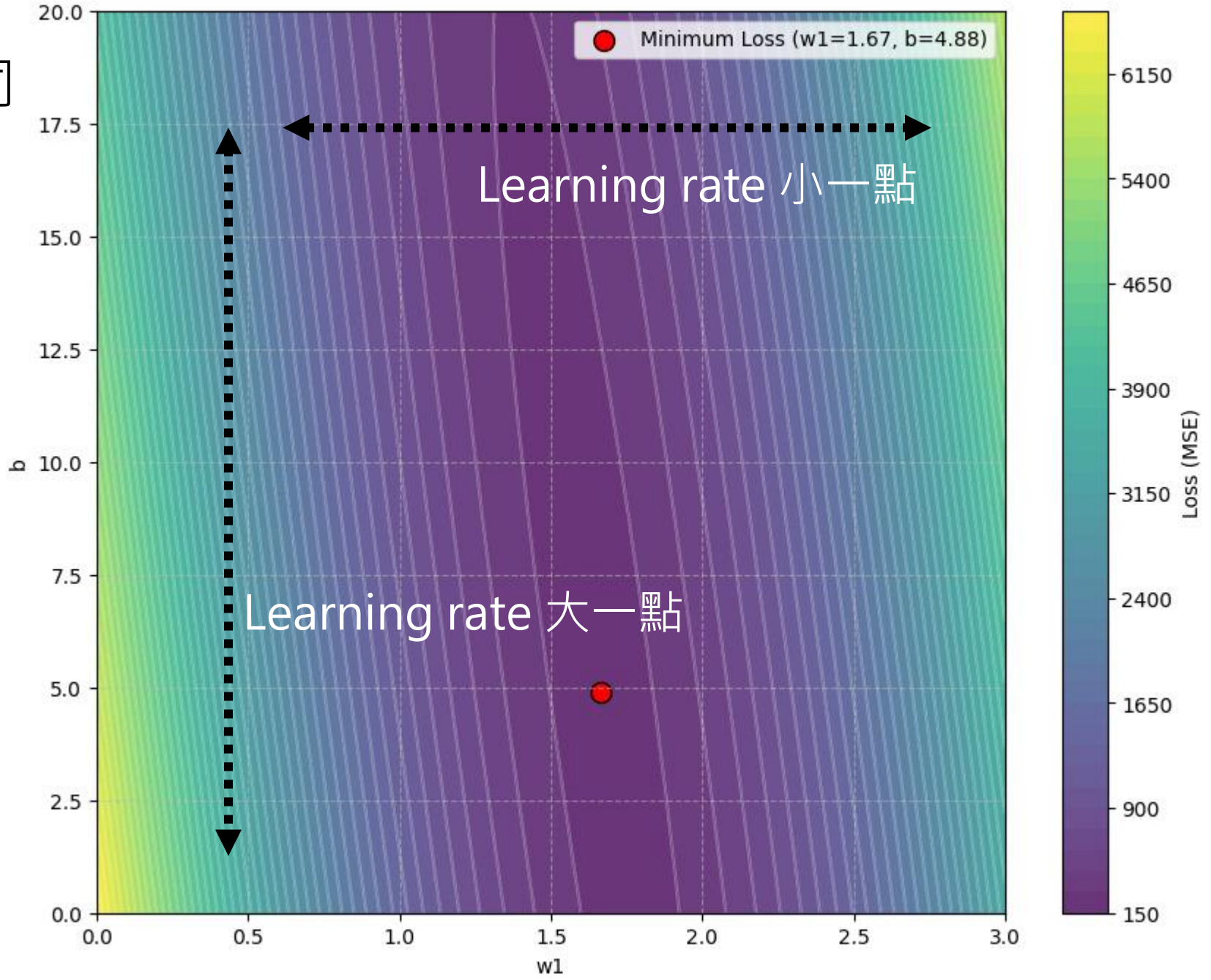
$\eta = 0.001$



$\eta = 0.0001$

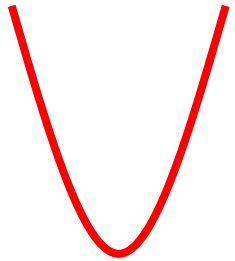


不同參數應該要有不同的 Learning Rate



怎麼知道那一個方向的 gradient 大、那一個小？

用過去的 Gradient 來決定 Learning Rate



Learning rate 小一點

$$g^0 = \begin{bmatrix} 500 \\ 0.4 \end{bmatrix} \quad g^1 = \begin{bmatrix} 432 \\ -0.3 \end{bmatrix} \quad g^2 = \begin{bmatrix} -211 \\ 0.2 \end{bmatrix} \quad g^3 = \begin{bmatrix} 139 \\ 0.1 \end{bmatrix}$$

Learning rate 大一點

Adagrad

- Compute gradient $\boldsymbol{g^0} = \nabla L(\boldsymbol{\theta^0})$

  For each dimension i: $\sigma_i^0 = \sqrt{(g_i^0)^2} = |g_i^0|$

  $\theta_i^1 \leftarrow \theta_i^0 - \frac{\eta}{\sigma_i^0} g_i^0$

- Compute gradient $g^1 = \nabla L(\boldsymbol{\theta}^1)$

  Average?

  For each dimension i: $\sigma_i^1 = \sqrt{\left[(g_i^0)^2 + (g_i^1)^2\right]}$

  $\theta_i^2 \leftarrow \theta_i^1 - \frac{\eta}{\sigma_i^1} g_i^1$

- Compute gradient $g^2 = \nabla L(\boldsymbol{\theta}^2)$

  For each dimension i: $\sigma_i^2 = \sqrt{\left[(g_i^0)^2 + (g_i^1)^2 + (g_i^2)^2\right]}$

  $\theta_i^3 \leftarrow \theta_i^2 - \frac{\eta}{\sigma_i^2} g_i^2$

⋮

- Compute gradient $\boldsymbol{g^t} = \nabla L(\boldsymbol{\theta}^t)$

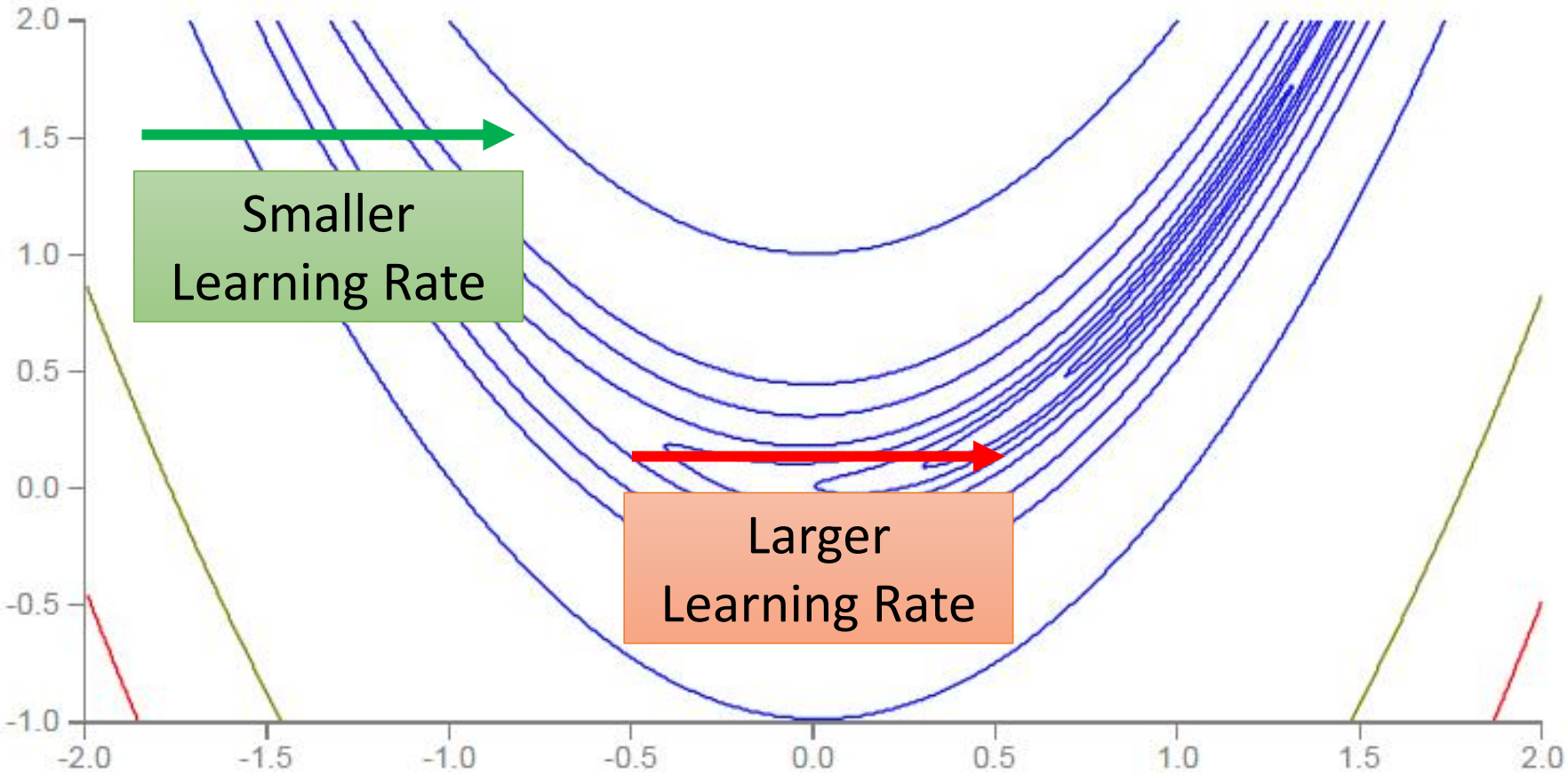
  For each dimension i: $\sigma_i^t = \sqrt{\sum_{i=0}^{t} (g_i^t)^2}$

  $\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t$

範例程式

連結：
https://colab.research.google.com/drive/1XPIU-I77dXL9W74jnevmfb8K8XoEPRso?usp=sharing

同一個參數的Gradient 大小不會一成不變



同一個參數的Gradient 大小不會一成不變

$$g^0 = \begin{bmatrix} 500 \\ 0.4 \end{bmatrix} \quad g^1 = \begin{bmatrix} 432 \\ -0.3 \end{bmatrix} \quad \cdots\cdots \quad g^{t-1} = \begin{bmatrix} -0.1 \\ 229 \end{bmatrix} \quad g^t = \begin{bmatrix} 0.1 \\ 100 \end{bmatrix}$$

變大　　變大

Adagrad: 全部平方加起來

RMSProp: 最近算出來的 gradient 給比較大的影響

RMSProp

- Compute gradient $\boldsymbol{g^0} = \nabla L(\boldsymbol{\theta^0})$

  For each dimension i: $\sigma_i^0 = \sqrt{(g_i^0)^2}$

  $\theta_i^1 \leftarrow \theta_i^0 - \frac{\eta}{\sigma_i^0} g_i^0$

- Compute gradient $\boldsymbol{g^1} = \nabla L(\boldsymbol{\theta^1})$

  $0 < \alpha < 1$

  For each dimension i: $\sigma_i^1 = \sqrt{\alpha(\sigma_i^0)^2 + (1-\alpha)(g_i^1)^2}$

  $\theta_i^2 \leftarrow \theta_i^1 - \frac{\eta}{\sigma_i^1} g_i^1$

- Compute gradient $\boldsymbol{g^2} = \nabla L(\boldsymbol{\theta^2})$

  For each dimension i: $\sigma_i^2 = \sqrt{\alpha(\sigma_i^1)^2 + (1-\alpha)(g_i^2)^2}$

  $\theta_i^3 \leftarrow \theta_i^2 - \frac{\eta}{\sigma_i^2} g_i^2$

⋮

- Compute gradient $\boldsymbol{g^t} = \nabla L(\boldsymbol{\theta^t})$

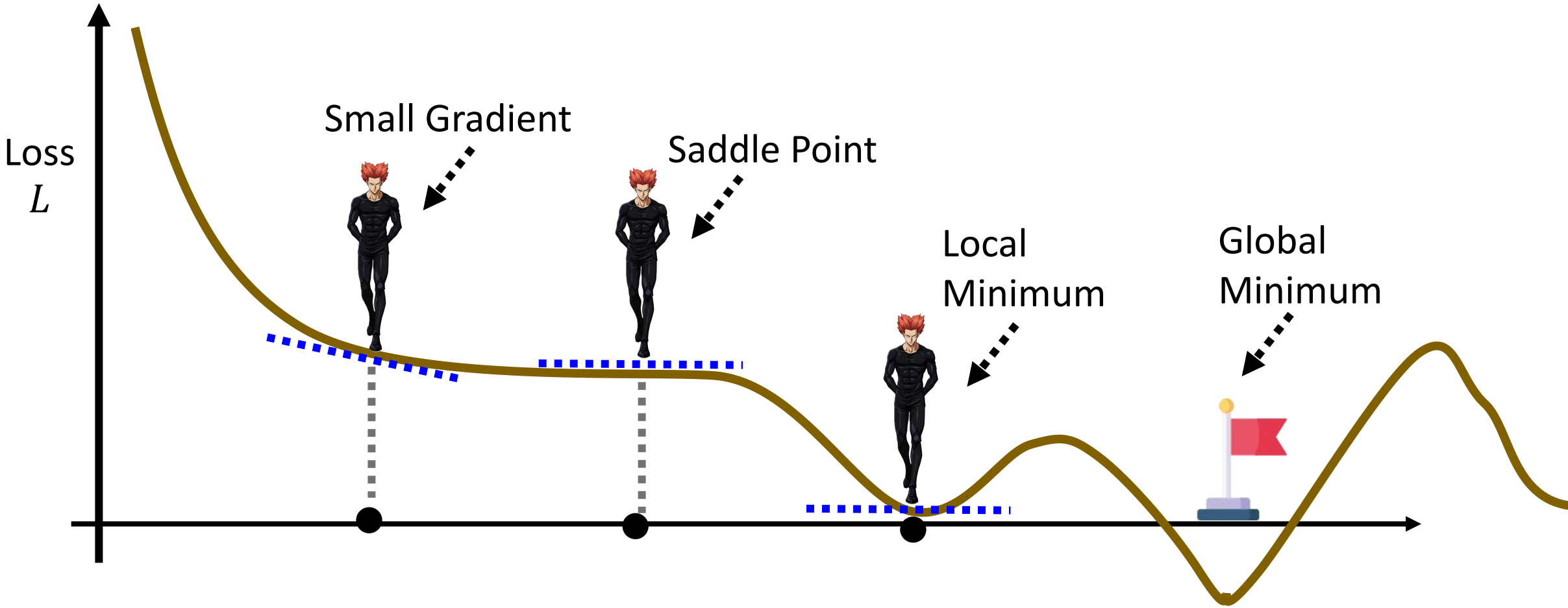
  For each dimension i: $\sigma_i^t = \sqrt{\alpha(\sigma_i^{t-1})^2 + (1-\alpha)(g_i^t)^2}$

  $\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t$

範例程式

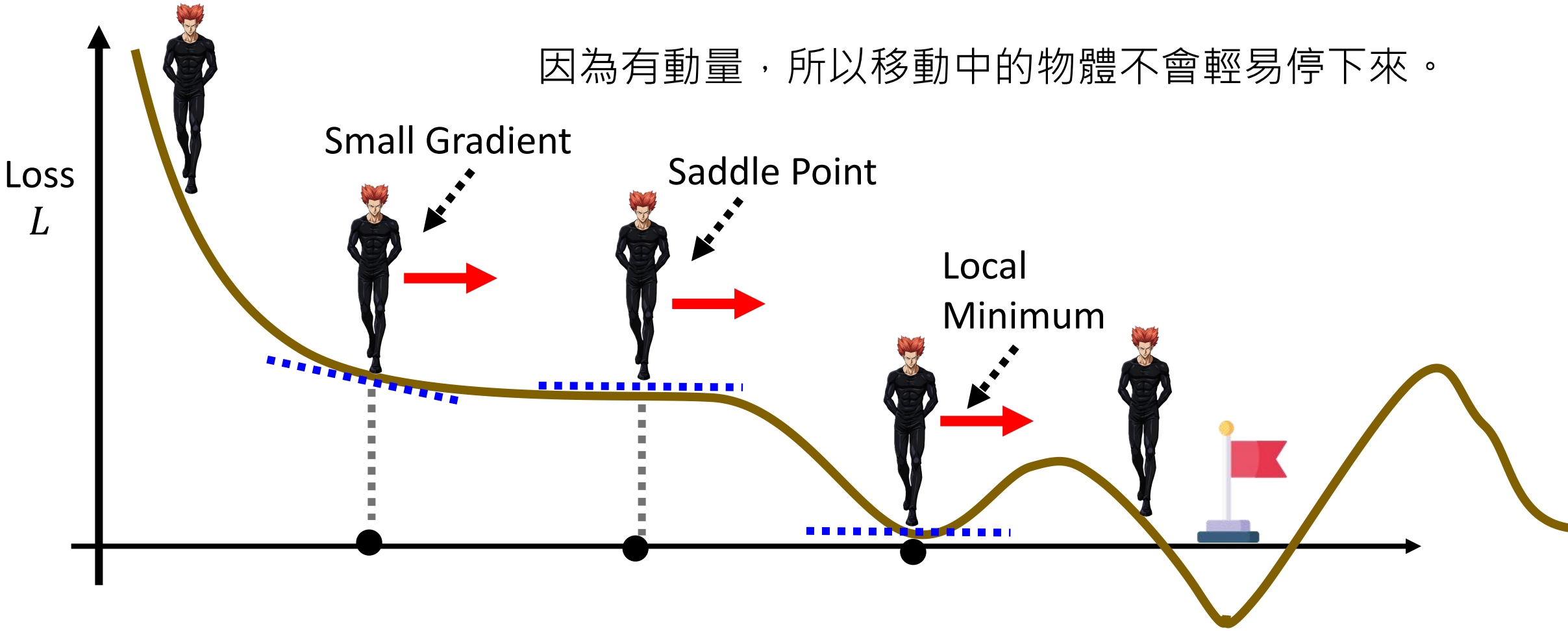
連結：
https://colab.research.google.com/drive/1XPIU-I77dXL9W74jnevmfb8K8XoEPRso?usp=sharing

Optimization 會在 Gradient 很小時停止



考慮動量 (Momentum)

因為有動量，所以移動中的物體不會輕易停下來。



Momentum

- Compute gradient $\boldsymbol{g^0} = \nabla L(\boldsymbol{\theta^0})$

  For each dimension i: $m_i^0 = g_i^0$ $\quad \theta_i^1 \leftarrow \theta_i^0 - \eta m_i^0$

- Compute gradient $\boldsymbol{g^1} = \nabla L(\boldsymbol{\theta^1})$

  For each dimension i: $m_i^1 = g_i^0 + g_i^1$ $\quad \theta_i^2 \leftarrow \theta_i^1 - \eta m_i^1$

- Compute gradient $\boldsymbol{g^2} = \nabla L(\boldsymbol{\theta^2})$

  For each dimension i: $m_i^2 = g_i^0 + g_i^1 + g_i^2$ $\quad \theta_i^3 \leftarrow \theta_i^2 - \eta m_i^2$

⋮

- Compute gradient $\boldsymbol{g^t} = \nabla L(\boldsymbol{\theta^t})$

  For each dimension i: $m_i^t = g_i^0 + g_i^1 + g_i^2 + \ldots + g_i^t$ $\quad \theta_i^{t+1} \leftarrow \theta_i^t - \eta m_i^t$

Momentum (這不是經典的 Momentum)

- Compute gradient $\boldsymbol{g^0} = \nabla L(\boldsymbol{\theta^0})$

  For each dimension i: $m_i^0 = g_i^0$ $\quad \theta_i^1 \leftarrow \theta_i^0 - \eta m_i^0$

- Compute gradient $\boldsymbol{g^1} = \nabla L(\boldsymbol{\theta^1})$ $\quad 0 < \beta < 1$

  For each dimension i: $m_i^1 = \beta m_i^0 + (1-\beta) g_i^1$ $\quad \theta_i^2 \leftarrow \theta_i^1 - \eta m_i^1$

- Compute gradient $\boldsymbol{g^2} = \nabla L(\boldsymbol{\theta^2})$

  For each dimension i: $m_i^2 = \beta m_i^1 + (1-\beta) g_i^2$ $\quad \theta_i^3 \leftarrow \theta_i^2 - \eta m_i^2$

⋮

- Compute gradient $\boldsymbol{g^t} = \nabla L(\boldsymbol{\theta^t})$

  For each dimension i: $m_i^t = \beta m_i^{t-1} + (1-\beta) g_i^t$ $\quad \theta_i^{t+1} \leftarrow \theta_i^t - \eta m_i^t$

範例程式

連結：
https://colab.research.google.com/drive/1XPIU-I77dXL9W74jnevmfb8K8XoEPRso?usp=sharing

Adam: RMSProp + Momentum

- Compute gradient $\boldsymbol{g}^t = \nabla L(\boldsymbol{\theta}^t)$

Momentum

For each dimension i: $m_i^t = \beta m_i^{t-1} + (1-\beta) g_i^t$

$\theta_i^{t+1} \leftarrow \theta_i^t - \eta m_i^t$

RMSProp

For each dimension i: $\sigma_i^t = \sqrt{\alpha\left(\sigma_i^{t-1}\right)^2 + (1-\alpha)\left(g_i^t\right)^2}$

$\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t$

Adam

$\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} m_i^t$

Adam has bias-corrected terms, which are omitted here for simplicity.