

# 李宏毅 (Hung-yi Lee) · HYLEE | Machine Learning (2021)

## HYLEE(2021) · 课程资料包 @ShowMeAI



视频

课件

笔记

代码

中英双语字幕

一键打包下载

官方笔记翻译

作业项目解析



视频 · B 站 [ 扫码或点击链接 ]

<https://www.bilibili.com/video/BV1fM4y137M4>



课件 & 代码 · 博客 [ 扫码或点击链接 ]

<http://blog.showmeai.tech/ntu-hylee-ml>

机器学习

Auto-encoder

生成式对抗网络

学习率

深度学习

卷积神经网络

GAN

自监督

自注意力机制

批次标准化

神经网络压缩

强化学习

元学习

Transformer

Awesome AI Courses Notes Cheatsheets 是 [ShowMeAI](#) 资料库的分支系列，覆盖最具知名度的 **TOP50+** 门 AI 课程，旨在为读者和学习者提供一整套高品质中文学习笔记和速查表。

点击课程名称，跳转至课程**资料包**页面，**一键下载**课程全部资料！

机器学习	深度学习	自然语言处理	计算机视觉
Stanford · CS229	Stanford · CS230	Stanford · CS224n	Stanford · CS231n
# Awesome AI Courses Notes Cheatsheets · 持续更新中			
知识图谱	图机器学习	深度强化学习	自动驾驶
Stanford · CS520	Stanford · CS224W	UCBerkeley · CS285	MIT · 6.S094



微信公众号

资料下载方式 2: 扫码点击**底部菜单栏**

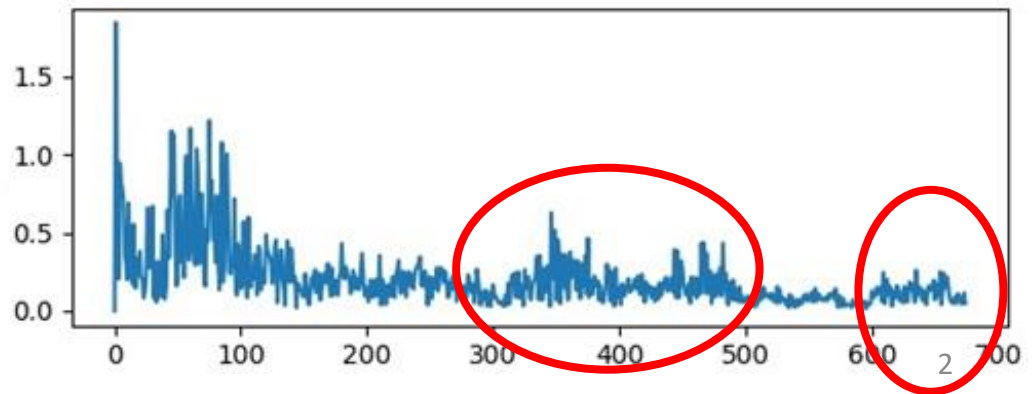
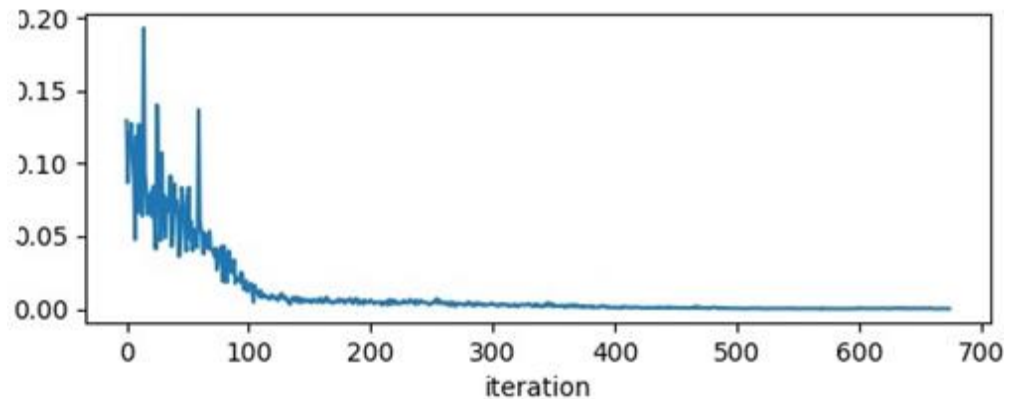
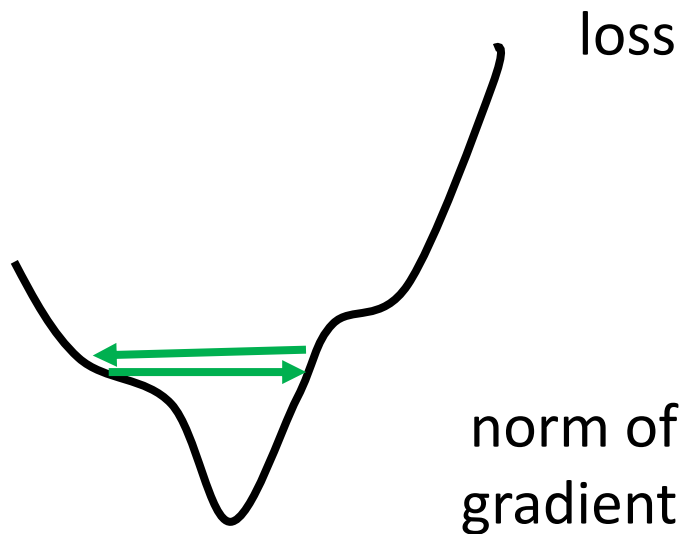
称为 **AI 内容创作者?** 回复 [ 添砖加瓦 ]

# Error surface is rugged ...

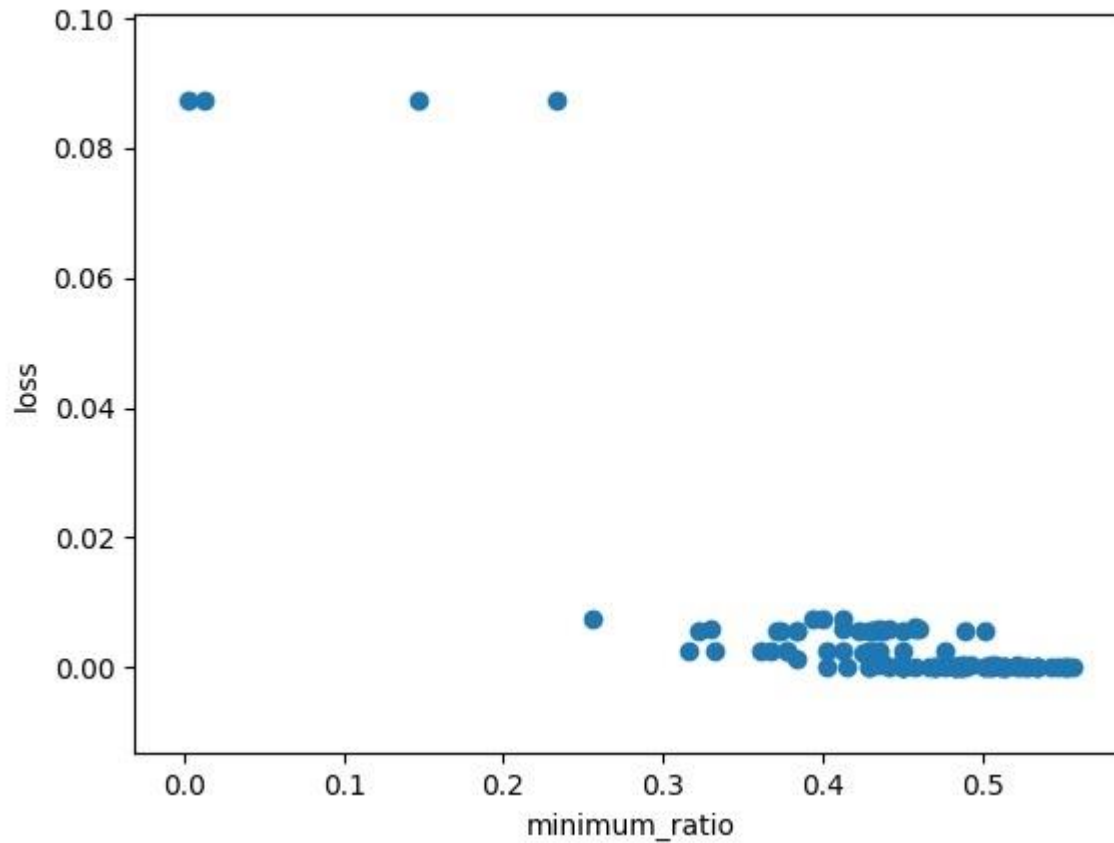
Tips for training: **Adaptive Learning Rate**

# Training stuck $\neq$ Small Gradient

- People believe training stuck because the parameters are around a critical point ...



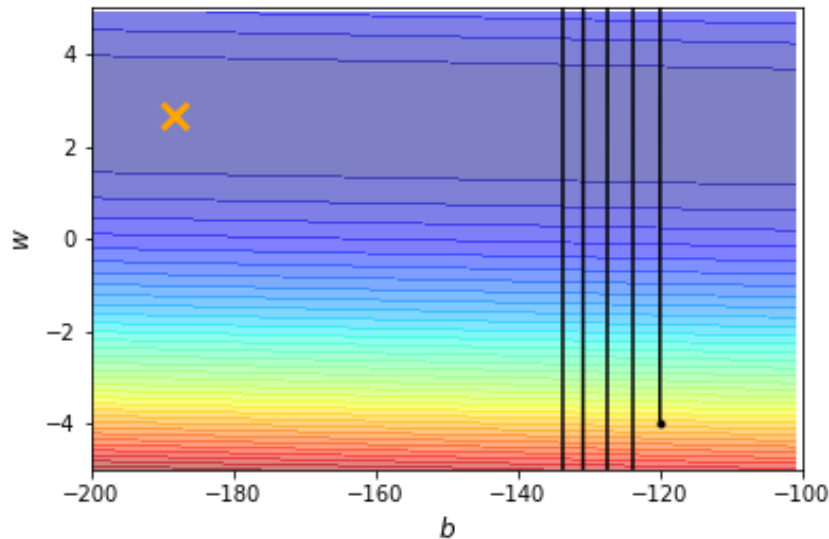
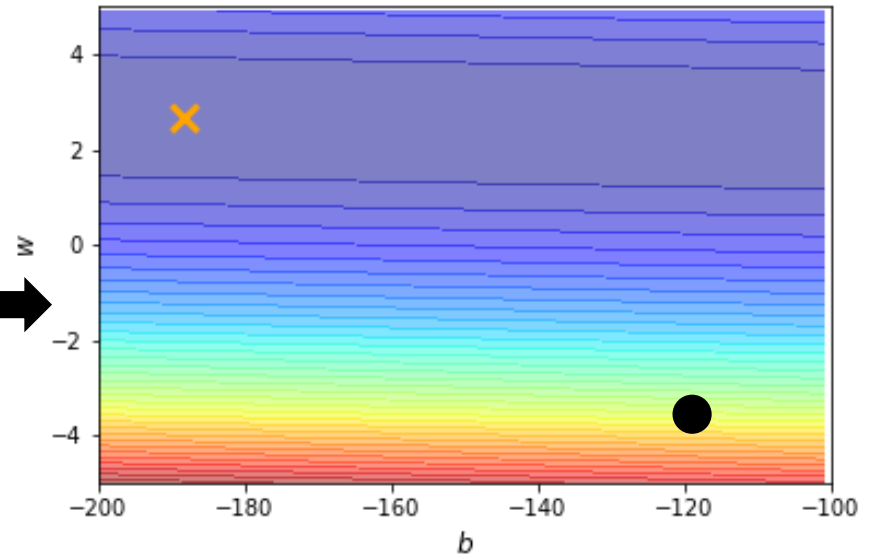
# Wait a minute ...



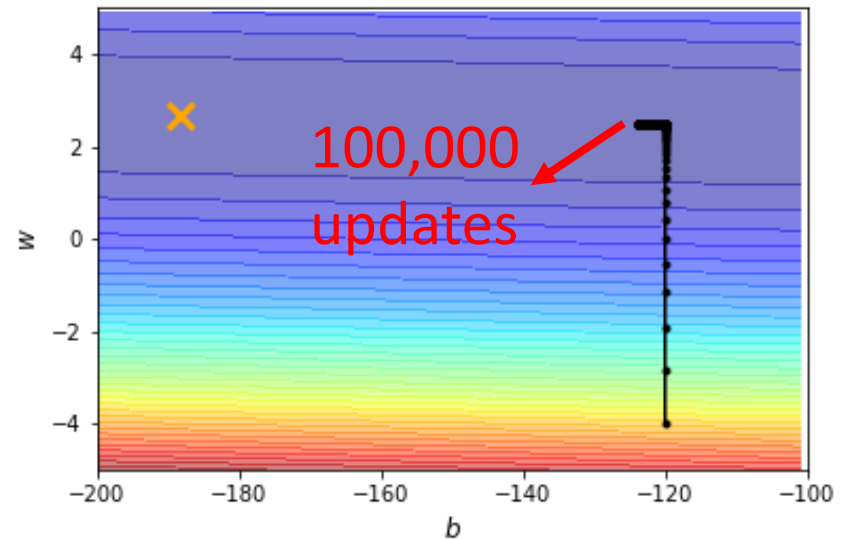
# Training can be difficult even without critical points.

This error surface is convex. ➡

Learning rate **cannot** be **one-size-fits-all**



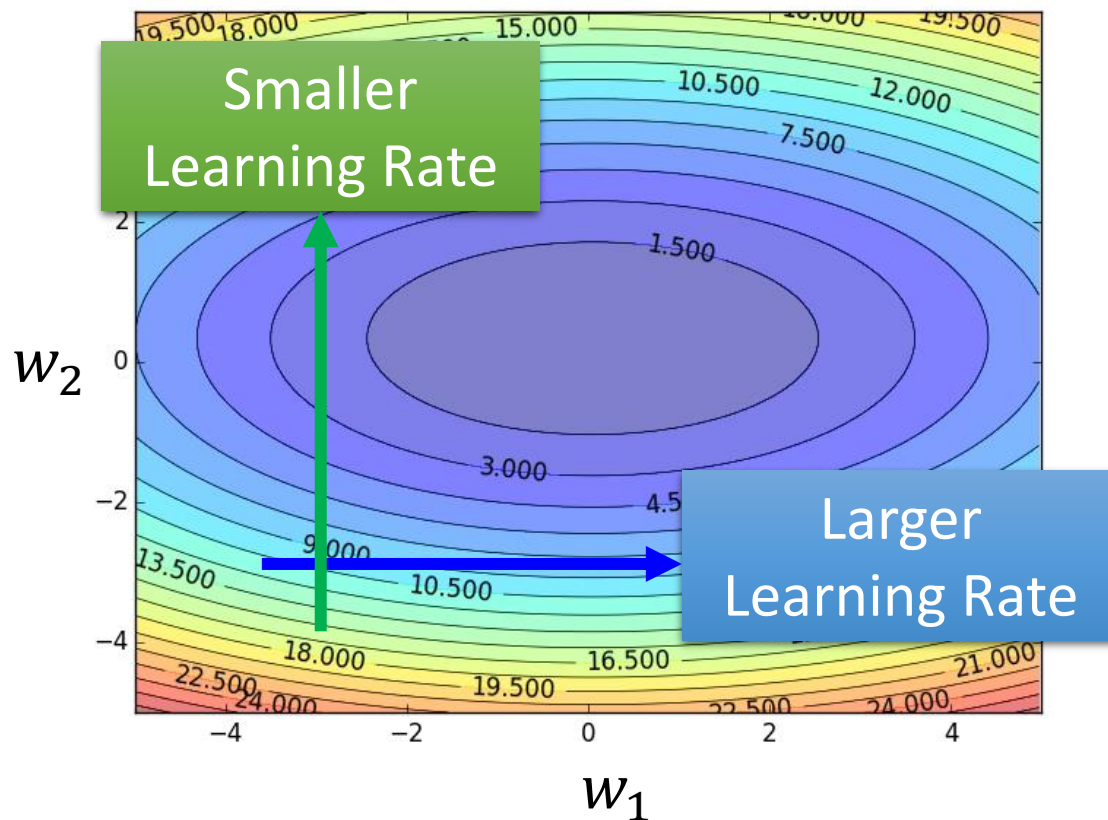
$$\eta = 10^{-2}$$



$$\eta = 10^{-7}$$

# Different parameters need different learning rate

Formulation for **one** parameter:



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

$$g_i^t = \frac{\partial L}{\partial \theta_i} \bigg|_{\theta = \theta^t}$$

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

Parameter  
dependent

# Root Mean Square

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

$$\theta_i^1 \leftarrow \theta_i^0 - \frac{\eta}{\sigma_i^0} g_i^0 \quad \sigma_i^0 = \sqrt{(g_i^0)^2} = |g_i^0|$$

$$\theta_i^2 \leftarrow \theta_i^1 - \frac{\eta}{\sigma_i^1} g_i^1 \quad \sigma_i^1 = \sqrt{\frac{1}{2} [(g_i^0)^2 + (g_i^1)^2]}$$

$$\theta_i^3 \leftarrow \theta_i^2 - \frac{\eta}{\sigma_i^2} g_i^2 \quad \sigma_i^2 = \sqrt{\frac{1}{3} [(g_i^0)^2 + (g_i^1)^2 + (g_i^2)^2]}$$

⋮

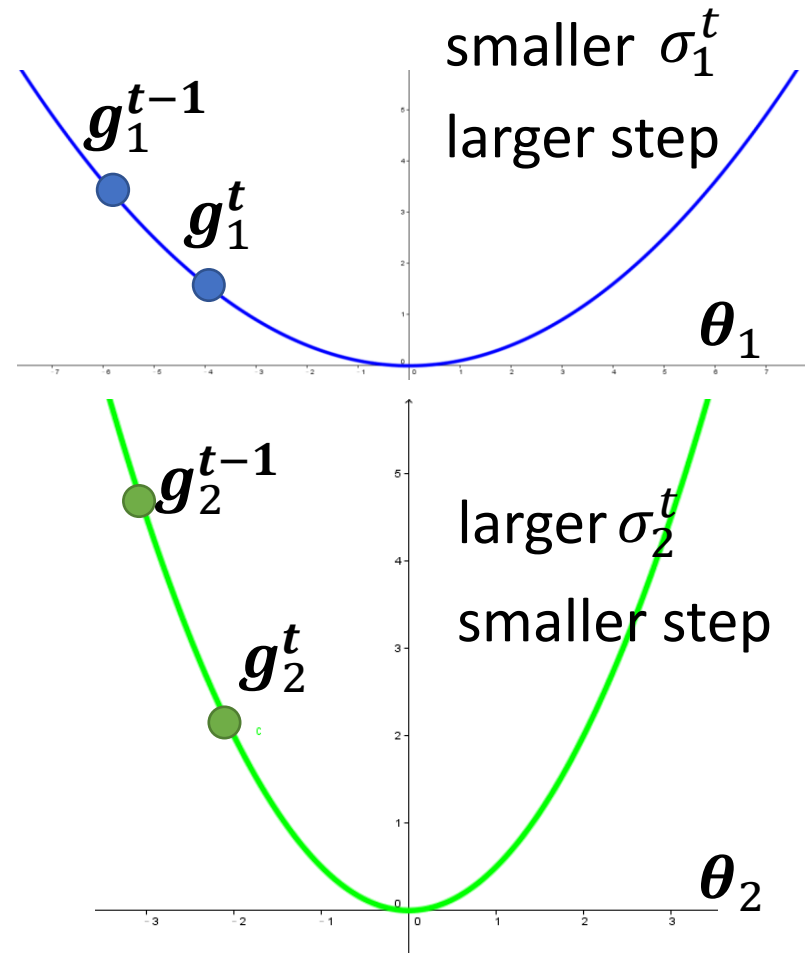
$$\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t \quad \sigma_i^t = \sqrt{\frac{1}{t+1} \sum_{i=0}^t (g_i^t)^2}$$

# Root Mean Square

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

$$\sigma_i^t = \sqrt{\frac{1}{t+1} \sum_{i=0}^t (g_i^t)^2}$$

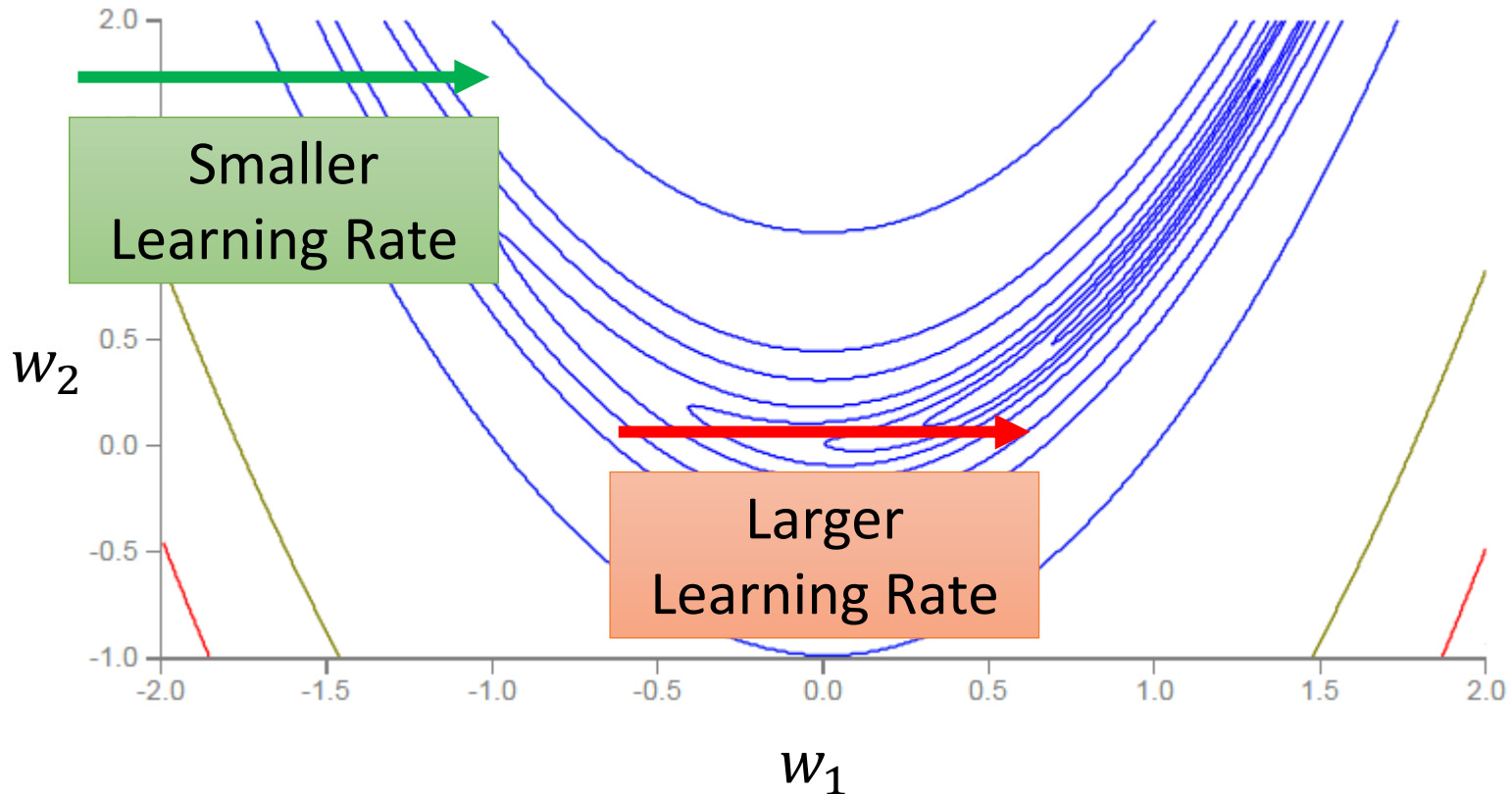
Used in **Adagrad**





# Learning rate adapts dynamically

Error Surface can be very complex.



# RMSProp

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

$$\theta_i^1 \leftarrow \theta_i^0 - \frac{\eta}{\sigma_i^0} g_i^0 \quad \sigma_i^0 = \sqrt{(g_i^0)^2} \quad 0 < \alpha < 1$$

$$\theta_i^2 \leftarrow \theta_i^1 - \frac{\eta}{\sigma_i^1} g_i^1 \quad \sigma_i^1 = \sqrt{\alpha(\sigma_i^0)^2 + (1 - \alpha)(g_i^1)^2}$$

$$\theta_i^3 \leftarrow \theta_i^2 - \frac{\eta}{\sigma_i^2} g_i^2 \quad \sigma_i^2 = \sqrt{\alpha(\sigma_i^1)^2 + (1 - \alpha)(g_i^2)^2}$$

⋮

$$\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t \quad \sigma_i^t = \sqrt{\alpha(\sigma_i^{t-1})^2 + (1 - \alpha)(g_i^t)^2}$$

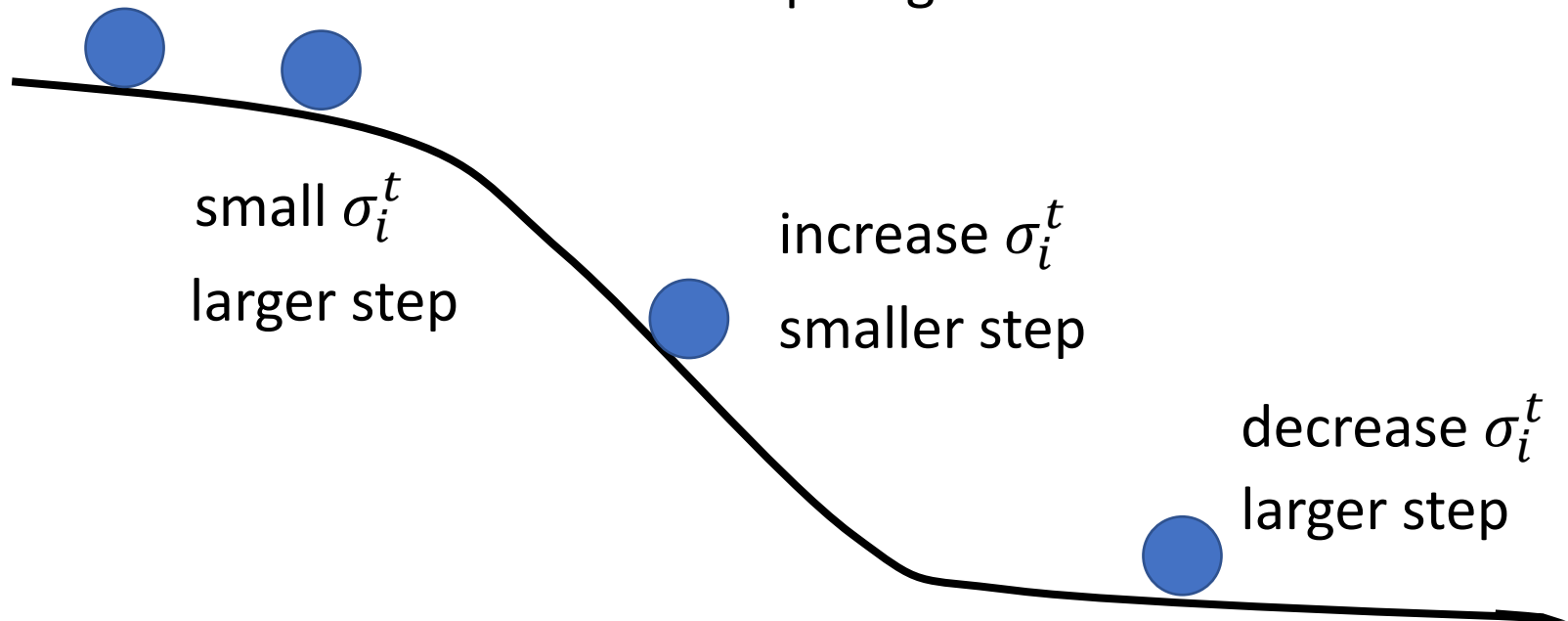
# RMSProp

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

$$\sigma_i^t = \sqrt{\alpha (\sigma_i^{t-1})^2 + (1 - \alpha) (g_i^t)^2}$$

$g_i^1 \quad g_i^2 \quad \dots \quad g_i^{t-1}$   
 $0 < \alpha < 1$

The recent gradient has larger influence, and the past gradients have less influence.



# Adam: RMSProp + Momentum

**Algorithm 1:** *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation.  $g_t^2$  indicates the elementwise square  $g_t \odot g_t$ . Good default settings for the tested machine learning problems are  $\alpha = 0.001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  and  $\epsilon = 10^{-8}$ . All operations on vectors are element-wise. With  $\beta_1^t$  and  $\beta_2^t$  we denote  $\beta_1$  and  $\beta_2$  to the power  $t$ .

**Require:**  $\alpha$ : Stepsize

**Require:**  $\beta_1, \beta_2 \in [0, 1)$ : Exponential decay rates for the moment estimates

**Require:**  $f(\theta)$ : Stochastic objective function with parameters  $\theta$

**Require:**  $\theta_0$ : Initial parameter vector

$m_0 \leftarrow 0$  (Initialize 1<sup>st</sup> moment vector)  $\rightarrow$  for momentum

$v_0 \leftarrow 0$  (Initialize 2<sup>nd</sup> moment vector)  $\rightarrow$  for RMSprop

$t \leftarrow 0$  (Initialize timestep)

**while**  $\theta_t$  not converged **do**

$t \leftarrow t + 1$

$g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$  (Get gradients w.r.t. stochastic objective at timestep  $t$ )

$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$  (Update biased first moment estimate)

$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$  (Update biased second raw moment estimate)

$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$  (Compute bias-corrected first moment estimate)

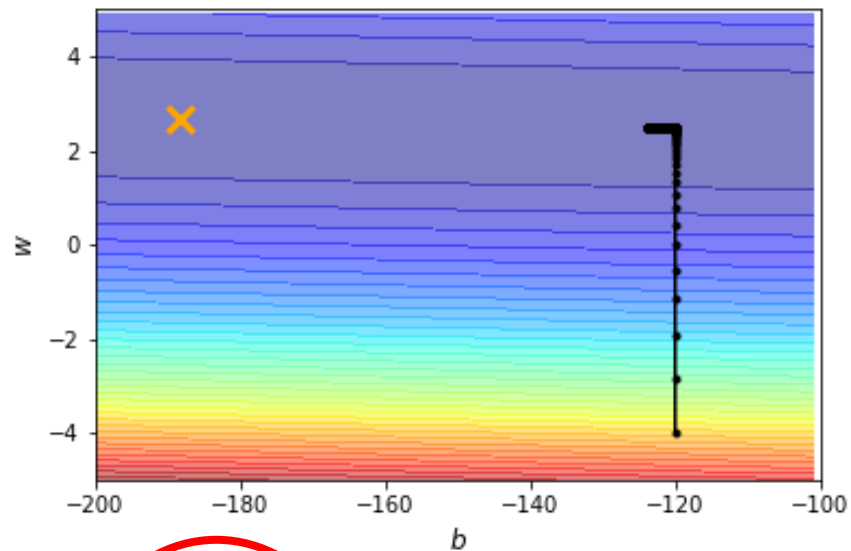
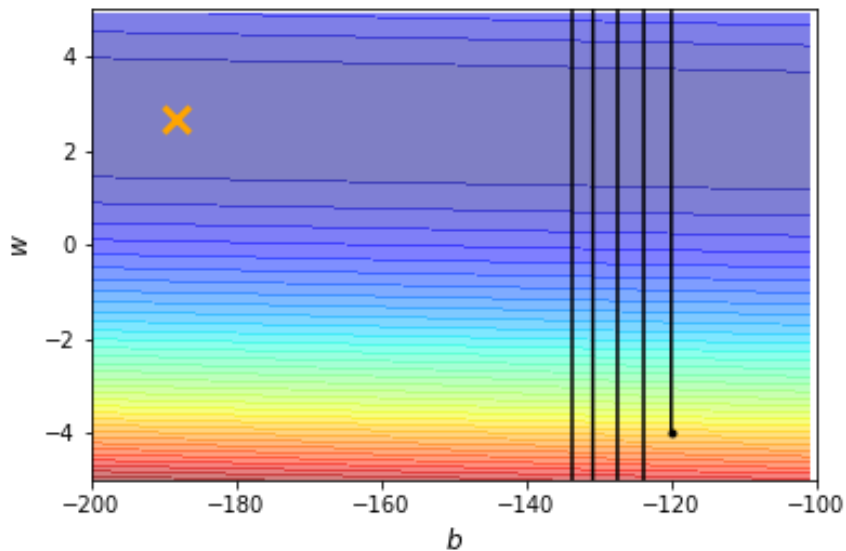
$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$  (Compute bias-corrected second raw moment estimate)

$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$  (Update parameters)

**end while**

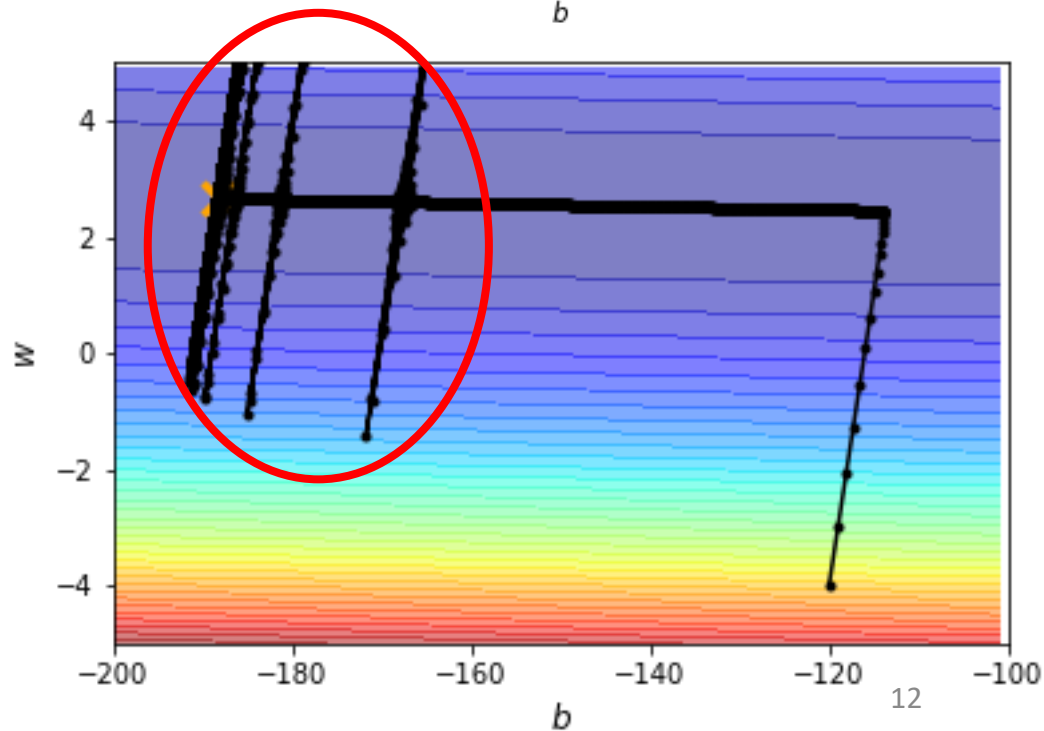
**return**  $\theta_t$  (Resulting parameters)

# Without Adaptive Learning Rate



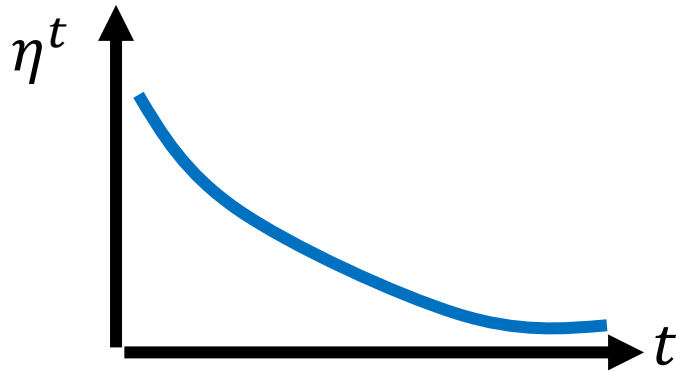
$$\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta}{\sigma_i^t} g_i^t$$

$$\sigma_i^t = \sqrt{\frac{1}{t+1} \sum_{i=0}^t (g_i^t)^2}$$



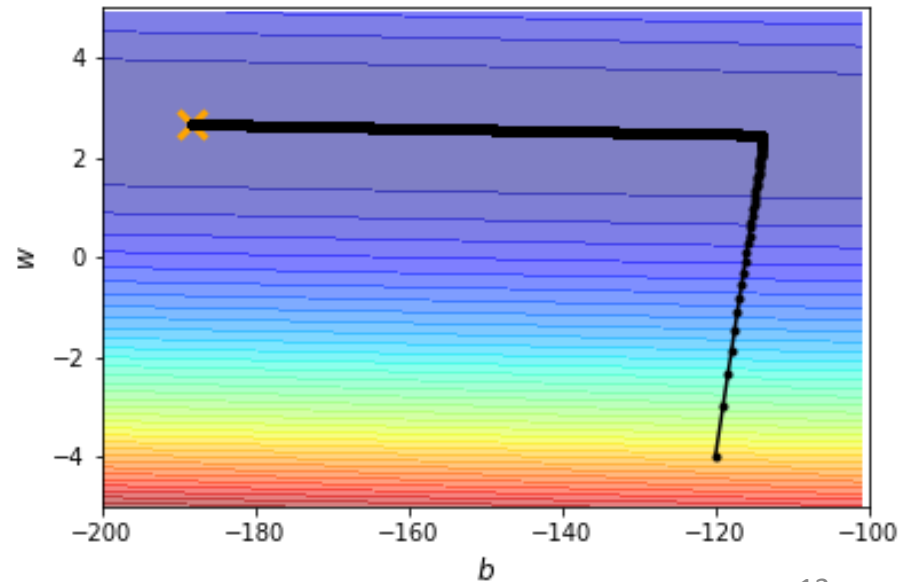
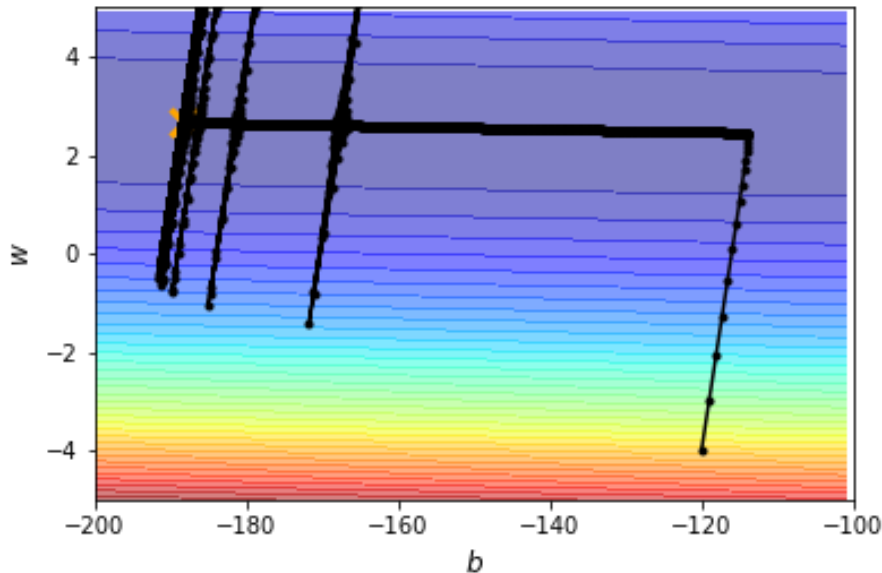
# Learning Rate Scheduling

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



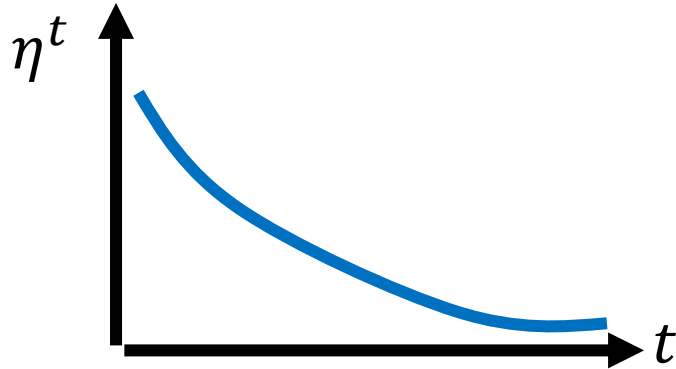
## Learning Rate Decay

As the training goes, we are closer to the destination, so we reduce the learning rate.



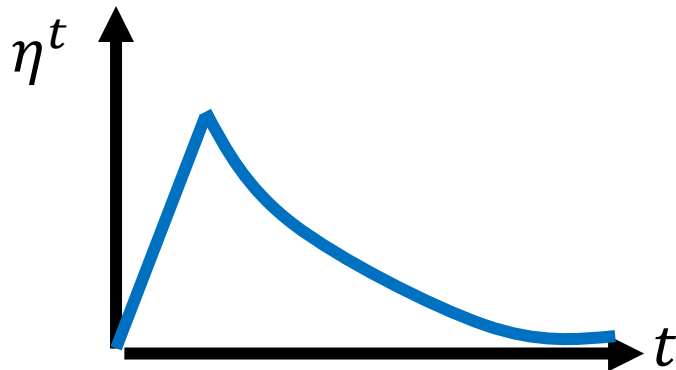
# Learning Rate Scheduling

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



## Learning Rate Decay

As the training goes, we are closer to the destination, so we reduce the learning rate.



## Warm Up

Increase and then decrease?

We further explore  $n = 18$  that leads to a 110-layer ResNet. In this case, we find that the initial learning rate of 0.1 is slightly too large to start converging<sup>5</sup>. So we use 0.01 to warm up the training until the training error is below 80% (about 400 iterations), and then go back to 0.1 and continue training. The rest of the learning schedule is as done previously. This 110-layer network converges well (Fig. 6, middle). It has *fewer* parameters than other deep and thin

---

<sup>5</sup>With an initial learning rate of 0.1, it starts converging (<90% error) after several epochs, but still reaches similar accuracy.

## Residual Network

<https://arxiv.org/abs/1512.03385>

### 5.3 Optimizer

We used the Adam optimizer [17] with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$  and  $\epsilon = 10^{-9}$ . We varied the learning rate over the course of training, according to the formula:

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(\text{step\_num}^{-0.5}, \text{step\_num} \cdot \text{warmup\_steps}^{-1.5}) \quad (3)$$

This corresponds to increasing the learning rate linearly for the first *warmup\_steps* training steps, and decreasing it thereafter proportionally to the inverse square root of the step number. We used *warmup\_steps* = 4000.

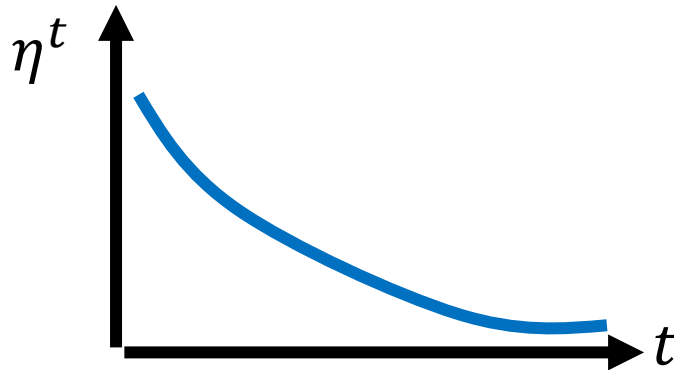
## Transformer

<https://arxiv.org/abs/1706.03762>



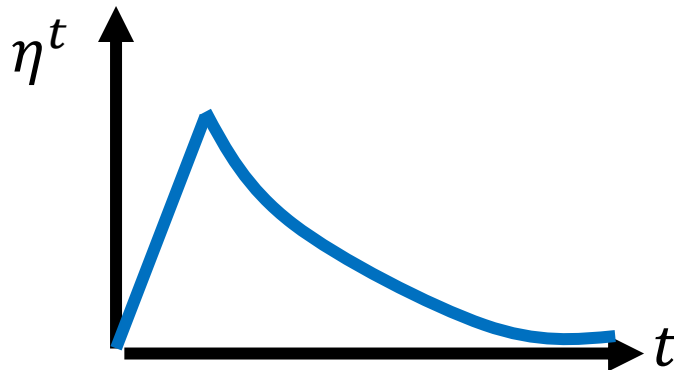
# Learning Rate Scheduling

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



## Learning Rate Decay

After the training goes, we are close to the destination, so we reduce the learning rate.



## Warm Up

Increase and then decrease?

At the beginning, the estimate of  $\sigma_i^t$  has large variance.

Please refer to **RAdam**

<https://arxiv.org/abs/1908.03265>

# Summary of Optimization

## (Vanilla) Gradient Descent

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

## Various Improvements

$$\theta_i^{t+1} \leftarrow \theta_i^t - \frac{\eta^t}{\sigma_i^t} \mathbf{m}_i^t$$

Learning rate scheduling

Momentum: weighted sum of the previous gradients

root mean square of the gradients

Consider direction

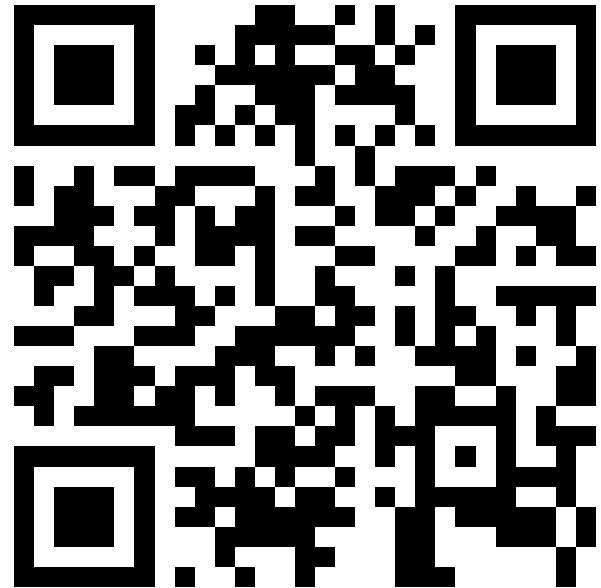
only magnitude

# To Learn More .....



<https://youtu.be/4pUmZ8hXlHM>

(in Mandarin)

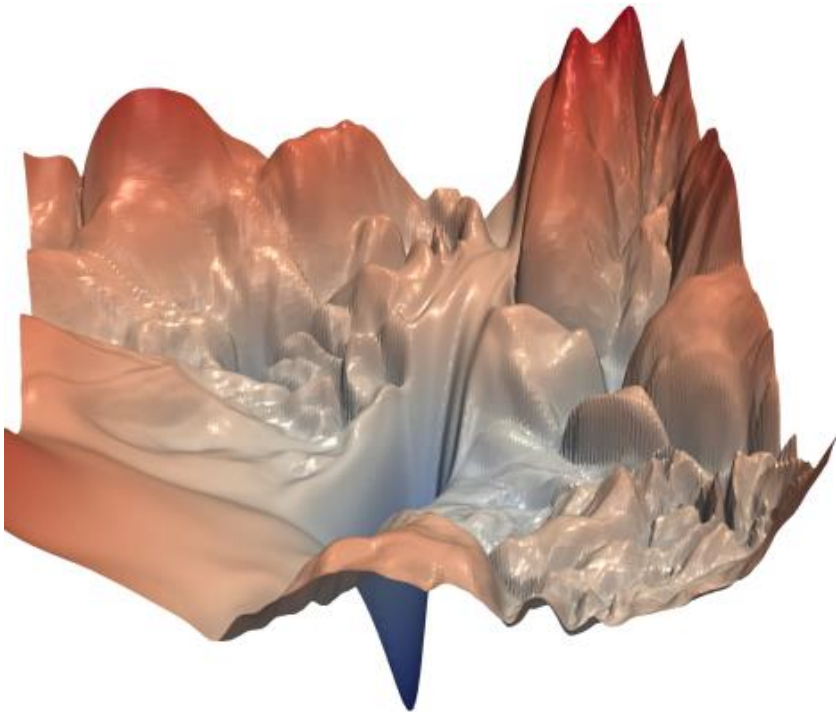


<https://youtu.be/e03YKGHXnL8>

(in Mandarin)

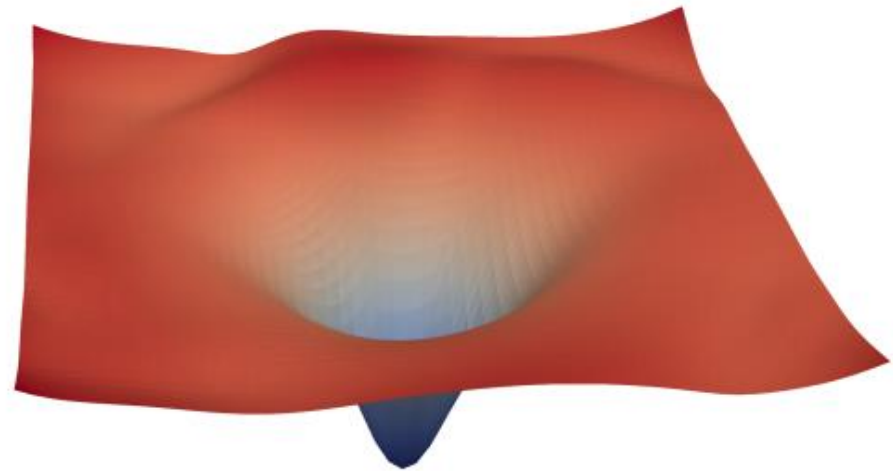
# Next Time

Source of image: <https://arxiv.org/abs/1712.09913>



Better optimization strategies:  
If the mountain won't move,  
build a road around it.

Next time



Can we change the error  
surface?  
Directly move the mountain!

# 李宏毅 (Hung-yi Lee) · HYLEE | Machine Learning (2021)

## HYLEE(2021) · 课程资料包 @ShowMeAI



视频

课件

笔记

代码

中英双语字幕

一键打包下载

官方笔记翻译

作业项目解析



视频 · B 站 [ 扫码或点击链接 ]

<https://www.bilibili.com/video/BV1fM4y137M4>



课件 & 代码 · 博客 [ 扫码或点击链接 ]

<http://blog.showmeai.tech/ntu-hylee-ml>

机器学习

Auto-encoder

生成式对抗网络

学习率

深度学习

卷积神经网络

GAN

自监督

自注意力机制

批次标准化

神经网络压缩

强化学习

元学习

Transformer

Awesome AI Courses Notes Cheatsheets 是 [ShowMeAI](#) 资料库的分支系列，覆盖最具知名度的 **TOP50+** 门 AI 课程，旨在为读者和学习者提供一整套高品质中文学习笔记和速查表。

点击课程名称，跳转至课程**资料包**页面，**一键下载**课程全部资料！

机器学习	深度学习	自然语言处理	计算机视觉
Stanford · CS229	Stanford · CS230	Stanford · CS224n	Stanford · CS231n
# Awesome AI Courses Notes Cheatsheets · 持续更新中			
知识图谱	图机器学习	深度强化学习	自动驾驶
Stanford · CS520	Stanford · CS224W	UCBerkeley · CS285	MIT · 6.S094



微信公众号

资料下载方式 2: 扫码点击**底部菜单栏**

称为 **AI 内容创作者?** 回复 [ 添砖加瓦 ]