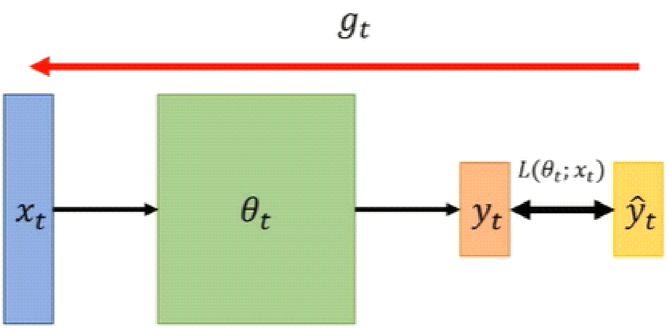
Optim

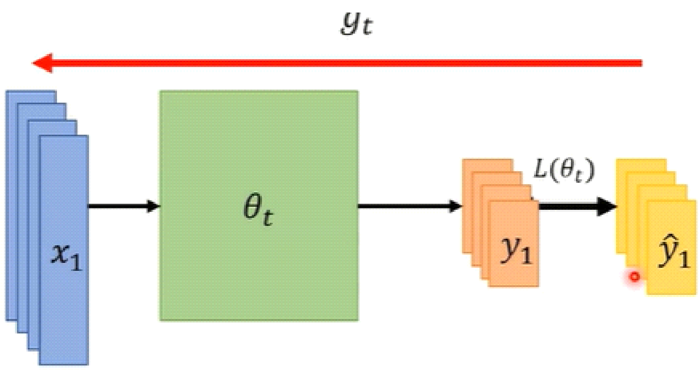
2021年4月1日 14:32

- Find a θ to get the lowest $\sum_x L(\theta; x)$!!

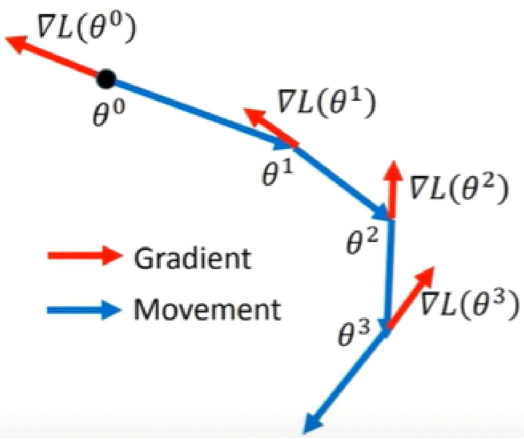
On-line vs Off-line

- On-line：one pair of $(x_t, \hat{y}_t)$ at a time step





SGD



Start at position θ^0

Compute gradient at θ^0

Move to $\theta^1 = \theta^0 - \eta \nabla L(\theta^0)$

Compute gradient at θ^1

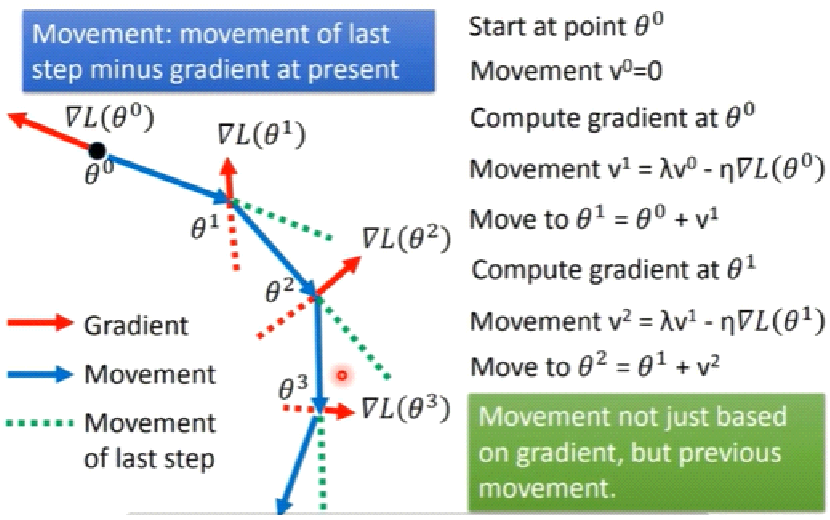
Move to $\theta^2 = \theta^1 - \eta \nabla L(\theta^1)$

⋮

Stop until $\nabla L(\theta^t) \approx 0$

SGD with Momentum(SGDM)

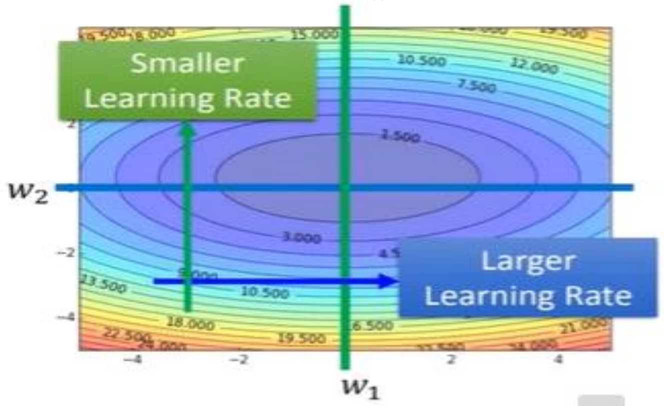
SGD with Momentum(SGDM)



加上了在时间上的Gradient Descent
在做这个东西的时候某个time step gradient 接近于0 就会卡在这里， 有点像local min

Adagrad

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\sum_{i=0}^{t-1}(g_i)^2}} g_{t-1}$$



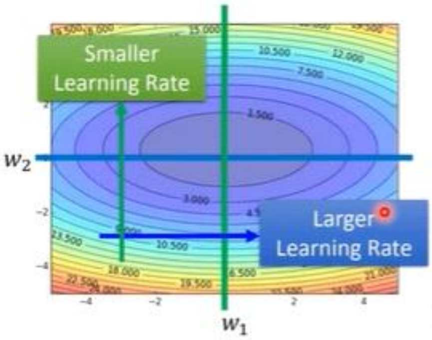
SGD的加上分母，SGD在前几个step太大， 会暴走，如果过去的gradient很大说明比较崎岖， 就走小一点，避免一下走太大，
以前的gradient比较小就说明比较平缓， 走到慢一点

RMSProp

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{v_t}} g_{t-1}$$

$$v_1 = {g_0}^2$$

$$v_t = \alpha v_{t-1} + (1-\alpha)(g_{t-1})^2$$



Credit to 李宏毅老師上課投影片

与Adagrad的唯一区别就是分母的算法不太一样，把过去的累计的grad乘上a现在的乘1-a加起来。 adagrad会无休止的累加， 一开始比较大的话后面可能会主动卡住。

可以解决永无止境的问题， 但是没有办法解决可能会卡在grad = 0的问题。

Adam

- SGDM

$$\theta_t = \theta_{t-1} - \eta m_t$$

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

- RMSProp

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{v_t}} g_{t-1}$$

$$v_1 = {g_0}^2$$

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

$$\boxed{\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} \hat{m}_t}$$

$$\hat{m}_t = \frac{m_t}{1-{\beta_1}^t}$$

$$\hat{v}_t = \frac{v_t}{1-{\beta_2}^t}$$

de-biasing

$$\beta_1 = 0.9$$

$$\beta_2 = 0.999$$

$$\varepsilon = 10^{-8}$$

五大基本算法

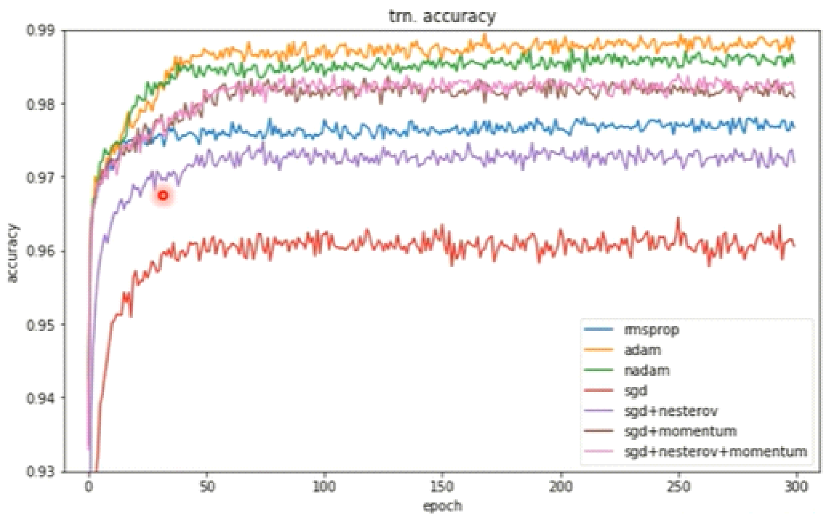
What you have known before?

- SGD [Cauchy, 1847]
- SGD with momentum [Rumelhart, et al., Nature'86]

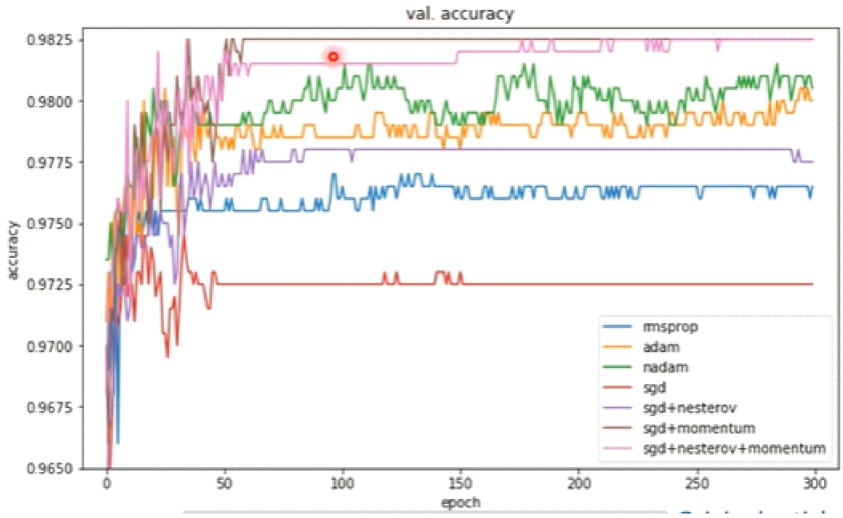
Adaptive learning rate

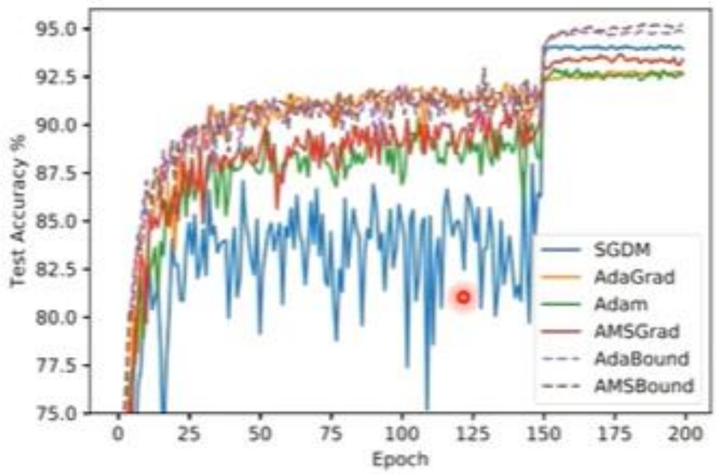
- Adagrad [Duchi, et al., JMLR'11]
- RMSProp [Hinton, et al., Lecture slides, 2013]
- Adam [Kingma, et al., ICLR'15]

为什么这些著名的optimizer都是2014年提出来的呢?
Bert Adan都是用Adam训练出来的
Mask-R-CNN，YoLo，ReNet是用SGDM训练出来的
Big - GAN 是Adam训练出来的
MEMO Adam

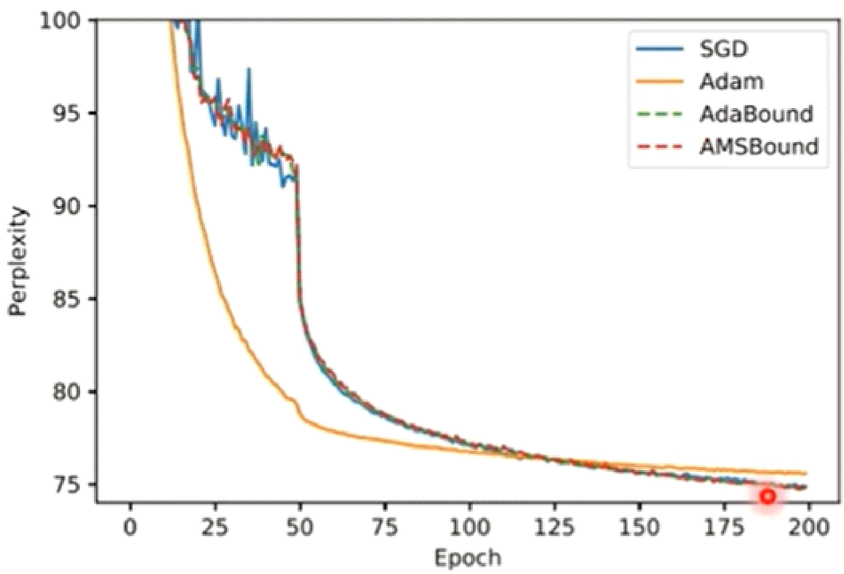


Adam和SGDM抢到了两个最极端的位置





(d) Test Accuracy for ResNet-34

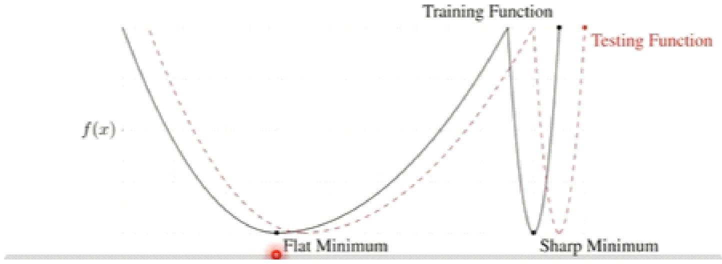


Adam训练比较快，但是泛化差些

SGDM训练平稳

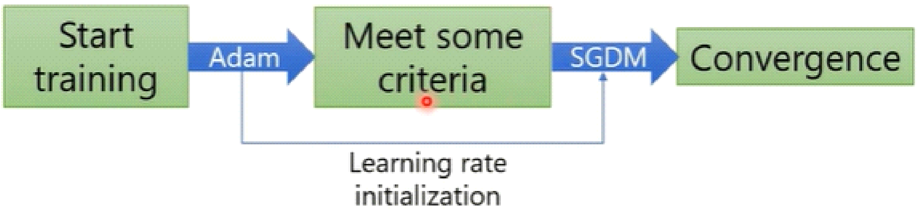
- Adam：fast training, large generalization gap, unstable
- SGDM：stable, little generalization gap, better convergence(?)

An intuitive illustration for generalization gap



Adam快SGDM比较稳

SWATS：一开始用Adam后面用SGDM



可不可以修正一下Adam?

- Trouble shooting

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}}\hat{m}_t$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_{t-1}, \beta_1 = 0.$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(g_{t-1})^2, \beta_2 = 0.999$$

The "memory" of v_t keeps roughly 1000 steps!!

In the final stage of training, most gradients are small and non-informative, while some mini-batches provide large informative gradient rarely.

- Trouble shooting

Maximum movement distance for one single update is roughly upper bounded by $\sqrt{\frac{1}{1-\beta_2}}\eta$

Non-informative gradients contribute more than informative gradients

time step	...	100000	100001	100002	100003	...	100999	101000
gradient		1	1	1	1		100000	1
movement		η	η	η	η		$10\sqrt{10}\eta$	$10^{-3.5}\eta$

提供更多Information的step作用效果比较小
前面的update乱走，造成不好的影响

- AMSGrad [Reddi, et al., ICLR'18]

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} m_t$$

$$\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$$

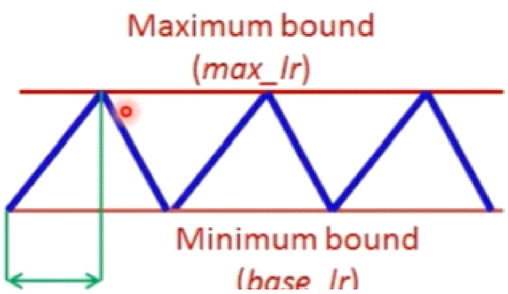
Reduce the influence of non-informative gradients

Remove de-biasing due to the max operation

移除小的影响，前面的大的gradient不能被忘记

Another way：
Change lr for SGDM

- Cyclical LR [Smith, WACV'17]
- learning rate：decide by LR range test
- stepsize：several epochs
- avoid local minimum by varying learning rate



探索-收敛

- SGDR [Loshchilov, et al., ICLR'17]