Chapter 4 New Optimization for Deep Learning

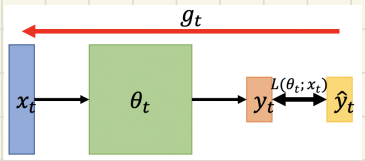
概念定义：

θ_t：t时刻的模型参数

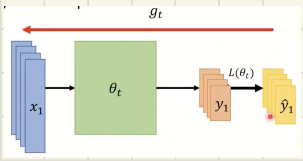
$\nabla L(\theta_t)$ or g_t：在参数θ_t下的梯度，用来计算θ_{t+1}

m_{t+1}：从起始时刻到到t时刻，关于梯度的累积值momentum，用来计算θ_{t+1}

网络模型：

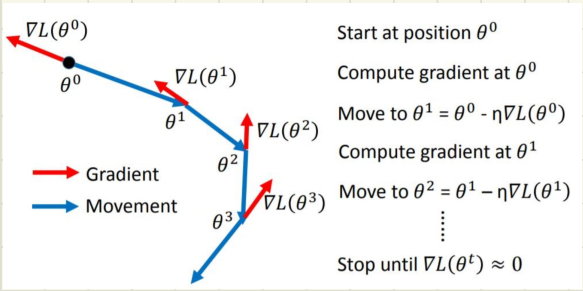


on-line model

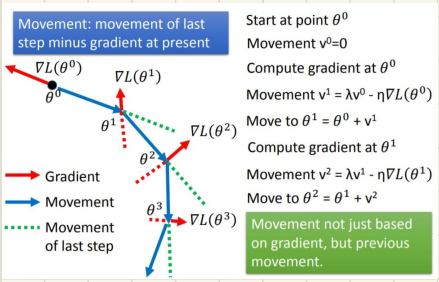


off-line model（本章重点关注）

1. Stochastic Gradient Descent（SGD，1847）



2. Stochastic Gradient Descent with Momentum（SGDM，1986）



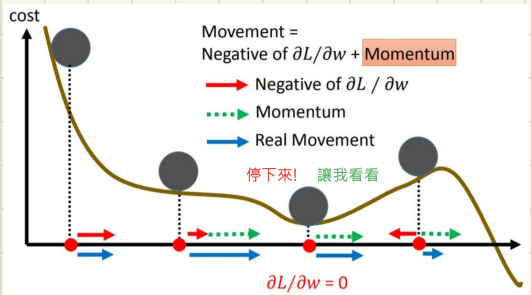
v^i is actually the weighted sum of all the previous gradient: $\nabla L(\theta^0), \nabla L(\theta^1), \ldots \nabla L(\theta^{i-1})$

$v^0 = 0$

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

$v^2 = -\lambda\eta\nabla L(\theta^0) - \eta\nabla L(\theta^1)$

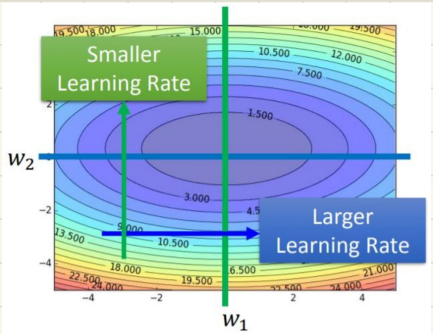
由于有在momentum，即使梯度为0算法也不会立即停止，避免陷入minima



3. Adagrad (2011)

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

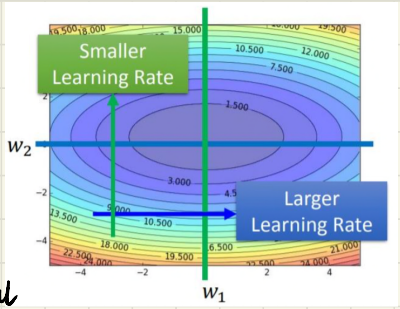
- 对于不同的参数采用不同的learning rate，以避免在一些梯度变化较大的维度步长过大



4. RMSProp (2013)

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{v_t}} g_{t-1}$$
$$v_1 = {g_0}^2$$
$$v_t = \alpha v_{t-1} + (1-\alpha)(g_{t-1})^2$$

- 改变了Adagrad中学习率的变化方法，Adagrad中由于分母中不断的累加梯度，如果梯度过大，学习率变得很小，则学习近乎停止
- RMSProp借鉴了momentum的概念，保证了分母不会无止境的变大（Exponential moving average (EMA) of squared gradients is not monotonically increasing.）



5. Adam (2014)

- SGDM

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

- RMSProp

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{v_t}} g_{t-1}$$
$$v_1 = {g_0}^2$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(g_{t-1})^2$$

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

- 学习率的变化部分借鉴了RMSProp
- 梯度部分借鉴了SGDM

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

de-biasing

$$\beta_1 = 0.9$$
$$\beta_2 = 0.999$$
$$\varepsilon = 10^{-8}$$

除以1-β的原因是，m_t经过一段时间会稳定下来，但最初m_t比较小，所以除以一个小于1的值适当放大以靠近后期的稳定值

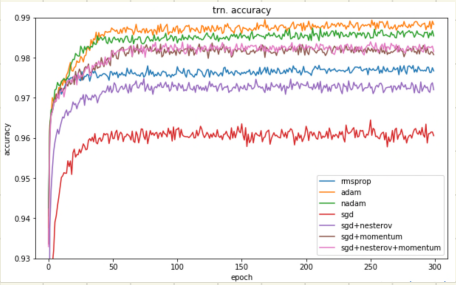
6. Optimizer in Real Application

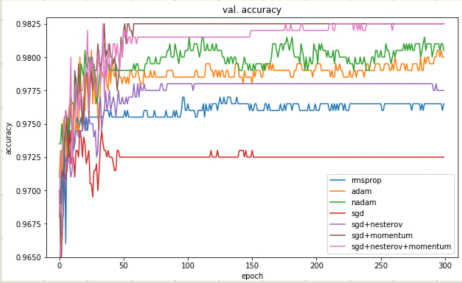
Adam：Bert，Transformer，Tacotron，Big-GAN，MAML

SGDM：YoLo，ResNet，Mask R-CNN

① Adam vs. SGD

领域内比较有名的Neural Network都出自于SGDM和Adam，原因可能是两者在optimizer中各占了一个极端

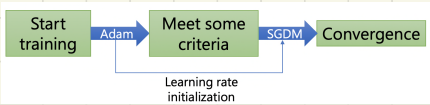




Adam: fast training, large generalization gap, unstable

SGDM: stable, little generalization gap, better convergence

因此，SWATS (2017)年提出了 Begin with Adam, end with SGDM



② 改进Adam

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

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

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

The "memory" of v_t keeps roughly 1000 steps!

每一次乘以0.999，则v_t表示的是一个比较数的相加，v_t受到梯度的影响是 $\frac{1}{1-0.999} = 1000$步

Adam的作者发现在训练的后期，大多数梯度都很小，是non-informative的，只有少数mini-batches会提供比较大的informative gradient以指明继续下降的方向

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

1000η

假设已经训练了10万个timestamp且之前的梯度都不大，在这之后碰到了一个很大的梯度

对于第100999步：$\beta_1 = 0 \rightarrow \hat{m}_t = g_{t-1} = 10^5$　假设之前的gradient均为1，则此时 $v_{t-1} = 1$，$v_t = 0.999 \times 1 + 0.001 \times 10^{10} \approx 10^7$

$$\frac{\eta}{\sqrt{v_t}+\varepsilon} \approx \eta \cdot \frac{10^5}{\sqrt{10^7}} = 10\sqrt{10}\eta$$

第100000～100998步并没有提供有意义的梯度，但前进了1000η；第100999步提供了有意义的梯度，却只前进了$10\sqrt{10}\eta$

⇕

受限于Adam中一次更新的最大移动距离为 $\sqrt{1-\beta_1}\eta$

- AMSGrad (ICLR'18)

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} m_t$$
$$\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$$

- 通过 max 使 $\hat{v}_t$ 记住之前遇到过的较大的梯度，防止在长时间的小梯度进行无效的学习；但只解决了 large learning rate 的情况
- 又有可能像 AdaGard 一样导致分母过大

- AdaBound (ICLR'19)

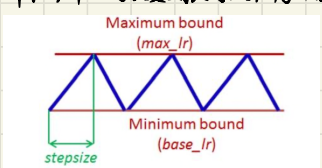
$$\theta_t = \theta_{t-1} - Clip\left(\frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}}\right)\hat{m}_t$$
$$Clip(x) = Clip\left(x, 0.1 - \frac{0.1}{(1-\beta_2)t+1}, 0.1 + \frac{0.1}{(1-\beta_2)t}\right)$$

- 通过限制上下界进行截断的方法防止学习率过大或过小（失去了 Adaptive 的意义）

③改进 SGDM

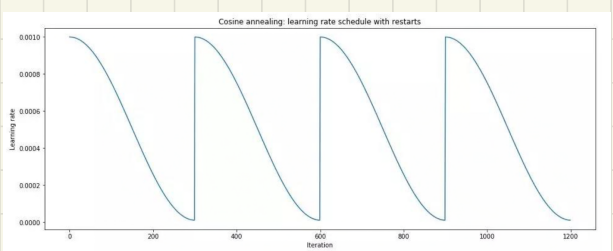
虽然 Adaptive learning rate algorithm 的思想是动态调整学习率，但 SGDM 无法做到动态调整，退而求其次寻找一个最合适的 "fix learning rate"

- LR Range test [Smith, WACV'17]：学习率过大或过小，performance 都不是最好的；学习率在中段效果最好
- Cyclical LR [Smith, WACV'17]：学习率在上界和下界之间反复变换可以取得不错的 performance

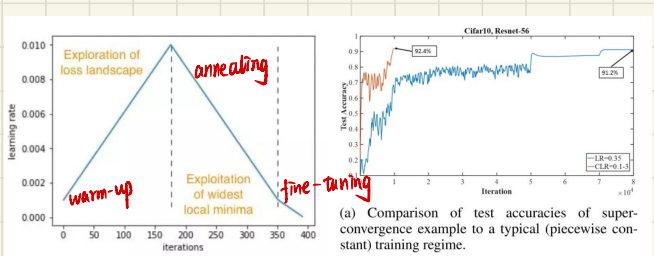


- 避免 local minima 通过增大学习率跳出局部极小
- 找到 global minima 后即便增大学习率也不会跳出极小点

- SGDR [Loshchilov, ICLR'17]



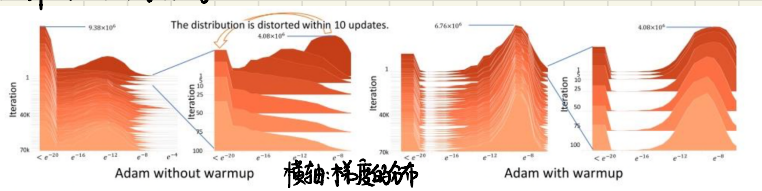
- One-cycle LR



(a) Comparison of test accuracies of super-convergence example to a typical (piecewise constant) training regime.

7. Warm-up in Adam

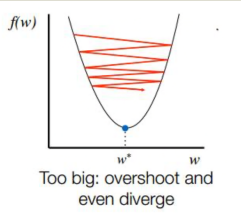
① Warm-up对Adam的影响



横轴：样本梯度的分布

Figure 2: The absolute gradient histogram of the Transformers on the De-En IWSLT' 14 dataset during the training (stacked along the y-axis). X-axis is absolute value in the log scale and the height is the frequency. Without warmup, the gradient distribution is distorted in the first 10 steps.

- 在无Warm-up的条件下，初始值是随机初值化的，再加上分母部分求的估计不是很准，导致前10次迭代中梯度的分布与后期不同
- 称为 distorted distribution



distorted gradient
↓
distorted EMA squared gradient
↓
bad learning rate

- 初期混乱的梯度导致EMA对方差估计不准，进而影响学习率，使得步长过大来回震荡
- 在前期保持一个小的步长，保证即便梯度估计不准确也不要大步前进

② RAdam

$$\rho_t = \rho_\infty - \frac{2t\beta_2{}^t}{1-\beta_2{}^t} \longrightarrow \text{effective memory size of EMA}$$

$$\rho_\infty = \frac{2}{1-\beta_2} - 1 \longrightarrow \text{max memory size}$$

$$r_t = \sqrt{\frac{(\rho_t - 4)(\rho_t - 2)\rho_\infty}{(\rho_\infty - 4)(\rho_\infty - 2)\rho_t}} \longrightarrow \text{approximated } \frac{\mathrm{Var}[\frac{1}{\hat{m}_t}]}{\mathrm{Var}[\frac{1}{\hat{v}_t}]} \text{ (for } \rho_t > 4\text{)}$$

When $\rho_t \le 4$ (first few steps of training)

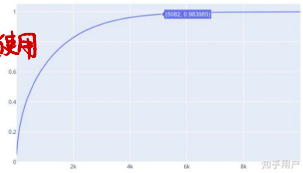
$$\theta_t = \theta_{t-1} - \eta \hat{m}_t$$

(当$\rho_t<4$无法用Adam，故改使用SGDM)

When $\rho_t > 4$

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

r_t is increasing through time!



r_t 随时间的变化规律如图

RAdam VS. SWATS：

	RAdam	SWATS
Inspiration	Distortion of gradient at the beginning of training results in inaccurate adaptive learning rate	non-convergence and generalization gap of Adam, slow training of SGDM
How?	Apply warm-up learning rate to reduce the influence of inaccurate adaptive learning rate	Combine their advantages by applying Adam first, then SGDM
Switch	SGDM to RAdam	Adam to SGDM
Why switch	The approximation of the variance of $\hat{v}_t$ is invalid at the beginning of training	To pursue better convergence
Switch point	When the approximation becomes valid	Some human-defined criteria

③ Lookahead [Zhang, et al., arXiv'19]

核心思想：k step forward, 1 step back "universal wrapper for all optimizer"

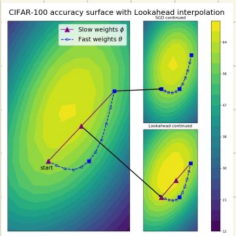
For $t = 1, 2, \ldots$ (outer loop)　　θ：fast weight（用于探索）　ϕ：slow weight

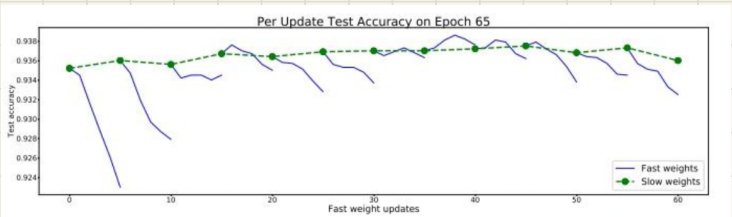
$\theta_{t,0} = \phi_{t-1}$

For $i = 1, 2, \ldots k$ (inner loop)　　使用optimizer前进k步

$\theta_{t,i} = \theta_{t,i-1} + Optim(Loss, data, \theta_{t,i-1})$

$\phi_t = \phi_{t-1} + \alpha(\theta_{t,k} - \phi_{t-1})$ ——→ 向k步之前的状态回退一部分





左图表示一次训练后期的两组权重的变化情况。fast weight不稳定，有可能向坏的方向前进，但在slow weight的控制下，可以保持比较稳定的结果

one step back的作用是避免过于危险的exploration，使算法更stable

Lookahead算法有效的原因是可以避免进入过于崎岖的地方，尽量在较为平坦的地方寻找最优解，使得模型具有更好的generalization

④ Momentum recap（补充知识，与Adam关系大）：用未来可能到达的位置梯度参与当前的计算

8. Future Position in Current Step

① Nesterov accelerated gradient (NAG　Nesterov, et al., SSSR'1983)

$$SGDM\begin{cases}\theta_t = \theta_{t-1} - m_t \\ m_t = \lambda m_{t-1} + \eta \nabla L(\theta_{t-1})\end{cases} \longrightarrow \begin{cases}\theta_t = \theta_{t-1} - m_t \\ m_t = \lambda m_{t-1} + \eta \nabla L(\theta_{t-1} - \lambda m_{t-1})\end{cases}$$

使用该项使下一次可能到达的位置参与到当前的计算中，在保留θ_{t-1}的时刻，拷贝一份用于计算该项，所以需要两份 θ_{t-1}

令 $\theta_t' = \theta_t - \lambda m_t = \theta_{t-1} - m_t - \lambda m_t$

$= \theta_{t-1} - \lambda m_t - \lambda m_{t-1} - \eta \nabla L(\theta_{t-1} - \lambda m_{t-1})$

$= \theta_{t-1}' - \lambda m_t - \eta \nabla L(\theta_{t-1}')$

$m_t = \lambda m_{t-1} + \eta \nabla L(\theta_{t-1}')$

两者相比，NAG在本次迭代中就用到了BP将要到达位置的momentum（将m_t超前部署一个timestamp）

将SGDM的一式代入一式可得：$\theta_t = \theta_{t-1} - \lambda m_{t-1} - \eta \nabla L(\theta_{t-1})$，$m_t = \lambda m_{t-1} + \eta \nabla L(\theta_{t-1})$

② Nadam [Dozat, ICLR workshop'16]

将NAG超前部署的概念加入到Adam中

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

$$\hat{m}_t = \frac{\beta_1 m_t}{1-\beta_1^{t+1}} + \frac{(1-\beta_1)g_{t-1}}{1-\beta_1^{t}}$$

SGDM

$$\hat{m}_t = \frac{1}{1-\beta_1^{t}}(\beta_1 m_{t-1} + (1-\beta_1)g_{t-1})$$

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

③ AdamW & SGDW with momentum [Loshchilov, arXiv'17]

在2014年至2017年，大家对于正则化和Adam的用法存在误区：

$$L_{l_2}(\theta) = L(\theta) + \gamma||\theta||^2$$

SGD

$$\theta_t = \theta_{t-1} - \nabla L_{l_2}(\theta_{t-1})$$
$$= \theta_{t-1} - \nabla L(\theta_{t-1}) - \gamma\theta_{t-1}$$

SGDM

$$\theta_t = \theta_{t-1} - \lambda m_{t-1} - \eta(\nabla L(\theta_{t-1}) + \gamma\theta_{t-1})$$
$$m_t = \lambda m_{t-1} + \eta(\nabla L(\theta_{t-1}) + \gamma\theta_{t-1})\ ?$$
$$m_t = \lambda m_{t-1} + \eta(\nabla L(\theta_{t-1}))\ ?$$

Adam

$$m_t = \lambda m_{t-1} + \eta(\nabla L(\theta_{t-1}) + \gamma\theta_{t-1})\ ?$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla L(\theta_{t-1}) + \gamma\theta_{t-1})^2\ ?$$

在计算momentum时是否应该加入正则项一同计算，14至17年大家普遍采用的是加入的方案。

在2017年，Loshchilov提出正则项不加入momentum中的计算效果更好，并取得了一定范围的应用（NLP领域）

SGDWM

$$\theta_t = \theta_{t-1} - m_t - \gamma\theta_{t-1}$$
$$m_t = \lambda m_{t-1} + \eta(\nabla L(\theta_{t-1}))$$

AdamW

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla L(\theta_{t-1})$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla L(\theta_{t-1}))^2$$
$$\theta_t = \theta_{t-1} - \eta(\frac{1}{\sqrt{\hat{v}_t}+\varepsilon}\hat{m}_t + \gamma\theta_{t-1})$$

* 本章节最常用的一个Optimizer

9、Something Helps Optimization

① More Exploration:

Shuffling：每一代训练时都要打乱数据集的顺序，每次要重新切分mini-batch，让不同的样本可以在一个batch中

Dropout：增加网络扰动

Gradient Noise：在梯度中增加一个高斯分布的噪声，噪声随着t的增大而减小

$$\begin{cases} g_{t,i} = g_{t,i} + N(0, \sigma_t^2) \\ \sigma_t = \dfrac{c}{(1+t)^\gamma} \end{cases}$$

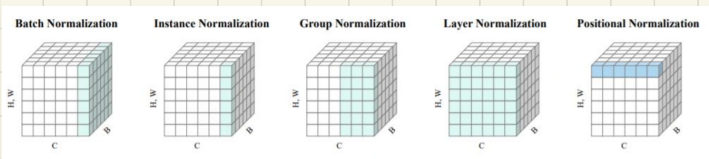
② Learning Rate (Teach your model patiently!)

Warm-Up：学习率开始比较小，之后逐渐变大

Curriculum Learning：开始使用比较简单的数据训练模型，之后再用带有噪声的数据提高模型的泛化能力

Fine-tuning：使用 pre-trained model 进行验证，避免过多的浪费资源和时间

③ Normalization



④ Regularization

10. Conclusion

SGD Team			Adam Team	
基本算法	• SGM • SGDM		基本算法	• Adagard • RMSProp • Adam
结合两者：SWATS				
学习率改进	• LR Range Test • Cyclical LR • SGDR • One-Cycle LR		避免极端学习率	• AMSGard • AdaBound
			Warm-up	• Radam
未来动量	• NAG		引入NAG	• NAdam
正则项修正	• SGDWM		正则项修正	• AdamW
Wrapper of Optimizer : Lookahead				

SGDM	Adam
• Slow • Better convergence • Stable • Smaller generalization gap	• Fast • Possibly non-convergence • Unstable • Larger generalization gap

SGDM	Adam
• Computer vision image classification segmentation object detection	• NLP QA machine translation summary • Speech synthesis • GAN • Reinforcement learning