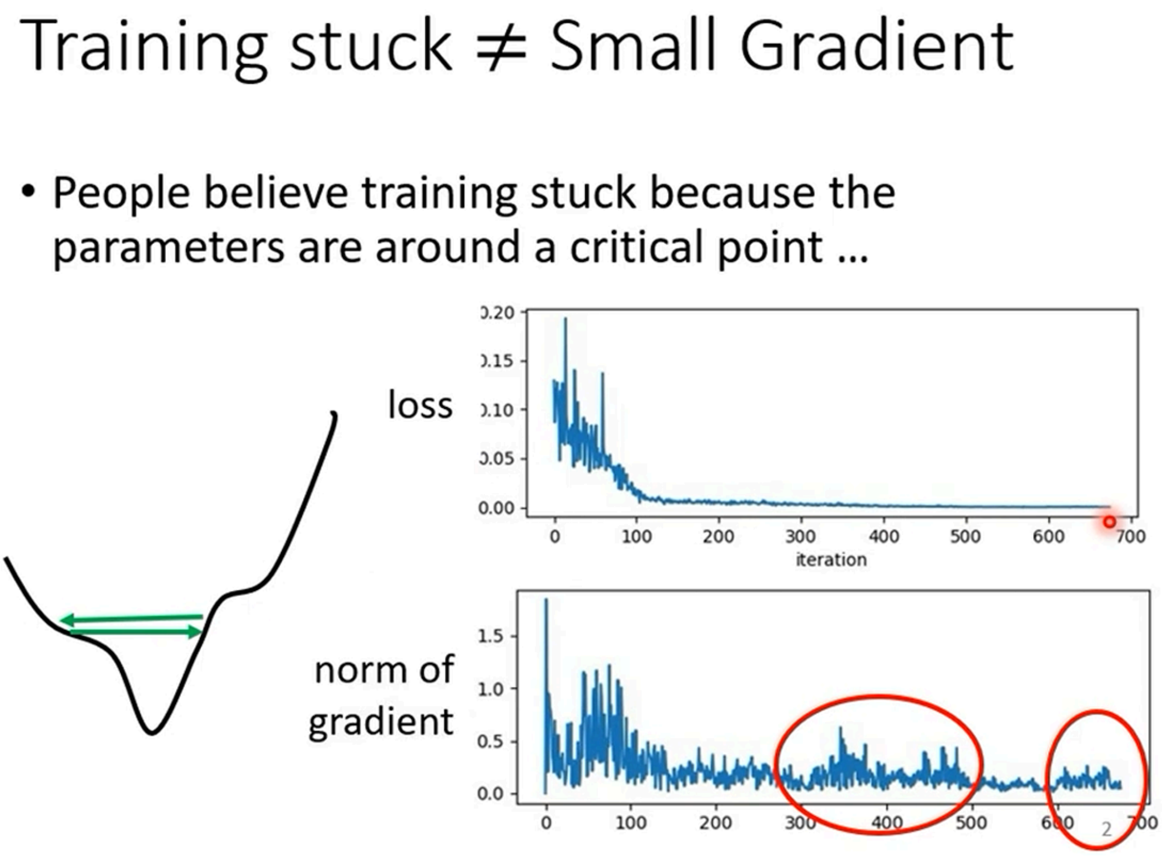
类神经网络训练不起来怎么办（三）自动调整学习率（Learning Rate）

loss不再下降不一定是卡在critical Point，有可能是单纯的loss不会下降。



可以看到，其实还是有梯度的，只是步长太大，导致无法更新loss。

所以我们是需要在梯度较小时（平缓）学习率较大，而在梯度较大时（陡峭）学习率较小。

下面是Apagrad方法：

Root Mean Square

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \boxed{\frac{\eta}{\sigma_i^t}} \boldsymbol{g}_i^t$$

$$\boldsymbol{\theta}_i^1 \leftarrow \boldsymbol{\theta}_i^0 - \frac{\eta}{\sigma_i^0} \boldsymbol{g}_i^0 \qquad \sigma_i^0 = \sqrt{\left(\boldsymbol{g}_i^0\right)^2} = \left|\boldsymbol{g}_i^0\right|$$

$$\boldsymbol{\theta}_i^2 \leftarrow \boldsymbol{\theta}_i^1 - \frac{\eta}{\sigma_i^1} \boldsymbol{g}_i^1 \qquad \sigma_i^1 = \sqrt{\frac{1}{2}\left[\left(\boldsymbol{g}_i^0\right)^2 + \left(\boldsymbol{g}_i^1\right)^2\right]}$$

$$\boldsymbol{\theta}_i^3 \leftarrow \boldsymbol{\theta}_i^2 - \frac{\eta}{\sigma_i^2} \boldsymbol{g}_i^2 \qquad \sigma_i^2 = \sqrt{\frac{1}{3}\left[\left(\boldsymbol{g}_i^0\right)^2 + \left(\boldsymbol{g}_i^1\right)^2 + \left(\boldsymbol{g}_i^2\right)^2\right]}$$

$$\vdots$$

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^t - \frac{\eta}{\sigma_i^t} \boldsymbol{g}_i^t \qquad \sigma_i^t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t}\left(\boldsymbol{g}_i^t\right)^2}$$

6

这样就可以根据不同的梯度，自适应学习率的大小。

下面是RMSProp方法

RMSProp

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^{t} - \boxed{\frac{\eta}{\sigma_i^t}}\boldsymbol{g}_i^{t}$$

$$\boldsymbol{\theta}_i^{1} \leftarrow \boldsymbol{\theta}_i^{0} - \frac{\eta}{\sigma_i^0}\boldsymbol{g}_i^{0} \qquad \sigma_i^0 = \sqrt{\left(\boldsymbol{g}_i^0\right)^2}$$

$$0 < \alpha < 1$$

$$\boldsymbol{\theta}_i^{2} \leftarrow \boldsymbol{\theta}_i^{1} - \frac{\eta}{\sigma_i^1}\boldsymbol{g}_i^{1} \qquad \sigma_i^1 = \sqrt{\alpha\left(\sigma_i^0\right)^2 + (1-\alpha)\left(\boldsymbol{g}_i^1\right)^2}$$

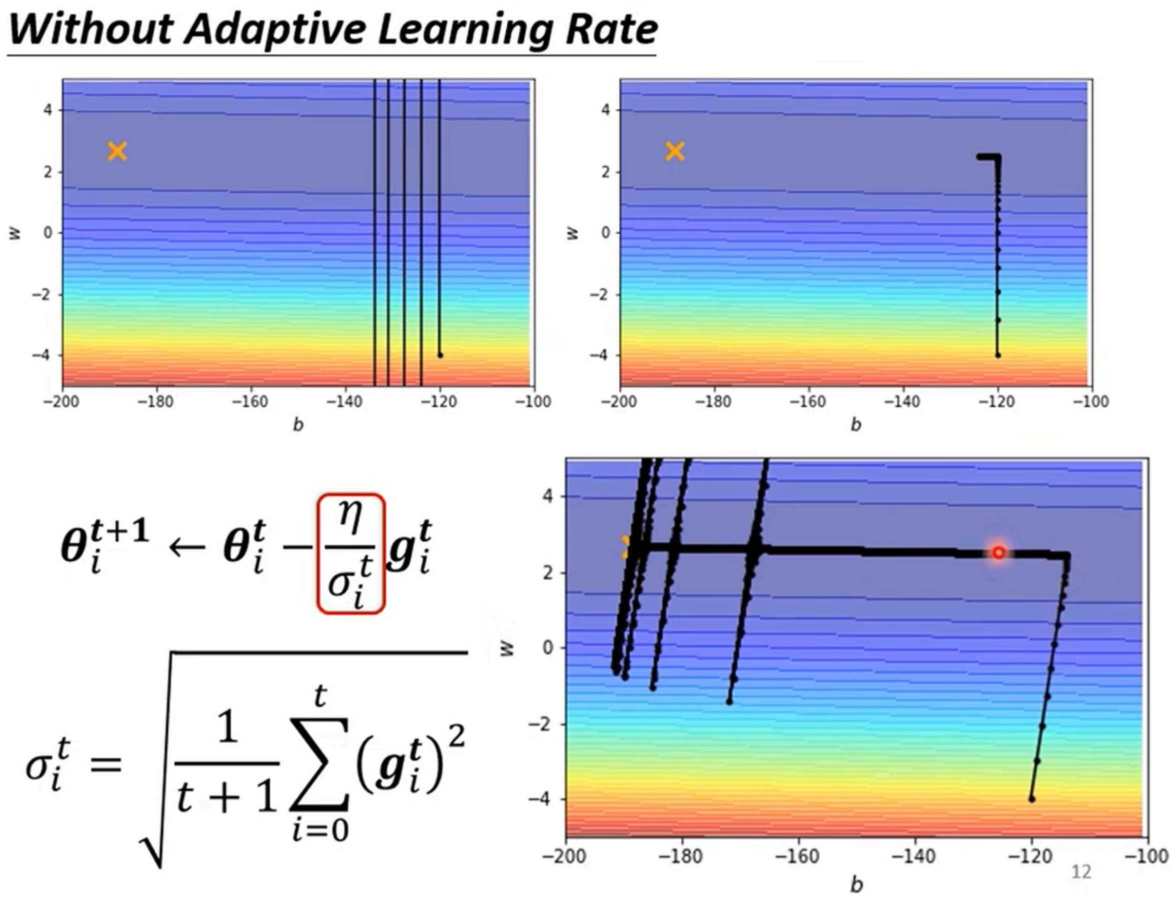
$$\boldsymbol{\theta}_i^{3} \leftarrow \boldsymbol{\theta}_i^{2} - \frac{\eta}{\sigma_i^2}\boldsymbol{g}_i^{2} \qquad \sigma_i^2 = \sqrt{\alpha\left(\sigma_i^1\right)^2 + (1-\alpha)\left(\boldsymbol{g}_i^2\right)^2}$$

$$\vdots$$

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^{t} - \frac{\eta}{\sigma_i^t}\boldsymbol{g}_i^{t} \qquad \sigma_i^t = \sqrt{\alpha\left(\sigma_i^{t-1}\right)^2 + (1-\alpha)\left(\boldsymbol{g}_i^t\right)^2}$$

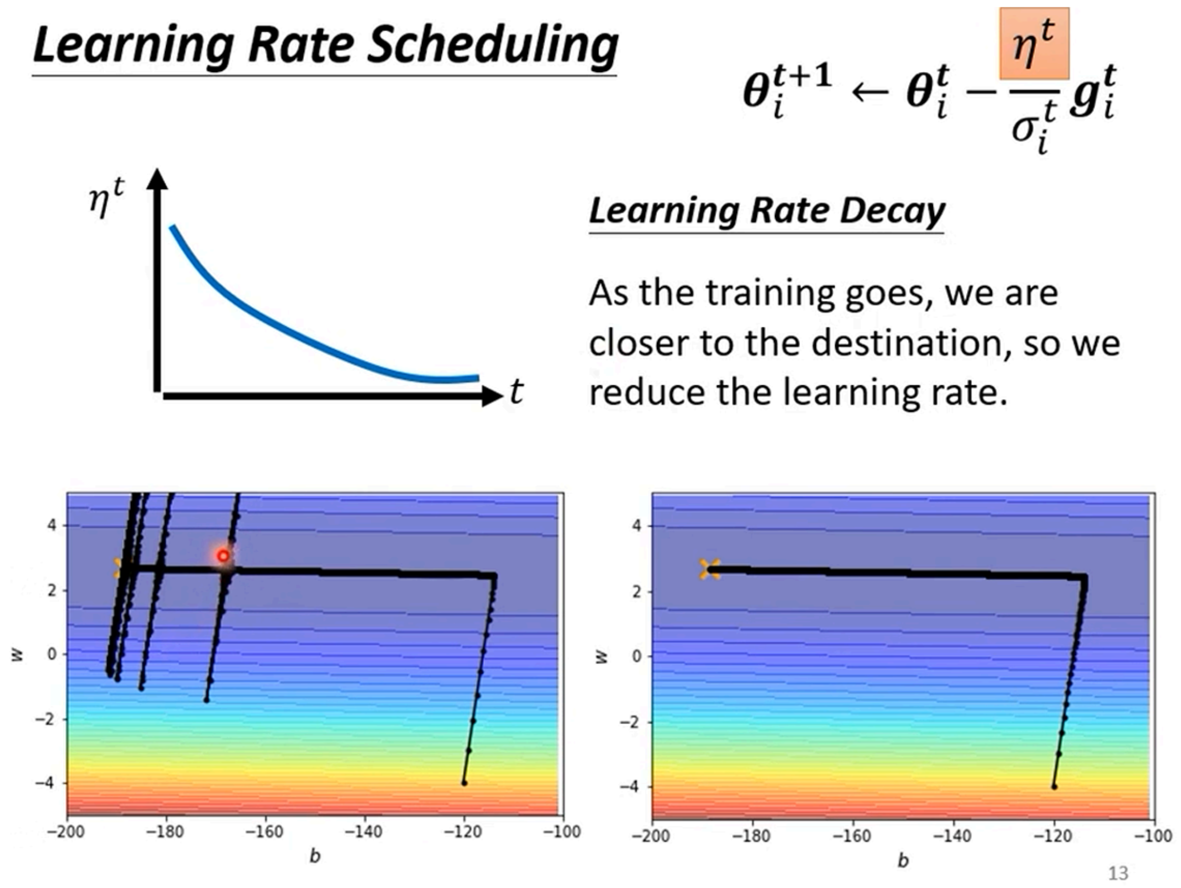
9

现在常用的就是Adam，就是RMSProp+Momentum。



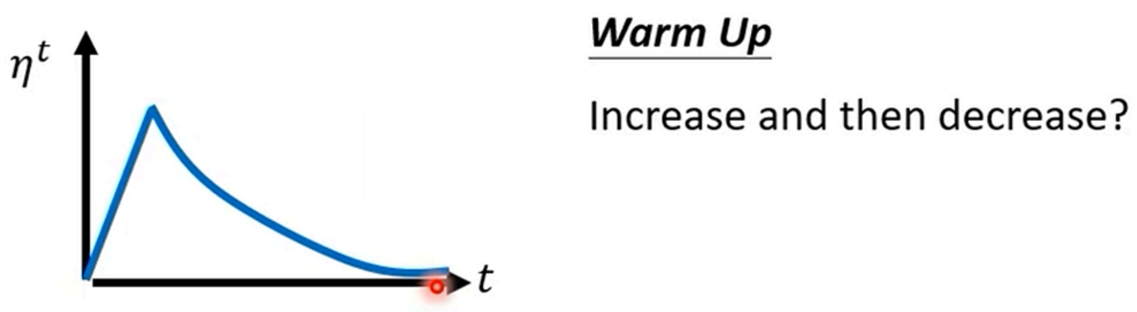
可以看到下面那个图就是加入了Apagrad的曲线，相对于没有自适应的Learning Rate，这个明显更靠近目标点，但是其中的爆炸，是因为在y方向先前积累了很多小的σ，达到一定程度就爆炸，但是爆炸跑远之后又会使梯度变大，从而学习率下降，恢复到原来状态。

当然上面的情况也是可以解决的。可以用Learning Rate decay



可以看到，我们随着训练次数的增加，不断地靠近目标点，所以我们不断地减少学习率。

除此以外我们也可以用warm up，先变大后变小。



一般训练BERT时会用到warm up。

关于优化的改良版：

Summary of Optimization

(Vanilla) Gradient Descent

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^{t} - \eta \boldsymbol{g}_i^{t}$$

Various Improvements

$$\boldsymbol{\theta}_i^{t+1} \leftarrow \boldsymbol{\theta}_i^{t} - \frac{\eta^t}{\sigma_i^t} \boldsymbol{m}_i^{t}$$

η^t: Learning rate scheduling

$\boldsymbol{m}_i^t$: Momentum: weighted sum of the previous gradients

Consider direction

σ_i^t: root mean square of the gradients

only magnitude

17