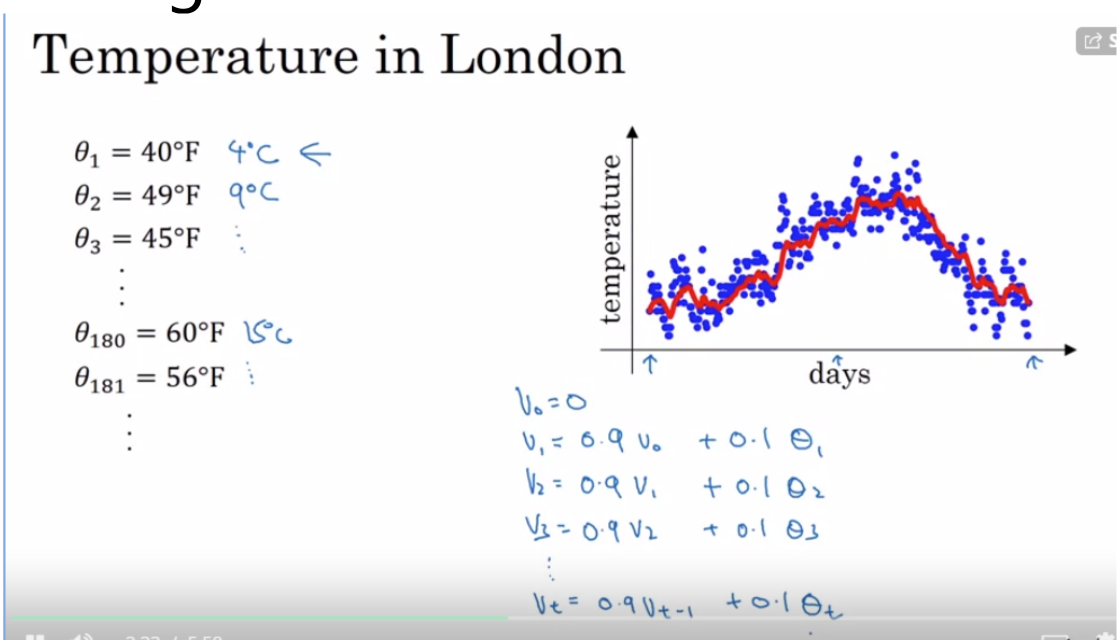
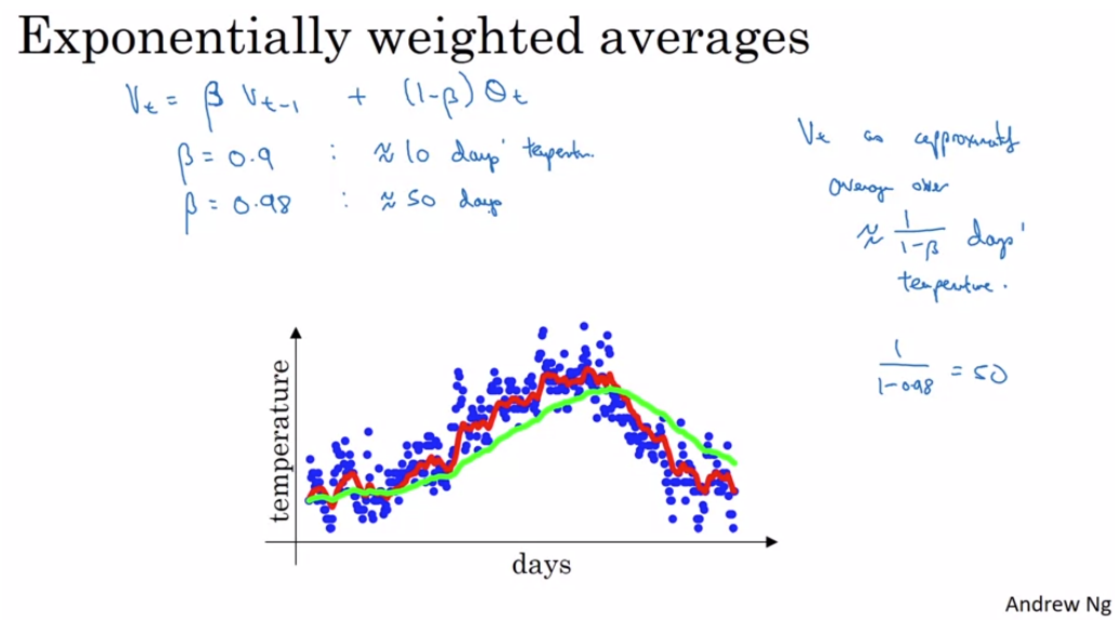
Week 2-2 Adam Algo, GC with Momentum (Exponentially Weight MA) + RMS prop

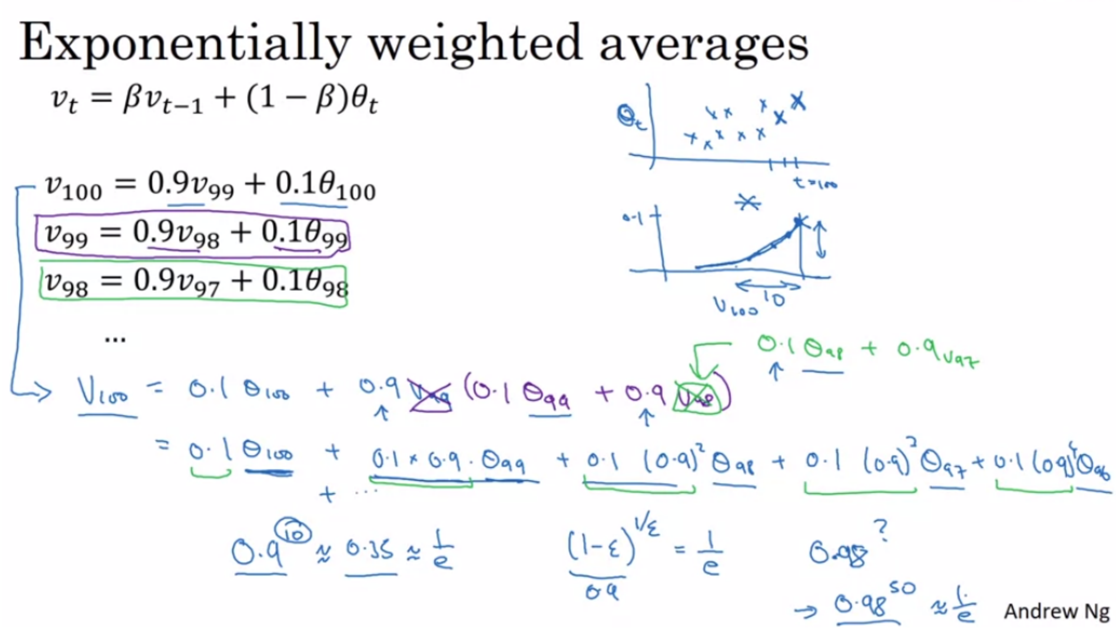
笔记本:	DL 2 - Deep NN Hyperparameter Tunning, Regularization & Optimization		
创建时间:	2021/1/9 11:15	**更新时间:**	2021/1/9 12:33

adam paper: https://arxiv.org/pdf/1412.6980.pdf

Exponentially weighted moving averages







(all these weights approximately add up to 1)
after 1/epi -> 1/e weight, rather small

Implementing exponentially weighted averages

$v_0 = 0$
$v_1 = \beta v_0 + (1-\beta)\,\theta_1$
$v_2 = \beta v_1 + (1-\beta)\,\theta_2$
$v_3 = \beta v_2 + (1-\beta)\,\theta_3$
...

$V_\theta := 0$
$V_\theta := \beta v + (1-\beta)\theta_1$
$V_\theta := \beta v + (1-\beta)\theta_2$
$\vdots$

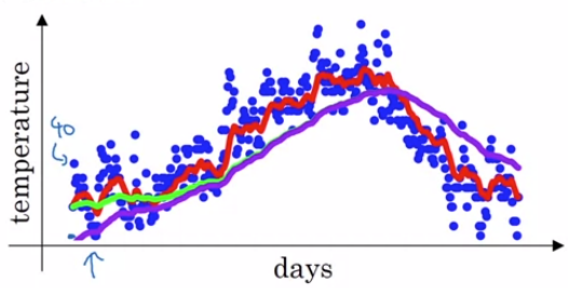
$\rightarrow V_\theta = 0$
Repeat {
Get next θ_t
$V_\theta := \beta V_\theta + (1-\beta)\theta_t \leftarrow$
}

Andrew Ng

(memory efficient)

Bias correction

Bias correction



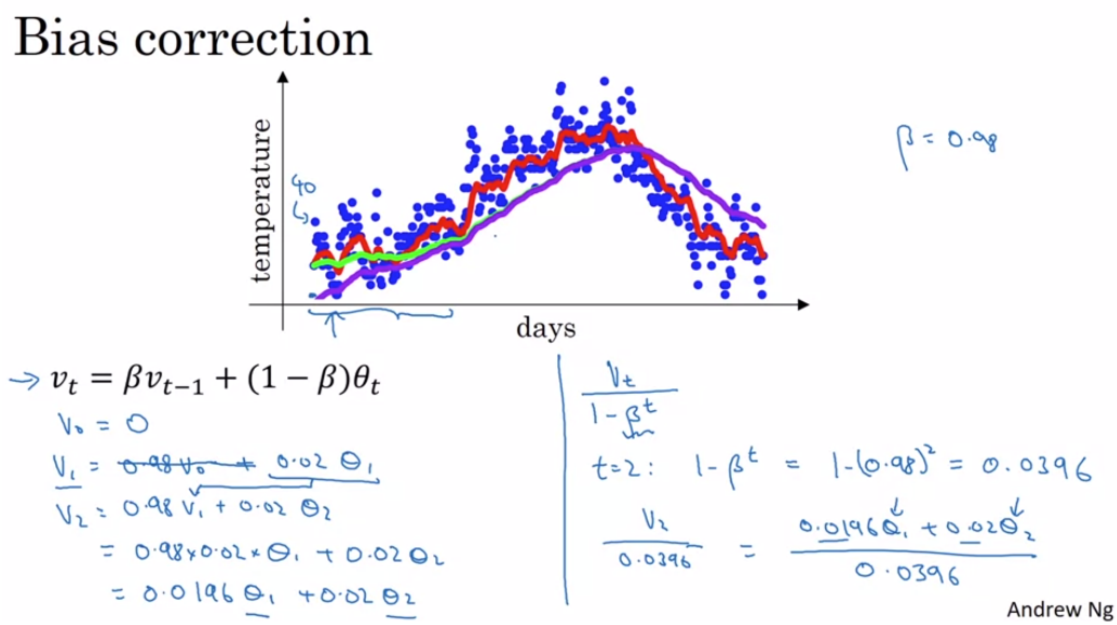
$\beta = 0.98$

$\rightarrow v_t = \beta v_{t-1} + (1-\beta)\theta_t$

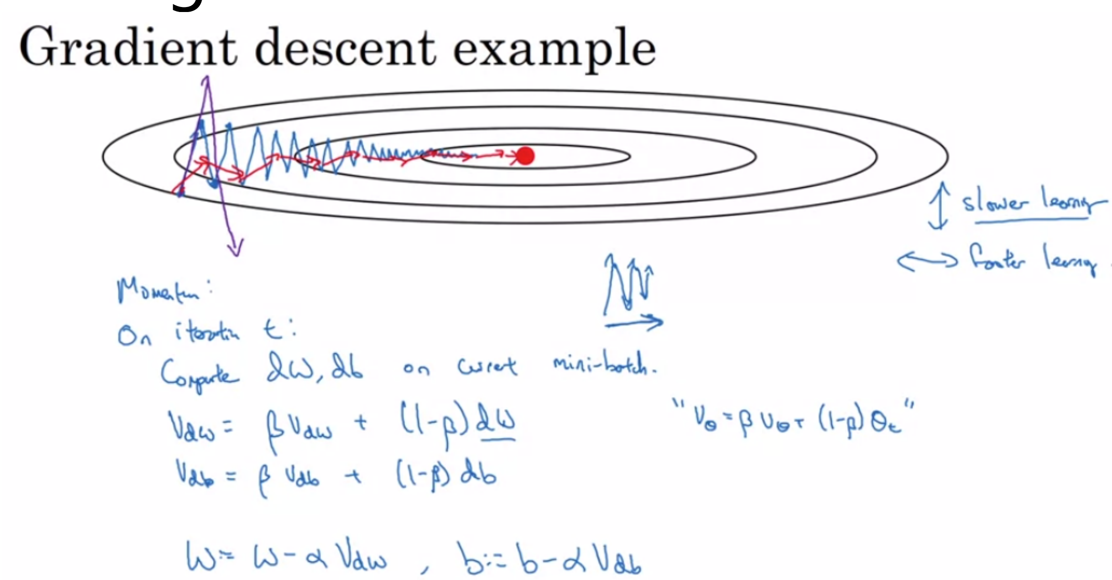
$V_0 = 0$
$V_1 = \cancel{0.98 V_0 +} 0.02\,\theta_1$
$V_2 = 0.98\,V_1 + 0.02\,\theta_2$
$= 0.98 \times 0.02 \times \theta_1 + 0.02\,\theta_2$
$= 0.0196\,\theta_1 + 0.02\,\theta_2$

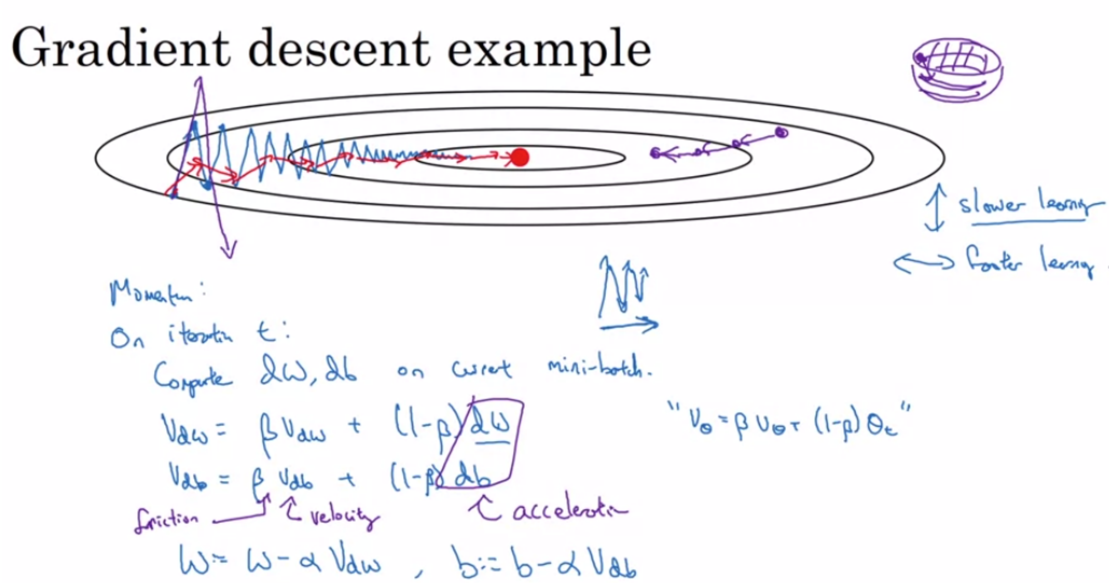
Andrew Ng

if we initialize with 0, it gives the purple curve



average of these derivatives





(add friction so that its horizontal speed along the bowl gets smaller)

Implementation details

$V_{dw} = 0$

On iteration t:

Compute dW, db on the current mini-batch

$$v_{dW} = \beta v_{dW} + (1-\beta)dW$$

$$v_{db} = \beta v_{db} + (1-\beta)db$$

$$W = W - \alpha v_{dW}, \quad b = b - \alpha v_{db}$$

$$\cancel{\frac{V_{dw}}{1-\beta^t}}$$

Hyperparameters: α, β $\qquad$ $\beta = 0.9$

average over last ≈10 gradients

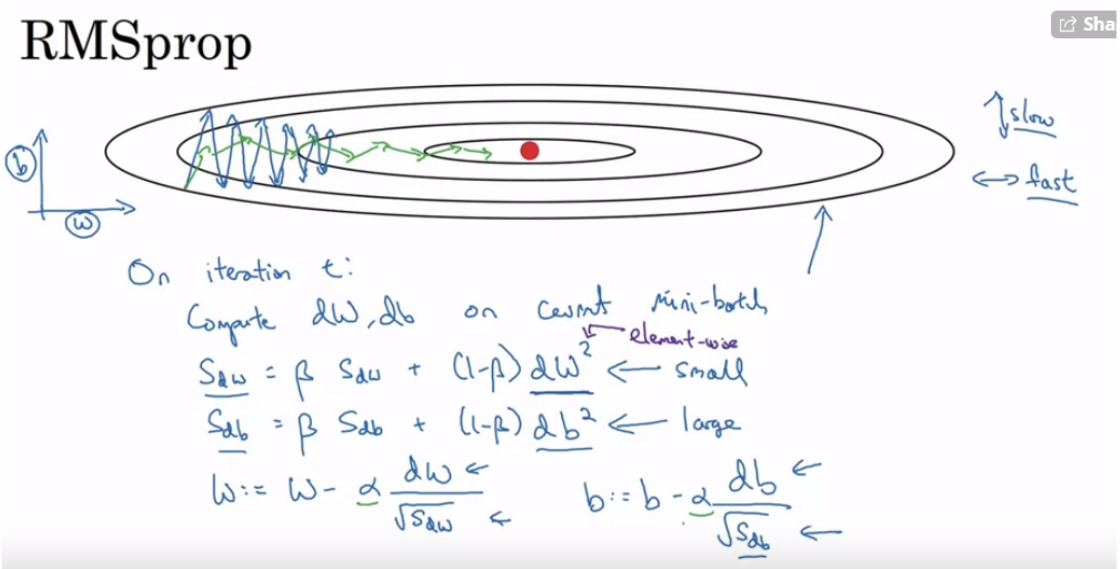
Andrew Ng

In practice, people don't usually do this (bias correction) because after just ten iterations, your moving average will have warmed up and is no longer a bias estimate.

We sometimes omit the (1-beta) term, but it is not so intuitive and we may need to adjust the learning rate correspondingly

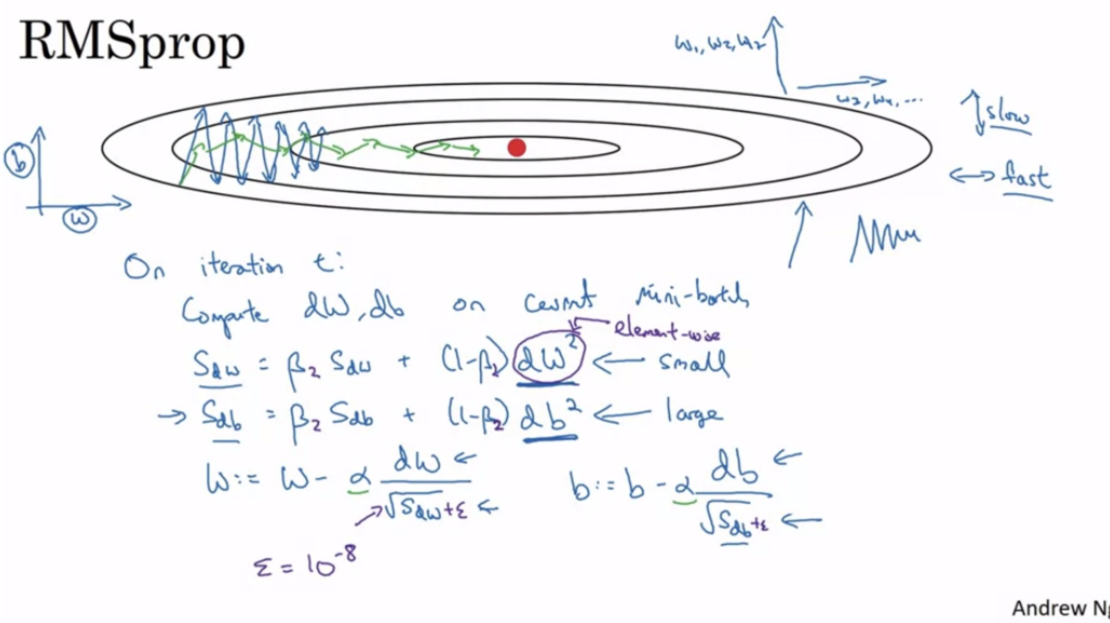
---

RMSprop



larger db gets more scale division, also enables larger learning rate

We also add a epsilon for numerical stability (division by zero)



Adam

Adam optimization algorithm

$V_{dw}=0$, $S_{dw}=0$. $V_{db}=0$, $S_{db}=0$

On iteration t:

Compute dW, db using current mini-batch

$V_{dw} = \beta_1 V_{dw} + (1-\beta_1) dW$, $V_{db} = \beta_1 V_{db} + (1-\beta_1) db$ ← "momentum" β_1

$S_{dw} = \beta_2 S_{dw} + (1-\beta_2) dW^2$, $S_{db} = \beta_2 S_{db} + (1-\beta_2) db$ ← "RMSprop" β_2

$V^{corrected}_{dw} = V_{dw}/(1-\beta^t)$, $V^{corrected}_{db} = V_{db}/(1-\beta_1^t)$

$S^{corrected}_{dw} = S_{dw}/(1-\beta_2^t)$, $S^{corrected}_{db} = S_{db}/(1-\beta_2^t)$

$$W := W - \alpha \frac{V^{corrected}_{dw}}{\sqrt{S^{corrected}_{dw}} + \varepsilon} \qquad b := b - \alpha \frac{V^{corrected}_{db}}{\sqrt{S^{corrected}_{db}} + \varepsilon}$$

Andrew Ng

(there is db^2)

Hyperparameters choice:

→ α : needs to be tune

→ β_1 : 0.9 → (dw)

→ β_2 : 0.999 → (dw^2)

→ ε : 10^{-8}

Adam: Adaptive moment estimation

Adaptive moment estimation (first order momentum + second order momentum)