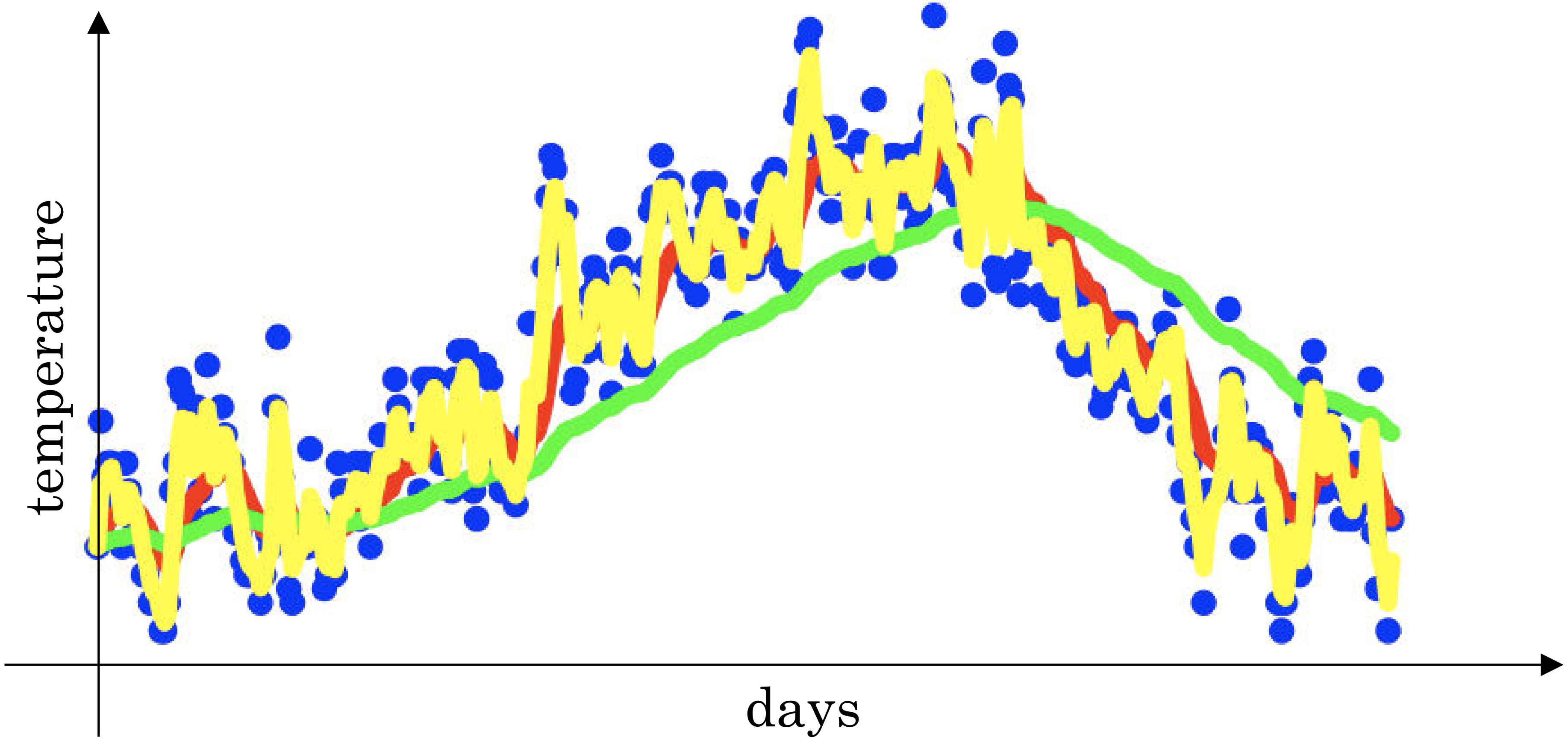
deeplearning.ai

Optimization Algorithms

Understanding exponentially weighted averages

Exponentially weighted averages

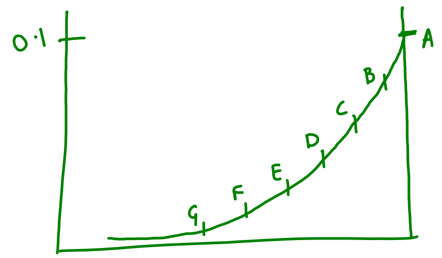
$$v_t = \beta v_{t-1} + (1-\beta)\theta_t$$



Andrew Ng

Exponentially weighted averages

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



$$v_{100} = 0.9v_{99} + 0.1\theta_{100}$$

$$v_{99} = 0.9v_{98} + 0.1\theta_{99}$$

$$v_{98} = 0.9v_{97} + 0.1\theta_{98}$$

$\ldots$

$A = 0.1$

$B = 0.1(0.9)$

$C = 0.1(.9)^2$

$D = 0.1(.9)^3$

$$V_{100} = 0.1\,\theta_{100} + 0.9V_{99}$$

$$= 0.1\,\theta_{100} + 0.9(0.1\theta_{99} + .9\underbrace{V_{98}}_{\hookrightarrow\, 0.1\cdot\theta_{98} + 0.9V_{97}})$$

$$= 0.1\theta_{100} + 0.1(.9)\theta_{99} + 0.1(.9)^2\theta_{98} + 0.1(.9)^3\theta_{97} + 0.1(.9)^4\theta_{96} + \ldots 0.1(.9)^n\theta_{100-n}$$

we know $(1-\varepsilon)^{1/\varepsilon} = \frac{1}{e}$

So if we consider $\beta = 0.1$

$(1-0.1)^{1/0.1} = (.9)^{10} \simeq .35 = \frac{1}{e}$

so if $\beta = 0.1$

then it takes about 10 days for the graph to decay from 0.1 to $\frac{1}{3}$rd of 0.1 in height

If $\beta = 0.02$

then it would take

$(.98)^{50} = .35$

ie; it takes 50 days to reach $\frac{1}{3}$rd of 0.1 if $\beta = 0.02$

This is where we got the formula for the earlier slide, that we're avging over $\frac{1}{1-\beta}$ days

Andrew Ng

Implementing exponentially weighted averages

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

$V_\theta = 0$

$V_\theta = \beta V + (1-\beta)\theta_1$

$V_\theta = \beta V + (1-\beta)\theta_2$

$V_\theta = \beta V + (1-\beta)\theta_3$

$\vdots$

$\Rightarrow$ we dont have to maintain $V_1, V_2, V_3 \dots$

```
V_θ = 0
Repeat {
    Get θ for next day
        V_θ = βV_θ + (1-β)θ_t
}
```

Low memory but not the best way to compute moving avg

Andrew Ng