deeplearning.ai

a key component of several optimization algorithms

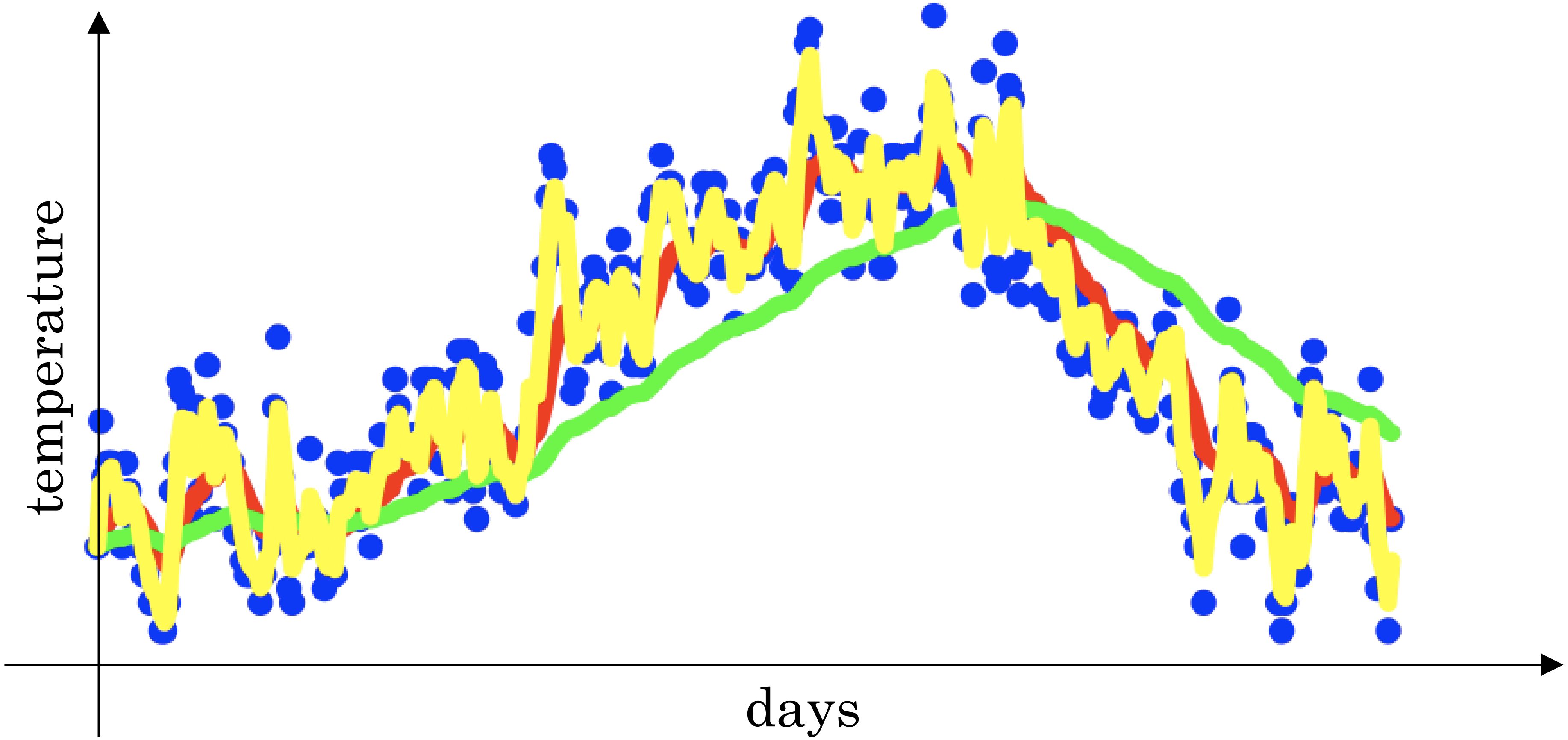
# Optimization Algorithms

## Understanding exponentially weighted averages

# Exponentially weighted averages

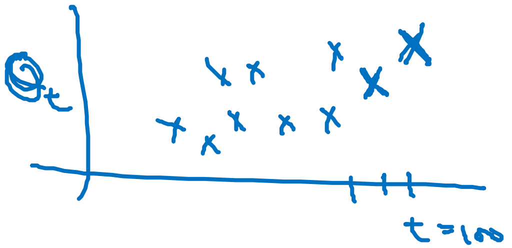
$$\boxed{v_t = \beta v_{t-1} + (1-\beta)\theta_t}$$

$\beta = 0.9$ $0.98$ $0.5$

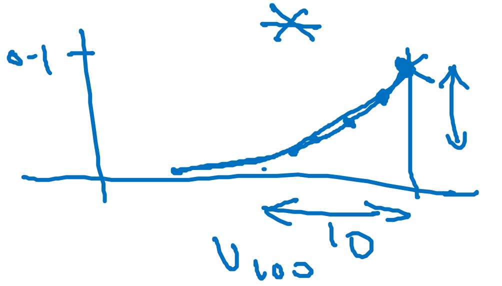


# Exponentially weighted averages

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




So it's really taking the daily temperature, multiply with this exponentially decaying function and summing it up.




$$\approx \frac{1}{1-\beta}$$

$$\varepsilon = 1-\beta$$

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

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

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

...

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

$$v_{100} = 0.1\theta_{100} + 0.9\cancel{v_{99}}\left(0.1\theta_{99} + 0.9\cancel{v_{98}}\right)$$

$$= 0.1\theta_{100} + 0.1\times 0.9\cdot\theta_{99} + 0.1(0.9)^2\theta_{98} + 0.1(0.9)^3\theta_{97} + 0.1(0.9)^4\theta_{96} + \dots$$

all of these coefficients, add up to one or add up to very close to one, up to a detail called bias correction which we'll talk about in the next video.

$$0.9^{10} \approx 0.35 \approx \frac{1}{e} \qquad \underbrace{(1-\varepsilon)}_{0.9}^{1/\varepsilon} = \frac{1}{e} \qquad 0.98\,?$$

After 10 days, that the weight decays to less than about a third of the weight of the current day.

$$\varepsilon = 0.02 \rightarrow 0.98^{50} \approx \frac{1}{e}$$

# Implementing exponentially weighted averages

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

...

theta to denote that V is computing this exponentially weighted average of the parameter theta.

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

$\rightarrow V_\theta = 0$

Repeat {

Get next $\theta_t$

$V_\theta := \beta V_\theta + (1-\beta)\theta_t \leftarrow$

}

So one of the advantages of this exponentially weighted average formula, is that it takes very little memory.

If you were to compute a moving window, where you explicitly sum over the last 10 days, the last 50 days temperature just divide by 10 or divide by 50, that usually gives you a better estimate.

But the disadvantage of that, of explicitly keeping all the temperatures around and sum of the last 10 days, it requires more memory, and it's just more complicated to implement and is computationally more expensive.

Andrew Ng