Introduction to Robot Intelligence [Spring 2023]

Gradient Descent

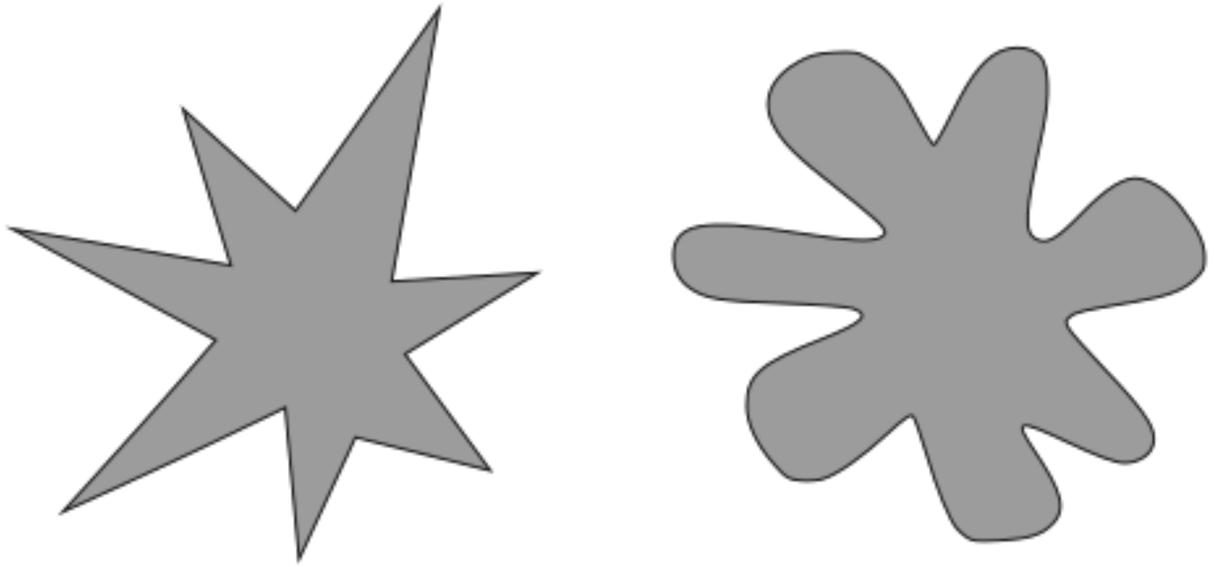
February 2, 2023

Lerrel Pinto

Some class logistics

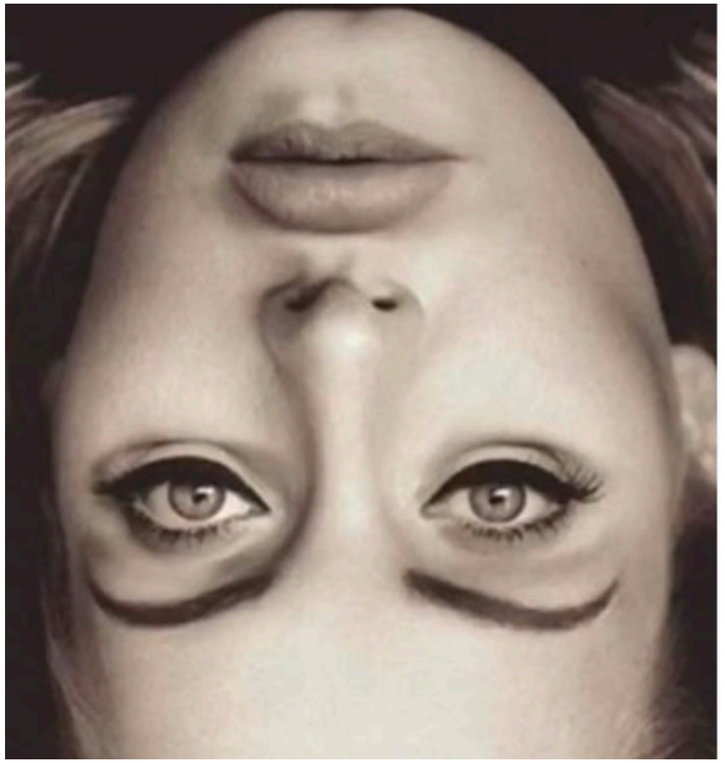
- HW1 is due today!

Musings



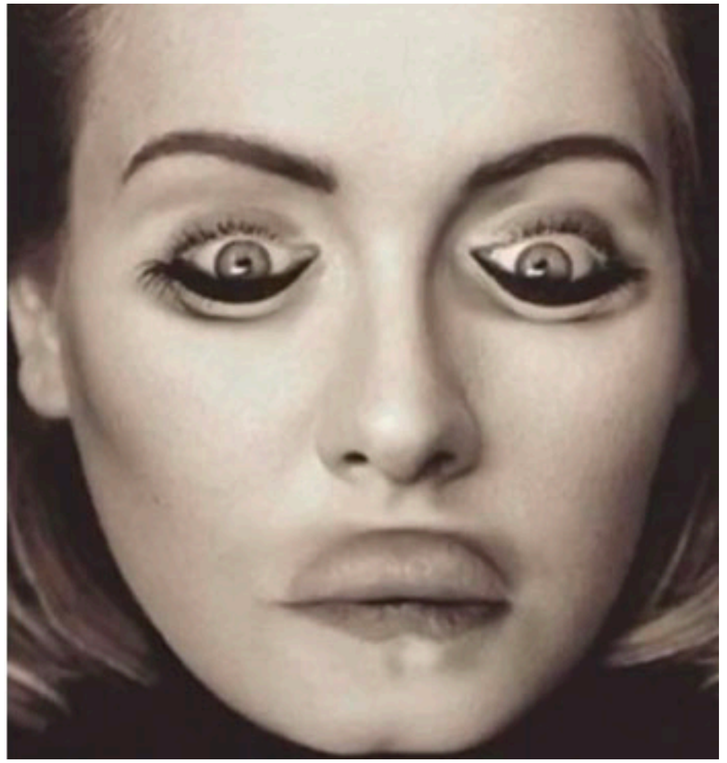
https://en.wikipedia.org/wiki/Bouba/kiki_effect

Musings



https://en.wikipedia.org/wiki/Thatcher_effect

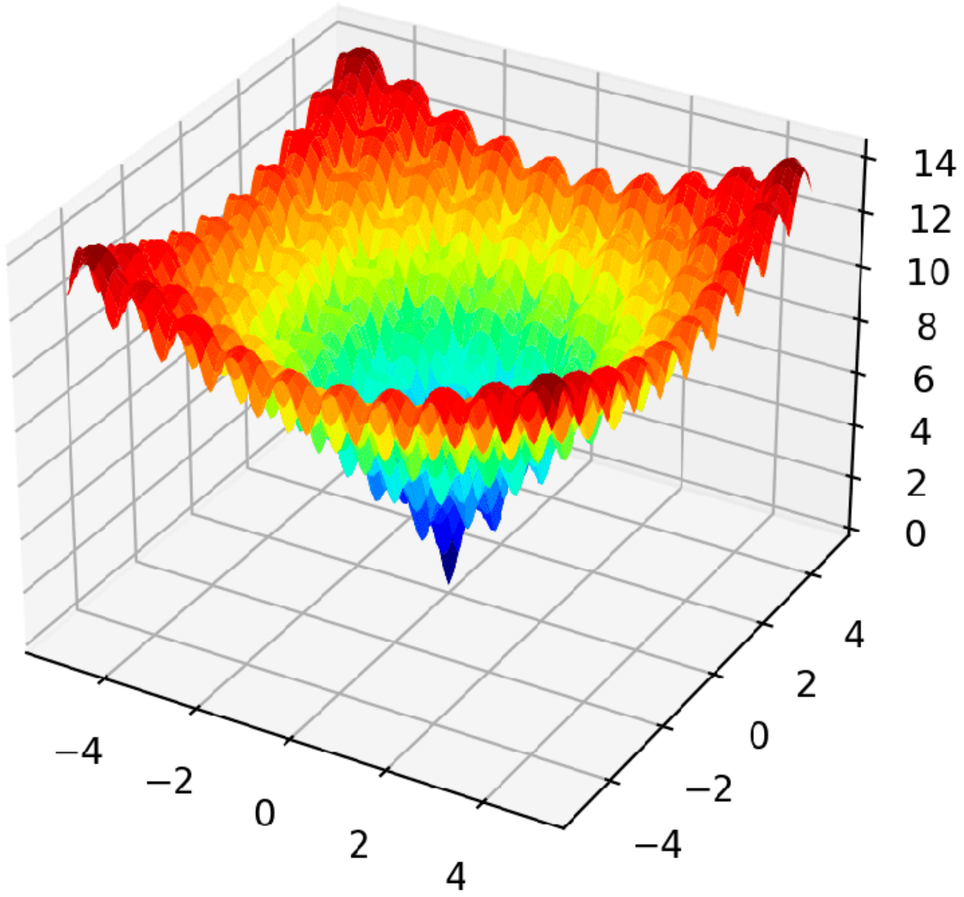
Musings



https://en.wikipedia.org/wiki/Thatcher_effect

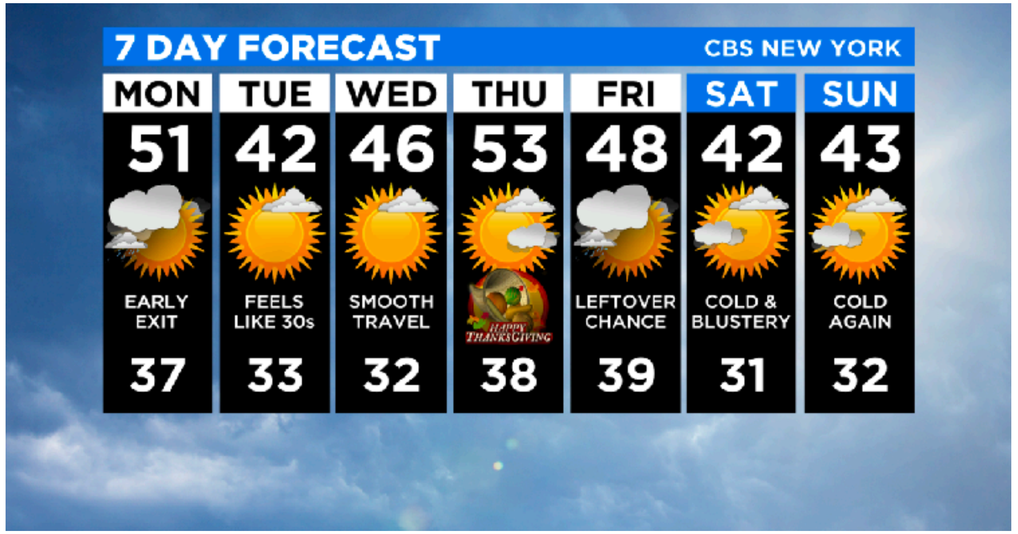
Goal for today: A soft introduction to gradient descent

Solve this:



What if you do not have the analytic equation?

Financial markets



Weather models

Image recognition

Cross Entropy Method

Initialize $\mu \in \mathbb{R}^d, \sigma \in \mathbb{R}^d$

for iteration $= 1, 2, \ldots$ **do**

Collect n samples of $\theta_i \sim N(\mu, \text{diag}(\sigma))$

Perform a noisy evaluation $R_i \sim \theta_i$

Select the top $p\%$ of samples (e.g. $p = 20$), which we'll call the **elite set**

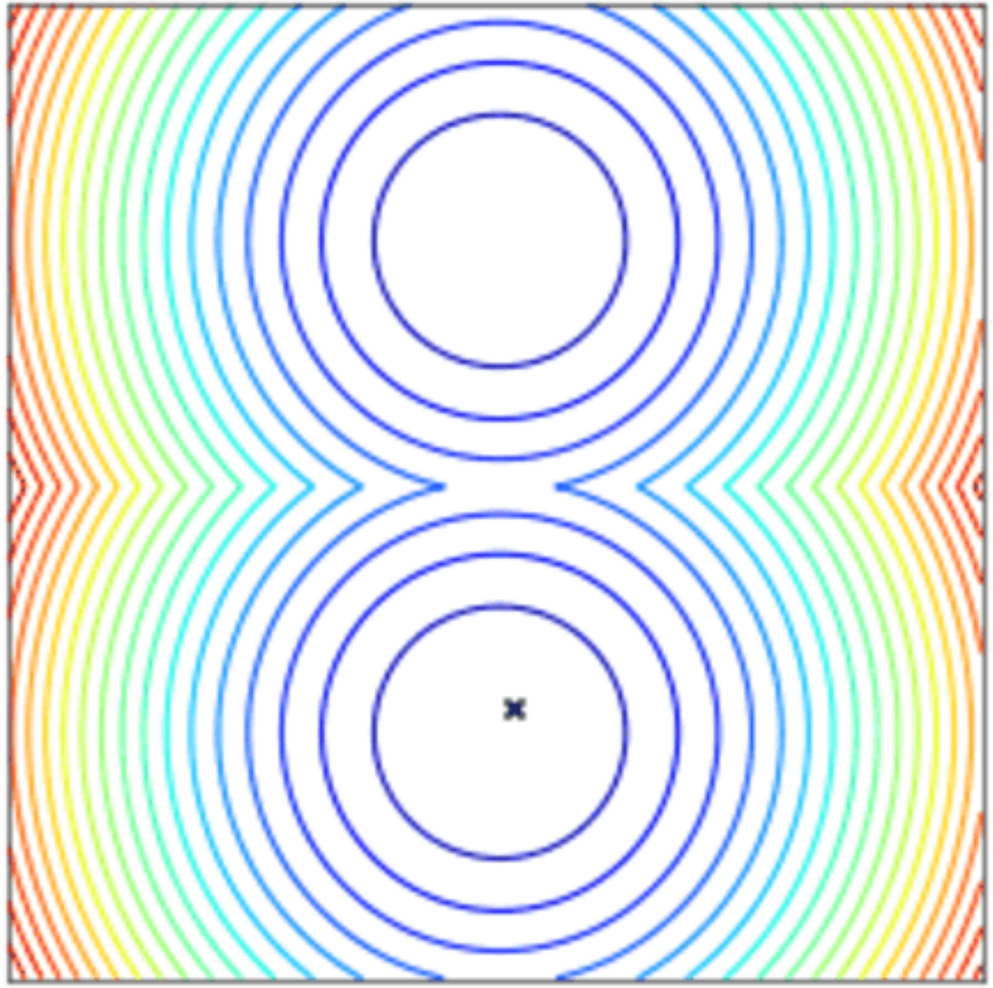
Fit a Gaussian distribution, with diagonal covariance, to the elite set, obtaining a new μ, σ.

end for

Return the final μ.

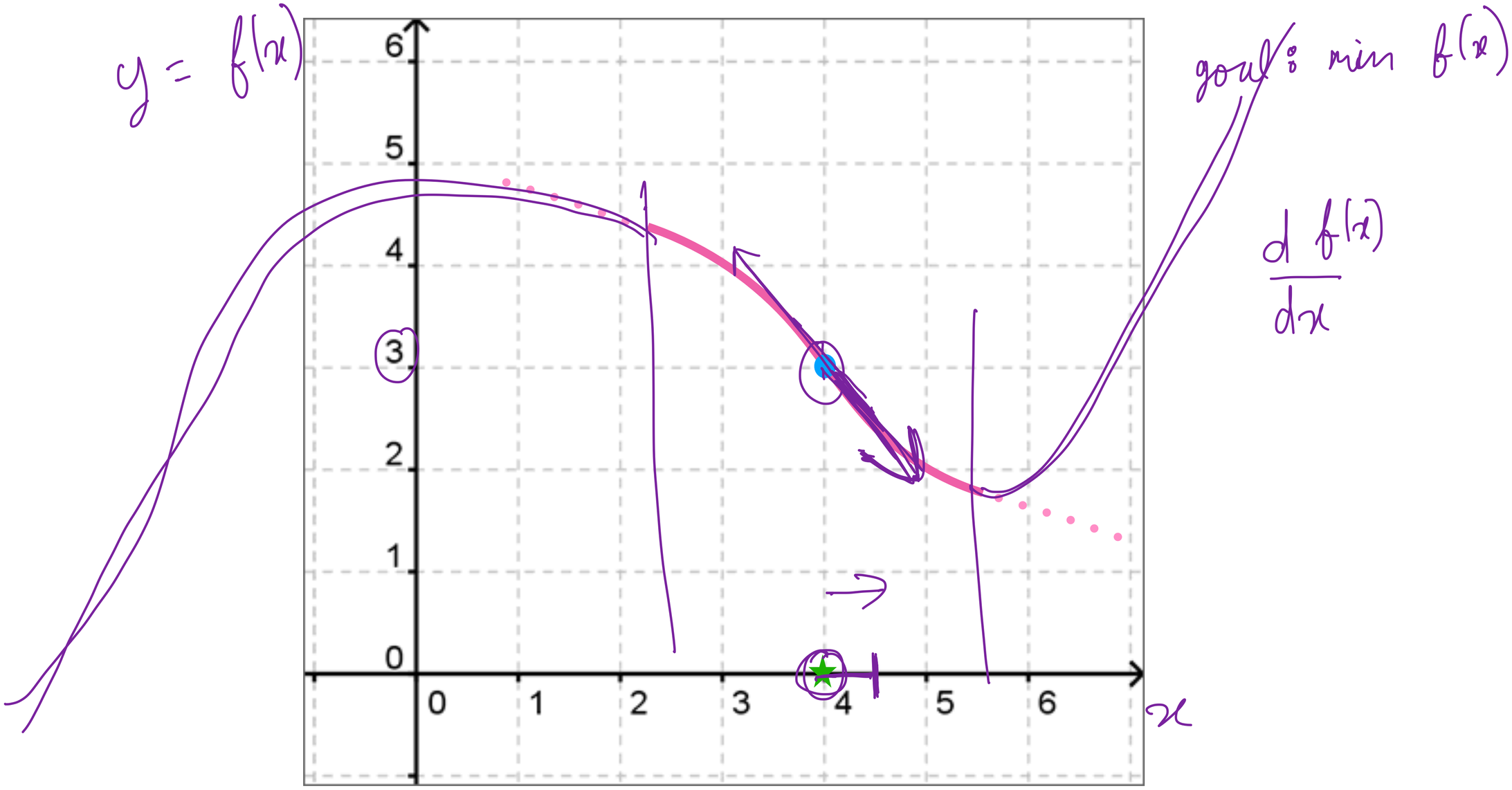
Slide credits: MLSS 2016 on Deep Reinforcement Learning by John Schulman

Visualizing DFO with two variables

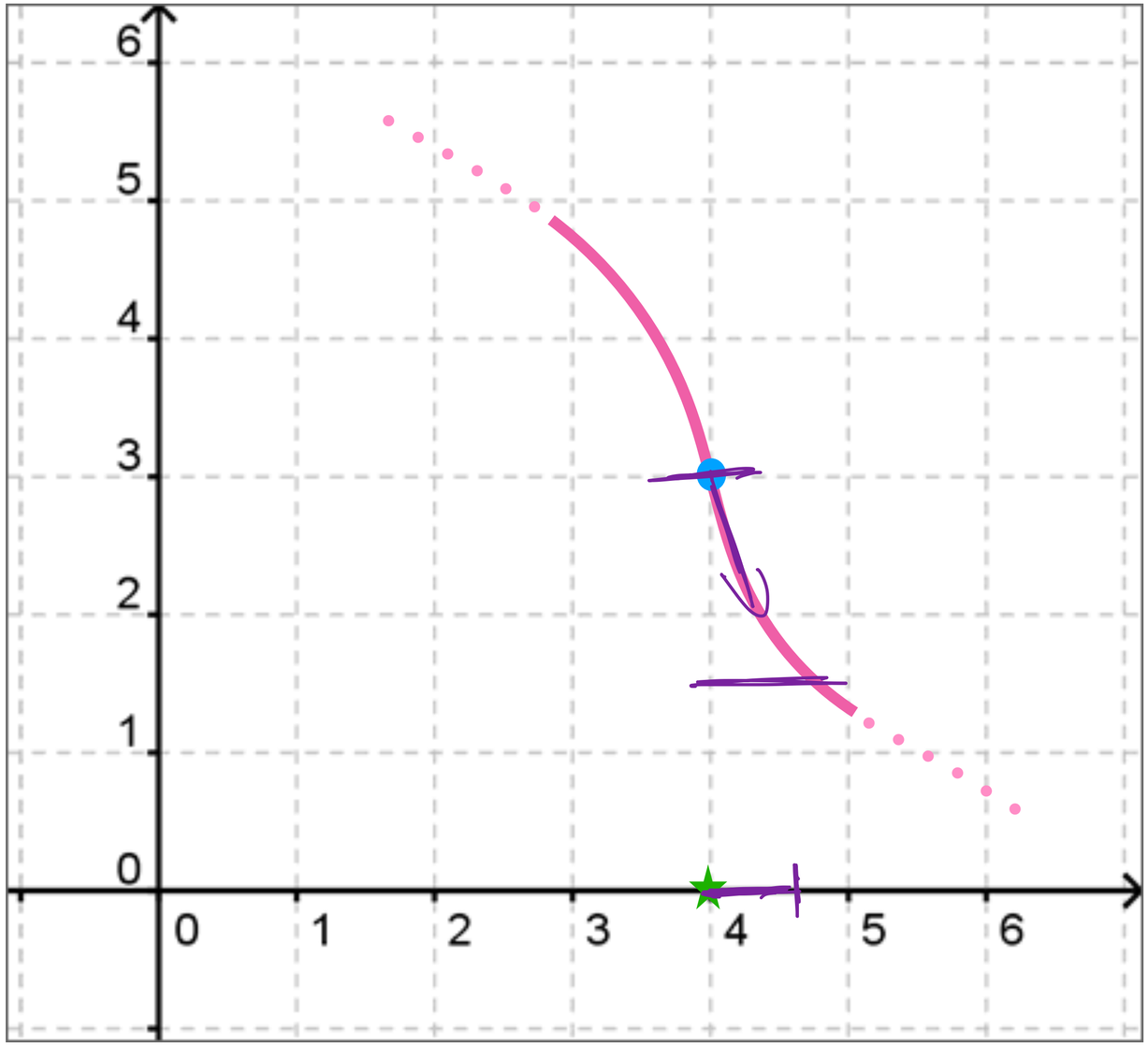


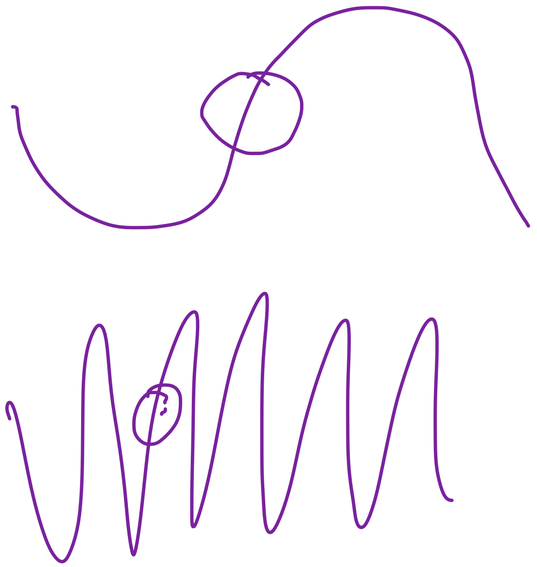
Lets go back to having analytic information

Where to search next?

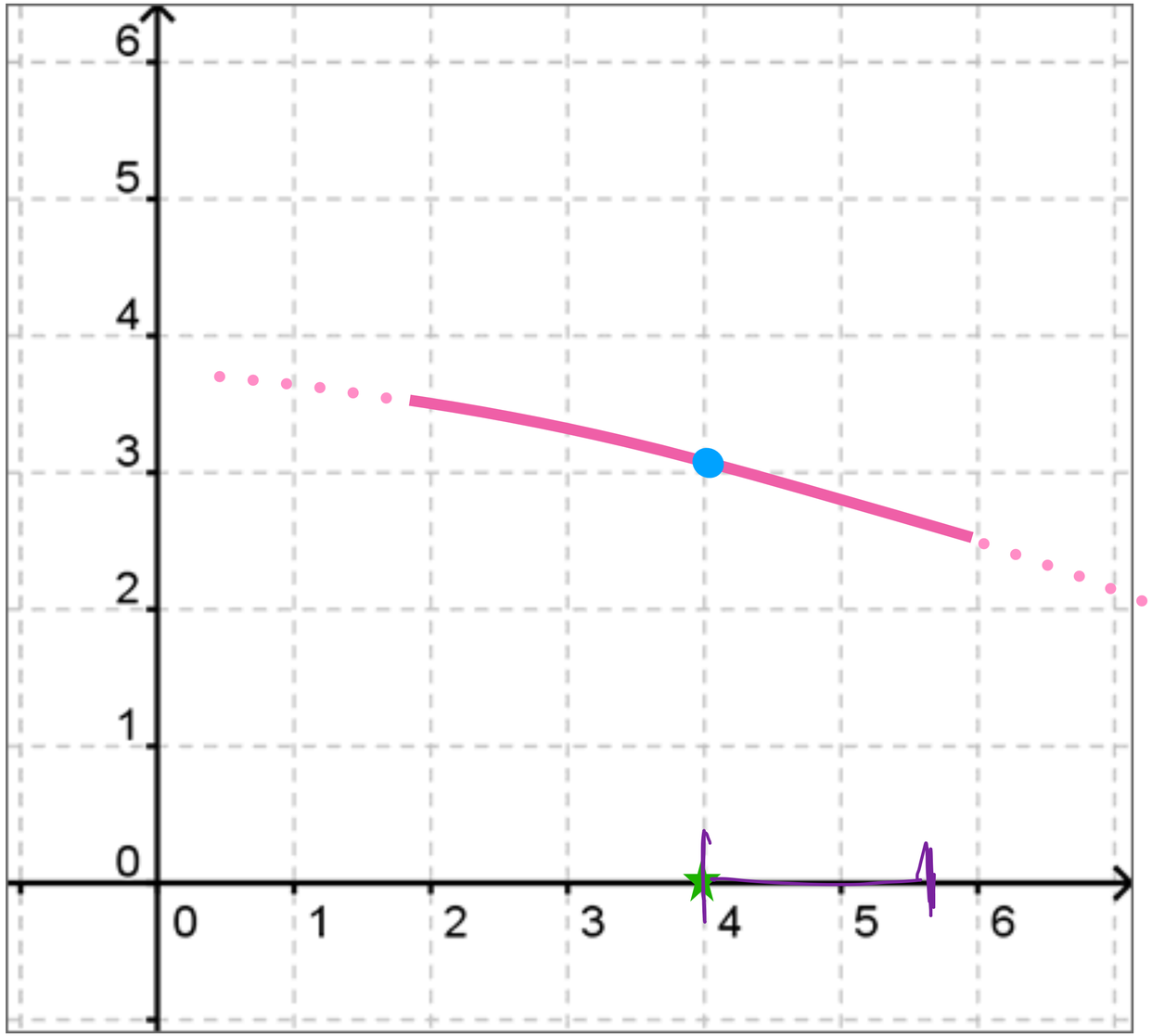


Where to search next?

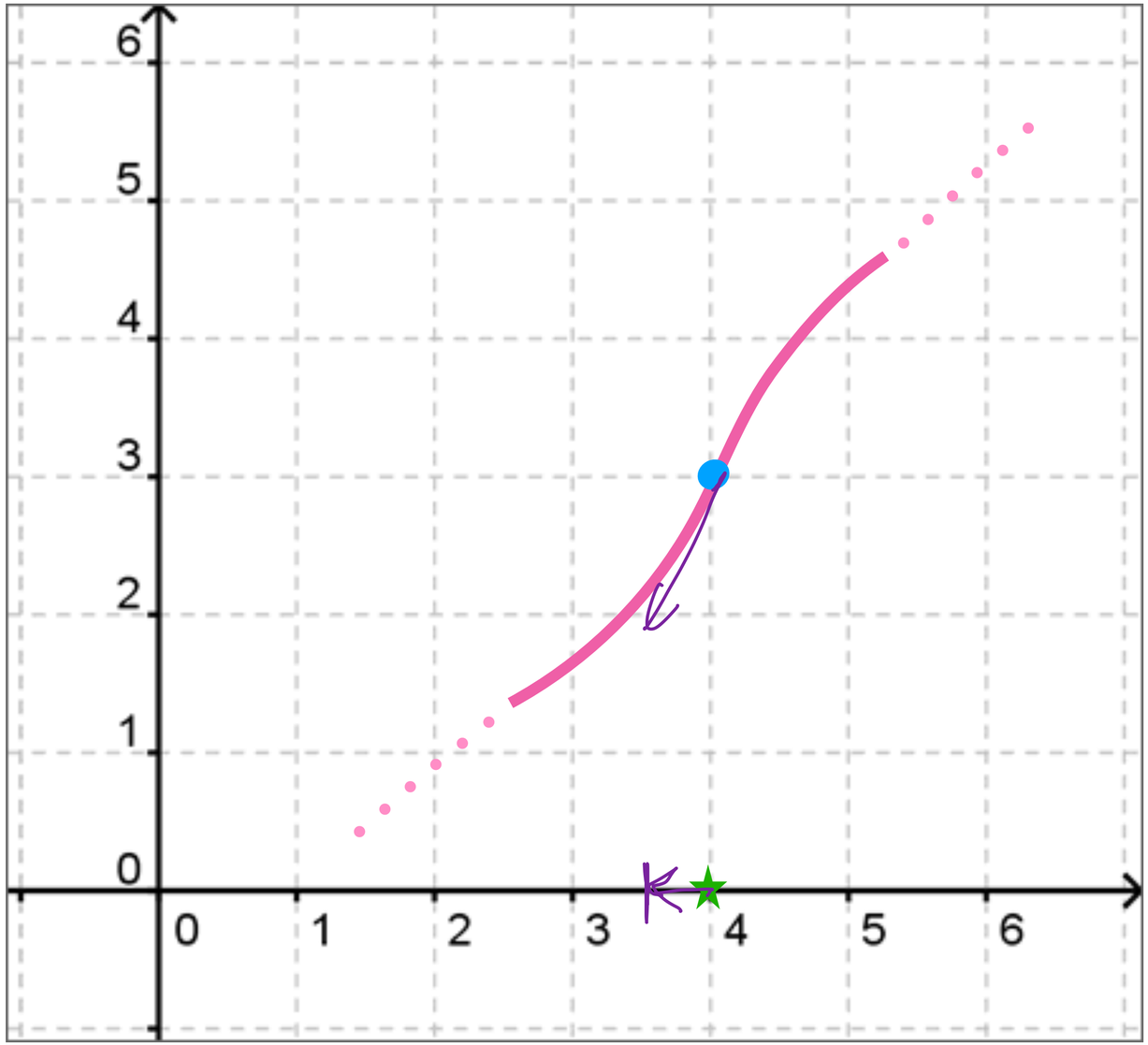




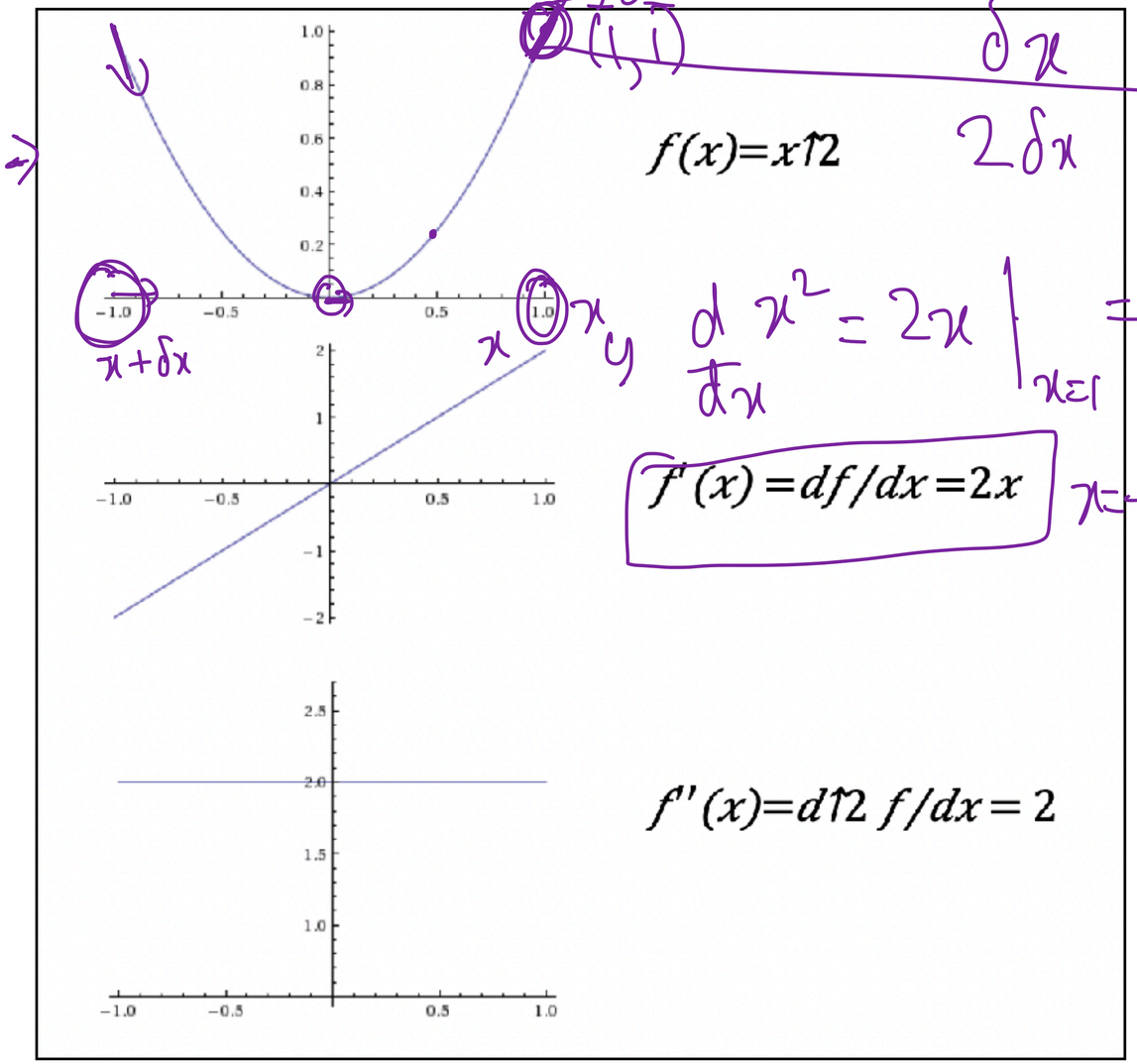
Where to search next?

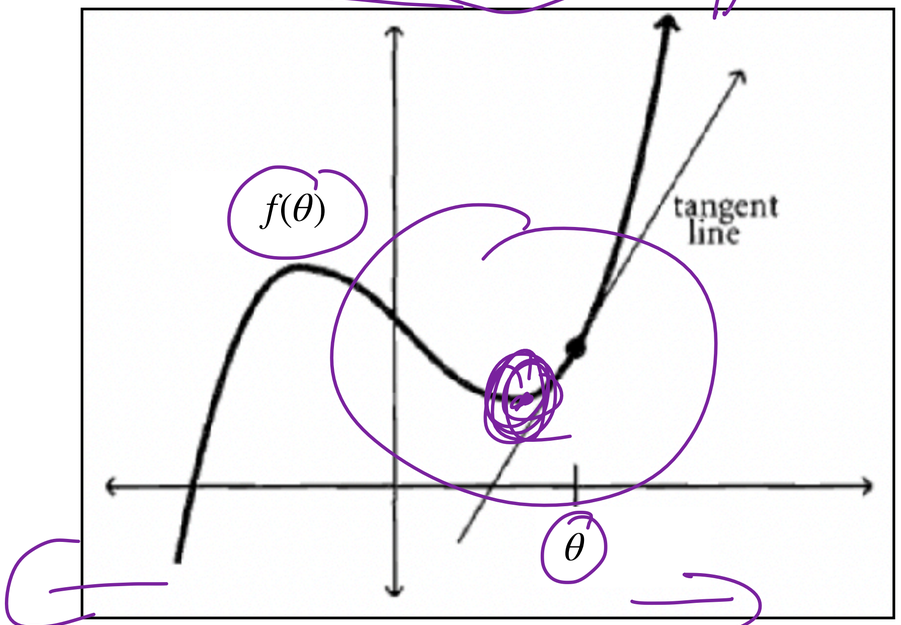


Where to search next?



Recap: Derivatives



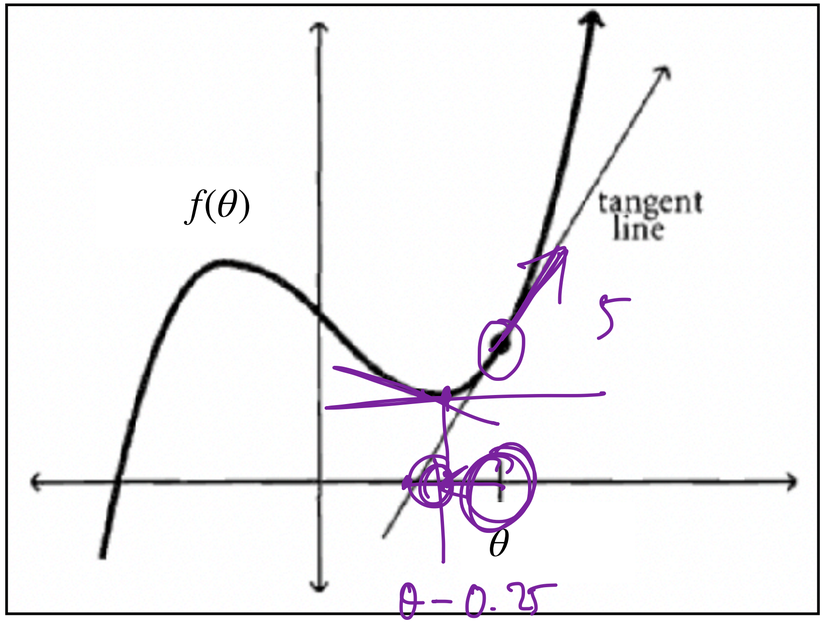


Gradient Descent Algorithm

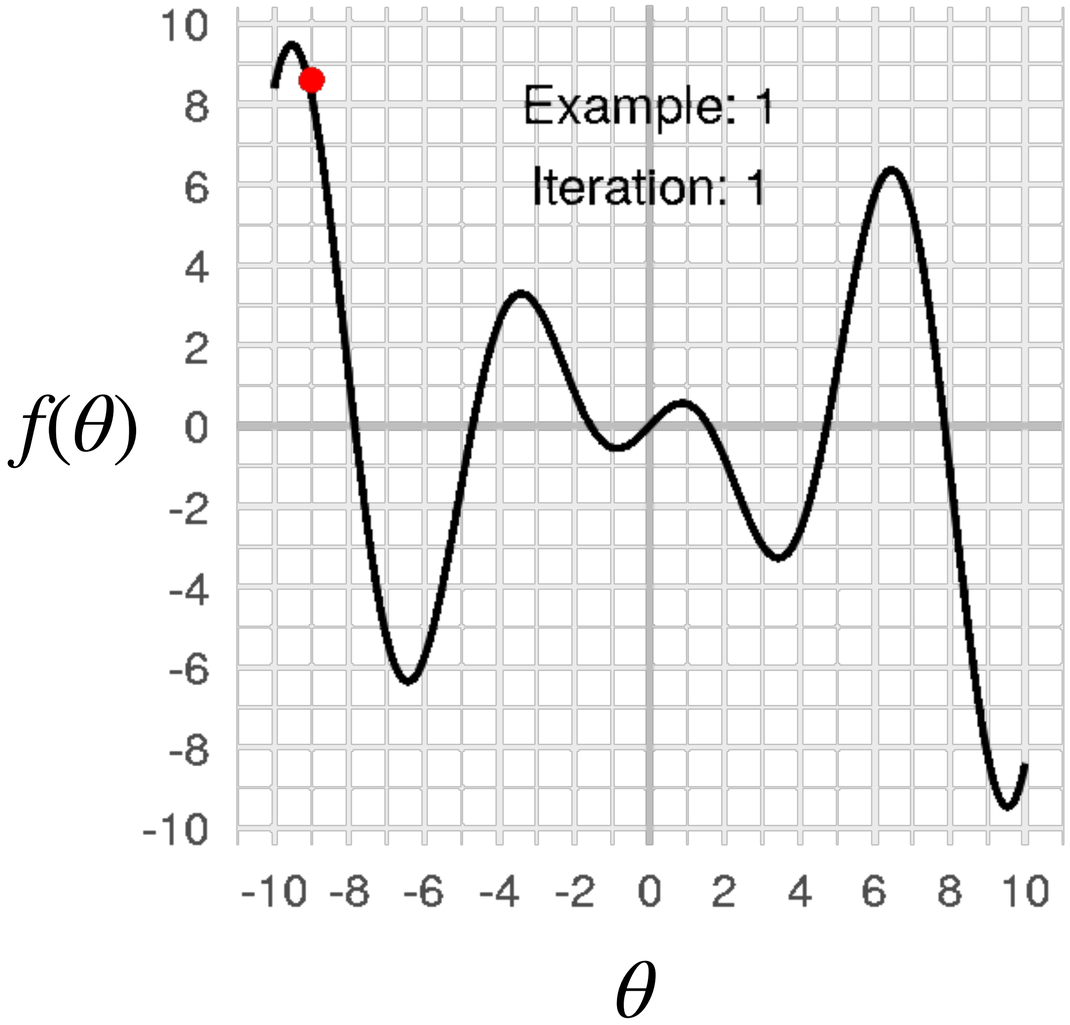
- Given: cost / loss/ objective function $f(\theta)$.
- Goal: find θ^* such that $f(\theta^*) = \min_{\theta} f(\theta)$.

Gradient Descent Algorithm

- Given: cost / loss/ objective function $f(\theta)$.
- Goal: find θ^* such that $f(\theta^*) = \min_{\theta} f(\theta)$.
- Gradient descent solution:
  - Start from initial guess θ^0 and learning rate α
  - Update $\theta^{i+1} \leftarrow \theta^i - \alpha \frac{df(\theta)}{d\theta}$
    - Repeat until change in θ is small, or maximum number of steps reached.



Gradient Descent Algorithm

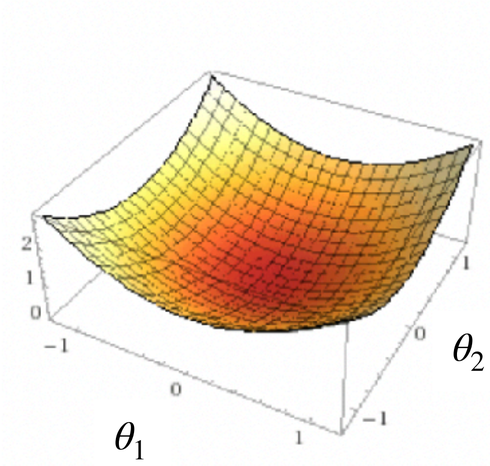


Credits: Charles Bordet

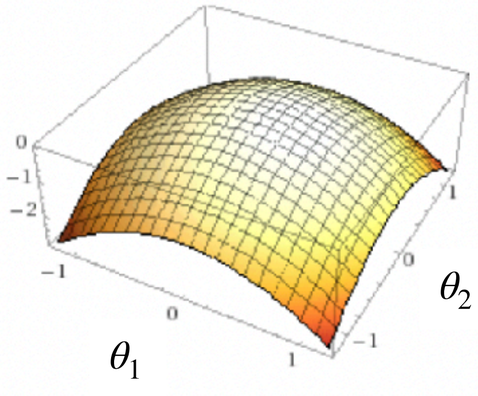
Interactive demo:

https://uclaacm.github.io/gradient-descent-visualiser/

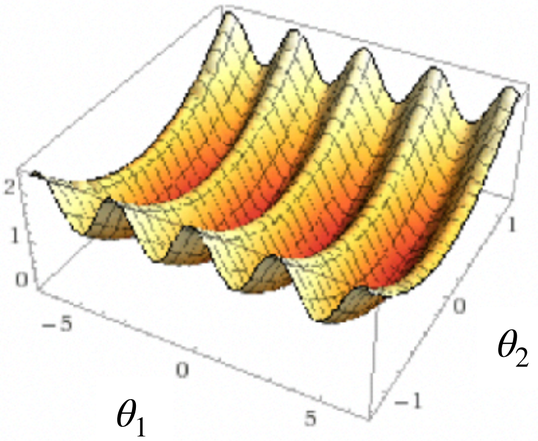
What happens when number of dimensions is high?



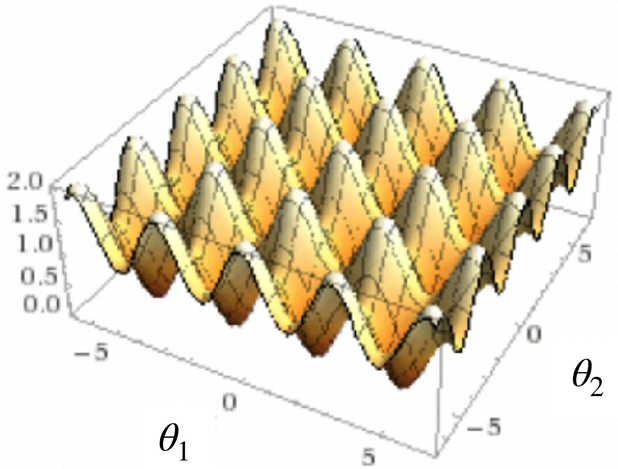
$f(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2$



$f(\theta_1, \theta_2) = -\theta_1^2 - \theta_2^2$



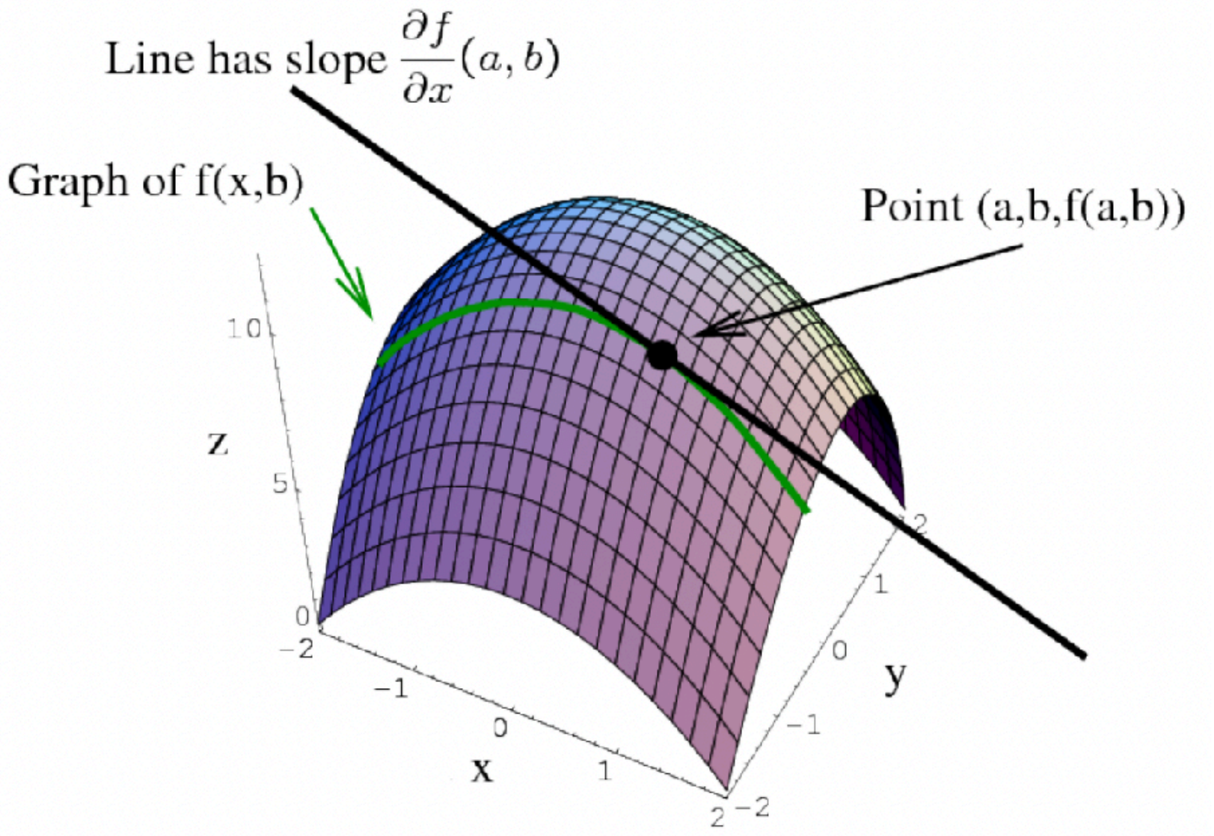
$f(\theta_1, \theta_2) = cos(\theta_1)^2 + \theta_2^2$



$f(\theta_1, \theta_2) = cos(\theta_1)^2 + cos(\theta_2)^2$

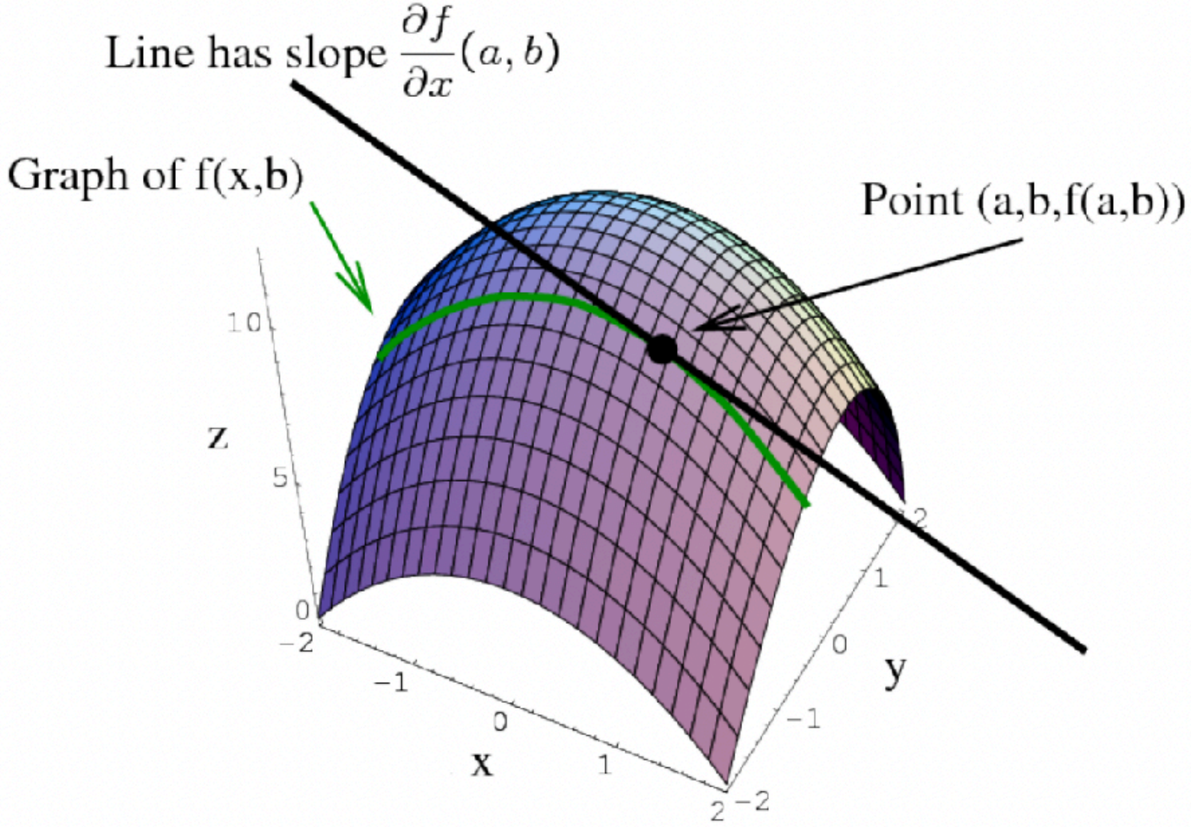
Credits: Michail Michailidis & Patrick Maiden

What happens when number of dimensions is high?



Credits: Michail Michailidis & Patrick Maiden

What happens when number of dimensions is high?



$$f(x,y) = 9 - x^2 - y^2$$

$$f(x,y) = 9 - x^2 - c^2 \qquad f(x,y) = 9 - c^2 - y^2$$

$$\frac{\partial f(x,y)}{\partial x} = -2x \qquad \frac{\partial f(x,y)}{\partial y} = -2y$$

Gradient vector:

$$\nabla f(x,y) = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial x})$$

$$\nabla f(x,y) = (-2x, -2y)$$

Credits: Michail Michailidis & Patrick Maiden

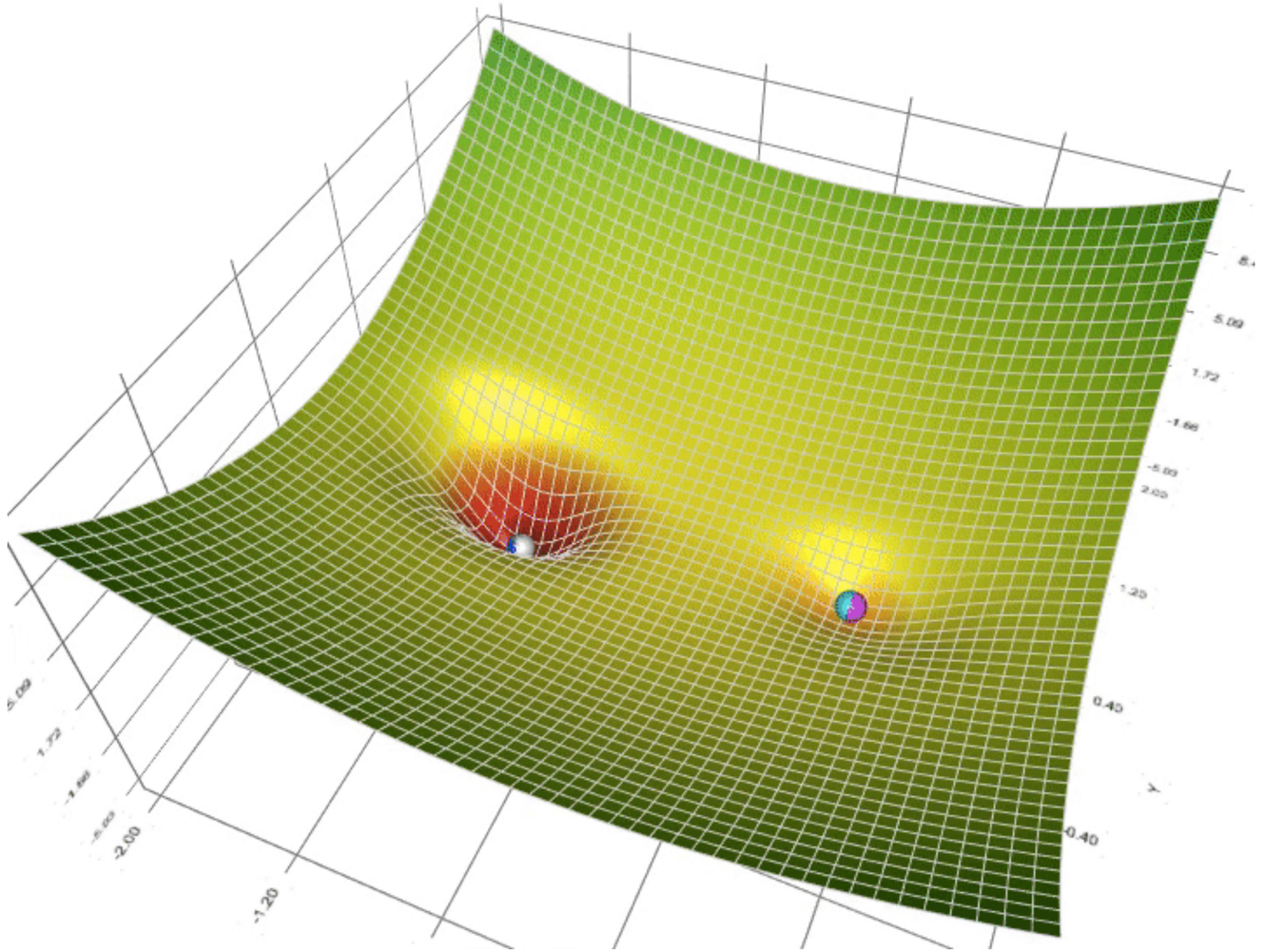
Gradient Descent Algorithm with multiple params

- Given: cost / loss/ objective function $f(\vec{\theta})$. Where $\vec{\theta} \in \mathbb{R}^d$.
- Goal: find $\vec{\theta}*$ such that $f(\vec{\theta}*) = \min_{\vec{\theta}} f(\vec{\theta})$.

Gradient Descent Algorithm with multiple params

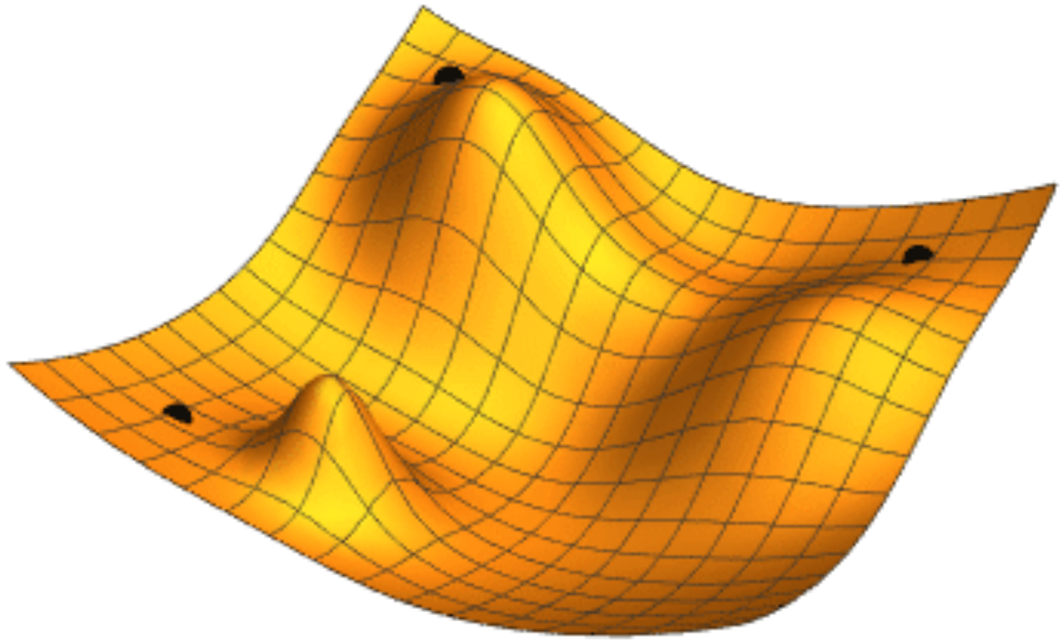
- Given: cost / loss/ objective function $f(\vec{\theta})$. Where $\vec{\theta} \in \mathbb{R}^d$.
- Goal: find $\vec{\theta}*$ such that $f(\vec{\theta}*) = \min_{\vec{\theta}} f(\vec{\theta})$.
- Gradient descent solution:
  - Start from initial guess $\vec{\theta}^0$ and learning rate α
  - Update $\vec{\theta}^{i+1} \leftarrow \vec{\theta}^i - \alpha \nabla f(\vec{\theta})$
  - Repeat until change in θ is small, or maximum number of steps reached.

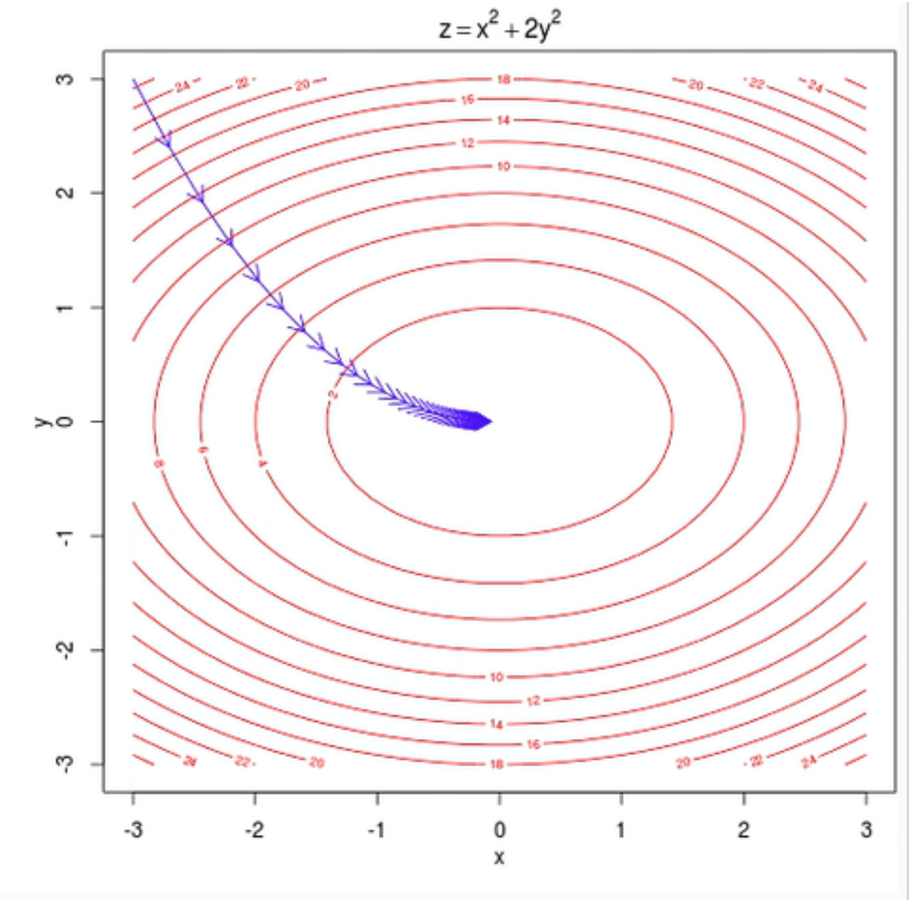
Gradient Descent Algorithm Visualization



Credits: Lili Jiang

Gradient Descent Algorithm Visualization





Credits: Wikimedia, Hoang Duong

Additional Reading

- Cross entropy method: https://people.smp.uq.edu.au/DirkKroese/ps/aortut.pdf
- Intro to Optimization: http://www.lewissoft.com/pdf/INTRO_OPT.pdf
- Linear Algebra Blog from Gregory Gundersen (https://gregorygundersen.com/blog/tags/la/)
- Mathematics for Machine Learning e-book by Deisenroth, Faisal, Ong (https://mml-book.github.io/book/mml-book.pdf)

Additional Reading

- Interactive tutorial: https://uclaacm.github.io/gradient-descent-visualiser/
- Book chapter: https://www.cs.utah.edu/~jeffp/IDABook/T6-GD.pdf
- SGD + variants: https://ruder.io/optimizing-gradient-descent

Questions?