Applied Optimization

Alexander G. Ororbia II

Introduction to Machine Learning

CSCI-635

9/11/2023

Optimization and Decision-Making Problems

1. Get a precise definition of problem, all relevant data and information on it
    - Uncontrollable factors (“random” variables)
    - Controllable inputs (“decision” variables)

2. Construct a mathematical (optimization) model of problem
    - Build objective functions and constraints

3. Solve model
    - Apply most appropriate algorithms for given problem

4. Implement solution

Mathematical Optimization in the "Real World"

Mathematical Optimization is a branch of applied mathematics which is useful in many different fields. Here are a few examples:

- Manufacturing
- Production
- Inventory control
- Transportation
- Scheduling
- Networks
- Finance
- Engineering
- Mechanics
- Economics
- Control engineering
- Marketing
- Policy Modeling

Optimization Vocabulary

Your basic optimization problem consists of…

- The objective function, f(x), which is the output you're trying to maximize or minimize.

Optimization Vocabulary

Your basic optimization problem consists of…

- The objective function, $f(x)$, which is the output you're trying to maximize or minimize.
- Variables, x_1 x_2 x_3 and so on, which are the inputs – things you can control. They are abbreviated x_n to refer to individuals or x to refer to them as a group.

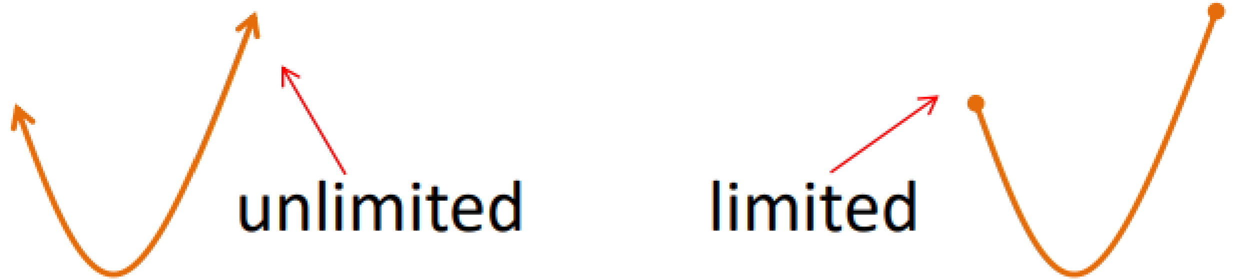
Optimization Vocabulary

Your basic optimization problem consists of…

- The objective function, $f(x)$, which is the output you're trying to maximize or minimize.
- Variables, x_1 x_2 x_3 and so on, which are the inputs – things you can control. They are abbreviated x_n to refer to individuals or x to refer to them as a group.
- Constraints, which are equations that place limits on how big or small some variables can get. Equality constraints are usually noted $h_n(x)$ and inequality constraints are noted $g_n(x)$.

Types of Optimization Problems

- Some problems have constraints and some do not.



Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.

x_1 x_3

x_4 x_2

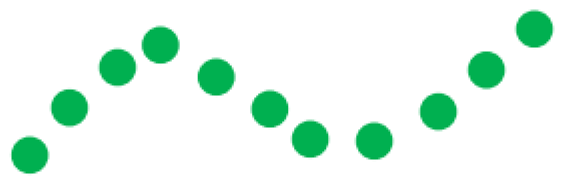
x_6 x_5

x_7

x_8

Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.



Types of Optimization Problems

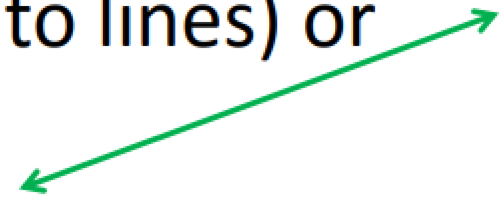
- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.
- Some problems are static (do not change over time) while some are dynamic (continual adjustments must be made as changes occur).

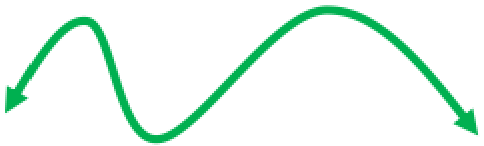
Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.
- Some problems are static (do not change over time) while some are dynamic (continual adjustments must be made as changes occur).
- Systems can be deterministic (specific causes produce specific effects) or stochastic (involve randomness/ probability).

Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.
- Some problems are static (do not change over time) while some are dynamic (continual adjustments must be made as changes occur).
- Systems can be deterministic (specific causes produce specific effects) or stochastic (involve randomness/ probability).
- Equations can be linear (graph to lines) or nonlinear (graph to curves)





Gradient-Based Optimization

- Most ML algorithms involve optimization
- Minimize/maximize a function $f(\boldsymbol{x})$ by altering $\boldsymbol{x}$
  - Usually stated a minimization
  - Maximization accomplished by minimizing $-f(\boldsymbol{x})$
- $f(\boldsymbol{x})$ referred to as objective function or criterion
  - In minimization also referred to as loss function cost, or error
  - Example is linear least squares $f(x) = \frac{1}{2} || Ax - b ||^2$
  - Denote optimum value by $\boldsymbol{x}^* = \operatorname{argmin} f(\boldsymbol{x})$

Calculus in Optimization

- Suppose we have function $y=f(x)$, x, y real nos.
  - Derivative of function denoted: $f'(x)$ or as dy/dx
    - Derivative $f'(x)$ gives the slope of $f(x)$ at point x
    - It specifies how to scale a small change in input to obtain a corresponding change in the output:

$$f(x+\varepsilon) \approx f(x) + \varepsilon f'(x)$$

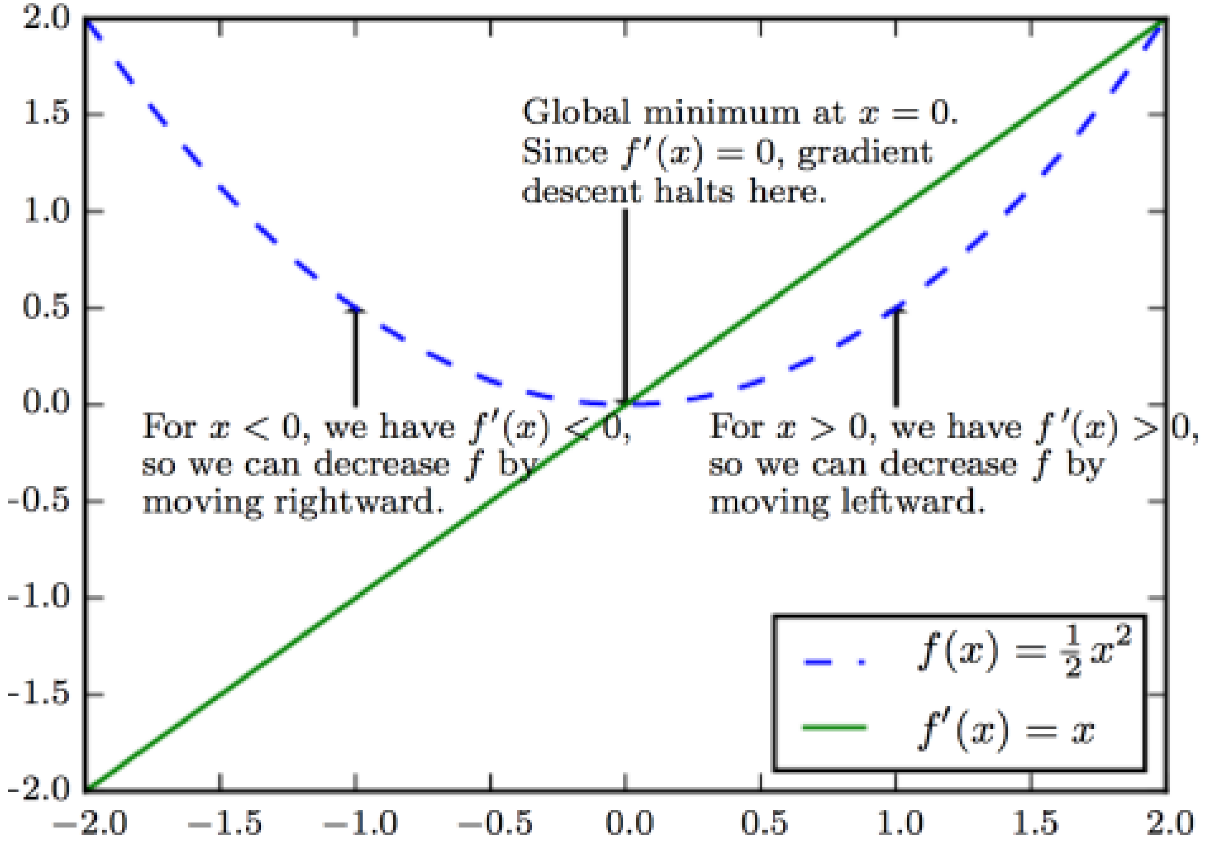
  - It tells how you make a small change in input to make a small improvement in y

signum(v)

  - We know that $f(x - \varepsilon \, \text{sign}(f'(x)))$ is less than $f(x)$ for small ε. Thus we can reduce $f(x)$ by moving x in small steps with opposite sign of derivative
    - This technique is called *gradient descent* (Cauchy 1847)

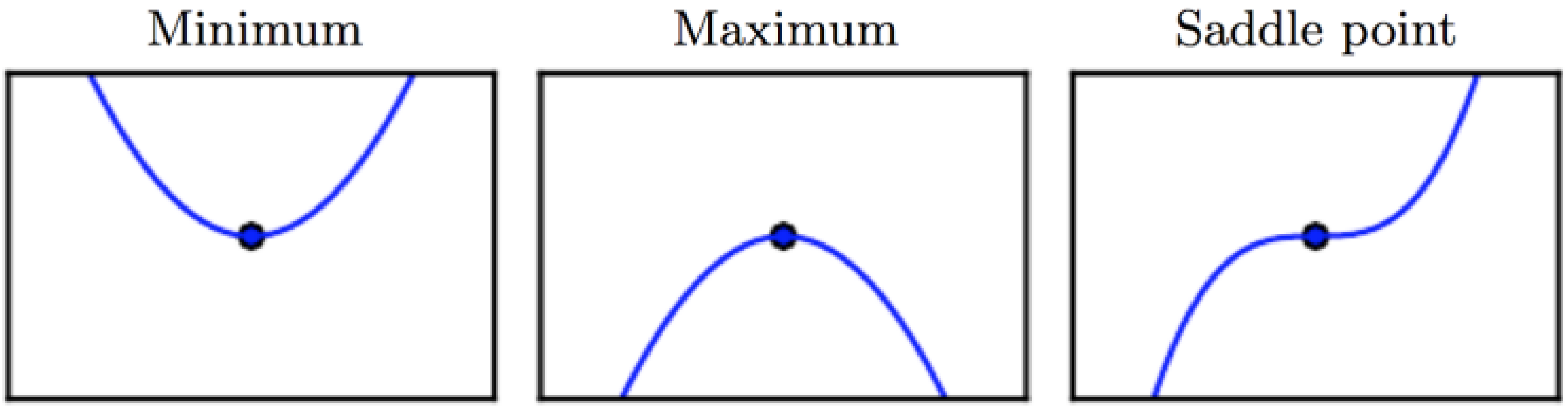
Gradient Descent Illustrated

- For $x>0$, $f(x)$ increases with x and $f'(x)>0$
- For $x<0$, $f(x)$ is decreases with x and $f'(x)<0$
- Use $f'(x)$ to follow function downhill
- Reduce $f(x)$ by going in direction opposite sign of derivative $f'(x)$



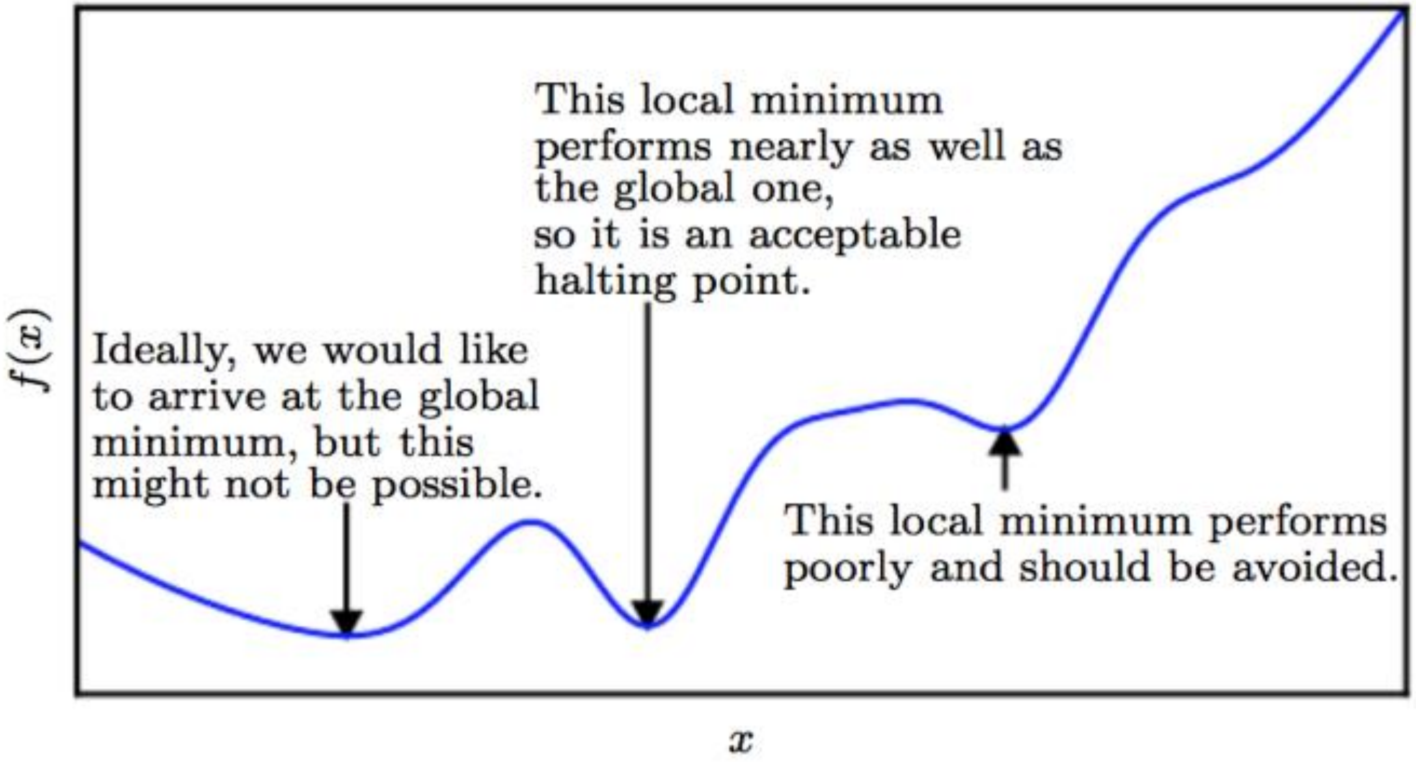
Stationary points, Local Optima

- When $f'(x)=0$ derivative provides no information about direction of move
- Points where $f'(x)=0$ are known as *stationary* or critical points
  - Local minimum/maximum: a point where $f(x)$ lower/ higher than all its neighbors
  - Saddle Points: neither maxima nor minima



Presence of Multiple Optima (Minima)

- Optimization algorithms may fail to find global minimum
- Generally accept such solutions



Minimizing with Multiple Inputs

- We often minimize functions with multiple inputs: $f: R^n \rightarrow R$
- For minimization to make sense there must still be only one (scalar) output

Functions with Multiple Inputs

- Need partial derivatives
- $\frac{\partial}{\partial x_i} f(\boldsymbol{x})$ measures how f changes as only variable x_i increases at point $\boldsymbol{x}$
- Gradient generalizes notion of derivative where derivative is wrt a vector
- Gradient is vector containing all of the partial derivatives denoted $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$
  - Element i of the gradient is the partial derivative of f wrt x_i
  - Critical points are where every element of the gradient is equal to zero

w.r.t. (with respect to)

Method of Gradient Descent

- The gradient points directly uphill, and the negative gradient points directly downhill
- Thus we can decrease f by moving in the direction of the negative gradient
  - This is known as the method of steepest descent or gradient descent
- Steepest descent proposes a new point

$$\boldsymbol{x}' = \boldsymbol{x} - \varepsilon \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$$

  - where ε is the learning rate, a positive scalar. Set to a small constant.

Choosing ε: Line Search

- We can choose ε in several different ways
- Popular approach: set ε to a small constant
- Another approach is called *line search*:
- Evaluate $f(\boldsymbol{x} - \varepsilon \nabla_{\boldsymbol{x}} f(\boldsymbol{x}))$ for several values of ε and choose the one that results in smallest objective function value

Ex: Gradient Descent on Least Squares

- Criterion to minimize $f(\mathrm{x}) = \frac{1}{2} || A\mathrm{x} - \mathrm{b} ||^2$ ← Evaluation
  - Least squares regression $E_D(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathrm{w}^T\phi(x_n)\}^2$
- The gradient is

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = A^T(A\boldsymbol{x} - b) = A^T A\boldsymbol{x} - A^T\boldsymbol{b}$$

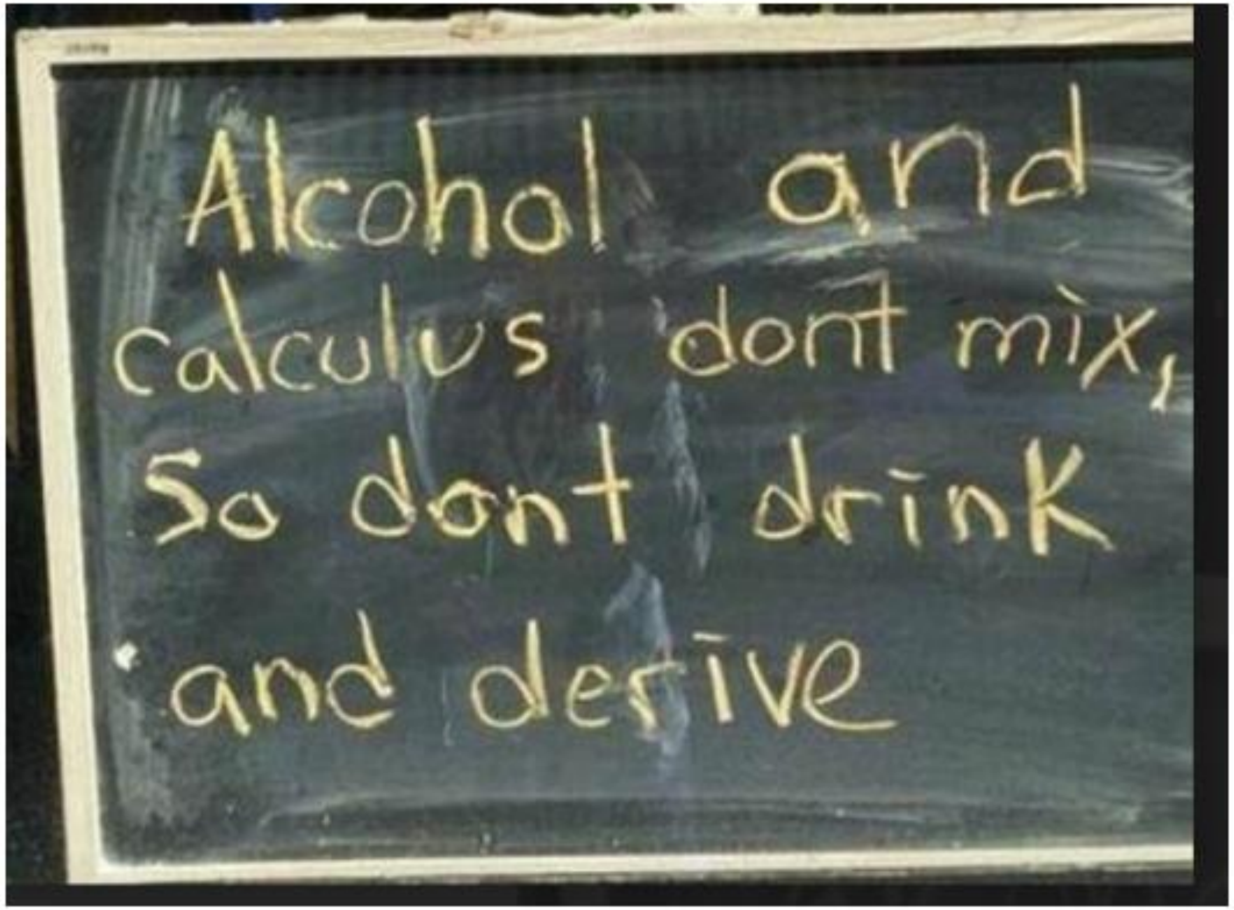
← Optimization

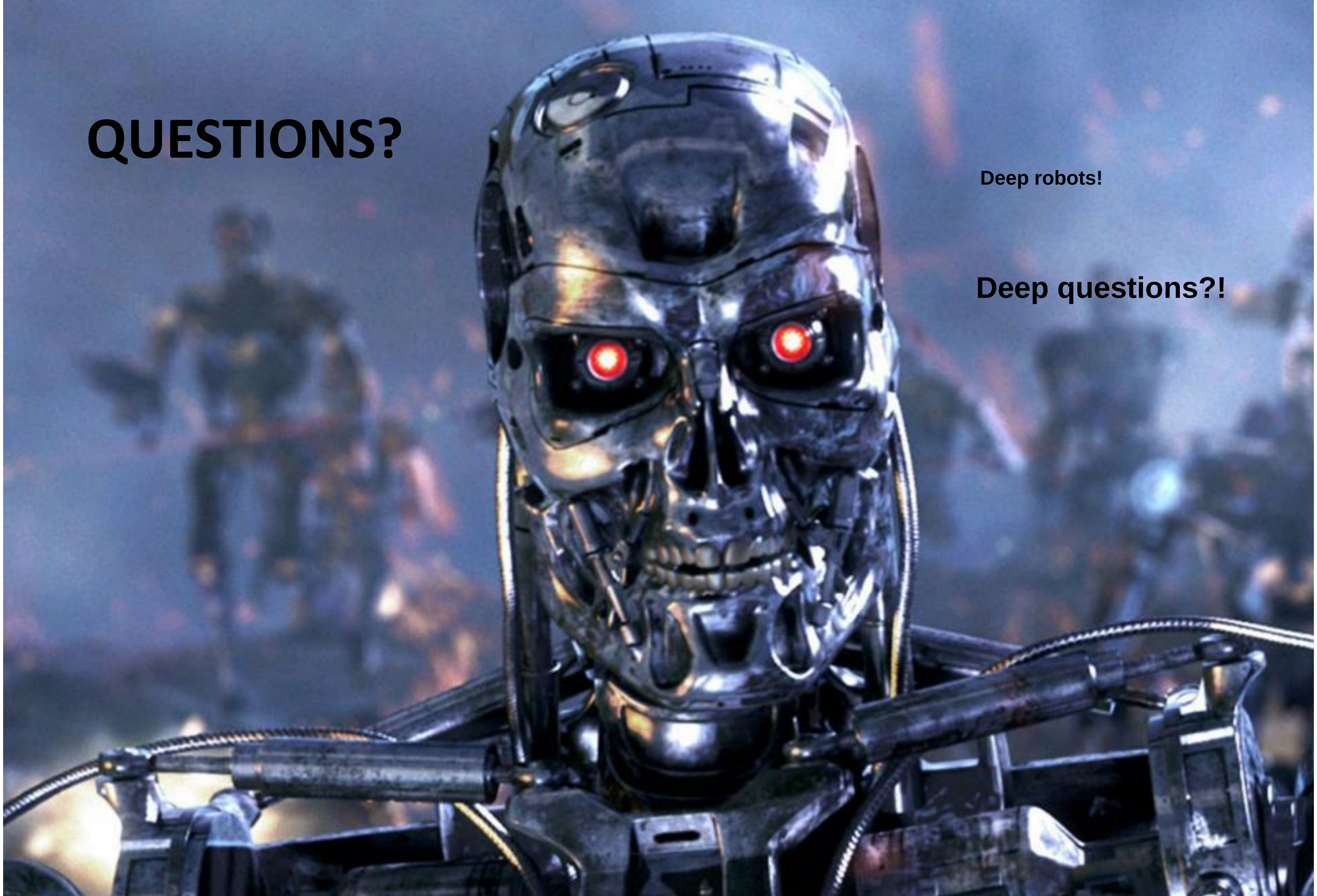
- Gradient Descent algorithm is
  1. Set step size ε, tolerance δ to small, positive nos.
  2. *while* $|| A^T A\boldsymbol{x} - A^T\boldsymbol{b} ||_2 > \delta$ do

     $$\boldsymbol{x} \leftarrow \boldsymbol{x} - \varepsilon(A^T A\boldsymbol{x} - A^T\boldsymbol{b})$$
  3. *end while*

Gradient

- Essential role of calculus




QUESTIONS?
Deep robots!
Deep questions?!