CM50264 Machine Learning 1, Lecture 5

Optimisation Basics 1

Xi Chen

UNIVERSITY OF BATH

What is optimisation

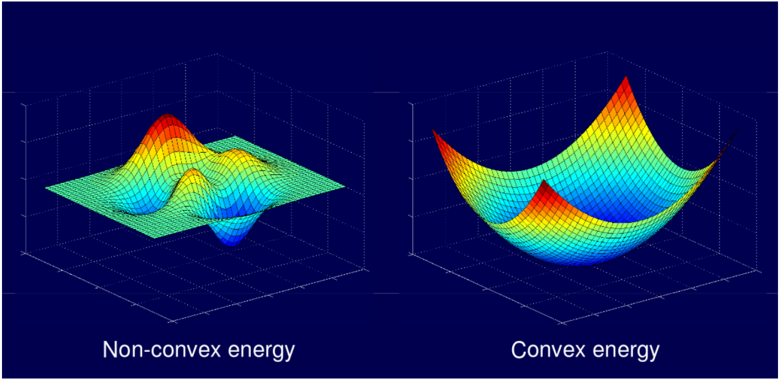


photo credit: Mathworks

Why is optimisation

Many (most) machine learning problems are eventually formulated as optimisation:

- Training decision trees.
- Training linear (or nonlinear) regression.
- Discovering cluster structure.
- Training neural networks.

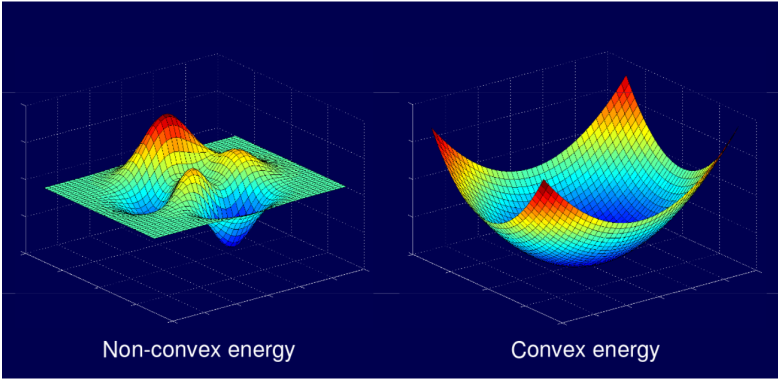
> *"Machine learning is statistics combined with optimisation."*
>
> — *Gunnar Rätsch*

Real world problems

- Real world problems are mostly non-convex.
- We don't really have an elegant approach to deal with all non-convex cases.
- Mathematics is much better established for convex cases.
- Perform convex within non-convex.

Real world problems - revisit

- Convex parts within non-convex.



Different ways to obtain solutions

- Closed form solution

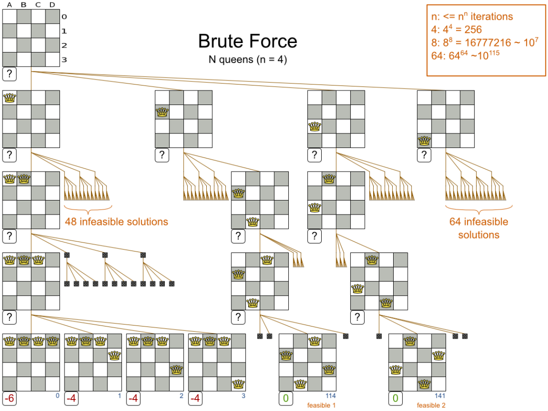
Different ways to obtain solutions

- Closed form solution
- But if it is computationally expensive or if there's no analytical solution?

Different ways to obtain solutions

- Naive exhaustive search (brute force)

Naive exhaustive search



It creates and evaluates every possible solution.

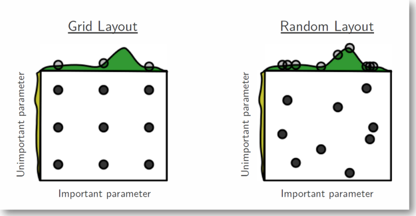
photo credit: jboss.org

Different ways to obtain solutions

- Naive exhaustive search (brute force)
- Random search and LHC

Random search and LHC

- Random search v.s. Grid search

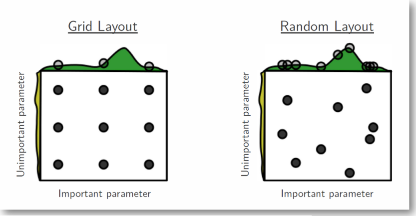


---

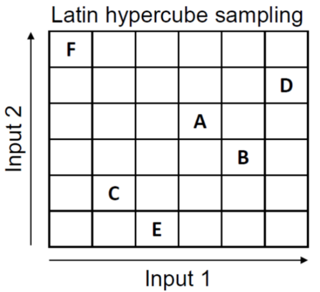
photos from Wikipedia

Random search and LHC

- Random search v.s. Grid search



- Latin hypercube sampling for some higher dimensional spaces



photos from Wikipedia

Different ways to obtain solutions

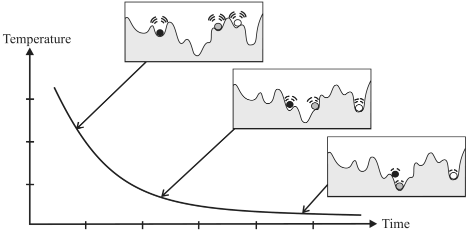
- Naive exhaustive search (brute force)
- Random search
- Numerical approximations / meta-heuristic methods

Numerical approximations / meta-heuristic methods

- Simulated annealing
- Genetic / particle swarm / ant colony algorithms
- Markov chain Monte Carlo / Sequential Monte Carlo / Nested sampling etc

Numerical approximations / meta-heuristic methods

- Simulated annealing



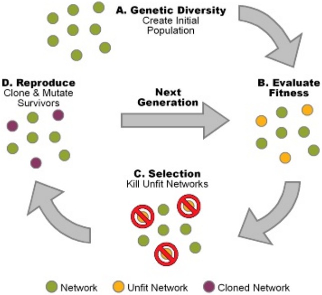
Numerical approximations / meta-heuristic methods

- Particle swarm algorithm

Numerical approximations / meta-heuristic methods

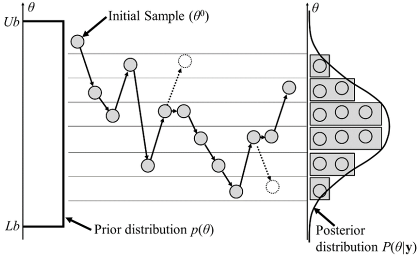
- Particle swarm algorithm

- Genetic algorithm



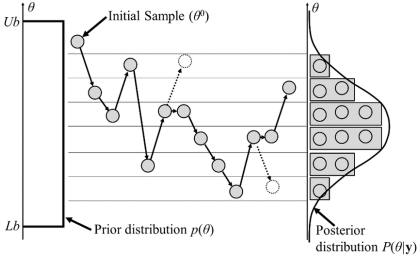
Numerical approximations / meta-heuristic methods

- MCMC

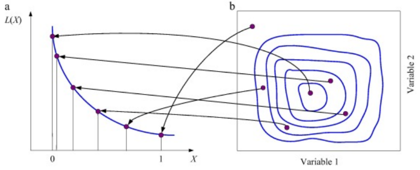


Numerical approximations / meta-heuristic methods

- MCMC



- Nested sampling



Different ways to obtain solutions

- Naive exhaustive search (brute force)
- Random search
- Numerical approximations / meta-heuristic methods
- Gradient based direction search
    - 1st order derivatives - gradient - gradient decent
    - 2nd order derivatives, Hessian - Newton's method

Structure of the Optimisation Lectures

- 3 Lectures to cover the optimisation basics.

Structure of the Optimisation Lectures

- 3 Lectures to cover the optimisation basics.
- 1st lecture: introduction, simple examples, and review/prepare some mathematical concepts, including gradient and Hessian matrix.

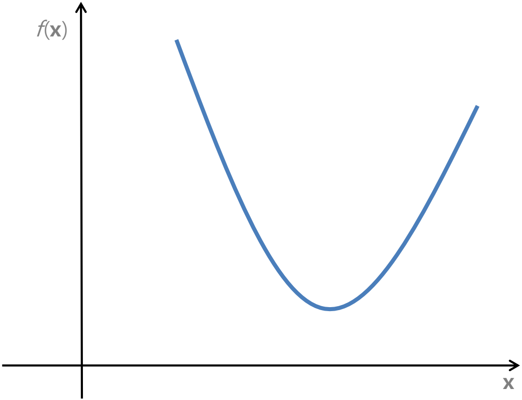
Structure of the Optimisation Lectures

- 3 Lectures to cover the optimisation basics.
- 1st lecture: introduction, simple examples, and review/prepare some mathematical concepts, including gradient and Hessian matrix.
- 2nd lecture: go through the linear regression example using analytical solution and the 1st order steepest gradient decent solution.

Structure of the Optimisation Lectures

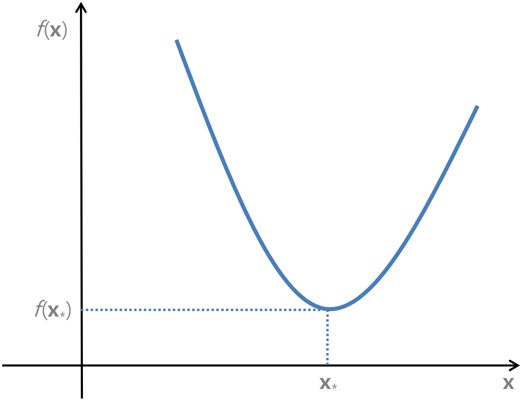
- 3 Lectures to cover the optimisation basics.
- 1st lecture: introduction, simple examples, and review/prepare some mathematical concepts, including gradient and Hessian matrix.
- 2nd lecture: go through the linear regression example using analytical solution and the 1st order steepest gradient decent solution.
- 3rd lecture: go through linear regression with the 2nd order Newton's method. Compare algorithms and introduce popular optimiser variants in machine learning.

1D example



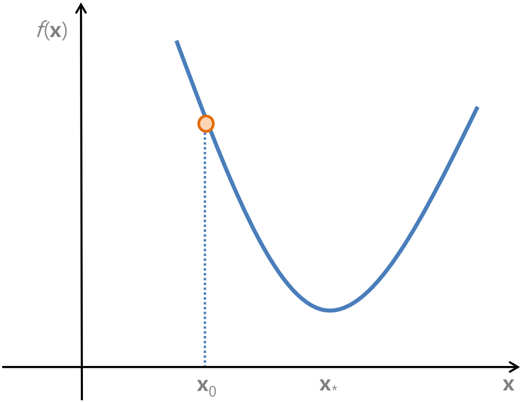
Our objective function $f(\mathbf{x})$ is one-dimensional: $f(\cdot) : \mathbb{R} \to \mathbb{R}$.

1D example



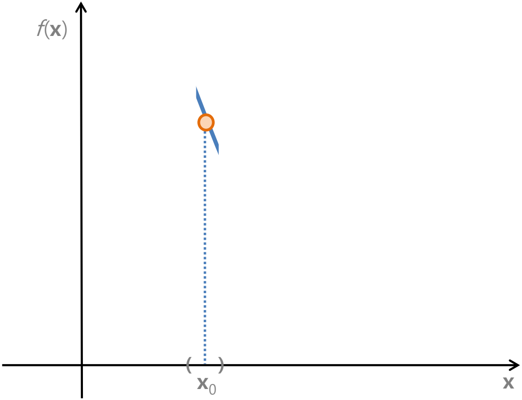
Visually inspecting the graph of $f(\cdot)$ shows that $\mathbf{x}_*$ is optimum.

1D example



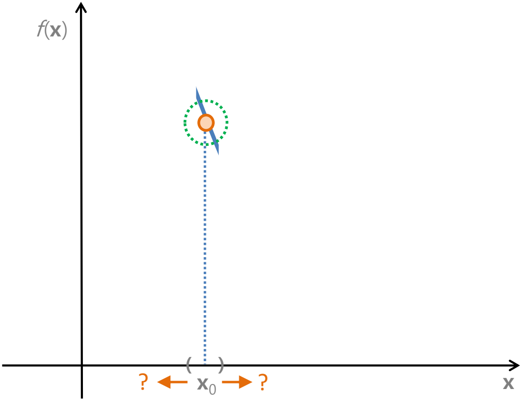
An iterative optimisation starts with an initial guess $\mathbf{x}_0$.

1D example



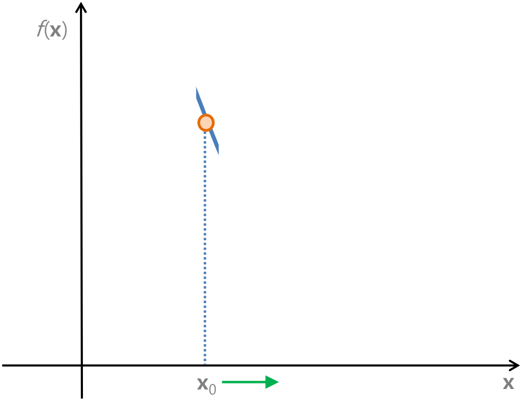
We can only observe $f(\cdot)$ in a (very small) local neighbourhood of $\mathbf{x}_0$.

1D example



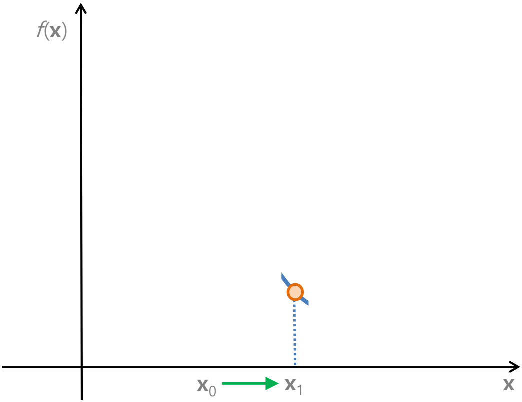
Decide direction to explore by inspecting $f(\cdot)$ values around $\mathbf{x}_0$.

1D example



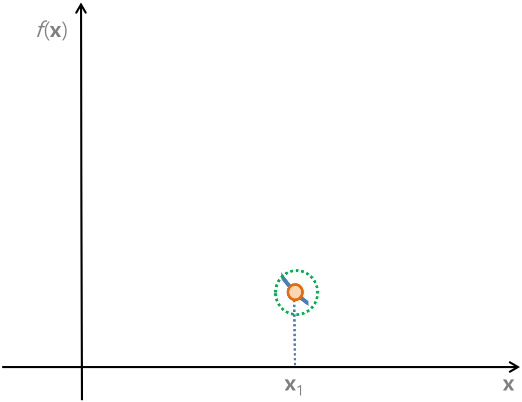
Decide direction to explore by inspecting $f(\cdot)$ values around $\mathbf{x}_0$.

1D example



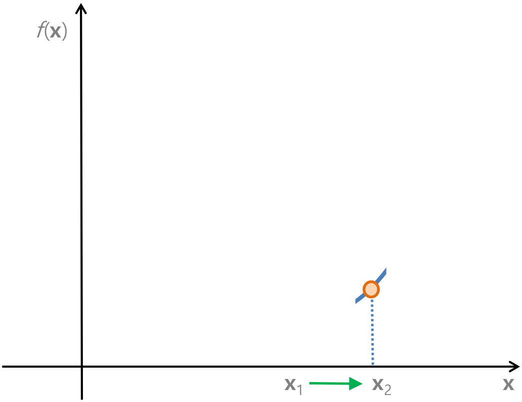
Now we are at the first solution $\mathbf{x}_1$.

1D example



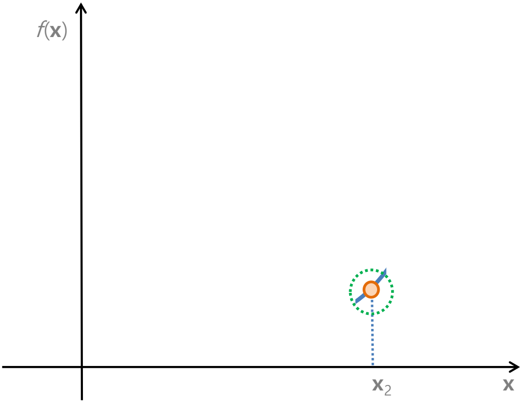
Again, decide the direction of next step at $\mathbf{x}_1$.

1D example



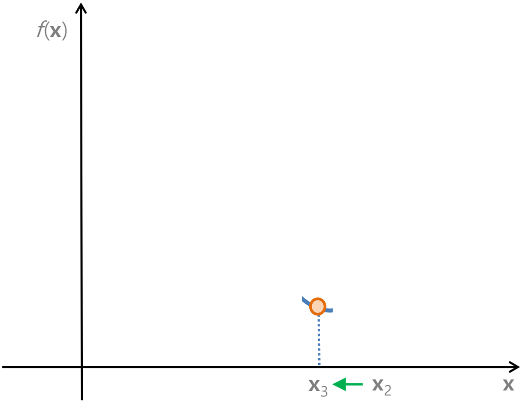
Now we are at the second solution $\mathbf{x}_2$.

1D example



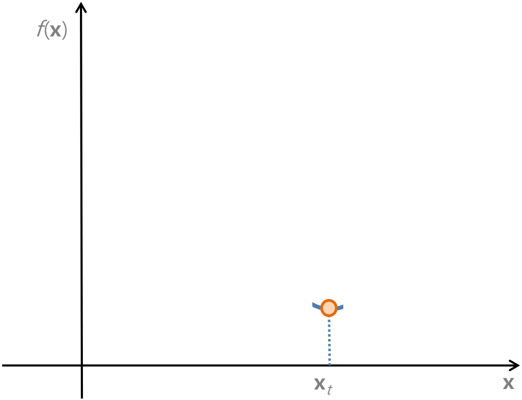
Decide the direction of next step at $\mathbf{x}_2$.

1D example



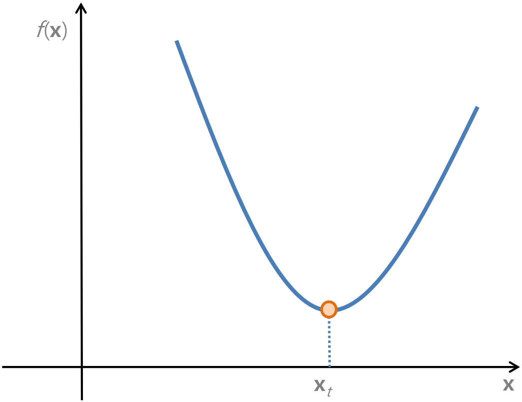
Decide the direction of next step at $\mathbf{x}_3$.

1D example



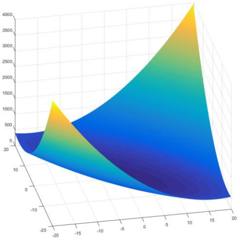
After a few iterations, stop (converge) at point $\mathbf{x}_t$.

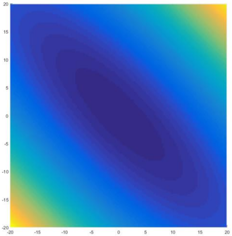
1D example



Global optimum $\mathbf{x}_t = \mathbf{x}_*$.

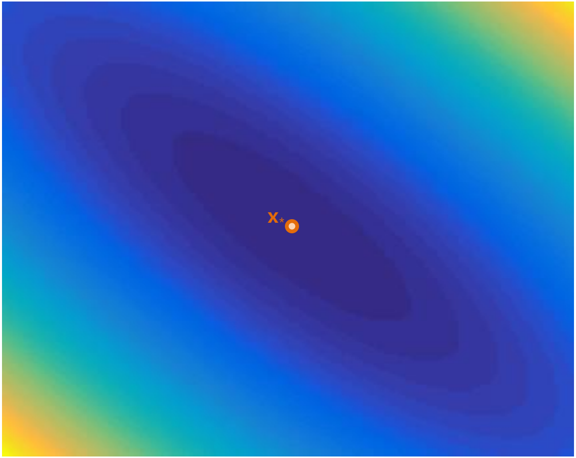
How does gradient search work? - 2D example





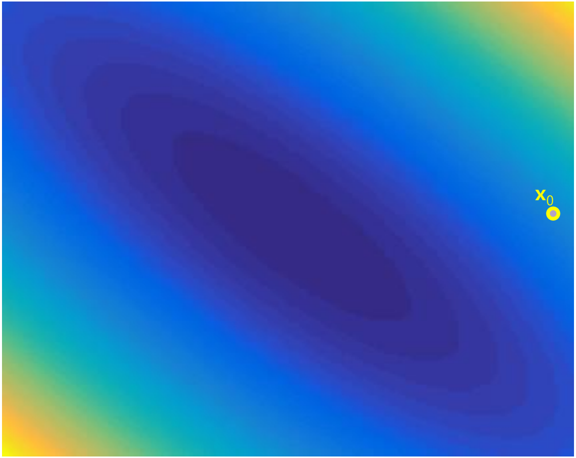
Our objective function f is two-dimensional: $f(\cdot) : \mathbb{R}^2 \to \mathbb{R}$.

2D example



The true optimum point $\mathbf{x}_*$ is at the centre of the contour.

2D example

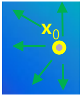


We start with an (randomly selected) initial solution $\mathbf{x}_0$.

2D example

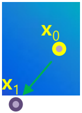
Again, we can observe $f(\cdot)$ only in a small neighbourhood of $\mathbf{x}_0$.

2D example



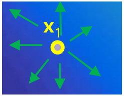
Inspecting $f(\cdot)$ around point $\mathbf{x}_0$.

2D example



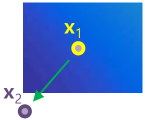
Decide a direction for the next step to decrease the $f(\cdot)$ value.

2D example



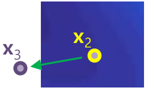
Now at the first step $\mathbf{x}_1$, and observing $f(\cdot)$ around the point.

2D example



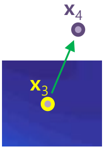
Decide a new direction.

2D example



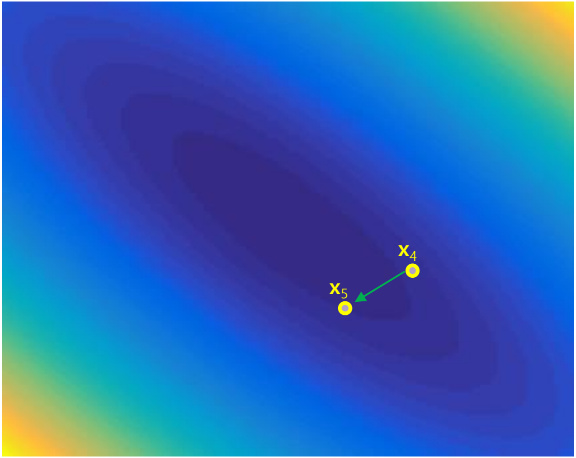
Repeat to reach the third solution $\mathbf{x}_3$.

2D example



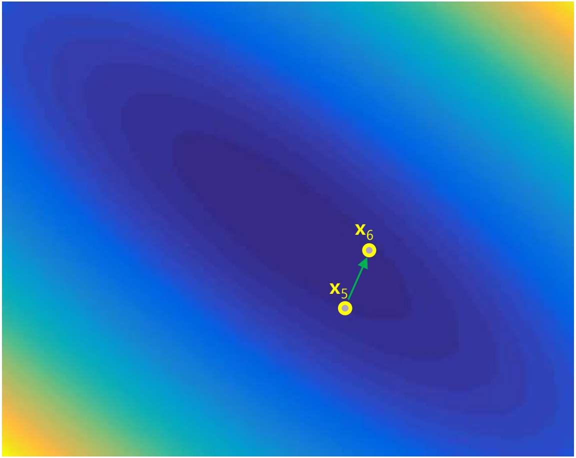
From the third solution to the fourth.

2D example



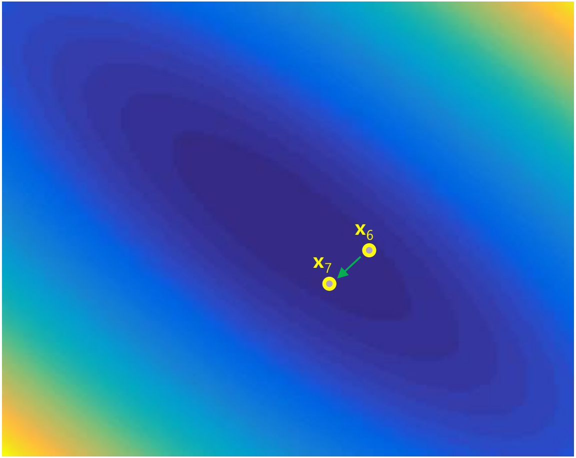
From $\mathbf{x}_4$ to $\mathbf{x}_5$

2D example



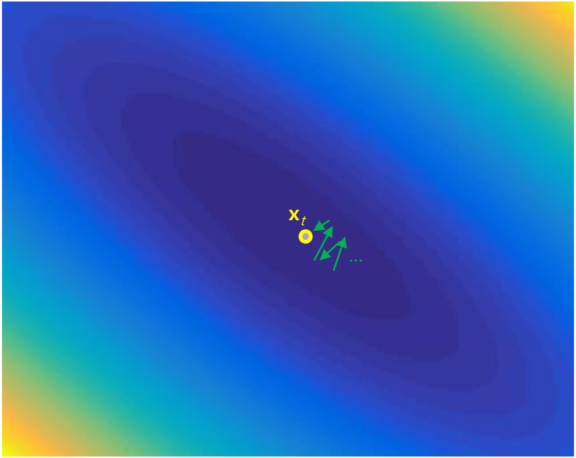
from $\mathbf{x}_5$ to $\mathbf{x}_6$

2D example



from $\mathbf{x}_6$ to $\mathbf{x}_7$

2D example



After a few iterations, we arrive at the optimum $\mathbf{x}_t = \mathbf{x}_*$.

2D example

1. $t = 0$; Make an initial guess $\mathbf{x}_t$;
2. Iterate until the termination condition is met.
   1. Find a direction $\mathbf{p}_t$ to move;
   2. Decide how much (α_t) to move along $\mathbf{p}_t$ direction;
   3. $\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha_t \mathbf{p}_t$;
   4. $t = t + 1$;

How do we decide

- direction to move $\mathbf{p}_t$,
- step size α_t,
- when to stop (termination condition)?

Some Maths review

Partial derivatives

For a function with two parameters $\mathbf{x} = [x_1, x_2]$

$$f(\mathbf{x}) = f(x_1, x_2) = x_1^2 + x_2^2$$

Some Maths review

Partial derivatives

For a function with two parameters $\mathbf{x} = [x_1, x_2]$

$$f(\mathbf{x}) = f(x_1, x_2) = x_1^2 + x_2^2$$

The partial derivatives w.r.t. x_1 and x_2:

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = \lim_{\Delta x_1 \to 0} \frac{f(x_1 + \Delta x_1, x_2) - f(x_1, x_2)}{\Delta x_1} = 2x_1$$

Some Maths review

Partial derivatives

For a function with two parameters $\mathbf{x} = [x_1, x_2]$

$$f(\mathbf{x}) = f(x_1, x_2) = x_1^2 + x_2^2$$

The partial derivatives w.r.t. x_1 and x_2:

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = \lim_{\Delta x_1 \to 0} \frac{f(x_1 + \Delta x_1, x_2) - f(x_1, x_2)}{\Delta x_1} = 2x_1$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = 2x_2$$

Some Maths review

Gradient

$$\nabla f(\mathbf{x}) = [2x_1, 2x_2]^\top$$

where ∇ is the vector differential operator.

Some Maths review

Gradient

$$\nabla f(\mathbf{x}) = [2x_1, 2x_2]^\top$$

where ∇ is the vector differential operator.

The gradient of a differentiable function $f(\cdot)$ of several variables, at a point $\mathbf{x}_P$, is the vector whose components are the partial derivatives of $f(\cdot)$ at $\mathbf{x}_P$.

Some Maths review

Gradient

$$\nabla f(\mathbf{x}) = [2x_1, 2x_2]^\top$$

where ∇ is the vector differential operator.

The gradient of a differentiable function $f(\cdot)$ of several variables, at a point $\mathbf{x}_P$, is the vector whose components are the partial derivatives of $f(\cdot)$ at $\mathbf{x}_P$.

so if at point $\mathbf{x}_P = [2, 2]$:

$$\nabla f(\mathbf{x}) = [4, 4]^\top$$

Some Maths review

Hessian matrix

It is a square matrix of second-order partial derivatives of function $f(\cdot)$

Some Maths review

Hessian matrix

It is a square matrix of second-order partial derivatives of function $f(\cdot)$

$\mathbf{H}(f(\cdot))$ is symmetric if $f(\cdot)$ is twice-continously differentiable.

Some Maths review

Hessian matrix

It is a square matrix of second-order partial derivatives of function $f(\cdot)$

$\mathbf{H}(f(\cdot))$ is symmetric if $f(\cdot)$ is twice-continously differentiable. If

we have $\mathbf{x} = [x_1, x_2, \cdots, x_n]$.

$$\mathbf{H}(f(\mathbf{x})) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Some Maths review

Extended questions

- What is the physical meaning of gradient and Hessian?

Some Maths review

Extended questions

- What is the physical meaning of gradient and Hessian?
- What is Jacobian matrix?

Some Maths review

Extended questions

- What is the physical meaning of gradient and Hessian?
- What is Jacobian matrix?
- What is the relationship between Jacobian matrix and Hessian matrix?