TC2 - Optimization for ML

CLASS 4

1/ About the EXAM :

written exam week from 14-18 December at the university. 13:30 $\longrightarrow$ 15:30 2Hours

without documents.

For the 3/4 of you who cannot be present, we will organize an oral exam.

- Gradient direction : $\nabla f(x)$
- Newton direction : $-\left[\nabla^2 f(x)\right]^{-1} \nabla f(x)$
- $f(x) = \frac{1}{2} x^T A x \qquad x \in \mathbb{R}^2, \quad A = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}$

Plot $\nabla f(x)$, $\left[\nabla^2 f(x)\right]^{-1} \nabla f(x)$ and level set of f.

$$\nabla f(x) = \begin{pmatrix} 9x_1 \\ x_2 \end{pmatrix}$$

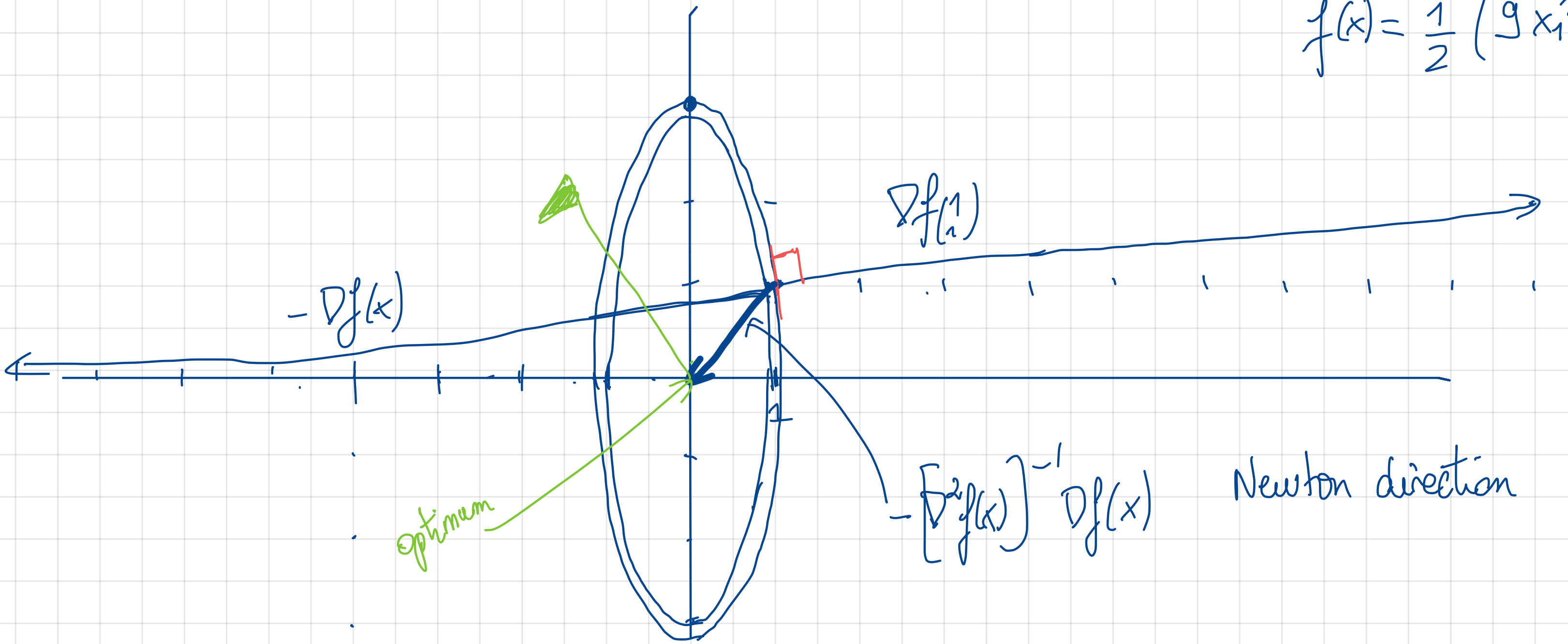
$$\nabla^2 f(x) = A = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\left[\nabla^2 f(x)\right]^{-1} = \begin{pmatrix} \frac{1}{9} & 0 \\ 0 & 1 \end{pmatrix}; \quad \left[\nabla^2 f(x)\right]^{-1} \nabla f(x) = \begin{pmatrix} \frac{1}{9} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 9x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$= x$$

Newton direction: $-x$

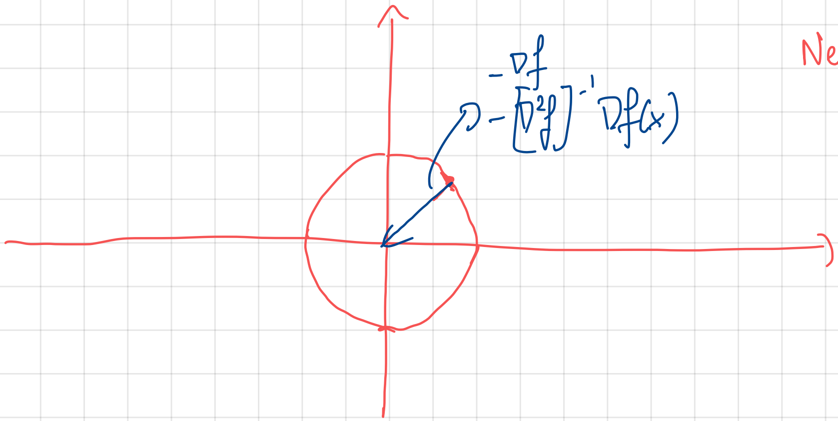
$$f(x) = \frac{1}{2}\left(9x_1^2 + x_2^2\right)$$



At $x = \begin{pmatrix} -4 \\ -5 \end{pmatrix}$ $\nabla f(x) = \begin{pmatrix} -9\cdot 4 \\ -5 \end{pmatrix}$

What if $f(x) = \frac{1}{2}\left(x_1^2 + x_2^2\right)$ $\nabla f^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

What about the Newton and $-\nabla f$ in this case?



Newton direction $= -\nabla f$

We observe that the Newton direction points towards the optimum independently of the condition number of the Hessian matrix.

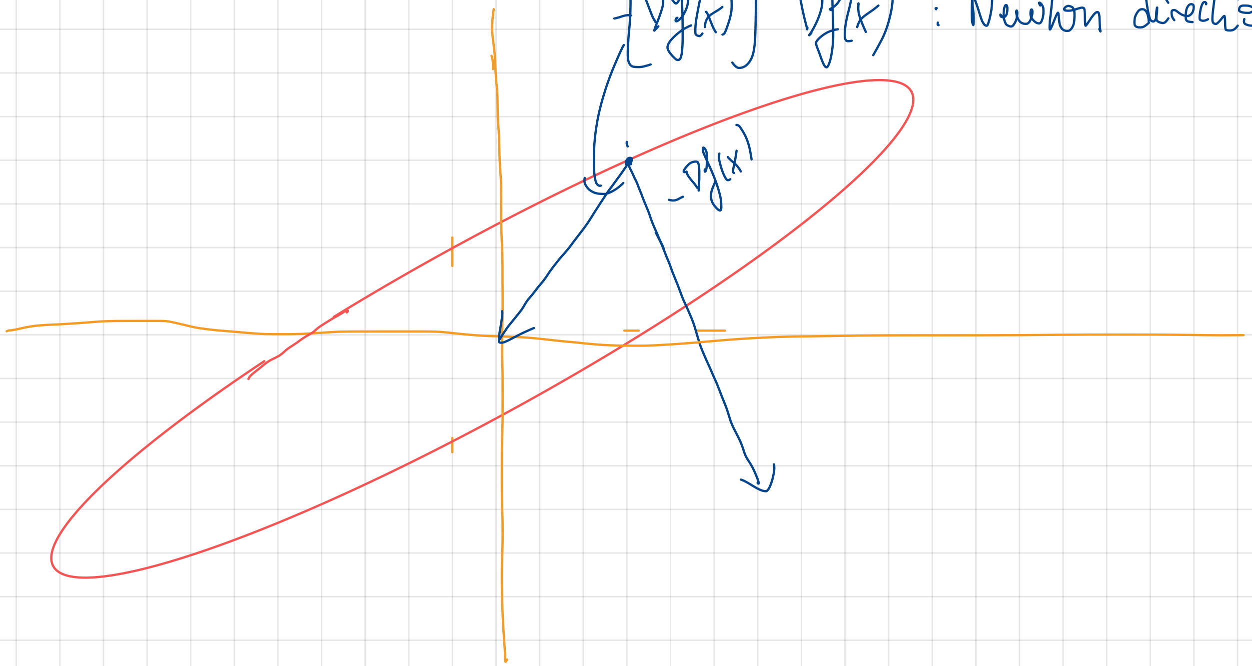
whereas $-Df(x)$ points towards the optimum at $x = \begin{pmatrix}1\\1\end{pmatrix}$ if and only if $D^2f(x) = Id$ and the condition number equal to 1.

If the Hessian matrix is not diagonal anymore: $f(x) = \frac{1}{2} x^T A x$

A symmetric positive, definite

A not diagonal

$-[D^2f(x)]^{-1} Df(x)$: Newton direction



$-Df(x)(h) = -Df(x) \cdot h$

Optimality conditions:

Assume $f : \mathbb{R} \longrightarrow \mathbb{R}$ is differentiable ($f'(x)$ exists for all x)

Which one of the following statements are correct:

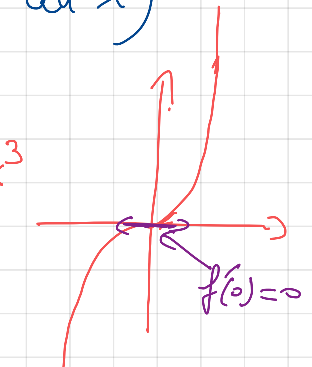
1. $f'(x^*) = 0 \Rightarrow x^*$ is a local optimum **WRONG**
2. x^* is a local optimum $\Rightarrow f'(x^*) = 0$ CORRECT
3. $f'(x^*) = 0 \Rightarrow x^*$ is a global optimum WRONG
4. x^* is a global optimum $\Rightarrow f'(x^*) = 0$ CORRECT

$f(x) = x^3$

$f'(x) = 3x^2$

$f'(0) = 0$

counter-example



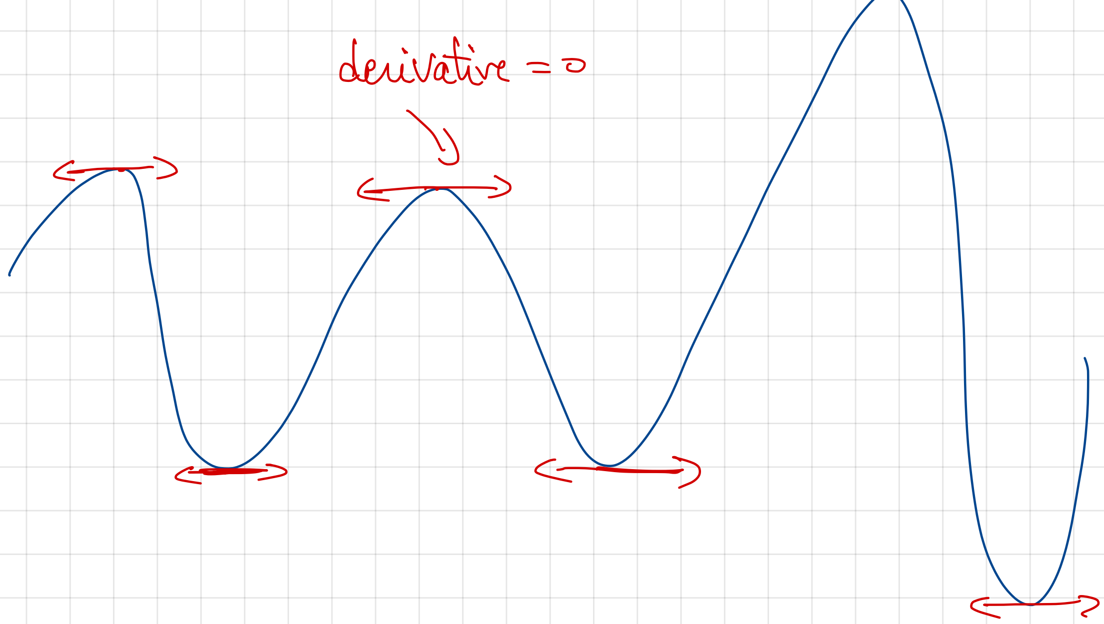
(2) gives a first order necessary condition.

THEOREM: (first order necessary condition)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. If x^* is a local optimum of f then $\nabla f(x^*) = 0$.

minimum or maximum

Interpretation when $n = 1$:



Proof for $n = 1$:

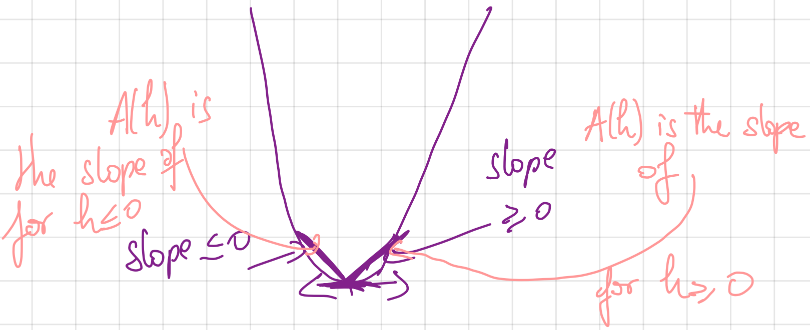
$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

assume that x^* is a local minimum : $f(x^*) \leq f(x^* + h)$ $\forall h$ small enough

$$A(h) = \frac{\overbrace{f(x^*+h) - f(x^*)}^{\geq 0}}{h}$$

$\rightarrow$ if $h \geq 0$ $\quad A(h) \geq 0$

if $h \leq 0$ $\quad A(h) \leq 0$



$$\left.\begin{aligned} &\lim_{\substack{h \to 0 \\ h \geq 0}} \underbrace{A(h)}_{\geq 0} = f'(x^*) \geq 0 \\ \text{if } &\lim_{\substack{h \to 0 \\ h \leq 0}} \underbrace{A(h)}_{\leq 0} = f'(x) \leq 0 \end{aligned}\right\} f'(x) = 0$$

SECOND ORDER NECESSARY AND SUFFICIENT CONDITIONS:

Let assume that f is twice continuously differentiable

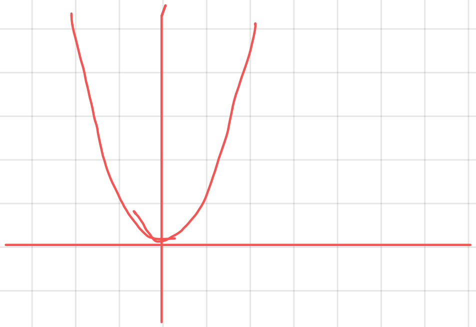
NECESSARY CONDITION: If x^* is a local minimum, then $Df(x^*) = 0$ and $D^2f(x)$ is positive semi-definite.

(if $n=1$, x^* is a local minimum $\Rightarrow f'(x^*) = 0$, $f''(x) \geq 0$)

SUFFICIENT CONDITION: If x^* which satisfies $Df(x^*) = 0$ and $D^2f(x)$ is positive definite, then x^* is a strict local minimum.

(if $n=1$, x^* such that $f'(x^*) = 0$ $f''(x) > 0$ $\Rightarrow$ x^* is a strict local minimum)

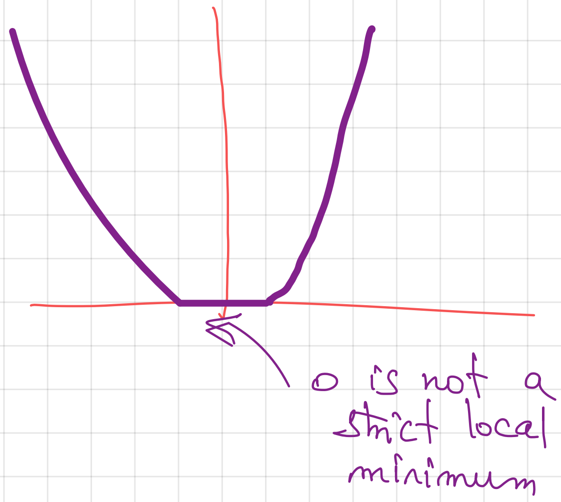
Example: $f(x) = x^2$, $f'(x) = 2x$ $f''(x) = 2$

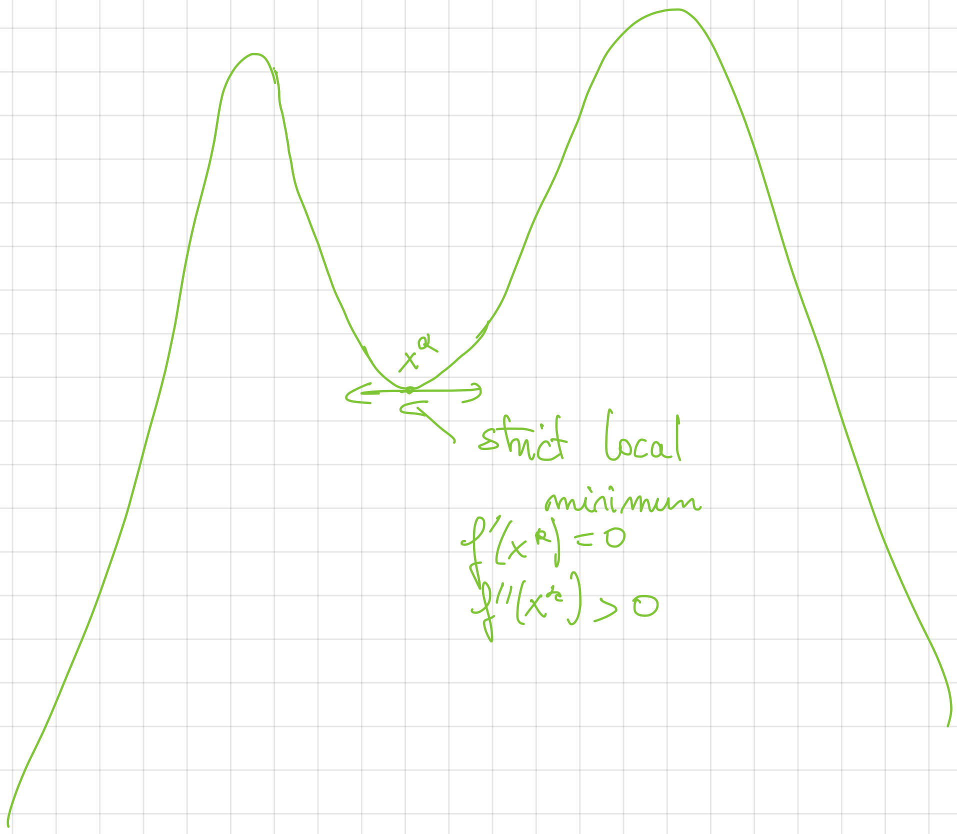


0 satisfies that $f'(0) = 2 \times 0 = 0$ and $f''(0) = 2 > 0$

$\Rightarrow$ 0 is a strict local minimum.

strict local minimum :



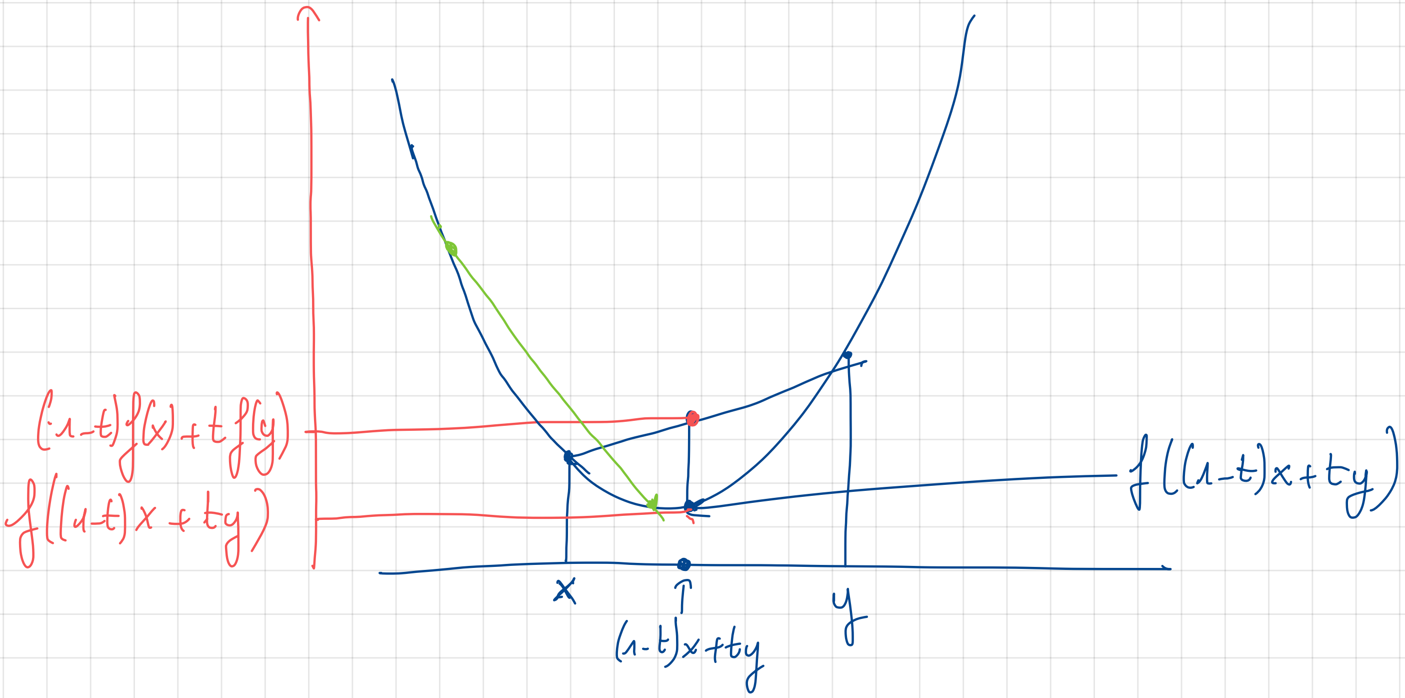


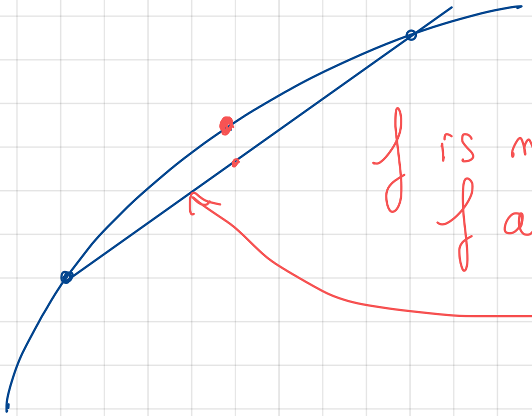
CONVEX FUNCTIONS

Let $f : U \subset \mathbb{R}^n \longrightarrow \mathbb{R}$. We say that f is convex, if for all $x, y \in U$ $\forall\, t \in [0, 1]$

open convex set

$$f((1-t)x + ty) \le (1-t) f(x) + t f(y)$$



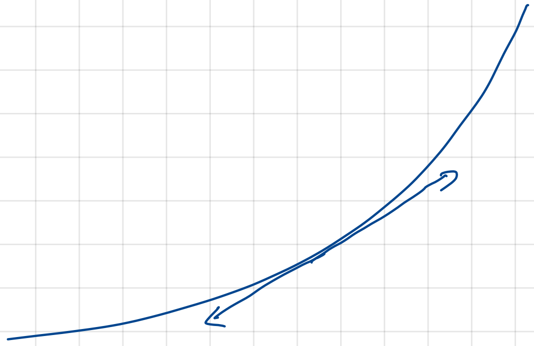


f is not convex because f above line

THEOREM: If f is differentiable, then f is convex if and only if for all x, y

$$f(y) - f(x) \geq Df(x)^T (y-x) = Df(x) \cdot (y-x)$$

If $n=1$ $\quad f(y) - f(x) \geq f'(x)(y-x)$



f is convex if and only if the function is above the tangent.

THEOREM: If f is twice continuously differentiable, then f is convex if and only if $D^2 f(x)$ is positive semi-definite for all x.

If $n=1$ f is twice derivable, then f is convex if and only if $f''(x) \geq 0$

Examples: $f(x) = x^2$ is convex (because $f''(x) = 2 \geq 0$)

$f(x) = -x^2$ ($f''(x) = -2 \to f$ is not convex)

$f(x) = \log(x)$ ($f'(x) = \frac{1}{x}$, $f''(x) = -\frac{1}{x^2} \underbrace{\leq 0}_{x>0} \to f$ is not convex)

$f(x) = x$ f is convex $f''(x) = 0$

Examples of convex functions:

- $f(x) = \frac{1}{2} x^T A x$ A sym. pos. definite.
- $f(x) = a^T x + b$ $a \in \mathbb{R}^n$, $b \in \mathbb{R}^n$
- the negative of the entropy: $f(x) = -\sum_{i=1}^{n} x_i \log(x_i)$

EXERCICE : Let $f : U \subset \mathbb{R}^n \to \mathbb{R}$ be a convex and differentiable function.

Prove that if $Df(x^*) = 0$, then x^* is a global minimum.

If f is convex and differentiable we have : $\forall x, y$

$$f(y) - f(x) \geq Df(x)^T (y - x)$$

If x^* is such that $Df(x^*) = 0$, then $f(y) - f(x^*) \geq \underbrace{Df(x^*)^T (y - x^*)}_{=0}$

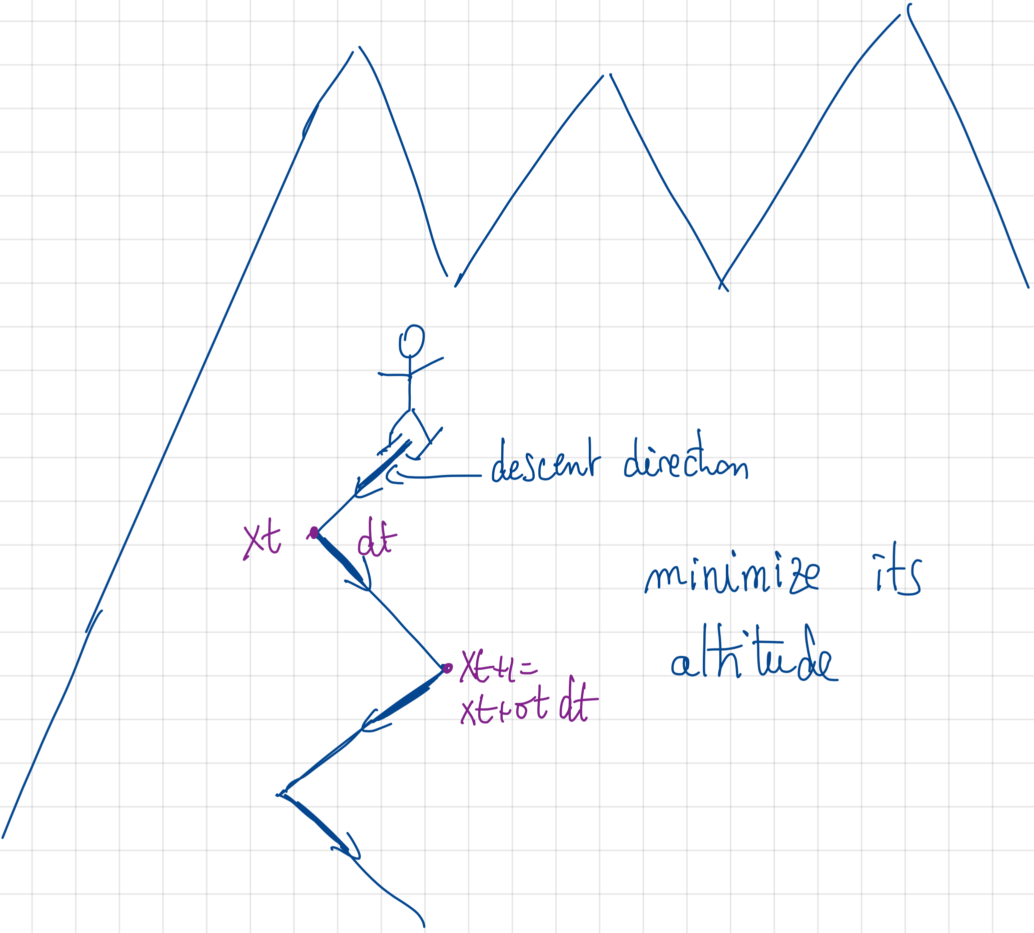
$$f(y) - f(x^*) \geq 0 \quad \forall y$$

then $\forall y \quad f(y) \geq f(x^*)$

which means that x^* is the global minimum of f.

The important consequence is that for convex differentiable functions critical points, points where $\nabla f(x) = 0$, are global minima of the functions.

DESCENT METHODS



OBJECTIVE: Minimize $f : \mathbb{R}^n \rightarrow \mathbb{R}$

General principle

1/ choose an initial point x_0, $t = 0$

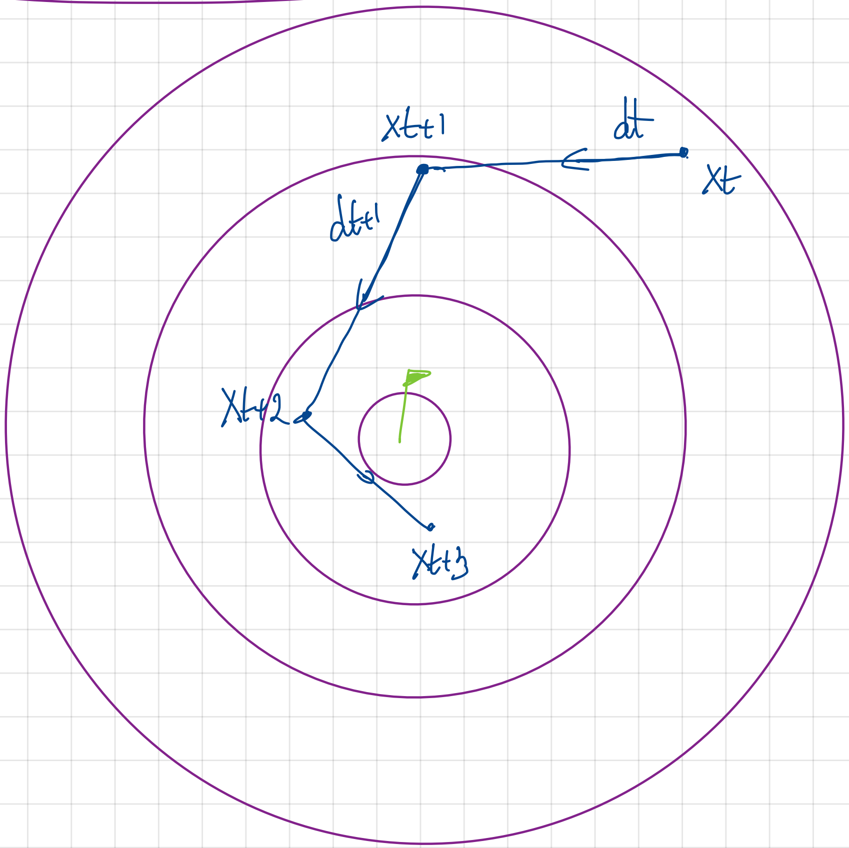
WHILE NOT HAPPY [WHILE f not minimized enough].

- choose a descent direction $d_t \neq 0$ $\quad d_t \in \mathbb{R}^n$
- line search
  - → choose a step-size $\sigma_t > 0$
  - → set $x_{t+1} = x_t + \sigma_t d_t$
- set $t = t+1$

Remaining questions:

- how to choose d_t ?
- how to choose σ_t ?

Picture with level sets



How to choose a descent direction ?

We can choose for $d_t = -Df(x_t)$

this is a descent direction:

if f is differentiable and if σ is small enough then

σ small enough

$$f(\underbrace{x_t - \sigma Df(x_t)}_{x_{t+1}}) \simeq f(x_t) - \sigma Df(x_t)^T Df(x_t)$$

$$= f(x_t) - \sigma \|Df(x_t)\|^2$$

$$< f(x_t)$$

$\Rightarrow$ $-Df(x_t)$ is a descent direction

from Taylor formula:

$$f(x+h) = f(x) + Df(x)^T h + o(\|h\|)$$

h small $\quad f(x+h) \simeq f(x) + Df(x)^T h$

$$\Rightarrow f(x_t - \underbrace{\sigma Df(x_t)}_{h}) \simeq f(x_t) + Df(x_t)^T(-\sigma Df(x_t)) = f(x_t) - \sigma Df(x_t)^T Df(x_t) = f(x_t) - \sigma \|Df(x_t)\|^2$$

Choice of the step-size?

$\sigma \overset{g}{\mapsto} f(x_t - \sigma Df(x_t))$

$\sigma_t = \underset{\sigma}{\operatorname{argmin}}\, g(\sigma)$

optimal step-size: $\sigma_t = \underset{\sigma \geq 0}{\operatorname{argmin}}\, f(x_t - \sigma Df(x_t))$

Typically too expensive to do those 1D optimization perfectly

There exists different techniques. One widely used one is Armijo rule.

When do we stop the overall algorithm

→ We can track $f(x_{t+1}) - f(x_t)$ (stop when it's small)

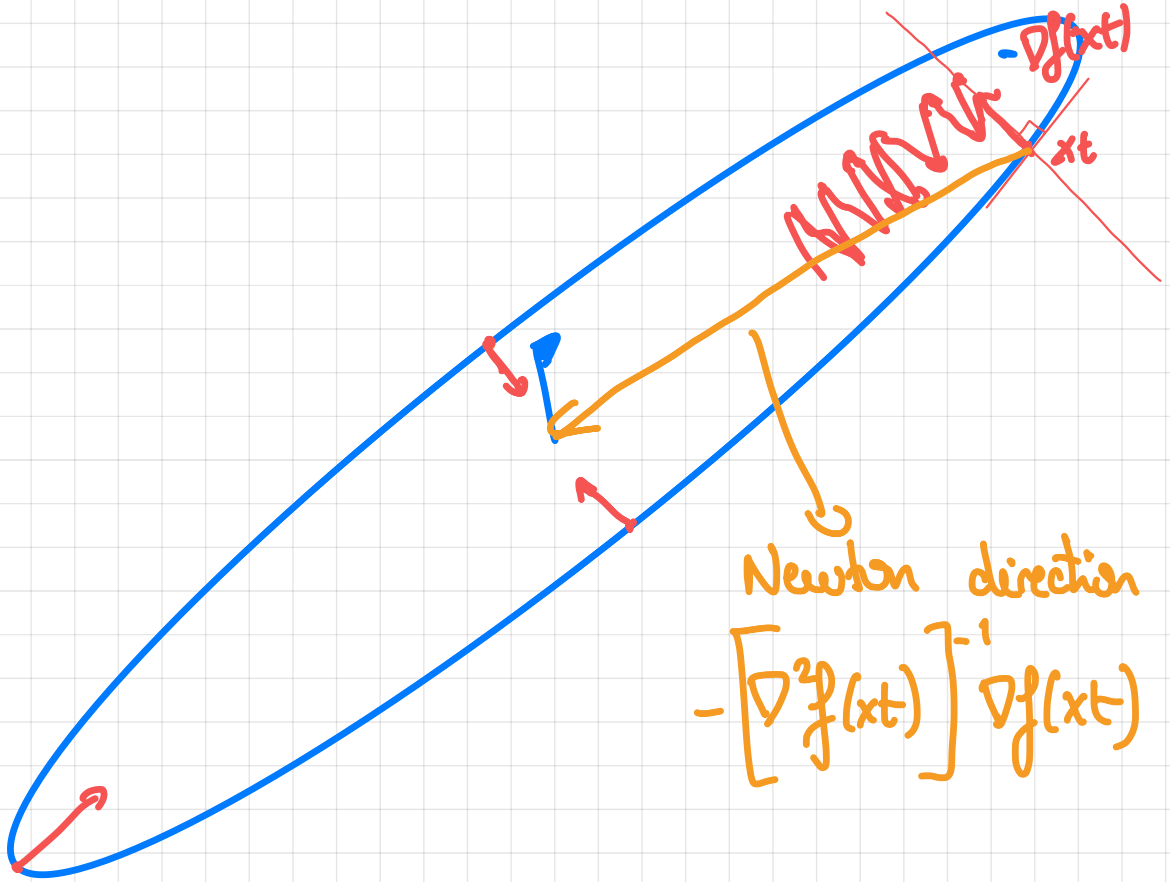
→ We can stop when $\|Df(x_t)\|$ is small.

Remark: If instead of minimizing f, I want to maximize f, we talk about gradient ascent (instead of gradient descent) and the update reads:

$$x_{t+1} = x_t + \sigma_t \nabla f(x_t)$$

You can always turn $\max_x f(x)$ into $\min(-f(x))$

Gradient descent is slow on ill-conditionned problems:



In a ill-conditionned function $-\nabla f$ typically points in the "wrong" direction and the convergence will be slow.

→ This is also something that can be proven: the convergence rate is slower the larger the condition number is.

The Newton direction points towards the optimum on convex quadratic functions.

On functions that are not convex-quadratic, the Newton direction will typically <u>NOT</u> points towards the optimum. Yet it will be a good direction to follow when you can approximate the function by its second order Taylor expansion (i.e for twice continuously differentiable function).

We can use the Newton direction $-[\nabla^2 f(x^t)]^{-1}\nabla f(x^t)$ as a descent direction.

↳ It minimizes the locally quadratic approximation of f.

$$f(x+\Delta x) = f(x) + \nabla f(x)^T \Delta x + \frac{1}{2}(\Delta x)^T \nabla^2 f(x)\Delta x$$

In some settings we can compute the Newton direction analytically, in which case we should do.

Yet we need to approximate numerically $[D^2 f(x)]$ and invert it, this can be too expensive.

QUASI-NEWTON METHOD : BFGS ["old" still state-of the art]

$$x_{t+1} = x_t - \sigma_t H_t Df(x_t)$$

approximation of the inverse of $D^2 f(x_t)$

H_t is updated iteratively using $Df(x_t)$ and approximates $D^2 f$

cf Wikipedia page for updates of algorithm

→ Implemented in toolboxes [also large-scale version, L-BFGS
limit memory BFGS]

STOCHASTIC GRADIENT DESCENT

Minimize loss function of the following form:

$$Q(w) = \frac{1}{N} \sum_{i=1}^{N} Q_i(w)$$

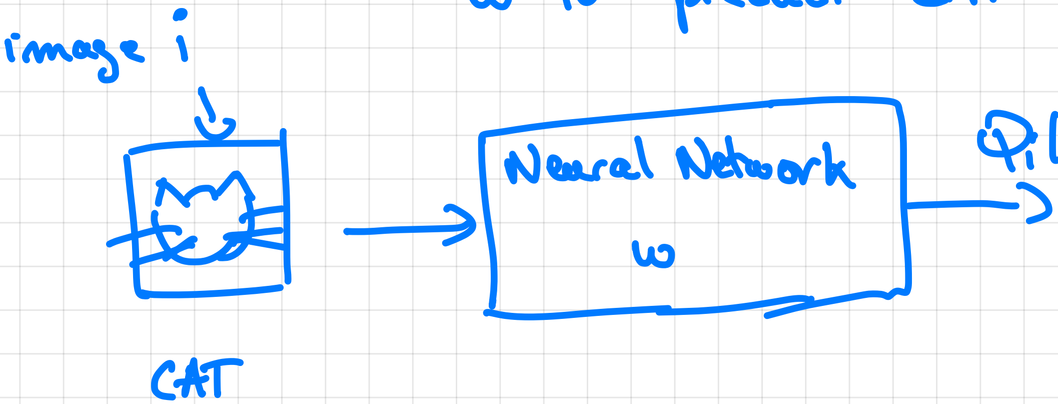
N = # Data

Examples

w can be the weights of Neural Network.

Assume we are in a supervised learning setting, we have a classification task.

$Q_i(w)$: prediction error made if we use weight w to predict CAT



$Q_i(w)$: 0 if w predicts that it's a cat

1 otherwise

How do we minimize Q ?

Gradient descent : $$\nabla Q(w) = \frac{1}{N} \sum_{i=1}^{N} \nabla Q_i(w)$$

$$w_{t+1} = w_t - \sigma_t \nabla Q(w_t)$$ [Update of weights]

BACKPROPAGATION algorithm is an algorithm to compute $\nabla Q_i(w)$

Typically N is very large, computation of all $\nabla Q_i(w)$, $i=1,\dots,N$ is too expensive.

Instead we use an approximation of $\nabla Q(w)$:

$$\nabla Q(w) \underset{\text{approximated}}{\approx} \nabla Q_i(w)$$ [Gradient of a single example]

Also do mini-batches :

$$\nabla Q(w) \approx \frac{1}{\text{nbatches}} \sum_{i=1}^{\text{nbatches}} \nabla Q_i(w) \qquad \text{nbatches} \ll N.$$

Stochastic Gradient Descent :

CHOOSE AN INITIAL VECTORS OF PARAMETERS AND A STEP-SIZE η

WHILE NOT HAPPY

- Randomly shuffle examples in training set
- For $i = 1, \ldots, N$

$$w \leftarrow w - \eta \nabla Q_i(w)$$

We loop over the examples

possibly mini-batches

Not covered: - choice of step-size. (Step-size adapted using "momentum techniques" in particular ADAM step-size update which is WIDELY used)

- increase | choice of mini-batches