Optimus II

CS771: Introduction to Machine Learning

Purushottam Kar

Announcements

Quiz/exam copies shall be graded on Gradescope

Will create accounts for you – no action needed from your side

It may take a week or so for grading (many graders are on leave)

Will release Assignment 1 this week as well

Recap of Last Lecture

Notions of (local/global) extrema, derivatives (first, second)

Multivariate analogues of these (gradient, Hessian)

Stationary points and the (multivariate) second derivative test

Gradient as offering the directions of *steepest* ascent/descent

Convex sets and convex functions

Several examples and exercises in course notes (see GitHub repo)

A Toy Example – Function Values

6	3	3	3	3	3	3	3	3	3
5	2	2	2	3	4	3	3	2	1
4	1	1	1	3	3	3	1	1	1
3	1	0	1	1	2	1	1	0	1
2	1	1	1	3	3	3	1	1	1
1	1	2	3	3	4	3	2	2	2
0	3	3	3	3	3	3	3	3	3
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**

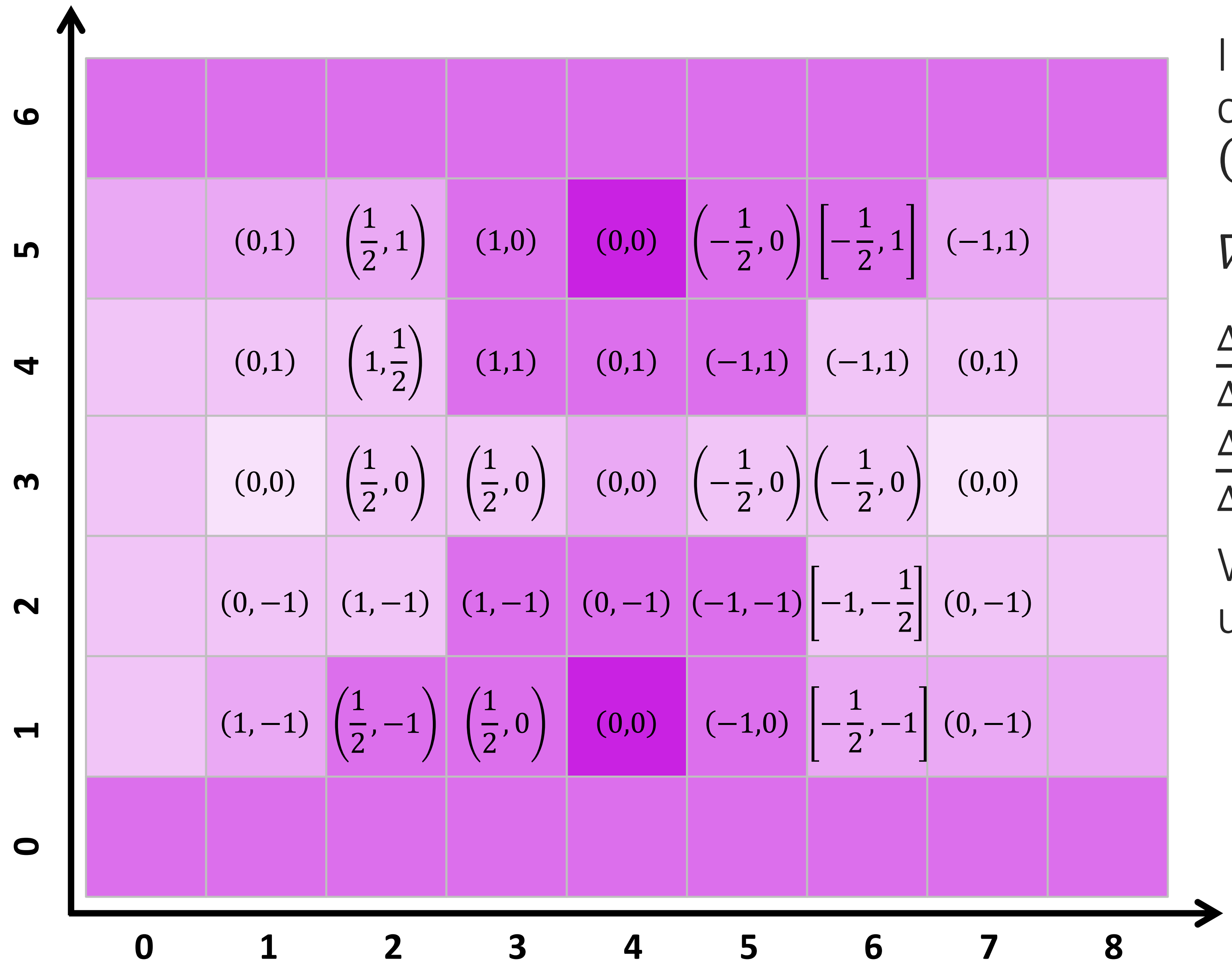
In this discrete toy example, we can calculate gradient at a point (x_0, y_0) as

$$\nabla f(x_0, y_0) = \left(\frac{\Delta f}{\Delta x}, \frac{\Delta f}{\Delta y}\right) \text{ where}$$

$$\frac{\Delta f}{\Delta x} = \frac{f(x_0+1, y_0) - f(x_0-1, y_0)}{2}$$

$$\frac{\Delta f}{\Delta y} = \frac{f(x_0, y_0+1) - f(x_0, y_0-1)}{2}$$

A Toy Example – Gradients



In this discrete toy example, we can calculate gradient at a point (x_0, y_0) as

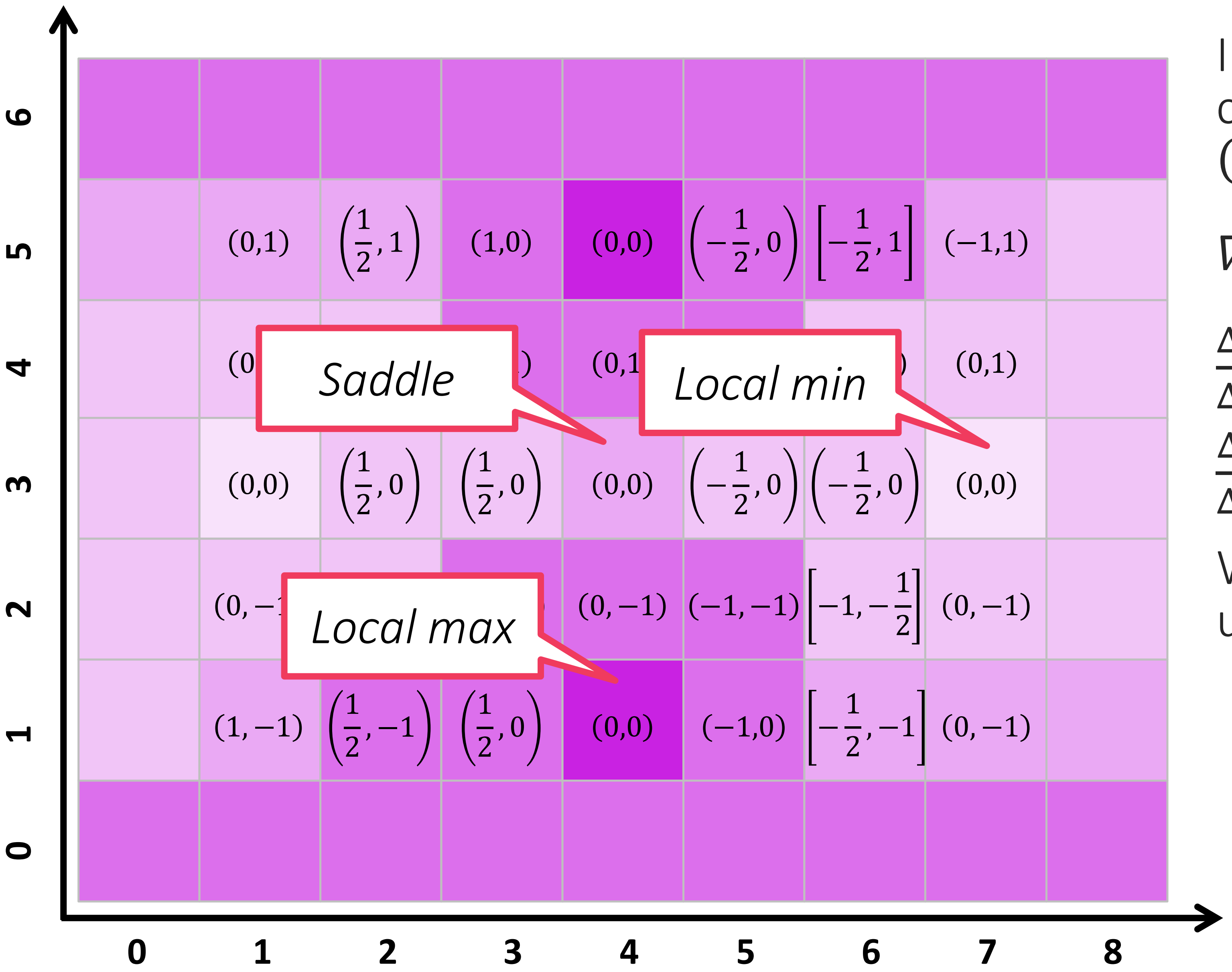
$$\nabla f(x_0, y_0) = \left(\frac{\Delta f}{\Delta x}, \frac{\Delta f}{\Delta y}\right) \text{ where}$$

$$\frac{\Delta f}{\Delta x} = \frac{f(x_0+1, y_0) - f(x_0-1, y_0)}{2}$$

$$\frac{\Delta f}{\Delta y} = \frac{f(x_0, y_0+1) - f(x_0, y_0-1)}{2}$$

We can visualize these gradients using simple arrows as well

A Toy Example – Gradients



In this discrete toy example, we can calculate gradient at a point (x_0, y_0) as

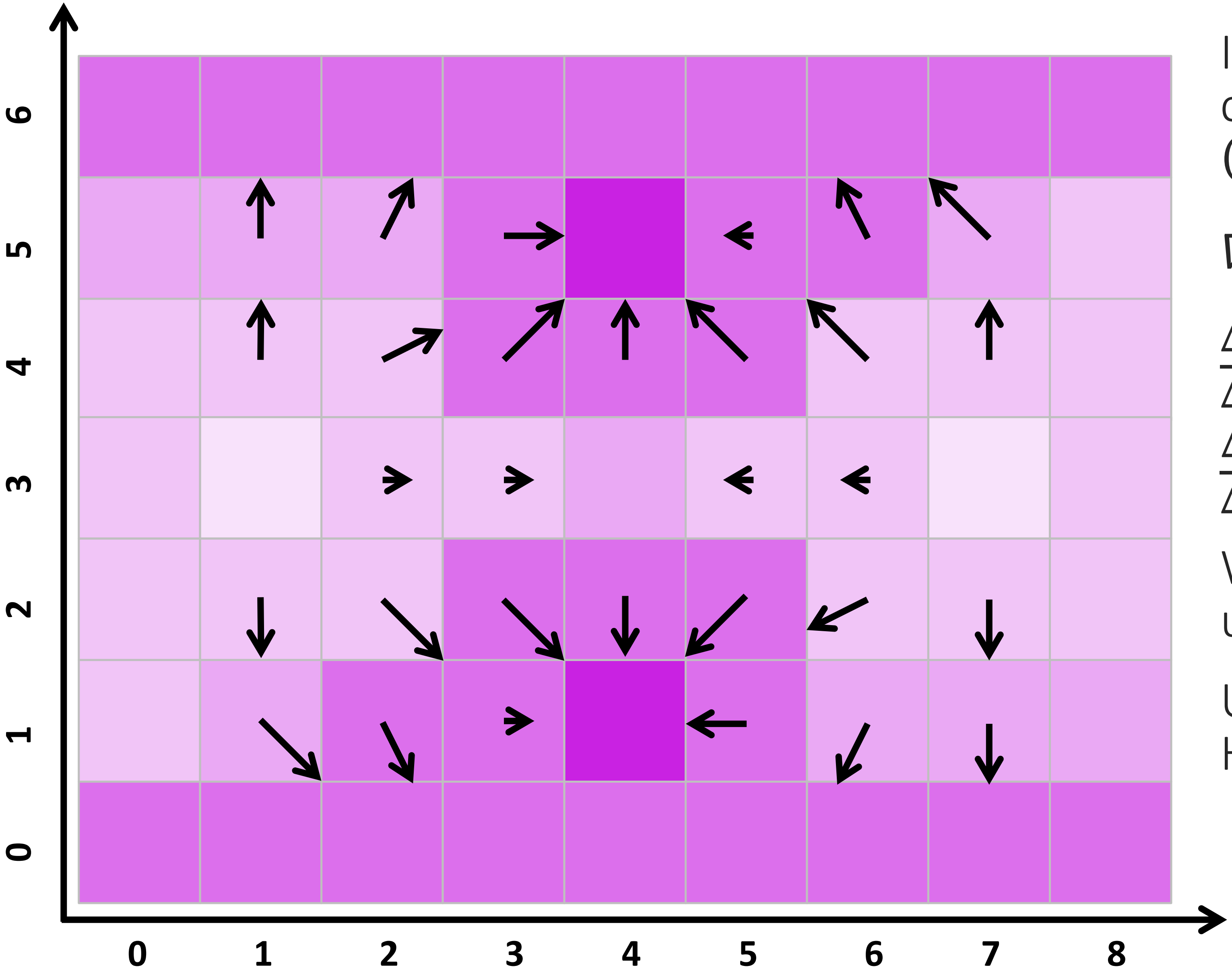
$$\nabla f(x_0, y_0) = \left(\frac{\Delta f}{\Delta x}, \frac{\Delta f}{\Delta y}\right) \text{ where}$$

$$\frac{\Delta f}{\Delta x} = \frac{f(x_0+1, y_0) - f(x_0-1, y_0)}{2}$$

$$\frac{\Delta f}{\Delta y} = \frac{f(x_0, y_0+1) - f(x_0, y_0-1)}{2}$$

We can visualize these gradients using simple arrows as well

A Toy Example – Gradients



In this discrete toy example, we can calculate gradient at a point (x_0, y_0) as

$$\nabla f(x_0, y_0) = \left(\frac{\Delta f}{\Delta x}, \frac{\Delta f}{\Delta y}\right) \text{ where}$$

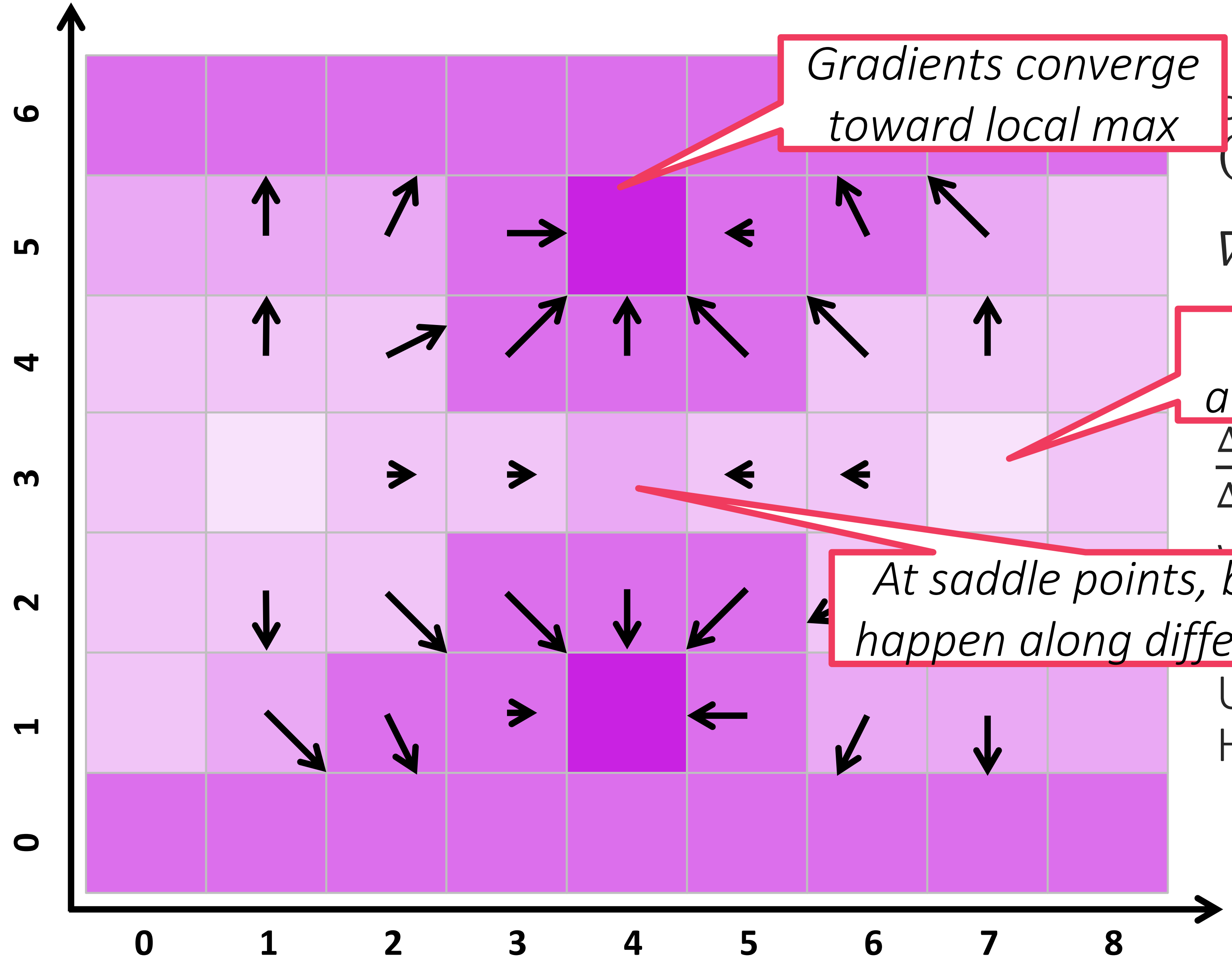
$$\frac{\Delta f}{\Delta x} = \frac{f(x_0+1, y_0) - f(x_0-1, y_0)}{2}$$

$$\frac{\Delta f}{\Delta y} = \frac{f(x_0, y_0+1) - f(x_0, y_0-1)}{2}$$

We can visualize these gradients using simple arrows as well

Using a similar method, the Hessian can be calculated as well!

A Toy Example – Gradients



n this discrete toy example, we can calculate gradient at a point (x_0, y_0) as

$$\nabla f(x_0, y_0) = \left(\frac{\Delta f}{\Delta x}, \frac{\Delta f}{\Delta y}\right) \text{ where}$$

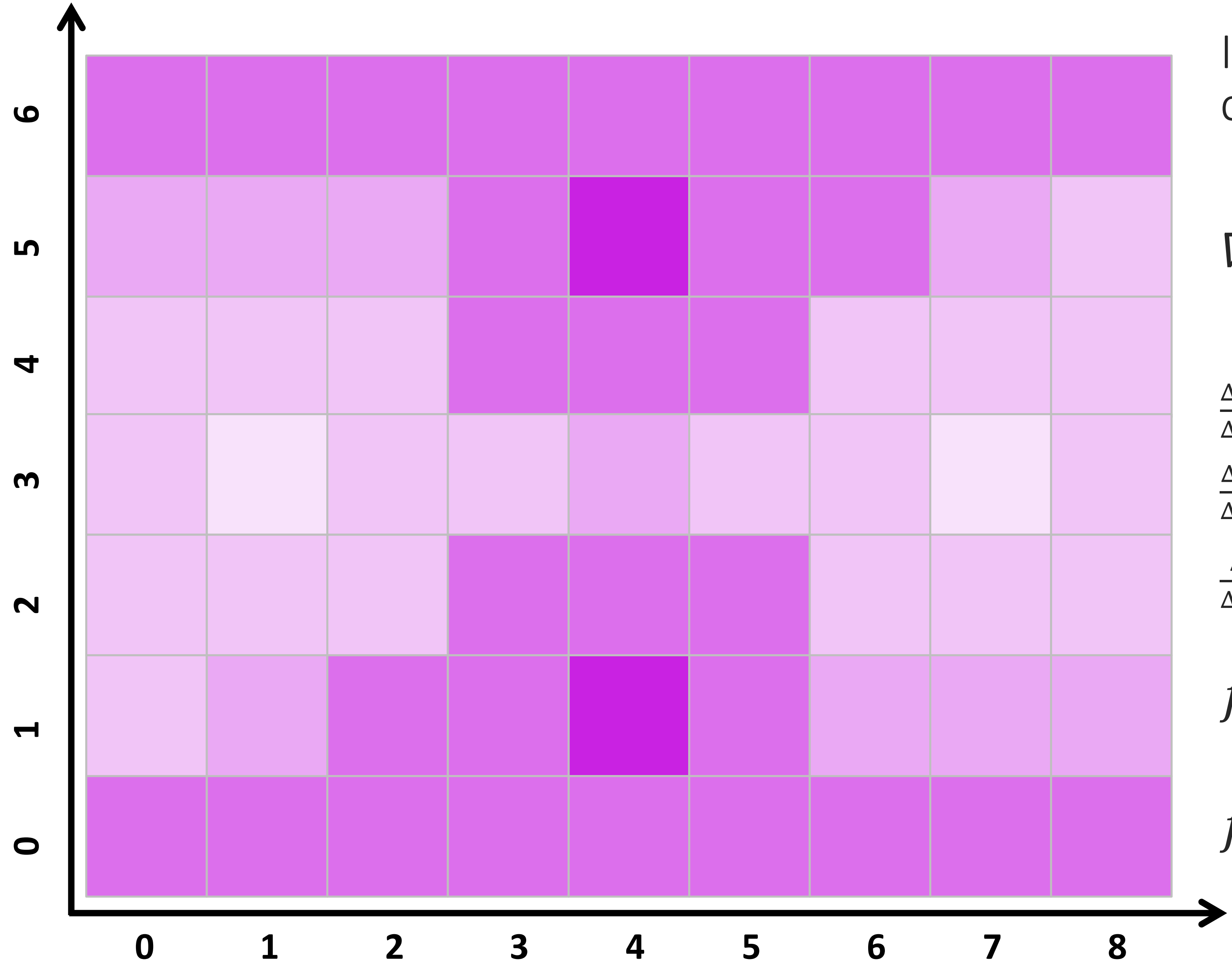
$-1, y_0)$

$$\frac{\Delta f}{\Delta y} = \frac{f(x_0, y_0+1) - f(x_0, y_0-1)}{2}$$

W... ualize these gradients ...e arrows as well

Using a similar method, the Hessian can be calculated as well!

A Toy Example – Hessians



In this discrete toy example, we can calculate Hessian at (x_0, y_0) as

$$\nabla^2 f(x_0, y_0) = \begin{bmatrix} \frac{\Delta^2 f}{\Delta x^2} & \frac{\Delta^2 f}{\Delta x \Delta y} \\ \frac{\Delta^2 f}{\Delta x \Delta y} & \frac{\Delta^2 f}{\Delta y^2} \end{bmatrix} \text{ where}$$

$$\frac{\Delta^2 f}{\Delta x^2} = f(x_0 + 1, y_0) + f(x_0 - 1, y_0) - 2f(x_0, y_0)$$

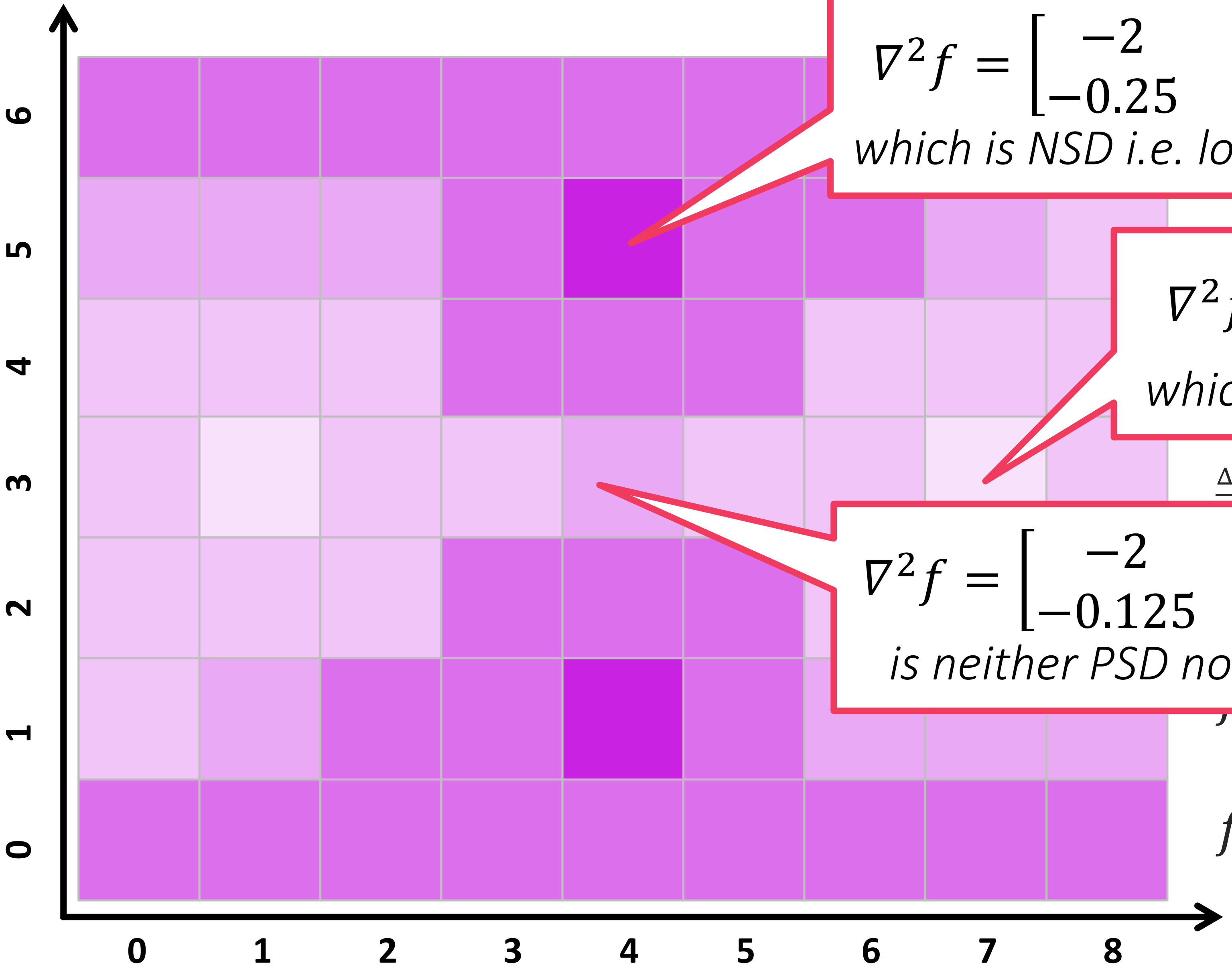
$$\frac{\Delta^2 f}{\Delta y^2} = f(x_0, y_0 + 1) + f(x_0, y_0 - 1) - 2f(x_0, y_0)$$

$$\frac{\Delta^2 f}{\Delta x \Delta y} = \frac{(f_{xy} + f_{yx})}{2} \text{ where}$$

$$f_{xy} = \frac{\frac{\Delta f}{\Delta x}(x_0, y_0 + 1) - \frac{\Delta f}{\Delta x}(x_0, y_0 - 1)}{2}$$

$$f_{yx} = \frac{\frac{\Delta f}{\Delta y}(x_0 + 1, y_0) - \frac{\Delta f}{\Delta y}(x_0 - 1, y_0)}{2}$$

A Toy Example – Hessians



$\nabla^2 f = \begin{bmatrix} -2 & -0.25 \\ -0.25 & -2 \end{bmatrix}$

which is NSD i.e. local max

$\nabla^2 f = \begin{bmatrix} 2 & 0.125 \\ 0.125 & 2 \end{bmatrix}$

which is PSD i.e. local min

$\nabla^2 f = \begin{bmatrix} -2 & -0.125 \\ -0.125 & 2 \end{bmatrix}$ *which is neither PSD nor NSD (saddle)*

crete toy example, we can
Hessian at (x_0, y_0) as

$$\begin{bmatrix} \frac{\Delta^2 f}{\Delta x^2} & \frac{\Delta^2 f}{\Delta x \Delta y} \\ & \frac{\Delta^2 f}{\Delta y^2} \end{bmatrix} \text{ where}$$

$) - 2f(x_0, y_0)$

$$\frac{\Delta^2 f}{\Delta y^2} = f(x_0, y_0 + 1) + f(x_0, y_0 - 1) - 2f(x_0, y_0)$$

$$f_{xy} = \frac{\quad - \frac{\Delta f}{\Delta x}(x_0, y_0 - 1)}{2}$$

$$f_{yx} = \frac{\frac{\Delta f}{\Delta y}(x_0 + 1, y_0) - \frac{\Delta f}{\Delta y}(x_0 - 1, y_0)}{2}$$

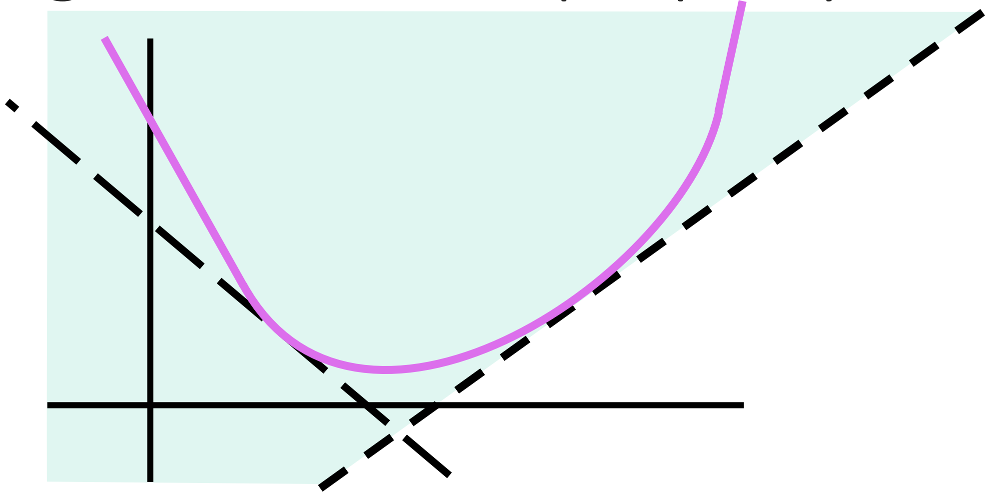
Non-differentiable Functions

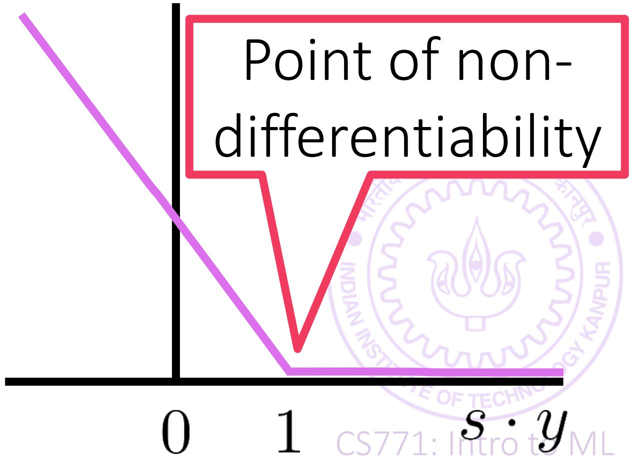
The hinge loss function is not differentiable everywhere ☹

- Can we define some form of gradient for non-diff functions as well?
- Yes, if a function is convex, then no matter if it is non-differentiable, a notion of gradient called *subgradient* can always be defined for it

Recall that for differentiable functions, the gradient defines a *tangent* hyperplane at every point and the function must lie above this plane

Subgradients exploit and generalize this property ☺





Subgradients and the subdifferential

Tangents of a convex differentiable function $f: \mathbb{R}^d \to \mathbb{R}$ are uniquely linked to its gradients

The tangent at $\mathbf{x}^0$ *is the hyperplane* $\nabla f(\mathbf{x}^0)^\top(\mathbf{x} - \mathbf{x}^0) + f(\mathbf{x}^0)$

Convex functions lie above all tangents $f(\mathbf{x}) \geq \nabla f(\mathbf{x}^0)^\top(\mathbf{x} - \mathbf{x}^0) + f(\mathbf{x}^0)$

Trick: turn the definition around and say that gradient at $\mathbf{x}^0$ is a vector $\mathbf{g}$ so that the hyperplane $\mathbf{g}^\top(\mathbf{x} - \mathbf{x}^0) + f(\mathbf{x}^0) = 0$ is tangent to f at $\mathbf{x}^0$

Subgradients: given a (possibly non-differentiable but convex) function $f: \mathbb{R}^d \to \mathbb{R}$ and a point $\mathbf{x}^0$, any vector $\mathbf{g}$ that satisfies

$$f(\mathbf{x}) \geq \mathbf{g}^\top(\mathbf{x} - \mathbf{x}^0) + f(\mathbf{x}^0)$$

is called a subgradient of f at $\mathbf{x}^0$

Subdifferential: the set of all subgradients of f at a point $\mathbf{x}^0$ is known as the subdifferential of f at $\mathbf{x}^0$ and denoted by $\partial f(\mathbf{x}^0)$

How can I find out the subgradients of a function?

Wait! Does this mean a function can have more than one subgradient at a point $\mathbf{x}^0$

S... gra... e sub...

...s of a convex differentiable function $f:\mathbb{R}^d \rightarrow \mathbb{R}$ are uniq...

...o its gradients

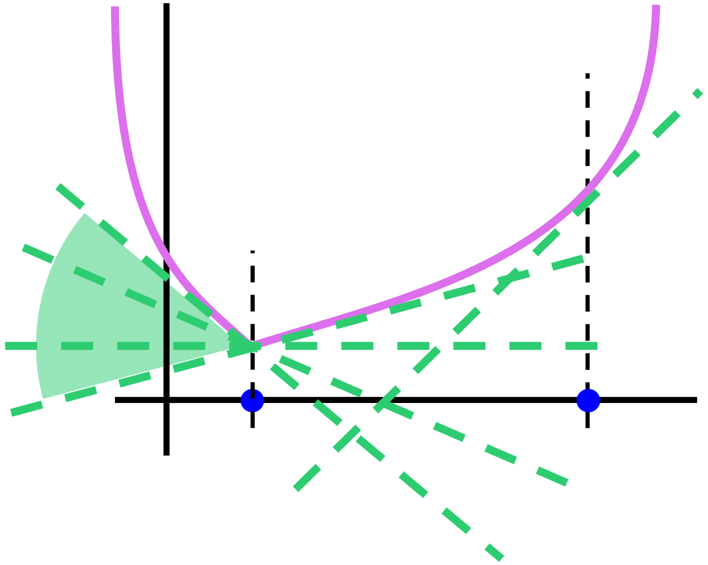
If f is non-differentiable at $\mathbf{x}^0$ then it can indeed have multiple subgradients at $\mathbf{x}^0$. However, if f is differentiable at $\mathbf{x}^0$, then it can have only one subgradient at $\mathbf{x}^0$, and that is the gradient $\nabla f(\mathbf{x}^0)$ itself ☺

The tangent at $\mathbf{x}^0$ is the hyperp...

Convex functions lie above all ta...

Trick: turn the definition arour... a ve... $\mathbf{g}$ so that the hyperplane $\mathbf{g}^\top(\mathbf{x}-\mathbf{x}^0)+f(\mathbf{x}^0)=0$ is tangent to f at $\mathbf{x}^0$

$$\partial f(\mathbf{x}^0) \triangleq \{\mathbf{g} : f(\mathbf{x}) \geq \mathbf{g}^\top(\mathbf{x}-\mathbf{x}^0)+f(\mathbf{x}^0) \ \ \forall \mathbf{x}\}$$



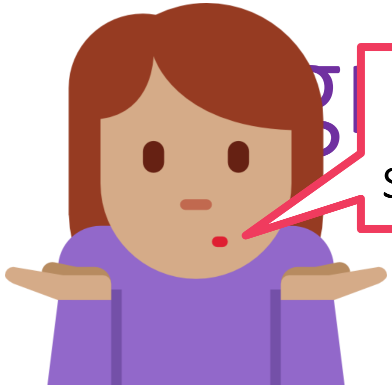
Subgradient Calculus

	Gradient Calculus	Subgradient Calculus
	$\mathbf{x} \in \mathbb{R}^d, \mathbf{a} \in \mathbb{R}^d, b, c \in \mathbb{R}$	
Scaling Rule	$\nabla(c \cdot f)(\mathbf{x}) = c \cdot \nabla f(\mathbf{x})$	$\partial(c \cdot f)(\mathbf{x}) = c \cdot \partial f(\mathbf{x}) = \{c \cdot \mathbf{v} : \mathbf{v} \in \partial f(\mathbf{x})\}$
Sum Rule	$\nabla(f + g)(\mathbf{x}) = \nabla f(\mathbf{x}) + \nabla g(\mathbf{x})$	$\partial(f + g)(\mathbf{x}) = \partial f(\mathbf{x}) + \partial g(\mathbf{x}) = \{\mathbf{u} + \mathbf{v} : \mathbf{u} \in \partial f(\mathbf{x}), \mathbf{v} \in \partial g(\mathbf{x})\}$
Chain Rule	$\nabla f(\mathbf{a}^\top \mathbf{x} + b) = f'(\mathbf{a}^\top \mathbf{x} + b) \cdot \mathbf{a}$	$\partial f(\mathbf{a}^\top \mathbf{x} + b) = \partial f(\mathbf{a}^\top \mathbf{x} + b) \cdot \mathbf{a} = \{c \cdot \mathbf{a} : c \in \partial f(\mathbf{a}^\top \mathbf{x} + b)\}$
Max Rule	**No counterpart in general**	$h(\mathbf{x}) = \max\{f(\mathbf{x}), g(\mathbf{x})\}$

If $f(\mathbf{x}^0) > g(\mathbf{x}^0), \partial h(\mathbf{x}^0) = \partial f(\mathbf{x}^0)$. If $g(\mathbf{x}^0) > f(\mathbf{x}^0), \partial h(\mathbf{x}^0) = \partial g(\mathbf{x}^0)$

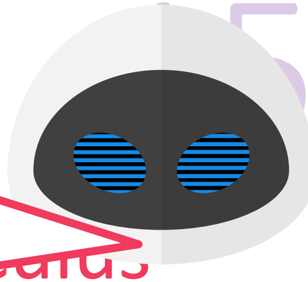
If $f(\mathbf{x}^0) = g(\mathbf{x}^0), \partial h(\mathbf{x}^0) = \{\lambda \mathbf{u} + (1 - \lambda)\mathbf{v} : \mathbf{u} \in \partial f(\mathbf{x}^0), \mathbf{v} \in \partial g(\mathbf{x}^0), \lambda \in [0,1]\}$

Gradien... ...ulus

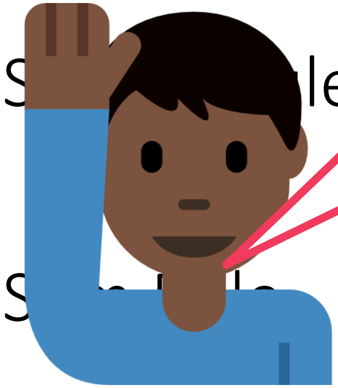


What about stationary points?

Good point! In subgradient calculus, a point $\mathbf{x}^0$ is a stationary point for a function f if the zero vector is a part of the subdifferential i.e. $\mathbf{0} \in \partial f(\mathbf{x}^0)$



$$\mathbf{x} \in \mathbb{R}^d, \mathbf{a} \in \mathbb{R}^d, b, c \in \mathbb{R}$$



Local minima/maxima must be stationary in this sense even for non-differentiable functions

S...le $\quad \partial(c \cdot f)(\mathbf{x}) = c \cdot \partial f(\mathbf{x})$

$$= \{c \cdot \mathbf{v} : \mathbf{v} \in \partial f(\mathbf{x})\}$$

S... ...le $\quad \nabla(f+g)(\mathbf{x}) = \nabla f(\mathbf{x}) + \nabla g(\mathbf{x}) \quad \partial(f+g)(\mathbf{x}) = \partial f(\mathbf{x}) + \partial g(\mathbf{x})$

$$= \{\mathbf{u} + \mathbf{v} : \mathbf{u} \in \partial f(\mathbf{x}), \mathbf{v} \in \partial g(\mathbf{x})\}$$

Chain Rule $\quad \nabla f(\mathbf{a}^\top \mathbf{x} + b) = f'(\mathbf{a}^\top \mathbf{x} + b) \cdot \mathbf{a} \quad \partial f(\mathbf{a}^\top \mathbf{x} + b) = \partial f(\mathbf{a}^\top \mathbf{x} + b) \cdot \mathbf{a}$

$$= \{c \cdot \mathbf{a} : c \in \partial f(\mathbf{a}^\top \mathbf{x} + b)\}$$

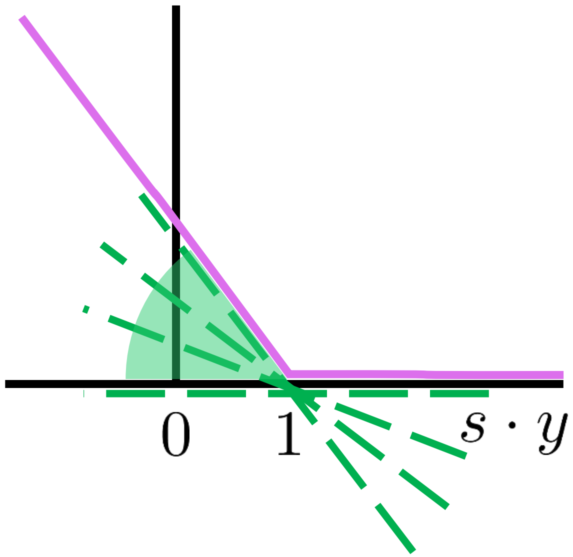
Max Rule $\quad$ **No counterpart in general** $\quad h(\mathbf{x}) = \max\{f(\mathbf{x}), g(\mathbf{x})\}$

If $f(\mathbf{x}^0) > g(\mathbf{x}^0), \partial h(\mathbf{x}^0) = \partial f(\mathbf{x}^0)$. If $g(\mathbf{x}^0) > f(\mathbf{x}^0), \partial h(\mathbf{x}^0) = \partial g(\mathbf{x}^0)$

If $f(\mathbf{x}^0) = g(\mathbf{x}^0), \partial h(\mathbf{x}^0) = \{\lambda \mathbf{u} + (1-\lambda)\mathbf{v} : \mathbf{u} \in \partial f(\mathbf{x}^0), \mathbf{v} \in \partial g(\mathbf{x}^0), \lambda \in [0,1]\}$

Example: subgradient for hinge loss

$\ell_{\text{hinge}}(x) = \max\{1 - x, 0\} = \max\ \{f(x), g(x)\}$



ℓ_{hinge} is differentiable at all points except $x = 1$

Thus, $\partial \ell_{\text{hinge}}(x) = \ell'_{\text{hinge}}(x)$ *if* $x \neq 1$

At $x = 1$ use subdifferential calculus

$f(x) = 1 - x$ *(differentiable) i.e.* $\partial f(x) = f'(x) = -1$

$g(x) = 0$ *(differentiable) i.e.* $\partial g(x) = g'(x) = 0$

Thus, $\partial \ell_{\text{hinge}}(1) = \{\lambda \cdot (-1) + (1 - \lambda) \cdot 0 : \lambda \in [0,1]\} = [-1,0]$

Example: subgradient for hinge loss

$\ell_{\text{hinge}}(x) = \max\{1 - x, 0\} = \max\{f(x), g(x)\}$

ℓ_{hinge} is differentiable at all points except $x = 1$

Thus, $\partial\ell_{\text{hinge}}(x) = \ell'_{\text{hinge}}(x)$ *if* $x \neq 1$

Applying subgradient chain rule gives us

$$\ell_{\text{hinge}}(y^i, \langle \mathbf{w}, \mathbf{x}^i \rangle) = [1 - y^i \langle \mathbf{w}, \mathbf{x}^i \rangle]_+$$

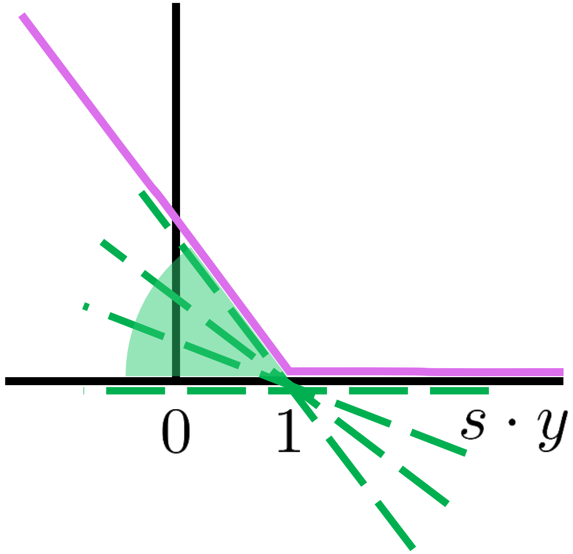
Need $\mathbf{v}^i \in \partial\ell_{\text{hinge}}(y^i, \langle \mathbf{w}, \mathbf{x}^i \rangle)$

$$\mathbf{v}^i = \begin{cases} \mathbf{0} & \text{if } y^i \langle \mathbf{w}, \mathbf{x}^i \rangle > 1 \\ -y^i \cdot \mathbf{x}^i & \text{if } y^i \langle \mathbf{w}, \mathbf{x}^i \rangle < 1 \\ c \cdot y^i \cdot \mathbf{x}^i & \text{if } y^i \langle \mathbf{w}, \mathbf{x}^i \rangle = 1 \\ c \in [-1, 0] \end{cases}$$

$) = -1$

0

$\in [0,1]\} = [-1,0]$



From calculUS to OPTIMization

Method 1: First order optimality Condition

Exploits the fact that gradient must vanish at a local optimum

Also exploits the fact that for convex functions, local minima are global

***Warning**: works only for convex functions and that too relatively simple ones*

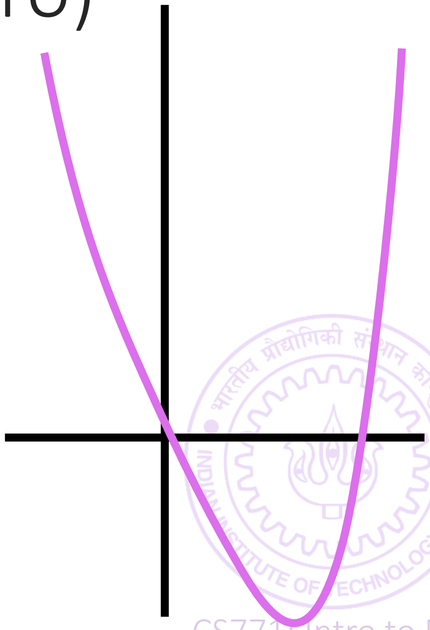
To Do: given a convex function that we wish to minimize, try finding all the stationary points of the function (set gradient to zero)

If you find only one, that has to be the global minimum ☺

Example: $f(x) = x^4 - 2x$

$f'(x) = 4x^3 - 2 = 0$ only at $x^* = \sqrt[3]{0.5}$

$f''(x) = 12x^2 \geq 0$ i.e. $f(x)$ is cvx i.e. x^* is global min



From calculUS to OPTIMization

Method 2: Perform (sub)gradient descent

Recall that direction opposite to gradient offers *steepest descent*

(SUB)GRADIENT DESCENT

1. **Given**: obj. func. $f: \mathbb{R}^d \rightarrow \mathbb{R}$ to minimize
2. Initialize $\mathbf{w}^0 \in \mathbb{R}^d$
3. For $t = 0, 1, \ldots$
    1. Obtain a (sub)gradient $\mathbf{g}^t \in \partial f(\mathbf{w}^t)$
    2. Choose a step length η_t
    3. Update $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta_t \cdot \mathbf{g}^t$
    4. Repeat until convergence

How to initialize $\mathbf{w}^0$?

How to choose η_t

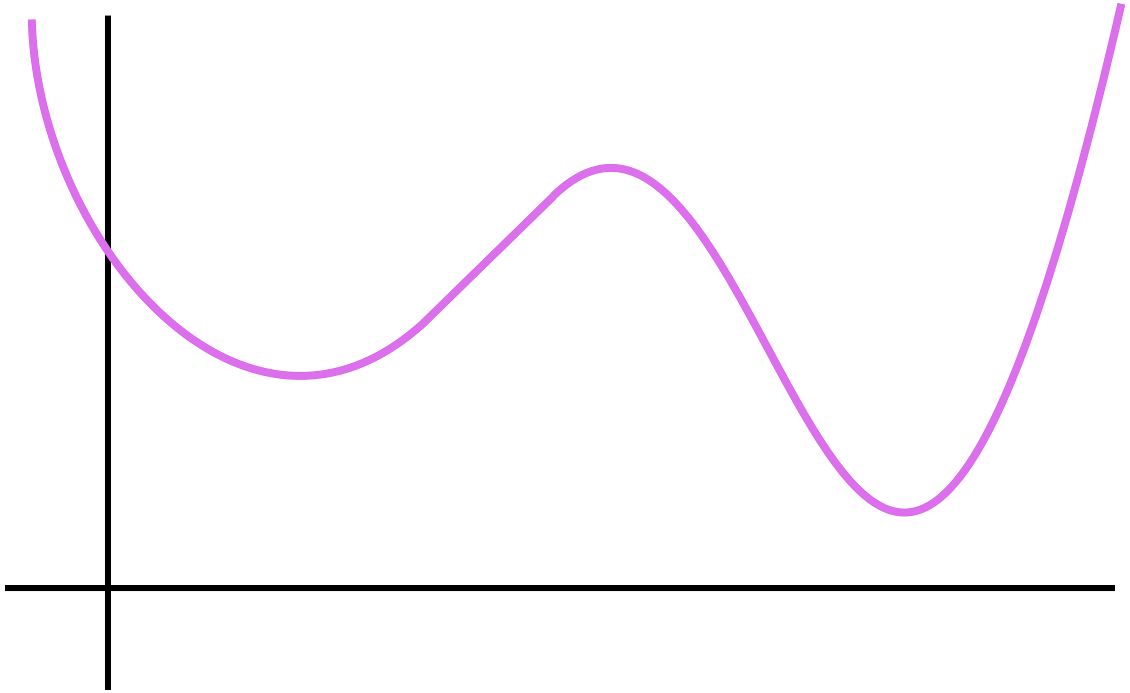
Often called step length or learning rate

How to decide if we have converged?

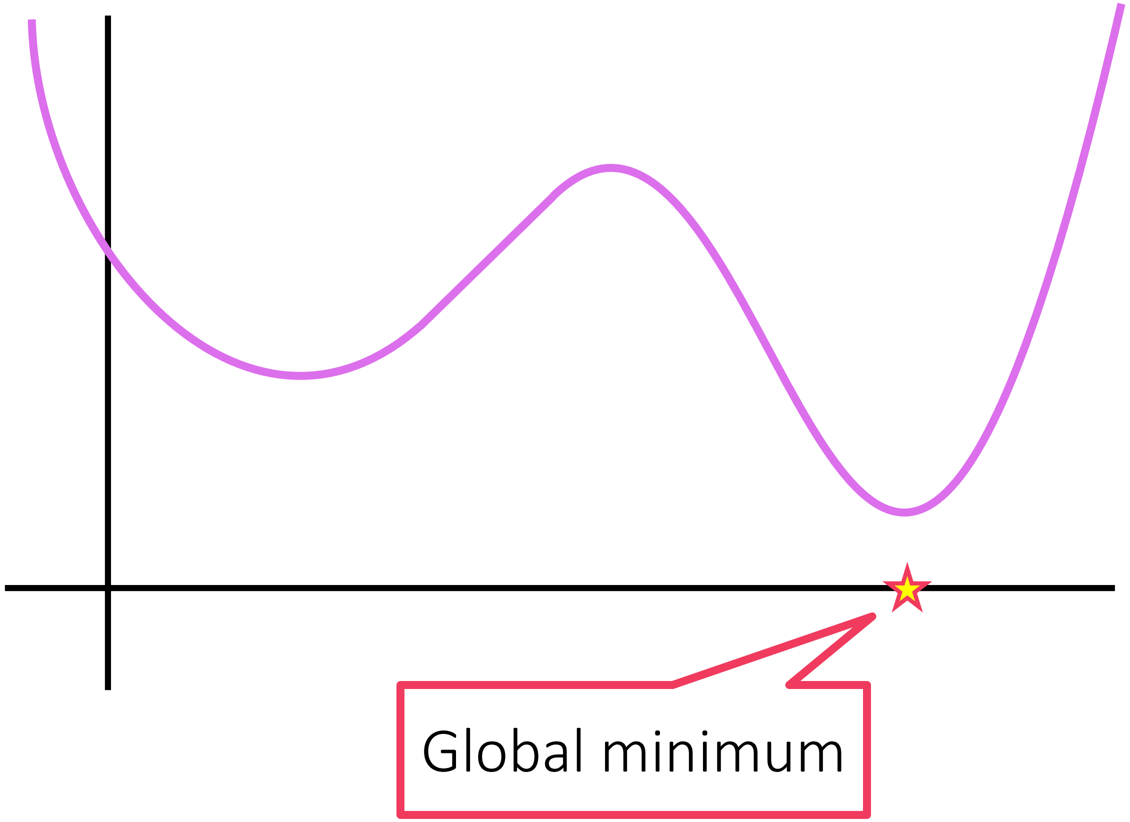
What does convergence even mean?

Gradient Descent (GD)

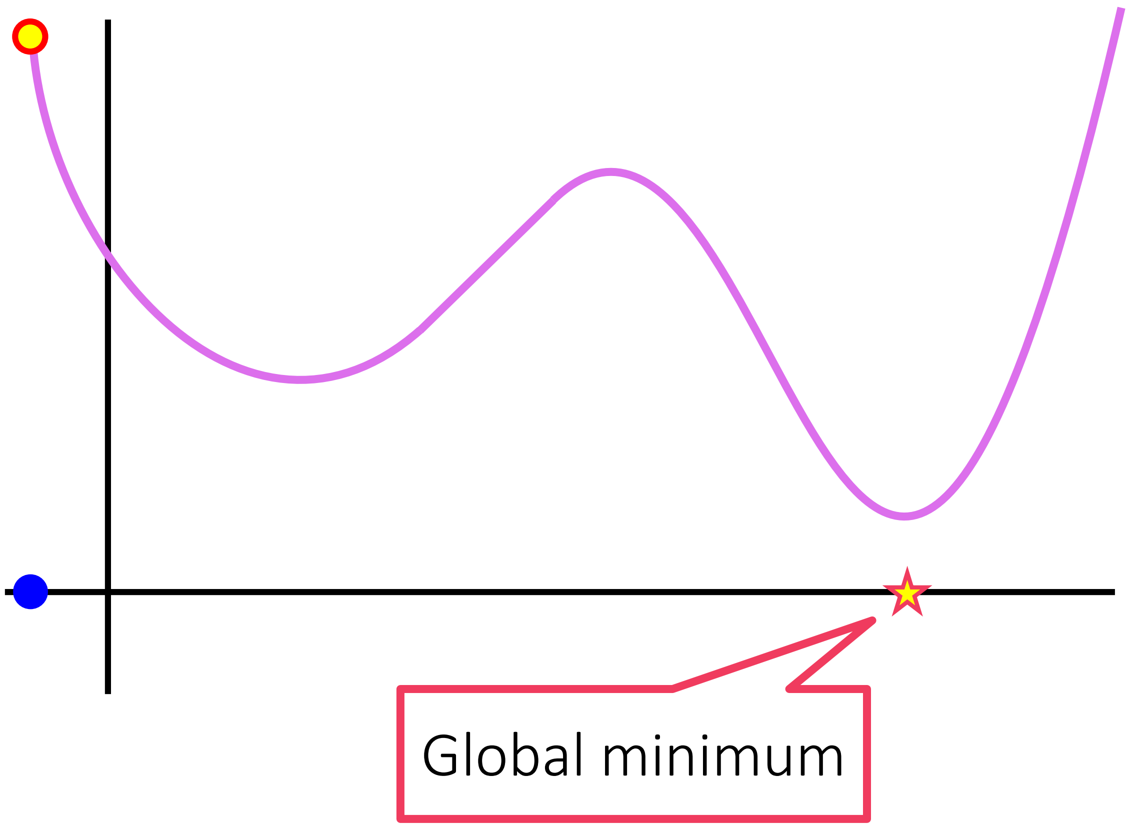
Gradient Descent (GD)



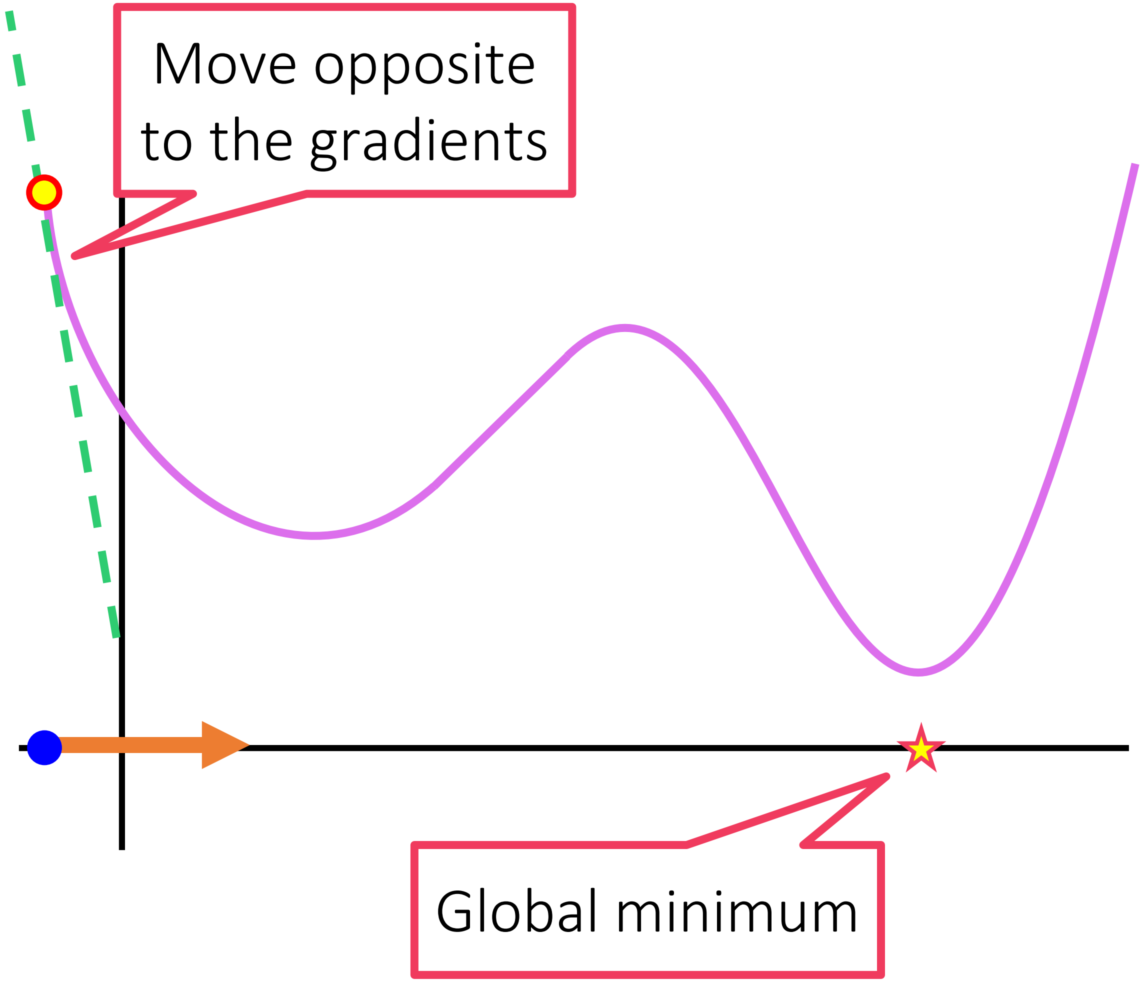
Gradient Descent (GD)



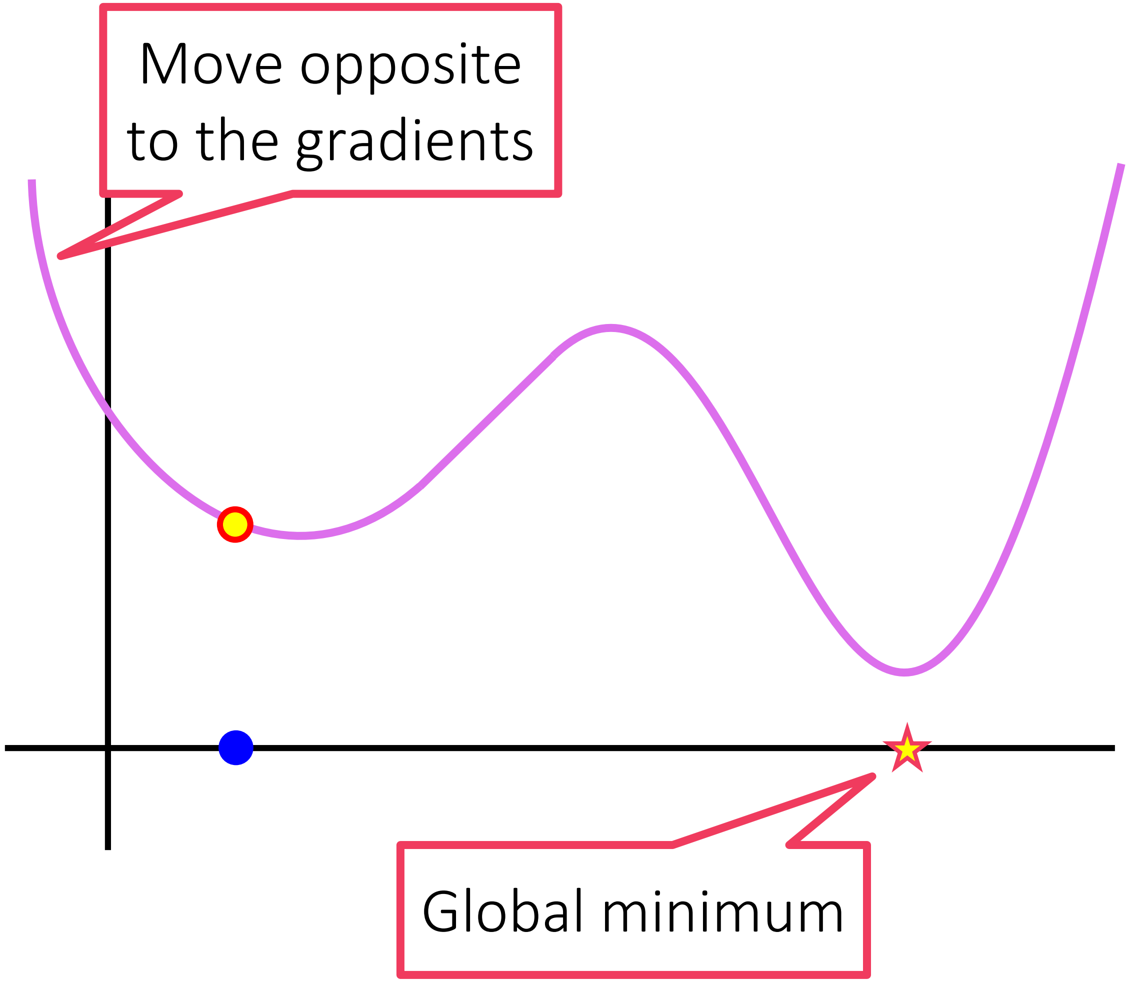
Gradient Descent (GD)



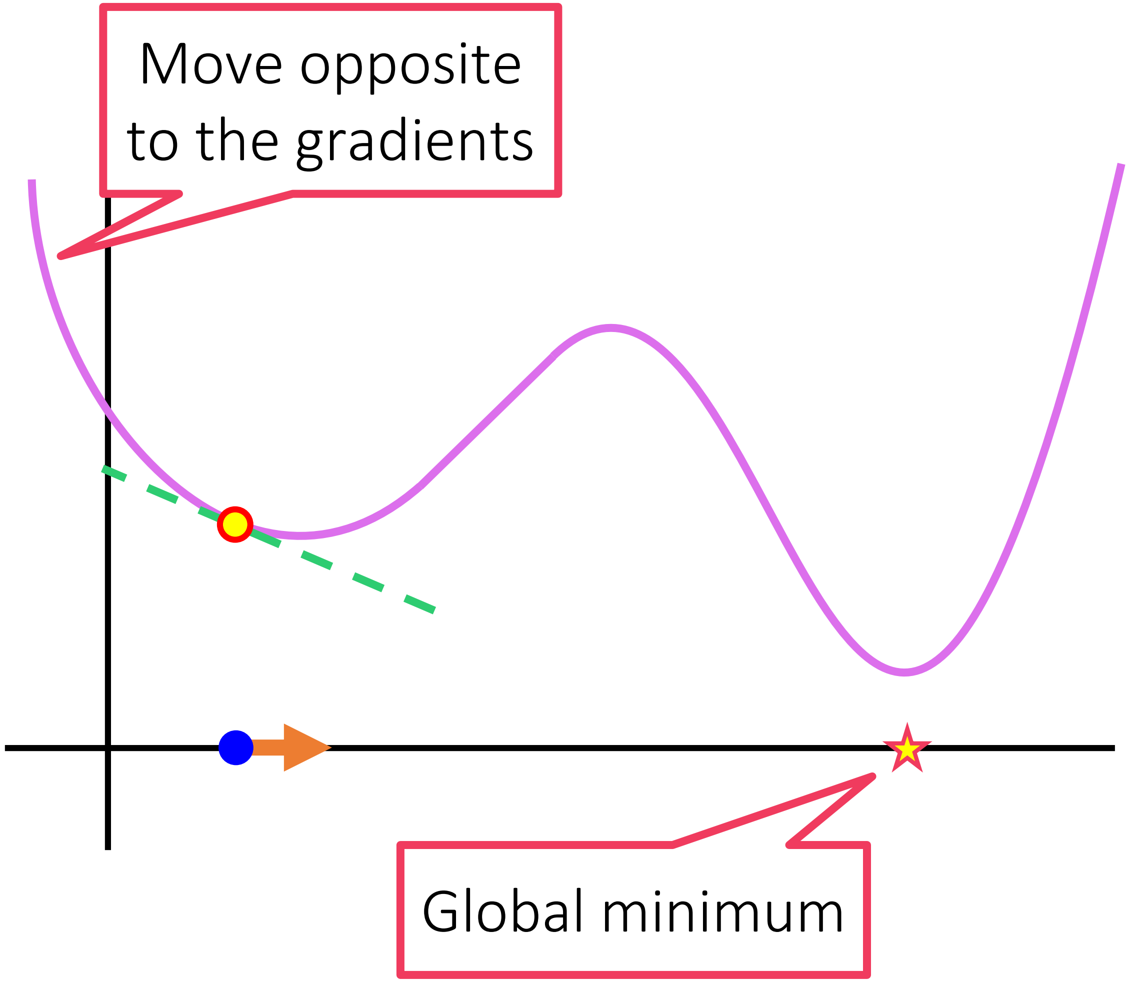
Gradient Descent (GD)



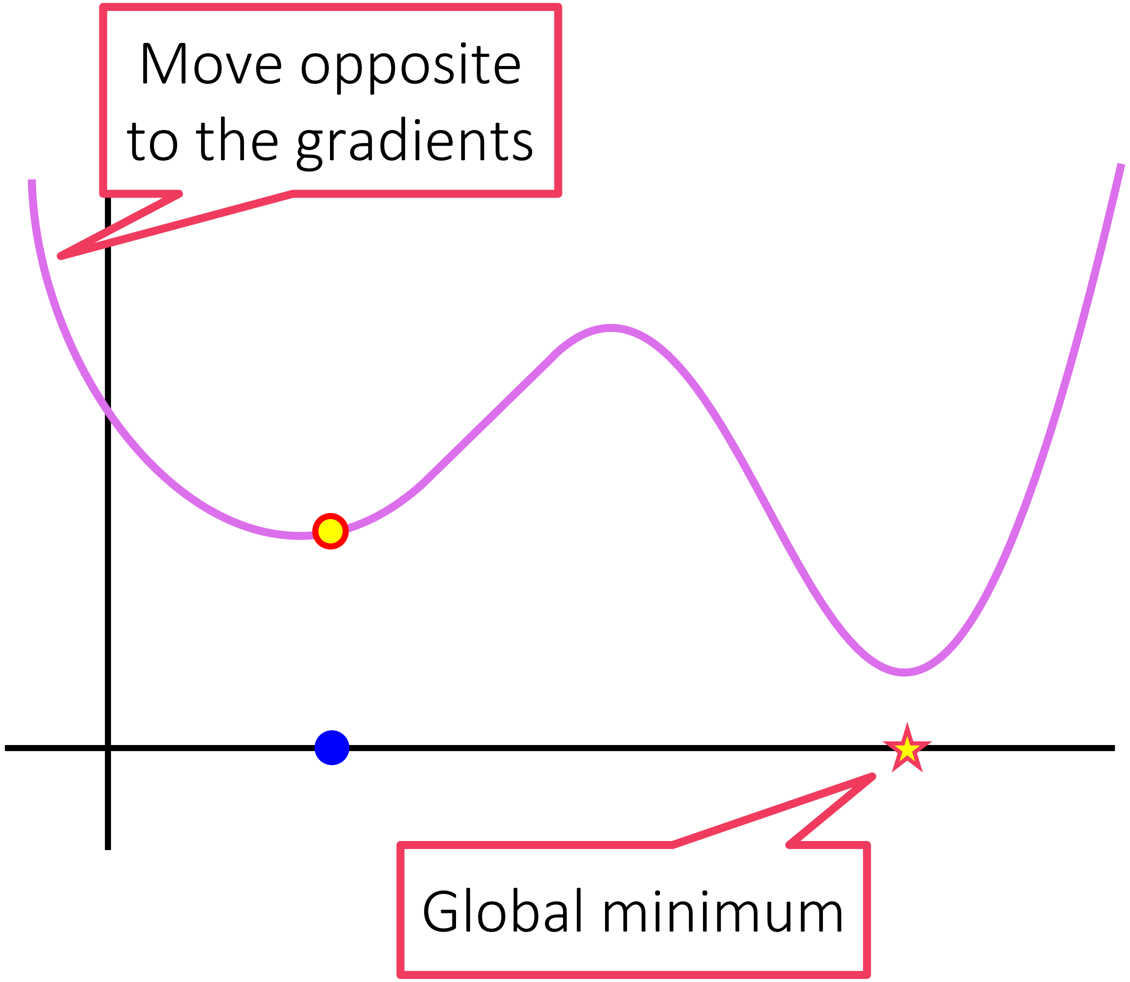
Gradient Descent (GD)



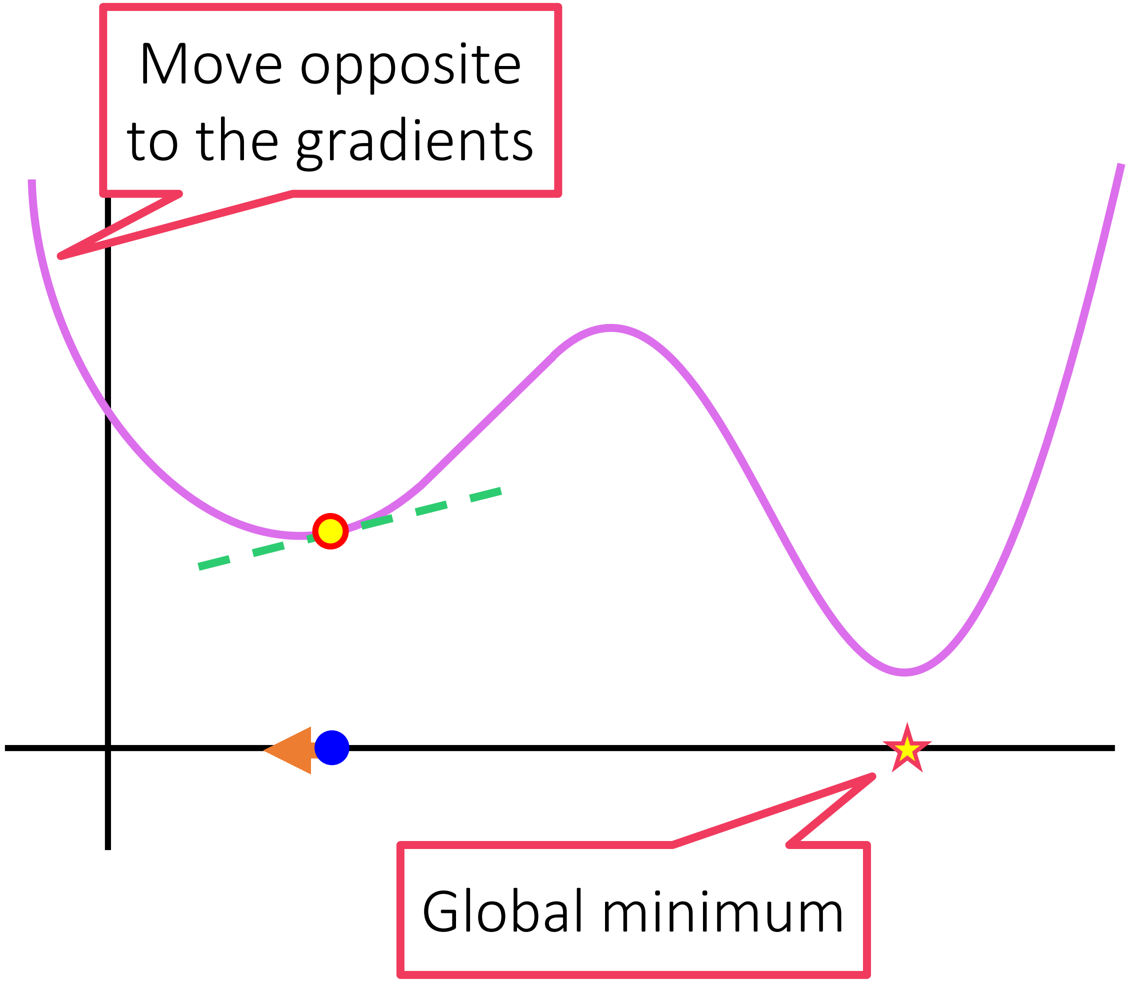
Gradient Descent (GD)



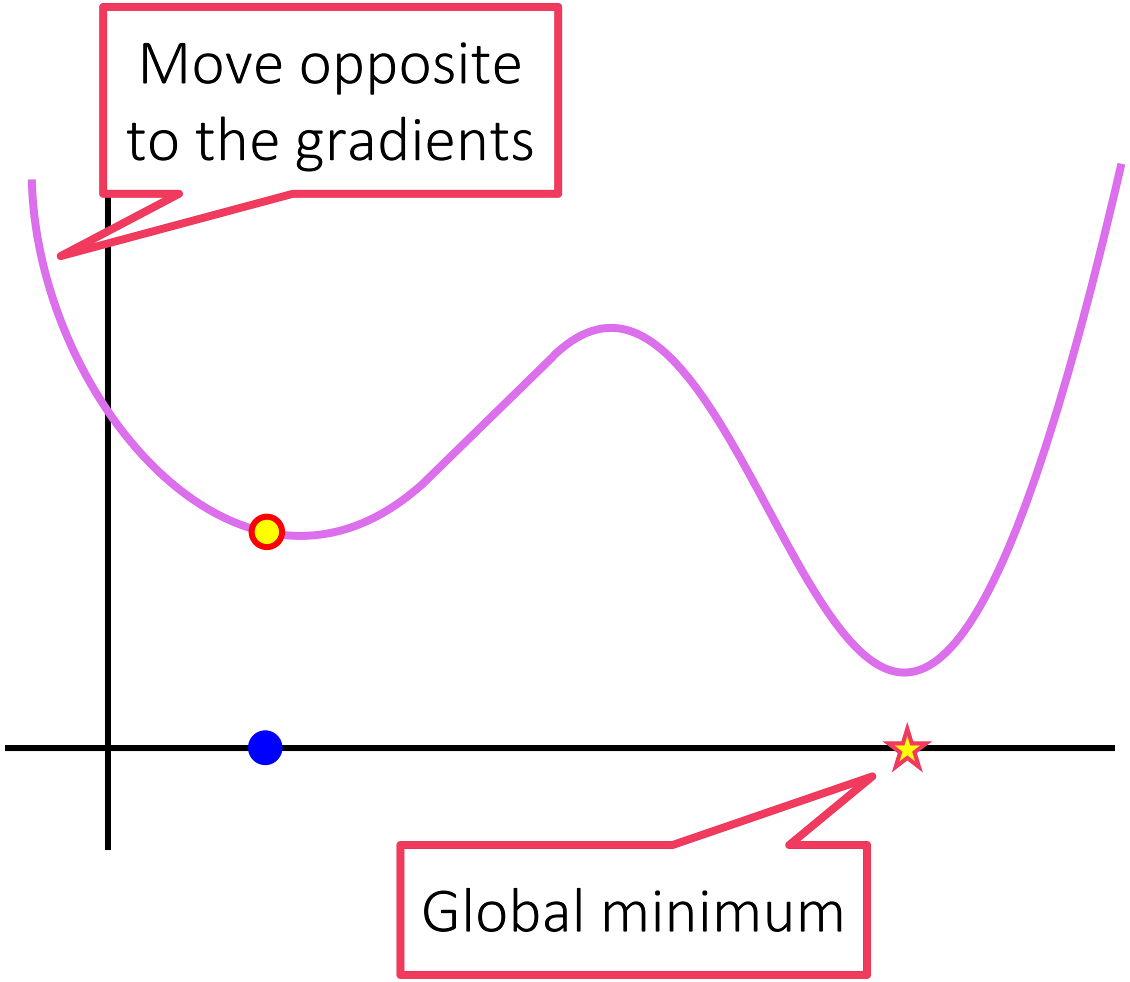
Gradient Descent (GD)



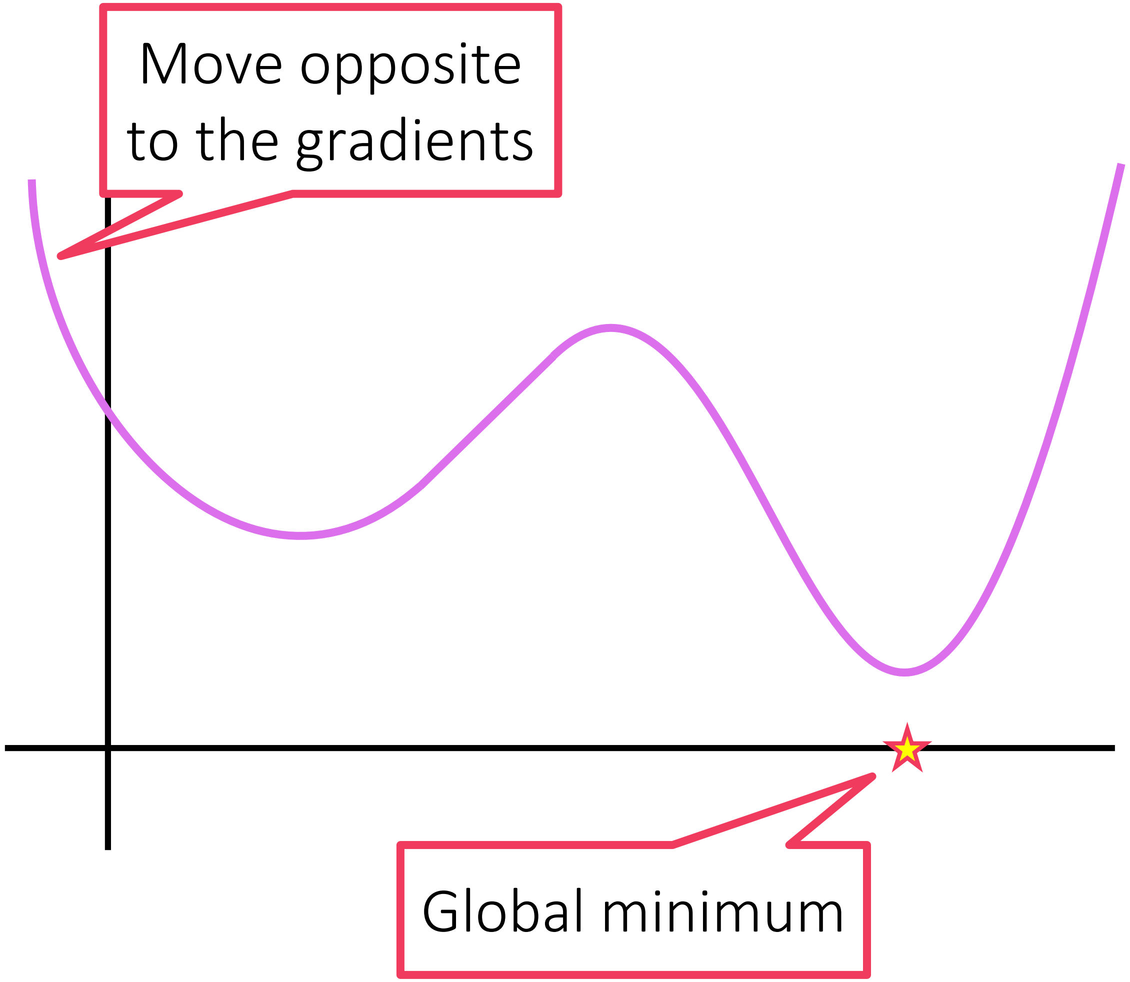
Gradient Descent (GD)



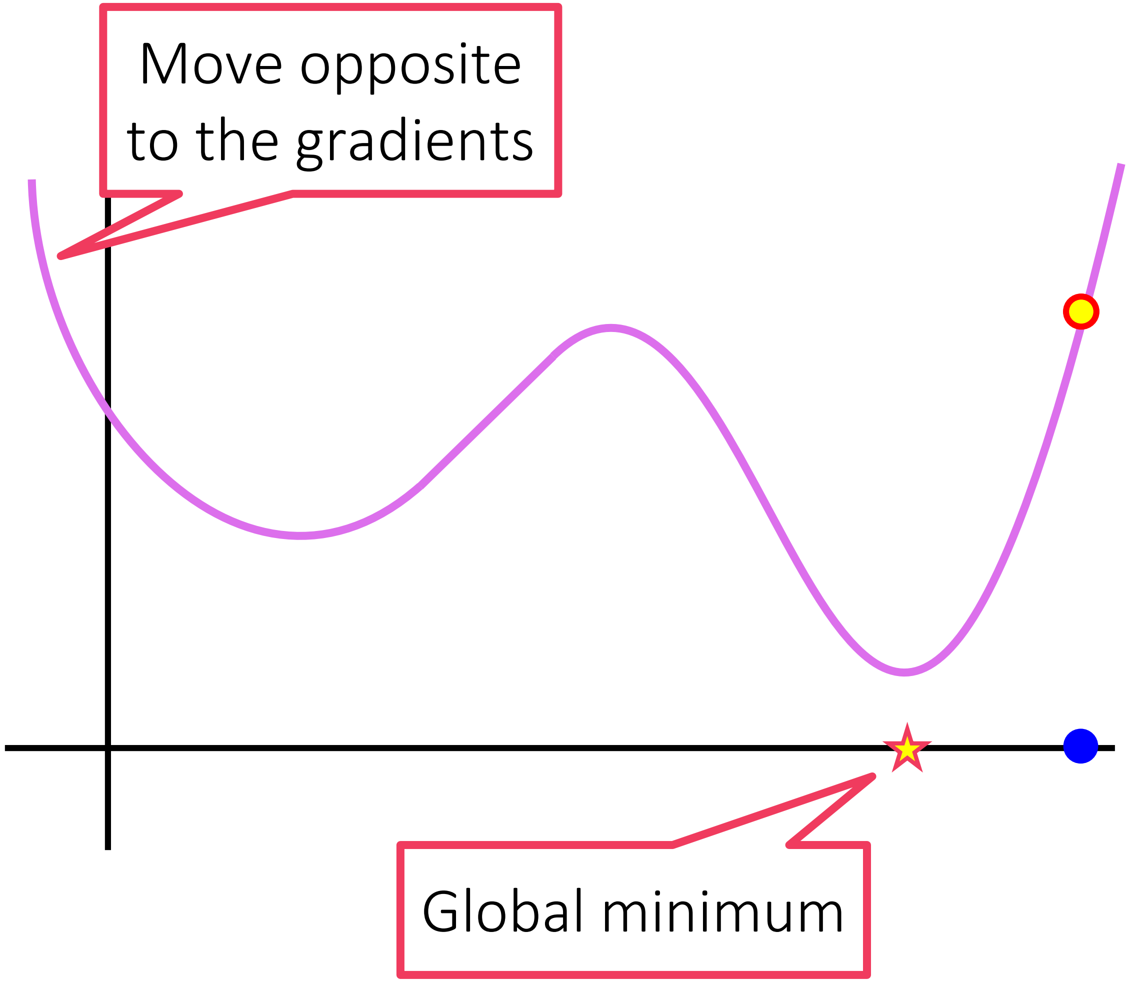
Gradient Descent (GD)



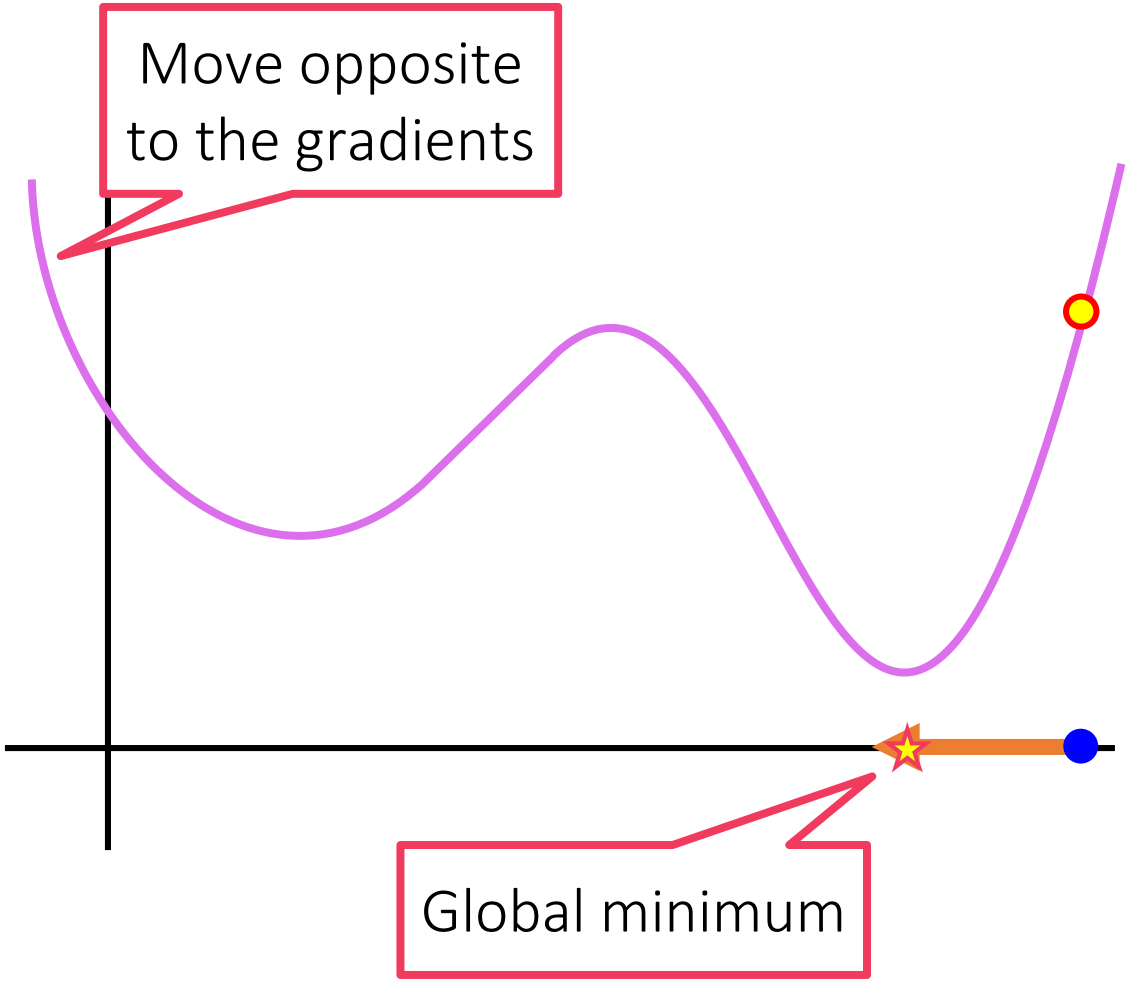
Gradient Descent (GD)



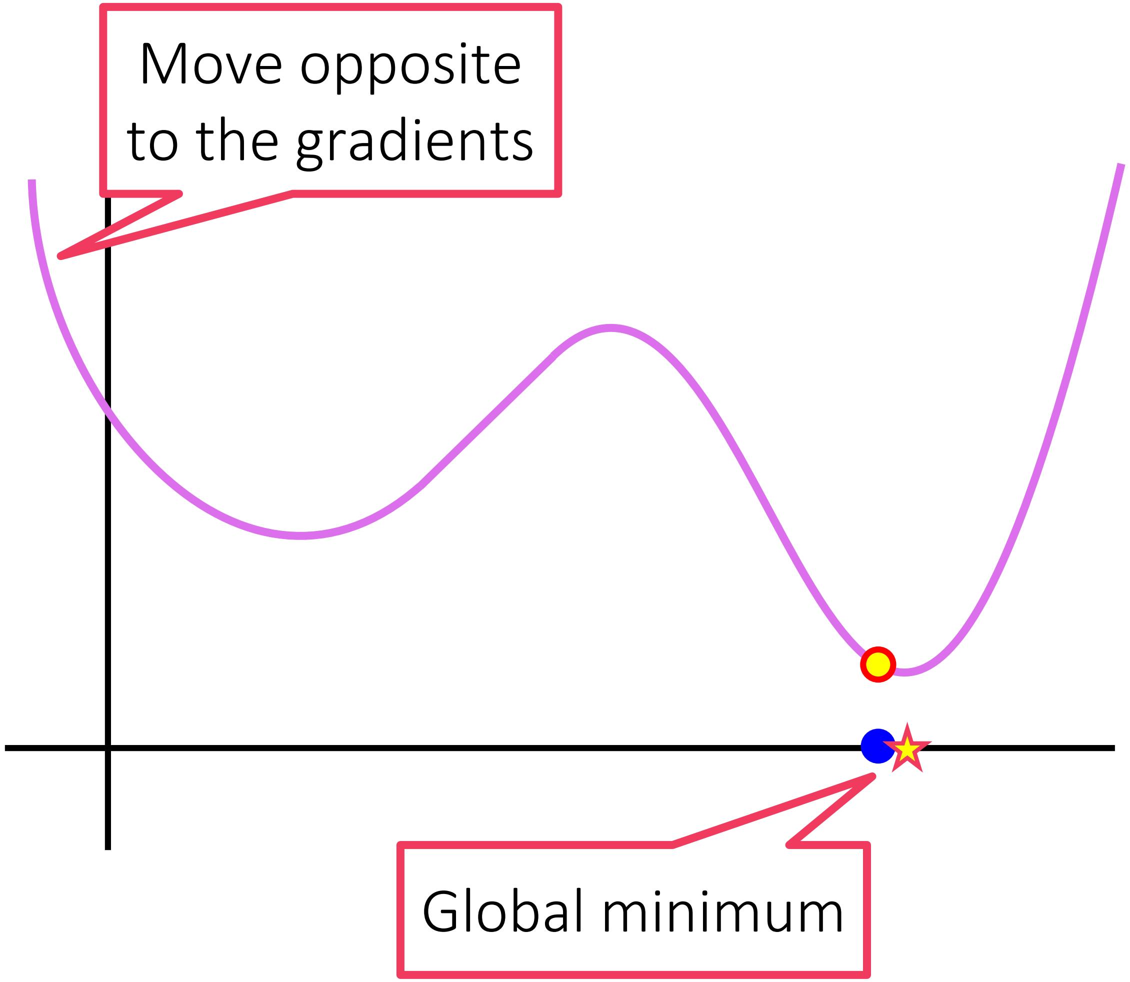
Gradient Descent (GD)



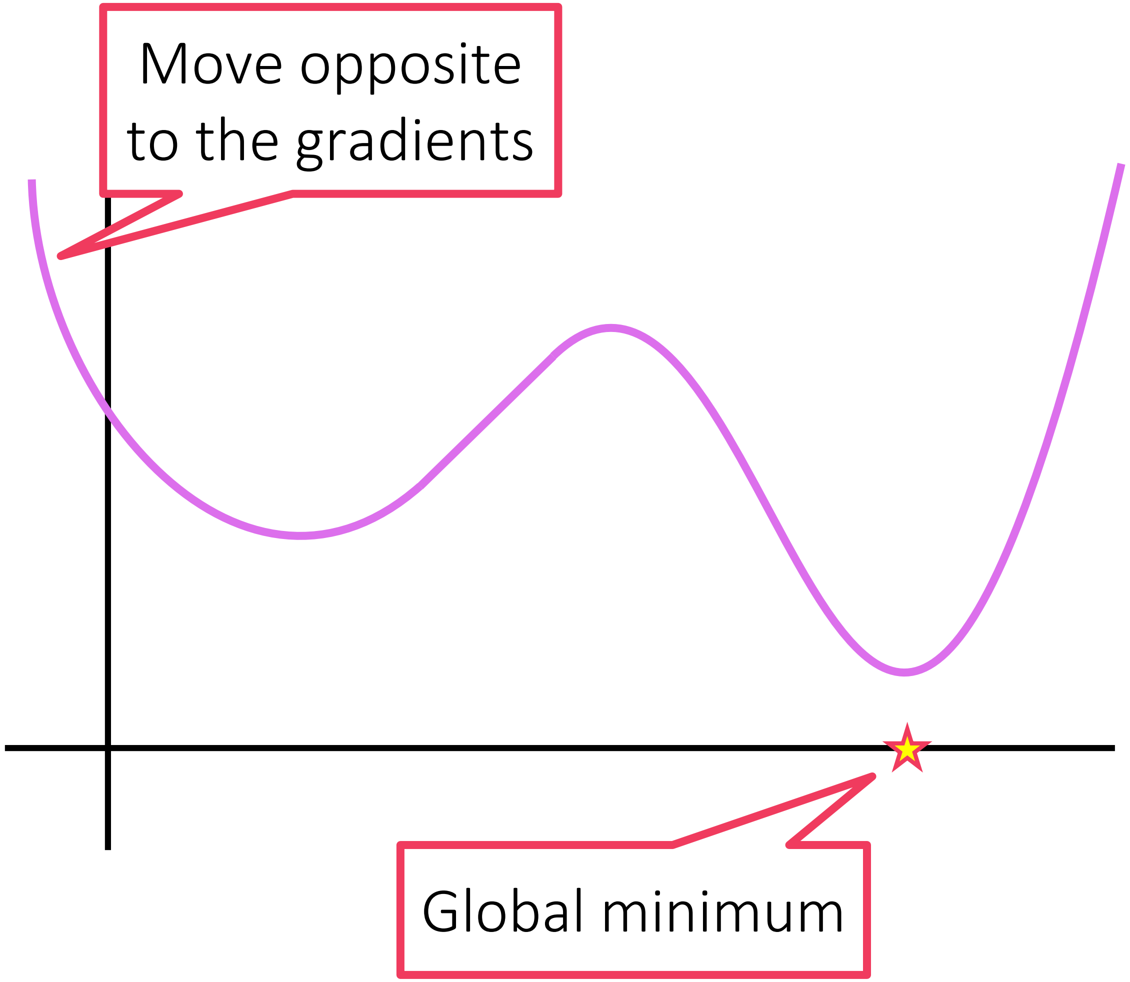
Gradient Descent (GD)



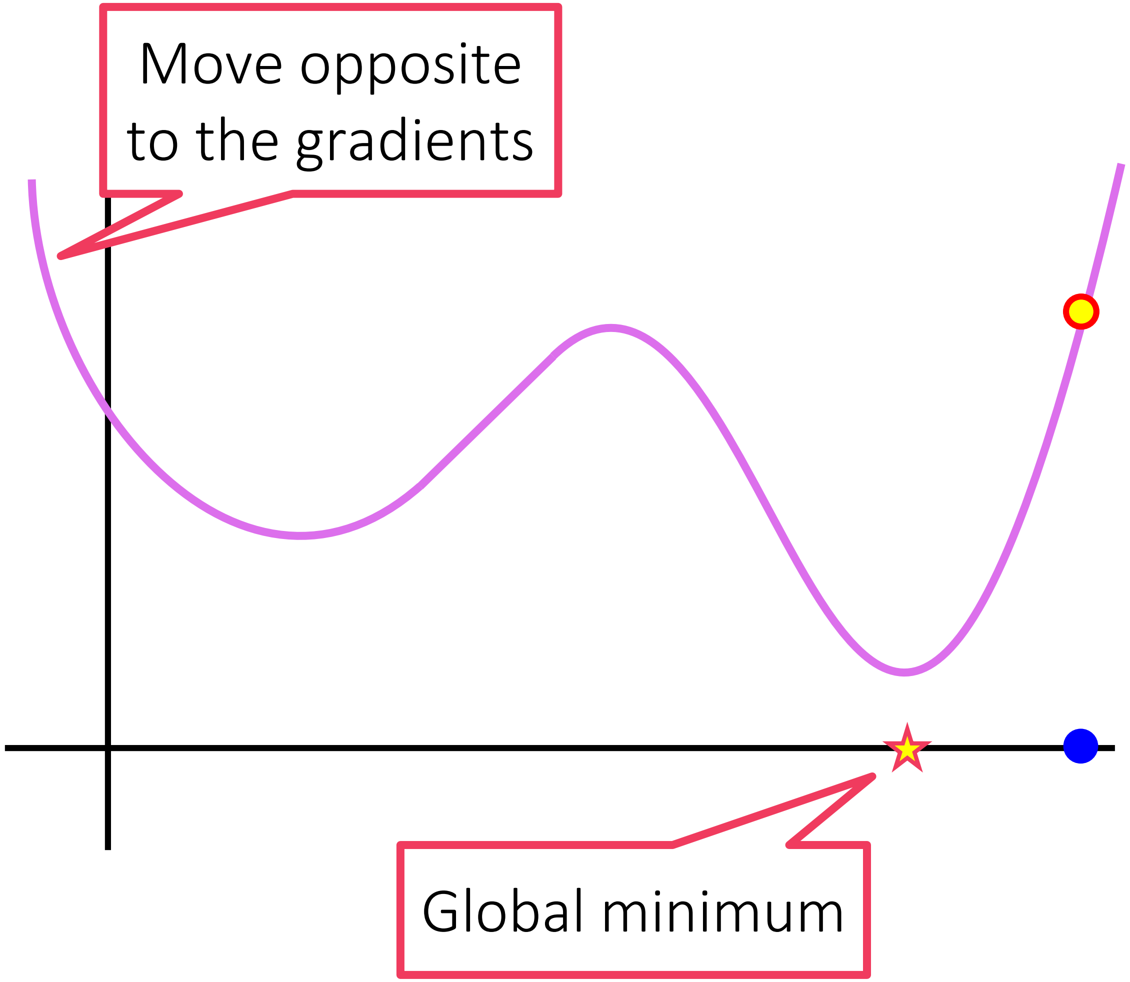
Gradient Descent (GD)



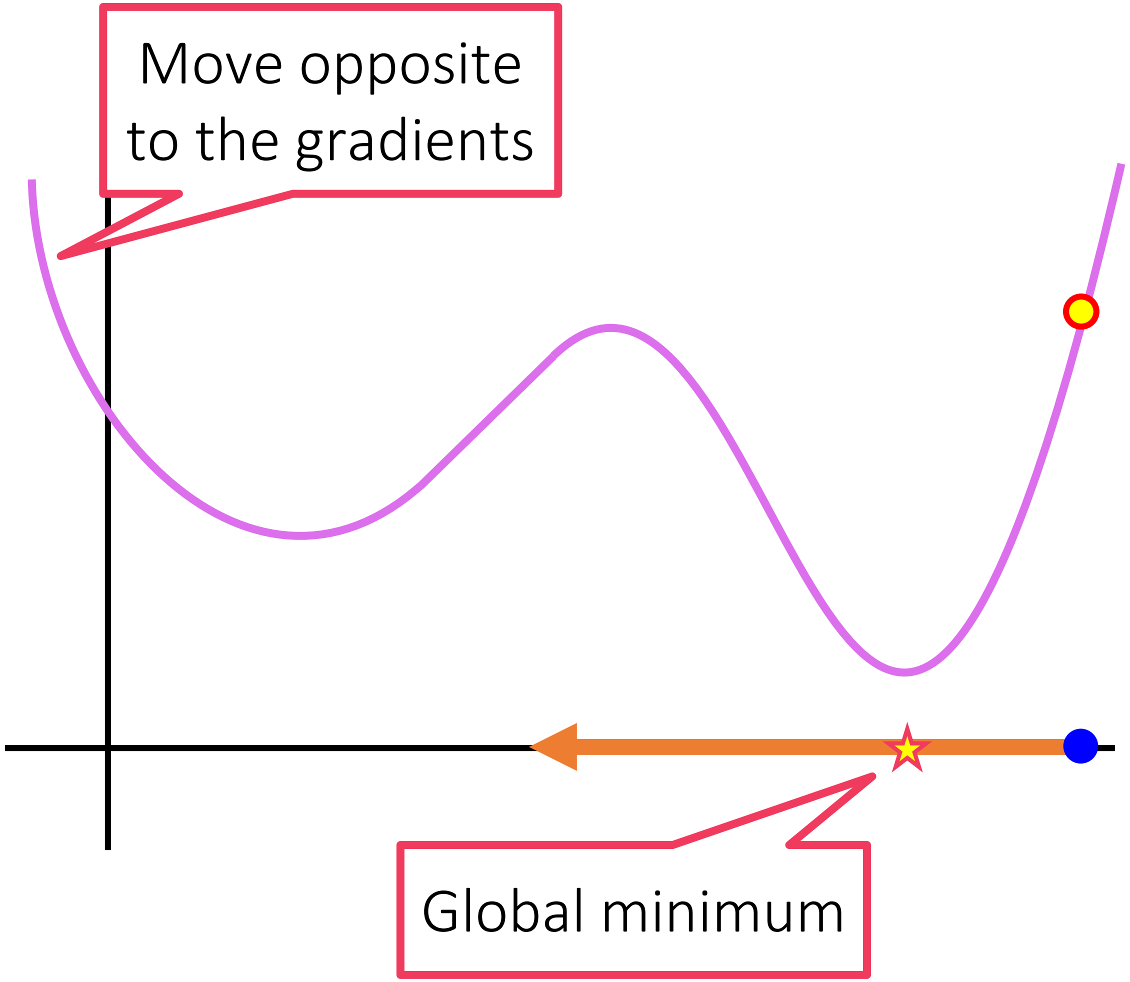
Gradient Descent (GD)



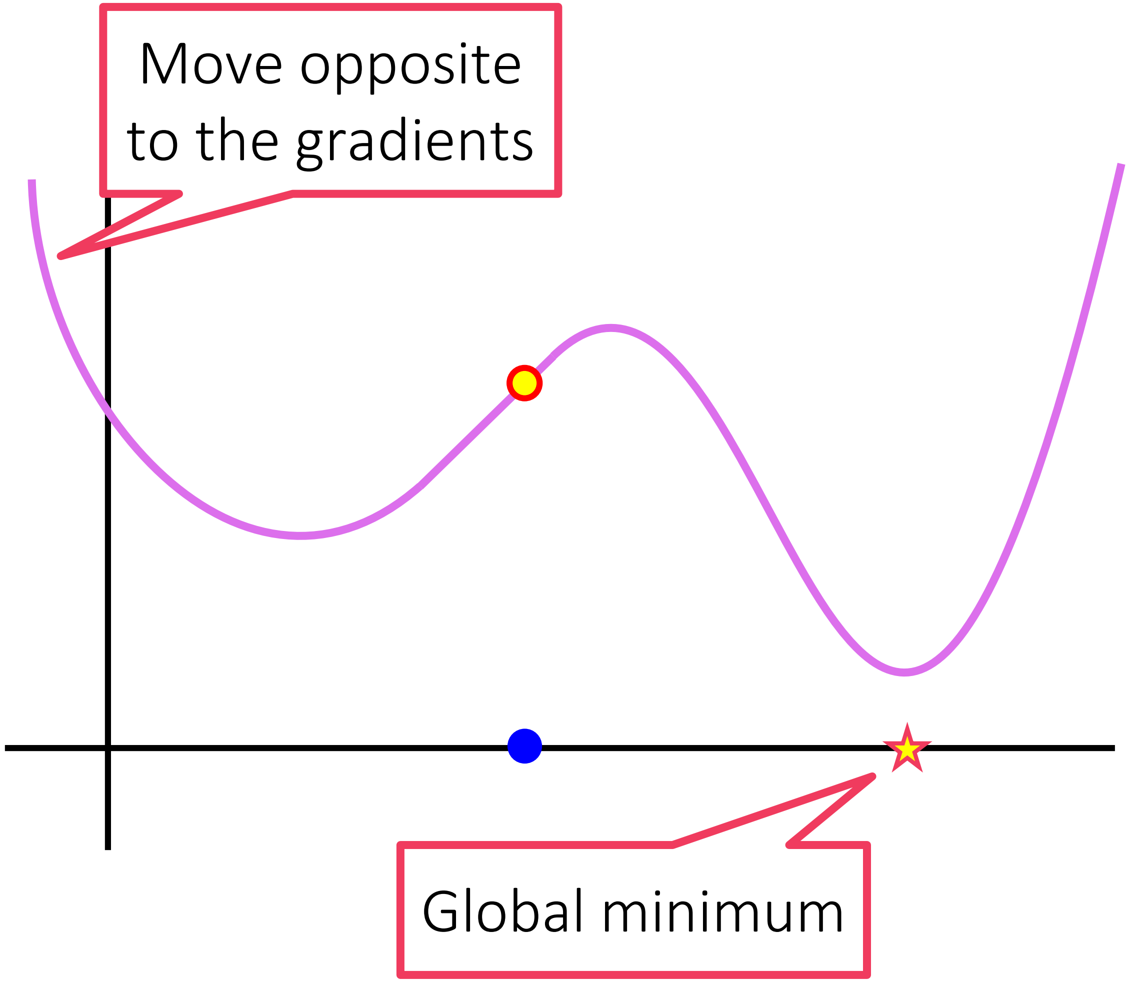
Gradient Descent (GD)



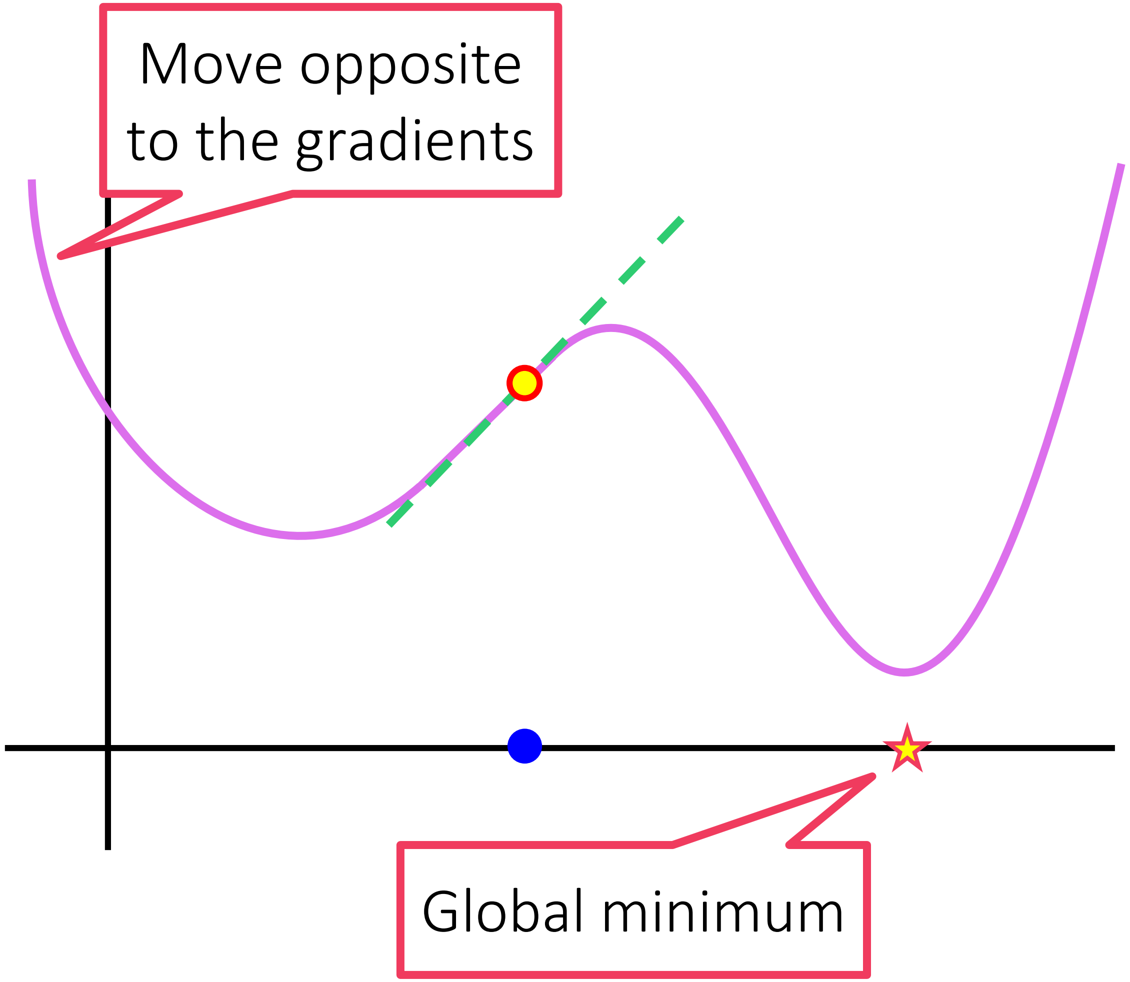
Gradient Descent (GD)



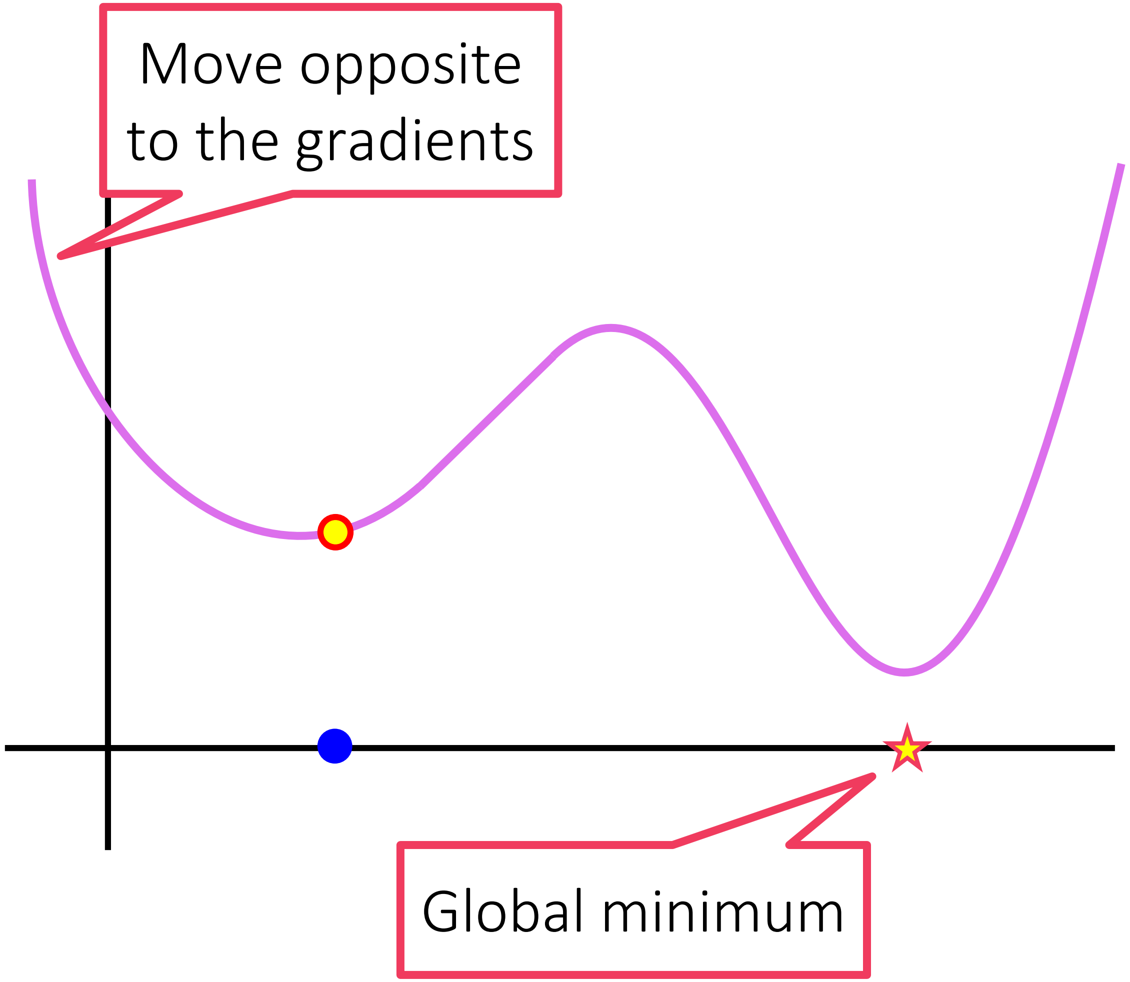
Gradient Descent (GD)



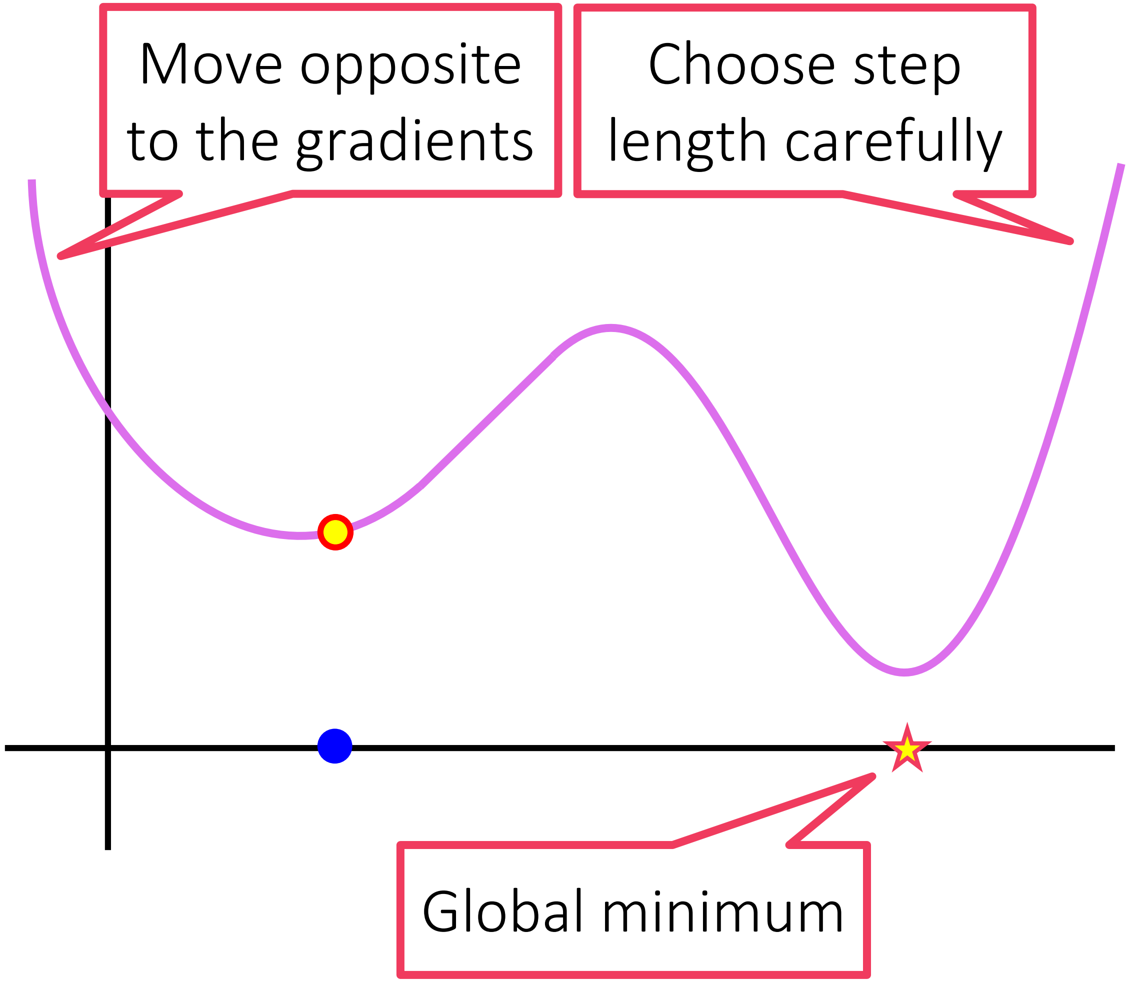
Gradient Descent (GD)



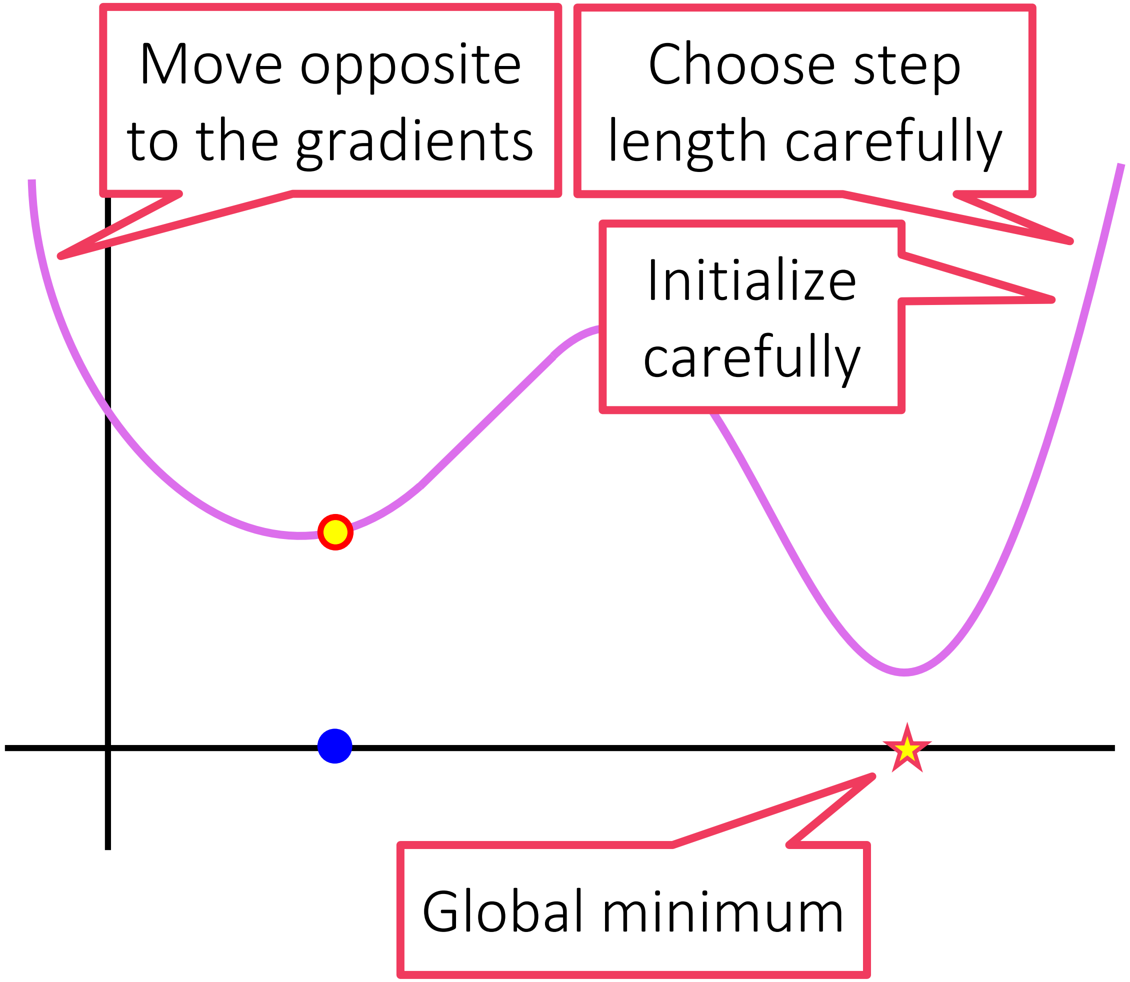
Gradient Descent (GD)



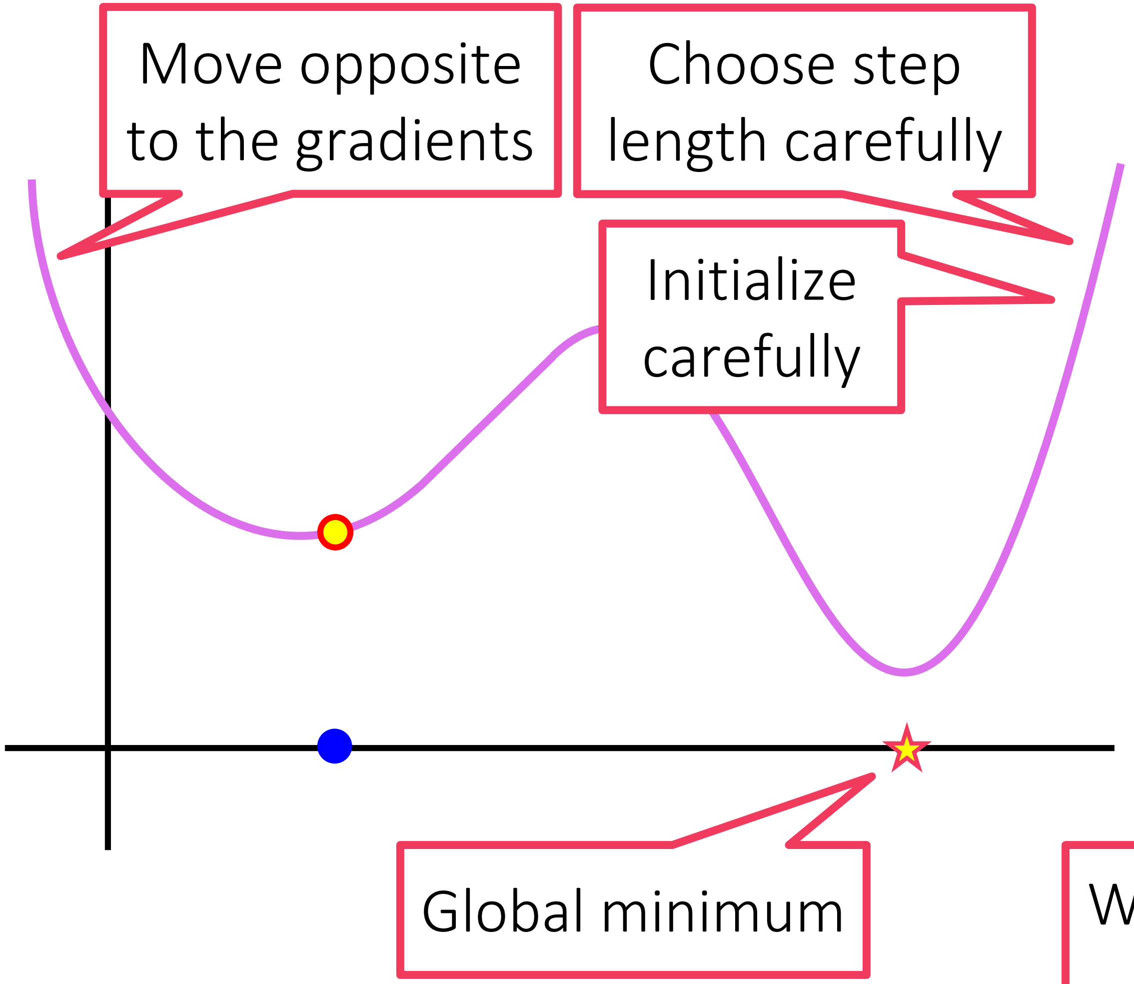
Gradient Descent (GD)

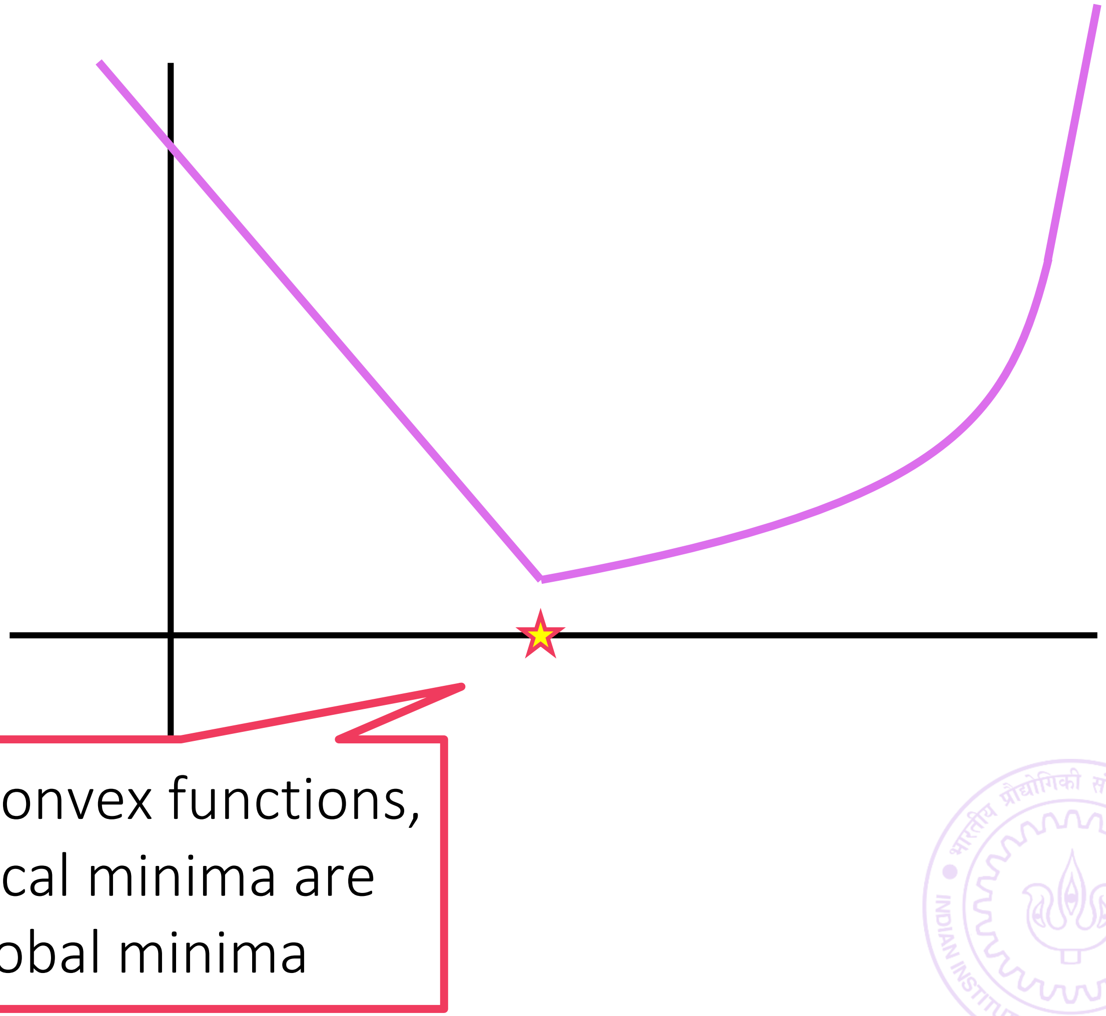


Gradient Descent (GD)

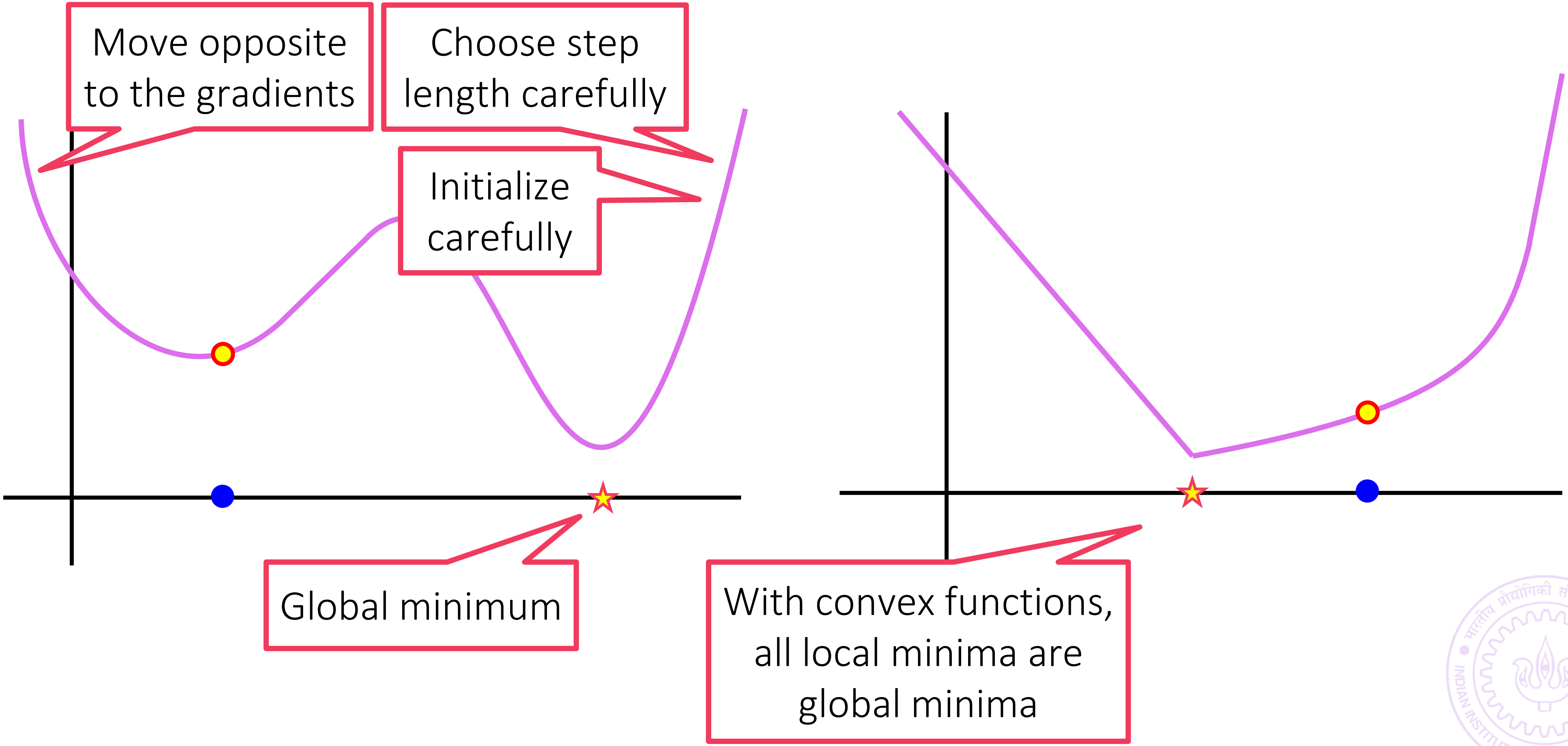


Gradient Descent (GD)

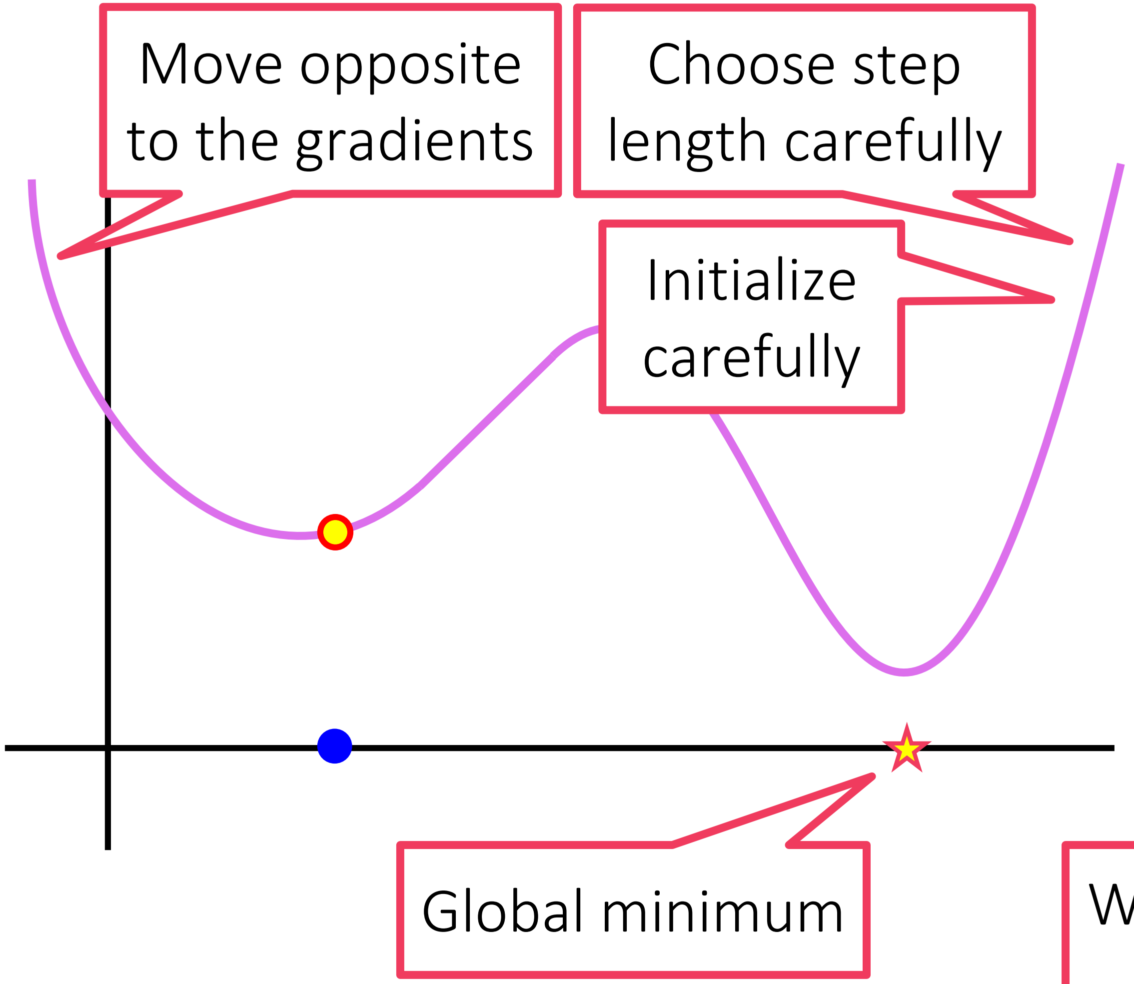


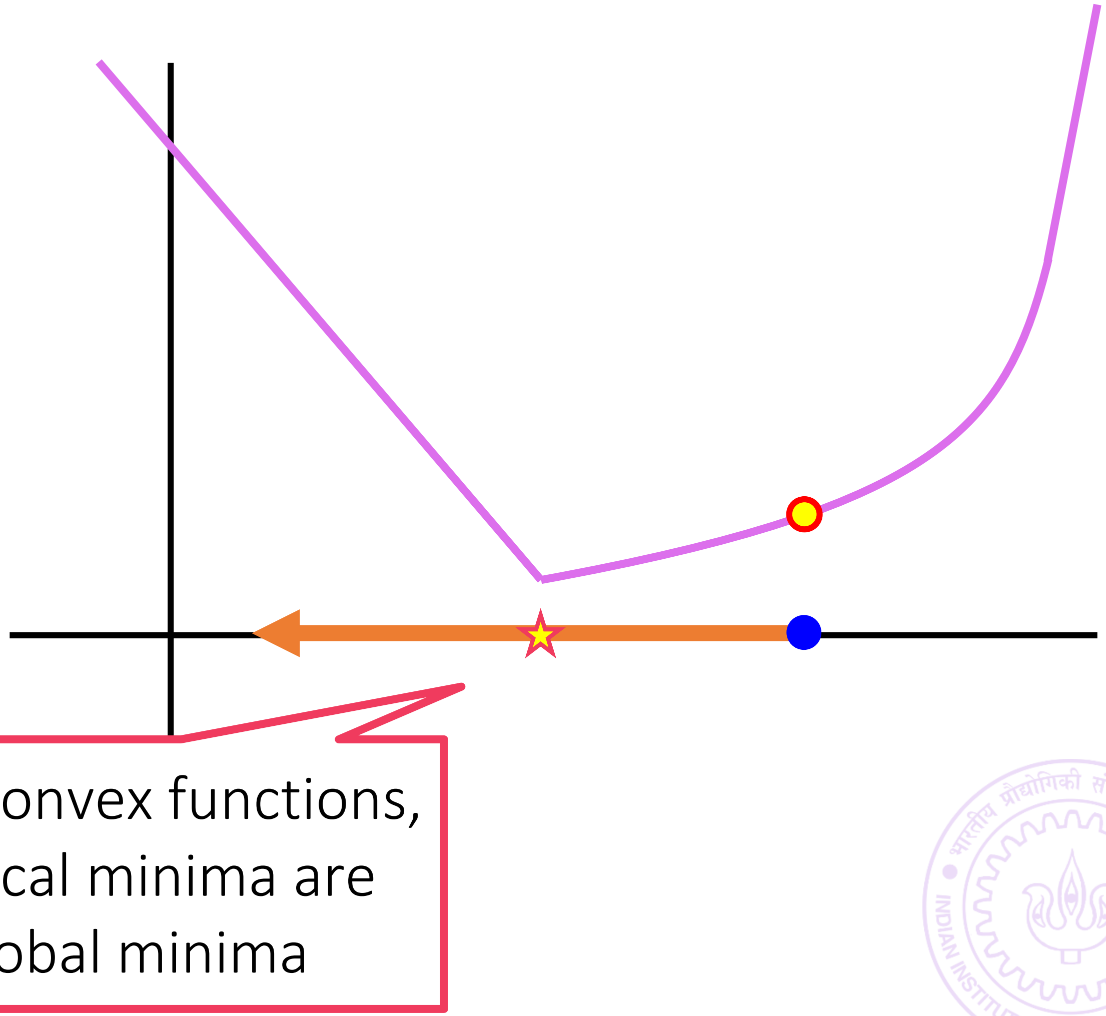


Gradient Descent (GD)

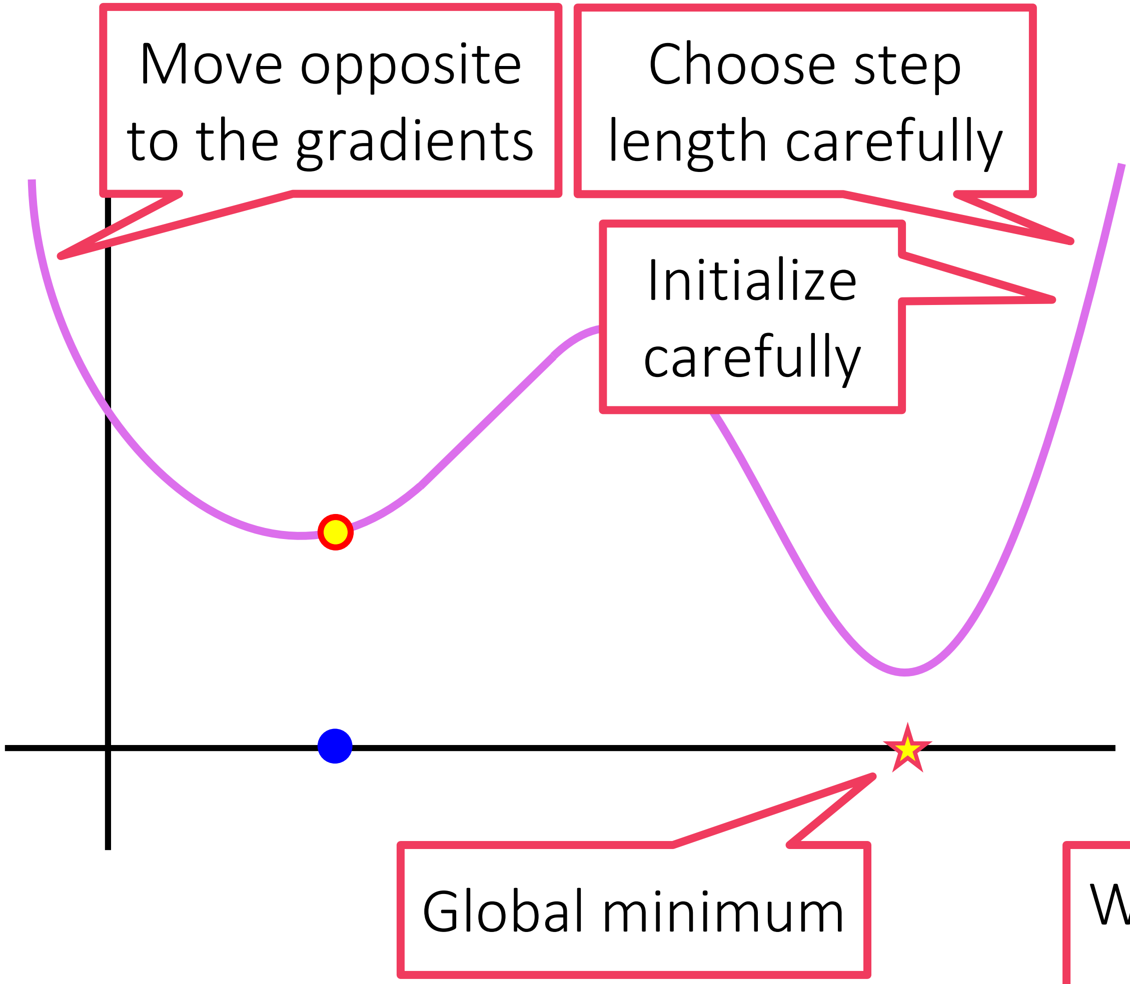


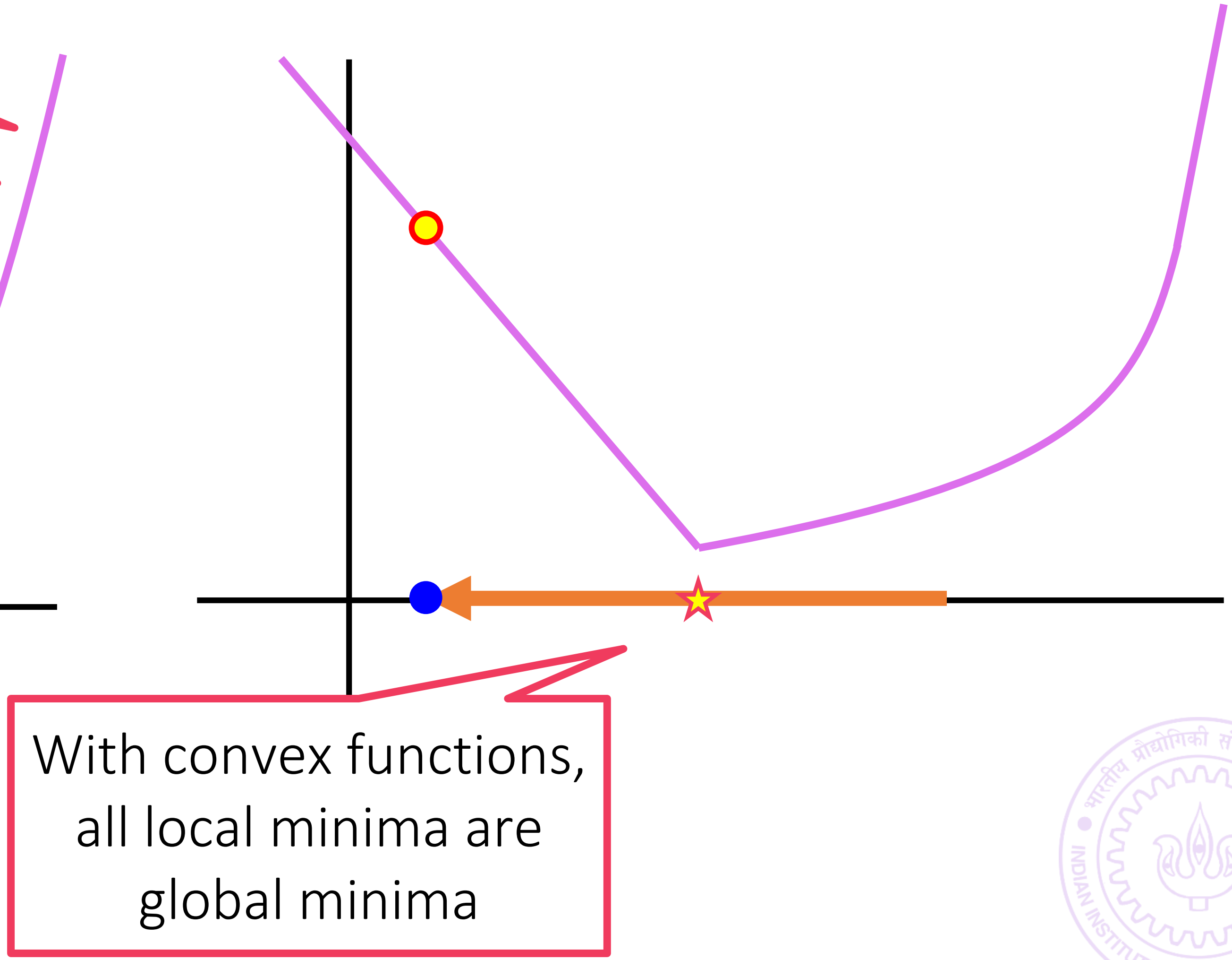
Gradient Descent (GD)



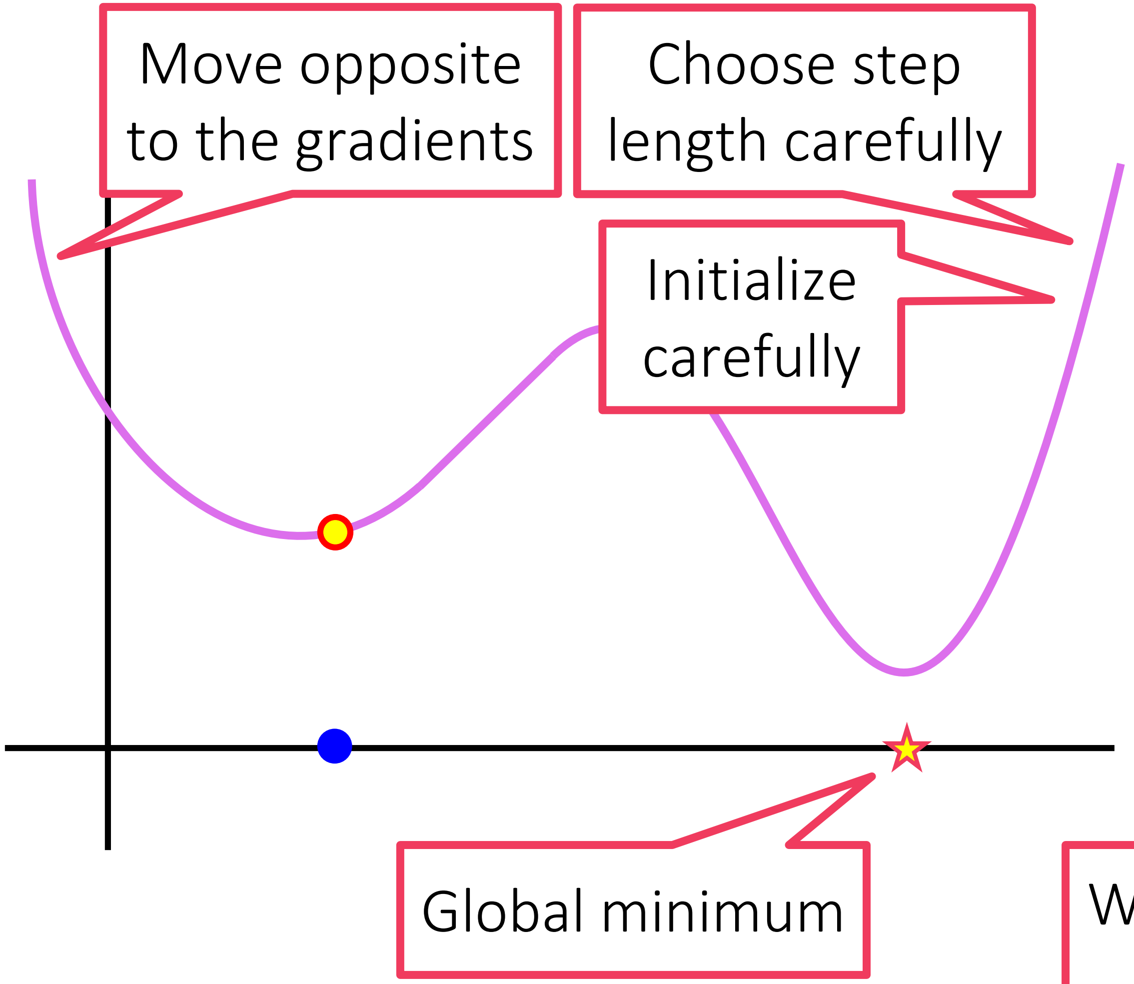


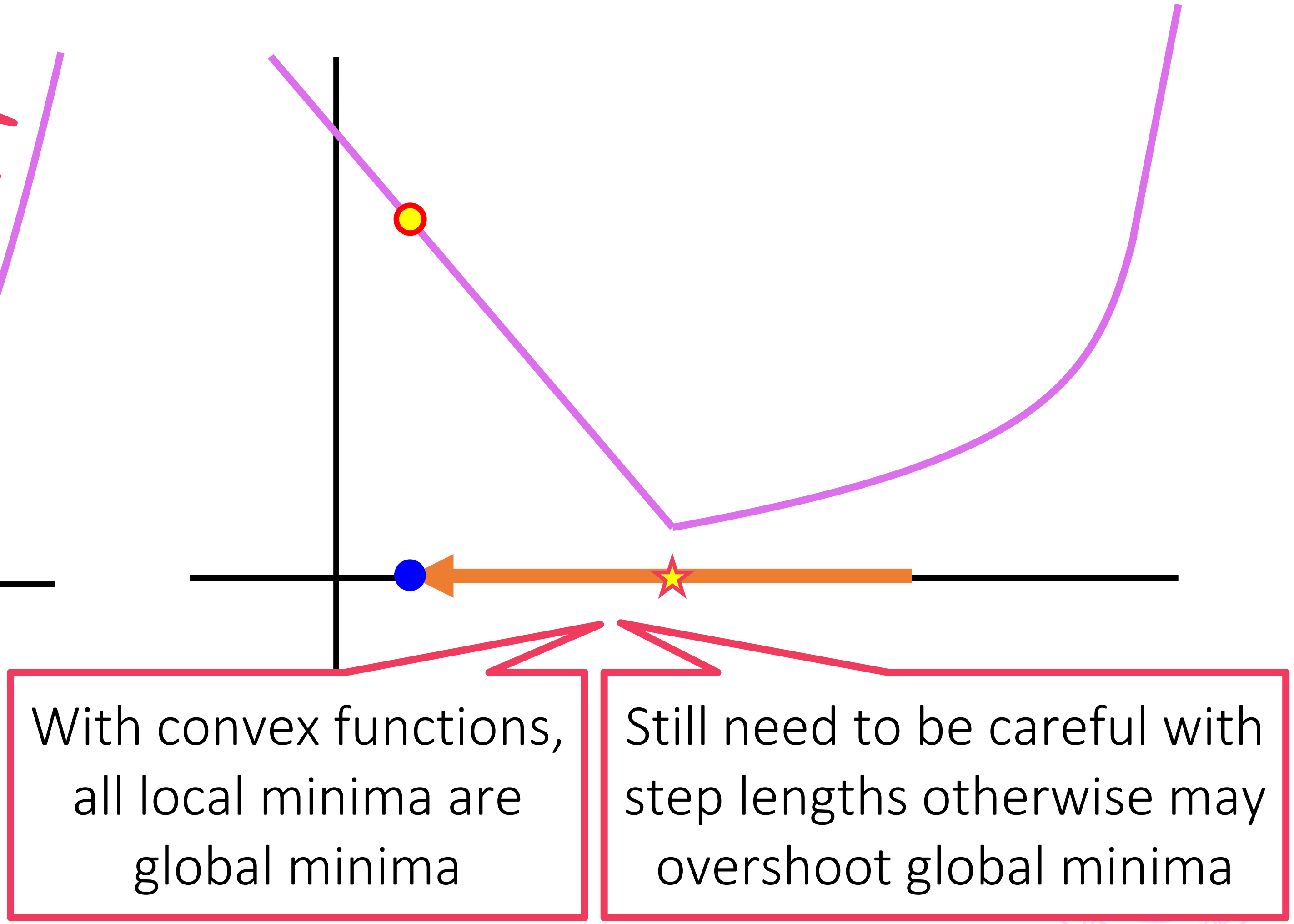
Gradient Descent (GD)





Gradient Descent (GD)





How to choose Step Length?

For "nicely behaved" convex functions, have formulae for step length

Set $\eta_t = \eta/\sqrt{t}$ or else $\eta_t = \eta/t$ where η

These are guaranteed to work for these nice convex functions

Details beyond scope of CS771 (usually a part of CS77X, X = 3,4,7)

For not so well behaved convex functions and non-convex functions, there exist several heuristics – no guarantee they will always work ☹

Armijo Rule: try a value of η_t, if not "nice" reduce η_t and try again

Adagrad: uses a different step length for each dimension of $\mathbf{w}$

η_t replaced with a diagonal matrix E^t i.e. $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - E^t\mathbf{g}^t$

Adam: uses *momentum* methods (essentially infuses previous gradients into the current gradient)

How to decide Convergence?

In optimization, convergence can refer to a couple of things

The algorithm has gotten within a small distance of a global/local optima ("small" depends on application)

The algorithm has stopped making progress e.g. $\|\mathbf{w}^{t+1} - \mathbf{w}^t\| \to \mathbf{0}$

GD stops making progress when it reaches a stationary point i.e. can stop making progress even without having reached a global optimum (e.g. if it has reached a saddle point)

Usually a few heuristics used to decide when to stop executing GD

If $\|\mathbf{g}^t\|_2$ has become too small

If $|f(\mathbf{w}^{t+1}) - f(\mathbf{w}^t)|$ has become too small

If $f(\mathbf{w}^t)$ is small enough that I don't care to reduce it further

The assignment submission deadline is 5 minutes away

How to Initialize?

Initializing close to the global optimum is obviously preferable ☺

For convex functions, bad initialization may mean slow convergence, but if step lengths are nice then GD should converge eventually

For non-convex functions (e.g. while training deepnets), bad initialization may mean getting stuck at a very bad saddle point

Random restarts used to overcome this problem

For some nice non-convex problems, we do know very good ways to provably initialize close to the global optimum (e.g. collaborative filtering in recommendation systems) – details beyond scope of CS771