AI VIETNAM
All-in-One Course

Basic Calculus

Derivative and its Applications

Quang-Vinh Dinh
PhD in Computer Science

Year 2023

Outline

- Derivative Rules
- Activation Functions
- Application to Edge Detection
- Application to Optimization

Activation Functions

Model (Network) Construction

Which activation function?

$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$	2010 $\text{ReLU}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$	2017 $SELU(x) = \begin{cases} \lambda x & \text{if } x > 0 \\ \lambda\alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}$ $\lambda \approx 1.0507$ $\alpha \approx 1.6733$
$\tanh(x) = \frac{2}{1+e^{-2x}} - 1$	2015 $\text{ELU}(x) = \begin{cases} \alpha(e^x - 1) & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$	
2001 $\text{softplus}(x) = \log(1 + e^x)$	2015 $\text{PReLU}(x) = \begin{cases} \alpha x & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$	2017 $swish(x) = x * \frac{1}{1+e^{-x}}$

...

Activation Functions

❖ Sigmoid function

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

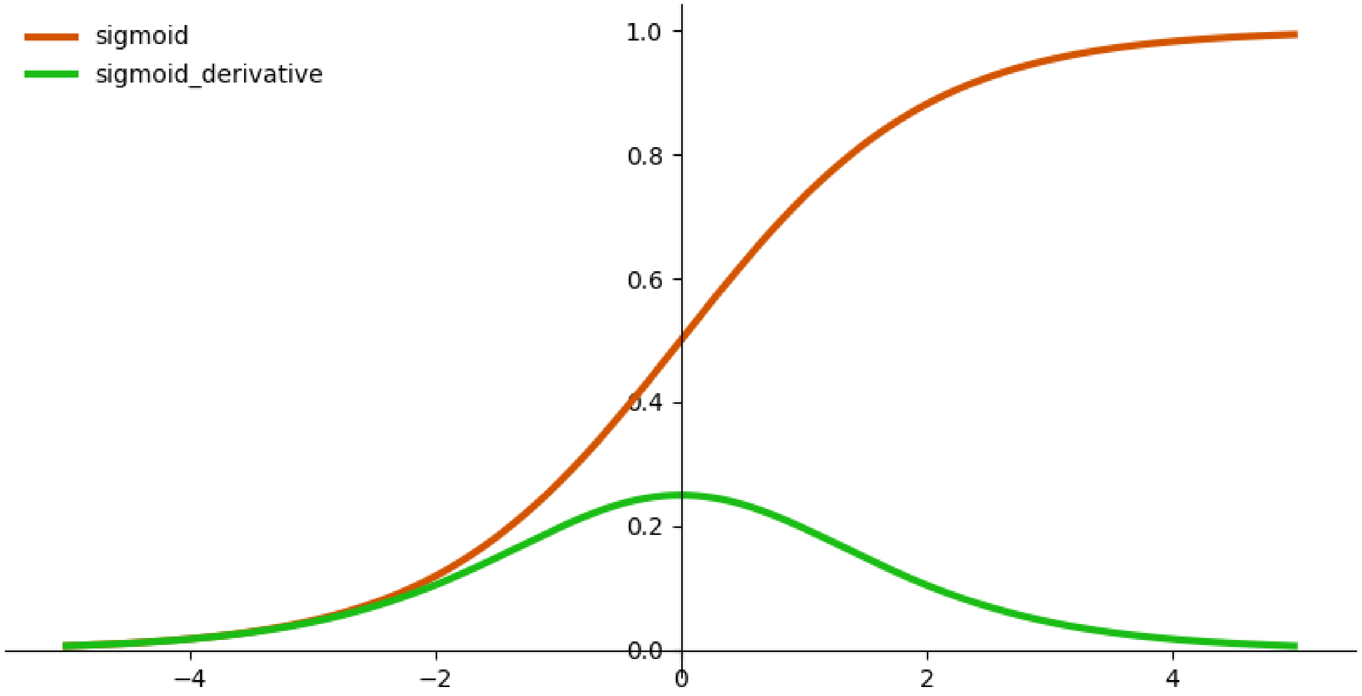
data =

1	5	-4	3	-2

data_a = sigmoid(data)

data_a =

0.731	0.993	0.017	0.95	0.119



$$\text{sigmoid}'(x) = \text{sigmoid}(x)\,(1 - \text{sigmoid}(x))$$

Activation Functions

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$$

data =

1	5	-4	3	-2

data_a = sigmoid(data)

data_a =

0.731	0.993	0.017	0.95	0.119

$$\text{sigmoid}'(x) = \left(\frac{1}{1+e^{-x}}\right)' = \frac{-1}{(1+e^{-x})^2}(-e^{-x})$$

$$= \frac{e^{-x}}{(1+e^{-x})^2} = \frac{e^{-x}+1-1}{(1+e^{-x})^2}$$

$$= \frac{1}{1+e^{-x}} - \frac{1}{(1+e^{-x})^2}$$

$$= \frac{1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right)$$

$$= \text{sigmoid}(x)\,(1 - \text{sigmoid}(x))$$

AI VIETNAM
All-in-One Course

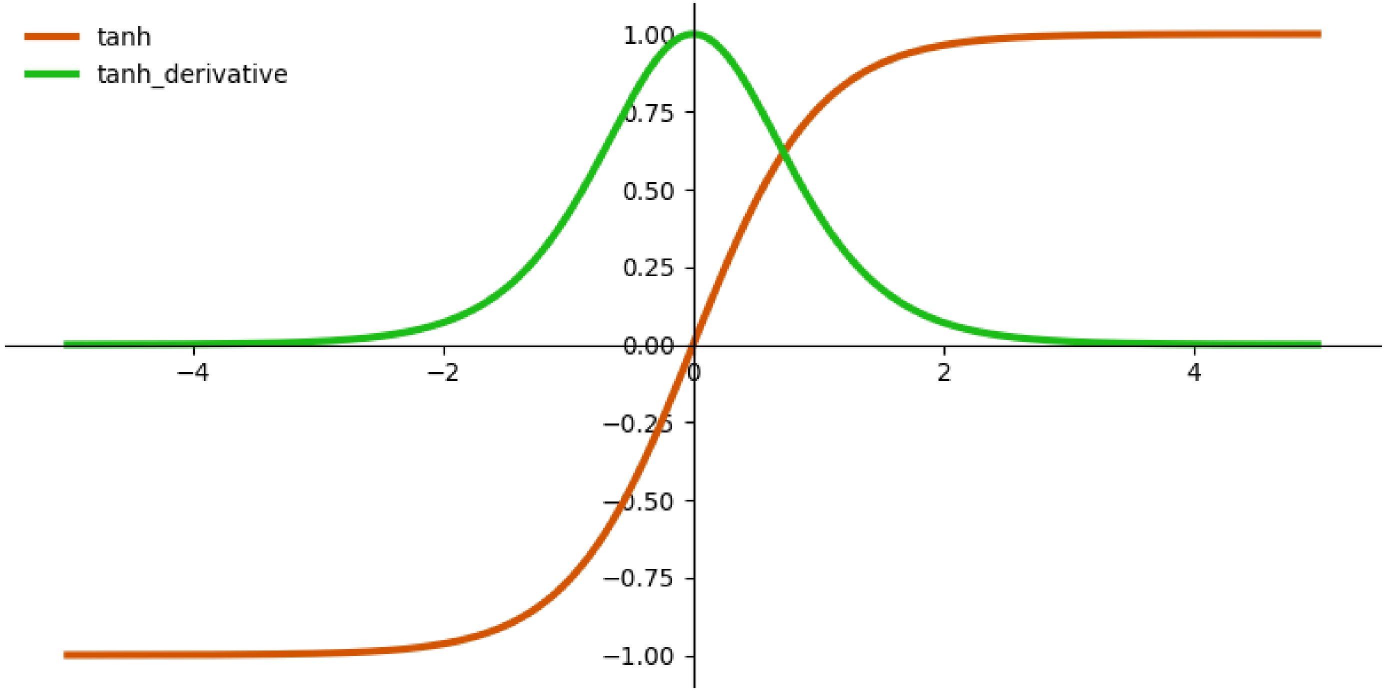
Activation Functions

❖ Tanh function

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$= \frac{2}{1 + e^{-2x}} - 1$$

$$= 1 - \frac{2}{e^{2x} + 1}$$



data =

1	5	-4	3	-2

data_a = tanh(data)

data_a =

0.761	0.999	-0.999	0.995	-0.964

$$tanh'(x) = 1 - tanh^2(x)$$

Activation Functions

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 1 - \frac{2}{e^{2x} + 1} = \frac{2}{e^{-2x} + 1} - 1$$

$$\tanh'(x) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)' = \frac{(e^x + e^{-x})(e^x + e^{-x}) - (e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2}$$

$$= \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2}$$

$$= 1 - \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)^2 = 1 - tanh^2(x)$$

Activation Functions

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = 1 - \frac{2}{e^{2x} + 1} = \frac{2}{e^{-2x} + 1} - 1$$

$$tanh'(x) = \left(\frac{2}{e^{-2x} + 1} - 1\right)' = \frac{4e^{-2x}}{(e^{-2x} + 1)^2} = 4\left(\frac{e^{-2x} + 1 - 1}{(e^{-2x} + 1)^2}\right)$$

$$= 4\left(\frac{1}{e^{-2x} + 1} - \frac{1}{(e^{-2x} + 1)^2}\right) = -\left(\frac{4}{(e^{-2x} + 1)^2} - \frac{4}{e^{-2x} + 1}\right)$$

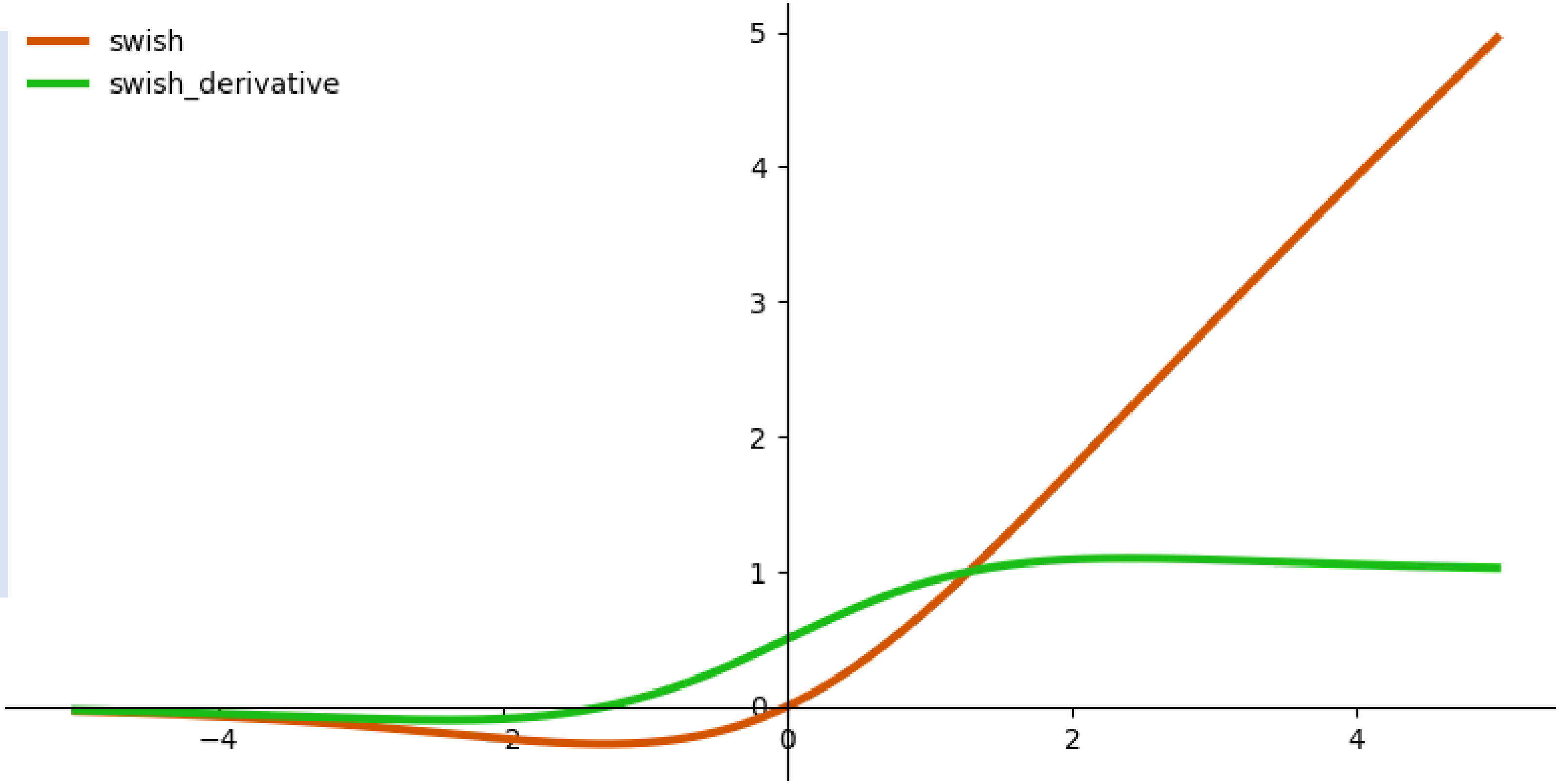
$$= -\left(\frac{4}{(e^{-2x} + 1)^2} - \frac{4}{e^{-2x} + 1} + 1 - 1\right) = 1 - \left(\frac{2}{e^{-2x} + 1} - 1\right)^2 = 1 - tanh^2(x)$$

Activation Functions

❖ Swish

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$swish(x) = \frac{x}{1 + e^{-x}} = x\,\sigma(x)$$



Activation Functions

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

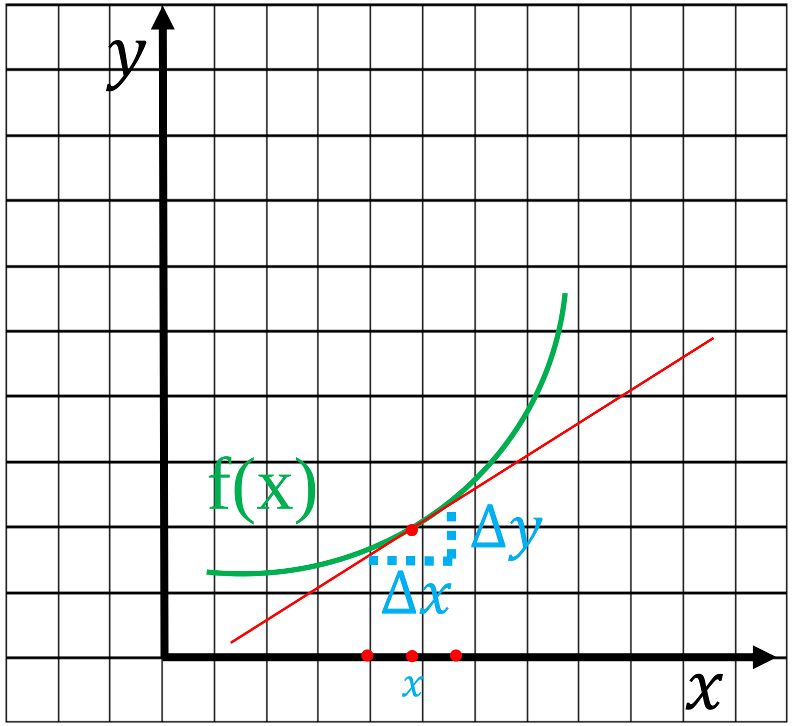
$$swish(x) = \frac{x}{1+e^{-x}} = x\,\sigma(x)$$

$$\begin{aligned}
swish'(x) &= (x\,\sigma(x))' = (x)'\,\sigma(x) + x(\sigma(x))' \\
&= \sigma(x) + x\,\sigma(x)\,(1-\sigma(x)) \\
&= \sigma(x) + x\,\sigma(x) - x\,\sigma(x)^2 \\
&= x\,\sigma(x) + \sigma(x)(1 - x\,\sigma(x)) \\
&= swish(x) + \sigma(x)\,(1 - swish(x))
\end{aligned}$$

Outline

- Derivative Rules
- Activation Functions
- Application to Edge Detection
- Application to Optimization

Áp dụng cho hàm rời rạc



Đạo hàm $= \dfrac{Thay\ đổi\ theo\ y}{Thay\ đổi\ theo\ x} = \dfrac{\Delta y}{\Delta x}$

$$\frac{d}{dx} f(x) = \lim_{\Delta x \to 0} \frac{f\left(x + \frac{\Delta x}{2}\right) - f\left(x - \frac{\Delta x}{2}\right)}{\Delta x}$$

f(x) =

32	30	45	36	160	156	155	170

x

$\Delta x = 2$

$$\frac{d}{dx} f(x) = \frac{f(x+1) - f(x-1)}{2} = \frac{156 - 36}{2} = 60$$

-1	0	1

x-derivative filter

*just ignore $\frac{1}{2}$

Derivative and Applications

❖ Grayscale images

Consider each row a 1D function

Compute derivative for each position with $\Delta x = 1$

f(x)

0	1	2	3	4	5	6	7	8	9
230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97

Input Image

Derivative and Applications

❖ Implementation

Discussion

f(x)

0	1	2	3	4	5	6	7	8	9
230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97

Input Image

Consider each row a 1D function

Compute derivative for each position with $\Delta x = 1$

How to visualize derivative values?

```python
def derivative_x(data, height, width):
    result = [[0]*width for _ in range(height)]

    # get rows and compute derivative
    for i in range(height):

        # for each row
        for j in range(width-1):
            d_value = data[i][j+1] - data[i][j]
            result[i][j] = d_value

    return result
```

Derivative and Applications

❖ Using magnitude

f(x)

0	1	2	3	4	5	6	7	8	9
230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97

Input Image

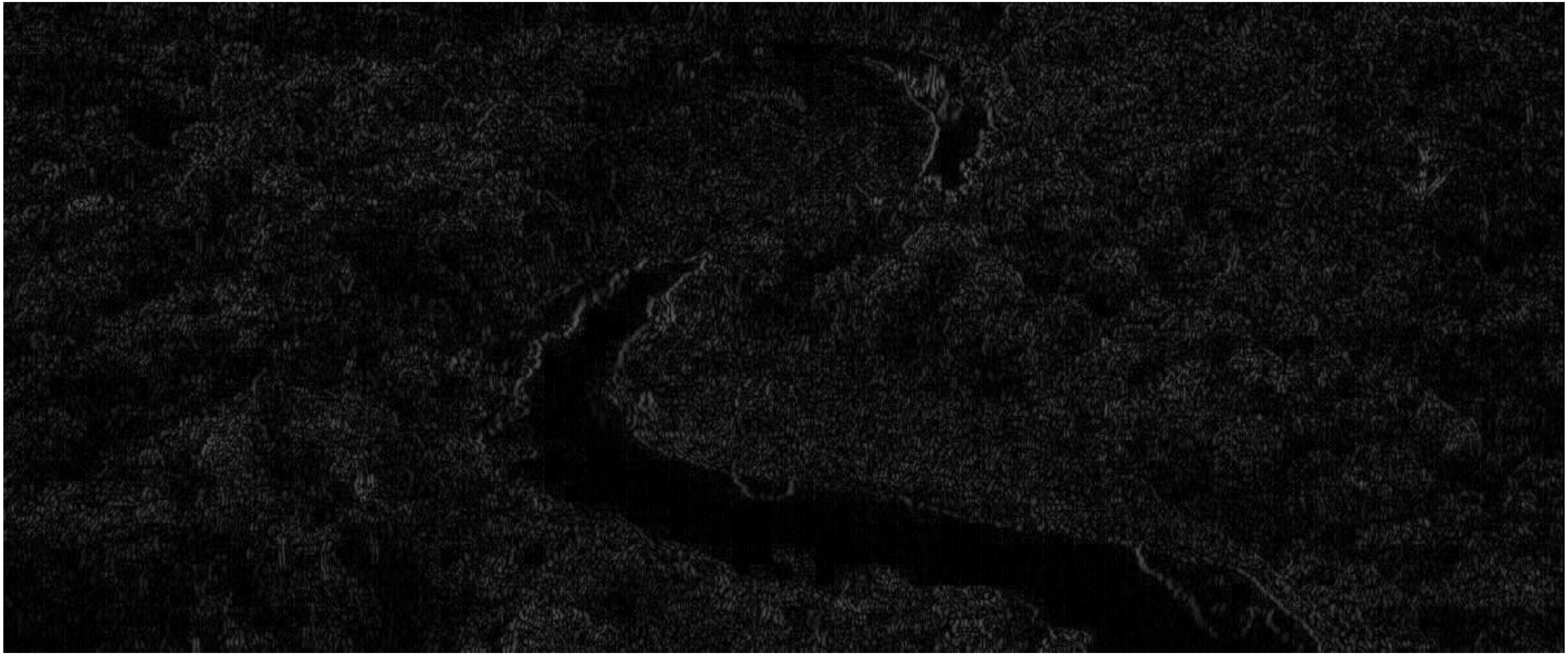
Consider each row a 1D function

Compute derivative for each position with $\Delta x = 1$

```python
def derivative_x(data, height, width):
    result = [[0]*width for _ in range(height)]

    # get rows and compute derivative
    for i in range(height):

        # for each row
        for j in range(width-1):
            d_value = data[i][j+1] - data[i][j]

            # d_value can be positive or negative
            # process d_value to adapt to an image
            result[i][j] = abs(d_value)

    return result
```

Derivative and Applications

❖ Using magnitude: Result



Derivative

❖ Shift values

f(x)

0	1	2	3	4	5	6	7	8	9
230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97

Input Image

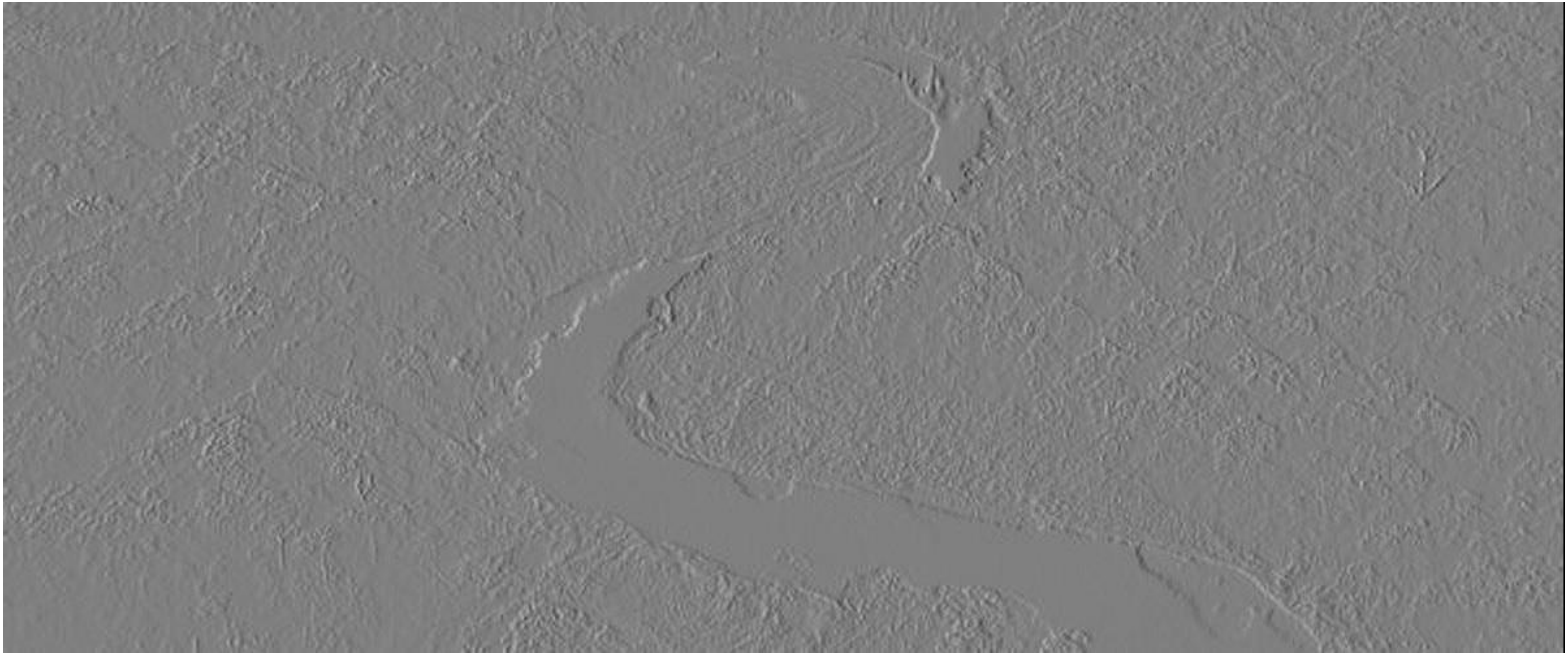
Consider each row a 1D function

Compute derivative for each position with $\Delta x = 1$

```python
def derivative_x(data, height, width):
    result = [[0]*width for _ in range(height)]

    # get rows and compute derivative
    for i in range(height):

        # for each row
        for j in range(width-1):
            d_value = data[i][j+1] - data[i][j]

            # d_value can be positive or negative
            # process d_value to adapt to an image
            d_value = d_value + 127.5
            d_value = max(d_value, 0)
            d_value = min(d_value, 255)
            result[i][j] = d_value

    return result
```

Derivative and Applications

❖ Shift values: Result



Derivative

❖ Central derivative

	0	1	2	3	4	5	6	7	8	9
f(x)	230	194	147	108	90	98	84	96	91	101
	237	206	188	195	207	213	163	123	116	128
	210	183	180	205	224	234	188	122	134	147
	198	189	201	227	229	232	200	125	127	135
	249	241	237	244	232	226	202	116	125	126
	251	254	241	239	230	217	196	102	103	99
	243	255	240	231	227	214	203	116	95	91
	204	231	208	200	207	201	200	121	95	95
	144	140	120	115	125	127	143	118	92	91
	121	121	108	109	122	121	134	106	86	97

Input Image

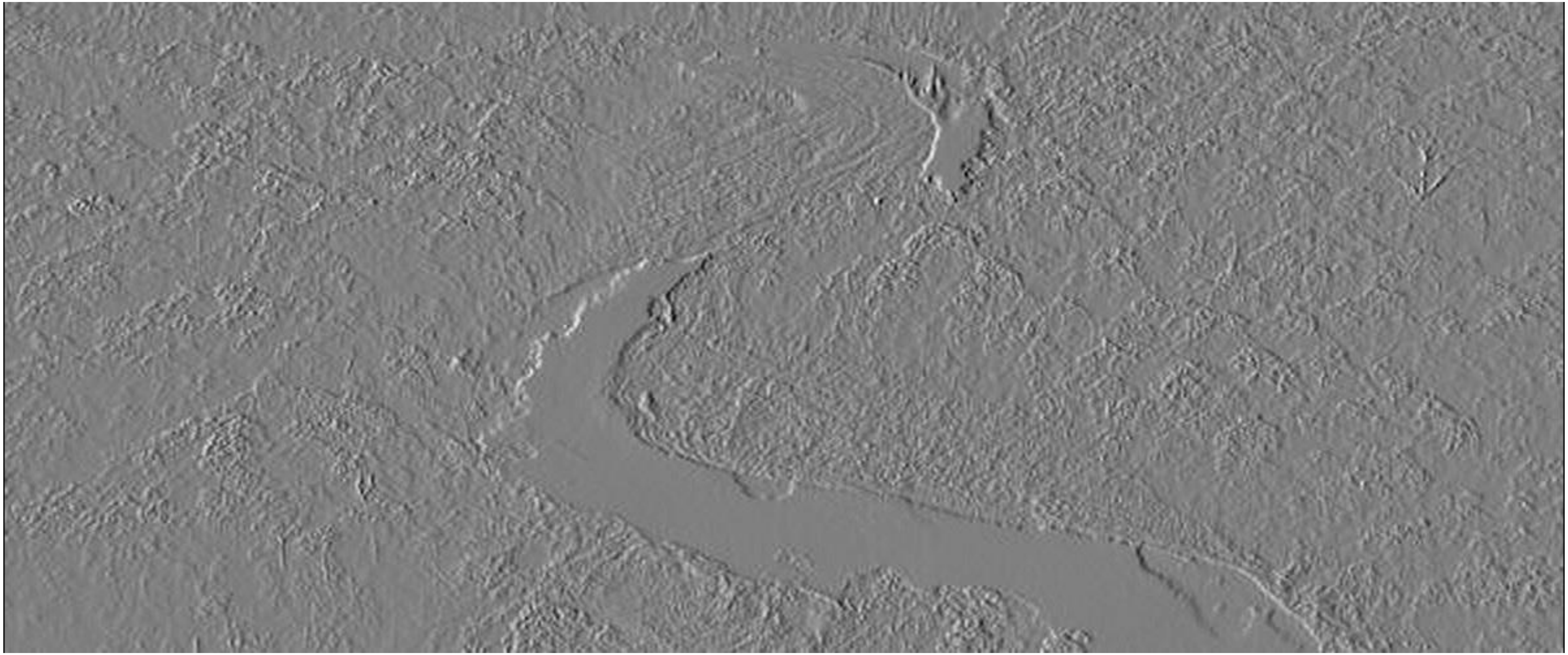
Consider each row a 1D function

Compute derivative for each position with $\Delta x = 2$

```python
def derivative_x(data, height, width):
    result = [[0]*width for _ in range(height)]

    # get rows and compute derivative
    for i in range(height):

        # for each row
        for j in range(1, width-1):
            d_value = data[i][j+1] - data[i][j-1]

            # d_value can be positive or negative
            # process d_value to adapt to an image
            d_value = d_value + 127.5
            d_value = max(d_value, 0)
            d_value = min(d_value, 255)
            result[i][j] = d_value

    return result
```

Derivative and Applications

❖ Central derivative: Result



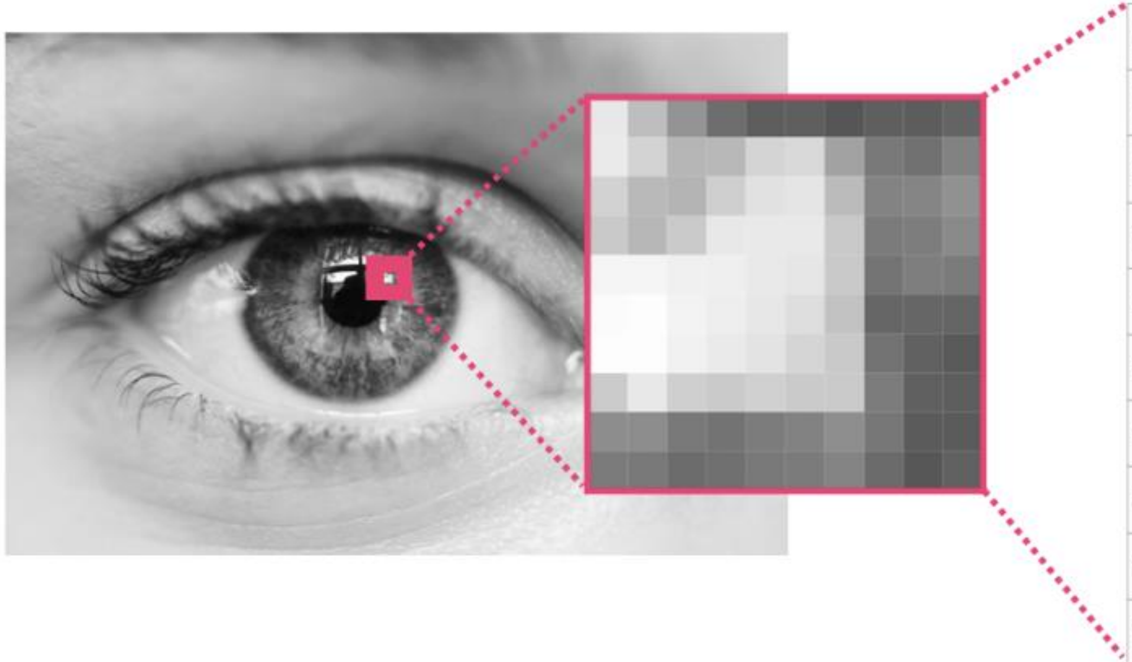
Derivative and Applications

Grayscale images

(Height, Width)

Pixel p = scalar

$0 \leq p \leq 255$



230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97

Tính đạo hàm trung bình theo hướng x

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

x-derivative

weighted average

Sobel for x direction

Tính đạo hàm trung bình theo hướng y

$$\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

weighted average

y-derivative

Sobel for y direction

Derivative and Applications

❖ Grayscale images

$a = -230 + 147 - 2 \times 237 + 2 \times 188 - 210 + 180 = -211$

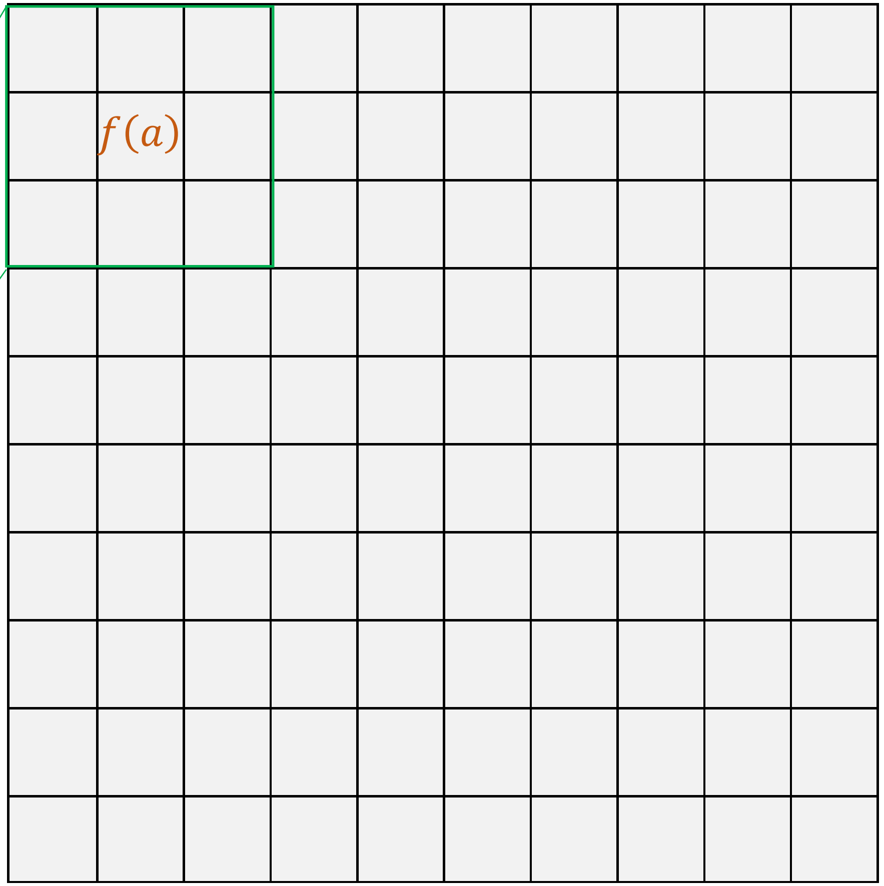
$f(a) = |a| = 211$

230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97

Input Image

-1	0	1
-2	0	2
-1	0	1

Sobel for x direction

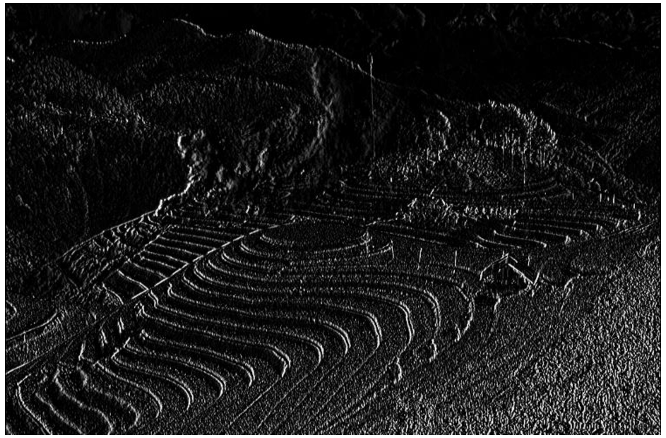


Derivative and Applications

Ứng dụng đạo hàm cho edge detection

Edge detection

Sobel-X



Sobel-Y

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I$$

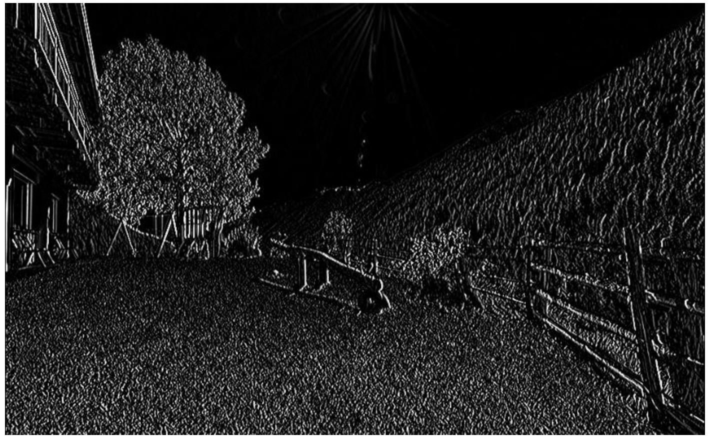
$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I$$

Derivative and Applications

Ứng dụng đạo hàm cho edge detection

Edge detection

Sobel-X



Sobel-Y

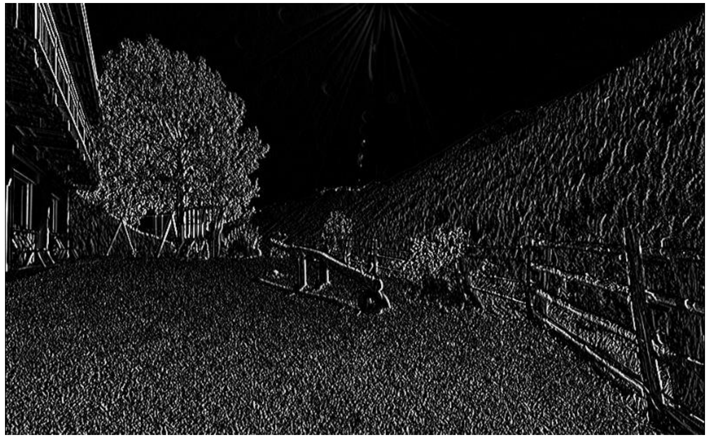
$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I$$

Derivative and Applications

❖ Code

Sobel-X



Sobel-Y

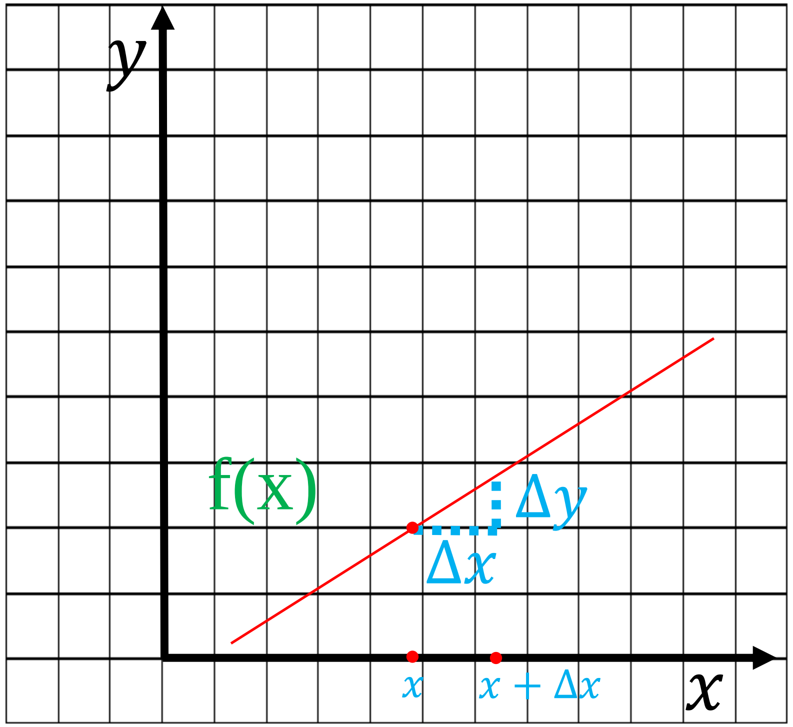
```python
import numpy as np
import cv2

# read image and convert to grayscale
img1 = cv2.imread('vn.jpeg', 0)

# compute sobel-x
sobelx = cv2.Sobel(img1, cv2.CV_64F,1,0)

# compute sobel-y
sobely = cv2.Sobel(img1, cv2.CV_64F,0,1)

# save results
cv2.imwrite('vn_edge_x.jpg', sobelx)
cv2.imwrite('vn_edge_y.jpg', sobely)
```

Outline

- Derivative Rules
- Activation Functions
- Application to Edge Detection
- Application to Optimization

Derivative and Applications

Đạo hàm cho hàm liên tục



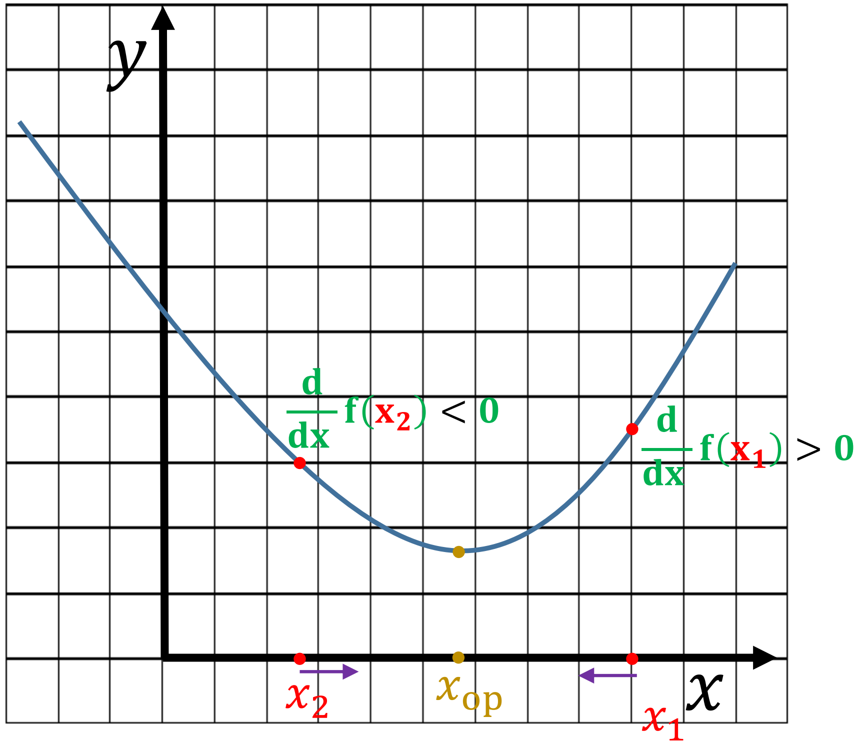
$$\text{Đạo hàm} = \frac{Thay\ đổi\ theo\ y}{Thay\ đổi\ theo\ x} = \frac{\Delta y}{\Delta x}$$

$$\frac{d}{dx} f(x) = \underbrace{\lim_{\Delta x \to 0}} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Δx cần tiến về 0 để đường tiếp tuyến tiến về hàm f(x) trong vùng lân cận tại x

Derivative and Applications

Tìm giá trị min



Quan sát: x_{op} ở vị trí ngược hướng đạo hàm tại $\mathbf{x_1}$ và $\mathbf{x_2}$

Cách xử lý việc di chuyển ngược hướng đạo hàm cho $\mathbf{x_1}$ và $\mathbf{x_2}$ (để tìm x_{op}) khác nhau hình thành các thuật toán tối ưu hóa khác nhau

Cách cập nhật giá trị x đơn giản

$$\mathbf{x} = \mathbf{x} - \eta \frac{\mathbf{d}}{\mathbf{dx}}\mathbf{f}(\mathbf{x})$$

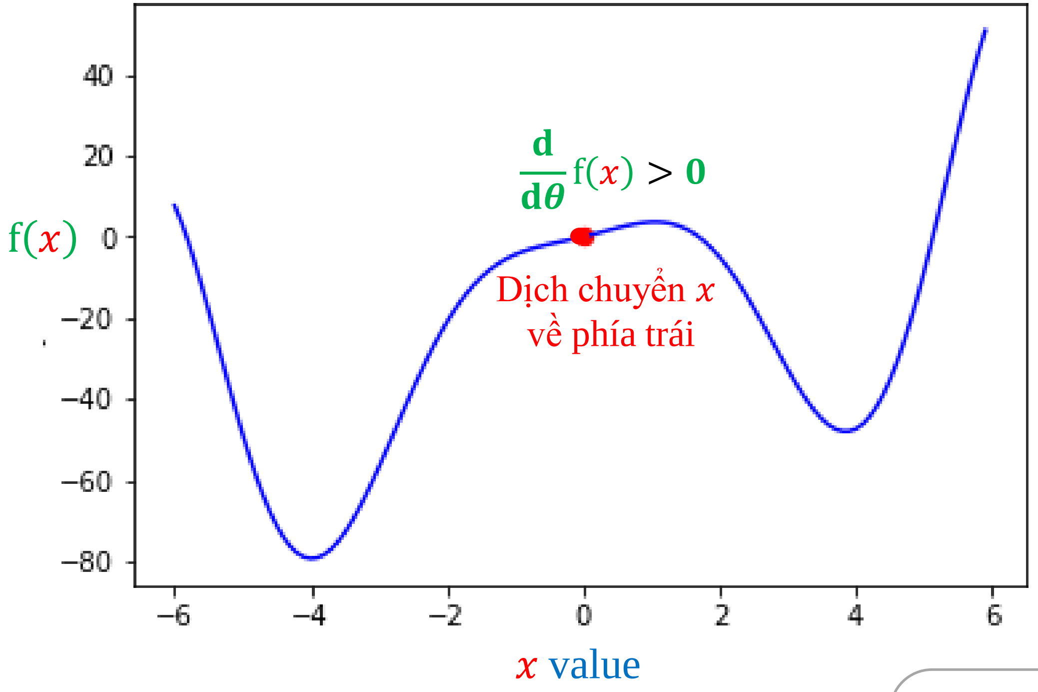
Đạo hàm tại x

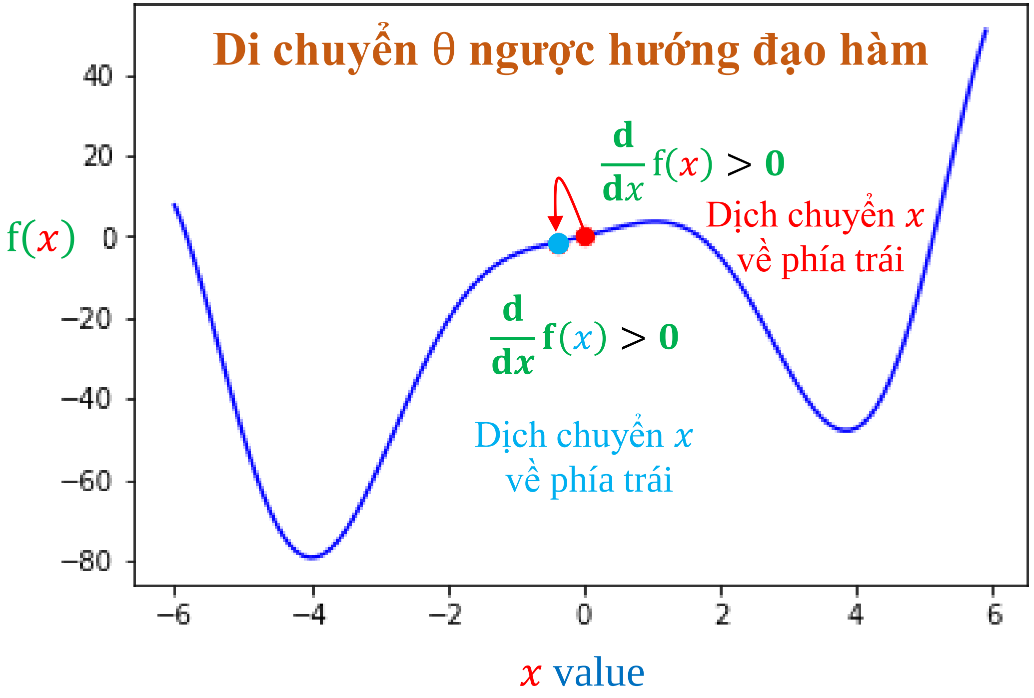
Trọng số

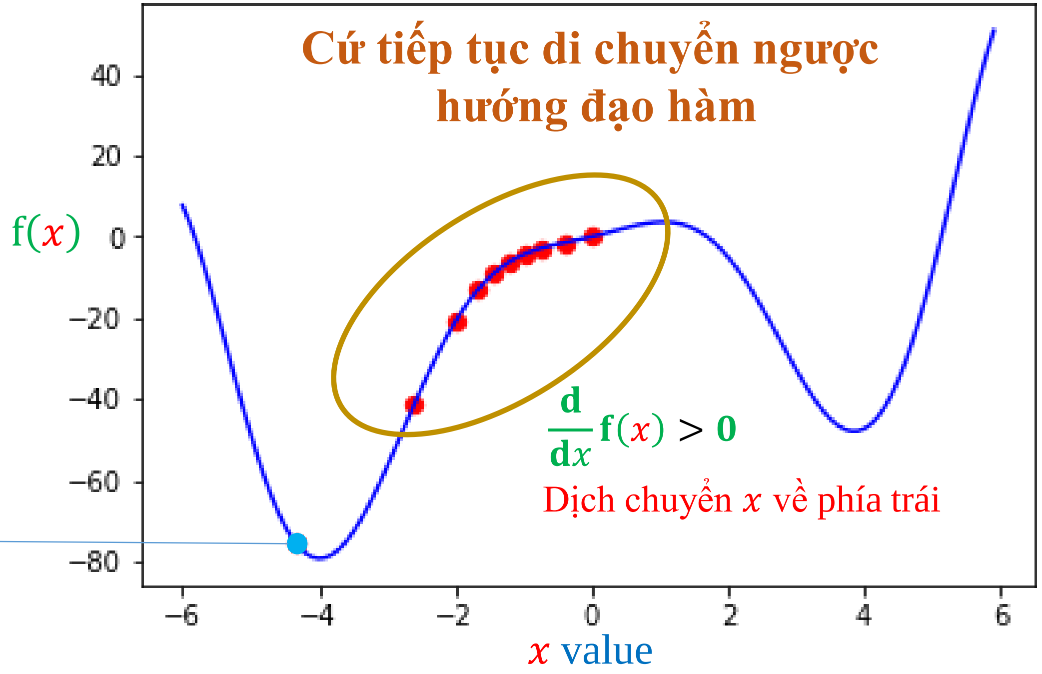
Derivative/Gradient

❖ **A cue to optimize a function**

Khởi tạo giá trị x





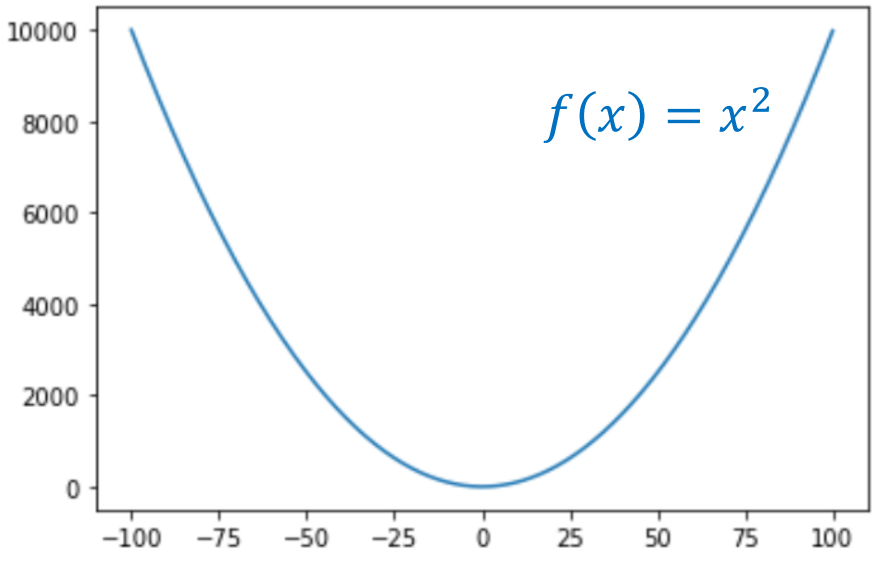


$\frac{d}{d\theta} f(x) < 0$

Dịch chuyển x về phía phải

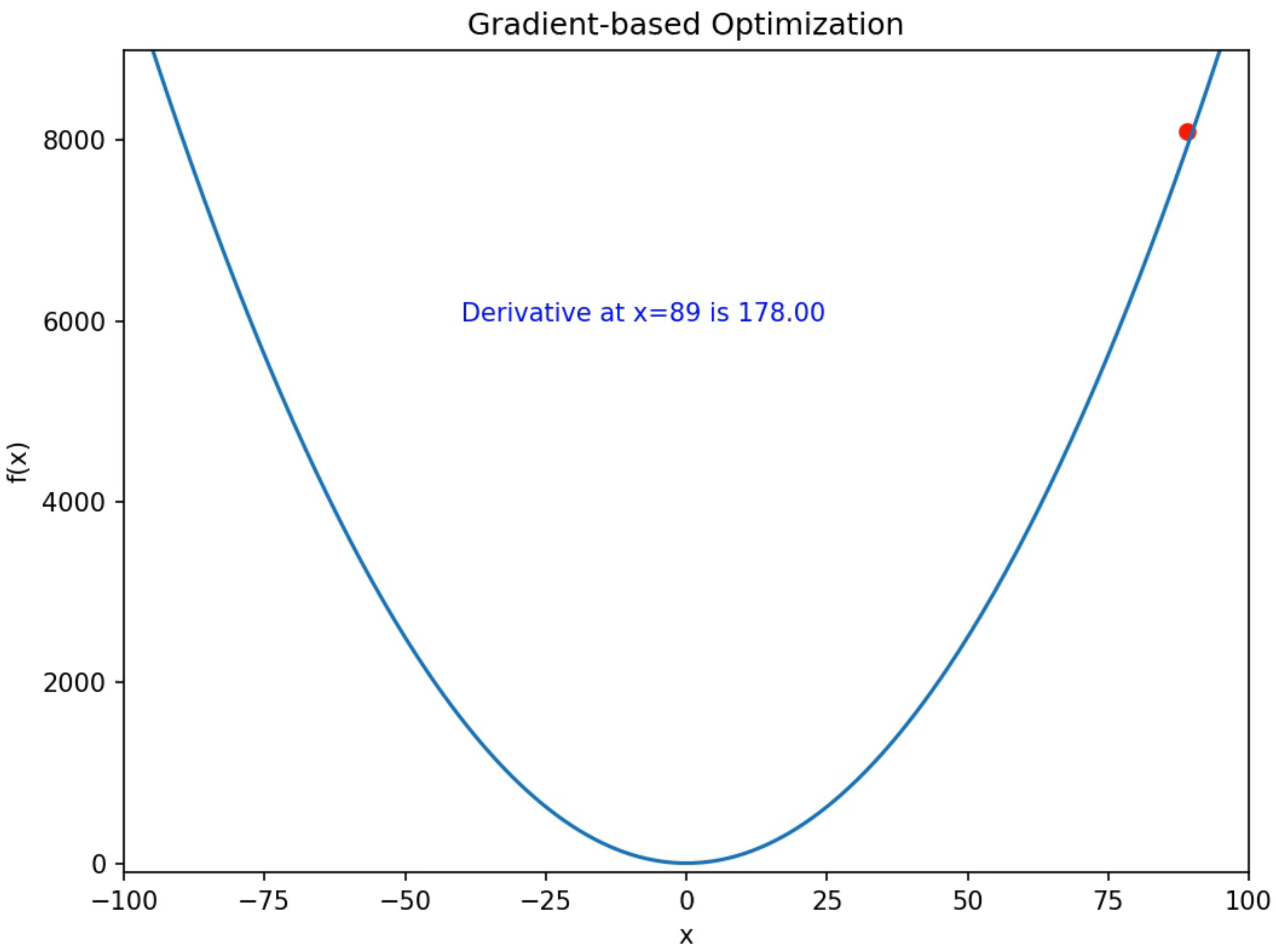
Derivative and Applications

❖ Optimization



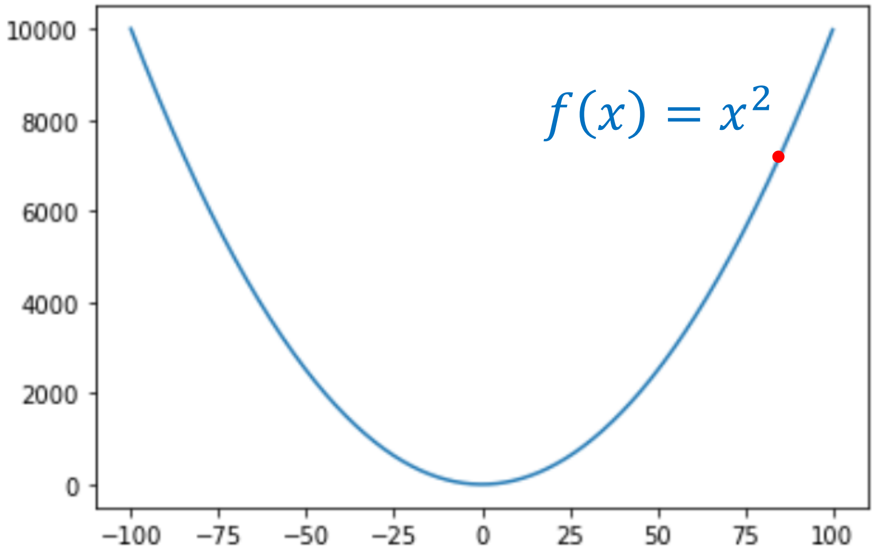
$$-100 \leq x \leq 100$$

$$x \in \mathbb{N}$$



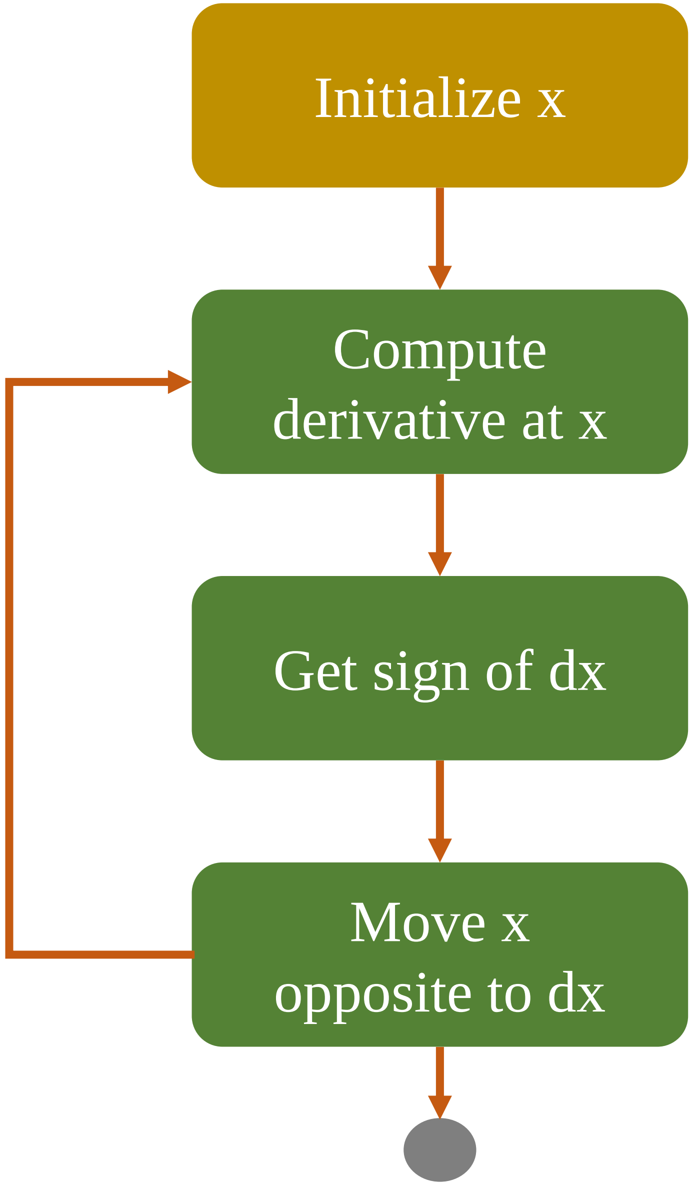
Derivative and Applications

❖ Simple Optimization



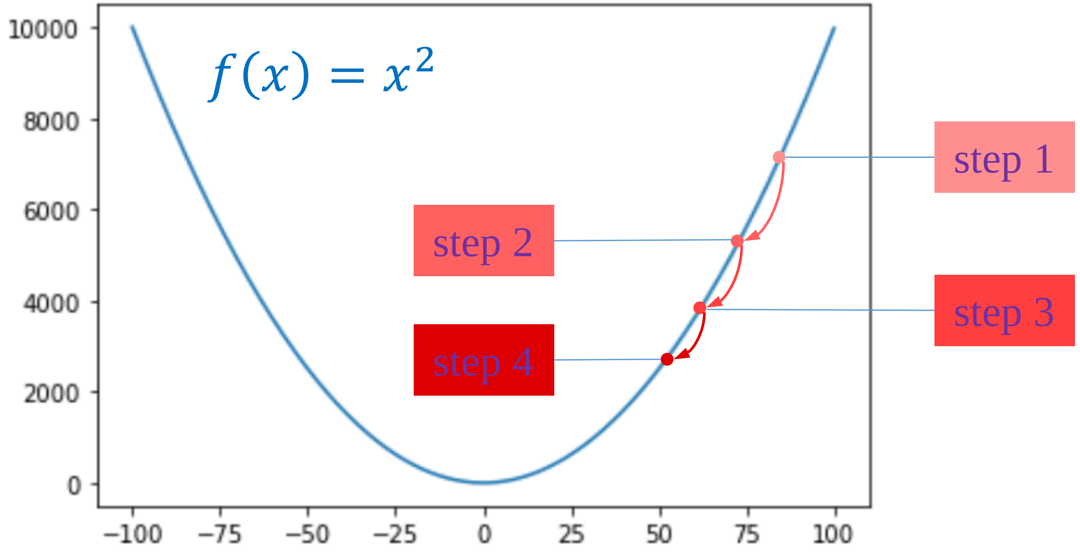
$$-100 \leq x \leq 100$$

$$x \in \mathbb{N}$$



Derivative and Applications

❖ Simple Optimization



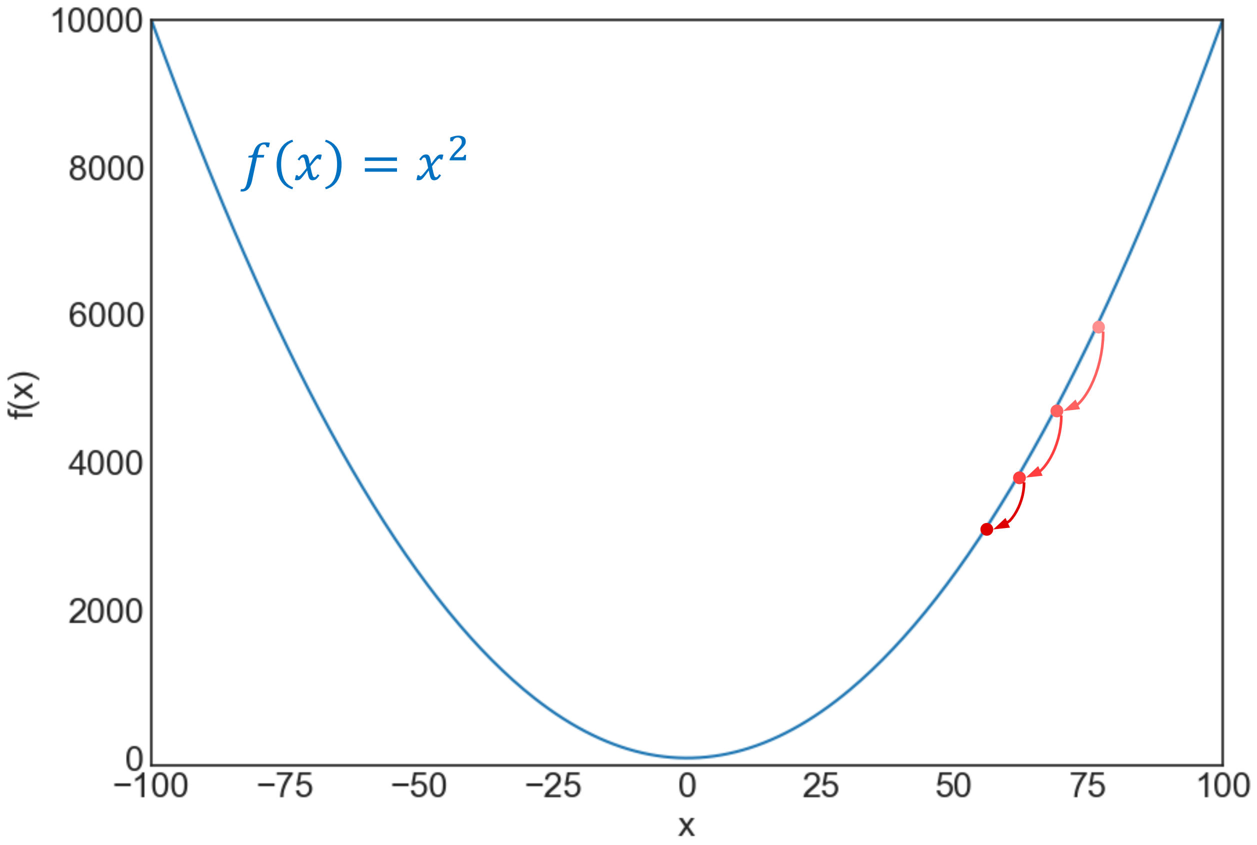
$$-100 \leq x \leq 100; \quad x \in \mathbb{N}$$

```python
# sphere function
def func(x):
    return x**2

# đạo hàm trung tâm
def gradient(f, x, e=1.0e-4):
    return (f(x + e/2) - f(x - e/2)) / e
```

```python
import random

# set x randomly
x = random.randint(-100, 100)

# params
num_iterations = 50
step = 1

# optimize
for _ in range(num_iterations):
    # compute the derivative at x
    dx = gradient(func, x)

    # get sign
    sign = 1
    if (dx < 0.0):
        sign = -1
    elif (dx == 0.0):
        sign = 0

    # update
    x = x - sign*step
```

Derivative and Applications

❖ Square function



$$-100 \leq x \leq 100$$
$$x \in \mathbb{N}$$

$$x_t = x_{t-1} - sign(f'(x_{t-1}))$$

$x_0 = 70.0$

$f'(x_0) = 140.0$

$x_1 = x_0 - sign(140) = 69.0$

$\mathrm{f}'(\mathrm{x}_1) = 138.0$

$x_2 = x_1 - sign(69.0\ ) = 68.0$

$\mathrm{f}'(\mathrm{x}_2) = 136.0$

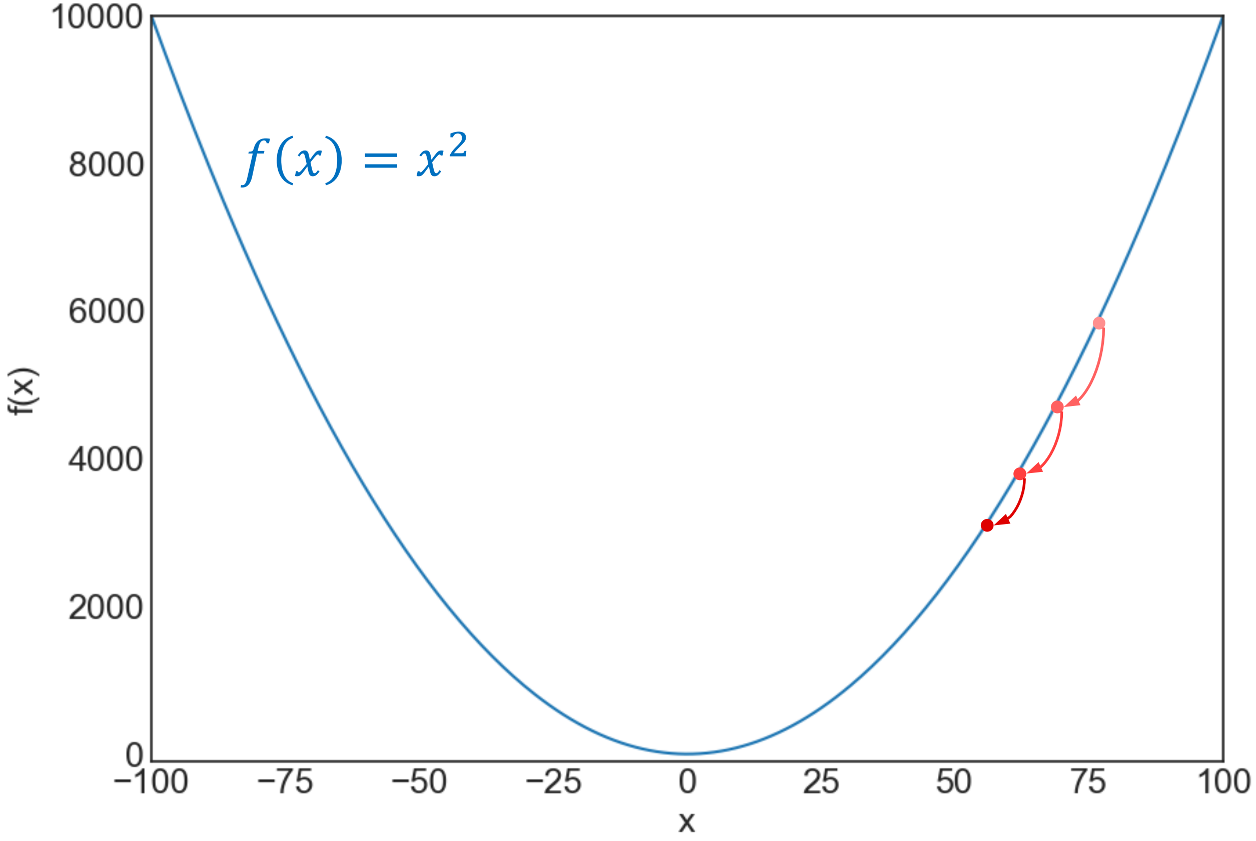
$x_3 = x_2 - sign(68.0\ ) = 67.0$

$\mathrm{f}'(\mathrm{x}_3) = 134.0$

$x_4 = x_3 - sign(67.0\ ) = 66.0$

Derivative and Applications

❖ **Square function**



Keep doing

$$x_t = x_{t-1} - sign(f'(x_{t-1}))$$

$x_{63} = 7.0$

$f'(x_{63}) = 14.0$

$x_{64} = x_{63} - sign(14.0\,) = 6.0$

$f'(x_{64}) = 12.0$

$x_{65} = x_{64} - sign(12.0\,) = 5.0$

$f'(x_{65}) = 10.0$

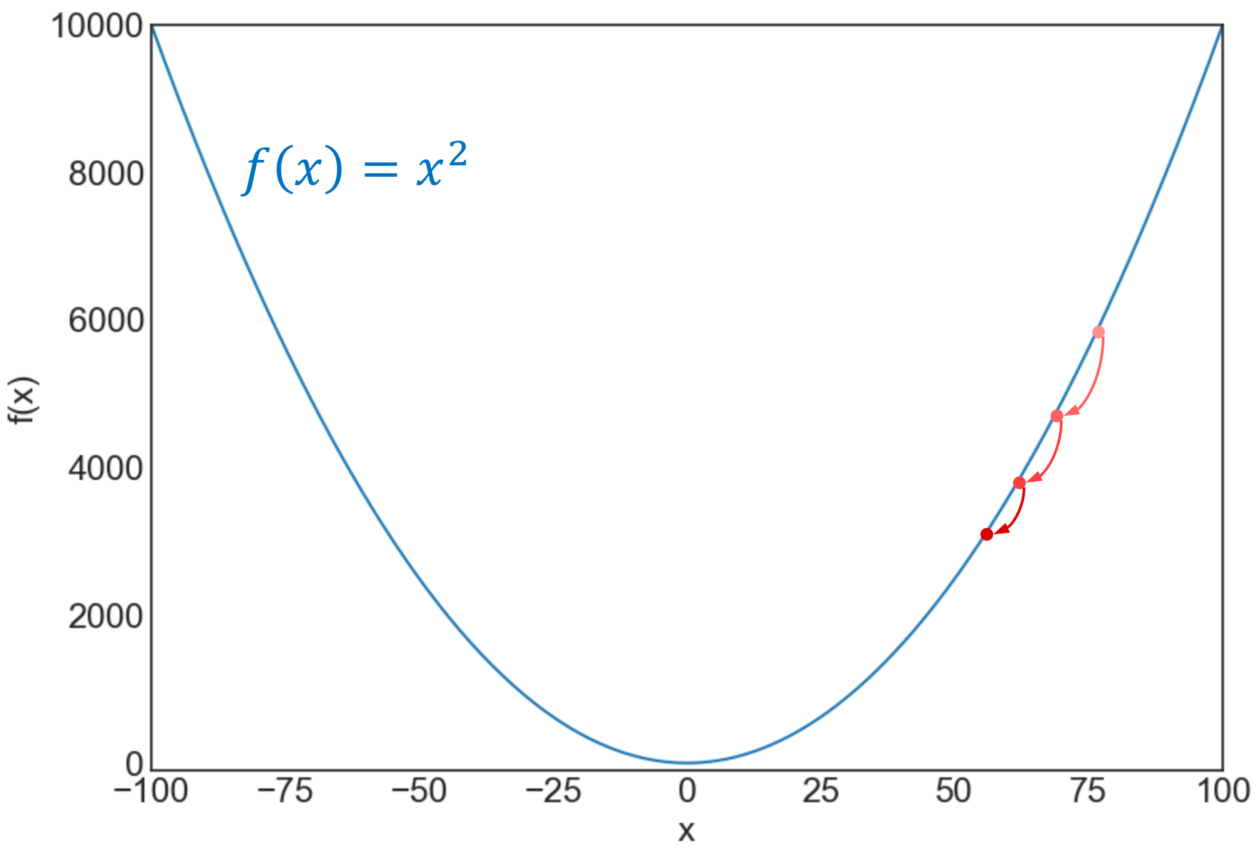
$x_{66} = x_{65} - sign(10.0\,) = 4.0$

$f'(x_{66}) = 8.0$

$x_{67} = x_{66} - sign(8.0\,) = 3.0$

Derivative and Applications

❖ **Square function**



Keep doing

$$x_t = x_{t-1} - sign(f'(x_{t-1}))$$

$$x_{67} = 3.0$$

$$f'(x_{67}) = 6.0$$

$$x_{68} = x_{67} - sign(6.0\,) = 2.0$$

$$\mathrm{f}'(\mathrm{x}_{68}) = 4.0$$

$$x_{69} = x_{68} - sign(4.0\,) = 1.0$$

$$\mathrm{f}'(\mathrm{x}_{69}) = 2.0$$

$$x_{70} = x_{69} - sign(2.0\,) = 0.0$$

$$\mathrm{f}'(\mathrm{x}_{70}) = 0.0$$

$$x_{67} = x_{66} - sign(0.0\,) = 0.0$$

Derivative

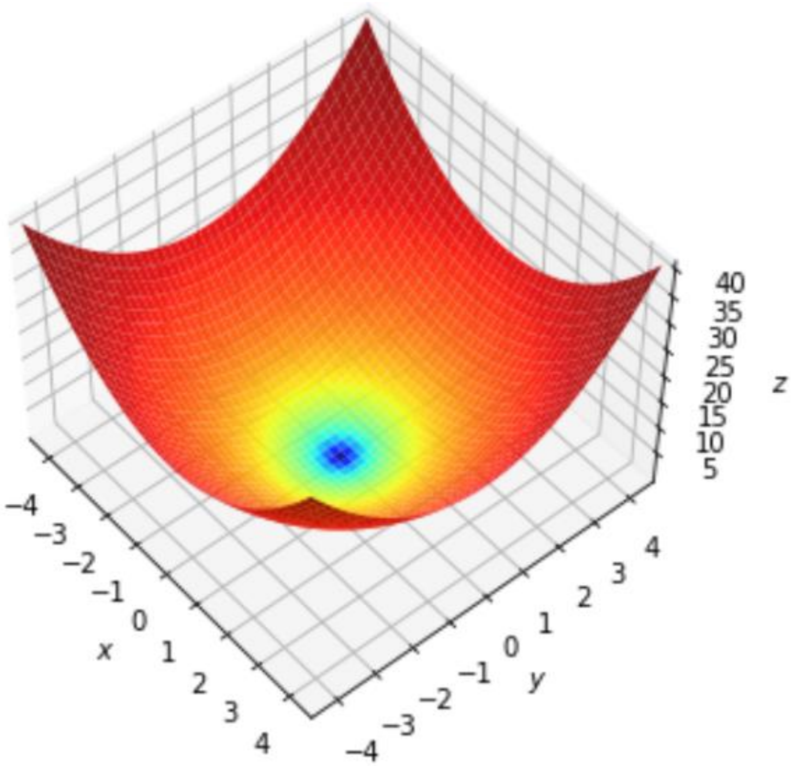
$$\frac{\partial}{\partial x} f(x, y) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x/2, y) - f(x - \Delta x/2, y)}{\Delta x}$$

$$\frac{\partial}{\partial y} f(x, y) = \lim_{\Delta y \to 0} \frac{f(x, y + \Delta y/2) - f(x, y - \Delta y/2)}{\Delta y}$$

❖ Optimization: 2D function

$$f(x, y) = x^2 + y^2$$

$$-100 \le x, y \le 100; x \in \mathbb{N}$$




```python
# sphere function
def func(x, y):
    return x**2 + y**2

```

```python
# đạo hàm trung tâm
def gradient(f, x, y, e=1.0e-4):
    dx = (f(x+e/2, y) - f(x-e/2, y)) / e
    dy = (f(x, y+e/2) - f(x, y-e/2)) / e
    return dx, dy
```

```python
print(gradient(func, -1, 1))
print(gradient(func, 0, 5))
print(gradient(func, 5, -5))
```

```
(-1.9999999967268423, 1.9999999967268423)
(0.0, 9.99999993922529)
(9.999999974752427, -9.999999974752427)
```

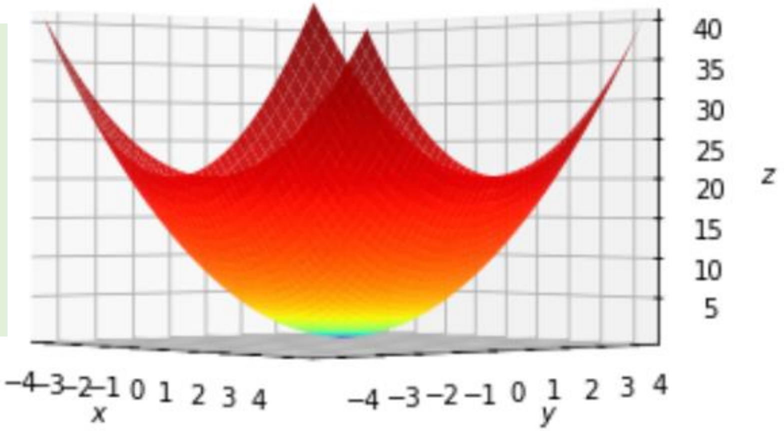
Derivative

❖ Optimization: 2D function

$$f(x, y) = x^2 + y^2$$

$$-100 \leq x, y \leq 100$$

$$x, y \in \mathbb{N}$$



$$x = x - sign\left(\frac{\partial f(x, y)}{\partial x}\right)$$

$$y = y - sign\left(\frac{\partial f(x, y)}{\partial y}\right)$$

$$x_0 = 3.0 \qquad y_0 = 4.0$$

$$\frac{\partial f(x_0, y_0)}{\partial x} = 6.0 \qquad \frac{\partial f(x_0, y_0)}{\partial y} = 8.0$$

$$x_1 = 2.0 \qquad y_1 = 3.0$$

$$\frac{\partial f(x_1, y_1)}{\partial x} = 4.0 \qquad \frac{\partial f(x_1, y_1)}{\partial y} = 6.0$$

$$x_2 = 1.0 \qquad y_2 = 2.0$$

$$\frac{\partial f(x_2, y_2)}{\partial x} = 2.0 \qquad \frac{\partial f(x_2, y_2)}{\partial y} = 4.0$$

$$x_3 = 0.0 \qquad y_3 = 1.0$$

$$\frac{\partial f(x_3, y_3)}{\partial x} = 0.0 \qquad \frac{\partial f(x_3, y_3)}{\partial y} = 0.0$$

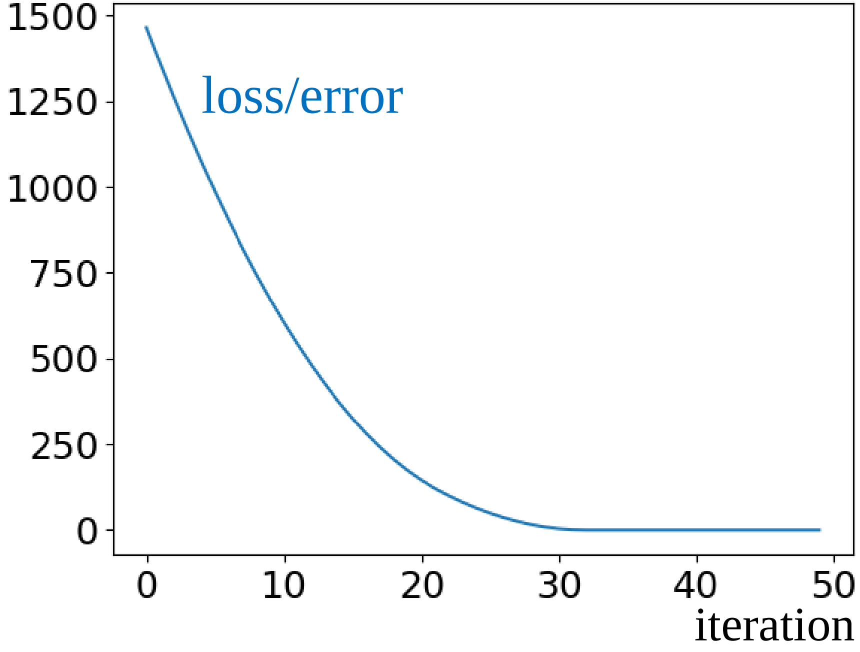
$$x_4 = 0.0 \qquad y_4 = 0.0$$

Derivative and Applications

$$f(x, y) = x^2 + y^2$$

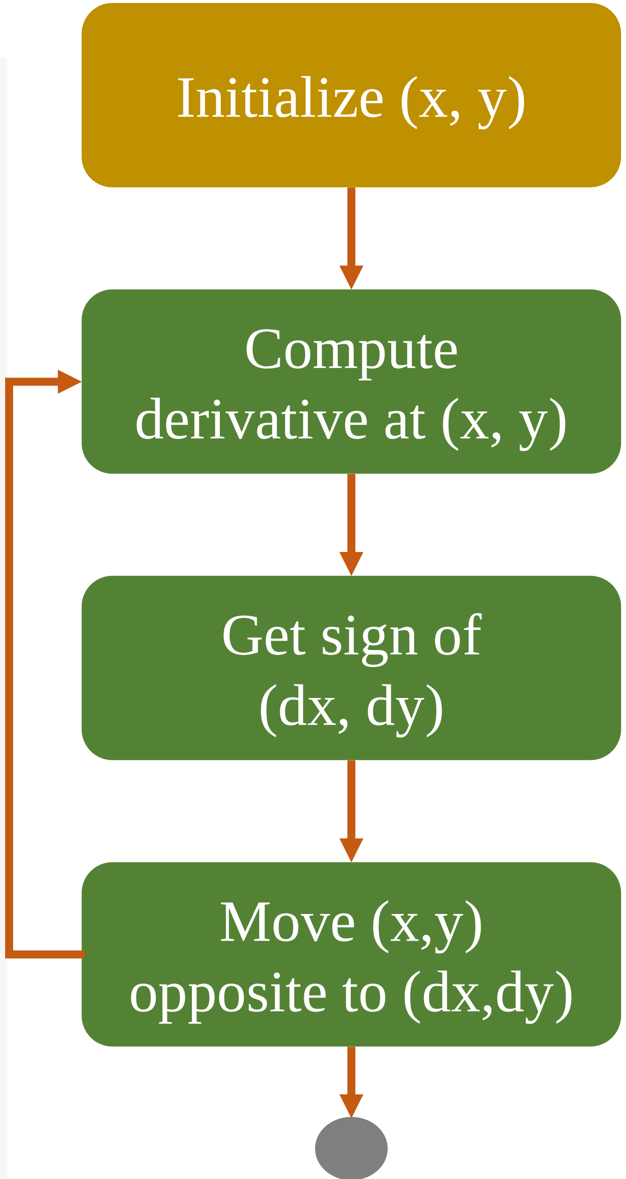
$$-100 \leq x, y \leq 100$$

$$x \in \mathbb{N}$$




```python
# sphere function
def func(x, y):
    return x**2 + y**2

# đạo hàm trung tâm
def gradient(f, x, y, e=1.0e-4):
    dx = (f(x+e/2, y) - f(x-e/2, y)) / e
    dy = (f(x, y+e/2) - f(x, y-e/2)) / e
    return dx, dy
```

```python
import random

# set (x,y) randomly
x = random.randint(-100, 100)
y = random.randint(-100, 100)

# params
num_iterations = 80
step = 1

# optimize
for _ in range(num_iterations):
    # compute the derivative at (x,y)
    dx, dy = gradient(func, x, y)

    # get sign of dx
    sign_x = np.sign(dx)
    sign_y = np.sign(dy)

    # update
    x = x - sign_x*step
    y = y - sign_y*step
```

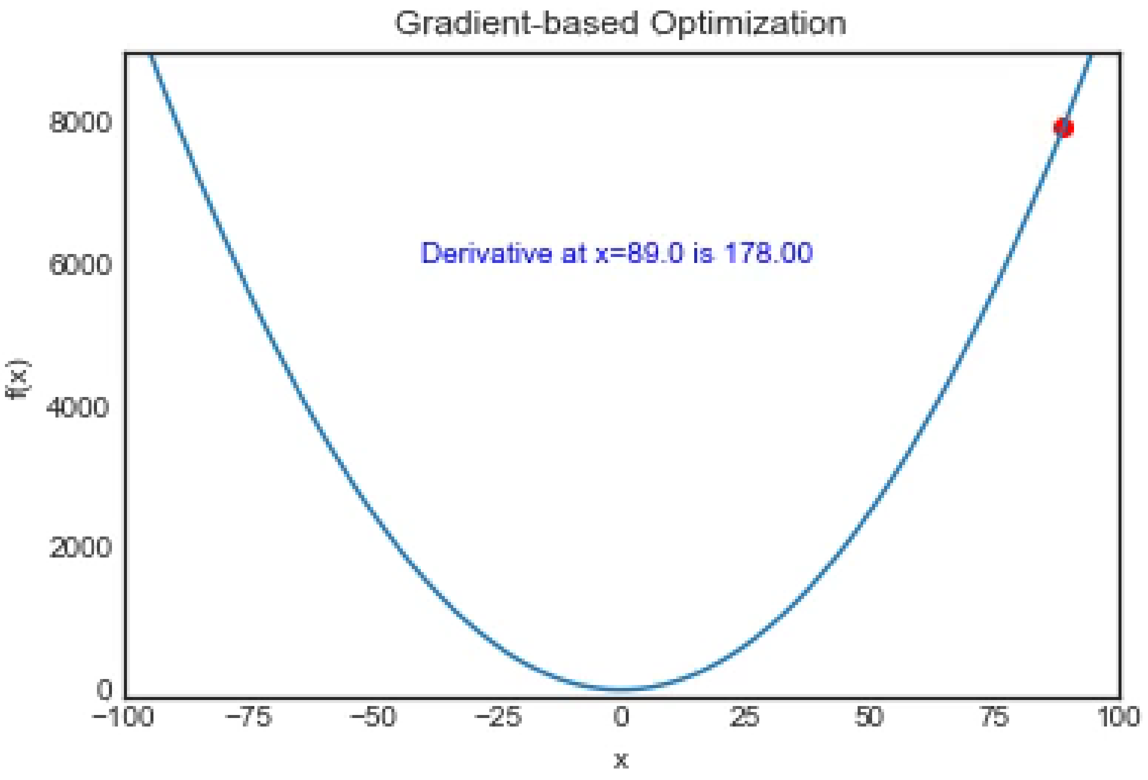


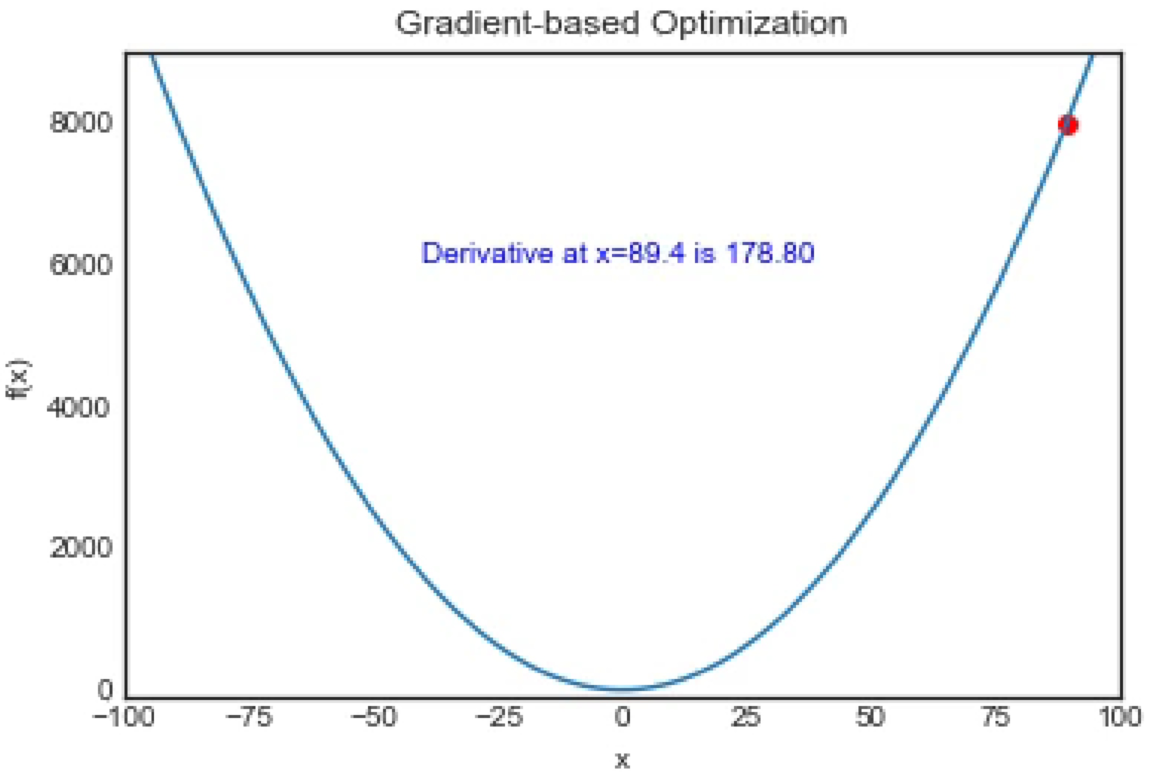
Discussion

Stochastic gradient descent

$$x_t = x_{t-1} - sign(\frac{d}{dx}f(x))$$

Depending on an initial value

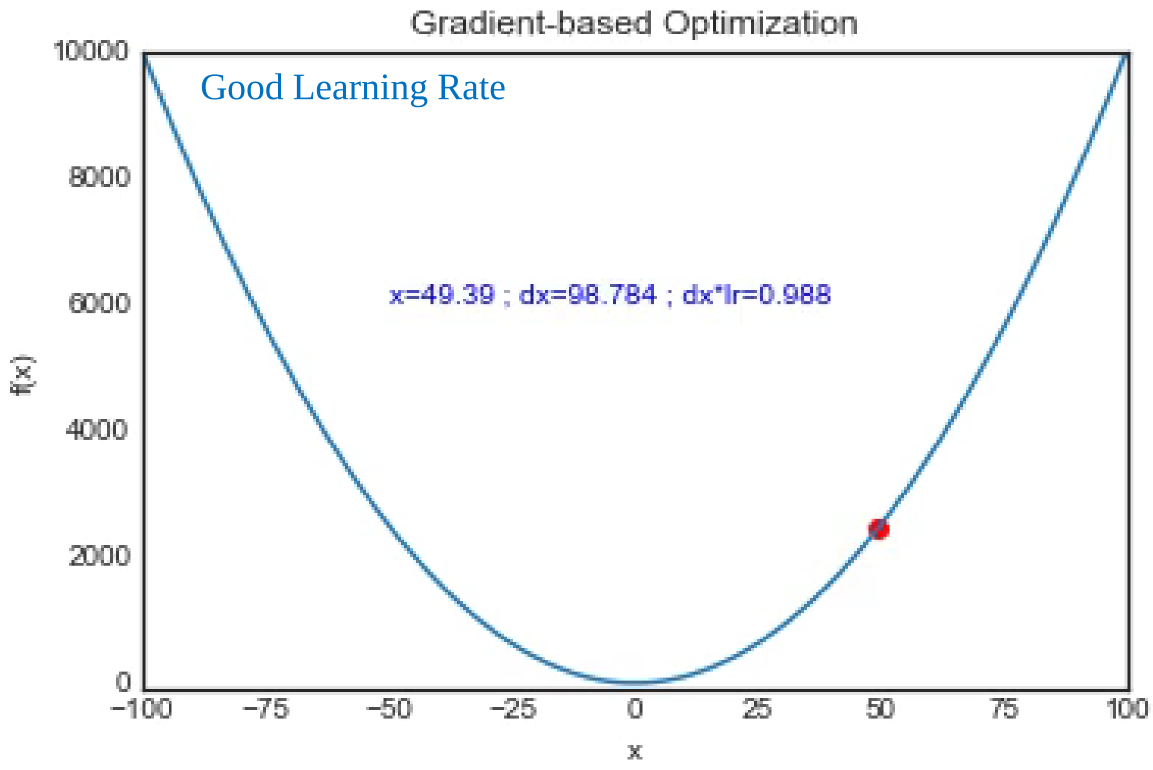


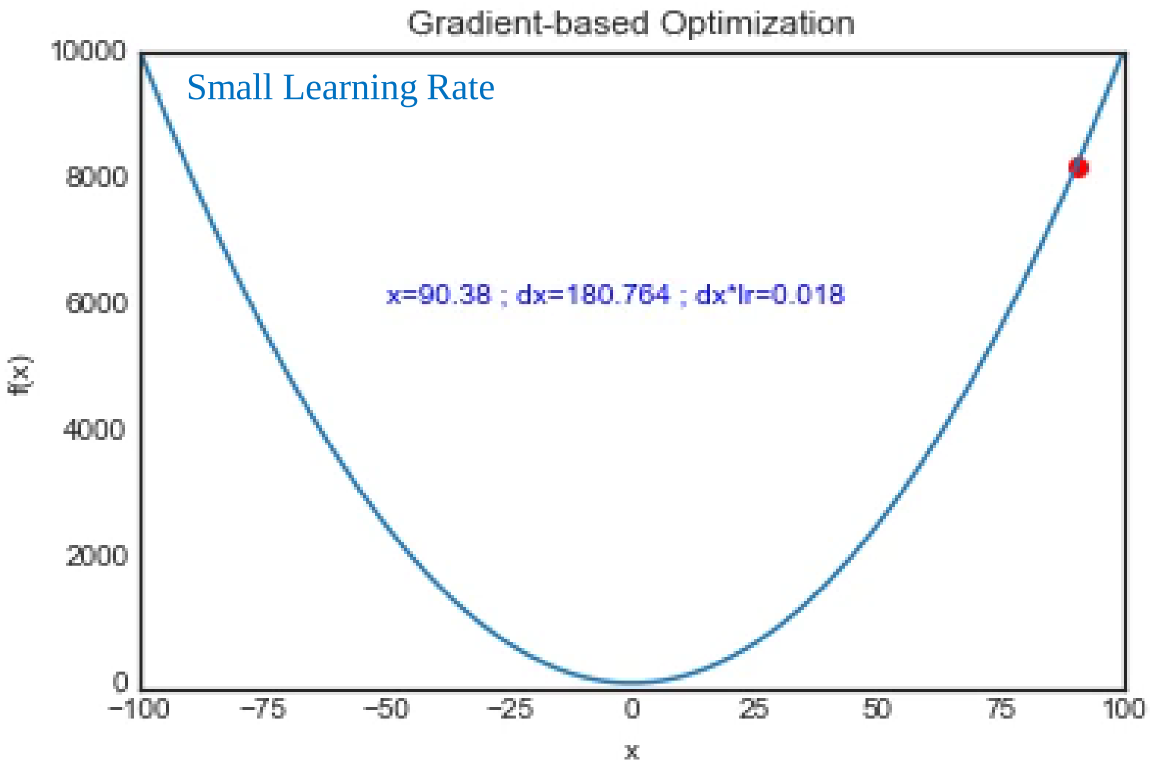


Discussion

Stochastic gradient descent

$$x_t = x_{t-1} - \eta \frac{d}{dx} f(x)$$

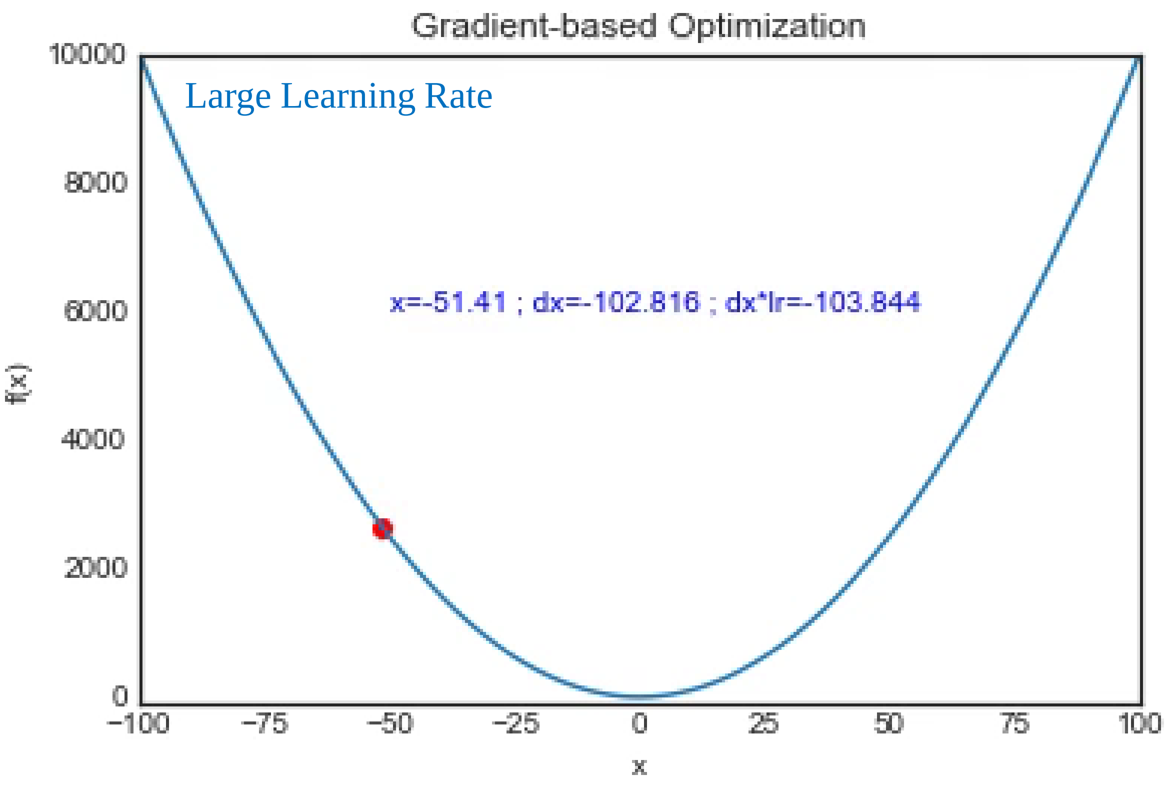


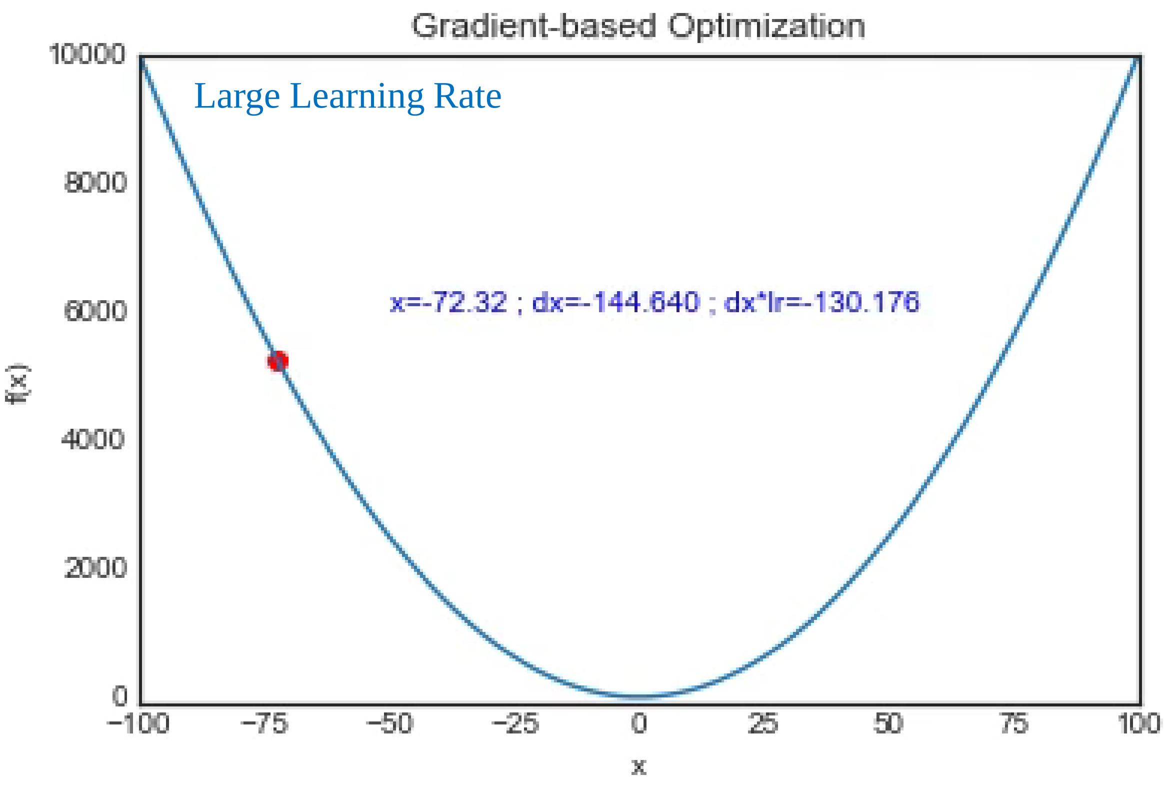


Discussion

Stochastic gradient descent

$$x_t = x_{t-1} - \eta \frac{d}{dx} f(x)$$





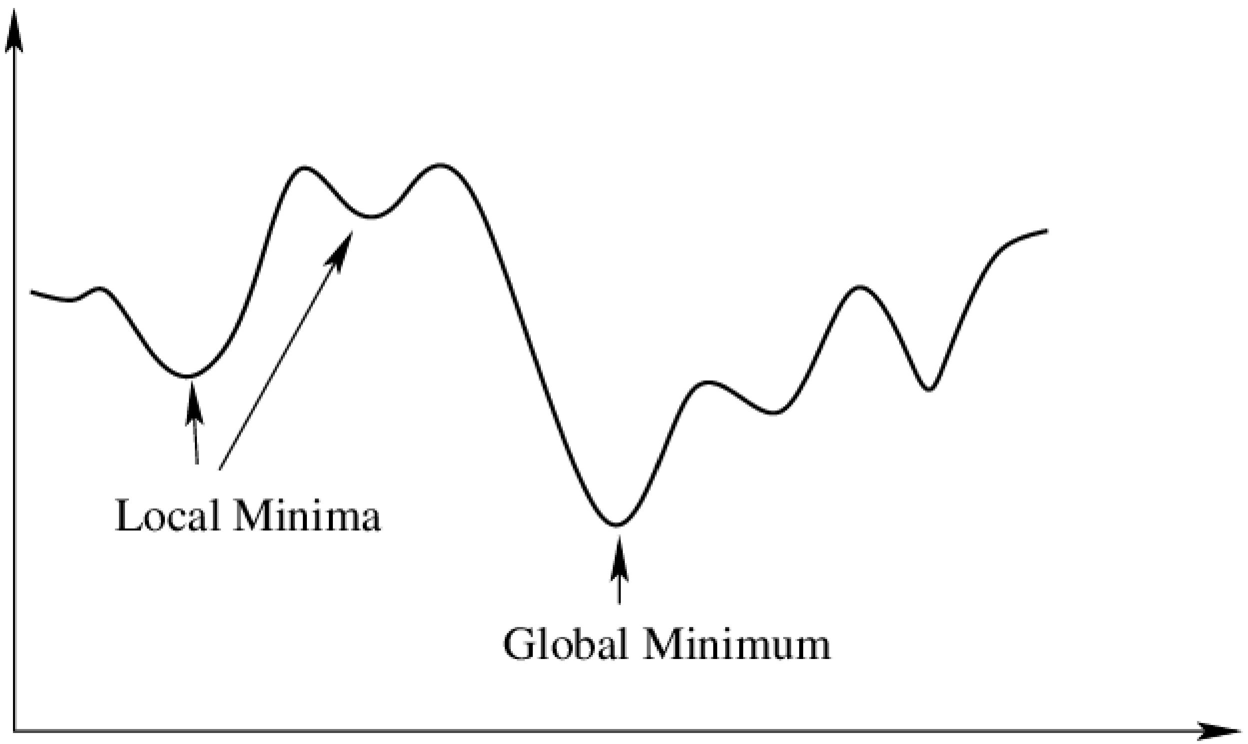
Discussion

Challenges

Local minima

Global minima

Saddle points

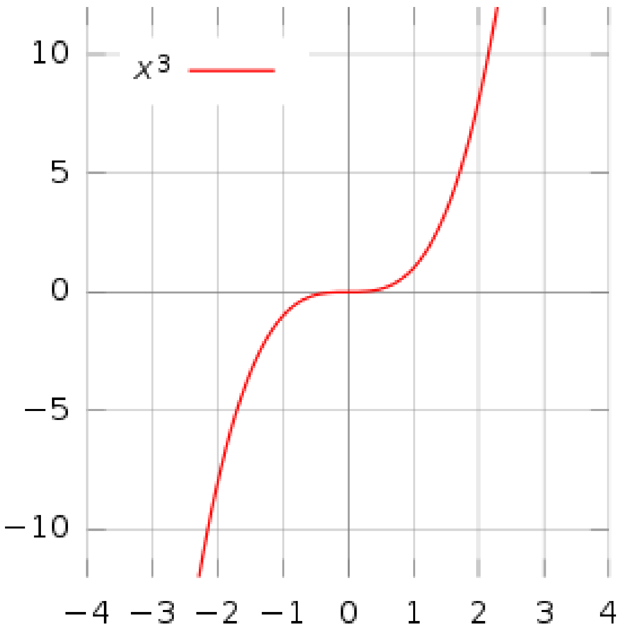


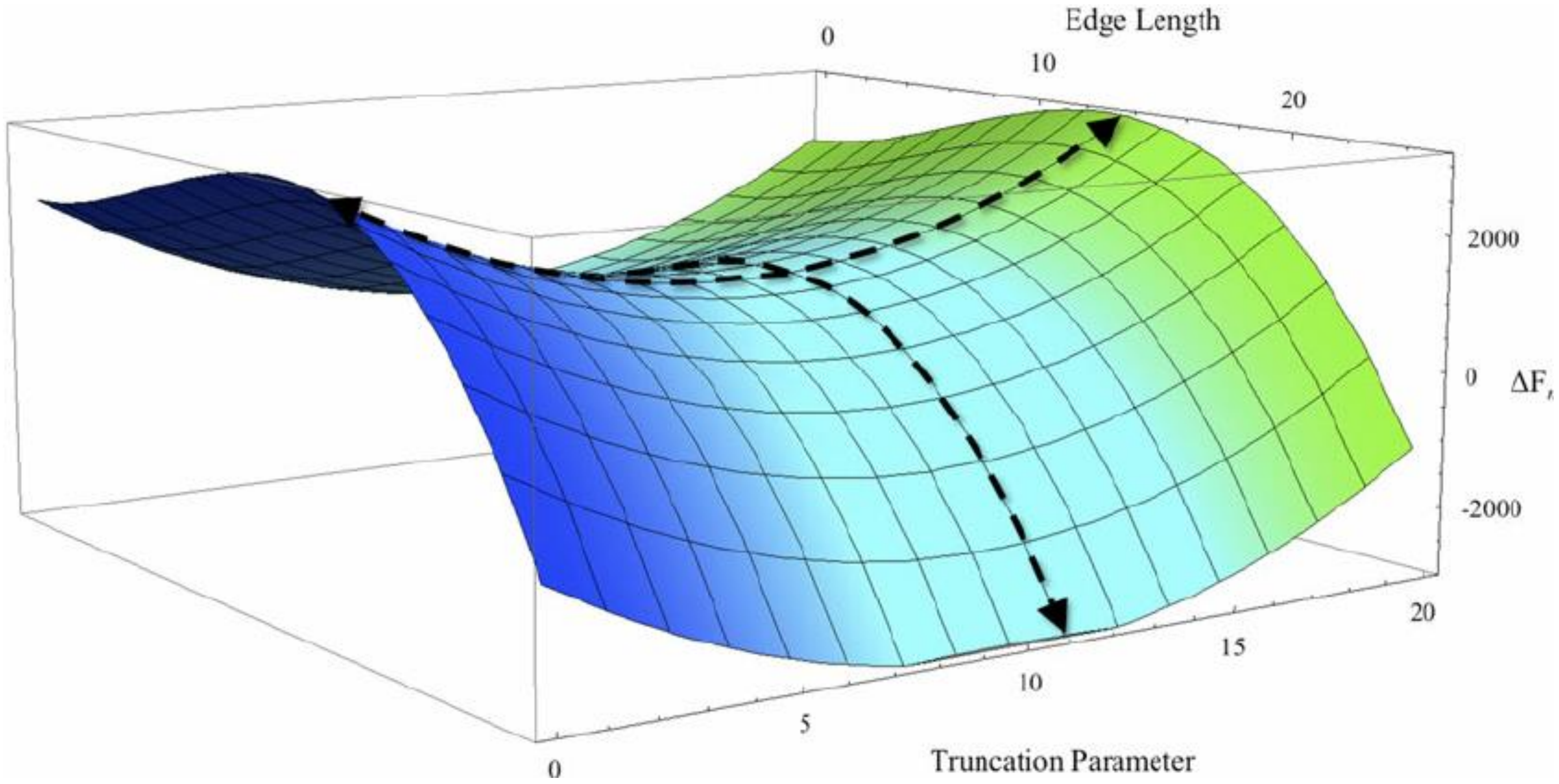
https://vitalflux.com/local-global-maxima-minima-explained-examples/

Discussion

Challenges

Saddle points





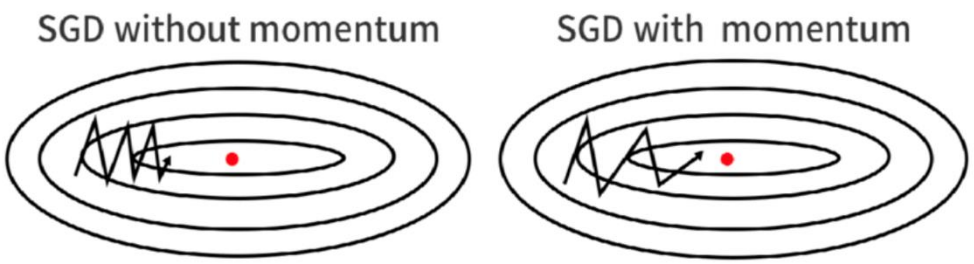
https://www.wikiwand.com/en/Saddle_point

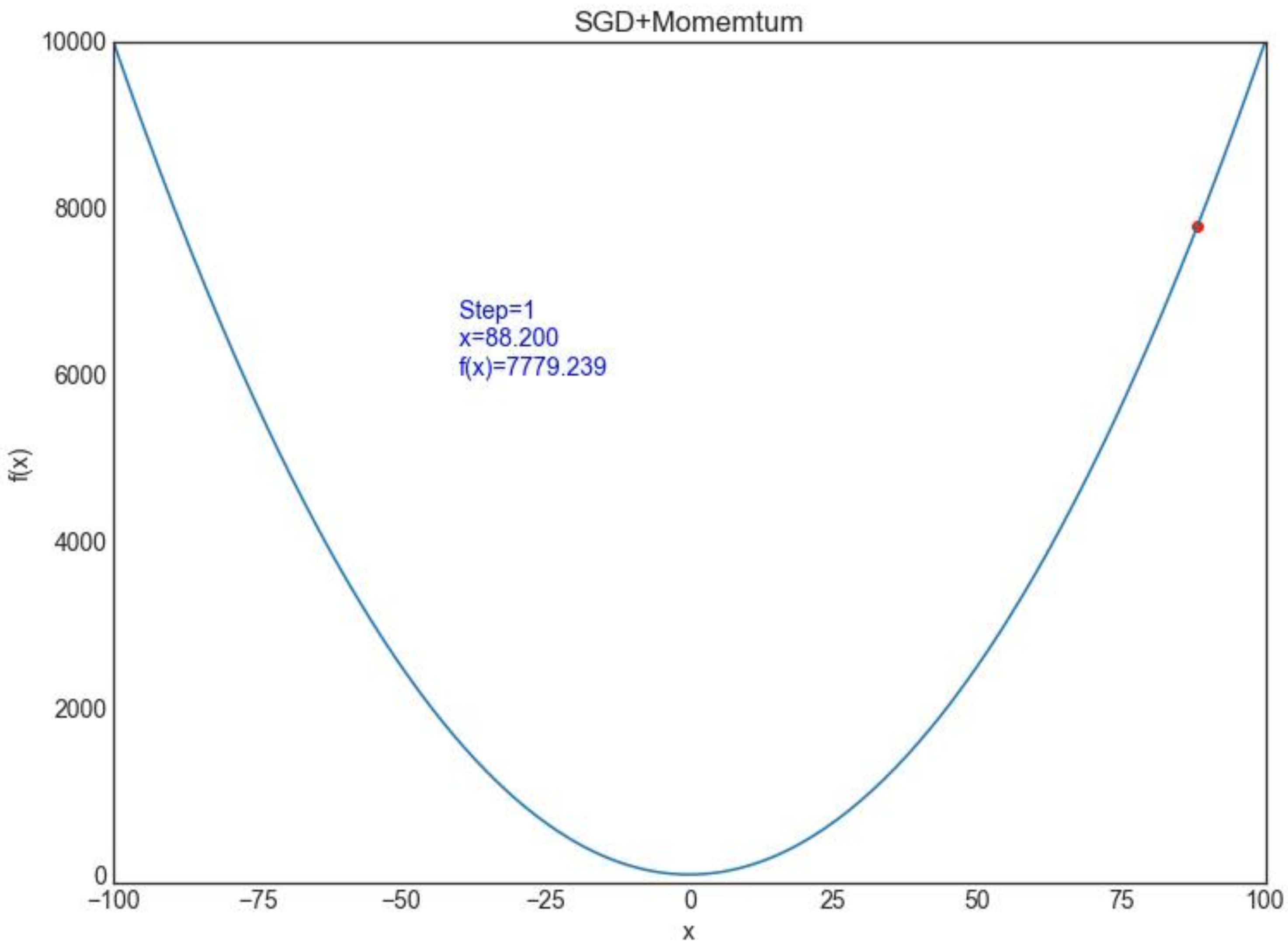
Discussion

SGD + Momentum

$$v_t = \gamma v_{t-1} - \eta \frac{d}{dx} f(x)$$

$$x_t = x_{t-1} - v_t$$

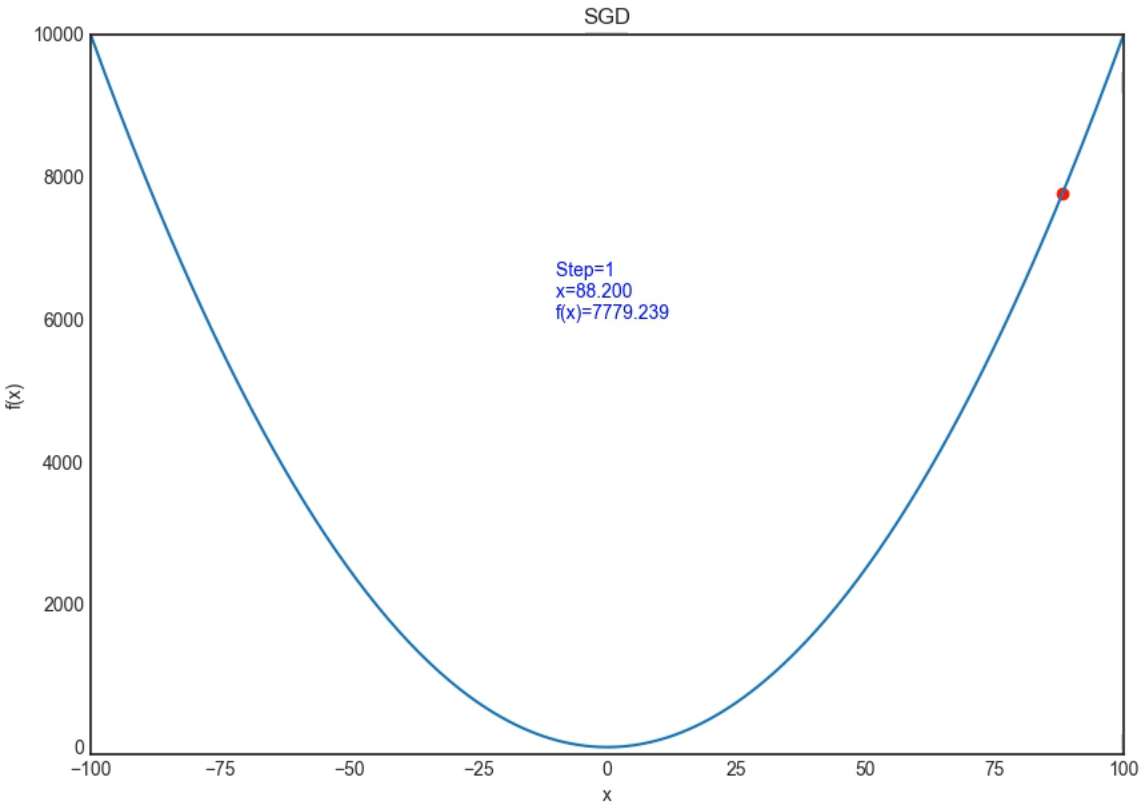


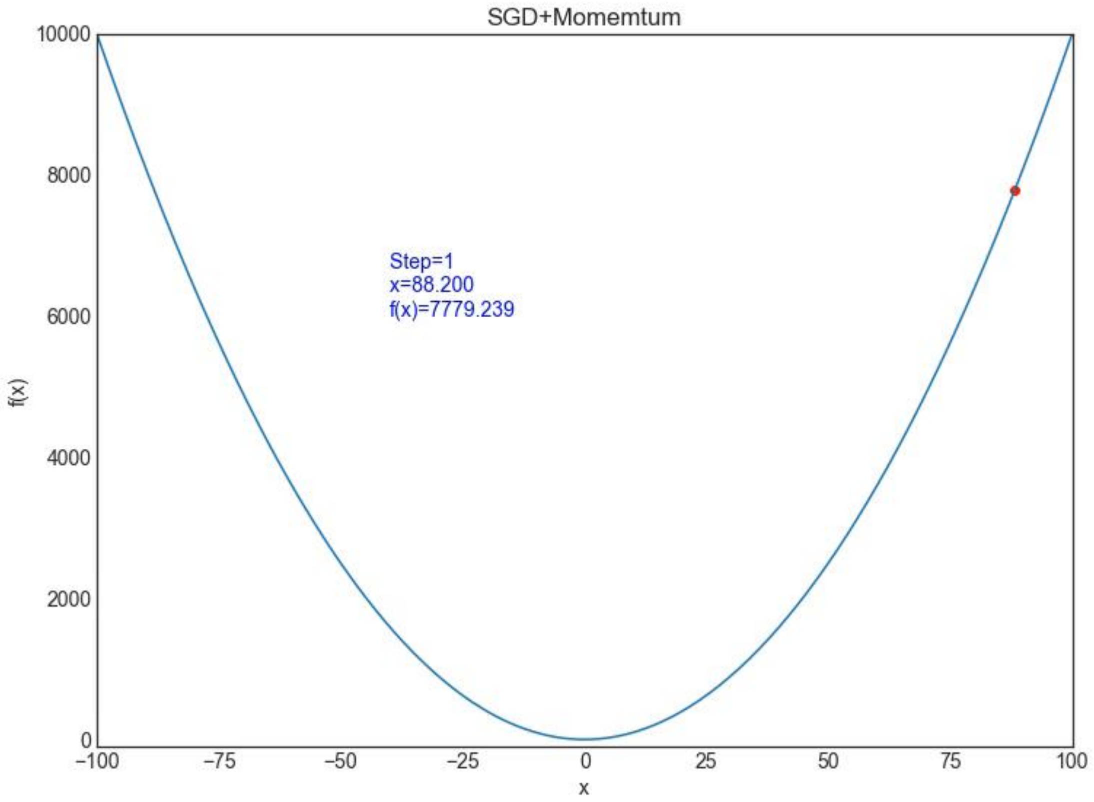


Genevieve B. Orr

Discussion

SGD + Momentum





Discussion

SGD + Momentum