EEP 596A
Computer Vision: Classical and Deep Methods
2025 Fall

Homework 5 Report

Name: Po Peng
NetID: ericpp
Date: November 03, 2025

Contents

Task 5 – Optimization

(1) Effect of learning rate

As the learning rate increases, convergence becomes faster.
However, in other starting positions or more complex surfaces, too large a learning rate could lead to instability or divergence.

I tested several learning rates and starting positions using SGD.
With a small learning rate (η = 0.01), the algorithm converged slowly after 266 iterations.
When η = 0.05 or 0.1, it reached the global minimum much faster, after 51 and 24 iterations respectively.
However, when η was increased to 1.0, the updates overshot the center and the position got stuck near the boundary (x = 246, y = 56), showing that the optimization diverged.

```
(base) eric@ericdeMacBook-Pro hw5 % python Assignment5.py
Testing SGD with different learning rates and starting points:

Start=(10,200), lr=0.01
[lr=0.01] iter=   0  ->  (x=12.36, y=198.56)
[lr=0.01] iter=  50  ->  (x=85.92, y=153.72)
[lr=0.01] iter= 100  ->  (x=112.68, y=137.36)
[lr=0.01] iter= 150  ->  (x=122.44, y=131.38)
[lr=0.01] iter= 200  ->  (x=125.94, y=129.28)
[lr=0.01] iter= 250  ->  (x=127.22, y=128.50)
[lr=0.01] Converged after 266 iterations -> (127.52, 128.50)

Start=(10,200), lr=0.05
[lr=0.05] iter=   0  ->  (x=21.80, y=192.80)
[lr=0.05] iter=  50  ->  (x=127.60, y=128.40)
[lr=0.05] Converged after 51 iterations -> (127.60, 128.40)

Start=(10,200), lr=0.5
[lr=0.5] iter=   0  ->  (x=128.00, y=128.00)
[lr=0.5] Converged after 1 iterations -> (128.00, 128.00)

Start=(20,240), lr=0.5
[lr=0.5] iter=   0  ->  (x=128.00, y=128.00)
[lr=0.5] Converged after 1 iterations -> (128.00, 128.00)

Start=(20,240), lr=0.1
[lr=0.1] iter=   0  ->  (x=41.60, y=217.60)
[lr=0.1] Converged after 24 iterations -> (127.60, 128.40)

Start=(10,200), lr=1.0
[lr=1.0] iter=   0  ->  (x=246.00, y=56.00)
[lr=1.0] iter=  50  ->  (x=246.00, y=56.00)
[lr=1.0] iter= 100  ->  (x=246.00, y=56.00)
[lr=1.0] iter= 150  ->  (x=246.00, y=56.00)
[lr=1.0] iter= 200  ->  (x=246.00, y=56.00)
[lr=1.0] iter= 250  ->  (x=246.00, y=56.00)
[lr=1.0] iter= 300  ->  (x=246.00, y=56.00)
[lr=1.0] iter= 350  ->  (x=246.00, y=56.00)
[lr=1.0] iter= 400  ->  (x=246.00, y=56.00)
```

```
[lr=1.0] iter= 450  ->  (x=246.00, y=56.00)
[lr=1.0] Did NOT converge within 500 iterations -> (10.00, 200.00)
```

Start (x_0, y_0)	Learning rate (η)	Final (x, y)	Iterations	Observation
(10, 200)	0.01	(127.52, 128.50)	266	Slow but stable convergence
(10, 200)	0.05	(127.60, 128.40)	51	Fast and stable convergence
(10, 200)	0.50	(128.00, 128.00)	1	Immediate convergence (near minimum)
(20, 240)	0.50	(128.00, 128.00)	1	Immediate convergence (near minimum)
(20, 240)	0.10	(127.60, 128.40)	24	Fast and stable
(10, 200)	1.00	(246.00, 56.00)	500	Did not converge (diverged)

(2) Divergence case

This experiment confirms that while small or moderate learning rates ensure stable convergence, an excessively large learning rate can make the updates unstable.
For example, with $\eta = 1.0$, the algorithm failed to converge within 500 iterations, proving that too large a step size leads to divergence.