دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

یادگیری ماشین

تمرین شماره 2

نام و نام خانوادگی
سید محمد جزایری

شماره دانشجویی
810101399

آبان 1404

فهرست (لطفاً پس از تکمیل گزارش، این فهرست را بهروز کنید.)

فهرست شکل‌ها

فهرست جدول‌ها

چکیده

In this homework we implemented our knowledge of optimization to solve some problems as well as some real-world hands-on problems. Some of the questions even compared two methods with each other to show us their convergence rate, time complexity etc.

پرسش 1 - عنوان پرسش

متن نمونه

1-1. عنوان بخش

$$min_{\alpha > 0} f(x^{(t)} + \ \alpha p^{(t)})$$

Search direction(p) determines the path we take to reach the extremum point.

Step length determines how long to go in the path, a sufficient step length should be neither too long nor too short.

2-1. عنوان بخش

A vector p is a descent direction if for $\alpha \ > \ 0 \ f(x) \ > \ f(x \ + \ \alpha p)$, hence the necessary condition is: $\Delta_x f^T(x)p \ < \ 0$ (google doc doesn’t have the gradient symbol so I put Δ).

3-1. عنوان بخش

Exact Line Search: This method finds the global minimizer α of the one-dimensional function $\phi(\alpha) \ = \ f(x^{(t)} \ + \ \alpha^{(t)})$ exactly. This is often done by setting the derivative

$\phi'(\alpha)$ to zero and solving for α. It is computationally expensive but theoretically ideal, especially for simple functions like quadratics.

Armijo/Backtracking Line Search: This is an inexact line search that focuses on finding a step length that guarantees a sufficient decrease in the function value. It starts with an initial step size (e.g., $\alpha = 1$) and repeatedly multiplies α by a reduction factor (e.g., 0.5) until the Armijo condition (or sufficient decrease condition) is satisfied: $f(x^{(t)} + \alpha^{(t)}) \le f(x^{(t)}) + c_1 \alpha \Delta f(x^{(t)})^T p^{(t)}$, where $c_1 \in (0, 1)$.

Wolfe Conditions (Strong or Weak): These are a set of two conditions (a sufficient decrease condition, similar to Armijo, and a curvature condition) that ensure the step length is not too short (Armijo condition) and not too long (curvature condition). The curvature condition ensures the slope at the new point is steeper than c_2 times the initial slope: $f(x^{(t)} + \alpha p^{(t)})^T p^{(t)} \ge c_2 \Delta f(x^{(t)})^T p^{(t)}$. This guarantees convergence for general descent methods.

4-1. عنوان بخش

1. Let’s calculate the gradient first.

$\Delta_x f(x) =$

$$\begin{vmatrix} 4x_1 + x_2 \\ x_1 + 8x_2 \end{vmatrix}$$

Now let’s plug in $x^{(t)}$

$$\begin{vmatrix} 1 \\ 8 \end{vmatrix}$$

$\Delta_x f^T (x^{(t)}) p^{(t)} = 1 * 1 + 8 * (-2) = -15 < 0$ so it is a descent direction.

2. $x^{(t)} + \alpha p^{(t)} =$
3.

$$\begin{vmatrix} \alpha \\ 1 - 2\alpha \end{vmatrix}$$

So $x_1 = \alpha \; and \; x_2 = 1 - 2\alpha$. let’s plug these into $f(x_1, x_2)$:

$\phi(\alpha) = 16\alpha^2 - 15\alpha + 4$

3. Take the derivative and set to zero.

$$\phi'(\alpha) = 32\alpha - 15 \Rightarrow \phi^*(\alpha) = \frac{15}{32}.\ \alpha^* \geq 0$$

پرسش 3 - عنوان پرسش

متن نمونه

3-1. عنوان بخش

$$min_x F(x, y^*(x)) = (x - 2)^2 + (y^*(x) - 3)^2 s.t.\ x \geq 0$$

$$y^*(x) \in argmin_y f(x, y) = (y - x^2)^2 + y s.t.\ y \geq 0$$

3-2. عنوان بخش

Since we are assuming x is fixed we can take the derivative with respect to y and set it to zero.

$$\frac{\partial f}{\partial y} = 2y - 2x^2 + 1$$

$$y^* = x^2 - 0.5$$

If $2x^2 - 1 \geq 0$ then the optimal y is y^* but otherwise it is the boundary of the feasible set $[0, \infty]$ which is 0. So $y^* = max\{0, x^2 - 0.5\}$

3-3. عنوان بخش

If y < 0 $\Rightarrow x < \frac{1}{\sqrt{2}}$ then $F(x, y) = (x - 2)^2 + 9$ else $F(x, y) = (x - 2)^2 + (x^2 - 3.5)^2$

3-3. عنوان بخش

If y < 0 then the minimum occurs at x = 2 which is outside the boundary, so the minimum should happen at the boundaries. $F(0) = 13,\ F(\frac{1}{\sqrt{2}}) \approx 10.686$

Else $F'(x) = 0 \Rightarrow x^3 - 3x - 1 = 0 \Rightarrow x^* \approx 1.879 > \frac{1}{\sqrt{2}}$

$F(x^*) \approx 0.0146$ which is way smaller than $F(\frac{1}{\sqrt{2}})$ so this is our global minimizer and $y^* \approx 3.032$

3-4. عنوان بخش

table 1. discussion

Aspect	Discussion
Convexity of Follower	Makes us overconfident
Uniqueness of y	Since the follower's problem is strictly convex in y and the constraint $y \geq 0$ is convex, the optimal response is unique for any fixed x. The existence of a unique y is crucial for defining the value function of the leader.
Validity of Substitution	The substitution of the solution $y^*(x)$ into the leader's objective $F(x, y^*(x))$ is valid because $y^*(x)$ is a unique, single-valued function of x. This transformation converts the bilevel problem into an equivalent single-level optimization problem $min_x F(x, y^*(x))$. This is the standard approach for solving simple bilevel problems where the lower level has a unique solution.

پرسش 5 - عنوان پرسش

5-1. عنوان بخش

The metric originally used was: $dissimilarity_{SSD} = \frac{1}{L\cdot C}\sum_{p=1}^{L}\sum_{c\in\{R,G,B\}}(Pixel_A[p,c] - Pixel_B[p,c])^2$ for two pixels A and B and L is the length of the side being compared. Here C is the number of channels (3).

The second metric I used was a similar score but based on HSV rather than RGB.

$D_{HSV} = \frac{1}{L\cdot C}\sum_{p=1}^{L}\sum_{c\in\{H,S,V\}}(HSV_A[p,c] - HSV_B[p,c])^2$. Here C is the number of channels (3).

5-2. عنوان بخش

The overall fitness factor is:

$$F_{overall} = \frac{FITNESS-FACTOR}{\frac{1}{FITNESS-FACTOR} + \sum_{ALL\,EDGES} D_{HSV/RGB}}$$

The overall score is an inverse function of the total dissimilarity. This ensures that arrangements with low total mismatch (low dissimilarity) receive the highest fitness score.

3-5. عنوان بخش

The original code implemented roulette selection which assigns each candidate a probability proportional to its score and then chooses as many as necessary from the sample pool.

The selection I implemented was tournament selection which takes a subset from the sample pool and then chooses the best candidate in that subset and repeats this action as long as necessary.

Important note: both algorithms apply elitism and pass the 4 best candidates to the next round without any change to ensure that the next batch is at least as good as the current one.

4-5. عنوان بخش

We will compare four different approaches.

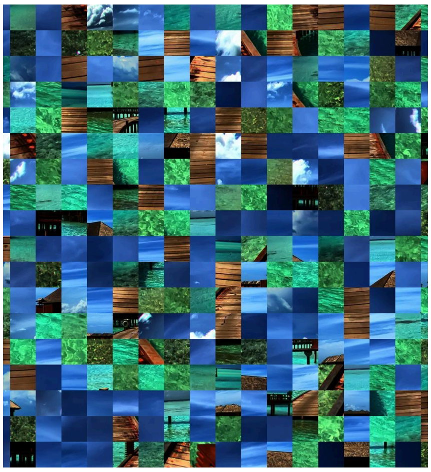

figure 1. The puzzle we started with

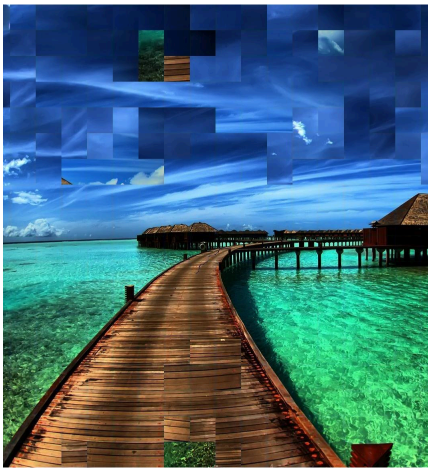

figure 2. solution_roulette_RGB

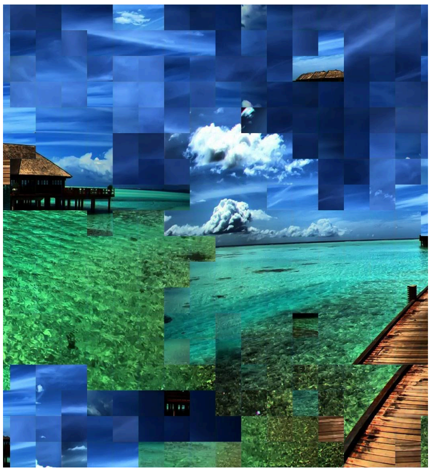

figure 3. solution_roulette_HSV

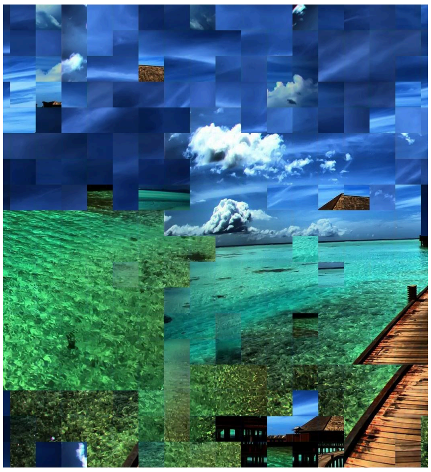

figure 4. solution_tournament_HSV

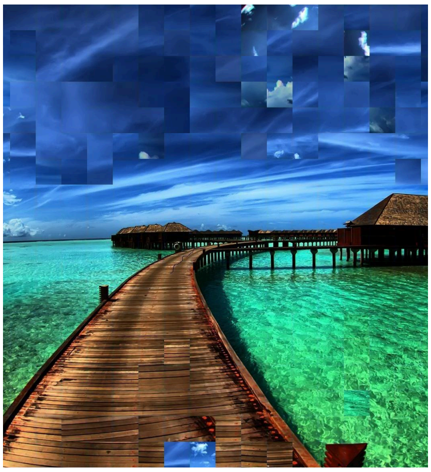

figure 5. solution_tournament_RGB

As it is obvious the RGB metric outperformed the HSV metric in both cases. But the selection metrics had a very close competition and it is my opinion that in this particular case tournament was better.

پرسش 7 - عنوان پرسش

7-1. عنوان بخش

Batch gradient descent is computationally very expensive because we need a full pass to update our parameter whereas stochastic gradient descent needs only one sample to do that.

7-2. عنوان بخش

BGD is less affected by noise since it needs the whole dataset for one update but SGD uses only a few, maybe even one sample so it’s more vulnerable to noise however this noise could help the algorithm escape local minima or saddle points!

7-3. عنوان بخش

BGD is very stable and has a smooth curve of descent and with a good learning rate and line search algorithm we are guaranteed to converge to the global minimum.

SGD is less stable because of noise and its trajectory is zig-zaggy. It does not settle exactly at the minimum. With a fixed, non-decreasing learning rate, SGD will oscillate around the minimum due to the high variance. To achieve true convergence, the learning rate must be gradually decreased (annealed) over time, ensuring the step size shrinks as the algorithm approaches the optimum. This ultimately allows it to converge near the minimum.

پرسش 8 - عنوان پرسش

We approximate the function at point $x + \alpha p$ with Taylor expansion.

$$f(x_k + \alpha p) \approx f(x_k) + \Delta f(x_k)^T(\alpha p)$$

$\Delta f = f(x_k + \alpha p) - f(x_k) \approx \alpha \Delta f(x_k)^T p$ (the second Δ is gradient)

The steepest descent direction p^* is the unit vector p that maximizes the decrease in f, which means minimizing the change Δf. For a fixed, small α > 0, minimizing Δf is equivalent to minimizing the directional derivative $\Delta f(x_k)^T p$ (Δ is gradient) subject to the constraint ||p||=1.

$$min_p \Delta f(x_k)^T p \ s.t. \ ||p|| = 1. \ \Delta \ is \ gradient$$

We will apply the Cauchy-Schwarz inequality: $a^T b = ||a|| \cdot ||b|| \cdot cos\theta$

$\Delta f(x_k)^T p = ||\Delta f(x_k)^T|| \cdot ||p|| \cdot cos\theta$ since ||p|| = 1:

$\Delta f(x_k)^T p = ||\Delta f(x_k)^T|| \cdot 1 \cdot cos\theta$ to make it minimum θ must be 180 degrees.

$$\Delta f(x_k)^T p = - ||\Delta f(x_k)^T|| \Rightarrow p^* =- \frac{\Delta f(x_k)^T}{||\Delta f(x_k)^T||}$$

پرسش 11 - عنوان پرسش

11-1. عنوان بخش

The convergence rate of an iterative method measures how quickly the error $e_k = x_k - x^*$ decreases towards zero as the number of iterations k increases, where x^* is the true minimum.

A sequence of iterates $\{x_k\}$ is said to converge quadratically to x^* if the ratio of successive errors is bounded by a constant times the square of the previous error:

$$\lim_{k \to \infty} \frac{||x_{k+1} - x^*||}{||x_k - x^*||^2} = \lim_{k \to \infty} \frac{||e_{k+1}||}{||e_k||^2} = C.$$ C is positive and constant.

A sequence of iterates $\{x_k\}$ is said to converge linearly to x^* if the ratio of successive errors is bounded by a constant strictly less than 1:

$$\lim_{k \to \infty} \frac{||x_{k+1} - x^*||}{||x_k - x^*||} = \lim_{k \to \infty} \frac{||e_{k+1}||}{||e_k||} = r, 0 < r < 1$$

Quadratic Convergence requires the error to be proportional to the square of the previous error, whereas Linear Convergence requires the error to be proportional to the first power of the previous error.

2-11. عنوان بخش

Taylor expansion of f(x) around the minimizer x^*:

$$\Delta f(x) = \Delta f(x^*) + H(x^*)(x^* - x) + O(||x - x^*||^2)\ \Delta\, is\ gradient$$

Since x^* is a minimizer, the necessary condition is $\Delta f(x^*) = 0$ ($\Delta\, is\ gradient$). The expansion simplifies:

$$\Delta f(x) \approx H(x^*)(x^* - x)\ \Delta\, is\ gradient$$

The exact displacement vector from the current point x_k to the minimizer x^* is $(x^* - x_k)$.

From the approximation, we can solve for $(x^* - x_k)$:

$$-\Delta f(x_k) \approx H(x^*)(x^* - x_k)\ \Delta\, is\ gradient$$

Assuming $H^{-1}(x^*)$ is invertible (which it must be for a strict local minimum):

$$(x^* - x_k) \approx -H^{-1}(x^*)\Delta f(x_k)\ \Delta\, is\ gradient$$

$-H^{-1}(x_k)\Delta f(x_k)\ \Delta\, is\ gradient.$ this is the Newton step (p_k)

If x_k is sufficiently close to x^*, the Hessian matrix at x_k is a good approximation of the Hessian matrix at x^* , i.e., $H(x^*) \approx H(x_k)$..

Substituting this approximation into the Newton step definition:

$$p_k \approx -H^{-1}(x^*)\Delta f(x_k)\ \Delta\, is\ gradient$$

Comparing this with the last equation, we see:

$$(x^* - x_k) \approx p_k$$

پرسش 12 - عنوان پرسش

1-12. عنوان بخش

$\Delta f(x_1, x_2) = \Delta$ is gradient.

$$\begin{vmatrix} 400x_1^3 - 400x_1x_2 + 2x_1 - 2 \\ 200(x_2 - x_1^2) \end{vmatrix}$$

$H(x) =$

$$\begin{vmatrix} 1200x_1 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{vmatrix}$$

5-12. عنوان بخش

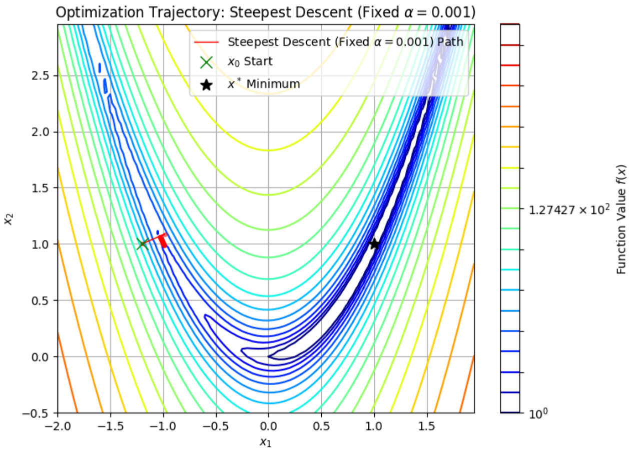

figure 6. steepest descent

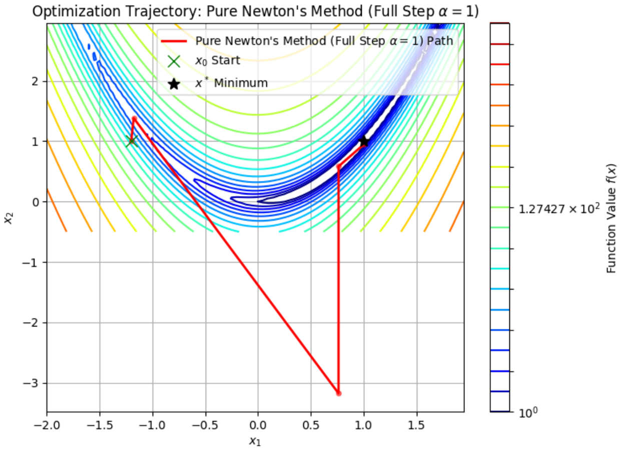

figure 7. pure Newton

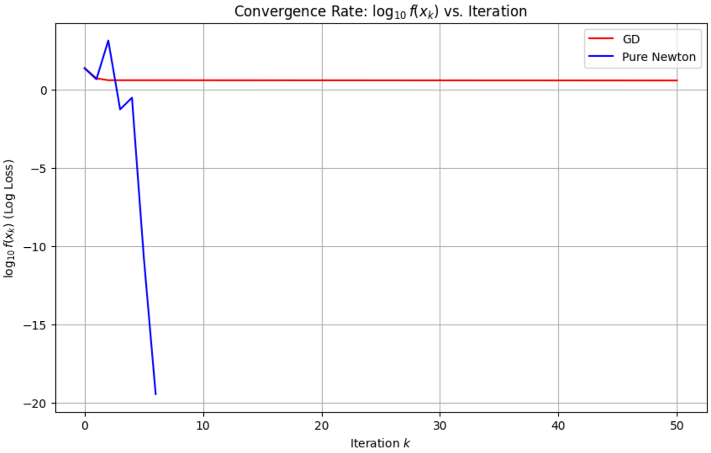

figure 8. Convergence rate

As we can see the Newton method converges in 6 iterations while GD is trapped in a local minimum and can’t escape it even with 50 iterations. It is fairly obvious that Newton is the better choice in this case.

پرسش 14 - عنوان پرسش

1-14. عنوان بخش

Two common methods for selecting the step size α_k (learning rate) in Gradient Descent are Backtracking (Armijo) Line Search and the Wolfe Conditions.

Armijo/Backtracking Line Search: This is an inexact line search that focuses on finding a step length that guarantees a sufficient decrease in the function value. It starts with an initial step size (e.g., $\alpha = 1$) and repeatedly multiplies α by a reduction factor (e.g., 0.5) until the Armijo condition (or sufficient decrease condition) is satisfied: $f(x^{(t)} + \alpha^{(t)}) \leq f(x^{(t)}) + c_1 \alpha \Delta f(x^{(t)})^T p^{(t)}$, where $c_1 \in (0, 1)$. This guarantees that the step is not too large, avoiding divergence or jumping over the minimum.

Wolfe Conditions (Strong or Weak): These are a set of two conditions (a sufficient decrease condition, similar to Armijo, and a curvature condition) that ensure the step length is not too short (Armijo condition) and not too long (curvature condition). The curvature condition ensures the slope at the new point is steeper than c_2 times the initial slope: $f(x^{(t)} + \alpha p^{(t)})^T p^{(t)} \geq c_2 \Delta f(x^{(t)})^T p^{(t)}$. This guarantees convergence for general descent methods. This guarantees that the algorithm is making progress and preventing extremely small steps near the minimum.

3-14. عنوان بخش

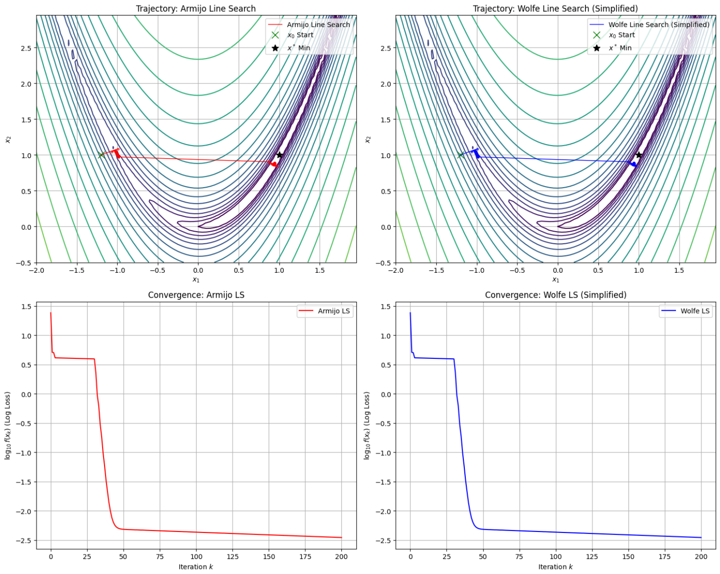

figure 9. Convergence rate and trajectory

As we can see both algorithms converged to the same solution.