AMSC808N-Final Exam-F1

Chenyang Fang
https://github.com/Chenyang-Fang/AMSC808N

December 21, 2020

1. In this problem, we want to approximate $g(x) = 1 - \cos x$ on the interval $[0, \pi/2]$ with the function $\textbf{Relu}(ax - b)$ where a and b are to be determined. And $\textbf{ReLU}(ax - b) = max(ax - b, 0)$. We can set up the loss function as:

$$f(a,b) = \frac{1}{12}\sum_{j=0}^{5}[ReLU(ax_j - b) - g(x_j)]^2. \tag{1}$$

The set of stationary points of f, i.e., the set of points where $\nabla f = 0$, consists of the global minimizer and a flat region. For the loss function, we need to calculate the gradient:

$$\nabla f = (\frac{\partial}{\partial a}f, \frac{\partial}{\partial b}f) \tag{2}$$

$$\frac{\partial}{\partial a}f = \begin{cases} \frac{1}{6}\sum_{j=0}^{5}[ReLU(ax_j - b) - g(x_j)][x_j] & ax_j - b > 0 \\ 0 & ax_j - b < 0 \end{cases} \tag{3}$$

$$\frac{\partial}{\partial b}f = \begin{cases} \frac{1}{6}\sum_{j=0}^{5}[ReLU(ax_j - b) - g(x_j)](-1) & ax_j - b > 0 \\ 0 & ax_j - b < 0 \end{cases} \tag{4}$$

We take 6 training points $x_j = \pi j/10, j = 0, 1, 2, 3, 4, 5$, and function $\textbf{ReLU(x)}$ is piecewise function. So we need to discuss the problem in different region.

- First, consider the condition $b > 0$. For the gradient $\nabla f = (\frac{\partial}{\partial a}f, \frac{\partial}{\partial b}f) = 0$, so $b > ax_j$. Then we can get that the flat region is $\Omega_1 = \{(a, b) : b > 0, b > a\pi/2\}$. Then the value of the loss function is $f(a, b) = \frac{1}{12}\sum_j g(x_j)^2 = 0.1405$.
- For the condition $b < 0$, first, consider the the region $\Omega_2 = \{b < 0, b < a\pi/2\}$. Then we can substitute the value into the gradient function (3) and (4), and get:

$$\frac{\partial}{\partial a}f = \frac{1}{6}\sum_{j=0}^{5}((ax_j - b) - g(x_j))(x_j) \tag{5}$$

$$\frac{\partial}{\partial b}f = \frac{1}{6}\sum_{j=0}^{5}((ax_j - b) - g(x_j))(-1) \tag{6}$$

By solving the gradient function $\nabla f = (\frac{\partial}{\partial a} f, \frac{\partial}{\partial b} f) = 0$, we can get

$$\frac{1}{6} a \sum_{j=0}^{5} x_j^2 - \frac{1}{6} b \sum_{j=0}^{5} x_j - \frac{1}{6} \sum_{j=0}^{5} x_j g(x_j) = 0, \tag{7}$$

$$\frac{1}{6} a \sum_{j=0}^{5} x_j - \frac{1}{6} \sum_{j=0}^{5} b - \frac{1}{6} \sum_{j=0}^{5} g(x_j) = 0. \tag{8}$$

By solving the equations, we can get the value of a and b are $a = 0.65$ and $b = 0.12$, which contradict with the assumption $b < 0$. So there is no stationary point in this region.

- Next, we need to discuss the condition $\{b > 0, b < a\pi/2\}$. Because we divide the interval $[0, \pi/2]$ into different sub-intervals, we need to discuss the piecewise function in their interval separately. Define the region $\Omega_j = \{b > 0, a(j-1)\pi/10 < b < aj\pi/10\}, j = 1, 2, 3, 4, 5$.

  - When $j = 1$, substitute the $j = 1$ in to the gradient function (7) and (8), then we can get the value of a and b, which is $a = 0.7646$ and $b = 0.252$, the value of b does not satisfies the equation $\{b > 0, a(j-1)\pi/10 < b < aj\pi/10\}$.
  - When $j = 2$, substitute the $j = 2$ in to the gradient function (7) and (8), then we can get the value of a and b, which is $a = 0.8613$ and $b = 0.3735$, and the value of b satisfies the equation $\{b > 0, a(j-1)\pi/10 < b < aj\pi/10\}$. And the value of the loss function is $3.6326 * 10 - 4$.
  - When $j = 3$, substitute the $j = 3$ in to the gradient function (7) and (8), then we can get the value of a and b, which is $a = 0.9355$ and $b = 0.4745$, and the value of b does not satisfies the equation $\{b > 0, a(j-1)\pi/10 < b < aj\pi/10\}$.
  - When $j = 4$, substitute the $j = 4$ in to the gradient function (7) and (8), then we can get the value of a and b, which is $a = 0.9836$ and $b = 0.5451$, and the value of b does not satisfies the equation $\{b > 0, a(j-1)\pi/10 < b < aj\pi/10\}$.
  - When $j = 5$, substitute the $j = 5$ in to the gradient function (7) and (8), then we can get two gradient functions are identical which is $a\pi/2 - b - 1 = 0$. Then region should be $\{b = a\pi/2 - 1, 4a\pi/10 < b < 5a\pi/10\}$. The value of the loss function is $f(a, b) = 0.0572$.

- Similarly, we need to consider the condition $\{b < 0, b < a\pi/2\}$, we can also discuss the piecewise function similar as previous section. Define the region $\Omega_j = \{b < 0, a(j-1)\pi/10 < b < aj\pi/10\}, j = 1, 2, 3, 4, 5$. Based on the procedure similar as previous section, we can find that there is no stationary points satisfied the conditions in this region.

In summary, based on the discussion, we can get that the flat region consists of three parts: (1).$\Omega_1 = \{(a, b) : b > 0, b > a\pi/2\}$. (2). $\Omega_2 = \{(a, b) : b = a\pi/2 - 1, 4a\pi/10 < b < 5a\pi/10\}$. (3). a single point $(a^* = 0.8613, b^* = 0.3735)$ which is the global minimizer, and the corresponding loss function value is $f(a, b) = 3.6327 * 10^{-4}$. I have plotted the region in the figure 1.

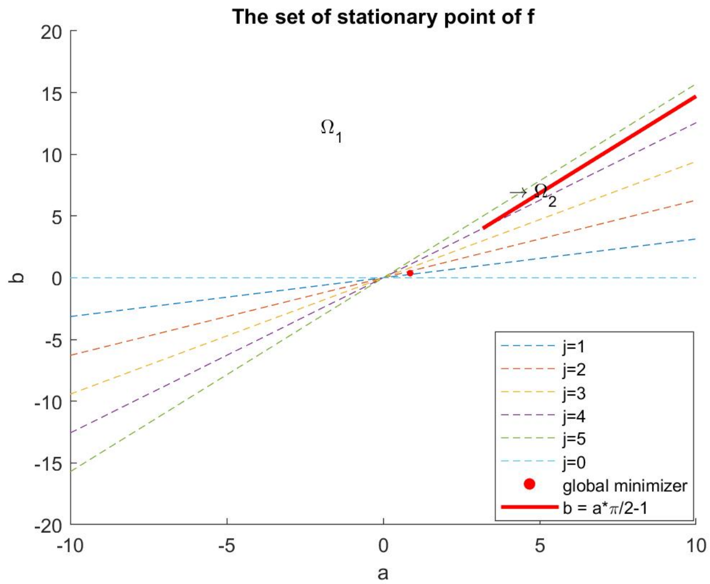

Figure 1: The set of stationary points of f.

2. In this problem, take $a = 1$ and $b = 0$ as the initial guess for the gradient descend with constant stepsize. Set the constant stepsize as α, then we can get the update function as:

$$a_{k+1} = a_k - \alpha \nabla_a f(a, b) \tag{9}$$

$$b_{k+1} = b_k - \alpha \nabla_b f(a, b) \tag{10}$$

Based on the calculation, we can get the gradient descend direction is

$$-\nabla f = -(\frac{\partial}{\partial a} f, \frac{\partial}{\partial b} f) = (-0.4109, 0.3949). \tag{11}$$

If we plot the direction on the plot of the stationary points, we can get the figure 2.

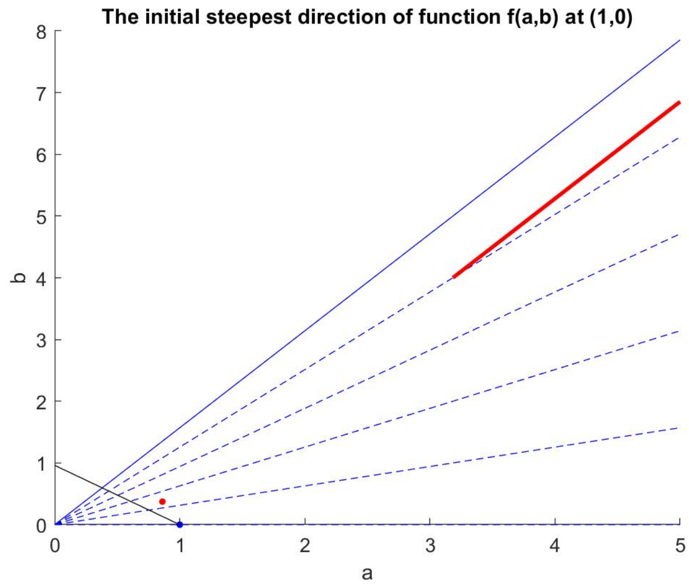

Figure 2: The direction of gradient descend of function f(a,b) at (1,0).

In this problem we want to calculate the minimal stepsize α^* such that the iterates end up in the flat region. In the figure 2, the black line is the direction of the gradient descent. So after we choose the initial guass as $(a, b) = (1, 0)$, the value of the loss function $f(a, b)$ will change along the direction of the gradient descent, which is depending on the stepsize. The region above solid blue line is the flat region. The function of the solid blue line is $b = a\pi/2$. So we can get the minimal stepsize α^* which can make the (a_1, b_1) lies on the solid blue line by solving:

$$\begin{cases} (a_1, b_1) = (a_0, b_0) - \alpha^* \nabla f(a_0, b_0) \\ b_1 = a_1 * \pi/2 \end{cases} \tag{12}$$

Then we can get:

$$b_0 - \alpha^* f_b(1, 0) = \frac{\pi}{2}(a_0 - \alpha^* f_a(1, 0) \tag{13}$$

$$\alpha^* = \frac{-\frac{\pi}{2}}{f_b(1, 0) - \frac{\pi}{2} f_a(1, 0)} \approx 1.5099 \tag{14}$$

So $\alpha^* = 1.5099$ is the minimal stepsize that the iterates end up in the flat. To check the accuracy of the theoretical result, a numerical experiment has been done in the matlab. In the matlab, set the tolerance as 10^{-3}, because in the part 1, we already know the value of loss function in the flat region is $f(a, b) = 0.1405$, so we start at the initial guess $(a, b) = (1, 0)$, each time, I compute the value of loss function, then compare with the theoretical result. If the error is larger than the tolerance, then gradient descend method will be used to calculate the direction of steepest descent and (a,b) will move along the steepest descent direction. If the error is smaller than the error, then the value of (a, b) will not change.

First, I set the stepsize as the minimal stepsize α^*, and plot the final result in the figure 3. From the figure 3, it is illustrated that after the first iteration, the function reach the flat region, and the value of loss function $f(a, b)$ does not change anymore.

Second, I set the stepsize as the $\alpha = 0.99 * \alpha^*$ and set the iteration number as 5000. Then I plottd the final result in the figure 4. From the figure 4, we can clearly get that the iterates did not end up in the flat region. The value of loss function is oscillating in a region. Also, the iterates did not approach the global minimizer. The numerical results are consistent with the theoretical result, which means the theoretical result is correct.

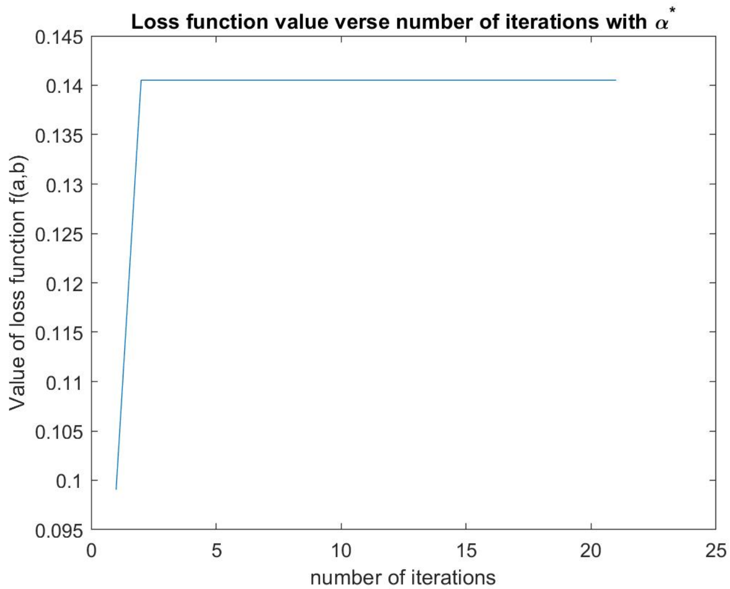

Figure 3: Loss function value verses number of iteration when the stepsize is α^*.

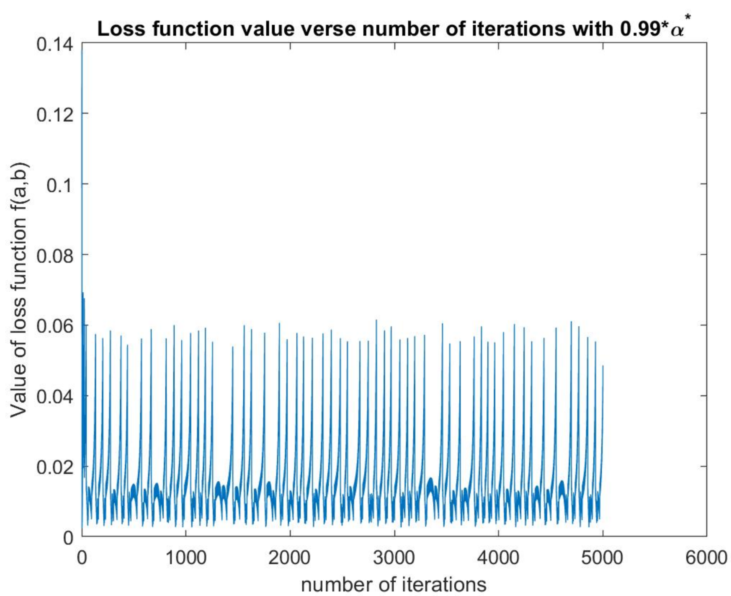

Figure 4: Loss function value verses number of iteration when the stepsize is $0.99 * \alpha^*$.

Next step, we need to propose a stepsize trying to make it as large as possible such that the iterates will necessarily converge to the global minimizer. In the the D Binder's notes, it is illustrated that consider gradient descent with a fixed step size α for the quadratic model problem $\phi(x) = \frac{1}{2}x^T Ax + b^T x + c$, where A is symmetric positive definite. The iteration converges provided $\alpha < 2/\lambda_{max}$, and the optimal α is

$$\alpha^* = \frac{2}{\lambda_{min} + \lambda_{max}}, \tag{15}$$

which leads to the spectral radius

$$1 - \frac{2\lambda_{min}}{\lambda_{min} + \lambda_{max}} = 1 - \frac{2}{\kappa(A)} \tag{16}$$

where $\kappa(A) = \lambda_{max}/\lambda_{min}$ is the condition number for the matrix A. Because of the **ReLU** function, the loss function is not a strictly quadratic function in the whole interval. However, according to the calculation, the initial guess is in the neighbor of the global minimizer $(0.8613, 0.3735)$. So we can use this lemma to calculate the stepsize. So the Hessian matrix of the loss function is

$$\nabla^2 f(a,b) = \begin{bmatrix} \frac{1}{6}\sum_{j=0}^{5} x_j^2 I & -\frac{1}{6}\sum_{j=0}^{5} x_j I \\ -\frac{1}{6}\sum_{j=0}^{5} x_j I & -\frac{1}{6}\sum_{j=0}^{5} I \end{bmatrix} \tag{17}$$

where I is the index function, $I = 1$ if $a * x - b > 0$, and $I = 0$ if $a * x - b < 0$. Then we can calculate the eigenvalues of the Hessian matrix. Then we can get the largest stepsize that the iterates will converge to the global minimizer:

$$\alpha < \frac{2}{\lambda_{max}} \to \lfloor \alpha \rfloor \approx 1.3 \tag{18}$$

To evaluate the theoretical result, I also check it in the matlab. The procedure is also same as before except I set the constant stepsize as 1.3 and number of iteration as 500. Then I plotted the result in the figure 5. It is shown that the iterates will converge to the global minimizer. The figure 6 is the trajectory of the iterates. Finally the loss function will get its global minimizer.

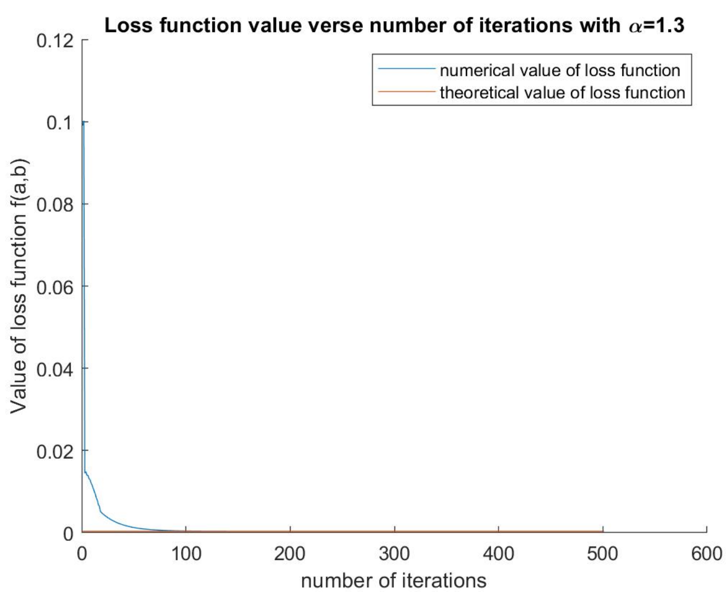

Figure 5: Loss function value verses number of iteration when the stepsize is $\alpha = 1.3$.

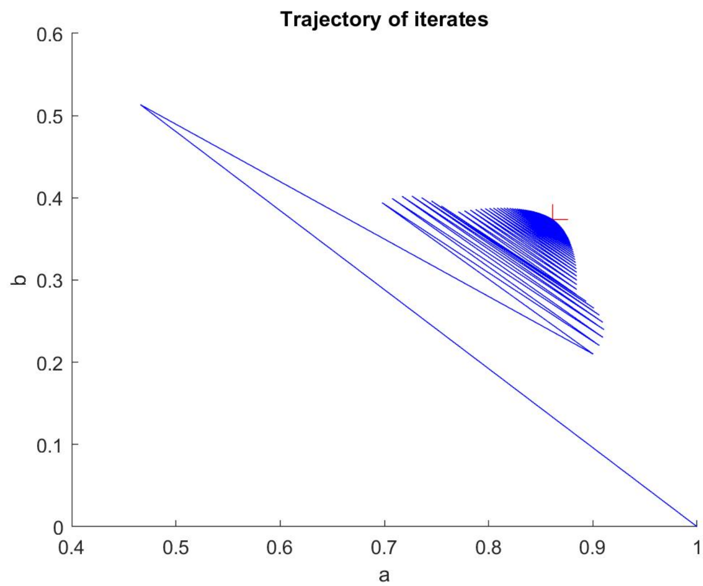

Figure 6: Trajectory of the iterates.

3. In this problem, we need take $(a, b) = (1, 0)$ as the initial guess and use a simple stochastic gradient descend with a single training point chosen randomly for approximating the gradient of f at each step. In the matlab, the basic procedure is same as before except there is a small change. Each time, I randomly chose a j as a single training point rather than using all of the nodes to calculate the gradient.
   For the strategy of the stepsize, the lecture note 2 gives a detailed illustration. In order to achivev convergence, we need to reduce stepsize as we progress but not too fast: a condition for stepsizes α_k is
   $$\sum_{k=0}^{\infty} \alpha_k = \infty, \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty \tag{19}$$
   So, based on the requirement, my strategy for the stepsize is
   $$\alpha_k = \alpha_0/(1 + i/100) \tag{20}$$
   where α_0 is a constant parameter and in the matlab code, $\alpha_0 = 1.5$. i is the iteration number. This strategy works in the previous homework and project. After implementing the matlab procedure, I plotted the final result in the figure 7. Form the figure 7, it is shown that the iterates will converge to the global minimizer. The figure 8 shows the trajectory of the iterates.

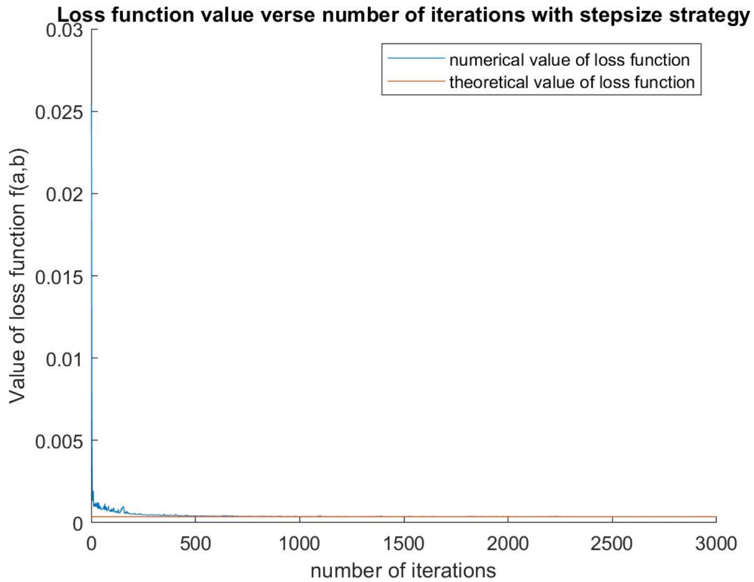

Figure 7: Loss function value verses number of iteration when the stepsize strategy.

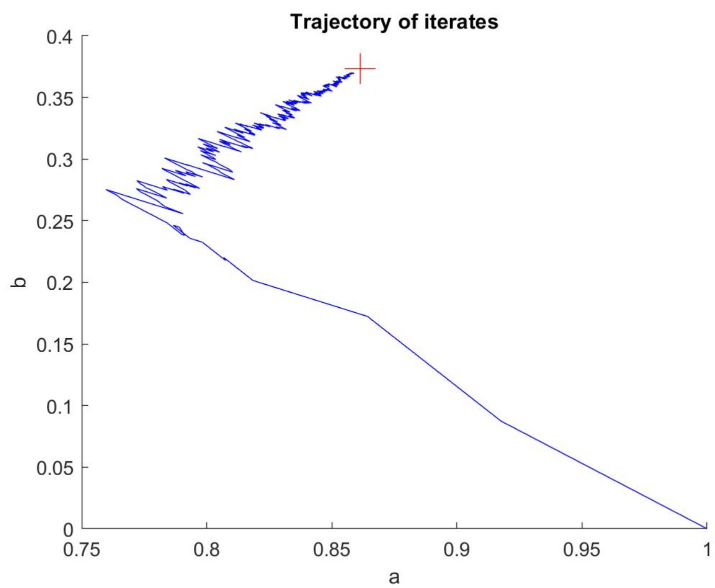

Figure 8: Trajectory of the iterates.

In summary, here are some comments about the stepsize strategy:

- Compared with the fixed stepsize strategy, my stepsize strategy can converge quickly. In the figure 5 and figure 7, we can get that for my stepsize strategy, the loss function can decrease quickly.
- For my stepsize strategy, the parameters in the formula need to be tuned for different equations. In this problem, the loss function is a piecewise function within the interval. It might not work well in other region which is far away from the global minimizer.