Gradient Descent using Fixed Point Theorem

Sing Kuang Tan
Email: singkuangtan@gmail.com

February 8, 2023

Abstract

In this paper, I am going to propose a gradient descent algorithm using fixed point theorem. Fixed point theorem comes from topology mathematics. This gradient descent is able to converge at exponential rate, faster than Netwon method which converges at quadratic rate. Besides that, it does not need second order derivatives. The algorithm is simple and can be implemented using a few lines of equations. It can be used for training Relu deep learning network.

1 Introduction

Gradient descent is a slow process which takes many iterations to find the minnimum. Newton method is faster but it has some issues such as numerical instability if 2nd order derivatives Hessian is very small, in some cases it does not converge, some cost functions do not have 2nd order derivatives so they cannot use Netwon method, and the Hessian matrix is very expensive to compute. Even with quadratic convergence rate, Netwon method is still too slow for optimizing many cost function. My gradient descent using fixed point theorem can converge at exponential rate. Beside that, there are some cost functions with no second order derivatives (e.g. Relu deep learning network), Newton method cannot be executed.

Compared to a simple gradient descent, my gradient descent method does not need to choose the step size. A small step size is inputed, the step size will increase automatically as it progress through the fixed point iterations.

Fixed point theorem for one dimension case is modeled as the output of a function is feedbacked into its input. This process is continued until the final output converge to a single value.

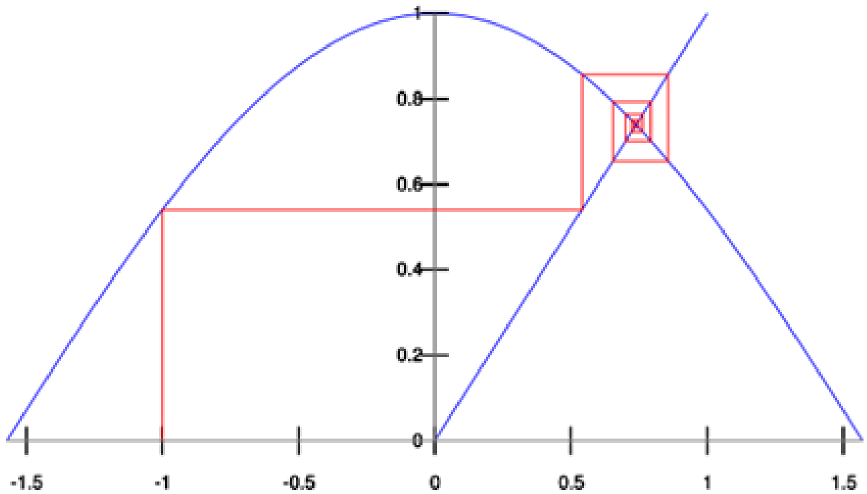

Figure 1: Fixed point theorem example in one dimension

Figure 1 shows an example of a set of fixed point theorem iterations in one dimension. Horizontal axis represent the x variable and vertical axis represents the y variable. Let $f(x) = cos(x)$ be the function of the N shape curve in the figure. Let $g(x) = x$ be the function of the blue line function in the figure. Let $x = -1$.

$$f(-1) = 0.5403 \tag{1}$$

The output of function $f(x)$ is feedbacked into its input.

$$g(0.5403) = 0.5403 \tag{2}$$

The output of $g(x)$ is inputed into $f(x)$.

$$f(0.5403) = 0.8576 \tag{3}$$

The process is continued until convergence. The points (x,y)=(-1,0.5403) and (0.5403,0.8576) represents the input and output of the functions.

$$\begin{aligned} y_1 = f(-1) &= 0.5403 \\ y_2 = f(g(f(-1))) &= 0.8576 \\ y_3 = f(g(f(g(f(-1))))) &= 0.6543 \\ ... & \\ y_\infty = f(g(...f(g(f(-1)))...)) &= 0.7391 \end{aligned} \tag{4}$$

$y_\infty = 0.7391$ is the converged final y value of infinite applications of $g()$ and $f()$ functions.

The line function $g(x)$ is the reflector. By placing the line at the correct position and orientation, fixed point theorem can do gradient descent.

2 My Model

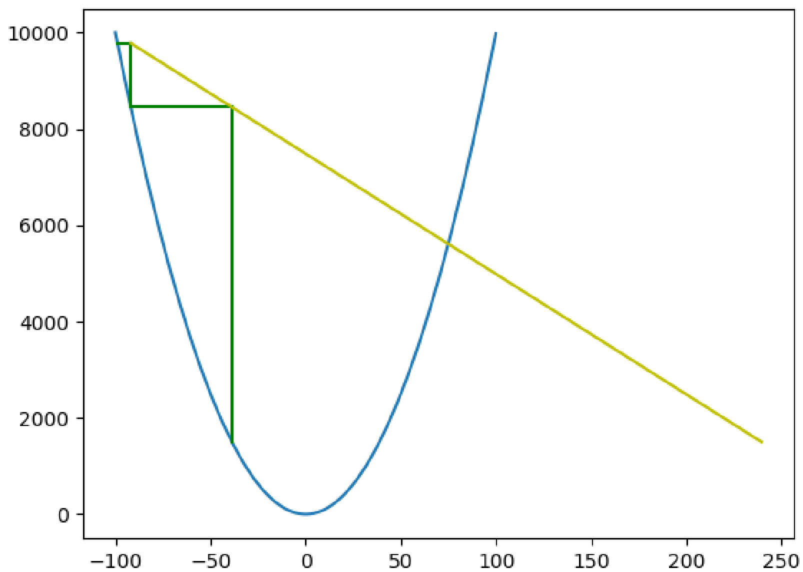

Figure 2: First set of iterations of gradient descent using fixed point theorem in action with the linear function g(x) drawn at the correct place

Figure 2 shows the first set of iterations of my gradient descent method. The yellow line is drawn downward from $x = -100$ where the gradient is less than the gradient at $f(x = -100)$. The steps of my gradient descent is shown as the green line. The descent will stop when the green line goes below the blue line, or the descrease of current step of gradient descent is less than the previous step descrease.

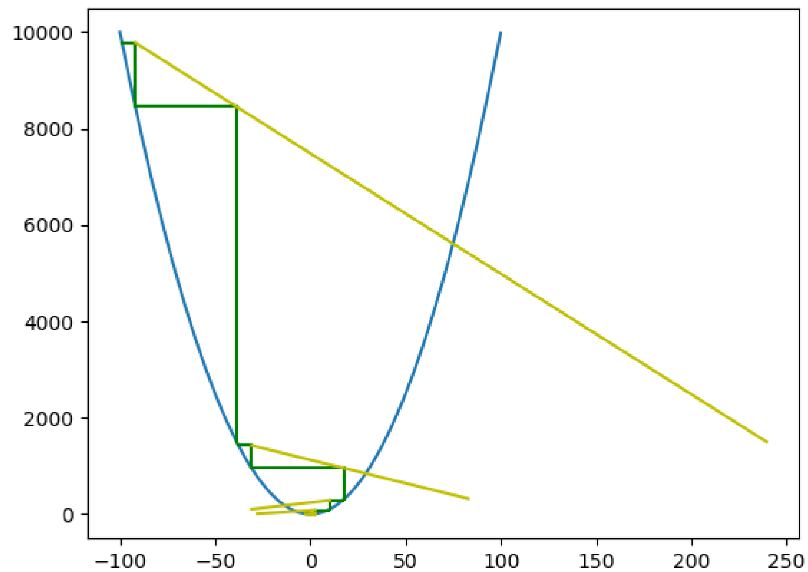

Figure 3: Gradient descent using fixed point theorem in action with the linear function g(x) drawn at the correct place

Figure 3 shows an example of my gradient descent using fixed point theorem in action. The steps of my gradient descent is shown as the green line. The starting point is $x = -100$. The line $g(x)$ is drawn at the starting point downwards, with a absolute gradient less than the actual gradient of the starting point. Fixed point iteration is started from an x position equal to the starting point plus a small step size. The iteration is continued until the decrease of current step of gradient descent is smaller than the decrease of previous step.

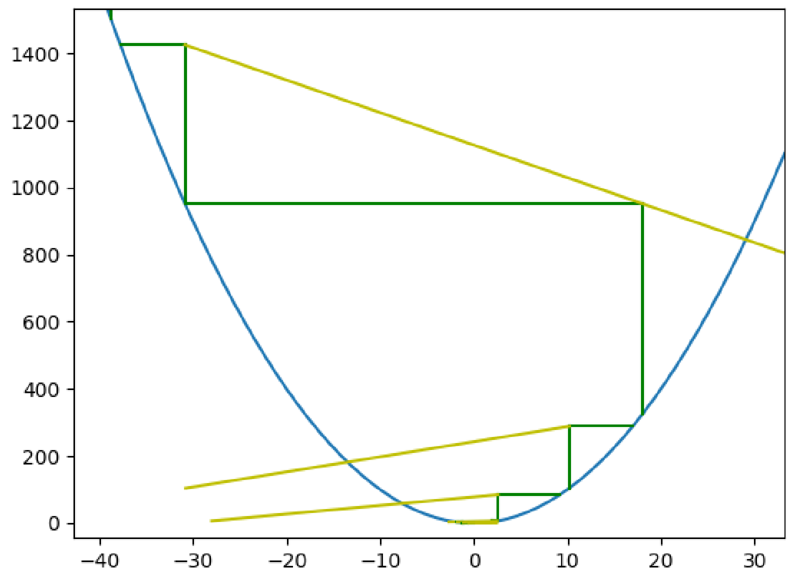

Figure 4: Second set of fixed point theorem iterations for gradient descent

Another line of $g(x)$ is drawn and gradient descent is continued until convergence (see figure 4). The starting point is $x = -38$.

3 Case where cost function is non-convex

My gradient descent using fixed point theorem can even work on cases where the cost function is non-convex.

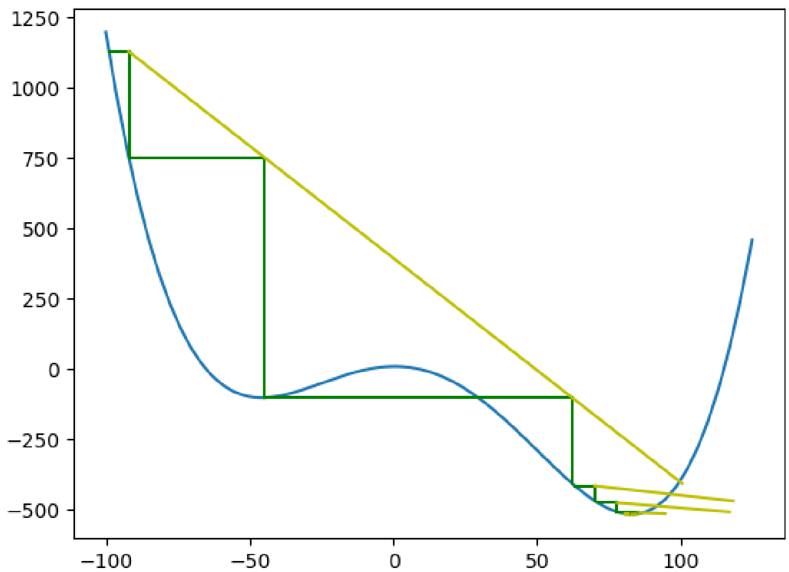

Figure 5: Gradient descent using fixed point theorem on non-convex function

From figure 5, we can see that as long as our line function $g(x)$ is above the cost function $f(x)$ during fixed point iterations, it can do gradient descent on non-convex function and find the minimum value of the function.

4 Proof of exponential convergence of gradient descent using fixed point theorem

Let $g(x) = -x$. Let $f(x) = -2x$.

Doing the fixed point iterations,

$$\begin{aligned} y_1 &= f(1) &&= -2 \\ y_2 &= f(g(f(1))) &&= -4 \\ y_3 &= f(g(f(g(f(1))))) &&= -8 \\ &... \\ y_n &= f(g(...f(g(f(-1)))...)) &&= -2^n, \end{aligned} \tag{5}$$

we can prove that y_n decrease exponentially as -2^n.

5 Gradient descent on high dimensional problem

For a cost function that has 2 or higher dimensions, my gradient descent using fixed point theorem will draw a line along the direction of the gradient. It will perform fixed point theorem iterations on the line. Then another line along the gradient at the point of current step will be drawn. Another fixed point theorem iterations are carried out and so on.

6 Experiment Results

To compare with Newton method, I will use the function $f(x) = 1/(x + 101) + 1/(-x + 101) - 0.01 \times x$. This is a one dimensional interior point method. The terms $1/(x + 101)$ and $1/(-x + 101)$ are the barrier functions to represent the hard constraints of left and right boundaries of the linear programming. The term $-0.01 * x$ is the cost function of the linear programming.

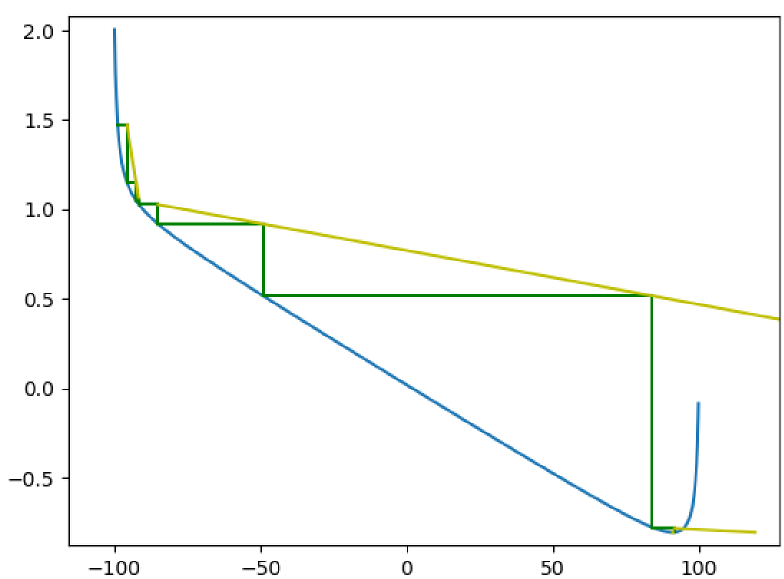

Figure 6: One dimensional interior point method cost function solved using my fixed point theorem gradient descent

Figure 6 shows how my fixed point theorem gradient descent solved the one dimensional interior point method cost function. In less than 10 iterations, my gradient descent has reached the minimum of the cost function.

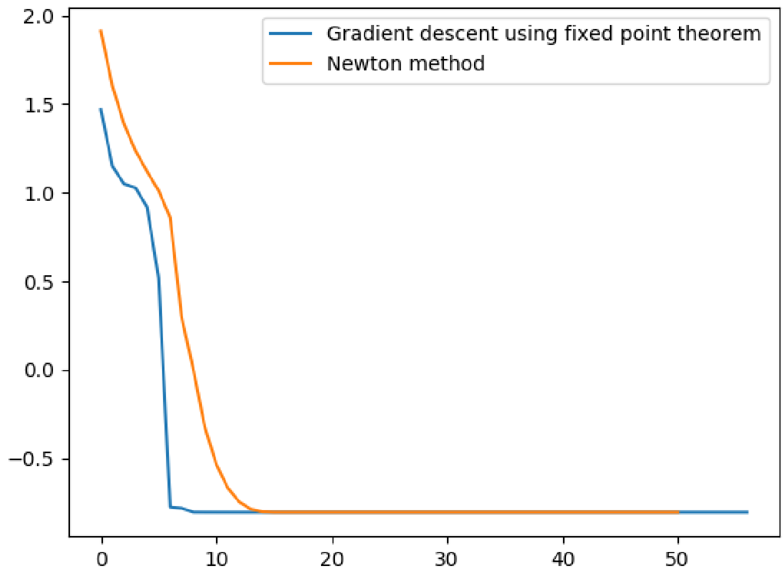

Figure 7: Comparsion of number of iterations to converge with gradient descent using fixed point theorem and Netwon method

From figure 7, we can see that my gradient descent algorithm using fixed point theorem converges faster than Newton method empirically.

Access my GitHub codes thru this link: https://github.com/singkuangtan/fixedpointdesc

7 Conclusion

I have shown that my gradient descent using fixed point theorem is better than Newton method in both theoretical and empirical ways. My gradient descent is the only method that has theoretical exponential convergence rate. I believe it can be used to speed up many optimization applications including deep learning.