CS559 Neural Networks HW #4

Author: Kai Bonsol

1.

In this computer experiment I implemented the gradient descent method and Newton's method on f(x,y) = -log(1-x-y) – logx – logy on the domain {(x, y): x+ y < 1, x > 0, y > 0}.

Choosing w in the domain randomly as: [0.5507979   0.31810149], and learning rate initially 1.

If an update during GD/newton's method moved the point out of the domain where f was undefined, I reinitialized w randomly within the domain and halved the learning rate.

a. **Gradient & Hessian calculations:**

09/29/2021
CS 559 hw #4
1. b

$$f(x,y) = -\log(1-x-y) - \log x - \log y$$

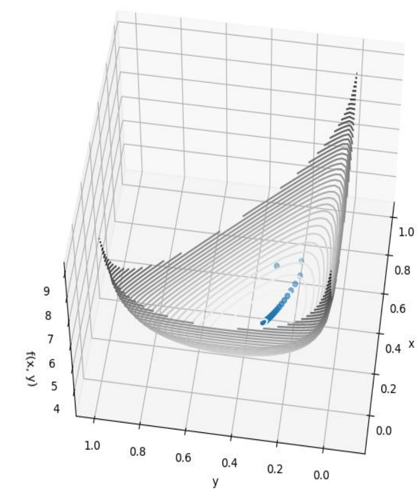
$$\mathbb{D} = \{(x,y) : x+y<1, x>0, y>0\}$$

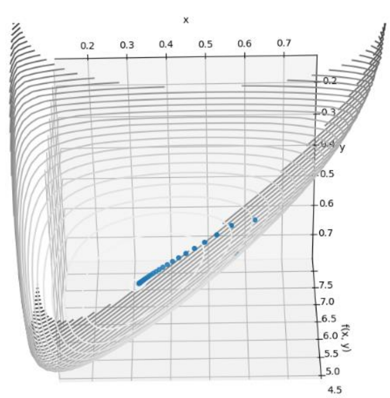
$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{1}{1-x-y} - \frac{1}{x} \\ \frac{1}{1-x-y} - \frac{1}{y} \end{bmatrix}$$

$$H_{f(x,y)} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial xy} \\ \frac{\partial^2 f}{\partial xy} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{(1-x-y)^2} + \frac{1}{x^2}, & \frac{1}{(1-x-y)^2} \\ \frac{1}{(1-x-y)^2}, & \frac{1}{(1-x-y)^2} + \frac{1}{y^2} \end{bmatrix}$$

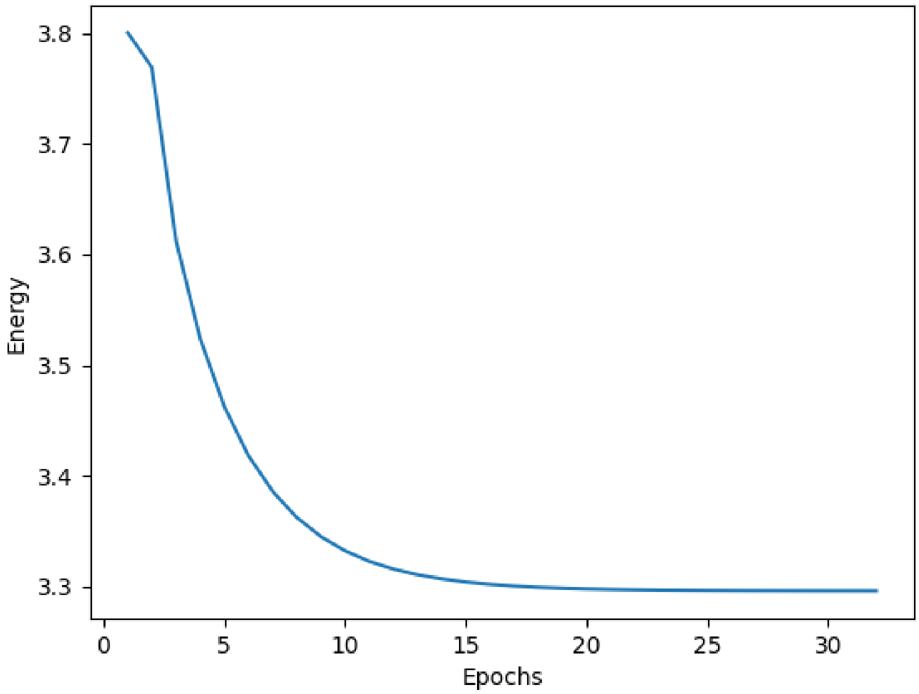
b. Newton's method:

Plotting the energy that each x,y point achieves for each iteration until it reaches convergence at the estimated global minimum:



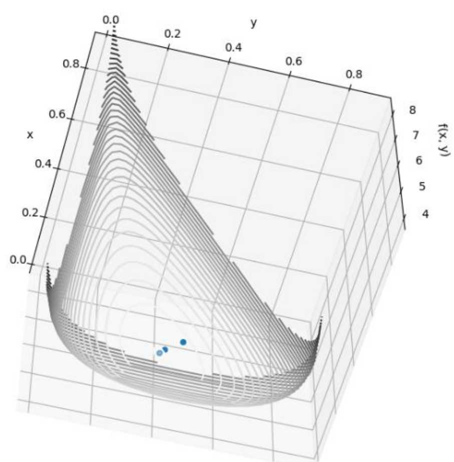


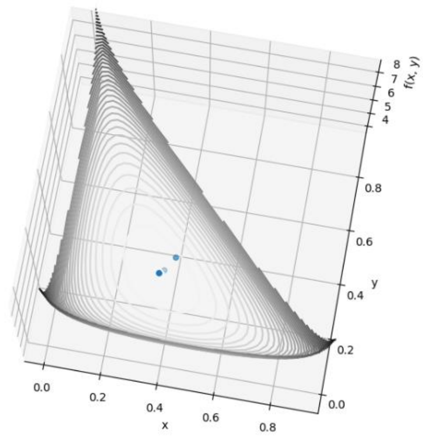
Plotting energy with respect to iterations:

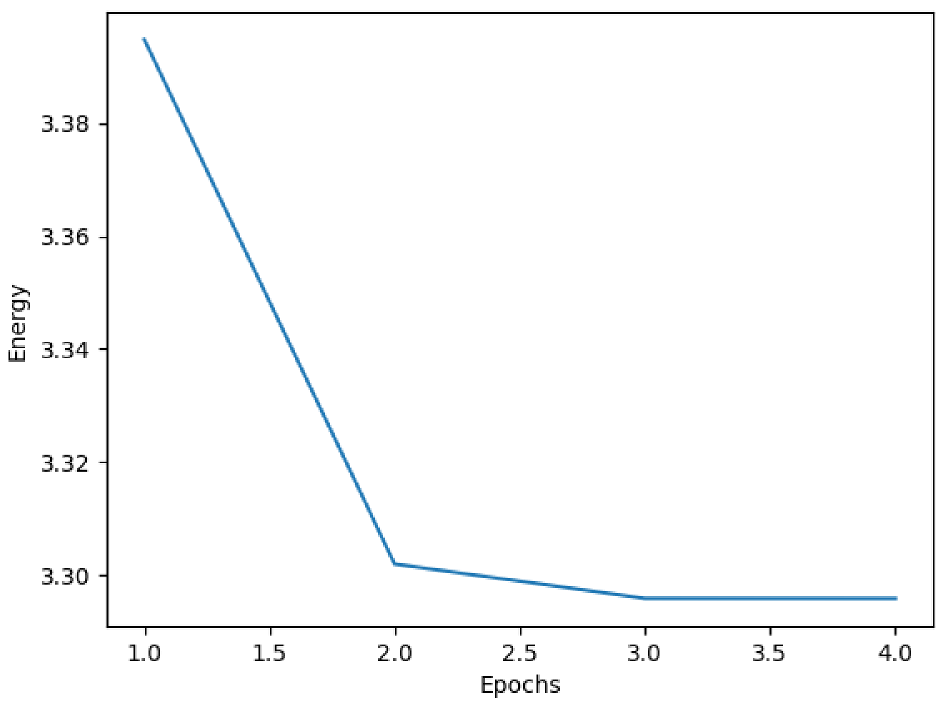


c. Newton's method:

Using the same initial w, Newton's method needs significantly less iterations to converge:







d.

Newton's method only needs 4 iterations to find the estimated global minimum whereas gradient descent needed about 30 iterations to converge. Both converge to the same global minimum, off only by the threshold tolerance of 1e-03.

2.

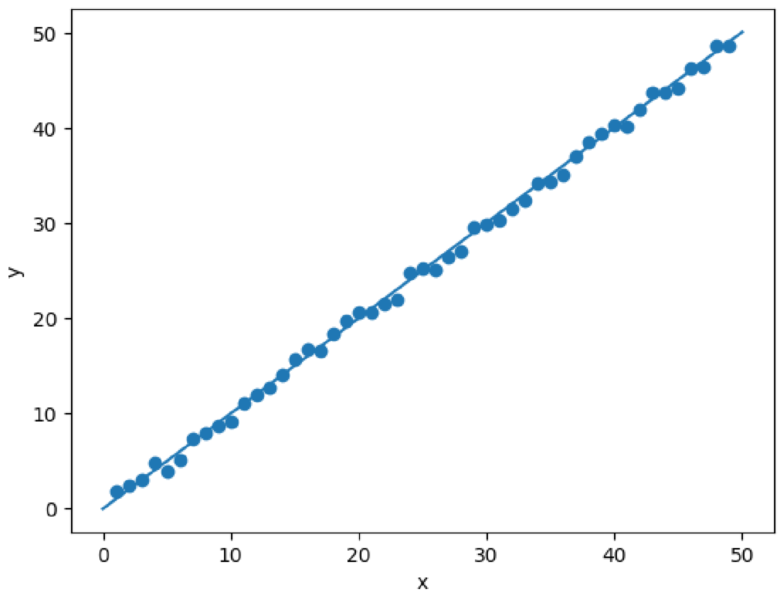
Now a linear regression exercise, creating a dataset with

X = I, for I = 1,….,50 and Y = I + uI, where uI is some noise sampled between -1 and 1.

We can quickly calculate the linear least squares fit by the closed form solution

$W = XY^{+}$ where $^{+}$ indicates the pseudo-inverse.

Achieving an intuitive W = [0.99], we can plot the best fit line with the data:



2. e

$$f(w_0, w_1) = \sum_{i=1}^{50} \left(y_i - (w_0 + w_1 x_i)\right)^2$$

$$\nabla f(w_0, w_1) = \begin{bmatrix} \frac{\partial f}{w_0} \\ \frac{\partial f}{w_1} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{50} \frac{\partial}{\partial w_0} \left(y_i - (w_0 + w_1 x_i)\right)^2 \\ \sum_{i=1}^{50} \frac{\partial}{\partial w_1} \left(y_i - (w_0 + w_1 x_i)\right)^2 \end{bmatrix}$$

$$= \begin{bmatrix} -\sum_{i=1}^{50} 2\left(y_i - (w_0 + w_1 x_i)\right) \\ -\sum_{i=1}^{50} 2\left(y_i - w_0 - w_1 x_i\right) x_i \end{bmatrix} = \left(-2 \sum_{i=1}^{50} \left(y_i - w_0 - w_1 x_i\right)\right) \vec{X}$$

$$= \left(-2 \sum_{i=1}^{50} \left(y_i - w^T x_i\right)\right) \vec{X}$$