Project 1

Neethu Renjith
neethur

1. **Description of the method(s):** I tried 2 different methods :
    1. Adam Optimization - a variation of gradient descent
    2. bfgs – a Quasi-Newton algorithm

Gradient Descent:

The direction of steepest descent is the direction opposite the gradient. The update rule is :

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)}$$

Where alpha is the learning rate and d is the descent direction.

The update, starting from a random point, is done till convergence criterion is satisfied.

Adam optimizer:

Adam is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. It uses an exponentially decaying squared gradient to remove sensitivity to learning rate and an exponentially decaying gradient to model moemtum.

The update step is as follows.

$$\begin{aligned}
\text{biased decaying momentum: } \mathbf{v}^{(k+1)} &= \gamma_v \mathbf{v}^{(k)} + (1-\gamma_v)\mathbf{g}^{(k)} \\
\text{biased decaying sq. gradient: } \mathbf{s}^{(k+1)} &= \gamma_s \mathbf{s}^{(k)} + (1-\gamma_s)\left(\mathbf{g}^{(k)} \odot \mathbf{g}^{(k)}\right) \\
\text{corrected decaying momentum: } \hat{\mathbf{v}}^{(k+1)} &= \mathbf{v}^{(k+1)} / (1-\gamma_v^k) \\
\text{corrected decaying sq. gradient: } \hat{\mathbf{s}}^{(k+1)} &= \mathbf{s}^{(k+1)} / (1-\gamma_s^k) \\
\text{next iterate: } \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} - \alpha \hat{\mathbf{v}}^{(k+1)} / \left(\epsilon + \sqrt{\hat{\mathbf{s}}^{(k+1)}}\right)
\end{aligned}$$

Quasi Newton methods:

These methods approximate the inverse Hessian (Q) and update as:

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{Q}^{(k)}\mathbf{g}^{(k)}$$

where alpha is step factor found using backtracking line search and Q*g forms the quasi-Newton direction.

Bfgs:

The hessian is calculated as a running update starting from identity matrix.

$$H_{k+1} = H_k + \frac{yy^T}{y^T s} - \frac{H_k s s^T H_k}{s^T H_k s}$$

where

$$s = x_{k+1} - x_k, \qquad y = \nabla f(x_{k+1}) - \nabla f(x_k)$$

Inverse update

$$H_{k+1}^{-1} = \left(I - \frac{sy^T}{y^T s}\right) H_k^{-1} \left(I - \frac{ys^T}{y^T s}\right) + \frac{ss^T}{y^T s}$$

Convergence criterion:

Initially chose to use relative change in function value to be my criteria for both approaches

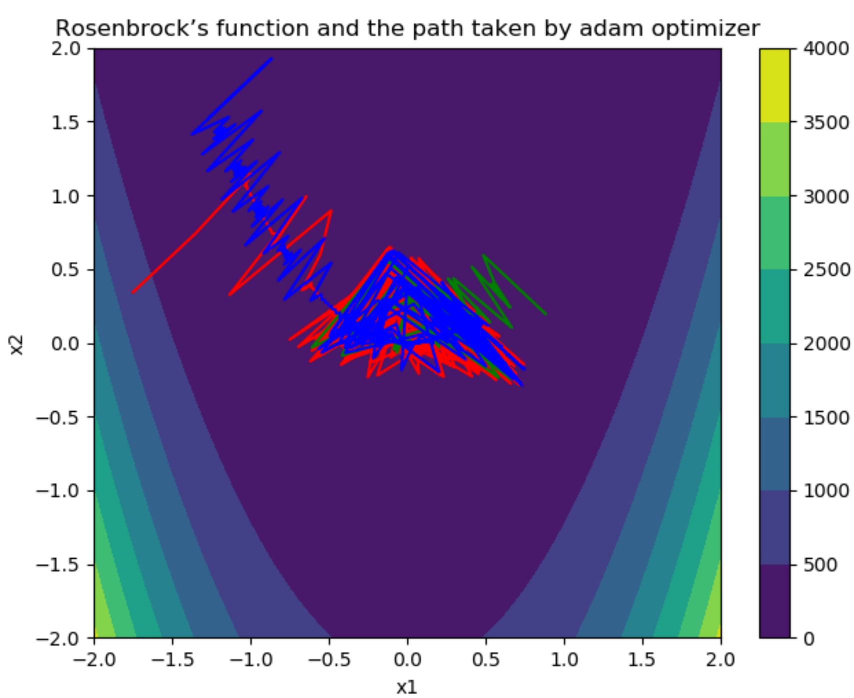
If $\|f^k(x) - f^{k+1}(x)\| / \|f^k(x) + eps\| < threshold$ then stop

Where eps is just a small value (1e-8) to prevent division by zero.

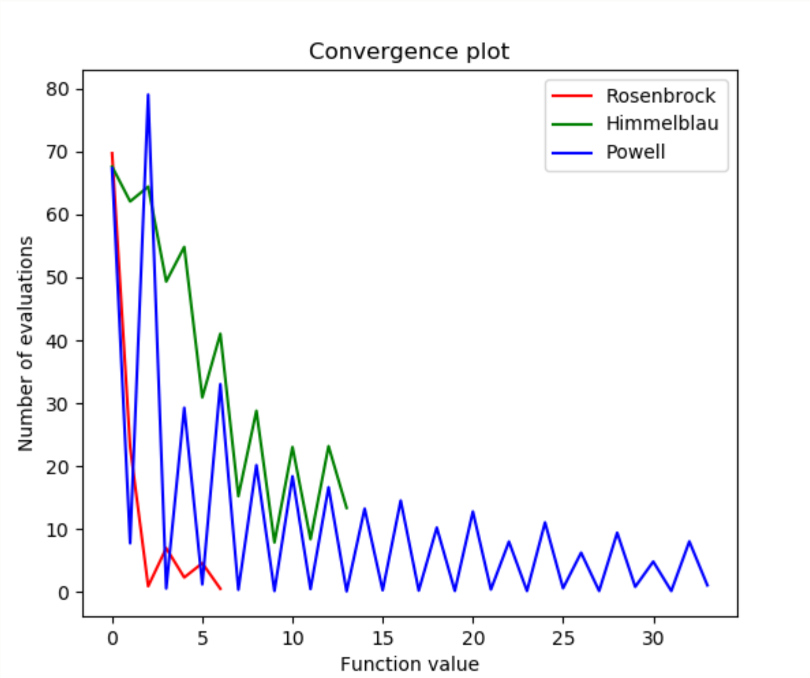
2. Plots (using Adam):

The plots for bfgs is not included as they did not pass test cases.

Plots showing the path for Rosenbrock's function with the objective contours and the path taken by algorithm from three different starting points.



Convergence plots for the three simple functions (Rosenbrock's function, Himmelblau's function, and Powell's function).



References:

1. Algorithms for Optimization; Mykel J. Kochenderfer, Tim A. Wheeler
2. http://www.seas.ucla.edu/~vandenbe/236C/lectures/qnewton.pdf