HW1
236330 – Introduction to Optimization and Deep Learning

Nerya Hadad
Yair Rada

1. **analytical differentiation:**

   gradient and hessian calculation:

Task 1:

$\underline{f_1(x) = \phi(Ax)}$

Define:

$f(x) = \varphi(u)$ Such that $u = Ax$

$du = A \cdot dx$

Gradient:

$df = \nabla^T \varphi(u) \cdot du$

$df = \nabla^T \varphi(u) \cdot A \cdot dx$

$df = (A^T \cdot \nabla\varphi(u))T \cdot dx$

Using the external definition $df = g(x)^T \cdot dx$:

$\boldsymbol{g_{f1} = A^T \nabla \phi(Ax)}$

Hessian:

$dg = A^T \cdot d\nabla\varphi(Ax)$

$dg = A^T \cdot H_\varphi(Ax)Adx$

Using the external definition $dg = H(x) \cdot dx$:

$\boldsymbol{H_\varphi(x) = A^T \cdot H_\varphi(Ax)A}$

Task 2:

$\underline{f_2(x) = h(\varphi(x))}$

Gradient:

$$df = \nabla h^T(\varphi(x)) \cdot d\varphi = \nabla h^T(\varphi(x)) \cdot \nabla\varphi^T \cdot dx$$

Using the external definition $df = g(x)^T \cdot dx$:

$$\boldsymbol{g(x) = \nabla\varphi(x) \cdot \nabla h(\varphi(x))}$$

Hessian:

$$\begin{aligned} dg &= d\nabla\varphi(x) \cdot \nabla h(\varphi(x)) + \nabla\varphi(x) \cdot d\nabla h(\varphi(x)) \\ &= H_\varphi(x) \cdot dx \cdot \nabla h(\varphi(x)) + \nabla\varphi(x) \cdot H_h(\varphi(x)) \cdot d\varphi \end{aligned}$$

$d\varphi = \nabla\varphi^T \cdot dx$ =>

$$dg = H_\varphi(x) \cdot dx \cdot \nabla h(\varphi(x)) + \nabla\varphi(x) \cdot H_h(\varphi(x)) \cdot \nabla\varphi^T \cdot dx$$

h: R->R is a scalar function, hence it's derivative in the point $\varphi(x)$ is a scalar too =>

$$\begin{aligned} dg &= H_\varphi(x) \cdot \nabla h(\varphi(x)) \cdot dx + \nabla\varphi(x) \cdot H_h(\varphi(x)) \cdot \nabla\varphi^T \cdot dx \\ &= \left[H_\varphi(x) \cdot \nabla h(\varphi(x)) + \nabla\varphi(x) \cdot H_h(\varphi(x)) \cdot \nabla\varphi^T\right] \cdot dx \end{aligned}$$

Using the external definition $dg = H(x) \cdot dx$:

$$\boldsymbol{H_\varphi(x) = H_\varphi(x) \cdot \nabla h(\varphi(x)) + \nabla\varphi(x) \cdot H_h(\varphi(x)) \cdot \nabla\varphi^T}$$

Task 3:

$$\varphi\left(\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\right)=\sin(x_1x_2x_3)$$

$$\bar{x}=\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}$$

$$df(\bar{x})=\nabla\sin(prod(\bar{x}))^t\,dx$$

$$\nabla\sin(prod(\bar{x}))=\begin{pmatrix}x_2x_3\\x_1x_3\\x_1x_2\end{pmatrix}\cos(prod(\bar{x}))$$

using the external definition $df=g(x)^t dx$:

$$g(x)=\begin{pmatrix}x_2x_3\\x_1x_3\\x_1x_2\end{pmatrix}\cos(prod(\bar{x})) \quad \Big| \quad prod(\bar{x})=x_1x_2x_3$$

$$\nabla g(x)=\begin{pmatrix}0&x_3&x_2\\x_3&0&x_1\\x_2&x_1&0\end{pmatrix}\cdot\cos(prod(\bar{x}))-\begin{pmatrix}x_2x_3\\x_1x_3\\x_1x_2\end{pmatrix}\begin{pmatrix}x_2x_3\\x_1x_3\\x_1x_2\end{pmatrix}^t\cdot\sin(prod(\bar{x}))$$

since $\nabla g(x)$ is symetric($\nabla g(x)^t=\nabla g(x)$) :

$dg=\nabla g(x)^t dx=\nabla g(x)dx$

using the external definition $dg=H(x)\,dx$:

$$H(x)=\begin{pmatrix}0&x_3&x_2\\x_3&0&x_1\\x_2&x_1&0\end{pmatrix}\cos(prod(\bar{x})-\begin{pmatrix}x_2x_3\\x_1x_3\\x_1x_2\end{pmatrix}\begin{pmatrix}x_2x_3\\x_1x_3\\x_1x_2\end{pmatrix}^t\cdot\sin(prod(\bar{x}))$$

$$h(x) = \exp(x)$$

$$dh = \nabla(e^x)^t\,dx = (e^x)^t\,dx$$

using the external definition $dh = g(x)^t\,dx$:

$$g(x) = e^x = h(x)$$

$$dg = dh = (e^x)^t\,dx = e^x\,dx$$

using the external definition $dg = H(x)\,dx$:

$$H(x) = e^x$$

explanation of implementation:

In function phi(x) we calculated the analytical expressions of the gradient and hessian of the given phi(x).

Using function f1 we calculated the analytical expressions of the gradient and hessian of f1(x).

In function h we return exp(x) as the first and second derivative.

Using function f2 we calculated the analytical expressions of the gradient and hessian of f2(x) using h.

Task 4:

explanation of implementation:

In function numdiff:

> we numerically calculated the gradient by the estimation of the gradient (using the diff of the analytical function's values in x+ and x-).
>
> we numerically calculated the hessian by the estimation of the hessian (using the diff of the analytical function of the gradient in x+ and x-).

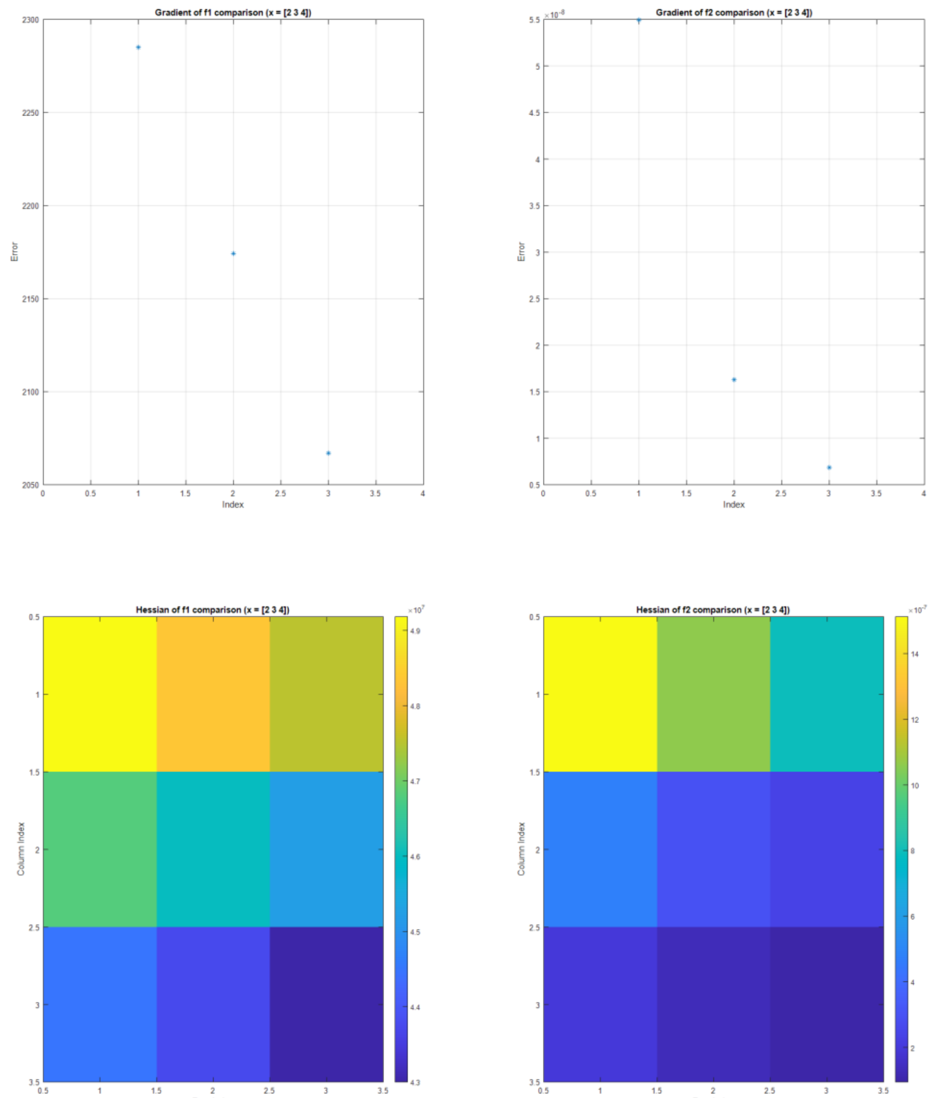
Using numdiff calculated numerically the gradient and hessian for random vector x.

Using f1 and f2 we calculated analytically the gradient and hessian for random vector x.

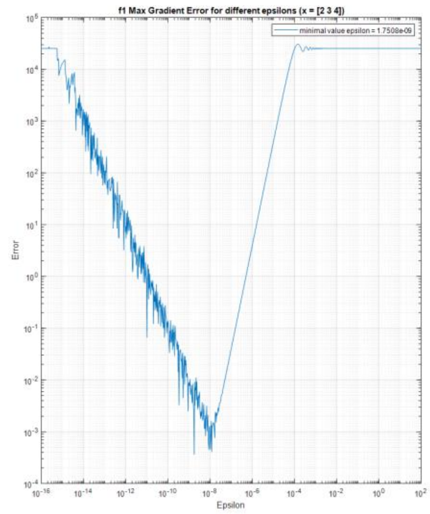
The error is accepted by the absolute difference of the above 2.

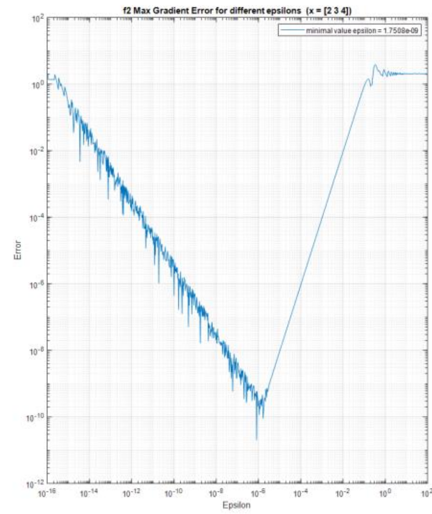
After doing the above for the given epsilon we used a lot different epsilon to find the max error and the fit epsilon.

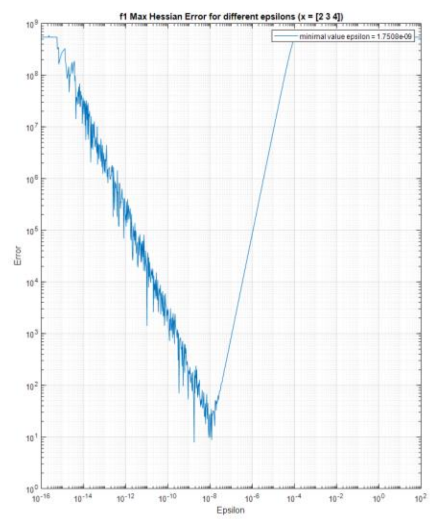
Task 5:

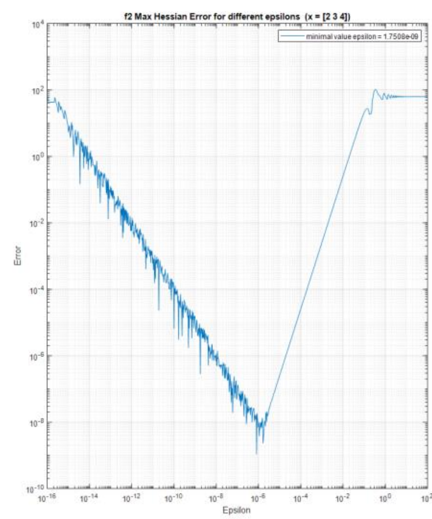


The color represents the error.


f1 Max Gradient Error for different epsilons (x = [2 3 4])
minimal value epsilon = 1.7508e-09
Error
Epsilon


f2 Max Gradient Error for different epsilons (x = [2 3 4])
minimal value epsilon = 1.7508e-09
Error
Epsilon


f1 Max Hessian Error for different epsilons (x = [2 3 4])
minimal value epsilon = 1.7508e-09
Error
Epsilon


f2 Max Hessian Error for different epsilons (x = [2 3 4])
minimal value epsilon = 1.7508e-09
Error
Epsilon