Mini Project 2 Report

Zhengyu Huang

December 13, 2023

1 Gradient

$$
\begin{aligned}
\frac{\partial fw}{\partial w_1} &= \phi(w_2x_1 + w_3x_2 + w_4x_3 + w_5) \\
\frac{\partial fw}{\partial w_2} &= w_1x_1\phi'(w_2x_1 + w_3x_2 + w_4x_3 + w_5) \\
\frac{\partial fw}{\partial w_3} &= w_1x_2\phi'(w_2x_1 + w_3x_2 + w_4x_3 + w_5) \\
\frac{\partial fw}{\partial w_4} &= w_1x_3\phi'(w_2x_1 + w_3x_2 + w_4x_3 + w_5) \\
\frac{\partial fw}{\partial w_5} &= w_1\phi'(w_2x_1 + w_3x_2 + w_4x_3 + w_5) \\
\frac{\partial fw}{\partial w_6} &= \phi(w_7x_1 + w_8x_2 + w_9x_3 + w_{10}) \\
\frac{\partial fw}{\partial w_7} &= w_6x_1\phi'(w_7x_1 + w_8x_2 + w_9x_3 + w_{10}) \\
\frac{\partial fw}{\partial w_8} &= w_6x_2\phi'(w_7x_1 + w_8x_2 + w_9x_3 + w_{10}) \\
\frac{\partial fw}{\partial w_9} &= w_6x_3\phi'(w_7x_1 + w_8x_2 + w_9x_3 + w_{10}) \\
\frac{\partial fw}{\partial w_{10}} &= w_6\phi'(w_7x_1 + w_8x_2 + w_9x_3 + w_{10}) \\
\frac{\partial fw}{\partial w_{11}} &= \phi(w_{12}x_1 + w_{13}x_2 + w_{14}x_3 + w_{15}) \\
\frac{\partial fw}{\partial w_{12}} &= w_{11}x_1\phi'(w_{12}x_1 + w_{13}x_2 + w_{14}x_3 + w_{15}) \\
\frac{\partial fw}{\partial w_{13}} &= w_{11}x_2\phi'(w_{12}x_1 + w_{13}x_2 + w_{14}x_3 + w_{15}) \\
\frac{\partial fw}{\partial w_{14}} &= w_{11}x_3\phi'(w_{12}x_1 + w_{13}x_2 + w_{14}x_3 + w_{15}) \\
\frac{\partial fw}{\partial w_{15}} &= w_{11}\phi'(w_{12}x_1 + w_{13}x_2 + w_{14}x_3 + w_{15}) \\
\frac{\partial fw}{\partial w_{16}} &= 1
\end{aligned}
$$

2 Derivative matrix Dr(w)

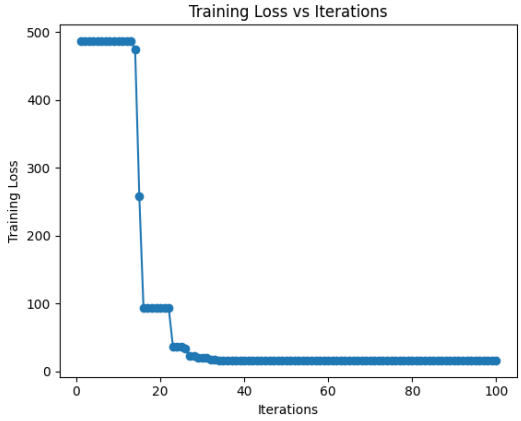
$$D_r(w) = \begin{bmatrix} \frac{\partial r_1}{\partial w_1} & \frac{\partial r_1}{\partial w_2} & \cdots & \frac{\partial r_1}{\partial w_{16}} \\ \frac{\partial r_2}{\partial w_1} & \frac{\partial r_2}{\partial w_2} & \cdots & \frac{\partial r_2}{\partial w_{16}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_N}{\partial w_1} & \frac{\partial r_N}{\partial w_2} & \cdots & \frac{\partial r_N}{\partial w_{16}} \end{bmatrix}$$

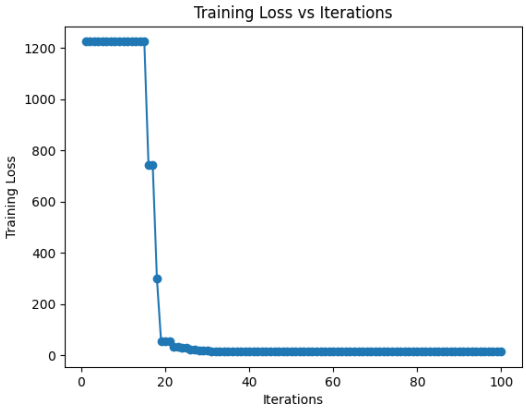
3 Neural Training

3.1

Choosing small residual as stopping criteria: if $||f(x^{(k+1)})||^2 < 10^{-6}$.

A few samples training loss v.s. iterations graph





Results of different initializations for the Levenberg-Marquardt algorithm.

```
Final training Loss for lambda 5e-06  = 0.008191700474334238:
Final training Loss for lambda 5e-05  = 13.289935030733163:
Final training Loss for lambda 0.0005  = 0.013719326432698208:
Final training Loss for lambda 0.005  = 0.004044181509930304:
Final training Loss for lambda 0.05  = 0.0035638234050368203:
Final training Loss for lambda 0.5  = 0.010553597688733514:
Final training Loss for lambda 5  = 0.0034309282904195 96:
```

3.2

training and test errors for different initializations, different choices of λ and Γ_T

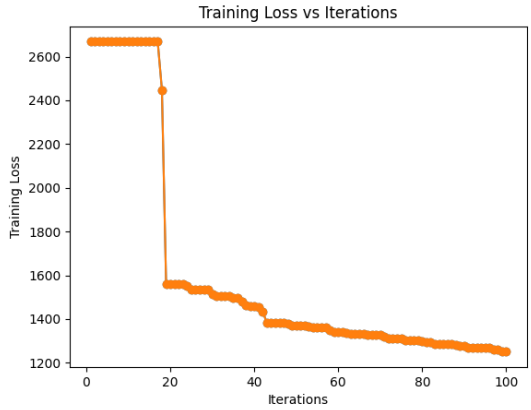
```
-----------------------------------------
    Test Error of lambda 5e-06, gamma 0.5 = 0.00000168
Training Error of lambda 5e-06, gamma 0.5 = 0.00000137
    Test Error of lambda 5e-06, gamma 1 = 0.00001719
Training Error of lambda 5e-06, gamma 1 = 0.00001564
    Test Error of lambda 5e-06, gamma 1.5 = 0.00006842
Training Error of lambda 5e-06, gamma 1.5 = 0.00005754
-----------------------------------------
    Test Error of lambda 5e-05, gamma 0.5 = 0.00149677
Training Error of lambda 5e-05, gamma 0.5 = 0.00196263
    Test Error of lambda 5e-05, gamma 1 = 0.02836497
Training Error of lambda 5e-05, gamma 1 = 0.02657987
    Test Error of lambda 5e-05, gamma 1.5 = 0.00007447
Training Error of lambda 5e-05, gamma 1.5 = 0.00006252
-----------------------------------------
    Test Error of lambda 0.0005, gamma 0.5 = 0.00000291
Training Error of lambda 0.0005, gamma 0.5 = 0.00000242
    Test Error of lambda 0.0005, gamma 1 = 0.00002818
Training Error of lambda 0.0005, gamma 1 = 0.00002461
    Test Error of lambda 0.0005, gamma 1.5 = 0.00007316
Training Error of lambda 0.0005, gamma 1.5 = 0.00006084
-----------------------------------------
    Test Error of lambda 0.005, gamma 0.5 = 0.00147550
Training Error of lambda 0.005, gamma 0.5 = 0.00194835
    Test Error of lambda 0.005, gamma 1 = 0.00000849
Training Error of lambda 0.005, gamma 1 = 0.00000775
    Test Error of lambda 0.005, gamma 1.5 = 0.00005795
Training Error of lambda 0.005, gamma 1.5 = 0.00004860
-----------------------------------------
    Test Error of lambda 0.05, gamma 0.5 = 0.00000126
Training Error of lambda 0.05, gamma 0.5 = 0.00000105
    Test Error of lambda 0.05, gamma 1 = 0.00000841
Training Error of lambda 0.05, gamma 1 = 0.00000713
    Test Error of lambda 0.05, gamma 1.5 = 0.00006300
Training Error of lambda 0.05, gamma 1.5 = 0.00005278
-----------------------------------------
    Test Error of lambda 0.5, gamma 0.5 = 0.00000140
Training Error of lambda 0.5, gamma 0.5 = 0.00000117
    Test Error of lambda 0.5, gamma 1 = 0.00002216
Training Error of lambda 0.5, gamma 1 = 0.00002001
    Test Error of lambda 0.5, gamma 1.5 = 0.00005644
Training Error of lambda 0.5, gamma 1.5 = 0.00004767
-----------------------------------------
    Test Error of lambda 5, gamma 0.5 = 0.00000112
Training Error of lambda 5, gamma 0.5 = 0.00000094
    Test Error of lambda 5, gamma 1 = 0.00000714
Training Error of lambda 5, gamma 1 = 0.00000666
    Test Error of lambda 5, gamma 1.5 = 0.00005326
Training Error of lambda 5, gamma 1.5 = 0.00004463
```

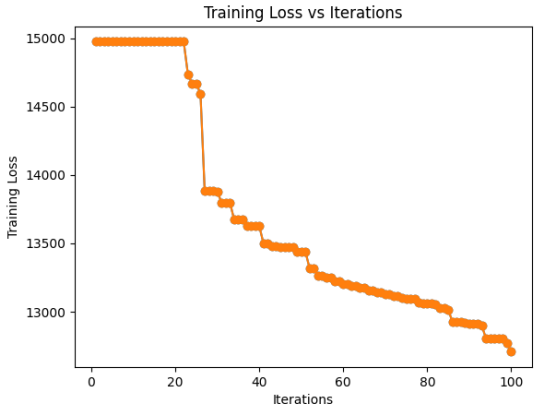
For error calculation, I used mean square error. Smaller values of lambda tend to result in smaller errors. Larger values of gamma seem to lead to more errors.

3.3

$$g(x^{(n)}) = \frac{x_0^{(n)}}{x_1^{(n)}} * x_2^{(n)}$$

A few samples training loss v.s. iterations graph





Results of different initializations for the Levenberg-Marquardt algorithm.

```
-----------------------------------------
Final training Loss for lambda 5e-06  = 1250.2197584616506:
Final training Loss for lambda 5e-06  = 5995.5861033418805:
Final training Loss for lambda 5e-06  = 12707.430258403348:
-----------------------------------------
Final training Loss for lambda 5e-05  = 1079.5211722898168:
Final training Loss for lambda 5e-05  = 5727.351151720053:
Final training Loss for lambda 5e-05  = 12995.704667306127:
-----------------------------------------
Final training Loss for lambda 0.0005  = 1092.2004527532272:
Final training Loss for lambda 0.0005  = 5303.096945929476:
Final training Loss for lambda 0.0005  = 12531.290803465994:
-----------------------------------------
Final training Loss for lambda 0.005  = 1092.5942723243193:
Final training Loss for lambda 0.005  = 5995.278037131428:
Final training Loss for lambda 0.005  = 13004.003660700475:
-----------------------------------------
Final training Loss for lambda 0.05  = 1076.64531512595:
Final training Loss for lambda 0.05  = 4722.7905121771055:
Final training Loss for lambda 0.05  = 11596.728520171113:
-----------------------------------------
Final training Loss for lambda 0.5  = 1082.8458377119637:
Final training Loss for lambda 0.5  = 4985.13783487937:
Final training Loss for lambda 0.5  = 11368.957918432494:
-----------------------------------------
Final training Loss for lambda 5  = 1451.6772669626475:
Final training Loss for lambda 5  = 5806.447271988076:
Final training Loss for lambda 5  = 12604.656762620036:
```

I reused most of the previous parts, except for changing the definition of mapping.
training and test errors for different initializations, different choices of λ and Γ_T

```
-----------------------------------------
    Test Error of lambda 5e-06, gamma 0.5 = 4.44233364
Training Error of lambda 5e-06, gamma 0.5 = 2.44222769
    Test Error of lambda 5e-06, gamma 1 = 21.72089646
Training Error of lambda 5e-06, gamma 1 = 11.99117221
    Test Error of lambda 5e-06, gamma 1.5 = 47.60623967
Training Error of lambda 5e-06, gamma 1.5 = 25.41486052
-----------------------------------------
    Test Error of lambda 5e-05, gamma 0.5 = 4.23389622
Training Error of lambda 5e-05, gamma 0.5 = 2.15904234
    Test Error of lambda 5e-05, gamma 1 = 21.20202646
Training Error of lambda 5e-05, gamma 1 = 11.41216053
    Test Error of lambda 5e-05, gamma 1.5 = 49.94969517
Training Error of lambda 5e-05, gamma 1.5 = 25.98579207
-----------------------------------------
    Test Error of lambda 0.0005, gamma 0.5 = 4.23342098
Training Error of lambda 0.0005, gamma 0.5 = 2.17367634
    Test Error of lambda 0.0005, gamma 1 = 17.52266163
Training Error of lambda 0.0005, gamma 1 = 10.60619389
    Test Error of lambda 0.0005, gamma 1.5 = 50.62056770
Training Error of lambda 0.0005, gamma 1.5 = 25.01181607
-----------------------------------------
    Test Error of lambda 0.005, gamma 0.5 = 4.29947497
Training Error of lambda 0.005, gamma 0.5 = 2.17676510
    Test Error of lambda 0.005, gamma 1 = 21.79539091
Training Error of lambda 0.005, gamma 1 = 11.99010294
    Test Error of lambda 0.005, gamma 1.5 = 47.43047630
Training Error of lambda 0.005, gamma 1.5 = 25.99375643
-----------------------------------------
    Test Error of lambda 0.05, gamma 0.5 = 4.22188709
Training Error of lambda 0.05, gamma 0.5 = 2.15329063
    Test Error of lambda 0.05, gamma 1 = 26.08957965
Training Error of lambda 0.05, gamma 1 = 9.37495175
    Test Error of lambda 0.05, gamma 1.5 = 41.54509978
Training Error of lambda 0.05, gamma 1.5 = 23.15264820
-----------------------------------------
    Test Error of lambda 0.5, gamma 0.5 = 4.18155249
Training Error of lambda 0.5, gamma 0.5 = 2.15664017
    Test Error of lambda 0.5, gamma 1 = 32.55203074
Training Error of lambda 0.5, gamma 1 = 9.96549210
    Test Error of lambda 0.5, gamma 1.5 = 42.33030830
Training Error of lambda 0.5, gamma 1.5 = 22.73791584
-----------------------------------------
    Test Error of lambda 5, gamma 0.5 = 5.22875469
Training Error of lambda 5, gamma 0.5 = 2.90335453
    Test Error of lambda 5, gamma 1 = 23.72685940
Training Error of lambda 5, gamma 1 = 11.11961751
    Test Error of lambda 5, gamma 1.5 = 46.55965675
Training Error of lambda 5, gamma 1.5 = 25.20931353
```

3.4

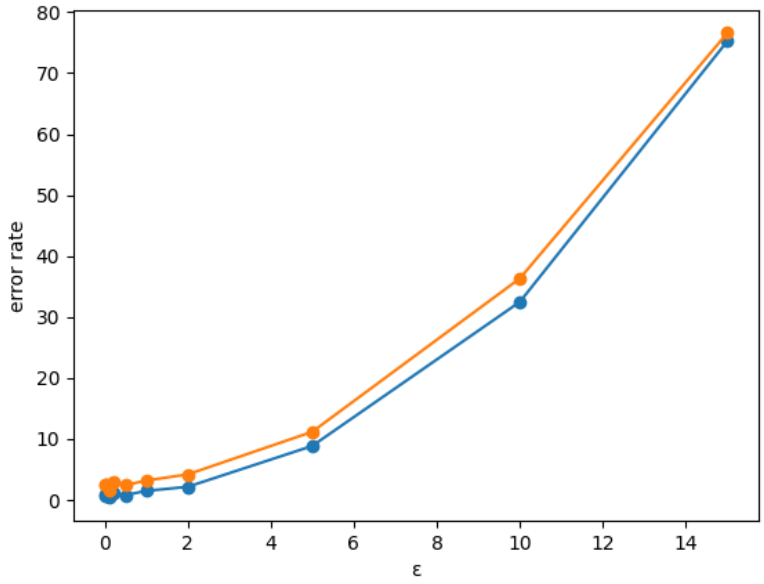


Figure 1: Error rate v.s. noise level

Using gamma = 0.5 for this part. Using 0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5 as values of ϵ. Still using mean square as error metrics. Obviously, the larger ϵ is, the higher error rate will be. However, when epsilon is small, the affect on error rate is negligible

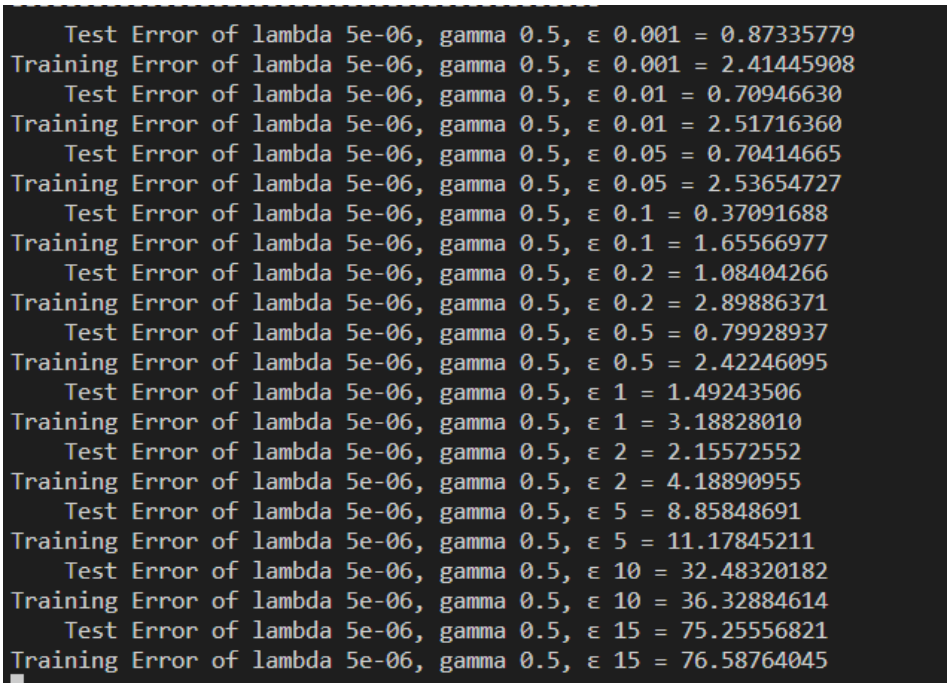


Figure 2: Error rate with noise