CMPS 142 HW1

Jordan Liss

January 31, 2017

1 Question 1: Learning Real Predictors-linear hypotheses

Generally using the different predictors produced relatively similar weight parameters, but weight parameter determined by the L_2 norm and L_∞ norm were noticably better then the L_1 weight parameter model.

(a) Look to Matlab code for function minL2(X,y)

```
function W = minL2(X, y)
        W = X\y;
end
```

(b) Before creating the sub elements that go into linprog we had to linearize the the non-linear L1 norm equation predictor.

$$\begin{aligned} -\delta \leq ||Xw - y|| &\geq \delta \\ \min \delta & \\ \text{subject to } Xw - \delta &\leq y \\ -Xw - \delta &\leq -y \end{aligned}$$

Now we need to reorganize the two equations into matricies that can be linearized by p=linprog (A,b,f)

$$A = \begin{bmatrix} [X][-I] \\ [-X][-I] \end{bmatrix}$$

$$p = \begin{bmatrix} W_1 \\ W_2 \\ \delta_1 \\ . \\ . \\ . \\ \delta_i \end{bmatrix}$$

$$f = \begin{bmatrix} 0 \\ 0 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}$$

$$b = \begin{bmatrix} [y] \\ [-y] \end{bmatrix}$$

The objective is to determine what the first two elements are in the p vector, which are the two elements W_1 and W_2

Look to Matlab code for actual implementation of the function minL1(X,y)

```
% minL1(X, y)
function W = minL1(X, y)
        [xr,xc] = size(X);
        [yr,yc] = size(y);
        Ipos = eye(xr);
        Ineg = Ipos*(-1);
        A=[[X,Ineg];-[X,Ipos]];
        f=[zeros(xc,1);eye(xr,1)]; % Objective function
        b=[y;-y];
        P = linprog(f,A,b);
        W= P(1:xc,:);
end
```

(c) Look to Matlab code for function minLoo(X,y)

```
% minLoo(X, y)
        function W = minLoo(X, y)
                [xr,xc] = size(X);
                [yr,yc] = size(y);
                Ipos = ones(xr);
                Ineg = Ipos*(-1);
                A=[[X,Ineg];-[X,Ipos]];
                f=[zeros(xc,1);eye(xr,1)]; % Objective function
                b=[y;-y];
                P = linprog(f,A,b);
                W= P(1:xc,:);
        end
```

(d) Perform tasks A to D: (A) Generate synthetic data and train by running the $w1 = minL2(X, y)$, $w2 = minL1(X, y)$ and $woo = minLoo(X, y)$ predictors. (B) Produce 2D plots. (C) Report training data error and testing data error. (D) Report average training data error and average testing data error.

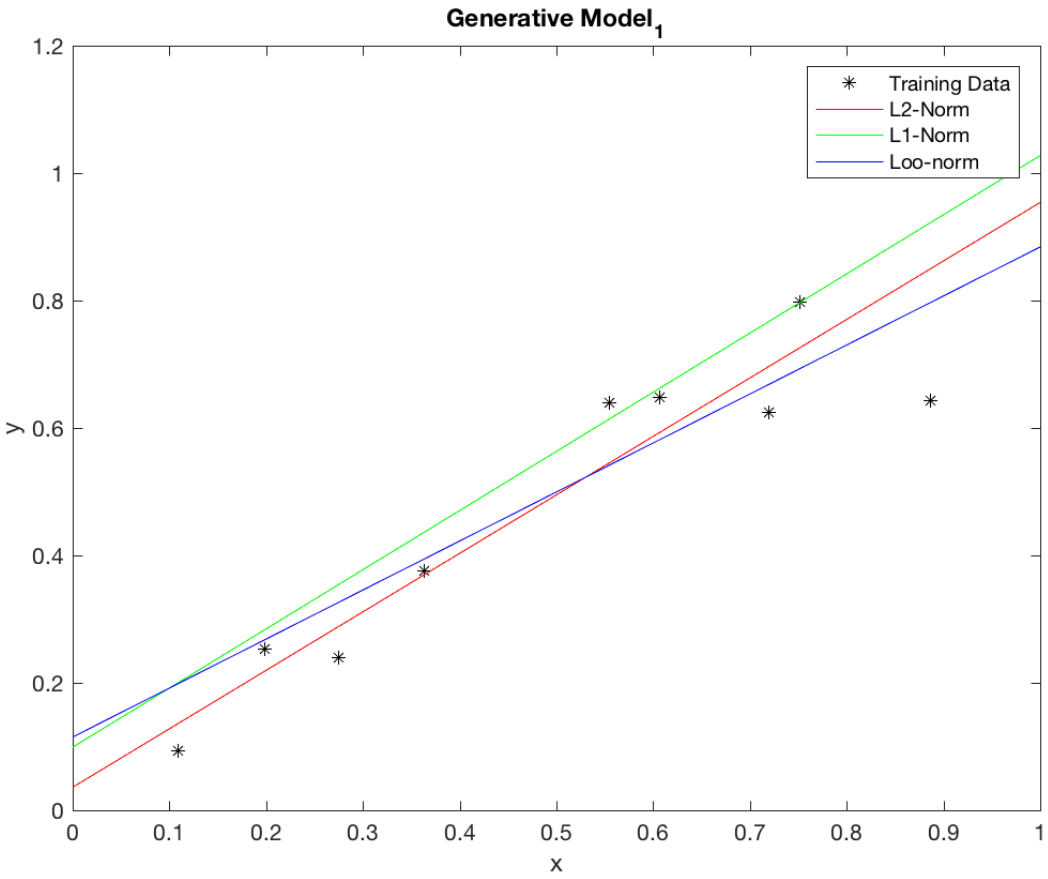

Figure 1: Part 1 Predictor for Generative Model 1

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	0.697329369904920	0.274978207355952	0.185020622989566
w_2	0.567239787716853	0.212806226723125	0.120065206179200
w_∞	.00685043698377	0.396761122799188	0.228629807277931

Table 1: Model 1 Error of Predictor on Training Data

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	99.3464100241549	3.93578244492570	0.511712000605006
w_2	99.578338150394	3.89284688146704	0.475087822357517
w_∞	119.238731289244	4.61086085024203	0.418315719030117

Table 2: Model 1 Error of Predictor on Test Data

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	62.9271680997838	22.0770638439754	11.9149543598173
w_2	0.932866673663514	0.377992915655674	0.234033604746163
w_∞	1.02185600242093	0.393166620019908	0.239092803186037

Table 3: Model 1 Average Error of Predictor on Training Data

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	110.405939333776	4.27738255092864	0.429043392791622
w_2	5686.95710339549	217.497772395070	12.9084416812671
w_∞	103.657369326111	4.05155026146335	0.411072115339784

Table 4: Model 1 Average Error of Predictor on Test Data

Generative Model 2 Part 1

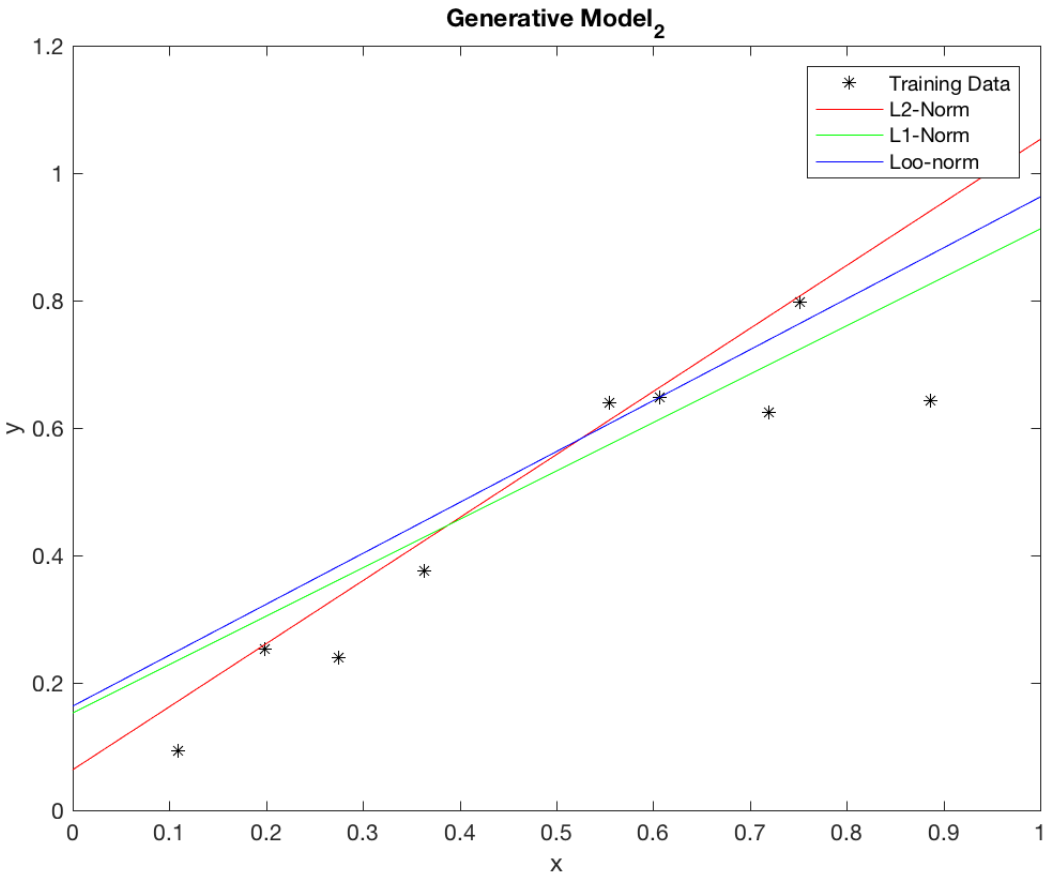

Figure 2: Part 1 Predictor for Generative Model 2

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	1.64466747111963	0.650574252405543	0.476222183760688
w_2	1.67519393876170	0.622154095624610	0.381841830950206
w_∞	1.66256041601797	0.675300334420521	0.491710591758382

Table 5: Model 2 Error of Predictor on Training Data

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	553.321426020904	207.653740868836	202.544096655929
w_2	552.950021777608	207.591231615492	202.459404591402
w_∞	549.536787180711	207.667170533447	202.544947474589

Table 6: Model 2 Error of Predictor on Test Data

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	12.1154828240420	10.7236694765059	10.5240352028956
w_2	19.2303489366538	9.88554287048076	8.97909462222926
w_∞	12.1090216264764	10.7261862430966	10.5240666970461

Table 7: Model 2 Average Error of Predictor on Training Data

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	1132.95371474223	693.594625463970	666.739555529568
w_2	2106.68472094465	722.163455018603	667.662108420260
w_∞	1132.39617023331	693.593765063588	666.740892576882

Table 8: Model 2 Average Error of Predictor on Test Data

Generative Model 3 Part 1

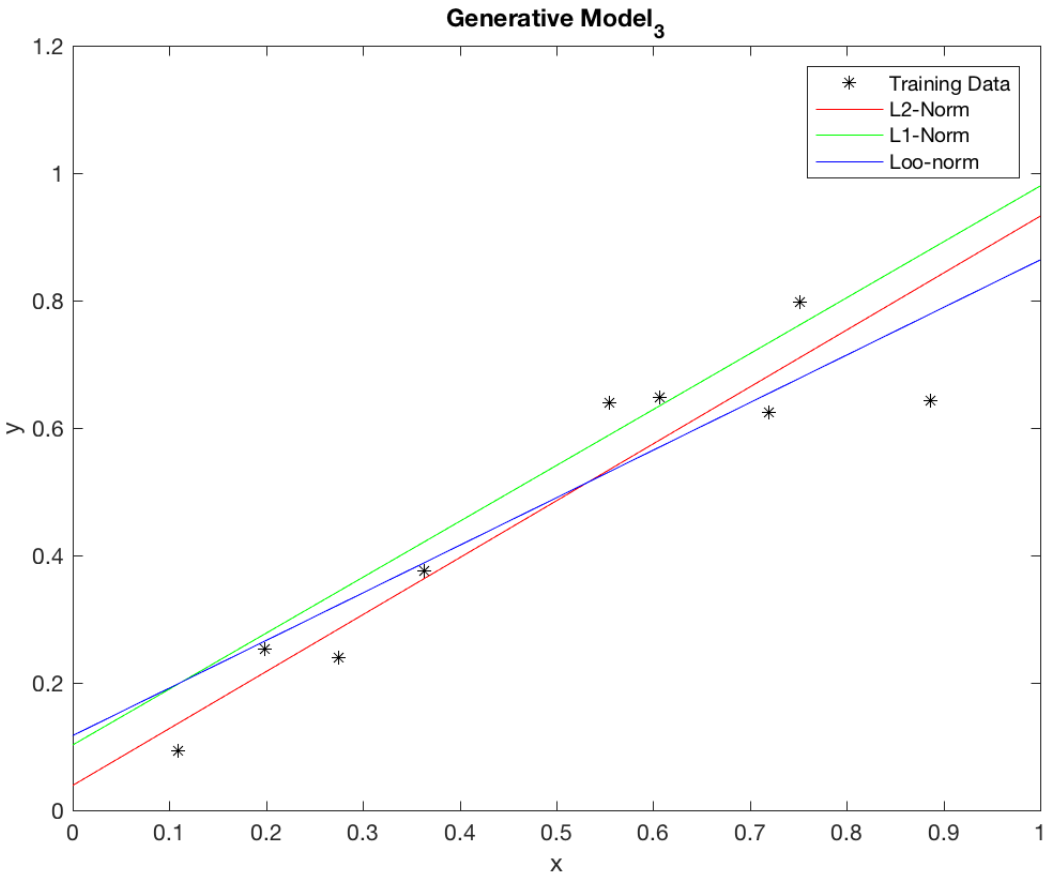

Figure 3: Part 1 Predictor for Generative Model 3

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	1.22648568439107	0.466349175990427	0.276479379061773
w_2	2.69325553760571	0.982597953889607	0.554178348491704
w_∞	1.12535235626370	0.431915947426161	0.279723122218481

Table 9: Part 1 Model 3 Error of Predictor on Training Data

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	98.6839361639095	3.99505413139004	0.494683776685453
w_2	209.772625673681	8.10244069646544	0.833662804487294
w_∞	113.048047411512	4.53769354768792	0.571201125179311

Table 10: Part 1 Model 3 Error of Predictor on Test Data

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	1.52602611411917	0.565426193007616	0.328263515353915
w_2	81.2905246837570	28.8215746349034	15.3583225095990
w_∞	1.46085766024037	0.560233255673686	0.313774078399470

Table 11: Part 1 Model 3 Average Error of Predictor on Training Data

$Types$	$L_1 error$	$L_2 error$	$L_\infty error$
w_1	151.466064803865	5.74125673568479	0.734572574003104
w_2	7555.56959191315	285.847036013301	16.7548822997011
w_∞	172.231999150482	6.48300692798046	0.763176775879400

Table 12: Part 1 Model 3 Average Error of Predictor on Test Data

2 Question 2: Learning Real Predictors-Polynomial Basis

(a) Look to Matlab code of function c=minL2poly(x,y,d). With vector c being weights for the polynomial p(x).

```
function c = minL2poly(x, y, d)
[t,xc]=size(x);
X=ones(t,d+1);
for i_d=1:d
x_temp=x.^(d-i_d+1);
X(:,i_d)=x_temp;
end
c=X\y;
end
```

[(A)] Various different weights for different degrees of polynomials (1,3,5 and 9).

[(B)] 2D plots of the training data.

Generative Model 1 Part 2

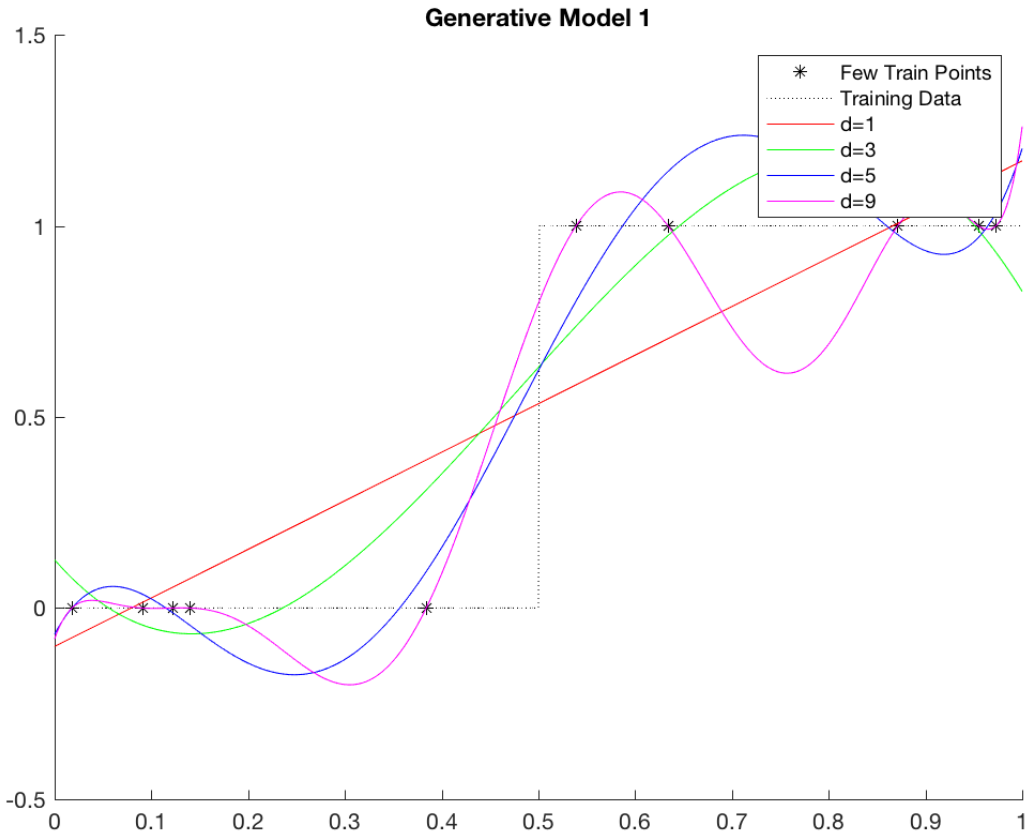

Figure 4: Part 2 Generative Model 1

Looking at the error of the 4 predictors when compared to training data and the test data. It's clear that the when the degress of polynomials is too large (For example c_9) the training data and the test data errors are vastly different. Which means the predictor weight parameters were overfitted. The optimal point is when the training error and the test data error are both minimized.The weight parameters that simultaneously minimized the training and test data error was when the polynomial degree was between 3-5.

$Types$	$L_2 error$
c_1	0.689630613208235
c_3	0.561041746600777
c_5	0.459583917740955
c_9	2.54971409490484e-11

Table 13: Model 1 Error of Predictor on Training Data

$Types$	$L_2 error$
c_1	8.17038050638767
c_3	6.31700452034233
c_5	7.75745020030541
c_9	37.0223908385901

Table 14: Model 1 Error of Predictor on Test Data

$Types$	$L_2 error$
c_1	0.690155099996523
c_3	0.383480787135169
c_5	0.210171453977691
c_9	6.29900514010916e-10

Table 15: Model 1 Average Error Table of Predictor on Training Data

$Types$	$L_2 error$
c_1	8.66627085095551
c_3	13.5910673276037
c_5	147.086690903302
c_9	54957.8473909081

Table 16: Model 1 Average Error Table of Predictor on Test Data

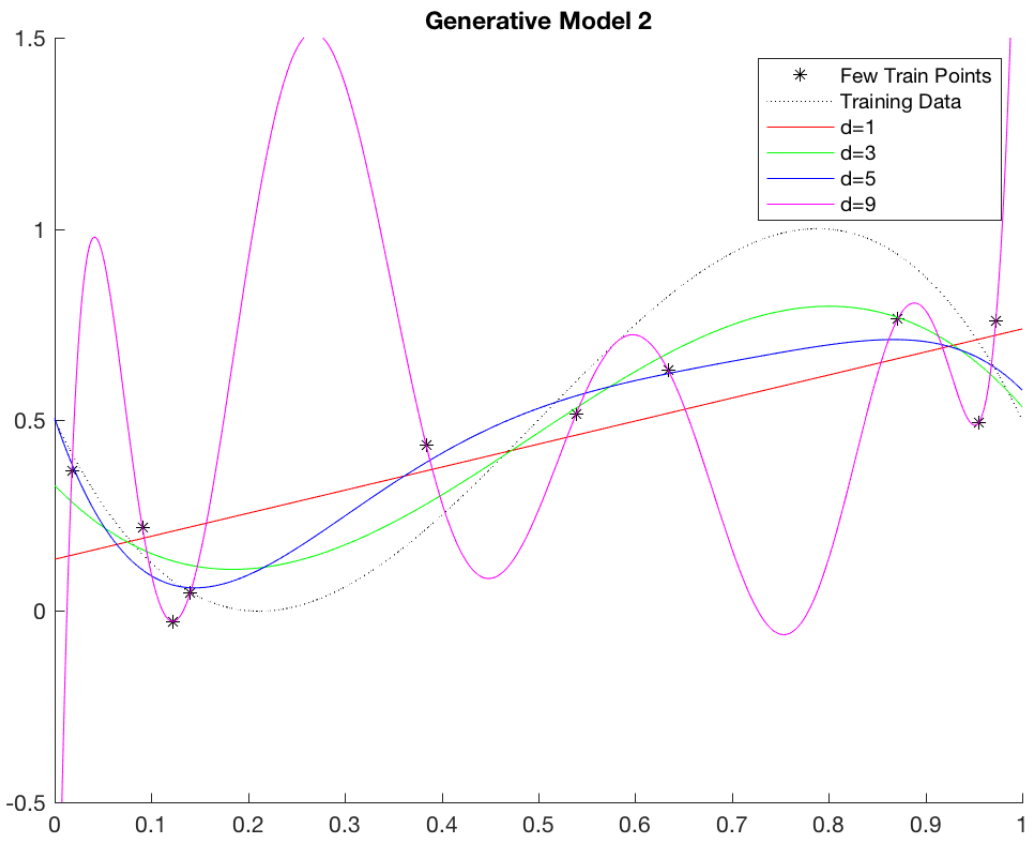

Figure 5: Part 2 Generative Model 2

$Types$	$L_2 error$
c_1	0.457835298065049
c_3	0.151946454605306
c_5	0.133706111039809
c_9	2.58721838587574e-11

Table 17: Model 2 Error of Predictor on Training Data

Looking at the error of the 4 predictors when compared to training data and the test data. It's clear that the when the degress of polynomials is too large (For example c_9) the training data and the test data errors are vastly different. Which means the predictor weight parameters were overfitted. The optimal point is when the training error and the test data error are both minimized.The weight parameters that simultaneously minimized the training and test data error was when the polynomial degree was between 3-5.

$Types$	$L_2 error$
c_1	7.24926903903986
c_3	3.47583084183123
c_5	3.83128070548510
c_9	43.3357596805554

Table 18: Model 2 Error of Predictor on Test Data

$Types$	$L_2 error$
c_1	0.586972543825614
c_3	0.238597453871402
c_5	0.197656491427826
c_9	1.10874462653068e-06

Table 19: Model 2 Average Error Table of Predictor on Training Data

$Types$	$L_2 error$
c_1	7.83695864456785
c_3	5.58222346180003
c_5	46.1977206017121
c_9	32357448.2045903

Table 20: Model 2 Average Error Table of Predictor on Test Data

3 Question 3: Hessian Matrix

(b) Proving the Hessian is a symmetric positive semi-definite starts first with looking at what the Hessian of the $f(w) = ||Xw - y||^2$. Where the Hessian is a t x n matrix.

$$H(f(w)) = \begin{bmatrix} \frac{\delta^2 f}{\delta w_1^2} \frac{\delta^2 f}{\delta w_1 \delta w_2} \cdots\cdots \frac{\delta^2 f}{\delta w_1 \delta w_n} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \frac{\delta^2 f}{\delta w_t \delta w_1} \cdots\cdots\cdots\cdots \frac{\delta^2 f}{\delta w_t \delta w_n} \end{bmatrix}$$

Now let's look at whether $f'(w) = 0$ and $f''(w) \geq 0$, because when the first derivative of a point (In this case w_i) on a function is equal to me, it means the function is at a minimum. When the first derivative is equal to 0 in combination with the $f''(w) \geq 0$ then the function is absolutely a minimum and not a saddle.

$f(w) = ||Xw - y||^2$

$f'(w) = 2wX^TX - 2y^TX$

$f''(w) = 2X^TX$

Any vector in , this case X will be multiplied by itself will always be positive, which concludes that $f''(w)$ will always be positive.

The Hessian is symmetric because $\frac{\delta^2 f}{\delta w_t \delta w_n} = \frac{\delta^2 f}{\delta w_n \delta w_t}$ or in otherwords the $\nabla^2 f(x)^T = \nabla^2 f(x)$. Because the Hessian is symmetric it means the minimum is not in a saddle but at a minimum(Actually a global minimum, because the function is convex). The values to the left of the minimum point will have a $f''(w) \leq 0$ and to the right will be $f''(w) \geq 0$.

(b) Given the Rosenbrock function compute the $\nabla f(x)$ and the Hessian $\nabla^2 f(x)$ and check whether $x^* = [1; 1]$ is the only local minimizer of the function. First let's prove that the $\nabla f(x) = 0$ and that the Hessian matrix is positive semi-definite and symmetric. Let's prove $\nabla f(x) = 0$

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

$$\nabla f(x) = \begin{bmatrix} 400(x_2 - x_1)x_1 - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{bmatrix} \tag{1}$$

$$\nabla f(x^*) = \begin{bmatrix} 400((1) - 1)(1) - 2(1 - 1) \\ 200(1 - 1^2) \end{bmatrix} \tag{2}$$

$$\nabla f(x^*) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3}$$

After examining the $\nabla f(x^*)$ at the point x^* it is confirmed that $\nabla f(x^*) = 0$ meaning that at this point there might be a minimum, saddle or maximum. To confirm that the point is at a minimum or a saddle I need to prove $\nabla^2 f(x) \geq 0$.

$$\nabla f(x) = \begin{bmatrix} 400(x_2 - x_1)x_1 - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{bmatrix} \tag{4}$$

$$\nabla^2 f(x) = \begin{bmatrix} 900(x_2 - x_1)^{-1}x_1^2 + (x_2 - x_1^2) + 2 & 400(x_2 - x_1^2)^{-1}x_1 \\ 400(x_2 - x_1^2)^{-1}x_1 & 200(x_2 - x_1^2)^{-1} \end{bmatrix} \tag{5}$$

$$\nabla^2 f(x^*) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \tag{6}$$

$\nabla^2 f(x^*) > 0$ and thus at x^* it can no longer be considered a maximum, because the function is postive semi-definite. Its a minimum or a saddle. Upon observation of the $\nabla^2 f(x^*)$ one can see the matrix is symmetric and thus its symmetric positive semi-definite. Its transpose is identical and because we can confirm that $\nabla^2 f(x^*)$ is symmetric we can confirm that at point x^* it is not a saddle or a maximum. Thus its a minimum and not only that a global minimum and the function is a convex function.