CSE 446 A3

Johan J How

0. Policies

0.1: List of Collaborators

Foris Kuang, Problem 1.3

0.2 List of Acknowledgements

0.3 I have read and understood these policies

1. Binary Classification with Linear Regression on MNIST

1.1: Linear Regression, using the Closed Form Estimator

1.1.1

Python spat out an error, saying that there was a singular matrix. This was because taking the inverse of 1/N X^TX does not work, as there exist rows that are not linearly independent. We must add lambda * Identity Matrix to avoid this.

1.1.2

a.

a. I chose a lambda value of 1.

b.

a. Training Average Squared Error: 0.0276063621216%

b. Test Average Squared Error: 0.0267821022407%

c. Dev Average Squared Error: 0.0247244576109%

c.

a. Training Misclassification Error: 4.82121333869%

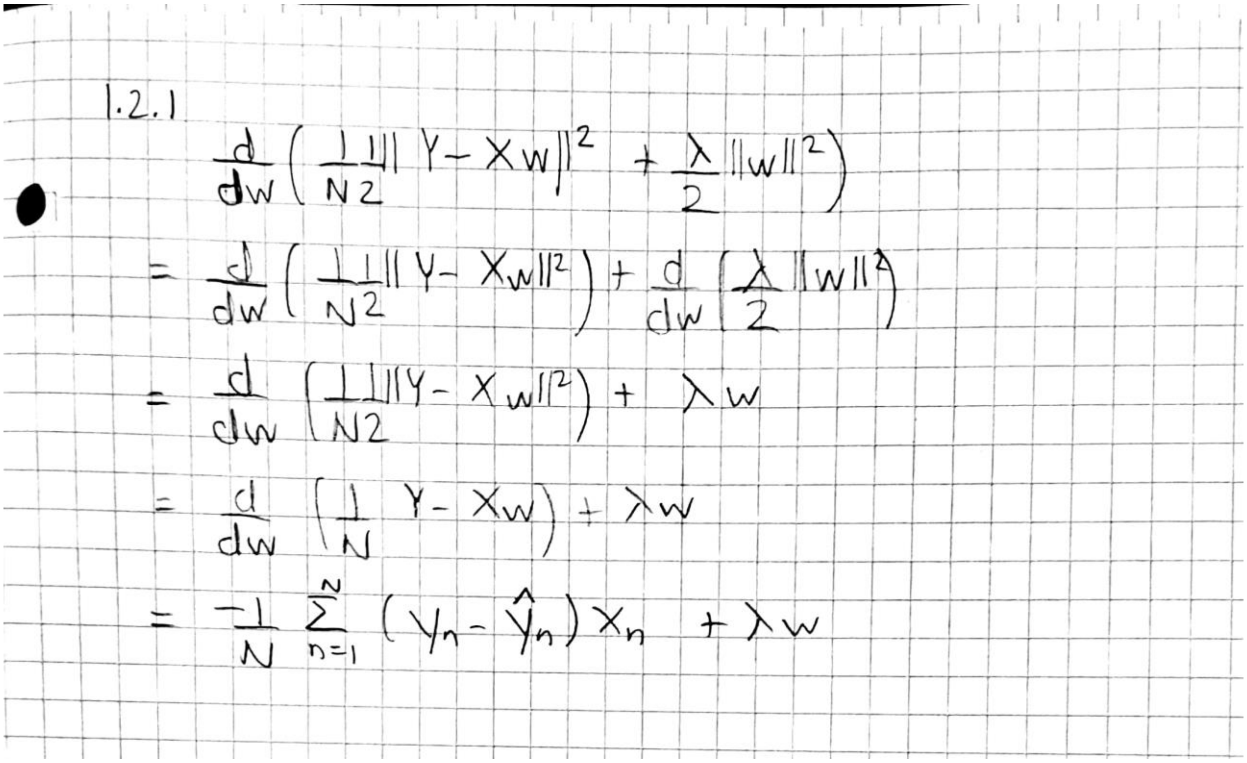
b. Test Misclassification Error: 4.94855463008%

c. Dev Misclassification Error: 4.1004613019%

1.1.3

Because linear regression gives labels, rather than probabilities. We would rather use logistic regression because it gives probabilities. Additionally, it is far faster to use gradient descent.

1.2: Linear regression using gradient descent



1.2.1

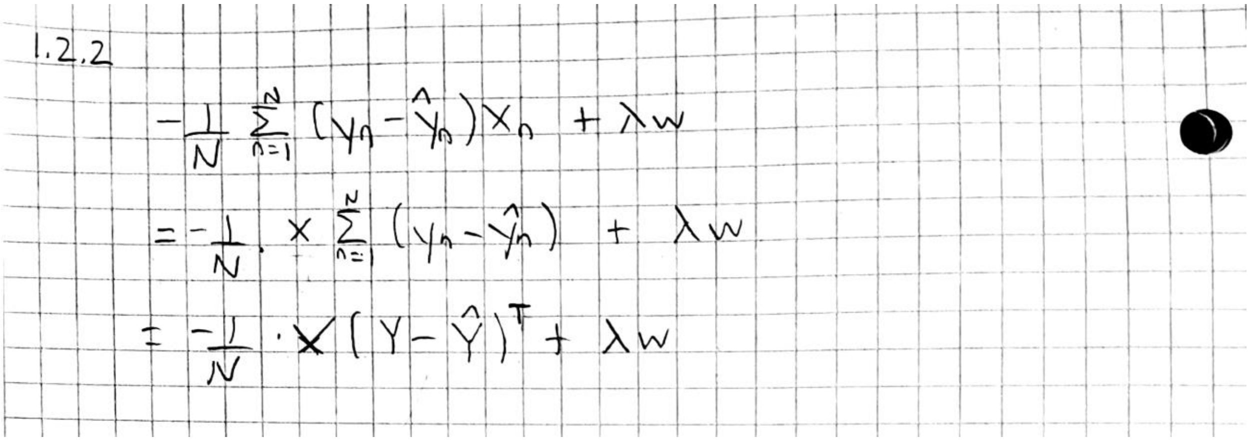
$$\frac{d}{dw}\left(\frac{1}{N}\frac{1}{2}\| Y - Xw\|^2 + \frac{\lambda}{2}\|w\|^2\right)$$

$$= \frac{d}{dw}\left(\frac{1}{N}\frac{1}{2}\| Y - Xw\|^2\right) + \frac{d}{dw}\left(\frac{\lambda}{2}\|w\|^2\right)$$

$$= \frac{d}{dw}\left(\frac{1}{N}\frac{1}{2}\| Y - Xw\|^2\right) + \lambda w$$

$$= \frac{d}{dw}\left(\frac{1}{N} Y - Xw\right) + \lambda w$$

$$= \frac{-1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)x_n + \lambda w$$



1.2.2

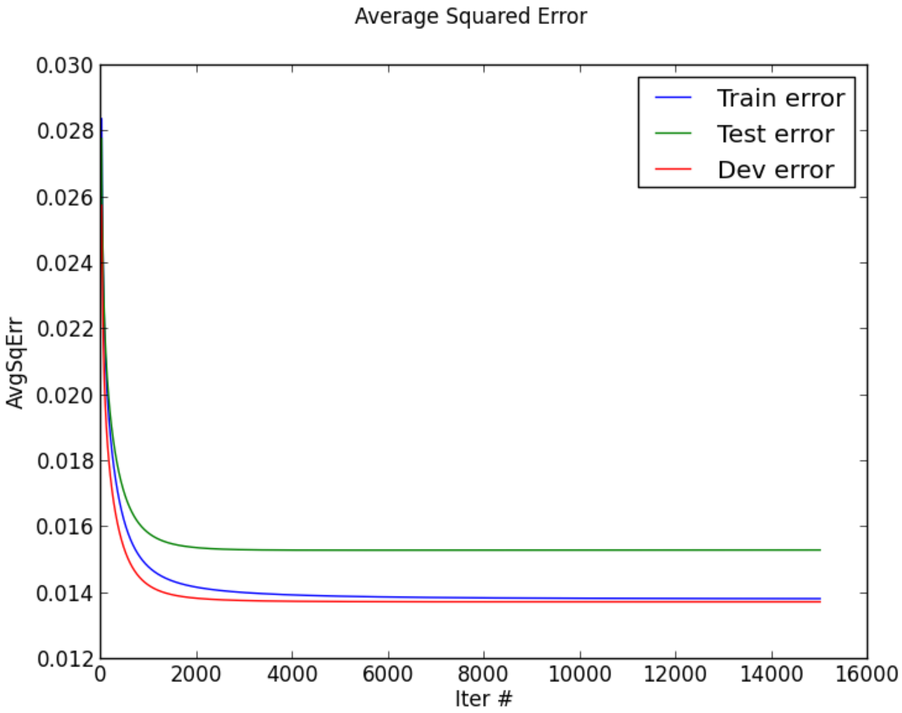
$$-\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)x_n + \lambda w$$

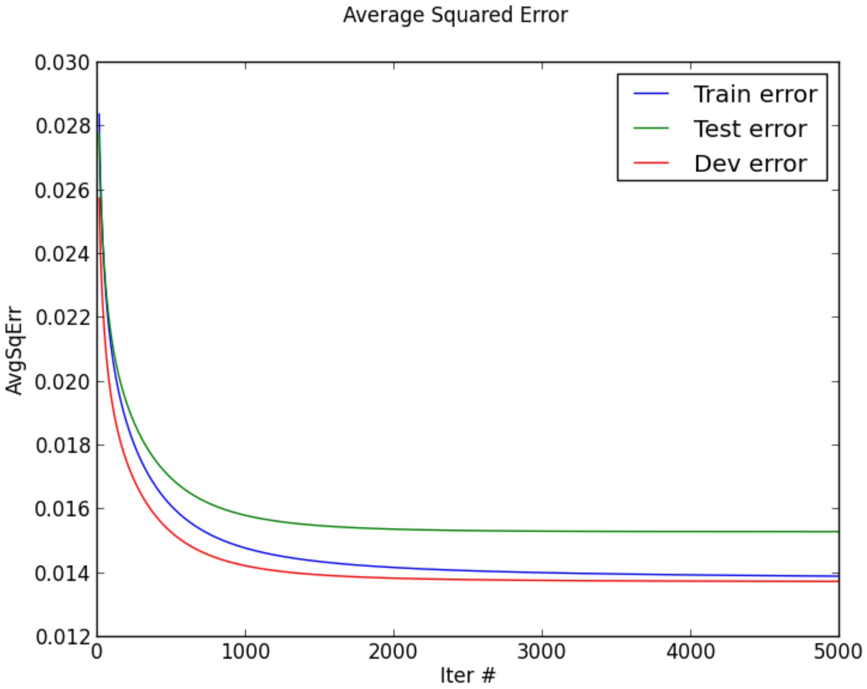
$$= -\frac{1}{N} \cdot X \sum_{n=1}^{N}(y_n - \hat{y}_n) + \lambda w$$

$$= -\frac{1}{N} \cdot X(Y - \hat{Y})^T + \lambda w$$

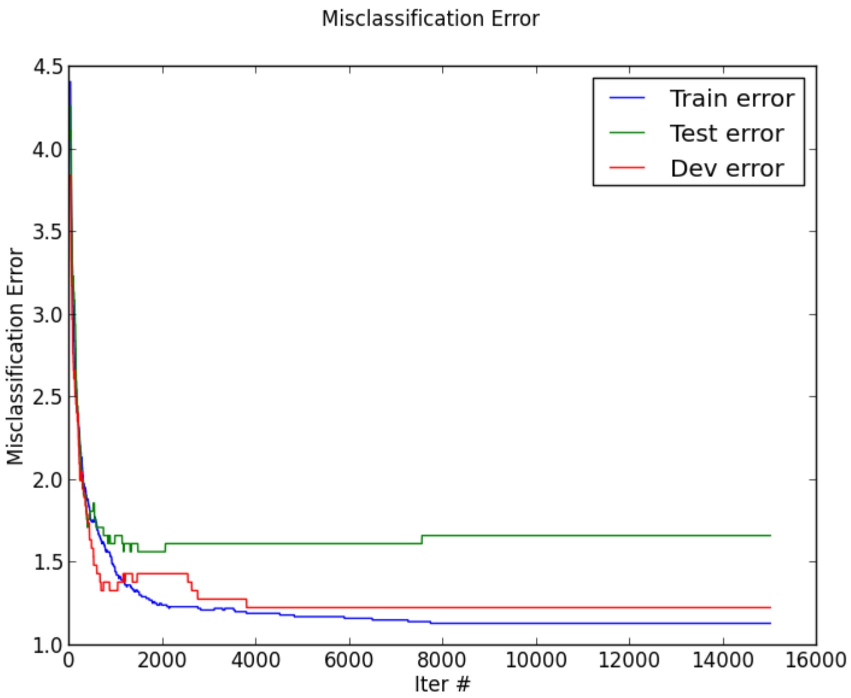
1.2.3

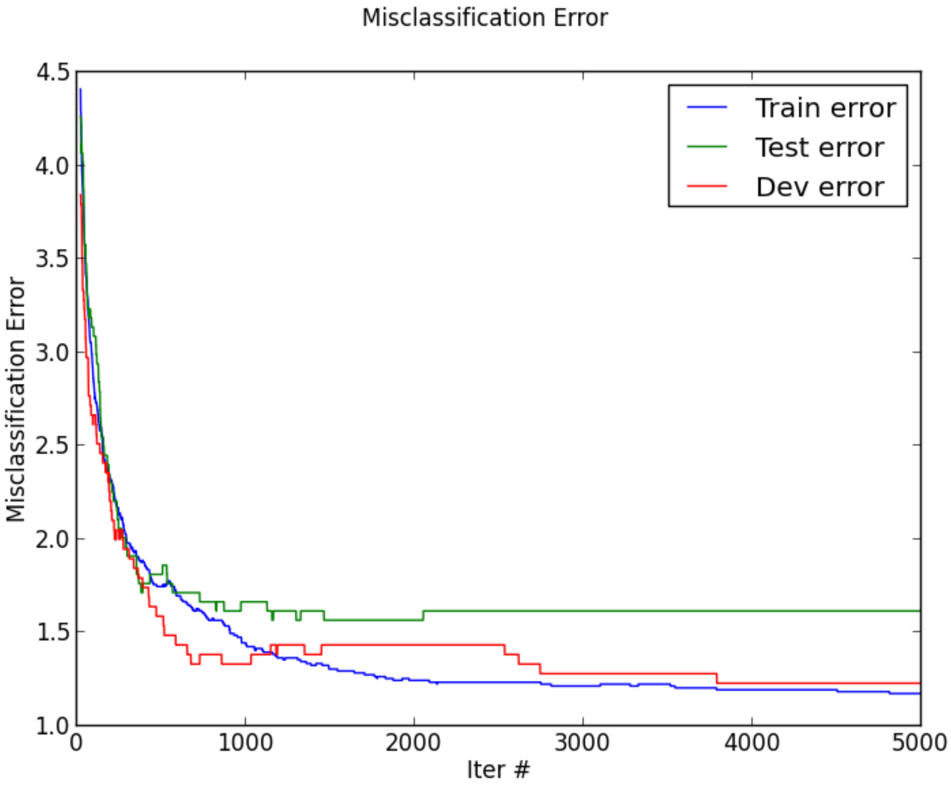
a. I chose a step size of 0.04, as this was one of the largest step sizes I could use.
b. I used a lambda value of 0.1





c. Lowest misclassification error: Train: 1.18%, Test: 1.57%, Dev: 1.23%

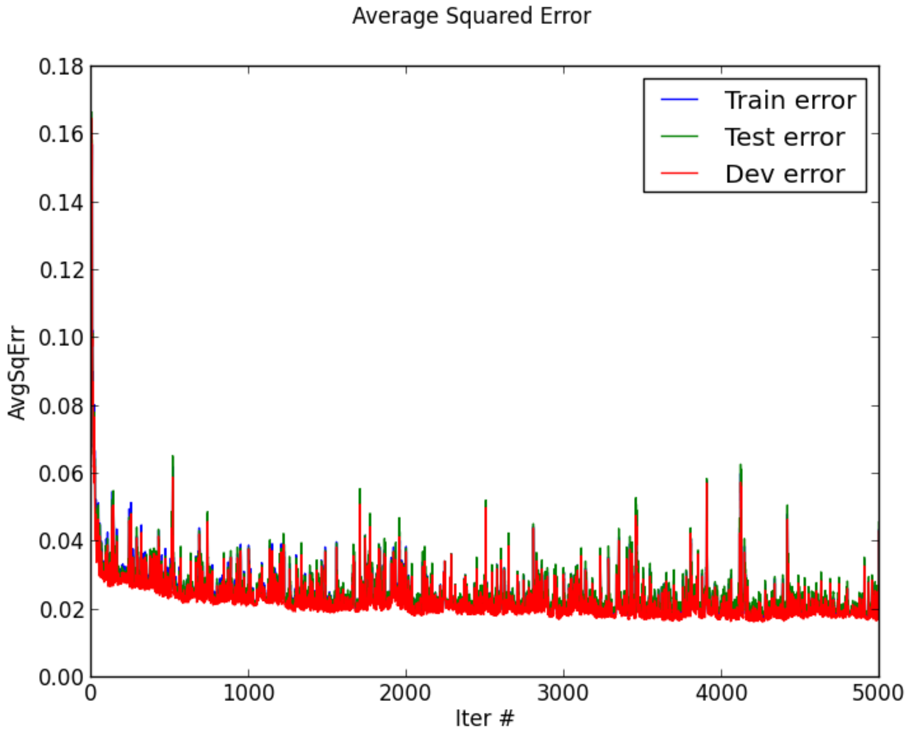




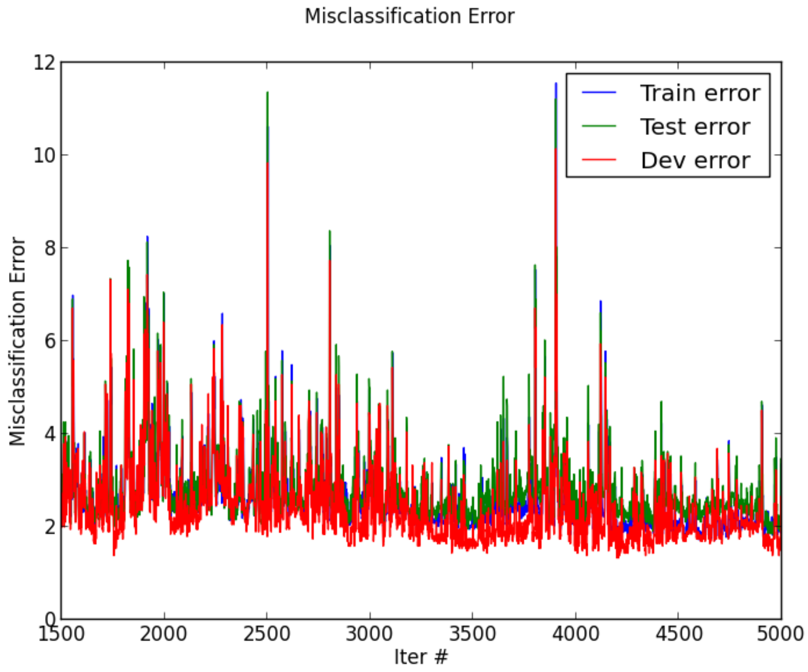
1.3 Linear regression using stochastic gradient descent

1.3.1

a. At a step size of around 0.01. Large step sizes might cause it to overshoot the minimum.
b. Used a lambda of 0.0001, decayed step size by 0.00000001 every iteration



c. Lowest test error: 1.62%



2. Binary Classification with Logistic Regression

2.1.

$$\frac{d}{dw} L_\lambda(w)$$

$$= \frac{d}{dw}\left(\frac{-1}{N}\sum_{n=1}^{N} \log p_w(y=y_n|x_n) + \frac{\lambda}{2}\|w\|^2\right)$$

$$= \frac{d}{dw}\left(\frac{-1}{N}\sum_{n=1}^{N} \log p_w(y=y_n|x_n)\right) + \lambda w$$

Case $y_n = 1$

$$\frac{d}{dw}\left(\frac{-1}{N}\sum_{n=1}^{N} \log p_w(y=1|x_n)\right) + \lambda w$$

$$= \frac{d}{dw}\left(\frac{-1}{N}\sum_{n=1}^{N} \log\left(\frac{1}{1+e^{-wx}}\right)\right) + \lambda w$$

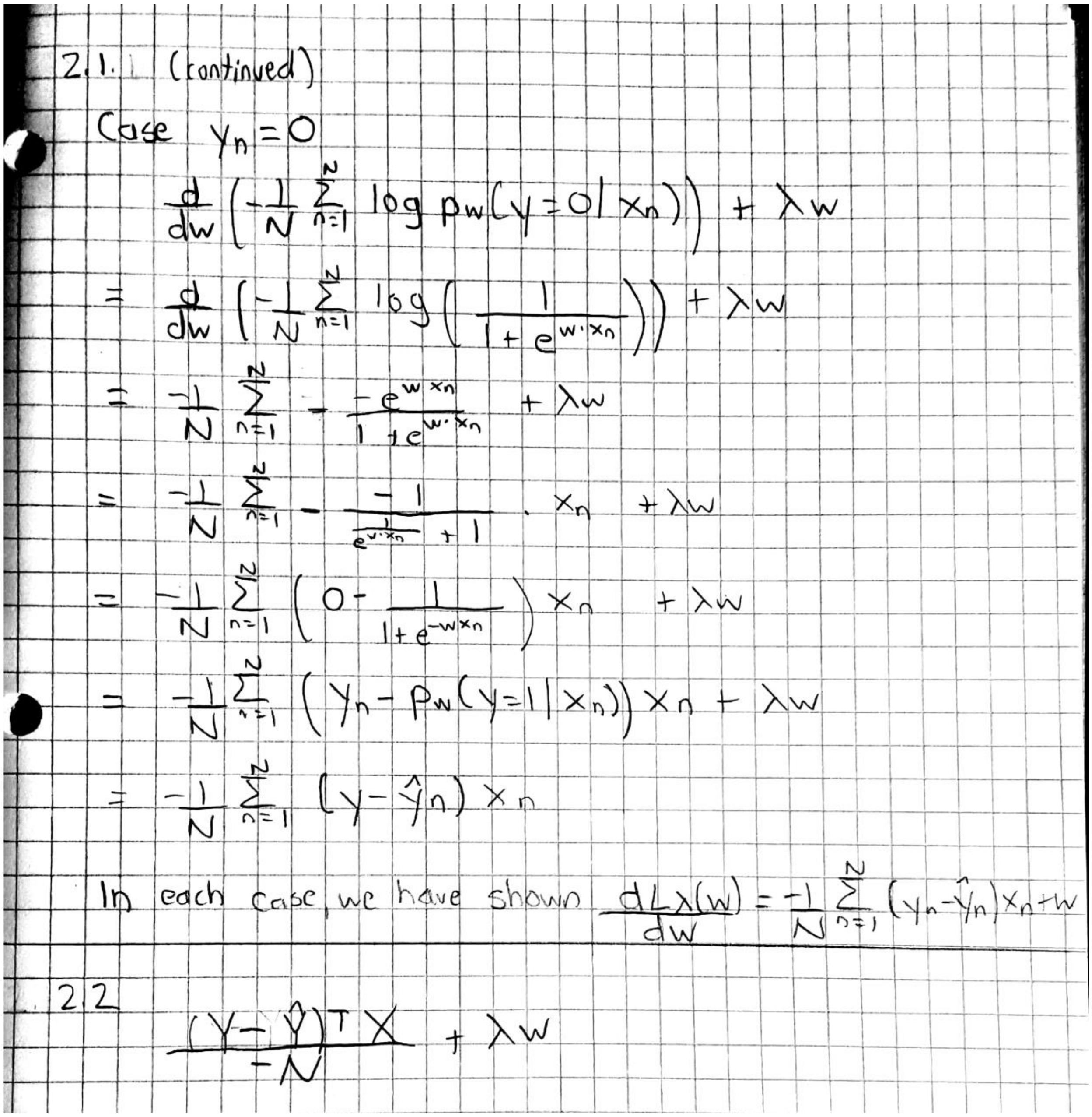
$$= \frac{-1}{N}\sum_{n=1}^{N} \frac{x_n}{1+e^{w\cdot x_n}} + \lambda w$$

$$= \frac{-1}{N}\sum_{n=1}^{N} \frac{1}{1+e^{w\cdot x_n}} \cdot x_n + \lambda w$$

$$= \frac{-1}{N}\sum_{n=1}^{N} \left(1 - \frac{1}{1+e^{-w\cdot x_n}}\right)\cdot x_n + \lambda w$$

$$= \frac{-1}{N}\sum_{n=1}^{N} (y_n - p_w(y=1|x_n))\, x_n + \lambda w$$

$$= \frac{-1}{N}\sum_{n=1}^{N} (y_n - \hat{y}_n)\, x_n + \lambda w$$



2.1. (continued)

Case $y_n = 0$

$$\frac{d}{dw}\left(-\frac{1}{N}\sum_{n=1}^{N}\log p_w(y=0|x_n)\right) + \lambda w$$

$$= \frac{d}{dw}\left(-\frac{1}{N}\sum_{n=1}^{N}\log\left(\frac{1}{1+e^{w\cdot x_n}}\right)\right) + \lambda w$$

$$= -\frac{1}{N}\sum_{n=1}^{N} - \frac{-e^{w x_n}}{1+e^{w\cdot x_n}} + \lambda w$$

$$= -\frac{1}{N}\sum_{n=1}^{N} - \frac{-1}{\frac{1}{e^{w\cdot x_n}}+1}\cdot x_n + \lambda w$$

$$= -\frac{1}{N}\sum_{n=1}^{N}\left(0-\frac{1}{1+e^{-wx_n}}\right)x_n + \lambda w$$

$$= -\frac{1}{N}\sum_{n=1}^{N}\left(y_n - p_w(y=1|x_n)\right)x_n + \lambda w$$

$$= -\frac{1}{N}\sum_{n=1}^{N}(y-\hat{y}_n)x_n$$

In each case, we have shown $\frac{dL_\lambda(w)}{dw} = \frac{-1}{N}\sum_{n=1}^{N}(y_n-\hat{y}_n)x_n + w$

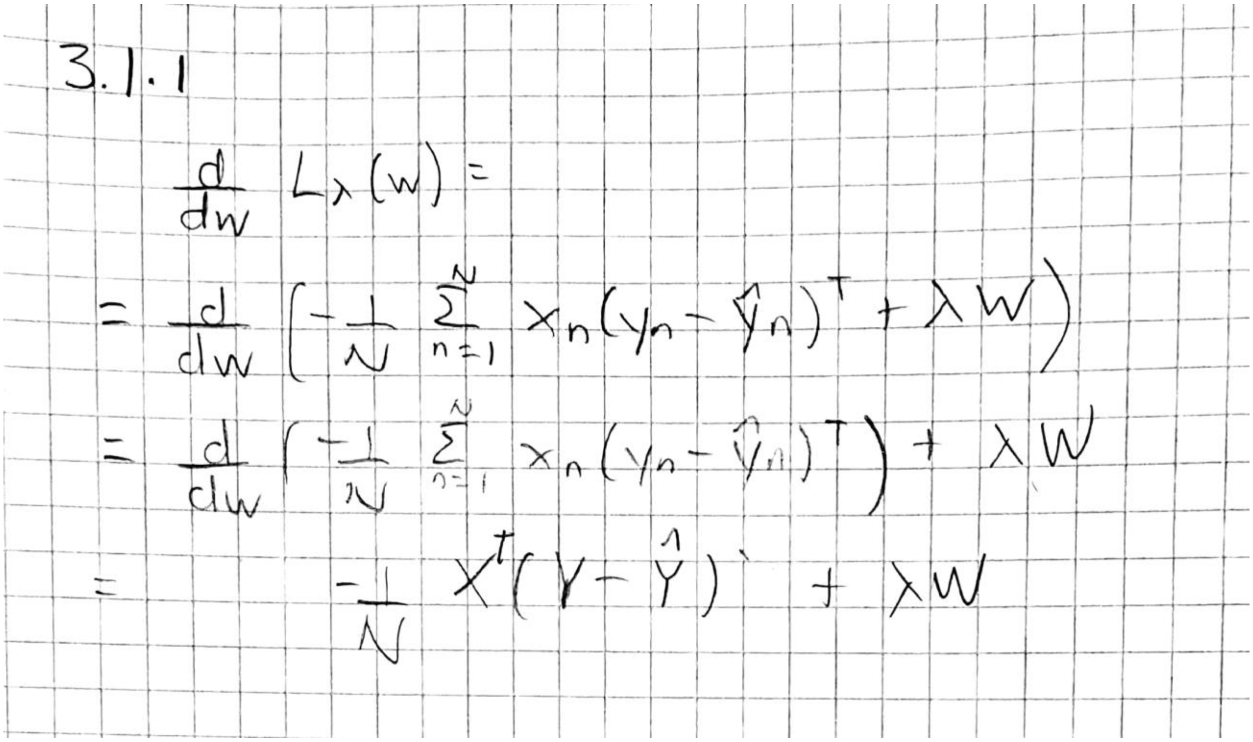
2.2

$$\frac{(Y-\hat{Y})^T X}{-N} + \lambda w$$

2.3

a. Our weight vector would converge to the linearly separable boundary. Without regularization, this would cause overfitting as the magnitude of the weights would go to infinity. This occurs because our data is linearly separable, causing the weight vector to converge to the boundary, but our lambda is 0, causing overfitting.
b. In the case of d > n, our weight vector will converge to a decision boundary that separates our data. In the case of d = n, our weight vector will converge to the exact solution. This is because our data is linearly independent. However, like (a), the magnitude of the weight vector will continue increasing to infinity, as lambda is 0.
c. Regularization makes sense as it will prevent the weight vector from overfitting.

3. Multi-Class Classification using Least Squares



3.1.1

$$\frac{d}{dw} L_\lambda(w) =$$

$$= \frac{d}{dw}\left(-\frac{1}{N}\sum_{n=1}^{N} x_n(y_n-\hat{y}_n)^T + \lambda W\right)$$

$$= \frac{d}{dw}\left(-\frac{1}{N}\sum_{n=1}^{N} x_n(y_n-\hat{y}_n)^T\right) + \lambda W$$

$$= -\frac{1}{N} X^T(Y-\hat{Y}) + \lambda W$$

3.1.2

Used the closed form classifier because it was fast. Tried gradient descent initially but it was very slow.

	Average Squared Error %	Misclassification Error %
Train	18.6%	0.265%
Test	17.5%	0.260%
Dev	16.7%	0.257%

4. Probability and Maximum Likelihood Estimation

4.1 Probability Review

4.1.1 a) good chance of false +

b) Let $+$ be positive test result $= ?$
Let $-$ be negative test resut $= ?$
Let D be have disease $= 0.0001$
Let D^c be don't have disense $= 0.9999$

$P(+|D) = 0.99 \qquad P(-|D^c) = 0.99$

$$P(D|+) = \frac{P(+|D)P(D)}{P(+|D)P(D) + P(+|D^c)P(D^c)}$$

$$= \frac{0.99(0.0001)}{0.99(0.0001) + 0.01 \cdot 0.9999}$$

$$= \frac{1}{102} = 0.0098039216$$

4.1.2 (a) $\frac{34}{151}$ (c) $\frac{41}{64}$

(b) $\frac{34}{87}$

4.1.3 We are looking at a sample of the real distribution, so our probabilities are estimations

4.2 Maximum Likelihood Estimation

4.2.1

Likelihood

$$\prod_{n=0}^{N} p_x(G_n | \lambda) = L(G|\lambda)$$

Log likelihood

$$\ln L(G|\lambda) = \sum_{n=0}^{N} \ln\left[\frac{\lambda^k}{k!} e^{-\lambda}\right]$$

$$= \sum_{n=0}^{N} \ln\left(\frac{\lambda^k}{k!}\right) \cdot -\lambda$$

$$= \sum_{n=0}^{N} \left[\ln(\lambda^k) - \ln(k!)\right] \cdot -\lambda$$

4.2.2

$$\hat{\lambda}_{MLE} = \underset{\lambda}{\operatorname{argmax}}\, L(G|\lambda) = \underset{\lambda}{\operatorname{argmax}}\, \ln L(G|\lambda)$$

$$= \underset{\lambda}{\operatorname{argmax}} \sum_{n=0}^{N} \ln\left(\frac{\lambda^k}{k!}\right) \cdot -\lambda$$

$$\frac{d}{d\lambda}\left[\sum_{n=0}^{N} -\lambda \cdot \left(\ln(\lambda^k) - \ln(k!)\right)\right] = 0$$

$$\hookrightarrow \left(-\lambda \cdot N + \sum_{n=0}^{N} \ln(\lambda^k) - \ln(k!)\right) \frac{d}{d\lambda}$$

$$\hookrightarrow \left(-\lambda \cdot N + \sum_{n=0}^{N} k \cdot \ln(\lambda)\right) \frac{d}{d\lambda}$$

$$\hookrightarrow \left(-N + \sum_{n=0}^{N} \frac{k}{\lambda}\right) = \boxed{\frac{1}{\lambda} \sum_{n=0}^{N} k - N}$$

Now, take 2nd derivative w/ respect to λ

$$\left[\left(\sum_{n=0}^{N} \frac{k}{\lambda}\right) - N\right] \frac{d}{d\lambda} = \boxed{\frac{-1}{\lambda^2} \sum_{n=0}^{N} k}$$

4.2.2 (cont.)

$$-\frac{1}{\lambda^2}\sum_{n=0}^{N} K$$

the 2nd derivative is clearly negative because $\frac{1}{\lambda^2}\sum_{n=0}^{N} k$ is always positive, as k cannot take on negative values squared values are always positive. Neg · positive = neg.

Solving for λ...

$$\frac{1}{\lambda}\sum_{n=0}^{N} k - N = 0$$

$$\frac{1}{\lambda}\sum_{n=0}^{N} k = N$$

$$\sum_{n=0}^{N} k = N\lambda$$

$$\boxed{\frac{\sum_{n=0}^{N} k}{N} = \lambda}$$

this is $\hat{\lambda}$ that maximizes.

4.2.3 Now, we plug in our actual observed values of G for $\sum_{n=0}^{N} K$

$$\lambda = \frac{\sum_{n=0}^{N} k}{N}$$

$$= \frac{6+4+2+7+5+1+2+5}{8} = \frac{32}{8} = \boxed{4}$$