Homework #2

Spring 2020, CSE 446/546: Machine Learning
Richy Yun
Due: 5/12/20 11:59 PM

Conceptual questions

A.0

- *[2 points]* Suppose that your estimated model for predicting house prices has a large positive weight on 'number of bathrooms'. Does it implies that if we remove the feature "number of bathrooms" and refit the model, the new predictions will be strictly worse than before? Why?

  No it does not. The high weight on 'number of bathrooms' could have been leading to overfitting to that specific feature, whereas removing it could lead to a more generalized model.

- *[2 points]* Compared to L2 norm penalty, explain why a L1 norm penalty is more likely to result in a larger number of 0s in the weight vector or not?

  A L1 norm penalty is more likely to result in a larger number of 0s in the weight vector as it provides much more sparse estimates. As visualized in lecture, the point that minimizes loss during regularization is more likely to fall on an axis which will simply choose that index in the weight vector and set the others to 0.

- *[2 points]* In at most one sentence each, state one possible upside and one possible downside of using the following regularizer: $(\sum_i |w_i|^{0.5})$

  A possible upside is that a square root doesn't punish large values of w as strongly, ensuring they remain influential at the risk of overfitting. A possible downside is that small differences in the smaller values of w will cause large changes leading to overgeneralization.

- *[1 points]* True or False: If the step-size for gradient descent is too large, it may not converge.

  True. If the step size is too large, you may skip over the minima and instead end up diverging.

- *[2 points]* In your own words, describe why SGD works.

- *[2 points]* In at most one sentence each, state one possible advantage of SGD (stochastic gradient descent) over GD (gradient descent) and one possible disadvantage of SGD relative to GD.

Convexity and Norms

A.1

a. *[3 points]* The first two properties clearly hold: i) $f(x)$ is nonnegative as it is a sum of positive values unless the values are 0. ii) A summation is a linear transformation so we can pull out any constant that is multiplied to x, thus giving absolute scalability. The third condition can be proven:

$$\begin{aligned} a &\leq |a| \text{ (by definition)} \\ a+b &\leq |a|+|b| \\ (a+b)^2 &\leq (|a|+|b|)^2 \\ |a+b| &\leq |a|+|b| \end{aligned}$$

The condition holds true through the summation, and thus $f(x)$ is a norm.

b. *[2 points]* Considering two points in $n = 2$: $(0, 1)$ and $(1, 1)$, we have:

$$\begin{aligned} g(0,1) &= 1 \\ g(1,1) &= 1 \\ g(1,2) &= (1+\sqrt{2})^2 \\ g(1,2) &> g(0,1) + g(1,1) \end{aligned}$$

Thus the triangle inequality does not hold and $g(x)$ is not a norm.

B.1 *[6 points]* For this problem we simply need to show $||x||_n \leq ||x||_{n-1}$:

$$\begin{aligned} ||x||_n &\leq ||x||_{n-1} \\ \left(\sum_{i=1}^{n} |x_i|^n\right)^{1/n} &\leq \left(\sum_{i=1}^{n} |x_i|^{n-1}\right)^{1/n-1} \\ \sum_{i=1}^{n} |x_i|^n &\leq \left(\sum_{i=1}^{n} |x_i|^{n-1}\right)^{n/n-1} \\ \sum_{i=1}^{n} y_i^m &\leq \left(\sum_{i=1}^{n} y_i\right)^{m} \end{aligned}$$

where $y_i = |x_i|^{n-1}$ and $m = \frac{n}{n-1}$. Thus the problem is showing that the sum of an mth power is less than or equal to the mth power of the sum:

$$\sum_{i=1}^{n} y_i^m \leq \left(\sum_{i=1}^{n} y_i\right)^{m}$$

$$\text{If } \sum_{i=1}^{n} y_i = Y \text{ then by definition } \frac{y_i}{Y} \in [0,1] \text{ and } \left(\frac{y_i}{Y}\right)^m \leq \frac{y_i}{Y}$$

$$\sum_{i=1}^{n} y_i^m = Y^m \sum_{i=1}^{n} \left(\frac{y_i}{Y}\right)^m \leq Y^m \sum_{i=1}^{n} \frac{y_i}{Y} = Y^m = \left(\sum_{i=1}^{n} y_i\right)^{m}$$

Therefore, we know $||x||_n \leq ||x||_{n-1}$ and it follows that $||x||_\infty \leq ||x||_2 \leq ||x||_1$.

A.2 *[3 points]* II is convex. I is not convex as the line between points b and c lies outside the set. III is not convex as the line between points a and d lies outside the set.

A.3 *[4 points]*

a. It is convex.

b. It is not convex. The line between points a and b lies below the function.

c. It is not convex. The line between points a and c lies below the function.

d. It is convex.

B.2

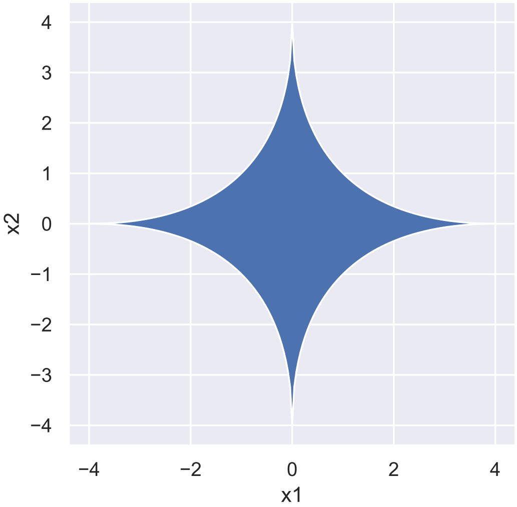
a. *[3 points]* As mentioned in lecture, we need to show $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$ to prove convexity. Since we know a norm follows absolute scalability and triangle inequality, the condition holds true and all norm functions are convex.

b. *[3 points]* $\{x \in \mathbb{R}^n : ||x|| \leq 1\}$ is essentially a ball around the origin which is intuitively convex. To prove we can show:

$$||\lambda x + (1-\lambda)y|| \leq \lambda||x|| + (1-\lambda)||y|| \leq \lambda + (1-\lambda) = 1$$

The first inequality is due to the triangle inequality of norm and the second due to the condition of the set. The inequality holds and all values are within the limits of the set and so the set is convex.

c. *[2 points]* The defined set is not convex as a line between two points can lie outside of the set. (note: the 'tips' of the plot go all the way to 4, but the values are too small to be accurately represented)



B.3

a. *[3 points]* If f, g are convex functions, then their sum is also convex:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$
$$g(\lambda x + (1-\lambda)y) \leq \lambda g(x) + (1-\lambda)g(y)$$
$$f(\lambda x + (1-\lambda)y) + g(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) + \lambda g(x) + (1-\lambda)g(y)$$
$$f(\lambda x + (1-\lambda)y) + g(\lambda x + (1-\lambda)y) \leq \lambda(f(x) + g(x)) + (1-\lambda)(f(y) + g(y))$$
$$h(x) = f(x) + g(x)$$
$$h(\lambda x + (1-\lambda)y) \leq \lambda h(x) + (1-\lambda)h(y)$$

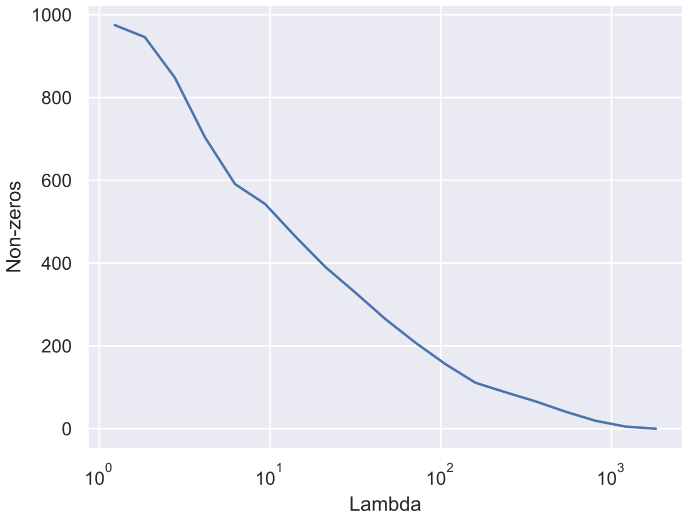
Since $\ell_i(w)$ is given to be convex over $w \in \mathbb{R}^d$ and a norm is always convex, their sum is convex over $w \in \mathbb{R}^d$ as well.

b. *[1 points]* If the function is convex the local minima is the global minima which allows us to find the absolute minimum loss.

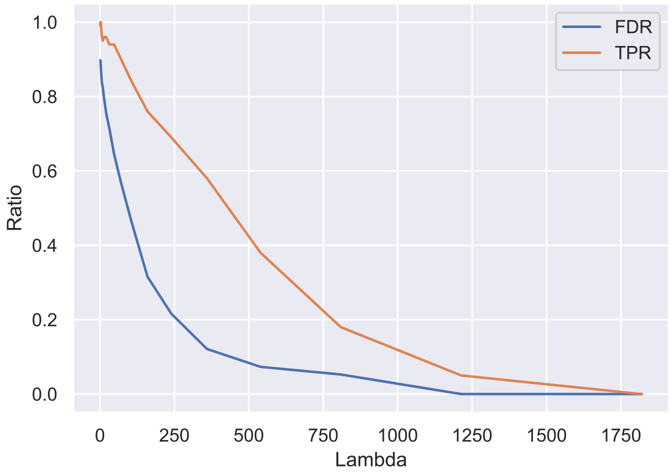
Lasso

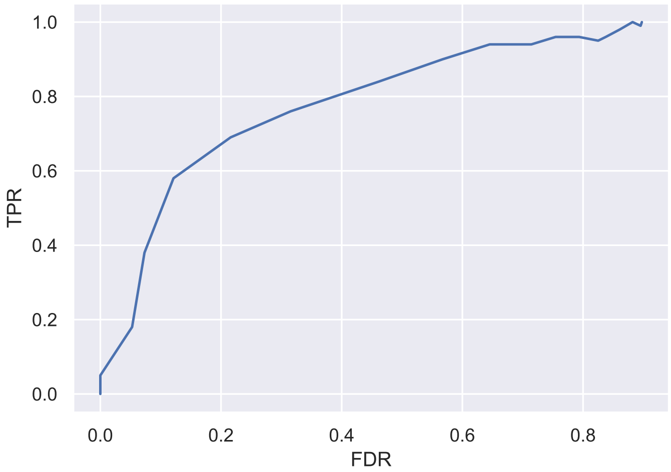
A.4

a. *[10 points]*



b. *[10 points]*

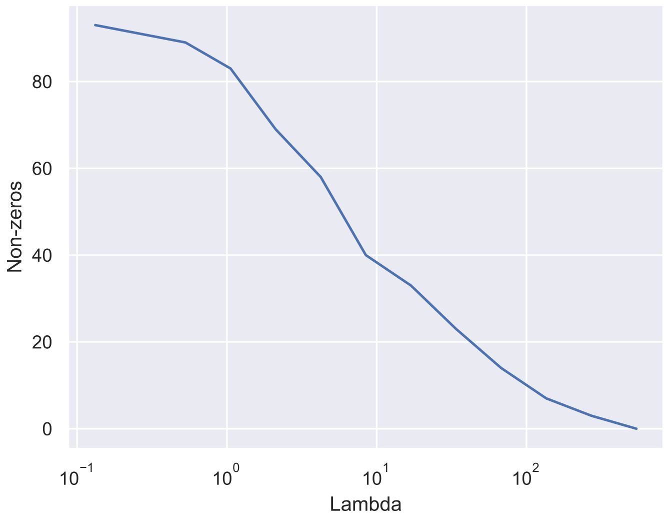




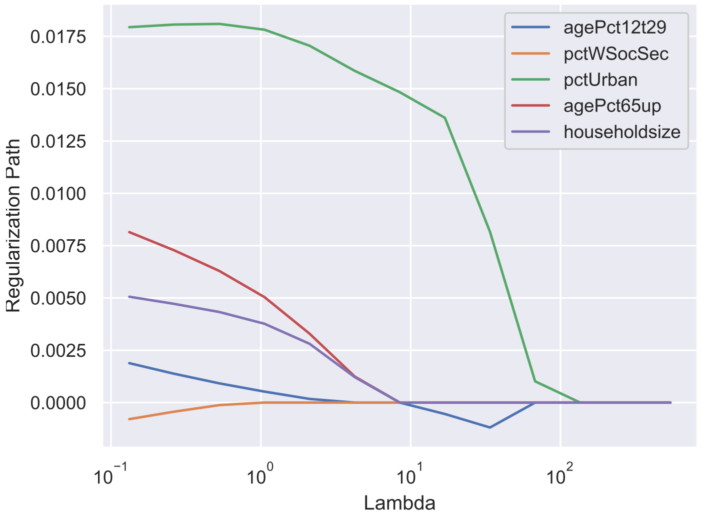
c. *[5 points]* As λ becomes smaller the regularization becomes weaker allowing for more features to be selected. As a result we see an increase in the number of non-zero weights as well as both FDR and TPR. TPR increases much faster than FDR until the "larger" weights are nearly all picked, showing the algorithm does a good job of picking the correct features.

A.5

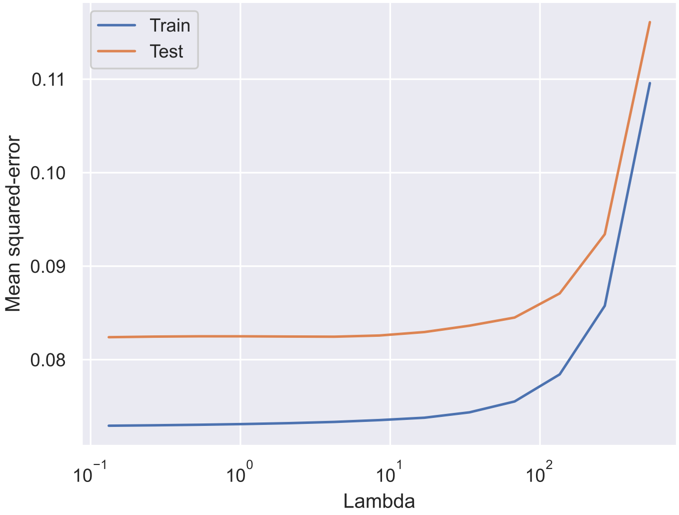
a. *[4 points]*



b. *[4 points]*



c. *[4 points]*



d. *[4 points]* For $\lambda = 30$ PctIlleg has the largest Lasso coefficient and PctFam2Par the lowest. PctIlleg is percentage of kids born to unmarried couples, and PctFam2Par is percentage of families with kids that are headed by two parents. They make intuitive sense as higher crime is often correlated with areas with low socioeconomic status, leading to poor healthcare and access to contraception. More families with both parents present generally indicate stability, thus negatively correlating with crime.

e. *[4 points]* Correlation does not mean causation. There are various factors that correlate with people living to be older, such as income, that could be the true underlying cause. Thus simply moving people over 65 to an area will not decrease the crime rate.

Logistic Regression

A.6

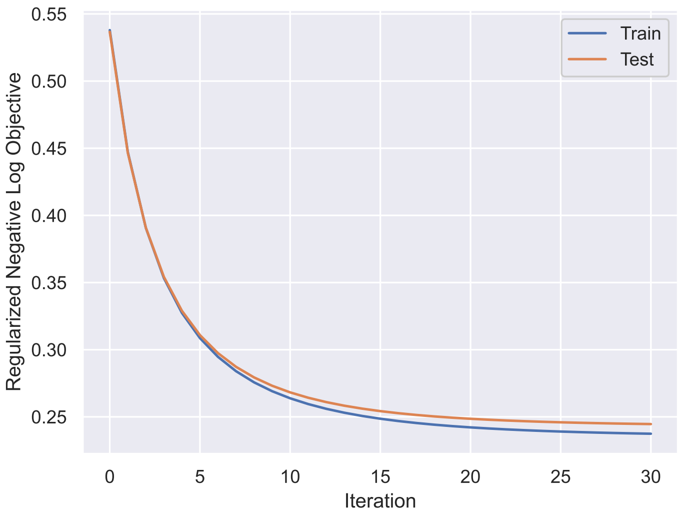
a. *[8 points]* To find $\nabla_w J(w, b)$:

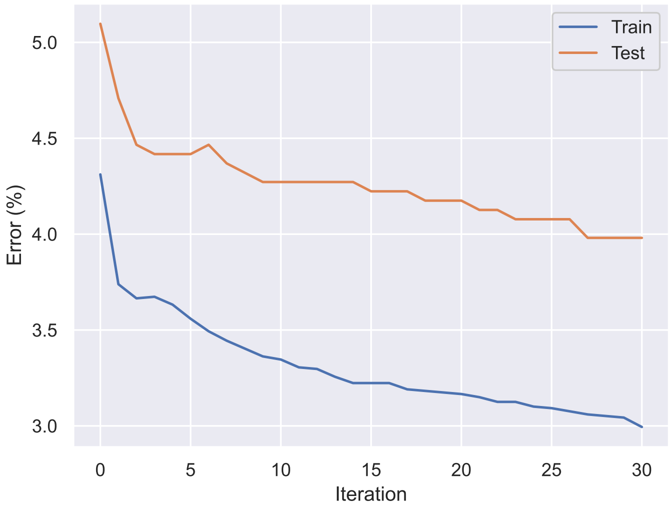
$$\begin{aligned}
\nabla_w J(w,b) &= \nabla_w \left[\frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i(b + x_i^T w))) + \lambda ||w||_2^2 \right] \\
&= \frac{1}{n} \sum_{i=1}^{n} \frac{-y_i x_i^T \exp(-y_i(b + x_i^T w))}{1 + \exp(-y_i(b + x_i^T w))} + 2\lambda w \\
&= \frac{1}{n} \sum_{i=1}^{n} -y_i x_i^T (1 - \mu_i(w,b)) + 2\lambda w
\end{aligned}$$

To find $\nabla_b J(w, b)$:

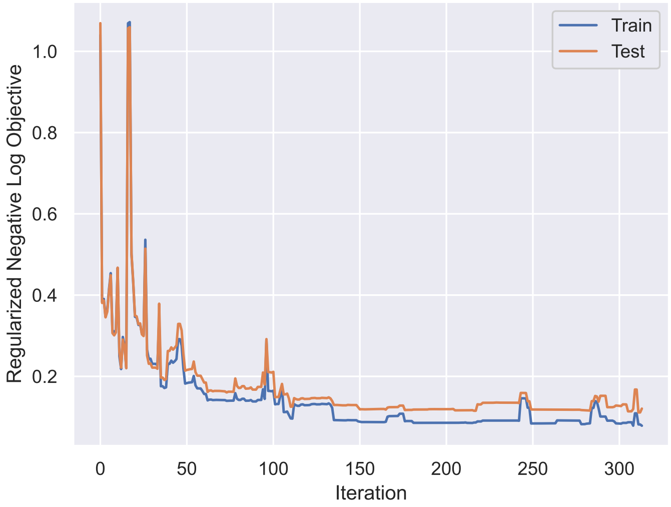
$$\begin{aligned}
\nabla_b J(w,b) &= \nabla_b \left[\frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i(b + x_i^T w))) + \lambda ||w||_2^2 \right] \\
&= \frac{1}{n} \sum_{i=1}^{n} \frac{-y_i \exp(-y_i(b + x_i^T w))}{1 + \exp(-y_i(b + x_i^T w))} \\
&= \frac{1}{n} \sum_{i=1}^{n} -y_i (1 - \mu_i(w,b))
\end{aligned}$$

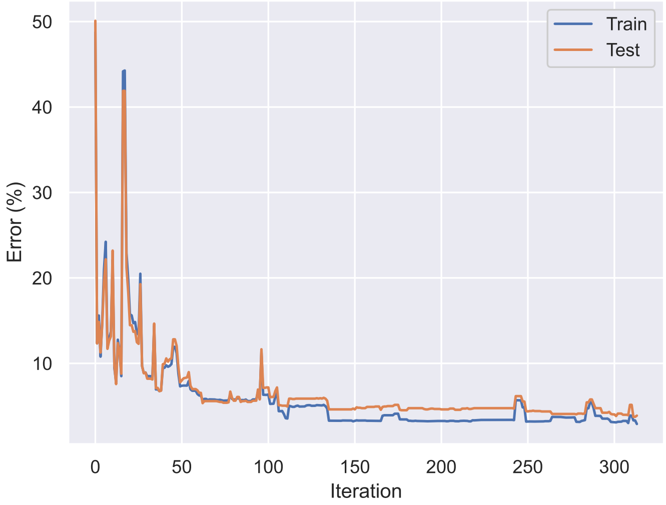
b. *[8 points]*


Regularized Negative Log Objective
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0
5
10
15
20
25
30
Iteration
Train
Test

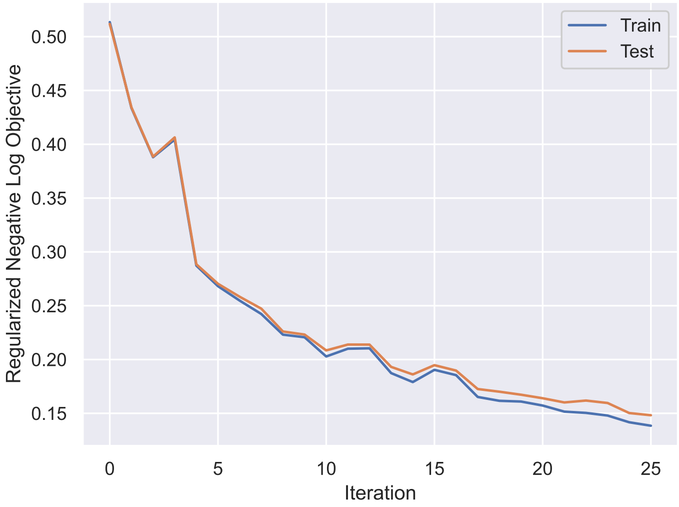

Error (%)
5.0
4.5
4.0
3.5
3.0
0
5
10
15
20
25
30
Iteration
Train
Test

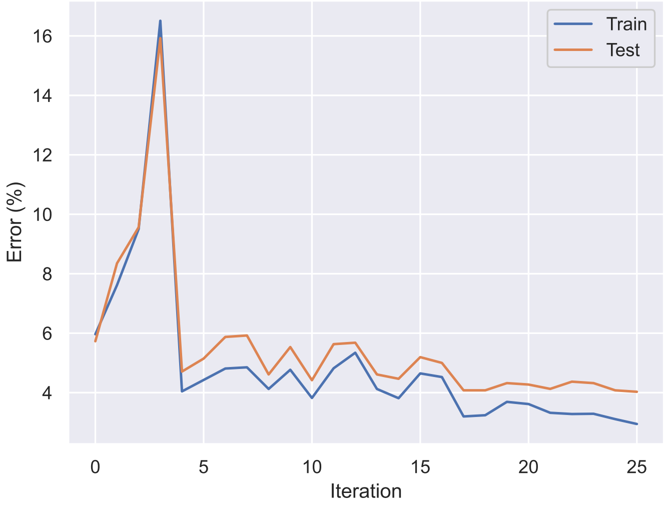
c. *[7 points]*


Regularized Negative Log Objective
1.0
0.8
0.6
0.4
0.2
0
50
100
150
200
250
300
Iteration
Train
Test


Error (%)
50
40
30
20
10
0
50
100
150
200
250
300
Iteration
Train
Test

d. *[7 points]*





B.4

a. *[5 points]* The partial derivative can be computed for a specific w then extrapolated to matrix form for W. If we set $\hat{\mathbf{y}}_i^{(\mathbf{w}^{(m)})} = \texttt{softmax}(\mathbf{w}^{(m)} \cdot \mathbf{x}_i)$:

$$\nabla_{\mathbf{w}^{(m)}}\mathcal{L} = \frac{\partial}{\partial \mathbf{w}^{(m)}} - \sum_{i=1}^{n} \sum_{\ell=i}^{k} \mathbf{1}\{y_i = \ell\} \log(\hat{\mathbf{y}}_i^{(\mathbf{w}^{(\ell)})})$$

$$= -\sum_{i=1}^{n} \sum_{\ell=i}^{k} \mathbf{1}\{y_i = \ell\} \frac{1}{\hat{\mathbf{y}}_i^{(\mathbf{w}^{(\ell)})}} \frac{\partial \hat{\mathbf{y}}_i^{(\mathbf{w}^{(\ell)})}}{\partial \mathbf{w}^{(m)}}$$

To solve for the partial derivative of the $\texttt{softmax}$ function, in the first case if $m = \ell$ we have:

$$\begin{aligned}\frac{\partial \hat{\mathbf{y}}_i^{(\mathbf{w}^{(\ell)})}}{\partial \mathbf{w}^{(m)}} &= \frac{\mathbf{x}_i \exp(\mathbf{w}^{(\ell)}\mathbf{x}_i) \sum_{j=1}^k \exp(\mathbf{w}^{(j)}\mathbf{x}_i) - \mathbf{x}_i \exp(\mathbf{w}^{(m)}\mathbf{x}_i) \exp(\mathbf{w}^{(\ell)}\mathbf{x}_i)}{(\sum_{j=1}^k \exp(\mathbf{w}^{(j)}\mathbf{x}_i))^2} \\ &= \mathbf{x}_i \frac{\exp(\mathbf{w}^{(\ell)}\mathbf{x}_i)}{\sum_{j=1}^k \exp(\mathbf{w}^{(j)}\mathbf{x}_i)} \frac{\sum_{j=1}^k \exp(\mathbf{w}^{(j)}\mathbf{x}_i) - x_i \exp(\mathbf{w}^{(m)}\mathbf{x}_i)}{\sum_{j=1}^k \exp(\mathbf{w}^{(j)}\mathbf{x}_i)} \\ &= \mathbf{x}_i \hat{\mathbf{y}}_i^{(\mathbf{w}^{(\ell)})}(1 - \hat{\mathbf{y}}_i^{(\mathbf{w}^{(m)})})\end{aligned}$$

In the second case when $m \neq \ell$ we have:

$$\begin{aligned}\frac{\partial \hat{\mathbf{y}}_i^{(\mathbf{w}^{(\ell)})}}{\partial \mathbf{w}^{(m)}} &= \frac{0 - \mathbf{x}_i \exp(\mathbf{w}^{(m)}\mathbf{x}_i) \exp(\mathbf{w}^{(\ell)}\mathbf{x}_i)}{(\sum_{j=1}^k \exp(\mathbf{w}^{(j)}\mathbf{x}_i))^2} \\ &= \mathbf{x}_i(-\hat{\mathbf{y}}_i^{(\mathbf{w}^{(m)})})(\hat{\mathbf{y}}_i^{(\mathbf{w}^{(\ell)})})\end{aligned}$$

Combining them, we have:

$$\begin{aligned}\nabla_{\mathbf{w}^{(m)}} \mathcal{L} &= -\sum_{i=1}^n \mathbf{x}_i \left(\mathbf{1}\{y_i = m\}(1 - \hat{\mathbf{y}}_i^{(\mathbf{w}^{(m)})}) - \sum_{\ell \neq m}^k \mathbf{1}\{y_i = \ell\} \hat{\mathbf{y}}_i^{(\mathbf{w}^{(m)})} \right) \\ &= -\sum_{i=1}^n \mathbf{x}_i \left(\mathbf{1}\{y_i = m\} - \hat{\mathbf{y}}_i^{(\mathbf{w}^{(m)})} \sum_{\ell=1}^k \mathbf{1}\{y_i = \ell\} \right) \\ &= -\sum_{i=1}^n \mathbf{x}_i \left(\mathbf{1}\{y_i = m\} - \hat{\mathbf{y}}_i^{(\mathbf{w}^{(m)})} \right)\end{aligned}$$

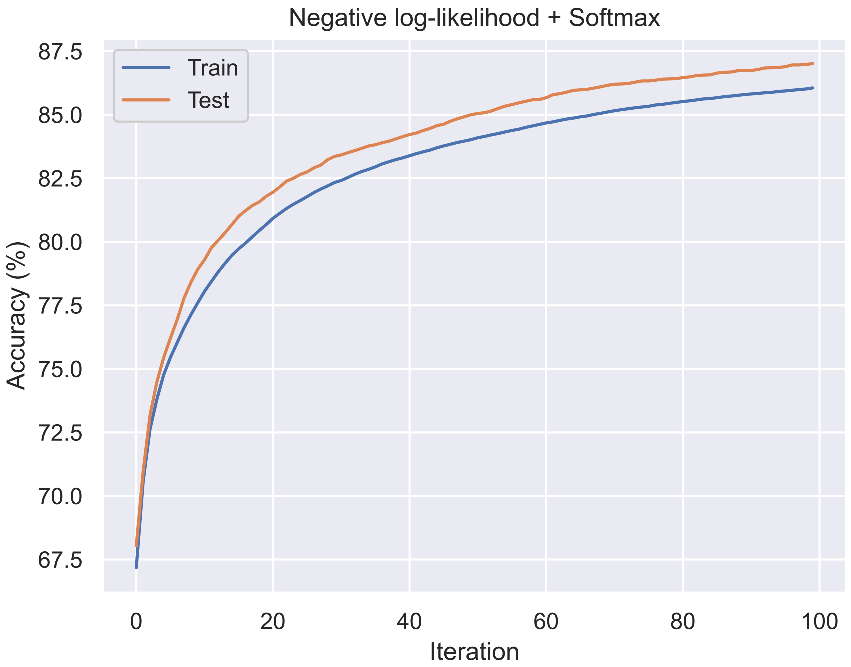
Expanding to matrix format for W we finally have:

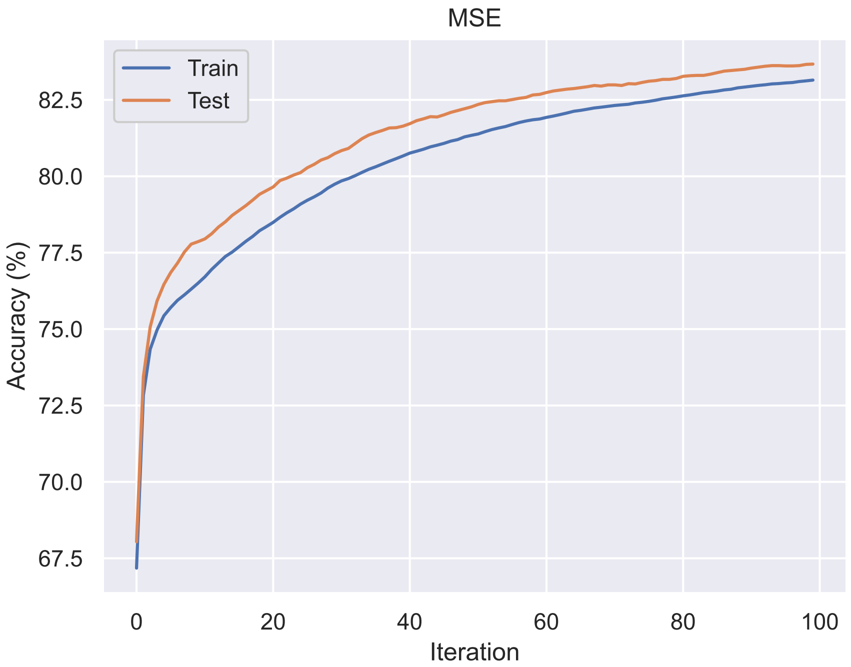
$$\nabla_W \mathcal{L} = -\sum_{i=1}^n \mathbf{x}_i(\mathbf{y}_i - \hat{\mathbf{y}}_i^W)^\top$$

b. *[5 points]* Since $\frac{d}{dx}||x||_2^2 = 2x$ we have:

$$\begin{aligned}\nabla_W J(W) &= \frac{\partial}{\partial W} \frac{1}{2} \sum_{i=1}^n ||\mathbf{y}_i - W^\top \mathbf{x}_i||_2^2 \\ &= -\sum_{i=1}^n \mathbf{x}_i(\mathbf{y}_i - W^\top \mathbf{x}_i)^\top \\ &= -\sum_{i=1}^n \mathbf{x}_i(\mathbf{y}_i - \widetilde{\mathbf{y}}_i^{(W)})^\top\end{aligned}$$

c. *[15 points]* Using a step size of 0.1, I noticed both methods converged at around 100 iterations and so did 100 to more easily compare them. Both methods work fairly well, with NLL+softmax reaching higher accuracy compared to MSE. MSE however performs better at much earlier iterations but plateaus faster.


Negative log-likelihood + Softmax
Train
Test
Accuracy (%)
87.5
85.0
82.5
80.0
77.5
75.0
72.5
70.0
67.5
0
20
40
60
80
100
Iteration


MSE
Train
Test
Accuracy (%)
82.5
80.0
77.5
75.0
72.5
70.0
67.5
0
20
40
60
80
100
Iteration