Ross Brancati
CS 589 – Homework 4 Report
October 29, 2021

1. Linear Regression and Beyond

1.

$$\|y - Xw\|_2^2 + \lambda\|w\|_2^2 = (y - Xw)^T(y - Xw) + \lambda w^T w$$

$$= w^T X^T Xw - 2y^T Xw + y^T y + \lambda w^T w$$

$$\nabla_w = 2(X^T X)w - 2X^T y + 2\lambda w = 2((X^T X + \lambda I)w - X^T y)$$

$\frac{\delta}{\delta w} = 0$ yields:

$$X^T y = (X^T X + \lambda I)w$$

$$w^* = (X^T X + \lambda I)^{-1} X^T y$$

2.

For this question, we replace the X matrix with $\Phi(X)$ which yields:

$$w^* = (\Phi(X)^T \Phi(X) + \lambda I)^{-1} \Phi(X)^T y$$

3.

The kernel trick: $k(x_i^T, x_{new}) = \langle x_i^T, x_{new} \rangle$

From the previous question, $w^* = (\Phi(X)^T \Phi(X) + \lambda I)^{-1} \Phi(X)^T y$

Using Matrix Inversion Lemma:
$w^* = \Phi(X)^T (\Phi(X)\Phi(X)^T + \lambda I)^{-1} y$

The predictions are then calculated as $\hat{y} = (\Phi(X)^T (\Phi(X)\Phi(X)^T + \lambda I)^{-1} y)^T \Phi(X_{new})$

$$\hat{y} = ((\Phi(X)\Phi(X)^T + \lambda I)^{-1} y)^T \, (\Phi(X) \cdot \Phi(X_{new}))$$

The above equation shows that this can be written in terms of $\langle x_i^T, x_{new} \rangle$

$$\hat{y} = ((\Phi(X)\Phi(X)^T + \lambda I)^{-1} y)^T \, k(x_i^T, x_{new})$$

4.

Table 1: Mean Absolute Error on the housing prices test dataset for Ordinary Least Squares, Lasso, and Ridge Linear Regression Models

Mean Absolute Error on Test Set			
Model	**OLS**	**Lasso**	**Ridge**
MAE	18619.22	17866.64	18340.01

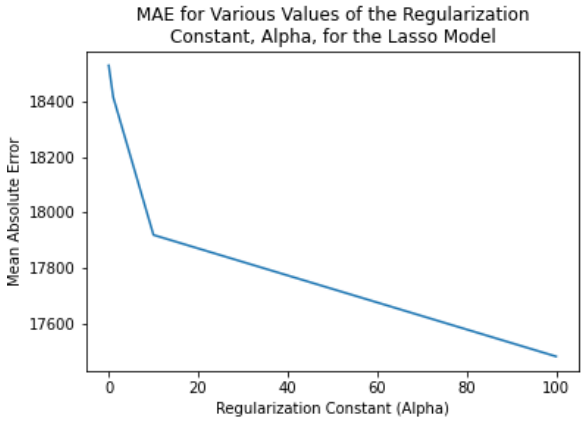

Figure 1: Mean Absolute Error plotted as a function of the Regularization Constant (Alpha) for the Lasso Linear Regression model with maximum number of iterations set to the best performing model from the hyper_parameter_tuning function. Alpha values tested were 0.001, 0.01, 0.1, 0.2, 0.5, 1, 10, and 100. In general, the MAE declines as alpha increases, and appears to have a linear relationship with the values of alpha once alpha is greater than or equal to 10.

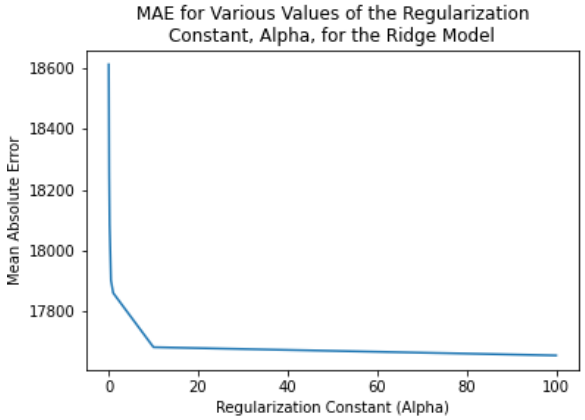

Figure 2: Mean Absolute Error plotted as a function of the Regularization Constant (Alpha) for the Ridge Linear Regression model with maximum number of iterations set to the best performing model from the hyper_parameter_tuning function. The MAE declines sharply when alpha is small, then begins to plateau for values of alpha greater than or equal to 10. This means that increasing alpha to a larger number would not help to decrease the MAE on this dataset.

2. Fully Connected Neural Network

1.

$$z^{(1)} = \sigma(W_1 x + b_1)$$

$$z^{(2)} = \sigma\left(W_2 z^{(1)} + b_2\right) = \sigma(W_2 \sigma(W_1 x + b_1) + b_2)$$

$$z^{(3)} = \hat{y} = \sigma\left(W_3 z^{(2)} + b_3\right) = \sigma(W_3 \sigma(W_2 \sigma(W_1 x + b_1) + b_2) + b_3)$$

2.

$$L(y, \hat{y}) = -(y(log\hat{y}) + (1-y)\log(1-\hat{y}))$$

(a)

$$\frac{\delta L}{\delta \hat{y}} = -\left(\frac{\delta}{\delta \hat{y}} y log(\hat{y}) + \frac{\delta}{\delta \hat{y}}(1-y) log(1-\hat{y})\right)$$

$$\frac{\delta L}{\delta \hat{y}} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)$$

(b)

$$\frac{\delta L}{\delta W_3} = \frac{\delta L}{\delta \hat{y}} \frac{\delta \hat{y}}{\delta W_3} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)\left(\frac{\delta}{\delta W_3}\left(\sigma\left(W_3 z^{(2)} + b_3\right)\right)\right)$$

$$\frac{\delta}{\delta W_3}\left(\sigma\left(W_3 z^{(2)} + b_3\right)\right) = \hat{y}(1-\hat{y})\left(z^{(2)}\right)^T$$

$$\frac{\delta L}{\delta W_3} = (\hat{y} - y)\left(z^{(2)}\right)^T$$

(c)

$$\frac{\delta L}{\delta b_3} = \frac{\delta L}{\delta \hat{y}} \frac{\delta \hat{y}}{\delta b_3} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)\left(\frac{\delta}{\delta b_3}\left(\sigma\left(W_3 z^{(2)} + b_3\right)\right)\right)$$

$$\left(\frac{\delta}{\delta b_3}\left(\sigma\left(W_3 z^{(2)} + b_3\right)\right)\right) = 1$$

$$\frac{\delta L}{\delta b_3} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)$$

(d)

$$\frac{\delta L}{\delta W_2} = \frac{\delta L}{\delta \hat{y}} \frac{\delta \hat{y}}{\delta z^{(2)}} \frac{\delta z^{(2)}}{\delta W_2}$$

From previous questions, we can determine that $\frac{\delta L}{\delta \hat{y}}\frac{\delta \hat{y}}{\delta z^{(2)}} = (\hat{y} - y)(W_3)^T$

$$\frac{\delta z^{(2)}}{\delta W_2} = \frac{\delta}{\delta W_2}\sigma\left(W_2 z^{(1)} + b_2\right) = z^{(2)}(1 - z^{(2)})(z^{(1)})^T$$

$$\frac{\delta z^{(2)}}{\delta W_2} = ((\hat{y} - y)(W_3)^T)(z^{(2)}\left(1 - z^{(2)}\right)\left(z^{(1)}\right)^T)$$

(e)

$$\frac{\delta L}{\delta b_2} = \frac{\delta L}{\delta \hat{y}}\frac{\delta \hat{y}}{\delta z^{(2)}}\frac{\delta z^{(2)}}{\delta b_2}$$

$$\frac{\delta z^{(2)}}{\delta b_2} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)$$

$$\frac{\delta L}{\delta b_2} = (\hat{y} - y)(W_3)^T\left(-\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)\right)$$

3.

$$z^{(1)} = \text{ReLU}(W_1 x + b_1) = \max(0, W_1 x + b_1) = \begin{cases} W_1 x + b_1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

$$z^{(2)} = \text{ReLU}(W_2\text{ReLU}(W_1 x + b_1) + b_2) = \max(0, W_2\text{ReLU}(W_1 x + b_1) + b_2)$$
$$= \begin{cases} W_2\text{ReLU}(W_1 x + b_1) + b_2 & z^{(1)} > 0 \\ 0 & z^{(1)} \leq 0 \end{cases}$$

$$z^{(3)} = \hat{y} = \sigma\left(W_3 z^{(2)} + b_3\right) = \sigma(W_3\text{ReLU}(W_2\text{ReLU}(W_1 x + b_1) + b_2) + b_3)$$

4.

(a)

$$\frac{\delta L}{\delta \hat{y}} = -\left(\frac{\delta}{\delta \hat{y}} y log(\hat{y}) + \frac{\delta}{\delta \hat{y}}(1 - y) log(1 - \hat{y})\right)$$

$$\frac{\delta L}{\delta \hat{y}} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)$$

(b)

$$\frac{\delta L}{\delta W_3} = \frac{\delta L}{\delta \hat{y}}\frac{\delta \hat{y}}{\delta W_3} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)\left(\frac{\delta}{\delta W_3}\left(\sigma\left(W_3 z^{(2)} + b_3\right)\right)\right)$$

$$\frac{\delta}{\delta W_3}\left(\sigma\left(W_3 z^{(2)} + b_3\right)\right) = \hat{y}(1 - \hat{y})\left(z^{(2)}\right)^T$$

$$\frac{\delta L}{\delta W_3} = (\hat{y} - y)\left(z^{(2)}\right)^T = \begin{cases} (\hat{y} - y)(z^{(2)})^T & z^{(1)} > 0 \\ 0 & z^{(1)} \leq 0 \end{cases}$$

(c)

$$\frac{\delta L}{\delta b_3} = \frac{\delta L}{\delta \hat{y}} \frac{\delta \hat{y}}{\delta b_3} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)\left(\frac{\delta}{\delta b_3}\left(\sigma\left(W_3 z^{(2)} + b_3\right)\right)\right)$$

$$\frac{\delta L}{\delta b_3} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)$$

(d)

$$\frac{\delta L}{\delta W_2} = \frac{\delta L}{\delta \hat{y}} \frac{\delta \hat{y}}{\delta z^{(2)}} \frac{\delta z^{(2)}}{\delta W_2}$$

$$\frac{\delta L}{\delta \hat{y}} \frac{\delta \hat{y}}{\delta z^{(2)}} = (\hat{y} - y)(W_3)^T$$

$$\frac{\delta z^{(2)}}{\delta W_2} = \begin{cases} z^{(1)} & z^{(2)} > 0 \\ 0 & z^{(2)} \leq 0 \end{cases}$$

$$\frac{\delta L}{\delta W_2} = \begin{cases} ((\hat{y} - y)(W_3)^T) z^{(1)} & z^{(2)} > 0 \\ 0 & z^{(2)} \leq 0 \end{cases}$$

(e)

$$\frac{\delta L}{\delta b_2} = \frac{\delta L}{\delta \hat{y}} \frac{\delta \hat{y}}{\delta z^{(2)}} \frac{\delta z^{(2)}}{\delta b_2}$$

$$\frac{\delta z^{(2)}}{\delta b_2} = \begin{cases} 1 & z^{(2)} > 0 \\ 0 & z^{(2)} \leq 0 \end{cases}$$

From question 4d, we know that $\frac{\delta L}{\delta \hat{y}} \frac{\delta \hat{y}}{\delta z^{(2)}} = (\hat{y} - y)(W_3)^T$

$$\frac{\delta L}{\delta b_2} = \begin{cases} (\hat{y} - y)(W_3)^T & z^{(2)} > 0 \\ 0 & z^{(2)} \leq 0 \end{cases}$$

5. You can use partial derivatives to optimize a neural network by only optimizing the partial derivatives that change with each epoch. At each layer, a new function is evaluated based on the parameters of the previous layer. Instead of taking the gradient of the entire function at each layer in the network, you can simply take the gradient of the partial derivative terms that are a function of the parameters of the layers within the network. In other words, partial derivatives and the chain rule allow us to generate a relationship between the weights of a layer and the cost function. This is important for the gradient descent process, and without it our networks

computational complexity would increase. Overall, the partial derivatives allow us to describe a relationship between the error and the weights, so we can then optimize the weights to end up with the least amount of error.

6.

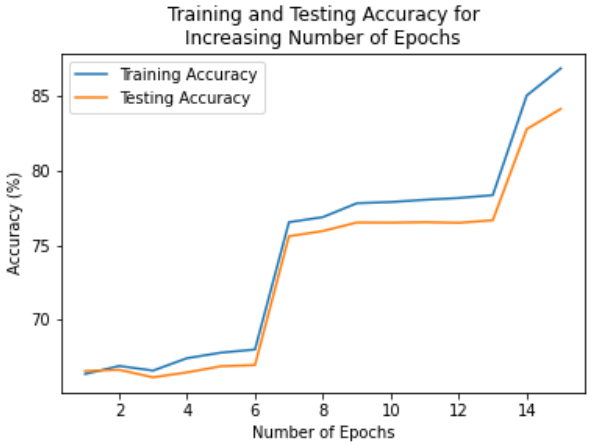

Figure 3: Training and testing accuracy for the MNIST dataset for increasing number of epochs. In general, the accuracy on both the training and test sets increases as the number of epochs is increased. There is a large accuracy improvement from epochs 6 to 7 and 13 to 14, but the accuracy steadily increases for increases in epoch count outside of these two sections.

Part 3: Stack different models

Best Model:

- Base Estimators: Random Forest with n_estimators = 10 and RBF Kernel SVM
- Final Estimator: Logistic Regression

Best Model F1 Score: 0.9859

Initially, I tried a stacked classifier with base estimators of Random Forest (n_estimators = 10) and a Linear SVC, which performed well with an F1 Score = 0.9589. So I thought that this combination of base estimators performed pretty well, but wanted to try the SVM with different kernels. Next, I tried using the normal SVC with linear, polynomial, and RBF kernels which yielded the best stacked classifier as outlined above. I also tried using a stacked model with KNN and Random Forest base estimators and a final estimator of Logistic Regression, which yielded an F1 Score of 0.9445. Given these trials, I ended up reporting the model shown above, which yielded a great F1 Score of 0.9859. The problem asked us to achieve an F1 Score of 0.95, so I figured this would suffice. I will say that it is awesome that this one function does it all for you so that you can efficiently test multiple models. The script for this problem is titled *Q3.py* in my submissions folder.