Richard Albright
ISYE 6740
Homework 5
Fall 2020

1. **SVM.** (45 points)

   (a) (5 points) Explain why can we set the margin $c = 1$ to derive the SVM formulation?

   $$\max_{w,b} \frac{2c}{||w||} \quad s.t.\ \ y^i(w^T x^i + b) \geq c, \forall i$$

   where the size of c is just scales the values of w and b. Therefore we can just set c=1. We can also drop the 2 to simplify the calculation and it becomes

   $$\max_{w,b} \frac{1}{||w||} \quad s.t.\ \ y^i(w^T x^i + b) \geq 1, \forall i$$

   (b) (10 points) Using Lagrangian dual formulation, show that the weight vector can be represented as

   $$w = \sum_{i=1}^{n} \alpha_i y_i x_i.$$

   where $\alpha_i \geq 0$ are the dual variables. What does this imply in terms of how to relate data to w? The SVM solution can be expressed as the following convex optimization problem.

   $$min_{w,b}\frac{1}{2}w^T w \quad s.t.\ \ 1 - y^i(w^T x^i + b) \leq 0, \forall i$$

   $$L(w, a, b) = \frac{1}{2}w^T w + \sum_{i=1}^{m} a_i(1 - y^i(w^T x^i + b))$$

   $$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m} a_i y_i x_i = 0$$

   $$w = \sum_{i=1}^{n} a_i y_i x_i$$

   This implies that the weight vector w linear combination of its datapoints and class labels.

   (c) (10 points) Explain why only the data points on the "margin" will contribute to the sum above, i.e., playing a role in defining w.

   $$min_{w,b}\frac{1}{2}w^T w \quad s.t.\ \ 1 - y^i(w^T x^i + b) \leq 0, \forall i$$

The KKT condition is

$$a_i g_i(w) = 0$$

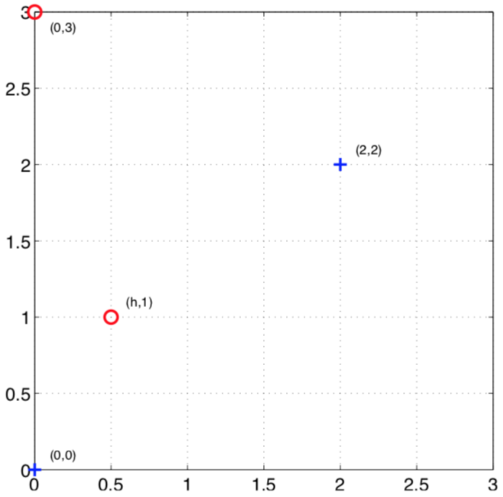
$$a_i(1 - y^i(w^T x^i + b)) = 0$$

where

$$a_i = 0: \quad 1 - y^i(w^T x^i + b)) < 0$$
$$a_i > 0: \quad 1 - y^i(w^T x^i + b)) <= 0$$

only a_i points that > 0 lie on the margin and are the support vectors.

(d) (20 points) Suppose we only have four training examples in two dimensions as shown in Fig. The positive samples at $x_1 = (0, 0)$, $x_2 = (2, 2)$ and negative samples at $x_3 = (h, 1)$ and $x_4 = (0, 3)$.



i. (10 points) For what range of parameter $h > 0$, the training points are still linearly separable?

The training points are linearly separable for the range of $h > 0$ and $h < 1$, and at $h > 6$.

ii. (10 points) Does the orientation of the maximum margin decision boundary change as h changes, when the points are separable?

The orientation of the maximum decision boundary does not change as h changes when $h > 0$ and $h < 1$. The decision boundary in this case has a positive slope and just shifts along the x axis as h changes. The orientation of the maximum decision boundary does change when $h > 6$. The slope is negative and gets more flat as h increases.

2. **Multi-class classification for MNIST data set, comparison.** (55 points)

This question is to compare different classifiers and their performance for multi-class classifications on the complete MNIST dataset at `http://yann.lecun.com/exdb/mnist/`. You can find the data file mnist_10digits.mat in the homework folder. The MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. We will compare **KNN, logistic**

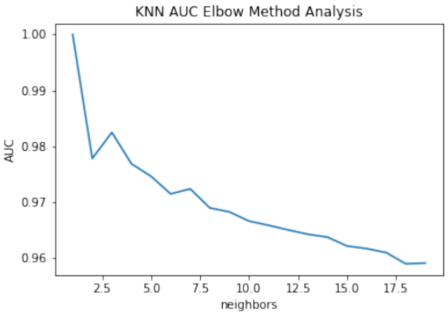
regression, SVM, kernel SVM, and neural networks. We suggest to use Scikit-learn, which is a commonly-used and powerful Python library with various machine learning tools. But you can also use other similar libraries in other programming languages of your choice to perform the tasks. Below are some tips.

- We suggest you to "standardize" the features before training the classifiers, by dividing the values of the features by 255 (thus map the range of the features from [0, 255] to [0, 1]).
- You may adjust the number of neighbors K used in KNN to have a reasonable result (you may use cross validation but it is not required; any reasonable tuning to get good result is acceptable).
- You may use a neural networks function sklearn.neural_network with hidden_layer_sizes = (20, 10).
- For kernel SVM, you may use radial basis function kernel, and a heuristic called "median trick": choose the parameter of the kernel $K(x, x') = \exp\{-\|x - x'\|^2/(2\sigma^2)\}$. Choose the bandwidth as $\sigma = \sqrt{M/2}$ where $M =$ the median of $\{\|x^i - x^j\|^2, 1 \leq i, j \leq m', i \neq j\}$ for pairs of training samples. Here you can randomly choose $m' = 1000$ samples from training data to use for the "median trick"[1].
- For KNN and SVM, you can randomly downsample the training data to size $m = 5000$, to improve computation efficiency.

Train the classifiers on training dataset and evaluate on the test dataset.

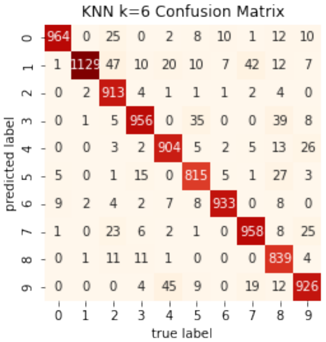
(a) (50 points) Report confusion matrix, precision, recall, and F-1 score for each of the classifiers. For precision, recall, and F-1 score of each classifier, we will need to report these for each of the digits. So you can create a table for this. For this question, each of the 5 classifier, **KNN, logistic regression, SVM, kernel SVM, and neural networks**, accounts for 10 points.

An area under the curve analysis was performed to find the optimal k value for knn. Based on the chart below, I chose $k = 6$.



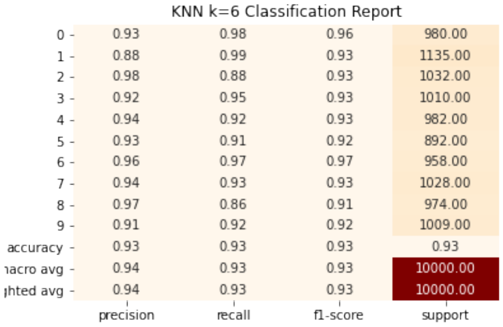
[1] Garreau, Damien, Wittawat Jitkrittum, and Motonobu Kanagawa. ”Large sample analysis of the median heuristic.” arXiv preprint arXiv:1707.07269 (2017).

Using the tips provided for question 2, below are the results for the classifiers KNN, Logistic Regression, Linear SVM, Kernel SVM, and Neural Network (Please excuse the word cutoffs for macro avg and weighted avg, I couldn't figure out how to fix it).



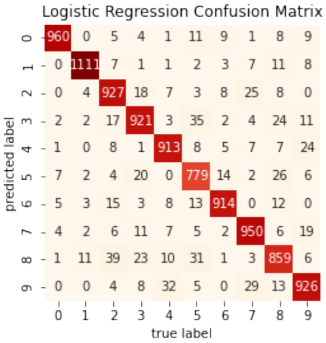
KNN k=6 Confusion Matrix

	0	1	2	3	4	5	6	7	8	9
0	964	0	25	0	2	8	10	1	12	10
1	1	1129	47	10	20	10	7	42	12	7
2	0	2	913	4	1	1	1	2	4	0
3	0	1	5	956	0	35	0	0	39	8
4	0	0	3	2	904	5	2	5	13	26
5	5	0	1	15	0	815	5	1	27	3
6	9	2	4	2	7	8	933	0	8	0
7	1	0	23	6	2	1	0	958	8	25
8	0	1	11	11	1	0	0	0	839	4
9	0	0	0	4	45	9	0	19	12	926



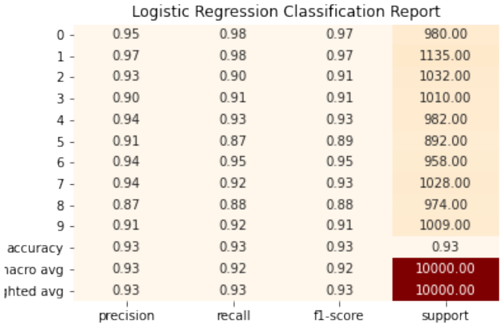
KNN k=6 Classification Report

	precision	recall	f1-score	support
0	0.93	0.98	0.96	980.00
1	0.88	0.99	0.93	1135.00
2	0.98	0.88	0.93	1032.00
3	0.92	0.95	0.93	1010.00
4	0.94	0.92	0.93	982.00
5	0.93	0.91	0.92	892.00
6	0.96	0.97	0.97	958.00
7	0.94	0.93	0.93	1028.00
8	0.97	0.86	0.91	974.00
9	0.91	0.92	0.92	1009.00
accuracy	0.93	0.93	0.93	0.93
ıacro avg	0.94	0.93	0.93	10000.00
ȷhted avg	0.94	0.93	0.93	10000.00



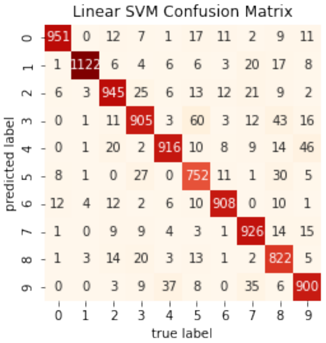
Logistic Regression Confusion Matrix

	0	1	2	3	4	5	6	7	8	9
0	960	0	5	4	1	11	9	1	8	9
1	0	1111	7	1	1	2	3	7	11	8
2	0	4	927	18	7	3	8	25	8	0
3	2	2	17	921	3	35	2	4	24	11
4	1	0	8	1	913	8	5	7	7	24
5	7	2	4	20	0	779	14	2	26	6
6	5	3	15	3	8	13	914	0	12	0
7	4	2	6	11	7	5	2	950	6	19
8	1	11	39	23	10	31	1	3	859	6
9	0	0	4	8	32	5	0	29	13	926



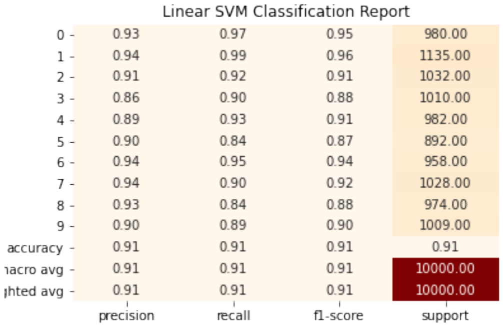
Logistic Regression Classification Report

	precision	recall	f1-score	support
0	0.95	0.98	0.97	980.00
1	0.97	0.98	0.97	1135.00
2	0.93	0.90	0.91	1032.00
3	0.90	0.91	0.91	1010.00
4	0.94	0.93	0.93	982.00
5	0.91	0.87	0.89	892.00
6	0.94	0.95	0.95	958.00
7	0.94	0.92	0.93	1028.00
8	0.87	0.88	0.88	974.00
9	0.91	0.92	0.91	1009.00
accuracy	0.93	0.93	0.93	0.93
ıacro avg	0.93	0.92	0.92	10000.00
ȷhted avg	0.93	0.93	0.93	10000.00



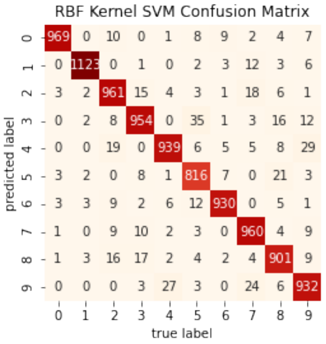
Linear SVM Confusion Matrix

	0	1	2	3	4	5	6	7	8	9
0	951	0	12	7	1	17	11	2	9	11
1	1	1122	6	4	6	6	3	20	17	8
2	6	3	945	25	6	13	12	21	9	2
3	0	1	11	905	3	60	3	12	43	16
4	0	1	20	2	916	10	8	9	14	46
5	8	1	0	27	0	752	11	1	30	5
6	12	4	12	2	6	10	908	0	10	1
7	1	0	9	9	4	3	1	926	14	15
8	1	3	14	20	3	13	1	2	822	5
9	0	0	3	9	37	8	0	35	6	900



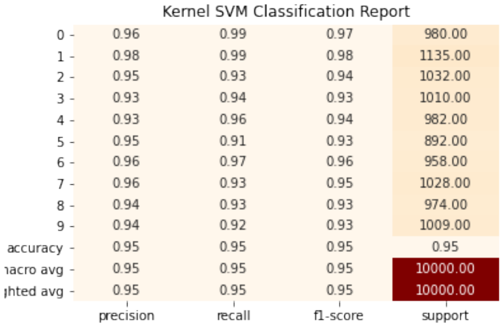
Linear SVM Classification Report

	precision	recall	f1-score	support
0	0.93	0.97	0.95	980.00
1	0.94	0.99	0.96	1135.00
2	0.91	0.92	0.91	1032.00
3	0.86	0.90	0.88	1010.00
4	0.89	0.93	0.91	982.00
5	0.90	0.84	0.87	892.00
6	0.94	0.95	0.94	958.00
7	0.94	0.90	0.92	1028.00
8	0.93	0.84	0.88	974.00
9	0.90	0.89	0.90	1009.00
accuracy	0.91	0.91	0.91	0.91
ıacro avg	0.91	0.91	0.91	10000.00
ȷhted avg	0.91	0.91	0.91	10000.00



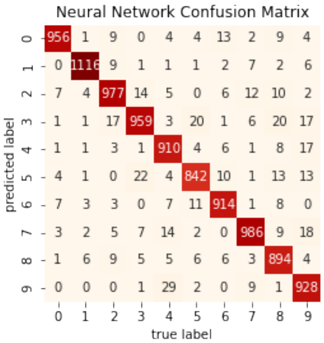
RBF Kernel SVM Confusion Matrix

	0	1	2	3	4	5	6	7	8	9
0	969	0	10	0	1	8	9	2	4	7
1	0	1123	0	1	0	2	3	12	3	6
2	3	2	961	15	4	3	1	18	6	1
3	0	2	8	954	0	35	1	3	16	12
4	0	0	19	0	939	6	5	5	8	29
5	3	2	0	8	1	816	7	0	21	3
6	3	3	9	2	6	12	930	0	5	1
7	1	0	9	10	2	3	0	960	4	9
8	1	3	16	17	2	4	2	4	901	9
9	0	0	0	3	27	3	0	24	6	932



Kernel SVM Classification Report

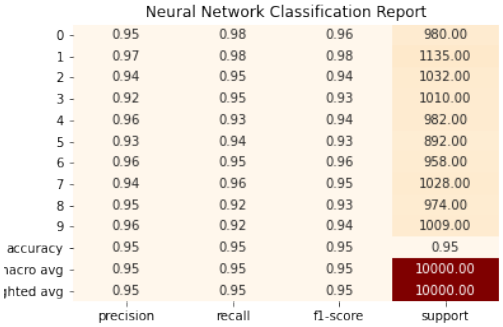
	precision	recall	f1-score	support
0	0.96	0.99	0.97	980.00
1	0.98	0.99	0.98	1135.00
2	0.95	0.93	0.94	1032.00
3	0.93	0.94	0.93	1010.00
4	0.93	0.96	0.94	982.00
5	0.95	0.91	0.93	892.00
6	0.96	0.97	0.96	958.00
7	0.96	0.93	0.95	1028.00
8	0.94	0.93	0.93	974.00
9	0.94	0.92	0.93	1009.00
accuracy	0.95	0.95	0.95	0.95
ıacro avg	0.95	0.95	0.95	10000.00
ȷhted avg	0.95	0.95	0.95	10000.00



Neural Network Confusion Matrix

predicted label
true label

	0	1	2	3	4	5	6	7	8	9
0	956	1	9	0	4	4	13	2	9	4
1	0	1116	9	1	1	1	2	7	2	6
2	7	4	977	14	5	0	6	12	10	2
3	1	1	17	959	3	20	1	6	20	17
4	1	1	3	1	910	4	6	1	8	17
5	4	1	0	22	4	842	10	1	13	13
6	7	3	3	0	7	11	914	1	8	0
7	3	2	5	7	14	2	0	986	9	18
8	1	6	9	5	5	6	6	3	894	4
9	0	0	0	1	29	2	0	9	1	928



Neural Network Classification Report

	precision	recall	f1-score	support
0	0.95	0.98	0.96	980.00
1	0.97	0.98	0.98	1135.00
2	0.94	0.95	0.94	1032.00
3	0.92	0.95	0.93	1010.00
4	0.96	0.93	0.94	982.00
5	0.93	0.94	0.93	892.00
6	0.96	0.95	0.96	958.00
7	0.94	0.96	0.95	1028.00
8	0.95	0.92	0.93	974.00
9	0.96	0.92	0.94	1009.00
accuracy	0.95	0.95	0.95	0.95
ıacro avg	0.95	0.95	0.95	10000.00
ȷhted avg	0.95	0.95	0.95	10000.00

(b) (5 points) Comment on the performance of the classifier and give your explanation why some of them perform better than the others.

All classifiers resulted in accuracies of over 90%. Linear SVM performed the worst, while RBF Kernel SVM and Neural Network performed the best. KNN had the hardest time classifying 1. My guess is that is harder to pick the nearest neighbors accurately beause there is less variation on the horizontal axis, leaving mostly the vertical axis to determine the nearest neighbors. Logistic regression had the hardest time classifying 8. Since the sigmoid function's 1st derivative is symmetric, it might have a harder time determinine 8 from other numbers since it is mostly symmetric along its vertical axis. Linear SVM performs generally worse, likely because the images are not strictly linearly separable. RBF Kernel SVM shows a marked improvement over Linear SVM. The gamma parameter was adjusted using the median trick. The classification rates for 3 and 4 were the lowest for the RBF Kernel SVM, while still maintaining an accuracy rate of over 93%. The number 3 was most oftenly misclassified as 8, while 4 was most oftenly misclassified as 9. Nueral Network classification had the worst performance classifying both 3 and 5. They both were most often misclassified as each other. I would have expected Neural Network to perform the best, because this algorithm is best candidate for classifying images.

3. **Neural networks.** (Bonus: 10 points)

(a) (2 points) Consider a neural networks for a binary classification using sigmoid function for each unit. If the network has no hidden layer, explain why the model is equivalent to logistic regression.

A neural network consists of an input layer, a hidden layer, and an output layer. The input layer consists of the independent variables + the bias term. The hidden layer consists of nonlinear transformation functions that bound the continuous input variables to a value between 0 and 1. Each nonlinear function is given a weight between 0 and 1 where the sum of weights = 1. The output layer consists of a sigmoid function to convert the output of the last layer into a probability of belonging to a given class. By removing the hidden layer, we end up taking the input layer as inputs into the sigmoid function, which is the logistic regression model.

(b) (8 points) Consider a simple two-layer network in the lecture slides. Given m training data (x^i, y^i), $i = 1, \ldots, m$, the cost function used to training the neural networks

$$\ell(w, \alpha, \beta) = \sum_{i=1}^{m} (y^i - \sigma(w^T z^i))^2$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function, z^i is a two-dimensional vector such that $z_1^i = \sigma(\alpha^T x^i)$, and $z_2^i = \sigma(\beta^T x^i)$. Show the that the gradient is given by

$$\frac{\partial \ell(w,\alpha,\beta)}{\partial w} = -\sum_{i=1}^m 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))z^i,$$

where $u^i = w^T z^i$. Also find the gradient of $\ell(w,\alpha,\beta)$ with respect to α and β and write down their expression.

The gradient of $\ell(w,\alpha,\beta)$ with respect to w

$$\frac{\partial \ell(w,\alpha,\beta)}{\partial w} = \frac{\partial \sum_{i=1}^m (y^i - \sigma(w^T z^i))^2}{\partial w}$$

$$= \sum_{i=1}^m \frac{\partial((y^i - \sigma(w^T z^i))^2)}{\partial w}$$

$$= \sum_{i=1}^m \frac{\partial((y^i - \sigma(w^T z^i))^2)}{\partial(y^i - \sigma(w^T z^i))} \frac{\partial(y^i - \sigma(w^T z^i))}{\partial w}$$

$$= \sum_{i=1}^m 2(y^i - \sigma(w^T z^i))(\frac{\partial y^i}{\partial w} - \frac{\partial \sigma(w^T z^i)}{\partial w})$$

$$= -\sum_{i=1}^m 2(y^i - \sigma(w^T z^i))\frac{\partial \sigma(w^T z^i)}{\partial w^T z^i}\frac{\partial w^T z^i}{\partial w}$$

$$= -\sum_{i=1}^m 2(y^i - \sigma(w^T z^i))\frac{\partial \sigma(w^T z^i)}{\partial w^T z^i} z^i$$

$$= -\sum_{i=1}^m 2(y^i - \sigma(u^i))\frac{\partial \sigma(u^i)}{\partial u^i} z^i$$

$$= -\sum_{i=1}^m 2(y^i - \sigma(u^i))\frac{\partial(1/(1+e^{-u^i}))}{\partial u^i} z^i$$

$$= -\sum_{i=1}^m 2(y^i - \sigma(u^i))z^i \frac{(\partial 1/\partial u^i)(1+e^{-u^i}) - \partial(1+e^{-u^i})/\partial u^i}{(1+e^{-u^i})^2}$$

$$= -\sum_{i=1}^m 2(y^i - \sigma(u^i))z^i \frac{e^{-u^i}}{(1+e^{-u^i})^2}$$

$$= -\sum_{i=1}^m 2(y^i - \sigma(u^i))\frac{1}{1+e^{-u^i}}(1 - \frac{1}{1+e^{-u^i}})z^i$$

$$= -\sum_{i=1}^m 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))z^i$$

The gradient of $\ell(w, \alpha, \beta)$ with respect to α.

Using the chain rule.

$$\frac{\partial \ell(w,\alpha,\beta)}{\partial \alpha} = \frac{\partial \ell(w,\alpha,\beta)}{\partial z_1^i}\frac{\partial z_1^i}{\partial \alpha}$$

$$\frac{\partial \ell(w,\alpha,\beta)}{\partial z_1^i} = \sum_{i=1}^{m} \frac{\partial((y^i - \sigma(w^T z_1^i))^2)}{\partial z_1^i}$$

$$= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))w_1$$

Therefore

$$\frac{\partial \ell(w,\alpha,\beta)}{\partial \alpha} = \frac{\partial \ell(w,\alpha,\beta)}{\partial z_1^i}\frac{\partial z_1^i}{\partial \alpha}$$

$$= \sum_{i=1}^{m} \frac{\partial((y^i - \sigma(w^T z_1^i))^2)}{\partial z_1^i}\frac{\partial(\sigma(\alpha^T x^i))}{\partial \alpha}$$

$$= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))w_1 \frac{\partial(\sigma(\alpha^T x^i))}{\partial \alpha}$$

$$= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))w_1 \frac{\partial(1/(1+e^{-\alpha^T x^i}))}{\partial \alpha}$$

$$= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))w_1 \frac{(\partial 1/\partial \alpha)(1+e^{-\alpha^T x^i}) - \partial(1+e^{-\alpha^T x^i})/\partial \alpha}{(1+e^{-\alpha^T x^i})^2}$$

$$= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))w_1 \frac{e^{-\alpha^T x^i}}{(1+e^{-\alpha^T x^i})^2} x_i$$

$$= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))w_1 \frac{1}{1+e^{-\alpha^T x^i}}(1 - \frac{1}{1+e^{-\alpha^T x^i}}) x_i$$

$$= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))w_1 \sigma(\alpha^T x^i)(1-\sigma(\alpha^T x^i)) x_i$$

The gradient of $\ell(w, \alpha, \beta)$ with respect to β.

Performing a similar calculation for β in the above derivation we get:

$$\frac{\partial \ell(w,\alpha,\beta)}{\partial \beta} = -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))w_1 \sigma(\beta^T x^i)(1-\sigma(\beta^T x^i)) x_i$$