Problem 1

$$\mathbf{L}(y_i|\mathbf{x}_i, \mathbf{w}) = y_i \log(\sigma(\mathbf{w}^T\mathbf{x}_i)) + (1 - y_i)\log(\sigma(-\mathbf{w}^T\mathbf{x}_i))$$

Find the gradient of $\mathbf{L}(y_i|\mathbf{x}_i, \mathbf{w})$ with respect to w_j.

Assign $\mathbf{w}^T\mathbf{x}_i = a$:

$$\mathbf{L}(y_i|\mathbf{x}_i, \mathbf{w}) = y_i \log(\sigma(a)) + (1 - y_i)\log(\sigma(-a))$$
where $\sigma(a) = \frac{1}{1+exp(-a)}$

Apply chain rule (first find the derivative of $\mathbf{L}$ with respect to a, then find the derivative of a with respect to w_j) :

$$\frac{\partial \mathbf{L}}{\partial w_j} = \frac{\partial \mathbf{L}}{\partial a}\frac{\partial a}{\partial w_j}$$

Derivative of $\mathbf{L}$ w.r.t. a :

$$\frac{\partial \mathbf{L}}{\partial a} = y_i \frac{1}{\sigma(a)} \frac{\mathrm{e}^a}{(\mathrm{e}^a + 1)^2} + (1 - y_i)\frac{1}{\sigma(-a)}\frac{-\mathrm{e}^a}{(\mathrm{e}^a + 1)^2}$$
$$= y_i \frac{1}{e^a + 1} + (1 - y_i)\frac{-e^a}{e^a + 1} = y_i - \frac{e^a}{e^a + 1} = y_i - \sigma(a)$$
$$\text{substitude } a = \mathbf{w}^T\mathbf{x}_i \longrightarrow \frac{\partial \mathbf{L}}{\partial a} = y_i - \sigma(\mathbf{w}^T\mathbf{x}_i)$$

Derivative of $a = \mathbf{w}^T\mathbf{x}_i$ w.r.t. w_j :

$$\frac{\partial a}{\partial w_j} = x_{ij}$$

All in all,

$$\frac{\partial \mathbf{L}}{\partial w_j} = (y_i - \sigma(\mathbf{w}^T\mathbf{x}_i))x_{ij}$$

Problem 2

List of η's used for this problem is $[0.00001, 0.0001, 0.001, 0.01, 0.1]$.

Error Rates for Logistic Regression: $\eta = 0.00001$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.01500	0.12000	0.15500	0.18500	0.10500	0.14500	0.22500	0.17500	0.15000	0.16500	0.14400	0.05342

Error Rates for Logistic Regression: $\eta = 0.0001$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.01000	0.03000	0.02500	0.01500	0.04500	0.03500	0.02500	0.02000	0.03500	0.02000	0.02600	0.00995

Error Rates for Logistic Regression: $\eta = 0.001$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.05500	0.06500	0.03500	0.04000	0.05500	0.04500	0.03000	0.06000	0.06500	0.03500	0.04850	0.01246

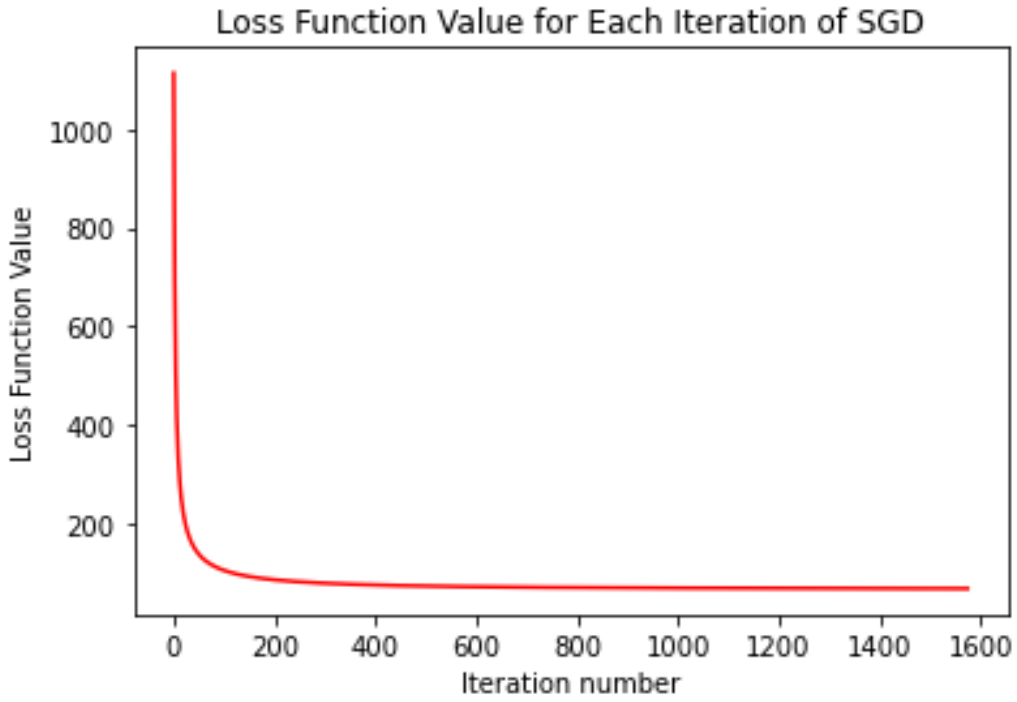
Error Rates for Logistic Regression: $\eta = 0.01$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.04500	0.06000	0.02000	0.04000	0.00500	0.09000	0.04000	0.02500	0.04000	0.02000	0.03850	0.02270

Error Rates for Logistic Regression: $\eta = 0.1$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.01500	0.04500	0.03500	0.06500	0.02500	0.03500	0.02500	0.03000	0.04500	0.02500	0.03450	0.01350

Among the η's in the list, $\eta = 0.0001$ is the most optimal as it gave the least mean of the error rates. The table below shows the logistic regression with SGD($\eta = 0.0001$). As it is seen, the mean of the folds is the least, smaller than all the means found using several η's. This shows that training was successful.

Error Rates for Logistic Regression on Test Data, using best η: $\eta = 0.0001$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.00000	0.00000	0.02500	0.00000	0.02500	0.02500	0.02500	0.02500	0.07500	0.00000	0.02000	0.02179

Plot below shows the loss function value for each iteration of SGD. After some iterations, the loss function value doesn't change because the convergence has been reached. After a certain amount of iterations, the change between consecutive values stayed below the pre-specified threshold.



If we used FGD instead of SGD, we would have to wait for a long time for convergence(assuming it'll not converge quickly). FGD requires computations on the entire dataset for each iteration. Hence, it results in smoother updates. That is computationally very expensive if one uses a huge dataset. FGD would result in convergence in less iterations due to its ability to handle the whole dataset at one time.

Problem 3

$$f(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{n}\max\Big(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\Big)$$

Find the gradient of $f(\mathbf{w})$ with respect to w_j.

When the hinge loss is 0:

$$1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) = 0$$

$$\max\Big(0, 0\Big) = 0$$

$$\frac{\partial f(\mathbf{w})}{\partial w_j} = \frac{\partial}{\partial w_j}\frac{1}{2}||\mathbf{w}||_2^2 = w_j$$

When the hinge loss is <u>not</u> 0:

$$\frac{\partial f(\mathbf{w})}{\partial w_j} = \frac{\partial}{\partial w_j}\frac{1}{2}||\mathbf{w}||_2^2 + \frac{\partial}{\partial w_j}C\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))$$

$$\frac{\partial f(\mathbf{w})}{\partial w_j} = w_j - C\sum_{i=1}^{n} y_i x_{ij}$$

Problem 4

An output example is the following:

<u>for eta = 1e-05 c = 1</u>
Error Rates for SVM: ['0.02500', '0.07500', '0.09500', '0.06500', '0.04000', '0.03500', '0.07500', '0.06000', '0.07500', '0.06000', **'Mean: 0.0605 Std: 0.02043'**]

<u>for eta = 1e-05 c = 10</u>
Error Rates for SVM: ['0.09500', '0.03000', '0.03500', '0.05000', '0.02500', '0.03500', '0.06500', '0.03500', '0.07000', '0.03500', **'Mean: 0.0475 Std: 0.02124'**]

<u>for eta = 1e-05 c = 100</u>
Error Rates for SVM: ['0.02500', '0.04500', '0.08500', '0.05500', '0.05000', '0.07000', '0.04000', '0.03500', '0.07000', '0.06000', **'Mean: 0.0535 Std: 0.01733'**]

<u>for eta = 0.0001 c = 1</u>
Error Rates for SVM: ['0.01000', '0.02500', '0.02000', '0.01500', '0.01500', '0.01500', '0.01000', '0.01500', '0.00500', '0.00500', **'Mean: 0.0135 Std: 0.00594'**]

<u>for eta = 0.0001 c = 10</u>
Error Rates for SVM: ['0.01000', '0.01500', '0.02000', '0.01500', '0.01500', '0.02500', '0.01000', '0.00000', '0.01000', '0.01500', **'Mean: 0.0135 Std: 0.00634'**]

<u>for eta = 0.0001 c = 100</u>
Error Rates for SVM: ['0.00000', '0.00500', '0.02500', '0.02500', '0.02000', '0.01500', '0.01000', '0.01000', '0.01500', '0.01000', **'Mean: 0.0135 Std: 0.00776'**]

<u>for eta = 0.001 c = 1</u>
Error Rates for SVM: ['0.02000', '0.01000', '0.01500', '0.01500', '0.01000', '0.01000', '0.01500', '0.01000', '0.02000', '0.00500', **'Mean: 0.013 Std: 0.00458'**]

<u>for eta = 0.001 c = 10</u>
Error Rates for SVM: ['0.03000', '0.01000', '0.01000', '0.00000', '0.02500', '0.01000', '0.01000', '0.01500', '0.03000', '0.01000', **'Mean: 0.015 Std: 0.00949'**]

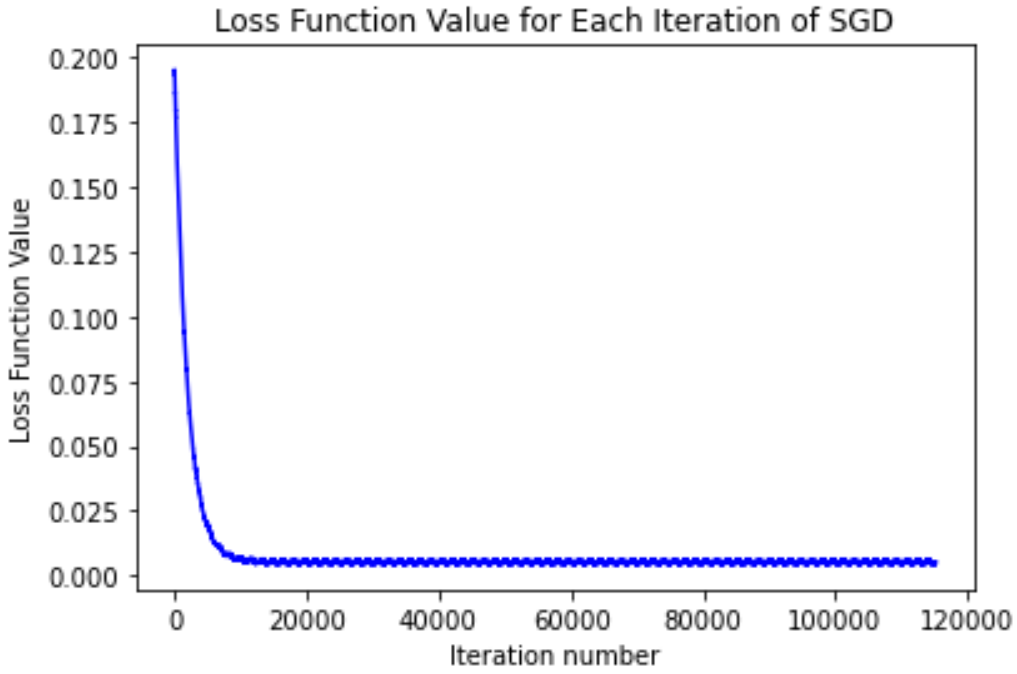
<u>for eta = 0.001 c = 100</u>
Error Rates for SVM: ['0.01500', '0.00500', '0.02000', '0.00500', '0.02000', '0.02000', '0.03500', '0.00000', '0.00000', '0.01500', **'Mean: 0.0135 Std: 0.0105'**]

<u>Best eta and c for SVM: eta = 0.001 c = 1</u>
Error Rates with Best eta and c ['0.02500', '0.00000', '0.00000', '0.00000', '0.00000', '0.00000', '0.00000', '0.00000', '0.02500', '0.00000', **'Mean: 0.005 Std: 0.01'**]

Among the η's and c's in the lists, $(\eta = 0.0001, c = 1)$ is the most optimal as it gave the least mean with the least standard deviation of the error rates. The table below shows the SVM with SGD$(\eta = 0.0001, c = 1)$. As it is seen, the mean of the folds is the least, smaller than all the means found using several η's and c's. This shows that training was successful.

Error Rates for SVM on Test Data, using best (η, c): $\eta = 0.0001$, $c = 1$											
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
0.02500	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.02500	0.00000	0.00500	0.01000

Plot below shows the loss function value for each iteration of SGD. After some iterations, the loss function value doesn't change because the convergence has been reached. After a certain amount of iterations, the change between consecutive values stayed below the pre-specified threshold.



If we used FGD instead of SGD, we would have to wait for a long time for convergence(assuming it'll not converge quickly). FGD requires computations on the entire dataset for each iteration. Hence, it results in smoother updates. That is computationally very expensive if one uses a huge dataset. FGD would result in convergence in less iterations due to its ability to handle the whole dataset at one time.