Machine Learning, 2024 Spring
Assignment 3

Notice

Plagiarizer will get 0 points.
LaTeXis highly recommended. Otherwise you should write as legibly as possible.

Problem 1 For logistic regression, show that

$$\nabla E_{\text{in}}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n\mathbf{x}_n}{1+e^{y_n\mathbf{w}^{\mathrm{T}}\mathbf{x}_n}} = \frac{1}{N}\sum_{n=1}^{N}-y_n\mathbf{x}_n\theta\left(-y_n\mathbf{w}^{\mathrm{T}}\mathbf{x}_n\right) \tag{1}$$

Argue that a 'misclassified' example contributes more to the gradient than a correctly classified one.
Solution
1. Let $\theta(x) = \dfrac{e^x}{1+e^x} = \dfrac{1}{1+e^{-x}}$ be the sigmoid function.
So we have $p(y_n|\mathbf{x}_n;\mathbf{w}) = \theta(y_n\mathbf{w}^T\mathbf{x}_n)$, and the maximum likelihood estimation is

$$\max\prod_{n=1}^{N}p(y_n|\mathbf{x}_n;\mathbf{w})$$

For a convinient calculation, we take the log of the likelihood function, and then take the negative of it, and take mean of each term, so we have what we want to minimize:

$$\begin{aligned}
E_{\text{in}}(\mathbf{w}) &= -\frac{1}{N}\log[\prod_{n=1}^{N}p(y_n|\mathbf{x}_n;\mathbf{w})] \\
&= -\frac{1}{N}\sum_{n=1}^{N}\log\theta(y_n\mathbf{w}^T\mathbf{x}_n) \\
&= -\frac{1}{N}\sum_{n=1}^{N}\log\frac{1}{1+e^{-y_n\mathbf{w}^T\mathbf{x}_n}} \\
&= -\frac{1}{N}\sum_{n=1}^{N}[-\log(1+e^{-y_n\mathbf{w}^T\mathbf{x}_n})] \\
&= \frac{1}{N}\sum_{n=1}^{N}\log(1+e^{-y_n\mathbf{w}^T\mathbf{x}_n})
\end{aligned}$$

So the gradient of $E_{\text{in}}\left(\mathbf{w}\right)$ is

$$\begin{aligned}
\nabla E_{\text{in}}\left(\mathbf{w}\right) &= \nabla\left[\frac{1}{N}\sum_{n=1}^{N}\log(1+e^{-y_n\mathbf{w}^T\mathbf{x}_n})\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\nabla\log(1+e^{-y_n\mathbf{w}^T\mathbf{x}_n}) \\
&= \frac{1}{N}\sum_{n=1}^{N}\frac{(-y_n\mathbf{x}_n)e^{-y_n\mathbf{w}^T\mathbf{x}_n}}{1+e^{-y_n\mathbf{w}^T\mathbf{x}_n}} \\
&= -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n\mathbf{x}_n}{1+e^{y_n\mathbf{w}^T\mathbf{x}_n}} \\
&= \frac{1}{N}\sum_{n=1}^{N}-y_n\mathbf{x}_n\theta(-y_n\mathbf{w}^T\mathbf{x}_n)
\end{aligned}$$

2. From the property of the sigmoid function, we know that the sigmoid function is monotonic increasing.
So for a correctly classified example, we have $y_n\mathbf{w}^T\mathbf{x}_n \geq 0$, so $\theta(-y_n\mathbf{w}^T\mathbf{x}_n) \in (0, 0.5]$.

And for a misclassified example, we have $y_n\mathbf{w}^T\mathbf{x}_n < 0$, so we have $\theta(-y_n\mathbf{w}^T\mathbf{x}_n) \in (0.5, 1)$.

From the gradient formula $\nabla E_{\text{in}}\left(\mathbf{w}\right)$, the difference between the correctly classified example and the misclassified example is only the sigmoid function.
And we also have the misclassified example's $\theta(-y_n\mathbf{w}^T\mathbf{x}_n)$ is larger than the correctly classified example's $\theta(-y_n\mathbf{w}^T\mathbf{x}_n)$, so the misclassified example contributes more to the gradient than a correctly classified one.

So above all, we have proved that

$$\nabla E_{\text{in}}\left(\mathbf{w}\right) = -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n\mathbf{x}_n}{1+e^{y_n\mathbf{w}^{\mathrm{T}}\mathbf{x}_n}} = \frac{1}{N}\sum_{n=1}^{N}-y_n\mathbf{x}_n\theta\left(-y_n\mathbf{w}^{\mathrm{T}}\mathbf{x}_n\right)$$

And we also proved that the 'misclassified' example contributes more to the gradient than a correctly classified one.

Problem 2 For linear regression, the out-of-sample error is

$$E_{\text{out}}(h) = \mathbb{E}[(h(x) - y)^2]. \tag{2}$$

Show that among all hypotheses, the one that minimizes E_{out} is given by

$$h^*(x) = \mathbb{E}[y|x]. \tag{3}$$

The function h^* can be treated as a deterministic target function, in which case we can write $y = h^*(x) + \epsilon(x)$ where $\epsilon(x)$ is an (input dependent) noise variable. Show that $\epsilon(x)$ has expected value zero.

Solution

Suppose the joint distribution of x and y is $f(x, y)$, the distribution of x is $f_X(x)$, the distribution of y is $f_Y(y)$, and the conditional distribution of y given x is $f_{Y|X}(y|x)$.

Lemma: The law of iterated Expectation

$$\mathbb{E}(Y) = \mathbb{E}_X[\mathbb{E}(Y|X)]$$

proof:

$$\begin{aligned}
\mathbb{E}_X[\mathbb{E}(Y|X)] &= \mathbb{E}_X\left[\int_{-\infty}^{+\infty} y f_{Y|X}(y|x)dy\right] \\
&= \mathbb{E}_X\left[\int_{-\infty}^{+\infty} y\frac{f(x,y)}{f_X(x)}dy\right] \\
&= \int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty} y\frac{f(x,y)}{f_X(x)}dy\right] f_X(x)dx \\
&= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} y f(x,y)dxdy \\
&= \int_{-\infty}^{+\infty} y\left[\int_{-\infty}^{+\infty} f(x,y)dx\right]dy \\
&= \int_{-\infty}^{+\infty} y f_Y(y)dy \\
&= \mathbb{E}(Y)
\end{aligned}$$

1. Since $E_{\text{out}}(h) = \mathbb{E}[(h(x) - y)^2] = \mathbb{E}_{X,Y}[(h(x) - y)^2]$, we have

$$\begin{aligned}
\frac{\partial E_{\text{out}}(h)}{\partial h} &= 2\mathbb{E}_{X,Y}[h(x) - y] \\
&= 2\mathbb{E}_X\left[\mathbb{E}\left[(h(x) - y)|x\right]\right] \qquad \text{(Lemma)} \\
&= 2\mathbb{E}_X\left[\mathbb{E}\left[h(x)|x\right] - \mathbb{E}\left[y|x\right]\right] \\
&= 2\mathbb{E}_X\left[h(x) - \mathbb{E}\left[y|x\right]\right]
\end{aligned}$$

$$\frac{\partial^2 E_{\text{out}}(h)}{\partial h^2} = 2 > 0$$

So $E_{\text{out}}(h)$ is a convex function for h.

For minimizing the $E_{\text{out}}(h)$, we just need to let $\dfrac{\partial E_{\text{out}}(h)}{\partial h} = 2\mathbb{E}_X\left[h(x) - \mathbb{E}\left[y|x\right]\right] = 0$.

If $h(x) - \mathbb{E}\left[y|x\right] = 0$, then $\dfrac{\partial E_{\text{out}}(h)}{\partial h} = 0$.

So we have $h^*(x) = \mathbb{E}[y|x]$ minimizes E_{out}.

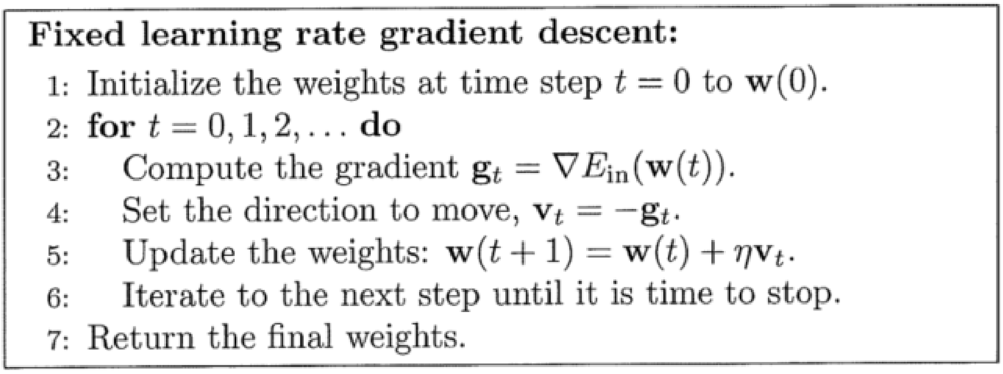
2. Since $y = h^*(x) + \epsilon(x)$, we have

$$\mathbb{E}[\epsilon(x)] = \mathbb{E}[y - h^*(x)] = \mathbb{E}(y) - \mathbb{E}[h^*(x)] = \mathbb{E}[y] - \mathbb{E}\left[\mathbb{E}[y|x]\right] \overset{\text{Lemma}}{=} \mathbb{E}[y] - \mathbb{E}[y] = 0$$

So above all, we have prove that $h^*(x) = \mathbb{E}[y|x]$ and $\epsilon(x)$ has expected value zero.

Problem 3 According to the iterative format provided as follow:

- Use the SUV dataset to implement (using Python or MATLAB) the Gradient Descent method to find the optimal model for logistic regression.
- What is your test error? What are the sizes of the training set and test set, respectively?
- What is your learning rate? How was it chosen? How many steps were iterated in total?
- Present the results of the last 10 steps produced by your algorithm, including the loss, learning rate, the L2 norm of the gradient, and the number of function evaluations and gradient evaluations.



```
Fixed learning rate gradient descent:
Initialize the weights at time step $t = 0$ to $\mathbf{w}(0)$.
for $t = 0, 1, 2, \ldots$ do
    Compute the gradient $\mathbf{g}_t = \nabla E_{\text{in}}(\mathbf{w}(t))$.
    Set the direction to move, $\mathbf{v}_t = -\mathbf{g}_t$.
    Update the weights: $\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \mathbf{v}_t$.
    Iterate to the next step until it is time to stop.
Return the final weights.
```

Dataset reference and description
Dataset and download

Solution

0. Data preperation
After checking the SUV dataset, we could find that each buyer's information has a unique 'User ID', so we can just drop the 'User ID' column.
To use the 'Gender' information, we map 'Male' into 1, and 'Female' into 0.
If we use exponential function during the logistic regression, it may have too big values with large 'Age' and 'EstimatedSalary', so we can seperate the 'Age' and 'EstimatedSalary' into different intervals. And use an one-hot vector to represent the which range the 'Age' and 'EstimatedSalary' belongs to. And the modified encoding for 'Age' and 'Estimated salary' are as followed in Figure 1 and Figure 2. The catagoies of 'Age' and 'EstimatedSalary' are chosen by the distribution graph of the original data, and each catagory is represented by a one-hot vector.

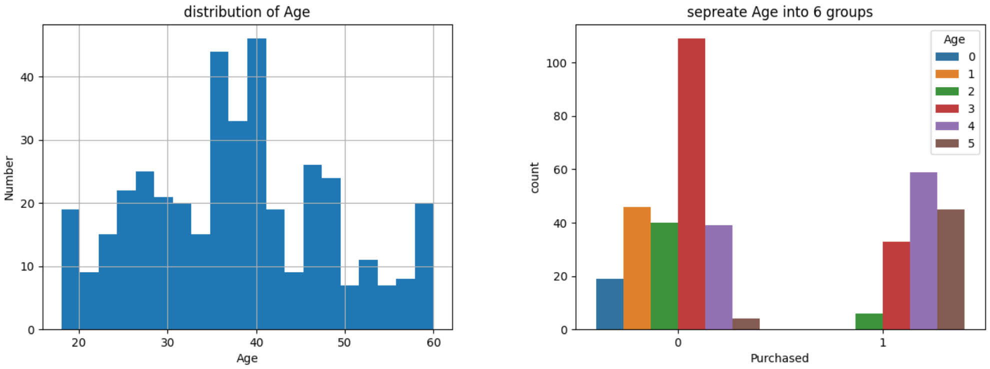


Figure 1: Analysis of Age

To suit better for logistic regression, the 'Purchased' column is also modified into '-1' for unpurchased and '1' for purchased.

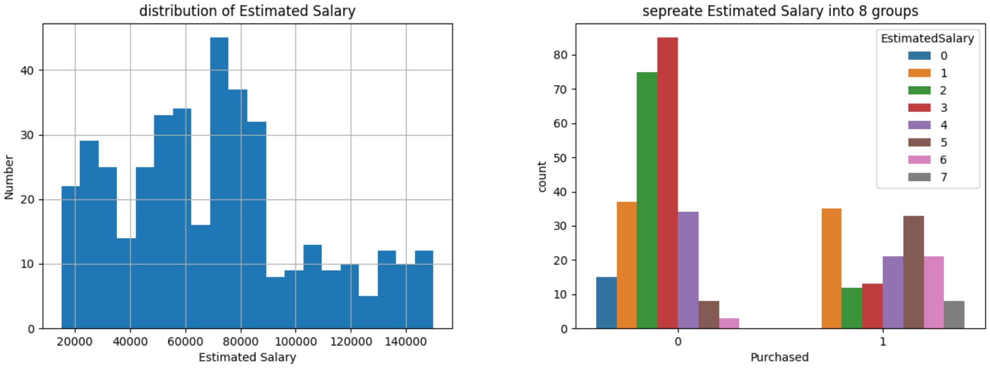


Figure 2: Analysis of Estimated Salary

At last, a bias term ‘1 is add for each input fetures. And before sending the data into the logistic regression, the input features are normalized.

Then the ‘Gender’, ‘Age’, ‘EstimatedSalary’ can be used as the input features, and the ‘Purchased’ can be used as the output label.
So this is a binary classification problem.
And we can use the logistic regression to solve this problem.

1. Since we are applying the logistic regression, so the testing error is:

$$E(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\log(1 + e^{-y_n\mathbf{w}^T\mathbf{x}_n})$$

which have been proved in Problem 1. And N is the size of the testing set's size. y_n are the labels of testing set, $\mathbf{x}_n$ are the input data for the testing set.
The variation curve for the testing error is shown in Figure 5.

The final test correct number $107/120 \approx 0.89$.
The final test error number $23/120 \approx 0.11$.

And the size of training set is 70 persent, which is totally 280 samples. And the testing set is 30 persent, which is totally 120 samples.
And before seperating the data, we random shuffled the total 400 data with a fixed seed in order to make sure the randomness, but could be stable reproduction.

2. We have known that the gradient of logistic regression from Problem 1 is:

$$\nabla E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} -y_n\mathbf{x}_n\theta(-y_n\mathbf{w}^T\mathbf{x}_n)$$

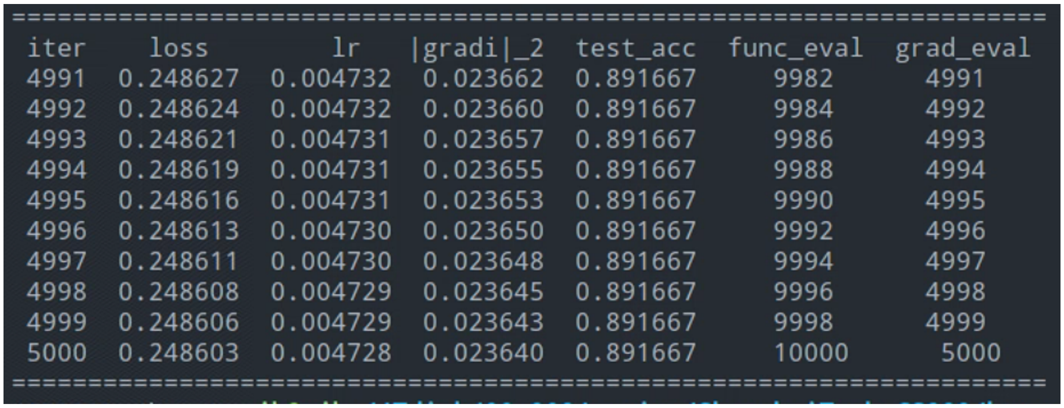
$\nabla E_{\text{in}}(\mathbf{w}), y_n, \mathbf{x}_n$ are set for the training set.
To set the learning rate getting smaller as the training iterations go on, we can set the learning rate as:

$$\eta_t = \eta \cdot \|\nabla E_{\text{in}}(\mathbf{w})\|$$

where $\eta = 0.2$ is the initial learning rate, and $\|\nabla E_{\text{in}}(\mathbf{w})\|$ is the norm of the gradient.
And the total iterated steps is set to be 5000 iterataions.

3. The results of the last 10 steps are shown in Figure 3.
We could see that the training loss, the L_2 norm of the gradient $\|\nabla E_{\text{in}}(\mathbf{w})\|$ are dropping, but quite

slow, which is close to get converage.
And the number of the gradient evaluations is once per iterataion, used for updating the $\mathbf{w}$.
And the number of the gradient evaluations is twice per iterataion, used for calculating the loss for both training set and the testing set.



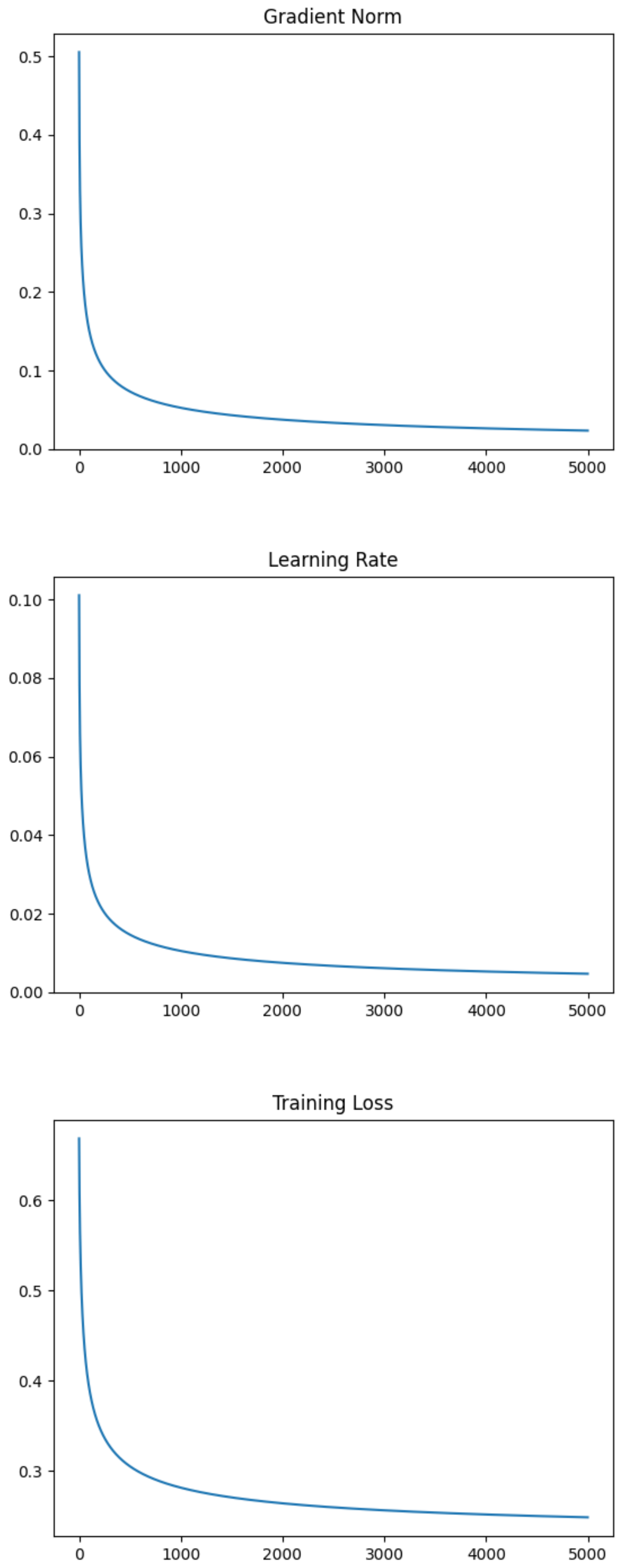
```
=================================================================
 iter    loss        lr    |gradi|_2  test_acc  func_eval  grad_eval
 4991  0.248627  0.004732  0.023662  0.891667     9982       4991
 4992  0.248624  0.004732  0.023660  0.891667     9984       4992
 4993  0.248621  0.004731  0.023657  0.891667     9986       4993
 4994  0.248619  0.004731  0.023655  0.891667     9988       4994
 4995  0.248616  0.004731  0.023653  0.891667     9990       4995
 4996  0.248613  0.004730  0.023650  0.891667     9992       4996
 4997  0.248611  0.004730  0.023648  0.891667     9994       4997
 4998  0.248608  0.004729  0.023645  0.891667     9996       4998
 4999  0.248606  0.004729  0.023643  0.891667     9998       4999
 5000  0.248603  0.004728  0.023640  0.891667     10000       5000
=================================================================
```

Figure 3: The last 10 steps' output

4. Analysis:
Figure 5 are the testing accuracy and testing loss. From the testing loss, we could find that the model somehow overfitted on the training data, as the testing loss gradually becomes higher as the iterataions grow. But the testing accuracy was not effected so much, it might because the dataset is small, and the data distribution are regular, so overfitting on the training set did not make testing accuracy got worse.

Figure 4 are the gradient's L_2 norm $\|\nabla E_{\text{in}}(\mathbf{w})\|$, the learning rate $\eta_t = \eta \cdot \|\nabla E_{\text{in}}(\mathbf{w})\|$, and the training loss.
The learning rate is obviously to have the same trend with the $\|\nabla E_{\text{in}}(\mathbf{w})\|$ as they have the proportional relationship. The training loss also has the quite same trend with them. Decreasing rapidly at first, and getting much slower as getting converage.

5. The code and the method to run the code are all in the folder 'code'.

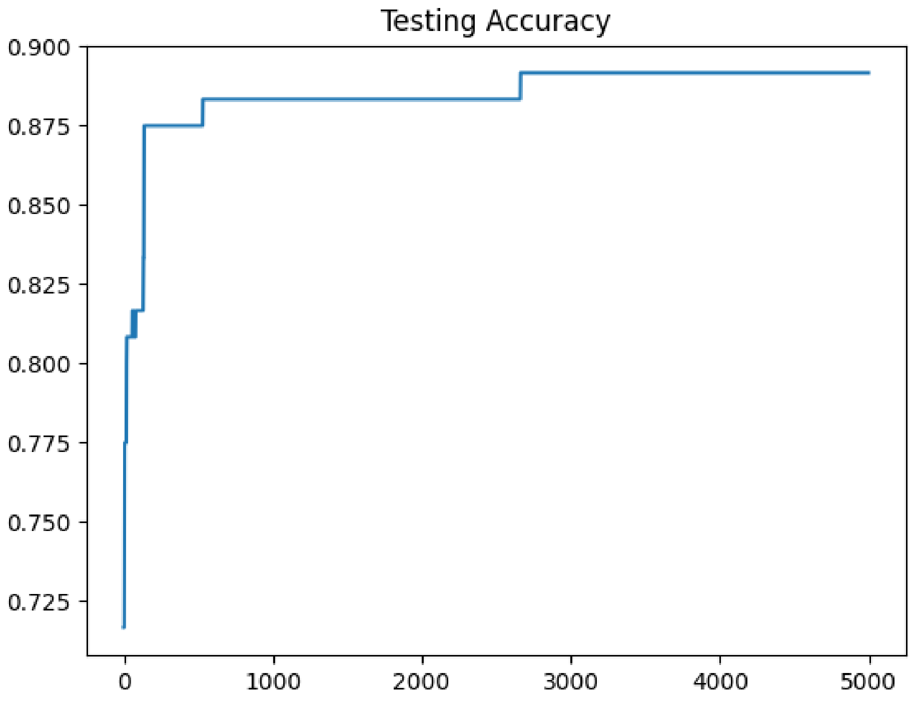


Gradient Norm

Learning Rate

Training Loss

Figure 4: The training information



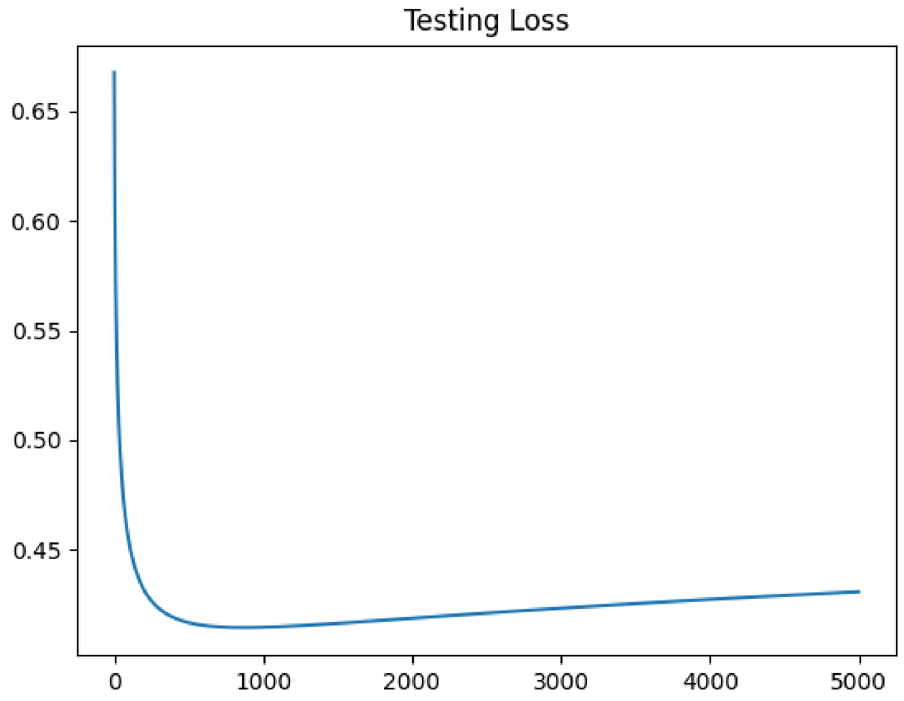


Figure 5: The testing information