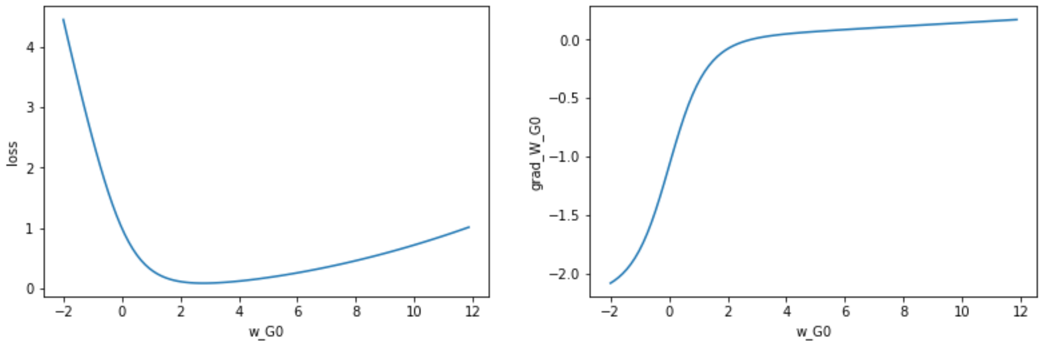
# Problem 1:

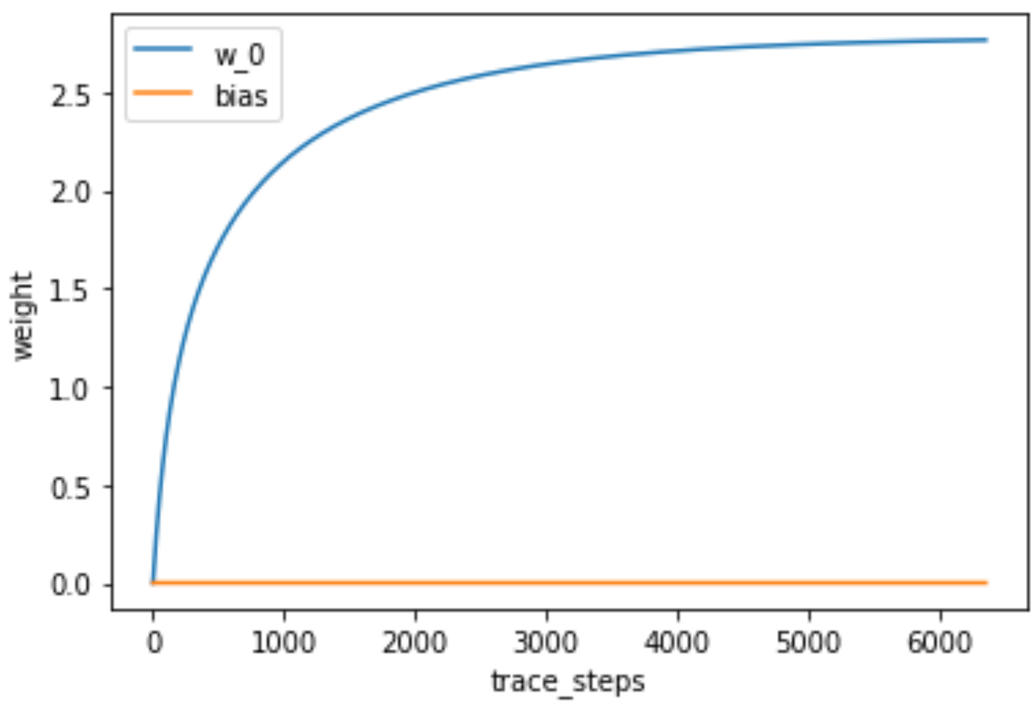
## a:

The loss become smaller at first and become larger, the gradient becomes larger and larger. Meanwhile, the min_loss's w_G is equal to the one with grad is zero.



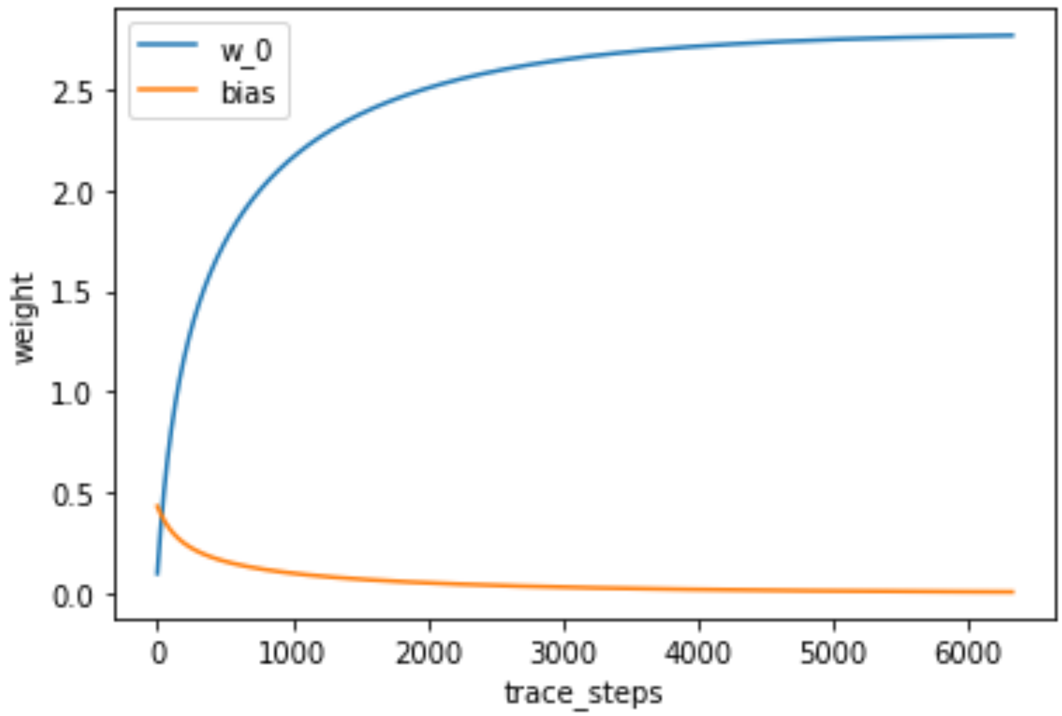

Yes, the loss trend makes sense and the gradient look as expected. Because there is a certain w_1 which can get the min loss.
I think it is 2.77.

b:




Yes, I saw the expected trend. The w_1 goes to 2.77 and the w_bias goes to 0.

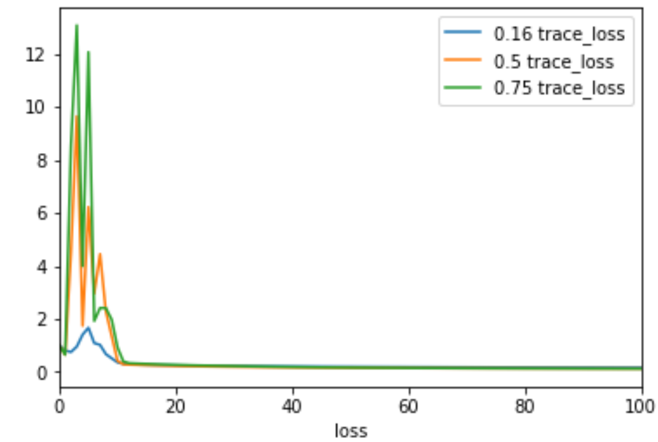
c:

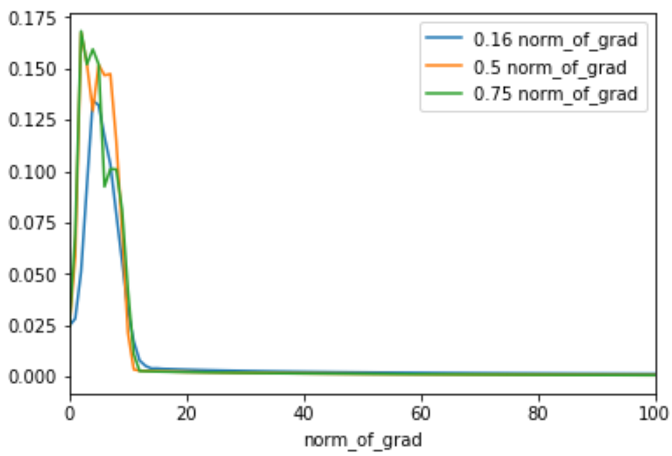


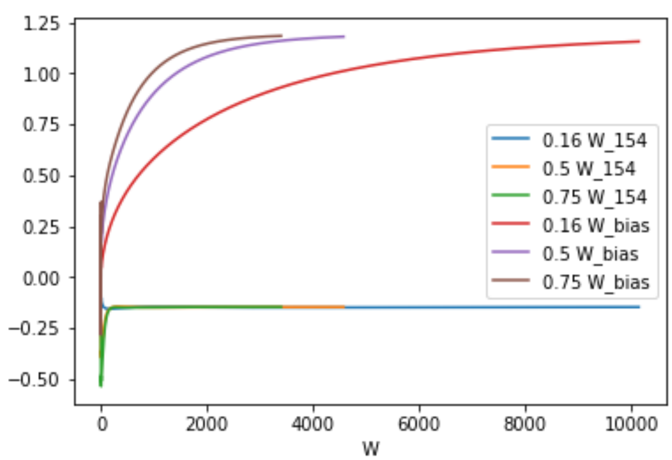

Yes, I saw the expected trend. The w_1 goes to 2.77 and the w_bias goes to 0.

# Problem 2

1a:


0.16 trace_loss
0.5 trace_loss
0.75 trace_loss
loss


0.16 norm_of_grad
0.5 norm_of_grad
0.75 norm_of_grad
norm_of_grad


0.16 W_154
0.5 W_154
0.75 W_154
0.16 W_bias
0.5 W_bias
0.75 W_bias
W

I noticed that the smaller the step size is, the smaller changer in loss it will be, and it quickly goes down within 10 iterations. And as the step_size become bigger the converge speed will be quicker.

## 2b:

$\alpha$=0.0001
Did not find a step_size can converge within 30000 iterations.
So I choosed 1.81 as the stepsize
The first slipt error rate was 0.0524. The second slipt error rate was 0.0478. The third slipt error rate was 0.0483.

$\alpha$=0.001
Did not find a step_size can converge within 30000 iterations.
So I choosed 1.7 as the stepsize
The first slipt error rate was 0.0524. The second slipt error rate was 0.0475. The third slipt error rate was 0.0481.

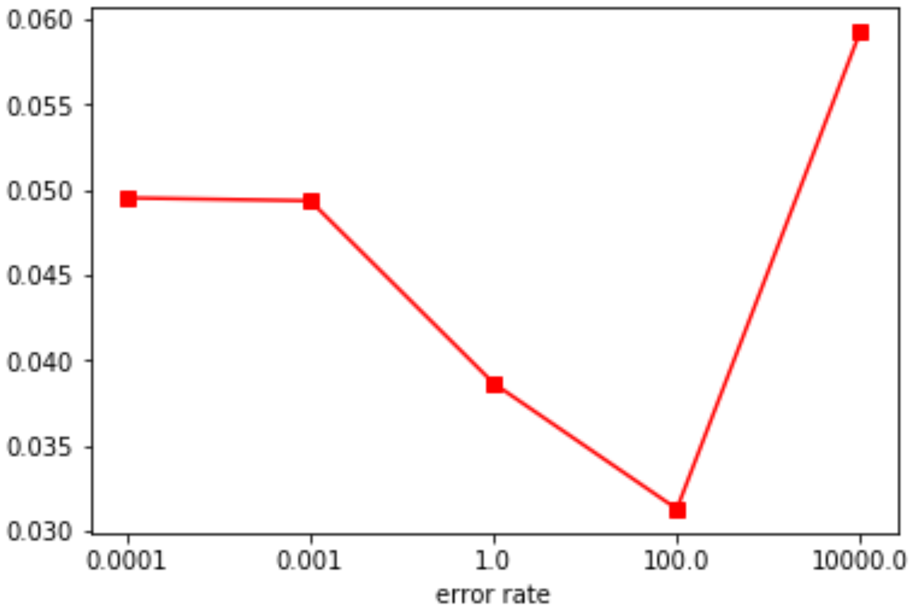
$\alpha$=1
The step_size I choosed was 0.8, it converged for the data.
The first slipt error rate was 0.0386. The second slipt error rate was 0.0389. The third slipt error rate was 0.0384.

$\alpha$=100
The step_size I choosed was 0.1, it converged for the data.
The first slipt error rate was 0.0333. The second slipt error rate was 0.0290. The third slipt error rate was 0.0315.

$\alpha$=10000
The step_size I choosed was 0.01, it converged for the data.
The first slipt error rate was 0.0666. The second slipt error rate was 0.0592. The third slipt error rate was 0.0519.

I think when $\alpha$ is smaller than 100, it is overfitting, so as the $\alpha$ become larger, the result become better. While when $\alpha$ is bigger than 100, it is underfitting, so as the $\alpha$ become larger, the result become worse.

So I choose $\alpha$ = 100.




**2c:**

| Predicted | 0 | 1 |
|---|---|---|
| **Actual** | | |
| **0.0** | 1860 | 62 |
| **1.0** | 69 | 1943 |

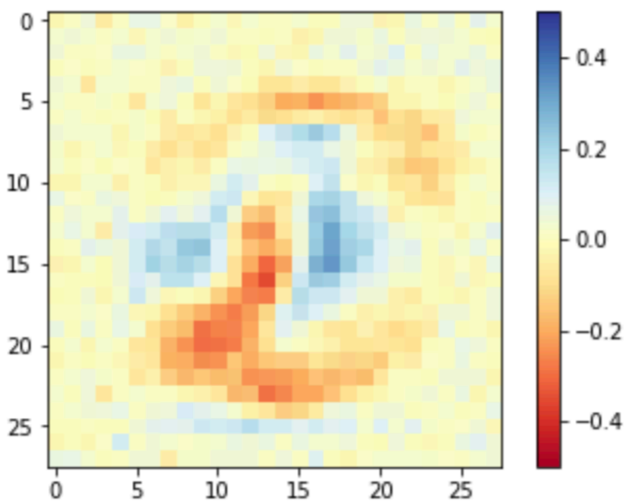


We can see that, for the false positive, most of their's lower circle is not very obvious. The lower circle is some kind like a straight line.

And for the false negative, most of their's lower straight line is not straight, which is bending, looks like a circle.

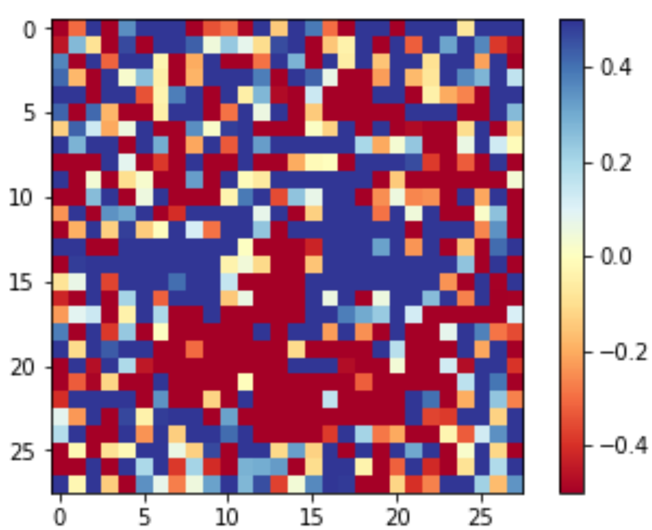
## 2d:

best alpha:

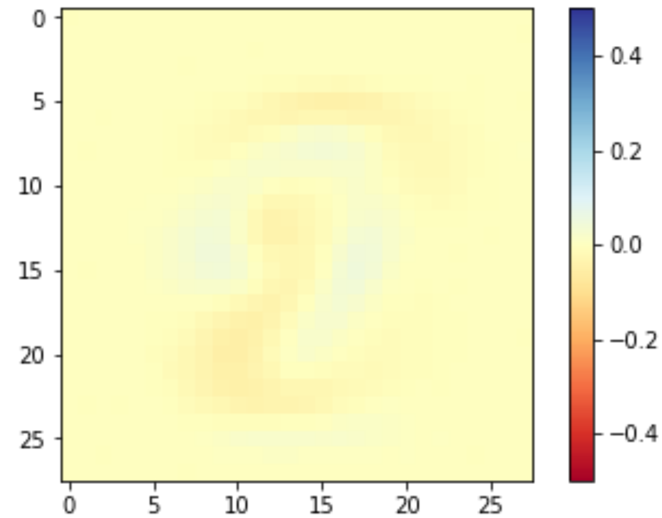



It seems the pixels associated with a digit '8' gathers on the pixels which is difference parts of '8' and '9', and the pixels associated with a digit '9' gathers on the parts same parts of '8' and '9'.
The lower and upper left arc pixels are associated with a digit '8'.
The middle arc pixels are associated with a digit '8'.

alpha is too small:




alpha is too large:

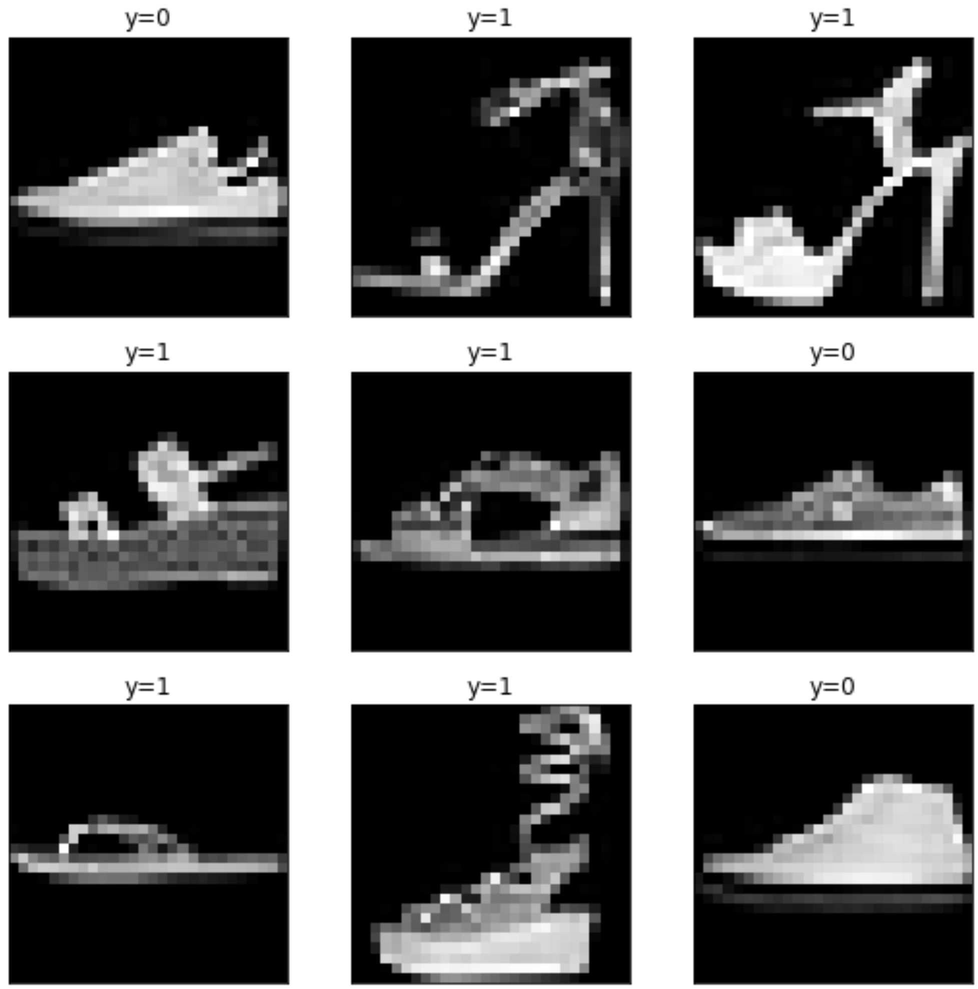



## 2e:

The error rate is 0.033787191124558746
The mean error rate of the 3-fold validation 0.031
They are match

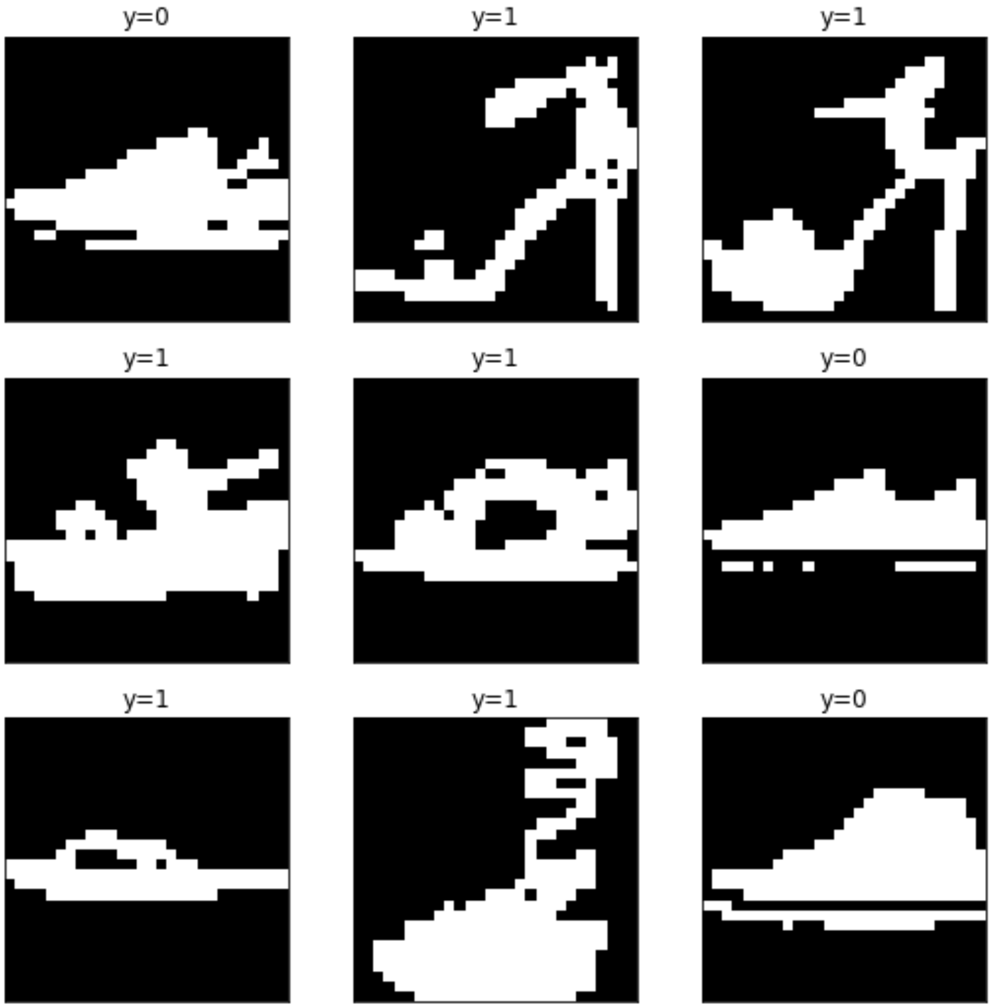
## Problem 3:

When I checked the data set, I noticed most of the picture are like this:




It is not very clear, and it seems has some noise on these pictures, especially on the outlier of the shoes, it doesn’t affect the judgement of the classifier of the shoes, but it actually has some very little value that will affect the result.
For human, what really affect the result is the profile of the picture. Like whether there is some hole in the shoes or whether it is very thick or thin.
So I was thinking whether there is some ways I can delete the irrelevant values and the noise, and just reserve the value of the profile. So the contrast of the image will be enhanced, then wo can clearly determine which part of the image is shoes and which part is hole.
So I made some try, I tried some threshold, the pixels which are larger than the threshold will be turned to 1 and the pixels which are smaller than the threshold will be turned to 0. After several times of trying, I noticed that 0.095 was the best

threshold. In this situation, the image was the most clear one and I can distinguish the image in a very high accuracy.

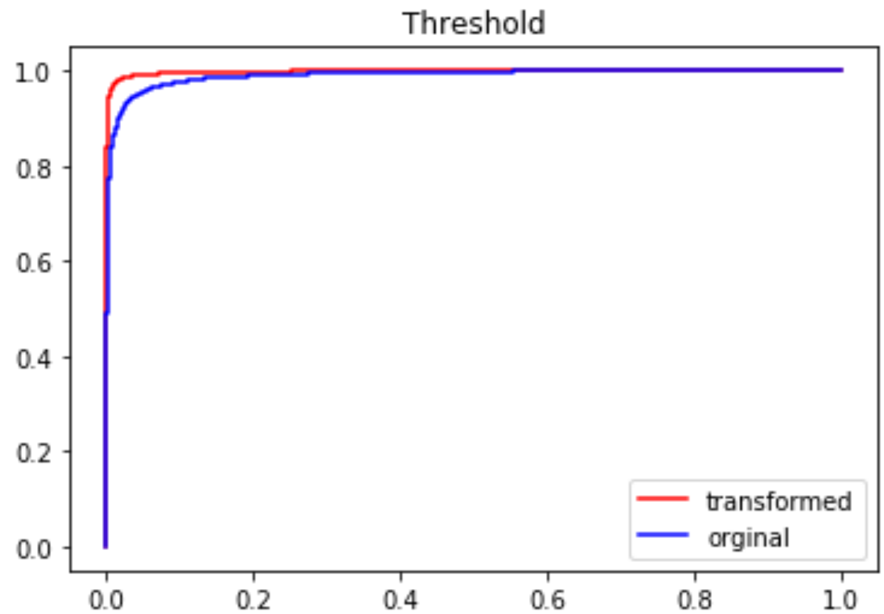



So I used the new set as my training set, and use the lr.fit(x_tr, y_tr) to get the w_G.
For the new training set and the original one, the confusion matrix are:

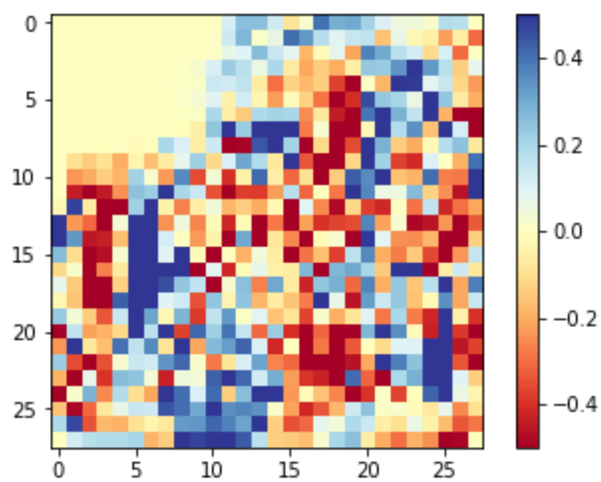
| **Predicted** | **0** | **1** |
|---|---|---|
| **Actual** | | |
| **0.0** | 5925 | 75 |
| **1.0** | 149 | 5851 |

| **Predicted** | **0** | **1** |
|---|---|---|
| **Actual** | | |
| **0.0** | 5820 | 180 |
| **1.0** | 383 | 5617 |

And the ROC curve is:




The weight is:




From the weight image, we can know the part of the shoes which should be holes for the sandal is negative, and the part which is have information for both sandal and sneaker is positive.
Before I predict the test set, I did the same preprocess to the test set, then I predicted it as result.