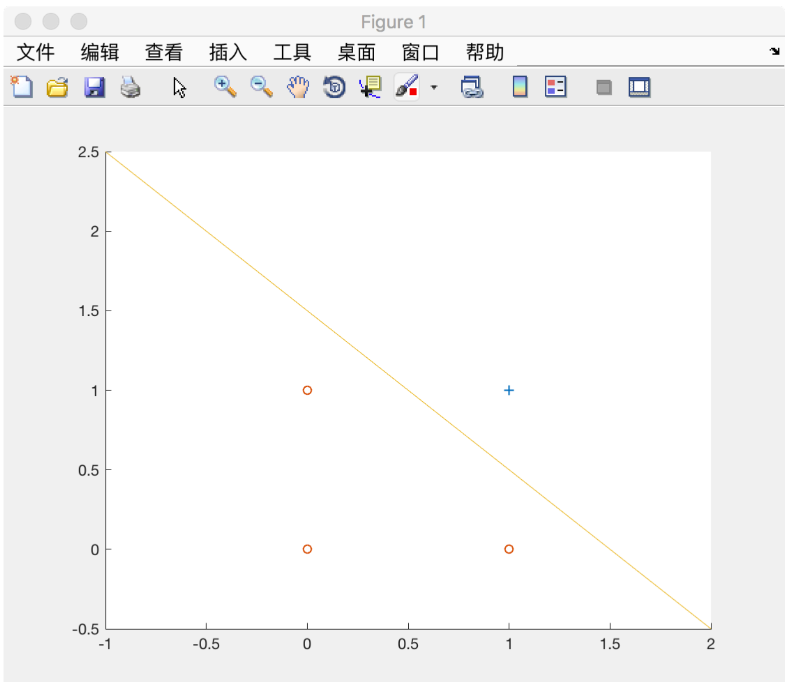
1 Perceptron Algorithm and Convergence Analysis

1.

(a) A two-input Boolean function can be: $y = x1 \wedge x2$:

x1	x2	$y = x1 \wedge x2$
0	0	0 (-)
0	1	0 (-)
1	0	0 (-)
1	1	1 (+)



The equation of a separating hyperplane is: $y = -x1 + x2 + 1.5 = 0$.

(b) A two-input Boolean function that can not be represented by a single perceptron can be: $y = x1$ XOR $x2$.

x1	x2	$y = x1$ XOR $x2$
0	0	0 (-)
0	1	1 (+)
1	0	1 (+)
1	1	0 (-)

Assume the hyperplane is w0 + w1x1 + w2x2 = 0.

w0 + 0 * w1 + 0 * w2 ⩽ 0 ⇔ w0 ⩽ 0

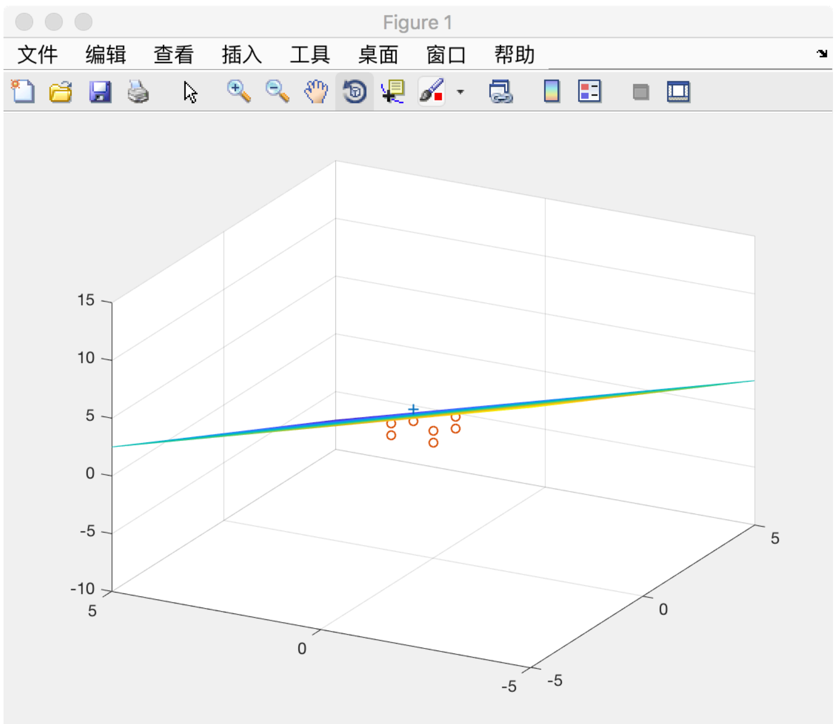
w0 + 0 * w1 + 1 * w2 ⩾ 0 ⇔ w0 ⩾ -w2

w0 + 1 * w1 + 0 * w2 ⩾ 0 ⇔ w0 ⩾ -w1

w0 + 1 * w1 + 1 * w2 ⩽ 0 ⇔ w0 ⩽ -w1 - w2 contradictory.

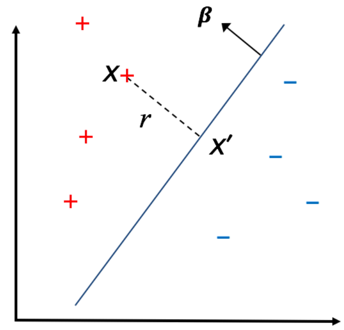
(c) A three-input Boolean function can be: y = x1 ^ x2 ^ x3 .

Only point (1, 1, 1) is classified as positive. All the other points are classified as negative.



The equation of a separating hyperplane is: $x_1/4 + x_2/4 + x_3/4 - 5/8 = 0$. The plane passes through point (1, 1, 0.5), point (1, 0.5, 1) and point (0.5, 1, 1).

2.



In the figure above, the Euclidean Distance from point x to the decision boundary is r. Point **x'** is point **x**' s corresponding point on the decision boundary. And **x**-**x'** is perpendicular to the decision boundary and parallel to the normal vector **w**.

So, **x'** = **x** - yr$\frac{\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|_2}$.

Since **x'** is on the decision boundary, $\boldsymbol{\beta}_0 + \boldsymbol{\beta}^T\mathbf{x}' = 0$.

$r = y\frac{\boldsymbol{\beta}_0+\boldsymbol{\beta}^T\mathbf{x}}{\|\boldsymbol{\beta}\|_2}$. Equivalently, r = $\frac{1}{\|\boldsymbol{\beta}\|_2}yf(x)$.

3.

$\left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^{SEP}\right\|^2 = \left\|\boldsymbol{w}^t + y_i\boldsymbol{x}_i - \boldsymbol{w}^{SEP}\right\|^2 = \left\|\boldsymbol{w}^t - \boldsymbol{w}^{SEP}\right\|^2 + y_i^2\|\mathbf{x}_i\|^2 + 2y_i\boldsymbol{w}^t\boldsymbol{x}_i - 2y_i\mathbf{w}^{SEP}\boldsymbol{x}_i$

$y_i^2\|\mathbf{x}_i\|^2 \leqslant 1$, $2y_i\boldsymbol{w}^t\boldsymbol{x}_i \leqslant 0$ and $-2y_i\mathbf{w}^{SEP}\boldsymbol{x}_i \leqslant -2$.

So, $\left\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^{SEP}\right\|^2 \leqslant \left\|\boldsymbol{w}^t - \boldsymbol{w}^{SEP}\right\|^2 - 1$

$0 = \left\|\boldsymbol{w}^{SEP} - \boldsymbol{w}^{SEP}\right\|^2 \leqslant \left\|\boldsymbol{w}^0 - \boldsymbol{w}^{SEP}\right\|^2 - T$ (Assume T steps to converge.)

$T \leqslant \left\|\boldsymbol{w}^0 - \boldsymbol{w}^{SEP}\right\|^2$.

2 Programming Assignment

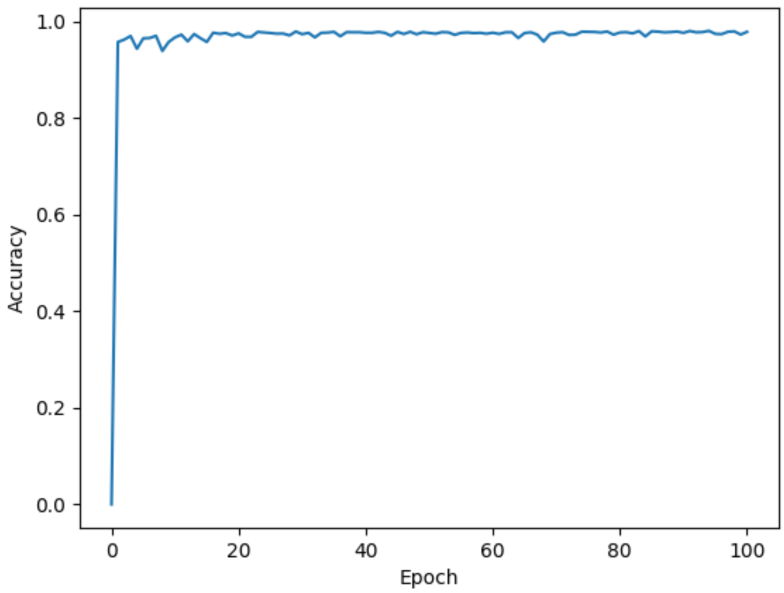
1.

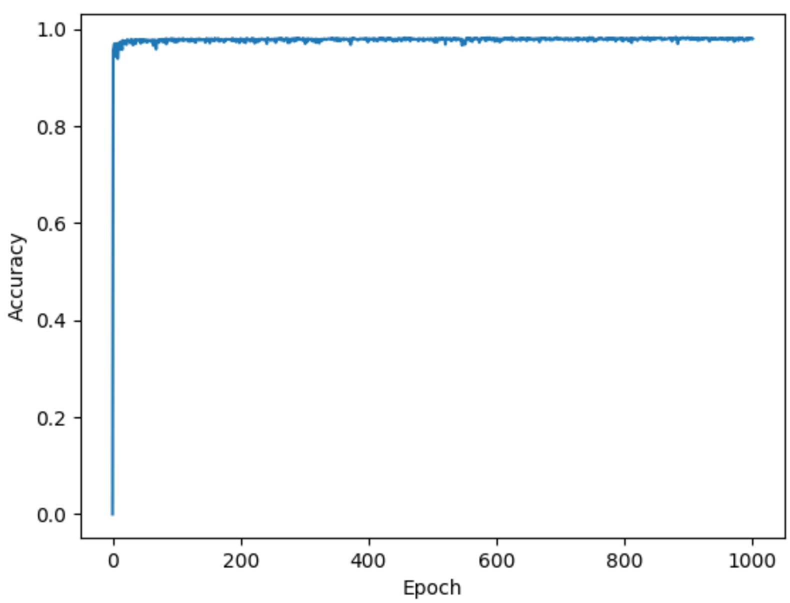
(a) Observation: the accuracy fluctuates and slowly approaches 1 as epochs grow. The following figures are when epochs = 100, epochs = 1000 and epochs = 10.

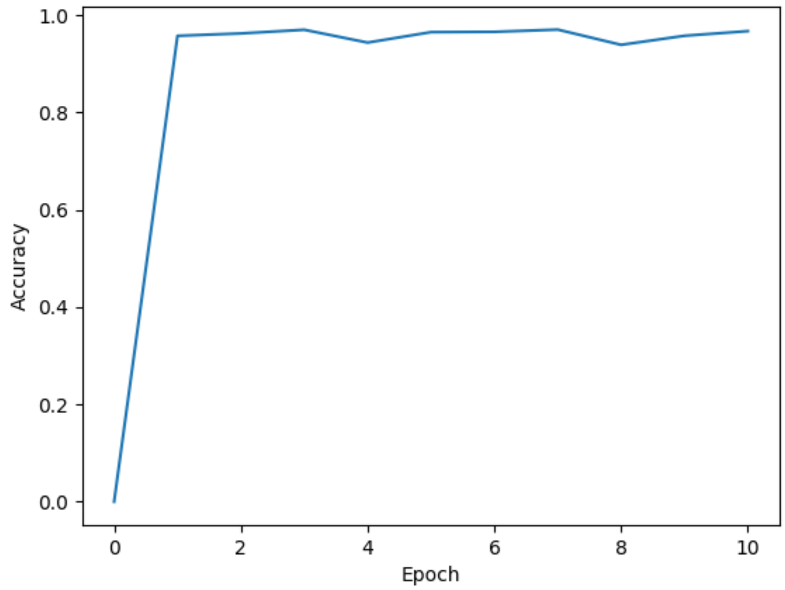
epochs	Final accuracy
10	0.9673518742442564
100	0.9787182587666263

1000	0.9804111245465538

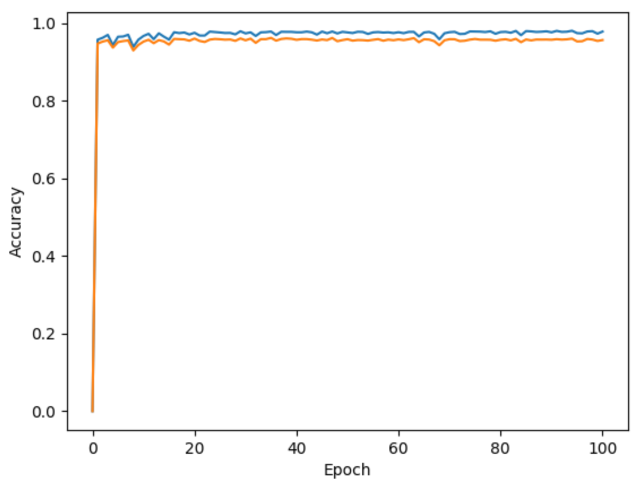
So we can go to the conclusion that the larger the epochs, the more accurate the model.







(b) Observation: Orange line for testing dataset and blue line for training dataset. So we can see the testing dataset always gets a lower accuracy than training dataset.

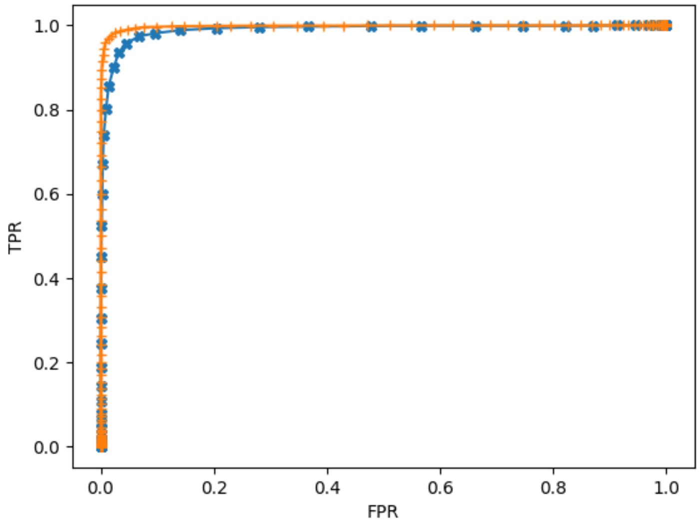


(c) the confusion matrix:

	y = +1	y = -1
$\hat{y}$ = +1	TP = 2637	FP = 137
$\hat{y}$ = -1	FN = 101	TN = 2637

Accuracy = 0.956821480406386

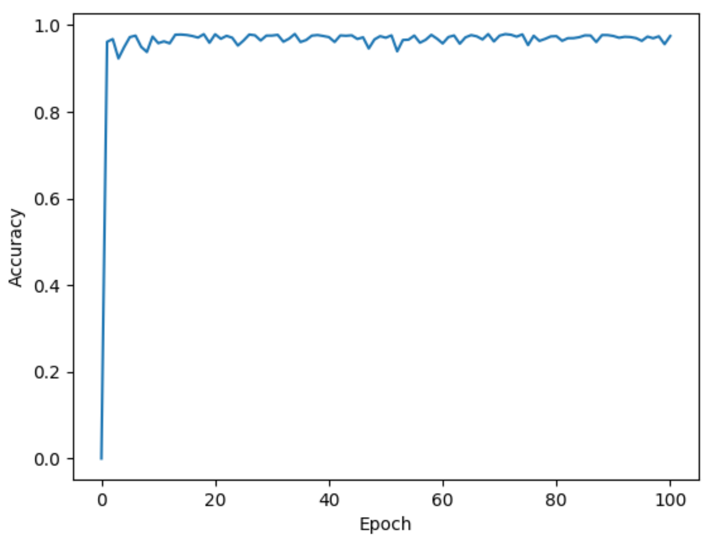
(d) The AUC of the classifier with weight vector w⋆ is larger than that of the classifier with w' . So, weight vector w⋆ leads to a better decision boundary.



(e) The AUC of the classifier with w' is 0.9912513137711291, and the AUC of the classifier with w⋆ is 0.998035309675626.

2.

(a) When eta = 0.1:



	y = +1	y = -1

$\hat{y}$ = +1	TP = 2582	FP = 80
$\hat{y}$ = -1	FN = 156	TN = 2694

Accuracy = 0.9571843251088534

(b) The technique I use to tune eta is: If $\boldsymbol{w}^*$ is a very good separator, $y_i(\boldsymbol{w}^* x_i) \geq \pounds$ for all i. So we need to find δ, which is the minimum margin. So I initialize δ to max float value and go through all the images x_i. If $y_i(\boldsymbol{w}^* x_i) > \mathbf{0}$ (since after 100 epochs the algorithm still can' t converge) and $y_i(\boldsymbol{w}^* x_i) < \pounds$, I update δ to $y_i(\boldsymbol{w}^* x_i)$. After the loop, I substitute δ into $\eta = \frac{1}{2}\ln\left(\frac{1+\pounds}{1-\pounds}\right)$ to get new η and test it on the test set to see if accuracy goes up. If the new accuracy we got is greater than the previous accuracy, we think this η is better than the previous η. We choose η with the highest accuracy to be our optimal η.

eta = 0.1, test accuracy = 0.9571843251088534

eta = 6.83143847e-06, test accuracy = 0.9571843251088534

eta = 4.86124361e-09, test accuracy = 0.9571843251088534

eta = 3.45812268e-12, test accuracy = 0.9609941944847605

eta = 3.33066907e-16, test accuracy = 0.9511973875181422

eta = 0.

So, optimal eta = 3.45812268e-12