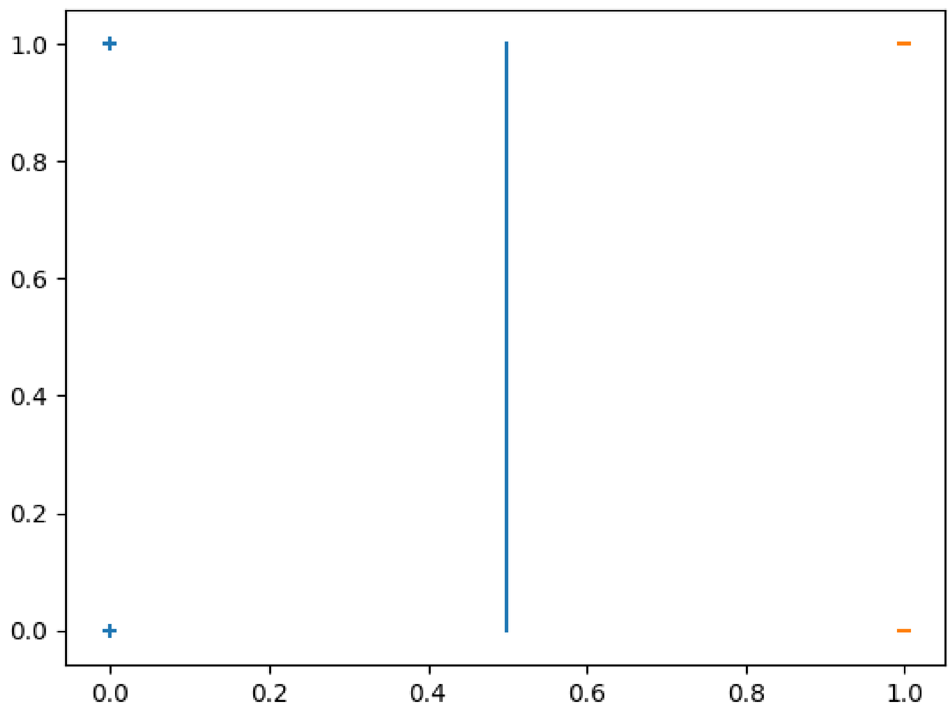
Homework 1

Weiyu Yan

1 Perceptron Algorithm and Convergence Analysis

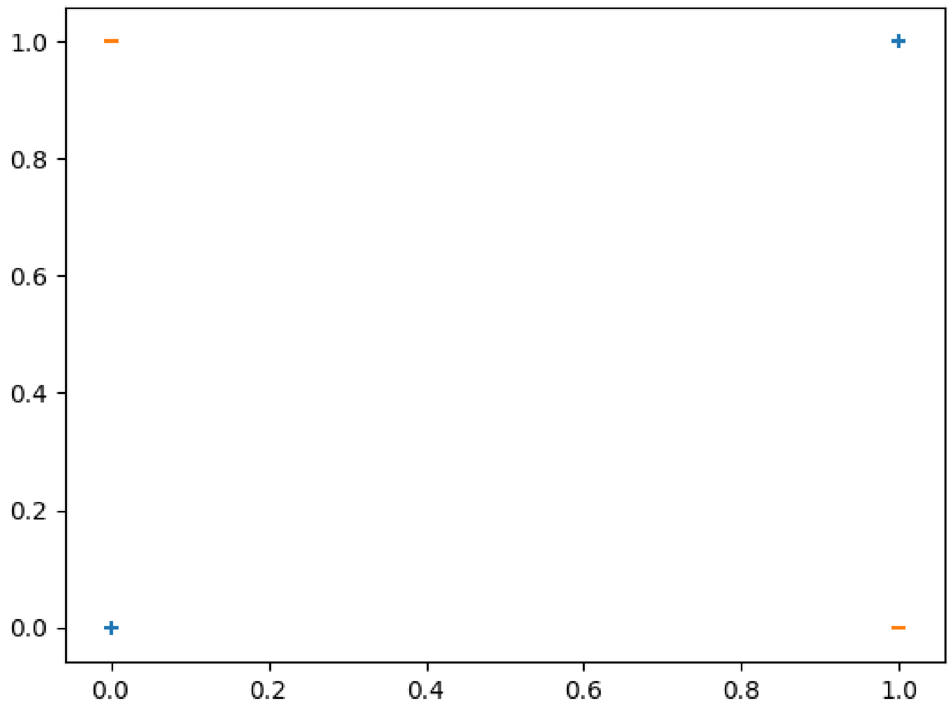
1. (a) weight vector is $w = (-1, 0)$



(b) A truth table for this function could be:

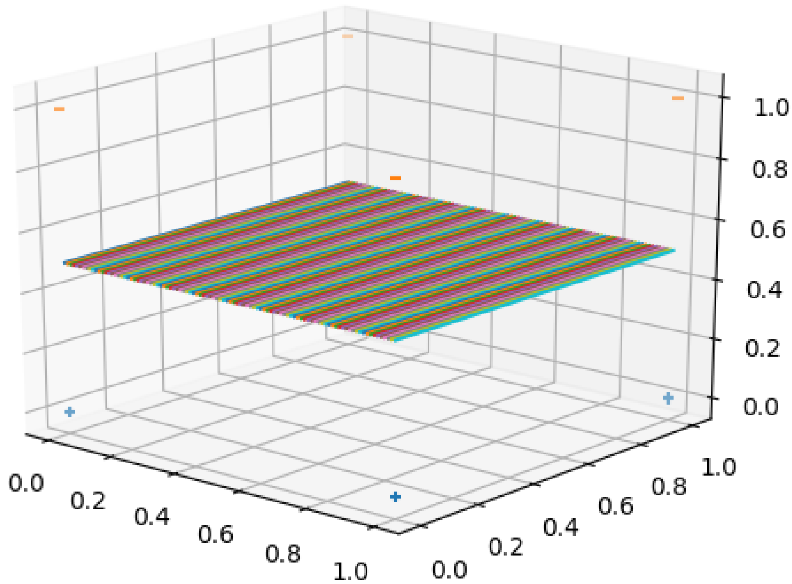
x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	1

We can plot those points in a figure.



The perceptron algorithm gives us a weight vector which corresponds to its orthogonal hyperplane to divide points into 2 parts. In this figure since we can't find a hyperplane to divide the 2 kinds, this function can't be represented by a single perceptron.

(c) weight vector is $w = (0, 0, -1)$



2. β is a weight vector perpendicular to the hyperplane the signed Euclidean Distance of the point x to the hyperplane is given by:

$$d = \frac{(x - x_0) \cdot \beta y}{\|\beta\|_2}$$

where

$$x_0 \in \{x | f(x) = 0\}$$

$$\because f(x_0) = \beta_0 + \beta^T x_0$$

$$\therefore \beta^T x_0 = -\beta_0$$

Therefore

$$
\begin{aligned}
d &= \frac{y(x \cdot \beta - x_0 \cdot \beta)}{\|\beta\|_2} \\
&= \frac{y(\beta^T x - \beta^T x_0)}{\|\beta\|_2} \\
&= \frac{y(\beta^T x + \beta_0)}{\|\beta\|_2} \\
&= \frac{y f(x)}{\|\beta\|_2}
\end{aligned} \tag{1}
$$

3.

$$
\begin{aligned}
&\|w^T - w^{sep}\|^2 \\
=&\|(w^{T-1} + x_i y_i) - w^{sep}\|^2 \\
=&\|w^{T-1} + x_i y_i\|^2 + \|w^{sep}\|^2 - 2(w^{T-1} + x_i y_i) w^{sep} \\
=&\|w^{T-1}\|^2 + \|x_i y_i\|^2 + 2w^{T-1} x_i y_i + \|w^{sep}\|^2 - 2w^{T-1} w^{sep} - 2x_i y_i w^{sep}
\end{aligned} \tag{2}
$$

$$
\because \|w^{T-1} - w^{sep}\|^2 = \|w^{T-1}\|^2 + \|w^{sep}\|_2^2 - 2w^{T-1} w^{sep}
$$

$$
\therefore \|w^T - w^{sep}\|^2 - \|w^{T-1} - w^{sep}\|^2 = \|x_i y_i\|^2 + 2w^{T-1} x_i y_i - 2x_i y_i w^{sep}
$$

since the weight vector only updates when making a mistake, we have:

$$
w^{T-1} x_i y_i \leq 0
$$

Also for the algorithm we persume: $\|x_i y_i\|^2 \leq 1$ and $x_i y_i w^{sep} \geq 1$
We can derive:

$$
\begin{aligned}
&(\|w^T - w^{sep}\|^2 - \|w^{T-1} - w^{sep}\|^2) \\
&+ (\|w^{T-1} - w^{sep}\|^2 - \|w^{T-2} - w^{sep}\|^2) + \cdots \\
&+ (\|w^1 - w^{sep}\|^2 - \|w^0 - w^{sep}\|^2) \\
&= \|w^T - w^{sep}\|^2 - \|w^0 - w^{sep}\|^2 \leq \sum_{t=1}^{T} -1 = -T
\end{aligned} \tag{3}
$$

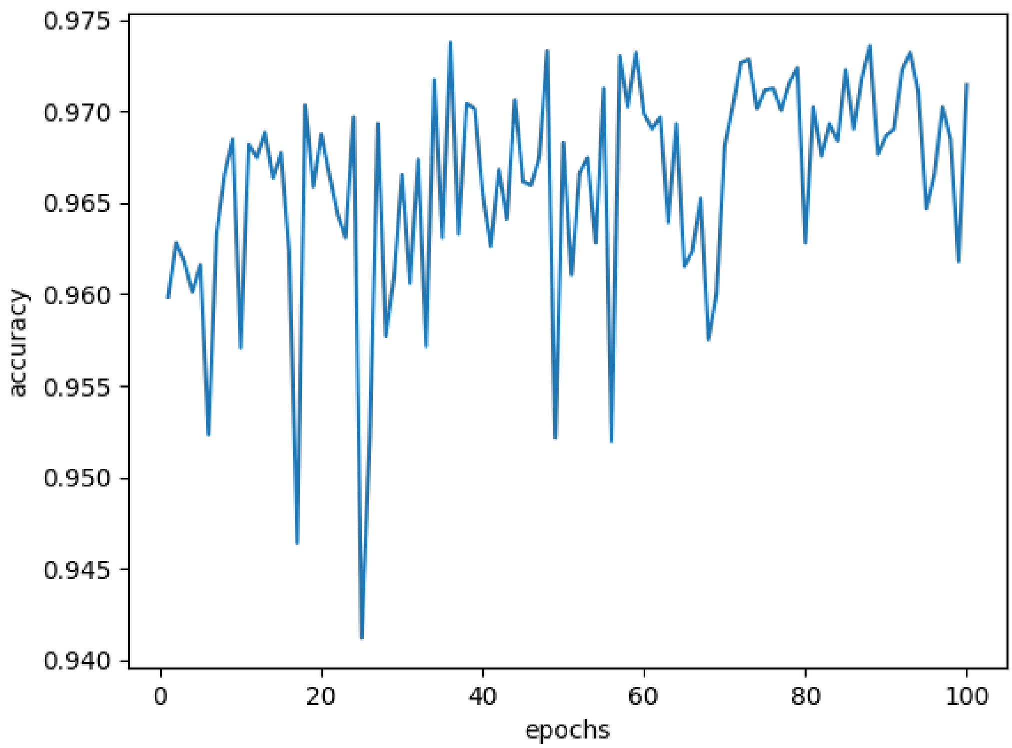
Therefore,

$$
0 \leq \|w^T - w^{sep}\|^2 \leq \|w^0 - w^{sep}\|^2 - T
$$

$$
\therefore T \leq \|w^0 - w^{sep}\|^2
$$

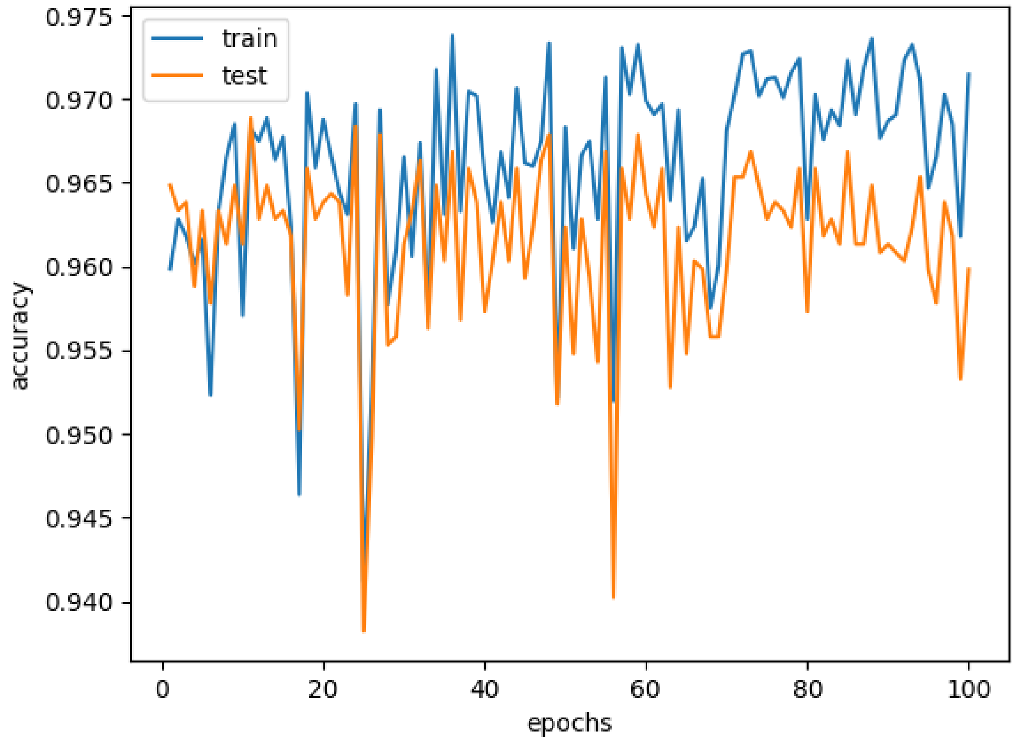
2 Programming Assignment

Please refer code in 'hw1.py'



1. (a)

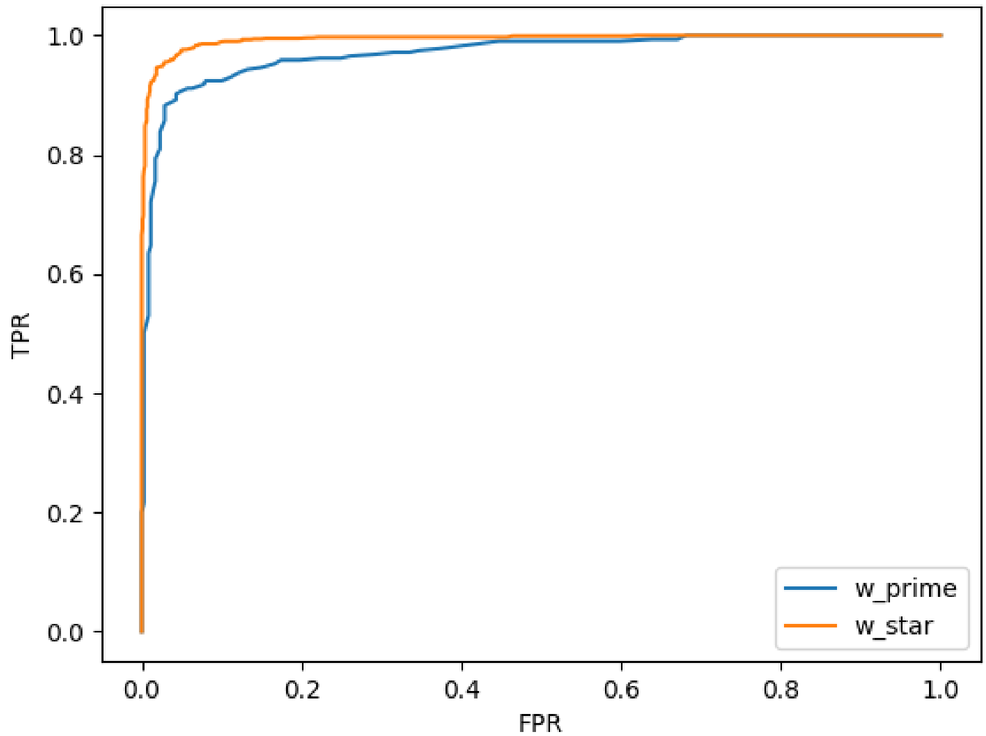
   Since first epoch the accuracy is already pretty good.
   Then it shakes drastically at first(within a small bound, in general the accuracy is always above 0.94), then becomes more steady, with the moving average increasing slowly.
   By increasing/decreasing number of epochs, the curve becomes denser/sparser – just revealing more/less information in the figure, the points' location won't change.



(b) The curve for test set roughly follows the same pattern of traning set. With the epoch number going up, it becomes lower than the training set.

(c) accuracy: 0.9598191863385234
confusion matrix:

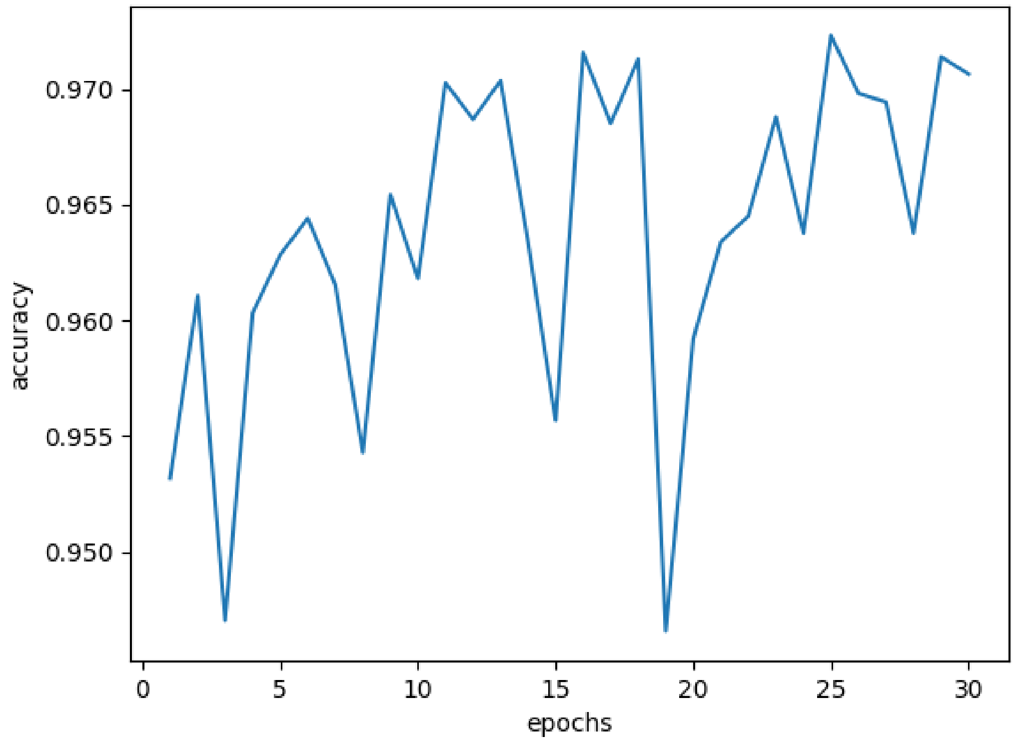
$$\begin{bmatrix} 965 & 38 \\ 44 & 944 \end{bmatrix}$$



(d)

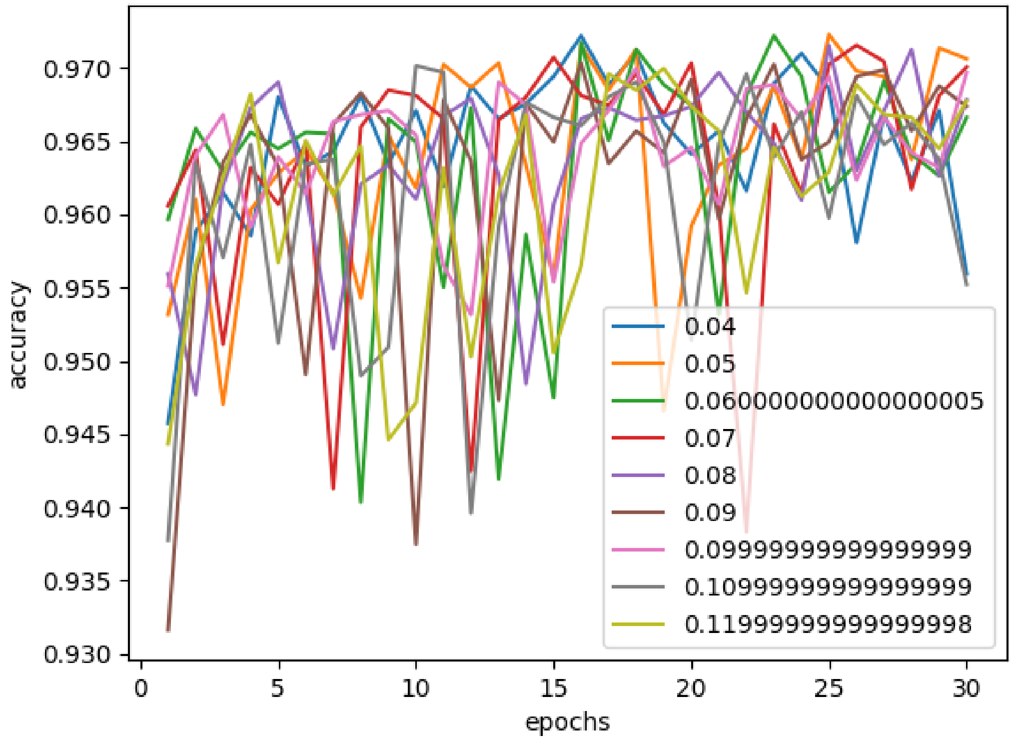
(e) $\text{AUC}(w') = 0.9698776549779418$
$\text{AUC}(w^*) = 0.9937542766829695$
The greater the AUC value, the bigger area that is under the ROC curve, the 'upper' that the ROC curve goes.

2. (a) Accuracy evolution for training set with $eta = 0.05$ after 30 epochs:



For test set:
accuracy: 0.9658463083877449
confusion matrix:

$$\begin{bmatrix} 982 & 41 \\ 27 & 941 \end{bmatrix}$$



(b)

The technique I use is to first plot accuracy curves using small epoch numbers and look for a smaller range of this best η, I presume $argmax(f(\eta))$ is a convex function with local min being its global min. Then I wrote a loop to plot curves with better precision in this small range. Observe from the figure that $\eta \approx 0.05$ is the best value in my experiment.