HW 1

Kevin Lin

1/26/2026

1

Let:

$$A = \begin{bmatrix} 4 & 1 & 3 & 6 \\ 2 & 7 & 5 & 3 \end{bmatrix}, B = \begin{bmatrix} 0 & 4 \\ 7 & 6 \\ 5 & 8 \\ 3 & 11 \end{bmatrix}, C = \begin{bmatrix} -13 & 0 & 2 \\ 5 & 2 & 10 \\ 0 & 7 & 9 \end{bmatrix}, D = \begin{bmatrix} 5 & -3 & -7 \\ 4 & 0 & 10 \\ 7 & 3 & 11 \end{bmatrix}, E = \begin{bmatrix} -4 & 5 \\ 12 & 7 \end{bmatrix}$$

(a) $(3B)^T$:

$$\begin{aligned} (3B)^T &= 3 \cdot B^T \\ &= 3 \cdot \begin{bmatrix} 0 & 7 & 5 & 3 \\ 4 & 6 & 8 & 11 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 21 & 15 & 9 \\ 12 & 18 & 24 & 33 \end{bmatrix} \end{aligned}$$

(b) $(A - B)^T$ is not possible due to dimension mismatch. A is 2×4 while B is 4×2.

(c) $(2B^T - A)^T$:

$$\begin{aligned} (2B^T - A)^T &= 2B - A^T \\ &= 2 \cdot \begin{bmatrix} 0 & 4 \\ 7 & 6 \\ 5 & 8 \\ 3 & 11 \end{bmatrix} - \begin{bmatrix} 4 & 2 \\ 1 & 7 \\ 3 & 5 \\ 6 & 3 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 8 \\ 14 & 12 \\ 10 & 16 \\ 6 & 22 \end{bmatrix} - \begin{bmatrix} 4 & 2 \\ 1 & 7 \\ 3 & 5 \\ 6 & 3 \end{bmatrix} \\ &= \begin{bmatrix} -4 & 6 \\ 13 & 5 \\ 7 & 11 \\ 0 & 19 \end{bmatrix} \end{aligned}$$

(d) $(C + 2D^T + E)^T$ is not possible due to dimension mismatch. C and D are both 3×3 while E is 2×2.

(e) $(-A)^T E$:

$$\begin{aligned}
(-A)^T E &= -A^T E \\
&= -\begin{bmatrix} 4 & 2 \\ 1 & 7 \\ 3 & 5 \\ 6 & 3 \end{bmatrix} \cdot \begin{bmatrix} -4 & 5 \\ 12 & 7 \end{bmatrix} \\
&= -\begin{bmatrix} 4 \cdot -4 + 2 \cdot 12 & 4 \cdot 5 + 2 \cdot 7 \\ 1 \cdot -4 + 7 \cdot 12 & 1 \cdot 5 + 7 \cdot 7 \\ 3 \cdot -4 + 5 \cdot 12 & 3 \cdot 5 + 5 \cdot 7 \\ 6 \cdot -4 + 3 \cdot 12 & 6 \cdot 5 + 3 \cdot 7 \end{bmatrix} \\
&= -\begin{bmatrix} -16 + 24 & 20 + 14 \\ -4 + 84 & 5 + 49 \\ -12 + 60 & 15 + 35 \\ -24 + 36 & 30 + 21 \end{bmatrix} \\
&= -\begin{bmatrix} 8 & 34 \\ 80 & 54 \\ 48 & 50 \\ 12 & 51 \end{bmatrix} \\
&= \begin{bmatrix} -8 & -34 \\ -80 & -54 \\ -48 & -50 \\ -12 & -51 \end{bmatrix}
\end{aligned}$$

2

No, $AB \neq BA$. Matrix multiplication is not commutative. We can prove this by calculating both AB and BA:

$$AB = \begin{bmatrix} 2 & 7 & 3 \\ 1 & 0 & 9 \\ -1 & 2 & 10 \end{bmatrix} \begin{bmatrix} -2 & 0 & 3 \\ 2 & -1 & 7 \\ 6 & 4 & -3 \end{bmatrix}$$

$$AB = \begin{bmatrix} 2 \cdot -2 + 7 \cdot 2 + 3 \cdot 6 & 2 \cdot 0 + 7 \cdot -1 + 3 \cdot 4 & 2 \cdot 3 + 7 \cdot 7 + 3 \cdot -3 \\ 1 \cdot -2 + 0 \cdot 2 + 9 \cdot 6 & 1 \cdot 0 + 0 \cdot -1 + 9 \cdot 4 & 1 \cdot 3 + 0 \cdot 7 + 9 \cdot -3 \\ -1 \cdot -2 + 2 \cdot 2 + 10 \cdot 6 & -1 \cdot 0 + 2 \cdot -1 + 10 \cdot 4 & -1 \cdot 3 + 2 \cdot 7 + 10 \cdot -3 \end{bmatrix}$$

$$AB = \begin{bmatrix} -4 + 14 + 18 & 0 - 7 + 12 & 6 + 49 - 9 \\ -2 + 0 + 54 & 0 + 0 + 36 & 3 + 0 - 27 \\ 2 + 4 + 60 & 0 - 2 + 40 & -3 + 14 - 30 \end{bmatrix}$$

$$AB = \begin{bmatrix} 28 & 5 & 46 \\ 52 & 36 & -24 \\ 66 & 38 & -19 \end{bmatrix}$$

$$BA = \begin{bmatrix} -2 & 0 & 3 \\ 2 & -1 & 7 \\ 6 & 4 & -3 \end{bmatrix} \begin{bmatrix} 2 & 7 & 3 \\ 1 & 0 & 9 \\ -1 & 2 & 10 \end{bmatrix}$$

$$BA = \begin{bmatrix} -2 \cdot 2 + 0 \cdot 1 + 3 \cdot -1 & -2 \cdot 7 + 0 \cdot 0 + 3 \cdot 2 & -2 \cdot 3 + 0 \cdot 9 + 3 \cdot 10 \\ 2 \cdot 2 + -1 \cdot 1 + 7 \cdot -1 & 2 \cdot 7 + -1 \cdot 0 + 7 \cdot 2 & 2 \cdot 3 + -1 \cdot 9 + 7 \cdot 10 \\ 6 \cdot 2 + 4 \cdot 1 + -3 \cdot -1 & 6 \cdot 7 + 4 \cdot 0 + -3 \cdot 2 & 6 \cdot 3 + 4 \cdot 9 + -3 \cdot 10 \end{bmatrix}$$

$$BA = \begin{bmatrix} -4 + 0 - 3 & -14 + 0 + 6 & -6 + 0 + 30 \\ 4 - 1 - 7 & 14 + 0 + 14 & 6 - 9 + 70 \\ 12 + 4 + 3 & 42 + 0 - 6 & 18 + 36 - 30 \end{bmatrix}$$

$$BA = \begin{bmatrix} -7 & -8 & 24 \\ -4 & 28 & 67 \\ 19 & 36 & 24 \end{bmatrix}$$

Thus, $AB \neq BA$.

3

- ℓ_1 norm of $[0, 0, 0]$:

  $$||[0, 0, 0]||_1 = |0| + |0| + |0| = 0$$

  ℓ_2 norm of $[0, 0, 0]$:

  $$||[0, 0, 0]||_2 = \sqrt{0^2 + 0^2 + 0^2} = 0$$

  ℓ_∞ norm of $[0, 0, 0]$:

  $$||[0, 0, 0]||_\infty = \max(|0|, |0|, |0|) = 0$$

- ℓ_1 norm of $[1,2,3]$:
$$||[1,2,3]||_1 = |1| + |2| + |3| = 6$$
ℓ_2 norm of $[1,2,3]$:
$$||[1,2,3]||_2 = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$$
ℓ_∞ norm of $[1,2,3]$:
$$||[1,2,3]||_\infty = \max(|1|, |2|, |3|) = 3$$
- ℓ_1 norm of $[2,4,6]$:
$$||[2,4,6]||_1 = |2| + |4| + |6| = 12$$
ℓ_2 norm of $[2,4,6]$:
$$||[2,4,6]||_2 = \sqrt{2^2 + 4^2 + 6^2} = \sqrt{56} = 2\sqrt{14}$$
ℓ_∞ norm of $[2,4,6]$:
$$||[2,4,6]||_\infty = \max(|2|, |4|, |6|) = 6$$
These norms are related to those of $[1,2,3]$ by a factor of 2.
- No, a vector can not have a negative norm by definition as the norm is positive definite.

4

Given $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}$ where $x_i \in \mathbb{R}^m$ for all i, and $Y^T = [y_1, y_2, \ldots, y_n] \in \mathbb{R}^{p \times n}$ where $y_i \in \mathbb{R}^p$ for all i, $XY = \sum_{i=1}^n x_i y_i^T$. We can prove this as follows:

$$\begin{aligned}
XY &= X \cdot Y \\
&= [x_1, x_2, \ldots, x_n] \cdot \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{bmatrix} \\
&= x_1 y_1^T + x_2 y_2^T + \cdots + x_n y_n^T \\
&= \sum_{i=1}^n x_i y_i^T
\end{aligned}$$

5

Given $X \in \mathbb{R}^{m\times n}$, we can prove X^TX is symmetric and positive semi-definite as follows:

- Symmetric:

$$\begin{aligned}(X^TX)^T &= X^T(X^T)^T \\ &= X^TX\end{aligned}$$

- Positive semi-definite: For any non-zero vector $z \in \mathbb{R}^n$,

$$\begin{aligned}z^T(X^TX)z &= (Xz)^T(Xz) \\ &= ||Xz||_2^2 \\ &\geq 0\end{aligned}$$

6

Given $g(x, y) = e^{(x+y)} + e^{3xy} + e^{y^4}$, we can compute the partial derivatives as follows:

$$\begin{aligned}\frac{\partial g}{\partial x} &= \frac{\partial}{\partial x}\left(e^{(x+y)} + e^{3xy} + e^{y^4}\right) \\ &= e^{(x+y)} + 3ye^{3xy} + 0 \\ &= e^{(x+y)} + 3ye^{3xy} \\ \frac{\partial g}{\partial y} &= \frac{\partial}{\partial y}\left(e^{(x+y)} + e^{3xy} + e^{y^4}\right) \\ &= e^{(x+y)} + 3xe^{3xy} + 4y^3e^{y^4} \\ &= e^{(x+y)} + 3xe^{3xy} + 4y^3e^{y^4}\end{aligned}$$

7

Let $A = \begin{bmatrix}1 & 4 \\ 2 & 3\end{bmatrix}$.

(a) We compute the eigenvalues and eigenvectors as follows:

$$\begin{aligned}\det(A - \lambda I) &= 0 \\ \det\begin{bmatrix}1-\lambda & 4 \\ 2 & 3-\lambda\end{bmatrix} &= 0 \\ (1-\lambda)(3-\lambda) - 8 &= 0 \\ \lambda^2 - 4\lambda - 5 &= 0 \\ (\lambda - 5)(\lambda + 1) &= 0 \\ \lambda_1 &= 5, \quad \lambda_2 = -1\end{aligned}$$

For $\lambda_1 = 5$:

$$\begin{aligned}
(A - 5I)v &= 0 \\
\begin{bmatrix} -4 & 4 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} &= 0 \\
-4v_1 + 4v_2 &= 0 \\
v_1 &= v_2
\end{aligned}$$

Thus, one eigenvector corresponding to $\lambda_1 = 5$ is $v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

For $\lambda_2 = -1$:

$$\begin{aligned}
(A + I)v &= 0 \\
\begin{bmatrix} 2 & 4 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} &= 0 \\
2v_1 + 4v_2 &= 0 \\
v_1 &= -2v_2
\end{aligned}$$

Thus, one eigenvector corresponding to $\lambda_2 = -1$ is $v_2 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$.

(b) The eigen-decomposition of A is then:

$$\begin{aligned}
A &= PDP^{-1} \\
P &= \begin{bmatrix} 1 & -2 \\ 1 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 5 & 0 \\ 0 & -1 \end{bmatrix} \\
A &= \begin{bmatrix} 1 & -2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & -2 \\ 1 & 1 \end{bmatrix}^{-1}
\end{aligned}$$

(c) A has rank 2 since its two eigenvalues are non-zero.

(d) A is not positive definite since it has a negative eigenvalue ($\lambda_2 = -1$).

(e) A is not positive semi-definite since it has a negative eigenvalue ($\lambda_2 = -1$).

(f) A is not singular as none of its eigenvalues are zero. Its determinant is:

$$\begin{aligned}
\det(A) &= \lambda_1 \cdot \lambda_2 \\
&= 5 \cdot -1 \\
&= -5 \neq 0
\end{aligned}$$

which also shows it is not singular.

8

We have linear classifier $h(x) = \text{sign}(w^T x)$ where $w = [w_0, w_1, w_2]^T$ and $x = [1, x_1, x_2]^T$.

(a) We can prove the regions where $h(x) = +1$ and $h(x) = -1$ are separated by a line as follows:

$$\begin{aligned} h(x) &= \text{sign}(w^T x) \\ &= \text{sign}(w_0 + w_1 x_1 + w_2 x_2) \end{aligned}$$

This equation is linear in nature, and thus forms a linear decision boundary separating the two regions.
Expressing the line as $x_2 = ax_1 + b$, the slope a and intercept b in terms of w_0, w_1, w_2 are:

$$\begin{aligned} w_0 + w_1 x_1 + w_2 x_2 = 0 \\ w_2 x_2 &= -w_0 - w_1 x_1 \\ x_2 &= -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2} \therefore \\ a &= -\frac{w_1}{w_2}, \quad b = -\frac{w_0}{w_2} \end{aligned}$$

9

See attached Jupyter notebook.

10

(a) Because we are given the blood pressure for 10k patients in the dataset and corresponding data related to each patient, this is a supervised learning regression problem.

(b) The label space consists of two values, the systolic and diastolic blood pressure of each patient $(y_{\text{systolic}}, y_{\text{dystolic}}) \in \mathbb{R}^2$.

(c) The output space is the same as the label space, as the ML system aims to predict both the systolic and dystolic blood pressure values.

11

(a)

		Truth	
		Positive (Spam)	Negative (Not Spam)
Test	Positive (Spam)	1750	250
	Negative (Not Spam)	250	7750

(b) The false positive rate of the system is $250/8000 = 0.03125$.

(c) The false negative rate of the system is $250/2000 = 0.125$.

(d) The error rate of the system is $(250 + 250)/10000 = 0.05$.

(e) The precision of the system is $1750/(1750 + 250) = 0.875$.

(f) The sensitivity of the system is $1750/(1750 + 250) = 0.875$.

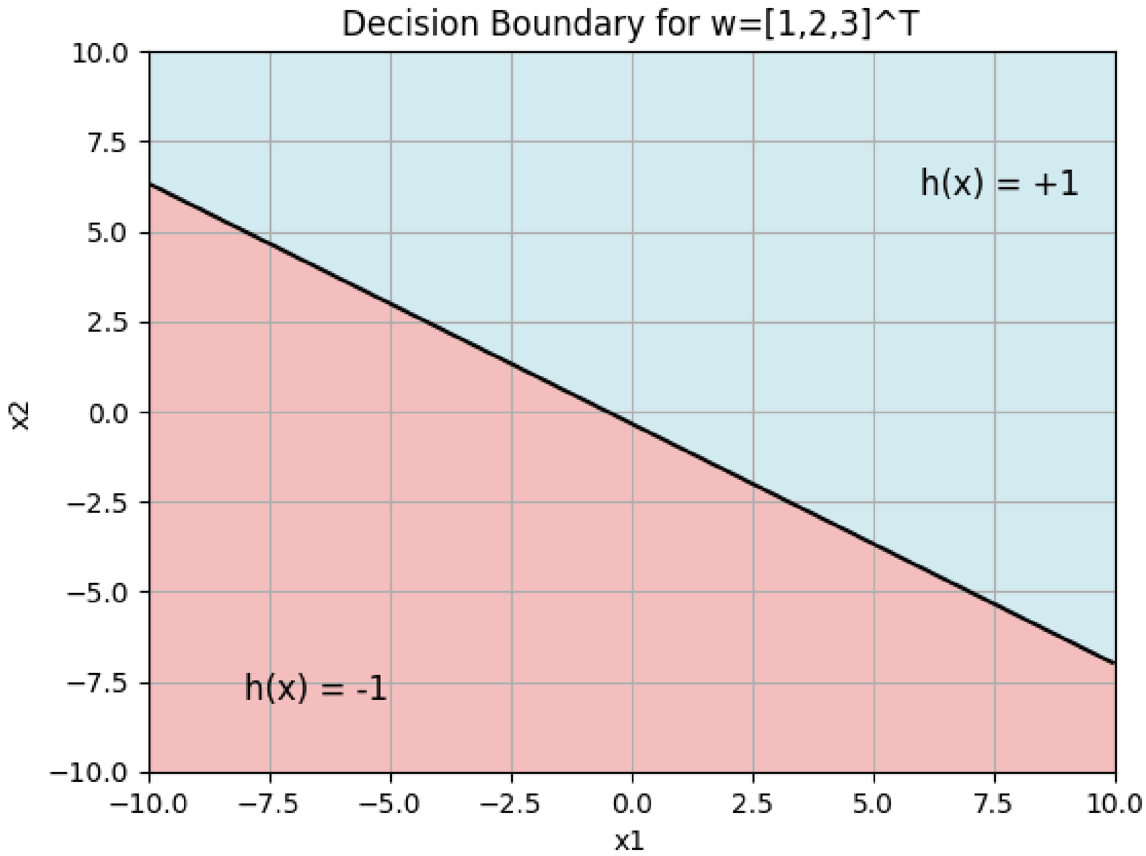
In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
```

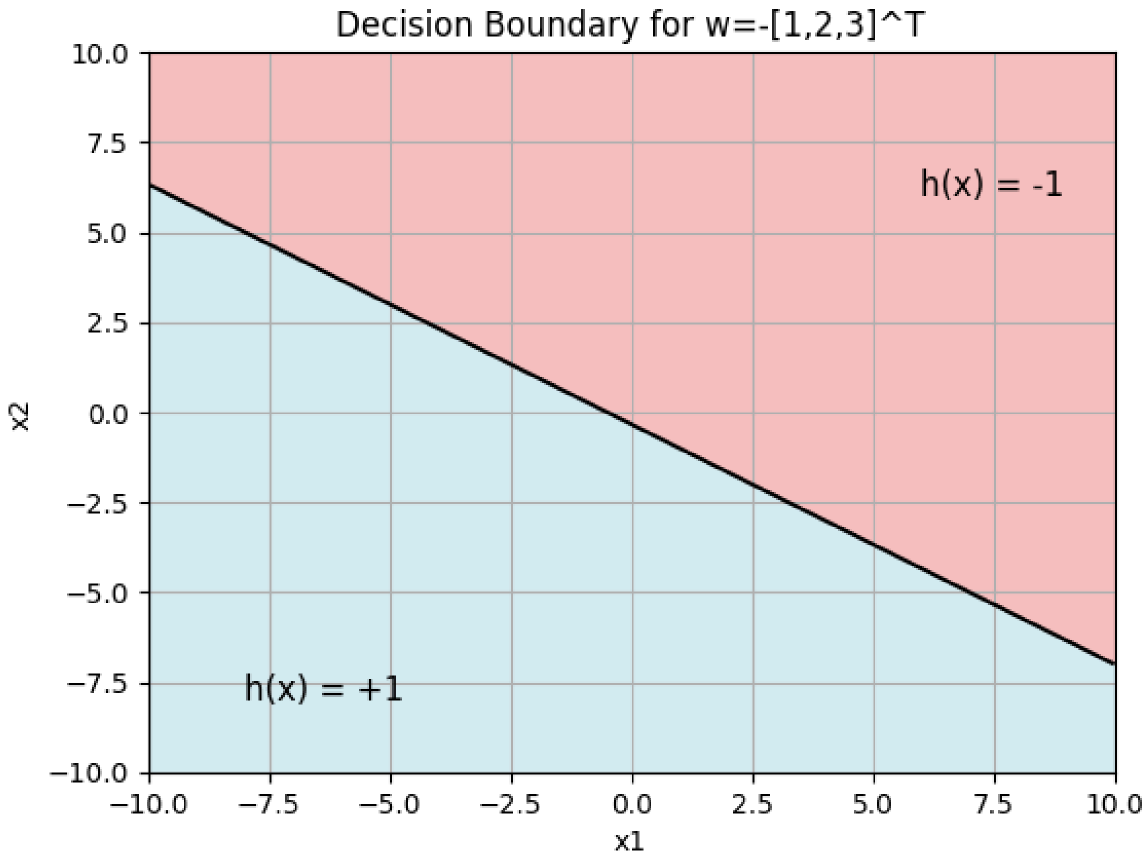
8b

In [ ]:
```
X1, X2 = np.meshgrid(np.linspace(-10, 10, 100), np.linspace(-10, 10, 100))
w1 = [1, 2, 3]
w2 = [-1, -2, -3]
Z1 = w1[0] + w1[1]*X1 + w1[2]*X2
Z2 = w2[0] + w2[1]*X1 + w2[2]*X2

plt.contour(X1, X2, Z1, levels=[0], colors="black")
plt.contourf(X1, X2, Z1, levels=[-np.inf, 0, np.inf], colors=["lightcoral", "lightb
plt.text(-8, -8, "h(x) = -1", fontsize=12, color="black")
plt.text(6, 6, "h(x) = +1", fontsize=12, color="black")
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.xlabel("x1")
plt.ylabel("x2")
plt.title("Decision Boundary for w=[1,2,3]^T")
plt.grid()
plt.show()

plt.contour(X1, X2, Z2, levels=[0], colors="black")
plt.contourf(X1, X2, Z2, levels=[-np.inf, 0, np.inf], colors=["lightcoral", "lightb
plt.text(-8, -8, "h(x) = +1", fontsize=12, color="black")
plt.text(6, 6, "h(x) = -1", fontsize=12, color="black")
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.xlabel("x1")
plt.ylabel("x2")
plt.title("Decision Boundary for w=-[1,2,3]^T")
plt.grid()
plt.show()
```


Decision Boundary for w=[1,2,3]^T
h(x) = +1
h(x) = -1
x1
x2


Decision Boundary for w=-[1,2,3]^T
h(x) = -1
h(x) = +1
x1
x2

9

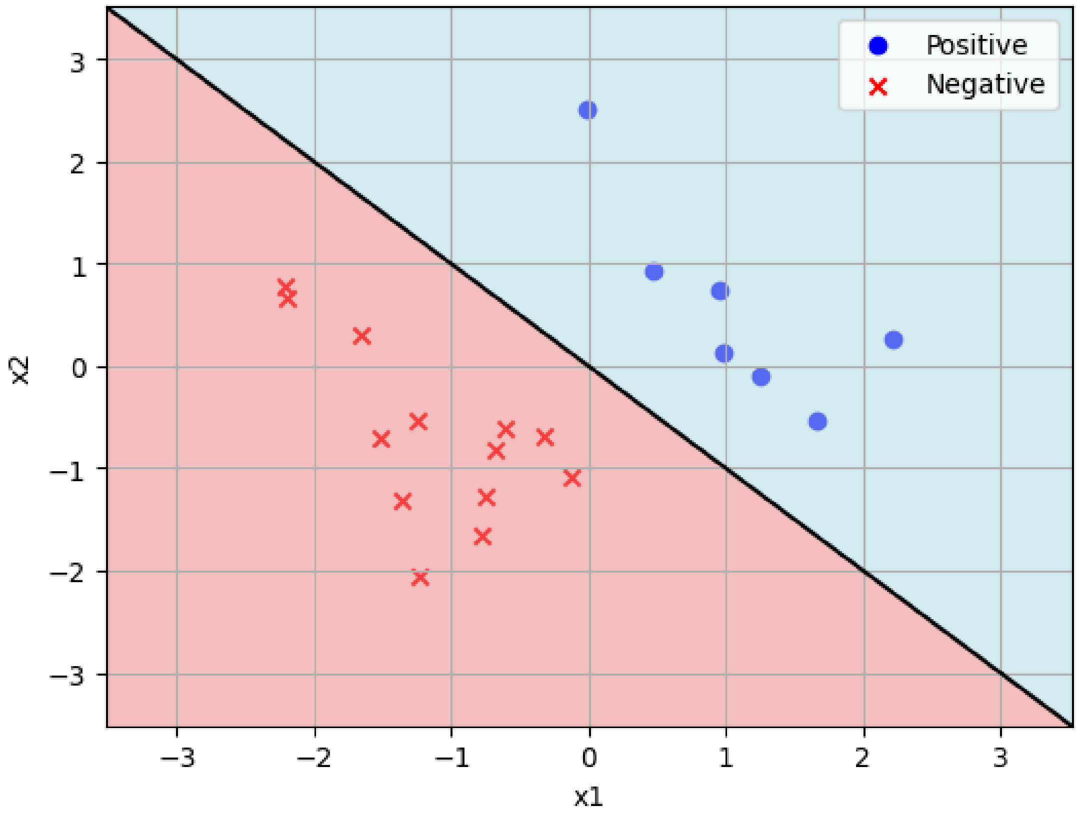
In [37]:
```
def GenerateData(margin, number):
    data = []
    i = 0
    while i < number:
        point = np.random.randn(2, 1)
        if point[0] + point[1] - margin > 0:
            data.append([point, 0])
            i += 1
        elif point[0] + point[1] + margin < 0:
            data.append([point, 1])
            i += 1
    return data

def plot_data(data):
    max_bound = 0
    for point, label in data:
        if abs(point[0]) > max_bound:
            max_bound = abs(point[0])
        if abs(point[1]) > max_bound:
            max_bound = abs(point[1])
        if label == 0:
            plt.scatter(point[0], point[1], color="blue", marker="o")
        else:
            plt.scatter(point[0], point[1], color="red", marker="x")

    plt.xlim(-max_bound-1, max_bound+1)
    plt.ylim(-max_bound-1, max_bound+1)
    plt.xlabel("x1")
    plt.ylabel("x2")
    plt.grid()
    plt.scatter([], [], color="blue", marker="o", label="Positive")
    plt.scatter([], [], color="red", marker="x", label="Negative")
    plt.legend(loc="best")
```

a

In [55]:
```
data = GenerateData(1, 20)
plot_data(data)
plt.contour(X1, X2, X1 + X2, levels=[0], colors="black")
plt.contourf(X1, X2, X1 + X2, levels=[-np.inf, 0, np.inf], colors=["lightcoral", "l
plt.show()
```



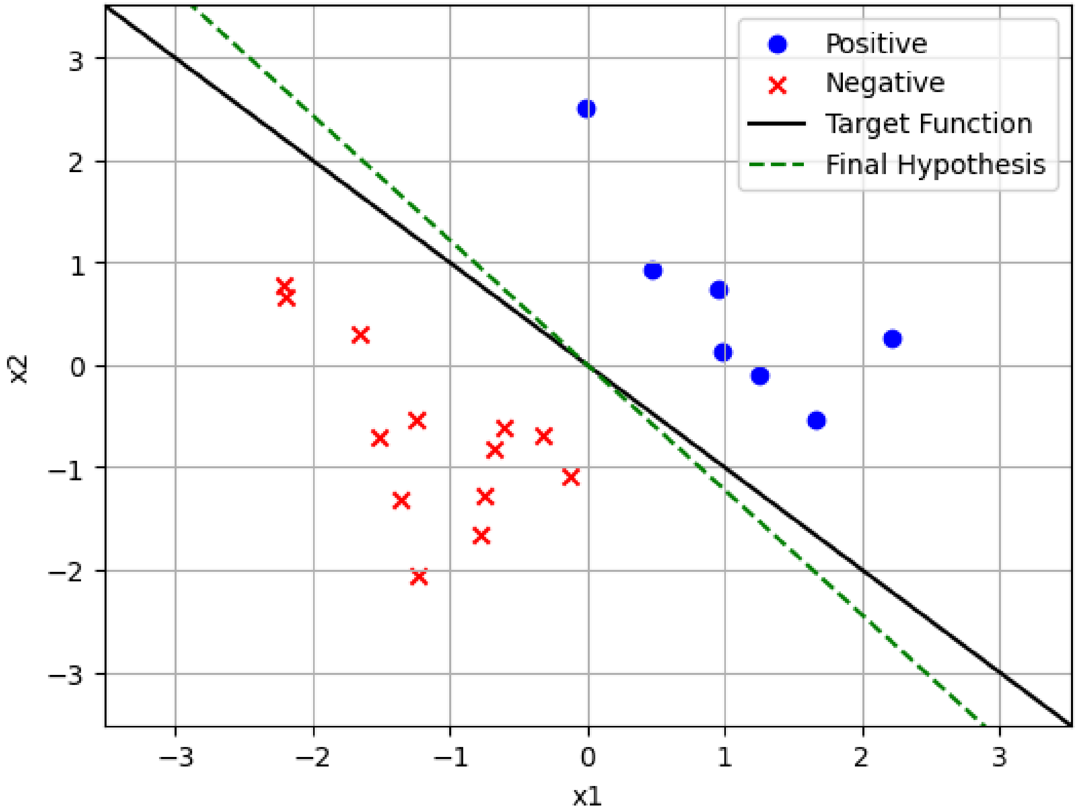
b

In [56]:
```
def perceptron_plot(data):
    w = np.zeros((3, 1))
    updates = 0
    converged = False
    while not converged:
        converged = True
        for point, label in data:
            x = np.vstack(([1], point))
            y = 1 if label == 0 else -1
            if y * (w.T @ x) <= 0:
                w += y * x
                updates += 1
                converged = False
    print(f"# updates until convergence: {updates}")

    plot_data(data)
    Z = w[0] + w[1]*X1 + w[2]*X2
    plt.plot([], [], color="black", label="Target Function")
    plt.plot([], [], color="green", linestyle="dashed", label="Final Hypothesis")
    plt.contour(X1, X2, X1 + X2, levels=[0], colors="black")
    plt.contour(X1, X2, Z, levels=[0], colors="green", linestyles="dashed")
    plt.legend(loc="best")
    plt.show()

perceptron_plot(data)
```

```
# updates until convergence: 2
```

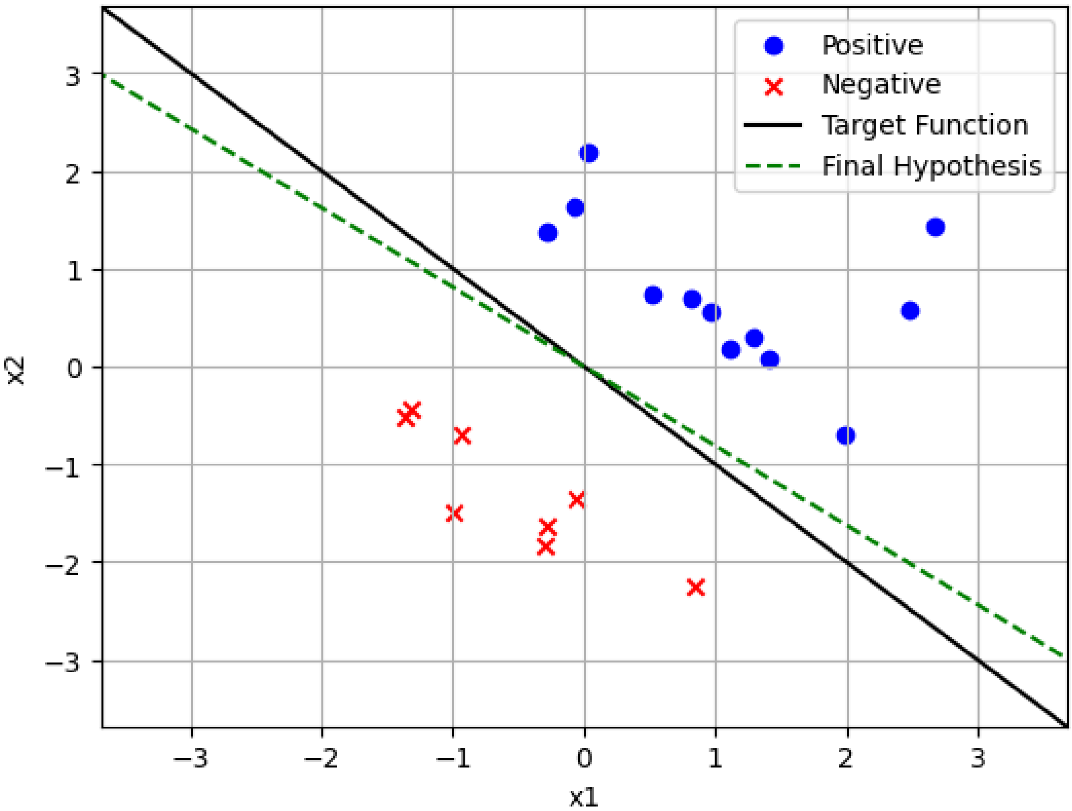


f is decently close to g. Perceptron algorithm aims to find a good balance between the two clusters and will consequently adapt to the data rather than the true target function. Because the data is generated from the normal distribution, it will always be distributed around (0, 0) hence the decision boundary will also always intersect the origin.

C

In [59]:
```
perceptron_plot(GenerateData(1, 20))
```

```
# updates until convergence: 2
```

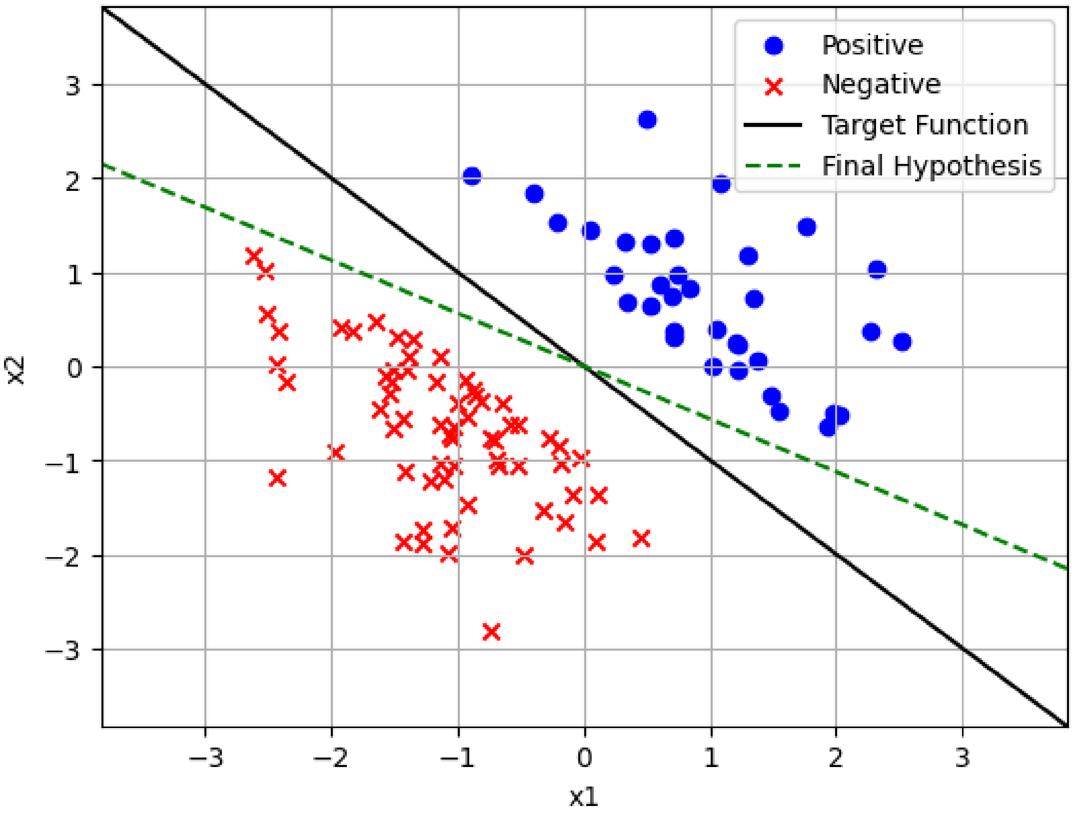


Same observation as in part (b).

d

In [60]:
```
perceptron_plot(GenerateData(1, 100))
```

```
# updates until convergence: 2
```

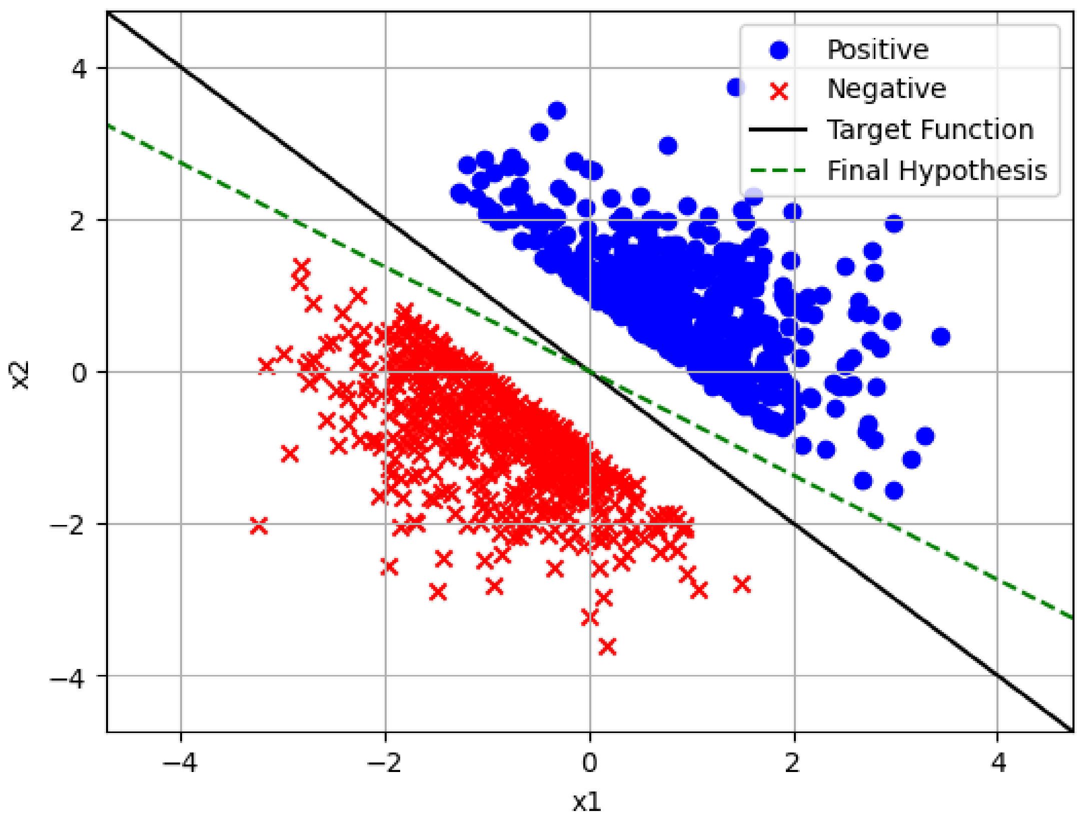


Same observation as in part (b).

e

In [61]:
```
perceptron_plot(GenerateData(1, 1000))
```

```
# updates until convergence: 4
```



Same observation as in part (b).