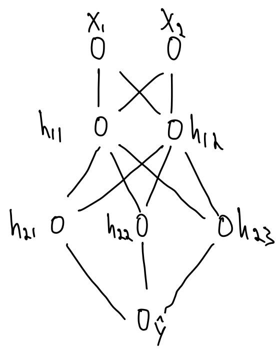
1.1 $W_0 \in R^{2\times2}$
$W_1 \in R^{3\times2}$
$W_2 \in R^{1\times3}$



a) weights: Input -> H1 = 2x2 = 4
H1 -> H2 = 3x2 = 6
H2 -> output = 1x3 = 3

total = 4 + 6 + 3 = $\underline{13}$

b) $\vec{h}_1 = W_0 \vec{x} = \begin{pmatrix} 0.1 & 0.4 \\ -0.5 & 0.6 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.1 \end{pmatrix} \text{->ReLU->} \begin{pmatrix} 0.5 \\ 0.1 \end{pmatrix}$

$\vec{h}_2 = W_1 \vec{h}_1 = \begin{pmatrix} 0.2 & -0.1 \\ 0.5 & 0.4 \\ 0.3 & -0.6 \end{pmatrix}\begin{pmatrix} 0.5 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.09 \\ 0.29 \\ 0.09 \end{pmatrix} \text{-> ReLU ->} \begin{pmatrix} 0.09 \\ 0.29 \\ 0.09 \end{pmatrix}$

$\vec{\hat{y}} = W_2 \vec{h}_2 = (0.7 \ \ -0.4 \ \ 0.9)\begin{pmatrix} 0.09 \\ 0.29 \\ 0.09 \end{pmatrix} = 0.028 \text{-> ReLU ->} \underline{0.028}$

c) $\vec{h}_1 = W_0 \vec{x} = \begin{pmatrix} 0.1 & 0.4 \\ -0.5 & 0.6 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.1 \end{pmatrix} \text{-> Sigmoid ->} \begin{pmatrix} (1+e^{-0.5})^{-1} \\ (1+e^{-0.1})^{-1} \end{pmatrix} = \begin{pmatrix} 0.6225 \\ 0.525 \end{pmatrix}$

d) $\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial W_2} = (\hat{y} - y)\vec{h}_2 = (0.028 - y)\begin{pmatrix} 0.09 \\ 0.29 \\ 0.09 \end{pmatrix}$, Shape: 3x1

1.2 a) $\vec{h}_1 = W_0 \vec{x} = \begin{pmatrix} 0.2 & 0.3 \\ 0 & -0.1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ -0.1 \end{pmatrix} \text{-> ReLU ->} \begin{pmatrix} 0.5 \\ 0 \end{pmatrix}$

$\vec{h}_2 = W_1 \vec{h}_1 = \begin{pmatrix} 0.4 & 0 \\ 0 & 0.6 \\ -0.2 & 0.1 \end{pmatrix}\begin{pmatrix} 0.5 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0 \\ -0.1 \end{pmatrix} \text{-> ReLU ->} \begin{pmatrix} 0.2 \\ 0 \\ 0 \end{pmatrix}$

$\hat{y} = W_2 \vec{h}_2 = (0.3 \ \ 0.5 \ \ 0)\begin{pmatrix} 0.2 \\ 0 \\ 0 \end{pmatrix} = 0.06$

b) $\vec{z}_1 = \begin{pmatrix} 0.5 \\ -0.1 \end{pmatrix}$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} z_i \ , \quad \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(z_i - \mu)^2$$

$$\mu = \frac{0.5 - 0.1}{2} = 0.2 \ , \quad \sigma^2 = \frac{(0.5-0.2)^2 + (-0.1-0.2)^2}{2} = 0.09$$

$$\hat{z}_i = \frac{z_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad \text{where } \epsilon \approx 0$$

$$\hat{\vec{z}}_1 = \begin{pmatrix} \frac{0.5 - 0.2}{\sqrt{0.09}} \\ \frac{-0.1 - 0.2}{\sqrt{0.09}} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\vec{h}_1 = \mathrm{ReLU}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\vec{h}_2 = W_1 \vec{h}_1 = \begin{pmatrix} 0.4 & 0 \\ 0 & 0.6 \\ -0.2 & 0.1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 0 \\ -0.2 \end{pmatrix} \to \mathrm{ReLU} \to \begin{pmatrix} 0.4 \\ 0 \\ 0 \end{pmatrix}$$

$$\hat{y} = W_2 \vec{h}_2 = (0.3 \ \ 0.5 \ \ 0)\begin{pmatrix} 0.4 \\ 0 \\ 0 \end{pmatrix} = 0.12$$

Normalization helps stabilize the network.

c) please see report at bottom of document.

2. a) $y_i(w^T x_i - b) \geq 1-\epsilon$

$$X = \begin{pmatrix} 2 & 6 \\ 6 & 2 \\ 4 & 6 \\ 5 & 3 \end{pmatrix}$$

$$1(.5(2) - .5(6) - 1) \geq 1-\epsilon$$
$$(1-3-1) \geq 1-\epsilon$$
$$-3 \geq 1-\epsilon$$
$$\epsilon \geq 4$$

$$-1(.5(6) - .5(2) - 1) \geq 1-\epsilon$$
$$-1 \geq 1-\epsilon$$
$$\epsilon \geq 2$$

$$1(.5(4) - .5(6) - 1) \geq 1-\epsilon$$
$$-2 \geq 1-\epsilon$$
$$\epsilon \geq 3$$

$$-1(.5(5) - .5(3) - 1) \geq 1-\epsilon$$
$$-1(2.5 - 1.5 - 1) \geq 1-\epsilon$$
$$\epsilon \geq 1$$

no need to use $\max(0, 1 - y_i(w^T x - b))$ since $1 - y_i(w^T x - b)$ is always positive

All constraints hold

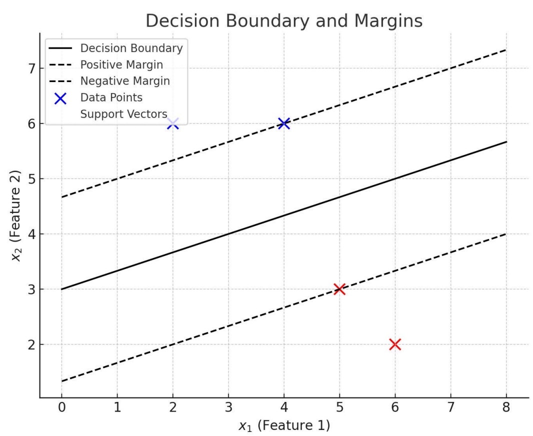
b)

```python
from sklearn.svm import SVC
import numpy as np

X = np.array([[2, 6], [6, 2], [4, 6], [5, 3]])
y = np.array([1, -1, 1, -1])

model = SVC(kernel='linear', C=1.0)
model.fit(X, y)

w_star = model.coef_[0]
b_star = model.intercept_[0]
print('w*:', w_star)
print('b*:', b_star)
```



With the coding implementation, we get $\underline{w = (-0.2, 0.6),\ b = 1.8}$

Find objective value : $y_i(w^{*T}x_i - b^*)$

$1((-0.2\ \ 0.6)\binom{2}{6} - 1.8)$ $-((-0.2\ \ 0.6)\binom{6}{2} - 1.8)$ $1((-0.2\ \ 0.6)\binom{4}{6} - 1.8)$ $-1((-0.2\ \ 0.6)\binom{5}{3} - 1.8)$

$-0.4 + 3.6 - 1.8 = 1.4$
$\epsilon = \max(0, 1-1.4)$
$\epsilon = 0$

$1.2 - 1.2 + 1.8 = 1.8$
$\epsilon = \max(0, 1 - 1.8)$
$\epsilon = 0$

$-.8 + 3.6 - 1.8 = 1$
$\epsilon = \max(0, 1-1)$
$\epsilon = 0$

$1 - 1.8 + 1.8 = 1$
$\epsilon = \max(0, 1-1)$
$\epsilon = 0$

$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\epsilon_i$

$\frac{1}{2}\sqrt{-.2^2 + 0.6^2}^2 + 0$

$\frac{1}{2}\sqrt{0.04 + 0.36}^2$

$\frac{1}{2}(0.4) = 0.2$

High C means narrow margin, few misclassifications allowed.
Low C means the opposite

c) Support vectors: (4,6), (5,3)

$y_i(w^T x_i - b)$

$1((-.2\ \ .6)\binom{4}{6} - 1.8) = -.8 + 3.6 - 1.8 = 1$ -> Lies on the margin

$-1((-.2\ \ .6)\binom{5}{3} - 1.8) = -1(-1 + 1.8 - 1.8) = 1$ -> Lies on the margin

d) a) Code for Gram: `K = np.dot(X, X.T)`

b) λ values : [0, 0, 0.19995376, 0.19995376]

c) Done for us in code

d) Verification

$w^* = \sum_{i=1}^{N}\lambda_i y_i x_i$

first two λ_i values are zero, so we have

$= (0.19995376)(1)(4,6) + (0.19995376)(-1)(5,3)$

$= (0.799815, 1.1997226) + (-0.999765, -0.59993)$

$w^* = (-0.19995, 0.5997931)$

The values match. (Slight error due to rounding)

e) Taking the derivative of the Generalized Lagrangian w.r.t w gets us $w = \sum_{i=1}^{N} \lambda_i y_i x_i$, which links primal to dual. We can see from our λ_i values which vectors are support (nonzero λ values), and these will zero out the respective $\lambda_i y_i x_i$ terms.

f) If we had a larger dataset, computing the Gram matrix would be slow as it is $O(N^2)$. It would also use $O(N^2)$ memory. The linear kernel is a good, fast option. Full-batch gradient descent would run slow. SGD would be a faster option.