University of Central Missouri
Department of Computer Science & Cybersecurity

CS5710 Machine Learning
Fall 2025

Home Assignment 3.

Student name: SAI SRAVAN CHINTALA
#700773836

Submission Requirements:

- Once finished your assignment push your source code to your repo (GitHub) and explain the work through the ReadMe file properly. Make sure you add your student info in the ReadMe file.
- Comment your code appropriately *IMPORTANT.*
- Any submission after provided deadline is considered as a late submission.

Part A: Calculation

Q1. Multi-Input Feedforward

Consider a neural network with 3 inputs x1=2, x2=1, x3=3, one hidden layer of 2 sigmoid units, and one sigmoid output.

Weights:

$$A = \begin{bmatrix} 0.2 & -0.5 \\ 0.1 & 0.3 \\ -0.2 & 0.8 \\ 0.4 & -0.6 \end{bmatrix}, \quad B = \begin{bmatrix} 0.7 \\ -1.2 \\ 0.5 \end{bmatrix}$$

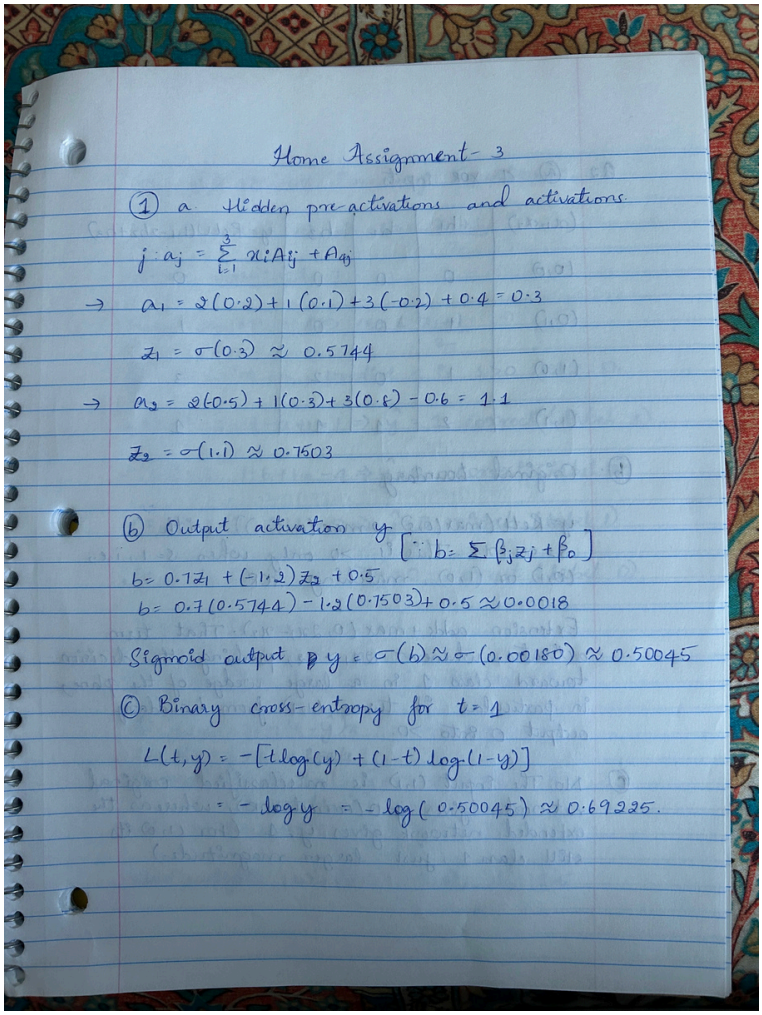
- A includes biases (last row).
- B includes bias (last element).

Tasks

a) Compute hidden pre-activations and activations.

b) Compute the output activation y.

c) If the true label is t=1, calculate the binary cross-entropy loss.



Home Assignment - 3

(1) a Hidden pre-activations and activations

$a_j = \sum_{i=1}^{3} x_i A_{ij} + A_{0j}$

$a_1 = 2(0.2) + 1(0.1) + 3(-0.2) + 0.4 = 0.3$

$z_1 = \sigma(0.3) \approx 0.5744$

$a_2 = 2(0.5) + 1(0.3) + 3(0.8) - 0.6 = 1.1$

$z_2 = \sigma(1.1) \approx 0.7503$

(b) Output activation y $[\because b = \sum \beta_j z_j + \beta_0]$

$b = 0.7 z_1 + (-1.2) z_2 + 0.5$

$b = 0.7(0.5744) - 1.2(0.7503) + 0.5 \approx 0.0018$

Sigmoid output $\hat{y} = \sigma(b) \approx \sigma(0.00180) \approx 0.50045$

(c) Binary cross-entropy for $t = 1$

$L(t, y) = -[t \log(y) + (1-t) \log(1-y)]$

$= -\log y = -\log(0.50045) \approx 0.69225.$

Q2. XOR with ReLU Network

We use the XOR ReLU network from the slides

$$h_1 = \mathbf{ReLU}(x_1 + x_2), \quad h_2 = \mathbf{ReLU}(x_1 + x_2 - 1), \quad y = \mathbf{ReLU}(h_1 - 2h_2)$$

Extend it with an additional hidden unit:

$$h_3 = \mathbf{ReLU}(2x_1 - x_2)$$

and new output:

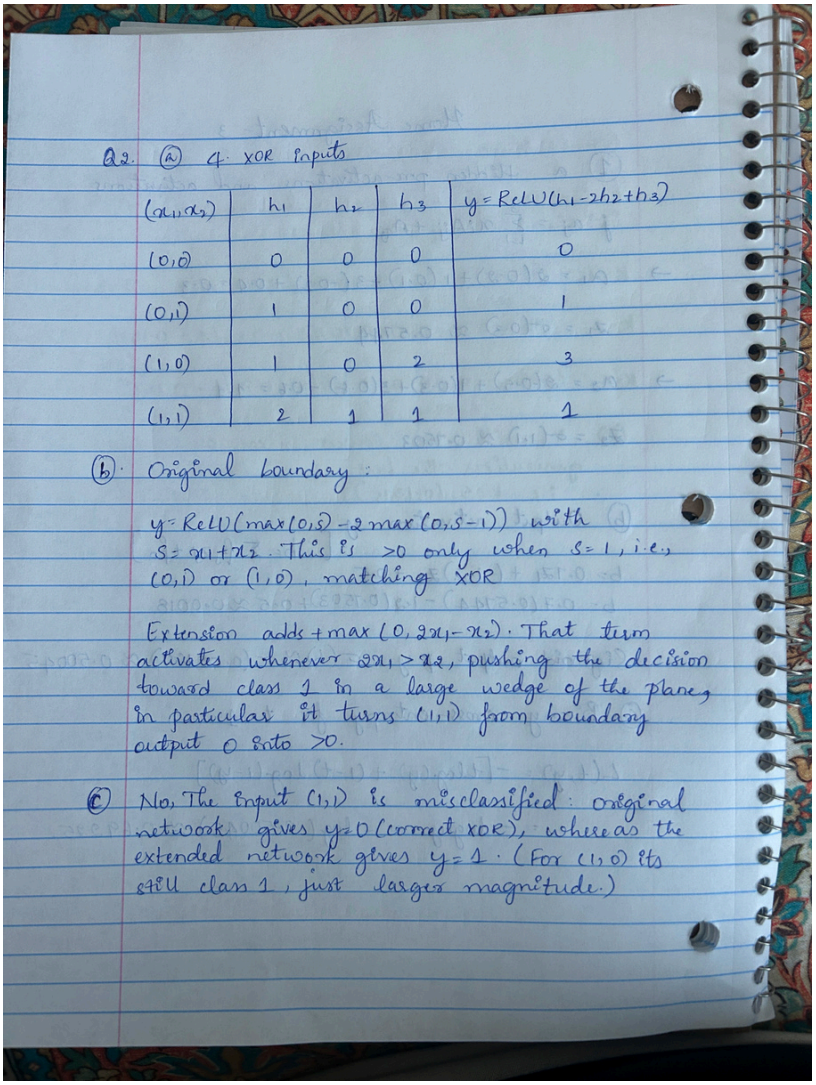
$$y = \mathbf{ReLU}(h_1 - 2h_2 + h_3)$$

Tasks

a) Compute outputs for all four XOR inputs (0,0),(0,1),(1,0),(1,1).

b) Compare the decision boundary with the original 2-hidden-unit XOR network.

c) Does this extension still compute XOR exactly? If not, which inputs differ?



Q2. (a) 4 XOR inputs

(x_1, x_2)	h_1	h_2	h_3	$y = \text{ReLU}(h_1 - 2h_2 + h_3)$
(0,0)	0	0	0	0
(0,1)	1	0	0	1
(1,0)	1	0	2	3
(1,1)	2	1	1	1

(b) Original boundary:

$y = \text{ReLU}(\max(0, s) - 2\max(0, s-1))$ with $s = x_1 + x_2$. This is >0 only when $s = 1$, i.e., (0,1) or (1,0), matching XOR

Extension adds $+\max(0, 2x_1 - x_2)$. That term activates whenever $2x_1 > x_2$, pushing the decision toward class 1 in a large wedge of the plane, in particular it turns (1,1) from boundary output 0 into >0.

(c) No, The input (1,1) is misclassified: original network gives $y = 0$ (correct XOR), whereas the extended network gives $y = 1$. (For (1,0) its still class 1, just larger magnitude.)

Q3. Decision Boundary and Misclassification

You train a perceptron with decision rule:

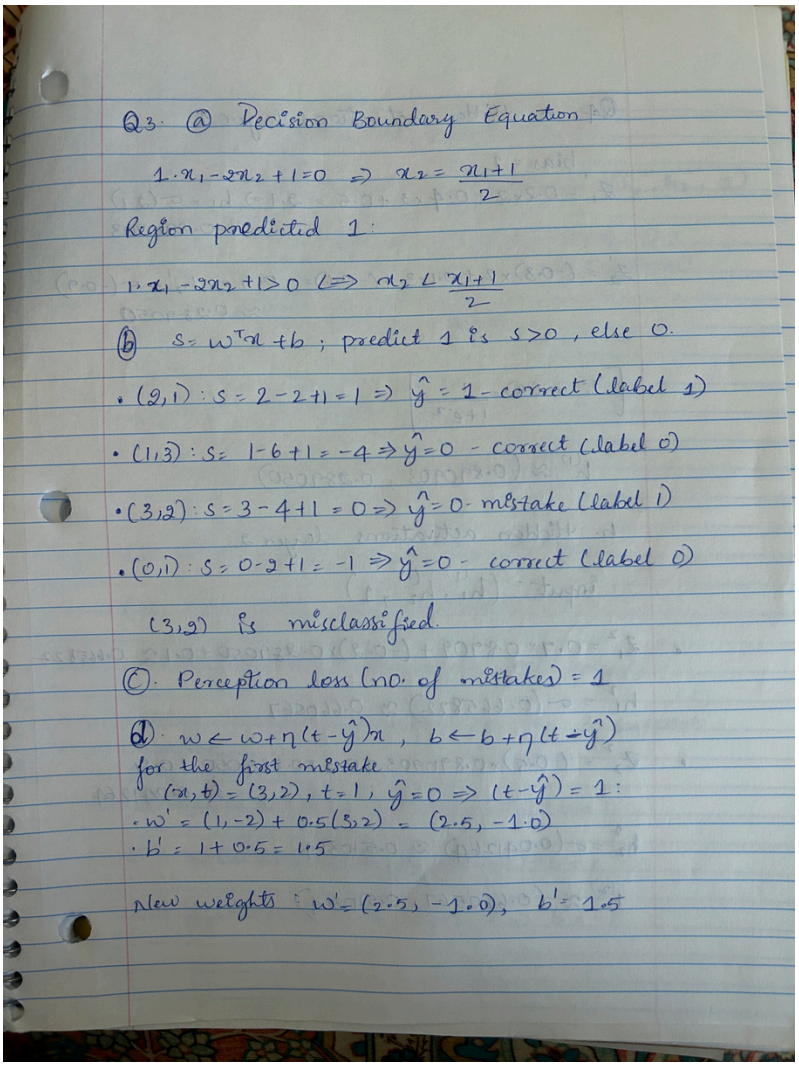
$$y = \begin{cases} 1 & \text{if } w_1 x_1 + w_2 x_2 + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

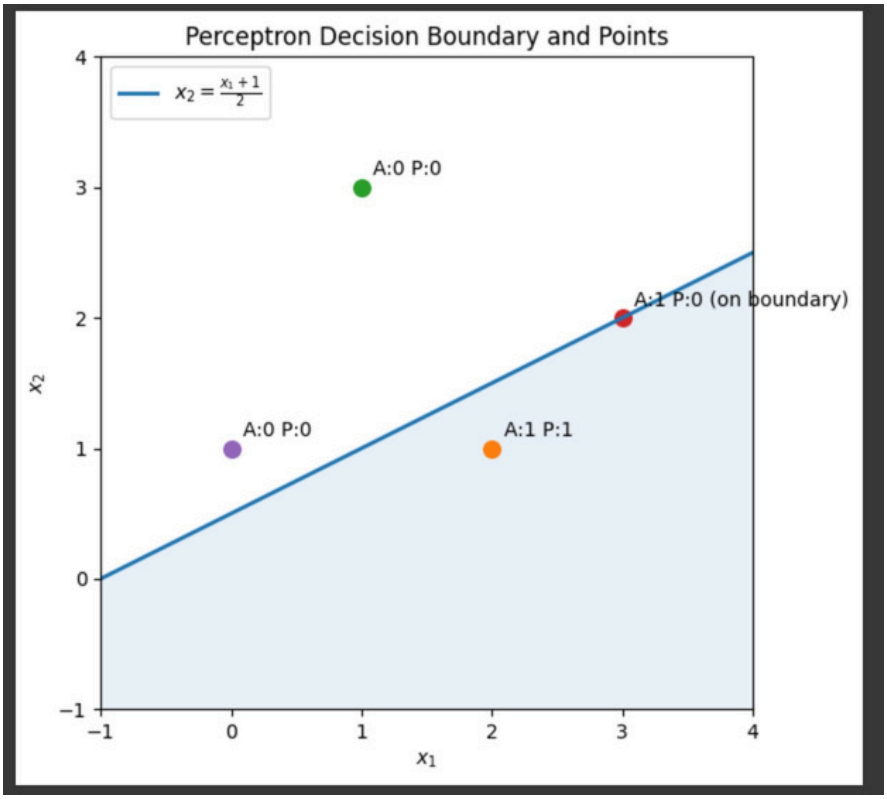
with parameters w1=1, w2=−2, b=1.

Dataset:
(2,1)→1, (1,3)→0, (3,2)→1, (0,1)→0.

Tasks

a) Write the decision boundary equation and sketch it.
b) Classify each point using the perceptron rule. Which ones are misclassified?
c) Compute the perceptron loss (number of mistakes).
d) If we apply one weight update with learning rate η=0.5 on the first mistake, what are the new weights?





Q4. Multi-Layer Forward Pass (Matrix Style)

A network has 2 hidden layers, each with 2 sigmoid units.

- Input (x1,x2)=(2,3)
- First layer weights:

$$A^{(1)} = \begin{bmatrix} 0.2 & -0.3 \\ 0.4 & 0.1 \\ 0.5 & -0.6 \end{bmatrix}$$

- Second layer weights:

$$A^{(2)} = \begin{bmatrix} 0.7 & -0.5 \\ -0.2 & 0.3 \\ 0.1 & 0.4 \end{bmatrix}$$

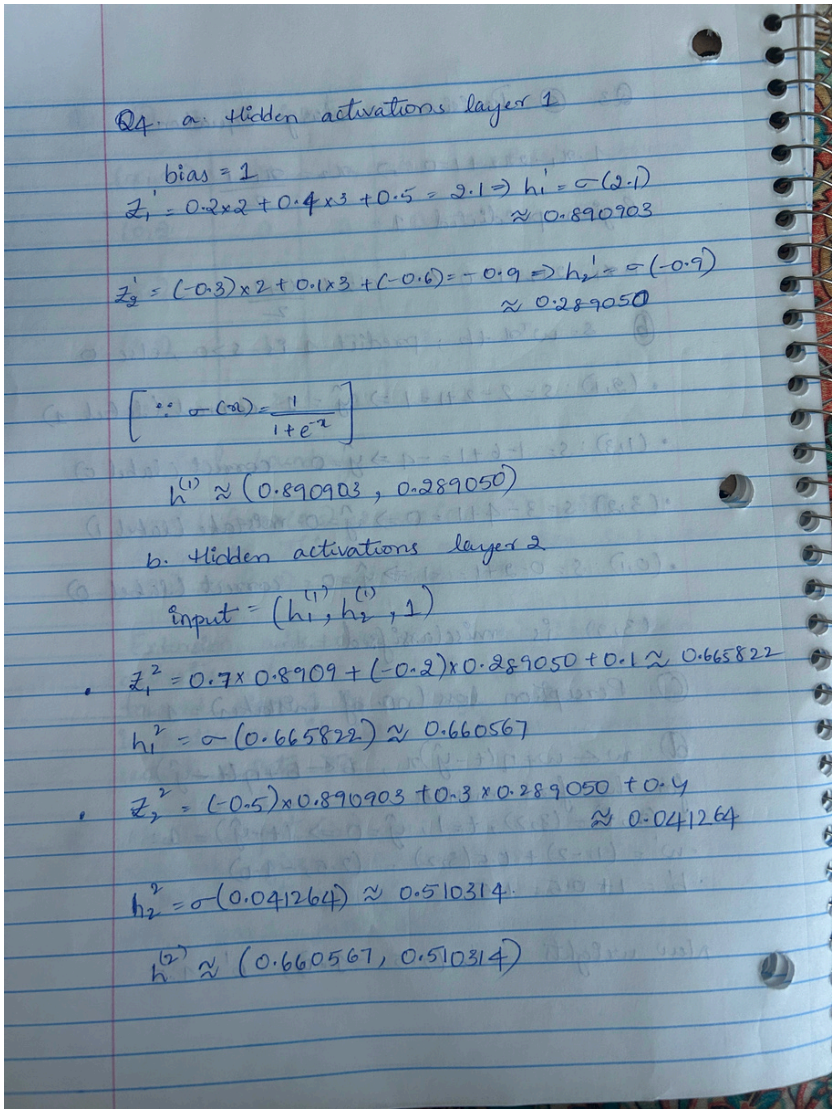
- Output weights: (1.0, −1.2, 0.3)

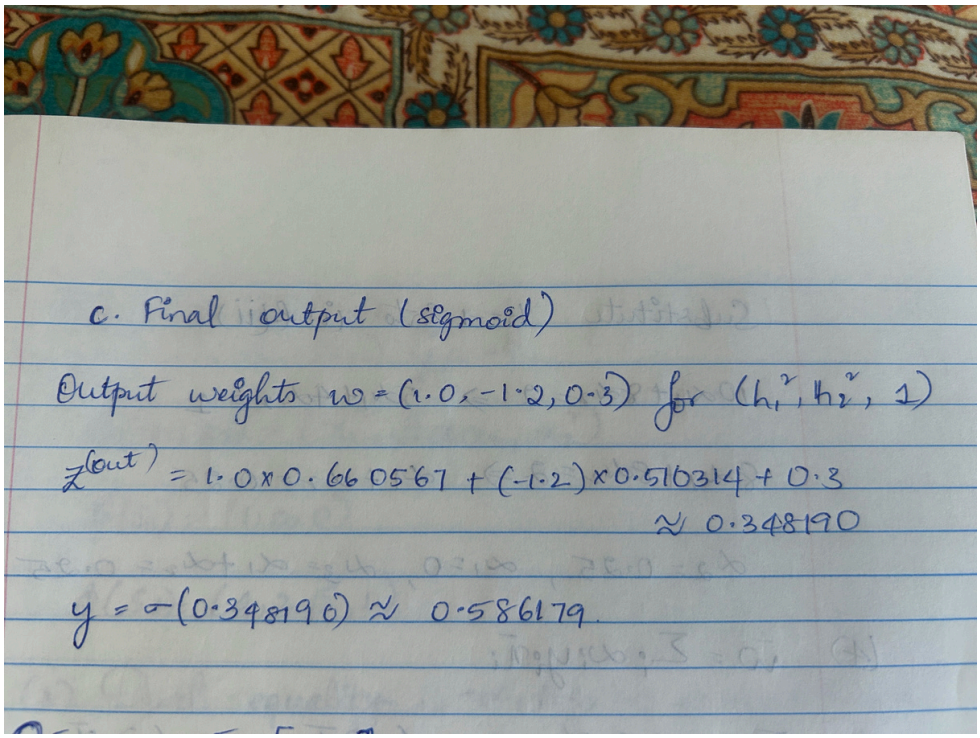
Tasks:

a) Compute hidden activations in layer 1.

b) Compute hidden activations in layer 2.

c) Compute the final output.





Q5. Linear SVM

We have a dataset in 2D:

- Positive points (y=+1): p1=(1,3), p2=(2,2)
- Negative points (y=−1): n1=(0,0)

Supposethese are the support vectors.

Tasks:

1.Augment each point with a bias term.

2.Form the dual constraint equations for α1,α2,α3 .

3. Solve for the dual variables.

4.Compute the weight vector $\tilde{w} = \sum \alpha_i y_i \tilde{x}_i$.

5.Extract www and b. Write the final separating hyperplane equation.

6.Verify that all support vectors satisfy the margin conditions.

Q5. (1) $\bar{x} = [x, 1]$

$$\bar{x}_1 = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}, \quad \bar{x}_2 = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} \quad \bar{x}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

(2) $\alpha_1 + \alpha_2 - \alpha_3 = 0 \Rightarrow \alpha_3 = \alpha_1 + \alpha_2$

(3) $x_3 = (0,0)$, $w = \alpha_1 x_1 + \alpha_2 x_2$

Let $w = (w_1, w_2) = (\alpha_1 + 2\alpha_2, 3\alpha_1 + 2\alpha_2)$

Using margin equalities on the 3 support vectors.

(i) $y_1(w^T x_1 + b) = 1 \Rightarrow (10\alpha_1 + 8\alpha_2) + b = 1$

(ii) $y_2(w^T x_2 + b) = 1 \Rightarrow (8\alpha_1 + 8\alpha_2) + b = 1$

(iii) $y_3(w^T x_3 + b) = 1 \Rightarrow 0 + b = -1 \Rightarrow b = -1$

Substitute $b=-1$ into (i) & (ii)

$10\alpha_1 + 8\alpha_2 = 2 \Rightarrow 5\alpha_1 + 4\alpha_2 = 1$

$8\alpha_1 + 8\alpha_2 = 2 \Rightarrow \alpha_1 + \alpha_2 = 0.25$

$\alpha_2 = 0.25, \quad \alpha_1 = 0, \quad \alpha_3 = \alpha_1 + \alpha_2 = 0.25$

(4) $\vec{w} = \sum_i \alpha_i y_i \vec{x}_i$

$\vec{w} = 0.(+1)\vec{x}_1 + 0.25 \times (+1)\vec{x}_2 + 0.25(-1)\vec{x}_3 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}$

(5) $w = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad b = -1$

Separating hyperplane:

$0.5x_1 + 0.5x_2 - 1 = 0 \Leftrightarrow \boxed{x_1 + x_2 = 2}$

(6) $(1,3): 0.5(1) + 0.5(3) - 1 = 1 \Rightarrow y_i(\cdot) = +1$

$(2,2): 0.5(2) + 0.5(2) - 1 = 1 \Rightarrow y_i(\cdot) = +1$

$(0,0): 0.5(0) + 0.5(0) - 1 = -1 \Rightarrow y_i(\cdot) = (-1)\cdot(-1) = +1$

All three satisfy $y_i(w^T x_i + b) = 1$ on the margin.

Q6. Nonlinear SVM

Consider two support vectors:

- s1=(1,0), y1=−1
- s2=(2,1), y2=+1

We map them using a quadraticfeature transformation:

$$\phi(x_1, x_2) = (x_1^2,\ \sqrt{2}x_1x_2,\ x_2^2)$$

Tasks:

1.Compute ϕ(s1) and ϕ(s2).

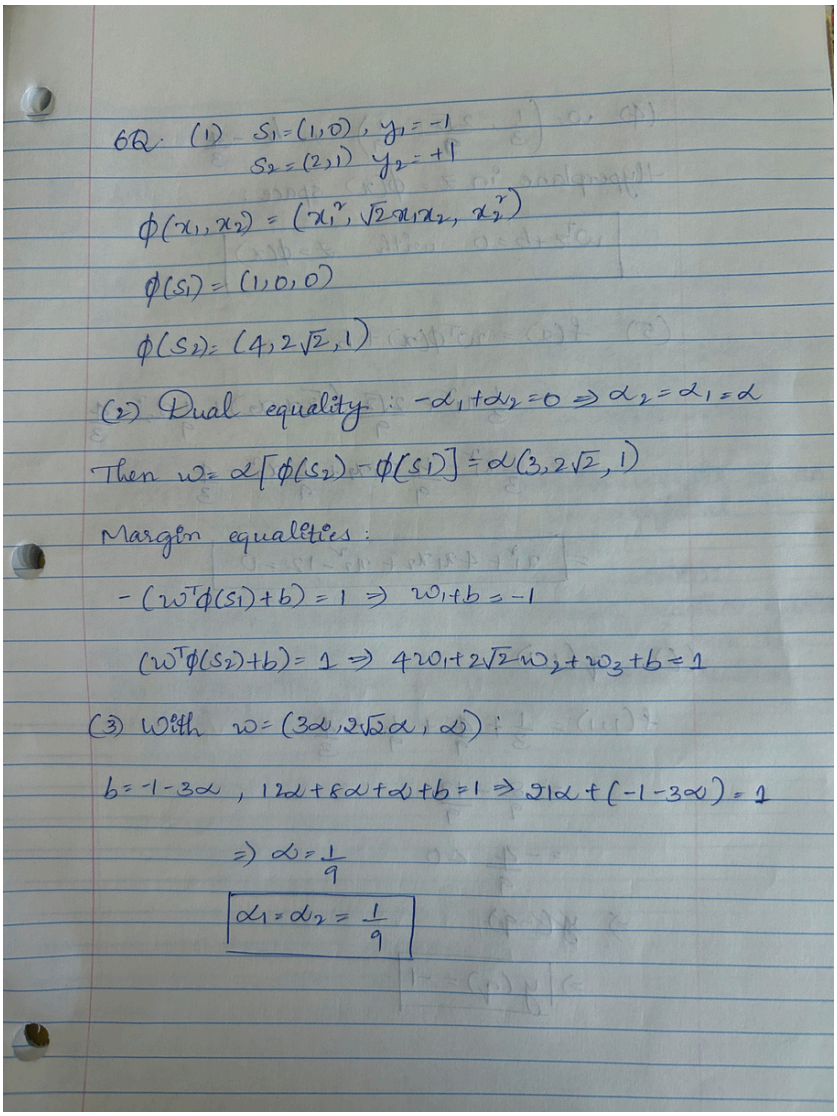
2.Write the margin equations in the transformed space using α1,α2.

3. Solve for α1,α2.

4.Compute the separating hyperplane in feature space.

5. Express the decision function $f(x) = w^\top \phi(x) + b.$

6. Using your decision function, classify the new point q=(1,1).



(4) $w = \left(\frac{1}{3}, \frac{2\sqrt{2}}{9}, \frac{1}{9}\right)$, $b = -\frac{4}{3}$

Hyperplane in $z = \phi(x)$ space:

$$\boxed{w^T z + b = 0 \quad \text{with} \quad z = \phi(x)}$$

(5) $f(x) = w^T \phi(x) + b$

$$= \frac{1}{3}x_1^2 + \frac{2\sqrt{2}}{9}(\sqrt{2}x_1x_2) + \frac{1}{9}x_2^2 - \frac{4}{3}$$

$$= \frac{1}{3}x_1^2 + \frac{4}{9}x_1x_2 + \frac{1}{9}x_2^2 - \frac{4}{3}$$

$$= \boxed{x_1^2 + 4x_1x_2 + x_2^2 - 12 = 0}$$

(6) $q(1,1)$

$$f(1,1) = \frac{1}{3} + \frac{4}{9} + \frac{1}{9} - \frac{4}{3}$$

$$= \frac{8}{9} - \frac{12}{9}$$

$$= -\frac{4}{9} < 0$$

$\Rightarrow$ ~~$y(-q)$~~

$$\Rightarrow \boxed{y(q) = -1}$$

Part B — Short-Answer

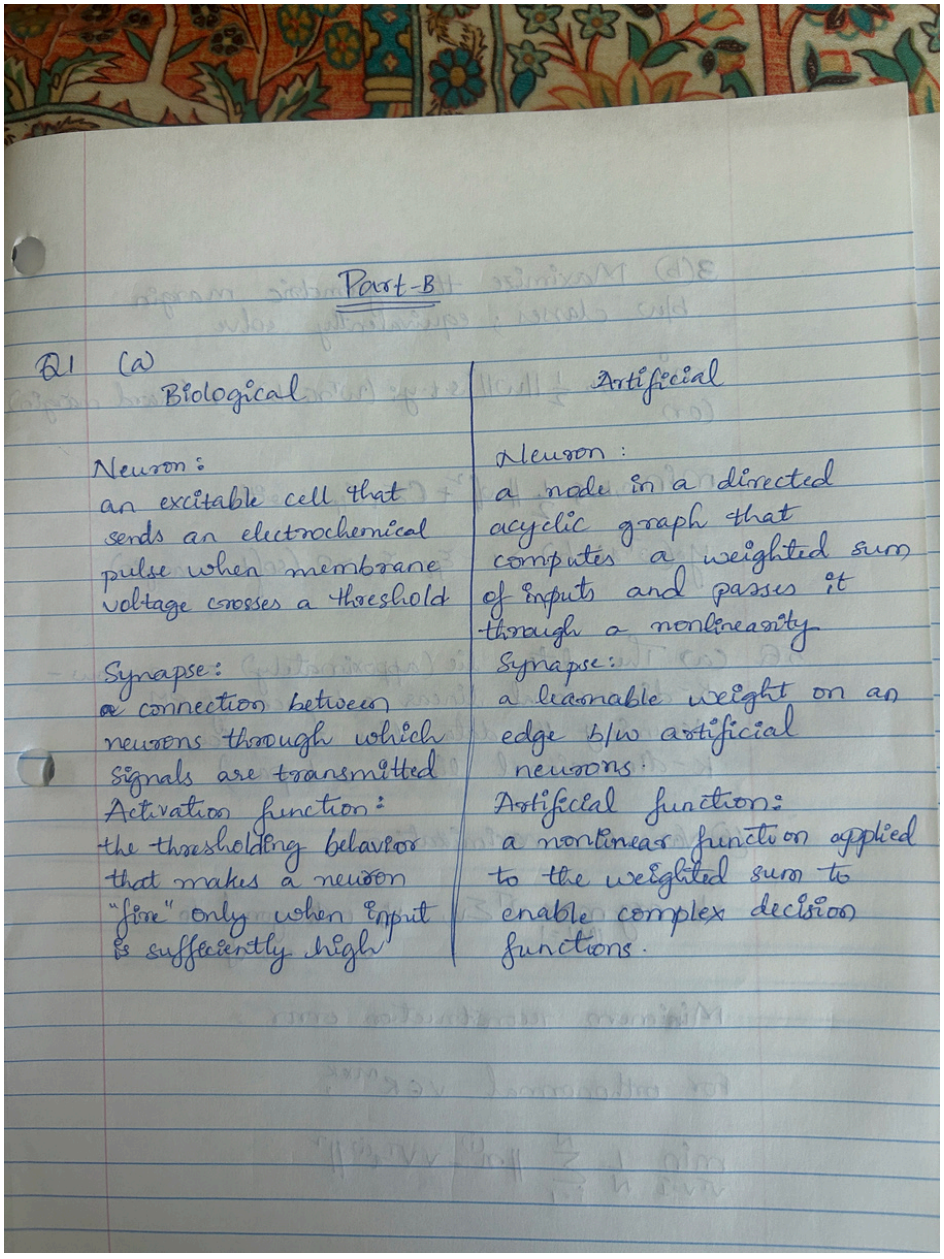
Q1. From Biological to Artificial (6 pts)

a) In one sentence each, define neuron, synapse, and activation function in the biological vs. artificial setting.

b) Give two reasons why nonlinearity is required in neural networks.

c) Briefly explain why a DAG assumption is used for feed-forward NNs.

Answer:



Part-B

Q1 (a)

Biological	Artificial
Neuron: an excitable cell that sends an electrochemical pulse when membrane voltage crosses a threshold	Neuron: a node in a directed acyclic graph that computes a weighted sum of inputs and passes it through a nonlinearity
Synapse: a connection between neurons through which signals are transmitted	Synapse: a learnable weight on an edge b/w artificial neurons.
Activation function: the thresholding behavior that makes a neuron "fire" only when input is sufficiently high	Artificial function: a nonlinear function applied to the weighted sum to enable complex decision functions.

(b) To represent non-linear decision boundaries—a single linear unit cannot solve problems like XOR; hidden nonlinear units make it separable.

To approximate complex functions / learn rich features—with nonlinear activations, NNs are universal function approximators and hidden units can learn nonlinear feature combinations.

(C) A feed-forward NN is defined as a DAG so its computation is a well-defined, acyclic pipeline from inputs to outputs—i.e., a differentiable function you can evaluate forward and optimize by backprop—without feedback loops

Q2. Architecture & Capacity (8 pts)

a) Define depth and width.
b)Whataretwodistinctroleshiddenunits might learn (e.g., feature selection vs. projection)?
c)Contrastthecases $D < M$ and $D > M$ in a single-hidden-layer network: what representationaltrade-offsdotheyimply?
d)Statethe universal approximationidea for 1-hidden-layer NNs (informally).

Answers:

(a) Depth: the number of hidden layers between input and output (how many learned transformations are stacked). Width: the number of units in a hidden layer.

(b) Lower-dimensional projec5on / feature selec5on (compress or pick the most useful features). **Nonlinear combina5ons / higher-dimensional projec5on** (compose features to make data more separable).

(c) D<M: acts as a bottleneck/compression → encourages compact representations and noise reduction; risk of underfittingif too small.
D>M: expands into a richer/higher-dimensional feature space → more capacity and easier separability; risk of overfittingand higher compute.
(d) A neural network with one hidden layer and suitable nonlinear activations (e.g., sigmoid) can approximate any continuous function on a compact domain arbitrarily well given enough hidden units.

Q3. Perceptron vs. SVM (6 pts)

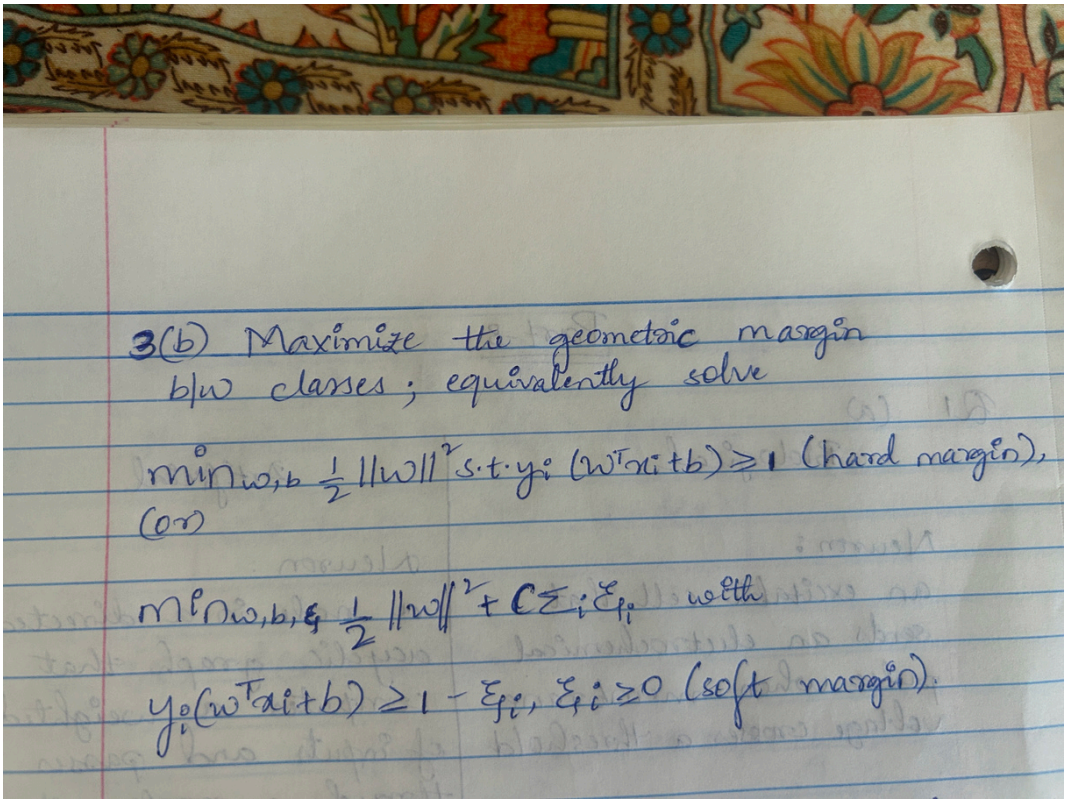
a) What does the perceptron algorithm optimize?
b) What is the main optimization goal of SVM?
c) Why does SVM typically generalize better than a perceptron? Give one real-world scenario where this difference matters.

Answers:

(a)A perceptron finds any separating hyperplane by making mistake-driven updates that reduce training misclassifications; it does not explicitly maximize a margin.

(c) Because SVM maximizes margin and uses L2 regularization, it controls model capacity and has a unique convex optimum that depends only on a subset of points (support vectors); the

perceptron doesn't optimize margin and is more sensitive to noisy/outlier data. In high-dimensional text/spam classification, SVMs commonly outperform perceptrons thanks to large-margin separation with regularization.



Q4. Margins and Support Vectors (6 pts)

a) Define the margin of a dataset.
b) Explain the role of support vectors in determining the separating hyperplane.
c) Why does maximizing the margin lead to robustness against noise?

Answers:

(a)Foraseparating hyperplane f(x)=wTx+b, the (geometric) margin is the minimum distance from any training point to the hyperplane. Under the SVM's canonical scaling yi (wTxi +b)≥1, the margin on each side is 1/ ||w|| and total margin width is 2 / ||w||.

(b) Support vectors are the points that lie on the margin and satisfy yi (wTxi +b)=1; they alone have non-zero dual weights and thus determine the hyperplane via w=∑i αi yi xi and set b. Removing non-SV points does not change the solution; moving an SV does.

(c) Maximizing the margin increases the perturbation needed to flip a label (larger safety buffer), controls capacity via ||w|| (regularization), and thus reduces overfitting—making the classifier less sensitive to noise and small input variations.

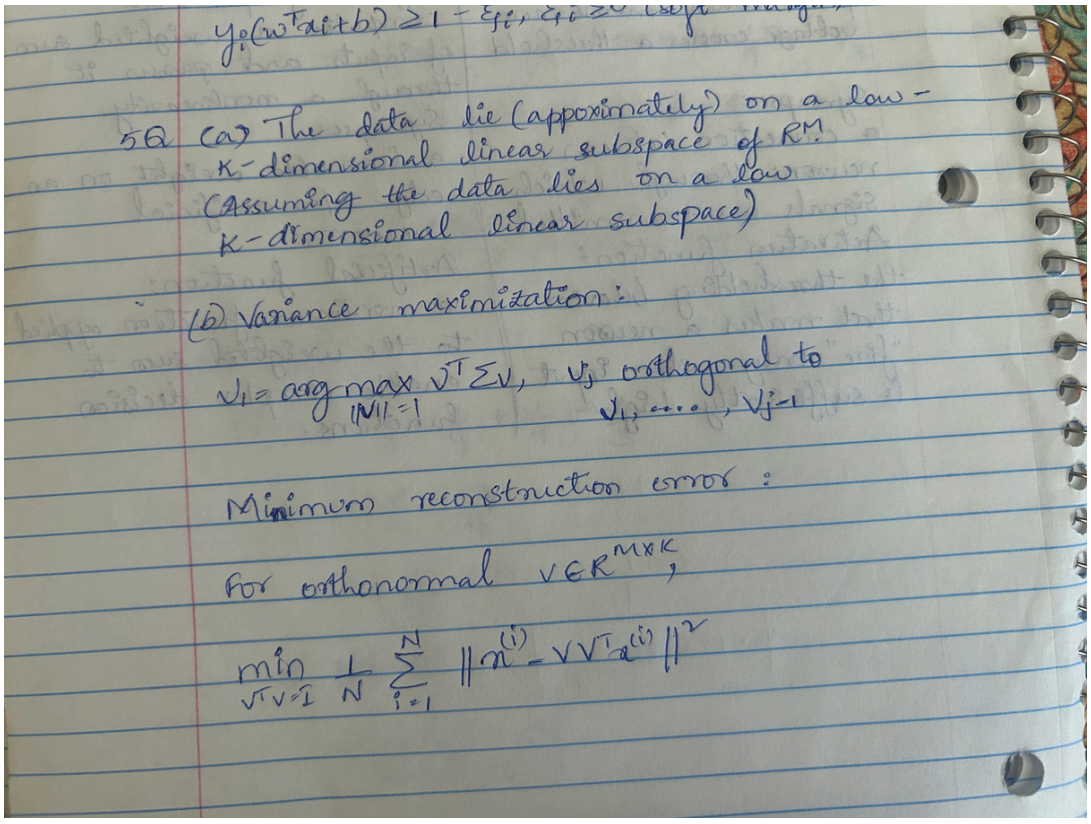
Q5. PCA Concepts (8 pts)

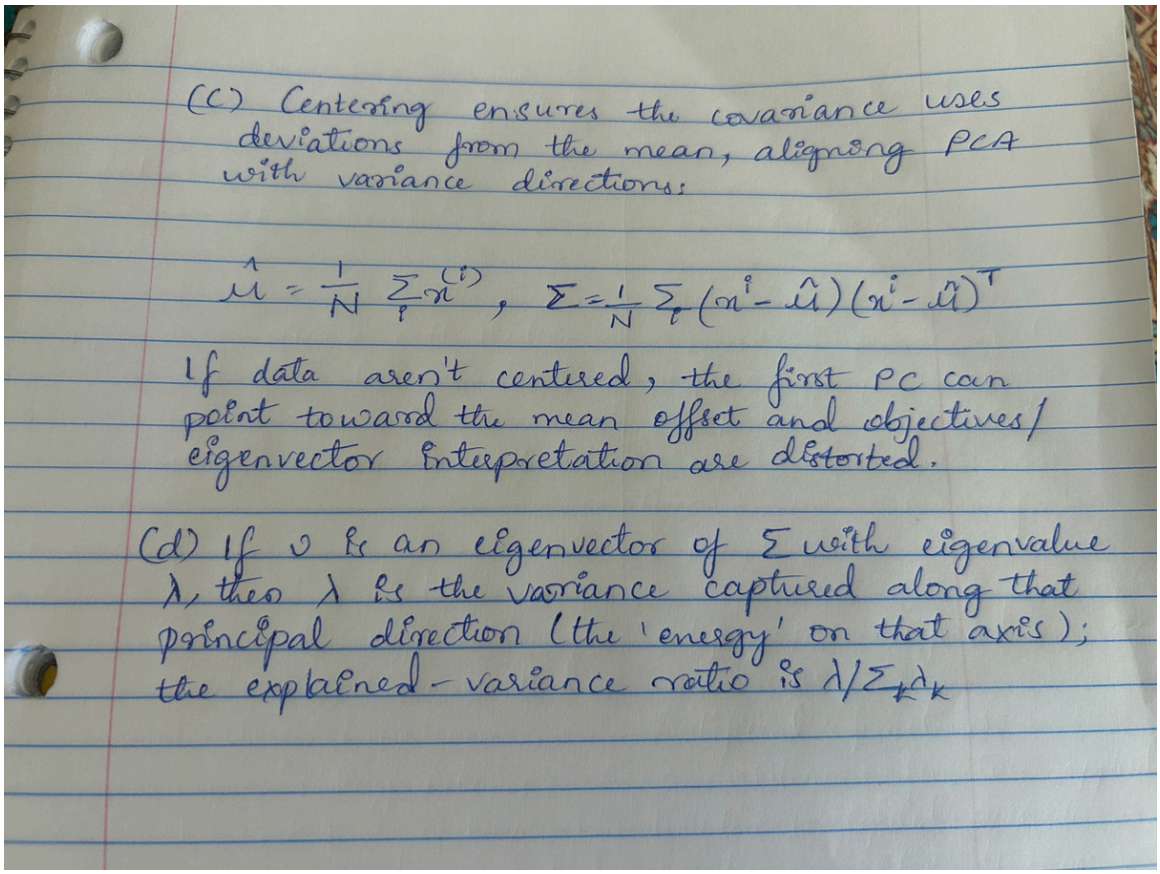
a) What assumption does PCA make about the structure of the data?

b) State two equivalent definitions of PCA.

c) Why must the data be centered before applying PCA?

d) Explain what the eigenvalue associated with a principal component represents.





Q6. PCA vs. Random Projection (5 pts)

a) How does PCA choose directions compared to random projection?
b) Give one advantage of PCA over random projection, and one disadvantage.

Answers

: (A)PCA:chooses data-dependentaxes—the eigenvectors of the sample covariance — equivalently the directions that maximize variance or minimize reconstruction error.

Random projection: chooses directions at random (e.g., sample a matrix V with i.i.d. Gaussian entries and project u=Vx) independent of thedata.

(B) Advantage (PCA): it preserves the most variance / least reconstruction error by construction; the eigenvalue on each principal component quantifies the variability captured.
Disadvantage (PCA): it requires eigendecomposition/SVD (potentially costly on large, high-dimensional data), whereas random projection is just sampling V and multiplying.