University of Central Missouri

Department of Computer Science & Cybersecurity

CS5710 Machine Learning

Fall 2025

Home Assignment 3.

Student name: SANTHOSH REDDY KISTIPATI

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.

Ans:**Weights:**
A = [[0.2, -0.5], [0.1, 0.3], [-0.2, 0.8], [0.4, -0.6]]
B = [0.7, -1.2, 0.5]

Step 1: Hidden pre-activation
z = A^T * [2, 1, 3, 1]

z1 = 0.22 + 0.11 + (-0.2)3 + 0.41 = 0.3
z2 = (-0.5)2 + 0.31 + 0.8*3 + (-0.6)*1 = 1.1

So z = [0.3, 1.1]

b) Compute the output activation y

Ans: h = sigmoid(z) = [0.5744, 0.7503]

Step 3: Output pre-activation
u = 0.7h1 + (-1.2)h2 + 0.5
u = 0.70.5744 - 1.20.7503 + 0.5 = 0.0018

Output = sigmoid(u) = 0.5004

Final Answer:

Hidden z = [0.3, 1.1], h = [0.5744, 0.7503],

Output y = 0.5004

Q2. XOR with ReLU Network

We use the XOR ReLU network from the slides

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

Extend it with an additional hidden unit:

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

and new output:

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

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

Ans:

Formulas:

h1 = ReLU(x1 + x2)
h2 = ReLU(x1 + x2 - 1)
h3 = ReLU(2x1 - x2)
y = ReLU(h1 - 2h2 + h3)

For all XOR inputs:

(0,0): h1=0, h2=0, h3=0 → y=0
(0,1): h1=1, h2=0, h3=0 → y=1
(1,0): h1=1, h2=0, h3=2 → y=3
(1,1): h1=2, h2=1, h3=1 → y=1

Results: [0, 1, 3, 1]

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

Ans: **Compared to the original 2-unit XOR network:**

Now an extra line 2x1 = x2 (new hinge) changes the boundary.

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

Ans: **The new net does NOT compute XOR exactly because (1,1) should give 0 but gives 1.**

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.

Ans : s(x)=0⟹x1−2x2+1=0⟺x1=2x2−1⟺x2=x1+1/2

- **Line passes through (−1,0) and (0,0.5)**

- **Normal vector (1,−2)**

  positive side (predict y^=1) is where x1>2x2−1

b) Classify each point using the perceptron rule. Which ones are misclassified?

Ans :**Compute score s = x1 - 2x2 + 1**

(2,1): s=1 → y=1 ✔
(1,3): s=-4 → y=0 ✔
(3,2): s=0 → y=0 ✖ (should be 1)
(0,1): s=-1 → y=0 ✔

c) Compute the perceptron loss (number of mistakes).

Ans:**Only (3,2) misclassified.**

Perceptron loss = 1

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.

Ans: **Input: (x1,x2)=(2,3)**
First layer weights:
A1 = [[0.2, -0.3], [0.4, 0.1], [0.5, -0.6]]
Second layer weights:
A2 = [[0.7, -0.5], [-0.2, 0.3], [0.1, 0.4]]
Output weights: B = [1.0, -1.2, 0.3]

Layer 1:
z1 = [2,3,1]*A1 = [2.1, -0.9]
h1 = sigmoid(z1) = [0.8909, 0.2891]

b) Compute hidden activations in layer 2.

Ans: **Layer 2:**

h1+ = [0.8909, 0.2891, 1]

z2 = h1+*A2 = [0.6658, 0.0413]

h2 = sigmoid(z2) = [0.6606, 0.5103]

c) Compute the final output.

Ans:

Output:
h2+ = [0.6606, 0.5103, 1]
u = h2+*B = 0.3465
y = sigmoid(0.3465) = 0.5862

Final: y = 0.5862

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)

Suppose these are **the support vectors.**

Tasks:

1. Augment each point with a bias term.

   Ans: We'll keep bias separate for clarity, but note the augmented vectors if needed:

   p1~ = [1,3,1], p2~ = [2,2,1], n1~ = [0,0,1]

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

   Ans: For support vectors, y_i * (w · x_i + b) = 1:

- **p1: +1 * (w1*1 + w2*3 + b) = 1 → w1 + 3w2 + b = 1**
- **p2: +1 * (w1*2 + w2*2 + b) = 1 → 2w1 + 2w2 + b = 1**
- **n1: −1 * (w1*0 + w2*0 + b) = 1 → −b = 1 → b = −1**

Plug b = −1 into the first two:

- **w1 + 3w2 − 1 = 1 → w1 + 3w2 = 2**
- **2w1 + 2w2 − 1 = 1 → 2w1 + 2w2 = 2 → w1 + w2 = 1**

Solve:

- **From w1 + w2 = 1 → w1 = 1 − w2**
- **Substitute into w1 + 3w2 = 2: (1 − w2) + 3w2 = 2 → 1 + 2w2 = 2 → w2 = 0.5**
- **Then w1 = 0.5**

Result:

w = (0.5, 0.5), b = −1

3. Solve for the dual variables.
   Ans:Hard-margin relations:

- **w = Σ α_i y_i x_i**
- **Σ α_i y_i = 0, α_i ≥ 0**
  Let α1, α2, α3 correspond to p1, p2, n1.

w = α1*(+1)*(1,3) + α2*(+1)*(2,2) + α3*(−1)*(0,0)
= (α1 + 2α2, 3α1 + 2α2) = (0.5, 0.5)

Solve:

- **α1 + 2α2 = 0.5**
- **3α1 + 2α2 = 0.5**
  Subtract → 2α1 = 0 → α1 = 0
  Then 2α2 = 0.5 → α2 = 0.25
  Constraint Σ α_i y_i = 0 → α1 + α2 − α3 = 0 → α3 = 0.25

Alphas:

- **α1 = 0**
- **α2 = 0.25**
- **α3 = 0.25**

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

   Ans: Already verified above → w = (0.5, 0.5)

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

   Ans: **w = (0.5, 0.5), b = −1**

   Hyperplane: 0.5*x1* + *0.5*x2 − 1 = 0 → x1 + x2 = 2

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

   Ans :Evaluate y_i*(w·x_i + b):

- p1: +1*(0.51 + 0.53 − 1) = +1*(1) = 1 ✓
- p2: +1*(0.52 + 0.52 − 1) = +1*(1) = 1 ✓
- n1: −1*(0.50 + 0.50 − 1) = −1*(−1) = 1 ✓

Extras (nice to include):

- Margin (geometric) = 1 / ||w|| = 1 / sqrt(0.5^2 + 0.5^2) = 1 / sqrt(0.5) = sqrt(2) ≈ 1.414
- Margin band width (distance between the two class margins) = 2 / ||w|| = 2*sqrt(2) ≈ 2.828
- Decision function: f(x) = w·x + b = 0.5x1 + 0.5x2 − 1
- Classify by sign(f(x)); boundary when f(x)=0 → x1 + x2 = 2

Q6. Nonlinear SVM

Consider two support vectors:

- s1=(1,0), y1=−1
- s2=(2,1), y2=+1

We map them using a quadratic feature transformation:

$$\Phi_1\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{cases} \begin{pmatrix} 4 - x_2 + |x_1 - x_2| \\ 4 - x_1 + |x_1 - x_2| \end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} > 2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \text{otherwise} \end{cases}$$

Tasks:

1. Compute $\phi(s1)$ and $\phi(s2)$.

   Ans :Support vectors:

- s1 = (1,0), y1 = −1
- s2 = (0,1), y2 = +1

Feature map:

- If sqrt(x1^2 + x2^2) > 2, then Φ(x) = (4 − x2 + |x1 − x2|, 4 − x1 + |x1 − x2|)
- Otherwise, Φ(x) = (x1, x2)

Since ||s1|| = 1 and ||s2|| = 1, both ≤ 2 → use Φ(x) = (x1, x2).

1) Compute φ(s1) and φ(s2)

φ(s1) = (1,0)

φ(s2) = (0,1)

2. Write the margin equations in the transformed space using α1,α2.
   Ans: Primal margin equalities:

- y1*(w·φ1 + b) = 1 → −1*(w1*1* + *w2*0 + b) = 1 → w1 + b = −1
- y2*(w·φ2 + b) = 1 → +1*(w1*0* + *w2*1 + b) = 1 → w2 + b = +1

Dual constraints:

- w = Σ α_i y_i φ_i = (−α1, +α2)
- Σ α_i y_i = 0 → −α1 + α2 = 0 → α1 = α2

3. Solve for α1,α2.
   Ans: From w = (−α1, +α2) and the margin equations:

   Use w1 + b = −1 and w2 + b = +1.
   Pick b from these once w is set. Minimal-norm w that separates φ1 and φ2 with unit margins is w = (−1, +1).

Thus:

$-\alpha 1 = -1 \rightarrow \alpha 1 = 1$

$+\alpha 2 = 1 \rightarrow \alpha 2 = 1$
Check $\Sigma\ \alpha\_i\ y\_i = -1 + 1 = 0$ ✓

4. Compute the separating hyperplane in feature space.
Ans:
$w = (-1, +1)$, $b = 0$ (from $w1 + b = -1$ and $w2 + b = 1$)

Hyperplane: $w \cdot z + b = 0 \rightarrow (-1)z1 + (1)z2 = 0 \rightarrow z2 - z1 = 0 \rightarrow z1 = z2$
(In original coordinates, since here $z = \varphi(x) = x$, boundary is $x1 = x2$ near the origin.)

5. Express the decision function $f(x) = w^\top \phi(x) + b.$
Ans:
$(x) = w^\wedge T\ \varphi(x) + b$

For points with $\mathrm{sqrt}(x1^\wedge 2 + x2^\wedge 2) \leq 2$: $\varphi(x) = (x1, x2) \rightarrow f(x) = -x1 + x2$

For points outside the circle (norm > 2): $\varphi(x) = (4 - x2 + |x1 - x2|, 4 - x1 + |x1 - x2|)$
so $f(x) = -(4 - x2 + |x1 - x2|) + (4 - x1 + |x1 - x2|) + 0$
$= -4 + x2 - |x1 - x2| + 4 - x1 + |x1 - x2|$
$= x2 - x1$
Interesting: the mapping was chosen so $f(x) = x2 - x1$ everywhere, so the same linear rule applies.

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

Ans:Norm <= 2 $\rightarrow$ use $\varphi(x) = (x1, x2)$

- $f(q) = -1 + 1 = 0 \rightarrow$ on the decision boundary (tie).
By convention, sign(0) can be treated as 0/boundary; with any infinitesimal perturbation, class depends on whether $x2 - x1$ is positive or negative.

Extras (good to add):

- $||w|| = \mathrm{sqrt}((-1)^\wedge 2 + 1^\wedge 2) = \mathrm{sqrt}(2)$
- Geometric margin $= 1 / ||w|| = 1 / \mathrm{sqrt}(2) \approx 0.707$

- Distance between margins = $2 / \|w\| = \sqrt{2} \approx 1.414$
- Alphas: $\alpha 1 = 1$ (for s1), $\alpha 2 = 1$ (for s2), both > 0, so both are true support vectors.

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.

Ans: A neuron in biology is a cell that receives and transmits electrical impulses, while in artificial networks it is a computational unit that processes weighted inputs through an activation function.

A synapse in biology is the junction where neurons exchange signals; in artificial networks, it is the weighted connection controlling signal strength between nodes.

The activation function in biology determines whether a neuron fires, while in artificial models it introduces nonlinearity and controls the neuron's output.

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

Ans :Nonlinearity is needed because it allows neural networks to learn complex nonlinear relationships, and prevents multiple layers from collapsing into a single linear transformation.

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

Ans: A directed acyclic graph (DAG) is assumed so signals flow only forward from input to output, ensuring stable computation without feedback loops.

Q2. Architecture & Capacity (8 pts)

a) Define **depth** and **width**.

Ans:Depth is the number of layers between input and output, and width is the number of neurons in each layer.

b) What are two distinct roles hidden units might learn (e.g., feature selection vs. projection)?

Ans:Hidden units can perform feature selection (identifying important inputs) or projection (transforming data into a new space for easier classification).

c) Contrast the cases **$D < M$** and **$D > M$** in a single-hidden-layer network: what representational trade-offs do they imply?

Ans:When $D < M$, the model expands inputs into higher dimensions for better separability but risks overfitting. When $D > M$, it compresses inputs into fewer features, improving efficiency but possibly losing information.

d) State the **universal approximation** idea for 1-hidden-layer NNs (informally).

Ans : The universal approximation idea states that a single hidden layer with enough neurons can approximate any continuous function on a bounded input region.

Q3. Perceptron vs. SVM (6 pts)

a) What does the perceptron algorithm optimize?

Ans:The perceptron optimizes for a separating hyperplane that classifies all training points correctly by minimizing misclassifications.

b) What is the main optimization goal of SVM?

Ans:The SVM optimizes for the hyperplane that maximizes the margin between classes while minimizing classification errors.

c) Why does SVM typically generalize better than a perceptron? Give one real-world scenario where this difference matters

Ans.SVMs generalize better because maximizing the margin reduces overfitting and improves robustness. For example, in spam email classification, an SVM performs better when data has overlapping or noisy samples.

Q4. Margins and Support Vectors (6 pts)

a) Define the **margin** of a dataset.

Ans:The margin is the minimum distance between the separating hyperplane and the closest data points from any class.

b) Explain the role of **support vectors** in determining the separating hyperplane.

Ans 👍Support vectors are the points that lie exactly on the margin boundaries and determine the position and orientation of the separating hyperplane.

c) Why does maximizing the margin lead to robustness against noise?

Ans: Maximizing the margin increases robustness to noise because small changes in the data are less likely to affect the classification boundary.

Q5. PCA Concepts (8 pts)

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

Ans:PCA assumes that data varies most along certain directions that capture the main structure or correlation in the dataset.

b) State two equivalent definitions of PCA.

Ans:PCA can be defined as (1) the projection that maximizes variance, and (2) the orthogonal transformation that diagonalizes the covariance matrix.

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

Ans:**Data must be centered before applying PCA to remove mean bias so that components represent actual variance rather than offsets.**

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

Ans:The eigenvalue represents the variance captured by its corresponding principal component; larger eigenvalues indicate directions with more significant information.

Q6. PCA vs. Random Projection (5 pts)

a) How does PCA choose directions compared to random projection?

Ans:PCA chooses projection directions based on the data's maximum variance, while random projection chooses directions randomly without considering data structure.

b) Give one advantage of PCA over random projection, and one disadvantage

Ans: .PCA's advantage is that it preserves meaningful variance and provides interpretable components. Its disadvantage is that it is computationally expensive and sensitive to data scaling, while random projection is faster but less accurate.