**ASSIGNMENT 4 AI – CSA1790**

**QUESTION 1**

**1. How Decision Trees Classify Emails as Spam/Not Spam**

Decision trees split data based on features (e.g., keywords, URLs) to maximize **information gain** using **entropy**. The tree stops growing when a stopping condition (e.g., max depth) is reached. New emails are classified by following the tree based on feature values.

**2. Key Features for Classification**

1. Spammy keywords (e.g., "win," "free").
2. Suspicious URLs.
3. Sender reputation.
4. Unusual attachments.
5. Email formatting (e.g., all caps, special characters).
6. Recipient count.
7. Subject line patterns.
8. Hidden HTML content.

**3. Training Process**

1. **Data Preparation**: Collect labeled data and preprocess features.
2. **Splitting**: Split data into training and testing sets.
3. **Tree Growth**: The tree splits nodes using **information gain**, with conditions:
   * **max\_depth=6** to limit tree depth.
   * **min\_samples\_split=3** to ensure enough samples for splits.
   * **min\_samples\_leaf=2** to avoid small splits.
   * **max\_features** to reduce overfitting.
4. **Evaluation**: Test on unseen data for performance validation.

**4. Limitations**

1. **Overfitting**: May struggle to generalize to new emails.
2. **Ambiguity**: Misclassifies mixed-characteristic emails.
3. **Imbalanced Data**: Can bias results toward dominant classes.
4. **Noise Sensitivity**: Susceptible to noisy or irrelevant features.

**QUESTION 2**

**1.Neural Network Structure (5 marks)**

* **Input Layer**: 3 neurons corresponding to the 3 input features (e.g., [0.5, -1.2, 0.3]).
* **Hidden Layer**: 4 neurons with weights and biases. Activation options: ReLU or Sigmoid.
* **Output Layer**: 1 neuron with a weight vector and bias. Activation options: Sigmoid or Tanh for binary classification (outputs a probability).

**2. Effects of Activation Functions**

* **ReLU**: Introduces non-linearity and avoids vanishing gradients. Suitable for hidden layers.
* **Sigmoid (Hidden)**: Squashes values between 0 and 1, but may cause vanishing gradients.
* **Sigmoid (Output)**: Outputs probabilities for binary classification.
* **Tanh (Output)**: Outputs values between -1 and 1, centering outputs around zero. Can be better for convergence.

**3. Chosen Activation Functions (5 marks)**

* **Hidden Layer**: **ReLU** (better for non-linearity and avoiding vanishing gradients).
* **Output Layer**: **Sigmoid** (suitable for binary classification to output probabilities).

**4. Calculate Network Output (6 marks)**

**Input Values**: [0.5,−1.2,0.3][0.5, -1.2, 0.3][0.5,−1.2,0.3]

* **Weights for Hidden Layer**:

[0.2−0.51.50.7−1.20.3−0.30.8−0.71.2−0.60.9]\begin{bmatrix} 0.2 & -0.5 & 1.5 \\ 0.7 & -1.2 & 0.3 \\ -0.3 & 0.8 & -0.7 \\ 1.2 & -0.6 & 0.9 \end{bmatrix} ​0.20.7−0.31.2​−0.5−1.20.8−0.6​1.50.3−0.70.9​​

* **Biases for Hidden Layer**: [0.1,−0.2,0.3,−0.4][0.1, -0.2, 0.3, -0.4][0.1,−0.2,0.3,−0.4]

1. **Hidden Layer Calculations**:  
   Multiply the input vector by the weight matrix:  
   Weighted Sum=W×Input+Bias\text{Weighted Sum} = W \times \text{Input} + \text{Bias}Weighted Sum=W×Input+Bias
2. Apply the ReLU activation.
3. **Output Layer Calculation**: Multiply hidden outputs by the output weights and add bias. Apply Sigmoid.

**5. Output with ReLU (Hidden) and Sigmoid (Output) (4 marks)**

1. Perform forward propagation using ReLU for hidden and Sigmoid for output.
2. This combination provides non-linearity while ensuring probabilistic binary classification.

QUESTION 3

**1. Role of Prior Probabilities (2 marks)**

Prior probabilities (P(h)P(h)P(h)) represent our initial belief about each hypothesis (e.g., bag type). After observing data (candy flavor), the posterior probability P(h∣D)P(h|D)P(h∣D) is updated based on how well each hypothesis explains the observed data. Strong priors can influence the outcome unless overridden by evidence.

**2. Posterior Probability Formula (2 marks)**

Using **Bayes’ Theorem**:

P(h∣D)=P(D∣h)×P(h)/P(D)

For one observed candy (e.g., "Cherry"):

* Calculate P(D∣h) for each bag (likelihood of drawing "Cherry").
* P(D)is the total probability across all hypotheses: P(D)=∑P(D∣h)×P(h)

**3. Update After First Candy (Cherry) (2 marks)**

Likelihoods for drawing "Cherry" from each bag:

* P(D="Cherry"∣h1)=1.0
* P(D="Cherry"∣h2)=0.75
* P(D="Cherry"∣h3)=0.5P
* P(D="Cherry"∣h4)=0.25
* P(D="Cherry"∣h5)=0.0
* Update posteriors for each hypothesis using Bayes’ formula.

**4. Next Candy Probability (2 marks)**

Probability of the next candy being "Cherry":

P("Cherry"∣D)=∑P(h∣D)×P("Cherry"∣h)

Weight each hypothesis' "Cherry" probability by its posterior.

**5. Effect of More Observations (2 marks)**

More observed candies reduce uncertainty by providing more evidence to update the posteriors. The predictions become more accurate as the model better identifies the bag type based on observed patterns.