Problem Set: Machine Learning Calculations

Ipshita Bonhi Upoma

Deadline: 15th May 2025, 11.59PM

Problem 1: Entropy and Information Gain in Decision Trees

You are given the following dataset for a binary classification problem where the goal is to predict whether to play tennis:

Outlook	Play Tennis
Sunny	No
Sunny	No
Overcast	Yes
Rainy	Yes
Rainy	No
Rainy	Yes
Overcast	Yes
Sunny	No
Sunny	Yes
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rainy	No

(a) Calculate the entropy of the entire dataset.

(b) Calculate the information gain when splitting on the **Outlook** attribute.

(c) If you split on **Outlook**, what is the resulting entropy for each branch?

Problem 2: Gradient Descent in Linear Regression

Consider a linear regression model:

$$\hat{y} = w_0 + w_1 x$$

You are given the following dataset:

x	y
1	2
2	3
3	5

Assume initial weights $w_0 = 0$, $w_1 = 0$, and a learning rate $\alpha = 0.1$.

(a) Write the mean squared error (MSE) loss function.

(b) Compute the gradient of the loss with respect to w_0 and w_1.

(c) Perform one step of gradient descent to update the weights.

Problem 3: Logistic Regression Gradient Descent

Consider the logistic regression model:

$$\hat{y} = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$

with the sigmoid activation $\sigma(z) = \frac{1}{1+e^{-z}}$. You are given one training example: $(x_1 = 1, x_2 = 2, y = 1)$. The initial weights are $w_0 = 0$, $w_1 = 0.5$, $w_2 = -0.5$, and learning rate $\alpha = 0.1$.

(a) Compute the predicted probability $\hat{y}$.

(b) Write the binary cross-entropy loss.

(c) Compute the gradient of the loss with respect to each weight.

(d) Update the weights using gradient descent.

Problem 4: Backpropagation in a Hidden Neuron

Consider a feedforward neural network with one hidden layer. You are given the following for a single training example:

- Input $x = [1, 2]$
- Hidden neuron: $z = w_1 x_1 + w_2 x_2 + b = 0.3$, activation $\sigma(z) = \frac{1}{1+e^{-z}}$
- Output neuron: receives $a = \sigma(z)$ and gives prediction $\hat{y} = \sigma(w_3 a + b_2) = 0.6$
- True label: $y = 1$

- Loss function: binary cross-entropy

Given:

$$w_1 = 0.1, \quad w_2 = 0.2, \quad w_3 = 0.4, \quad b = 0.1, \quad b_2 = 0.1$$

(a) Compute the derivative of the loss with respect to the output $\hat{y}$.

(b) Use the chain rule to compute the gradient of the loss with respect to w_1 and w_2.

(c) Update the weights using learning rate $\alpha = 0.1$.