Binary-class classification

Multiclass Classification

Training Set:

Description	**Label**
"Small, furry, meows"	Cat
"Barks, loyal, furry"	Dog
"Long ears, hops, small"	Rabbit

Test Set:

Description	**Label**
"Furry, long ears"	Rabbit

Solution:

Step 1: Preprocessing

Vocabulary:

The vocabulary consists of 8 unique words:

Vocabulary = ["small", "furry", "meows", "barks", "loyal", "long", "ears", "hops"]

Vocabulary = ["small", "furry", "meows", "barks", "loyal", "long", "ears", "hops"]

Vocabulary size = 8.

Vectorization:

Each sentence is converted into a binary vector based on the presence or absence of words in the vocabulary.

Sentence	Vector
"Small, furry, meows"	[1, 1, 1, 0, 0, 0, 0, 0]
"Barks, loyal, furry"	[0, 1, 0, 1, 1, 0, 0, 0]
"Long ears, hops, small"	[1, 0, 0, 0, 0, 1, 1, 1]
"Furry, long ears"	[0, 1, 0, 0, 0, 1, 1, 0]

Step 2: Neural Network Architecture

- **Input Layer**: 8 nodes (one for each word in the vocabulary).
- **Hidden Layer**: (assume) 3 nodes (ReLU activation).
- **Output Layer**: 3 nodes (Softmax activation for multiclass classification).

Step 3: Forward Propagation for the Test Sentence

Let's classify **"Furry, long ears"**.

3.1 Input Vector:

$$\boldsymbol{X} = [\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}]$$

3.2 Hidden Layer Calculation

Weights and Biases for Hidden Layer:

$$W_1 = \begin{bmatrix} 0.2 & 0.5 & 0.1 \\ -0.3 & 0.7 & 0.2 \\ -0.5 & 0.3 & -0.1 \\ 0.3 & 0.6 & 0.8 \\ -0.4 & -0.2 & 0.1 \\ -0.7 & 0.4 & -0.2 \\ 0.1 & 0.5 & 0.9 \\ 0.3 & 0.6 & 0.8 \end{bmatrix}$$

$$b_1 = [0.1,0.2,0.3]$$

hidded layer Calculation:

$$Z_1 = X.W_1 + b_1$$

$$[\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}] * \begin{bmatrix} 0.2 & 0.5 & 0.1 \\ -0.3 & 0.7 & 0.2 \\ -0.5 & 0.3 & -0.1 \\ 0.3 & 0.6 & 0.8 \\ -0.4 & -0.2 & 0.1 \\ -0.7 & 0.4 & -0.2 \\ 0.1 & 0.5 & 0.9 \\ 0.3 & 0.6 & 0.8 \end{bmatrix} + [0.1, 0.2, 0.3]$$

$$= [(0.3 - 0.7 + 0.1), (0.7 + 0.4 + 0.5), (0.2 - 0.2 + 0.9)] + [0.1, 0.2, 0.3]$$

$$= [-0.3,1.6,0.9] + [0.1,0.2,0.3] = [-0.2,1.8,1.2]$$

Step 4: Apply ReLU Activation

ReLU activation is applied to each Z:

$$ReLU(Z) = \max(0, Z)$$
$$ReLU(Z_1) = \max(0, -0.2) = 0$$
$$ReLU(Z_2) = \max(0, 1.8) = 1.8$$
$$ReLU(Z_3) = \max(0, 1.2) = 1.2$$

Step 5:

Output Layer calculation

$$h = [0, 1.8, 1.2]$$
$$W_2 = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.1 & 0.6 & 0.8 \\ -0.5 & 0.2 & 0.8 \end{bmatrix}$$
$$b_2 = [0.1, -0.1, 0.05]$$

Calculation:

$$[0,1.8,1.2] * \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.1 & 0.6 & 0.8 \\ -0.5 & 0.2 & 0.8 \end{bmatrix} + [0.1, -0.1, 0.05]$$
$$= (0 + 0.18 - 0.6), (0 + 1.08 + 0.24), (0 + 1.44 + 0.96) + [0.1, -0.1, 0.05]$$
$$= [-0.42, 1.32, 2.4] + [0.1, -0.1, 0.05]$$
$$= [-0.32, 1.22, 2.45]$$

Step 6: Apply Softmax Activation

Softmax converts Z-values into probabilities:

$$P(y = c_i) = \frac{e^{z_{c_i}}}{\sum_j e^{z_j}}$$

Calculate Exponentials:

$$e^{-0.32} \approx 0.726\,, e^{1.22} \approx 3.87, e^{2.45} \approx 11.59,$$

$$\sum = (0.726 + 3.87 + 11.59) \approx 16.186$$

Probability calculation:

$$P(cat) = \frac{0.726}{16.186} \approx 0.044$$
$$P(dog) = \frac{3.87}{16.186} \approx 0.239$$
$$P(rabbit) = \frac{11.59}{16.186} \approx 0.716$$

Error Calculation (Cross-Entropy Loss):

Given the true label is **Rabbit** $(y = [0,0,1])$:

The cross-entropy loss is:

$$\mathcal{L} = \sum_{i=1}^{3} -y_i \log(\hat{y}_i)$$

$$\mathcal{L} = -((0. \log(0.044) + 0. \log(0.239) + 1. \log(0.716)) = \log\left(\frac{250}{179}\right) = 0.145$$