Non-Programming Assignment

Question 1.

The **reason for using softmax** is to convert the raw output (logits) of a neural network into a **probability distribution** over multiple classes. In many classification problems, especially multi-class classification, the network needs to assign a probability to each possible class. Softmax is ideal in this scenario because it ensures:

1. **Non-negative Outputs**: It converts raw output values into positive numbers.
2. **Sum to One**: The output probabilities sum up to 1, which makes it easier to interpret the result as a valid probability distribution.
3. **Normalization**: It helps in normalizing the outputs so that they are on a consistent scale, which is crucial when comparing different predictions.

In essence, softmax helps in making the model's output interpretable, especially when it is required to predict **one class out of multiple possible classes**.

Question 2

**Softmax** is a mathematical function used in machine learning models, particularly in the final layer of neural networks used for **multi-class classification**. It transforms a vector of arbitrary real-valued scores (logits) into a vector of **probabilities**.

The formula for the softmax function is:

σ(zi)=ezi/∑j=1Nezj

Where:

* zi ​ is the input score for the ith class.
* ezi ​ is the exponential function applied to the score.
* The denominator sums over all N classes, ensuring that the output values sum to 1.

**How Does Softmax Work?**

1. **Exponentiation**: The softmax function takes the exponential of each logit. This step ensures that all outputs are positive and amplifies differences between larger and smaller logits.
2. **Normalization**: The result of each exponentiated logit is divided by the sum of all exponentiated logits. This step converts the values into probabilities that sum to 1.
3. **Interpretation**: The resulting vector represents the probabilities of each class. The class with the highest probability is typically selected as the model's prediction.

**Example**

Suppose a neural network outputs the following logits for a 3-class classification problem:

z = [2.0,1.0,0.1]

Applying the softmax function:

1. Exponentiate each score:

e2=7.389, e1=2.718, e0.1=1.105

1. Compute the sum of the exponentials:

7.389+2.718+1.105=11.212

1. Normalize to get probabilities:

σ(z1) = 7.389/11.212 =0.659,σ(z2)=2.718/11.212=0.242,σ(z3)=1.105/11.212=0.099

This means:

* Class 1: 65.9% probability
* Class 2: 24.2% probability
* Class 3: 9.9% probability

The model would predict **Class 1** as it has the highest probability.