1. Define Binary cross entropy as a cost function.

Ans:

Binary cross-entropy is a loss function that is often used in binary classification tasks. It measures the distance between the predicted probability distribution and the actual distribution for a binary label.

In other words, the binary cross-entropy loss function is used to evaluate the performance of a model that has been trained to predict a binary outcome. It compares the predicted probability of the positive class to the actual label and penalizes the model if the predicted probability is far from the actual label.

The binary cross-entropy loss is defined as

```
Loss = -(y * log(p) + (1 - y) * log(1 - p))
```

Where y is the actual label (0 or 1) and p is the predicted probability of the positive class.

The binary cross-entropy loss function is often used in conjunction with a sigmoid activation function in the final layer of a neural network, as the sigmoid function outputs a probability value between 0 and 1. This makes it well-suited for binary classification tasks, where the model is trying to predict the probability that an input belongs to a certain class.

2. Difference between Adam optimizer & gradient descent.

Ans:

Adam Optimizer and gradient descent are both optimization algorithms that are commonly used for training neural networks. Both algorithms work by iteratively updating the weights of the network in order to minimize the loss function, which measures how well the model is able to predict the desired output.

However, Adam Optimizer includes some additional features that make it more efficient and effective compared to vanilla gradient descent. These features include

- Moving averages of the parameters: Adam Optimizer keeps track of the moving averages of the parameters, which helps to reduce the variance of the gradients and improve the stability of the learning process. Which gradient descent does not.

- Adaptive learning rates: Adam Optimizer adjusts the learning rates for different parameters automatically, based on the historical gradient information. This helps to ensure that the learning rates are well-suited for the specific characteristics of the data and the model. In contrast, gradient descent uses a fixed learning rate for all parameters.

Overall, Adam Optimizer is a more sophisticated optimization algorithm that is particularly well-suited for training deep neural networks, where the gradients can have high variance and the model may be prone to overfitting.