Question : Difference between adam optimizer and gradient descent

Answer : Gradient Descent and Adam optimizer are both optimization algorithms that are commonly used to update the parameters of a machine learning model during training. However, there are some key differences between the two.

Gradient Descent is a first-order optimization algorithm that iteratively updates the model parameters in the direction of the negative gradient of the cost function with respect to the parameters. The magnitude of the update step is controlled by the learning rate, which determines how quickly the algorithm converges to a minimum.

Adam optimizer is an optimization algorithm that combines ideas from gradient descent and momentum optimization. It uses moving averages of the gradients to provide an estimate of the first and second moments of the gradients; the first moment provides information about the direction of the gradient, and the second moment provides information about the variability of the gradient. Based on this information, Adam optimizer adjusts the learning rate for each parameter in a manner that is more sophisticated than the fixed learning rate used by gradient descent.

In summary, Adam optimizer is a more advanced optimization algorithm than gradient descent and is generally faster and more efficient, especially in deep learning applications where the cost functions are highly non-linear and have many local minima. However, gradient descent is simpler and easier to understand, and is often a good choice for simple problems or as a baseline to compare against more advanced algorithms.

Question : Define binary cross entropy as cost function

Answer : Binary cross-entropy is a commonly used cost function in binary classification problems. The binary cross-entropy loss measures the dissimilarity between the predicted probability distribution and the true distribution of the binary target. The binary target is typically represented as a one-hot encoded vector with values 0 or 1. The predicted probabilities can be generated by a model, such as a neural network, that outputs a probability for each class.
The binary cross-entropy is defined as:

$$- (y * \log(p) + (1 - y) * \log(1 - p))$$

where:

- y is the true binary target
- p is the predicted probability for the positive class

The binary cross-entropy is a measure of how well the predicted probabilities match the true binary targets. The goal of training a model is to minimise the binary cross-entropy, so that the model produces probabilities that are as close as possible to the true binary targets.