1) Logistic regression gives output as the probability(Values between 0 and 1) of a particular input belonging to a certain class. The cross-entropy loss function is appropriate for probability estimation. It measures the difference between the predicted probabilities and the true class labels. On the other hand, mean squared error (MSE) is more suitable for regression tasks where the output is continuous. It measures the average squared difference between the predicted values and the true values. It can lead to sub-optimal results, especially when dealing with classification tasks.

The Choice of loss function is important because we can choose a loss function that can deal with probability estimation and also depending on whether it is a classification task or regression. Cross-entropy loss encourages the model to output probabilities close to 1 for the true class and close to 0 for other classes. This would lead to more accurate predictions.

2) Both options A and B are correct. Cross-entropy (CE) Loss and Mean Squared Error (MSE) Loss guarantee convex optimization problems when used with linear activation functions.

The squared function is convex that's why squaring the difference between the actual and predicted value for MSE loss ensures convexity.

For the Cross-Entropy loss, since the output of the neural network with linear activation functions can be any real value, the cross-entropy loss function can be rewritten to consider only the linear part, effectively reducing it to a convex optimization problem.

3) Dataset: We have used CIFAR-10 dataset. The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class.

Data Preprocessing:

a) The CIFAR-10 dataset is loaded using TensorFlow/Keras.

b) Input images are preprocessed by scaling pixel values to the range [0, 1] to ensure numerical stability during training.

c) Class labels are one-hot encoded to convert them into binary class matrices.

Data Augmentation:

a) Data augmentation is applied to the training dataset using the ImageDataGenerator class from Tensor-

Flow/Keras.

b) Augmentation techniques include rotation, width shift, height shift, and horizontal flipping to increase the diversity of training data and improve model generalization.

Model Architecture:

The network has 2 hidden layers. The first hidden layer has 128 neurons with ReLU activation, and the second hidden layer has 64 neurons also with ReLU activation. Finally, there is an output layer with 10 neurons (for 10 classes) using a softmax activation function.

Input layer: Flatten layer to convert input images into a vector.

Hidden layer: Dense layer with 128 neurons and ReLU activation function.

Hidden layer: Dense layer with 64 neurons and ReLU activation function.

Output layer: Dense layer with 10 neurons (corresponding to the number of classes in CIFAR-10) and softmax activation function for multi-class classification.

Model Training

a) The model is compiled using the Adam optimizer and categorical cross-entropy loss function.

b) Training is performed using the augmented training data generator.

c) The model is trained for 10 epochs with a batch size of 32.

Model Evaluation a) The trained model is evaluated using the test dataset.

b) Test accuracy is computed to assess the performance of the model on unseen data.

Results The trained model achieved a test accuracy of 41.67%, indicating its effectiveness in classifying CIFAR-10 images.