1 Question 1

For logistic regression, Cross Entropy (also known as Log Loss) is generally the preferred loss function over Mean Squared Error (MSE).

Importance of a Single Best Answer:

Choosing a loss function that aligns with the nature of the problem and the model is crucial for ensuring that the model converges to a single best solution - that is, the set of parameters that minimize the loss. Cross Entropy, by being more aligned with the probabilistic output of logistic regression, ensures that the model's training process is efficient and directed towards minimizing the discrepancy between the predicted probabilities and the actual class labels.

Gradient Behavior:

The gradient of the Cross Entropy loss function with respect to the model's parameters yields more informative updates for the weights, especially when the model is wrong. This is because the gradient of the Cross Entropy loss with respect to the model's parameters is proportional to the difference between the predicted probability and the actual label. This property ensures that when the model is confident and wrong, the weight updates are significant, helping the model to learn faster and more effectively.

Vanishing Gradient with MSE:

When MSE is used as a loss function for logistic regression, the gradient can become very small when the model's predictions are close to 0 or 1, even if those predictions are incorrect. This leads to the vanishing gradient problem, where the model stops learning or learns very slowly because the updates to the weights become insignificantly small.

Cross Entropy loss provides a more direct and informative gradient for updating the model's weights, especially when the predicted probability is wrong, thus guiding the model more effectively towards the single best solution.

2 Question 2

Neither Cross Entropy nor MSE leads to a convex optimization problem in the context of a deep neural network with linear activation functions.

(a) Cross Entropy (CE):

Cross Entropy Loss in a deep neural network, even with linear activations, does not lead to a convex optimization problem. The reason is that the output of a neural network, even with linear activations, is a composition of linear functions, which remains a linear function (essentially collapsing all layers into a single linear transformation). However, when this linear output is passed through the sigmoid function (which is common in binary classification tasks to obtain probabilities), the resulting function is no longer linear. Applying Cross Entropy to these probabilities introduces non-linearity due to the logarithmic component of the CE loss. Consequently, the loss surface with respect to the weights becomes non-convex due to this composition of non-linear functions (sigmoid and logarithm).

(b) Mean Squared Error (MSE):

Mean Squared Error Loss, when used in a deep neural network with linear activation functions, also does not guarantee a convex optimization problem. The issue again lies in the composition of functions. Although MSE itself is convex with respect to the predicted output, the presence of multiple layers and the final application of a non-linear function (like a sigmoid for converting the output to a probability in binary classification) render the overall optimization landscape non-convex.

The presence of multiple layers, even with linear activations, coupled with the application of a non-linear function at the output layer for binary classification tasks, ensures that the optimization landscape is non-convex regardless of whether Cross Entropy or Mean Squared Error is used as the loss function.

Therefore, the correct answer is (d) None.