Q2

A binary decision tree with maximally allowed depth of 2 can split the feature space into up to four regions. The decision boundary in this case would consist of two thresholds, one for each of the two continuous-valued features, that divide the input space into four rectangles.

Assuming the two features are called X1 and X2, a possible decision boundary for a binary decision tree of depth 2 could be as follows:

If X1 <= t1 and X2 <= t2, then classify as Class 1.

If X1 <= t1 and X2 > t2, then classify as Class 2.

If X1 > t1 and X2 <= t2, then classify as Class 3.

If X1 > t1 and X2 > t2, then classify as Class 1.

In this case, the decision boundary consists of two perpendicular lines, one parallel to the X1 axis and the other parallel to the X2 axis, that intersect at the threshold values t1 and t2. The decision boundary splits the feature space into four rectangles, where each rectangle is associated with a different class label.

Note that this is just one possible decision boundary, and the actual decision boundary may look different depending on the data and the specific tree algorithm used.

If we have a binary decision tree classifier with two continuous-valued features and the maximum tree depth allowed is 2, the decision boundary can be a piecewise linear function that divides the feature space into four regions.

At the first level of the tree, the decision boundary splits the feature space into two regions using one of the features. The split can be represented as a vertical or horizontal line. For example, if we have features x1 and x2, the decision boundary could be x1 = c or x2 = d, where c and d are constants.

At the second level of the tree, the decision boundary further splits each of the two regions created by the first split into two smaller regions. The split can again be represented as a vertical or horizontal line. For example, if the first split was x1 = c, the second split could be x2 = e or x2 = f, where e and f are constants.

The resulting decision boundary will be a combination of the splits at each level of the tree, creating a piecewise linear function. The decision boundary will classify the points in the feature space into one of two classes, based on which side of the boundary they lie.

Overall, the decision boundary will be a combination of two straight lines or planes, depending on the number of features and the dimensionality of the feature space. This decision boundary may not be optimal for all datasets, as it may not be able to capture more complex relationships between the features and the target variable.

Thank you for providing the architecture. There seem to be a few errors in this convolutional neural network architecture. Here are the corrections and explanations:

Dropout(p=1.0): This means that the dropout layer will drop out all the neurons during training, resulting in the network not learning anything. The correction would be to remove the dropout layer or set the dropout probability to a value less than 1.0.

Dense(27 input, 1 output): The number of inputs is incorrect. The input should match the output of the previous layer, which is the output of the flatten layer. Since the input image is 8x8 grayscale, the output of the flatten layer will be 64. Therefore, the correction would be to change the input of the dense layer to 64.

SoftMax output: Since this is a binary classification task, the output activation function should be sigmoid, not softmax.

2D max pooling (2×2): There is no error in this layer.

2D convolution (3 filters, 3×3, ReLU activation): There is no error in this layer.

Here is the corrected architecture:

Input image(8×8, grayscale) -> 2D convolution (3 filters, 3×3, ReLU activation) -> 2D max pooling(2×2) -> Flatten -> Dense(64 input, 1 output, sigmoid activation)

This architecture should be able to perform the binary classification task of 8x8 grayscale images with better accuracy.