Practical Machine Learning with Tensorflow

Assignment 2

1. Given the output data distributions, as most of the training data lies around 0, the model learns the corresponding parameters well enough and will tend to have higher accuracies if the test data (y_test) is also around 0 as well.
2. False as the training data distribution varies from the test data distribution. The best way to split training and test data is to take parts of each city's data into training and test dataset.
3. Temperature data might contain has seasons, trend components etc. Hence simple models like (ii) will underfit this data. Much more complex models like NN() in the (iii) part is too large and contains too many parameters that might overfit the data. Hence (i) T = a **sin(**bX**)** + c **sin(**dX**)** + **f(**X) where f() is a polynomial function is the correct model to use here as it is possible to learn seasonal component using trigonometric functions and trend as f() i.e., polynomial function.
4. (a) is true as by increasing the number of units and layers, we are adding more parameters to learn the training data distribution. As we keep increasing the number of parameters, the model will start overfitting the data. If the model overfits the data, the test performance of the model will fall and hence, (b) is false. Adding many new features gives us more expressive models which are able to better fit our training data. Hence, (c) is true. But if too many new features are added, this can lead to overfitting of the training data. Overfitting leads to reduced test performance and hence, (d) is false.
5. Classification report:

```
             precision    recall  f1-score   support

          0       0.00      0.00      0.00         2
          1       1.00      0.60      0.75         5
          2       0.50      1.00      0.67         3

avg / total       0.65      0.60      0.57        10
```

6. Refer Classification report in Q5
7. Refer Classification report in Q5.
   (Support is the total number of samples of the true response (actual values) that lie in that class.)
8. In case of uneven class distribution f1-score is better evaluation metric than accuracy
9. Decision boundary in the given data resembles a circle. Hence, the **d** option is correct.
10. As inputs are linear, it as decision boundary needs to be polyhedral, we need 3 hidden variables so that the model could at least learn a triangular-shaped boundary.
11. The first hidden layer can only learn linear decision boundaries. No matter how many neurons are in it.

12. If activation is a Linear function, the model won't be able to learn the optimal decision boundary. To verify, you can run it on the tensorflow playground.
13. This function won't be able to learn XOR function, as it can only work with linear decision boundary.
14. The range of tanh function is [-1, 1] and the range of sigmoid is [0, 1]. Hence the expected output of tanh is closer to 0 than expected output of sigmoid and hence, it centers the data better. Since the data is centered around 0, we have two advantages. First, the gradients are higher. Second, we avoid bias in our gradients. For a more detailed explanation, you can read this paper.
15. We have a neural network with an input layer of h_0 nodes, hidden layers of $h_1, h_2, h_3, \ldots, h_{l-1}$ nodes respectively and an output layer of h_l nodes.

    Total number of parameters = $\sum_{i=0}^{l-1} [\,(h_i * h_{i+1}) \; + \; h_{i+1}]$ = ((4 * 6) + 6) + ((6 * 6) + 6) + ((6 * 6) + 6) = 30 + 42 + 42 = 114

    Note: We realise that the assignment page wrongly shows 156 as correct. We're currently looking into it. Sorry for the confusion.
16. Yes, we can change the activation function accordingly. 1/0 is an asymptotic case.