HW5

1. Data Preprocessing:

The first preprocessing step is to calculate the citation ratio for each university. This ratio is obtained by dividing the number of citations in 2022 by the number of citations in 2021. It's an essential feature that captures the change in citations over time, which is a key factor in determining the category.

The citation data from 2017 to 2022 is then normalized using Min-Max scaling. Normalization is essential to ensure that all features are on the same scale (between 0 and 1), preventing some features from dominating others during training.

The 'category' column is created by categorizing universities based on their citation ratios. The categorization assigns values as follows:

0 for 'Low' (<1.05): Universities with a minimal increase in citations from 2021 to 2022.
1 for 'Medium' (1.05-1.15): Universities with a moderate increase in citations.
2 for 'High' (>1.15): Universities with a significant increase in citations.

2. Data Splitting:

The dataset is split into two main parts: features (X) and labels (Y). The features (X) consist of the normalized citation data for the years 2017 to 2022. These are the input variables that the model will use for prediction.

The labels (Y) are created by one-hot encoding the 'category' column. One-hot encoding converts categorical labels into a binary matrix representation, where each category corresponds to a unique combination of binary values.

3. Neural Network Model Design:

A neural network model is created using TensorFlow and Keras, a popular deep learning library. The model architecture consists of three layers:

Input Layer: This layer has 6 nodes, one for each year's normalized citation data. It serves as the entry point for data.
Hidden Layer: This layer contains 6 nodes and uses the Rectified Linear Unit (ReLU) activation function. It is responsible for learning complex patterns in the data.
Output Layer: This layer has 3 nodes, one for each category (Low, Medium, High). It uses the softmax activation function to produce probability scores for each category.

4. Model Compilation:

The model is compiled with the following settings:
Loss Function: Categorical cross-entropy is used as the loss function. It measures the dissimilarity between predicted and actual category labels.

Optimizer: The Adam optimizer is chosen. It's an adaptive learning rate optimization algorithm.
Metrics: The model's performance is evaluated using accuracy, which is a common metric for classification tasks.

5. Model Training:

The model is trained using the training data (X_train and y_train) for a specified number of epochs (in this case, 1000) and with a batch size of 32. During training, the model learns to make predictions based on the input features and minimize the loss.

6. Model Evaluation:

The final step involves evaluating the model's performance on the test data (X_test and y_test). The accuracy is reported, indicating how well the model can classify universities into the three categories based on their citation ratios.

7. Output:

```
1/1 [==============================] - 0s 226ms/step - loss: 0.6832 - accuracy: 0.7500
Test Accuracy: 75.00%
```

In this specific example, the model achieves a test accuracy of 75%, meaning that it correctly classifies universities into 'Low,' 'Medium,' and 'High' categories with 75% accuracy.
I have checked the results by varying the number of epochs. For 10000 epochs we get the Test accuracy as 80%. But as our dataset is small, we may not need a large number of epochs.