Brijesh verma
SUID: 826257344

CIS 662: Intro to Machine Learning and Algorithms

HW-5

1. Problem Statement:

Based on the ratio **(citations in 2022)/(citations in 2021)**,
approximated to two decimal places, determine the category of each individual as one of the three shown below:

1. Low (<1.05).
2. Medium (1.06-1.15).
3. High (>1.15).

Use a **1-hidden layer 6-6-3 neural network** to solve this **classification problem**, using 80% of the data for training.
The inputs to the network would be the citation numbers **from 2017 to 2022, normalized** as you consider appropriate.
Evaluate the results on the remaining (20%) test data.

Your submission should include:
1. code
2. report_HW5.pdf (Explain your approach to this classification problem. Comment on your results.)

2. Solution Steps:

1. Data Preprocessing:

a. Data Loading: The dataset was loaded from a CSV file ('31-40.csv') using the Pandas library.

b. Data Cleaning: The 'cit_2017' column was cleaned by removing commas and converting it to integers.

c. Feature Engineering: The 'Ratio' column was created by calculating the ratio of citations in 2022 to citations in 2021 and rounding it to two decimal places.

d. Categorization: A function was defined to categorize the numerical value in the 'Ratio' column into one of the three categories: Low, Medium, or High.

e. Feature Selection: The input features (X) and target labels (y) were extracted from the dataset. Features included citation numbers from 2017 to 2022, and the 'Ratio' column was used as the target label.

f. Label Encoding: The target labels were encoded into numerical values (0, 1, 2) representing Low, Medium, and High categories, respectively.

g. Feature Scaling: Min-Max scaling was applied to normalize the input features within a range of [0, 1].

h. Data Split: The dataset was split into training and test sets with an 80% training set and a 20% test set.

3. Model Building:

a. Neural Network: A feedforward neural network with one hidden layer (6-6-3) was created using the Keras library.

b. Activation Functions: ReLU (Rectified Linear Unit) activation functions were used in the hidden layers, and softmax activation was used in the output layer.

c. Loss Function and Optimizer: sparse_categorical_crossentropy was chosen as the loss function, and the Adam optimizer with a learning rate of 0.1 was used.

4. Model Training:

The model was trained on the entire dataset for 100 epochs with a batch size of 4, and a validation split of 20% was used during training.

4. Confusion Matrix and Classification Report:

Calculated the confusion matrix and generated a classification report for the test results.

5. Results:

a. Training Results: The training process was displayed epoch by epoch, showing the loss and accuracy. The model seemed to converge quickly, reaching a training accuracy of 79.69% with a loss of approximately 0.5049.

```
In [14]:  # Train the model on the dataset
          model.fit(X_train, y_train, epochs=100, batch_size=16, validation_split=0.2)
4/4 [==============================] - 0s 10ms/step - loss: 0.5496 - accuracy: 0.7969 - val_loss: 0.7240 - val_accuracy:
0.6875
Epoch 95/100
4/4 [==============================] - 0s 14ms/step - loss: 0.5009 - accuracy: 0.8281 - val_loss: 0.7045 - val_accuracy:
0.7500
Epoch 96/100
4/4 [==============================] - 0s 10ms/step - loss: 0.5393 - accuracy: 0.7656 - val_loss: 0.7395 - val_accuracy:
0.6875
Epoch 97/100
4/4 [==============================] - 0s 10ms/step - loss: 0.5260 - accuracy: 0.7812 - val_loss: 0.7354 - val_accuracy:
0.6875
Epoch 98/100
4/4 [==============================] - 0s 10ms/step - loss: 0.5084 - accuracy: 0.7812 - val_loss: 0.7587 - val_accuracy:
0.6875
Epoch 99/100
4/4 [==============================] - 0s 10ms/step - loss: 0.4726 - accuracy: 0.7969 - val_loss: 0.7344 - val_accuracy:
0.6875
Epoch 100/100
4/4 [==============================] - 0s 10ms/step - loss: 0.5049 - accuracy: 0.7969 - val_loss: 0.6445 - val_accuracy:
0.7500
```

b. Test Results: When evaluated on the test set, the model achieved a accuracy of 80%, with a test loss of 0.5756.

```
1/1 [==============================] - 0s 102ms/step - loss: 0.5
756 - accuracy: 0.8000
```

Test Loss: 0.5756, Test Accuracy: 0.8000

CLASSIFICATION REPORT

Classification Report:

	precision	recall	f1-score	support
Low	0.79	1.00	0.88	11
Medium	0.00	0.00	0.00	2
High	1.00	0.71	0.83	7
accuracy			0.80	20
macro avg	0.60	0.57	0.57	20
weighted avg	0.78	0.80	0.78	20

3. Conclusion:

- The model is trained for 100 epochs, and you can see the training and validation loss and accuracy for each epoch.
- After training, the model is evaluated on the test set, and it achieves a test accuracy of 80%. This means that the model performs well in classifying the 'Ratio' categories on the test data.
- The 'y_pred' variable contains the predicted labels for the test data, with probability values for each class.
- The confusion matrix and classification report provide more detailed insights into the model's performance, showing how well it classifies instances into the three categories.
  The code appears to be working as intended and has achieved a reasonably good accuracy on the test data. we can further fine-tune the model or try different neural network architectures and hyperparameters to potentially improve its performance.

Brijesh verma
SUID: 826257344

References:

https://chat.openai.com/,
https://keras.io
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
https://towardsdatascience.com/machine-learning-classification-52241849468a
https://scikit-learn.org/stable/modules/model_evaluation.html