**HANDWRITTEN TEXT GENERATION**

1. Data Preparation:

* A sample text is provided as input, which appears to contain information about the IAM database.
* The unique characters in the text are identified and stored in the chars variable.
* The total number of unique characters is calculated and stored in num\_chars.
* Character-to-index and index-to-character dictionaries are created to map characters to integers and vice versa.

1. Sequence Generation:

* The script then generates sequences of characters from the input text. It creates input sequences of a fixed length (seq\_length) and their corresponding output characters. These sequences are stored in X and y, respectively.

1. Model Architecture:

* A sequential Keras model is defined with two LSTM layers, each containing 128 units, and a final dense layer with a softmax activation function. This architecture is used for text generation.

1. Model Compilation:

* The model is compiled using categorical cross-entropy as the loss function and the Adam optimizer.

1. Model Training:

* The model is trained on the generated sequences for a specified number of epochs (in this case, 20) with a batch size of 128.

1. Model Saving:

* After training, the model is saved to a file named 'text\_generation\_model.h5'.

1. Text Generation:

* A function generate\_text is defined to generate text using the trained model. It takes a seed text and generates a continuation of text based on the model's predictions. The length of the generated text is specified as 200 characters.

1. Generating Text:

* The script loads the trained model from the saved file and generates text using the seed text "The IAM database contains."

1. Printing Generated Text:

* The generated text is printed to the console.