

**Experiment No 9**

Design and develop a model for character level text generation using LSTM

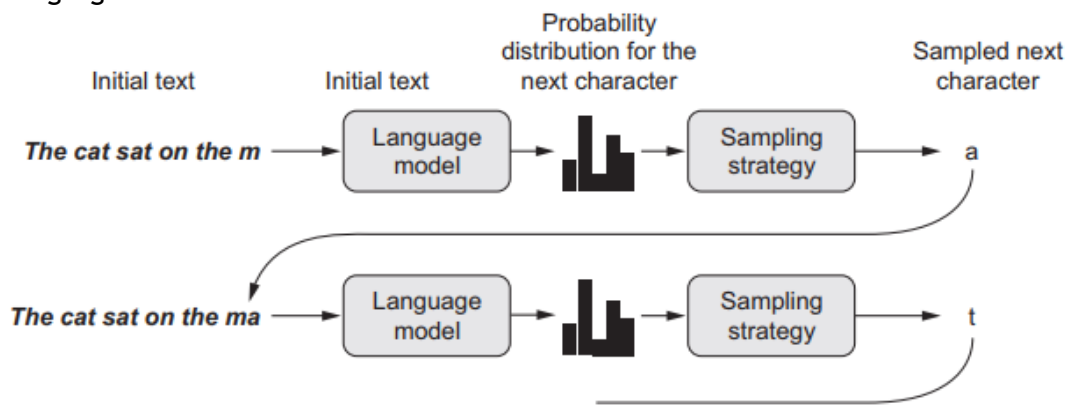
**Objective:**

At the end of this practical session, student will be able to generate text after building the LSTM network.

**Theory:****A. Introduction**

The universal way to generate sequence data in deep learning is to train a network (usually an RNN or a convnet) to predict the next token or next few tokens in a sequence, using the previous tokens as input. For instance, given the input “the cat is on the ma,” the network is trained to predict the target  $t$ , the next character. As usual when working with text data, tokens are typically words or characters, and any network that can model the probability of the next token given the previous ones is called a language model. A language model captures the latent space of language: its statistical structure.

Once you have such a trained language model, you can sample from it (generate new sequences): you feed it an initial string of text (called conditioning data), ask it to generate the next character or the next word (you can even generate several tokens at once), add the generated output back to the input data, and repeat the process many times as shown in following figure.



This loop allows you to generate sequences of arbitrary length that reflect the structure of the data on which the model was trained: sequences that look almost like human-written sentences. In the example, you'll take a LSTM layer, feed it strings of  $N$  characters extracted from a text corpus, and train it to predict character  $N + 1$ . The output of the model will be a softmax over all possible characters: a probability distribution for the next character. This LSTM is called a character-level neural language model.

**B. Generative model in Keras**

For generative model, the first thing you need is a lot of text data that you can use to learn a language model. You can use any sufficiently large text file or set of text files like Wikipedia. In keras, download the corpus with `keras.utils.get_file()`. To work with this corpus, build the network with a single LSTM layer followed by a Dense classifier and softmax over all possible characters. Use `layers.LSTM` and `layers.Dense` from keras to build LSTM and Dense layers. Train this model to develop language model.

Given a trained model and a seed text snippet, generate new text by doing the following repeatedly:

- 1 Draw from the model a probability distribution for the next character, given the generated text available so far.
- 2 Reweight the distribution to a certain temperature.
- 3 Sample the next character at random according to the reweighted distribution.
- 4 Add the new character at the end of the available text.

**Keyword:**

LSTM, Natural Language Model, Character Level Generative Model

**Procedure:**

1. Download and parse the initial text file i.e. corpus
2. Vectorize sequences of characters
3. Build a Single-layer LSTM model for next-character prediction
4. Set Model compilation configuration
5. Write function to sample the next character given the model's predictions
6. Write text-generation loop