	Marwadi University **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Artificial Intelligence (01CT0616)	**Aim: To remember the important tasks and forget the less important tasks using LSTM**	
Experiment No: 05	**Date:**	**Enrolment No: 92200133030**

Aim: To remember the important tasks and forget the less important tasks using LSTM

IDE: Google Colab

Theory:

Countless learning tasks require dealing with sequential data. Image captioning, speech synthesis, and music generation all require that models produce outputs consisting of sequences. In other domains, such as time series prediction, video analysis, and musical information retrieval, a model must learn from inputs that are sequences. These demands often arise simultaneously: tasks such as translating passages of text from one natural language to another, engaging in dialogue, or controlling a robot, demand that models both ingest and output sequentially-structured data.

Recurrent neural networks (RNNs) are deep learning models that capture the dynamics of sequences via *recurrent* connections, which can be thought of as cycles in the network of nodes. This might seem counterintuitive at first. After all, it is the feedforward nature of neural networks that makes the order of computation unambiguous. However, recurrent edges are defined in a precise way that ensures that no such ambiguity can arise. Recurrent neural networks are *unrolled* across time steps (or sequence steps), with the *same* underlying parameters applied at each step. While the standard connections are applied *synchronously* to propagate each layer's activations to the subsequent layer *at the same time step*, the recurrent connections are *dynamic*, passing information across adjacent time steps. As the unfolded view reveals, RNNs can be thought of as feedforward neural networks where each layer's parameters (both conventional and recurrent) are shared across time steps. Like neural networks more broadly, RNNs have a long discipline-spanning history, originating as models of the brain popularized by cognitive scientists and subsequently adopted as practical modeling tools employed by the machine learning community.

The name of LSTM refers to the analogy that a standard RNN has both "long-term memory" and "short-term memory". The connection weights and biases in the network change once per episode of training, analogous to how physiological changes in synaptic strengths store long-term memories; the activation patterns in the network change once per time-step, analogous to how the moment-to-moment change in electric firing patterns in the brain store short-term memories. The LSTM architecture aims to provide a short-term memory for RNN that can last thousands of timesteps, thus "long short-term memory".

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

Methodology:

1. Load the basic libraries and packages
2. Load the dataset
3. Analyse the dataset
4. Apply LSTM Model
5. Apply the training over the dataset to minimize the loss
6. Observe the cost function vs iterations learning curve

Results:

To be attached with

a. Model Summary
b. Training and Validation accuracy v/s epochs
c. Training and Validation loss v/s epochs

Program (Code):

To be attached with

Observation and Result Analysis:

a. Nature of the dataset

b. During Training Process

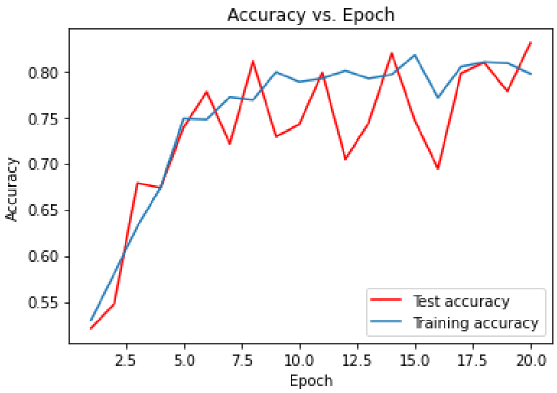
c. After the training Process

d. Observation over the Learning Curve

Post Lab Exercise:

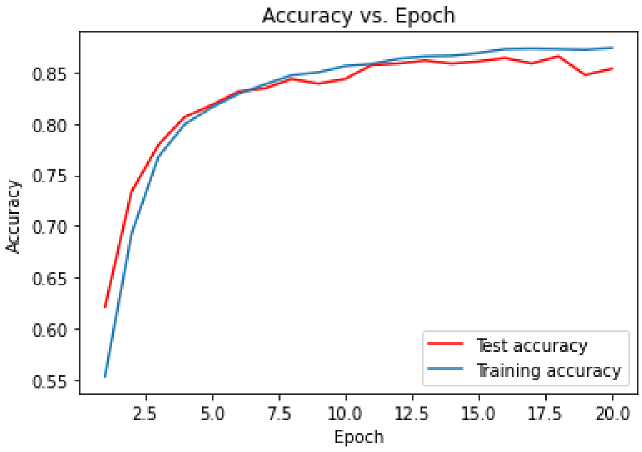
a. What is the requirement of LSTM over RNN?

b. Interpretation of graph



What is the meaning of the above graph? How can the graph be smoothened?

c.



What is the meaning of the above graph? How can the graph be smoothened?

Post Lab Activity

Obtain the prediction value of the TCS stock for the historical data obtained from the link mentioned as: https://in.investing.com/equities/tata-consultancy-services-historical-data

Select the Date Range from (01-01-2020 to 31-03-2024). Select the price range to be as “DAILY” stock values. Train your LSTM model to obtain the predicted value as close as the actual value. Paste the screenshot of the output