**Long Short-Term Memory (LSTM)**

SEMINAR REPORT

Submitted By

**ANLIN ALBERT**

**(Reg No: MAC20MCA2003)**

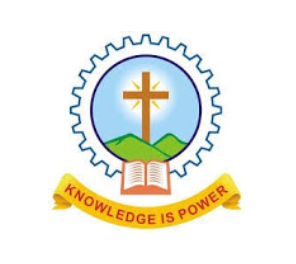
To

The APJ Abdul Kalam Techological University

In partial fulfillment of the requirements of the award of the Degree

Of

Master of Conputer Application



**Department of Computer Application**

Mar Athanasisus College of Engineering

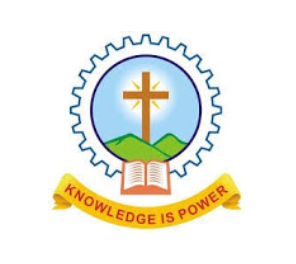
Kothamangalam, Kerala, India 686666

June 2022

**DEPARTMENT OF COMPUTER APPLICATION**

**MAR ATHENASIUS COLLEGE OF ENGINEERING**

**KOTHAMANGALAM**



**CERTIFICATE**

This is to certify that the seminar report entitled “Long Short-Term Memory (LSTM)” is a bonafide record of the work done by Anlin Albert (MAC20MCA2003) in the partial fulfillment of the requirements for the award of the Degree of MASTER OF COMPUTER APPLICATION in Computer Science and Engineering of APJ Abdul Kalam Technology University.

**Faculty Guide Head of the Department**

Prof Sonia Abraham Prof Biju Skaria

Assoc. Professor Assoc. Professor

MCA Dept. MACE MCA Dept. MACE

**Internal Examiner 1 Internal Examiner 2**

Prof Shinu S Kurian Prof Sonia Abraham

MCA Dept. MACE MCA Dept. MACE

**ACKNOWLEDGEMENT**

First and foremost, I thank God Almighty for his divine grace and blessings in making all this possible. May he continue to lead me in the years to come.

I am highly indebted to Prof. Biju Skaria, Associate Professor, Head of Computer Application Department for his guidance and constant supervision as well as for providing necessary information regarding the seminar and also for his support.

I would like to express my special gratitude and thanks to Seminar coordinators Prof. Shinu S Kurian, Assistant Professor, Department of Computer Application and Prof. Sonia Abraham, Associate Professor, Department of Computer Application for giving me such attention and time.

I profusely thank other Asst. Professors in the department and all other staffs of MACE, for their guidance and inspirations throughout my course of study. No words can express my humble gratitude to my beloved parents who have been guiding me in all walks of my journey. My thanks and appreciations also go to my friends and people who have willingly helped me out with their abilities.

**ABSTRACT**

Long Short-Term Memory is an advanced version of recurrent neural network (RNN) architecture that was designed to model chronological sequences and their long-range dependencies more precisely than conventional RNNs. The major highlights include the interior design of a basic LSTM cell, the variations brought into the LSTM architecture, and a few applications of LSTMs that are highly in demand such as fall detection, and human activity recognition.

LSTM has been so designed that the vanishing gradient problem is almost completely removed, while the training model is left unaltered. Long-time lags in certain problems are bridged using LSTMs which also handle noise, distributed representations, and continuous values. LSTMs provide us with a large range of parameters such as learning rates, and input and output biases. Hence, no need for fine adjustments.

The LSTM contains special units called memory blocks in the recurrent hidden layer. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information. Each memory block in the original architecture contained an input gate and an output gate. The input gate controls the flow of input activations into the memory cell. The output gate controls the output flow of cell activations into the rest of the network. Later, the forget gate was added to the memory block. This addressed a weakness of LSTM models preventing them from processing continuous input streams. The forget gate scales the internal state of the cell before adding it as input to the cell through the self-recurrent connection of the cell, therefore adaptively forgetting or resetting the cell’s memory. In addition, the modern LSTM architecture contains peephole connections from its internal cells to the gates in the same cell to learn the precise timing of the outputs

1. **Introduction**

The term "LSTM" refers to a type of artificial neural network used in deep learning and artificial intelligence. LSTM features feedback connections as opposed to typical feedforward neural networks. Such a recurrent neural network may analyze complete data sequences in addition to single data points (such as photos) (such as speech or video). For instance, LSTM can be used to perform tasks like networked, unsegmented handwriting identification, speech recognition, machine translation, robot control, video games, and healthcare. The most frequently used neural network of the 20th century is LSTM.

A cell, an input gate, an output gate, and a forget gate make up a typical LSTM unit. The three gates control the flow of information into and out of the cell, and the cell remembers values across arbitrary time intervals.

Since there may be lags of uncertain length between significant occurrences in a time series, LSTM networks are well-suited to categorizing, processing, and making predictions based on time series data. To solve the vanishing gradient problem that might occur when training conventional RNNs, LSTMs were created. The advantage of LSTM over RNNs, hidden Markov models, and other sequence learning techniques in many applications is their relative insensitivity to gap length.

1. **Supporting Literature**

**2.1. Literature Review**

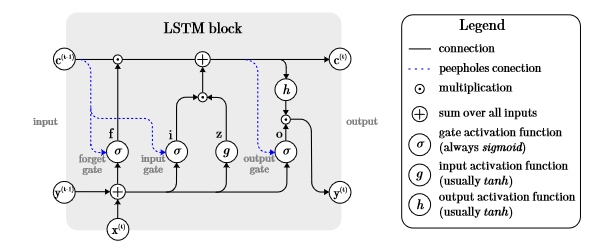
**Abstract**: The fields of machine learning and neurocomputing have both been revolutionised by long short-term memory (LSTM). Several internet sources claim that this methodology has significantly enhanced Google Translate's machine translations, Alexa's responses, and speech recognition. Facebook also uses this neural network, and as of 2017, it was performing more than 4 billion LSTM-based translations every day. It's interesting to note that recurrent neural networks performed rather discretely prior to the appearance of LSTM. The capacity of this recurrent network to handle the exploding/vanishing gradient problem, which is a challenging problem to be avoided when training recurrent or very deep neural networks, is one factor in its success.

**Introduction**: It is challenging to train recurrent or extremely deep neural networks because they frequently experience the exploding/vanishing gradient problem. The LSTM architecture was developed to address this issue while learning long-term dependencies. The learning capability of LSTM had a significant theoretical and practical impact on many domains, making it a cutting-edge model. Due to its high applicability and popularity, this neural architecture has also found its way into the world of gaming.

**Long Short-Term Memory**: The LSTM model is an effective recurrent neural system that was created specifically to solve the exploding/vanishing gradient difficulties that frequently occur while learning long-term dependencies, even when the smallest time lags are quite long. Using a constant error carousel (CEC), which keeps the error signal within each unit's cell, may often prevent this. In actuality, these cells are recurrent networks in and of themselves, with an intriguing architecture that results from the CEC's extension to include the input gate and output gate, which together make up the memory cell. Feedback with a one time step lag is indicated by the self-recurrent connections.

A cell, an input gate, an output gate, and a forget gate make up an LSTM unit. The LSTM network did not originally include this forget gate, but this was suggested so that the network could reset its state. The three gates control the flow of information associated with the cell, and the cell remembers values across arbitrary time intervals.

In a nutshell, the LSTM architecture is made up of a number of memory blocks that are recurrently connected sub-networks. The memory block's purpose is to keep its state constant over time and control information flow using non-linear gating units.



The so-called highway networks used the gating technique to enable unrestricted information flow across numerous tiers. Another proof-of-concept demonstrating the functionality of the gates, this may be said to be. Although LSTM already performs very well, various works investigated the potential for performance enhancement.

The LSTM model discussed employs Graves and Schmidhuber's complete gradient training method to modify the network's learnable parameters (weights). The weights connecting the various parts of the network are calculated via Backpropagation Through Time.

**Relavent applications**: Both individually and in conjunction with other deep learning designs, the LSTM network is used in a wide range of problem domains. One of the most sophisticated networks for processing temporal sequences is the LSTM, as was previously mentioned. Since it is possible to integrate the LSTM with other networks to generate hybrid models, it is still one of the most widely used network options. Any issue requiring temporal memory can be handled by LSTM, including time series forecasts.

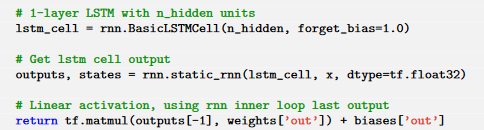
**Time series prediction**: Time series data come to mind first when thinking about temporal sequences in data. This is a broad concept, though. The LSTM model has been applied to financial market predictions using a standard random forest, a conventional deep neural network, and a standard logistic regression in the more literal sense of time series predictions. Receiving a time series as input does not necessarily mean the model will predict the next values in the series, as it can also be used to train a classifier.

**Natural Language Processing**: LSTM is a formidable force in the field of context-free and context-sensitive language acquisition. The study of natural language understanding and manipulation by computers to carry out useful tasks is known as natural language processing. For instance, dialogue systems, usually referred to as conversational agents, let people communicate verbally with a machine. The LSTM model was used for the first time to do speech recognition because its key advantage is its ability to handle lengthy time delays. In this experiment, outcomes similar to those of the hidden Markov model (HMM) were attained.

**Sentiment analysis**: Natural language processing and sentiment analysis go hand in hand. Emotions can be detected using a variety of data sources, including physiological information, environmental factors, movies, and more. Sensor signals from these multi-modal data sources were used by many. These signals came specifically from wearable technology and smartphones. In fact, they were the first to recognise emotions based on physiological, environmental, and geographic information. Four models, each based on a CNN-LSTM architecture, were created to analyse all the data: one for the on-body data, one for the environment, one for location, and one for the fusion of all the data inputs.

The accuracy level was found to have enhanced by more than 20% when employing this hybrid network as opposed to a conventional multi-layer perception model.

**Implementation of LSTM**: The softmax() function is used to normalise the LSTM's output vector of probabilities for the subsequent symbol. The anticipated index of the symbol in the reverse dictionary is that of the element with the highest probability. The application's main component is this model, which Tensorflow makes incredibly easy to implement.



In the training process, at each step, three symbols are retrieved from the training data to form the input vector. These three symbols are converted to numeric values.

**2.2. Findings and Proposals**

The study has shown that this recurrent system is capable of handling a wide range of issues, including sentiment analysis, computer vision, time series forecasting, text recognition, natural language processing, picture and video captioning, and text recognition. It was discovered that combining CNNs with LSTM in order to achieve the best performance is a typical strategy when modelling the majority of these issues.

Convolution and pooling layers were utilised in such hybrid models to drastically eliminate representational redundancy while reducing the problem's dimensionality. Additional architecture customization might always be used to increase precision.

Based on the study, the learning rate is the most significant hyperparameter in the backpropagation algorithm, while the forget gate and output transfer function are the most crucial parts of the LSTM block. Therefore, additional research into these elements may result in LSTM variants with enhanced prediction skills. Another equally important study area discusses less computationally intensive learning techniques to modify the parameters that can be learned.

1. **Chapters**

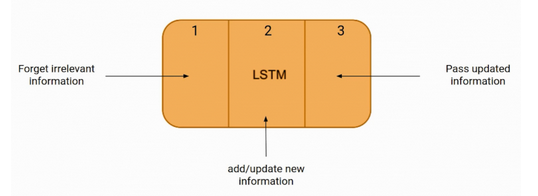
**2.1. Exploding and vanishing gradient**

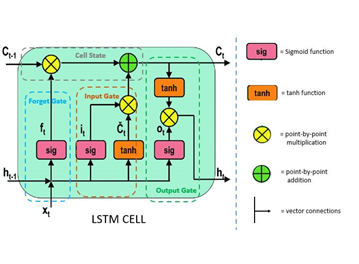
The main goal of network training is to lower the losses (in terms of cost or error) visible in the network's output when training data is fed through it. We first determine the gradient, or loss, in respect to a certain weight set, alter the weights in light of this, and then repeat the procedure until we find the weights that will ensure the loss is as minimal as possible. Reverse-tracking is designed with this in mind. The gradient can occasionally become quite minimal. It is significant to remember that certain characteristics of the layers below affect how much gradient is present in a given layer. The gradient will appear smaller if any component is tiny (less than one). The scaling effect is another name for this. Lower value is produced when this effect is multiplied by the rate of learning, which is a negligible number that lies between 0.1 and 0.001. As a result, the findings are almost unchanged and the weights haven't changed much known as vanishing gradient.

If the weights are modified to be greater than the ideal value and the gradients are severe due to the large components. The problem is also known as the explosive gradients problem. The neural network unit was constructed with the scale factor set to one in order to stop this scaling effect. The cell was later improved with several gating units, and it was given the name LSTM.

**2.2. LSTM Architecture**

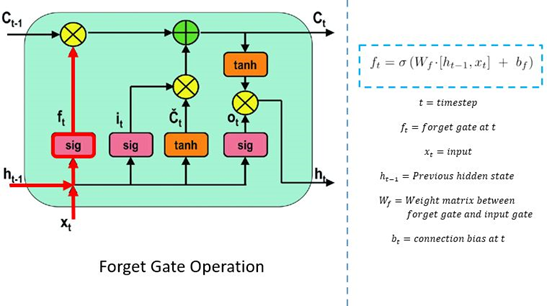
The LSTM network's internal operation is seen below. As seen in the image below, the LSTM is composed of three sections, each of which has a distinct function. The first section determines whether the information from the preceding timestamp needs to be remembered or can be ignored. The cell attempts to learn new information from the input to this cell in the second section. The cell finally transmits the revised data from the current timestamp to the next timestamp in the third section.





Gates refers to these three LSTM cell components. The Forget gate, Input gate, and Output gate are the names of the three components, respectively.

Forget gate:



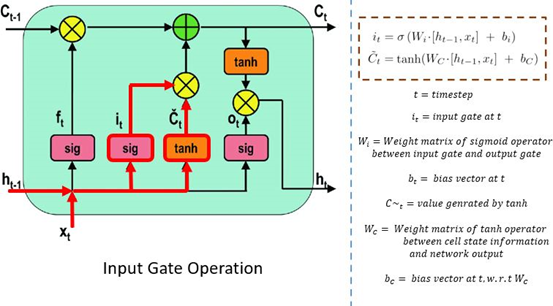
The forget gate is the initial stage of the procedure. In this step, we will determine which pieces of the cell state - the network's long-term memory - are relevant in light of both the prior hidden state and the fresh incoming data.

A neural network is fed with the prior hidden state as well as the fresh input data to do this. This network produces a vector with each member falling within the range [0, 1]. (ensured by using the sigmoid activation). This network (inside the forget gate) is trained to output a value near 0 when an input component is regarded irrelevant and a value closer to 1 when the input component is deemed important. It is helpful to think of each component of this vector as a sort of filter or sieve that lets through more data as the value approaches 1.

The preceding cell state is pointwise multiplied with these output values before being transferred upward. The components of the cell state that the forget gate network has determined to be irrelevant will be multiplied by a value near 0 as a result of this pointwise multiplication, which means they will have less of an impact on the subsequent steps.

In summary, based on the previous concealed state and the new data point in the sequence, the forget gate determines which parts of the long-term memory should now be forgotten (have less weight).

Input gate:



The new memory network and the input gate are involved in the following step. This step's objective is to decide what new information, in light of the prior concealed state and the incoming input data, has to be added to the network's long-term memory (cell state).

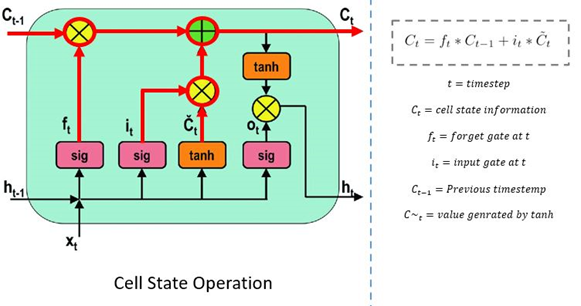
The new memory network is a tanh-activated neural network that has mastered the art of fusing the prior hidden state with fresh input data to produce a "new memory update vector". Given the context from the previous hidden state, this vector essentially contains information from the new input data. Given the new information, this vector indicates how much to update each part of the network's long-term memory (cell state).

As you can see, we are using a tanh because its values fall between and might be negative. If we want to lessen the influence of a component on the cell state, the possibility of negative values is required here.

The second sigmoid function is initially provided the current state X(t) and the previously hidden state h(t-1). Transformed values range from 0 (important) to 1. (not-important).

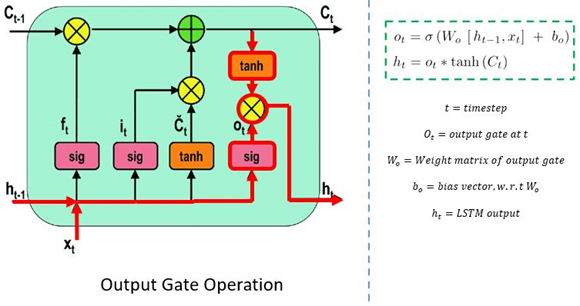
The tanh function will then receive identical data from the hidden state and current state. The tanh operator will build a vector (C(t)) containing every possible value between -1 and 1 to control the network. The output values produced by the activation functions are prepared for multiplication on a point-by-point basis.

Cell state:



The input gate and forget gate have provided the network with sufficient information. Making a decision and storing the data from the new state in the cell state comes next. The forget vector f multiplies the previous cell state C(t-1) (t). Values will be removed from the cell state if the result is 0. The network then executes point-by-point addition on the output value of the input vector i(t), updating the cell state and creating a new cell state C. (t).

Output gate:



The value of the following hidden state is decided by the output gate. Information about prior inputs is contained in this state.

The third sigmoid function is first called with the values of the previous concealed state and the current state. The tanh function is then applied to the new cell state that was created from the original cell state. These two results are multiplied one by one. The network determines which information the hidden state should carry based on the final value. Predictions are made using this hidden state.

The new concealed state and the new cell state are then carried over to the following time step.

The forget gate selects which pertinent information from the earlier processes is required to conclude. The output gates complete the next concealed state, while the input gate determines what pertinent information can be supplied from the current stage.

**2.3. LSTM Applications**

Before being used in real-world applications, LSTM models must be trained using a training dataset. The following is a discussion of some of the most demanding applications:

1. When a word sequence is provided as input for language modelling or text production, words are computed. Language models can be used at several levels, including characters, n-grams, sentences, and even paragraphs.
2. Image processing is the process of analysing a picture and turning the results into sentences. For this, a dataset with a sizable number of images and the correspondingly detailed captions is needed. To forecast the characteristics of the photos in the dataset, a trained model is employed. Data for a photo. The dataset is then analysed such that only the most intriguing terms are included in it. Data in text format. We attempt to fit the model using these two sources of data. By using input words that were previously predicted by the model and the image, the model's task is to produce a descriptive phrase for the picture, one word at a time.
3. Speech and Handwriting Recognition
4. LSTMs forecast musical notes instead of text by studying a combination of supplied notes fed as input in music production, which is relatively similar to that of text generation.
5. Language translation entails translating a sequence from one language to another. Similar to image processing, only a portion of a dataset comprising phrases and their translations is used to train the model once it has been cleaned. The input sequence is converted to a vector representation (encoding) before being output to a translated version using an encoder-decoder LSTM model.

**2.4. LSTM Drawbacks**

As is common knowledge, everything in this world has advantages as well as downsides. The following are some of the disadvantages of LSTMs:

1. Due to their ability to address the issue of vanishing gradients, LSTMs gained popularity. As it turns out, they are unable to entirely remove it. The fact that the data still needs to be transferred from cell to cell for analysis is the difficulty. Additionally, the cell has grown rather sophisticated as a result of the introduction of new features, such forget gates.
2. To be taught and prepared for use in the real world, they need a lot of resources and time. Technically speaking, they require a high memory bandwidth because each cell contains linear layers, which the system typically is unable to provide. Thus, LSTMs become relatively inefficient in terms of hardware.
3. Developers are searching for a model that can retain historical data longer than LSTMs in light of the growth of data mining. The human propensity to break down a certain amount of information into manageable chunks for simple memory serves as the motivation for this type of approach.
4. Because of how different random weight initializations affect them, LSTMs exhibit behaviour that is very similar to that of a feed-forward neural network. Instead, they choose minimal weight initialization.
5. It is challenging to use the dropout technique to address the overfitting problem with LSTMs. A regularisation technique called dropout excludes input and recurrent connections to LSTM units probabilistically from weight and activation updates during network training.

**Conclusion**

Long short-term memory networks are unquestionably an improvement over recurrent neural networks since they are far more creative in their execution of tasks that recurrent neural networks might be able to complete. Long short-term memory is actually a key advancement in LSTM Deep Learning and enhances performance.

You can anticipate more precise predictions and a deeper knowledge of LSTM as these advancements continue to materialise. It dealt with the problem of long-haul conditions of recurrent neural network, where the recurrent neural network can produce more accurate forecasts from the incoming input but cannot foresee the word stored in the long-term memory. A recurrent neural network cannot provide efficient execution as the total length increases.