Long Short-Term Memory (LSTM)

SEMINAR REPORT

Submitted By

ANLIN ALBERT

(Reg No: MAC20MCA2003)

to

The APJ Abdul Kalam Technological University
In partial fulfillment of the requirements of the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

Mar Athanasius College of Engineering

Kothamangalam, Kerala, India 686666

June 2022

DEPARTMENT OF COMPUTER APPLICATIONS

MAR ATHANASIUS 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 APPLICATIONS in APJ Abdul Kalam Technological University.

Faculty Guide
Prof Sonia Abraham
MCA Dept. MACE

Head of the Department
Prof Biju Skaria
MCA Dept. MACE

Internal Examiner 1

Internal Examiner 2

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, Head of the Computer Applications Department and seminar coordinator for his guidance and constant supervision as well as for providing necessary information regarding the seminar and also for his support.

I am also grateful to my faculty guide Prof. Sonia Abraham, Department of Computer Applications for giving me such attention and time.

I profusely thank other Professors in the department and all other staff of MACE, for their guidance and inspiration 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.

List of Tables

Table 1: Comparison of Proposed Model..4

List of Figures

Figure 1: Accelerometer 3
Figure 2: CNN-LSTM 5
Figure 3: RNN-LSTM 7
Figure 4: Activity Recognition RNN 8
Figure 5: Confusion Matrix 9
Figure 6: LSTM 20
Figure 7: LSTM Cell 20
Figure 8: Forget gate 20
Figure 9: Input gate 21
Figure 10: Cell state 22
Figure 11: Output gate 23

Contents

1. Introduction 1
2. Supporting Literature..2
2.1. Literature Review 2
2.2. Findings and Proposals 18
3. Working Principle..19
3.1. Exploding and vanishing gradient 19
3.2. LSTM Architecture..19
3.3. LSTM Applications 25
3.4. LSTM Drawbacks..26
4. Conclusion..27
5. References 28

Chapter 1

1. Introduction

The term "LSTM" refers to a type of artificial neural network used in deep learning and artificial intelligence. Compared to traditional RNNs, Long Short-Term Memory is an enhanced recurrent neural network (RNN) architecture that was created to better accurately simulates chronological sequences and their long-range dependencies. The main highlights include the layout of a fundamental LSTM cell, the modifications made to the LSTM architecture, and a few highly sought-after LSTM applications including fall detection and human activity recognition.

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.

Chapter 2

2. Supporting Literature

2.1. Literature Review

1. Das, S., Partha, S. B., & Imtiaz Hasan, K. N. (2020). Sentence Generation using LSTM Based Deep Learning. 2020 IEEE Region 10 Symposium (TENSYMP).

Abstract: The process of predicting pertinent words in a particular order is served by sentence generation. This study aims to develop a process for producing sentences while upholding correct grammatical structure.

Using the Long Short-Term Memory (LSTM) architecture creates a phrase creation system here. The fundamentals of word embedding are generally followed by the system, where words from the dataset are tokenized and transformed into vector shapes.

Following processing, a layer of long short-term memory is used to store these vectors. After each repetition, the system generates a new set of words. As a result of this process, a sentence or passage will eventually be formed using pertinent words. In comparison to other existing approaches, the system's results are fairly compelling.

Methodology: A suitable dataset with a large number of words is used to train the system. A vocabulary is created using special terms. To create new words that are appropriate for the situation, the labels and features are retrieved, and long short-term memory architecture is then utilized. The general steps are broken down into a few key modules, which are discussed in more detail below.

- Data collection and description

  Collect data and prepare model

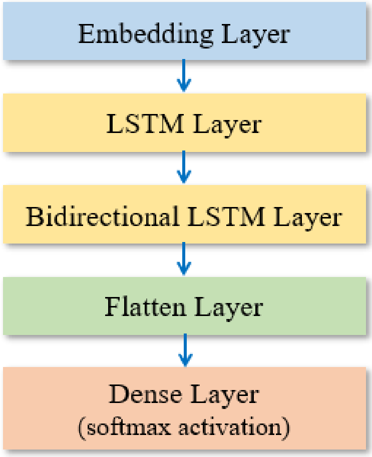
- Processing data for model

  Process collected data and train the model
  - Creating vocabulary
    Here the words are split and a vocabulary of unique words is created
  - Tokenization
    Each word gets tokenized meaning each of the words can be uniquely identified using a number

- N-grams generation
- Padding
  N-grams generated may be of different lengths, here padding is used to acquire uniformity in length
- Retrieving labels and features
- One-hot encoding

Proposed Model: In this paper, the proposed model has several sequential layers

Figure 1: Accelerometer

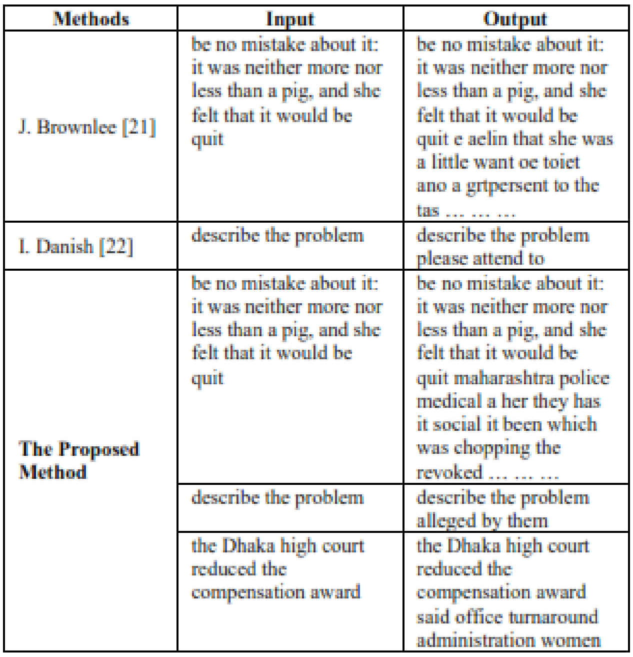


- Embedding layer: In this layer, a set of words is mapped into vector forms to improve the ability of neural networks because working on numerical data is much easier
- LSTM
- Bidirectional LSTM: Bidirectional LSTM has two hidden states working in opposite directions and hence can work on both past and future states at a time
- Flatten layer: This layer takes the output of the previous layer and puts the value in a single vector
- Dense layer: The dense layer in the model uses a softmax activation function which is nonlinear and uses adam optimizer to fit features and labels
- Generating new word: After training the model, input is given to it in tokenized form. The model predicts a new word after each iteration maintaining the context. After generating every new word, the word also is added to the previous input and the new combination is considered the next input.

Result: 80 percent of the dataset is part of the system's training set. The corresponding loss and accuracy rate are noticed after the training data has been fitted into the model. From Fig, it is clear that the rate of loss is steadily declining while the rate of accuracy per epoch is steadily rising.

The table below shows that the first compared model generates words that aren't even real. Additionally, the meaning of the model's output is not particularly clear. The suggested model produces more insightful results for the same input sequence. No non-existent words are generated by the model either. They compared the model in the second comparison and generates an absurd sequence. To make the resilience of the system more understandable, it also shows a third example produced by the suggested system.

Table 1: Comparison of Proposed Model



Methods	Input	Output
J. Brownlee [21]	be no mistake about it: it was neither more nor less than a pig, and she felt that it would be quit	be no mistake about it: it was neither more nor less than a pig, and she felt that it would be quit e aelin that she was a little want oe toiet ano a grtpersent to the tas … … …
I. Danish [22]	describe the problem	describe the problem please attend to
The Proposed Method	be no mistake about it: it was neither more nor less than a pig, and she felt that it would be quit	be no mistake about it: it was neither more nor less than a pig, and she felt that it would be quit maharashtra police medical a her they has it social it been which was chopping the revoked … … …
	describe the problem	describe the problem alleged by them
	the Dhaka high court reduced the compensation award	the Dhaka high court reduced the compensation award said office turnaround administration women

Conclusion: This study presents the idea of developing a deep learning-based model that can produce novel sentences. Word embedding and the Long Short-Term Memory (LSTM) architecture, a modified version of recurrent neural networks, serve as the foundation for all of the methodologies (RNN). The text data are transformed into vector form using word embedding since the neural network finds it easier to operate with numerical data. LSTM and Bidirectional LSTM networks are used to preserve the correct context. Because it can work on both the present and the past at once, the bidirectional LSTM is a crucial layer for the model. The model was effectively trained, and the outcomes were carefully examined and contrasted with those from other models.

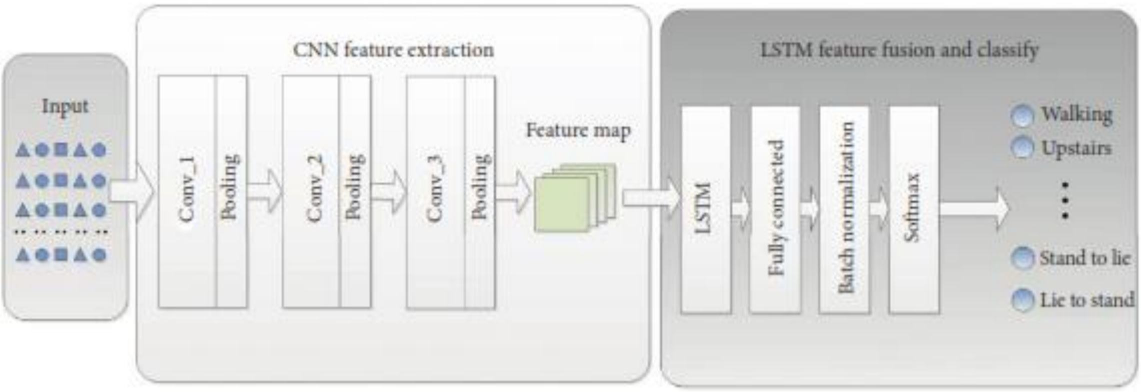
Future Work: In subsequent work, the model can be improved even more so that it can be applied to creating dynamic chatbot suggestions or summarizing material.

2. Wang, H., Zhao, J., Li, J., Tian, L., Tu, P., Cao, T., … Li, S. (2020). Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques. Security and Communication Networks, 2020, 1–12.

Introduction: Human behavior recognition (HAR) is the detection, interpretation, and recognition of human behaviors, which can use smart health care to actively assist users according to their needs. Human behavior recognition has wide application prospects, such as monitoring in smart homes, sports, game controls, health care, elderly patients care, bad habits detection, and identification. It plays a significant role in in-depth study and can make our daily life smarter, safer, and more convenient. This work proposes a deep learning-based scheme that can recognize both specific activities and the transitions between two different activities of short duration and low frequency for health care applications.

Dataset: This paper adopts the international standard Data Set, Smartphone-Based Recognition of Human Activities, and Postural Transitions Data Set to conduct an experiment, which is abbreviated as HAPT Data Set. The data set is an updated version of the UCI Human Activity Recognition Using Popularity Data set. It provides raw data from smartphone sensors rather than preprocessed data and collects data from accelerometer and gyroscope sensors. In addition, the action category has been expanded to include transition actions. The HAPT data set contains twelve types of actions. A total of 815,614 valid pieces of data are used here.

Figure 2: CNN-LSTM



The above figure displays the three-part architecture design for the strategy suggested in this paper. The initial step is the preprocessing and transformation of the raw data, which combines the raw acceleration and gyroscope data into a two-dimensional array that resembles an image. The composite image must then be entered into a CNN network with three layers so that it can automatically detect from the activity image, extract the motion features, abstract the features, and then map them into the feature map. The third step involves feeding the feature vector into the LSTM model, establishing a connection between time and action sequence, and then introducing the entire connection layer to fuse the numerous features. Additionally, Batch Normalization (BN) is presented. BN can normalize the data in every layer before sending it to the Softmax layer for action classification.

This paper also examines transition actions in addition to typical fundamental acts. Transition actions are present in a few accessible data sets. For this reason, the HAPT Data Set, also known as the Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set, is used in this paper to experiment. A newer version of the UCI Human Activity Recognition Using Popularity Data collection is used in this dataset. Instead of preprocessed data, it offers raw data from smartphone sensors. Additionally, transition actions have been added to the category of activities. All of the data without labels were removed after the initial processing of the original data. In the end, 815,614 valid data points were gathered. There is a sizable disparity in data volume between transition action and basic action as a result of the low frequency and brief duration of transition action as well as the high frequency and lengthy duration of fundamental action. The six transition actions account for just about 8% of the overall data, which is significantly less than the data amount of the other basic acts. The initial collection of data is divided into three components: a training set, a verification set, and a test set. The training set is used to train the model, the verification set to alter its parameters, and the test set to assess the quality of the resultant model.

3. Agarwal, P., & Alam, M. (2020). A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices. Procedia Computer Science, 167, 2364–2373.

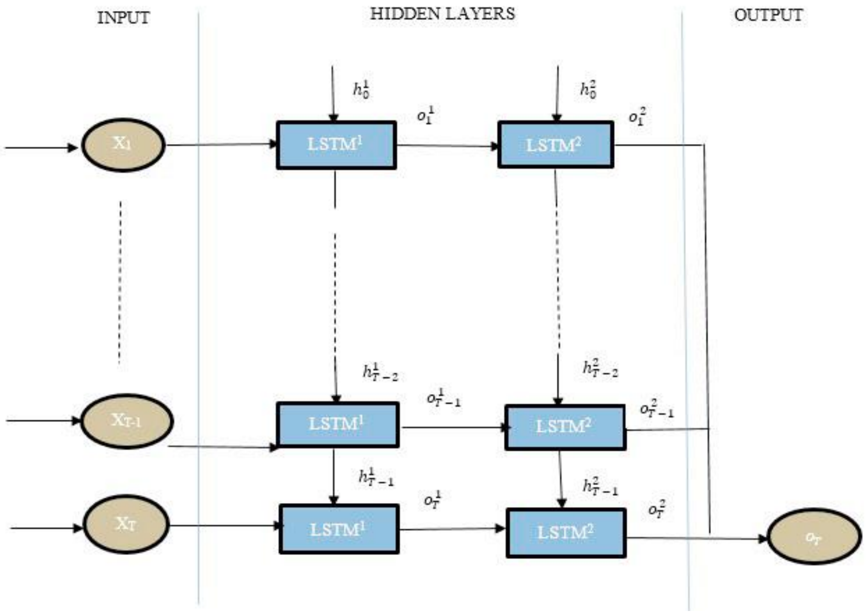
Introduction: Here the architecture for the proposed Lightweight model is developed using Shallow Recurrent Neural Network (RNN) combined with Long Short-Term Memory (LSTM) deep learning algorithm. then the model is trained and tested for six HAR activities on resource-constrained edge devices like RaspberryPi3, using optimized parameters. The experiment is conducted to evaluate the efficiency of the proposed model on the WISDM dataset containing sensor data of 29 participants performing six daily activities: Jogging, Walking, Standing, Sitting, Upstairs, and Downstairs. And lastly, the performance of the model is measured in terms of accuracy, precision, recall, f-measure, and confusion matrix and is compared with certain previously developed models.

Dataset: Here Android smartphone having an inbuilt accelerometer is used to capture tri-axial data. The dataset consists of six activities performed by 29 subjects. These activities include walking, upstairs, downstairs, jogging, standing, and sitting. Each subject performed different activities by carrying a cell phone in the front leg pocket. A constant Sampling rate of 20 Hz was set for the accelerometer sensor. A detailed description of the dataset is given in table 1 below.

- Total no of samples: 1,098,207
- Total no of subjects: 29
- Activity   Samples: Percentage
- Walking   4,24,400   38.6%
- Jogging   3,42,177   31.2%
- Upstairs   1,22,869   11.2%
- Downstairs   1,00,427   9.1%
- Sitting   59,939   5.5%
- Standing   48,397   4.4%

Proposed method: RNN and LSTM are used to create the proposed model. With only two hidden layers and 30 neurons, it has a shallow structure that makes it practical to install on edge computing devices such as IoT boards (Raspberry Pi, Audrino, etc.), Android, and iOS-based resource-constrained devices.

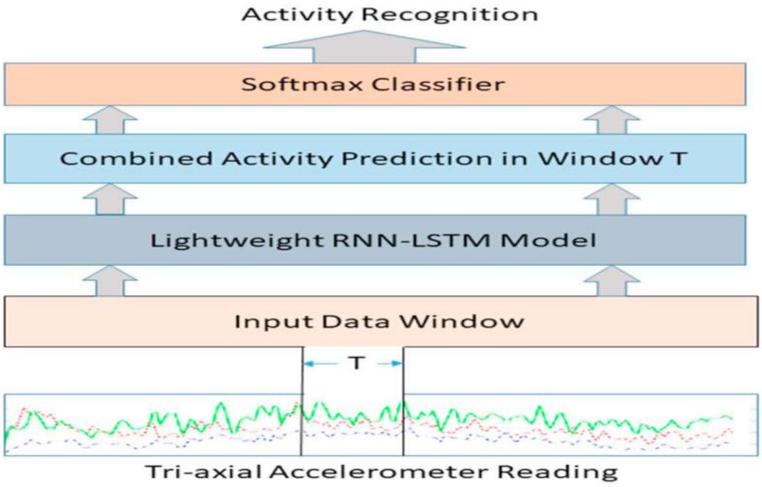
Figure 3: RNN-LSTM



The below figure depicts the operation of a lightweight RNN-LSTM-based HAR system for edge devices. The reading from the accelerometer is divided into fixed windows of size T. A collection of readings (x1, x2, x3..., xT-1, xT) recorded in time

T, where xt is the reading recorded at any time instance t, serves as the input to the model. Readings from this segmented window are then fed into a lightweight RNN-LSTM model. The model combines the output from many states using a softmax classifier to create a single final output for that specific window as oT.

Figure 4: Activity Recognition RNN

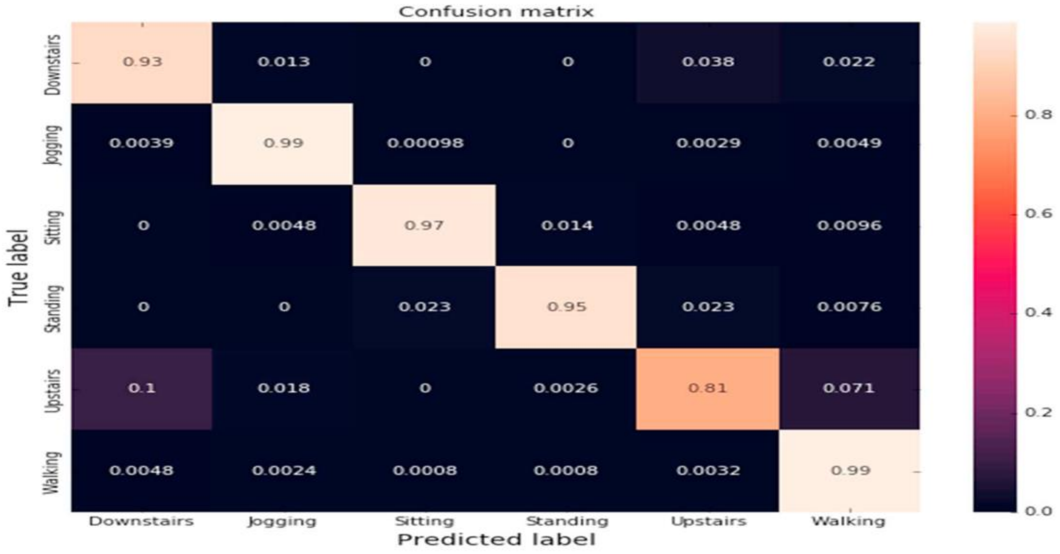

Activity Recognition
Softmax Classifier
Combined Activity Prediction in Window T
Lightweight RNN-LSTM Model
Input Data Window
T
Tri-axial Accelerometer Reading

Wisdm dataset, split into 70:30 for training and testing, is used to train the lightweight RNN-LSTM. According to the activation function, the model's weights are updated. The cost function between the predicted labels and the ground truth is the mean cross-entropy. The Adam optimizer is employed to update model parameters and minimize the cost function. This model was developed on a Raspberry Pi 3 to test its edge device compatibility. Using the hit-and-trial method, different combinations of parameters, including the number of epochs, batch size, window size, and learning rate, were examined.

- Performance metrics include – Accuracy, Precision, Recall & F1 Score

Result: The evaluation findings for the Lightweight RNN-LSTM model are presented in this part, along with comparisons to some of the earlier efforts. The below figure depicts the model's confusion matrix. For walking and jogging exercises, the Lightweight RNN-LSTM obtained a 99 percent accuracy. For upstairs action, a minimum accuracy of 81 percent is attained.

Figure 5: Confusion Matrix



Equations are used to determine accuracy, precision, recall, and f1-score. Accuracy, precision, recall, and F1-score for Lightweight RNN-LSTM were all 95.78 percent, 95.81 percent, and 95.73 percent, respectively. Recall and precision are calculated to validate performance since accuracy could generate false results if the data in each class of dataset is unbalanced.

Conclusion: In this paper, a lightweight HAR model is built. This model is implemented on a Raspberry Pi3 edge device. Communication delay, costs, and network traffic are all decreased when human activities are recorded on edge devices. In comparison to numerous other machine learning and deep learning models, the proposed model yields better outcomes

Future Work: This research can be expanded in the future to distinguish more intricate behaviors. It may be installed on different edge devices running iOS or Android. Static sliding window architecture was used in the development of this model. Future testing of this design can also include a dynamic windowing mechanism. A single tri-axial accelerometer was also used in the development of this system. Multi-sensor data can be supported by expanding it.

4. Abobakr, A., Hossny, M., Abdelkader, H., & Nahavandi, S. (2018). RGB-D Fall Detection via Deep Residual Convolutional LSTM Networks. 2018 Digital Image Computing: Techniques and Applications (DICTA).

Introduction - A robust fall detection system can be defined as an assistive device that aims at detecting and alerting fall incidents. Hence, its main objective is to distinguish between fall events and normal activities of daily living. The significant similarities of some ADL activities to falls challenge the robustness of fall detection systems. Accelerometer devices are the most commonly used wearable sensors for fall detection. Readings from the accelerometer attached to the human body are evaluated using thresholding or machine learning methods to detect fall events. Despite the effectiveness of wearable devices, they have limitations such as battery lifetime, being easily disconnected, and being forgotten. Moreover, wearing electronic devices is not preferable for the aging societies. Context-aware systems, on the other hand, rely on sensors deployed in the environment such as floor vibration sensors, microphones, and cameras to detect falls.

This paper proposes a vision-based integrable and automated fall detection system. The fall events are detected using an end-to-end deep machine learning model composed of convolutional and recurrent neural networks. The deep convolutional network extracts visual features from input sequences of depth frames. We use a ConvNet architecture that follows the residual learning approach to optimize the visual representations.

Dataset - The UR Fall Detection Dataset is used for training and evaluating the performance of the proposed method. So, the experiments have been conducted on the URFD dataset only. We split each of these activity classes into 80% for training and 20% for validation.

This dataset contains 70 (30 falls + 40 activities of daily living) sequences. Fall events are recorded with 2 Microsoft Kinect cameras and corresponding accelerometric data. ADL events are recorded with only one device (camera 0) and an accelerometer. Sensor data was collected using PS Move (60Hz) and x-IMU (256Hz) devices. The dataset is organized as follows. Each row contains a sequence of depth and RGB images for camera 0 and camera 1 (parallel to the floor and ceiling-mounted, respectively), synchronization data, and raw accelerometer data. Each video stream is stored in a separate zip archive in form of a png image sequence.

- Synchronization data contains frame number, time in milliseconds since sequence start, and interpolated accelerometric data.
- Raw accelerometric data contains time in milliseconds since sequence start and accelerometer data.

Proposed method – This paper proposes an end-to-end ConvLSTM model for fall event detection. This model combines a deep residual convolutional network ResNet with a recurrent LSTM neural network module. ConvNet architecture that follows the residual learning approach (ResNet) to learn discriminative features from articulated body postures is used. The motivation for this combination is twofold. First, the ResNet model has powerful capabilities to learn and extract deep hierarchical visual features from raw input images. Second, using the extracted body features, the LSTM module can learn long-term temporal dynamics that can discriminate sequential input data, e.g., fall events.

- Preprocessing depth sequences
    - It has been concluded that depth images provide weak local gradient information of objects which makes it difficult for deep learning models to generalize and biases the ConvNet model towards detecting objects silhouettes. Therefore, several depth encoding methods have been proposed to make efficient use of depth measurements and provide a better learning signal for the deep network. combined the depth map as an additional channel with the RGB image forming an RGB-D modality. In, depth representation using the HHA encoding was proposed. Learning from HHA encoded maps have demonstrated better results than using raw depth data. This method spreads depth measurements over three RGB color channels. The values of RGB color components vary according to the distance from the depth camera, and hence, provide a more powerful input signal to the ConvNet model, ResNet in this work. Finally, an RGB color map is applied to produce the colorized depth image.
- Deep feature extraction using ResNet
    - ConvNets are the basic building block for the state-of-the-art methods for visual perception tasks such as object recognition, localization and detection, and semantic segmentation. This stack of layers learns multiple levels of feature extractors with increasing levels of abstraction. Incorporating ConvNets with the residual learning paradigm has led to the ResNet architecture which ensures better and faster generalization performance, easier optimization, and makes efficient use of network depth.

        - The ResNet model is composed of residual blocks where each block learns a residual mapping regarding its input, instead of learning a direct unreferenced mapping. The layers of a residual block are formulated as CONV layers. This approach provides a strong initialization that reduces the effect of overfitting due to the small number of videos in fall detection datasets.

- LSTM modeling temporal dynamics
    - In addition to the hidden state, it incorporates a memory unit or cell state that is continuously modified using non-linear gating functions, which are learned. These gating functions manipulate the memory unit through forget and update operations to allow storing only relevant information.

5. Bulbul, E., Cetin, A., & Dogru, I. A. (2018). Human Activity Recognition Using Smartphones. 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT).

Introduction - Smartphones are the most useful tools in our daily life and with the advancing technology they get more capable day by day to meet customer needs and expectations. An accelerometer has been standard hardware for almost all smartphone manufacturers. Since there is a meaningful difference in characteristics between data retrieved from these sensors, many features could be generated from these sensors' data to determine the activity of the person that is carrying the device. In this study, a dataset consisting of signals from the accelerometer and gyroscope of a smartphone carried by different men and women volunteers while doing different activities are classified using different machine learning approaches.

Dataset - Dataset consists of signals from a smartphone carried by 9 individuals performing 6 different activities. Activities performed are listed below with their corresponding codes.

- WALKING
- CLIMBING UP THE STAIRS
- CLIMBING DOWN THE STAIRS
- SITTING
- STANDING
- LAYING

Signals are recorded with a sampling rate of 50Hz and stored as time series for each dimension so 6 different signals were obtained (3 are from the accelerometer and the other 3 are from the gyroscope). The noise was filtered using median and 20Hz Butterworth filters to get more precise results. A second 3hz Butterworth filtering was applied to eliminate the effect of gravity in accelerometer signals. Values then normalized to (-1,1) interval. Euclid magnitudes of the values of 3 dimensions were calculated to merge 3-dimensional signals into one dataset. Finally, class codes (activity codes) given above for each row are added at the end of them along with the number that is given to each individual. In the end, the dataset consists of 2947 records with 561 features.

Proposed method - Supervised machine learning is used to recognize activity from dataset records. Different supervised machine learning models are designed using different classification approaches. Designed models first trained with training data that consists of 80% of the total dataset and then tested with the rest. Classification precision of models is tested and observed using 5-fold cross-validation. Methods used for classification are as follows:

- Decision Trees
- Support Vector Machines
- K-nearest neighbors (KNN)
- Ensemble classification methods
    - Boosting
    - Bagging
    - Stacking

6. Shojaei-Hashemi, A., Nasiopoulos, P., Little, J. J., & Pourazad, M. T. (2018). Video-based Human Fall Detection in Smart Homes Using Deep Learning. 2018 IEEE International Symposium on Circuits and Systems (ISCAS).

Introduction - The concept of a “smart home” is a major step towards wellness and improved quality of life and a hot interdisciplinary research topic bringing together artificial intelligence, cloud computing, communications and networks, psychology, and healthcare. Monitoring the well-being of the residents is an expected service to be provided by a smart home. Wearable devices are relatively inexpensive and can directly measure kinematic quantities. Nowadays, inexpensive depth cameras, such as the Microsoft Kinect, can address some of the privacy issues and under proper implementation, conditions could be a promising and feasible option for human fall detection in the context of a smart home.

Dataset – NTU RGB+D is a large-scale dataset for RGB-D human action recognition. It involves 56,880 samples of 60 action classes collected from 40 subjects. The actions can be generally divided into three categories: 40 daily actions (e.g., drinking, eating, reading), nine health-related actions (e.g., sneezing, staggering, falling), and 11 mutual actions (e.g., punching, kicking, hugging). These actions take place under 17 different scene conditions corresponding to 17 video sequences (i.e., S001–S017). The actions were captured using three cameras with different horizontal imaging viewpoints, namely, −45∘,0∘, and +45∘. Multi-modality information is provided for action characterization, including depth maps, 3D skeleton joint position, RGB frames, and infrared sequences. The performance evaluation is performed by a cross-subject test that split the 40 subjects into training and test groups, and by a cross-view test that employed one camera (+45∘) for testing, and the other two cameras for training.

Proposed method - Because the 3D locations of major body joints carry most of the body kinematic information required for discriminating different actions, keeping track of the body joints, as shown by the evaluations, proves to be sufficient for action recognition and fall detection, while it is computationally much cheaper. Since the existing algorithms to extract skeletons from the depth map, such as the one provided by the Microsoft Software Developer Kit, process the video sequence frame by frame, the use of body skeleton information by the model can be implemented in real-time. As actions usually take place within a long sequence of frames, vanilla RNN encounters the vanishing gradient issue, so LSTM is specifically taken, which can go deep in time.

As a deep neural network, LSTM requires abundant training data. A multi-class LSTM on the abundant samples of human regular actions is initially trained. Then, transfer all the learned weights, except for those of the last layer, to a two-class LSTM that is designed for fall detection. Finally, train the last layer of the two-class LSTM on scarce human fall samples in combination with part of the regular action samples. To prevent the LSTMs from getting biased toward training data. These include the depth of the LSTM in time, the number of layers, the number of the hidden units in each layer, and the dropout ratio.

7._Alemayoh, T. T., Hoon Lee, J., & Okamoto, S. (2019). Deep Learning-Based Real-time Daily Human Activity Recognition and Its Implementation in a Smartphone. 2019 16th International Conference on Ubiquitous Robots (UR).

Introduction - Human activity recognition is a broad area of study mainly concerned with identifying specific movement or action of a person based on given input data. Mostly, input data signals are obtained from videos, where video frames are taken for analysis or multi-axis time-series IMU devices. Comparably, wearable IMU sensors became a popular and convenient way of data collection mechanisms without an extensive installation of equipment and privacy issues.

The processed version of the data was used to fit statistical and machine learning models such as SVM as in Anguita et al. Deep learning methods have shown the capability and even achieve state-of-the-art results by automatically learning high-level and meaningful features from raw data. In large-scale data classification, CNN is competent to extract features from signals and it has demonstrated excellent performance in image classification, speech recognition, and sentence classification.

Dataset – The smartphone used was attached tightly to the waist of the subjects. Out of the various motion-related sensors of a smartphone, the 3-axis Acc and 3-axis Gyro were chosen for a better result. Motion data of eight activities were collected. The activities are: walking, jumping, running, bicycle riding, stairs ascending, and descending, laying down, and still.

Proposed method - A CNN is applied to the activities' one-dimensional virtual images prepared. Each convolutional layer performs a 2D convolution on its inputs followed by a non-linear activation function, ReLU (Rectifier Linear Unit). To reduce the effect of internal covariance shift of activations, batch normalization was utilized, which forces each mini-batch input of a layer to have similar distribution throughout the hidden layers. Besides, it allows the use of larger learning rates to speed up the optimization process. After the output of the second pooling is flattened to form a long 1D feature map vector, the classification is decided by the probability distribution of an eight-class softmax layer. All the parameters of the network are updated by Adam optimizer using back optimization.

8. Ullah, M., Ullah, H., Khan, S. D., & Cheikh, F. A. (2019). Stacked Lstm Network for Human Activity Recognition Using Smartphone Data. 2019 8th European Workshop on Visual Information Processing (EUVIP).

Introduction - With the exponential growth of computing technology, wear-able electronics are widespread in human communities for daily usage. With the daily usage of the smartphone, embedded sensors like accelerometers and gyroscopes produce a large amount of useful data that can be used to automatically predict and classify human activities. Potentially, human activity recognition can be used in elderly houses, especially in the countries where the average old population is on the rise. In essence, the human activity algorithm can be divided into the following two broad categories – Vision based & Sensor based

Dataset - The network is evaluated on a public domain UCI dataset and quantitative results are compared against six state-of-the-art methods. The performance is calculated in terms of precision-recall and average accuracy.

Proposed method – The method mainly consists of two parts i.e. a single layer neural network and a network of stacked LSTM cells. Initially, sensor data is obtained from the smartphone that is worn by a human subject. Here two types of sensor data i.e. accelerometer and the gyroscope. The raw sensor data is passed through a single-layer neural network which acts as a pre-processing and normalized input data for the succeeding network. The normalization is achieved through a linear discriminant function and ReLU activation. After that, the data is fed to the stacked LSTM network. The network consists of five LSTM cells that have learned the temporal dependencies of the sensor sequential data. The output of the stacked LSTM network is given to a six-way softmax which gives the individual probability of the six human behavior i.e. (walking, walking upstairs, walking downstairs, sitting, standing, lying).

9. Sun, B., Liu, M., Zheng, R., & Zhang, S. (2019). Attention-based LSTM Network for Wearable Human Activity Recognition. 2019 Chinese Control Conference (CCC).

Introduction - Human activity recognition is an important area of research in ubiquitous computing, human behavior analysis, and human-computer interaction. It can be used widely, including in health monitoring, smart homes, and human-computer interactions. HAR focuses on the motion data from smart sensors such as accelerometers, gyroscopes, Bluetooth, light sensors, and so on. Although the video-based recognition method excels other recognition methods in indoor activity, it has several restrictions such as space limitation and interference from the environment. With the development of sensor technology and computing power, the sensor-based HAR is becoming more promising with privacy well-protected.

In this paper, an LSTM network with an attention mechanism, which can automatically focus on the time series that has a decisive effect on classification, to capture the most important temporal dependencies from the input, without using extra handcrafted features and human domain knowledge. The attention mechanism allows a model to learn a set of weights over raw sensor data, which is used to leverage weight of the temporal context.

Dataset - The Opportunity data set for HAR from wearable, object, and ambient sensors is a data set devised to benchmark human activity recognition algorithms. The data sets contain- s activities from 4 subjects and each has 6 different runs. Five of them, termed activity of daily living, are composed of temporally unfolding situations in which a large number of action primitives occur. The remaining one, a drill run, is designed to generate a large number of scripted sequence instances.

Notably, the use 17 mid-level gesture classes for predictions. This group contains the "NULL" class, which is common, for a total of 18 classes.

Proposed method - The model contains four components - Input layer: input sensor data to this model, LSTM layer: utilize LSTM to get high-level features, Attention layer: produce a weight vector, and merge features from each time step into a temporal feature vector to find relevant temporal context by multiplying the weight vector and Output layer: the temporal feature is finally used for activity recognition.

10. Deep, S., & Zheng, X. (2019). Hybrid Model Featuring CNN and LSTM Architecture for Human Activity Recognition on Smartphone Sensor Data. 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT).

Introduction - The proliferation of smartphones with various embedded sensors have eased the method of gathering human activity data in recent time. With the development of unprecedented characteristics of sensors such as accelerometers and gyroscopes, sensor-based human activity recognition has received extensive concerns. In wearable-based HAR, sensors or other external devices are attached to the human body. HAR is a method of predicting activities from the data obtained from sensors. The process involves extracting motion features and classifying the activities into different categories. The data collected from the sensors are a sequence of time series data and traditional machine learning algorithms may not exploit the temporal correlations of input data.

In this paper, a combination of CNN and LSTM for HAR is used. Furthermore, it is also possible to apply LSTM for activity recognition tasks in the same dataset and compare the results with the CNN-LSTM model.

Dataset - To evaluate the effectiveness of the CNN-LSTM model, experiment on the UCI HAR dataset. The dataset consists of time series data collected from 30 volunteers of the 19-48 age group. Each volunteer performed six activities (walking, walking upstairs, walking downstairs, sitting, standing, laying) with a smartphone attached to their waist. The 3-axial linear acceleration (tAcc-XYZ) from the accelerometer and 3-axial angular velocity (tGyro-XYZ) from gyroscope data were collected. The data were collected with a constant sampling rate of 50Hz. The activities were video-recorded for ground truth and data were manually labeled. The dataset is randomly divided into 70% training and 30% testing data.

Proposed method – Here, a kernel size of 6 and several filters 128 for both the convolutional layers. The output data size is then passed through the dropout layer. This output data is further used to pass through the LSTM layer. The output 3D data is fed to the LSTM layer.

The LSTM layer is used in this hybrid architecture because it works well with the time series data and is designed to handle time dependence problems. Each LSTM layer in this architecture produces hidden cell information. The LSTM layer is followed by a dense layer, hyperbolic tangent activation, and a soft-max layer at the end. It is then passed through a dense layer with hyperbolic tangent activation and used Adam optimizer which ends the LSTM networks in this hybrid model.

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 to achieve the best performance is a typical strategy when modeling the majority of these issues.

Convolution and pooling layers were utilized 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.

Chapter 3

3. Working Principle

3.1. 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. First determine the gradient, or loss, concerning 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. A 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 the 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 to stop this scaling effect. The cell was later improved with several gating units, and it was given the name LSTM.

Addressing Vanishing & Exploding gradients

- Use of ReLU
- Proper Weight Initialization
- Use of Gradient Clipping

3.2. 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.

Figure 6: LSTM

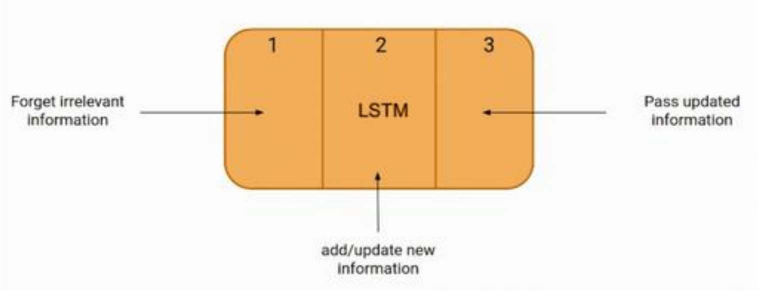
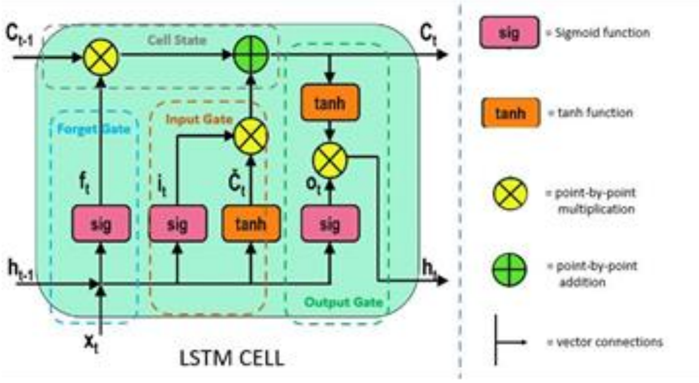


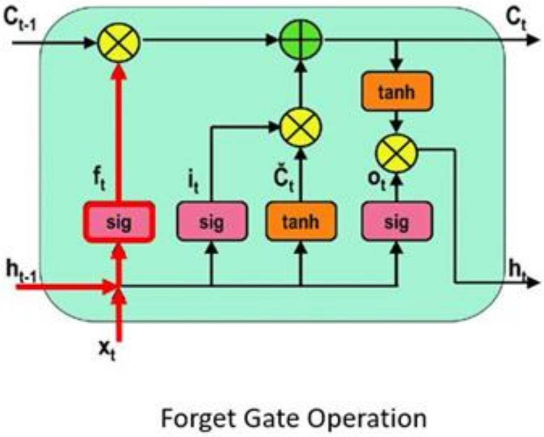
Figure 7: LSTM Cell



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:

Figure 8: 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.

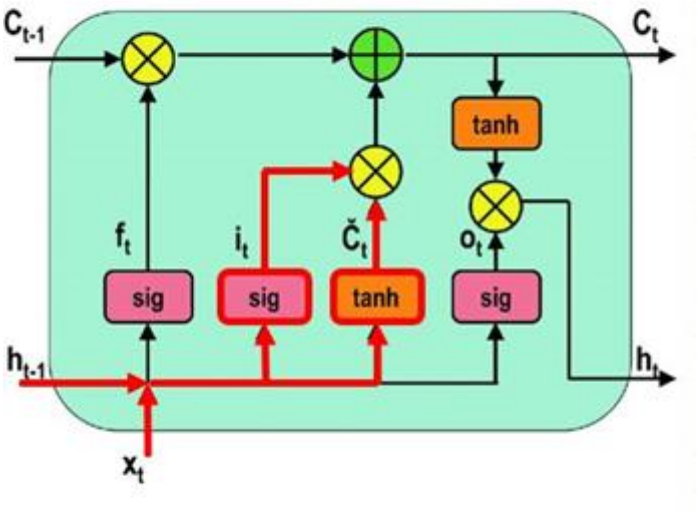
A neural network is fed with the prior hidden state as well as the fresh input data to do this. This network produces a vector with each member falling within the range [0, 1]. (ensured by using the sigmoid activation). This network (inside the forget gate) is trained to output a value near 0 when an 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:

Figure 9: Input gate



Input Gate Operation

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).

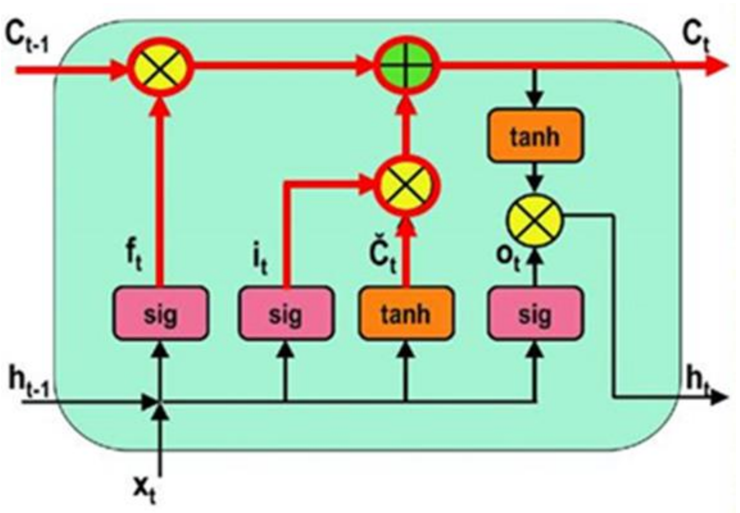
As you can see, we are using a tanh because its values fall between and might be negative. If we want to lessen the influence of a component on the cell state, the possibility of negative values is required here.

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 to 1.

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:

Figure 10: Cell state

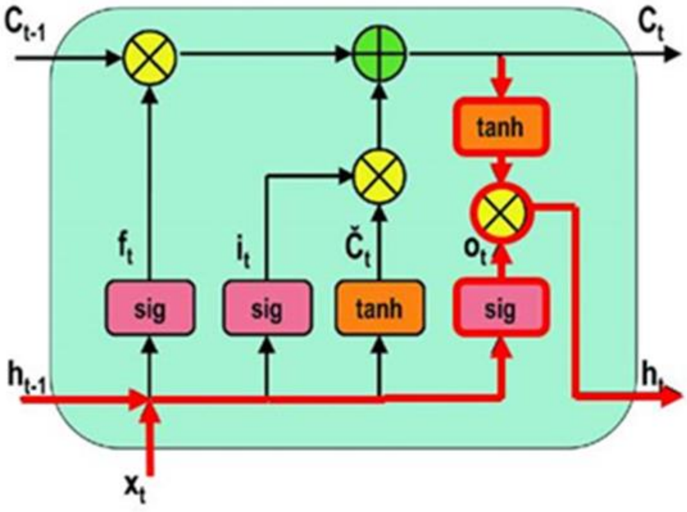


Cell State Operation

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:

Figure 11: Output gate



Output Gate Operation

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.

Sigmoid

Gates contains sigmoid activations. A sigmoid activation is similar to the tanh activation. Instead of squishing values between -1 and 1, it squishes values between 0 and 1. That is helpful to update or forget data because any number getting multiplied by 0 is 0, causing values to disappears or be “forgotten.” Any number multiplied by 1 is the same value therefore that value stay’s the same or is “kept.” The network can learn which data is not important therefore can be forgotten or which data is important to keep.

$$S(x) = \frac{1}{1 + e^{-x}}$$

$$S(x) = sigmoid\ function$$

$$e^{-x} = Euler's\ number$$

Tanh

The tanh activation is used to help regulate the values flowing through the network. The tanh function squishes values to always be between -1 and 1. When vectors are flowing through a neural network, it undergoes many transformations due to various math operations. So imagine a value that continues to be multiplied by let’s say 3. You can see how some values can explode and become astronomical, causing other values to seem insignificant.

A tanh function ensures that the values stay between -1 and 1, thus regulating the output of the neural network. You can see how the same values from above remain between the boundaries allowed by the tanh function.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

3.3. 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. Language Modeling: When a word sequence is provided as input for language modeling 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: It is the process of analyzing a picture and turning the results into sentences. For this, a dataset with a sizable number of images and 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 analyzed 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: It entails translating a sequence from one language to another. Similar to image processing, only a portion of a dataset comprising phrases and their translations are 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.

6. Human Activity Recognition: For the human activity detection it is possible to use Long Short-Term Memory (LSTM), an artificial recurrent neural network architecture, which will automatically learn complicated properties from the raw accelerometer signal to be able to distinguish between common human activities such as walking, standing, sitting etc.

3.4. 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 as 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 behavior 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 regularization technique called dropout excludes input and recurrent connections to LSTM units probabilistically from weight and activation updates during network training.

Chapter 4

4. Conclusion

The long short-term memory model (LSTM) is a special structure type of the RNN model, which adds three control units ("cell"): input gate, output gate, and forget gate. As information enters the model, the cell in LSTM The information will be judged, the information that meets the rules will be left, and the non-compliant information will be forgotten. Based on this principle, the problem of long sequence dependence in the neural network can be solved.

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 a key advancement in Deep Learning and enhances performance.

You can anticipate more precise predictions and a deeper knowledge of LSTM as these advancements continue to materialize. It dealt with the problem of long-haul conditions of the 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.

We can say that as we switch to LSTM, we add more and more control levers that regulate the input flow and mixing in accordance with training weights and adding additional control over the outputs in the process.

The most control-ability and, hence, the best results are provided by LSTM but it increases operating costs and complexity.

Chapter 5

5. References

1. Abobakr, A., Hossny, M., Abdelkader, H., & Nahavandi, S. (2018). RGB-D Fall Detection via Deep Residual Convolutional LSTM Networks. 2018 Digital Image Computing: Techniques and Applications (DICTA).

2. Bulbul, E., Cetin, A., & Dogru, I. A. (2018). Human Activity Recognition Using Smartphones. 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT).

3. Wang, H., Zhao, J., Li, J., Tian, L., Tu, P., Cao, T., … Li, S. (2020). Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques. Security and Communication Networks, 2020, 1–12.

4. Agarwal, P., & Alam, M. (2020). A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices. Procedia Computer Science, 167, 2364–2373.

5. Das, S., Partha, S. B., & Imtiaz Hasan, K. N. (2020). Sentence Generation using LSTM Based Deep Learning. 2020 IEEE Region 10 Symposium (TENSYMP).

6. Shojaei-Hashemi, A., Nasiopoulos, P., Little, J. J., & Pourazad, M. T. (2018). Video-based Human Fall Detection in Smart Homes Using Deep Learning. 2018 IEEE International Symposium on Circuits and Systems (ISCAS).

7. Alemayoh, T. T., Hoon Lee, J., & Okamoto, S. (2019). Deep Learning-Based Real-time Daily Human Activity Recognition and Its Implementation in a Smartphone. 2019 16th International Conference on Ubiquitous Robots (UR).

8. Ullah, M., Ullah, H., Khan, S. D., & Cheikh, F. A. (2019). Stacked Lstm Network for Human Activity Recognition Using Smartphone Data. 2019 8th European Workshop on Visual Information Processing (EUVIP).

9. Sun, B., Liu, M., Zheng, R., & Zhang, S. (2019). Attention-based LSTM Network for Wearable Human Activity Recognition. 2019 Chinese Control Conference (CCC).

10. Deep, S., & Zheng, X. (2019). Hybrid Model Featuring CNN and LSTM Architecture for Human Activity Recognition on Smartphone Sensor Data. 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT).