Analog Integrated Circuits and Signal Processing (2018) 95:467–472
https://doi.org/10.1007/s10470-018-1180-y

A memristor-based long short term memory circuit

Kamilya Smagulova[1] · Olga Krestinskaya[1] · Alex Pappachen James[1]

Received: 16 December 2017 / Revised: 27 March 2018 / Accepted: 28 March 2018 / Published online: 2 April 2018

Abstract

Long-short term memory (LSTM) is a cognitive architecture that aims to mimic the sequence temporal memory processes in human brain. The state and time-dependent based processing of events is essential to enable contextual processing in several applications such as natural language processing, speech recognition and machine translations. There are many different variants of LSTM and almost all of them are software based. The hardware implementation of LSTM remains as an open problem. In this work, we propose a hardware implementation of LSTM system using memristors. Memristor has proved to mimic behavior of a biological synapse and has promising properties such as smaller size and absence of current leakage among others, making it a suitable element for designing LSTM functions. Sigmoid and hyperbolic tangent functions hardware realization can be performed by using a CMOS-memristor threshold logic circuit. These ideas can be extended for a practical application of implementing sequence learning in real-time sensory processing data.

Keywords LSTM · Memristor · Memristor crossbar array

1 Introduction

Growing amount of data requires development of powerful and reliable tools for processing it. Artificial neural networks (ANN) are biologically inspired architectures that outperform most of conventional methods for data processing in many tasks. For instance, feedforward neural networks are quite popular for classification problems [1, 2]. However, they are ineffective in dealing with sequential data with long-term dependencies. Unlike feedforward neural networks, recurrent neural networks (RNNs) possess internal memory and are capable of retaining order of information and sharing the parameters across sequence.

Early RNN architectures were introduced by Hopfiled and Jordan in the 1980s [3, 4]. Presently RNN can be represented as a chain of neural networks each passing information to a successor network (Fig. 1). Current state cell S_1 gets a new input X_1 along with hidden layer information of a previous cell S_0 and produces an output Y_1. The algorithm used to train RNN is called backpropagation through time (BPTT). It counts derivatives of the loss at each timestep and sums it up across time for each parameter. As the gap between timesteps gets bigger, vanishing gradient problem arises [5]. Long Short-term memory (LSTM) is a special configuration of RNN introduced to overcome this vanishing gradient problem [6].

Similar to RNN, LSTM also has a chain-like structure (Fig. 2) but each unit of a chain has a gated structure. Traditional LSTM cell consists of a memory cell to store state information and three gate layers that control flow of information within cells and network.

2 Long sort-term memory circuit architecture

2.1 LSTM structure

Figure 3 shows a basic LSTM cell structure. The core of a unit is an internal state storage. The information in a cell state is updated by forget, input and output layers.

Input gate First of all, a new input data and an output data from a previous cell are concatenated into a single vector. Then the vector goes through input gate, which behavior is described by equation:

✉ Alex Pappachen James
alex.james@nu.edu.kz

[1] School of Engineering, Nazarbayev University, 53, Kabanbay Batyr Ave., Astana, Kazakhstan

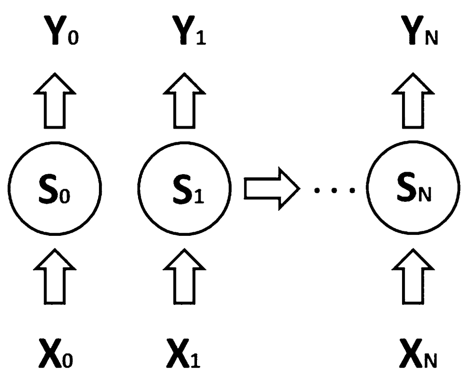

Fig. 1 Unfolded recurrent neural network

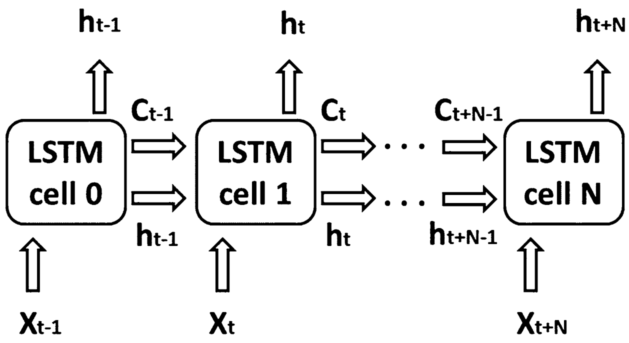

Fig. 2 LSTM network

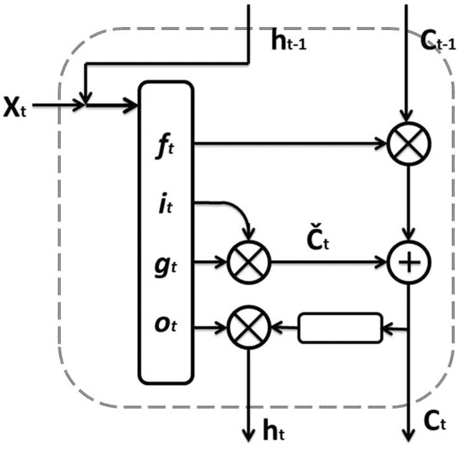

Fig. 3 LSTM unit structure

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)})$$

At the same time, concatenated vector is squashed between − 1 and 1 by applying hyperbolic tangent activation function:

$$g_t = tanh((W^{(g)}x_t + U^{(g)}h_{t-1} + b^{(g)})$$

The obtained output of g_t is elementwise multiplied with the output of the input gate i_t. Since the value of i_t is between 0 and 1, the output $i_t \odot g_t$ acts as a filter which form an intermediate cell state $\tilde{C}_t$.

Forget gate Decision whether to allow information from a previous cell state to a current cell or completely block it, is made by the output of the forget gate, which is given by following equation:

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)})$$

It also takes the values between 0 and 1.

Cell state The internal state of a LSTM cell is a sum of two components—an output of the input gate and output of the forget gate. Since forget gate decides whether to keep information or remove it, LSTM does not suffer from the vanishing gradient problem.

$$c_t = \tilde{C}_t + f_t \odot c_{t-1}$$

Output gate The output of an LSTM cell is a vector h_t. It is a pointwise multiplication of sigmoid layer of an output gate and cell state squashed between − 1 and 1 by hyperbolic tangent activation function:

$$h_t = o_t \odot tanh(c_t),$$

where

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}) .$$

2.2 Matrix-vector multiplication

Matrix-vector multiplication (also known as a Hadamard-product multiplication) is a significant operation in LSTM gating systems and its accuracy plays a vital role. The proposed architecture for matrix-vector multiplication implementation is based on crossbar array using a novel device called "memristor".

Memristor and memristor crossbar array Memristor is a non-volatile two-port device with variable resistor. Its existence was first postulated by Leon Chua in 1971. He predicted a device that maintain the relationship between charge and magnetic flux [7]. In 2008 HP labs announced the discovery of a device that possesses mentioned characteristics [8, 9].

A memristive crossbar array (Fig. 4) consists of a large number of intersecting rows and columns with memristors at junctions. An input vector is applied to a row of a crossbar and multiplied by the conductance of memristors. The output of a crossbar is a sum of currents across each column [10].

Memristor neuron circuit Figure 5 shows memristor neuron circuit with three inputs and one output. Each input is connected to a pair of memristors with conductances σ^{+} and σ^{-}. If $\sigma^{+} > \sigma^{-}$ then resulting memristors conductance gives a positive weight, otherwise a negative weight [11].

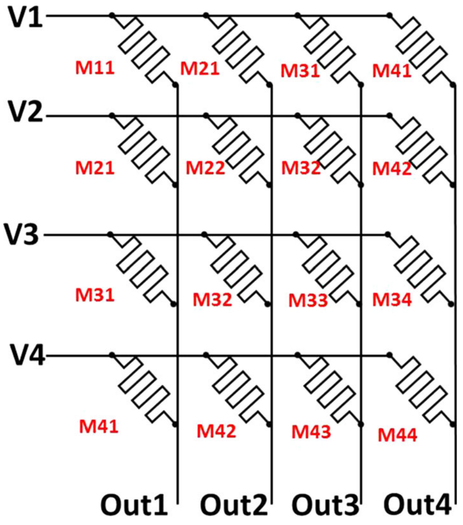

Fig. 4 A memristor crossbar array with 4 inputs and 4 outputs

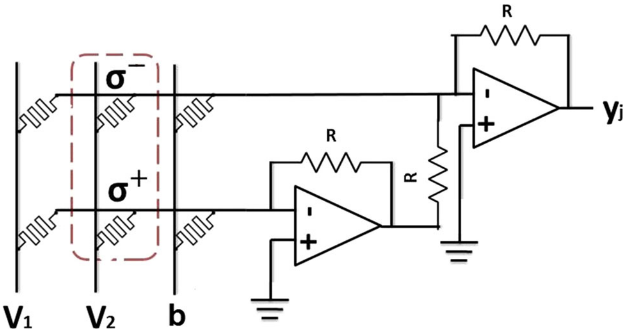

Fig. 5 Memristor neuron circuit diagram

$$(V_1 * \sigma^{(1)+} + V_2 * \sigma^{(2)+}) - (V_1 * \sigma^{(1)-} + V_2 * \sigma^{(2)-}) = \\ = (V_1(\sigma^{(1)+} + \sigma^{(1)-}) + V_2(\sigma^{(2)+} + \sigma^{(2)-})$$

The idea of this circuit can be further extended to implement vector-matrix multiplication of gating layers in LSTM. The sum of two memristances gives a required resulting weight, which can take both positive and negative values.

2.3 Activation layer circuit

Activation function in a traditional LSTM cell squashes each element of the output of vector-matrix multiplier either between 0 and 1 or − 1 and 1. To perform sigmoid and hyperbolic tangent function the CMOS-memristive thresholding circuit has been used. It is depicted in the Fig. 6. Memristor-inverter combination sets the threshold level whereas the breakworn voltage of a Zener diode determines the maximum height of the output voltage.

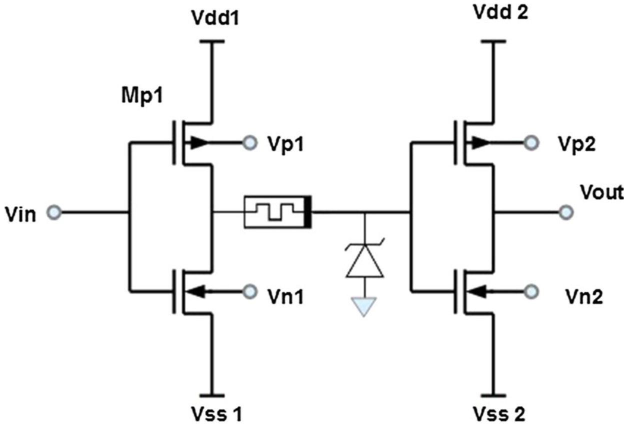

Fig. 6 Activation circuit

2.4 Voltage multiplier circuit

Implementation of a pointwise multiplication (also known as Hadamard or Schur product) is presented in the Fig. 8. It consists of one NMOS transistor T1, two differential amplifiers, two inverters, buffer and IV converter circuits.

3 Results

The responses of a circuit in the Fig. 6 for sigmoid and hyperbolic tangent functions are provided in the Fig. 7. The utilized CMOS technology in the circuit is 0.18um. To obtain hyperbolic tangent values the corresponding voltages should be set $V_{dd1} = 1.3\,V; V_{ss1} = -0.5\,V; V_{p1} = 0.4\,V; V_{n1} = -1.2\,V; V_{dd2} = 1.3\,V; V_{ss2} = -1.1\,V; V_{p2} = 0.5\,V$

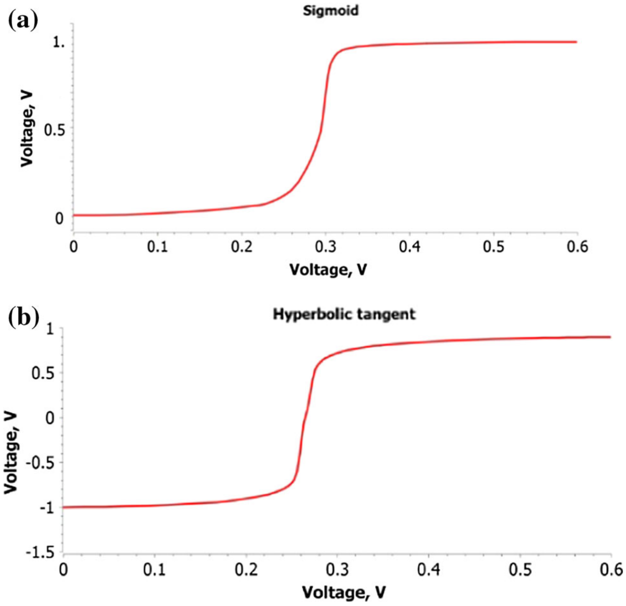

Fig. 7 Activation circuit response: **a** sigmoid; **b** hyperbolic tangent

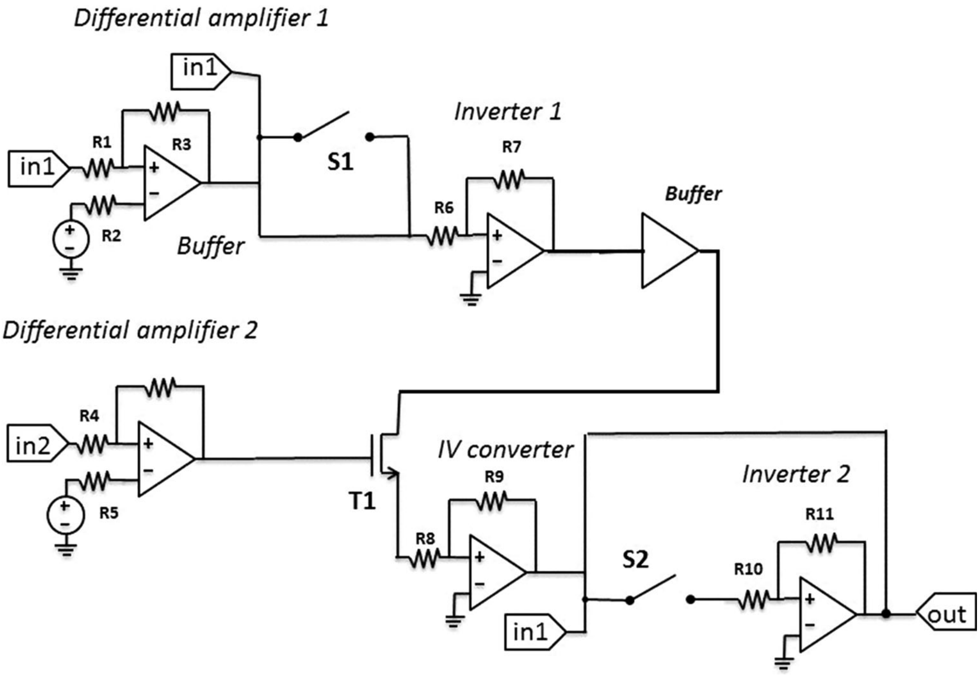

Fig. 8 Two voltages multiplication circuit

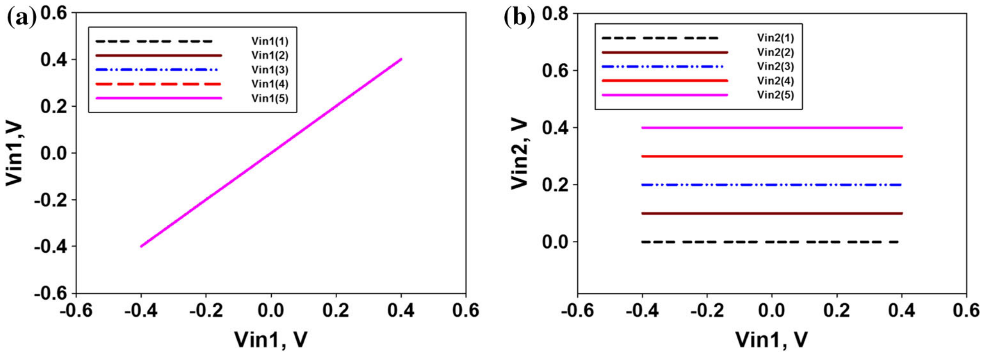

Fig. 9 Voltage multiplier inputs, **a** multiplier input 1, **b** multiplier input 2

and $V_{n2} = -0.4\,V$. The MOSFET transistor sizes are: $M_{p1} = M_{p2} = 0.18\,\mu\text{m}/3\,\mu\text{m}$, $M_{n1} = 0.18\,\mu\text{m}/4\,\mu\text{m}$, $M_{n2} = 0.18\,\mu\text{m}/4.5\,\mu\text{m}$.

Voltage multiplication (Fig. 8) is performed by a transistor T1(CMOS 0.18um technology, $T_1 = 2\,\mu\text{m}/2\,\mu\text{m}$) . Voltages to be multiplied V_{in1} and V_{in2} (see Fig. 9) are applied to the gate and drain of the transistor T1. The resulting output current is converted back to voltage by IV converter. Obtained voltage has an opposite sign therefore it is inverted again by Inverter 2. Since voltage applied to drain of transistor T1 should take only negative value and must not exceed the range (– 0.45:0)V, a differential amplifier 1 is used between input 1 and T1. Similarly, as transistor T1 gate voltage should always take positive values and lay in the range (0:0.45)V, a differential amplifier 2 is used between input 2 and transistor T1.

In our LSTM unit, Hadamard product is used to multiply LSTM gate outputs, which take values between (0:1) if activation function is a sigmoid and between (– 1:1) if activation function is hyperbolic tangent. Considering requirements for input values of transistor T1, V_{in1} is set to be an output hyperbolic tangent function and V_{in2} is an output of sigmoid function. By this, sigmoid fulfill the requirement of positive input voltage entering gate of T1 as its values are between 0 V and 1 V. As hyperbolic tangent function values are between (– 1:1)V and the drain input value can be only negative voltage, switches 1 and switch 2 are utilized to control multiplier output voltage polarity. If hyperbolic tangent output value is in the range (0:1)V, switch 1 passes a signal through Inverter 1 and buffer. Then inverted negative voltage enter the drain of T1 and multiplied with V_{in1}. Switch 2 is open and the resulting V_{out} is taken from IV converter output. Figures 9 and 10 illustrate operation of the voltage multiplier circuit. Total area of one voltage multiplier circuit is 2,871.00 μm^2 and power consumption 8.517 mW (Fig. 11).

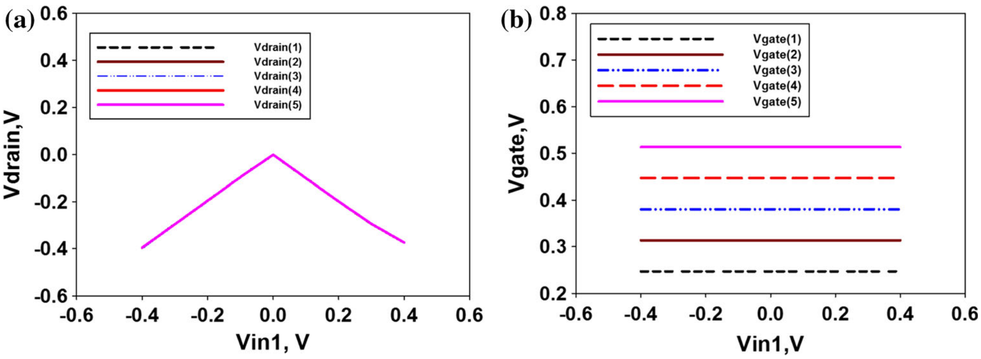

Fig. 10 Transistor T1 voltages, **a** transistor T1 drain voltage, **b** transistor T1 gate voltage

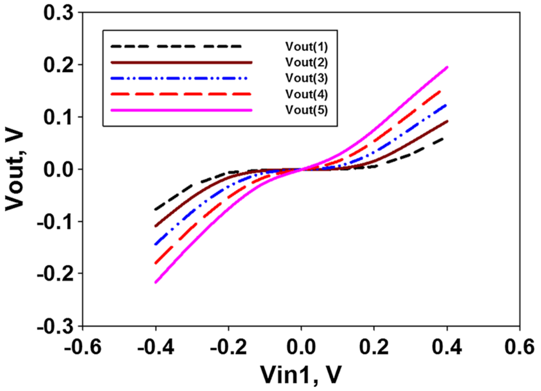

Fig. 11 Voltage multiplier output

4 Conclusion

This work proposes a hardware architecture design for implementation of LSTM algorithm for processing and storing sequential data. The architecture was designed based on 0.18 µm CMOS technology and novel devices called memristors. Utilization of memristor crossbar array for realization of vector-matrix multiplication within gate layers allows high scalability along with compatibility with CMOS technologies due to its nanoscale size and absence of leakage. The simulation results of the circuits for realization of basic computational operations of the traditional LSTM showed that they can be used to design other types of LSTM configuration.

References

1. Benediktsson, J. A., Swain, P. H., & Ersoy, O. K. (1990). Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, *28*(4), 540–552.
2. Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, *3*(1), 109–118.
3. Hopfield, D. A. (1982). Diagonal recurrent neural networks for dynamics control. *Proceedings of the National Academy of Sciences of the United States of America*, *79*, 2554–2561.
4. Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. *Advances in Psychology*, *121*, 471–495.
5. Lipton, Z. C., Berkowitz, J., Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. //arXiv preprint arXiv:1506.00019.
6. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.
7. Chua, L. (1971). Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, *18*(5), 507–519.
8. Strukov, D. B., et al. (2008). The missing memristor found. *Nature*, *453*(7191), 80–83.
9. Williams, R. S. (2008). How we found the missing memristor. *IEEE Spectrum*, *45*, 12.
10. Kim, K. H., et al. (2011). A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. *Nano Letters*, *12*(1), 389–395.
11. Hasan, R., Taha, T. M., & Yakopcic, C. (2017). On-chip training of memristor crossbar based multi-layer neural networks. *Microelectronics Journal*, *66*, 31–40.

Kamilya Smagulova has a M.Sc. in Nanotechnology from University College London. Her research interests are on neuro-memristive and learning systems. Currently, she is working towards her Ph.D. at Nazarbayev University.

Olga Krestinskaya is working towards her graduate degree thesis in the area of neuromorphic memristive system from Electrical Engineering department at Nazarbayev University. She completed her bachelor of Engineering degree with honors in Electrical Engineering, with a focus on bio-inspired memory arrays in May 2016. Currently, she focuses on hierarchical temporal memory and pattern recognition algorithms. He is a Graduate Student Member of IEEE.

Alex Pappachen James received the Ph.D. degree from the Queensland Micro and Nanotechnology Centre, Griffith University, Brisbane, QLD, Australia. He works on brain-inspired circuits as well as algorithms and systems. Currently, he is chairing the Electrical Engineering Department and leads the Circuits and Systems Group at Nazarbayev University. He is actively engaged in research commercialization and startups. He has several years experience of managing industry projects and academic projects in board design and pattern recognition circuits, and data and business analytics consulting for IT and semiconductor industry. He has been the founding chair for IEEE Kerala Section Circuits and Systems Society and Executive Member of IET Vision and Imaging Network. He is the founding chair of IEEE Kazakhstan subsection, and mentor to IEEE NU Student Branch. He was an editorial member of Information Fusion, Elsevier, and is an Associate Editor for HCIS, Springer, IEEE Transactions on Emerging Topics in Computational Intelligence, and IEEE Transactions on Circuits and Systems 1. He is a Senior Member of IEEE, Life Member of ACM, and Senior Fellow of HEA.