



**HAL**  
open science

# Long Short-Term Memory Neural Equalizer

Zihao Wang, Zhifei Xu, Jiayi He, Hervé Delingette, Jun Fan

► **To cite this version:**

Zihao Wang, Zhifei Xu, Jiayi He, Hervé Delingette, Jun Fan. Long Short-Term Memory Neural Equalizer. 2020. hal-03022865

**HAL Id: hal-03022865**

**<https://inria.hal.science/hal-03022865>**

Preprint submitted on 25 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Long Short-Term Memory Neural Equalizer

Zihao Wang, Zhifei Xu, *Member, IEEE*, Jiayi He, Hervé Delingette, Jun Fan, *Fellow, IEEE*

**Abstract**—A trainable neural equalizer based on Long Short-Term Memory (LSTM) neural network architecture is proposed in this paper to recover the channel output signal. The current widely used solution for the transmission line signal recovering is generally realized through DFE or FFE-DFE combination. The novel learning-based equalizer is suitable for highly non-linear signal restoration thanks to its recurrent design. The effectiveness of the LSTM equalizer is shown through an ADS simulation channel signal equalization task including a quantitative and qualitative comparison with an FFE-DFE combination. The LSTM neural network shows good equalization results compared to that of the FFE-DFE combination. The advantage of a trainable LSTM equalizer lies in its ability to learn its parameters in a flexible manner, to tackle complex scenario without any hardware modification. This can reduce the equalizer implantation cost for variant transmission channels and bring additional portability in practical applications.

**Index Terms**—decision feedback equalizer, neuromorphic computing, deep learning, LSTM

## I. INTRODUCTION

NOWADAYS, the breakthrough of massive data processing and data centers increases the demand for highly efficient data transmission at macro-level (telecommunication), meso-level (onboard systems), and micro-level (inside chips). With the dramatic increase of transmission speed, it is challenging to maintain the signal integrity (SI), power integrity (PI) in high-speed links design. The impedance discontinuities, field interaction, simultaneous switching noise (SSN), power delivery networks (PDN), etc, will distort the high-speed signals [1], [2], [3], will increase the intersymbol interference (ISI) to a point where the output of a high-speed channel may hardly be recognized by the receiver. This is why equalization techniques have been developed to reconstruct the signal at the output of the high-speed channel.

A typical Serializer-Deserializer (SerDes) system targeted for high-speed data transmission includes three modules (see Fig. 1): transmitter TX, receiver RX, and phase-locked loop (PLL) block [4]. The signal is transmitted from TX through the channel and received by the receiver. The PLL performs clock signal synchronization which is a naive idea for identifying the delay of the transmission channel that properly match with the

delayed signal. In the RX block, the equalization techniques are implemented to reconstruct the deformed signal due to the channel loss and other complex environmental influence. There are three types of equalizers that are generally employed in the RX module. The feed-forward equalizer (FFE), the decision feedback equalizer (DFE) [5], [6], [7], and the continuous-time linear equalizer (CTLE) [8]. The FFE is dedicated to compensating the loss of the channel which is assumed to be an FIR filter with a transfer function inverse to the channel transfer function. Since the compensation of that loss only is not sufficient due to the noise from the complex environment, the FFE and DFE are jointly used to have a better performance on the signal reconstruction. Since all bit signals are pulses, when the bit signal goes through the channel, there will be delays and tails. The DFE is applied to drop the tail down to 0, then the pulse tail will not affect the next bit signal, thus, the better signal reconstruction can be reached. However, due to the design of the channel and inherent non-linear representation limitations, the FFE and DFE combinations are unable to equalize original signals especially with the complex channel in very high-speed data transmission.

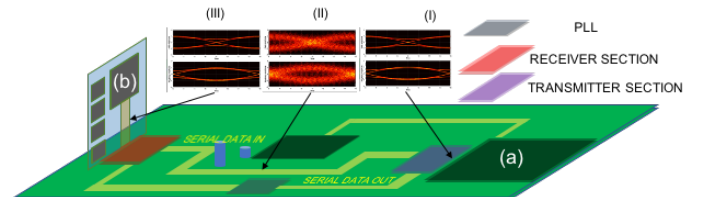


Fig. 1: Backboard SerDes architecture. Signal emission from chips is transferred through a transmission line which leads to loss and distortion.

Model based equalizers usually requires a good understanding of the transmission systems attributions and a rich expert knowledge in design but often lack of generalization ability [9]. Facing this challenge with the emergence of artificial intelligence technology [10], several learning-based equalizers have been proposed that are more adaptable over varied channel conditions. Most of the learning-based equalizers are nonlinear equalizers. Some representative works [11], [12], [13] were using a Bayesian statistical model for channel equalization. Neural network-based approaches are used by some prior works [14], [15], [16], [9] for channels equalization. Some uncommonly used low-speed equalization technologies such as Support Vector Machine based equalizer [17], [18], Fuzzy based networks equalizer [19] were more difficult for hardware implementation and are not elaborated in detail here.

Recent work of Chu *et al.* [20] introduced a neural network circuitry equalizer (NNE) for high-speed signal link construction. An artificial neural network (ANN) is integrated into

Manuscript received xxx. This work has been partially supported by the French government, through the 3IA Côte d’Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002. And in part by the US National Science Foundation under Grant No. IIP-1916535.

Zihao Wang and Hervé Delingette are with the French National Institute for Research in Computer Science and Control (INRIA) and Université Côte d’Azur (UCA), and Hervé Delingette is with French Interdisciplinary Institutes of Artificial Intelligence (3IA Côte d’Azur). 06902 Valbonne, France (e-mail: zihao.wang, herve.delingette@inria.fr).

Zhifei Xu, Jiayi He and Jun Fan are with the Electromagnetic Compatibility Laboratory, Missouri University of Science and Technology, Rolla, MO 65401, USA (e-mail: zxfdc, hejiay, jfan@mst.edu).

the continuous-time linear EQ (CTLE) as the replacement of the DFE block (block (b) in Fig. 1). Their quantitative experiments have shown that the ANN-based equalizer is more robust than the DFE approach for degraded signal restoring accuracy. In terms of the transmission speed improvement, the NNE improved about 30% in data rate vis-à-vis the traditional equalization method. A similar work of Rajbhandari *et al.* employed ANN for multiple-input-multiple-output visible light communication (MIMO-VLC) system signal equalization. The MIMO-VLC equalization problem is usually more challenging than the PCB level signal equalization as the channel of the MIMO-VLC system can be an open environment which may be easily violent by the external optical noisy. The experimental study has shown that the ANN equalizer outperformed the temporal-spatial cross-talks compensation [15] and possible for the hardware implementation.

The conventional ANN-based equalizer is not capable of capturing the time series information as the input signals are processed without considering the temporal relationship [21]. However, the signal equalization task is typically a temporal processing problem that is suitable for employing our LSTM based equalizer. Several prior works have shown that the high potential of neural network-based equalization for the future high-speed, complexity electromagnetic channels and long-distance transmission systems design. We see that most of the neuromorphic computing systems used for equalization are still limited to the basic perceptions or multi-layer perceptron neural systems.

In this paper, we propose a novel approach based on the long short-term memory neural network and deep learning for channel signal equalization. In many circumstances, neural network computing is software implemented and run on GPGPU (General-purpose processing on graphics processing units) or TPU (Tensor Processing Unit) acceleration which limit its application for highly integrated circuit systems. Yet, the proposed approach can be implemented through programmable gate array or ASIC or general neuromorphic computing instruments (related literature: [22], [23], [24]) which realize the embedding for the integrated system. To the best of our knowledge, our proposed method is the first application of a physical realizable temporal memory-based neural equalizer. Furthermore, the proposed approach is adaptable to variable frequency signal equalization thanks to the ability of neural networks for performing powerful approximations.

The paper is organized as follows: In section II we first describe the memory gates mechanism of the LSTM network and the construction procedures for LSTM based equalizer architecture design. Section III describes the implementation of the proposed method both simulated and physical realization. The conclusion section V summarizes the full paper and look into the future of applying deep learning approaches for equalizer design.

## II. LONG SHORT-TERM MEMORY NEURAL EQUALIZER

### A. Long Short-Term memory neural network

The long Short-Term memory neural network (LSTM) [25] is a type of recurrent neural networks (RNN) which is specially

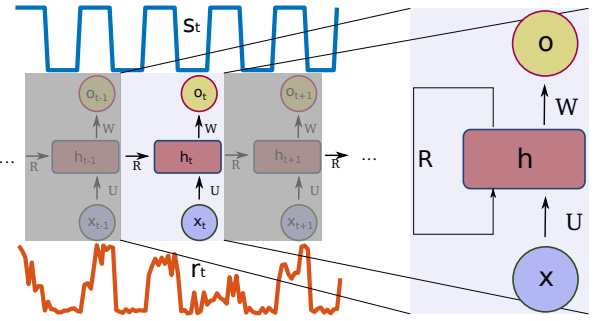


Fig. 2: Functional description of a recurrent neural network for time series processing. Input signal is red series  $r_t$  and output signal is shown with blue line  $s_t$

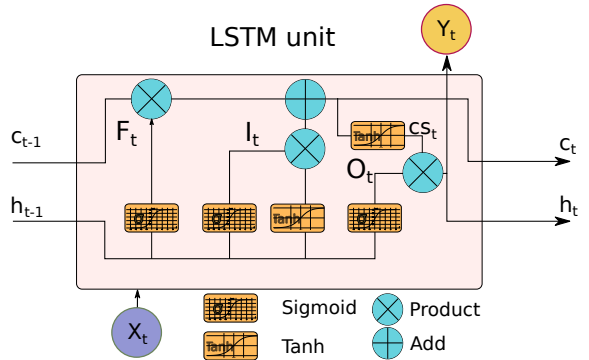


Fig. 3: Architecture of a LSTM cell. The cell's long memory is  $c_{t-1}$  and hidden state is  $h_{t-1}$ . At time  $t$ , the input signal of current state  $x_t$  is processed by the LSTM cell to generate the current hidden and cell state.

designed for processing time series or sequential data. Differing from fully connected neural networks, RNNs can learn temporal information by performing recurrent backpropagation [26]. Fig. 2 shows a RNN for signal series regularization. At time  $t$ , the RNN is fed with a noisy signal  $x_t = r_t$  and the memory information  $R_{t-1}$  passed from the last time  $t-1$ . The neural network  $h_t$  will then generate as output the processed signal  $O_t = s_t$  at time  $t$  and memory information  $R_t$ . Yet, RNN is limited by the transmission of past information since the memory  $R_{t-1}$  might be not sufficient for processing the input of current time  $t$ . The LSTM design tackles this problem by introducing the long short-term memory gates mechanism [25]. It has been shown in practice that the LSTM is better than RNN for long-distance memory processing, and LSTM has been successfully applied to many areas related to time series signals such as Google neural machine translation system, Microsoft voice recognition system, ECG beats signal classification, and Apple's Siri assistant systems, etc [27], [28], [29].

As shown in Fig. 3, the LSTM cell consists of a group of neurons and several information transformation flows. The cell state variable  $c_{t-1}$  carries the memory information of the previous cell whereas the hidden variable  $h_{t-1}$  is the cell output information of the previous cell. Within the LSTM cell, two activation functions are used to perform non-linear transformations of the information flows:

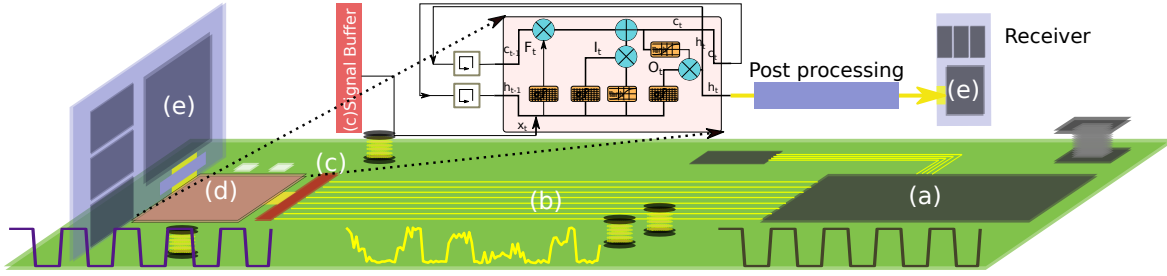


Fig. 4: The signal link of the proposed LSTM neural network based channel equalizer. (a) Chip generates logic bits. (b) Arbitrary signal channel. (c) N bits signal delay. (d) LSTM neural network signal interpreter. (e) Signal receiver.

- The Sigmoid function:  $\sigma(x) = \frac{1}{1+\exp(-x)} \in ]0, 1[$  computes  $\sigma(h_{t-1})$  which controls the input gate  $I_t$ , and the forget gate  $F_t$  by a Hadamard product (or element-wise multiplication noted  $\odot$ ) with the last cell state  $c_{t-1}$ . The activation level of the forget gate is correlated with the information passing from the last cell state.
- The hyperbolic tangent function :  $\tanh(x) = \frac{\exp(2x)-1}{\exp(2x)+1} \in ]-1, 1[$  is used for memory selection by controlling the activation level of the current cell input state  $x_t$ .

The computation of the intermediate variables shown in Fig: 3 are given by:

$$\begin{aligned} f_t &= \sigma(w_f x_t + w_{rf} h_{t-1} + b_f) \\ i_t &= \sigma(w_i x_t + w_{ri} h_{t-1} + b_i) \\ o_t &= \sigma(w_o x_t + w_{ro} h_{t-1} + b_o) \\ cs_t &= \tanh(w_c x_t + w_{rc} h_{t-1} + b_c) \end{aligned} \quad (1)$$

where  $f$ ,  $i$ ,  $o$  are the forget, input, and output gates while  $cs$  is the cell input activation vector. The update rules for the memory ( $c_t$ ) and output ( $h_t$ ) are given by :

$$\begin{aligned} c_t &= f_t \odot c_{t-1} + i_t \odot cs_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \quad (2)$$

Thus, the current cell state is a combination of the previous cell state and the cell input activation controlled by the forget and input gate variables. The hidden state is a function of the cell state weighted by the output gate variable.

In addition, the trainable parameters for the LSTM networks are:

- The neural networks weight parameters:

$$W = \{\{w_f^n, w_i^n, w_c^n, w_o^n\}, n \in \mathcal{N}\} \quad (3)$$

- The bias parameters :

$$b = \{\{b_f^n, b_i^n, b_c^n, b_o^n\}, n \in \mathcal{N}\} \quad (4)$$

- The recurrent weights:

$$W_r = \{\{w_{rf}^n, w_{ri}^n, w_{rc}^n, w_{ro}^n\}, n \in \mathcal{N}\} \quad (5)$$

Furthermore, an LSTM cell can be easily implemented in a digital circuit with adjustable neural activation parameters and the LSTM circuit is amenable to analog computing in which the neurons are directly modeled by electronic signals [30], [31], [32]. The analog neural network implementation can overcome the frequency limitation of equalizers where the

main issue is the frequency of the converter of the digital and analog signal (DAC).

### B. Long short-term memory neural equalizer

---

#### Algorithm 1: Signal forward propagation at time $t$ .

---

**Inputs:** N bit Signal delayed:  $r_{mem}^n$

**Output:** Equalized signal:  $s'$

Restore neural parameters from RAM:  $W, WR, b$ ;

// LSTM

Initialize states cache:  $h = \mathbf{0}, c = \mathbf{0}$ ; //

**while** Clock Refresh **do**

$t \leftarrow$  clock signal

$h_{t-1}, c_{t-1} \leftarrow h, c$ ; // Get last cell states

$r_t \leftarrow r_{mem}^n$ ; // Get current signals

// Compute the three cell gates and activation based on the input signal vector  $r_t$  and last states

$\{h_{t-1}, c_{t-1}\}$

$f_t \leftarrow \sigma(w_f r_t + w_{rf} h_{t-1} + b_f)$ ; // forget gate

$i_t \leftarrow \sigma(w_i r_t + w_{ri} h_{t-1} + b_i)$ ; // input gate

$cs_t \leftarrow \tanh(w_c r_t + w_{rc} h_{t-1} + b_c)$ ; // cell gate

$o_t \leftarrow \sigma(w_o r_t + w_{ro} h_{t-1} + b_o)$ ; // output gate

// Compute the current cell outputs:  $c_t, h_t$

$c_t \leftarrow f_t \otimes c_{t-1} + i_t \otimes cs_t$ ; // cell activation

$h_t \leftarrow \phi_t \otimes \tanh(c_t)$ ; // hidden gate

$h, c \leftarrow h_t, c_t$ ; // Refresh states cache

$s^* \leftarrow \mathcal{F}(h)$ ; // Fully connected layer

$\hat{s} \leftarrow FIR(s^*)$ ; // Post processing filtering

**end**

---

The long short-term memory neural equalizer is an intrinsic-feedback equalizer system where the spoiled signal series  $r$  shown in Alg: 1 from the transmission line are directly processed by the LSTM neural block while the feedback information is achieved by the long-short memories states  $h, c$ . As Fig. 4 shows, the LSTM is integrated as a terminal

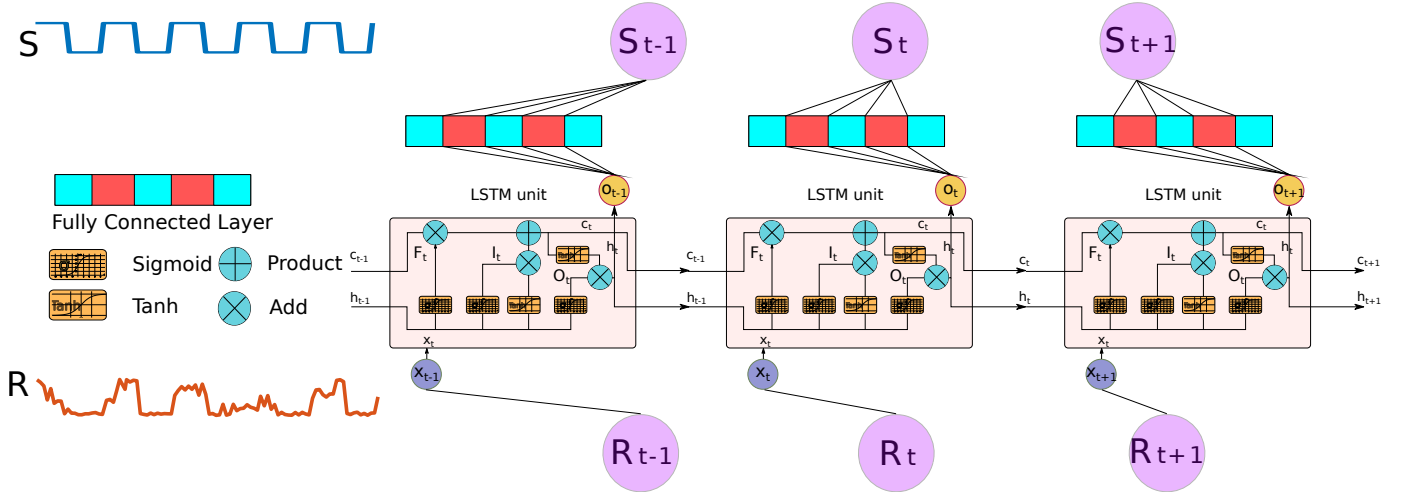


Fig. 5: LSTM Equalizer diagram shows input spoiled signals  $R_t$  and equalized signals  $S_t$ . The cell states and hidden states are feedback signals between different time-frame.

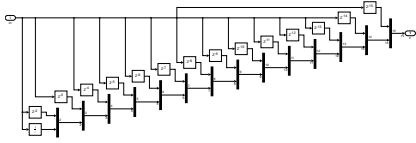


Fig. 6: A simulated shift register buffer with 15 delays[33]. The physical implementation can be realized via cascade flip-flops circuits.

block in the signal channel link, where the input sequential signals  $r$  come from the transmission channels (b) and are re-sampled by a sample-hold circuit and stored in a  $n$  bits memory buffer  $r_{mem}^n$  (see block (c) of Fig. 4). This buffer is a Serial-In Parallel-Out (SIPO) buffer where the first register  $r_{mem}^1$  stores the  $r_t$  signal and the last register stores the  $r_{t-n}$  signals. The signals stored in the buffer are the input to the LSTM interpreter (d) to generate the equalized signal. The equalized signals are post-processed by a signal processing block to get the equalized bits  $\hat{s}$ . A detailed presentation of the forward propagation of the signals at the time  $t$  is given in Alg. 1.

The output hidden state  $h_t$  at time  $t$  is decoded by a fully connected neuron  $\mathcal{F}(h_t)$  to get the equalized signal output. This fully connected cell is parameterized by  $w_{full}$  and  $b_{full}$ . The output of the full connected cell is given by:

$$\mathcal{F}(h_t) = \sigma(w_{full}h_t + b_{full}) \quad (6)$$

Finally, the signal is processed by a FIR filter to remove the noise and stabilize the waveform. The LSTM network aims to minimize the difference between the equalized bits  $\hat{s}$  and the bits signal  $s$  emitted from the chip (a). We can train the LSTM by minimizing the Mean-Square-Error (MSE) of the two items:

$$MSE = \sum_{i=1}^N \frac{(s_i - \hat{s}_i)^2}{N} \quad (7)$$

The LSTM network parameters can be trained through the backpropagation algorithm [34] to estimate the gradient of the

MSE functional on a batch of data.

### C. LSTM training

a) *Dataset*: Both a training and a validation dataset need to be collected for training the LSTM equalizer. The neural network is trained through a batch  $B_t$  of training data and corresponding target signal. The validation dataset is used to control the training process in order to avoid overfitting the training dataset. We write the training set as  $\mathcal{D}_{train}^K : \{S_{train}|R_{train}\}$  in Alg. 2, where

$$R_{train} = \begin{Bmatrix} \{R_{t-n} \dots R_t\}^1 \\ \{R_{t-n-1} \dots R_{t-1}\}^2 \\ \dots \\ \{R_0 \dots R_n\}^K \end{Bmatrix}; S_{train} = \begin{Bmatrix} \{S_t\}^1 \\ \{S_{t-1}\}^2 \\ \dots \\ \{S_n\}^K \end{Bmatrix} \quad (8)$$

$R_{t-n} \dots R_t$  in equation (8) represents a sequence of signals which terminates at time  $t$  (see Fig. 5), and the same notation is used for the validation dataset  $\mathcal{D}_{valid}^K : \{S_{valid}|R_{valid}\}$ . The training dataset can be either measured from physical transmission platforms or can be collected from a simulation of the transmission channels when the S-parameters of the transmission line is known. In the training phase, the  $\mathcal{D}_{train}^K$  is fed to the LSTM network by batch to minimize the MSE loss function by stochastic gradient descent based algorithms (*i.e.*: Adam in this work) [35].

b) *Backpropagation*: The parameters of the LSTM equalizer are determined through a batch-based backpropagation algorithm [34] which is performed in two steps: (I) forward propagation as described in Alg. 1. (II) backward propagation where the neural network parameters are updated based on the gradients' information of error feedback. The details of backpropagation are described in Alg. 2

## III. EXPERIMENTS AND RESULTS

### A. Experimental architecture

The proposed LSTM equalizer proposed in this paper was implemented and verified through Simulink modeling for a

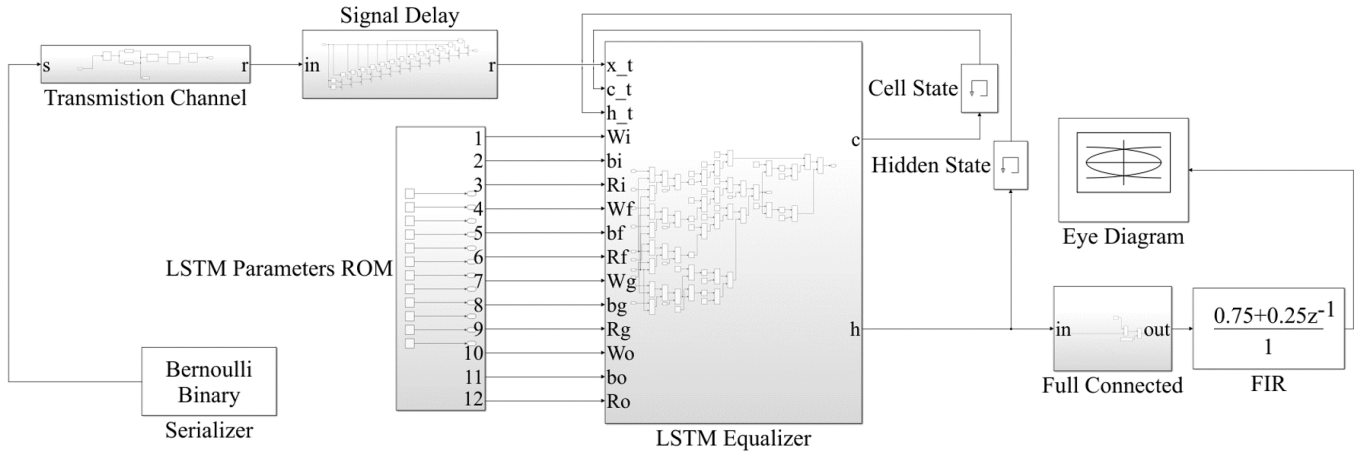


Fig. 7: The Simulink simulation architecture diagram of the proposed LSTME applied for a simulated channel signal equalization.

single frequency signal. As shown in Fig. 7, the original signals are generated by a Bernoulli binary generator. Signals are transmitted through a transmission channel and then delayed by the signal delay circuits as the input to the LSTM equalizer. The trained LSTM neural networks are stored in the LSTM parameters ROM and the LSTM equalizer will read the parameters  $w, b, R$  from LSTM Parameters ROM to determine the output signals: cell states  $c$  signal and hidden states  $h$  signal. Both signal flows are feedback to the LSTM as inputs for the next signal equalization cell. The cell state signal is further processed by the fully connected network block to generate the response signal of the LSTME. In the final stage, the FIR filter is applied for the final signal correction. Generally, the LSTM equalizer can be realized through the following steps:

- Get the characteristics of the channel
- Obtain the output data of the channel by simulation
- Train the dataset
- Implement the trained parameters to the neural network
- Realize the hardware implementation

Two experiments have been conducted including a comparison with optimized FFE-DFE. The former experiment is focused on speed for which a designed PCB channel characterized up to 50GHz is used. The latter explores the waveform of the channel output at 10 Gbps speed, including a simulation-based channel. In both cases, the waveform and the characteristics of eye diagrams are compared.

### B. Experimental result for a high-speed PCB channel

A PCB channel, containing transmission line, via, connector, etc, has been used for this experiment. The S parameter of the channel characterized up to 50 GHz is shown in Fig. 8 with both insertion and return losses. The insertion loss exposes very high loss in high frequencies. The transmitted signal is severely attenuated and therefore hard to recognize by the receiver. Therefore, a FFE-DFE or our proposed method is needed to recover the signal.

A simulation is performed in ADS with the S parameters of the fabricated PCB channel. The schematic is shown in Fig. 9.

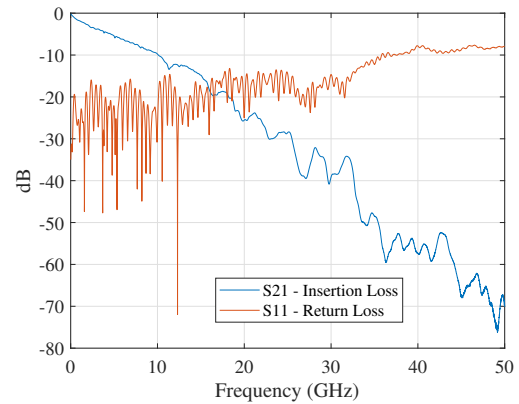


Fig. 8: Return loss and insertion loss of the simulated channel.

There are two channels within this S-parameter file, one of the channels is employed in this simulation for simplicity. A 50 Gbps data rate is chosen for the simulation. The high/low levels are defined as 1 V/0 V, respectively. The rise and fall times are equal to 4 psec.

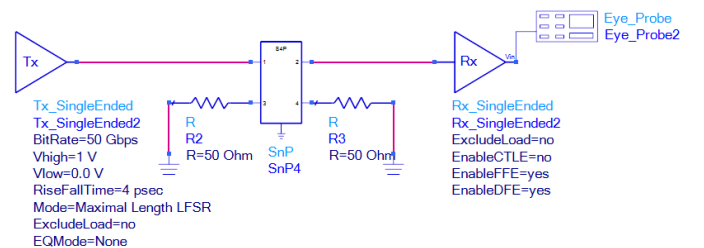


Fig. 9: Schematic of the ADS channel simulation.

1) *Eyediagram comparison:* To assess the performance of the proposed approach, we use eye diagram analysis which provides intuitive evaluation for the performance of digital signal equalization. As shown in Fig. 10, the eye is not open at all in the eyediagram of the output data which means the output signals will have high bit-error rate (BER) without the FFE-DFE equalization. The FFE and DFE cursors

**Algorithm 2:** LSTM training based on backpropagation.

```

Inputs: Training data :  $\mathcal{D}_{train}^K : \{S_{train}|R_{train}\}$ 
Inputs: Validation data :  $\mathcal{D}_{valid}^K : \{S_{valid}|R_{valid}\}$ 
Inputs: Learning rate: $\alpha$ ; Validation Steps:  $V$ 
Output: LSTM Parameters:  $W, WR, b$ 
// Initialize with Xavier method[36]
Xavier initializer:  $W, WR, b$ ; //
Initialize states:  $h = \mathbf{0}, c = \mathbf{0}$ ; //
while  $\epsilon_{valid} < \delta_{converge}$  do
  for  $i = 1; i < K; i ++$  do
     $R^i \leftarrow \{R_{t-n} \dots R_t\}^i$ ; // Get ith input
     $S^i \leftarrow \{S_t\}^i$ ; // Get ith training
    target
    // Forward propagation with
    Algo: 1
     $R' \leftarrow LSTM(R^i)$ ; //
    // Get the loss between output
    and target based on loss
    function:7
     $\epsilon = loss(R', S^i)$ ; //
     $\Delta W \leftarrow \alpha \times \frac{\partial \epsilon}{\partial W}$ ; // Jacobian W
     $\Delta WR \leftarrow \alpha \times \frac{\partial \epsilon}{\partial WR}$ ; // Jacobian WR
     $\Delta b \leftarrow \alpha \times \frac{\partial \epsilon}{\partial b}$ ; // Jacobian b
     $W \leftarrow \Delta W + W$ ; // Update W
     $WR \leftarrow \Delta WR + WR$ ; // Update WR
     $b \leftarrow \Delta b + b$ ; // Update b
    if  $0 \equiv i \pmod V$  then
       $\epsilon_{valid} \leftarrow 0$ ; //
      for  $j = 1; j < K'; j ++$  do
        // Get validation dataset
         $R^j \leftarrow \mathcal{D}_{valid}^{K'} : \{R_{t-n} \dots R_t\}^j$ ; //
         $S^j \leftarrow \mathcal{D}_{valid}^{K'} : \{S_t\}^j$ ; //
        // Forward propagation with
        Algo: 1
         $R' \leftarrow LSTM(R^j)$ ; //
         $\epsilon_{valid} \leftarrow \epsilon_{valid} + loss(R', S^j)$ ; //
      end
       $\epsilon_{valid} = \frac{\epsilon_{valid}}{K'}$ ; //
      break; //
    else
      continue;
    end
  end
end
    
```

values applied in this case are shown in Table I. The active FFE is applied to recover the loss due the channel physical parameters.

Yet, with the FFE-DFE equalizer processing, the eye is opened slightly as shown in Fig. 11 with eye height is 0.18 V, eye width is 5.1 ps and jitter is 13.8 ps.

Fig. 12 shows the eye diagram when using the LSTM equalizer. The eye characteristics comparison between the active FFE-DFE and LSTM is shown in Table. II. We see that the eye width, height and jitter are much improved compared

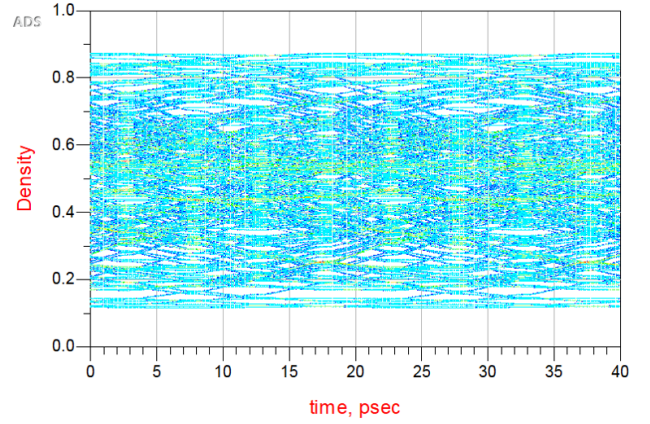


Fig. 10: Eye diagram of ADS channel signal simulation (raw signal from channels).

TABLE I: DFE and FFE parameters

FFE		DFE
PreCursor	PostCursor	Taps
-2.337340	4.038660	0.322812
	-2.185680	-0.017401
0.782150	0.534350	0.048581
		-0.065590
		0.039204
		-0.021085

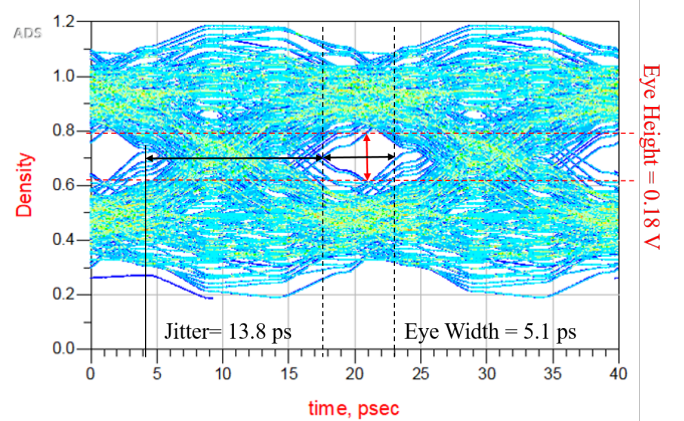


Fig. 11: Eye diagram of ADS channel signal equalized via the conventional active FFE-DFE system.

to the active FFE-DFE approach.

2) *Waveform comparison:* The waveform comparison is shown in Fig. 13. The red curve is the output signal without any equalization whereas the blue signal in the top (resp. bottom) figure is the output signal after the LSTM (resp. active FFE/DFE) equalization. Even with active FFE-DEF equalization, the loss and shape of the waveform are significantly degraded compare to the original signal. However, the case

TABLE II: Eyediagram characteristic comparison

	Jitter (ps)	Eye width (ps)	Eye height (V)
Active FFE-DFE	13.8	5.1	0.18
LSTM	9.8	12.8	0.87

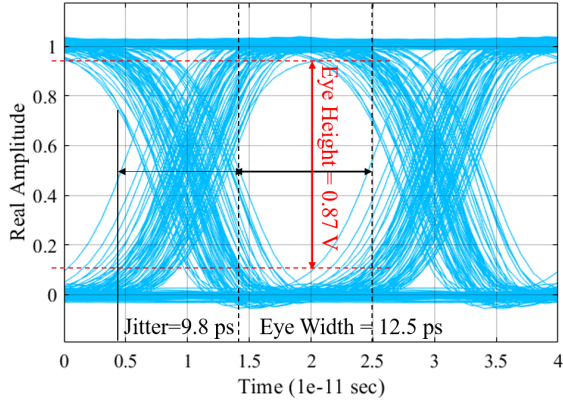


Fig. 12: Eye diagrams of ADS channel signal (50 GHz) equalized via the proposed LSTM equalizer system.

with LSTM equalizer shown in the top figure of Fig. 13 present great signal recovery ability. This clearly shows that our proposed method has a very good ability to recover signals compare to the traditional active FFE-DFE equalizers.

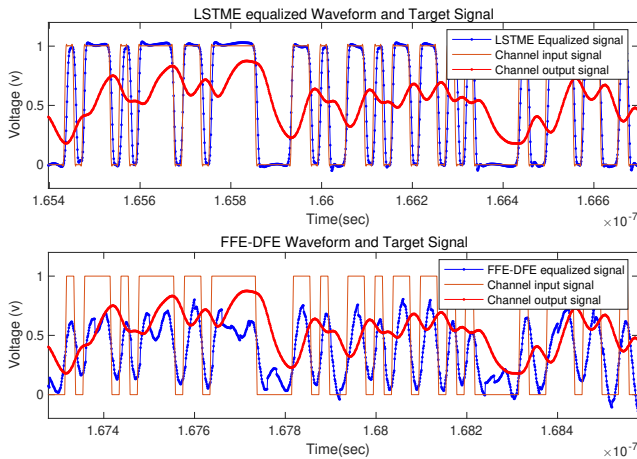


Fig. 13: Comparison between LSTME (top) and FFE-DFE (bottom) equalized signal waveform with input and output signal of the transmission channel shown.

### C. Experimental results for a mismatched channel

In the world of signal integrity, due to the limitation of PCB manufacturing, impedance mismatching is a general problem for high-speed transmission system designs. The overshoot and undershoot problems will appear due to the impedance mismatching. And the defect on the channel will also distort the output signals. Therefore, in this experiment, our proposed equalizer is used for a badly designed channel with mismatched loads to exhibit the strength of our method. The channel characteristic is shown in Fig. 14, the channel exhibit the high reflection which is mainly due to the impedance mismatching. The eye diagram for this badly designed channel is first obtained through ADS simulation with optimized FFE and DFE settings. A 20 Gbps signal with 10 ps rise/fall time is applied to this badly designed channel.

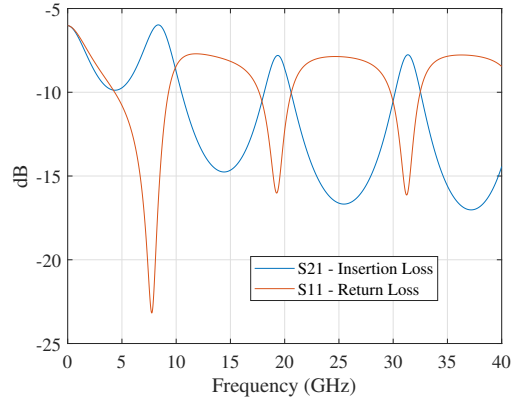


Fig. 14: Return loss and insertion loss of the bad designed channel

1) *Eyediagram comparison:* Fig. 15 shows the output eyediagram of the bad design channel with FFE-DFE equalization from ADS simulation. However, the overshoot and undershoot are having a high level which also induce a large jitter. Compare to the eyediagram generated with LSTME, the improvements from LSTME are obvious in jitter, eye height and width and the overshoot, undershoot.

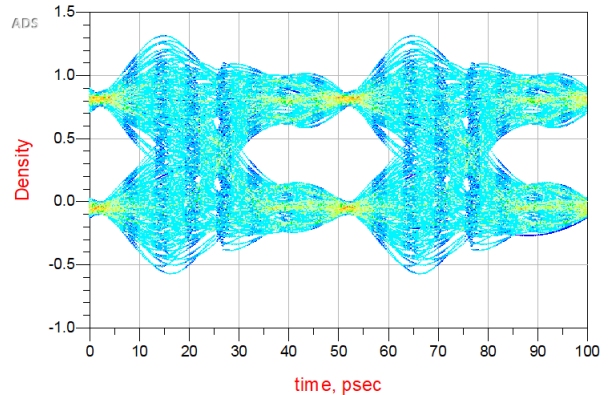


Fig. 15: ADS simulated eye diagram with bad designed channel

2) *Waveform comparison:* The waveforms from active FFE-DFE equalized results and LSTME results are shown in Fig. 17. The blue curve shown in the lower figure of Fig. 17 indicates the output waveform of the active FFE-DFE. The waveform is significantly degraded with overshoot and undershoot. However, for the output waveform of the LSTME shown in the top figure of Fig. 17 exposes perfect match to the original signal (red curve). The efficiency of our proposed method is proved again.

## IV. DISCUSSION

The output waveforms with different equalization techniques are shown in Fig. 13 and Fig. 17. For FFE-DFE based equalization approaches, the restored transmission signals are spoiled and very different from the original signal. The output



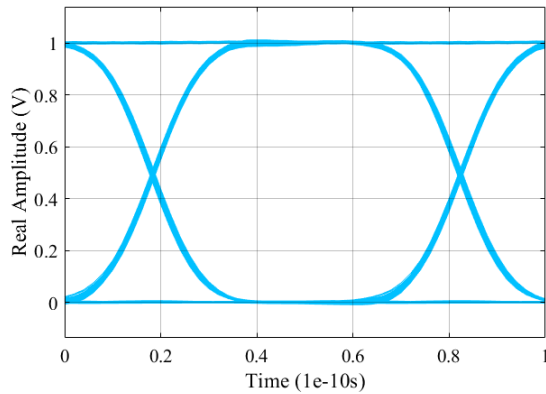


Fig. 16: Eye diagrams of badly designed channel signal (20 GHz) equalized via the proposed LSTM equalizer system.

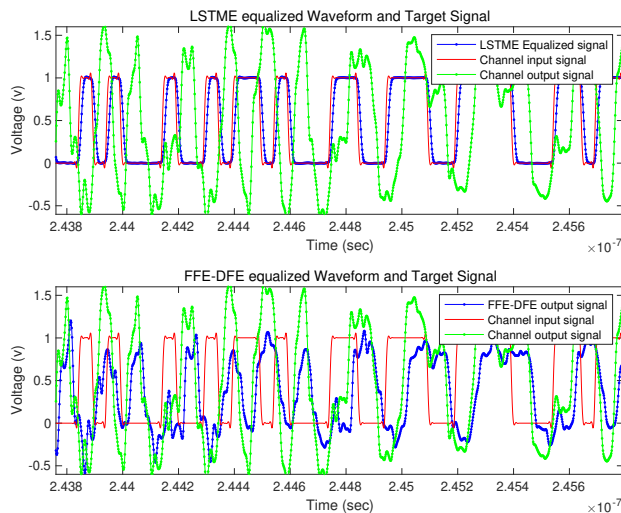


Fig. 17: Comparison of the waveform quality of the LSTM equalizer (Top) and FFE-DFE (Bottom) for a badly designed channel of 20 GHz signal.

signal from the LSTM equalizer greatly recovers the degraded signal. The proposed method also shows good performances on the eye diagram of Fig. 11 and Fig. 12 for the case with 50 GHz signal channel and Fig. 15, and Fig. 16 for 20 GHz signal channel.

A limitation of LSTM equalizer lies in the relative increase of computational complexity compared to the traditional FFE-DFE combination. This increase is related to highly non-linear representation capability of deep neural networks. Empirically, we noted that in section III for the equalization of a fabricated PCB channel with a fixed frequency 50 Gbps input signal, an LSTM equalizer with 20 hidden cells and 15 delays with the resolution of 5 ps is sufficient to get a much improved equalized output. In this case, the computational complexity is acceptable for practical use. However, the complexity of a deep learning LSTM equalizer may increase significantly with the demand for variable-frequency equalization. Therefore, a more thorough study of the required neural model complexity

with respect to variable equalization frequency, and different transmission channels model is necessary.

The proposed LSTM equalizer is based on supervised learning with a consistent paired training dataset to learn the non-linear mapping. This requires to collect specific simulated data and measurements, although it is also necessary for FFE-DFE based channel equalization. With the recent advances of semi-supervised/unsupervised learning [37], [38], there is a possibility to reduce the efforts for collecting the training dataset by applying those self-adaptive learning approaches to signal equalizer design.

## V. CONCLUSION

In this paper, we have proposed a novel recurrent feedback equalizer based on the LSTM neural network. The LSTM network is trained offline with a collected signal sequence dataset. We have shown that our approach outperforms commonly used FFE-DFE signal equalization methods with meaningful metrics and results. Furthermore, the LSTM equalizer is amenable to hardware/software implementation with flexible portability for different channels signals equalization. Future research will address the issue of variable frequency signal equalization and weakly-supervised learning approaches for the next-generation AI-based equalizer design. Moreover, the compatibility of the neural learning-based approaches for high-frequency, low-cost, low-power, and low-latency signal equalization needs to be further studied.

## REFERENCES

- [1] J. Fan, X. Ye, J. Kim, B. Archambeault, and A. Orlandi, "Signal integrity design for high-speed digital circuits: Progress and directions," *IEEE Transactions on Electromagnetic Compatibility*, vol. 52, no. 2, pp. 392–400, 2010.
- [2] J. Kim and E. Li, "Special issue on pcb level signal integrity, power integrity, and emc," *IEEE Transactions on Electromagnetic Compatibility*, vol. 52, no. 2, pp. 246–247, 2010.
- [3] Z. Xu, Y. Liu, B. Ravelo, J. Gantet, N. Marier, and O. Maurice, "Direct time-domain tan model of 3d multilayer hybrid pcb: Experimental validation," *IEEE Access*, vol. 6, pp. 60645–60654, 2018.
- [4] Y. He, Z. Wang, H. Liu, F. Lv, S. Yuan, C. Zhang, X. Xie, and H. Jiang, "An 8.5–12.5ghz multi-pll clock architecture with lc pll and ring pll for multi-lane multi-protocol serdes," in *2017 International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, pp. 1–2, 2017.
- [5] H. Wang and J. Lee, "A 21-gb/s 87-mw transceiver with ffe/dfe/analog equalizer in 65-nm cmos technology," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 909–920, 2010.
- [6] C. Ahn, J. Hong, J. Shin, B. Kim, H. Park, and J. Sim, "An 18-gb/s nrz transceiver with a channel-included 2-ui impulse-response filtering ffe and 1-tap dfe compensating up to 32-db loss," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2020.
- [7] X. Zheng, H. Ding, F. Zhao, D. Wu, L. Zhou, J. Wu, F. Lv, J. Wang, and X. Liu, "A 50–112-gb/s pam-4 transmitter with a fractional-spaced ffe in 65-nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 7, pp. 1864–1876, 2020.
- [8] B. Li, B. Jiao, C. H. Chou, R. Mayder, and P. Franzon, "Ctle adaptation using deep learning in high-speed serdes link," in *2020 IEEE 70th Electronic Components and Technology Conference (ECTC)*, pp. 952–955, 2020.
- [9] Q. Zhou, F. Zhang, and C. Yang, "Adann: Adaptive neural network-based equalizer via online semi-supervised learning," *Journal of Lightwave Technology*, vol. 38, no. 16, pp. 4315–4324, 2020.
- [10] E. Tohidi, R. Amiri, M. Coutino, D. Gesbert, G. Leus, and A. Karbasi, "Submodularity in action: From machine learning to signal processing applications," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 120–133, 2020.

- [11] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive bayesian equalizer with decision feedback," *IEEE Transactions on Signal Processing*, vol. 41, no. 9, pp. 2918–2927, 1993.
- [12] S. Chern, K. K. Lee, Y. Bai, and R. H. Yang, "Semi-blind channel estimation scheme with bayesian dfe for prp-ofdm system," in *2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 602–607, 2015.
- [13] Z. Kiguradze, N. Dikhaminija, M. Tsiklauri, J. He, B. Mutnury, A. Chada, and J. Drewniak, "Bayesian optimization for high-speed channel equalization," in *2019 Electrical Design of Advanced Packaging and Systems (EDAPS)*, pp. 1–3, 2019.
- [14] H. Abdulkader, B. Benammar, C. Poulliat, M.-L. Boucheret, and N. Thomas, "Neural networks-based turbo equalization of a satellite communication channel," in *15th International Workshop on Signal Processing Advances in Wireless Communications - SPAWC 2014*, (Toronto, CA), pp. 494–498, IEEE, 2014. Thanks to IEEE editor. The definitive version is available at [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6941914&tag=1](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6941914&tag=1).
- [15] S. Rajbhandari, H. Chun, G. Faulkner, H. Haas, E. Xie, J. J. D. McKendry, J. Herrnsdorf, E. Gu, M. D. Dawson, and D. O'Brien, "Neural network-based joint spatial and temporal equalization for mimo-vc system," *IEEE Photonics Technology Letters*, vol. 31, no. 11, pp. 821–824, 2019.
- [16] Q. Zhang, Y. Jiang, H. Zhou, C. Deng, S. Duan, Z. Wang, Y. Li, Y. Song, C. Jian, J. Zhang, Y. Wu, T. Wang, and M. Wang, "137 gb/s pam-4 transmissions at 850 nm over 40 cm optical backplane with 25 g devices with improved neural network-based equalization," *Applied Sciences*, vol. 9, p. 5095, 11 2019.
- [17] B. R. Taub, "Svm based method for multi-equalizer optimization," in *2019 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)*, pp. 1–4, 2019.
- [18] M. Bi, J. Yu, X. Miao, T. Huang, L. Li, H. Chi, and W. Hu, "A powerful equalizer based on modified svm classifier without nonlinear kernel enabled 100-gb/s ng-epon system with 10-g class," *IEEE Access*, vol. 7, pp. 71185–71194, 2019.
- [19] K. S. Mayer, M. S. D. Oliveira, C. Müller, F. C. C. D. Castro, M. C. F. D. Castro, and J. A. García-Naya, "Blind fuzzy adaptation step control for a concurrent neural network equalizer," *Wirel. Commun. Mob. Comput.*, vol. 2019, Jan. 2019.
- [20] Y. Chu, F. Chen, J. Lang, and B. Lee, "Equalization with neural network circuitry for high-speed signal link," in *2019 IEEE International Symposium on Electromagnetic Compatibility, Signal Power Integrity (EMC+SIPI)*, pp. 625–628, 2019.
- [21] Jing Dai, Pinjia Zhang, J. Mazumdar, R. G. Harley, and G. K. Venayagamoorthy, "A comparison of mlp, rnn and esn in determining harmonic contributions from nonlinear loads," in *2008 34th Annual Conference of IEEE Industrial Electronics*, pp. 3025–3032, 2008.
- [22] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and E. Eleftheriou, "Low-power neuromorphic hardware for signal processing applications: A review of architectural and system-level design approaches," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 97–110, 2019.
- [23] Y. LeCun, "1.1 deep learning hardware: Past, present, and future," in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 12–19, 2019.
- [24] A. Balaji, A. Das, Y. Wu, K. Huynh, F. G. Dell'Anna, G. Indiveri, J. L. Krichmar, N. D. Dutt, S. Schaafsma, and F. Catthoor, "Mapping spiking neural networks to neuromorphic hardware," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 1, pp. 76–86, 2020.
- [25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, Nov. 1997.
- [26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [27] Y. Wu, M. Schuster, Z. Chen, Q. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, u. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 09 2016.
- [28] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "Lstm time and frequency recurrence for automatic speech recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop*, December 2015.
- [29] E. D. Übeyli, "Combining recurrent neural networks with eigenvector methods for classification of ecg beats," *Digital Signal Processing*, vol. 19, no. 2, pp. 320 – 329, 2009.
- [30] J. Han, H. Liu, M. Wang, Z. Li, and Y. Zhang, "Era-lstm: An efficient reram-based architecture for long short-term memory," *IEEE Transactions on Parallel & Distributed Systems*, vol. 31, pp. 1328–1342, jun 2020.
- [31] Z. Zhao, A. Srivastava, L. Peng, and Q. Chen, "Long short-term memory network design for analog computing," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 15, pp. 1–27, 01 2019.
- [32]
- [33] W. Hua, R. N. Tador, and P. Beerel, "2 ps resolution, fine-grained delay element in 28 nm fdsoi," *Electronics Letters*, vol. 51, no. 23, pp. 1848–1850, 2015.
- [34] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85 – 117, 2015.
- [35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.
- [36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," vol. 9 of *Proceedings of Machine Learning Research*, (Chia Laguna Resort, Sardinia, Italy), pp. 249–256, JMLR Workshop and Conference Proceedings, 13–15 May 2010.
- [37] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4: Self-supervised semi-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [38] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.