A Deep Learning Based Approach for Analog Hardware Implementation of Delayed Feedback Reservoir Computing System

Jialing Li, Kangjun Bai, Lingjia Liu, Yang Yi
Bradley Department of Electrical and Computing Engineering,
Virginia Tech, Blacksburg, Virginia, 24061, USA
E-mail: {ljl, kangjun, ljliu, yangyi8}@vt.edu

Abstract

As the 2020 roadblock approaches, the need of breakthrough in computing systems has directed researchers to novel computing paradigms. The recently emerged reservoir computing model, delayed feedback reservoir (DFR) computing, only utilizes one nonlinear neuron along with a delay loop. It not only offers the ease of hardware implementation but also enables the optimal performance contributed by the inherent delay and its rich intrinsic dynamics. The field of deep learning has attracted worldwide attention due to its hierarchical architecture that allows more efficient performance than a shallow structure. Along with our analog hardware implementation of the DFR, we investigate the possibility of merging deep learning and DFR computing systems. By evaluating the results, deep DFR models demonstrate 50%-81% better performance during training and 39%-64% performance improvement during testing than shallow leaky echo state network (ESN) model. Due to the difference in architecture, the training time of MI (multiple inputs)-deep DFR requires approximately 21% longer than that of the deep DFR model. Our approach offers the great potential and promise in the realization of analog hardware implementations for deep DFR systems.

Keywords

Neuromorphic computing, edge of chaos, delayed feedback reservoir computing, deep learning

1. Introduction

The rapid evolvement of computing systems was perfectly predicted by Moore's law during past few decades. However, it has been observed that the rate of enhancement is starting to saturate and slow down which indicates the end of Moore's prediction due to the fundamental physical limits of the chips. This phenomenon is also known as the 2020 roadblock [1]. The need of breakthrough directed researchers to a new path and to reconsider how biological systems, from living organism to human beings, process information.

To mimic the brain neurology, the artificial neural network (NN) [2] is built whereby electronic circuits are used to model the biological neural networks. In the early 1940s, a neurophysiologist and a mathematician, Warren McCulloch and Walter Pitts, published a paper on modeling neural network using electrical circuits [3]. Training NNs generally requires a lot of data. However, at that time, the data storage and computational capability did not facilitate the development of NNs. Nowadays, with the explosion of data and supercomputers, such as Tianhe-2 [4], the technology is up to the right standard allowing researchers to investigate on NNs. Designing novel and revolutionary computing systems that possess as low power consumption as possible has been a tantalizing goal for dedicated researchers. In the endeavor to approach this goal, neuromorphic computing was proposed by Carver Mead in the 1980s which is defined as the very-large-scale-integrated (VLSI) circuits that mimic the mammalian neurology [5]. To reduce the complexity of recurrent neural networks (RNNs), the reservoir computing architecture has been proposed in the field of machine learning. Unlike RNNs, synaptic connections within the reservoir are not trained whereby randomly generated weights are assigned. Although reservoir computing models drastically reduced the training process in RNNs, the hardware realization of reservoir computing systems is hindered by thousands of nonlinear neurons within the reservoir.

Inevitably, delay is omnipresent, especially in biological systems. For instance, approximately 20 ms of the 300 ms delay in the pupil light reflex can be accounted for by axonal conduction time, which is the time delay between two neurons [6]. Recently, another reservoir computing model called delayed feedback reservoir (DFR) computing has emerged. This model utilizes a single nonlinear neuron and delayed feedback to create a reservoir with a ring topology. The introduction of delay enables the Mackey-Glass (MG) function to handle time-delayed feedback structures in a way that mimics biophysical processes in biological neurons. Such system showed its potential to be served as the reservoir for reservoir computing in [7]. While DFRs offer simplification in hardware implementations due to the ring topology, most research focuses on photonic systems. They lack energy efficient integrate circuit design on hardware implementations.

In this paper, we introduce our analog hardware implementation of the DFR. The field of deep learning has attracted worldwide attention due to its hierarchical architecture that allows more efficient performance than a shallow structure, not only on accuracy but also the processing speed [8]. Along with the analog implementation of DFR, we investigate the possibility of merging deep learning and DFR. Two deep DFR structures, deep DFR and MI (multiple inputs)-deep DFR, are proposed and evaluated under two different time series predictions with the leaky echo state network (ESN) as the baseline comparison. From the performance evaluation, deep DFR structures outperform the shallow leaky ESN on both time series prediction tasks. With the existing analog implementation of DFR and proposed deep structures, our approach offers the great potential and promise in the realization of analog hardware implementations for deep DFR systems.

2. Deep Delayed Feedback Reservoir Computing

The field of deep learning has attracted worldwide attention due to its hierarchical architecture that allows more efficient performance than a shallow structure, not only on accuracy but also the processing speed [8]. The superior performance is a result of its intrinsic deep structure. Deep neural networks (DNNs) are constructed by multiple layers working in a fashion of processing pipeline [9]. Deep learning architecture has proven to have exceptional performance in high-dimensional data that is applicable to many fields, ranging from business to science [2]. Many performance records are broken by deep learning architectures in applications of image recognition, handwritten recognition [2, 9, 10]. The depth is generally defined as stacking multiple hidden layers in between the input and output layers. This could either defined in time or space.

RNNs could also be defined as a variant of DNNs. For RNNs, the depth arises from inherent recurrent connections, which lead to depth in time. However, the training process of such NNs is considered complex and time-consuming. In an endeavor to reduce the complexity of RNNs, reservoir computing architecture has been proposed in the field of machine learning. The architecture of the reservoir is based on the RNN. Unlike the RNNs, the connections within the reservoir are not trained whereby randomly synaptic weights are assigned. The input connections serve as the scaling of the input signal and transfer the scaled signal to the reservoir. Within the reservoir, nodes are connected in a random manner whereby the nonlinear mapping takes place. Two well-known reservoir computing models, echo state network (ESN) and liquid state machine (LSM), employ the strength of RNNs as their reservoir or liquid in which the synaptic connection within these layers are not trained. By only training the output weights, the complexity of training process has greatly reduced resulting in less computation power. Any reservoir computing systems should possess two properties, 1) high dimensionality, and 2) short-term memory [11]. Reservoir computing systems have shown its performance capabilities in various applications [12-14].

In reservoir computing, RNNs are adapted as the reservoir or so-called hidden layer. Hence, traditional reservoir computing systems are all deep in time. Recently, another reservoir computing model called the DFR computing has emerged. This model utilizes a nonlinear neuron and delayed feedback to create reservoirs with a ring topology. In this paper, we evaluate the intrinsic dynamics of the DFR model, introduce the analog hardware implementation of single DFR, and investigate its possibility of merging deep learning and DFR.

2.1. Delayed Feedback Reservoir

Delay is ubiquitous in almost every system, especially biological systems from diffusion or transport of substances conduction time of nerves, to intrinsic times for synthesis, growth, and reproduction [6]. For instance, approximately 20 ms of the 300 ms delay in the pupil light reflex can be accounted for by axonal conduction time, which is the time delay between two neurons [6].

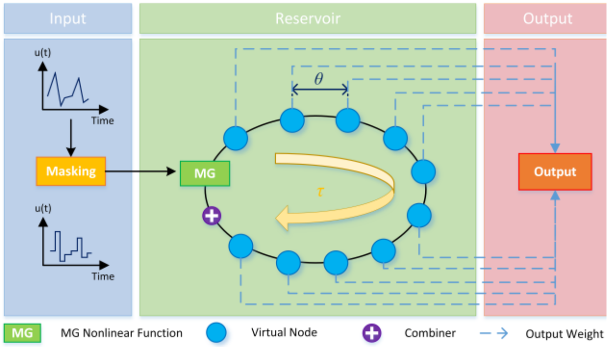


Figure 1: Delayed Feedback Reservoir Computing

The DFR computing, which is constructed by a single nonlinear neural node with dynamic delay loop, has shown its comparable performance to other traditional reservoir computing models. With the delay embedded in the system, DFR does not only have a resemblance to human brains and ease of hardware implementation but also exhibits near chaotic regime behavior, which is known as the edge of chaos. Different from the traditional reservoir, DFR possesses a ring topology. The input will be injected along with a masking scheme directly to the nonlinear node whereby the nonlinear mapping takes place. The output is then obtained through output weights. The training is commonly done by minimizing the mean square error using linear regression [15]. In [7], simulation results have shown the performance of DFR is equivalent to that of traditional reservoir computing model, ESN.

The governing equation for DFR is expressed in the form of,

$$\dot{x}(t) = -x(t) + f(x(t-\tau), I(t), \theta) \tag{1}$$

where f is a nonlinear differentiable function, also known as the nonlinear mapping, τ is the delay, $x(t)$ is the states of DFR, $I(t)$ is the input signal along with a masking scheme that injects into the DFR. In our work, the nonlinear function is chosen to be the MG function. Within the delay loop, the total delay time, τ, is divided by N equidistant delay units, which can be expressed as:

$$\tau = N \cdot \theta \tag{2}$$

where θ is the time interval between each virtual node.

2.2. Analog Integrate Circuit Design

In traditional reservoir computing, thousands of nonlinear neurons are needed to carry out tasks with adequate performance. Although reservoir computing reduces the complexity of RNNs, in hardware implementation sense, thousands of nonlinear neurons imply modifying thousands of parameters [16]. This leads to challenging hardware implementation; hence, scarce research on the hardware implementation of reservoir computing. Different from reservoir computing, due to the ring topology of DFR, the number of nonlinear nodes is drastically reduced. With only single nonlinear node simplification, DFR is much easier to be implemented in electronics [7, 16, 17]. Most hardware implementations are focused on photonic chips for DFR systems. The realization of DFR on analog hardware implementation lacks investigation. In our work, we successfully implemented the DFR onto analog hardware.

In our analog hardware implementation of a single layer DFR model, there are two important design features, nonlinear node and delay loop respectively. These circuits are shown in Figure 2.

The nonlinear node is comprised of an input trigger (M_1 and M_2) with a first order passive low-pass filter (R_1 and C_1), a nonlinear mapping transformer (R_2 and C_2), a feedback current mirror (M_3~M_5) and an output current mirror (M_6~M_8), as illustrated in Figure 2. In the reset operation, the input is charged up to V_{DD}, which deactivates the input trigger of the nonlinear device to discharge the nonlinear mapping transformer, thus, the output current, I_{out}, is reset to 0 A. In the decision-making operation, the inverted spike-based input signal is firstly filtered by the passive low-pass filter.

Once the input trigger is enabled, the input current, I_{RC}, continues to charge up the nonlinear mapping transformer to regulate the biasing voltage, V_{RC}, of M_7; consequently, the drain-to-source voltage, V_{DS}, of M_7 increases quickly and eventually reaches its saturation potential level. While M_7 is within its sub-threshold region ($V_{RC} < V_{th7}$, where V_{th7} is the threshold voltage of M_7), the V_{DS} of M_7 is nearly 0 V, the diode-connected structure of M_6 fully enables the output current mirror to achieve the maximum output current. Contrarily, as the V_{DS} of M_7 reaches its saturation potential level, the transistor M_6 is fell into its sub-threshold region, which starts reducing the output current. Meanwhile, the feedback current mirror generates a high voltage at V_{FB} to disenable the input trigger; consequently, this positive feedback loop quickly reduces the output current to 0 A until the next process takes place. As such, the nonlinear transformation of one input spike is completed.

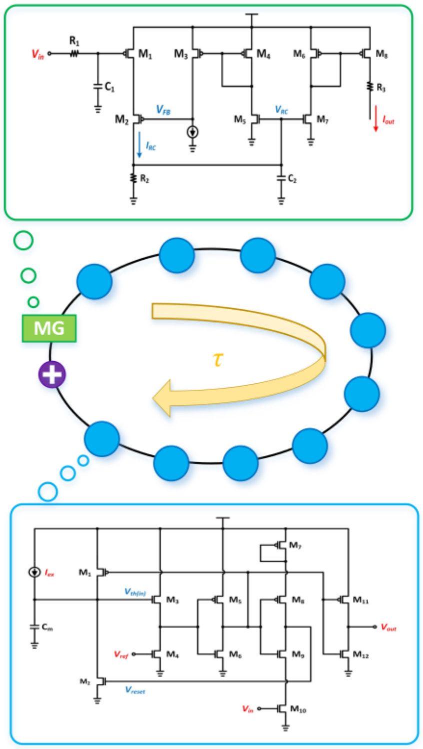


Figure 2: Analog Circuits for single layer DFR model

With the nonlinear node and delay line, we built a complete single DFR circuit model. The nonlinear transformation is mainly achieved by the charging and discharging process of the nonlinear mapping transformer; in other words, the nonlinearity of the transfer function is proportional to the time constant, τ_{ND}, of resistor and capacitor. In the integrate-and-fire-based delay unit, the delay time is regulated by the integrating time of membrane capacitor. The delay time constant, τ_{delay}, can be expressed as,

$$\tau_{delay} = C_m \cdot R_{in} \tag{3}$$

where C_m is the membrane capacitance, R_{in} is the input impedance.

2.3. Intrinsic Dynamic of Nonlinear System

As aforementioned, the MG function is employed as the nonlinear mapping function in the DFR. MG function was originally created to cope with diseases that exhibit symptom with oscillatory instabilities [18]. The governing equation for MG function is given as,

$$\frac{dx}{dt} = \frac{ax(t-\tau)}{1+x(t-\tau)^n} - bx(t) \tag{4}$$

where a, b are design parameters, and n is the nonlinearity exponent. Figure 3 shows a simplified block diagram for MG function.

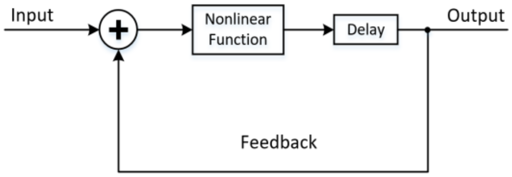


Figure 3: Simplified block diagram of MG function

The introduction of delay enables the MG function to handle time-delayed feedback structures in a way that mimics biophysical processes in the biological neuron. Such system showed its potential to be served as the reservoir for reservoir computing in [7]. Such system is described by delay differential equation (DDE). DDE equation is generally expressed as,

$$\frac{dy}{dt} = f[y(t), y(t-\tau)] \tag{5}$$

where f is an arbitrary function, which can be either linear or nonlinear function depending on the application, and τ is the time delay.

The reason that MG function is chosen is not only due to its inherent delay feature but also its rich intrinsic dynamics. The best computational performance occurs in the transition region between the stable and the chaotic regimes, which is known as the "edge of chaos". The introduction of delay enables the MG function to handle time-delayed feedback structures in a way that mimics biophysical processes in the biological neuron. Such system showed its potential to be served as the reservoir for reservoir computing in [7]. Both high dimensionality and short-term memory are achieved by delayed systems.

To closely examine dynamic behaviors, the solution to the DDE equation carried out. The dynamic behavior of the nonlinear function is modeled by the DDE with varied delay time, as demonstrated in Figure 4.

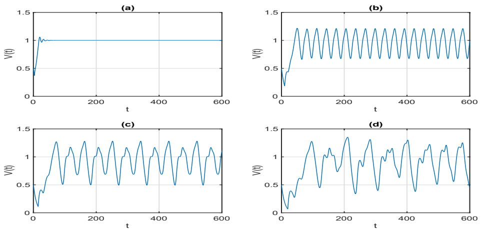


Figure 4: Dynamic behavior of nonlinear function when (a)$\tau = 3$; (b) $\tau = 12$; (c) $\tau = 16$; (d) $\tau = 20$.

As plotted in Figure 4, the solution converges to an equilibrium state when the delay is small. The dynamic behavior varies accordingly as the delay starts to increase. With the increasing of time delay, the dynamic alters from periodic to chaotic as shown in Figure 4(a) to (d).

The phase portrait is a representation of the solutions tracing the path of each particular solution. It is a graphical tool to visualize how the solutions of a given system of differential equations would behave in the long run. In other words, phase portrait is a tool to track the dynamic behavior of a system's solutions. By varying the time delay, the phase portraits are illustrated in Figure 5.

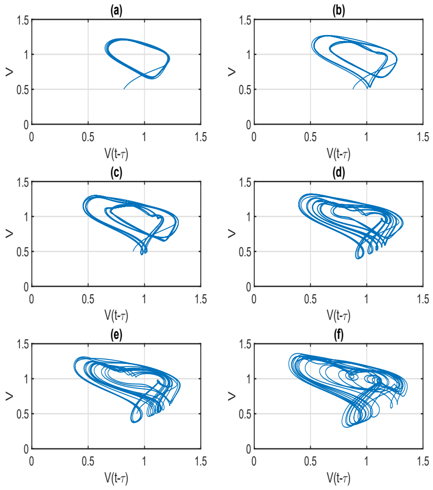


Figure 5: Phase portrait of dynamic system in (a) $\tau = 12$; (b) $\tau = 14$; (6) $\tau = 16$; (d) $\tau = 18$; (e) $\tau = 20$; (f) $\tau = 22$.

As demonstrated in Figure 5, the change in dynamic behavior can be clearly observed. As delay increases, the dynamic behavior varies from ordered to the edge of chaos to completely chaotic.

2.4. Deep Delayed Feedback Reservoir

For DFR computing systems, depth in time arises from the delayed signal that combines with the new input. However, for both RNNs and DFRs, single reservoir does not create any depth in space. Similar to stacking feedforward neural networks in deep learning field, depth in space could also be achieved by stacking multiple reservoirs on top of each other between the input and output layers.

Along with the analog implementation of DFR, we investigate the possibility of merging deep learning and DFR. Two deep DFR structures, deep DFR and MI-deep DFR, are proposed. In the deep DFR model, the output from the previous layer will be injected into the successive reservoir layers. The governing equation is expressed as,

$$\dot{x_1^l}(t) = -x_1^l(t) + f(x_1^l(t-\tau), I_1^l(t), \theta) \qquad (6)$$

where $x_1^l(t)$ is the state at l-th layer, θ is the time interval between each virtual node; f is the nonlinear mapping function which is using the MG nonlinear function as shown below,

$$f(x_1^l, I_1^l) = \frac{a(x_1^l + I_1^l)}{1+(x_1^l + I_1^l)^n} \qquad (7)$$

where I_1^l is the input signal that injects to each layer for deep DFR model, the input signal is organized as,

$$I_1^l = \begin{cases} mu_1(t) & for\ l = 1 \\ x_1^{l-1}(t) & for\ l > 1 \end{cases} \qquad (8)$$

where $u_1(t)$ is the original input signal, m is the masking, $x_1^{l-1}(t)$ is the output states from the previous layer. This topology of deep DFR model is illustrated in Figure 6(a).

The other deep structure of DFR is similar to deep DFR, but the input will be injected into each layer along with the output states from the previous layer. By adding external input to each reservoir layer, each layer would have a more recent memory of the input signal. This might be useful when carrying out prediction tasks. To verify this, two time series prediction tasks carried out to evaluate the computational performance of each deep DFR model. The governing equation for this multiple inputs-deep DFR (MI-deep DFR) is expressed as,

$$\dot{x_2^l}(t) = -x_2^l(t) + f(x_2^l(t-\tau), I_2^l(t), \theta) \qquad (9)$$

where $x_2^l(t)$ is the state at l-th layer, θ is the time interval between each virtual node; f is the nonlinear mapping function which is using the MG nonlinear function as shown below,

$$f(x_2^l, I_2^l) = \frac{a(x_2^l + I_2^l)}{1+(x_2^l + I_2^l)^n} \qquad (10)$$

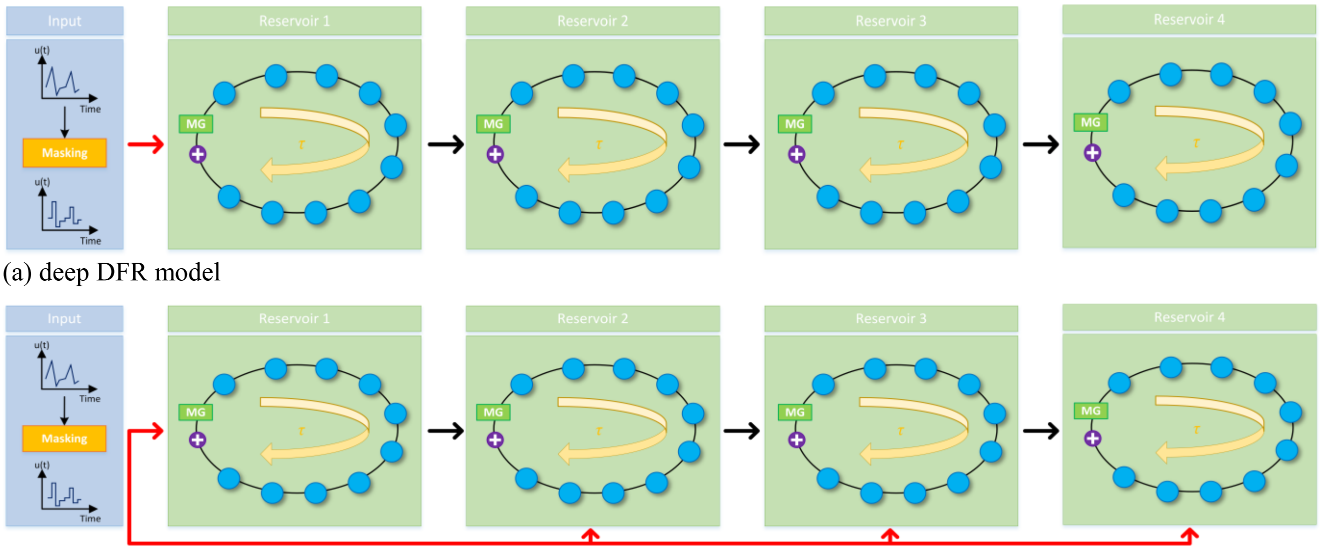
where I_2^l is the input signal that injects to each layer for MI-deep DFR model, the input signal is organized as,

$$I_2^l = \begin{cases} mu_2(t) & for\ l = 1 \\ [mu_2(t) \quad x_2^{l-1}(t)]^T & for\ l > 1 \end{cases} \qquad (11)$$

where $u_2(t)$ is the original input signal, m is the masking, $x_2^{l-1}(t)$ is the output states from the previous layer. This topology of MI-deep DFR model is illustrated in Figure 6(b).

3. Performance Evaluation

To evaluate the performance of these two deep DFR models, two time series prediction tasks carried out to study the performance of deep DFR systems. Time series prediction tasks are important in real-world applications not only in the engineering field but also in medical care [19, 20]. The computational abilities of proposed deep DFR models are examined using the normalized root mean square error (NRMSE) which is then compared to a baseline comparison model. Each deep DFR model contains 4 reservoir layers with 10 virtual nodes. In this work, the baseline comparison model is constructed by a leaky ESN model [21]. The total number of neurons used in the leaky ESN model is 40, which is equivalent to the total number of virtual nodes in the deep



(a) deep DFR model

(b) MI-deep DFR (multiple inputs-deep DFR)

Figure 6: Illustration of deep DFR models: (a) deep DFR; (b) MI-deep DFR

structures of DFR. The governing state equation for the leaky ESN is given as [21],

$$x(t) = (1 - a)x(t-1) + a\tanh(W_{in}u(t) + W_{res}x(t-1)) \quad (12)$$

$$y(t) = W_{out}x(t) \quad (13)$$

where a is the leakage term, W_{in} is the input weight, W_{res} is the weight in reservoir.

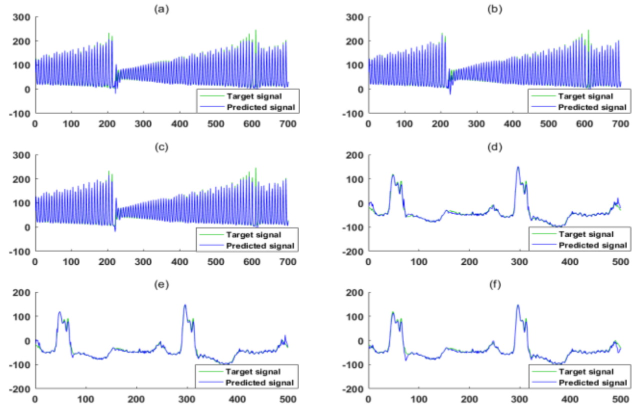


Figure 7: Target signal vs. predicted signal for (a) Santa Fe time series using leaky ESN; (b) Santa Fe time series using deep DFR; (c) Santa Fe time series using MI-deep DFR; (d) ECG using leaky ESN; (e) ECG using deep DFR; (f) ECG using MI-deep DFR

The first prediction task is the Santa Fe time series which is a typical benchmark test in the field of machine learning [16, 22]. The Santa Fe dataset utilized in the task contains a total of 6600 values, which are generated by a laser working in the chaotic region. The dataset is divided into three portions, 100 values are used for initialization, 4000 samples are used for training, and 2500 samples are used for testing.

The other prediction task carried out is the prediction of ElectroCardioGram (ECG) signal. The dataset consists of 7100 points whereby 100 samples are used for initialization, 5000 samples are used for training, and the rest samples are used for testing. Figure 7 shows a portion of the testing results for both Santa Fe and ECG time series prediction tasks. The outputs from the virtual nodes are linearly combined through output weights. During the training process, the output weights are trained by minimizing the deviation of the predicted output to the target signal or correct output signal. After the training, testing carries out using the trained output weight. Both training and testing errors are obtained by computing the NRMSE, which uses the following equation,

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N\sigma_{\hat{y}}^2}} \quad (14)$$

where y_i is the predicted output, $\hat{y}_i$ is the correct output, N is the total length of testing, $\sigma_{\hat{y}}^2$ is the variance of correct output. The training time for deep DFR models is normalized with respect to the training time of leaky ESN model.

Table 1: Training results comparison for different models

Model	Santa Fe NRMSE	ECG NRMSE	Normalized Training Time
Leaky ESN	0.0896	0.0831	
Deep DFR	0.0213	0.0418	0.45X
MI-deep DFR	0.0167	0.0319	0.57X

Table 2: Testing results comparison for different models

Model	Santa Fe NRMSE	ECG NRMSE
Leaky ESN	0.0914	0.0917
deep DFR	0.0508	0.0561
MI-deep DFR	0.0395	0.0328

The training and testing results for each model are tabulated in Table 1 and 2. As can be seen in Table 1, MI-deep DFR exhibits the lowest NRMSE for both prediction tasks among these three models during training. It is clear that in both prediction tasks, training NRMSEs for deep DFR models are lower than that of leaky ESN. By evaluating training results, deep DFR models show 76%-81% better performance than shallow leaky ESN model in Santa Fe time series prediction task. Whereas in ECG prediction task, deep DFR models exhibit 50%-62% performance improvement. Although MI-deep DFR illustrates better computational ability than that of the deep DFR model, the training time of MI-deep DFR requires approximately 21% longer than that of deep DFR. Due to the difference in architecture, there is a tradeoff between accuracy and training time.

In Table 2, testing NRMSEs are listed for different models. During testing stage, deep DFR models exhibit 44%-57% better performance in Santa Fe time series prediction task compared to that of shallow leaky ESN model. In ECG prediction task, the testing performance of deep DFR models shows 39%-64% improvement than the shallow model.

4. Conclusion

In this paper, we introduced our analog hardware implementation of a single DFR model with MG nonlinear node and a delay loop. Along with the analog implementation of DFR, we investigate the possibility of merging deep learning and DFR. Two deep DFR structures, deep DFR and MI-deep DFR, are proposed. The computational capability of these two deep DFR models is tested using two time series prediction tasks, Santa Fe time series and ECG time series respectively. The results are then compared to the shallow baseline model, leaky ESN with an equivalent number of neuron nodes. In general, deep architectures exhibit lower NRMSE than that of shallow leaky ESN model. Due to the difference in deep architecture, there is a tradeoff between accuracy and training time.

5. Acknowledgment

This material is based upon work funded by AFRL, under AFRL Grant No. FA8750-16-2-0120, No. FA8750-15-1-0052. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL.

6. References

[1] Crutchfield, J.P., W.L. Ditto, and S. Sinha, "Introduction to focus issue: intrinsic and designed computation: information processing in dynamical systems—beyond the digital hegemony." 2010, AIP.

[2] LeCun, Y., Y. Bengio, and G. Hinton, "Deep learning." Nature, 2015. 521(7553): p. 436-444.

[3] McCulloch, W.S. and W. Pitts, "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics, 1943. 5(4): p. 115-133.

[4] Xue, W., C. Yang, H. Fu, X. Wang, Y. Xu, L. Gan, Y. Lu, and X. Zhu. "Enabling and scaling a global shallow-water atmospheric model on tianhe-2". in Parallel and Distributed Processing Symposium, 2014 IEEE 28th International. 2014. IEEE.

[5] Furber, S., "Large-scale neuromorphic computing systems." Journal of neural engineering, 2016. 13(5): p. 051001.

[6] Milton, J.G., "Time delays and the control of biological systems: An overview." IFAC-PapersOnLine, 2015. 48(12): p. 87-92.

[7] Appeltant, L., M.C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C.R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system." Nature communications, 2011. 2: p. 468.

[8] Pascanu, R., C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks." arXiv preprint arXiv:1312.6026, 2013.

[9] Hermans, M. and B. Schrauwen. "Training and analysing deep recurrent neural networks". in Advances in Neural Information Processing Systems. 2013.

[10] Bueno, J., D. Brunner, M.C. Soriano, and I. Fischer, "Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback." Optics Express, 2017. 25(3): p. 2401-2412.

[11] Li, J., C. Zhao, K. Hamedani, and Y. Yi. "Analog hardware implementation of spike-based delayed feedback reservoir computing system". in Neural Networks (IJCNN), 2017 International Joint Conference on. 2017. IEEE.

[12] Zhang, X., Z. Qiao, L. Tang, W. Fan, E. Fox, and G. Wang, "Identifying Product Defects from User Complaints: A Probabilistic Defect Model." 2016.

[13] Lukoševicius, M., "Reservoir computing and self-organized neural hierarchies." 2012, Jacobs University Bremen.

[14] Li, J., L. Liu, C. Zhao, K. Hamedani, R. Atat, and Y. Yi, "Enabling Sustainable Cyber Physical Security Systems Through Neuromorphic Computing." IEEE Transactions on Sustainable Computing, 2017.

[15] Grigoryeva, L., J. Henriques, L. Larger, and J.-P. Ortega, "Optimal nonlinear information processing capacity in delay-based reservoir computers." Scientific reports, 2015. 5.

[16] Ortín, S. and L. Pesquera, "Reservoir Computing with an Ensemble of Time-Delay Reservoirs." Cognitive Computation, 2017: p. 1-10.

[17] Soriano, M.C., S. Ortín, L. Keuninckx, L. Appeltant, J. Danckaert, L. Pesquera, and G. Van der Sande, "Delay-based reservoir computing: noise effects in a combined analog and digital implementation." IEEE transactions on neural networks and learning systems, 2015. 26(2): p. 388-393.

[18] Mackey, M.C. and L. Glass, "Oscillation and chaos in physiological control systems." Science, 1977. 197(4300): p. 287-289.

[19] Gutiérrez, J.M., D. San-Martín, S. Ortin, and L. Pesquera. "Simple reservoirs with chain topology based on a single time-delay nonlinear node". in ESANN. 2012.

[20] Kantz, H. and T. Schreiber, Nonlinear time series analysis. Vol. 7. 2004: Cambridge university press.

[21] Jaeger, H., M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons." Neural networks, 2007. 20(3): p. 335-352.

[22] Solow, A.R., "Time series prediction: Forecasting the future and understanding the past." Science, 1994. 265(5179): p. 1745-1747.