
Deep Reservoir Networks with Learned Hidden Reservoir Weights using Direct Feedback Alignment

Matthew S. Evanusa *

Department of Computer Science
University of Maryland
College Park, MD
mevanusa@umd.edu

Cornelia Fermüller

Institute for Adv. Computer Studies
University of Maryland
College Park, MD

Yiannis Aloimonos

Department of Computer Science
University of Maryland
College Park, MD

Abstract

Deep Reservoir Computing has emerged as a new paradigm for deep learning, which is based around the reservoir computing principle of maintaining random pools of neurons combined with hierarchical deep learning. The reservoir paradigm reflects and respects the high degree of recurrence in biological brains, and the role that neuronal dynamics play in learning. However, one issue hampering deep reservoir network development is that one cannot backpropagate through the reservoir layers. Recent deep reservoir architectures do not learn hidden or hierarchical representations in the same manner as deep artificial neural networks, but rather concatenate all hidden reservoirs together to perform traditional regression. Here we present a novel Deep Reservoir Network for time series prediction and classification that learns through the non-differentiable hidden reservoir layers using a biologically-inspired backpropagation alternative called Direct Feedback Alignment, which resembles global dopamine signal broadcasting in the brain. We demonstrate its efficacy on two real world multidimensional time series datasets.

1 Introduction

1.1 Backpropagation Struggles with Temporal Data

While much progress has been made towards optimizing feed-forward networks with backpropagation [16], the workhorse for deep learning weight updates since its invention in the 1980s, research has struggled to expand this synchronous, time-less network structure to networks that are capable of learning temporal sequences. Effectiveness aside, while there are some voices in the community that backpropagation *can* be biologically plausible [8], the available evidence seems to indicate that the brain, while potentially performing an operation that reduces error according to some metric, cannot in fact backpropagate errors due to the synaptic structure and strict rules about symmetrical weight updates [10]. Because feed-forward networks contain no memory, one way to use feed-forward networks to encode temporal data is to remove the temporal component entirely, and flatten the entire "series" into one long vector. Alternatively, the current mode of training gradient-based recurrent neural networks (RNNs) is to directly apply the backpropagation techniques optimized for feed-forward non-temporal networks, to cyclical networks, in the form of RNNs or Long Short Term

*Corresponding author. Code available at: <https://github.com/Symbiomancer>

Memory Networks (LSTMs) [6]. These networks can be difficult to train, may require large datasets, and suffer from issues of vanishing gradients and a move away from biological plausibility, although LSTMs aim to address the vanishing gradient concerns. More recent variants of LSTMs do away entirely with components in an attempt to simplify the architecture in the Gated Recurrent Unit (GRU) [3]. Other recent branches of backpropagation-led time series learning - Transformer architectures - have eschewed entirely with the recurrent network structure [18] and go back to learning "time" as a concatenated vector. Even with these supercharged LSTM, GRU, and Transformer architectures, time series learning remains an open challenge. Here, we encode time directly in the dynamics of simplified neurons with random connectivity, directly attempting to encode the dynamical system of the real world in the dynamical system of the network - with the hopes of gleaning insights into how the brain does this.

1.2 Reservoir Computing

Reservoir computing [11] is a paradigm for recurrent neural network learning that tries to move away and relax the rigid per-time-step gradient updates of LSTMs. In the reservoir paradigm, there are three components: an input weight matrix (generally untrained) that takes in the input and expands it into a higher dimensional "reservoir". This second component, the "reservoir", contains random feedback connections, with the maximum eigenvalue set to be less than 1, that bounces around the combined input and recurrent activity for a set time period (while the activity is required to continuously "die out", in order to wash out old inputs [11]). The reservoir keeps a history of the expanded temporal input as the different time steps are fed in. Third, a "readout" mechanism looks at the patterns of activity of the reservoir and performs some learning, which can be a regression or classification task. For this paper, we show the results of a classification task, although chaotic time series are of high interest to the reservoir community [15].

1.3 Current Deep Reservoir Computing Frameworks Do Not Allow for Error-Based Hidden Temporal Representation Learning

It is well understood that the deep layers of the brain allow for increasingly more complex feature extraction, across multiple domain input types [1]. In the same way that deep ANNs leveraged deeper networks for more complex feature extraction, Deep ESNs [5] attempt to leverage the same hierarchy to extract temporal features from input time series. However, the current state of deep reservoirs does not learn hidden representations in the same manner that deep learning of ANNs does, for the simple reason that one cannot backpropagate gradients through the reservoir-layers. Unsupervised learning of these connections does exist in the form of PCA [12], although this does not allow for the same kind of powerful prediction-based error learning as in the hidden layers of deep ANNs. Here, we take one step towards a deep reservoir that can learn hidden representations with an error generated by the prediction, via direct feedback alignment. This affords the capability of learning complex *temporal* hidden representations from time series data.

2 Contributions

Our contributions are as follows:

- Introduce a novel Deep Reservoir Computer (Deep ESN), with novel trained hidden weights using Direct Feedback Alignment
- Demonstrate that Direct Feedback Alignment can induce learning through extremely non-differentiable jumps, here over randomly connected neuron pools

3 Proposed Architecture

Reservoir networks consists of the rate encoding variant known as Echo State Networks (ESN) [7], and a spike (event) based variant, known as Liquid State Machines (LSM) [13], discovered independently. Here we adopt the ESN approach, although transferring to a LSM is in development. Each reservoir updates according to the following formula of a simplified reservoir taken from [17]:

$$x_t = (1 - \alpha)x_{t-1} + \alpha f(W_{in}u_t + W_{rec}x_{t-1}) \quad (1)$$

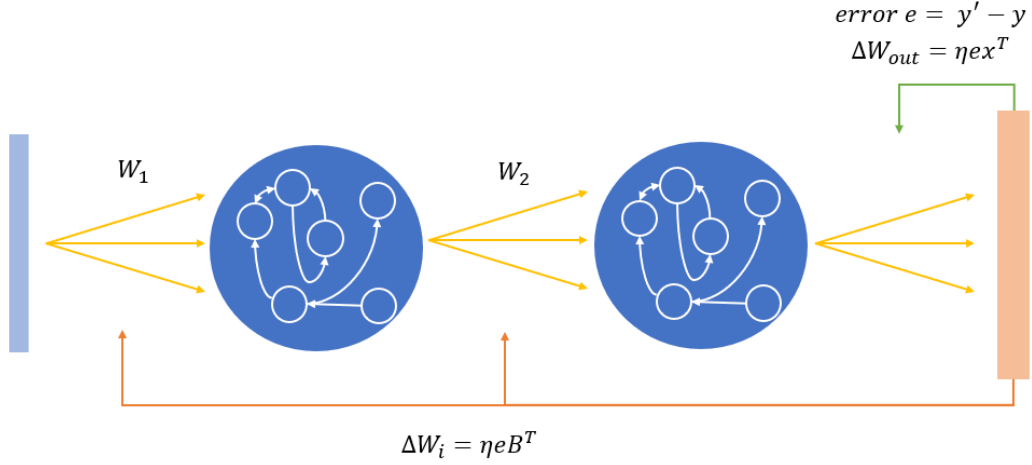


Figure 1: A schematic of the Deep Reservoir Computing with Learned Hidden Representations using Feedback Alignment framework. The hidden layer representations, stored as the W_{in} input matrices to the reservoir-layers, are learned through the direct feedback alignment learning mechanism [14]. The error for the last layer is directly fed back to the previous layers through a random matrix B into the corresponding W_{in} weight matrix for that reservoir-layer, and the network *learns to align* with this random matrix. The weights to the output from the last reservoir layer are computed using the standard delta rule for single-layer ANNs, where x is the activity vector of the last reservoir. The yellow lines indicate learned connection weights. The blue circles are the randomly connected reservoir for that layer (while shown with identical connectivity here, they are not necessarily identical). B is a random feedback matrix per [10, 14] with appropriate dimensions, η is the learning rate. Shown here are two reservoir-layers, although in our experiments we run with deeper networks effectively.

$$y_t = W_{out}x_t, \quad (2)$$

where y_t is the output vector at time t , x_t is the N -dimensional vector of the states of each reservoir neuron at time t , u_t is the input vector at time t , α is the leak rate in the interval $(0,1)$, $f(\cdot)$ is a saturating non-linear activation function (here we use a sigmoid), and W_{in} , W_{rec} , W_{out} are the input-to-reservoir, reservoir-to-reservoir, and reservoir-to-output weight matrices, respectively. If the input vector has length A , and the output vector has length B , and the reservoir vector size is N , then the dimensions of W_{in} , W_{rec} , W_{out} are $N \times A$, $N \times N$, and $B \times N$, respectively,

3.1 Feedback Alignment and Direct Feedback Alignment

Feedback alignment (FA) has been shown to be a simple yet powerful tool in the quest to move towards more local-update and biologically inspired variants of backpropagation, while at the same time, achieving the same stated end goal of reduction of prediction error for some task. In the original paper [10], the authors point out a few of the major weaknesses of relating backpropagation to biological processes: the fact that these updates must occur in synchronous lock-step, that the synaptic updates must coordinate across multiple neuron layers in a chain to deliver the correct gradient update, and that these updates require weight updates that are symmetric with the output of the neuron. Feedback alignment attempts to tackle the issue of symmetric weight updates by showing one can simply use a random feedback matrix for the gradient updates, completely unrelated to the feed-forward activity, and still wind up with weight changes within 90 degrees of the true gradient update. Direct Feedback Alignment [14] takes this idea one step further, and does away with the locality requirements of FA. There, it was shown that not only can this matrix be random, but it need not come from the layer above: all the updates can be "projected" from the output layer. This addition throws FA and DFA much closer towards biological theories about reward-modulated STDP with dopamine projections [9, 4], and other neuromodulators, with the error signal e acting as the globally-generated neuromodulator.

3.2 Training the Deep ESN with Direct Feedback Alignment

For the Deep ESN formulation [5], reservoirs are stacked in layers (which we refer to as "reservoir-layers"), analogous to how perceptrons are stacked for deep ANNs. Each sub-reservoir is given its own unique weight matrices. The reservoir equations in (1) and (2) remain the same for all hidden reservoir-layers but there are two differences. First the size of x_t : for the input reservoir-layer, x_t is the size of the input vector at time t , whereas for "hidden" reservoirs, x_t is the size of the preceding reservoir-layer. This also is analogous to deep ANNs and perceptrons, as each layer takes in as input the output from the last layer. Second, in the deep formulation, there is only a single W_{out} matrix at the terminal reservoir-layer. The W_{in} of each subsequent non-terminal reservoir-layer serves as the W_{out} for the previous reservoir-layer. At regularly sampled time intervals, the correct label y is compared against the output of the network, y' . The last layer of the network is trained using the traditional delta rule for perceptron learning (i.e., the last layer of a deep ANN) which reduces error via gradient descent; the W_{in} matrices of each preceding reservoir-layer are trained using the direct feedback alignment algorithm. The hypothesis is that the error from the last output layer will "align" these intermediary reservoir-layer weights to reduce the output error, *even though* the "true" gradient is not propagated backwards through all time steps. We illustrate the architecture in figure 1 for a simple two-reservoir-layer network. We refer to the deep ESN trained with DFA as DFA-DeepESN.

The values of the input weight matrix leading into reservoir i , W_i , are updated at each update step using the direct feedback alignment update:

$$\Delta W_i = \eta e B_i^T \quad (3)$$

where η is the learning rate, e is the last-layer error, and B^T is the transpose of the random matrix of appropriate dimension (See Fig. 1).

4 Results

We tested our network on several challenging real-world high-dimensional input time-series classification datasets: BasicMotion and ERing, which are taken from the UEA Multivariate Time Series database [2] and are freely and publicly available online. The results are shown below. BasicMotions has 6 input dimensions per time step and 4 outputs corresponding to different movements (walking, running, standing, and playing badminton) taken from a HAR sensor. ERing has 4 input dimensions of a prototype finger sensor and 6 outputs corresponding to the finger. For all series, the reservoir activity was sampled at regular intervals for readout training to the target label for the given series. The weight change updates were performed as one large batch at the end of each epoch for both the last layer and the W_i weights. For BasicMotion, reservoirs of size 800 were used. For the deep networks, 4 reservoir-layers were used for all experiments. We set α to 0.1, weight decay at $10e^{-9}$, learning rate η to 0.01, and learning rate decay at $10e^{-7}$ per epoch.

BasicMotions	train	test	ERing	train	test
Single-Reservoir	82.5	77.5	Single-Reservoir	68.15	73.3
DeepESN w/out DFA	100	97.5	DeepESN w/out DFA	93.3	78.15
DFA-DeepESN	100.0	100.0	DFA-DeepESN	96.7	79.26

Table 1: Results of the accuracy of variants of the network, either without DFA or non-deep reservoirs, on both the BasicMotions and ERing datasets. Results are shown as % correct classification accuracy.

Conclusion and Future Work

Here we propose a novel deep reservoir network with learned hidden weight matrices (DFA-DeepESN), using the Direct Feedback Alignment method. Our hope is that this work will allow future recurrent networks that have highly non-differentiable components to train effectively towards temporal predictions. Future work will involve analysis of the hidden temporal features themselves, and testing of alternative backpropagation alternatives for the weight training.

Acknowledgments and Disclosure of Funding

The authors would like to thank Vaishnavi Patil for her invaluable assistance in this endeavor. This work was supported in part by NSF award DGE-1632976.

References

- [1] C. A. Atencio, T. O. Sharpee, and C. E. Schreiner. Hierarchical computation in the canonical auditory cortical circuit. *Proceedings of the National Academy of Sciences*, 106(51):21894–21899, 2009.
- [2] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.
- [3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [4] N. Frémaux and W. Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in neural circuits*, 9:85, 2016.
- [5] C. Gallicchio and A. Micheli. Deep echo state network (deepesn): A brief survey. *arXiv preprint arXiv:1712.04323*, 2017.
- [6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [7] H. Jaeger. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.
- [8] B. J. Lansdell, P. R. Prakash, and K. P. Kording. Learning to solve the credit assignment problem. *arXiv preprint arXiv:1906.00889*, 2019.
- [9] R. Legenstein, D. Pecevski, and W. Maass. A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. *PLoS Comput Biol*, 4(10):e1000180, 2008.
- [10] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):1–10, 2016.
- [11] M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.
- [12] Q. Ma, L. Shen, and G. W. Cottrell. Deep-esn: A multiple projection-encoding hierarchical reservoir computing framework. *arXiv preprint arXiv:1711.05255*, 2017.
- [13] W. Maass. Liquid state machines: motivation, theory, and applications. In *Computability in context: computation and logic in the real world*, pages 275–296. World Scientific, 2011.
- [14] A. Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Advances in neural information processing systems*, pages 1037–1045, 2016.
- [15] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical review letters*, 120(2):024102, 2018.
- [16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [17] Z. Tong and G. Tanaka. Reservoir computing with untrained convolutional neural networks for image recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1289–1294. IEEE, 2018.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.