

Distance-Based Delay Networks*

Stefan Iacob¹, Joni Dambre¹

1. IDlab-AIRO, Ghent University-Imec

Summary. Biological neurons have finite propagation speed and non-zero distances between each other, which results in variable, distance-based delays. We implemented this phenomenon in echo state networks, by modelling reservoir units as points in a physical space. We optimized the distribution of neuron locations using an evolutionary algorithm. The resulting networks outperform regular ESNs on the NARMA-10 task.

Signal delay inherently exists in all physical and biological systems. Delay is often perceived as a physical limitation that needs to be overcome. However, inter-neuron delay is an essential computational feature of the brain. Biological neurons have a physical position in space, and neuron axons have a non-zero length and a finite propagation speed. The alignment of delays can for example be used to synchronize incoming signals and has an effect on plasticity [1]. In contrast, until recently implementations of artificial neural networks (ANNs) consisted of abstract computational units with an inter-neuron communication speed of exactly one simulation step, thus lacking distributed, location-based inter-neuron delay. In recent work, we have introduced distance-based delays in recurrent neural networks [2]. We take echo state networks (ESNs) as a starting point for our implementation. ESNs [3] are a type of rate-based recurrent neural network with randomly sampled and fixed weights. Instead of training these “reservoir” weights, a read-out layer is trained based on a teacher signal, usually using linear regression. While originally introduced as an alternative way to train RNNs, nowadays ESNs are mainly used as model systems for physical reservoirs. The ESN reservoir activity is described by

$$\mathbf{x}(n+1) = (1-a)\mathbf{x}(n) + a \cdot f(\mathbf{W}_{res}\mathbf{x}(n) + \mathbf{b}_{res} + \mathbf{W}_{in}\mathbf{v}(n)) \quad (1)$$

where $\mathbf{x}(n)$, a , \mathbf{W}_{res} , \mathbf{W}_{in} , \mathbf{b} , f , and $\mathbf{v}(n)$ are respectively, the firing frequencies at timestep n , the decay parameter, the recurrent reservoir weights, the input weights, the bias weights, the activation function, and the input at timestep n .

We implement distance-based delay networks (DDNs) by modelling reservoir nodes as points in a 2D or 3D space. The discretized distances between nodes determine the amount of simulation steps of delay that is applied. As such, an increase in firing rate from neuron A is only perceived at the input for neuron B after d steps, with d being the length of the connection between neuron A and B. This is illustrated in Figure 1.

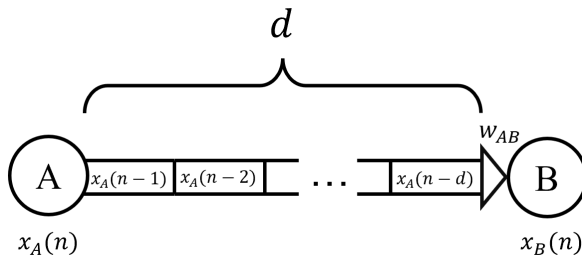


Figure 1: Diagram describing the flow of activity between DDN reservoir neurons A and B, with weight w_{AB} and a connection length of d . $x_A(n)$ and $x_B(n)$ refer to the firing frequency of neuron A and B at time n .

The neuron activity update rule for DDNs is described by the following equation, which is an extension of the conventional ESN equation.

$$\mathbf{x}(n) = (1-a)\mathbf{x}(n-1) + a\sigma(\mathbf{y}(n-1)) \quad (2)$$

$$\mathbf{y}(n) = \sum_{d=0}^{D_{max}} (\mathbf{W}_{D=d}^{res}\mathbf{x}(n-d) + \mathbf{W}_{D=d}^{in}\mathbf{v}(n-d)) + \mathbf{b}_{res} \quad (3)$$

*This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860949.

$\mathbf{x}(n)$, a , $\sigma(\cdot)$, $\mathbf{v}(n)$, D_{\max} , and \mathbf{b}_{res} are respectively, the firing frequencies of all neurons at time n , the decay parameter, the sigmoid activation function, the task input at time n , the delay applied by the longest connection, and the bias weights. $\mathbf{W}_{D=d}$ is a partial weight matrix, or a *masked* weight matrix, where all elements corresponding to connections with a delay different from d are set to 0.

In general, any reservoir can be seen as a sample from a distribution defined by a set of hyperparameters. For conventional ESNs, these hyperparameters are the connectivity of the network, the scaling of the reservoir weights, input weights and bias weights, and the leakage parameter. Additionally, in DDNs the delays in the reservoir need to be optimized to match the task at hand. Hence, neuron locations (on which the delays are dependent) need to be optimized. In order to reduce the total amount of hyperparameters, we do not optimize each neuron’s coordinate. Instead, we sample neuron locations from a multimodal Gaussian distribution (Gaussian Mixture Model, or GMM), with a fixed number of components. We treat the means and covariance matrices of this distribution as hyperparameters, to be optimized along with the conventional ESN hyperparameters. However, because we know from which component the neurons are drawn, the reservoir is naturally clustered. Thus, we can define the conventional ESN hyperparameters on a per-cluster basis. For example, instead of one global scaling parameter for all reservoir weights, we now have a scaling parameter for each cluster. Instead of having a single connectivity parameter, we have a connectivity matrix containing connectivity parameters for each cluster pair, and a diagonal that specifies the within-cluster connectivity. Although our baseline ESNs do not make use of neuron location, we can nonetheless cluster ESN neurons such that we can define hyperparameters for each cluster (pair). This is to ensure a fair comparison between the baseline ESNs and DDNs. We used CMA-ES, an evolutionary strategy, to optimize the hyperparameters of both network types. At each generation of the evolution run, candidate hyperparameters are generated, from which several DDN/ESN reservoirs are sampled. The read-outs of these reservoirs are trained using a NARMA-10 training set. We use the reservoir validation performance as a fitness measure. An evolution run was performed for both the baseline ESN and DDN models, using single-cluster models and four-cluster models. The performance on a NARMA-10 test set of the networks drawn from the best hyperparameter sets can be seen in Table 1.

Most importantly, we showed that DDNs significantly and consistently outperformed conventional ESNs. Moreover, the use of multiple clusters further improved both baseline models and DDN models.

Table 1: Average normalized root mean squared error (NRMSE) on the test set of 40 networks sampled from the best hyperparameter sets found during the evolution. Bold font indicates the best performing model type.

Type	K	NRMSE (test)
Baseline	1	0.1588 ± 0.0124
DDN	1	0.0639 ± 0.0018
Baseline	4	0.0848 ± 0.0056
DDN	4	0.0391 ± 0.0025

Our findings have important implications for the field of physical reservoirs, suggesting it should be possible to match inherently present delays to specific tasks.

Furthermore, in biological networks, delays influence synaptic plasticity. In subsequent work, we explored the effects of delays on local plasticity in DDNs by introducing a delay-sensitive local learning rule. This proved to be beneficial for learning capacity, while outperforming its delay-less counterpart.

References

- [1] H. A. Swadlow and S. G. Waxman, “Axonal conduction delays,” *Scholarpedia*, vol. 7, no. 6, p. 1451, 2012, revision #125736. DOI: 10.4249/scholarpedia.1451.
- [2] S. Iacob, M. Freiberger, and J. Dambre, “Distance-based delays in echo state networks,” in *Intelligent Data Engineering and Automated Learning – IDEAL 2022*, H. Yin, D. Camacho, and P. Tino, Eds., Cham: Springer International Publishing, 2022, pp. 211–222, ISBN: 978-3-031-21753-1.
- [3] 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*, vol. 148, no. 34, p. 13, 2001.