

Welcome!

Reservoir Computing (RC) is an approach to design, train, and analyse recurrent neural networks (RNNs). More specifically, RC offers methods for designing and training artificial neural networks, and it yields computational and sometimes analytical models for biological neural networks. The fundamental principle of RC, which distinguishes it from other views on recurrent neural networks, can be summarized as follows (see also this [Scholarpedia article](#) and this [overview article](#)):

- use a large, random RNN as an excitable medium - called a *reservoir* in this context -, such that when driven by input signals, each unit in the RNN creates its own nonlinear transform of the input;
- output signals are read out from the excited RNN by some readout mechanism, typically a simple linear combination of the reservoir signals;
- outputs can be *trained* in a supervised way, typically by linear regression of the teacher output on the tapped reservoir signals.

Reservoir computing, as a recently coined term, subsumes a number of independently found instantiations of this fundamental idea:

- *Temporal Recurrent Neural Network* ([Dominey 1995](#))
- *Liquid State Machines* ([Natschläger, Maass and Markram 2002](#))
- *Echo State Networks* ([Jaeger 2001](#))
- *Decorrelation-Backpropagation Learning* ([Steil 2004](#))

Today, RC can be regarded an established paradigm for neural computation, both as a computational technique for technical applications and as an explanatory model for processes in biological brains.

More in-depth information about RC can be found on these pages:

- [The main "flavors" of RC](#)
- [Hints for optimizing practical applications](#)

This website has been jointly conceived (and is jointly maintained) by the pioneering researchers of the RC field, and is hosted at [B. Schrauwen's Reservoir Lab](#) at Gent University. Funding for this website is currently covered through the [ORGANIC](#) EU FP7 project.
