

Reservoir Computing: uma Abordagem Conceitual

Reservoir Computing: a Conceptual Approach

Estevao Rada Oliveira^a; Fernando Juliani^{*ab}

^aFaculdade Anhanguera de Taubaté, SP, Brasil.

^bUniversidade Estadual Paulista Júlio de Mesquita Filho, Programa de Pós-Graduação Stricto Sensu em Engenharia Mecânica, SP, Brasil

*E-mail: frdjuliani@gmail.com.

Resumo

Reservoir computing é um paradigma de rede neural recorrente construída de forma aleatória, onde sua camada intermediária não necessita ser treinada. O presente artigo sintetiza os principais conceitos, métodos e pesquisas recentes realizadas sobre o paradigma de reservoir computing, objetivando servir como apoio teórico para outros artigos. Foi realizada uma revisão bibliográfica fundamentada em bases de conhecimento científico confiáveis enfatizando pesquisas compreendidas no período de 2007 a 2017 e direcionadas à implementação e otimização do paradigma em questão. Como resultado do trabalho, tem-se a apresentação de trabalhos recentes que contribuem de forma geral para o desenvolvimento de reservoir computing, e devido à atualidade do tema, é apresentada uma diversidade de tópicos abertos à pesquisa, podendo servir como norteamento para a comunidade científica.

Palavras-chave: Aprendizado de Máquina. Inteligência Artificial. Redes Neurais Recorrentes.

Abstract

Reservoir computing is a randomly constructed recurrent neural network paradigm, where the hidden layer does not need to be trained. This article summarizes the main concepts, methods and recent researches about reservoir computing paradigm, aiming to offer a theoretical support for other articles. Were made a bibliographic review based on reliable scientific knowledge bases, emphasizing researches published between 2007 and 2017 and focused on implementation and optimization of aforementioned paradigm. As a result, there's a report of recent articles that contribute in general to the development of reservoir computing, and due to its topicality, a diversity of topics that are still open to research are given, that may possibly work as a guide for the research community.

Keywords: Artificial Intelligence. Machine Learning. Recurrent Neural Network.

1 Introdução

Uma rede neural artificial (RNA) é uma técnica de inteligência computacional que consiste em um sistema de nós distribuídos que simulam neurônios biológicos capazes de aplicar funções matemáticas nas informações recebidas em sua entrada, sendo capaz de aprender a partir de referências e posteriormente generalizar a informação absorvida (SERGIO, 2013). De acordo com Sergio (2013) e Benini (2008), esta técnica é comumente aplicada em problemas de previsão de séries temporais, reconhecimento de padrões, reconhecimento e síntese de voz, resolução do problema do caixeiro viajante, entre outros. Uma das limitações impostas por uma rede neural artificial é a dificuldade de descobrirmos quais foram os passos utilizados e quais os principais elementos de sua estrutura responsáveis por uma determinada conclusão, sendo consideradas como “caixas pretas” (SERGIO, 2013). Outra característica operosa de uma rede neural artificial é o treinamento custoso e a definição de parâmetros que variam de acordo com problema, requerendo ao projetista um árduo trabalho de estruturação de certos parâmetros.

Uma rede neural recorrente é uma classe RNA que possui pelo menos uma conexão de feedback. Pertencente a

essa classe, o *reservoir computing* é uma abordagem onde o custo de treinamento é menor, devido ao fato de não se treinar a camada intermediária, apenas a camada de saída. Este modelo foi inicialmente introduzido em 2001 de forma separada como *Liquid State Machine* (LSM) e *Solid State Machine* (SSM) (FERREIRA, 2011). Em poucas palavras o *reservoir computing* consiste na construção randômica de uma rede neural recorrente cujos pesos desta são preservados, e posteriormente uma função de regressão linear é empregada para treinar a camada de saída.

O objetivo deste trabalho foi sintetizar os principais conceitos de *reservoir computing*, baseando-se na literatura acadêmica disponível e contribuir para a realização de trabalhos correlatos no futuro. Esta pesquisa é estruturada em forma de revisão bibliográfica, com conceituação e discussão de trabalhos relevantes ao tema.

2 Desenvolvimento

2.1 Metodologia

O presente artigo é uma revisão bibliográfica acerca do paradigma de *reservoir computing*, que apresenta sua conceituação teórica e seus métodos pioneiros, além de

estudos recentes que abordam técnicas de implementação, otimização e aplicabilidade do paradigma em questão. Dessa forma foi possível evidenciar o auxílio científico das pesquisas e apresentar, de forma sucinta, o estado da arte e os tópicos que ainda estão abertos ao estudo científico.

As fontes de pesquisa utilizadas foram artigos científicos presentes em importantes periódicos da área, além de trabalhos de tese e dissertação disponíveis no diretório institucional de diversas instituições. A busca foi realizada utilizando os motores de busca EBSCO Host e Google Scholar, aplicando o critério de inclusão de artigos com idade inferior a dez anos, objetivando melhor garantir o caráter hodierno do artigo.

Os termos de busca utilizados compreenderam: reservoir computing, liquid state machines, echo state networks, recurrent neural network.

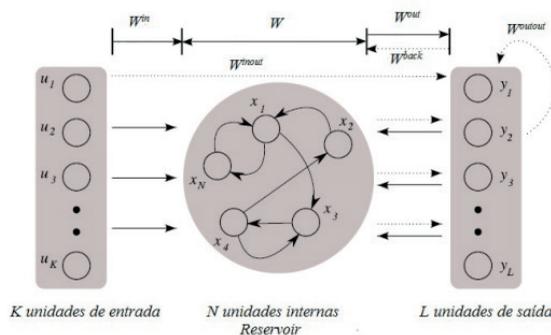
2.2 Reservoir Computing

Reservoir computing é um paradigma de modelagem eficiente de redes neurais recorrentes, que segundo Bacciu et al. (2013) tem se tornado bastante popular. Este modelo foi proposto por Verstraeten et al. (2007) que unifica as *Echo State Networks* e *Liquid State Machines* em um único campo de pesquisa. *Reservoir Computing* tem a finalidade de colocar a rede neural em um estado “sólido” pela ativação conjunta dos neurônios.

Uma topologia recorrente não-linear de elementos de processamento (*reservoir*) e uma camada de saída (*readout*) formam a estrutura básica de reservoir computing (FERREIRA, 2011). Os sinais de entradas são enviados ao *reservoir* que posteriormente os envia à camada de saída, que é um circuito reduzido. O *reservoir* também pode receber de forma facultativa os sinais da conexão de *feedback*. Uma característica excepcional é o fato da camada de saída ser única que recebe treinamento, enquanto a topologia de elementos de processamento tem pesos randômicos, tornando a atividade de treinamento extremamente eficiente neste paradigma (BACCIU et al., 2013).

A Figura 1 ilustra uma abordagem de *reservoir computing*.

Figura 1 – Arquitetura de *Reservoir Computing*.



Fonte: Ferreira (2011)

Embora *reservoir computing* seja uma abordagem eficiente, esta apresenta algumas desvantagens, como custo

computacional dispendioso e dificuldade na definição de parâmetros por pessoas pouco familiarizadas com a resolução de determinados problemas (SERGIO, 2013).

2.2.1 Echo State Networks

Proposta por Jaeger (2001), as *Echo State Networks* são métodos pioneiros de *reservoir computing* que possuem uma camada de neurônios conectados aleatoriamente que desempenham a função de memória de curto prazo (o efeito “eco” da rede). O efeito “eco” é responsável pela percepção da rede do efeito de aproximação no sistema de aprendizagem (WANG; HAN; SHEN, 2014). As Echo States Networks utilizam neurônios sigmóides simples, ou seja, neurônios ativados por uma função sigmoide.

A Equação 2.1 descreve a mudança de estados em uma rede de K inputs, N nodos no *reservoir* e L nodos de saída:

$$x(n+1) = f(Wx(n) + W^{in}u(n+1) + W^{back}y(n+1)) \quad (2.1)$$

Segundo Sergio (2013), f é a função sigmoide, podendo ser a logística ou a tangente hiperbólica, W é a matriz de peso, W^{in} a de entrada, W^{out} a de feedback, $u(n)$ e $y(n)$ o sinal de entrada e saída respectivamente. O estado estendido do sistema é dado pela concatenação de $x(n)$ e $u(n)$. A Equação 2.2 equivale a saída da rede, onde W^{out} e g é a matriz de pesos e função de ativação dos neurônios do *readout* respectivamente.

$$y(n) = g(W^{out}z(n)) \quad (2.2)$$

Além da característica de memória de curto prazo, as Echo State Networks podem ser utilizadas em medidores de frequência, atiradores caóticos, reconhecimento de formas de ondas, palavras e padrões dinâmicos, etc. (SERGIO, 2013).

2.2.2 Liquid State Machines

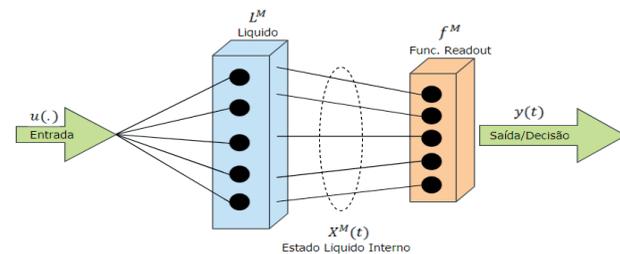
Outro método pioneiro de *reservoir computing* são as *Liquid State Machines*, proposto por Maass et al. (2002) que consiste na computação neural baseada em perturbações que se atentam ao neurônio do tipo *readout* (GRANDO et al., 2010). Este neurônio é o que recebe várias entradas de outros neurônios e as envia para outro circuito após estabilizá-las, onde vários destes neurônios podem gerar um sistema tautócrono quando treinados para realizar tarefas distintas.

Pode-se estabelecer uma relação análoga com o funcionamento das *Liquid State Machines* com uma piscina cheia de água, onde sua superfície sofre perturbações quando são arremessados itens no seu interior ou quando o vento lhe é incidente, ficando em repouso quando nenhum evento externo é aplicado. O estado de movimento do líquido reflete tanto o estado corrente quanto o passado do que lhe foi aplicado. Essa informação de eventos recentes é possível ser recuperada quando “escaneamos” a superfície da água em certo instante, assim é possível associar o conceito do líquido perturbado com os microcircuitos neurais sobre estados perturbados,

baseado na diversidade de unidades de processamento e conexões (SERGIO, 2013).

O funcionamento de uma *Liquid State Machine* é formulado por Maass et al. (2002) como um fluxo contínuo de perturbações inserido em um meio excitável (denominado “circuito líquido”). O “líquido” pode ser tudo aquilo à qual se possa recuperar o estado corrente denominado em um instante de tempo t pela função \cdot . Por fim, o segundo elemento denominado por f^M consiste na função sem memória *readout*, responsável por extrair o estado do líquido no instante t em uma saída desejada com base em uma determinada tarefa, podendo haver diversas destas funções (GRANDO et al., 2010). A Figura 2 apresenta um modelo desta arquitetura.

Figura 2 – Arquitetura de LSM.



Fonte: Os autores.

2.3 Resultados

Pelo fato do conceito *Reservoir Computing* ser relativamente novo, ainda existem diversas áreas para pesquisa, como sua modelagem, treinamento, otimização e estudo de propriedades. Este tópico sumariza as principais pesquisas recentes que contribuem para o aprimoramento da aplicação de *Reservoir Computing* e algumas de suas aplicações reportadas na literatura.

2.3.1 Implementação

Objetivando aprimorar a velocidade de implementação, investigação e comparação de arquiteturas de construídas a partir de técnicas de processamento robustas como *reservoir computing*, Verstraeten et al. (2012) propôs o *framework Oger*. Este framework é codificado em *Python* e possui diversas implementações de *reservoir* (não limitado à LSM e ESN), suprimindo a deficiência e aumentando a flexibilidade dos *frameworks* propostos até então. Além do *Oger* é possível encontrar outros diversos frameworks em diretórios na internet, como *pyReservoir* (*Python*) e *atureservoir* (*C++*).

2.3.2 Otimização

Utilizando algoritmos genéticos e o *framework Oger* (previamente denominado *rctoolbox*), Ferreira (2011) propôs um método capaz de atingir o melhor reservatório para previsão de séries temporais. O método intitulado RCDESIGN é capaz de otimizar a configuração do reservatório baseado em alguns dados específicos. Este apresenta-se como uma alternativa de baixo custo computacional quando confrontado com outros trabalhos.

Em Ludermir e Sergio (2012) é utilizada uma adaptação do método proposto por Ferreira (2011) utilizando algoritmo de otimização por enxame de partículas (*PSO*) e duas de suas extensões: *EPUS-PSO* e *APSO*. Os resultados alcançados foram comparados com outros trabalhos da literatura, apresentando-se como uma opção viável na tarefa de otimização de *reservoir computing*. Uma abordagem semelhante pode ser encontrada em Martin e Reggia (2015). Neste trabalho é criado um modelo denominado *SINOSA* que utiliza *PSO* e *self-assembly* para otimização de uma *Echo State Network*. A importância deste trabalho se destaca no fato de se demonstrar a integração de *PSO* e *self-assembly* como um meio eficaz de otimizar os pesos e a topologia de uma rede neural de forma simultânea.

Sergio (2013) propõe, para estudos futuros, a análise da otimização de *reservoir computing* utilizando algoritmos como *Ant Colony Optimization* (*ACO*), *Group Search Optimizer* (*GSO*) e a combinação destes com outras técnicas. Em Ferreira (2011), além de otimização, são sugeridos estudos acerca do entendimento da geração de medidas da dinâmica e da análise das propriedades responsáveis pela força ou fraqueza de *reservoir computing* em uma tarefa específica.

2.3.3 Aplicabilidade

Em Jalalvand (2017) é estudada a utilização de *reservoir computing* para reconhecimento e recuperação de ruídos em imagens, estendendo a aplicação do paradigma em questão ao campo de processamento de imagens. O estudo demonstrou que o corretor de ruídos proposto pode reduzir efetivamente a taxa de erro em vários tipos de ruído.

Além do exemplo anterior, a literatura tem reportado a aplicação de *reservoir computing* com sucesso em tópicos abstratos e de engenharia, Ferreira (2011) e Sergio (2013) destacam algumas aplicações de destaque: aprendizado por reforço; reconhecimento de fala e modelagem de ruído; previsão de séries temporais caóticas; classificação de padrões dinâmicos; geração de sinos autônomos; reconhecimento de imagens em eletroencefalogramas; previsão de geração de energia e velocidade eólica; previsão do caminho e do movimento de objetos e detecção de eventos.

3 Conclusão

Apresentou-se uma síntese dos principais conceitos e métodos de *reservoir computing*, além de um compilado de pesquisas recentes que abordam desde ferramentas de implementação e otimização até a aplicabilidade deste paradigma.

Pelo fato de *reservoir computing* ser uma abordagem relativamente recente, ainda existem inúmeras técnicas e tópicos que devem ser estudados. Para a otimização deste paradigma, além da utilização de outras extensões de *PSO*, propõe-se um estudo das propriedades responsáveis pela aptidão relativa de *reservoir computing* em tarefas específicas,

e a utilização deste juntamente com outros tipos de sensores ainda não testados e pouco abordados na literatura, como por exemplo narizes artificiais.

Referências

BACCIU, D. et al. An experimental characterization of reservoir computing in ambient assisted living applications. *Neural Comp. Applic.*, v.24, n.6, p.1451-1464, 2013.

BENINI, F.A.V. Rede neural recorrente com perturbação simultânea aplicada no problema do caixeiro viajante. São Carlos: Universidade de São Paulo, 2008.

FERREIRA, A.A. Um método para design e treinamento de reservoir computing aplicado à previsão de séries temporais. Recife: Universidade Federal de Pernambuco, 2011.

GRANDO, N. et al. Forecasting electric energy demand using a predictor model based on liquid state machine. *Int. J. Artificial Int. Expert Syst.*, v.1, n.2, 2010.

JAEGER, H. The “echo state” approach to analysing and training recurrent neural networks. no 148. GMD - German National Research Institute for Computer Science, 2001. Disponível em: <<http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf>>. Acesso em 2 out 2017.

JALALVAND, A. et al. On the application of reservoir computing

networks for noisy image recognition. *Neurocomputing*, 2017.

LUDERMIR, T.B.; SERGIO, A.T. Artificial Neural Networks and Machine Learning – ICANN 2012. *Lecture Notes Computer Sci.*, v.7552, p.685-692, 2012.

MAASS, W. et al. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comp.*, v.14, n.11, p.2531-2560, 2002.

MARTIN, C.E.; REGGIA, J.A. Fusing swarm intelligence and self-assembly for optimizing echo state networks. *Comp. Intel. Neurosc.*, v.2015, p.1-15, 2015.

SERGIO, A.T. Otimização de reservoir computing com PSO. 2013. Pernambuco: Universidade Federal de Pernambuco, 2013.

VERSTRAETEN, D. et al. An experimental unification of reservoir computing methods. *Neural Networks*, v.20, n.3, p.391-403, 2007.

VERSTRAETEN, D. et al. Oger: Modular learning architectures for large-scale sequential processing. 2012. Disponível em: <<http://www.jmlr.org/papers/v13/verstraeten12a.html>>. Acesso em: 23 out. 2017.

WANG, J.; HAN, S.; SHEN, N.. Improved GSO Optimized ESN Soft-Sensor Model of Flotation Process Based on Multisource Heterogeneous Information Fusion. *Scie. World J.*, v.2014, p.1-12, 2014.