2016

Emerging Adaptive Architectures for Biomolecular Computation

Matthew Fleetwood
Portland State University

Department of Electrical and Computer Engineering (ECE)

Matt Fleetwood
Professor Christof Teuscher
Portland State University, Portland, Oregon 97201-5256
E-mail: **fleet@pdx.edu**

Reservoir Computing Applications for Biomolecular Computation

Portland State UNIVERSITY

teuscher.:Lab
teuscher-lab.com

Abstract

The goal of this work is to explore applications of reservoir computing in biomolecular computation. Reservoir computing is a unique model for representing a recurrent neural network. The hidden layer is comprised of randomly connected neurons, which are linked with a single or multiple output neuron(s). The output layer is trained using a learning algorithm. The reservoir model is investigated using the Python programming language and object oriented programming. Neurons are created by bundling attributes like input data and attributes of the network, which utilize methods (for instance the sum of a dot product, the hyperbolic tangent function) to operate on data (e.g. arbitrary input arrays, two variable binary inputs). This work is motivated by the idea of using adaptable algorithms instead of hardcoding information to solve classification problems in biomolecular computation, such as identifying molecular information like presence of a virus.

Abstraction roadmap

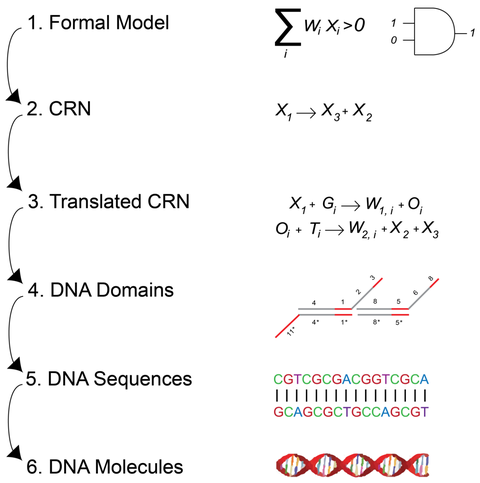

Fig 1. A roadmap by former PhD student Peter Banda captures the levels of abstraction from a formal neural network model to DNA molecules. The formal model is mapped to a chemical reaction network (CRN), which can be described by kinetics without regarding atomic structure or identity. In 3 the CRN is translated to a realistic series of chemical reactions. These reactions are then defined by DNA domains (4), sequences (5), and molecules (6). This suggests that DNA molecules can be used to carry out therapeutical reactions as governed by a higher level of design methodology, i.e. artificial neurons and networks. (Figs 1 and 5 redrawn from [2].)

Biological inspiration

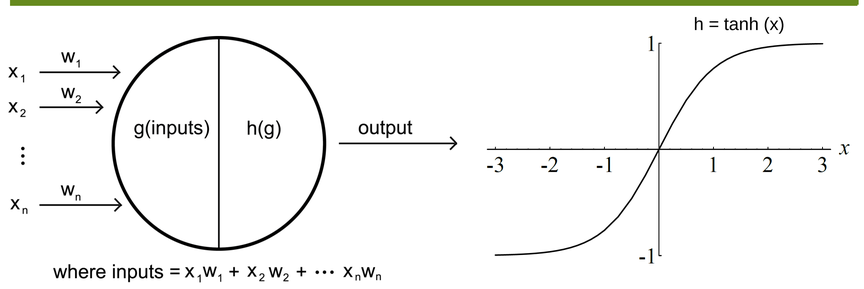

Fig. 2a An artificial neuron is inspired by and attempts to model biological neurons in brains of organisms. Individually, artificial neurons can be used with limited success to map functions.

Fig. 2b Graph of the activation function h(g), the hyperbolic tangent function [4]. This function is popular as an activation function because it is differentiable and the backpropagation algorithm can be used to train the neuron and networks of neurons.

Neural networks and DNA

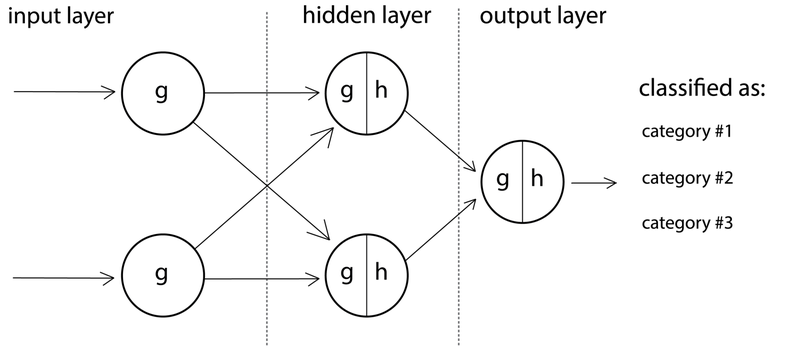

Fig 3. A feedforward neural network comprised of three columns (or layers) of 5 artificial neurons: input, hidden, and output. Feedforward networks improve the limited performance of an individual neuron by making a network out of them. Algorithmic training such as the backpropagation can modify all weights of the network such that the network improves classification performance. However this method is inefficient for networks of increasing size and complexity.

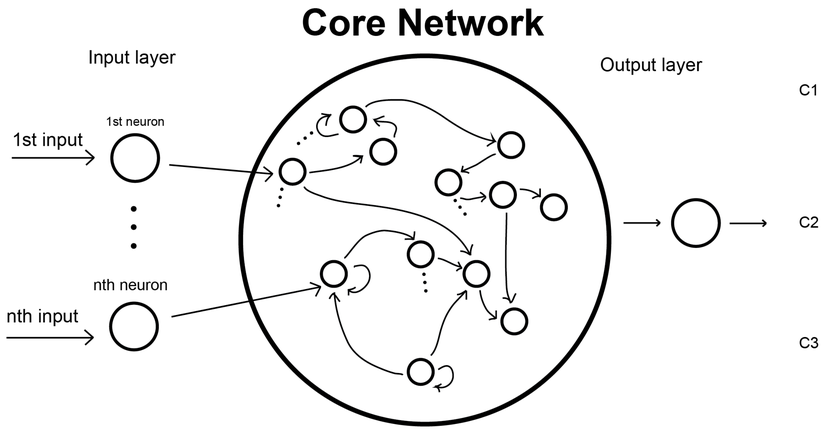

Fig 4. A reservoir computing model, which reflects a new design and training methodology for recurrent networks. The core network is made up of random and fixed connections between artificial neurons. Only the readout layer or classifier neurons are trained. This improves the efficiency of training recurrent networks because a significant amount of connections do not need to be trained.

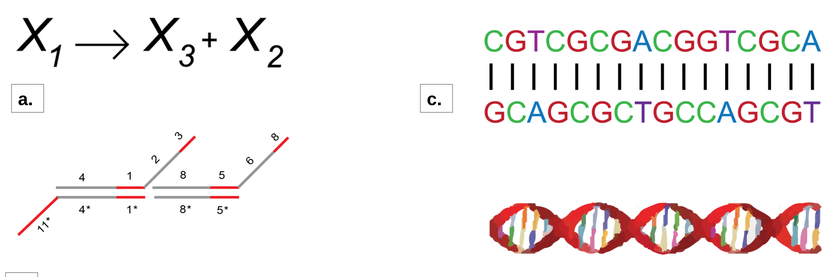

Fig 5. Reservoir networks as dynamic systems approximators of CRNs. The reservoir computing architecture is ideal for mapping to chemical reaction networks (a), which can be thought of as dynamic systems and can be described in detail using DNA domains (b) sequences and molecules (c). DNA molecules are desirable for parallelism and base-pair properties and avoid the limitations of a traditional semiconductor material such as silicone. Soloveichik et. al have also shown that DNA molecules "can be constructed that closely approximate the dynamic behavior of arbitrary systems … " [4].

Feedforward and reservoir pros and cons

Table I. Advantages and disadvantages of using a feedforward or a reservoir structure for design of a neural network.

Feedforward networks		Reservoir networks	
Pros	**Cons**	**Pros**	**Cons**
Less ambiguous structure	Slow training	Efficient training	Challenging to optimize
Guranteed convergence	All weights are trained	Only output weights are trained	Harder to construct
Universal function mapping machine	Undesirable for time series functions	Dynamic system approximator	Requires the Echo State Property

Neural network in Python

```python
class objectBlueprint(object):
    """A class is a blueprint of an object"""
    #The object uses attributes and methods (data members) to
    #operate on data provided to the object

    #Inheritance of types
    type = 'Type one'

    #Initializer / constructor for the class
    #initializes / defaults data members
    def __init__(self, input1, input2, ... , inputN):
        #Automatically done upon object creation
        self.attribute1 = defaultValue1
        self.attribute2 = defaultValue2
        self.attributeN = defaultValueN

    #Set methods set data member values based on the input
    def setAttribute1(self, data1):
        self.attribute1 = data1

    #Get methods get or return the value of a data memeber
    def getAttribute1(self):
        return self.attribute1

    #An arbitrary number and type of methods is possible
    def methodN(self, dataN):
        self.attributeN = mathOperation(dataN)
```

Fig 6. A class blueprint in the Python programming language. Python was used in conjunction with Object Oriented Programming methodologies, which encapsulate objects with data, their attributes, and methods. A class represents a blueprint for an object (such as a neuron, a connection link, and a network). Attributes are used to represent features of an object, like inputs or an output. Function definitions within the class definition are methods that operate on data given to the object and the defined attributes. Common methods typically include getters and setters for attributes and an initializer that defines default attribute values for an object.

Conclusion

Reservoir computing appears to be more efficient as a training methodology for recurrent networks. This is clear from the fact that only the output layer needs to be trained. The work presented in Python suggests step 1 in Fig 2 can be achieved with marginal error using such a programming language and OOP. Standardizing steps one through four of the roadmap would improve the accessibility of designing and applying molecular computation for desired medicinal purposes. Regardless it is evident from research in reservoir computing [3] and the abstraction roadmap that DNA chemical reaction networks [2] as modeled by a reservoir network would be ideal for therapeutic or industrial applications of biomolecular circuits.

Bibliography

[1] Banda *et al.*, "Training an asymmetric signal perceptron through reinforcement in an artificial chemistry." Journal of The Royal Society Interface 11.93 (2014): 20131100.
[2] P. Banda, "Novel Methods for Learning and Adaptation in Chemical Reaction Networks," Ph.D. dissertation, Dept. Comp. Sci., Portland State Univ., Portland, OR, 2015.
[3] Jaeger *et al.*, "Reservoir computing approaches to recurrent neural network training." Computer Science Review 3.3 (2009): 127 - 149.
[4] Soloveichik *et al.*, "DNA as a universal substrate for chemical kinetics." Proceedings of the National Academy of Sciences 107.12 (2010): 5393-5398.

The authors acknowledge the support of the Semiconductor Research Corporation (SRC) Education Alliance (award # 2009-UR-2032G) and of the Maseeh College of Engineering and Computer Science (MCECS) through the Undergraduate Research and Mentoring Program (URMP).

EducationAlliance for Science, Engineering & Technology

SRC