Emerging Adaptive Architectures for Biomolecular Computation

Matthew Fleetwood
Portland State University

Let us know how access to this document benefits you.
Follow this and additional works at: http://pdxscholar.library.pdx.edu/mcecs_mentoring
Part of the Computer Engineering Commons, and the Numerical Analysis and Scientific Computing Commons

Citation Details
http://pdxscholar.library.pdx.edu/mcecs_mentoring/8

This Poster is brought to you for free and open access. It has been accepted for inclusion in Undergraduate Research & Mentoring Program by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.
Reservoir Computing Applications for Biomolecular Computation

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.

Neural networks and DNA

![Graph showing neural network and DNA](image)

Abstraction roadmap

1. Formal Model
2. CRN
3. Translated CRN
4. DNA Domains
5. DNA Sequences
6. DNA Molecules

Biological inspiration

![Biological inspiration diagram](image)

Feedforward and reservoir pros and cons

<table>
<thead>
<tr>
<th>Feedforward networks</th>
<th>Reservoir networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pros</td>
<td>Cons</td>
</tr>
<tr>
<td>Less ambiguous</td>
<td>Slow</td>
</tr>
<tr>
<td>Training</td>
<td>All weights are trained</td>
</tr>
<tr>
<td>Universal function</td>
<td>Undesirable</td>
</tr>
</tbody>
</table>

Neural network in Python

![Code snippet](image)

![Python code](image)

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 computing 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 neural network would be ideal for therapeutic or industrial applications of biomolecular circuits.

Bibliography