Predicting RNA Secondary Structure with Reinforcement Learning

Pedro Carneiro Neto
Machine Learning in Bioinformatics, School of Science, Aalto University

Objectives

The secondary structure of RNA plays an important role in several aspects of biological systems. This project intents to aims predict these structures with reinforcement learning. Some goals are:

- A simple reinforce algorithm with monte carlo policy updates
- A simple reinforce algorithm with TD(0) policy updates
- An A2C algorithm with monte carlo policy updates
- An A2C algorithm with TD(0) policy updates

Introduction

To better understand the behaviour of biological systems, it is crucial to be able to accurately predict the structure. This prediction can be computationally made. One of the most popular algorithms is based on dynamic programming or nearest neighbours by minimizing the free energy. Nevertheless, these approaches are computationally expensive and require significant resources [1]. Other similar approaches were able to decrease the cost, although they are bounded by sub-optimal solutions [2]. Hence, it become imperative to find a new approach able to work with less computational resources and retain a high accuracy.
Reinforcement learning is a popular machine learning branch with some peculiarities and technicalities that allow it to be efficiently implemented on problems without meaningful data. It is the branch that is closer to mimic the human learning process through try and error.

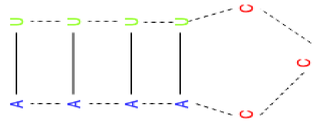

Figure 1: RNA Secondary Structure

A A A A C C C U U U U

Figure 2: RNA sequence

There are several known folded RNA sequences that can be useful to evaluate the performance of the model. For example on Figure 2 it is possible to see the RNA sequence that is folded on Figure 1. This paper is going to address a reinforcement learning approach to learn how to fold RNA sequences. The folding proposed is constrained by the following rules:

- Pseudoknot or crossing are not allowed
- A certain base must be at most paired to another one
- A-U pairing is permitted
- C-G pairing is permitted

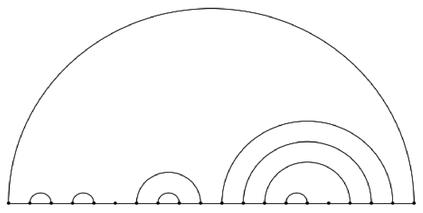

Figure 3: RNA Secondary Structure Arc Diagram created from a dot matrix representation

Reinforcement Learning

Reinforcement learning techniques are vast and build with several different components. The learning process is made by exploitation where the current knowledge is used to decide what to do next and exploration, where a random action is selected. Furthermore this process ensures a bigger navigation on the search space and a better learning, converging to the optimal and avoiding staying stuck in a local minimum [3].
State - The state represents the system at an given step. In this problem, the state is defined as the RNA structure at that given step.
Action - An action in this problem is characterized by and attempt to pair/unpair two given bases together, generating a new state.
Reward - When an action is performed over a state, the action generates a change on the state. From this change it will result a value representing how good was the action performed. Regarding this problem, the reward is the difference of connected bases on the state_t and state_{t+1}.
Environment - The state of the system, and all the necessary functions to perform a step are packed in the environment. It is responsible to initialize all the necessary components to the system, and performs the actions, generating a new state, and a reward value.
Policy - Policies are mathematical representations of the current knowledge of the system dynamics. For this project a convolutional neural network is the choosen policy.

Monte Carlo vs TD(0)

Monte Carlo - Monte Carlo is a policy update method that executes at the end of each epoch. It updates the policy accordingly to the average reward value obtained. It does not give specific information regarding specific actions.
TD(0) - TD(0) is also a policy update method that updates at each action the weights of the policy. This adds some bias to the solutions. Converge faster than Monte Carlo.

Reinforce vs Actor Critic

Reinforce - This method uses only 1 policy that predicts the probability of choosing an action. Single output.
Actor Critic - This method predicts the probability of choosing an action, but also the expected value of choosing the action. Usually has a network with 2 different output layers.

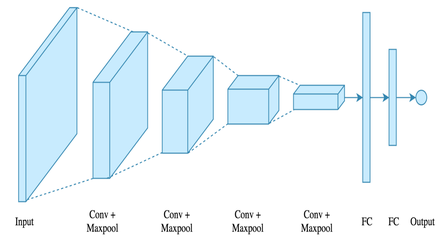

Figure 4: Example of a CNN

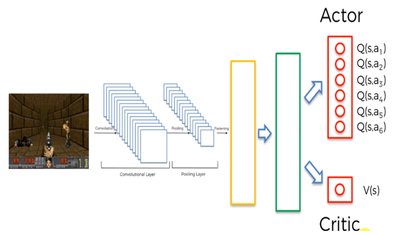

Figure 5: Example of a actor critic CNN [4]

Conclusion

The project presented some difficult challenges to address. Nevertheless the main goals were achieved and the understanding of RNA secondary structures, reinforcement learning methods and how to connect both were also achieved. Those were the key elements of the project.

References

[1] David H. Mathews, Matthew D. Disney, Jessica L. Childs, Susan J. Schroeder, Michael Zuker, and Douglas H. Turner.
Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of rna secondary structure.
Proceedings of the National Academy of Sciences, 101(19):7287–7292, 2004.

[2] M Zuker.
On finding all suboptimal foldings of an rna molecule.
Science, 244(4900):48–52, 1989.

[3] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore.
Reinforcement learning: A survey.
CoRR, cs.AI/9605103, 1996.

[4] Sameer Khan.
The Advantage of the Asynchronous Actor-Critic Algorithm, 2018.