Methodology

Dereck Piché Guillermo Martinez Jonas Gabirot

March 13, 2023

1 Task

2 Planned Methodology

2.1 Training dataset

First, we will have to generate our training data. We shall implement a simple python script which uses the NUPACK python library in order to generate a .json file containing our training data. We randomely generated a regressively labeled datasets of a milion DNA strands of length 30.

2.2 Baseline Algorithms

2.2.1 Multilayered Perceptron

2.2.2 AdaBoost

2.3 Advanced Algorithms

2.3.1 Recurrent Neural Network

2.3.2 Tranformer

Architecture The transformer architecture was initially made to translate text. However, it's subsequent use was mostly tied to token generation. This use only required the decoder to be part of the architecture, and recursively fed the predicted token in the current input while truncating if the input size was over the limit. Our task is vastly different. We are dealing with a regressive task, since we are trying to learn a function of the form $R^n \mapsto R$. Thus, we shall only keep the decoder part of the transformer architecture. We only need to set the feedworward neural network in the decoder such that it only has one output neuron.

Cost Function Since this is a regressive task, we shall use various instances of the l_p-norm as our cost function.

Tokenisation and encoding Since the tranformer learns the embedding in the attention heads, we shall simply use an integer mapping for the set of tokens $\{A, C, G, T\}$.

$$\begin{cases} 1 & a \\ 2 & a \end{cases} \tag{1}$$

as opposed to one-hot encoding.

Positionnal Embedding We will train and compare our the accuracy of our decoder-only tranformer with respect to the absence and inclusion of positionnal embedding to the input tokens.

Advantages What makes the decoder-only transformer different from other models? What are we taking advantage of by it's use? As opposed to recurrent neural networks, this model is less sequential in nature. We are predicting by taking the sequence of tokens all at once. We have high hopes for the distributivity of attention made possible by the head multiplicity. We can imagine that there is high importance between the ends of the DNA sequence, and at the center. A transformer, given enough data, would be able to take advantage of this structure in order to simplify the task.

2.4 Form of result analysis

3 Preliminary Results and Challenges Ahead

4 Changes from the initial plan