

WIP: comparison of learning techniques in artificial vs biological neural networks

Michael Ivanitsky, Connor Puritz

1 Introduction

we aim to compare the structure and learning techniques of artificial and biological neural networks, as well as assess the simplifications made by artificial neural network models. We then hope to construct a fast-learning artificial neural network by closely modeling biological networks.

2 Motivation

We assume that the graph structure of a biological neural network at some time t can be modeled by a time dependent function \mathbb{M} that transforms the edge and vertex sets of an initial graph G in some fashion. Further, we will model networks under the assumption that the initial structure of G can be approximated by the tensor product of other graphs:

$$G = (H_1 \otimes H_2 \otimes \cdots \otimes H_L)$$

The reason for this assumption:

The tensor product $C = A \otimes B$ of two graphs A, B with $|V_A| = h, |V_B| = k$, the size of the vertex set of C is $h \cdot k$. However, we can simulate a random walk on C with only $(h + k)$ memory, despite the vastly increased complexity of the graph. For our applications, of course, we are not trying to simulate a random walk, but there are computational savings nonetheless. An important distinction is that a neural networks changes with time as it learns, and this is where our function $\mathbb{M} : (G, t) \mapsto G_t$ comes into play.

3 Code outline

When storing the graph G , we store the tensor product components H_1, H_2, \dots, H_L where L is the number of layers. To specify a vertex v in G , we store in an L -element array the vertices $v_i \in H_i$ to which v corresponds to.

To store an edge e that is an addition to G or the change of an existing edge weight in G s we store the vertices v_1, v_2 which it connects, and store the edge in a data structure sorted by the input vertex v_1 . Given a vertex v_1 , this lets us easily access the edges leading away from v_1 .

priority queue for vertices?

4 Challenges

difficult to choose good initial H_1, H_2, \dots, H_L