Simulating the Evolution of Neural Pathways and Structures

Development Journal

Kári Hlynsson[1] and Young Jun Lee[2]

[1]University of Iceland, Department of Mathematics
[2]University of Oxford, Department of Biology

Contents

Index of notation

Abbreviation	Definition
$\mathscr{E} = [0; w] \times [0; h]$	Environment with *width* w and *height* h. The environment can also be expressed as the set of locations ℓ such that $\mathscr{E} = \{\ell := \langle x_i, y_j \rangle \mid x_i \in [0; w] \wedge y_j \in [0; h]\rangle\}$
n	Size of the population.
$E = (e_x, e_y)$	An entity with x and y coordinates in the environment such that $\ell_E \in \mathscr{E}$ (ℓ_E is the location which corresponds to the entity's location)
$\mathbf{C} = \{C_1, \ldots, C_k\}$	The partition of $\mathscr{E}$ into k chunks $C_1, \ldots, C_k$ such that $\bigcap_{i=1}^{k} C_i = \emptyset$ and $\bigcup_{i=1}^{k} C_i = \mathscr{E}$, i.e. the *set of chunks* $\mathbf{C}$ forms a complete partition of $\mathscr{E}$.
$\mathbf{O} = \{O_1, \ldots, O_n\}$	The population, the set of all organisms present within the environment.
$\theta_R = \frac{1}{\delta_R}$	An organism's *ray resolution*. The ray resolution is defined as the inverse of the angle between the uniformly spaced sensory rays, δ_R which construct the organism's sensory field. The higher the value of θ_R, the more rays in the sensory field and vice versa.
$\mathcal{P} : \mathbf{O} \rightarrow \ell$	Positional mapping. Returns vector representation of an entity's location at some point in time. Note that $\mathcal{P}$ is a random variable.

Common Acronyms

Acronym	Definition
CBS	Chunk-Based System
UGP	Undirected-Graph Partitioning
BFS	Breadth-First Search

Chapter 1

Theoretical Basis

Chapter 2

Model Outline

2.1 Model Outline

The aim of the model is to study the natural evolution of neural pathways in a population of organism when exposed to survivalistic conditions. A rigid logical and syntactical foundation will make all succeeding articulation on the model parameters and attributes easier. We therefore dedicate this first section towards establishing a foundation of terms and definitions which we build on later.

The most critical aspects of the model we define here is the *environment* and the *entities* contained therein. Neglecting any elevation, we define the environment as the bounded subset of the Cartesian plane, which we symbolize $\mathscr{E}$ [1].

Contained within the environment are *entities* which we can think of as actors within the simulation. The two types that occur in this model are *organisms* and *food*. Again, a simple intuitive definition is that the organism is an individual of a species present within the environment and nutrition is the foodstuffs which it consumes to gain energy and thus survive.

Entities can be divided into two types: *organisms* and *food*. What follows is simple: organisms are motile, can sense their surroundings and consume food to gain energy. On the other hand, food has none of these qualities. We represent an entity as the object E, while organisms are denoted O and food by F.

[1]Although elevation certainly plays a vital role in the foraging patterns of organisms in natural environments, we refrain from its implementation as it only adds a level of complexity to the model design while minimally increasing simulation findings.

2.2 Simulation Phases

2.2.1 Foraging phase

2.2.2 Reproductive phase

2.3 Runtime optimization

One of the run-ins we've had so far is determining how to design the sensory mapping capabilities of organisms within the environment. By sensory mapping, I am referring to the organism's ability to sense its proximal surroundings, sensing the proximity and types of the various entities they may encounter. This will be fed into their neural network, which outputs some response which instructs the organism how to behave given its current surroundings.

The first attempt I made was in the days where the environment was grid-based instead of a float-based environment. There, sensory mapping was quite easy as all that had to be done was inspect the proximal tiles and check for the entity type present in the tile. This is not possible in the float-based environment, so we propose another solution.

An excellent idea you came up with was the idea of partitioning the environment into separate chunks, which organisms restrict their sensory mapping to unless there sensory fields intersect another adjacent chunk (more on that later). We will start by discussing this idea, which as you will see, will be of great use.

2.3.1 Chunk partitioning

In this section, we will be doing a mathematical analysis of the chunk system to see how it will benefit the simulation. To start off, we inspect what fundamental laws apply to this system.

Proposition 1. *Let $\mathscr{E}$ be an environment paritioned into k chunks such that* $\mathbf{C} = \{C_1, \ldots, C_k\}$. *The probability of an entity being present in a generic chunk C_i equals $1/k$, i.e.*

$$\Pr\{E \in C_i\} = \frac{1}{k}$$

Proof. Let $\mathscr{E}$ be the space $[0; w] \times [0; h]$ with $\text{area}(\mathscr{E}) = wh$ and the partition $\mathbf{C}$. Under the assumption that the chunks are of uniform size, we assume

$$\text{area}(C_i) = \frac{\text{area}(\mathscr{E})}{k} \tag{$*$}$$

for all $C_i \in \mathbf{C}$ where $i \in [1; k]$. Under conventional probability theory, we can express the probability of an entity being in a generic chunk as the area of that particular chunk over the area of the environment, i.e.

$$\begin{aligned} \Pr\{E \in C_i\} &= \frac{|C_i|}{|\mathscr{E}|} \\ &= \frac{\text{area}(C_i)}{\text{area}(\mathscr{E})} \\ &= \frac{1}{k} \end{aligned}$$

The result of the calculations above are immediate of the definition of the area of the chunks, which is derived in ($*$). □

Definition 1 (Chunk load)**.** The random variable $\mathcal{L}$, or the *chunk load* of some generic chunk C_i, denotes the number of entities contained within the chunk. Immediate of proposition 1, we have that $\mathcal{L} \sim \text{Bin}(n, 1/k)$, where n is the total number of entities in the environment. [2]

2.3.2 Adjacent chunk loading

The following algorithm can be easily

2.4 Performance comparison

[2]Note that this assumes the uniform distribution of entities within the environment, which is obviously true. This is because there is no logical restraint on where an entity can be at any given time, i.e. there is not a consistent probabilistic hindrance in an entity having a certain position.

Appendix A

Preliminaries

Definition 2 (Probability)**.** Let ω be some event from the probability space Ω. We represent the probability of the event ω occuring using the notation $\Pr\{\omega\} = x$ where $x \in [0; 1]$.

Definition 3 (Random variable)**.** A random variable X is the mapping from the probability space Ω to the real number line, i.e. $X : \Omega \to \mathbb{R}$.

Remark. An example of a random variable is the varying height of a population, where Ω is the space of all possible outcomes and the random variable H (height) is for example 180 cm, or 157 cm.

Definition 4 (Expected value)**.** Let X be a random variable. The expected value is denoted $\mathbb{E}[X]$.

Definition 5 (Probability distribution)**.** Let X be a random variable. When it follows for example the Poisson distribution, we write $X \sim \text{Poisson}(\lambda)$.