Morphozoic

Cellular automata with nested neighborhoods as a metamorphic representation of morphogenesis

Objectives

- Devise an abstraction that models morphogenesis in a cellular automaton (CA) using a nested neighborhood approach.
  - Neighborhoods provide a straightforward representation of local vs. global fields.
  - Balance local vs. global information to achieve morphogenesis.
- Embody a scheme that is computationally feasible:
  - Each cell has a limited information processing capacity.
  - Information about a more local cell neighborhood having fewer cells is more precise and finer grained than information about a larger, more global neighborhood.
  - More formally:
    - A specific number of bits are used to represent each neighborhood.
    - The most local neighborhood is represented precisely; larger neighborhoods are increasingly “fuzzy” because they cannot be represented by the available bits.

Definitions

- A cell possess a state.
- A cell emits, senses, and reacts to signals.
- Signals carry information about the state of neighborhood cells.
- A field is the confluence of signals sensed by each cell.

Requirements

- Compact.
- Noise tolerant.
- Generalizes from exemplars.
- Evolvable by “natural” processes.

Cellular automaton

- A cellular automaton (CA) is a two-dimensional (2D) grid of cells.
- Cells have a state value.
  - In Morphozoic the cell state is a type, including an optional empty type.
- Each cell senses the presence of nearby cells, typically a 3x3 Moore neighborhood.
- The cell states in a cell’s neighborhood and a set of rules determine how the cell state changes.

Morphogen

A *morphogen* is a morphogenetic field modeled as a set of nested neighborhoods in a cellular automaton (nnCA):

A cell defines an elementary neighborhood:

$neighborhood_0 = cell$

A non-elementary neighborhood consists of an *NxN* set of *sectors* surrounding a lower level neighborhood:

$neighborhood_i = NxN(neighborhood_{i-1})$

where *N* is an odd positive number.

Morphogen (cont.)

A morphogen is defined as a set of nested neighborhoods:

$morphogen(cell) = \{ neighborhood_0(cell), neighborhood_1(neighborhood_0), \ldots neighborhood_n(neighborhood_{n-1}) \}$

Where the number of cells in $neighborhood_i = N^i x N^i$

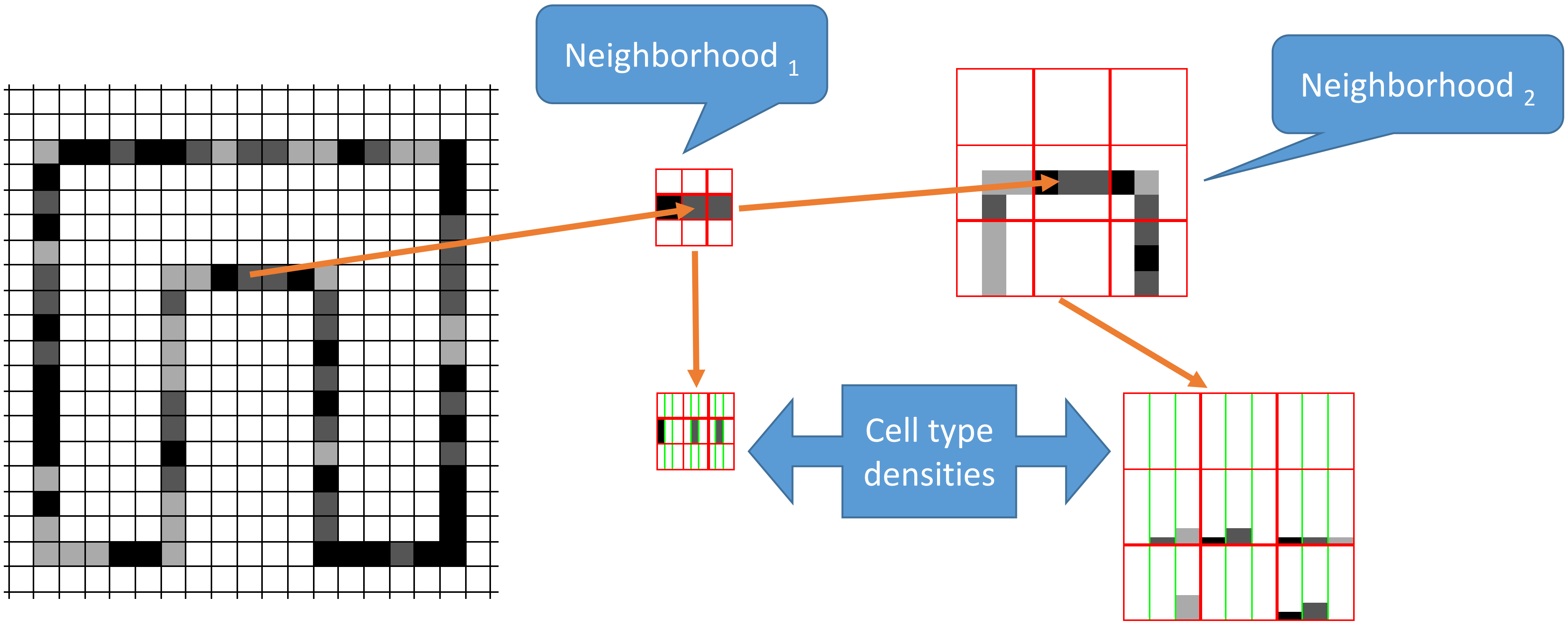
Morphogen (cont.)

The value of a sector is a vector representing a histogram of the cell type densities contained within it:

value(sector) = (density(cell-type$_0$), density(cell-type$_1$), … density(cell-type$_n$))

The number of cells contributing to the density histogram of a sector of *neighborhood$_i$* = $N^{i-1} x N^{i-1}$

Morphogen (cont.)



Metamorph

- A *metamorph* embodies a cellular automaton state→action rule, defined as a mapping from a morphogen to a cell type.
- Aspects:
  - Generation of metamorphs.
  - Execution of metamorphs.

Metamorph generation

- A set of metamorphs can be generated from a manual or programmed sequence of cellular automaton configurations.
- For example, the Game of Life application uses the programmed Game of Life rules to process the CA cell states.

Metamorph execution

- Once generated, metamorphs can be used to "execute" the application as CA rules.
- Metamorph execution consists generating a morphogen for each cell and finding the closest morphogen contained in the generated metamorphs, where:

$$distance(metamorph_i, metamorph_j) = \sum_{x}^{neighborhoods} \sum_{y}^{sectors} \sum_{z}^{cell\,types} abs(cell\ type\ density_{i,x,y,z} - cell\ type\ density_{j,x,y,z})$$

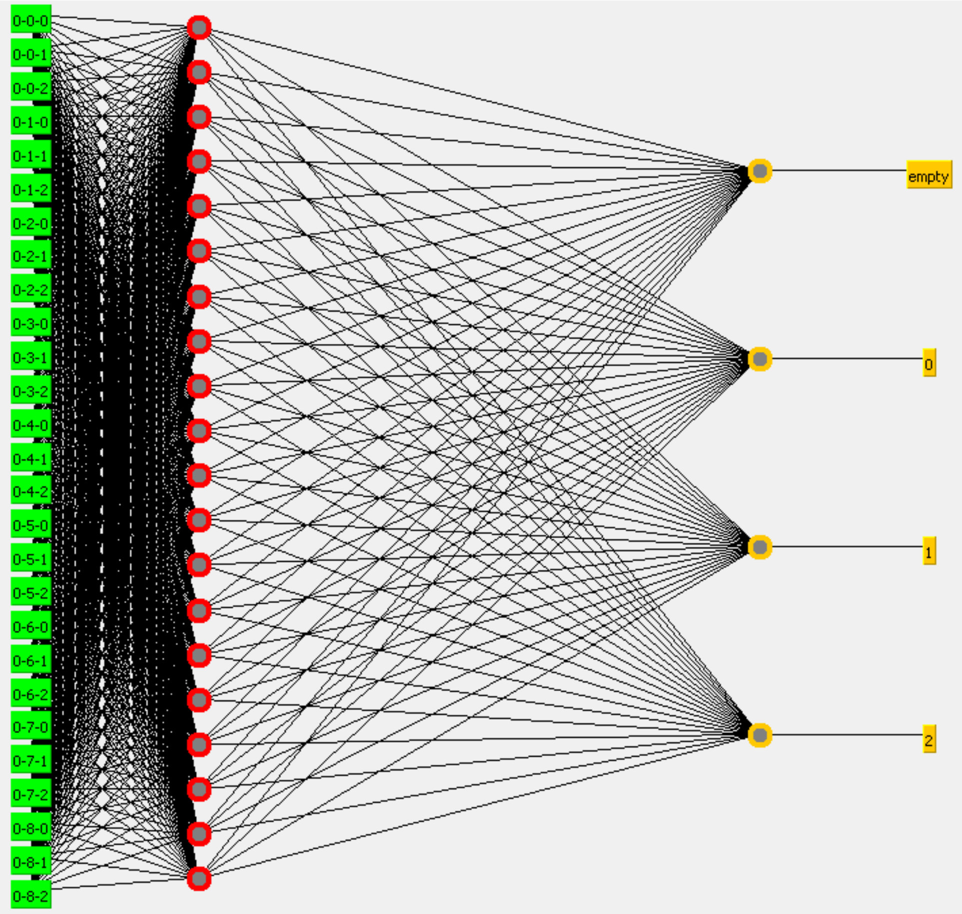
Metamorph artificial neural network implementation

- Alternatively, instead of searching a database of metamorphs, the morphogen can be input to an artificial neural network (ANN) that has been trained with generated metamorphs to map morphogen inputs to cell type outputs:
  - Faster.
  - More compact.
  - More noise tolerant.

Metamorph artificial neural network

Input: morphogen neighborhood cell type densities

Key: neighborhood-sector-cell type



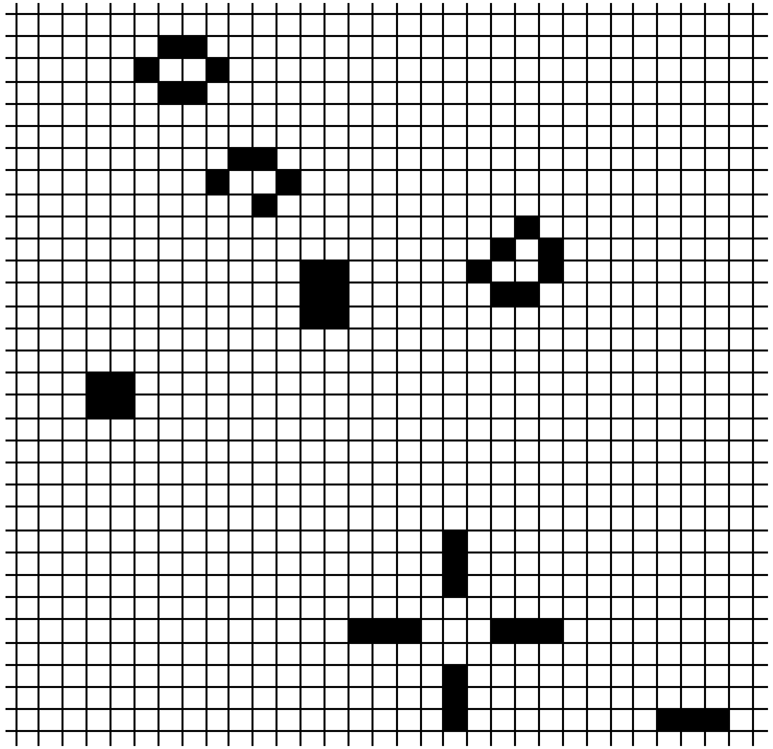
Output: cell type

Applications

- Conway’s Game of Life
- Cell regeneration
- Evolution of a simple gastrulation sequence.
- Neuron path finding demonstrating exemplar generalization.
- Simulation of Turing's reaction-diffusion morphogenesis.

Conway’s Game of Life

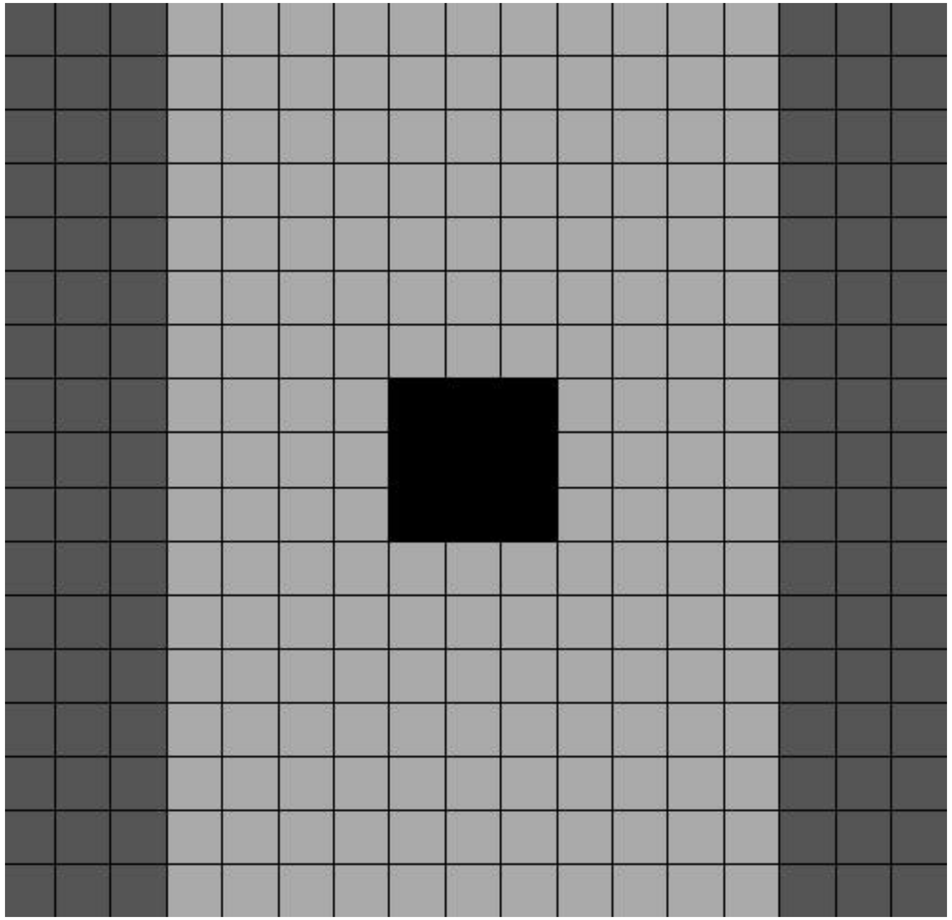
Rules:
Any live cell with fewer than two live neighbors dies, as if caused by under-population.
Any live cell with two or three live neighbors lives on to the next generation.
Any live cell with more than three live neighbors dies, as if by overcrowding.
Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

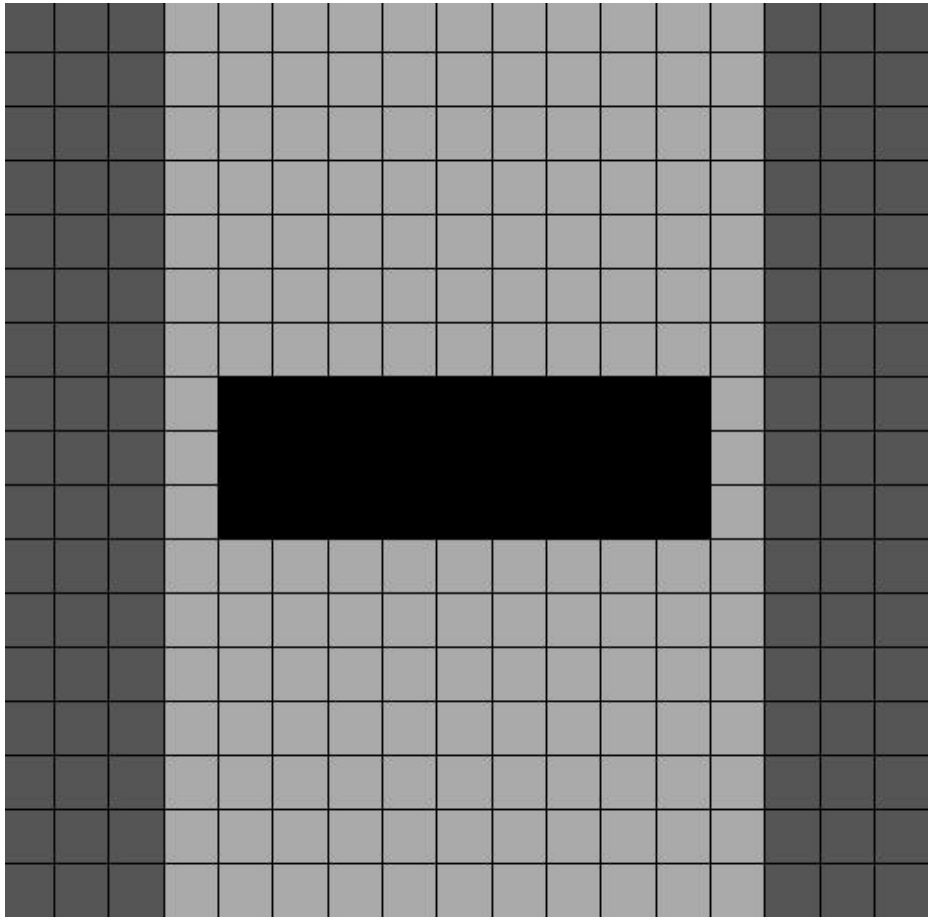


Game of Life (cont.)

- Demonstrates ability to “reverse engineer” a well-known set of CA rules.

Cell regeneration

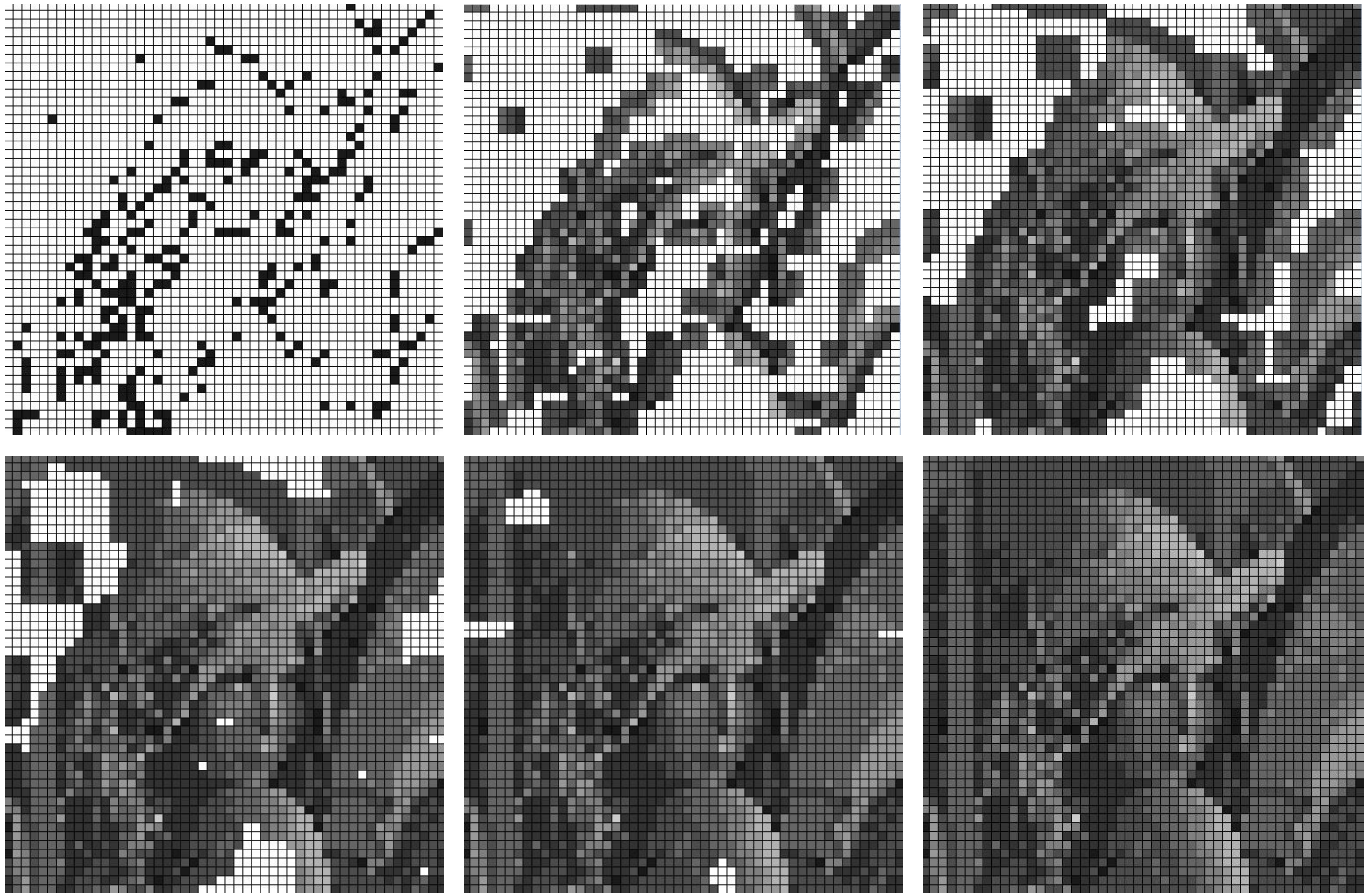




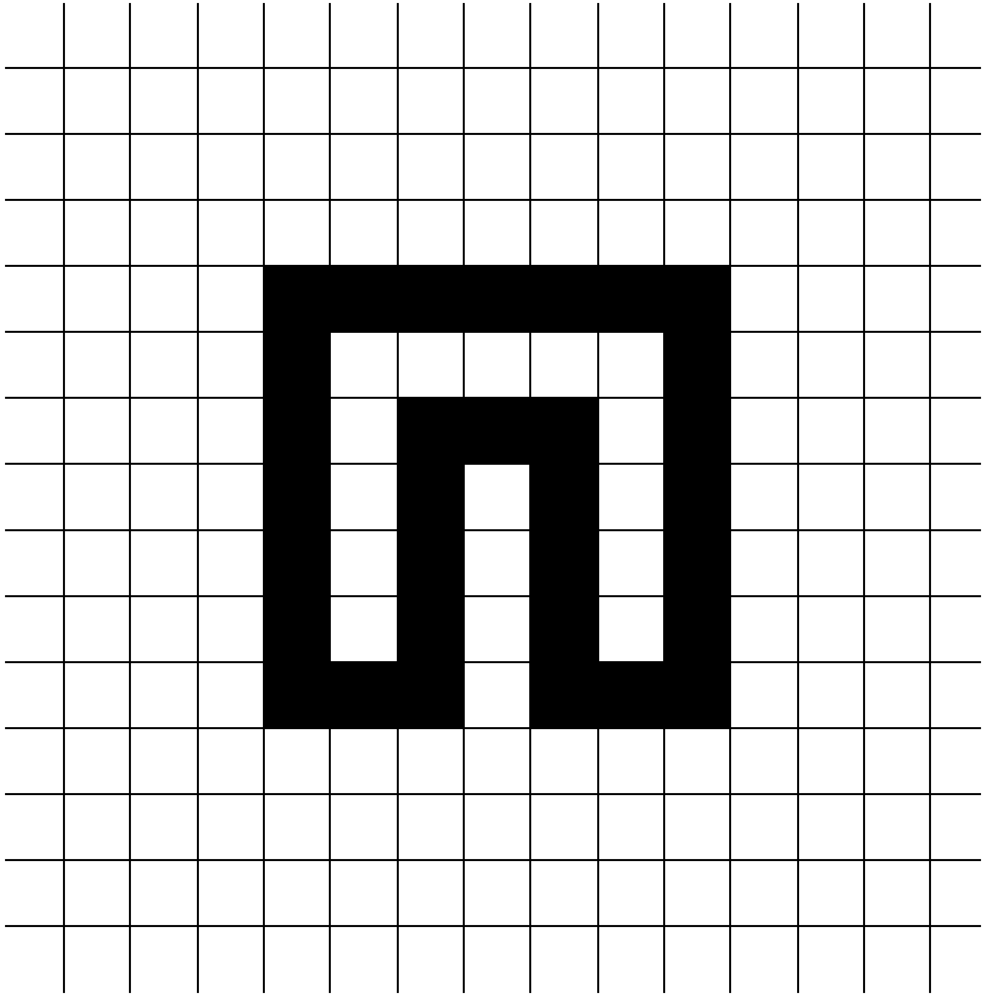
Cell regeneration(cont.)

- Demonstrates ability regenerate a pattern from an incomplete one.

Image repair using cell regeneration technique



Gastrulation evolution

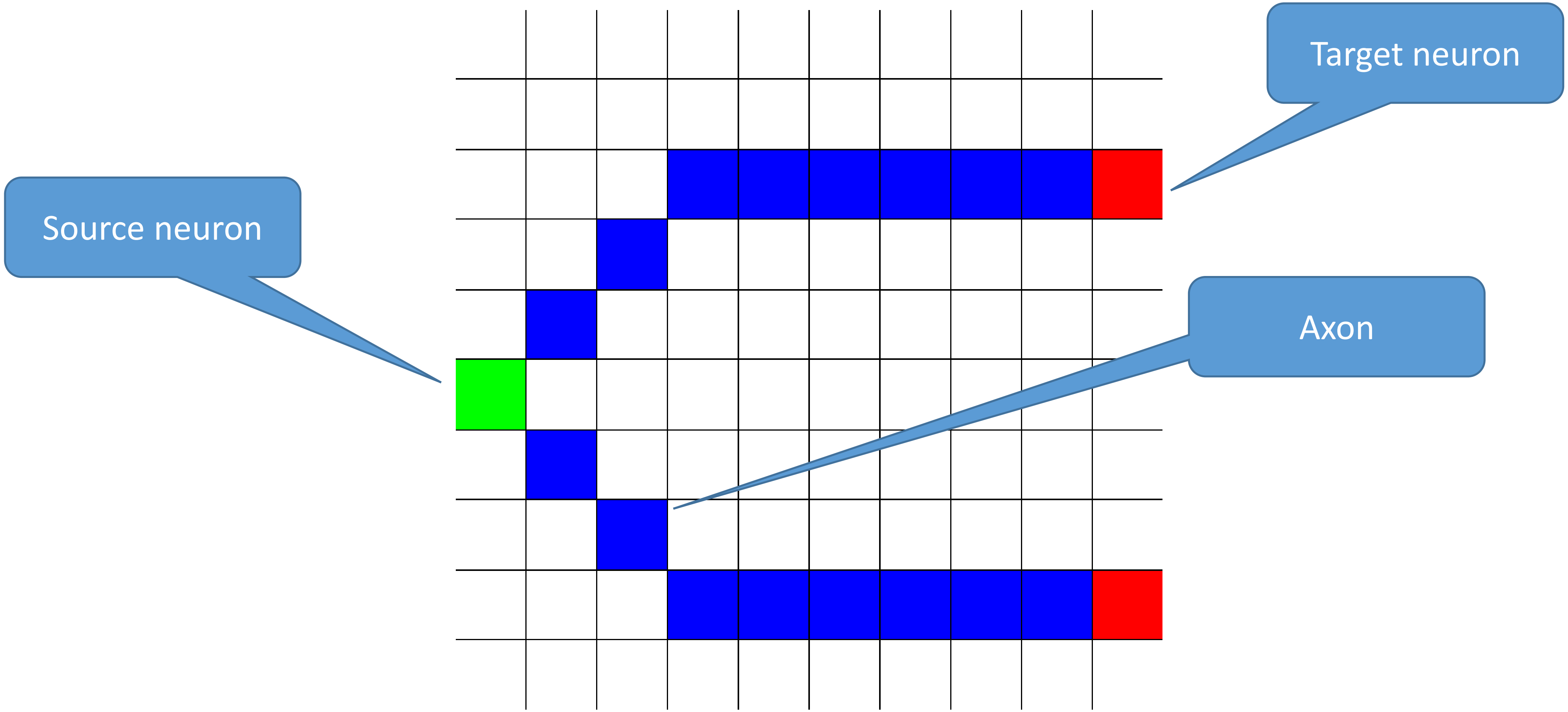


```
Fittest:
Member  Fitness
0       2475.0
_______________
_______________
_______________
_______________
____0000000____
____0_____0____
____0_000_0____
____0_0_0_0____
____0_0_0_0____
____0_0_0_0____
____000_000____
_______________
_______________
_______________
_______________
Parameters:
ORGANISM_DIMENSIONS = width:15 height:15
NUM_CELL_TYPES = 1
NEIGHBORHOOD_DIMENSION = 3
NUM_NEIGHBORHOODS = 2
NESTED_NEIGHBORHOOD_IMPORTANCE_WEIGHTS = null
MAX_MORPHOGEN_COMPARE_DISTANCE = 1000.0
METAMORPH_DIMENSION = 3
MAX_CELL_METAMORPHS = 1
METAMORPH_RANDOM_BIAS = 0.0
PROBABILISTIC_METAMORPH = true
INHIBIT_COMPETING_MORPHOGENS = false
MORPHOGENETIC_CELL_DISPERSION_MODULO = 1
METAMORPH_EXEC_TYPE = SEARCH_TREE
DEFAULT_ORGANISM = morphozoic.applications.Gastrulation
RANDOM_SEED = 48
Organism saved to morph_fittest.dat
```

Gastrulation (cont.)

- Demonstrates that the model is mutable and evolvable.

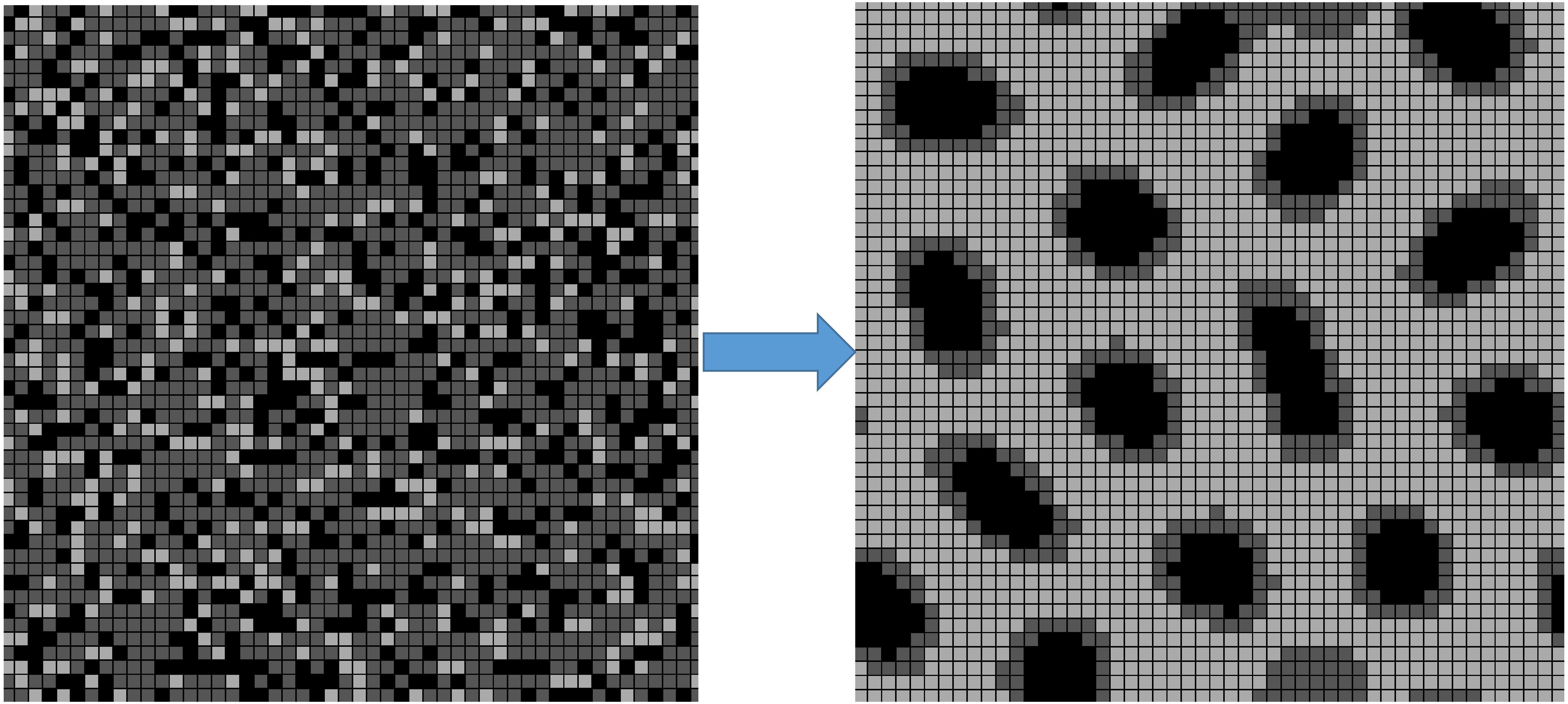
Neuron path finding



Neuron path finding (cont.)

- Metamorphs generated from multiple random configurations.
- Artificial neural network trained from metamorphs.
- Produces successful path finding on novel patterns.
- Demonstrates generalization from exemplars.

Turing’s reaction-diffusion morphogenesis



Turing (cont.)

Morphozoic output.

- Starts with different random pattern.
- Uses ANN for smooth approximation.
- Demonstrates simulation of a well-known morphogenesis algorithm.