Introduction to Neural Networks

Johns Hopkins University
Engineering for Professionals Program
605-447/625-438
Dr. Mark Fleischer

Module 14.1: Cellular Neural Networks

My Brain

In This Module We Will Cover:

- State-Based Representation of Linear Systems
- Cellular Automata—brief intro
- Cellular Neural Networks

Linear Systems

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}$$

State Equation Representation of a
Continuous, Time Invariant, Linear System.

Given a set of initial conditions, the trajectories (values through time) of $\mathbf{x}(t)$ and $\mathbf{y}(t)$ can be determined.

A good foundation for thinking about and modeling complex systems!

Cellular Automata

- Explored in the 1940s by Stanislaw Ulam and John von Neumann
- Exhibits complex behavior
- Sensitive dependence on initial conditions
- Paradigm for exploring how simplicity evolves to complexity

Game of Life

- Conway formulated a simple 2-state CA.
- Very simple rules govern the evolution of states.
- All states updated synchronously.

Game of Life

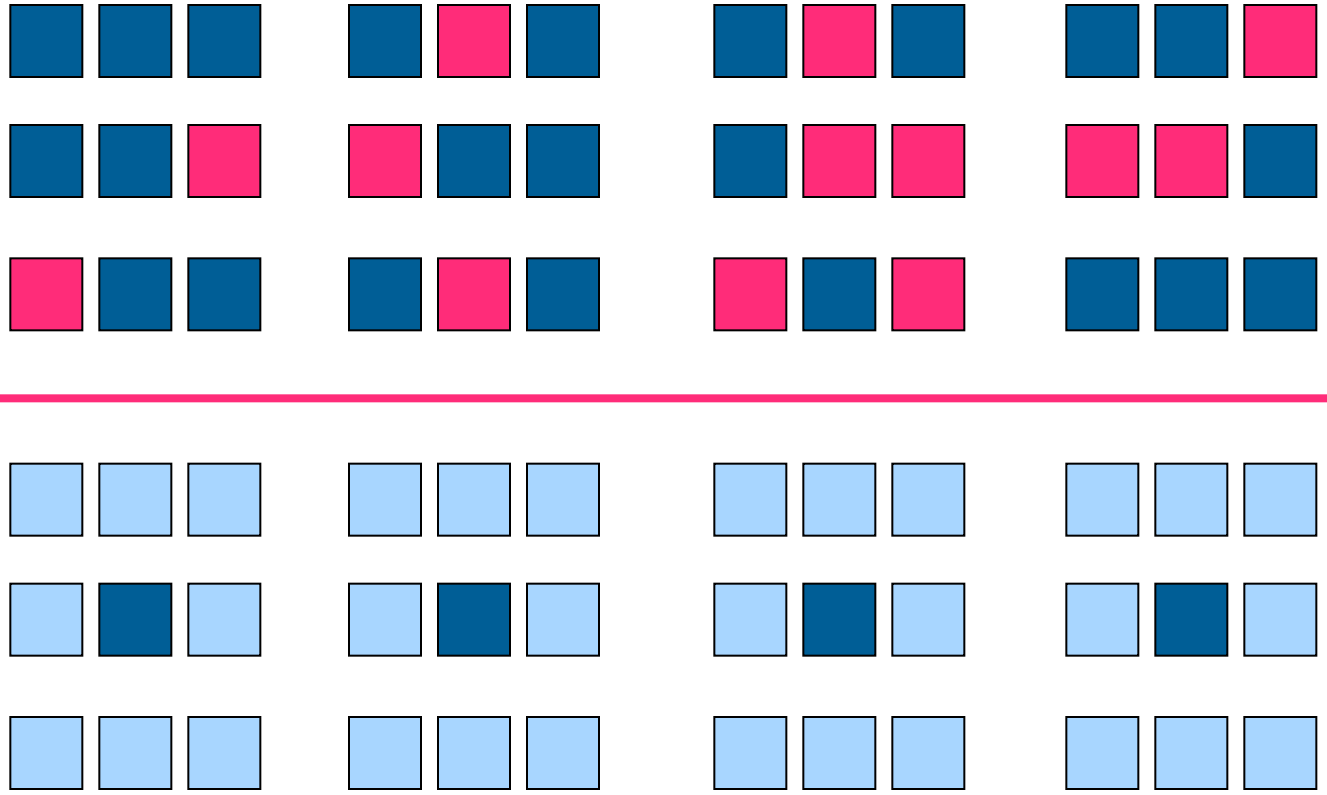
- 2 dimensional array of “cells”
- Each cell can be in one of two states: “alive” or “dead”.
- Each cell’s state in the next iteration depends on how “crowded” it is within its neighborhood.

Game of Life

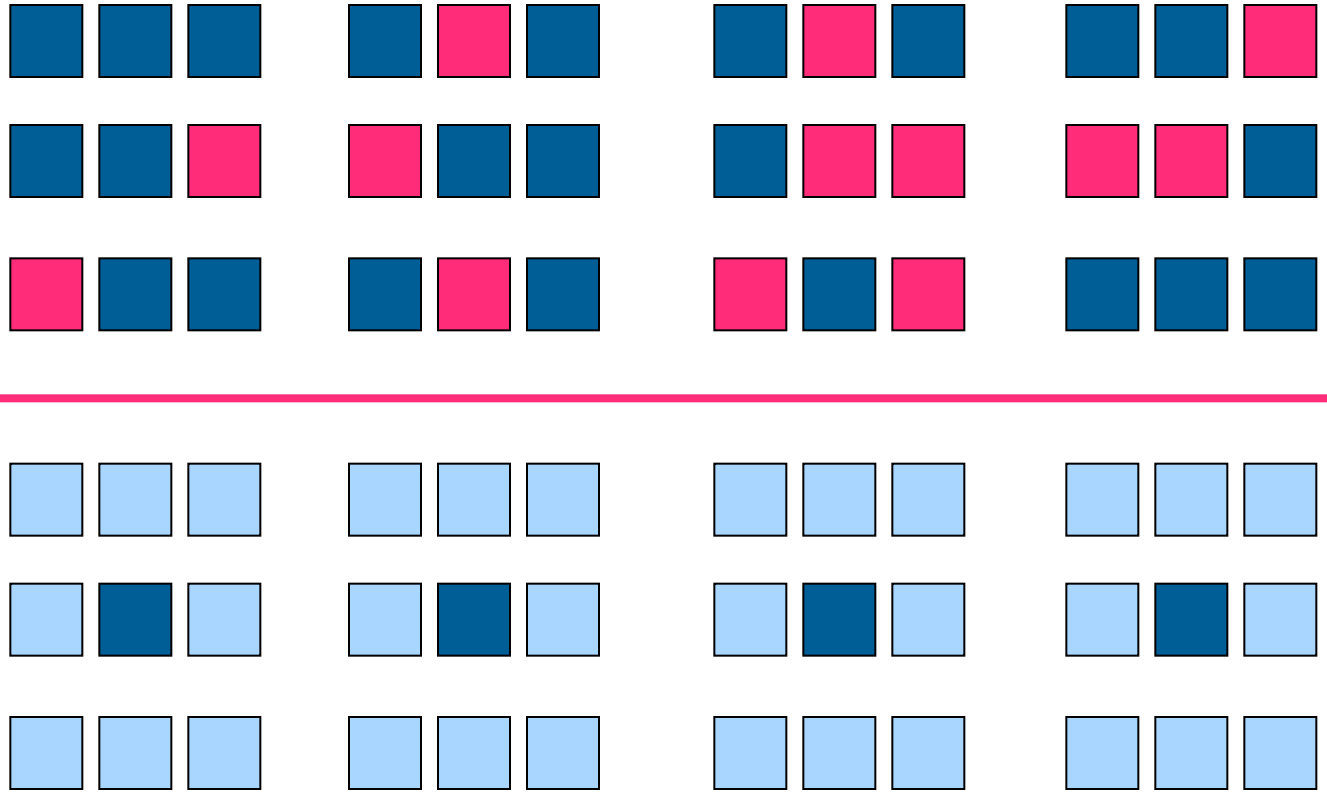
Simple rules to determine state in next iteration:

- If current state of a cell is 'dead', it becomes 'alive' (i.e., turns 'on') if and only if 3 neighbors are alive.
- If current state of a cell is 'alive',
  - a cell with 0 or 1 neighbors alive dies
  - a cell with 2 or 3 neighbors alive lives
  - a cell with 4 or more neighbors alive dies

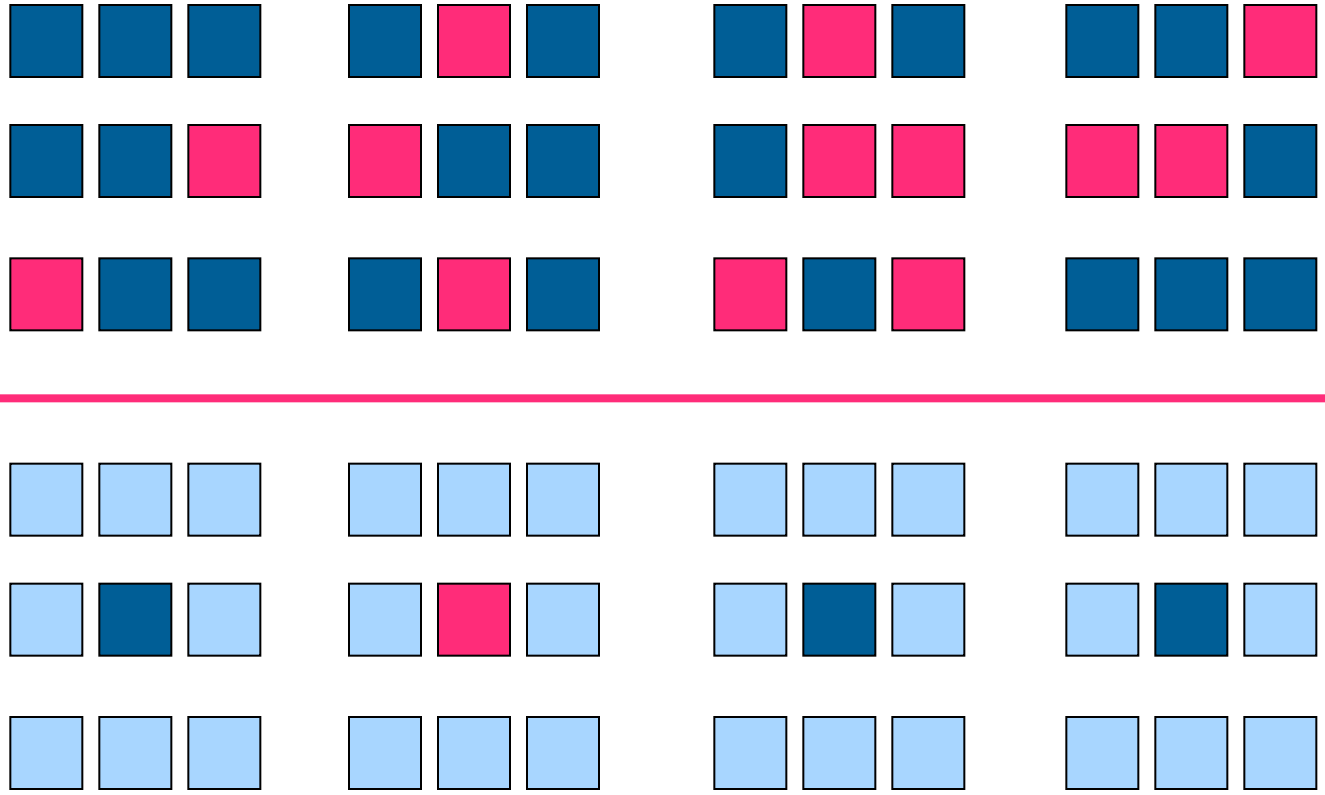
Game of Life



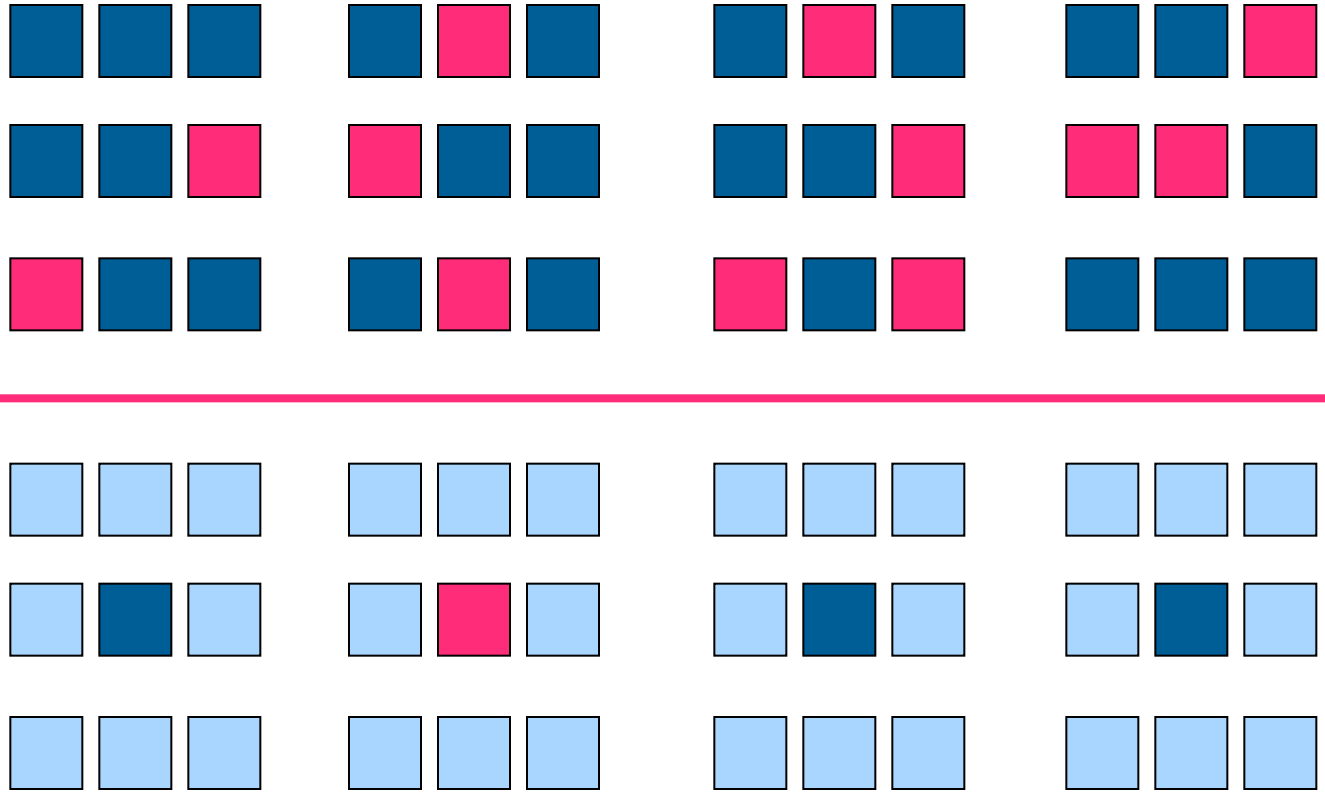
Game of Life



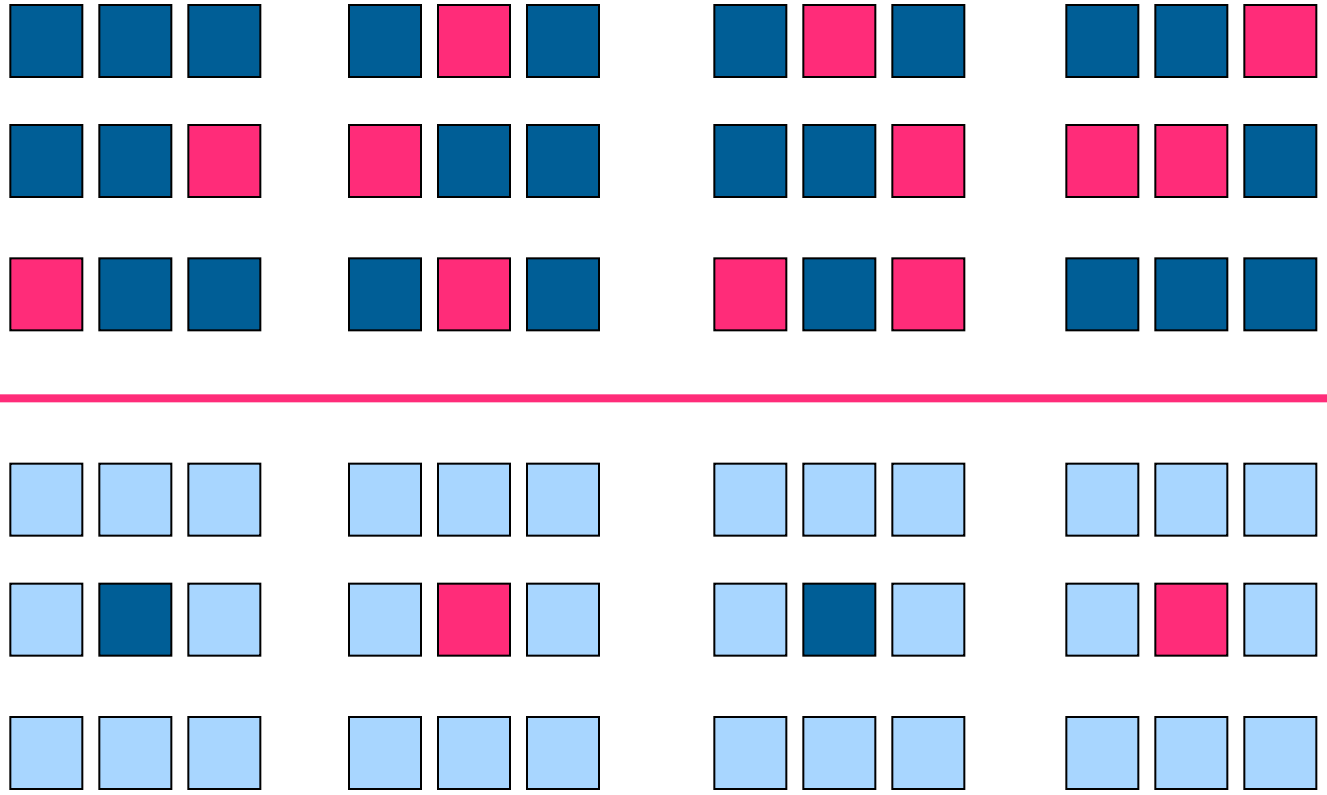
Game of Life



Game of Life



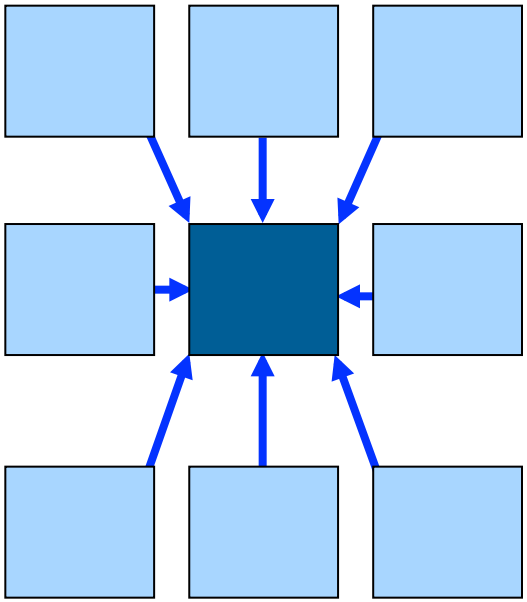
Game of Life

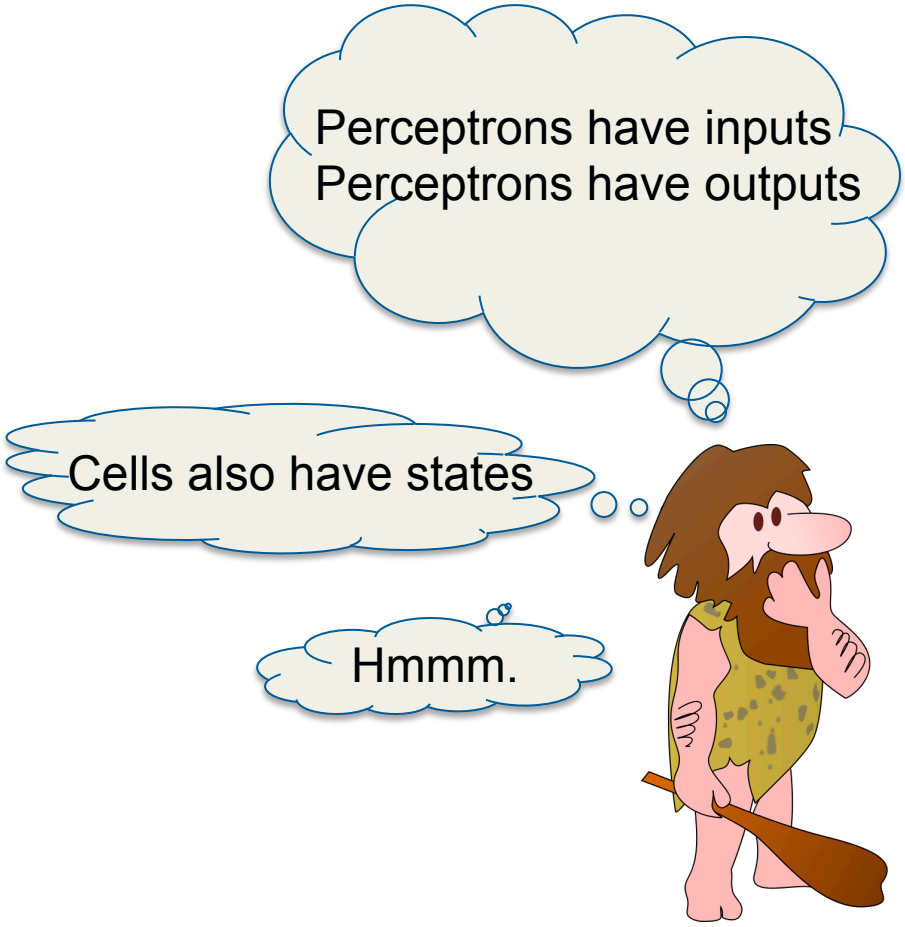


Interesting Patterns Emerge

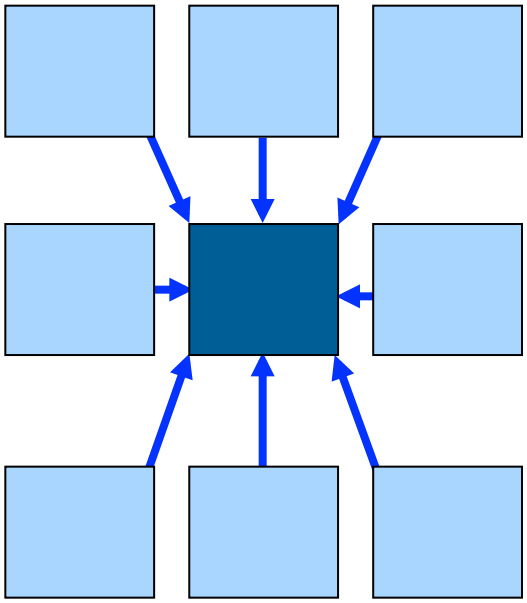
- Glider Guns --- patterns that repeat themselves but are displaced a few ‘pixels’. Thus, they appear to move.
- Space Ships --- another pattern that moves.
- Can perform gating logic.
- Can constitute a Universal Turing Machine.

Suppose a ‘cell’ was a perceptron?

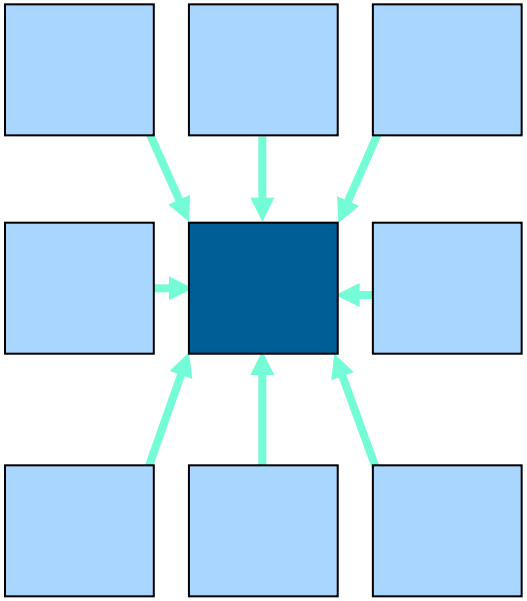




Suppose a ‘cell’ was a perceptron?



A: weighted
state values



B: weighted
output values

Neighborhoods

$$S_r(i, j) = \{C(k, l) | \max_{1 \le k \le M, 1 \le l \le N} \{|k - i|, |l - j|\} \le r\}$$

Cambridge University Press
0521652472 - Cellular Neural Networks and Visual Computing: Foundations and Applications
Leon O. Chua and Tamas Roska

Cell Equations of State

$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l)\in S_r(i,j)} A(i,j;k,l)y_{kl} + \sum_{C(k,l)\in S_r(i,j)} B(i,j;k,l)u_{kl} + z_{ij}$$

state

output

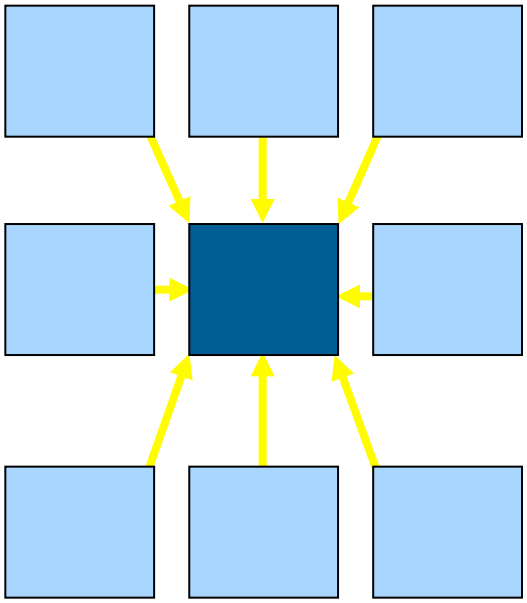
input

Bias/
Threshold

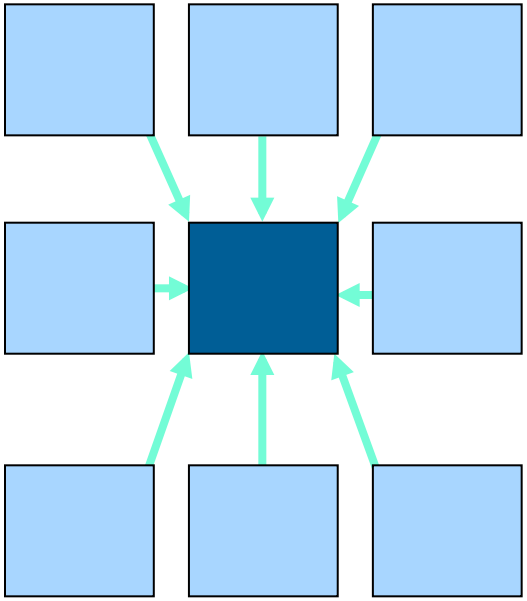
Cambridge University Press
0521652472 - Cellular Neural Networks and Visual Computing: Foundations and Applications
Leon O. Chua and Tamas Roska

How Many CNNs Possible?

A template



B template



If only 2 values are permissible for each link…2^{16} possible CNNs.

Let's Take a Look

http://www.isiweb.ee.ethz.ch/haenggi/CNN_web/CNNsim_adv.html