# Simulating Recurrent Neural Networks on GPUs

## Summer Project

Zhangsheng Lai

September 27, 2017

# Introduction

**Definition**
A *feedforward neural network* is an artificial neural network wherein connections between the units do not form a cycle.

**Definition**
A *recurrent neural network* is a class of artificial neural network where connections between units form a directed cycle.

# Feedforward Neural Network

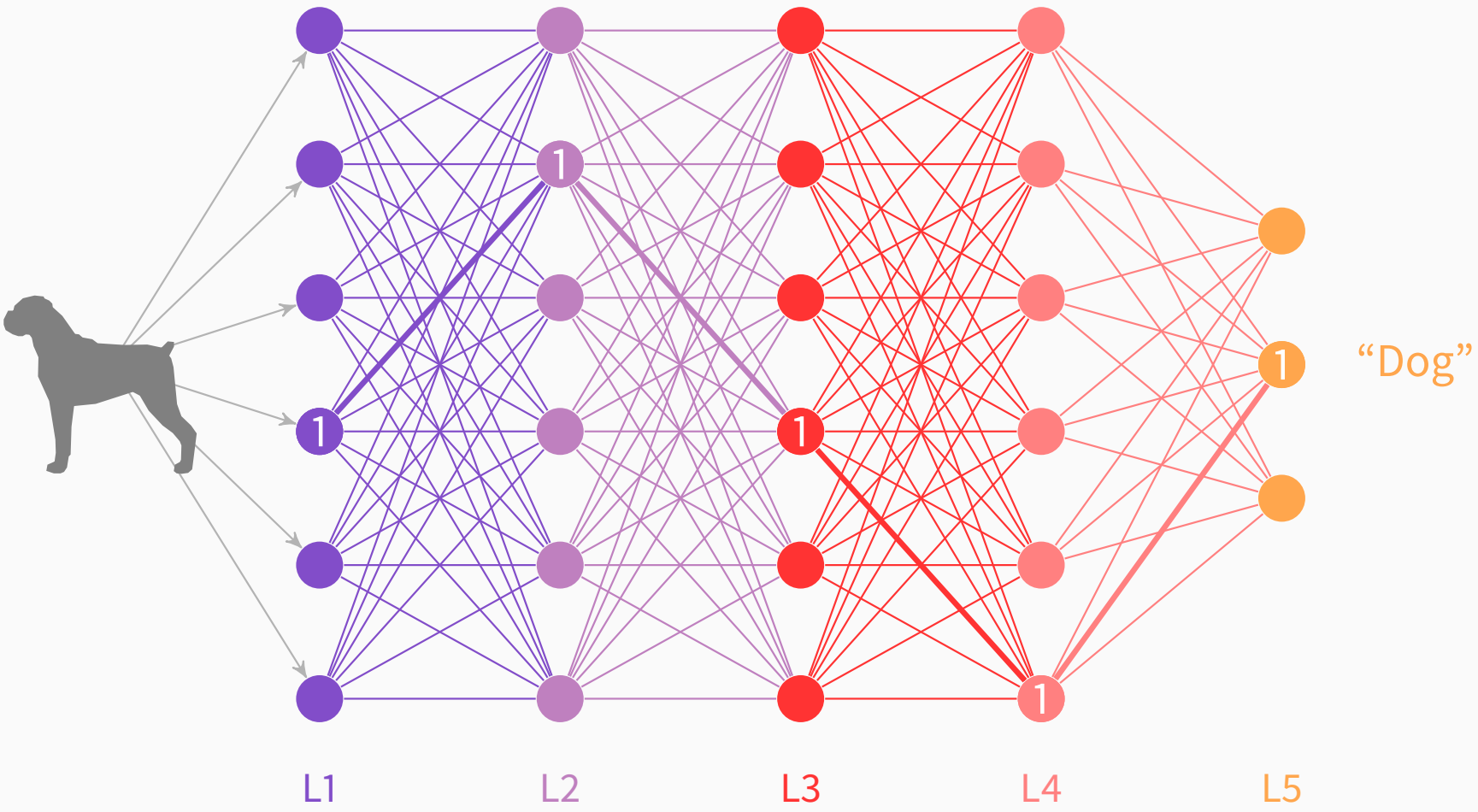
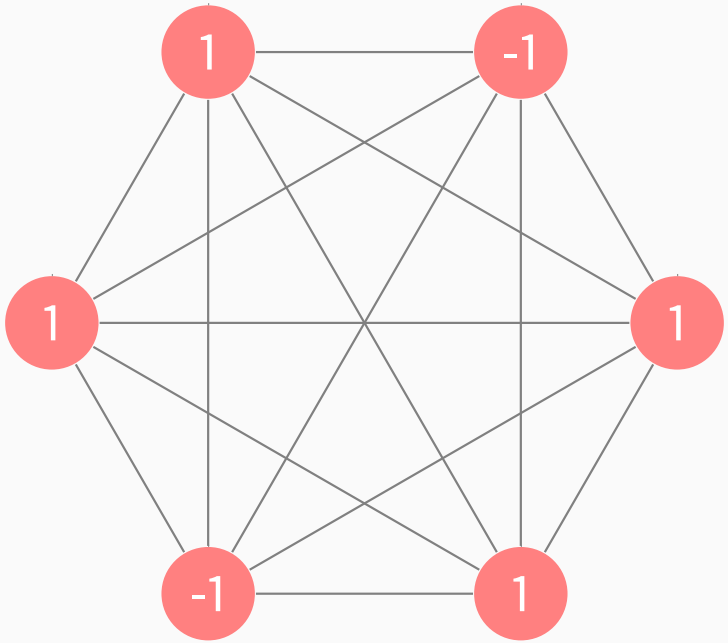



**Figure 1:** Feedforward Neural Network

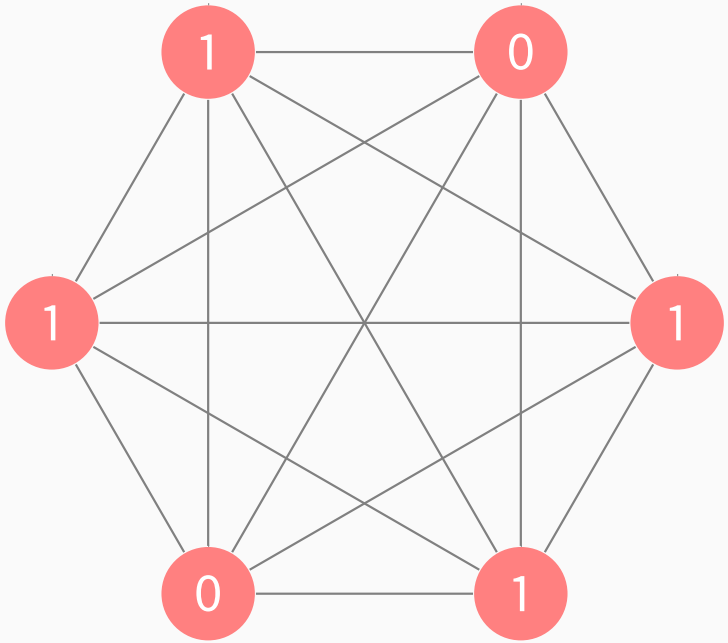
# Recurrent Neural Network

# Hopfield Networks and Boltzmann Machines

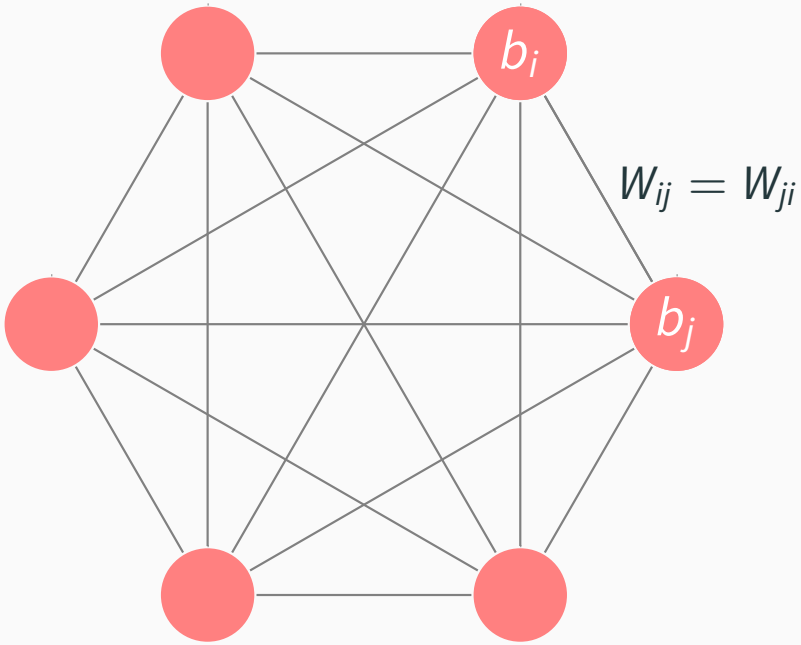
# Hopfield Networks



# Hopfield Networks



# Hopfield Networks

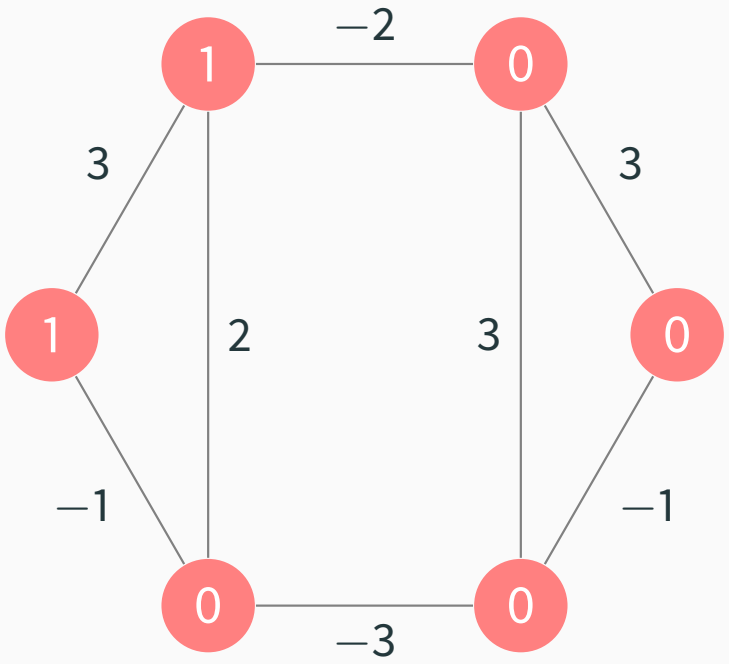



$$\text{Energy configuration, } E = -\sum_{i<j} W_{ij} x_i x_j - \sum_i b_i x_i$$

$$\text{Energy gap, } \Delta E_i = E(x_i = 0) - E(x_i = 1) = \sum_j W_{ij} x_j + b_i$$

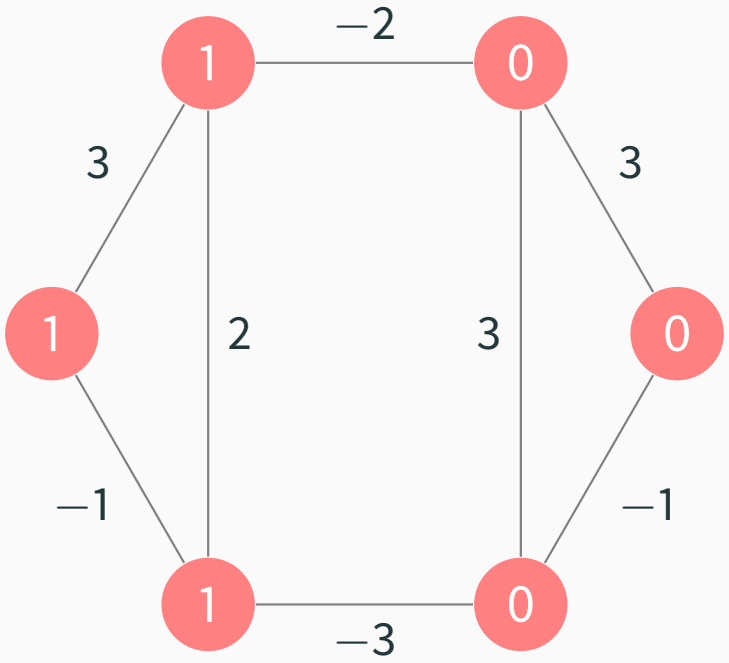
$$\text{Update rule, } x_i := \begin{cases} +1 & \sum_j W_{ij} x_j + b_i \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

# Hopfield Networks




(1, 0, 0, 0, 0, 1)

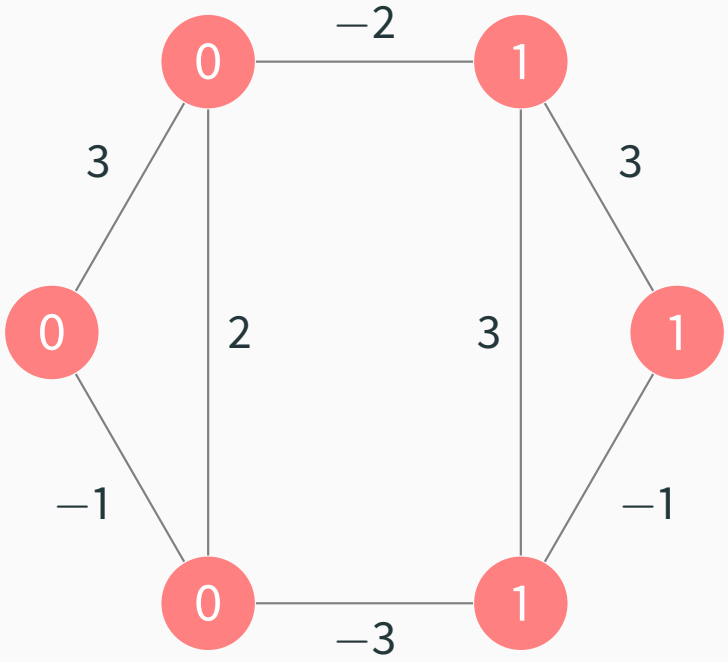
# Hopfield Networks




$(1, 0, 0, 0, 0, 1)$

$(1, 1, 0, 0, 0, 1)$

# Hopfield Networks

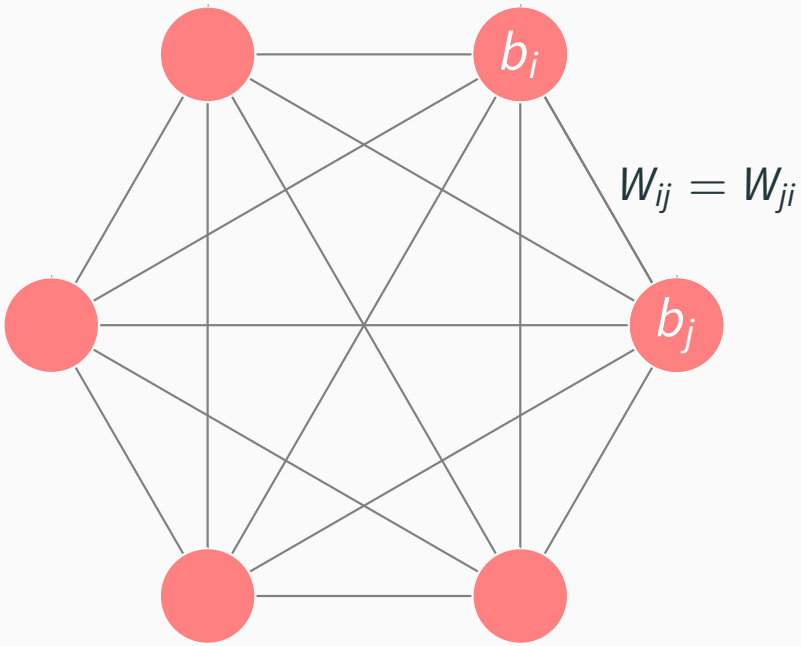



$(1, 0, 0, 0, 0, 1)$

$(1, 1, 0, 0, 0, 1)$

$(0, 0, 1, 1, 1, 0)$

# Boltzmann Machines




$$E = -\sum_{i<j} W_{ij} x_i x_j - \sum_i b_i x_i$$

$$\Delta E_i = E(x_i = 0) - E(x_i = 1) = \sum_j W_{ij} x_j + b_i$$

$$\mathbb{P}(x_i = 1) = \frac{1}{1 + e^{-\Delta E_i / \tau}}$$

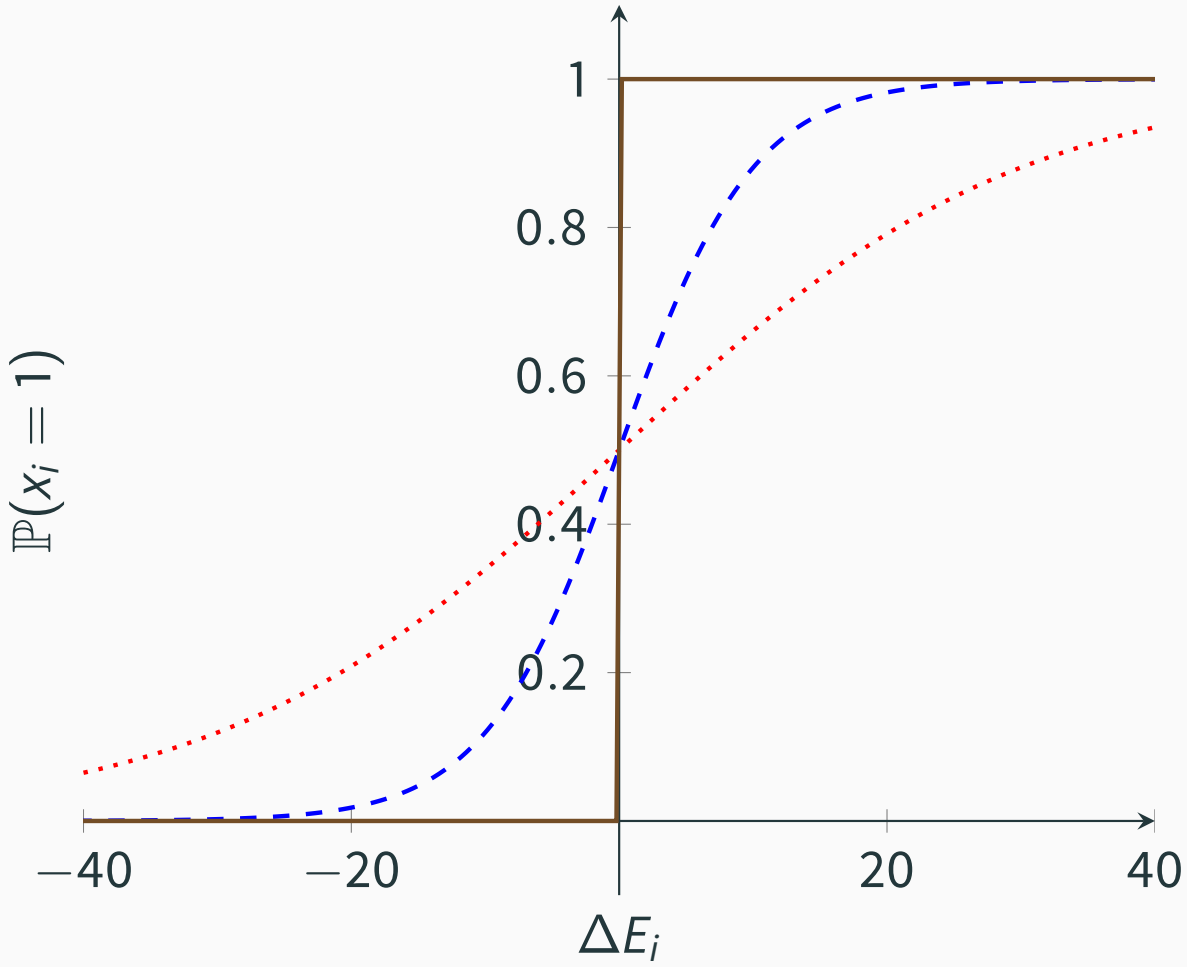
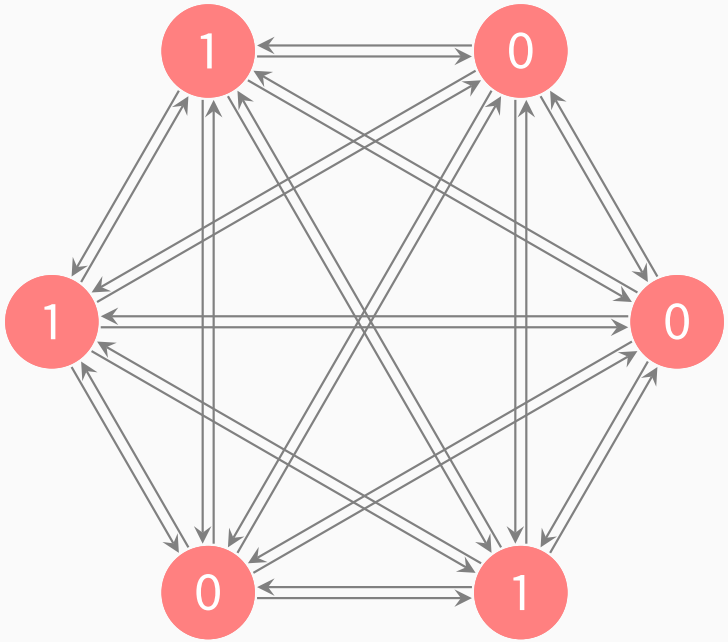


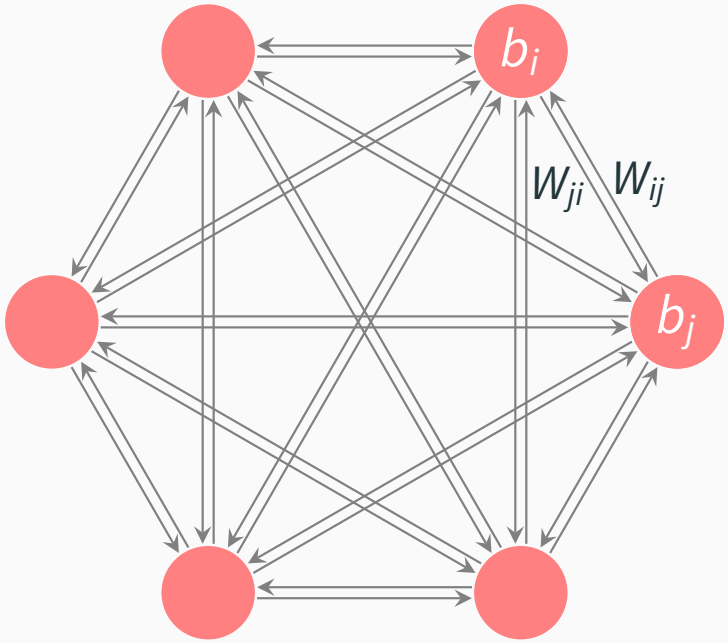

**Figure 2:** $\tau = 0$ (solid), $\tau = 5$ (dashed), $\tau = 15$ (dotted)

# McCulloch-Pitts Machines

# McCulloch-Pitts Machines



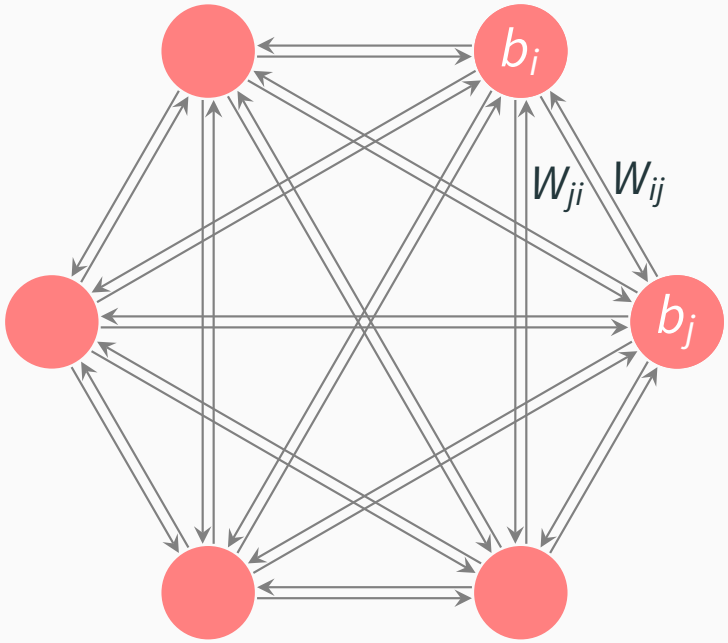
# McCulloch-Pitts Machines




$$\text{Transition Energy, } E(y, x|\theta) = -\sum_{ji \in E} W_{ji} y_j x_i - \sum_{j \in V} b_j s_j - \sum_{i \in V} b_i s_i$$

$$\Gamma_{yx} = \exp\left(-\frac{1}{2\tau} E(y, x|\theta) + \frac{1}{2\tau} E(x, x|\theta)\right)$$

# McCulloch-Pitts Machines

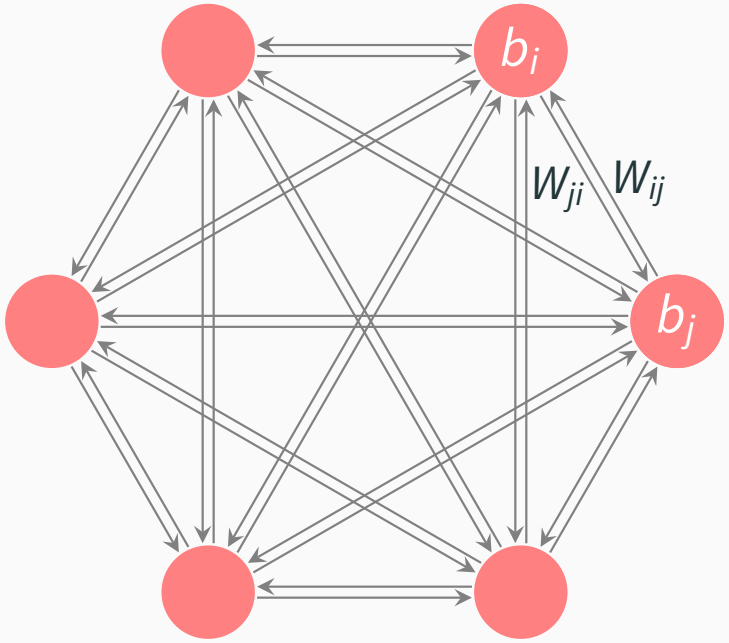



$$\text{Transition Energy, } E(y, x|\theta) = -\sum_{ji \in E} W_{ji} y_j x_i - \sum_{j \in V} b_j s_j - \sum_{i \in V} b_i s_i$$

$$\Gamma_{yx} := \exp\left(\frac{1}{2\tau} s_j z_j\right)$$

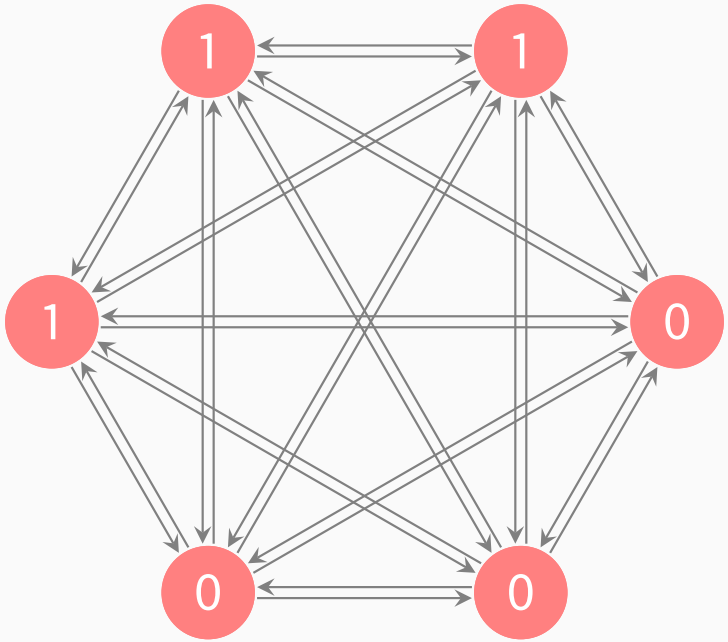
where $s_j = 1 - 2x_j$, $z_j = \sum_j W_{ji} x_i + b_j$ and $x, y$ differ by the $j$th unit.

# McCulloch-Pitts Machines

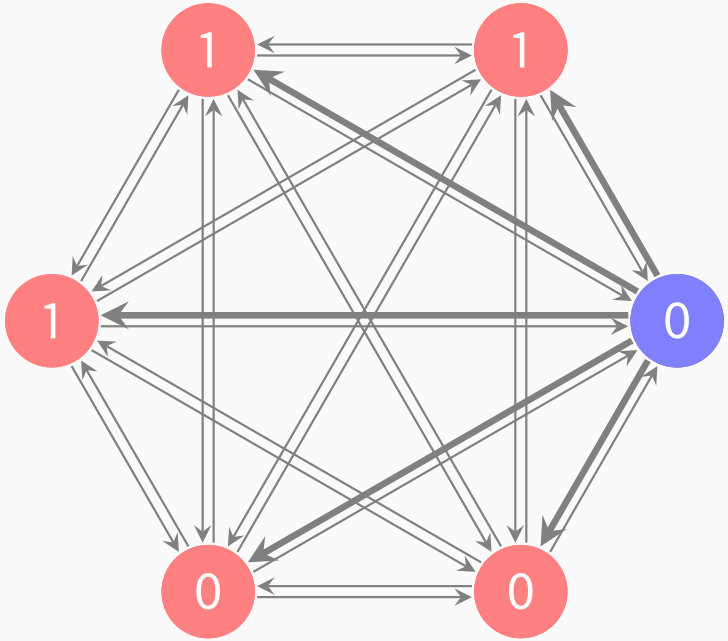



Transition probability from $x$ to $y$, $p_{yx} = \dfrac{\lambda_j}{\sum_{j'} \lambda_{j'}}$

# McCulloch-Pitts Machines

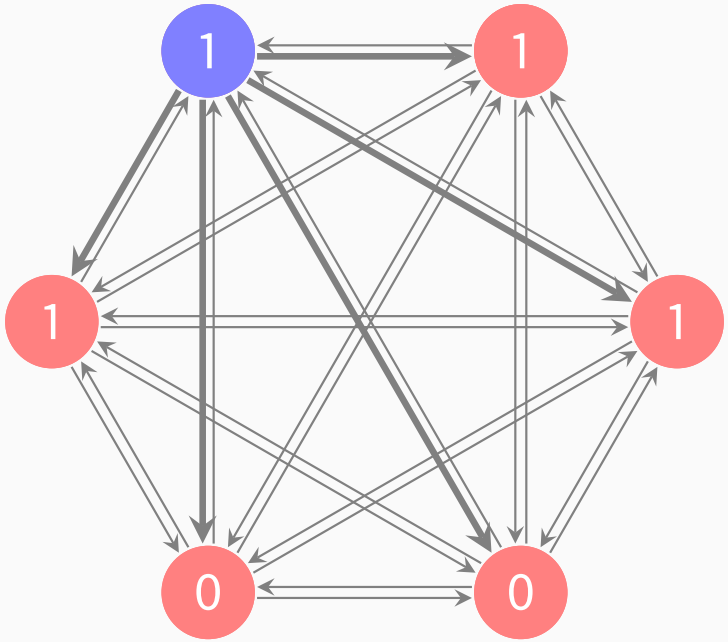


# McCulloch-Pitts Machines




$(T_0, (1, 0, 0, 0, 1, 1))$

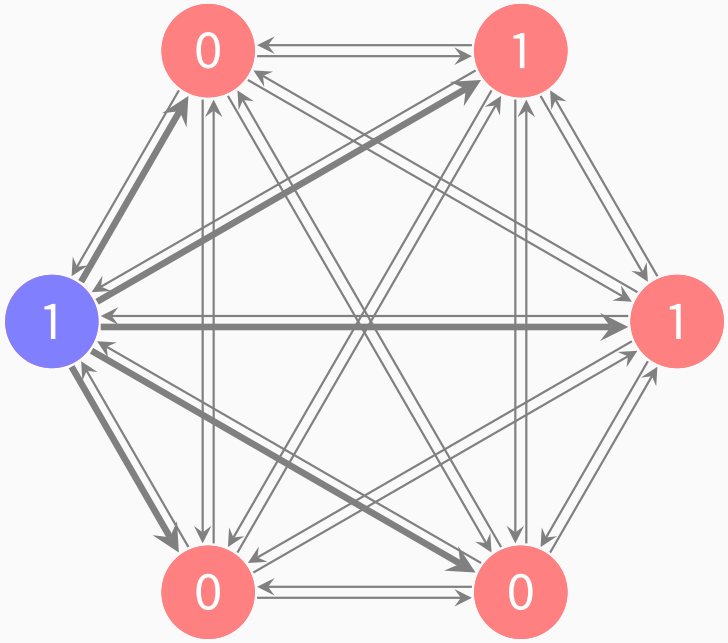
# McCulloch-Pitts Machines




$(T_0, (1, 0, 0, 0, 1, 1))$

$(T_1, (1, 0, 0, 1, 1, 1))$

# McCulloch-Pitts Machines




$$(T_0, (1, 0, 0, 0, 1, 1))$$

$$(T_1, (1, 0, 0, 1, 1, 1))$$

$$(T_2, (1, 0, 0, 1, 1, 0))$$

# McCulloch-Pitts Machines

# McCulloch-Pitts Machines

# Simulating on GPUs

# Simulating on GPUs

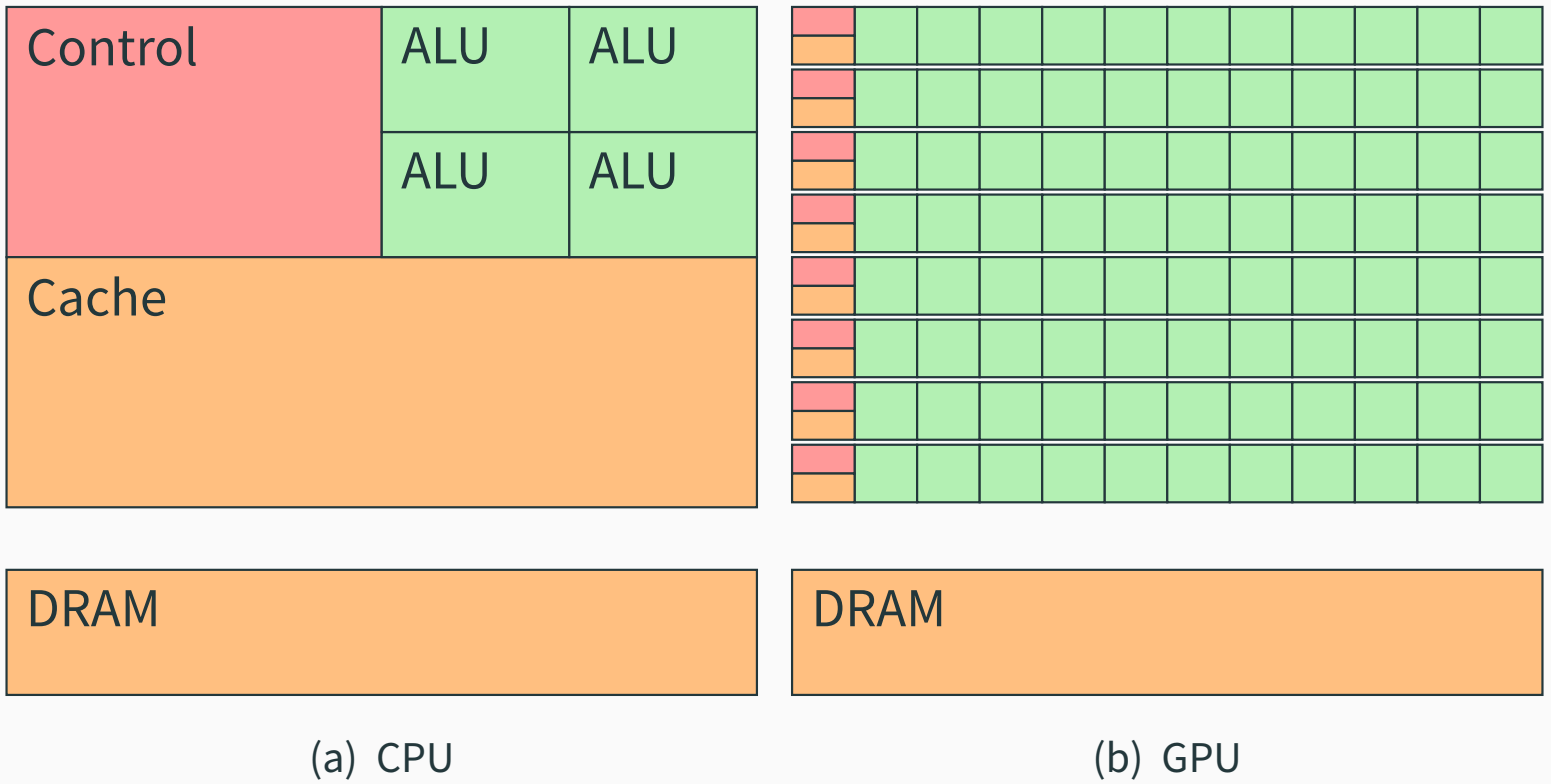



**Figure 3:** Comparison between the amount of transistors devoted to different functions inside a CPU and a GPU.

# Simulating on GPUs

# Simulating on GPUs

# References


J. Macor.

***A Brief Introduction to Type Theory and the Univalence Axiom***

http://math.uchicago.edu/ may/REU2015/REUPapers/Macor.pdf

The Univalent Foundations Program

***Homotopy Type Theory: Univalent Foundations of Mathematics*.**

https://homotopytypetheory.org/book

The n-Category Café

***From Set Theory to Type Theory***

https://golem.ph.utexas.edu/category/2013/01/from_set_theory_to_type_theory.html

The *n*Lab

***Function Type***

https://ncatlab.org/nlab/show/function+type