

Computation  
and the Brain  
2019



welcome  
to Week 5

Project

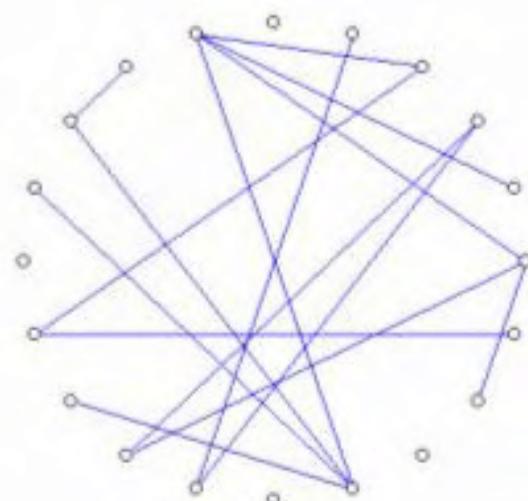


# Erdős – Renyi graphs



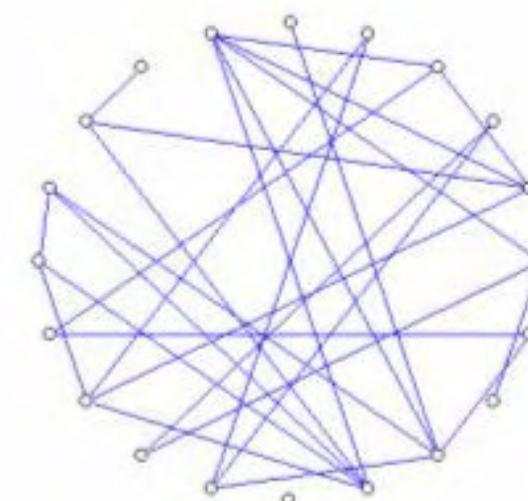
$p = 0$

(a)



$p = 0.1$

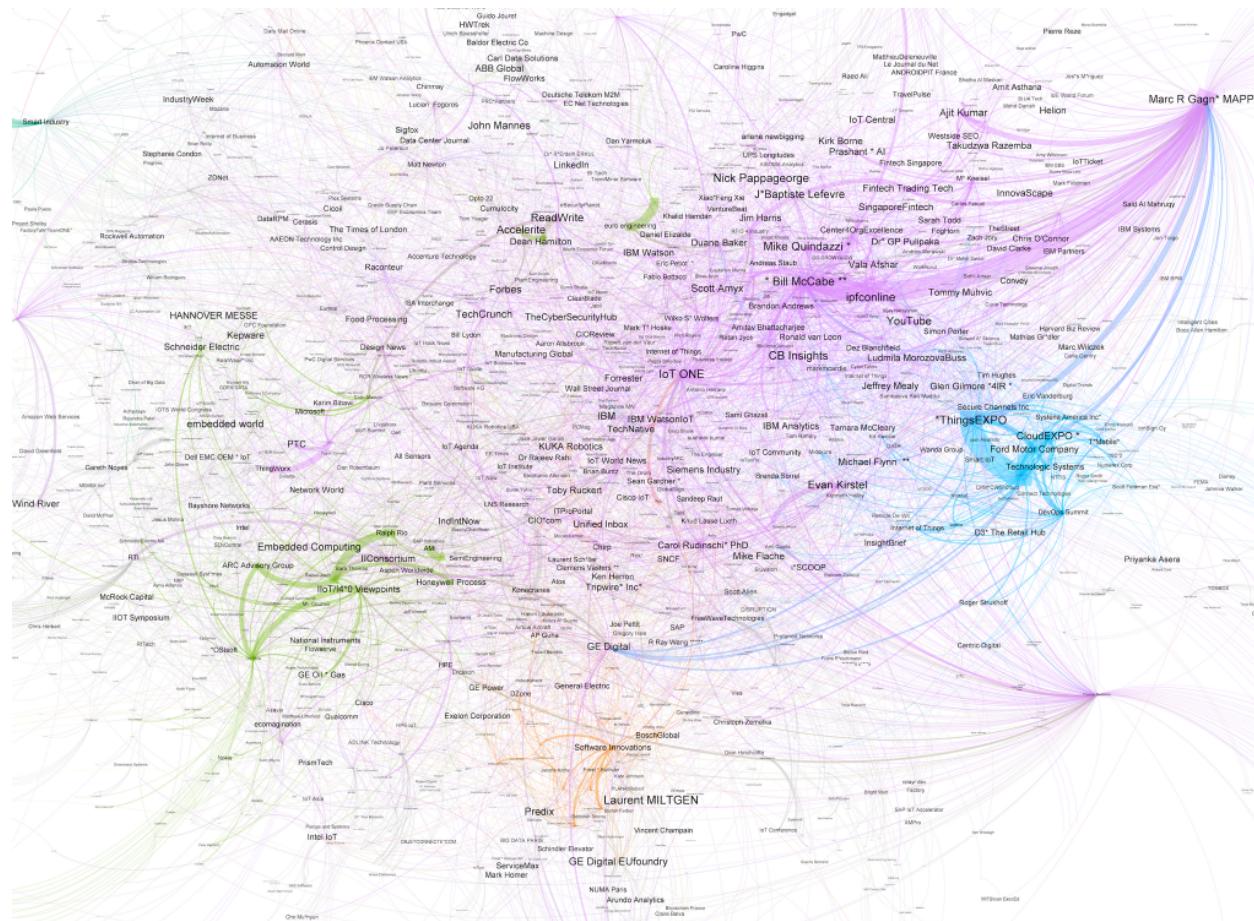
(b)



$p = 0.2$

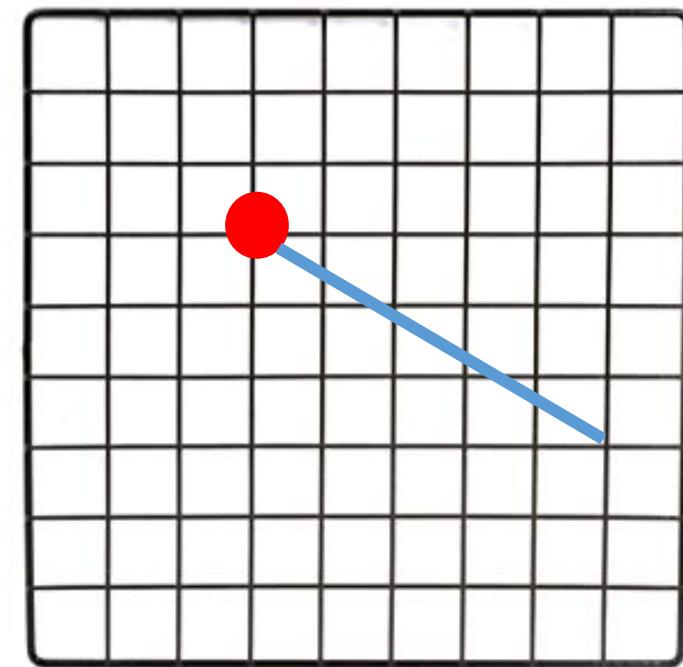
(c)

# Power law / Internet like graph



# The small world graph [Kleinberg 2000]

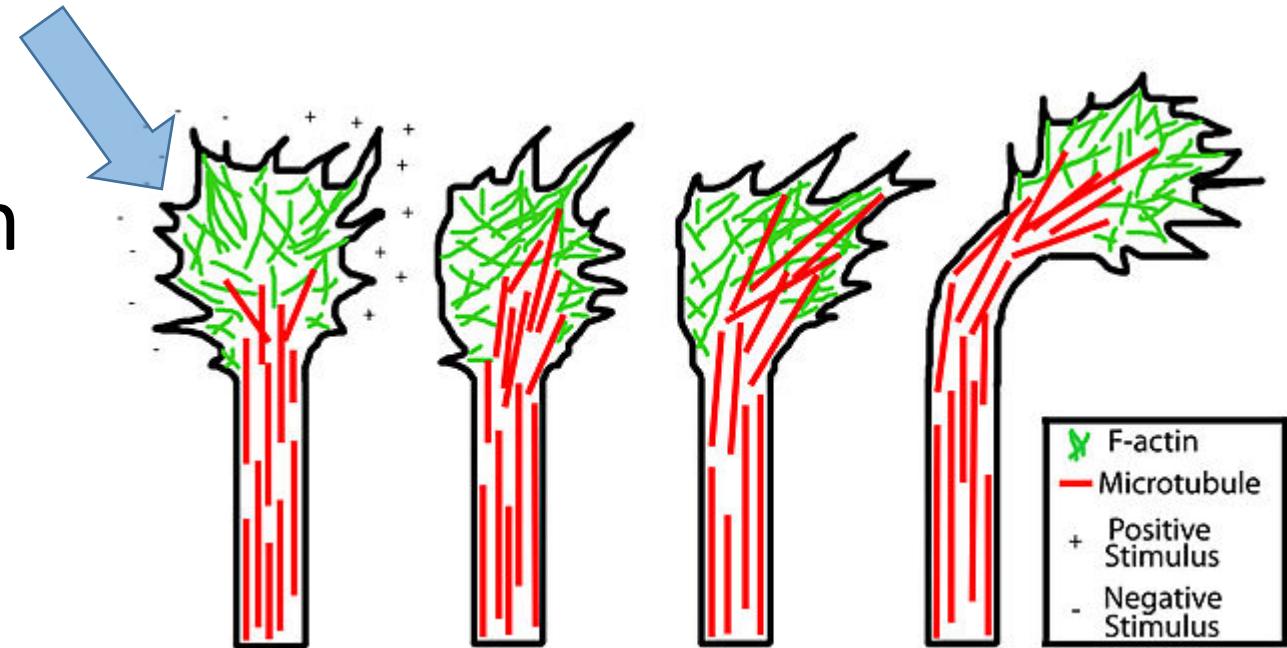
- A grid (**2D geometry!**)
- Plus from each **node** very few random **edges**
- Going distance  $d$  away with probability  $\sim d^{-2}$
- **Theorem:** Greedy algorithm routes in very few steps



# What does the axon grow?

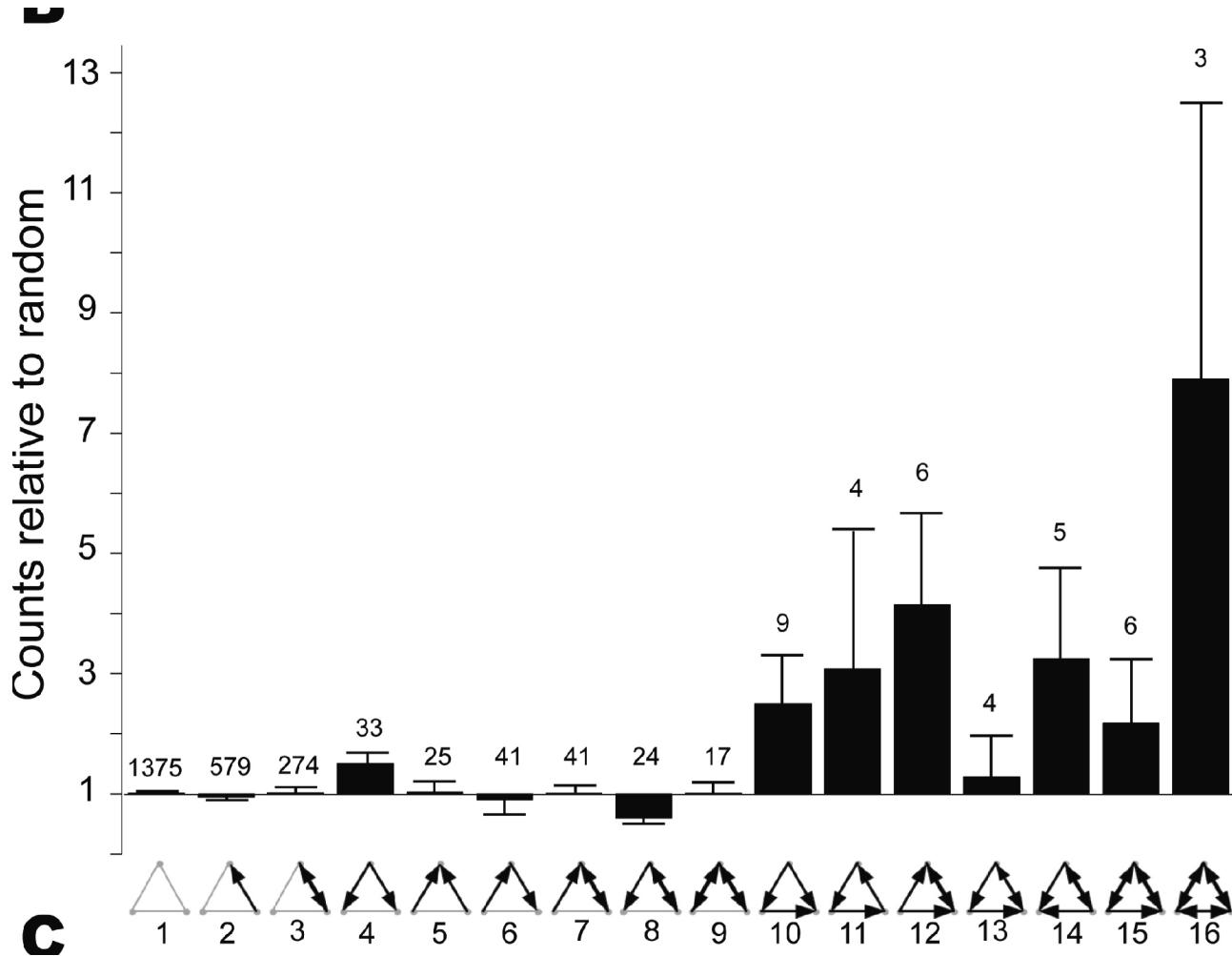
## The axon's **growth cone**

- A sensory – motor organ
- Navigates the brain following chemical cues  
**(advance, turn, stop, *split*)**



# So, what kind of graph is the connectome?⚙️

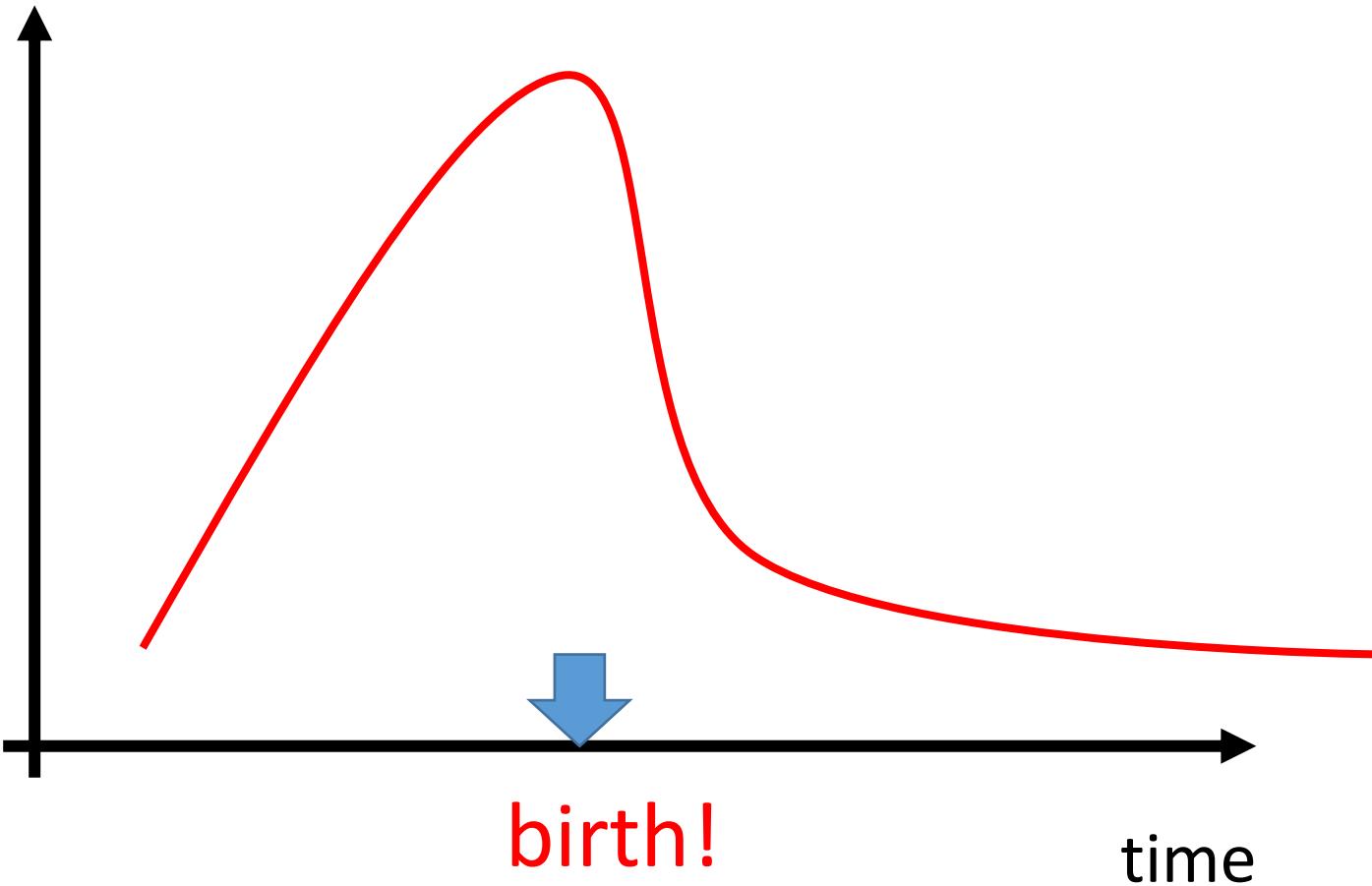
## Three neuron connectivity



**“Highly Nonrandom Features of Synaptic Connectivity in Local Cortical Circuits”**  
S. Song et al., PLOS Bio  
2005

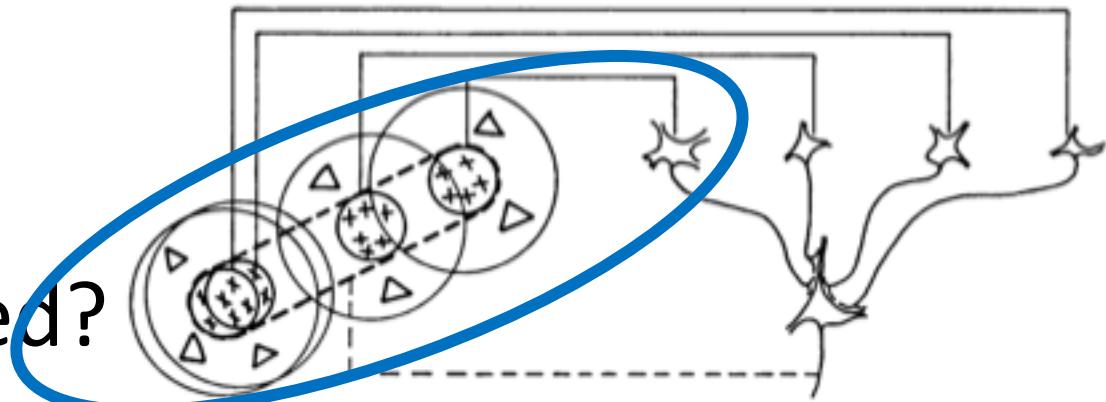
# Number of synapses

## Sparsification algorithm?⚙️



Incidentally, remember Hubel and Wiesel's simple cells?

- How are these synapses formed?
- How do all **these ganglia** know that they are on a straight line in the retina? 🚧
- Was it evolution?
- Is it done during development?
- Or is it learning and synapse deletion?



Btw: yet another thing to consider⚙️

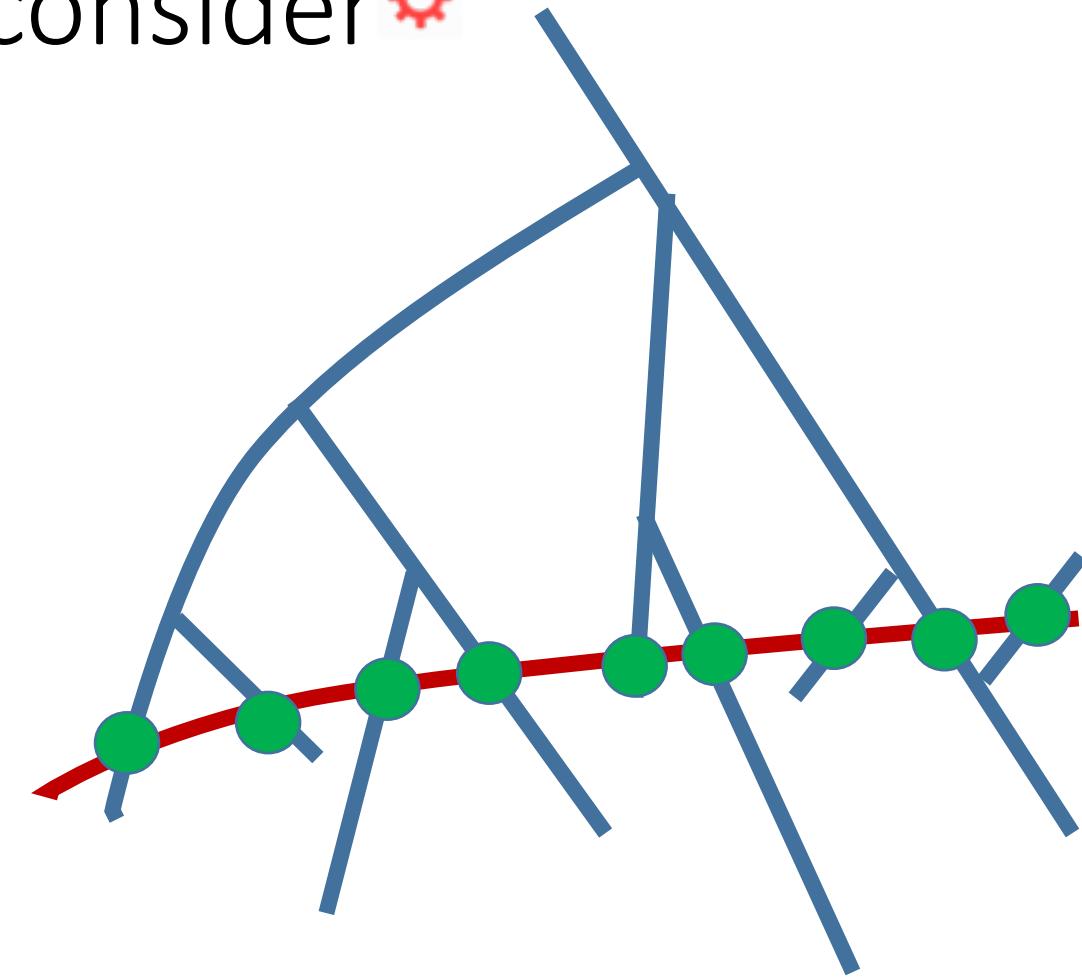
An **axon** can touch a postsynaptic **dendrite** at *many* places

How do these add up?

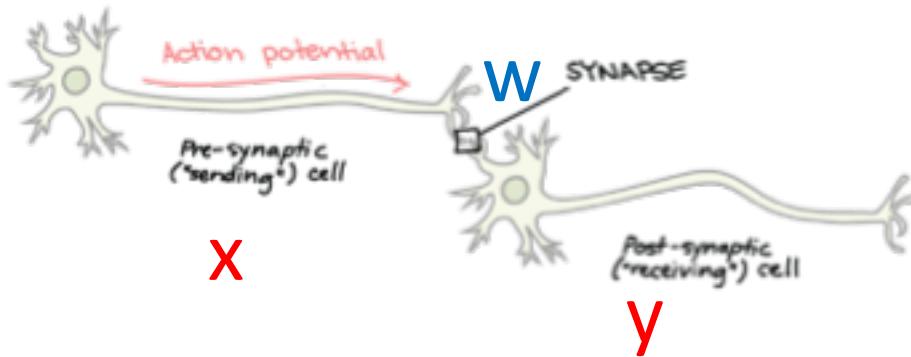
*Sum of products?*

*Products of sums? Other?*

Some say, **that** is where the brain computes...



# Models of plasticity



Hebb:  $\Delta w \sim x y$

Vector Hebb:  $\Delta W \sim X y$  ( $X$  vector of presynaptic neurons)

Covariance form of Hebb (assuming  $y$  is the average input times  $W$ ):  $\Delta W \sim \text{cov}[X] W$

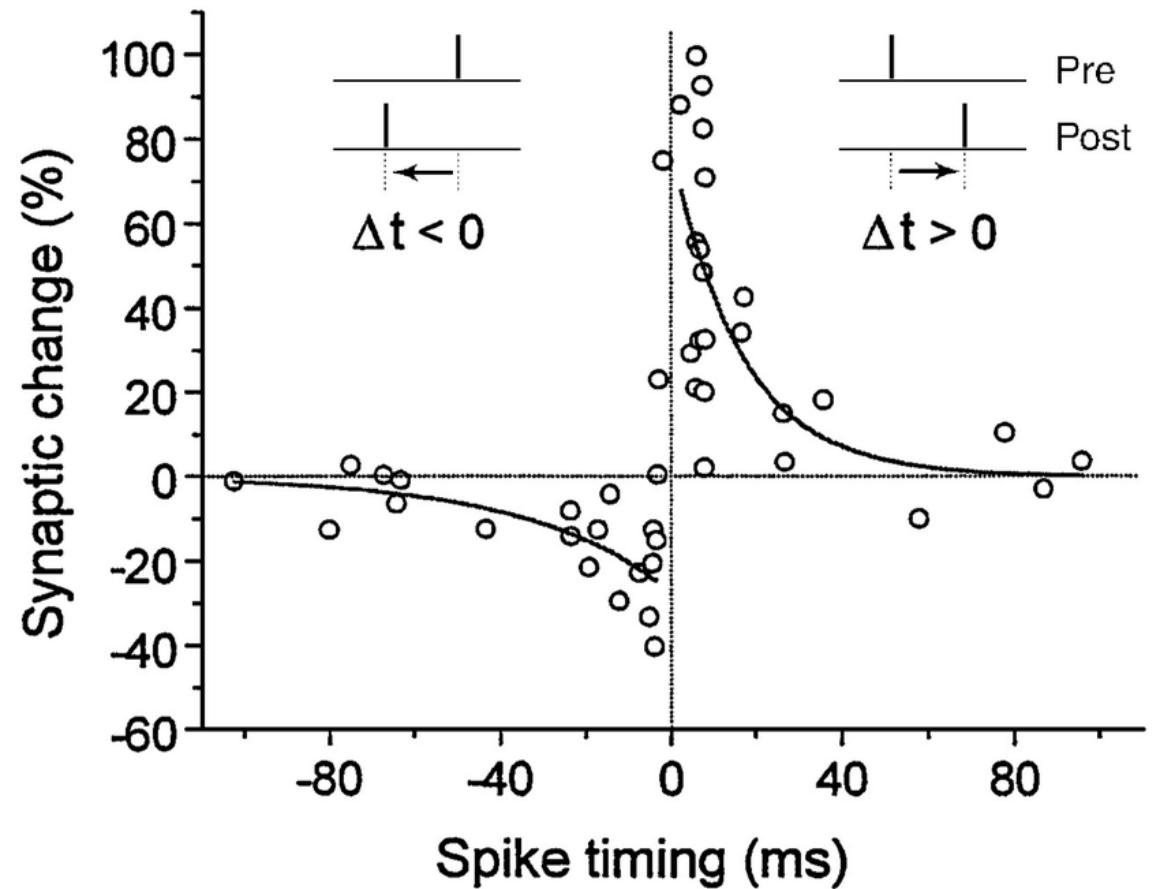
Hebb with LTD:  $\Delta W \sim X (y - \vartheta)$  – but then LTD if  $y = 0$

BCM rule corrects this:  $\Delta W \sim X y (y - \vartheta)$

Homeostasis, normalization and competition: at a slower time scale, the sum of all presynaptic weights is renormalized to 1

# Spike timing-dependent plasticity (STDP)

If spike arrives in time, some gain. Just in time, big gain.  
If it just misses it, some loss.  
Just misses it, big loss.



- Biologically, what causes depolarization?
- Are there **spiking neuron models** that do not make the assumptions that the H-H model makes? How would the computation change if we did not make the assumptions?
- What makes **membrane potential vary** within a single neuron exactly?
- Where does the **higher resistance near the soma** that Mel was talking about come from?
- Has someone tried ANNs where each unit has **two layers** as described by Mel? Why would it work any better?
- What other research is being done on the difference between our ANN models and the biological NN? It seems pretty bad that we are **using only one type of neuron to try to perform many different systems** when biologically, there are many different types of neurons used.
- **What did Hodgkin and Huxley overlook** in their model of the neuron? Other than fully simulating the neuron, what other **models exist for the neuron**? Are they feasible to use in a NN?
- How well does the space given for neuron models (**morphology vs dynamics**) translate to AI systems? Are there better axes?

- Very few neurons are **ready to fire several milliseconds after the spike**. Why is this?
- Are there examples in computational neuroscience in which **H&H model of the neuron is actually used**, perhaps for a smaller ANN at least?
- Despite many neuron models, **ANNs seem to use the most basic ones**, put into large networks. Is this **choice only for convenience**, do these basic neuron models perform just as good, and would using more complex neuron models in ANNs provide better understanding of the network's computation?
- The brain is known to have 100s of types of neurons. **Which ones does the HH model apply to?**
- Due to how short the pulse of a neuron is, followed by the refractory period, is it **possible for a neuron to be used for more than one process simultaneously?**
- **Where does one draw the line** between an accurate simulation, say Hodgkin and Huxley equations, and a biologically inspired abstraction, say a perceptron?
- Is there any particular need for a firing rate? **What does firing rate encode?**
- Do you think the **firing rate or spiking models** is more convincing? Why?
- **Biologically plausible model** in which a rich human-like language can be shown to emerge

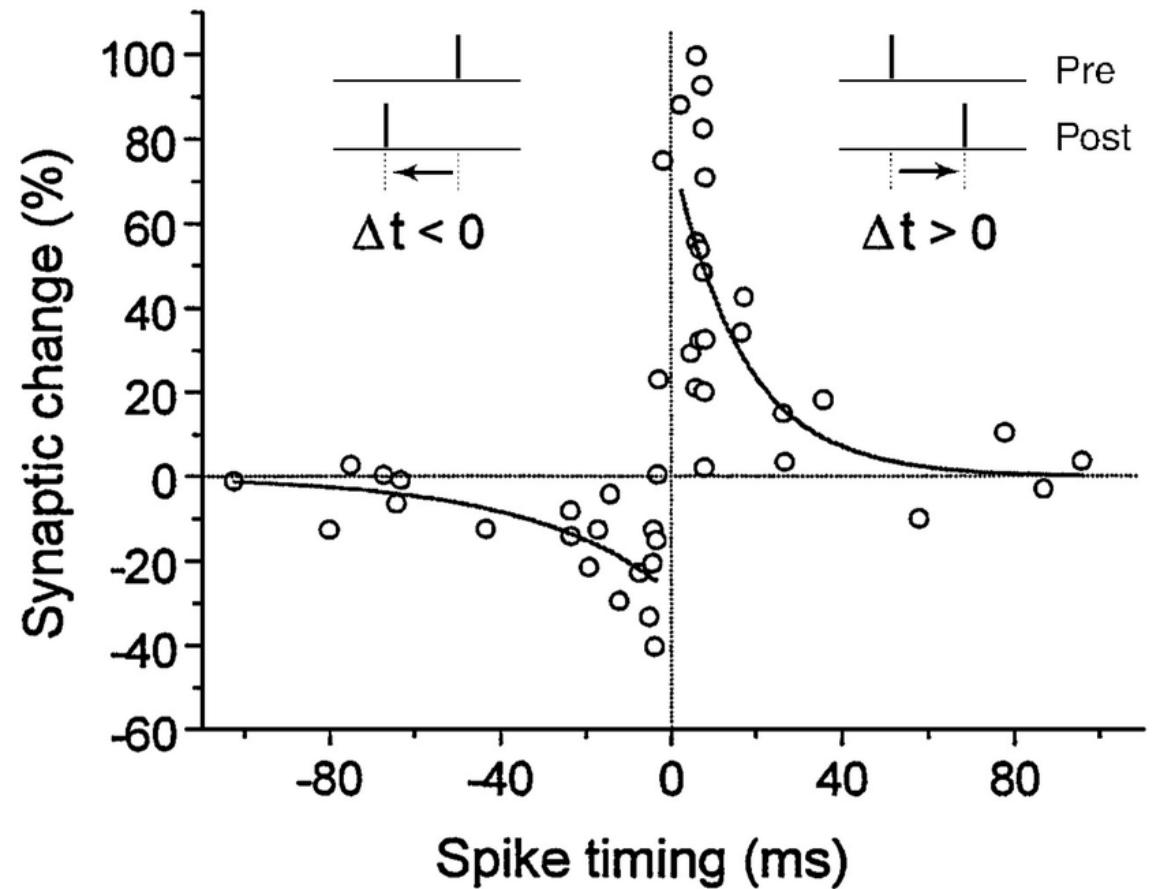
- Has there been any progress on dynamically complex artificial neural networks or networks with spike coding?
- Is there a sense that **networks with dynamics governed by ODEs** or using timing instead of rate information would **perform better**?
- How are H-H coefficients fit to real data? Non-linear least squares? Where do the numbers for  $\alpha$  and  $\beta$  come from?
- How would we use the Hodgkin Huxley model on something like the **Purkinje cell** or something like a **pyramidal neuron**?
- Can we incorporate dynamics in neuron networks, i.e., can we incorporate some continuous process in the neuron network? Like this paper “Neural Ordinary Differential Equations”.
- How can we simplify the dynamic? I.e., can we use some simple function, like sigmoid or Relu, to simulate the neuron activation?
- In Barlett Mel’s talk, he mentions how most neuron models don’t include the dendrite because historically it was difficult to measure and over time people forgot about modeling the dendrites. Given current advances in understanding dendrites, **would neural networks be significantly improved if we added dendritic components to them** or should we rely on computation power/complexity instead?

# Today

- Finish plasticity discussion
- Biologically plausible deep nets? 
- Dynamical systems (aka ODEs)

# Spike timing-dependent plasticity (STDP)

If spike arrives in time, some gain. Just in time, big gain.  
If it just misses it, some loss.  
Just misses it, big loss.



Btw: Hebb's exact words

Rhyming soundbite: “fire together, wire together”

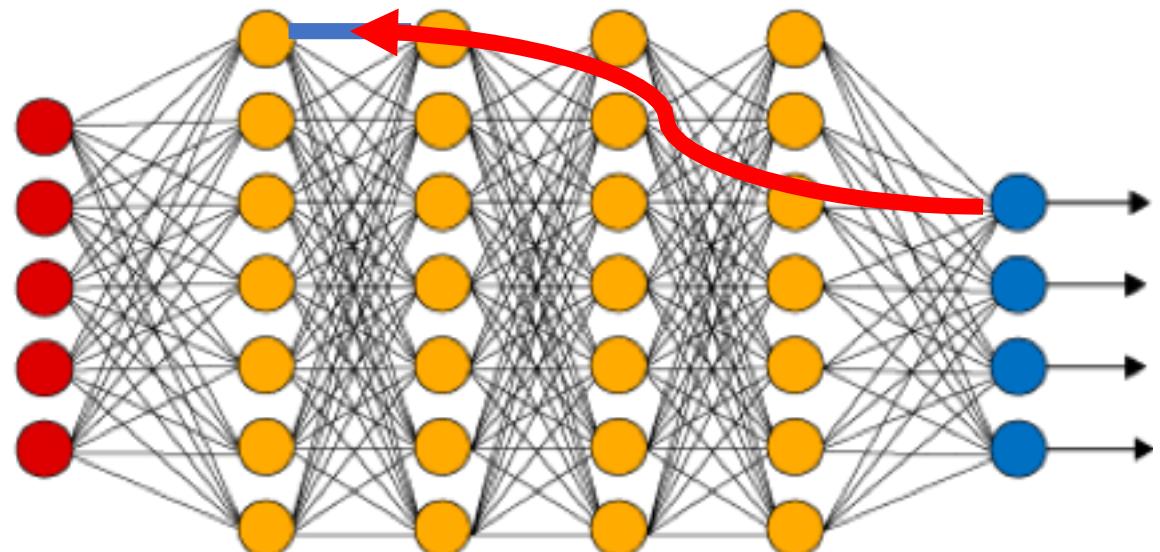
Actual quote from [Hebb 1949 *The Organization of Behaviour*]:

“When an axon of cell A is near enough to excite cell B or repeatedly or persistently **takes part in firing it**, some growth process or metabolic change takes place in one **or both cells** such that A’s efficiency, as one of the cells firing B, is increased.”

NB: “**or both cells**” is an impossibility

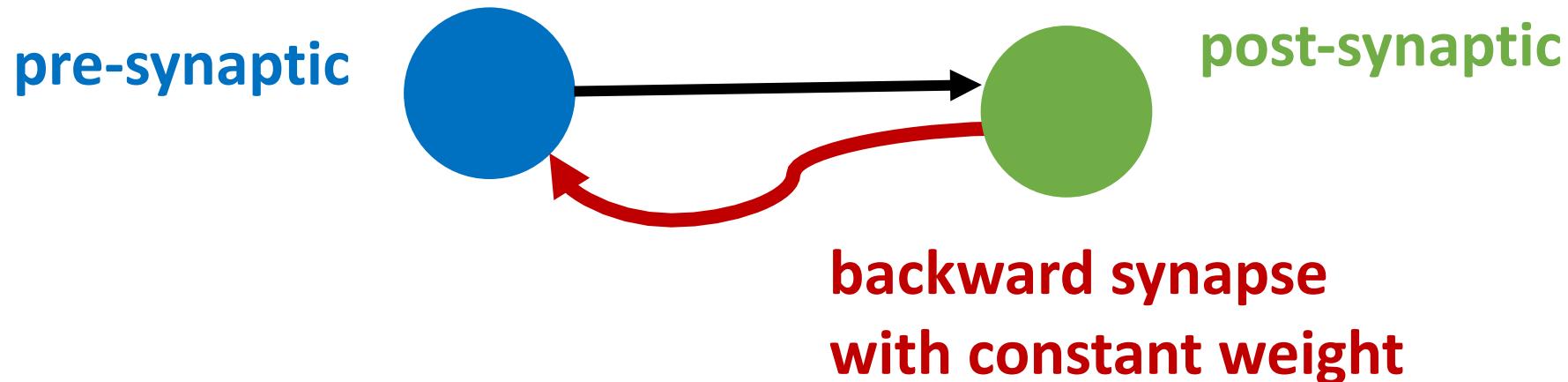
# Plasticity in the Brain vs in ANNs

- The Brain apparently learns through the plasticity of the synapses
- DNNs learn through back propagation, a kind of synaptic plasticity
- ***Big*** difference: in back propagation you need information ***from downstream activity***



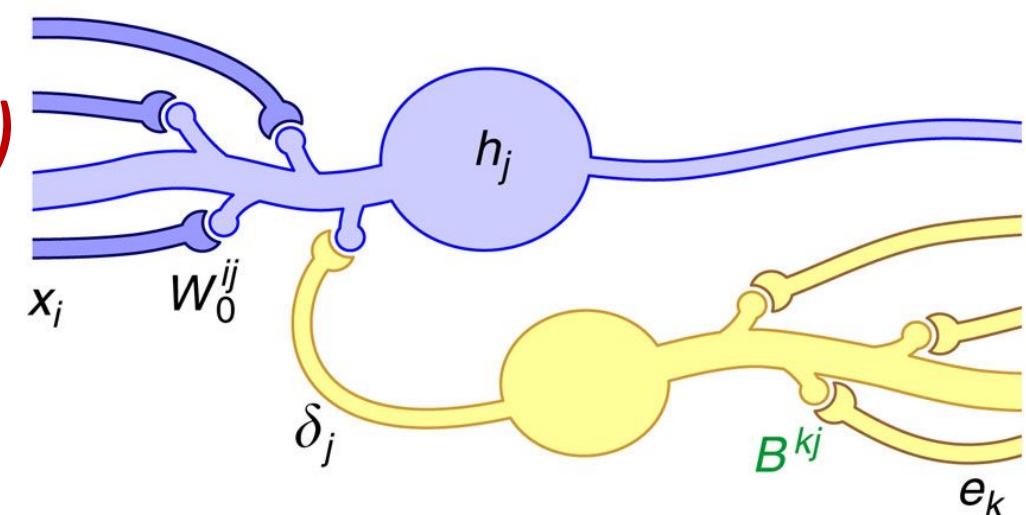
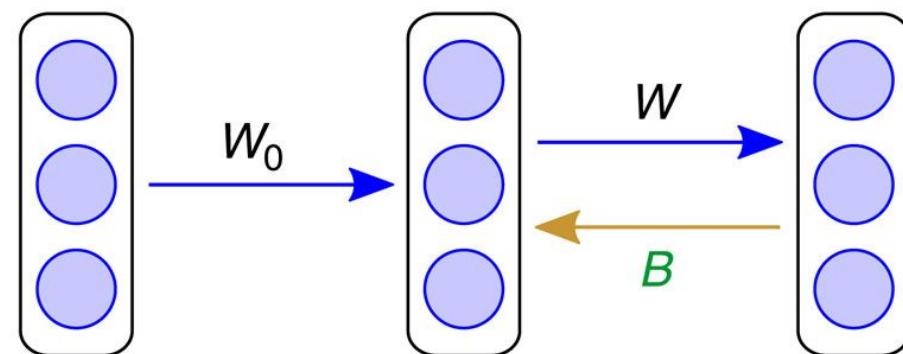
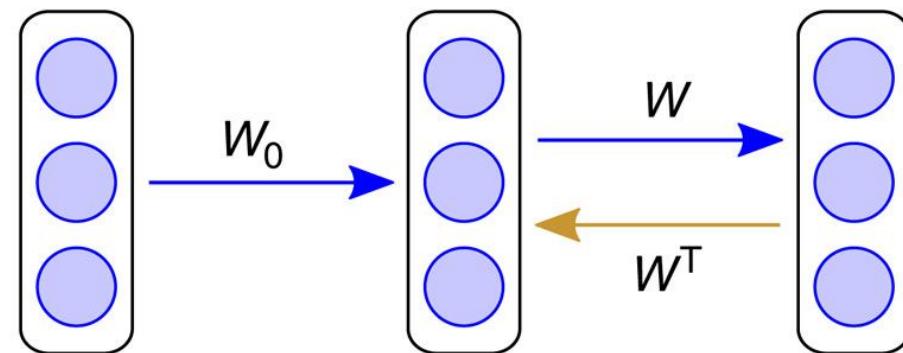
# Biologically plausible ANNs

- [Lillicrap et al. 1914]: constant (non-plastic) random synaptic weights in a **backwards** synapse suffice for some learning!



# Biologically plausible ANNs

[Lillicrap et al. 1914]:  
constant (non-plastic)  
random synaptic weights  
in a **backwards** synapse  
suffice for some learning!  
*(and this is biologically plausible)*



# Bengio et al. 2016 “Towards biologically plausible deep learning”

- Gradient descent:  $\Delta x^t = \alpha \delta(t) \nabla f(x^t)$  update happens at time t
- SDTP:  $\Delta w^t = \beta \delta(t) \nabla V(w^t)$  t is the time the spike arrives at the synapse
- Some similarity, huh?
- Idea: What if we use an STDP feedforward net to optimize some objective function whose “local derivative” is  $\nabla V$ ?
- This idea is pursued in the paper; some learning can be done, but there are catches and different kinds of biological implausibility...

# Biologically Plausible ANNs

- Several other versions of these ideas
- For another example, ***signSGD***: only use the *sign* of backpropagated **derivative**
- *(but is it biologically plausible? Naaah...)*
- Also [Hopfield et al 2019 PNAS] “*Learning through plasticity and competition between neurons*”

# Biologically Plausible ANNs: Some new ideas

- **Dopaminergic NNs**, see Yagishita et al, “A critical time window for dopamine actions on the structural plasticity of dendritic spines,” *Science* 2014.
- What if **every link of the ANN that fired** is increased by, say  $\alpha (1/4 - \text{error}^2)$
- Initial experiments: competitive with signSGD

Btw: the brain has evolved...

- “*Nothing in life can be understood except in the light of evolution*” Theodosius Dobzhansky
- Evolution is a frequently mentioned consideration in brain research
- (esp. cf: language...)
- The existential question:  
**“*but can brain circuits evolve?*”**

# Biologically Plausible ANNs:

## Nnevolution: The model

- $G$  genes
- Each gene has two alleles, 0-1, iid binomial  $p_i$
- Every genotype is a bitstring in  $\{0, 1\}^G$
- The weight of each link  $L$  is a sparse linear function of the genes
- $\sum a_{Lj} x_j$  where each  $a_{Lj}$  is 0 with probability  $1 - \delta$  and otherwise random in  $[-1, +1]$

# Biologically Plausible ANNs: NNevolution, the experiment

Repeat for T generations:

- Generate P genotypes and the corresponding ANNs
- Evaluate each on a minibatch
- Find the *fitness*  $f(A)$  of each allele A (say, 1/4 minus the average square error over all genotypes that have it)
- Update the gene probabilities  $p_A \rightarrow \sim p_A (1 + \varepsilon f(A))$

Experiments: competitive with signSGD

*Next: Dynamical Systems  
(aka Ordinary Differential Equations, ODEs)*

NONLINEAR  
*With Applications to Physics,  
Biology, Chemistry, and Engineering*  
DYNAMICS  
AND CHAOS



 CRC Press  
Taylor & Francis Group

A CHAPMAN & HALL BOOK

Steven H. Strogatz  
SECOND EDITION

*Excellent and  
accessible book!*

# Dynamical Systems

- Solve  $\dot{x}(t) = f(x(t))$ , given the value of  $x$  at  $t = 0$
- $x(t)$  is an unknown function of time  $t$ , usually a vector function;  $\dot{x}$  denotes  $dx/dt$
- Linear dynamical system:  $\dot{x} = A x$
- One dimension, solution:  $x(t) = x(0) e^{At}$
- True for any number of dimensions...
- Linear systems are useful only as local approximations for solving nonlinear systems (helps, sometimes)

# The dawn of dynamical systems: The two-body problem [Newton 1687]

- E.g., the earth and the moon (ignoring all else)

$$\begin{aligned} F(x,y) &= M\ddot{x} \\ -F(x,y) &= m\ddot{y} \end{aligned}$$

- (Second derivatives simulated by an extra equation)
- Two body problem can be solved easily
- Add: the center of mass moves with constant velocity (so, assume it doesn't move)
- Subtract: the vector of the two bodies moves on a plane
- Etc.

# The Three-Body Problem? sun – earth – moon [Euler 1770]

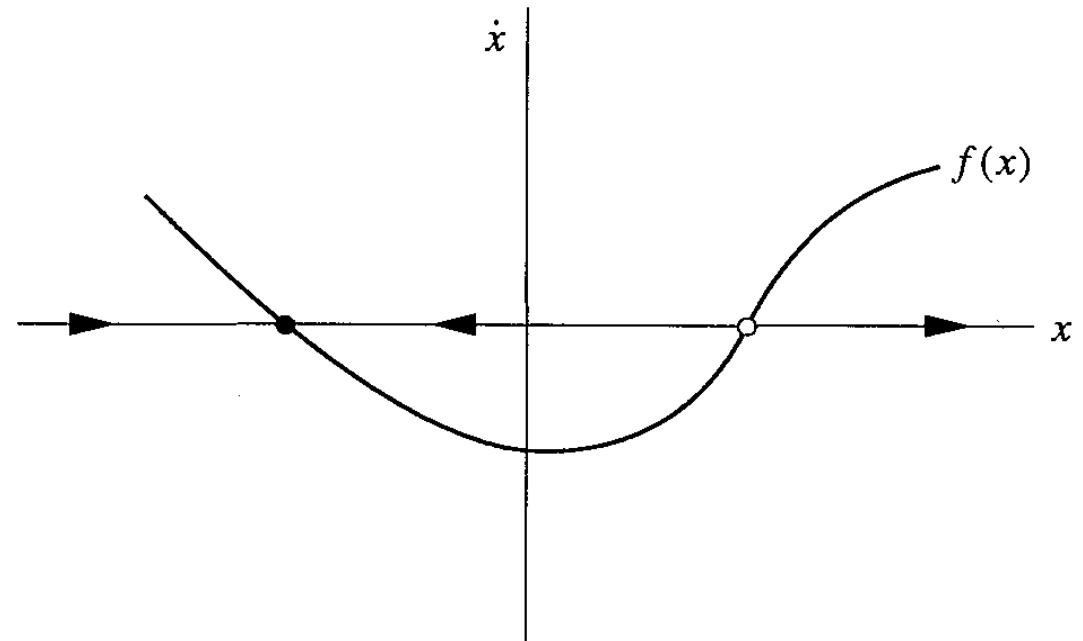
- Surprise: **essentially unsolvable** (e.g., in closed form)
- Families of **periodic** solutions found, but not the full realm of solutions
- Field stuck right after its first success...
- Breakthrough [Poincaré 1890s]: focus on qualitative asymptotic questions: “**will the moon ever fly away?**”
- The **limit behavior** of the system

The Three-Body Problem?  
sun – earth – moon *[Euler 1770]*

- cf: Euler's method for solving ODEs

# 1D systems

- There can be no periodic solution: only equilibria (**stable/unstable**) where the graph of  $f(x)$  intersects the x-axis
- Q: how does one prove convergence?
- A: potential/Lyapunov functions

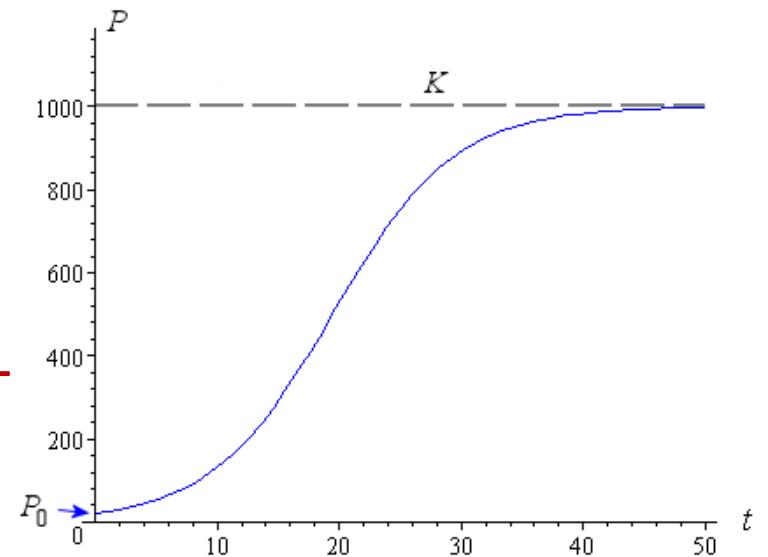


# 1D systems: more examples

- Exponential growth.  $\dot{x} = a x$   
solution:  $x(t) = x_0 \exp(at)$
- The logistic equation: growth within limits

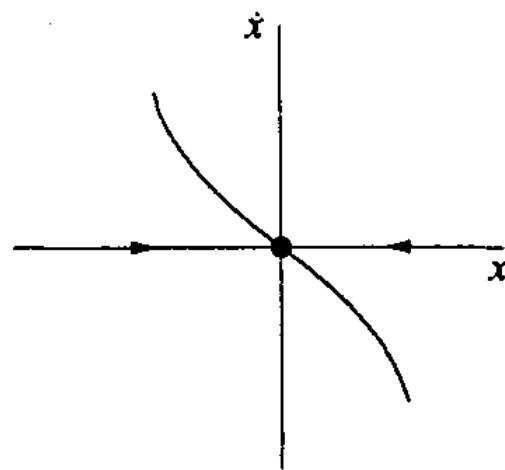
$$\dot{x} = a x (1 - x)$$

solution:  $x(t) = [1 + (1 - x_0/K) \exp(-at)]^{-1}$

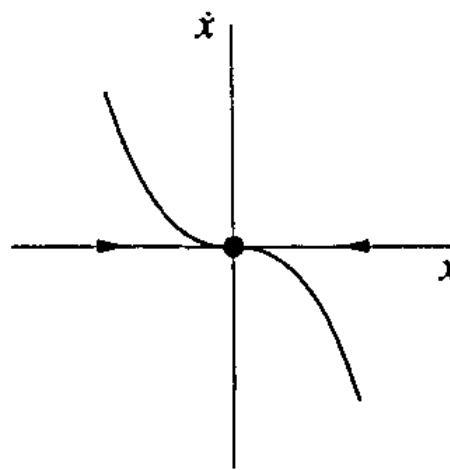


# 1D systems: bifurcation

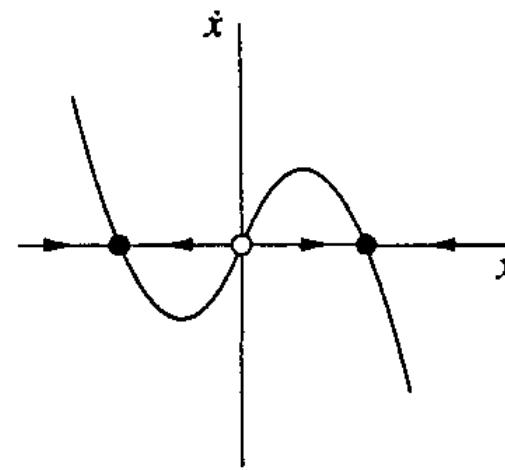
$$\dot{x} = r x - x^3$$



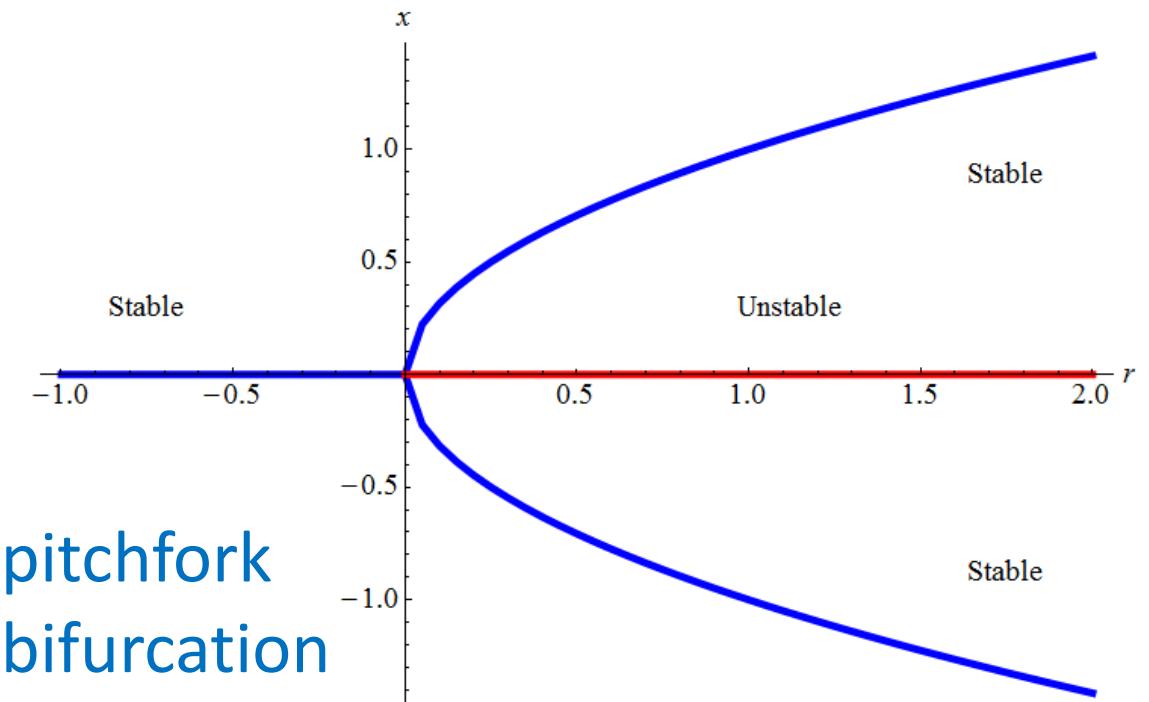
(a)  $r < 0$



(b)  $r = 0$



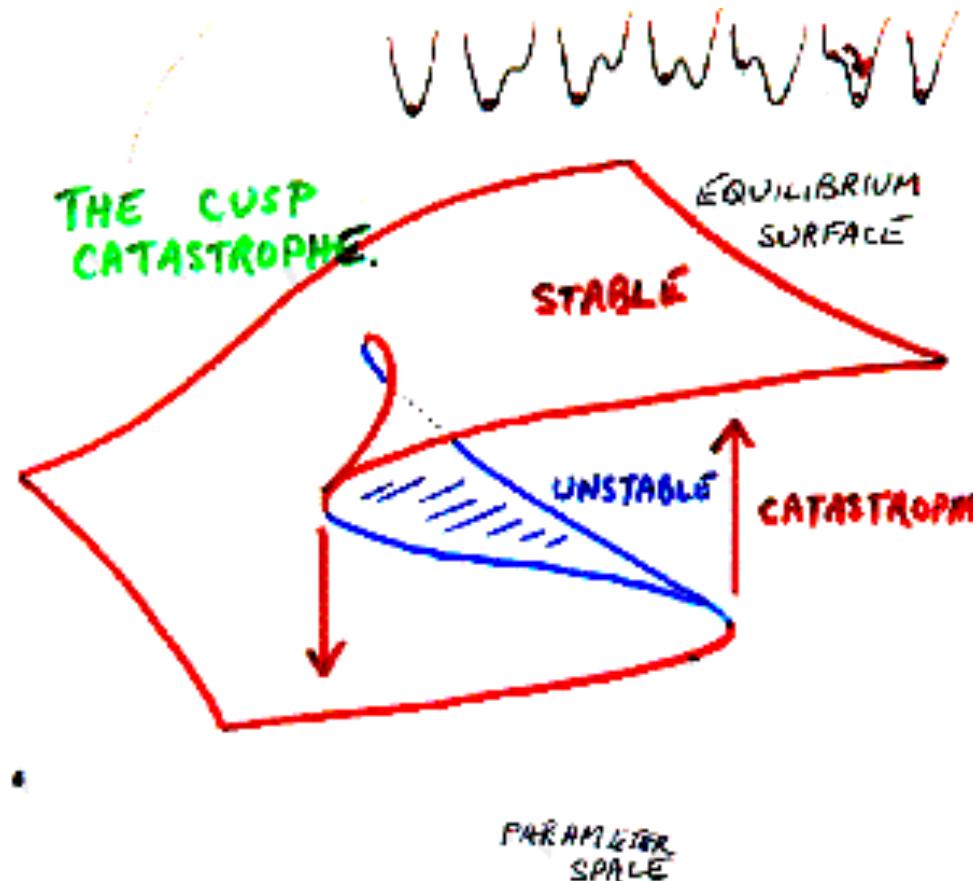
(c)  $r > 0$



pitchfork  
bifurcation

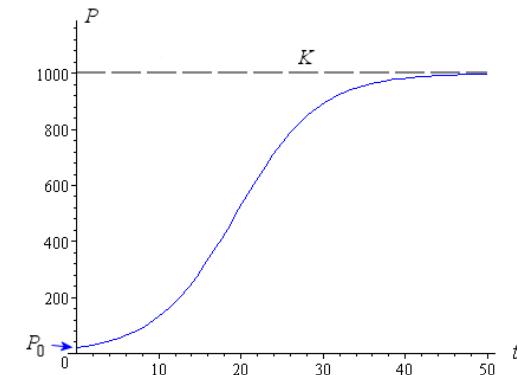
# 1D systems: catastrophic bifurcation

$$\dot{x} = h + r x - x^3$$



Btw, recall the logistic equation

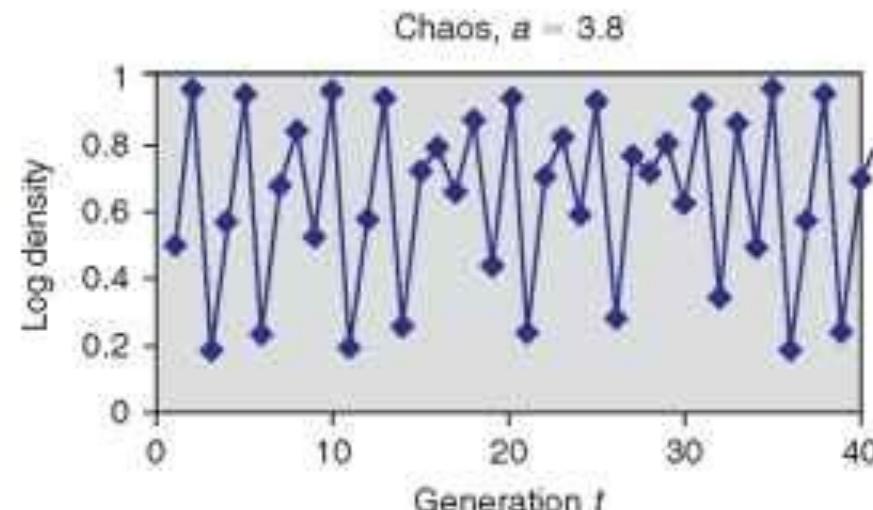
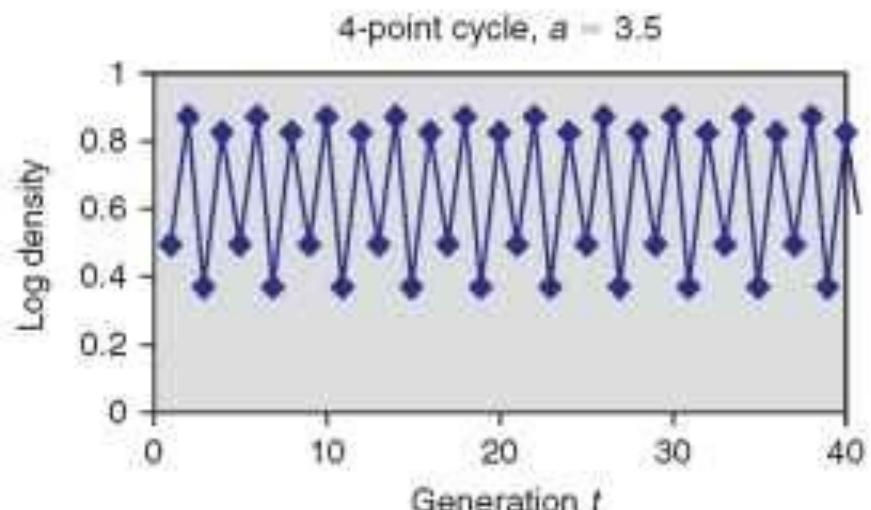
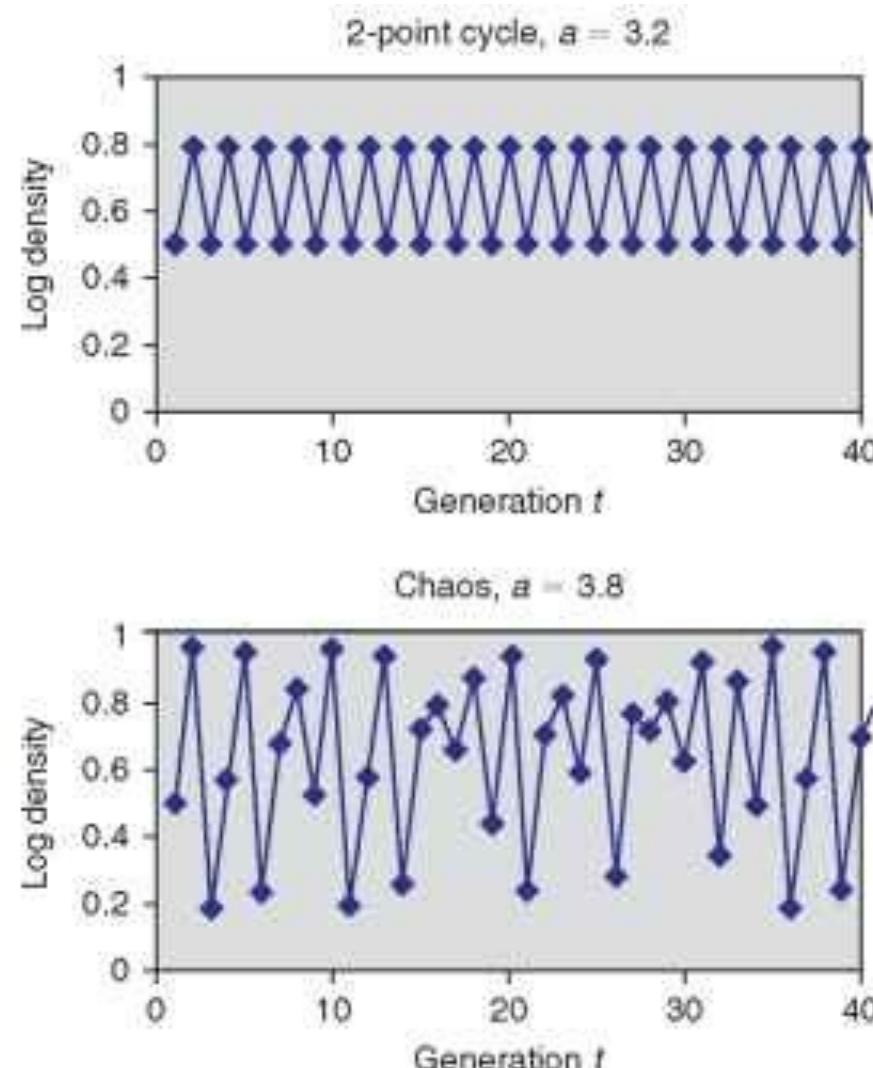
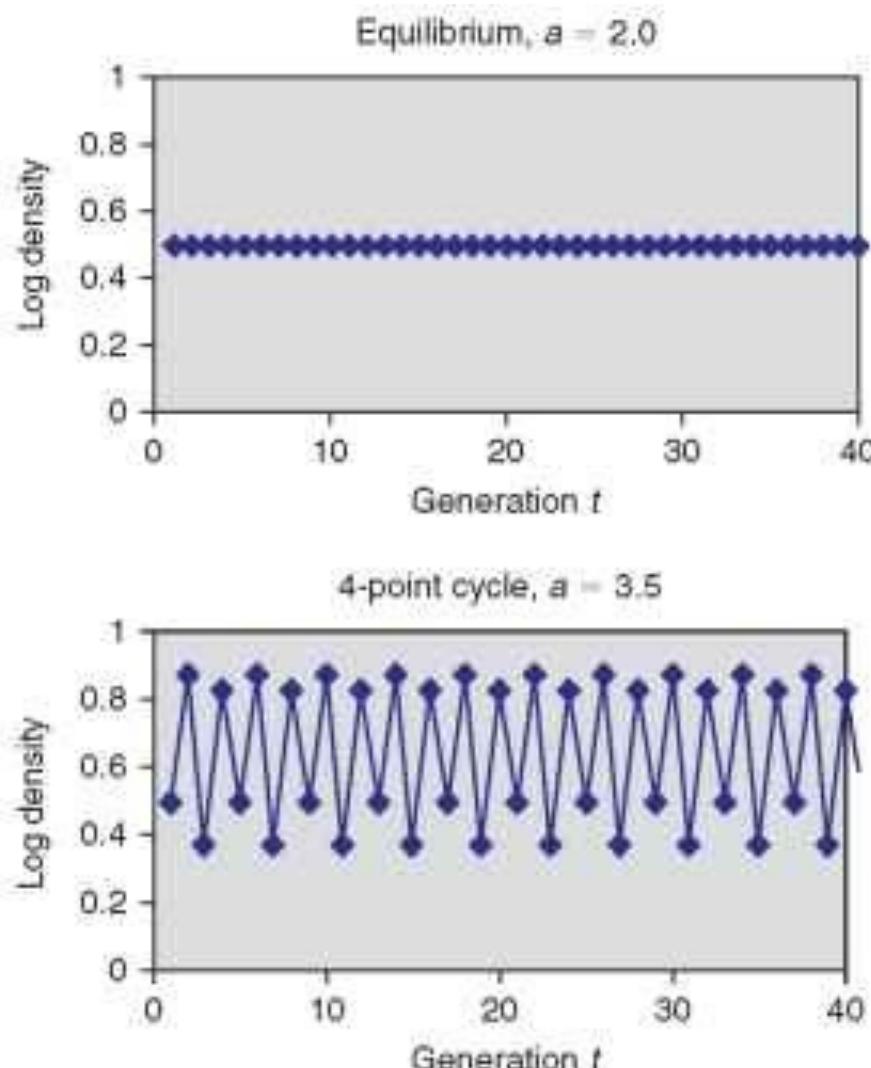
$$\dot{x} = a x (1 - x)$$



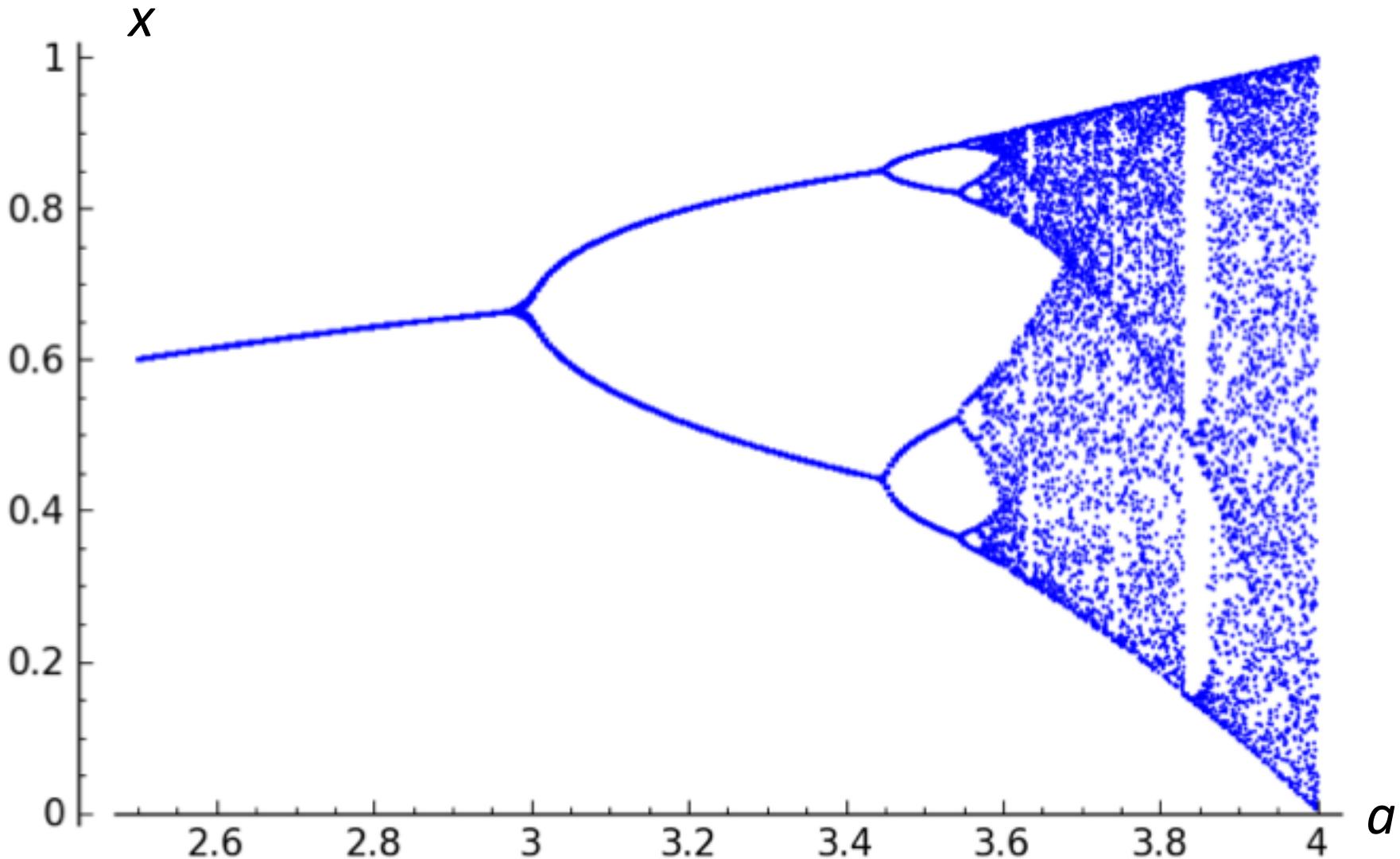
- The discrete-time logistic equation

$$x^{t+1} = a x^t (1 - x^t)$$

$$x^{t+1} = a x^t (1 - x^t)$$



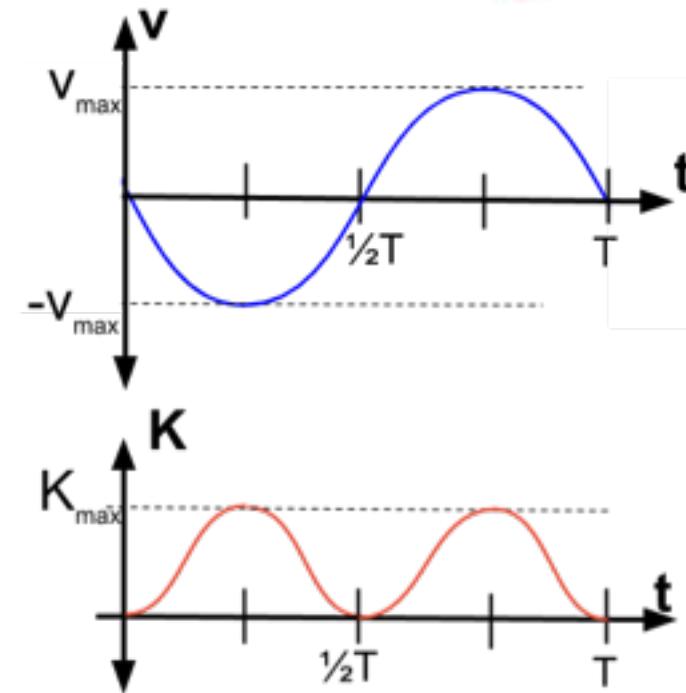
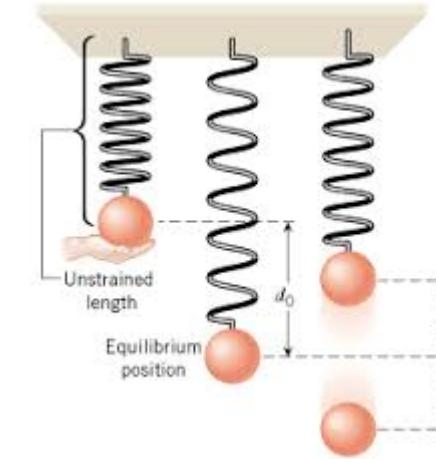
$x^{t+1} = a x^t (1 - x^t)$ : a taste of chaos...



# 2D systems: periodic solutions (cycling!)

$$m\ddot{x} = -kx$$

- The harmonic oscillator
- Or the pendulum



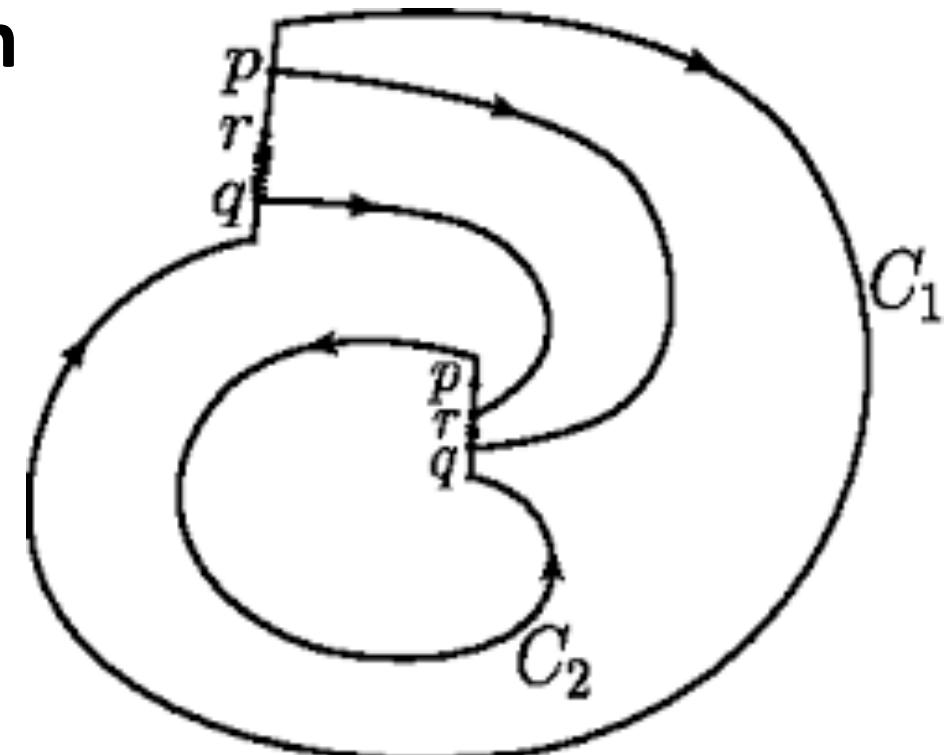
Still, 2D is a relatively safe place...

**Poincaré – Bendixson theorem** (ca. 1900): In any 2D dynamical system, starting from any point either (1) you converge to an equilibrium, or (2) you end up cycling forever\*.

**Reason:** In 2D *planarity* constrains the orbits  
(\*or you are at a rare situation that is the combination of the two...)

# One and two dimensional ODEs, summary

- In 1D dynamical systems, the limit behavior is **equilibrium** (or growth): there are no cycles
- In 2D? **Poincaré – Bendixson theorem**  
In 2D the limit behavior is either **stationary** (equilibrium) or **periodic** (cycles)
- There can be no **chaos** here, the flow “restrains itself”



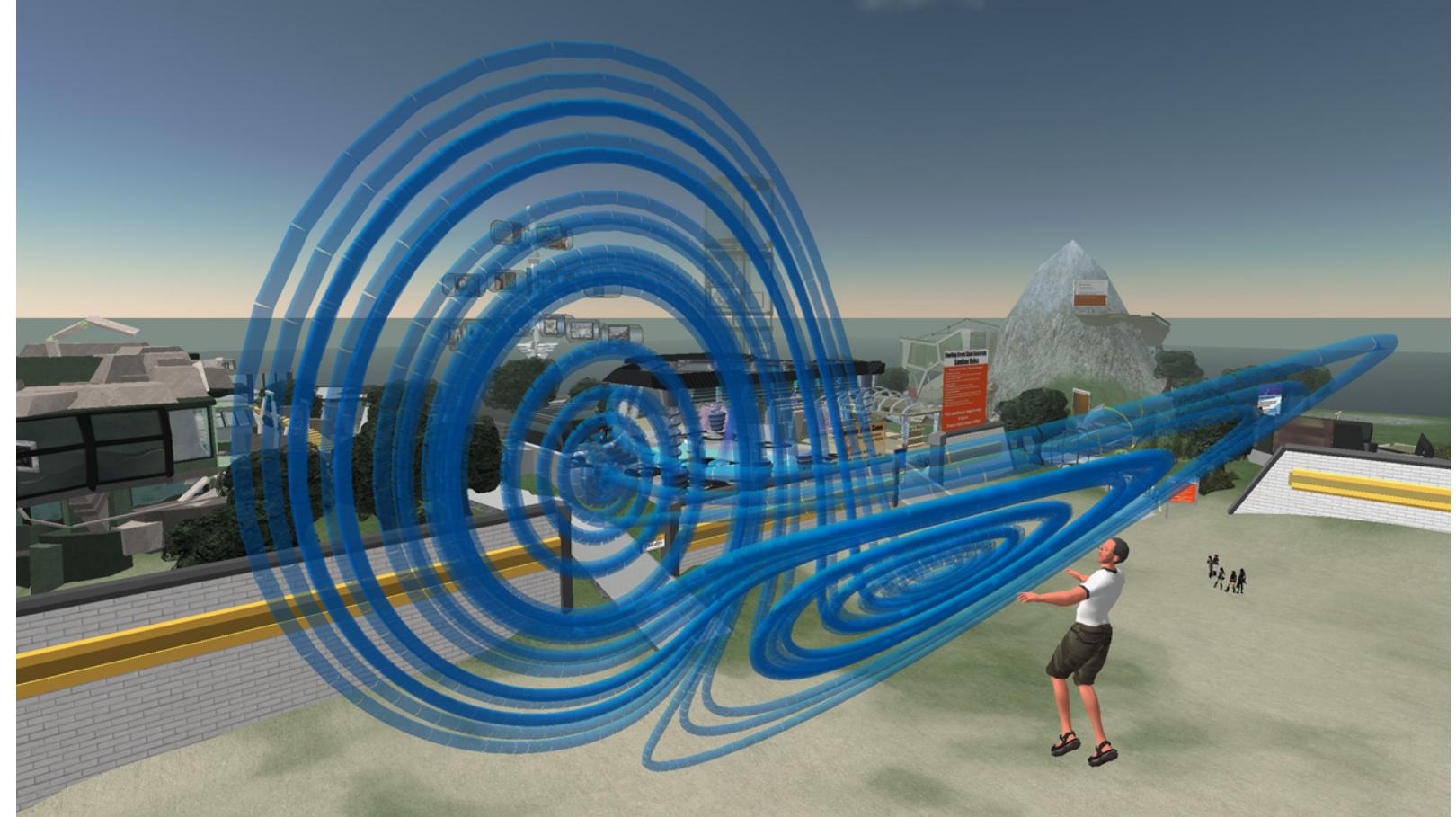
# Three dimensional dynamical systems

Lorenz oscillator, 1963: **CHAOOS**

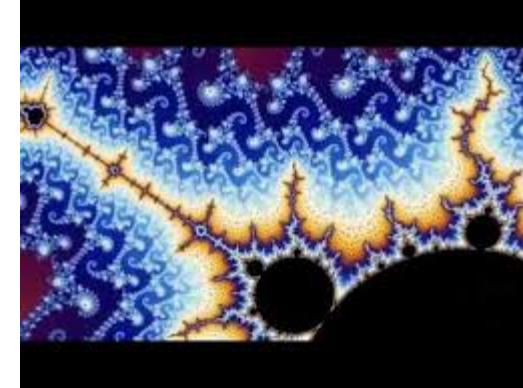
$$\dot{x} = a(y - x)$$

$$\dot{y} = x(b - z) - y$$

$$\dot{z} = xy - cz$$



# What is Chaos?



- Exponentially small **perturbations** in parameters and initial conditions lead to **qualitatively** different behaviors
- A seemingly periodic behavior repeats forever, except that the system **never exactly cycles** (Lorenz)
- An **attractor** is **strange** (fractal-like)
- In discrete time: there cycles of all kinds of periods (but a cycle of period three is enough....)
- The system cannot be **solved** (or understood...) in any satisfactory way

# Mandelbrot set



The set of all complex numbers for which  
the mapping  $z \rightarrow z^2 + c$  remains bounded

# Against chaos: Properties you want your dynamical system to have

- **Conservative** systems: they conserve energy, or other quantities of interest
- **Reversible** systems: they can be “run backwards”
- Systems that have a **Lyapunov function** (progress toward convergence)

# The fundamental theorem of dynamical systems: “Poincare’-Bendixson beyond 2D”

- $D > 2$ : is there a notion of a **cycle** so that the P-B theorem is **restored** (despite chaos)?
- 1900 – 1980: topologists looked for it
- Discrete time, say (continuous time follows)
- Suppose for all  $\varepsilon > 0$  there is a  $N$  such that from  $x$  I can come back to  $x$  with a sequence of **<  $N$  steps** alternating with **jumps of length  $< \varepsilon$**
- Call such a point  $x$  **chain-recurrent**

# The fundamental theorem of dynamical systems (cont.)

- **Theorem** [Conley 1984]: The domain of any dynamical system (*both continuous- and discrete-time*) can be decomposed in the **chain recurrent components (CRC)** and the **transient** parts. There is a Lyapunov function that drives any transient point towards the CRCs.
- In other words “**if you squint a little, chaos goes away and the dynamics converges to equilibria or “cycles”...**”

OK, that was our quick introduction to dynamical systems

- Next: dynamical systems for modeling parts of the Brain
- Continuous and discrete
- A few representative examples
- Avoiding chaos

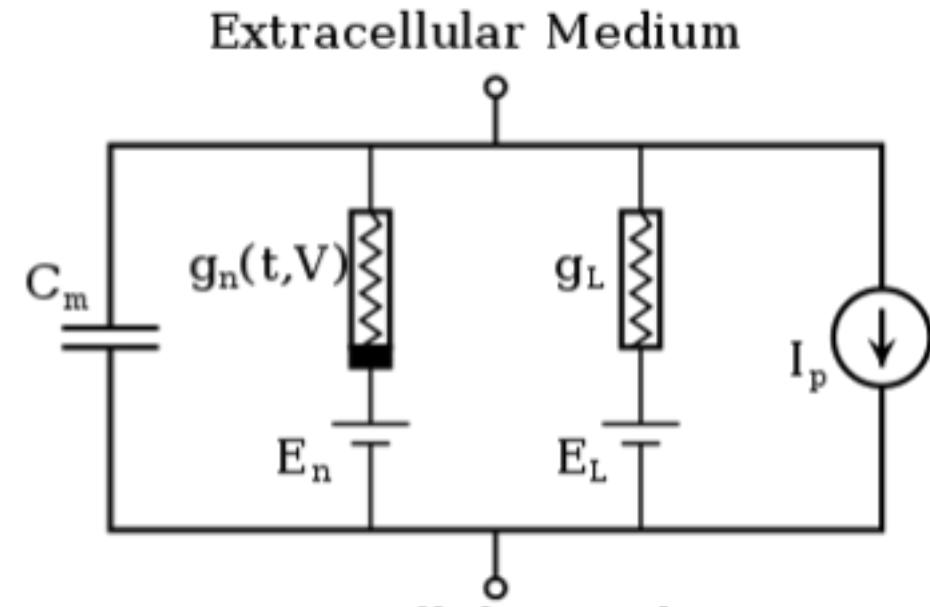
We have seen one

- The Hodgkin-Huxley oscillator

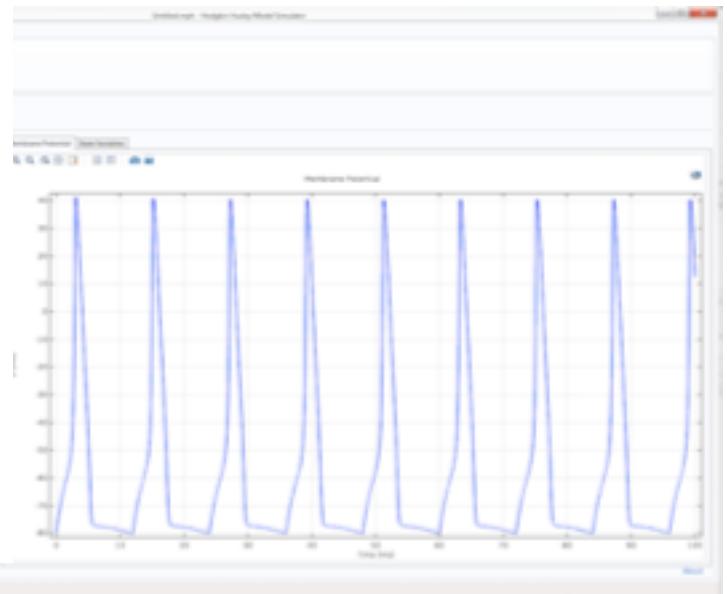
$$\frac{dv}{dt} = \frac{1}{C_m} [I - g_{Na}m^3h(v - E_{Na}) - g_Kn^4(v - E_K) - g_L(v - E_L)]$$

$$\frac{dm}{dt} = \alpha_m(v)(1 - m) - \beta_m(v)m \quad (1)$$

$$\frac{dh}{dt} = \alpha_h(v)(1 - h) - \beta_h(v)h$$



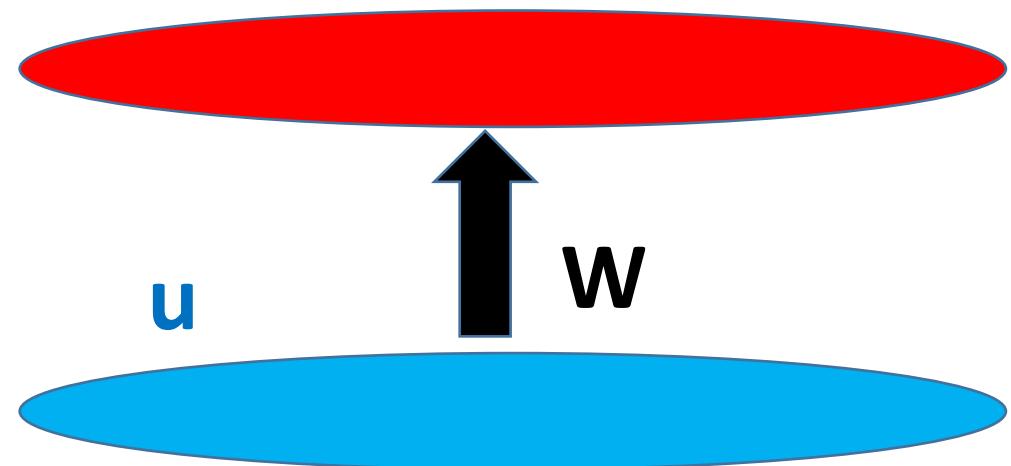
Intracellular Medium



# Feedforward network

- Two populations of neurons
- Feedforward synaptic connections  $v$
- $u$ ,  $v$ : vectors of spiking rates
- $W$ : matrix of synaptic weights

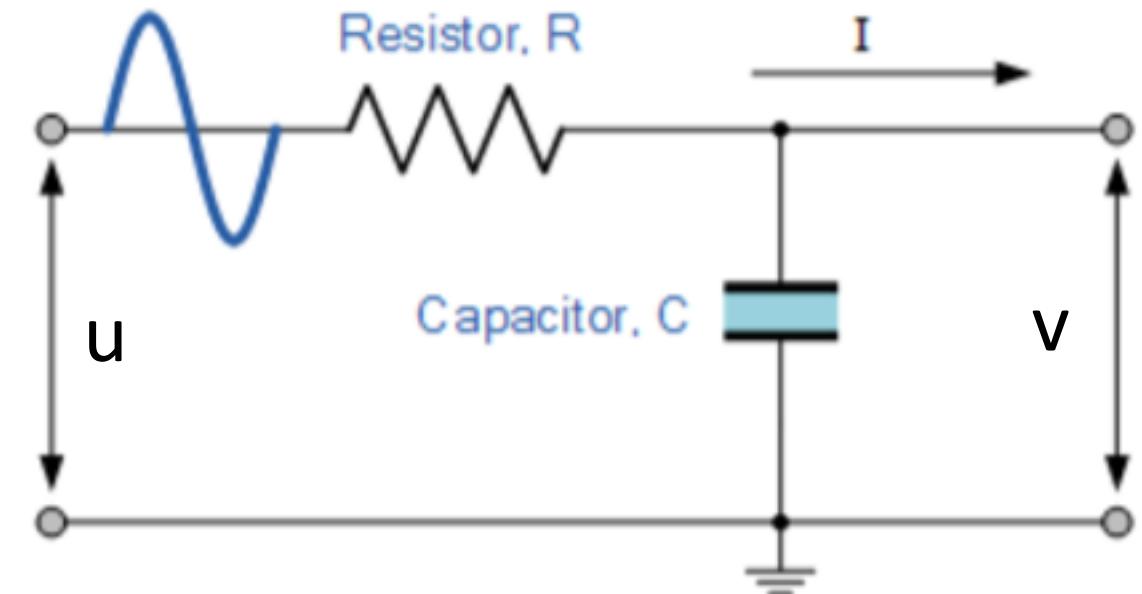
$$\tau \cdot dv/dt = -v + F(W \cdot u)$$



cf: the low-pass filter

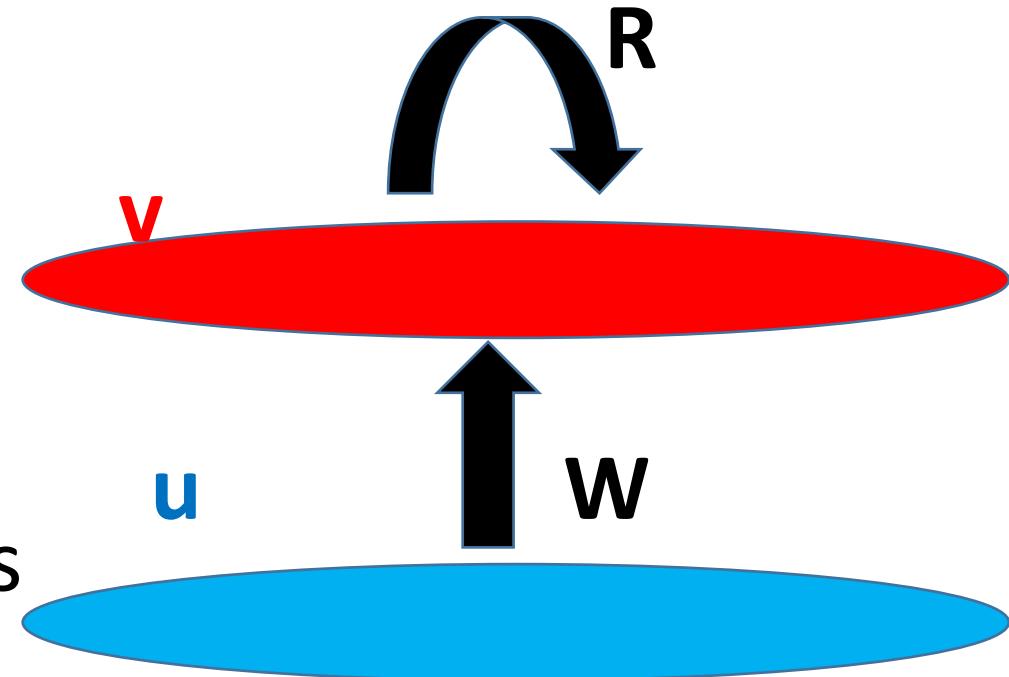
$$\tau \cdot \frac{dv}{dt} = -v + u$$

$$\tau = RC$$



# Feedforward and Recurrent network

- Two populations of neurons
- Feedforward and recurrent synaptic connections
- $u$ ,  $v$ : vectors of firing rates
- $W$ : matrix of synaptic weights



$$\tau \cdot dv/dt = -v + F(W \cdot u + R \cdot v)$$

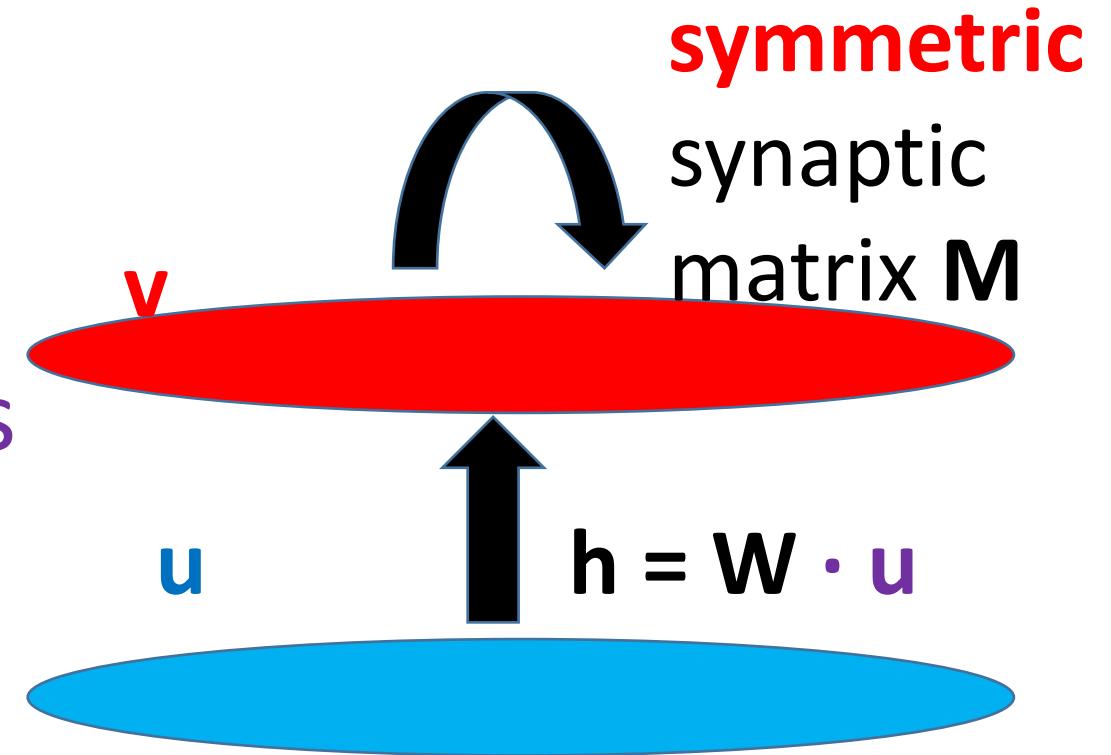
Suppose  $F$  is *linear* and  $M$  *symmetric*

$$\tau \cdot dv/dt = h + (M - I) \cdot v$$

$M$  has positive eigenvalues

$\{\lambda_k\}$  with orthonormal  
eigenvectors  $v_k$

eigenvalues  $\lambda_1 \geq \lambda_2 \geq \dots$



# What is the solution?

Write  $v(t) = \sum_k f_k(t) e_k$  and write ODEs for the  $f_k(t)$ 's :

$$\tau \cdot df_k/dt = -(1 - \lambda_k) \cdot f_k(t) + e_k \cdot h \quad \text{Solve ( assume } \lambda_k \neq 1\text{)}$$

$$f_k(t) = \exp(-t(1 - \lambda_k)/\tau) \cdot (e_k \cdot (v(0) - h/(1 - \lambda_k))) \\ + e_k \cdot h / (1 - \lambda_k)$$

First term:  $\lambda_1 > 1 \rightarrow \text{exponential growth}$  (hence impossible)

Suppose that  $\lambda_2 \ll \lambda_1 \approx 1$

We can ignore all other terms, and we have the asymptotic solution

$$v^* \approx (e_1 \cdot h) \cdot e_1 / (1 - \lambda_1)$$

So, this circuit takes the feedforward input  $h$  and projects it on  $e_1$ , amplifying it by a large number  
(If two eigenvalues of  $M$  are close to one, it projects on a plane...)

Finally, suppose  $\lambda_1 = 1$

Recall :  $\tau \cdot dc_1/dt = n - (1 - \lambda_1) \cdot c_1(t) + e_1 \cdot h$

So, the circuit *integrates* the feedforward input's projection on  $e_1$

NB: Integration means memory. (Why?)

A system like this seems to be at work in the brain stem of vertebrates, remembering eye position

# ReLU FFR networks

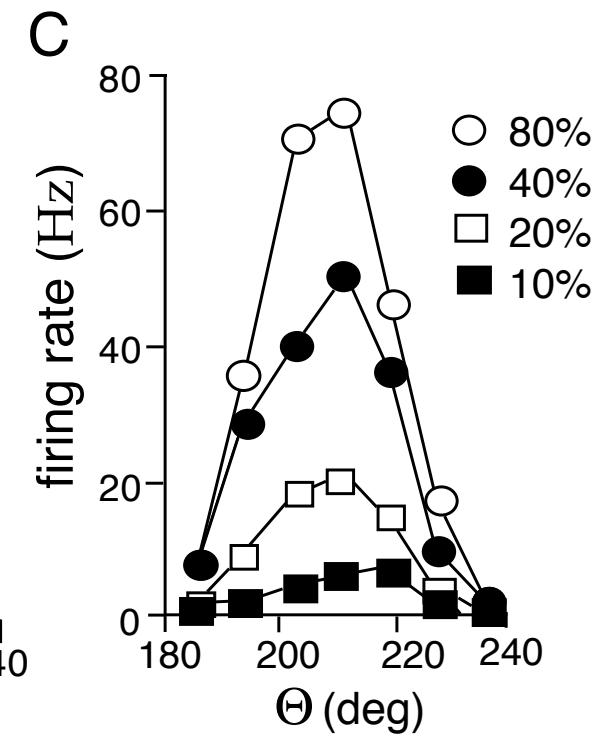
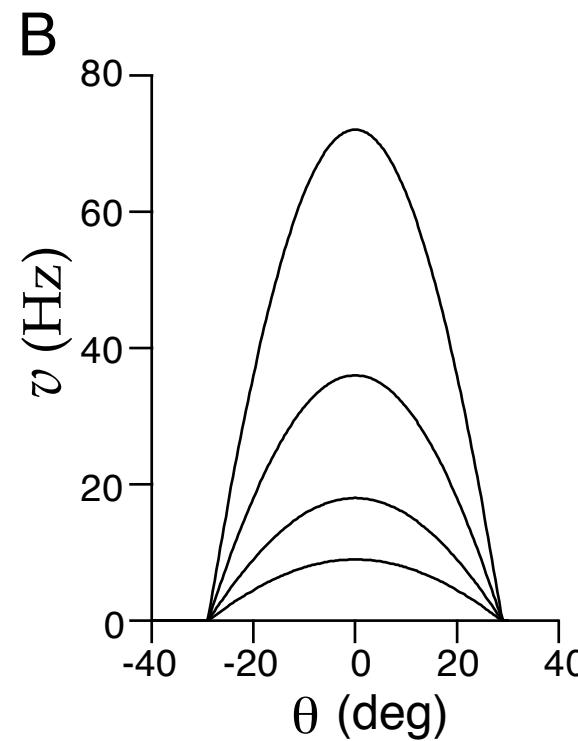
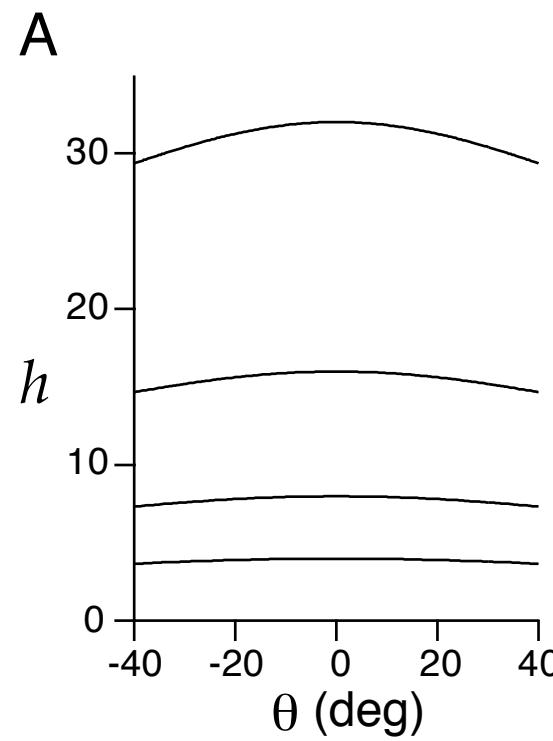
$$\tau \cdot dv/dt = -v + [h + M \cdot v - T]_+$$

Subtract a vector of thresholds  $T$  and set to zero if negative

*Can model simple and complex cells!*

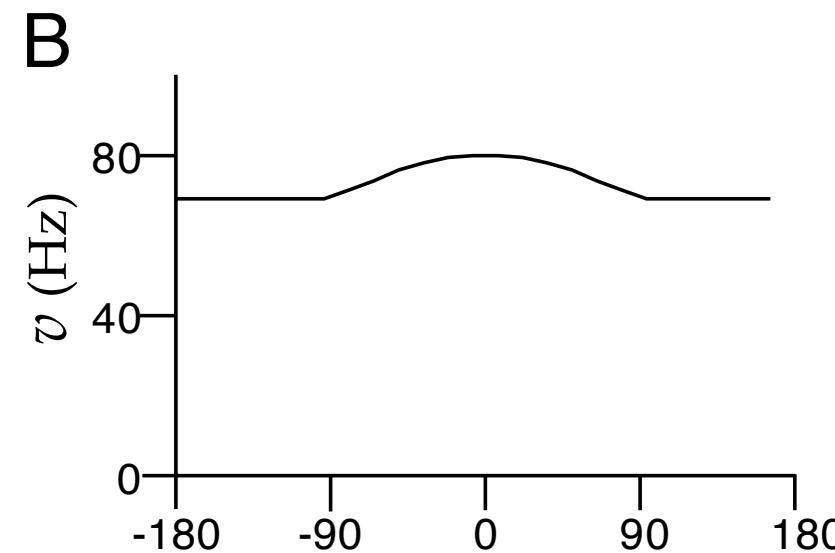
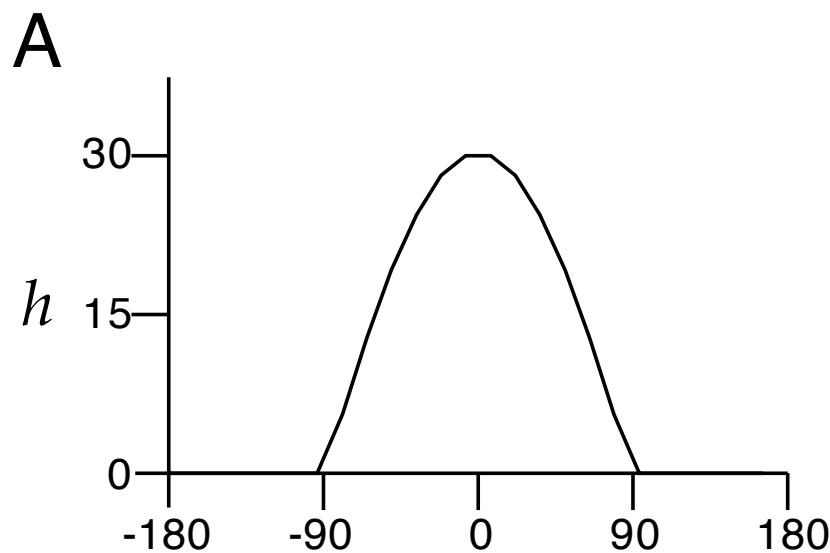
Recurrent input from V1-V2 strong, must be modeled by FFR network with array of cells indexed by angle  $\theta$  in  $-40^\circ$  to  $40^\circ$ , with synaptic weight  $M(\theta_1, \theta_2) \sim \cos(\theta_1 - \theta_2)$

# Modeling simple cells in V1



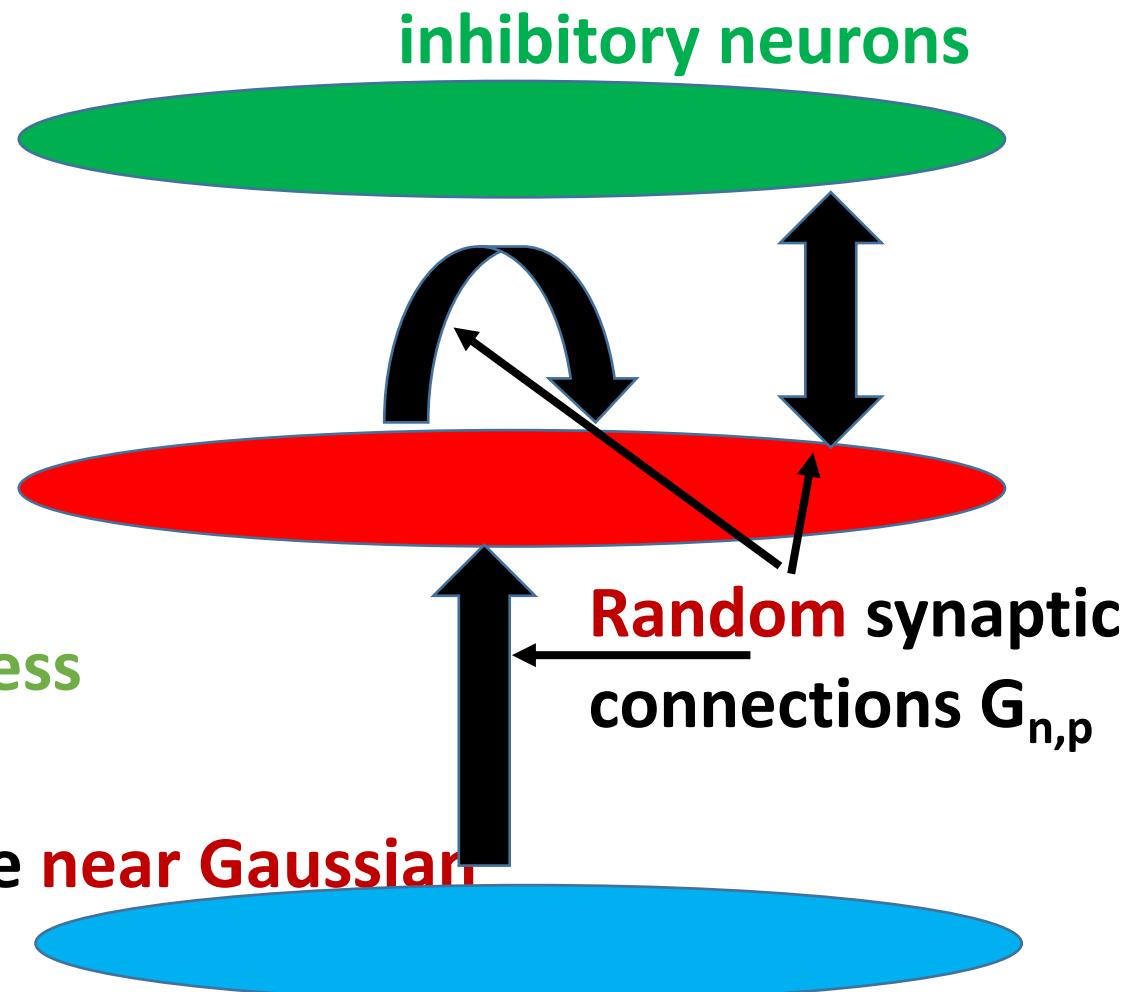
# Modeling complex cells

- Again, an array of neurons indexed by  $\theta$
- Input  $h(\theta)$  is a simple cell with preferred angle  $\theta$  (here zero). It feeds the  $\theta$ -neuron in the array
- Constant recurrent synaptic weights  $M$



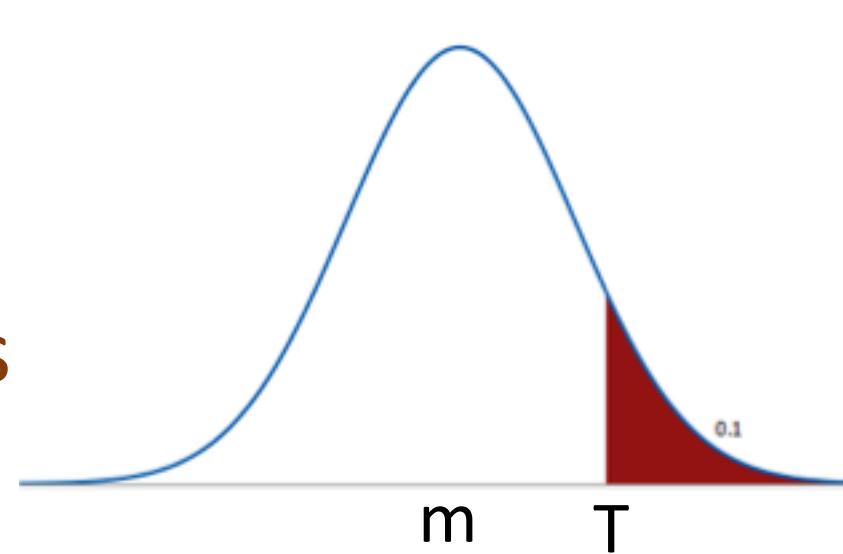
# Another FFR network: Excitation-Inhibition balance

- The blue cells spike
- Many red cells receive input and fire
- The green cells receive input
- They fire, and inhibit the red
- Maybe too much
- Now they receive less input from the red, and they inhibit less
- All these inputs are random
- By the law of large numbers, they are near Gaussian



# E – I balance

Notation:  $GT(T, m, \text{var})$  is the Gaussian tail with parameters  $m$  and  $\text{var}$  above threshold  $T$



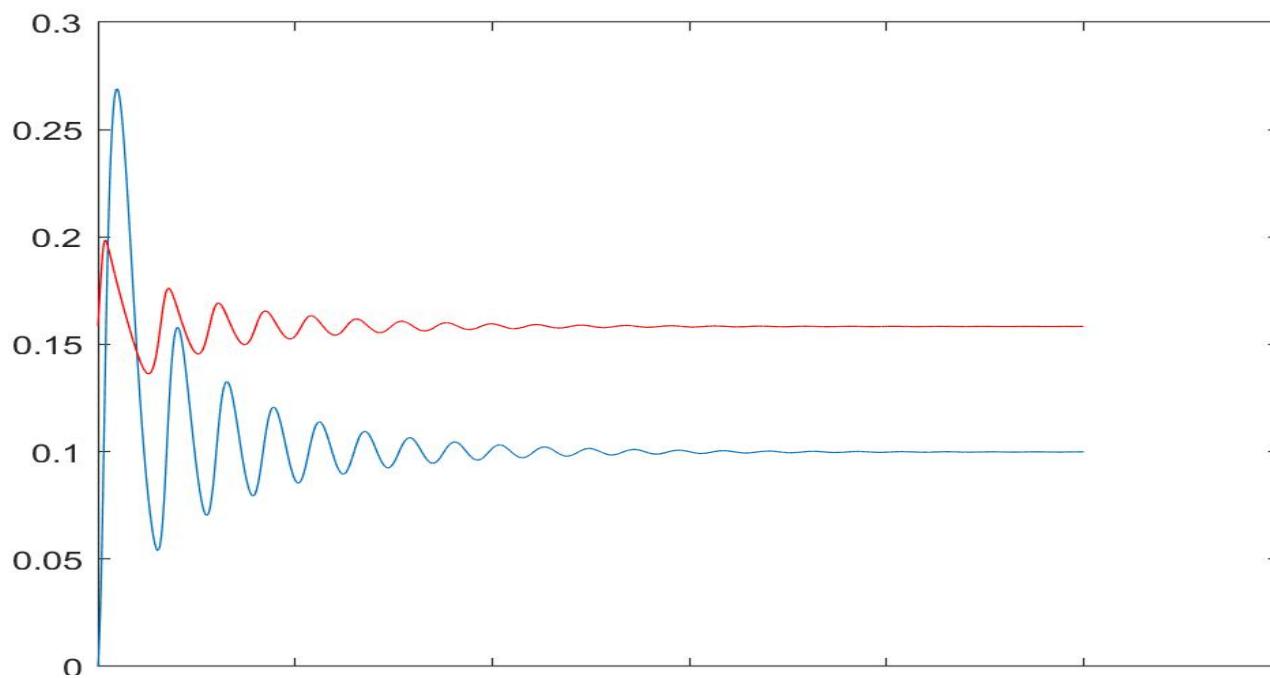
## Nonlinear ODE

$$\tau_E \dot{E} = GT(T_E, np(E - I), np(1-p)(E + I)) - E$$

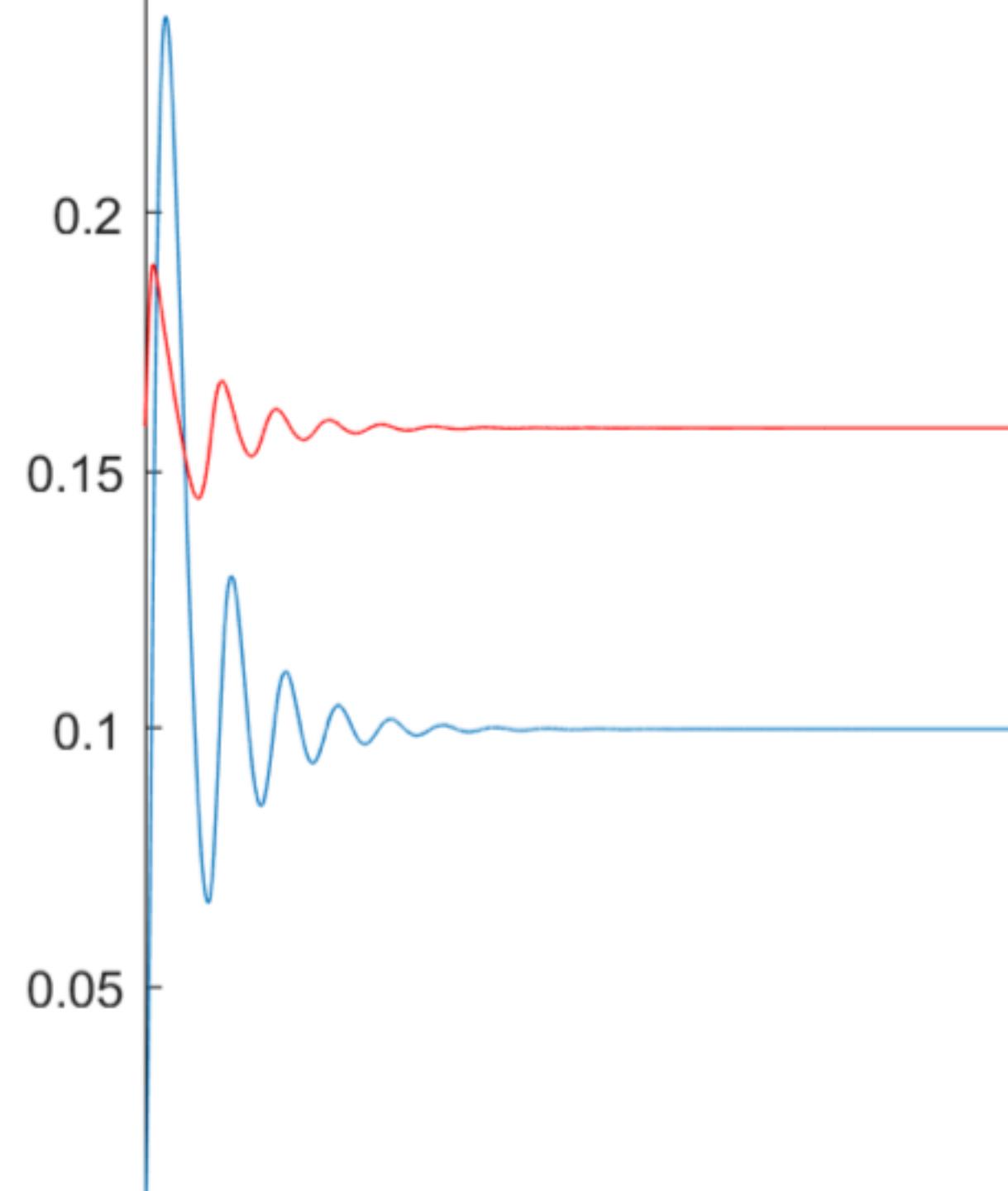
$$\tau_I \dot{I} = GT(T_I, npl, np(1-p)I) - I$$

## $E - I$ balance (cont.)

If  $\tau_E$  is sufficiently larger than  $\tau_I$ , an  $E - I$  balance will be reached after a few up and down oscillations



$E - I$  balance



# A discrete-time neural system: Hopfield net

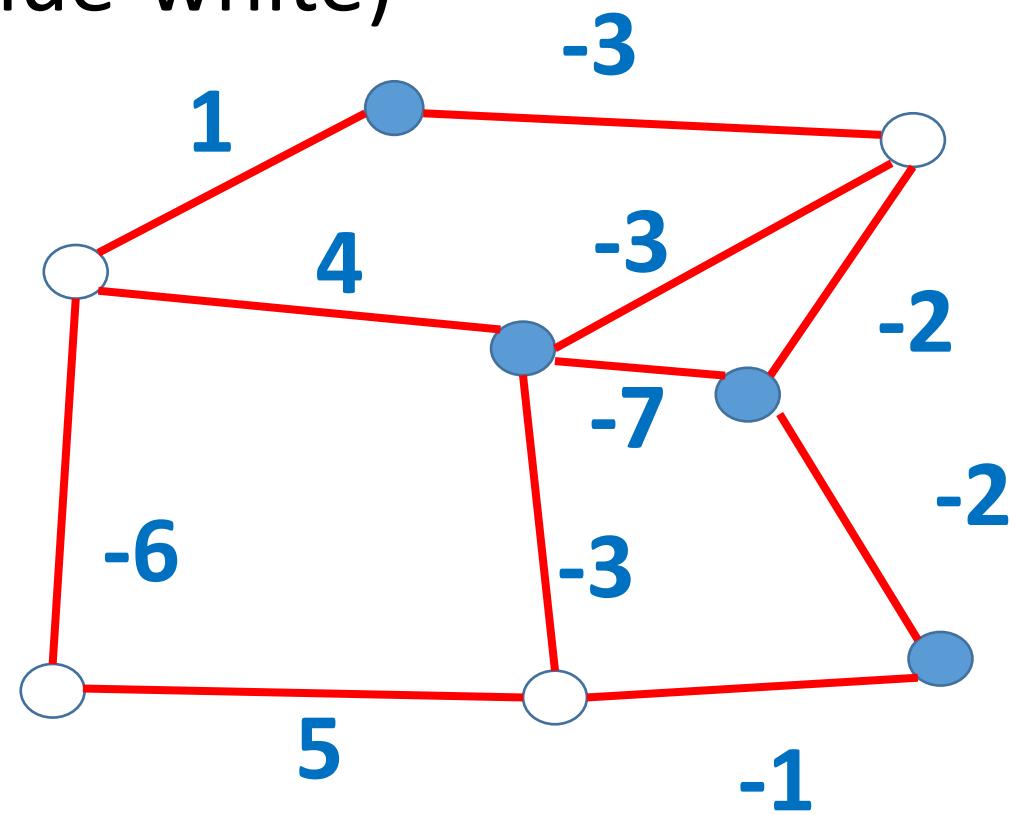
Nodes have two values: +1, -1 (blue-white)

***Symmetric synaptic weights***

Node i is happy if  $\sum_j v_i v_j w_{ij} \geq \theta_i$

Algorithm/dynamical system:

**while there is  
an unhappy node  
flip it**



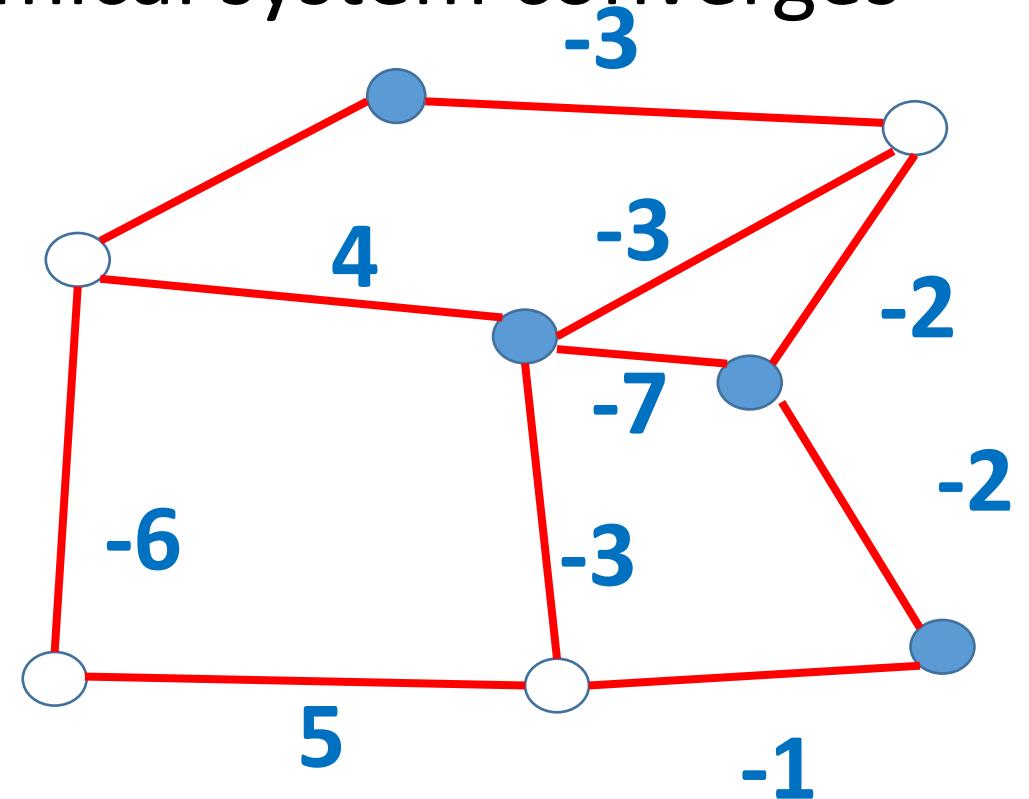
# A discrete-time neural system: Hopfield net

**Theorem** [Hopfield 1982]: Dynamical system converges

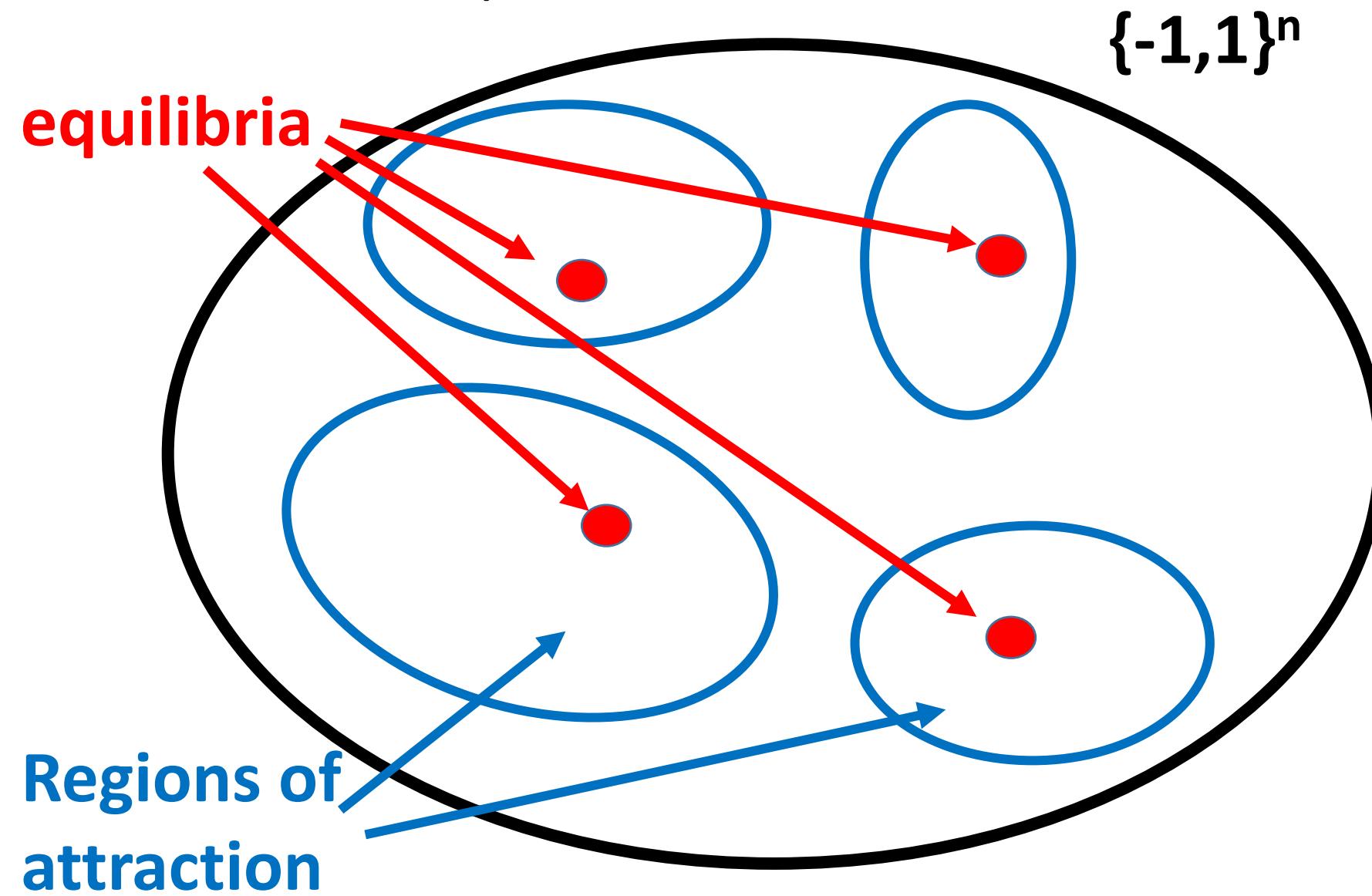
Proof: **Lyapunov** function

$$\chi(v) = \sum_{i,j} v_i v_j w_{ij}$$

always increases



Goal: Pattern completion



How do you train a Hopfield net so that it pattern completes?

- Given a set of desired memories  $M_1, \dots, M_m \in \{-1, +1\}^n$
- Set the **weight** of edge a-b to  $\sum_k M_k^a \times M_k^b$
- That is, for every memory k we bias the weight in the direction “memory k wants the weight to be”.
- Question: does it work?

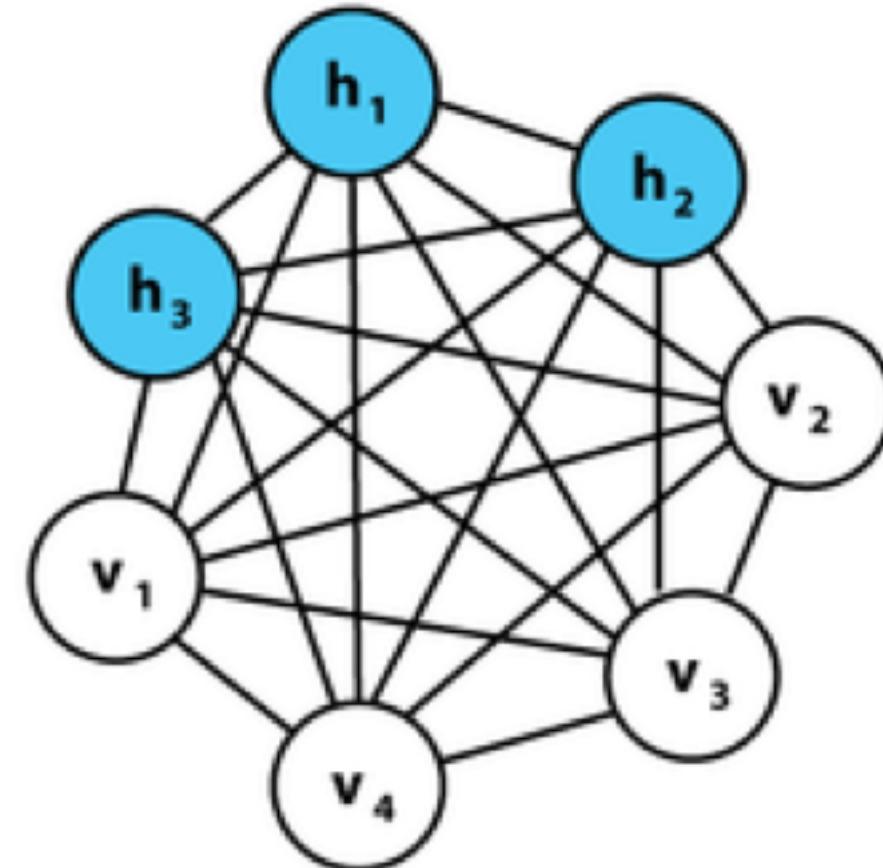
**Theorem:** with n nodes,  $0.138n$  random patterns can be stored with probability of erroneous retrieval  $< 0.4\%$

# Do Hopfield nets work?

- Spurious memories: if  $\mathbf{M}$  is stored,  $-\mathbf{M}$  is also retrieved
- Also, if  $\mathbf{M}, \mathbf{M}', \mathbf{M}''$  are stored, so are  $\pm \mathbf{M} \pm \mathbf{M}' \pm \mathbf{M}''$
- Finally, if you store  $\mathbf{M}$  K times, it will be retrieved  $L \gg K$  times more often than other memories
- Fun: [Hopfield and Tank 1984] show that Hopfield nets solve the traveling salesman problem with 10 cities

# Boltzmann Machines

- Some of the nodes are **hidden**
- The visible nodes receive the training data by clamping
- After each training data vector is input, the whole network is left free to run until it reaches “thermal equilibrium”



# Boltzmann Machines (cont.)

- With some engineering, learning can happen
- Restricted Boltzmann machine: graph is bipartite [Hinton 2005]
- Can be stacked to form a deep net...

