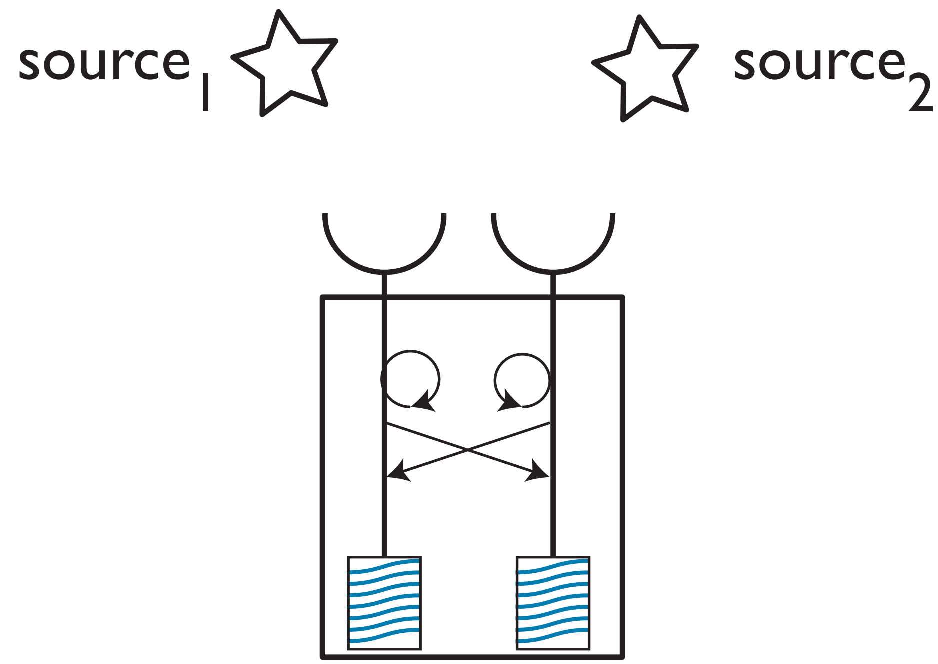
Neural Dynamics

Gregor Schöner
gregor.schoener@ini.rub.de

Activation

- how to represent the inner state of the Central Nervous System?
- => activation concept

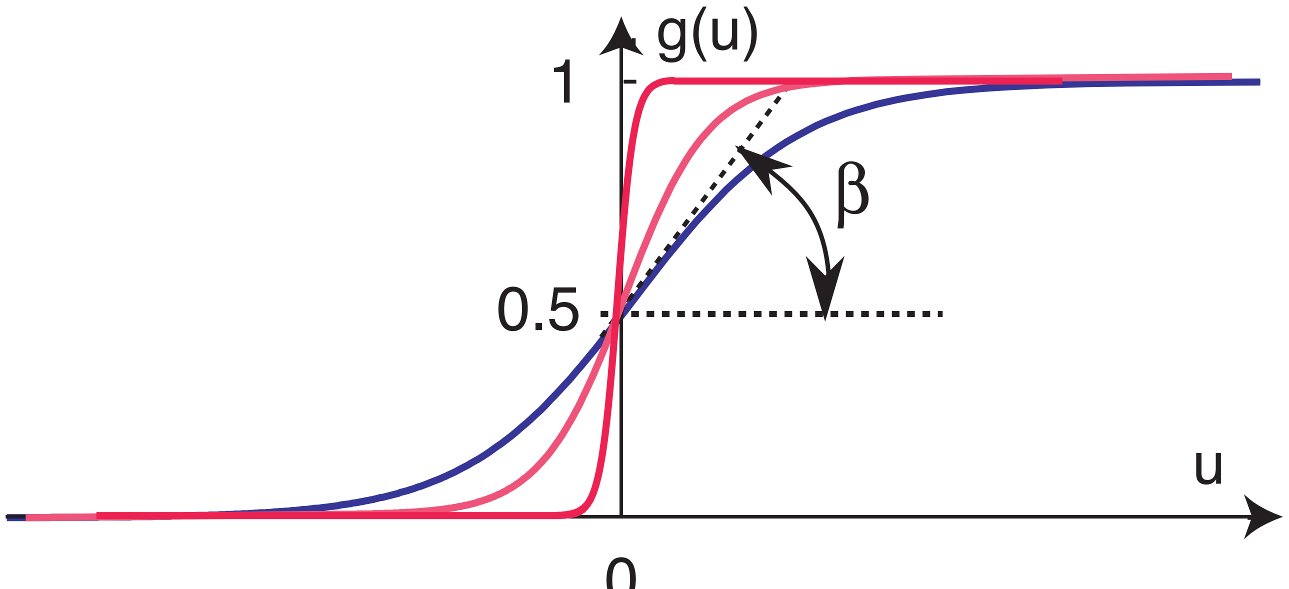


Activation

- neural state variables
  - membrane potential of neurons?
  - spiking rate?
  - ... population activation...

Activation

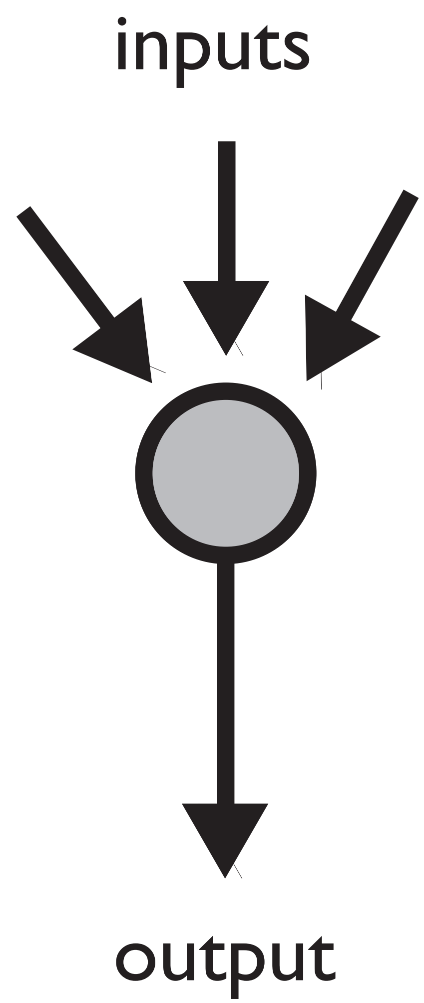
- activation as a real number, abstracting from biophysical details
  - low levels of activation: not transmitted to other systems (e.g., to motor systems)
  - high levels of activation: transmitted to other systems
  - as described by sigmoidal threshold function
  - zero activation defined as threshold of that function

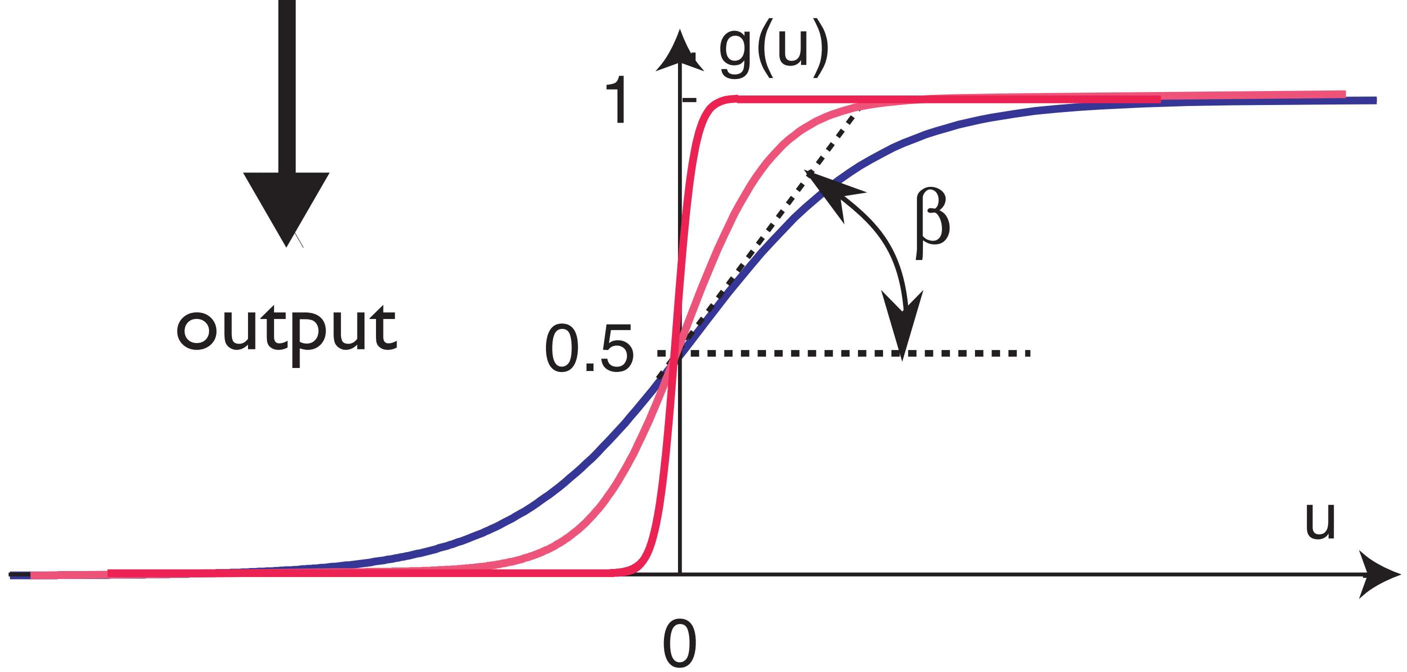


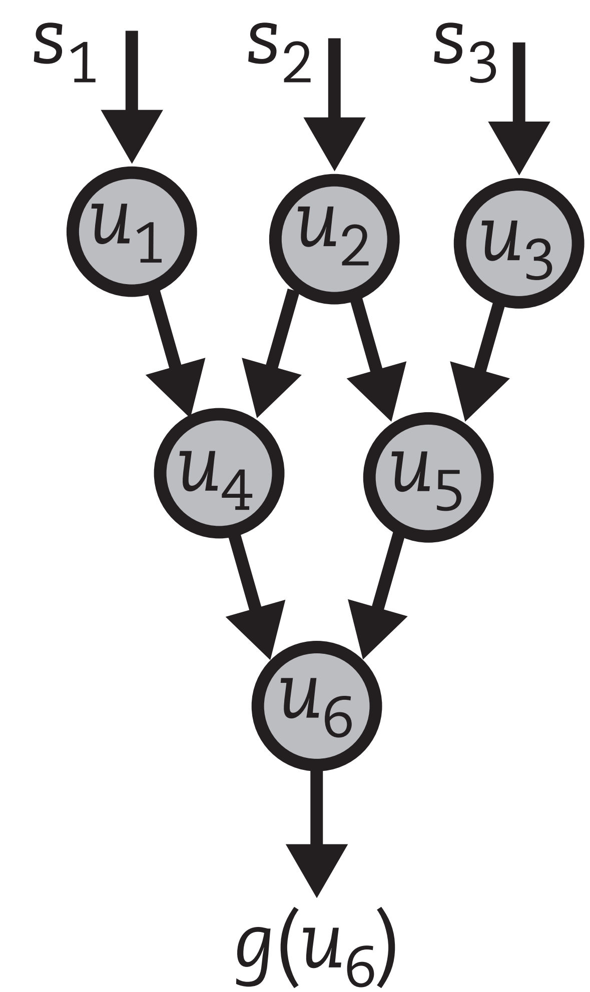
Activation

- compare to connectionist notion of activation:
  - same idea, but tied to individual neurons
- compare to abstract activation of production systems (ACT-R, SOAR)
  - quite different... really a function that measures how far a module is from emitting its output...

Neurons as input-output threshold elements that form feed-forward neural networks



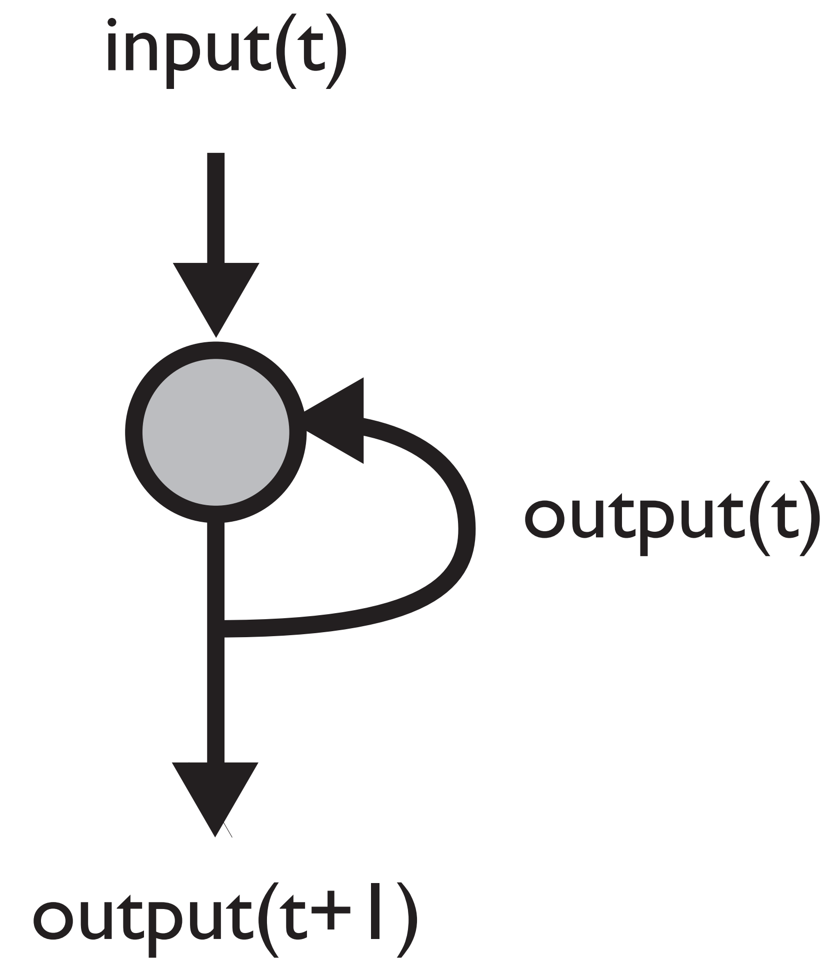




$$\text{output} = g\left(\sum(\text{inputs})\right)$$

Recurrent neural networks

- require a concept of time
- time is not discrete (spiking is asynchronous) => neural dynamics…

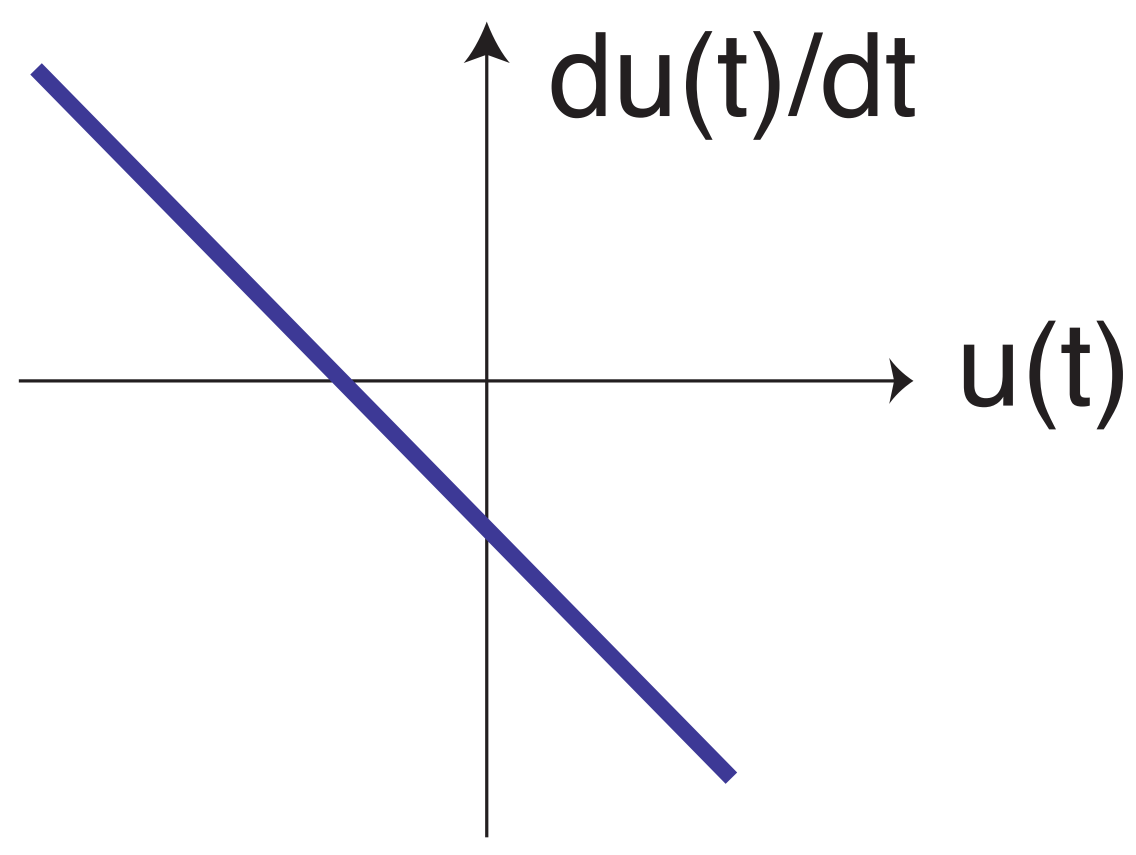


Activation dynamics

- activation variables u(t) as time continuous functions...

$$\tau\dot{u}(t) = f(u)$$

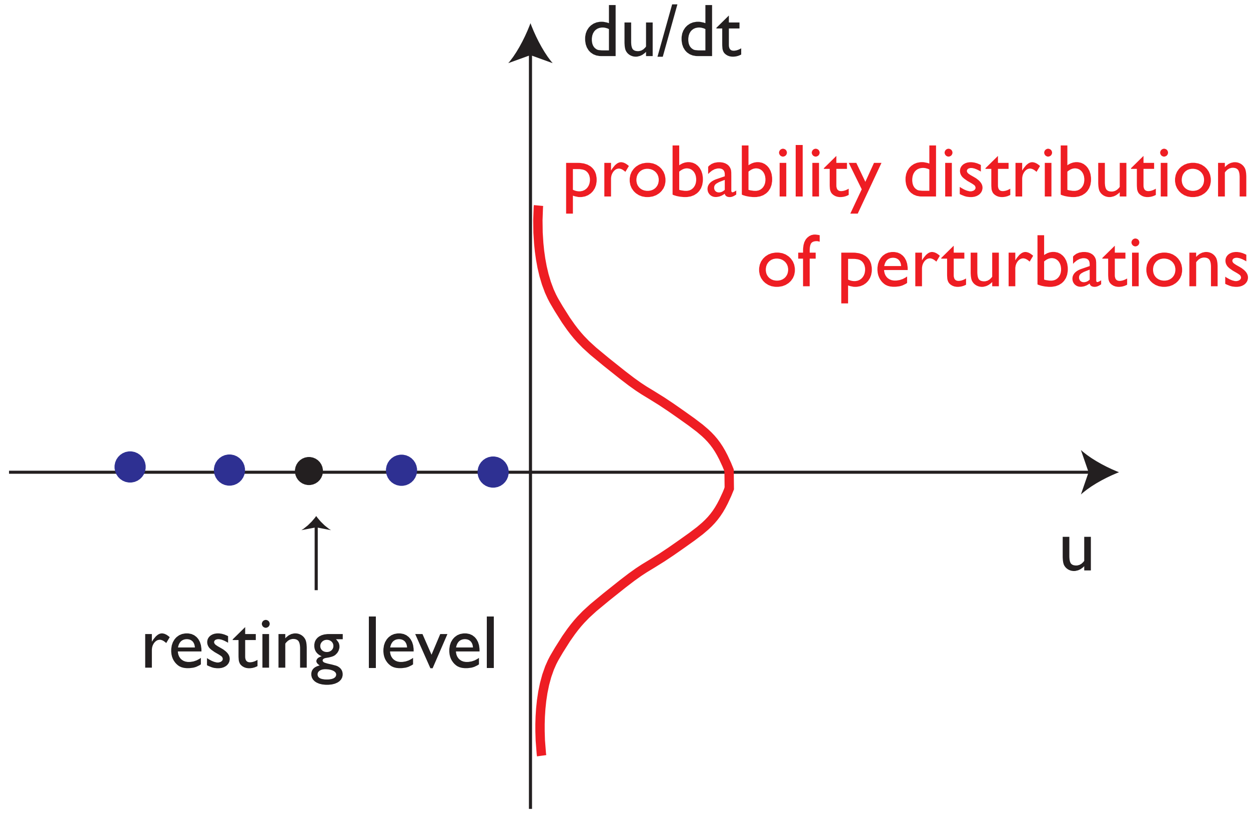
- what function f?

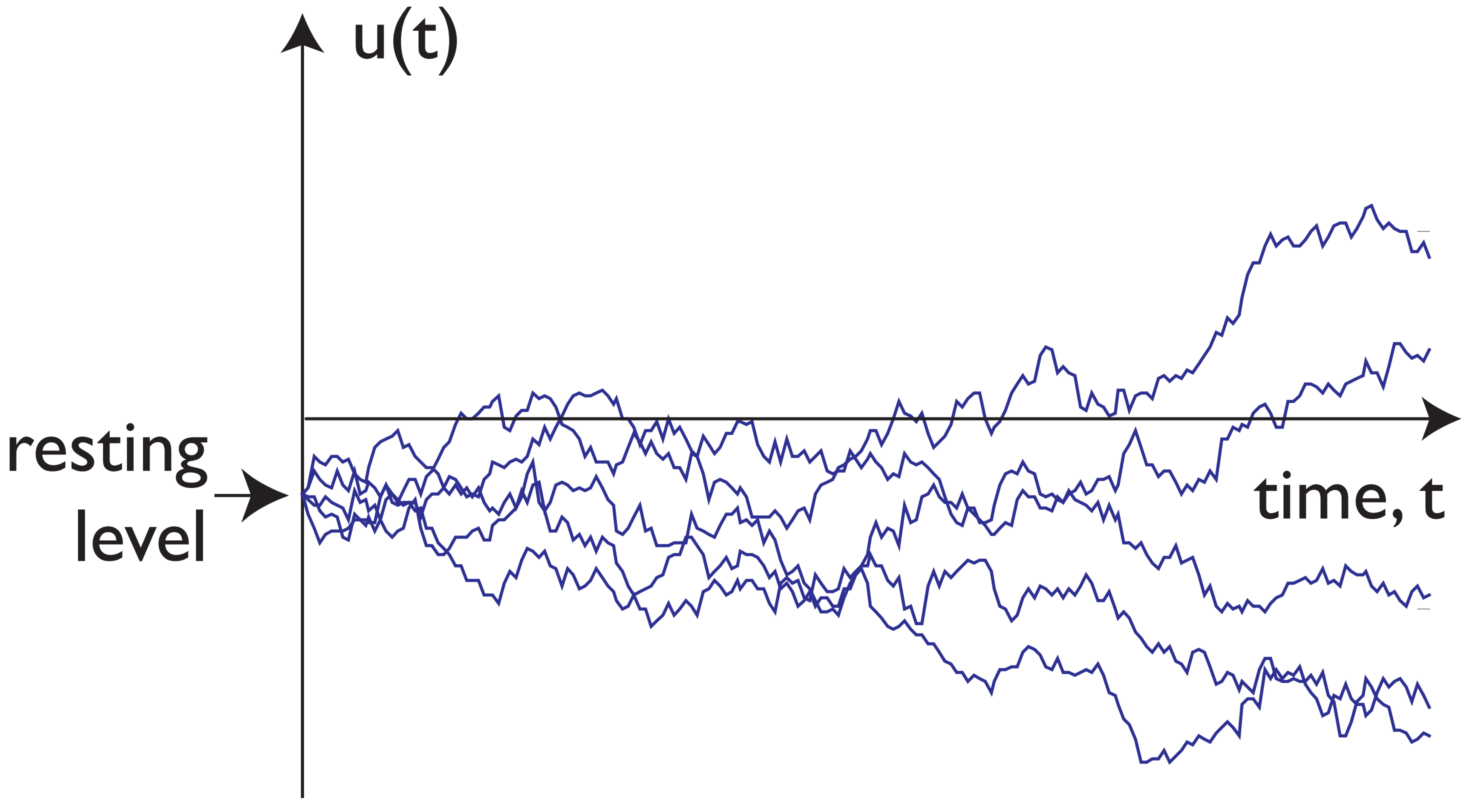


Activation dynamics

- start with f=0

$$\tau \dot{u} = \xi_t$$

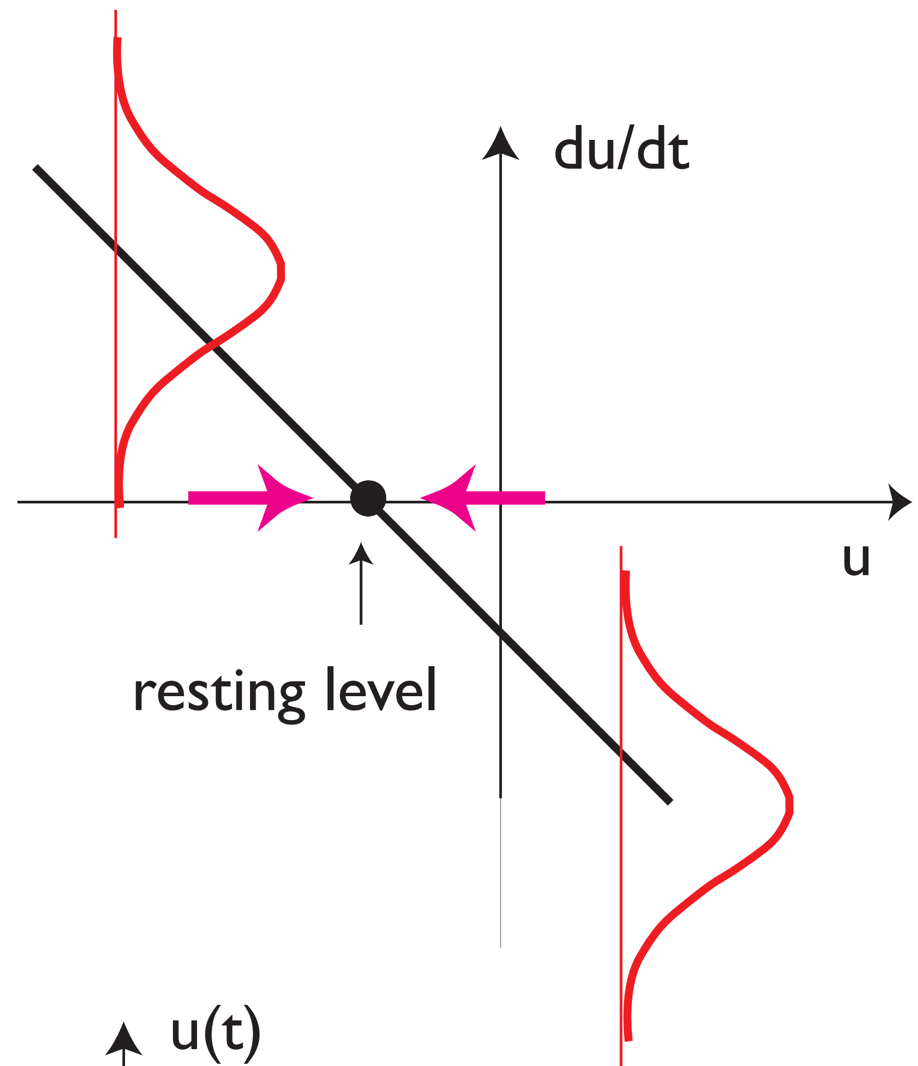


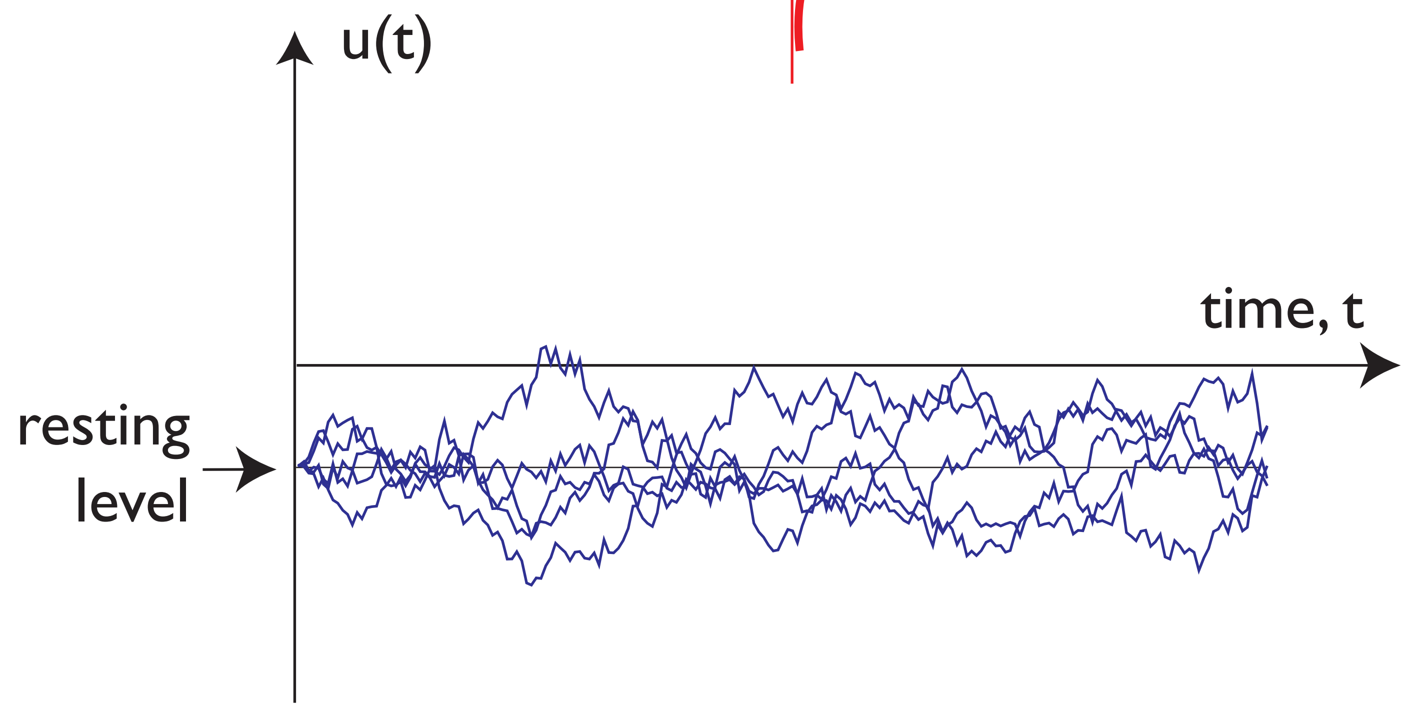


Activation dynamics

- need stabilization

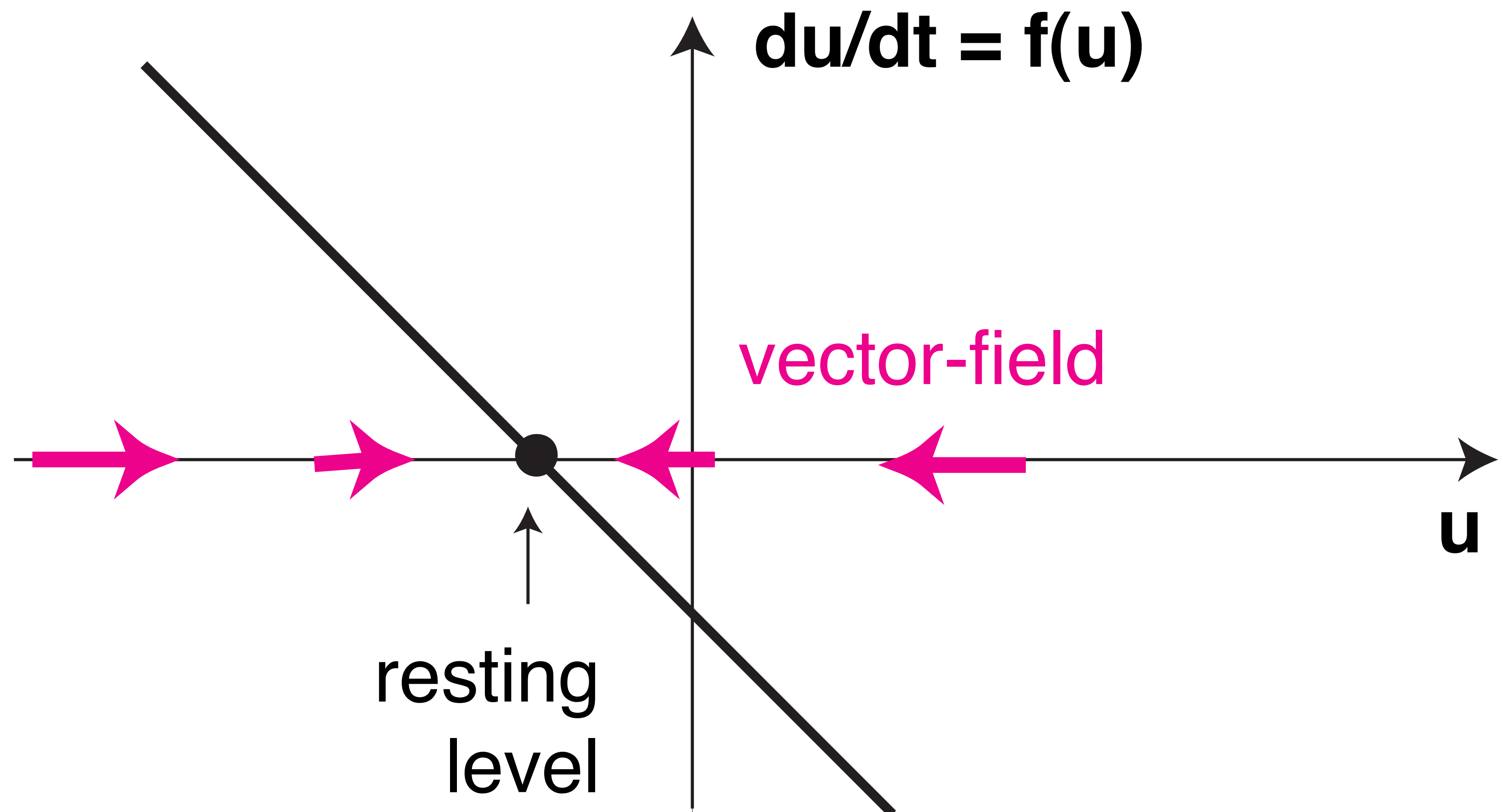
$$\tau \dot{u} = -u + h + \xi_t.$$





Neural dynamics

- In a dynamical system, the present predicts the future: given the initial level of activation u(0), the activation at time t: u(t) is uniquely determined



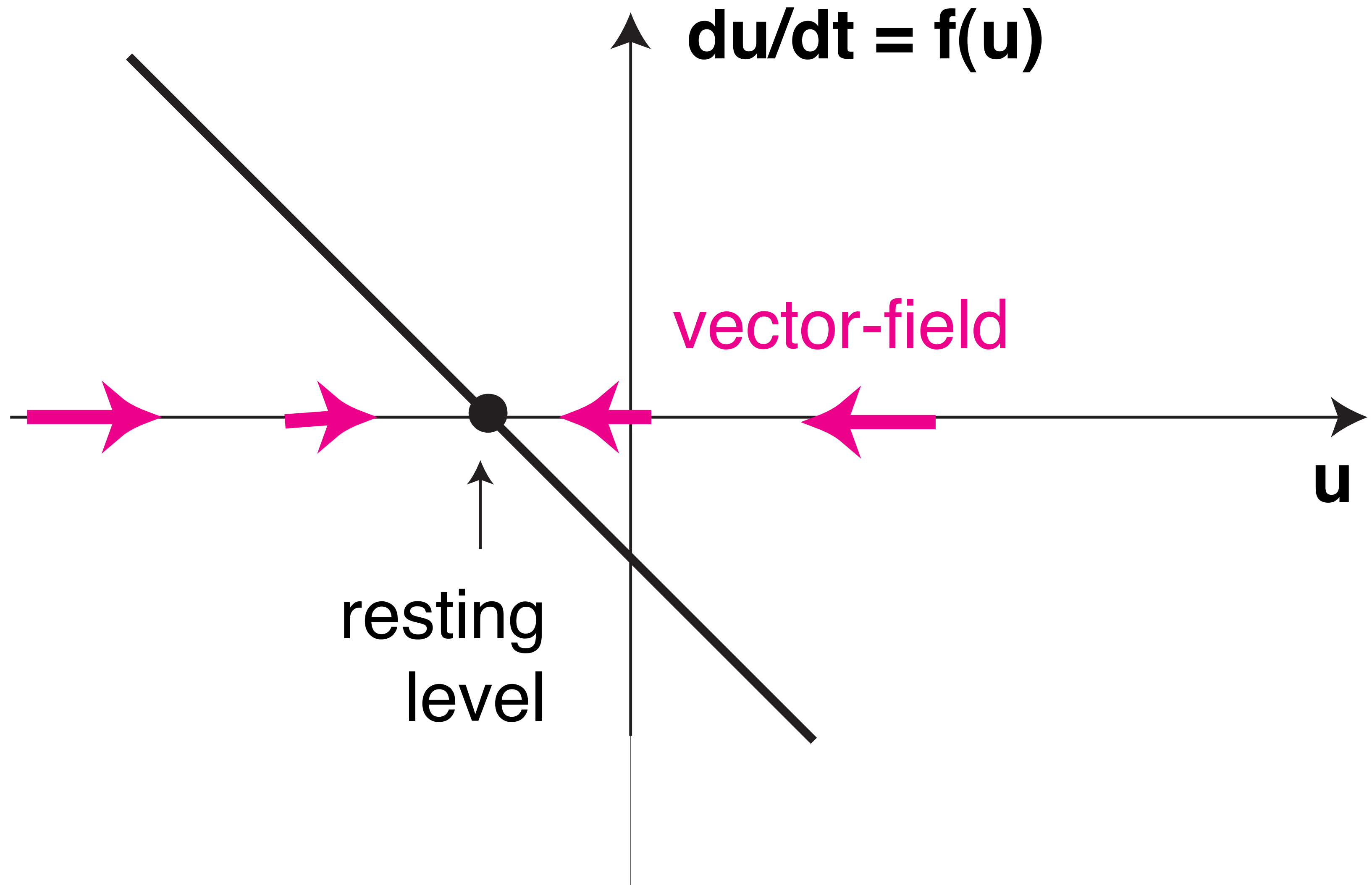
$$\frac{du(t)}{dt} = \dot{u}(t) = -u(t) + h \qquad (h < 0)$$

mental simulation

- => dynamical systems tutorial Mathis Richter

Neural dynamics

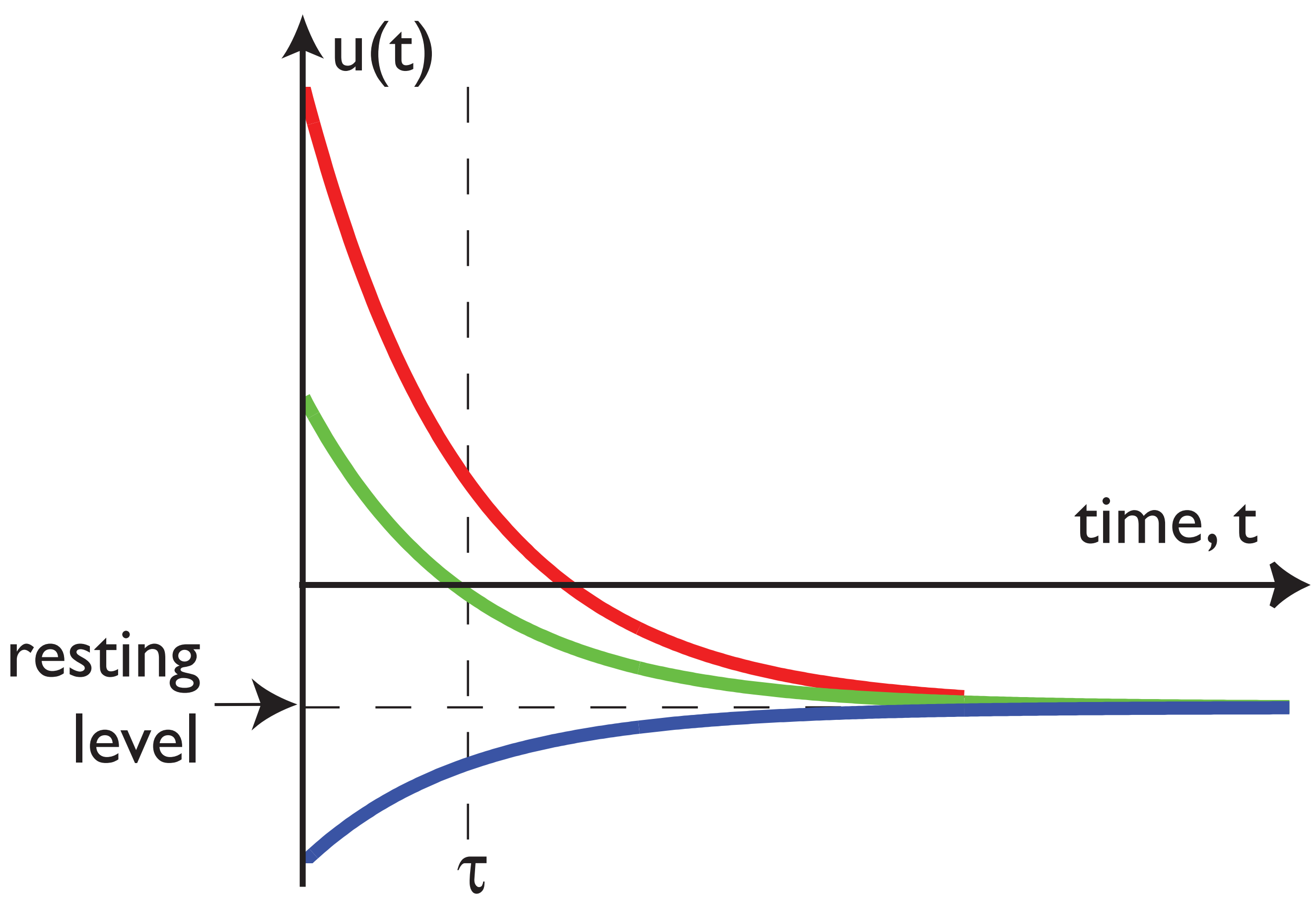
- stationary state=fixed point= constant solution
- stable fixed point: nearby solutions converge to the fixed point=attractor

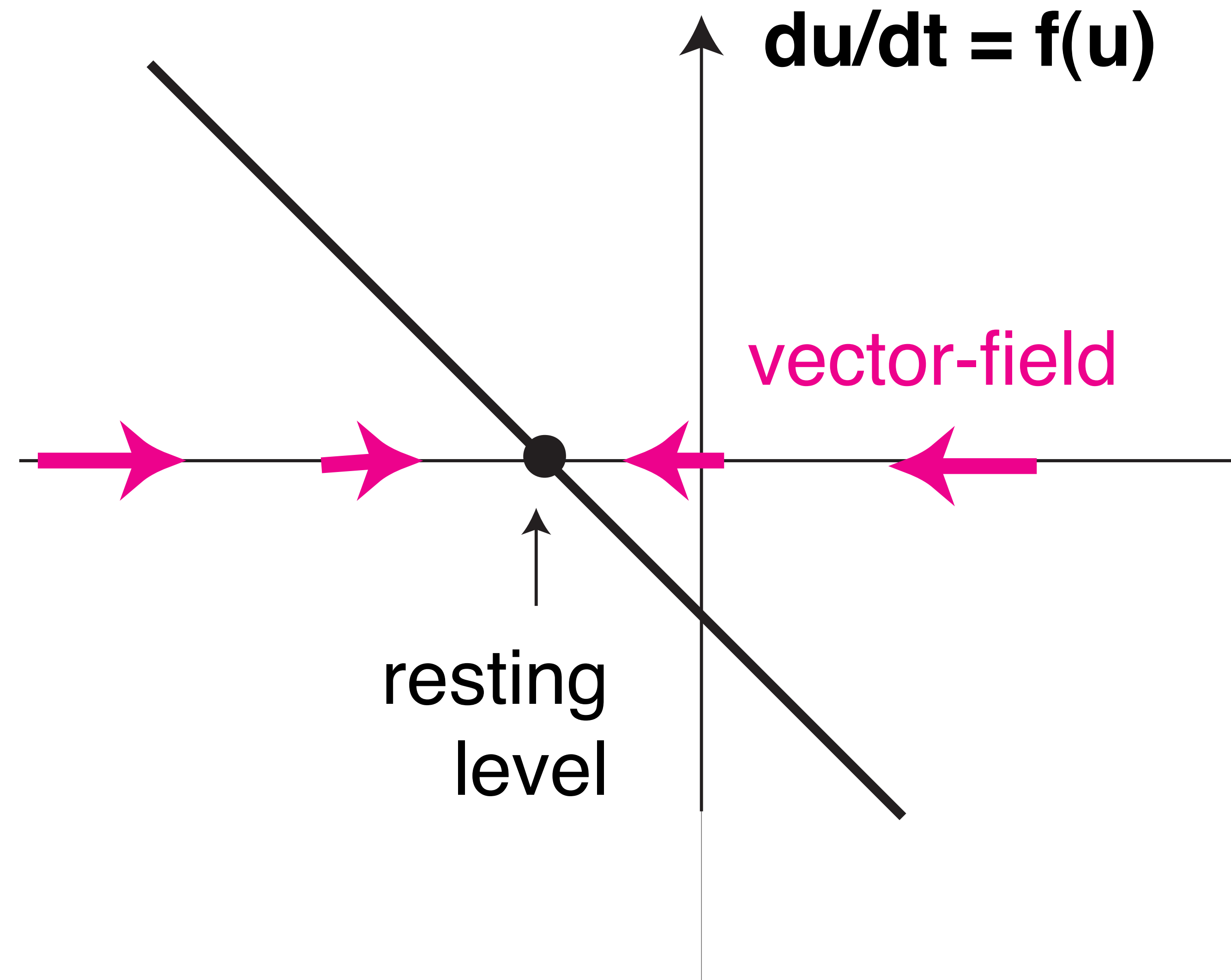


$$\frac{du(t)}{dt} = \dot{u}(t) = -u(t) + h \qquad (h < 0)$$

Neural dynamics

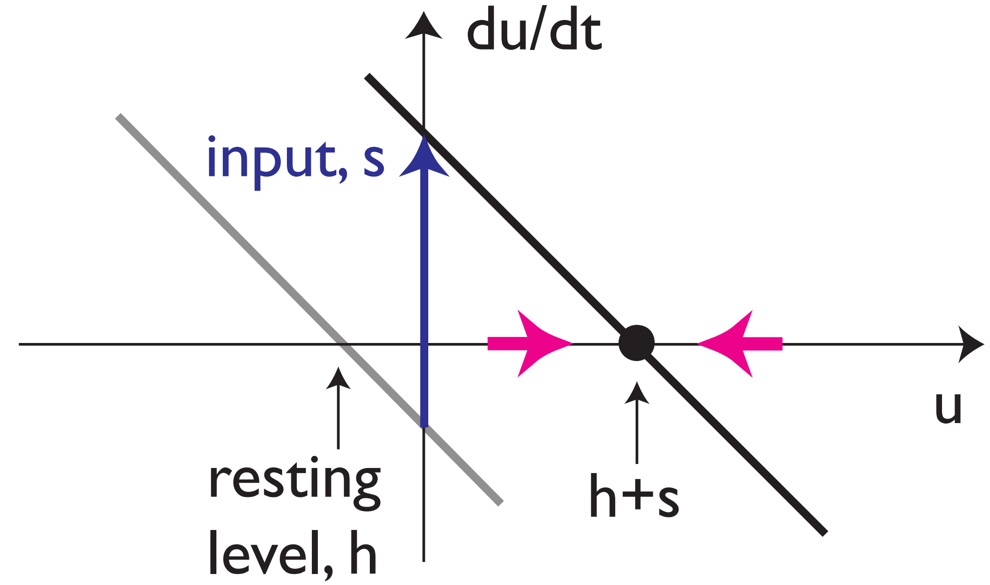
- attractor structures ensemble of solutions=flow



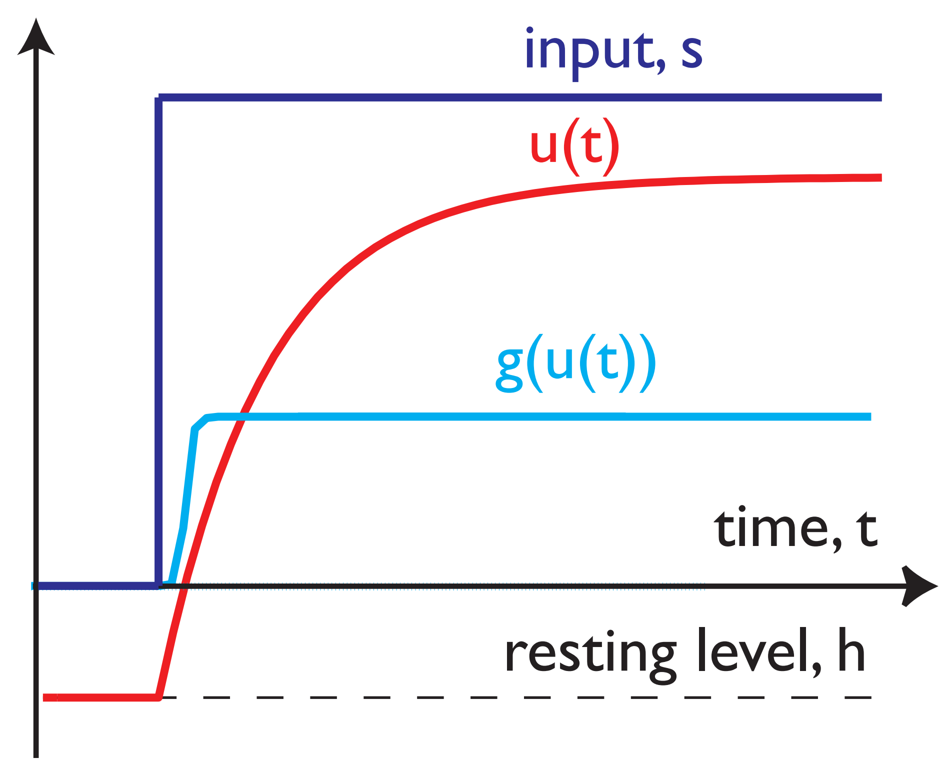


$$\tau \dot{u}(t) = -u(t) + h$$

Neuronal dynamics



- inputs=contributions to the rate of change
  - positive: excitatory
  - negative: inhibitory
- => shifts the attractor
- activation tracks this shift (stability)



$$\tau \dot{u}(t) = -u(t) + h + \text{ inputs(t)}$$

=> simulation

tutorial on numerics

- dynamical system continuous time

$$\dot{u} = f(u).$$

- differential quotient approximates the derivative in discrete time

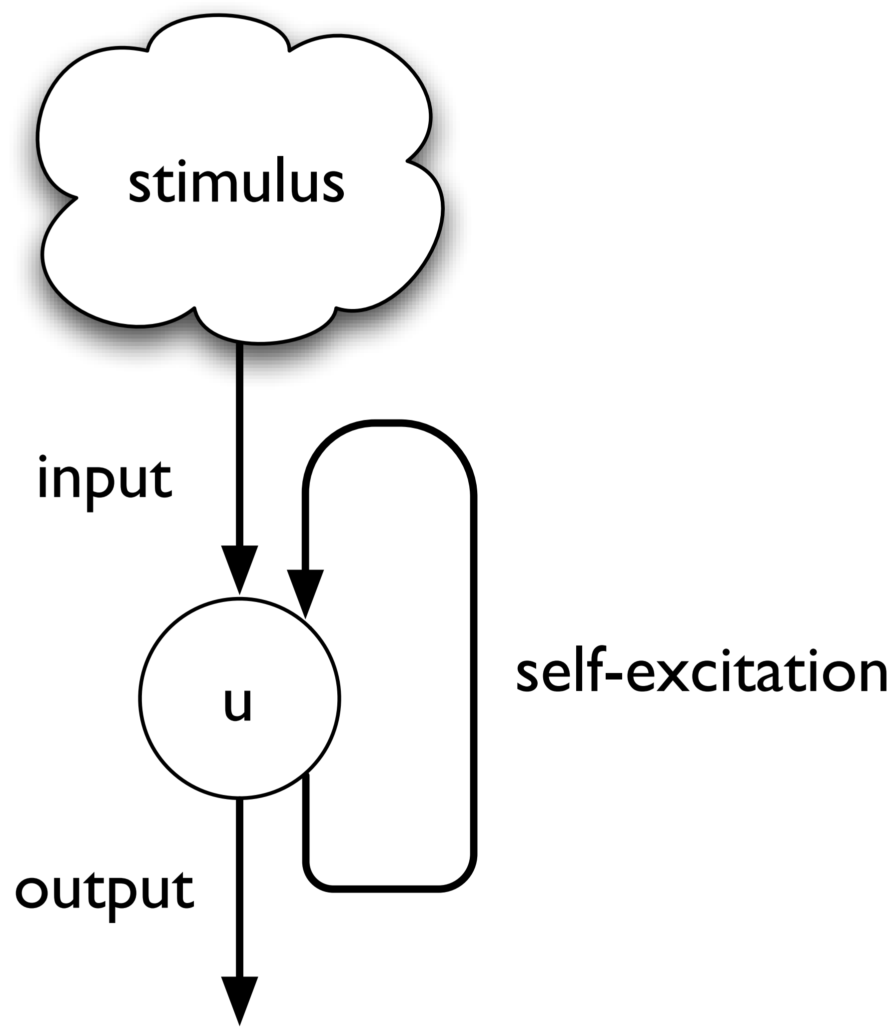
$$\dot{u}(t_i) \approx \frac{u(t_i) - u(t_{i-1})}{\Delta t}$$

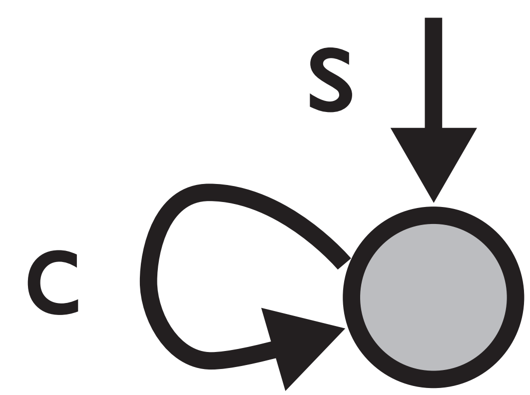
- Euler iteration equation in discrete time

$$u(t_i) = u(t_{i-1}) + \Delta t\, f\left(u(t_{i-1})\right).$$

Matlab code

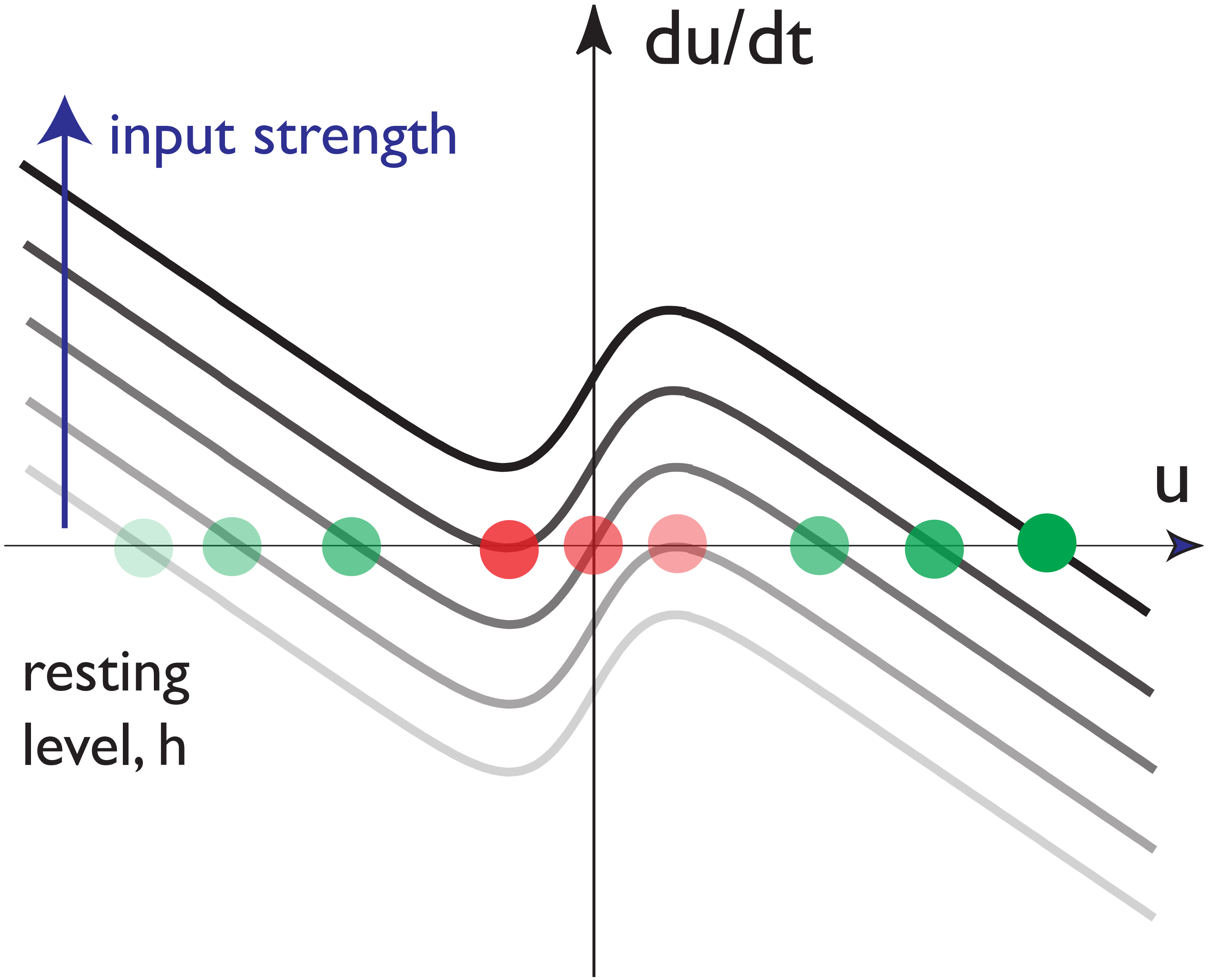
Neuronal dynamics with self-excitation





$$\tau \dot{u}(t) = -u(t) + h + S(t) + c\sigma(u(t))$$

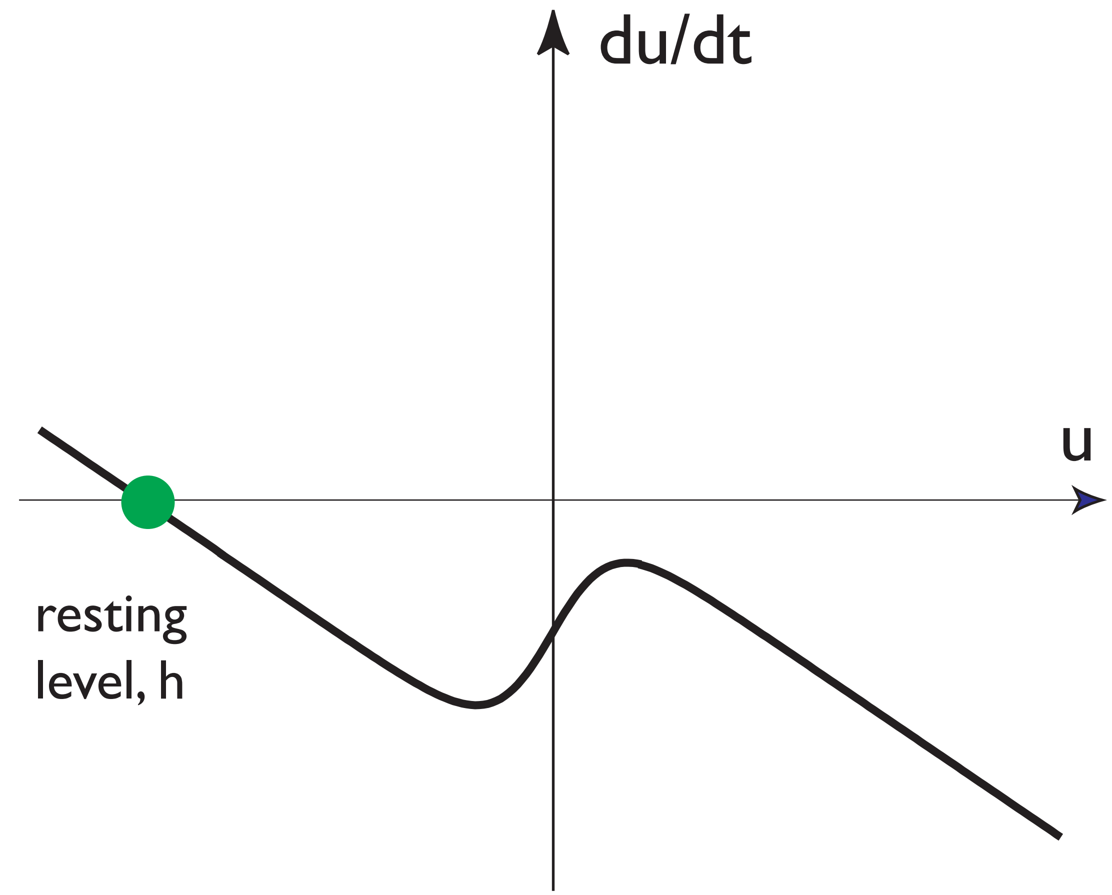
Neuronal dynamics with self-excitation



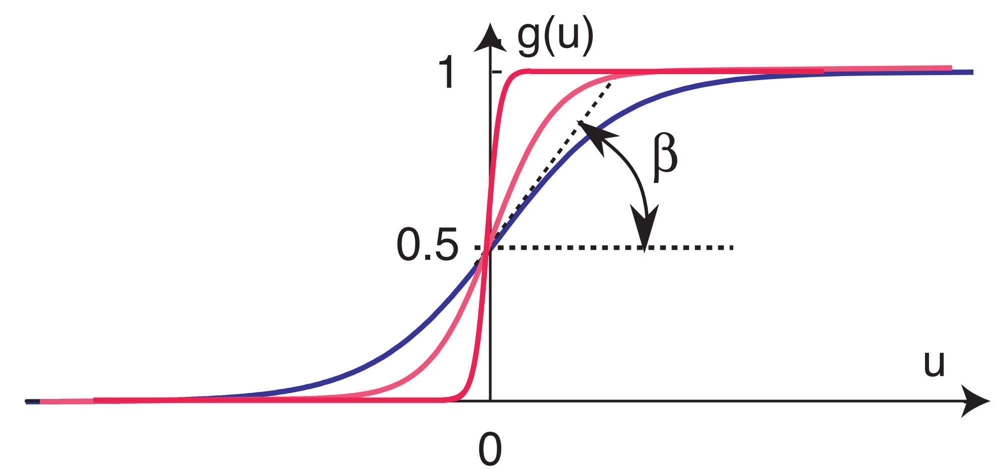
$$\tau \dot{u}(t) = -u(t) + h + S(t) + c\sigma(u(t))$$

Neuronal dynamics with self-excitation

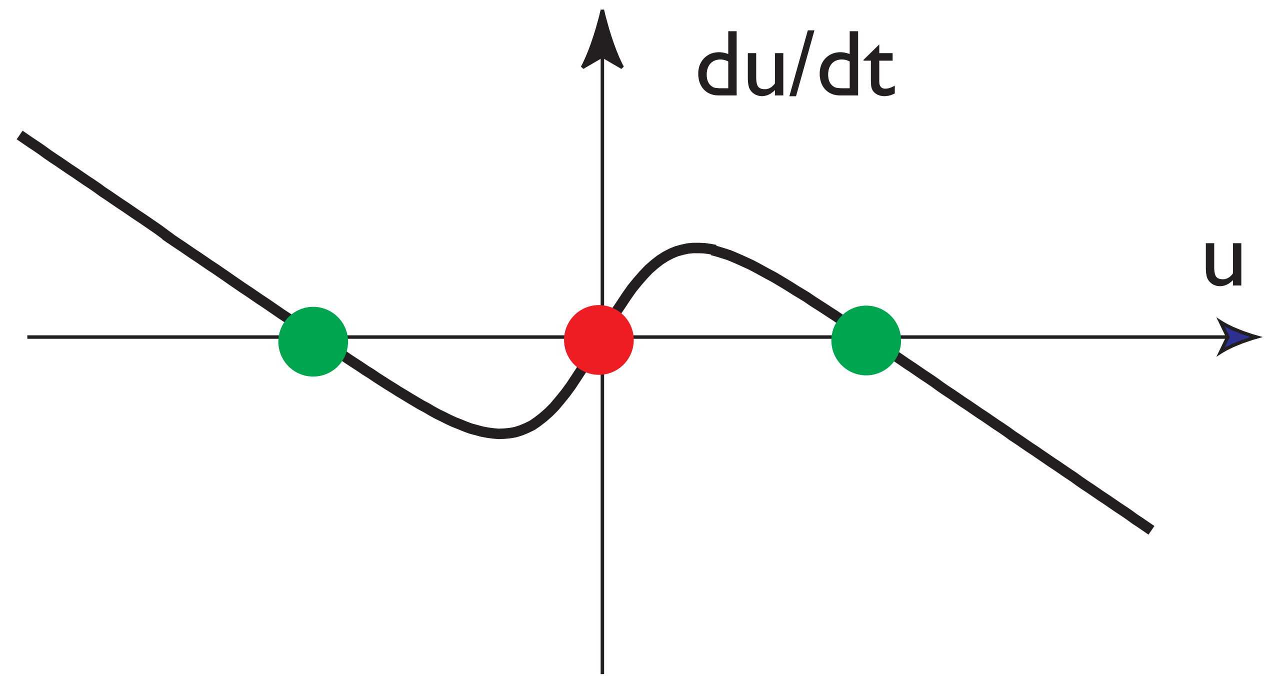
$$\tau \dot{u}(t) = -u(t) + h + S(t) + c\sigma(u(t))$$

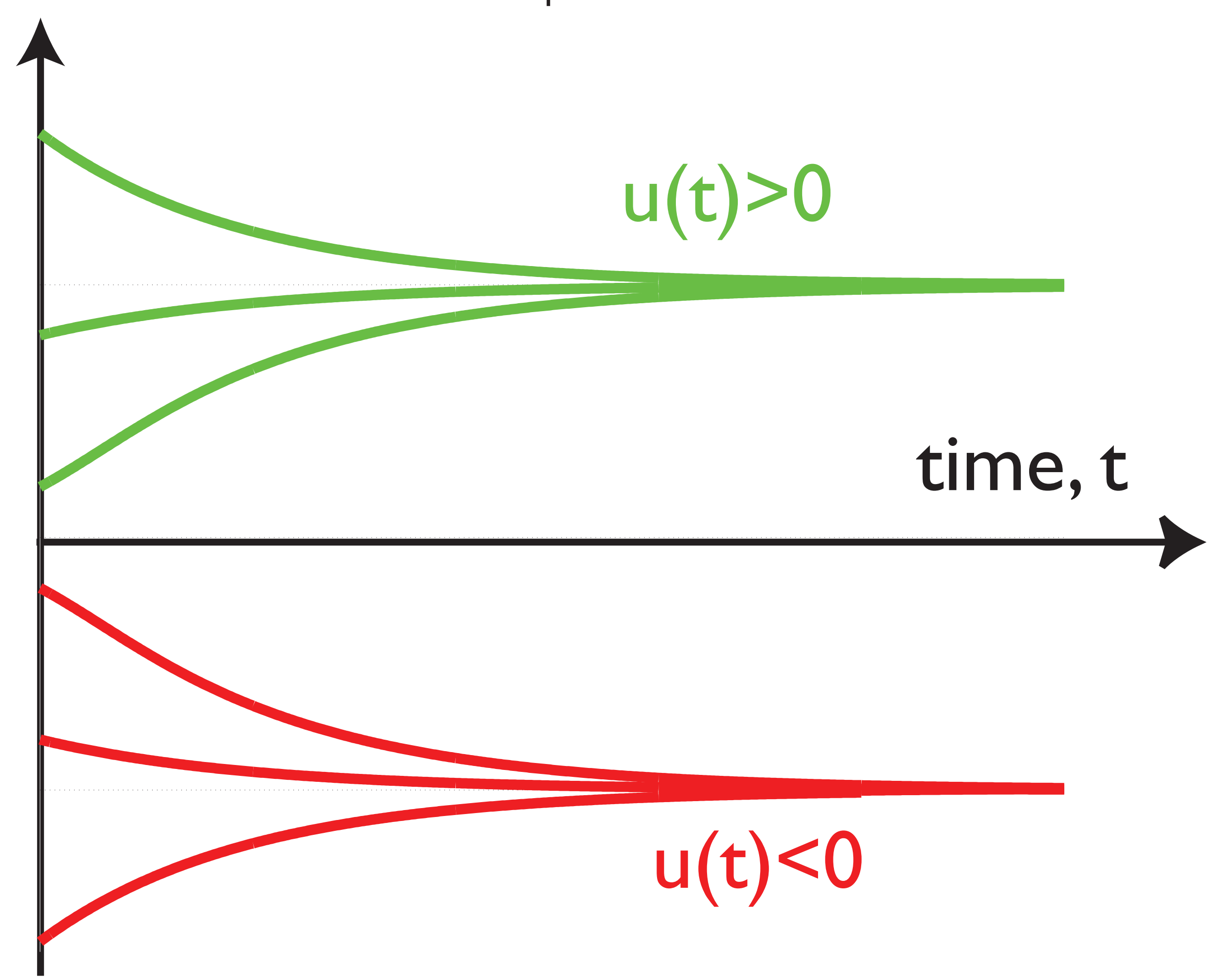


- => nonlinear dynamics!



Neuronal dynamics with self-excitation



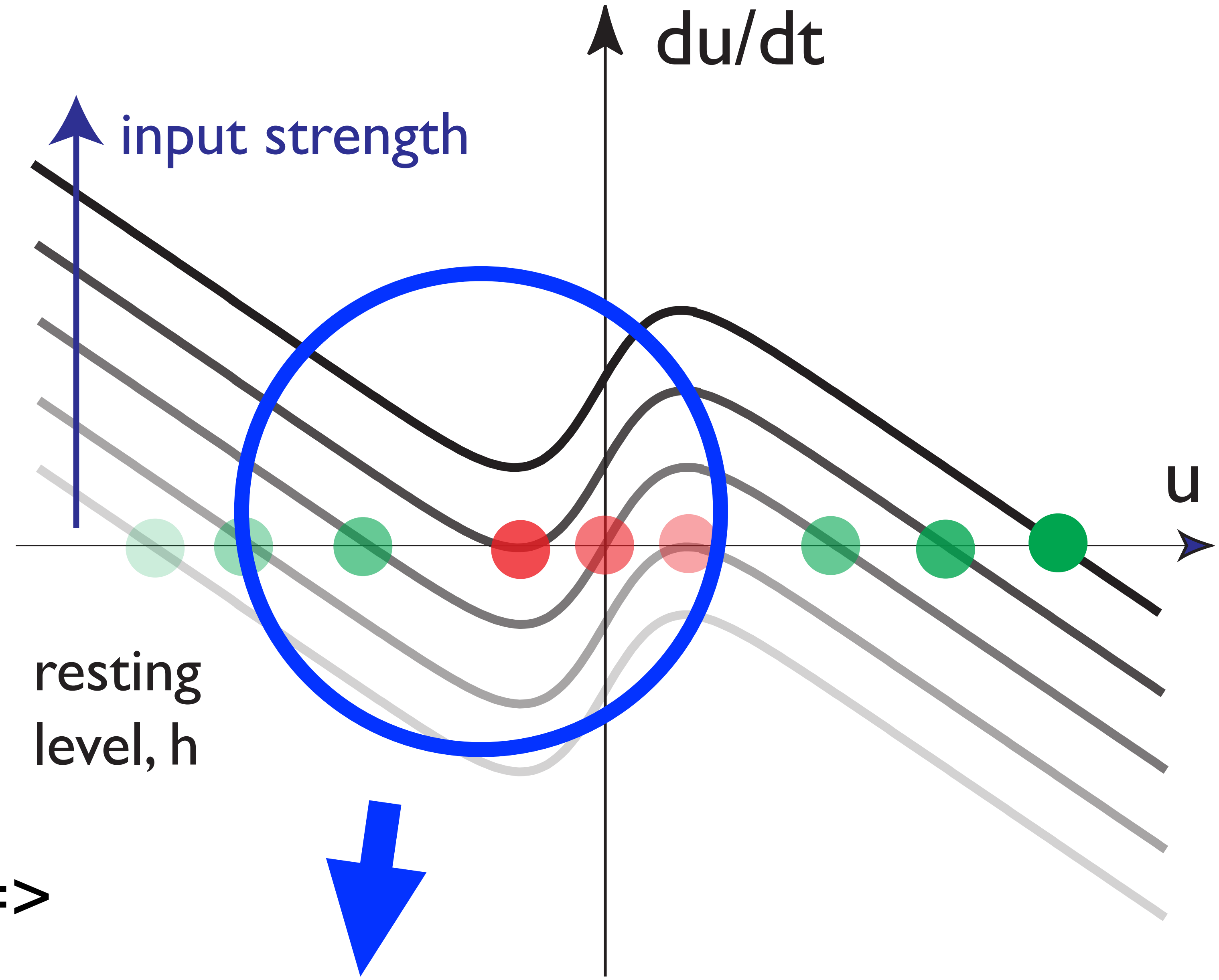


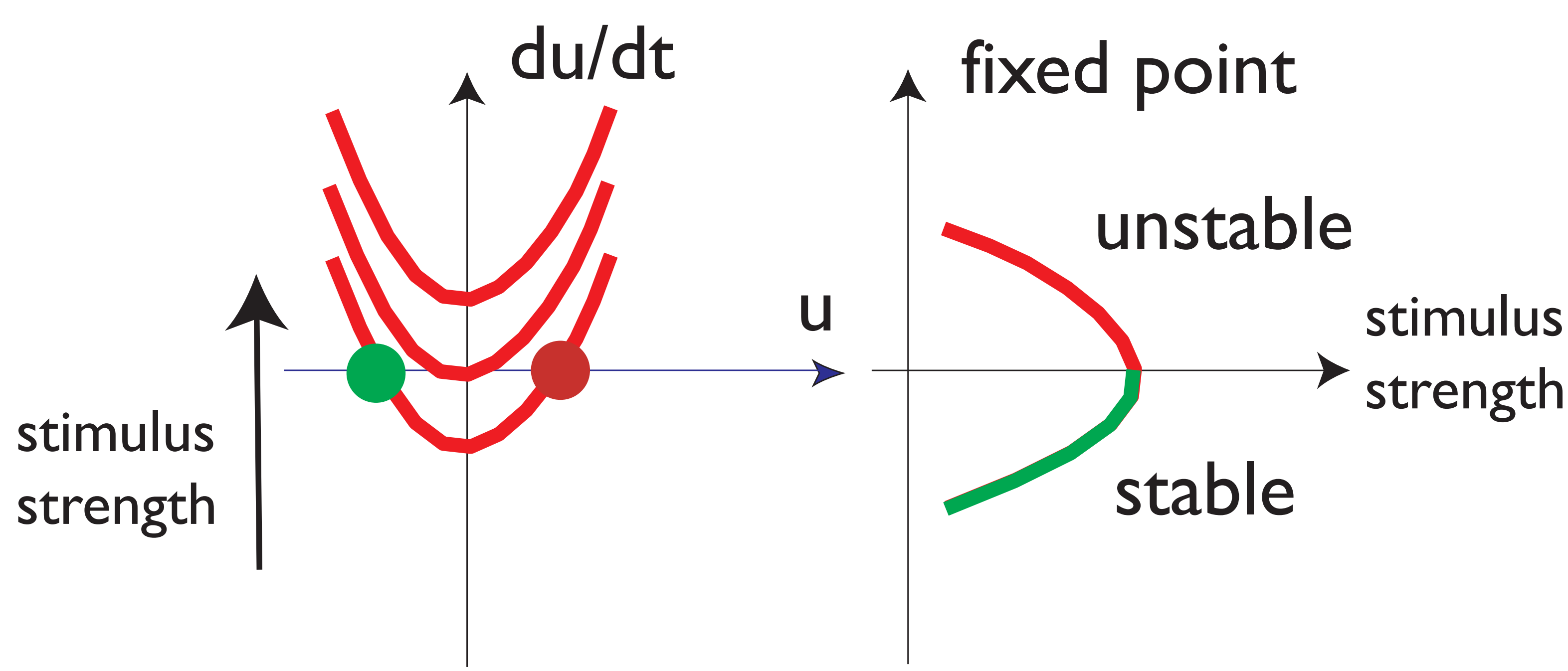
- at intermediate stimulus strength: bistable
- “on” vs “off” state

$$\tau \dot{u}(t) = -u(t) + h + S(t) + c\sigma(u(t))$$

Neuronal dynamics with self-excitation

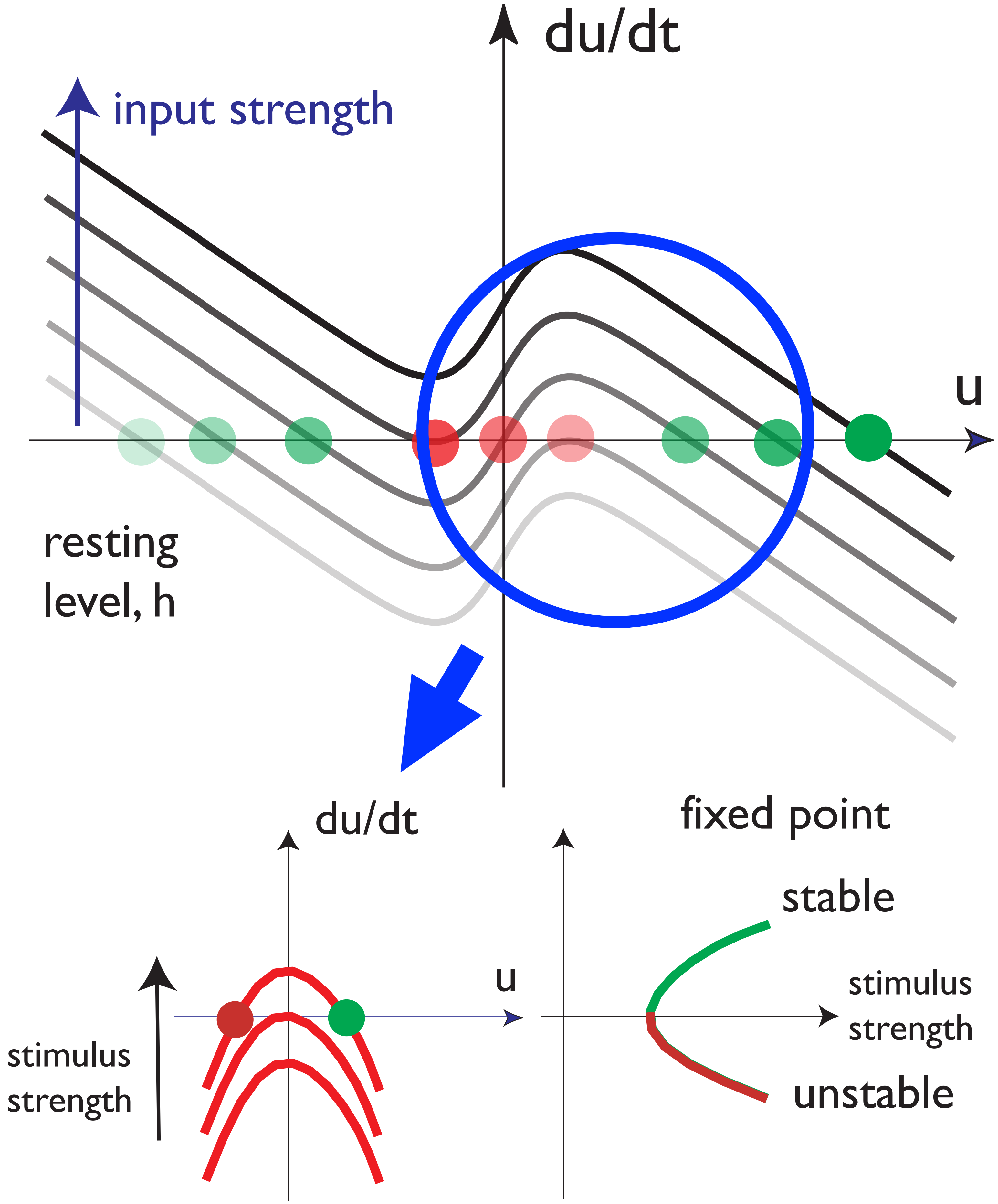
- increasing input strength => detection instability



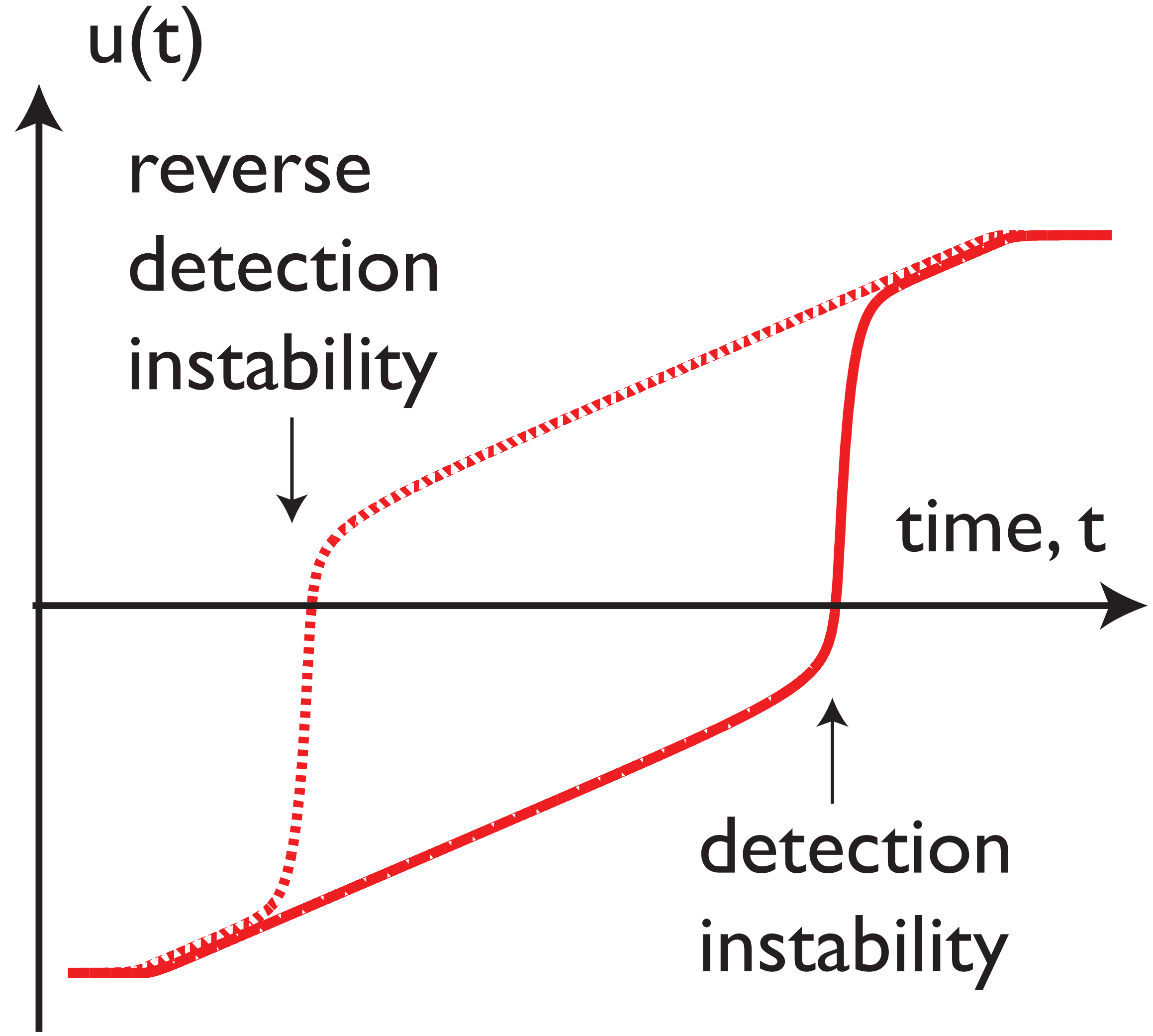


Neuronal dynamics with self-excitation

- decreasing input strength => reverse detection instability



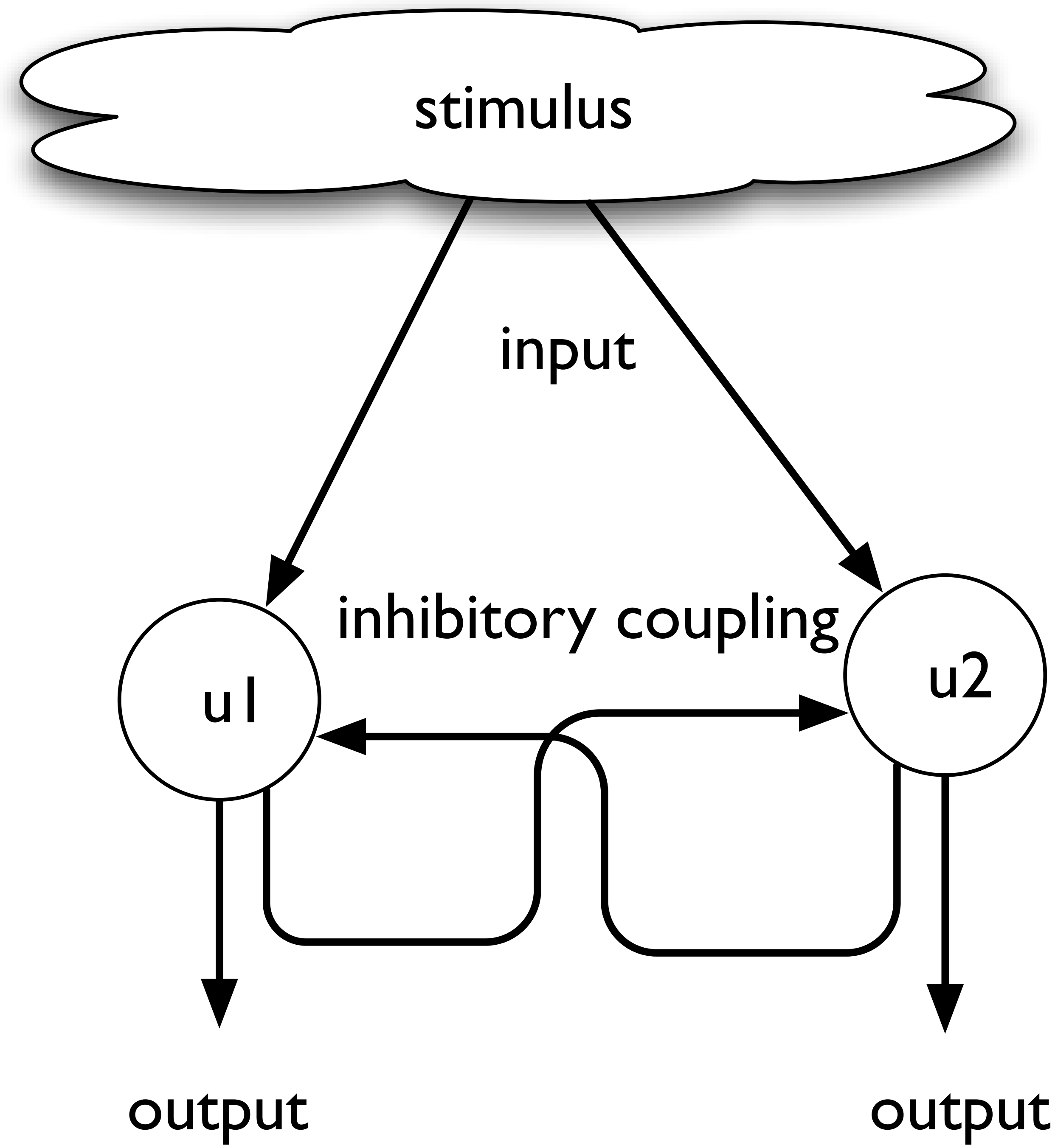
Neuronal dynamics with self-excitation



- the detection and the reverse detection instability create discrete events out of input that changes continuously in time

- => simulation

Neuronal dynamics with competition



$$\tau \dot{u}_1(t) = -u_1(t) + h - \sigma(u_2(t)) + S_1$$
$$\tau \dot{u}_2(t) = -u_2(t) + h - \sigma(u_1(t)) + S_2$$

Neuronal dynamics with competition

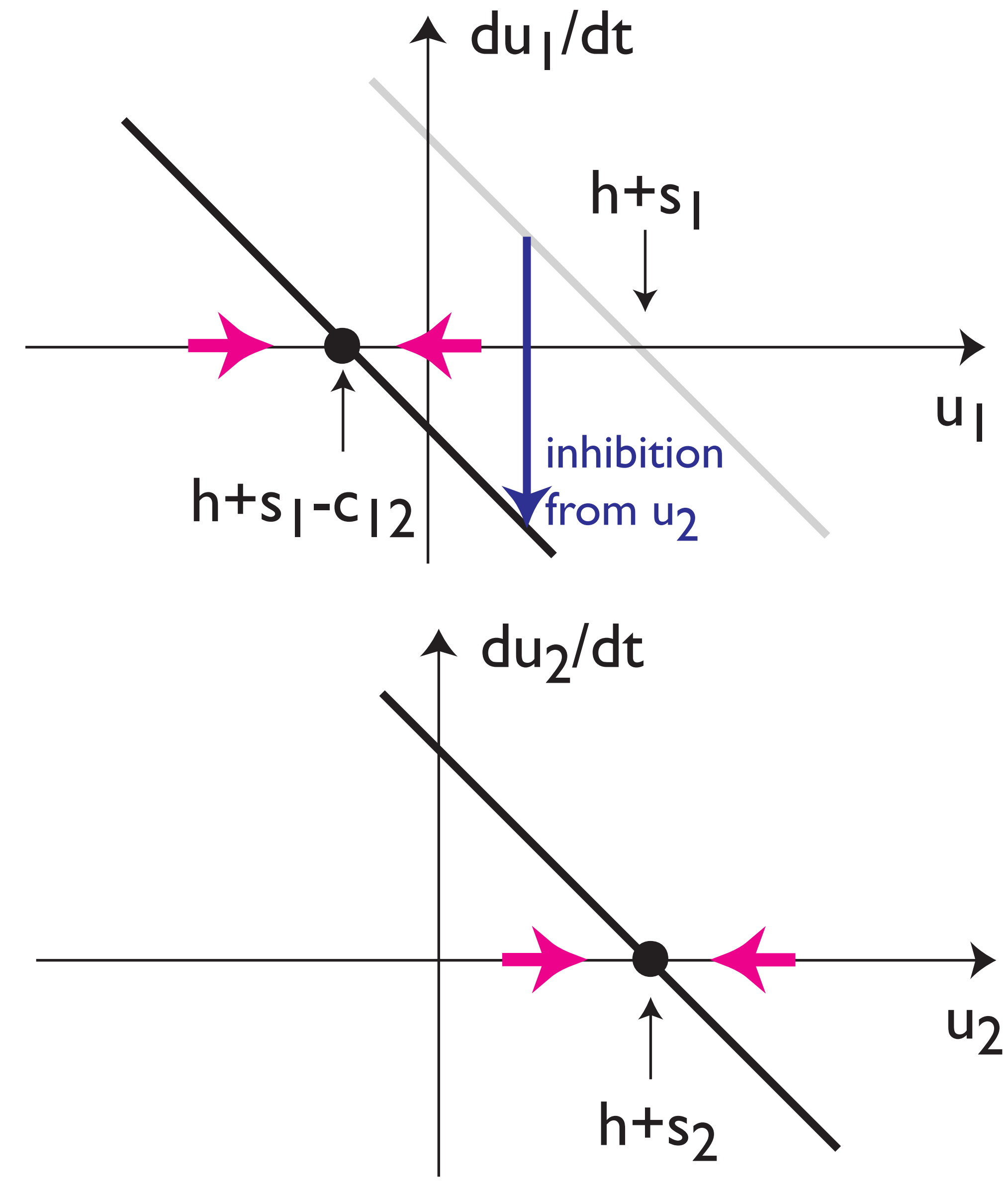
- the rate of change of activation at one site depends on the level of activation at the other site
- mutual inhibition

$$
\begin{aligned}
\tau \dot{u}_1(t) &= -u_1(t) + h - \sigma(u_2(t)) + S_1 \\
\tau \dot{u}_2(t) &= -u_2(t) + h - \sigma(u_1(t)) + S_2
\end{aligned}
$$

sigmoidal nonlinearity

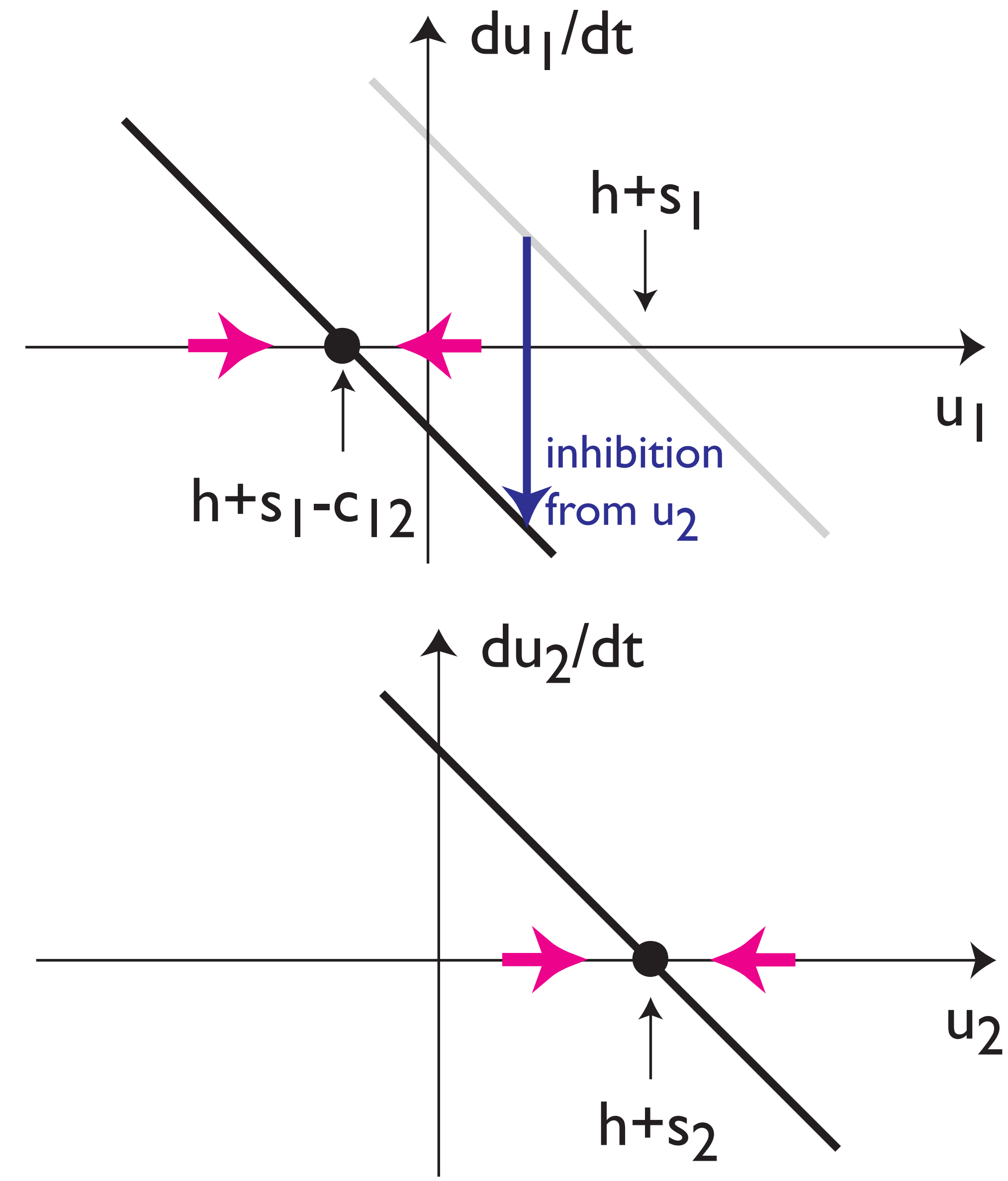
Neuronal dynamics with competition

- to visualize, assume that u_2 has been activated by input to positive level
- => then u_l is suppressed



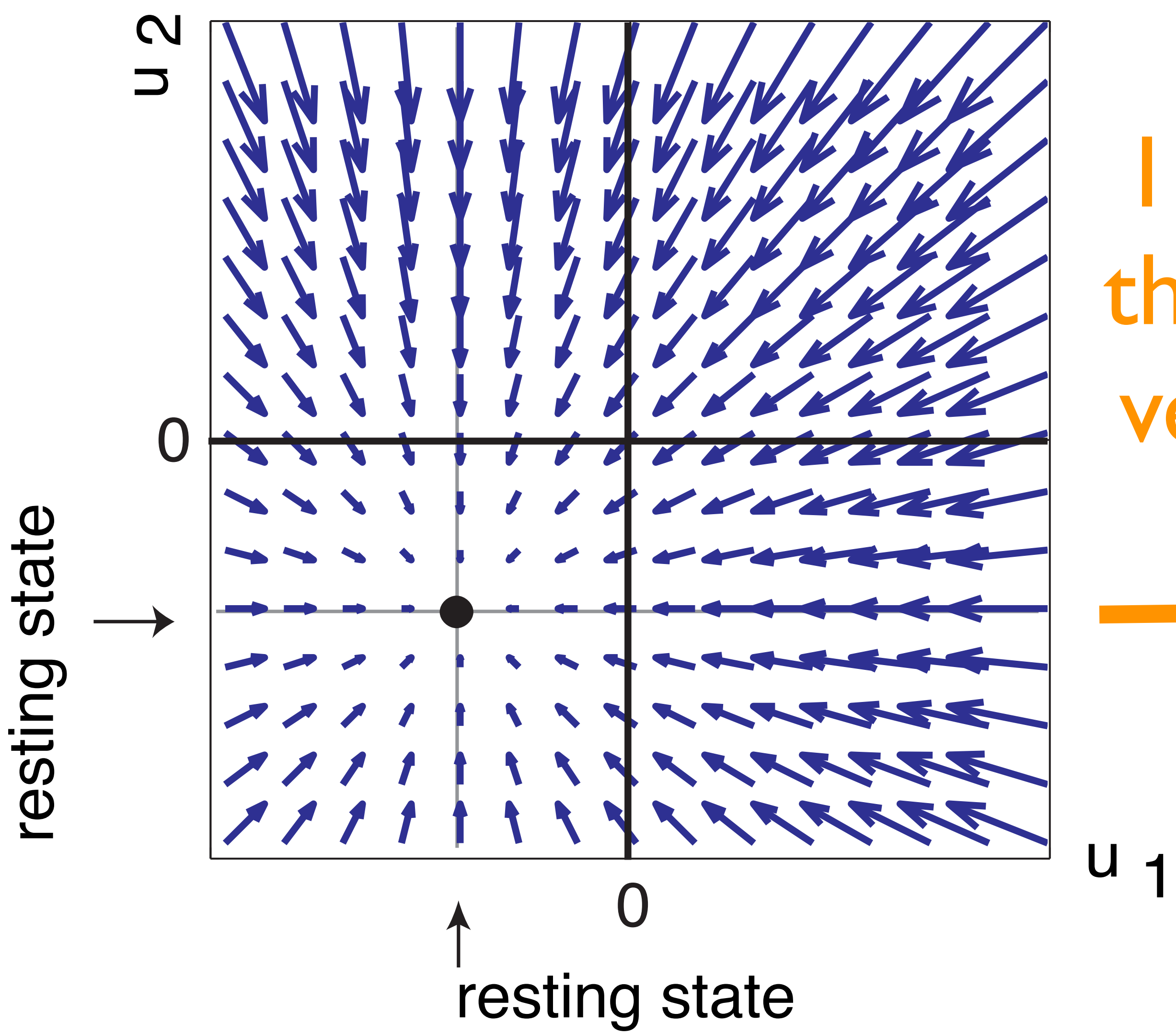
Neuronal dynamics with competition

- why would u_2 be positive before u_1 is? E.g., it grew faster than u_1 because its inputs are stronger/inputs match better
- => input advantage translates into time advantage which translates into competitive advantage

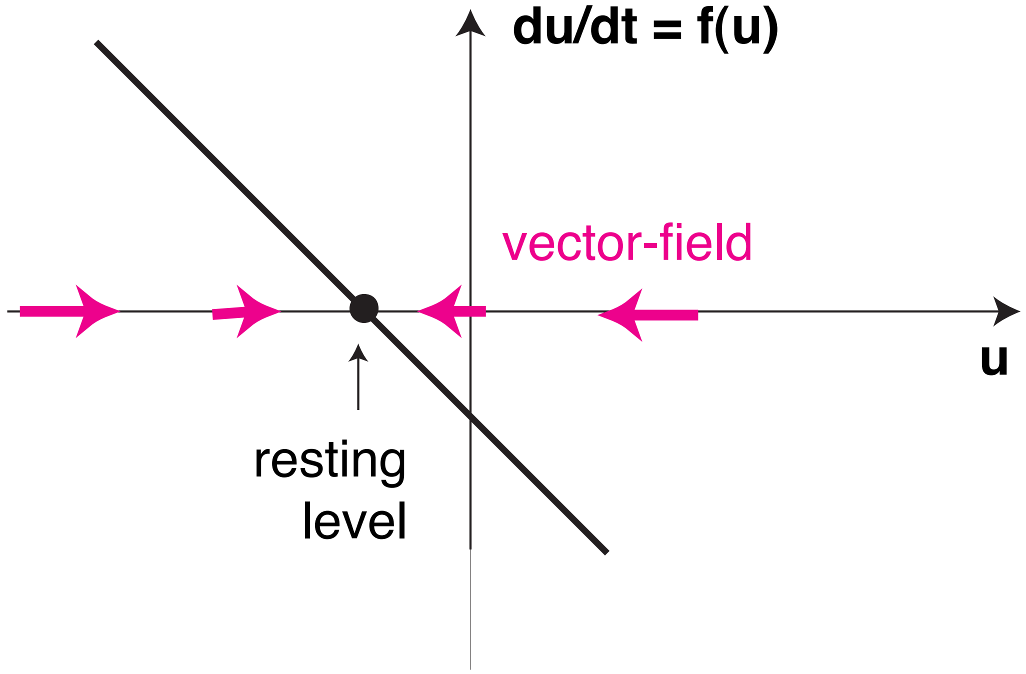


Neuronal dynamics with competition

vector-field in the absence of input

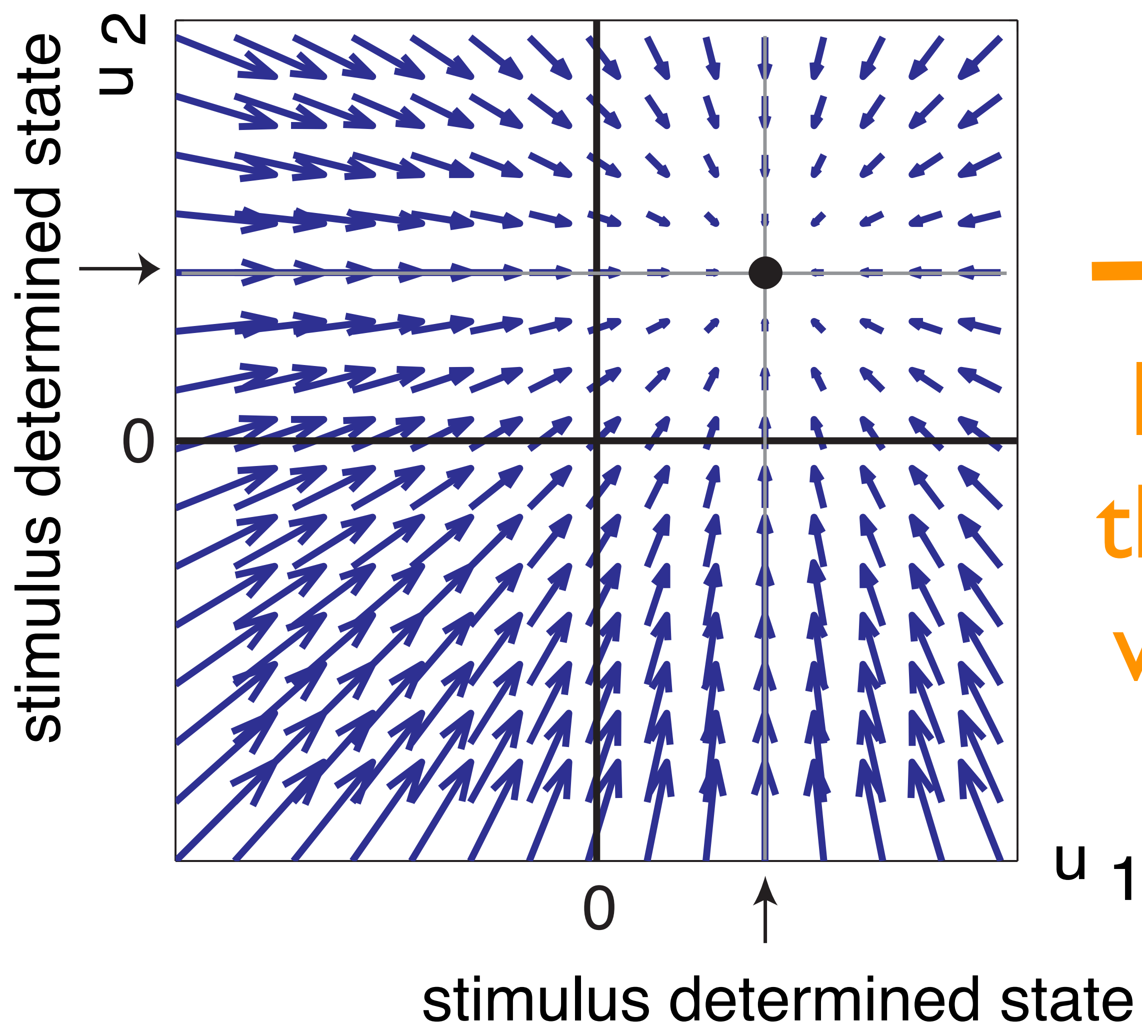


1D cut through vector-field

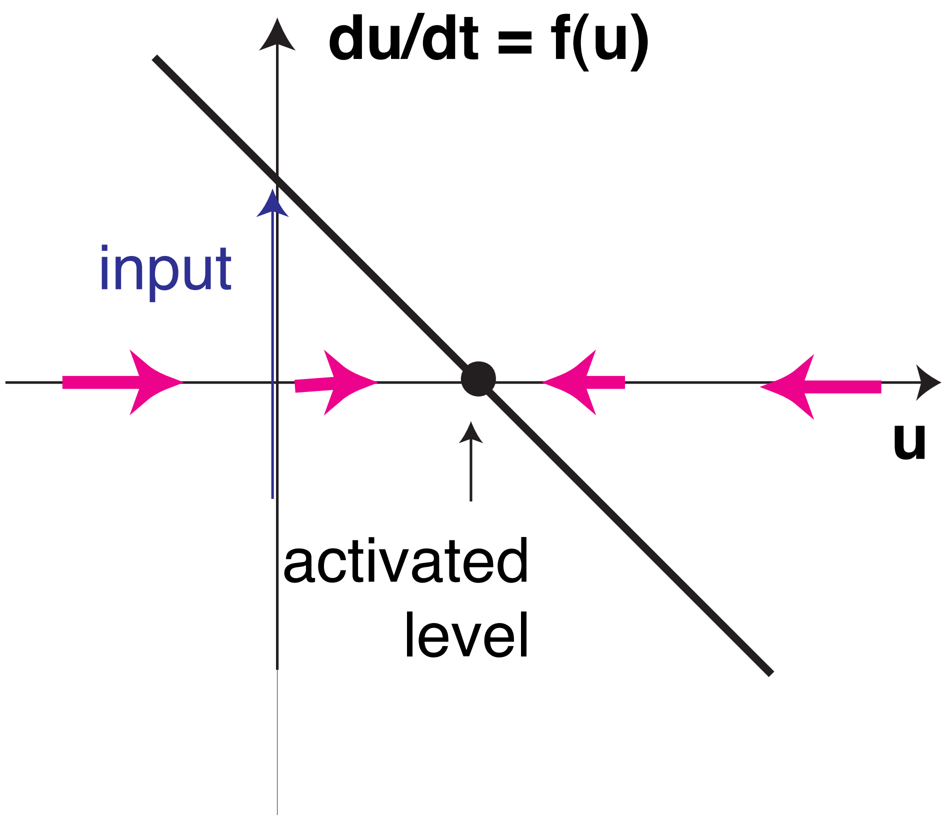


Neuronal dynamics with competition

vector-field (without interaction) when both neurons receive input



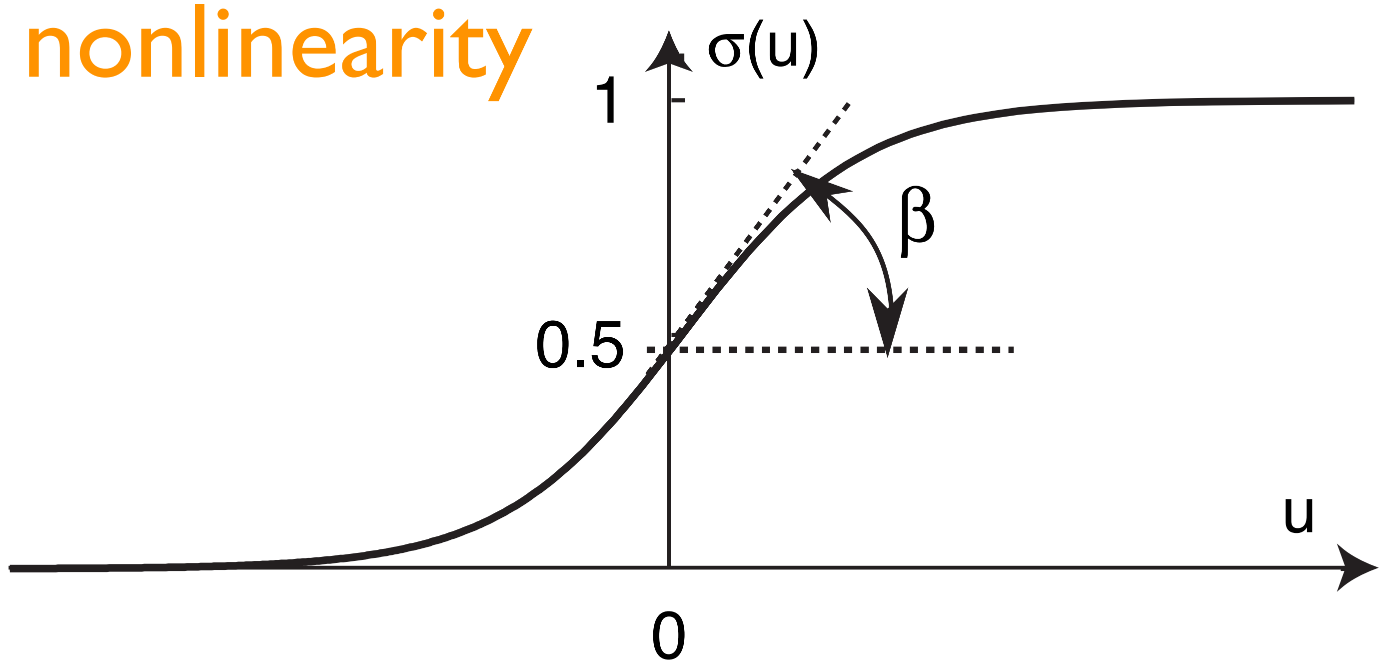
1D cut through vector-field



Neuronal dynamics with competition

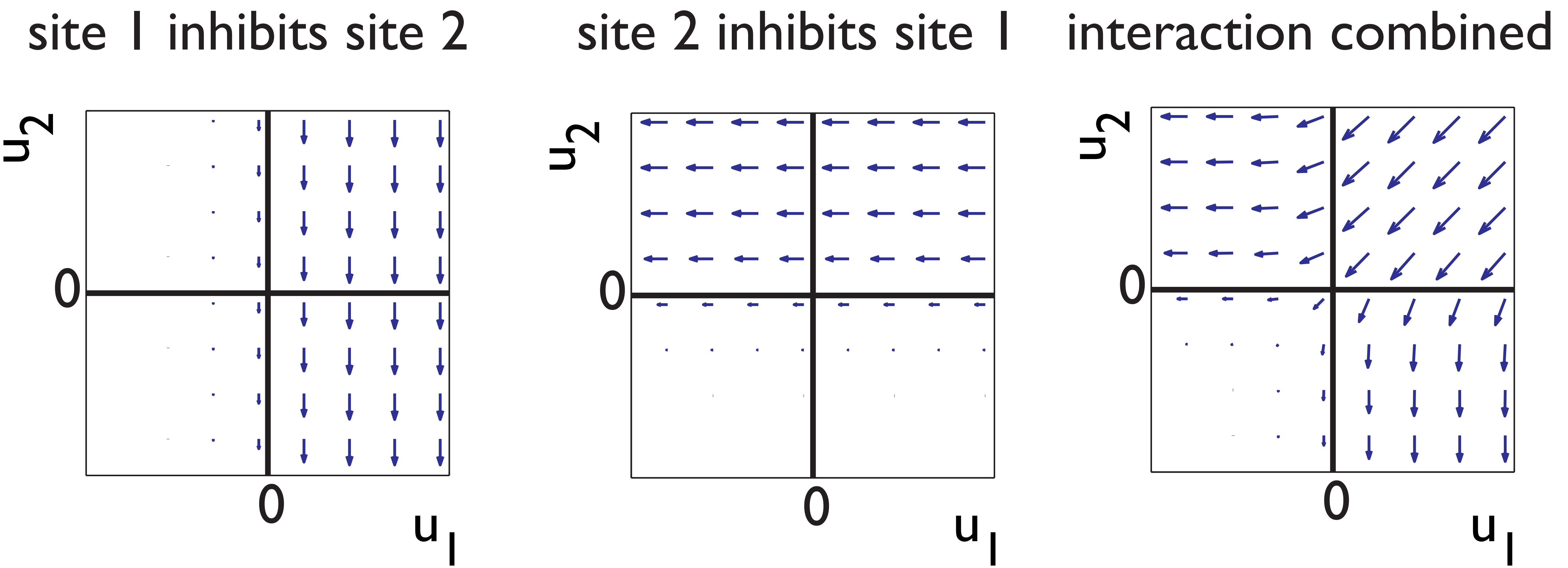
- only activated neurons participate in interaction!

sigmoidal nonlinearity



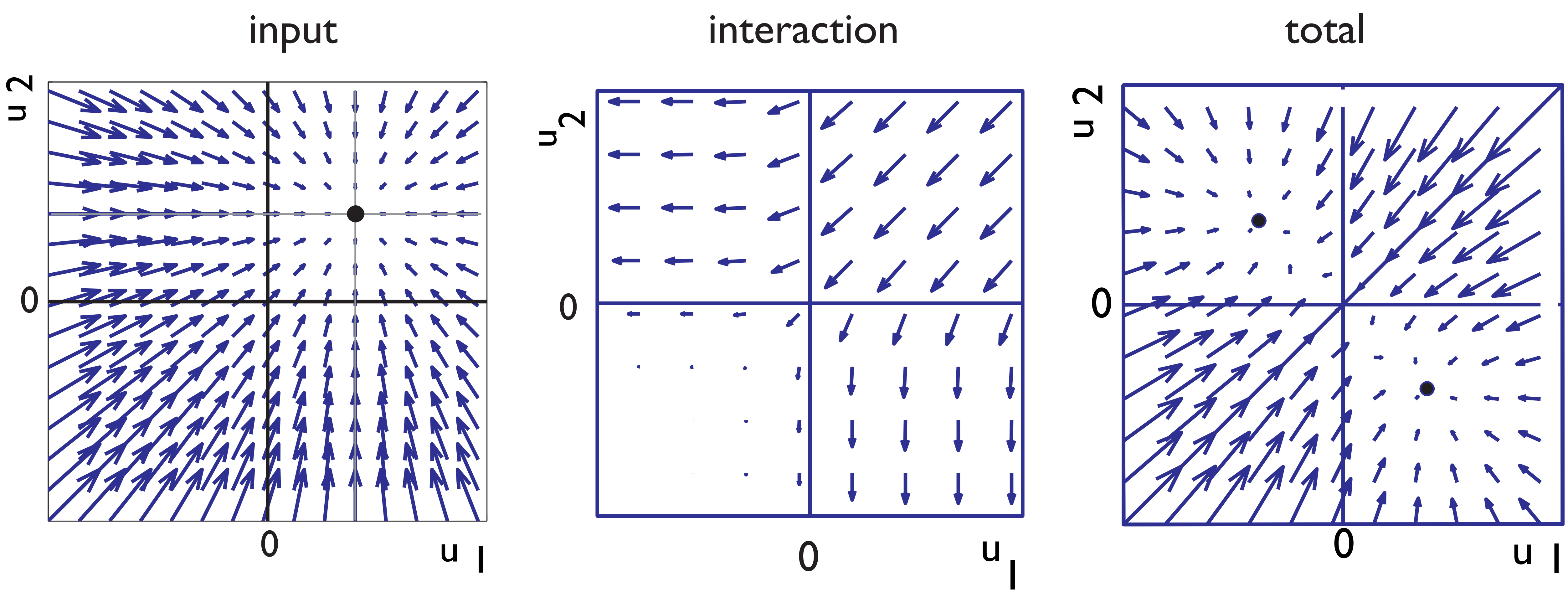
Neuronal dynamics with competition

- vector-field of mutual inhibition



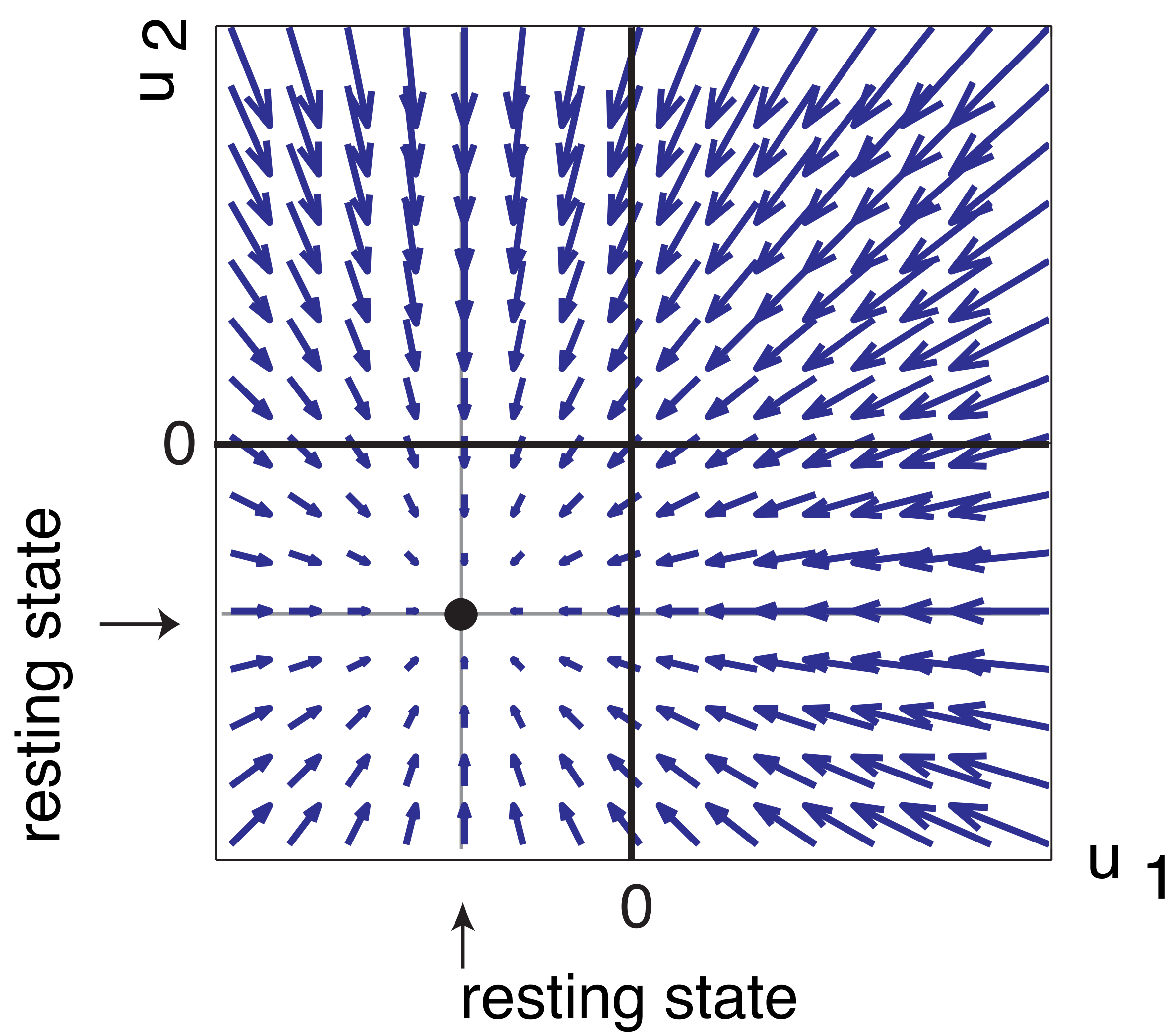
Neuronal dynamics with competition

vector-field with strong
mutual inhibition:
bistable

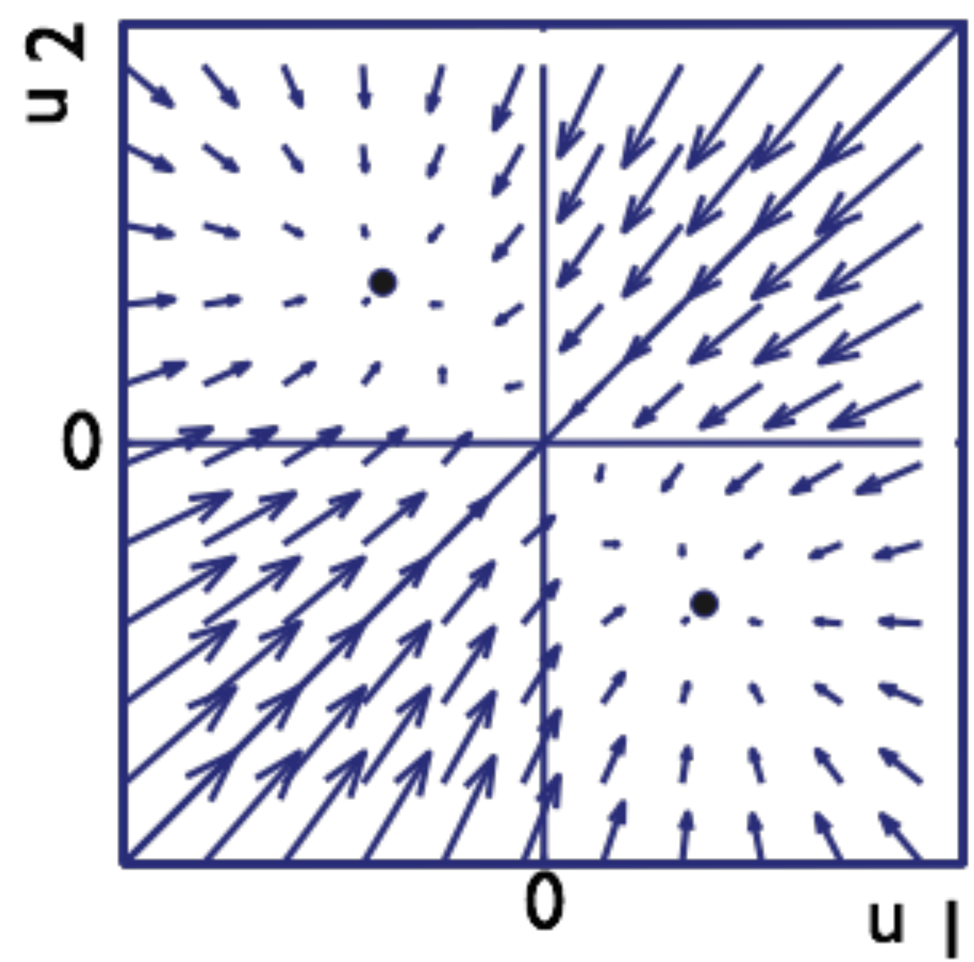


Neuronal dynamics with competition

before input is presented



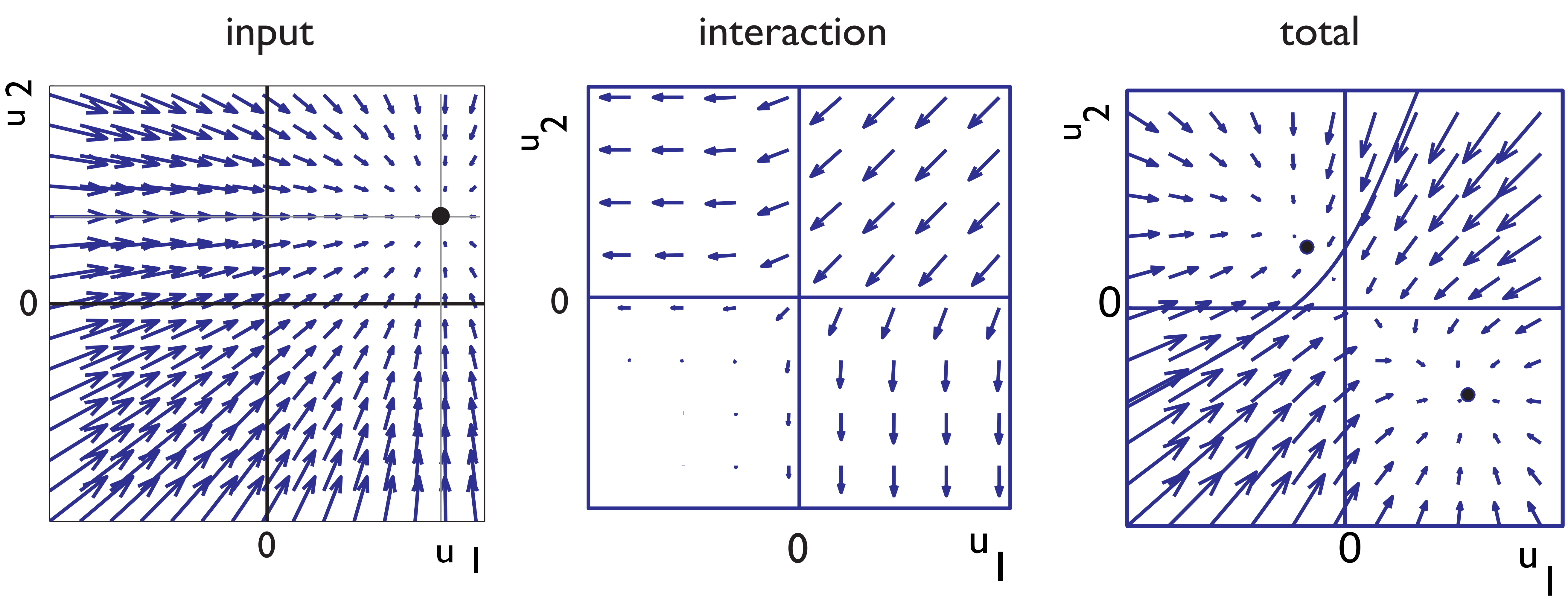
after input is presented



Neuronal dynamics with competition

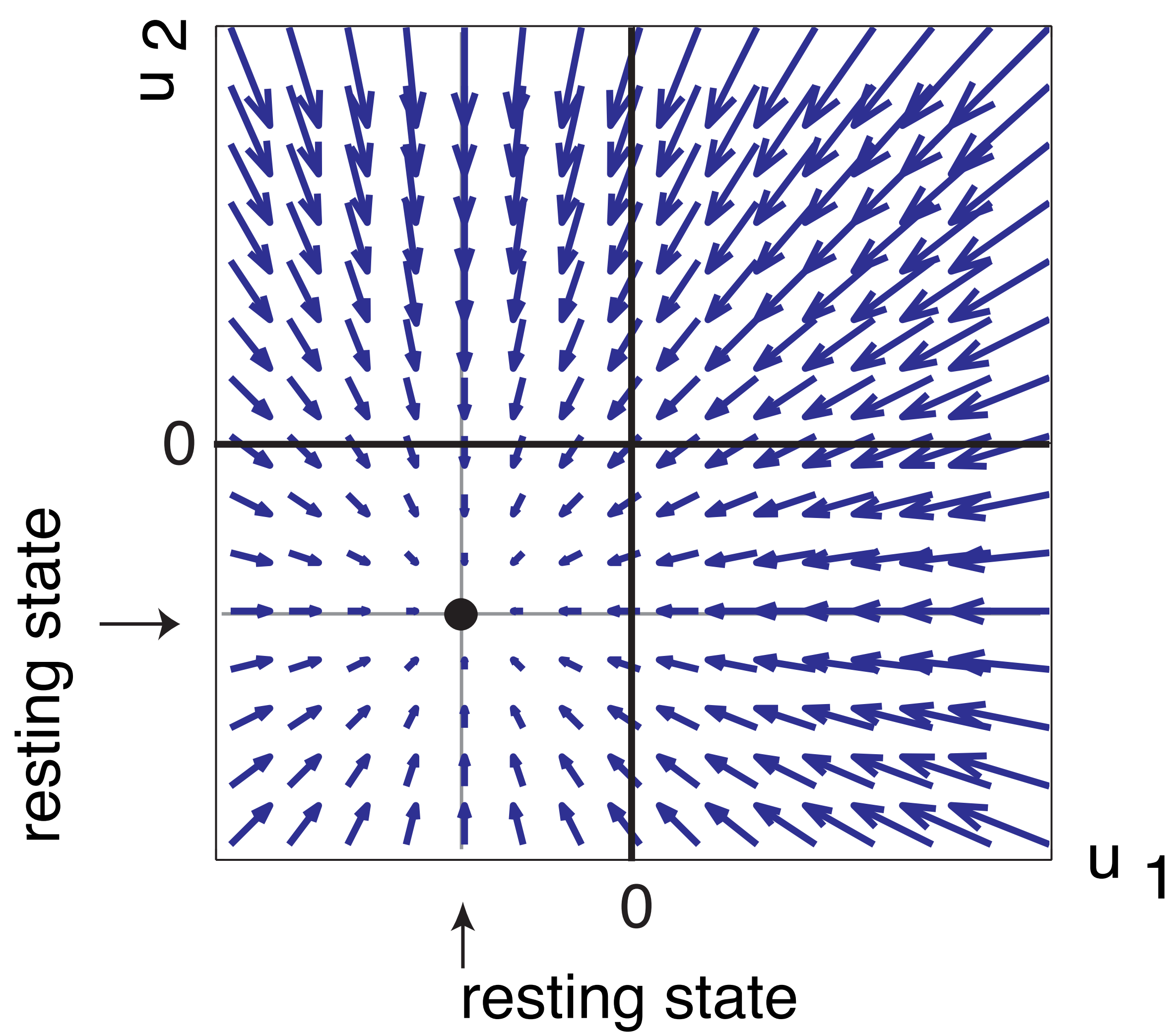
=>biased competition

stronger input to site 1:
attractor with activated u_1 stronger,
attractor with activated u_2 weaker, may become unstable

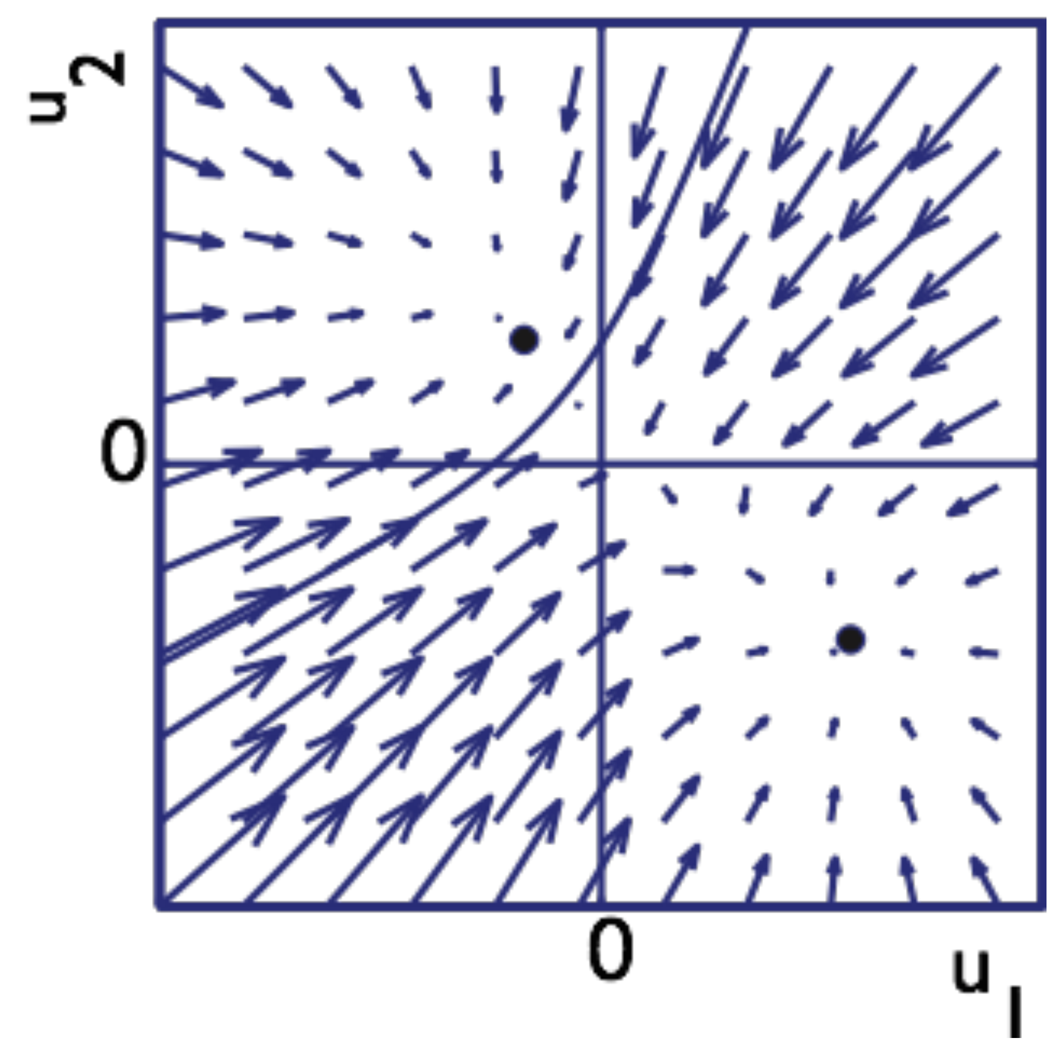


Neuronal dynamics with competition =>biased competition

before input is presented



after input is presented



=> simulation

- => hands-on exercise NOW
- in the robotics lab..

next

- where do activation variables come from?
- => DFT lecture