PROGRAMMING NEUROGRID I
THE PYTHON WAY

BEN VARKEY BENJAMIN

PEIRAN GAO

KWABENA BOAHEN

Practical Stuff

- One Neurogrid system will be available for class with at least 16 chips--a million neurons!
- Access to Neurogrid will be time-shared through a sign up website after next Wednesday
- Available time slots depending on lab members' schedule;
- Both hardware and software are under constant state of development:
  - Expect frustrating bugs and crashes
  - TA and other lab members will be there to help
- Goal is to make the experience *mutually* beneficial for you and the lab

Topics for Today:

- Synapse, Arbor, Soma and Neuron Models
- Hierarchical network constructs
- Making vertical (i.e., topographic) connections
- Mapping network to hardware
- Saving data from hardware (basic)

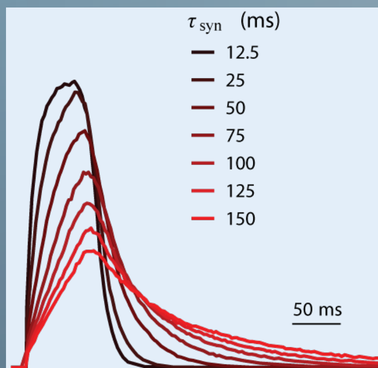
Topics for Next Time:

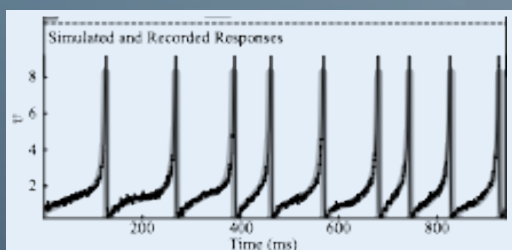
- Making horizontal connections
- Sending stimulus
- Running experiments
- Possibly more....

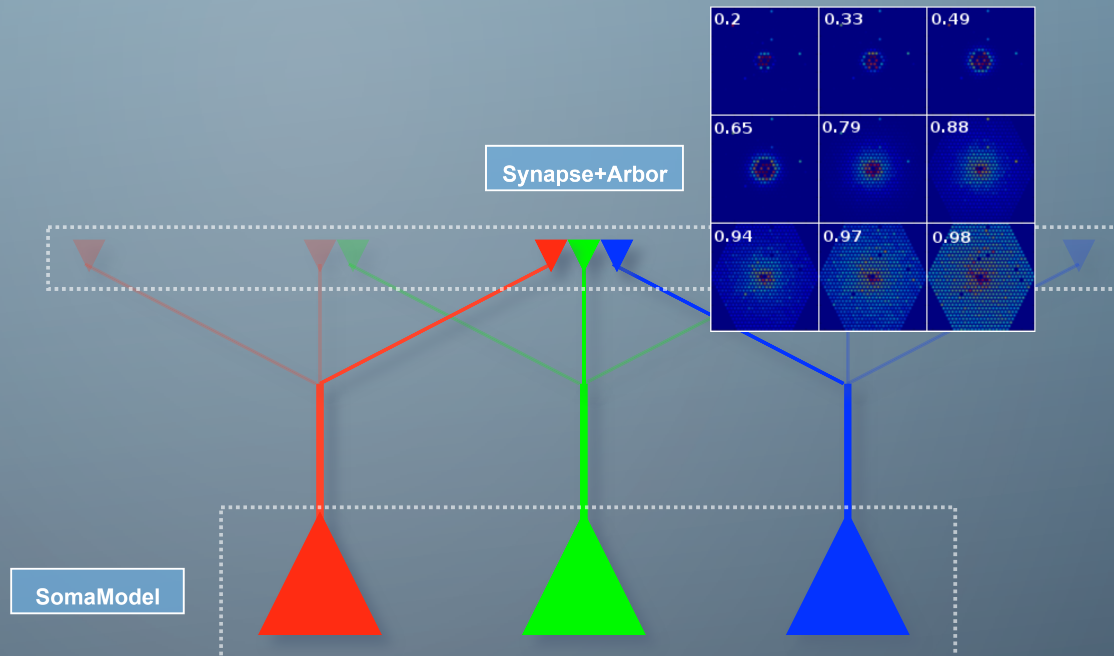
From NEST to Neurogrid

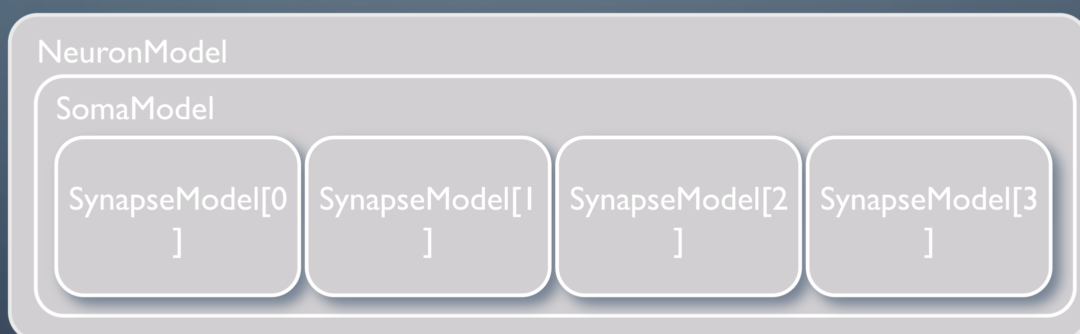
	NEST	Neurogrid
Models	Identical	Heterogeneous
Network	Flat, small	Hierarchical, large
Speed	Size-dependent	Real-time

Hierarchical NeuronModel





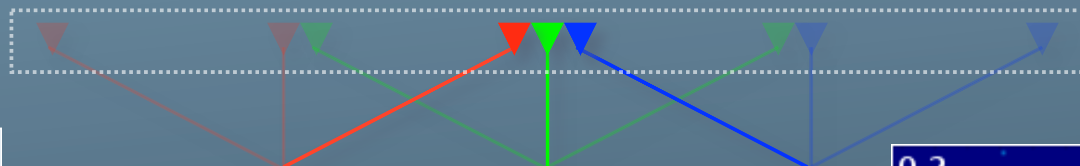




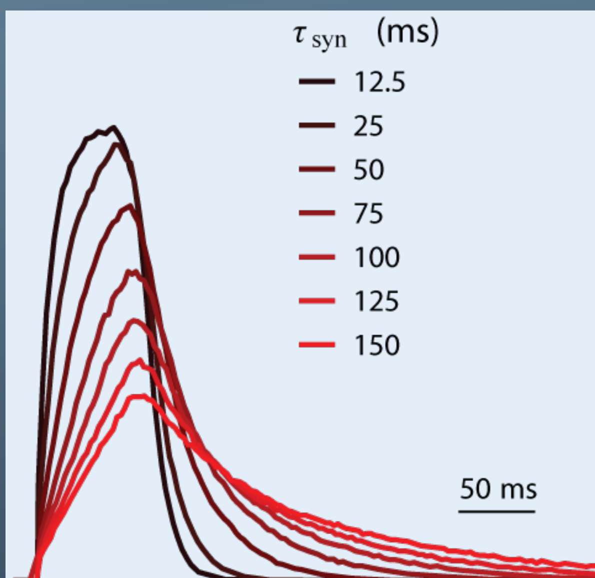
Building Synapse and Arbor

```
>>> syn = Synapse(“cond_syn”, {“erev”: .1, “g_max”: 40, “tau_syn”: 2e-3, “t_xmt”: 5e-3, “lambda”: .2)
```

Default parameter (shown) used if omitted

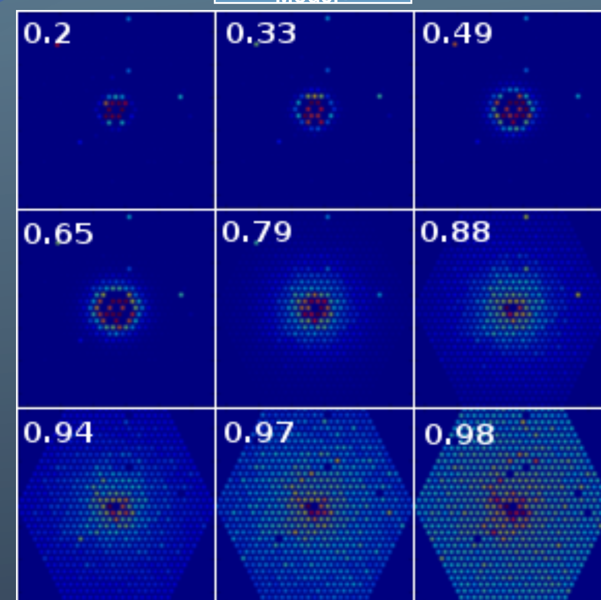


SynapseModel



Inhibitory: 0<erev<.5
Shunting: .5<erev<1.5
Excitatory: 1.5<erev
Default: erev=0.1

SynapseArbor Model+



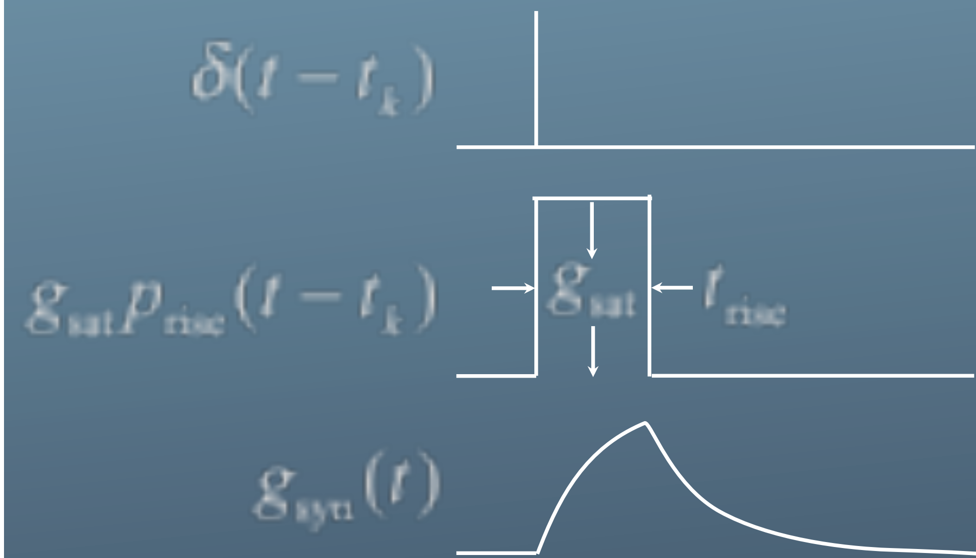
$$I_{syn} = g_{syn}(t)(E_{rev} - V_m)$$

$$\tau_{syn}\dot{g}(t) = -g(t) + \sum_i pulse(t - t_i)$$

$$w(r) = \frac{1}{Z}\frac{\lambda^r}{\sqrt{r}}, \quad 0 < \lambda < 1$$

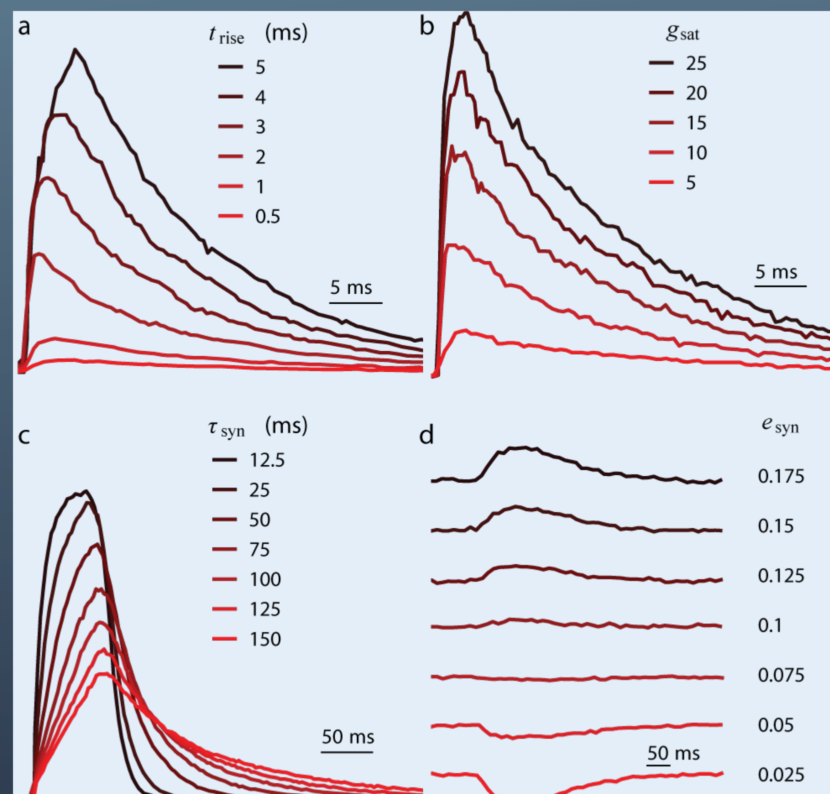
Normalized such that total conductance equals $g_{syn}(t)$

SYNAPSE DYNAMICS



$$\tau \dot{g}_{syn} = -g_{syn} + g_{sat} p_{rise}(t)$$

- The cleft sets $g_{syn}(t)$'s rise time, t_{rise}, by producing a unit pulse, $p_{rise}(t)$, with width t_{rise}.
- The receptor sets $g_{syn}(t)$'s steady-state value, g_{sat}, by scaling $p_{rise}(t)$, and its time constant, τ_{syn}.



Ben's data

Building Soma and Neuron

Default parameters (shown) used if omitted

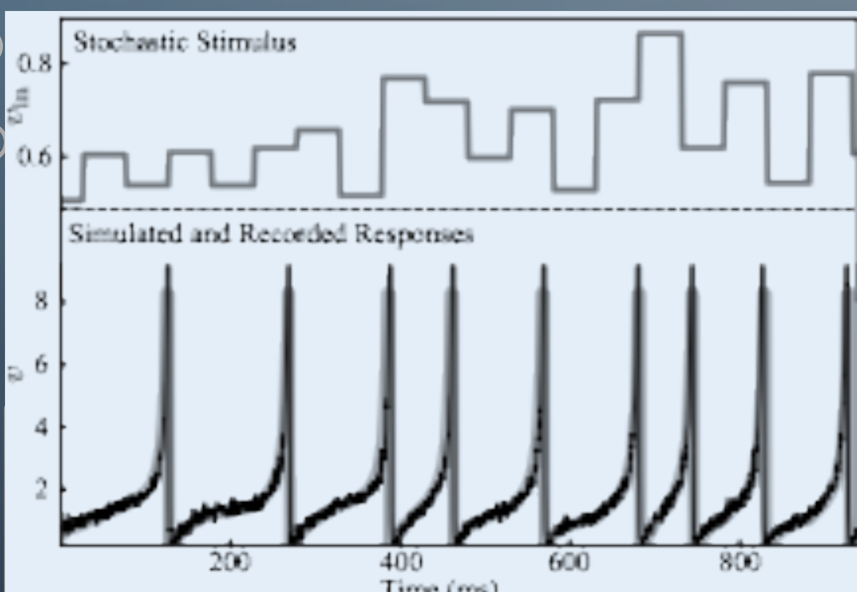
```
>>> som_quad = Soma("quadratic", {"x0": .9, "tau": 10e-3, "tau_ref": 2e-3})
>>> nrn_quad = Neuron("quad_neuron", som_quad)
>>> som_cub = Soma("cubic", {"x0": .385, "tau": 10e-3, "tau_ref": 2e-3})
>>> nrn_cub = Neuron("cub_neuron", som_cub)
>>> som_quad_ad = Soma("quadratic_adaptive", {"x0": .5, "tau": 10e-3, "tau_ref": 2e-3, "g_inf": 0.02})
>>> som_cub_ad = Soma("cubic_adaptive", {"x0": .385, "tau": 10e-3, "tau_ref": 2e-3, "g_inf": 0.02})
```

Quadratic Model:

$$\tau \dot{x} = -x + \frac{1}{2}x^2 + x_0, t_{ref}$$

Cubic Model:

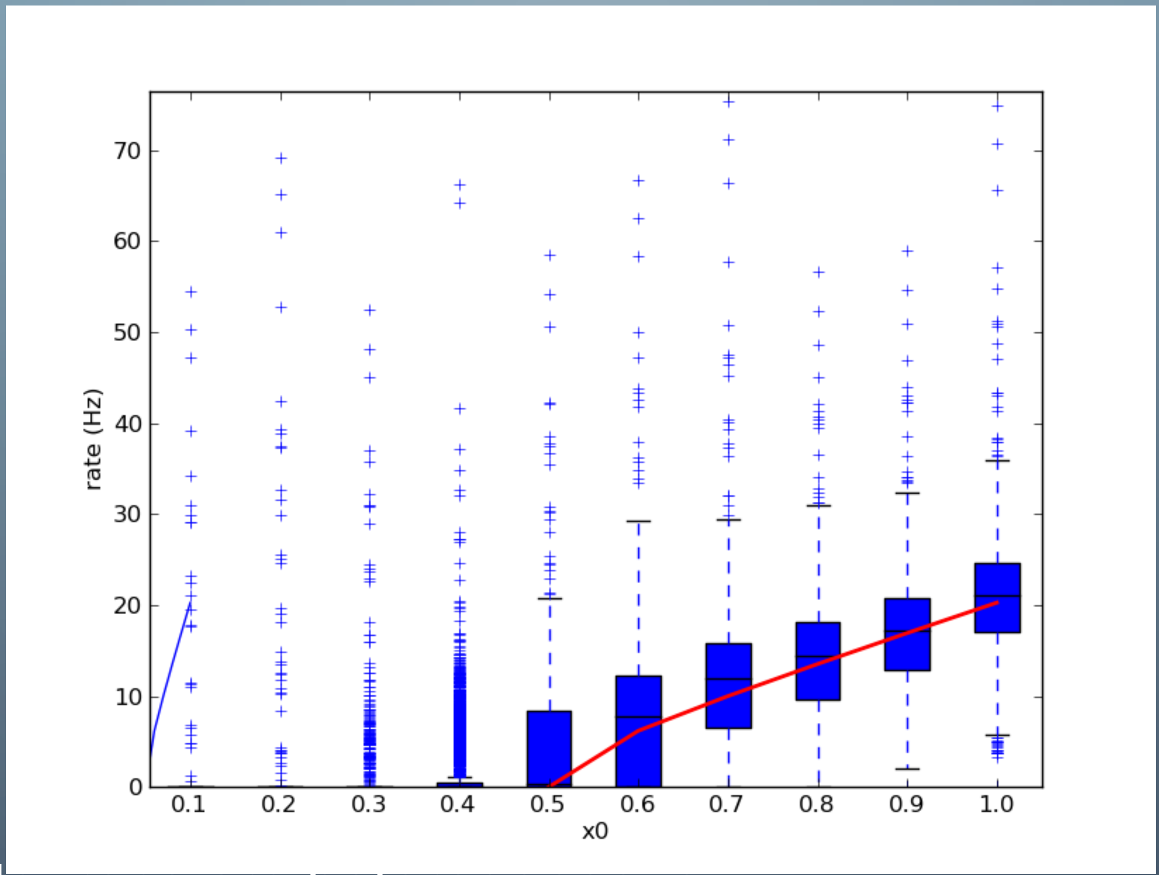
$$\tau \dot{x} = -x + \frac{1}{3}x^3 + x_0, t_{ref}$$



Bring the Neuron Model Together

```
>>> # Making inhibitory synapse and arbor
>>> syn_i = Synapse(“cond_syn”, {“erev”: .5, “tau_syn”: 10e-3, “lambda”: .8})
>>>
>>> # Making excitatory synapse and arbor
>>> syn_e = Synapse(“cond_syn”, {“erev”:2.5, “tau_syn”: 2e-3, “lambda”: .8})
>>>
>>> # Making quadratic SomaModel
>>> som_quad = Soma(“quadratic”, {“x0”: .9, “tau”: 10e-3})
>>>
>>> # Add arbors to the soma
>>> som_quad.AddASynapse(arb_e)
>>> som_quad.AddSynapse(arb_i)
>>>
>>> # Putting together the Neuron model
>>> nrn_quad = Neuron(“quad_neuron”, som_quad)
```

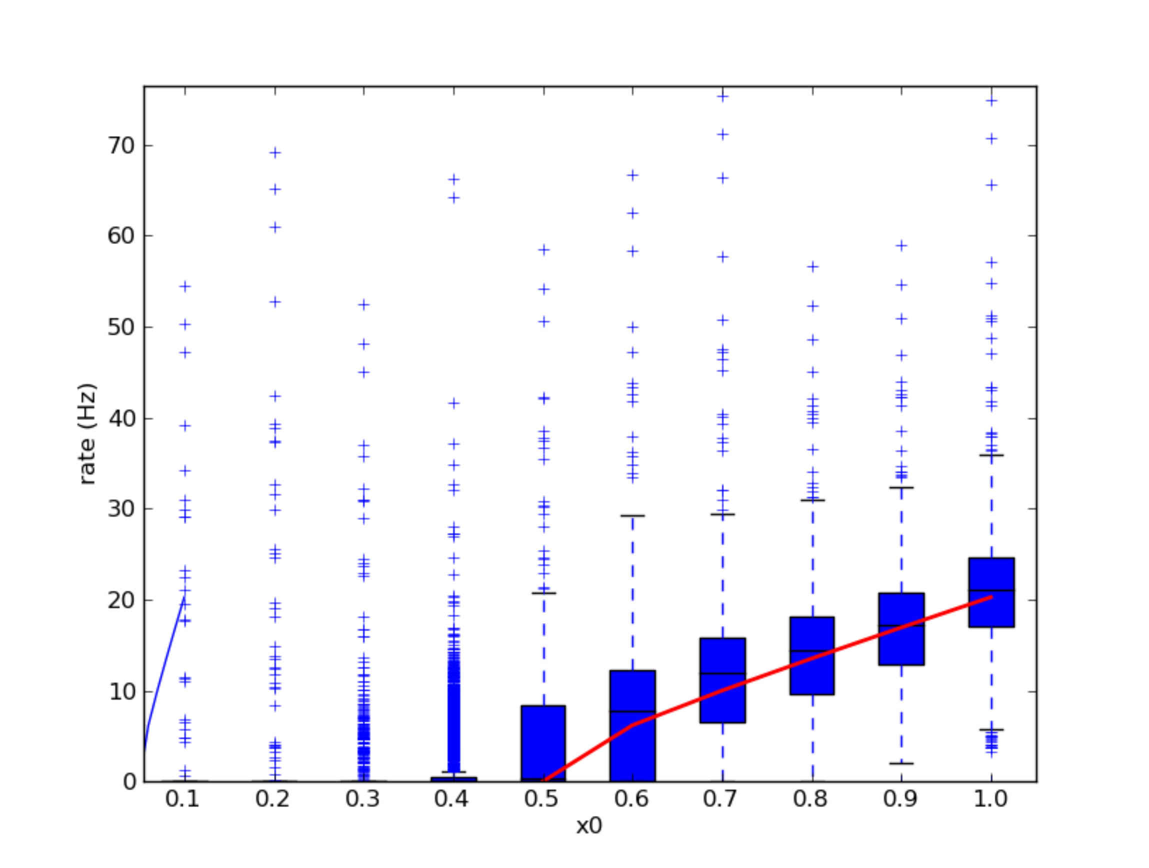
Driving neuron with input current



Peiran's data

- Neurons are heterogeneous
- Mean rate matches theory (red)
- Rate increases sublinearly (square root for quadratic)
- Starts spiking at x0>0.5

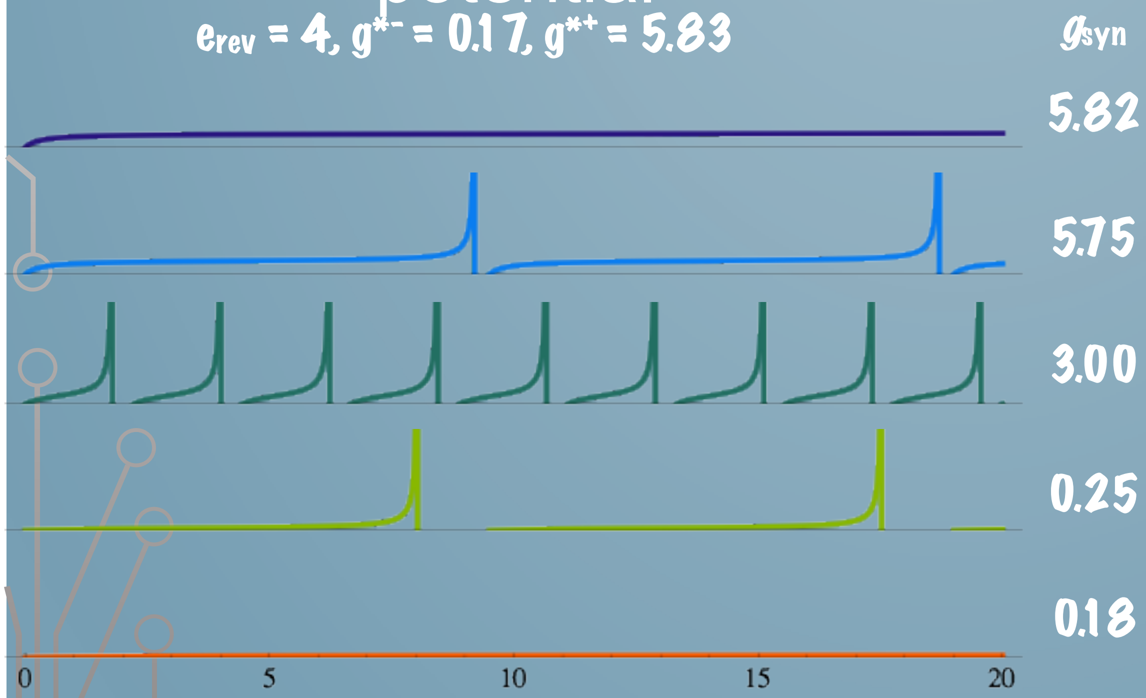
Show firing rate distribution and biological comparison



- Starts spiking at x0>0.5
- Rate increases sublinearly (square root for quadratic)
- Neurons are heterogeneous

DRIVING NEURON WITH SYNAPSE

Analytic membrane potential



- Quadratic positive-feedback can't overcome conductances larger than a maximum value (g^{*+})
- Such large shunts hold the membrane potential close to e_{rev}, preventing spiking even when $e_{rev} \gg 1$

Next week

Neurogrid III lecture on building networks