assignment-4

SYDE 556: Simulating Neurobiological Systems

Assignment 4: Nengo and Dynamics

Author: *Jonathan Johnston*

Course Instructor: *Professor C. Eliasmith*

This assignment details implementation of basic neuronal simulation and dynamic systems using nengo.

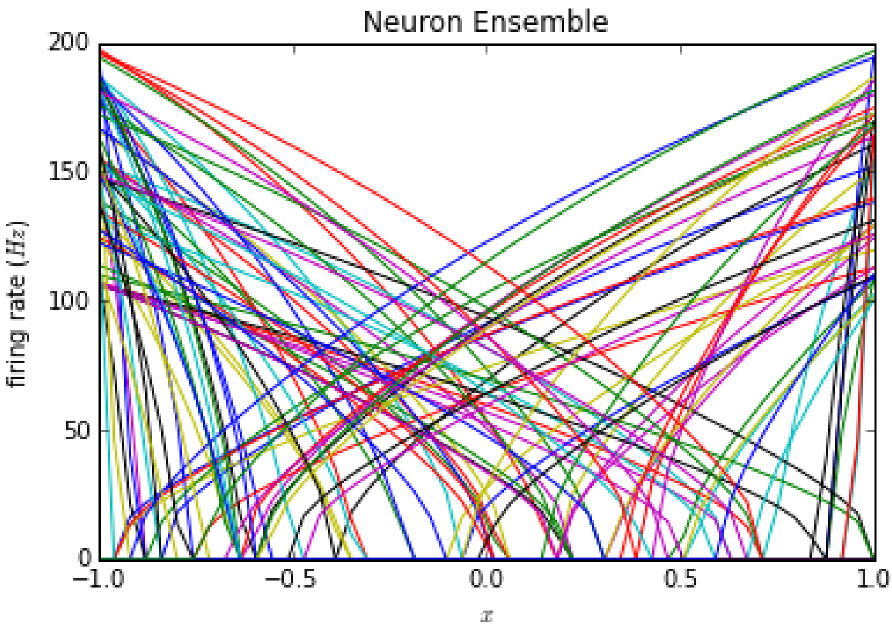
The assignment corresponds to the document hosted at:

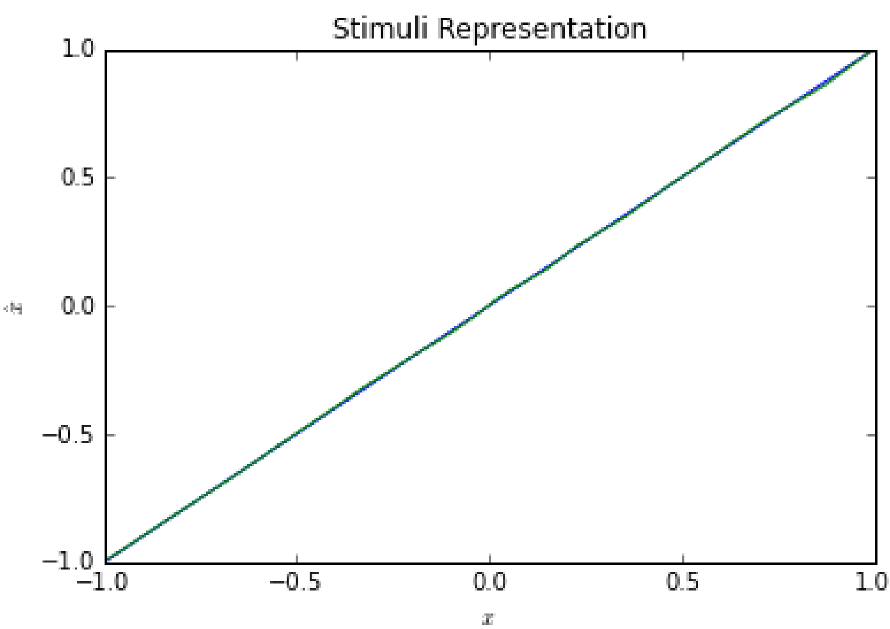
http://nbviewer.ipython.org/github/celiasmith/syde556/blob/master/Assignment%204.ipynb

Section 1: Building an Ensemble of Neurons

Part A First, we build a simple representational ensemble of neurons in one dimension and test its RMSE.

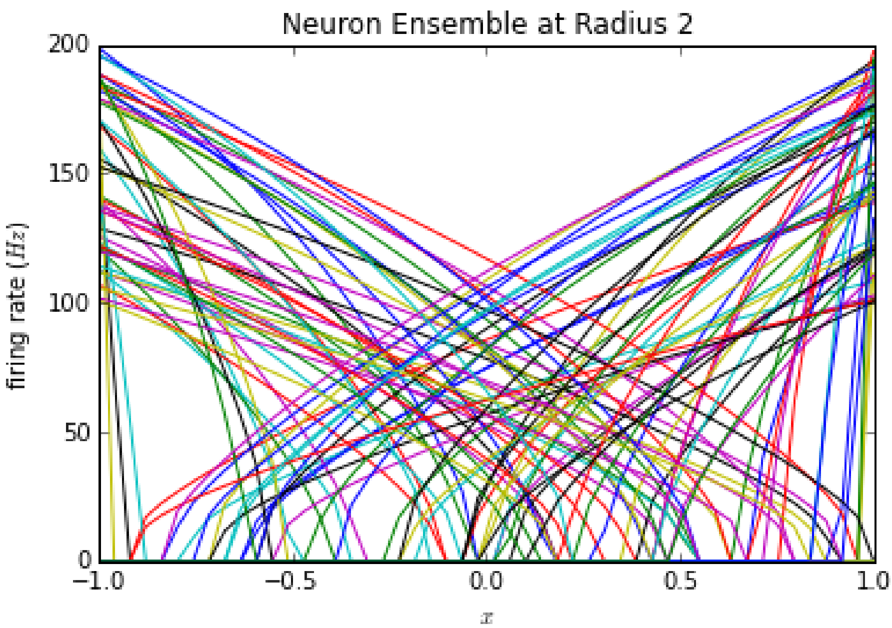
```
RMSE with 100 neurons is 0.0044631
```

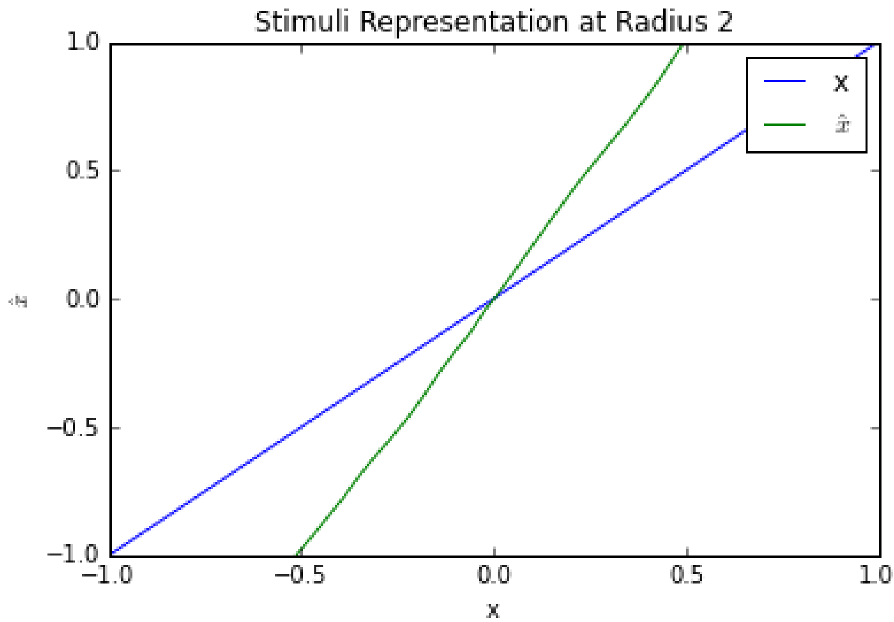


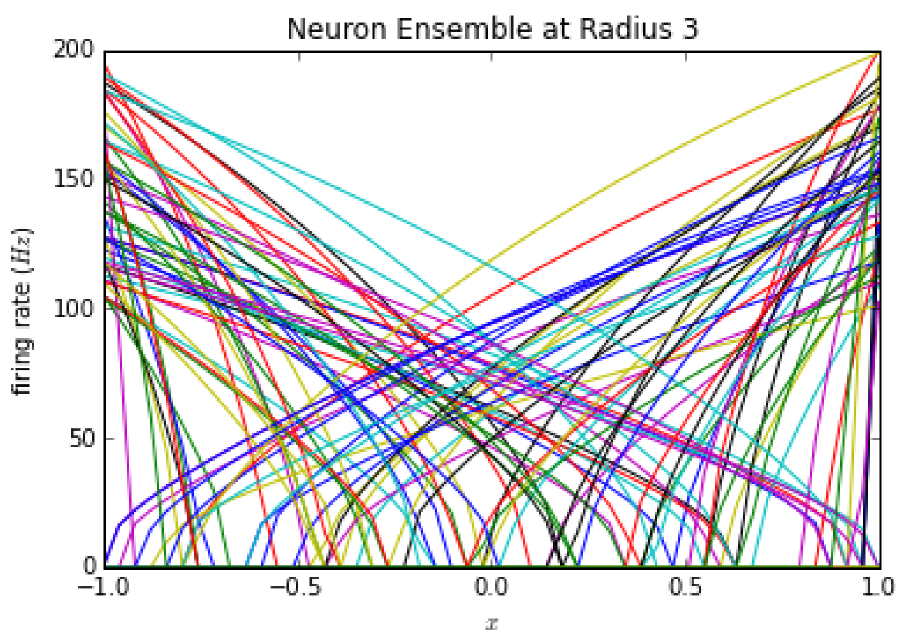


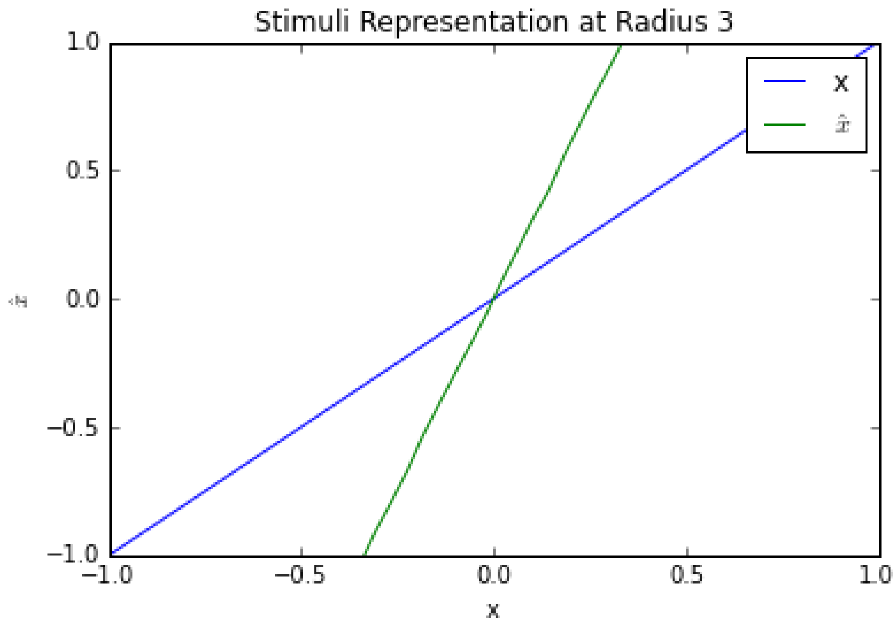
Part B The 'radius' is the representational radius of the ensemble. In most of the examples, we use a unit radius to represent stimuli from -1 to 1 in one dimension or around the unit circle in two dimensions. Increasing the radius means that the neuron ensemble is tuned, via decoding, to represent stimuli in a larger range of values. The RMSE increases as this radius increases because we simulate over -1 to 1 linearly, so essentially the increased radius is doing a multipliative transformation on the input.

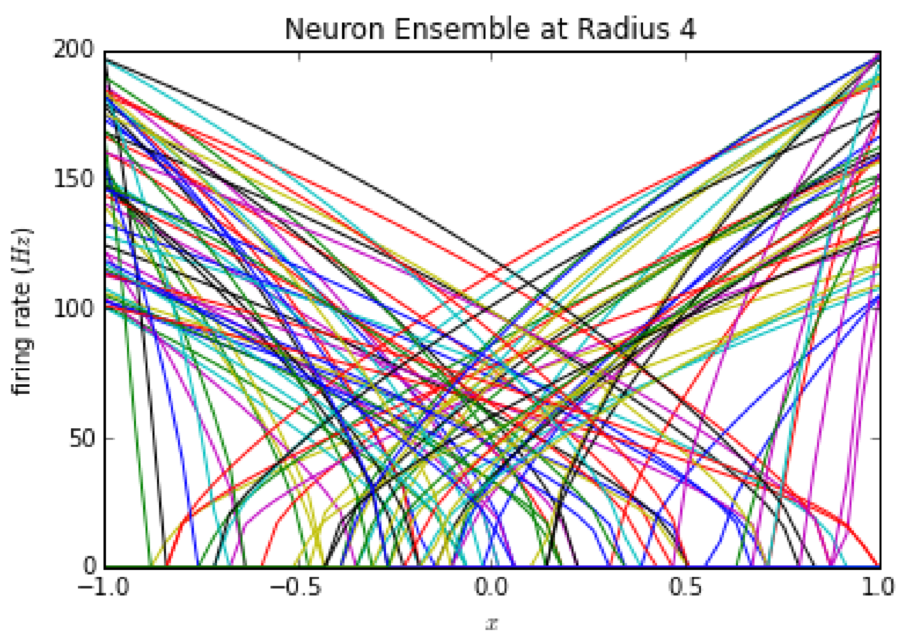
```
RMSE with 100 neurons at radius 2 is 0.586188
RMSE with 100 neurons at radius 3 is 1.17317
RMSE with 100 neurons at radius 4 is 1.76082
RMSE with 100 neurons at radius 5 is 2.34868
```

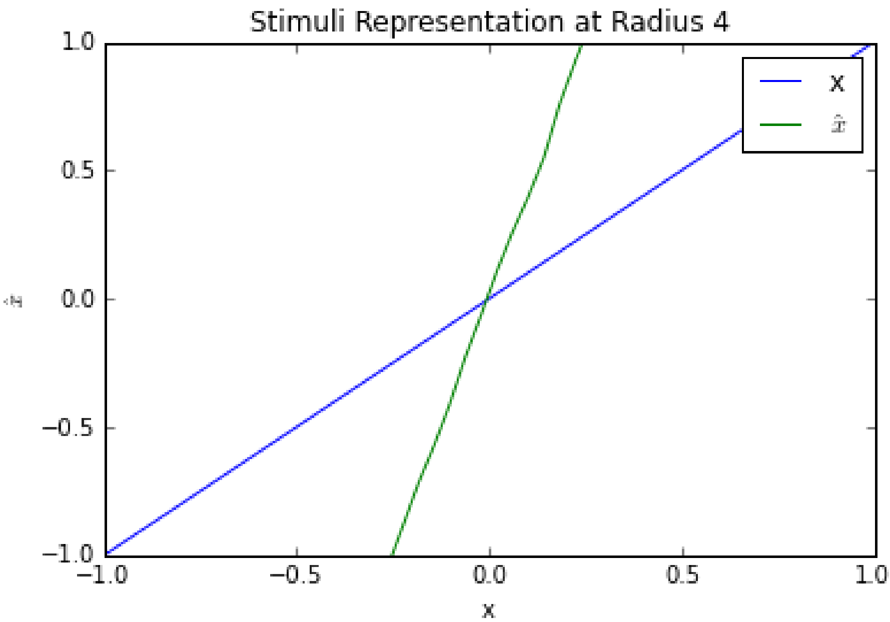


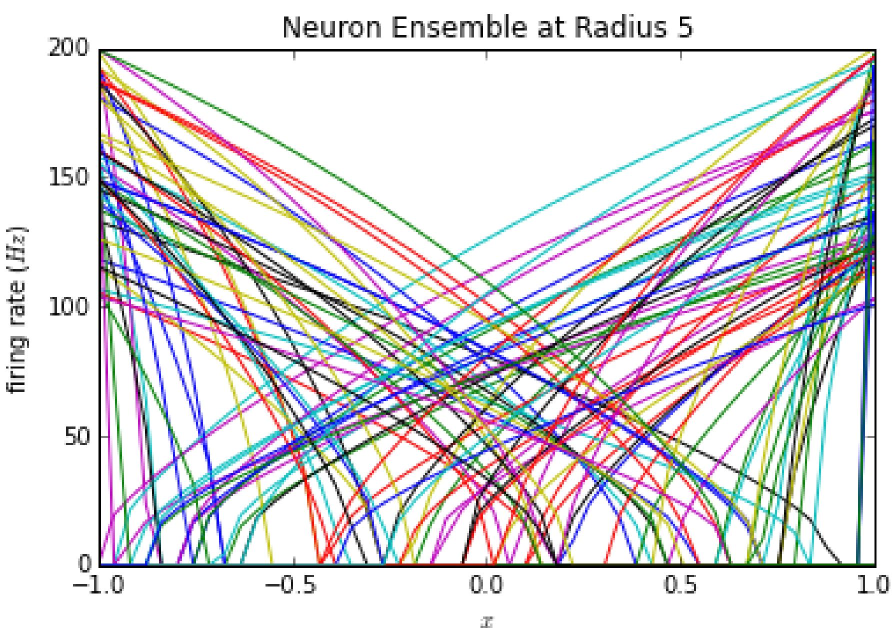

Stimuli Representation at Radius 2
x
$\hat{x}$
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
x
$\hat{x}$

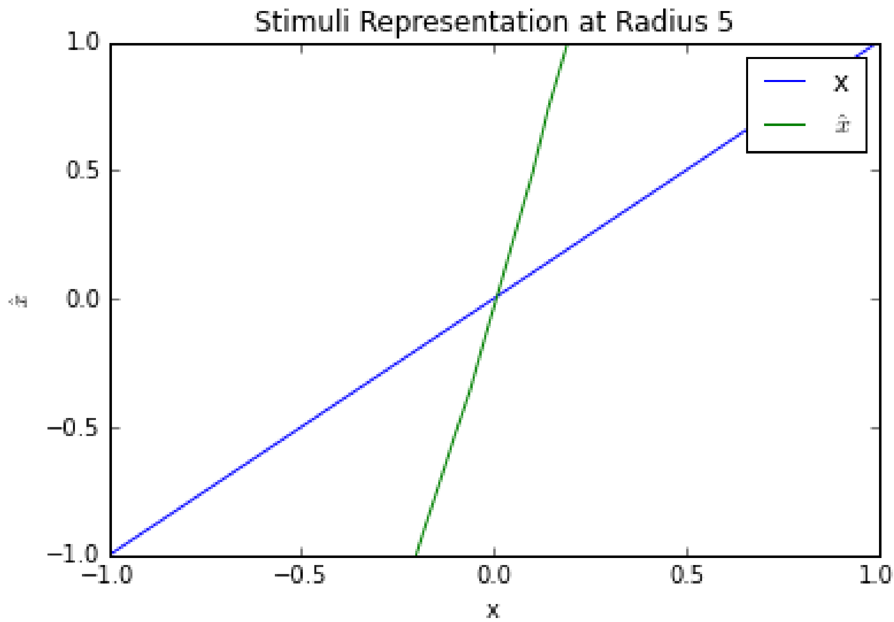

Neuron Ensemble at Radius 3
200
150
100
50
0
firing rate (Hz)
−1.0
−0.5
0.0
0.5
1.0
x


Stimuli Representation at Radius 3
1.0
0.5
0.0
−0.5
−1.0
$\hat{x}$
−1.0
−0.5
0.0
0.5
1.0
x
x
$\hat{x}$


Neuron Ensemble at Radius 4
200
150
100
50
0
firing rate (Hz)
−1.0
−0.5
0.0
0.5
1.0
x


Stimuli Representation at Radius 4
x
$\hat{x}$
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
x
$\hat{x}$


Neuron Ensemble at Radius 5
200
150
100
50
0
−1.0
−0.5
0.0
0.5
1.0
firing rate (Hz)
x



Part C As tau ref increases, the RMSE increases slightly. This is most likely due to the fact that neurons cannot fire at their maximum rates due to an increasing refractory period. This creates a small difference between the rate that was intended and the rate achieved, therefore causing error in representation.

```
RMSE with 100 neurons at Tau Ref 0.0001 is 0.00409225
RMSE with 100 neurons at Tau Ref 0.001 is 0.00416193
RMSE with 100 neurons at Tau Ref 0.002 is 0.00435972
RMSE with 100 neurons at Tau Ref 0.004 is 0.00597729
RMSE with 100 neurons at Tau Ref 0.005 is 0.00992824
```

Part D At low tau RC levels, the RMSE is relatively high, but decreases to a point around 0.02, after which the RMSE levels off. For very large tau RC, the decaying exponentials which it defines become negligible; however for very small tau RC, the neurons ramp up very quickly and misfire, causing higher error in representation.

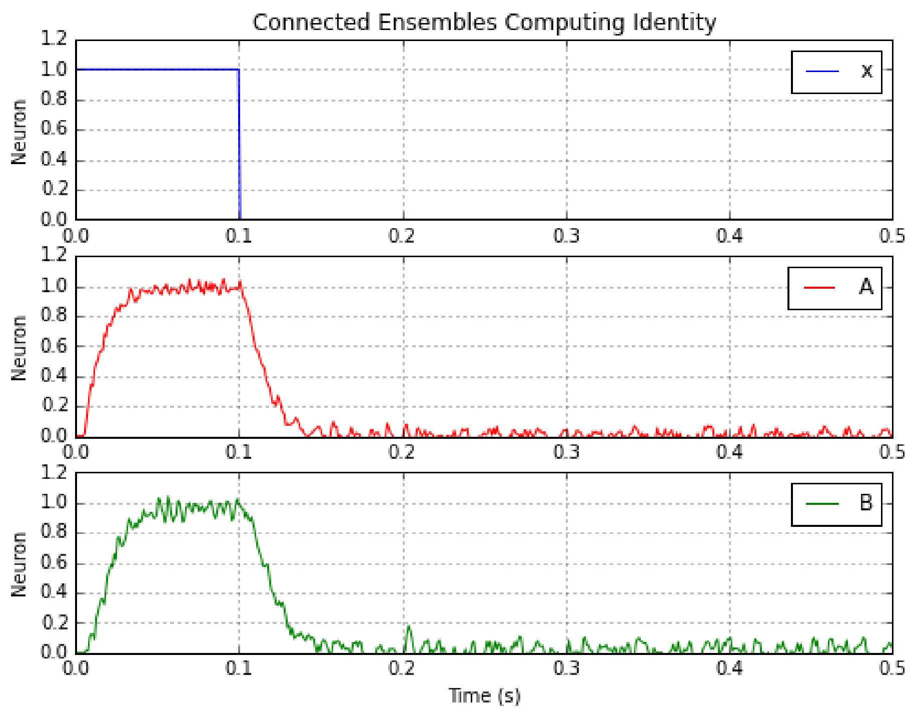
```
RMSE with 100 neurons at Tau RC 0.0001 is 0.0478531
RMSE with 100 neurons at Tau RC 0.001 is 0.0132544
RMSE with 100 neurons at Tau RC 0.01 is 0.00665679
RMSE with 100 neurons at Tau RC 0.1 is 0.0033354
RMSE with 100 neurons at Tau RC 1 is 0.00382249
RMSE with 100 neurons at Tau RC 10 is 0.00314218
```

```
RMSE with 100 neurons at Tau RC 100 is 0.00413528
```

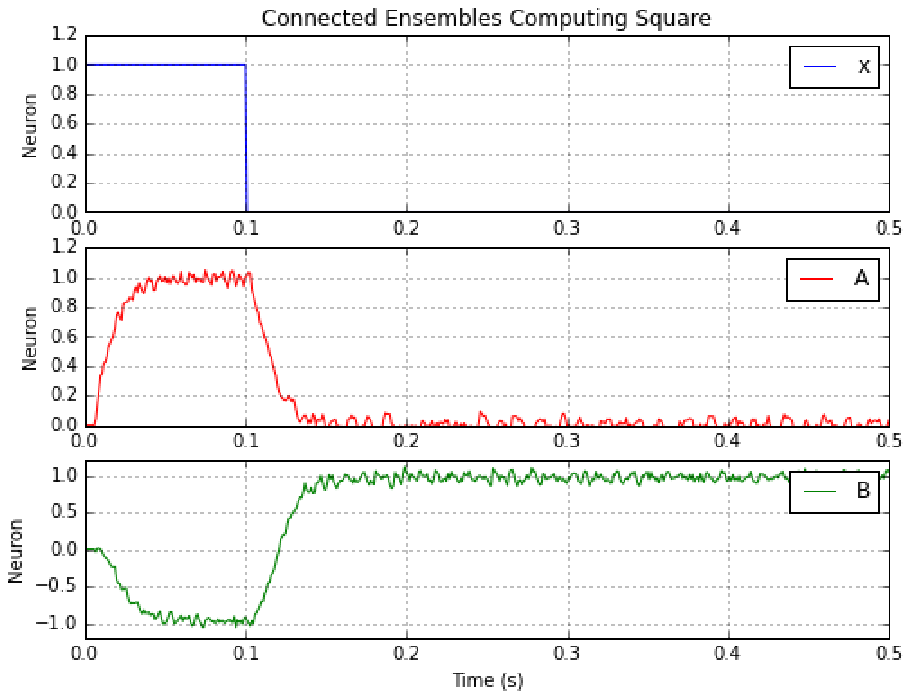
Section 2: Connecting Neurons

In this section we connect ensembles of neurons to compute various functions and transformations.

Part A First, hook two neuron ensembles together to compute the identity of the input.

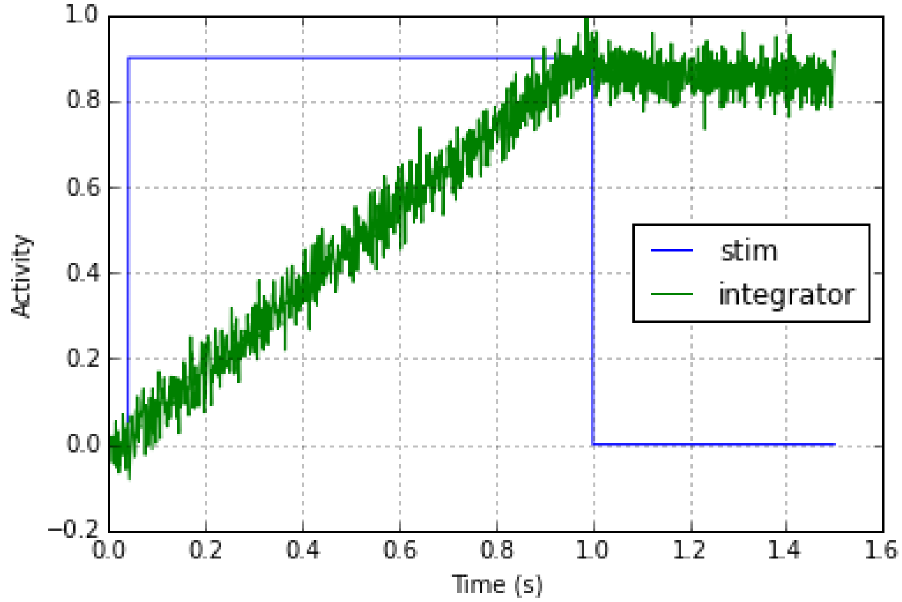


Part B Now we connect two neuron ensembles to compute a simple function y = 1 - 2*x.

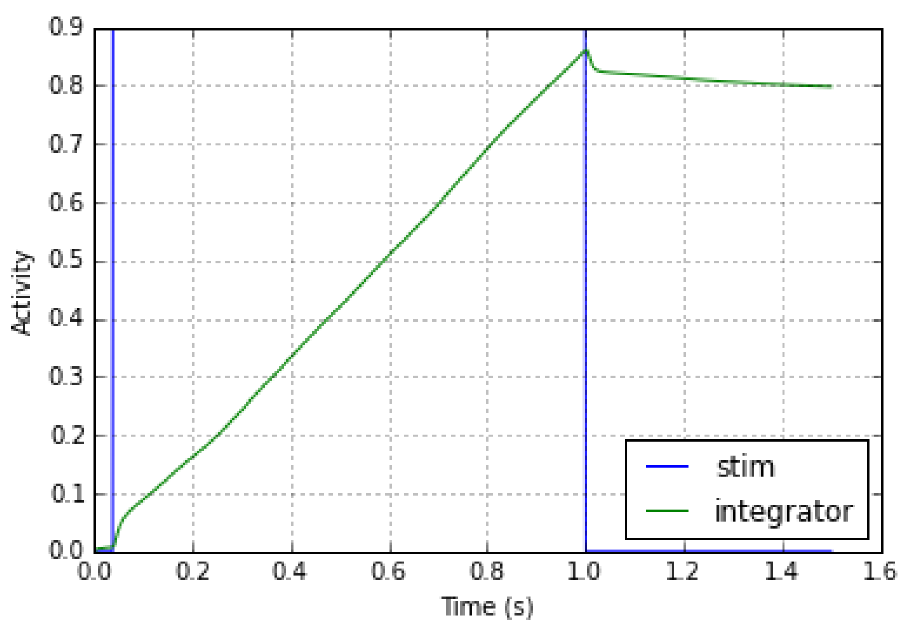


Section 3: Dynamics

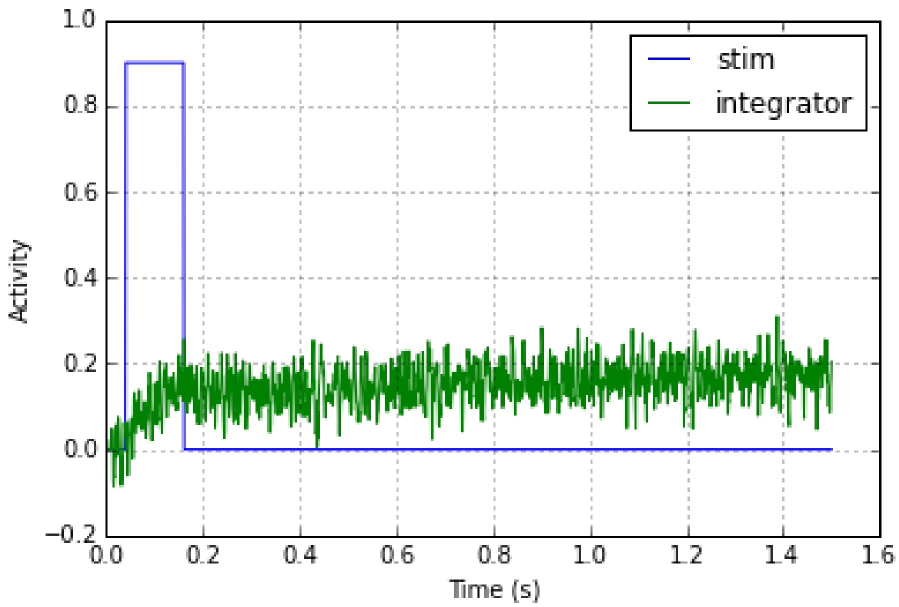
Part A The expected result of an integrated unit step function is simply the ramp function; therefore a unit step with a cutoff will produce a ramp and then a level-off, as the current value is fed into the feedback loop with no further input. The simulated result actually resembles the mathematical integration fairly closely; however there is a significant amount of noise in the ensemble population.



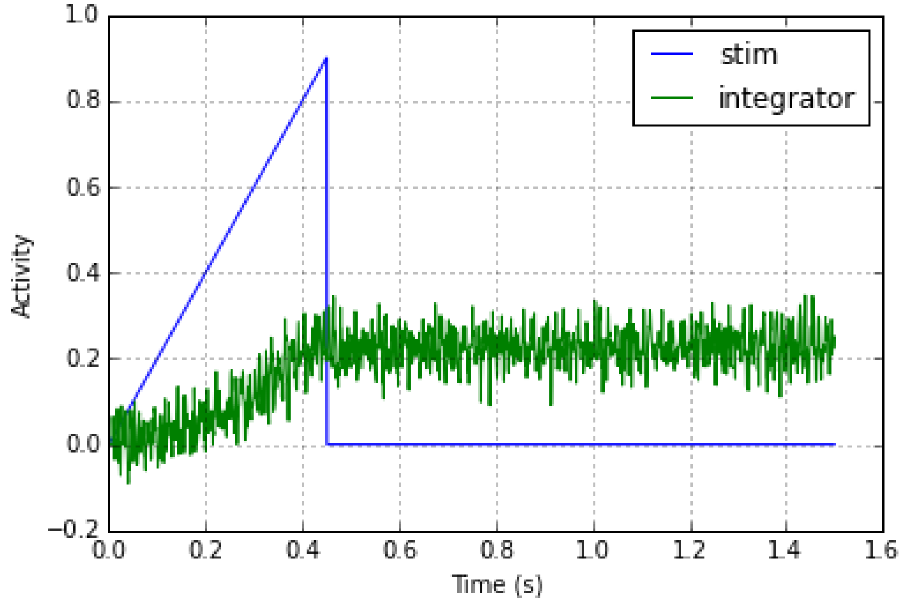
Part B The rate LIF neuron seems to have a bit of undershoot as compared to the non-rate neurons. It looks like the previous simulation was slightly better, even though the LIF rate neurons have almost zero noise.

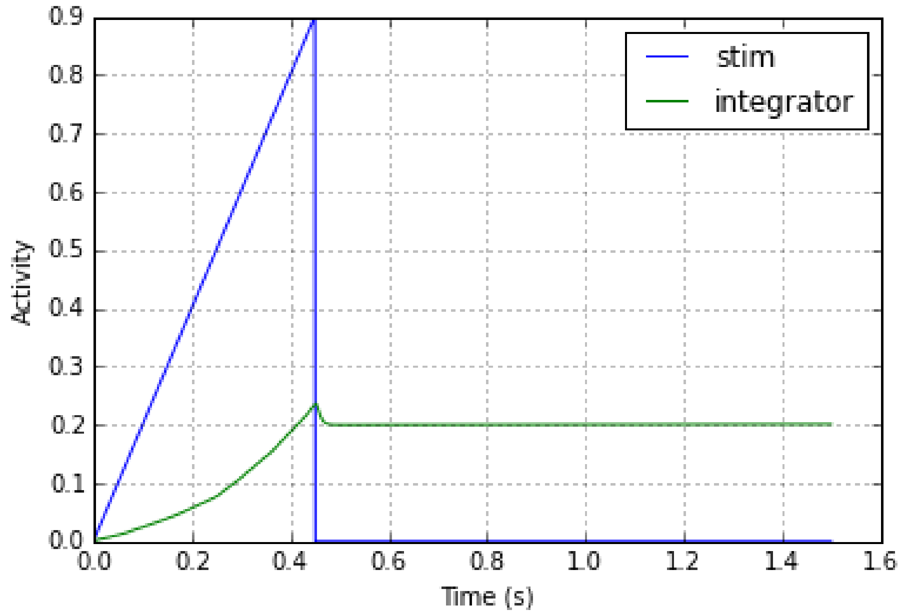


Part C As compared to part A, this input function is shorter, and so it does not allow for enough buildup of the integrator to reach the full input value. This doesn't seem to be better or worse, as the input is the only thing that has changed, but it does reveal something about the rate at which the model reaches input reference level, which is quite slow.



Part D The ensemble ends up representing a sort of quadratic curve, and then levels off, which is exactly the expected behaviour. The integral of a ramp is a quadratic. If the ramp ends, the integrator should continue at the last value. In the simulated models there is a bit of drop-off after the ramp input ends.





Part E The integral of 5sin(5t) is -cos(5t). The simulation approximates this relatively well in that the amplitude is correct, and the period is the same as the input. The beginning of the approximation should begin at a negative value; however since the integrator has no past information, and only adds, it cannot approximate the true integral at the beginning of the input, and thus starts at zero. It is due to this that it is slightly out of phase with the input, and so it is not represented properly.