FYS388_V15_Exercise08_Solution

March 20, 2015

1 FYS388 Exercise 8: Solution

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np
        import pandas as pd
        import sys
        sys.path.append('/Users/plesser/NEST/code/releases/nest-2.6.0/ins/lib/python2.7/site-packages/')

        %load_ext autoreload
        %autoreload 2
        %matplotlib inline

        plt.rcParams['figure.figsize'] = (16, 8)
```

1.1 Task 1

1.1.1 Simulation 1

```
In [2]: run brunel-delta-nest.py
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15637600
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 31.86 Hz
Inhibitory rate   : 31.98 Hz
Building time     : 8.33 s
Simulation time   : 91.55 s
```

1.1.2 Simulation 2

```
In [3]: from brunel_delta import sim_brunel_delta
```

```
In [4]: esp, isp = sim_brunel_delta()
```

```
Building network
Connecting devices
```

```
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15637600
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 31.86 Hz
Inhibitory rate   : 31.98 Hz
Building time     : 8.19 s
Simulation time   : 100.74 s
```

```
In [5]: esp.describe()
```

Out[5]:

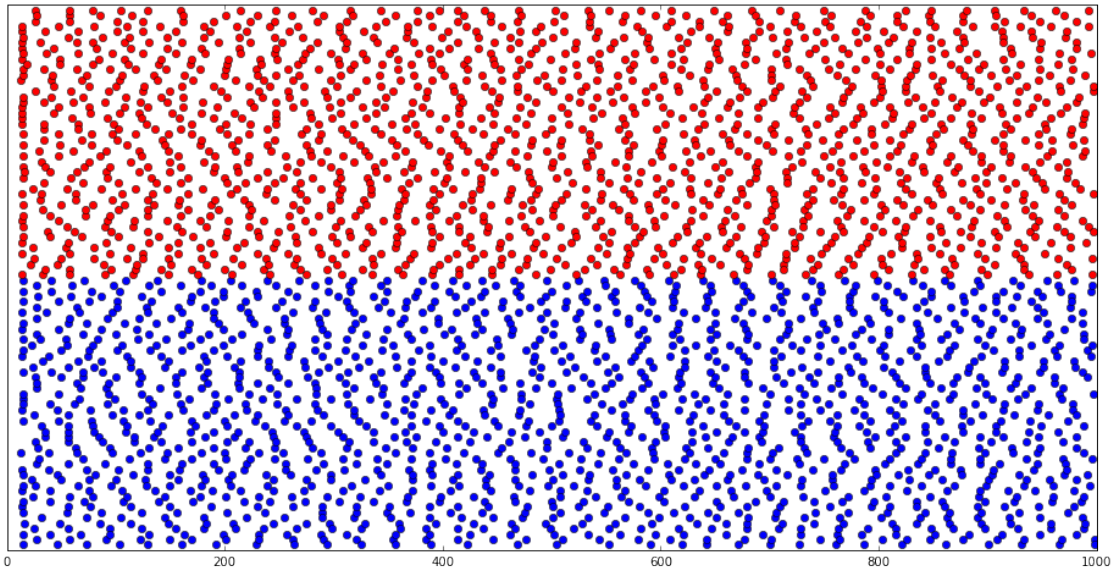
	senders	times
count	1593.000000	1593.000000
mean	25.497175	501.450345
std	14.433182	287.833542
min	1.000000	13.200000
25%	13.000000	249.200000
50%	25.000000	501.600000
75%	38.000000	751.900000
max	50.000000	998.100000

```
In [6]: esp[:5]
```

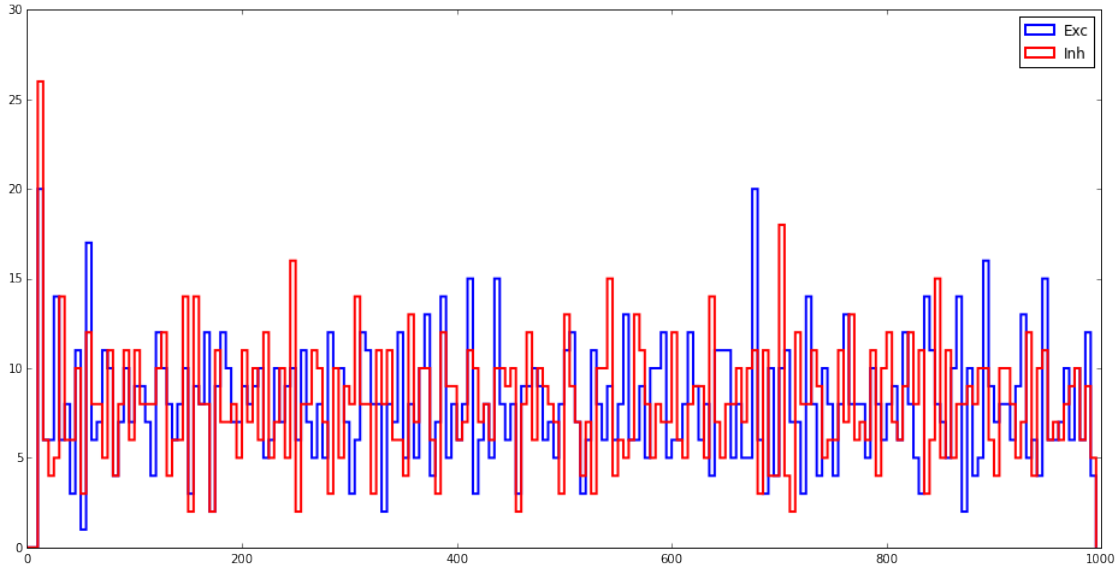
Out[6]:

	senders	times
0	18	13.2
1	11	13.5
2	41	13.6
3	44	13.8
4	26	14.0

```
In [9]: plt.plot(esp.times, esp.senders, 'bo')
        plt.plot(isp.times, isp.senders-10000+50, 'ro')
        plt.ylim(0, 101)
        plt.yticks([]);
```




```
In [10]: bins = np.arange(0, 1000, 5.)
         plt.hist(esp.times, bins=bins, histtype='step', color='b', lw=2, label='Exc')
         plt.hist(isp.times, bins=bins, histtype='step', color='r', lw=2, label='Inh')
         plt.legend();
```



1.1.3 Simulation 3

```
In [11]: esp, isp = sim_brunel_delta(simtime=250, num_threads=1)
```

```
Building network
Connecting devices
```

```
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15637600
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 20.00 Hz
Inhibitory rate   : 20.00 Hz
Building time     : 8.01 s
Simulation time   : 22.00 s
```

```
In [12]: esp, isp = sim_brunel_delta(simtime=250, num_threads=2)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15637600
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 20.00 Hz
Inhibitory rate   : 20.00 Hz
Building time     : 5.34 s
Simulation time   : 11.86 s
```

```
In [13]: esp, isp = sim_brunel_delta(simtime=250, num_threads=4)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15637600
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 20.00 Hz
Inhibitory rate   : 20.00 Hz
Building time     : 3.67 s
Simulation time   : 7.18 s
```

1.2 Task 2: Reproducing Brunel

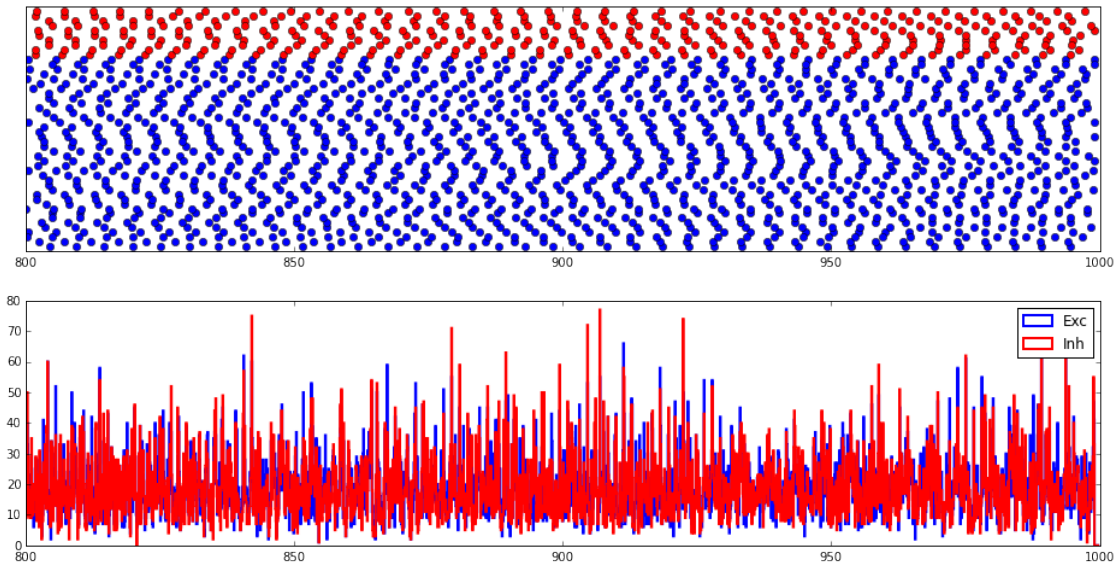
```
In [14]: def plot_task2(esp, isp):
             e_gid_min = esp.senders.min()
             i_gid_min = isp.senders.min()
```

```
e_hist = (800 <= esp.times) & (esp.times <= 1000)
i_hist = (800 <= isp.times) & (isp.times <= 1000)
e_plot = e_hist & (esp.senders < (e_gid_min + 40))
i_plot = i_hist & (isp.senders < (i_gid_min + 10))

plt.subplot(211)
plt.plot(esp.times[e_plot], esp.senders[e_plot] - e_gid_min +  1, 'bo')
plt.plot(isp.times[i_plot], isp.senders[i_plot] - i_gid_min + 41, 'ro')
plt.ylim(0, 51)
plt.yticks([])

plt.subplot(212)
bins = np.arange(800., 1000., 0.1)
plt.hist(esp.times[e_hist].values, bins=bins, histtype='step', color='b', lw=2, label='Exc
plt.hist(isp.times[i_hist].values, bins=bins, histtype='step', color='r', lw=2, label='Inh
plt.legend();
```

1.2.1 Simulation 1: Case A ($g = 3$, $\eta = 2$)

```
In [15]: esp, isp = sim_brunel_delta(order=2500, N_rec=1000, g=3., eta=2., num_threads=4)
         plot_task2(esp, isp)
```

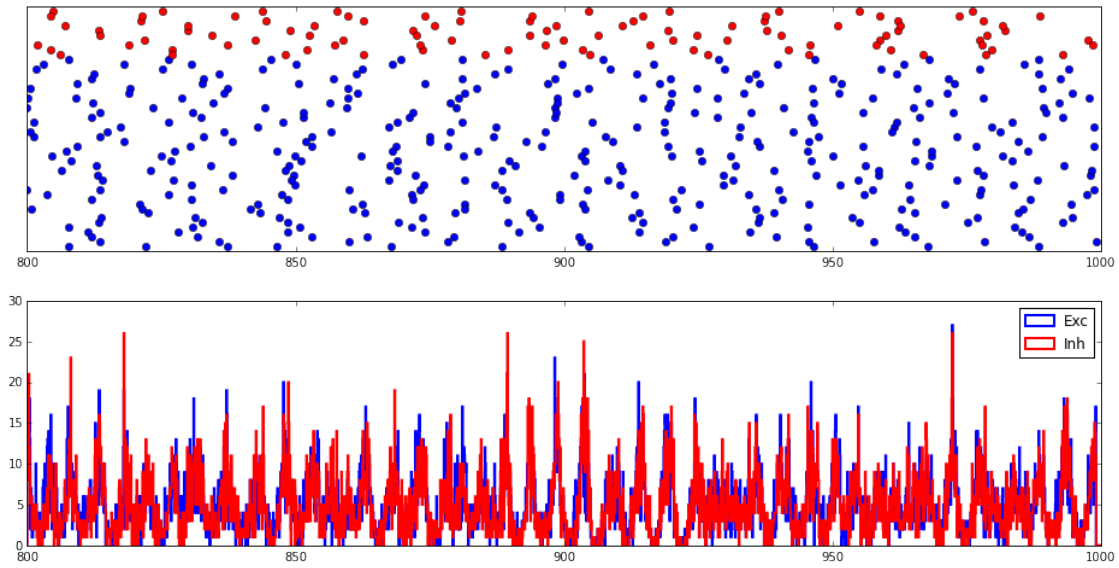
```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15639500
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 189.36 Hz
Inhibitory rate   : 189.36 Hz
Building time     : 3.81 s
Simulation time   : 122.48 s
```



1.2.2 Simulation 2: Case B ($g = 6$, $\eta = 4$)

```
In [16]: esp, isp = sim_brunel_delta(order=2500, N_rec=1000, g=6., eta=4., num_threads=2)
         plot_task2(esp, isp)
```

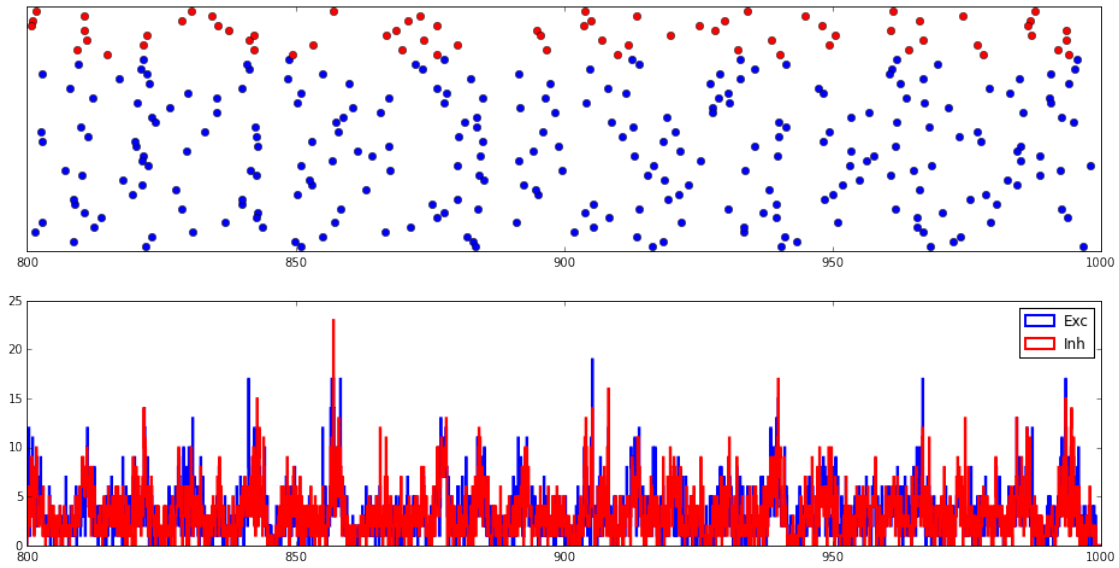
```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15639500
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 49.97 Hz
Inhibitory rate   : 50.08 Hz
Building time     : 5.20 s
Simulation time   : 69.71 s
```



1.2.3 Simulation 3: Case C ($g = 5$, $\eta = 2$)

```
In [17]: esp, isp = sim_brunel_delta(order=2500, N_rec=1000, g=5., eta=2., num_threads=2)
         plot_task2(esp, isp)
```

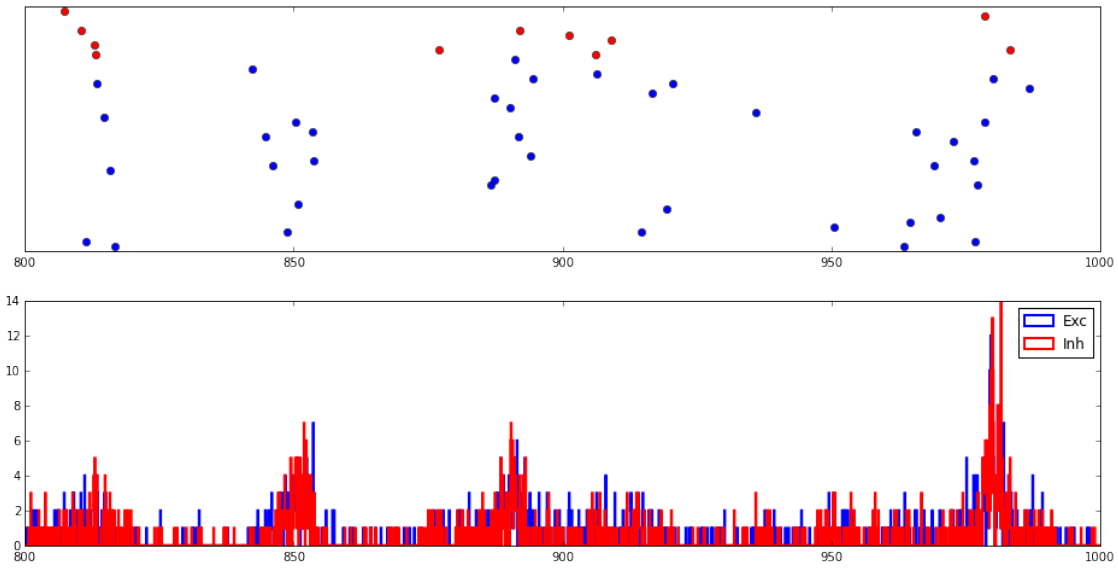
```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15639500
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 32.02 Hz
Inhibitory rate   : 32.00 Hz
Building time     : 5.15 s
Simulation time   : 49.37 s
```



1.2.4 Simulation 4: Case D ($g = 4.5$, $\eta = 0.9$)

```
In [18]: esp, isp = sim_brunel_delta(order=2500, N_rec=1000, g=4.5, eta=0.9, num_threads=2)
         plot_task2(esp, isp)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15639500
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 5.12 Hz
Inhibitory rate   : 5.09 Hz
Building time     : 4.99 s
Simulation time   : 19.28 s
```



1.3 Task 3

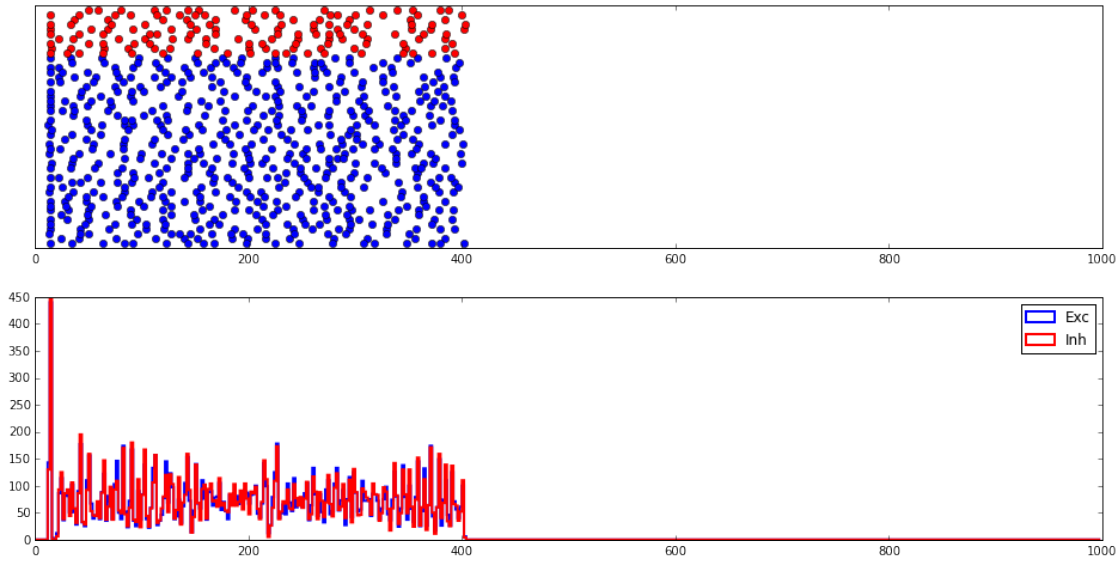
```
In [19]: def plot_task3(esp, isp):
             e_gid_min = esp.senders.min()
             i_gid_min = isp.senders.min()

             e_hist = (0 <= esp.times) & (esp.times <= 1000)
             i_hist = (0 <= isp.times) & (isp.times <= 1000)
             e_plot = e_hist & (esp.senders < (e_gid_min + 40))
             i_plot = i_hist & (isp.senders < (i_gid_min + 10))

             plt.subplot(211)
             plt.plot(esp.times[e_plot], esp.senders[e_plot] - e_gid_min +  1, 'bo')
             plt.plot(isp.times[i_plot], isp.senders[i_plot] - i_gid_min + 41, 'ro')
             plt.ylim(0, 51)
             plt.yticks([])
             plt.xlim(0, 1000)

             plt.subplot(212)
             bins = np.arange(0., 1000., 2.0)
             plt.hist(esp.times[e_hist].values, bins=bins, histtype='step', color='b', lw=2, label='Exc
             plt.hist(isp.times[i_hist].values, bins=bins, histtype='step', color='r', lw=2, label='Inh
             plt.legend();
```

1.3.1 Simulation 1: Case C parameters

```
In [20]: esp, isp = sim_brunel_delta(order=2500, N_rec=1000, g=5., eta=2., V_reset=10., input_stop=400.
         plot_task3(esp, isp)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
```

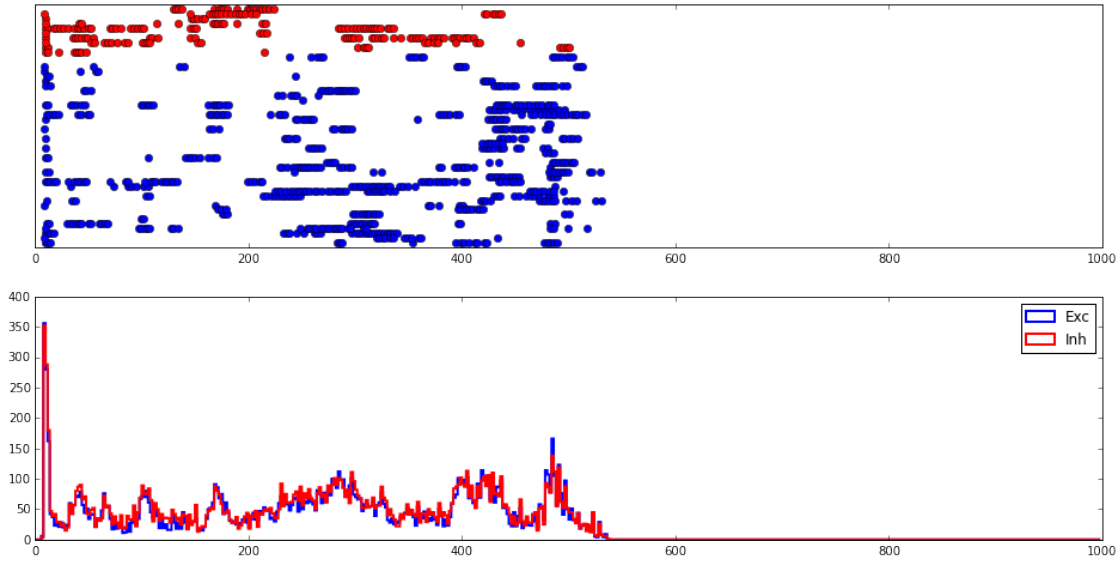
```
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15639500
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 14.84 Hz
Inhibitory rate   : 14.95 Hz
Building time     : 4.92 s
Simulation time   : 23.04 s
```



1.3.2 Simulation 2: Synaptic weight increased tenfold

```
In [21]: esp, isp = sim_brunel_delta(order=2500, N_rec=1000, g=5., eta=2., J=1., V_reset=10., input_sto
         plot_task3(esp, isp)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Simulating
Brunel network simulation (Python)
Number of neurons : 12500
Number of synapses: 15639500
       Exitatory  : 12512500
       Inhibitory : 3125000
Excitatory rate   : 13.64 Hz
Inhibitory rate   : 14.44 Hz
Building time     : 4.74 s
Simulation time   : 19.29 s
```



1.4 Task 4

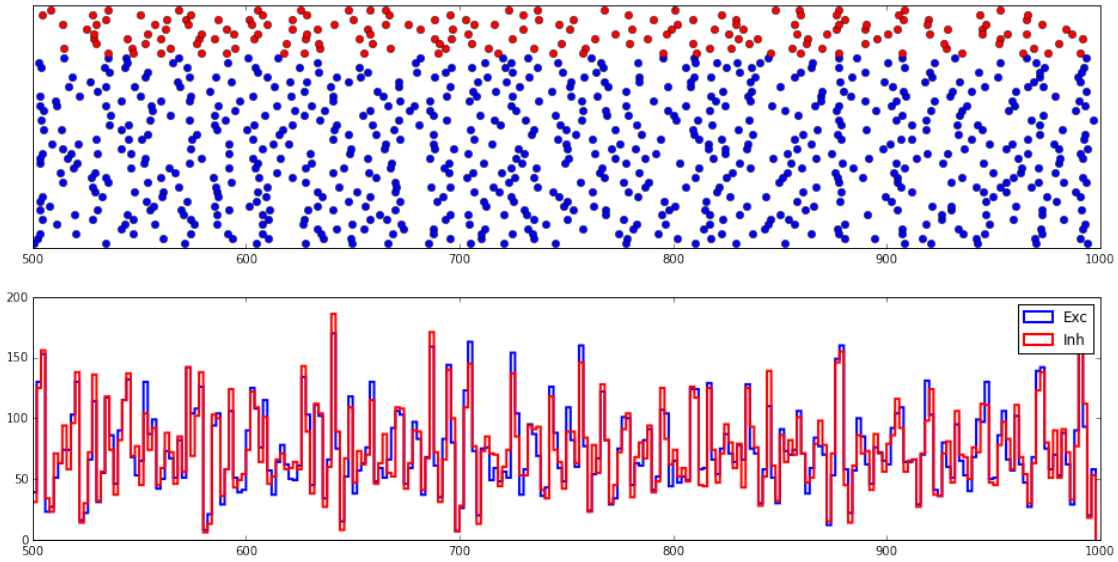
```
In [22]: def plot_task4(esp, isp):
             e_gid_min = esp.senders.min()
             i_gid_min = isp.senders.min()

             e_hist = (500 <= esp.times) & (esp.times <= 1000)
             i_hist = (500 <= isp.times) & (isp.times <= 1000)
             e_plot = e_hist & (esp.senders < (e_gid_min + 40))
             i_plot = i_hist & (isp.senders < (i_gid_min + 10))

             print "Excitatory rate: {:.2f} Hz".format(len(esp[e_hist]) / 125.)    # 125 = 250 neurons *
             print "Inhibitory rate: {:.2f} Hz".format(len(isp[i_hist]) / 125.)    # 125 = 250 neurons *

             plt.subplot(211)
             plt.plot(esp.times[e_plot], esp.senders[e_plot] - e_gid_min +  1, 'bo')
             plt.plot(isp.times[i_plot], isp.senders[i_plot] - i_gid_min + 41, 'ro')
             plt.ylim(0, 51)
             plt.yticks([])

             plt.subplot(212)
             bins = np.arange(500., 1000., 2.)
             plt.hist(esp.times[e_hist].values, bins=bins, histtype='step', color='b', lw=2, label='Exc
             plt.hist(isp.times[i_hist].values, bins=bins, histtype='step', color='r', lw=2, label='Inh
             plt.legend();
```

1.4.1 Reference case

Full model with $V_{\text{reset}} = 10$.

```
In [23]: esp, isp = sim_brunel_delta(order=2500, N_rec=1000, g=5., eta=2., V_reset=10.0, num_threads=2,
         plot_task4(esp, isp)
```

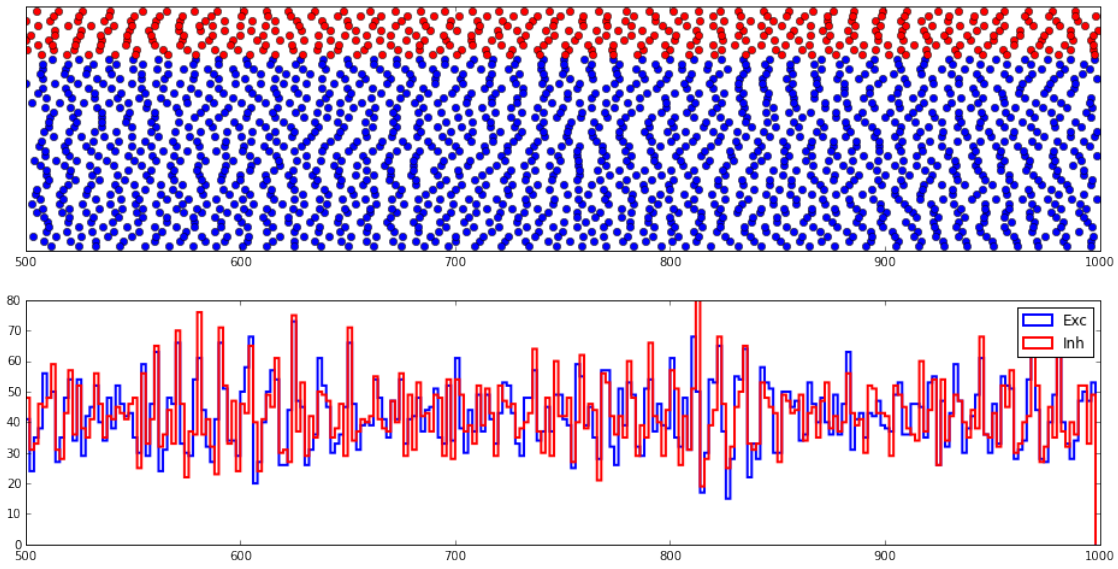
```
Excitatory rate: 148.95 Hz
Inhibitory rate: 149.56 Hz
```



1.4.2 Reduced to 1/10

```
In [24]: esp, isp = sim_brunel_delta(order=250, N_rec=250, g=5., eta=2., V_reset=10.0, num_threads=2, pr
         plot_task4(esp, isp)
```

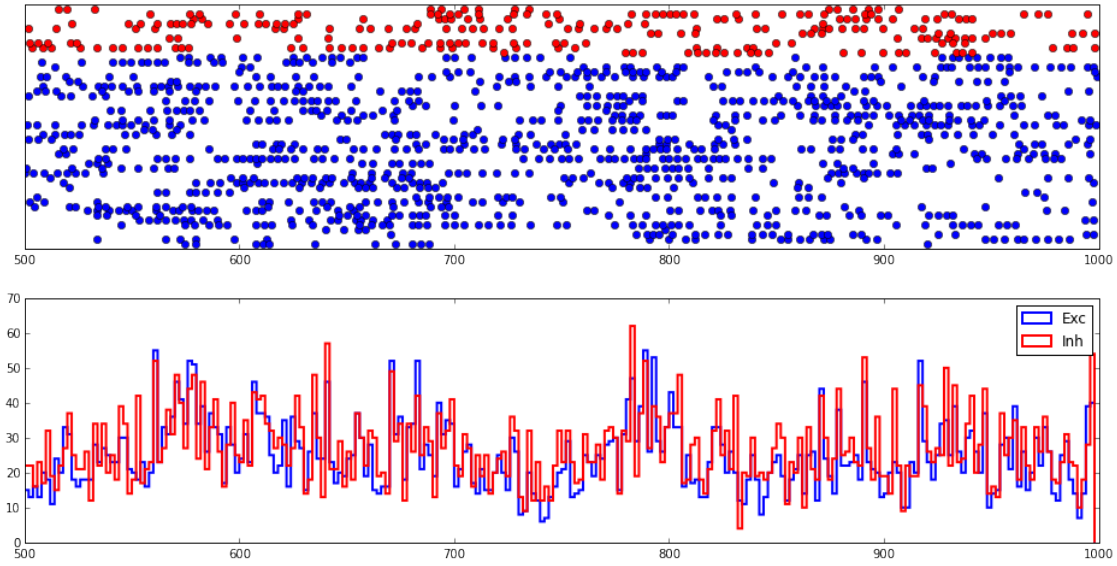
```
Excitatory rate: 85.13 Hz
Inhibitory rate: 85.50 Hz
```



Rate now far too high, but then each neuron receives only 1/10 the input. Compensate by increasing J by factor 10.

```
In [25]: esp, isp = sim_brunel_delta(order=250, N_rec=250, g=5., eta=2., J=1.0, V_reset=10.0, num_threa
         plot_task4(esp, isp)
```

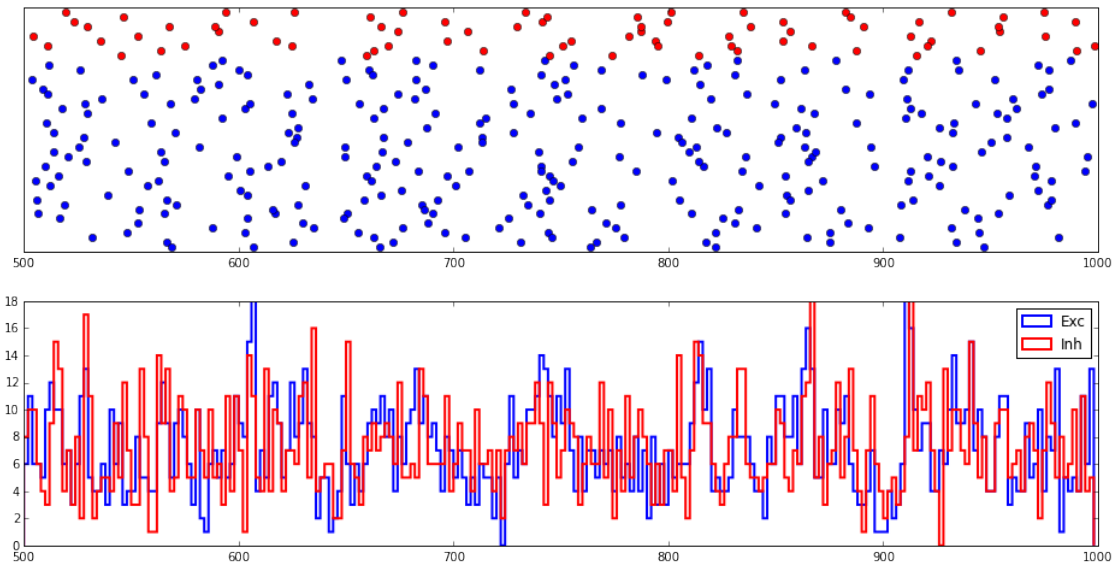
```
Excitatory rate: 49.42 Hz
Inhibitory rate: 54.10 Hz
```



The rate is almost correct now, but quite different, bursty firing pattern. Let's try reducing external input, back to original J:

```
In [26]: esp, isp = sim_brunel_delta(order=250, N_rec=250, g=5., eta=1., J=0.1, V_reset=10.0, num_threa
         plot_task4(esp, isp)
```

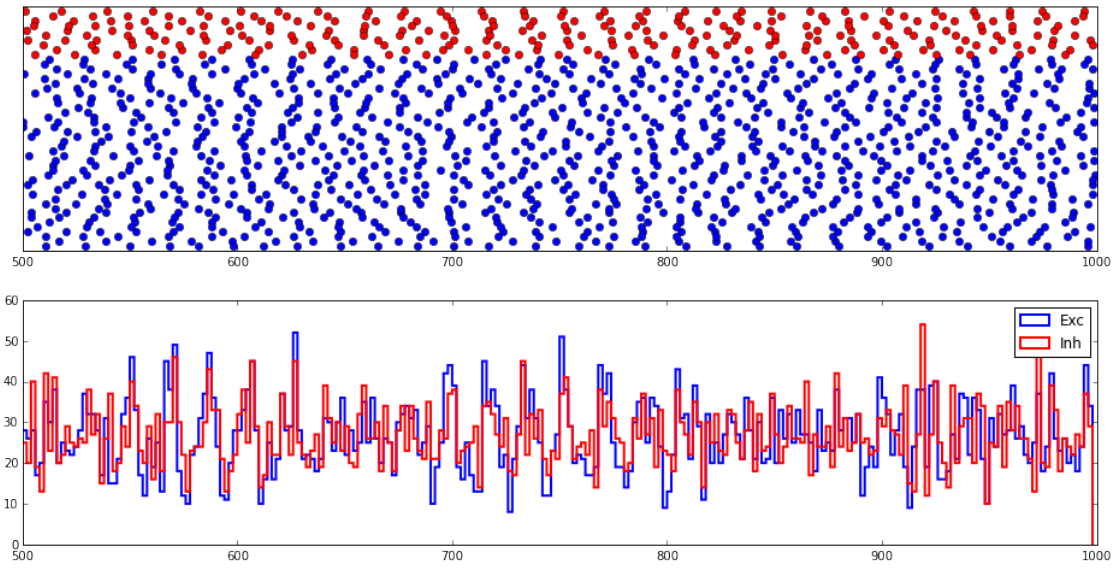
```
Excitatory rate: 14.29 Hz
Inhibitory rate: 14.46 Hz
```



The rate is too low now, but the firing pattern less bursty. We increase the input a bit:

```
In [27]: esp, isp = sim_brunel_delta(order=250, N_rec=250, g=5., eta=1.5, J=0.1, V_reset=10.0, num_threa
         plot_task4(esp, isp)
```

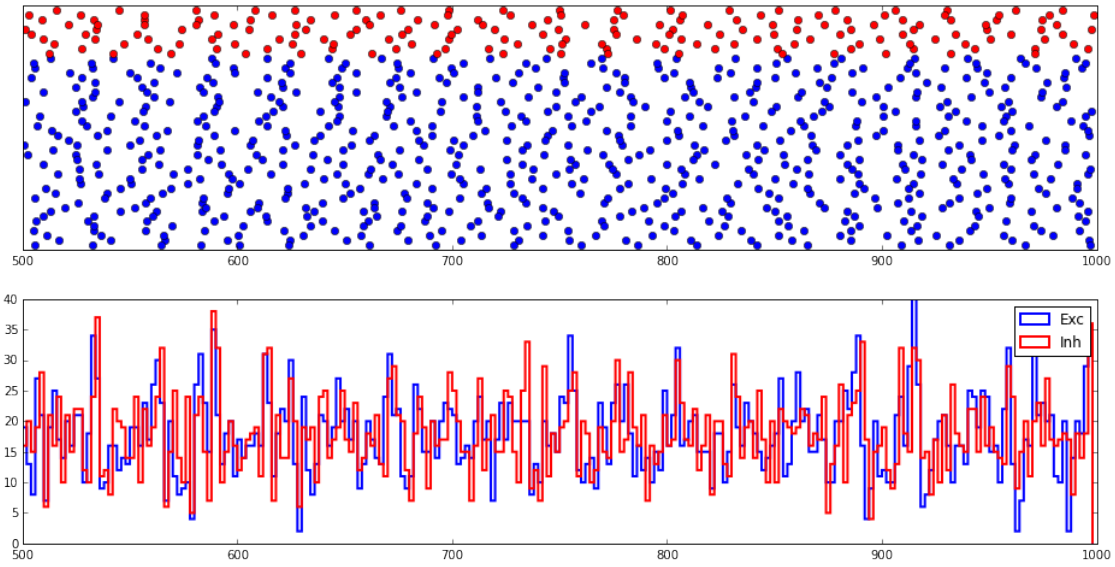
```
Excitatory rate: 53.56 Hz
Inhibitory rate: 53.81 Hz
```



Firing rate too high, reduce input

```
In [28]: esp, isp = sim_brunel_delta(order=250, N_rec=250, g=5., eta=1.25, J=0.1, V_reset=10.0, num_thre
         plot_task4(esp, isp)
```

```
Excitatory rate: 35.50 Hz
Inhibitory rate: 35.91 Hz
```



This is good enough. But one should check second order statistics ...

1.5 Task 5

```
In [29]: from brunel_delta import build_brunel_delta_plastic, sim_brunel_delta_plastic

In [30]: plt.rcParams['figure.figsize'] = (16, 12)

In [31]: def results(esp, isp, t_wgts, e_wgts, nrec=250):

             dt = t_wgts[1] - t_wgts[0]

             num_spikes = lambda sp, t: np.array([len(sp[(t[k] < esp.times) & (esp.times <= t[k+1])])
                                                  for k in range(len(t)-1)])

             rates = pd.DataFrame(
                        {'Time [ms]': t_wgts[1:],
                         'Excitatory rate [Hz]': num_spikes(esp, t_wgts)  / (dt/1000. * nrec),
                         'Inhibitory rate [Hz]': num_spikes(esp, t_wgts)  / (dt/1000. * nrec)})

             print rates

             plt.subplot(311)
             plt.hist(e_wgts, histtype='step', bins=100)

             plt.subplot(312)
             e_gid_min = esp.senders.min()
             i_gid_min = isp.senders.min()
             e_plot = (esp.senders < (e_gid_min + 40))
             i_plot = (isp.senders < (i_gid_min + 10))
             plt.plot(esp.times[e_plot], esp.senders[e_plot] - e_gid_min +  1, 'bo', markersize=2, mark
             plt.plot(isp.times[i_plot], isp.senders[i_plot] - i_gid_min + 41, 'ro', markersize=2, mark
             plt.ylim(0, 51)
             plt.yticks([])

             plt.subplot(313)
             bins = np.arange(0, 2000, 5.)
             plt.hist(esp.times.values, bins=bins, histtype='step', color='b', lw=2, label='Exc')
             plt.hist(isp.times.values, bins=bins, histtype='step', color='r', lw=2, label='Inh')
             plt.legend();
```

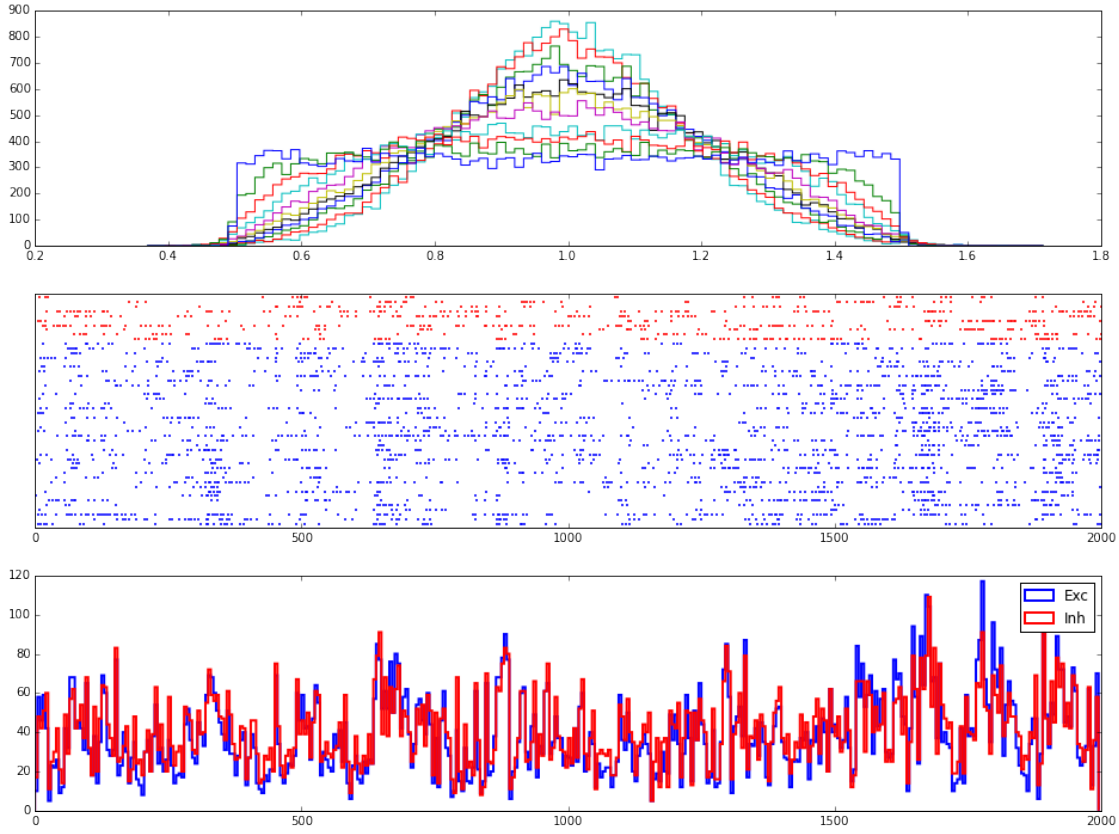
1.5.1 Simulation 1: Intermediate $\alpha = 2$

```
In [32]: esd, isd, enrn, inrn = build_brunel_delta_plastic(order=250, N_rec=250, g=5., eta=1.25, J=1., a

         t_step = 200.
         n_step = 10

         ini_wgts = nest.GetStatus(nest.GetConnections(source=enrn[:250], synapse_model='excitatory_plas
         e_wgts = np.nan * np.ones((len(ini_wgts), n_step+1))
         e_wgts[:, 0] = ini_wgts
         t_wgts = t_step * np.arange(n_step+1)
```

```
        for k in range(10):
            esp, isp, e_wgts[:, k+1] = sim_brunel_delta_plastic(200., esd, isd, enrn[:250], print_repo

        results(esp, isp, t_wgts, e_wgts)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Brunel network (Python)
Number of neurons : 1250
Number of synapses: 158000
       Exitatory  : 126250
       Inhibitory : 31250
Building time     : 0.09 s
```

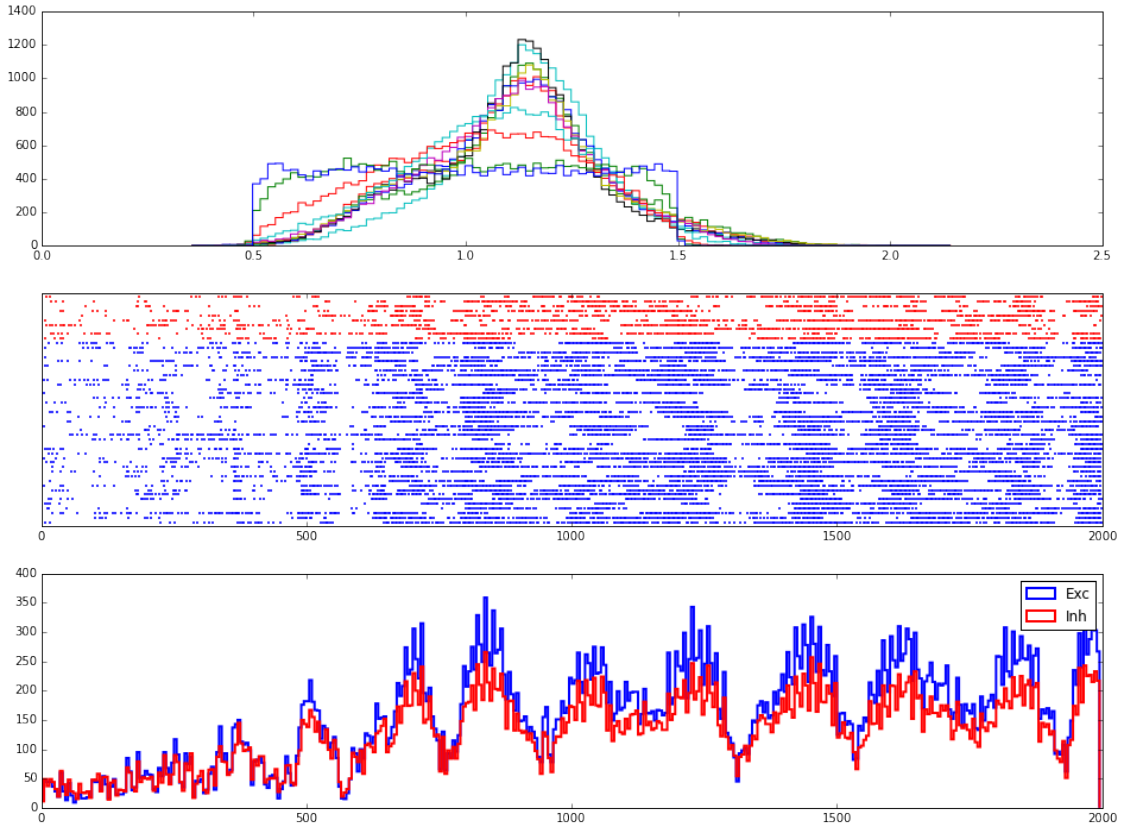
	Excitatory rate [Hz]	Inhibitory rate [Hz]	Time [ms]
0	27.88	27.88	200
1	27.62	27.62	400
2	24.36	24.36	600
3	33.56	33.56	800
4	30.22	30.22	1000
5	22.82	22.82	1200
6	30.76	30.76	1400
7	32.14	32.14	1600
8	43.00	43.00	1800
9	37.14	37.14	2000



1.5.2 Simulation 2: Low $\alpha = 1.5$

```
In [33]: esd, isd, enrn, inrn = build_brunel_delta_plastic(order=250, N_rec=250, g=5., eta=1.25, J=1., a

         t_step = 200.
         n_step = 10

         ini_wgts = nest.GetStatus(nest.GetConnections(source=enrn[:250], synapse_model='excitatory_plas
         e_wgts = np.nan * np.ones((len(ini_wgts), n_step+1))
         e_wgts[:, 0] = ini_wgts
         t_wgts = t_step * np.arange(n_step+1)

         for k in range(10):
             esp, isp, e_wgts[:, k+1] = sim_brunel_delta_plastic(200., esd, isd, enrn[:250], print_repo

         results(esp, isp, t_wgts, e_wgts)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Brunel network (Python)
Number of neurons : 1250
Number of synapses: 158000
       Exitatory  : 126250
       Inhibitory : 31250
Building time     : 0.09 s
   Excitatory rate [Hz]  Inhibitory rate [Hz]  Time [ms]
0                 28.98                 28.98        200
1                 58.24                 58.24        400
2                 71.78                 71.78        600
3                129.30                129.30        800
4                157.14                157.14       1000
5                160.18                160.18       1200
6                150.34                150.34       1400
7                168.62                168.62       1600
8                166.78                166.78       1800
9                176.34                176.34       2000
```



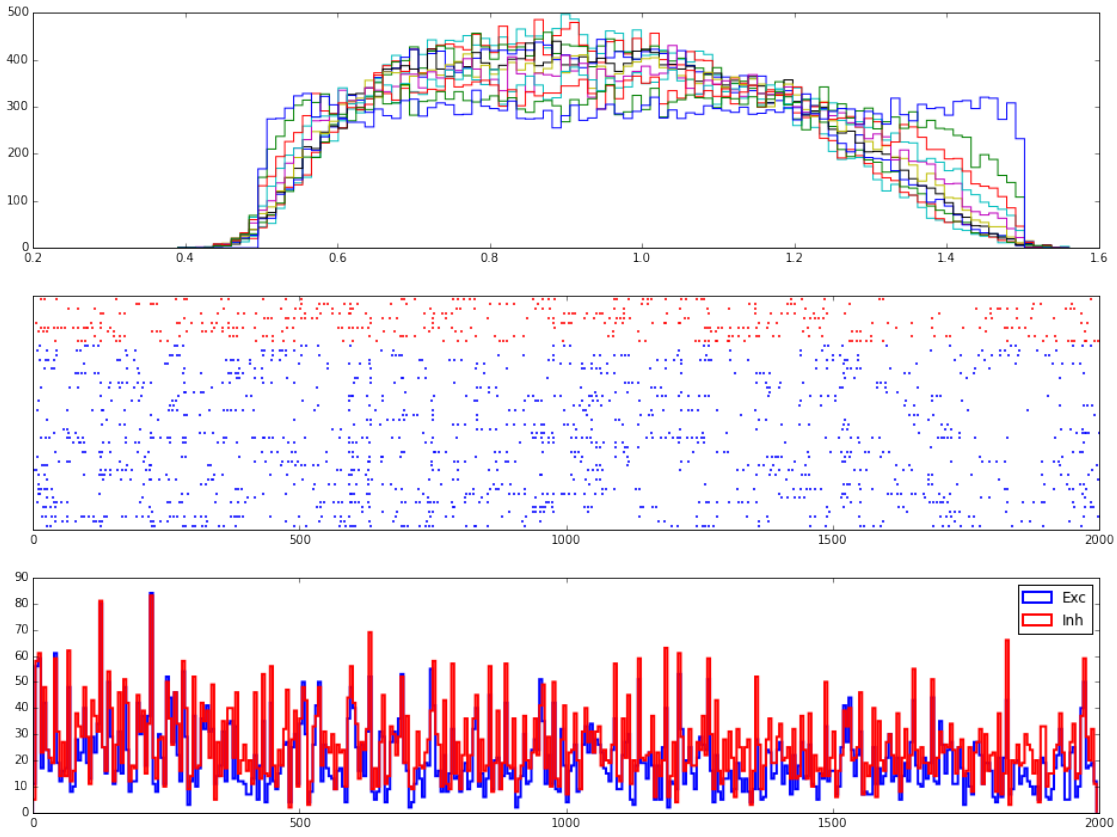
1.5.3 Simulation 3: High $\alpha = 3$

```
In [34]: esd, isd, enrn, inrn = build_brunel_delta_plastic(order=250, N_rec=250, g=5., eta=1.25, J=1., a

         t_step = 200.
         n_step = 10

         ini_wgts = nest.GetStatus(nest.GetConnections(source=enrn[:250], synapse_model='excitatory_plas
         e_wgts = np.nan * np.ones((len(ini_wgts), n_step+1))
         e_wgts[:, 0] = ini_wgts
         t_wgts = t_step * np.arange(n_step+1)

         for k in range(10):
             esp, isp, e_wgts[:, k+1] = sim_brunel_delta_plastic(200., esd, isd, enrn[:250], print_repo

         results(esp, isp, t_wgts, e_wgts)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Brunel network (Python)
Number of neurons : 1250
Number of synapses: 158000
```

```
       Exitatory  : 126250
       Inhibitory : 31250
Building time     : 0.10 s
```

	Excitatory rate [Hz]	Inhibitory rate [Hz]	Time [ms]
0	21.64	21.64	200
1	21.20	21.20	400
2	16.12	16.12	600
3	16.70	16.70	800
4	15.50	15.50	1000
5	15.48	15.48	1200
6	13.44	13.44	1400
7	13.80	13.80	1600
8	14.86	14.86	1800
9	12.82	12.82	2000

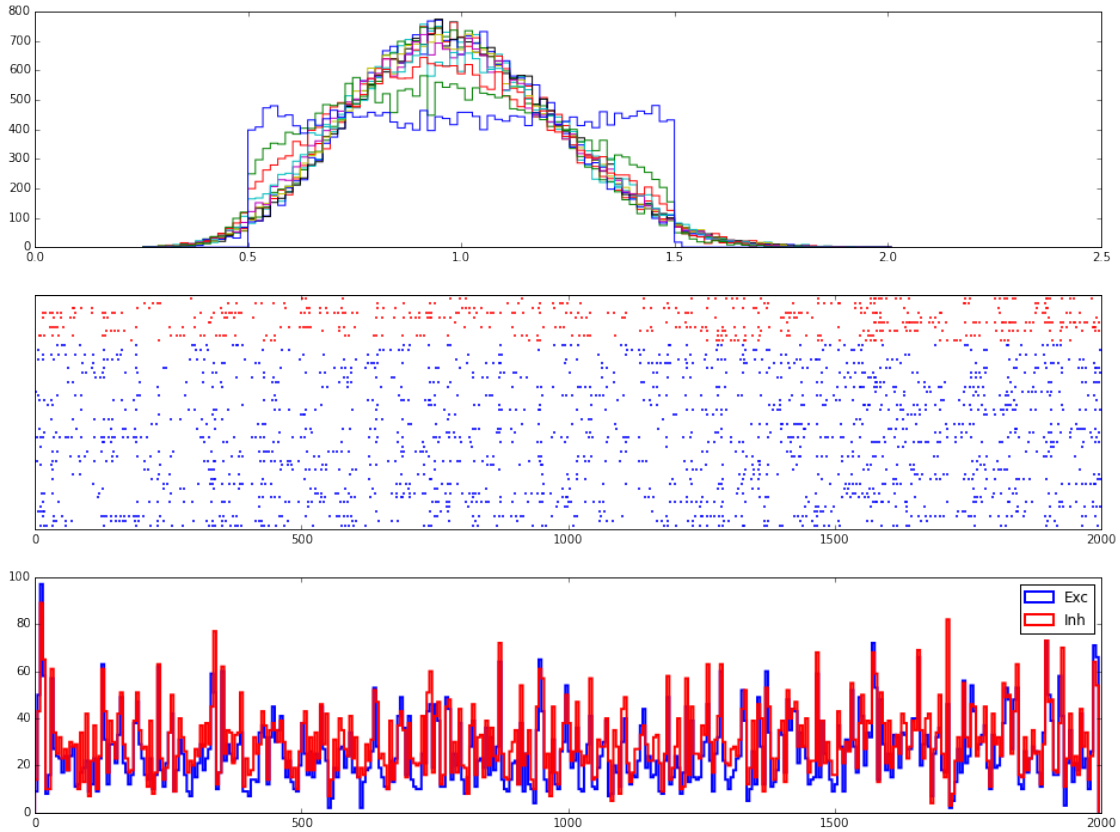


1.5.4 Simulation 4: Intermediate $\alpha = 2$, high $\lambda = 0.05$

```
In [36]: esd, isd, enrn, inrn = build_brunel_delta_plastic(order=250, N_rec=250, g=5., eta=1.25, J=1.,
                                                           lambd=0.05, alpha=2., Wmax=3., V_reset=10.0,

         t_step = 200.
         n_step = 10

         ini_wgts = nest.GetStatus(nest.GetConnections(source=enrn[:250], synapse_model='excitatory_plas
         e_wgts = np.nan * np.ones((len(ini_wgts), n_step+1))
```

```
        e_wgts[:, 0] = ini_wgts
        t_wgts = t_step * np.arange(n_step+1)

        for k in range(10):
            esp, isp, e_wgts[:, k+1] = sim_brunel_delta_plastic(200., esd, isd, enrn[:250], print_repo

        results(esp, isp, t_wgts, e_wgts)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Brunel network (Python)
Number of neurons : 1250
Number of synapses: 158000
       Exitatory  : 126250
       Inhibitory : 31250
Building time     : 0.09 s
   Excitatory rate [Hz]  Inhibitory rate [Hz]  Time [ms]
0                 21.72                 21.72        200
1                 19.10                 19.10        400
2                 18.18                 18.18        600
3                 19.74                 19.74        800
4                 18.34                 18.34       1000
5                 17.18                 17.18       1200
6                 19.70                 19.70       1400
7                 24.00                 24.00       1600
8                 22.64                 22.64       1800
9                 24.96                 24.96       2000
```



1.5.5 Simulation 5: Intermediate $\alpha = 2$, low $\lambda = 0.002$

```
In [37]: esd, isd, enrn, inrn = build_brunel_delta_plastic(order=250, N_rec=250, g=5., eta=1.25, J=1.,
                                                           lambd=0.002, alpha=2., Wmax=3., V_reset=10.0

         t_step = 200.
         n_step = 10

         ini_wgts = nest.GetStatus(nest.GetConnections(source=enrn[:250], synapse_model='excitatory_plas
         e_wgts = np.nan * np.ones((len(ini_wgts), n_step+1))
         e_wgts[:, 0] = ini_wgts
         t_wgts = t_step * np.arange(n_step+1)

         for k in range(10):
             esp, isp, e_wgts[:, k+1] = sim_brunel_delta_plastic(200., esd, isd, enrn[:250], print_repo

         results(esp, isp, t_wgts, e_wgts)
```

```
Building network
Connecting devices
Connecting network
Excitatory connections
Inhibitory connections
Brunel network (Python)
```

```
Number of neurons : 1250
Number of synapses: 158000
       Exitatory  : 126250
       Inhibitory : 31250
Building time     : 0.11 s
```

	Excitatory rate [Hz]	Inhibitory rate [Hz]	Time [ms]
0	26.48	26.48	200
1	34.52	34.52	400
2	35.64	35.64	600
3	32.16	32.16	800
4	32.70	32.70	1000
5	22.52	22.52	1200
6	29.38	29.38	1400
7	30.16	30.16	1600
8	34.30	34.30	1800
9	29.30	29.30	2000