Date - 14/08/2015

1. Initialised git repo at hamborger/brian.git
2. uploaded stable.py,unstable.py and stateSpace.py to github
3. Made a backup in the /storage under himanshu/brian
4. BRIAN version 1.3.1 is being used.
5. <u>Explanation on how sigma for a group is determined</u>:

   In stateSpace.py, function estimate_params is used to get sigma. The estimate_params function takes in two input arguments, mon 'spike monitor' and time_est 'initial burst time' specified in default_params structure. The initial_burst_t is 50msec. In estimate_params the spike times from the last layer are taken as input. Zip(*mon.spikes) puts the spike time lists from all the 100 neurons in one package. The spike times, which is 100 x runtime(90ms, step size 0.1), in a time window of 30ms + synaptic delay from the initial burst time are taken. A standard deviation of spike times is then calculated in that time window by finding the mean spike time.

   In short the logic is to determine the variation in spike times developed from the case if everything is synchronous in that time window.

Code snippets:

```
(91)  initial_burst_t=50 * ms
(196) net.run()  #Runs the network simulation
(153) i, times = zip(*mon.spikes)
(154) times = array(times) #datatype conversion from simple list to array
(155) times = times[abs(times - time_est) < 15 * ms]
(162) return (len(times),times.std())
(197) (newa, newsigma) = estimate_params(net.mon[-1], params.initial_burst_t)
#newsigma is the standard deviation for the last layer.
```

Date - 17/08/2015

1. Made changes in the stateSpace.py default_params initial_burst_t set to 10ms not 50ms.
2. In stable.py added user input for a,sigma
3. StateMonitor(P,'v',record=[1,101,201,301,401,501,601,701,801,901])
   Instead of SpikeMonitor this function will give the voltage trace for the selected neurons out of total 1000 neurons.

   X axis : runtime(in msec) ,Y axis : Membrane voltage(in mV)
   Ymax : threshold voltage
   Ymin : resting potential

4. I reproduced figure 2C from Diesmann et. al. 1999 paper. In stateSpace.py, sigma_vs_a function calculates the probability or percentage of spikes occurring in the last group by finding new a and dividing it by the number of neurons in that layer which is 100. It does so by changing the input volley's neuronal number 'a' fixing sigma. Hence we get response probabilities accordingly for different temporal dispersion in input.

In the figure below it is evident that violet (sigma=0ms) has the highest probability of response compared to yellow (sigma=5ms) where each line is evaluated for 10 different values of a.

Plotted for 50 equally spaced a's from 0 to 100.

Date - 18/08/2015

1. Plotted sigma out vs sigma in for different numbers of neurons ‘a’. It is seen that sigma is not dependent on ‘a’ where blue is 45, green 65 and red 100.

2. To reproduce figure 3a, figure 4(a,b,c,d)
3. Figure 3a:
   In this figure the output of spikes from the last group is measured with respect to input spikes having fixed temporal dispersion of 0(Blue)-5ms(Yellow). This plot has lines each containing 10 points.

4. Figure 3c:
   State space plot generated using stateSpace.py with a grid of 10x10 size in the alpha-sigma phase plane. In the state space plot the top left region is converging to the stable fixed point defined as propagation with increasing synchronisation. The central region is the saddle node where the propagation can be either stable or unstable defined as propagation with increasing spread or dispersion in spike times. Neuron_multiplicity is 1 and 50 for the corresponding figures below. Neuron_multiplicity increases the probability of stable propagation by changing the number of neurons in a layer by a factor. Then new a is calculated by dividing the newa, returned by estimate_param function, by neuronal_muliplicity scaling it back to 100 or w as set in the model_parameters. A neuron_multiplicity of 50 means that there are 5000 (100 x 50) neurons per layer. It is observed that as we increase the multiplicity number the central region of the state space corresponding to a saddle node shrinks.

Synfire chain state space
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Synfire chain state space
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

5. Figure 4a,4b,4c,4d:
   In the plots below the role of neurons per layer w is being investigated. It is seen that as w is increased from 80 to 90 a stable fixed point comes into the picture. The stable fixed point shifts top-leftwards as we further increase the w from 90 to 110.

Synfire chain state space
100
80
60
40
20
0
a
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
sigma (ms)

Synfire chain state space
100
80
60
40
20
0
a
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
sigma (ms)

Synfire chain state space
120
100
80
60
40
20
0
a
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
sigma (ms)

Date - 19/08/2015

1. Plot isoclines as a function of w. (Methods: Isoclines for figure 4(a,b,c,d))
2. Isoclines :
   a. Sigma-isocline:
      1. Not dependent on w.
      2. Maintaining state of temporal spread irrespective of a.
   b. a-isocline:
      1. Dependent on w
      2. Maintaining state of having a spike number which doesn't change with sigma.
3. Methods : ?
4. a- isocline is plotted by picking all those set of points for which the a_out-a_in == 0 or change in a is zero.
5. sigma-isocline is plotted by picking all those points for which sigma_out-sigma_in == 0.
6. Since it is being done on a limited number of grid points it is difficult to exactly arrive at the isoclines. So a coloring of the region is done the blue region is where the 'a' increases and red where the 'sigma_out' decreases. The boundary of the two regions is where the isoclines lie.

   The figure below is generated for a w of 100 (neuron_multiplicity 50). Red region is all those a_in,sigma_in for which the sigma_out decreases and Blue region is where a_out increases. The boundary of each region marks the respective nullclines.

7. Plot for w = 80 (neuron_multiplicity 50):

8. Plot for w = 90 (neuron_multiplicity 50):

9. Plot for w = 100 (neuron_multiplicity 50):

10. Plot for w = 110 (neuron_multiplicity 50):

Date - 20/08/2015

1. Improved the coloring algorithm to find out overlapping grid points by giving them a unique color.

2. Plot for w = 110 (neuron_multiplicity 50)

3. Plot for w (neurons/layer) = 100 (neuron_multiplicity 50)

4. Plot for w (neurons/layer) = 90 (neuron_multiplicity 50)

Date - 21/08/2015

1. Update mid-term report with figures generated using brian and attach codes.
2. Leaky integrate and fire neuron with Potassium conductance dynamics
3. Voltage trace of a single LIF neuron (runtime 50ms):

4. K conductance trace (runtime 50ms,a = 1, sigma = 0):

5. Noisey synaptic conductance trace (runtime 50ms,a = 1, sigma = 0):

6. Voltage bump due to a single spike (runtime 50ms,a = 1, sigma = 0):

   Neurons were initialized at resting voltage, -70mV. Voltage difference created due to single spike was calculated by taking a difference between the neuron seeing the

spike and a neuron with no synaptic input.

Date - 23/08/2015

1. Latex tutorial
2. Equations were written in tex and a cropped image from pdf was inserted in report.

$$\frac{dV}{dt} = \frac{(-(V - Vr) + x)}{taum} \tag{1}$$

$$\frac{dx}{dt} = \frac{(-x + y)}{taupsp} \tag{2}$$

$$\frac{dy}{dt} = \frac{-y}{taupsp} + 25.27 + \sqrt{39.24} * nrand \tag{3}$$

where (1) is the voltage derivative, (2) conductance derivative and (3) exponential synapse.

Date - 24/08/2015

1. Intragroup connections were made with varying probabilities.
2. Report editing.
3. Within the group connections were all excitatory.
4. Plot with a = 53, sigma = 0, weight = 2.43:

5. Plot with a = 53, sigma = 0, weight = 1.43:

Date - 25/08/2015:

1. In intragroup connected network by excitatory synapses it was found that the spiking activity remained active long after the pulse was given, This happened because of neurons within the group excitating each other acting as a pulse source themselves.
2. A phase plane analysis was done to look at the nature of synchronization attained by this network.
3. State space (Grid = 10, nm = 10, weight = +34, w = 20):

4. State space (Grid = 10, nm = 10, weight = +34, w = 25):

5. State space (Grid = 10, nm = 10, weight = +34, w = 30):

6. It is seen that at w = 30 the activity converges to a highly spiking but asynchronous point.
7. Similarly to inspect the role of inhibition, the intragroup synapses were made purely inhibitory with strength of inhibition within group same as negative of strength of excitation between the groups.

8. Plot with a = 53, sigma = 0, weight = -2.43:

9. Plot with a = 53, sigma = 0, weight = -4.86:

10. Plot with a = 53, sigma = 0, weight = -5.43:

11. It is seen that as the inhibition strength is increased within the group the propagation becomes stable to unstable. Because of inhibition the input to neighbour group goes down which causes less excitation in that group and gets dominated by constant inhibition from within the group.
12. Again the behaviour of the network with inhibition is investigated by looking at the phase portrait.

13. State space (Grid = 10, nm = 10, weight = -34, w = 110):

compared with the activity of network lacking intragroup connection,

14. State space (Grid = 10, nm = 10, weight = -34, w = 100):

15. State space (Grid = 10, nm = 10, weight = -34, w = 100):

comparing it with w = 90 for no intragroup connection case,

16. From the comparisons it is evident that inhibition within the group reduces the green region which encloses the stable fixed point and saddle node.

Date - 26/08/2015

1. Have intragroup connections with both inhibition and excitation.
2. An algorithm to pick the grid points from the boundaries of blue, red and green regions in state space plot for any network.
3. Fit a polynomial to the boundaries and get the point of intersection.
4. Intersecting points are the eigenvalues from which a characteristic polynomial will be determined.

Date - 31/08/2015

Excitation-Inhibition balance and how does it affect the nature of propagation.

1. To set up a feed-forward network with groups containing 88% excitatory and 12% inhibitory neurons mimicking the ratio found in the Cortex (Ref:Braitenberg, V. & SchuÈz, A. Anatomy of the Cortex (Springer, Berlin, 1991)).
2. In state space if the stable fixed point shifts then how is the shift a function of the weights of E-I.
3. Read an article on E-I balance:
   Formation of excitation-inhibition balance: inhibition listens and changes its tune
   Huizhong W. Tao 1,2 , Ya-tang Li 1 , and Li I. Zhang 1,3
   Cell Press 2014
   http://www.ncbi.nlm.nih.gov/pubmed/25248294

Date - 01/09/2015

1. Updated the github repository separating the development and stable version of codes.
2. E-to-I ratio 1:1, weight_E = 4.86 mV, weight_I = -4.86 mV :

The excitatory and inhibitory neurons of the first layer received excitatory and inhibitory spikes with certain a and sigma as the input. The weight of connections were same for both excitatory and inhibitory neurons but differed in sign.
This was then changed to excitatory only input to see response in the inhibitory neurons of the first group with the exact same weights for the below plots.

Group number
Time (ms)
0
2
4
6
8
10
12
0
10
20
30
40
50
60
70
80
90

Group number
Time (ms)
0
2
4
6
8
10
12
0
10
20
30
40
50
60
70
80
90

State space with w = 100, E = 88% of w, I = 12% of w, neuron_multiplicity = 50 :

State space with w = 110, E = 88% of w, I = 12% of w, neuron_multiplicity = 50 :

Date - 02/09/2015

1. A function sfp_vs_inh was written to plot the changing fixed points w.r.t. increasing inhibition in the FFN.
2. The inhibition is increased from 0 to 1 in ratio with excitation.
3. A 3D plot is created with X axis as a, Y axis as sigma and Z axis as inhibition weights.
4. Movement of stable fixed point and saddle node for every different weight is plotted separately on same plane.

Date - 03/09/2015

1. In the file dev/balEI.py major changes were made in the way connection is made.
2. It was found that variables had same name for excitatory and inhibitory input connect.
3. Original connection method of the stable/stateSpace.py in dev/balEI.py was replaced with a simpler and a clear method from dev/propagation.py.
4. Next a plot for increasing excitation versus shift of fixed points was generated.
5. <u>The fixed points are determined by the following coarse method</u>:
   1. Boundaries of the green grid is read into a dictionary.
   2. Dictionary has 'a' as the key and the different sigma as the values.
   3. Stable fixed point is the left top most point of the green grid. It is determined by finding the max of keys and min of sigma. The value of calculated 'newa' and 'newsigma' for that grid point becomes the stable fixed point as it is the closest converging point to the real fixed point.
   4. Saddle node is the right bottom most point determined by finding the min of keys and max of sigma. The new calculated values at that point are taken as saddle node.
6. For a network with neurons per layer, w, 125 where 110(88%) were excitatory and 15(12%) inhibitory neurons, an excitation versus fixed point shift was done with excitation ranging from 34 to 50 (see model_params, we) keeping wi fixed at -34.

7. In the plot above Z axis is the increasing weight of excitation, X axis (0.4-2.0) is the sigma and Y axis, a.
8. It is seen that as the excitation increases the saddle node and stable fixed point separate.

Date - 04/09/2015

1. A key parameter num_layer was changed to 2 from 1.
2. net.mon_E holds the spikes of the last group (2nd one in my case).
3. In the reinit function a bug was found which initialized values to default params inspite input being provided.
4. All the simulations from now on are done with neuron_multiplicity 1.
5. fundamentalunits.DimensionMismatchError gets raised unreasonably when coloring the grids. To avoid termination of program due to encountered error handling of the error is done which ignores that particular grid point coloring it black.
6. Plot with w = 125, E = 110, I = 15, weighti = -18:

7. Plot to check propagation with a = 90, sigma = 0ms and a = 100, sigma = 3ms in EI FFN:

3.0
2.5
2.0
1.5
1.0
0.5
0.0
Group number
0
10
20
30
40
50
60
70
80
90
Time (ms)

3.0
2.5
2.0
1.5
1.0
0.5
0.0
Group number
0
10
20
30
40
50
60
70
80
90
Time (ms)

Date - 05/09/2015

1. For the EI-FFN containing 125 neurons per layer with 110 excitatory and 15 inhibitory neurons a 3D plot was generated to investigate how fixed points separated away from each other with decreasing inhibition.
2. Below plots are phase portrait with decreasing inhibition.
3. Weight of inhibtion for the below are 0,-18,-49,-109,-127 in order

Isoclines
120
100
80
60
40
20
0
a
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
sigma (ms)

Isoclines
120
100
80
60
40
20
0
a
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
sigma (ms)

Isoclines
120
100
80
60
40
20
0
a
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
sigma (ms)

Isoclines
120
100
80
60
40
20
0
a
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
sigma (ms)

4. 3D plot to see the separation of fixed points:

stable fixed points
saddle node points
0
−20
−40
−60
−80
−100
−120
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
120

stable fixed points
saddle node points
0
−5
−10
−15
−20
−25
−30
−35
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
50
60
70
80
90
100
110
120

stable fixed points
saddle node points
0
−5
−10
−15
−20
−25
−30
−35
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
120
110
100
90
80
70
60
50

5. From the above plots it is seen that as the inhibition is decreased the fixed points tend to come closer until disappearance. All the simulations for which no fixed point showed up the point (0,0) was assigned in the plot.

Date - 07/09/2015

1. A second degree polynomial function was fit to get the borders of blue and red region for EI-FFN and E-only FFN.

2. E-only FFN with w = 100:

3. E-only FFN with w = 110:

Isoclines
a : y = -7.61904761905e-07 + 0.000146666666667x + -0.00422857142857x^2
sigma :y = 1.09897028836e-21 + -1.71428571429e-05x + 0.0024x^2
a
sigma (ms)
0
20
40
60
80
100
120
140
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

4. E-only FFN with w = 90:

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

5. EI-FFN with w = 110, nE = 97, nI = 13:

6. Implementing neuron_multiply for EI-FFN. In all the previous plots the network is balanced at higher inhibition to excitation ratio since only 12% of neurons make inhibitory connection with other neurons. Introducing neuron_multiply increases the number of neurons in a layer hence the weight for inhibition per neuron gets reduced lowering the ratio.
7. Encountered fundamentalunits.DimensionMismatchError frequently while implementing 6. All changes were reverted to an earlier version.
8. Since nullclines are determined by curve fitting a more closely packed grid was chosen to increase the number of points in the boundary for a better curve fit.
9. By curve fitting it was found that on increasing inhibition the spiking activity became less synchronous.
10. Below are state space and nullclines plot for increasing weights from 0 to -70.
11.

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a-nullcline
sigma-nullcline
a
sigma (ms)
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

The respective nullclines and their polynomial form in the plots below accordingly.

Isoclines
a : y = -8.5108204722e-07 + 0.000169278849241x + -0.00455387631911x^2
sigma :y = 2.25534931417e-07 + -4.5019910667e-05x + 0.00263515140503x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -1.05830602036e-06 + 0.000198876142832x + -0.00562070381232x^2
sigma :y = 1.67034872917e-07 + -3.79925808161e-05x + 0.002445470485x^2
200
150
100
50
0
a
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
sigma (ms)

Isoclines
a : y = -9.10342654418e-07 + 0.000182466843793x + -0.0052888522589x^2
sigma :y = 1.61253690665e-07 + -3.41370277841e-05x + 0.00225054553713x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -9.10180324937e-07 + 0.000180358493753x + -0.00522947814213x^2
sigma :y = 3.03925009807e-07 + -5.51321068968e-05x + 0.0029702488799x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -1.08206502699e-06 + 0.000198652146139x + -0.00565902488599x^2
sigma :y = 6.60018307077e-08 + -2.35357710652e-05x + 0.00195207102595x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -6.63363332996e-07 + 0.000144322479523x + -0.00412368793609x^2
sigma :y = 1.83483712895e-07 + -4.0287076993e-05x + 0.00257797353044x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -8.3178799379e-07 + 0.000170204631706x + -0.0051628372434x^2
sigma :y = 6.80665386548e-08 + -2.33601332425e-05x + 0.00196949980385x^2
a
sigma (ms)
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -1.06775385349e-06 + 0.000192706697918x + -0.00569432480534x^2
sigma :y = -8.9470677706e-08 + 7.17692482398e-07x + 0.00116287060475x^2
a
sigma (ms)
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -7.3546002501e-07 + 0.000154101152545x + -0.00474730007658x^2
sigma :y = 3.55818002877e-08 + -1.60917556212e-05x + 0.001649115603x^2
a
sigma (ms)
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -6.20552601421e-07 + 0.000134123963031x + -0.00409337281455x^2
sigma :y = 5.83624113036e-08 + -1.87738663033e-05x + 0.00173817950571x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -6.78609526049e-07 + 0.000141679958184x + -0.00439533609435x^2
sigma :y = 5.92571180806e-08 + -2.21076538724e-05x + 0.00199737567361x^2
a
sigma (ms)
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -1.38854679803e-06 + 0.000238722032417x + -0.0075948320356x^2
sigma :y = 2.18308453603e-07 + -4.56906654554e-05x + 0.00287613387566x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -1.26420228826e-06 + 0.000219961663753x + -0.00703359510567x^2
sigma :y = 1.57055451173e-07 + -3.54315996669e-05x + 0.00244970198005x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Isoclines
a : y = -1.46099111616e-06 + 0.000245751336786x + -0.00795005010315x^2
sigma :y = 1.18927177751e-07 + -2.68356423651e-05x + 0.00205715474986x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

The plot below shows how the fixed points evolve with increasing inhibition in the EI-FFN.

It is seen that the blue line drifts rightward indicating that the decreasing inhibition in the EI-FFN is causing more synchronous spiking and that the balance of the network is more towards excitation.

Date - 08/09/15

1. Plot trajectories for some grid points in state space.
2. Implement neuron_multiply for layer 2. Neuron_multiply increases the independent sample size in a layer which helps in better estimation of sigma and a.
3. Have probabilistic connections between layers.

Date - 09/09/2015

1. Implemented neuron_multiply for layer n, where n is 2.
2. Below is the set of plots with neuron_multiply = 1 and 50 for weight of inhibition,wi = -34
3. wi = -34, neuron_multiply = 1:

Isoclines:

Isoclines
a : y = -8.57142857143e-07 + 0.000193714285714x + -0.00725714285714x^2
sigma :y = 9.52380952381e-08 + -2.7619047619e-05x + 0.00251428571429x^2
a
sigma (ms)
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

4. wi = -34, neuron_multiply = 50:

Isoclines:

Isoclines
a : y = -1.74603174603e-06 + 0.000282222222222x + -0.00887619047619x^2
sigma :y = 2.42424242424e-07 + -5.50116550117e-05x + 0.00377156177156x^2
a
sigma (ms)
0
50
100
150
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

5. wi = -34, neuron_multiply = 50 (amax = 150):

Date – 10/09/2015:

1. A function propTrace to plot the propagation on the a-sigma grid.
2. Spikemonitor is on for all the groups/layers and the number of layer is 10.
3. Neuron_multiply is implemented as a for loop running the network n times.
4. Delay is added to initial_burst_time to center the window.
5. A plot showing propagation trace with wi = -30, neuron_multiply = 100:
6. Propagation trace with wi = -90, neuron_multiply = 100:

Date – 11/09/2015:

1. Implemented propTrace inside fp_vs_Inh function.
2. Neuron_multiply and inhibition weight are taken as input parameters from fp_vs_Inh
3. Below are plots showing isoclines and propagation from few points.
4. In the plots increase in inhibition is from 0 to -100 with neuron_multiply 50.
5.

Date – 13/09/2015:

1. To check for balance of excitation and inhibition previously, EI-FFN was created. It was found that inhibition in EI-FFN only affected the spike count which was evident from the a-isocline moving leftwards as inhibition increased. This movement of a-isocline is similar to the case when neurons per layer ‘ w’ is changed.

2. The behavior is completely noise driven and noise parameters like number of noisy neurons, proportion of excitatory and inhibitory noisy neurons, rate of inhibition and excitation will be investigated.

Date – 14/09/2015:

1. In the functions fp_vs_inh and propTrace input parameters were changed from params.wi to params.noise_inh_rate.
2. Inhibition can be increased by either increasing the proportion of inhibitory neurons, by increasing the weight of inhibition or by increasing the rate of inhibition.
3. The inhibition rate increases the incoming inhibitory current by the following formula.
noise_mu = noise_neurons * (noise_exc * noise_exc_rate - noise_inh * noise_inh_rate) * psp_peak * we
where noise_mu is the mean noise current to every neuron in the network except the input group, noise_exc & noise_inh are the proportion of excitatory and inhibitory neruons, 0.88 & 0.12 respectively.
4. Below are plots with increasing frequency of inhibition from 2 to 20 Hz for 12% of the neurons which are inhibitory.

5. Below plot shows the change in fixed points with changing rate of inhibition:
6. It is seen that inhibition rate change from 2 Hz to 20 Hz hasn't affected the nature of fixed points. It could be that greater amount of inhibition will affect which can be achieved by increasing the proportion of inhibitory neurons or by further increasing the rate of inhibition.

Report for 10th to 14th is kept in the offline 13-09-2015.docx.
(link : Download Offline 13-09-2015.docx)