Computational Cognitive Science (CS786A), Spring 2019
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Bhavy Khatri
Roll Number: 150186
Date: January 28, 2019

QUESTION 1

A vector symbol $\mathbf{b}$, a symbol in blackboard font $\mathbb{R}$, a symbol in calligraphic font $\mathcal{A}$, some colored text
In this question I had to train the boolean function $f(a, b, c)$ for five different boolean functions. The following neural architecture was used:

1. 3 input nodes for Input layer.
2. 4 input nodes in single hidden layer with sigmoid activation function.
3. A single node in the output layer.

For updating the weight parameters of the model, backpropogation gradient descent update was used with the update equation as $\mathbf{w}^{t+1} = \mathbf{w}^t - \eta grad_{\mathbf{w}}(\mathcal{L})$. The learning rate was taken to be 0.1.After running the update algorithm for 10000 epochs following results were obtained.

Actual	0.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0
Predicted	0.081	0.946	0.958	0.945	0.039	0.035	0.021	0.988
Actual	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000
Predicted	0.006	0.012	0.003	0.960	0.006	0.032	0.038	0.963
Actual	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
Predicted	0.988	0.245	0.999	0.248	0.958	0.021	0.629	0.002
Actual	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Predicted	0.041	0.000	0.956	0.000	0.016	0.012	0.000	0.000
Actual	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000
Predicted	1.000	0.915	0.923	0.914	0.072	0.068	0.089	0.004

As you can see that the predicted boolean values are approaching close to the actual ones. This experiment also give us an idea that the perceptron algorithm is able to learn boolean function of 3 variables.

Computational Cognitive Science (CS786A), Spring 2019
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Bhavy Khatri
Roll Number: 150186
Date: January 28, 2019

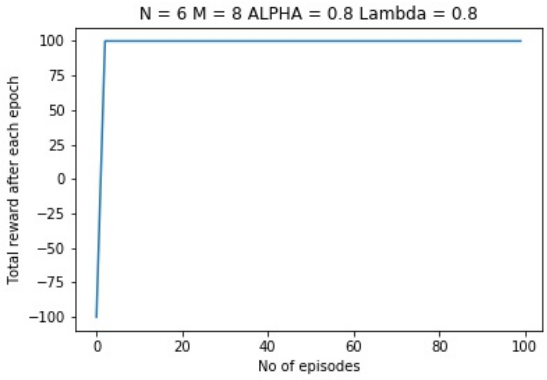
QUESTION 2

For the following matrix optimal path obtained after applying the Q-learning algorithm is as follows:

```
[[-1.  1.  1.  0.  0.  1.]
 [ 1.  1.  1.  0.  1.  1.]
 [ 1.  1.  1.  1.  1.  1.]
 [ 1.  1.  0.  1.  0.  1.]
 [ 0.  1.  1.  1.  1.  1.]
 [ 0.  1.  1.  1.  0. -2.]]
```

```
(0, 0)->(0, 1)->(1, 1)->(2, 1)->(2, 2)->(2, 3)->(2, 4)->(2, 5)->(3, 5)->(4, 5)->(5, 5)->
```

Note that -1, -2, 1, 0 represent the starting point, goal point, frozen lake and hole respectively in the frozen lake problem. And the second figure denote the indices of the path to be followed to reach the goal position. Also the plot of reward with no of episodes is obtained as follows:

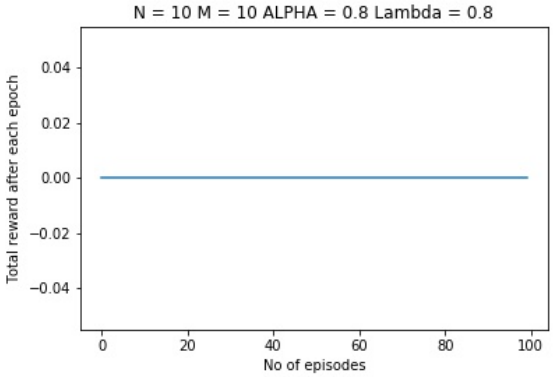


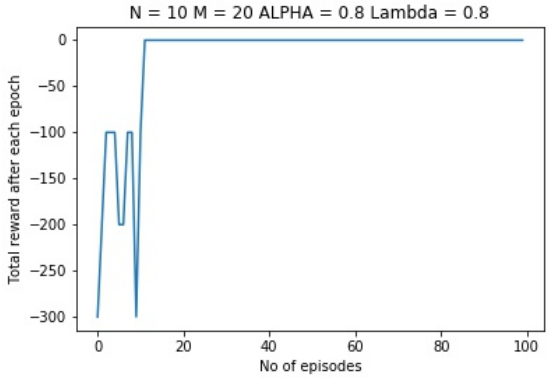
Now we will look at how learning performance changes with M, N, α, λ. For this experiment the following matrix with $N = 10, M = 20$ was taken into account:

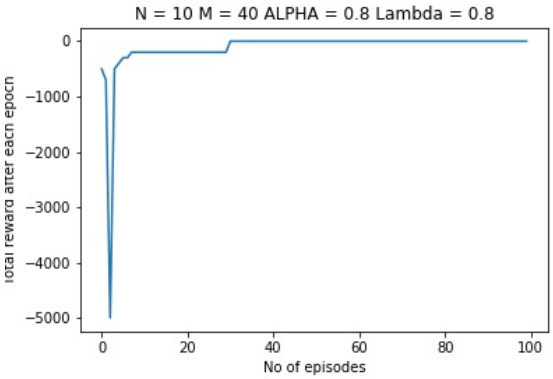
```
[[-1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]
 [ 0.  1.  1.  1.  1.  1.  0.  1.  1.  1.]
 [ 1.  1.  1.  1.  1.  1.  1.  1.  1.  0.]
 [ 1.  1.  1.  1.  1.  0.  1.  0.  1.  1.]
 [ 1.  1.  1.  1.  1.  0.  1.  1.  1.  1.]
 [ 1.  0.  1.  1.  1.  1.  1.  1.  1.  1.]
 [ 1.  1.  1.  1.  1.  1.  1.  1.  0.  1.]
 [ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]
 [ 1.  1.  1.  1.  1.  1.  1.  0.  1.  1.]
 [ 1.  1.  1.  0.  1.  1.  1.  1.  1. -2.]]
```

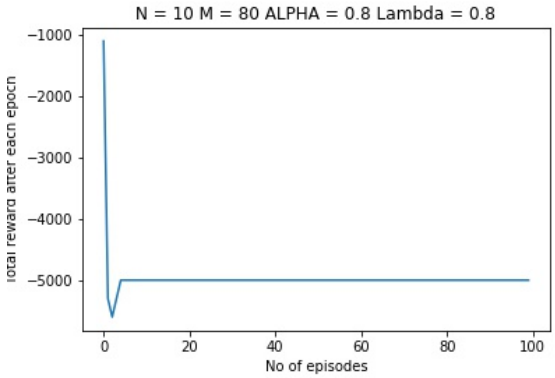
- **Change with $M = [10, 20, 40, 80]$:** For a fixed size as no of holes increases but the number of frozen points remain constant it becomes harder and harder for the learning algorithm

to find the optimal path. So the number of episodes increases to converge to a certain total reward point.

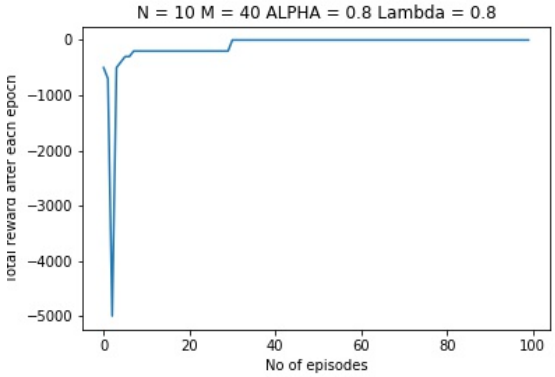


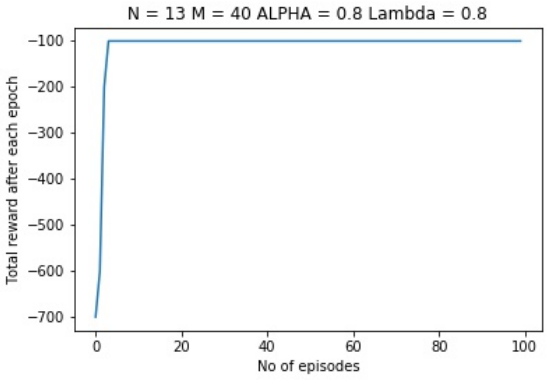


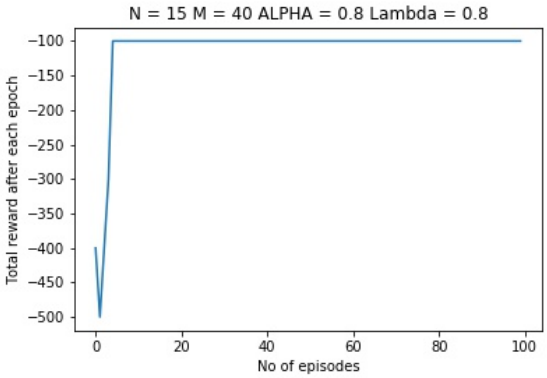


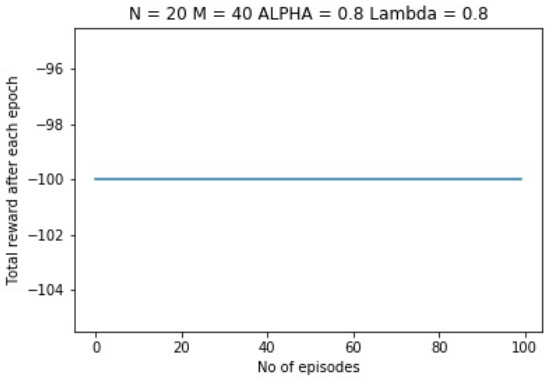


- **Change with** $N = [10, 13, 15, 20]$**:** For a fixed no of holes as the size increases but the number of holes points remain constant it becomes easier and easier for the learning algorithm to find the optimal path. This is due to the fact the ratio of frozen points to that of hole points increases with size. So the number of episodes decreases to converge to a certain total reward point.

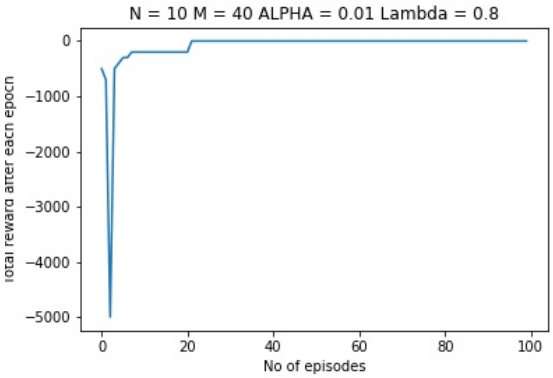


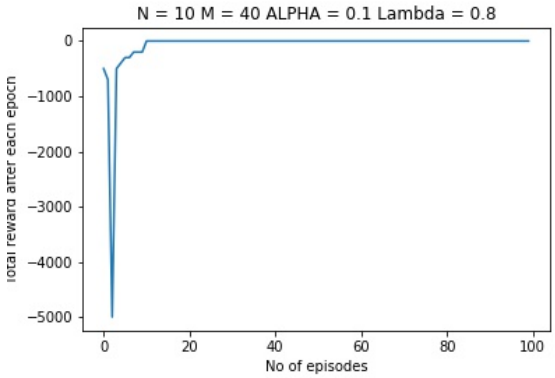


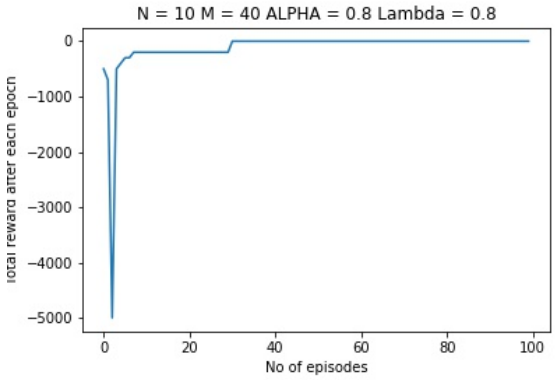


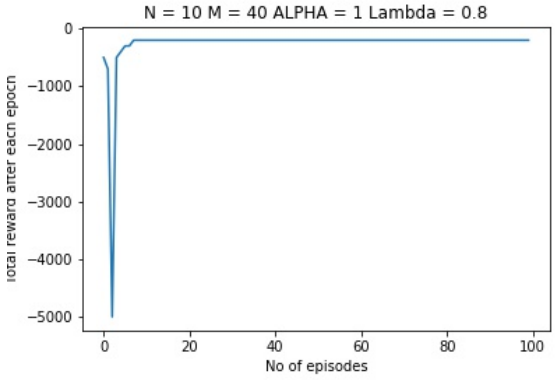


- **Change with** $\alpha = [0.01, 0.1, 0.8, 1]$**:** This one is the most easy one to interpret as with smaller learning rate algorithm becomes slower to find the optimal path from starting point to goal point.

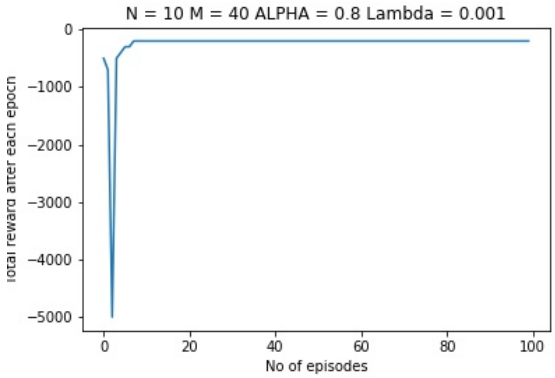


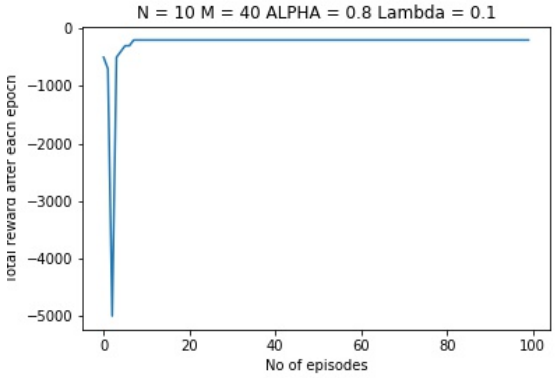


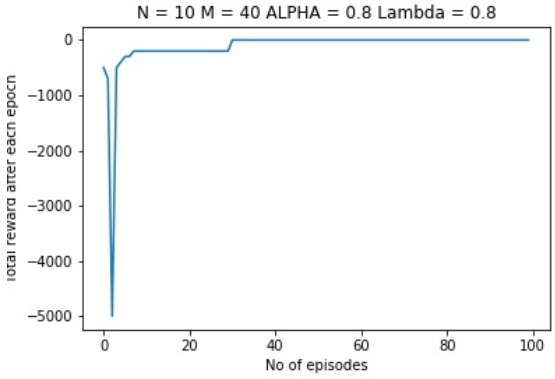


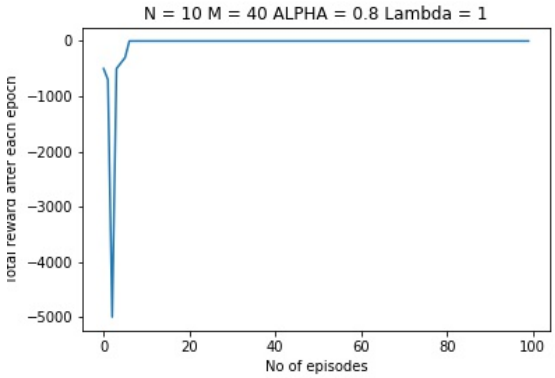


- **Change with $\lambda = [0.001, 0.1, 0.8, 1]$:** Discount factor doesn't have any affect on the learning performance.


N = 10 M = 40 ALPHA = 0.8 Lambda = 0.001
Total reward after each epocn
No of episodes


N = 10 M = 40 ALPHA = 0.8 Lambda = 0.1
Total reward after each epocn
No of episodes


N = 10 M = 40 ALPHA = 0.8 Lambda = 0.8
Total reward after each epocn
No of episodes

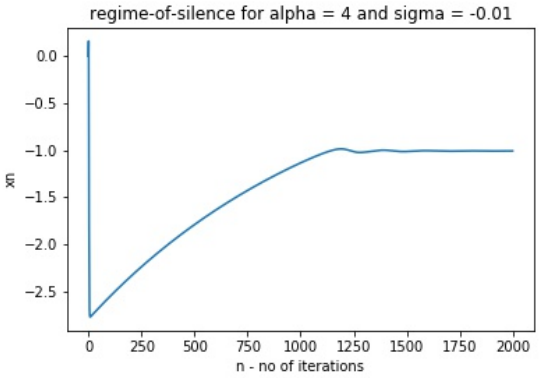

N = 10 M = 40 ALPHA = 0.8 Lambda = 1
Total reward after each epocn
No of episodes

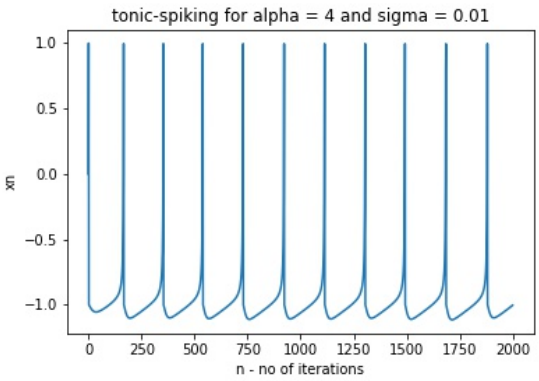
Computational Cognitive Science (CS786A), Spring 2019
Indian Institute of Technology Kanpur
Homework Assignment Number 1

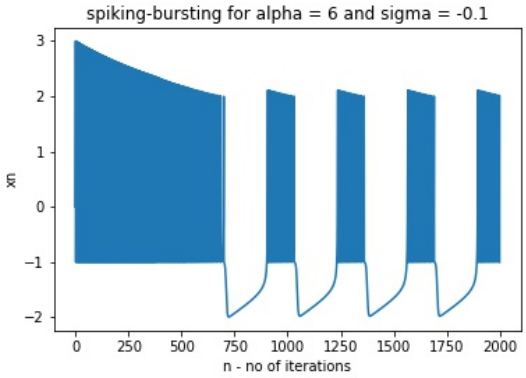
Student Name: Bhavy Khatri
Roll Number: 150186
Date: January 28, 2019

QUESTION 3

The following plots were obtained for three different neuron behaviors namely regime of silence, tonik spiking and spiking bursting. The alpha and sigma values are written on the top.



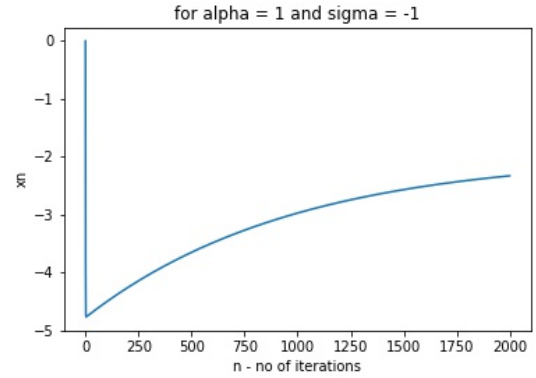


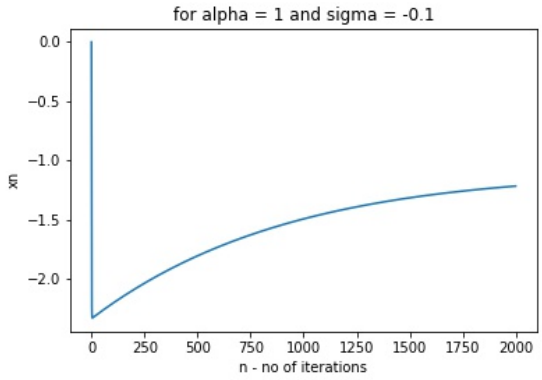


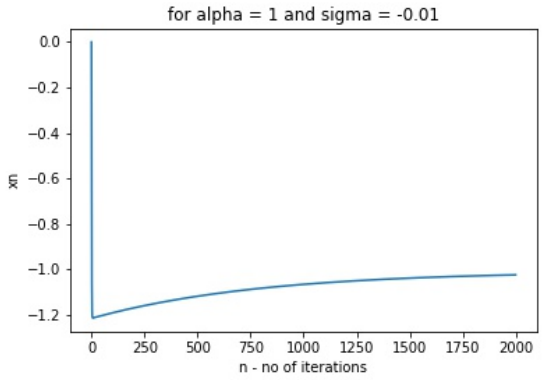
I did a grid search and following results were obtained which are included in the plots below. The finding can be summarised as follows:

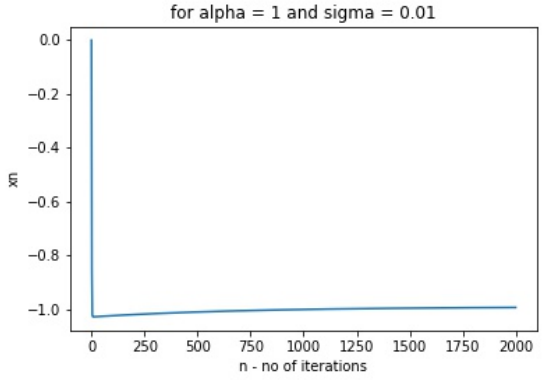
- For $\alpha > 4$ spiking burst were obtained.

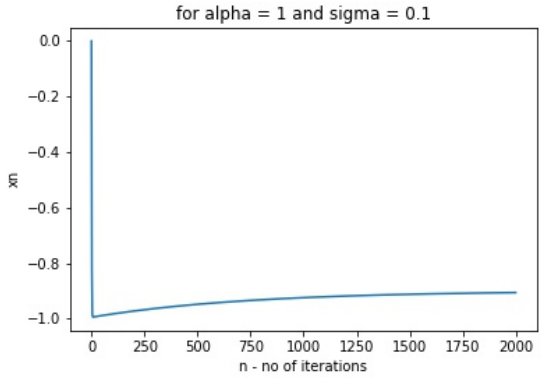
- For $\alpha = 1, 2$ and $\sigma < 1$ regime of silence was obtained.
- For $\alpha = 4$ and $\sigma > 0$ tonic spiking is obtained else region of silence is obtained.

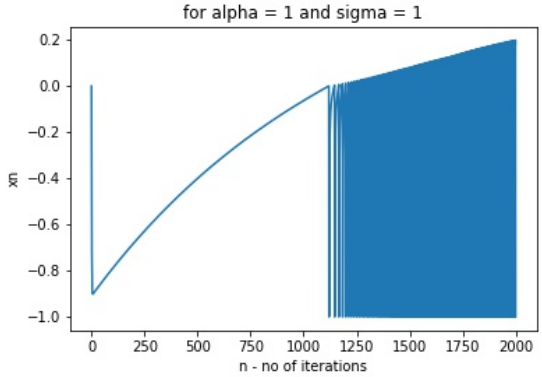


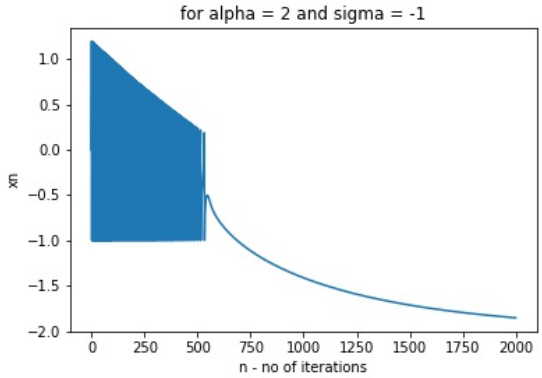


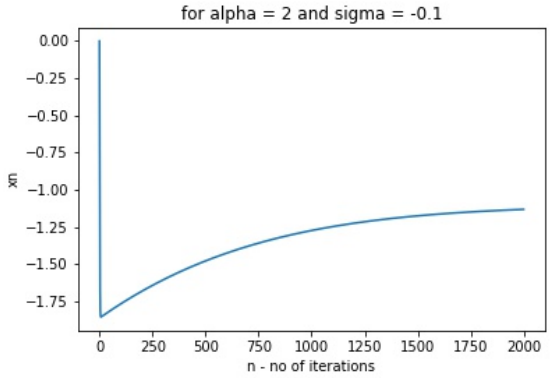


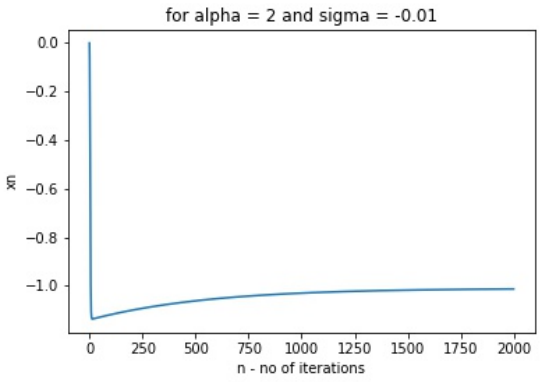


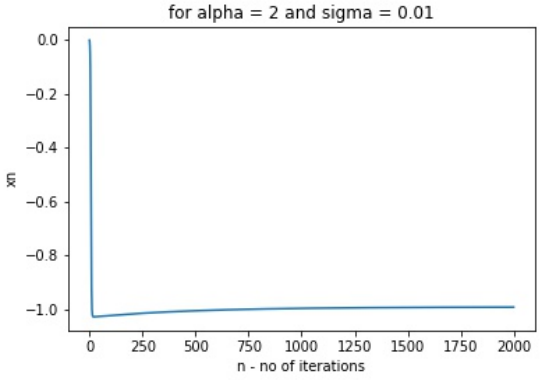

for alpha = 1 and sigma = 0.1
xn
n - no of iterations

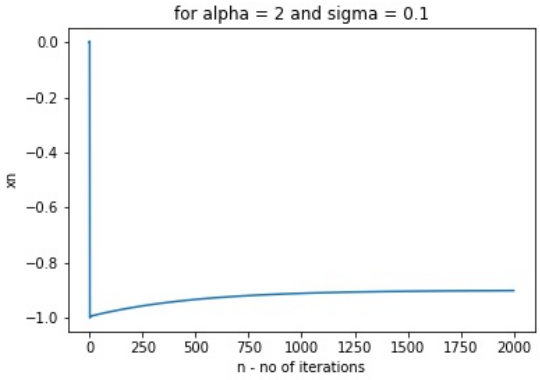

for alpha = 1 and sigma = 1
xn
n - no of iterations

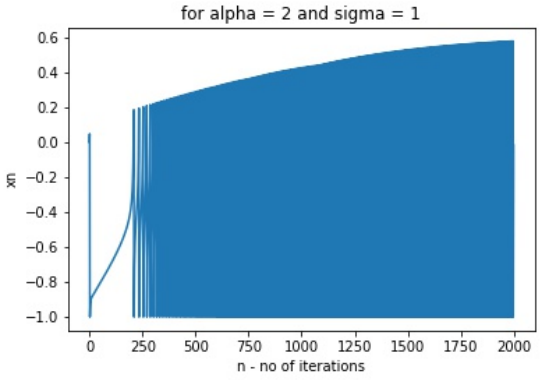

for alpha = 2 and sigma = -1
xn
n - no of iterations

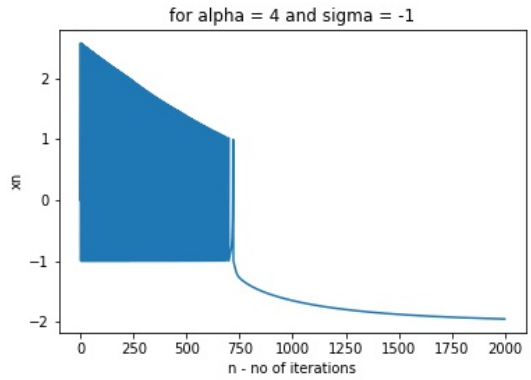

for alpha = 2 and sigma = -0.1
xn
n - no of iterations

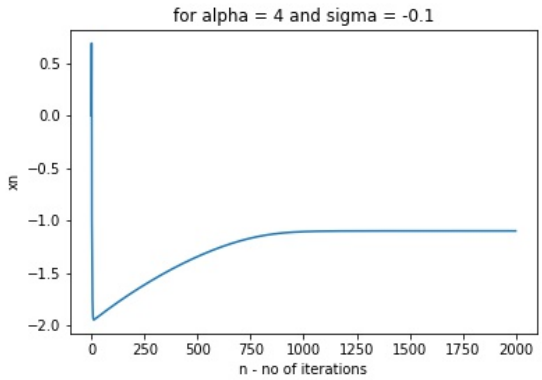

for alpha = 2 and sigma = -0.01
xn
n - no of iterations

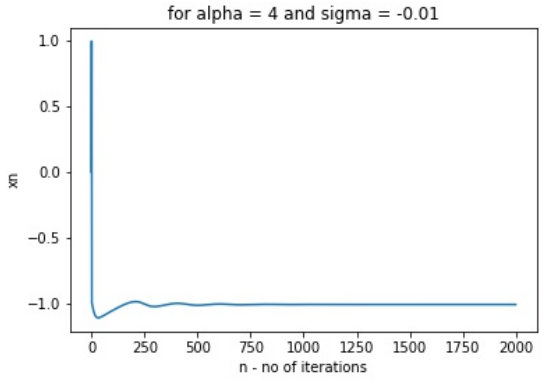

for alpha = 2 and sigma = 0.01
xn
n - no of iterations

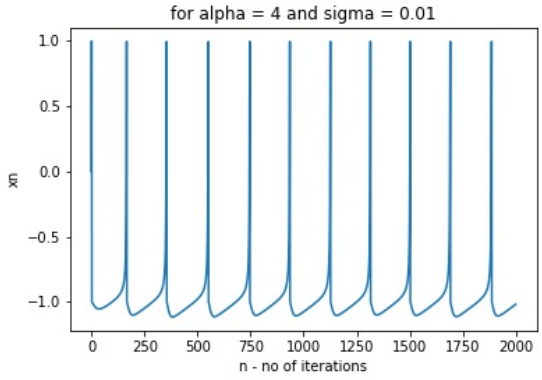

for alpha = 2 and sigma = 0.1
xn
n - no of iterations

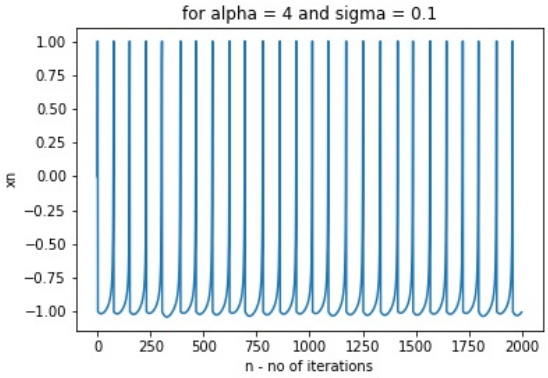

for alpha = 2 and sigma = 1
xn
n - no of iterations

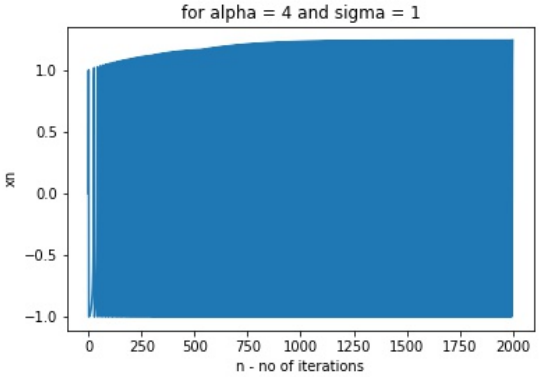

for alpha = 4 and sigma = -1
xn
n - no of iterations

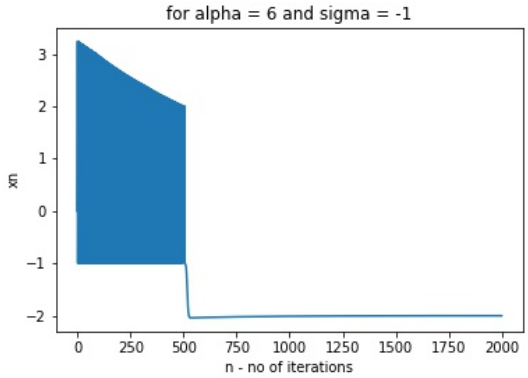

for alpha = 4 and sigma = -0.1
xn
n - no of iterations

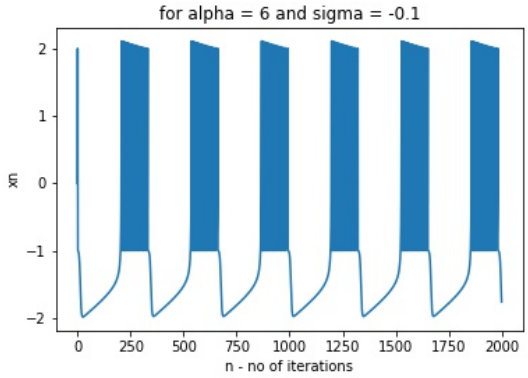

for alpha = 4 and sigma = -0.01
xn
n - no of iterations

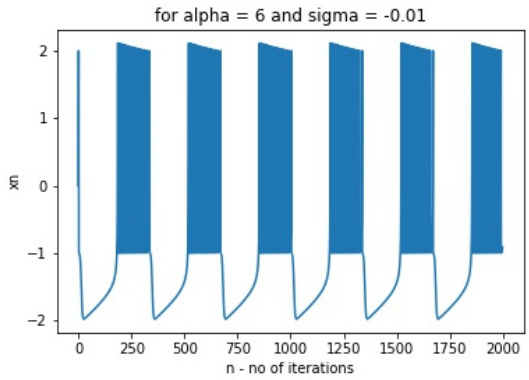

for alpha = 4 and sigma = 0.01
xn
n - no of iterations

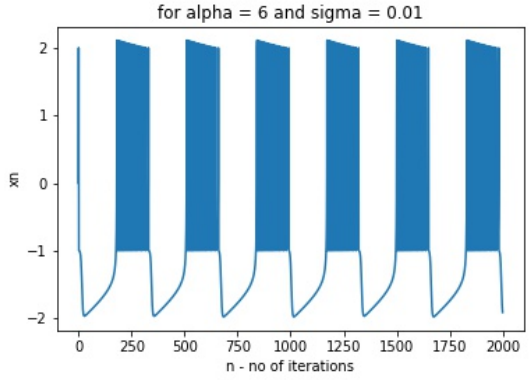

for alpha = 4 and sigma = 0.1
xn
n - no of iterations

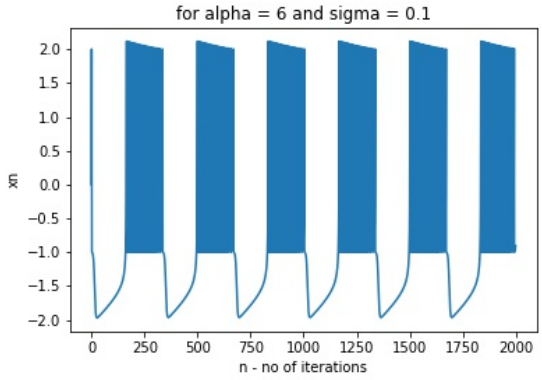

for alpha = 4 and sigma = 1
xn
n - no of iterations

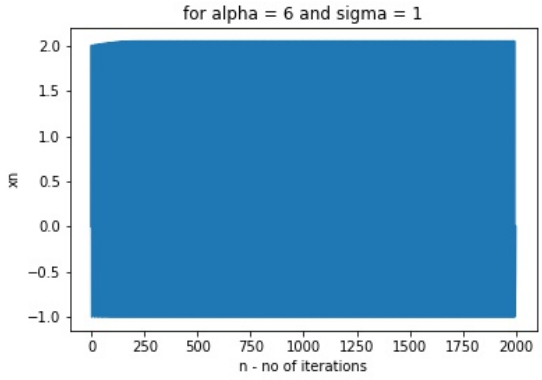

for alpha = 6 and sigma = -1
xn
n - no of iterations

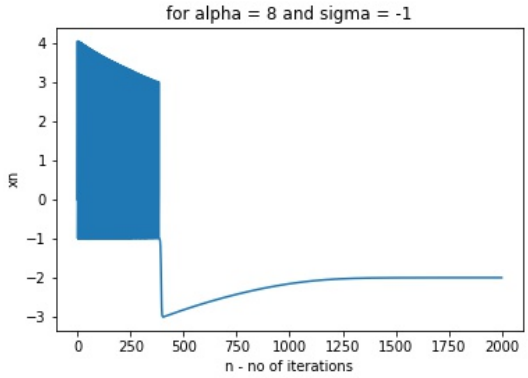

for alpha = 6 and sigma = -0.1
xn
n - no of iterations

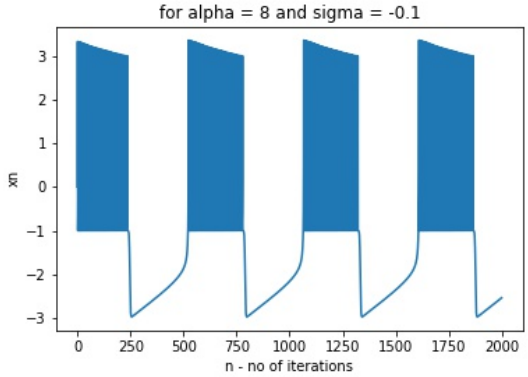

for alpha = 6 and sigma = -0.01
xn
n - no of iterations

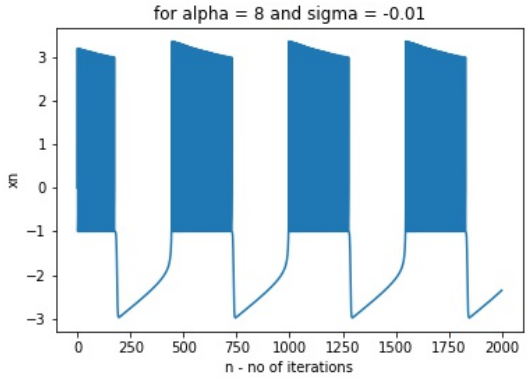

for alpha = 6 and sigma = 0.01
xn
n - no of iterations

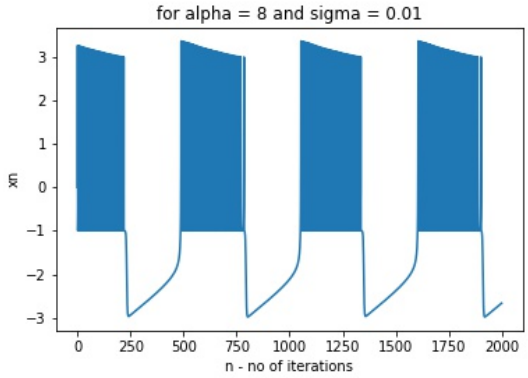

for alpha = 6 and sigma = 0.1
xn
n - no of iterations

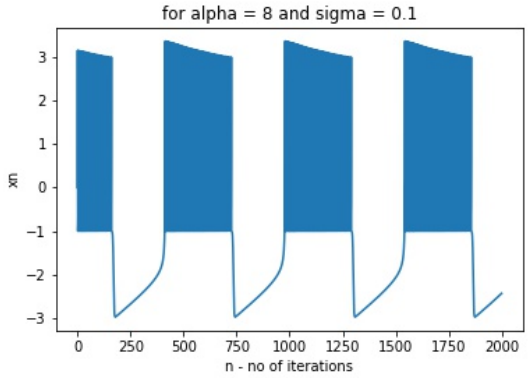

for alpha = 6 and sigma = 1
xn
n - no of iterations

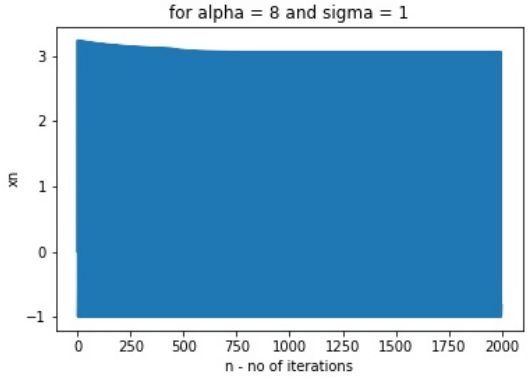

for alpha = 8 and sigma = -1
xn
n - no of iterations


for alpha = 8 and sigma = -0.1
xn
n - no of iterations


for alpha = 8 and sigma = -0.01
xn
n - no of iterations


for alpha = 8 and sigma = 0.01
xn
n - no of iterations


for alpha = 8 and sigma = 0.1
xn
n - no of iterations


for alpha = 8 and sigma = 1
xn
n - no of iterations