University of Waterloo
Faculty of Engineering
Department of System Design Engineering

SYDE 556 - Assignment 1
Representation in Population of Neurons

200 University Avenue West
Waterloo, Ontario, Canada
N2L 3G1

Prepared by
Sugandha Sharma
ID: 20401693
userid: s72sharm
4B Electrical Engineering
24 February 2015

Table of Contents

List of Figures

List of Tables

1 Representation of Scalars

1.1 Basic Encoding and Decoding

a) Rectified Linear Neuron Model - Neuron Responses

Neuron responses ai for 16 randomly generated neurons with x values between -1 and 1, sampled with dx=0.05

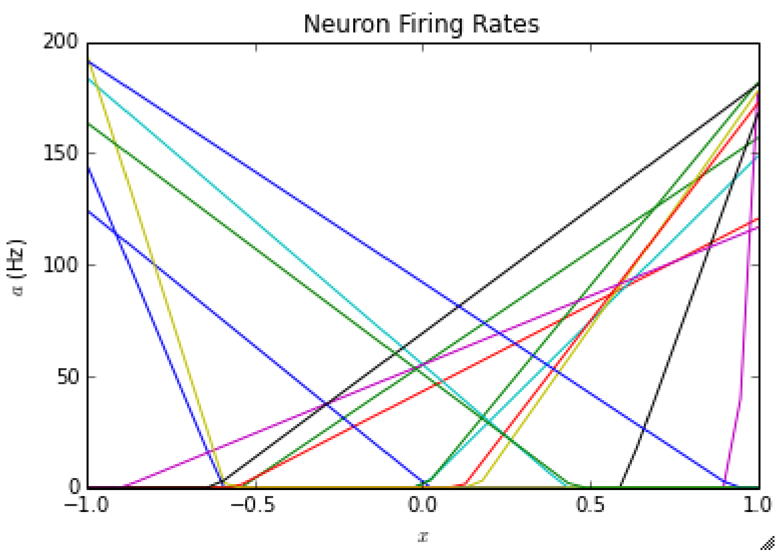

Figure 1. Neuron responses for 16 randomly generated neurons

b) Optimal Decoders

Following are the values of the optimal decoders for the 16 neurons:

```
 Decoder Values are:
[  6.23193964e-07   6.39914951e-06  -7.01409462e-03   6.95258944e-03
  -1.22779885e-04   5.27733996e-04  -3.74027813e-06   1.43664806e-03
  -1.45999572e-04  -1.41150546e-05  -2.59195992e-07  -9.52532620e-04
   5.93023628e-07  -5.45163322e-05   6.52325000e-07   1.10942428e-03]
```

c) Plots of xhat compared to x and RMSE

RMSE (without noise) = 0.000248660984824

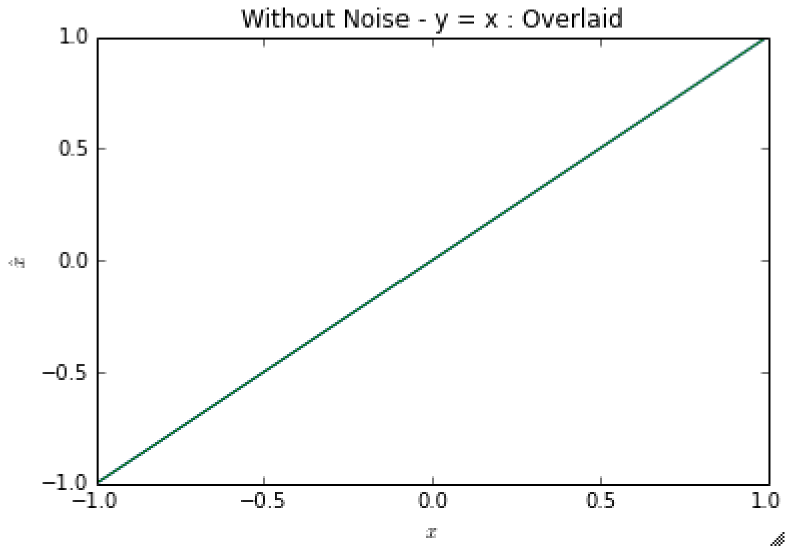

Figure 2. Plot of xhat overlaid on y = x (without noise case)

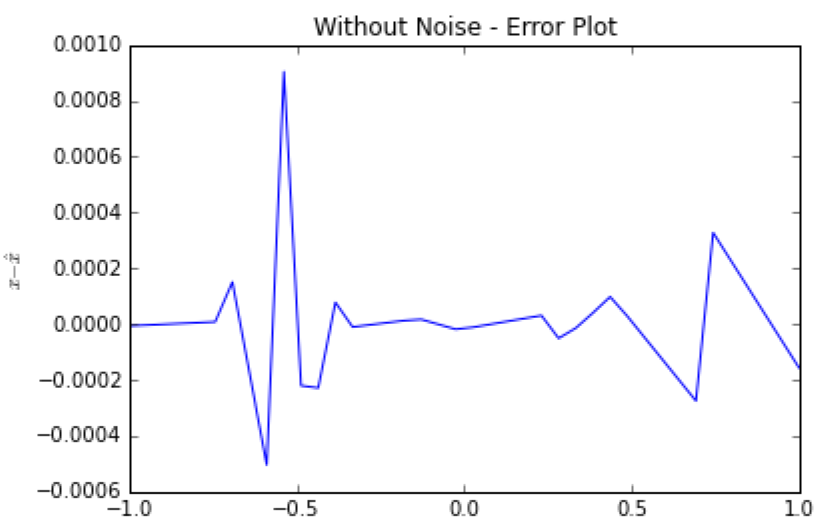

Figure 3. Plot of x - xhat to examine the error (without noise case)

It can be seen that without the noise, the error is almost zero. To confirm this fact, the actual error values were computed and printed as shown below:

```
x - xhat (Without Noise - ):
[ -7.03717437e-06  -4.14545362e-06  -1.25373286e-06   1.63798790e-06
   4.52970866e-06   7.42142941e-06   1.50479460e-04  -1.76732747e-04
  -5.03944955e-04   9.02864082e-04  -2.21375094e-04  -2.29620980e-04
   7.71783071e-05  -1.10167530e-05  -3.38026576e-06   4.25622147e-06
   1.18927087e-05   1.60747420e-05  -1.48438612e-06  -1.90435143e-05
  -1.21849486e-05  -1.59607282e-06   8.99280296e-06   1.95816787e-05
   3.01705545e-05  -5.16843361e-05  -1.49569505e-05   3.95540233e-05
   9.72433198e-05   2.77792667e-05  -4.82543027e-05  -1.24287872e-04
  -2.00321441e-04  -2.76355011e-04   3.27353846e-04   2.30425384e-04
   1.33496923e-04   3.65684607e-05  -6.03600011e-05  -1.57288463e-04]
```

From the above values it can be seen that for all the cases xhat takes a value slightly greater than x

d) Decoding Under Noise added to a (activity)

Added random normally distributed noise to a and decoded again. The noise added is a random variable with mean 0 and standard deviation of 0.2 times the maximum firing rate of all the neurons. This variable was re-sampled for every different x value for every different neuron.

RMSE (noisy activity, non noisy decoder) = 0.932190069651

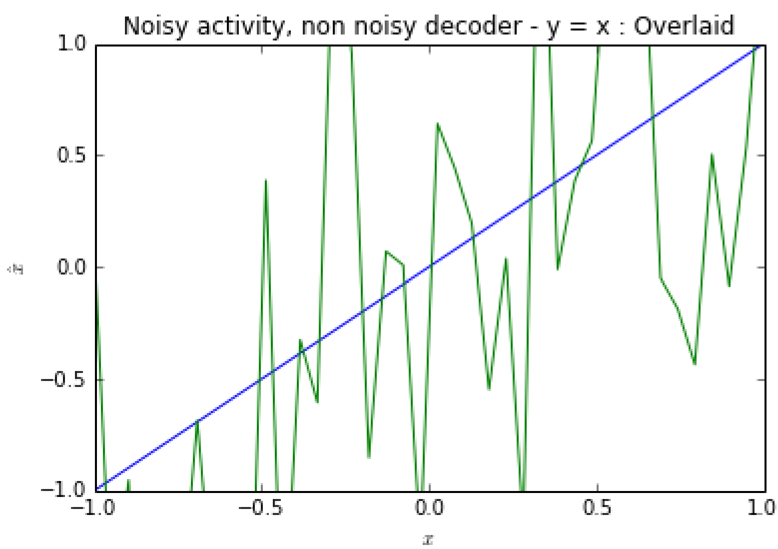

Figure 4. Plot of xhat overlaid on y = x (noisy activity, non noisy decoder case)

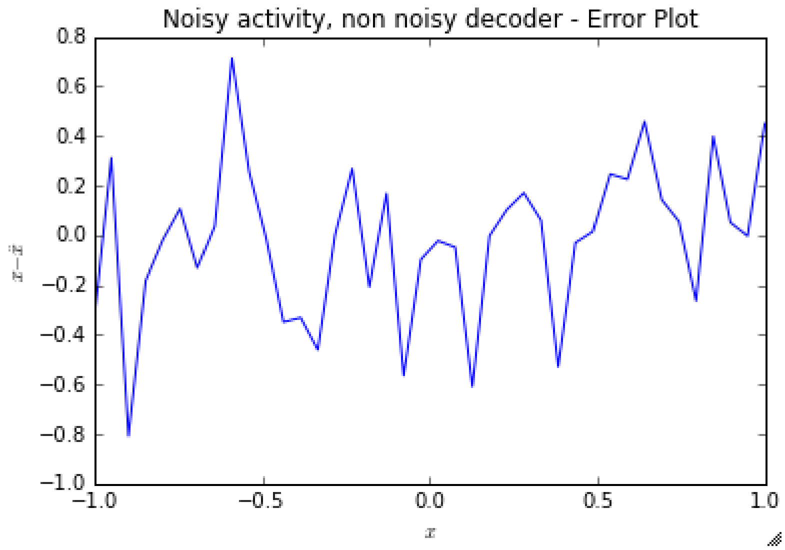

Figure 5. Plot of x - xhat to examine the error (noisy activity, non noisy decoder case)

It can be seen that when the noise is added to the neural activity of each neuron at each x, there is a considerable amount of error which is obtained. This is clear from the graphs and the actual error values which were computed and found to be as shown below:

```
x - xhat (Noisy activity, non noisy decoder - ):
[-1.09944702  0.60084073  0.05522022  2.07681563  1.03346388  0.83988048
 -0.00769217  0.92523569  0.47570507  1.42407955 -0.87468408  1.29868066
 -0.06026436  0.27199061 -1.98173963 -1.18529998  0.67239918 -0.19793593
 -0.08513906  1.19786158 -0.61580523 -0.36677084 -0.06714186  0.72602179
  0.19312576  1.44165203 -1.79253632  0.39506487  0.04949531 -0.07723163
 -1.08251328 -1.12235042 -0.90314099  0.74180341  0.9292253   1.23028097
  0.34156279  0.98248167  0.4105609  -0.48029544]
```

From the above values it can be seen that for some cases xhat takes a value greater than x and for some other cases xhat takes a value lower than x. The non noisy decoder has been very unsuccessful in decoding the values.

e) Re-computing the decoders accounting for noise

In this part the decoders di were re-computed taking noise into account. A comparison was made to compare the behaviour of the when decoding both with and without noise added to a (neural activity). Following are the decoder values computed taking noise into account:

```
Decoder Values with noise are:
[ -4.62461783e-04   1.15256115e-03   1.64100913e-03   4.59629243e-05
  -2.09668455e-04   9.53157746e-04   1.86924815e-04  -1.79546878e-03
   8.60841785e-04   5.89947642e-04  -4.06238789e-04   1.00479179e-04
  -7.45863983e-04   7.15669959e-04  -1.87610739e-03  -8.49209886e-04]
```

RMSE (with noisy activity and noisy decoder) = 0.131212723644

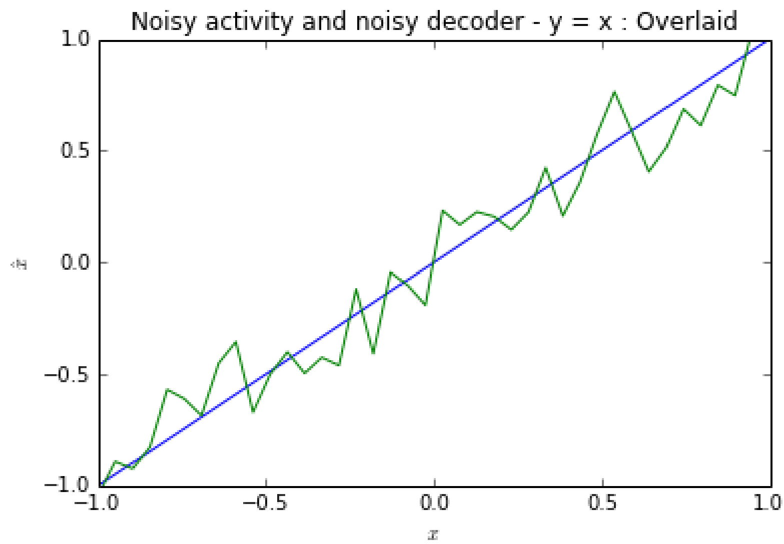

Figure 6. Plot of xhat overlaid on y = x (noisy activity, noisy decoder case)

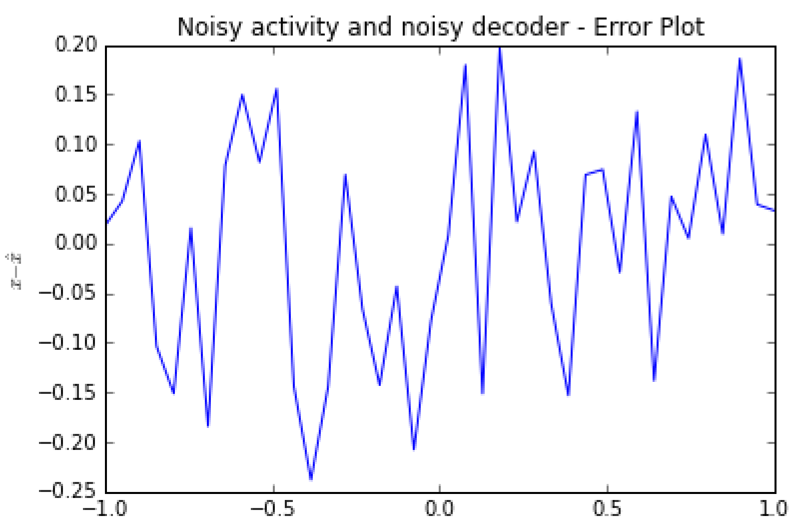

Figure 7. Plot of x - xhat to examine the error (noisy activity, noisy decoder case)

RMSE (with non noisy activity and noisy decoder) = 0.0349894292346

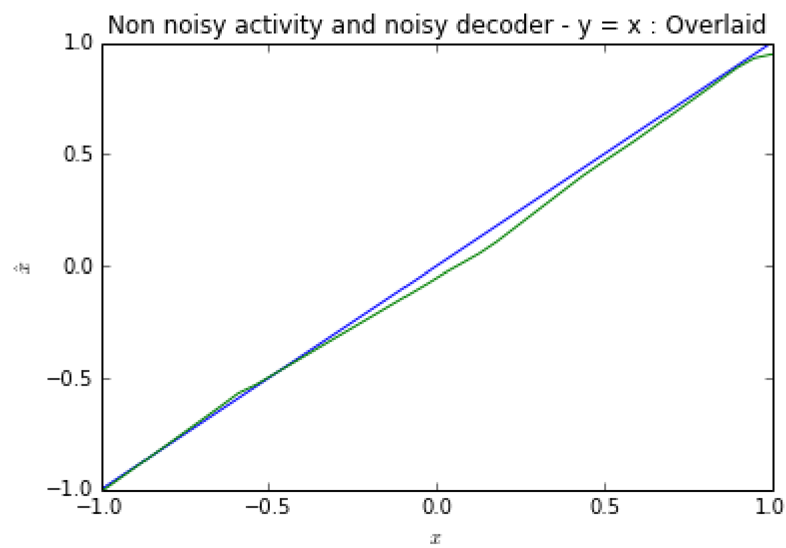

Figure 8. Plot of xhat overlaid on y = x (non noisy activity, noisy decoder case)

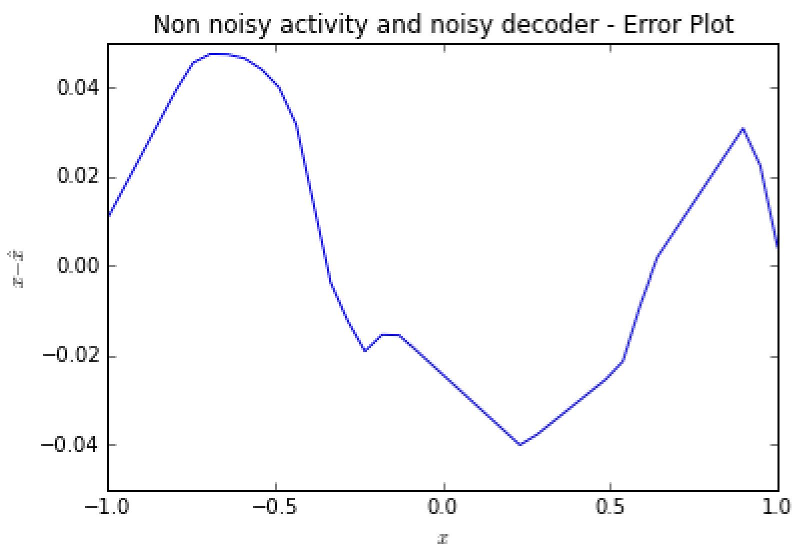

Figure 9. Plot of x - xhat to estimate the error (non noisy activity, noisy decoder case)

f) RMSE Analysis

Terminology:

Noisy Decoder - Decoders Computed taking noise into account
Noisy Activity - Neural Activities and the neural activity matrix is computed taking noise into account.

Table 1. Comparison of RMSE values obtained with and without noise

	RMSE Values	Comments
1. Activity and Decoder Without Noise	1.34481218402e-09	The RMSE was minimum without noise
2. Noisy Activity, non noisy Decoder	0.932190069651	RMSE increased considerably on adding noise to the activity
3. Noisy activity, noisy Decoder,	0.131212723644	RMSE decreased as compared to that in 2 when noisy activity was decoded using noisy decoder
4. Non noisy Activity, noisy Decoder	0.0349894292346	RMSE increased when a noisy decoder was used as compared to a non noisy decoder (in 1). However, it was less compared to case 3.

Table 2. 2x2 table of the RMSE Values

1.34481218402e-09 (non noisy activity, non noisy decoder)	0.932190069651 (noisy activity, non noisy decoder)
0.131212723644 (noisy activity, noisy decoder)	0.0349894292346 (non noisy activity, noisy decoder)

It can be seen from the Table1 above that the RMSE value was the lowest (almost negligible) when no noise had been added to the neural activities of the neurons. However, on adding the noise to the neural activities, the RMSE value increased considerably. In order to decode noisy neural activities, the results were better when the decoder which was computed taking the noise into account was used. This is clear from Table1 above, which shows that the RMSE value in case 3 was found to be lower than in case 2 since a decoder optimized for the noisy neural activities was used. This shows that the decoders are optimized for certain specific scenarios and perform best for the scenarios they have been optimized for. This is confirmed from the observation that in case 4, where the noisy decoder was used to decode the neural activities without noise, the results were considerably worse than case 1 where non noisy decoder optimized to decode neural activities without noise was used. The RMSE in case 4 was found to be a lot higher than in case 1. Further, it can be seen from the values of decoders reported in previous sections that the values of noisy and non-noisy decoders were considerably different. A noisy decoder also does a better job at decoding non noisy activity as compared to noisy activity (though a non noisy decoder is much better for decoding non noisy activity).

1.2 Exploring Sources of Error

a) Plots of Error due to distortion and noise (Standard Deviation = 0.1)

In this part two different loglog plots were generated (one for each type of error) with N (number of neurons) values of [2, 4, 8, 16, 32, 64, 128]. For each N value, 10 runs were done in order to obtain the average results. For each run, different α, Jbias, and e were generated for each neuron. Decoders were computed under noise, with σ equal to 0.1 times the maximum firing rate.

The average distortion error and noise error computed for the N values of [2, 4, 8, 16, 32, 64, 128] and σ equal to 0.1 times the maximum firing rate are shown below:

```
 Distortion Error
[2.1009561200848843, 0.06623938196548497, 0.025846729017044982, 0.035263615123308903, 0.010475967572383773, 0.0030957174818984231]

 Noise Error
[0.022614345759516534, 0.011283376759227286, 0.0081889247287237361, 0.010651616777210495, 0.003701393085861491, 0.00090140973946104311
]
```

The error due to noise was found to be considerably higher than the error due to distortion. Both errors decrease as the number of neurons is increased from 2 to 128. However, the error due to distortion decreases at a much faster rate as compared to the error due to noise. These errors were compared to 1/N , $1/N^2$ and $1/N^4$ and the results can be seen in figure10 and figure 11.

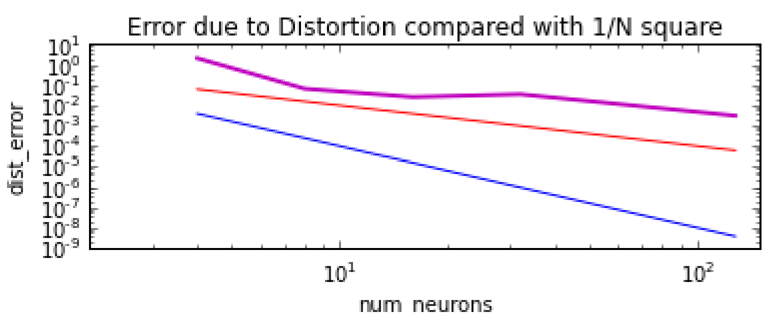

Figure 10. Error due to distortion (majenta),I/N^2(red) and 1/N^4(blue) for std_deviation = 0.1

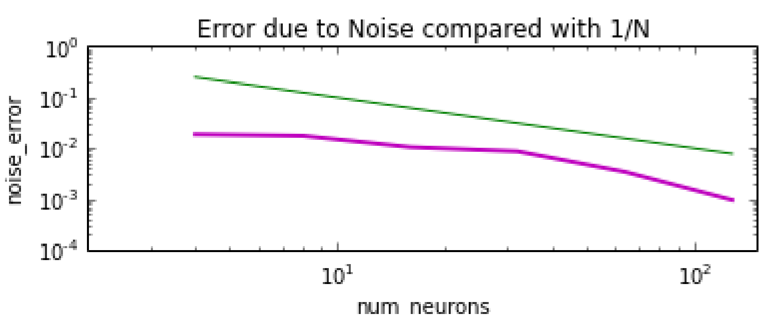

Figure 11. Error due to noise (majenta) and I/N^2 (red) for std_deviation = 0.1

Figure11 shows that the error due to noise is almost proportional to 1/N and figure10 shows that the error due to distortion is proportional to $1/N^2$.

b) Plots of Error due to distortion and noise (Standard Deviation = 0.01)

The average distortion error and noise error computed for the N values of [2, 4, 8, 16, 32, 64, 128] and σ equal to 0.01 times the maximum firing rate are shown below:

```
 Distortion Error
[0.47067084874418813, 0.034719055867694734, 0.001244131701085602, 0.0016435397852239356, 0.00026002559463317017, 9.0290157982444202e-0
5]

 Noise Error
[0.00032276579018625473, 0.0003082737627893591, 0.00012918679693253905, 0.00020426788173382487, 3.1601360652853703e-05, 9.057270350861
6281e-06]
```

Graphs of error due to distortion and error due to noise

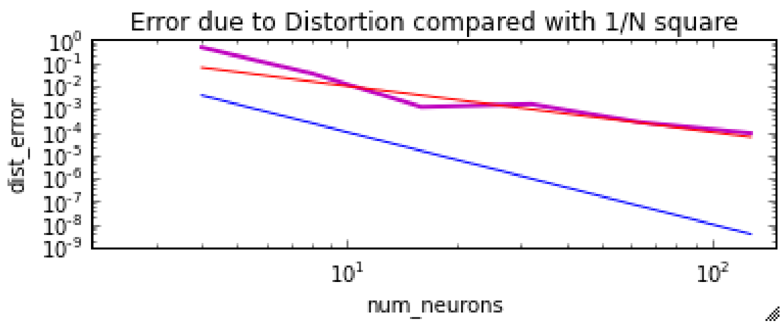

Figure 12. Error due to distortion (majenta), I/N^2 (red) and $1/N^4$ for std_deviation = 0.01

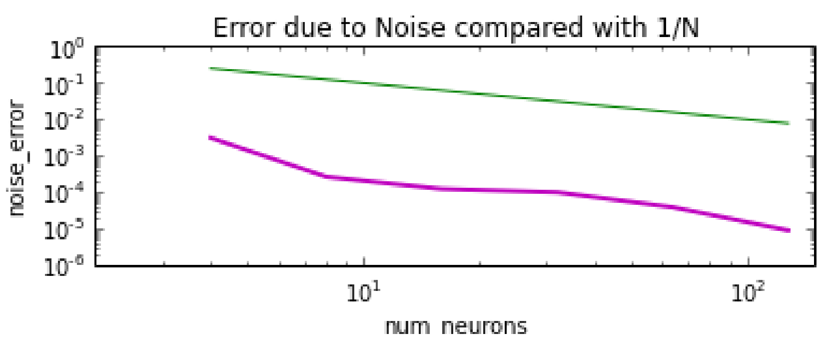

Figure 13. Error due to noise (majenta) and I/N (red) for std_deviation = 0.01

Figure13 shows that the error due to noise is almost proportional to 1/N and figure12 shows that the error due to distortion is proportional to $1/N^2$.

c) Difference between graphs in a and b

From the graphs and data presented in part a) and b) above, it can seen that on decreasing the standard deviation, the average error due to distortion remains almost the same (with very little difference). However, the error due to noise is found to decrease with a decrease in the standard deviation. This is as expected since the error due to distortion is the error introduced by the decoders themselves and this is present regardless of noise. However the error due to noise is directly dependant on the standard deviation of the distribution from which the noise is sampled which affects the neural activity of each neuron.

As shown in the figure below, lower standard deviation means there is a narrower sample space from which the noise is sampled. Hence there is a less variation in the noise components added to various neurons implying more similarity of noise componenets among the population of neurons. Thus the error due to noise decreases on decreasing the standard deviation.

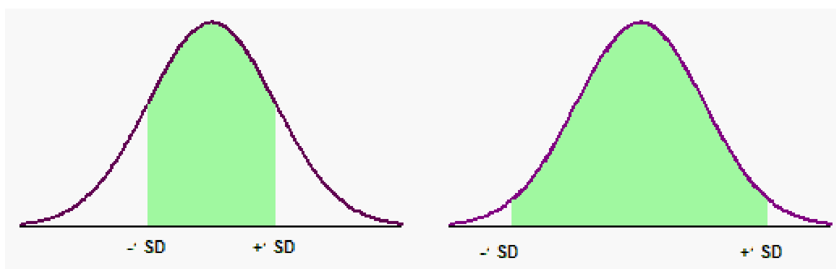

Figure 14. Standard Deviation (left - lower deviation and right - higher deviation)

1.3 Leaky Integrate and Fire Neurons

a) Neuron Reponses

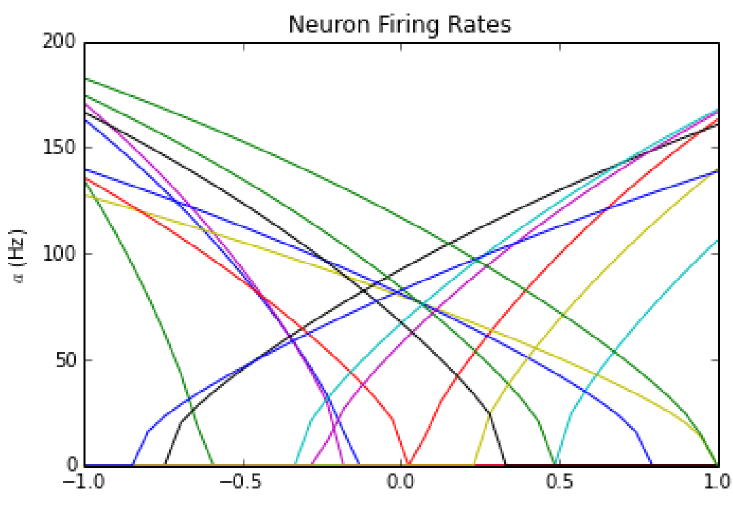

Figure 15. Neuron Responses for 16 randomly generated neurons

RMSE (without noise) = 0.00812292249089 (Using Optimal Decoders without noise)

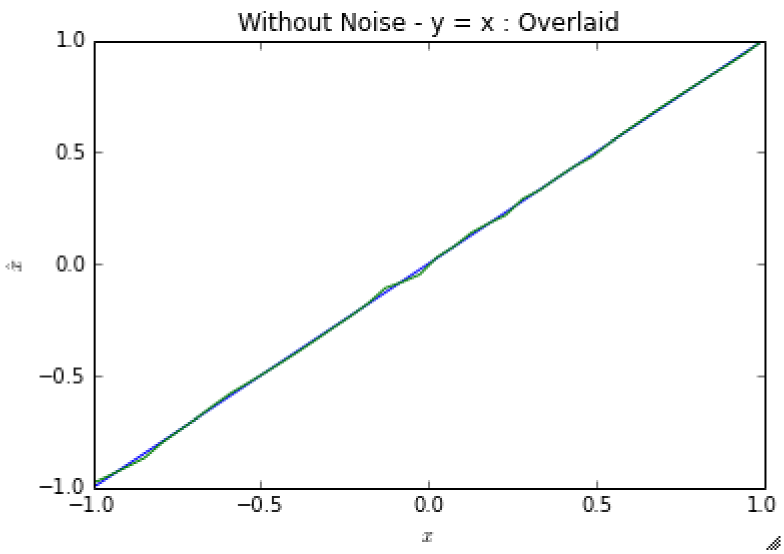

Figure 16. Plot of xhat overlaid on y = x (non noisy activity, non noisy decoder case)

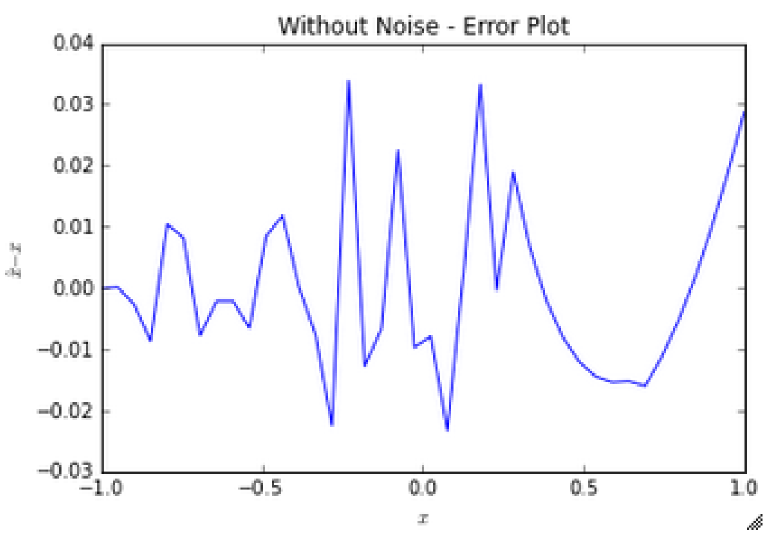

Figure 17. Plot of x - xhat to estimate the error (non noisy activity, non noisy decoder case)

b) Re-computing decoders accounting for noise

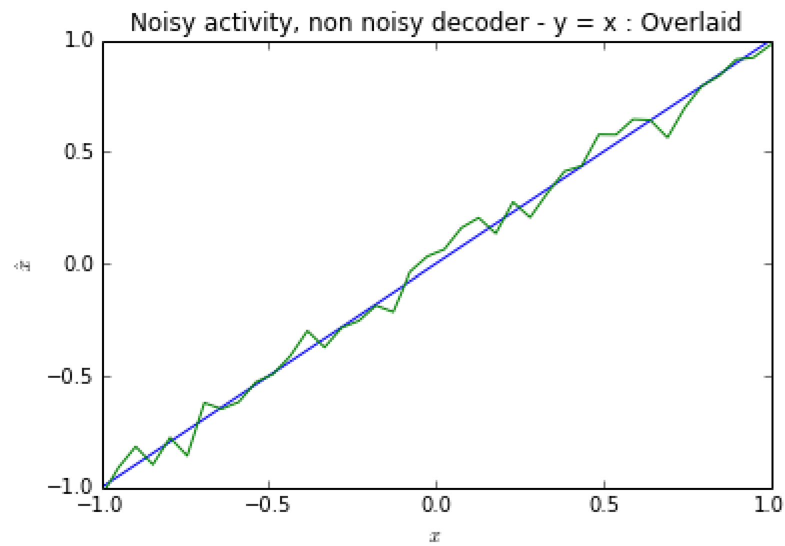

Figure 18. Plot of xhat overlaid on y = x (noisy activity, non noisy decoder case)

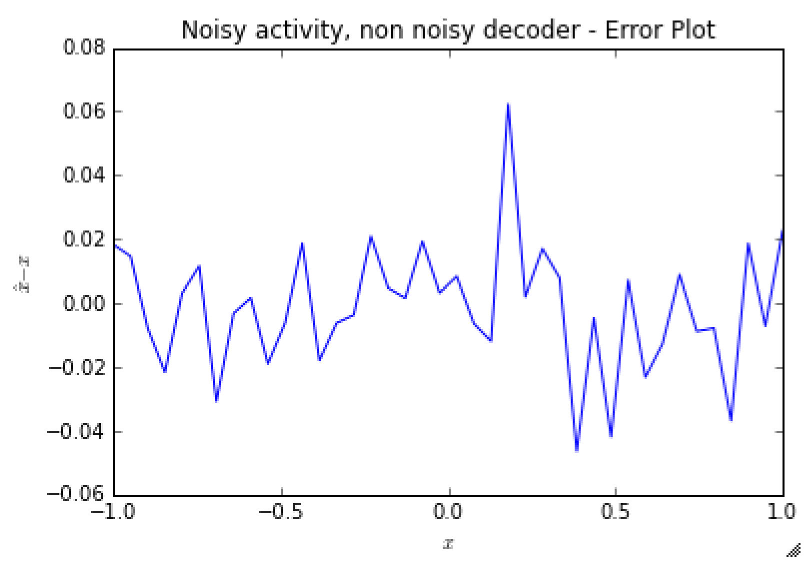

Figure 19. Plot of x - xhat to estimate the error (noisy activity, non noisy decoder case)

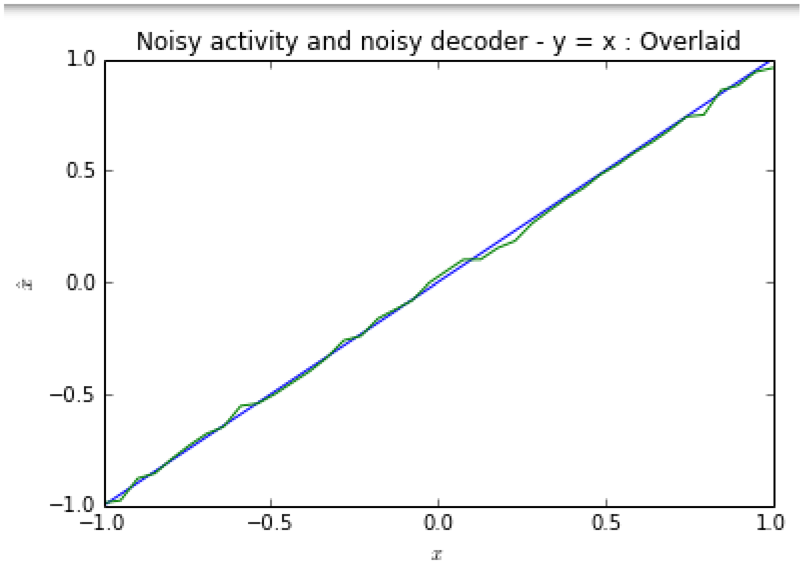

Figure 20. Plot of xhat overlaid on y = x (noisy activity, noisy decoder case)

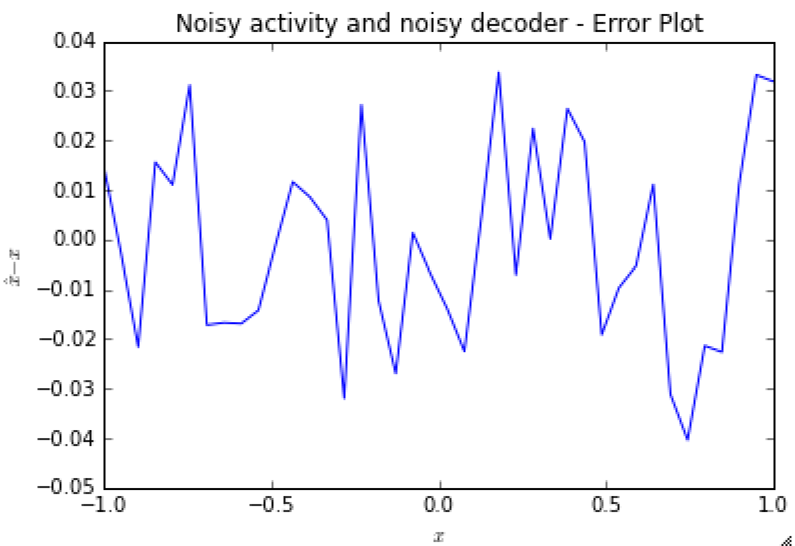

Figure 21. Plot of x - xhat to estimate the error (noisy activity, noisy decoder case)

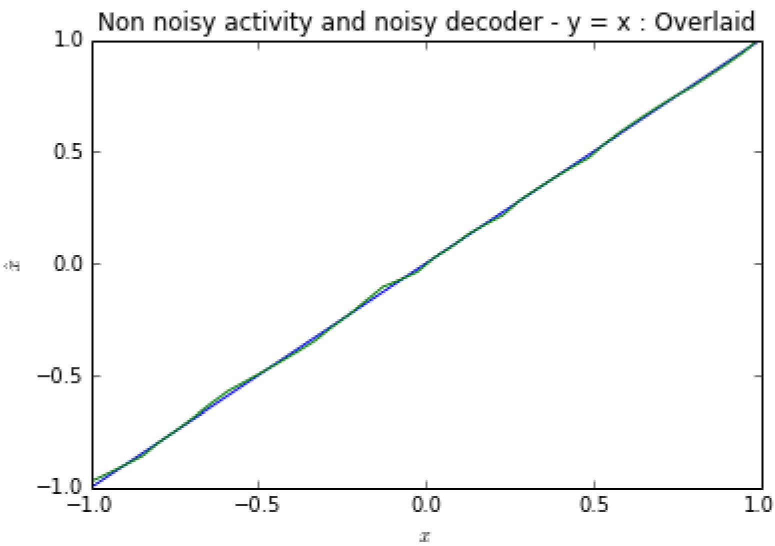

Figure 22. Plot of xhat overlaid on y = x (non noisy activity, noisy decoder case)

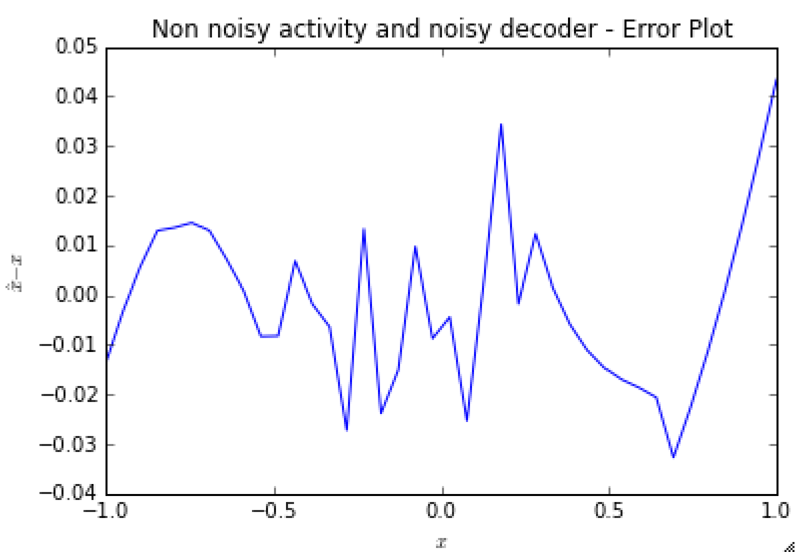

Figure 23. Plot of x - xhat to estimate the error (noisy activity, noisy decoder case)

RMSE Values (The trend in the values was similar to that found in the case of rectified linear neuron:

	RMSE Values	Comments
1. Activity and Decoder Without Noise	0.00480676926067	The RMSE was minimum without noise
2. Noisy Activity, non noisy Decoder	0.0136443776118	RMSE increased on adding noise to the activity
3. Noisy activity, noisy Decoder,	0.0106384504765	RMSE decreased as compared to that in 2 when noisy activity was decoded using noisy decoder
4. Non noisy Activity, noisy Decoder	0.00741586188502	RMSE increased when a noisy decoder was used as compared to a non noisy decoder (in 1).

2 Representation of Vectors

2.1 Vector Tuning Curves

a) LIF neuron with 2D preferred direction

In this part, the tuning curve of an LIF neuron with 2D preferred direction vector is at an angle of $\theta=-\pi/4$ was plotted. The neuron has an x-intercept at the origin (0,0), and has a maximum firing rate of 100Hz.

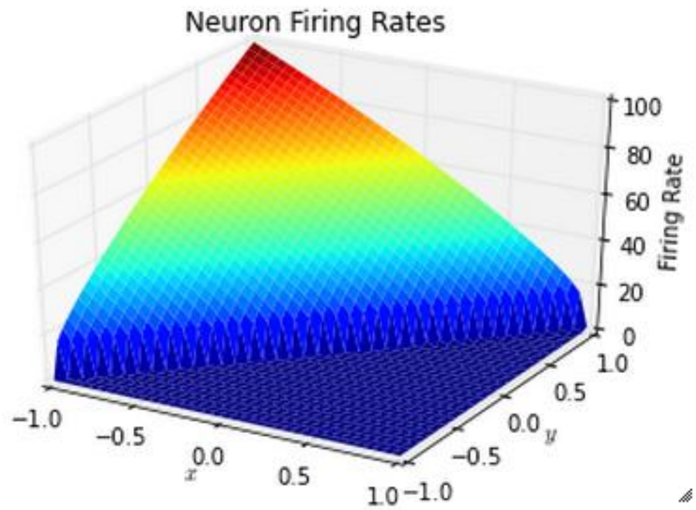

Figure 24. Tuning Curve of an LIF neuron with a 2D preferred direction vector at −π/4

b) Tuning curve considering the points around a unit circle

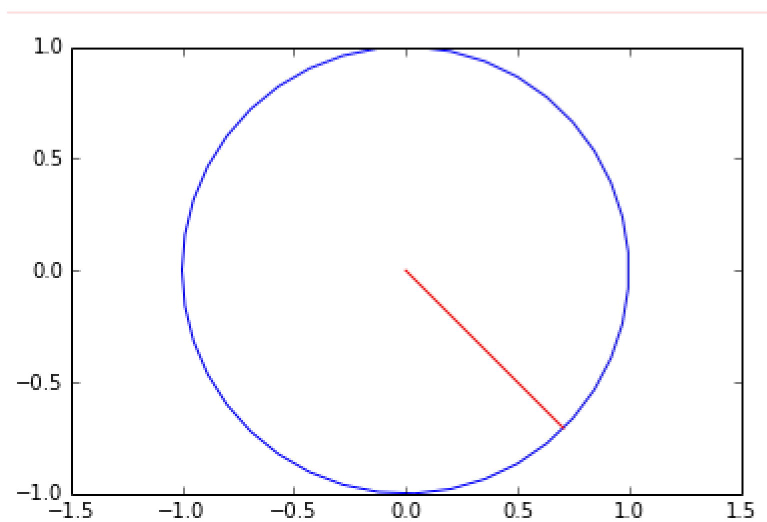

Figure 25. Points around a unit circle

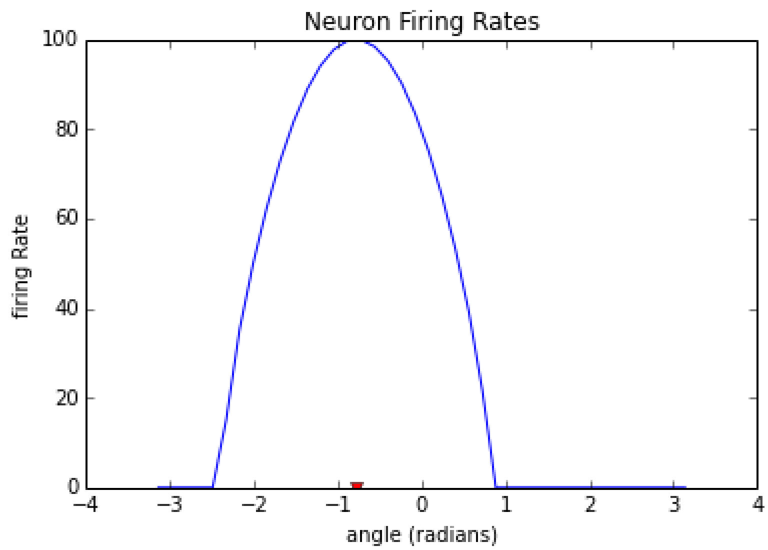

Figure 26.Tuning Curve of a neuron considering the points around a unit circle

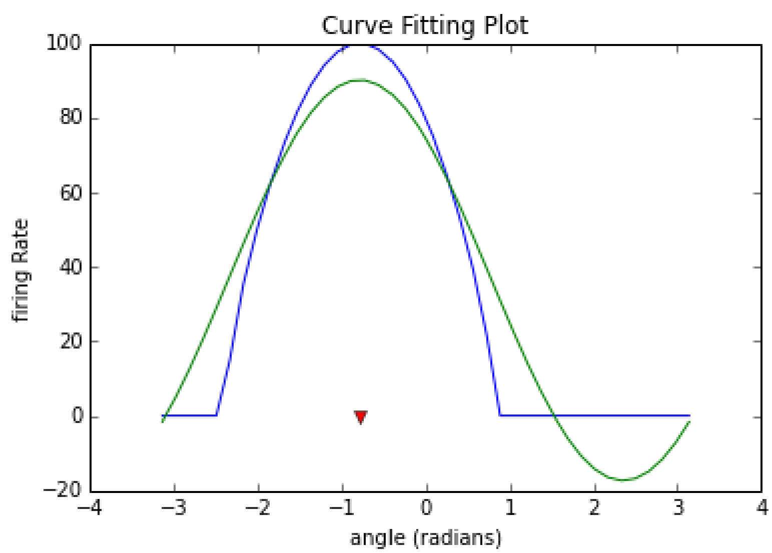

Figure 27. Cosine Curve Fitted to the tuning curve

It was expected that the maximum firing rate should occur at $-\pi/4$ (-0.785) since that is the direction of the preferred direction vector of the LIF neuron. This matches exactly with the result obtained as shown in Figure 26.

<u>What Makes Cosine a good choice?</u>

For curve fitting, we need to construct a curve that is a best fit to the data points (firing rates) obtained. Since the firing rates found in response to the points around the unit circle show a similar trend to a cosine wave, so cosine seems to be a good choice for curve fitting since it aids in the data visualization. Furthermore, the maximum of the cosine curve occurs at the same point as the maximum of the firing rate curve suggesting that it can provide a good approximation.

Why does it differ from the ideal curve?

The cosine curve differs from the ideal curve since it takes negative values and also doesn't take the exact y-values. The firing rate is not an exact cosine function since it is computed using the LIF model which takes the current (shown in the equation below) as an input. So even though the x values are around a unit circle, the LIF model itself introduces variations.

$J = \alpha e \cdot x + J_{bias}$

2.2 Vector Representation

a) 100 unit vectors around a unit circle

A set of 100 random unit vectors uniformly distributed around the unit circle was generated. These were later used as the encoders e for 100 neurons.

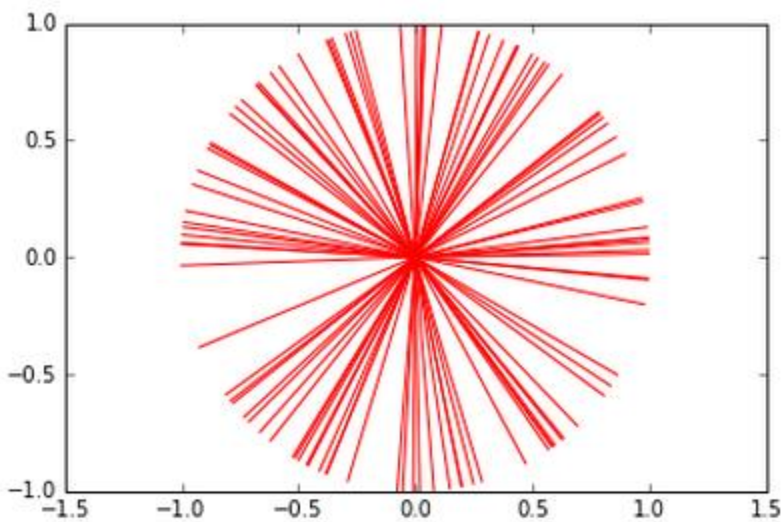

Figure 28. 100 Unit vectors around a circle (Encoders)

b) Computing Optimal Decoders

When computing the decoders noise with σ as 0.2 times the maximum firing rate was taken into account.

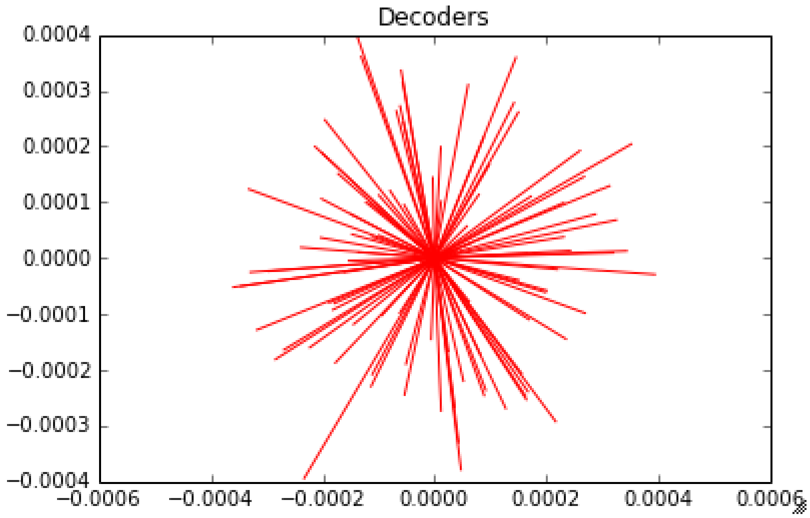

Figure 29. Decoders

<u>How do these decoding vectors compare to the encoding vectors?</u>

The magnitude of the decoding vectors is less than that of the encoding vectors. The encoding vectors were uniformly distributed around a unit circle which means that each of the 100 neurons has their own preferred direction. Each neuron will fire the most when the stimulus is in it's referred direction. Since the stimulus in this case is 1600 points (x,y) therefore, different neurons have different firing rates in response to all these points. A neuron will respond the more strongly to the points which lie in its preferred direction. Therefore, in order to reflect this fact, the decoders have different magnitudes in different directions which is as expected.

c) RMSE using 20 random x values around a unit circle

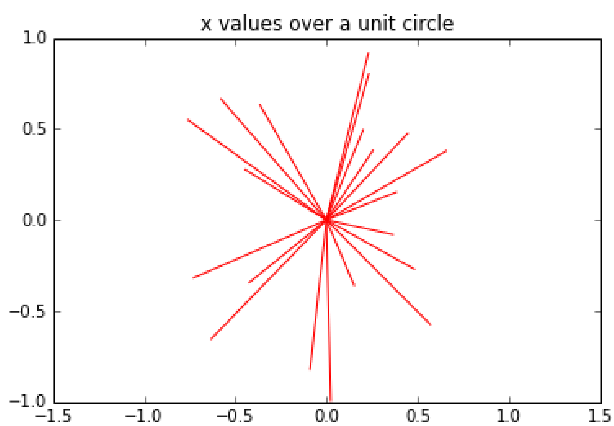

Figure 30. 20 random x values around a unit circle

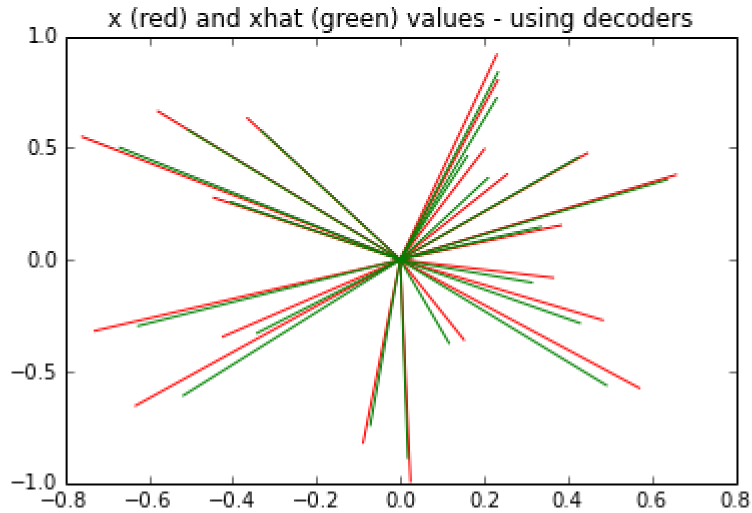

Figure 31.Plot of x and xhat where xhat is decoded using the decoders

RMSE (using decoders) = 0.053738903153

d) Using Encoders as Decoders

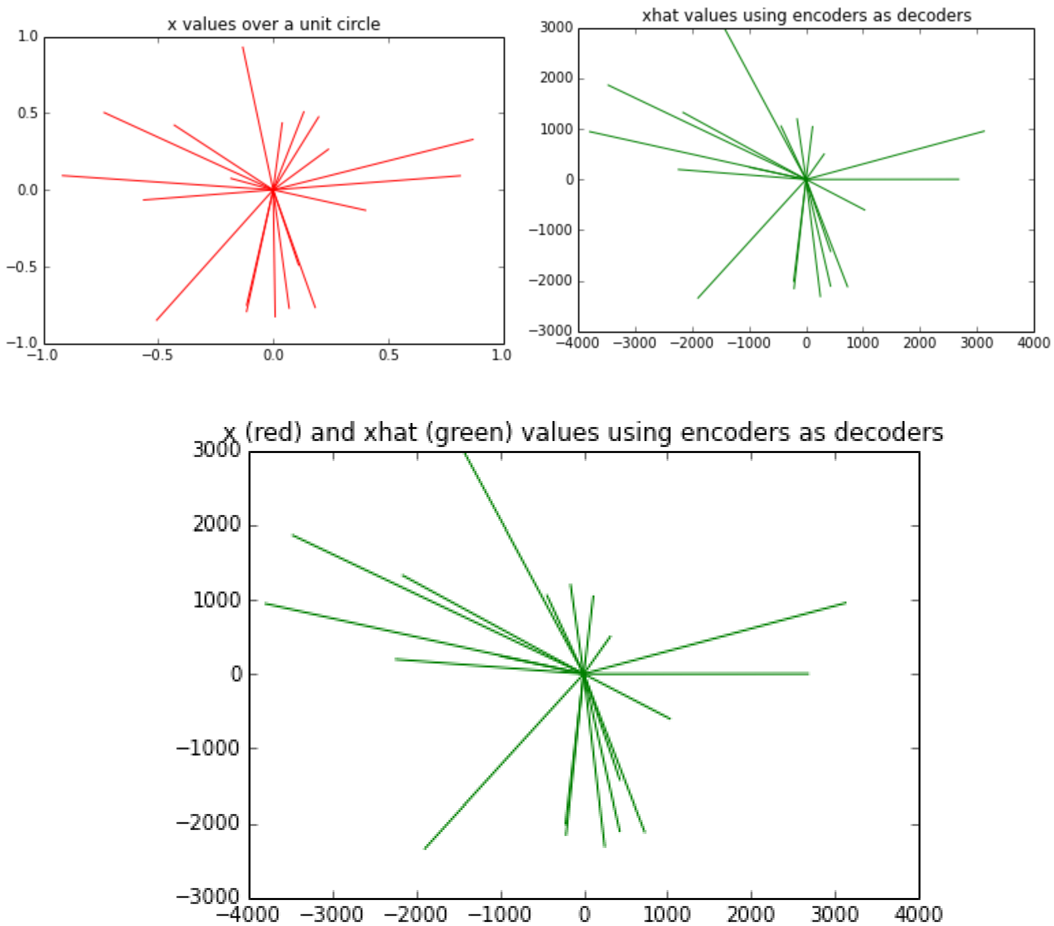

Figure 32. Plot of x and xhat where xhat is decoded using the encoder

The magnitude of xhat values is so large that the x values are not clearly visible in the third plot of Figure32. However, the first two plots show the x and xhat values plotted separately. The encoder magnitudes are very big to act as decoders and hence the RMSE value found is very large.

RMSE (using encoders as decoders) = 1733.6655522
RMSE (using decoders ignoring magnitudes) = 0.22763422118
RMSE (using encoders as decoders , ignoring magnitudes) = 0.215307251085

<u>What are the relative merits of these two approaches to decoding?</u>

Decoding using an encoder gives very bad results when the magnitude is not ignored. This is because the magnitude of the encoder is too big to act as a decoder. However, when the magnitude is ignored the encoders do a fairly good job at decoding. The results obtained on ignoring the magnitude are very similar in case of decoding using an encoder as well as decoder. This implies that encoders are good at decoding the direction but give bad results when magnitude is also considered. Thus the benefit of using encoders is that its efficient to decode the direction. On the other hand benefit of using an encoder is that it is able to decode both the magnitude and direction. Therefore, if we are only interested in finding the direction of stimulus then encoders can do a good job.