>>> Implementation Hand Written Digit Recognition with Neural Network
>>> in MATLAB

Name: Ali Nehrani[†]
Date: December 27, 2017

[†]ali.nehrani@ozu.edu.tr

>>> This presentation contains:

1. Intro
2. Creating Data
3. Neural Network
4. Learning
5. Results

>>> Hand written digits

Hand written (HR) digits; are generally the digits that is written by hand.

>>> Hand written digits

Hand written (HR) digits; are generally the digits that is written by hand.
In this project the HR digits are changed into vaectors with certain structure.

>>> Hand written digits

Hand written (HR) digits; are generally the digits that is written by hand.
In this project the HR digits are changed into vaectors with certain structure.
This process is doen by getUserTraj.m program.

>>> Hand written digits

Hand written (HR) digits; are generally the digits that is written by hand.
In this project the HR digits are changed into vaectors with certain structure.
This process is doen by getUserTraj.m program.
getUserTraj.m takes hand writing (and rate dt) as an input and returns the following feature vectors:

>>> Hand written digits

Hand written (HR) digits; are generally the digits that is written by hand.
In this project the HR digits are changed into vaectors with certain structure.
This process is doen by getUserTraj.m program.
getUserTraj.m takes hand writing (and rate dt) as an input and returns the following feature vectors:

1. $[x,\ y]$: coordinate of points in the (HR) image

>>> Hand written digits

Hand written (HR) digits; are generally the digits that is written by hand.
In this project the HR digits are changed into vaectors with certain structure.
This process is doen by getUserTraj.m program.
getUserTraj.m takes hand writing (and rate dt) as an input and returns the following feature vectors:

1. $[x,\ y]$: coordinate of points in the (HR) image
2. $[vx,\ vy]$: velocity in x and y directions

>>> Hand written digits

Hand written (HR) digits; are generally the digits that is written by hand.
In this project the HR digits are changed into vaectors with certain structure.
This process is doen by getUserTraj.m program.
getUserTraj.m takes hand writing (and rate dt) as an input and returns the following feature vectors:

1. $[x,\ y]$: coordinate of points in the (HR) image
2. $[vx,\ vy]$: velocity in x and y directions
3. $[ax,\ ay]$: accelarations in x and y directions

>>> Hand written digits

Hand written (HR) digits; are generally the digits that is written by hand.
In this project the HR digits are changed into vaectors with certain structure.
This process is doen by getUserTraj.m program.
getUserTraj.m takes hand writing (and rate dt) as an input and returns the following feature vectors:

1. $[x,\ y]$: coordinate of points in the (HR) image
2. $[vx,\ vy]$: velocity in x and y directions
3. $[ax,\ ay]$: accelarations in x and y directions
4. nt: the time line

>>> getUserTraj.m interface

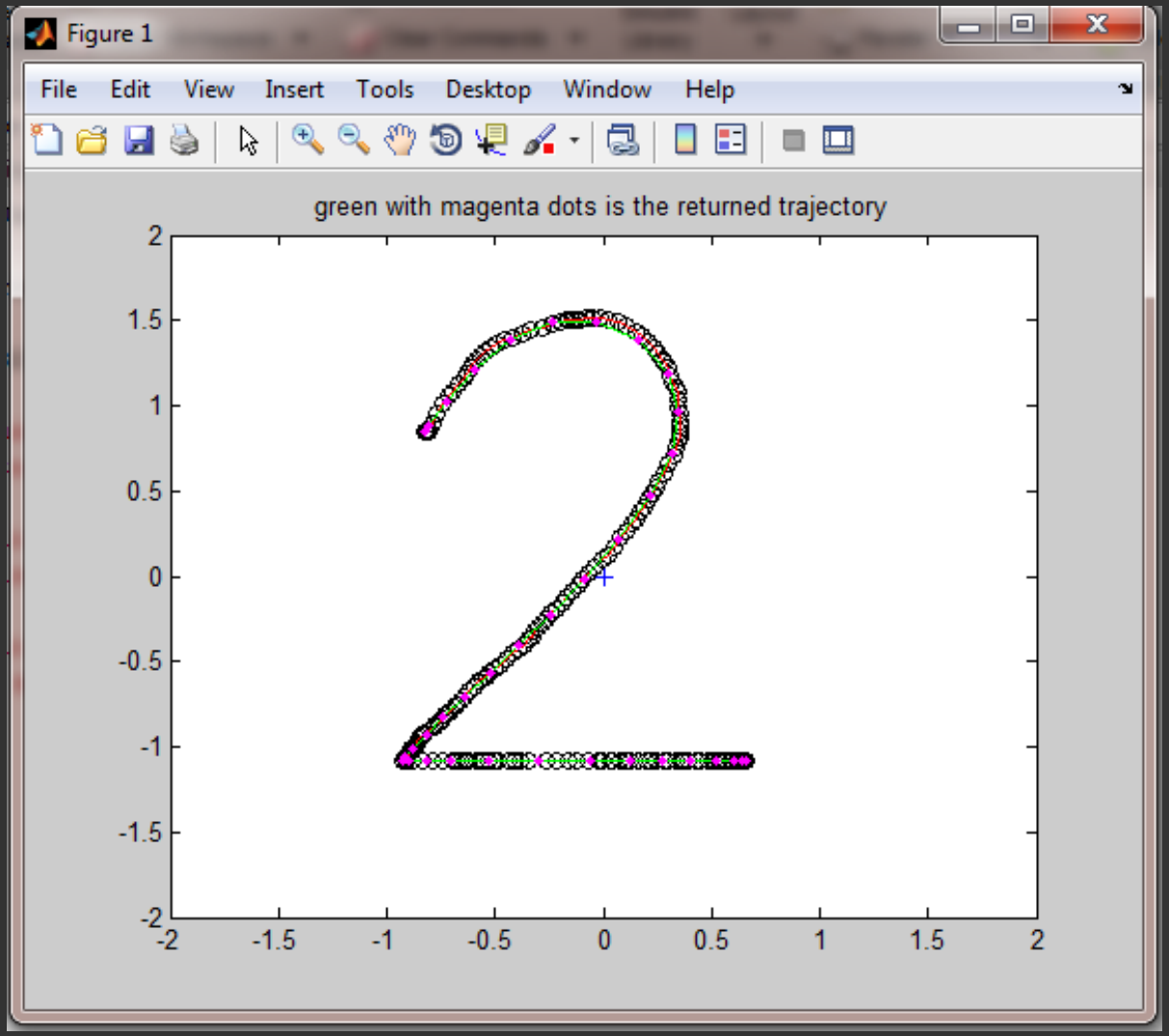


Figure: HR digit which is taken as an input to the data generator - number "2"

>>> Creating Data

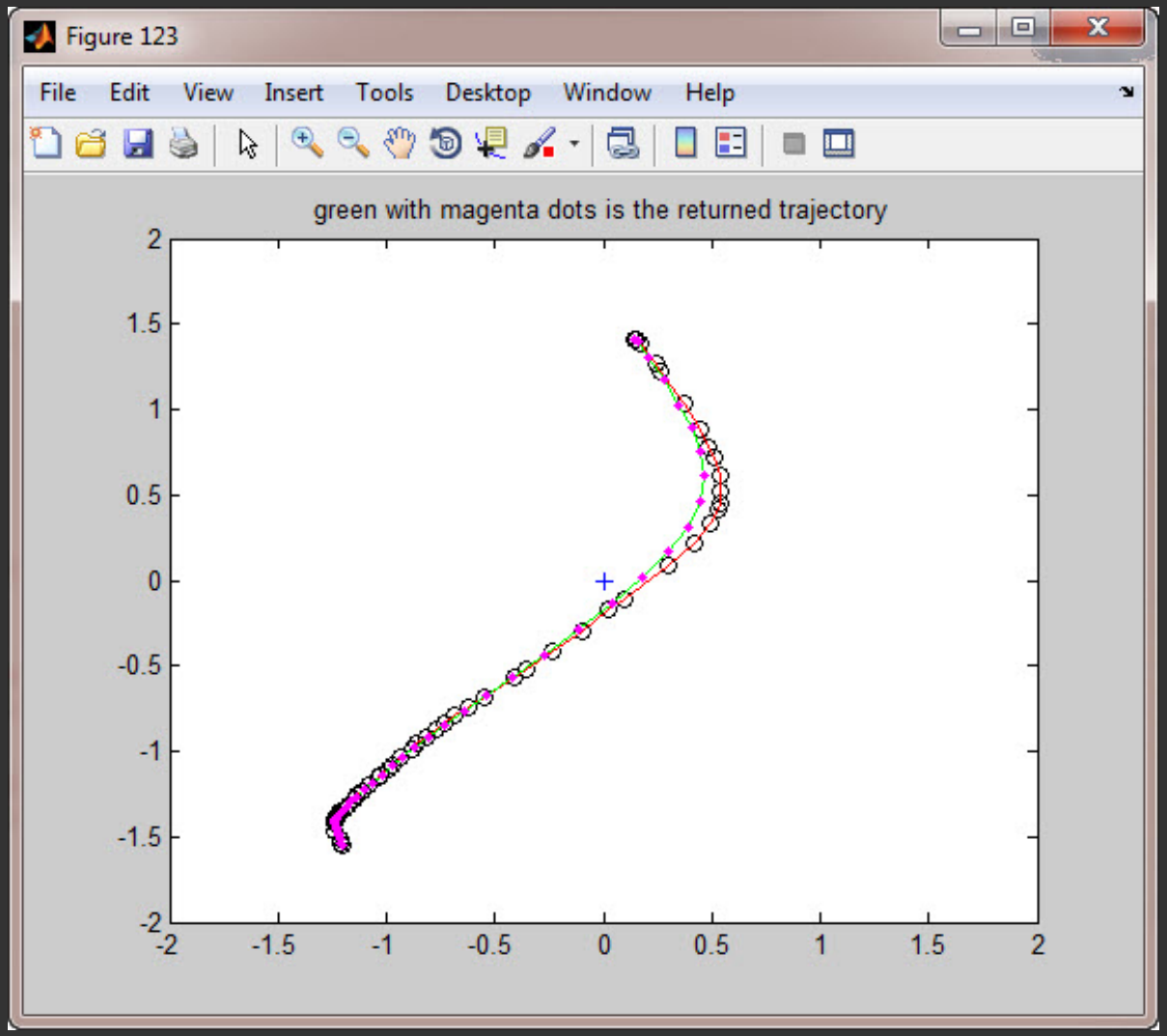


Figure: HR digit which is taken as an input to the data generator - number "7"

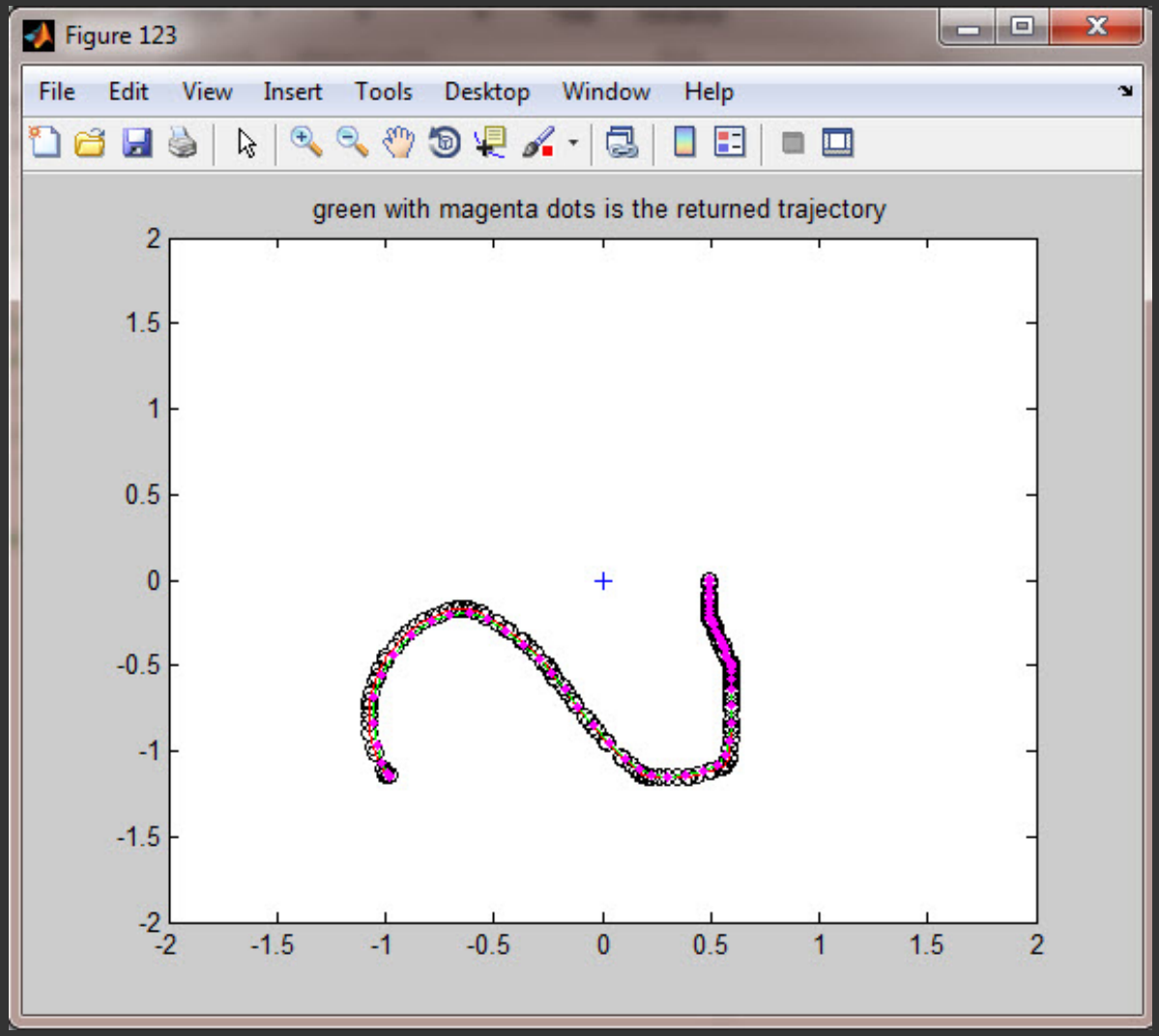


Figure: HR digit which is taken as an input to the data generator - number "2"

>>> Data Structure

* I stored data as $n \times 201$ matrix, where n is nymber of all data
* First 200 columns are $[x\ y]$ coordinates (output of the getUserTraj.m)
* column 201 is target

>>> Neural Network

* I used three layer Neural Network (provided in LMS)
* 50 hidden layers is used
* with one softmax layer in estimating final solution

$$z_i = \frac{\exp O_i}{\sum_i \exp O_i}$$

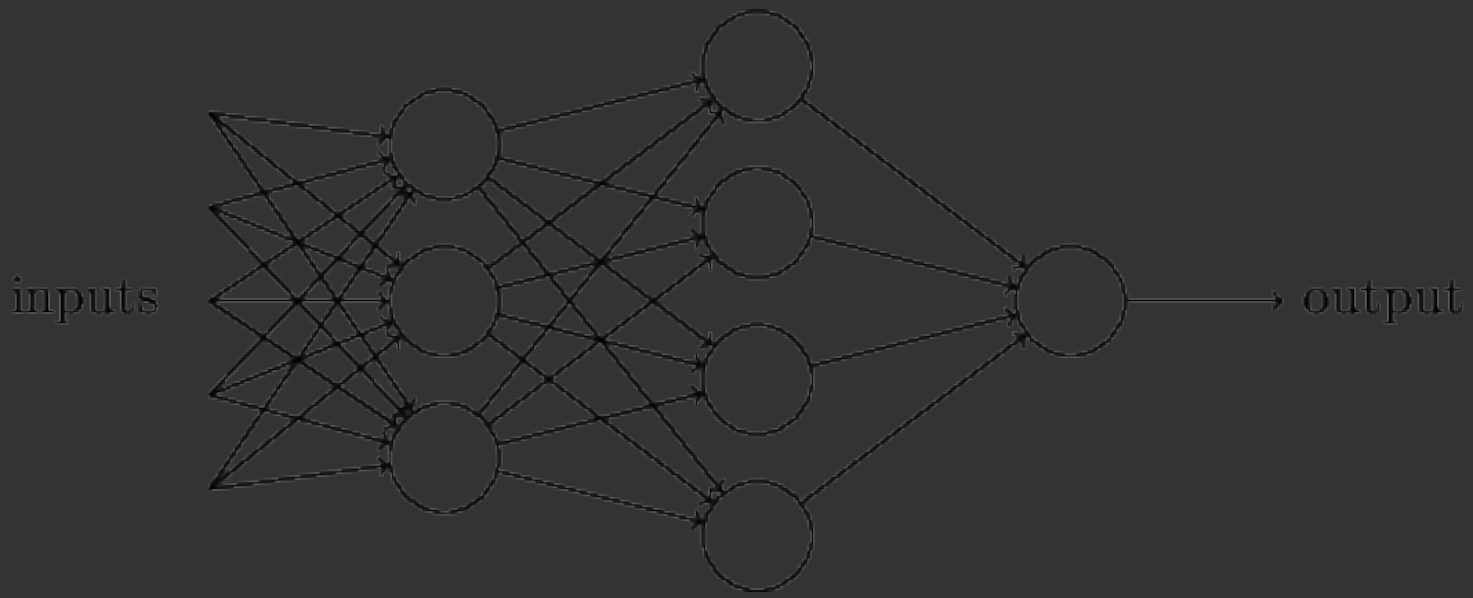


Figure: Structure of the Neural Network

Activation functions: Sigmoid, x-Sigmoid and Leaky RELU

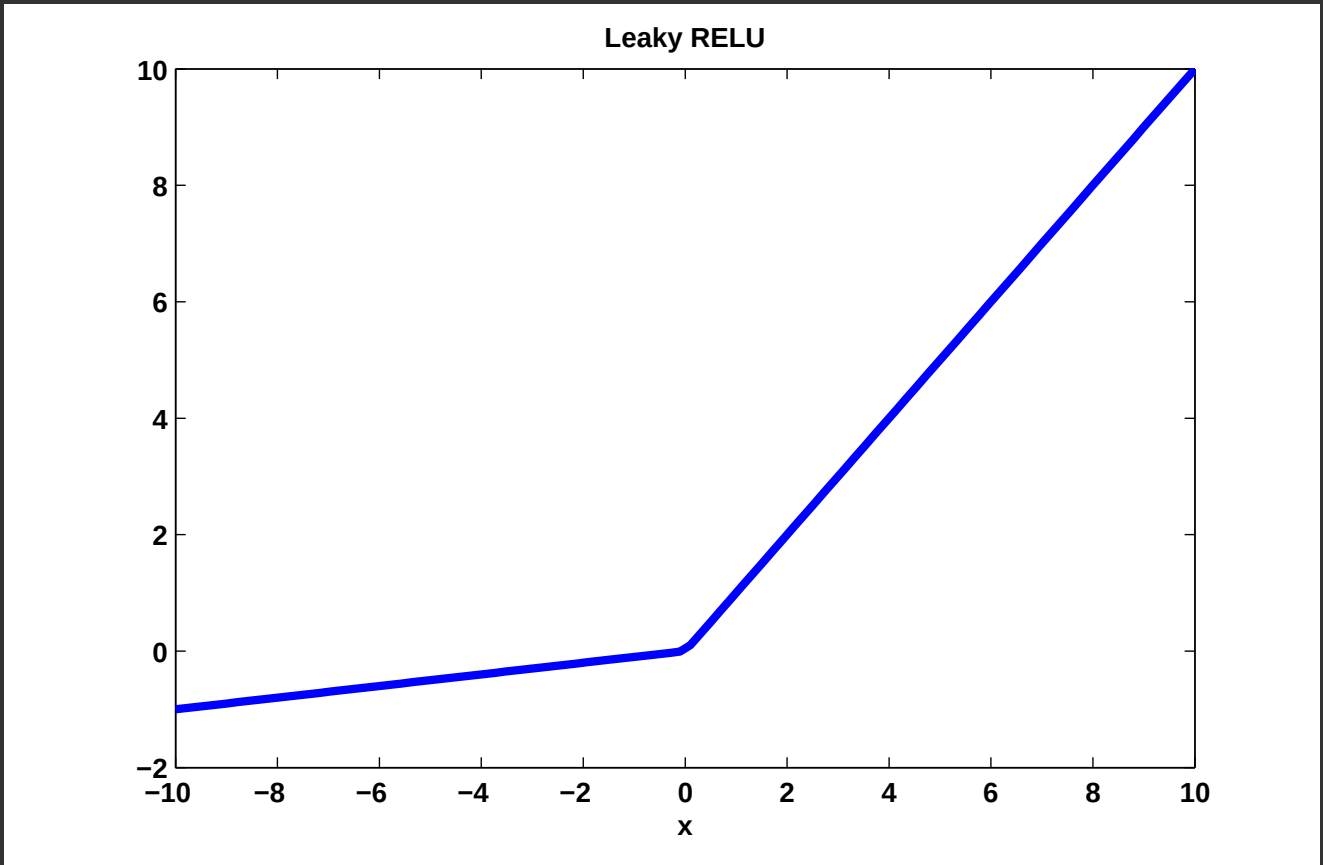


Figure: Structure of Leaky RELU

>>> Hyperparameters

* Size of data: 2000
* Maximum iteration: 1500000
* Learning rate: $\eta = 0.002$
* Noise coefficient: $\beta = 1e - 10$
* Guide vector (Accelerated Gradient) $\mu = 0.001$
* batch size: 20

>>> Preprocessing; Rotation and Scalling

* I prepared data carefully so there is no need to preprocessing to remove dummy data
* for rotation I provided rotated data and let the network to learn it
* for rotation I rotated data randomly using rotation function
* for scaling I normalized the data with mean 0.5 and Variance 0.5 (as best option)

```
function r_data = rotation(Data, rtheta)
```
$\text{Data}_x = Data(:, 1:100);$
$\text{Data}_y = Data(:, 101:200);$
```
sinus = sin(rtheta*pi/180);
cosinus = cos(rtheta*pi/180);
r_data
```
$= [\text{Data}_x * cosinus + Data_y * sinus - Data_x * sinus + Data_y * cosinus; ...$
$\text{Data}_x * cosinus + Data_y * sinus - Data_x * sinus + Data_y * cosinus];$

>>> Scalling

```
function [SData, mean_X, std_X] = standardize(varargin)switchnargin
case 1
mean_X = mean(varargin1);
std_X = std(varargin1);
SData = varargin1 - repmat(mean_X, [size(varargin1, 1)1]);
for i = 1:size(SData, 2)
SData(:, i) = SData(:, i) / std(SData(:, i));
end
case 3
mean_X = varargin2;
std_X = varargin3;
SData = varargin1 - repmat(mean_X, [size(varargin1, 1)1]);
for i = 1:size(SData, 2)
SData(:, i) = SData(:, i) / std_X(:, i);
end
end
```

* MSE cost function
* Accelerated gradient decent
* Stochastic gradient method is applied, each time the batch is trained which increased the accuracy
* Batch samples are secting randomly; MATLAB randi() function

if it==1
vect$_p$1 $= -eta * dW$;
end;
vect1 = mu*vect$_p$1 $- eta * dW$;
W = W - mu * vect$_p$1 $+ (1 + mu) * vect1$;
vect$_p$1 $= vect1$;

>>> Results On Provided Data

As number of data were too small, Network learns, but not good performance in recognizing HR digits

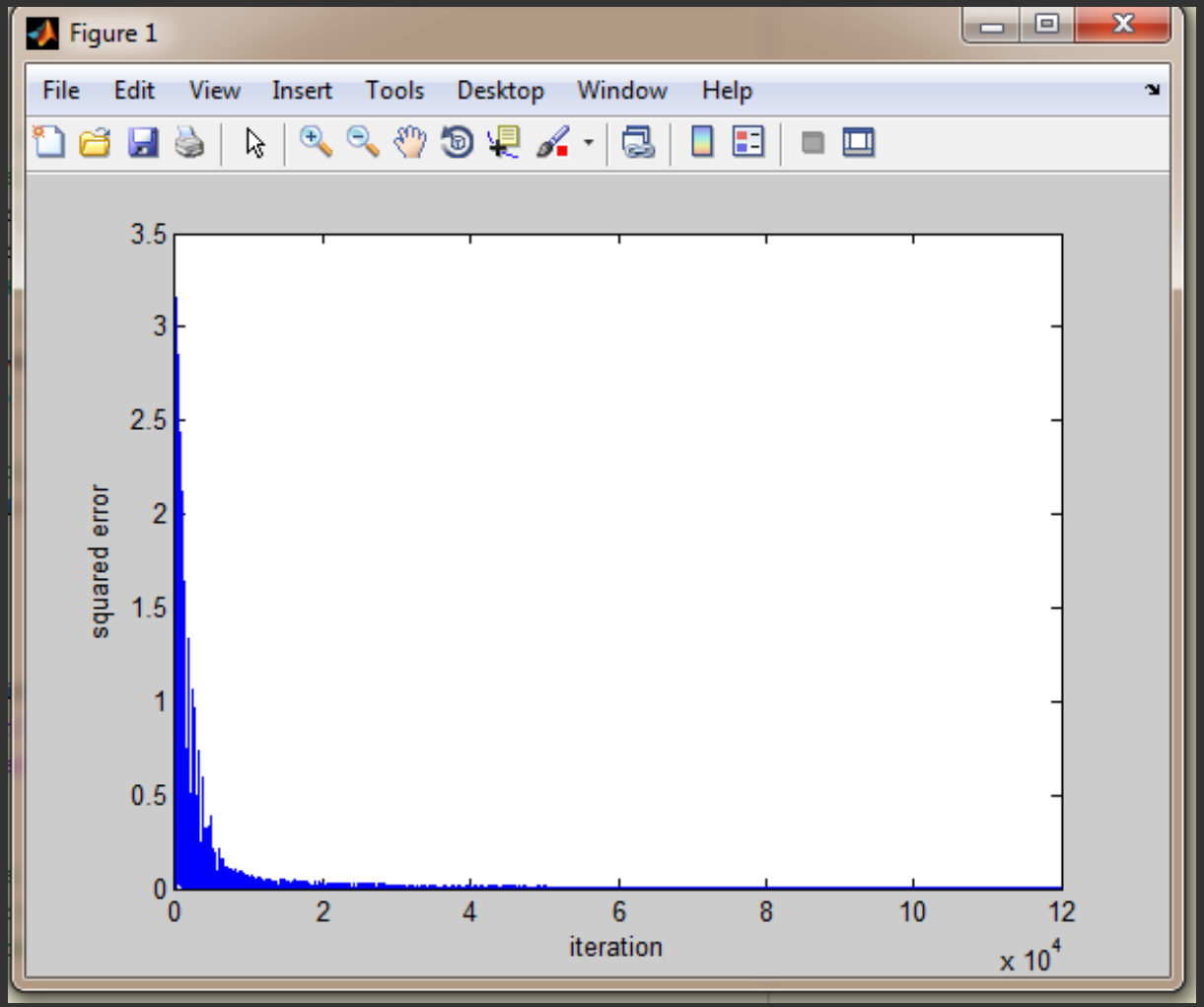


Figure: Results of this network over provided data

>>> Results On Large Data

Number of data large

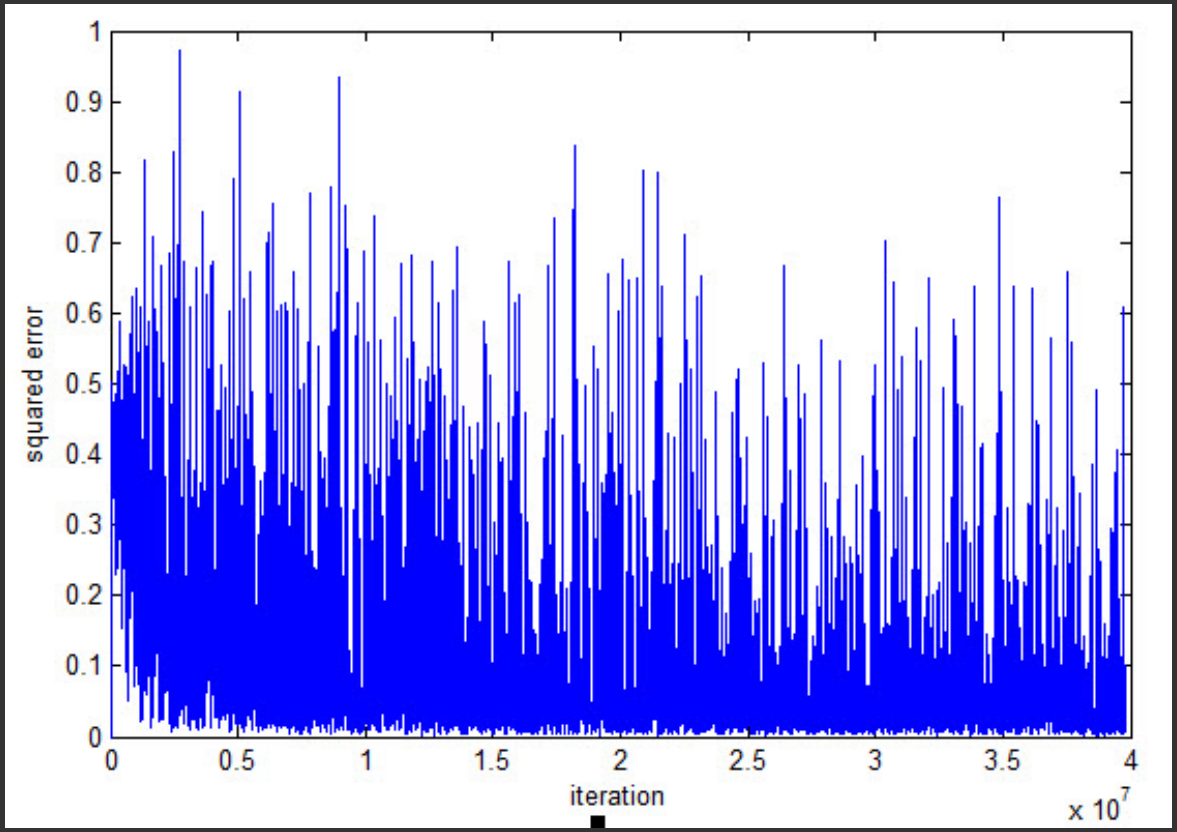


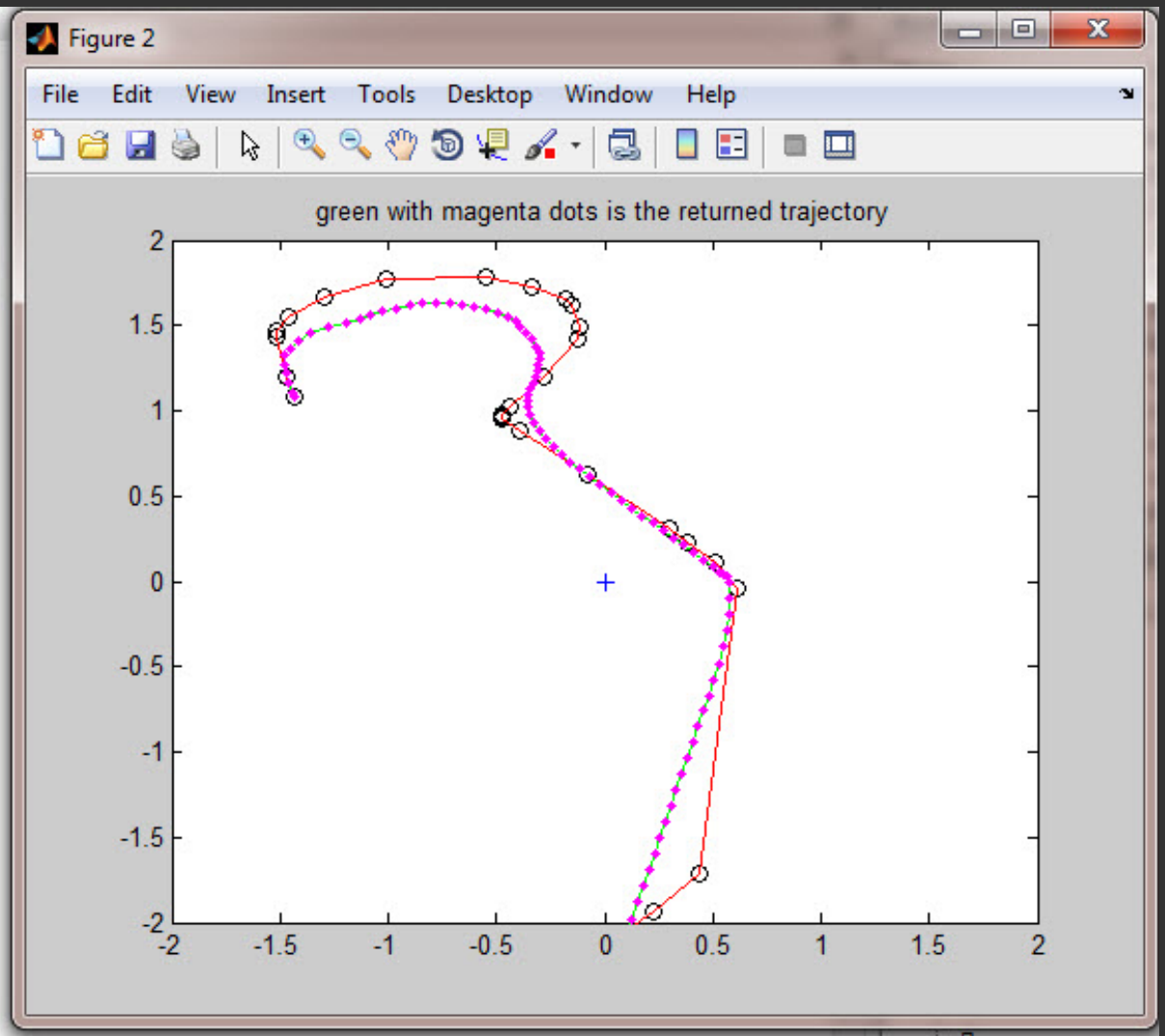
Figure: Results of this network over large data

>>> Results On Large Data

Testing results

Command Window

```
trajduration =

    2.8019

Network digit estimation is 0
:Enter true y: 6

y_true =

     6


trajduration =

    1.9334

Network digit estimation is 6
:Enter true y: 3

y_true =

     3


trajduration =

    1.8483

Network digit estimation is 3
:Enter true y:
```

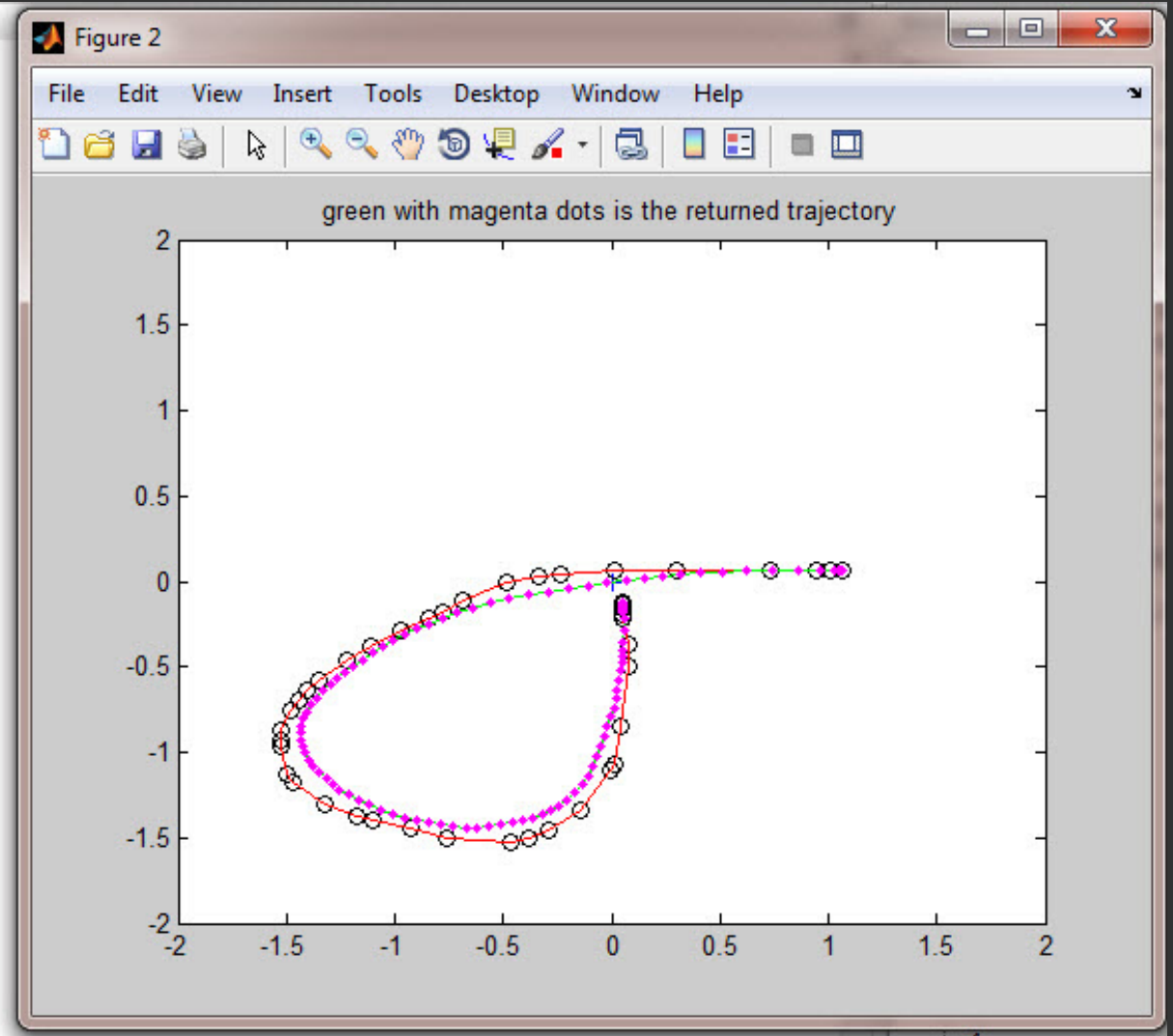


>>> Results On Large Data

Testing results- digit 6 with rotation

Command Window

```
trajduration =

    2.4804

Network digit estimation is 3
:Enter true y: 6

y_true =

     6


trajduration =

    2.8019

Network digit estimation is 0
:Enter true y: 6

y_true =

     6


trajduration =

    1.9334

Network digit estimation is 6
:Enter true y:
```



>>> Future Improvements

* Accuracy percentage is depend on the provided data, but in uniform random case it returns accuracy close to 90%
* Increasing data
* improving convergence
* setting proper hyperparameters
* Applying SVM
* Cross Validation technique

>>> Questions?

Thank you