CMSC 409 - Final Take Home Exam

I fully understand that I am on my honor to do my own work on this exam. If I violate this confidence, I amy receive a letter of "F" for this exam or for the course, or be expelled from the program.

Keith Herbert, 13 December 2015

1. Kohonen Winner-Take-All Unsupervised Clustering

I had a difficult time getting started on this problem. My first fault was using the wrong lecture slides as a reference. I didn't understand how to use only 2 neuron's for a Kohonen network when the slides showed a whole grid of neurons. I also struggled with adapting my language of choice, R, to the problem. Fustrated, I decided to try learning MATLAB to attack the problem but felt overwhelmed and gave up to work on other projects I had due. After finishing a High Performance Computing project in Fortran, I realized that I could save my sanity by starting over again in Fortran. I also found that I was trying to implement the wrong kind of clustering network and started fresh with the slides from Session 11.

My solution to this problem is written in Fortan and uses Gnuplot for visualization. To run it, first go to the `kohonen` directory and compile the Fortran source code into an executable.

```
$ cd kohonen $ gfortran kohonen.f90 -o kohonen.exe
```

To execute it, provide the filename containing the input patterns, the number of neurons, the number of features in each pattern, the learning rate alpha, and an integer to seed the random number generator. Here, we specify the Ex1 dataset with 2 neurons, 2 dimensions, alpha = 0.1 and the PRNG seeded to '12345'. (As a side note, the header line of Ex1_data.txt was removed to simplify reading).

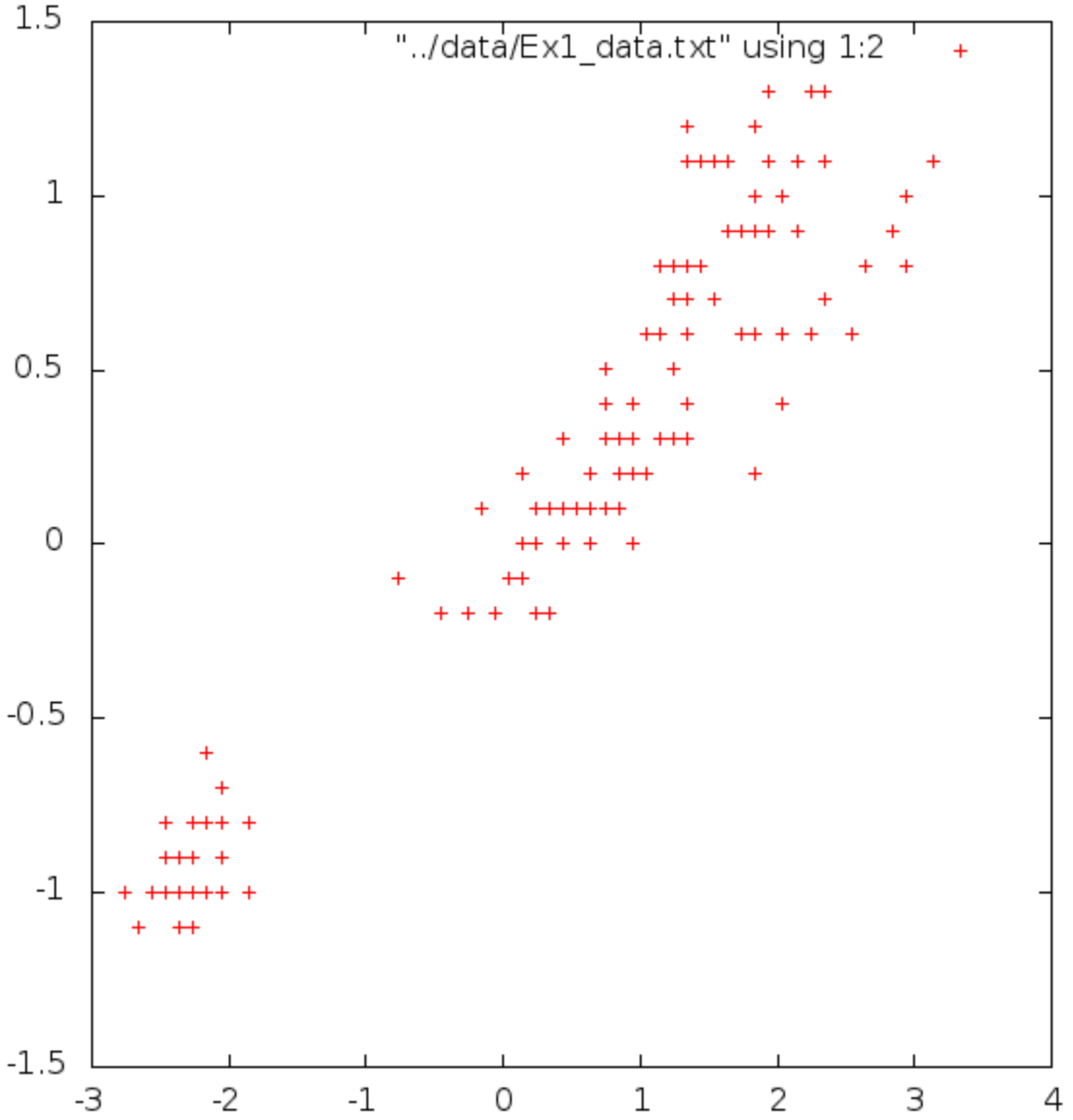
```
$ ./kohonen.exe ../data/Ex1_data.txt 2 2 0.1 12345
```

The program prints the values for each pattern, their associated net values for each neuron, the number of the winning neuron, its new weighting after adjusting by the learning constant, and its normalized new weight. The final weights for each neuron are printed at the end. This can flood your terminal, so it might be best to append a `| less` to the command string. Plotting ready data files are written in the background for the normalized weights of the input patterns (default: `normalized_patterns.txt`) and for the normalized final neuron weights(`neurons.txt`).

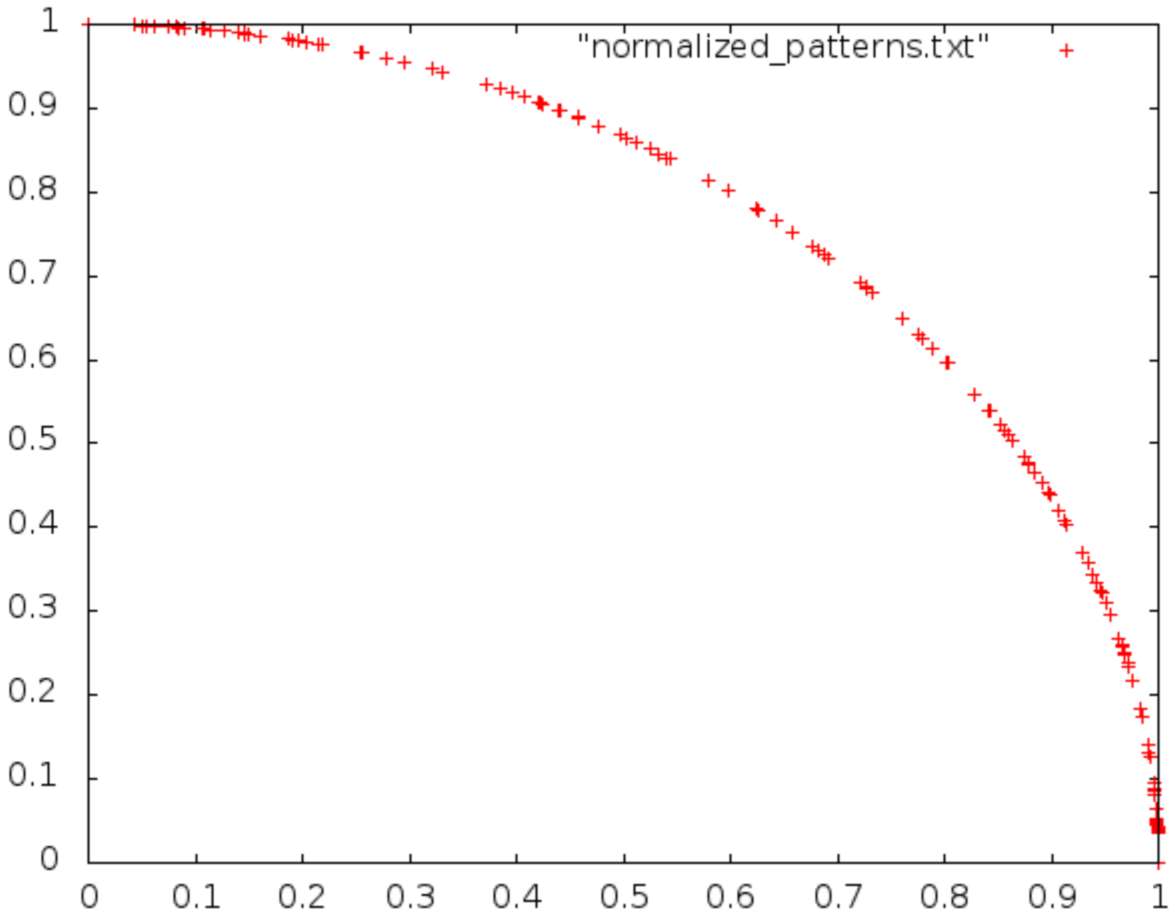
To plot, run gnuplot with the provided script `kohonen.plt`. This generates a .png file with the normalized patterns as tiny red crosses and the final neuron weights as big blue dots.

```
$ gnuplot kohonen.plt Plotting to kohonen_7_neurons.png
```

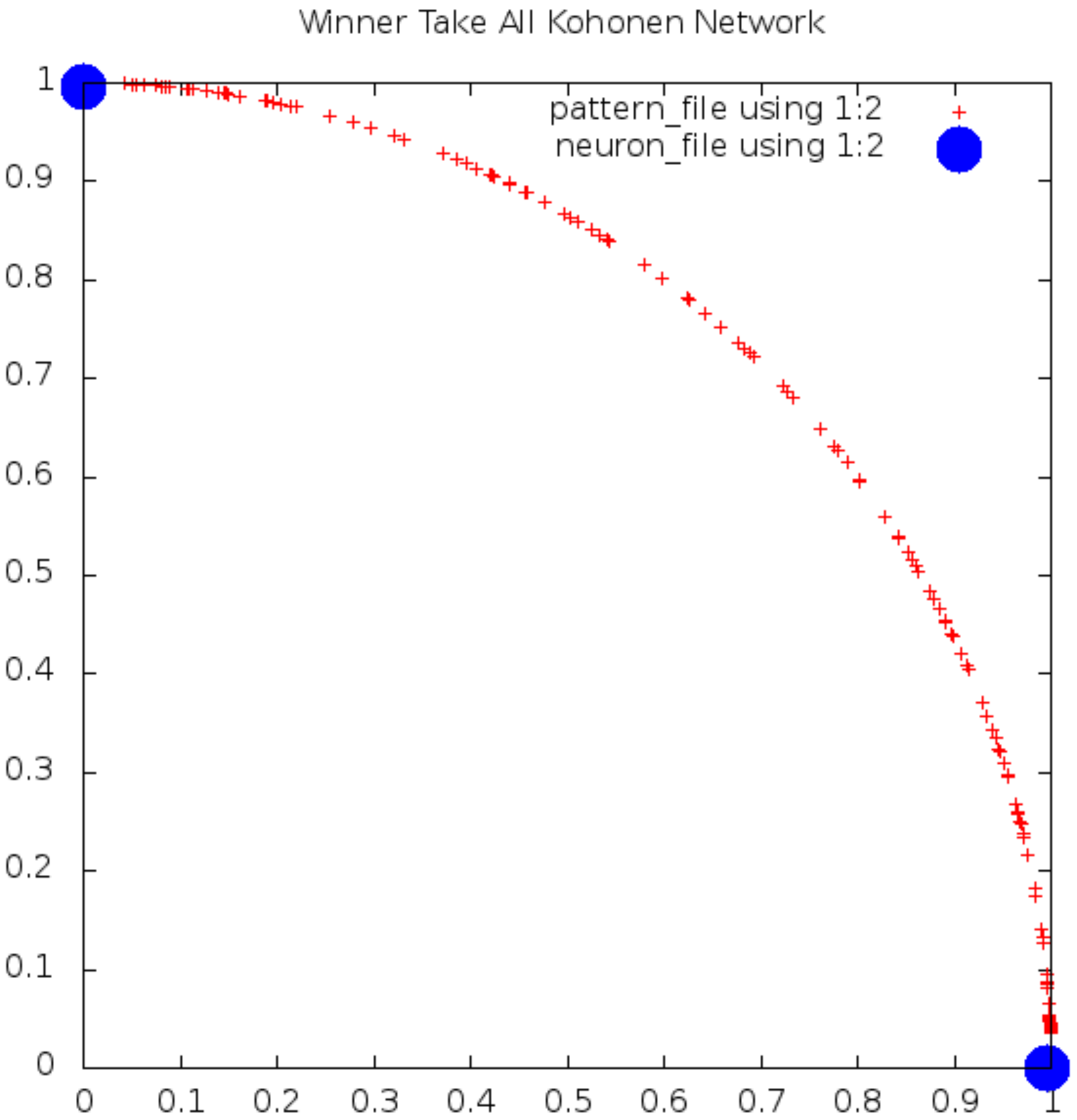
Raw data patterns



Normalized data patterns

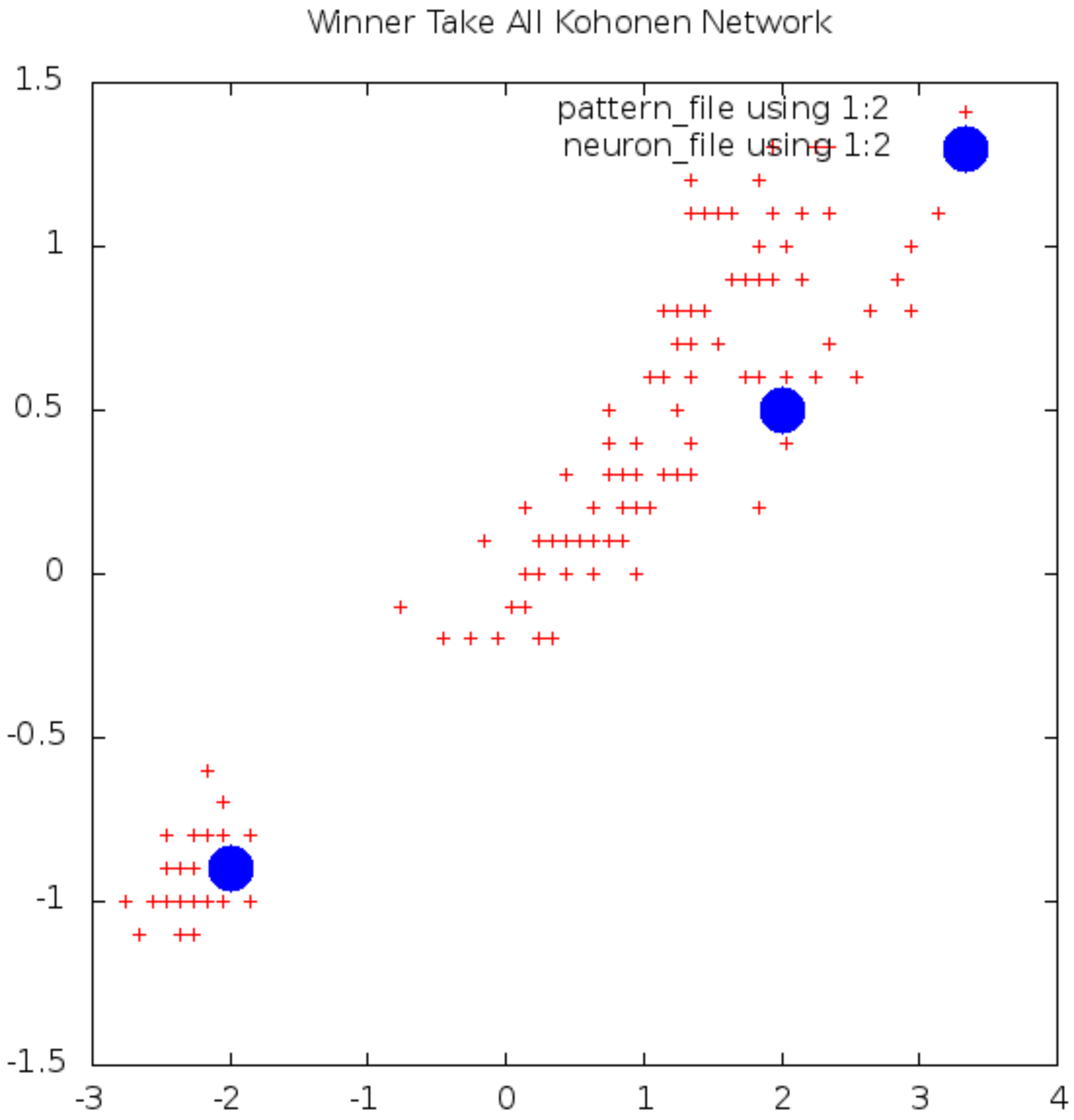


Two-neuron network with random weights

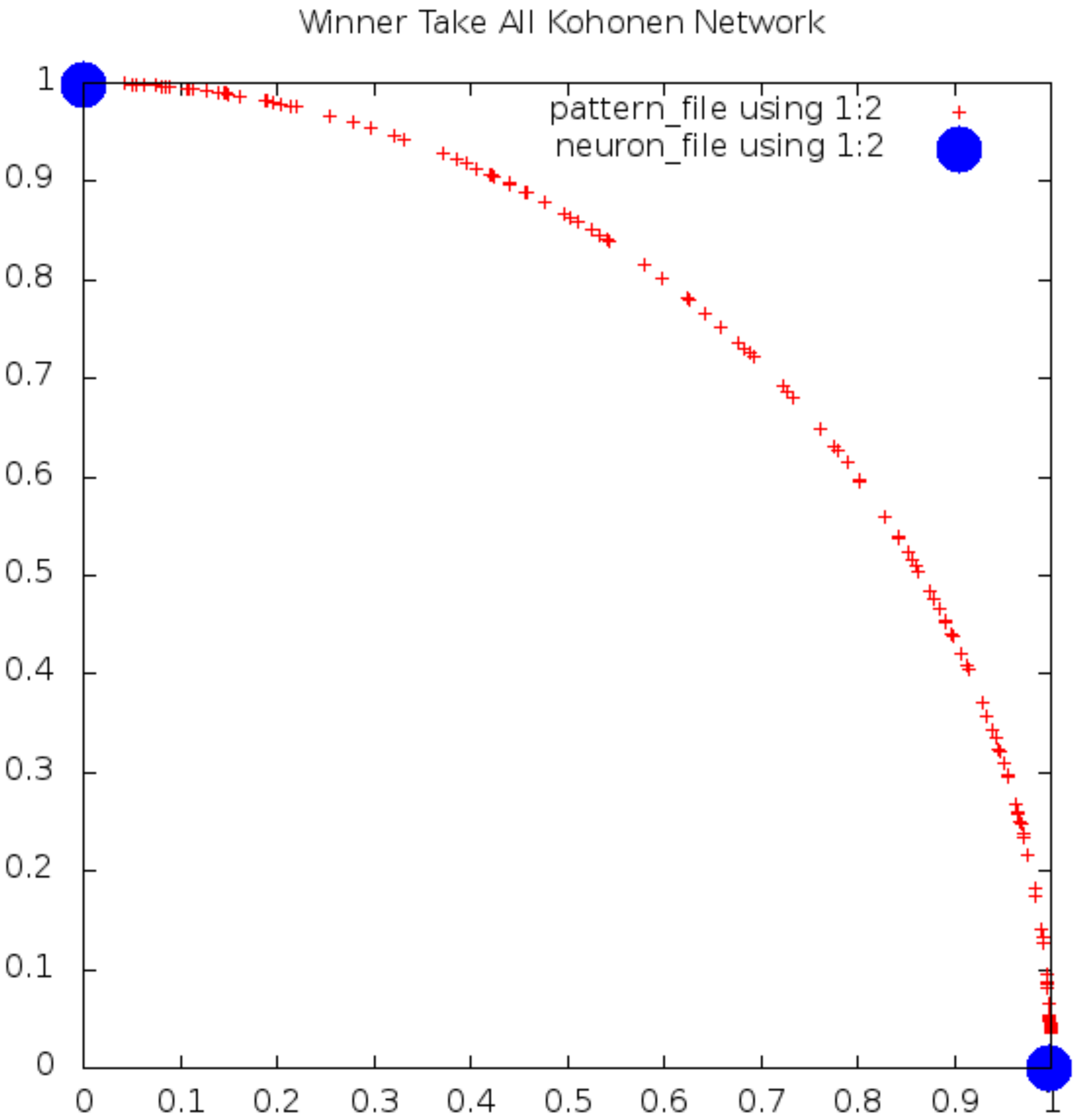


Two-neuron network with pre-set weights

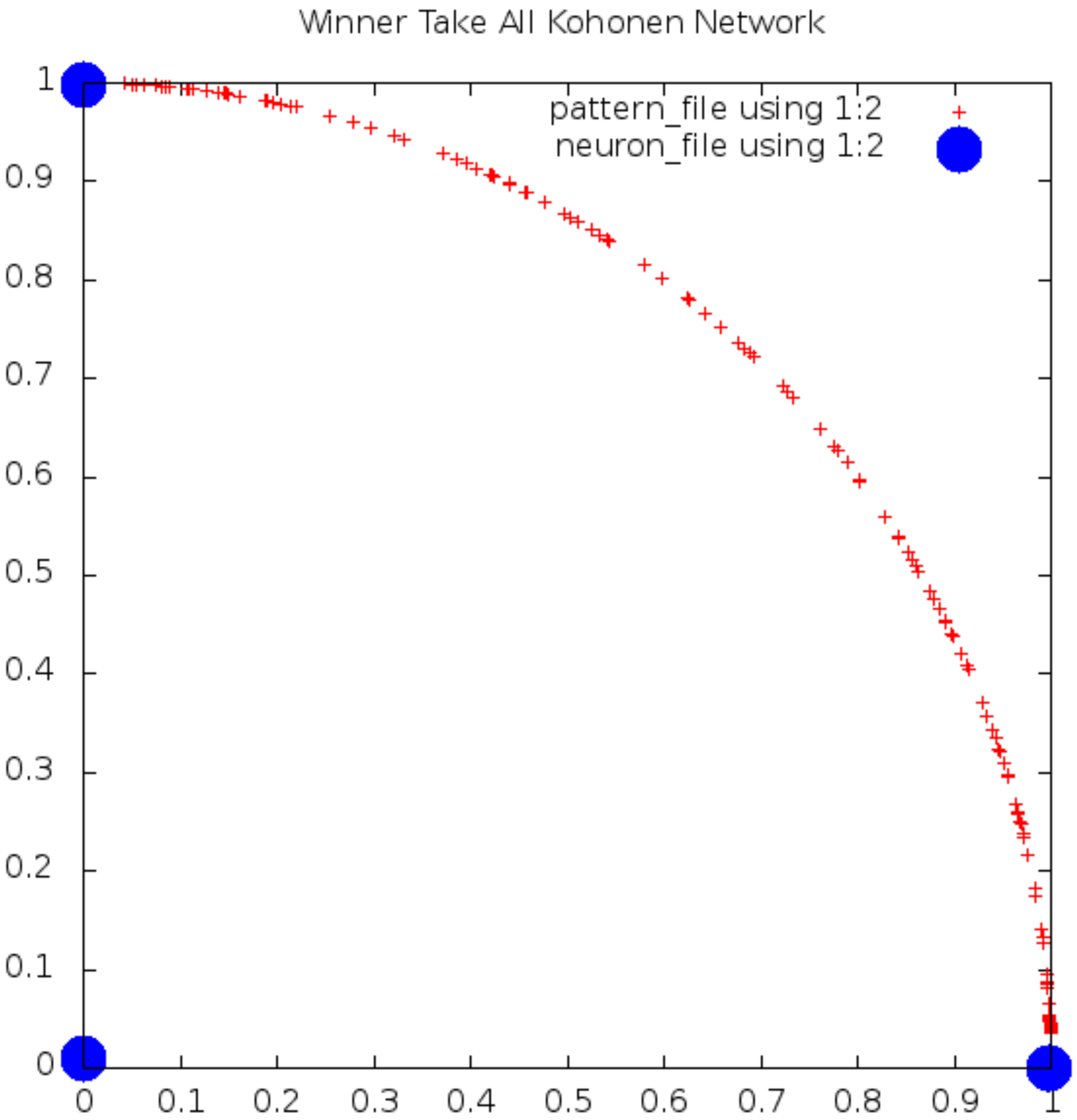
```
$ cat neuron_preset.txt
-2  -0.9
2   0.5
```



```
$ ./kohonen.exe ../data/Ex1_data.txt 2 2 0.1 9999 neuron_preset.txt
```

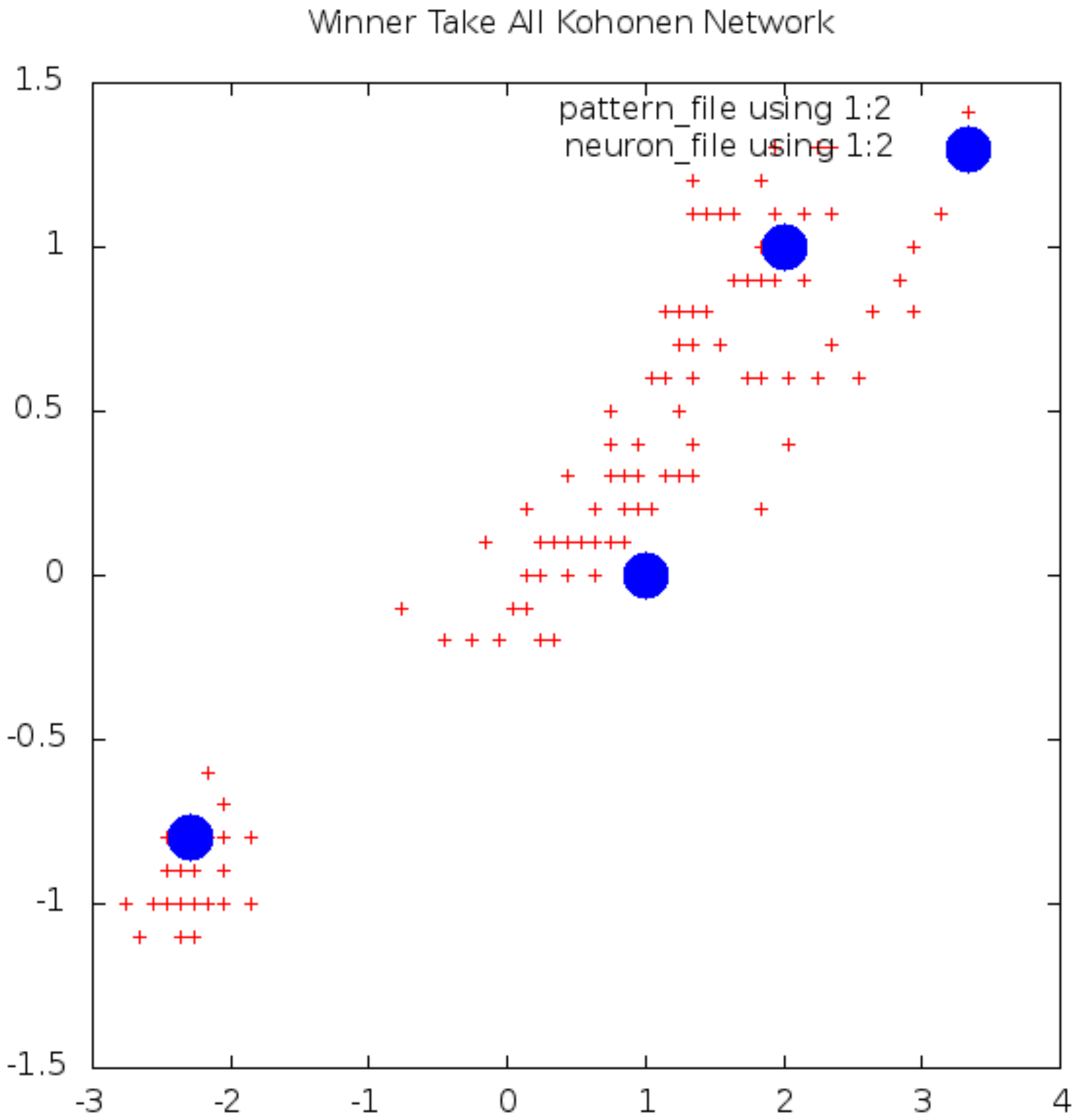


Three-neuron network with random weights



Three-neuron network with pre-set weights

```
$ cat neuron_preset.txt
-2.3, -0.8
1, 0
2, 1
```



```
$ ./kohonen.exe ../data/Ex1_data.txt 3 2 0.1 9999 neuron_preset.txt
```

Seven-neuron network with random weights

Seven-neuron network with pre-set weights

2. Bayesian Classification

3. Zadeh Fuzzy Logic Controller