The MINT neural network library

A quick introduction

Stefano Ghirlanda

July 7, 2014

Problems of existing simulation software

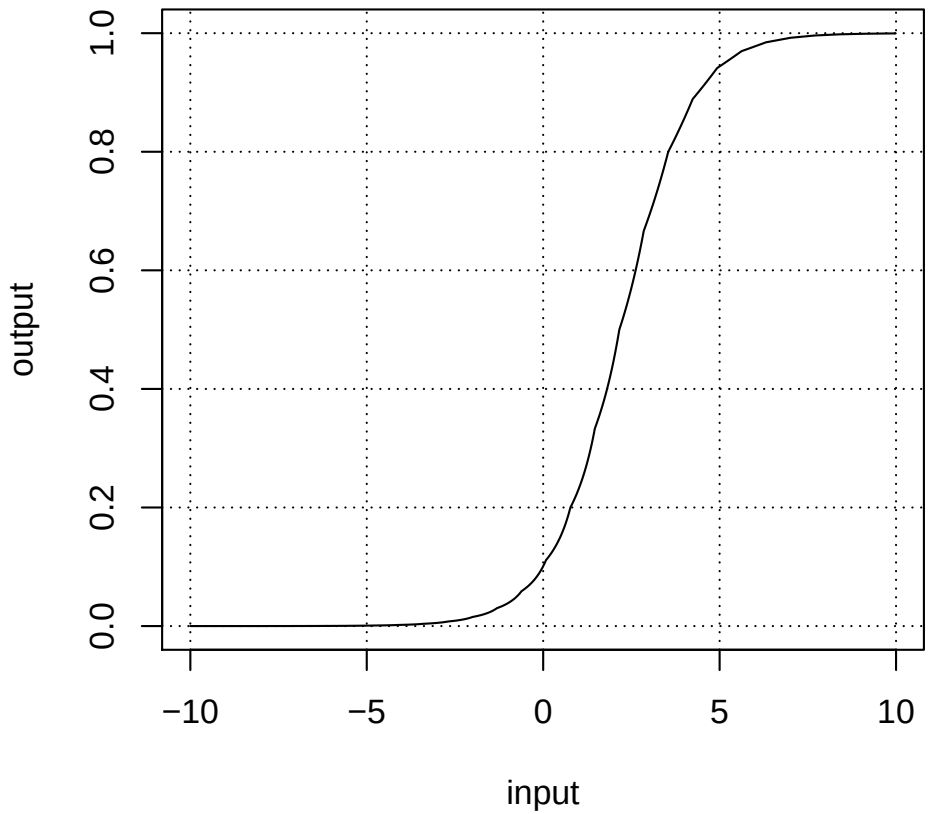
- Engineering view of neural networks
- Not general enough
- Too low level for my (our) purpose
- Makes things complicated
- Costs money

Data structures: Nodes

```
1  mint_nodes n;
2
3  /* create 10 nodes with 1 state variable each */
4  n = mint_nodes_new( 10, 1 );
5
6  n[0][i]; /* input to node i */
7  n[1][i]; /* output of node i */
8  n[2][i]; /* current value of 1st state variable of node i */
9
10 for( i=0; i<mint_nodes_size(n); i++ )
11    n[0][i] = mint_random();
12
13 mint_nodes_update( n );
```

Architecture files

```
1 nodes brain
2 size 10 logistic 0.1 1
```



Data structures: Weights

Weights represent connections between nodes:

```
1  mint_weights w;
2
3  /* weight matrix with 10 rows, 5 columns, 1 state variable each */
4  w = mint_weights_new( 10, 5, 1 );
5
6  w[0][i][j]; /* value of weight in row i, column j */
7  w[1][i][j]; /* 1st state of weight in row i, column j */
```

Data structures: Weights

Weights represent connections between nodes:

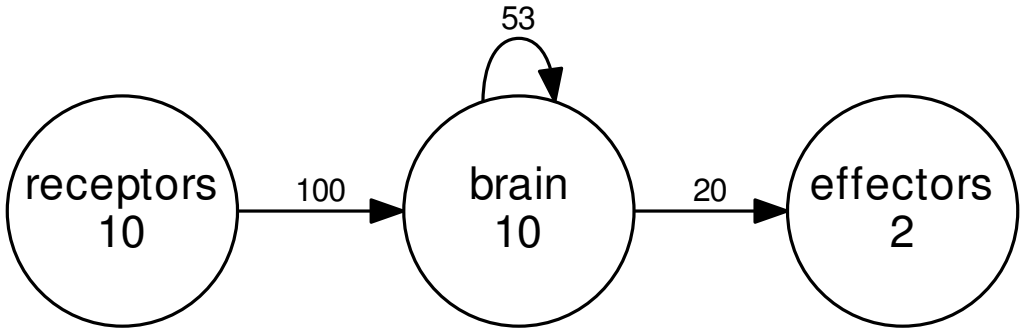
```
1  mint_weights w;
2
3  /* weight matrix with 10 rows, 5 columns, 1 state variable each */
4  w = mint_weights_new( 10, 5, 1 );
5
6  w[0][i][j]; /* value of weight in row i, column j */
7  w[1][i][j]; /* 1st state of weight in row i, column j */
```

In configuration files:

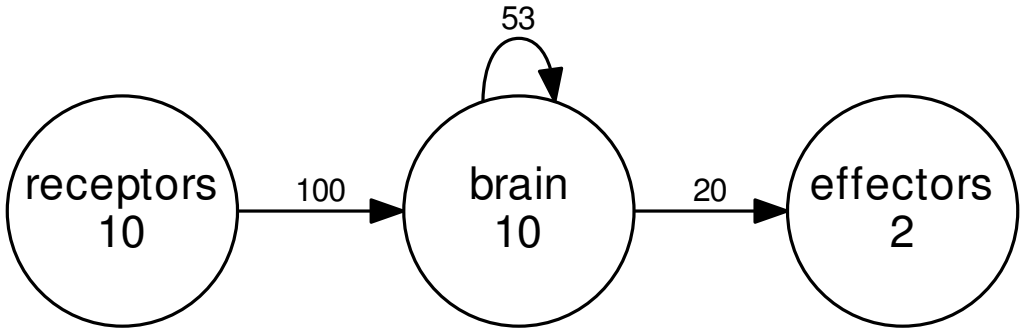
```
1  weights w
2  rows 10 cols 5 states 1
3  uniform -1 1
```

(But there is a better way)

Data structures: Networks



Data structures: Networks




```
1 network
2 nodes receptors size 10 noise 0.1 0.5 0 logistic 0.1 1
3 nodes brain size 10 logistic 0.1 1
4 nodes effectors size 2
5 weights receptors-brain uniform -0.1 1
6 weights brain-brain sparse normal 0 0.1 0.5
7 weights brain-effectors uniform -0.1 1
```

Code (1)

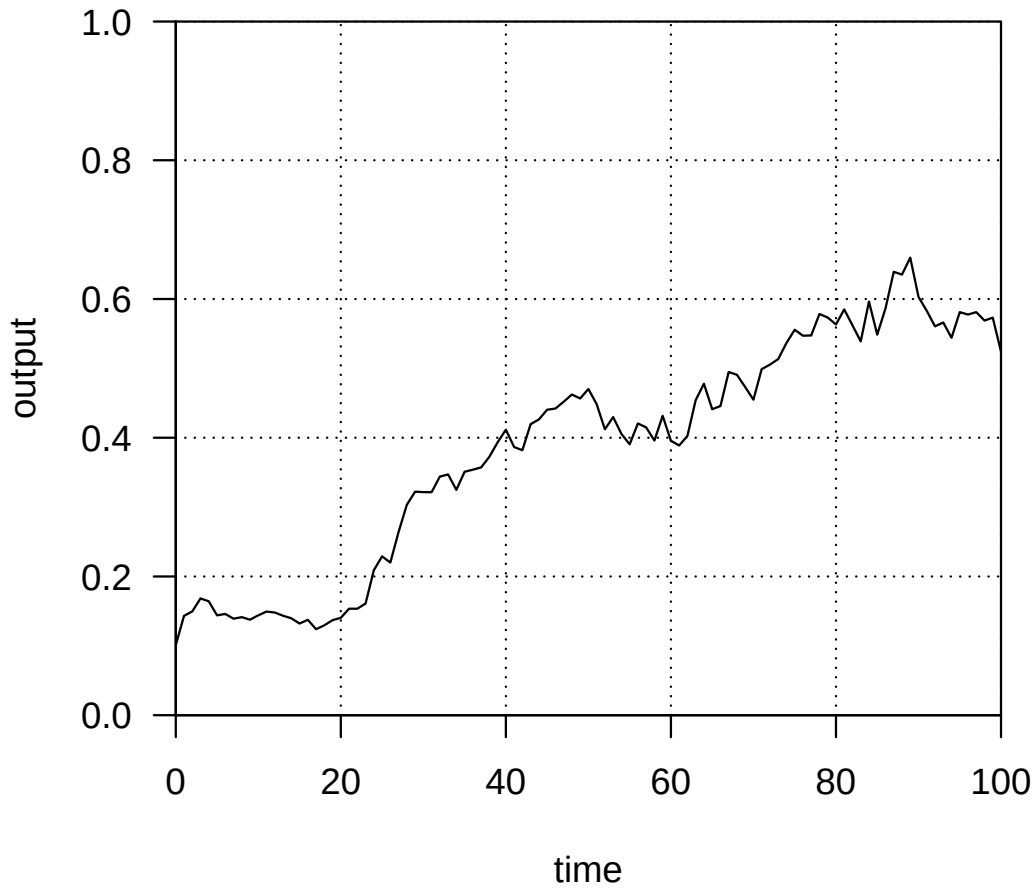
```
1  #include "mint.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <time.h>
5
6  int main( void ) {
7     FILE *f;
8     struct mint_network *net;
9     int t;
10    mint_nodes brain;
11
12    f = fopen( "recnet.arc", "r" );
13    net = mint_network_load( f ); /* load net from file */
14    fclose( f );
15    mint_network_save( net, stdout ); /* display the network */
```

Code (2)

```
17    f = fopen( "recnet.dot", "w" );
18    mint_network_graph( net, f );
19    fclose( f );
20
21    brain = mint_network_nodes( net, 1 ); /* handy shortcut */
22
23    f = fopen( "recnet.dat", "w" );
24    for( t=0; t<=100; t++ ) {
25       mint_network_operate( net );
26       fprintf( f, "%d %f\n", t, brain[1][0] );
27    }
28    fclose( f );
29
30    mint_network_del( net ); /* free memory */
31    return EXIT_SUCCESS;
32 }
```

Results

If you run this network, you get an output file with the output of "brain" cell 0 in response to noise input to the network:



What next

- Is it usable?
- Better documentation
- Better learning facilities
- Visualization has some rough edges
- Tests
- Feature requests...