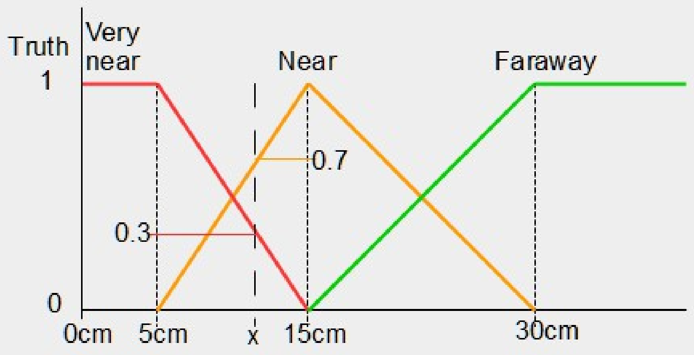
NNFuzzy Library

NNFuzzyLib implements a specific Neural Network that uses a kind of RBF (Radial Basis Function) to realize a Fuzzy System.

A Fuzzy System traslates input values into classes (Fuzzy Sets) then uses a list of rules "if...then..." that describes the functional link between input and output. Output values are expressed in terms of output classes (Mamdani system inference) or in terms of singletons (Sugeno system inference). In this case we use Sugeno version.

Example of fuzzy set for a distance sensor;
sensor value will be represented by its class membership degree.
That means that distance can be: Very-near, Near or Faraway



If value of sensor is x then it means that its truth value is:
very near 30% , near 70% , faraway 0%

Rules example:

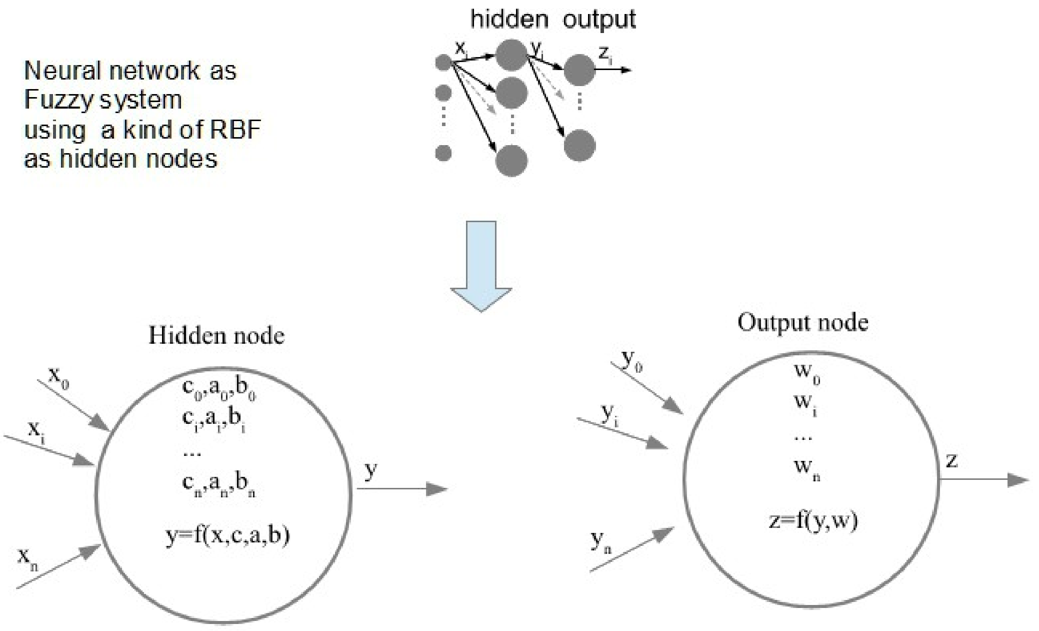
If sensor-1 is Near and sensor-2 is Faraway then motor-1 has to be Medium and motor-2 has to be Low
If sensor-1 is Faraway and sensor-2 is Faraway then motor-1 has to be High and motor-2 too
:
:

Where motor Low, Medium and High means for example power percent 0.3 , 0.6 and 1.0 (Singletons)

Each rule result: minimum truth value of antecedents (fuzzy logic AND = minimum truth value);
this result is applied to consequents as weight for motor power fraction.

Inference result (following Sugeno singletons method):
For each motor the total result is the weighted sum of each rule result (barycenter)

The Neural Network that implements this engine uses a specific local activation system for hidden nodes and a weighted sum as output nodes. The local activation system is made by a specific RBF node as represented below.

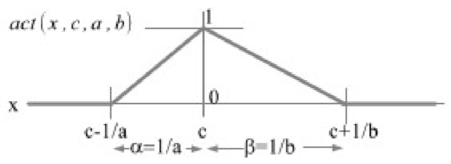


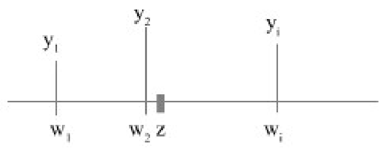
$$act_i(x_i, c_i, a_i, b_i) = \begin{cases} if\ (x<c) & act_i = max(0,\ 1-a_i*(c_i-x_i)) \\ if\ (x>c) & act_i = max(0,\ 1-b_i*(x_i-c_i)) \end{cases}$$

$$y = f(x, c, a, b) = min_0^n(act_i)$$

$$z = f(y, w) = \frac{\sum w_i * y_i}{\sum y_i}$$

(barycenter of rules activation)





Anyway this library can be used in the same way as NNet library. In a word, the library function are the same. As the NNet library, NNFuzzy on Arduino or on PC. Some features (as save on file or load from file) are availlable only if you comment this define:

```
#define ARDUINONNFuzzy
```

A utility C program is provided, to trasform a Fuzzy description of a problem to the exact equivalent NNFuzzy network. The utility is provided as source and as Windows (V10) executable:

- FuzzyDefine.cpp

- **FuzzyDefine.exe** (Windows 10)

Format: FuzzyDefine.exe *filename-fuzzy-definition.txt*

Where the fuzzy definition file is a file where are used this keywords: nput,output,inpfuzzyset,outclass,if,and,then.

Example: (/* are comments)

Ardusumo rover Fuzzy model	

input 2
output 2
/* sensor 0 and 1 classes */
inpfuzzyset 0 L=0.000 V=0.400 VV=0.800
	// input (sensors) number
	//output (motors) number
	
	// L:faraway, V:near, VV:very near

```
inpfuzzyset 1 L=0.000 V=0.400 VV=0.800     // L:faraway, V:near, VV:very near
/* output classes (singletons)*/
outclass BD -0.200 -0.800                  //back and right rotate
outclass RD 0.400 -0.400                   // right rotate
outclass RS -0.400 0.400                   // left rotate
outclass AM 0.600 0.600                    // go medium speed
outclass AS 1.000 0.600                    // go and left rotate
outclass AD 0.600 1.000                    // go and right rotate
outclass AV 1.000 1.000                    // go fast
/* Rules */
if 0=VV and 1=VV then BD
if 0=VV and 1=V then RD
if 0=V and 1=VV then RS
if 0=V and 1=V then AM
if 0=V and 1=L then AS
if 0=L and 1=V then AD
if 0=L and 1=L then AV
```

FuzzyDefine.exe tranform it into:

Fuzzy Neural Network definition

```
char* netdef=
"L0 2 "
"L1 7 NodeFuzzy "
"FCT0 0.8 0.8 "
"FCT1 0.8 0.4 "
"FCT2 0.4 0.8 "
"FCT3 0.4 0.4 "
"FCT4 0.4 0.0 "
"FCT5 0.0 0.4 "
"FCT6 0.0 0.0 "
"FWA0 2.5 2.5 "
"FWA1 2.5 2.5 "
"FWA2 2.5 2.5 "
"FWA3 2.5 2.5 "
"FWA4 2.5 0.1 "
"FWA5 0.1 2.5 "
"FWA6 0.1 0.1 "
"FWB0 0.1 0.1 "
"FWB1 0.1 2.5 "
"FWB2 2.5 0.1 "
"FWB3 2.5 2.5 "
"FWB4 2.5 2.5 "
"FWB5 2.5 2.5 "
"FWB6 2.5 2.5 "
"L2 2 NodeDFTnh "
"OLW0 -0.2 0.4 -0.4 0.6 1.0 0.6 1.0 "
"OLW1 -0.8 -0.4 0.4 0.6 0.6 1.0 1.0 "
;
```

And now you have just to instanziate Neural Network: NNfuzzy net(netdef). And use it.