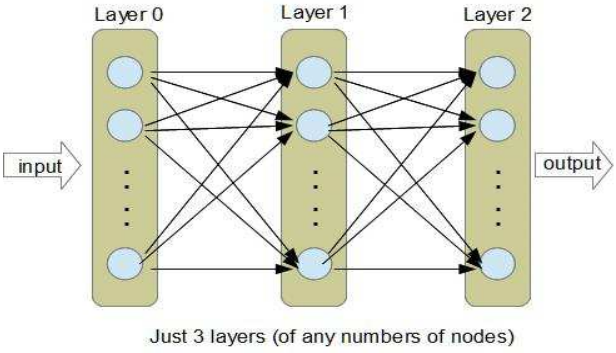
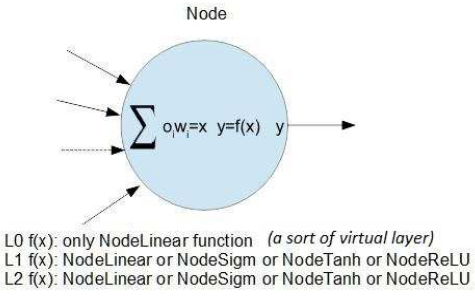
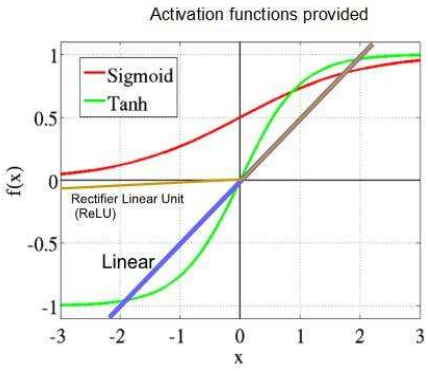
NNet Library

NNet library contains simplified functions to create, load, train and use a basical Neural Network with 3 layers. Actually there are only **2** buffered layers in memory, because input layer is provided by input array. (For this reason a 3 layers NN is offen called 2 layers Neural Net)



Just 3 layers (of any numbers of nodes)





This library can be used on Arduino or on PC. Some features (as save on file or load from file) are availlable only if you comment this define:

#define ARDUINONNET

First of all, instance:

Include library, then instance class by two ways:

1) a new random NN to train
2) a trained and well defined NN for using it

New NN

NNet(int L0n, int L1n, const char* L1Type, int L2n, const char* L2Type);

L0n : number of nodes for L0;

L1n : number of nodes for L1 (hidden layer)

L1Type: name of defined activation function for layer 1

L2n : number of nodes for L2 (otput layer)

L2Type: name of defined activation function for layer 2

Activation functions names:

- "NodeLin"
- "NodeSigm"
- "NodeTanh"
- "NodeReLU"

Es.: NNet net(2,2,"NodeTanh",1,"NodeLin");

Load trained NN

NNet(char netdef[]);

or

NNet(const char* filename);

Example of definition by string or by records of file:

```
char* netdef=
    "L0 2 "                     //layer 0 with 2 nodes
    "L1 2 NodeTanh "            //layer 1 with 2 nodes NodeTanh
    "HLW0 -2.3404 -2.3427 "     //HiddenLayerWeightsOfnode0 valfromN0ofL0 ...
    "HLW1 0.4820 0.4669 "       //  "       "    "        1    "        " ...
    "L2 1 NodeLin "             //layer 2 with 1 node NodeLin
    "OLW0 -2.3558 -3.0901 ";    //OutputLayerWeightsOfnode0 valfromN0ofL1 ...
NB. Always separate tokens  with spaces ! (even if it's the last one in the line)
```

Train or use NN

Train NN

float learn(float inp[],int diminp,float train[],int dimtrain);

inp: array of input values to transmit to NN

diminp: dimension of inp array

train: array of correct output values NN has to learn

dimtrain: dimension of train array

Function returns the squared error.

Es.: float err=net.learn(inp,2,trn,2);

Use NN

void forw(float inp[],int diminp,float out[],int dimout);

inp: array of input values to transmit to NN

diminp: dimension of inp array

out: array that receives output values from NN

dimout: dimension of out array

Es.: net.forw(inp,2,out,2);

Services

print();

save(char* filename);

savexardu(char* filename);(save as string on file)

Library load on Arduino Uno

Some approximate values:

- Library: <8K (<25% of program space)
- Neural network (some examples of maximum size):
  - 6 inp, 24 hid, 8 out
  - 10 inp, 14 hid, 10 out
  - 2 inp, 50 hid, 2 out

In any case I suggest you to verify if RAM is not full, printing immediately the NN instanced.
