Artificial neural networks in Matlab - preliminary information

1 Preliminary information

1.1 Activation functions

There are many activation functions available in Matlab, e.g

- hardlim - Heaviside'a - Warning! It is not suitable for trained networks because it is not differentiable
- purelin - liniear
- logsig - Fermi - values from 0 to 1 `f(x) = 1/(1 + exp( x))`
- tansig - hyperbolic tangent - values from -1 to 1

Activation functions can be easily demonstrated by drawing it on the screen. E.g:

```
n = -5:0.1:5;
plot(n,hardlim(n),'c+:');
```

Users can also create their own activation functions.

1.2 Creating a network

To create networks, they are used functions:

- newp - creates a single perceptron
- newff - creates feed-forward network
- newhop - creates Hopfield network

Creating a two-layer (with one hidden layer) feedforward network with two inputs:

```
net = newff([0 10; 0 10],[5 1],{'tansig' 'purelin'});
```

The created network has 5 neurons in a hidden layer and one output neuron. Input values should be in the range <0; 10>. Matlab's neural networks are objects, so you can access their parameters. The most important parameters are:

- network input weights - from inputs to first n-th layer - `net.IW{n}` – n - layer index (always 1 for simple newff network)

- weights between layers- `net.LW{n, m}` – n - target layer index, m - previous layer index

- threshold values - `net.b{n}` – n - layer index

- number of iterations during learning - `net.trainParam.epochs`

- acceptable error- `net.trainParam.goal`

- number of steps in the training algorithm after which the message is displayed - `net.trainParam.show`

The above parameters can be saved and read.

Before you start training a network, you must initialize it:

```
net = init(net);
```

The `train` function is used to train the network, e.g .:

```
net = train(net, X, Y);
```

where, X - vector of arguments and Y - vector of the value of the learned function.
You can experiment with the learned network using the sim function, e.g .:

```
Y = sim(net, T);
```

where, T - test vector, Y - result of network operation.