Matlab Neural Network Summary

By Hilal Huda (CAS, Telkom University, email: hilalh@kfupm.edu.sa)

Declaration

Declaring single layer neural network (NN)

```
nets=fitnet([ ]);
```

Declaring a multilayer NN, with units N1 N2 in the hidden layers

```
net=feedforwardnet([N1 N2]);
```

Defining the number of units in layer n with number of units N

```
net.inputs{n}.size =N;
```

Defining type of activation function of a layer n

```
net.layers{n}.transferFcn = 'logsig';
 %'logsig' for sigmoid, 'purelin' for pure linear, 'tansig' for hyperbolic tangent
```

Manually defining weights

```
net.IW{1,1}=W;% input weights
net.LW{n,n-1}=W;% layer n weights
net.b{n,1}=b;% layer n bias
view(net); %viewing the network architecture
```

Training

Defining the training algorithm

```
net.trainFcn = 'trainscg';% scaled conjugate gradient training algorithm
%'trainlm' Levenberg-Marquart training algorithm, default
```

Defining cost function

```
net.performFcn = 'mse';% mean squared error cost function
% crossentropy for cross entropy cost function
```

Manually defining training sets

```
net.divideFcn='divideind';
net.divideParam.trainInd=training_index;
net.divideParam.testInd=test_index;
net.divideParam.valInd=val_index;
```

Stopping criteria

```
net.trainParam.epochs=1000;
net.trainParam.min_grad=1e-6;
net.trainParam.max_fail=6;
```

Training declaration

```
[net,tr,Y,E] = train(net,D,T);
```

This instruction will train the network with data input $D \in R^{N \times S}$ and target $T \in R^{R \times S}$, where S is the number of samples. net is the trained network, tr training details, Y output after training, E error after training

Obtaining the output from a network with input in

```
out=net(in);
```