what happen in neural net Back propagation in Matlab

I am newbie in MATLAB, I want to verify the online back propagation(BP) code in C. I need to test the code whether it is exactly the same with the same network setting. The network setting is original BP (for XOR problem) 2 inputs, 2 hidden nodes and 1 output. The learning rate setting used is 0.01, momentum 0.95 while stopping criteria is 0.01 and the performance measure is sse. the epoch is 1 (because I want to check the exactly calculation from forward propagation to backward propagate, in order to verify the network setting exactly the same as in C) here is my code:

```
clear all;clc
input =  [0 0; 0 1; 1 0; 1 1]';
target = [0 1 1 0];   % p = [-1 -1 2 2; 0 5 0 5]; % t = [-1 -1 1 1];
state0 =  1367;
rand('state',state0)
net = newff(input,target,2,{},'traingd');
net.divideFcn = '';

%set max epoh, goal, learning rate, show stp
net.trainParam.epochs        =1;
net.trainParam.goal        = 0.01;
net.performFcn ='sse';
net.trainParam.lr          = 0.01;
net.adaptFcn=' ';

net.trainParam.show        = 100;
net.trainparam.mc          = 0.95;
net.layers{1}.transferFcn = 'logsig';
net.layers{2}.transferFcn = 'logsig';

wih     = net.IW{1,1};
wihb= net.b{1,1};
who   = net.LW{2,1};
whob = net.b{2,1};

%Train
net = train(net,input,target); %adapt
y= sim(net,input);
e=target-y;
perf = sse(e)
```

after run, I've found that the y(1) is 0.818483286935909 it is different from manual count which is 0.609299823823181 (i recheck by calculate ==>

```
for i=1:size(input,2)
hidden(1) = logsig( wih (1)*input(1) + wih(2)*input(2) + wihb(1) );
hidden(2) = logsig( wih (3)*input(1) + wih(4)*input(2) + wihb(2) );
out(i) = logsig( hidden(1)*who(1) + hidden(2)*who(2) + whob(1) );end  )
```

my questions is: 1) is the original MATLAB is using traingd? 2) what does really net = train(net,input,target); y= sim(net,input); do where manual calculation resulted 0.609299823823181 rather than 0.818483286935909 using train and sim.

3) what are the different that my crude forward propagation in C compared to matlab code as above?

please,please help me.

matlab | neural-network | backpropagation

edited Apr 30 at 11:48
Grzegorz Adam Kowalski
1,973 3 15

asked Nov 14 '11 at 5:56
Ummu Rifqi
1 1

add comment

2 Answers

1) I believe that Matlabs "train" command uses batch learning, not online. Perhaps you should look into the "adapt" function in Matlab for online training, don't know if it's any good though. Are you asking if train() and traingd() are actually the same methods or are you asking if train also use gradient-descent?

2) Matlab help says "Typically one epoch of training is defined as a single presentation of all input vectors to the network. The network is then updated according to the results of all those presentations."

I guess this means train will backpropagate and "train" the network one time, and then you simulate a answer based on this trained network.

3) Is the C code listed here all the code in your program? If so, i guess the difference is that Matlab updates the weights once and then feed-forward, while your C code only seem to feed-forward?? Or have i missed something/you left something out?

Hope i have understood all your questions correctly, they were a bit unclear at times, please comment if i got something wrong..

edited Nov 14 '11 at 13:05

answered Nov 14 '11 at 12:51
Niclas
819 2 8

thank Niclas, I have seen adapt function, I guess the newff function initialize different weight (during newff init and reinit activation function) 2) I also believe traingd using batch training. but when I checked for i=1:size(input,2) hidden(1) = logsig(wih (1)*input(1) + wih(2)*input(2) + wihb(1)); hidden(2) = logsig(wih (3)*input(1) + wih(4)*input(2) + wihb(2)); out(i) = logsig(hidden(1)*who(1) + hidden(2)*who(2) + whob(1));end) 3) the C code just as follows – Ummu Rifqi Dec 6 '11 at 8:09

add comment

thank Niclas, I have seen adapt function, I guess the newff function initialize different weight (during newff init and reinit activation function)

2) I also believe traingd using batch training. but when I checked the output:

```
for i=1:size(input,2)
    hidden(1) = logsig( wih (1)*input(1) + wih(2)*input(2) + wihb(1) );
    hidden(2) = logsig( wih (3)*input(1) + wih(4)*input(2) + wihb(2) );
    out(i) = logsig( hidden(1)*who(1) + hidden(2)*who(2) + whob(1) );
end
```

3) the C code just as follows:

```
void forwardPropagate(double *Input)
{
  int i,j,k;
  double sumIH=0.0,sumHO=0.0;

for(j=0; j< numHid; j++)
{
    for(i=0; i<numInput; i++) //numInput+1
    {     //
        sumIH+=Input[i] * wtIH[j][i];
    }
    sumIH+=(1.0 * wtIH[j][numInput]);
    Hidden[j]=sigmoid(sumIH);
}


    for(k = 0 ; k< numOutput ; k++ )
{
    for(j =0 ; j <numHid ; j++ ) //numHid+1
    {
        sumHO+=Hidden[j] * wtHO[k][j];
    }
    sumHO+=(1.0 * wtHO[k][numHid]);
    Output[k]=sigmoid(sumHO);
}
}


void backPropagate (double *target)
{
  int j,k;
  double sumOutErr, desired[numOutput];

  for(k = 0 ; k<numOutput ; k++ )
  {
```

thanks.

edited Dec 6 '11 at 8:26
Chadwick
6,272 4 27 53

answered Dec 6 '11 at 8:17
Ummu Rifqi
1 1

add comment

Not the answer you're looking for? Browse other questions tagged matlab neural-network backpropagation **or ask your own question.**