neural network-back propagation, error in training

after reading some articles about neural network(back-propagation) i try to write a simple neural network by myself.

ive decided XOR neural-network, my problem is when i am trying to train the network, if i use only one example to train the network,lets say 1,1,0(as input1,input2,targetOutput). after 500 trains +- the network answer 0.05. but if im trying more then one example (lets say 2 different or all the 4 possibilities) the network aims to 0.5 as output :(i searched in google for my mistakes with no results :S ill try to give as much details as i can to help find what wrong:

-ive tried networks with 2,2,1 and 2,4,1 (inputlayer,hiddenlayer,outputlayer).

-the output for every neural defined by:

```
double input = 0.0;
        for (int n = 0; n < layers[i].Count; n++)
            input += layers[i][n].Output * weights[n];
```

while 'i' is the current layer and weight are all the weights from the previous layer.

-the last layer(output layer) error is defined by:

```
value*(1-value)*(targetvalue-value);
```

while 'value' is the neural output and 'targetvalue' is the target output for the current neural.

-the error for the others neurals define by:

```
foreach neural in the nextlayer
           sum+=neural.value*currentneural.weights[neural];
```

-all the weights in the network are adapt by this formula(the weight from neural -> neural 2)

```
weight+=LearnRate*neural.myvalue*neural2.error;
```

while LearnRate is the nework learning rate(defined 0.25 at my network). -the biasweight for each neural is defined by:

```
bias+=LearnRate*neural.myerror*neural.Bias;
```

bias is const value=1.

that pretty much all i can detail, as i said the output aim to be 0.5 with different training examples :(

thank you very very much for your help ^_^.

c# neural-network backpropagation

asked Sep 20 '11 at 0:55
aliyaho
26 2

2 Have you written your own implementation or is it using a framework? – Chris S Sep 20 '11 at 15:04

That's really funny that he couldn't answer his question after it was protected :) – PythEch Sep 21 '11 at 20:15

add comment

1 Answer

It is difficult to tell where the error is without seeing the complete code. One thing you should carefully check is that your calculation of the local error gradient for each unit matches the activation function you are using on that layer. Have a look here for the general formula: http://www.learnartificialneuralnetworks.com/backpropagation.html .

For instance, the calculation you do for the output layer assumes that you are using a logistic sigmoid activation function but you don't mention that in the code above so it looks like you are using a linear activation function instead.

In principle a 2-2-1 network should be enough to learn XOR although the training will sometime get trapped into a local minimum without being able to converge to the correct state. So it is important not to draw conclusion about the performance of your algorithm from a single training session. Note that simple backprog is bound to be slow, there are faster and more robust solutions like Rprop for instance.

There are books on the subject which provide detailed step-by-step calculation for a simple network (e.g. 'A.I.: A guide to intelligent systems' by Negnevitsky), this could help you debug your algorithm. An alternative would be to use an existing framework (e.g. Encog, FANN, Matlab) set up the exact same topology and initial weights and compare the calculation with your own implementation.

answered Sep 22 '11 at 17:30
Bob Roberts
160 5

add comment

protected by Community ♦ Sep 21 '11 at 10:18

Thank you for your interest in this question. Because it has attracted low-quality answers, posting an answer now requires 10 reputation on this site.

Would you like to answer one of these unanswered questions instead?

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