

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour ×

## Target values for neural network

**CAREERS 2.0**  
by stackoverflow



+



Have projects on BitBucket?  
Import them easily to your profile

I've been using MATLAB's neural network toolbox to generate good for character recognition. I wanted to develop my own to compare against and also try and use different methods for weight updates. I've got the network to return only values between 0 and 1. I was wondering how I can adapt this so that the network returns say 1, 2, 3, 4; numbers corresponding to rows in my data matrix, where each row is a different letter.

[matlab](#) [neural-network](#) [targets](#)

asked May 9 '13 at 4:37



[roldy](#)

45 1 7

[add comment](#)

### 1 Answer

Usually when talking about character multiple character recognition it is a good approach to instead of using only 1 output to use an output for each of the characters you have. So if you have 10 characters (say digits from 0-9) you can have N inputs (1 per pixel) and 10 outputs (1 per character). Like that you can not only get a result, but analyze how good is your NN tuned and how good is your result precision.

So if you input an image and the result is  $[0.2 \ 0.1 \ 0.98 \ 0.3 \ 0.12 \ 0.2 \ 0.1 \ 0.4 \ 0.1 \ 0.2]$  you know that the NN is pretty well tuned and the result is very accurate, but instead if you get  $[0.4 \ 0.1 \ 0.6 \ 0.54 \ 0.5 \ 0.3 \ 0.5 \ 0.3 \ 0.57 \ 0.2]$  you know that even if the number seems to be the same as before, the fiability of the result is much lower.

Having said that you can just train you NN to give you a nu,ber from 0 to 1 for each case, if you still want to do it with 1 output.

```
0 ->0
1->0.1
2->0.2
...
```

Still I can assure you that your result are going to be worse like this. Use 1 output from 0 to 1 for each digit!

answered May 9 '13 at 8:03



[Ander Biguri](#)

1,913 9 29

if my input vector is as follows, letter\_X =  $[0 \ 1 \ 0; \ 1 \ 0 \ 1; \ 0 \ 1 \ 0]$ ; letter\_C =  $[0 \ 0 \ 0; \ 0 \ 1 \ 1; \ 0 \ 0 \ 0]$ ; input =  $[letter\_X(:), letter\_C(:)]$ ; what should my target vector look like? It shouldn't be the same as input. — [roldy](#) May 10 '13 at 1:13

@roldy First choose how many characters do you want to recognise. So if you want to recongines 4 different characters (e.g  $[X, Y, C, M]$ ) your desired output for letter\_X is  $[1, 0, 0, 0]$  and for letter\_C is  $[0, 0, 1, 0]$ . — [Ander Biguri](#) May 10 '13 at 12:53

Ok, thank you. That makes things much more clearer. — [roldy](#) May 10 '13 at 20:06

@roldy if it helped you dont forget to mark the aswer as correct! — [Ander Biguri](#) May 11 '13 at 0:24

7/1/2014

matlab - Target values for neural network - Stack Overflow

I forgot to ask in the original question how I would go about using this trained network to recognize a selected letter? Would I just multiply the weights with the selected input? That doesn't seem right. – [roldy](#) May 11 '13 at 7:50

---

show 4 more comments

---

Not the answer you're looking for? Browse other questions tagged [matlab](#)

[neural-network](#) [targets](#) or [ask your own question](#).