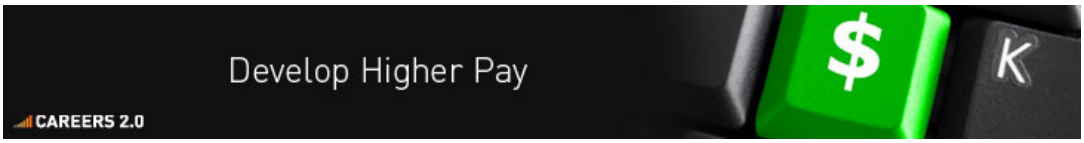
Which multiplication and addition factor to use when doing adaptive learning rate in neural networks?



I am new to neural networks and, to get grip on the matter, I have implemented a basic feed-forward MLP which I currently train through back-propagation. I am aware that there are more sophisticated and better ways to do that, but in Introduction to Machine Learning they suggest that with one or two tricks, basic gradient descent can be effective for learning from real world data. One of the tricks is *adaptive learning rate*.

The idea is to increase the learning rate by a constant value **a** when the error gets smaller, and decrease it by a fraction **b** of the learning rate when the error gets larger. So basically the learning rate change is determined by:

```
+(a)
```

if we're learning in the right direction, and

```
-(b * <learning rate>)
```

if we're ruining our learning. However, on the above book there's no advice on how to set these parameters. I wouldn't expect a precise suggestion since parameter tuning is a whole topic on its own, but just a hint at least on their order of magnitude. Any ideas?

Thank you,
Tunnuz

neural-network backpropagation

asked Sep 8 '11 at 8:22

tunnuz
5,894 12 50 96

add comment

1 Answer

I haven't looked at neural networks for the longest time (10 years+) but after I saw your question I thought I would have a quick scout about. I kept seeing the same figures all over the internet in relation to increase(***a***) and decrease(***b***) factor (**1.2** & **0.5** respectively).

I have managed to track these values down to Martin Riedmiller and Heinrich Braun's RPROP algorithm (1992). Riedmiller and Braun are quite specific about sensible parameters to choose.

See: RPROP: A Fast Adaptive Learning Algorithm

I hope this helps.

answered Sep 12 '11 at 22:43

Mark McLaren
8,146 2 14 46

I would like to accept this answer, but these look like very large numbers considering that they are used to expand or shrink the learning rate value which should be in [0.0,1.0]. Are you sure that they don't have a different meaning in RPROP? – tunnuz Sep 13 '11 at 13:06

The values mentioned are specifically for the increase/decrease factors (used to correct you when you start going off course). If you look through the PDF (search does't work for me, could be a LaTex thing - and the

page order seems to be backwards!) I think it mentions a learning rate (epsilon and delta zero - depending on algorithm). In one table it gives it a value of 0.05. It also mentions other potential algorithms (BP, SuperSAB, QuickProp) and results given when other values/algorithms are used. – Mark McLaren Sep 13 '11 at 14:18

See also: stackoverflow.com/questions/2865057/… – Mark McLaren Sep 13 '11 at 15:02

I have been trying to locate a simple RPROP explanation; simple explanations of neural networks seem a little hard to come by (perhaps someone should write a book - maybe one that avoids scary calculus notation). Anyway, I managed to locate some reasonably clear pseudo code at: heatonresearch.com/wiki/Resilient_Propagation – Mark McLaren Sep 13 '11 at 17:06

Ok. RPROP is very good and I should give it a look as soon as possible however: I do believe that the a, b parameters I am referring to have a significantly different meaning than eta+ and eta- in RPROP. In my understanding eta+ and eta- are used *directly* as learning rates, while a and b are used to *influence* the learning rate based on the last error received, but the learning rate is then used to compute the weight update. That's why *I will not accept this answer*, nevertheless, I will reward you with the bounty since your effort to answer this question is admirable. – tunnuz Sep 14 '11 at 7:06

show **4** more comments