Why LSTM

张东旭

RNN

- $s(t) = f(Ws(t-1) + x(t))$

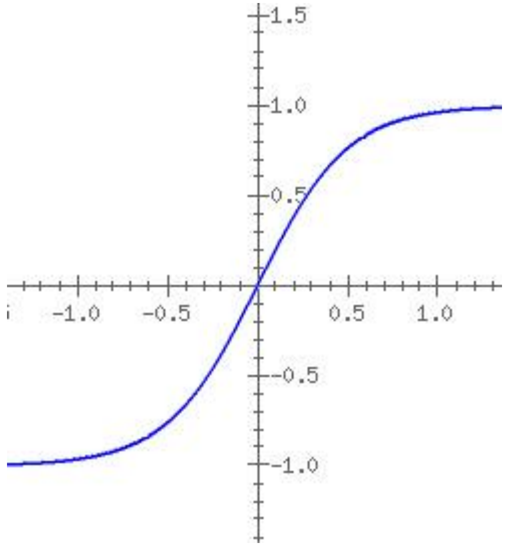
  f()--Nonlinearity function; s(t) hidden state(one dimemsion);

  x(t) input(one dimension)

If W>1/f’(0), then there will be two attractors where $s = f(ws)$, $(s \neq 0)$

Store is accomplished by keeping a large input x.

Larger w, more robustness against noise.



BPTT

- $s(t) = f(Wf(Wf(Wf(Ws(t-3) + x(t-3)) + x(t-2)) + x(t-1)) + x(t))$

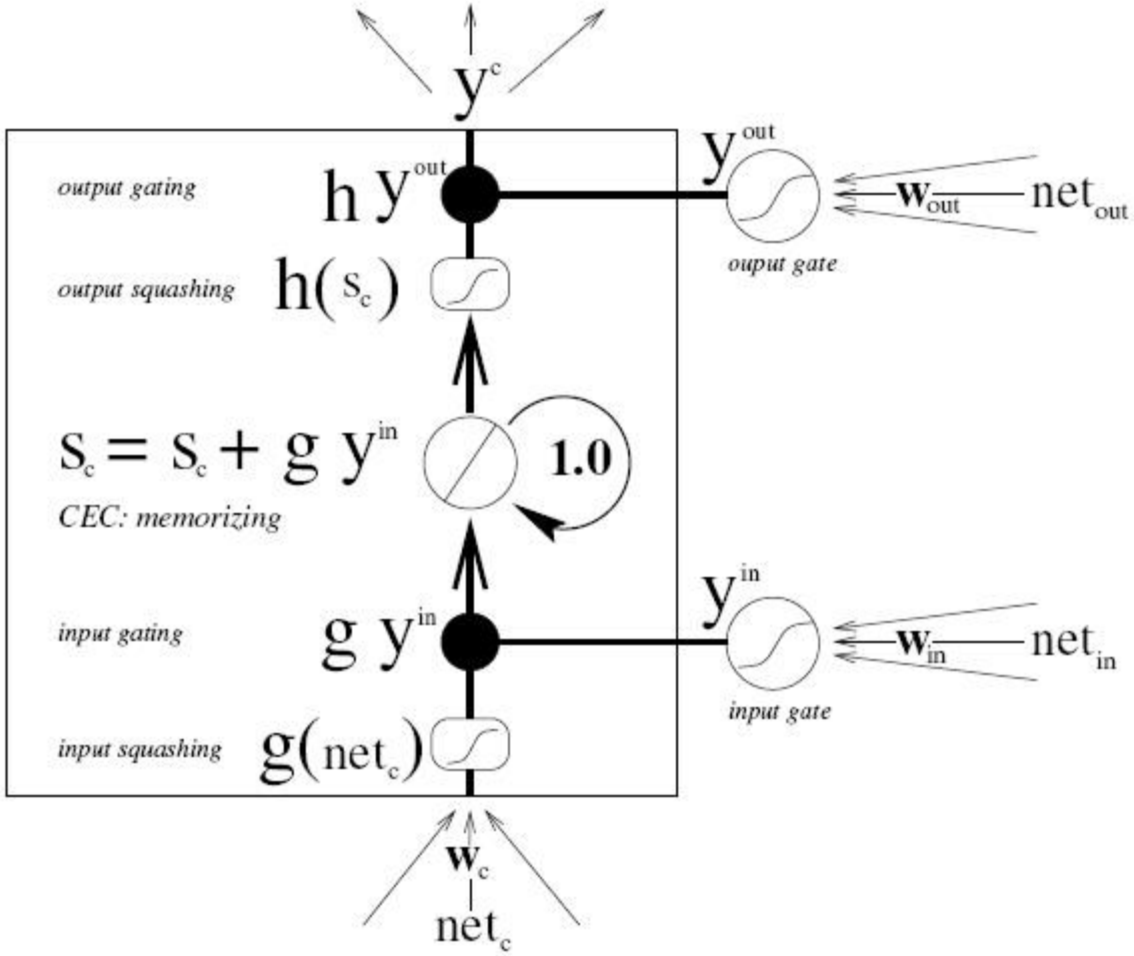
$$\frac{\partial C_t}{\partial x(t-3)} = \frac{\partial C}{\partial s(t-1)} \frac{\partial s(t-1)}{\partial s(t-2)} \frac{\partial s(t-2)}{\partial x(t-3)}$$

$$\frac{\partial C_t}{\partial W} = \sum_{i=1}^{2} \frac{\partial C}{\partial S(t-1)} \frac{\partial s(t-1)}{\partial s(t-i)} \frac{\partial s(t-i)}{\partial W}$$

LSTM

- The LSTM algorithm overcomes this problem by enforceing non-decaying error flow “back into time”. (CECs, constant error carrousels)

$$s(t) = s(t-1) + y^{in} \cdot x(t)$$



- Gers F. Long short-term memory in recurrent neural networks[J]. Lausanne, EPFL, 2001, 2366.
- Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. Neural Networks, IEEE Transactions on, 1994, 5(2): 157-166.

THANK YOU