Recurrent Neural Networks

Topic 7

Note: lecture notes by Bob Keller (Harvey Mudd College, CA) are used

Main idea: include feedback from neuron output to input into NN models

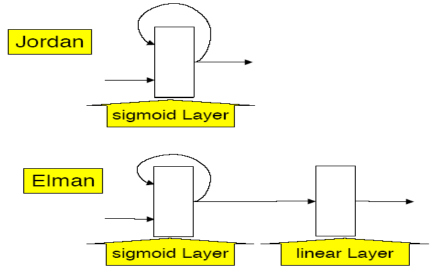
- A recurrent network is one in which there is feedback from a neuron's output to its input.
- Various models exist:

  Jordan Network (feedback from net output to input)

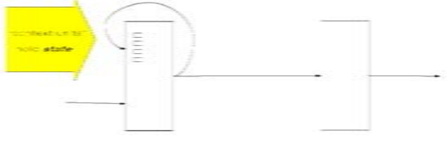
  Elman Network ("partially recurrent": feedback from internal state output to input)

  Hopfield Network

Jordan vs. Elman Networks



Elman networks



- References:

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14:179-211.

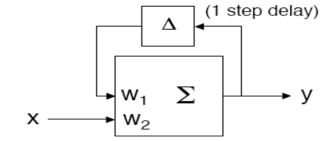
Neural Networks: Automata and Formal Models of Computation, Mikel L. Forcada, http://www.dlsi.ua.es/~mlf/nnafmc/pbook/pbook.html

How to train Elman networks?

- One way:
- Initialize the state values to nominal.
- Repeat
  - Simulate one step of the network.
  - Compute the actual output.
  - Backpropagate the error.
  - Adjust the weights.
  - Compute the next state.
- Until the error is sufficiently low.

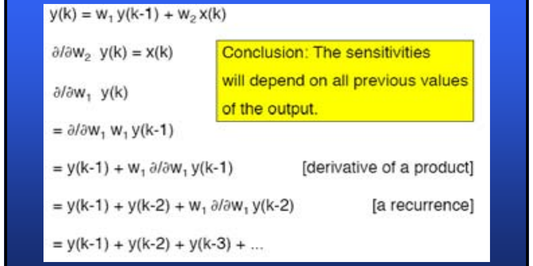
Training feedback weights

- Elman Schematic



$y(k) = w_1\, y(k-1) + w_2\, x(k)$

Training feedback weights



$y(k) = w_1\, y(k-1) + w_2\, x(k)$

$\partial/\partial w_2\;\; y(k) = x(k)$

$\partial/\partial w_1\;\; y(k)$

$= \partial/\partial w_1\, w_1\, y(k-1)$

$= y(k-1) + w_1\, \partial/\partial w_1\, y(k-1)$ [derivative of a product]

$= y(k-1) + y(k-2) + w_1\, \partial/\partial w_1\, y(k-2)$ [a recurrence]

$= y(k-1) + y(k-2) + y(k-3) + \ldots$

Conclusion: The sensitivities will depend on all previous values of the output.

Demos of Elman Networks

Two demos:

- Matlab appelm1
- NAS demo 11.2

Other possible ways to train

- BPPT (Backpropagation Through Time) would be another way:

  unroll the network some large number of levels,

  backpropagate,

  average the weight changes over the **unrolled** stages to get a single set of weight changes