

# Recurrent Neural Networks

Marc de Kamps

Biosystems Group  
University of Leeds

April 14, 2008

# Outline

- 1 Other recurrent networks
  - Elman Networks

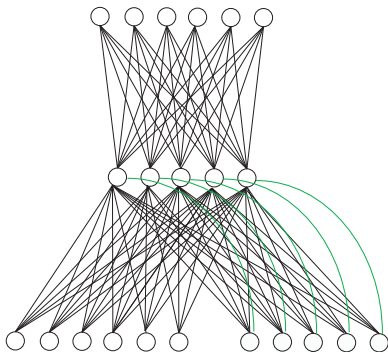
# Outline

- 1 Other recurrent networks
  - Elman Networks

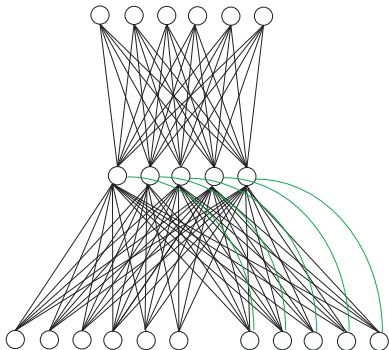
# Using Feedforward Networks Recurrently

Finding Structure in Time - Elman (1990)

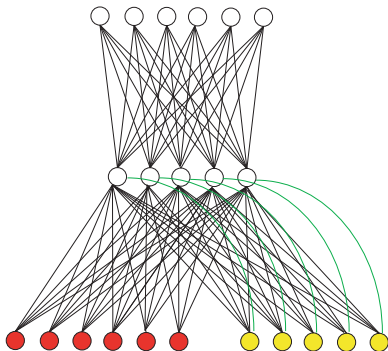
- Based on Jordan (1986)
- Use a normal feedforward network
- Use the hidden layer of previous pattern as part of input for the next pattern



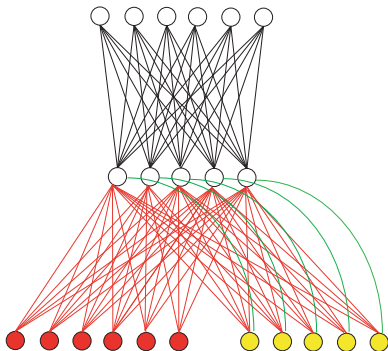
# Updating Elman Networks



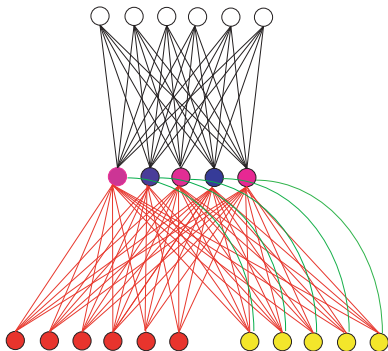
# Updating Elman Networks



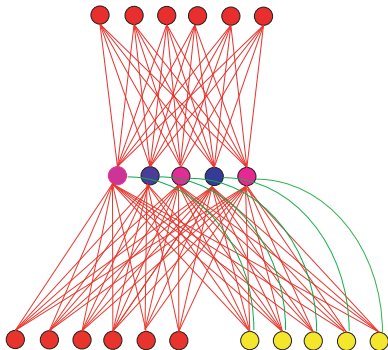
# Updating Elman Networks



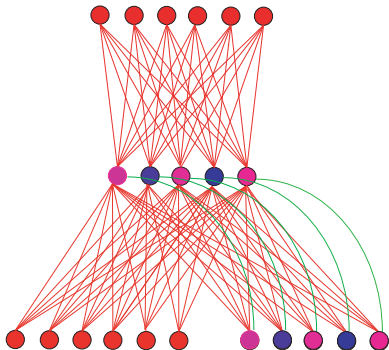
# Updating Elman Networks



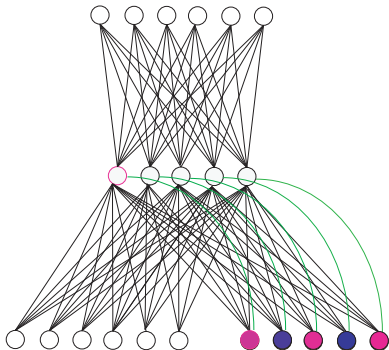
# Updating Elman Networks



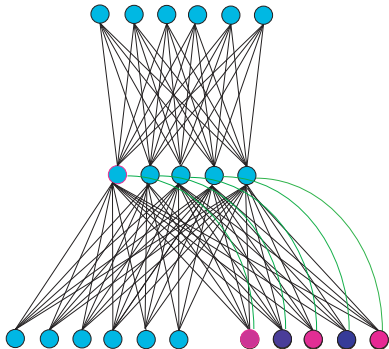
# Updating Elman Networks



# Updating Elman Networks



# Updating Elman Networks

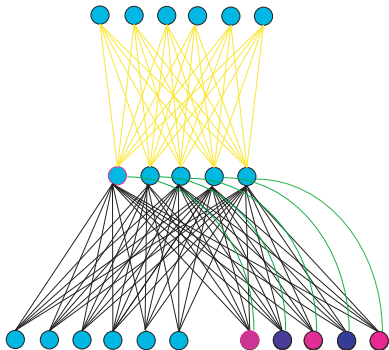


# Training Elman Networks

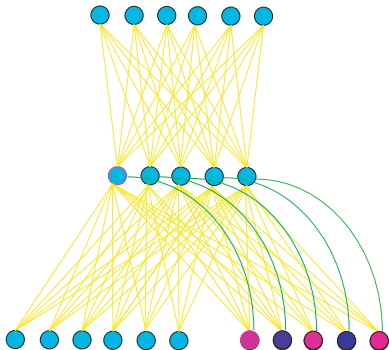
- At any time in the input two patterns are present:
- **Input pattern** and **Context**
- Together they produce observed output  $\vec{o}$
- Compare observed output with desired output  $\vec{t}$

**Use backpropagation**

# Training Elman Networks



# Training Elman Networks



# Simple Experiment: XOR

- 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, ...
- Every third element is the XOR of previous two
- Task network: predict next element in series:
  - in: 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, ...
  - out: 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1

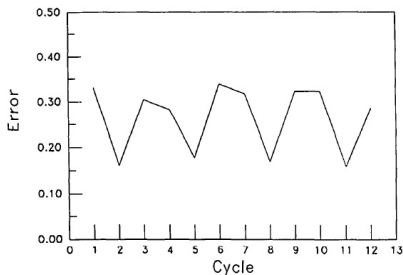


Figure 3. Graph of root mean squared error over 12 consecutive inputs in sequential XOR task. Data points are averaged over 1200 trials.

# More Complex Example

## Structure in Letter Sequences

- Random sequence of 3 consonants {b, d, g} (1000 long)
- Vowels are added according to the following rules:
  - $b \rightarrow ba$
  - $d \rightarrow dii$
  - $g \rightarrow guuu$
- So a typical sequence could read: dbgdbddg, which would expand to diibaguubadiidiiguuu, etc.

# Structure in Letter Sequences

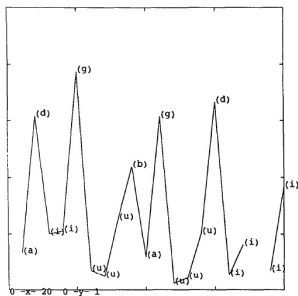
- Letters represented as 6 bits
- Network: 6 input, 6 output, 20 hidden, 20 context nodes

	consonant	vowel	interrupted	high	back	voiced
b	1	0	1	0	0	1
d	1	0	1	1	0	1
g	1	0	1	0	1	1
a	0	1	0	0	1	1
i	0	1	0	1	0	1
u	0	1	0	1	1	1

- Network trained on 200 passes, tested on a completely new sequence

# Extracting regularities

- Note: that prediction errors are high for consonants
- Low for vowels
- It knows how many vowels to expect



**Figure 4.** Graph of root mean squared error in letter prediction task. Labels indicate the correct output prediction at each point in time. Error is computed over the entire output vector.

# Extracting Regularities

- Error on bit 1 is low:
- It knows not what consonant is coming, but it knows when it is coming
- It even knows after how many trials it is coming!
- but it can not predict high or low very well

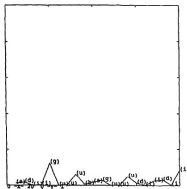


Figure 3 (a). Graph of root mean squared error in letter prediction task. Error is computed on bit 1, representing the feature CONSONANTAL.

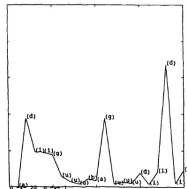


Figure 3 (b). Graph of root mean squared error in letter prediction task. Error is computed on bit 4, representing the feature HIGH.