Learning, performing and remembering serially ordered actions

Aims

- Explain how feedback and recurrent architectures can be used to learn a chain of associations.
- Understand the general limitations of chaining models, and the particular strengths and weaknesses of Jordan/Elman, and Hopfield approaches.
- Describe the competitive queuing architecture, how it is related to the general issue of action selection, and how it is used to learn action sequences.
- Understand the strengths and weaknesses of competitive queuing.

Reading

- Elman J L (1990) `Finding structure in time', Cognitive Science 14 179-211.
- G Houghton & T Hartley (1996) 'Parallel Models of Serial Behaviour: Lashley Revisited' PSYCHE, 2(25). http://psyche.cs.monash.edu.au/v2/psyche-2-25-houghton.html
- Houghton G & Tipper S P (1996) `Inhibitory mechanisms of neural and cognitive control: application to selective attention and sequential action' Brain and Cognition 30 20-43.
- Burgess N & Hitch G J (1999) Memory for Serial Order: A Network Model of the Phonological Loop and its Timing, Psychological Review, 106 551-581.
- D Bullock (2004) 'Adaptive neural models of queuing and timing in fluent action' Trends Cog. Sci. 426-33

Importance of Serial Order in behaviour

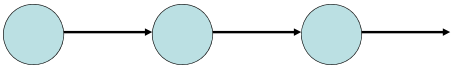
- Motor Control
  - E.g., a fish needs to generate a sequence of muscle contractions to swim (+ heartbeat etc).
- Speech and Language
  - E.g., a sequence of speech sounds must be produced to say a word.
- Executive Function
  - E.g., a sequence of actions must be produced to make a cup of tea.

Two ways in which serial learning might be implemented in neural networks

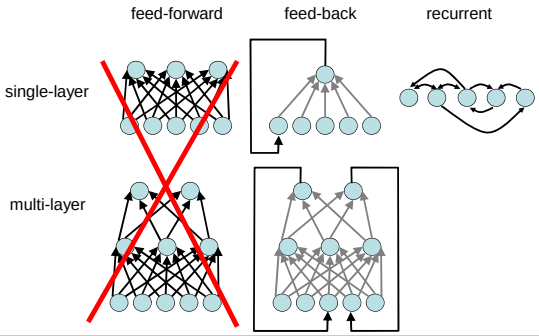
1. Recurrent/feedback connections, and/or modified serial learning algorithms. Extensions of architectures covered in previous lectures. “Associative Chaining”.
2. Special mechanisms for the representation of time/serial position and for the serial selection of actions. “Competitive Queuing”.

Part 1 “Associative Chaining”.

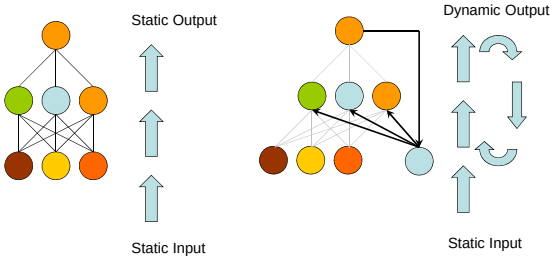
- Recurrent/feedback connections, and/or modified serial learning algorithms. Extensions of architectures covered in previous lectures, e.g., to include asymmetric recurrent connections.



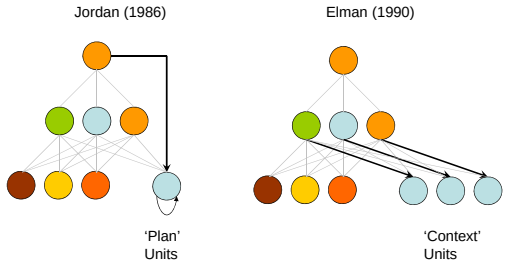
Standard feedforward networks can’t produce serial output.



With added feedback connections, serial output becomes possible.



Partially recurrent (feedback) network architectures permit serial *learning* using standard supervised learning rules (e.g., error back-propagation).



Advantages of Jordan/Elman approach

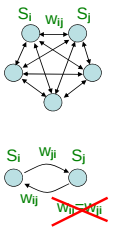
- Can learn any sequence in principle (but might require huge numbers of hidden units and huge numbers of training epochs in practice).
- Simple extension of feedforward architecture. Doesn't require much additional machinery.

Shortcomings of Jordan/Elman nets.

- Backpropagation algorithm is not biologically plausible (might be resolved by using more plausible algorithms e.g., reinforcement learning).
- Requires many exposures to a sequence to learn it (so can't explain single trial learning).

Serial Learning in Hopfield-Type networks

- Hopfield-type networks comprise a single layer of fully interconnected units (recurrent).
- Normally used for static pattern completion.
- Can also be trained to produce a sequence of patterns by abandoning the assumption that weights are symmetrical and modifying the learning rule to associate one pattern to the next.



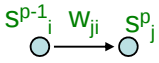
Serial Learning in Hopfield-Type Networks

When learning symmetric weights (Hebbian rule for standard Hopfield network for static pattern completion):

$$W_{ji} \rightarrow W_{ji} + \varepsilon S^p_i S^p_j$$

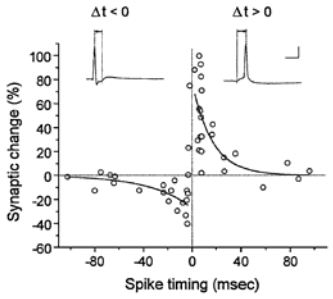
To learn asymmetric weights (dynamic serial memory). 'Hebbian' rule including increase for pre- *then* post- synaptic activation (plausible over short timescales, e.g. < 100 ms):

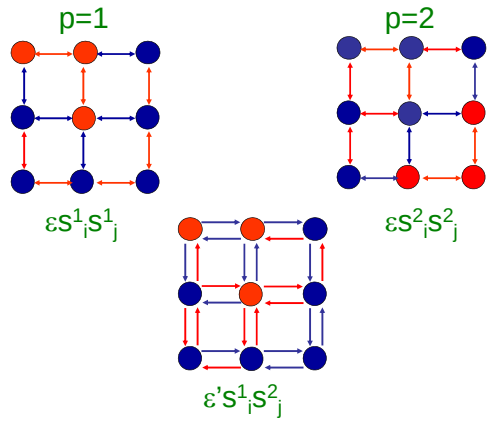
$$W_{ji} \rightarrow W_{ji} + \underbrace{\varepsilon S^p_i S^p_j}_{\text{Maintains stability of each pattern}} + \underbrace{\varepsilon' S^{p-1}_i S^p_j}_{\text{Drives transition from each pattern p-1 to its successor p}}$$



LTP (vs LTD) depends on pre-synaptic activity *preceeding* post-synaptic activity

Bi and Poo (1998)





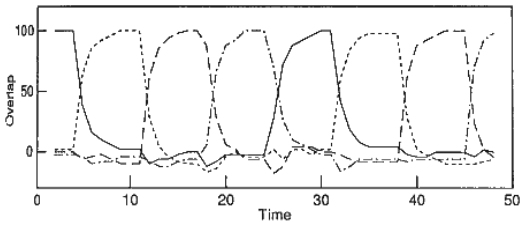


FIGURE 3.10 Example of sequence generation. The curves show the overlap of the state S_i at time t with each of the embedded patterns ξ^1_i, ξ^2_i, ξ^3_i, ξ^4_i. The overlaps were calculated using $\sum_i S_i \xi^\mu_i$. The parameters used were $N = 100$, $p = 4$, $\lambda = 2$, $\tau = 8$ using the step function kernel (3.59).

Advantages of Hopfield Serial Learning

- Simple modification to the static pattern associator.
- Simple 'Hebbian' learning rule.
- Can do fixed association of small number of elements in order, learning of stereotypical action sequences, e.g. 'central pattern generators' (controlling muscle contractions in walking, swimming, heartbeat etc., Kleinfeld & Sompolinski, 1988).
- Providing neural "clocks" or "counters" (Amit 1988),

DJ. Amit (1988) Neural networks counting chimes Proc. Nat. Acad. Sci. USA 85 2141-2145

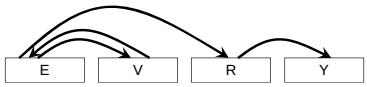
Kleinfeld D, Sompolinsky H. (1988) Associative neural network model for the generation of temporal patterns. Theory and application to central pattern generators. J. Biophys 54(6):1039–1051

Shortcomings of Hopfield Serial Learning

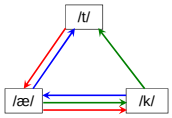
- Patterns must be uncorrelated.
- Number of patterns in the stored sequence must be much less than the number of units in the network.

Fundamental Problems for Chaining Models

(Lashley, 1951. these apply to both Hopfield and Jordan/Elman networks)



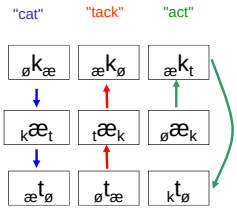
Conflict within a chain. E.g., chain required to type "EVERY"



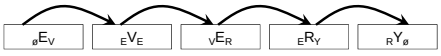
Conflict between chains. E.g.,Chains representing the spoken words "cat", "tack" and "act" ?

Context-specific coding

(Workaround: Wickelgren, 1969)



Different representations of the same action occurring in different contexts.
This is similar to the way Jordan/Elman networks eventually learn to solve the problem using hidden units.
But you need a lot of different units, and separate learning in each (labour intensive) .



Chaining: the Problem of Serial Order

(Lashley, 1951. Applies to both Hopfield and Jordan/Elman networks)

- Difficulty in representing "conflicting" sequences of the same items/actions. E.g., to say "cat", "act", or "tack".
- Difficulty in representing sequences containing repeated items/actions. E.g., to type "EVERY".
- Doesn't fit well with error data from e.g., speech and language, short-term memory in which order interchanges are the most frequent (Henson, 1996). E.g., "3,1,4,2,0" → "3,4,1,2,0"

Part 2 "Competitive Queuing".

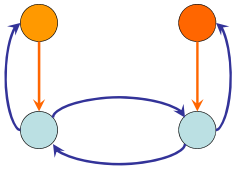
A special mechanisms for the representation of time/serial position and for the serial selection of actions (Grossberg 1978; Houghton, 1990).

Insight:

- Perception is parallel.
- The brain can process information in parallel.
- But action is constrained to be serial (e.g., "I've only got one pair of hands!").

Response Competition and Action Selection

"Eat Cake" "Drink Tea"



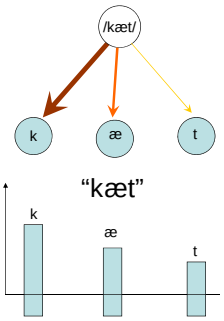
Competitive Filter

Competitive Queuing

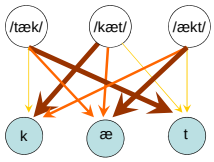
(Grossberg, 1978; Houghton, 1990)

- Establish an activation gradient over the units representing the actions to be sequenced.
- Select the most active unit, and perform the associated action.
- Suppress the most active unit.
- Repeat until all the actions have been performed.

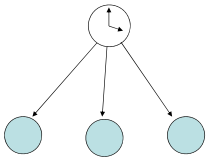
Basic Mechanism (ordered connection strengths)



Storing Multiple Sequences

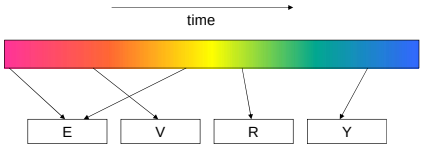


Dynamic Control Signal
(can deal with repeats)

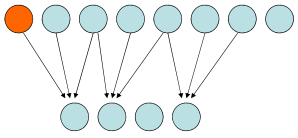


Each response is associated with a different state of a postulated neural clock or “context signal”

Dealing with Repeated Items

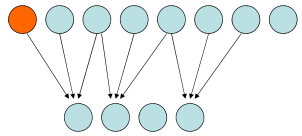


Learning a sequence



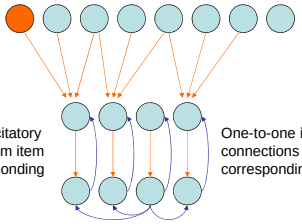
Perceiving an item activates an associated item unit. Links from context units to item units are strengthened using a Hebbian learning rule.
•Unsupervised
•Single trial

Retrieving a sequence



•When the context signal is “replayed” the item units are re-activated in the order they were perceived.

Full implementation, including competitive filter to select then inhibit one item-at-a-time.



One-to-one excitatory connections from item units to corresponding filter units

One-to-one inhibitory feedback connections from filter units to corresponding item units

Filter units have mutual inhibitory connections (lateral inhibition)

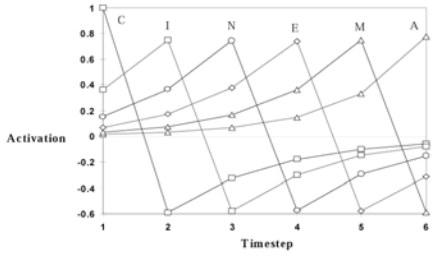
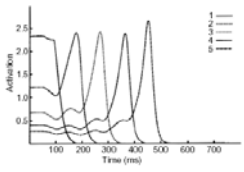


Figure 1. CQ activation dynamics of nodes representing letters during production of the word “CINEMA”. The activation level of each letter is shown at each timestep during production of the word. The trace for each letter is labelled at the point where it wins the competition for output.

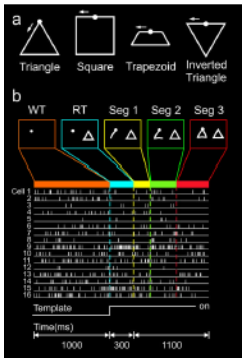
Advantages of Competitive Queuing

- Can do single trial learning
- Accounts well for error data e.g., paired transpositions.
- Can manage different sequences of the same items.
- Dynamic control signals allow representation of repeated items.
- Competitive filter fits well with more general requirement for action selection.

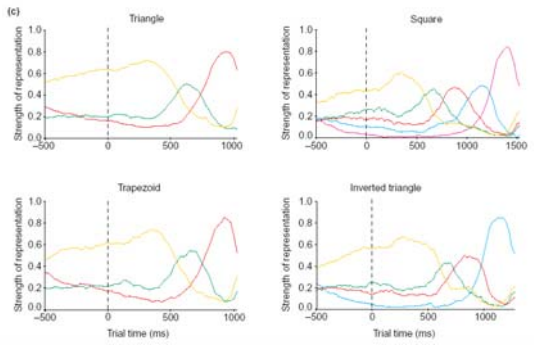
Neurophsyiological Evidence for Competitive Queuing



Simulation of CQ of elements in a 5-part line drawing, see
Bullock (2004) Trends Cog. Sci. 426-33



Recording in primate prefrontal cortex
Averbeck et al. (2002) PNAS 99 13172-7



Strength of representation: fraction of trials in which pattern of activity classified as given movement
Averbeck et al. (2002) PNAS 99 13172-7

Summary I

- Feedforward networks can't produce serial output.
- By adding feedback connections, sequences can be learned using standard supervised learning algorithms.
- Fully recurrent networks can be trained to learn sequences using a modified learning rule.
- Both these solutions implement a form of Associative Chaining.

Summary 2

- An alternative approach is called Competitive Queuing
- Local representations of the actions to be sequenced are associated with a dynamic control signal, using a Hebbian rule for 'One-shot learning'.
- A competitive filter allows only one action at a time. After each response the selected action is inhibited.
- Neurophysiological evidence exists for CQ in primate prefrontal cortex