Lecture 3

Neural Modeling of Episodic Memory: Encoding, Retrieval, and Forgetting
Contents

- Previous ARTs
- Introduction of Episodic Memory and EM ART
- Issues and Challenges
- The EM Model
- Event Encoding and Retrieval
- Episode Learning and Retrieval
- Forgetting in Episodic Memory
- Experiments
- Summary
Adaptive Resonance Associative Map

cf: ARTMAP (Carpenter et al. 92)

ARAM Dynamics

Step 2: Code competition & selection

\[ T_j = \max \{ T_j \}, \quad y_j = \begin{cases} 1 & \text{if } j = J \\ 0 & \text{otherwise} \end{cases} \]

Step 1: Bottom-up propagation

\[ T_j = \gamma \frac{|A \cdot W_j^a|}{\beta + |W_j^a|} + (1 - \gamma) \frac{|B \cdot W_j^b|}{\beta + |W_j^b|} \]
Step 4: Template learning

\[ w_j^{a(new)} = (1 - \beta^a)w_j^{a(old)} + \beta^a(A \wedge w_j^{a(old)}) \]
\[ w_j^{b(new)} = (1 - \beta^b)w_j^{b(old)} + \beta^b(B \wedge w_j^{b(old)}) \]

Step 3: Top down priming

\[ m_j^a = \frac{|A \wedge W_j^a|}{|A|} \]
\[ m_j^b = \frac{|B \wedge W_j^b|}{|B|} \]
ARAM for Personalized Information Management
Predictive ART for Gene Expression Data Analysis

A predictive approach for gene expression data analysis is presented, involving the following steps:

1. Expression Database
2. Dimensionality Reduction
3. Fisher feature selection (biology)
4. Entropy-based discretization
5. Predictive Modelling
6. Self-organizing Neural Networks
7. Rule Extraction
8. IF-THEN symbolic rules

Ah-Hwee Tan & Hong Pan, “Predictive Neural Networks for Gene Expression Data Analysis,” Neural Networks 18 (2005) 297-306
A generalized multi-channel ART architecture

Capable of supporting many distinct learning paradigms as well as symbolic knowledge extraction and integration
Fusion Architecture for Learning & Cognition (FALCON)

- Self-organizing NN for learning cognitive nodes across multi-modal pattern fields
- Compatible with rule-based representation
Temporal Difference - FALCON

- Use *temporal difference learning rule* to estimate future value of performing an action in a state
- Useful for situations without immediate rewards
- Fast action searching through direct code access

TD-FALCON with Direct Code Access

1. Initialize the FALCON network
2. Sense the environment and formulate a state representation $s$.
3. Following an action selection policy
   - If exploring, take a random action.
   - If exploiting, present the state vector to TD-FALCON to identify action $a$ with $\max Q(s,a)$ for situation $s$.
4. Perform action $a$, observe next state $s'$, and receive a reward $r$.
5. Estimate the revised value function $Q(s,a)$.
6. Present the state, action, and reward (Q-value) vectors to TD-FALCON for learning.
7. Update the current state by $s = s'$.
8. Repeat from Step 2 until $s$ is a terminal state.

Direct Code Access for Action Selection

Step 1: Code Activation

Step 2: Code competition

$T_j = \sum_{k=1}^{3} \gamma^{ck} \frac{|x^{ck} \wedge w_{j}^{ck}|}{\alpha^{ck} + |w_{j}^{ck}|}$

$T_j = \max\{T_j\}, y_j = \begin{cases} 1 & \text{if } j = J \\ 0 & \text{otherwise} \end{cases}$

F2

Step 3: Activity Readout

Activity readout:

$x^{ck(new)} = x^{ck(old)} \wedge w^{ck} \Rightarrow x^{c2(new)} = x^{c2(old)} \wedge w^{c2}$. 

Select an action $a_j$ such that

$x_i^{c2} = \max\{x_i^{c2} : \text{for all } F_1^{c2} \text{ node } i\}$
Learning Value Function

Step 4: Template learning

\[ w_{j}^{ck\text{(new)}} = (1 - \beta^a) w_{j}^{ck\text{(old)}} + \beta^a (x^c_k \land w_{j}^{ck\text{(old)}}) \]

Step 3: Top down priming

\[ m_{j}^{c1} = \frac{|x_{j}^{c1} \land W_{j}^{c1}|}{|x_{j}^{c1}|} \]

Step 1 & 2: Code Activation and Competition

\[ m_{j}^{c2} = \frac{|x_{j}^{c2} \land W_{j}^{c2}|}{|x_{j}^{c2}|} \]

\[ m_{j}^{c3} = \frac{|x_{j}^{c3} \land W_{j}^{c3}|}{|x_{j}^{c3}|} \]

State | Action | Estimated Q-value

Or, immediate reward using a utility function, e.g.

\[ \frac{1}{1 + rd} \]
Self-Organizing NNs Integrating Domain Knowledge and RL

- **Motivation**
  - Direct insertion of not-easily learned domain knowledge
  - Allows the systems to start operating at a reasonable level of performance
  - Improve learning efficiency and reduces model complexity

- **Challenges**
  - Difficulty in reconciling human-specific symbolic knowledge and specialized knowledge representation of learning models
  - Dilemma between exploitation of inserted knowledge and exploring to discover new knowledge

*Teck-Hou, Teng, Ah-Hwee Tan, & Jacek M. Zurada. Self-Organizing Neural Networks Integrating Domain Knowledge and Reinforcement Learning, IEEE TNNLS, DOI 10.1109/TNNLS.2014.2327636*
Integration of Domain Knowledge and RL

- Direct insertion of domain knowledge
  - Bridges the gap between human-specific knowledge and knowledge representation of learning models

- Use of low state vigilance favors use of domain knowledge with sparse knowledge structure
  - Ensures the effective use of inserted domain knowledge and learned knowledge

- Greedy exploitation of inserted and learned knowledge
  - Addresses the exploitation-exploration dilemma

- Adaptation of reward vigilance for identifying effective action policies
  - Addresses the exploitation-exploration dilemma
Using FALCON Networks to Store, Retrieve, and Adapt Knowledge

- UT2004 is a commercial First-Person Shooter (FPS) computer game that allows embodiments of virtual agents for combats.
- Pogamut is an IDE for rapid development and provides samples. (IDE: Integrated Development Environment)

Creating autonomous adaptive agents in a real-time FPS computer game

Di Wang & Ah-Hwee Tan, IEEE Transaction on Computational Intelligence and AI in Games, in press, DOI: 10.1109/Tciaig.2014.2336702.
A TD FALCON is used for behavior modeling:

1. Running around: Explores randomly in the neighborhood;
2. Collecting item: Runs to particular locations to collect useful items;
3. Escaping from battle: Flees and collects nearby health boosts;
4. Engaging fire: The bot tries to kill the opponent and avoids being hit;

To avoid having a hard-coded agent, but can evolve in real-time.

A FALCON network is used for weapon selection:
To learn the different coping strategies of different opponents.
Using FALCON Networks to Store, Retrieve, and Adapt Knowledge

Behavior modeling:

Weapon selection:
Interpretable Rules

- Translated example rules for *behavior modeling*:

  | IF | health is around [87, 109], not being damaged, opponent is in sight, has adequate ammo, has health boost nearby, has no weapon nearby, possessing only primitive weapons, and currently in RUN_AROUND state; |
  | THEN | go into ENGAGE state; |
  | WITH | reward of 0.729. |

  | IF | health is around [2, 21], being damaged, opponent is in sight, has adequate ammo, has no health boost nearby, has no weapon nearby, possessing only default weapons, and currently in ENGAGE state (0.9); |
  | THEN | go into ESCAPE state; |
  | WITH | reward of 0.05. |

- Translated example rules for *weapon selection*:

  | IF | distance is very near [108, 317]; |
  | THEN | use flak cannon; |
  | WITH | reward of 0.838. |

  | IF | distance is far [1781, 2364]; |
  | THEN | use lightning gun; |
  | WITH | reward of 0.781. |
1. Introduction of Episodic Memory and EM ART

Episodic memory (EM):

- To remember one’s own experiences in an explicit and conscious manner
- Crucial in supporting many cognitive capabilities
- Needed during learning about context and about configurations of stimuli
- Not just a storage of one’s past experiences, but supporting the representation of complex conceptual and spatio-temporal relations among one’s experienced events and situations
1. Introduction of Episodic Memory and EM ART

- **EM-ART**
  - Fusion ART (Multi-channel Fuzzy ART) + Fuzzy ART
  - For encoding of EM in terms of events as well as spatio-temporal relations between events
  - incorporates a novel memory search procedure, which performs a continuous parallel search of stored episodic traces
  - Combined with a mechanism of gradual forgetting, the model is able to achieve a high level of memory performance and robustness.
Experimental results show

1) EM-ART produces highly robust performance in encoding and recalling events and episodes even with incomplete and noisy cues.

2) EM-ART provides enhanced performance in a noisy environment due to the process of forgetting.

3) EM-ART shows a higher tolerance toward noise and errors in the retrieval performance, compared with prior models of spatio-temporal memory.
2. Issues and Challenges

A. Memory Formation

- Two basic elements
  - Event: a snapshot of experience
    - By aggregating attributes of interest, a remembered event can be used to answer critical questions about the corresponding experience, such as what, where, and when
  - Episode: a temporal sequence of events

- To enable efficient encoding of events and episodes
  - Distinguish highly similar but semantically different events
  - Tolerate minor differences for events within a single episode, such as slight changes within observed events and their temporal order
2. Issues and Challenges

B. Memory Retrieval

- Event detection: search for similar memorized events

- Episode recognition:
  - Identification of a stored episode in the EM in response to a partial event sequence
  - Two basic requirements:
    1) Tolerance to incomplete cues, which only form parts of the stored episodes
    2) Tolerance to errors, e.g., noise in event attributes and variations in the order of event sequences

- Episode recall:
  - Playback of episode(s) in response to an external cue, e.g. “what did I do yesterday?”
  - The EM model should be able to answer the cue with the most closely matched episode, while identifying and tolerating the imperfection during recall
2. Issues and Challenges

C. Forgetting

- To avoid possible information overflow
- To preserve and strengthen important or frequently used episodes, and remove (or forget) unimportant ones
- A necessary condition for promoting efficient memory storage, as well as fast and accurate operation of EM in real-time environments
D. Summary

Basic requirements of an EM model:

1) Efficient event representation describing complex situations and events
2) Efficient episode representation for exploring spatio-temporal relations among events in the episode
3) Well-defined generalizations on representations, which accurately distinguish critical and irrelevant differences among them
4) High level of tolerance to incomplete or noisy cues
5) Fast memory operations
6) Tracking the importance of events and episodes in real time based on rewards, surprises, emotions, interpretation, and access frequency
7) Forgetting mechanism to deal with the limited memory capacity
3. EM-ART

- EM model
  - The $F_1$ layer holds the activation values of all situational attributes
  - Based on the $F_1$ pattern of activations, a cognitive node in $F_2$ is selected and activated as a recognition of the event
  - The activation pattern of an incoming event can be learned by adjusting the weights in the connections between $F_1$ and $F_2$
    - A sequence of events produces a series of activations in $F_2$
    - The activations in $F_2$ decay over time such that a graded pattern of activations is formed representing the order of the sequence

![NN architecture of the EM model](image-url)
3. EM-ART

- This activity pattern representing an episode, is similarly learned as weighted connections between $F_2$ and the selected category in $F_3$.

- Once an episode is recognized through a selected node in $F_3$, the complete episode can be reproduced by a top-down activation process (readout) from $F_3$ to $F_2$.

- The events in the episode can also be reproduced by reading out the activations from $F_2$ to $F_1$ following the order of the sequence held in the $F_2$ layer.
4. Event Encoding and Retrieval

- Event encoding
  - What (e.g. subject, relation, action, object, etc.)
  - Where (e.g. location, country, place, etc.)
  - When (e.g. date, time, day, night, etc.)

Ex) The *location* is expressed using a 3-D Cartesian coordinate system; *other task and internal states* include the observed distance from the enemy, the availability of collectable items, and the agent’s health and ammo level. There are *four behavior choices* available for the agent. The consequence of a battle situation (*e.g.*, kill and damages) is presented to the model as a *reward value*.

An example of the structure of an input event in first person shooter computer games
4. Event Encoding and Retrieval

- Fusion ART: A multi-channel ART
  - Used to learn individual events encoded as weighted connections between the $F_1$ and $F_2$ layers
    - *An event is represented as a multichannel input vector*
    - Each event’s attribute is represented as the activity of a node in the corresponding input field
A summary of the fusion ART dynamics

- **Input vectors:**
  - $I^k = (I_1^k, I_2^k, ..., I_n^k)$: an input vector, where $I_i^k \in [0, 1]$ denotes the input $i$ to channel $k$, for $k = 1, ..., n$. With complement coding, $I^k$ is augmented with a complement vector $\overline{I^k}$ such that $\overline{I_i^k} = 1 - I_i^k$

- **Input fields:**
  - $F_1^k$: an input field, holding the input pattern for channel $k$.
    - $x^k = (x_1^k, x_2^k, ..., x_n^k)$: the activity vector of $F_1^k$, receiving $I^k$ (including the complement)

- **Category fields:**
  - A category field $F_i$ and $i > 1$ denote that it is the $i$th field. The standard multichannel ART has only one category field, $F_2$.
    - $y = (y_1, y_2, ..., y_m)$: the activity vector of $F_2$
• Weight vectors:
  - \( w_j^k \): the weight vector associated with the \( j^{th} \) node in \( F_2 \) for learning the input pattern in \( F_1^k \)

• Parameters:
  - Choice parameters \( \alpha^k \geq 0 \), learning rate parameters \( \beta^k \in [0,1] \), contribution parameters \( \gamma^k \in [0,1] \), and vigilance parameters \( \rho^k \in [0,1] \), which determine each field’s dynamics

Basic operations
• Code activation:
  - A node \( j \) in \( F_2 \) is activated by the choice function:

\[
T_j = \sum_{k=1}^{n} \gamma^k \frac{|x^k \wedge w_j^k|}{\alpha^k + |w_j^k|}
\]
4. Event Encoding and Retrieval

- **Code competition:**
  - A code competition process selects a $F_2$ node with the highest choice function value:
    \[
    T_j = \max \{ T_j : \text{for all } F_2 \text{ node } j \} 
    \]
    where the winner is indexed at $J$.
  - A winner-take-all strategy: when a category choice is made at node $J$,
    \[
    y_j = 1; \text{ and } y_j = 0 \text{ for all } j \neq J. 
    \]

- **Template matching:**
  - To check if resonance occurs
  - For each channel $k$, it checks if the match function $m_j^k$ of the chosen node $J$ meets its vigilance criterion:
    \[
    m_j^k = \frac{|x_j^k \wedge w_j^k|}{|x_j^k|} \geq \rho^k 
    \]
4. Event Encoding and Retrieval

- If any of the vigilance constraints is violated, mismatch reset occurs or $T_j$ is set to 0 for the duration of the input presentation.
- Another $F_2$ node $J$ is selected using choice function and code competition until a resonance is achieved.
- If no selected node in $F_2$ meets the vigilance, an uncommitted node is recruited in $F_2$ as a new category node.

- **Template learning:**
  - Once a resonance occurs, for each channel $k$, the weight vector $w_j^k$ is modified by the following iterative learning rule:
    \[
    w_j^{k(new)} = (1 - \beta^k)w_j^{k(old)} + \beta^k (x^k \land w_j^{k(old)})
    \]

- **Activity readout:**
  - The chosen $F_2$ node $J$ may perform a readout of its weight vectors to an input field $F_1^k$ such that $x^{k(new)} = w_j^k$. 

A fusion ART network

A flexible architecture that can be made for a wide variety of purposes

- It can learn and categorize inputs and can be made to map a category to some predefined fields by a readout process to produce the output.

- No separate phase of operation is necessary for conducting recognition (activation) and learning
  - Learning can be conducted by adjusting the weighted connections while the network searches and selects the best matching node

- It can grow in response to novel patterns.
  - When no existing node can be matched, a new node is allocated to represent the new pattern.
Algorithm for Event Encoding and Retrieval

- An event can be encoded as an input vector to the network.
- The recognition task can be realized by a bottom-up activation given the input vector.
- The top-down activation (readout operation) achieves the recall task.
- The bottom-up and top-down operations for learning, recognition, and recalling an event.
4. Event Encoding and Retrieval

**Algorithm 1** Event Encoding

1. Given an input pattern of event as vector $\mathbf{x}^k$ in $F_1$
2. Activate every node $j$ in $F_2$ by choice function
   \[
   T_j = \sum_{k=1}^{n} \gamma_k \frac{|x^k \land w_j^k|}{a^k + |w_j^k|}
   \]
3. select node $J$ such that $T_J = \max \{T_j : \text{for all } F_2 \text{ node } j\}$,
4. set node activation $y_J \leftarrow 1$
5. WHILE match function $m_J^k = \frac{|x^k \land w_J^k|}{|x^k|} < \rho^k$
     (not in resonance)
     OR $J$ was selected previously
6. deselect and reset $J$ by $T_J \leftarrow 0$, $y_J \leftarrow 0$
7. select another node $J$ with $T_J = \max \{T_j : \text{for all } F_2 \text{ node } j\}$
8. IF no matching (resonance) $J$ can be found in $F_2$
9. THEN let $J \leftarrow J^0$, where $J^0$ is a newly recruited uncommitted node in $F_2$
10. learn $J$ as a novel event with $w_J^{k(\text{new})} = x^k$
5. Episode Learning and Retrieval

A. Episode Representation and Learning Algorithm

- The EM-ART extends the fusion ART to associate and group patterns across time
- An activation value in $F_2$ denotes a time point or a position in an ordered sequence
  - The most recently activated node in $F_2$ has the maximum activation of 1, while the previously selected ones are multiplied by a certain factor decaying the values over time
- Concurrently, the sequential pattern can be stored as weighted connections in the fusion ART network
  - Each node in $F_3$ represents an episode encoded as a pattern of sequential order.

Operations between $F_2$ and $F_3$. 
Algorithm 2 Episode Activation and Learning

1. FOR EACH subsequent event in episode $S$
2. select a resonance node $J$ in $F_2$ based on input $I^k$ in $F_1$
3. let node activation $y_J \leftarrow 1$ (or a predefined maximum value)
4. FOR EACH previously selected node $i$ in $F_2$
5. decay its activation by $y_i^{(new)} = y_i^{(old)}(1 - \tau)$
   or 0 if $y_i^{(old)} \leq 0$
6. Given activation vector $y$ formed in $F_2$ after the subsequent presentation of $S$
7. select a resonance node $J'$ in $F_3$ based on activation vector $y$
8. learn its associated weight vector as $w_J^{(new)} = y$ if $S$ is a novel episode
B. Episode Retrieval

- After the episodes are learned, any such episode can be recalled using various types of cues.
- A cue for the retrieval can be a partial sequence of any episode starting from the beginning or any position in the sequence.
- Based on the cue, the entire episode can be reproduced through the read out operation

**Algorithm 3 Episode Recognition**

1. FOR EACH incoming event
2. select a resonance node \( J \) in \( F_2 \) based on the corresponding event
3. let node activation \( y_J \leftarrow 1 \) (or maximum)
4. FOR EACH previously selected node \( i \) in \( F_2 \)
5. decay its activation by \( y_i^{(new)} = y_i^{(old)}(1 - \tau) \) or 0 if \( y_i^{(old)} \leq 0 \)
6. select a resonance node \( J' \) in \( F_3 \) based on \( y \) in \( F_2 \)
7. IF \( J' \) can be found THEN exit loop
Once an episode is recognized, the complete pattern of sequence can be reproduced readily in the $F_2$ layer by the read out operation from the selected node in $F_3$ to the nodes in $F_2$.

EM-ART uses a vector complementing the values in $F_2$, before reading out the complete events in $F_1$.

After the sequential pattern is read out to the field in $F_2$, expressed as vector $\mathbf{y}$, a complementing vector $\mathbf{\bar{y}}$ can be produced so that for every element $i$ in the vector $\bar{y}_i = 1 - y_i$.

Given $\mathbf{\bar{y}}$, the node corresponding to the largest element in $\bar{y}$ is selected first to be read out to the $F_2$ fields.

Subsequently, the current selected element in $\mathbf{\bar{y}}$ is suppressed by resetting it to zero, and the next largest is selected for reading out until everything is suppressed.

In this way, the whole events of the retrieved episode can be reproduced in the right order.
6. Forgetting in Episodic Memory

- Forgetting is essential to preserve and strengthen important and/or frequently used experiences, while removing unimportant or rarely occurred ones.

- A memory strength value \( s_j \in [0,1] \), is associated with each event encoded by a \( F_2 \) node.
  - Initially, \( s_j \) is set to \( s_{init} \) and gradually decays by decay factor \( \delta_s \in [0,1] \).
  - Upon an event re-activation, \( s_j \) is increased by an amount proportional to reinforcement rate \( r_s \in [0,1] \).
  - The strength of an event \( e_j \) at time \( t \) can be computed as follows:
    \[
    s_j(t) = \begin{cases} 
      s_{init} & \text{if } e \text{ is just created at } t \\
      s_j(t - 1) + (1 - s_j(t - 1)) r_s & \text{if } e \text{ is reactivated at } t \\
      s_j(t - 1)(1 - \delta_s) & \text{otherwise}
    \end{cases}
    \]

- An event having \( s_j \), falling below a threshold \( t_s \in [0,1] \) will be removed from EM together with all of its weighted connections to/from other event and episode nodes.
7. Experiments

Experiment setup

- First-person shooter game
- A non-player character (NPC) agent receives events describing the situation it experiences
- 100 battles (i.e. episodes), 7735 events in the data set.
- The number of events within an episode varies from 7 to over 250

EM Model Sizes at Various Vigilances

<table>
<thead>
<tr>
<th>$(\rho^e, \rho^s)$</th>
<th>Number of episodes</th>
<th>Number of events</th>
<th>Number of weights $F_1 - F_2(k)$</th>
<th>Number of weights $F_2 - F_3(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1.0, 1.0)</td>
<td>100</td>
<td>6705</td>
<td>509</td>
<td>1341</td>
</tr>
<tr>
<td>(1.0, 0.9)</td>
<td>100</td>
<td>6705</td>
<td>509</td>
<td>1341</td>
</tr>
<tr>
<td>(0.95, 1.0)</td>
<td>98</td>
<td>2692</td>
<td>294</td>
<td>527</td>
</tr>
<tr>
<td>(0.95, 0.9)</td>
<td>98</td>
<td>2692</td>
<td>294</td>
<td>527</td>
</tr>
</tbody>
</table>

- The similarity between events is relatively high, while most of exemplar episodes are distinct from each other.
### TABLE III
ACCURACIES (IN %) OF RETRIEVING WITH INCOMPLETE CUES

<table>
<thead>
<tr>
<th>Cue type</th>
<th>$(\rho^e, \rho^s)$</th>
<th>Cue length</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1.0, 1.0)</td>
<td>Full length</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>(a) Partial cues from the beginning of episodes</td>
<td>(1.0, 0.9)</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(0.95, 1.0)</td>
<td>98</td>
<td>93</td>
<td>93</td>
<td>89</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>(0.95, 0.9)</td>
<td>98</td>
<td>93</td>
<td>93</td>
<td>89</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>(1.0, 1.0)</td>
<td>Full length</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>(b) Partial cues from the end of episodes</td>
<td>(1.0, 0.9)</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(0.95, 1.0)</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>97</td>
<td>96</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>(0.95, 0.9)</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>97</td>
<td>96</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>(1.0, 1.0)</td>
<td>Full length</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>(c) Partial cues from arbitrary location of episodes</td>
<td>(1.0, 0.9)</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>99</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>(0.95, 1.0)</td>
<td>98</td>
<td>97</td>
<td>93</td>
<td>88</td>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>(0.95, 0.9)</td>
<td>98</td>
<td>94</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
</tr>
</tbody>
</table>
7. Experiments

- Retrieving from beginning of episodes:
  - Partial sequences from the beginning of recorded episodes as cues for retrieving the episodes

- Retrieving from end of episodes
- Retrieving from arbitrary location of episodes
8. Summary

- EM-ART: a new EM model, based on a class of self-organizing NNs known as fusion ART and fuzzy ART, which can
  - learn complex relations between events and retrieve episodes with imperfect or noisy cues,
  - consider memory trace as a continuous series of events with coherent representation of chunks of episodes as units of experience, by representing events as multichannel activation patterns allowing retrieval based on partial matching,
  - suppress irrelevant attributes of an event through learning, since the fusion ART fuzzy operations and the complement coding technique enable patterns to be generalized,
  - cluster both individual events and their sequential patterns based on similarities instead of holding all incoming information in a trace buffer → more compact storage and efficient processing
8. Summary

- handle learning timed responses and also other aspects of EM, in particular, sequential ordering and multimodal association, while existing associative networks may still be limited in recalling information based on sequential cues,

- learn various lengths of complex sequential patterns at once by employing two levels of fusion ART:
  - The first level deals with repetition by growing separate categories
  - The second level clusters sequential patterns formed at the first level,

- offer modularity and flexibility by employing two levels of clustering that may be used by other systems.