Lecture 10

Unsupervised Learning NN:
Hierarchical Temporal Memory - IV

Implementation of SP and TP:
Pseudo Code for MATLAB
Spatial pooling

- **Initialization**
  - Input dimension
    - The number of columns
  - Column dimension
    - The number of input bits
Potential mapping
- Each column is mapped to ‘a subset of input bits’ through synapses.
- Each column can only be connected to the subset of input bits.
- The subset of input bits is called ‘receptive field’.
- Parameters: potential radius and potential percent

Potential radius: Global
Potential percent: 50% (10 out of 20)
Receptive field: 0th bit, 2nd bit, 3rd bit. ..., 19th bit

Potential radius: 2, local
Potential percent: 60% (3 connections out of 5)
Receptive field of col₁: 7th bit, 9th bit, 11th bit
Receptive field of col₂: 8th bit, 9th bit, 12th bit
Permanence

- To show how permanent a synapse connection is
  - A synapse is a connection between a column and an input bit.
  - Parameter: \textit{synPermConnected} as a threshold
    - Valid synapse: if the permanence of the synapse is above \textit{synPermConnected}

- Every synapse has its permanence value.

- Three kinds of initialization methods:
  - Gaussian
  - Random
  - Constant
After permanence initialization:

- Gaussian initialization
- Random initialization
- Constant initialization
## SDR

- Calculate ‘overlap’ score
  - Overlap score
    - An overlap score of a column is the number of active bits (‘1’) connected to the column.
    - Parameter: *MinOverlap*
      - Overlap scores below *MinOverlap* are set to 0.
      - *Exclude columns with meaningless overlap scores.*

\[
\text{OverlapScore}(Col_i) = \sum \text{SynapseWeight} \times \text{bit} = 0\times1+1\times1+0\times0+1\times1+1\times0+1\times1+1\times1 = 4
\]
Global inhibition

- For all columns, find the $N^{th}$ largest overlap score.
- Columns that have overlap score larger than the $N^{th}$ largest overlap score become activated columns.
- Other columns are inhibited (deactivated).
- Parameter: $N$
  - The density of activated columns is controlled by adjusting $N$.

Before inhibition:

After inhibition:

Overlap scores:

$\uparrow$ Global inhibition with $N = 2$
Learning: update connections

- Learning for activated columns
  - If a column is activated:
    - Increase the permanence of a synapse, if the synapse is connected to input bit ‘1’
    - Decrease the permanence of a synapse, if the synapse is connected to input bit ‘0’
  - Parameters: permInc and permDec
Learning for columns that rarely overlap with input bits
   (Those columns are not used often for SP.)
   - Increase all permanence values of those columns:
     - Calculate average overlap scores for every column for \( dutyCyclePeriod \).
       - \( dutyCyclePeriod \) is the period used to calculate the average overlap score.
     - If an average overlap score of a column is too small (below \( minDutyCycle \)),
       increase all permanence values connected to the column.
     - Parameter: \( minDutyCycle \) and \( dutyCyclePeriod \)

Learning disabled
   - Learning can be disabled.
   - In this case, SP stops just after computing the global inhibition.
Temporal Pooler Implementation

Temporal pooling

- Start with a set of activated columns representing the FF input or the output of SP.

Initialization

- Column dimension = The number of columns
- Cell dimension = The number of cells per column
Phase 1: Activate the correctly predicted cells

- For each active column, check for cells in the column that were in a predictive state at $t - 1$, and activate them.

Correctly predicted cells
- Marked as an active cell
- Marked as a winner cell

Predicted but inactive cells
- Marked as a predicted but inactive cell

Temporal Pooler Implementation

Before Phase 1 at $t$

After Phase 1 at $t$

After Phase 4 at $t - 1$
Pseudo code

Phase 1: Activate the correctly predicted cells.

For each previously predicted cell (predicted at time t-1)
   if in active column
      mark it as an active cell
      mark it as a winner cell
      (cell in active column at t and predicted cell at t-1 => active cell at t)
   if not in active column
      mark it as a predicted but inactive cell

Input1 : prevPredictiveCells (Indices of predicted cells in `t-1`)
Input2 : activeColumns (Indices of activated columns in `t`)
Return :
`activeCells`, `winnerCells`, `predictedActiveColumns`, `predictedInactiveCells`
Phase 2: Burst unpredicted columns

- If an activated column has no predicted cells (predicted at $t - 1$), activate all the cells in the column (‘burst’).
- Choose the best-matching cell and its best-matching segment in a burst column.
  - Mark the best-matching cell as a winner cell.
  - If the best-matching cell has no best-matching segment,
    - & if there are any previous winner cells (winner cells at $t - 1$)
      - Add a segment to the cell.
- Mark those segments as ‘learning segments’

The best-matching cell (marked as a winner cell)

After Phase 2 at $t$

After Phase 1 at $t$
Temporal Pooler Implementation

- Choosing the best-matching cell
  - Get the cell with the best matching segment
    - Best matching segment is a segment that has the biggest ‘matching score’
    - Matching score is the number of synapses connected to activated cells at $t - 1$.
    - If none were found, pick the least used cell
      - Randomly select a cell among the cells of the smallest number of segments

Cell activation status after Phase 4 at $t - 1$

- $cell_2$ of $col_{18}$ becomes a winner cell
- Matching score ($seg_1$) = 3
- Matching score ($seg_2$) = 2

$col_{18}$: the burst column at $t$
Temporal Pooler Implementation

- **Pseudo code**

  Phase 2: Burst unpredicted columns.
  - for each unpredicted active column
    - mark all cells as active
    - mark the best matching cell as a winner cell
      - if it has no matching segment
        - (optimization) if there are prev winner cells
          - add a segment to it, and mark the segment as a matching segment
        - mark the matching segments as learning segments

<table>
<thead>
<tr>
<th>Input1 : activeColumns</th>
<th>Indices of active columns in <code>t</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Input2 : predictedActiveColumns</td>
<td>Indices of predicted =&gt; active columns in <code>t</code></td>
</tr>
<tr>
<td>Input3 : prevActiveCells</td>
<td>Indices of active cells in <code>t-1</code></td>
</tr>
<tr>
<td>Input4 : prevWinnerCells</td>
<td>Indices of winner cells in <code>t-1</code></td>
</tr>
<tr>
<td>Return :</td>
<td></td>
</tr>
<tr>
<td>‘activeCells’, ‘winnerCells’, ‘learningSegments’</td>
<td></td>
</tr>
</tbody>
</table>
Phase 3: Perform learning by adapting segments

- For all learning segments and previously activated segment at \( t - 1 \),
  - If a segment is a learning segment or from a winner cell,
    - Update synapses on the segment.
    - Strengthen synapses connected to activated cell at \( t - 1 \);
    - Weaken synapses connected to inactive cell at \( t - 1 \).

\( \rho \): Permanence of a synapse

Cell activation status after Phase 4 at \( t - 1 \)
- If a learning segment,
  - Add some synapses to the segment
  - The number of newly generated synapses = \( \text{maxNewSynapseCount} \)
  - The number of active synapses
  - The newly generated synapses are connected to previous winner cells at \( t-1 \).

\[
\text{Cell activation status after Phase 4 at } t - 1
\]

\[
\# \text{ of newly generated synapses} = \text{maxNewSynapseCount} - \# \text{ of active synapses} = 4 - 3 = 1
\]
Temporal Pooler Implementation

- Pseudo code

Phase 3: Perform learning by adapting segments
- (learning) for each prev active or learning segment
  - if learning segment or from winner cell
    - strengthen active synapses
    - weaken inactive synapses
  - if learning segment
    - add some synapses to the segment
    - subsample from prev winner cells

Input1 : prevActiveSegments
  - Indices of active segments in `t-1`
Input2 : learningSegments
  - Indices of learning segments in `t`
Input3 : prevActiveCells
  - Indices of active cells in `t-1`
Input4 : winnerCells
  - Indices of winner cells in `t`
Input5 : prevWinnerCells
  - Indices of winner cells in `t-1`

Return :
Phase 4: Compute predictive cells due to lateral input on dendrites

- For all segments, calculate segment activation
  - A segment is activated when its activation score is above \( activationThreshold \).
  - An activation score of a segment shows the number of activated cells connected to the segment at time \( t \).

\[
ActivationScore(seg_{1th}) = \sum \text{SynapseWeight} \times \text{TargetcellState}
\]

\[
= 1 \times 0 + 0 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 1 = 3 > activationThreshold = 2
\]

\[
ActivationScore(seg_{2th}) = 1 \times 1 + 1 \times 0 + 0 \times 0 + 0 \times 0 + 1 \times 0 + 0 \times 1 + 0 \times 0 + 1 \times 0 = 1 < activationThreshold
\]
If a segment is activated, change its cell states into ‘predicted’.

(A cell is changed into predicted states if one of its segments is activated.)
● Pseudo code

Phase 4: Compute predictive cells due to lateral input on distal dendrites.
- for each distal dendrite segment with activation score $\geq$ activationThreshold
  - mark the segment as active
  - mark the cell as predictive

Input1 : activeCells
  - Indices of active cells in ‘t’
Return :
  - ‘activeSegments’, ‘predictiveCells’.
■ Reset (only when at the end/start of a new sequence)
  - Indicating that this is the end of a sequence.
  - Also indicating the next-coming input will be the start of a new sequence.
  - Resetting the sequence state of the temporal pooler
    - Reset all cell states:
      - Activated cell
      - Predicted cell
      - Activated segment
      - Winner cell
Learning disabled

- Learning can be disabled.
- In this case, TP skips Phase 3 (Learning by adapting segments)
Example

- Suppose a sequence of active columns

  - TP is going to learn this sequence.

  ![Diagram showing sequence of active columns at different times](image)
- Initialize TP:
  - ColumnDimension = 22
  - CellDimension = 5

- \( t = 0 \)
  - Input:
    - Phase 1: Activate the correctly predicted cells.
      - No cells are activated at phase 1.
        - Because there were no predicted cells at \( t = -1 \).
Phase 2: Burst unpredicted columns.
- All activated columns are unpredicted. Thus, burst all columns.
- Winner cells (the best-matching cells) are randomly chosen
  - Because there are no segments to compare
- No segments are added to the winner cells
  - Because there are no previous winner cells (winner cell at $t = -1$)
- No segments are marked as learning segments
- Phase 3: Perform learning by adapting segments
  - No segments are adapted (no learning)
    - Because there are no previously activated or learning segments.

Cell activation status after phase 3

- Phase 4: Compute predictive cells due to lateral input on dendrites.
  - No segments are activated, and no cells are predicted.
    - Because there are no segments.

Cell activation status after phase 4
Example

- $t = 1$
  - Input:
    
    ![Input(1) at $t = 1$](image)
  
  - Phase1: Activate the correctly predicted cells.
    - No cells are activated at phase 1.
      - Because there were no predicted cells at $t = 0$.

    ![After phase 1](image)
– Phase2: Burst unpredicted columns.
  – All activated columns are unpredicted. Thus, burst all columns.
  – Winner cells (the best-matching cells) are randomly chosen
    – Because there are no segments to compare
  – Segments are added to the winner cells.
    – Each winner cell now has a segment.
      – Each segment has no synapse yet.
  – Each segment is marked as a learning segment.
Phase 3: Perform learning by adapting segments

- Synapses are added to the learning segments.
- Suppose \( maxNewSynapseCount = 6 \)
- \# of newly generated synapses
  \[ = \min( maxNewSynapseCount - \# of active synapses, \# of previous winner cells) \]
  \[ = \min(6 - 0, 5) = 5 \]
- The newly generated synapses are connected to previous winner cells at \( t-1 \).
  - (columnIndex, cellIndex) = (1, 4), (2, 3), (3, 5), (4, 1), (5, 2)
- Totally 5*6=30 synapses are generated.
- Suppose \( synInitialPerm = 0.5 \).
- Then, the permanences of the synapses equal to 0.5.

Cell activation status after phase 3
- Phase 4: Compute predictive cells due to lateral input on dendrites.
  - No segments are activated, and no cells are predicted.
  - Activation scores of the segments are zero.
Example

- $t = 2$
  - Input:
  
  ![Input(2) at $t = 2$]

- Phase1: Activate the correctly predicted cells.
  - No cells are activated at phase 1.
    - Because there were no predicted cells at $t = 1$.

  ![After phase 1](Segments are not drawn here.)
Phase 2: Burst unpredicted columns.
- All activated columns are unpredicted. Burst all columns.
- Winner cells (the best-matching cells) are randomly chosen
  - Because there are no segment to compare
- Segments are added to the winner cells
  - Each winner cell now has a segment.
    - Each segment has no synapse yet.
- Each segment is marked as a learning segment.

Cell activation status after phase 2

(Only segments of activated cells are drawn here.)
- Phase 3: Perform learning by adapting segments
  - Synapses are added to the learning segments.
  - Suppose $\text{maxNewSynapseCount} = 6$
  - # of newly generated synapses
    - $= \min(\text{maxNewSynapseCount} - \# \text{ of active synapses, } \# \text{ of previous winner cells})$
    - $= \min(6 - 0, 6) = 6$
  - The newly generated synapses are connected to previous winner cells at $t-1$.
    - $(\text{columnIndex, cellIndex}) = (6, 5), (7, 5), (8, 1), (9, 2), (10, 3), (11, 4)$
  - Totally $6 \times 5 = 30$ synapses are generated.
  - Suppose $\text{synInitialPerm} = 0.5$.
  - Then, the permanences of the synapses equal to 0.5.

Cell activation status after phase 3

(Only segments of activated cells are drawn here.)
Phase 4: Compute predictive cells due to lateral input on dendrites.

- No segments are activated, and no cells are predicted.
- Activation scores of the segments are zero for all segments.

(Only segments of activated cells are drawn here.)
Example

- $t = 3$
  - Input:
    
    Image of input at time $t = 3$
    
    - Phase1: Activate the correctly predicted cells.
      - No cells are activated at phase 1.
        - Because there were no predicted cells at $t = 2$.
    
    Image after phase 1
    
    (Segments are not drawn here.)
– Phase 2: Burst unpredicted columns.
  – All activated columns are unpredicted. Burst all columns.
  – Winner cells (the best-matching cells) are randomly chosen
    – Because there are no segment to compare
  – Segments are added to the winner cells
    – Each winner cell now has a segment.
      – Each segment has no synapse yet.
  – Each segment is marked as a learning segment.

Example

Cell activation status after phase 2

(Only segments of activated cells are drawn here.)
Phase 3: Perform learning by adapting segments

- Synapses are added to the learning segments.
- Suppose $maxNewSynapseCount = 6$
- Number of newly generated synapses
  \[ = \min(maxNewSynapseCount - \# \text{ of active synapses}, \# \text{ of previous winner cells}) \]
  \[ = \min(6 - 0, 5) = 5 \]
- The newly generated synapses are connected to previous winner cells at $t-1$.
  - (columnIndex, cellIndex) = (12, 3), (14, 1), (15, 4), (16, 5), (17, 2)
- Totally $5 \times 6 = 30$ synapses are generated.
- Suppose $synInitialPerm = 0.5$.
- Then, the permanences of the synapses equal to 0.5.

Cell activation status after phase 3

(Only segments of activated cells are drawn here.)
Phase 4: Compute predictive cells due to lateral input on dendrites.
- No segments are activated, and no cells are predicted.
- Activation scores of the segments are zero for all segments.

Cell activation status after phase 4

(Only segments of activated cells are drawn here.)
Example

- Reset:
  - Reset all cell states
    - Because a sequence has ended and a new sequence will begin:
      - Activated cell
      - Predicted cell
      - Activated segment
      - Winner cell

Cell activation status
After reset at $t = 3$

(Segments are not drawn here.)
- \( t = 4 \)
  - Input:
    
    ![Input(0) at \( t = 4 \)](image-url)

  - Phase 1: Activate the correctly predicted cells.
    - No cells are activated at phase 1.
      - Because there were no predicted cells at \( t = 3 \).

    ![After Phase 1](image-url)

    (Segments are not drawn here.)
Phase 2: Burst unpredicted columns.
- All activated columns are unpredicted. Burst all columns.
- Winner cells (the best-matching cells) are randomly chosen
  - Because there are no segment to compare
- No segments are added to the winner cells
  - Because there are no previous winner cells (due to the reset at $t = 3$)
- No segments are marked as learning segment

Cell activation status after phase 2

(Segments are not drawn here.)
- Phase 3: Perform learning by adapting segments
  - No segments are adapted (no learning)
    - Because there are no previously activated or learning segment.

Cell activation status after phase 3

(Segments are not drawn here.)
Phase 4: Compute predictive cells due to lateral input on dendrites.

- Suppose $activationThreshold=2$
- Activation scores of segments of cell (6,5), (7,5), (8,1), (9,2), (10,3), (11,4) $= 5 \geq activationThreshold$
  - Those segments are activated.
  - Cell states of those cells are changed into predicted.
- Activation scores of other segments are zero.
• $t = 5$
  - Input:

  ![Input(1) at $t = 5$]

  - Phase 1: Activate the correctly predicted cells.
    - Predicted cells at $t = 4$ are activated.
    - Those activated cells are marked as a winner cell.

  ![After phase 1]

  (Segments are not drawn here.)
- Phase 2: Burst unpredicted columns.
  - There are no predicted but inactive columns.

Cell activation status after phase 2

(Segments are not drawn here.)
Phase 3: Perform learning by adapting segments
- There are 6 previously activated segments (activated segments at $t = 4$).
  - They are from winner cells.
  - Therefore adapt these segments.
- Suppose $perInc = 0.01$
  - The permanences of those synapses are now 0.51.

Cell activation status after phase 3

(Only segments of activated cells are drawn here.)
Phase 4: Compute predictive cells due to lateral input on dendrites.

- Suppose $activationThreshold = 2$
- Activation score of segments of cell (12,3), (13,1), (14,4), (15,5), (16,2) $\geq activationThreshold$
  - Those segments are activated.
  - Cell states of those cells are changed into predicted.
- Activation scores of other segments are zero.

Cell activation status after phase 4

(Only segments of predicted cells are drawn here.)
$t = 6$

- Input:

\[
\begin{align*}
\text{Input(2) at } t = 6
\end{align*}
\]

- Phase 1: Activate the correctly predicted cells.
  - Predicted cells at $t = 5$ are activated.
  - Those activated cells are marked as a winner cell.

\[
\begin{align*}
\text{After phase 1}
\end{align*}
\]

(Segments are not drawn here.)
Phase 2: Burst unpredicted columns.
  - There are no predicted but inactive columns.

Cell activation status after phase 2

(Segments are not drawn here.)
Phase 3: Perform learning by adapting segments
- There are 5 previously activated segments (activated segments at $t = 5$).
  - They are from winner cells.
    - Therefore, adapt these segments.
- Suppose $perInc = 0.01$
  - The permanence of those synapses are now 0.51.

Cell activation status after phase 3

(Only segments of activated cells are drawn here.)
- Phase 4: Compute predictive cells due to lateral input on dendrites.
  - Suppose $activationThreshold=2$
  - Activation scores of segments of cell (17,1), (18,4), (19,5), (20,5), (21,2), (22,3) $\geq activationThreshold$
    - Those segments are activated.
    - Cell states of those cells are changed into predicted.
  - Activation scores of other segments are zero.

Cell activation status after phase 4

(only segments of predicted cells are drawn here.)
$t = 7$

- Input:

- Phase 1: Activate the correctly predicted cells.
  - Predicted cells at $t = 6$ are activated.
  - Those activated cells are marked as a winner cell.
- Phase 2: Burst unpredicted columns.
  - There are no predicted but inactive columns.
Phase 3: Perform learning by adapting segments

- There are 6 previously activated segments (activated segments at $t = 6$).
  - They are from winner cells.
    - Therefore, adapt these segments.
- Suppose $perInc = 0.01$
  - The permanence of those synapses are now 0.51.

Cell activation status

(Only segments of activated cells are drawn here.)
- Phase 4: Compute predictive cells due to lateral input on dendrites.
  - No segments are activated, and no cells are predicted.
  - Activation scores of the segments are zero for all segments.

Cell activation status after phase 4

(Segments are not drawn here.)
Result

- TP successfully learned the sequence in 8 steps (from $t = 0$ to $t = 7$).