Mathematical Framework for Neuron Pattern Learning via Eigenvalue-Guided Tensor Updates

The Devengers Project

December 2025

Abstract

We present a complete mathematical specification of a neuron learning system that operates over DOM element observations. The system uses eigenvalue decomposition with softmax-normalized eigenvectors to update bias distributions and position matrices, culminating in pattern classification via tensor similarity analysis. The framework distinguishes between UNKNOWN patterns (requiring initialization) and defined patterns (DATA_INPUT, ACTION_ELEMENT, CONTEXT_ELEMENT, STRUCTURAL), with different eigenvalue update flows for each.

Contents

1 System Overview

1.1 Pattern Space

The system operates over a discrete pattern space:

$$\mathcal{P} = \{\text{DATA\_INPUT}, \text{ACTION\_ELEMENT}, \text{CONTEXT\_ELEMENT}, \text{STRUCTURAL}, \text{UNKNOWN}\} \tag{1}$$

Indexed as:

$$\text{DATA\_INPUT} \mapsto 0 \tag{2}$$
$$\text{ACTION\_ELEMENT} \mapsto 1 \tag{3}$$
$$\text{CONTEXT\_ELEMENT} \mapsto 2 \tag{4}$$
$$\text{STRUCTURAL} \mapsto 3 \tag{5}$$
$$\text{UNKNOWN} \mapsto 4 \tag{6}$$

1.2 Position Space

Each neuron observes 6 positions in the DOM tree:

$$\mathcal{Q} = \{\text{self}, \text{parent}, \text{up}, \text{down}, \text{left}, \text{right}\} \tag{7}$$

1.3 State Variables

Bias Vector $\mathbf{b}_{\text{initial}}, \mathbf{b}_{\text{final}} \in \mathbb{R}^5$: Probability distribution over patterns, where:

$$\sum_{i=0}^{4} b_i = 1, \quad b_i \geq 0 \tag{8}$$

Position Matrix $\mathbf{B} \in \mathbb{R}^{5\times5}$: Row-stochastic matrix encoding position-to-pattern mapping:

$$\sum_{j=0}^{4} B_{ij} = 1, \quad B_{ij} \geq 0 \tag{9}$$

Eigenvalues $\alpha, \beta, \gamma, \zeta \in \mathbb{R}$: Four scalar eigenvalues controlling updates.

2 Observation and Transformation Pipeline

2.1 DOM Observation (25D Space)

For each position $q \in \mathcal{Q}$ and pattern $p \in \mathcal{P}$, we observe a DOM element and extract a 25-dimensional feature vector:

$$\mathbf{v}_{q,p} = [d_0, d_1, \ldots, d_8, \text{coverage}, c_0, c_1, \ldots, c_{14}] \in \mathbb{R}^{25} \tag{10}$$

Where:

- $d_0, \ldots, d_8$: Base dimensions (semantic, state, data, visual, interaction, relational, validation, accessibility, domain)
- coverage $= \frac{1}{9}\sum_{i=0}^{8} \mathbb{1}[d_i \geq 0.5]$
- $c_0, \ldots, c_{14}$: Binary combination features derived from base dimensions

2.2 Relational Transform $T : \mathbb{R}^{n\times 25} \to \mathbb{R}^{n\times 87}$

Given n vectors in $\mathbb{R}^{25}$, the transform T maps them to $\mathbb{R}^{87}$ by encoding relational properties:

$$T(\mathbf{V}) = [\mathbf{V}_{:,10:25}, \mathbf{R}_{\text{sharing}}, \mathbf{R}_{\text{optional}}] \in \mathbb{R}^{n\times 87} \tag{11}$$

Where:

- First 15 dimensions: combination features (already binary)
- Next 72 dimensions: For each base dimension $d \in \{0, \ldots, 8\}$, 8 relational flags:
  - 4 sharing flags: ≥ 4 share, ≥ 3 share, exactly 2 share, exactly 1 (unique)
  - 4 optionality flags: ≥ 4 have 0.5, ≥ 3 have 0.5, exactly 2 have 0.5, exactly 1 has 0.5

Example: If 3 patterns have $d_0 = 0.5$:

$$R_{0,\text{opt},\geq 3} = 1, \quad R_{0,\text{opt},\geq 4} = 0 \tag{12}$$

2.3 Expectation Tensor $E \in \mathbb{R}^{5\times 6\times 87}$

Constructed once during initialization. For each pattern p and position q, we have a predefined 25D expectation vector. We stack all 5 patterns at each position and apply T:

$$E_{:,q,:} = T(\mathbf{V}^{(q)}_{\text{expect}}) \tag{13}$$

Where $\mathbf{V}^{(q)}_{\text{expect}} \in \mathbb{R}^{5\times 25}$ contains the 5 pattern expectations at position q.

2.4 Observation Tensor $O \in \mathbb{R}^{5\times 6\times 87}$

Built dynamically via $T_\zeta()$. For each position q:

1. Observe DOM element (handling voids via membrane rerouting)
2. Create observation vectors for all 5 patterns: $\mathbf{o}_{q,0}, \ldots, \mathbf{o}_{q,4} \in \mathbb{R}^{25}$
3. Apply transform: $O_{:,q,:} = T([\mathbf{o}_{q,0}, \ldots, \mathbf{o}_{q,4}])$

3 True Similarity Operator $\odot$

3.1 Definition

For matrices $A \in \mathbb{R}^{m\times d}$ and $B \in \mathbb{R}^{n\times d}$, the true similarity matrix is:

$$(\mathbf{A} \odot \mathbf{B})_{ij} = \frac{1}{d}\sum_{k=1}^{d}(1 - |A_{ik} - B_{jk}|) \tag{14}$$

This measures L1-based similarity:

- When $A_{ik} = B_{jk}$: contributes 1 (perfect match)
- When $|A_{ik} - B_{jk}| = 1$: contributes 0 (complete mismatch)

Example:

$$\mathbf{a} = [1, 0, 1, 0.5] \tag{15}$$

$$\mathbf{b} = [1, 1, 1, 0] \tag{16}$$

$$\mathbf{a} \odot \mathbf{b} = \frac{1}{4}[(1-0) + (1-1) + (1-0) + (1-0.5)] = \frac{2.5}{4} = 0.625 \tag{17}$$

3.2 Properties

- Symmetric when $m = n$ and $A = B$: $(A \odot A)^T = A \odot A$
- Range: $[0, 1]$ (normalized similarity)
- Handles binary and continuous features uniformly

4 Softmax Normalization

All eigenvectors are normalized using softmax before being used in updates:

$$\text{softmax}(\mathbf{v})_i = \frac{e^{v_i - \max(\mathbf{v})}}{\sum_j e^{v_j - \max(\mathbf{v})}} \tag{18}$$

This ensures:

- $\sum_i \text{softmax}(\mathbf{v})_i = 1$ (probability distribution)
- Positive values: $\text{softmax}(\mathbf{v})_i > 0$
- Numerical stability via $\max(\mathbf{v})$ subtraction

5 Eigenvalue Extraction

5.1 Dominant Eigenvalue Selection

Given a covariance matrix $\mathbf{C} \in \mathbb{R}^{5\times5}$, we extract eigenvalue-eigenvector pairs $(\lambda_i, \mathbf{v}_i)$ for $i = 1, \ldots, 5$.

The selection strategy uses softmax-scaled values to choose the optimal eigenvalue:

Algorithm 1 Dominant Eigenvalue Selection

```
1: Compute all eigenvalues: $\{\lambda_1, \ldots, \lambda_5\}$
2: Compute corresponding eigenvectors: $\{\mathbf{v}_1, \ldots, \mathbf{v}_5\}$
3: for each eigenvalue $i$ do
4:     $\mathbf{u}_i \leftarrow \text{softmax}(\mathbf{v}_i)$
5:     $s_i \leftarrow \lambda_i \cdot \mathbf{u}_i$                          ▷ Scaled vector
6:     $w_{\text{unknown},i} \leftarrow s_i[4]$                      ▷ UNKNOWN component
7:     $\Delta_i \leftarrow \max(s_i[0:4]) - \min(s_i[0:4])$   ▷ Pattern differentiation
8: end for
9: Select $i^* = \arg\min_i w_{\text{unknown},i}$ subject to $\Delta_i = \max_j \Delta_j$
10: return $(\lambda_{i^*}, \mathbf{v}_{i^*})$
```

Objective:

- Minimize UNKNOWN component (prefer defined patterns)
- Maximize pattern differentiation (clear pattern separation)

6 Eigenvalue Update Flow

6.1 UNKNOWN Pattern Flow

For neurons with current pattern = UNKNOWN:

$$\boxed{\text{UNKNOWN: } \alpha \to \mathbf{b}_{\text{initial}}, \quad \gamma \to \mathbf{B}, \quad \zeta \to \mathbf{b}_{\text{final}}} \tag{19}$$

Phase 1: α Update (Self-Identity)

1. Construct self-observation matrix: $\mathbf{S}^* \in \mathbb{R}^{5\times 87}$
$$\mathbf{S}^* = T(\mathbf{X}) \tag{20}$$
where $\mathbf{X} \in \mathbb{R}^{5\times 25}$ is the self-expectation matrix.
2. Compute covariance:
$$\mathbf{C}_\alpha = \mathbf{S}^* \odot \mathbf{S}^* \in \mathbb{R}^{5\times 5} \tag{21}$$
3. Extract dominant eigenvalue: $(\alpha, \mathbf{v}_\alpha) = \text{DominantEigen}(\mathbf{C}_\alpha)$
4. Apply softmax:
$$\tilde{\mathbf{v}}_\alpha = \text{softmax}(\mathbf{v}_\alpha) \tag{22}$$
5. Construct update matrix:
$$\mathbf{M}_\alpha = \alpha \cdot (\tilde{\mathbf{v}}_\alpha \otimes \tilde{\mathbf{v}}_\alpha) \tag{23}$$
6. Update bias:
$$\mathbf{b}_{\text{initial}} \leftarrow \frac{\mathbf{M}_\alpha \mathbf{b}_{\text{initial}}}{\|\mathbf{M}_\alpha \mathbf{b}_{\text{initial}}\|_1} \tag{24}$$

Phase 2: γ Update (Position Structure Initialization)

1. Build void-aware tensor $T_\gamma \in \mathbb{R}^{5\times 6\times 25}$ (observations at all positions)
2. Flatten: $\mathbf{T}_{\gamma,\text{flat}} \in \mathbb{R}^{5\times 150}$
3. Compute covariance:
$$\mathbf{C}_\gamma = \mathbf{T}_{\gamma,\text{flat}} \odot \mathbf{T}_{\gamma,\text{flat}} \tag{25}$$
4. Extract: $(\gamma, \mathbf{v}_\gamma) = \text{DominantEigen}(\mathbf{C}_\gamma, \text{use\_second} = \text{True})$
5. Update B matrix:
$$\mathbf{B} \leftarrow \text{RowNormalize}(\mathbf{M}_\gamma \mathbf{B}) \tag{26}$$
where $\mathbf{M}_\gamma = \gamma \cdot (\tilde{\mathbf{v}}_\gamma \otimes \tilde{\mathbf{v}}_\gamma)$.

Phase 3-4: Skip β UNKNOWN patterns skip β update. The B matrix from γ is preserved.

Phase 5: ζ Update (Final Decision)

1. Build observation tensor: $O = T_\zeta() \in \mathbb{R}^{5\times6\times87}$
2. For each position $q \in \{0, \ldots, 5\}$, compute:
$$\mathbf{G}_q = E_{:,q,:} \odot O_{:,q,:} \in \mathbb{R}^{5\times5} \tag{27}$$
3. Average across positions:
$$\mathbf{G} = \frac{1}{6}\sum_{q=0}^{5} \mathbf{G}_q \tag{28}$$
4. Extract: $(\zeta, \mathbf{v}_\zeta) = \text{DominantEigen}(\mathbf{G}, \text{use\_second} = \text{True})$
5. Update final bias:
$$\mathbf{b}_{\text{final}} \leftarrow \frac{\mathbf{M}_\zeta \mathbf{b}_{\text{initial}}}{\|\mathbf{M}_\zeta \mathbf{b}_{\text{initial}}\|_1} \tag{29}$$
where $\mathbf{M}_\zeta = \zeta \cdot (\tilde{\mathbf{v}}_\zeta \otimes \tilde{\mathbf{v}}_\zeta)$.

6.2 Normal Pattern Flow

For neurons with defined patterns (DATA_INPUT, ACTION_ELEMENT, etc.):

$$\boxed{\text{Normal: } \alpha \to \mathbf{b}_{\text{initial}}, \quad \beta \to \mathbf{B}, \quad \zeta \to \mathbf{b}_{\text{final}}} \tag{30}$$

Phase 1: α Update Same as UNKNOWN (updates $\mathbf{b}_{\text{initial}}$).

Phase 2-3: Skip γ Normal patterns skip γ (no position initialization needed).

Phase 4: β Update (Position Confirmation)

1. Construct observation matrix $\mathbf{D} \in \mathbb{R}^{5\times5}$ via true similarity between expected and observed positions
2. Compute:
$$\hat{\mathbf{B}} = \mathbf{D}\mathbf{B}, \quad \mathbf{B}^* = \text{RowNormalize}(\hat{\mathbf{B}}) \tag{31}$$
3. Extract: $(\beta, \mathbf{v}_\beta) = \text{DominantEigen}(\mathbf{B}^*)$
4. Update:
$$\mathbf{B} \leftarrow \text{RowNormalize}(\mathbf{M}_\beta \mathbf{B}) \tag{32}$$

Phase 5: ζ Update Same as UNKNOWN (updates $\mathbf{b}_{\text{final}}$).

7 Pattern Decision Logic

7.1 Confidence Check

After computing $\mathbf{b}_{\text{final}}$, we check if the neuron should enter RECYCLING or continue learning:

Algorithm 2 Phase 5 Confidence Decision

```
1: $p_{\text{current}} \leftarrow \mathbf{b}_{\text{final}}[i_{\text{current}}]$
2: $i_{\text{dominant}} \leftarrow \arg\max_i \mathbf{b}_{\text{final}}[i]$
3: $p_{\text{dominant}} \leftarrow \mathbf{b}_{\text{final}}[i_{\text{dominant}}]$
4: if $i_{\text{dominant}} = i_{\text{current}}$ and $p_{\text{current}} \geq 0.7$ and cycle $\geq 3$ then
5:     Enter RECYCLING mode
6: else
7:     Continue to $T_\zeta$ (tensor fallback)
8: end if
```

7.2 Pattern Switching

During tensor fallback, if a different pattern is dominant:

$$\text{Switch if: } p_{\text{dominant}} > p_{\text{current}} + 0.05 \tag{33}$$

7.3 Oscillation Detection

To prevent unstable switching:

Algorithm 3 Oscillation Detection and Reset

```
1: Track all pattern switches in history $H = \{(t_i, p_{\text{from},i}, p_{\text{to},i})\}$
2: if $|H| \geq 5$ then
3:     Examine last 5 switches: $H[-5:]$
4:     $\Delta_{\max} \leftarrow \max_i(t_{i+1} - t_i)$                ▷ Max pattern duration
5:     if $\Delta_{\max} \leq 3$ then
6:         Oscillation detected!
7:         $p^* \leftarrow \arg\max_p \sum_i ⊮[p_{\text{from},i} = p]$              ▷ Most common
8:         Switch to $p^*$, ignoring bias distribution
9:         Reset switch counter
10:    end if
11: end if
```

8 Worked Example

Consider a neuron starting as UNKNOWN observing a text input field.

8.1 Initial State

$$\mathbf{b}_{\text{initial}} = [0.2, 0.2, 0.2, 0.2, 0.2] \quad \text{(uniform)} \tag{34}$$

$$\mathbf{B} = \frac{1}{5}\mathbf{1}_{5\times 5} + \mathcal{N}(0, 0.01) \quad \text{(slightly noisy uniform)} \tag{35}$$

8.2 Cycle 1: α Update

Observe self element and extract eigenvalue:

$$\alpha = 0.081, \quad \mathbf{v}_\alpha = [-0.75, 0.41, 0.09, -0.24, 0.46] \tag{36}$$

Apply softmax and update:

$$\tilde{\mathbf{v}}_\alpha = [0.04, 0.13, 0.09, 0.07, 0.14] \tag{37}$$

$$\mathbf{b}_{\text{initial}} \leftarrow \text{normalize}(\mathbf{M}_\alpha \mathbf{b}_{\text{initial}}) = [0.087, 0.276, 0.200, 0.145, 0.292] \tag{38}$$

Notice shift toward UNKNOWN (index 4) and ACTION_ELEMENT (index 1).

8.3 Cycle 3: ζ Update

Build observation tensor and compute position-wise similarities, then average:

$$\mathbf{G} = \frac{1}{6}\sum_{q=0}^{5} \mathbf{G}_q \tag{39}$$

Extract and update:

$$\zeta = 0.078, \quad \mathbf{b}_{\text{final}} = [0.42, 0.21, 0.15, 0.12, 0.10] \tag{40}$$

DATA_INPUT (index 0) is now dominant! Switch pattern to DATA_INPUT.

9 Conclusion

This mathematical framework provides a complete specification of the neuron learning system. The eigenvalue-guided updates with softmax normalization ensure smooth convergence while maintaining probabilistic constraints. The distinction between UNKNOWN and normal patterns allows for both exploration (via γ) and exploitation (via β), culminating in robust pattern classification via tensor similarity analysis.