frontiers

Supplementary Material: A processing-oriented investigation of inflectional complexity

APPENDIX

TEMPORAL SELF-ORGANIZING MAPS

A *TSOM* consists of a D-dimensional input vector, and a map composed by N nodes (Figure 1). The nodes, topologically organized in a two-dimensional Euclidean space, are fully connected to the input vector through the spatial connection layer ($W = \{w_{i,j}\}$, $i = 1...N$, $j = 1...D$) and to all the nodes of the map through the temporal connection layer ($M = \{m_{i,h}\}$, $i = 1...N$, $h = 1...N$).

Short-term dynamics: recoding

A time series Q of q symbols is input to a *TSOM* as a sequence $Q = \{X(1), X(2), ..., X(q)\}$ where $X(t)$ represents the symbol shown at time t, encoded as an input vector with D components.

In *recoding*, the input vector $X(t) = \{x_1(t), ..., x_D(t)\}$ is propagated across the map. As a result, the unnormalized activation level of the map's i-th node at time t is calculated as:

$$y_{u,i}(t) = \alpha \cdot y_{S,i}(t) + (1 - \alpha) \cdot y_{T,i}(t) \quad i = 1...N \tag{S1}$$

where α and $(1 - \alpha)$ weigh up the respective contribution of the spatial (S) and temporal (T) layers, and

$$y_{S,i}(t) = 1 - RMSE_{S,i}(t) = 1 - \sqrt{\frac{1}{D}\sum_{j=1}^{D}\left[x_j(t) - w_{i,j}(t)\right]^2} \tag{S2}$$

is the Euclidean proximity (scaled in the range $[0; 1]$) between the input vector $X(t)$ and the spatial weight vector associated with the i-th node, and

$$y_{T,i}(t) = \sum_{h=1}^{N}\left[m_{i,h}(t) \cdot y_h(t-1)\right] \tag{S3}$$

is the weighted temporal pre-activation of the i-th node at time t prompted by the state of activation of all N nodes of the map at time $t - 1$. Note that $Y(0) = 0_{N\times 1}$, meaning that no temporal context is used in equation (S3) upon recoding the first symbol of the time series.

The element $w_{i,j}$ is the weight of the connection from the j-th element of the input vector to the i-th node, and $m_{i,h}$ is the weight of the temporal connection from the h-th node to the i-th node. The elements of the input vector X and of the weight matrices W and M are in the range $[0; 1]$.

The resulting unnormalized activation level of the map can be rewritten in matrix notation as:

$$Y_u(t) = \alpha \cdot \left[1 - \frac{1}{\sqrt{D}} \left\| 1_{N\times 1} \times X(t) - W(t) \right\|\right] + (1-\alpha) \cdot [M(t) \times Y(t-1)] \tag{S4}$$

This overall map activation pattern at time *t* (or $MAP(t)$) thus reflects the activation of nodes best fitting (i) the given input stimulus X (equation (S2)), and (ii) the current temporal context (equation (S3)).

The Best Matching Unit (*BMU*) at time t is defined as the node with the maximum activation level:

$$BMU(t) = \underset{i}{\operatorname{argmax}} \{y_{u,i}(t)\} \tag{S5}$$

The overall activation pattern $Y = \{y_1, ..., y_N\}$ is calculated in its normalized form (i.e. with values in the range $[0; 1]$) to ensure network stability over time:

$$Y(t) = \frac{Y_u(t)}{y_{u,BMU}(t)} \tag{S6}$$

Long-term dynamics: learning

In learning, discriminative rules are applied to both the topological and spatial connection layers so that nodes that are highly responsive to a stimulus in context will be even more responsive to the same stimulus as training goes on. Conversely, nodes weakly responsive to a stimulus will be even less responsive to it.

Topological learning

The spatial connections of a node that is highly responsive to an input stimulus are modified to make the node more responsive to the same stimulus. This modification is also propagated to the neighbor nodes. In the model, this effect is taken into account by a neighborhood function centered in *BMU*. Nodes that lie close to *BMU* on the map are strengthened (i.e. their connection vector W_i is modified to be closer to the input vector X) as a function of *BMU*'s neighborhood. The distance between *BMU* and the i-th node on the map is calculated through the following Euclidean metrics:

$$d_i(t) = \sqrt{\sum_{k=1}^{2} [i_k - BMU_k(t)]^2} \tag{S7}$$

where k indexes the k-th topological dimension in a two-dimensional space. For the spatial connection layer, the topological neighborhood function of the i-th node is defined as a Gaussian function, centred on the *BMU*, with a cut-off threshold:

$$c_{S,i}(t) = \begin{cases} e^{-\frac{d_i^2(t)}{2\sigma_S^2(t_E)}} & if\ d_i(t) \leq \nu_S(t_E) \\ 0 & otherwise \end{cases} \tag{S8}$$

where $\sigma_S(t_E)$ is the topological neighborhood shape coefficient for the spatial connection layer at epoch time t_E (i.e. the variance of the Gaussian centered in *BMU*), and $\nu_S(t_E)$ is the topological neighborhood

cut-off coefficient (i.e. the learning radius) for the spatial connection layer at epoch time t_E (i.e. the radius of influence of the Gaussian).

The weight on the j-th topological connection of the i-th node at time $t+1$ and epoch t_E is finally modified as follows:

$$w_{i,j}(t+1) = w_{i,j}(t) + \Delta w_{i,j}(t) \tag{S9}$$

$$\Delta w_{i,j}(t) = \gamma_S(t_E) \cdot c_{S,i}(t) \cdot [x_j(t) - w_{i,j}(t)] \qquad \begin{matrix} i = 1...N \\ j = 1...D \end{matrix} \tag{S10}$$

where $\gamma_S(t_E)$ is the spatial learning rate at epoch t_E.

Temporal learning

Temporal connections are modified by rewarding synchronous node activity and penalizing unrelated node activity according to:

$$m_{i,h}(t+1) = m_{i,h}(t) + \Delta m_{i,h}(t) \tag{S11}$$

As with topological learning, weight modification propagates from *BMU* to its neighbor nodes. Temporal connections from $BMU(t-1)$ to the neighborhood of $BMU(t)$ are strengthened:

$$\Delta m_{i,h}(t) = \gamma_T(t_E) \cdot c_{T,i}(t) \cdot \big[1 - m_{i,h}(t) + \delta_T(t_E)\big] \qquad \begin{matrix} h = BMU(t-1) \\ i = 1...N \end{matrix} \tag{S12}$$

and temporal connections from all nodes but $BMU(t-1)$ to $BMU(t)$ neighborhood are depressed:

$$\Delta m_{i,h}(t) = \gamma_T(t_E) \cdot c_{T,i}(t) \cdot \big[0 - m_{i,h}(t) - \delta_T(t_E)\big] \qquad \begin{matrix} h \neq BMU(t-1) \\ i = 1...N \end{matrix} \tag{S13}$$

where $\gamma_T(t_E)$ is the temporal learning rate at epoch t_E. The offset δ_T, which has usually a very small value, emphasizes the strength modification thus leading to saturation for connections with very high and very low strength value[1].

For the temporal layer, the topological neighborhood function of the i-th node is defined as a Gaussian function, centered on the *BMU*, with a cut-off threshold:

$$c_{T,i}(t) = \begin{cases} e^{-\frac{d_i^2(t)}{2\sigma_T^2(t_E)}} & if\ d_i(t) \leq \nu_T(t_E) \\ 0 & otherwise \end{cases} \tag{S14}$$

where $\sigma_T(t_E)$ is the topological neighborhood shape coefficient for the spatial connection layer at epoch time t_E (i.e. the variance of the Gaussian centered in *BMU*), and $\nu_T(t_E)$ is the topological neighborhood

[1] Apart from the δ_T coefficient, equation (S12) and equation (S13) are formally identical to the Rescorla-Wagner equations.

cut-off coefficient (i.e. the learning radius) for the temporal connection layer at epoch time t_E (i.e. the radius of influence of the Gaussian).

Learning decay

At the end of each learning epoch, an exponential decay process applies to each learning parameter so that the generic parameter p at epoch t_E is calculated according to the following equation:

$$p(t_E) = p(0) \cdot e^{-\frac{t_E}{\tau_p}} \tag{S15}$$

The time-constant τ_p can be interpreted as the number of epochs after which the generic parameter p has a value which is equal to 63% of the initial value $p(0)$. For computational efficiency, the previous equation is approximated on the basis of the value at the previous learning epoch:

$$p(t_E) = p(t_E - 1) \cdot \left(1 - \frac{1}{\tau_p}\right) \tag{S16}$$

Evaluating the map

Map labeling

After training, a label L_i is assigned to the i-th node of the map. L_i corresponds to the symbol X_c that best fits the i-th node connection weights on the spatial layer:

$$L_i = \underset{c}{\operatorname{argmin}} \left\{ \sqrt{\frac{1}{D} \sum_{j=1}^{D} \left[x_{c,j}(t) - w_{i,j}(t)\right]^2} \right\} \quad i = 1...N \tag{S17}$$

Labels are used to test the ability of a *TSOM* to recode the input sequence Q correctly.

Recoding

The recoding accuracy of an input sequence Q of q symbols $\{X(1), X(2), ..., X(q)\}$ is calculated by checking, at each time tick t, that $L_{BMU(t)}$=$X(t)$. The overall recoding score for the entire input sequence can be calculated as a function of the recoding scores of each symbol.

Prediction

Upon presentation of each input letter, the *TSOM* is prompted to complete the currently input sequence by guessing its possible continuation. A *TSOM* can predict a progressively presented input sequence by propagating the activation of the current *BMU*(*t*) through its forward temporal connections, to predict $L_{BMU(t+1)}$:

$$BMU(t+1) = \underset{i=1...N}{\operatorname{argmax}} \left\{ y_{T,i}(t+1) \right\} = \underset{i=1...N}{\operatorname{argmax}} \left\{ \sum_{h=1}^{N} \left[m_{i,h} \cdot y_h(t) \right] \right\} \tag{S18}$$

If the upcoming symbol is correctly guessed, then a prediction score 1 is assigned to the symbol, and the *TSOM* is prompted again to guess one more symbol. The prediction score is increased by 1 for each ensuing correctly guessed symbol, and it is set back to 0 at each failure. This is a way to assess how quickly the

current input sequence can be uniquely accessed. The overall prediction score for the entire input sequence can be calculated as a function of the prediction scores assigned to each of its symbols.

Recall

In recoding a time series Q of q symbols, the activation pattern $Y(t)$ accumulates in a short-term buffer to yield an *Integrated Activation Pattern* (*IAP*) $\hat{Y}$:

$$\hat{y}_i = \max_{t=1...q} \{y_i(t)\} \quad i = 1...N \tag{S19}$$

The time series Q is thus associated with *IAP*, where no explicit timing information is provided. Recall is modeled as the task of restoring the input sequence, by priming the map with $X(1)$, and then replacing $y_{S,i}(t)$ with $\hat{y}_i$ in equation (S1) for $t > 1$:

$$y_{S,i}(t) = \begin{cases} 1 - \sqrt{\frac{1}{D}\sum_{j=1}^{D}[x_j(t) - w_{i,j}(t)]^2} & t = 1 \\ \hat{y}_i & t = 2...q \end{cases} \qquad i = 1...N \tag{S20}$$

As a result, during recall the equation (S1) for $t = 2...q$ becomes:

$$y_{u,i}(t) = \alpha \cdot \hat{y}_i + (1 - \alpha) \cdot y_{T,i}(t) \qquad \begin{matrix} i = 1...N \\ t = 2...q \end{matrix} \tag{S21}$$

As with recoding, given a Q and its associated *IAP* we can check, at each time tick t, that $L_{BMU(t)} = X(t) \in Q$. The overall recall accuracy for *IAP* is then calculated as a function of the recall scores of each $X(t) \in Q$.