Space-to-time-conversion: Oscillations in an artificial neural network generate a temporal code representing simultaneous visual inputs

Deep Neural Networks applied in computer vision are often compared to the hierarchical architecture of the ventral stream; but they typically disregard the temporal dynamics experimentally observed in the visual system. For instance, alpha oscillations (8-12 Hz) dominate the temporal dynamics of the human visual system; however, oscillations are rarely considered in artificial neural networks (ANNs). Here, we propose a fully connected network that embraces oscillatory dynamics, to convert spatial information into a temporal code; organizing multiple objects in time. The network was trained end-to-end to classify three letters in four quadrants. We then added semi-realistic temporal dynamics to the hidden layer, based on differential equations, where we introduced adaptation and a 10 Hz pulsed inhibition mimicking neuronal alpha oscillations. A relaxation term was added to each hidden unit to ensure a pulsed (non-sustained) activation. Attention was implemented by selectively increasing the input of one of the presented letters. The trained network correctly classified individually presented letters but showed high uncertainty when presented with two letters simultaneously, indicating the bottleneck problem. When introducing pulsed inhibition, the output nodes of the presented letters activated sequentially; thus generating a temporal code. Our model provides a novel proof-of-principle for solving the bottleneck problem when multiple objects are presented. This is achieved by generating sequential outputs of the competing objects, organized according to the phase of 10 Hz oscillations. The resulting temporal code suggests a computational benefit for oscillations in ANNs. Future work will expand the network to a deeper architecture, and relate the time-course of the dynamics to electrophysiological observations to further constrain the implementation.

Temporal dynamics

$$\tau_H \frac{dH_j}{dt} = -H_j + \sigma\left(\sum_i W_{ji}^{(I)} I_i - R_j - \alpha(t) + H_j\right) \tag{1}$$

$$\tau_R \frac{dR_j}{dt} = -R_j + aH_j \tag{2}$$

$$O_k = \frac{\exp\left(\sum_j W_{kj}^{(H)} H_j\right)}{\sum_{k'} \exp(\sum_j W_{k'j}^{(H)} H_j\Big)} \tag{3}$$

I: input layer
O: output layer H_j: hidden layer
$W_{ji}^{(I)}$: weight matrix (I $\rightarrow$ H)
$W_{kj}^{(H)}$: weight matrix (H $\rightarrow$ O)
R_j: relaxation term
σ: sigmoid
$\alpha(t)$: 10 Hz sawtooth
Adjustable parameters: $\tau_H = 0.01$; $\tau_R = 0.011$; $a = 20$

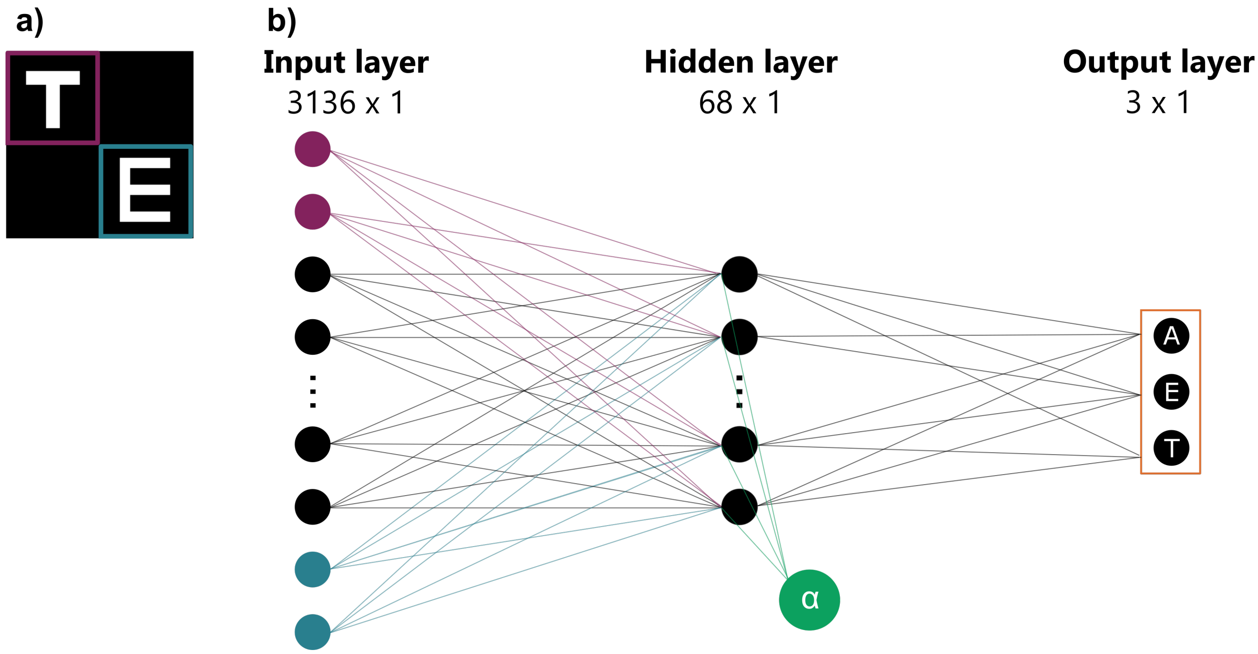

Figure 1: **a** Example of two simultaneously presented letters. **b** A standard feedforward network with one hidden layer was trained to classify the three letters in the four quadrants. After training, we introduced temporal dynamics to the hidden nodes including a relaxation term ('R' in the equation above). α signifies the 10 Hz pulsed inhibition applied to the hidden layer.

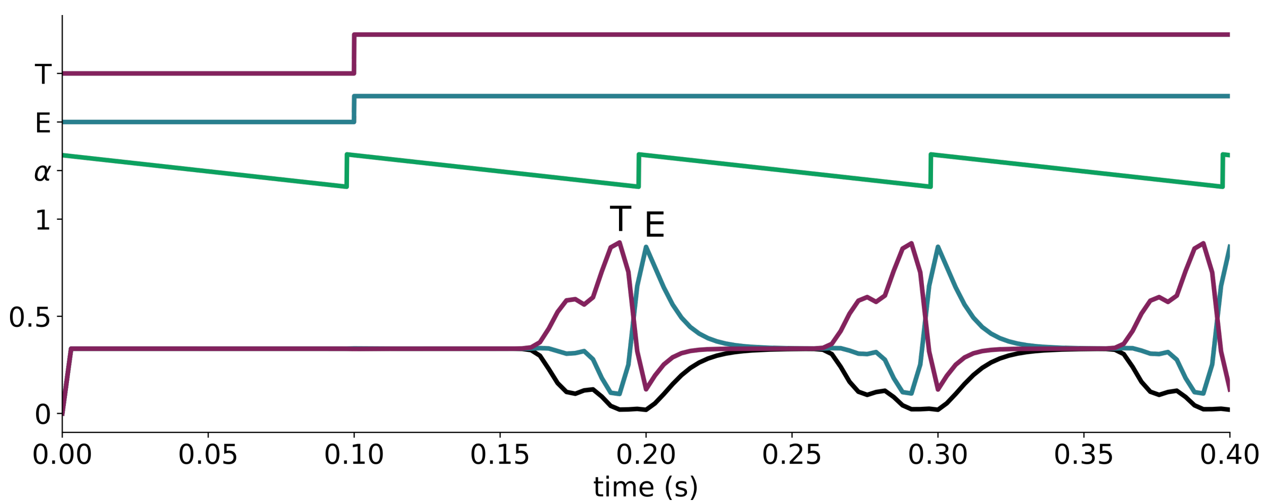

Figure 2: Converting a parallel multi-item input into a temporal code. Top: Example of one attended (T) and unattended input (E) input signal. The pulsed inhibition (α) followed a 10 Hz sawtooth-shape. Bottom: The attended stimulus activates ~60 ms after stimulus onset, followed by the unattended stimulus at ~100 ms. The resulting temporal code emerges once per alpha cycle.