Using serial order to learn and replay a sequence

DFT Summer School 2022

Goal of the Project

The goal of this project is to create a neural mechanism for learning and reproducing sequences. Such a mechanism might be used, for instance, when a behavioral task requires chaining successive motor actions in a fixed order. The project illustrates how continuous neural dynamics can support controlled sequences that progress only once individual sub-tasks have been completed successfully.

Scenario

The scenario used here includes a learning and a replay phase. In the learning phase, a sequence of colors will be presented to the model, and it will learn that sequence. In the replay phase, a scene will be presented to the model, and it will indicate, if it contains the color that was learned for the current ordinal position. It then moves on and looks for the next color in the sequence. The architecture presented here ties into the Visual Search architecture from the previous task, more precisely the *Color Attention* field. Replaying the learned sequence causes the architecture to attend different colors in the learned order. The sequence continues, whenever this color attention leads to a peak in the *Spatial Attention* field.
While we use color in the current scenario to make things more concrete, the space of learned values may in principle be spanned by any perceptual, cognitive, or behavioral dimensions.

Model components

A sketch of the architecture is shown in Figure 1. The explanations in this document should be sufficient to solve the task, but if you are interested in additional information please refer to chapter 14 of our book [1].

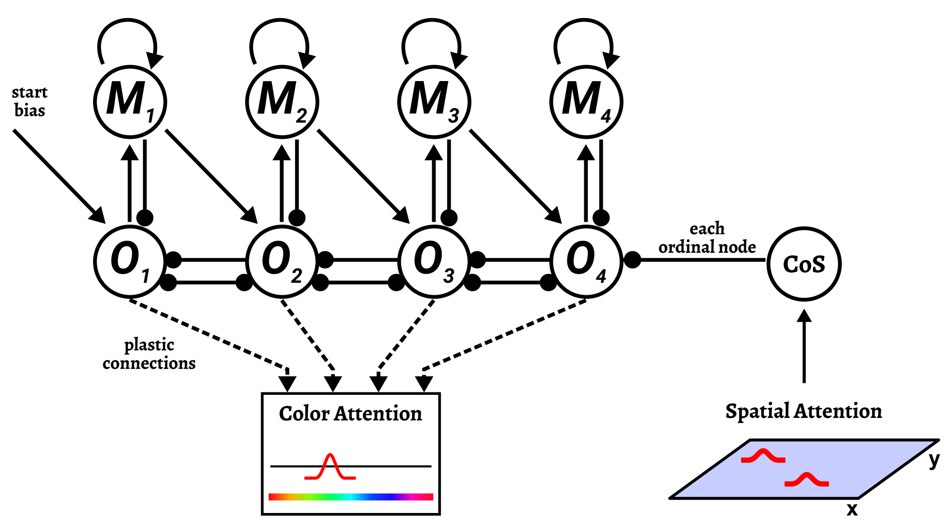

Figure 1: Architecture Sketch. *Color Attention* and *Spatial Attention* are already part of the Visual Search Architecture.

- A peak in the one-dimensional **Color Attention field** causes the architecture to attend objects sharing the represented color. During learning use the *Color Nodes* from the Visual Search architecture to evoke color peaks at different locations. In the replay phase, peaks in the color field will be induced through the plastic synaptic connections.

- The **ordinal nodes** $(O_1, ..., O_4)$ stand for different positions within the learned sequence. During learning or reenacting, these will become active one after the other. Only one can be active at any point in time, which is ensured by inhibitory connections between all of them (not all of which are shown in the figure).

- The **memory nodes** $(M_1, ..., M_4)$ keep track of which of the ordinal nodes have already been active while learning or reenacting a given sequence. At the start of the sequence, all memory nodes are inactive. Only when the corresponding ordinal node becomes active will a memory node be activated through an excitatory connection. The memory nodes are in a self-sustained regime, so that the activated node will stay active for the rest of the sequence. An active memory node does two things. First, it inhibits its ordinal node, but not strongly enough to deactivate it right away. Second, it excites the next ordinal node, but not strongly enough to activate it right away (since that ordinal node is suppressed by inhibition from the currently active one).

- The **plastic connections** are the synaptic patterns from ordinal nodes to the color field that are learned in the learning phase. *Implementation hint:* In CEDAR, use `HebbianConnection` steps to make these connections. This step needs three inputs: from the source neurons (node), from the target neurons (color field), and one "reward" input (yes, this name is not well-chosen). For learning to take place, it is required that source and target neurons are active and the "reward" input must be larger than 0.5. Thus, by connecting one `Boost` step to all "reward" inputs, you can switch between learning mode and replay mode conveniently.

- The **CoS node** has inhibitory connections to all ordinal nodes. When the CoS node is active, all ordinal nodes are suppressed below threshold. In the replay phase it is activated through any peak in the *Spatial Attention* field. To simplify the current model, also add a `Boost` step (*Learning Boost*) with which you can activate that node manually during learning.

Learning and Replaying

Once you have implemented the architecture according to the description and Figure 1, you should be able to learn and then replay a sequence.

Learn a Sequence

1. Enable learning mode by activating the *Learning Boost* to the Hebbian steps' "reward" inputs.

2. During Learning a peak in the *Color Attention* field should have no effect on the Visual Search architecture. We should temporally disable the *Color/Space Attention* field during learning. Do so by connecting the output of the *Learning Boost* with sufficient inhibitory strength to the *Color/Space Attention* field (Use a `StaticGain` in between).

3. Also supply an external `Boost` input to the first ordinal node, activating it and thus also the corresponding memory node.

4. At the same time, activate one of the *Color nodes* to induce a peak.

5. Check that learning has taken place by plotting the Hebbian step.

6. Deactivate the *Color node*.

7. Activate the CoS node through a boost. All ordinal nodes will be suppressed. The first memory node will remain active.
8. Deactivate the boost to the CoS node. Being dis-inhibited, the ordinal nodes can now become active again. The first memory node inhibits the first ordinal node, though, and concurrently pre-activates the second ordinal node, which thus wins the competition and becomes active.
9. Activate a *Color node* again and proceed in the same manner until a synaptic weight pattern has been learned for each ordinal node.

Replay a Sequence

1. Deactivate the inhibitory boost to the *Color/Space Attention* field and the learning boost.
2. Supply an external input to the first ordinal node, activating it and thus also the corresponding memory node.
3. This should induce a peak in the *Color Attention* field through the plastic connections.
4. If at least one object matching the color is present in the scene, a peak should form in the *Spatial Attention* field.
5. Once a peak has formed in the *Spatial Attention* field, the CoS node activates and inhibits the current ordinal node.
6. The model should the autonomously proceed to activating the next ordinal node, once the peak in the *Spatial Attention* field has vanished again due to missing color input.
7. If all colors are present, the architecture will autonomously attend all colors in the learned sequence.
8. If a learned color is not present, consider loading a different source image or adding a different object.

Extending the Architecture

There are multiple ways in which you can extend the project. You are of course welcome to try out your own ideas, but if you look for inspiration, we recommend to integrate one of our other exercises.

- **Reaching Architecture**: In this task you implement an architecture that allows the robotic arm to reach for targets selected in the *Spatial Attention* field. In combination with this project you can build an architecture that autonomously executes a sequence of reaching movements to differently colored targets.
- **Spatial Language Architecture**: In this project you extend the Visual Search architecture to to be able to select objects that match a linguistic description. You might integrate this project with the sequence project by learning a sequence of linguistic phrases rather than a color sequence.

References

[1] Yulia Sandamirskaya. Autonomous sequence generation in dynamic field theory. *G. Schöner, JP Spencer, & T. DFT Research Group (Eds.), Dynamic thinking: A primer on dynamic field theory*, pages 353–368, 2015.