CEDAR Tutorial

Neural Dynamics for Embodied Cognition

KogWis 2022

Get CEDAR

- Please download the precompiled version of CEDAR for your operating system from `https://cedar.ini.rub.de` and extract the zip archive.
- Run Cedar by executing the `cedar.app` or `cedar.bat` file in the main folder.

Exercise: A Field Architecture

We will build a very much simplified version of an architecture for spatial language that is able to select objects based on their spatial position and commit the selected object to working memory. Figure 1 shows a simple diagram of the architecture.

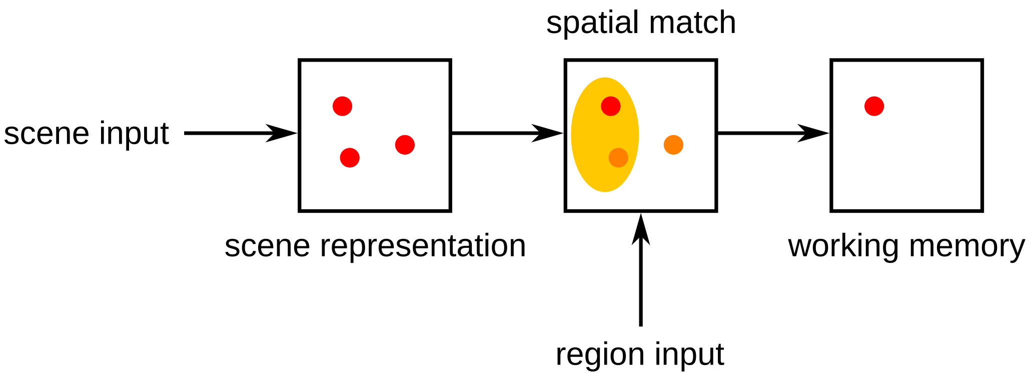

Figure 1: Diagram of the small architecture for spatial language. Peaks are denoted by red circles, and subthreshold bumps by orange circles. The subthreshold region-input is shown as a yellow ellipse, here highlighting the left part of the field.

Creating the Architecture

1. Create a two-dimensional multi-peak neural field that receives input from the simulated input and creates peaks.
   - This will be our representation of a "scene" of multiple objects on a table. We will call this the "scene representation" field.

2. Create another two-dimensional field and call it “spatial match”.
3. Let it receive input from the “scene representation” field, strong enough to only form subthreshold bumps of activation in the “spatial match” field, but not form peaks.
4. Additionally, add input that highlights specific regions within the field.
   - For each region that we want to highlight, for instance, the entire left side of the field, we add a `GaussInput` step as input to the “spatial match” field and set the parameters such that the Gauss function covers the region of the field we want to highlight.
   - Only when the subthreshold bumps (that represent the objects) overlap with these highlighted regions, may the field form peaks.
5. Tune the “spatial match” field to be selective, that is, to only allow for a single peak to form at a time. Once this works, it should naturally select the object that fits the highlighted region best.
6. Create one more two-dimensional multi-peak field, and call it “working memory”.
7. Let it receive input from the “spatial match” field and form a peak whenever there is a peak in that field.
8. Tune the “working memory” field to be in a self-sustained regime such that the peaks remain stable even if the “spatial match” field later creates a peak at a different location.
9. Play with the spatial position of objects in the scene, as well as highlighting different regions within the field.
   - You can do so by turning off the input of all but one of the region-inputs or activate different combinations. You could even implement more complex regions like “central” or “peripheral” by combining different `GaussInput` steps.
   - You can think of the region-input as a “command”, telling the architecture to “select an object on the left side” or “select the central object”.