Interactive CEDAR exercise—Part 1

Sophie Aerdker, Mathis Richter

January 21, 2021

1 Detection of simulated input

Create a two-dimensional field (`NeuralField`). This field may represent a two-dimensional surface, like a table top. Peaks in the field represent objects on that surface.

The field receives three inputs, which we simulate, for now, by 2D Gaussian functions (`GaussInput`) at different positions.

Tune the field so that it forms a stabilized peak at every input location.

2 Camera input & field regimes

We will now replace the simulated input with input from a camera of a 3D simulator. Open the file `cedar_architecture_template_part1.json` that you can download from the course website. Plot the step `Virtual Camera` and get a sense for the scene that is visible there by carefully changing the position parameters of the three `Shape Visualization` steps. The camera image runs through a larger box entitled `Visual preprocessing`, which you do not have to understand in detail. It uses regular image processing algorithms to produce activation from the camera image. Feel free to inspect it if you are interested in understanding this more. The larger box produces two outputs; hover over the outputs to see their names. Run the activation coming from the output called `2d space` through a `StaticGain` step, which enables you to vary the overall strength of the activation; then connect the output of that step to the field that you have created before.

First, tune the field to create a peak for every object in the scene. Once that works, try (slowly) moving the objects in the scene and observe whether the peaks follow the moving input. This may be a little tricky with the simulated environment we are using this year...

Second, create a copy of the field (`Ctrl+D`) which then also receives input from the preprocessing box. Tune this field to have working memory about objects in the scene. That is, once peaks have formed, you should be able to quickly remove an object without the corresponding peak decaying. You can "remove" objects by deleting the connection to its corresponding `Shape Visualization` step. Once that works, try again to slowly move an object in the scene. The working memory peak should follow the moving input.

Third, create another copy of the field which then also receives input from the preprocessing box. Tune this field to make a selection decision, that is, it should only form a peak at a single position, even when multiple objects are in the scene. Make sure that the field is not in a working memory regime, that is, the peak should disappear when you remove the object from the scene. Play with adding and removing objects to and from the scene. At the position of which object does the peak form? Is it always the same object? Why do you think that is? Can you observe hysteresis in the field?

3 Projections

We will now tune a field to make a selection decision for an object of a certain color. We will need a field that represents both the spatial position as well as the color of the scene, that is, a three-dimensional field. Create a field like this with the size 50x50x8 (50x50 for the spatial dimensions of the image, 8 for the color dimension). The field should receive input from the other output of the preprocessing box called `3d color space`. Adding a `StaticGain` step in between will be beneficial for later use.

To represent colors that the model is to attend, we first need a one-dimensional field (the color field; size 8), which spans the color-dimension. This color field will receive a Gaussian input (`GaussInput`) (size 8) and create a peak based on that input. Now, using a `Projection` step and a `StaticGain` step, connect the color field to the three-dimensional field. When this is working, you should only see a thin layer of the three-dimensional field being affected by input from the color field. Why is that? Why is the thin layer of increased activation at this position in the three-dimensional field and not somewhere else? Make sure you understand exactly how the two fields are connected and what the `Projection` step does! It helps to observe what happens in the plots of the field activation and output when you manipulate the scene (by moving, removing or re-adding objects) and the Gaussian input to the color field.

Once you have understood the connectivity between the color field and the three-dimensional field, choose the position of the Gaussian input so that it specifies a color you would like the model to pick. Tune the strength of the input and the parameters of the three-dimensional field so that it only creates a peak when input from the camera overlaps with input from the color field. Play with the Gaussian input to the color field and select different objects in the scene.

Create a new two-dimensional field (size 50x50) and, using both a `Projection` step and a `StaticGain` step, connect the output of the three-dimensional field to the new two-dimensional field. For the projection, choose the mode `Maximum`. Set up the connection strength so that there is a peak in the two-dimensional field whenever there is a peak in the three-dimensional field. What does the `Projection` step do in this case?

The two-dimensional field can be interpreted as the spatial attention of the small model we created. You could also create a peak here from a Gaussian input and select certain positions in the scene. The one-dimensional field can be interpreted as a feature attention (color) of the model. The three-dimensional field binds the features of color and space together.