

tangible dreams

by Cédric Colas

Exploring visual worlds through physical neural networks

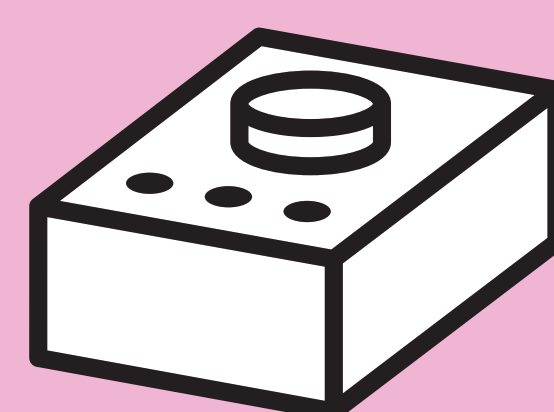
An interactive neural network for creating images. By plugging in cables, turning knobs, and flipping switches, you shape how the network transforms pixel coordinates into colors—sculpting complex visuals in real time.

How does it work?

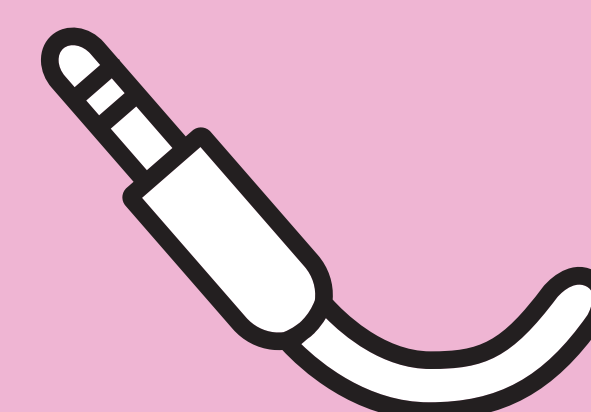
composing transformations

Nodes carry out simple transformations which are then composed into more complex ones following the wiring of the network. Reshape the network's structure by connecting jack cables between nodes, from **out** to **in** (never *out* to *out*). Make sure there exists a path between a *coordinate* node and an *output* node.

components



nodes (neurons)



connections



parameters

coordinates

Coordinate nodes produce information about the location of pixels in the canvas.

X and Y

give the raw horizontal/vertical coordinates. You can **invert** them with the toggle switch, **zoom**, or **shift** the scale.

Source

lets you switch between different coordinate frames (distance to middle axe, distance to center, angle, coordinate in grid cell, spiral).

Use **mod 1** and **mod 2** to play with parameters specific to each coordinate system.

transformation

Transform nodes and color nodes each perform a simple local function:
$$\text{out} = \text{shape}((\text{in1} \times \text{w1} + \text{in2} \times \text{w2} + \text{in3} \times \text{w3}) \times \text{scale})$$

in are signals received from other nodes connected via jack cables.

weights are weights set by the knobs controlling how much the corresponding signals should influence the output.

scale is a general amplification.

shape is a mathematical “curve” you can choose via the switch (sigmoid, sine, relu, gaussian, modulo, fractal, structured noise).

CV makes **weight 2** controlling **in 2** time-reactive, modulated by **weight 1**.

colors

Red, Green, and Blue nodes output the final red, green, blue values for all pixels in the canvas.

CV makes **weight 2** controlling **in 2** time-reactive, modulated by **weight 1**.

watch out!

- 1 Start with the small knobs to their default position (center/up) and explore from there.
- 2 Never connect outputs together. Only go from an *out* to an *in*.
- 3 Be gentle!

something's wrong?

- 1 Make sure there's at least one path from an input node to an output node.
- 2 Glitching? a source, or activation selector might be set between two choices.
- 3 Still not working?
Text me at 857-396-3602
or email me at ccolas@mit.edu



Learn more about the project, and see patterns saved by visitors

Experiment, collaborate, and see what dreams you can create.