

The Jets and the Sharks

Here are the user-interface commands for our network:

- `show()` prints out all units with positive activations
- `clamp(unit)` clamps the specified unit
- `unclamp(unit)` unclamps the specified unit
- `reset()` unclamps all units and resets their activations to the resting level
- `run(n)` runs the network for n cycles

For example, we can turn on external input to the unit for Ken's name and then let activation spread through the network for 100 time steps by typing the following commands:

```
>>> reset()
>>> clamp(nameKen)
>>> run(100)
```

The activation of each unit gets updated in the graphical display, with green indicating positive activation and red indicating negative. Typing the `show()` command will print out the numerical activation values of the positively-activated units:

```
>>> show()
Sharks      0.5065
in20s       0.3782
in30s       0.0051
H.S.        0.5244
Single      0.5065
Burglar     0.3783
Ken         0.8064
(pete)      0.0382
(fred)      0.0382
(nick)      0.2383
(ken)       0.6832
(earl)      0.0382
(rick)      0.0384
(neal)      0.2383
```

- **Retrieving an individual from their name.** After performing the above test, why are some of Ken's properties more strongly activated than others? None of the other name units are activated, yet several other instance units are active. Why is this?
- **Retrieval from a partial description.** Ken can be uniquely described by two properties: his gang and his age. Reset the network, then clamp the units `sharks` and `in20s`. Run the network for 100 cycles. What are the main differences between this state and the state shown above? Why do the occupation units show partial activations of units other than Ken's occupation (Burglar)?

(continued on back)

- **Graceful degradation.** To investigate the effect of erroneous information in the memory probe supplied to the network, first run the network with all of Ken's properties other than his name clamped. Then try it again with `juniorhigh` clamped in place of `highschool`. How well does the model do with the "noisy" version of Ken compared to the correct version? Would it do this well with all noisy versions of individuals? Test with at least one other individual, and explain what happens.
- **Default assignment.** Sometimes we don't know something about an individual; for example, we may never have been exposed to the fact that Lance is a burglar. Yet we are able to give plausible guesses about such missing information. The network can do this too. To see this, we will erase the network's explicit knowledge that Lance is a burglar by disabling the connections between the instance unit for Lance and the Burglar unit. But first we need a way to temporarily disable connections between units.

Add methods to the Connection class called `disable` and `restore`, both of which take no arguments. `Disable` should set the strength of the connection to 0; `restore` should set it back to its original value.

Add methods to the Unit class called `disableConnectionFrom` and `restoreConnectionFrom`, both of which take another unit `u` as an argument. These methods should scan the list of incoming connections for one whose from-unit matches `u`, and then either disable or restore the connection.

For convenience, define the user interface commands `cut(unit1, unit2)` and `restore(unit1, unit2)`, to make it easy to disable or restore the bi-directional connections between `unit1` and `unit2`.

Now run the network for 100 cycles with Lance's name unit (`nameLance`) clamped, to see what happens with all connections intact. Next, reset the network, disable the connections between Lance's instance unit (`instLance`) and the `burglar` unit, and run the network again for 100 cycles with `nameLance` clamped. Was the network able to fill in Lance's occupation correctly? What is the resulting activation level of the `burglar` unit in each case?

- **Spontaneous generalization.** Now let's consider the network's ability to retrieve appropriate generalizations over sets of individuals—that is, its ability to answer questions like "What are Jets like?" or "What are people who are in their 20s and have only a junior high education like?" (Note: Be sure to restore the connections between `burglar` and `instLance` before going on.) Ask the network to generalize about the Jets by clamping the `jets` unit and cycling for 100 steps. What is the general profile of a Jets gang member, according to the network? Next, ask it to generalize about people in their 20s with a junior high education by clamping `in20s` and `juniorhigh`. What can be said about the other properties of this type of person, in general?