

Homework-NeuralNet-A

February 12, 2024

1 Homework - Neural networks - Part A (35 points)

1.1 Interactive activation and competition

by *Brenden Lake* and *Todd Gureckis*

Computational Cognitive Modeling

NYU class webpage: <https://brendenlake.github.io/CCM-site/>

This homework is due before midnight on Feb. 15, 2024.

Note: Please complete the responses to these questions as a markdown cell inserted beneath the question prompts.

In this assignment, you will get hands on experience with a classic neural network model of memory known as the Interactive Activation and Competition (IAC) model. We will go through a series of exercises that will stretch your understanding of the IAC model in various ways. The exercises below examine how the mechanisms of interactive activation and competition can be used to illustrate two key properties of human memory: - Retrieval by name and by content. - Spontaneous generalization over a set of familiar items.

These exercises are from Chapter 2 of the online PDP Handbook by James McClelland.

You should review the slides from lecture and read Section 2.1 of the PDP Handbook before continuing. This has important background and technical details on how the IAC model works. The IAC model instantiates knowledge that someone may have from watching the 1960s musical “West Side Story,” where two gangs the “Jets” and “Sharks” struggle for neighborhood control in Manhattan. The “database” for this exercise is the Jets and Sharks data base shown in Figure 1, which has the central characters from the two gangs. You are to use the IAC model in conjunction with this data base to run illustrative simulations of these basic properties of memory.

Figure 1: Characteristics of a number of individuals belonging to two gangs, the Jets and the Sharks. (From “Retrieving General and Specific Knowledge From Stored Knowledge of Specifics” by J. L. McClelland, 1981, Proceedings of the Third Annual Conference of the Cognitive Science Society.)

1.2 Software and architecture

We will be using IAC software from Axel Cleeremans which you can download [here for Mac OS](#) and [here for Windows](#).

Figure 2: Screen shot from Cleeremans’ IAC software. Units are organized into 7 groups. For illustration here, all groups have a different color background, while the group of visible name units

have no background.

Upon downloading and loading the software, you will see a display that looks like Figure 2. The units are grouped into seven pools: a pool of *visible* name units, a pool of *gang* units, a pool of *age* units, a pool of *education* units, a pool of *marital status* units, a pool of *occupation* units, and a pool of *hidden* units. The name pool contains a unit for the name of each person; the gang pool contains a unit for each of the gangs the people are members of (Jets and Sharks); the age pool contains a unit for each age range; and so on. Finally, the *hidden* pool contains an instance unit for each individual in the set.

The units in the first six pools can be called visible units, since all are assumed to be accessible from outside the network. Those in the gang, age, education, marital status, and occupation pools can also be called property units. The instance units are assumed to be inaccessible, so they can be called hidden units.

Figure 3: The units and connections for some of the individuals in Figure 1. The arrows represent excitatory connections. The outlined groups of units have mutually inhibitory connections (not shown). (From “Retrieving General and Specific Knowledge From Stored Knowledge of Specifics” by J. L. McClelland, 1981, Proceedings of the Third Annual Conference of the Cognitive Science Society.)

Each unit has an inhibitory connection to every other unit in the same pool. In addition, there are two-way excitatory connections between each instance unit and the units for its properties, as illustrated in Figure 3. Note that the figure is incomplete, in that only some of the name and instance units are shown. These names are given only for the convenience of the user, of course; all actual computation in the network occurs only by way of the connections. You can also view the different connections using the IAC software by hovering your mouse over a particular unit (Figure 4).

Figure 4: You can view the connections to a unit by placing your mouse over it. Green connections are excitatory and red connections are inhibitory.

Since everything is set up for you, you are now ready to do each of the separate parts of the exercise. Each part is accomplished by using the interactive activation and competition process to do pattern completion, given some probe that is presented to the network. For example, to retrieve an individual’s properties from his name, you simply provide external input to his name unit, then allow the IAC network to propagate activation first to the name unit, then from there to the instance units, and from there to the units for the properties of the instance.

1.3 Exercise: Retrieving an individual from his name

To illustrate retrieval of the properties of an individual from his name, we will use Ken as our example. Make sure the simulation is paused (press SPACE) and press ‘r’ to reset it. Set the external input of Ken’s name unit to 1 by clicking on the name unit (not the hidden unit!). The circle’s background should turn bright green to represent the external input.

A unit’s activity level can be visualized by the colored dot, where yellow dots are positive activation and red dots are negative activation. The larger the yellow dot, the stronger the activation. A unit’s precise activity level can be examined by rolling the mouse over the unit.

Figure 5: The display screen after about 150 cycles with external input to the name unit for Ken.

Press SPACE to unpause and allow the network to run for approximately 150 cycles (the cycle counter is in the bottom left of the panel). The simulation runs quickly, so be sure to pause at about 150 cycles exactly! A picture of the screen after 150 cycles is shown in Figure 5. At this point, you can check to see that the model has indeed retrieved the pattern for Ken correctly. There are also several other things going on that are worth understanding. Answer all of the following questions below regarding the network at this state (you'll have to refer to the properties of the individuals, as given in Figure 1).

Problem 1 (10 points) None of the visible name units other than Ken were activated, yet a few other hidden instance units are active (i.e., their activation is greater than 0). Explain why these units are active. Keep your response short (about 3 sentences).

1.3.1 Problem 1:

After running approximately 150 cycles, we observe partial activations in the hidden instance units for Nick, Neal, Rick, Earl, Pete, and Fred (in addition to the hidden Ken unit) and can see that each of these nodes shares at least three mutual connections with the Ken unit (leading all of their net inputs to be positive). Each of these partial activations can be attributed to the model's activation dynamics, which are highly variable depending on the sign of the net input: If Positive: $\Delta a_i = (\max - a_i) * \text{net}_i - \text{decay} * (a_i - \text{rest})$ If Negative: $\Delta a_i = (a_i - \min) * \text{net}_i - \text{decay} * (a_i - \text{rest})$

Since all of the net inputs for the above-mentioned hidden units were positive, the activation is more likely to increase towards the maximum regardless of how small the initial net inputs were, exemplifying the "rich get richer" effect.

Problem 2 (10 points) Some of Ken's properties are activated more strongly than others. Why? Keep your response short (about 3 sentences).

1.3.2 Problem 2

Comparing the in20s unit with the HS unit was interesting, as both had the same number of excitatory and inhibitory connections, but the HS unit became more strongly activated (0.522) at this point than the in20s unit (0.395). The excitatory connections to the HS unit may simply have greater weights that lead to more activation, however, I believe that the main reason the HS unit has the strongest activation at this point is because it started with the greatest net input, again exemplifying the "rich get richer" effect.

1.4 Retrieval from a partial description

Next, we will use the IAC software to illustrate how it can retrieve an instance from a partial description of its properties. We will continue to use Ken, who, as it happens, can be uniquely described by two properties, Shark and in20s. Reset the network ('r') and make sure everything is paused and that all units have input of 0. Click to set the external input of the Sharks unit and the in20s unit to 1.00. Run a total of 150 cycles again, and take a look at the state of the network.

Of all of the visible name units, Ken's name should be the most active. Compare the state of the network's with the a screen shot of the previous network state when activating Ken's name directly, such as that in Figure 5.

Problem 3 (10 points) Explain why the occupation units show partial activations of units other than Ken's occupation, which is Burglar. While being succinct, try to get to the bottom of this, and contrast the current case with the previous case. Keep your response short (about 3 sentences).

1.4.1 Problem 3

Unlike our previous case, here we examine how activation spreads throughout the network when we start with 2 separate property nodes as opposed to starting with Ken's specific node. Activation spreads through the network via both direct and indirect connections, thus, even though there are no Sharks that are Bookies or Pushers in their 20s, the distributed effects of the IAC model can cause these other occupational nodes to become partially activated.

1.5 Spontaneous generalization

Now we 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?” Reset (‘r’) the network. Make sure all units have input of 0 and none are highlighted green.

Set the external input of Jets to 1.00 by clicking on it. Run the network for 150 cycles and observe what happens.

Problem 4 (5 points) Given the network's state, what can you infer about a typical Jet? (1-2 sentences is plenty).

1.5.1 Problem 4

After running about 150 cycles, we can see that most Jets are single, in their 20s, and were educated up to JH. It also seems as though they are evenly distributed in terms of their occupations.