Assignment 6: Alphabet-Arithmetic in Nengo

This assignment should be done in pairs.

Goal

The goal of this assignment is to explore a more advanced Nengo model, which relies on semantic pointers. You're asked to complete an Alphabet-Arithmetic-retrieval model (assuming the facts are already learned), and afterwards reflect on it and compare it to the ACT-R model you developed in Assignment 3.

Assignment Part 1: Exploring and completing the model.

Start by doing Nengo tutorials 19-22, which explain the use of the semantic pointer architecture (spa) and the basal ganglia. See also the lecture slides of lecture 10.

Download the attached zip-file. It contains the Nengo model, `zbrodoff_retrieve.py` & `zbrodoff_retrieve.py.cfg`, and two supporting Nengo components. Make sure you include the .cfg file in your folder, as it will make sure that you see the relevant output of the model.

The Zbrodoff task concerns recognizing whether problems like A + 2 = C and B + 3 = E are correct or wrong. In Assignment 3 you developed a model of this task in ACT-R, which started out by counting and used instance-based learning to gradually move to a retrieval-based strategy. If you don't remember the task anymore, read pp. 131-134 of the ACT-R reader.

The Nengo model does not count; it only retrieves facts from memory to determine whether presented problems are correct or wrong – the assumption is that the model previously learned all the necessary facts. Start Nengo and open `zbrodoff_retrieve.py`. The model already retrieves a fact from memory and compares it to what is presented on the screen. It is your job to add a motor state and write two actions ('production rules') to make it give the correct response.

First, browse through the model code and read all comments. The model is more complex than you've used so far, and also much larger. The current model uses 98,220 neurons, and that will increase when you add a motor state and the required actions. Because it is so large it will run quite slowly. To make it go a little faster, I set the random seed to a fixed number, so Nengo will use cached objects after you run it the first time. Even so, be prepared to wait for this model to build and start running – think carefully about your code before you click play.

The model uses 'Vocabularies' to pre-specify the semantic pointers that it needs. A vocabulary is a collection of semantic pointers with their associated vectors. For example, I could create a 6D vocabulary to represent the semantic pointers 'A' and 'B':

```
D = 6  #dimensionality of the vectors
testvocab = spa.Vocabulary(D)
testvocab.parse('A')
testvocab.parse('B')

print('\nVector A')
print(testvocab['A'].v)
print('\n\nVector B')
print(testvocab['B'].v)
```

This would give the following output:

```
Vector A
[-0.25124976  0.18372757 -0.60161856 -0.34652795 -0.13951407 -0.63374052]

Vector B
[ 0.59545435  0.40510802 -0.02749763 -0.21841606  0.12885182  0.64517991]
```

Each spa.Vocabulary has a fixed dimensionality (= a fixed vector length). The vocabularies with knowledge about numbers, letters, and the Zbrodoff problems use 128-dimensional semantic pointers, while the vocabularies with goal states and motor states use only 16 dimensions. If you increase the dimensionality of the model its reliability improves, but at the cost of more neurons (e.g., if you increase the dimensionality to 512 the number of neurons increases to almost 350,000). If you decrease the dimensionality, the model will be more often confused.

Problems are encoded as an addition of three circular convolutions. For example, B + THREE = E, becomes:

```
ITEM1*B + ITEM2*THREE + RESULT*E
```

where * is used to indicate convolution. You can compare this to a chunk in ACT-R:

```
chunk B3E
      isa         zbrodoff-problem
      item1       B
      item2       THREE
      result      E
```

To keep the size of the model somewhat manageable, I did not include a visual system. Instead, the presented problems are directly 'injected' into the goal and imaginal states of the model. The current problem is "B + TWO = D"; you can change the code on line 100 to present other problems. In addition, the goal state is set to 'START' for the first 50 ms, so the model knows what to do. This is comparable to setting the goal-focus in ACT-R.

The model itself follows the structure and setup of the ACT-R model of Assignment 3. In addition to the components that you're already familiar with, it uses two new Nengo components that are useful for symbolic processing. These components are a simple memory system (model.declarative, line 149) and a comparison system (model.comparison, line 152). The memory system knows about the facts specified in the vocabulary *vocab_problems*. It uses an accumulator that indicates the current status of the retrieval (called *memory_status* in the GUI, *declarative_status* in the Basal Ganglia actions). The comparison component is used to compare two semantic pointers, which can be set with *comparison_cleanA* and *comparison_cleanB*. It uses an accumulator to indicate the similarity of the two presented pointers (called *comparison_status* both in the GUI and the actions).

The GUI shows the contents of most parts of the model. On the left, it shows the current output and status of declarative memory. On top, it shows the goal state, as well as the current contents of the imaginal slots `arg1` and `arg2`, and the goal slot `target` (together, these slots contain the presented problem). On the bottom, the graph labeled 'input' shows the utility of the action in the Basal Ganglia. The action with the highest utility is chosen at any moment, which is shown in the 'output' graph, indicating the output of the Thalamus. Finally, on the bottom right the current status of the comparison is shown. This is what you need to use to make the right decision.

To complete the model:

1. Start by adding a motor state on line 155. Use the right dimensionality *and vocabulary*. Afterwards, you can uncomment the 'done' action on line 173 and run the model. (1 point)
2. Add *very specific* comments to each existing action in the basal ganglia. Explain what all parts on the left- and right-hand side of the actions do. If you don't understand certain things, change them to see what happens.
(1.5 points)
3. Add two actions under 'respond': one for Yes and one for No. These actions should set the motor state to the correct answer based on *comparison_status*. Afterwards, the model should continue to the 'y_done' action. Make sure the model does not return to one of the earlier actions after it responds. Note that *comparison_status* always decreases at first, and only increases a little later for correct problems. Start the names of the actions with 'w' and 'x' to make sure they end up in the right order in the input and output displays of the BG and Thalamus.
(1.5 points)

What to hand in for Part 1:

- The python file with your completed model, including the *very specific* comments on the existing actions.
- Two screenshots: one where you present B+TWO=D and the model says 'Yes', and one in which you present B+THREE=D and the model says 'No'.

Assignment Part 2: Reflection and comparison to ACT-R

1. If you present the model with "C + ONE = D" the model makes a mistake. Explain why that happens, and what it tells us about the semantic pointer ONE and its relation to other semantic pointers. (1 point).
2. The current model only retrieves learned facts from memory. Which elements would you need to add to also make it count and learn? Try to think of all the different things you need, including the type of learning (see slides of lecture 11). You do not have to provide code, try to stay at a conceptual level (e.g., "actions to do *xyz*") (2 points)
3. One structural difference between the Nengo and the ACT-R model is that we cannot compare two model components in Basal Ganglia actions in Nengo, something that is often done in ACT-R (e.g., compare a retrieval slot with an imaginal slot). Instead, we have to add a separate 'Comparison' component. Explain what the theoretical implication is, that is what does this claim about cognition and/or the brain? (1 point)
4. Compare the Nengo model to the ACT-R model of Assignment 3:
   a. Give a *theoretical* advantage of the Nengo model. (1 point)
   b. Give a *theoretical* advantage of the ACT-R model. (1 point)

What to hand in for Part 2:

- a pdf with your answers.