Python Productions

Models and Modules

Mon, 19/05/2008 - 20:16 — terry

The following code creates a simple ACT-R model that fires a single production.

```
from ccm.lib.actr import *           # allows use of Python ACT-R

class SimpleModel(ACTR):             # everything in here defines the model
  goal=Buffer()                            # create the goal buffer

  # now define the productions
  def first_production(goal='starting'):    # IF the goal buffer matches this
    goal.set('ending')                      #  THEN do this


# now we actually create an instance of the model
model=SimpleModel()
model.goal.set('starting')      # and initialize its goal buffer
model.run()                     # and run it
```

Building Models from Modules

The first step in model creation is to define the modules that will comprise your model. These modules are the separate interacting brain mechanisms that comprise your model. The following modules are available:

Buffers: These are simple ACT-R buffers which are able to store a single chunk. These are the main system for communicating between modules. You can create as many as you like and give them any names you like. Traditionally, ACT-R models have included a goal buffer, a retrieval buffer, and a visual buffer. More recently this has been expanded to include an imaginal buffer as well. (Note: There has been a tradition for ACT-R buffers to end in 'al').

```
  goal=Buffer()
  retrieval=Buffer()
  imaginal=Buffer()
```

Productions: Every ACT-R model automatically has a procedural memory system. This is what causes productions which match the current situation to 'fire' (i.e. to perform actions by passing messages to modules or changing buffer values). The default procedural memory system is a very simple one that does not perform any learning to adjust the utility of each production, so if more than one production can fire at a given time, one will be chosen at random.

By default, productions require 0.05 seconds to fire. You can adjust this by setting the `production_time` value. You can also set `production_time_sd` to adjust the standard deviation of the production times (default is 0). Finally, you can set the `production_threshold`, which indicates that a production must have a utility above this threshold before it will fire.

```
class SimpleModel(ACTR):
    production_time=0.05
    production_sd=0.01
    production_threshold=-20
```

There are a variety of modules that can affect how the production system works. The primary effect of these modules is to adjust the utility values within the procedural memory system. All of the parameters shown below are optional (with default values given), and can be adjusted while the model is running. They are all optional and can even be combined together.

```
class SimpleModel(ACTR):
  # production noise
  pm_noise=PMNoise(noise=0,baseNoise=0)

  # standard PG-C
  pm_pgc=PMPGC(goal=20)

  # success-weighted PG-C (see Gray, Schoelles, & Sims, 2005)
  pm_pgcs=PMPGCSuccessWeighted(goal=20)

  # mixed-weighted PG-C (see Gray, Schoelles, & Sims, 2005)
  pm_pgcm=PMPGCMixedWeighted(goal=20)

  # Temporal Difference Learning (see Fu & Anderson, 2004)
  pm_td=PMTD(alpha=0.1,discount=1,cost=0.05)

  # The new TD-inspired utility learning system
  pm_new=PMNew(alpha=0.2)
```

Most of these systems require some sort of indication of a 'success' or 'failure' for a given production. This is done within a production using one of the following commands:

```
  self.success()
  self.fail()
  self.reward(0.8) # or any other numerical reward value,
               #  positive being good and negative being bad)
```

Production Compilation

There is also a mechanism for supporting production compilation. This requires the precise specification as to what commands to compile over.

more to come

Multiple Production Systems

One unique feature of Python ACT-R is the ability to have multiple production systems. This allows us to quickly create new brain modules using the familiar syntax used to create the core production system.

more to come

Comments

You do not have permission to add comments.