ACT-R – subsymbolic parameters

Base-level learning

Switched on by subsymbolic=True.
The equation describing learning of base-level activation for a chunk i is:

$$B_i = ln(\sum_{j=1}^{n}(t_j^{-d})) + \eta$$

- n: The number of presentations for chunk i
- t_j : The time since the jth presentation
- d: The decay parameter (set by decay)
- η: the instantaneous noise

The (instantaneous) noise:

$$\sigma^2 = s^2 * \pi^2/3$$

- s: The noise parameter (set by instantaneous_noise)

Retrieval latency:

$$T = Fe^{-A}$$

- A: Activation of the chunk retrieved
- F: The latency parameter (set by latency_parameter)

Retrieval latency when retrieval fails:

$$T = Fe^{-\tau}$$

- τ: The retrieval threshold (set by retrieval_threshold)
- F: The latency parameter (set by latency_parameter)

For an example see u4_paired in **tutorials**.

Source and activation

Switched on by subsymbolic=True and specifying buffer_spreading_activation (see below).

$$A_i = B_i + \sum_k \sum_j W_{kj} * S_{ji}$$

- A_i: activation of the chunk i
- B_i: base-level activation, see above
- W_{kj}: the amount of activation from source j in buffer k
- S_{ji}: the strength of association from source j to chunk i

W_{kj} is set by buffer_spreading_activation. The value of this parameter is a dictionary in which keys specify what buffers should be used for spreading activations, values specify the amount of activation in these buffers.

$$S_{ji} = S - ln(fan_j)$$

- S: the maximum associative strength (set by strength_of_association)
- fan_j: the number of chunks in declarative memory in which j is the value of a slot plus one for chunk j being associated with itself

For an example see u5_fan in **tutorials**.

Adding partial matching

Switched on by subsymbolic=True and partial_matching=True.

$$A_i = B_i + \sum_k \sum_j W_{kj} * S_{ji} + \sum_l M_{li}$$

- M_{li}: The similarity between the value l in the retrieval specification and the value in the corresponding slot of chunk i

The similarity currently only uses default values - a maximum similarity (0) and a maximum different (-1). To be added: let the modeler set these values. For an example see u5_grouped in **tutorials**.

Utility in production rules

Switched on by partial_matching=True. The (utility) noise:

$$\sigma^2 = s^2 * \pi^2/3$$

- *s*: The noise parameter (set by utility_noise)

Each rule can specify its own utility (by having the parameter utility=n, where n is a number). Each rule can also specify reward it creates for utility learning (by having the parameter reward=n, where n is a number). Utility learning is set by utility_learning=True. The learning rate for utility learning is set by utility_alpha. For an example see u6_simple in **tutorials**.