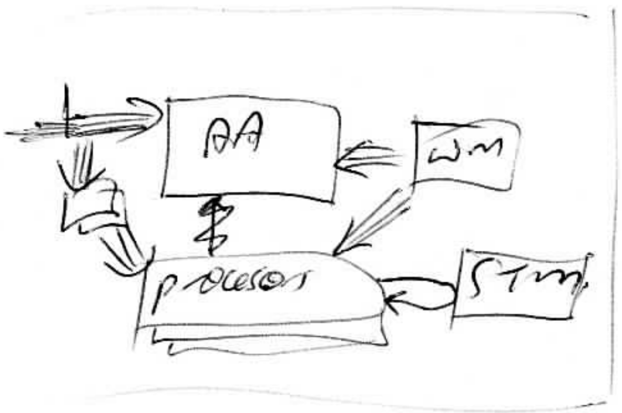
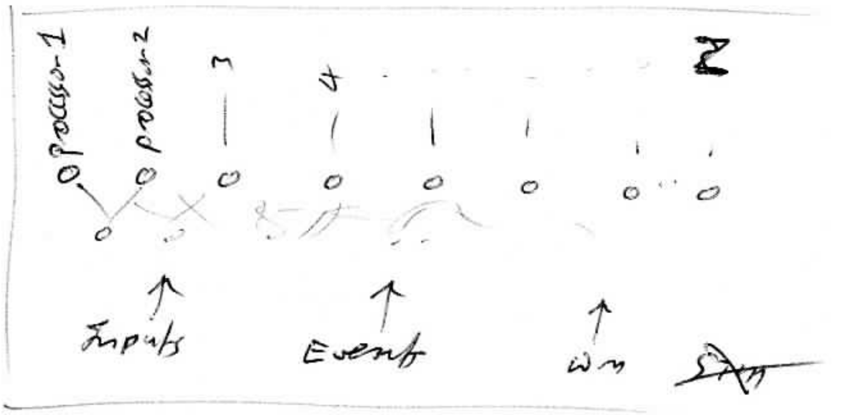
Optimization - Processor Selection.

At first all processors are executed before choosing a winner. With 1000s that will be too slow. So use a Neural network to learn the strength outputs from the processors.





Each processor outputs:
- Event
- writes
- Strength

used to choose.

- N.N. learns and outputs just the strength.
- When using, picks 10 greatest strengths & runs those processors fully, before ~~Finally~~ AA finally chooses outcome.
  - If not enough certainty in output then it falls back to running all processors

- To help with continued learning probably also run ~~the~~ 10 random processors and feed the expectations back into the N.N.

- Has interesting artifact for overall learning:

Learning will be slower, because the selection N.N. needs to learn the new behaviour of the new/updated processors. Until then it may ignore the newly skilled processor.

This will probably give a more human behaviour, even though this is totally an artifact of using a sequential CPU to simulate an intensely parallel brain.