Memory consists of a nT0 martingale process + nF features. (Ignoring C because not simulating cycles.)

Currently, I build the Hopfield net on a full timeline of all patterns with a small baseline alpha. (Creates noise.)

In order to make the flood be future-related only, I reset the Martingale process after the flood.

Reminders = current martingale + old feature.

# Properties

Baseline noise is proportional to the baselineAlpha in the Agent.

Decay time is proportional to the MemoryTimeline proportion of martingaleBits and to the proportion of nT0 to nF.

Having the Hopfield net flattens out the recall events.

More salience 🡪 longer decay times from Hopfield, but not no-H.

More cost 🡪 longer decay times for both.

Reminders do increase cost around reminded times. The more reminders you have, the more it connects to other times and there is a non-linear increase. Also, the more reminders you have at similar times, the stronger the effect.

Alleviation works as well, but it does not erase the whole memory – because the recall is never perfect.

# Notes

Should be able to store about 0.14\*N patterns in the Hopfield network. (400 neurons = 56 patterns).

Do I need to build the models to work forward in time?