“popco” transmission in CultranDejanet, SuccessBias1, etc.

Marshall Abrams

Let a be the activation of a node, and a' be the next activation. Then, if the incoming communication is .05, the update function is:

$$x' = \min(1, x + .05(1 - x)) = \min(1, .95x + .05)$$

(A positive increment can’t push the activation below the minimum, so we can ignore that possibility.)

When the incoming communiction is -.05, the update function is:

$$x' = \max(-1, x - .05(x \;-\; -1)) =$$

$$\max(-1, x - .05(x + 1)) = \max(-1, .95x - .05)$$

So the general update function is

$$.95x \pm .05$$

with the additional constraint that values outside of [-1,1] are mapped back to the nearest extremum.

Note that although the original formula was intended to reflect distance of the activation from the extremum in the direction of push, we see here that the formula reduces to one in which:

- We simply reduce the activation by 5%, whatever the activation was ...
- Then add or subtract .05 (no scaling here)
- And then reduce the result back to the extremum, if necessary.

Well, I suppose there is still a different way to think about the process. After all, $1 - x$ times .05, where $x = -1$, is equal to .10, whereas when $x = 1$, it’s 0. This is reflected in the fact that the result of 0.95 times x differs depending on x. It’s still a linear function, $ax + b$, though.

Note that when $x = 1$ and the communication is .05, the effect is $.95 + .05$, i.e. to simply leave x as is. Likewise when $x = -1$ and the communication is -.05. Oh—so the min/max operations are not needed. Ah—They’re not needed for the above formulas, but clipping to 1 and -1 is needed when `min-dist-from-extremum` is not 0, since in that case the increment will often be greater than .05, in effect. e.g. when `min-dist-from-extremum` = 1, the first formula above would become $\min(1, x + .05)$ whenever x was positive, so when e.g. $x = 1$, $x + .05 > 1$. (This also conveys how setting `min-dist-from-extremum` to a value other than 1 causes convergence to occur more quickly, and causes departure from an extreme to become more difficult. cf. NodeUpdateNotes.md.)

(Note that the $.95x$ is not the same as a decay function in the sense that's common in connectionist/neural networks, since if a node receives no transmission, this reduction isn't applied here. A real decay function acts on every tick regardless of what happens with incoming transmissions.)