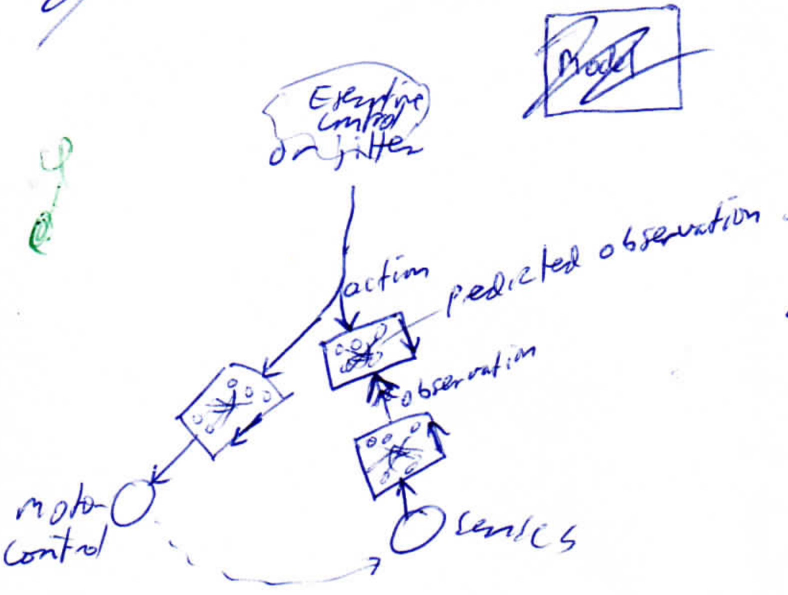
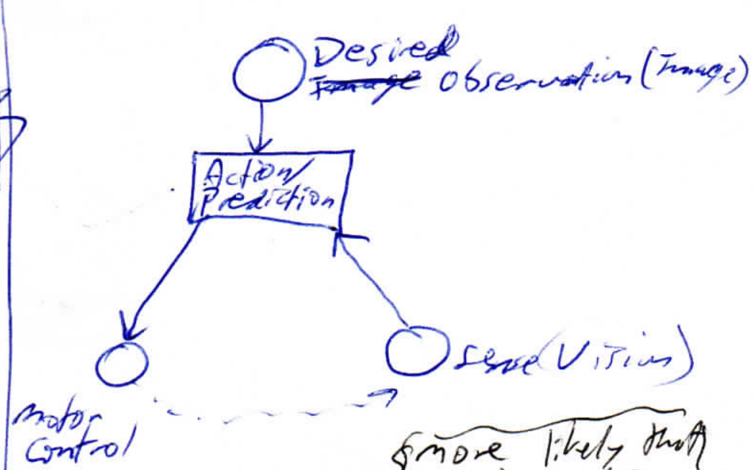
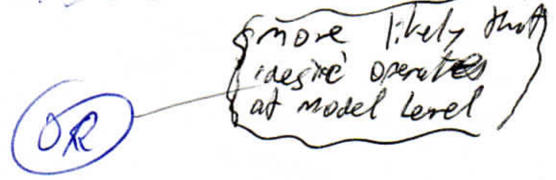
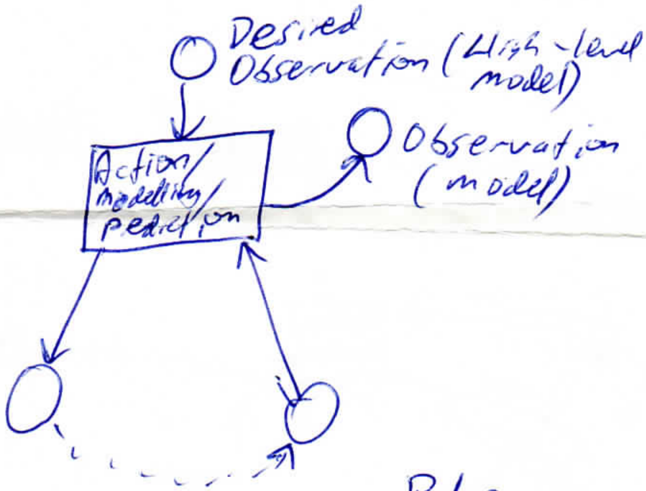


Low-level intuitive controls:
See arm ~~and~~ desire to see it move to particular location (in order to pick something up). (Ignore executive control side of Jig for now).





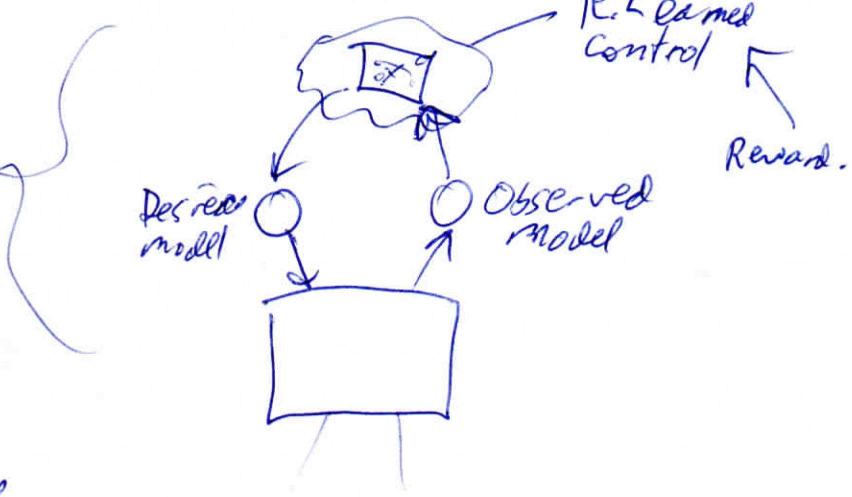


Needs jitter to bootstrap & lead to convergence (stability).

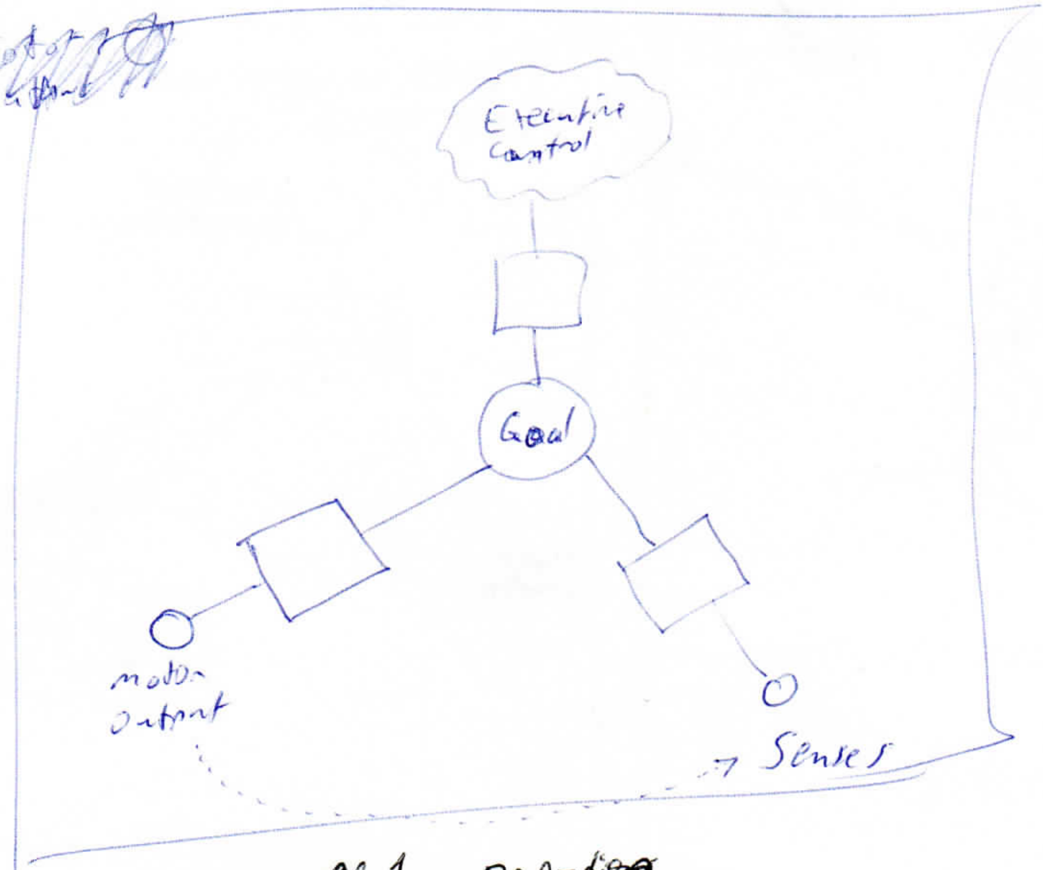
Jitter 2 ~~mot~~: Random signals to Desired model.

Jitter 1: Random motor control signals. Use touch as shared sense that builds up model.

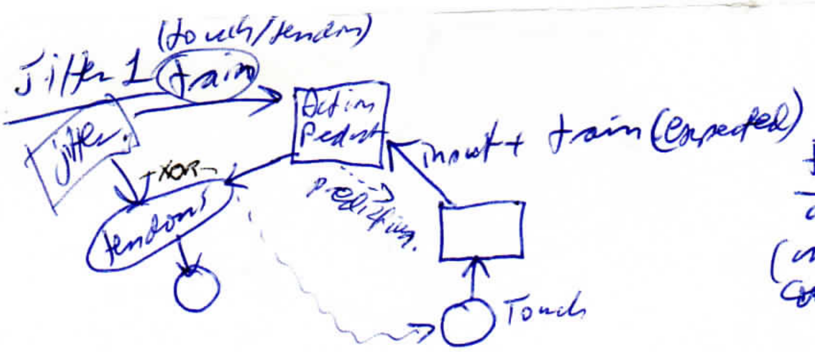
(a la Daniel Kahneman)

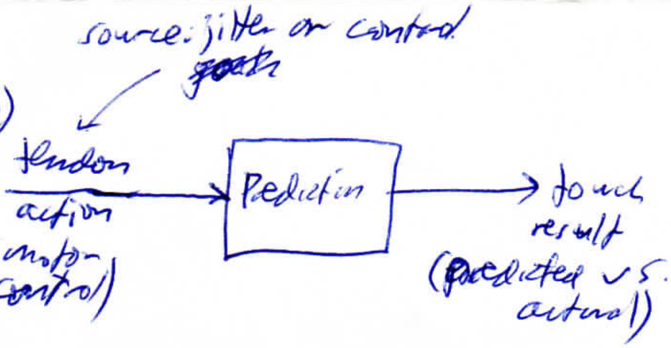


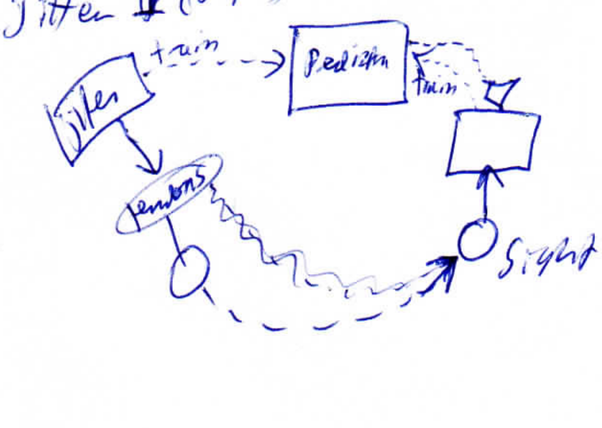
A5Z2YTR NNL



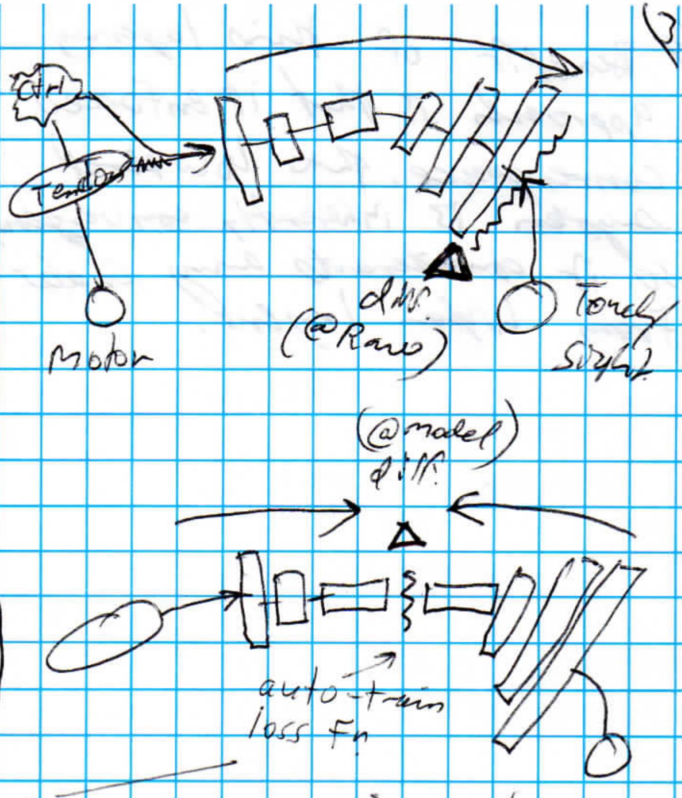
Unconscious ~~mode~~ 1 prediction







Now somehow layer this up. This prediction becomes input to jitter 2 circuit.



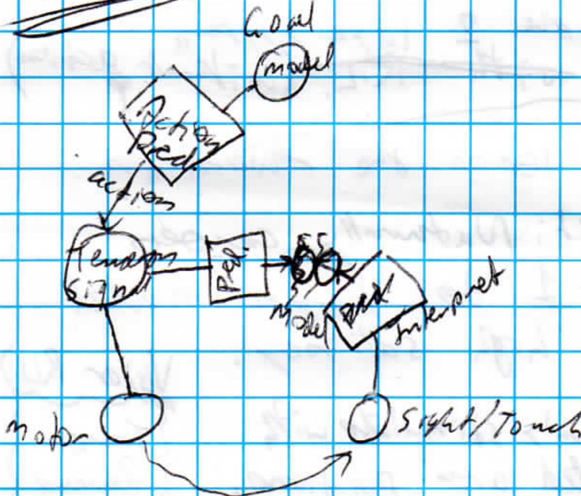
Basically an adversarial model

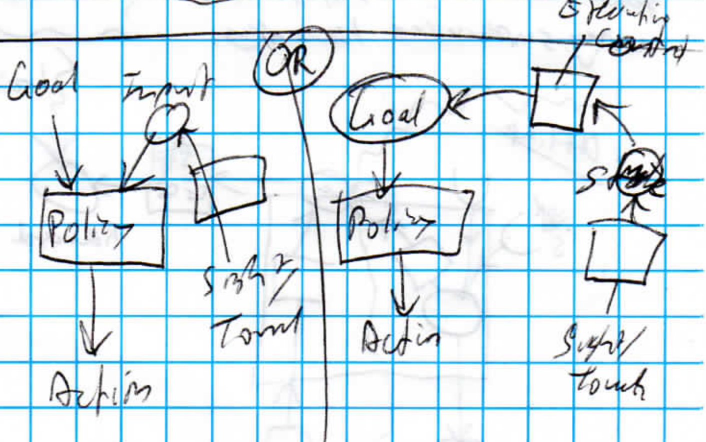
Risk of learning easiest soln: always blank. Evolutionary pressure will maximise contrast/saliency/utility: will add other pressures/mechanics.

eg: some sort of white-balance or normalisation.

4) Benefit of this layering approach is that it enforces convergence. The lowlevel system is inherently convergent, so it counteracts any chaos from higher layers.

Layer 2





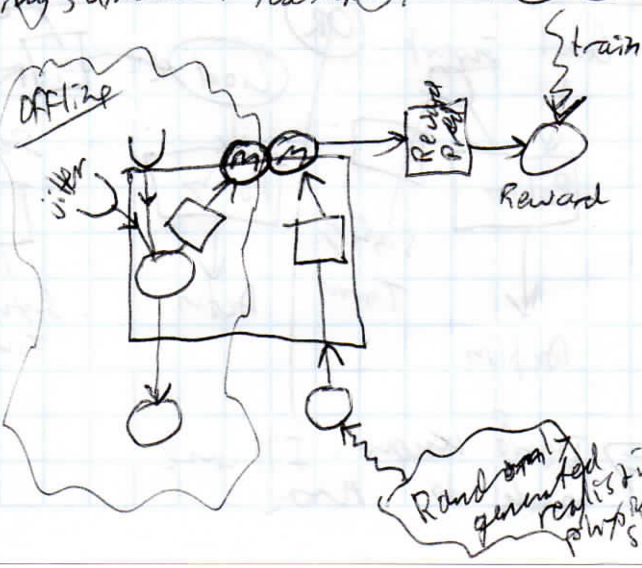
⇒ Don't know. I'll come back to this

6/ Simple 2 layer system ~~with R.L.~~ (without action)

Trick: learn the reward fn

Benefit: Naturally causes layer 1 to learn model with high saliency.

Randomly stimulate with generated arm positions. using supervised learning.



Result:

Learns a model representation with some utility, but biased towards just predicting return.

Alternate with random-model-since network training from filter.

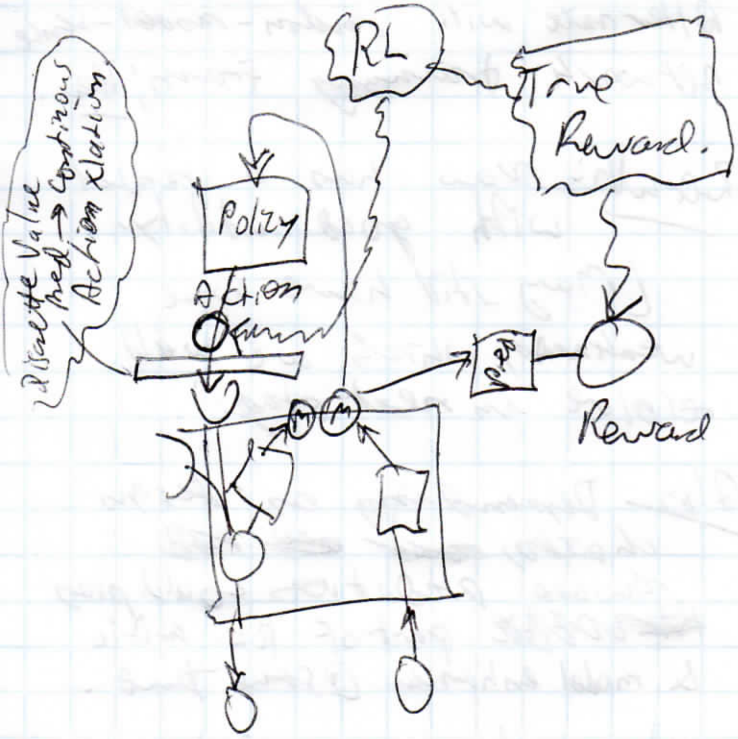
Result: now has a model with good utility.

(May still have some weakness, which we will resolve in next stage).

Also: Depending on design choices, ~~add~~ ~~as~~ ~~to~~ reward prediction could play ~~RL~~ ~~critic~~ part of RL critic & model enhancer @ same time.

Alternatively, go straight to full RL (as per next page). But 2 the idea of internalised reward.

Simulate 2 layer with action Policy & RL.

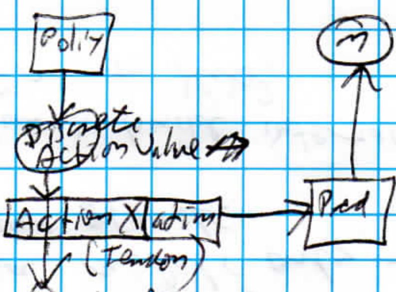


Result:

model with full utility.

Q: how does continuous action policy learning work.

However, maybe we can work with any solution, in order to train tendon → model prediction.



Next: layer 3 produces the goal ← and this is what we are conscious of.

For Experiment below

Idea:

High Level (HL) communicates via models only - no lowlevel raw signals.

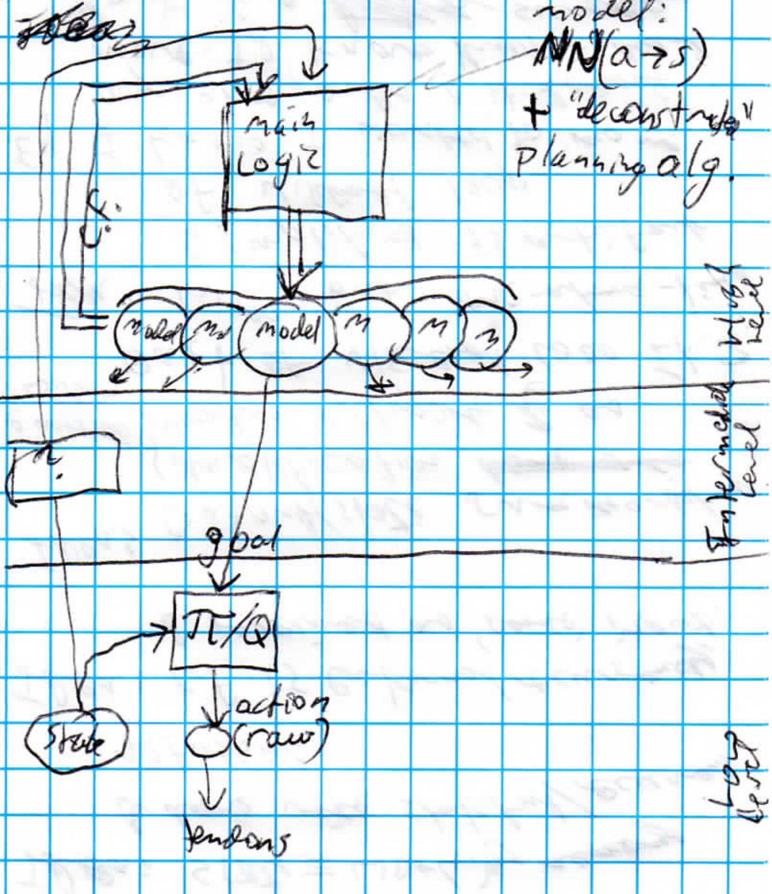
Q: How to build internalised reward fn into this?

Future addition:

Add "suggestion" prediction ~~net~~ $NN(s, g \rightarrow a)$ as input that learns best approach from past experience & optimises action search space.

Experiment - From High Level

~~Ideas~~



For below

Idea: state ≡ working memory & done via stateful/recurrent network.

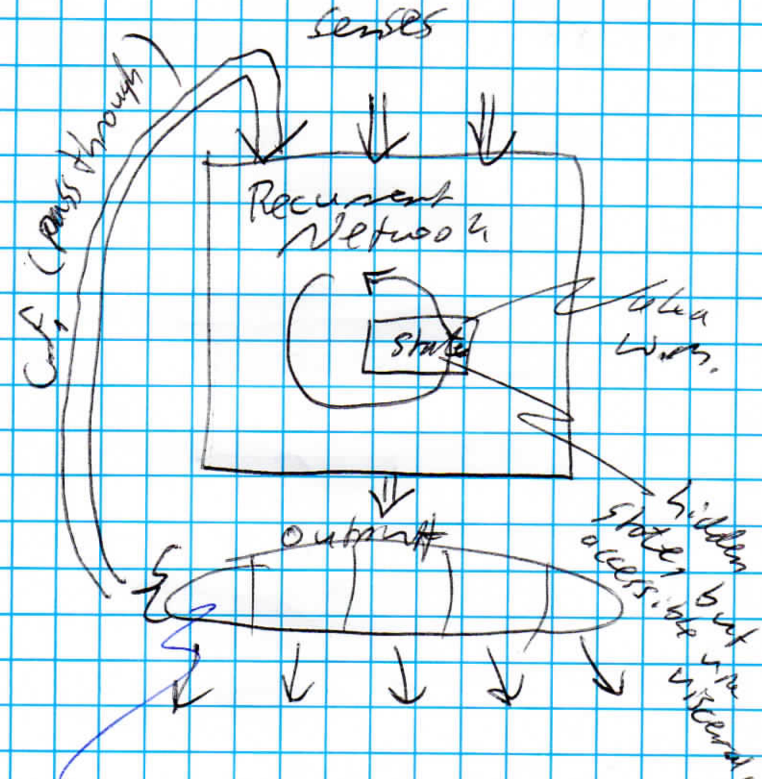
Idea: C.f. is external recurrency & received as 'sense' input.

Idea: Output/state summarizing simplification ~~done via~~ embedded within meta network & as part of visceral loop It.2.

Idea: W.M. but one-at-a-time & simplified is a-to fact of visceral loop.

Eg: I know I wanted to move my arm, even if it doesn't move. I'd know that I didn't do it if a ~~doctor~~ surgeon moved my tendon.

Review High Level State + C.F.



physically structured output array
sections allocated & directed
to specific targets.

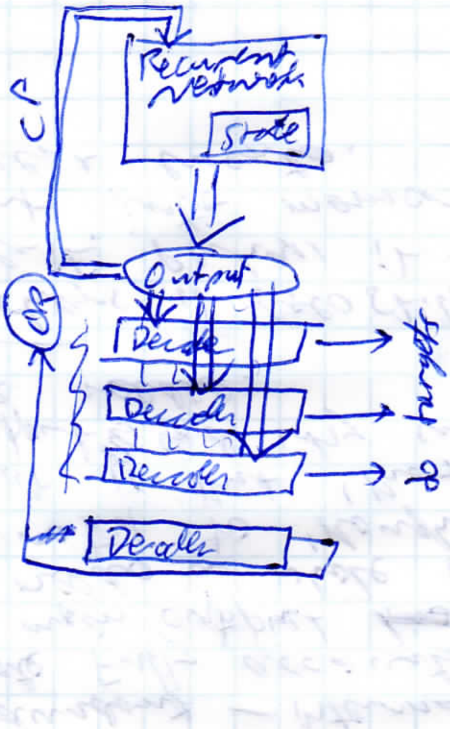
14

~~[illegible]~~

· Above doesn't quite fit with receiving own thoughts as CF input ∵ it's redundant — internal hidden state fully accounts for it & main output ~~[illegible]~~ needs a whole section devoted to 'thought'. What is that 'thought' output and why would it be needed.

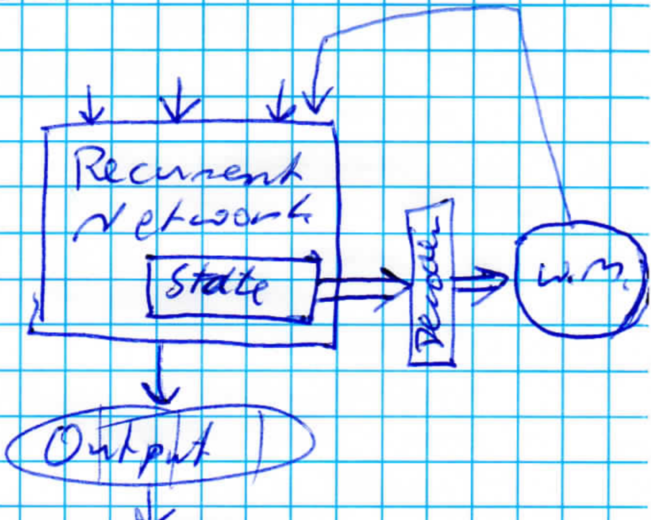
Perhaps it does still have merit. Perhaps it becomes input into memory of recent events.

Variant



Recurrent network
State
Output
Decoder
Decoder
Decoder
Decoder
targets

Variant

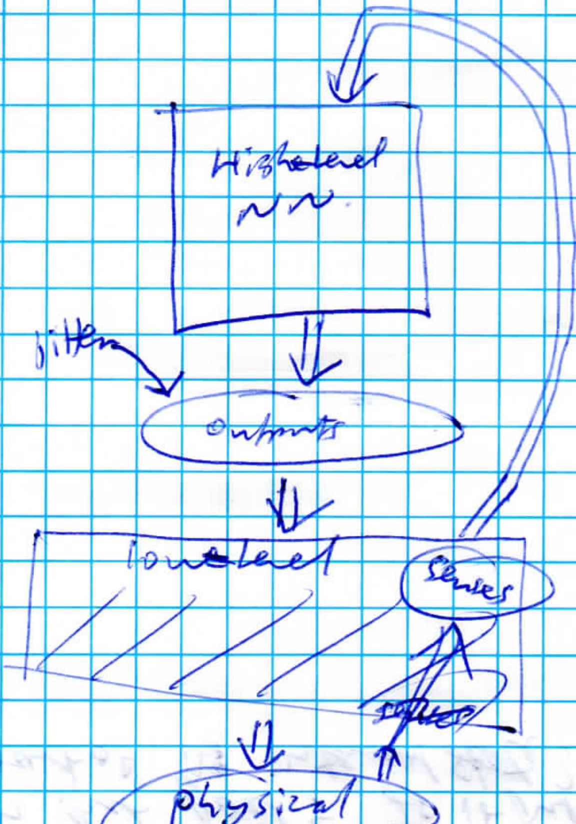


Variant

- By original external recurrency for both state & CF

Executive control

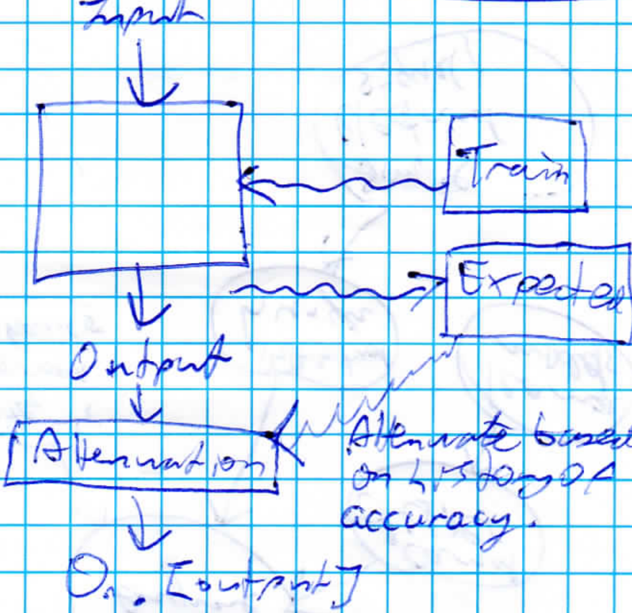
Needs to "discover" that it can control low-level. For now, re-use 'filler' approach.


Highlevel
NN
jitter
outputs
lowlevel
senses
physical

Goal

Now Executive control needs a goal. How does it decide on that goal? Is it just part of its internal state?

Aside: ~~Attenuated~~ Accuracy Attenuated networks



Self Driven Goal - Basics.

- ~~Robots~~ In humans,
- External reward is vague & sparse. This allows for a great variability in internal goal/reward systems.
- The internal reward systems give much finer grained rewards and thus can drive our 'style' of behaviour. As long as they are consistent with the long term reward.
- much of our learning is governed by hard-coded internal rewards (eg: hunger, ~~[illegible]~~ surprise), so there is plenty of room to ignore external rewards. This is quite different to current RL techniques, which are 100% external.

This makes complete sense, because:

There is no external reward, until the agent can grok the existence of an externality.

So all rewards start as internal rewards. This explains why we have so much leeway to choose to ignore external rewards — we are hard-wired for internal rewards, but using external rewards is a ~~optional add-on~~. learned and optional add-on.