Adaptive Systems - Evolutionary Robotics P1

3 forms of adaptation in this topic

Recap:
- not all changes are adapations
- Sucess orgs are adapted to envs
  - evolution (over generations)
  - learning (During lifetime)
- genetic mutations are random - useful / hiderance is what wins reproduction
- Best mutated gens ~~win~~ survive + multi
- Selected For , Selected against

Main ingredients for evolution

1. Reproduction
   - genes copied through generations

2. Mutation
   - to trigger variation
   - Most mutations are worse

3. Selection
   - how to determin good mutation (survival)
   - otherwise = just gene drift

Mutation & Selection

- Based on fitness stochastically
- Better = more likely but not definete
- worse = still able to reproduce
- this increase diversity
- otherwise result = homogenous

Artifical neural Networks

Gen 1 =
- McCulloagh & Pitts ANN
- threshold neurons

Gen 2 =
- Activation neurons
- inc RNNS
- Most Robotics Resarc

Gen 3 =
↳
- Spiking Neural networks (SNNs)
- More bio plausable

Continuous-time recurrent neural net (CTRNN)

→ from chaos lectures

- Dynamical system approximators
- like feedforward but for Dynamic systems

EvoRobotics Case Studies Part 2

Urzelai et al - Evolving learning rules

- Khepra Robot - 6cm diameter
- Robots task = 2 parts
  - Drive to zone, switch light
  - Park by light
- camera, sensor, wheels
- network topoly = • RNN, all nodes connected
  - no output neurons
  - All neuron = single input
  - except 2 motor neurons
  - Discrete time RNN, no time constant
  - input to neuron is an input from every other neuron
- neurons that fire together, wire together
- Synapse update rules:
  - ~~B~~hebb rule: $(1 - w_{ij}) x_j y_i$

    Min deltamin

---

fine et al. Adaptiy to your body

- Simulation, no real model
- ~~Robot~~ ~~referred~~ ~~to~~ as model
- L, R Motor, ~~single~~ light sensor
- 4 neurons, CTRNN
- N1 = sensor inpt — input
  3,4 = L, R motors — Output
  2 = hidden neuron, no outside interaction
- couple to env via 1,3,4
- experiment = move sensor from front to back & adjust

- 1, 3, 4 neurons change Quickly
- Sensor & evolve to drive towards light
- once trained/evolved change the problem
  (change location of the sensor)
- CTRNN weights are FIXED
- looks like an ultrastable adaptive system
- unlike homeostat - it does change its params

  how? CTRNN are dynamical sys approx
  in principle, A CTRNN w/ fixed weights
  can approx ANN w/ weights that do change

  the states of the CTRNN become the weights of
  the ANN weights

Case study = Homeoplastic Plasticity - Williams

- No robot
- CTRNN, fully connected
- Sigmoid of 0, 1 saturates neuron = bad, no info change
- Synaptic homeostasis used to keep neuron
  in optimal zone
- use gain to change slope of the activation funct
- bias used to shift curve