A REINFORCEMENT LEARNING PERSPECTIVE ON AGI

Itamar Arel, Machine Intelligence Lab (http://mil.engr.utk.edu)
The University of Tennessee

Tutorial outline

- What makes an AGI system?
- A quick-and-dirty intro to RL
- Making the connection RL $\leftrightarrow$ AGI
- Challenges ahead
- Closing thoughts

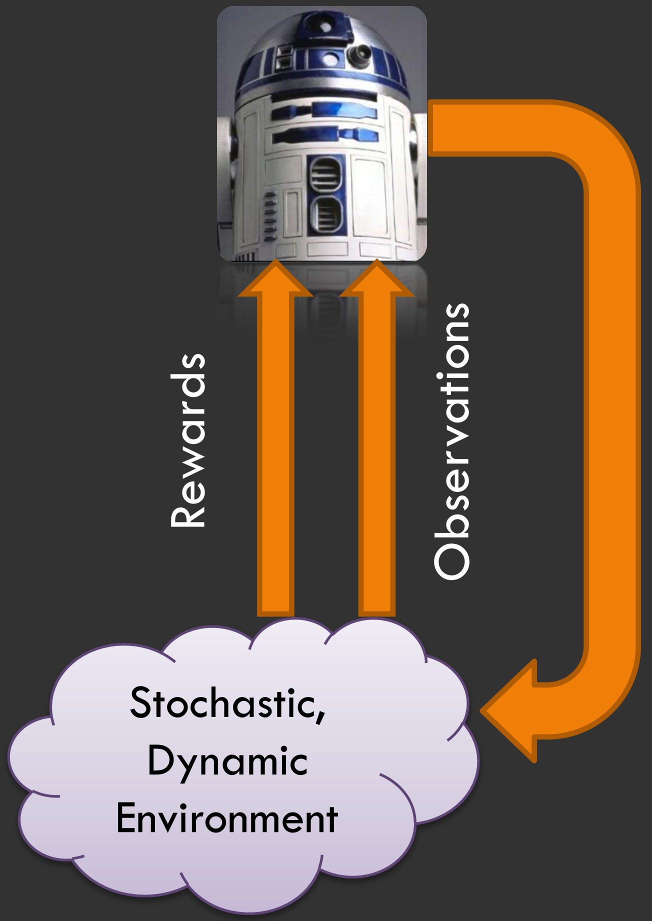
What makes and AGI system?

- Difficult to define "AGI" or "Cognitive Architectures"
- Potential "must haves" …
  - Application domain independence
  - Fusion of multimodal, high-dimensional inputs
  - Spatiotemporal pattern recognition/inference
  - "Strategic thinking" – long/short term impact

Claim - If we can achieve the above, we're off to a great start …

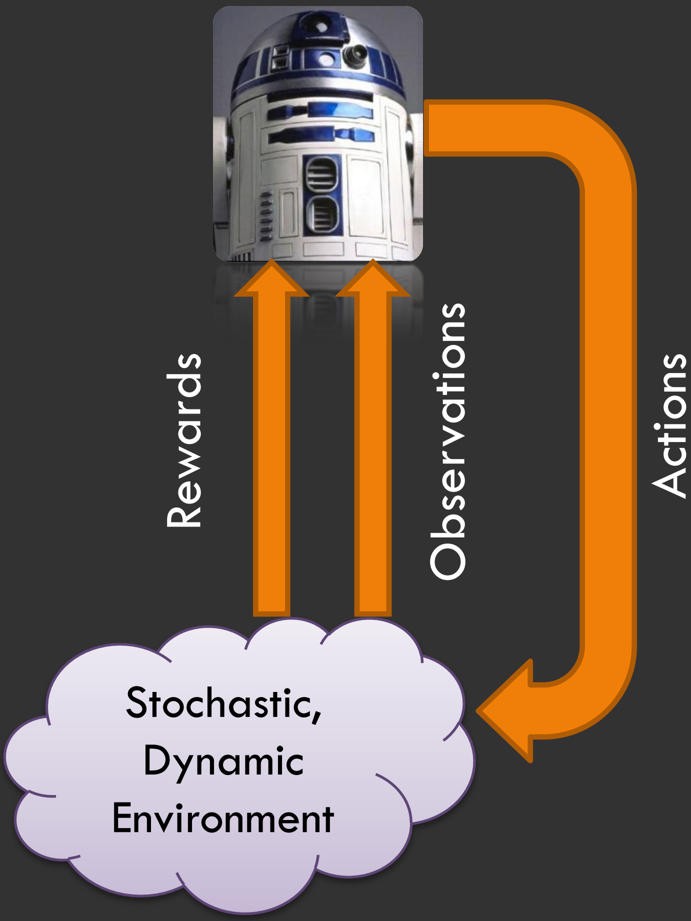
RL is learning from interaction

- Experience driven learning
- Decision-making under uncertainty
- Goal: Maximize a utility("value") function
  - Maximize long-term rewards prospect
- Unique to RL: *solves the credit assignment problem*



RL is learning from interaction (cont')

- A form of unsupervised learning
- Two primary components
  - Trial-and-error
  - Delayed rewards
- Origins of RL: Dynamic Programming



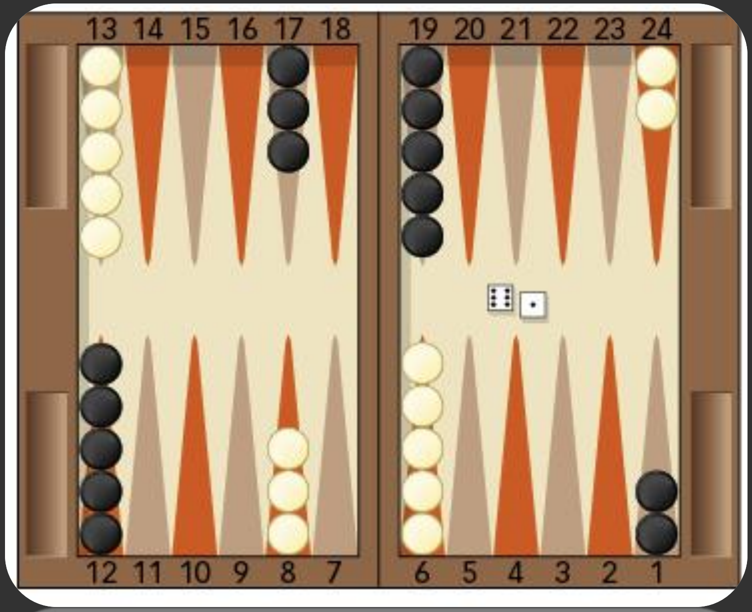
Brief overview of RL

- Environment is modeled as a Markov Decision Process (MDP)
  - S – state space
  - $A(s)$ – set of actions possible in state $s \in S$
  - $P_{ss'}^{a}$ – probability of <u>transitioning</u> from state s to s' given that action a is taken
  - $R_{ss'}^{a}$ – expected <u>reward</u> when transitioning from state s to s' given that action a is taken

Goal is to find a good policy: States $\rightarrow$ Actions

Backgammon example

- Fully-observable problem (state is known)
- Huge state set (board configurations) ~ 10^{20}
- Finite action set – permissible moves
- Rewards: Win +1
  Lose -1
  else 0



RL intro: MDP basics

- An MDP is defined by the state transition probabilities

$$P_{ss'}^{a} = \Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\}$$

and the expected reward

$$R_{ss'}^{a} = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\}$$

- Agent's goal is to maximize the rewards prospect

$$\boxed{R(t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+\tau+1}}$$

RL intro: MDP basics (cont')

- The state-value function for policy π is

$$V^{\pi}(s) = E_{\pi}\left[R_t \mid s_t = s\right] = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \mid s_t = s\right]$$

- Alternatively, we may deal with the state-action value function

$$Q^{\pi}(s,a) = E_{\pi}\left[R_t \mid s_t = s, a_t = a\right] = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \mid s_t = s, a_t = a\right]$$

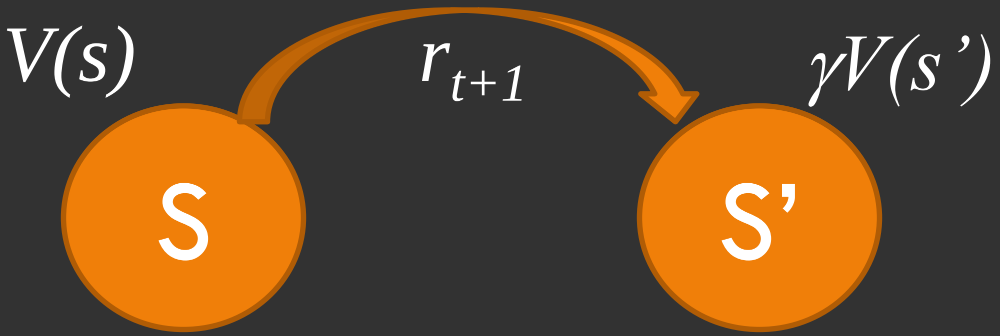
- The latter is often easier to work with

RL intro: MDP basics (cont')

- Bellman equations

$$V^{\pi}(s) = \sum_{s'} P^{a}_{ss'} \left[R^{a}_{ss'} + \gamma V^{\pi}(s') \right]$$

$$Q^{\pi}(s,a) = \sum_{s'} P^{a}_{ss'} \left[R^{a}_{ss'} + \gamma Q^{\pi}(s',a') \right]$$



Temporal difference learning

$$V(s_t) = r_{t+1} + \gamma V(s_{t+1})$$

RL intro: policy evaluation

- We're looking for an optimal policy π^* that would maximize $V^{\pi}(s) \quad \forall s \in S$
- Policy evaluation – for some π

Dynamics unknown

$$V_{k+1}(s) = \sum_{s'} P_{ss'}^{\pi(s)} \left[R_{ss'}^{\pi(s)} + \gamma V_k(s') \right]$$

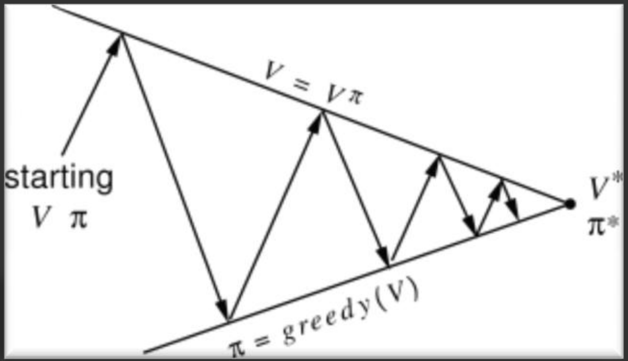
- RL problem – solve MDP when environment model is unknown
- Key idea – use samples obtained by interaction with the environment to determine value and policy

RL intro: policy improvement

- For a given policy π with value function $V^{\pi}(s)$

$$\pi'(s) = \arg\max_{a} \sum_{s'} P^{a}_{ss'}\left[R^{a}_{ss'} + \gamma V^{\pi}(s')\right]$$

- The new policy is always better
- Converging iterative process (under reasonable conditions)



Exploration vs. exploitation

- A fundamental trade-off in RL
  - Exploitation of actions that worked in the past
  - Exploration of new, alternative action paths so as to learn how to make better action selections in the future
- The dilemma is that neither pure exploration nor pure exploitation is good
- Stochastic tasks – must explore
- Real-world is stochastic – forces explorations

Back to the real (AGI) world …

- No "state" signal provided
  - Instead, we have (partial) observations
  - Agent needs to infer state
- No model - dynamics need to be learned
- No tabular form solutions (don't scale) …
  - Huge/continuous state spaces
  - Huge/continuous action spaces
  - Multi-dimensional reward signals

Toward AGI: what is a “state” ?

State is a consistent (internal) representation of perceived regularities in the environment

- Each time agent sees a “car” the same *state* signal is invoked
- States are individual to the agent
- State inferences can occur only when environment has regularities and predictability

Toward AGI: learning a Model

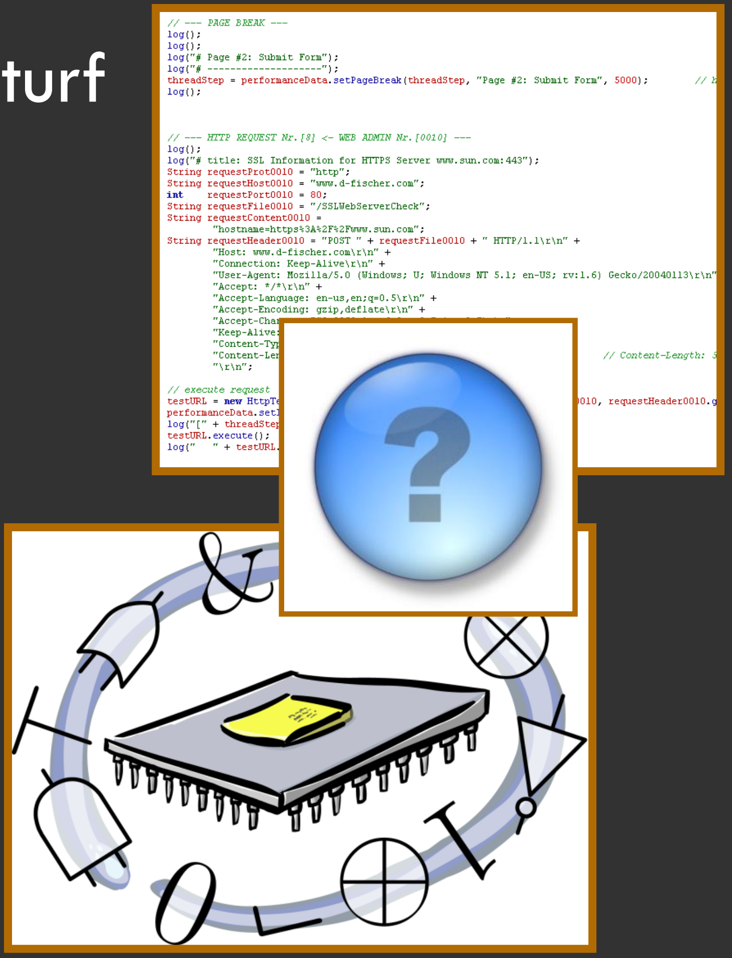
- Environment dynamics unknown
- What is a model – any system that helps us characterize the environment dynamics
- Model-based RL – model is not available, but is explicitly learned

Toward AGI: replace tabular form

- Function approximation (FA) - a <u>must</u>
  - Key to generalization
- Good news: many FA technologies out there
  - Radial basis functions
  - Neural networks
  - Bayesian networks
  - Fuzzy logic
  - …

Hardware vs. software

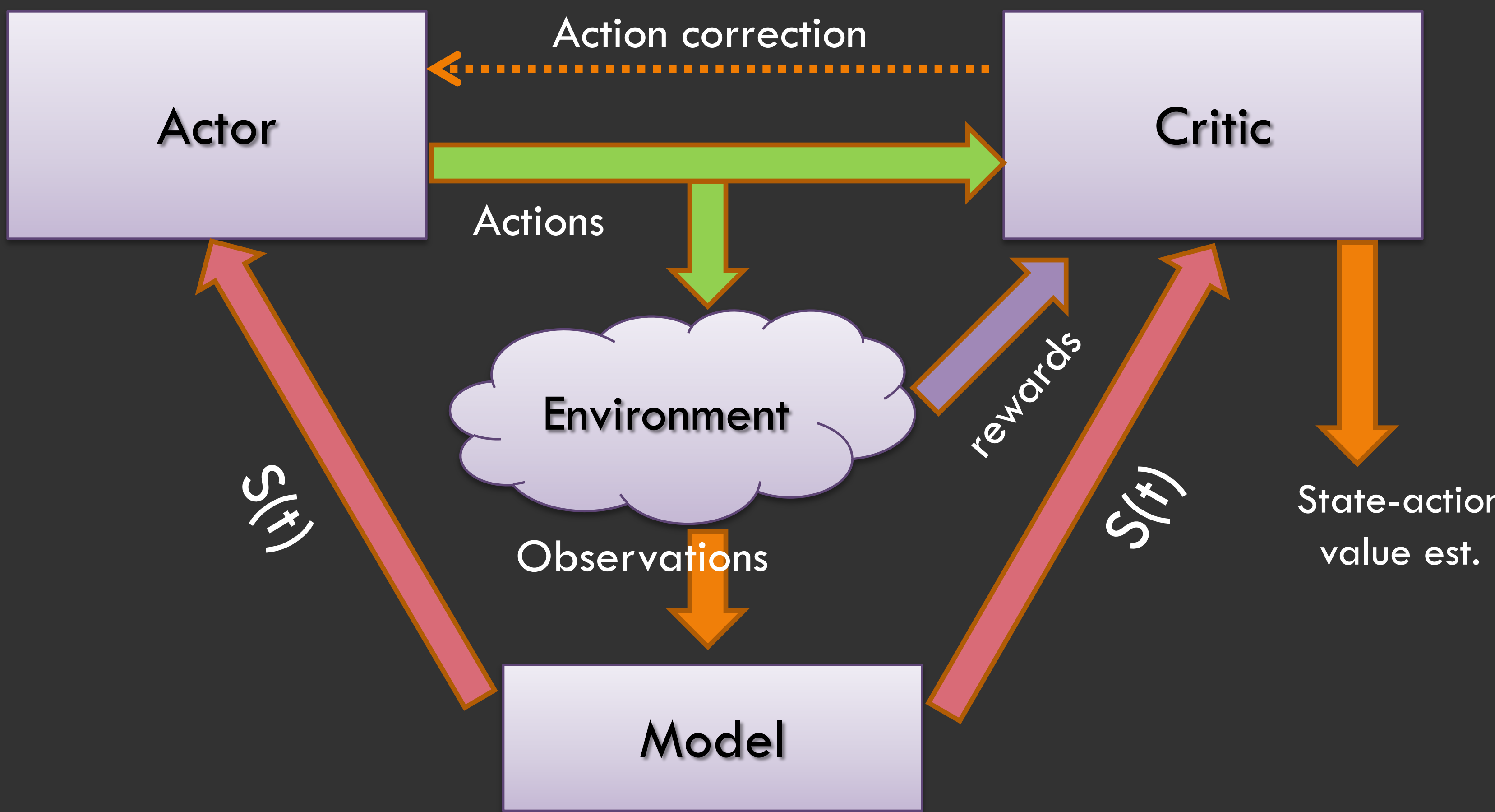
- Historically, ML has been in CS turf
  - Von Neumann architecture?
- Brain operates @ ~150 Hz
- Hosts 100 billion processors
- Software limits scalability
  - 256 cores is still not "massive parallelism"
- Need vast memory bandwidth
  - Analog circuitry



Toward AGI: general insight

- Don't care for "optimal policy"
- Stay away from reverse engineering
- Learning takes time!
- Value function definition needs work
  - Internal ("intrinsic") vs. external rewards
  - Exploration vs. exploitation
- Hardware realization
- Scalable function approximation engines

Tripartite unified AGI architecture



Closing thoughts

- The general framework is promising for AGI
  - Offers elegance
  - Biologically-inspired approach
- Scaling model-based RL
- VLSI technology exists today!
  - >2B transistors on a chip

AGI IS COMING ….

Thank you