The Predictron: End-to-End Learning and Planning

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, Thomas Degris

DeepMind

ICLR, 2017/ Presenter: Anant

Outline

Outline

1. Introduction
   - Model-Based RL
2. Predictron
   - Architecture
   - Learning Updates
3. Experiments
   - Architecture Variants
   - Comparison
   - Analysis of Depth
4. Summary

Model-Based RL

- Two types of reinforcement learning
  - Model-free: Q-learning, Actor-Critic
  - Model-based: learn model of environment
- RL 4-tuple: $(S, A, R, T) \rightarrow (state, action, reward, transition)$
- $T(s'|s, a)$ not known - can learn via model

Motivation

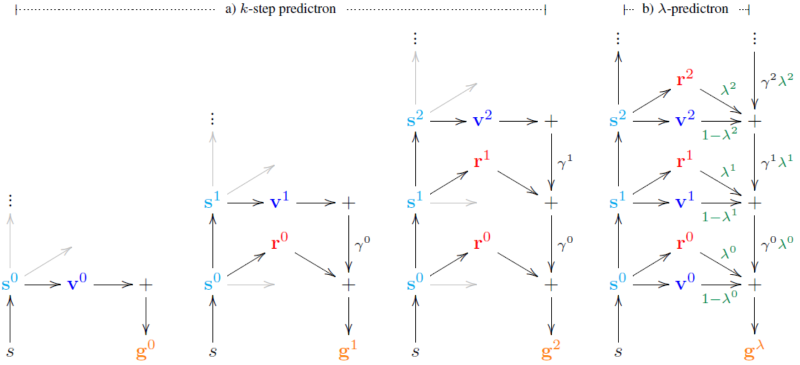
- Model-based methods inferior to model-free methods (Q-learning, actor-critic) for RL based on raw input
- Model-based RL: learn model of environment, use for planning
- Environment model trained independently of planning step - model may be suboptimal for task

Outline

1. Introduction
   - Model-Based RL
2. Predictron
   - Architecture
   - Learning Updates
3. Experiments
   - Architecture Variants
   - Comparison
   - Analysis of Depth
4. Summary

Predictron Architecture

- Four components:
  - State representation $s = f(raw)$
  - Model $s', r, \gamma = m(s, \beta)$
  - Value function (future internal return) $v = v(s)$
  - Accumulator - combines internal r, γ, v into overall value g



Accumulators

- k-step predictron
  - Roll model forward over k steps
  - $g^k = r^1 + \gamma^1(r^2 + \gamma^2(...(r^{k-1} + \gamma^{k-1}(r^k + \gamma^k v^k))...))$
- λ-predictron
  - Combine k-step preturns
  - λ^k - weight matrix
  - $g^\lambda = \sum_{k=0}^{K} w^k g^k$, w product of λ's

Outline

Learning Updates

- k-step predictron

$$L^k = \frac{1}{2}\|\mathbb{E}_p[g|s] - \mathbb{E}_m[g^k|s]\|^2$$

∇ Sample loss: $\frac{\partial l^k}{\partial \theta} = (g - g^k)\frac{\partial g^k}{\partial \theta}$

- λ-predictron - average preturn losses

$$L^{0:K} = \frac{1}{2K}\sum_{k=0}^{K}\|\mathbb{E}_p[g|s] - \mathbb{E}_m[g^k|s]\|^2$$

∇ Sample loss: $\frac{\partial l^{0:K}}{\partial \theta} = \frac{1}{K}\sum_{k=0}^{K}(g - g^k)\frac{\partial g^k}{\partial \theta}$

Experiment Contexts

- Mazes
  - 20 x 20 random maze
  - Consider locations along top-left bottom-right diagonal
  - Objective: are diagonal points connected to bottom-right?
- Pool
  - 4 balls, 4 pockets
  - Implemented in graphical physics engine - image frames
  - Objective: predict events (collisions, entering quadrants, entering pocket

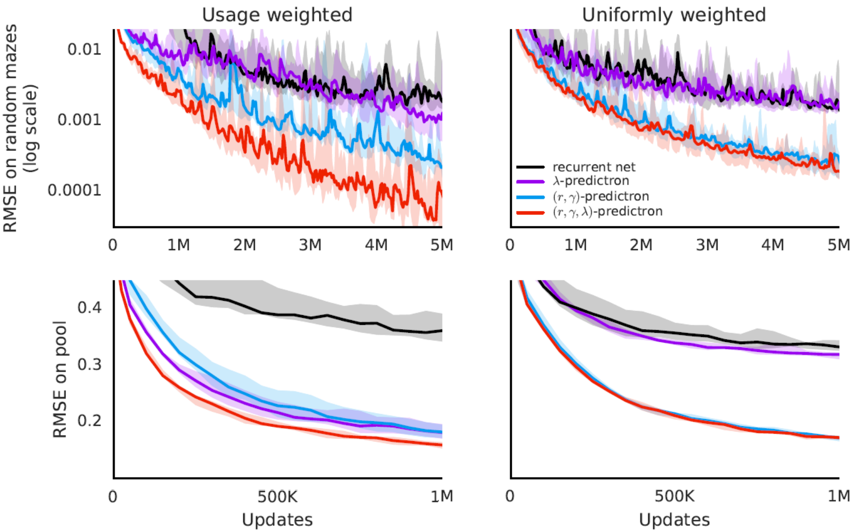
Outline

Architecture Variants

Three dimensions of predictron

1. MRP model structure
   - MRP: internal rewards/discounts learned
   - non-MRP: internal rewards/discounts ignored (set to 0 and 1)
2. K-step or λ accumulator
3. Usage weighting
   - Weight k preturn losses using w^k weights (from λ-weighting)

Architecture Variants



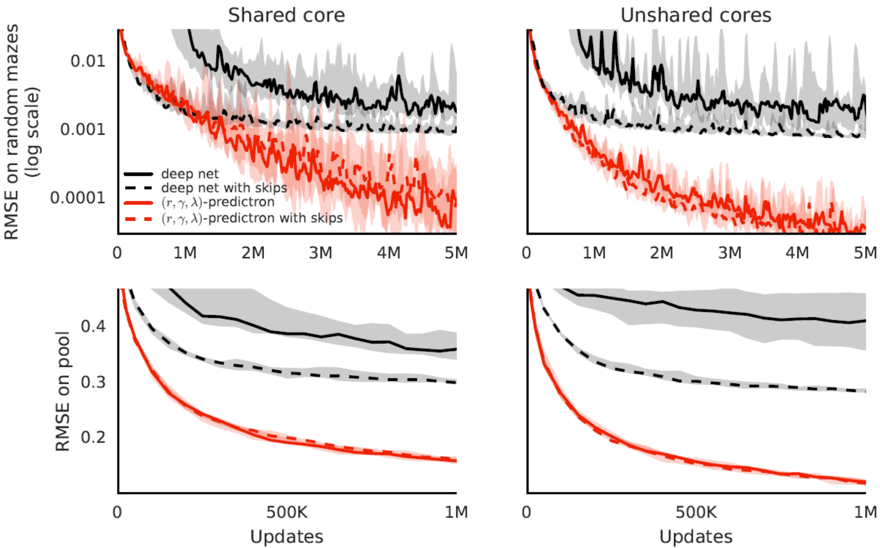
Outline

Comparison to Other Deep Nets

Three dimensions

1. Model type
   - (r, γ, λ)-Predictron
   - Other deep net (feedforward/recurrent)
2. Weight sharing
   - Cores share weights (recurrent)
   - Cores have separate weights (feedforward)
3. Skip connections
   - Output $\Delta s : s^{k+1} = H(s^k + \Delta s^k)$
   - Deep network + skip connections = ResNet

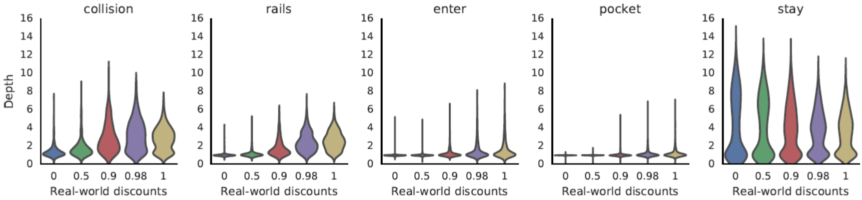
Comparison to Other Deep Nets



Outline

Analysis of Depth

- Predictron can adapt "depth" based on task
- Depth - number of model steps
- Properties:
  - Different prediction - different depth
  - Depth $\sim$ discount
  - Distributions not strongly peaked - depth can differ



Summary

- Traditional model-based RL trains model independent of planning
- Predictron: end-to-end differentiable architecture for learning/planning
- Outperforms other model-based approaches on random mazes, pool