Leveraging World Models in RL for Circuit Design Optimisation

R244: Large Scale Data Processing and Optimization
Pedro Sousa

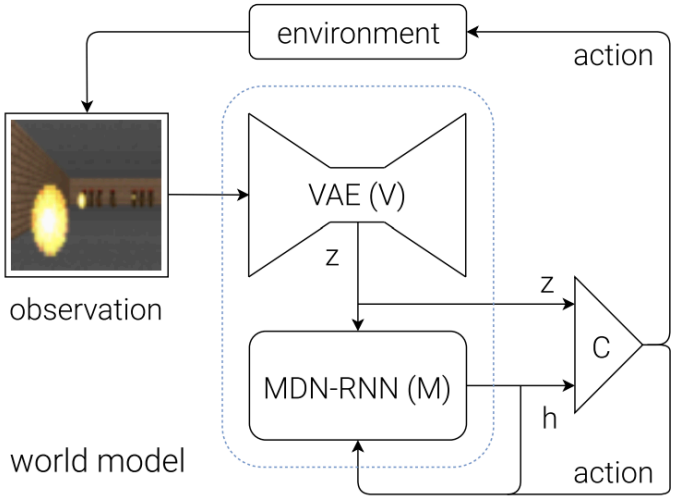
Background

- Circuit design: arrange electronic components to form a functional and optimized circuit.
- It is critical to create efficient and powerful computer systems.
- However, it is an intricate and time-consuming process, often requiring extensive human expertise and iteration.
- **Solution:** Reinforcement Learning (RL) lets systems learn optimal behaviors by trial and error, automating and enhancing circuit design.
- **New Approach:** Experiment with World Models in the RL pipeline, enhancing an agent's ability to make informed decisions.

World Models (1/2)

- Inspired by how humans develop mental models of the world, understand and predict the environment using limited sensory input.
- Agent will benefit from abstract space and time representations.
- Predictive model of future = better experience-based decision-making
- World Models have 2 main components:
  - Vision model (V) = Variational Autoencoder to encode observations
  - Memory model (M) = Recurrent Neural Network for predictions

World Models (2/2)




```
def rollout(controller):
  ''' env, rnn, vae are '''
  ''' global variables  '''
  obs = env.reset()
  h = rnn.initial_state()
  done = False
  cumulative_reward = 0
  while not done:
    z = vae.encode(obs)
    a = controller.action([z, h])
    obs, reward, done = env.step(a)
    cumulative_reward += reward
    h = rnn.forward([a, z, h])
  return cumulative_reward
```

Figure 2: Flow diagram showing how V, M, and C interacts with the environment (left). Pseudocode for how our agent model is used in the OpenAI Gym [5] environment (right).

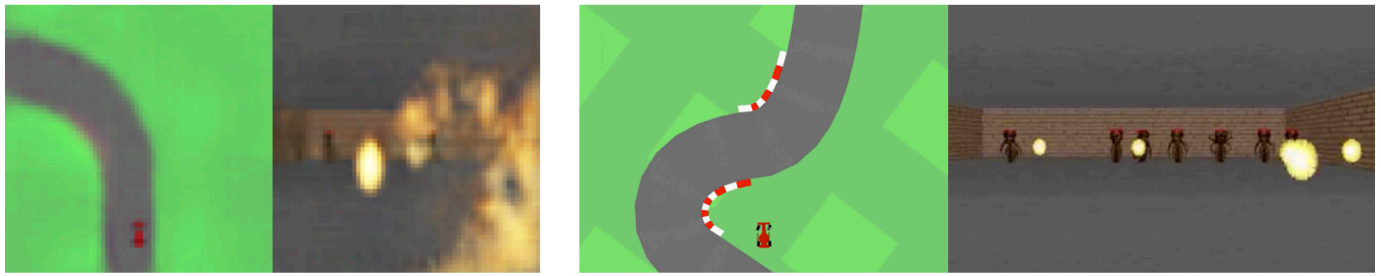
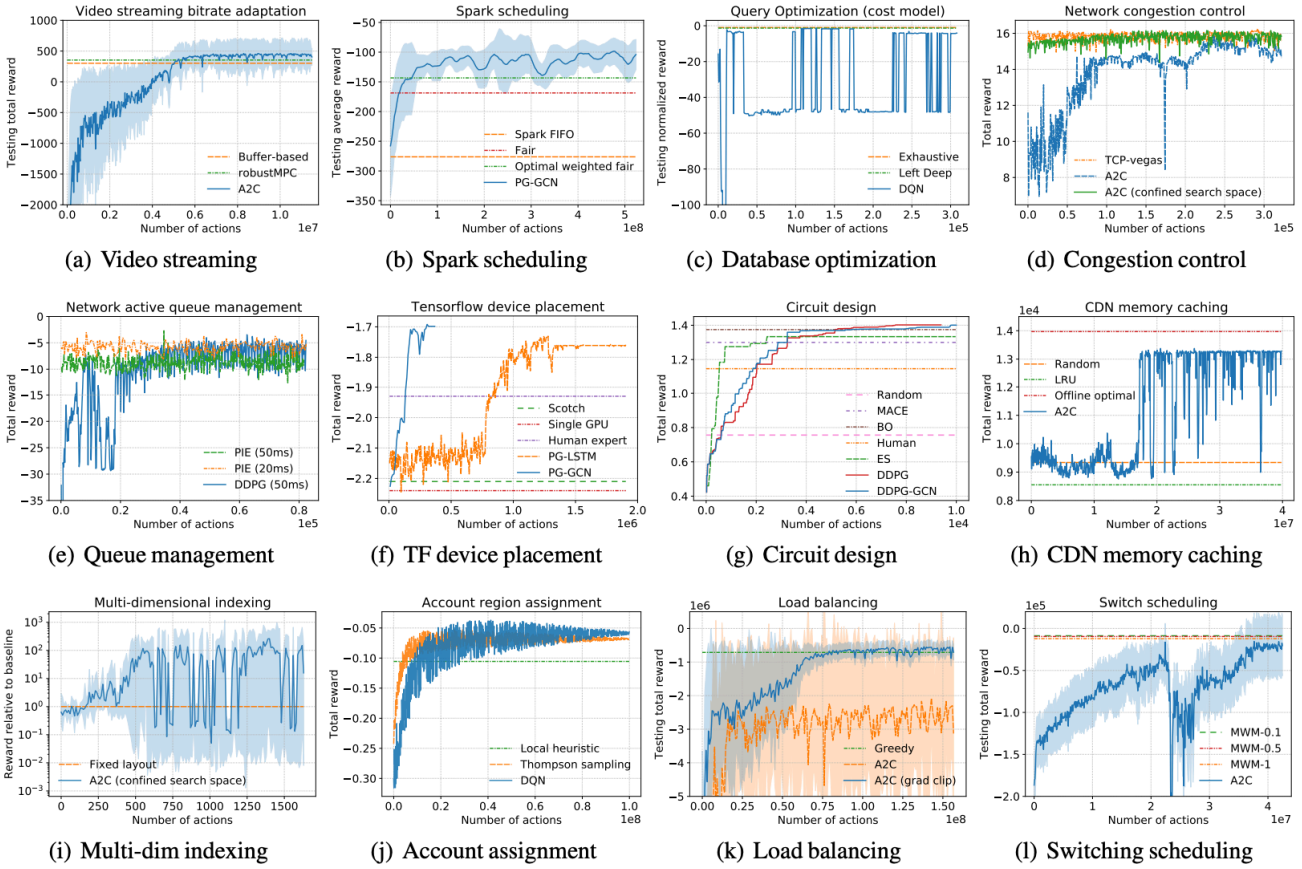


Figure 1: We build probabilistic generative models of OpenAI Gym [5] environments. These models can mimic the actual environments (left). We test trained policies in the actual environments (right).

Park (1/2)

- Open-source platform to experiment with RL for computer systems.
- Abstracts away the complexity of running experiments in real systems.
- Support 12 different computer system environments:
  - Switch Scheduling
  - Server Load Balancing
  - Adaptive video streaming
  - Circuit design
  - …

Park (2/2)


Video streaming bitrate adaptation
Testing total reward
Number of actions
1e7
Buffer-based
robustMPC
A2C
Spark scheduling
Testing average reward
Number of actions
1e8
Spark FIFO
Fair
Optimal weighted fair
PG-GCN
Query Optimization (cost model)
Testing normalized reward
Number of actions
1e5
Exhaustive
Left Deep
DQN
Network congestion control
Total reward
Number of actions
1e5
TCP-vegas
A2C
A2C (confined search space)
(a) Video streaming
(b) Spark scheduling
(c) Database optimization
(d) Congestion control
Network active queue management
Total reward
Number of actions
1e5
PIE (50ms)
PIE (20ms)
DDPG (50ms)
Tensorflow device placement
Total reward
Number of actions
1e6
Scotch
Single GPU
Human expert
PG-LSTM
PG-GCN
Circuit design
Total reward
Number of actions
1e4
Random
MACE
BO
Human
ES
DDPG
DDPG-GCN
CDN memory caching
1e4
Total reward
Number of actions
1e7
Random
LRU
Offline optimal
A2C
(e) Queue management
(f) TF device placement
(g) Circuit design
(h) CDN memory caching
Multi-dimensional indexing
Reward relative to baseline
Number of actions
Fixed layout
A2C (confined search space)
Account region assignment
Total reward
Number of actions
1e8
Local heuristic
Thompson sampling
DQN
Load balancing
1e6
Testing total reward
Number of actions
1e8
Greedy
A2C
A2C (grad clip)
Switch scheduling
1e5
Testing total reward
Number of actions
1e7
MWM-0.1
MWM-0.5
MWM-1
A2C
(i) Multi-dim indexing
(j) Account assignment
(k) Load balancing
(l) Switching scheduling

Contribution

- State-of-the-art models for RL in circuit design uses Graph Convolutional Neural Networks.
- Apply World Models to circuit design problems.
- VAE to encode circuit schematics into compact latent variables.
- RNN to predict how latent space changes given component tweaks.
- Output fed to evolutionary control network (decision-making).

Expected Results

- World models able to accurately emulate circuit simulations in training.
- Evolution finds competitive policies using learnt model.
- Hopefully: outperform black-box Bayesian Optimisation in same or smaller amount of number of steps.
- Leverage Park's standardized benchmarks to compare techniques.

Progress

- Literature review, both on current approaches to optimizing circuit design and implementations of World Models.
- Explored Park's documentation and existing benchmarks.
- To Do:
  - Implementation + Evaluation
  - If given the time, try combine GCNN with World Model or/and expand the Controller model beyond a simple evolution-based optimization strategy.
  - Write up.