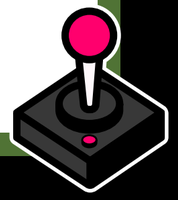
MODEL BASED RL FOR ATARI

START MENU

Introdução

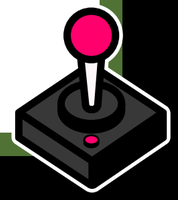
- Como modelos de vídeo permitem o aprendizado no Atari Learning Environment (ALE)
- Restrição de 100 mil intervalos de tempo (2 horas de jogo)
- Não se tinha obtido resultados competitivos dos modelos existentes anteriormente comparado a RL livre de modelo.
- “Até agora, não houve nenhuma demonstração clara de planejamento bem-sucedido com um modelo aprendido no ALE”



Introdução

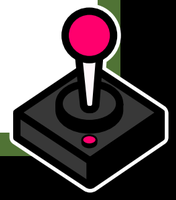
- Primeiro sistema de modelos capaz de lidar com vários jogos do benchmark do ALE

- Experimentadas diversas técnicas estocásticas de previsão de vídeo
  - Novo modelo baseado em variáveis latentes discretas



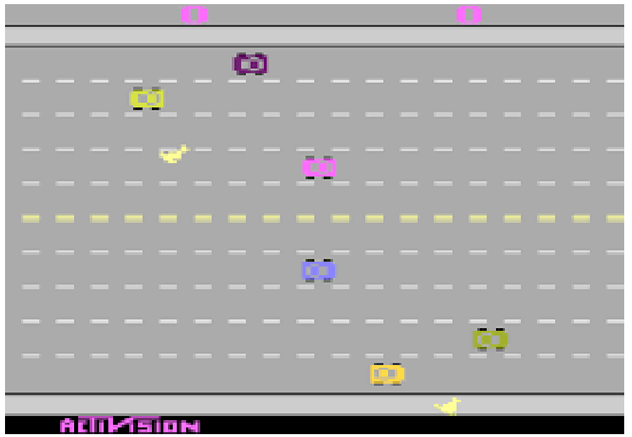
Introdução

- Simulated Policy Learning (SimPLe)
  - Utiliza as técnicas de previsão de vídeo
  - Treina uma política para jogar o jogo dentro do modelo aprendido
  - Várias iterações para adicionar informações ao dataset
  - Política feita para coletar mais dados no jogo original
  - Joga com sucesso no ambiente real



Introdução

- Simulated Policy Learning (SimPLe)
  - Significativamente mais eficiente que o algoritmo Rainbow em estado da arte (amostragem)
  - Com 100k amostras, atinge pontuações que exigem ao menos o dobro de amostras no Rainbow em mais da metade dos jogos
  - Mais de 10x mais eficiente no jogo Freeway, no melhor caso

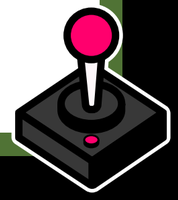


Simulated Policy Learning (SimPLe)

- A aprendizagem por reforço é formalizada em processos de decisão de Markov (MDP)
  - MDP é uma tupla (S, A, P, r, γ),
  - S é o espaço do estado
  - A é o conjunto de ações disponíveis para o agente
  - P é o kernel de transição desconhecido
  - r é a função de recompensa
  - $\gamma \in (0, 1)$ é o fator de desconto

Simulated Policy Learning (SimPLe)

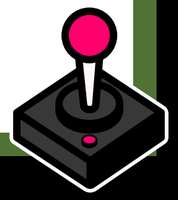
- MDPs são ambientes
- Ambientes não têm acesso direto ao estado
- São usadas observações visuais (imagens RGB 210 × 160)
- Uma única imagem não determina o estado
- Observação feita a partir de 4 frames consecutivos
- Agente RL interage com MDP emitindo ações de acordo com uma política
- política π é um mapeamento de estados para distribuições de probabilidade sobre A
- Qualidade de uma política medida por uma função



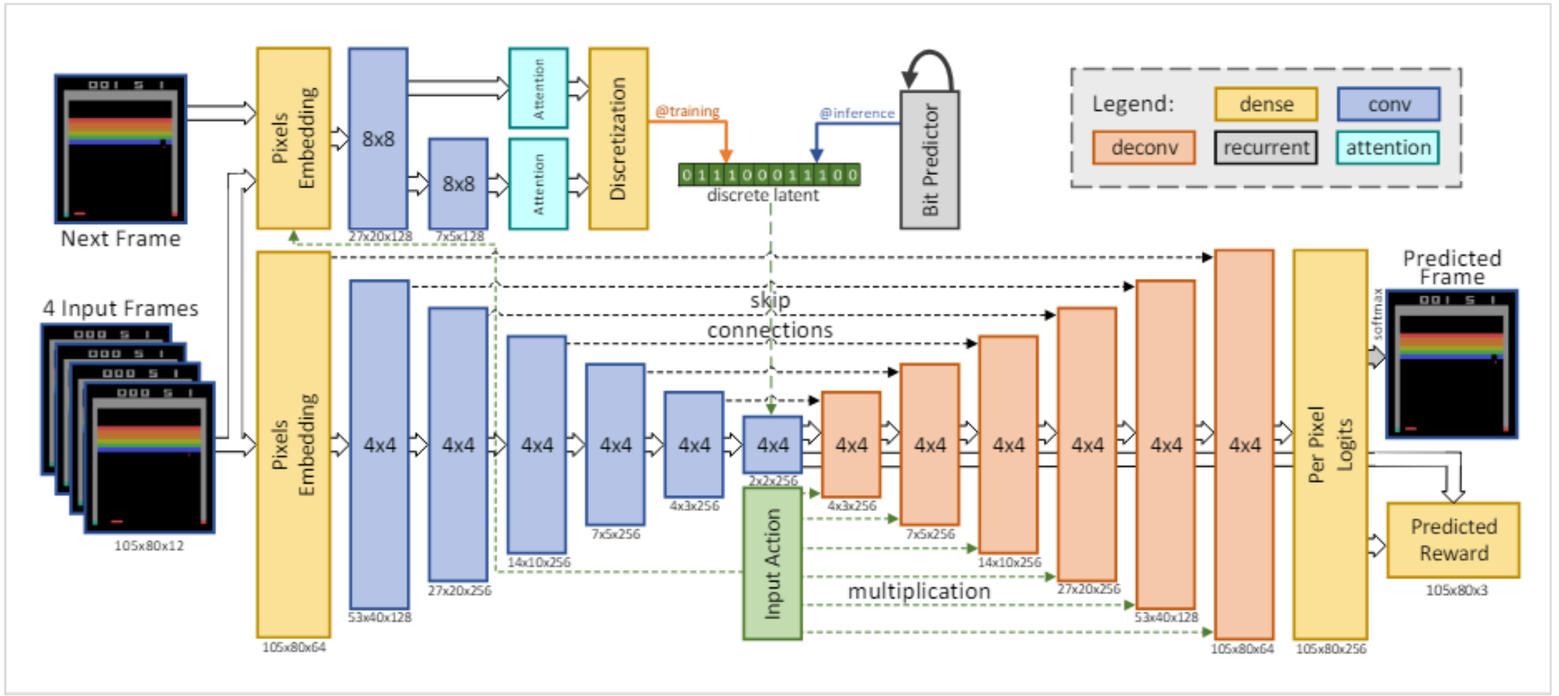
- Encontrar uma política que maximiza a função de valor desde o início de o jogo
- Além de env, há env′ (modelo global)
  - compartilha o espaço de ação e recompensa espaço com env
  - produz observações visuais no mesmo formato (treinado para imitar env)
- Objetivo é treinar uma política π usando um ambiente simulado env′ para que π atinja um bom desempenho no ambiente original
  - mínimo de interações possíveis com env
  - dados vem de implementações aleatórias de env
  - uso do método iterativo apresentado no Algoritmo 1 para complementar a captura de aspectos do ambiente

Algorithm 1: Pseudocode for SimPLe

Initialize policy π
Initialize model parameters θ of env'
Initialize empty set $\mathbf{D}$
while not done **do**
  ▷ collect observations from real env.
  $\mathbf{D} \leftarrow \mathbf{D} \cup \text{COLLECT}(env, \pi)$
  ▷ update model using collected data.
  $\theta \leftarrow \text{TRAIN\_SUPERVISED}(env', \mathbf{D})$
  ▷ update policy using world model.
  $\pi \leftarrow \text{TRAIN\_RL}(\pi, env')$
end while



Arquitetura do modelo

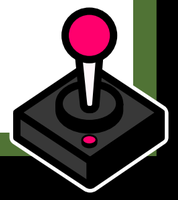


Política de Treinamento

- Proximal policy optimization (PPO) com γ = 0,95.
- O algoritmo gera implementações no ambiente simulado env′ e os utiliza para melhorar a política π.
- Principal dificuldade reside nas imperfeições do modelo que se agravam ao longo do tempo.
  - Implementações mais curtas de env′ para mitigar esse problema.
- Normalmente, a cada N = 50 passos, são criadas amostras uniformes do estado inicial do buffer D e env′ é reiniciado.

Algorithm 1: Pseudocode for SimPLe

Initialize policy π
Initialize model parameters θ of env'
Initialize empty set $\mathbf{D}$
while not done **do**
  ▷ collect observations from real env.
  $\mathbf{D} \leftarrow \mathbf{D} \cup \text{COLLECT}(env, \pi)$
  ▷ update model using collected data.
  $\theta \leftarrow \text{TRAIN\_SUPERVISED}(env', \mathbf{D})$
  ▷ update policy using world model.
  $\pi \leftarrow \text{TRAIN\_RL}(\pi, env')$
end while



Política de Treinamento

- O uso de implementações curtas pode ter um efeito ruim, pois o PPO não tem como inferir efeitos mais longos que a duração da implementação.
- Para aliviar esse problema, na última etapa de uma implementação, é adicionada à recompensa a avaliação da função de valor.

Algorithm 1: Pseudocode for SimPLe

Initialize policy π
Initialize model parameters θ of env'
Initialize empty set $\mathbf{D}$
while not done **do**
  ▷ collect observations from real env.
  $\mathbf{D} \leftarrow \mathbf{D} \cup \text{COLLECT}(env, \pi)$
  ▷ update model using collected data.
  $\theta \leftarrow \text{TRAIN\_SUPERVISED}(env', \mathbf{D})$
  ▷ update policy using world model.
  $\pi \leftarrow \text{TRAIN\_RL}(\pi, env')$
end while

Política de Treinamento

- O loop principal no Algoritmo 1 é iterado 15 vezes
- O modelo global é treinado para 45 mil passos na primeira iteração e 15 mil passos em cada uma das seguintes.
- Treinamento mais curto em iterações posteriores não degradam o desempenho porque o modelo global após a primeira iteração já captura parte da dinâmica do jogo e só precisa ser estendido a situações novas.

Algorithm 1: Pseudocode for SimPLe

Initialize policy π
Initialize model parameters θ of env'
Initialize empty set $\mathbf{D}$
while not done **do**
  ▷ collect observations from real env.
  $\mathbf{D} \leftarrow \mathbf{D} \cup \text{COLLECT}(env, \pi)$
  ▷ update model using collected data.
  $\theta \leftarrow \text{TRAIN\_SUPERVISED}(env', \mathbf{D})$
  ▷ update policy using world model.
  $\pi \leftarrow \text{TRAIN\_RL}(\pi, env')$
end while

Política de Treinamento

- Em cada uma das iterações, o agente é treinado dentro do modelo global mais recente utilizando PPO.
- Cada época da PPO possui 16 agentes paralelos coletando 25, 50 ou 100 passos do ambiente simulado env′
- O número de épocas PPO é $z \cdot 1000$, onde z é igual a 1 em todos as passagens exceto na última (onde $z = 3$) e nas passagens número 8 e 12 (onde $z = 2$).
- No processo de treinamento o agente realiza 15,2 milhões de interações com o ambiente simulado env′.

Algorithm 1: Pseudocode for SimPLe

Initialize policy π
Initialize model parameters θ of env'
Initialize empty set $\mathbf{D}$
while not done **do**
  ▷ collect observations from real env.
  $\mathbf{D} \leftarrow \mathbf{D} \cup \text{COLLECT}(env, \pi)$
  ▷ update model using collected data.
  $\theta \leftarrow \text{TRAIN\_SUPERVISED}(env', \mathbf{D})$
  ▷ update policy using world model.
  $\pi \leftarrow \text{TRAIN\_RL}(\pi, env')$
end while

Experimentos – Eficiência da amostra

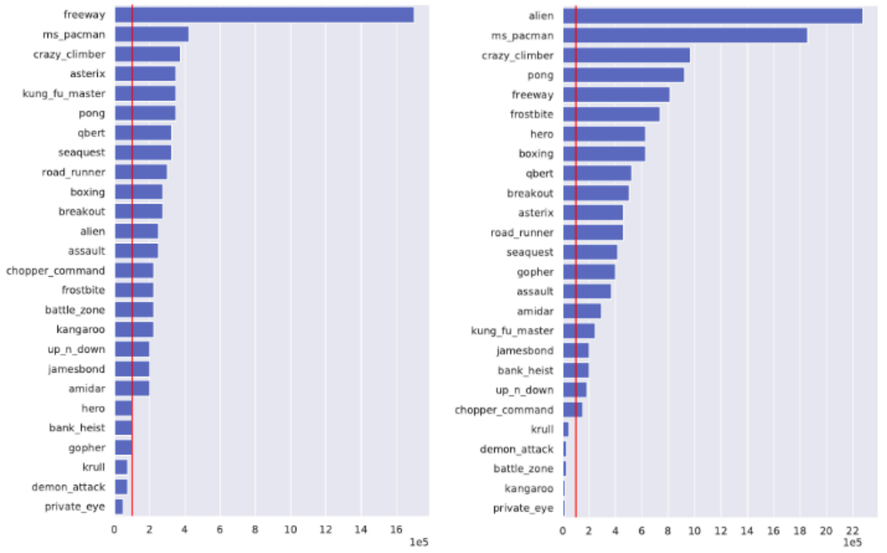


Figure 3: Comparison with Rainbow and PPO. Each bar illustrates the number of interactions with environment required by Rainbow (left) or PPO (right) to achieve the same score as our method (SimPLe). The red line indicates the 100K interactions threshold which is used by the our method.

Experimentos – Número de frames

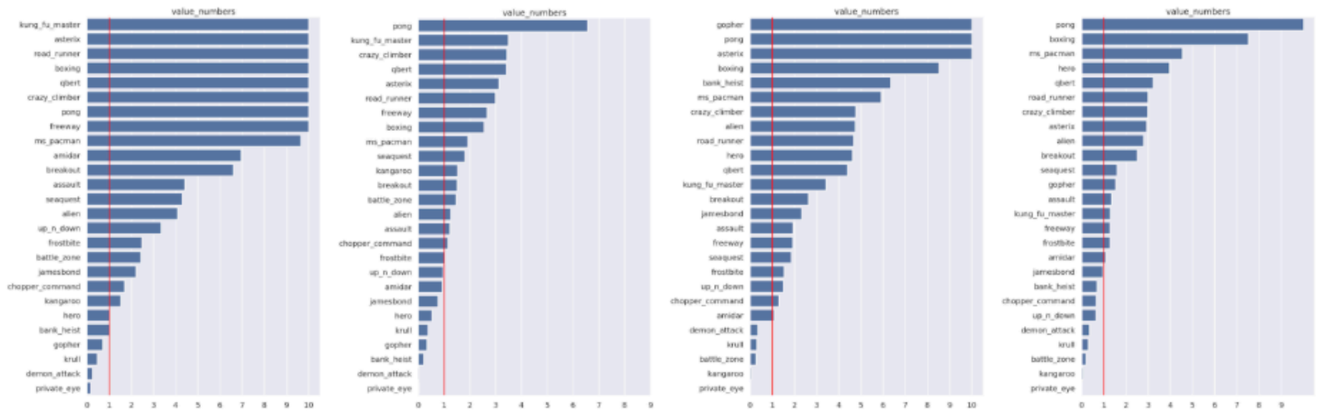


Figure 4: Fractions of Rainbow and PPO scores at different numbers of interactions calculated with the formula $(SimPLe\_score@100K - random\_score)/(baseline\_score - random\_score)$*; if denominator is smaller than 0, both nominator and denominator are increased by 1. From left to right, the baselines are: Rainbow at 100K, Rainbow at 200K, PPO at 100K, PPO at 200K. SimPLe outperforms Rainbow and PPO even when those are given twice as many interactions.*

Experimentos – Estocasticidade do ambiente

- O Atari é um ambiente determinístico, porém o SimPLe é estocástico, dado que utiliza 4 frames.
- A nível da estocasticidade depende do jogo.
- No Kung Fu Master, por exemplo, após derrotar um set de oponentes o fundo permanecerá igual até que o próximo set apareça.

Experimentos – Estocasticidade do ambiente

- Dada a estocasticidade do SimPLe, ele pode ser usado em ambientes verdadeiramente estocásticos.
- Para demonstrar isso, foi realizado um experimento onde o pipeline completo (tanto o modelo mundial quanto a política) foi treinado na presença de sticky actions.
  - Modelo global leva em conta a rigidez maior das ações
  - Resultados finais semelhantes nos casos determinísticos, em maioria, mesmo sem nenhum ajuste.

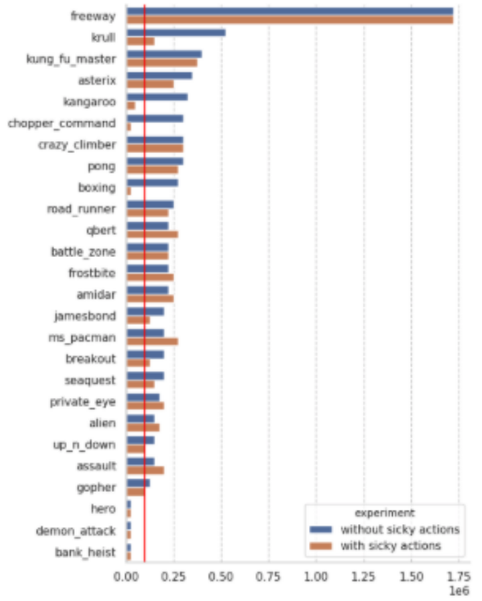


Figure 6: Impact of the environment stochasticity. The graphs are in the same format as Figure 3: each bar illustrates the number of interactions with environment required by Rainbow to achieve the same score as SimPLe (with stochastic discrete world model) using 100k steps in an environment with and without sticky actions.

Experimentos – Ablações

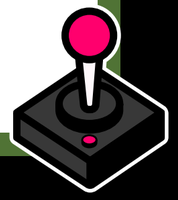
- Arquitetura do modelo e hiperparâmetros
  - Entre as opções para o modelo global, o modelo discreto estocástico proposto teve melhor desempenho por uma margem significativa.
  - A duração do treinamento do modelo global foi outro parâmetro importante:
    - Quanto mais longo melhor, porém o alto custo fez com que ele fosse reduzido em outras ablações.
  - Tamanho do treinamento com N = 50.
    - N = 25 tem resultado comparável.
    - N = 100 é um pouco pior.
  - Fator de desconto definido como $\gamma = 0{,}99$ (pode ser especificado como outro).
    - $\gamma = 0{,}95$ um pouquinho melhor, mas nada muito significativo

Experimentos – Ablações

- Iterações baseadas em modelos
  - O processo iterativo de treinar o modelo, a política e coletar dados é crucial, quando a coleção de dados aleatórios não é suficiente.
  - Na análise entre os jogos, foi quantificado em quais houve melhor resultado em iterações tardias do treinamento.
  - Em alguns casos, boas políticas puderam ser aprendidas rapidamente.
    - Pode ter acontecido devido a alta variabilidade do treinamento.
    - Sugere que pode ser feito um treinamento ainda mais rápido (com menos de 100k passos)

Experimentos – Ablações

- Começos aleatórios (random starts)
  - Implementações curtas são cruciais para mitigar a sequência de erros do modelo.
  - Para garantir a exploração, o SimPLe inicia implementações a partir de estados selecionados aleatoriamente retirados do buffer de dados real D.



Conclusão

- Em muitos casos, o número de amostras necessário para que os métodos anteriores aprendam a atingir o mesmo valor de recompensa é várias vezes maior.
- Pode ser usado em ambientes amplamente estocásticos.
- Pontuações finais são, em geral, inferiores às dos melhores métodos livres de modelo em estado da arte.
- O desempenho do método do artigo geralmente variou substancialmente entre diferentes execuções do mesmo jogo.

Obrigado!