A3C: Deep Reinforcement Learning aplicado a Multitarefas

MARCO ANTÔNIO FERREIRA BIRCK[1]; ULISSES BRISOLARA CORRÊA[2]; RICARDO MATSUMURA DE ARAÚJO[3]

[1]*Universidade Federal de Pelotas – mafbirck@inf.ufpel.edu.br*

[2]*Universidade Federal de Pelotas – ubcorrea@inf.ufpel.edu.br*

[3]*Universidade Federal de Pelotas – ricardo@inf.ufpel.edu.br*

1. INTRODUÇÃO

A aprendizagem por reforço multitarefa (Taylor and Stone, 2009) é uma área de pesquisa fértil na qual busca-se que modelos de aprendizagem de máquina aprendam múltiplas tarefas de maneira eficiente e factível, mais especificamente no escopo deste trabalho Multi Tarefas em Deep Reinforcement Learning.

A área de Deep Reinforcement Learning tem ganhado grande atenção da comunidade de aprendizado de máquina, onde redes neurais profundas são utilizadas em problemas de aprendizado por reforço, conseguindo resultados muito acima dos alcançados por métodos tradicionais. Conquistando o estado da arte em muitos desses problemas e até mesmo superando o nível humano (Mnih, 2013). Uma solução que parece intuitiva para o problema de Multi-Tarefas é treinar a rede para executar uma tarefa e logo após ajustar os pesos para uma próxima tarefa, entretanto, quando aplicado ao regime MT dessa maneira direta, verifica-se que há uma degradação da capacidade na tarefa anterior, sendo assim, há a necessidade de mecanismos que mitiguem esse fenômeno (Kirkpatrick, 2017).

O presente trabalho tem por objetivo estudar a possibilidade do uso do A3C(Asynchronous Advantage Actor-Critic) aplicado ao aprendizado de multitarefa. O algoritmo A3C foi proposto por (Mnih, 2016) como uma alternativa mais eficaz do que as soluções que haviam sido propostas até então. O algoritmo A3C ao utilizar de múltiplas instâncias do mesmo jogo que são apresentadas ao modelo visa proporcionar uma esparsidade de dados durante o treinamento, bem como fazer bom uso do paralelismo para reduzir o gasto computacional e melhorar a eficácia do treinamento. Dada essa característica multi-thread do algoritmo, avaliou-se a capacidade do A3C de praticar o regime de multi-tarefa. O algoritmo é então modificado para ser treinado com metade de seus workers/threads na task A e metade dos seus workers/threads na task B, essas threads constantemente se comunicam com o modelo global, que aprenderá a política híbrida. Sendo esse modelo alimentado de maneira assíncrona com as atualizações recebidas pelas suas instâncias distribuídas, mais especificamente 2 threads para cada ambiente.

2. METODOLOGIA

O presente trabalho visou analisar como o A3C em regime multi-tarefa impacta a performance nas tarefas individuais. Aproveitou-se a natureza distribuída do algoritmo para permitir que múltiplos e diferentes data-streams fossem apresentados para o mesmo modelo de maneira paralela e assincrona. Para que isso acontecesse dividimos as múltiplas threads intrínsecas ao A3C original para múltiplos jogos e observamos o efeito de tal modificação.

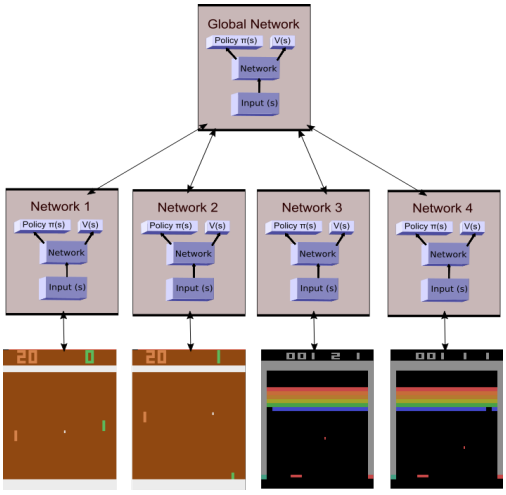

Figure (1): Procedimento de treinamento do A3C hibrido.

Nomeou-se o método como Hybrid Asynchronous Advantage Actor-Critic Training (Hybrid A3C), dada a capacidade de aprender ambientes de maneira híbrida, em tempo real e em conjunto. O A3C original que dá fomento a esse trabalho é um algoritmo multi-thread que usa um modelo actor-critic. O principal ganho ao usar multi-threads advém do fato de que o modelo é apresentado a uma maior variabilidade de estados possíveis em uma mesma tarefa/ambiente, fazendo com que ele tenha acesso a estados esparsos e generalize mais rapidamente.

3. RESULTADOS E DISCUSSÃO

Á análise do método proposto é realizada a partir de dois conjuntos de tasks, partindo disso então fez-se uma análise de como os ambientes que compartilham de uma certa similaridade semântica afetam uns aos outros no regime multitarefa:

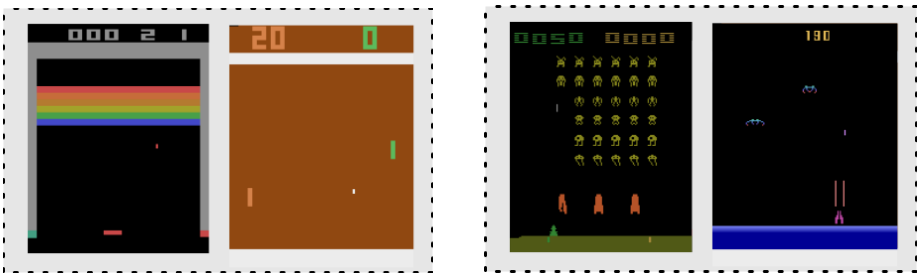

Figure (2): Screenshots for the four selected games. From left to right: Breakout, Pong, Space Invaders, Demon Attack.

Para cada um dos pares acima descritos é executado uma rodada, o modelo é treinado por 50 Milhões de steps, que corresponde a 200 Milhões de frames(1 step = 4 frames), é feito uma rodada da maneira canônica, onde o modelo é solicitado a aprender somente uma tarefa, e uma de maneira híbrida como reportado nos resultados das figuras 3 e 4.

Na figura (3) temos os resultados obtidos pelo par Breakout e Pong. Tem-se os resultados do treino independente e do treino híbrido, sendo possível perceber que o treinamento acaba por impulsionar o treinamento do Breakout, dado que a política híbrida atinge rewards muito maiores do que o treinamento do zero, o Pong por outro lado tem o início do aprendizado prejudicado, mas ao final da rodada acaba por chegar a uma política melhor do que o método canônico de treinamento.

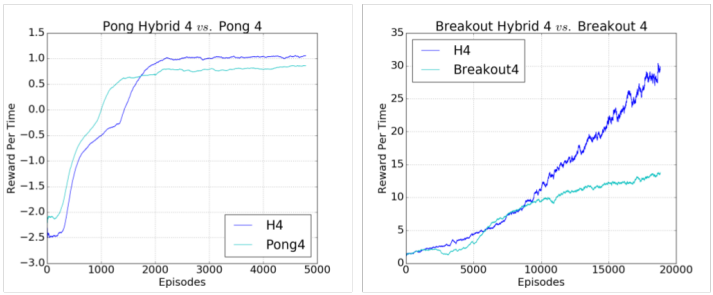

Figura (3): Resultados para treinamento sozinho e híbrido para:

(a) Pong e (b) Breakout. H4 representa o procedimento híbrido, enquanto Breakout4 e Pong4 são 4 threads dedicadas ao jogo solo. Foi utilizada Exponential Weighted Moving Average.

Na figura (4) temos os resultados do par Space Invaders e Demon Attack. Na task Demon Attack podemos ver uma melhora no treinamento quando no método híbrido. Diferentemente na tarefa Space Invaders temos uma eficiência discutivelmente igual para os dois procedimentos.

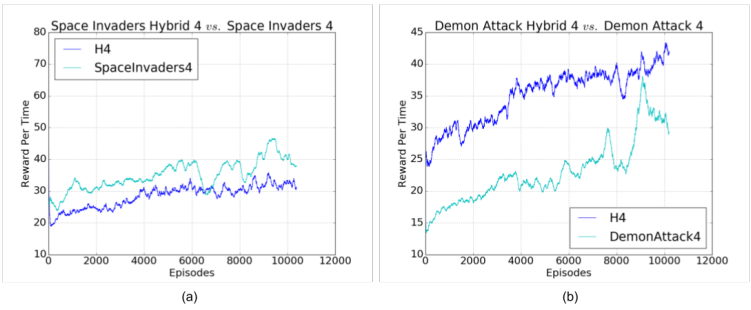

Figura (4): Resultados para treinamento sozinho e híbrido para:

(a) Space Invaders e (b) Demon Attack. H4 representa o procedimento hibrido, enquanto Breakout4 e Pong4 são 4 threads dedicadas ao jogo solo. Foi utilizada Exponential Weighted Moving Average.

Os resultados supracitados nos levam a análise de que o algoritmo A3C consegue alcançar o regime de Multitarefas, e curiosamente faz com que tarefas similares alcancem ganho assintótico, deixando ainda mais evidente a necessidade de aprofundamento dos experimentos.

4. CONCLUSÕES

Neste trabalho reporta-se resultados iniciais da utilização do algoritmo A3C para multitarefa em jogos de Atari treinado on-the-fly, que diferentemente de propostas anteriores (Rusu, 2015), (Parisoto, 2015), não necessita de modelos previamente masterizados no jogos em questão, pode-se notar baseados nos resultados obtidos, que o algoritmo é capaz de executar o regime multitarefa sem degradação de performance aparente, além disso é possível visualizar melhoras em uma das tarefas a qual o modelo é submetido no regime multitarefa.

Mais resultados são necessários para um entendimento melhor do fenômeno, além de propostas de isolamento do mesmo para que seja utilizado como método regularizador de treinamentos individuais ou de multitarefa que viriam a permitir ganhos assintóticos consideráveis.

Além disso, uma melhorar análise do efeito das similaridades semânticas é necessário para verificar a necessidade ou não dela no regime de treinamento, ou até mesmo que essa similaridade possa ser utilizada como parte inerente ao algoritmo de aprendizagem em multitarefa em trabalhos futuros.

5. REFERÊNCIAS BIBLIOGRÁFICAS

Artigo

TAYLOR, Matthew E.; STONE, Peter. Transfer learning for reinforcement learning domains: A survey. **Journal of Machine Learning Research**, v. 10, n. Jul, p. 1633-1685, 2009.

MNIH, Volodymyr et al. Playing atari with deep reinforcement learning. **arXiv preprint arXiv:1312.5602**, 2013.

MNIH, Volodymyr et al. Human-level control through deep reinforcement learning. **Nature**, v. 518, n. 7540, p. 529-533, 2015.

MNIH, Volodymyr et al. Asynchronous methods for deep reinforcement learning. In: **International Conference on Machine Learning**. 2016. p. 1928-1937.

PARISOTTO, Emilio; BA, Jimmy Lei; SALAKHUTDINOV, Ruslan. Actor-mimic: Deep multitask and transfer reinforcement learning. **arXiv preprint arXiv:1511.06342**, 2015.

RUSU, Andrei A. et al. Policy distillation. **arXiv preprint arXiv:1511.06295**, 2015.

KIRKPATRICK, James et al. Overcoming catastrophic forgetting in neural networks. **Proceedings of the National Academy of Sciences**, p. 201611835, 2017.