Airline Revenue Management using A2C

Tyler Sam and Sam Tan

May 17, 2021

(Synchronous) Advantage Actor Critic

- Uses the advantage function in the policy gradient:

$$\nabla_\theta J(\theta) \approx E_\tau[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) A_w(s_t, a_t)]$$

(Synchronous) Advantage Actor Critic

- Uses the advantage function in the policy gradient:

$$\nabla_\theta J(\theta) \approx E_\tau[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) A_w(s_t, a_t)]$$

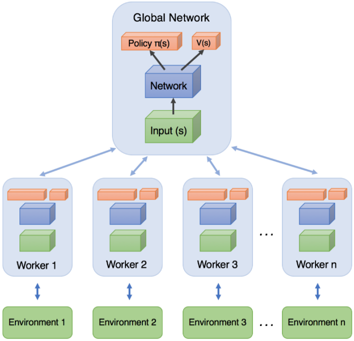
- Inherent variance reduction as the value function is treated as a baseline function.

(Synchronous) Advantage Actor Critic

- Uses the advantage function in the policy gradient:

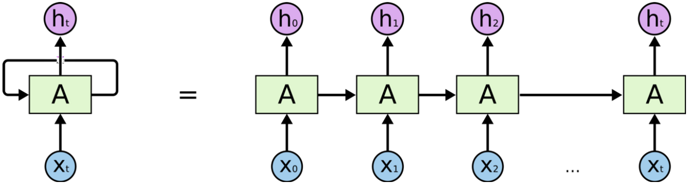
$$\nabla_\theta J(\theta) \approx E_\tau[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) A_w(s_t, a_t)]$$

- Inherent variance reduction as the value function is treated as a baseline function.



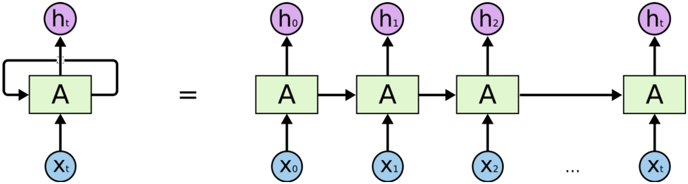
Recurrent Neural Networks

- Unlike traditional feed-forward neural networks, recurrent neural networks (RNN) use previous information to produce an output.



Recurrent Neural Networks

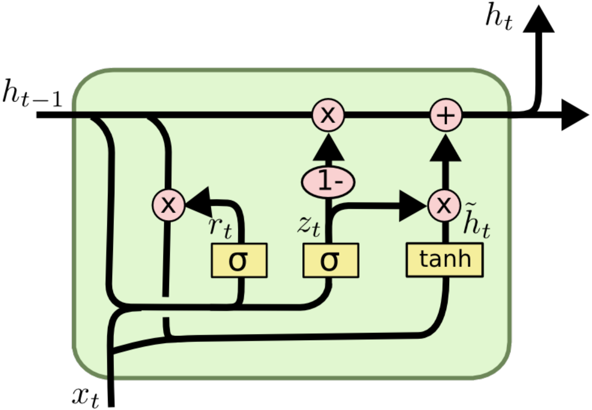
- Unlike traditional feed-forward neural networks, recurrent neural networks (RNN) use previous information to produce an output.



- *The LSTM-Based Advantage Actor-Critic Learning for Resource Management in Network Slicing With User Mobility* used A2C with a Long Short Term Memory (LSTM) RNN to get good results.

GRU

- Gated Recurrent Units (GRUs) are similar to LSTMs but have fewer gates in each cell.



Architectures

- Default MLP from `stable-baselines3`, separate for policy and value.

Architectures

- Default MLP from `stable-baselines3`, separate for policy and value.

- One shared layer between policy and value, then two layers each.

Architectures

- Default MLP from `stable-baselines3`, separate for policy and value.
- One shared layer between policy and value, then two layers each.
- GRU for value, MLP for policy.

Hyperparameters

- `learning_rate = 0.0007`
- `n_steps = 5`
- Use RMSprop for optimization.

Hyperparameters

- `learning_rate = 0.0007`
- `n_steps = 5`
- Use RMSprop for optimization.
- Larger network sizes did not seem to improve results while adding runtime.

Hyperparameters

- `learning_rate = 0.0007`
- `n_steps = 5`
- Use RMSprop for optimization.
- Larger network sizes did not seem to improve results while adding runtime.
- Learn for 500,000 steps, evaluate over 500 simulations.

Results

L	τ	c	Base MLP	Shared	GRU	DBPC	PPO
3	20	2	364.29(150.89)	317.01(139.27)	188.17(108.42)	567.78(20.54)	764.39(6.98)
	50	6	845.68(247.119)	706.30(235.96)	737.47(228.16)	1759.91(33.76)	1790.66(13.81)
	100	12	1546.01(341.09)	1556.57(342.26)	1543.84(336.87)	3730.04(53.87)	3744.49(20.99)
	200	24	2913.53(488.92)	2901.09(487.35)	3107.65(521.16)	7683.04(70.09)	7551.33(33.77)
	500	61	8189.19(850.59)	7257.78(754.45)	8089.46(832.38)	19793.50(132.23)	20884.33(65.56)
5	20	1	116.18(72.08)	106.38(72.87)	93.72(69.54)	486.92(17.86)	654.81(9.28)
	50	4	585.96(252.78)	608.05 (243.24)	654.85(250.75)	1874.34(36.70)	1233.67(17.02)
	100	8	1489.12(457.99)	1641.90(449.97)	1650.27(459.54)	3905.97(50.49)	3237.69(31.58)
	200	16	3660.86(792.38)	3942.27(775.49)	3625.08(742.68)	8109.55(73.00)	7369.48(50.79)
	500	42	8698.46(1266.06)	8292.50(1140.22)	10985.99(1419.44)	21189.10(125.73)	21938.52(95.40)

Why so bad?

- Sensitivity to hyperparameters, e.g. learning rate, architecture.

Why so bad?

- Sensitivity to hyperparameters, e.g. learning rate, architecture.

- Computationally expensive.

Why so bad?

- Sensitivity to hyperparameters, e.g. learning rate, architecture.

- Computationally expensive.
  - We observed slow convergence, high reward variance.

Why so bad?

- Sensitivity to hyperparameters, e.g. learning rate, architecture.

- Computationally expensive.
  - We observed slow convergence, high reward variance.
  - However, our results showed promise, as the **maximum** observed values were more competitive.

Conclusions and Future Directions

- A2C seems to have limitations when applied to Airline Revenue Management; PPO may be a better choice.

Conclusions and Future Directions

- A2C seems to have limitations when applied to Airline Revenue Management; PPO may be a better choice.
- More computational power may allow for better results with A2C.

Conclusions and Future Directions

- A2C seems to have limitations when applied to Airline Revenue Management; PPO may be a better choice.
- More computational power may allow for better results with A2C.
- Try hyperparameter tuning, different architectures; there is some evidence that GRUs may improve performance.

References I

- Simple reinforcement learning with tensorflow part 8: Asynchronous actor-critic agents (a3c).
- Understanding lstm networks.
- Introduction to recurrent neural networks.