Rainbow

Reading

- Hessel, Matteo, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. "Rainbow: Combining improvements in deep reinforcement learning." In Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1. 2018.

Overview

- A lot of extensions to the standard deep Q-learning (DQN) algorithm have been made
  - Double DQN (DDQN)
  - Prioritized Experience Replay (PER)
  - N-step learning
  - Dueling DDQN
  - Noisy DQN
  - Distributional DQN
- A lot of them can be combined in a single framework
- Rainbow combines them and shows that the combination performs much better than each individual approach

Double DQN

Rensselaer

- The idea is to have two critics
  - One critic is used to select the maximizing action
  - One critic is used to bootstrap the returns
  - Alleviates maximization bias
- Suppose $Q_{\boldsymbol{\theta}_1}$ is used to determine the max Q value, i.e.,

$$A^* = \operatorname{argmax}_a Q_{\boldsymbol{\theta}_1}(S_t, a)$$

- And $Q_{\boldsymbol{\theta}_2}$ is used to get the actual value of A^*, i.e.,

$$Q_{\boldsymbol{\theta}_2}(S_t, A^*) = Q_{\boldsymbol{\theta}_2}\left(S_t, \underset{a}{\operatorname{argmax}}\, Q_{\boldsymbol{\theta}_1}(S_t, a)\right)$$

- E.g., $Q_{\boldsymbol{\theta}_1}$ is trained using loss

$$\left(R_t + \gamma Q_{\boldsymbol{\theta}_2}\left(S_{t+1}, \underset{a}{\operatorname{argmax}}\, Q_{\boldsymbol{\theta}_1}(S_t, a)\right) - Q_{\boldsymbol{\theta}_1}(S_t, A_t)\right)^2$$

Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." Proceedings of the AAAI conference on artificial intelligence. Vol. 30. No. 1. 2016.

Prioritized Experience Replay (PER)

- DQN samples uniformly from the replay buffer
  - This treats all experiences/transitions as equally important
  - A lot of them are similar and not relevant
  - PER alleviates this issue by assigning different weights to different transitions when sampling
- Transition t is weighted according to its current DDQN loss

$$w_t = \left| R_t + \gamma Q_{\boldsymbol{\theta}_2}\left(S_{t+1}, \underset{a}{\operatorname{argmax}}\, Q_{\boldsymbol{\theta}_1}(S_t, a)\right) - Q_{\boldsymbol{\theta}_1}(S_t, A_t) \right|^{\omega}$$

  - where ω is a hyperparameter
- Transitions with larger loss more likely to be selected
  - They are more important to learn
  - Same for new transitions

Schaul, Tom, John Quan, Ioannis Antonoglou, and David Silver. "Prioritized experience replay." arXiv preprint arXiv:1511.05952 (2015).

Multi-step Learning

- Already seen this idea also
  - Sutton&Barto book describes it well
- Consider the n-step return

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n}$$

- Bootstrap the n-label and minimize loss

$$\left(G_{t:t+n} + \gamma Q_{\boldsymbol{\theta}_2}\left(S_{t+n+1}, \operatorname*{argmax}_a Q_{\boldsymbol{\theta}_1}(S_{t+n+1}, a)\right) - Q_{\boldsymbol{\theta}_1}(S_t, A_t)\right)^2$$

- Can learn faster with the right choice of hyperparameter n
- Paper below describes an interesting asynchronous version (A3C)
  - Don't have time to cover

Mnih, Volodymyr, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous methods for deep reinforcement learning." In International conference on machine learning, pp. 1928-1937. PMLR, 2016.

Dueling Networks

- DDQN learns the q-value for each state-action pair
  - This may lead to high variance if a state has low value, but some state-action pair has spuriously high value estimate
  - May be better to separate learning the state values from the action values
- The dueling networks method uses the advantage function

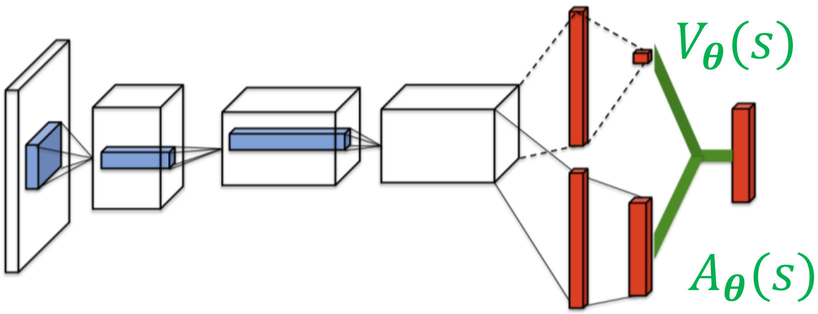
$$A_\pi(s, a) = q_\pi(s, a) - v_\pi(s)$$

  - Measures how good is the current action relative to the best
- Advantage function is a popular concept in RL
  - Used in other popular algorithms such as TRPO and PPO

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016, June). Dueling network architectures for deep reinforcement learning. In International conference on machine learning (pp. 1995-2003). PMLR.

Dueling Networks, cont'd

- Given a state input, the dueling network has one output for the state value and one output for each action's advantage



- Final output is

$$Q_{\boldsymbol{\theta}}(s,a) = V_{\boldsymbol{\theta}}(s) + \left(A_{\boldsymbol{\theta}}(s,a) - \frac{1}{|A|} \sum_{a'} A_{\boldsymbol{\theta}}(s,a') \right)$$

  - Where the average over all other actions is subtracted
    - Authors claim this modification stabilizes learning
    - Trained with standard deep double Q-learning, as usual

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016, June). Dueling network architectures for deep reinforcement learning. In International conference on machine learning (pp. 1995-2003). PMLR.

Noisy Nets

- Exploration is tricky in sparse reward settings where rewards are only received after a lot of actions (e.g., in mountain car)
  - Standard ϵ-greedy exploration does not work so well here
- Authors propose to add noisy fully connected layers

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} + \left(\left(\boldsymbol{W}_{noisy} \odot \boldsymbol{\epsilon}_w\right)\boldsymbol{x} + \left(\boldsymbol{b}_{noisy} \odot \boldsymbol{\epsilon}_b\right)\right)$$

  - which is followed by an activation as usual
  - two sets of parameters: $\boldsymbol{W}, \boldsymbol{b}$ and $\boldsymbol{W}_{noisy}, \boldsymbol{b}_{noisy}$
  - noises $\boldsymbol{\epsilon}_w$ and $\boldsymbol{\epsilon}_b$ are sampled randomly for each step
- This promotes more randomness and exploration
  - Over time, network learns to ignore noise for states where exploration is no longer needed (i.e., has strong gradients)

Fortunato, M.; Azar, M. G.; Piot, B.; Menick, J.; Osband, I.; Graves, A.; Mnih, V.; Munos, R.; Hassabis, D.; Pietquin, O.; Blundell, C.; and Legg, S. 2017. Noisy networks for exploration. CoRR abs/1706.10295.

Distributional RL

- Instead of learning the Q-value per state-action pair, authors propose to learn the full distribution of returns G_t
  - Distribution satisfies a similar Bellman equation as Q values
  - If distribution is multi-modal, this approach may stabilize learning as opposed to standard Q-learning
- Learn a discrete distribution $z_1, \ldots, z_N$ for the bootstrapped return from each state-action pair

$$\big(R_t + \gamma z_1, p_1(S_t, A_t)\big), \ldots, \big(R_t + \gamma z_N, p_N(S_t, A_t)\big)$$

  - Where the p_i are inferred from the z_i (e.g., using softmax)
- Finally, train a neural net to match its predicted distribution to the target
  - E.g., by minimizing KL divergence

Bellemare, Marc G., Will Dabney, and Rémi Munos. "A distributional perspective on reinforcement learning." In International conference on machine learning, pp. 449-458. PMLR, 2017.

Integrated Agent

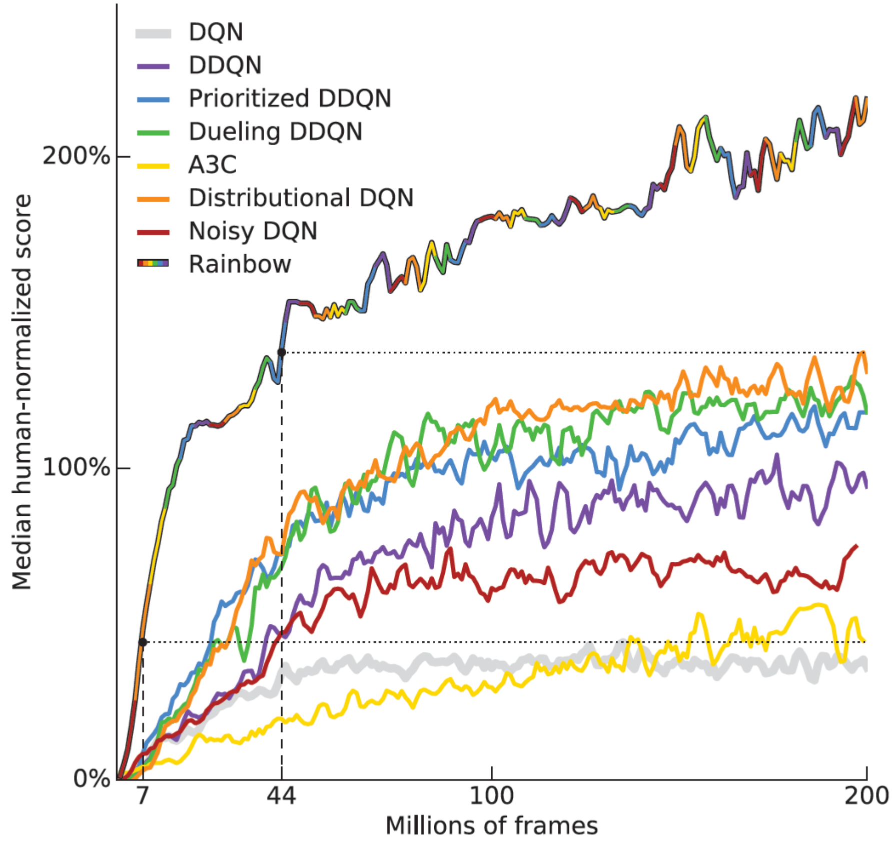
Rensselaer

- Uses distributional loss (minimize KL divergence)
- Uses n-step returns
- Uses double networks
- Uses experience replay (with distributional loss)
- Uses dueling network architecture
  - With noisy linear (fully connected) layers

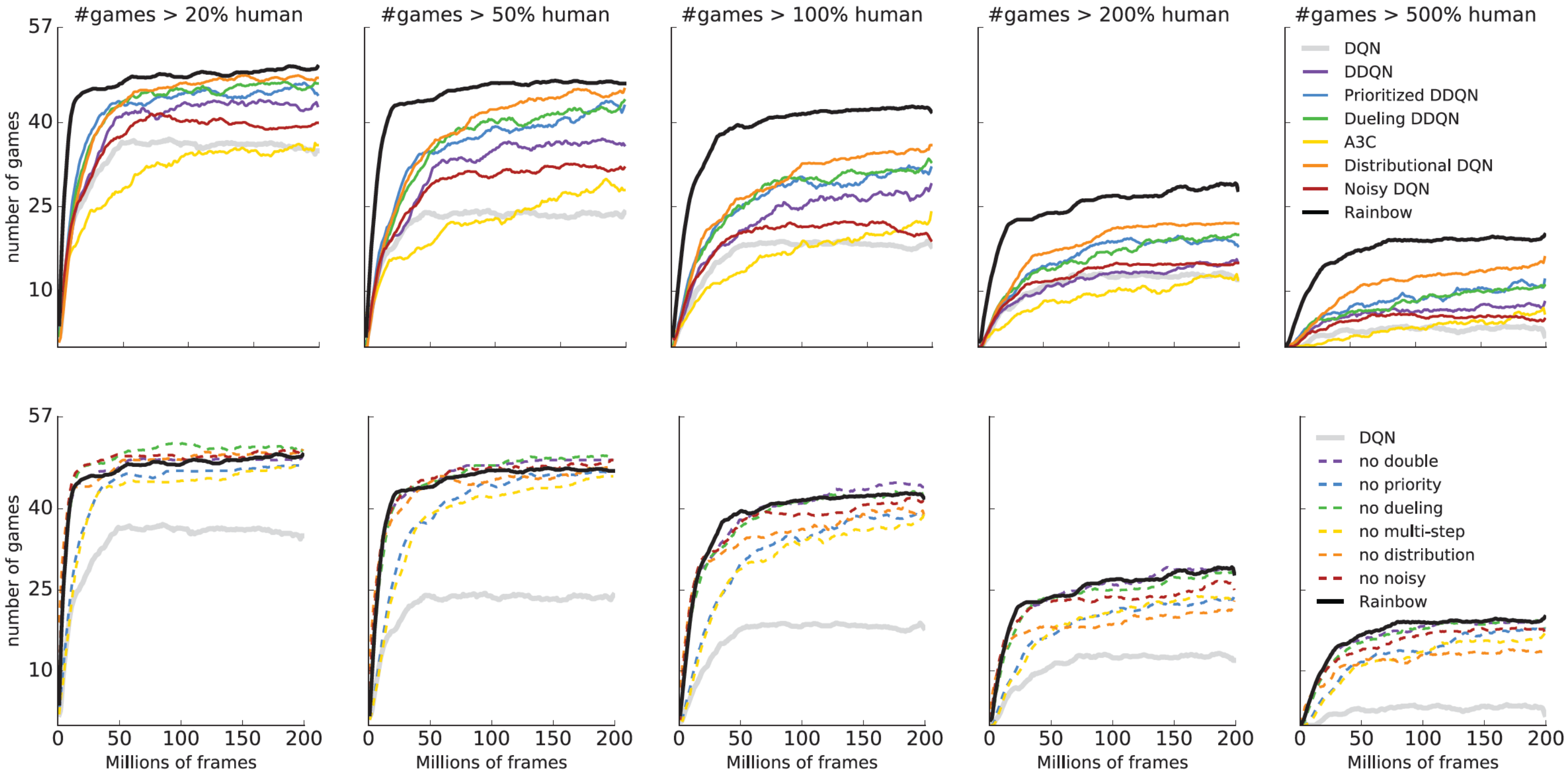
Aggregate Performance Over all 57 Atari Games

Rensselaer

- Rainbow beats all individual algorithms after 44M steps

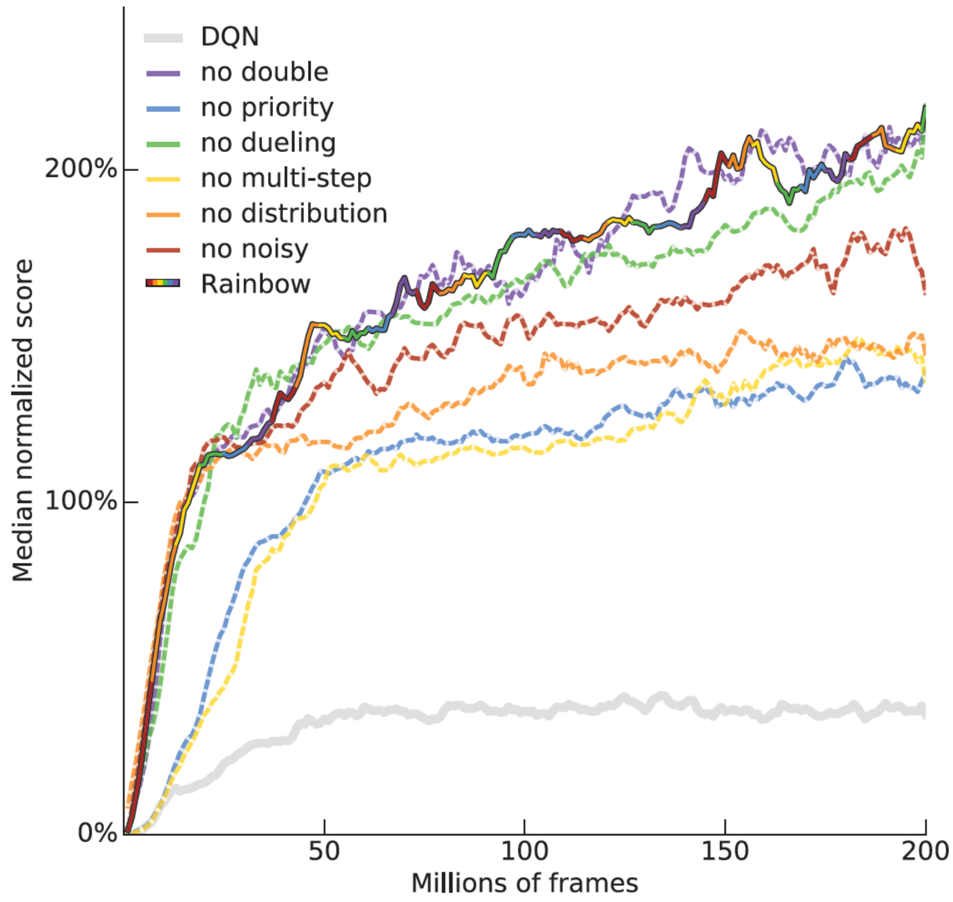


Evaluation per Game



Individual Component Importance

- Most important aspects seem to be
  - prioritized replay
  - multi-step learning
  - distributional loss
- Least important are
  - double DQN
  - dueling networks
- Noisy nets in the middle
- Some methods may have overlapping properties



Conclusion

Rensselaer

- Combining different methods does seem to bring a significant advantage
- There are improvements to Rainbow already as well
- There are also foundational RL models which supposedly can play all Atari games
  - Check out Google's Gato model
- Learning still requires an enormous amount of computation
  - So a lot of work is still left to do
    - Why do we need so much data?
    - Can we generalize from one game to another?
    - Can we guarantee safety?