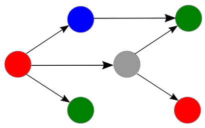
Deep Q Learning For Directed Acyclic Graph Generation

Laura D'Arcy (Cardiff), **Padraig Corcoran** (Cardiff), **Alun Preece** (Cardiff),
Krishna Kesari (Purdue, IBM US), **Raghu Ganti** (IBM US).

Objectives

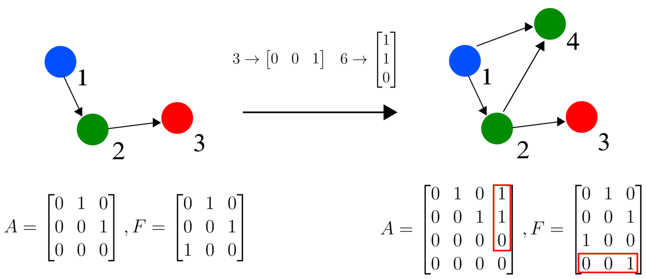
- Generate Directed Acyclic Graphs (DAGs) with Deep Reinforcement Learning (RL), specifically deep Q-learning
- Generated DAGs must have specified topology and node types, to simulate developing workflows on a distributed system within a coalition space
- No example graphs available to learn from, learning must rely on a singular reward value at the end of each attempt



Technical Challenges

- Most generation methods work with only undirected edges
- Non-Euclidean state space is highly complex
- No example graphs mean a highly sparse reward space

Approaches



- **Environment:** a topologically sorted DAG with multiple node types
- **State:** the current DAG, fed to the network as an adjacency matrix *A*, which encodes edge information, and a feature matrix *F*, which encodes each node type
- **Action:** adding a new node along with a complete set of incoming edges
- **Reward:** a positive value dependent on whether the DAG generated by the agent is isomorphic to the desired "ground truth" DAG, which is otherwise hidden to the learning agent
- We develop a graph convolutional network that encodes the state into a Q-value using concatenation:

$$\boldsymbol{H}^{l+1} = \mathrm{CONCAT}(\boldsymbol{H}^l, \boldsymbol{A}\boldsymbol{H}^l, \boldsymbol{A}^T\boldsymbol{H}^l)$$

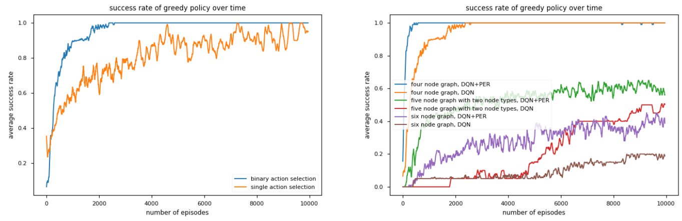
- And use the Q-learning algorithm as the loss function:

$$Q^{\pi}(S_t, A_t) = r + \gamma \max_a Q(S_{t+1}, a)$$

Military & Coalition Relevance

- Coalition systems need to be combined and used efficiently "on the fly", without extensive preparation time
- A deep RL approach can combine services to develop workflows in the form of DAGs without prior examples or human intervention
- RL handles dynamic environments better than standard ML, working well for coalitions on the edge

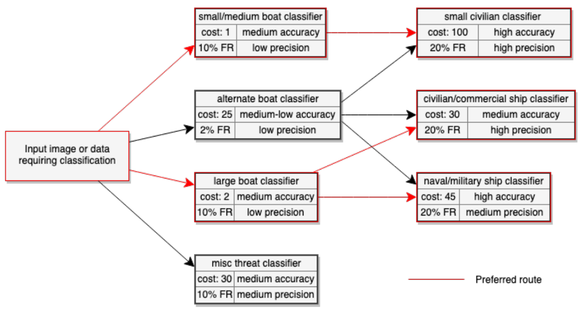
Results



- All data is averaged over 20 runs and training takes place over 10,000 episodes
- Binary action selection results in a faster convergence rate, because it accounts for symmetries in the addition of edges
- Decrease in accumulated reward when scaling graph size and number of node types is to be expected due to the exponentially-increasing size of the action space
- For a DAG of only 6 nodes with only a single node type, there are 9,765 possible terminating states, of which approximately 5 are isomorphic to the 'true' DAG
- For a DAG with 7 nodes, there are 615,195 possible graphs, again with the same amount of true graphs that provide any reward

Summary & Future Work

- Current work uses a highly sparse reward space – while real world scenarios may lack training examples they will likely have detailed reward spaces containing positive and negative examples
- In current work we aim to apply this method to an ensemble of pre-trained machine learning (ML) services that cannot be altered
- With a richer reward space this work will be applicable to coalition scenarios in which workflows are dynamically composed from ensembles of pre-trained ML services provided by multiple partners



Publication(s) & Impact

- Laura D'Arcy, Padraig Corcoran, and Alun Preece. Deep q-learning for directed acyclic graph generation. CoRR, abs/1906.02280, 2019, Accepted to ICML 2019 Learning and Reasoning with Graph-Structured Representations Workshop