Toward Interpretable Deep Reinforcement Learning with Linear Model U-Trees

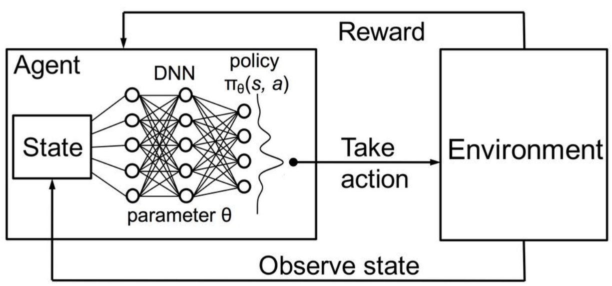
Guiliang Liu, Oliver Schulte, Wang Zhu, Qingcan Li

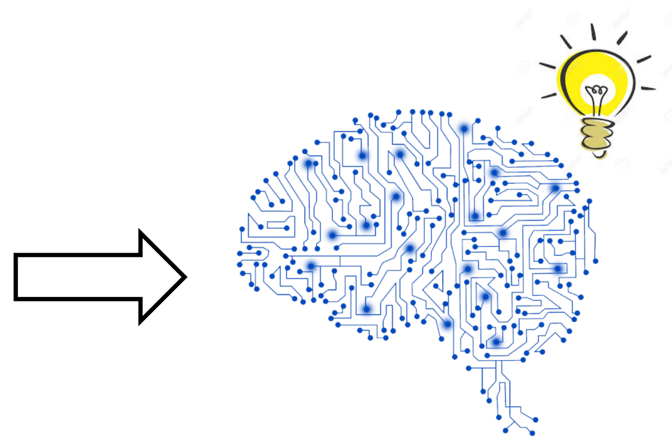
Machine Learning Lab,

SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

ECML-PKDD 2018 Presentation

PROBLEM DEFINITION





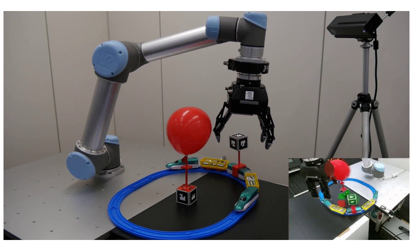
Understand the knowledge learned by Deep Reinforcement Learning (DRL) Model

MOTIVATION

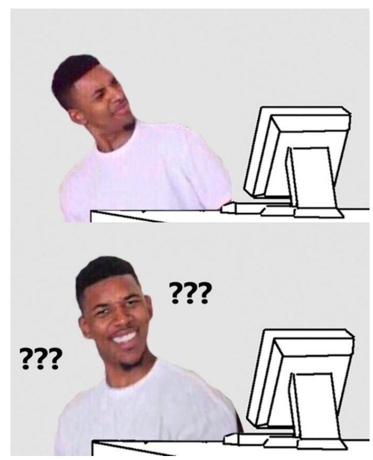
Recent Success of Deep Reinforcement Learning

- Game Environment

- Physical Environment



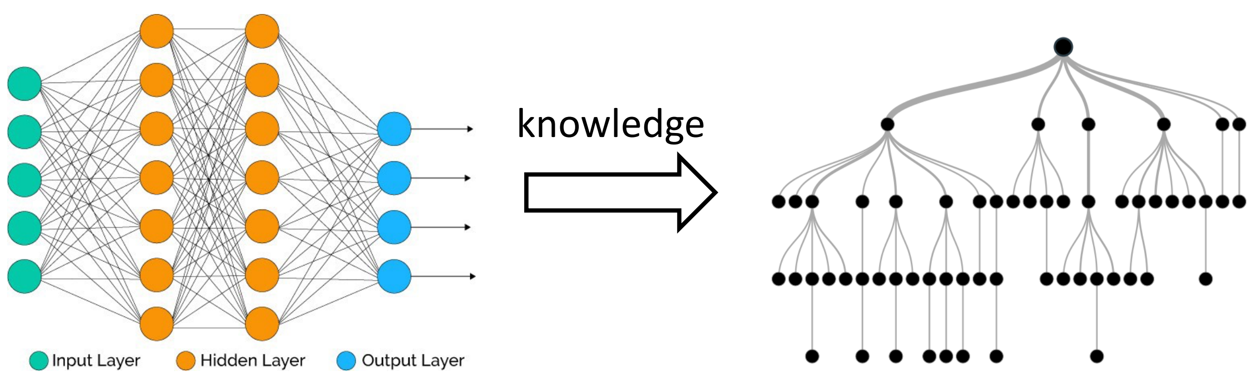
But



MIMIC LEARNING

Interpretable Mimic Learning

- Transfer the knowledge from deep model to transparent structure (e.g. Decision Tree).
- Train the transparent model with the same input and soft output from neural networks.

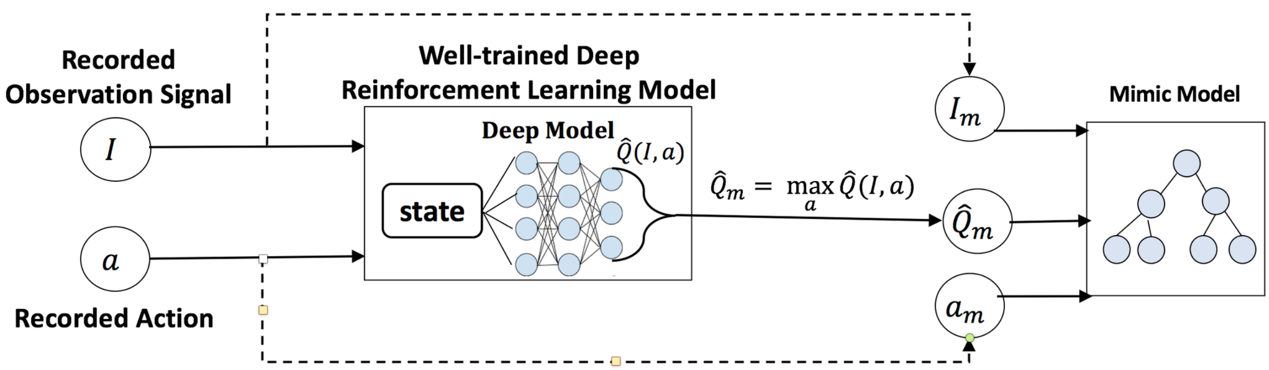


Neural Network   Decision Tree

MIMIC LEARNING FOR DRL

Experience Training Setting

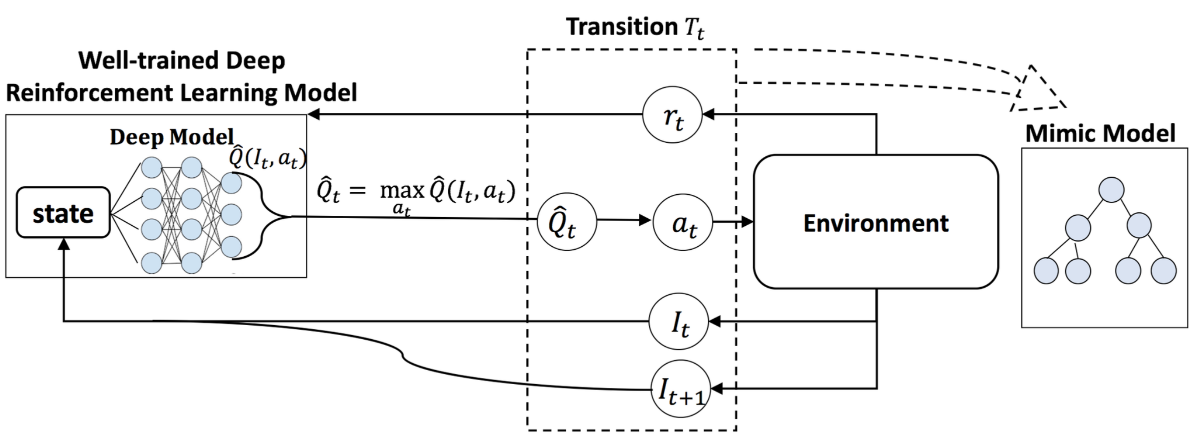
- Recording observation signals I and actions a during DRL training.
- Input them to a mature DRL model, obtain the soft output $Q(I, a)$.
- Generates data for *batch training.*



MIMIC LEARNING FOR DRL

Active Play Setting

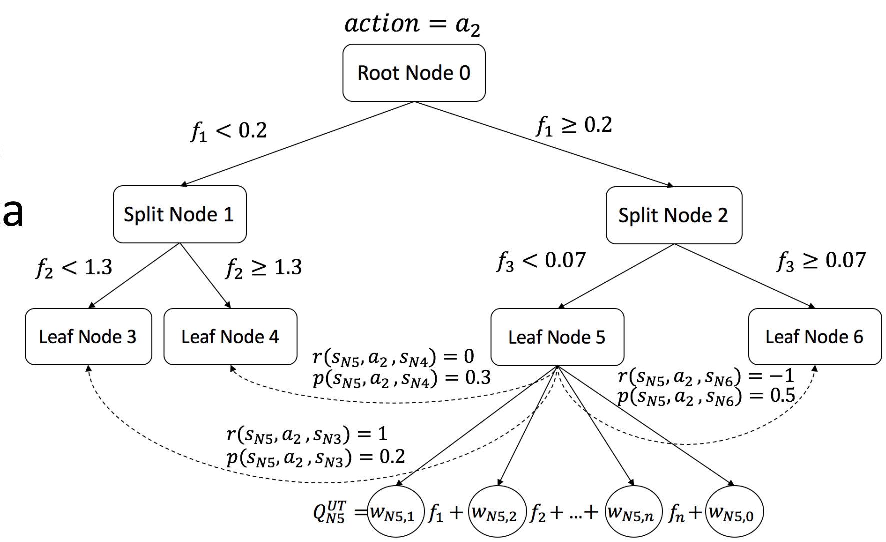
- Applying a mature DRL model to interact with the environment.
- Record a labelled transition $Tt \; =< I_t,\; a_t,\; r_t,\; I_{t+1},\; \hat{Q}(I_t, a_t) >$
- Repeat until we have training data for the *active learner* to finish sufficient updates over mimic model.



MODEL

Linear Model U Tree (LMUT):

- **U tree**: an online reinforcement learning algorithm with a tree structure representation.
- LMUT allows CUT leaf nodes to contain **a linear model**, rather than simple constants.
- LMUT builds a **Markov Decision Process (MDP)** from the interaction data between environment and deep model.



MODEL

Training the Linear Model U Tree (LMUT):

- **Data Gathering Phase:** it collects transitions ($Tt \ = < I_t, \ a_t, \ r_t, \ I_{t+1}, \ \hat{Q}(I_t, a_t) >$) on leaf nodes and prepares for fitting linear models and splitting nodes.
- **Node Splitting Phase:**

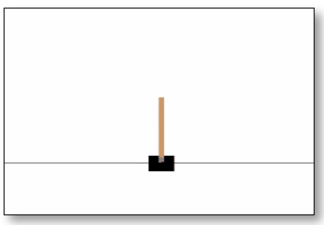
  (1) LMUT scans the leaf nodes and updates their linear model with *Stochastic Gradient Descent (SGD).*

  (2) If SGD achieves sufficient improvement, LMUT determines a *new split* and adds the resulting leaves to the current partition cell.

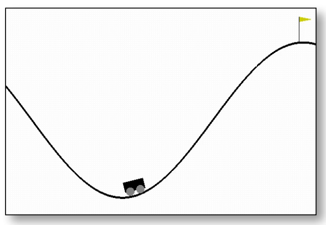
EMPIRICAL EVALUATION

Evaluate the mimic performance of LMUT

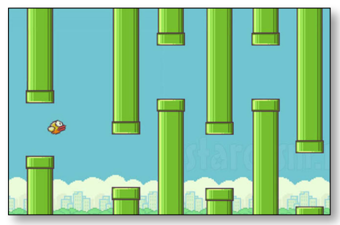
- Evaluation environments:



Mountain Car



Cart pole



Flappy Bird

- Baseline Methods:

(1) For the **Experience Training** environment: Classification And Regression Tree (CART), M5-(Regression/Model)Tree.

(2) For the **Active Play** environment: Fast Incremental Model Trees (FIMT).

EMPIRICAL EVALUATION

Fidelity: Regression Performance

- Evaluate how well our LMUT approximates the soft output from Q function in a Deep Q-Network (DQN).

Table 2: Result of Mountain Car

Method		Evaluation Metrics		
		MAE	RMSE	Leaves
Experience Training	CART	0.284	0.548	1772.4
	M5-RT	0.265	0.366	779.5
	M5-MT	**0.183**	**0.236**	240.3
	FIMT	3.766	5.182	4012.2
	FIMT-AF	2.760	3.978	3916.9
	LMUT	0.467	0.944	620.7
Active Play	FIMT	3.735	5.002	1020.8
	FIMT-AF	2.312	3.704	712.4
	LMUT	0.475	1.015	453.0

Table 3: Result of Cart Pole

Method		Evaluation Metrics		
		MAE	RMSE	Leaves
Experience Training	CART	15.973	34.441	55531.4
	M5-RT	25.744	48.763	614.9
	M5-MT	19.062	37.231	155.1
	FIMT	43.454	65.990	6626.1
	FIMT-AF	31.777	50.645	4537.6
	LMUT	**13.825**	**27.404**	658.2
Active Play	FIMT	32.744	62.862	2195.0
	FIMT-AF	28.981	51.592	1488.9
	LMUT	14.230	43.841	416.2

Table 4: Result of Flappy Bird

Method		Evaluation Metrics		
		MAE	RMSE	Leaves
Experience Training	CART	0.018	0.036	700.3
	M5-RT	0.027	0.041	226.1
	M5-MT	**0.016**	**0.030**	412.6
	LMUT	0.019	0.043	578.5
Active Play	LMUT	0.024	0.050	229.0

(MAE = Mean Absolute Error, RMSE=Root Mean Square Error.)

- LMUT achieves a better fit to the neural net predictions with a much smaller model tree.

EMPIRICAL EVALUATION

Matching Game Playing Performance:

- Evaluate by directly *playing the games with mimic model* computing the Average Reward Per Episode (ARPE).
- LMUT achieves the Game Play Performance APER closest to the DQN.
- The batch learning models have strong fidelity in regression, but they do not perform as well in game playing as the DQN.

Table 5: Game Playing Performance

Model		Game Environment		
		Mountain Car	Cart Pole	Flappy Bird
Deep Model	*DQN*	*-126.43*	*175.52*	*123.42*
Basic Model	CUT	-200.00	20.93	78.51
Experience Training	CART	-157.19	100.52	79.13
	M5-RT	-200.00	65.59	42.14
	M5-MT	-178.72	49.99	78.26
	FIMT	-190.41	42.88	N/A
	FIMT-AF	-197.22	37.25	N/A
	LMUT	-154.57	145.80	97.62
Active Play	FIMT	-189.29	40.54	N/A
	FIMT-AF	-196.86	29.05	N/A
	LMUT	**-149.91**	**147.91**	**103.32**

INTERPRETABILITY

Feature Influence:

- In a LMUT model, feature values are used as splitting thresholds to form partition cells for input signals.

$$Inf_f^N = \left(1 + \frac{|w_{Nf}|^2}{\sum_{j=1}^{J} |w_{Nj}|^2}\right)\left(var_N - \sum_{c=1}^{C} \frac{Num_c}{\sum_{i=1}^{C} Num_i} var_c\right)$$

- We evaluate the influence of a splitting feature by the total variance reduction of the Q values.

Table 6: Feature Influence

	Feature	Influence
Mountain Car	Velocity	376.86
	Position	171.28
Cart Pole	Pole Angle	30541.54
	Cart Velocity	8087.68
	Cart Position	7171.71
	Pole Velocity At Tip	2953.73

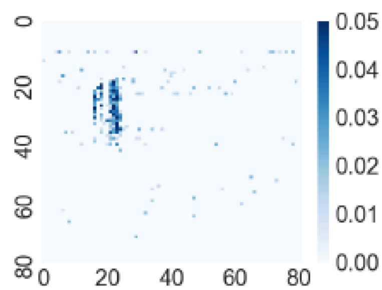
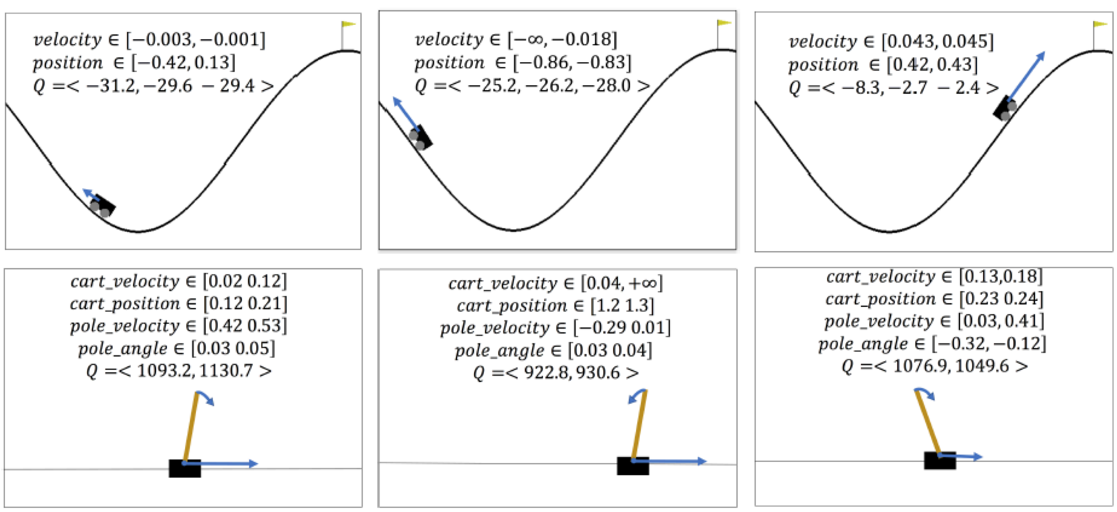


Fig. 6: Super pixels in Flappy Bird

INTERPRETABILITY

Rule Extraction:

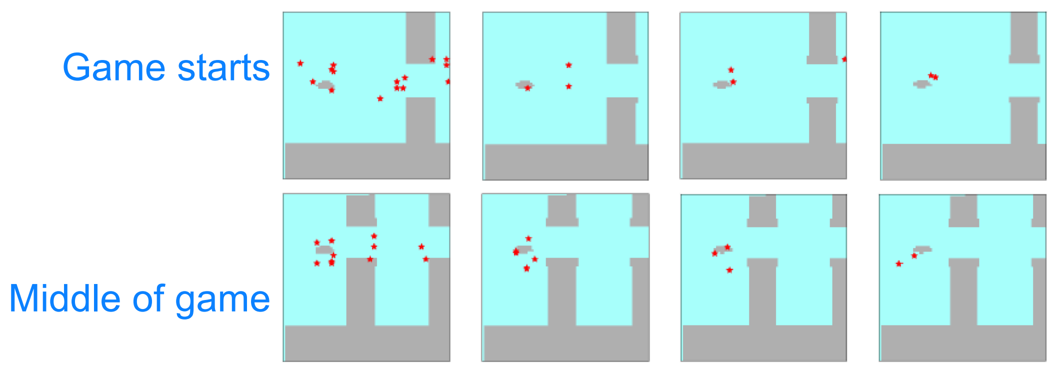
- The rules are presented in the form of partition cells (constructed by the splitting features in LMUT).
- Each cell describes a games situation (similar Q values) to be analyze.



INTERPRETABILITY

Super-pixel Explanation:

- Deep models for image input can be explained by super-pixels.
- We highlight the pixels that have feature influence > 0.008 along the splitting path from root to the target partition cell.



- We find 1) most splits are made on the first image 2) the first image is often used to locate the pipes and the bird, while the remaining images provide further information about the bird's velocity.

THANK YOU!

For more information:

Poster: #xxx

My homepage: http://www.galenliu.com/