POLICY DISTILLATION*

Midterm 4 - Michele Morisco

*Rusu et al, Policy distillation, ICLR 2016. URL: https://arxiv.org/abs/1511.06295

Introduction

- Pixel-to-action policies can deliver superhuman performance on many challenging tasks* using deep Q-networks (DQN).
  - But it needs **large** networks and **extensive** training to achieve good performance in a single task.
- The idea behind policy distillation is to transfer one or more policies from DQNs to an untrained network leading to multiple benefits.
  - But in RL, it is difficult to apply because NN encodes real-valued and unbounded values where scale depends on the expected future **rewards** in the game. And they are **blurred** and **non-discriminative** when multiple actions have similar consequences, on the contrary, they are **sharp** and **discriminative** when actions are important.

*Mnih et al., Human-level control through deep reinforcement learning. Nature, 2015. https://www.nature.com/articles/nature14236

Policy Distillation

It is a method to transfer knowledge from a *teacher* model T to a *student* model S.
The distillation targets are typically obtained by passing the weighted sums of the last network layer through a softmax function.

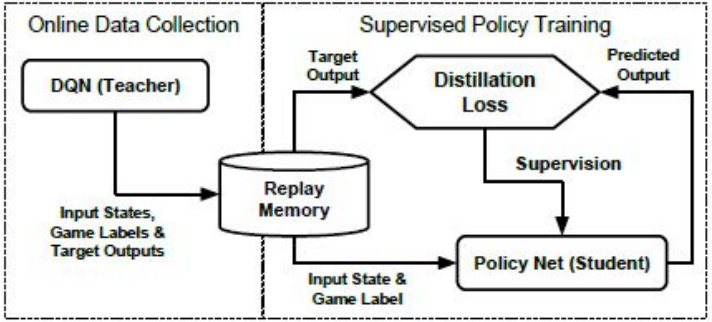
Output teacher: $softmax(\frac{\mathbf{q^T}}{\tau})$ where $\mathbf{q^T}$ is the vector of Q-values of model T and τ is the temperature.

With a higher temperature τ the model T outputs can be softened by passing the network output on softmax. This has the effect to transfer more of the knowledge of the network.

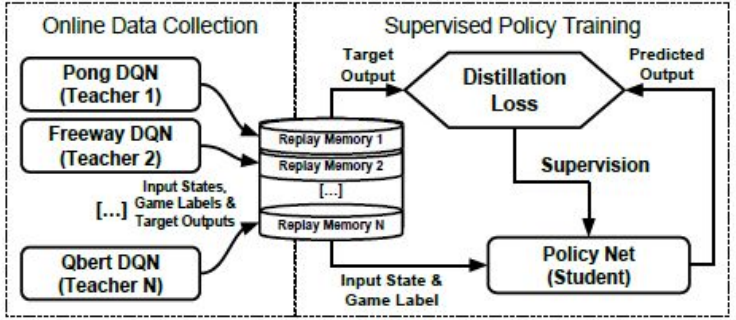
The outputs of the teacher are not a distribution, rather they are the expected future discounted reward of each possible action.

These can be learned by model S using regression.

Policy Distillation - Single-Task vs Multi-Task



- Single task policy distillation is a process of data generation by T and supervised training by S.
- DQN agent periodically adds gameplay to the replay memory while S network is trained. The outputs (Q-values for all actions) alongside the inputs (images) were stored in it.
- The network is optimized to predict the average discounted return of each possible action given a small number of consecutive observations.
- For each game a separate DQN agent is trained, fixed (no Q-learning) and used as a teacher for a single student policy network.



- Use *n* trained DQN single-game experts. The inputs, targets and data is stored in a separate replay memory buffers.
- Distillation agents learned from *n* data stores sequentially, switching to a different agent every episode.
- A separate MLP output (controller) layer is trained for each task.
- The game label is used to switch between different output layers.

Distillation Loss

In the research, comparing with three methods of policy distillation.
All cases, the teacher T generates a dataset $D^T = \{(s_i, q_i)\}_{i=0}^{N}$ where each sample consists of a short observation sequence s_i and a vector q_i of unnormalized Q-values with one value per action.

Negative log likelihood loss (NLL)

$$L_{NLL}(D^T, \theta_S) = -\sum_{i=1}^{|D|} logP(a_i = a_{i,best}|x_i, \theta_S)$$

θ_S are the S model parameters

x_i is an observation at step i.

a_i is an action at step i.

$a_{i,best} = argmax(q_i)$ the highest valued action from teacher.

The student model is trained with NLL to predict the same teacher's action.

Kullback-Leibler divergence (KL)

$$L_{KL}(D^T, \theta_S) = \sum_{i=1}^{|D|} softmax(\frac{q_i^T}{\tau}) ln \frac{softmax(\frac{q_i^T}{\tau})}{softmax(q_i^S)}$$

τ is the temperature.

KL quantifies how much one probability distribution differs from another probability distribution.

It can be used to measure the divergence between discrete and continuous probability distributions. In this case, measures the **divergence** between *softmax*.

Mean-squared-error loss (MSE)

$$L_{MSE}(D^T, \theta_S) = \sum_{i=1}^{|D|} \left\| q_i^T - q_i^S \right\|_2^2$$

q_i^T are vectors of Q-values from teacher.

q_i^S are vectors of Q-values from student.

PRO: it preserves the full set of action-values in the resulting student model.

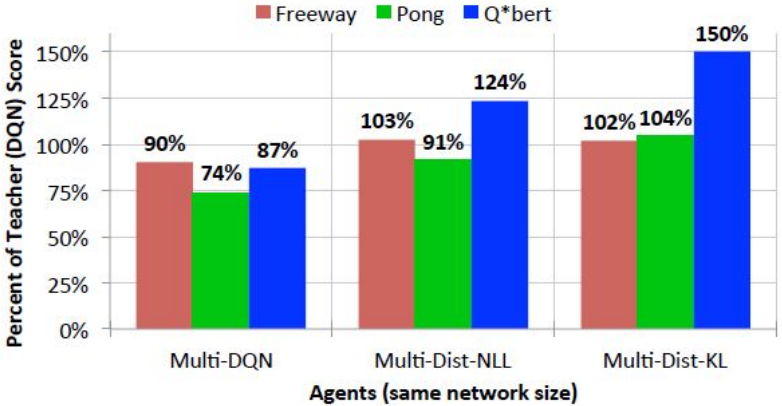
Single-Task Results

	DQN	Dist-MSE		Dist-NLL		Dist-KL	
	score	**score**	**%DQN**	**score**	**%DQN**	**score**	**%DQN**
Breakout	303.9	102.9	33.9	235.9	77.6	287.8	**94.7**
Freeway	25.8	25.7	99.4	26.2	101.4	26.7	**103.5**
Pong	16.2	15.3	94.4	15.4	94.9	16.3	**100.9**
Q*bert	4589.8	5607.3	122.2	6773.5	147.6	7112.8	**155.0**

Comparison of learning criteria used for policy distillation from DQN teachers to students on four Atari games.

- **MSE** shows poor performance because during DQN training mean discounted future rewards are very similar in a large number of states coupled with residual errors of non-linear function approximation.
- **NLL** loss assumes that a single action choice is correct any point in time, but without an optimal teacher, minimizing the NLL could amplify the noise.
- **KL** cost function leads to the best-performing student agents, and distilled agents outperform their DQN teachers on most games. For this domain $\tau = 0.01$ and minimizing KL divergence cost is best suited for distillation.
- Exploring **model compression** through distillation, the distilled agents which are four times smaller than DQN with *428,000* parameters, outperform it.
- Distilled agents with *15* times fewer parameters perform with their DQN teachers.
- These results show that DQN could benefit from a reduced capacity model.

Multi-Task Results



Performance of multi-task agents with identical network architecture and size, relative to respective single-task DQN teachers.

- Training a multi-task DQN agent using the standard DQN algorithm applied to interleaved experience from three games and compare it against distilled agents.
- All 3 agents are using an identical multi-controller architecture.
- DQN agent learns the three tasks to *83,5%* of single-task DQN performance.
- Distilled agents perform better than their DQN teachers with **mean scores**: *105.1%* NLL, *116.9%* KL.
- Testing with 3 of the games achieving much higher scores than the teacher and an overall relative performance of *89.3%*.

Conclusions

In this research:

- The policy learned on single games can be to **compress** in a smaller model without a degradation in the performances.
  - Indeed, the smallest distilled agent (only *62.000* parameters) achieves a mean of *84%*.
- Build agents that are capable of playing multiple games.
  - But, in the experiment, there isn't a comparison with jointly trained *10* games DQN agent, because in the preliminary experiments DQN **failed** to reach high performances on most of the games.
  - However, in general, distillation can **alleviate** some issues such as common input modality, very various images and not sharing a common statistical basis.
- Need to choose the **correct loss function** for distillation in the Reinforcement Learning (RL) setting.
  - In the experiments, the best results are obtained by weighing action classification by a softmax of the action-gap.
  - With **Kullback-Leibler divergence** (KL) achieved good results. The worst was the mean-squared error (MSE) loss function.
- By these results, the distillation can be applied to RL without using an iterative approach and don’t allow the student model to control the data distribution.
  - In addition, it confirms that **distillation** is a general principle for model regularization.