ISPR Midterm 4 - Paper n. 6 - Policy distillation

Geremia Pompei (MAT. 638432)

May 24, 2022

1. Introduction to the problem

Learning policy of complex visual tasks with **Deep Q-Networks** (**DQN**) brings to have:

- large networks
- long training time

Distillation policy is a new technique that extracts policies from one or more teacher agents to create a student agent with a more compact and effective knowledge. This method is able to learn:

- Single-task policy
- Multi-task policy
- In online mode

2. Model description - Single-task policy distillation

Distillation is e method to transfer knowledge from a **teacher** model T to a **student** model S. It's built in a *replay memory* a dataset $D^T = \{(s_i, q_i)\}_{i=0}^{N}$ (dataset produced by teacher model T). S model is fitted using this memory as training set.

- *Classification*: In this kind of task is passed a **weighted sum** from last network layer through a **softmax** function to produce an output of probability distributions. To transfer more knowledge the softmax function is **softened** using temperature parameter τ, so a T network output becomes $softmax(\frac{q^T}{\tau})$ where q^T are *Q-values* of T. These outputs can be learned by S network using regression.
- *Regression*: This case imply learn *Q-function* and is difficult because scale of *Q-values* are **unbounded** and could be **unstable**. Also training S model to predict best action is complex because in the end could have many actions with **similar** *Q-values*.

2. Model description - Multi-task policy distillation

- Here n DNQ expert networks are used to compose the *replay memory* with **n teacher datasets**. The unique S model learns policies switching from a dataset to another every episode.
- Different datasets have different *action sets* so there is a specific layer called **controller layer** that is trained for each specific task. Each of this layers are switched according to their tasks using their id during training and evaluation. These controller layers when are selected become output layers.

3. Key catch of the model, represented by a commented equation

Three methods of policy distillation used to transfer knowledge from T to S:

- **NLL-Loss** (Negative Log-Likehood Loss): Here is used only the highest valued action from teacher $a_{i,best} = max(q_i)$ and S model is trained using NLL loss

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

- **MSE-Loss** (Mean Square Error Loss): In this loss function is preserved the full set of action-values. Can be seen the mean square error between *Q-values* of T and S models

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

3. Key catch of the model, represented by a commented equation

- **KL-Loss** (Kullback-Leibler Loss): Loss that use the temperature τ to make softmax function softer to transfer more teacher knowledge to student. The output distribution of q^T is very peaked so in this way student model is able to learn in more homogeneous way outputs of teacher

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

4. Key (empirical) result

- **Comparison of losses**: Making a comparison among the different presented losses using same architecture with different datasets results shows that in most of cases *KL-Loss* it's able to improve learning capability better than the others. In term of score the network trained using KL-Loss was able to overcome the DQN teacher network in almost all datasets.
- **Compression**: Evaluating single-task distilled agents and DQN teachers using 10 different ATARI games as datasets results tells that S models that are 25%, 7% and 4% of T model in term of size reach performances that are 108%, 102% and 84% with respect teacher network in scores.

5. Comment on novelties, strong points and weaknesses

- Pro
  - Very **compressed** student models can perform very **high scores** that sometimes are also better than ones of teacher models
  - More teacher models can compose together a single compressed student model that is able to predict all these **different tasks**
- Cons
  - To use this technique is **required an expert model** and the knowledge of student is related to the good results of the teacher