Knowledge-free domain-independent automated planning for games Results on the Atari video game

Raphael Lopes Baldi

Supervisor: Prof. Dr. Felipe Rech Meneguzzi

Pontifical Catholic University of Rio Grande do Sul
raphael.baldi@edu.pucrs.br

Porto Alegre, November, 2018

Pontifical Catholic University
of Rio Grande do Sul

Table of Contents

1. Introduction
   - Motivation
   - Theoretical foundation
2. Game Latplan
   - Architecture and dataflow
   - Dataset acquisition
   - Knowledge acqusition
   - Automatic planning
3. Experiments and results
   - Implementation details
   - Experiments
   - Discussion
4. Conclusion

Pontifical Catholic University of Rio Grande do Sul

Introduction

Motivation

Pontifical Catholic University
of Rio Grande do Sul

Motivation

- Building classical planning models using symbolic languages represent a knowledge acquisition bottleneck;
- Describing a video game as a symbolic domain is a hard task for a human to perform;
- Agents for Atari games usually rely on reinforcement learning algorithms or running look-ahead techniques for playing;
- We wanted to evaluate a technique to do automated planning using the knowledge we obtain using deep learn.

Pontifical Catholic University
of Rio Grande do Sul

Introduction

Theoretical foundation

Pontifical Catholic University
of Rio Grande do Sul

Automated Planning

- Introduces symbolic language to generalize search;
- The model is a 4-tuple $\Sigma = (\mathtt{S}, \mathtt{A}, \mathtt{E}, \gamma)$, where:
  - $\mathtt{S} = \langle s_1, s_2, \ldots, s_n \rangle$ is a set of states;
  - $\mathtt{A} = \langle a_1, a_2, \ldots, a_n \rangle$ is a set of actions.
  - $\gamma = \mathtt{S} \times \mathtt{A} \times \mathtt{E} \rightarrow 2^{\mathtt{S}}$ is a state-transition function.
- Given a representation of the domain finds a sequence of actions to go from an initial state to a goal state;

Pontifical Catholic University
of Rio Grande do Sul

Deep Learning

- Machine learning algorithms to extract patterns from data and make predictions about it;
- Uses processing units and connections between them to approximate functions which represent the patterns we are interested in;
- Learn by adjusting parameters and validating the output of the neural network – using a loss function;
- Deep: multiple hidden layers connecting the input to the output.

Latplan

- Proposes to bridge the gap between subsymbolic-symbolic boundary using deep learning to obtain a categorical representation from domain's images;
- Successful planning on puzzles: 8-puzzle, Towers of Hanoi and LightsOut;
- Uses the reparameterization trick to make the latent layer of the encoders converge into a categorical representation;
- Extracts PDDL directly from the categorical representation of domains' images.

Pontifical Catholic University
of Rio Grande do Sul

Table of Contents

1 Introduction
- Motivation
- Theoretical foundation

2 Game Latplan
- Architecture and dataflow
- Dataset acquisition
- Knowledge acqusition
- Automatic planning

3 Experiments and results
- Implementation details
- Experiments
- Discussion

4 Conclusion

Pontifical Catholic University of Rio Grande do Sul

Game Latplan

Architecture and dataflow

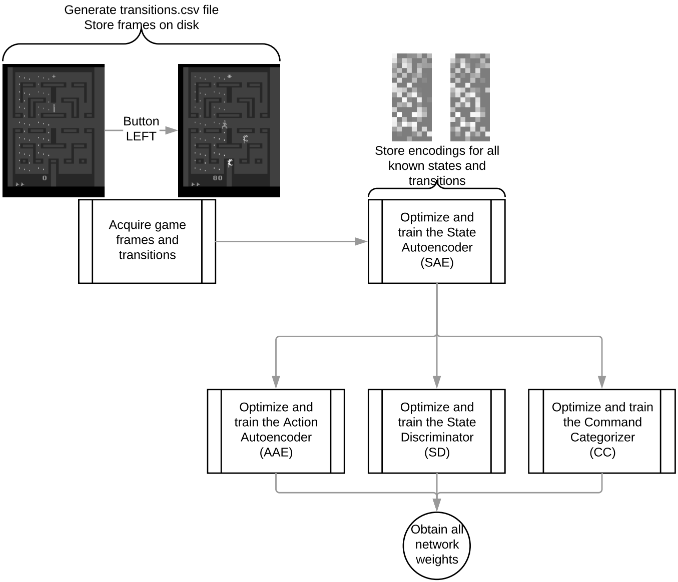
Pontifical Catholic University
of Rio Grande do Sul

Overview

- Our solution adds to the Latplan's architecture:
  - A framework to extract frames and transitions from Atari games;
  - An autoencoder to extract a latent representation from those frames;
  - A neural network to obtain a sequence of commands from a sequence of states' latent representations.
- Three core components:
  - Dataset acquisition
  - Knowledge acquisition
  - Automated planner
- Two phases:
  - Learning
  - Planning

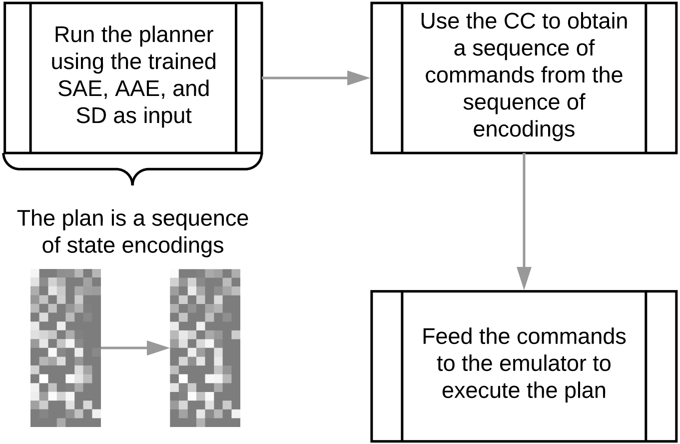
Pontifical Catholic University
of Rio Grande do Sul

Learning phase



ntifical Catholic University of Rio Grande do Sul

Planning phase



Pontifical Catholic University of Rio Grande do Sul

Game Latplan

Dataset acquisition

Pontifical Catholic University
of Rio Grande do Sul

Dataset acquisition

- We use the Arcade Learning Environment;
- Two methods to obtain the dataset:
  - Random agent;
  - Human agent;
- We store frames as grayscale images;
- We store the transitions we observe, including commands and rewards.

Pontifical Catholic University
of Rio Grande do Sul

Game Latplan

Knowledge acqusition

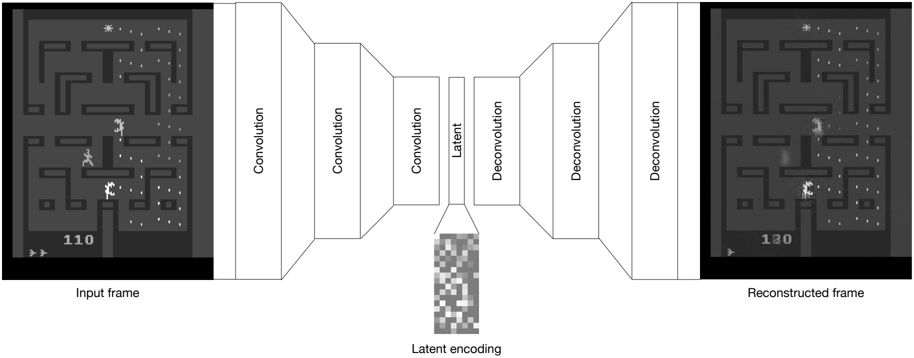
Pontifical Catholic University
of Rio Grande do Sul

Overview

- Four artificial neural networks:
  - State Autoencoder
  - Action Autoencoder
  - State Discriminator
  - Command Categorizer
- We first train the State Autoencoder;
- We use the frames' latent representations to train the other three artificial neural networks.

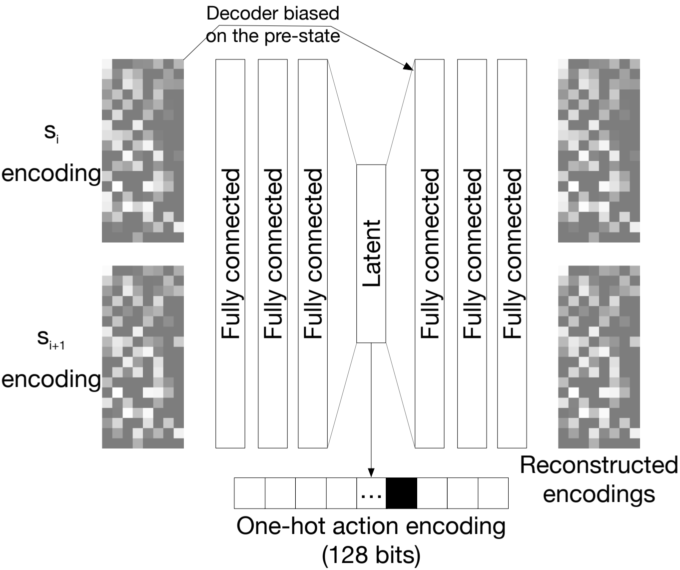
Pontifical Catholic University of Rio Grande do Sul

State Autoencoder



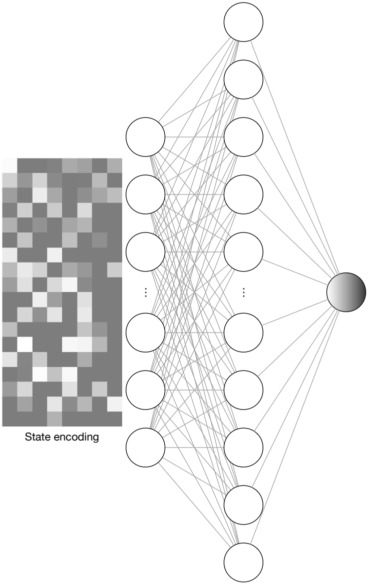
Pontifical Catholic University
of Rio Grande do Sul

Action Autoencoder



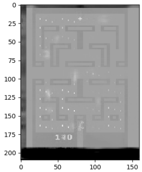
ntifical Catholic University
of Rio Grande do Sul

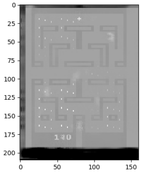
State Discriminator

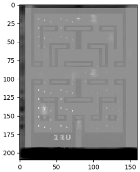


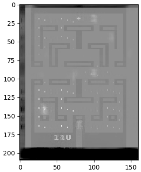
Pontifical Catholic University of Rio Grande do Sul

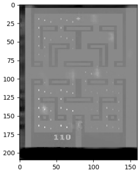
State Discriminator - Fake States





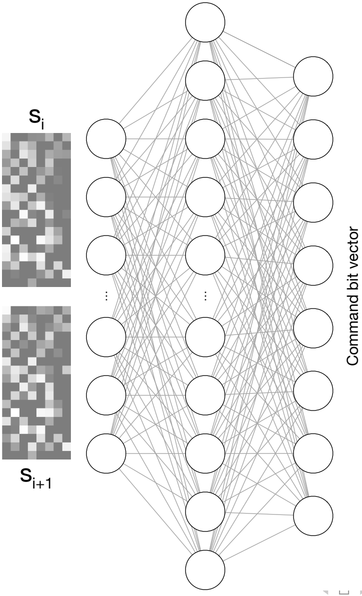






Pontifical Catholic University
of Rio Grande do Sul

Command Categorizer



Game Latplan

Automatic planning

Pontifical Catholic University
of Rio Grande do Sul

Automatic planning

- Use a search algorithm to find a plan to go from the initial state to the goal state;
- Use the Action Autoencoder to expand the current state;
- Use the State Discriminator to prune out invalid states;
- Use the Command Categorizer to convert the plan – a sequence of frames' latent representations – into a sequence of commands.

Pontifical Catholic University
of Rio Grande do Sul

Table of Contents

1 Introduction
- Motivation
- Theoretical foundation

2 Game Latplan
- Architecture and dataflow
- Dataset acquisition
- Knowledge acqusition
- Automatic planning

3 Experiments and results
- Implementation details
- Experiments
- Discussion

4 Conclusion

Pontifical Catholic University
of Rio Grande do Sul

Experiments and results

Implementation details

Pontifical Catholic University
of Rio Grande do Sul

Implementation details

- We implemented the solution using Python, Numpy, and Keras;
- We load all of our datasets on-demand using dataset generators;
- We obtain all the hyperparameters for our artificial neural network using the following steps:
  1. Manually obtain an estimation for hyperparameters' lower and upper bounds;
  2. Input intermediate values between the lower and upper bounds and run a grid search on all combinations;
  3. Manually adjust the best set of hyperparameters to reduce the number of weights in the neural network.

Pontifical Catholic University of Rio Grande do Sul

Hyperparameter influence on training

- Using smaller batch sizes makes the neural network to train faster and overfit less;
- Removing Batch Normalization and using full frames makes the State Autoencoder obtain smaller reconstruction loss;
- Adding Gaussian noise to the input makes the neural network to overfit less at the expense of larger reconstruction losses.

Pontifical Catholic University of Rio Grande do Sul

Hyperparameter influence on training

Batch size	Training epochs	Loss	Validation loss
2000	1200	6.0820e-04	0.0106
1000	600	5.8458e-04	0.0015
500	300	5.7769e-04	0.0014
250	150	5.2938e-04	5.1800e-04
120	70	5.0823e-04	5.8021e-04

Impact of batch size in reconstruction loss.

Pontifical Catholic University of Rio Grande do Sul

Hyperparameter influence on training

Batch Normalization	Frame Cropping	Training loss	Validation loss
Yes	Yes	0.0027	0.1137
Yes	No	5.5038e-04	0.0324
No	Yes	0.0025	0.0027
No	No	4.1907e-04	5.5773e-04

Impact of batch normalization and frame cropping in reconstruction loss.

Pontifical Catholic University of Rio Grande do Sul

Hyperparameter influence on training

Gaussian noise	Training loss	Validation loss
0	2.3350e-04	2.3169e-04
0.2	3.1260e-04	4.3848e-04
0.4	5.0859e-04	5.5853e-04

Impact of adding Gaussian noise to the input in reconstruction loss.

Pontifical Catholic University
of Rio Grande do Sul

Experiments and results

Experiments

Pontifical Catholic University
of Rio Grande do Sul

Categorical State Autoencoder

- We tested three architectures:
  - Fully connected autoencoder;
  - Mixed autoencoder;
  - Fully convolutional autoencoder.
- We were not able to make the autoencoder's latent layer to converge into a categorical representation;
- As a result we could not extract PDDL directly from game's frames.

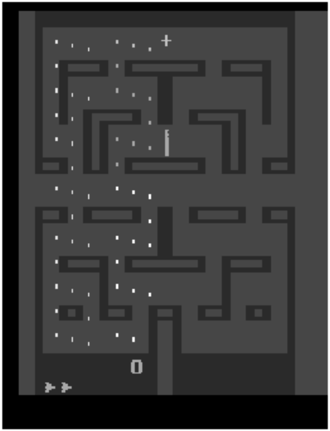
Pontifical Catholic University of Rio Grande do Sul

Planning with Game Latplan

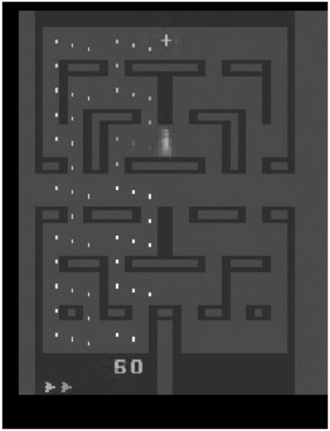
- Run training using the Alien game: 400 thousand frames, 500 thousand transitions;
- Select a pair of states from the transitions we observe during dataset acquisition;
- 97% of the transitions the Action Autoencoder generates are invalid;
- The State Discriminator marks less than 15% of the states as invalid;
- After the expansion of the fifth state, we had over six billion states to analyze.

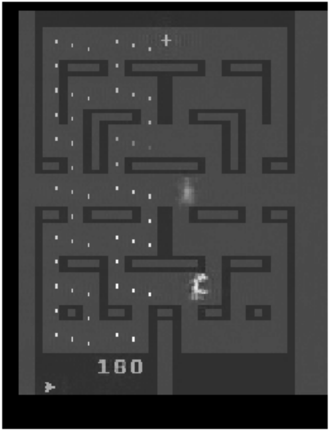
Pontifical Catholic University
of Rio Grande do Sul

Planning with Game Latplan - Initial/Goal State



Pontifical Catholic University of Rio Grande do Sul





Pontifical Catholic University
of Rio Grande do Sul

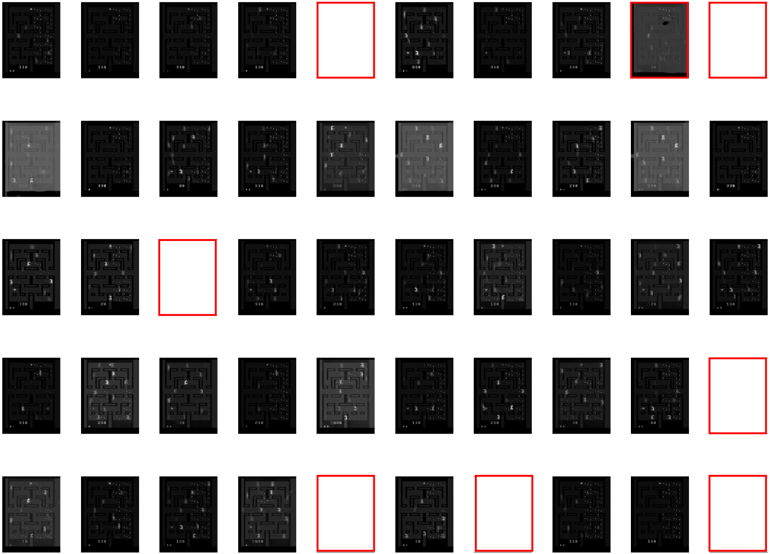
Planning with Game Latplan - Initial/Goal Reconstruction

Metric	Initial state reconstruction loss	Goal state reconstruction loss
Mean absolute error	0.00447	0.00521
Binary cross-entropy	0.44695	0.44788
Mean squared error	2.78381	3.61213

Reconstruction loss for the initial and goal states.

Pontifical Catholic University
of Rio Grande do Sul

Planning with Game Latplan - State expansion



Experiments and results

Discussion

Pontifical Catholic University
of Rio Grande do Sul

Discussion

- The categorical autoencoder could not converge due to the high number and low variability of the game's frames;
- Most of the frame is static data. We reward the neural network for reconstructing and categorizing irrelevant areas;

Pontifical Catholic University of Rio Grande do Sul

Discussion

- Since we are not pruning enough states, the branching factor becomes a bottleneck;
- The reconstruction loss we observe is still too high for the planner to know when it is at the goal state;
- Without an Action Discriminator, the planner is trying to reconstruct the goal from the initial state.
- The Command Categorizer has a low accuracy to allow us to construct useful command sequences.

Pontifical Catholic University
of Rio Grande do Sul

Table of Contents

1 Introduction
- Motivation
- Theoretical foundation

2 Game Latplan
- Architecture and dataflow
- Dataset acquisition
- Knowledge acqusition
- Automatic planning

3 Experiments and results
- Implementation details
- Experiments
- Discussion

4 Conclusion

Pontifical Catholic University
of Rio Grande do Sul

Conclusion

- Latplan could not plan effectively on the domain of Atari games;
- We were able to understand how to organize a deep learning project, and how to conduct scientific research to obtain a neural network architecture for a given problem;
- We reached a better understanding of the difficulties to obtain a symbolic representation for complex – and large – domains.

Pontifical Catholic University of Rio Grande do Sul

Future work

- Combine our neural networks into a single architecture to reduce the reconstruction loss accumulation;
- Experiment with Generative Adversarial Networks to obtain fake states to train the State Discriminator;
- Research methods to obtain an Action Discriminator for Atari games;

Pontifical Catholic University of Rio Grande do Sul

Future work

- Research frameworks – or implement one – to allow for hyperparameter optimization to run multiple sessions in parallel;
- Test our State Autoencoder with other Atari games, and use it as part of a game streaming solution (video compression);
- Keep on learning and researching!

Pontifical Catholic University of Rio Grande do Sul

Thank you!

Pontifical Catholic University
of Rio Grande do Sul