Mutlitask for Learning Disentangled Representations in Maze Environments

Alexander Ivanov

May 2019

Abstract

Machine learning is very dependent on the structured and processed data that it is provided making the unsupervised process of lowering data dimensionality very valuable to save manual labor and improve algorithm performance. The proposed model with the multitask setting is capable of producing a disentangled representation for high dimensional noisy data in grid worlds with complex transition functions. The dimensionality reduction of images to vectors is done without a goal but, the resulting representation is beneficial for planning.

1 Introduction

Machine learning today is very effective at playing complex games such as chess and go through the application of deep nets and thousands of training hours. However, the great success of many state of the art models can be attributed to the preprocessed data on which they are trained. The alphaGO AI doesn't have a video feed of the table, room, and game board but, instead works from an internal representation of the game state that is provided by a person. Often times, without a human made representation or one that is badly formed, a model does not surpass human play if it gets to playing at all. This is why a lot of time and effort is put into making a good representation or labeling data for the AI model to then train on.

Representation learning seeks to bridge this gap where data needs to manually be processed before being fed into a network. The goal is to create a representation of the domain in an unsupervised manner that then can be used as a function to convert raw input into values that the model can then train on. The representation needs to be disentangled, in that the representation is a bijection from states to representations, and it needs to be structured and predictable so that observing transitions in the representations can tell you the actions that occurred.

Existing approaches such as auto encoders, and RBM's often fail to create disentangled representations or fail to represent the underlying state structure. Auto encoders for example can greatly reduce the dimension of the data that they are given but, the representation they produce needs to store enough to reconstruct the initial input. This means that random features are stored by auto encoders when they don't help identify the actual environment state.

To learn structured disentangled representation we use a combination of four fully connected neural networks. The phi network is responsible for creating a representation. The representation learned by the phi network is optimized based on the loss of the forward model, the inverse model, and the distinguishing model. All together the model is able to create a phi network capable of extracting features from noisy observations that are useful for planning.

2 Related Work

VAE [D. P. Kingma and M. Welling] is a variation on the previously described auto encoder method and similarly is used to learn low dimensional representations for high dimensional data. However, it is used to convert single data points into representations where the data has no underlying transitions or environment.

Classical Planning in Deep Latent Space [Masataro Asai and Alex Fukunaga], uses a model to learn a representation for a domain with complex transitions in the context of planning. A representation is learned for the observations of the state space and this representation is used for creating a plan from the start to goal state. This directly uses the representation to produce a plan. Instead of learning the representation and planning jointly, our model first learns a representation that can then be used for planning.

3 Technical Approach

We want to create a model that can create a representation of the states in a gridworld domain from noisy observations. We then want to make sure that this representation is good for planning and that it can be achieved in gridworlds with complex transitions such as mazes.

The testing domain is a gridworld with discrete states and an agent with four actions; up, down, left, right. Walls in the grid world are present around the perimeter and can be between states, preventing transitions. Hidden state refers to the actual grid location that the agent is in. Observations, grayscale with a blurred spot corresponding the the agents location, are images given to the representation learner. The spot is randomly shifted while still corresponding to a unique hidden state and a layer of background noise is added. States or representation states refer to the output of the representation learner. These are low dimensional vectors and are intended to correspond to the hidden states.

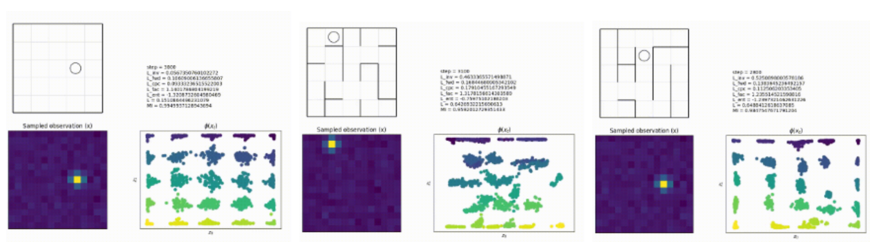
The base algorithm consists of four neural networks. The first is the phi network which learns the representation. Another network, called the forward model, is trained to predict the next representation state from a state and action pair. The inverse model is trained to predict the action that occurred between two representation states. The final network is trained to distinguish between real transitions and fake transitions in the environment. The loss of the other three models is used to adjust the phi network towards a better representation.

The base algorithm is extended to have improved functionality on maze structures. Instead of training in a single environment the agent is placed into a new maze after a relatively small number of network updates. The process of training on multiple environments is called the multitask setting.

Both the base algorithm and the multitask setting are tested on empty gridworlds and mazes. To test whether the learned representation is good the representation is used to run a DQN agent in the environment.

4 Evaluation

The base algorithm has expected performance on empty gridworlds [figure 1.a]. However, for a maze environment the learned representation is clearly less structured [figure 1.b]. The unique hidden states are separated in the representation but there isn't a structured pattern that corresponds to the actions in the environment. The multitask setting, using only 3000 steps as opposed to 5000 used by the baseline, is able to create the structured representation that is desired [figure 1.c]. The structure is similar to the one visible in figure 1.a.



(a)
5x5 empty grid world
Base algorithm, 5000 steps.

(b)
5x5 maze world
Base algorithm, 5000 steps.

(c)
5x5 maze world
multitask, 3000 total steps.

Figure 1: For each visual The top image shows the environment that the agent is in. The agent can get stuck walking into walls and the observations that it gets, the dark grey-scale image, have a layer of background noise that varies for each step in the environment. The image on the right represents how each state in the environment (a color) is mapped to a position in the representation space. Seeing a grid in the plot on the right indicates that the agent was able to learn the grid structure of the environment and has an effective forward model.

To measure the information that is stored in the representation we can compute the mutual information between the representation and the original state distribution. Figure 2 shows the change in mutual information over time for the multitask setting and the base algorithm. Because there is little difference in

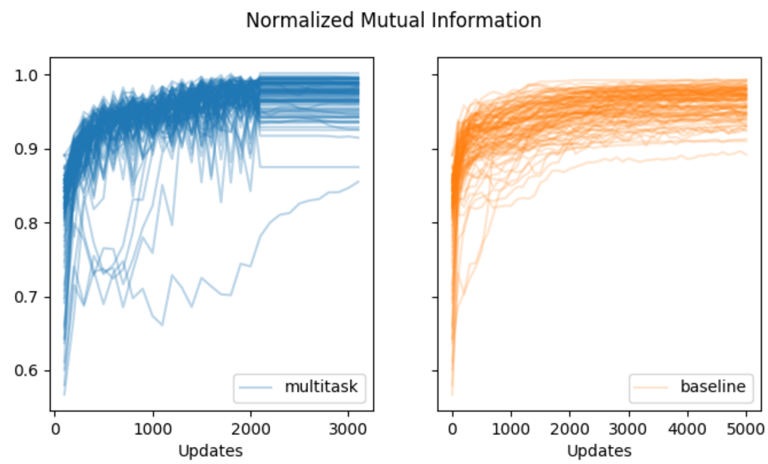


Figure 2

the final values for both methods we know that the structures seen in figure 1.b versus figure 1.c are just as effective in retaining the state space distribution. However, this does not indicate that both methods are equally good at creating representation useful for planning.

To qualitatively evaluate the effectiveness of the representation a DQN is implemented to test the representations usefulness for planning. A fixed goal is added to the environment and the DQN is fed the outputs of the phi network for individual state observations. This means that the Q function learns values in the representation space.

For both the base algorithm and the multitask setting 100 seperate phi networks are produced. For each phi network a separate DQN is trained for 100 trials and with 100 episodes. Figure 3 shows the compiled loss of the DQN agent across the episodes for both the multitask setting and the base algorithm. The higher reward of the agent with the multitask setting shows that the multitask setting representation is more useful for planning. This also shows that having a structured representation [figure 1.c] is important on top of separating states [figure 1.b].

5 Conclusion

The base algorithm is able to construct a disentangled representation from sequential, noisy, and variable state observations greatly reducing the dimensionality of the data. The model is effective at separating distinct states in representation space and contains structure corresponding to the actions in the environment.

The multitask setting is an effective method for constructing representations in grid worlds with complex transitions. Furthermore, the representations

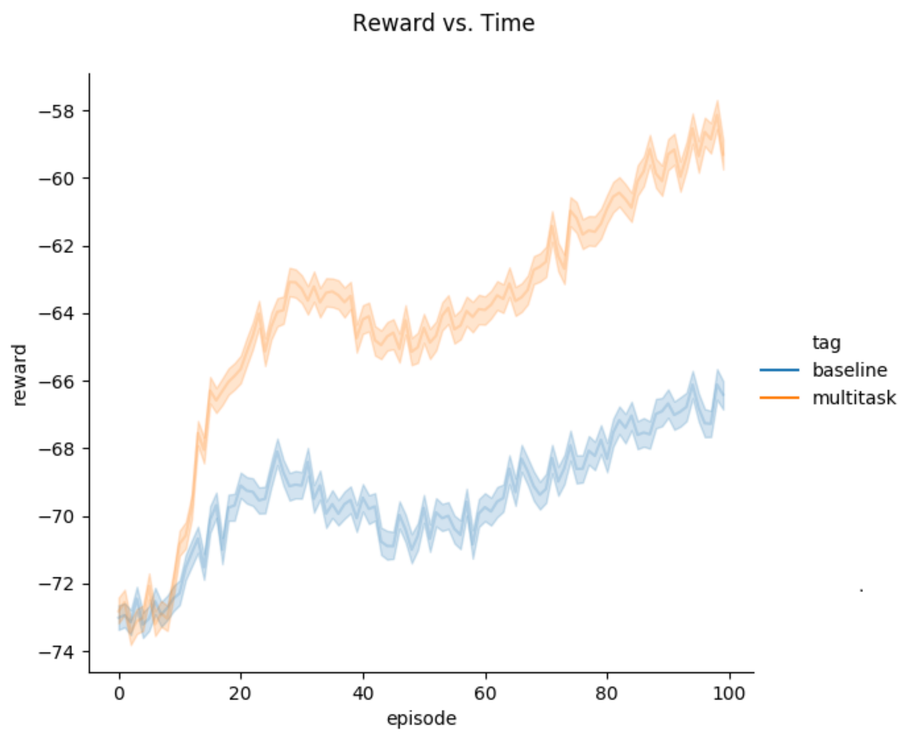


Figure 3

produced by using the multitask setting is effective for planning in the domain.

Further extension of this problem would involve testing the algorithm with larger and more complex gridworlds. It is not certain that the current method will have the same functionality in much larger grid or maze worlds.

The transitions can be made more complex with one-way transitions or "portals" connecting states. Additional objects can also be added to the environment such as fixed objects with no interaction, movable objects, and randomly or predictably moving entities. These test if the model can represent various types of objects in the environment along with the environment itself. The observations that the agent receives can also be modified by showing more or less of the environment or adding various visual effects. Finally the model can be applied to a variety of different domains to test the robustness.

Another extension of the problem is creating a representation for a partially observable environment. In this scenario identical observations may correspond to completely different states meaning that the existing algorithm may not even be applicable as it maps a single observation to the representation space at any given time.

6 References

1. D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In ICLR, 2014. https://arxiv.org/abs/1312.6114

2. Masataro Asai and Alex Fukunaga. Classical Planning in Deep Latent Space. In AAAI, 2018
https://arxiv.org/pdf/1705.00154.pdf