DLAI Project: World Models in MiniGrid

June 3, 2022

Dennis Rotondi 1834864

Abstract

In this report are presented reasoning and investigations to complete the project 06. Code here.

1. Introduction

Sequential decision making, commonly formalized as Markov Decision Process (MDP) optimization, is a key challenge in artificial intelligence. Two successful approaches to solve this problem are planning and reinforcement learning. These may actually be combined, in a field which is known as model-based reinforcement learning; for which a definition is: ”any MDP approach that $i)$ uses a model (known or learned) and $ii)$ uses learning to approximate a global value or policy function” (Moerland et al., 2020). It’s important to notice that the only engagement of a deep learning architecture does not make an algorithm model based, indeed in traditional RL (model-free RL) small networks are used to improve the learned policy through trial and error, but due to the bottleneck of credit assignment problem, i.e. figuring out which steps caused the resulting feedback or which should be blamed for the final result, it’s hard to handle NNs with million of parameters as predictive structure. Instead in MBRL it’s all about the anology that to handle the vast amount of information that flows through our daily lives, our brain learns an abstract representation of both spatial and temporal aspects of this information (Quian et al., 2005), thus large structures are not only possible but also encouraged.

2. Related Work

Model Based RL is not an invention of the new millennium, many research papers yet in ’90s were discussing and analyzing the different RL branches (Atkeson & Santamaria, 1997). However it’s with the recent explosion of deep learning that new progresses have been achieved, a line of the new era can be delimited by the nice World Model work (Ha & Schmidhuber, 2018): we’ll focus on this generation of algorithms. To solve the Car Racing and VizDoom scenario, here the authors built three NNs: one to learn the environment, one to memorize and predict the next observation and one to control the agent. On this skeleton google research realized and published different models to master the Atari games: SimPLe (Kaiser et al., 2019), where a policy is deployed to collect more data in the original game, achieving SOTA in sample-efficiency at that time; DreamerV2 (Hafner et al., 2021) builds upon the Recurrent State-Space Model also used in PlaNet (Hafner et al., 2018) and DreamerV1 (Hafner et al., 2019), each visual observation into a 32 distributions over 32 classes, the meanings of which are determined automatically as the world model learns: it achieves human-level performance!

3. Method

I’ve decided to reproduce the structure of the reference paper suggested applied to the the Minigrid gym (Chevalier-Boisvert et al., 2018), a very simple grid-based env which has the pro of having many variants, I’ve focused on the obstacle grid to build the training set. This toy problem has been chosen maily for practical reasons: due to limited hardware I cannot train using 200M samples like Dreamer who uses the google servers, then my main objective is to investigate how this solution works while doing practice with the course tools at my best. For the sake of completeness I’ve also tried more complex gyms, but what prevents to reach high performance is, like experienced by the google team, the object vanishing problem: crucial aspects of the world (enemies to kill, the agent etc.) are not captured by the vision structure, but what’s this vision system?

3.1. Variational Autoencoder

The vision model is a Variational Autoencoder which embed the 3D 64x64x3 image observation into a 1D lower dimension latent vector z_t. It does not only learn how to reconstruct the input but also to compress the training distribution into $N_{0,I}$ using the KLD loss term. The best training results are thanks to Adam optimizer and a starting LR: 1e-3, a WD: 1e-5 and a Beta: 0.01 factor to balance KLD term and the MSE reconstruction loss. Moreover, to reduce

Email: Dennis Rotondi <rotondi.1834864@studenti.uniroma1.it>.

Deep Learning and Applied AI 2022, Sapienza University of Rome, 2nd semester a.y. 2021/2022.

the generalization error Earlystopping and ReduceLROn-Plateau techniques have been employed.

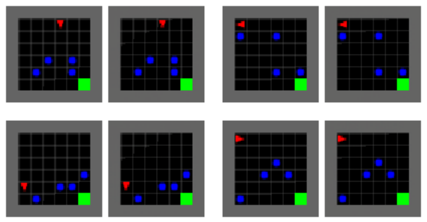

Figure 1. Result of the VAE training: on the right recon. images.

3.2. MDN-RNN

The second network has the role of memory and to predict the future (next state and termination condition). It consist in an LSTM (Staudemeyer & Morris, 2019) which output is processed by a Mixture Density Network (Bishop, 1994). LSTM is a particular RNN developed to overcome the vanishing gradient problem, the general idea is is that it takes as input a sequence of vectors $x_1, ..., x_n$ and an initial state vector h_0 and it returns a list of hidden state vectors $h_1, ..., h_n$ and a list of output vectors $\hat{y}_1, ..., \hat{y}_n$ where each $\hat{y}_i$ is a function of the corresponding state vector h_i: a kind of memory of the previous i inputs; a constant error flow within special cells and access to the cells is handled by multiplicative gate units. To not overfit the next state prediction into a sigle z_{t+1} MDN will predict a class of probability distributions called Gaussian Mixture Models, where the output value is modelled as a weighted sum of multiple Gaussians, each with different means and stds. The general training setup is similar to vae's one, the difference is in the training loss: part is a MSE on for done state and part is the log-likelihood of the gmm eq 1: K is the number of gaussians, Π_k, ϕ_k, μ_k, σ_k the weight, distribution, mean and std for gaussian k.

$$loss(y|x) = -log[\sum_{k}^{K} \Pi_k(x)\phi(y, \mu_k(x), \sigma_k(x))] \quad (1)$$

The big drawback of this loss is that is not lower bounded since the mixture components may collapse in a data point making the loss decrease to arbitrarily small values.

3.3. Controller

Eventually who's responsible to choose the action, given the latent embedding of the observation and the hidden prediction, is a simple MLP to leave all the model complexity to the vision and memory. Learning controller's parameters for this task is a difficult non-linear non-convex black-box optimization problem, where the CMA-ES (Hansen, 2016) is know to work well. It is an algorithm that can take the results of each generation, and adaptively increase or decrease the search space for the next generation around a variance parameter sigma: 100. The learning procedure is based on an ask-tell interface, where the solver initially guess some possible set of parameters, then rollouts are performed and rewards for each set given back to the solver.

4. Results

I've decided to test the performances also in simpler variants of the MiniGrid: empty (no obstacles), random (empty + random start). Results are summarized in table 1, average of 1000 rollouts reward is between [-1, 0.96]. Training only the controller in the empty world will solve easily the task receiving every time the highest possible reward. Almost the same apply for the random empty grid, here we always win but not always with the shortest path, my guess is that since the lstm is initialized always with the same hidden states, it needs some steps to figure out where it is and predict the right state. In the obstacle grid, the optimal policy, that wins in the 20% of the cases is the same of empty grid: the upper rightmost square is reached by the agent, then it turn right and go straight to the green gol square.

Table 1. Trained on (row) vs Tested on (col)

	empty	random	obstacles
empty	0.96	0.22	-0.93
random	0.91	0.88	-0.98
obstacles	0.96	0.16	-0.66

5. Discussion and conclusions

World models facilitate generalization and can predict the outcomes of potential actions to produce faster good policies. Learning with statistical uncertainty in the virtual environment make the final policy more robust as shown in (Ha & Schmidhuber, 2018) for the VizDoom game, there the score attained in the real environment was much higher than the one obtained inside the dream. But being in a dream has not only positive sides: inevitably we introduce sources of approximation error that could be converted into quibble that allows the agent to cheat: for example some position where enemies cannot hit him; this implies to take into account a problem not present in the original environment. Possible evolutions of this paradigm are the unification of the structure, so that running an action at the end of the pipeline take a role in the learning process of the agglomerate, one could also try to replace VAE and MDN-RNN with higher capacity models like Transformers (Vaswani et al., 2017). Extending the latent space as done by (Hafner et al., 2021) is the key to reduce the object vanishing problem, furthermore google ai sees world models that leverage large offline datasets, long-term memory, hierarchical planning, and directed exploration.

References

Atkeson, C. G. and Santamaria, J. C. A comparison of direct and model-based reinforcement learning. In *IN INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, pp. 3557–3564. IEEE Press, 1997.

Bishop, C. Mixture density networks. Workingpaper, Aston University, 1994.

Chevalier-Boisvert, M., Willems, L., and Pal, S. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

Ha, D. and Schmidhuber, J. World models. 2018. doi: 10.5281/ZENODO.1207631. URL https://zenodo.org/record/1207631.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels, 2018. URL https://arxiv.org/abs/1811.04551.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination, 2019. URL https://arxiv.org/abs/1912.01603.

Hafner, D., Lillicrap, T., Norouzi, M., and Ba, J. Mastering atari with discrete world models, 2021. URL https://arxiv.org/abs/2010.02193.

Hansen, N. The cma evolution strategy: A tutorial, 2016. URL https://arxiv.org/abs/1604.00772.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., and Michalewski, H. Model-based reinforcement learning for atari, 2019. URL https://arxiv.org/abs/1903.00374.

Moerland, T. M., Broekens, J., Plaat, A., and Jonker, C. M. Model-based reinforcement learning: A survey, 2020. URL https://arxiv.org/abs/2006.16712.

Quian, R., Reddy, L., Kreiman, G., Koch, C., and Fried, I. Invariant visual representation by single neurons in the human brain. *Nature*, 435:1102–7, 07 2005. doi: 10.1038/nature03687.

Staudemeyer, R. C. and Morris, E. R. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 2019. URL https://arxiv.org/abs/1909.09586.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.