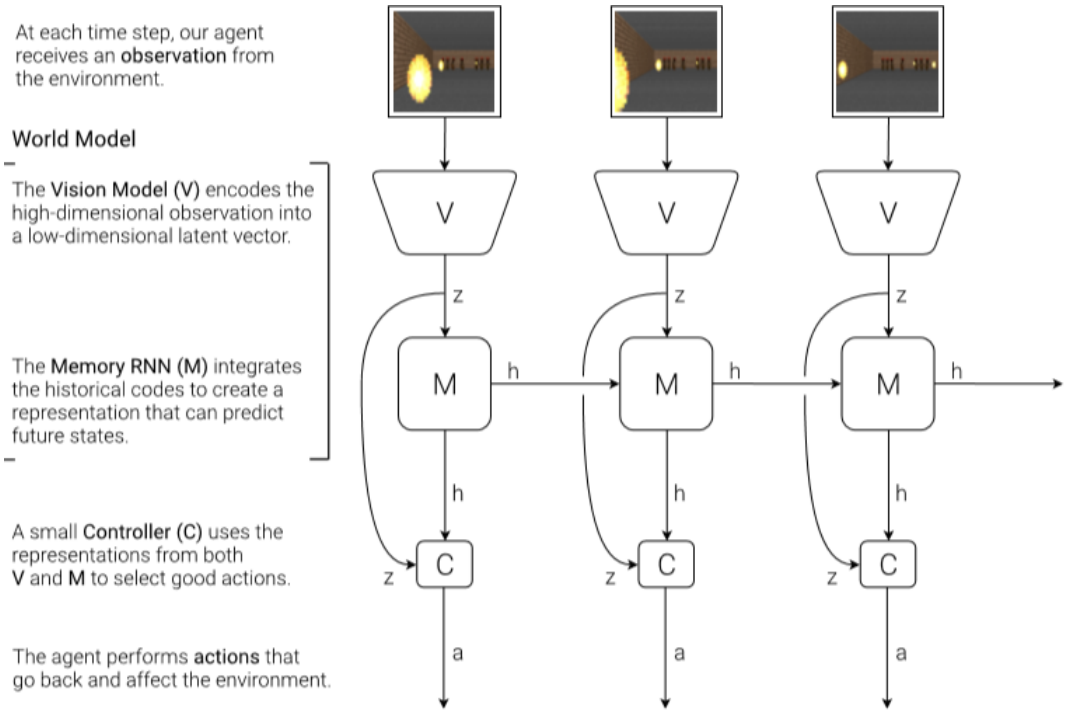
Review of World Models by David Ha and Jurgen Schmidhuberr

Ganeshan Malhotra

August 7, 2019

Abstract

Humans develop a model of the world based on what they are able to perceive by their senses. Our brain models the spatial and temporal information of the environment around us. Evidence also suggests that what we perceive at any given moment is governed by our brain's prediction of the future based on our internal model. In this work the authors aim to train a RL based agent by training the neural net to learn the world model and then using a controller to perform specific tasks in the world model.



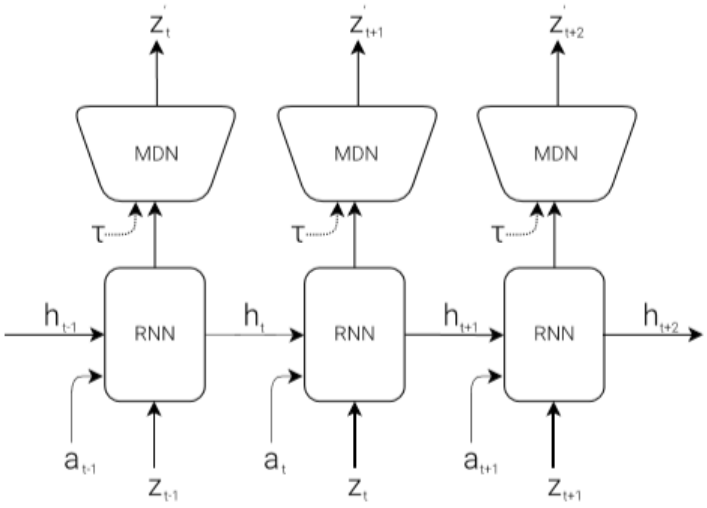
1 Model Architecture

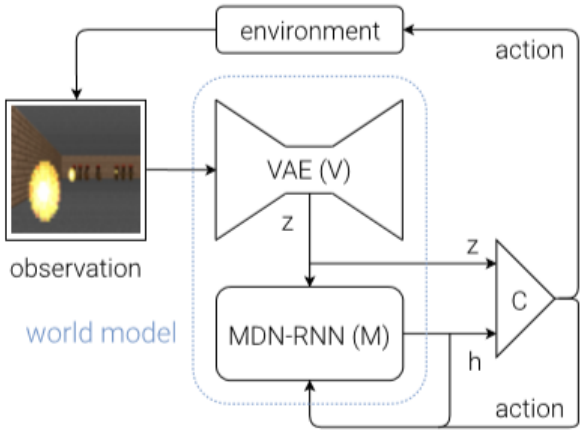
The authors create the world model by dividing it into 3 main the parts— VISION Model, MEMORY Model and the CONTROLLER Model. The vision model takes in high dimensional observation of the world as its input and converts it into a low dimensional latent vector. The memory model makes predictions about the

future using the temporal data. The Controller model selects the actions based on maximization of its reward function.

For the Vision model the authors use a Variational Autoencoder which is trained so as to maximize the similarities in the observed model and the actual one.

The memory model is basically a RNN with Mixture Density Layer as the output layer. Fundamentally the memory model attempts to predict the probability distribution of the next latent vector and not the deterministic prediction of what the next vector will be. The Controller decides the course of action of the agent. It is a single layer of neural network so as to keep the number of parameters small. To optimize the parameters of the Controller a technique called Covariance Matrix-Adaption Evolution Strategy is used as the optimization strategy.





2 Car Racing Experiment

The agent model was trained to solve a car racing experiment. In this environment, the tracks are randomly generated for each trial, and our agent is rewarded for visiting as many tiles as possible in the least amount of time. For training the model uses 10,000 random rollouts of the environment. The V model is trained to encode each frame into low dimensional latent vector z by minimizing the difference

between a given frame and the reconstructed version of the frame produced by the decoder from z. The M model is trained separately using the preprocessed data to model the probability distributions of the next frame. And for the controller the CMA-ES is used for optimization keeping in mind that only the controller has access to the actual reward function of the environment. For experiments the agent was first handicapped by giving it access to only V and not M. And then in the next set of experiments the full world model was deployed consisting of V, M and C. Clearly the agent performed reasonably well using the full world model.

3 VizDoom Experiment

Our agent does not directly observe the reality, but only sees what the world model lets it see. In this experiment, the agent is trained inside the hallucination generated by its world model. The agent must learn to avoid reballs shot by monsters from the other side of the room with the sole intent of killing the agent. Here the setup is mostly same but some key differences are that the M model model not only gives the probability distribution of the next frame but also predicts whether the agent dies in the next frame.