Solving Stochasticity using MuZero and Conditional RealNVP

Modelling stochasticity using Normalizing Flows

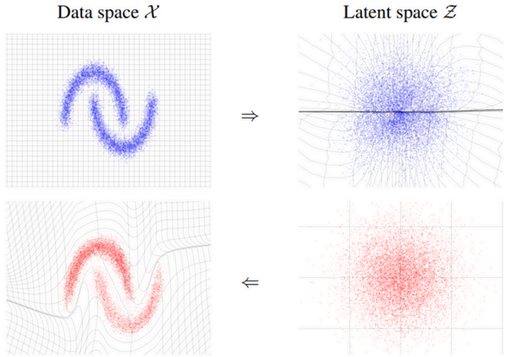
Author: Bogdan Radu Damian

Responsible Professor: Frans A. Oliehoek
Supervisor: Jinke He | Examiner: Michael Weinmann

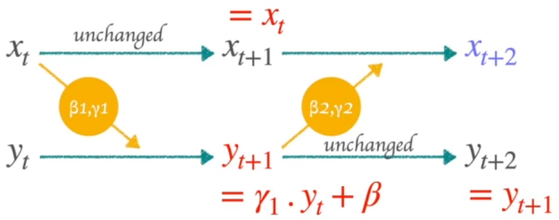
TU Delft

1.RealNVP

Using a satisfyingly complex set of invertible transformations can effectively map a *complex distribution* onto a very *simple* one

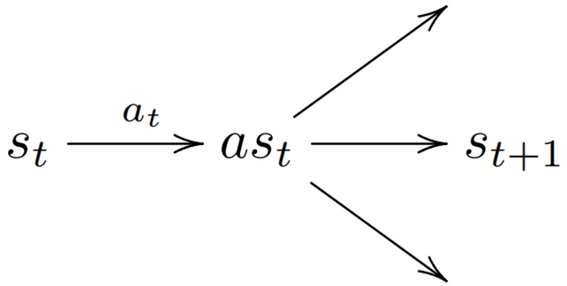


Ensure invertibility by keeping each transformation simple, using arbitrarily complex methods for their parameters



2.Modified MCTS

We *split* a step into applying a deterministic action which leads to an afterstate followed by the modeled stochasticity of the environment



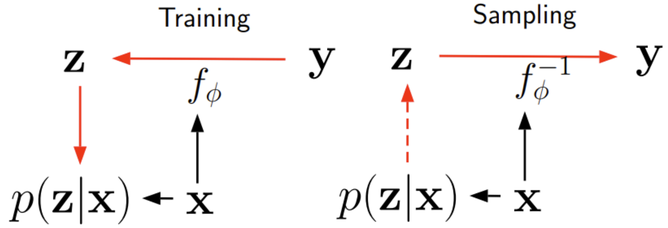
3.Conditional Flows

Learn the future state **x** as a probability distribution *conditional* on the previous afterstate **y**

$$\mathbf{x} \sim p_{X|Y=\mathbf{y}}(\mathbf{x})$$

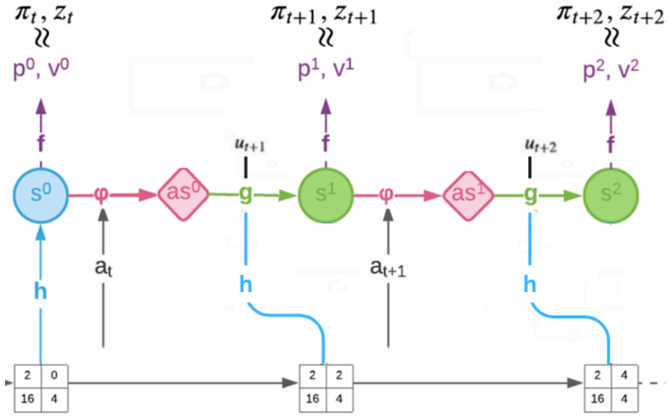
4.Conditional Transformations

Instead of passing **just x_t** to the transformations that give us β and γ, we first *concatenate* it with the previous afterstate, *conditioning the transformations* to predict the **future state y** *with relation to* this **afterstate x**



5.Proposed Architecture

Train the RealNVP flow *end-to-end* in MuZero fashion alongside the representation function, the prediction function, and the afterstate dynamics function



6.Results and Conclusions

- We could not make FlowZero work yet
- The CNF suffers from posterior collapse, cheating the log-likelihood loss by **always mapping to the mean of the prior**
- This leads to a **hard cap** on the accuracy of value, policy, and reward predictions, stopping it from learning *in the online FlowZero setting*
- *In spite of this,* good performance in offline predictions was achieved

7.Limitations and Future Work

- The model was only tested with the single-player game 2048
- We only managed to get success in an offline setting, work should be done towards fixing the posterior collapse