QMIX Analysis

Yanjie Ze, May 2021

1 QMIX's disadvantages

1. Monotonic $\Rightarrow$ Suboptimal
2. Lack of `committed exploration` (a general disadvantage for the MARL algorithm like QMIX and QTRAN)

   $\Rightarrow$ In our setting, Only QBot uses ϵ-greedy.

2 Improve?

1. Delete the **absolute** constraint in **QMIX**. Will this work? (possibly hard to converge)
2. How to add the committed exploration in multi agent?

MAVEN

MAVEN: Multi-Agent Variational Exploration (NeurIPS2019)

1 Motivation

To overcome the detrimental effects of QMIX's monotonicity constraint on exploration.

(Actually, the original model name of MAVEN is **Noise Q** ,shown in their codes. So I think the original idea of them is to simply add some noise to increase the exploration.)

2 Architecture

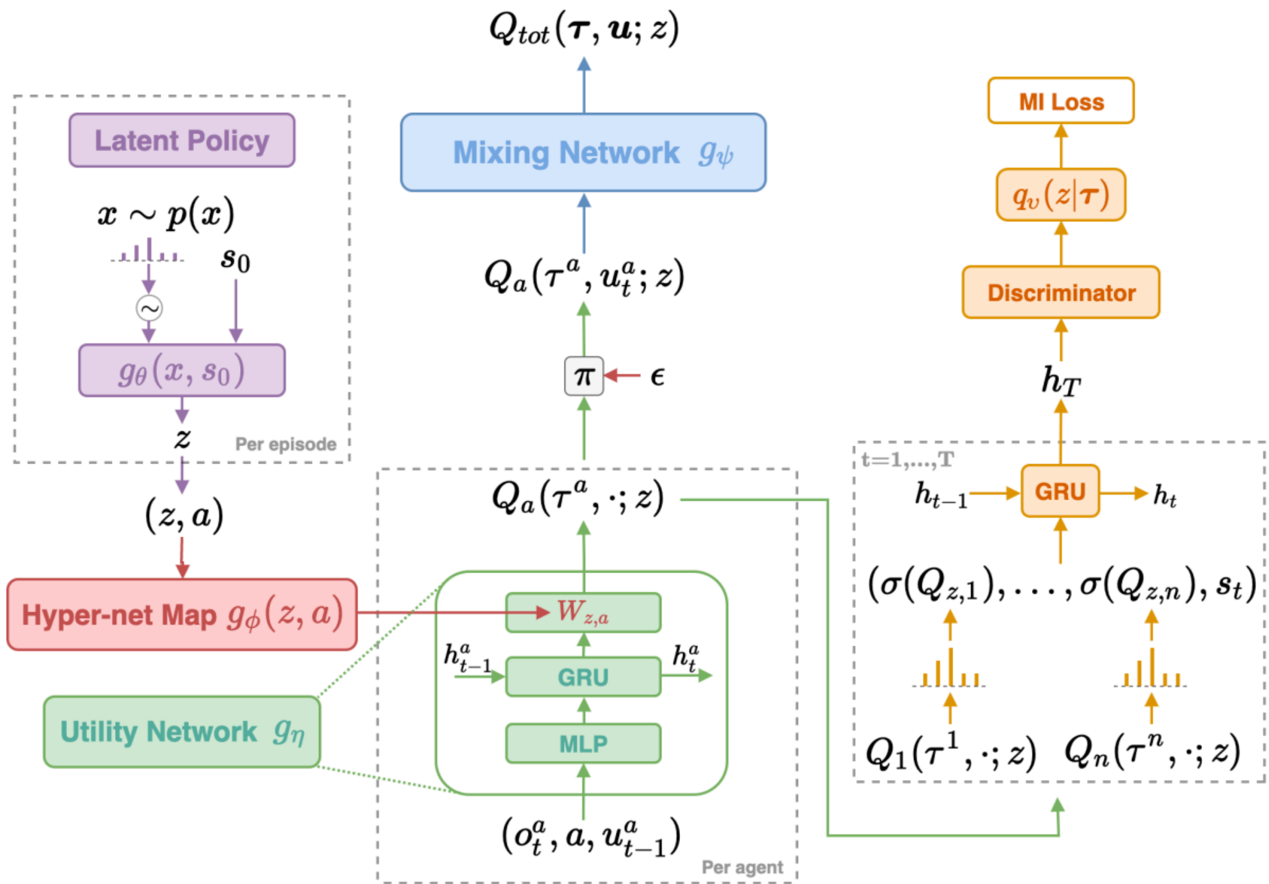

Figure 2: Architecture for MAVEN.

3 Latent Policy

x is a simple random variable, $x \sim p(x)$

Natural choices for *p*(*x*) are uniform for discrete *z* and uniform or normal for continuous *z*.

s_0 is the initial state.

z is the latent variable, $z \sim g_\theta(x, s_0)$

4 Optimise the parameter

Fix z, get the Q-learning loss:

$$\mathcal{L}_{QL}(\phi|\eta, \psi) = \mathbb{E}_{\pi_\mathcal{A}}[(Q(\mathbf{u}_t, s_t; z) - [r(\mathbf{u}_t, s_t) + \gamma \max_{\mathbf{u}_{t+1}} Q(\mathbf{u}_{t+1}, s_{t+1}; z)])^2],$$

The hierarchical policy objective for z, freezing the parameters ψ, η, ϕ is given by:

$$\mathcal{J}_{RL}(\theta) = \int \mathcal{R}(\tau_\mathcal{A}|z) p_\theta(z|s_0) \rho(s_0) dz ds_0.$$

However, the formulation so far does not encourage diverse behaviour corresponding to different values of z and all the values of z could collapse to the same joint behaviour. To prevent this, we introduce a *mutual information* (MI) objective between the observed trajectories $\tau = (u_t, s_t)$,

Use an **RNN** to encode the entire trajectory.

Intuitively:

---

the MI objective encourages visitation of diverse trajectories $\boldsymbol{\tau}$ while at the same time making them identifiable given z, thus elegantly **separating the z space into different exploration modes**.

$$\mathcal{J}_{MI} = \mathcal{H}(\sigma(\boldsymbol{\tau})) - \mathcal{H}(\sigma(\boldsymbol{\tau})|z) = \mathcal{H}(z) - \mathcal{H}(z|\sigma(\boldsymbol{\tau})),$$

Where H is the entropy. The model is shown in the right side of the architecture.

Overall:

$$\max_{\upsilon,\phi,\eta,\psi,\theta} \mathcal{J}_{RL}(\theta) + \lambda_{MI}\mathcal{J}_V(\upsilon,\phi,\eta,\psi) - \lambda_{QL}\mathcal{L}_{QL}(\phi,\eta,\psi),$$

5 Training

Algorithm 1 MAVEN

Initialize parameter vectors $\upsilon, \phi, \eta, \psi, \theta$
Learning rate $\leftarrow \alpha$, $\mathcal{D} \leftarrow \{\}$
for each episodic iteration **do**
  $s_0 \sim \rho(s_0), x \sim p(x), z \sim g_\theta(x; s_0)$
  for each environment step t **do**
    $\mathbf{u}_t \sim \pi_{\mathcal{A}}(u|s_t;; z, \phi, \eta, \psi)$
    $s_{t+1} \sim p(s_{t+1}|s_t, \mathbf{u}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, \mathbf{u}_t, r(s_t, \mathbf{u}_t), r^z_{aux}(\mathbf{u}_t, s_t), s_{t+1})\}$
  end for
  for each gradient step **do**
    $\phi \leftarrow \phi + \alpha\hat{\nabla}_\phi(\lambda_{MI}\mathcal{J}_V - \lambda_{QL}\mathcal{L}_{QL})$ (Hypernet update)
    $\eta \leftarrow \eta + \alpha\hat{\nabla}_\eta(\lambda_{MI}\mathcal{J}_V - \lambda_{QL}\mathcal{L}_{QL})$ (Feature update)
    $\psi \leftarrow \psi + \alpha\hat{\nabla}_\psi(\lambda_{MI}\mathcal{J}_V - \lambda_{QL}\mathcal{L}_{QL})$ (Mixer update)
    $\upsilon \leftarrow \upsilon + \alpha\hat{\nabla}_\upsilon\lambda_{MI}\mathcal{J}_V$ (Variational update)
    $\theta \leftarrow \theta + \alpha\hat{\nabla}_\theta\mathcal{J}_{RL}$ (Latent space update)
  end for
end for

6 Test

At test time, we sample *z* at the start of an episode and then perform a decentralised argmax on the corresponding *Q*-function to select actions.