DeepAcTS Agent Structure

October 4, 2024

1 Notational Conventions

I use capital letters like P and Q to denote an arbitrary probability distribution, as opposed to a *particular* probability distribution like the Gaussian: $\mathcal{N}$ or Categorical distribution: $\mathcal{C}$.

Lowercase letters that prefix a pair of parenthesis — like $p(\ldots)$ — are always used to denote a *neural network*. This neural network either predicts the sufficient statistics for a multivariate Gaussian, or it predicts a categorical distribution. I hope it is sufficiently clear from the context as to which meaning is employed in particular instances.

I use Greek letter sub/superscripts to label neural networks, their predicted values, the associated distribution that their predictions parameterize and any value sampled from this distribution. Hence p_α denotes a *particular* neural network and $(\mu_\alpha, \Sigma_\alpha)$ denote the network's predicted values. Finally, $\mathcal{N}_\alpha$ denotes the *actual* approximate distribution, parameterized by $(\mu_\alpha, \Sigma_\alpha)$.

If x is a variable, then I use $\hat{x}^{(\alpha)}$ to denote that this variable has been *sampled* from the probability distribution: P_α. A subscript τ indexes a particular value in some time series. Hence, $\hat{x}^{(\alpha)}_\tau$ is the value sampled from P_α at time τ. I use $\hat{x}^{(*)}$ to denote that x is sampled from *differing* sources, depending on the training/testing context. See Sections 4 and 5 for details on where these values are sampled.

I use $\mathbb{A}$ to denote the *action space*. This is the set of all action affordances for the agent at all times. Following [2], I use $\tilde{G}_\tau$ to denote the *Expected Free Energy* of each action in the action space, at time τ. Hence the expected free energy for action a_τ is denoted $\tilde{g}(a_\tau)$.

Finally, o_τ denotes an *observation* at time τ. s_τ denotes the *hidden state* at time τ and a_τ denotes the *action* at time τ.

2 Models and VFE Loss

I use the form of the Variational Free Energy from [2] as the free energy loss function. This is:

$$\begin{aligned} -F_\tau = &-\mathbb{E}_{Q(s_\tau)}[\log\left(P(o_\tau \mid s_\tau)\right)] \\ &+ \mathrm{D}_{\mathrm{KL}}[Q(s_\tau) \parallel P(s_\tau \mid s_{\tau-1}, a_{\tau-1})] \\ &+ \mathrm{D}_{\mathrm{KL}}[Q(a_\tau \mid s_\tau) \parallel P(a_\tau \mid s_\tau)] \end{aligned} \tag{1}$$

$P(s_\tau \mid s_{\tau-1}, a_{\tau-1})$ is the prior state-transition probability. $Q(s_\tau)$ is the *Variational* posterior over hidden states. $P(o_\tau \mid s_\tau)$ is the observation likelihood. The Variational action posterior is $Q(a_\tau \mid s_\tau)$ and the *true* action posterior is $P(a_\tau \mid s_\tau)$.

3 Neural Network Approximations

Following a standard assumption in the literature, I aim to approximate $P(s_\tau \mid s_{\tau-1}, a_{\tau-1})$, $Q(s_\tau)$ and $P(o_\tau \mid s_\tau)$ as diagonal, multivariate Gaussian distributions. Hence, my neural network approximations to these densities output a mean vector: μ and a vector that constitutes the *diagonal* entries of the covariance matrix: Σ. For the sake of numerical stability, my Gaussian models actually output *log-variances*: $\log(\Sigma)$.

Hence any one of my Gaussian models may be notated as:

$$(\mu_\alpha(v), \log(\Sigma_\alpha(v))) = r_\alpha(v \mid \Omega) \tag{2}$$

Here α denotes the parameters of the Gaussian neural network: r_α. The variable v is the variable over which the multivariate Gaussian predicted by r_α is defined. v could be a state, observation or action (for example). Ω is an arbitrary set of variables upon which v is conditioned. Hence, in equation 4 we have that:

$$\begin{aligned} \alpha &= \theta \\ r_\alpha &= q_\theta \\ v &= s_\tau \\ \Omega &= \{\hat{s}^{(\theta)}_{\tau-1}, \hat{a}^{(*)}_{\tau-1}, \hat{o}^{(*)}_\tau\} \end{aligned} \tag{3}$$

Thus are my approximations:

$$(\mu_\theta(s_\tau), \log(\Sigma_\theta(s_\tau))) = q_\theta(s_\tau \mid \hat{s}^{(\theta)}_{\tau-1}, \hat{a}^{(*)}_{\tau-1}, \hat{o}^{(*)}_\tau) \tag{4}$$

$$(\mu_\phi(s_\tau), \log(\Sigma_\phi(s_\tau))) = p_\phi(s_\tau \mid \hat{s}^{(\theta)}_{\tau-1}, \hat{a}^{(*)}_{\tau-1}) \tag{5}$$

$$(\mu_\nu(o_\tau), \log(\Sigma_\nu(o_\tau))) = p_\nu(o_\tau | \hat{s}^{(\theta)}_\tau) \tag{6}$$

$$\{\text{Prob}(a_\tau)\ \forall a_\tau \in \mathbb{A} \mid \mu_\theta(s_\tau), \Sigma_\theta(s_\tau)\} = q_\xi(a_\tau \mid \mu_\theta(s_\tau), \Sigma_\theta(s_\tau)) \tag{7}$$

$$\tilde{G}_\tau = \{\tilde{g}(a_\tau)\ \forall a_\tau \in \mathbb{A} \mid \mu_\theta(s_\tau), \Sigma_\theta(s_\tau)\} = f_\psi(\mu_\theta(s_\tau), \Sigma_\theta(s_\tau)) \tag{8}$$

$$\{\text{Prob}(a_\tau)\ \forall a_\tau \in \mathbb{A} \mid \tilde{g}(a_\tau) \in \tilde{G}_\tau\} = \sigma(-\gamma_t \cdot \tilde{G}_\tau) = p_\lambda(a_\tau \mid \tilde{G}_\tau) \tag{9}$$

Where the following values are sampled from 4 by means of the "Reparameterization Trick":

$$\hat{s}^{(\theta)}_{\tau-1} = \mu_\theta(s_{\tau-1}) + \epsilon \odot \Sigma_\theta(s_{\tau-1}) \tag{10}$$

$$\hat{s}^{(\theta)}_\tau = \mu_\theta(s_\tau) + \epsilon \odot \Sigma_\theta(s_\tau) \tag{11}$$

Note: f_ψ is a neural network that predicts the Expected Free Energy of future actions, given the multivariate Gaussian statistics over the current state. See [2] for details on this network. We can now obtain an approximation to the densities in 1 by instantiating a diagonal, multivariate Gaussian distribution for each respective prediction in the above. That is:

$$\begin{aligned} Q(s_\tau) &\approx \mathcal{N}_\theta(\mu_\theta(s_\tau), \Sigma_\theta(s_\tau)) \\ P(s_\tau \mid s_{\tau-1}, a_{\tau-1}) &\approx \mathcal{N}_\phi(\mu_\phi(s_\tau), \Sigma_\phi(s_\tau)) \\ P(o_\tau \mid s_\tau) &\approx \mathcal{N}_\nu(\mu_\nu(o_\tau), \Sigma_\nu(o_\tau)) \end{aligned} \tag{12}$$

Similarly for the Categorical distributions:

$$\begin{aligned} Q(a_\tau \mid s_\tau) &\approx q_\xi(a_\tau \mid \mu_\theta(s_\tau), \Sigma_\theta(s_\tau)) = \mathcal{C}_\xi(a_\tau) \\ P(a_\tau \mid s_\tau) &\approx p_\lambda(a_\tau \mid \tilde{G}_\tau) = \mathcal{C}_\lambda(a_\tau) \end{aligned} \tag{13}$$

Thus, equation 1 becomes:

$$\begin{aligned}-F_\tau = &-\mathbb{E}_{\mathcal{N}_\theta}[\log(\mathcal{N}_\nu)] \\ &+ \mathrm{D}_{\mathrm{KL}}[\mathcal{N}_\theta \parallel \mathcal{N}_\phi] \\ &+ \mathrm{D}_{\mathrm{KL}}[\mathcal{C}_\xi(a_\tau) \parallel \mathcal{C}_\lambda(a_\tau)]\end{aligned} \tag{14}$$

COMMENT:
I think I actually want to train on this free-energy loss (Catal et al):

$$\begin{aligned}\mathcal{L} = &\sum_\tau D_{\mathrm{KL}}\left[q_\theta(s_\tau \mid \hat{s}^{(\theta)}_{\tau-1}, \hat{a}^{(\mathcal{D})}_{\tau-1}, \hat{o}^{(\mathcal{D})}_\tau) \parallel p_\phi(s_\tau \mid \hat{s}^{(\theta)}_{\tau-1}, \hat{a}^{(\mathcal{D})}_{\tau-1})\right] \\ &- \log\left(p_\nu(o_\tau | \hat{s}^{(\theta)}_\tau)\right)\end{aligned} \tag{15}$$

4 Training Phase

To train these networks, I follow the procedure laid out in [1], described here.

I first obtain a dataset: $\mathcal{D}$ of N observation-action pairs solicited by means of a random policy:

$$\mathcal{D} = \{(o_\tau, a_\tau)\} \mid \tau \in [0, N] \subset \mathbb{Z}\} \tag{16}$$

Hence the "starred" terms (with superscript *) in 4 and 5 become the following:

$$\begin{aligned}\hat{a}^{(*)}_{\tau-1} &= \hat{a}^{(\mathcal{D})}_{\tau-1} \\ \hat{o}^{(*)}_\tau &= \hat{o}^{(\mathcal{D})}_\tau\end{aligned} \tag{17}$$

to indicate that they are sampled from the training dataset $\mathcal{D}$. The networks are then trained via SGD on mini-batches from $\mathcal{D}$ with respect to equation 14. Figure 3 in [1] depicts this process very nicely.

5 Testing Phase

Once trained, these networks can then be used to perform Active Inference and planning. In the Testing phase, actions are now sampled from the action posterior: $q_\xi(a_\tau \mid \mu_\theta(s_\tau), \Sigma_\theta(s_\tau))$, instead of randomly:

$$\begin{aligned}\hat{a}^{(*)}_{\tau-1} &= \hat{a}^{(\xi)}_{\tau-1} \\ \hat{o}^{(*)}_\tau &= \hat{o}^{(\mathcal{D})}_\tau\end{aligned} \tag{18}$$

As an aside: for any possible counter-factual *planning* procedures such as a tree search over future states/observations, the learned observation model: $p_\nu(o_\tau | \hat{s}^{(\theta)}_\tau)$ can be used to generate plausible observations from hidden state estimates. Thus, I would use:

$$\hat{o}^{(*)}_\tau = \hat{o}^{(\nu)}_\tau \tag{19}$$

in the course of a "planning as inference" procedure. I do not consider any counter-factual *planning* procedures here — although that is my ultimate goal — and so I leave this for another time.

6 References

References

[1] Ozan Çatal et al. "Learning Generative State Space Models for Active Inference". In: *Frontiers in Computational Neuroscience* 14 (2020). ISSN: 1662-5188. DOI: 10.3389/fncom.2020.574372. URL: https://www.frontiersin.org/articles/10.3389/fncom.2020.574372.

[2] Otto van der Himst and Pablo Lanillos. "Deep Active Inference for Partially Observable MDPs". In: *Active Inference*. Ed. by Tim Verbelen et al. Cham: Springer International Publishing, 2020, pp. 61–71. ISBN: 978-3-030-64919-7.