View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Ghent University Academic Bibliography

LEARNING PERCEPTION AND PLANNING WITH DEEP ACTIVE INFERENCE

Ozan Çatal *Tim Verbelen* *Johannes Nauta* *Cedric De Boom* *Bart Dhoedt*

IDLab
Department of Information Technology at Ghent University - imec

ABSTRACT

Active inference is a process theory of the brain that states that all living organisms infer actions in order to minimize their (expected) free energy. However, current experiments are limited to predefined, often discrete, state spaces. In this paper we use recent advances in deep learning to learn the state space and approximate the necessary probability distributions to engage in active inference.

Index Terms— active inference, deep learning, perception, planning

1. INTRODUCTION

Active inference postulates that action selection in biological systems, in particular the human brain, is actually an inference problem where agents are attracted to a preferred prior state distribution in a hidden state space [1]. To do so, each living organism builds an internal generative model of the world, by minimizing the so-called free energy. The idea of active inference stems from neuroscience [2, 3] and has already been adopted to solve different control and learning tasks [4, 5, 1]. These experiments however typically make use of manually engineered state transition models and predefined, often discrete, state spaces.

In this paper we show that we can also learn the state space and state transition model, by using deep neural networks as probability density estimators. By sampling from the learnt state transition model, we can plan ahead minimizing the expected free energy, trading off goal-directed behavior and uncertainty-resolving behavior.

2. ACTIVE INFERENCE

Active inference states that every organism or agent entertains an internal model of the world, and implicitly tries to minimize the difference between what it believes about the world and what it perceives, hence minimizing its own variational free energy [2]. Moreover, the agent believes that it will minimize its expected free energy in the future, in effect turning action selection into an inference problem. This boils down to optimizing for two distinct objectives. On the one hand the agent actively samples the world to update its internal model of the world and better explain observations. On the other hand the agent is driven to visit preferred states that it believes a priori it will visit—a kind of global prior— which carry little expected free energy.

Formally, we assume an agent entertains a generative model $P(\tilde{\boldsymbol{o}}, \tilde{\boldsymbol{s}}, \tilde{\boldsymbol{a}}, \pi)$ of the environment, which specifies the joint probability of observations, actions and their hidden causes, where actions are determined by some policy π. If the environment is modelled as a Markov Decision Process (MDP) this generative model factorizes as:

$$P(\tilde{\boldsymbol{o}}, \tilde{\boldsymbol{s}}, \tilde{\boldsymbol{a}}, \pi) = P(\pi)P(\boldsymbol{s}_0) \prod_{t=1}^{T} P(\boldsymbol{o}_t|\boldsymbol{s}_t)P(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1})P(\boldsymbol{a}_{t-1}|\pi) \tag{1}$$

The free energy is then defined as:

$$\begin{aligned} F &= \mathbb{E}_Q[\log Q(\tilde{\boldsymbol{s}}) - \log P(\tilde{\boldsymbol{o}}, \tilde{\boldsymbol{s}}, \tilde{\boldsymbol{a}})] \\ &= D_{\mathrm{KL}}(Q(\tilde{\boldsymbol{s}}) \| P(\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{a}}|\tilde{\boldsymbol{o}})) - \log P(\tilde{\boldsymbol{o}}) \\ &= D_{\mathrm{KL}}(Q(\tilde{\boldsymbol{s}}) \| P(\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{a}})) - \mathbb{E}_Q[\log P(\tilde{\boldsymbol{o}}|\tilde{\boldsymbol{s}})] \end{aligned} \tag{2}$$

where $Q(\tilde{\boldsymbol{s}})$ is an approximate posterior distribution. The second equality shows that the free energy is minimized when the KL divergence term becomes zero, meaning that the approximate posterior becomes the true posterior, in which case the free energy becomes the negative log evidence. This can also be rewritten as the third equality, which is the negative evidence lower bound (ELBO) that also appears in the variational autoencoders (VAE) framework [6, 7].

In active inference, agents infer the actions that will result in visiting states of low expected free energy. They do this by sampling actions from a prior belief about policies according to how much expected free energy that policy will induce. Formally, this means that the probability of picking a policy is given by [8]:

$$\begin{aligned} P(\pi) &= \sigma(-\gamma G(\pi)) \\ G(\pi) &= \sum_{\tau} G(\pi, \tau) \end{aligned} \tag{3}$$

where σ is the softmax function with precision parameter γ, which governs the agents goal-directedness and randomness

Ozan Çatal is funded by a Ph.D. grant of the Flanders Research Foundation (FWO). This work is supported by AI Research Flanders

in its behavior. Here G is the expected free energy at future timestep τ when following policy π [8]:

$$\begin{aligned} G(\pi,\tau) &= \mathbb{E}_{Q(\boldsymbol{o}_\tau,\boldsymbol{s}_\tau|\pi)}[\log Q(\boldsymbol{s}_\tau|\pi) - \log P(\boldsymbol{o}_\tau,\boldsymbol{s}_\tau|\pi)] \\ &= \mathbb{E}_{Q(\boldsymbol{o}_\tau,\boldsymbol{s}_\tau|\pi)}[\log Q(\boldsymbol{s}_\tau|\pi) - \log P(\boldsymbol{o}_\tau|\boldsymbol{s}_\tau,\pi) \\ &\quad - \log P(\boldsymbol{s}_\tau|\pi)] \\ &= D_{\mathrm{KL}}(Q(\boldsymbol{s}_\tau|\pi)\|P(\boldsymbol{s}_\tau)) + \mathbb{E}_{Q(\boldsymbol{s}_\tau)}[H(\boldsymbol{o}_\tau|\boldsymbol{s}_\tau)] \end{aligned} \tag{4}$$

We used $Q(\boldsymbol{o}_\tau,\boldsymbol{s}_\tau|\pi) = P(\boldsymbol{o}_\tau|\boldsymbol{s}_\tau)Q(\boldsymbol{s}_\tau|\pi)$ and that the prior probability $P(\boldsymbol{s}_\tau|\pi)$ is given by a preferred state distribution $P(\boldsymbol{s}_\tau)$. This results into two terms: a KL divergence term between the predicted states and the prior preferred states, and an entropy term reflecting the expected ambiguity under predicted states. Action selection in active inference thus entails:

1. Evaluate $G(\pi)$ for each policy π
2. Calculate the belief $P(\pi)$ over policies
3. Infer the next action using $P(\pi)P(\boldsymbol{a}_{t+1}|\pi)$

3. DEEP ACTIVE INFERENCE

For the agent's model, we use deep neural networks to parameterize the various factors of equation (1): i.e. the transition model $p_\theta(\boldsymbol{s}_t|\boldsymbol{s}_{t-1},\boldsymbol{a}_t)$ and the likelihood distribution $p_\xi(\boldsymbol{o}_t|\boldsymbol{s}_t)$. Also the approximate posterior is parameterized by a neural network $p_\phi(\boldsymbol{s}_t|\boldsymbol{s}_{t-1},\boldsymbol{a}_t,\boldsymbol{o}_t)$. All distributions are parameterized as i.i.d multivariate Gaussian distributions, i.e. the outputs of the neural networks are the means μ and standard deviations σ of each Gaussian. Sampling is done using the reparameterization trick, computing $\mu + \epsilon\sigma$ with $\epsilon \sim N(0,1)$, which allows for backpropagation of the gradients. Minimizing the free energy then boils down to minimizing the following loss function:

$$\begin{aligned} \forall t : \underset{\phi,\theta,\xi}{\text{minimize}} : &- \log p_\xi(\boldsymbol{o}_t|\boldsymbol{s}_t) \\ &+ D_{\mathrm{KL}}(p_\phi(\boldsymbol{s}_t|\boldsymbol{s}_{t-1},\boldsymbol{a}_{t-1},\boldsymbol{o}_t)\|p_\theta(\boldsymbol{s}_t|\boldsymbol{s}_{t-1},\boldsymbol{a}_{t-1})) \end{aligned} \tag{5}$$

Figure 1 shows an overview of the information flow between the transition model, approximate posterior and likelihood neural networks. To engage in active inference using these models, we need to estimate $G(\pi,\tau)$, which involves estimating $Q(\boldsymbol{s}_\tau|\pi)$. As our model takes a state sample as input, and only estimates the state distribution of the next timestep, the only way to get an estimate of the state distribution at a future timestep $\tau > t+1$ is by Monte Carlo sampling. Concretely, to infer $P(\pi)$, we sample for each policy N trajectories following π using p_θ. This results in N state samples $\hat{\boldsymbol{s}}_\tau$, for which we can get N observation estimates $\hat{\boldsymbol{o}}_\tau$ via p_ξ. To be able to calculate the KL divergence and entropy, we use a Gaussian distribution fitted on the samples' mean and variance. We then estimate the expected free energy for each policy from current timestep t onward as follows:

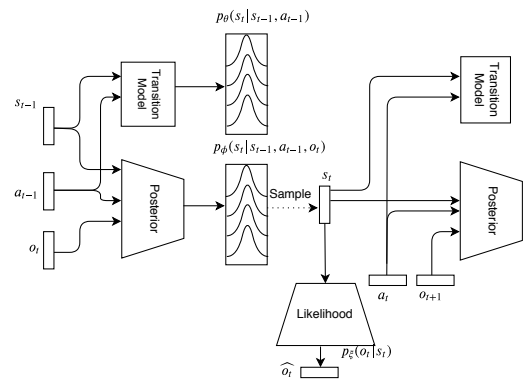


Fig. 1: We train simultaneously a transition model $p_\theta(\boldsymbol{s}_t|\boldsymbol{s}_{t-1},\boldsymbol{a}_t)$, an approximate posterior distribution model $p_\phi(\boldsymbol{s}_t|\boldsymbol{s}_{t-1},\boldsymbol{a}_t,\boldsymbol{o}_t)$, and a likelihood distribution $p_\xi(\boldsymbol{o}_t|\boldsymbol{s}_t)$ model, by minimizing the variational free energy.

$$\begin{aligned} \hat{G}_t(\pi) = &\sum_{\tau=t+1}^{t+K} D_{\mathrm{KL}}(\mathcal{N}(\mu_{\hat{\boldsymbol{s}}_\tau},\sigma_{\hat{\boldsymbol{s}}_\tau})\|P(\boldsymbol{s}_\tau)) + \frac{1}{\rho}H(\mathcal{N}(\mu_{\hat{\boldsymbol{o}}_\tau},\sigma_{\hat{\boldsymbol{o}}_\tau})) \\ &+ \sum_{\pi'} \sigma(-\gamma\hat{G}_{t+K}(\pi'))\hat{G}_{t+K}(\pi') \end{aligned} \tag{6}$$

The first summation term looks K timesteps ahead, calculating the KL divergence between expected and preferred states and the entropy on the expected observations. We also introduce an additional hyperparameter ρ, which allows for a trade-off between reaching preferred states on the one hand and resolving uncertainty on the other hand. The second summation term implies that after K timesteps, we continue to select policies according to their expected free energy, hence recursively re-evaluating the expected free energy of each policy at timestep $t+K$. In practice, we unroll this D times, resulting into a search tree with an effective planning horizon of $T = K \times D$.

4. EXPERIMENTS

We experiment with the Mountain Car problem, where an agent needs to drive the car up the mountain in 1D by throttling left or right, as shown on Figure 2. The top of the mountain can only be reached by first building up momentum before throttling right. The agent spawns at a random position, and only observes a noisy position sensor and has no access to its current velocity. At each timestep, the agent can choose between two policies: π_l to throttle to the left, π_r to throttle to the right. We experiment with two flavors of this environment: one where the agent starts with fixed zero velocity, and one where the agent starts with a random initial velocity.

For our generative model, we instantiate $p_\theta(\boldsymbol{s}_t|\boldsymbol{s}_{t-1},\boldsymbol{a}_t)$, $p_\phi(\boldsymbol{s}_t|\boldsymbol{s}_{t-1},\boldsymbol{a}_t,\boldsymbol{o}_t)$ and $p_\xi(\boldsymbol{o}_t|\boldsymbol{s}_t)$ as fully connected neural

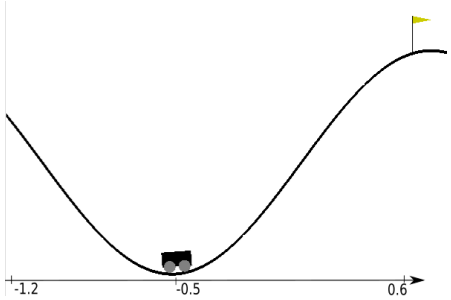


Fig. 2: The mountain car environment. The shown position of the car at -0.5 is the starting position in our evaluations.

networks with 20 hidden neurons, and a state space with 4 dimensions. To bootstrap the model, we train on actions and observation of a random agent minimizing the loss function in Equation (5) using stochastic gradient descent. Next, we instantiate an active inference agent that uses Equation (6) to plan ahead and select the best policy. As preferred state distribution $P(\boldsymbol{s}_\tau)$, we manually drive the car up the mountain and evaluate the model's posterior state at the end of the sequence $\hat{\boldsymbol{s}}_{end}$, and set $P(\boldsymbol{s}_\tau) = \mathcal{N}(\hat{\boldsymbol{s}}_{end}, 1)$. To limit the computations, the active inference agent plans ahead for 90 timesteps, allowing to switch policy every 30 timesteps, effectively evaluating a search tree with depth 3, using 100 samples for each policy ($K = 30, D = 3, N = 100$).

Figure 3 shows the sampled trajectories for all branches of the search tree, in the case the model is bootstrapped with only a single observation at position -0.5. This is a challenging starting position as the car needs enough momentum in order to reach up the hill from there. In the case of a random starting velocity, the generative model is not sure about the velocity after only the first observation. This is reflected by the entropy (i.e. the expected ambiguity) of the sampled trajectories. Now following π_r from the start will sometimes reach the preferred state, depending on the initial velocity. In this case the active inference agent's behavior is determined by the parameter ρ. For $\rho > 1$, the agent will act greedily, preferring the policy that has a chance of reaching the top early, cf. Figure 3e. When setting $\rho << 1$, the entropy term will play a bigger role, and the agent will select the policy that is less uncertain about the outcomes, rendering a more cautious agent that prefers a more precise and careful policy, moving to the left first see Fig. 3a. We found setting $\rho = 0.1$ yields a good trade-off for the mountain car agent.

In the environment with no initial velocity, the transition model learnt by the agent is quite accurate and the entropy terms are an order of magnitude lower, as shown in Figure 4. However, in terms of preferred state the lowest KL is still achieved by following π_r. This is due to the fact that the KL term is evaluated each timestep, and moving to the left, away from the preferred state in the sequence outweighs the benefit of reaching the preferred state in the end. Choosing $\rho = 0.1$ again forces the agent to put more weight on resolving uncertainty, preferring the policy in Figure 4a.

5. DISCUSSION

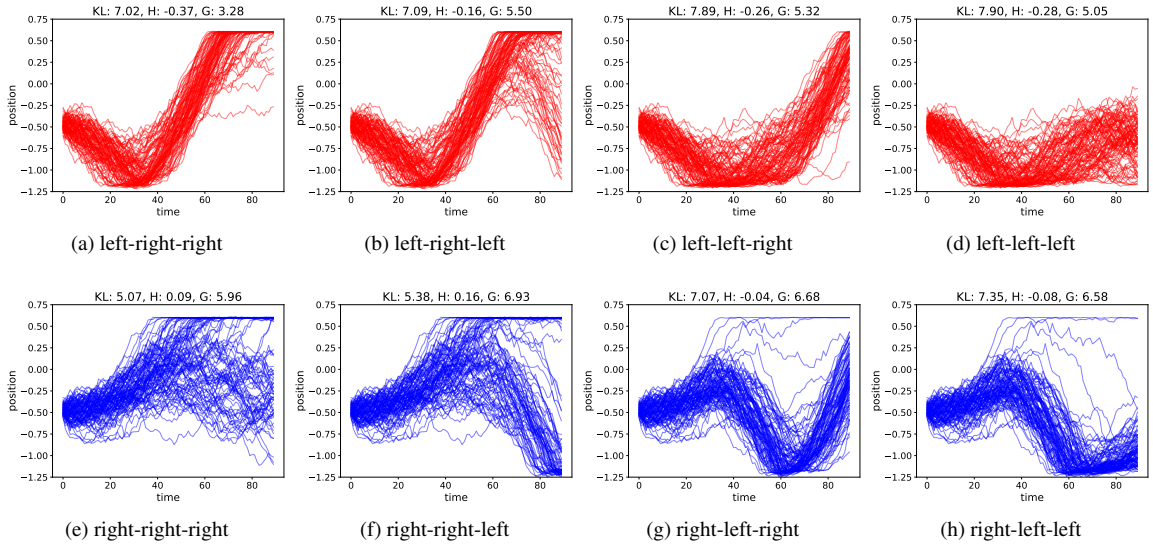
Using deep neural networks to instantiate the generative model and to approximate both prior and posterior distributions, has the advantage that the generative model is independent of any state representation. The model can learn the best state representation for the observed data. Employing deep neural networks also opens up the possibility of using high-dimensional sensor inputs, e.g. images. The downside of our model, however, is the required sampling step, which means that a distribution is only calculated for the next timestep, and distributions for timesteps τ further in the future can only be approximated by sampling.

Another point of discussion is the definition of the preferred state distribution. In our case we opted for a Gaussian state distribution, centered around the state visited by an expert demonstration, similar to our earlier work [9]. However, the standard deviation of this distribution will determine the absolute value of the KL term in Equation (6). A small standard deviation will blow up the KL term, completely ignoring the entropy term. A large standard deviation will assign probability mass to neighboring states, possibly introducing local optima that don't reach the actual goal state. To mitigate this, we introduced an additional ρ parameter that balances risk and ambiguity.

Finally, planning by generating and evaluating trajectories of the complete search tree is computationally expensive. In this paper, we intentionally pursued this approach in order to directly investigate the effect of the KL term versus the entropy term. To mitigate the computational load, one might amortize the resulting policy by training a policy neural network $p_\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$ based on the visited states and the planned actions by the agent, similar to [10]. In other approaches, the policy is learned directly through end-to-end training. For example, K. Ueltzhöffer uses evolution strategies to learn both a model and a policy, that requires part of the state space to be fixed, to contain information of the preferred state [11]. B. Millidge on the other hand amortizes the expected free energy G as function of the state, similar to value function estimation in reinforcement learning [12]. Again, however, the perception part is omitted and the state space is fixed upfront.

6. CONCLUSION

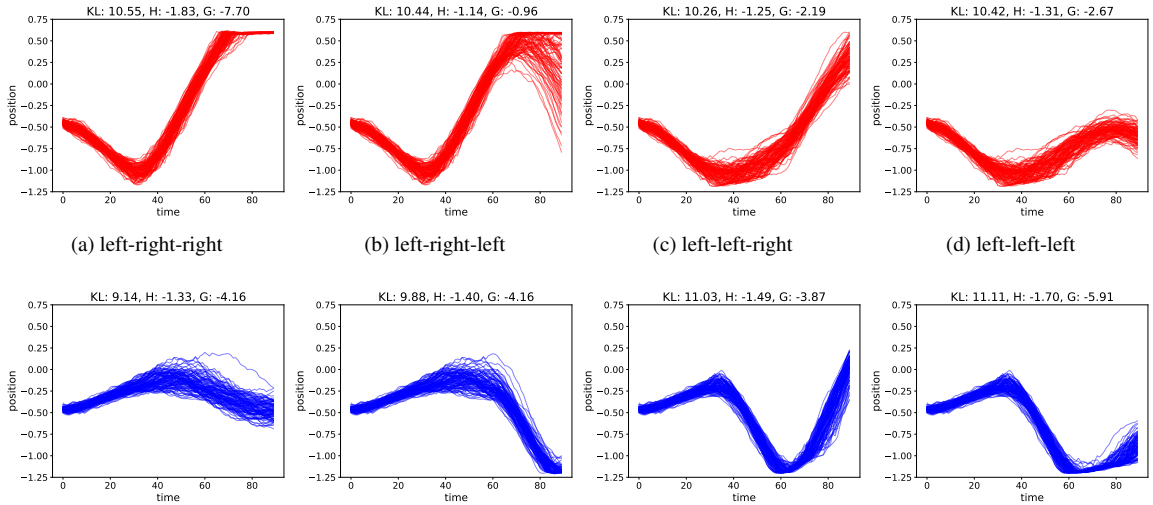
In this paper, we have shown how generative models parameterized by neural networks are trained by minimizing the free energy, and how these can be exploited by an active inference agent to select the optimal policy. We will further extend our models to work in more complex environments, in particular towards more complex sensory inputs such as camera, lidar or radar data.



(a) left-right-right (b) left-right-left (c) left-left-right (d) left-left-left

(e) right-right-right (f) right-right-left (g) right-left-right (h) right-left-left

Fig. 3: Depending on the random initial velocity, the car will reach the hill fast using the right policy only part of the cases (e), however starting with the left policy first also reaches the hill top and with lower entropy on the trajectories (a). A greedy agent ($\rho > 1$) will pick (e) whereas a cautious agent ($\rho << 1$) will favor (a). For each policy we report the values of KL, H and G for $\rho = 0.1$.



(a) left-right-right (b) left-right-left (c) left-left-right (d) left-left-left

(e) right-right-right (f) right-right-left (g) right-left-right (h) right-left-left

Fig. 4: When the environment starts the car with a fixed zero velocity, the model is on average much more certain on the predicted trajectories, resulting in lower entropy terms. However, policy (e) still achieves the lowest KL value, as this term is evaluated each timestep, and moving away from the preferred state yields a high KL penalty. When choosing $\rho = 0.1$, the agent again favors (a). For each policy we report the values of KL, H and G for $\rho = 0.1$.

7. REFERENCES

[1] Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo, "Active inference: A Process Theory," *Neural Computation*, vol. 29, pp. 1–49, 2017.

[2] Karl Friston, "The free-energy principle: A unified brain theory?," *Nature Reviews Neuroscience*, vol. 11, no. 2, pp. 127–138, 2010.

[3] Karl Friston, James Kilner, and Lee Harrison, "A free energy principle for the brain," *Journal of Physiology Paris*, vol. 100, no. 1-3, pp. 70–87, 2006.

[4] Karl Friston, Spyridon Samothrakis, and Read Montague, "Active inference and agency: Optimal control without cost functions," *Biological Cybernetics*, vol. 106, no. 8-9, pp. 523–541, 2012.

[5] Karl Friston, Philipp Schwartenbeck, Thomas FitzGerald, Michael Moutoussis, Timothy Behrens, and Raymond J. Dolan, "The anatomy of choice: active inference and agency," *Frontiers in Human Neuroscience*, vol. 7, no. September, pp. 1–18, 2013.

[6] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.

[7] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[8] Philipp Schwartenbeck, Johannes Passecker, Tobias Hauser, Thomas H B FitzGerald, Martin Kronbichler, and Karl J Friston, "Computational mechanisms of curiosity and goal-directed exploration," *bioRxiv*, p. 411272, 2018.

[9] Cedric De Boom, Tim Verbelen, and Bart Dhoedt, "Deep active inference for state estimation by learning from demonstration," in *Conference of Complex Systems (CCS), Satellite Symposium on Complexity from Cells to Consciousness: Free Energy, Integrated Information, and Epsilon Machines*, 2018.

[10] Ozan Catal, Johannes Nauta, Tim Verbelen, Pieter Simoens, and Bart Dhoedt, "Bayesian policy selection using active inference," in *Workshop on Structure & Priors in Reinforcement Learning at ICLR 2019 : proceedings*, 2019, p. 9.

[11] Kai Ueltzhöffer, "Deep active inference," *Biological Cybernetics*, vol. 112, no. 6, pp. 547–573, Dec 2018.

[12] Beren Millidge, "Deep active inference as variational policy gradients," *CoRR*, vol. abs/1907.03876, 2019.