COMBINING ACTIVE INFERENCE AND HIERARCHICAL PREDICTIVE CODING: A TUTORIAL INTRODUCTION AND CASE STUDY

A PREPRINT

Beren Millidge
Department of Informatics
University of Edinburgh
United Kingdom

March 11, 2019

ABSTRACT

This paper combines the active inference formulation of action (Friston et al., 2009) with hierarchical predictive coding models (Friston, 2003) to provide a proof-of-concept implementation of an active inference agent able to solve a common reinforcement learning baseline – the cart-pole environment in OpenAI gym. It demonstrates empirically that predictive coding and active inference approaches can be successfully scaled up to tasks more challenging than the mountain car (Friston et al., 2009, 2012). We show that hierarchical predictive coding models can be learned from scratch during the task, and can successfully drive action selection via active inference. To our knowledge, it is the first implemented active inference agent to combine active inference with a hierarchical predictive coding perceptual model. We also provide a tutorial walk-through of the free-energy principle, hierarchical predictive coding, and active inference, including an in-depth derivation of our agent.

1 Introduction

In the last few decades there has been a paradigm shift in computational and cognitive neuroscience towards Bayesian and Helmhotzian views of the brain (Gopnik and Tenenbaum, 2007; Friston, 2012; Knill and Pouget, 2004; Seth, 2014). These paradigms conceptualize the brain as fundamentally a prediction and inference machine, actively trying to predict, experiment with, and understand its environment. Additionally, recent currents of philosophy, such as enactivism and embodied cognition stress that we are fundamentally embodied beings, enmeshed and in constant engagement with the world. We cannot be platonic thinking machines abstractly speculating and processing information with philosophical detachment. Instead, our existence must be oriented towards a direct and active engagement with the world (Clark, 1999, 2017; Rowlands, 2009; Thompson and Varela, 2001). The free-energy principle has arisen as a unifying thread between these perspectives and promises not only a unified theory of the brain, but ultimately a new perspective on life, autopoiesis, and self-organization in complex far-from-equilibrium systems (Friston et al., 2006; Karl, 2012).

The free-energy principle states that any system which survives the dissipative pressures of the second law of thermodynamics must in some sense engage in both modelling and acting in the world so as to minimize an information-theoretic quantity called the variational free energy. This quantity has an algebraic form similar to, but is fundamentally different

from, the thermodynamic free energy of statistical mechanics. According to the free-energy principle, complex systems at any scale, from molecules to organisms (Friston, 2010), all must in some sense minimize their variational free energy (Friston, 2013; Kirchhoff et al., 2018; Ramstead et al., 2018). The free energy principle has been applied to cells organizing during development (Kiebel and Friston, 2011; Friston et al., 2015a), understanding brain function (Friston, 2009, 2010), and various psychiatric disorders (Lawson et al., 2014; Limongi et al., 2018)[1] The original home for the free-energy principle, however, has been in explaining and modelling brain function. Specifically, under several additional assumptions, the free energy principle can be applied to derive and generalize hierarchical predictive coding models (Rao and Ballard, 1999) which can be seen as a form of Bayesian filtering (Friston, 2003, 2005, 2008). Action can also be neatly tied in with perception under the free energy principle through a paradigm called active inference (Friston et al., 2009, 2010b, 2009). In this paper, we provide a case study of an agent that implements active inference using a hierarchical predictive coding perceptual model, which can learn to solve a common reinforcement learning task from the OpenAI gym. We also provide a detailed mathematical walk-though, starting from scratch, of the free-energy principle and how to apply it to derive both the hierarchical predictive coding model our agent uses, and the way it chooses its actions according to active inference.

1.1 The Free Energy Principle

The free-energy principle appears to be esoteric, but it ultimately derives from a very simple question: what properties *must* a complex system have in order to survive for an appreciable length of time? We consider the system and its surroundings from the perspective of dynamical systems theory. This means that there is some giant, extremely high-dimensional, probably infinite space of possible states that a system could be in. Over time the system moves between states according to some complex (and likely stochastic) set of rules. Complex systems and especially living biological systems are generally very finicky and fragile – they have to keep certain parameters such as their body temperature, water levels, and various chemical concentrations – within very tight bounds. If they do not, they die, and the system rapidly dissipates. Thus, for any biological system to survive over a long period, it must keep all these parameters within tight ranges over that period, which implies that the system must generally occupy a tiny proportion of all the states it could possibly occupy. This means that the probability distribution over the states the system could occupy is extremely peaked, with some states being very likely and the vast majority being of infinitesimally low probability. We can mathematically formalize this property of the probability distribution being peaked through the use of a quantity called entropy.The entropy $H(x)$ of a distribution x is a sum (or integral in continuous spaces) of the probability of each point multiplied by the log probability of each point. The entropy is largest when the distribution is uniform over all possible values and smallest when all the probability mass is at a single value. Entropy is defined as:

$$H(S) = -\int p(s)log(p(s))ds$$

In the equation above, the probability of a system being in a particular state is denoted p(s). A living system, then, must have low entropy in the state-space since it is only occupying a small fraction of the possible states with any appreciable probability. However, the living system is affected by fluctuations from the external environment. On average these fluctuations tend to push the system away from its safe high-probability states towards the dangerous low-probability states simply because there are many more low probability states than high-probability ones in the overall state-space, due to the relatively low entropy of the system (Mittag and Evans, 2003). If the system does nothing, therefore, it will slowly be pushed towards lower-probability states, which will disperse its probability mass in the state space, ultimately leading to its dissipation. To avoid this, the system must actively attempt to minimize its own entropy. [2]

[1] The free-energy principle has recently been proposed, under the heading of variational ecology, to also be an organizing principle of the dynamics of coupled systems of interacting components at larger scales, including entire societies. (Ramstead et al., 2019; Kirchhoff et al., 2018).

[2] Of course often a system does not "want" entropy to be fully minimized such that it is frozen into a single state forever. In reality this should rarely be an issue since random fluctuations should prevent any complete minimization and, in any case, this problem can

It is interesting to note that the minus sign in the entropy can be brought inside of the integral to get:

$$H(s) = \int -p(s)log(p(s))ds = E[-log(p(s))]$$

Here, $E[x]$ denotes the expectation of x. The expectation of x is effectively the average value of x over a large amount of samples. It is defined as $E[x] = \sum_i p(x_i)x_i$ in discrete spaces, or $\int p(x)xdx$ in continuous ones. The formula for the expectation is equal to the arithmetical mean if the distribution is uniform. Looking at the entropy formula, we see that entropy is just the expected value of a quantity $-log(p(s))$. This quantity is a measure of the unlikeliness of a state, and is often called surprisal [3]. If we flip the negative sign and assume that the probability is assigned by some model m, the quantity becomes $log(p(s|m))$. In Bayesian statistics this is known as the log model evidence. In effect, by saying that the system or organism must minimize its own entropy, we are also saying that it must minimize its expected surprisal, or alternatively maximize its own log-model evidence.[4]

There is a slight complication here. The organism should be minimizing the entropy of its own states in the state space, but it does not have direct access to its own states. Instead, we assume that the organism minimizes the entropy of its own observations as a proxy, which should ultimately minimize the entropy of its internal states. This works because there should be a sensible relation between the entropy of the observations and that of the states, due to the selective pressures of evolution. [5]. The core function of perception is to represent the external reality in a sensible way that is useful for action and ultimately for survival and reproduction. If an organism possessed no sensory receptors, or had receptors only that reported a constant percept, or else had receptors with a completely random mapping between external causes and sensory percepts, it would be rapidly out-competed by organisms which did possess sensory receptors with useful mappings between external states and percepts. Thus, all organisms we observe now, after billions of years of evolution, should inevitably have exquisitely optimized mappings between states and observations. [6]

This means we can write the minimization objective as the entropy of the observations of the organism.

$$H(o) = -\int p(o)log(p(o))do$$

One further assumption we must make is that the path the system takes through the state-space over time is ergodic – that the time and ensemble averages of the system are the same. This means that given a large number of identical systems which start at random points in the state space and evolve from there, the proportion of all systems in a specific state is the same as the proportion of time a single system spends in that state. [7] Under the assumption of ergodicity, we

be finessed by defining a 'state' to be some complex dynamical attractor moving through the previous state-space instead of just a bundle of physical values. Moreover, the law of requisite variety (Ashby and Goldstein, 2011) will prevent a complete minimization, as the entropy minimization capabilities of the system will themselves be degraded if the entropy minimization progresses to such an extent that the internal state-space of the system shrinks beyond that needed to model and control external perturbations.

[3] Surprisal is just $-log(p(x))$, a technical quantity and not related to the emotional/physiological concept of surprise, except insofar as low probability events are often surprising

[4] This is where the claim comes from that organisms are self-evidencing - they maximize their own likelihood of existence under their model.

[5] There is a proof in Appendix A of Friston et al. (2010b) that the entropy of the observations is an upper bound on that of the states. This proof, however, assumes the mapping between states and observations is diffeomorphic (i.e. smooth and invertible). This is not the case in general. For instance in vision the mapping is definitely not invertible as objects out in the world (hidden states) can occlude one another, making the occluded regions impossible to reconstruct. Additionally, we should expect there to be a huge dimension compression between all the hidden states in the world and the actual observations an organism can make, which depend upon the sophistication of its sensing apparatus.

[6] This argument applies less well to larger applications of the free energy principle, for instance to societies. Societies are probably under much less evolutionary pressure, have been around for an utterly minuscule amount of time compared to biological organisms, and likely evolve on much slower timescales.

[7] In effect ergodicity requires that the transitions in the state-space need to be reversible so a system can always (eventually) get back to where it left.

can write the entropy as a function of time instead of of the observations, so the system must now minimize:

$$H(O(t)) = \lim_{T-\infty} -\frac{1}{T}\int_0^T log(p(o(t)))dt$$

Now, instead of the system having to minimize its entropy over every possible observation, which is likely infinite, it only needs to minimize the entropy of the observation it receives at every point in time, an objective which is at least potentially tractable. Despite this simplification, the surprisal $-logp(o(t))$ is still not something that can be realistically calculated by an organism. This is because, in reality, the surprisal must be relative to a model (the probability $p(o)$ has to come from somewhere), and this model likely has parameters of some sort. The surprisal, therefore, must be computed as a sum or integral over all the various parameters of the model.

$$log(p(o(t)) = \int log((p(o|\theta)) + log(p(\theta))d\theta$$

Where θ is the set of all the possible parameters specifying the model. For very simple models, this quantity might be directly computable, but for more complex models, it must be approximated. Luckily this quantity can be approximated using variational inference. Variational methods have a long history, originating in statistical physics (Blanchard and Brüning, 2012) and then being applied in Bayesian statistics and Machine Learning (Ghahramani and Beal, 2001; Beal et al., 2003; Blei et al., 2017), where it is generally used to estimate the posterior over the parameters of a model given the data – i.e. approximate $p(\theta|D)$ where θ are some set of model parameters, and D is a large dataset. For many complex models, computing this exactly is intractable because of the need to compute the normalization term $p(D)$ in the Bayesian formula $p(\theta|D) = \frac{p(D|theta)p(\theta)}{p(D)}$. This normalization term $p(D)$ is directly analogous to our surprisal term $log(p(o))$. [8]

Variational methods approximate the posterior by creating some arbitrary density Q, which we have total control over, and updating the parameters of Q to minimize the divergence between our variational density Q and the true posterior. When this divergence is fully minimized, we can take our variational density Q to be our approximation of the true posterior [9]. The divergence that is minimized is usually the KL divergence [10], which is defined as:

$$KL[Q||P] = \int Q(x)log(\frac{Q(x)}{p(x)})dx$$

The KL divergence effectively measures the overlap between two distributions. In information-theoretic terms, the KL divergence is the expected extra cost, in bits, of sending a message with the statistics of Q encoded in a code that is designed for the statistics of P. The KL divergence is always greater than or equal to zero and is equal to zero only when the two distributions are identical. We try to minimize the divergence between the posterior of the parameters of our model given the observations $p(\theta|o)$ and our variational distribution $Q(\theta;\Psi)$ where Ψ are the set of parameters we use to specify our variational distribution. The variational distribution Q and its parameters Ψ are represented implicitly or explicitly in the brain of the organism, while the parameters θ represent states of the external world. The KL between the two is thus:

$$KL[Q(\theta;\Psi)||p(\theta|o)] = \int Q(\theta;\Psi)log(\frac{Q(\theta;\Psi)}{p(\theta|o)})dQ$$

[8] The main other method used to approximate densities in Bayesian statistics is Markov Chain Monte-Carlo (MCMC). This involves taking samples from a Markov chain on the posterior distribution and using those samples to approximate the distribution. It is generally assumed that this method is a poor fit for the brain because of its computation-heavy and extremely serial nature. Nevertheless, it is potentially possible that the brain may implement sampling schemes somewhere.

[9] This does not necessarily mean that the variational density Q is a good approximation. Many things can go wrong. One of the most obvious is if the family of distributions Q is a poor match for P, for instance if Q is unimodal and P is heavily multimodal then generally Q will only find one peak.

[10] Recent advances in variational methods include using other possible divergences. One currently popular candidate is the α-divergence (Li and Turner, 2016; Hernández-Lobato et al., 2016), which is a generalization of the KL divergence and has many useful properties.

We can now use the definition of conditional probability that $p(\theta|o) = \frac{p(\theta,o)}{p(o)}$ to obtain:

$$KL[Q(\theta;\Psi)||p(\theta|o)] = \int Q(\theta;\Psi)log(\frac{Q(\theta;\Psi)p(o)}{p(\theta,o)})dQ$$

Using the fact that for logarithms $log(a*b) = log(a)+log(b)$ and the linearity of the integral - i.e. $\int a+b = \int a + \int b$, we see we can split the KL divergence to obtain:

$$KL[Q(\theta;\Psi)||p(\theta|o)] = \int Q(\theta;\Psi)log(\frac{Q(\theta;\Psi)}{p(\theta,o)})dQ + \int Q(\theta;\Psi)log(p(o))dQ$$

Looking at the second term, $p(o)$ has no dependence on the variational parameters Ψ in Q. This means that $p(o)$ is unaffected by the integral and, because Q is a probability distribution, it must integrate to 1 – i.e. $\int Q(\theta;\Psi)dQ = 1$. So, the second term becomes:

$$\int Q(\theta;\Psi)log(p(o))dQ = log(p(o)) \int Q(\theta;\Psi)dQ = log(p(o)*1 = log(p(o))$$

This means the full expression becomes:

$$KL[Q(\theta;\Psi)||p(\theta|o)] = \int Q(\theta;\Psi)log(\frac{Q(\theta;\Psi)}{p(\theta,o)})dQ + log(p(o))$$

Rearranging this to isolate the log probability, we get:

$$-log(p(o)) = \int Q(\theta;\Psi)log(\frac{Q(\theta;\Psi)}{p(\theta,o)})dQ - KL[Q(\theta;\Psi)||p(\theta|o)]$$

We call the second term - $\int Q(\theta;\Psi)log(\frac{Q(\theta;\Psi)}{p(\theta,o)})dQ$ the free-energy, and denote it as F. We thus get

$$-log(p(o)) = -F - KL[Q(\theta;\Psi)||p(\theta|o)]$$

Since the KL term is always ≥ 0, the free energy F is a lower bound on the negative log probability, or surprisal. Thus to minimize the negative log probability, or surprisal, all we have to do is to minimize F. This F is the variational free energy described above, and this is precisely what it means to say that the brain, or all living systems, must minimize their variational free energy.

Let's look at F in more detail. We can break it down into two components using the properties of logs that $log(\frac{a}{b}) = log(a) - log(b)$ and the linearity of the integral as before:

$$F = -\int Q(\theta;\Psi)log(\frac{Q(\theta;\Psi)}{p(\theta,o)})dQ \tag{1}$$

$$= \int Q(\theta;\Psi)log(p(\theta,o)dQ - \int Q(\theta;\Psi)log(Q(\theta;\Psi))dQ \tag{2}$$

The first of these two terms is $\int Q(\theta;\Psi)log(p(\theta,o)dQ$. Looking closely, we can see that it is the expectation under the variational distribution of the joint distribution of the observations and states of the environment. This term effectively scores the closeness of the variational distribution Q to the true distribution P. This term is called the *energy* term, by analogy with similar terms in statistical mechanics, which do actually represent the energy of a system.

The second term is $\int Q(\theta;\Psi)log(Q(\theta;\Psi))dQ$. This is just the entropy of the variational distribution Q – recall that the definition of entropy is $H(x) = \int p(x)log(p(x))dx$ – and so the second term is called the *entropy*. We thus get the "equation":

$$\textit{Free-Energy} = \textit{Energy} - \textit{Entropy}$$

This formulation is why F is called the variational free-energy, as there is a direct correspondence with an analogous quantity in statistical mechanics. It is worth reiterating, however, that the analogy is only on the surface level: there is no intrinsic connection between the variational free energy and the thermodynamic free energy.

To recap, we have seen that for any complex biological system to continue to exist, it must primarily occupy a tiny proportion of the states it could potentially occupy. This means that the system must minimize the entropy of its states or, equivalently its average surprisal, and it must do this actively to guard against random fluctuations. Since it cannot minimize directly on the external states, it must minimize its observations. Directly minimizing the entropy of these observations is still generally intractable so instead it minimizes an upper bound on the entropy called the variational free-energy. That is why all living systems must minimize variational free-energy.

At this level of generality, the theory is still vague and high-level – more of a general principle than a falsifiable theory. In the next section, we drill down deeper and see how under specific assumptions about the variational density Q, we can derive a model of hierarchical predictive coding that could, in theory, be implemented in the brain.

1.2 Hierarchical Predictive Coding

In this section we derive hierarchical predictive coding, as expounded in (Friston, 2003, 2005). Hierarchical Predictive Coding can be derived from a simple assumption that the variational density Q takes the form of a multivariate Gaussian[11]. A slight shift in interpretation is required here. Previously in the $p(o, \theta)$ term, the θ referred to external states out in the environment. However, an agent has no way of accessing these external states directly and thus cannot compute anything regarding these states. Instead, the agent must build an internal model of these external states and use the model to predict the observations it receives. From now on the $p(o, \theta)$ is the agent's internal model of the environment and its observations, and the θs are states and parameters of the agent's internal model - not of existing states in the outside world. The agent's model $p(o, \theta)$ is called a generative model or generative density since it can be used to generate predicted "observations" given some setting of the model parameters. This capability arises from the fact that $p(o, \theta) = p(o|\theta)p(\theta)$ – i.e. one can select a θ and then obtain the distribution $p(o|\theta)$ from which one can sample expected observations. The variational density Q now becomes an approximation of the posterior, or recognition, density $p(\theta|o)$ which is the distribution over and parameters given the observations encountered. In the brain, these parameters may be thought of as being instantiated by physical brain states, such as synaptic connection weights.

The hierarchical predictive coding models assumes that Q is a multivariate Gaussian, such that:

$$Q(\theta; \mu, \Sigma) = N(\theta; \mu, \Sigma)$$

The free-energy formula then becomes:

$$F = \int N(\theta; \mu, \Sigma) log(p(o, \theta)) - \int N(\theta; \mu, \Sigma) log(N(\theta; \mu, \Sigma))$$

The second term, corresponding to the entropy in the free-energy equation, is just the entropy of a Gaussian for which a known analytic result exists (see appendix A for a full derivation). This result has no serious dependence on the variational parameters μ, Σ, so for the purposes of optimizing these parameters, we can ignore it. Thus, we focus in on the first term:

$$\int N(\theta; \mu, \Sigma) log(p(o, \theta))$$

To simplify this expression, we make a further assumption. We approximate the variational density with another normal distribution with the same mean, and a covariance matrix that is an analytic function of the mean. This is called the Laplace approximation (Friston et al., 2007). This means that we no longer need to worry about the variance terms of the variational distribution but only the mean (For a full derivation, see appendix B). In terms of brain function, this means that the brain only explicitly represents the most likely state of the world and not a full distribution over states

[11] Other assumptions about Q derive different theories. For instance, there is a mean-field approximation (Friston et al., 2010b; Friston, 2008; Friston et al., 2010a, 2012) which assumes that Q is split into independent sub-densities such that $Q(\theta) = \prod_i Q(\theta_i; \Psi_i)$. Recent work in variational inference has involved using deep neural networks to parametrize Q instead of simple known densities (Hoffman et al., 2013; Kingma and Welling, 2013; Rezende and Mohamed, 2015)

and observations. Although this might seem like it throws away all representation of uncertainty, this is not the case as uncertainty can still be encoded in a quantity called precision, which will be introduced later. With this approximation, we are able to do away with the integral to derive a simple expression for the free energy:

$$F \approx log(p(\mu, \theta))$$

Where μ denotes the mean of the variational density. To progress further, we now need to make some assumptions about the generative model $p(\mu, \theta)$ that the organism has. We assume that this density is also Gaussian. Additionally, we assume that the agent has a hierarchical model, meaning that the agent believes that the world is composed of hierarchical layers with higher levels directly causing the states and causes of the levels below. Hierarchical models are useful because they can represent the real hierarchical causal structures in the world as well as largely solve the problem in Bayesian reasoning of where priors come from in the first place. In a hierarchical model, the priors are simply the states of the layer above. For the top layer there still needs to be exogenous priors, but flat priors here seem much more reasonable than at the lower levels. Hierarchical models are also consonant with the organization of the brain – especially the sensory cortices, which are organized into a hierarchy of layers of increasing abstraction (although the brain is not strict with this, and there are many skip and cross-connections).

The hierarchical generative model proposed for the organism is as follows:

$$o = f_0(\mu_1, \theta_0) + z_0 \tag{3}$$

$$\mu_1 = f_1(\mu_2, \theta_1) + z_1 \tag{4}$$

$$\mu_2 = f_2(\mu_3, \theta_2) + z_2 \tag{5}$$

$$... \tag{6}$$

Where f is some arbitrary, likely nonlinear function, μ and θ are the means of the recognition density and the parameters of the generative model, and z is some Gaussian noise. This model states that the representations at each level is some function of the representations at the level above, plus noise. The representations at the lowest level of the hierarchy are the observations and the top layer is just noise ($\mu_{max} = z_{max}$). We can write the probability density of the representation of one layer, given the layer above as:

$$p(\mu_i|\mu_{i+1}) = N(\mu_i; f(\mu_{i+1}\theta), \Sigma_{z_i})$$

Since a layer only depends on the layer above, this means that the generative model factorizes into a product of each layer.

$$p(\mu, \theta) = \prod_i p(\mu_i, |\mu_{i+1}\theta_i)$$

Since the free-energy is approximated simply as the log probability of the generative model, then by the properties of logs that $log(a * b) = log(a) + log(b)$, the product of densities becomes a sum of log densities such that:

$$F \approx log(p(\mu, \theta)) = \sum_i log(p(\mu_i|\mu_{i+1}\theta_i))$$

Since the probability density for each layer is a Gaussian, we can replace each density with a Gaussian to get:

$$F \approx \sum_i log(N(\mu_i; f(\mu_{i+1}\theta), \Sigma_{z_i}))$$

Then, substituting in the formula for the probability density of a multivariate Gaussian - $N(x; \mu, \Sigma) = \frac{1}{2\pi|\Sigma^{\frac{1}{2}}|}e^{(x-\mu)^T\Sigma^{-1}(x-\mu)}$, we can write the free energy as:

$$F \approx \sum_i log(\frac{1}{2\pi|\Sigma_i^{\frac{1}{2}}|}e^{(\mu_i - f(\mu_{i+1}, \theta_i))^T\Sigma_i^{-1}(\mu_i - f(\mu_{i+1}, \theta_i))}) \tag{7}$$

$$\approx \sum_i (\mu_i - f(\mu_{i+1}, \theta_i))^T\Sigma_i^{-1}(\mu_i - f(\mu_{i+1}, \theta_i)) - \frac{1}{2}log(2\pi\Sigma_i) \tag{8}$$

Looking at the repeated terms $\mu_i - f(\mu_{i+1}, \theta)$, we see this is just a *prediction error* between the representation predicted from the layer above and the actual representation at that layer. At the lowest level, this prediction error is between the actual observations o and the predicted observations. Let us denote the prediction error as $\epsilon_i = \mu_i - f(\mu_{i+1}, \theta)$. Thus, the free energy can be expressed as a simple quadratic sum of prediction errors at each level:

$$F = \sum_i \epsilon_i^T \Sigma_i^{-1} \epsilon_i - \frac{1}{2} log(2\pi\Sigma_i)$$

The Σs encode the precision, or inverse variance of the prediction errors. The brain can represent the uncertainty about its predictions and observations by using these precisions to modulate the relative strength of the various prediction errors. To minimize the free-energy, it is necessary to adjust the parameter μ of the variational density as well as the parameters θ and Σ of the generative model. Although there is no analytic solution to this minimization, it can be achieved through a simple gradient descent scheme. This works by taking the gradient of the free-energy with respect to each of the parameters and then repeatedly updating the values of the parameter in the direction of the gradients until the free energy is minimized. These gradients can be calculated straightforwardly. As each θ_i and Σ_i is only found at a single layer, these gradients can be calculated in a simple layer-wise fashion. The μ parameters appear in their layer as the observation and also as part of the prediction from the layer above, which means two layers need to be considered in their derivation. The gradient for the precisions with respect to the free energy can be derived as:

$$\frac{dF}{d\Sigma} = \frac{d}{d\Sigma}(\epsilon_i^T \Sigma_i^{-1} \epsilon_i) - \frac{d}{d\Sigma}(\frac{1}{2} log(2\pi\Sigma_i)) \tag{9}$$

$$= \Sigma_i^{-1} \epsilon_i \epsilon_i^T \Sigma_i^{-1} - \Sigma_i^{-1} \tag{10}$$

The first step follows from the linearity of the derivative such that $\frac{d}{dx}(a+b) = \frac{da}{dx} + \frac{db}{dx}$. The second step is done by taking the matrix derivatives – a result which can be looked up in a matrix calculus reference book such as Petersen et al. (2008).

The gradient for the weights θ is as follows:

$$\frac{dF}{d\theta_i} = \frac{d}{d\theta_i}(\epsilon_i^T \Sigma_i^{-1} \epsilon_i) \tag{11}$$

$$= 2\epsilon_i \Sigma_i^{-1} \frac{d\epsilon_i}{d\theta_i} \tag{12}$$

$$= 2\epsilon_i \Sigma_i^{-1} \frac{df}{d\theta_i} \mu^T \tag{13}$$

The second step uses the product rule – $\frac{duv}{dx} = v\frac{du}{dx} + u\frac{dv}{dx}$ – to push the derivative through the product while the second step uses the chain rule to push the derivative through the function, and ultimately differentiate the linear combination $\frac{d\theta^T \mu}{d\theta} = \mu^T$.

The gradient for the means μ of the variational density is similar and derived as follows:

$$\frac{dF}{d\mu_i} = \frac{d}{d\mu_i}(\epsilon_i \Sigma_i^{-1} \epsilon_i) + \frac{d}{d\mu_i}(\epsilon_{i-1} \Sigma_{i-1}^{-1} \epsilon_{-1})$$

The derivative of the first term can be calculated as follows, in a manner directly analogous to that of the derivative of the weights θ above:

$$\frac{d}{d\mu_i}(\epsilon_i^T \Sigma_i^{-1} \epsilon_i) = 2\epsilon_i \Sigma_i^{-1} \frac{d\epsilon_i}{d\mu} \tag{14}$$

$$= 2\epsilon_i \Sigma_i^{-1} \frac{df}{d\mu_i} \theta^T \tag{15}$$

The derivative of the second term can be calculated simply:

$$\frac{d}{d\mu_i}(\epsilon_{i-1}\Sigma_{i-1}^{-1}\epsilon_{-1} = \epsilon_{i-1}\Sigma_{i-1}^{-1}\frac{d\epsilon_{i-1}}{d\mu_i} \tag{16}$$

$$= 2\epsilon_{i-1}\Sigma_{i-1}^{-1}\frac{d}{d\mu_i}(\mu_i - f(\mu_{i+1}, \theta_i)) \tag{17}$$

$$= 2\epsilon_{i-1}\Sigma_{i-1}^{-1} * 1 \tag{18}$$

The first step simply applies the product rule in the same way that was used for the derivative of the weights. The second step simply expands out the prediction error using its definition. The derivative of this prediction error is 1 since the second $f(\mu_{i+1}, \theta_i)$ term contains no μ_i and thus falls out of the derivative. Thus, the total rule for updating the weights is as follows:

$$\frac{dF}{d\mu_i} = 2\epsilon_i\Sigma_i^{-1}\frac{df}{d\mu_i}\theta^T + 2\epsilon_{i-1}\Sigma_{i-1}^{-1} * 1$$

These learning rules are largely biologically plausible: they only rely on local connectivity [12] and only use pre or post-synaptically available quantities. Additionally if the prediction errors and μs are represented in different populations of neurons, then weight updates correspond to simple associative Hebbian plasticity. Detailed neural implementations of this scheme have been proposed (see Friston (2005); Bastos et al. (2012); Kanai et al. (2015) for details). In general these neural implementations propose that the prediction errors are calculated in the superficial layers and are transmitted up the cortical hierarchy, synapsing onto the layer IV spiny stellate cells of the region above. Predictions are computed in the deep layers and are transmitted down, synapsing into the superficial layers of the region below.

To recap, the hierarchical predictive coding scheme has been derived directly from the formula for the free-energy, and the mathematical assumptions underlying this scheme have been detailed. In theory, the hierarchical predictive coding scheme described above enables an agent to process its sense-data to find and learn hierarchical patterns and regularity in its incoming sense-data and then to use these learned representations of the world to predict possible future observations. We now turn to the other side of the equation – how an agent might learn to act to change the world instead of just passively perceiving it.

1.3 Active Inference

Hierarchical predictive coding enables agents to perceive and learn a hierarchical structure in its environment. But agents are not only able to perceive; they also can act in the world to change what they are perceiving. This is the premise of active inference (Friston et al., 2009, 2011; Brown et al., 2011). Since an agent is driven to minimize its free energy, its actions as well as its perceptions must also minimize its free energy. This sets up a double optimization process whereby the agent must minimize free energy simultaneously through both perception and action.

$$\theta = \min_a F(o, a, \theta) \tag{19}$$

$$a = \min_a F(o, a, \theta) \tag{20}$$

Rewriting the free energy from above to now include action, we get:

$$F = \int Q(\theta; \Psi) log(\frac{Q(\theta; \Psi}{p(\theta, o, a)})dQ$$

This can be rearranged to obtain:

$$F(o, a, \theta) = E_{q(\theta)}[-log(p(o, a|\theta))] + KL[q(\theta)||p(\theta)]$$

[12] Except for the update rule for the precisions, and this can be fixed with a slight variation upon the scheme (Bogacz, 2017).

The first term means that to minimize free energy, the agent needs to seek out the sensations which it expects to encounter.[13] This enables abstract goals to be programmed into the agent by altering its priors over its expected sensations. This formulation cleverly converts what would previously be control, planning, or reinforcement learning problems (Pezzulo, 2012; Friston et al., 2012) into inference problems. It follows a large literature on planning as inference (Botvinick and Toussaint, 2012; Attias, 2003; Rawlik et al., 2013, 2010) and risk-sensitive KL control (Rawlik et al., 2013), while generalizing these schemes into a single framework. The various terms in the free-energy can be rearranged to give terms for extrinsic (utility) vs intrinsic (epistemic) value, which has been claimed (Friston et al., 2015b) to lead to agents which value purely epistemic rewards, thus theoretically dissolving the exploration/exploitation dilemma. The active inference framework has been applied to a wide range of tasks including optimal choice tasks (Friston et al., 2013), simple maze tasks with epistemically important cues (Friston et al., 2012, 2016), a simple reinforcement learning mountain car task (Friston et al., 2009), and others. Furthermore, active inference in the brain has been formulated as a neurophysiologically realistic process theory (Friston et al., 2017), which has generated predictions compatible with a wide range of electrophysiological findings such as mismatch negativity, repetition-suppression, and the theta and gamma band oscillations, and for which the variational updates are biologically plausible and strongly resemble associative plasticity.

Despite this wide application, few attempts appear to have been made to test the framework on more challenging tasks than the mountain car or the rat in the T-maze. Additionally, few active inference experiments deal with learning the generative model from scratch instead of assuming it a-priori and then observing the resultant inference dynamics. Finally, even when learning is defined, the models are often relatively simple and non-hierarchical. It is currently empirically unknown whether, despite its strong theoretical background, active inference is able to scale to solve tasks and challenges comparable to those faced in contemporary reinforcement learning. Nor is it known if the active inference framework can be integrated with other predictive processing models of perceptual inference and learning.

The unique contribution of this paper is that it brings together these two models of hierarchical perceptual predictive processing models and active inference to provide a proof-of-concept "end-to-end" predictive processing agent that is able to model and solve a complex, non-toy task using an entirely biologically plausible predictive processing network. The agent also learns its generative model of the world from scratch while the task is in progress and uses it to infer actions via active inference.

2 Methods

Active inference is typically set-up over a Partially Observable Markov Decision Process (POMDP) (Friston et al., 2012). This process contains a generative process which models the external environment and which produces observations for the agent, given the agent's previous actions. As this process represents the environment, its exact internal workings are hidden from the agent. From the perspective of the agent, the generative process can be modelled as:

$$R(o_{t+1}|o_t a_t)$$

[13] There is an important and subtle point here. Naively, this seems to catastrophically intertwine perception and goals such that expectations on action would contaminate perception, and vice-versa. This is a real issue and has caused several difficulties in our implementation. There are several subtle ways to finesse this problem. One way is to make a distinction between perceptual models and transition models. If there are hidden states in the world that map to observations, and over time the agent transitions between states, the agent builds two separate models - one mapping observations to hidden states - a perceptual model - and a separate model mapping hidden states at time t to expected hidden states at time t+1 - the transition model. Action expectations and goals therefore affect the transition model but not the perceptual model. For instance, if you are hungry and try to reach for some food, this doesn't involve perceiving yourself already reaching for or eating the food when you are not, but instead makes you perceive yourself being more likely to transition towards state in which you are reaching for the food. A related means of finessing this is to have the agent predict sequences of future events,and the goal states only affect expected prediction in the future,which avoids the issue of contaminating the current perceptual model.

Where the dependence of the observations upon the hidden states of the environment has been marginalized out as the agent is never able to observe these states. In order to minimize the free energy in such an environment, the agent must be equipped with a structured generative model that can infer hidden states of the environment, predict future observations, and predict the effect of its own actions on the environment. A very general form of such a generative model can be written as:

$$p(o, s, u, \theta) = p(o|s, \theta)p(s|u, \theta)p(u|\theta)$$

Here the u denote control states which represent the actions that the agent wishes to take. The form of the generative model implies that the agent believes that in the world there are states, observations, and actions. Observations are derived solely from states, and that states are affected by previous states and its own actions.

Our agent exists in a world that advances by discrete time-steps. Continuous time formulations of active inference exist (Friston et al., 2009, 2010b), but we do not engage with them here. The states and observations in the generative model refer to whole trajectories of future states and observations. If we assume the Markov property, the generative model can be further factorized such that the next state and current observation only depend on the current state.

$$p(o|s, \theta) = \prod_t p(o_t|s_t, \theta) \tag{21}$$

$$p(s|u, \theta) = \prod_t (p(s_t|s_{t-1}, u_{t-1}, \theta) \tag{22}$$

The environment chosen to test the model was the OpenAI gym cart-pole environment (Brockman et al., 2016). This is a baseline reinforcement learning environment that is simple for state-of-the-art reinforcement learning algorithms, but it is definitely not a toy-task. The agent must learn to balance a pole atop a cart in a simple physics simulator. The pole starts slightly unbalanced and will quickly topple over if left to its own devices. The agent can take only two possible actions – it may exert a force on the cart to the left and the right. The movement of the cart then transfers force to the pole, with the objective of balancing it. If the pole topples more than 15 degrees from the vertical in any direction, or if the cart is more than 2.4 units away from the center of the environment, the agent will lose and the task will reset.

An active inference agent was implemented that instantiated the equations described above. The agent assumed that the observations it received were a correct description of the hidden states – $p(o|s) = I$ – a correct assumption in the cart-pole environment. The prior the agent attempted to reach by active inference was simply $[0, 0, 0, 0]$, which requires that the pole is upright and stationary, and the cart is centered and stationary.

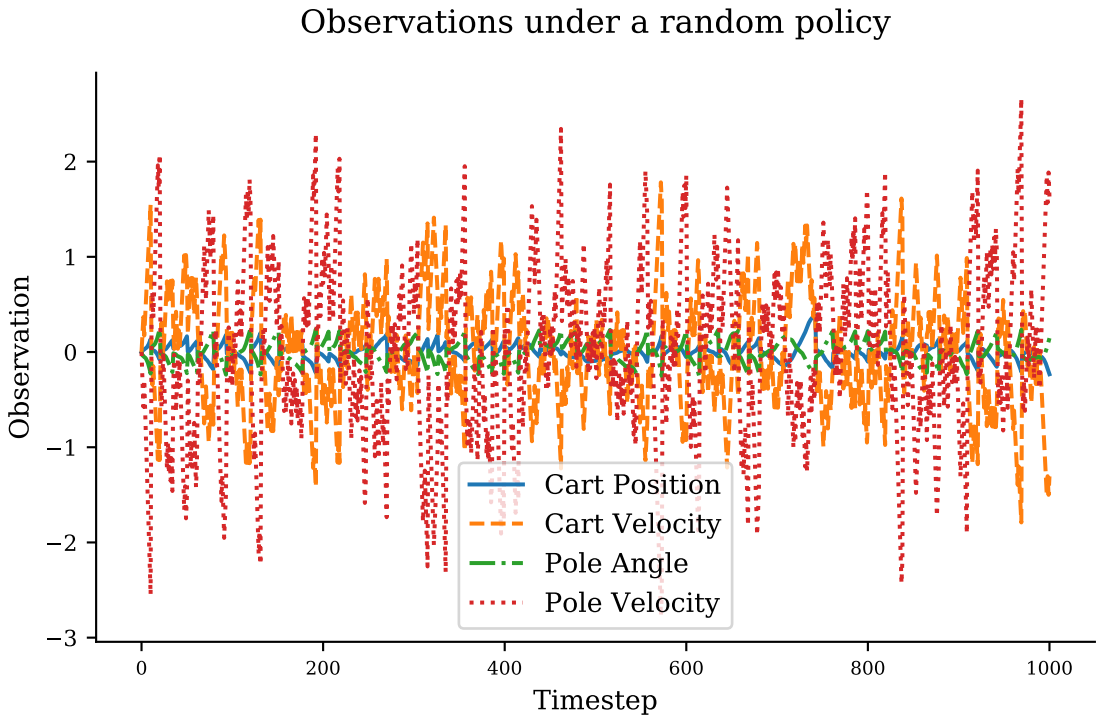
The state transition probabilities $p(s_t|s_{t-1})$ were modelled as a three layer linear predictive coding model. At each time-step, the active inference agent used its hierarchical generative predictive coding model to predict the expected observations given each of its two actions and then pick the action that most minimized the free energy under the generative model. The agent was guided to keep the cart-pole upright through strong priors that the pole should be upright at 0 degrees from vertical and moving with 0 speed. The predictive coding perceptual model began randomly initialized, with no knowledge of environmental dynamics: the agent had to learn the dynamics using the hierarchical predictive coding update equations while the task was in progress.

The hierarchical predictive processing model consisted of three layers above the incoming sense data, each consisting of 20 neurons. Knowledge of the previous action was added to the model as a simple weighted linear addition to the representation units in the first layer. Their activation was defined as:

$$\mu_1 = f(\theta_2^T \mu_2) + \theta_{a1}^T a_1 + \theta_{a2}^T a_2$$

Where a_1 and a_2 are kronecker delta functions which are only "on" when the action a1 or a2 are selected. These are thus the control states u of the active inference generative model. For simplicity these are not probabilistic, and there is an exact one-to-one mapping between the control state being activated by the generative model and the relevant action being taken in the environment.

The observations fed to the agent by the environment on every time-step consisted of four variables: the position of the cart, the speed of the cart, the angle of the pole, and the velocity of the pole. These observations evolve rapidly as the task progresses. A graph of the value of the observations fed to the agent under a random policy is shown below. As is clear from looking at the graph, predicting these observations and using them to guide action is not a trivial task.



The network was trained for 1000 epochs at a time. Between each time step, the value of the representation units μ were inferred for 100 inference-steps, during which no updates to the weights θ was undertaken. The learning rate was set to 0.0005. All weights were initialized from a normal distribution centered at 0 with a standard deviation of 1. At each time-step, predicted future observations were generated by the network and the action which generated future observations with the smallest free energy was selected. The trajectories of future states and actions were truncated in the generative model at length one, meaning that the agent acted "greedily" with respect to free energy, only maximizing the expected free energy of the action on the next state. In a relatively straightforward task like the cart-pole, where there are no subtle long-term dependencies between states, this is a reasonable assumption made for reasons of ease of computation.

3 Results

The performance of the agent was benchmarked against two very simple agents: a random agent, which at each timestep picked an action uniformly at random, and a naive control agent, which picked actions to push the pole to the left whenever it was to the right, and to push the pole to the right whenever it was toppling to the left.

Each agent was run for 50 trials. Each trial lasted for 1000 epochs, where each epoch is a single interaction with the environment. The average reward obtained over these trials for each agent is shown in Figure 1 below.

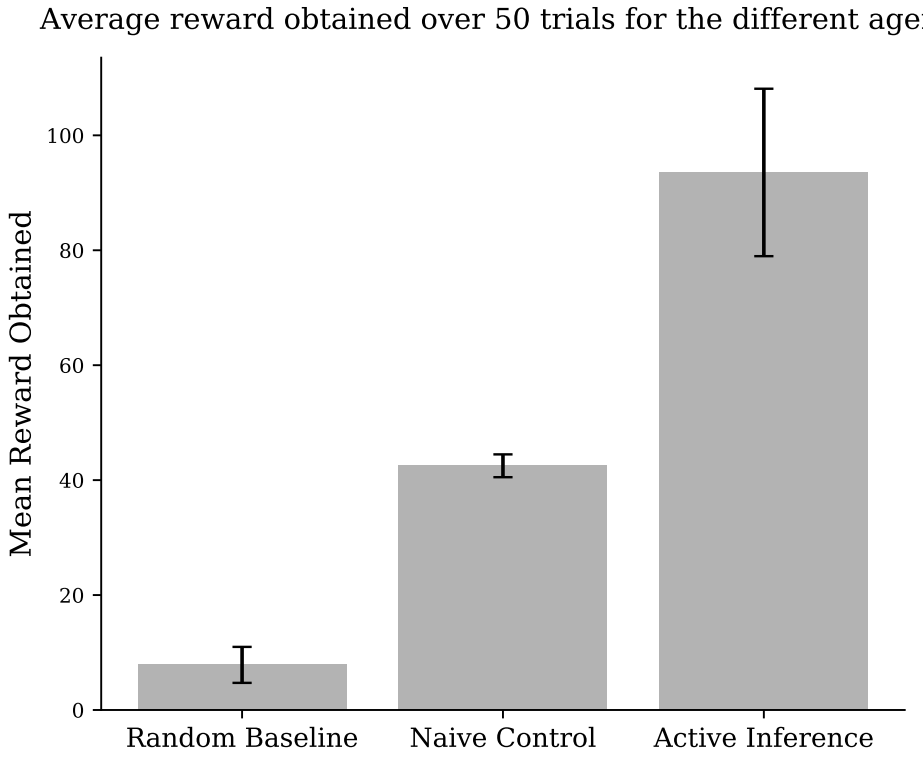


Figure 1: The average reward obtained over 50 trials for the active inference and baseline agents. The error bars are the standard deviations of the reward obtained.

The average reward for the random agent was 7.9, for the naive control agent was 42.5, and for the active inference agent was 93.5. The active inference agent significantly outperformed the naive control agent. The difference in reward between the two was statistically significant under an independent-samples, two-tailed t-test ($t = 24.3; p \approx 0$). The active inference agent also significantly outperformed the random agent ($t = 65.4, p \approx 0$).

Although the active inference agent significantly outperformed the other two methods, it suffers from high variance in its performance and a large sensitivity to its random initialization. Often, with a poor initialization the agent failed to perform above the naive control baseline. The results we report here are obtained from running the active inference agent for 100 trials and picking the best initialization parameters, then running it for the comparison 50 trials with those initialization parameters. There is still significant variance in its performance on these comparison trials due to the random start-points of the cart-pole environment, but it is significantly reduced. This demonstrates that with good initial parameter choices the active inference agent is able to perform significantly above the other baseline methods.

To get a better understanding of the variance of the performance of the agents, we plot the rewards obtained on each of the 50 trials as Figure 2 below.

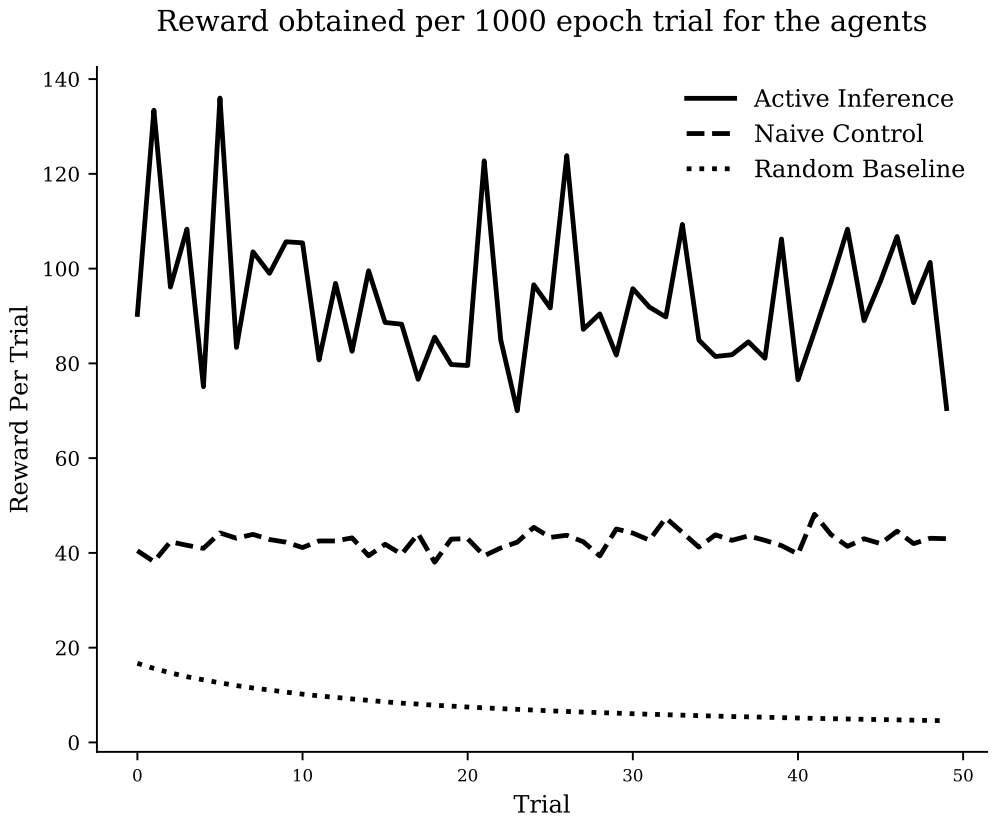
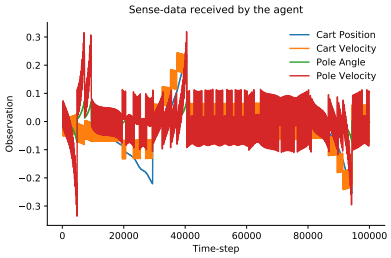


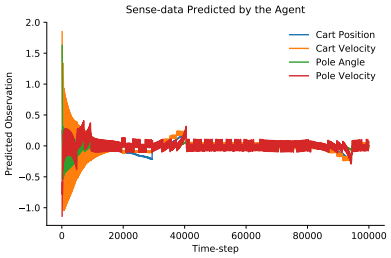
Figure 2: A plot of the rewards obtained by the active inference and baseline agents across 50 trials

The naive control and random baselines show very little variance in performance between trials, while the active inference agent has more variance. However, with a good initialization, the active inference agent always performs significantly better than the other two agents.

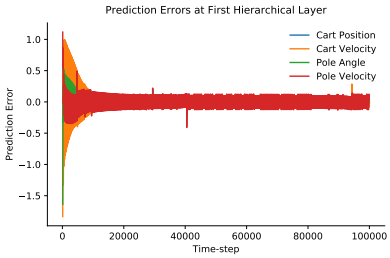
A series of graphs showing the internal states of the agent over a representative run are shown below. Look especially at the prediction errors, which very rapidly decrease to a small baseline rate from the start. The second and third hierarchical level of prediction errors are decreased almost to zero while the first layer shows some baseline prediction errors that are due to effectively random fluctuations in the status of the cart that the model is not able to account for completely. These results clealy show that the agent is able to learn to grapple with the task successfully and can reduce the prediction errors at all hierarchical levels of the generative model to a small baseline rate. Additionally, several animations of the agent acting in the cart-pole environment can be found in the supplementary material.



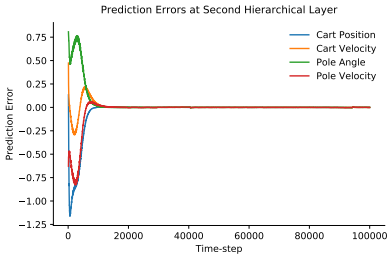
(a) The sense-data given to the agent



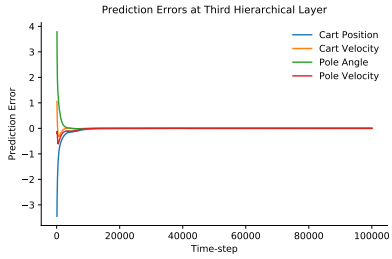
(b) Agent's predictions of sensory input.



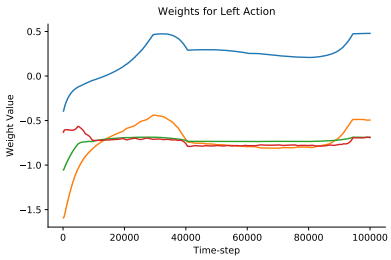
(c) The prediction error at the first hierarchical layer.



(d) The prediction error at the second hierarchical layer



(e) The prediction error at the third hierarchical layer



(f) The weights of the agent's left action.

Figure 3: The training graphs of the active inference cart-pole model. The agent significantly outperforms the naive control and the random policy agent and achieves high rewards on the task. It is not perfect, however, and some significant prediction errors remain, especially at the lowest level of the hierarchy.

4 Discussion

The key contribution of this paper is to take a step towards scaling up active inference and predictive processing. We show empirically that these methods can be extended beyond toy-tasks to more difficult and complex reinforcement learning baselines. We provide a proof-of-concept active inference agent, equipped with a hierarchical predictive coding model that selects actions according to active inference which outperforms other baseline agents. We demonstrate empirically that hierarchical predictive coding models can learn regularities in complex, rapidly changing sense-data and be used to successfully drive action selection in a baseline reinforcement learning environment. Unlike some implementations (Friston et al., 2009), the agent was never shown any optimal trajectories or given any guidance. Instead, the agent had to learn to perceive and act from scratch purely from the prediction errors encountered as it grappled with the task.

To our knowledge, there are few other demonstrations of predictive processing and active inference on non-toy tasks. Cullen et al. (2018) train an active inference agent on the OpenAI environment VizDoom, but they artificially pre-process the input to reduce it to only six possible states. Buckley et al. (2017) and Baltieri and Buckley (2017) have implemented minimal active inference agents either as a tutorial case-study or to simulate the kind of simple generative models they assume that plants involved in phototaxis might possess. Perhaps the most ambitious paper is Ueltzhöffer (2018) which uses deep recurrent neural networks and evolution strategies to parametrize the generative model to solve the mountain car problem. This work is interesting and valuable, but our contribution is unique in that it implements an agent with a hierarchical predictive processing generative model. Our proof-of-concept agent, therefore is constructed entirely out of biologically plausible predictive processing models and components. Additionally, their evolution strategies learning method required an immense use of the environment to obtain low-variance estimates of the empirical gradients with respect to their model parameters – 10^4 environmental interactions per learning step for 30,000 learning steps. In our

method, by contrast, the agent only ever received 1000 environmental inputs per trial and,looking at the graphs of its prediction error, required only a small fraction of that to attain its maximum performance.

The sample efficiency of our agent was noteworthy. From inspecting the graphs of the prediction error in Figure 3, it generally only required up to 200 environmental interactions (often fewer than 5 episodes) to obtain its minimum prediction error. This is impressive even compared to state-of-the-art model-free reinforcement learning approaches, which generally need substantial quantities of environmental interactions to learn robust policies. Although some of the apparent sample efficiency may be a result of the limited capability of our linear predictive coding model compared to deep neural networks, it is likely that biologically plausible predictive coding approaches may be intrinsically more sample efficient, and as such they may resemble biological intelligence which can often learn policies and models of new environments with relatively few interactions compared to model-free reinforcement learning agents.

One drawback of our model is that it proved highly sensitive to its initialization and other hyperparameters such as the learning rate. The learning rate of 0.0005 was used after comparing it with several other candidate learning rates. The search was by no means exhaustive but several other potential learning rates, such as 0.1, 0.05,0.01, 0.001, 0.0001 and so on were considered. At high learning rates, the values of the representation units, weights, and prediction errors typically rapidly diverged. At low learning rates the model generally failed to learn at all, likely due to the statistics of the input changing faster (due to the pole repeatedly toppling) faster than the model could learn them. The input observations were normalized to lie within the range [0,1] which helped prevent exploding weights and prediction errors. The sensitivity of the model to initial settings should not necessarily be taken to be indicative of a lack of potential and scalability of hierarchical predictive coding and active inference. We note that deep neural network also prove to be highly sensitive to initialization (Mishkin and Matas, 2015), and hyperparameter choices (Sutskever et al., 2013), and a large amount of work has gone into both optimizing their initialization (Mishkin and Matas, 2015; Hendrycks and Gimpel, 2016; Glorot and Bengio, 2010) and producing adaptive learning rate schemes (Zeiler, 2012; Kingma and Ba, 2014) and normalization schemes (Ioffe and Szegedy, 2015; Ba et al., 2016) for aiding convergence and preventing vanishing or exploding gradients.

Our results are strong evidence that hierarchical predictive coding and active inference models of perception and action can be scaled up beyond the relatively small-scale tasks they are currently used in to achieve good performance on reinforcement learning baseline tasks. We show that hierarchical predictive coding models can be used to successfully learn to predict a rapidly changing environment, and that this learning can take place online while the agent grapples with the task. Moreover, we show that action selection according to active inference, when combined with the hierarchical predictive coding perceptual model, is able to infer policies that perform well on the cart-pole task and significantly outperform the other baseline agents. Furthermore, the model presented in this paper is ultimately a proof-of-concept and has not been extensively optimized. There are many potential avenues in which to seek improvements in performance. These include adding nonlinear activation functions to the model, removing the truncation of predictions at one step into the future, increasing the number of layers or neurons, and the addition of more complex action models than a simple linear weighted combination added to the cause-units. There is still much work to do, however, before predictive processing can be scaled to the kinds of tasks that reinforcement learning, especially state-of-the-art deep reinforcement learning, algorithms can solve.

5 Conclusion

We have presented a proof-of-concept active inference agent possessing a hierarchical predictive coding network for its generative model, and we have shown that this agent is able to successfully learn to navigate a relatively complex reinforcement learning baseline task. To our knowledge this is the first implemented active inference model available to attempt a significant non-toy task beyond the mountain-car and the only active inference model to implement hierarchical predictive coding for its perceptual model, thus implementing an "end-to-end" predictive processing solution. We have shown empirically that predictive processing approaches are able to successfully scale up to more challenging tasks and

that the way appears open for a more dramatic scaling of predictive processing and active inference to a wide array of tasks which currently use reinforcement learning. We have provided suggestions and ideas for the improvement of current predictive processing and active inference models. Additionally, we have provided an in-depth introduction to the key ideas of the free-energy principle, hierarchical predictive coding, and active inference, with the aim of making these ideas less esoteric and intimidating to the uninitiated and stimulating further research into this promising area.

6 Acknowledgements

I would like to thank Mycah Banks and Richard Shillcock for helpful comments and suggestions on this manuscript.

7 Conflict of Interest

We have no conflicts of interest to declare.

A Derivation of Entropy of Gaussian Distribution

The entropy of a Gaussian distribution – here represented as $N(\theta; \mu, \sigma)$ – can be derived as follows. First we write out the formula for the entropy and substitute in the form of the Gaussian probability density:

$$H(\theta) = \int N(\theta; \mu, \Sigma) log(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}})$$

We then use the properties of logs to split the formula, and take expectations of the resulting expression:

$$H(\theta) = \frac{1}{2} E[log(2\pi\sigma^2)] + E[\frac{(x-\mu)^2}{2\sigma^2}]$$

The first term has no θs in it, so the expectation vanishes. In the second term we expand out the quadratic and use the linearity of expectation to distribute the expectation across the terms:

$$H(\theta) = \frac{1}{2} log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(E[x^2] - 2E[x]\mu + E[\mu^2]$$

The μ^2 term has no dependence on x, so the expectation vanishes. Using the fact that $E[x] = \mu$, we can see that the second term in the quadratic becomes $2E[x]\mu = 2\mu\mu = 2\mu^2$. In the first term we use the fact that the variance is defined as: $\sigma^2 = E[x^2] - E[x]^2$, so, rearranging for $E[x^2]$, we get: $E[x^2] = \sigma^2 + E[x]^2 = \sigma^2 + \mu^2$. The quadratic thus simplifies to:

$$H(\theta) = \frac{1}{2} log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(\sigma^2 + \mu^2 - 2\mu^2 + \mu^2)$$

The μs cancel leaving:

$$H(\theta) = \frac{1}{2} log(2\pi\sigma^2) + \frac{\sigma^2}{2\sigma^2} \tag{23}$$

$$= \frac{1}{2} log(2\pi\sigma^2) + \frac{1}{2} \tag{24}$$

This result has no dependence on the variational parameter θ and so is a constant term in the free energy with respect to those parameters. Thus, when optimizing the free energy, it can be ignored.

B Derivation of the Result of the Laplace Approximation on the Free Energy

We have the following expression for the free energy:

$$\int N(\theta; \mu, \Sigma) log(p(o, \theta))$$

With the Laplace assumption, we assume that the distribution of Q is tightly peaked around the mean μ. Since this is the case, the integral over Q will have non-negligible contributions only close to the mean. We can approximate this integral, therefore, with a Taylor Expansion around the mean μ of the variational density. This gives:

$$\int N(\theta; \mu, \Sigma) log(p(o, \theta)) = log(p(\mu, \theta)) + E[\frac{dF}{d\theta}(\theta - \mu) + E[\frac{d^2F}{d\theta^2}](\theta - \mu)^2]$$

The second term in this equation is 0 since, by the linearity of the expectation, it can be distributed through the brackets and $E[\theta] = \mu$. Next we can recognize that in the third term, $E[(\theta - \mu)^2]$ is the second moment of Q, and thus the variance. This allows us to write:

$$\int N(\theta; \mu, \Sigma) log(p(o, \theta)) = log(p(\mu, \theta)) + \frac{d^2F}{d\theta^2}\Sigma$$

Taking the derivative of Σ with respect to F, and setting it to zero, we can see that the optimal value for Σ is:

$$\Sigma^* = \frac{d^2F}{d\theta^2}^{-1}$$

Since the optimal Σ is known analytically, and has no dependence on the variational mean, it is irrelevant to the optimization and can be ignored.

References

Ashby, W. R. and Goldstein, J. (2011). Variety, constraint, and the law of requisite variety. *Emergence: Complexity and Organization*, 13(1/2):190.

Attias, H. (2003). Planning by probabilistic inference. In *AISTATS*. Citeseer.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Baltieri, M. and Buckley, C. L. (2017). An active inference implementation of phototaxis. In *Proceedings of the European Conference on Artificial Life 14*, volume 14, pages 36–43. MIT Press.

Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., and Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711.

Beal, M. J. et al. (2003). *Variational algorithms for approximate Bayesian inference*. university of London London.

Blanchard, P. and Brüning, E. (2012). *Variational methods in mathematical physics: a unified approach*. Springer Science & Business Media.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211.

Botvinick, M. and Toussaint, M. (2012). Planning as inference. *Trends in cognitive sciences*, 16(10):485–488.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Brown, H., Friston, K. J., and Bestmann, S. (2011). Active inference, attention, and motor preparation. *Frontiers in psychology*, 2:218.

Buckley, C. L., Kim, C. S., McGregor, S., and Seth, A. K. (2017). The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*, 81:55–79.

Clark, A. (1999). An embodied cognitive science? *Trends in cognitive sciences*, 3(9):345–351.

Clark, A. (2017). Embodied, situated, and distributed cognition. *A companion to cognitive science*, pages 506–517.

Cullen, M., Davey, B., Friston, K. J., and Moran, R. J. (2018). Active inference in openai gym: A paradigm for computational investigations into psychiatric illness. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, 3(9):809–818.

Friston, K. (2003). Learning and inference in the brain. *Neural Networks*, 16(9):1325–1352.

Friston, K. (2005). A theory of cortical responses. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1456):815–836.

Friston, K. (2008). Hierarchical models in the brain. *PLoS computational biology*, 4(11):e1000211.

Friston, K. (2009). The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences*, 13(7):293–301.

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127.

Friston, K. (2012). The history of the future of the bayesian brain. *NeuroImage*, 62(2):1230–1233.

Friston, K. (2013). Life as we know it. *Journal of the Royal Society Interface*, 10(86):20130475.

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., and Pezzulo, G. (2017). Active inference: a process theory. *Neural Computation*, 29(1):1–49.

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., Pezzulo, G., et al. (2016). Active inference and learning. *Neuroscience & Biobehavioral Reviews*, 68:862–879.

Friston, K., Kilner, J., and Harrison, L. (2006). A free energy principle for the brain. *Journal of Physiology-Paris*, 100(1-3):70–87.

Friston, K., Levin, M., Sengupta, B., and Pezzulo, G. (2015a). Knowing one's place: a free-energy approach to pattern regulation. *Journal of the Royal Society Interface*, 12(105):20141383.

Friston, K., Mattout, J., and Kilner, J. (2011). Action understanding and active inference. *Biological cybernetics*, 104(1-2):137–160.

Friston, K., Mattout, J., Trujillo-Barreto, N., Ashburner, J., and Penny, W. (2007). Variational free energy and the laplace approximation. *Neuroimage*, 34(1):220–234.

Friston, K., Rigoli, F., Ognibene, D., Mathys, C., Fitzgerald, T., and Pezzulo, G. (2015b). Active inference and epistemic value. *Cognitive neuroscience*, 6(4):187–214.

Friston, K., Samothrakis, S., and Montague, R. (2012). Active inference and agency: optimal control without cost functions. *Biological cybernetics*, 106(8-9):523–541.

Friston, K., Schwartenbeck, P., FitzGerald, T., Moutoussis, M., Behrens, T., and Dolan, R. J. (2013). The anatomy of choice: active inference and agency. *Frontiers in human neuroscience*, 7:598.

Friston, K., Stephan, K., Li, B., and Daunizeau, J. (2010a). Generalised filtering. *Mathematical Problems in Engineering*, 2010.

Friston, K. J., Daunizeau, J., and Kiebel, S. J. (2009). Reinforcement learning or active inference? *PloS one*, 4(7):e6421.

Friston, K. J., Daunizeau, J., Kilner, J., and Kiebel, S. J. (2010b). Action and behavior: a free-energy formulation. *Biological cybernetics*, 102(3):227–260.

Ghahramani, Z. and Beal, M. J. (2001). Propagation algorithms for variational bayesian learning. In *Advances in neural information processing systems*, pages 507–513.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Gopnik, A. and Tenenbaum, J. B. (2007). Bayesian networks, bayesian learning and cognitive development. *Developmental science*, 10(3):281–287.

Hendrycks, D. and Gimpel, K. (2016). Generalizing and improving weight initialization. *arXiv preprint arXiv:1607.02488*.

Hernández-Lobato, J. M., Li, Y., Rowland, M., Hernández-Lobato, D., Bui, T., and Turner, R. E. (2016). Black-box α-divergence minimization.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Kanai, R., Komura, Y., Shipp, S., and Friston, K. (2015). Cerebral hierarchies: predictive processing, precision and the pulvinar. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1668):20140169.

Karl, F. (2012). A free energy principle for biological systems. *Entropy*, 14(11):2100–2121.

Kiebel, S. J. and Friston, K. J. (2011). Free energy and dendritic self-organization. *Frontiers in systems neuroscience*, 5:80.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kirchhoff, M., Parr, T., Palacios, E., Friston, K., and Kiverstein, J. (2018). The markov blankets of life: autonomy, active inference and the free energy principle. *Journal of The royal society interface*, 15(138):20170792.

Knill, D. C. and Pouget, A. (2004). The bayesian brain: the role of uncertainty in neural coding and computation. *TRENDS in Neurosciences*, 27(12):712–719.

Lawson, R. P., Rees, G., and Friston, K. J. (2014). An aberrant precision account of autism. *Frontiers in human neuroscience*, 8:302.

Li, Y. and Turner, R. E. (2016). Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081.

Limongi, R., Bohaterewicz, B., Nowicka, M., Plewka, A., and Friston, K. J. (2018). Knowing when to stop: Aberrant precision and evidence accumulation in schizophrenia. *Schizophrenia research*.

Mishkin, D. and Matas, J. (2015). All you need is a good init. *arXiv preprint arXiv:1511.06422*.

Mittag, E. and Evans, D. J. (2003). Time-dependent fluctuation theorem. *Physical Review E*, 67(2):026113.

Petersen, K. B., Pedersen, M. S., et al. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15):510.

Pezzulo, G. (2012). An active inference view of cognitive control. *Frontiers in Psychology*, 3:478.

Ramstead, M. J., Constant, A., Badcock, P. B., and Friston, K. J. (2019). Variational ecology and the physics of sentient systems. *Physics of life reviews*.

Ramstead, M. J. D., Badcock, P. B., and Friston, K. J. (2018). Answering schrödinger's question: a free-energy formulation. *Physics of life reviews*, 24:1–16.

Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79.

Rawlik, K., Toussaint, M., and Vijayakumar, S. (2010). Approximate inference and stochastic optimal control. *arXiv preprint arXiv:1009.3958*.

Rawlik, K., Toussaint, M., and Vijayakumar, S. (2013). On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.

Rowlands, M. (2009). Enactivism and the extended mind. *Topoi*, 28(1):53–62.

Seth, A. K. (2014). *The cybernetic Bayesian brain*. Open MIND. Frankfurt am Main: MIND Group.

Sutskever, I., Martens, J., Dahl, G. E., and Hinton, G. E. (2013). On the importance of initialization and momentum in deep learning. *ICML (3)*, 28(1139-1147):5.

Thompson, E. and Varela, F. J. (2001). Radical embodiment: neural dynamics and consciousness. *Trends in cognitive sciences*, 5(10):418–425.

Ueltzhöffer, K. (2018). Deep active inference. *Biological Cybernetics*, 112(6):547–573.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.