

# Predictive Coding Notes

Adam Kosiorek

June 22, 2017

## 1 Predictive Coding

Modern sequential predictive models tend to update its hidden state at every time-step. It can be argued, however, that if a model is able to predict the world perfectly, it should not update its state. On the contrary, if a perfect prediction is available, the model should also be capable of evolving its hidden state so as to reflect the change of the state of the world w. r. t. the prediction. Predictive coding formalises this behaviour by using only prediction errors to update the world.

The idea dates back to the Kalman filter and beyond . Recent advances in neural networks make it possible to frame the predictive coding problem as an auto-regressive neural network. Indeed, it has been addressed in a number of papers . These models use the mean-squared error between the prediction and the corresponding input to update their hidden state. It leads to an ill-posed problem since there are many futures possible. The prediction, which is a maximum-likelihood estimate of what might happen, is only a single instantiation thereof. It would be theoretically more sound to normalise prediction errors by the covariance matrix of errors, an approach adopted by Kalman filtering. Friston argues this is also the approach taken by the brain, and it explains a lot of phenomena happening there. He suggests that the computational architecture of the brain forms a hierarchical system, where each layer constantly predicts the output of the lower levels of hierarchy in a fully Bayesian fashion. It gives rise to surprise, which is the negative log-likelihood of the inputs under the predictive distribution of the model. Friston argues that surprise leads to error normalisation w. r. t. prediction uncertainty, which can be interpreted as attention.

Free-energy minimisation problems can be often written as variational inference, which is also true in this case. In what follows, we specify the model as a variational autoencoder, show its computational graph and derive the variational lower bound.

Add reference(s).

Add reference(s).

- cortexnet
- prednet
- some others

Add reference(s).

- Friston

## 2 Variational Distribution

Let  $\mathbf{x}_{1:T}$  be a timeseries of observations e.g., image sequence or a series of joint angles. The probability distribution over input sequences  $p(\mathbf{x}_{1:T})$  can be expressed by the product law as  $p(\mathbf{x}_{1:T}) = \prod_{t=1}^T p(\mathbf{x}_t | \mathbf{x}_{1:t-1})$  and is in general intractable. We introduce a hierarchy of latent variables  $\mathbf{y}_{1:T}$ , where  $\mathbf{y}_t = \{\mathbf{z}_{1:T}^l\}_{l=1}^L$ , such that the joint probability distribution  $p_\theta(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = p_\theta(\mathbf{x}_{1:T} | \mathbf{y}_{1:T}) p_\theta(\mathbf{y}_{1:T})$  lives in the space of parametric probability distributions parametrised by parameters  $\theta$  and is tractable. We further assume the following factorisation:

**prior** The prior distribution is a dynamical model of the latent space. It specifies the probability distribution over the initial states  $\mathbf{y}_0$  and the transition probabilities between states. It may or may not be stationary.

$$p_\theta(\mathbf{y}_{0:T-1}) = p_\theta(\mathbf{y}_0) \prod_{t=1}^{T-1} p_\theta(\mathbf{y}_t | \mathbf{y}_{t-1}) \quad (1)$$

**generating distribution** The generating distribution decodes the samples from the latent space back into the observation space. It uses the learnt dynamics model internally to propagate the state forward.

$$\begin{aligned} p_\theta(\mathbf{x}_{1:T} | \mathbf{y}_{1:T}) &= \int p_\theta(\mathbf{y}_0) \prod_{t=0}^{T-1} p_\theta(\mathbf{x}_{t+1} | \mathbf{y}_t) d\mathbf{y}_0 \\ &= \int p_\theta(\mathbf{y}_0) \prod_{t=0}^{T-1} \int p_\theta(\mathbf{x}_{t+1} | \hat{\mathbf{y}}_{t+1}) p_\theta(\hat{\mathbf{y}}_{t+1} | \mathbf{y}_t) d\hat{\mathbf{y}}_{t+1} d\mathbf{y}_0 \\ &= p_\theta(\mathbf{x}_{1:T} | \mathbf{y}_{1:T-1}) \end{aligned} \quad (2)$$

**posterior distribution** The posterior distribution is intractable, although its general form reads as follows:

$$p(\mathbf{y}_{1:T-1} | \mathbf{x}_{1:T-1}) = \int \prod_{t=1}^{T-1} p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{t-1}) p_\theta(\mathbf{y}_0) d\mathbf{y}_0 \quad (3)$$

**approximate posterior** We approximate the posterior with a tractable parametric distribution  $q_\phi$  by assuming that the hidden state  $\mathbf{y}_t$  is Markovian.

$$q_\phi(\mathbf{y}_{1:T-1} | \mathbf{x}_{1:T-1}, \mathbf{y}_0) = \prod_{t=1}^{T-1} q_\phi(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{t-1}) \quad (4)$$

Moreover, since we learn the latent dynamics model, the generating model predicts the input at the next time-step as opposed to standard VAE practice, where it merely reconstructs the input from the current time-step.

### 3 ELBO

It's looking pretty straightforward, but to be derived...

### 4 Architecture

We need  $L$  encoders,  $L$  decoders,  $L$  dynamic models and  $L - 1$  combinators. Each of the components could in principle be either stochastic or deterministic. To contain the model within the VAE framework, we take stochastic encoders and decoders, stochastic dynamic models, which we use as latent space priors, and deterministic combinators.

**encoder** Given the previous latent state and a prediction error, it computes the parameters of the approximate posterior  $q_\phi(\mathbf{z}_t^l \mid \mathbf{z}_{t-1}^l, \Delta \mathbf{z}_t^l)$ .

**dynamic model** Computes the probability distribution over the new state estimate given a previous state estimate, it's part of the prior as  $p_\theta(\mathbf{z}_{t+1} \mid \mathbf{z}_{t-1})$ .

**combinator** State is predicted based on more abstract representation. This module refines the prediction by using lower-level information available in the previous state estimate. It is a part of the generating model.

**decoder** The final part of the generating model, transforms the refined state estimate computed by the combinator and computes the probability distribution over the lower level state estimate  $p_\theta(\mathbf{z}_{t+1}^l \mid \mathbf{z}_t^{l+1}, \hat{\mathbf{z}}_{t+1}^{l+1})$ .

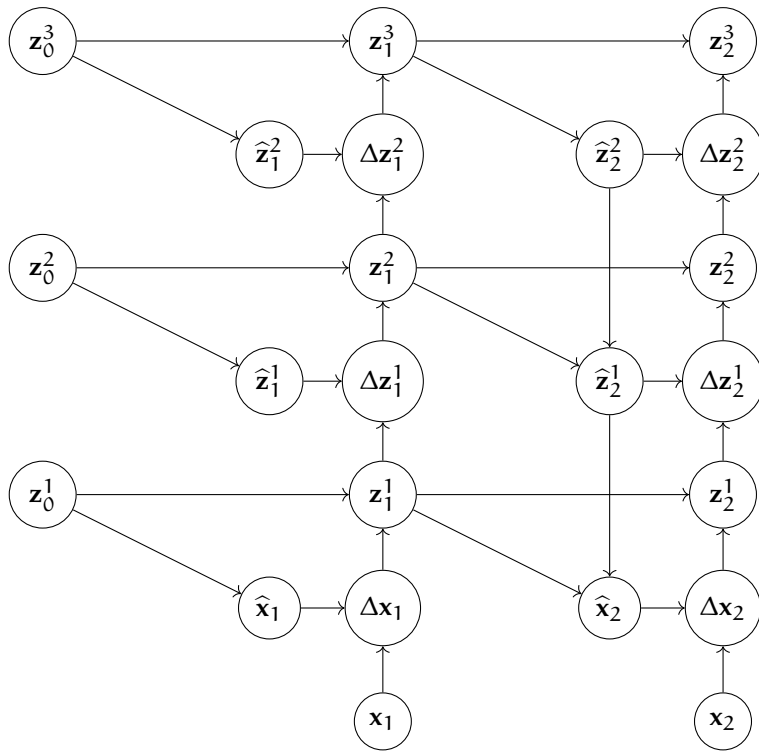


Figure 1: Computation graph of the predictive coding framework.