Predictive Coding, Variational Autoencoders, and Biological Connections

Anonymous Author(s)
Affiliation
Address
email

Abstract

Predictive coding, within theoretical neuroscience, and variational autoencoders, within machine learning, both involve latent Gaussian models and variational inference. We outline connections and contrasts between these areas, then highlight two frontiers for cross-pollination: backpropagation and normalizing flows.

1 Introduction

Predictive coding \[35, 12\] and variational autoencoders (VAEs) \[21, 37\] both frame perception as a generative process, modeling observations, \(x\), using latent variables, \(z\), through a probabilistic model, \(p_\theta(x, z) = p_\theta(x|z)p_\theta(z)\). Both areas also use variational inference, introducing an approximate posterior, \(q(z|x)\), to infer \(z\) and learn the model parameters, \(\theta\). These similarities are the result of a common origin, with Mumford \[33\], Dayan et al. \[7\], and others formalizing earlier ideas \[43\]. Since their inception, these areas have developed their own techniques largely independently. We explore these relationships (see also \[42, 26\]) and highlight two opportunities for the transfer of ideas.

2 Background

2.1 Predictive Coding

Predictive coding is a theory of thalamocortical function, formulated with probabilistic models. Under this theory, the brain constructs a generative model of sensory inputs. Top-down projections convey predictions of lower-level activity, while bottom-up projections convert the prediction error at each level into an updated perceptual state estimate. Such models are often formulated with Gaussian distributions, with analytical non-linear (e.g. polynomial) functions parameterizing the generative mappings \[35, 12\]. Variational inference is performed using gradient-based optimization on the distribution parameters of \(q(z|x)\), where the gradient is a linear combination of (prediction) errors for each variable. Learning can also be performed using gradient-based optimization.

2.2 Variational Autoencoders

VAEs are a class of Bayesian machine learning models, combining probabilistic models with deep neural networks. They consist of an encoder network (with parameters \(\phi\), parameterizing \(q_\phi(z|x)\), and a decoder network, parameterizing \(p_\theta(x|z)\). Thus, rather than performing gradient-based inference, VAEs amortize inference optimization with a learned network \[14\], improving the computational efficiency. The gradients for the inference network parameters are obtained by reparameterizing stochastic samples, \(z \sim q(z|x)\) \[21, 37\], however, other gradient estimators are also possible \[45, 52\]. Learning the parameters, \(\theta\) and \(\phi\), of the deep networks is performed using gradient-based optimization, with gradients obtained using backpropagation \[39\].

Submitted to the Real Neurons & Hidden Units Workshop @ NeurIPS 2019. Do not distribute.
3 Connections & Contrasts

Model  Predictive coding tends to use analytical functions to parameterize conditional mappings [12], whereas VAEs use deep networks. If these functions are analogous to neural projections, then deep networks would correspond to dendrites. In sequential models, predictive coding models temporal changes explicitly [13], whereas VAEs typically use recurrent networks, e.g. [6]. Predictive coding implements prediction covariance matrices with lateral connections, which are optimized as a form of “attention” [10]. VAEs have a similar mechanism, called normalizing flows (NFs) [36] (Section 4).

Inference  Predictive coding and VAEs both use variational inference, often setting $q(z|x)$ as Gaussian. Predictive coding uses errors ($\epsilon_x$ and $\epsilon_z$) to perform gradient-based inference (Fig. 1), whereas VAEs use amortized inference, encoding $x$. However, amortized inference can also be performed by iteratively encoding errors [29, 28]. This resembles predictive coding, but learns the inference model weights.

4 Frontiers

Backpropagation & Learning  The biological plausibility of backpropagation [39] remains an open question. Backprop requires non-local information, whereas biology relies on local learning rules [17, 30, 5]. Yet, biologically-plausible formulations of backprop have been proposed [40, 23, 47, 18, 25], with recent formulations in generative models [4, 24, 2] and predictive coding [44]. In these set-ups, errors at each latent level provide a local signal to drive learning of inference and generative weights. However, these formulations place biological neurons in correspondence with artificial neurons, while we have drawn this correspondence with deep networks. This suggests a larger role for 1) non-linear dendritic computation [31] and segmented dendrites [4, 16], as well as 2) backpropagating action potentials within neurons (Fig. 2a, 46, 22). Thus, backprop may be analogous to learning within particular neuron types, while signaling across synapses may be handled via alternative techniques [45, 37, 21].

Normalizing Flows  We often consider parametric distributions, as they enable efficient evaluation and sampling. However, simple distributions are limiting. Normalizing flows (NFs) [38, 8, 36] provide more complexity while maintaining tractable evaluation and sampling. They consist of a tractable base distribution and one or more invertible transforms. With the base distribution as $p_\theta(u)$ and the transforms as $v = f_\theta(u)$, the probability $p_\theta(v)$ is given by the change of variables formula:

$$p_\theta(v) = p_\theta(u) \left| \det \left( \frac{dv}{du} \right) \right|^{-1},$$

where $\det(\cdot)$ denotes the determinant and $|\cdot|$ denotes absolute value. The determinant term is the local scaling of space when moving from $u$ to $v$. A popular family of transforms is that of autoregressive affine transforms [20, 34]. One example is given by

$$v_i = \alpha_\theta(v_{<i}) + \beta_\theta(v_{<i}) \cdot u_i,$$

where $v_i$ is the $i$th dimension of $v$ and $\alpha_\theta$ and $\beta_\theta$ are functions. The inverse transform (Fig. 2b) is

$$u_i = (v_i - \alpha_\theta(v_{<i}))/\beta_\theta(v_{<i}),$$

a normalization (whitening) operation. Thus, we can sample from complex distributions by starting with simple distributions and applying local affine transforms. Conversely, we can evaluate inputs from complex distributions by applying normalization transforms, then evaluating in a simpler space.

Local lateral inhibition is ubiquitous in neural systems, thought to implement normalization [11]. These circuits give rise to decorrelation in retina [15], LGN [9], and cortex [19], as well as correlation in central pattern generators [27]. NFs may describe these circuits and help justify design choices in predictive coding. Evaluating flow-based conditional likelihoods [1] involves whitening the observations, as in [35]. When applied across time, NFs can resemble temporal derivatives [13]. Likewise, a prior covariance with lateral weights [12] corresponds to a NF with linear dependencies [20]. NFs have also been explored in the context of action [41], providing correlated motor outputs.
(a) Backpropagation within Neurons

(b) Normalizing Flow via Lateral Inhibition

Figure 2: Frontiers. (a) Backpropagation may be more analogous to backpropagating action potentials within neurons rather than across networks of neurons. (b) Affine normalizing flows can be implemented with lateral inhibitory connections. In the figure, a correlated vector, $v$, is inversely transformed into a decorrelated vector, $u$, simplifying the evaluation of predictions.

References

processing systems, 2016.


