Predictive Coding with Topographic Variational Autoencoders

T. Anderson Keller
UvA-Bosch Delta Lab
University of Amsterdam
t.anderson.keller@gmail.com

Max Welling
UvA-Bosch Delta Lab
University of Amsterdam
welling.max@gmail.com

Abstract

Predictive coding is a model of visual processing which suggests that the brain is a generative model of input, with prediction error serving as a signal for both learning and attention. In this work, we show how the equivariant capsules learned by a Topographic Variational Autoencoder can be extended to fit within the predictive coding framework by treating the slow rolling of capsule activations as the forward prediction operator. We demonstrate quantitatively that such an extension leads to improved sequence modeling compared with both topographic and non-topographic baselines, and that the resulting forward predictions are qualitatively more coherent with the provided partial input transformations.

1. Introduction

Topographic organization in the brain describes the observation that nearby neurons on the cortical surface tend to have more strongly correlated activations than spatially distant neurons. From the simple orientation of lines [22] to the complex semantics of natural language [23], organization of cortical activity is observed for a diversity of stimuli and across a range of species. Given such strong and ubiquitous observations, it seems only natural to wonder about the computational benefits of such organization, and if the machine learning community can take advantage of such design principles to develop better inductive priors for deep neural network architectures.

One inductive prior which has gained popularity in recent years is that of equivariance. At a high level, a representation is said to be equivariant if it transforms in a known predictable manner for a given transformation of the input. A fundamental method for constructing equivariant representations is through structured parameter sharing, constrained by the underlying desired transformation group [10, 41, 17, 18]. The most well known example of an equivariant map is the convolution operation, which is equivariant to translation. One can think of a convolutional layer as a function which shares the same feature extractor parameters over all elements of the translation group, i.e. all spatial locations. Similarly, a model which is equivariant to rotation is one which shares parameters across all rotations. Existing group equivariant neural networks [10] therefore propose to maintain ‘capsules’ of tied-parameters which are correlated by the action of the group. By reducing the number of trainable parameters while simultaneously increasing the information contained in the representation, equivariant neural networks have demonstrated significant improvements to generalization and data efficiency [10, 44, 53].

These sets of transformed weights, which we refer to as ‘equivariant capsules’, are reminiscent of a type of topographic organization observed in the primary visual cortex (V1), namely orientation columns [22]. This insight encouraged the development of the Topographic Variational Autoencoder (TVAE) [29], linking equivariance and topographic organization in a single framework. In the original work, the TVAE was introduced and demonstrated to learn topographic equivariant capsules in an entirely unsupervised manner from observed transformation sequences. Further, the inductive priors of equivariance and ‘slowness’ integrated into the TVAE were demonstrated to be beneficial for modeling sequence transformations, ultimately resulting in higher log-likelihood on held-out data when compared with VAE baselines.

In this work, we propose to extend the TVAE with an
Equivariance is the mathematical notion of symmetry for functions. A function is said to be an equivariant map if the result of transforming the input and then computing the function is the same as first computing the function and then transforming the output. In other words, the function and the transformation commute. Formally, \( f(\tau \rho(x)) = \Gamma \rho (f(x)) \), where \( \tau \) and \( \Gamma \) denote the (potentially different) operators on the domain and co-domain respectively, but are indexed by the same element \( \rho \). The introduction of the Group-convolution [10] and ensuing work [11, 53, 52, 18, 49] allowed for the development of analytically equivariant neural network architectures to a broad range of group transformations, demonstrating measurable benefits in domains such as medical imaging [48, 2] and molecular generation [43]. Recently, more work has begun to explore the possibility of ‘learned’ equivariance guided by the data itself [4, 14, 20]. The TVAE and the extension presented in this paper are another promising step in this direction.

2.2. Topographic Organization

Topographic generative models can be seen as a class of generative models where the latent variables have an underlying topographic organization which determines their correlation structure. As opposed to common generative models such as Independent Component Analysis (ICA) [3, 27] or VAEs [30, 42], the latent variables in a topographic generative model are not assumed to be entirely independent, but instead are more correlated when they are spatially ‘close’ in a predetermined topology. Typically, simple topologies such as 1 or 2 dimensional grids are used, often with circular boundaries to avoid edge effects.

One way a topographic generative model can be described, as in [25], is as a hierarchical generative model where there exist a set of higher level independent ‘variance generating’ variables \( V \) which are combined locally along the topology to generate the variances of the lower level topographic variables \( T \). Formally, for an adjacency matrix \( W \), and an appropriate non-linearity \( \phi \), the variances are computed as \( \sigma = \phi(WV) \). In the second stage, the lower level variables are sampled independently, but with their scale determined by the now topographically organized variable \( \sigma: T \sim \mathcal{N}(0, \sigma^2 I) \). In later work [26], Hyvärinen et al. further showed this framework to be a generalization Independent Subspace Analysis (ISA) [24] and some variants of Slow Feature Analysis (SFA) [45, 51, 46] by careful choice of topography \( W \). The Topographic VAE takes advantage of both this framework and these connections to construct slow-transforming capsules which learn to become equivariant to observed sequence transformations.

3. Predictive Coding with Topographic VAEs

In this section we introduce the generative model underlying the Predictive Coding Topographic VAE (PCTVAE) and highlight the differences with the original model – including making the conditional generative distribution forward predictive, and limiting the temporal coherence window to only include past variables.

3.1. The Forward Predictive Generative Model

In the machine learning literature, one of the most intuitive and common frameworks for unsupervised learning relies on predicting unseen or missing contextual data from a given input. This idea, informally called predictive coding, has led to some of the most well known advances in the field across a range of domains including: natural language processing (word2vec [37], GPT3 [6], Bert [13]), vision (CPC [47], SimCLR [7], GreedyInfoMax [35]), speech (APC [8]), and more [21, 33]. In the theoretical neuroscience literature, predictive coding denotes a framework by which the cortex is a generative model of sensory inputs [16, 40, 19, 9], and has been linked to probabilistic latent variable models such as VAEs [36]. Substantial evidence has been gathered supporting the existence of some form of predictive coding in the brain [1, 12, 15], and numerous computational models have been proposed which replicate empirical observations [40, 34, 28]. Given these computational successes, and the mounting support for such a mechanism underlying biological intelligence, we strive to formalize the relationship between predictive coding and VAEs in this work.
and the prior:
\[ p(x_{i+1}|t_i) = \prod_l p_X(x_{i+1}|t_i)p_T(t_i) \]  
(1)

The prior distribution is assumed to be a Topographic Product of Student’s-t (TPoT) distribution \([50, 38]\), i.e. \( p_T(t_i) = \text{TPoT}(t_i; \nu) \), and we parameterize the conditional distribution with a flexible function approximator:
\[ p_X(x_{i+1}|t_i; \theta) = \text{TPoT}(x_{i+1} | \theta \circ g(t_i)) \]  
(2)

The goal of training is thus to learn the parameters \( \theta \) such that the marginal distribution of the model \( p_{\theta}(x_i) \) matches that of the observed data.

To allow for efficient training, we follow the construction outlined in [29], whereby we construct a TPoT random variable from simpler independent normal random variables \( Z_l \) and \( U_l \) which are amenable to variational inference:
\[ T_l = \frac{Z_l - \mu}{\sqrt{\mathbf{W} U_l^2}} \quad Z_l, U_l \sim \mathcal{N}(0, I) \]  
(3)

where \( \mathbf{W} \) defines the chosen topology, and \( \mu \) is learned.

**Past Temporal Coherence**  
As mentioned in the Section 2.2, the Topographic VAE takes advantage of the generalized framework of topographic generative models to induce structured correlations of activations over time - thereby achieving equivariance. In this work, this is achieved by making \( T_l \) a function of a sequence \( \{U_{l-1}^\delta\}_{\delta=0}^L \), and defining \( \mathbf{W} \) to connect sequentially rolled copies of past \( U_l \):
\[ T_l = \frac{Z_l - \mu}{\sqrt{\mathbf{W} \left[U_l^2; \cdots; U_{l-L}^2\right]}} \]  
(4)

where \( \left[U_l^2; \cdots; U_{l-L}^2\right] \) denotes vertical concatenation of the column vectors \( U_l \), and \( L \) can be seen as the past window size. Then, by careful definition of \( \mathbf{W} \), we can achieve the ‘shifting temporal coherence’, defined in [29], yielding equivariant capsules. Explicitly, \( \mathbf{W} \) is thus given by:
\[ \mathbf{W} \left[U_l^2; \cdots; U_{l-L}^2\right] = \sum_{\delta=0}^{L} \mathbf{W}_\delta \text{Roll}_\delta \left(U_{l-\delta}^2\right) \]  
(5)

where \( \mathbf{W}_\delta \) defines a set of disjoint ‘capsule’ topologies for each time-step, and \( \text{Roll}_\delta(U_{l-\delta}) \) denotes a cyclic permutation of \( \delta \) steps along the capsule dimension (see [29] for exact implementation details).

**3.2. The Predictive Coding VAE**

To train the parameters of the generative model \( \theta \), we use equation 4 to parameterize an approximate posterior for \( t_l \) in terms of a deterministic transformation of approximate posteriors over simpler Gaussian latent variables \( z_l \) and \( u_l \):
\[ q_\phi(z_l|x_l) = \mathcal{N}(z_l; \mu_\phi(x_l), \sigma_\phi(x_l)I) \]  
(6)
\[ q_\gamma(u_l|x_l) = \mathcal{N}(u_l; \mu_\gamma(x_l), \sigma_\gamma(x_l)I) \]  
(7)
\[ t_l = \frac{z_l - \mu}{\sqrt{\mathbf{W} \left[u_l^2; \cdots; u_{l-L}^2\right]}} \]  
(8)

Additionally, to further encourage the capsule \( \text{Roll} \) as the forward prediction operator, we integrate a capsule \( \text{Roll} \) of \( t_l \) by one unit as the first step of the generative model, before decoding \( x_{l+1} \):
\[ p_{\theta}(x_{l+1}|g_\theta(t_i)) = p_{\theta}(x_{l+1}|g_\theta(\text{Roll}_1|t_i)) \]  
(9)

We denote this model the Predictive Coding Topographic VAE (PCTVAE) and present an overview of forward prediction in Figure 1. We optimize the parameters \( \theta, \phi, \gamma \) (and \( \mu \)) through the ELBO, summed over the sequence length \( S \):

\[ \sum_{l=1}^{S} \mathbb{E}_{Q_{\phi, \gamma}(z_l, u_l|x_l)} \left( \log p_{\theta}(x_{l+1}|g_\theta(\text{Roll}_1|t_i)) \right) \]  
(10)

where \( Q_{\phi, \gamma}(z_l, u_l|x_l) = q_{\phi}(z_l|x_l) \prod_{\delta=0}^{L} q_{\gamma}(u_{l-\delta}|x_{l-\delta}) \).

The fundamental differences of this model with the VAE are that this model is trained to maximize the likelihood of future inputs through the Roll operation present in the
ELBO, and that the construction of $t_\delta$ is now only a function of past inputs. As we will demonstrate in the next section, these extensions yield significant improvements to sequence modeling, while simultaneously increasing flexibility by allowing for online training and inference.

4. Experiments

In this section we measure the performance of our model, compared with non-predictive coding baselines, on the transforming color MNIST dataset from [29]. The dataset is composed of MNIST digits [32] sequentially transformed by one of three randomly chosen transformations: spatial rotation, rotation in color (hue) space, or scaling. For each training example, the starting pose (color, angle, scale) is randomly set, and a cyclic sequence of 18 examples is generated according to the chosen transform. The same model architecture as [29] (a 3-layer MLP with ReLU activations) is used for all encoders and decoders of all models presented. For topographic models, the latent space is structured as 18 1-dimensional circular capsules, each of 18 dimensions. Further training details can be found at https://github.com/akandykeller/PCTVAE.

4.1. Forward Prediction Likelihood

To quantitatively measure the ability of the PCTVAE to predictively model sequences, we train the model to maximize Equation 10 with stochastic gradient descent, and measure the likelihood of held-out test sequences, with only partial sequences as input. Explicitly, for both the TVAE and PCTVAE, a window size of 9 observations are provided as input and used to generate a capsule representation $t_0$. The likelihood of the remaining 9 sequence elements is then measured by sequentially rolling the capsule activations forward, and measuring $p_{\theta}(x_{\delta t}|g_0(\text{Roll}_{\delta t}(t_0)))$ for $\delta t \in \{0, ..., 9\}$. The final reported likelihood values are computed by importance sampling with 10 samples. In Table 4.1 we report the average log-likelihood over this forward predicted sequence for both the original TVAE and PCTVAE, in addition to the log-likelihood at $\delta t = 0$ (no forward prediction) with a standard VAE. We see the PCTVAE achieves a significantly lower average negative likelihood than both the TVAE model and a non-topographic VAE, TVAE, and PCTVAE. The PCTVAE achieves the lowest average NLL over the forward predicted sequence while also maintaining low equivariance error. Mean ± std. over 3 random initializations.

<table>
<thead>
<tr>
<th>Model</th>
<th>NLL Avg. Seq.</th>
<th>NLL @ $\delta t = 0$</th>
<th>NLL Avg. Seq.</th>
<th>$\mathcal{E}_{eq}$ Avg. Seq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE</td>
<td>190 ± 1</td>
<td>N/A</td>
<td>13274 ± 0</td>
<td></td>
</tr>
<tr>
<td>TVAE</td>
<td><strong>187 ± 1</strong></td>
<td>452 ± 16</td>
<td>2122 ± 21</td>
<td></td>
</tr>
<tr>
<td>PCTVAE</td>
<td>207 ± 1</td>
<td><strong>232 ± 1</strong></td>
<td>2201 ± 9</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Neg. log-likelihood (NLL in nats) without forward prediction ($\delta t = 0$), NLL averaged over the forward predicted sequence, and equivariance error $\mathcal{E}_{eq}$ for a non-topographic VAE, TVAE, and PCTVAE. The PCTVAE achieves the lowest average NLL over the forward predicted sequence while also maintaining low equivariance error. Mean ± std. over 3 random initializations.

As a qualitative evaluation of the PCTVAE’s sequence modeling capacity, we show forward predicted sequences generated by both models in Figure 2. The top row shows the input sequence with grey images held out, and the lower row shows the forward predicted sequence, generated by sequentially rolling the representation $t_0$ forward, and decoding at each step. As can be seen, the PCTVAE (left) appears to generate sequences which are more coherent with the provided input sequence, while the TVAE (right) is observed to quickly diverge from the true transformation, in agreement with likelihood values of Figure 3.

5. Discussion

In this paper we have proposed an extension of the Topographic VAE to the framework of predictive coding, and have demonstrated an improved ability to model the imme-
References


[31] David Klindt, Lukas Schott, Yash Sharma, Ivan Ústuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton.


6. Acknowledgements

We would like to thank the creators of Weight & Biases [5] and PyTorch [39]. Without these tools our work would not have been possible. We thank the Bosch Center for Artificial Intelligence for funding, and the reviewers for their helpful comments.