Similarity-preserving Neural Networks from GPLVM and Information Theory

Anonymous Author(s)
Affiliation
Address
email

Abstract

This work proposes a way of deriving the structure of plausible canonical microcircuit models, replete with feedforward, lateral, and feedback connections, out of information-theoretic considerations. The resulting circuits show biologically plausible features, such as being trainable online and having local synaptic update rules reminiscent of the Hebbian principle. Our work achieves these goals by rephrasing Gaussian Process Latent Variable Models as a special case of the more recently developed similarity matching framework. One remarkable aspect of the resulting network is the role of lateral interactions in preventing overfitting. Overall, our study emphasizes the importance of recurrent connections in neural networks, both for cognitive tasks in the brain and applications to artificial intelligence.

1 Introduction

The idea that the brain is an information-processing entity and its microcircuits deal with separating signal from noise pervades cognitive science, in general, and systems neuroscience, in particular. It is then natural that, for decades, there have been efforts to use information theory (IT) [1, 2, 3, 4, 5, 6] for explaining psychophysical observations and direct measurements of neural activity. On the other hand, efforts directed toward deriving learning rules for neural networks (NN) based on information-theoretic problems [7, 8, 9, 10, 11] need to be examined for biological plausibility [12, 13] before one hopes to posit such NNs as candidates for one of the brain's canonical microcircuits.

In this work, we take up a concrete and simple problem of compressed representation learning [14] via linear Dimension Reduction (DR) [15]. The problem of constructing biologically plausible NNs from similarity-preserving objective functions, i.e., connecting Gramians of inputs and that of corresponding neural activities, has been addressed recently [16, 17, 18, 19, 20, 21]. We keep many of the attractive aspects of the framework but derive a NN and its learning rules starting from a probabilistic formulation known as Gaussian Process Latent Variable Model (GPLVM)[22]. Using GPLVM, we express the objective function in terms of Gramians but also relate it to the IT criterion that, within some constraints, neural codes are to be maximally predictive of the inputs.

In Section 2, we review the IT criterion and focus on the likelihood of a conditional distribution. In this section, we also touch upon GPLVM, which optimizes a model-averaged version of the same likelihood. We argue that, in the large data limit, both these methods recover the true generative model. In Section 3, we then derive the neural dynamics and the learning rule from a min-max formulation of the GPLVM optimization, with neural implementation discussed in Section 4. Finally, we note and discuss in Section 5 the fact that the learning rule for lateral interaction between neurons is very different from the one obtained in the similarity matching framework, where it fosters diversity of response by inhibitory interaction between neurons. We show that, in the probabilistic model, this lateral interaction between neurons essentially leads to redundancy and feature selection, reducing the potential for overfitting.

Submitted to Information-Theoretic Principles in Cognitive Systems Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

2 Background and related works

In this section, we introduce the necessary background on information theory and its formulation when ones considers the task of compression [23, 24]. We make the connection with GPLVM [22], a popular framework for dimensionality reduction. In particular, we discuss the large data limit in which both these methods recover the true generative model.

2.1 Encoding inputs

Let (X, Y) be two random variables. The mutual information (MI) $I(X;Y)$ could be written as

$$I(X;Y) := \int dxdy\, p(x,y) \ln \frac{p(x,y)}{p_X(x)p_Y(y)} = H(Y) - H(Y|X). \tag{1}$$

where $H(Y) := -\int dy\; p_Y(y) \ln p_Y(y)$ and $H(Y|X) := -\int dxdy\, p(x,y) \ln p_{Y|X}(y|x)$ are respectively the entropy of the variable Y and is its conditional entropy, given X. In this work, we interpret Y as the sensory inputs and X as the brain's internal encoding of that input. Since the input distribution is exogenous, the task of mutual information maximization is the same as the task of minimizing the conditional entropy above, subject to constraints (e.g., dimensionality of X). In other words, given the encoding, we want, on average, to be the least "uncertainty" regarding the input.

Let $\{\mathbf{y}_t\}_{t=1}^T$ be sampled *i*id from $p_Y(\cdot)$ and $\mathbf{x}_t \sim p_{X|Y}(\cdot|\mathbf{y}_t)$, $p_{X|Y}$ being the ideal encoder. Then

$$H(Y|X) = -\frac{1}{T}\mathbb{E}[\sum_{t=1}^{T} \ln p_{Y|X}(\mathbf{y}_t|\mathbf{x}_t)] = -\frac{1}{T}\mathbb{E}[\ln p(\mathbf{Y}|\mathbf{X}, \text{Model})], \tag{2}$$

where $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_T]^\top$ refers to n-dimensional observations and $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_T]^\top$, to the m-dimensional encodings. We make the dependence on the true generative model explicit in this expression $p(\mathbf{Y}|\mathbf{X}, \text{Model})$, in contrast to the model averaged conditional distribution below.

Agakov and collaborators developed an MI-based variational formulation for learning embeddings from a finite-size sample, $\{\mathbf{y}_t\}_{t=1}^T$, in which they maximize the logarithm of the variational conditional likelihood $\sum_{t=1}^T \ln q_{Y|X}(\mathbf{y}_t|\mathbf{x}_t)$ over an encoder $p_{X|Y}$, and a variational decoder $q_{Y|X}$ [23, 24]. We, instead, turn to a Bayesian approach for latent variable discovery, dealing with the likelihood involving only $\{\mathbf{y}_t\}_{t=1}^T$ and $\{\mathbf{x}_t\}_{t=1}^T$, obtained by integrating over potential generative model parameters.

2.2 Generative models and probabilistic linear DR

The generative model for linear DR we consider is the following. We assume that n-dimensional observations $\mathbf{Y}$ are a linear function of m-dimensional latent variables $\mathbf{X}$ [25, 22], i.e.,

$$\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \varepsilon_t, \;\; \text{with} \;\; \varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n) \;\; \text{and} \;\; m \ll n \;\; . \tag{3}$$

The linear map $\mathbf{A} \in \mathbb{R}^{n \times m}$ relates the two sets of variables. Furthermore, in GPLVM [22] a prior is imposed on the model parameters $\mathbf{A}$, unlike for probabilistic PCA [25], as

$$\text{GPLVM} \;\; \Rightarrow \;\; \mathbf{A} \sim \Pi_{d=1}^n \mathcal{N}(0, \alpha^{-1}\mathbf{I}_m) \;\; . \tag{4}$$

Since the conditional distribution $\mathbf{Y}|\mathbf{X}$ is linear in $\mathbf{A}$ and $[\varepsilon_1, \dots, \varepsilon_T]$, we can replace our original proposal of the objective function, Eq.(2), $p(\mathbf{Y}|\mathbf{X}, \text{Model}) := p(\mathbf{Y}|\mathbf{X}, \mathbf{A})$ by taking an expectation over $\mathbf{A}$ with the GPLVM prior, Eq. (4), namely $p(\mathbf{Y}|\mathbf{X}) := \mathbb{E}_A[p(\mathbf{Y}, \mathbf{A}|\mathbf{X})]$. We recall the conditional for samples of each component of $\mathbf{y}$, namely $\mathbf{y}^{(d)} = [y_{d1}, \dots, y_{dT}]$ in GPLVM as

$$p(\mathbf{y}^{(d)}|\mathbf{X}) = \mathcal{N}_T(\mathbf{0}, \alpha^{-1}\mathbf{X}^\top\mathbf{X} + \sigma^2\mathbf{I}), \qquad \forall d \in \{1, \dots, n\} \;\; . \tag{5}$$

The model averaged distribution of $\mathbf{Y}|\mathbf{X}$ for GPLVM, in terms of the Gramians, $\mathbf{X}^\top\mathbf{X}, \mathbf{Y}^\top\mathbf{Y}$, is

$$\mathcal{L}(\mathbf{X}) = \log p(\mathbf{Y}|\mathbf{X}) = -\tfrac{nT}{2}\log(2\pi) - \tfrac{n}{2}\log|\mathbf{X}^\top\mathbf{X} + \sigma^2\mathbf{I}| - \tfrac{1}{2}\,\text{Tr}\left[\left(\mathbf{X}^\top\mathbf{X} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{Y}^\top\mathbf{Y}\right], \tag{6}$$

where $|\cdot|$ denotes the matrix determinant. In the limit of large T, for a given Y, the variational procedure mentioned in Section 2.1, carried out over a limited class of encoders and decoders parameterized by $\mathbf{A}$ would recover the true $\mathbf{A}$ with which the data was generated [24]. In the Bayesian approach leading to the GPLVM objective, given $\mathbf{Y}$, Bernstein-von Mises type theorems [26, 27, 28] state that the posterior distribution of $\mathbf{A}$ will be concentrated around the true $\mathbf{A}$, as $T \to \infty$. Therefore, the averaging over $\mathbf{A}$, for the high probability $\mathbf{X}$, approximates picking the optimal/true $\mathbf{A}$.

3 A Min-max objective function for GPLVM

While the GPLVM objective (6) can be minimized by gradient descent [29], this procedure would not lead to an online algorithm as it requires combining data from different time steps. Instead, following the work of [17], we modify the objective (6) by introducing auxiliary matrix variables, $\mathbf{W}$, $\mathbf{M}$, and $\mathbf{Z}$ allowing for the GPLVM computation using solely instantaneous inputs. Such substitution leads to a min-max optimization problem that is solved by gradient descent/ascent and maps onto an NN with local learning rules presented in Sec. 4.

$$\mathcal{L}(\mathbf{X}) = -\frac{nT}{2}\log 2\pi - \frac{n}{2}\log|\mathbf{X}^\top\mathbf{X} + \sigma^2\mathbf{I}| - \frac{1}{2}\operatorname{Tr}\left[(\mathbf{X}^\top\mathbf{X} + \sigma^2\mathbf{I})^{-1}\mathbf{Y}^\top\mathbf{Y}\right]. \tag{7}$$

We now introduce auxiliary matrix variables $\mathbf{W}$ and $\mathbf{M}$. To do so we first address the term that only depends on $\mathbf{X}$, i.e., the log-determinant term (more details in Appendix A)

$$\log|\sigma^2\mathbf{I} + \mathbf{X}^\top\mathbf{X}| = \max_{\mathbf{M}} \operatorname{Tr}\left[(\sigma^2\mathbf{I} + \mathbf{X}\mathbf{X}^\top)\mathbf{M}\right] - \operatorname{Tr}\left[\log(\mathbf{M})\right] + \text{cst} \ . \tag{8}$$

Now the cross-term involving both $\mathbf{X}$ and $\mathbf{Y}$ using Woodbury matrix identity and by only keeping the term that depend on $\mathbf{X}$ we obtain

$$\operatorname{Tr}\left[(\sigma^2\mathbf{I} + \mathbf{X}^\top\mathbf{X})^{-1}\mathbf{Y}^\top\mathbf{Y}\right] = \min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_F^2 + \sigma^2\|\mathbf{W}\|_F^2 + \text{cst} \ . \tag{9}$$

It is important for the neural interpretation that we introduce another auxiliary variable $\mathbf{Z}$ into Eq. (9) that will account for the projection error as

$$\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_F^2 + \sigma^2\|\mathbf{W}\|_F^2 = \max_{\mathbf{Z}} \min_{\mathbf{W}} (\mathbf{Y} - \mathbf{W}\mathbf{X})^\top\mathbf{Z} + \sigma^2\|\mathbf{W}\|_F^2 - \frac{1}{2}\|\mathbf{Z}\|^2 \ . \tag{10}$$

Now we combine Eq.(8) and Eq.(10) to obtain a min-max objective function for linear GPLVM as

$$\min_{\mathbf{X},\mathbf{W}} \max_{\mathbf{Z},\mathbf{M}} \mathcal{L}(\mathbf{W},\mathbf{M},\mathbf{X},\mathbf{Z}) = \min_{\mathbf{X},\mathbf{W}} \max_{\mathbf{Z},\mathbf{M}} \operatorname{Tr}\Bigg[\frac{1}{\sigma^2}(\mathbf{Y} - \mathbf{W}\mathbf{X})^\top\mathbf{Z} - \frac{1}{2\sigma^2}\mathbf{Z}^\top\mathbf{Z} + \frac{1}{2}\mathbf{X}^\top\mathbf{X} + \frac{T}{2}\mathbf{W}^\top\mathbf{W} + \frac{n}{2}\left(\sigma^2\mathbf{M} + \mathbf{X}^\top\mathbf{M}\mathbf{X} - \log(\mathbf{M})\right)\Bigg] \ . \tag{11}$$

4 Online Algorithm and neural implementation

From the min-max objective function Eq. (11) we first derive an online algorithm and then present its implementation by a biologically inspired neural network.

Gradient optimization. We first optimize Eq. (11) with respect to the "neural variables" $\mathbf{x}_t$ and $\mathbf{z}_t$, for fixed $\mathbf{W}$ and $\mathbf{M}$, by gradient descent/ascent. At each time step, t, we repeat the following gradient descent ascent steps until convergence:

$$-\partial_{\mathbf{x}_t}\mathcal{L} = \frac{1}{\sigma^2}\mathbf{W}^\top\mathbf{z}_t - (\mathbf{I}_m + \frac{n}{T}\mathbf{M})\mathbf{x}_t \quad ; \quad \partial_{\mathbf{z}_t}\mathcal{L} = \frac{1}{\sigma^2}(\mathbf{y}_t - \mathbf{W}\mathbf{x}_t) - \frac{1}{\sigma^2}\mathbf{z}_t \ . \tag{12}$$

Then, for fixed $\mathbf{x}_t$ and $\mathbf{z}_t$, we minimize the objective function over $\mathbf{W}$ and $\mathbf{M}$, take stochastic gradient descent-ascent steps, which yields

$$-\partial_{\mathbf{W}}\mathcal{L} = \frac{1}{\sigma^2 T}\mathbf{z}_t\mathbf{x}_t^\top - \mathbf{W} \quad ; \quad \partial_{\mathbf{M}}\mathcal{L} = \frac{n}{2}\left(\sigma^2\mathbf{I}_m + \frac{1}{T}\mathbf{x}_t\mathbf{x}_t^\top\right) - \frac{n}{2}\mathbf{M}^{-1} \ . \tag{13}$$

This yields our online GPLVM algorithm (Algorithm 1) and the neural implementation Fig. 1.

Neural implementation. The algorithm for online GPLVM (Algorithm 1) summarized by the dynamics Eqs. (12) and update rules in Eqs. (13) can be implemented in a neural circuit with schematic shown in Fig. 1. In this circuit, the individual components of the output $\{x_{1,t}, \ldots, x_{m,t}\}$, are represented as the outputs of m neurons, with $\mathbf{M}$ the lateral synaptic connections between them. The auxiliary variable $\mathbf{z}_t$ is represented by the activity of m inter/hidden neurons with $\mathbf{W}$ encoding the connection between $\mathbf{z}_t$ and $\mathbf{x}_t$. In a biological setting, the implied equality of weights of synapses from $\mathbf{z}_t$ to $\mathbf{x}_t$ and the transpose of those from $\mathbf{x}_t$ to $\mathbf{z}_t$ can be guaranteed approximately by application of the same Hebbian learning rule [30]. Another possible way to interpret the model would be as $\mathbf{y}_t$ and $\mathbf{z}_t$ being respectively the somatic and axonal terminal activities of two-compartment unit neurons as in [31]. Update rules requiring computing weight matrix inverse are also present for tasks such as independent component analysis [32] for which local alternatives exist [10].

Algorithm 1: Online algorithm for GPLVM.

input data $\{\mathbf{y}_1, \dots, \mathbf{y}_T\}$; dimension n
output $\{\mathbf{x}_1, \mathbf{z}_1, \dots, \mathbf{x}_T, \mathbf{z}_T\}$; dimension m and n ▷ estimated embedding and error
initialize the matrix $\mathbf{W}$, and positive definite matrix $\mathbf{M}$.
for $t = 1, 2, \dots, T$ **do**
  $\mathbf{z}_t \leftarrow \mathbf{y}_t$; ▷ first error no estimated $\mathbf{x}_t$
  run the following until convergence:
    $\frac{d\mathbf{x}_t(\gamma)}{d\gamma} = \mathbf{W}^\top \mathbf{z}_t - (\mathbf{I}_m + \mathbf{M})\mathbf{x}_t(\gamma)$; ▷ output dynamics
    $\frac{d\mathbf{z}_t(\gamma)}{d\gamma} = \mathbf{y}_t - \mathbf{W}\mathbf{x}_t(\gamma) - \mathbf{z}_t(\gamma)$; ▷ error dynamics
  $\mathbf{W} \leftarrow \mathbf{W} + \eta \mathbf{z}_t \mathbf{x}_t^\top$; ▷ Feedforward weight updates
  $\mathbf{M} \leftarrow \mathbf{M} + \frac{\eta}{\tau}(\sigma^2 \mathbf{I}_m + \mathbf{x}_t \mathbf{x}_t^\top - \mathbf{M}^{-1})$; ▷ Lateral weight updates
end for

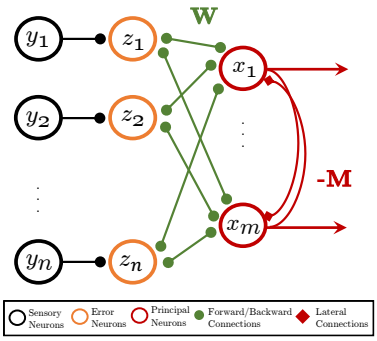

Figure 1: Neural implementation of Algorithm 1 derived from GPLVM.

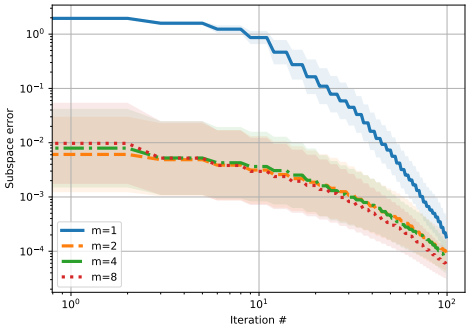

Figure 2: Numerical evaluation of our algorithm on a toy dataset for different output dimensions.

Numerical experiments. As an illustration of the capability of the NN derived from GPLVM, we provide experimental results of our algorithm. We evaluate our algorithm on an artificially generated datasets, $\mathbf{X} \in \mathbb{R}^{20 \times 10{,}000}$, with a linear spectrum, $(\lambda_j = j/10, \forall j \in \{1, 20\})$, shown in Fig.2. The performance of our algorithm is measured based on the subspace alignment error, i.e., the difference in norm square with normalized projector obtained by PCA and the one encoded in $(\mathbf{W}, \mathbf{M})$. In Fig. 2, we show that after convergence, our algorithm leads to a projection on the true basis vectors for different output dimensions.

5 Discussion

In this work, we asked the question of the implication of IT as an organizing principle for cognition and understanding of the brain. More precisely, how does the brain learn from input distributions that it has not seen before or in novel perceptive environments? Nonetheless, inspired by work on similarity-preserving NNs (recalled in Appendix B), we proposed a novel min-max objective function that gives rise to a neural implementation for GPLVM. Our work illustrates the point by discussing linear GPLVM, but this could be extended to other kernels easily, using random features [33].

We observe, however, two striking differences with non-IT-based NNs. First, the intermediate variables $\mathbf{z}_t$ play the role of error computing neurons rather than projection neurons. As a result, the update of the feedforward weights, $\mathbf{W}$, (Eq. (13)Left) minimizes the prediction error rather than maximizing the correlation of the projected input with the inputs. Secondly, the learning rule in $\mathbf{M}$, (Eq. (13)Right) involves the matrix $\mathbf{M}^{-1}$, which in turn increases the connection between correlated output. It is unlike in more standard models where lateral connection aims at decorrelating output activities through means of anti-Hebbian updates [17]. More precisely, in IT, the lateral connections $\mathbf{M}$ prevent overfitting, while in SM, it tries to avoid underfitting. This raises the question of how we can probe such an observation experimentally and favor IT or non-IT-based frameworks.

References

[1] Joseph J Atick. Could information theory provide an ecological theory of sensory processing? *Network: Computation in neural systems*, 3(2):213–251, 1992.

[2] F Rieke, D Warland, and W Bialek. Coding efficiency and information rates in sensory neurons. *EPL (Europhysics Letters)*, 22(2):151, 1993.

[3] Karl Friston. Hierarchical models in the brain. *PLoS computational biology*, 4(11):e1000211, 2008.

[4] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138, 2010.

[5] Andy Clark. Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204, 2013.

[6] Khalid Sayood. Information theory and cognition: a review. *Entropy*, 20(9):706, 2018.

[7] Ralph Linsker. An application of the principle of maximum information preservation to linear systems. *Advances in neural information processing systems*, 1, 1988.

[8] Ralph Linsker. Local synaptic learning rules suffice to maximize mutual information in a linear network. *Neural computation*, 4(5):691–702, 1992.

[9] Erkki Oja. The nonlinear pca learning rule in independent component analysis. *Neurocomputing*, 17(1):25–45, 1997.

[10] Ralph Linsker. A local learning rule that enables information maximization for arbitrary input distributions. *Neural Computation*, 9(8):1661–1665, 1997.

[11] Te-Won Lee, Mark Girolami, Anthony J Bell, and Terrence J Sejnowski. A unifying information-theoretic framework for independent component analysis. *Computers & Mathematics with Applications*, 39(11):1–21, 2000.

[12] Cengiz Pehlevan and Dmitri B Chklovskii. Neuroscience-inspired online unsupervised learning algorithms: Artificial neural networks. *IEEE Signal Processing Magazine*, 36(6):88–96, 2019.

[13] Johannes Friedrich. Neuronal gaussian process regression. *Advances in Neural Information Processing Systems*, 33:7090–7100, 2020.

[14] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[15] Konstantinos I Diamantaras and Sun Yuan Kung. *Principal component neural networks: theory and applications*. John Wiley & Sons, Inc., 1996.

[16] Cengiz Pehlevan and Dmitri B Chklovskii. A Hebbian/anti-Hebbian network derived from online non-negative matrix factorization can cluster and discover sparse features. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 769–775. IEEE, 2014.

[17] Cengiz Pehlevan, Anirvan M Sengupta, and Dmitri B Chklovskii. Why do similarity matching objectives lead to Hebbian/anti-Hebbian networks? *Neural Computation*, 30(1):84–124, 2017.

[18] Anirvan Sengupta, Cengiz Pehlevan, Mariano Tepper, Alexander Genkin, and Dmitri Chklovskii. Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[19] Yanis Bahroun, Anirvan Sengupta, and Dmitri B Chklovskii. A similarity-preserving network trained on transformed images recapitulates salient features of the fly motion detection circuit. In *Advances in Neural Information Processing Systems*, pages 14178–14189, 2019.

[20] David Lipshutz, Charles Windolf, Siavash Golkar, and Dmitri B Chklovskii. A biologically plausible neural network for slow feature analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 14986–14996, 2020.

[21] Yanis Bahroun, Dmitri Chklovskii, and Anirvan Sengupta. A normative and biologically plausible algorithm for independent component analysis. *Advances in Neural Information Processing Systems*, 34:7368–7384, 2021.

[22] Neil Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16, 2003.

[23] David Barber Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

[24] Felix Vsevolodovich Agakov. *Variational Information Maximization in Stochastic Environments*. PhD thesis, University of Edinburgh, 2005.

[25] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[26] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[27] Lucien Le Cam. *Asymptotic methods in statistical decision theory*. Springer Science & Business Media, 2012.

[28] Yixin Wang and David M Blei. Frequentist consistency of variational bayes. *Journal of the American Statistical Association*, 114(527):1147–1161, 2019.

[29] Neil Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[30] Siavash Golkar, David Lipshutz, Yanis Bahroun, Anirvan M Sengupta, and Dmitri B Chklovskii. A simple normative network approximates local non-hebbian learning in the cortex. In *Advances in Neural Information Processing Systems*, volume 33, pages 7283–7295, 2020.

[31] Nikolai M Chapochnikov, Cengiz Pehlevan, and Dmitri B Chklovskii. Normative and mechanistic model of an adaptive circuit for efficient encoding and feature extraction. *bioRxiv*, 2021.

[32] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[33] Georgios V Karanikolas, Qin Lu, and Georgios B Giannakis. Online unsupervised learning using ensemble gaussian processes with random features. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3190–3194. IEEE, 2021.

[34] Erkki Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5(6):927–935, 1992.

Appendix

A Detailed min-max steps

We provide below more details on how to obtain the min-max objective function Eq. (11). We start from the log-likelihood for GPLVM Eq.(7) recalled below

$$\mathcal{L}(\mathbf{X}) = -\frac{nT}{2}\log 2\pi - \frac{n}{2}\log\det(\sigma^2\mathbf{I} + \mathbf{X}^\top\mathbf{X}) - \frac{1}{2}\operatorname{Tr}\left[(\sigma^2\mathbf{I} + \mathbf{X}^\top\mathbf{X})^{-1}\mathbf{Y}^\top\mathbf{Y}\right] . \tag{14}$$

We now introduce auxiliary matrix variables $\mathbf{W}$ and $\mathbf{M}$. To do so we first address the term that only depends on $\mathbf{X}$, i.e., the logdet term

$$\begin{aligned}\log\det(\sigma^2\mathbf{I} + \mathbf{X}^\top\mathbf{X}) &= \operatorname{Tr}\left[\log(\sigma^2\mathbf{I} + \mathbf{X}\mathbf{X}^\top)\right] + \text{cst} \ , \\ &= \max_{\mathbf{M}} \operatorname{Tr}\left[(\sigma^2\mathbf{I} + \mathbf{X}\mathbf{X}^\top)\mathbf{M}\right] - \operatorname{Tr}\left[\log(\mathbf{M})\right] + \text{cst} \ . \end{aligned} \tag{15}$$

Now the cross-term involving both $\mathbf{X}$ and $\mathbf{Y}$ using Woodbury matrix identity and by only keeping the term that depend on $\mathbf{X}$ we obtain

$$\begin{aligned}\operatorname{Tr}\left[(\sigma^2\mathbf{I} + \mathbf{X}^\top\mathbf{X})^{-1}\mathbf{Y}^\top\mathbf{Y}\right] &= +\frac{1}{\sigma^2}\mathbf{Y}^\top\mathbf{Y} - \operatorname{Tr}\left[\mathbf{X}^\top(\sigma^2\mathbf{I} + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{Y}^\top\mathbf{Y}\right] \ , \\ &= \min_{\mathbf{W}} -2\operatorname{Tr}\left(\mathbf{Y}^\top\mathbf{W}\mathbf{X}\right) + \operatorname{Tr}\left[\mathbf{W}(\sigma^2\mathbf{I} + \mathbf{X}\mathbf{X}^\top)\mathbf{W}^\top\right] + \text{cst} \ , \\ &= \min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_F^2 + \sigma^2\|\mathbf{W}\|_F^2 + \text{cst} \ . \end{aligned} \tag{16}$$

It is important for the neural interpretation that we introduce another auxiliary variable $\mathbf{Z}$ into Eq. (9) that will account for the projection error as

$$\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_F^2 + \sigma^2\|\mathbf{W}\|_F^2 = \max_{\mathbf{Z}} \min_{\mathbf{W}} (\mathbf{Y} - \mathbf{W}\mathbf{X})^\top\mathbf{Z} + \sigma^2\|\mathbf{W}\|_F^2 - \frac{1}{2}\|\mathbf{Z}\|^2 \ . \tag{17}$$

Now we combine Eq.(8) and Eq.(10) to obtain a min-max objective function for linear GPLVM as

$$\begin{aligned}\min_{\mathbf{X},\mathbf{W}} \max_{\mathbf{Z},\mathbf{M}} \mathcal{L}(\mathbf{W},\mathbf{M},\mathbf{X},\mathbf{Z}) = \min_{\mathbf{X},\mathbf{W}} \max_{\mathbf{Z},\mathbf{M}} \operatorname{Tr}\Bigg[&\frac{1}{\sigma^2}(\mathbf{Y} - \mathbf{W}\mathbf{X})^\top\mathbf{Z} - \frac{1}{2\sigma^2}\mathbf{Z}^\top\mathbf{Z} + \frac{1}{2}\mathbf{X}^\top\mathbf{X} \\ &+ \frac{T}{2}\mathbf{W}^\top\mathbf{W} + \frac{n}{2}\left(\sigma^2\mathbf{M} + \frac{1}{T}\mathbf{X}^\top\mathbf{M}\mathbf{X} - \log(\mathbf{M})\right)\Bigg] \ . \end{aligned} \tag{18}$$

B Similarity matching for non-probabilistic linear DR

In order to derive a NN from GPLVM, we look at the similarity matching (SM) framework [17], which proved able to derive such NNs for non-probabilistic similarity-preserving objective function. SM has the following objective function, written in terms of the Gramians $\mathbf{X}^\top\mathbf{X}$ and $\mathbf{Y}^\top\mathbf{Y}$:

$$\text{SM} \ \Rightarrow \ \min_{\mathbf{X}\in\mathbb{R}^{m\times T}} \left\|\mathbf{Y}^\top\mathbf{Y} - \mathbf{X}^\top\mathbf{X}\right\|_F^2 \tag{19}$$

The work on SM took a different approach than that of Oja [34] on PCA, as detailed in [12], and showed that the resulting rules were strictly local unlike Oja's rule. They solve Eq. (19) by introducing a min-max objective function and auxiliary variables as

$$\min_{\mathbf{X}} \min_{\mathbf{W}\in\mathbb{R}^{k\times n}} \max_{\mathbf{M}\in\mathcal{S}_{++}^k} \frac{1}{T}\operatorname{Tr}(-4\mathbf{X}^\top\mathbf{W}\mathbf{Y} + 2\mathbf{X}^\top\mathbf{M}\mathbf{X}) + 2\operatorname{Tr}(\mathbf{W}\mathbf{W}^\top) - \operatorname{Tr}(\mathbf{M}\mathbf{M}^\top) \ . \tag{20}$$

The objective (20) can be optimized by an online algorithm that maps onto a NN whose synapses obey local learning rules [17]. However, it is unclear how uncertainty can be incorporated and thus produce an IT equivalent of SM for deriving NNs. We address this shortcoming by deriving NNs from the GPLVM objective function instead.