Maggie von Ebers, Yuhan Gao, Edward Leung
CS 391L Machine Learning
04/28/23

ML Final Project Final Report

https://github.com/mx60s/ML-Final-Project

Background

Application of machine learning onto neural recordings is not a novel concept. In fact, in the case of functional magnetic resonance imaging (fMRI) analysis for example, scientists in the early 2000s have already been able to achieve high accuracy neural decoding with classic algorithms such as support vector machines (SVM, Cox & Savoy, 2003). However, recent advances in deep learning, as well as the quality of neural recordings, have motivated scientists to revisit the existing practices and perhaps improve on them by adopting deep learning and unsupervised learning strategies. One promising approach is the work on latent variable identifiability.

The idea was first proposed in Khemakhem et al (2020), in which the authors synthesized ideas from previous work on identifiability in nonlinear ICA (Hyvärinen, 2018) with the deep learning concept of variational autoencoders (VAE, Kingma and Welling, 2013). Identifiability refers to the ability to recover the true latents of the data, where in the ICA paradigm, this involves identifying the transformations **f** as well as the latent variables $\mathbf{v} = (v_1, \dots, v_n)$ such that $\mathbf{x} = \mathbf{f}(\mathbf{v})$ when you are only given **x**. While this is well-defined in the linear case , it is not in the nonlinear case. This is because "for any (x_1, x_2), a $\mathbf{y} = \boldsymbol{g}(x_1, x_2)$ that is independent of x_1 can always be constructed" (Hyvärinen 2020) as

$$g(\xi_1, \xi_2) = P(x_2 < \xi_2 | x_1 = \xi_1)$$

However, if some constraint is applied to the analysis, such as by utilizing the temporal structure in the data or some auxiliary variable, then the separation of identifiable source signals can be achieved through nonlinear ICA. This concept is mirrored in VAEs, as demonstrated in Khemakhem et al (2020) where if some conditionally factorial priors are introduced, then the latents variables learned can be identified. The problem can be formulated in the following equation:

$$p_\theta(x, z|u) = p_f(x|z)p_{T,\lambda}(z|u)$$

where **x** and **u** denote two observed random variables, and **z** ($dz < dx$) denotes the latent variables. The equation is parameterized by $\theta = (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$, where **T** are the sufficient statistics, **λ** are the corresponding parameters of **u**, and **f** is the mixing function, which in this context is approximated through the neural network.

More recent work has built on this idea, notably Zhou and Wei (2020), specifically to adapt this paradigm onto neural data. Their proposed architecture, Poisson identifiable VAE (pi-VAE), can account for Poisson noise in the observed variable, as is the case for typical spike data. In this context, the observed spike count per time bin can be conceptualized as a Poisson process, which means it can be modeled as $p_f(\mathbf{x}|\mathbf{z})$ = Poisson ($\mathbf{f}(\mathbf{z})$) where $\mathbf{f}(\mathbf{z})$ is the instantaneous firing rate as a function of the latent variables **z** (Zhou and Wei, 2020). Furthermore, this innovation enabled the modeling of latent variables in relation to task variables, augmenting the model's capacity to reveal insights about the fundamental neurobiological processes. The authors demonstrated this on rat hippocampal data, which consists of neurophysiological recordings and positional data obtained when rats freely explored a 1.6m linear track. Using pi-VAE, they successfully recovered the two dimensions of the latent that encoded 1) the positional and 2) the temporal information. However, one limitation of this model is that the learned latents tend to be strongly influenced by the label priors, which can result in convergence on bad local optima

[Kim et al, 2022]. This negatively affects the model's robustness especially in self-supervised learning paradigms, leading it to produce inconsistent results [Schneider 2022].

Two groups tried to improve on the pi-VAE model, namely Schneider et al 2022 (CEBRA) and Kim et al 2022 (CI-iVAE), but their approaches were distinct enough that made it difficult to directly compare the effects of their modifications. As such, the primary focus of this project is to devise a method to perform that comparison, and in the process perhaps find some way to combine the advantages of both.

Current Models

CI-iVAE is built directly on the architecture of pi-VAE. The innovation of this model lies in the modified loss function. Consider this formulation of the evidence lower bound (ELBO):

$$E_{q\varphi(z|x,u)} \log p_f(x|z) - D_{KL}(q_\varphi(z|x, u)||p_{T,\lambda}(z|u))$$

In non-collapse cases, this loss function is able to identify an optimal $\log p_\theta(x|u)$ such that ground-truth latent variables are recovered. However, the authors showed that there are cases when the KL divergence term instead results in enforcing the equality $q_\varphi(z|x, u) = p_{T,\lambda}(z|u)$, which they refer to as the posterior collapse problem of iVAEs. This results in convergence to a bad local optimum, whereby the KL divergence term dominates the solution and the resulting latents are independent of the observations given the covariates. In other words, this means that the latents only describe the covariates and thus nothing is learned about the relationship between the neural data and the latents. Their solution was to modify the ELBO to the formulation below, which they showed was also an ELBO:

$$E_{q\varphi(z|x,u)} \log p_f(x|z) - D_{KL}(q_\varphi(z|x)||p_{T,\lambda}(z|u))$$

This new KL divergence term compares the distributions of latents obtained from the encoder $q_\varphi(z|x)$ with the distribution of latents conditioned on the priors u, essentially ensuring that the term could never reach $q_\varphi(z|x, u) = p_{T,\lambda}(z|u)$, rather only obtain the best approximation.

CEBRA instead uses a completely different architecture. It utilizes a contrastive optimization objective, InfoNCE, along with a unique self-supervised learning algorithm to generate the latent embeddings. Furthermore, it has the capability to take in either user-defined labels (hypothesis-driven or behavior based), time-only labels (discovery-driven or time-based), or even both to obtain consistent embeddings of neural activity. Running it in hypothesis-driven mode involves training with known knowledge, which achieves the same purpose as modeling the covariate structure in CI-iVAE. In contrast, running it in time-only labels or discovery-driven mode assumes nothing about the activities of the subjects, and the model is trained using the inherent structure of the data, e.g. temporal.

Impressively, CEBRA produces similar embeddings from both Behavior- and Time-based modes. This is in stark contrast to the latents obtained from pi-VAE, which the authors of CEBRA have shown were inconsistent when it was run with versus without label priors. They suggested that this meant the label prior strongly shapes the output embedding structure of pi-VAE, a critique not dissimilar to the posterior collapse problem addressed by CI-iVAE. Given this, we were able to narrow down our comparison to examining the consistency and reliability of the latent structures generated by the models. Also, while the papers explored multiple datasets, including generated data and classic datasets like the MNIST, we only focused on the rat hippocampal dataset.

Maggie von Ebers, Yuhan Gao, Edward Leung
CS 391L Machine Learning
04/28/23

Methods

Conv-CI-iVAE

In their comparison of CEBRA against pi-VAE, Schneider et al (2022) first proposed conv-pi-VAE, which involves adding convolutional layers to the original pi-VAE implementation (the same encoder neural network as CEBRA). This enabled it to accommodate larger time bin inputs and thus achieve higher levels of precision. Our group decided that this modification was necessary not only to make the comparison as equal as possible, but also to accommodate the unique properties of neurophysiological data, which we discuss in a later section as well. Therefore, the first thing our group did was to apply similar changes to CI-iVAE. Since we do not have the source code for CEBRA, we had to build our own encoder network for CI-iVAE using the methods described in CEBRA.

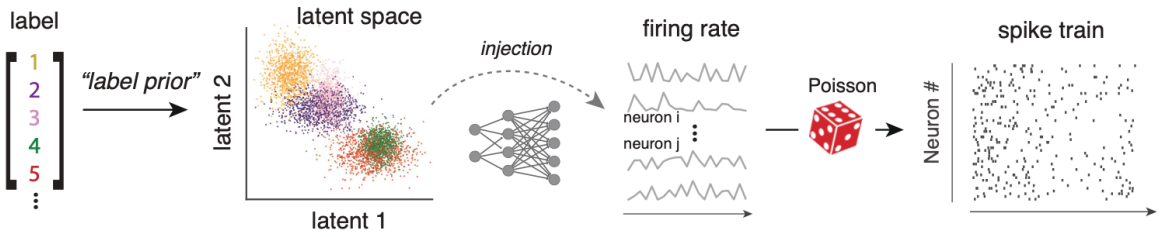
Specifically, they described a 5-layer convolution starting with a time-window of 10, applying skip connections that first reduced the dimensionality by half, then again down to the latent dimension. In the hippocampal data, this meant starting with a window of dimensions (120,10), then convolving over the second dimension and linearly reducing the first dimension down to 2. However, when we tried to implement that in CI-iVAE, we realized that the input to the decoder was supposed to be of the shape (2,2), which represent two latent variables that describe two distributions each with a mean ($\boldsymbol{\mu}$) and variance ($\boldsymbol{\sigma}^2$). This was confusing, since it was unclear how they were able to bring the second dimension down while adding skip connections. Furthermore, this presented a secondary problem, which was that we would need to recover the original dimensionality in order for the x and x' to be compared properly. While we came up with several ideas to overcome these challenges, such as adding a transpose convolution layer into the decoder to recover the dimensionality, these problems were overshadowed by another bigger problem.

Assumption of Poisson Distribution

The original pi-VAE implementation involves a Softmax activation layer at the end of the decoder to recover the Poisson structure of the neural activity (fig 1). However, if we were to apply the same strategy in the conv-CI-iVAE architecture, that unfortunately interferes with the formulation of the ELBO, which evaluates the distributions based on an assumption of Gaussianity. The crucial difference is the application of the convolution, which is our novel implementation that did not exist in both the original pi-VAE and CI-iVAE architectures. Given this, we reconsidered our original approach. Instead of attempting to apply a convolution in the encoder, then recovering the Poisson spike train data in the decoder, we could instead apply Gaussian smoothing to the input before it is fed into the encoder of CI-iVAE. The rationale behind this is that if the neural spiking data draws from a Poisson distribution, then by smoothing 10 of the time bins, we would get something closer to a Gaussian distribution by the Central Limit Theorem. We would then feed the now smoothed firing rate into the loss calculation along with the output from the decoder, which will now both be Gaussian distributed.

Maggie von Ebers, Yuhan Gao, Edward Leung
CS 391L Machine Learning
04/28/23

a



b

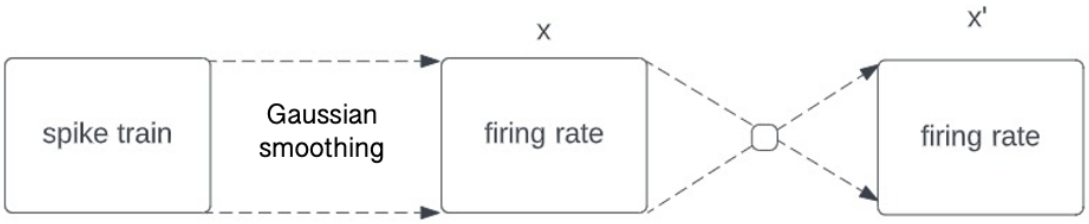
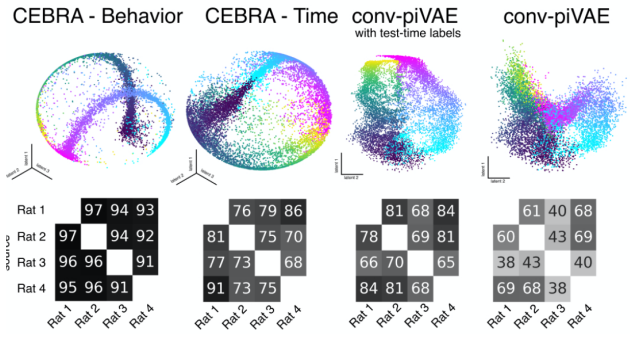


Figure 1. The model architectures of pi-VAE and our proposed conv-CI-iVAE. **a**) taken from Zhou and Wei (2020). Depicts the architecture of pi-VAE, which involves transforming the labels into some latent space through a label prior network, then an injective mapping between these latents and firing rate (generated from applying VAE architecture onto neural data), before finally incorporating Poisson noise to recover x' in the form of spike train data. **b**) Our modified CI-iVAE architecture. Involves performing Gaussian smoothing on the spike train input to obtain firing rate, then feeding that as input into the original CI-iVAE architecture. The smaller square denotes the (2,2) matrix containing the latent variables that are output from the encoder and input into the decoder.

Results and Discussion

To compare the reliability of the latents generated by the models, we visualized the latents in a 2-D space. CEBRA offers 3-D latents as their encoders are based off of a simple non-linear neural network, but our model encoders give 2-D latents as they are describing the mean and variance of encoded distributions.

a.



b.

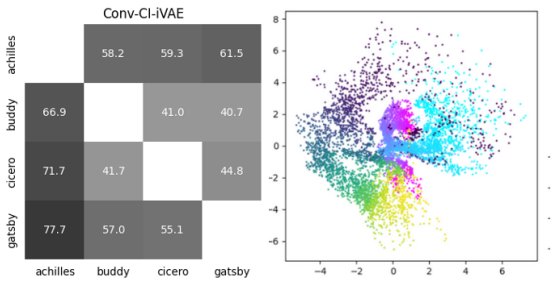


Figure 2. We graphed the individual data points based on their transformation into latent space, and then, similar to how both Schneider et al and Kim et al presented their results, we colored each point with its corresponding location on the 1.6m track, with two different colorbars representing the left and right

directions. **a**) taken directly from Schneider et al 2020, and depict their comparisons of CEBRA and their implementation of conv-piVAE. **b**) Conv-CI-iVAE latent visualization and cross-subject latent representation correlation.

We mainly consider the Achilles rat, referred to as “Rat 1” in the CEBRA paper, as he generated the highest quality and largest amount of neural data, and was thus used for most of the CEBRA paper demonstrations. Conv-CI-iVAE does a comparable job at separating and organizing its latent variables when compared to the best non test-time Conv-PI-iVAE results, which appear to have collapsed and condensed in the middle. Additionally, we used linear regression to fit latents for each of the rats with each other in order to measure the generalizability of the latents across subjects. Because our latents are meant to explain the underlying mechanisms of how the hippocampus keeps track of a rat’s position and presumably this function is the same for every brain, they should not be too different across animals. This was a concern stated in the CEBRA paper about PiVAE – that it collapsed on behavioral priors and thus wasn’t generalizable. Conv-CI-iVAE shows a marked improvement over Conv-CI-iVAE in this respect. This indicates that Conv-CI-iVAE improves upon Conv-PiVAE in the way CI-iVAE was meant to improve upon PiVAE.

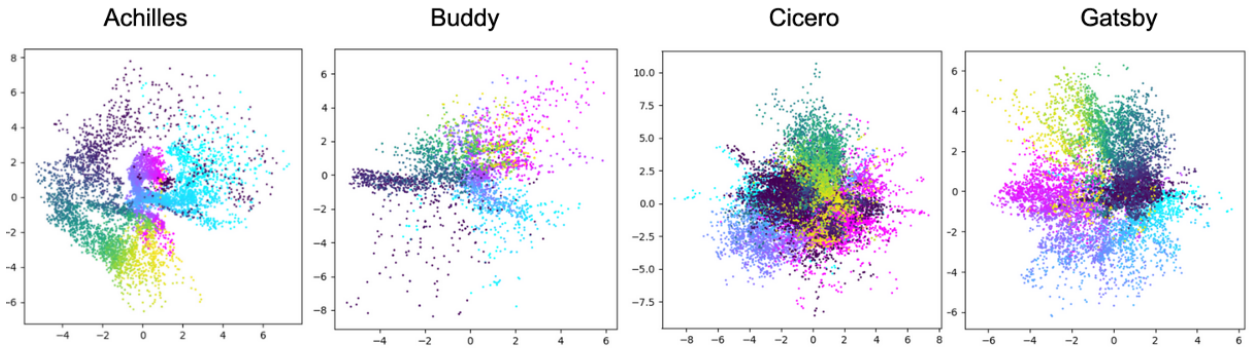


Figure 3. Latent visualization for each individual rat generated from conv-CI-iVAE..

Another method of examining reliability of the latent representations is to train the models under different random seeds. As suggested by the authors Zhou and Wei (2020), pi-VAE could perform differently depending on the random seeds used to initialize the weights. Specifically, this was referring to the posterior collapse cases. Importantly, the alternate loss function formulation of CI-iVAE directly addresses this problem, and therefore, latents obtained using different random seeds should be consistent. As such, we would expect the latents generated to also look consistent in our visualizations, which is in fact what we see. Fig 4 depicts this comparison, where first, the latents generated using the same seed for the train, test and validation splits show consistent layout. Second and more importantly, the latents generated using different random seeds (fig 4a & d), are also mostly consistent, with the exception that the representations look flipped. However, given that we know that one of the ground-truth latent dimensions represent the positional information, and given that position is encoded by a continuous position on the track as well as the left and right directions, we conclude that the flipping of the visualizations on the first latent dimension can be explained by the invariance of the model to the left and right directions.

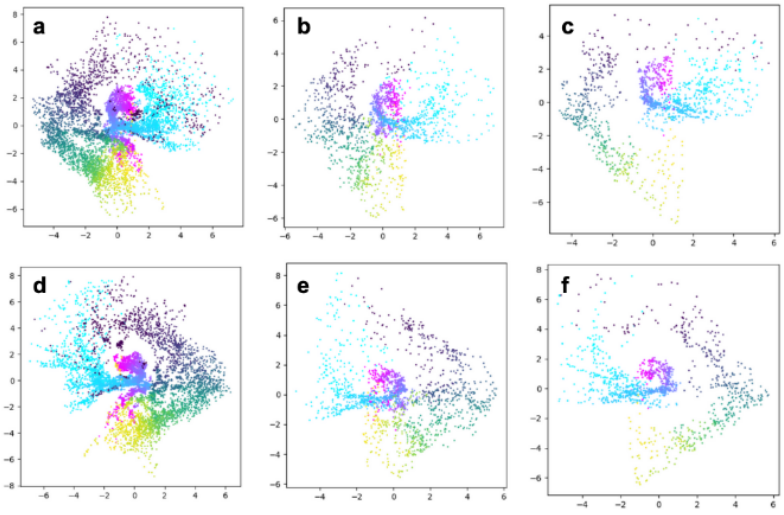


Figure 4. Performance of latent generation using different seeds. **a**,**b**,**c**) Latents of rat “achilles” generated for the train, test and validation sets using seed 420. **d**,**e**,**f**) Latents of rat “achilles” generated for train, test and validation sets using seed 100.

Future Work

There are a number of ways in which we believe this work could be expanded upon. To start with the basics, our model could benefit from some additional time spent fine-tuning. We decided to use the learning rate, number of epochs, and other hyperparameters that were used for Conv-PiVAE in order to draw a direct comparison, but faced some issues with our model becoming unstable, especially with some of the rats that had less neuronal data to work with. Given that CI-iVAE uses a slightly different loss calculation than PiVAE and that we have a simpler convolutional architecture for the encoder, it makes sense that there might need to be different training strategies implemented. There are some hyperparameters that we took directly from the non-convolutional CI-iVAE architecture which include hidden layer sizes. These could be tweaked as well, as it is reasonable to expect that when changing an encoder, the decoder may need to be revisited.

Additionally, these aforementioned convolutional architectures could be improved upon. Given that Conv-PiVAE uses a deep convolutional network to reduce its dimensionality, whereas Ci-iVAE simply uses one convolution with a fixed Gaussian kernel, there are likely many fruitful convolutional architectures that could be used. There was a significant amount of discussion in our group throughout the project as to whether or not we could convert the CI-iVAE model to have the same firing rate inputs to the encoder as PiVAE does. This would enable us to ditch the smoothing that this Gaussian kernel introduces and give our model some additional information to work with, but it would also necessarily impact the calculation of the loss function significantly, as the current Ci-IVAE loss calculates the KL divergence between two distributions which are assumed to be Gaussian. We believe that this is possible, but the necessary derivations were outside of the scope and time constraints of this project. If we did refactor the loss function, we could convolve over the 10 time bins in the same way that Conv-PiVAE does and perhaps increase performance.

Finally, further comparisons to CEBRA could be drawn with additional experiments on the Conv-CI-iVAE model. CEBRA is shown to work in several different contexts, including

conditioning on a hypothesis in order to aid in neuroscience research, as well as using continuous time labels instead of behavioral labels (which is what we have trained Conv-CI-iVAE on, and drawn all of the comparisons in this paper on). Conv-CI-iVAE has already demonstrated that it shows clear structures in its latent space when conditioned on discrete behavioral data, so it could be reasonably expected to perform well across these other in-domain experiments.

Hyvarinen, Aapo, et al. “Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning.” *ArXiv.org*, 4 Feb. 2019, https://arxiv.org/abs/1805.08651.

Khemakhem, Ilyes, et al. “Variational Autoencoders and Nonlinear ICA: A Unifying Framework.” *ArXiv.org*, 21 Dec. 2020, https://arxiv.org/abs/1907.04809.

Kim, Young-geun, et al. “Covariate-Informed Representation Learning to Prevent Posterior Collapse of Ivae.” *ArXiv.org*, 14 Oct. 2022, https://arxiv.org/abs/2202.04206.

Kingma, Diederik P, and Max Welling. “Auto-Encoding Variational Bayes.” *ArXiv.org*, 10 Dec. 2022, https://arxiv.org/abs/1312.6114.

RL;, Cox DD;Savoy. “Functional Magnetic Resonance Imaging (Fmri) ‘Brain Reading’: Detecting and Classifying Distributed Patterns of Fmri Activity in Human Visual Cortex.” *NeuroImage*, U.S. National Library of Medicine, https://pubmed.ncbi.nlm.nih.gov/12814577/.

Schneider, Steffen, et al. “Learnable Latent Embeddings for Joint Behavioral and Neural Analysis.” *ArXiv.org*, 5 Oct. 2022, https://arxiv.org/abs/2204.00673.

Zhou, Ding, and Xue-Xin Wei. “Learning Identifiable and Interpretable Latent Models of High-Dimensional Neural Activity Using PI-Vae.” *ArXiv.org*, 9 Nov. 2020, https://arxiv.org/abs/2011.04798.