Efficient decomposition of latent representation in generative models

Vsevolod Nikulin
Cognitive Neurorobotics Research Unit
Okinawa Institute of Science and Technology
Okinawa, Japan
vsevolod.nikulin@oist.jp

Jun Tani
Cognitive Neurorobotics Research Unit
Okinawa Institute of Science and Technology
Okinawa, Japan
jun.tani@oist.jp

***Abstract*—In designing self-organizing generative models of robot behaviour, it is important to address the issue of generalization for multiple patterns, while keeping latent representation in a low dimension. We are investigating the possibility of introducing local coordinates for samples of each pattern in shared latent representation. Moreover, recent advances in efficient, approximate computation of Jacobians allows us to introduce specific regularization that ensures directional robustness in introduced local coordinates.**

***Index Terms*—machine learning, generative model, regularization, Jacobian**

I. Introduction

Generative models allow representation of high-dimensional behaviour patterns (sequences of action-perception pairs) in much lower-dimensional latent representations. The choice of a latent dimension must serve two conflicting purposes: generalization capacity, which can be achieved by providing more degrees of freedom, and the capacity to reconstruct unseen samples, which is more tractable with fewer degrees of freedom. Reconstruction of unseen samples is very important [1]–[4]. In this paper we assume that samples belonging to a given pattern can be described with fewer dimensions than the model's latent representation. In such case, it is possible to explicitly define an embedding of hypersurfaces corresponding to each pattern in shared latent space. This solves both aforementioned problems: shared latent representation will have enough degrees of freedom to generalize for many patterns, while keeping the problem of reconstructing specific points for unseen samples of a given pattern fairly tractable. A similar question was raised in [5], where the possibility of assigning independent linear subspaces for each pattern was addressed.

However, this approach demands learning additional parameters for each pattern. Maps of local coordinates corresponding to each pattern to shared latent representation expand the search space and add more local minima. Moreover, these maps may introduce additional instability and may make the search space for reconstruction of unseen samples less robust. We solve these new problems by developing a novel regularization technique. This regularization forces these mappings to be directionally robust: changes in shared latent representation caused by small shifts in local coordinates are invariant to directions of these shifts. We show that this condition is equivalent to local conformal flatness of embedded hypersurfaces, and can be described in terms of Jacobian matrices of these embeddings. Furthermore, we present an efficient way of approximating computations of the required properties of these Jacobians.

To investigate the impact of such a geometrical approach, independent of any statistical properties of the model, we consider only deterministic generative models.

The rest of the paper is organized as follows. In Section 2 we provide a formal description of the problem. Then, in Section 3 we introduce the proposed approach for general generative models with finite dimensional latent representations (e.g. parametric bias). Next, in Section 4 we describe the architecture we use to empirically prove the conjecture, and we supply experimental results. Finally we draw conclusions in Section 5.

II. Formulation of the problem

Let us consider a simple deterministic generative model family $P_\theta(X = \mathbf{x}|Z = \mathbf{z}) = \delta(\mathbf{x} - \mathbf{f}_\theta(\mathbf{z}))$, where δ is the Dirac delta function, X is the observed variable, Z is the parameter of the model (i.e. latent variable), and θ parametrizes the whole family of mappings $\mathbf{f}_\theta$. We assume values that Z have finite dimension: $\mathbf{z} \in \mathbb{R}^L$. Values of X describe generated sequences: $\mathbf{x} = \{(\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^T) | \mathbf{x}^t \in \mathbb{R}^O \text{ for } t = 1, 2, ..., T\}$. The goal is to find values of θ and $\mathbf{z}_i$ for each observed sample $\mathbf{x}_i$ (for $i = 1, 2, ..., N$) which minimize the error function

$$\mathcal{L}(\theta, \mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N) = \frac{1}{N * T} \sum_{i=1}^{N} \sum_{t=1}^{T} \left\| \mathbf{x}_i^t - \mathbf{f}_\theta^t(\mathbf{z}_i) \right\|^2. \quad (1)$$

Notice that if we have observed samples belonging to a diverse set of patterns, we must keep dimensions of $\mathbf{z}$ fairly high.

A. Reconstruction of partially observed sequences

After learning the model and fixing θ, another interesting problem to consider is identifying $\mathbf{z}$ by $\mathbf{x}$ given only partially. Remember that $\mathbf{x}$ describes a sequence of action-perception pairs: $\mathbf{x}^t = (\mathbf{a}^t, \mathbf{p}^t)$. It is very common [1]–[4] to ask how

978-1-7281-2547-3/20/$31.00 ©2020 IEEE

we can complete a sequence by having only part of it (e.g. perception at the first time step):

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \left\| \mathbf{p}^1 - \hat{\mathbf{p}}^1(\mathbf{z}) \right\|^2, \tag{2}$$

where $\hat{\mathbf{p}}^1(\mathbf{z})$ corresponds to projection of $\mathbf{f}_\theta^1(\mathbf{z})$ on coordinates corresponding to the perceptible part of observed samples. Then, knowing $\mathbf{z}^*$ it is possible to complete the sequence using $\mathbf{f}_\theta(\mathbf{z}^*)$. For simpler solution of this problem it is desirable to have dimensions of $\mathbf{z}$ as low as possible. Notice how it contrasts with the original problem.

III. Method

We assume that each observed sample can be identified with a specific pattern and that the number of patterns M is much lower than the number of samples N. According to our next assumption, samples belonging to each pattern p can be described by a generative model with lower dimensions K_p of latent variables Z than the dimension needed to describe all samples of all patterns L. The method requires manual assignment of labels to each sample to identify which pattern a given sample represents.

A. Local coordinates

We explicitly define for each pattern p an embedding $\mathbf{m}_\mu^p : \mathbb{R}^K \mapsto \mathbb{R}^L$. Now instead of learning shared latent coordinates $\mathbf{z}_i$ for each sample, we learn local coordinates $\xi_i \in \mathbb{R}^K$. Furthermore, embedding maps $\mathbf{m}_\mu^p$ are also equipped with parameters μ to be learned.

B. Robustness analysis

Consider a small perturbation vector $\epsilon \in \mathbb{R}^K$ at point ξ in local coordinates of a pattern. The corresponding values of shared latent representation $\mathbf{z}$ shift to

$$\begin{aligned}\tilde{\mathbf{z}} = \mathbf{m}_\mu^p(\xi+\epsilon) &= \mathbf{z}(\xi) + \sum_{j=1}^{L}\sum_{i=1}^{K} \epsilon_i \frac{\partial z_j}{\partial \xi_i}(\xi)\mathbf{e_j} + o(\|\epsilon\|^2) \\ &= \mathbf{z}(\xi) + \mathbf{J}(\xi)\epsilon + o(\|\epsilon\|^2),\end{aligned}$$

where $\mathbf{J}(\xi)$ (for the sake of brevity, we will write just $\mathbf{J}$) is the Jacobian matrix of embedding $\mathbf{m}_\mu^p$:

$$J_{j;i} = \frac{\partial z_j}{\partial \xi_i}.$$

Notice that ignoring the $o(\|\epsilon\|^2)$ term, the absolute value of the shift in shared latent representation depends not only on the absolute value of ϵ, but also on its direction. It is possible that perturbations in different directions will have different effects on values of latent variables. For robust gradient descent in local coordinate space, it is better to avoid such differences. In other words, for uniformly sampled unit vectors $\hat{\mathbf{v}} \in S^{K-1}$ at each point of local coordinate space, we want to have linear approximations of their projections to shared latent representation (this is known as *pushforward* in differential geometry) to have low variance in their absolute values. The quantity

$$\mathrm{Var}_{\hat{\mathbf{v}}\sim S^{K-1}}\left(\left\|\mathbf{J}\hat{\mathbf{v}}\right\|^2\right) \tag{3}$$

is to be minimized.

C. Efficient approximate algorithm

It can be shown that minimization of (3) is equivalent to minimization of the following value (see Appendix A for the derivation):

$$\mathcal{L}_\perp = \mathbb{E}_{\substack{\hat{\mathbf{v}},\hat{\mathbf{w}}\sim S^{K-1} \\ \hat{\mathbf{v}}\cdot\hat{\mathbf{w}}=0}}\left(\hat{\mathbf{v}}^T\mathbf{J}^T\mathbf{J}\hat{\mathbf{w}}\right)^2. \tag{4}$$

For uniformly sampled orthogonal pairs of unit vectors $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ at every point in local coordinate space, their projections in shared latent variable space should also be close to orthogonal, on average. Ideally, when orthogonal directions are transformed to orthogonal directions by $\mathbf{m}_\mu^p$, we say that the embedding is *locally conformally flat*.

Then, as in [6], we introduce a regularization term to the loss function, based on Monte-Carlo sampling for (4). There is no need for explicit computation of the Jacobian of the embedding. In order to compute the quantity under the expectation in (4), it is sufficient to compute the Jacobian vector product, which has the same computational complexity as forward computation of $\mathbf{m}_\mu^p$ through forward automatic differentiation using dual numbers [7], which can be efficiently computed in mini-batches.

First, the number of samples in Monte-Carlo approximation of (4) is denoted as n_{MC}. Then, the pseudocode is presented in Algorithm 1. Lines 3-6 describe the process of sampling pairs of unit orthogonal vectors. Random vectors are selected for each sample in a mini-batch from standard normal distribution and a Gram-Schmidt orthonormalization process is performed for each pair. Lines 7-8 contain the call of the Jacobian vector product-computing function, which can be implemented using the aforementioned forward automatic differentiation.

Having $\mathcal{L}_\perp$ computed, it is added to the original loss function with a multiplier λ for regularization. Retaining a full computational graph it is possible to compute $\partial\mathcal{L}_\perp/\partial\mu$ for gradient descent optimization using any automatic differentiation algorithm.

The full loss function is summarized in the following equation:

$$\begin{aligned}\mathcal{L}(\theta,\mu,\xi_1,\xi_2,...,\xi_N) = &\frac{1}{N*T}\sum_{i=1}^{N}\sum_{t=1}^{T}\left\|\mathbf{x}_i^t - \mathbf{f}_\theta^t \circ \mathbf{m}_\mu^{p_i}(\xi_i)\right\|^2 \\ &+ \lambda\cdot\mathbb{E}_{\substack{\hat{\mathbf{v}},\hat{\mathbf{w}}\sim S^{K-1} \\ \hat{\mathbf{v}}\cdot\hat{\mathbf{w}}=0}}\left(\hat{\mathbf{v}}^T\mathbf{J}^T(\xi_i)\mathbf{J}(\xi_i)\hat{\mathbf{w}}\right)^2,\end{aligned}$$

where p_i is the pattern label for ith sample.

Algorithm 1

Input: Mini-batch of B examples $\mathbf{x}_i$, pattern labels p_i, local coordinates ξ_i and embedding parameters μ

Output: Average scalar product $\mathcal{L}_\perp$ of pushforwards of orthogonal unit vector pairs by embedding maps $\mathbf{m}_\mu^{p_i}$

$\mathcal{L}_\perp = 0$
for $l = 1, 2, \ldots, n_{MC}$ **do**
  $\{\hat{v}_i^j\}, \{\hat{w}_i^j\} \sim \mathcal{N}(0, 1)$
  $\hat{\mathbf{v}}_i = \hat{\mathbf{v}}_i / \|\hat{\mathbf{v}}_i\|$
  $\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_i - (\hat{\mathbf{w}}_i \cdot \hat{\mathbf{v}}_i)\hat{\mathbf{v}}_i$
  $\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_i / \|\hat{\mathbf{w}}_i\|$
  $\mathbf{J}_i \hat{\mathbf{v}}_i = \mathrm{jvp}(\mathbf{m}_\mu^{p_i}, \xi_i, \hat{\mathbf{v}}_i)$
  $\mathbf{J}_i \hat{\mathbf{w}}_i = \mathrm{jvp}(\mathbf{m}_\mu^{p_i}, \xi_i, \hat{\mathbf{w}}_i)$
  $\mathcal{L}_\perp = \mathcal{L}_\perp + (\hat{\mathbf{v}}_i^T \mathbf{J}_i^T \mathbf{J}_i \hat{\mathbf{w}}_i)^2 / (B n_{MC})$
end for
return $\mathcal{L}_\perp$

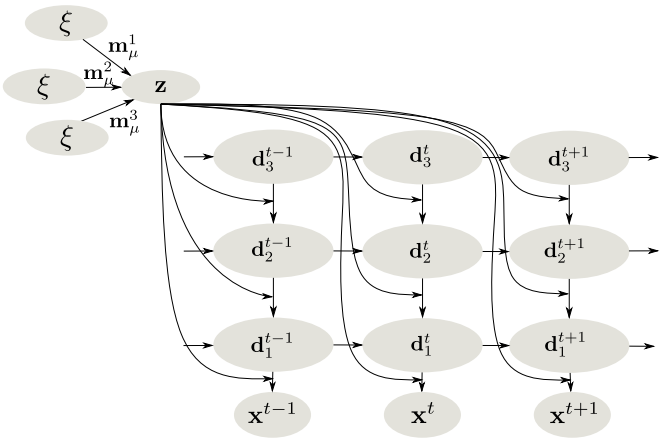


Fig. 1: Unfolded schematics of the recurrent generative model. The model is standard PVRNN with no stochastic nodes and multiplicative parametric bias $\mathbf{z}$ serving as latent variables.

IV. Experiments

In this section, we evaluate the effectiveness of the proposed approach. Performance is measured by the quality of reconstruction of partially observed sequences.

A. Generative model architecture

For these experiments we use a generative model based on modified PVRNN architecture [8]. In contrast to the original implementation, all stochastic nodes of the network were removed. Latent variables are introduced as parametrizations of weights of top-down connections between layer $l-1$ and l corresponding to different time scales, W_{ij}^l:

$$W_{ij}^l(\mathbf{z}) = \sum_{k=1}^{L} M_{ijk}^l z_k + B_{ij}^l,$$

where M_{ijk}^l and B_{ij}^l are tensors of learnable parameters. This architecture is depicted in Fig. 1. Moreover, initial states are also regarded as parameters to learn. They differ for different patterns, but are the same for samples belonging to one pattern.

B. Experimental setup

Experiments were performed using a Torobo Arm robot simulated in CoppeliaSim [9]. The experimental setup consists of the robot arm, a table in front of it, and a small red block. The arm interacts with the block, located at different positions on the table. See Fig. 2

There are two behaviour patterns:

1) The arm approaches the block and returns to the initial position
2) The arm approaches the block, makes a circular motion in proximity to the block, and returns to the initial position.

The position of the Torobo arm is defined by 9 joint angles. Training samples comprise sequences of these joint angles (actions) together with coordinates of the block (perceptions).

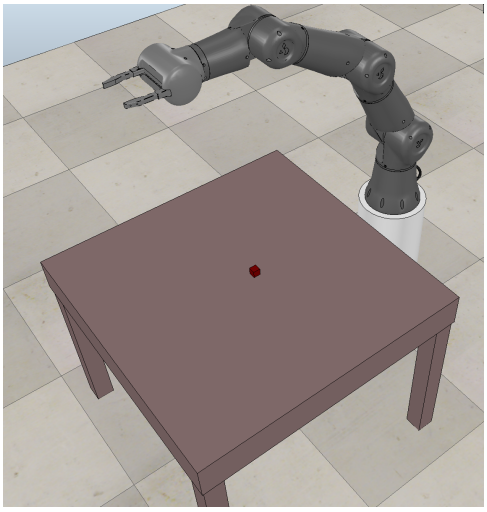


Fig. 2: The experimental setup.

Time resolution of the sequences is 100 milliseconds and there are 20 time steps in each sequence. These samples correspond to interactions with the block at different locations on the table. The dimension of shared latent space is 4. The dimension of embedding is 2. Furthermore, there are also test samples to reconstruct. Test samples are not provided during learning. The goal is to infer latent variable values for these unseen sequences by knowing only block coordinates at the first time step. See Fig. 3.

C. Experimental results

The error of reconstruction of unseen samples is compared for three cases: (i) no embedding of hypersurfaces corresponding to different patterns, (ii) embedding without regularization, and (iii) embedding with proposed regularization. In the first case, deciding which pattern to reconstruct while knowing only the initial position of the block is accomplished by providing only learned initial states, while in the remaining cases, we also use different embeddings. The learning process

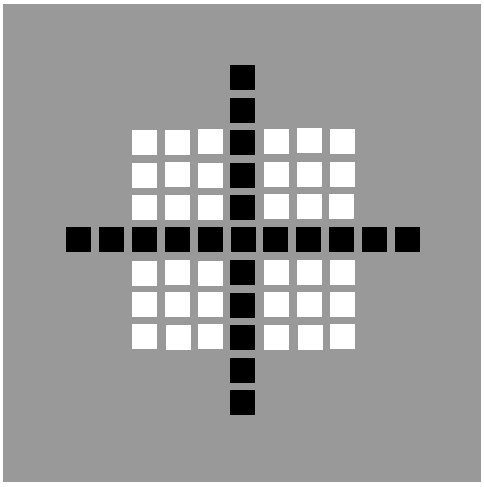
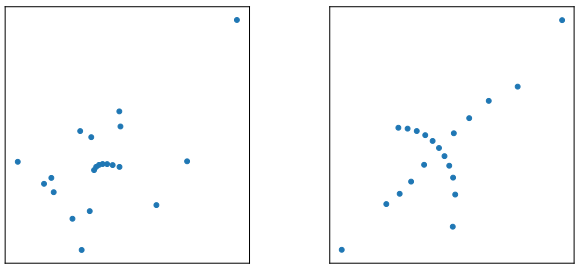


Fig. 3: The block positions on the table. Black corresponds to training samples. White corresponds to test samples.



(a) Without regularization (b) With regularization

Fig. 4: Learned local coordinates ξ of the training samples of pattern 1 in two cases. (a) Without regularization it is hard to see any pattern looking at the points. (b) With regularization, it is clear that this placement of the points resembles the locations of the block on the table.

in all cases consists of the same number of epochs. For reconstruction of unseen samples, there is also a fixed number of steps for all approaches. The convergence rate is roughly the same and will not be compared. The average squared errors of reconstruction of unseen sequences are based only on perception at the first time step together with the standard deviation of errors for three independent runs for each considered case (Table I).

TABLE I: Average squared error in reconstruction of unseen samples

	Pattern 1	Pattern 2
No embedding	0.0032 ± 0.0025	0.0015 ± 0.0009
Embedding, no regularization	0.0045 ± 0.0034	0.0037 ± 0.0037
Embedding, regularization	0.0004 ± 0.0001	0.0009 ± 0.0002

Notice how introduction of embedding further increases the reconstruction error. This happens because of additional non-linear transformations, which contribute to instability. Regularization is needed to offset that. Compare local coordinates for training samples in a case of embedding with and without regularization at Fig. 4

D. Conclusion

Explicit distinctions between patterns in generative model makes reconstruction of unseen samples more tractable. This approach could be extended further by also learning a distance function in shared latent space, which would help to identify the pattern of an unseen sample, if it is not given explicitly. The proposed Jacobian-based regularization helps to restrict enlargement by new parameters of the embedding map search space.

For future work, it will be very interesting to investigate global geometrical properties of embeddings corresponding to different patterns. Shifts in local coordinates of a given pattern could be interpreted as a symmetry Lie group: performing the same action with an object located at different positions, or changing the point of view. Topology of these Lie groups may reveal some features of symmetries, such as the periodic nature of rotation of an object or the camera.

REFERENCES

[1] J. Tani and S. Nolfi, "Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems," *Neural Networks*, vol. 12, no. 7-8, pp. 1131–1141, 1999.

[2] M. Ito and J. Tani, "On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system," *Adaptive Behavior*, vol. 12, no. 2, pp. 93–115, 2004.

[3] J. Hwang, J. Kim, A. Ahmadi, M. Choi, and J. Tani, "Dealing with large-scale spatio-temporal patterns in imitative interaction between a robot and a human by using the predictive coding framework," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 5, pp. 1918–1931, 2020.

[4] A. Ahmadi and J. Tani, "A novel predictive-coding-inspired variational rnn model for online prediction and recognition," *Neural computation*, vol. 31, no. 11, pp. 2025–2074, 2019.

[5] J. Klys, J. Snell, and R. Zemel, "Learning latent subspaces in variational autoencoders," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 6444–6454. [Online]. Available: http://papers.nips.cc/paper/7880-learning-latent-subspaces-in-variational-autoencoders.pdf

[6] J. Hoffman, D. A. Roberts, and S. Yaida, "Robust learning with jacobian regularization," 2019.

[7] P. H. Hoffmann, "A hitchhiker's guide to automatic differentiation," *Numer. Algorithms*, vol. 72, no. 3, p. 775–811, Jul. 2016. [Online]. Available: https://doi.org/10.1007/s11075-015-0067-6

[8] A. Ahmadi and J. Tani, "A novel predictive-coding-inspired variational rnn model for online prediction and recognition," *Neural Computation*, vol. 31, no. 11, pp. 2025–2074, 2019, pMID: 31525309. [Online]. Available: https://doi.org/10.1162/neco_a_01228

[9] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013, www.coppeliarobotics.com.

Appendix A

If the exact solution $\mathbf{J}^*$ for the problem of minimization (3) is achieved:

$$\mathrm{Var}_{\hat{\mathbf{v}}\sim S^{K-1}}\left(\left\|\mathbf{J}^*\hat{\mathbf{v}}\right\|^2\right)=0.$$

Then there is a constant $C\in\mathbb{R}$ such that for any unit vector $\hat{\mathbf{v}}\in S^{K-1}$ we have

$$\left\|\mathbf{J}^*\hat{\mathbf{v}}\right\|^2=\hat{\mathbf{v}}^T\mathbf{J}^{*T}\mathbf{J}^*\hat{\mathbf{v}}=C.$$

In other words,

$$\min_{\hat{\mathbf{v}}\in S^{K-1}}\hat{\mathbf{v}}^T\mathbf{J}^{*T}\mathbf{J}^*\hat{\mathbf{v}}=\max_{\hat{\mathbf{v}}\in S^{K-1}}\hat{\mathbf{v}}^T\mathbf{J}^{*T}\mathbf{J}^*\hat{\mathbf{v}}=C.$$

This means all eigenvalues of quadratic form $\mathbf{J}^{*T}\mathbf{J}^*$ are the same. There is only one option what it can be: $\mathbf{J}^{*T}\mathbf{J}^*=C\mathbb{I}$.

Next, remember another defining property of matrices having the form $C\mathbb{I}$: all vectors are eigenvectors; hence, all of them preserving direction, and all angles are also preserved. It is necessary and sufficient to demand all vectors $\hat{\mathbf{w}}$ from orthogonal subspace of any vector $\hat{\mathbf{v}}$ to stay orthogonal for transformed vector $\mathbf{J}^{*T}\mathbf{J}^*\hat{\mathbf{v}}$:

$$\left(\hat{\mathbf{v}}\cdot\hat{\mathbf{w}}=0\implies\mathbf{J}^{*T}\mathbf{J}^*\hat{\mathbf{v}}\cdot\hat{\mathbf{w}}=0\right)\iff\left(\mathbf{J}^{*T}\mathbf{J}^*=C\mathbb{I}\right).$$

This condition expressed in terms of minimisation of unbiased estimator gives us (4):

$$\mathbf{J}^*=\underset{\mathbf{J}\in\mathbb{R}^K\otimes\mathbb{R}^L}{\arg\min}\ \mathbb{E}_{\substack{\hat{\mathbf{v}},\hat{\mathbf{w}}\sim S^{K-1}\\ \hat{\mathbf{v}}\cdot\hat{\mathbf{w}}=0}}\left(\hat{\mathbf{v}}^T\mathbf{J}^T\mathbf{J}\hat{\mathbf{w}}\right)^2.$$