Gradient-based Training of Slow Feature Analysis by Differentiable Approximate Whitening

Merlin Schüler   MERLIN.SCHUELER@INI.RUB.DE
Hlynur Davíð Hlynsson   HLYNUR.HLYNSSON@INI.RUB.DE
Laurenz Wiskott   LAURENZ.WISKOTT@INI.RUB.DE
Institute for Neural Computation, Ruhr-Universität Bochum, Germany

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

We propose Power Slow Feature Analysis, a gradient-based method to extract temporally slow features from a high-dimensional input stream that varies on a faster time-scale, as a variant of Slow Feature Analysis (SFA) that allows end-to-end training of arbitrary differentiable architectures and thereby significantly extends the class of models that can effectively be used for slow feature extraction. We provide experimental evidence that PowerSFA is able to extract meaningful and informative low-dimensional features in the case of (a) synthetic low-dimensional data, (b) ego-visual data, and also for (c) a general dataset for which symmetric non-temporal similarities between points can be defined.

Keywords: Slow Feature Analysis, Temporal Coherence, Spectral Embeddings, Deep Learning, Representation Learning

1. Introduction

Finding meaningful representations in data is a core challenge in modern machine learning as the performance in many goal-directed frameworks, such as reinforcement learning or supervised learning, is strongly influenced by the quality of the underlying representation. Usually, these representations are either domain specific, heuristically chosen, or acquired through task-specific adaptation of parameterized models. Many currently successful approaches for either deep supervised learning (Goodfellow et al. (2016)) or reinforcement learning (Sutton and Barto (1998) and Mnih et al. (2013)) fall in the latter category, as they rely on a labeled data or a reward signal to provide sufficient indication which features of the input data should be extracted to increase performance. However, in most real-world scenarios labels have to be provided by expert knowledge and reward signals are sparse and thus an inefficient driver to adapt large models. In unsupervised representation learning one typically tries to find and apply a principle, such as the minimization of reconstruction error, by which to extract meaning from data without assuming the availability of any goal-driven metrics.

We focus on the principle of temporal coherence as applied in slow feature analysis (SFA, Wiskott and Sejnowski (2002)) or regularized slowness optimization (Bengio and Bergstra (2009)) which has been shown to provide a useful proxy for extracting underlying causes from data. For example, Franzius et al. (2007) have used SFA to learn a representation from ego-visual data that sparsely encodes position, head direction, or spatial view, similar

to activities observed in rodent brains. SFA has also been successfully applied to determining object configuration (identity, position, angle) from a tabletop view of moving objects (Franzius et al. (2011)).

A graph-based generalization allows for arbitrary neighborhood-respecting embeddings and achieved (at that time) state-of-the-art age estimation results in Escalante-B. (2017). An experiment utilizing the same generalized perspective is described Section 6.3.

Until recently, the class of models and training algorithms that could be used for SFA have been limited. While the original proposal of SFA by Wiskott and Sejnowski (2002) uses non-linear basis functions as a method to introduce non-linearity to the otherwise linear model, this classical approach has two limitations: **(a)** The basis functions have to be chosen beforehand by experience or heuristics, and **(b)** as the resulting model is shallow, its expressivity tends to scale unfavorably in the dimension of the expansion (Raghu et al. (2016)) compared to hierarchical extensions discussed later. In the case of polynomial expansion, expanding to degree d on e-dimensional input data results in $\binom{d+e}{d}$-dimensional expanded data, e.g., quadratically expanding grayscale images of 180×90 pixels results in an output dimension $> 131 \cdot 10^6$. Extracting a single feature with a linear model would thus require 5.7-times more parameters than the modern and powerful *Xception* network (Chollet (2016)). A kernelized version of SFA introduced by Böhmer et al. (2011) solves this problem by implicit expansion (the *kernel trick*), but shares the typical limitations of other kernel methods, namely poor scaling in the number of training points as well as a necessity for finely-tuned regularization.

An alternative approach to increase expressivity is to apply low-degree non-linearities repeatedly in a receptive-field fashion, interlacing them with projection steps. This has been done in Hierarchical SFA (HSFA, Franzius et al. (2007) and Escalante-B. and Wiskott (2016)) and deep architectures in general. But while the latter are typically trained in an end-to-end fashion by variants of stochastic gradient-descent, HSFA is trained in a greedy layer-wise procedure, solving the linear SFA problem consecutively for each layer as closed-form solution and thereby assuming that globally slow features can be composed of (decreasingly) local slow features. Escalante-B. and Wiskott (2016) shows that this assumption is sub-optimal and can be partially relaxed by adding information by-passes to the model.

We propose an SFA-variant that approximately enforces the similar constraints while being differentiable and thereby allowing training by gradient-descent. This makes it possible to leverage the representational power of complex models, such as deep neural networks, and useful ideas from that domain to extract slow and informative features from data by optimizing a **global** slowness objective. To demonstrate the applicability of this approach, we provide three distinct experimental evaluations.

2. Related work

2.1. Slowness-based Methods

Apart from being strongly related to closed-form SFA and its variants/extensions, our algorithm is in line with recent work harnessing the temporal coherence prior (Bengio et al. (2013)) in deep, self-supervised feature learning. This can be done by having a slowness term in the loss function, but adding regularizing terms to avoid trivial (constant) solutions. For example, a reconstruction loss (Goroshin et al. (2015a)) or one-step latent code prediction

loss (Goroshin et al. (2015b)). Deep temporal coherence has also been considered via the lens of similarity metric learning, for example by optimizing a contrastive loss (Jayaraman and Grauman (2016), Mobahi et al. (2009)) or a triplet loss (Jansen et al. (2017), Wang and Gupta (2015)). These metric learning approaches manage to avoid degenerate solutions by pushing points away from each other (in feature space) that are not temporal neighbors, as opposed to our work, where informativeness is ensured by the architecture instead of the objective function.

2.2. Graph-based Methods

SFA has been generalized from temporally-coherent embeddings to respecting arbitrary similarities in *graph-based SFA* (GSFA, Escalante-B. and Wiskott (2013)) and *generalized SFA* (Sprekeler (2011)). The former adapts only the objective function, while the latter additionally generalizes the constraints to **D**-orthogonality (**D** being the degree matrix of an underlying graph). Following generalized SFA, SFA can be shown to be a special case of Laplacian Eigenmaps. The generalization of PowerSFA proposed in Section 6.3 is close to GSFA, assuming regularity in the graph for the orthogonality constraint.

Spectral Inference Networks (SpIN, Pfau et al. (2018)) utilize this connection to successfully derive a gradient-based SFA training as a special case. They are based on correcting a biased gradient when directly optimizing the Rayleigh-Quotient with respect to the model's parameters. The constraints are implicitly enforced through the loss function as opposed to directly whitening the output. Similar to our work, SpINs allow for employing any architecture to find these embeddings. However, the whitening proposed in this paper is applicable to any loss function as it is part of the model architecture and not inherently a part of the optimized objective.

SpectralNets (SN, Shaham et al. (2018)) are another closely related approach in which a differentiable approximator is trained to learn spectral embeddings used in subsequent k-means clustering. Opposed to PowerSFA and SpINs, they split a single optimization step into two parts: an ortho-normalization step based on explicitly calculating the Cholesky decomposition of the batch covariance matrix to set and freeze the weights of a linear output layer, followed by a stochastic optimization step. While this might be considered end-to-end depending on the definition of the paradigm, Shaham et al. (2018) do not indicate if and how this can be implemented as single architecture.

2.3. Other Related Approaches

In Section 6.3, we consider a generalization of PowerSFA's loss similar to GSFA and apply it to the NORB dataset (LeCun et al. (2004)). We thereby loosely follow the experimental procedure in Hadsell et al. (2006). The authors use a siamese neural network architecture (Bromley et al. (1993)) for optimizing pair-wise distances of embedded points to reflect a (provided) similarity and dissimilarity structure in the data. By including dissimilarity, they provide incentive for the optimization procedure to ensure informativeness of the embedding instead of enforcing decorrelation through constraints.

3. Slow Feature Analysis

Slow Feature Analysis (SFA) is based on the hypothesis that interesting high-dimensional streams of data that vary quickly in time are typically caused by a low number of underlying factors that vary comparably slow. Therefore, slowness can be used as a proxy criterion by which to extract meaningful representations of these low-dimensional underlying causes, even in the absence of labels.

There is strong evidence in favor of this hypothesis, as it has been shown that features extracted by SFA tend to encode highly relevant information about the data-generating environments (Franzius et al. (2011), Franzius et al. (2007)).

The notion of extracting slow, informative features from a time-series dataset can be formalized as an optimization problem. Given a time-series $\{\mathbf{x}_t\}_{t=0...N-1}$ with $\mathbf{x}_t \in \mathbb{R}^d$, sequentially find a continuous function $g : \mathbb{R}^d \to \mathbb{R}^e$ with:

$$\min_g \quad \left\langle \|g(\mathbf{x}_{t+1}) - g(\mathbf{x}_t)\|^2 \right\rangle_t \tag{1a}$$

$$\text{s.t.} \quad \left\langle g(\mathbf{x}_t) \right\rangle_t = \mathbf{0}, \tag{1b}$$

$$\left\langle g(\mathbf{x}_t)g(\mathbf{x}_t)^T \right\rangle_t = \mathbf{I}_e \tag{1c}$$

where $\left\langle \cdot \right\rangle_t$ is the time-average and $\mathbf{I}_e$ is the e-dimensional identity. Typically, an ordering of the extracted features is assumed. That is

$$\Delta(g_i) = \left\langle \|g_i(\mathbf{x}_{t+1}) - g_i(\mathbf{x}_t)\|^2 \right\rangle_t \leq \left\langle \|g_j(\mathbf{x}_{t+1}) - g_j(\mathbf{x}_t)\|^2 \right\rangle_t = \Delta(g_j)$$

for $i < j$. In this work, we mainly consider unordered features, but, if needed, the ordering can be established after optimization as briefly explained in Section 4.

The constraints ensure that each of the extracted features is informative (decorrelated to all others, equation (1c)) and non-trivial (unit variance, equation (1c)). Originally, solutions to SFA were proposed for the space of linear/affine functions g, for which a closed-form solution exists, and later in a kernelized version that requires strong regularization Böhmer et al. (2011).

For convenience, we will also refer to datasets/mini-batches in matrix notation, i.e., $\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 & \cdots & \mathbf{x}_{N-1} \end{bmatrix} \in \mathbb{R}^{d \times N}$ and by a slight abuse of notation $g(\mathbf{X}) = \begin{bmatrix} g(\mathbf{x}_0) & \cdots & g(\mathbf{x}_{N-1}) \end{bmatrix} \in \mathbb{R}^{e \times N}$.

If the data is mean-free, the covariance matrix can be calculated as $\mathbf{C} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$. Further, we assume it to be positive definite and decomposable into $\mathbf{C} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ with $\mathbf{U}$'s columns containing its eigenvectors and $\mathbf{D}$ containing the corresponding positive eigenvalues on the diagonal.

4. (Approximate) Whitening

The standard implementation of linear SFA (Zito et al. (2008)) is based on computing the closed-form solution to a generalized eigenvalue problem (Berkes and Wiskott (2005)). An equivalent approach is to first whiten the data, followed by a projection onto the minor

components of the time-series of differences $\{\dot{x}_t = x_{t+1} - x_t\}_{t=0...N-2}$. If all components are used and the eigenvectors are ordered by eigenvalues, the second part corresponds to a rotation that produces output features that are ordered by slowness. This rotation will not change the slowness performance of a solution, as defined in equation 1a, and thus we can apply it after training our model to establish an ordering, e.g., for visual presentation. We indicate when we have done so.

Whitened data has three important properties: **(a)** it is mean-free (constraint (1b)), **(b)** has unit variance if projected onto an arbitrary unit vector (constraint (1c)), and **(c)** projections onto orthonormal vectors are decorrelated (constraint (1c)).

Given a mean-free dataset $\tilde{\mathbf{X}} \in \mathbb{R}^{d \times N}$ of size N and dimension d, a corresponding whitened dataset is given by

$$\mathbf{X} = \mathcal{W}(\tilde{\mathbf{X}}) = \mathbf{W}\tilde{\mathbf{X}}$$

where $\mathbf{W} = \mathbf{U}\mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T$ is called the whitening matrix. As $\mathbf{C}$ is assumed to be positive definite, $\mathbf{D}^{-\frac{1}{2}}$ is well-defined.

One widely used method to calculate an eigen-decomposition is **power iteration**. It utilizes the fact that repeatedly applying

$$\mathbf{u}^{[i+1]} = \frac{\mathbf{C}\mathbf{u}^{[i]}}{\lambda^{[i]}} \qquad \text{with} \qquad \lambda^{[i]} = \|\mathbf{C}\mathbf{u}^{[i]}\|$$

will converge to the eigenvector $\mathbf{u}$ corresponding to the largest eigenvalue λ of the matrix $\mathbf{C}$ for a random vector $\mathbf{u}^{[0]} \in_R \mathbb{R}^d$. Subsequently, the spectral component corresponding to this eigenvector can be removed as

$$\mathbf{C} \leftarrow \mathbf{C} - \lambda\mathbf{u}\mathbf{u}^T.$$

allowing to extract all eigenvector/-value pairs $(\lambda_j, \mathbf{u}_j)$ in descending eigenvalue order by repeating the procedure.

The corresponding whitening matrix $\mathbf{W}$ can be calculated as:

$$\mathbf{W} = \sum_{j=0}^{d-1} \frac{1}{\sqrt{\lambda_j}}\mathbf{u}_j\mathbf{u}_j^T$$

We used a fixed number of power iterations which resulted in sufficient whitening without explicitly checking for convergence (see Section 6.1). Furthermore, the full decomposition is based on differentiable operations. Thus, the whitened output $\mathbf{X}$ is differentiable with respect to the input $\tilde{\mathbf{X}}$.

5. Gradient-based Slow Feature Analysis

The key idea for gradient-based SFA is that such a whitening layer can be applied to any differentiable architecture (such as deep neural networks) to enforce outputs that approximately obey the SFA constraints, while the architecture stays differentiable. Therefore, it can be trained using gradient-descent, allowing for hierarchical architectures in which every

parameter is modified iteratively towards optimizing a **global** slowness objective, as opposed to assuming a local-to-global slowness as the greedy-training of HSFA. To formalize, if

$$\tilde{g}_\theta : \quad \mathbb{R}^d \to \mathbb{R}^e$$

is a differentiable function approximator, such as a neural network, parameterized by θ and

$$\mathcal{W} : \quad \mathbb{R}^{N \times e} \to \mathbb{R}^{N \times e}$$

denotes the approximate whitening procedure, then the output

$$\mathbf{Y} = \mathcal{W}(\mathbf{H}) \qquad \text{with} \qquad \mathbf{H} = \tilde{g}_\theta(\mathbf{X})$$

approximately obeys the SFA constraints for a given dataset or mini-batch $\mathbf{X} \in \mathbb{R}^{d \times N}$. We are using the slight abuse of notation introduced in Section 3 for $\mathbf{X}$, $\mathbf{H}$ and $\mathbf{Y}$

The approximation error is defined by a general differentiable loss function as

$$E = L_\mathcal{S}(\mathbf{Y}) = \frac{1}{N} \sum_i \sum_j s_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \tag{2}$$

where s_{ij} is the similarity between two points x_i and x_j comparable to weights in spectral graph embeddings (cf. Sprekeler (2011) and Escalante-B. and Wiskott (2013)).

For optimizing slowness, the similarity is provided by temporal proximity and can be written as the Kronecker delta

$$s_{ij} = \delta_{i,j+1}$$

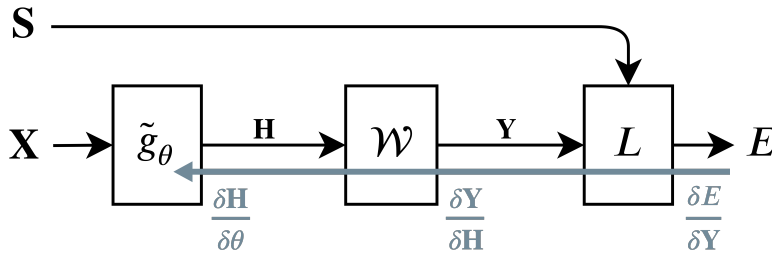thereby connecting consecutive steps in the time-series.

Figure 1: An illustration of the overall architecture given processing a dataset $\mathbf{X}$ with similarity $\mathbf{S}$. The θ are the only trainable parameters and are learned end-to-end via error back-propagation.

The approximator can be trained to minimize eq. 2 by following the negative gradient (or an estimate) of E with respect to θ, $-\nabla_\theta E$. As $\mathcal{W}$ is part of the architecture, the idea is that $\tilde{g}_\theta$ learns to produce features that minimize the loss *after* being whitened. For our experiments, the whitening is implemented as a layer in the widely-used *Keras*-package (Chollet et al. (2015)) and utilizes the underlying reverse-mode automatic-differentiation of *Tensorflow* (Abadi et al. (2015)). This allows us to conveniently train a network through the whitening layer without explicitly formalizing the gradient of the whitening operation.

Note, that the training is *end-to-end* as the optimization procedure does not have distinct phases, except the typical forward-pass/backward-pass as illustrated in Figure 1.

In our experiments, we used the ADAM optimization algorithm (Kingma and Ba (2014)) with Nesterov-accelerated momentum (Dozat (2015)) to train the $\tilde{g}_\theta$.

6. Experiments

6.1. Synthetic Trigonometric Data

To show the general feasibility of this approach, we first demonstrate that PowerSFA finds near-optimal solutions in the linear case for synthetic data. Standard SFA is a linear method that relies on **(a)** non-linear basis function expansion and **(b)** hierarchical processing to induce non-linearity. This means that PowerSFA could hypothetically be used in a similar fashion (even though a gradient-based approach allows for more complex models in a natural way).
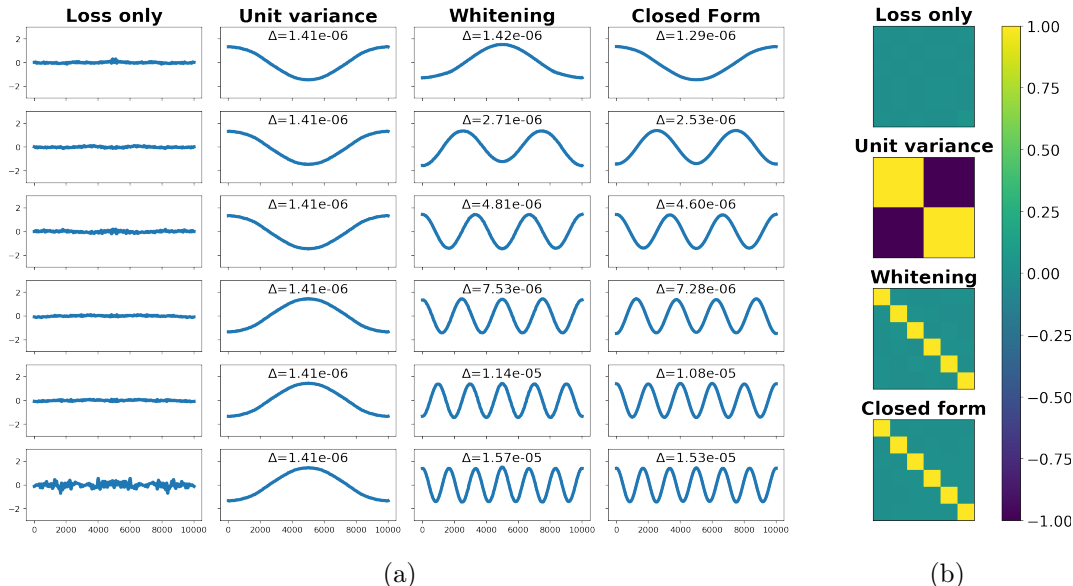

Figure 2: Six slow features learned by a linear model with different constraints: (a) shows the individual features found for different constraints, with Δ-values as defined in 3, and (b) shows the corresponding covariance matrices. Approximate whitening and the closed-form solution find decorrelated, informative output signals with comparable slowness.

The data is generated by trigonometric polynomials of degree M as:

$$\mathbf{x}(t) = \boldsymbol{\varepsilon}_t + \sum_{m=1}^{M} \boldsymbol{\alpha}_n \cos{(mt)}$$

with $\mathbf{x}, \boldsymbol{\varepsilon}, \boldsymbol{\alpha} \in \mathbb{R}^D$ and coefficients $\alpha_{in} \sim \mathcal{N}(0,1)$. A noise term $\varepsilon_{it} \sim \mathcal{N}(0,0.01)$ is added to avoid numerical instabilities in the implementation of closed-form SFA we compared our

method to, as singular covariance matrices can cause the underlying eigendecomposition to break.

We implemented a temporal step-size of $\frac{2\pi}{10^4}$, and generated 10000 steps with degree $M = 100$ and dimension $D = 500$. The data is whitened with 100 power iterations.

Figure 2a shows the extracted features for different variants of SFA. If the slowness loss is optimized without any constraints on the output features, the optimal solution is to collapse all signals to a noisy constant ($\Delta \approx 0$), while if unit variance is enforced, only the features with slowness very close to the smallest Δ are extracted multiple times and thus the representation becomes highly redundant across features. When using the approximate whitening, the quality of the solutions is comparable to the optimal solution gained by solving a generalized eigenvalue problem. For visualization purposes, the features were ordered by slowness by rotation as explained in Section 4. In Figure 2b, the covariance matrices of the extracted signals are visualized.

Note that it is not a sensible approach to use PowerSFA to optimize a linear model for such low-dimensional data as the optimal solution is easily attainable. For this reason, this experiment should be understood as a proof-of-concept and a general demonstration of applicability rather than a recommendation for a use-case.

A very low number of power iterations will render the optimization unstable, but here seems to be no continuous trade-off between correlation and slowness mediated by the number of iterations. Thus, we recommend to find a minimal setting that allows for stable optimization. Supplement S1 contains a small hyper-parameter study illustrating that behavior for the linear case.

To provide evidence on how gradient-based SFA can improve on solutions found by closed-form SFA, we encapsulate the original signal in a component-wise non-linear distortion:

$$\mathbf{u}(t) = \cos(e^{\mathbf{x}(t)})$$

This makes it impossible to optimally extract slow signals by linearly unmixing the original components.

When applying closed-form SFA to non-linear problems, it is common to apply multiple low-degree polynomial expansions interlaced with linear SFA steps to reduce the dimensionality, in a greedy layer-wise training. We use an architecture with three quadratic expansion layers (normalized to unit norm to avoid exploding gradients), calculate the greedy closed-form solution and then use this solution as initialization for gradient-based training. We perform the same comparison for a multi-layer perceptron with three hidden layers and *tanh* activation since polynomial expansion functions are uncommon in gradient-trained models. Both architectures are provided in supplement S2.

	Closed-form	Gradient-based
Slowness$_{\text{quadratic}}$	$1.53 \cdot 10^{-3} \pm 9.00 \cdot 10^{-5}$	$3.35 \cdot 10^{-4} \pm 4.43 \cdot 10^{-5}$
Slowness$_{\text{tanh}}$	$1.84 \cdot 10^{-1} \pm 5.96 \cdot 10^{-3}$	$8.936 \cdot 10^{-4} \pm 5.955 \cdot 10^{-5}$

Table 1: Average slowness of five output features over five runs extracted by greedy layer-wise training and gradient-based training from non-linearly distorted cosine polynomials. Results for three-layer quadratic expansion network and neural network with *tanh* activation.

The results in Table 1 show that gradient-based training can significantly improve slowness in multi-layer architectures compared to greedy layer-wise SFA. Note that increasing the output dimension of the dimensionality reduction steps (or dropping them altogether) in the quadratic expansion network will, unsurprisingly, lead to improved performance and ultimately to convergence to similar optima in both networks. However, this performance increase comes at the cost of high memory requirements and is usually not applicable in high-dimensional non-synthetic problems.

6.2. Place Cells from Visual Data

Slowness has been shown to be a useful principle for extracting underlying causes from high-dimensional data. Franzius et al. (2007) used SFA to learn representations from visual time-series: Depending only on the movement statistics in a 3D environment, they were able to extract low-dimensional encodings of either position, direction, or viewpoint from 320° ego-perspective images. When additional sparse coding in the form of ICA is applied, these features become distinctly localized and each one encodes for a particular neighborhood in space, direction angle, or viewpoints - similar to cell types discovered in the hippocampal formation of the mammalian brain (O'Keefe and Dostrovsky (1971), Rolls (1999) and Barry and Burgess (2014)).

We reproduced one such experiment, but replaced HSFA with PowerSFA using an untrained *MobileNet* architecture (Sandler et al. (2018)) as convolutional network model. The *Unity* game engine (Unity Technologies (2019)) was used to generate visual data with a more narrow 90° field-of-view. Figure 3 shows the layout of the environment and samples from different poses.
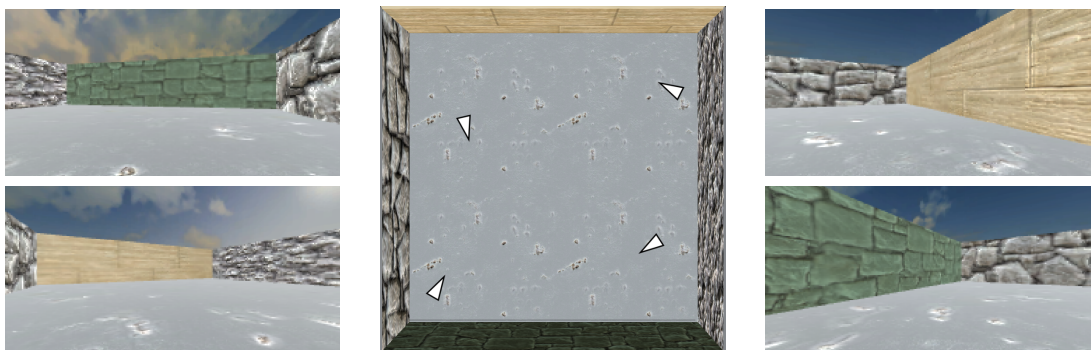

Figure 3: The data-generating box environment. Each wall has a different texture so that they can be distinguished from each other when the agent is standing close.

From this environment, we generated a 2.5×10^5-step time-series with 240×120 RGB-pixels to train the SFA network. This time-series is driven by two independent stochastic processes - one generating a movement sequence and an other one generating a sequence of rotations in parallel. In each step, position and rotation are used to generate an ego-perspective image (Figure 3). When the position changes more slowly than the rotation, the former will be encoded in the slowest features of a sufficiently powerful model and vice versa, leading to either a low-dimensional representation of place or orientation. The details of this procedure can be found in Franzius et al. (2007).

In line with their experiments, we applied independent component analysis (ICA, Comon (1994)) after training to acquire a sparse representation. Figure 4 shows the activation of the individual slow/sparse features at each position in the environment. Every feature is averaged over a full rotation in 16 steps for a compact presentation. The slow features have been ordered by slowness through a rotation as explained in Section 4 for visualization and easier comparison to the responses found in Franzius et al. (2007).
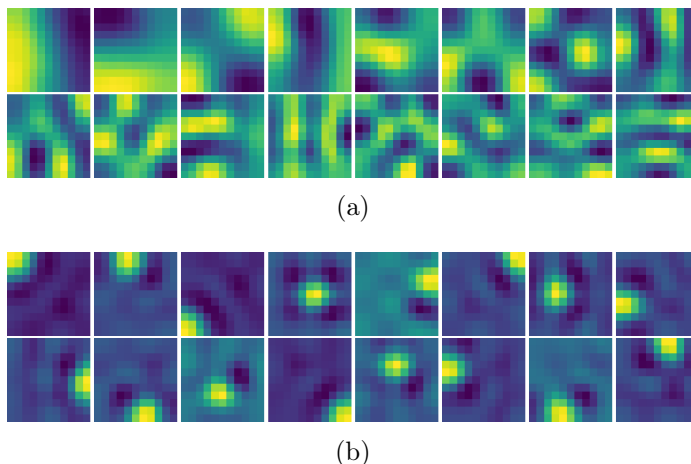

(a)

(b)

Figure 4: The activation of the 16 output features for all positions in the environment, averaged over 16 equiangular directions. 4a shows the slow features (after subsequent ordering) and 4b shows the sparse features. Each sparse feature is active in a distinct part of the environment and mainly inactive in all other parts.

The features for all positions in each of the directions can be found in the supplementary material to justify the averaging in Figure4. Representative for all features, Figure 5 shows the activation map of the first sparse feature under rotation. All features are largely invariant to rotation of the agent, despite its narrow field of view, and clearly encode a specific location, independent of the current viewing direction, resembling place-cells in the mammalian Hippocampus (O'Keefe and Dostrovsky (1971)).
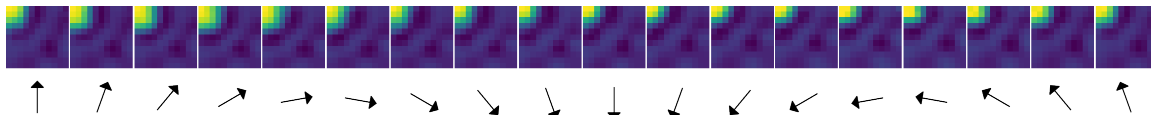

Figure 5: The first sparse feature under $20°$ rotations. Location is encoded independent of the current view.

Such a specific, low-dimensional representation is likely to be an appropriate foundation for faster goal-directed learning, for example, when locating rewards in a connected environment or general navigation tasks. Furthermore, as the architecture is a CNN as often used in modern reinforcement learning Mnih et al. (2013), the trained network (without the whitening layer) might also be further utilized as efficient initialization for subsequent training in such a setting. Neither assumption is tested in this work and both should be considered promising hypothetical benefits although not far-fetched.

6.3. NORB

While gradient-based SFA is the main contribution of this work, previous work on generalizing SFA (Sprekeler (2011)) has shown that the SFA optimization problem is strongly related to a special case of the problem solved by a more general spectral embedding method, i.e., Laplacian Eigenmaps (Belkin and Niyogi (2003)), or, with small differences, graph-based SFA (Escalante-B. and Wiskott (2013)). For this reason, we defined a more general loss function in equation 2.
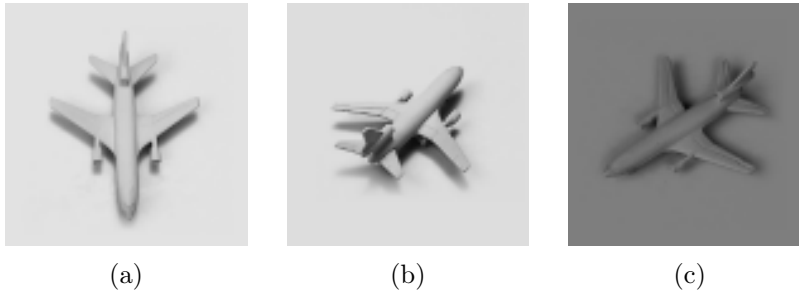

(a)   (b)   (c)

Figure 6: The embedded object from the NORB dataset. Samples differ in azimuth, elevation and lighting.

Similarities s_{ij} can be defined between any two points x_i and x_j of a given data-set, not just consecutive ones, thus allowing to optimize neighborhood-respecting embeddings for general graphs. This is similar (but not fully equivalent) to spectral embeddings on graph data, as used in algorithms such as Laplacian eigenmaps. Our approach exhibits four significant differences:

1. Features do not have an order,

2. the found solution is a local optimum (or saddle point),

3. like in GSFA, the graph regularity is ignored when enforcing the orthogonality constraint, and

4. it allows for a natural and scalable way to embed unseen points.

Computing an out-of-sample embedding is no different than embedding a training point, except for a frozen whitening matrix. In particular, its complexity does not scale with the number of training points used as is often the case in other out-of-sample schemes, such as Nyström approximation (Williams and Seeger (2001)).

We demonstrate the usefulness of such an approach by embedding an object of the NORB data-set, a collection of photographs of toys taken at different elevations, azimuths, and under different lighting conditions with the *MobileNet* architecture (Sandler et al. (2018)) scaled with $\alpha = 0.5$ and a depth multiplier of 2 in the *Keras* implementation. Following Hadsell et al. (2006), we embed a toy plane (Figure 6) in 972 configurations (18 azimuths × 6 lighting conditions × 9 elevations angles) that were randomly split into a train- and test-set of sizes 660 and 312 ,respectively. The similarities s_{ij} were chosen as 1 if x_i and x_j
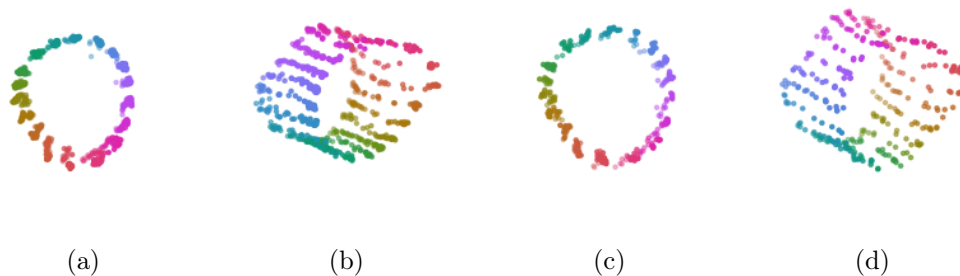
(a) (b) (c) (d)

Figure 7: Cylindrical embedding of NORB plane with azimuth colored. 7a and 7b show the embedded training data from the front and the side of the cylinder respectively, while 7c and 7d show the test data for the same configuration. Circumference of the cylinder encodes the rotation of the plane.
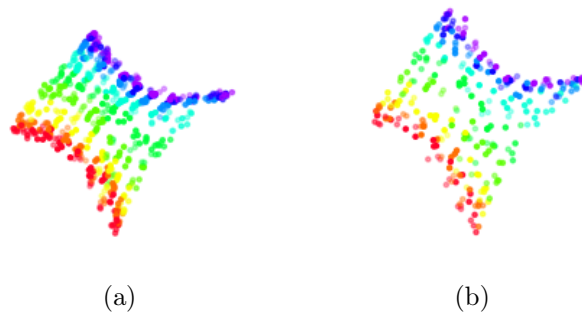

(a) (b)

Figure 8: Embedding from Figure 7 with elevation colored. 8a shows the embedded training data from the side of the cylinder, while 8b shows the test data for the same configuration. Height on the cylinder encodes the photograph's elevation angle (from low to high when moving through it).

differed only in one step either in rotation (i.e., one azimuth) or elevation (i.e., one level) and 0 otherwise. The s_{ij} were independent of lighting condition.

Figures 7 and 8 show the three-dimensional embedding that was found in this setting. The data was embedded in a cylindrical shape in which the circumference encodes the rotation angle of the embedded object, and the length along the cylinder encodes the elevation configuration of the object for the train-set and in the out-of-sample case of the test-set. Hadsell et al. (2006) found a similar cylindrical encoding but include a maximization of distance for dissimilar samples in the objective function instead of a decorrelation constraint.

7. Discussion

We propose a new way of extracting informative slow features from quickly varying inputs based on differentiable whitening of processed batches of the input data. To experimentally show the feasibility of the method, we trained a linear and two non-linear models to extract slowly varying output signals from synthetic time-series by gradient-descent, and find that the differentiable whitening ensures informativeness of the extracted features when

optimizing a slowness loss function by gradient-descent. Furthermore, the linear features corresponded closely to the optimal solutions found by closed-form SFA. [1]

To show applicability to visual data, we trained a CNN on an input stream of an ego-perspective time-series generated from a 3D environment. Gradient-based SFA could successfully extract the slowly changing position in the environment invariant to the quickly changing rotation from the images alone, despite a narrow field-of-view. By subsequently applying ICA, we were able to show that the place-cell phenomenon that has been found in the mammalian Hippocampus and that has been successfully modelled by HSFA could also be acquired with deep neural networks by the means of our method. We consider evaluating the efficacy of either directly using this representation or using it as model initialization for goal-directed learning a promising direction for future research, in particular, when concerned with navigation tasks.

The last experiment was conducted on a non-time-series image dataset on which symmetric similarity relations between the data are defined. We show that by the generalization of gradient-based SFA in the spirit of graph-based SFA is able to extract a low-dimensional representation that preserves and disentangles the configuration parameters used to define the similarity, in this case, azimuth and elevation of a photographed toy. This representation generalizes well to previously unseen configurations of the object.

While the algorithm is still in a prototypical state, the proof-of-concept results presented in this paper show promise for gradient-based SFA by differentiable whitening to extract meaningful representations for goal-oriented learning while leveraging the expressive power of modern architectures. Futhermore, differentiable whitening ensures non-redundancy of output features regardless of the optimized loss function and can be used for other unsupervised principles as well.

One limitation of PowerSFA is that it currently does not scale favorably in the number of output features e. We see two main reasons for this: the necessary batch size to get a meaningful estimate of the batch-covariance estimate and its calculation. The latter is due to the complexity of a naive $\mathbb{R}^{e \times N_{\text{batch}}} \cdot \mathbb{R}^{N_{\text{batch}} \times e}$ matrix-multiplication being $\mathcal{O}(N_{\text{batch}} e^2)$, while the former is due to the whitening procedure expecting a covariance matrix of full rank and thus $N_{\text{batch}} \geq e$ samples.

To reduce the lower bound on the batch-size at batch t, a convex mixture with the covariance matrix of the previous

$$\mathbf{C}_t = (1 - \gamma)\mathbf{C}_\theta + \gamma\mathbf{C}_{t-1}$$

Note, that only the current batch's covariance matrix $\mathbf{C}_\theta$ is considered parameter-dependent and allows to propagate a gradient for training. Thus, large values for γ might cause a significant bias in the gradient-estimate. This has not been used to generate the proof-of-concept results for this paper.

---

1. We put the *Keras* implementation of PowerSFA as well as implementations for qualitatively reproducing the synthetic data and NORB experiments online at https://s3.amazonaws.com/acml2019/powersfa_code.zip.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

C. Barry and N. Burgess. Neural mechanisms of self-location. *Current Biology*, 24(8):R330 – R339, 2014. ISSN 0960-9822. doi: 10.1016/j.cub.2014.02.049.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, June 2003. ISSN 0899-7667. doi: 10.1162/089976603321780317.

Yoshua Bengio and James S. Bergstra. Slow, decorrelated features for pretraining complex cell-like networks. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 99–107. Curran Associates, Inc., 2009.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Pietro Berkes and Laurenz Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, 5(6), 2005.

Wendelin Böhmer, Steffen Grünewälder, Hannes Nickisch, and Klaus Obermayer. Regularized sparse kernel slow feature analysis. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 235–248, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23780-5.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

Francois Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

Francois Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3): 287 – 314, 1994. ISSN 0165-1684. Higher Order Statistics.

Timothy Dozat. Incorporating nesterov momentum into adam. http://cs229.stanford.edu/proj2015/054_report.pdf, 2015.

Alberto N. Escalante-B. *Extensions of Hierarchical Slow Feature Analysis for Efficient Classification and Regression on High-Dimensional Data*. PhD thesis, Ruhr-University Bochum, 2017.

Alberto N. Escalante-B. and Laurenz Wiskott. How to solve classification and regression problems on high-dimensional data with a supervised extension of slow feature analysis. *J. Mach. Learn. Res.*, 14(1):3683–3719, December 2013. ISSN 1532-4435.

Alberto N. Escalante-B. and Laurenz Wiskott. Improved graph-based SFA: information preservation complements the slowness principle. *CoRR*, abs/1601.03945, 2016.

Mathias Franzius, Henning Sprekeler, and Laurenz Wiskott. Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLOS Computational Biology*, 3(8):1–18, 08 2007.

Mathias Franzius, Niko Wilbert, and Laurenz Wiskott. Invariant object recognition and pose estimation with slow feature analysis. *Neural Computation*, 23(9):2289–2323, 2011. PMID: 21671784.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised feature learning from temporal data. *arXiv preprint arXiv:1504.02518*, 2015a.

Ross Goroshin, Michael F Mathieu, and Yann LeCun. Learning to linearize under uncertainty. In *Advances in Neural Information Processing Systems*, pages 1234–1242, 2015b.

Raia Hadsell, Sumit Chopra, and Yann Lecun. Dimensionality reduction by learning an invariant mapping. In *In Proc. Computer Vision and Pattern Recognition Conference (CVPR'06)*. IEEE Press, 2006.

Aren Jansen, Manoj Plakal, Ratheet Pandya, Daniel PW Ellis, Shawn Hershey, Jiayang Liu, R Channing Moore, and Rif A Saurous. Unsupervised learning of semantic audio representations. *arXiv preprint arXiv:1711.02209*, 2017.

Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3852–3861, 2016.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'04, pages 97–104. IEEE Computer Society, 2004.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744. ACM, 2009.

J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171 – 175, 1971. ISSN 0006-8993. doi: https://doi.org/10.1016/0006-8993(71)90358-1.

D. Pfau, S. Petersen, A. Agarwal, D. Barrett, and K. Stachenfeld. Spectral Inference Networks: Unifying Spectral Methods With Deep Learning. *ArXiv e-prints*, June 2018.

M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein. On the Expressive Power of Deep Neural Networks. *ArXiv e-prints*, June 2016.

Edmund T. Rolls. Spatial view cells and the representation of place in the primate hippocampus. *Hippocampus*, 9(4):467–480, 1999.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.

Uri Shaham, Kelly Stanton, Henry Li, Ronen Basri, Boaz Nadler, and Yuval Kluger. Spectralnet: Spectral clustering using deep neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJ_aoCyRZ.

Henning Sprekeler. On the Relation of Slow Feature Analysis and Laplacian Eigenmaps. *Neural Computation*, 23(12):3287–3302, December 2011. ISSN 0899-7667.

Richard S. Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Unity Technologies. Unity game engine, 2019. URL https://docs.unity3d.com/Manual/UnityManual.html.

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

Laurenz Wiskott and Terrence Sejnowski. Slow feature analysis: unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

T. Zito, N. Wilbert, L. Wiskott, and P. Berkes. Modular toolkit for data processing (mdp): a python data processing framework. *Frontiers in Neuroinformatics*, 2:8, 2008.