Autoencoding any Data through Kernel Autoencoders

Pierre Laforgue* **Stephan Clémençon*** **Florence d'Alché-Buc***
*LTCI, Télécom ParisTech, Université Paris Saclay, Paris, France

Abstract

This paper investigates a novel algorithmic approach to data representation based on kernel methods. Assuming that the observations lie in a Hilbert space $\mathcal{X}$, the introduced Kernel Autoencoder (KAE) is the composition of mappings from vector-valued Reproducing Kernel Hilbert Spaces (vv-RKHSs) that minimizes the expected reconstruction error. Beyond a first extension of the autoencoding scheme to possibly infinite dimensional Hilbert spaces, KAE further allows to autoencode any kind of data by choosing $\mathcal{X}$ to be itself a RKHS. A theoretical analysis of the model is carried out, providing a generalization bound, and shedding light on its connection with Kernel Principal Component Analysis. The proposed algorithms are then detailed at length: they crucially rely on the form taken by the minimizers, revealed by a dedicated Representer Theorem. Finally, numerical experiments on both simulated data and real labeled graphs (molecules) provide empirical evidence of the KAE performances.

1 INTRODUCTION

As experienced by any practitioner, data representation is critical to the application of Machine Learning, whatever the targeted task, supervised or unsupervised. An answer to this issue consists in feature engineering, a step that requires time-consuming interactions with domain experts. To overcome these limitations, Representation Learning (RL) (Bengio et al., 2013) aims at building automatically new features in an unsupervised fashion. Recent applications to neural nets pre-training, image denoising and semantic hashing have renewed a strong interest in RL, now a proper research field. Among successful RL approaches, mention has to be made of Autoencoders (AEs) (Vincent et al., 2010), and their generative variant, Deep Boltzman Machines (Salakhutdinov and Hinton, 2009).

AEs attempt to learn a pair of encoding/decoding functions under structural constraints so as to capture the most important properties of the data (Alain and Bengio, 2014). If they have mostly been studied under the angle of neural networks (Baldi, 2012) and deep architectures (Vincent et al., 2010), the concepts underlying AEs are very general and go beyond neural implementations. In this work, we develop a general framework inspired from AEs, and based on Operator-Valued Kernels (OVKs) (Senkene and Tempel'man, 1973) and vector-valued Reproducing Kernel Hilbert Spaces (vv-RKHSs). Mainly developed for supervised learning, OVKs provide a nonparametric way to tackle complex output prediction problems (Álvarez et al., 2012), including multi-task regression, structured output prediction (Brouard et al., 2016b), or functional regression (Kadri et al., 2016). This work is a first contribution to combine OVKs with AEs, enlarging the latters' applicability scope - so far restricted to $\mathbb{R}^d$ - to any data described by a similarity matrix.

We start from the simplest formulation in which a Kernel Autoencoder (KAE) is a pair of encoding/decoding functions lying in two different vv-RKHSs, and whose composition approximates the identity function. This approach is further extended to a general framework involving the composition of an arbitrary number of mappings, defined and valued on Hilbert spaces. A crucial application of KAEs arises if the input space is itself a RKHS: it allows to perform autoencoding on any type of data, by first mapping it to the RKHS, and then applying a KAE. The solutions computation, even in infinite dimensional spaces, is made possible by a Representer Theorem and the use of the kernel trick. This unlocks new applications on structured objects for which feature vectors are missing or too complex (*e.g.* in chemoinformatics).

Kernelizing an AE criterion has also been proposed by Gholami and Hajisami (2016). But their approach differs from ours in many key aspects: it is restricted

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

to AEs with 2 layers and composed of linear maps only; it relies on semi-supervised information; it comes with no theoretical analysis, and within a hashing perspective solely. Despite a similar title, the work by Kampffmeyer et al. (2017) has no connection with ours. It uses standard AEs, and regularize the learning by aligning the latent code with some predetermined kernel. In the experimental section, we implement autoencoding on graphs, which cannot be done by means of standard AEs. Graph AEs (Kipf and Welling, 2016) do not autoencode graphs, but $\mathbb{R}^d$ points with an additive graph characterizing the data structure.

The rest of the article is structured as follows. The novel kernel-based framework for RL is detailed in Section 2. A generalization bound and a strong connection with Kernel PCA are established in Section 3, whereas Section 4 describes the algorithmic approach based on a Representer Theorem. Illustrative numerical experiments are displayed in Section 5, while concluding remarks are collected in Section 6. Finally, technical details are deferred to the Appendix.

2 THE KERNEL AUTOENCODER

In this section, we introduce a general framework for building AEs based on vv-RKHSs. Here and throughout the paper, the set of bounded linear operators mapping a vector space E to itself is denoted by $\mathcal{L}(E)$, and the set of mappings from a set A to an ensemble B by $\mathcal{F}(A, B)$. The adjoint of an operator M is denoted by M^*. Finally, $[\![n]\!]$ denotes the set $\{1, \dots, n\}$ for any integer $n \in \mathbb{N}^*$.

2.1 Background on vv-RKHSs

Vv-RKHSs allow to cope with the approximation of functions from an input set $\mathcal{X}$ to some output Hilbert space $\mathcal{Y}$ (Senkene and Tempel'man, 1973; Caponnetto et al., 2008). Vv-RKHS can be defined from an OVK, which extends the classic notion of positive definite kernel. An OVK is a function $K\colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$, that satisfies the following two properties:

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X}, \qquad K(x, x') = K(x', x)^*,$$

and $\forall n \in \mathbb{N}^*, \forall\, \{(x_i, y_i)\}_{1 \le i \le n} \in (\mathcal{X} \times \mathcal{Y})^n$,

$$\sum_{1 \le i,j \le n} \langle y_i, K(x_i, x_j) y_j \rangle_{\mathcal{Y}} \ge 0.$$

A simple example of OVK is the *separable kernel* such that: $\forall\ (x, x') \in \mathcal{X} \times \mathcal{X}$, $K(x, x') = k(x, x')A$, where k is a positive definite scalar-valued kernel, and A is a positive semi-definite operator on $\mathcal{Y}$. Its relevance for multi-task learning has been highlighted for instance by Micchelli and Pontil (2005).

Let K be an OVK, and for $x \in \mathcal{X}$, let $K_x\colon y \in \mathcal{Y} \mapsto K_x y \in \mathcal{F}(X, Y)$ the linear operator such that:

$$\forall x' \in \mathcal{X},\ (K_x y)(x') = K(x', x) y.$$

Then, there is a unique Hilbert space $\mathcal{H}_K \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ called the vv-RKHS associated to K, with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_K}$ and norm $\|\cdot\|_{\mathcal{H}_K}$, such that $\forall x \in \mathcal{X}$:

- K_x spans the space $\mathcal{H}_K$ ($\forall y \in \mathcal{Y}\colon K_x y \in \mathcal{H}_K$)
- K_x is bounded for the uniform norm
- $\forall f \in \mathcal{H},\ f(x) = K_x^* f$ (*i.e.* reproducing property)

2.2 Input Output Kernel Regression

Now, let us assume that the output space $\mathcal{Y}$ is chosen itself as a RKHS, say $\mathcal{H}$, associated to the positive definite scalar-valued kernel $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, with $\mathcal{Z}$ a non-empty set. Working in the vv-RKHS $\mathcal{H}_K$ associated to an OVK $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{H})$ opens the door to a large family of learning tasks where the output set $\mathcal{Z}$ can be a set of complex objects such as nodes in a graph, graphs (Brouard et al., 2016a) or functions (Kadri et al., 2016). Following the work of Brouard et al. (2016b), we refer to these methods as Input Output Kernel Regression (IOKR). IOKR has been shown to be of special interest in case of Ridge Regression, where closed-form solutions are available besides classical gradient descent algorithms. Note that in a general supervised setting, learning a function $f \in \mathcal{H}_K$ is not sufficient to provide a prediction in the output set, and a pre-image problem has to be solved. In sections 2.5 and 4.3, a similar idea is applied at the last layer of our KAE, allowing for auto-encoding non-vectorial data while avoiding complex pre-image problems.

2.3 The 2-layer Kernel Autoencoder (KAE)

Let $S = (x_1, \dots, x_n)$ denote a sample of n independent realizations of a random vector X, valued in a separable Hilbert space $(\mathcal{X}_0, \|\cdot\|_{\mathcal{X}_0})$ with unknown probability distribution P, and such that there exists $M < +\infty, \|X\|_{\mathcal{X}_0} \le M$ almost surely. On the basis of the training sample S, we are interested in constructing a pair of encoding/decoding mappings $(f : \mathcal{X}_0 \to \mathcal{X}_1,\ g : \mathcal{X}_1 \to \mathcal{X}_0)$, where $(\mathcal{X}_1, \|\cdot\|_{\mathcal{X}_1})$ is the (Hilbert) *representation space*. Just as for standard AEs, we regard as good internal representations the ones that allow for an accurate recovery of the original information in expectation. The problem to be solved states as follows:

$$\min_{\substack{(f,g) \in \mathcal{H}_1 \times \mathcal{H}_2 \\ \|f\|_{\mathcal{H}_1} \le s,\ \|g\|_{\mathcal{H}_2} \le t}} \epsilon(f, g) \coloneqq \mathbb{E}_{X \sim P} \|X - g \circ f(X)\|_{\mathcal{X}_0}^2, \tag{1}$$

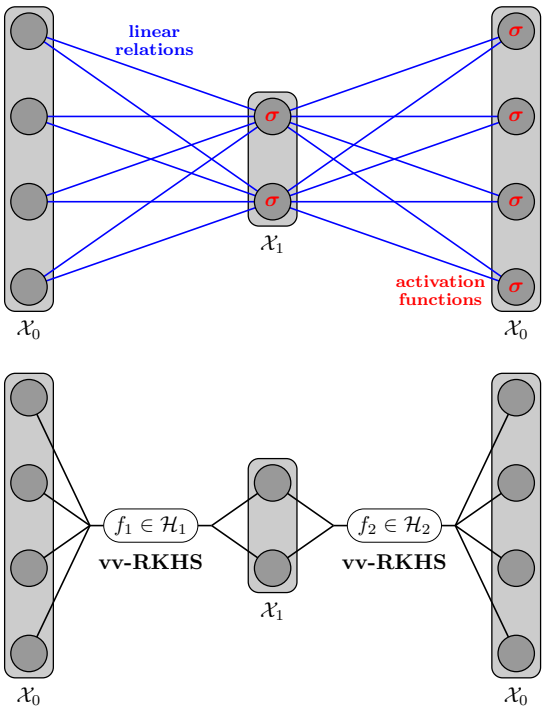


Figure 1: Standard and Kernel 2-layer Autoencoders

where $\mathcal{H}_1$ and $\mathcal{H}_2$ are two vv-RKHSs, and s and t two positive constants. $\mathcal{H}_1$ is associated to an OVK $\mathcal{K}_1 : \mathcal{X}_0 \times \mathcal{X}_0 \to \mathcal{L}(\mathcal{X}_1)$, while $\mathcal{H}_2$ is associated to $\mathcal{K}_2 : \mathcal{X}_1 \times \mathcal{X}_1 \to \mathcal{L}(\mathcal{X}_0)$. Figure 1 illustrates the parallel and differences between standard and kernel 2-layer Autoencoders.

Following the Empirical Risk Minimization (ERM) paradigm, the true risk (1) is replaced by its empirical version

$$\hat{\epsilon}_n(f, g) \coloneqq \frac{1}{n} \sum_{i=1}^{n} ||x_i - g \circ f(x_i)||^2_{\mathcal{X}_0},$$

and a penalty term $\Omega(f, g) \coloneqq \lambda \|f\|^2_{\mathcal{H}_1} + \mu \|g\|^2_{\mathcal{H}_2}$ is added instead of the norm constraints (see Theorem 1). Solutions to the following regularized ERM problem shall be referred to as *2-layer KAE*:

$$\min_{(f,g) \in \mathcal{H}_1 \times \mathcal{H}_2} \hat{\epsilon}_n(f, g) + \Omega(f, g). \tag{2}$$

2.4 The Multi-layer KAE

Like for standard AEs, the model previously described can be directly extended to more than 2 layers. Let $L \geq 3$, and consider a collection of Hilbert spaces $\mathcal{X}_0, \ldots, \mathcal{X}_L$, with $\mathcal{X}_L = \mathcal{X}_0$. For $0 \leq l \leq L-1$, the space $\mathcal{X}_l$ is supposed to be endowed with an OVK $\mathcal{K}_{l+1} : \mathcal{X}_l \times \mathcal{X}_l \to \mathcal{L}(\mathcal{X}_{l+1})$, associated to a vv-RKHS $\mathcal{H}_{l+1} \subset \mathcal{F}(\mathcal{X}_l, \mathcal{X}_{l+1})$. We then want to minimize $\epsilon(f_1, \ldots, f_L)$ over $\prod_{l=1}^{L} \mathcal{H}_l$. Setting $\Omega(f_1, \ldots, f_L) \coloneqq \sum_{l=1}^{L} \lambda_l \|f_l\|^2_{\mathcal{H}_L}$ allows for a direct extension of (2):

$$\min_{f_l \in \mathcal{H}_l} \frac{1}{n} \sum_{i=1}^{n} \|x_i - f_L \circ \ldots \circ f_1(x_i)\|^2_{\mathcal{X}_0} + \sum_{l=1}^{L} \lambda_l \|f_l\|^2_{\mathcal{H}_l}. \tag{3}$$

2.5 The General Hilbert KAE and the K^{2}AE

So far, and up to the regularization term, the main difference between standard and kernel AEs is the function space on which the reconstruction criterion is optimized: respectively neural functions or RKHS ones. But what should also be highlighted is that RKHS functions are valued in general Hilbert spaces, while neural functions are restricted to $\mathbb{R}^d$. As shall be seen in section 4.3, this enables KAEs to handle data from infinite dimensional Hilbert spaces (*e.g.* function spaces), what standard AEs are unable to do. To our knowledge, this first extension of the autoencoding scheme is novel.

But even more interesting is the possible extension when the input/output Hilbert space is chosen to be itself a RKHS. Indeed, let $\mathcal{X}_0$ denote now any set (without the Hilbert assumption). In the spirit of IOKR, let us first map $x \in \mathcal{X}_0$ to the RKHS $\mathcal{H}$ associated to some scalar kernel k, and its canonical feature map ϕ. Since the $\phi(x_i)'s$ are by definition valued in a Hilbert, KAE can be applied. This way, we have extended the autoencoding paradigm to any set, and finite dimensional representations can be extracted from all types of data. Again, such extension is novel to our knowledge. Figure 2 depicts the procedure, referred to as K^{2}AE, since the new criterion is a *kernelization* of the KAE that reads:

$$\frac{1}{n} \sum_{i=1}^{n} \|\phi(x_i) - f_L \circ \ldots \circ f_1(\phi(x_i))\|^2_{\mathcal{H}} + \sum_{l=1}^{L} \lambda_l \|f_l\|^2_{\mathcal{H}_l}. \tag{4}$$

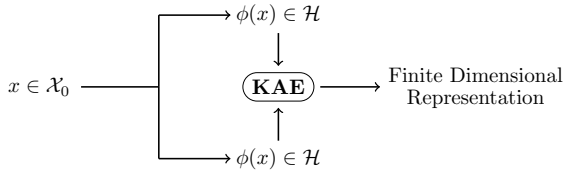


Figure 2: Autoencoding any data thanks to K^{2}AE

3 THEORETICAL ANALYSIS

It is the purpose of this section to investigate theoretical properties of the introduced model, its capacity to be learnt from training data with a controlled generalization error, and the connection between K^{2}AE and Kernel PCA (KPCA) namely.

3.1 Generalization Bound

While the algorithmic formulation aims at minimizing the regularized risk (2), the subsequent theoretical analysis focuses on the constrained problem (1). Theorem 1 relates the solutions from the two approaches to each other, so that bounds derived in the latter setting also apply to numerical solutions of the first one.

Theorem 1. *Let $V : \mathcal{H}_1 \times \ldots \times \mathcal{H}_L \to \mathbb{R}$ be an arbitrary function. Consider the two problems:*

$$\min_{f_l \in \mathcal{H}_l} \left\{ V(f_1, \ldots, f_L) + \sum_{l=1}^{L} \lambda_l \|f_l\|_{\mathcal{H}_l}^2 \right\}, \tag{5}$$

$$\min_{\substack{f_l \in \mathcal{H}_l \\ \|f_l\|_{\mathcal{H}_l} \leq s_l}} V(f_1, \ldots, f_L). \tag{6}$$

Then, for any $(\lambda_1, \ldots, \lambda_L) \in \mathbb{R}_+^L$, there exists $(s_1, \ldots, s_L) \in \mathbb{R}_+^L$ such that any (respectively, local) solution to problem (5) is also a (respectively, local) solution to problem (6).

Refer to Appendix A.1 for the proof and a discussion on the converse statement.

In order to establish generalization bound results for empirical minimizers in the present setting, we now define two key quantities involved in the proof, *i.e.* Rademacher and Gaussian averages for classes of Hilbert-valued functions.

Definition 2. *Let $\mathcal{X}$ be any measurable space, and H a separable Hilbert space. Consider a class $\mathcal{C}$ of measurable functions $h : \mathcal{X} \to H$. Let $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n$ be $n \geq 1$ independent H-valued Rademacher variables and define:*

$$\widehat{\mathscr{R}}_n(\mathcal{C}(S)) = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{\sigma}_i, h(x_i) \rangle_H \right].$$

If $H = \mathbb{R}$, it is the classical Rademacher average (see *e.g.* Mohri et al. (2012) p.34), while, when $H = \mathbb{R}^p$, it corresponds to the expectation of the supremum of the sum of the Rademacher averages over the p components of h (see Definition 2.1 in Maurer and Pontil (2016)). If H is an infinite dimensional Hilbert space with countable orthonormal basis $(e_k)_{k \in \mathbb{N}}$, we have:

$$\widehat{\mathscr{R}}_n(\mathcal{C}(S)) = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{\infty} \sigma_{i,k} \langle h(x_i), e_k \rangle_H \right].$$

The Gaussian counterpart of $\widehat{\mathscr{R}}_n(\mathcal{C}(S))$, obtained by replacing Rademacher random variables/processes with standard H-valued Gaussian ones, is denoted by $\widehat{\mathscr{G}}_n(\mathcal{C}(S))$ throughout the paper.

For the sake of simplicity, results in the rest of the subsection are derived in the 2-layer case solely, with $\mathcal{X}_1$ finite dimensional (*i.e.* $\mathcal{X}_1 = \mathbb{R}^p$), although the approach remains valid for deeper architectures.

Let $\mathcal{H}_{1,s} := \{f \in \mathcal{H}_1 : \|f\|_{\mathcal{H}_1} \leq s\}$, and similarly $\mathcal{H}_{2,t} := \{g \in \mathcal{H}_2 : \|g\|_{\mathcal{H}_2} \leq t, \sup_{y \in \mathbb{R}^p} \|g(y)\|_{\mathcal{X}_0} \leq M\}$. We shall use the notation $\mathcal{H}_{s,t} \subset \mathcal{F}(\mathcal{X}_0, \mathcal{X}_0)$ to mean the space of composed functions $\mathcal{H}_{1,s} \circ \mathcal{H}_{2,t} := \{h \in \mathcal{F}(\mathcal{X}_0, \mathcal{X}_0) : \exists (f, g) \in \mathcal{H}_{1,s} \times \mathcal{H}_{2,t},\ h = g \circ f\}$. To simplify the notation, ϵ (and $\hat{\epsilon}_n$) may be abusively considered as a functional with one or two arguments: $\epsilon(f, g) = \epsilon(g \circ f) = \mathbb{E}_{X \sim P} \|X - g \circ f(X)\|_{\mathcal{X}_0}^2$. Finally, let $\hat{h}_n$ denote the minimizer of $\hat{\epsilon}_n$ over $\mathcal{H}_{s,t}$, and ϵ^* the infimum of ϵ on the same functional space.

Assumption 3. *There exists $K < +\infty$ such that:*

$$\forall x \in \mathcal{X}_0, \quad \mathbf{Tr}\big(\mathcal{K}_1(x, x)\big) \leq Kp.$$

Assumption 4. *There exists $L < +\infty$ such that for all y, y' in $\mathbb{R}^p$:*

$$\mathbf{Tr}\big(\mathcal{K}_2(y, y) - 2\mathcal{K}_2(y, y') + \mathcal{K}_2(y', y')\big) \leq L^2 \|y - y'\|_{\mathbb{R}^p}^2.$$

Theorem 5. *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be OVKs satisfying Assumptions 3 and 4 respectively. Then, there exists a universal constant $C_0 < +\infty$ such that, for any $0 < \delta < 1$, we have with probability at least $1 - \delta$:*

$$\epsilon(\hat{h}_n) - \epsilon^* \leq C_0 LMst \sqrt{\frac{Kp}{n}} + 24M^2 \sqrt{\frac{\log(2)/\delta}{2n}}.$$

The proof relies on a Rademacher bound, which is in turn upper bounded using Corollary 4 in Maurer (2016), an extension of Theorem 2 in Maurer (2014) proved in the Supplementary Material, and several intermediary results derived from the stipulated assumptions. Technical details are deferred to Appendix A.2.

Attention should be paid to the fact that constants in Theorem 5 appear in a very interpretable fashion: the less spread the input (the smaller the constant M), the more restrictive the constraints on the functions (the smaller K, L, s and t), and the smaller the internal dimension p, the sharper the bound.

3.2 K^{2}AE and Kernel PCA: a Connection

Just as Bourlard and Kamp (1988) have shown a mere equivalence between PCA and standard 2-layer AEs, a similar link can be established between 2-layer K^{2}AE and Kernel PCA. Throughout the analysis, a 2-layer K^{2}AE is considered, with decomposable kernels made

of linear scalar kernels and identity operators. Also, there is no penalization (*i.e.* $\lambda_1 = \lambda_2 = 0$). We want to autoencode data into $\mathbb{R}^p$, after a first embedding through the feature map ϕ, like in (4).

3.2.1 Finite Dimensional Feature Map

Let us assume first that ϕ is valued in $\mathbb{R}^d$, with $p < d < n$. Let $\Phi = (\phi(x_1), \dots, \phi(x_n))^T \in \mathbb{R}^{n \times d}$ denote the matrix storing the $\phi(x_i)^T$ to autoencode in rows. Note that $K_\phi = \Phi\Phi^T \in \mathbb{R}^{n \times n}$ corresponds to the Gram matrix associated to ϕ. As shall be seen in Theorem 6, the optimal f and g have a specific form, so that they only depend on two coefficient matrices, $A \in \mathbb{R}^{n \times p}$ and $B \in \mathbb{R}^{n \times d}$ respectively. Equipped with this notation, one has: $Y = f_A(\Phi) = \Phi\Phi^T A \in \mathbb{R}^{n \times p}$, and $\widetilde{\Phi} = g_B(Y) = YY^T B \in \mathbb{R}^{n \times d}$. Without penalization, the goal is then to minimize in A and B:

$$\|\Phi - \widetilde{\Phi}\|_{Fr}^2.$$

$\widetilde{\Phi}$ being at most of rank p, we know from Eckart-Young Theorem that the best possible $\widetilde{\Phi}$ is given by $\Phi^* = U\Sigma_p V^T$, where $\left(U \in \mathbb{R}^{n \times d}, \Sigma \in \mathbb{R}^{d \times d}, V^T \in \mathbb{R}^{d \times d}\right)$ is the *thin* Singular Value Decomposition (SVD) of Φ such that $\Phi = U\Sigma V^T$, and Σ_p is equal to Σ, but with the $d - p$ smallest singular values zeroed.

Let us now prove that there exists a couple of matrices (A^*, B^*) such that $g_{B^*} \circ f_{A^*}(\Phi) = \Phi^*$. One can verify that $(A^* = U_p \overline{\Sigma}_p^{-3/2}, B^* = UV^T)$, with $U_p \in \mathbb{R}^{n \times p}$ storing only the p largest eigenvectors of K_ϕ, and $\overline{\Sigma}_p \in \mathbb{R}^{p \times p}$ the $p \times p$ top left block of Σ_p, satisfy it. Finally, the optimal encoding returned is $Y^* = f_{A^*}(\Phi) = \left(\sqrt{\sigma_1}u_1, \dots, \sqrt{\sigma_p}u_p\right)$, with $u_1, \dots, u_p$ the p largest eigenvectors of K_ϕ, while the KPCA's new representation is $(\sigma_1 u_1, \dots, \sigma_p u_p)$.

We have shown that a specific instance of K^{2}AE can be solved explicitly using a SVD, and that the optimal coding returned is close to the one output by KPCA.

3.2.2 Infinite Dimensional Feature Map

Let us assume now that ϕ is valued in a general Hilbert space $\mathcal{H}$. Φ is now seen as the linear operator from $\mathcal{H}$ to $\mathbb{R}^n$ such that $\forall \alpha \in \mathcal{H}$, $\Phi\alpha = (\langle \alpha, \phi(x_1) \rangle_{\mathcal{H}}, \dots, \langle \alpha, \phi(x_n) \rangle_{\mathcal{H}}) \in \mathbb{R}^n$. Since Theorem 1 makes no assumption on the dimensionality, everything stated in the finite dimensional scenario applies, except that $B \in \mathcal{L}(\mathcal{H}, \mathbb{R}^n)$, and that we minimize the Hilbert-Schmidt norm: $\|\Phi - \widetilde{\Phi}\|_{HS}^2$. We then need an equivalent of Eckart-Young Theorem. It still holds since its proof only requires the existence of an SVD for any operator, which is granted in our case since we deal with compact operators (they have finite rank n). The end of the proof is analogous to the finite dimensional case.

4 THE KAE ALGORITHMS

This section describes at length the algorithms we propose to solve problems (3) and (4). They raise two major issues as their objective functions are non-convex, and their search spaces are infinite dimensional. However, this last difficulty is solved by Theorem 6.

4.1 A Representer Theorem

Theorem 6. *Let $L_0 \in [\![L]\!]$, and $V : \mathcal{X}_{L_0}^n \times \mathbb{R}_+^{L_0} \to \mathbb{R}$ a function of $n + L_0$ variables, strictly increasing in each of its L_0 last arguments. Suppose that $(f_1^*, \dots, f_{L_0}^*)$ is a solution to the optimization problem:*

$$\min_{f_l \in \mathcal{H}_l} V\Big((f_{L_0} \circ \dots \circ f_1)(x_1), \dots, (f_{L_0} \circ \dots \circ f_1)(x_n), \|f_1\|_{\mathcal{H}_1}, \dots, \|f_{L_0}\|_{\mathcal{H}_{L_0}}\Big).$$

Let $x_i^{*(l)} := f_l^* \circ \dots \circ f_1^*(x_i)$, with $x_i^{*(0)} := x_i$. Then, $\exists \left(\varphi_{1,1}^*, \dots, \varphi_{1,n}^*, \dots, \varphi_{L_0,n}^*\right) \in \mathcal{X}_1^n \times \dots \times \mathcal{X}_{L_0}^n$:

$$\forall\, l \in [\![L_0]\!], \quad f_l^*(\cdot) = \sum_{i=1}^n \mathcal{K}_l\left(\ \cdot\ , x_i^{*(l-1)}\right)\varphi_{l,i}^*.$$

Proof. Refer to Appendix A.3 □

This Theorem exhibits a very specific structure for the minimizers, as each layer's support vectors are the images of the original points by the previous layer.

4.2 Finite Dimension Case

In this section, let us assume that $\mathcal{X}_l = \mathbb{R}^{d_l}$ for $l \in [\![L]\!]$. The objective function of (3), viewed as a function of $(f_L \circ \dots \circ f_1)(x_1), \dots, (f_L \circ \dots \circ f_1)(x_n), \|f_1\|_{\mathcal{H}_1}, \dots, \|f_L\|_{\mathcal{H}_L}$ satisfies the condition on V involved in Theorem 6. After applying it (with $L_0 = L$), problem (3) boils down to the problem of finding the $\varphi_{l,i}^*$'s, which are finite dimensional. This crucial observation shows that our problem can be solved in a computable manner. However, its convexity still cannot be ensured (see Appendix A.4).

The objective only depending on the $\varphi_{l,i}$'s, problem (3) can be approximately solved by Gradient Descent (GD). We now specify the gradient derivation in the decomposable OVKs case, *i.e.* for any layer l there exists a scalar kernel k_l and $A_l \in \mathcal{L}(X_l)$ positive semidefinite such that $\mathcal{K}_l(x, x') = k_l(x, x')A_l$. All detailed computations can be found in Appendix B. Let $\Phi_l := (\varphi_{l,1}, \dots, \varphi_{l,n})^T \in \mathbb{R}^{n \times d_l}$ storing the coefficients $\varphi_{l,i}$ in rows, and $K_l \in \mathbb{R}^{n \times n}$ such that $[K_l]_{i,i'} = k_l\left(x_i^{(l-1)}, x_{i'}^{(l-1)}\right)$. Let $(l_0, i_0) \in [\![L]\!] \times [\![n]\!]$, the gradient of the distortion term reads:

$$\left(\nabla_{\varphi_{l_0,i_0}} \frac{1}{n}\sum_{i=1}^{n} \|x_i - f_L \circ \ldots \circ f_1(x_i)\|^2_{\mathcal{X}_0}\right)^T \tag{7}$$
$$= -\frac{2}{n}\sum_{i=1}^{n}\left(x_i - {x_i}^{(L)}\right)^T \mathbf{Jac}_{{x_i}^{(L)}}(\varphi_{l_0,i_0}).$$

On the other hand, $\|f_l\|^2_{\mathcal{H}_l}$ may be rewritten as:

$$\|f_l\|^2_{\mathcal{H}_l} = \sum_{i,i'=1}^{n} k_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right)\langle \varphi_{l,i}, A_l\varphi_{l,i'}\rangle_{\mathcal{X}_l}, \tag{8}$$

so that it may depend on φ_{l_0,i_0} in two ways: 1) if $l_0 = l$, there is a direct dependence of the second quadratic term, 2) but note also that for $l_0 < l$, the $\varphi_{l_0,i}$ have an influence on the ${x_i}^{(l-1)}$ and so on the first term. This remark leads to the following formulas:

$$\nabla_{\Phi_l}\ \|f_l\|^2_{\mathcal{H}_l} = 2\ K_l\Phi_l A_l, \tag{9}$$

with $\nabla_{\Phi_l}F \coloneqq \left((\nabla_{\varphi_{l,1}}F)^T, \ldots, (\nabla_{\varphi_{l,n}}F)^T\right)^T \in \mathbb{R}^{n\times d_l}$ storing the gradients of any real-valued function F with respect to the $\varphi_{l,i}$ in rows. And when $l_0 < l$:

$$\left(\nabla_{\varphi_{l_0,i_0}}\|f_l\|^2_{\mathcal{H}_l}\right)^T = 2\sum_{i,i'=1}^{n}\Big\{ \tag{10}$$
$$[N_l]_{i,i'}\left(\nabla^{(1)}k_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right)\right)^T \mathbf{Jac}_{{x_i}^{(l-1)}}(\varphi_{l_0,i_0})\Big\},$$

where $\nabla^{(1)}k_l(x,x')$ denotes the gradient of $k_l(\cdot,\cdot)$ with respect to the 1^{st} coordinate evaluated in (x,x'), and N_l the $n\times n$ matrix such that $[N_l]_{i,i'} = \langle\varphi_{l,i}\ ,\ A_l\ \varphi_{l,i'}\rangle_{\mathcal{X}_l}$. Again, assuming the matrices $\mathbf{Jac}_{{x_i}^{(L)}}(\varphi_{l_0,i_0})$ are known, the norm part of the gradient is computable. Combining expressions (7), (9) and (10) using the linearity of the gradient leads readily to the complete formula.

If n, L, and p denote respectively the number of samples, the number of layers, and the size of the largest latent space, the algorithm complexity is no more than $\mathcal{O}(n^2Lp)$ for objective evaluation, and $\mathcal{O}(n^3L^2p^3)$ for gradient derivation. Hence, it appears natural to consider stochastic versions of GD. But as shown by equation (10), the norms gradients involve the computation of many Jacobians. Selecting a mini-batch does not affect these terms, which are the most time consuming. Thus, the expected acceleration due to stochasticity must not be so important. Nevertheless, a *doubly stochastic* scheme where both the points on which the objective is evaluated, as well as the coefficients to be updated, are chosen randomly at each iteration, might be of high interest since it would dramatically decrease the number of Jacobians computed. However, this approach goes beyond the scope of this paper, and is left for future work.

4.3 General Hilbert Space Case

In this section, $\mathcal{X}_0$ (and so $\mathcal{X}_L$) are supposed to be infinite dimensional. Despite this relaxation, KAEs remains computable. As Theorem 6 makes no assumption on the dimensionality of $\mathcal{X}_0$, it can be applied. The only difference is that coefficients $\varphi_{L,i}$'s $\in \mathcal{X}_L^n$ are infinite dimensional, preventing from the use of a global GD. But assuming the $\varphi_{L,i}$'s to be fixed, a GD can still be performed on the $\varphi_{l,i}$'s, $l \in [\![L-1]\!]$. On the other hand, if one assumes these coefficients fixed, the optimal $\varphi_{L,i}$'s are the solutions to a Kernel Ridge Regression (KRR). Consequently, a hybrid approach alternating GD and KRR is considered. Two issues remain to be addressed: 1) how to compute the KRR in $\mathcal{X}_L$, 2) how to propagate the gradients through $\mathcal{X}_L$.

From now, A_L is assumed to be the identity operator. If the $\varphi_{l,i}$'s, $l \in [\![L-1]\!]$ are fixed, then the best $\varphi_{L,i}$'s shall satisfy (Micchelli and Pontil, 2005) for all $i \in [\![n]\!]$:

$$\sum_{i'=1}^{n}\left(\mathcal{K}_L\left({x_i}^{(L-1)}, {x_{i'}}^{(L-1)}\right) + n\lambda_L\delta_{ii'}\right)\varphi_{L,i'} = x_i. \tag{11}$$

In particular, the computation of N_L becomes explicit (Appendix B.5) as long as we know the dot products $\langle x_j, x_{j'}\rangle_{\mathcal{X}_0}$. In the case of the K^{2}AE, these dot products are the input Gram matrix K_{in}. Let N_{KRR} be the function that computes N_L from the $\varphi_{l,i}$'s, $l \in [\![L-1]\!]$, K_{in} and λ_L. What is remarkable is that knowing N_L (and not each $\varphi_{L,i}$ individually) is enough to propagate the gradient through the infinite dimensional layer.

Indeed, let us assume now that N_L is fixed. All spaces but $\mathcal{X}_L$ remaining finite dimensional, changes in the gradients only occur where the last layer is involved, namely for the distortion and for $\|f_L\|^2_{\mathcal{H}_L}$. As for the gradients of $\|f_L\|^2_{\mathcal{H}_L}$, equation (10) remain true. If N_L is given, there is no difficulty. As for the distortion, the use of the differential (see Appendix B.6) gives:

$$\nabla_{\varphi_{l_0,i_0}}\left\|x_i - x_i^{(L)}\right\|^2_{\mathcal{X}_0}{}^T = -2\sum_{i'=1}^{n}\Big\{ \tag{12}$$
$$\left\langle x_i - x_i^{(L)}, \varphi_{L,i'}\right\rangle_{\mathcal{X}_L}\left(\nabla_{\varphi_{l_0,i_0}}k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right)\right)^T\Big\}.$$

It is a direct extension of (7), where $\mathbf{Jac}_{{x_i}^{(L)}}(\varphi_{l_0,i_0})$, has been replaced using the definition of ${x_i}^{(L)}$. Using again (11), $\langle x_i - x_i^{(L)}, \varphi_{L,i'}\rangle_{\mathcal{X}_L}$ can be rewritten as $n\lambda_L\langle\varphi_{L,i}, \varphi_{L,i'}\rangle_{\mathcal{X}_L} = n\lambda_L[N_L]_{i,i'}$, and infinite dimensional objects are dealt with. The crux of the algorithm is that infinite dimensional coefficients $\varphi_{L,i}$'s are never computed, but only their scalar products. Not knowing the $\varphi_{L,i}$'s is of no importance, as we are interested in the encoding function, which does not rely on them. Let T be a number of epochs, and γ_t a step size rule, the approach is summarized in Algorithm 1.

Algorithm 1 General Hilbert KAE and K^{2}AE

input : Gram matrix K_{in}
init : $\Phi_1 = \Phi_1^{init}, \dots, \Phi_{L-1} = \Phi_{L-1}^{init}$,
$N_L = N_{\text{KRR}}(\Phi_1, \dots, \Phi_{L-1}, K_{in}, \lambda_L)$
for *epoch* t *from* 1 *to* T **do**
  // inner coefficients updates at fixed N_L
  for *layer* l *from* 1 *to* $L-1$ **do**
    $\Phi_l = \Phi_l - \gamma_t \nabla_{\Phi_l}(\hat{\epsilon}_n + \Omega \mid N_L)$
  // N_L update
  $N_L = N_{\text{KRR}}(\Phi_1, \dots, \Phi_{L-1}, K_{in}, \lambda_L)$
return $\Phi_1, \dots, \Phi_{L-1}$

5 NUMERICAL EXPERIMENTS

Numerical experiments have been run in order to assess the ability of KAEs to provide relevant data representations. We used decomposable OVKs with the identity operator as A, and the Gaussian kernel as k. First, we present insights on the interesting properties of the KAEs via a 2D example. Then, we describe more involved experiments on the NCI dataset to measure the power of KAEs.

5.1 Behavior on a 2D problem

Let us first consider three noisy concentric circles such as in Figure 3(a). Although the main strength of KAEs is to perform autoencoding on complex data (Section 2.5), they can still be applied on real-valued points. Figures 3(b) and 3(c) show the reconstructions obtained after fitting respectively a 2-1-2 standard and kernel AE. Since the latent space is of dimension 1, the 2D reconstructions are manifolds of the same dimension, hence the curve aspect. What is interesting though is that the KAE learns a much more complex manifold than the standard AE. Due to its linear limitations (the nonlinear activation functions did not help much in this case), the standard AE returns a line, far from the original data, while the KAE outputs a more complex manifold, much closer to the initial data.

Apart from a good reconstruction, we are interested in finding representations with attractive properties. The 1D feature found by the previous KAE is interesting, as it is a discriminative one with respect to the original clusters: points from different circles are mapped around different values (Figure 3(d)). Interestingly, after a few iterations, some variability is introduced around these *cluster values*, so that all codes shall not be mapped back to the same point (Figure 3(e)).

Finally, a KAE with 1 hidden layer of size 2 gives the internal representation shown in Figure 3(f). This new 2D representation has a disentangling effect: the circle structure is kept so as to preserve the intra-cluster specificity, while the inter-cluster differentiation is ensured by the circles' dissociation. These visual 2D examples give interesting insights on the good properties of the KAE representations: discrimination, disentanglement (see further experiments in Appendix C.1).

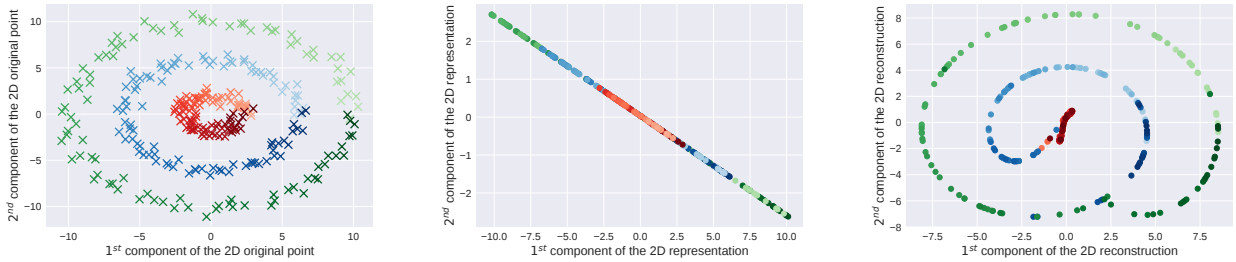
5.2 Representation Learning on Molecules

We now present an application of KAEs in the context of chemoinformatics. The motivation is triple. First, such complex data cannot be handled by standard AEs. Second, kernel methods being prominent in the field, data are often stored as Gram matrices, suiting perfectly our framework. Third, finding a compressed representation of a molecule is a problem of highest interest in Drug Discovery. We considered two different problems, one supervised, one unsupervised.

As for the supervised one, we exploited the dataset of Su et al. (2010) from the NCI-Cancer database: it consists in a Gram matrix comparing 2303 molecules by the mean of a Tanimoto kernel (a linear path kernel built using the presence or absence of sequences of atoms in the molecule), as well as the molecules activities in the presence of 59 types of cancer. The dataset containing no vectorial representations of the molecules (but only Gram matrices), only kernel methods were possible to benchmark. As a good representation is supposed to facilitate ulterior learning tasks, we assess the goodness of the representations through the regression scores obtained by Random Forests (RFs) from scikit-learn (Pedregosa et al., 2011) fed with it.

2-layer K^{2}AEs with respectively 5, 10, 25, 50 and 100 internal dimension were run, as well as Kernel Principal Component Analyses (KPCAs) with the same number of components. Finally, these representations were given as inputs to RFs. KRR was also added to the comparison. The Normalized Mean Squared Errors (NMSEs), averaged on 10 runs, for 5 strategies and on the first 10 cancers are stored in Table 1 (the complete results are available in Appendix C.2). A visualization with all strategies is also proposed in Figure 8. Clearly, methods combining a data representation step followed by a prediction one performs better. But the good performance of our approach should not be attributed to the use of RFs only, since the same strategy run with KPCA leads to worse results. Indeed, the K^{2}AE 50 + RF strategy outperforms all other procedures on all problems, managing to extract compact and useful feature vectors from the molecules.

The data for the unsupervised problem is taken from Brouard et al. (2016a). It is composed of two sets (a train set of size 5579, and a test set of size 1395), each one containing metabolites under the form of 4136-long binary vectors (called fingerprints), as well as a Gram matrix comparing them. 2-layer standard AEs



(a) 2D Input Data (b) AE Reconstruction (c) KAE Reconstruction

(d) 1D Representation (5 it.) (e) 1D Representation (20 it.) (f) 2D Re-Representation

Figure 3: KAE Performance on Noisy Concentric Circles

Table 1: NMSEs on Molecular Activity for Different Types of Cancer

	KRR	KPCA 10 + RF	KPCA 50 + RF	K^{2}AE 10 + RF	K^{2}AE 50 + RF
Cancer 01	0.02978	0.03279	0.03035	0.03097	**0.02808**
Cancer 02	0.03004	0.03194	0.02978	0.03099	**0.02775**
Cancer 03	0.02878	0.03155	0.02914	0.02989	**0.02709**
Cancer 04	0.03003	0.03274	0.03074	0.03218	**0.02924**
Cancer 05	0.02954	0.03185	0.02903	0.03065	**0.02754**
Cancer 06	0.02914	0.03258	0.03083	0.03134	**0.02838**
Cancer 07	0.03113	0.03468	0.03207	0.03257	**0.03018**
Cancer 08	0.02899	0.03162	0.02898	0.03065	**0.02770**
Cancer 09	0.02860	0.02992	0.02804	0.02872	**0.02627**
Cancer 10	0.02987	0.03291	0.03111	0.03170	**0.02910**

Table 2: MSREs on Test Metabolites

Dimension	AE (sigmoid)	AE (relu)	KAE
5	99.81	96.62	**76.38**
10	87.36	84.02	**65.76**
25	72.31	68.77	**51.63**
50	63.00	58.29	**40.72**
100	55.43	48.63	**36.27**

from Keras (Chollet et al., 2015) with sigmoid and relu activation functions, and 2-layer KAEs with internal layer of size 5, 10, 25, 50 and 100, were trained. In absence of a supervised task, we measured the Mean Squared Reconstruction Errors (MSREs) induced on the test set, and stored them in Table 2. Again, the KAE approach shows a systematic improvement.

6 CONCLUSION

We introduce a new framework for AEs, based on vv-RKHSs and OVKs. The use of RKHS functions enables KAEs to handle data from possibly infinite dimensional Hilbert spaces, and then to extend the autoencoding scheme to any kind of data. A generalization bound and a strong connection to KPCA are established, while the underlying optimization problem is tackled by a Representer Theorem and the kernel trick. Beyond a detailed description, the behavior of the algorithm is carefully studied on simulated data, and yields relevant performances on graph data, that standard AEs are typically unable to handle. Further research may consider a semi-supervised approach, that would ideally tailor the representation according to the future targeted task.

Acknowledgment This work has been funded by the Industrial Chair *Machine Learning for Big Data* from Télécom ParisTech, Paris, France.

References

Alain, G. and Bengio, Y. (2014). What regularized auto-encoders learn from the data-generating distribution. *J. Mach. Learn. Res.*, 15(1):3563–3593.

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266.

Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 37–49.

Bauschke, H. H., Combettes, P. L., et al. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer.

Bengio, Y., Courville, A., Vincent, P., and Umanità, V. (2013). Representation learning: a review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Bourlard, H. and Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294.

Brouard, C., Shen, H., Dührkop, K., d'Alché-Buc, F., Böcker, S., and Rousu, J. (2016a). Fast metabolite identification with input output kernel regression. *Bioinformatics*, 32(12):28–36.

Brouard, C., Szafranski, M., and d'Alché-Buc, F. (2016b). Input output kernel regression: Supervised and semi-supervised structured output prediction with operator-valued kernels. *Journal of Machine Learning Research*, 17:176:1–176:48.

Caponnetto, A., Micchelli, C. A., , M., and Ying, Y. (2008). Universal multitask kernels. *Journal of Machine Learning Research*, 9:1615–1646.

Chollet, F. et al. (2015). Keras, https://keras.io.

Gholami, B. and Hajisami, A. (2016). Kernel autoencoder for semi-supervised hashing. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE.

Kadri, H., Duflos, E., Preux, P., Canu, S., Rakotomamonjy, A., and Audiffren, J. (2016). Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17:20:1–20:54.

Kampffmeyer, M., Løkse, S., Bianchi, F. M., Jenssen, R., and Livi, L. (2017). Deep kernelized autoencoders. In *Scandinavian Conference on Image Analysis*, pages 419–430. Springer.

Kipf, T. N. and Welling, M. (2016). Variational graph autoencoders. *NIPS Workshop on Bayesian Deep Learning*.

Ledoux, M. and Talagrand, M. (1991). *Probability in Banach Spaces: Isoperimetry and Processes*. Springer Science & Business Media.

Maurer, A. (2014). A chain rule for the expected suprema of gaussian processes. In *Algorithmic Learning Theory: 25th International Conference, ALT 2014, Bled, Slovenia, October 8-10, 2014, Proceedings*, volume 8776, page 245. Springer.

Maurer, A. (2016). A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer.

Maurer, A. and Pontil, M. (2016). Bounds for vector-valued function estimation. *arXiv preprint arXiv:1606.01487*.

Micchelli, C. A. and Pontil, M. (2005). On learning vector-valued functions. *Neural computation*, 17(1):177–204.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning*. MIT press.

Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pisier, G. (1986). Probabilistic methods in the geometry of banach spaces. In *Probability and analysis*, pages 167–241. Springer.

Salakhutdinov, R. and Hinton, G. (2009). Deep boltzmann machines. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 448–455. PMLR.

Senkene, E. and Tempel'man, A. (1973). Hilbert spaces of operator-valued functions. *Lithuanian Mathematical Journal*, 13(4):665–670.

Su, H., Heinonen, M., and Rousu, J. (2010). Structured output prediction of anti-cancer drug activity. In Dijkstra, T., Tsivtsivadze, E., Marchiori, E., and Heskes, T., editors, *Pattern Recognition in Bioinformatics - 5th IAPR International Conference, PRIB 2010, Proceedings*, volume 6282 of *Lecture Notes in Computer Science*, pages 38–49. Springer.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408.