Autoencoding any Data through Kernel Autoencoders

Pierre Laforgue, Stephan Clémençon, Florence d'Alché-Buc

LTCI, Télécom ParisTech
Chaire Machine Learning for Big Data

Outline

Deep neural networks

Neural Networks

- Raw data / Flexibility of architectures
- A unique optimization tool: stochastic gradient descent and variants
- But many tricks and heuristics that make difficult to reproduce some results
- Spectacular results on *deep architectures* trained on massive datasets
- A very few theoretical insights

Kernel methods

- Nonlinear and Non-vectorial data
- Shallow architectures / Linear regression in feature space (RKHS)
- Control of the approach: the **kernel** rules everything (RKHS)
- Quadratic programming / online methods
- Best results on either structured data or low-data regime
- Requires approximations to scale up
- A lot of theoretical results

Motivation 1

Deep learning and kernel methods

- Convolutional kernel networks (Mairal, 2016)
- Random Fourier Features (Rahimi and Recht 2007)
- Deep Kernel Learning (with GP's): Wilson et al. 2015

First general goal: understand deep learning in the context of kernels methods (Belkin et al. 2018)

Motivation 2

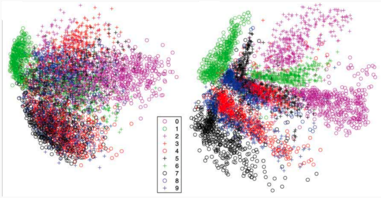
Second Goal: Address Representation learning with deep kernel methods

Representation learning (Bengio et al. 2017) opposed to feature engineering/design with experts:

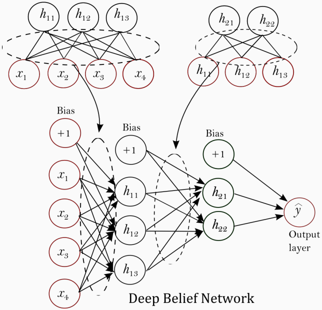
- Dimension reduction from raw data (for visualization, for efficient computations)
- Denoising representations
- Generation of new samples
- Pre-training of neural architectures

Focus on Autoencoders

- Data compression (PCA) [*Bourlard 1988, Hinton 2006*]
- Pre-training of neural networks [*Bengio & al. 2007*]
- Denoising [*Vincent, Larochelle & al. 2010*]
- Recent works: variational autoencoders, adversarial autoencoders etc ...



(a) PCA / AE



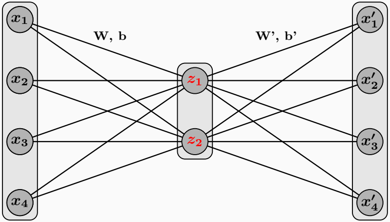
(b) Pre-training by AE

Autoencoders (AEs): Principle

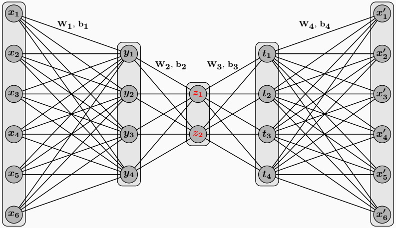
- **Idea:** compress and reconstruct inputs by a Neural Net (NN)
- Elementary mapping: $f : [0,1]^d \to [0,1]^p$ such that

$$f(x) = \sigma(Wx + b), \;\; W \in \mathbb{R}^{p \times d}, b \in \mathbb{R}^p$$

- Neural network: symmetric, hour-glass shaped
- **AE:** output x' must match input x (self-supervised)



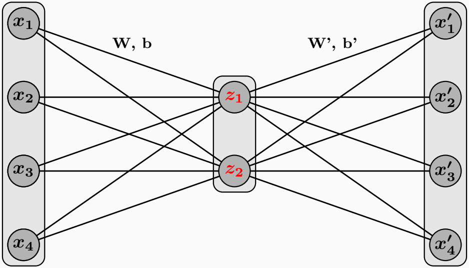
(c) 1 hidden layer AE



(d) 3 hidden layers AE

Autoencoders: Training

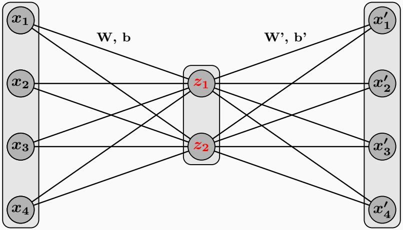
- $z = f_{\boldsymbol{W},\boldsymbol{b}}(x) = \sigma(\boldsymbol{W}x + \boldsymbol{b}) \quad x' = f_{\boldsymbol{W}',\boldsymbol{b}'}(z) = \sigma(\boldsymbol{W}'z + \boldsymbol{b}')$
- $\boldsymbol{\theta}^* = argmin_\theta \|x - x'\|^2 = argmin_\theta \left\| x - f_{\boldsymbol{W}',\boldsymbol{b}'} \circ f_{\boldsymbol{W},\boldsymbol{b}}(x) \right\|^2$
- Optimal encoding $z^* = \sigma(\boldsymbol{W}^*x + \boldsymbol{b}^*)$



Autoencoders: Summary

❶ $$\min_{f_l \in \mathrm{NN}_{\mathrm{em}}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|^2$$

❷ $x_i \in [0,1]^d$ or $x_i \in \mathbb{R}^d$



Goal of this work

Extend autoencoder to structured, complex data and propose kernel-based autoencoders:

- KAE where neural layers are replaced by functions in vector-valued Reproducing Kernel Hilbert Spaces
- K^2AE that takes **any data** under the form of a **Gram matrix**
- Representer theorem ? Optimization ? Connection to Kernel PCA ? Generalization bounds ?

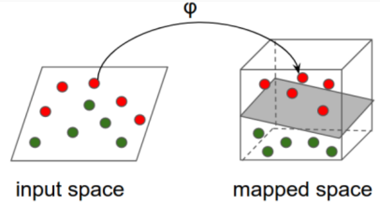
Outline

Kernel Methods: (Scalar) Reminder

- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
- $\forall (x, x') \in \mathcal{X} \times \mathcal{X}, \quad k(x, x') = k(x', x)$ (symmetry)
- $\sum_{i,j=1}^{n} \alpha_i k(x_i, x_j) \alpha_j = \alpha^T K \alpha \geq 0$ (positiveness)

- $\exists \mathcal{H}_k$ Hilbert, $\varphi : \mathcal{X} \to \mathcal{H}_k, \quad k(x, x') = \left\langle \varphi(x), \varphi(x') \right\rangle_{\mathcal{H}_k}$
- $\mathcal{H}_k = \overline{Span\left\{\varphi(x) = k(\cdot, x) : x \in \mathcal{X}\right\}} \subset \mathcal{F}(\mathcal{X}, \mathbb{R})$ (RKHS)

Kernel Methods: Kernelization of the Ridge Regression



$X \in \mathbb{R}^{n \times p}$, $Y \in \mathbb{R}^n$

- $\min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|^2 + 2n\lambda\|\beta\|^2$
- $\min_{\beta \in \mathbb{R}^p} \sum_i (y_i - \langle x_i, \beta \rangle_{\mathbb{R}^p})^2 + 2n\lambda\|\beta\|^2_{\mathbb{R}^p}$

- $\min_{\omega \in \mathcal{H}_k} \sum_i (y_i - \langle \varphi(x_i), \omega \rangle_{\mathcal{H}_k})^2 + 2n\lambda\|\omega\|^2_{\mathcal{H}_k}$ $\quad \omega^* = \sum_j \varphi(x_j)\alpha_j^*$
- $\min_{\alpha \in \mathbb{R}^n} \|Y - K\alpha\|^2 + 2n\lambda\alpha^T K\alpha$

From Autoencoders to Kernel Autoencoders

Autoencoders

❶ $$\min_{f_l \in \mathrm{NN}_{\mathrm{em}}} \quad \frac{1}{n} \sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathbb{R}^d}^2$$

❷ $x_i \in [0,1]^d$ or $x_i \in \mathbb{R}^d$

Kernelization

❸ allows to deal with non-vectorial data

❹ $x \longleftrightarrow \varphi(x)$

❺ computable as long as only dot products (or norms) are involved

$$\min_{f_l \in \mathrm{NN}_{\mathrm{em}}} \quad \frac{1}{n} \sum_{i=1}^{n} \left\| \varphi(x_i) - f_L \circ \ldots \circ f_1(\varphi(x_i)) \right\|_{\mathcal{H}_k}^2$$

$\rightarrow$ Need for OVKs and vv-RKHSs

Kernel Methods: (Operator) Definitions

- $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ $\quad$ $\mathcal{Y}$ a Hilbert space $\quad$ (ov-K)
- $\forall (x, x') \in \mathcal{X} \times \mathcal{X}, \quad \mathcal{K}(x, x')^* = \mathcal{K}(x', x)$
- $\sum_{i,j=1}^{n} \langle y_i, \mathcal{K}(x_i, x_j) y_j \rangle_{\mathcal{Y}} \geq 0$

- $\mathcal{H}_{\mathcal{K}} = \overline{Span\{\mathcal{K}(\cdot, x)y : x, y \in \mathcal{X} \times \mathcal{Y}\}} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ (vv-RKHS)
- $f^* \in \underset{f \in \mathcal{H}_{\mathcal{K}}}{\operatorname{argmin}} \; V(f(x_1), \ldots, f(x_n), \|f\|), \qquad f^* = \sum_{i=1}^{n} \mathcal{K}(\cdot, x_i)\beta_i$

Schema

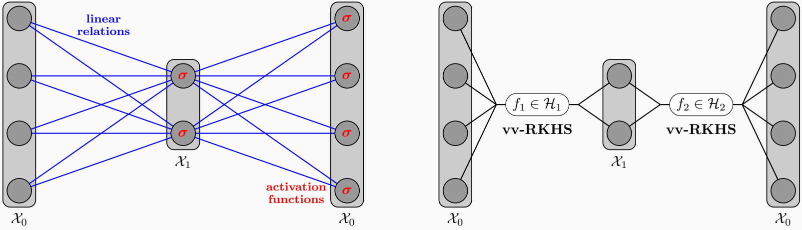


Figure 1: Standard and Kernel 2-layer Autoencoders

Formally

$$\textbf{AE :} \quad \min_{f_l \in \text{NN}_{\text{em}}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2$$

$$\textbf{KAE :} \quad \min_{f_l \in \text{vv-RKHS}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2 + \sum_{l=1}^{L} \lambda_l \| f_l \|_{\mathcal{H}_l}^2$$

1. Novel algorithm of Representation Learning
2. $\mathcal{X}_0$ Hilbert non necessarily Euclidean (not only $\mathbb{R}^d$)
3. Interesting Hilbert: (kernel) feature space

Autoencoding any data

$$\textbf{K}^2\textbf{AE:}\ \min_{f_l \in \text{vv-RKHS}} \frac{1}{n}\sum_{i=1}^{n} \left\| \varphi(x_i) - f_L \circ \ldots \circ f_1(\varphi(x_i)) \right\|_{\mathcal{X}_0}^2 + \sum_{l=1}^{L} \lambda_l \|f_l\|_{\mathcal{H}_l}^2$$

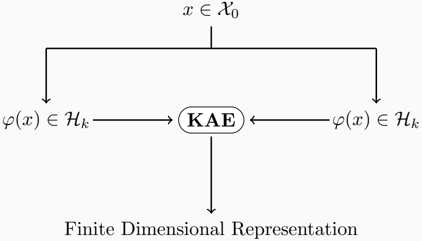


Figure 2: Autoencoding on any $\mathcal{X}_0$

Connection with Kernel PCA

2-layer K^2AE with internal layer of size p, only linear kernels, and without penalization. $K_\phi \in \mathbb{R}^{n \times n}$ denotes the input Gram matrix, $((\sigma_1, u_1) \ldots, (\sigma_p, u_p))$ its p largest eigenvalues/vectors. Then:

$\mathbf{K^2}$AE output: $\left(\sqrt{\sigma_1} u_1, \ldots, \sqrt{\sigma_p} u_p\right) \in \mathbb{R}^{n \times p}$

KPCA output: $(\sigma_1 u_1, \ldots, \sigma_p u_p) \in \mathbb{R}^{n \times p}$

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:*

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

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

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

Optimization

Algorithm 1 General Hilbert KAE and K^2AE

---

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

Stochastic gradient descent with mini-batch is applied except for the last layer.

Last layer: kernel trick in the output space (Input Output Kernel Regression, Brouard et al. JMLR 2016)

Generalization Bound

2-layer KAE on data bounded in norm by M, with:

- internal layer of size p
- encoder $f \in \mathcal{H}_1$ such that $\|f\| \leq s$
- decoder $g \in \mathcal{H}_2$ such that $\|g\| \leq t$, with Lipschitz constant L

Then it holds:

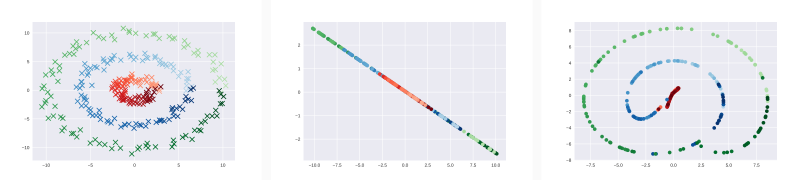
$$\epsilon(\hat{g}_n \circ \hat{f}_n) - \epsilon^* \leq C_0 L M s t \sqrt{\frac{Kp}{n}} + 24 M^2 \sqrt{\frac{\log(2)/\delta}{2n}}.$$

with $\epsilon(g \circ f) = \mathbb{E}_X \|X - g \circ f(X)\|^2_{\mathcal{X}_0}$

Outline

A toy problem: Concentric Circles in 2D

AE and KAE: 2-1-2 architecture



1. 2D example; 2.Reconstruction in 2D (AE); 3. Reconstruction in 2D (KAE)

Testing KAE on Molecular Data

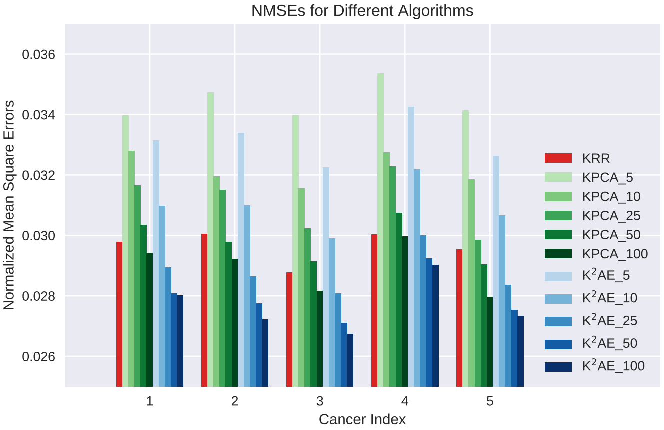
Chemoinformatics: metabolites :=labeled graphs, initially represented by 4136-size fingerprints (Brouard et al., 2016). Training data: 5579 molecules, Test data: 1359 molecules.

Table 1: 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**

Testing K^2AE on Molecular Data (Graphs)

Chemoinformatics: Cancer activity prediction of molecules, dataset for NCi-cancer database avalaible from Su et al. (2010), Gram matrix (Tanimoto kernel) on molecules.



Outline

Introduction

Kernel Autoencoder

Experiments

Conclusion & Future Work

Conclusion & Future Work

Conclusion

- Flexible tool: Advantages from AEs and Kernel Methods
- Extension of standard AEs to any type of data
- Connection with Kernel PCA

Next steps

- Sparse architectures / speeding up learning with better optimization scheme / Approximation
- Combination with a supervised criterion
- Direct extension to feed-forward networks: application to structured output prediction

Preprint available at: http://arxiv.org/abs/1805.11028

Generalization Bound (Sketch of proof, 1)

With $\mathcal{H}_{s,t} \subset \mathcal{F}(\mathcal{X}_0, \mathcal{X}_0) = \mathcal{H}_{1,s} \circ \mathcal{H}_{2,t}$, ℓ the squared norm on $\mathcal{X}_0$.

$$
\begin{aligned}
\widehat{\mathscr{R}}_n\Big((\ell \circ (\mathrm{id} - \mathcal{H}_{s,t}))(S)\Big) &\leq 2\sqrt{2}M \; \widehat{\mathscr{R}}_n\Big((\mathrm{id} - \mathcal{H}_{s,t})(S)\Big), \\
&\leq 2\sqrt{2}M \; \Big[\widehat{\mathscr{R}}_n\Big(\{\mathrm{id}\}(S)\Big) + \widehat{\mathscr{R}}_n\Big(\mathcal{H}_{s,t}(S)\Big)\Big], \\
&\leq 2\sqrt{2}M \; \widehat{\mathscr{R}}_n\Big(\mathcal{H}_{s,t}(S)\Big), \\
\widehat{\mathscr{R}}_n\Big((\ell \circ (\mathrm{id} - \mathcal{H}_{s,t}))(S)\Big) &\leq 2\sqrt{\pi}M \; \widehat{\mathscr{G}}_n\Big(\mathcal{H}_{s,t}(S)\Big).
\end{aligned}
$$

[*Maurer 2016*]

Generalization Bound (Sketch of proof, 2)

$$
\begin{aligned}
\widehat{\mathscr{G}}_n\Big(\mathcal{H}_{s,t}(S)\Big) \leq & C_1 L\Big(\mathcal{H}_{2,t}, \mathcal{H}_{1,s}(S)\Big)\widehat{\mathscr{G}}_n\Big(\mathcal{H}_{1,s}(S)\Big) + \\
& \frac{C_2}{n} R\Big(\mathcal{H}_{2,t}, \mathcal{H}_{1,s}(S)\Big) D\Big(\mathcal{H}_{1,s}(S)\Big) + \\
& \frac{1}{n} G\Big(\mathcal{H}_{2,t}(0)\Big)
\end{aligned}
$$

Extension of [*Maurer 2014*], and bound each term individually.

Connection with KPCA (Proof)

- $X \in \mathbb{R}^{n \times d}$
- $Y = f(X) = XX^T A \in \mathbb{R}^{n \times p}, \quad A \in \mathbb{R}^{n \times p}$
- $Z = g(Y) = YY^T B \in \mathbb{R}^{n \times d}, \quad B \in \mathbb{R}^{n \times d}$
- Goal: $\min_{A,B} \quad \|X - Z\|_{Fr}^2 = \sum_{i=1}^n \|x_i - z_i\|_2^2$

SVD (thin with $d < n$):

- $X = U_d \, \overline{\Sigma}_d \, {V_d}^T$
- $Y = U_d \, \overline{\Sigma}_d{}^2 \, {U_d}^T \, A$
- $Z = U_d \, \overline{\Sigma}_d{}^2 \, {U_d}^T \, A \, A^T \, U_d \, \overline{\Sigma}_d{}^2 \, {U_d}^T \, B$

Eckart-Young:

$$Z^* = U_d \, \overline{\Sigma}_p \, {V_d}^T$$

Sufficient:

$$A = U_p \, \overline{\Sigma}_p{}^{-\frac{3}{2}} \in \mathbb{R}^{n \times p} \qquad B = U_d \, {V_d}^T \in \mathbb{R}^{n \times d}$$