Foundations of Deep Learning from a Kernel Point of View

Julien Mairal

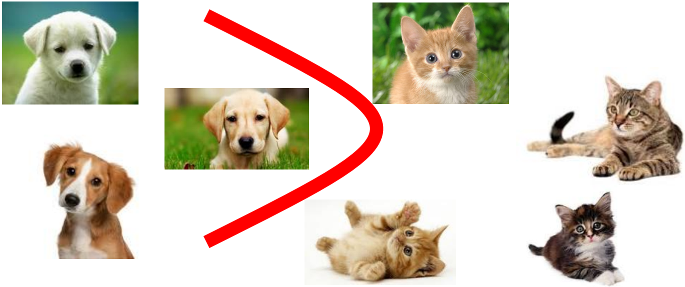
Inria Grenoble

Future of Random Projections II, 2018

Convolutional Neural Networks Behind the Scenes

The goal is to learn a **prediction function** $f : \mathbb{R}^p \to \mathbb{R}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathbb{R}^p$, and y_i in $\mathbb{R}$:

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}}.$$



Convolutional Neural Networks Behind the Scenes

The goal is to learn a **prediction function** $f : \mathbb{R}^p \to \mathbb{R}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathbb{R}^p$, and y_i in $\mathbb{R}$:

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}}.$$

What is specific to multilayer neural networks?

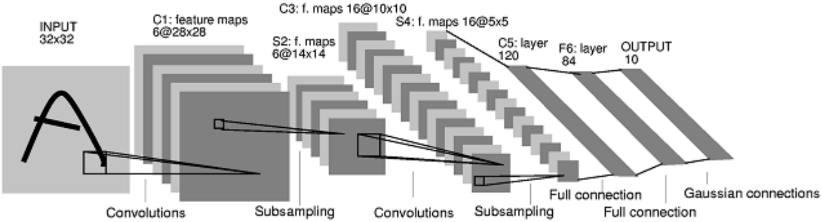
- The "neural network" space $\mathcal{F}$ is explicitly parametrized by:

$$f(x) = \sigma_k(\mathbf{A}_k \sigma_{k-1}(\mathbf{A}_{k-1} \ldots \sigma_2(\mathbf{A}_2 \sigma_1(\mathbf{A}_1 x)) \ldots)).$$

- Linear operations are either unconstrained (fully connected) or involve parameter sharing (e.g., convolutions).
- Finding the optimal $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$ yields a **non-convex** optimization problem in **huge dimension.**

Convolutional Neural Networks Behind the Scenes

Picture from LeCun et al. [1998]



What are the main features of CNNs?

- they capture **compositional** and **multiscale** structures in images;
- they provide some **invariance**;
- they model **local stationarity** of images at several scales;
- they are **state-of-the-art** in many fields.

Convolutional Neural Networks in Front of the Scene

Picture from Olah et al. [2017]:

Edges (layer conv2d0) **Textures** (layer mixed3a) **Patterns** (layer mixed4a)

Convolutional Neural Networks in Front of the Scene

Picture from Olah et al. [2017]:

Patterns (layer mixed4a) **Parts** (layers mixed4b & mixed4c) **Objects** (layers mixed4d & mixed4e)

Future of Convolutional Neural Networks

What are current high-potential problems to solve?

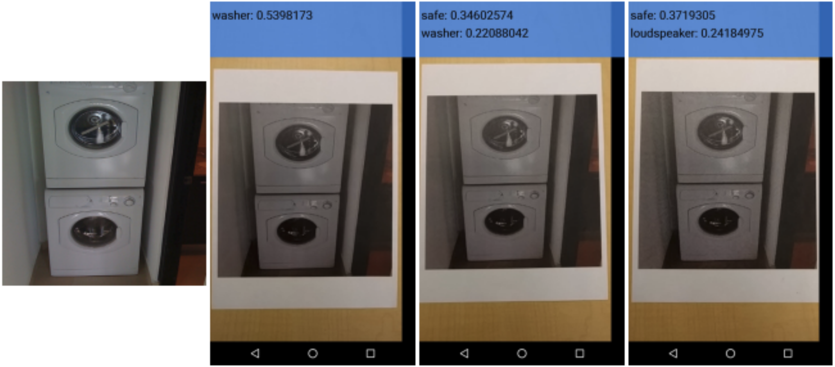
1. lack of **robustness** (see next slide).
2. learning with **few labeled data**.
3. learning with **no supervision** (see Tab. from Bojanowski and Joulin, 2017).

Method	Acc@1
Random (Noroozi & Favaro, 2016)	12.0
SIFT+FV (Sánchez et al., 2013)	55.6
Wang & Gupta (2015)	29.8
Doersch et al. (2015)	30.4
Zhang et al. (2016)	35.2
[1]Noroozi & Favaro (2016)	38.1
BiGAN (Donahue et al., 2016)	32.2
NAT	36.0

Table 3. Comparison of the proposed approach to state-of-the-art unsupervised feature learning on ImageNet. A full multi-layer perceptron is retrained on top of the features. We compare to several self-supervised approaches and an unsupervised approach.

Future of Convolutional Neural Networks

Illustration of instability. Picture from Kurakin et al. [2016].



(a) Image from dataset (b) Clean image (c) Adv. image, $\epsilon = 4$ (d) Adv. image, $\epsilon = 8$

Figure: Adversarial examples are generated by computer; then printed on paper; a new picture taken on a smartphone fools the classifier.

Future of Convolutional Neural Networks

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}} .$$

The issue of regularization

- today, heuristics are used (DropOut, weight decay, early stopping)...
- ...but they are not sufficient.
- how to **control variations of prediction functions**?

$$|f(x) - f(x')| \text{ should be close if } x \text{ and } x' \text{ are "similar".}$$

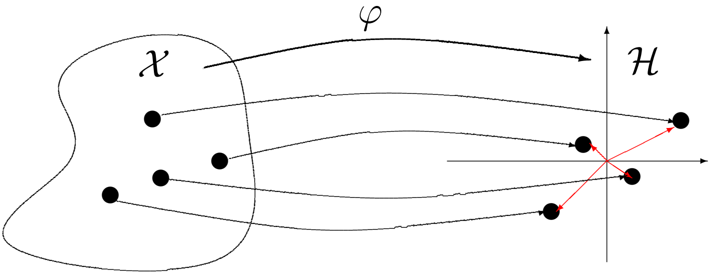
- what does it mean for x and x' to be "similar"?
- what should be a good **regularization function** Ω?

Back to the Past: Kernel Methods

$$\min_{f \in \mathcal{H}} \;\; \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \; + \; \lambda \|f\|_{\mathcal{H}}^2.$$

- **map** data x in $\mathcal{X}$ to a Hilbert space and work with **linear forms**:

$$\varphi : \mathcal{X} \to \mathcal{H} \qquad \text{and} \qquad f(x) = \langle \varphi(x), f \rangle_{\mathcal{H}}.$$



Back to the Past: Kernel Methods

$$\min_{f \in \mathcal{H}} \quad \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \; + \; \lambda \|f\|_{\mathcal{H}}^2.$$

Main purpose: embed data in a vectorial space where

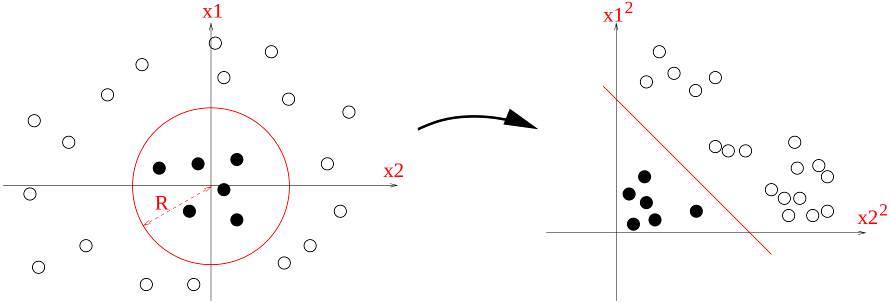
- many **geometrical operations** exist (angle computation, projection on linear subspaces, definition of barycenters....).
- one may learn potentially **rich infinite-dimensional models**.
- **regularization** is natural:

$$|f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \|\varphi(x) - \varphi(x')\|_{\mathcal{H}}.$$

Back to the Past: Kernel Methods

Second purpose: unhappy with the current Euclidean structure?

- lift data to a higher-dimensional space with **nicer properties** (e.g., linear separability, clustering structure).
- then, the **linear** form $f(x) = \langle \varphi(x), f \rangle_{\mathcal{H}}$ in $\mathcal{H}$ may correspond to a **non-linear** model in $\mathcal{X}$.



Back to the Future: Deep Kernel Machines

What is the relation with deep neural networks?

- it is possible to design functional spaces $\mathcal{H}$ where deep neural networks live [Mairal, 2016].

$$f(x) = \sigma_k(\mathbf{A}_k \sigma_{k-1}(\mathbf{A}_{k-1} \ldots \sigma_2(\mathbf{A}_2 \sigma_1(\mathbf{A}_1 x)) \ldots)) = \langle f, \varphi(x) \rangle_{\mathcal{H}}.$$

- we call the construction "**convolutional kernel networks**".

Why do we care?

- $\varphi(x)$ is related to to **network architecture** and is **independent of training data**. Is it stable? Does it lose signal information?

Back to the Future: Deep Kernel Machines

What is the relation with deep neural networks?

- it is possible to design functional spaces $\mathcal{H}$ where deep neural networks live [Mairal, 2016].

$$f(x) = \sigma_k(\mathbf{A}_k \sigma_{k-1}(\mathbf{A}_{k-1} \ldots \sigma_2(\mathbf{A}_2 \sigma_1(\mathbf{A}_1 x)) \ldots)) = \langle f, \varphi(x) \rangle_{\mathcal{H}}.$$

- we call the construction "**convolutional kernel networks**".

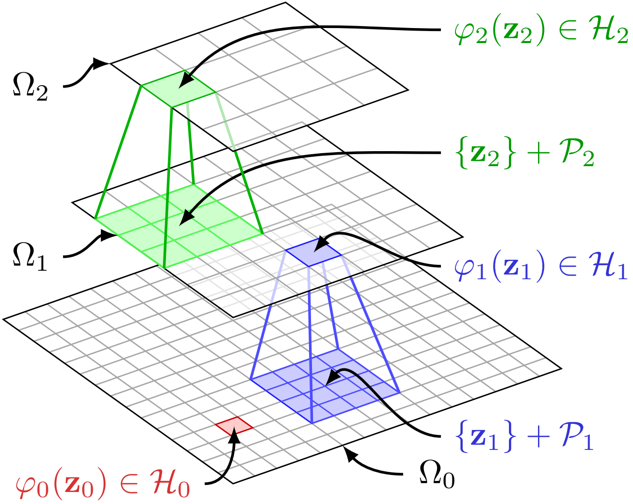
Why do we care?

- $\varphi(x)$ is related to to **network architecture** and is **independent of training data**. Is it stable? Does it lose signal information?
- f is a **predictive model**. Can we control its stability?

$$|f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \|\varphi(x) - \varphi(x')\|_{\mathcal{H}}.$$

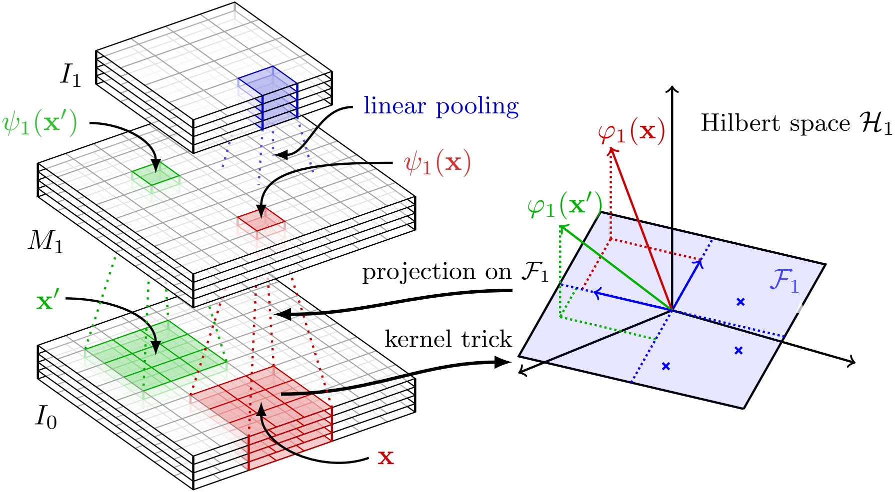
Back to the Future: Deep Kernel Machines

What is $\varphi(x)$?



Back to the Future: Deep Kernel Machines

Convolutional kernel networks in practice.



Back to the Future: Deep Kernel Machines

Technical details

Formally, a CKN is a sequence of operators

$$\Phi_n(x) = A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \ldots A_1 M_1 P_1 A_0 x.$$

- P_k performs **patch extraction**;
- M_k performs **kernel mapping**

$$K(z, z') = \|z\| \|z'\| \kappa \left(\frac{\langle z, z' \rangle}{\|z\| \|z'\|} \right).$$

- A_k performs **linear pooling** with a Gaussian filter.

The projection of a patch onto a finite-dimensional subspace yields a convnet-type of operation:

$$\psi(z) = \|z\| \kappa \left(W^\top W \right)^{-1/2} \kappa \left(\frac{W^\top z}{\|z\|} \right).$$

Back to the Future: Deep Kernel Machines

Short summary

- We have designed a **functional space** $\mathcal{H}$ to do deep learning.
- Approximation of the kernel map yields the **CKN model**, whose parameters can be learned **with or without supervision**.
- Each layer of CKNs perform a **geometrical operation** (projection).
- The functional space contains also **classical convolutional neural networks** with smooth homogeneous activation functions.
- For all these models $f(x) = \langle f, \varphi(x) \rangle$, and we study $\varphi(x)$ and f.

Back to the Future: Deep Kernel Machines

Short summary

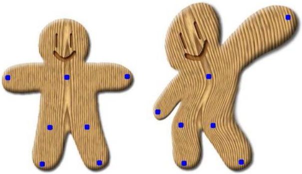
- We have designed a **functional space** $\mathcal{H}$ to do deep learning.
- Approximation of the kernel map yields the **CKN model**, whose parameters can be learned **with or without supervision**.
- Each layer of CKNs perform a **geometrical operation** (projection).
- The functional space contains also **classical convolutional neural networks** with smooth homogeneous activation functions.
- For all these models $f(x) = \langle f, \varphi(x) \rangle$, and we study $\varphi(x)$ and f.

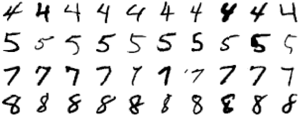
Performance of CKNs

- same as classical convnets in fully supervised setting (92% on CIFAR-10 with VGG-like architecture and simple DA).
- very competitive results for unsupervised learning on CIFAR-10.
- seems robust to learning parameter choices.

Back to the Future: Deep Kernel Machines

- $\tau : \Omega \to \Omega$: C^1-diffeomorphism
- $L_\tau x(u) = x(u - \tau(u))$: action operator
- Much richer group of transformations than translations





- Representation $\varphi(\cdot)$ is **stable** [Mallat, 2012] if:

$$\|\varphi(L_\tau x) - \varphi(x)\| \leq (C_1 \|\nabla \tau\|_\infty + C_2 \|\tau\|_\infty)\|x\|$$

- $\|\nabla \tau\|_\infty = \sup_u \|\nabla \tau(u)\|$ controls deformation
- $\|\tau\|_\infty = \sup_u |\tau(u)|$ controls translation

Back to the Future: Deep Kernel Machines

Proposition [Bietti and Mairal, 2017]

if $\|\nabla\tau\|_\infty \leq 1/2$ and Φ_n is the representation at layer n,

$$\|\Phi_n(L_\tau x) - \Phi_n(x)\| \leq \left(C_1\,(1+n)\,\|\nabla\tau\|_\infty + \frac{C_2}{\sigma_n}\|\tau\|_\infty\right)\|x\|$$

Remarks and additional results

- The result requires **small patches**, as in recent architectures.
- **signal recovery:** x can be recovered from $\varphi(x)$.
- It is possible to gain invariance to **any group of transformation**.
- For a given deep network

$$f(x) = \sigma_k(\mathbf{A}_k\sigma_{k-1}(\mathbf{A}_{k-1}\ldots\sigma_2(\mathbf{A}_2\sigma_1(\mathbf{A}_1 x))\ldots)) = \langle f, \varphi(x)\rangle_{\mathcal{H}}$$

the norm $\|f\|_{\mathcal{H}}$ is controlled by the product $\prod_i \|\mathbf{A}_i\|_2$.

Papers

First model (not the right one)

- J. Mairal, P. Koniusz, Z. Harchaoui and C. Schmid. Convolutional Kernel Networks. NIPS 2014.

The right model with unsupervised and supervised learning

- J. Mairal. End-to-End Kernel Learning with Supervised Convolutional Kernel Networks. NIPS 2016.

Theoretical foundations

- A. Bietti and J. Mairal. Group Invariance, Stability to Deformations, and Complexity of Deep Convolutional Representations. preprint arXiv:1706.03078. 2018. (also NIPS 2017).

Practical application to biological sequences (ongoing work)

- D. Chen, L. Jacob, and J. Mairal. Predicting Transcription Factor Binding Sites with Convolutional Kernel Networks. preprint BiorXiv. 2017.

Conclusion and Perspectives

Stability and generalization are related through **regularization**. There are two types of perpectives for this approach:

For existing deep networks

- new regularization functions, along with algorithmic tools to learn with **less labeled data**, and obtain **more stable** models?

For designing new deep models

- design deep models that are **stable by design** and that are **easy to regularize**? $\Rightarrow$ We already have models that are stable w.r.t hyper-parameter choices.

References I

Alberto Bietti and Julien Mairal. Group invariance and stability to deformations of deep convolutional representations. *arXiv preprint arXiv:1706.03078*, 2017.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *P. IEEE*, 86(11):2278–2324, 1998.

J. Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Adv. NIPS*, 2016.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. 2017.