Deep learning and Image Classification

advisor: Stéphane Mallat
following the works of Laurent Sifre, Joan Bruna, ...
High Dimensional classification

\((x_i, y_i) \in \mathbb{R}^{512^2} \times \{1, \ldots, 100\}, i = 1 \ldots 10^4 \rightarrow \hat{y}(x)\)

Training set to predict labels

Rhino class from Caltech 101

Rhino

Not a rhino
High dimensionality issues

Translation

Rotation

$\|x - y\|_2 = 2$

Averaging is the key to get invariants
Image variabilities

Geometric variability
Groups acting on images:
translation, rotation, scaling

Class variability

Intraclass variability
Not informative

Extraclass variability

Other sources: luminosity, occlusion, small deformations

\[ x_\tau(u) = x(u - \tau(u)), \tau \in C^\infty \]
Fighting the curse of dimensionality

- **Objective:** building a representation $\Phi x$ of $x$ such that a simple (say euclidean) classifier $\hat{y}$ can estimate the label $y$:

\[
\hat{y}(x) = \hat{y}(x_0)
\]

- Designing $\Phi$ consist of building an approximation of a low dimensional space which is regular with respect to the class:

\[
\| \Phi x - \Phi x' \| \ll 1 \Rightarrow \hat{y}(x) = \hat{y}(x')
\]

- How can we do that?
Existing approach

- Unsupervised learning: Bag of Words, Fisher Vector, ... \[ \{x_1, \ldots, x_N\} \rightarrow \Phi \]

- Supervised learning: Deep Learning, ... \[ \{(x_1, y_1), \ldots, (x_N, y_N)\} \rightarrow \Phi \]

- Non learned: HMax, Scattering Transform. \[ \{G_1, \ldots\} \rightarrow \Phi \]
Separation - Contraction

- In high dimension, typical distances are huge, thus an appropriate representation must contract the space:
  $$\|\Phi x - \Phi x'\| \leq \|x - x'\|$$

- While avoiding the different classes to collapse:
  $$\exists \epsilon > 0, y(x) \neq y(x') \Rightarrow \|\Phi x - \Phi x'\| \geq \epsilon$$
**Invariance**

- Let $L$ be a variability that preserves the class, e.g.:
  \[ y(Lx) = y(x) \]
- One wish to build an invariant to this variability.
  \[ \hat{y}(\Phi Lx) = \hat{y}(\Phi x) \]
- A simple way to build an invariant would be linearizing the action of $L$, then projecting it via a first order approximation:
  \[
  \Phi Lx \approx \Phi x + \partial \Phi_x L + o(\|L\|)
  \]

**A linear operator**

\[
\Phi
\]

**Displacement**

+ projection
An example: translation

- Translation is a linear action:
  \[ \forall u \in \mathbb{R}^2, L_a x(u) = x(u - a) \]
- In many cases, one wish to be invariant globally to translation, a simple way is to perform an averaging:
  \[ Ax = \int L_a x da = \int x(u) du \]
- Even if it can be localized, the averaging keeps the low frequency structures: the invariance brings a loss of information!
Data

Wavelets

- Convolutions are the natural operators linked to translation:
  \[(L_a x) \ast f = L_a (x \ast f)\]

- A linear averaging can still build a (trivial) invariant:
  \[
  \int (L_a x) \ast f(u) du = \int L_a (x \ast f)(u) du = \int x \int f
  \]

- A wavelet is a complex localized filter that describes structures of images, with average 0. It is discriminative. Family of wavelets are obtained by rotating and dilating a mother wavelet \(\psi\):
  \[
  \psi_\lambda(u) = \frac{1}{|\lambda|} \psi(\lambda^{-1}u)
  \]

Morlet Wavelets
- Phase in color
- Amplitude in contrast
Wavelets to build invariance

• Interestingly, a translation displacement corresponds to a phase displacement, e.g.:

$$a \ll 1, (L_a x) \star \psi(u) \approx e^{i\phi(u,a)}x \star \psi(u)$$

• If one wish to build an invariant to this variability a modulus helps:

$$a \ll 1, |(L_a x) \star \psi(u)| \approx |x \star \psi(u)|$$

• It implies that more energy are in the low frequency domain: an averaging captures more information.

• Besides, it is possible to reconstruct signal from their modulus wavelet transform

Ref.: Phase retrieval for the Cauchy wavelet transform, Mallat S, Waldspurger I
Scattering Transform

- A Scattering Transform is the cascade of wavelets and modulus non-linearity:

\[ Sx = \{ \int x, \int |x * \psi_{\lambda_1}|, \int ||x * \psi_{\lambda_1}| * \psi_{\lambda_2}|, \ldots \} \]

- It is a deep cascade!

- Sort of super-SIFT.

- Observe that if no non-linearity is applied, the operator would be linear.
Properties

• Non-linear

• Isometric
  \[ \| Sx \| = \| x \| \]

• Stable to noise
  \[ \| Sx - Sy \| \leq \| x - y \| \]

• Invariant to translation
  \[ SL_a x = Sx \]

• Linearize the action of small deformation
  \[ Sx_{\tau} \approx Sx + \partial_x \nabla \tau \]

• Reconstruction properties

Ref.: Reconstruction of images scattering coefficient. Bruna J

Ref.: Group Invariant Scattering, Mallat S
## Benchmarks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Type</th>
<th>Paper</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caltech101</td>
<td>Scattering</td>
<td>Ask the locals</td>
<td>77.3</td>
</tr>
<tr>
<td></td>
<td>Unsupervised</td>
<td>DeepNet</td>
<td>96.4</td>
</tr>
<tr>
<td></td>
<td>Supervised</td>
<td>DeepNet</td>
<td></td>
</tr>
<tr>
<td>CIFAR100</td>
<td>Scattering</td>
<td>RFL</td>
<td>54.2</td>
</tr>
<tr>
<td></td>
<td>Unsupervised</td>
<td>DeepNet</td>
<td>80.4</td>
</tr>
<tr>
<td></td>
<td>Supervised</td>
<td>DeepNet</td>
<td></td>
</tr>
</tbody>
</table>

There is a gap with supervised architecture.
**DeepNet?**

- A scattering transform is a non-learned DeepNet

- A DeepNet is a **cascade** of linear operators with a point-wise non-linearity.

- Each operators is **supervisedly** learned

- **State of the art** on many problems in vision, audio, generation of signals, text, strategy, …
Benchmarks

ImageNet

CIFAR10

% accuracy

Ref.: image-net.org

Ref.: http://rodrigob.github.io/are_we_there_yet/build/
When do you want to do Deep Learning?

- When you have a lot of data
- When you do not need to have guarantee on why it works
- When you do not need to have guarantee it works or have enough data to validate it
- When you’re lazy and want quickly results
Generality

• For natural images, one can train a DeepNet on ImageNet: 1000 classes, 1,000,000 images.

• A DeepNet pretrained on ImageNet does generalise on Caletch’s datasets (10k images).

• Also, the deep features are often used in situations where there is no data. (e.g. medical imaging)

• Conclusion: pretrained features can be used as SIFT!

Ref.: Visualizing and Understanding Convolutional Networks, M Zeiler, R Fergus
How transferable are features in deep neural networks? Yosinski et al.
Architecture of a CNN

- Cascade of convolutional operator and non-linear operator:

\[ x_j(u, \lambda_2) = f \left( \sum_{\lambda_1} x_{j-1}(., \lambda_1) \ast h_{j,\lambda_1}(u) \right) \]

- Can be interpreted as neurons sharing weights:

- Designing a state-of-the-art deep net is generally hard and requires a lot of engineering
AlexNet

Ref.: ImageNet Classification with Deep Convolutional Network, A Krizhevsky et al.
Inception Net

Ref.: Going Deeper with Convolutions, C Szegedy et al.
ResNet

Ref.: Deep Residual Learning for Image Recognition, K He et al.
Optimizing a DeepNet

• The output $\Phi x$ of the DeepNet are optimised via the neg cross entropy:

$$- \sum_{x_n} \sum_i 1_{y(x)=i} \log(\Phi x_n)_i$$

• All the functions are differentiable: back propagation algorithm:

$$x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_n = \Phi x \rightarrow \text{loss}$$

$$\partial x_1 \leftarrow \partial x_2 \leftarrow \ldots \leftarrow \partial x_n = \partial \Phi x \leftarrow \partial \text{loss}$$

• It is absolutely non-convex! No guarantee to converge. Besides, many parameters: requires a lot of regularisation:

BatchNormalization, DropOut, data augmentation, and many data!

Ref.: Convolutional network and applications in vision. Y. LeCun et al.
DeepNet fancy claims

• In deep learning, deeper networks are better.

• "Highly non-linear" is better

• Your model must overfit your data to generalise

• Universal approximation theorem proves it will work

• Many inspiration from the "brain"
A black box

- **Pure black box**: few mathematical results explain how it works.

- **Hypothesis of low-dimensionality is wrong**. Ex: stability to diffeomorphisms

- **What is the nature of each operators? Necessary dimensionality reduction**
Softwares...

- All the packages are based on GPUs, select your favorite via: simplicity of benchmarking, coding, inputting the data, ...

- Torch: based on Lua, in a MATLAB style

- Tensorflow: Python friendly

![Diagram showing simplicity and modularity of different deep learning frameworks: Torch, Theano, Tensorflow, Caffe, MatConvNet. The diagram notes that simplicity to input the data and modularity are subjective.](image)
TensorFlow

• TensorFlow is a deep learning environment in Python/C++.

• It makes easier the data input via a system of queue and multi-threads

• Use on GPU/CPUs is transparent

• Typical code samples: here
TensorBoard!

- A very nice tool of visualisation, demo: here
Understanding deep learning
Two strategies:

- Structuring progressively the layers/ obtaining general properties on them
- Specifying a class of CNN that is interpretable and leads to state-of-the-arts results
Hybrid deep networks

Ref.: A Hybrid Network: Scattering and Convnet, submitted, EO

- Cascading a CNN on top of a Scattering Transform:
  \[
  \text{S} \rightarrow \text{CNN}
  \]

- **Claim**: incorporating geometric knowledge regularises the classification process

- It helps in situation with limited samples (ex: medical imaging)

- Interpretability of the first layer: local rotation invariance is learned.

![Accuracy on CIFAR10](chart.png)
A progressively better representation

- Progressively, there is a linear separation that occurs

\[ F_1 \rightarrow F_2 \rightarrow F_3 \rightarrow F_4 \]

- In fact, euclidean distances become more meaningful with depth.

Ref.: Building a Regular Decision Boundary with Deep Networks, submitted, EO

Indicates a progressive dimensionality reduction!

Ref.: Visualizing and Understanding Convolutional Networks, M Zeiler, R Fergus
Identifying the variabilities?

- Several works showed a deepnet exhibits some covariance:
  
  - Manifold of faces at a certain depth:

- Can we generalise these?

References:

- Understanding deep features with computer-generated imagery, M Aubry, B Russel
- Unsupervised Representation Learning with Deep Convolutional GAN, Radford, Metz & Chintalah
Conclusion

- Deep Learning architectures are of interest thanks to their outstanding numerical results yet not well understood properties.

- Scattering Network provides a first attempt to structure DeepNets.

- Check my website for softwares and papers: http://www.di.ens.fr/~oyallon/ (or send me an email to get the latest soft!)

Thank you!