Mathematical analysis of Convolutional Neural Networks

Ahmed MAZARI, Laurent CETINSOY, Hafed RHOUMA

January 19, 2017

Abstract

A summary of Understanding Convolutionnal Neural Network (CNN) By Stephan Mallat is Given. This article gives an interpretation of actions performed by Deep neural networks on input data space. A simplified version of CNN without channels recombination is first considered. It is shown that they can be seen as scatering trees and show invariance to translations and small diffeomorphism. It is shown that cascade scattering acts as linearization of the orbits action of symmetries of function f. In order to also get rotation invariance, the channels recombination are taken into account and studied with hierarchical wavelet scattering.

1 Introduction

A neural network with one hidden layer can be seen as a decomposition of ridge functions [4]. However, deep neural networks are characterized by a deep layer by layer complex interactions with a considerable numbers of neurons alternating between linearities and non linearities. This complex architecture brings us to a new mathematical world of high dimensional analysis.
In this study, we are going to explain the foundation of convolutional neural networks from group theory, stationary process and learning kernel (in a high dimension) perspective.

1.1 Supervised Learning

Supervised learning can be seen as an interpolation problem in which one seeks a mapping $f(x)$ from n training examples $\{x^i, f(x^i)\}_{i\leq n}$. In practice $x \in R^d$ where $d >> 1$. Trying to learn f without any assumption on the structure of f is doomed to fail because of the curse of dimensionality. Indeed with a naive approach one needs e^d samples to learn a mapping from $\mathbb{R}$.

1.2 Convolutional Neural networks

Convolutional neural networks is largely successful class of Deep Neural networks introduced by Lecun [1]. It consists of a succession of convolutional layers, followed most of the time by pooling layers.

For each convolutionnal layer a set of linear filters are convolved with the output of the previous layer and are inputs for non linear activation function. A convolution mask is illustrated in figure 1. These filters are learned by the network during the training phase.

In practice, a second hidden layer, called pooling, follows each convolutionnal layer. Their aim is to reduce the complexity of the model by sub-sampling the maps computed by convolutions.

If one neglects pooling, a CNN can be viewed as succession of linear filters (convolution) with non linearity (the neuron function activation) as depicted in figure 2. Several activation functions has been used like sigmoid. Recently, RELU has become the more popular one.

1.3 Representations, invariants and groups

Let us consider the task of digit classification on MNIST dataset [3]. One seeks to represent Φ with as much invariants as possible. Φ should have several invariance properties. Indeed a digit should be identically classified if it is translated, rotatated or smally distorded. Formally such properties can be expressed with group theory.
Let G a group of all symmetries :
$\forall (g, g') \in G^2 => g.g' \in G$.

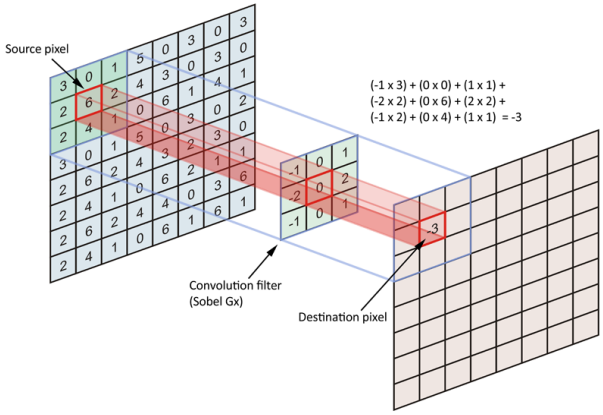

Figure 1: Illustration of a convolution mask. A dot product is computed between the filter and a moving window on the source signal.

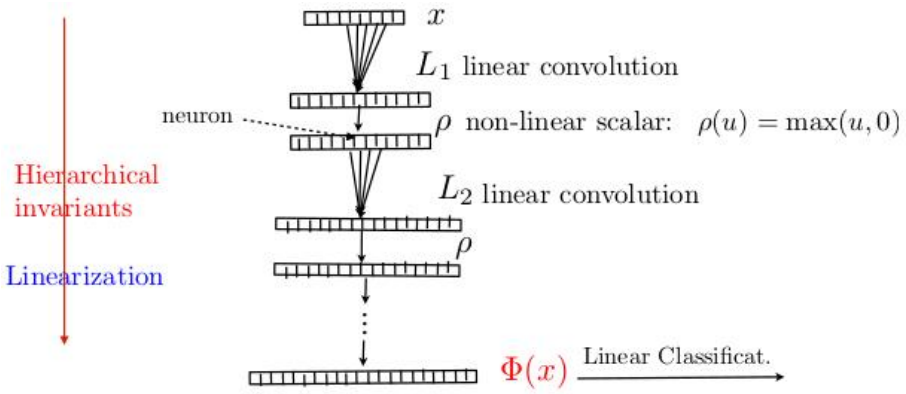

Figure 2: Architecture of CNN. Each layer computes a set of linear filters followed by a point wise non linearity. Pooling is ignored.

A group is a set of elements acting on the data. It is characterized by the following properties :

1. Inverse : each group has an inverse
   $\forall g \in G, g^{-1} \in G$
2. Associative :
   $(g, g').g'' = g.(g'.g'')$
   if commutative g.g'=g'.g : Abelian group
3. Dimension of group : group is of dimension n if it has n generators such that $g = g_1^{p_1} g_2^{p_2} ... g_n^{p_n}$
4. Lie group : infinitely small generators
   The movement defined by a generator can be infinitely small

The dimension of group shows the number of generators that allows to move in order to get all the elements of the group. If we are in n dimension, we have n generators to get g (when we iterate on n generators $p_1 ... p_n$ we get g).
Let T be the group of translations or rotations : for any g belonging to one of these groups, then one want to have $f(g.x) = f(x)$. It can easily be verified that a given symmetry is indeed a group: let $g_1, g_2 \in G$ where G is one of the above transformations then $f(g_1.g_2 x) = f(g_1.x) = f(x)$ besides such transformation has natural inverses. (trivial for translation, rotation and diffeomorphism are bijections).
Let $\tau(u)$ be the deformation of a signal. It depends on the position of the pixels for images. One seeks a representation which does not varies much with small deformations:
Diffeomorphism invariant $||\Phi(x) - \Phi()|| <= C sup_t |\nabla \tau(t)|| ||x||$ which preserves information.
In fourier transform to get translation invariants, take the modulus $|\check{x}| = |\check{x}|_c = \Phi(x)$ to kill the

phase because the modulus is invariant to translation:

$$\check{x}(\omega) = \int x(t).e^{-i\omega t}.dt \tag{1}$$

$$x_c(t) = x(t-c) => \check{x}_c(\omega) = e^{-ic\omega}\check{x}(\omega) \tag{2}$$

However, when it comes to deformation fourier transform is instable to small deformation such that :

$$x_\tau = x(t - \tau(t)) \tag{3}$$

High frequencies are instable to deformation. It's the reason why fourier transform is not used to solve classification problems.

Figure 3: Translations and small distortion of digits does not change their values.

Local symmetry :

$$\forall x \in \Omega, \exists C_x > 0, \forall g \in G : |g|_C < C_x, f(g.x) = f(x) \tag{4}$$

$|g|_C < C_x$ means that g is not far for the identity and C_x is a number which measure the size of the translations which will not change $f(x)$. So, this property explains that $f(x)$ remains the same when x is translated by a small translation which is the same as saying that f is locally invariant. g is a global symmetry of f if $\forall x, f(g.x) = f(x)$. One can see that global symmetries is a group. One can iterate on a given level sets thanks to the action group of symmetries of f. However such symmetries do not provide much invariance in Big DATA. In computer vision, a very important class of symmetry is small Diffeomorphism: they represent the small deformations of objects. One can easily understand that a digit will remains the same if a small deformation is applied to it. The question is how to extract these symmetries ? We define a Tradeoff between invariance by averaging and information loss.
Let $\Omega_t = \{x : f(x) = t\}$ be the level sets of f. In a regression task, level sets are the values. In classification, each t represents a class. we look at the level sets of $f(x)$ as classes. Learning $f(x)$ from x depends on the properties of $f(x)$: its symmetries (regularity) and invariants. If level sets Ω_t are not parallel to the linear space, they should be linearized with a change of variable $\Phi(x)$, then reduce dimension with linear projection. Some questions should be asked : What are the properties of this variable change ? and how to define a linear projector which preserves information ?
The variable change and linear projection allow to make a linear separation such that :

$$\Phi(x) = \phi_k(x)_{k \leq d'} \quad and \quad \langle\ \Phi(x), W\rangle = \sum_k w_k \phi_k(x). \tag{5}$$

Rather than doing a change of variable at once. Deep neural networks do it progressively in the way that the more we go in depth the more we kill variability and create hierarchical invariants. Local symmetry preserves class when we move points. A symmetry is an operator g which preserves level sets such that : $\forall x, f(g.x) = f(x)$. If g_1 and g_2 are symmetries than $g1.g2$ is a also a symmetry. In order to be able to separate different level sets, these properties should be verified. One of the most difficult but important task is the linearization of symmetries. The idea is to come up with a change of variable $\phi(x)$ that linearizes the orbits $\{g.x\}_{g \in G}$ under a regularity condition which is called liptchiz continuous such that :

$$\forall x, g : ||\Phi(x) - \Phi(g.x)|| \leq C||g|| \tag{6}$$

This means that the distance $||\Phi(x) - \Phi(g.x)||$ in the euclidian space has to be in the order of the size of the group element $||g||$. The group that can be build are : group of translation, rotation and deformation. A question occurs, what are the other groups than can be built ? How to define $||g||$ for diffeomorphism ? Once can think about :

- Rotation and Deformation :
  group : $SO(2) * Diff(SO(2))$
- Scaling and deformation :
  group : $R * Diff(R)$
- Invariance to translation :
  $g.x(u) = x(u - c) => \Phi(g.x) = \Phi(x)$
- Small diffeomorphism :
  $g.x(u) = x(u - \tau(u))$
  Metric : $||g|| = ||\nabla_\tau||_\infty$ maximum scaling
  τ is a regular function of u that dilates or deflates u. The amount of dilation/deflation represents the maximum size of the jacobian of τ. It is defined as a weak norm of diffeomorphism. This property allows to get linearization by liptchiz continuity :

$$||\Phi(x) - \Phi(g.x)|| \leq C||\nabla_\tau||_\infty \tag{7}$$

- Discriminative change of variable :

$$||\Phi(x) - \Phi(x')|| \geq C^{-1}|f(x) - f(x')| \tag{8}$$

1.4 Wavelets

Wavelets are localized oscillating functions which can form a basis for function decomposition such as sine and cosine function for Fourier transform. They have zero mean, are squared integrable

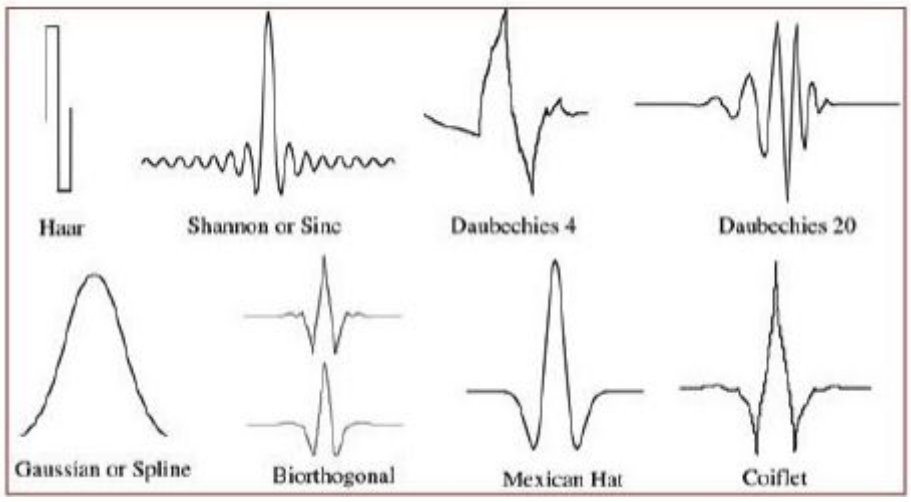

Figure 4: Examples of different wavelets

and fast decaying. Wavelet are stable to distortions. The wavelet transform of a given function f is then defined by

$$g(s, \tau) = \int f(t) * \phi(\frac{t - \tau}{s})dt \tag{9}$$

Wavelets can be computed with filter banks, It is called a fast wavelet transform:
let $w_{j,0}$ be a low pass filter and $w_{j,k}, 1 < k <= K$ a set of K band pass filters, then a wavelet transform of signal x is obtained by :
SIFT : modulus and average
Scattering : $Scat(x(u))$ and $Scat(x(u - \tau))$ which has an upper bound. Contrary to fourier transform, killing phase on wavelet transform does not kill information thanks to redundancy.

1.5 Stationary processes

A Stochastic process is a sequence of discretely indexed random variables X_t. Several processes exist such as Bernoulli Processes, Random Walk or Markov chains.
Let F be the cumulative distribution function of the joint distribution $X_t, X_{t_1}, ..., X_{t_k}$. A stochastic process strongly stationary if $F(x_{t_1+\tau}, ..., x_{t_k+\tau}) = F(x_{t_1}, ..., x_{t_k})$.
A stochastic process is weakly stationary if the expected value and the autocovariance of the process do not vary with time.
Let $\mu_X = \mathbb{E}$. A process is said to be ergodic if $\tilde{\mu}_X = \frac{1}{N}\sum X_t$ converges to μ_X as N increases to infinity.
A texture is a stationary process. The scattering transform of a stationary process $X(t)$ is as follow :

$$S_J X = \begin{pmatrix} x * \Phi_{2^J}(t) \\ |x * \psi_{\lambda_1}| * \Phi_{2^J}(t) \\ ||x * \psi_{\lambda_1}| * \lambda_2| * \Phi_{2^J}(t) \\ |||x * \psi_{\lambda_1}| * \lambda_2| * \lambda_3| * \Phi_{2^J}(t) \\ ... \end{pmatrix}_{\lambda_1,\lambda_2,\lambda_3} \tag{10}$$

such that λ_i represents the wavelet .
If we apply convolution to this stationary process, it remains stationary because covoluating a signal with a wavelet is covariant to translation and averaging remains also stationary. When averaging goes to infinity, it converges to the mean and to the gaussian distribution thanks to Central Theorem Limit under weak ergodicity assumption :

$$Gaussian\ Distribution \quad {}_J X \sim \mathcal{N}(\mathbb{E}(SX), Cov_J \to 0) \tag{11}$$

and converges to the expected value $\mathbb{E}(SX)$ which is called the scattering moment:

$$\mathbb{E}(SX) = \begin{pmatrix} \mathbb{E}(X) \\ \mathbb{E}(|X * \psi_{\lambda_1}|) \\ \mathbb{E}(||X * \psi_{\lambda_1}| * \psi_{\lambda_2}|) \\ \mathbb{E}(|||X * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \psi_{\lambda_1}|) \\ .... \end{pmatrix}_{\lambda_1,\lambda_2,\lambda_3} \tag{12}$$

The fact that we contract, variance goes to zero. Now, if we would like to reconstruct the gaussian process, we compute $\check{x}$ which minimizes $||S_j\check{x} - S_j x||^2$.

2 Texture classification with stationary process

Wavelet scattering : computes average and wavelet transform at each stage. For any path important : scattering coefficient are invaratiants to translation and lipschitz continous to deformations. For the scaterring order: band pass filters measure interactions between variations of x at a scale 2^{j_1} within a distance 2^{j_2} and along orientation of frequency bands defined by k_1 and k_2 which are the channels.
In this section a simpler model of CNN is analyzed in which channel recombination have been removed. IE. Where pooling layers have been removed. In this setting at a given layer:

$$x_j = \rho W_j x_{j-1} \tag{13}$$

The convolution cascade can be replaced by $m\ band-pass\ filters$ convolutions. To avoid variance issues and loss of information, channels recombination is needed. Since the network is structured by factorizing groups of symmetries as depth increases, it implies that all linear operators can be written as generalized convolutions across multiple channels. When we apply translation, displacement are obvious. One can wonder how to preserve classes due to these displacements. To do so, classes are preserved by local symmetries as follow :
They are transformation $\check{g}$ such that $f_{j-1}(x_{j-1}) = f_{j-1}(\check{g}.x_{j-1})$. $\rho(W)$ computes an approximate mapping of such transformation $\check{g}$ into a parallel transport which moves coefficients of x_j.
High dimensional learning is characterized by an intra-class variability, for instance : a class of chair contains a set of chairs with different styles, so a huge variability within the class arises. To circumvent that, the group of local symmetries H_j is associated with complex transformations

when j increases which allow to capture large transformations between different patterns in the same class. The distance of examples from the same class defines weighted graphs which sample underlying continuous manifold in the space. The symmetry of H_j is associated with common transformation over all manifolds of examples x_j^i which preserves the class while capturing large intra-class variance. These manifolds learn progressively regular structure in high dimension thanks to the properties of invariance, covariance and group of sysmmetries.
In order to build invariants we focus on the phase of a the signal x because it has the information about variation (oscillation). At this stage we need to introduce non-linearity such as $ReLu$:
$\sigma(\alpha) = |\alpha| = |x * \psi_{2^j,\theta}(u)|$: eliminates phases which encodes local translation then we get sparse representation as it's depicted in figure 5 . We get invariants to translation by averaging $x * \phi_{2^j}$

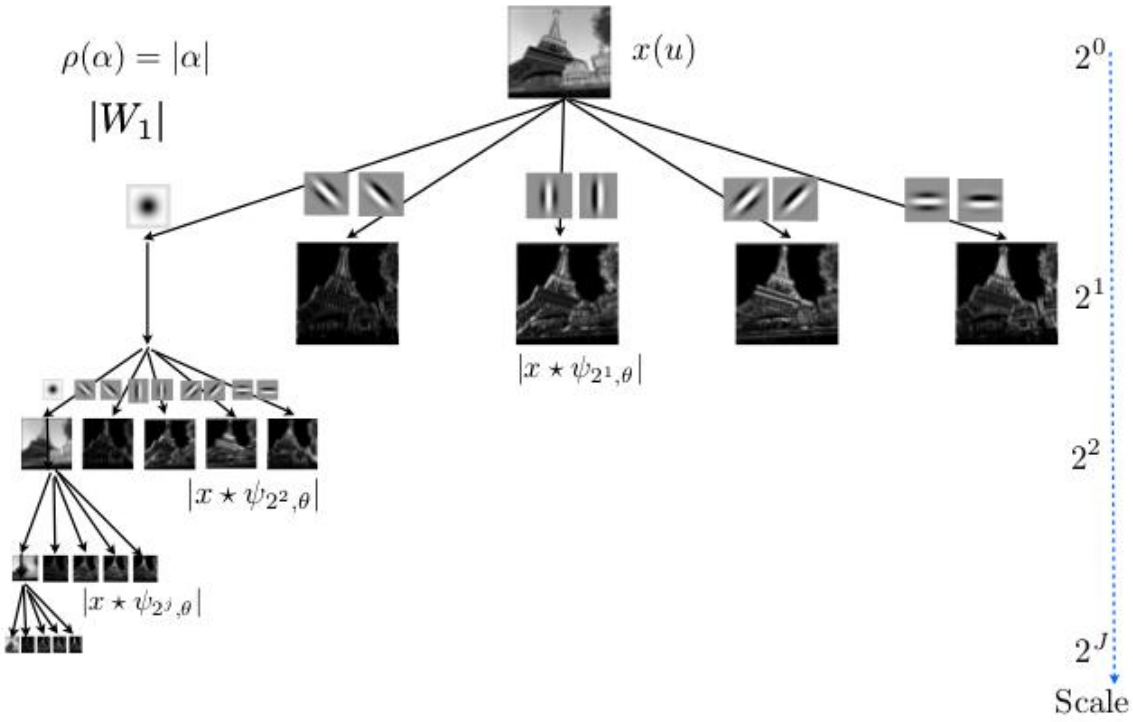

Figure 5: Sparse separation with wavelets

but we lose informations due to this linearization by averaging as illustrated in 6. In order to build invariants that preserve information, we apply wavelets at the coordinates where the information is lost because they are stable to small deformation. The process is to extract the variability using wavelets, kill the phase and get $(|x * \psi_{2^j}(t)|)$ which is an invariant to translation by averaging , the process is called Scale-Invariant Feature Transform (SIFT). However, we still lose lots of informa-

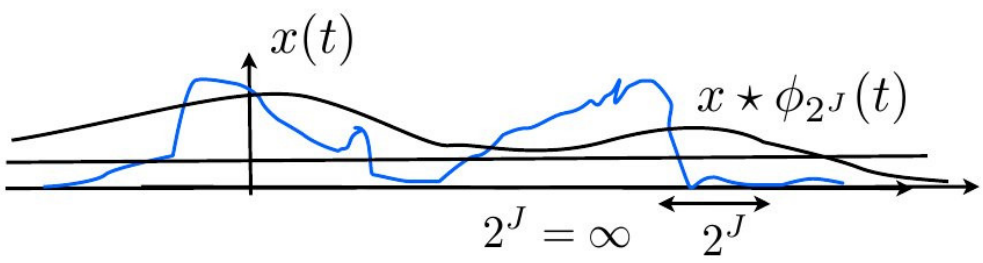

Figure 6: Translation invariance creation by local and global averaging. If $J = \infty$ the averaged signal is completely invariant to translation but all information is lost.

tion due to averaging.

In order to retrieve back the information lost after averaging, we extract the variation of the signal by applying again a wavelet transform. In other term, this double process is called convolutional neural network :
$|W_2||x * \phi_{\lambda_1}| = (|x * \phi_{\lambda_1}| * \phi_{\lambda_2}(t)|)_{\lambda_2}$.
We define : $S_J X$ the interaction across scales
$S_J X = ||x * \psi_{\lambda_1}| * \psi_{\lambda_2} * ..| * \psi_{\lambda_m}| * \phi_{j\,\lambda_K}$.
$S_J X$ creates invariants by extracting component at different scale, looking at their interactions then averaging them.
$S_J X$ is equivalent to

$$S_J X = \begin{pmatrix} x * \phi_{2^j} \\ |x * \psi_{\lambda_1}| * \phi_{2^J} \\ ||x * \psi_{\lambda_1}| * \psi_{\lambda 2}| * \phi_{2^J} \\ |||x * \psi_{\lambda_1}| * \psi_{\lambda 2}| * \psi_{\lambda 3}| * \phi_{2^J} \\ |... \end{pmatrix}_{\lambda 1, \lambda 2, \lambda 3} \tag{14}$$

theorem of wavelets proves that the energy of $S_J X$ is the same as the original signal x :
$||S_J X|| = ||x||$ and
invariance to translation and deformation linearization are ensured :
if $x_T(u) = x(u - T(u))$ then :

$$lim_{j \to \infty} ||S_j x_T - S_j x|| \leq C ||\nabla_T||_\infty ||x|| \tag{15}$$

3 CNN with channel recombination

A scattering is implemented by a deep convolutional neural network which have a specific architecture. As opposed to standard convoluionanl networks, output scattering coefficients are produced by each layer as opposed to the last layer. Filters are not learned from data but are predefined wavelets.
In this section, the limitation of wavelets scattering of the translation are overcome with channels recombination.
The network is defined by the factorization of groups of symmetries along network layers. In addition, in order to preserve classification margin, wavelets are replaced by adapted filter weights. This theorem explains that when we deform the original signal and look its effect on the vector $S_J X$ the distance is at the order of length of deformation ($C||\nabla_T||_{\text{inf}}\ ||x||$) which means that the phenomena is almost locally linearized. The channels recombination will linearize the groups of translation and rotation symmetries. In the context of classification, we are looking for invariant and covariant representations. In one hand, invariant representations allow to reduce dimension and kill variability. In other hand, covariant representations accept to be translated. In other words, building something covariant and linear implies convolution.
Invariance and covariance are important properties in order to get stable representation for classification. In order to build invariant representation, it's fundamental to satisfy the property of covariant to translation. It can be explained by the fact that getting invariant representations which are obtained by time averaging requires to intermediate representation and operator to be covariant.
One of the power of deep networks is the channels combinations accross channels.

$$x_j(u, K_j) = \rho(\sum_K x_{j-1}(., K) * h_{Kj}(u)) \tag{16}$$

They allow to build invariance and convariance to various groups (rotation, translation, deformation, sclaling) and linearize the built symmetries.
In the context of wavelet, invariance to rotations are computed by covolution along the rotation variable with wavelet filters because filters are indexed by scale(wavelet) and by rotation. It implies that wavelet are invariant to translation and rotation which is called rigid movement. The difference between group of translation and group of rigid movement is that group of rigid movement is not commutative (translation and rotation don't commute) because on the rotation effect of translation. This is why, in the rigid movement we need to capture the variability of spatial directions.
The power of deep neural networks can be explained by the ability to factorize the group of symmetries : $g_1\ g_2\ g_3$ ... DNN progressively builds given groups. In the first layer,it gives a group of

translation because image is indexed by translation. Then, it builds up wavelet with rotation and translation and then it can create other groups. At each layer we introduce a new group which still propagates to deeper layer thanks to covariance. This latter allows to propagate the whole formed group to deeper layers. In other words, if we mix several groups, we have one group which is a product of several groups. For instance, we set g_1 : group of translation and g_2 : group of rotation then $g_1.g_2$ is a roto-translation group. This process is called symmetry factorization with respect to generators incorporation

4 Conclusion

A mathematical framework based on wavelet was provided to better understand the process happening in Deep Covolutional Neural Network.
Convolution Neural Networks linearize non-linear transformation (symmetries), reduce dimension with projections and learn invariant to translation, to rotation, to scaling, to elastic deformation and to rigid movement.
They have high dimensional approximation capabilities and learn hierarchical invariants of complex symmetries.
However, we lack of mathematical understanding to build a grounding theory of deep learning. It seems to be tempting to retrieve back notions of complexity, regularity and approximation theorem. Several follow-up questions to understand the intrinsic structure of deep neural networks remain open problem :

- Can we define a class of high dimensional regular functions that are well approximated by deep neural networks ?
- Can we characterize the regularity of $f(x)$ from the factorized symmetry groups ?
- Can we recover symmetry groups from the learned matrices ?
- What is the functional analysis needed for high dimensional learning ?

References

[1] Le Cun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W, Jackelt L. 1990 Handwritten digit recognition with a back-propagation network, In Proc. of NIPS.3. 1, 7

[2] Stéphane Mallat. Understanding deep convolutional neural networks.

[3] Yann LeCun, Corinna Cortes, J.C. Burges.MNIST DATABASE of handwritten digits

[4] Candés E, Donoho D. 1999, Ridglets : a key to high dimensional intermittency ? Phil. Trans. Roy. S. A 357. 1