Notes and Comments on S. Mallat's Lectures at Collège de France (2020)

Jean-Eric Campagne

Notes and Comments on S. Mallat's Lectures at Collège de France (2020)

Multiscale Models and Convolutional Neural Networks

J.E Campagne *

Feb. 2020; rév. 3 novembre 2023

*If you have any comments or suggestions, please send them to `jeaneric DOT campagne AT gmail DOT com`

Table des matières

1. Foreword

***Disclaimer**: What follows are my informal notes in French, translated into rough English, taken on the fly and reformatted with few personal comments ("NDJE" or dedicated sections). It is clear that errors may have crept in, and I apologize in advance for them. You can use the email address provided on the cover page to send me any corrections. I wish you a pleasant read.*

Please note that the Collège de France website has been redesigned. You can find all the course videos, seminars, as well as course notes not only for this year but also for previous years [1].

I would like to thank the entire Collège de France team for producing and editing the videos, without which the preparation of these notes would have been less convenient.

Also, note that S. Mallat [2] provides open access to chapters of his book *"A Wavelet Tour of Signal Processing"*, 3rd edition, as well as other materials on his ENS website.

This year, 2020, is the third in the cycle of S. Mallat's Data Science Chair, and the theme is: **Multiscale models and convolutional neural networks**.

2. Lecture 22 Jan.

2.1 Thematic Introduction to the Course

A series of questions arise concerning learning systems from an applied mathematics perspective, and particularly this year regarding neural networks.

2.1.1 Understanding a Learning System?

What do we mean by "understanding" in the context of Applied Math.? Indeed, according to the engineers who use them, they fully understand the algorithms they

1. https://www.college-de-france.fr/chaire/stephane-mallat-sciences-des-donnees-chaire-statutaire/events
2. https://www.di.ens.fr/~mallat/CoursCollege.html

implement and the architectures of the networks that are completely specified. On the other hand, we do not truly understand the "why it works." That is, we do not understand, for example, neither the generalization performance, nor when the network will provide very high accuracy for any example for which we estimate a function $f(x)$. Therefore, the notion of understanding in the mathematical sense is not the same as in the engineering sense, for example.

That being said, when we begin to study subjects such as hearing and vision from a physiological point of view, we quickly realize the marvel of Nature that has developed highly sophisticated systems from receptors to the central nervous system for information processing and vice versa. After a while, faced with the complexity of the system, we think it is too complicated and that we should focus on the learning itself, which seems more approachable. Of course, this seems very frustrating and, in some respects, unsatisfying not to delve into the details of how these systems work. But the hope is that it is simpler, however, as we will see, it is not. It is not simpler at all to understand this learning loop, especially when we want to delve into the problem beyond algorithms.

From a very practical point of view, there are problems:

— **Robustness**: Often, systems are not very stable and lead to quite dramatic errors (e.g., in autonomous vehicles, in medical applications, etc.). How can we improve this robustness?

— **Efficiency**: The amount of data required for learning, the energy required to perform this learning in a reasonable time (cf. GPU farms), are entirely prohibitive. Is this optimal or not at all?

— **Control**: The architecture will allow learning a certain type of problem, but the time to bridge the gap between architecture specifications and the type of problem represents 99% of the design time.

This last aspect will be at the heart of the course, and we will see it from the concept of **a priori information**. What a priori information do we have, how can we express this information mathematically? So, what we will see in the future sessions is:

1. what is the link between **architecture** and **generalization**,
2. that these tools (e.g., NN [3]) can be used as **tools for exploring complexity**. This is a topic that goes beyond this course. For example, we take a learning system (NN, DT,

3. Note: NN for Neural Network, DT for Decision Tree, RF for Random Forest

kernel methods...), we try to calculate physical functions (e.g., energy of a system), and we see if it works. If it does, it teaches us something about the complexity of the underlying functional. For example, teams in quantum chemistry calculate the energy of molecules, which allows them to access their stabilities or other properties, and thus access the properties of materials. So, if we can do this with an NN, then we can go back to physics and ask questions about why a particular property of the molecule is the way it is. Currently, we are completely in the empirical realm; there are systems that work, but we do not understand the connection to the Schrödinger equation. Thus, we are witnessing a new perspective on chemistry, somewhat similar to statistical physics, where we look at the system as a whole. We identify "macro-variables" calculated by these networks that can specify the energy of the molecule. The question is, "Why?" The complexity is probably less than we thought. In short, this is how learning systems are tools for exploring complexity.

3. understanding the learning of these network architectures. Here, after spending a lot of time designing network architectures, we begin to wonder if we could not "learn these architectures" after all. There are a number of meta-parameters that define these architectures (e.g., number of layers, number of neurons per layer, type of layer...), so why not consider a high-level learning system that learns to find the best architecture for a particular application. We could think of a genetic (Darwinian) algorithm that selects the right solution through transformations/mutations of the system.

The last point is a level of questioning that we will not address this year, but it is part of the landscape of all the questions that can be addressed.

2.1.2 The Role of Mathematics

Before going any further, we can address the role of mathematics in the problem of learning. Indeed, we have a perspective on linguistics (see the 2019 course), and now we have systems capable of translation, text analysis to recognize the author, text generation, and more. There is a sort of infinite loop where we ask ourselves why the mathematical language is so effective in describing natural phenomena. The viewpoint we take here is that of applied mathematics, and this language evolves as we ask questions. In "pure" mathematics, the problem lies within the field of mathematics, while in "applied" mathe-

matics, questions come from outside the disciplinary field. Here, in a sense, the language of applied mathematics evolves like a learning system where data (input) is the problems posed. In this sense, until the end of the 19th century roughly, mathematical questions were underpinned by problems in physics, so it is not surprising that mathematics is remarkably suited to describing physical phenomena.

In the case at hand, the problems are very complex, and mathematics needs to evolve. The disciplinary field at work here is that of **very high dimension**, which deals with functions with a very large number of variables. It is very transversal within today's mathematics: probability, analysis (harmonic) [4], etc. Questions in this field of very high dimension are open, and research is very active in evolving mathematics.

2.1.3 Principles of Architectural Organization

The problem, as we saw, for example, in the 2018 course, is the high dimension: a typical image has millions of pixels, similarly, a sound has millions of samples per second, text has a million characters, and even worse, a mole has by definition 10^{23} entities. Therefore, Physics/Chemistry has always been the science of very high dimension... How to approach this problem? In fact, we will implement four principles: **separability, symmetries, and sparsity** [5], and a fourth that can be called a meta-principle, which is **evolution**.

An idea that immediately comes to mind is to perform **dimensionality reduction** using *a priori*, for example, by trying to find forms of **separability**. Thus, in the (simple) case of an image, **interactions between pixels** are essentially **local** [6]. In this case, we can treat patches separately (though we should be cautious about seams) and then combine individual results to obtain a solution to the original problem.

That said, there are problems that require larger structures for which we do not have local separability. But in most cases, we can introduce **separability between various scales** of the problem (cf. **hierarchy**). This scale separability is very common in Physics [7],

4. In analysis, we will realize that a priori knowledge is deduced from very little, and the data is structured on spaces with low topological dimension (time, images), allowing for extensive analysis.

5. See, for example, the 2019 course Sections 3.6, 3.7, 3.8.

6. Note: We can conceive this reasoning after removing a global contribution that would influence the entire field of view.

7. It's, for example, Cartesian reductionism of 1648 in his *Treatise on Man*.

allowing for dimensionality reduction of the problem, so even if interactions between scales may complicate it, it remains solvable. We will see tools from harmonic analysis (time-frequency/wavelets).

Another point is the search for **symmetries**, which is a fundamental ingredient, especially in Particle Physics, as it allows for finding **invariants** (cf. Emmy Noether's theorem in 1915). For example: spatial symmetries that lead to invariance by translation/rotation/flip. Once the symmetry is identified, the dimensionality of the problem can be reduced again. A simple example: a 2D problem invariant under rotation symmetry is described by a function in which the angle no longer appears in its list of variables. Of course, this reduction is more effective the more symmetries there are. From a mathematical point of view, we will use **group theory**, including Lie groups for continuous symmetries.

The last organizational principle is **sparsity**. Note that in the field of recognition, it was dominated by the theme of pattern recognition before the era of learning. Thus, the main subject was the concept of structure [8] that needed to be recognized: e.g., a dog's/cat's ears, eyes in a face... This approach is very well understood, and the applications developed by engineers are often rediscovered after many detours by mathematicians wanting to start from scratch. And once again here, understanding these structures also allows for dimensionality reduction. It is characterized by a number of elementary structures that is much smaller than the initial dimension, so the goal is to decompose the problem into these **elementary structures**. A point in passing, this decomposition into elementary structures may require a "search for symmetries" reduction beforehand. Thus, one must keep in mind the different types of organizational principles. However, it is in the context of discovering structures that learning will play its role.

Above all these principles, there is the notion of **evolution**. Indeed, for learning, we set the goal of minimizing risk, which will evolve the network's parameters towards a solution. The path of minimization can be considered as a **time variable**. This variable is viewed differently depending on the case. Thus, for signal processing problems $x(t)$, time is an **indexing of the digitized signal**. In this case, time has no particular property compared to a spatial variable, for example.

8. See in 2019 Section 2.3, which deals with the influence of Noam Chomsky's semantics in this field, for example.

In many physics subjects, we may want to describe not the state of the system, which is too complex, but rather its evolution. For example, in Mechanics (Quantum or not), the differential equation governed by a Hamiltonian plays a crucial role:

$$\frac{\partial x(t)}{\partial t} = H(t)x(t)$$

Time is viewed through **evolution operators**.

There is a third way to view time, namely a form of event indexing, i.e., time viewed from the perspective of **coincidence**. This occurs, for example, when we want to track an object of a particular color in a series of photographs. What fundamentally changes is the light intensity at the boundary of the object in question (cf. the background remains unchanged during the shooting). Therefore, conversely, if in a series of photographs we observe "temporal" changes in brightness for several pixels in "coincidence", then we can legitimately ask whether all of these pixels belong to the same structure. So, the idea here is that behind "temporal" coincidences, we can detect structures. This is not trivial; indeed, a baby a few days old, whose vision is not yet fully operational, is capable of recognizing its surroundings. Presumably, this is through its hand that it learns, because by grasping and its complex movement, the hand creates discontinuities. In the case of machine learning, this notion of coincidence is a research axis to reduce the number of training examples. Currently, the number of examples typically reaches 100k to 1M, which is much too high compared to natural systems. So, there is room for improvement, especially since machine learning is currently rather static, i.e., examples are treated one after the other quite independently [9], whereas we need to introduce dynamics and cohesion.

2.1.4 Prior Information

This will be one of the main themes of this year because ultimately, one cannot learn without prior information. First of all, prior information is translated in terms of **classes of hypotheses**. Given data, denoted as x, we want to find a function f such that it can produce y, i.e., $y = f(x)$. The problem is, therefore, to approximate f, which is not known. Thus, the hypothesis typically translates into defining a set $\mathcal{H}$ for which the

9. However, recurrent neural networks, LSTM, GRU, etc., should be mentioned.

elements $\tilde{f}$ satisfy various properties, such that the risk $\mathcal{R}(f, \tilde{f})$ is small. In simple cases, the risk is a norm, and $\|f - \tilde{f}\|$ is the approximation error.

So, prior information is concentrated on the set $\mathcal{H}$ in which we try to find the solution. In this framework, data will help us find the right function $\tilde{f}$ (note: through the minimization of empirical risk). The stronger the prior information, the more restricted the set of solutions, and the smaller the amount of data needed to find the solution. Conversely, if we do not have prior information, then the set $\mathcal{H}$ is so large that we face the curse of dimensionality (cf. 2018 Course) because we can never have enough data to find the solution.

As we have seen in previous years, explicitly defining $\mathcal{H}$ is essentially defining a level of **regularity** for the considered functions. However, prior information remains models, which must be questioned at some point about their levels of accuracy. One must neither lose sight of this "arbitrary" aspect nor forget that the results obtained are within the framework of these hypotheses. Also, one must keep in mind that there is a constant exchange between prior knowledge and the results (error) obtained, which can lead to a return to the hypotheses to narrow them down. It all depends on the number of samples/data available.

Until the 2005-10s, only prior information was used to define the system's structures, and in the end, a simple linear classification was done (e.g., to distinguish between dogs and cats). We will take this viewpoint to explore how far we can go and at what point it does not work anymore (cf. the initial hypotheses are too restrictive or the sets $\mathcal{H}$ are too large).

Finally, the course theme will be: **The Mathematical Nature of Prior Information in High Dimensions**. It will, therefore, be a question of what standard mathematical tools can teach us on this subject. Moreover, when approaching a subject, it is always interesting to ask oneself the following question: 1) what prior information do I have, 2) I try to create a linear classifier that includes this information, 3) I take a neural network and compare the results, 4) if the network gives me better results, how does it translate in terms of information that I wouldn't have seen *a priori*.

2.2 Supervised Learning in High Dimensions

In this course, we will only focus on supervised learning cases. We, therefore, place ourselves in a data space of very high dimension, $x \in \mathbb{R}^d$ with d very large (cf. $d \approx 10^{6-9}$). We will study two types of classical problems:

— **Classification**, where the class of labels is given by $f(x)$, which can also be quite large (e.g., 1000 for ImageNet). Moreover, we have n classified samples $\{x_i, y_i = f(x_i)\}_{i \leq n}$ (training set). The underlying problem here is the enormous variability within the same class.

— And **regression**, where the main difference is that $f(x)$ is not an index but a real number. The complexity is essentially the same because it comes from the high dimension d. In traditional Physics, to answer a question (e.g., the distribution of mass in a galaxy), fundamental forces are studied, which, for example, provide evolution equations and give the system's state at time t by integration. However, here, we do not have the fundamental equations, but we have prior information (e.g., system symmetries, continuity properties), and the question is whether we can deduce, for example, the system's energy for any configuration from a few known configurations. Here, we see that we have a completely different perspective from tradition: we do not start from fundamental interactions to design a model and answer the question, but we build an approximation from examples that will interpolate to answer the posed question.

The intuitively simple interpolation problem is very complicated for learning. In fact, to perform interpolation in x, one must, in a way, have known samples $\{x_i, y_i\}$ in the neighborhood of x to practice an "average." For example, let's say the x_i satisfy the following condition (Euclidean distance):

$$\forall x \in [0,1]^d, \exists x_i \in [0,1]^d \quad / \quad ||x - x_i|| \leq \epsilon \tag{1}$$

If the x_i are uniformly distributed, then at least ϵ^{-d} points covering the entire space $[0,1]^d$ are needed. In fact, one must realize that the points are very far from each other in very high dimensions. This theme was that of the 2018 course, it is the **curse of dimensionality**. So, to estimate $f(x)$ at a point $x \in \Omega \subset \mathbb{R}^d$, strong regularity of f on Ω must be imposed to interpolate it between very isolated points. The question is, what type of regularity is this?

However, if the points x_i accumulate on a manifold Ω of much lower dimension than d, then their distances will be much smaller. The problem in this case becomes "easy", for example, those that describe the motion of an articulated robot or those of very simple images (e.g., binary digit classification). But this type of problem is not what we are considering here, because consider the case of an image with 10^6 pixels of everyday life, the description requires a colossal number of variables to describe it. So, while x belongs to a subset of $\mathbb{R}^d$, it remains of high dimension, and if it is randomly sampled, it results in a completely structureless white noise image. So, we need to understand the regularity of f but within the framework of the space Ω that defines the set of images of the type we are interested in.

2.3 Change of Variables: Representation

Let's consider that we have a set of samples $\{x_i\}$ with 2 labels $\{0, 1\}$. What we would like is to have a function Φ from $\mathbb{R}^d$ to $\mathbb{R}^d$ that transforms $\{x\}_i$ into $\{x'\}_i$ in such a way that the boundary between the two populations is a hyperplane:

$$x = (v_1, \ldots, v_d) \xrightarrow{\Phi} \Phi(x) = \{x'\} = (v'_1, \ldots, v'_d) \tag{2}$$

The classification then boils down to finding the separating hyperplane, which means finding the vector w normal to the plane. The classifier has a simple expression (see Sec. 1.6 of the 2019 Course):

$$\tilde{f}(x) = \text{sign}\left\{\langle w, \Phi(x)\rangle + b\right\} = \text{sign}\left(\sum_k w_k v'_k + b\right) \tag{3}$$

which amounts to taking a linear combination of the coordinates of vector w with the variables of x in the representation given by Φ, and taking its sign.

In the literature, the elements $\{\phi^k(x)\}_{k \leq d'} = \Phi(x)$ are called *features*, and the classification operation resembles a vote among weak pieces of information from the problem to obtain a stronger decision $\tilde{f}(x)$. A change has been made: the nonlinear contour (boundary) in the original problem has been transformed into a linear contour. **Thus, we have expressed the regularity of the function $f(x)$ through the change of representation via** Φ. Furthermore, if this change of representation is simple, it corresponds to a very regular

function f, and conversely, the complexity of the function f is seen through the dimension of the function Φ.

Now, the very practical problem that arises is how to find $\Phi(x)$? Because once we have the variable transformation, finding the vector w is done simply by minimizing margin criteria as we have seen in previous years: see SVM methods, Ridge Regression [10]. So, the tough part is finding Φ. There are two viewpoints:

— The first one is quite extreme, it prevailed before around 2010 and relies solely on the *a priori* information we have about the function $f(x)$, and we encode it in the function $\Phi(x)$ in an attempt to linearize the problem.

— The second one consists of learning from data: that is, the neural network will learn $\Phi(x)$ while performing classification/regression.

2.4 Neural Networks

A little reminder [11], these "artificial" neural networks were introduced in the 1940s by W. Pitts and W. McCulloch, and then in 1957, F. Rosenblatt built a one-layer neural network capable of learning. It computes the coordinates of the vector w and a threshold b that completely defines the equation of a separating hyperplane. In Figure 1, the different stages of the Perceptron are depicted: the inputs, their weighted summation, then the rectifier/activator, and finally the output. Note that with the rectifier with a threshold (ReLU type), the output is sparse as it is zero as long as the weighted sum is smaller than the threshold.

Next, a **multi-layer network** is a stack of interconnected Perceptrons as shown in Figure 2. The last layer aggregates all the outputs $\Phi(x)$ to perform classification or regression. Learning involves optimizing inter-layer parameters (weights and biases) to minimize the error on examples. However, this is a challenging optimization problem, and stochastic gradient descent methods are used. The miracle, in a way, is that even though there are many minima, it works; the network is capable of generalization. From the perspective of **prior information**, it lies in the **architecture** of the network, and its design is what takes

10. See Sec. 4.2.4 of the 2019 Course
11. You can also review Sec. 4 of the 2019 Course as an introduction.

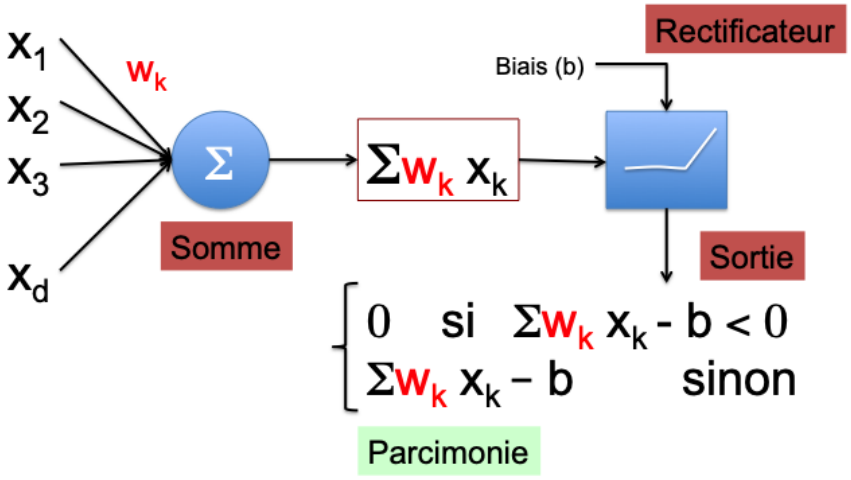

FIGURE 1 – Graphical representation of a linear classifier.

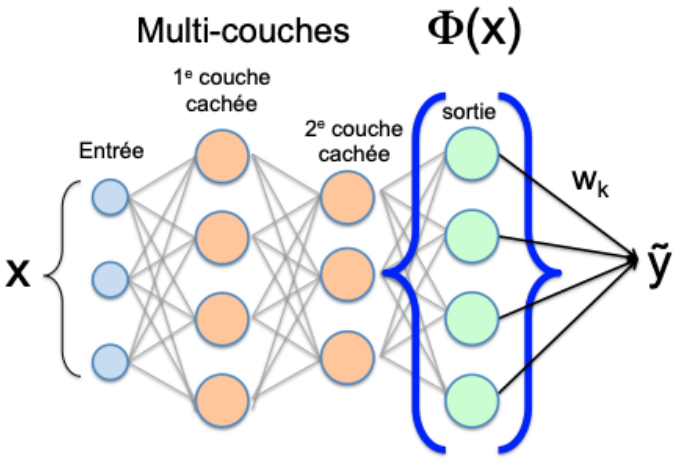

FIGURE 2 – Simplified diagram of a multi-layer neural network where the representation Φ is the result of all the sub-layers; it is used at the end of the network, for example, to perform classification.

the most time in the end. So, let's be clear about the following point: the network is not a system that learns all by itself; we introduced a *priori*.

A very important step in this context was the introduction of **convolutional networks** (or convnets) by Y. LeCun and J. Bengio in the 1990s [12]. At the heart of these networks is the concept of **filters** (see Figure 3) that rely on **translation invariance** in many problems. Translation invariance implies a *convolution operator*, hence the notion of a convolution filter. In Figure 3, this translates into the fact that the weights associated with filter F1, which handles typically 3x3, 5x5 patches, are the same for all patches of the original image x. Now, it is customary to stack different types of filters, these are the F_i. In the next layer, a new axis appears; it is the list of channels that allows mixing different filters (Figure 4). Therefore, the operator connecting the two layers (1 and 2) is not only translation-invariant but also, by taking all the small patches along the channels, it needs to fix the parameters along this new axis. The great difficulty is understanding the **nature of the mathematical operators** that act along this dimension.

Another important point in the architecture developed by Y. LeCun is that the

12. See Sec. 1.8 of the 2019 Course. Y. LeCun, J. Bengio, and G. Hinton received the Turing Award in 2019.

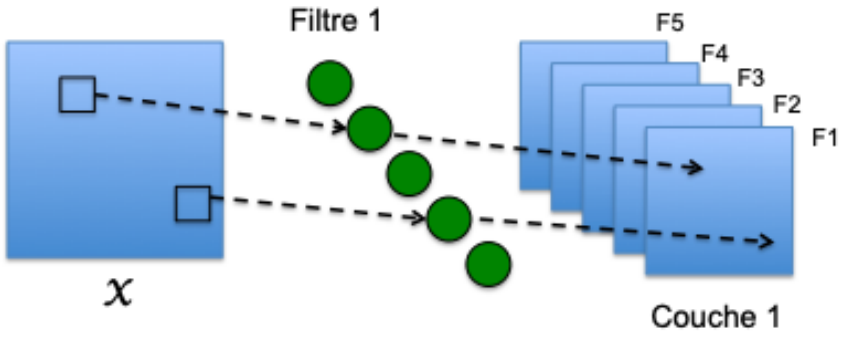

FIGURE 3 – Schematic of a first convolutional stage with 5 different filters. For each filter, e.g., filter F1, each neuron handles a small part of the original image, and all weights associated with each neuron are identical from one neuron to another.

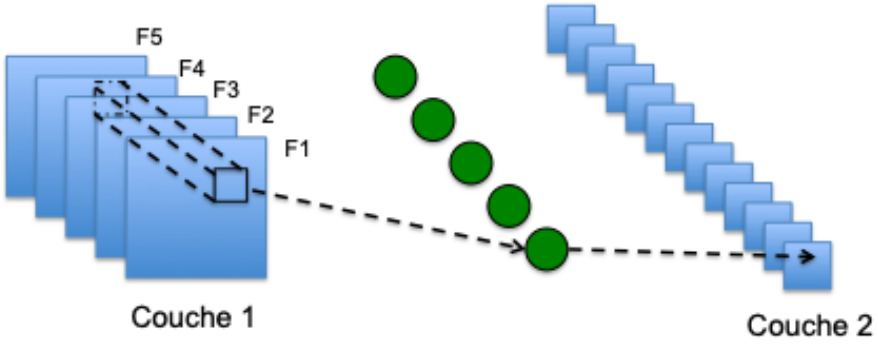

FIGURE 4 – The operation between layer 1 and 2 not only operates in the axes of the original image but also takes care of a new dimension along the results of the previous filtering. This is where the difficulty lies in understanding the mathematical nature of the operators involved.

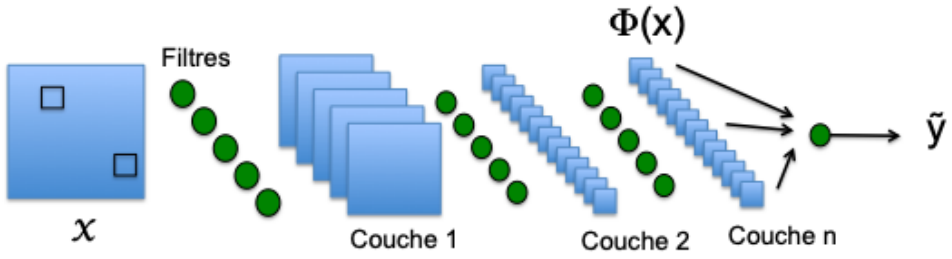

FIGURE 5 – Deep convolutional network consisting of several layers of filters, and at the end of the chain, a single linear layer or multiple fully connected layers that provide the response (classification/regression).

spatial support of input filters (cf. F1) is small compared to the size of the image, but from the second layer onwards, there is a mixing of filters. A deep network can be schematized as shown in Figure 5, which has millions of parameters. So, there are many structures in this architecture that were deduced from prior knowledge. What is fascinating is that this type of architecture is capable of addressing a wide variety of problems: in imaging, sound, language, text, physics, chemistry, etc. (see Introduction of the 2019 Course). This generic aspect will keep us occupied. Note that $\Phi(x)$ has considerably fewer parameters than the original image; that is, the dimensionality reduction is considerable so that the classification boundary at the end has been flattened. Of course, the question is: what are the **principles of this architecture**? It is clear that the notion of **symmetry** is important, it comes from the very first layer; moreover, there is the notion of **multi-scale**, because as we advance through the layers, a neuron "sees" portions of the original image that are increasingly large (this is the counterpart of the natural cognitive system); finally, there is the notion of **sparsity** through the neural response (cf. the ReLU activation function).

A typical example of the ability of these architectures to perform much better than previous solutions is the classification of images from the ImageNet database [13] (see Sec 2.1.1 of the 2019 Course). Before 2012, architectures included all the *a priori* of the problem and performed linear regression, while neural networks performed less well. The game changed with 1) increased computational capabilities, 2) large training datasets. Thus, in 2012, Goeffrey Hinton, Alex Krizhevsky, and Ilya Sutsever developed **AlexNet**,

13. http://www.image-net.org/

which outperformed everyone else. Currently, ImageNet has 1 million images labeled into 2000 classes.

It is very important to repeat once again that it takes many samples to understand why deep convolutional neural networks work so well and compare them to systems that do not learn at all (where the filters are fixed in advance), as we will address in the course. What we would like to understand, of course, is what has been "learned", especially when compared to a system that is fixed from the start. Experimentally, for example, by changing the initial conditions of the parameters, the solution to minimization is likely to be different, yet the networks exhibit the same statistical behaviors (cf. the same generalization capacity [14]). From a mathematical point of view, the architecture has somehow captured a form of regularity in the function that answers the question $y = f(x)$.

Why is it important to understand "Why it works?" Well, there are times when generalization fails, and not just a little! This is about **adversarial examples** [15]. Such an example, as close as possible to an example used for training, exists, and all systems have such pathologies. It is clear that we do not want to put such a system in devices where people's lives are at stake, for example. Therefore, we need tools to guarantee the safety level of a system, and this requires an understanding of the nature of the mathematical operations used.

2.5 Classical Systems

If we set aside neural networks for a moment, what are the more "classical" systems? The first example is that of speech recognition (see Sec. 2.2 of the 2019 Course), which we will delve into in upcoming courses. **Time-frequency analysis** is a valuable tool, and one quickly realizes its significant variability (rhythm, change of pitch, etc.) depending on the speaker, for the spectrogram obtained, even when it's the same word being spoken. What techniques have been used despite these variabilities?

The case of speech is interesting because it has been studied since the 1960s, and over time, it has become highly specialized with a technique that has been refined but

14. We can even say the same problem in the face of adversarial examples.
15. Szegedy C., Zaremba W., Sutskever I., Bruna J., Erhan D., Goodfellow I. J., Fergus R., 2013, CoRR, abs/1312.6199.

has changed very little until the 2000s. The basic idea is to elucidate the time-frequency structures representing sounds, and then we attempt to recognize the phonemes that make up words, and so on. The structures are the states of the system, and there are transition probabilities from one state to another (dictated by knowledge of spelling and grammar). Thus, a Markov chain is defined on which a Gaussian mixture model is placed. This type of technique was completely transformed by neural networks. First, the Gaussian mixture model was changed, and then the complete analysis (including time) was taken over by convolutional networks. Currently, speech recognition is much better handled by these deep networks.

A very interesting problem is source separation (known as the "Cocktail Challenge"), which is fundamental for hearing aids. It is not just a matter of amplifying everything because the signal and the ambient noise are not effectively separated. In 2018, a solution was implemented by Yi Luo and Nima Mesgarani [16]. Separation is almost perfect in a matter of milliseconds, making it compatible with hearing aids. Of course, the network was trained with a large number of samples (not from the people it needs to recognize), but we do not understand how it works beyond the first layer of neurons where a semblance of a spectrogram is reconstructed.

In the field of Physics (Chemistry), there is the "classical" approach of trying to integrate fundamental equations (Newton, Boltzmann, Navier-Stokes, Maxwell, Schrödinger...), but the calculation becomes difficult when dealing with a large number of interacting entities. The alternative approach is to answer specific questions (e.g., what is the energy of the system I am studying?) based on a known database of examples.

Finally, what is the connection between artificial neurons and biological neurons? If we can understand the operation of artificial systems, which are becoming increasingly powerful, we are legitimately entitled to ask what we can learn about the structures/functions of biological systems. There is also a practical interest in going back and forth between biological and artificial systems. Indeed, it is observed that humans can perform object recognition in just 1/10th of a second. Biological neurons are very slow, so recognition relies on very few layers (around 7) and there is no very complex feedback loop. How can we modify current (artificial) architectures to achieve this level of performance?

16. See the article arXiv:1809.07454v2

2.6 Questioning

So, the question is: why are convolutional neural network architectures "generic?" There are three types of problems, in fact:

— **Estimation**: analyzing generalization error (bias/variance, see Course 2018);

— **Optimization**: minimizing empirical error (see Course 2019);

— **Approximation**: the connection with architecture, i.e., with the *a priori* information of the problem, which is the subject of this year's course.

So, we will ask the following questions: what is the *a priori* information, why convolutions, what types of filters, what is the role of non-linearities (activation), and the connection with neurophysiology (context of image and sound). We will address the following points:

— Regularity: separability, symmetry, sparsity

— Symmetry: convolution and Fourier analysis

— Separation and sparsity: uncertainty principle and time-frequency representations

— Multi-scale transforms and wavelets

— Invariance to translations, rotations, deformations

— Classification without learning (SIFT [17] and MFCC [18])

— Multi-scale invariants: wavelet networks and scattering

— Applications in various domains

— Finally, why there are shortcomings as soon as the system becomes complex...

3. Lecture 29 Jan.

3.1 Basic Questions (Recap) and Course Outline

In the previous session, we listed a number of questions:

17. Scale-Invariant Feature Transform
18. Mel Frequency Cepstral Coefficients

— Why are **CNNs capable of generalizing** to very different generic problems (images, sounds, language, chemistry, physics, biology, etc)? If this is the case, it means that the function to be approximated has a particular regularity.

— So, what is the **generic regularity** (if it exists) that would underlie problems such as physics, perception, biology, symbolics (language), and so on? Herbert Simon in 1962 provided an answer in his book "Architecture of the Complexity" [19]. However, even though this book provided an emerging framework, it was not yet possible to grasp problems as we do today. There has been a significant leap from qualitative arguments to their practical implementation.

— If there is a generic regularity, there is concomitantly **generic information** expressed in CNNs. How can we express it mathematically (and in an algorithm)? We know that $x \in \mathbb{R}^d$ with a very large d, but x has structure. In particular, x is indexed by $x(u)$ (where u is an index of a pixel in an image, a time frame of sounds, a word in a text, etc). However, u resides in **a low-dimensional space** (e.g., $u \in \mathbb{R}$ for sound, $u \in \mathbb{R}^2$ for an image, $u \in \mathbb{R}^3$ for a video, etc.), and thus, we have low-dimensional structure that will essentially provide the content of the ***a priori* information** that we can "hard code" into the network's architecture. Indeed, **through u, we can impose problem symmetries**: e.g., translation, which leads to all Fourier analysis, more generally group structures that act on the variable u, neighborhoods that lead to multi-scale representations, and we can also use graphs to account for interaction structures between temporal variables (e.g., stock prices in finance, actors in social networks, etc). So, **it is through low-dimensional structures that current mathematics have an angle of attack on the problem**.

The course outline for this year 2020 is as follows:

1. Architecture of Complexity, Estimation, Optimization, Approximation
2. Approximation: Multi-scale, Group and Symmetry (dimensionality reduction)
3. Time/frequency, uncertainty principle, locality (problem beyond CNNs)
4. Scale: Wavelet Transform (WT), Auditory Perception
5. Dyadic Wavelet Filter Banks (connection with invariants)
6. Visual Perception, (WT and invariants), and nonlinear shapes (Scattering)

19. See Sec. 4.1 of the 2019 Course on "Cybernetics"

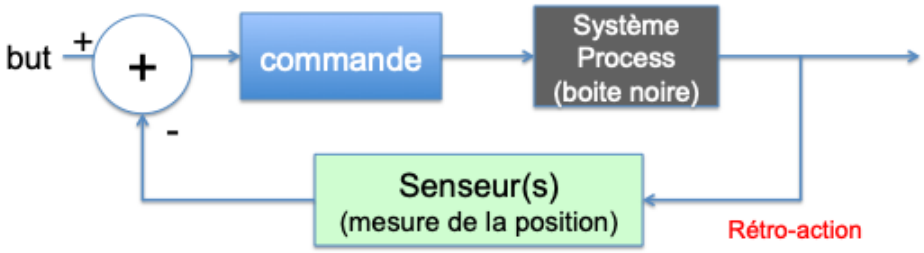

FIGURE 6 – The principle of analyzing the evolution of a complex system through feedback loop.

7. Applications: Image/sound, quantum chemistry. Limitation of introducing *a priori* information that is not sufficient to explain the performance of CNNs.

One remark, we mentioned that by studying CNNs, we would revisit classical harmonic analysis. In this regard, CNNs will provide a different perspective, particularly because they are nonlinear, whereas harmonic analysis is essentially linear. Thus, we need to question the role of these **non-linearities**.

3.2 Architecture of Complexity

In 1962, we are in the field of "Cybernetics", introduced by the mathematician N. Wigner (1947) [20], whose motto is to analyze the evolution of a complex system through a feedback loop (Fig. 6). The main idea is that we refrain from modeling the complex system (the "black box"), and what matters is the desired goal (response). Through this rather simple scheme, we can understand the behavior of very complex systems of interactions among agents, for example, and we can also grasp one of the major feedback systems, which is evolution.

Now, if we try to open the "black box" to understand the internal structure, H. Simon highlights several fundamental ideas:

20. Some areas in which Wigner had a decisive impact: probability with Wigner's measure, revised harmonic analysis, signal processing, and control.

— The structure is **hierarchical** almost all the time.

— A **dynamic explanation** (temporal) of this hierarchical structure is the search for stability (survival).

— **Scale separability** (in the hierarchy) allows us to overcome the curse of dimensionality.

— The temporal description must be seen as **processes** and not a succession of static states. This resonates with the current CNN learning concept where training data is analyzed separately without any connection between them, explaining the need for a large amount of data, whereas humans can learn at a lower cost because images are part of a dynamic context. It is clear that the basis of dynamics (e.g., physics) is differential equations.

3.2.1 Hierarchical Structures

The observation of the existence of hierarchical structures is evident: in Physics, we have many examples of structures and hierarchies, from elementary particles (micro) to materials (macro); similarly in Biology, one can start from molecules (micro) to complex organisms (macro); also in Perception, from pixels to scenes via objects; in Symbolic Systems, from letters to books; and finally in Social Systems. Counterexamples where there are no hierarchical structures typically exhibit notable symmetries.

The question that immediately arises is why these hierarchical structures exist. H. Simon provides an answer related to the **dynamic evolution of the system**. According to him, for there to be evolution, all intermediate states (substructures) must be stable [21]. The number of substructures with which there is direct interaction is quite limited, and strong interactions occur within the structures, while interactions between structures are weaker. Thus, high-frequency dynamics are governed by the interior of the structures, and low-frequency dynamics are governed by inter-structure interactions.

The major difficulty is that the tree-like description that attempts to capture hierarchical structuring (vertical) does not work most of the time because there is horizontal structuring, and there is, in fact, interaction with all structures in the end. Therefore, there is **weak separability**, which complicates the formulation of a tree-based solution.

21. See Sec. 4.1 of the 2019 course, the metaphor of the watchmaker.

Another fundamental point is the notion of **symmetries** related to these structures. For example, in the micro world, the wave function has particular shapes generated by the indistinguishability of the composing electrons; in the macro world, the symmetry of the architecture of certain buildings is evident (e.g., windows are interchangeable without affecting the building); the same applies to people in a company where, for a certain number of tasks, people are interchangeable, etc. It is clear that more symmetries imply less need for elementary structures: think of LEGO bricks, which allow for a multitude of complex structures, in biology, there are only about twenty amino acids, and the number of atoms in the periodic table is around 94 if we restrict it to those encountered in nature. So, there is a form of **sparsity** underlying this.

However, even if the number of elementary structures is relatively small, when you stack structures, it becomes more complicated: interchangeability between humans has limits, after all.

3.2.2 Temporal Description

In a simplified manner, the temporal description can be approached through two methods:

— The first is through the **state** $x(t)$, where time serves as an index, and from one moment to another, the state changes (e.g., a slideshow of family photos). The problem is that if you want a detailed description, you need to have a very large number of states.

— The other approach is through the **evolution of the process**, which involves differential equations of the form:

$$\frac{\partial x(t)}{\partial t} = H(t)x(t)$$

with the Hamiltonian $H(t)$, the evolution operator. So, in principle, one should focus on studying this evolution operator. However, this is not what is done in neural networks (such as CNNs). That being said, in this year's course, we will study problems without time, i.e., problems of the type $x(u)$, although this approach is certainly very suboptimal. For example, when observing the brain's visual perception system, it is noted that feedback

loops have much larger time constants than feed-forward processes. For example, image (face) recognition occurs in 1/10 second without feedback loops, i.e., as if it were a static analysis, which led to developments that omit time. This means that to recognize an image, there is no need to process a video. However, to recognize a very large number of images without video, you need to have an enormous number of images for training. So, in summary, it is clear that for studying certain types of problems, there may be no need to consider time, but this is paid for with training inefficiency.

3.3 CNN Networks

In a way, neural networks are systems that attempt to address H. Simon's program: the learning of a neural network is a closed-loop system, where the cost function plays the role of measuring the error that influences the adjustment of parameters in reaction. The big advantage is that we have an algorithm that we can analyze.

So, the input [22] $x(u)$ will be transformed by the first series of filters into $x_1(u,k)$ where k is a channel index, with the operation being a convolution, followed by a non-linearity ρ of the ReLU (Rectified Linear Unit) type [23]. Therefore, the relationship from x to x_1 is given by:

$$x_1(u,k) = \rho(x * h_k(u)) \tag{4}$$

Then, most of the time, a downsampling operation follows (e.g., *max-pooling* or *average-pooling*) .

The next layer is more complicated because a filter is applied not on a single patch of an image but on a data cube. Then, convolution/pooling layers are repeated to achieve the representation $\Phi(x)$, which depends on all the operators defining the used filters, i.e., the system's parameters are $\theta = \{L_i\}_{i \leq p}$ for a system with p layers [24]. So, with a neural network with p layers, one can approximate a class of functions, defined by the set $\mathcal{H}$ as

22. Note that if we consider an image, it can itself be made up of several channels, e.g., RGB when taking a color photo, or even *ugriz* when considering astronomical filters.

23. $\rho(x) = \max(0, x)$

24. Note that in the case of a classic ReLU, there are no associated parameters, which is not the case, for example, with PReLU, which assigns a slightly non-zero value a for negative input, and this value is optimized along with the filter parameters.

follows:

$$\mathcal{H} = \{f_\theta \ / \ \theta = \{L_i\}_{i \leq p}\} \tag{5}$$

To determine the parameters θ suitable for the posed problem, we will minimize the error between the "true" f and f_θ. It is immediately apparent that if the number of parameters is in the millions or even billions, $\mathcal{H}$ is of colossal dimension. Model complexity is given by $\log |\mathcal{H}|$.

Now, in order to understand how a network generalizes, three areas must be addressed: **estimation** (statistics), which will allow us to obtain θ from a training set $\{x_i, y_i = f(x_i)\}_{i \leq n}$; **optimization**, which involves minimizing the error, for example, through stochastic gradient descent; finally, there is the problem of **approximation**, which involves looking at the minimum error by choosing the best θ then knowing if the error decreases as the size of θ (cf. the cardinality of $\mathcal{H}$) increases. Before delving into the details, it should be noted that these three areas are intertwined, while the Machine Learning community is somewhat divided between statisticians who focus on estimation, optimizers who work to make the numerical problem as efficient/robust as possible, and there are specialists in approximation theory who answer the question of the size of $\mathcal{H}$ to approximate f with an error ϵ. These three disjointed communities have been compelled, so to speak, to evolve to tackle the problem of networks, especially estimation and approximation have become very closely related problems.

3.4 Estimation: Bias-Variance

This is a topic that was addressed during the 2018 course, and here are the key ideas. The challenge is to approximate f (unknown) using f_θ. So, we have a risk (error) R defined as the expectation of the error between the prediction $f_\theta(x)$ and the true value y, according to

$$R(f_\theta) = \underset{(x,y)\sim\Omega}{E}[r(f_\theta(x), y)] \tag{6}$$

where Ω is the joint distribution space that relates x to y. We want to minimize $R(f_\theta)$ and therefore find:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \ R(f_\theta) \tag{7}$$

Depending on the problem (regression/classification), the nature of the function r changes. For example, a quadratic risk is often used for $y \in \mathbb{R}$. In the case of classification with K classes, the network outputs an array z with K values indexed by y. The idea is that the response should be greater as it points to the correct class. To do this, we provide an estimate of the joint probability of y and x indexed by θ, i.e., $p_\theta(x, y)$. In doing so, we are in a probabilistic estimation framework (while we were in an approximation problem), which leads us to the concept of **maximum likelihood** where θ maximizes $p_\theta(x_i, y_i)$ **on the training examples**, which is equivalent to minimizing $-\log p_\theta(x_i, y_i)$.

However, the network's output is not a probability distribution (the sum is not equal to 1), so to do this, we use the `softmax` function (sums over y run over the K components of the output vector, and in this case, the "target" is encoded in a hot-vector whose only non-zero component is that of the correct class):

$$f_\theta(x) = z_y(x) \xrightarrow{\text{softmax}} \frac{e^{z_y(x)}}{\sum_{y'} e^{z_{y'}(x)}} = p_\theta(x, y) \tag{8}$$

So, by summing over all samples, the function to be minimized is written as (note that θ is included in the calculation of $z_y(x)$):

$$L(\theta) = \sum_i -\log\left(\frac{e^{z_{y_i}(x_i)}}{\sum_{y'} e^{z_{y'}(x_i)}}\right) = -\sum_i \left[z_{y_i}(x_i) - \log\left(\sum_{y'} e^{z_{y'}(x_i)}\right)\right] \tag{9}$$

If $z_{y_i}(x_i)$ achieves the maximum, then the function is minimized, and the advantage is that $L(\theta)$ is differentiable, which makes it possible to use a gradient descent algorithm. Thus, the problem of approximation/estimation takes into account the use of an optimization algorithm.

Therefore, if we go back to our initial problem of estimating $R(f_\theta)$, ultimately, we are only able to handle an empirical risk $\tilde{R}(f_\theta)$ obtained from the examples in the training set, i.e.,

$$\tilde{R}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} r(f_\theta(x_i), y_i) \tag{10}$$

So, we estimate the expectation using an empirical average. Let $\tilde{\theta}$ be the value of the

parameters that minimizes the empirical risk

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmin}}\ \tilde{R}(f_\theta) \tag{11}$$

We should be able to control the difference (the error) between θ^* and $\tilde{\theta}$, or the corresponding risks.

Let's look at how the (true) minimum risk $R(f_{\theta^*})$ and the generalization risk given by $R(f_{\tilde{\theta}})$ compare, for example, by giving the network test batches.

Proposition [25]

$$\boxed{R(f_{\theta^*}) \leq R(f_{\tilde{\theta}}) \leq R(f_{\theta^*}) + 2 \max_\theta |R(f_\theta) - \tilde{R}(f_\theta)|} \tag{12}$$

The left inequality is obvious by the definition of $R(f_{\theta^*})$. For the right inequality,

$$\begin{aligned} R(f_{\tilde{\theta}}) - R(f_{\theta^*}) &= R(f_{\tilde{\theta}}) - \tilde{R}(f_{\tilde{\theta}}) + \tilde{R}(f_{\tilde{\theta}}) - \tilde{R}(f_{\theta^*}) + \tilde{R}(f_{\theta^*}) - R(f_{\theta^*}) && (13) \\ &\leq 2 \max_\theta |R(f_\theta) - \tilde{R}(f_\theta)| + \tilde{R}(f_{\tilde{\theta}}) - \tilde{R}(f_{\theta^*}) && (14) \end{aligned}$$

where $\tilde{R}(f_{\tilde{\theta}})$ is minimized when considering the empirical risk $\tilde{R}$, so $\tilde{R}(f_{\tilde{\theta}}) - \tilde{R}(f_{\theta^*}) \leq 0$, and thus

$$R(f_{\tilde{\theta}}) - R(f_{\theta^*}) \leq 2 \max_\theta |R(f_\theta) - \tilde{R}(f_\theta)| \tag{15}$$

So, when we replace the optimal parameter θ^* with the learned parameter $\tilde{\theta}$, the risk is higher (first inequality), but the error is governed by **the error we make when approximating the true risk with the empirical risk** (second inequality).

This error depends on the quality of the function approximation. This puts us in the context of the problem of statistics of the concentration of an estimator around its mean. The error in the estimation will depend on two terms: an **irreducible bias** term given by $R(f_{\theta^*})$, which is a minimum, and a **variance** term due to the fluctuation of the estimation over θ.

In machine learning, we make a crucial assumption that is not always valid, namely that the training observations are **independent**. This is fundamental because for an average to converge to an expectation, fluctuations need to cancel each other out, which is

25. Also see Section 2.3 of the 2018 course.

much more favorable if the samples are iid. This is one of the problems: **we must ensure that the samples are not biased and that they follow the "general" distribution of the problem**. Given these assumptions and the condition that the function r is sufficiently regular, we can prove (see PAC Theorem of the 2018 course):

Theorem 1
With probability $P \geq 1 - \delta$, we have

$$\max_{\theta} |R(f_\theta) - \tilde{R}(f_\theta)| \leq \sqrt{\frac{\log |\mathcal{H}| + \log(2/\delta)}{n}} \tag{16}$$

The term with $1/\sqrt{n}$ is classic when you have n independent samples, and the numerator consists of a term related to the size of the hypothesis class and another to the confidence level you want to achieve. On one hand, the smaller δ is, the more samples are needed (n large), and on the other hand, with n and δ fixed, you must choose the hypothesis class carefully so that $\log |\mathcal{H}|$ (cf. the number of parameters) is not too large. This result is the classical view of error estimation.

The practical consequence is that there is a trade-off between bias and variance when looking at the evolution of error as a function of the size of $\mathcal{H}$ (see Fig. 7). So, you need neither a model that is too large (risk of overfitting) nor a model that is too small (underfitting).

How do we limit the size of $\mathcal{H}$, which amounts to limiting the exploration space of θ values? This can be done using **regularization**. One type of regularization will penalize the loss, for example, with a norm of type $\|\theta\|^2$ ([26]). Another type involves early stopping.

That said, the need to limit the number of parameters is not what those who practice neural networks have observed. On the contrary, it tends to show the opposite, that it works better as larger and larger networks are used. Recently, it has been observed that, contrary to the overall curve of the two bias/variance terms, there is a curve of the type shown in Figure 8. That is, when the number of parameters exceeds a critical size, the

26. See Sec. 4.2.3 of the 2019 course, and also see Sec. 7.2.4.2, which provides the Bayesian and frequentist viewpoints on the subject. Note also that the *weight decay* technique introduces L2 regularization.

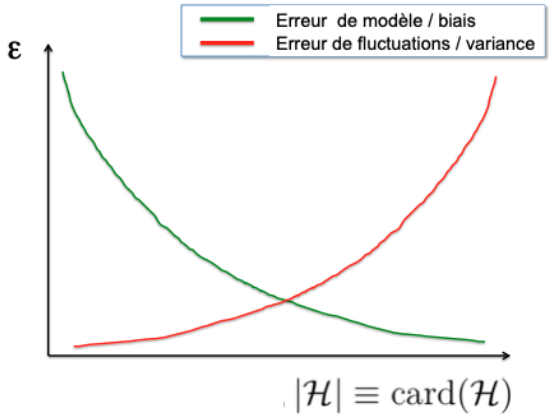

FIGURE 7 – Bias-Variance Trade-off as a function of the size of the set $\mathcal{H}$.

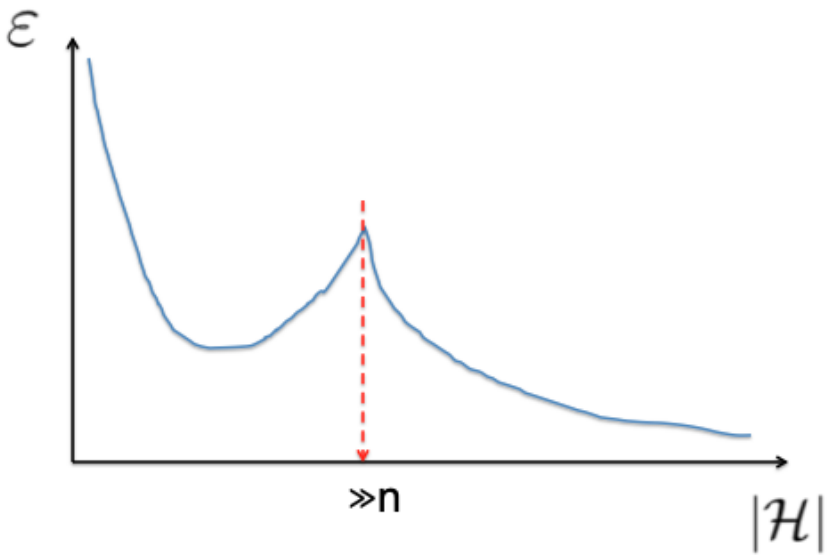

FIGURE 8 – The classical upper bound is surpassed in very deep neural networks; beyond a critical size, the error decreases instead of increasing.

error (in this case, the generalization error) starts to decrease again. This means that we have been able to beat the "classic" upper bound of the PAC theorem.

How can we explain this phenomenon? We start to have redundancy in θ, i.e., in the parameter space, it seems that we are increasing the sampling density. **The transition occurs when the number of parameters reaches a critical value much larger than the number of examples.** However, these "double-slope" curves derived from recent results do not seem to explain current results on ImageNet because we would be on the left side of the curve. What is important is that as soon as the parameters are not independent, collective phenomena are observed.

3.5 Optimization

How do we obtain $\tilde{\theta}$? We perform stochastic gradient descent by batch [27] and use the backpropagation algorithm. But, the problem is highly non-convex, so it is easy to get stuck in local minima. However, despite the different minimization conditions that lead to different solutions, the generalization properties are the same. Faced with this observation, mathematics will attempt to describe the landscape of the cost function during minimization and demonstrate that local minima "far" from the absolute minimum are rare and extremely narrow, so their basins of attraction are unlikely, and therefore, they are not problematic [28]. This type of problem is identical to what is encountered in statistical physics when trying to minimize an energy function. In addition to this problem, in machine learning, we add forms of regularization: early stopping, L2 penalization, or dropout.

That being said, what will interest us later is the problem of **approximation**, which corresponds to the model bias curve (Fig. 7), i.e., how to configure the network so that the obtained error decreases rapidly as a function of the number of parameters.

27. See Section 9 of the 2019 Course
28. What about saddle points?

4. Lecture 5 Feb.

4.1 Estimation/Optimization

We begin with a recap of some concepts to set the stage. Our goal is to understand the class $\mathcal{H}$ of functions f_θ that a network can approximate. Recall that θ represents the network's parameters (i.e., filters), and our objective is to minimize the true risk on average

$$R(f_\theta) = \underset{(x,y)\sim\Omega}{E}[r(f_\theta(x), y)] \tag{17}$$

Ideally, we would like to find the algorithm that discovers θ^* minimizing this "true" risk. However, what is currently accessible to us is more of an empirical risk calculated with a training set:

$$\tilde{R}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} r(f_\theta(x_i), y_i) \tag{18}$$

whose minimization can provide us with $\tilde{\theta}$. Consequently, the risk of generalization $R(f_{\tilde{\theta}})$ is bounded by Equation 12. **The approximation error, also known as the "model bias",** is given by the curve (green) in Figure 7 ([29]). It **decreases as the network's complexity increases** (i.e., the cardinality of $\mathcal{H}$). The other contribution due to fluctuations in the function f_θ within the class $\mathcal{H}$ is represented by the increasing curve (red) as a function of $|\mathcal{H}|$, bounded by the expression (Eq. 16) from the PAC theorem. The "ideal" size of the class $\mathcal{H}$ is typically the one where the two contributions are equal.

Therefore, we immediately deduce that it is not advantageous to have an overly large class, as there would be too many parameters (i.e., the $\log|\mathcal{H}|$ increases), necessitating a larger number of samples (i.e., n). However, this is not what we observe when working with neural networks: from a critical threshold onwards, larger networks lead to better generalization performance. **This observation contradicts the previous prediction.** Moreover, it contradicts the intuition one might form from an interpolation problem. The question is: Why?

Mikhail Belkin and collaborators provide an explanation for this phenomenon in a

29. NDJE: In some versions of my notes on the 2018 course, there is an inversion in the legend color that can be confusing.

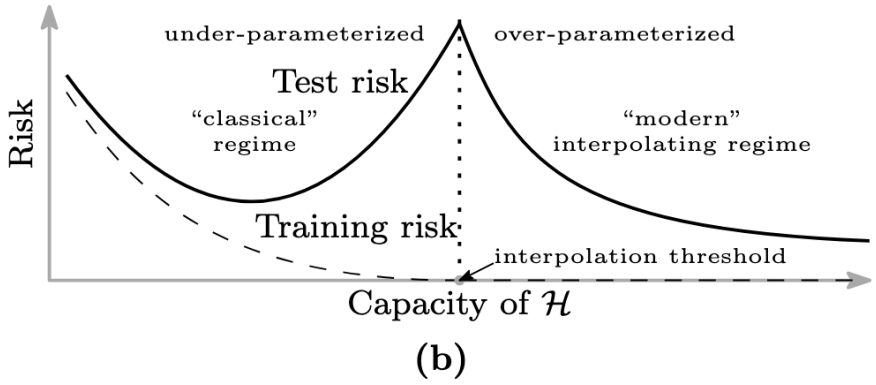

FIGURE 9 – Excerpt from Figure 1 of the article arXiv:1812.11118v2

recent article [30]. The observation is as follows (Figure 9): next to the classical error curve given by $R(f_{\theta^*})$, we should consider **the curve of empirical risk** (training error) given by $\tilde{R}(f_{\tilde{\theta}})$, which reaches 0 for large networks at a point called the **interpolation threshold**. In other words, for 1D interpolation, we have found a function that "passes through" all the data points exactly. However, we would expect the network not to generalize well for examples that are not in the training set, but what is observed is that beyond this point, there is a new decrease in the total error for even larger networks. The curve of the total error is referred to as the **"double descent risk curve"**. In practice, the message is clear: significantly increasing the network size helps to beat the minimum of the "classic" curve that occurs before the interpolation threshold.

Can we understand this second descent of the risk? **We are clearly beyond the realm of interpolation; the number of parameters far exceeds the number of degrees of freedom.** There is a form of redundancy in the parameters, and the slope of the second descent depends on the optimization algorithm. Stochastic gradient descent (SGD) inherently has regularization mechanisms due to the random noise on the gradients. Among all solutions (over θ), the algorithm converges toward the smoothest ones. Therefore, the behavior of this second descent (a current research topic) is related to a regularization phenomenon, and the key question is whether it is beneficial to hyperparameterize and let the minimization algorithm take control or if it is preferable to explicitly and directly

30. Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal, "Reconciling modern machine learning practice and the bias-variance trade-off", arXiv:1812.11118v2

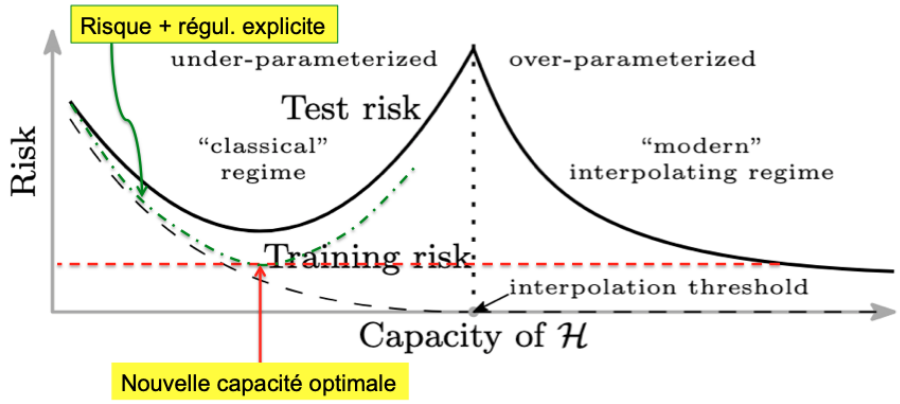

FIGURE 10 – By not going into the hyperparameterization region and instead regularizing the risk explicitly, is it possible to obtain the green dashed line to reach the same minimum but with far fewer parameters, making training easier?

regularize the risk (see Figure 10)? And the subsequent question is: what type of regularization should be used? These are very current questions, and the answers depend on specific cases. We understand that the estimation problem is intricately linked to the optimization problem; both must be considered simultaneously. The following question arises: if we have optimized well, what about the minimum of the error? In other words, what is the asymptotic convergence rate? The faster the decrease of $R(f_{\theta^*})$, the less significant the fluctuation error will be. The idea is to be able to address the problem while using as few parameters as possible.

4.2 The Approximation Problem

In 2018, we studied the **curse of dimensionality problem**, and here we provide a brief recap of the main ideas. The central point arises from the fact that x (sample/data) belongs to $\mathbb{R}^d$ with d very large. Therefore, a form of regularity on the function we use must be imposed to limit the class $\mathcal{H}$. What kind of regularity are we talking about? For example, continuity (weak regularity), differentiability (strong regularity), etc. We have also seen a regularity known as locally Lipschitz (intermediate regularity) and uniformly

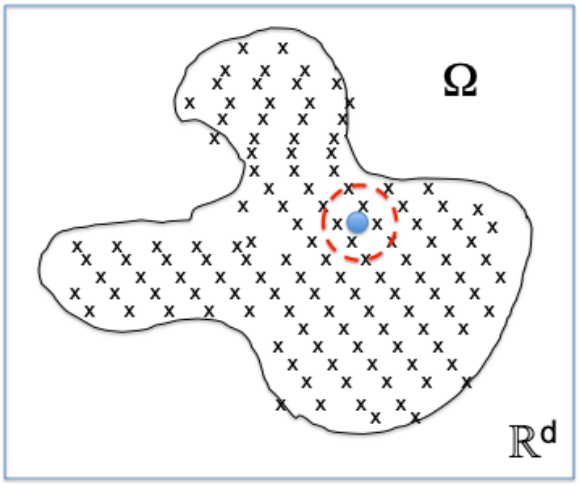

FIGURE 11 – Tiling of the space Ω using balls with radius ε.

Lipschitz on a space that can be considered as a form of differentiability [31]:

— **Definition**: f is **uniformly Lipschitz** over $\Omega \subset \mathbb{R}^d$ if

$$\exists\ C/\ \forall (x, x') \in \Omega \quad |f(x) - f(x')| \leq C||x - x'||$$

What can we deduce about the ability to approximate with this type of regularity? Given samples x_i for which we know $y_i = f(x_i)$, we want to provide an approximation of $f(x)$. Therefore, we need to study the distance $||x - x_i||_{i \leq n}$ and we would like the minimum of this distance to be less than ε. However, how many balls with radius ε are needed to tile the entire space Ω (Figure 11)?

Prop. [32]: If $\Omega = [0, 1]^d$, the radius ε of n balls covering Ω satisfies the relation

$$\varepsilon \approx \sqrt{d}\ n^{-1/d} \tag{19}$$

so, to satisfy the criterion, we need to have a number of samples with a scaling law

$$n \geq C\varepsilon^{-d} d^{d/2} \tag{20}$$

31. NDJE: We could also introduce the concept of Hölder functions.
32. NDJE: see the 2018 course sec. 3.4; there is a constant in the bound on ε given by $1/\sqrt{2\pi e}$).

With $d \sim 30$, it is evident that this number is enormous, even in relatively "low" dimensions. We also conclude that

$$\boxed{\|f - f_\theta\|_\infty = \sup_{x\in\Omega} |f(x) - f_\theta(x)| \leq C\sqrt{d}\, n^{-1/d}} \tag{21}$$

meaning that there is indeed a decrease as a function of n, but it is exponentially slow. From here, two reflections can be developed:

— What about Ω? Its dimensionality may be much smaller than d. Certainly, there are simple problems with small degrees of freedom in robotic systems. However, as soon as we deal with images, sounds, social phenomena, even if the dimensionality of Ω may be smaller than d, it is not that small for the curse of dimensionality problem to be far away.

— Let's assume that the dimension of Ω is large; then, **stronger regularity needs to be imposed** than that of Lipschitz functions. It must be understood that we cannot adopt local constraints because the probability of finding "enough" samples to achieve interpolation at the point of interest is nearly zero. Therefore, we must turn to **strong global regularities**. The entire problem lies in finding/defining these uniform regularities. This is where H. Simon's text provides avenues for investigation by analyzing the types of hierarchies and interactions among "actors" in complex systems.

4.3 Global Regularities: Separability, Symmetry

First, let's take a step back to discuss the notions of **separability, symmetry, and sparsity**, which we will analyze in detail with concrete cases later on.

4.3.1 Variable Separability

This is the strongest notion in a sense: we hope that the problem is **separable into low dimensions**. This assumption arises from the following idea: $f(x)$ is a function theoretically of d variables, but suppose that the problem allows the following representation:

$$f(x) = f_1(P_{V_1}x) + f_2(P_{V_2}x) + \cdots + f_K(P_{V_K}x) \tag{22}$$

where the P_{V_k} are orthogonal projections onto sets such that $\dim(V_k) \leq q$. $P_{V_k}x$ has q variables (at most), which are linear combinations of the original d variables. A particular case is:

$$f(x_1, \ldots, x_d) = \sum_{k=1}^{K} f_k(x_{i_1^k}, \ldots, x_{i_q^k}) \tag{23}$$

meaning that the functions f_k are functions of q original variables, or in other words, the projections are simply selections of q original variables. More generally, we have linear combinations of the original variables, and what's important is that the f_k are functions of q variables.

How does this help solve the dimension problem? In fact, we move from one problem of dimension d to K problems of dimension q. If we impose a bit of regularity (like Lipschitz type) on each f_k, then

$$\|f_k - f_{\theta_k}\|_\infty \leq C_k \sqrt{q} n^{-1/q} \tag{24}$$

and thus, by summing over k, we finally obtain

$$\|f - f_\theta\|_\infty \leq CK\sqrt{q} n^{-1/q} \tag{25}$$

($C = \max C_k$). In other words, we have increased the convergence rate from $n^{-1/d}$ to $n^{-1/q}$. **So, if the initial problem is separable into small problems of low dimension, it becomes manageable**.

This method of separating a problem into subproblems has a fundamental application in probability density estimation, where $f(x) = \log p(x)$. The separability assumption amounts to assuming that

$$p(x_1, \ldots, x_d) = \Pi_{k=1}^{K}\ p_k(x_{i_1^k}, \ldots, x_{i_q^k}) \tag{26}$$

meaning that we are dealing with Markov models (or graphical models). In this context, we can separate groups of q variables that include, for example, x_1, and those that do not. For those that include x_1, they describe processes interacting with x_1; the others describe processes independent of x_1. So, to understand the variation with respect to x_1, we focus on the first processes. It's a "local" approach.

In the case of images, the problem's separability can be translated into dividing the original image into small patches that are considered independent of each other, meaning

that only the pixels within each patch interact with each other. Thus, we can approach image classification (e.g., dog vs. cat) as the sum of likelihoods over the small images because we believe we have enough feature recognition power on the patches. If we can do that, the problem of dimension $10^3 \times 10^3$ becomes a problem of dimension 8×8. This approach was widely used before the 2010s when local invariant descriptors were used to reduce the dimensionality of the problem beyond 64 using the SIFT (scale-invariant feature transform) method from 1999, followed by classification methods. It was the state of the art at that time.

For tasks like music recognition and phonemes, we can divide time into small intervals, typically 25 ms (up until the 2010s). We have, for example, 200 samples, if we sample at 8 kHz, and we calculate the final decision as a sum of individual decisions. More subtle things can be done (Markov processes), but for music type classification, we used the sum of local evidence. Local descriptors were Mel-Frequency Cepstral Coefficients (MFCC), designed to construct invariants that are stable under deformations (see Sec. 7.4). In parallel, there were neural networks, but clearly, in the years before 2010, they did not perform well for these problems.

We can also ask about this separability technique in quantum chemistry. The x variables are positions and charges, and the function to find is quantum energy. Covalent bonds are those that connect atoms by sharing electrons. It is reasonable to think that energy is dominated by terms that describe the local neighborhood relationship between atoms. We know that this is not entirely accurate because there are long-distance terms, but we can incorporate them. However, the big challenge is the quantum terms, but we can use descriptors that provide information about the local structure (similar to linguistic bags of words). This can work but not always.

4.3.2 Scale Separability

In fact, H. Simon's article provides a clue because there are global interactions that cannot be neglected. Let's imagine interactions on a social network; one might think that typically, with about ten people we interact with, our sphere of influence is confined to those ten individuals, and events happening on the other side of the planet shouldn't matter. However, it is indeed a fact that geopolitics on the other side of the world does have an impact on our own decisions. Therefore, long-distance interactions between groups

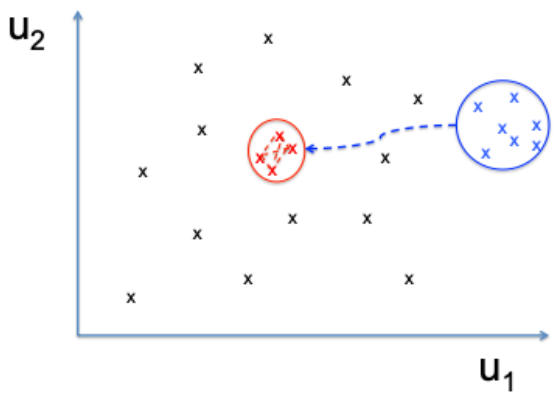

FIGURE 12 – Illustration of short-range interactions within the red group and long-range interactions between the red group and the blue group.

of individuals must be taken into account. We can apply this reasoning to the classification of dogs and cats as well, because implementing it solely based on 8x8 patches is probably not as straightforward. Likewise, to recognize the composer of a musical piece, we cannot rely solely on 25 ms analyses; we need to establish connections between different samples.

Let's consider a hierarchical structure: why does this provide a solution? In essence, we will simultaneously focus on studying interactions between all agents/entities on small scales and consider only large-scale interactions between groups taken as interacting global forms. An illustration of this concept is given in Figure 12. This type of scheme can be envisioned in chemistry/physics and even in sociology.

If we can treat the initial problem in this way, we don't have to deal with d variables but typically $\log d$ groups. Transitioning from d to $\log d$ solves the curse of dimensionality problem. **But there is an assumption that won't work: it's the form of the decomposition (Eq. 22) into a sum of independent problems.** However, we can conceive that for certain problems, we need to consider all interactions between all groups to understand what happens for a given group (see Figure 13). This is a major challenge because even if the groups have weak interactions between them, 1) the sum of their interactions is not necessarily negligible compared to short-distance interactions, and 2) according to H. Simon, they dominate the slow dynamics of the global system.

That said, we will implement **scale separation** with a mathematical tool called

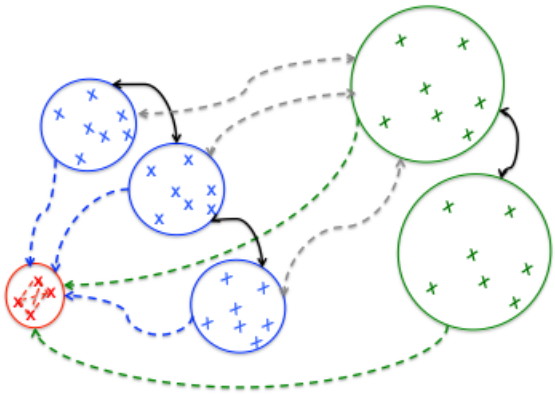

FIGURE 13 – Illustration of the role of interactions between groups at different scales.

wavelet analysis, and we will consider how to capture scale interactions.

4.3.3 General Notions about Groups

This concept implies that there are equivalent states that we know in advance. To understand the regularity of $f(x)$, we can study its regularity concerning transformations [33]. Initially, we can use *local* transformations to determine if the function is continuous, differentiable, etc. However, what interests us here are classes of global transformations, which are symmetry groups of f:

$$G = \{g \ / \ \forall x \in \Omega, \ f(g.x) = f(x)\} \tag{27}$$

The functions g preserve level sets. Indeed, consider the set Ω_t such that

$$\Omega_t = \{x \ / \ f(x) = t\} \tag{28}$$

This is indeed a level set, and Ω_t is invariant under g.

Definition of a Group: Let there be an operation from $G \times G \to G$, such that $(g_1, g_2) \to g_1.g_2$. It defines the group structure (morphism) if it has the following properties:

33. NDJE: see 2019 Course Sec. 3.5

— Associativity: $\forall g_1, g_2, g_3 \in G$, we have $(g_1, g_2).g_3 = g_1.(g_2.g_3)$

— Identity element: $\exists Id \in G$, such that $g.Id = Id.g = g$

— Inverse: $\forall g \in G$, $\exists g^{-1} \in G$ such that $g.g^{-1} = g^{-1}.g = Id$

— Commutativity (optional; Abelian group): $\forall g_1, g_2 \in G$, $g_1.g_2 = g_2.g_1$

Galois's brilliant idea was to identify that, to study functions, we could do so through their symmetry groups. For example, in the case of finding solutions to a polynomial equation, $P(x) = 0$, which corresponds to the level set 0, we can examine the transformations that turn one solution into another. Why do the solutions of a level curve form a group structure? So let $x \in \Omega_t$ and $g \in G$, then $g(x) \in \Omega_t$. It can be easily verified that associativity holds. It is also clear that the identity element is part of G, and it commutes with all elements $g \in G$. Furthermore, if we restrict ourselves to invertible operators, so if g transforms one solution x_0 into another solution x_1, its inverse operates the transformation from x_1 to x_0, and we indeed have a group structure. This type of approach is central in the study of (partial) differential equations: if we have one solution, what are the operations that preserve the solution?

For example, translation on a discrete grid $u \in \mathbb{Z}^2$, $g.x(u) = x(u - g)$. Once we have a group, the question arises about its generators, and the dimension of the group is the number of its generators: $\{g_k\}_{k \leq P}$. In the case of a commutative group, an element of the group can be decomposed as $g = g_1^{n_1}.g_2^{n_2} \dots g_k^{n_k} \dots g_P^{n_P}$. But more generally (without commutativity), the action of the generators must occur sequentially while respecting an order.

Before delving into continuous groups, let's ask what this notion of a group is for. It achieves **dimensionality reduction**. If we have the *a priori* information that we know a subgroup H of the symmetry group G, what can we conclude? By the way, if we know the entire symmetry group of the function f, we *de facto* know all the level sets of the function, so we know all the solutions to equations of the form $f(x) = t$, so we know the topology of the function f; in short, we know it completely. In contrast, if we only know $H \subset G$, which is the practical case: we cannot know all of G, but in image classification problems, it is clear that translation, rotation, or flipping of an object does not change its label. Thus, for $g \in H$, x and $g.x$ belong to the same equivalence class, and **we will use the quotient of Ω by H, denoted Ω/H**. If we take $x_0 \in \Omega/H$, it defines an **equivalence**

class H_{x_0} such that

$$H_{x_0} = \{x \in \Omega \ / \ g \in H \text{ s.t. } g.x = x_0\} \tag{29}$$

And we cannot distinguish, in a sense, $f(x \in H_{x_0})$ from $f(x_0)$. If x_0 is an image, $f(x_0)$ is its label (e.g., dog/cat), then if we translate it, we get $x = g(x_0) \in H_{x_0}$, and $f(x) = f(x_0)$, the label remains the same. **Since the label is the same for all elements of H_{x_0}, we can reduce the number of variables to account for the variability within the equivalence class**. This is indeed dimensionality reduction.

How many variables will be reduced by this quotient operation: typically, in a continuous setting, we expect $\dim(\Omega/H) = \dim(\Omega) - \dim(H)$. This means that for it to be useful, **the dimension of H must be large**. For example, if we take fixed-step translation on an image, its dimension is 2, but compared to the 10^6 variables of a 1024×1024 image, the dimension reduction is entirely negligible. So, **invariance by translation alone is not sufficiently informative**.

4.3.4 Diffeomorphism, Group of Deformations

Let's introduce the ideas of Lie groups, with continuous translation as the guiding thread. We are dealing with translation g in $\mathbb{R}^2 = G$. G is a differentiable manifold of dimension P. The group properties are the same as those of "classic" groups, but now there is a new concept, which is **transport** (Fig. 14). An element x_0 by successive action of the generators (e.g., those of the symmetry group) is transported to x_1, x_2, etc., and each time $f(x_i) = f(x_0)$ remains unchanged. Moreover, it can be transported over possibly long distances. The key difference is that the transport is continuous (or even differentiable), and by considering all possible ways to transport x_0 with the generators (of the symmetry group), we obtain a **differentiable surface with the same label, which we call the *orbit* of** x, defined by $O_x = \{g.x\}_{g \in G}$.

If we want to characterize the differentiable surface O_x, we study **tangent hyperplanes**, which give the dimension of the manifold and are defined by infinitesimal generators. The Lie algebra is the algebra that takes us from one transformation to another infinitesimally close one. In the case of a 2-dimensional image $u = (u_1, u_2)$, then for a

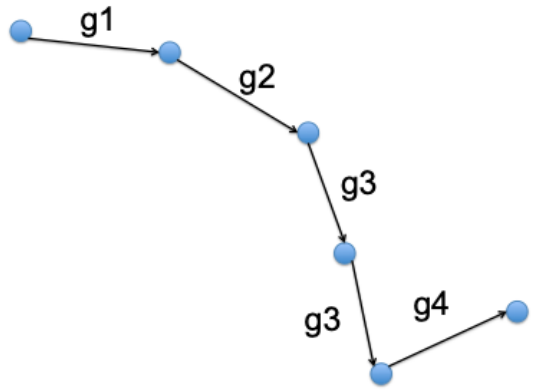

FIGURE 14 – Illustration of the concept of transport.

small transformation g applied to $x(u)$, we have:

$$g(x(u)) = x(u-g) = x(u) - \nabla x(u).g + \cdots = x(u) - \left\{ \frac{\partial x}{\partial u_1} g_1 + \frac{\partial x}{\partial u_2} g_2 \right\} + \dots \qquad (30)$$

So, locally, we have a group of dimension 2, and the generators of the Lie algebra are the partial derivatives $(\partial x/\partial u_1, \partial x/\partial u_2)$, and the translation direction in the tangent plane is given by the coordinates (g_1, g_2).

Thus, in continuous space, the Lie group is a manifold whose tangent planes are generated by generators of fixed dimension. **The peculiarity of manifolds generated by a Lie group is that the tangent planes are all identical, as they are generated by the same generators**. Thus, Lie groups are in a sense extremely rigid structures. A few examples of Lie groups (finite dimension):

— Translation: $g.x(u) = x(u-g)$ with $g \in \mathbb{R}^2$ (dim: 2)

— Rotation: $g.x(u) = x(r_g.u)$ with $g \in [0, 2\pi]$ (dim: 1)

The group of deformations (Figure 15) is interesting because if it acts on the number 3, for example, we might expect that the result of classification should not change after small deformations. This *a priori* information is thus interesting to be able to translate into the class of functions f_θ that classify digits. However, a deformation acting on each pixel is potentially governed by many parameters, which implies **a very high dimensionality of the group of deformations**; this is what we are seeking. This will deeply structure

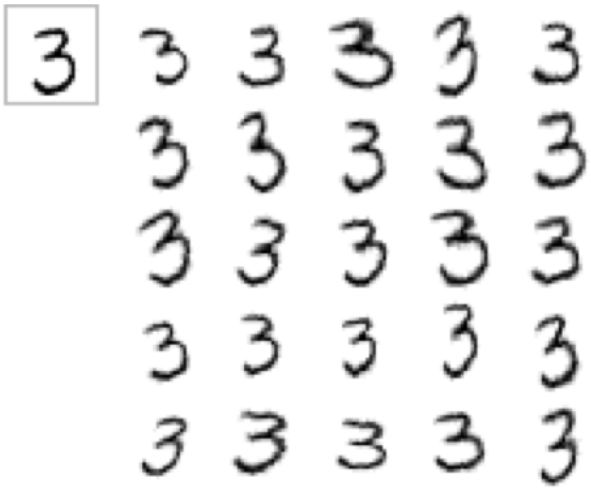

FIGURE 15 – Deformation of an image of a 3.

the classification problem but also other types of problems in image processing, sound processing, physics/chemistry...

The group is that of diffeomorphisms, and an element g of the group acts according to the following relation:

$$g.(x(u)) = x(g(u)) \qquad u \in [0,1]^2 \tag{31}$$

In fact, g acts on the underlying variables of x, namely u, which is a low-dimensional quantity. We will impose that g is a continuously differentiable function C^1: $g : [0,1]^2 \to [0,1]^2$. As at each point (pixel) of a 3, we have an infinite number of possibilities to transform this pixel, the dimensionality of the group is infinite.

5. Lecture 12 Feb.

5.1 Introductory Reminders

This session is a continuation of the previous one. After a brief recap, S. Mallat continues the discussion on the theme of symmetries of $f(x)$.

The three notions, separability, symmetry, and "sparsity" ("the three S's"), are all related to the regularity of the function to be learned, $f(x)$, with x evolving in high dimensions. We have seen that these three notions allow dimensionality reduction, which helps combat the curse of dimensionality.

Separability allows us to decompose the initial problem into subproblems that deal with "blocks" of x that have local interactions and little interaction between blocks. Through the concept of symmetry, we look at the invariants of the function $f(g.x) = f(x)$, and the level sets are invariant under g, forming a group structure H [34]. This group structure allows us to quotient Ω by H and keep only the effects of the function f on the equivalence classes. In other words, we reduce the dimension of the problem according to $\dim(\Omega/H) = \dim(\Omega) - \dim(H)$ (except in pathological cases). The more *a priori* information we can obtain using a group of very high dimension, the more efficient the reduction of the initial problem.

Some examples of groups:

— In certain problems, such as medical image analysis, it is advisable to normalize the pixel values [35] to typical intervals $[0, 255]$ or $[0, 1]$: the transformation is of the form $g.x(u) = \theta\ x(u)$, which is the multiplicative group with $\theta \in \mathbb{R}$;

— translation: $g.x(u) = x(u-\theta)$ within an image, $\theta \in \mathbb{R}^2$; the same applies to rotation, etc.

— deformation: $g.x(u) = x(\theta(u))$, which represents a local action that is assumed to be invertible and C^1, $\theta : \mathbb{R}^2 \to \mathbb{R}^2$. This is the group of diffeomorphisms, which has infinite dimension.

The transformations of x by the actions of the generators (Figure 14) define a hypersurface called the orbit of x, denoted O_x, which has the same "label" $f(x)$. To describe the orbit, we study its tangent hyperplanes, and the remarkable property is that they are all identical, making the group structure a very rigid one.

34. Reminder: if we know all the symmetries of f, i.e., the group G, then we know the function; which is not the case here, so we denote H as the group of known symmetries *a priori*.

35. Note: sometimes it is not possible to perform this transformation because the problem must take into account differences in pixel values between images. For example, to measure the distance of a galaxy using CCD images, one typically needs to consider not only the extension of the galaxy that could be done with renormalized pixel values but also the intensity of the pixels, which provides information about the received flux.

5.2 The Representation $\Phi(x)$

We will use the *a priori* information about the symmetry group of $f(x)$ to define the representation $\Phi(x)$, which is the step before the final classification/regression (Figure 2):

$$\tilde{f}(x) = \langle \Phi(x), w \rangle = \sum_k w_k \phi_k \tag{32}$$

Now, we want $\tilde{f}$ to be a good approximation of f, the sought-after function. To achieve this, we will impose that $\tilde{f}$ has the same invariants under $g \in G$ as f. This will be accomplished by ensuring that G is a group of symmetries of Φ. At this stage, two scenarios arise.

Either the group G is of low dimension and known (e.g., translation, rotation, multiplication), in which case we will construct $\Phi(x)$ accordingly. However, keep in mind that this type of group does not change the original problem of working in high dimensions.

Or, the group G is very large, possibly of infinite dimension: in this case, the problem is that we do not know the group. Let's revisit the case of digit deformations, for example, Figure 15 for the digit 3, but you should imagine a similar type of deformation for other digits. If we're not careful, a large deformation can turn a 3 into a 5, a 1 into a 2, etc. Therefore, if we denote $\bar{G}$ as the "total" group of diffeomorphisms, we will have to consider only a subgroup $G \subset \bar{G}$. But we cannot know it completely: certainly, small deformations undoubtedly belong to G, but from what "magnitude" a deformation does not belong to it, that is the question. We have **partial *a priori* information**: the symmetry group belongs to $\bar{G}$. This is what motivates learning, and we will "learn" G through w, the classification parameter. Thus, we do not impose that $\Phi(x)$ be invariant under $\bar{G}$ because it contains deformations that are much too large and would result in enormous errors. **However, we can impose weak forms that are linearizations, and the deformations that need to be removed are learned through the learning of w.**

5.3 Failure of Canonical Invariants

The traditional way of introducing invariants is not suitable in our case. Traditionally (or the simplest approach), we use **canonical invariants**. The idea is to "normalize" the

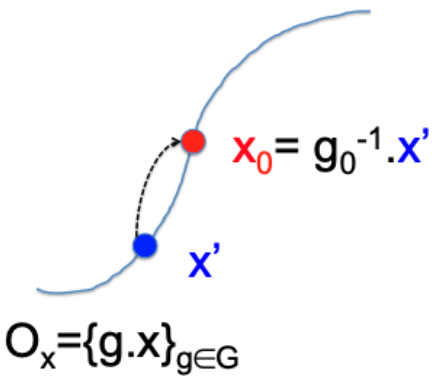

FIGURE 16 – Transformation of all $x' \in O_x$ to an invariant point x_0 under a new action.

points $x' \in O_x$ from the orbit of x to an invariant point x_0 under a new transformation (Figure 16). Consider the multiplicative group used to normalize the pixels of an image. In this case, every $x' \in O_x$ undergoes the transformation:

$$x_0 = g_\theta.x' = \theta \times x' \quad \text{such that: } x'(u) \to \frac{x'(u)}{\sum_u |x'(u)|} = x_0(u) \tag{33}$$

thus

$$\theta_0 \equiv \sum_u |x'(u)| \Rightarrow x'(u) \to \theta_0^{-1} \times x'(u) = x_0(u) \tag{34}$$

After normalization, $\sum_u |x_0(u)| = 1$, so any multiplicative factor no longer changes the configuration of x. In the case of translation:

$$g_\theta.x(u) = x(u - \theta) \tag{35}$$

and, for example, if we want to "align" all images with respect to the centroid of the pixels:

$$\theta_0 \equiv \frac{\int u|x(u)|du}{\int |x(u)|du} \tag{36}$$

then we define $x_0(u) = g_{\theta_0}^{-1}.x(u) = x(u + \theta_0)$. The new centroid is equal to 0; we have successfully "normalized" the elements of the orbit of x. However, it is important to note that all points in the orbit of x are normalized to the same point x_0, but x from different orbits are normalized to different points. For example, consider a "3" in an image; if it

is translated (rotated, etc.), we want to align it so that the centroid is at the center of the image. Once we have found the reference "3", all new "3"s are normalized to conform to the reference "3." However, for "2"s, the transformation is different (the orbit of "3"s is different from the orbit of "2"s).

So, the idea of a canonical invariant is to estimate the parameter of the transformation θ_0. **When dealing with a large group, can we do the same?** Let's consider the case of diffeomorphisms; what would be the idea (see the work of Grenender and Miller from the 1990s: *Deformable templates*)? We describe all objects based on deformations of a reference object. For example, we take standardized font digits, and we describe a handwritten digit as the transformation of its archetype. To identify a handwritten digit, we choose the archetype that is identified by the smallest deformation that transforms one into the other. We try to estimate the deformation relative to a reference template. This works well in medical imaging: if we want to describe the brain atlas, we can use MRI images from many subjects and "align" the images to create a reference atlas. Then, new images of a patient are identified relative to this atlas. **The crucial point is that we can define this reference atlas.** In the case of more diverse images, it may be difficult or even impossible to obtain such an atlas, either because there are no reference objects or because there would need to be a very large number of them. Furthermore, it would be necessary to estimate all deformations of the object to be identified relative to all reference templates! **The program cannot be completed; this approach is a failure (except in special cases).** Ultimately, we are not interested in calculating the deformation; what we want to know is whether we are dealing with a 1, 2, 3, etc. in the case of digit recognition. So, **instead of calculating the deformation, we will eliminate it**.

What are the conditions for this to work? If we revisit the representation $\Phi(x)$, we would like to have:

$$\tilde{f}(x) = \tilde{f}(g.x) \Rightarrow \langle \Phi(x), w \rangle = \langle \Phi(g.x), w \rangle \Rightarrow \langle \Phi(x) - \Phi(g.x), w \rangle = 0 \tag{37}$$

so

$$\Phi(x) - \Phi(g.x) \in V \perp w \tag{38}$$

as depicted in Figure 17. Therefore, if V is a hyperplane, this implies **linearizing the transformations**, typically taking infinitesimal g.

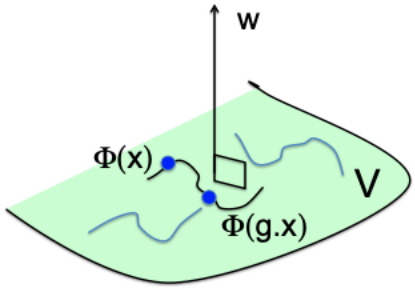

FIGURE 17 – The transformed x via the representation Φ, as well as the transformed $g.x$, lie on the same plane $V \perp w$.

5.4 Linearization of Group Action

We will focus on the group of diffeomorphisms (deformation) because it plays a very important role, whether for images whose deformations reflect various object view situations, or for voices whose variability reflects that of tones, rhythms, the articulation of a subject, etc. What does it mean to linearize a small deformation? We consider group actions with a small transformation τ:

$$g.x = x + \tau.x \tag{39}$$

Note that locally, the tangent hyperplane to the orbit of x is given by the independent vectors τ (Lie algebra). We rewrite the previous transformation as follows:

$$g.x(u) = x(u - \tau(u)) = x(u) - \nabla x(u).\tau(u) \tag{40}$$

The $\tau(u)$ represents the **displacement field**: moving along a diffeomorphism locally is equivalent to making a displacement in the tangent plane. Therefore, the generators of the tangent hyperplane are a displacement field. If $x(u)$ is smooth (C^1), then by Taylor expansion, we obtain the second equality above (note: $u = (u_1, u_2)$ in an image). However, to better understand how we can obtain invariants in a general way, we will detail this expression (Eq. 40) a bit more. We will allow ourselves to write the action of τ as a "global"

action and a "local" small action, as follows:

$$\tau(u) \approx \tau(u_0) + \nabla\tau(u_0)(u - u_0) \tag{41}$$

Note that we have

$$\nabla\tau(u) = \begin{pmatrix} \partial_{u_1}\tau_1(u) & \partial_{u_2}\tau_1(u) \\ \partial_{u_1}\tau_2(u) & \partial_{u_2}\tau_2(u) \end{pmatrix}$$

Therefore, by simple expansion, we get:

$$\begin{aligned} x(u - \tau(u)) &= x\left(u - \tau(u_0) - \nabla\tau(u_0)\ (u - u_0)\right) \\ &= x\left(\underbrace{\left(\mathbb{I} - \nabla\tau(u_0)\right)(u - u_0)}_{\text{local deformation}} + \underbrace{u_0 - \tau(u_0)}_{\text{global translation}}\right) \end{aligned} \tag{42}$$

with the two types of actions of the deformation $\tau(u)$. Moreover, we can define the size of the group of small diffeomorphisms as follows:

$$|g|_G = \underbrace{\|\tau\|_\infty}_{\text{translation}} + \underbrace{\|\nabla\tau\|_\infty}_{\text{deformation}} \tag{43}$$

where $||\tau||_\infty = \sup_u |\tau(u)|$ and $||\nabla\tau||_\infty = \sup_u ||\nabla\tau(u)|| < 1$, meaning that the largest eigenvalue of the Jacobian is smaller than 1 (invertibility). What this means is that the size of the group action of deformations is equal to the sum of the maximum value of global translation and the maximum value of local deformation. For example, if $\tau(u) = \varepsilon u$, representing an infinitesimal dilation, then $||\nabla\tau||_\infty = \varepsilon$. Thus, in principle, we have a general formulation that encompasses all types of deformations.

Can we, and how do we, linearize deformations through Φ? Ultimately, we are trying to establish conditions on Φ, and we will see that this leads us to neural networks and the problem of scale separation. By revisiting Equation 37 and injecting the expansion (Eq. 40), we have:

$$\Phi(x) - \Phi(g.x) \approx \Phi(x) - \Phi(x) - \nabla\Phi(x)(\tau.x) = -\nabla\Phi(x)(\tau.x) \tag{44}$$

So, at the first order, the set V is expressed as [36]:

$$V = \{\nabla\Phi(x)(\tau.x) \ / \ \forall x \in \Omega, \forall \tau \text{ gen. G}\} \tag{45}$$

What are the unknowns? 1) the "exact" group G, even if not explicitly mentioned, 2) the generators, which is equivalent to 1), and thus we do not have much knowledge about V. **However, with learning, we will determine w (at least what is accessible through learning), and indirectly identify V, the generators, and thus a subgroup H of the symmetries of f.** However, for this to work, we need to be able to compute the "Taylor formula" on $\Phi(x)$, so Φ **must be differentiable with respect to the action of the group $\tau.x$.** This is where the problem lies! At the very least, it would be necessary to ensure that $\Phi(x)$ is Lipschitz, meaning that under the action of a (small) transformation g from group G [37]:

$$||\Phi(x) - \Phi(g.x)|| \leq C \underbrace{d(g, \mathbb{I})}_{|g|_G} ||\Phi(x)|| \tag{46}$$

This means that the difference between the action of Φ on the transformed x (cf. $g.x$) must be of the order of g. If the problem is translation invariant (simple case), $|g|_G = \|\nabla\tau\|_\infty$, so the variation of Φ must be small compared to the gradient of the deformation (in this case, local translation). More generally, using Eq. 43, we can write the Lipschitz condition for Φ for small deformations as follows:

$$\boxed{||\Phi(x) - \Phi(g.x)|| \leq C||\Phi(x)|| \left(\underbrace{\|\tau\|_\infty}_{\text{translation}} + \underbrace{\|\nabla\tau\|_\infty}_{\text{deformation}} \right)} \tag{47}$$

To recap at this point: we do not know the group G, thus its invariants, so we linearize to eliminate small transformations, which is only possible if Φ is Lipschitz. Then we can identify the generators through learning and "quotient" the space Ω.

Now, let's get back to deformations after eliminating translations; it is necessary that $|g|_G \ll \|\nabla\tau\|_\infty$. This is a condition that we will apply systematically to all deformations;

36. Here, we introduce G, the group of "exact" symmetries of f, even though we only have access to a subgroup

37. NDJE: To understand this formulation of the Lipschitz condition, let's take $\Phi(x) = \langle x \rangle$, i.e., the average of x, and consider the action of an infinitesimal dilation $g = \mathbb{I} + \varepsilon$. Then $||\Phi(x) - \Phi(g.x)|| = ||\langle x \rangle - \langle (1+\varepsilon)x \rangle = \varepsilon||\langle x \rangle|| = d(g, \mathbb{I})||\Phi(x)||$.

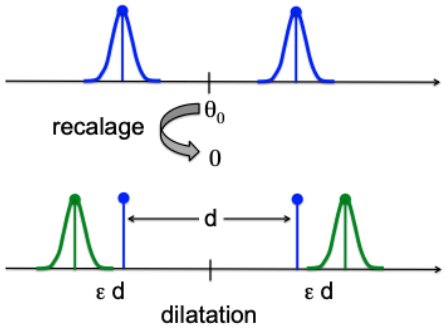

FIGURE 18 – Effects of a "global" alignment followed by a small "dilation" on two distant bumps at a distance d.

otherwise, we will not be able to identify them for quotienting. **However, the property that Φ is Lipschitz (Eq. 46) is difficult to obtain**. Indeed, a local deformation is a dilation, so the representation must be based on dilations, i.e., it involves **scale separation obtained through wavelet transform**. This will typically lead to structures that resemble neural networks.

5.5 Study of Alignment

Let θ_0 denote the centroid of x. An alignment is then given by:

$$g.x(u) = x(u - \theta_0) \tag{48}$$

The problem is that $x(u)$ is not smooth; for example, in an image, there are natural discontinuities due to different objects/landscapes/textures present in the scene, similar issues occur in a sample of sounds, etc.

If we apply a small dilation after alignment, the distance between two bumps changes from d to $d + 2d\varepsilon$ (see Figure 18). Consequently, the bumps before and after dilation have no spatial overlap. Therefore, the distance between $\Phi(x)$ and $\Phi(g.x)$ is given by:

$$||\Phi(x) - \Phi(g.x)||^2 = ||\Phi(x)||^2 + ||\Phi(g.x)||^2 - 2\int \Phi(x)\Phi(g.x)dx \approx 2||\Phi(x)||^2 \tag{49}$$

Hence, the difference $||\Phi(x) - \Phi(g.x)||$ can be very large in the end. **In fact, irregularities (high frequencies) are highly unstable under dilation**. This is why it is not sufficient to perform alignments for shape recognition (e.g., faces). We will encounter this phenomenon in the Fourier domain in Section 6.3.2.7.

5.6 Another Invariant: Group Covariance

If we have the orbit of x (O_x), let's try to impose a **linear invariant**, meaning consider combinations of the form $\sum_{g\in G} \alpha_g g.x$. However, this combination is invariant *if and only if* the α_g are identical, so this amounts to taking an **average**:

$$\sum_{g\in G} g.x \tag{50}$$

For translation, this means summing over all possible translations:

$$\sum_{\theta} x(u-\theta) \tag{51}$$

which is nothing but the mean of the signal x, which is indeed a translation invariant [38]. Similarly, the mean of the signal is invariant under dilation [39]. **Therefore, while we have an invariant for translation and dilation, namely the mean of the signal, we have lost all information about its structure.** From this perspective, a linear invariant is far too simple, even naive. The idea will be to average many channels derived from x.

Let $\tilde{\phi}(x)$ be defined as the set of images at one step of a neural network (Figure 19): each $\widetilde{\phi_k}(x)$ ($k \in \{1, \ldots, K\}$) is an image indexed by u. So now, we have a collection of images, and we will try to make them invariant according to spatial averaging:

$$\sum_{\theta} [\tilde{\phi}(x)](u-\theta) = \sum_{g\in G} g.[\tilde{\phi}(x)] \equiv \Phi(x) \tag{52}$$

38. To convince yourself, consider a periodic signal $x(t)$ on $[0, 2\pi]$. Simply summing over θ corresponds to an integral $(2\pi)^{-1}\int_0^{2\pi} x(u-\theta)d\theta$. With a change of variable $u' = u - \theta$ and the signal's periodicity, this yields $(2\pi)^{-1}\int_0^{2\pi} x(t)dt$, which is the mean of the signal.

39. With the signal from footnote 38, calculating the mean over θ of $x(u\theta)$ over a period $[0, 2\pi/u]$ yields the mean of the signal again.

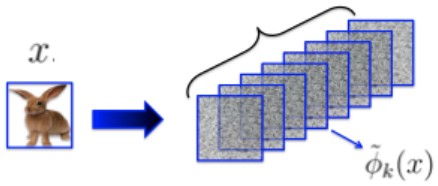

FIGURE 19 – Step in a neural network that transforms the input image x into a set of images $\left\{\tilde{\phi}_k(x)\right\}_k$ (note that they are not necessarily of the same dimension).

This is what we call **pooling**, and it is typically done at the end of the network, just before the purely *fully-connected* part of the classifier, as everything has already been spatially averaged.

The question is whether the transformation 52 is invariant, i.e., if $g_0 \in G$ then $\Phi(g_0.x) = \Phi(x)$? The condition is that $\Phi(x)$ be **equivariant under the action of group** G (note: true for all $\widetilde{\phi_k}(x)$), meaning that

$$\boxed{\Phi(g.x) = g.\Phi(x)} \tag{53}$$

in essence, Φ and g commute in their action on x. Note well the difference between **invariance** and **equivariance** [40]

$$\begin{cases} f(g.x) = f(x) & \text{invariance} \\ f(g.x) = g.f(x) & \text{equivariance} \end{cases} \tag{54}$$

For example, if we translate x, then the $\widetilde{\phi_k}(x)$ must be the same as the original image but

40. Note that S. Mallat introduces covariance, but I think it's a language habit because equivariance, which he also mentions, is probably less known. In physics, covariance is very common: e.g., covariant derivatives in fluid mechanics, covariance of equations in classical and relativistic mechanics, etc. In the case studied in the course, it's equivariance under translation, but this notion is more general. See, for example, https://arxiv.org/pdf/1805.12301.pdf for "Rotation Equivariance and Invariance in Convolutional Neural Networks" by B. Chidester et al.

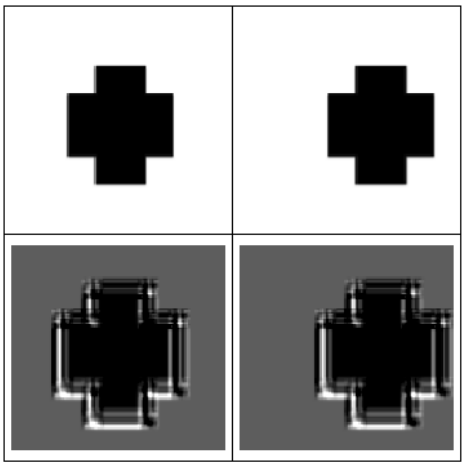

FIGURE 20 – Illustration of the equivariance property of convolution by applying a translation: (top) from left to right: original image, translated image; (bottom) from left to right: convolution of the original image, convolution of the translated image. The convolution of the translated image is the translation of the convolution of the original image. If x is the image, f is the convolution, and g is the translation, then $f(g.x) = g.f(x)$.

translated (Figure 20). Thus, if we have equivariance, then

$$\Phi(x) = \sum_{g \in G} g.[\tilde{\phi}(x)] = \sum_{g \in G} \tilde{\phi}(g.x) \Rightarrow \Phi(g'.x) = \sum_{g \in G} \tilde{\phi}(g.(g'.x)) = \sum_{g \in G} \tilde{\phi}(g.x) = \Phi(x) \quad (55)$$

because summing over all elements of the group G allows absorbing the action of g' through a change of variable.

So, in the end, we do not simply average the original image, which would eliminate all structure, but we create a collection of "channels" ($\left\{\widetilde{\phi_k}(x)\right\}_k$) that we average individually, and this only works if the channels are equivariant under translation (or more generally under group action).

5.7 Equivariant Operations

The immediate question that arises is **how to create equivariant channels**? Suppose $\phi(x)$ is **linear**, if we impose **equivariance** under G (e.g., under translation, but the same

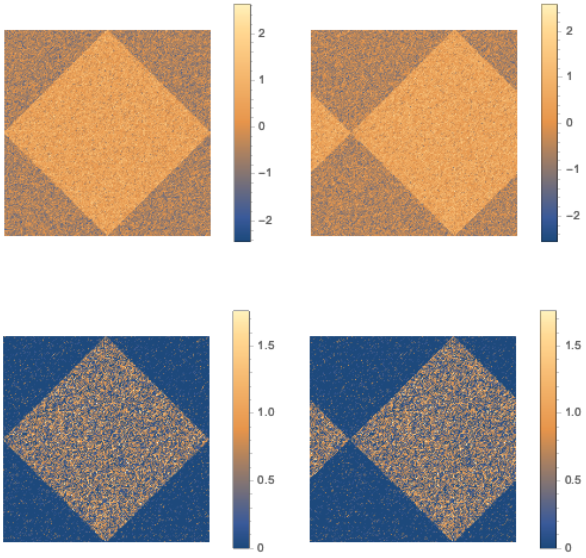

FIGURE 21 – Illustration of the equivariance property of pointwise nonlinearity: (top) from left to right: original image, translated image; (bottom) from left to right: rectification (ReLU) of the original image, and rectification of the translated image.

applies to rotation, etc.), then $\phi(x)$ **is a convolution** over G (see Figure 20). Therefore, it makes sense to use convolutions everywhere. Moreover, we need to use nonlinearities (otherwise, we lose the structure), and we want them to be equivariant as well. The choice then turns to **pointwise nonlinearities**. That is, if we denote ρ as this nonlinearity, then [41]:

$$\rho(x)(u) = \rho(x(u)) \overset{ReLU}{=\!=} \max(0, x(u)) \tag{56}$$

This means that we apply nonlinearity individually to all coordinates of x (in practice, we use ReLU or sigmoid functions), leading to equivariance of this operation as illustrated in Figure 21.

In summary, we have convolutions and pointwise nonlinearities, all of which are equivariant operations. Therefore, the same holds true if we cascade these operations. This is the case in a convolutional neural network from the steps of x to $\Phi(x)$, which are all equivariant. Then we finish it with pooling to obtain a translation-invariant result,

41. Note: I'm using a different notation than S. Mallat to differentiate it from other transformations, such as convolutions used earlier.

which can be expressed as follows:

$$\underbrace{Pool(\Phi(g.x)) = \overbrace{Pool(g.\Phi(x))}^{\text{invariance}}}_{\text{equivariance}} = Pool(\Phi(x)) \tag{57}$$

Information about the structure of the original image x is retained because we have many channels at the pooling stage in the hope that we can obtain a good approximation of $f(x)$.

The question that will occupy us later is: **how can we ensure that the result linearizes deformations**? This constraint generates specific properties on the convolutions that can be used in a CNN, and we will come across multi-scale structure: cascades of "convolution/subsampling". Before addressing this topic, let's see how sparsity allows dimension reduction.

5.8 Sparsity

This is one of the three properties that enable dimension reduction, along with separability and symmetry (recall: "the three S"). We would like the image $x \in \mathbb{R}^d$ (with a large d) to be sparse, meaning we want to reduce the number of variables to retain only a small number of non-zero ones. This can be done through a linear operator D, a list of *dictionaries*, such as:

$$D(x) = \begin{pmatrix} \dots \\ \ell_i \\ \dots \end{pmatrix}_{d' \times d} \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}_{d \times 1} = \begin{pmatrix} \dots \\ \langle \ell_i, x \rangle \\ \dots \end{pmatrix}_{d' \times 1} \tag{58}$$

Each row of D can be seen as an elementary *pattern*, and thus we project x onto each of these elementary patterns. The goal is to find a set of patterns such that almost all inner products are zero. For example, we can think of using an **image compression algorithm** that produces many nearly zero coefficients, but **to achieve sparsity, we need to add**

thresholding to it. We can use the ReLU function with a threshold $\lambda > 0$:

$$\rho_\lambda(x) \equiv \rho(x - \lambda) = \begin{cases} 0 & x < \lambda \\ x - \lambda & x \geq \lambda \end{cases} \tag{59}$$

So, if we apply this function to the output of the projection onto the patterns, each projection is thresholded, and this produces more zero coefficients as the threshold increases. Through this process, we can eliminate structures that do not correlate with the dictionary patterns.

Now, we assume **invertible** (pseudo-) dictionaries D^+ such that $D^+Dx = x$. Naturally, if we have $f(x) = f(D^+Dx)$, what we hope for is that the thresholding operation (denoising) allows signal recovery:

$$\underbrace{f(x) = \overbrace{f(D^+Dx)}^{\text{truebydefinition}}}_{} = \overbrace{f(D^+\rho_\lambda(Dx))}^{\text{denoising}} \tag{60}$$

Using thresholding, only $p \ll d'$ non-zero coefficients remain (note: we don't know which ones in advance), meaning that the vector $\rho_\lambda(Dx)$ is of low dimension. Then, we can project it with a **random matrix W**, resulting in $\mathbf{W}\rho_\lambda(Dx)$ of dimension $O(p \log d/p)$, which allows us to recover the original vector. Therefore, **making x sparse significantly reduces the dimensionality of the problem**.

However, questions arise: what is this dictionary (or these dictionaries if cascaded)? Should they be learned? We will see that we can start by using *a priori* dictionaries that sparsify the problem simply because we **know the indexing variable** u. Thus, we have *a priori* information to build a first type of dictionary (windowed Fourier analysis, wavelet analysis). And typically, these are the first filters that are highlighted in neural networks.

So, ultimately, just by knowing the indexing variable of x, i.e., the u, we can implement *a priori* information using the three S: separability, symmetry, and sparsity.

6. Lecture 26 Feb.

NDJE: I am including the last 5 minutes of S. Mallat's lecture from Feb 12 as it introduces the topic of this lesson.

6.1 Introduction

In broad strokes, before 2010, we would construct a feature vector $\Phi(x)$ from x using some mathematics, and then perform linear regression. We will see why this makes sense. We will work on **the indexing variable** u and implement the 3S (separability, symmetries, and sparsity) to reduce the dimensionality of the problem. In doing so, we will arrive at the Fourier Transform (FT). However, no one uses the FT for shape recognition. Why? Firstly, it is unstable under deformation, and secondly, it is not sparse at all: almost all coefficients are non-zero. So, we need to move away from Fourier and explore time-frequency or space-frequency representations to reveal the separability and sparsity of most signals (sounds, images). Among these dual representations, we will study the Wavelet Transform (WT), and subsequently, we will delve into filter cascades. We will discover that CNNs do these kinds of things, at least in the early stages, and that later they perform more subtle operations.

6.2 Equivariance (Covariance)

So let's consider the input $x(u)$ and an operator L such that the transformation is denoted by $Lx(u)$. We would like that if u is translated, then the transformation should also be translated: this is equivariance or covariance[42]. **If we impose that L is linear, which is the case, for example, in neural networks (cf. filters), then L is a convolution operator**. In the following, to simplify the proofs, we will consider that u is continuous[43], meaning that $u \in \mathbb{R}^q$ (cf. $u \in \mathbb{R}$ for sound, $u \in \mathbb{R}^2$ for an image, etc).

42. NDJE: later on, we use both words for the same concept as described in Sec. 5.6.
43. NDJE: see the 2018 Course Sec. 5.2 for a development of Fourier analysis in discrete settings.

Theorem 2

Let L be a linear operator, weakly continuous, equivariant under translation, then

$$\exists h \ / \quad Lx(u) = (x * h)(u) = \int x(u-v)h(v)dv \tag{61}$$

Proof 2. $x(u)$ can be represented as an integral involving Dirac delta functions, as follows:

$$x(u) = \int x(v)\delta(u-v)dv \tag{62}$$

By applying the operator L to act on u, and introducing the **impulse response** h defined as $L[\delta(u)] \equiv h(u)$, we have

$$\begin{aligned} L[x(u)] &= \int L[x(v)\delta(u-v)]dv = \int x(v)L[\delta(u-v)]dv \\ &= \int x(v)h(u-v)dv = (x*h)(u) = (h*x)(u) \end{aligned} \tag{63}$$

■

This result is not specific to translations; it holds for any group: if an **operator is equivariant** under the action of the group and is **linear**, then the operator is a **convolution on the group**.

6.3 Fourier Transform

So, once we have defined the convolution operator, the idea is to find out if we can diagonalize it, which naturally leads us to the Fourier Transform. If we subject a sinusoid $e^{i\omega u}$ to the operator, we get

$$L\left[e^{i\omega u}\right] = \int e^{i\omega(u-v)}h(v)dv = e^{i\omega u}\int h(v)e^{-i\omega v}dv = e^{i\omega u}\hat{h}(\omega) \tag{64}$$

So, firstly, $e^{i\omega u}$ **is an eigenvector of** L, and **the eigenvalue is related to its impulse response**, namely $\hat{h}(\omega)$, which is nothing but the **transfer function of the operator** L [44].

44. In higher dimensions, consider ωu as a dot product $\omega.u$ with $(u,\omega) \in \mathbb{R}^q$.

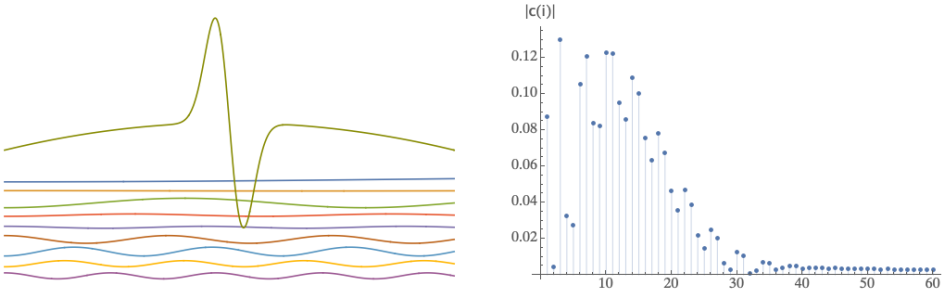

FIGURE 22 – Decomposition of a function into sinusoids, or rather, according to Discrete Cosine Transform type III. On the left are the different components in real space, and on the right is the evolution of the amplitude of the Fourier coefficients.

We will review (without proof) the basic properties of the FT [45]. The FT is ubiquitous in physics because it diagonalizes linear operators equivariant under translation, especially differential operators, making it a preferred tool for solving differential equations.

6.3.1 Inversion

Theorem 3

If $x \in \mathbb{R}^p$ is integrable, i.e., $\int |x(u)|du < \infty$, and its Fourier transform is also integrable, i.e., $\int |\hat{x}(\omega)|d\omega < +\infty$, then

$$x(u) = \frac{1}{2\pi} \int \hat{x}(\omega) e^{i\omega u} d\omega \tag{65}$$

So, this result allows us to reconstruct a signal from its Fourier transform, which is to say, from a sum of sinusoids with frequencies ω whose amplitude and phase are $\hat{x}(\omega)$. As already mentioned in Course 2018 (Sec 5.2.2), this result is not intuitive at all: in Figure 22, the very rapid local variability involves a high-frequency sinusoid, but elsewhere, rapid oscillations must be exactly compensated to leave room for a slowly varying function represented by low-frequency sinusoids.

45. S. Mallat indicates that on the course's website, there are links to references/books not only on the FT but also on time-frequency analysis in general.

Intuitively, we can sense that **the decay of the Fourier coefficients**, i.e., $|\hat{x}(\omega)|$, informs us about the **regularity** of the function $x(u)$. However, the function exhibits only **local** irregularity in real space, whereas sinusoids are completely delocalized, which is the real problem of the FT. In fact, the decay of the Fourier coefficients is determined by the worst singularity of the function.

We extend Theorem 3 to functions in $L^2(\mathbb{R})$, i.e., square-integrable functions, where $\int |x(u)|^2 du < +\infty$, and we also consider an extension to distributions.

6.3.2 Some Properties of the Fourier Transform

NDJE. See also the 2018 course on a point on TF convention sets.

6.3.2.1 Convolution Product

Theorem 4

The Fourier transform of a convolution product is the product of the Fourier transforms, provided they exist, i.e.,

$$\widehat{x * h}(\omega) = \hat{x}(\omega)\hat{h}(\omega) \tag{66}$$

This directly follows from sinusoids being eigenvectors of the convolution operator and using the inversion property.

6.3.2.2 Plancherel's Formula

Theorem 5

Let x_1 and x_2 be elements of $L^2(\mathbb{R}^p)$ (i.e., functions of finite energy), and the inner product is defined as

$$\langle x_1, x_2 \rangle = \int x_1(u) x_2^*(u) du \tag{67}$$

then we have the following equation[a]

$$\langle x_1, x_2 \rangle = \frac{1}{2\pi} \int \hat{x}_1(\omega) \hat{x}_2^*(\omega) d\omega = \frac{1}{2\pi} \langle \hat{x}_1, \hat{x}_2 \rangle \tag{68}$$

a. NDJE: Pay attention 1) to the FT convention and 2) to the dimension p to obtain the normalization constant in Plancherel's formula.

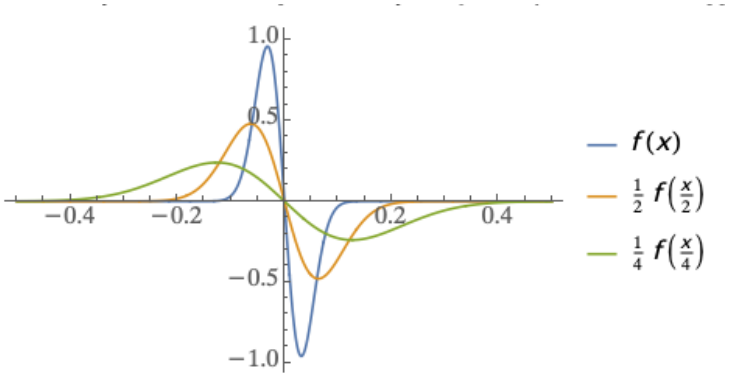

FIGURE 23 – Examples of dilations of $f(x)$.

This is a fundamental isometry property and a direct consequence of the convolution theorem.

6.3.2.3 Dilation

Let the function $x_s(u)$ be defined in dimension p based on $x(u)$ as follows (Figure 23):

$$x_s(u) = \frac{1}{s^p}\, x(\frac{u}{s}) \tag{69}$$

If $s > 1$, we dilate the function, while if $0 < s < 1$, we contract it. In the Fourier domain, we easily obtain

$$\hat{x}_s(\omega) = \frac{1}{s^{p-1}}\hat{x}(s\omega) \tag{70}$$

which means that if $x(u)$ is dilated, its Fourier transform is contracted, and vice versa.

6.3.2.4 Derivatives

Differentiation is represented by multiplication by $i\omega$, which, after successive differentiations, gives the relationship (note that the function $x(u)$ tends to 0 at infinity):

$$\hat{x}^{(q)}(\omega) = (i\omega)^q \hat{x}(\omega) \tag{71}$$

In particular, the analysis of Sobolev spaces considers derivatives not in real space but in Fourier space through the behavior of the product $\omega^q|\hat{x}(\omega)|$ at infinity.

6.3.2.5 Translation

Let the function defined by $x_\tau(u) = x(u - \tau)$, then its Fourier transform is equal to

$$\hat{x}_\tau(\omega) = e^{-i\omega\tau}\hat{x}(\omega) \tag{72}$$

What is the condition for a linear operator to be invariant under translation? It requires that $e^{-i\omega\tau}$ is independent of τ, i.e., that $\omega = 0$. However,

$$\hat{x}_\tau(0) = \int x(u)du \tag{73}$$

is the **mean** of the signal. This is a result that generalizes; **if a linear operator is invariant under the action of a group, we sum the coefficients over all orbits to obtain the mean**. What about non-linearity?

We have seen that "realignments" (Sec. 5.5) are unstable under dilation, so if we want a nonlinear operator invariant under translation, we need something else. The Fourier transform (via Plancherel) offers us a great tool. Let's take the absolute value of $\hat{x}$ to define the representation Φ (here we have a discrete variable u):

$$\Phi(x) = \{|\hat{x}(\omega)|\}_\omega$$

However, nobody uses the FT for shape recognition. Why? One reason we will see later is that the FT does not provide a sparse representation and does not allow zooming in on structures. The second reason we discussed earlier is that it is not stable under local deformation (high frequencies).

6.3.2.6 Instability Under Deformation

The problem of instability under deformation guides us in defining the right representation and most constraints. A deformation is represented by the action of the group of diffeomorphisms as follows:

$$x(u) \to g.x(u) = x(\theta(u)) = x(u - \tau(u))$$

with the function $\theta \in C^1$ (continuously differentiable). This function is also invertible when considering "small" deformations (which do not change the class of x), which trans-

lates into the action of a small local translation $\tau \in C^1$. To ensure that $\mathbb{I} - \tau$ is also invertible, we impose the constraint $\|\nabla\tau\|_\infty < 1$ (in 1 dimension, this translates to $|\tau'(u)| < 1$).

To illustrate the instability of the FT, take $\tau(u) = \varepsilon u$ with $|\tau'(u)| = \varepsilon \ll 1$. If $\varepsilon \ll 1$, we should not expect the representation $\Phi(x)$ to be disrupted. For this to happen, Φ should be differentiable, or at least Lipschitz (Eq. 46), which means in this case:

$$\|\Phi(g.x) - \Phi(x)\| \leq C\|\Phi(x)\| \underbrace{\|\nabla\tau\|_\infty}_{\text{deformation}} = C\|\Phi(x)\|\varepsilon$$

In other words, we want $\|\Phi(g.x) - \Phi(x)\|$ to be of the same order of magnitude as the deformation amplitude ε. Let's see what happens in Fourier space because our representation is $\Phi(x) = |\hat{x}(\omega)|$. However [46],

$$g.x = x[u - \tau(u)] = x[(1-\varepsilon)u] \xrightarrow{FT} \hat{x}[(1+\varepsilon)\omega] \tag{74}$$

If the signal has high-frequency power like two "bumps" centered at $\omega = \pm\omega_0$, then by the action of τ, the power will concentrate around ω_0' such that $(1+\varepsilon)\omega_0' = \pm\omega_0$ or $\omega_0' = \pm(1-\varepsilon)\omega_0$ (see Figure 18, but this time the sign of ε has changed because we are in the Fourier domain). As a result, **the bumps move towards lower frequencies by an amount** $\varepsilon\omega_0$. **However, there is no guarantee that this quantity is small enough for the bumps before and after the transformation not to overlap**. Therefore, instead of having $\|\Phi(x) - \Phi(g.x)\| \propto \varepsilon$, we have something more like [47]

$$\|\Phi(x) - \Phi(g.x)\| = \||\hat{x}(\omega)| - |\hat{x}_\tau(\omega)|\| = 2\||\hat{x}(\omega)|\| = 2||\Phi(x)|| \tag{75}$$

In this way, $\Phi(x)$ **cannot satisfy the Lipschitz condition** as soon as there is high-frequency power. However, these high frequencies come from all the small details of the function $x(u)$, which should not interfere with recognition/classification.

Therefore, we need representations that are invariant under translation, **stable under deformation** to keep the difference $\|\Phi(x) - \Phi(g.x)\|$ of the same order as the deformation

46. In p dimensions, there is a multiplicative term $(1-\varepsilon)^{1-p}$ which is irrelevant for the purpose here.

47. NDJE: you can see this by taking $x(u) = e^{-(1/2)u^2\sigma^2}\cos[u\omega_0]$, where the Fourier transform corresponds to two Gaussians centered at $\pm\omega_0$ with sigmas σ. For $\omega_0 = 2$ and $\sigma = 0.1$, you can see that for $\varepsilon = 0.5$, the two "bumps" no longer overlap, and the difference $\||\hat{x}(\omega)| - |\hat{x}((1+\varepsilon)\omega)|\|$ is no longer given by the first-order expansion in ε but by the two series of "bumps" before and after dilation.

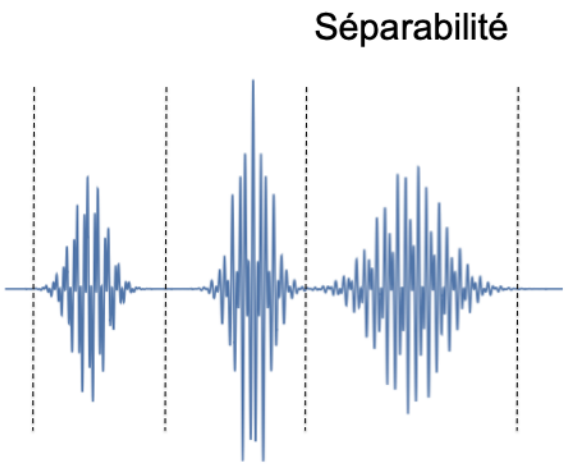

FIGURE 24 – Example of using separability to isolate signals with information, and on each interval, we can perform a sparse representation (sparsity).

amplitude (at least to first order). To do this, **we need to go through scale separation**.

6.4 Time-Frequency Representation with Windows

The "3S" (separability, symmetry, sparsity) motivate the Time-Frequency representation [48]. Note that this is a subject that emerged in the 1950s with Dennis Gabor (1900-1979), a Hungarian physicist who received the Nobel Prize in 1971 for inventing holography. He was interested in the connections between Quantum Mechanics and Information Theory, particularly how to represent information, for example, when it is carried by sounds, speech. The problem is that, regarding speech, phonemes are oscillating, which would motivate the use of TF for frequency analysis, but they are also **localized in time**. Therefore, we need a combined Time-Frequency analysis. To do this, we will use **separability** (Fig. 24) to isolate the time intervals where there is a signal, extracting lower-dimensional information from each interval, and we will use a representation where most coefficients are nearly zero (**sparsity**).

48. More generally, any other pair of dual variables by TF.

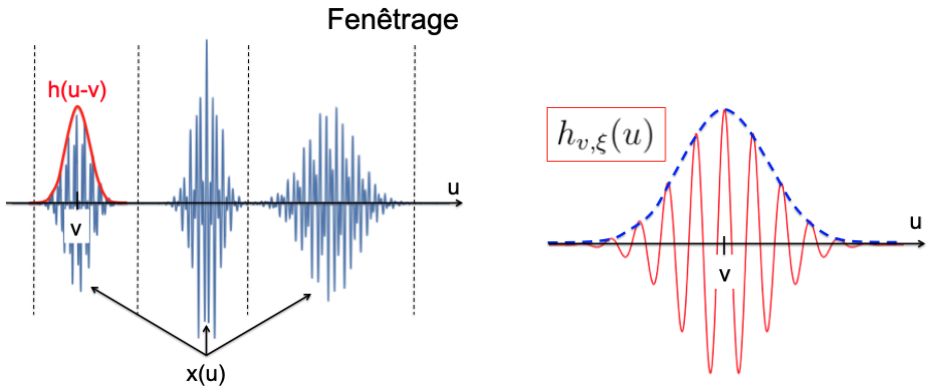

FIGURE 25 – (left): Example of using a window h centered at $u = v$ to isolate a part of the signal $x(u)$. The generalization to 2D is immediate with a two-dimensional Gaussian window. (right): Representation of the function $h_{v,\xi}$.

6.4.1 Windowing (Short Time Fourier Transform)

The idea of time-frequency analysis will involve revisiting the Fourier integral, and since we want to introduce localization (separability), we will multiply $x(u)$ by a **window** (Fig. 25 left) h to isolate a slice of the signal, and then apply a Fourier Transform (FT) to extract the main frequencies. Thus, we define the **Short Time Fourier Transform** as follows:

$$\boxed{Sx(v,\xi) = \int x(u)h(u-v)e^{-i\xi u}du = \int x(u)h^*_{v,\xi}(u)du = \langle x, h_{v,\xi}\rangle} \tag{76}$$

This is an extension of the FT where we simultaneously consider the time variable v and the frequency variable ξ. What are the properties of this type of transformation, and how do we choose the window h?

We can view the transformation as a correlation of the signal $x(u)$ with a function $h_{v,\xi}(u)$, which is essentially a window centered at v multiplied by a sinusoid (Fig. 25 right). In this context, **the size of the window is fixed**, and the only parameters that can vary are the central position v and the frequency ξ. In the literature, $h_{v,\xi}(u)$ is called a *time-frequency atom*, due to its connection with Quantum Mechanics (cf. wave packets).

Now, using the Plancherel formula, we can have a frequency view of Equation 76:

$$Sx(v,\xi) = \frac{1}{2\pi}\int \hat{x}(\omega)\hat{h}^*_{v,\xi}(\omega)d\omega = \frac{1}{2\pi}\langle \hat{x}, \hat{h}_{v,\xi}\rangle \tag{77}$$

Thus, $Sx(v,\xi)$ **is also the correlation between the FT of** $x(u)$ **and the FT of an atom**. What is the form of this atom? Using 1) that $h_{v,\xi}(u) = \tilde{h}(u-v)e^{i\xi v}$ with $\tilde{h}(u) = h(u)e^{i\xi u}$, and 2) the rule that there is a link between translation and phase multiplication, so $\hat{\tilde{h}}(\omega) = \hat{h}(\omega - \xi)$, we have:

$$\boxed{h_{v,\xi}(u) = h(u-v)e^{i\xi u}} = \tilde{h}(u-v)e^{i\xi v} \xrightarrow{FT} \boxed{\hat{h}_{v,\xi}(\omega) = e^{-i(\omega-\xi)v}\hat{h}(\omega-\xi)} \tag{78}$$

The question is, what window should we choose? We would like to have good localization both around $u = v$ and $\omega = \xi$. Can we define a box as small as we want? The answer is no, as you can see in Figure 27. **Heisenberg's uncertainty principle is at play to constrain the size of the box**. We will impose that $||h||^2 = 1$, meaning that the integral of its square is equal to 1 (note that this is the case for the Gaussian expression in Eq. 92). Let's define time and frequency widths. For the time and frequency parts, we can define variances with the corresponding masses:

$$\begin{aligned}\sigma_u^2 &\equiv \int (u-v)^2 \left\|h_{v,\xi}(u)\right\|^2 du \\ &= \int (u-v)^2 \left|h(u-v)\right|^2 du = \int u^2 h^2(u)du \qquad \left(\text{note } \int h^2(u)du \equiv 1\right) \end{aligned} \tag{79}$$

$$\sigma_\omega^2 \equiv \frac{1}{2\pi}\int (\omega-\xi)^2 \left\|\hat{h}_{v,\xi}(\omega)\right\|^2 d\omega = \frac{1}{2\pi}\int \omega^2 |\hat{h}(\omega)|^2 d\omega \tag{80}$$

Notice that σ_u and σ_ω do not depend on the location (v,ξ) in the (u,ω) plane. But can we fix σ_u and σ_ω as small as we want? Let's take simple examples:

- If we take a Dirac centered at $u = v$, thus localized in time to the extreme, in Fourier, the Dirac transforms into the function 1, which is completely delocalized in frequency;
- Symmetrically, a sine wave is localized at one frequency, it's a Dirac in frequency, and by the Inverse FT, it transforms into 1 on the time scale, thus being completely delocalized.

So squeezing to the extreme in one dimension does not work. Let's take a window $h(u)$

and perform a dilation s, in FT this gives:

$$\frac{1}{s}h(u/s) \xrightarrow{FT} \hat{h}(s\omega) \tag{81}$$

Therefore, if we localize in time with a small Δu (scaling in s), then the width in frequency increases with $\Delta\omega$ by a scaling of $1/s$.

However, let's view the problem from another angle: that of derivatives. To specify that a function is regular, we generally say that it has derivatives, and that these derivatives have finite energy (cf. square-integrable): $\int \left|h^{(p)}(u)\right|^2 du < \infty$. Using the Parseval equality, we have:

$$\int \left|h^{(p)}(u)\right|^2 du = \frac{1}{2\pi}\int \left|\widehat{h^{(p)}}(\omega)\right|^2 d\omega \tag{82}$$

$$= \frac{1}{2\pi}\int \omega^{2p}\left|\hat{h}(\omega)\right|^2 d\omega \tag{83}$$

So, the regularity of the function is reflected in the Fourier domain by

$$\boxed{\int \left|h^{(p)}(u)\right|^2 du < +\infty \Leftrightarrow \int \omega^{2p}\left|\hat{h}(\omega)\right|^2 d\omega < +\infty} \tag{84}$$

Therefore, **the more regular the function, the faster the coefficients (or the FT) must decay**. So, if we try to concentrate a window in time, we inevitably create an irregular function, so its Fourier coefficients cannot decay rapidly, leading to frequency delocalization.

However, we can sense that the area of the box seems constant, as you can also see in Figure 27 when ξ varies. The uncertainty principle accounts for all the effects of squeezing one or the other window $h_{v,\xi}(u)$ and $\hat{h}_{v,\xi}(\omega)$ by providing a bound on the time-frequency localization area.

Theorem 6

Consider the window (in this case, the non-oscillatory part) $h \in L^2(\mathbb{R})$ such that $\|h\|^2 = 1$ (finite energy), centered at 0, i.e., $\int u|h(u)|^2 du = \int \omega|\hat{h}(\omega)|^2 d\omega = 0$. For example, one can think of the Gaussian in Eq. 92. Then, if we define the variances

in time and frequency as follows:

$$\sigma_u^2 = \int u^2 |h(u)|^2 du \qquad \sigma_\xi^2 = \frac{1}{2\pi} \int \omega^2 |\hat{h}(\omega)|^2 d\omega \tag{85}$$

then

$$\boxed{\sigma_u \sigma_\omega \geq \frac{1}{2}} \tag{86}$$

This theorem is an expression of the **Uncertainty Principle** in Quantum Mechanics, where the squared wave function (e.g., the $h_{v,\xi}(u)$) gives the probability of finding an electron near a certain point: one cannot simultaneously measure the position and momentum with infinite precision, as they are dual quantities by the FT. This result is a consequence of the regularity properties of the localization function.

Proof 6. Let's take the product $\sigma_u^2 \sigma_\omega^2$, it follows that

$$\sigma_u^2 \sigma_\omega^2 = \left(\int u^2 |h(u)|^2 du \right) \times \left(\frac{1}{2\pi} \int \omega^2 |\hat{h}(\omega)|^2 d\omega \right) \tag{87}$$

Notice that $\widehat{h'(u)}(\omega) = i\omega \hat{h}(\omega)$ and using Plancherel, we have

$$\frac{1}{2\pi} \int d\omega \ \omega^2 |\hat{h}(\omega)|^2 = \frac{1}{2\pi} \int d\omega \ |\widehat{h'(u)}(\omega)|^2 = \int du \ |h'(u)|^2 \tag{88}$$

Then, by applying the Cauchy-Schwarz inequality, we get

$$\sigma_u^2 \sigma_\omega^2 = \left(\int u^2 |h(u)|^2 du \right) \times \left(\int |h'(u)|^2 du \right) \geq \left(\int |u \ h^*(u) h'(u)| \, du \right)^2 \tag{89}$$

However, notice that

$$|u \ h^*(u) h'(u)| \geq \frac{u}{2} \left(h^*(u) h'(u) + h(u) h'^*(u) \right) = \frac{u}{2} \times \frac{d|h(u)|^2}{du} \tag{90}$$

and using integration by parts with the constraint [49] $u|h(u)|^2 \xrightarrow[u \to \pm\infty]{} = 0$, we obtain

$$\sigma_u^2 \sigma_\omega^2 \geq \frac{1}{4} \left(\int |h(u)|^2 \ du \right) = \frac{1}{4} \tag{91}$$

49. This constraint simplifies the proof, but the result is more general and does not require this constraint.

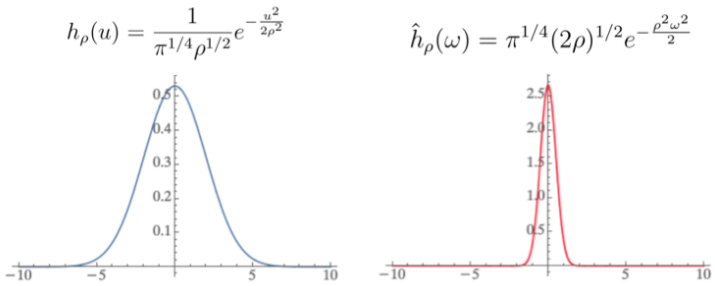

FIGURE 26 – Gaussian with width $\rho = 2$ (left) and its FT (right).

■

Note that the result relies on the fact that if we want to squeeze $h(u)$, its integral becomes very large and delocalizes in Fourier.

In the case of a **Gaussian window** satisfying $\|h\|^2 = 1$ and centered, we have

$$\boxed{h_\rho(u) = \frac{1}{\pi^{1/4}\rho^{1/2}} e^{-\frac{u^2}{2\rho^2}} \xrightarrow{FT} \hat{h}_\rho(\omega) = \pi^{1/4}(2\rho)^{1/2} e^{-\frac{\rho^2\omega^2}{2}}} \tag{92}$$

and

$$\sigma_u^2 = \frac{\rho^2}{2}, \ \sigma_\omega^2 = \frac{1}{2\rho^2} \Rightarrow \sigma_u^2\sigma_\omega^2 = \frac{1}{4} \tag{93}$$

So, in this case, the lower bound is achieved; it's the best we can do. Figure 26 shows ellipses with semi-major axes $(\sigma_u, \sigma_\omega)$ evolving depending on the location of introspection.

6.4.2 The spectrogram

Let's go back to the definition of the STFT (Eq. 76) with x translated $x_\tau(u) = x(u-\tau)$ with τ very small compared to the window of $h(u)$ then

$$\begin{aligned} Sx_\tau(v,\xi) &= \int x(u-\tau)h(u-v)e^{-i\xi u}du = e^{-i\xi\tau}\int x(u)h(u+\tau-v)e^{-i\xi u}du \\ &\simeq e^{-i\xi\tau}Sx(v,\xi) \end{aligned} \tag{94}$$

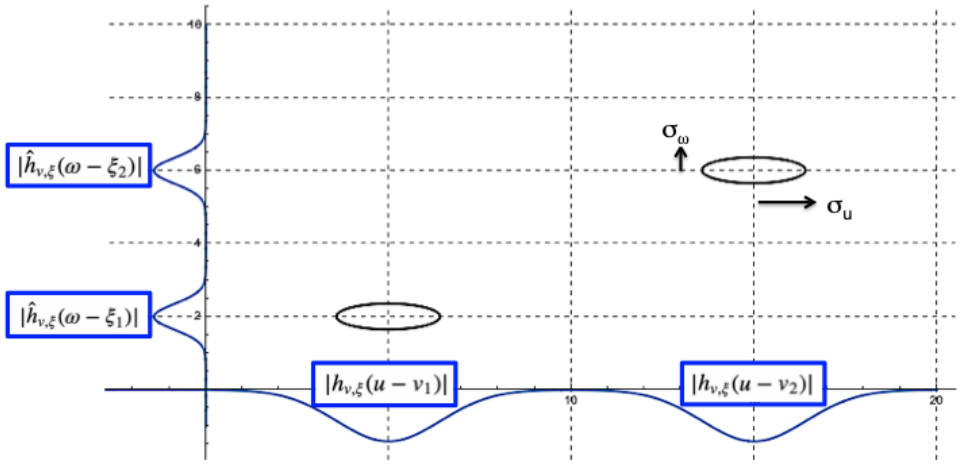

FIGURE 27 – A representation in the (u, ω) plane of the localisation of introspection by an atom $h_{v,\xi}$ in the (u) time domain, and by its Fourier transform in the (ω) frequency domain

.

So, if we want to be indifferent to small translations, all we have to do is take the modulus $|Sx(v, \xi)|$ to remove the phase term, **this is the spectrogram**. Does this tool make it easier to see structures? In figure 27, we have represented the "box" of introspection, here represented by an ellipse of semi-major axes (σ_u, σ_ξ), obtained by an atom $h_{v,\xi}$, which is moved in the plane (u, ω).

6.4.3 Some Examples

Figure 28 presents the time-domain representation and the spectrogram of a signal composed of 2 chirps, while Figure 29 shows the spectrogram of a signal with a divergence at $t \approx 1$: it models the signal of bats. Note that in the vicinity of the singularity, the box size results in "smearing" as it becomes too wide in frequency.

Such phenomena are generic whenever there are transients. For example, when one wants to detect the attack phase of a piece of music, it is desirable to pinpoint the exact moment. However, the Uncertainty Principle dictates that as the window becomes larger in frequency, it becomes impossible to distinguish between harmonics and instruments.

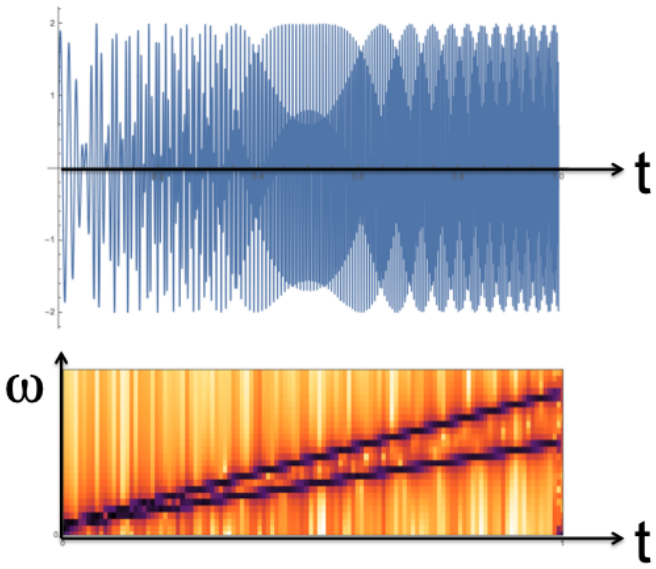

FIGURE 28 – Spectrogram of the superposition of 2 chirps. A chirp is a signal of the form $x(t) = \sin(\phi_0 + \pi t^2 + 2\pi f_0 t)$. While the time-domain representation is complex, the spectrogram exhibits 2 linearly increasing frequency functions.

Finally, the structure of a spectrogram (Fig. 30) can also be very complex with various structures that one would like to capture.

6.4.4 Limitations of the STFT

Through the spectrograms, we can observe limitations of windowing in the presence of singularities/transients. There is another limitation related to its instability under deformation: indeed, the Short Time Fourier Transform is, in essence, a Fourier Transform and shares this problem. If two notes only differ by a small dilation, then

$$x(u) \to x(u(1-\tau)) \Rightarrow Sx_\tau(v,\xi) \approx Sx(v,(1+\tau)\xi) \tag{95}$$

So, if $Sx(v,\xi)$ concentrates at $\xi = 1, 2, 3, \ldots$, then $Sx_\tau(v,\xi)$ concentrates at $1-\tau, 2-2\tau, 3-3\tau, \ldots$. This means that as we move to higher frequencies, the values of the concentration frequencies of $Sx(v,\xi)$ and $Sx_\tau(v,\xi)$ will shift, making it increasingly difficult to recognize

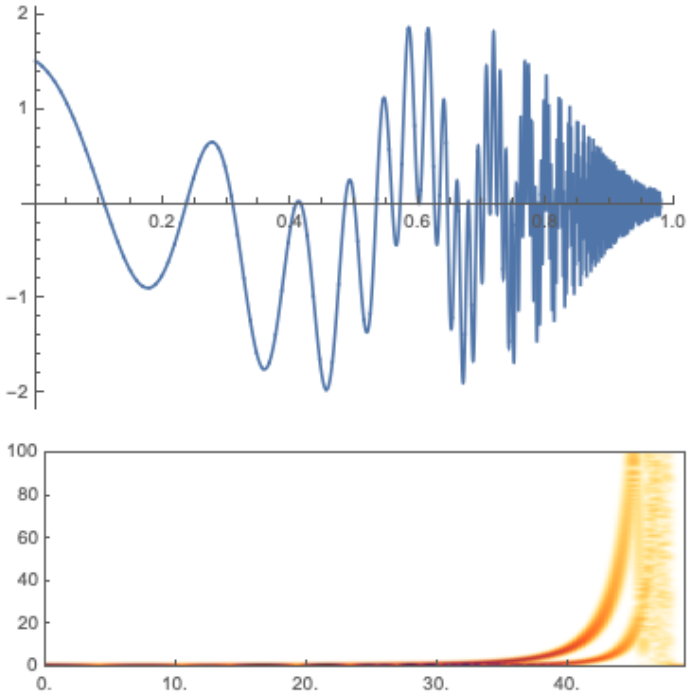

FIGURE 29 – Spectrogram of the superposition of 2 signals of the form $\cos(a/(t-1)^2)$ with a logistic damping factor.

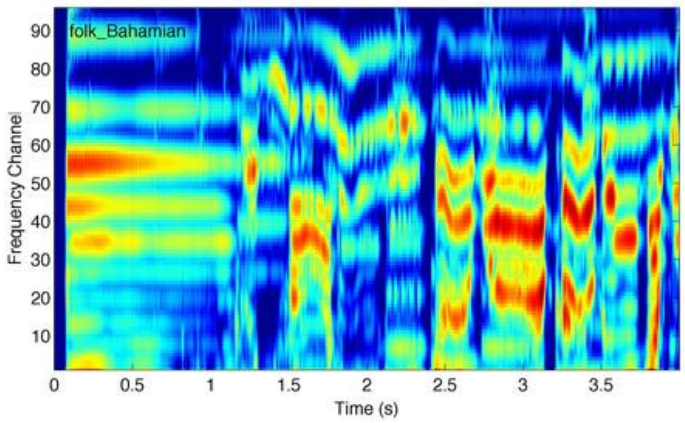

FIGURE 30 – Spectrogram of music.

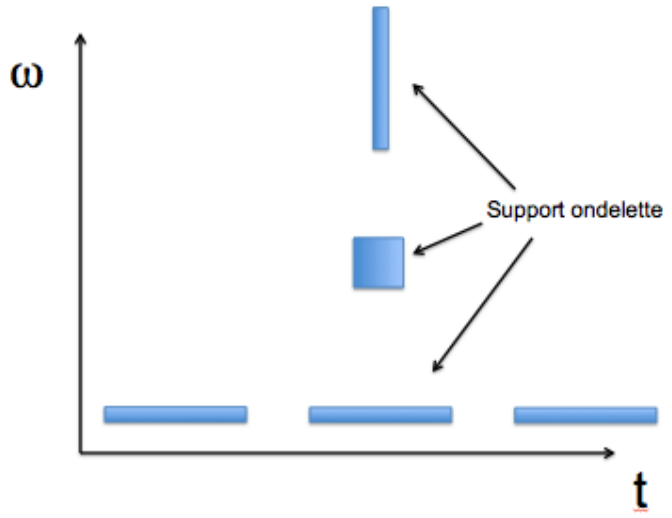

FIGURE 31 – Adjustment of the analysis box size according to frequency.

that these two sounds are actually similar.

For this reason, windowed analysis is not used in sound processing, nor is it what the ear does. The trick is to change the size of the analysis boxes, as shown in Figure 31. We will practice a **multi-resolution analysis using Wavelets**, where the concept of **scale** replaces the concept of frequency. This leads to descriptors (MFCC) that capture many invariants (changes in amplitude, translation in time, in amplitude). They are very compact and are based on signal sparsity. Until 2010, they dominated sound analysis, but with neural networks, they disappeared, but now they are reappearing as common filters are being identified.

7. Lecture 4 Mar.

7.1 Preamble

NDJE: In this session, S. Mallat takes us into the world of **Wavelets**. *You can also refer to Section 6 of the 2018 Course. However, S. Mallat not only demonstrates the classical properties of Wavelets in time-frequency analysis but also proves that it is the suitable*

representation for linearizing ***deformations****, which is crucial for scale separation and recognition. Furthermore, in CNNs, we find the famous filtering/downsampling cascades that produce these scale separations, and we will see the connection with Wavelets.*

7.2 Time-Frequency with Wavelets

Let's motivate the search for a representation other than the windowed Fourier transform (Sec. 6.4.1):

- We want to **precisely isolate transients**, so as not to displace temporal information (especially at high frequencies).
- We want to be **stable under deformation**: meaning that a small deformation should also result in a small difference in representation, unlike the phenomenon described in Sec. 6.4.4.
- Finally, the Uncertainty Principle imposes limitations on the size of the introspection box (Th. 6), but we want to be able to **capture both small and large time scales**. For example, in music, think of the time scales of a note, a chord, a melody, movement, etc. So there are structures at all scales, and capturing long-duration ones is a current research effort.

7.2.1 The Wavelet Family

Consider a function $\psi(u) = h(u)e^{i\xi u}$ where $h(u)$ is a window (see, for example, Figure 25). However, compared to the STFT (Eq. 76), **we fix the frequency** ξ. We perform two transformations: **a translation** ($b \in \mathbb{R}$) **and a dilation** ($s \in \mathbb{R}^{+*}$) applied to the base wavelet $\psi(u)$ to obtain the family

$$\boxed{\psi_{s,b}(u) \equiv \frac{1}{s}\psi\left(\frac{u-b}{s}\right) \equiv \psi_s(u-b)} \tag{96}$$

NDJE: For readers of the 2018 course notes or S. Mallat's book, they may have noticed the change in normalization, from $1/\sqrt{s}$ to $1/s$. I discussed this with him, and it seems interesting to trace the motivation for this normalization change. In the 2018 context, we were in the "inner product" view of the Wavelet transform with the idea of constructing

an orthonormal basis with sampling in scale and space. In this context, the normalization of $1/\sqrt{s}$ *is necessary, as well as the subsequent appearance of the complex conjugate in the definition of the Wavelet coefficient* $Wx(v,s)$*. In this 2020 course, S. Mallat wants to emphasize the "convolutional filtering" view of the Wavelet transformation, so we may want to define* $Wx(v,s)$ *as a convolution product and also simplify calculations in the Fourier space, hence the normalization change to* $1/s$ *in 1D, and subsequently to* $1/s^2$ *in 2D. This clarification, which some might call "a simple normalization", helps to understand the philosophy of this year's perspective.*

7.2.2 Wavelet Transform

We define the Wavelet Transform (WT) of the function $x(u)$ with the wavelet ψ_s as (unless otherwise specified, integration bounds are $\pm\infty$):

$$\boxed{Wx(v,s) \equiv (x * \psi_s)(v) = \int x(u)\psi_s(v-u)du = \int x(v-u)\psi_s(u)du} \tag{97}$$

So, **the wavelet transform is seen as the convolution** of x by the wavelet ψ_s.

Using rules for the Fourier transform (Tab. 1), we have

$$\widehat{\psi_s}(\omega) = \hat{\psi}(s\omega) \qquad \text{and} \qquad \widehat{\psi_s(v-u)}(\omega) = e^{i\omega v}\widehat{\psi_s}(\omega) \tag{98}$$

If we now apply the Parseval's theorem, the integral giving $Wx(v,s)$ becomes

$$\boxed{Wx(v,s) = \frac{1}{2\pi}\int \hat{x}(\omega)\hat{\psi}(s\omega)e^{i\omega v}d\omega} \tag{99}$$

indicating that the integral has contributions in the frequency domain where $\hat{\psi}(s\omega)e^{i\omega v}$ is non-zero. Since

$$\hat{\psi}(\omega) = \hat{h}(\omega - \xi) \tag{100}$$

we impose a condition on the wavelet such that

$$\boxed{\int_{-\infty}^{+\infty} \psi(u)du = \hat{\psi}(0) = \hat{h}(-\xi) = 0} \tag{101}$$

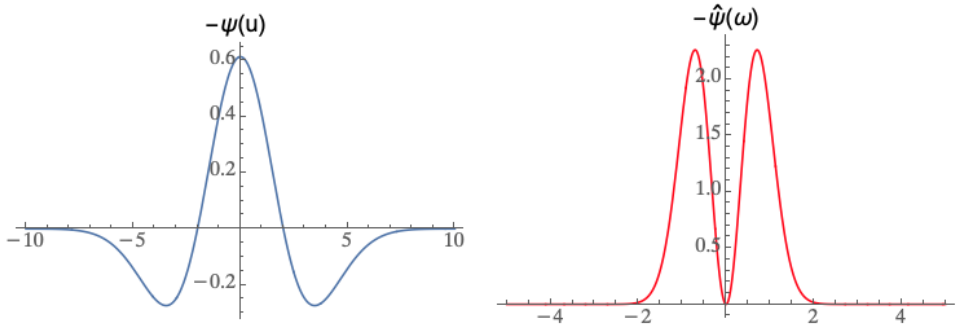

FIGURE 32 – Example of a Mexican hat-shaped wavelet (Eq. 102), along with its Fourier transform (Eq. 103) with $\sigma = 2$. Note: This is the default wavelet in Mathematica.

We can either deal with *complex* (analytical) wavelets, which are mainly used to study signals with well-defined frequency, or *real* wavelets for rapidly changing time-domain signals. *NDJE: In the following, we will consider real wavelets, and a section will mention the differences in the case of analytical wavelets that could not be covered by S. Mallat due to lack of time.*

Here's an example of a real ψ that satisfies the condition in Eq. 101; it has the shape of a Mexican hat:

$$\psi(u;\sigma) \equiv \frac{2\left(\frac{u^2}{\sigma^2} - 1\right) e^{-\frac{u^2}{2\sigma^2}}}{\pi^{1/4}\sqrt{3\sigma}} \tag{102}$$

Its Fourier transform is then given by

$$\hat{\psi}(\omega;\sigma) = -2\sqrt{\frac{2}{3}}\pi^{1/4}\sigma^{5/2}\omega^2 e^{-\frac{1}{2}\sigma^2\omega^2} \tag{103}$$

Figure 32 shows the graphs of ψ and $\hat{\psi}$ for $\sigma = 2$.

Depending on the value of s (dilation/contraction), the position of the maximum and the width of the wavelet in frequency change, as well as the position of the first zero in time. In the case of the Mexican hat wavelet that we dilate, the position of the first zero in time is given by $u_s^* = s\sigma \propto s$, and that of the maximum in frequency is located at $\omega_s^* = (\sqrt{2}/\sigma)/s \propto 1/s$. When s decreases (increases), the position of the maximum in frequency shifts to higher (lower) frequencies. The opposite is true for the position of the

first zero in time. This is illustrated in Figure 33.

Compared to the windowed Fourier transform (Fig. 27), **the size of the introspection window changes in the time-frequency plane**, as can be seen in Figure 34. However, **the area of the box remains constant** due to the uncertainty principle. So, it's not that we have better resolution with a WT, but **at low frequencies, where the signal evolves slowly, the size in u is large, whereas at high frequencies, the box adjusts to achieve better temporal localization**. Finally, **by shifting along u, at a fixed ω, the size of the box doesn't change**.

7.2.3 Some Examples

In Figure 35, the signal $x(u) = \sin(2000\pi u^2)$ is analyzed both by a Windowed Fourier Transform (STFT) and a Wavelet Transform (WT), where the size of the introspection window follows the scheme in Figure 34. For the STFT, the analysis box is of constant size, so the time-frequency delocalization remains constant with respect to frequency and time. For the WT, the effect of enlarging the box along the frequency axis as the frequency increases is clearly visible; it completely dominates the scalogram [50]. For this type of signal that doesn't exhibit singularities, the STFT is more suitable because it is more parsimonious, meaning there are fewer coefficients beyond a threshold (e.g., 0.5).

The phenomenon of loss of frequency resolution at high frequencies in the WT becomes even more problematic when two similar signals are superimposed (see Figure 36). The wavelet is too wide in frequency, and beating phenomena between the two signals become visible, making it difficult to separate them. So, **here as well, the WT is not useful because there are no transient phenomena**.

NDJE: Next, for transient phenomena, it is better to refer to the video of the course because S. Mallat plays the sound effects.

Now, if the signal is slightly dilated, we can wonder how its wavelet transform evolves. This is the counterpart of the study in Sec. 6.4.4. We have

$$x(u) \to x(u(1-\tau)) \Rightarrow v \to v(1-\tau), s \to s(1-\tau), \xi \to \xi(1+\tau) \tag{104}$$

50. NDJE: The term *scalogram* is the accepted term for the time-frequency analysis in WT.

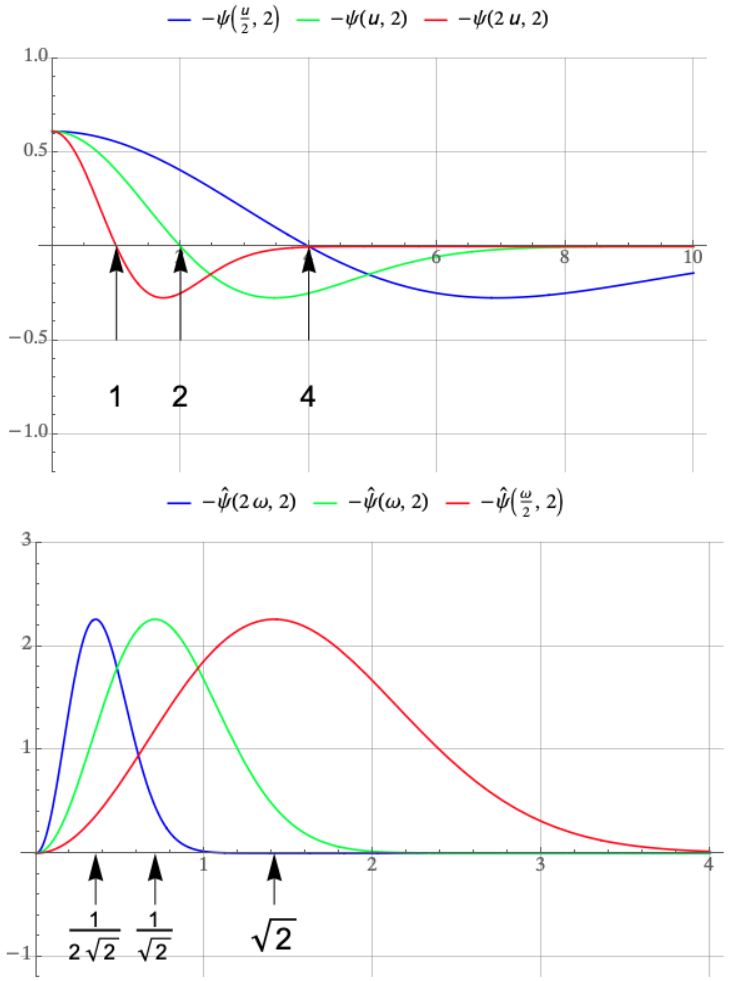

FIGURE 33 – Example of the evolution of $\psi_s(u) \propto \psi(u/s)$ and $\hat{\psi}_s(\omega) \propto \hat{\psi}(s\omega)$ as a function of $s = 2, 1, 1/2$ (blue, green, red curves). The position of the first zero of ψ_s follows a law in s, while the position of the maximum of $\hat{\psi}_s$ follows a law in $1/s$. Illustrated with the Mexican hat wavelet from Figure 32.

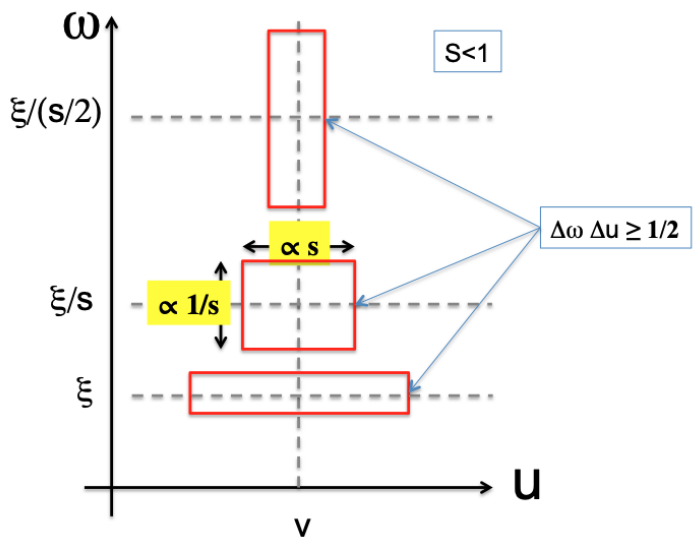

FIGURE 34 – Evolution of the wavelet's support in the time-frequency plane when changing the scale factor.

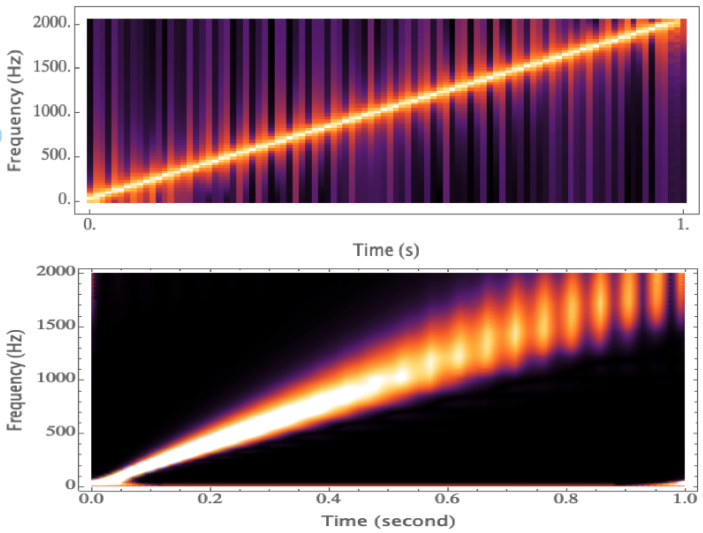

FIGURE 35 – Comparison between windowed analysis with a fixed introspection window size (top) and wavelet analysis (bottom) where the window size evolves according to the scheme in Figure 34. The signal $x(u) = \sin(2000\pi u^2)$ is sampled at a rate of $1/4095$ over $u \in [0, 1]$. For the WT, a Gabor wavelet $1/\pi^{1/4} e^{-u^2/2} e^{i\xi u}$ with $\xi = 6$ was used.

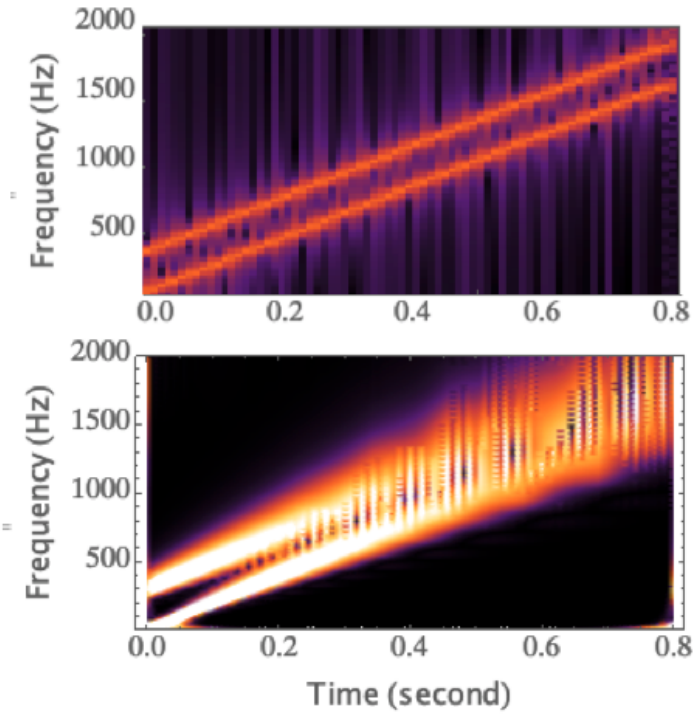

FIGURE 36 – Superposition of 2 signals of the type used in Figure 35: (top) by STFT, (bottom) by WT.

So, the harmonics of the signal shift towards higher frequencies, and as the box size elongates along the frequency axis, there is a confusion of harmonics. **But in return, this lack of frequency resolution provides deformation stability**. Indeed, the STFT is precise, but its instability to small perturbations (deformations) renders this quality useless because it tends to make signals that are very similar appear different. This ultimately harms classification. The WT, which is more deformation stable at the cost of lower frequency resolution at high frequencies, is more suitable. So, **it's not so much the resolution that matters, but the stability**.

7.3 A Naturalistic Digression

Before delving into the technical aspects of the wavelet transform, S. Mallat takes us through a description of the psychophysics of the auditory system, which is the subject of an associated seminar (March 12, 2020, Prof. Shihab Shamma). The diagram of the human ear (Fig. 37) shows the main sound receptors that transmit the pressure wave and transform it into electrical signals in the cochlea. The organ of Corti, located at the center of the cochlea, is divided into approximately 10,000 cells covered with cilia that are immersed in a liquid. Depending on the frequency of the sound signal, it is analyzed by a specific part of the organ of Corti: the lower part analyzes high-pitched sounds with short wavelengths, while the upper part (apex) analyzes low-pitched sounds with long wavelengths. **The modeling of the response of cilia is as follows: they act as a bandpass**

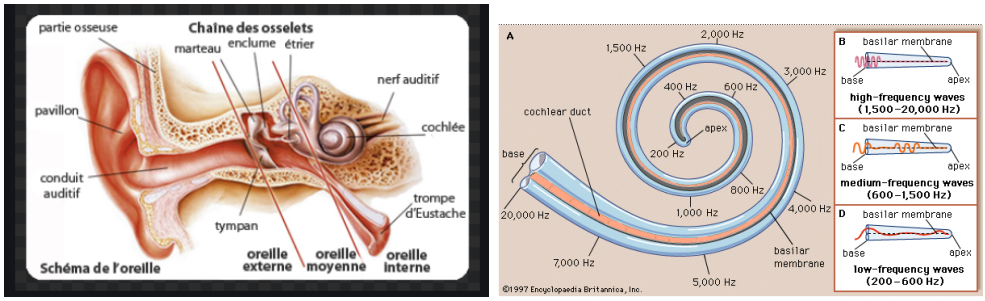

FIGURE 37 – Diagram of the human ear and the cochlea.

filter that closely resembles a wavelet (except at low frequencies where the width is constant). If we unroll the cochlea, it appears as a long tube with a varying cross-section,

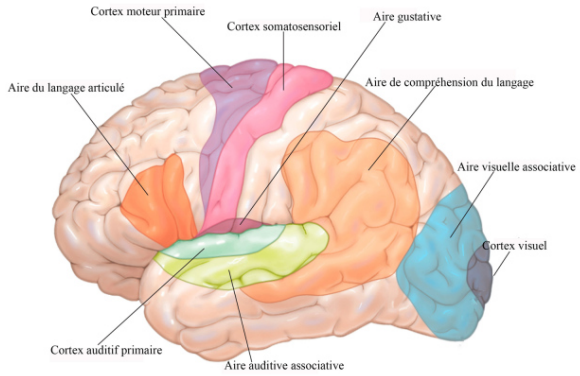

FIGURE 38 – Specialized areas of the cerebral cortex (left).

and the position along this tube actually gives the scale factor of the wavelet. Moreover, the representation is rather **logarithmic in scale** (cf. $\log s$). The question is, why does an organ that has benefited from millions of years of adaptation so closely resemble a wavelet transform? Without a doubt, the localization of transients, as well as stability to small deformations, allows for better recognition/classification.

Once the electrical signal is produced, it is transmitted through the auditory nerve (for simplicity) to the auditory cortex, which consists of two main areas (Fig. 38). In fact, in part A1 of the auditory cortex, the processing, after the filtering $x * \psi_s$, amounts to rectifying using a function ρ, as in a CNN, and then performing two-dimensional filtering that acts both in time and on the scale. Thus, the modeling of the response to a sound signal can be schematically represented as follows:

$$x(t) \xrightarrow{cochlea} x(t) * \psi_s(t) \xrightarrow{cortex(A1)} \rho[\rho[x(t) * \psi_s(t)] * \psi_\alpha(t, s)]$$

In part A2 of the auditory cortex, it becomes much more complicated with kinds of pattern analysis. What is observed is that the deeper one goes into the deep cortex, the coefficients evolve slowly in time, like time invariance over time scales of a second or more.

Referring to Geoffroy Peters' seminar (February 12, 2020), all the representations he

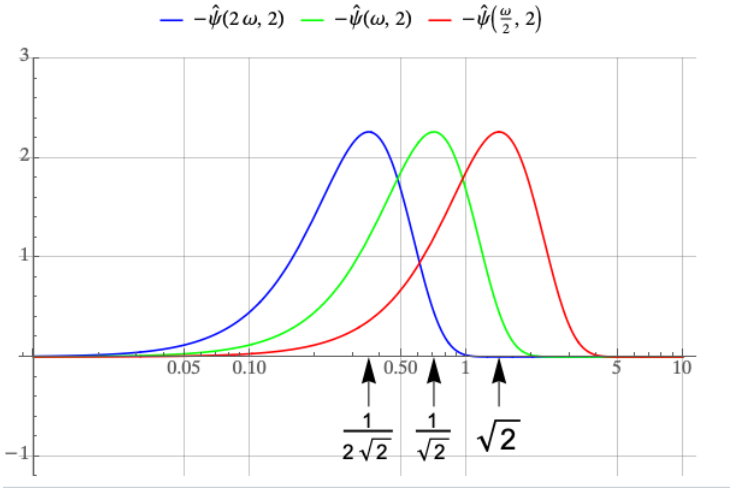

FIGURE 39 – Illustration of the constancy of the wavelet width in frequency, in a logarithmic representation. (See Fig. 33).

presented are wavelet transforms, which he referred to as *"the Q constant transform"*. The Q-value is the width of the wavelet in the $\log \omega$ representation, which remains constant when changing the scale s, as illustrated in Figure 39.

To gain an intuition for the Q-width in the case of the ear, consider the concept of an octave. The division of the frequency band into intervals of constant *relative* width, or $\Delta\omega/\omega = \text{Const}$, generally defines an octave [51]. The width of an octave band is equal to the distance to frequency 0. The so-called "equal temperament" range divides one octave into 12 (half-tone) intervals in a geometric progression that the human ear can distinguish. So, we have better resolution than an octave. In fact, when measuring the width of cochlear filters, one obtains approximately 1/16 of an octave. The wavelets are "gammatone" filters (Fig 40) that have an asymmetric shape in time to capture the non-reversibility of the signal. However, the frequency width of the gammatone is much smaller than its distance to 0 [52].

These audio wavelets are different from those encountered in image processing, but

51. NDJE: Typically, the bandwidth of an octave is defined by a central frequency ω_c, and a frequency ω belongs to the octave if $\log \omega = \log \omega_c \pm \log \sqrt{2}$. Thus, $\Delta \log f = \Delta f/f = \log 2$.

52. NDJE: This remark refers to how one can construct an analytical complex wavelet from a Gaussian of width η (in Fourier); it suffices to translate its spectrum by the same amount towards higher frequencies.

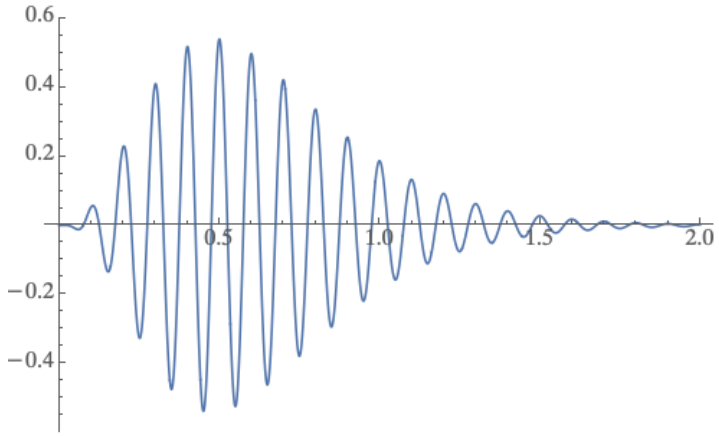

FIGURE 40 – Example of a "gammatone" with a generic expression $at^{n-1}e^{-2\pi bt}\cos(2\pi ft)$.

their mathematical properties are essentially the same. The choice of the wavelet is dictated by the sparsity of the representation.

7.4 Mel-Frequency Cepstrum Coefficients (MFCCs)

In audio analysis, signal processing specialists defined Mel-Frequency Cepstrum Coefficients (MFCCs) in the 1970s-80s. These MFCCs were systematically used in all algorithms until around 2010. From one perspective, MFCC technology remained unchanged for 30-40 years. During this period, Convolutional Neural Networks (CNNs) emerged and became the most performant starting in 2014-15. However, these MFCCs are well understood analytically and provide a good basis for understanding the first layers of CNNs. Let's go through how these MFCCs are obtained:

1. We begin by obtaining $\text{Mel}(x)$ by calculating **the magnitudes of wavelet coefficients** at a scale $s = a^j$, i.e., $|x * \psi_{a^j}|$ [53]. The coefficient a must be chosen so that the filters cover the entire frequency band. Note that $\log s = j \log a$, which means that we uniformly sample the frequency axis because, as a reminder, the central frequency of the introspection box is $\omega_c = \xi/s$. Therefore, $\log \omega_c = \log \xi - j \log a$ with $\log \xi$ and $\log a$ being constants. Next, **we perform a convolution** that shifts over a time

53. One could also take the square magnitude, but as we will see, we later take the logarithm of Mel, so the power is not important.

scale of 2^J using $|x * \psi_{a^j}| * \phi_J(t)$ with $\phi(t)$ being a window of width ~ 1 that we dilate to adjust its duration, cf. $\phi_J(t) = 2^{-J}\phi(2^{-J}t)$ with a width of $\sim 2^J$, typically around 25ms to 50ms. These descriptors will have a very significant impact on **local invariance by translation ϕ_J and stability by deformation with wavelets ψ_{a^j}**.

2. Once these Mel descriptors are obtained, which filter the signal into frequency bands of constant width in logarithmic scale, we will construct **invariants**.

   The first invariant we think of is related to getting rid of a **multiplicative factor of the signal**. If $x(t) \to \alpha x(t)$, then $\text{Mel}(x) \to \alpha \text{Mel}(x)$. To isolate the unknown factor α, we can take the logarithm, which gives:

$$\log(\alpha \text{Mel}(x)) = \log \alpha + \log(\text{Mel}(x)) \tag{105}$$

   Next, we want to eliminate an **amplitude modulation**, which is an extension of the previous case, allowing α to depend on time but to be approximately constant over intervals of duration 2^J. Therefore, if $\alpha(t)$ has **slow variations** compared to those of ψ_{a^j} and ϕ_J, then we can take it out of the convolutions, and thus,

$$\log(\alpha \text{Mel}(x)) = \log \alpha(t) + \log(\text{Mel}(x)) = \log \alpha(t) + \log[|x * \psi_{a^j}| * \phi_J(t)] \tag{106}$$

   There are **two time-dependent components** that need to be separated because they contain information of different natures: the first one ($\log \alpha(t)$) depends only on time, while the second one depends on j and t for a fixed J (cf. temporal resolution) and is related to j. Then, we perform **a transformation along the $\log \omega$ axis to separate the constant part (independent of j) and the part that varies**. This transformation cannot be a Discrete Fourier Transform (DFT) [54] because the "signal" along the $\log \omega$ axis is not periodic, which would have harmful edge effects. Instead, **we perform a Discrete Cosine Transform (DCT)** [55]:

$$DCT \to \left\langle z(j), c_k(j) = \cos\left(\frac{i2\pi k}{K}(j + 1/2)\right)\right\rangle \tag{107}$$

54. NDJE: DFT stands for Discrete Fourier Transform, to distinguish it from FFT, which stands for Fast Fourier Transform, although in practice FFT is often used to refer to DFT.

55. NDJE: There are different variants of DCT, so you should consult the documentation of the libraries used.

where $z(j) = \log|x * \psi_{a^j}| * \phi_J(t)$ for a fixed t. It can be proven that $\{c_k(j)\}_{0\leq k<K}$ define an orthogonal basis. Therefore, we calculate

$$a_k = \sum_j z(j)c_k(j) \tag{108}$$

which extracts information about **variations of the signal along the frequency axis**.

The result, called an MFCC, is a collection of Mel-frequency cepstral coefficients that depend on a time t and a frequency k (note that j relates to the logarithmic frequency scale of the wavelet ψ_{a^j}, and J relates to the smoothing function $\phi_J(t)$ scale):

$$\boxed{MFCC(t,k) = \langle \log \mathrm{Mel}_x(t,j,J), c_k(j)\rangle = DCT_j(\log \mathrm{Mel}_x)(t,k)} \tag{109}$$

The parameter j is a frequency parameter, k is a frequency frequency parameter, and t is sampled with scale 2^J (cf. $t_n = n2^J$). A diagram of these different stages is shown in Figure 20-40. In summary, **every 25ms, we have a vector that, depending on k, gives the variation of the wavelet transform magnitude smoothed over 25ms**.

The value for $k = 0$, the corresponding MFCC, depends on slow frequency modulations. Therefore, we isolate them into a single coefficient. For all other coefficients ($k > 0$), we see the signature of harmonics. Subsequently, these coefficients $MFCC(n2^J, k)$ are either sent directly to a classifier, for example, to perform music genre recognition, or Hidden Markov Models (HMMs) are used for speech recognition to connect the $MFCC(n2^J, k)$ according to n using Gaussian models (GMMs), which are then sent to classifiers. These developments were state-of-the-art until the 2010s.

7.5 Inversion and Stability of Wavelet Transform

We will now discuss why it has been essential to use wavelets and explore the properties of stability when performing recognition tasks. First, we will demonstrate the inversion properties of wavelets, and then we will study their stability under deformation.

In the calculation of an MFCC, we use the wavelet transform ψ_{a^j} with a scale $s = a^j$. However, we also want to capture very low frequencies [56]. To do this, we use a special filter

56. NDJE: Here, the function ϕ serves the sole purpose of covering the low-frequency spectrum. In the

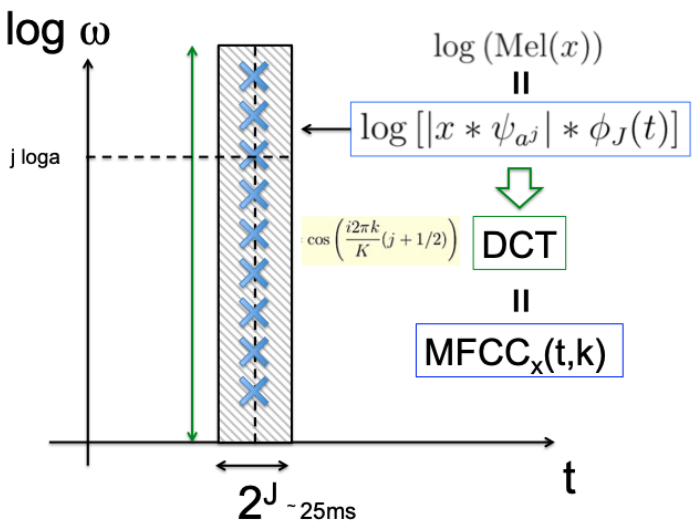

FIGURE 41 – Schematic representation of the various stages involved in obtaining the MFCCs of the x signal. They use the wavelet transform ψ_{a^j} smoothed in the time domain by $\phi_J(t)$, then a DCT along the frequency axis via the cosines $c_k(j)$ (Eq. refeq-2020-MFCdef).

.

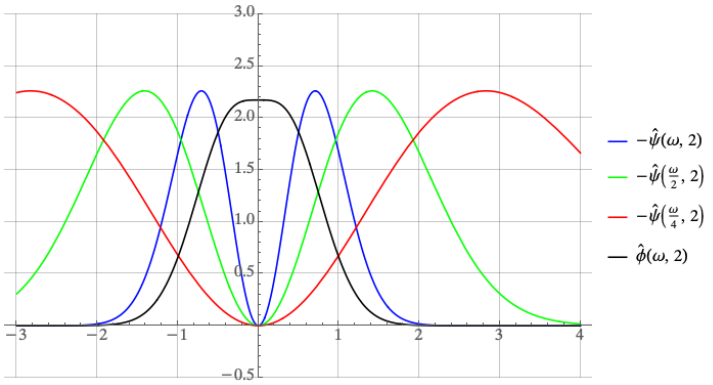

FIGURE 42 – Illustration of the representation of the $\hat{\psi}$ wavelet filters (Eq. 103) and the $\hat{\phi}$ low-pass filter.

$\phi(t)$ that we dilate in time, $\phi_J(t) = 2^{-J}\phi(2^{-J}t)$ (recall: $\hat{\phi_J}(\omega) = \hat{\phi}(2^J\omega)$). Thus, we have a collection of filters for x at low and high frequencies, as follows:

$$\boxed{Wx \equiv \Big((x * \psi_{a^j})(t), (x * \phi_J)(t)\Big)_{a^{-j} \geq 2^{-J}}} \tag{110}$$

and all scales a^j are smaller than 2^J, or $j \leq J(\log 2/\log a) = \alpha J$. In Figure 42, you can see that if $J = 0$ (cf. $\hat{\phi}_0 = \hat{\phi}$), we retain all wavelet filters for which $2^j : j = 0, -1, -2, \ldots$ (with $a = 2$).

The first two questions that come to mind are: 1) can we synthesize $x(t)$ from W_x, and 2) is this transformation stable, meaning $\|Wx\| \sim \|x\|$? We will consider the case where $\psi(t)$ is a **real wavelet**, meaning that $\hat{\psi}(-\omega) = \hat{\psi}^*(\omega)$. Later, we will discuss the case of a **complex and analytical wavelet**, where $\hat{\psi}(\omega)$ has bounded support on $\mathbb{R}^+$ [57].

Theorem 7

In the case of a ***real wavelet****, we need the set of filters* $\left\{\hat{\psi}(a^j\omega)\right\}_{j \leq \alpha J}$ *and* $\hat{\phi}(2^J\omega)$

2018 course and his book, S. Mallat introduced the scaling function, which imposes a specific filter shape because it is related to that of ψ (and vice versa). In the illustrative figures, I have chosen the function ϕ associated with the Mexican hat wavelet.

57. NDJE: S. Mallat did not have time to cover this last case in his oral presentation.

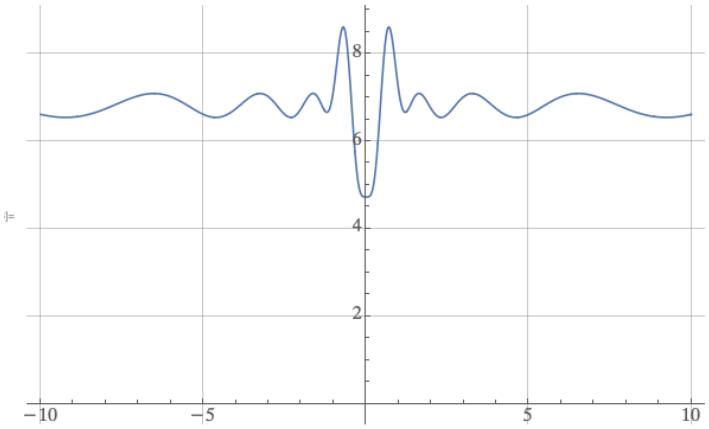

FIGURE 43 – Example of $S(\omega)$ (Eq. 111) for $\hat{\psi}$ (Eq. 103) and the low-pass filter $\hat{\phi}$. The important thing is that $S(\omega)$ does not have gaps, and the upper bound of 1 is conventional.

to satisfy the relation:

$$\exists c < 1 \ s.t. \ \forall \omega, \ 1 - c \leq S(\omega) = \sum_{j \leq \alpha J} |\hat{\psi}(a^j \omega)|^2 + |\hat{\phi}(2^J \omega)|^2 \leq 1 \tag{111}$$

then we define two filters $\widehat{\overline{\psi}}(\omega)$ *and* $\widehat{\overline{\phi}}(\omega)$ *as follows:*

$$\widehat{\overline{\psi}_{a^j}}(\omega) \equiv \frac{\hat{\psi}^*(a^j \omega)}{S(\omega)}, \quad \widehat{\overline{\phi}_J}(\omega) \equiv \frac{\hat{\phi}^*(2^J \omega)}{S(\omega)} \tag{112}$$

such that

$$\boxed{x(t) = \underbrace{\sum\nolimits_j (x * \psi_{a^j} * \overline{\psi}_{a^j})(t)}_{\textit{high frequencies}} + \underbrace{(x * \phi_J * \overline{\phi}_J)(t)}_{\textit{low frequencies}}} \tag{113}$$

and

$$\boxed{(1-c)\|x\|^2 \leq \|Wx\|^2 = \sum\nolimits_j \|x * \psi_{a^j}\|^2 + \|x * \phi_J\|^2 \leq \|x\|^2} \tag{114}$$

The constraint (Eq. 111) essentially states that the spectrum covered by the set of filters should not have gaps. An example with Mexican hat wavelet filters and the associated low-pass filter is shown in Figure 43.

Proof 7. Naturally, with convolution sums, the demonstration is done in the Fourier domain. Let $b(t)$ be the right-hand side (r.h.s) of Equation 113, and take its Fourier transform:

$$\begin{aligned}\hat{b}(\omega) &= \sum_j \hat{x}(\omega)\widehat{\psi_{a^j}}(\omega)\widehat{\overline{\psi}_{a^j}}(\omega) + \hat{x}(\omega)\widehat{\phi_J}(\omega)\widehat{\overline{\phi}_J}(\omega) \\ &= \hat{x}(\omega)\left\{\sum_j \frac{|\hat{\psi}(a^j\omega)|^2}{S(\omega)} + \frac{|\hat{\phi}(2^J\omega)|^2}{S(\omega)}\right\} \\ &= \frac{\hat{x}(\omega)}{S(\omega)}\left\{\sum_j |\hat{\psi}(a^j\omega)|^2 + |\hat{\phi}(2^J\omega)|^2\right\} = \hat{x}(\omega) \end{aligned} \tag{115}$$

For the second relation (known as *energy conservation*), we use Parseval's theorem. Now,

$$\begin{aligned}\|Wx\|^2 &= \frac{1}{2\pi}\left\{\sum_j \int |\hat{x}(\omega)|^2|\hat{\psi}(a^j\omega)|^2 d\omega + \int |\hat{x}(\omega)|^2|\hat{\phi}(2^J\omega)|^2 d\omega\right\} \\ &= \frac{1}{2\pi}\int |\hat{x}(\omega)|^2\left\{\sum_j |\hat{\psi}(a^j\omega)|^2 + |\hat{\phi}(2^J\omega)|^2\right\} d\omega \\ &= \frac{1}{2\pi}\int |\hat{x}(\omega)|^2 S(\omega) d\omega \end{aligned} \tag{116}$$

and using the constraint Eq. 111 on $S(\omega)$, we obtain the result. ■

Therefore, this theorem states that once the filters cover all frequencies, the decomposition is both complete and stable.

Finally, the factor a in image processing will be shown to be equal to 2, while in audio processing, $a = 2^{1/Q}$, where Q is the filter width, typically set to $8, 16, 32$. This means that Q frequencies (half-tones) are placed per octave.

7.5.1 The Case of Analytic Wavelets

NDJE: This is a case that S. Mallat could not cover in his lecture due to time constraints. Here is the additional content he kindly approved.

By definition, a wavelet ψ whose Fourier spectrum is nonzero only for $\omega \leq 0$, i.e.,

$$\hat{\psi}(\omega) = 0 \quad \text{if} \quad \omega < 0 \tag{117}$$

is called an *analytic and complex* function. The condition Eq. 111 is valid only for $\omega \geq 0$. If we denote $b(t)$ as the right-hand side of Eq. 113, then $\hat{x}(\omega) = \hat{b}(\omega)$ for $\omega \geq 0$, and $\hat{b}(\omega) = 0$ for $\omega < 0$. For **a real signal** for $\omega < 0$, we know that $\hat{x}(-\omega) = \hat{x}^*(\omega)$, so $\hat{x}(\omega) = \hat{b}^*(-\omega)$ for $\omega < 0$. Therefore, to reconstruct $x(t)$ via the Inverse Fourier Transform, we have

$$\begin{aligned} x(t) &= \frac{1}{2\pi}\left\{\int_0^{+\infty} \hat{b}(\omega)e^{i\omega t}d\omega + \int_{-\infty}^{0} \hat{b}^*(-\omega)e^{i\omega t}d\omega\right\} \\ &= \frac{1}{2\pi}\left\{\int_{-\infty}^{+\infty} \hat{b}(\omega)e^{i\omega t}d\omega + \int_{-\infty}^{+\infty} \hat{b}^*(\omega)e^{-i\omega t}d\omega\right\} \\ &= b(t) + b^*(t) = 2\,\mathrm{Re}[b(t)] \end{aligned} \tag{118}$$

Therefore, signal synthesis for a **real signal**, in the case of an analytic wavelet, is done using the following relation:

$$\boxed{x(t) = 2\,\mathrm{Re}\left[\sum\nolimits_j (x * \psi_{a^j} * \overline{\psi}_{a^j})(t) + (x * \phi_J * \overline{\phi}_J)(t)\right]} \tag{119}$$

Regarding the energy conservation, by the same calculation as in Eq. 116, we obtain a relation valid only for positive frequencies:

$$\|Wx\|^2 = \frac{1}{2\pi}\int_0^{+\infty} |\hat{x}(\omega)|^2 S(\omega)d\omega \leq \frac{1}{2\pi}\int_0^{+\infty} |\hat{x}(\omega)|^2 d\omega \tag{120}$$

Now, for a **real signal**

$$\|x\|^2 = \frac{2}{2\pi}\int_0^{+\infty} |\hat{x}(\omega)|^2 d\omega \tag{121}$$

so, we obtain a double constraint in the case of an analytic wavelet and a real signal:

$$\boxed{(1-c)\frac{\|x\|^2}{2} \leq \|Wx\|^2 \leq \frac{\|x\|^2}{2}} \tag{122}$$

7.6 The Representation $\Phi(x)$

Let's keep in mind that we want to build a representation $\Phi(x)$ that addresses the curse of dimensionality by using the $3S$ properties (separability, symmetry, sparsity). So, starting from the MFCCs, what should we do? To focus on the essentials, we will omit the "log" as well as the DCT. Instead, we will demonstrate that if:

$$\boxed{\Phi(x) \equiv \{\rho(x * \psi_{a^j}) * \phi_J(t)\}_j} \tag{123}$$

where ρ is either the absolute value or a ReLU (or any other contracting operator), then we have a representation that:

1. Does not amplify noise (cf. *contractance property*).
2. Is stable under deformations.

Theorem 8

Let's revisit the assumptions of Theorem 7. Then it follows that Φ *is* ***contracting****, which means that:*

$$\forall x, x' \quad \|\Phi(x) - \Phi(x')\| \leq \|x - x'\| \tag{124}$$

and, since $\Phi(0) = 0$*, it also implies:*

$$\|\Phi(x)\| \leq \|x\| \tag{125}$$

(Note: a constant can be placed on the first inequality).

In classification, the first property is crucial, especially in high dimensions where points are far apart from each other. Therefore, we aim to obtain a representation that brings them as close as possible, especially for points belonging to the same class.

Proof 8. The proof begins by noting that $\Phi(x)$ involves three operations:

1. The wavelet transformation, which is linear.
2. Taking the absolute value, which can be represented as $\rho(z) = |z|$ (the absolute value of "z" if a is complex). It's worth noting that this could also be a rectifier function (ReLU) since it's also a contracting operator.
3. Smoothing by Φ_J, which is an average of the form $A_J(z) = z * \Phi_J$.

Therefore, we can express $\Phi(x)$ as:

$$\Phi(x) = A_J[\rho(Wx)] \tag{126}$$

Based on Theorem 7, we know that W is a linear operator for which:

$$\|Wx\| \leq \|x\| \Rightarrow \|W(x - x')\| = \|Wx - Wx'\| \leq \|x - x'\|$$

Hence, W is a contracting operator [58].

Next, for the operator ρ (absolute value or ReLU), we know that the absolute value satisfies the reversed triangular inequality (this is also true for ReLU):

$$|\rho(z) - \rho(z')| \leq |z - z'|$$

So, ρ is a contracting operator as well. Now, let's focus on A_J.

Lemma: Consider the linear operator A_J defined as $A_J z = z * \Phi_J$. To demonstrate that it is contracting, it suffices to show that:

$$\|A_J z\| \leq \|z\|$$

Now, since A_J is a convolution operator, we can move to the Fourier domain. By using the second inequality (Eq. 114) from Theorem 7, we can conclude that:

$$|z * \Phi_J||^2 \leq ||z||^2$$

Therefore, A_J is indeed a contracting operator.

In the end, the composition of operators $A_J \rho W$ is contracting, and thus, the theorem is proven. ■

Note that **the contraction** will not be strong at the level of W since it preserves the norm, but it **will be strong at the level of the operator ρ (e.g., ReLU) and in the smoothing operation (*pooling*, in neural networks, we use "max pooling" or "average pooling")**.

58. Note that for a nonlinear operator, we can have the left property on the norm of $O_{NL}(x)$ without having the property of "contractance"

However, in this contraction operation, it is possible to bring closer x values that do not belong to the same class, so mechanisms should be in place to prevent this. Nevertheless, these operations can be cascaded to maximize the contraction. Let's now address the result of stability under deformation, which is somewhat novel. In harmonic analysis, the focus has traditionally been on linear operators. Therefore, results regarding nonlinear operators are revisited to understand MFCCs and CNNs.

Theorem 9

The goal is to demonstrate Lipschitz continuity under small deformations, which means that[a]*:*

$$g.x(u) = x(u - \tau(u)), \text{ with } ||\nabla\tau||_\infty < 1/2$$

Therefore, if we restrict ourselves to signals $x \in L^2(\Omega)$ on a compact support Ω[b]*, which can be very irregular, then:*

$$\exists C > 0 \text{ s.t. } \forall \tau \in C^2 \ ||\nabla\tau||_\infty < 1/2, \forall x \in L^2(\Omega),$$
$$\|\Phi(x) - \Phi(g.x)\| \leq C\|x\| \left(\underbrace{||\nabla\tau||_\infty + \|H\tau\|_\infty}_{|g|_G} + \underbrace{\|\tau\|_\infty 2^{-J}}_{\text{translation}} \right) \quad (127)$$

Here, $\|H\tau\|_\infty$ represents the infinity norm of the Hessian (2nd derivatives), and $\|\tau\|_\infty = \sup_u |\tau(u)|$. Note that for the norm of $\Phi(x)$, being a vector, we have:

$$\|\Phi(x)\|^2 = \sum\nolimits_j \|\rho(x * \psi_{a^j}) * \phi_J(t)\|^2$$

a. The condition applies to the norm of the Jacobian matrix in 2D; in 1D, it simply requires that $|\tau'(u)| < 1$. Although one could go up to 1, a bound of 1/2 is set to eliminate any instability around 1.

b. It can be extended to the entire $\mathbb{R}$, but this involves technical complications

What this theorem tells us is that **as $J \to \infty$, the translation component will vanish, and the bound will depend only on the deformation size** (this can be related to the reasoning in Eq. 43). This is a significant result because it applies to the entire representation. If the deformation is small, the representation linearizes these deformations (stability), allowing us to select and learn invariants for small deformations by learning

linear operators, something a linear classifier applied to $\Phi(x)$ can achieve. The challenge here is to approximate the sources of variability that we want to eliminate by linearization. However, if the deformation is too large, this linearization is no longer valid.

NDJE: The proof is deferred to the next session in Sec. 8.5.

8. Lecture 11 Mar.

During this session, we will first revisit the descriptors computed from wavelet transforms, which are covariant and multi-scale translation-invariant. We will explore how they behave in the context of image processing and discuss their connection to neurophysiology. In the second part, we will demonstrate the covariance and stability under diffeomorphism of these descriptors, which allows us to linearize them for classification (an important theme). The third and final part will be dedicated to the connection with neural networks, showing how these descriptors can be implemented using cascades of filters (convolution/pooling and ReLU). This will be an initial way to approach the architecture of CNNs, with a highly simplified architecture. The following week will be devoted to applications of this kind of architecture.

8.1 Reminder of MFCCs in Audio

We consider a family of wavelets ψ_j defined as follows:

$$\psi_j(u) = \frac{1}{a^j}\psi\left(\frac{u}{a^j}\right) \Rightarrow \hat{\psi}_j(\omega) = \hat{\psi}(a^j\omega) \tag{128}$$

The wavelet ψ is chosen to create a bandpass filter, which is dilated by the scale factor a^j (see Figure 33 for example). In audio, the factor a is smaller than 2, specifically $a = 2^{1/Q}$ where Q is the width of the filter in logarithmic scale (Figure 39). To cover an octave (a factor of 2), we need Q wavelets. Therefore, Q determines the frequency precision; the larger Q is, the better the resolution. Typically, $Q \sim 16$, which is slightly larger than the number of half-tones in music and is also the precision of the filters in the cochlea.

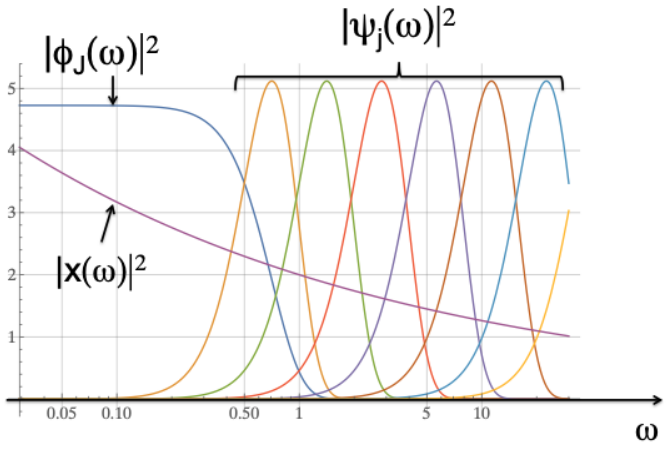

FIGURE 44 – Illustration of the representation of the wavelet filters $\hat{\psi}$ (Eq. 103) and the low-pass filter $\hat{\phi}$ in logarithmic frequency scale. This allows capturing different parts of the signal x that are introspected by the various filters.

The wavelet transform is defined as a collection of convolutions, both with the wavelet family ψ_j to cover high frequencies and with a complementary filter $\phi_J = 2^{-J}\phi(2^{-J}u)$ that handles low frequencies (see Section 7.5):

$$Wx = \left(\{x * \psi_j\}_j, x * \phi_J\right) \Rightarrow \widehat{Wx}(\omega) = \left(\left\{\hat{x}(\omega)\hat{\psi}(a^j\omega)\right\}_j, \hat{x}(\omega)\hat{\phi}(2^J\omega)\right) \tag{129}$$

This amounts to analyzing the signal in all frequency bands, as illustrated in Figure 44. It is important to note that ψ naturally oscillates significantly, which may not be the case for ϕ.

We have seen that this kind of representation is **complete** and **stable** provided that we cover the entire **frequency axis without gaps**. This condition is expressed by the relation:

$$0 < 1 - c \leq S(\omega) \equiv \sum_j |\hat{\psi}(a^j\omega)|^2 + \hat{\phi}(2^J\omega) \leq 1 \tag{130}$$

The crucial part is that the lower bound is **strictly greater than 0** (the upper bound is conventionally normalized to 1). This has two fundamental consequences:

— **Completeness**: We can define a wavelet reconstruction $\overline{\psi}$ for high frequencies and

its counterpart for low frequencies $\overline{\phi}$ defined by their Fourier transforms as:

$$\widehat{\overline{\psi}_{a^j}}(\omega) \equiv \frac{\hat{\psi}^*(a^j\omega)}{S(\omega)}, \quad \widehat{\overline{\phi}_J}(\omega) \equiv \frac{\hat{\phi}^*(2^J\omega)}{S(\omega)} \tag{131}$$

Thus,

$$\boxed{x(t) = \underbrace{\sum\nolimits_j (x * \psi_{a^j} * \overline{\psi}_{a^j})(t)}_{\text{high frequencies}} + \underbrace{(x * \phi_J * \overline{\phi}_J)(t)}_{\text{low frequencies}}} \tag{132}$$

— **Contracting Operator**: The wavelet transform satisfies the double inequality:

$$\boxed{(1-c)\|x\|^2 \leq \|Wx\|^2 = \sum\nolimits_j \|x * \psi_j\|^2 + \|x * \phi_J\|^2 \leq \|x\|^2} \tag{133}$$

From the wavelet transform Wx, we have defined first-order Mel frequency descriptors (MFC) as follows (we omit the low-frequency part for brevity):

$$\boxed{\Phi(x) = \{\rho(x * \psi_{a^j}) * \phi_J(t)\}_j} \tag{134}$$

Here, ρ is **a non-linearity** (absolute value, ReLU). The idea of studying these descriptors, which were surpassed in the 2010s, is that we have **all the basic ingredients of neural networks**:

1. We choose the wavelet ψ to have a **sparse representation** of the signal during the operation $x * \psi_{a^j}$ (Wx), which is also a **covariant** operation [59] under translation.
2. We perform a **non-linear rectification** by ρ, which is also a **covariant** operator.
3. The third operation is given by averaging (**pooling**) ϕ_J (A_J), which is an **invariant** operation.

These operations allow us to write the cascade of operators to compute $\Phi(x)$ as follows:

$$\boxed{\Phi(x) = (A_J \rho W)(x)} \tag{135}$$

Non-linearity is essential! Imagine removing the action of ρ; then we average the wavelet coefficients, resulting in 0. In fact, in Fourier space:

$$x * \psi_{a^j} * \phi_J(t) \xrightarrow{T.F} \hat{x}(\omega)\hat{\psi}(a^j\omega)\hat{\phi}(2^J\omega) \tag{136}$$

59. Note: Covariant and equivariant concepts are discussed in Section 5.7.

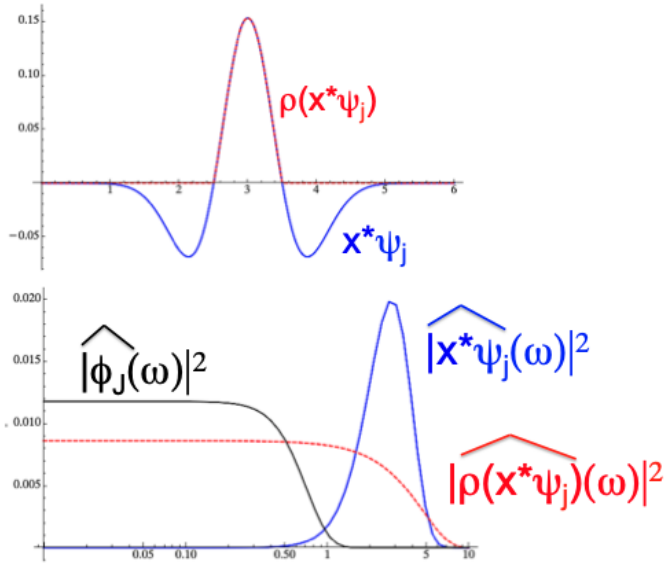

FIGURE 45 – Illustration of the effect of a ReLU on the convolution $x * \psi_{a^j}$. At the top, the result in real space, and at the bottom, what happens in Fourier space. With ReLU, the low-frequency component produced with the low-pass filter ϕ_J gives a non-zero contribution (here, for simplicity, we took $x(u) = \delta(u - u_0)$, so the convolution yields the translated wavelet).

Now, $\hat{\phi}(2^J\omega)$ is a low-frequency filter, while $\hat{\psi}(a^j\omega)$ are high-pass filters at high frequencies. Their product is almost zero. However, when we apply ReLU (for example), it retains only the positive part, ensuring that there is a non-zero overlap (Figure 45). It is the non-linearity that allows us to have **new invariants** beyond just the mean (recall: the only linear operator invariant under the action of a group is the mean). Here, the mean is applied after non-linearity, giving rise to new properties:

— $\Phi(x)$ is **contracting** (see Theorem 8). This is because each of the three operators W, ρ, and A_J is contracting.

— $\Phi(x)$ is **stable under deformation**, which we will see in a later section.

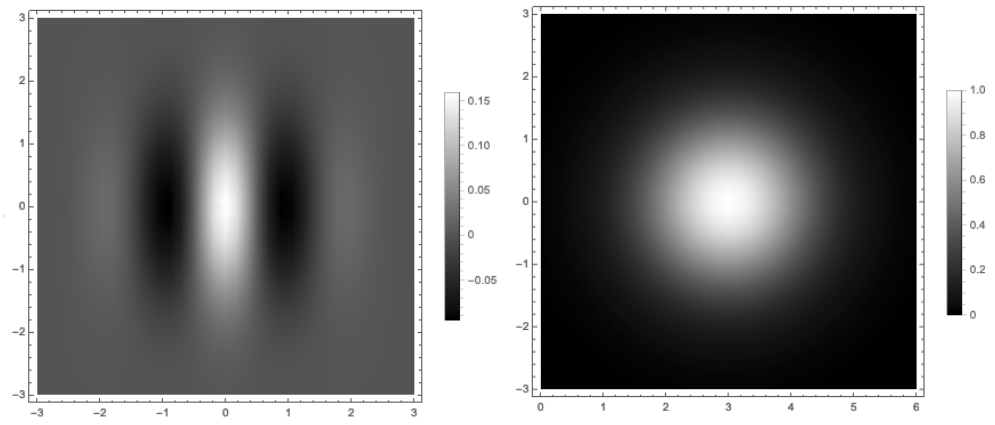

FIGURE 46 – Example of a two-dimensional wavelet $\mathcal{N}_{(0,1)}(x,y)e^{i3x}$: on the left, the real part, and on the right, the Fourier transform (note the shifted Gaussian along the horizontal axis by 3 units).

8.2 Descriptors for Images

Before discussing the stability of $\Phi(x)$, let's see how these descriptors are adapted for images. In this case, the wavelets are two-dimensional objects. First, as in the case of audio, we define ψ as follows:

$$\psi(u) = g(u)e^{i\xi.u} \xrightarrow{T.F} \hat{\psi}(\omega) = \hat{g}(\omega - \xi) \tag{137}$$

Here, g is a Gaussian (or another regular window function), and ξ is a fixed "frequency". See an example in Figure 46.

So, we want an operator W (wavelet transform) that **sparisfies the signal**, meaning it can capture only the relevant structures (such as contours with their orientations), and we also want it to be **complete**, which means it should be able to cover the entire "frequency" domain. To achieve this, we introduce a **rotation of wavelets**:

$$\psi_\theta(u) \equiv \psi(r_{-\theta}.u) \tag{138}$$

Next, following the principle of 1D wavelets, we will dilate/compress the scale to have a complete set of filters: small ones for small structures and large ones for large

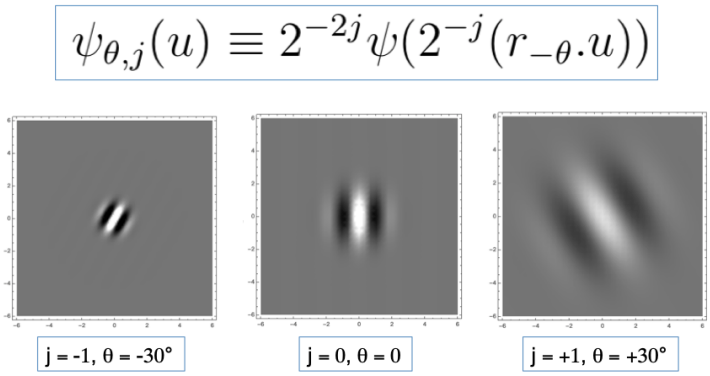

FIGURE 47 – Examples of rotated and scaled wavelets based on the wavelet from Figure 46.

structures. Thus, we define rotated and scaled wavelets as follows [60]:

$$\boxed{\psi_{\theta,j}(u) \equiv 2^{-2j}\psi(2^{-j}(r_{-\theta}.u))} \tag{139}$$

Note that in this case, we have one wavelet per octave, and some examples are presented in Figure 47.

The natural question is how to choose the angles? First, we need to realize that the Fourier transform of $\psi_{\theta,j}(u)$ is given by [61]:

$$\widehat{\psi_\theta}(\omega) = \hat{\psi}(r_\theta.\omega) \tag{140}$$

Therefore, the idea of orientations and covering the ring (Fig. 48) produced by successive rotations leads to the choice of K necessary rotations. During dilation:

$$\boxed{\widehat{\psi_{\theta,j}}(\omega) = \hat{\psi}(2^j r_\theta.\omega)} \tag{141}$$

See Figure 48 for an illustration of the effect of rotation and dilation in Fourier space. Thus, we can cover the entire Fourier plane by choosing the value of K appropriately. We

60. Note the scaling dimension n; we would have $1/s^n\psi(u/s)$.
61. In 2D, the Fourier transform involves $u^T\omega$, and r_θ is an orthogonal matrix $r_\theta^T = r_\theta$, $r_{-\theta}^{-1} = r_\theta$.

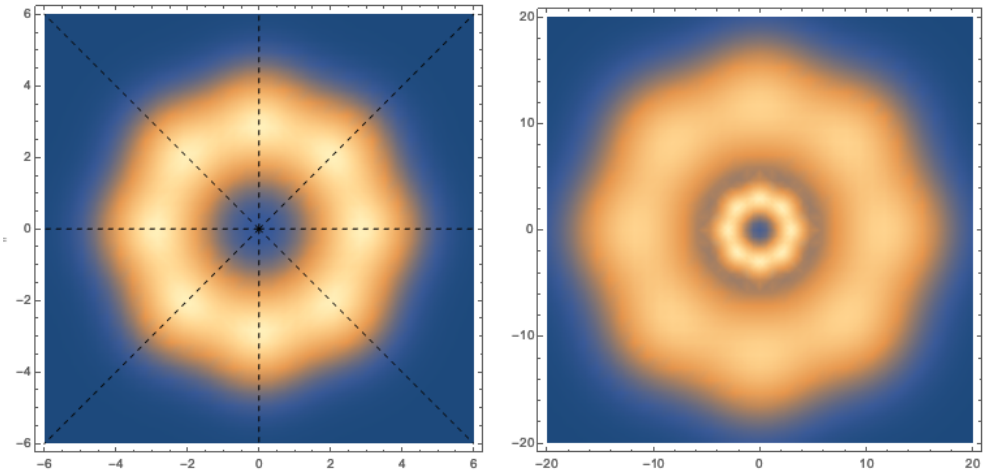

FIGURE 48 – Example of results in the Fourier plane, wavelet rotations from figure 46 in 45° steps (left). On the right, we add the wavelet TF with $j = -2$ (note the change in axis scales)

.

then define:

$$\boxed{\psi_{j,k}(u) = 2^{-j}\psi(2^{-2j}r_{-\theta_k}.u) \qquad \theta_k = 2\pi\frac{k}{K}} \tag{142}$$

Next, we define the wavelet transform following the scheme developed in 1D:

$$\boxed{Wx = (x * \psi_{j,k}, x * \phi_J)_{(j\geq J, 1\leq k\leq K)}} \tag{143}$$

Here, ϕ_J is the low-frequency filter, this time in 2D, defined by the Gaussian g used to define ψ, as follows:

$$\phi_J(x) = 2^{-2J}g(2^{-J}x) \tag{144}$$

So, we have defined a wavelet transform that explicitly uses the group of rotations in addition to dilation. The question that arises now is whether W is a *complete and stable* operator. The answer is yes, as long as we **cover the Fourier plane** without leaving any gaps. This is expressed as:

$$1 - c \leq \sum_{k=1}^{K}\sum_{j=J}^{\infty} |\widehat{\psi_k}(2^j\omega)|^2 + |\hat{\phi}(2^J\omega)|^2 \leq 1 \tag{145}$$

As in 1D, we deduce that W is a contracting operator:

$$(1-c)||x||^2 \leq ||Wx||^2 = \sum_{k=1}^{K} \sum_{j=J}^{\infty} ||x * \psi_{j,k}||^2 + ||x * \phi_J||^2 \leq ||x||^2 \tag{146}$$

Finally, **if we identify x as the input image, the action of W is to produce a collection of "channels" indexed by j and k for the $\psi_{j,k}$, in addition to the result of the low-pass filter, just like in a neural network**.

Next, we can construct the equivalent of MFC, i.e., a descriptor (representation) $\Phi(x)$ as follows:

$$\boxed{\Phi(x) = \rho(x * \psi_{j,k}) * \phi_J = A_J \rho W x} \tag{147}$$

This has the same properties:

1. W produces **sparsity** and is **covariant/equivariant**.
2. ρ (ReLU) produces a **covariant/equivariant** operator that is not a simple average.
3. The **pooling** (averaging) ϕ_J plays the role of creating an invariant that eliminates what is not necessary for classification.

In a neural network, we cascade these kinds of operations, and in addition, we learn the W instead of using fixed wavelets.

8.3 Some Examples

NDJE: In the following part, S. Mallat presents slides to illustrate what he has just discussed about the 2D case. I'll provide the key points he emphasizes.

Fast algorithms for 2D wavelet transforms work by cascading banks of filters (see Lecture 2018, Sec. 6.4). Each wavelet $\psi_{j,k}$ is convolved with the original image, so gradually, it is translated in 2D, and the result is non-zero only at boundaries/contours. Indeed, by definition, the integral of ψ is zero, which means that for constant regions within the support of ψ, the result is zero. Rotations allow us to detect boundaries in all directions. Thus, wavelets reveal the presence of local variations (e.g., contours). On a smaller scale, we subsample the original image, then reapply the filters, and so on...

This is the principle behind, for example, JPEG2000 compression from the 1990s, but we see these filter/subsample cascades reappear in neural networks. However, we have introduced the rectifier ρ, which makes the coefficients positive and adds power to low frequencies, interacting with the filter ϕ_J (as in 1D). Until the 2010s (cf. 2004-12), descriptors like $\Phi(x)$ were constructed "by hand" (e.g., DAISY [62]), i.e., without using learning, and they formed the basis for all image processing algorithms. So, we see that whether in audio or imaging, although it was done independently, the same type of descriptor was developed, and it is very effective.

The question is why? We have seen that it was useful to make the signal sparse and use symmetries to linearize it, to use non-linearity to obtain an invariant that is not a simple average, which would lose all the structure of the signal, and finally, pooling to reduce dimensionality. However, the point that remains to be addressed is **invariance to deformations**. However, before addressing this important point, let's take a detour through neurophysiology.

8.4 Connection with Neurophysiology

The connection with audio was discussed briefly with the rapid description of the cochlea, and you can refer to Shihab Shamma's seminar for more details. In the field of imaging, in the human brain, there are two specialized areas for processing (Fig. 38 and 49) located at the back of the skull. These areas are known as the "visual cortex", called V1, and the "associative visual area", known as V2. Interestingly, V1 contains **neurons sensitive to orientation** (Fig. 49) [63]. This was identified, among others, in the 1960s by **David Hubel** and **Torsten Wiesel** (Nobel 81), who shared the Nobel Prize with **Roger W. Sperry**, who, in turn, discovered the functional specialization of the two brain hemispheres.

Hubel and Wiesel modeled neurons as simple linear filters and measured the impulse response of neurons, namely, h in the operation $x * h$. They found **wavelets** (Gabor wavelets) with specific orientations! Then, they noticed that there were **more complex cells**

62. NDJE: While browsing the web, I found this application that describes all descriptors: https://tel.archives-ouvertes.fr/tel-01611384/document

63. NDJE: Source Michael C. Crair, Edward S. Ruthazer, Deda C. Gillespie, and Michael P. Stryker, "Ocular Dominance Peaks at Pinwheel Centre Singularities of the Orientation Map in Cat Visual Cortex", Journal of Neurophysiology, vol. 77, 1997, pp. 3381-3385.

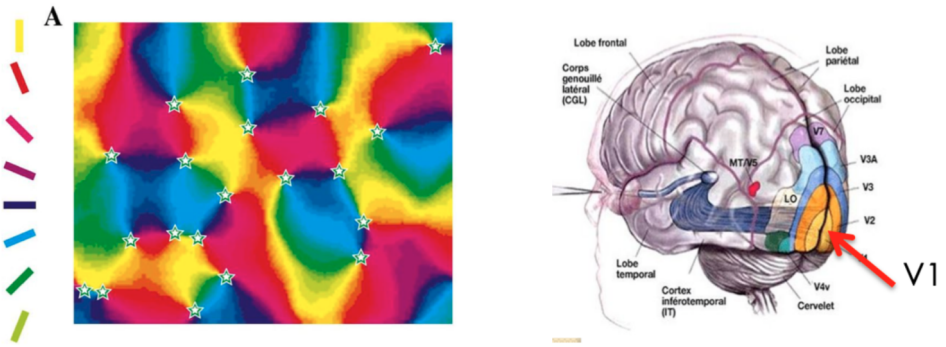

FIGURE 49 – On the left: a false-color image showing that neurons in the V1 area of the brain (right image) are sensitive to the orientation of basic patterns: horizontal bar, vertical bar, left oblique, and right oblique.

in the V2 area: 1) they are very **non-linear**, 2) they cover much **larger receptive fields**, and 3) most importantly, they are **invariant to transformations**, including translations but also much more complicated transformations. Then, in V4 and IT (Fig. 49 right), we have networks of cells **sensitive to contours, more complex shapes, and even objects, including faces** (the 2005 English experiment). This gave rise, in a humorous way, to the "grandmother neuron theory". According to this theory, it is a few neurons that allow the recognition of a well-known face. This theory refers to the computationalist theory in the philosophy of the mind, originally derived from Thomas Hobbes, which responded to Chomskyan formalism in language [64].

The question is, how did we go from "orientation" information in V1 to very complex and transformation-invariant representations in IT? There are deep neural network models for this, but the question remains the same: how do we transition from wavelet coefficients (V1) to invariant descriptors $\Phi(x)$ (IT).

8.5 Stability under Deformations

Behind the stability of $\Phi(x)$, we will discover the property of equivariance (covariance). Indeed, we will show that a small deformation results in a small translation, but

64. NDJE: I've reworded it according to the research I did for a course for Philosophy of Nature students.

this translation occurs both in space and scale. And once again, the issue is the same whether you're working with audio, imagery, or even quantum chemistry: sparsity, invariance concerning symmetry groups of the problem that allows eliminating degrees of freedom, deformations play a role in scale separation with wavelets, and then applying non-linearity and pooling.

Theorem 10

We aim to prove Lipschitz continuity under small deformations, meaning that [a]

For an image x, and a small deformation g applied to it:

$$g.x(u) = x(u - \tau(u)), \text{ with } ||\nabla\tau||_\infty < 1/2$$

If we restrict ourselves to signals $x \in L^2(\Omega)$ on a compact support Ω [b]*, which can be very irregular, then:*

$$\exists C > 0 \text{ s.t. } \forall \tau \in C^2 \ ||\nabla\tau||_\infty < 1/2, \forall x \in L^2(\Omega),$$
$$\text{then} \qquad \|\Phi(x) - \Phi(g.x)\| \leq C\|x\| \underbrace{\left(\underbrace{||\nabla\tau||_\infty + \|H\tau\|_\infty}_{|g|_G} + \underbrace{\|\tau\|_\infty 2^{-J}}_{\text{translation}} \right)}_{\text{deformation size}} \tag{148}$$

with $\|H\tau\|_\infty$ as the infinity norm of the Hessian (i.e., second derivatives), and $\|\tau\|_\infty = \sup_u |\tau(u)|$. Note that since $\Phi(x)$ is a vector, we have:

$$\|\Phi(x)\|^2 = \sum_j \|\rho(x * \psi_j) * \phi_J(t)\|^2$$

a. The condition concerns the norm of the Jacobian matrix in 2D; in 1D, it simply means that $|\tau'(u)| < 1$. Although we could go up to 1, to remove any instability around 1, we set the bound to 1/2. The crucial point is to ensure the invertibility of the diffeomorphism.

b. We can extend it to the entire $\mathbb{R}$, but that introduces technical complications

Proof 10. We will outline the structure of the proof. What we want to constrain is

$\|\Phi(x) - \Phi(g.x)\|$, and we have (assuming that ϕ_J is always positive, e.g., a Gaussian):

$$\begin{aligned}\|\Phi(x) - \Phi(g.x)\| &= \|\rho(W(x)) * \phi_J - \rho(W(g.x)) * \phi_J\| \\ &= \|(\rho(W(x)) - \rho(W(g.x)) * \phi_J\| && (* \text{ is a linear operator}) \\ &\leq \|(W(x) - W(g.x)) * \phi_J\| && (\rho \text{ is contractive, and } \phi_J \geq 0)\end{aligned} \tag{149}$$

We define the wavelet transform as follows: $Wx(u, \log s) = x * \psi_s(u)$, where $s = a^j$. So, the question is whether

$$g.Wx(u, \log s) = Wx(u - \tau(u), \log s) \overset{?}{\sim} W(g.x) \tag{150}$$

Equality would mean that W and g commute. It's not exactly the case, but consider the following lemma:

Lemma 1

$$\begin{aligned}&\exists C > 0 \text{ s.t. } \forall \tau \in C^2 \ ||\nabla \tau||_\infty < 1/2, \forall x \in L^2(\Omega), \\ &\qquad \|W(g.x) - g.W(x)\| \leq C\|x\| \left(||\nabla \tau||_\infty + \|H\tau\|_\infty\right)\end{aligned} \tag{151}$$

This is essentially the computation of a commutator $[Wg - gW]$; if W is equivariant under the action of g, then the result is zero. In fact, the bound is governed by the size of the perturbation (i.e., $|g|_G$).

If we accept this lemma, the proof of Theorem 10 becomes clear. Let's go back to

the calculation in Eq. 149; we have:

$$\begin{aligned}
&\|(W(x) - W(g.x)) * \phi_J\| \\
&\quad = \|(W(x) - g.W(x)) * \phi_J + (g.W(x) - W(g.x)) * \phi_J\| \\
&\quad \leq \underbrace{\|(W(x) - g.W(x)) * \phi_J\|}_{[Wg-gW]=0} + \underbrace{\|(g.W(x) - W(g.x)) * \phi_J\|}_{[Wg-gW]\neq 0} \\
&\quad \leq \|(W(x) - g.W(x)) * \phi_J\| + \|(g.W(x) - W(g.x))\| \qquad (\phi_J \text{ is contractive}) \\
&\quad \leq \|(W(x) - g.W(x)) * \phi_J\| + C\|x\| \, (||\nabla\tau||_\infty + \|H\tau\|_\infty) \qquad (\text{Lemma } 1)
\end{aligned} \tag{152}$$

The part that remains to be addressed concerns what's left after averaging by ϕ_J of the difference between a wavelet transform ($W(x)$) and its transformation by the deformation ($g.W(x)$). We'll see that what remains is the global translation. Consider the following lemma:

Lemma 2 *For a signal z,*

$$\boxed{\exists C > 0 \text{ s.t. } \|(z - g.z) * \phi_J\| \leq C' 2^{-J} \|\tau\|_\infty \|z\|} \tag{153}$$

By assuming this lemma, and noting that W is contractive, we have:

$$\|(W(x) - W(g.x)) * \phi_J\| \leq C' 2^{-J} \|\tau\|_\infty \|x\| + C\|x\| \, (||\nabla\tau||_\infty + \|H\tau\|_\infty) \tag{154}$$

This gives us the theorem if we take the maximum between C and C'.

To summarize the steps of the proof, it shows that **a deformation commutes *almost* with the wavelet transform**. The error comes from both the size of the deformation and the result of translation after averaging, which is nearly zero with an error term depending on the size of the translation compared to the averaging size. Certainly, as the averaging size by ϕ_J tends to infinity (i.e., $J \to \infty$), this term disappears.

Therefore, we need to prove the two lemmas (1 and 2), with the first one being challenging (but interesting). Complete proofs can be found in the article on the website [65].

65. Here is the link: https://www.di.ens.fr/~mallat/College/TPAMI-Mallat-Bruna-Scat-CNN.pdf

S. Mallat suggests not going through the complete proof of the article as it's quite lengthy, but he proposes explaining the phenomenon of translation in scale-space using a Taylor expansion.

Property: If we take the wavelet coefficient of a deformed signal at a position u and a scale $\log s$, then

$$W(g.x)(v, \log s) \simeq Wx(v - \tau(v), \log s - \tau'(v)) \tag{155}$$

meaning **the deformation induces a translation both in spatial and logarithmic scale domains**. Note that the translation along the $\log s$ axis (changing the index because $\log s = i \times \log a$) can be absorbed in later convolution steps. So, let's see where this comes from:

$$W(g.x)(v, \log s) = (g.x) * \psi_s(v) = \int x(u - \tau(u)) \frac{1}{s} \psi \left(\frac{v - u}{s} \right) du \tag{156}$$

Next, we make the change of variables $u' = u - \tau(u)$ and take into account that the wavelet $\psi(x)$ is localized around $x = 0$, so we expand around $u = v$. This yields (in 1D):

$$\tau(u) \simeq \tau(v) + (u - v)\tau'(v) \rightarrow u' = u - \tau(v) - (u - v)\tau'(v) \tag{157}$$

So, if $\tau(v)$ and $\tau'(v)$ are small, we have:

$$\begin{aligned} W(g.x)(v, \log s) &\simeq \int x(u') \psi \left(\frac{v - \tau(v) - u'}{s(1 - \tau'(v))} \right) \frac{du'}{s(1 - \tau'(v))} \\ &\simeq Wx(v - \tau(v), \log s - \tau'(v)) \end{aligned} \tag{158}$$

This recognizes the convolution of x with a translated wavelet at $v - \tau(v)$ and rescaled in $\log s' \simeq \log s - \tau'(v)$. Once we have this result, we want to remove the part related to scales (see Eq. 150). To do that, we perform a Taylor expansion on W:

$$W(g.x)(v, \log s) \simeq W(g.x)(v - \tau(v), \log s) - \tau'(v) \left(\frac{\partial Wx(v - \tau(v), \log s)}{\partial \log s} \right) \tag{159}$$

Therefore, we need to know the sensitivity of the wavelet transform along the scale axis. For this, we'll use the following lemma (in 1D):

Lemma 3

$$\frac{\partial Wx(u,\log s)}{\partial \log s} = -Wx(u,\log s) - \overline{W}x(u,\log s) \tag{160}$$

$$\overline{W}x(u,\log s) = x * \bar{\psi}_s \quad \text{with} \quad \bar{\psi}(u) = u\psi'(u) \tag{161}$$

This can be easily derived by starting from the definition of the wavelet transform at position u and scale s, and then taking the logarithmic derivative, which introduces the new wavelet $\bar{\psi}$. The error term becomes:

$$\begin{aligned} \|W(g.x)(v,\log s) - W(g.x)(v-\tau(v),\log s)\| &\leq \|\tau'\|_\infty \|Wx + \overline{W}x\| \\ &\leq \|\tau'\|_\infty \left(\|x\| + \|\overline{W}x\|\right) \end{aligned} \tag{162}$$

Thus, this proves Lemma 1, although we have truncated the Taylor series at order 1 to give the idea. **So, qualitatively, when a deformation is applied, the result is a translation both in space and in scale, with the error being on the scale translation, which is** τ'. One remark: the "real" proof follows a completely different path and is much more complex, not relying on a Taylor expansion, which has issues with controlling higher orders. The details can be found in the aforementioned article. However, the main idea for why it works is as described here.

Now, we are left with Lemma 2, for which we will provide the main qualitative argument. We need to control:

$$(g.z) * \phi_J - z * \phi_J = \int z(u-\tau(u))\phi_J(v-u)du - \int z(u)\phi_J(v-u)du \tag{163}$$

As $\|z\|^2 = \int |z(v)|^2 dv$, it follows after a change of variables $u' = u - \tau(u)$, and dropping the terms involving $\tau'(u)$:

$$\|(g.z) * \phi_J - z * \phi_J\|^2 = \int \left| \int z(u) \left(\phi_J(v-u-\tau(u)) - \phi_J(v-u)\right) du \right|^2 dv \tag{164}$$

The estimate for the difference in the kernel ϕ_J at $v-u-\tau(u)$ and $v-u$ leads to the constraint that this difference is bounded by the sup. value of the derivative ϕ'_J, multiplied

by the sup. value of the translation $\tau(u)$, all over the support of ϕ_J. Thus,

$$\begin{aligned}\|(g.z) * \phi_J - z * \phi_J\|^2 &\leq \int\int |z(u)|^2 |\left(\phi_J(v-u-\tau(u)) - \phi_J(v-u)\right)|^2 dudv \\ &\leq \int\int |z(u)|^2 ||\phi'_J||^2_\infty ||\tau||^2_\infty \mathbb{I}_{2^J}(v-u) dudv \end{aligned} \tag{165}$$

The infinity norm of ϕ'_J brings out a factor of 2^{-J}, and the remaining integral over u with the indicator function brings out the norm of z. This completes the "demonstration" of the lemma. ■

8.6 Summary

Beyond the technical aspects of the demonstration, a fundamental concept emerges: when dealing with a form of variability that we wish to eliminate, we attempt to transform it into a form of translation along a new parameter. This translation can then be absorbed through averaging. In the example presented thus far, this parameter was the scale of dilation, denoted as s, which "absorbs" the deformation, and we can remove it using a pooling operation. However, there is a significant drawback to this approach: it leads to information loss.

This highlights a critical challenge faced by various descriptors such as Mel-frequency cepstral coefficients (MFCCs) or DAISY descriptors, which were commonly used until the 2010s. Researchers were aware that the averaging process was problematic. Indeed, excessive averaging results in the loss of the original structure, which is crucial for tasks like classification or recognition. Therefore, averaging operations were often performed over relatively short temporal or spatial windows, e.g., 25-millisecond windows in audio processing.

However, when we aim to capture structures at much larger scales, we cannot widen the averaging window arbitrarily. Alternative techniques are required. For images, averaging was done over patches of 8×8 pixels (or a maximum of 16×16). Nevertheless, local descriptors could not capture structures beyond these patch sizes. The key challenge lies in finding descriptors that are sensitive to larger scales. To achieve this, we must cas-

cade transformations (linear transformation, ReLU, pooling), but this introduces greater complexity in their control.

What is surprising, however, is that these cascaded structures are observed in neurophysiology in the auditory and visual cortex, as well as in deep neural networks.

9. Lecture 15 June

This session was recorded after the COVID-19 lockdown period.

9.1 Some Reminders

9.1.1 Convolutional Networks

S. Mallat starts by briefly outlining the architecture of convolutional networks (Fig. 5). Cascades of convolutions reduce the dimensionality of the representation $\Phi(x)$, which is then processed by a dense classifier, for example. The most remarkable aspect is that this type of architecture can yield remarkable results in diverse domains, including image classification, audio analysis, various aspects of natural language processing, as well as problems in physics, chemistry, and medicine. The real challenge lies in being able to **interpret these performances**.

We have studied three types of mathematical properties to reveal the structures in these neural networks: **multi-scale** aspects, **symmetries**, and **sparsity**. The algorithmic use of filter cascades and subsampling reveals that as we delve deeper into the network, neurons are responsible for handling larger-scale aspects of the input, such as the original image. Symmetries are also crucial in designing the architecture, and the first symmetry exploited is **translation covariance**, which arises through the use of **convolutions**. Sparsity emerges somewhat incidentally in the learning process, and it is manifested by the fact that, in a layer of neurons, the response to a stimulus is primarily encoded by a few neurons. However, this sparsity is fundamental, and we will revisit it in the future.

Key questions include:

— How does **dimension reduction**, taking $x \in \mathbb{R}^n$ to $\Phi(x) \in \mathbb{R}^d$ with $d \ll n$, allow us to answer the question $y = f(x)$ without losing information?

— Why are **convolution/subsampling cascades** effective in addressing this problem?

— What is the purpose of **"non-linearities"**?

— The final layer $\Phi(x)$ essentially linearizes the problem, but **what are we ultimately linearizing**?

— In convolutional cascades, a third parameter, **"the channel",** emerges quite early in the network. The question is to interpret the role of this third dimension, what mathematical concept lies behind it? We will see how it allows us to express **notions of symmetry** and is also used to construct **sparse representations**.

9.1.2 Symmetries of the Problem

As we know, the challenge is the representation/approximation of a function f in very high dimensions. This poses a problem because the data points (e.g., training examples) are far apart, necessitating very strong forms of **regularity** on the underlying function f. We have seen in this year's course (e.g., Sec. 5) how studying the symmetries of the system allows us to find these forms of high-dimensional regularity. We can think of a symmetry as an operator that transforms elements from one class to another within the same class. Let Ω_t be the iso-value set of f defined as:

$$\Omega_t = \{x, f(x) = t\} \tag{166}$$

Then, g is a symmetry of f if:

$$f(g.x) = f(x) \tag{167}$$

if g preserves the Ω_t. For instance, if $x \in \Omega_t$ and g is a symmetry of f, then $f(g.x) = f(x) = t$, implying that $g.x$ is an element of Ω_t. In the case of regression where t takes continuous values, g preserves *level sets* of f. The set of symmetries g of f forms a group [66]; this is **the group of symmetries of** f.

The use of symmetries of f is, in fact, through **the symmetries of the representation** Φ, which is learned in the case of a neural network. There is another way to impose

66. Simply put, we study the composition $g_1.g_2$ and note that $f(g_1.g_2.x) = f(g_2.x) = f(x)$.

symmetries directly by having a *priori* knowledge about the symmetries of the problem, as we have seen through wavelet-based MFCC descriptors (see Secs. 7 and 8), and the final phase (linearization) is learned to adapt to the problem. Note that in the case of a neural network, the *a priori* information is contained in the network's architecture [67]. Thus, there are **two distinct steps**: one linearization from the action of the group from x to $\Phi(x)$, and one linearization from $\Phi(x)$ to y, the network's response.

The first commonly encountered symmetry is **translation**. However, we can go a step further by analyzing examples such as digit recognition. It's clear that a small local deformation doesn't change the digit's class. So, if x is a 3, then x', defined by [68] a position-dependent translation as:

$$x'(u) = x(u - \tau(u)) \tag{168}$$

with $||\tau|| \ll 1$, is also recognized as a 3, as shown in Figure 15. Therefore, these small deformations, which constitute **diffeomorphisms**, are symmetries of this problem. This is crucial because **the dimension of the underlying group determines the dimensionality reduction factor** of the problem. In the case of diffeomorphisms, we have a **colossal-dimensional group**, well-suited for reducing the dimensionality of images with millions of pixels or sounds with billions of samples. This reduction allows us to **tackle problems that require viewing either the entire image or a musical/vocal phrase in their entirety**. In the analysis of voices, temporal translations and frequency transpositions transform a female speaker into a male speaker with different timbres and rhythms.

There are other types of symmetries, such as **rotation**, either associated with or without local deformations. For example, the recognition of tree bark textures must incorporate these $SO(2)$ group symmetries. It is also common to encounter **scale changes** (with/without deformations) simply due to zooming in on structures. In this case, the group is $\mathbb{R}$. Thus, as we can observe, the symmetries of the problem we want to preserve in the representation $\tilde{f}$ of the solution function f are very diverse and problem-dependent (image or voice recognition). We also realize that there are many symmetries that we do not know.

67. We should also remember that our understanding of the world is influenced by *a priori* knowledge due to the architecture of our brain.

68. u is the position variable, e.g., of a pixel in 2D

9.1.3 Creation/Use of Invariants

Let's recall the steps: starting from x as input, we establish/learn a representation $\Phi(x)$, and to obtain an approximation $\tilde{f}(x)$ of the underlying function $f(x)$, we linearize as follows (see Sec. 5.4)

$$\tilde{f}(x) = \langle w, \Phi(x) \rangle = \sum_k w_k \phi_k(x) \tag{169}$$

(Note: remember that w is learned even if we have Φ in the first phase). If we want to ensure that the symmetries of f are preserved, i.e., that

$$\forall g \in G,\ \tilde{f}(g.x) = \tilde{f}(x) \tag{170}$$

then we want

$$\forall x,\ \langle w, \Phi(x) - \Phi(g.x) \rangle = 0 \tag{171}$$

meaning that w and $\Phi(x) - \Phi(g.x)$ should be orthogonal. We have two strategies depending on the scenario:

— **Either G is known**, then we will try to find Φ such that $\Phi(x) = \Phi(g.x)$, i.e., Φ is equivariant with respect to the elements of G. However, apart from problems with small dimensions or very specific cases, we often only partially know the group of symmetries.

— **Or $G \subset G'$ with G known but G' unknown**. In this case, we try to ensure that there exists a w that defines the normal to a hyperplane in which $\Phi(x) - \Phi(g.x)$ evolves with $g \in G'$ (see Fig. 17). Through this mechanism, **we "kill" the variability of the problem according to the action of** G (Note: there remains variability in other potentially hidden symmetries that one could try to uncover in a post-analysis). The direction w is learned in the final linear layer (dense part) of the network.

So, to learn the group action $g \in G'$, we can linearize $\Phi(x) - \Phi(g.x)$ using small transformations, and if we impose that Φ is Lipschitz (a form of regularity), then (see the developments leading to Eq. 46)

$$\|\Phi(x) - \Phi(g.x)\| \leq C|g|_{G'} \|\Phi(x)\| \tag{172}$$

meaning that **the difference between $\Phi(x)$ and $\Phi(g.x)$ should be of the order of the transformation** g. If Φ is "differentiable" (in the weak sense) with respect to the action of

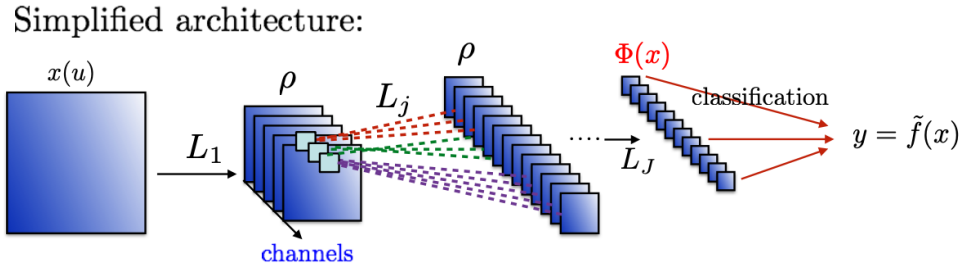

FIGURE 50 – Graphical representation of an architecture purely based on Wavelet filters.

g, then the tangent plane exists.

We have seen how this scheme can be implemented in the following cases:

— The case of translations $g.x(u) = x(u - \tau)$ where we impose that $\Phi(g.x) = \Phi(x)$.

— The case of diffeomorphisms where

$$g.x(u) = x(u - \tau(u))$$

an expression that can be linearized, and we then find that the transformation decomposes into a *global translation* and a *local deformation* such that (see Eq. 43)

$$|g|_G = \|\tau\|_\infty + \|\nabla\tau\|_\infty$$

9.2 Application in a Neural Network

We will demonstrate that we can apply the program of taking into account the *a priori* symmetries of the problem in a network with completely known filters and based on wavelets (**Scattering network**). That is to say, **once we know the symmetry group of the problem, there is no need to learn the filters.** This is particularly true for symmetries such as translation, rotation, frequency transposition, or even deformations.

The architecture we arrive at is schematically represented in Figure 50. It closely resembles a convolutional network as commonly conceived today (cascade of filters/subsampling and non-linearity). A notable difference is that here we have tree-like

structures from one layer to another of filters (see the colored L_j) and initially, **the different channels do not communicate** (along the 3rd axis), which simplifies the problem. However, what we will learn/observe is that we will end up with **networks whose performance is limited and lower than that of 'classical' CNNs**. We will thus learn what **these channels** contribute to problem-solving and why they make 'classical' CNNs so effective.

9.3 Step 1: Scale Separation

We will use the Wavelet Transformation that we introduced in 2018 and detailed during this year more specifically. Let's assume that the problem is related to the analysis of signals that depend on the time variable t. We will use a wavelet $\psi(t)$ that we translate and dilate by a scale λ:

$$\psi_\lambda(t) = \lambda^{-1}\psi(\lambda^{-1}t) \tag{173}$$

In the case of audio, we take $\lambda = 2^{j/Q}$. The wavelet basis is formed by dilations and translations. The Wavelet Transform involves calculating the correlation of the signal x with the dilated wavelet, which is expressed as convolution/filtering:

$$(x * \psi_\lambda)(t) = \int x(u)\psi_\lambda(t-u)du \tag{174}$$

In the Fourier domain, this convolution corresponds to the product of Fourier transforms, namely:

$$\widehat{(x * \psi_\lambda)}(\omega) = \widehat{x}(\omega)\ \widehat{\psi_\lambda}(\omega) \tag{175}$$

We have seen in the course examples of wavelets, and in the Fourier domain, $|\widehat{\psi_\lambda}|^2$ are **band-pass filters** (see Fig. 40).

If we compress the wavelet in the time domain, it is dilated and shifted to higher frequencies in the Fourier domain. In logarithmic scale, the width of the Fourier filter is constant (see Fig. 39), which leads to the term "*Q-constant band-pass filters*". We have seen that if we low-pass filter the decomposition at a scale 2^J, we need to associate a **low-pass filter ϕ_{2^J} that provides an average approximation** of the function, while the **Q-filters provide the details** of the function. Therefore, the complete Wavelet decomposition

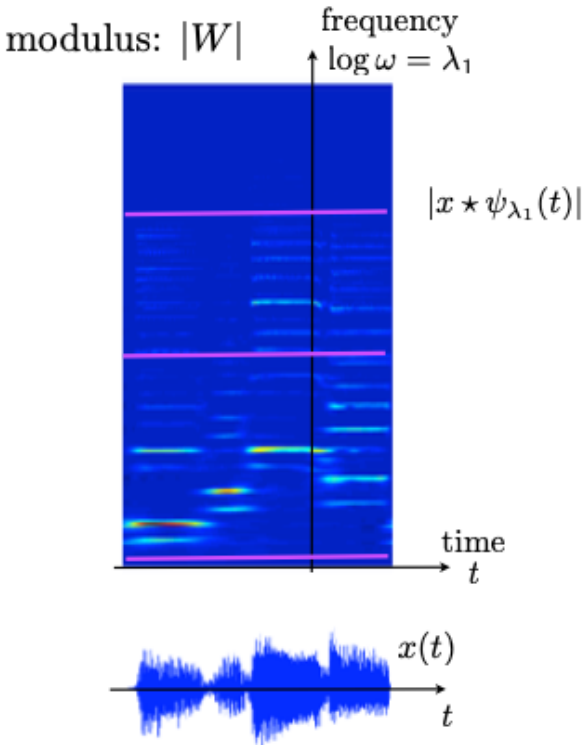

FIGURE 51 – Example of Wavelet decomposition of a signal $x(t)$: the color represents the value of $\|Wx\|$ with blue indicating zero values. We can observe the sparsity of the frequency decomposition and its temporal evolution.

is as follows ($\lambda = a^j$ with $a = 2^{1/Q}$):

$$Wx = \begin{pmatrix} x * \phi_{2^J} \\ x * \psi_{a^j} \end{pmatrix}_{a^j \leq 2^J} \tag{176}$$

If the wavelets cover the frequency band well, then the transformation preserves the signal norm, i.e., $\|Wx\|^2 = \|x\|^2$.

An example of decomposition is shown in Figure 51. We have also seen in Section 7.3 and in the seminars by Geoffroy Peters on February 12, 2020, and Shihab Shamma on March 12, 2020, that these types of wavelet decompositions are physiologically implemented, especially in the *cochlear auditory system* and in the *auditory cortex*, where strong

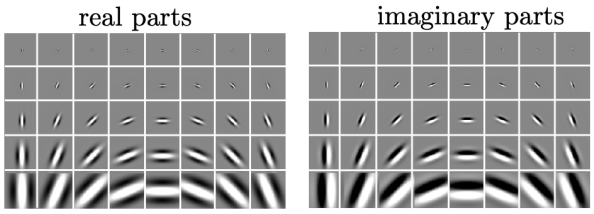

FIGURE 52 – Representation of complex dilated and rotated wavelets.

non-linearities appear, and the neuron responses are insensitive to translation, allowing the processing of information over a large time scale. Of course, we would like to understand how the brain processes this information at different stages (similarly to the visual cortex).

In the case of *images*, the principles are essentially the same. For example, the wavelet can be represented by a complex function like a Gaussian modulated by a cosine for the real part and a sine for the imaginary part. However, with translation and dilation, we add rotations. Thus, a transformed wavelet is expressed as (see Sec. 8.2, Fig. 47):

$$\psi_{\theta,j}(u) = 2^{-2j}\psi(2^{-j}(r_{-\theta}.u)) \tag{177}$$

An example is shown in Figure 52, which demonstrates that such wavelets are sensitive to the orientation of image details. The wavelet transform, much like in audio, is an image filtering operation by convolution:

$$(x * \psi_{\theta,j})(u) = \int x(v)\psi_{\theta,j}(u-v)dv \tag{178}$$

This can also be visualized in the "frequency" domain as shown in Figure 48. We can then add a function ϕ_{2^J} that covers the "low frequencies", and we define Wx based on the same principle as equation 176 (note that here for images, $a = 2$):

$$Wx = \begin{pmatrix} x * \phi_{2^J} \\ \\ x * \psi_{2^j,\theta} \end{pmatrix}_{j \leq J,\theta} \tag{179}$$

with the same properties regarding the preservation of the signal norm. What is interesting

(see Simon Thorpe's seminar) is that these wavelets have been found in the visual cortex. Similarly, as we delve deeper into the brain structures, we find strong non-linearities and forms of translation invariance, followed by small networks sensitive to very global effects, which enable, for example, face recognition invariant to various changes in position, texture, aging, etc. Understanding how the brain processes this information at different stages is a current research topic in relation to advances in neurophysiology.

From a mathematical point of view, why are wavelets effective and appear to be at work in the brain? Let's revisit arguments discussed this year:

— Firstly, there is a form of **stability with respect to deformations**. If we perform the following transformation:

$$\psi_\lambda(u) \to \psi_\lambda(u - \tau(u)) = \psi_{\lambda,\tau}(u)$$

then the difference between the transformed wavelet and the original copy is of the order of the deformation:

$$\|\psi_\lambda - \psi_{\lambda,\tau}\| \leq C\|\nabla\tau\|_\infty$$

— We naturally have **scale separability** with the wavelet transformation, which aids in dimensionality reduction (see Sec. 4.3);

— Finally, the wavelet transformation provides a very **sparse** representation, allowing the highlighting of *signal features*.

9.4 Step 2: Translation Invariance

Descriptors must be able to capture this type of **translation invariance**. Wavelet coefficients help in constructing these descriptors, as we have seen for MFCCs (audio) and their equivalents for image processing.

In fact, we don't have much choice: **we need to average the signal in a certain way**. If we want to obtain an average at a scale of 2^J, we proceed by convolving the signal x with the "low-frequency" function ϕ_{2^J}. The result is an approximation of f that, when translated, gives wavelet coefficients that do not vary much. At the extreme, if we truly

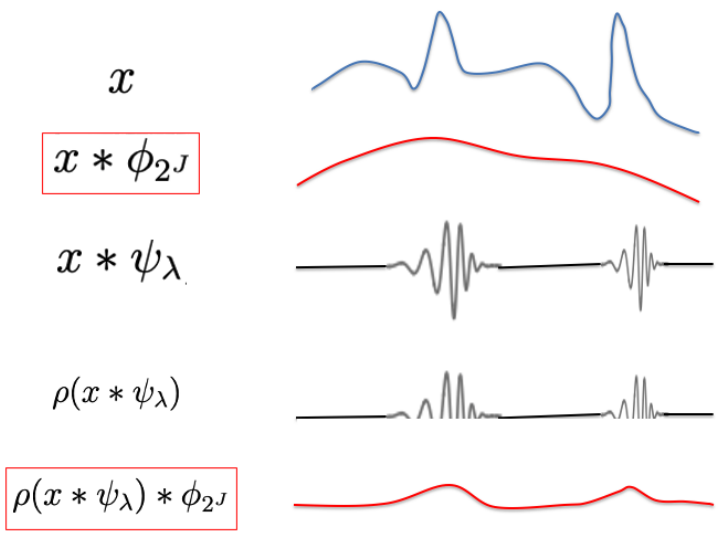

FIGURE 53 – Development of translation invariants (in red) from wavelet coefficients of the signal using the rectifier ρ for high-frequency components and averaging with ϕ_{2^J}.

impose translation invariance, then we must completely average the signal, which happens when $2^J \to \infty$. However, in this case, all the structure of the signal is lost. What we have seen in the course, however, is that **non-linearities** are necessary to obtain invariants that capture information "lost" by averaging.

The "lost" information is the high-frequency variations of the signal, which can be captured by the wavelet transform using different values of λ (note that here λ is a generic scale suitable for audio or image cases). However, as ψ is an oscillatory function with zero mean, **averaging the wavelet coefficients of the signal alone yields a zero value**. Therefore, a linear filter cannot be used. Instead, a rectifier ρ can be applied, which sets the coefficients to zero if they are negative. **Thus, averaging $\rho(x * \psi_\lambda)$ no longer yields a zero value, preserving the information about the location of rapid signal variations**. To construct an invariant from the rectified coefficients, we proceed with averaging using ϕ_{2^J}, just like for the signal itself. This yields $\rho(x * \psi_\lambda) * \phi_{2^J}$. The different steps of building translation-invariant descriptors that do not lose the signal's structure are schematically shown in Figure 53.

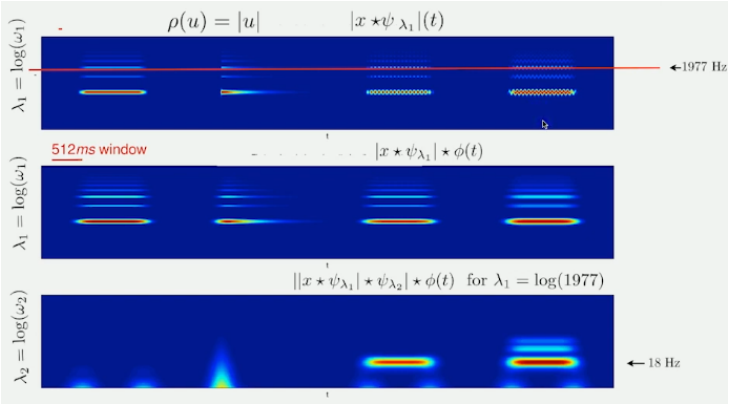

FIGURE 54 – Example of the result of the cascade of scattering operators. The top scalogram is for the signal composed of three sounds from left to right: a simple note, an "attack", a tremolo, and a vibrato. The first stage (middle scalogram) that calculates $\rho(x * \psi_{\lambda_1}) * \phi_{2^J}$ will keep the harmonic structures but lose some of the fine structure: the attack is not clearly visible, and the oscillations of the tremolo and vibrato are lost. The second stage (bottom scalogram) that performs $\rho(\rho(x * \psi_{\lambda_1}) * \psi_{\lambda_2}) * \phi_{2^J}$, for example, with λ_1 corresponding to 1977 Hz, yields low-intensity coefficients for the first note, shows fairly large coefficients to mark the attack of the second note, and yields large coefficients exactly at the frequencies of the tremolo and vibrato.

9.5 Scattering Operators

However, what remains true in the described steps is that we are averaging over a potentially large scale 2^J, so something is still lost. We have seen that such descriptors (MFCCs, SIFT), which were widely used (before convolutional networks), have the major disadvantage of being sensitive only to relatively small scales. Why? The reason is that if the support of ϕ_{2^J} is too large, information is lost, but if it is too small, translation invariance is lost. Therefore, we would like to obtain invariants over much larger domains. In fact, we can recover the high frequencies of the rectified coefficients, e.g., $\rho(x * \psi_{\lambda_1}) * \psi_{\lambda_2}$, which can then be rectified and averaged again... **It is clear that cascading series of "filter/rectification" pairs must be linked together.** These cascades are called **scattering operators**, which are a simplified version of neural networks. A complete example is given in Figure 54 with four types of sounds having different structures.

At each scattering stage, we recover the information that was removed in the previous stage.

How is this architecture related to neural networks? In a neural network, we apply a sequence of convolutions and non-linearities. What we will illustrate is that these sequences of convolutions precisely involve a wavelet decomposition. To do this, let's take a 1D signal $x(u)$ and set the goal of calculating the average of this signal. Of course, we could calculate it directly. However, we can achieve this goal efficiently by aggregating values in pairs. We then obtain successively:

$$\begin{aligned} x_0(u) &= x(u) \\ x_1(u) &= H[x_0](u) = \frac{x_0(2u) + x_0(2u+1)}{2} \\ x_j(u) &= H[x_{j-1}](u) = H^{(j)}[x_0](u) \\ &\dots \end{aligned}$$

So, if $x(u)$ has 2^N initial values, we obtain successively 2^{N-1}, 2^{N-2}, and so on, intermediate values, and finally 1 value in N steps, which is the average of the signal. Therefore, obtaining information invariant to translation is very simple.

But at each step, we lose information. Here, the lost information corresponds to pairwise differences. So we have a first filter H for pairwise averaging and a second filter G for pairwise differences:

$$\{x(u)\}_{u\leq d} \to \begin{cases} \left\{\frac{x(2u)+x(2u+1)}{\sqrt{2}}\right\}_{u\leq d/2} & (H) \\ \left\{\frac{x(2u)-x(2u+1)}{\sqrt{2}}\right\}_{u\leq d/2} & (G) \end{cases} \tag{180}$$

Both operations Hx and Gx are actually two convolutions, with the first being a low-pass filter h (values $(1,1)$) and the second being a band-pass filter g (values $(1,-1)$):

$$Hx(u) = x * h(2u) \qquad Gx(u) = x * g(2u) \tag{181}$$

If we iterate the decomposition on successive approximations (H, $H^{(2)}$, $H^{(3)}$, ...), we obtain a tree structure as shown in Figure 55, which reveals the Haar wavelet ψ and its associated scaling function ϕ.

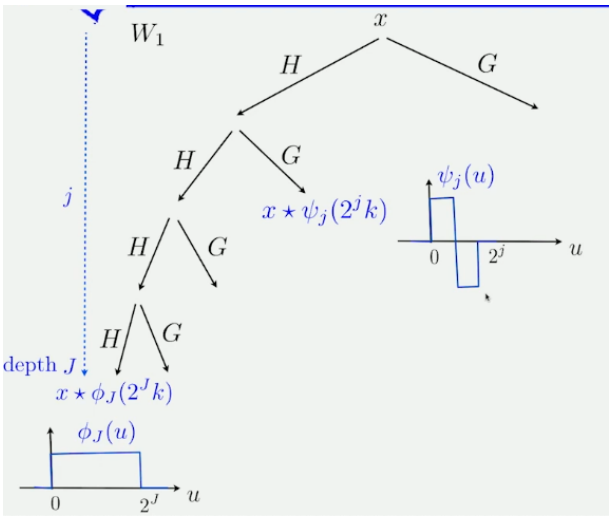

FIGURE 55 – Successive cascade of low-pass (H) and high-pass (G) filter applications, revealing the Haar wavelet ψ and its associated scaling function ϕ.

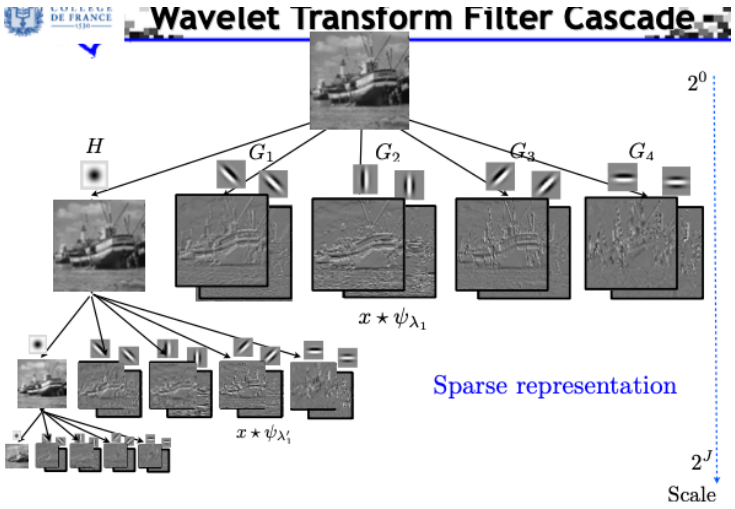

FIGURE 56 – Similar cascade as in Figure 55 with two types of low-pass and high-pass filters in the case of image processing. Dark areas indicate regions where the coefficients are zero.

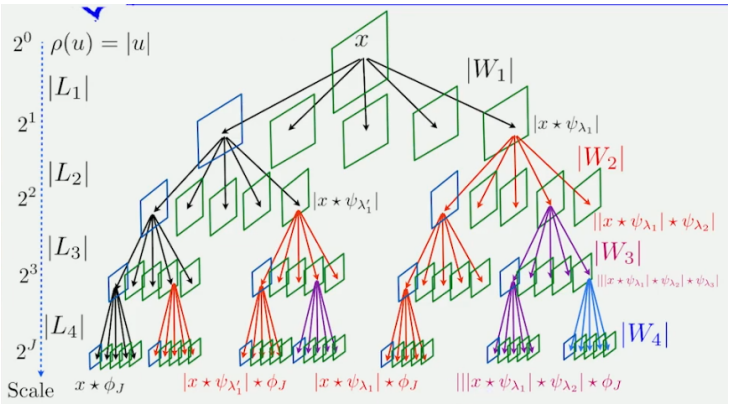

FIGURE 57 – Cascade decomposition of an image: at each stage, we obtain a wavelet decomposition from which we extract an invariant by applying a non-linearity (here, the modulus) and averaging at scale 2^J. We successively obtain a first-order invariant $x * \phi_J$, second-order invariants $|x * \psi_{\lambda_1}| * \phi_J$, and third-order invariants $||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi_J$, and so on.

Of course, this cascade filter structure can be generalized beyond the use of the Haar wavelet. A 2D version is illustrated in Figure 56, where, once again, the sparsity of the decomposition is evident: almost all coefficients are zero except at the boundaries between regions of the same gray intensity in the original image.

Now, as we have seen, once we have the wavelet decomposition, to find invariants, we must first apply a non-linear operator to the detail coefficients at each scale 2^j (such as ReLU or modulus), and secondly, perform averaging. Simultaneously, each detail image can also be analyzed by a cascade of second-order wavelets, and so on. This results in the tree-like structure shown in Figure 57.

The interpretation of this decomposition tree can be seen as that of **a neural network where all filters are wavelets** (scattering network):

$$S_J(x) = \{|||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \dots \psi_{\lambda_m}| * \phi_J\}_{\lambda_k} \tag{182}$$

Horizontally, we have the application of filters W_k, and vertically, **we extract increasingly global invariants** (the deeper we go into the network, the larger the introspection scale).

We have seen (cf. Th. 8) that this cascade decomposition of filters, nonlinearities,

and averaging is **contracting**. As a reminder:

— The wavelet decomposition is a linear operator that preserves the norm, cf. $\|Wx\| = \|x\|$.

— The non-linearity is a contracting operator because for any a and b, $|\rho(a) - \rho(b)| \leq |a - b|$, so

$$\|\rho W(x) - \rho W(x')\| \leq \|x - x'\|$$

Therefore, since $S_J(x)$ consists of a chain of contracting operators, **S_J, the output of the scattering network, is a contracting operator** and is stable in L2 norm:

$$\boxed{\|S_J(x) - S_J(x')\| \leq \|x - x'\|} \tag{183}$$

Another result we studied in detail (Sec. 8.5) is **the stability of S_J with respect to (small) deformations**, which is a Lipschitz regularity. We have this type of inequality:

$$\boxed{\text{If } D_\tau(x)(u) = x(u - \tau(u)), \text{ then } \lim_{J \to +\infty} \|S_J.D_\tau(x) - S_J(x)\| \leq C\|\nabla\tau\|_\infty \|x\|} \tag{184}$$

(Note that here we have omitted the Hessian of the transformation to simplify.) In fact, this is because a deformation corresponds approximately to a translation along the scales.

Why is it important to obtain invariants that are stable under deformations? Again, the reasoning is as follows: if the problem is invariant under certain transformations, incorporating this *a priori* information into the network's structure allows us to linearize the problem or, in other words, to eliminate this variability, making the approximation problem simpler due to lower dimensionality. And this is even more effective when the invariance group has a large dimension, hence the importance of diffeomorphisms.

9.6 Some Applications of Scattering Networks

9.6.1 Digit Classification

The first case study is the **classification of MNIST digits** carried out by Joan Bruna Estrach [69] in 2012. The network consisted of two blocks: the first one was the Scattering

69. See his thesis https://www.di.ens.fr/data/publications/papers/phd_joan.pdf and related articles.

network, which linearized translations and small deformations (diffeomorphisms) with wavelet filters, and a second block specifically designed for digit classification. The first block requires no learning, while the second block learns the classification vector w through logistic regression, for example. The results at the time showed a classification error of 0.4%, identical to that obtained by Y. LeCun et al.'s convolutional network [70]. In the case of MNIST, there is a training database of 50,000 digits (and 10,000 for testing), so there are enough data to learn all the filters as well as the parameters of the final classifier. Therefore, it's not surprising that the results are identical, especially since the main goal is to capture variability due to translations and deformations, a task for which wavelets are inherently well-suited.

9.6.2 Texture Classification

Another problem is **the classification of textures from the CUReT database** (Columbia-Utrecht Reflectance and Texture Database [71]), also analyzed by Joan Bruna Estrach. **This is a small set of textures** of 200×200 pixels distributed across 61 classes. Here, the training set is extremely small compared to the classification task, as there are only 46 images per class. Clearly, a network like `AlexNet` or similar, with about 60M parameters [72], cannot be used. **The advantage of the scattering network is that the first part of the network does not need to be learned**, and learning is a "simple" logistic regression with 61 parameters. The obtained error rate is 0.2%, much lower than the result obtained by Fourier analysis, which gave 1% error. This is because two different textures can have the same power spectrum, so to differentiate them, **it's necessary to analyze non-Gaussianities**, which the scattering network does [73].

9.6.3 The Role of Channel Connections

If we analyze the typical diagram of a scattering network in Figure 57 (or 50), we notice the **absence** of the operation shown in Figure 4, which connects images along **the**

70. Note that PCA or SVM methods do not achieve this level of error.
71. https://www.cs.columbia.edu/CAVE/software/curet/, and see Joan Bruna Estrach's thesis as well.
72. Note that most of the parameters are in the dense part of the classifier, not the convolutional part.
73. S. Mallat indicates that this will be the subject of a lecture: the analysis of non-Gaussianities with this type of network.

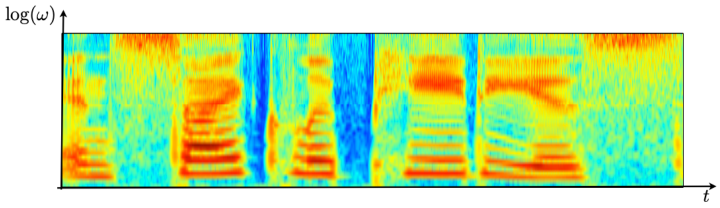

FIGURE 58 – Scalogram of an audio frame: the horizontal axis is time, and the vertical axis is the scale of the wavelet decomposition, also the axis of "channels". The color scale indicates the intensity of wavelet coefficients.

channel axis. **The results of the filters W_k are not connected**. This is a significant difference from a 'classic' neural network. So, the questions that come to mind are: **what is the role of these connections?** and **why are they important?** In fact, we will see that they contribute to the linearization of **other types of symmetries** beyond translations/deformations and enable the learning of discriminant patterns.

Let's return to the analysis of an audio frame, whose scalogram is shown in Figure 58. Notice that the vertical scale, the scale of the wavelet decomposition, is also **the axis of "channels"**, as at a fixed t, we have the results of the filters at different scales. We have seen that this scalogram varies for the same spoken word depending on the speaker due to differences in phoneme attacks, rhythm, etc. **So, if we want to extract features that allow us to discern two speakers, we need to perform a transformation that connects the coefficients along the scale axis/the channel axis at a fixed t**. This has been achieved, for example, by a time-frequency analysis in the paper by J. Anden, V. Lostanlen, and S. Mallat [74]. The signal $x(t)$ is first decomposed using cochlear-type wavelet filters to obtain the scalograms of the coefficients $|x \overset{t}{*} \psi_\lambda|$, where convolution is performed along the time axis (as usual). We interpret this scalogram as an image, which we then decompose into bi-dimensional wavelets, allowing filtering in both directions t and $\log \lambda$. At the end of the process, we have invariants of the signal through averaging [75] at the level of the traditional scalogram $|x \overset{t}{*} \psi_\lambda| \overset{t}{*} \phi_J$ and also at the level of the analysis of the time-frequency image, which involves both temporal and "frequency" convolution $||x \overset{t}{*} \psi_\lambda| \overset{t}{*} \psi_\alpha \overset{\log \lambda}{*} \psi_\beta| \overset{t}{*} \phi_J$. What is remarkable is that this type of analysis first appeared in neurophysiology, in

74. https://www.di.ens.fr/~mallat/papiers/IEEESignalAndenLostanlen.pdf
75. ϕ_J is a Gaussian average over 500 ms.

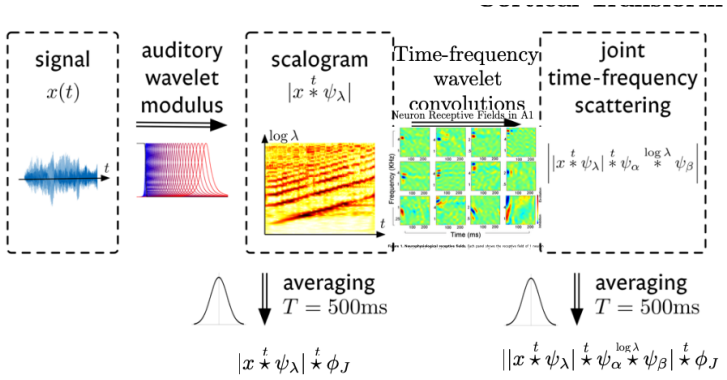

FIGURE 59 – Example of processing a 1D signal decomposition by wavelets, in which the scalogram is treated as an image by 2D wavelets, filtering along both the time and scale axes, which is the channel axis.

what S. Shamma calls *cortical transform*, which performs bi-dimensional filtering and was identified in the auditory cortex of ferrets.

S. Mallat presents other works from 2016-17 on the recognition of instrumental phrases (see MedleyDB with 10,000 training examples for 8 classes: clarinet, electric guitar, singer, violin, etc.) or urban sounds (see UrbanSound8k: 8,000 samples for 10 classes: horns, barking, sirens, etc.) that show that with small training databases, **the analysis of time series with scattering network scalograms**, one obtains error rates for misclassification much better (around 20%) than those obtained with MFCC descriptors (state of the art before neural networks), which achieve 40%, or with convolutional networks, or even with a simple scattering network (without channel connections), which achieve around 30%. It is clear that these are somewhat older examples but they demonstrated the importance of channel correlations.

9.6.4 Texture Classification with Rotations/Zooms

Another example [76] concerns the recognition of textures where **rotations and scale variations** (zoom) can be of very significant magnitudes, as illustrated in Figure 60.

76. Laurent Sifre's Ph.D. thesis (2014) https://www.di.ens.fr/data/publications/papers/phd_sifre.pdf

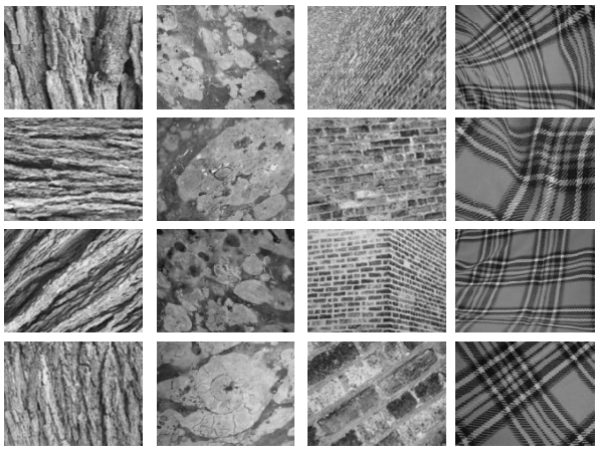

FIGURE 60 – Example textures from the UIUC database.

We start with images and first need to perform a wavelet decomposition with elements that can capture "edges" in all directions, as shown in Figure 56. We use wavelets $\psi_{2^j,\theta}$ as in Eq. 177. Then, if we want rotation invariance, we need to connect the results along the θ axis. If we want invariance to zoom/scale changes (dilation/contraction), we do the same along the scale axis, and we can combine both (see Figure 61).

The goal is to capture variability along the angles while designing rotation invariants while still being able to retain structure details along the angles. The image $x(u)$ (where u is the pixel index) passed through the wavelet decomposition yields coefficients $x_j(u,\theta) = |x * \psi_{2^j,\theta}|$, which is an output of the first stage of the network (W_1) that corresponds to bandpass analysis. The other output is the result of the low-pass filter, which provides access to the mean pixel values $\int x(u)du$. Now, at a given scale 2^j, the collection $x_j(u,\theta)$ can be seen as a 2D image indexed by u and θ (using discrete values of angles, as in Eq. 142).

Note that if we apply a rotation of angle α to the original image and a translation of c pixels, the coefficients $x_j(u,\theta)$ become

$$x(u) \to x(r_\alpha(u-c)) \Rightarrow x_j(u,\theta) \to x_j(r_\alpha(u-c),\theta-\alpha) \tag{185}$$

Thus, we observe that the coefficients move along the θ axis. To overcome this variation, how can we construct a rotation invariant? We need to perform a convolution that acts

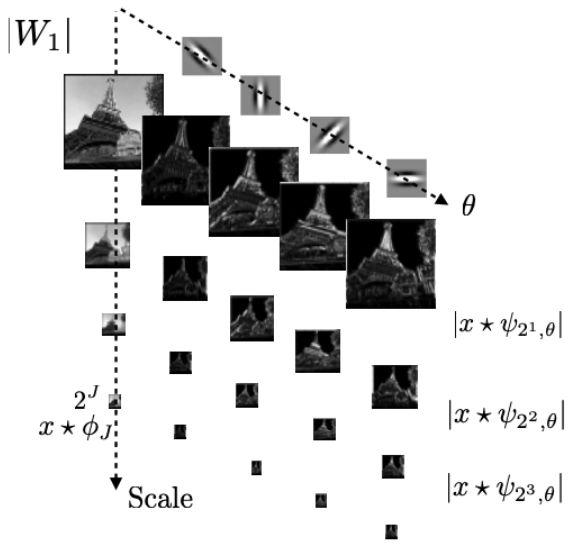

FIGURE 61 – Wavelet decomposition of type $\psi_{2^j,\theta}$ of an image that allows highlighting two types of channels that can be filtered to obtain invariants either by rotation, dilation/contraction or zoom, or both.

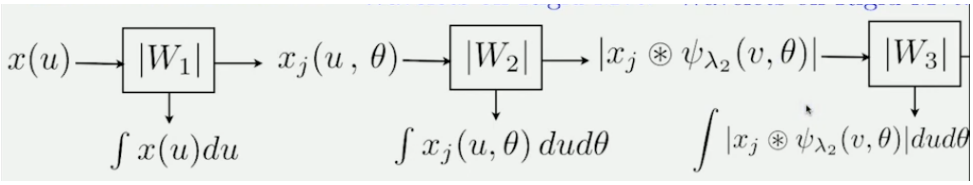

FIGURE 62 – Processing of an image through a series of filters, which, from the second stage onwards, involves convolutions along the angle axis to obtain rotation invariants.

not only in space (u_1, u_2) (pixel indices in 2D) but also along the rotation axis:

$$x_j \circledast \psi_{j',k}(u,\theta) = \int_0^{2\pi} \left(\iint_{\mathbb{R}^2} x_j(u',\theta')\psi_{2^{j'},\theta}(r_{-\theta'}(u-u'))du' \right) \psi_{2^k}(\theta-\theta')d\theta' \tag{186}$$

Here, we use 2D rotated and translated wavelets $\psi_{2^j,\theta}$ and a 1D wavelet ψ_{2^k} translated along the θ axis. This leads to a cascade filtering scheme similar to audio processing, which, in the case of images, yields global rotation invariants (see Figure 62).

Now, if we look at the classification error, we find that if we use a scattering network without considering correlations along the angle axis, we obtain a 20% error. However, if we use rotation invariants, the error rate drops to 0.6%, demonstrating the importance of these invariants and hence the use of channels.

9.6.5 Example in Quantum Chemistry

Taking into account these 'rigid' movements is extremely important as they are present everywhere, particularly in Quantum Physics and Chemistry. The goal is to calculate the energy $f(x)$ of a molecule from its configuration x, including the positions and charges of its constituents. Ultimately, we want to determine whether the molecule is stable or not. The basic idea is that instead of solving the Schrödinger equations (see the Kohn-Sham single-electron technique) to obtain the electron density distribution, we will use deep neural networks while incorporating all the *a priori* knowledge we have about the problem to limit the number of "unknown" parameters that need to be learned. Obviously, $f(x)$ is invariant to translation and rotation, and there are forces at multiple characteristic scales (Van der Waals, covalent bonding, etc.). Thus, we find that the symmetries of the problem in Chemistry are not very different from those encountered in the

texture classification discussed earlier.

When analyzing the physical problem, the Kohn-Sham model writes the energy $E(\rho)$ of the molecule as a sum of different contributions dependent on the electron density ρ:

— Kinetic energy term,

— Attractive interaction term of electrons via an effective potential,

— Coulombic repulsive interaction term between electrons,

— A quantum term, difficult to compute, responsible for the cohesion of the molecule.

At equilibrium, $f(x)$ is given by the minimum of $E(\rho(x))$ where ρ is a function of the configuration x (positions, etc.).

Now, instead of solving the differential equations of the Kohn-Sham system, we naively attempt to use statistical learning techniques by finding an approximation $\tilde{\rho}(x)$ of $\rho(x)$ and writing an energy decomposition of $E(\rho(x))$ as:

$$E(\rho(x)) \approx \sum_k w_k \phi_k(\tilde{\rho}(x)) \tag{187}$$

but we incorporate **our knowledge of the physics of the problem into the descriptors** ϕ_k. The weights represent chemical potentials that we will learn to adapt to the problem based on a database of cases solved by solving differential equations.

So, the first thing to decide is: what type of descriptor ϕ_k will we use? Our guidelines here are the symmetries of the problem. Initially, we use a very naive representation of the electron density, by using the position r_k of each charge z_k as a sum of Dirac functions centered on the atoms:

$$x(u) = \tilde{\rho}(u) = \sum_{k=1}^{d} z_k \delta(u - r_k) \tag{188}$$

If we then perform a wavelet decomposition, convolution with a Dirac function yields the wavelet itself, so

$$\rho\left(x * \psi_{2^j,\ell}(u)\right) = \rho\left(\sum_{k=1}^{d} z_k \psi_{2^j,\ell}(u - r_k)\right) \tag{189}$$

(where ℓ is an orientation index). Each charge k emits a wave $\psi_{2^j,\ell}(u - r_k)$, and all these waves interfere to create increasingly complex interference patterns as the scale becomes larger, as illustrated in Figure 63.

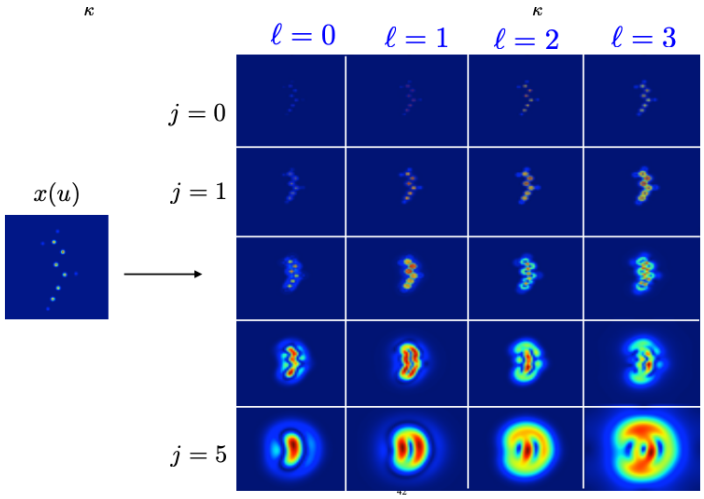

FIGURE 63 – Result of $\rho\left(x * \psi_{2^j,\ell}(u)\right)$ (Eq. 189) from a point charge distribution at the positions of a molecule's atoms.

Then, these interference patterns are used in the decomposition of the system's energy. Naturally, as we have seen in the previous sections, we don't stop there, as we can cascade filtering levels and compute at the second level:

$$\rho\left(\rho\left(x * \psi_{2^j,\ell}\right) * \psi_{2^{j'},\ell'}\right) \tag{190}$$

Then, we average at a certain scale to obtain translation and rotation invariants. Finally, by linear regression (the only learned part), we try to predict the energy.

Quantum chemistry databases (QM9 [77]) contain about 130,000 organic molecules for which configurations and energies have been calculated using traditional methods. Once the learning is done, errors with scattering networks are of the order of 0.5 kcal/mol, which is about the same order of magnitude as the errors of traditional methods for solving Schrödinger's equations. Note that 0.5 kcal/mol is small and of the same order of magnitude as the errors of traditional techniques for solving Schrödinger's equations. But the question is, **why is it so good? Where have the quantum aspects gone?** However, let's put this in perspective: these QM9-type databases consist of small molecules (e.g., a maximum of 29 atoms with 9 heavy atoms), so we can say that the problem solved here

77. http://quantum-machine.org/datasets/

FIGURE 64 – Examples of images from the CIFAR-10 dataset.

is similar in image processing to MNIST, which is now considered a child's play.

9.7 Failure of Scattering Networks

In the previous sections, we presented cases where we can inject *a priori* information and find wavelet-based descriptors whose performance is as good as that of 'classical' convolutional networks where no *a priori* information is used [78]. However, there are cases where scattering networks perform less well.

Let's take the CIFAR-10 database as an example, consisting of 50,000 images of size 32×32 pixels labeled into 10 classes, where there is a large variability within each class that has nothing to do with translations, rotations, or deformations (see Fig. 64). The results show that the approach of scattering networks reaches its limits, and this is the current focus of research.

S. Mallat discusses the work of Edouard Oyallon [79], whose seminar was canceled due to COVID-19. If we take a 'classic' neural network, we achieve a common error rate of 7%. However, when using a scattering network with the wavelet filter component encompassing all *a priori* known symmetry groups (without any learning) and a regression component that is learned, the error rate caps at 20%.

In summary, scattering networks perform well in cases where the known symmetry groups adequately describe the problem's variability, such as digits (10 classes), textures (60 classes), and quantum chemistry with a small number of atoms. However, when tackling more complex problems like ImageNet classification (approximately 2 million 256×256 pixel images, 1000 classes), things change. In 2012, AlexNet, which was a breakthrough in the field, had an error rate of 16.4% (modern convolutional networks achieve error

78. Note: the architecture itself is a *a priori* to keep in mind.
79. Source: https://edouardoyallon.github.io/thesis.pdf

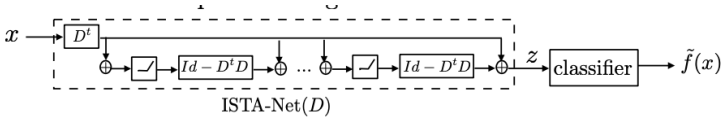

FIGURE 65 – Example of an algorithm (ISTA-Net) for decomposing x into a sparse representation with a cascade of operators consisting of a dictionary D followed by a rectifier to ensure the positivity of z.

rates approximately 1% lower than human error rates [80]). In contrast, scattering networks plateau at 60%, significantly worse than the state of the art before AlexNet, which had an error rate of 26% (see the 2019 course). The question then arises: **What do convolutional networks learn that scattering networks lack?**

One way to approach this question is to remember that historically, this field was called "pattern recognition" or "shape recognition". In other words, we still expect to **recognize structures** [81] that we need to learn to solve problems. The mathematical approach involves using the concept of **dictionaries** and finding ones that provide a **sparse description** of the problem. Typically, we want to project x onto a dictionary $\mathcal{D}$ consisting of k patterns $\{D_k\}_{k\leq d}$ such that

$$x = \sum_{k=1}^{d} D_k z_k = \mathbf{D}.\mathbf{z} \tag{191}$$

and the representation is sparse if $(z_k)_{k\leq d}$ consists primarily of zeros. To achieve such sparsity, we use **L1 norm regularization** to constrain the space in which we search for z so that $\mathbf{D}.\mathbf{z}$ approximates x (see the 2018 and 2019 courses):

$$\tilde{z} = \underset{z}{\operatorname{argmin}} \ \|x - \mathbf{D}.\mathbf{z}\|_2^2 + \alpha\|\mathbf{z}\|_1 \tag{192}$$

This is a convex optimization problem that can be solved with algorithms that guarantee convergence. Some of these algorithms are iterative and can be implemented as neural networks, appearing as a cascade of operators consisting of a single dictionary followed by a rectifier (non-linearity), as shown in Figure 65.

With the machinery in place, the remaining question is: **What are the useful pat-**

80. With certain caveats in this comparison
81. See the Chomsky grammar from the 2019 course

Model	Learning	Error Rate
Scattering Net (1)	No	60%
Dictionaries (2)	Yes	50%
Scattering + Dictionary (3)	Mixed	18%

TABLE 1 – Comparison of different types of networks (see text) on ImageNet classification. For reference, the 2012 AlexNet had an error rate similar to (3).

terns for solving the problem? We don't know them in advance, and **we need to learn them**. The idea is to construct a large matrix representing the dictionary and perform gradient descent optimization using a training set $\{x_i, y_i\}$ to minimize the error (ℓ for *loss*):

$$\ell(D) = \sum_i \ell(y_i, \tilde{f}_D(x_i)) \tag{193}$$

The result is indeed a matrix of elementary structures that sparsely describe the variable x.

An illustration is provided by comparing the results obtained on the ImageNet dataset (Table 2):

1) First, we can implement a scattering network where we try to incorporate all the *a priori* known information about the symmetries of the problem and optimize a classifier that provides the final answer of the network. The result is a 60% error rate.
2) We can also use ISTA-Net-type decomposition, where we jointly learn the dictionaries to obtain a sparse description and the classifier parameters. The result is of the same order, around 50% error, and we have not gained much.
3) Finally, we can combine both networks. In the first step, we use prior knowledge in a scattering network (1). However, **this time, the invariants obtained are passed through a dictionary network** (2), and its output is used for classification. Joint optimization of the dictionary and classifier ultimately yields **an 18% error rate**, comparable to the score of the 2012 AlexNet network.

How can we interpret this reduction from 50-60% to 18%? Returning to scheme (2), we create dictionaries directly from ImageNet images. However, given the high variability within each class, the number of dictionary patterns would be enormous, and effectively learning them would require a colossal number of samples per class. **Conversely, by linearizing with respect to the known symmetries, including diffeomorphisms, we significantly reduce variability, allowing for an efficient search for patterns that capture the remaining symmetries**. It seems that the variability has decreased enough for such a pattern search to be effective and reduce the error rate by a factor of 3.

The advantage of architecture (3) is that we can interpret the different layers of the network, and we only have one matrix to learn. In contrast, while 300-layer architectures like ResNet have error rates of approximately 3%, they remain challenging to interpret. However, it must be said that we do not fully understand the 3-fold reduction in error achieved between architectures (1)-(2) and the combination (3). What is the mathematical nature behind this? There are hypotheses, but there is no good mathematical model that explains the phenomenon.

10. Epilogue

S. Mallat concludes this year's course with some observations. Firstly, we have seen that deep convolutional neural networks (CNNs) achieve remarkable results in very high dimensions for various types of problems (image processing, sound/speech, quantum chemistry, language, etc.). This is even more remarkable because, *a priori*, if the function to be approximated lacks strong regularity, this problem is difficult, if not impossible. Undoubtedly, these CNN architectures can learn something about the objects they process, especially about the symmetries of the problem. However, there remains a mystery about the **Why** of such effectiveness, and in particular, why the same architecture can capture the regularity of very complex and diverse functions. Why do these problems share the same type of regularity/symmetry that CNNs capture?

This year, we attempted to show that there are three approaches to explaining the regularity of the underlying function we are trying to approximate.

1) The first approach focuses on **separability**. By scale separation using multi-resolution analysis, we obtain structures, but more importantly, we can linearize by the action

of **symmetry groups** (e.g., translation, rotation, deformation) and obtain **invariants** that are very rigid structures and reduce problem variability (dimension reduction). This is a common approach in Particle Physics, for example. However, when we reach the macroscopic scale, we inevitably encounter the notion of **patterns**. This is evident in Chemistry, where the elementary building blocks of even quantum-based basic interactions work well at the microscopic scale, for instance, to explain the emergence of conduction bands or other phenomena. But when it comes to the properties of complex molecules, traditional non-quantum Chemistry still accounts for generic properties learned "empirically". The same reasoning applies to sound analysis, where the ear recognizes structures that are not at the level of individual isolated notes, but to understand melodies, a learning phase is required. The same goes for image analysis, where common features of specific faces are recognized quite quickly, for example, but recognizing different tree species requires a learning phase. What applies to hearing, vision, certainly applies to other domains [82], leading to the concept of a **dictionary** with learned elements and **sparse representation**.

2) The second axis that has been developed in the course and seminars is the connection with **neurophysiology**. We have seen this in the context of the auditory and visual systems, and it is very impressive to observe the similarities in architectures (computational/neurophysiological) uncovered by independent or mixed studies between the two communities. However, it is essential not to overstate the similarity: artificial neurons do not function the same way as biological neurons, the human brain is much more complex than an artificial neural network, etc. Nevertheless, the similarities are interesting enough to motivate further investigations.

3) Finally, there is another theme that we have not yet developed during the courses but will explore in the future, which is the connection with **Physics**. By nature, Physics was, until recently, the only **science of high dimension**. In particular, Statistical Physics deals with problems where Avogadro's number is the standard. This is the domain where mathematical models have been developed to understand high-dimensional dynamic systems. Interfaces are emerging recently on problems where Physics struggles: we mentioned Quantum Chemistry, and there is also the understanding of turbulence in fluid dynamics, which has remained an open problem since Kolmogorov's work in the 1940s. In the field of neural networks, possibilities for

82. Note: The same could be said for learning mathematical concepts.

simulating turbulence seem to be emerging, motivating joint research.

To conclude, the study of neural networks from a mathematical perspective remains essentially open. We do not yet have a complexity scale that would allow us, for example, to design an architecture (number of layers, type, number of neurons per layer, etc.) knowing the problem to be solved while being certain that it satisfies the posed question. We would like to have mathematical tools to capture the regularity/symmetry of functions in very high dimensions. Finally, we would like to have approximation theorems that guarantee the level of error and the stability of results (e.g., adversarial examples). S. Mallat points to an article [83] in which he reviews concepts developed in the course.

Next year, the envisaged theme is Sparsity.

83. https://www.di.ens.fr/~mallat/papiers/RSTA2015Published.pdf