

Emerging Directions in Deep Learning

Towards convolution-like operations in the non-Euclidean framework

Mihai-Sorin Stupariu, 2025-2026

CNNs: what is an image?

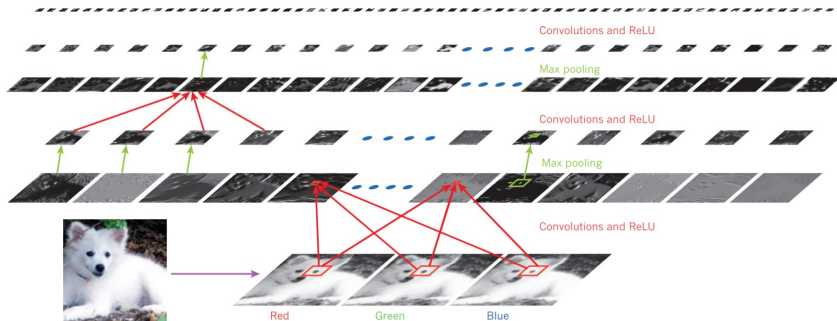


Figure 2 | Inside a convolutional network. The outputs (not the filters) of each layer (horizontally) of a typical convolutional network architecture applied to the image of a Samoyed dog (bottom left; and RGB (red, green, blue) inputs, bottom right). Each rectangular image is a feature map

corresponding to the output for one of the learned features, detected at each of the image positions. Information flows bottom up, with lower-level features acting as oriented edge detectors, and a score is computed for each image class in output. ReLU, rectified linear unit.

[LeCun et al., 2015]

What comes next?

Need to understand:

- techniques that do not work on manifolds (e.g. translation);
- what could be done for finding alternatives way of acting?

Look at <http://geometricdeeplearning.com/>; in particular the survey by Bronstein et al, 2017

Overview

	"Continuous "	"Discrete"
"Euclidean"		
"non-Euclidean"		

Overview

	“Continuous “	“Discrete”
“Euclidean”	(compact) Euclidean domain	regular grid
“non-Euclidean”		

Overview

	“Continuous “	“Discrete”
“Euclidean”	(compact) Euclidean domain	regular grid
“non-Euclidean”	manifold	triangle mesh (graph)

Setting - “continuous, Euclidean”

- ▶ Compact d -dimensional Euclidean domain $\Omega(=[0,1]^d) \subset \mathbb{R}^d$; the set of square-integrable functions $f \in L^2(\Omega)$

Setting - “continuous, Euclidean”

- ▶ Compact d -dimensional Euclidean domain $\Omega(=[0, 1]^d) \subset \mathbb{R}^d$; the set of square-integrable functions $f \in L^2(\Omega)$
- ▶ Ω in image analysis applications: take the square $X = [0, 1]^2$ and consider a lattice $\tilde{X}$ (the centers of a cells of a regular grid) derived from it. Then a grayscale image can be viewed as a map $f : \tilde{X} \rightarrow \mathbb{R}$. A color image can be regarded as a map $f : \tilde{X} \rightarrow \mathbb{R}^3$

Setting - “continuous, Euclidean”

- ▶ Compact d -dimensional Euclidean domain $\Omega = [0, 1]^d \subset \mathbb{R}^d$; the set of square-integrable functions $f \in L^2(\Omega)$
- ▶ Ω in image analysis applications: take the square $X = [0, 1]^2$ and consider a lattice $\tilde{X}$ (the centers of a cells of a regular grid) derived from it. Then a grayscale image can be viewed as a map $f : \tilde{X} \rightarrow \mathbb{R}$. A color image can be regarded as a map $f : \tilde{X} \rightarrow \mathbb{R}^3$
- ▶ Supervised learning setting: unknown function $y : L^2(\Omega) \rightarrow \mathcal{Y}$ observed on a training set

$$\{f_i \in L^2(\Omega) | y_i = y(f_i)\}_{i \in \mathcal{I}}$$

Setting - “continuous, Euclidean”

- ▶ Compact d -dimensional Euclidean domain $\Omega = [0, 1]^d \subset \mathbb{R}^d$; the set of square-integrable functions $f \in L^2(\Omega)$
- ▶ Ω in image analysis applications: take the square $X = [0, 1]^2$ and consider a lattice $\tilde{X}$ (the centers of a cells of a regular grid) derived from it. Then a grayscale image can be viewed as a map $f : \tilde{X} \rightarrow \mathbb{R}$. A color image can be regarded as a map $f : \tilde{X} \rightarrow \mathbb{R}^3$
- ▶ Supervised learning setting: unknown function $y : L^2(\Omega) \rightarrow \mathcal{Y}$ observed on a training set

$$\{f_i \in L^2(\Omega) | y_i = y(f_i)\}_{i \in \mathcal{I}}$$

- ▶ The target space $\mathcal{Y}$ can be discrete (classification), a simplex (representing the posterior class probabilities, in multiple object recognition) or $\mathbb{R}^m$ (regression tasks).

CNNs - formalization

Convolutional layer: has the form $g = C_{\Gamma, \xi}(f)$, where

CNNs - formalization

Convolutional layer: has the form $g = C_{\Gamma, \xi}(f)$, where

- $f = (f_1, \dots, f_p)$ is the p -dimensional input; $\forall l' \ f_{l'} \in L^2(\Omega)$
- $g = (g_1, \dots, g_q)$ is the q -dimensional output $\forall l \ g_l \in L^2(\Omega)$

CNNs - formalization

Convolutional layer: has the form $g = C_{\Gamma, \xi}(f)$, where

- $f = (f_1, \dots, f_p)$ is the p -dimensional input; $\forall l' \ f'_l \in L^2(\Omega)$
- $g = (g_1, \dots, g_q)$ is the q -dimensional output $\forall l \ g_l \in L^2(\Omega)$
- $\Gamma = (\gamma_{l,l'})_{l=1, \dots, q, l'=1, \dots, p}$ is a bank of filters; $\forall l, l' \ \gamma_{l,l'} \in L^2(\Omega)$
- $\xi : \mathbb{R} \rightarrow \mathbb{R}$ is a pointwise non-linearity (e.g. ReLU)

$$g_l(x) = \xi \left(\sum_{l'=1}^p (f'_{l'} \star \gamma_{l,l'})(x) \right), \text{ where} \quad (1)$$

$$(f \star \gamma)(x) = \int_{\Omega} f(x - x') \gamma(x') dx' \quad (2)$$

CNNs - formalization

Convolutional layer: has the form $g = C_{\Gamma, \xi}(f)$, where

- $f = (f_1, \dots, f_p)$ is the p -dimensional input; $\forall l' f_{l'} \in L^2(\Omega)$
- $g = (g_1, \dots, g_q)$ is the q -dimensional output $\forall l g_l \in L^2(\Omega)$
- $\Gamma = (\gamma_{l,l'})_{l=1, \dots, q, l'=1, \dots, p}$ is a bank of filters; $\forall l, l' \gamma_{l,l'} \in L^2(\Omega)$
- $\xi : \mathbb{R} \rightarrow \mathbb{R}$ is a pointwise non-linearity (e.g. ReLU)

$$g_l(x) = \xi \left(\sum_{l'=1}^p (f_{l'} \star \gamma_{l,l'})(x) \right), \text{ where} \quad (1)$$

$$(f \star \gamma)(x) = \int_{\Omega} f(x - x') \gamma(x') dx' \quad (2)$$

Downsampling (pooling) layer: has the form $b = P(a)$, where

CNNs - formalization

Convolutional layer: has the form $g = C_{\Gamma, \xi}(f)$, where

- $f = (f_1, \dots, f_p)$ is the p -dimensional input; $\forall l' f_{l'} \in L^2(\Omega)$
- $g = (g_1, \dots, g_q)$ is the q -dimensional output $\forall l g_l \in L^2(\Omega)$
- $\Gamma = (\gamma_{l,l'})_{l=1, \dots, q, l'=1, \dots, p}$ is a bank of filters; $\forall l, l' \gamma_{l,l'} \in L^2(\Omega)$
- $\xi : \mathbb{R} \rightarrow \mathbb{R}$ is a pointwise non-linearity (e.g. ReLU)

$$g_l(x) = \xi \left(\sum_{l'=1}^p (f_{l'} \star \gamma_{l,l'})(x) \right), \text{ where} \quad (1)$$

$$(f \star \gamma)(x) = \int_{\Omega} f(x - x') \gamma(x') dx' \quad (2)$$

Downsampling (pooling) layer: has the form $b = P(a)$, where

- a, b are again $*$ -tuples of functions in $L^2(\Omega)$
- P is a permutation-invariant function (such as the L^p -norm)

$$b_l(x) = P(\{a_l(x') | x' \in \mathcal{N}(x)\}), \quad \forall l, \quad \mathcal{N} \text{ neighborhood of } x \quad (3)$$

CNNs - formalization

Convolutional layer: has the form $g = C_{\Gamma, \xi}(f)$, where

- $f = (f_1, \dots, f_p)$ is the p -dimensional input; $\forall l' \ f'_l \in L^2(\Omega)$
- $g = (g_1, \dots, g_q)$ is the q -dimensional output $\forall l \ g_l \in L^2(\Omega)$
- $\Gamma = (\gamma_{l,l'})_{l=1, \dots, q, l'=1, \dots, p}$ is a bank of filters; $\forall l, l' \ \gamma_{l,l'} \in L^2(\Omega)$
- $\xi : \mathbb{R} \rightarrow \mathbb{R}$ is a pointwise non-linearity (e.g. ReLU)

$$g_l(x) = \xi \left(\sum_{l'=1}^p (f'_{l'} \star \gamma_{l,l'})(x) \right), \text{ where} \quad (1)$$

$$(f \star \gamma)(x) = \int_{\Omega} f(x - x') \gamma(x') dx' \quad (2)$$

Downsampling (pooling) layer: has the form $b = P(a)$, where

- a, b are again $*$ -tuples of functions in $L^2(\Omega)$
- P is a permutation-invariant function (such as the L^p -norm)

$$b_l(x) = P(\{a_l(x') | x' \in \mathcal{N}(x)\}), \quad \forall l, \quad \mathcal{N} \text{ neighborhood of } x \quad (3)$$

Convolutional network: composition of convolutional / pooling layers

$$U_{\Theta}(f) = (C_{\Gamma(S)} \circ \dots \circ P \circ \dots \circ C_{\Gamma(1)})(f) \quad (4)$$

CNNs - formalization

Convolutional layer: has the form $g = C_{\Gamma, \xi}(f)$, where

- $f = (f_1, \dots, f_p)$ is the p -dimensional input; $\forall l' f_{l'} \in L^2(\Omega)$
- $g = (g_1, \dots, g_q)$ is the q -dimensional output $\forall l g_l \in L^2(\Omega)$
- $\Gamma = (\gamma_{l,l'})_{l=1, \dots, q, l'=1, \dots, p}$ is a bank of filters; $\forall l, l' \gamma_{l,l'} \in L^2(\Omega)$
- $\xi : \mathbb{R} \rightarrow \mathbb{R}$ is a pointwise non-linearity (e.g. ReLU)

$$g_l(x) = \xi \left(\sum_{l'=1}^p (f_{l'} \star \gamma_{l,l'})(x) \right), \text{ where} \quad (1)$$

$$(f \star \gamma)(x) = \int_{\Omega} f(x - x') \gamma(x') dx' \quad (2)$$

Downsampling (pooling) layer: has the form $b = P(a)$, where

- a, b are again $*$ -tuples of functions in $L^2(\Omega)$
- P is a permutation-invariant function (such as the L^p -norm)

$$b_l(x) = P(\{a_l(x') | x' \in \mathcal{N}(x)\}), \quad \forall l, \quad \mathcal{N} \text{ neighborhood of } x \quad (3)$$

Convolutional network: composition of convolutional / pooling layers

$$U_{\Theta}(f) = (C_{\Gamma(S)} \circ \dots \circ P \circ \dots \circ C_{\Gamma(1)})(f) \quad (4)$$

Supervised learning task: minimize a cost function L on the training set

$$\min_{\Theta} \sum_{i \in \mathcal{I}} L(U_{\Theta}(f_i), y_i)$$

CNNs - formalization

Convolutional layer: has the form $g = C_{\Gamma, \xi}(f)$, where

- $f = (f_1, \dots, f_p)$ is the p -dimensional input; $\forall l' f_{l'} \in L^2(\Omega)$
- $g = (g_1, \dots, g_q)$ is the q -dimensional output $\forall l g_l \in L^2(\Omega)$
- $\Gamma = (\gamma_{l,l'})_{l=1, \dots, q, l'=1, \dots, p}$ is a bank of filters; $\forall l, l' \gamma_{l,l'} \in L^2(\Omega)$
- $\xi : \mathbb{R} \rightarrow \mathbb{R}$ is a pointwise non-linearity (e.g. ReLU)

$$g_l(x) = \xi \left(\sum_{l'=1}^p (f_{l'} \star \gamma_{l,l'})(x) \right), \text{ where} \quad (1)$$

$$(f \star \gamma)(x) = \int_{\Omega} f(x - x') \gamma(x') dx' \quad (2)$$

Downsampling (pooling) layer: has the form $b = P(a)$, where

- a, b are again $*$ -tuples of functions in $L^2(\Omega)$
- P is a permutation-invariant function (such as the L^p -norm)

$$b_l(x) = P(\{a_l(x') | x' \in \mathcal{N}(x)\}), \quad \forall l, \quad \mathcal{N} \text{ neighborhood of } x \quad (3)$$

Convolutional network: composition of convolutional / pooling layers

$$U_{\Theta}(f) = (C_{\Gamma(S)} \circ \dots \circ P \circ \dots \circ C_{\Gamma(1)})(f) \quad (4)$$

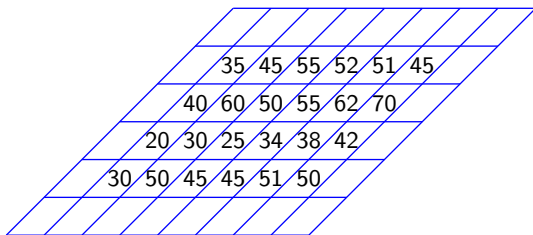
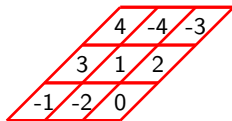
Supervised learning task: minimize a cost function L on the training set

$$\min_{\Theta} \sum_{i \in \mathcal{I}} L(U_{\Theta}(f_i), y_i)$$

Key ingredients / assumptions

- if $\Omega \subset \mathbb{R}^d$, one has the shift invariance (equivariance), due to

$$(f \star \gamma)(x + t) = \int_{\Omega} f(x + t - x') \gamma(x') dt = (f_t \star \gamma)(x), \quad f_t(z) = f(x + z)$$



- behavior w.r.t. local deformation

Possible approach

- ▶ Instead of defining the convolution operation, one could try to use one of its properties

Possible approach

- ▶ Instead of defining the convolution operation, one could try to use one of its properties
- ▶ Ideas?

Possible approach

- ▶ Instead of defining the convolution operation, one could try to use one of its properties
- ▶ Ideas?
- ▶ One could try to define the convolution operation with respect to a basis, but the definition would be basis-dependent

Possible approach

- ▶ Instead of defining the convolution operation, one could try to use one of its properties
- ▶ Ideas?
- ▶ One could try to define the convolution operation with respect to a basis, but the definition would be basis-dependent
- ▶ Recall that $L^2(\Omega)$ is a Hilbert space, it can be endowed with an inner product, $\langle f, g \rangle = \int_{\Omega} f(x)g(x)dx$. There exists a “particular” orthogonal basis, namely the one given by the *Laplace operator*

The Laplace operator and the Convolution Theorem

- In the “continuous” Euclidean set-up, the Laplace operator is given by $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$, its discrete counterpart is defined by the matrix

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

The Laplace operator and the Convolution Theorem

- ▶ In the “continuous” Euclidean set-up, the Laplace operator is given by $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$, its discrete counterpart is defined by the matrix
$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$
- ▶ The Laplacian operator Δ on $L^2(\Omega)$ is self-adjoint and positive definite. Under certain assumptions it admits non-negative real eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \dots$ (the *spectrum* of the Laplacian) and a set of orthonormal eigenfunctions $\phi_0, \phi_1, \dots$ verifying

$$\Delta \phi_i = \lambda_i \phi_i, \quad \forall i$$

The Laplace operator and the Convolution Theorem

- ▶ In the “continuous” Euclidean set-up, the Laplace operator is given by

$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$, its discrete counterpart is defined by the matrix

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

- ▶ The Laplacian operator Δ on $L^2(\Omega)$ is self-adjoint and positive definite. Under certain assumptions it admits non-negative real eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \dots$ (the *spectrum* of the Laplacian) and a set of orthonormal eigenfunctions $\phi_0, \phi_1, \dots$ verifying

$$\Delta \phi_i = \lambda_i \phi_i, \quad \forall i$$

- ▶ Further, under the same assumptions, a function $f \in L^2(\Omega)$ admits a decomposition into Fourier series

$$f(x) = \sum_{i \geq 0} \langle f, \phi_i \rangle \phi_i(x)$$

This is a decomposition of f with respect to an orthonormal basis

The Laplace operator and the Convolution Theorem

- ▶ In the “continuous” Euclidean set-up, the Laplace operator is given by

$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$, its discrete counterpart is defined by the matrix

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

- ▶ The Laplacian operator Δ on $L^2(\Omega)$ is self-adjoint and positive definite. Under certain assumptions it admits non-negative real eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \dots$ (the *spectrum* of the Laplacian) and a set of orthonormal eigenfunctions $\phi_0, \phi_1, \dots$ verifying

$$\Delta \phi_i = \lambda_i \phi_i, \quad \forall i$$

- ▶ Further, under the same assumptions, a function $f \in L^2(\Omega)$ admits a decomposition into Fourier series

$$f(x) = \sum_{i \geq 0} \langle f, \phi_i \rangle \phi_i(x)$$

This is a decomposition of f with respect to an orthonormal basis

- ▶ The Convolution Theorem

$$(f \star g)(x) = \sum_{i \geq 0} \langle f, \phi_i \rangle \langle g, \phi_i \rangle \phi_i(x)$$

(5)

Ideas...

- ▶ Construct a Laplace operator on manifolds and on triangle meshes (graphs)

Ideas...

- ▶ Construct a Laplace operator on manifolds and on triangle meshes (graphs)
- ▶ Replace the convolution operation (2) with a suitable operation and then apply similar mechanisms

On a Riemannian manifold (M, g) : the tangent bundle

- ▶ (M, g) a n -dimensional Riemannian manifold; for each $x \in M$ one has the tangent space $T_x M$ at x (a vector space of dimension n)

On a Riemannian manifold (M, g) : the tangent bundle

- ▶ (M, g) a n -dimensional Riemannian manifold; for each $x \in M$ one has the tangent space $T_x M$ at x (a vector space of dimension n)
- ▶ One defines the **tangent bundle** TM of M as the disjoint union of all tangent spaces

On a Riemannian manifold (M, g) : the tangent bundle

- ▶ (M, g) a n -dimensional Riemannian manifold; for each $x \in M$ one has the tangent space $T_x M$ at x (a vector space of dimension n)
- ▶ One defines the **tangent bundle** TM of M as the disjoint union of all tangent spaces
- ▶ The tangent bundle is itself a Riemannian manifold of dimension $2n$

On a Riemannian manifold (M, g) : the tangent bundle

- ▶ (M, g) a n -dimensional Riemannian manifold; for each $x \in M$ one has the tangent space $T_x M$ at x (a vector space of dimension n)
- ▶ One defines the **tangent bundle** TM of M as the disjoint union of all tangent spaces
- ▶ The tangent bundle is itself a Riemannian manifold of dimension $2n$
- ▶ A vector field V is a (smooth) map $V : M \rightarrow TM$; a scalar field is a smooth map $V : M \rightarrow \mathbb{R}$

On a Riemannian manifold (M, g) : the tangent bundle

- ▶ (M, g) a n -dimensional Riemannian manifold; for each $x \in M$ one has the tangent space $T_x M$ at x (a vector space of dimension n)
- ▶ One defines the **tangent bundle** TM of M as the disjoint union of all tangent spaces
- ▶ The tangent bundle is itself a Riemannian manifold of dimension $2n$
- ▶ A vector field V is a (smooth) map $V : M \rightarrow TM$; a scalar field is a smooth map $V : M \rightarrow \mathbb{R}$
- ▶ More general: the concept of **fiber bundle** (special cases: vector bundles, principal bundles) - [implemented in Geomstats](#). Methods relying on bundles can be useful for various ML tasks ([Smola et al., 2007](#), [Sommer and Bronstein, 2020](#)).

On a Riemannian manifold (M, g) : the Laplace operator

- ▶ The metric g yields a volume form, allowing the definition of the integral of scalar fields over M (assumed to be compact): $\int_M f(x) dx$.

On a Riemannian manifold (M, g) : the Laplace operator

- ▶ The metric g yields a volume form, allowing the definition of the integral of scalar fields over M (assumed to be compact): $\int_M f(x) dx$.
- ▶ One defines the Hilbert spaces of scalar and vector fields, denoted by $L^2(M), L^2(TM)$ with the inner products

$$\langle f, g \rangle_{L^2(M)} = \int_M f(x) g(x) dx, \quad \langle V, W \rangle_{L^2(TM)} = \int_M \langle V(x), W(x) \rangle_{T_x M} dx$$

On a Riemannian manifold (M, g) : the Laplace operator

- ▶ The metric g yields a volume form, allowing the definition of the integral of scalar fields over M (assumed to be compact): $\int_M f(x) dx$.
- ▶ One defines the Hilbert spaces of scalar and vector fields, denoted by $L^2(M), L^2(TM)$ with the inner products

$$\langle f, g \rangle_{L^2(M)} = \int_M f(x) g(x) dx, \quad \langle V, W \rangle_{L^2(TM)} = \int_M \langle V(x), W(x) \rangle_{T_x M} dx$$

- ▶ One defines the **intrinsic gradient** $\nabla : L^2(M) \rightarrow L^2(TM)$; it verifies the relation

$$df(x) = \langle \nabla f(x), \cdot \rangle_{T_x M},$$

where $df : TM \rightarrow \mathbb{R}$ is the differential of f and it acts on tangent vector fields

On a Riemannian manifold (M, g) : the Laplace operator

- ▶ The metric g yields a volume form, allowing the definition of the integral of scalar fields over M (assumed to be compact): $\int_M f(x) dx$.
- ▶ One defines the Hilbert spaces of scalar and vector fields, denoted by $L^2(M), L^2(TM)$ with the inner products

$$\langle f, g \rangle_{L^2(M)} = \int_M f(x) g(x) dx, \quad \langle V, W \rangle_{L^2(TM)} = \int_M \langle V(x), W(x) \rangle_{T_x M} dx$$

- ▶ One defines the **intrinsic gradient** $\nabla : L^2(M) \rightarrow L^2(TM)$; it verifies the relation

$$df(x) = \langle \nabla f(x), \cdot \rangle_{T_x M},$$

where $df : TM \rightarrow \mathbb{R}$ is the differential of f and it acts on tangent vector fields

- ▶ The **divergence** $\operatorname{div} : L^2(TM) \rightarrow L^2(M)$ is the formal adjoint of the gradient

On a Riemannian manifold (M, g) : the Laplace operator

- ▶ The metric g yields a volume form, allowing the definition of the integral of scalar fields over M (assumed to be compact): $\int_M f(x) dx$.
- ▶ One defines the Hilbert spaces of scalar and vector fields, denoted by $L^2(M), L^2(TM)$ with the inner products

$$\langle f, g \rangle_{L^2(M)} = \int_M f(x) g(x) dx, \quad \langle V, W \rangle_{L^2(TM)} = \int_M \langle V(x), W(x) \rangle_{T_x M} dx$$

- ▶ One defines the **intrinsic gradient** $\nabla : L^2(M) \rightarrow L^2(TM)$; it verifies the relation

$$df(x) = \langle \nabla f(x), \cdot \rangle_{T_x M},$$

where $df : TM \rightarrow \mathbb{R}$ is the differential of f and it acts on tangent vector fields

- ▶ The **divergence** $\operatorname{div} : L^2(TM) \rightarrow L^2(M)$ is the formal adjoint of the gradient
- ▶ The **Laplace-Beltrami operator** $\Delta : L^2(M) \rightarrow L^2(M)$ is defined as

$$\Delta : L^2(M) \rightarrow L^2(M), \quad \Delta f = -\operatorname{div}(\nabla f)$$

On a Riemannian manifold (M, g) : the Laplace operator

- ▶ The metric g yields a volume form, allowing the definition of the integral of scalar fields over M (assumed to be compact): $\int_M f(x) dx$.
- ▶ One defines the Hilbert spaces of scalar and vector fields, denoted by $L^2(M), L^2(TM)$ with the inner products

$$\langle f, g \rangle_{L^2(M)} = \int_M f(x) g(x) dx, \quad \langle V, W \rangle_{L^2(TM)} = \int_M \langle V(x), W(x) \rangle_{T_x M} dx$$

- ▶ One defines the **intrinsic gradient** $\nabla : L^2(M) \rightarrow L^2(TM)$; it verifies the relation

$$df(x) = \langle \nabla f(x), \cdot \rangle_{T_x M},$$

where $df : TM \rightarrow \mathbb{R}$ is the differential of f and it acts on tangent vector fields

- ▶ The **divergence** $\operatorname{div} : L^2(TM) \rightarrow L^2(M)$ is the formal adjoint of the gradient
- ▶ The **Laplace-Beltrami operator** $\Delta : L^2(M) \rightarrow L^2(M)$ is defined as

$$\Delta : L^2(M) \rightarrow L^2(M), \quad \Delta f = -\operatorname{div}(\nabla f)$$

- ▶ The Laplace operator is self-adjoint (symmetric) and positive definite. **The theory for the Euclidean case can be extended to a Riemannian manifold M , in particular one could define a convolution operation for scalar fields.**

Discussion about applications on manifolds will continue later.

In the sequel: the 'discrete' analogous (graphs). This is an important case and will be treated separately - see 07_CNNs_Graphs.