Alma Mater Studiorum – Università di Bologna

in cotutela con Sorbonne Université

DOTTORATO DI RICERCA IN
MATEMATICA

Ciclo XXXII

Settore Concorsuale: 01/A3

Settore Scientifico Disciplinare: MAT/05

A METRIC MODEL OF THE VISUAL CORTEX

Presentata da: Noemi Montobbio

Coordinatore Dottorato

Fabrizio Caselli

Supervisori

Giovanna Citti

Alessandro Sarti

Co-Supervisore

Laurent Bonnasse-Gahot

Esame finale anno 2019

Contents

Introduction

The aim of this thesis is the development of a model for the geometry of the connectivity of the primary visual cortex (V1), by means of functional analysis tools on metric measure spaces. The metric structure proposed to describe V1's internal connections implements a notion of correlation between neurons, based on their feature selectivity: this results in a connectivity pattern that is directly induced by the local feature analysis performed by the cells. We then apply this model to insert biologically inspired connections in deep learning algorithms, to enhance their ability to perform pattern completion in image classification tasks.
The main novelty in our approach lies in its ability to recover global geometric properties of V1's functional architecture without imposing any parameterization or invariance, but rather by exploiting the local information naturally encoded in the behavior of single V1 neurons in presence of a visual stimulus.

V1 is the first cortical area which receives the visual signal from the retina, and it is the most studied and the best understood among the visual areas in the brain. The first, celebrated description of its geometry was provided by D. H. Hubel and T. N. Wiesel in the '60s [82], starting from the crucial finding that cortical neurons are not only sensitive to the intensity of the visual stimulus, but they also display a sharp selectivity to other features, such as orientation, scale, velocity. According to Hubel and Wiesel's model, every retinal location is associated to a whole set of cells of V1, sensitive to all the possible values of these variables, that are "engrafted" onto the positional map with a *finer subdivision* [81]. More precisely, the position variable is sampled onto the cortex at a coarser resolution with respect to the engrafted variables: as a consequence, at each *discrete* spatial location all values of these other variables are represented. For instance, if we denote by (x, y) the spatial coordinates in the retinal plane, the arrangement of the orientation preference variable w.r.t. them can be described through an *orientation map*

$$\Theta : \mathbb{R}^2 \to S^1. \tag{0.1}$$

These topographic maps typically contain regular regions as well as singular points, called *pinwheels*, around which all values of θ are arranged as the spokes of a wheel.

The first processing of a visual stimulus in V1 is performed by a class of neurons called *simple cells*, which act in a quite complicated way w.r.t. the retinal image. Using a largely simplified model, a visual stimulus can be represented as a function $I = I(x,y)$ defined on the retinal plane, and the action of a simple cell in presence of this stimulus can be modeled as a linear integral operator. Its associated kernel will be denoted here by $\psi = \psi(x,y)$, and will be called the classical *receptive profile* (RP) of the neuron. The RP of a V1 simple cell is typically very concentrated, i.e. it is supported onto a localized domain of the retina, also referred to as the *receptive field* (RF) of the neuron. Since the behavior of these neurons is essentially characterized by the linear filtering operation performed by their RPs, the set of simple cells is classically identified with a family $\{\psi_p\}_{p\in\mathscr{G}} \subseteq L^2(\mathbb{R}^2)$ of linear filters. For each p in $\mathscr{G}$, the filter ψ_p acts on an image I as follows:

$$O_{\psi_p}(I) := \int_{\mathbb{R}^2} \psi_p(x,y) I(x,y) dxdy.$$

We say that the family $\{\psi_p\}_p$ *lifts* the image I to a new function $\mathscr{G} \ni p \longmapsto O_{\psi_p}(I)$ defined on the set $\mathscr{G}$ indexing the family. In the following, we will refer to $\mathscr{G}$ as the *feature space* associated to the bank of filters: intuitively, each element $p \in \mathscr{G}$ encodes the features extracted by the corresponding filter ψ_p when it is applied to an image. In many classical models, the feature space is defined as a product space $\mathscr{G} = \mathbb{R}^2 \times \mathscr{F}$. The coordinates $(x,y) \in \mathbb{R}^2$ denote the center of the smallest ball containing the localized support of the filter (i.e. the point of the retina at which the profile is centered): this encodes the variable of *position*. The third index $\Phi \in \mathscr{F}$ parameterizes the other features extracted by the filters. This translates in a continuous setting the idea of engrafted variables: at each point (x,y), all values of Φ are represented.

A well-established model for the RPs of V1 simple cells is represented by a set of bi-dimensional Gabor filters [84, 48, 99]: the whole bank of filters $\{\psi_{x,y,\theta}\}_{x,y,\theta}$ is obtained by translations of $(x,y) \in \mathbb{R}^2$ and rotations of $\theta \in S^1$ of a *mother function*

$$\psi(u,v) = \exp\left(\frac{2\pi i u}{\lambda}\right) \exp\left(-\frac{u^2+v^2}{2\sigma^2}\right).$$

Thus, the corresponding feature space $\mathscr{G}$ is the rototranslation group $SE(2) = \mathbb{R}^2 \times S^1$, representing the retinal position (x,y) and the *orientation* θ. Note that this product representation contains no information about the topographic organization of the variable θ onto the positional map; one may insert this constraint by considering an orientation map Θ as in (0.6):

this defines a sub-family $\{\psi_{x,y,\Theta(x,y)}\}_{(x,y)\in\mathbb{R}^2}$ of the above bank of filters, with a more realistic bi-dimensional feature space.

The neural activity is known to propagate across V1 through intra-cortical connections, often referred to as *horizontal* (or *lateral*) in the sense that they link cells belonging to the same level of the hierarchy of visual areas. These have been investigated in a number of experiments [72, 125, 27] and were found to link neurons that are far apart in the positional map but sensitive to similar orientations. The spatial extent and the marked orientation specificity of such connections have led to the hypothesis that lateral connections may be a neurophysiological counterpart to those perceptual mechanisms leading to the integration of local features into contours, described by the Gestalt law of good continuation and investigated in several psychophysical experiments [83, 69, 61]. The results of the behavioral experiments carried out in [61] are summarized through the notion of *association field*, describing the strength of reciprocal influence between two perceived edge elements in terms of their relative position and orientation. Notably, it is experimentally clear that there is a relationship between the set of receptive profiles (sharply selective w.r.t. orientation and position) and the structure of connectivity linking them.

Over the past twenty years, a number of models were proposed that characterize the functional architecture of V1 through differential structures [89, 79, 121, 154, 41, 131, 1]. We refer to [39] for a review. The main idea, anticipated by J. Koenderink [89] and W. Hoffman [79] and further developed by J. Petitot and Y. Tondut [121], is to represent V1 as a fiber bundle whose basis is the space of positions and whose fiber contains the engrafted variables. This is typically endowed with a Lie group structure, often associated to a parameterized bank of filters. For instance, the space $SE(2)$ associated to a family of Gabor filters was taken into consideration in [41], where the authors define a sub-Riemannian structure which is invariant with respect to the group law; they describe the spreading of neural activity in V1 through the lateral connectivity by means of a propagation along the integral curves of this structure. This idea can in principle be replicated as long as the bank of filters modeling the RPs can be parameterized by a group. Yet, this condition is not verified for computational models where the filters may be obtained for instance through automatic learning procedures [117, 8]. It is nonetheless essential for a cortex model to describe the set of receptive profiles and the functional geometry of intra-cortical connections with strictly connected instruments.

In the present work, we show that such geometrical properties can indeed be recovered through a notion of *correlation* between RPs of simple cells, with or without the presence of a group structure. Specifically, we propose a model of V1 as a metric space whose structure

is induced directly by the shapes of such profiles. V1 is still represented by the feature space $\mathscr{G}$ associated to a bank of filters $\{\psi_p\}_{p\in\mathscr{G}}$, and the distance between two points $p_0, p \in \mathscr{G}$ is defined as

$$d(p,p_0) := \|\psi_p - \psi_{p_0}\|_{L^2(\mathbb{R}^2)}. \tag{0.2}$$

Therefore, the filters do not only provide a set of parameters on which one defines a geometric structure, but rather they contribute to the characterization of such a structure. We stress that this construction does not require any invariance or group structure onto the set $\mathscr{G}$: for example, note that the metric space would still be well defined even for a set of filters known numerically and parameterized by a list of indices. The distance d is naturally associated to the kernel

$$K(p,p_0) := Re\langle\psi_p,\psi_{p_0}\rangle_{L^2(\mathbb{R}^2)} = \left(\int_{\mathbb{R}^2}\psi_p(x,y)\,\overline{\psi_{p_0}(x,y)}\,dx\,dy\right). \tag{0.3}$$

Indeed, the squared distance can be written as

$$d^2(p,p_0) = K(p,p) + K(p_0,p_0) - 2K(p,p_0).$$

K expresses a notion of *correlation* between filters w.r.t. the metric. This is straightforward if $\|\psi_p\|^2 = t$ for every p. In this case, the kernel is obtained as $K(p,p_0) = t - \frac{d^2(p,p_0)}{2}$ and its value for a couple of points increases as they get closer w.r.t the distance.

In order to describe the long-range spreading of neural activity across V1, we propose to adapt to our setting the diffusion-based approach employed in many differential models. We first equip our metric space with the spherical Hausdorff measure μ associated to the distance d, thus defining the *cortical metric measure space* $(\mathscr{G},d,\mu)$. We then refer to the classical approach of K.-T. Sturm [137], which provides a general method to construct a diffusion process on a metric measure space (X,d,μ). This technique consists in defining a Dirichlet form on $L^2(X,\mu)$ whose associated positive self-adjoint operator has a heat kernel h_t admitting Gaussian estimates in terms of the distance d, provided that a *Measure Contraction Property* (MCP, see Definition 2.18 at page 43) holds on the space. In order to produce an explicit algorithm to compute the cortical connectivity associated to a general bank of filters (not necessarily known analytically) we propose to approximate the propagation along the horizontal connectivity through an iterative procedure based on the estimation of the heat kernel h_t in terms of the kernel in (0.8). Specifically, given a nonlinear activation function ν and a normalization operator N, we first consider the following local kernel around a starting

point p_0:

$$K_1^{p_0}(p) := N[\nu(K)](p, p_0).$$

We then construct a wider kernel through a mechanism of repeated integrations, as follows:

$$K_n^{p_0} := \int N[\nu(K)](p,q) \cdot K_{n-1}^{p_0}(q)\, d\mu(q).$$

We will provide an extensive analysis of our results for the example of a feature space determined by a family of Gabor filters. This is a case worth investigating for two reasons. First, it is convenient in terms of intuition and manageability, since the invariances of the feature space in this setting make it possible to perform some explicit calculations. Second, it links the present metric model to the differential approach: indeed, the distance function obtained in this case turns out to be locally equivalent to a Riemannian distance which approximates the sub-Riemannian structure defined on $\mathbb{R}^2 \times S^1$ in the model presented in [41]. In this case, we will present some numerical simulations comparing the propagations obtained respectively through a discretized heat equation associated to the Riemannian structure, and through repeated integrations against the kernel as mentioned before.

Another significant example is given by the case of a surface $\Sigma \subseteq \mathbb{R}^2 \times S^1$, obtained as the feature space defined by a sub-family of Gabor filters, in the limit case where the external metric on $\mathbb{R}^2 \times S^1$ becomes sub-Riemannian. This case provides a motivation behind the choice of the general setting of metric measure spaces: indeed, the distance induced on Σ as a metric subspace of $\mathbb{R}^2 \times S^1$ cannot be obtained as the intrinsic metric associated to the horizontal curves on the surface. We will prove that the resulting metric measure space satisfies the MCP: this allows to describe the horizontal connectivity associated e.g. to a surface

$$\Sigma = \{\theta = \Theta(x,y)\} \subseteq \mathbb{R}^2 \times S^1$$

defined by an orientation map Θ as in (0.6).

Further numerical simulations for the feature spaces associated to different banks of filters will be provided, including a bank of *endstopped* profiles, sensitive to the length of the oriented stimuli: in this case, the connectivity pattern obtained turns out to recover the property of curvature selectivity observed in these neurons [82, 51]. We also examined the case where the RPs of simple cells are represented by a bank of filters obtained through an unsupervised regression algorithm, as in [117]: even starting from a non-structured bank of numerically known filters, our construction leads to outcomes that are qualitatively consistent with neurophysiological and psychophysical experimental data. Our results suggest that *the geometry of the horizontal connections in V1 can indeed be obtained from the RPs*, regardless

of their parameterization.

The promising outcomes obtained even in the case of a non-structured bank of learned filters led us to the idea of applying our construction in the context of Convolutional Neural Networks (CNNs) for image classification. Specifically, we propose to modify the otherwise purely feedforward architecture of these algorithms by inserting biologically plausible lateral connections inspired by our kernel-based approach.

CNNs [63, 96] are a class of deep learning algorithms designed for image processing, inspired by Hubel and Wiesel's hierarchical model of the visual system [82], where translation invariance is enforced by means of local convolutional windows shifting over the spatial domain. These algorithms generally have a *feedforward* structure, whose output is the result of transforming the input through a cascade of subsequent operators, corresponding to the network's *layers*. For $l \in \{1,\dots,L\}$, the *activation* h_l of the l-th layer is typically obtained from h_{l-1} through convolution with a bank of filters ψ^l:

$$h_l = s_l(\psi^l * h_{l-1} + b_l), \tag{0.4}$$

where b_l is a further additive term called *bias*, and s_l is a nonlinear activation function applied pointwise. The filters ψ^l and the bias terms b_l are unknown, and they are determined through minimization of a *loss function*, expressing the distance between the generated output and its value known a priori on the data.
Despite the strong analogy with the feature extraction performed in the human visual system, there are significant differences in structure and functioning between CNN algorithms and biological mechanisms of object processing. Critically, CNNs turn out to be very unstable to local perturbations of contours [13], and their ability to recognize objects seems to rely heavily on local features, rather than on global shapes [29]. This may be at least partly due to their feedforward structure: in the human visual system, the global analysis of a visual scene is made possible by mechanisms of contour integration and figure-ground segregation which are probably anatomically implemented by intra-layer and feedback recurrent connections. This observation inspired some recent works, where *horizontal* recurrent mechanisms are inserted in CNN architectures and their effects on performance in pattern completion tasks are examined [139, 101, 135]. In particular, in [101] lateral connections of convolutional type are added to a standard CNN. In the resulting architecture, referred to as Recurrent CNN (RecCNN in the following), the horizontal connections are defined by *learned* filters: as such, they are determined by additional parameters that are independent of the feedforward filters,

and no geometrical prior apart from the convolutional structure is inserted.

Our contribution consists in the introduction of another variant of the CNN architecture, given by the inclusion of a biologically inspired geometrical constraint encoding a notion of correlation between convolutional filters. In each convolutional layer l we learn a family $\{\psi_k^l(\cdot - i, \cdot - j)\}_{(i,j,k)\in\mathscr{G}_l}$ of filters, and we apply the metric model described previously. In this way the set $\mathscr{G}_l$ of learned features is endowed with a metric space structure induced by the filters, and we can apply the associated diffusion kernel defined in (0.8). Our main claim is that the introduction of this prior allows the networks to *spontaneously* implement perceptual mechanisms of global shape analysis and pattern completion. To this end, we compared the proposed model with the corresponding pure CNN architecture on *generalization* tasks involving images corrupted by a variety of degradation types. That is, we trained the networks on a simple task of image classification, and analyzed their generalization ability by presenting them with perturbed images *in the testing phase*. We emphasize that the models were not shown any corrupted images during the training stage, therefore their classification performance on these data only depends on the representations that they learn on unperturbed data.

As a first exploratory test, we considered a standard CNN architecture and inserted an *implicit* geometric prior by only adding to the loss function a regularization term, expressed as a squared gradient in the metric space defined earlier. The loss function obtained in this way, as sum of a gradient term and a similarity term, can be considered as a typical functional of calculus of variations. This led to an improved generalization ability of the network for images subject to perturbations not affecting their topology. Yet, it proved insufficient in the case of images corrupted by occlusions. We then enhanced such model by introducing a new architecture, which will be referred to as KerCNN: this is obtained from a base CNN architecture by inserting lateral connections defined through an iterative procedure, where the kernels K_l act as "transition kernels" onto the activations of the convolutional layers. The update rule for the l-th layer is defined as follows:

$$\begin{cases} h_l^1 = s_l(\psi^l * h_{l-1}^{T_{l-1}} + b_l) \\ h_l^t = \frac{1}{2}\left(K_l * h_l^{t-1} + h_l^{t-1}\right) \quad \text{for } 1 < t \leq T_l, \end{cases} \tag{0.5}$$

where h_l^t represents the activation of the l-th layer at the step t of the iteration. The first step in (0.10) implements the standard lifting operation of convolutional layers, as in (0.9). In subsequent steps, this activation is then averaged with an updated version of it obtained through convolution with the kernel K_l, until a stopping time $t = T_l$ is reached. The proposed

recursive formula is similar to the one presented in [101, 135], although carefully modified to implement a biologically plausible propagation of neural activity. Most importantly, the lateral kernels themselves are not learned, but rather they are constructed to allow diffusion in the metric defined by the learned filters. In particular, they establish a link between the geometrical properties of feedforward connections and horizontal connectivity, being defined as a function of the convolutional filters. This also implies that such kernels do not depend on any additional trainable parameters: therefore, their insertion does not increase the original network's complexity in terms of number of parameters, which allows a fair comparison in performance.

We shall give a complete report of our results for the popular MNIST digit classification dataset [97]. We will fix a base CNN with 2 convolutional layers, and define for each combination of stopping times (T_1, T_2) the corresponding KerCNN architecture. The classification accuracies achieved by the resulting models on corrupted testing images are then systematically compared. From our analysis it emerges that the KerCNN models with appropriate stopping times largely outperform the corresponding base CNN in classification accuracy on images subject to a variety of degradations, while mantaining the same performance on unperturbed images. We also compared the CNN and KerCNN models with the RecCNN architecture obtained by adding recurrent connections to the base model as in [135] – where the number of parameters of the networks is matched by decreasing the size of feedforward filters, to compensate for the additional recurrent parameters: in particular, for each task we inspected the performance of the *best* KerCNN and RecCNN architectures (i.e. the ones with the optimal number of iterations), and our results show that our biologically inspired model outruns the recurrent one in practically all experiments. We will conclude the chapter by giving a synthetic account on the same study carried out on different datasets.

The thesis is organized as follows. The first chapter contains the necessary anatomical and psychophysical background on the visual pathways, as well as a review of the state of the art for what concerns the mathematical modeling of the functional architecture of the visual cortex. In the second chapter, we present our connectivity model, corresponding to the content of [109, 110]. Finally, the third chapter is devoted to its application to deep learning models for pattern completion. We developed these topics in a third article [108].

Résumé

L'objectif de cette thèse est le développement d'un modèle pour la géométrie de la connectivité du cortex visuel primaire (V1), au moyen d'instruments d'analyse fonctionnelle dans les espaces métriques mésurés. La structure métrique proposée pour la description des connexions horizontales de V1 implémente une notion de corrélation entre les neurones, basé sur leur sélectivité à certains attributs : cela donne une configuration de connectivité qui est directement induite par l'analyse locale effectuée par les cellules. Nous appliquons ensuite ce modèle en ajoutant des connexions biologiquement inspirées dans des algorithmes d'apprentissage profond, afin de renforcer leur robustesse aux dégradations (tels que des occlusions) dans la classification d'images.
La principale nouveauté de cette approche est sa capacité à retrouver des propriétés globales de l'architecture fonctionnelle de V1 sans imposer aucune paramétrisation ou invariance, mais plutôt en exploitant l'information locale naturellement codifiée dans le comportement de chaque neurone de V1 en présence d'un stimulus visuel.

V1 est la première aire corticale qui reçoit le signal visuel de la rétine. Elle est la plus étudiée et la mieux comprise des aires visuelles du cerveau. La première célèbre description de sa géométrie a été donnée par D. H. Hubel et T. N. Wiesel dans les années 1960 [82], basée sur la découverte cruciale que les neurones corticaux ne sont pas seulement sensibles à l'intensité du stimulus visuel, mais qu'ils montrent aussi une forte sélectivité à d'autres caractéristiques, telles que l'orientation, l'échelle, la vitesse. Selon le modèle de Hubel et Wiesel, chaque entour sur la rétine est associée à tout un ensemble de cellules de V1, sensibles à toutes les valeurs de ces variables, qui sont "greffées" sur la carte positionelle avec *une subdivision plus fine* [81] : cela signifie que la position est échantillonnée sur le cortex avec une résolution plus grossière par rapport à elles, de sorte qu'à chaque endroit *discret* toutes les valeurs possibles des variables greffées soient représentées. Par exemple, si nous dénotons (x,y) les cordonnées spatiales, la disposition de la variable d'orientation

préférentielle par rapport à elles peut être décrite par une *carte d'orientation*

$$\Theta : \mathbb{R}^2 \to S^1. \tag{0.6}$$

Typiquement, ces cartes topographiques contiennent des régions régulières ainsi que des points singuliers, appelés *pinwheels* (roues d'orientation), autour desquelles toutes les valeurs de θ sont disposées comme les rayons d'une roue.

La première analyse d'un stimulus visuel en V1 est effectuée par une classe de cellules appelés *neurones simples*, qui agissent d'une façon très compliquée par rapport à l'image rétinienne. En utilisant un modèle simplifié, une image peut être représenté par une fonction $I(x,y)$ définie sur le plan rétinien, et l'action d'un neurone simple en présence de I peut être modélisée comme un opérateur intégral linéaire. Son noyau associé sera dénoté avec $\psi(x,y)$ et appelé le *profil récepteur* (*receptive profile*, RP) de la cellule. Le RP d'un neurone simple de V1 est typiquement très concentré, c'est à dire il est supporté dans un domaine localisé de la rétine, également appelé le *champ récepteur* (*receptive field*, RF) du neurone. Puisque le comportement de ces neurones se caracterise essentiellement par l'opération de filtrage linéaire effectuée par leurs RPs, l'ensemble des neurones simples est classiquement identifié avec un banc $\{\psi_p\}_{p\in\mathscr{G}} \subseteq L^2(\mathbb{R}^2)$ de filtres linéaires, où $\mathscr{G}$ est un ensemble paramétrant la famille. Pour chaque p dans $\mathscr{G}$ le filtre ψ_p agit sur une image I comme suit :

$$O_{\psi_p}(I) := \int_{\mathbb{R}^2} \psi_p(x,y) I(x,y) dxdy.$$

Donc l'action du banc de filtres sur I produit une fonction $\mathscr{G} \ni p \longmapsto O_{\psi_p}(I)$ définie sur $\mathscr{G}$: nous dirons que la famille $\{\psi_p\}_p$ *relève* l'image I à $\mathscr{G}$. Dans ce qui suit, nous appelerons $\mathscr{G}$ l'espace des "traits" (*features*) associé au banc de filtres : intuitivement, chaque élément $p \in \mathscr{G}$ code les caractéristiques extraites par le filtre correspondant ψ_p quand il est appliqué à une image. Dans nombreux modèles classiques, l'espace des features est défini comme un espace produit $\mathscr{G} = \mathbb{R}^2 \times \mathscr{F}$: les cordonnées $(x,y) \in \mathbb{R}^2$ désignent le centre du disque le plus petit qui contient son RF localisé (c'est à dire le point de la rétine sur lequel le profil est centré), ce qui représente la variable de *position* ; le troisième indice $\Phi \in \mathscr{F}$ paramètre les autres attributs extraits par les filtres. Cela traduit dans un contexte continu l'idée des variables greffées : à chaque point (x,y), toutes les valeurs de Φ sont représentées.

Un modèle bien établi des RPs des neurones simples de V1 est donné par un ensemble de filtres de Gabor [84, 48, 99] : un tel banc de filtres $\{\psi_{x,y,\theta}\}_{x,y,\theta}$ est obtenu par translations de

$(x,y) \in \mathbb{R}^2$ et rotations de $\theta \in S^1$ d'une fonction *mère*

$$\psi(u,v) = \exp\left(\frac{2\pi i u}{\lambda}\right)\exp\left(-\frac{u^2+v^2}{2\sigma^2}\right).$$

Ainsi, l'espace de features $\mathscr{G}$ correspondant est le groupe $SE(2) = \mathbb{R}^2 \times S^1$ des déplacements du plan Euclidien, représentant la position rétinienne (x,y) et l'orientation θ. Notez que cette représentation produit ne contient pas d'informations sur l'organisation topographique de la variable θ sur la carte positionelle. Il est possible d'insérer cette contrainte en considerant une carte d'orientation Θ comme décrit par (0.6) : cela définit une sous-famille $\{\psi_{x,y,\Theta(x,y)}\}_{(x,y)\in\mathbb{R}^2}$ du banc de filtres ci-dessus, avec un espace de features bidimensionel plus réaliste.

On sait que l'activité neuronale se propage le long de V1 par des connexions intracorticales, souvent appelées *horizontales* (ou *latérales*) au sens où elles relient des cellules qui appartiennent au même niveau de la hiérarchie des zones visuelles. Elles ont été étudiées au moyen de nombreuses expériences [72, 125, 27], et il a été constaté qu'elles relient des neurones qui peuvent être très éloignés dans la carte positionelle, mais sensibles à des orientations similaires. L'étendue spatiale et la forte spécificité de ces connexions à l'égard de l'orientation ont conduit à proposer que les connexions latérales soient une contrepartie neurophysiologique de mécanismes perceptifs liés à l'intégration des données locales pour former des contours. Ceux-ci ont été décrits par le concept Gestaltiste de bonne continuation et étudiés dans différentes expériences psychophysiques [83, 69, 61]. Les résultats des expériences comportementales réalisées par [61] sont résumés dans la notion de *champ d'association*, qui décrit la force de l'influence réciproque entre la perception de deux éléments orientés locaux en fonction de leurs position et orientation relatives. Notamment, il est expérimentalement connu qu'il y a une relation entre l'ensemble des profils (sensibles à l'orientation et à la position) et la structure de la connectivité qui les relie.

Au cours des vingt dernières années, de nombreux modèles qui caractérisent l'architecture fonctionnelle de V1 à travers des structures différentielles ont été proposés : voir [89, 79, 121, 154, 41, 131, 1] ; nous faisons aussi référence à [39] pour un aperçu complet. L'idée principale, anticipée par J. Koenderink [89] et W. Hoffman [79] et développée par J. Petitot et Y. Tondut [121], est de représenter V1 comme un espace fibré ayant comme base l'espace des positions et dont la fibre contient les variables greffées. Cet espace est typiquement doté d'une structure de groupe de Lie, souvent associée à un banc de filtres paramétré. Par exemple, l'espace $SE(2)$ associé à une famille de filtres de Gabor a été adopté dans [41], où les auteurs y définissent une structure sous-Riemannienne invariante par rapport à la loi de

groupe. Ils décrivent la propagation de l'activité neuronale sur V1 au moyen d'une diffusion le long des courbes intégrales de cette structure. Cette idée peut en principe être reproduite dans la mesure où le banc de filtres modélisant les RPs peut être paramétré par un groupe. Pourtant, cette condition n'est pas satisfaite par des modèles computationnels où les filtres peuvent être obtenus par exemple à travers des procédures d'apprentissage automatique [117, 8]. Il est néanmoins essentiel pour un modèle de cortex de décrire l'ensemble des RPs et l'architecture fonctionnelle gouvernant la géométrie des connexions horizontales avec des instruments étroitement liés.

Dans le présent travail, nous montrons que de telles propriétés peuvent être en effet modélisées par une notion de *corrélation* entre les RPs des neurones simples, avec ou sans la présence d'une structure de groupe. Plus spécifiquement, nous proposons un modèle de V1 comme espace métrique, dont la structure est induite directement par les conformations de tels profils. V1 est toujours représenté par l'espace de features $\mathscr{G}$ associé à un banc de filtres $\{\psi_p\}_{p\in\mathscr{G}}$, et la distance entre deux points $p_0, p \in \mathscr{G}$ est définie par

$$d(p,p_0) := \|\psi_p - \psi_{p_0}\|_{L^2(\mathbb{R}^2)}. \tag{0.7}$$

En conséquence, non seulement les filtres fournissent un ensemble de paramètres sur lequel on peut définir une structure géométrique, mais également ils contribuent à la caractérisation d'une telle structure. Nous soulignons que cette construction ne nécessite aucune invariance ou loi de groupe sur l'ensemble $\mathscr{G}$: par exemple, notez que l'espace métrique serait encore bien défini même en partant d'un ensemble de filtres connus numériquement et paramétrés par une liste d'indices. La distance d est naturellement associée à un noyau

$$K(p,p_0) := Re\langle\psi_p,\psi_{p_0}\rangle_{L^2(\mathbb{R}^2)} = \left(\int_{\mathbb{R}^2} \psi_p(x,y)\,\overline{\psi_{p_0}(x,y)}\,dx\,dy\right). \tag{0.8}$$

En effet, la distance au carré peut être écrite comme

$$d^2(p,p_0) = K(p,p) + K(p_0,p_0) - 2K(p,p_0).$$

K exprime une notion de *corrélation* entre les filtres par rapport à la métrique. Cela est simple à voir si $\|\psi_p\|^2 = t$ pour chaque p : dans ce cas, le noyau est obtenu comme $K(p,p_0) = t - \frac{d^2(p,p_0)}{2}$ et sa valeur pour un couple de points augmente lorsqu'ils se rapprochent par rapport à la distance.

Afin de décrire la propagation à longue portée de l'activité neuronale sur V1, nous proposons d'adapter à notre contexte l'approche basée sur la diffusion utilisée dans les modèles différentiels. Nous commençons par équiper cet espace métrique avec la mesure de Hausdorff sphérique μ associée à la distance d, définant ainsi l'espace métrique de mesure $(\mathscr{G}, d, \mu)$. Nous adoptons ensuite l'approche classique de K.-T. Sturm [137], qui fournit une méthode générale pour construire un processus de diffusion sur un espace métrique de mesure (X, d, μ). Cette technique consiste à définir une forme de Dirichlet sur $L^2(X, \mu)$ dont l'opérateur autoadjoint positif associé admet un noyau de la chaleur h_t, pour lequel on a des estimations Gaussiennes par rapport à la distance d – à condition qu'une propriété appelée *Measure Contraction Property* (MCP, voir Definition 2.18 à page 43) soit satisfaite sur l'espace. De manière à produire un algorithme explicite pour calculer la configuration de connectivité associé à un banc de filtres général (pas forcément connu analytiquement), nous proposons d'approximer la propagation le long de la connectivité horizontale par une procédure itérative basée sur l'estimation du noyau de la chaleur h_t en fonction du noyau (0.8). Plus précisement, nous commençons par considérer le noyau local suivant autour d'un point de départ p_0 :

$$K_1^{p_0}(p) := N\left[\nu\left(K\right)\right](p, p_0),$$

où N est un opérateur de normalisation et ν est une fonction d'activation non linéaire. Nous construisons ensuite un noyau plus large au moyen d'un mécanisme d'intégrations répétées, comme suit :

$$K_n^{p_0} := \int N\left[\nu\left(K\right)\right](p, q) \cdot K_{n-1}^{p_0}(q)\, d\mu(q).$$

Nous allons présenter une analyse complète de nos résultats pour l'example d'un espace de features déterminé par une famille de filtres de Gabor. Cela est un cas qui mérite d'être examiné pour deux raisons. En premier lieu, il est pratique pour ce qui concerne l'intuition et la "tractabilité" mathématique, puisque les invariances de l'espace de features dans ce cadre permettent d'effectuer des calculs explicites. Ensuite, il relie le présent modèle métrique à l'approche différentielle : en effet, la fonction distance obtenue dans ce cas se révèle être localement équivalente à une distance Riemannienne, qui approxime la structure sous-Riemannienne définie sur $\mathbb{R}^2 \times S^1$ dans [41]. Dans ce cas, nous allons montrer des simulations numériques comparant les propagations obtenues respectivement par une équation de la chaleur discretisée associée à la structure Riemannienne, et par intégration répétée du noyau comme indiqué précédemment.

Un autre exemple significatif est donné par le cas d'une surface $\Sigma \subseteq \mathbb{R}^2 \times S^1$, obtenue comme espace de features défini par une sous-famille de filtres de Gabor, dans le cas limite où la métrique extérieure sur $\mathbb{R}^2 \times S^1$ devient sous-Riemannienne. Ce cas fournit une motivation

du choix d'un contexte général tel que celui des espace métriques de mesure : en effet, la distance induite sur Σ comme sous-espace métrique de $\mathbb{R}^2 \times S^1$ ne peut pas être obtenue comme métrique intrinsèque associée aux courbes horizontales sur la surface. Nous allons prouver que l'espace métrique de mesure résultant satisfait la MCP : cela permet de décrire la connectivité horizontale associée par exemple à une surface

$$\Sigma = \{\theta = \Theta(x,y)\} \subseteq \mathbb{R}^2 \times S^1$$

définie par une carte d'orientation Θ comme celle décrite par (0.6).

Nous allons ainsi montrer d'autres simulations numériques pour les espaces de features associés à différentes familles de filtres, y compris un banc de profils *endstopped*, c'est à dire sensibles à la longueur des stimuli orientés : dans ce cas, la configuration de connectivité obtenue se révèle capable de retrouver la propriété de sélectivité à la courbure observée sur ces neurones [82, 51]. Nous examinons également le cas où les RPs des neurones simples sont représentés par un banc de filtres obtenus par un algorithme de régression non supervisé, comme dans [117] : même en partant d'une famille non structurée de filtres connus numériquement, notre construction produit des résultats qui sont qualitativement compatibles avec les données expérimentales neurophysiologiques and psychophysiques. Nos résultats suggèrent que *la géométrie des connexions horizontales de V1 peut effectivement être obtenue à partir des RPs*, quelle que soit leur paramétrisation.

Les résultats encourageants obtenus même dans le cas d'un banc de filtres appris ont conduit à l'idée d'appliquer notre construction dans le contexte des *réseaux de neurones convolutifs* (CNNs, de l'anglais *Convolutional Neural Networks*) pour la classification d'images. Plus spécifiquement, nous proposons de modifier l'architecture purement hiérarchique de ces algorithmes, en introduisant des connexions latérales biologiquement plausibles, inspirées par notre approche par noyau.

Les CNNs [63, 96] sont une classe d'algorithmes d'apprentissage profond pour le traitement d'images, inspirés par le modèle hiérarchique du système visuel de Hubel et Wiesel [82], où l'invariance par translation est imposée par des fênetres de convolution glissantes sur le domaine spatiale. Ces modèles ont généralement une structure de *réseau à propagation avant* (*feedforward*), dont la sortie résulte de la transformation du signal d'entrée par une cascade d'opérateurs, correspondants aux *couches* du réseau. Pour $l \in \{1,\dots,L\}$, l'*activation* h_l de la couche l-ième est typiquement obtenue à partir de h_{l-1} par la convolution avec un banc de filtres ψ^l :

$$h_l = s_l(\psi^l * h_{l-1} + b_l), \tag{0.9}$$

où b_l est un terme additif appelé *biais*, et s_l est une fonction d'activation non linéaire, appliquée ponctuellement. Les filtres ψ^l et les biais b_l ne sont pas connus, et ils sont déterminés par la minimisation d'une fonction de *loss* qui exprime la distance entre l'input généré et sa valeur connue a priori sur les données.
Malgré la forte similitude avec l'extraction de caractéristiques effectuée dans le système visuel humain, il y a des différences significatives de structure et de fonctionnement entre les CNNs et les mécanismes biologiques de traitement d'objets. Notamment, les CNNs se révèlent très peu stables aux perturbations locales des contours [13], et leur capacité à reconnaitre les objets semble faire largement appel aux caractéristiques locales, plutôt qu'aux formes globales [29]. Cela peut être au moins partiellement dû à leur structure feedforward : dans le système visuel humain, l'analyse globale d'une scène visuelle est rendue possible grâce à des mécanismes d'intégration de contours et de ségrégation figure-fond, qui sont censés être anatomiquement implémentés par de connexions intracorticales et en feedback. Cette observation a inspiré de récents travaux, où des mécanismes récurrents *horizontales* ont été inserés dans l'architecture des CNNs, et leurs effets sur la performance des réseaux en présence d'images corrompues ont été examinés [139, 101, 135]. Notamment, dans [101], des connexions latérales de type convolutif sont ajoutées à un CNN de base. Dans l'architecture résultante, appelé CNN récurrent (RecCNN ci-après), les connexions horizontales sont définies par des filtres *appris* : par conséquent, elles sont déterminées par des paramètres supplémentaires qui sont indépendants des filtres feedforward, et aucune information géométrique n'est insérée au-delà de la structure convolutive.

Notre contribution consiste à introduire une autre variante d'un CNN, donnée par l'inclusion d'une contrainte biologiquement plausible qui code une notion de corrélation entre les filtres convolutifs. Dans chaque couche convolutive l une famille $\{\psi^l_k(\cdot - i, \cdot - j)\}_{(i,j,k)\in\mathscr{G}_l}$ de filtres est apprise, et nous appliquons le modèle métrique décrit précédemment. De cette manière, l'ensemble $\mathscr{G}_l$ des features est doté d'une structure d'espace métrique induite par les filtres, et nous pouvons appliquer le noyau de diffusion défini par (0.8). Notre hypothèse est que l'introduction de cet élement permette d'implémenter *spontanément* des mécanismes perceptifs d'analyse globale des formes et de complètion de contours (*contour completion*). À cette fin, nous avons comparé le modèle proposé avec l'architecture purement convolutive correspondante, sur des tâches de généralisation impliquant des images perturbés par une varieté de types de dégradation. C'est à dire, les réseaux ont été entraînés sur une simple tâche de classification d'images, et nous avons analysé leur capacité de généralisation en les évaluant sur des images perturbées en phase de test. Nous soulignons qu'aucune image corrompue n'a été montrée aux modèles lors de la phase d'entraînement, donc leur perfor-

mance de classification ne dépend que des représentations qu'ils ont appris sur les données non perturbées.

Dans un premier temps, nous avons considéré une architecture convolutive standard et inséré une contrainte géométrique uniquement par l'ajout dans la fonction de *loss* d'un terme de régularisation exprimé comme un gradient au carré dans l'espace métrique défini précédemment. La fonction de loss obtenue, somme d'un terme de gradient et d'un terme de similarité, peut être considérée comme un fonctionnel typique du calcul des variations. Cela a conduit à une meilleure capacité de généralisation du réseau pour images avec des perturbations qui ne changeaient pas leur topologie. Pourtant, il s'est avéré insuffisant dans le cas d'images corrompues par occlusions. Nous avons alors renforcé tel modèle en introduisant une nouvelle architecture, ci-après appelée KerCNN : elle est obtenue à partir d'un CNN de base en insérant des connexions latérales définies par une procédure itérative, où les noyaux agissent comme des "noyaux de transition" sur les activations des couches convolutives. La règle de mise à jour pour la l-ième couche est définie comme suit :

$$\begin{cases} h_l^1 = s_l(\psi^l * h_{l-1}^{T_{l-1}} + b_l) \\ h_l^t = \frac{1}{2}\left(K_l * h_l^{t-1} + h_l^{t-1}\right) \quad \text{pour } 1 < t \leq T_l, \end{cases} \tag{0.10}$$

où h_l^t représente l'activation de la l-ième couche au pas t de l'itération. Le premier pas dans (0.10) implémente l'opération standard de relèvement des couches convolutives, comme présenté dans (0.9). Aux pas suivants, cette activation est moyennée avec sa version "actualisée" obtenue par convolution avec le noyau K_l, jusqu'à un temps d'arrêt $t = T_l$. La formule récurrente proposée est proche à celle présentée dans [101, 135], mais modifiée attentivement pour implémenter une propagation de l'activité neuronale qui soit biologiquement plausible. Notamment, les noyaux latéraux eux-mêmes ne sont pas appris, mais il sont conçus pour générer une diffusion dans la métrique définie par les filtres appris. Notamment, ils établissent un lien entre les propriétés géométriques des connexions feedforward et horizontales, puisqu'ils sont définis comme fonction des filtres convolutifs. Cela implique également que de tels noyaux ne dépendent d'aucun paramètre supplémentaire, ce qui permet une comparaison équitable en termes de performance.

Nous fournissons un rapport complet de nos résultats sur MNIST [97], une base de données très célèbre pour la classification de chiffres manuscrits. Nous allons fixer un CNN de base avec 2 couches convolutives, et définir pour chaque combinaison de temps d'arrêt (T_1, T_2) le KerCNN correspondant. Les taux de bon classement atteints par les réseaux résultants sur les images de test corrompues sont ensuite systématiquement comparés. D'après notre analyse, il apparaît que les KerCNNs avec des temps d'arrêt appropriés surpassent

largement le CNN correspondant dans la classification d'images avec une varieté de dégradations, tout en gardant la même performance sur les images non perturbées. Nous avons aussi comparé nos KerCNNs avec les RecCNN obtenus per l'ajout de connexions récurrentes au modèle de base, comme dans [135] – où le nombre de paramètres des réseaux est ajusté en réduisant la taille des filtres feedforward, pour compenser l'ajout des paramètres récurrents. Notamment, pour chaque tâche nous avons examiné la performance des *meilleurs* KerCNN et RecCNN (c'est à dire ceux avec le nombre optimal d'itérations), et les résultats montrent que notre réseau inspiré biologiquement dépasse le modèle récurrent dans pratiquement toutes les expériences. Nous concluons le chapitre en donnant un rapport de synthèse sur la même étude effectuée sur différentes bases de données.

La thèse est organisée comme suit. Le premier chapitre contient le contexte anatomique et psychophysique nécessaire sur les voies visuelles, ainsi qu'une revue sur l'état de l'art pour ce qui concerne la modélisation mathématique de l'architecture fonctionnelle du cortex visuel. Dans le deuxième chapitre, nous presentons notre modèle de connectivité, correspondant au contenu des travaux [109, 110]. Pour terminer, le troisième chapitre est consacré à l'application de ce modèle à des algorithmes d'apprentissage profond pour l'analyse globale de formes. Nous avons développé ce sujet dans un troisième papier [108].

Chapter 1

Neurophysiology and psychophysics of vision

1.1 The early visual pathways

We start by giving some background on the main structures composing the early visual pathways, with a particular focus on the primary visual cortex.

The visual signal is first mapped onto the *retina*, from which it is conveyed through the optic

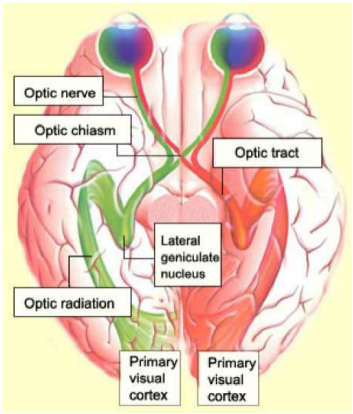


Figure 1.1 Structure of the visual pathways. Source: [120].

nerve to the *lateral geniculate nucleus* (LGN). This structure is the main central conjunction to the occipital lobe, in particular to the *primary visual cortex* (V1). From V1, different specialized parallel pathways depart, leading to higher cortical areas performing further processing. See Figure 1.1 for a schematic depiction.

1.1.1 Receptive fields and receptive profiles

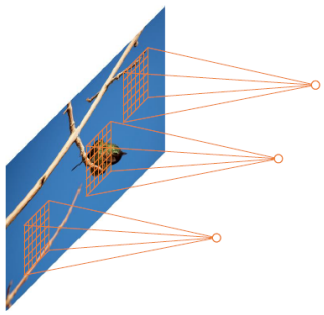


Figure 1.2 A schematic depiction of the concept of receptive field of a neuron.

Through the above-mentioned connections, each cell is linked to a specific domain D of the retina which is referred to as its *receptive field* (RF), as schematically depicted in Figure 1.2. A retinal cell in the RF of the neuron can react in an excitatory or in an inhibitory way to a luminous stimulation, with different modulation: this can be described through a function $\psi : D \to \mathbb{R}$, called the *receptive profile* (RP) of the cell, which measures its reaction to punctual stimuli located at every point $(x,y) \in D$. Positive values of ψ correspond to excitatory responses, and negative values correspond to inhibitory ones, while the absolute value of ψ represents their intensity. Neurons in the early visual pathways typically have very localized RFs, and their RPs are generally modeled as compactly supported or exponentially decaying functions.
Throughout the text, we will typically consider a visual stimulus to be an L^2 function defined onto the retina $R \subseteq \mathbb{R}^2$. In most cases these functions will be real-valued, meaning that we refer to grayscale stimuli – otherwise they may take values in $\mathbb{R}^3$, encoding the three RGB channels. However, it is perhaps more natural [120] to treat the signal (which may be a very noisy function) as a Schwartz distribution, i.e. a continuous linear functional onto a space of test functions. Note that this allows to include the description of a *punctual* stimulus located at a retinal point $(x_0,y_0) \in R$ as the Dirac delta $\delta_{(x_0,y_0)}$.

The response of certain types of visual neurons to an optic signal has been shown to be approximately *linear* with respect to the retinal image. In other words, a simplified model for the behavior of such a cell can be written through a linear operator O_ψ defined in terms of the profile ψ of the cell:

$$O_\psi(I) := \int_D I(x,y)\psi(x,y)dxdy. \tag{1.1}$$

This means that the RPs of these neurons not only provide information about the *impulse response* of the cell, but they also allow to model its reaction to a general (non-punctual) visual stimulus $I \in L^2(R)$. More generally, the integral in Eq. (1.1) can be seen in the sense of distributions as the representation $\langle I|\psi\rangle$. Note that, in the case of a punctual stimulus $I = \delta_{(x_0,y_0)}$, this gives $\psi(x_0,y_0)$ as a result.

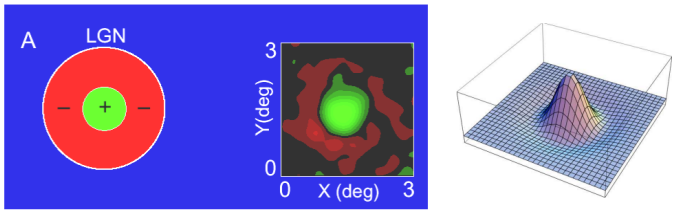


Figure 1.3 The receptive profile of an ON-center LGN cell. On the left, a schematic representation of the ON and OFF domains; in the middle,a visualization of the level sets of the RP. On the right, the "Laplacian of Gaussian" model for LGN RPs. Source: [120].

A first example of this linear behavior is given by the neurons of the LGN. It is a classic result of neurophysiology that the RPs of such neurons are best modeled as Laplacians of Gaussians. See for instance Figure 1.3, showing the RP of an "ON-center" LGN cell, i.e. one responding positively to punctual stimuli located in the central region of its RF, and negatively to those outside this region.

1.1.2 The primary visual cortex (V1)

Main types of neurons

As for V1, two main classes of cells can be observed in this area. These neurons are referred to as *simple* and *complex* cells, and they were first discovered by D. H. Hubel and T. N. Wiesel in the '60s [82]. While simple cells still exhibit an approximately linear behavior w.r.t. the retinal image, the response of complex cells cannot be represented through a linear operation.

Simple cells are the first neurons in the visual pathways showing orientation selectivity, given by a strongly anisotropic RP, as shown in Figure 1.4: this means that they respond strongly only to stimuli containing edges or contours aligned with the orientation of their elongated RP. These cells receive most of the outgoing projections from the LGN: it is presumed that each simple receptive field arises from multiple isotropic LGN receptive fields converging in a line [81, 144], as schematically depicted in Figure 1.5. The set of RPs of

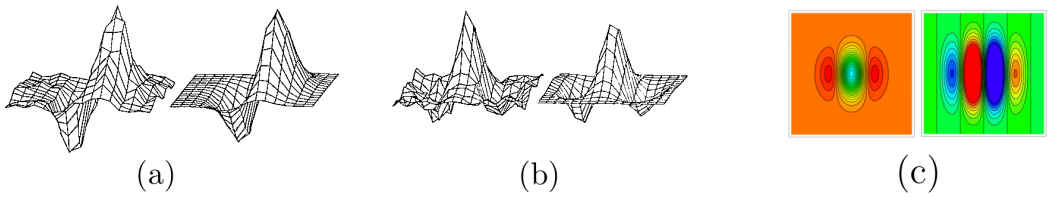


Figure 1.4 (a) Left: an example of experimentally measured odd RPs of simple cells in cat V1. Right: the best-fitting 2D Gabor function for the cell's RP. (b) The same as (a) for an even RP. Both examples are taken from [48]. (c) A quadrature pair of Gabor RPs, given by the real (left) and imaginary (right) parts of a complex Gabor function. Source: [120].

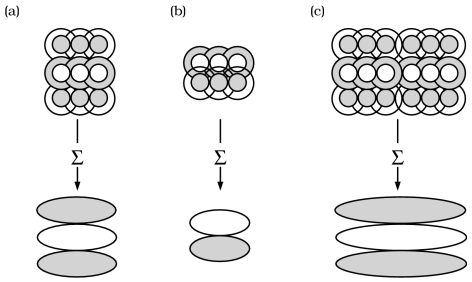


Figure 1.5 Orientation selective V1 RPs can be obtained by summing the responses of LGN non-oriented cells. Source: [144].

V1 simple cells has classically been modeled [84, 48, 99] through a bank of *Gabor filters* $\{\psi_{x,y,\theta}\}_{x,y,\theta}$. These are obtained from a *mother filter*

$$\psi_{0,0,0}(u,v) = \exp\left(\frac{2\pi i u}{\lambda}\right)\exp\left(-\frac{u^2+v^2}{2\sigma^2}\right), \tag{1.2}$$

by translations $T_{(x,y)}$ of $(x,y)\in\mathbb{R}^2$ and rotations R_θ of $\theta\in S^1$:

$$\psi_{x,y,\theta}(u,v) = \psi_{0,0,0}\left(T_{(x,y)}^{-1}R_\theta^{-1}(u,v)\right). \tag{1.3}$$

The standard deviation σ of the Gaussian function represents the *scale* of the profiles: the smaller the value of σ, the more concentrated the corresponding profile.

Note that these are complex-valued functions: each filter $\psi_{x,y,\theta}$ actually represents two RPs, given by its real and imaginary parts, sharing the same orientation but shifted by 90° in phase. These are referred to as a *quadrature pair* of cells. Real and imaginary parts of Gabor filters represent so-called *even* and *odd* cells respectively (see Figure 1.4b).
The family is indexed by $\mathbb{R}^2\times S^1$: $(x,y)\in\mathbb{R}^2$ encodes the position at which the filter is centered and $\theta\in S^1$ expresses its preferred orientation. Here we take the scale σ to be fixed, but that may be let vary as well, yielding a *multiscale* feature extraction.

Indeed, the shape of the RPs of these neurons contains information about the *features* that

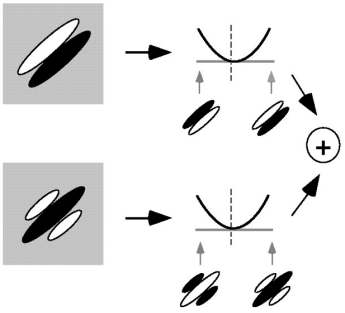


Figure 1.6 Development of phase-invariance through the energy model of a V1 complex cell.

they extract as *linear filters* on the visual signal. For instance, a profile ψ with a local support is sensitive to *position*, in the sense that the neuron only responds to stimuli in a localized region of the image. Or again, a profile with an elongated shape will be sensitive to a certain *orientation*. If we denote the whole set of RPs of simple cells by $\{\psi_p\}_{p\in\mathscr{G}}$, where $\mathscr{G}$ is a set of indices, we may intuitively regard each $p \in \mathscr{G}$ as the instance of such features to which ψ_p responds the most. In these terms, we shall refer to $\mathscr{G}$ as the *feature space* associated to the bank of filters $\{\psi_p\}_p$. According to this framework, in the classical model of Gabor filters the feature space is $\mathscr{G} = \mathbb{R}^2 \times S^1$.

The information extracted by simple cells is believed to determine the behavior of complex cells, which perform a second order analysis: in particular, according to the *energy model* [105], the response of each complex cell is modeled as the square sum of a quadrature pair of simple cells (see Figure 1.6). This leads to the *phase invariance* of these neurons, whose behavior cannot be described through linear filtering.

A further cell type was found within V1 in Hubel and Wiesel's studies: the neurons belonging to this third class were first referred to as *hypercomplex* and defined to be all those cells showing a more "intricate" behavior than simple and complex cells [82]. Hypercomplex cells displayed orientation selectivity, but confined to stimuli of a limited size: this property is referred to as *end-stopping*. However, this attribute was observed shortly thereafter in simple and complex neurons [70], which contradicted the existence of an independent class of hypercomplex cells. The notions *end-stopped simple* and *end-stopped complex* cells were then accepted in place of the preceding characterization of end-stopping as exclusive to a separate class.
The RPs of endstopped cells are characterized by antagonistic "end zones" suppressing the response to stimuli longer than the central excitatory area. There also exist endstopped

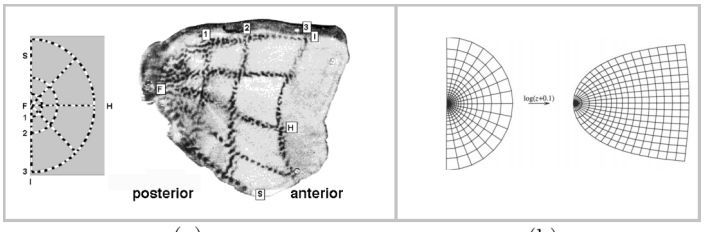
profiles which are only stopped at one end: only a stimulus that extends too far in one specific direction will stimulate the inhibitory region and reduce the strength of the neuron's response. Endstopped cells have also been shown [82] to react to curved stimuli; a mathematical description of the relation between endstopping and curvature has been developed in [51].

Retinotopy

The projection of a visual signal from the retina to the visual cortices is performed in a "continuous" fashion, so that adjacent spots on the retina are coded by adjacent neurons in these areas: this phenomenon is referred to as *retinotopy*, and it is one of the *topographic maps* observed in the visual areas. The deformation that a signal on the retina undergoes when it is represented on the cortex is referred to as the *retino-cortical mapping*, and it is shown to be well approximated, close to the center of the field of view, by the complex logarithmic function

$$\ell_{a,k}(z) = k \cdot \log(z+a).$$

The role of parameters a and k, changing in different animals, is to specify the fit [119]. Figure 1.7 displays a comparison between an experimental measurement of the mapping and



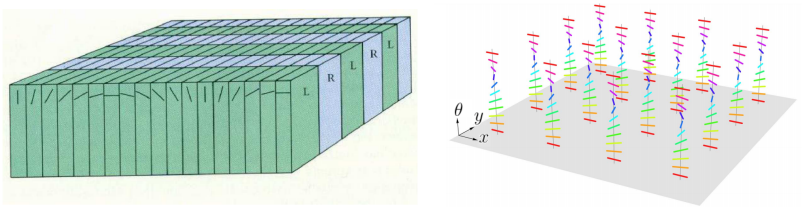
(a) (b)

Figure 1.7 (a) From [141]: the visual stimulus used to estimate the retino-cortical mapping (on the left) and the flattened visual cortex of a macaque with the corresponding activated regions (on the right). (b) The conformal mapping of the unit half disk by $\log(z+0.1)$ *[119]. The vertical meridian of the disk is mapped to the curved boundary on the left in the range of the map.*

its approximation through $\ell_{a,k}$.

Hypercolumnar structure

It has been shown, through recording of the responses to certain stimuli (e.g. oriented bars passing through the RF), that the preferred orientation of V1 neurons is roughly constant moving perpendicularly to the cortical surface [82]. These groups of neurons with similar orientation selectivity are called *orientation columns*. On the other hand, the preferred orientation varies gradually in the directions parallel to the surface, so that different columns



(a) (b)

Figure 1.8 (a) The classical "ice cube" model of V1. (b) Orientation displayed as a separate variable over the retinal coordinates at each point.

are sensitive to different orientations. V1 neurons are also organized into alternating ocular dominance columns, containing cells that are responsive only to input from the left or right eye, see Figure 1.8a. According to the classical "ice cube" model [82], these groupings form computational units called *hypercolumns*, each containing cells sensitive to approximately the same retinal position but spanning all orientations, thus monitoring information from one point in the visual field.

Orientation maps

The arrangement of orientation preference of V1 neurons is described by *orientation maps*, providing another example of the topographic organization of the cortex besides retinotopy. These maps can be measured through in-vivo optical imaging techniques [25, 27] based on the acquisition of activity of cells from the superficial layers of V1 in presence of visual stimuli consisting of oriented gratings. For each orientation value, the corresponding

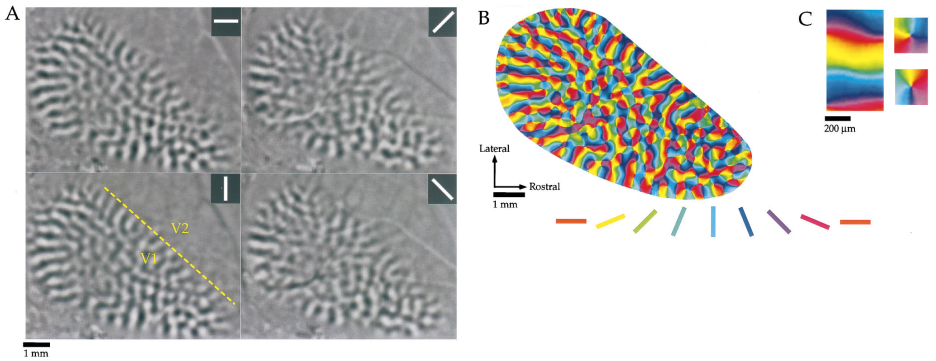


Figure 1.9 (A) Difference images obtained for stimulus angles $\theta = 0°, 45°, 90°, 135°$. Dark areas were preferentially activated by a stimulus with orientation θ; light areas were active during presentation of the orthogonal angle. (B) The orientation map $\Theta(x,y)$ obtained by vector summation of data obtained for each angle. The orientation preference measured at each location (x,y) is color-coded according to the key shown below. (C) Enlarged portions of the map show linear two zones (left) and two pinwheel arrangements (right). Source: [27].

activity pattern is subtracted from the pattern acquired during presentation of the orthogonally oriented grating; these individual *difference images*, after undergoing some smoothing and a normalization, are then combined by vector summation to create an orientation preference map, where the orientation θ is given as a function of position. Figure 1.9 shows four difference maps (A) and the corresponding color-coded orientation map (B) in tree shew V1 [27]. These maps typically contain regions where the orientation value changes linearly, as well as singular points around which all values of θ are arranged as the spokes of a wheel: the latter are commonly referred to as *pinwheels*.
Each pinwheel arrangement corresponds to a hypercolumn of orientation. Position is sampled onto the cortex at a coarser resolution with respect to orientation, so that at each discrete spatial location all orientations are coded. In other words, orientation is "engrafted" onto the positional map with a *finer subdivision* [81].

1.1.3 Beyond a hierarchical organization

According to the classical *feedforward* model of the visual system, the processing of a visual stimulus involves the transfer of the signal through a hierarchy of sub-cortical and cortical areas performing a cascade of processing stages which code for increasingly complex features, as mentioned at the beginning of this chapter. However, neural responses do not only result from an ordered sequence of hierarchical computations, but rather they are modulated by both horizontal (intra-area) connections and feedback signals from higher areas. This allows to integrate the information to perform complicated perceptual tasks such as grouping, contour integration, figure-ground segregation. Indeed, a *global* analysis is necessary in order to correctly recognize objects and interpret a visual scene: single receptive profiles alone cannot account for such non-local features [76, 115, 6, 71, 121]. In the following, we will focus on the geometrical properties of the horizontal connectivity of V1.

The intra-cortical circuitry of V1 can be described in terms of two main mechanisms: a inhibitory short-range connectivity taking place within each hypercolumn, and a long-range *horizontal* (or lateral) connectivity, linking neurons belonging to different hypercolumns. The former essentially selects the orientation of maximum output in response to a visual stimulus and suppresses the others: this mechanism explains the sharp orientation tuning experimentally observed in most simple cells, and it is commonly referred to as *non-maxima suppression*. As for horizontal connections, these exhibit some precise geometrical properties that have been investigated in several neurophysiological experiments [72, 125, 27]. In [27], small biocytin injections in a site of tree shew V1 were made, and the resulting distribution of boutons was tracked: this allowed to analyze the geometry of the connections departing

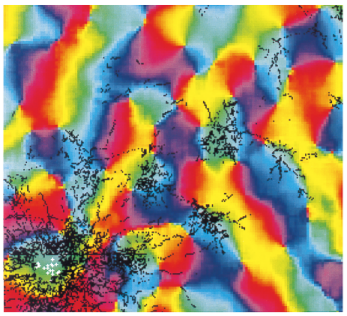


Figure 1.10 Bouton distributions resulting from a biocytin injection, shown over the orientation preference map. Image modified from [27].

from the localized set of neurons who took up the biocytin. This study highlighted the wide range of these connections, linking neurons even with markedly separated RFs, as well as their orientation specificity: neural activity turned out to spread around each neuron along the axis of its preferred orientation, targeting other cells sensitive to similar orientations. This last property leads to a *patchy* configuration, with terminations concentrating selectively in the regions of the orientation preference maps corresponding to the angle of the starting cell. See Figure 1.10, from [27].

1.2 Perceptual phenomena

1.2.1 Gestalt theory

The processing mechanism taking place throughout the visual pathways allows to efficiently group local elements into complete objects, and to segregate a figure from its background. The first systematic studies of the principles ruling perceptual organization were carried out in the 1920s by German psychologists Max Wertheimer, Wolfgang Köhler and Kurt Koffka, who founded the school of thought known as *Gestalt psychology*. The German word "Gestalt" can be translated as "organized whole", and it refers to the general concept that parts identified individually are not sufficient to describe the whole, which has a reality of its own. In other terms, when observing an object we are not directly conscious of its parts even if they can be clearly seen, yet we are aware of the overall scene. Classical references for these topics are [90, 91, 147]. We also refer to [143] for a recent review.

The purpose of Gestalt theorists was to understand the laws behind the ability of our mind to build meaningful perceptions out of a disorganized reality. This requires a form of self-organization which allows to combine smaller elements to form larger objects, according

to certain rules: the Gestalt approach is based on the development of a set of principles known as *laws of perceptual organization*, in an attempt to systematize these innate mental rules. The first Gestalt studies focused on grouping, while further research also approached the problem of perceptual organization in terms of figure-ground segregation. Well-known Gestalt principles related to grouping are those of *good continuation* (aligned elements are perceived as a group) and *proximity* (elements that are close to each other and apart enough from the rest of the objects form a cluster). On the other hand, an example of principle linked to figure-ground segregation is *convexity*, stating that convex rather than concave patterns tend to be perceived as figures.

1.2.2 Association fields

The Gestalt law of good continuation has been investigated quantitatively in a variety of studies [83, 69, 61]. In [61], psychophysical experiments were carried out to determine these rules in terms of reciprocal position and orientation of local edge elements. Observers were

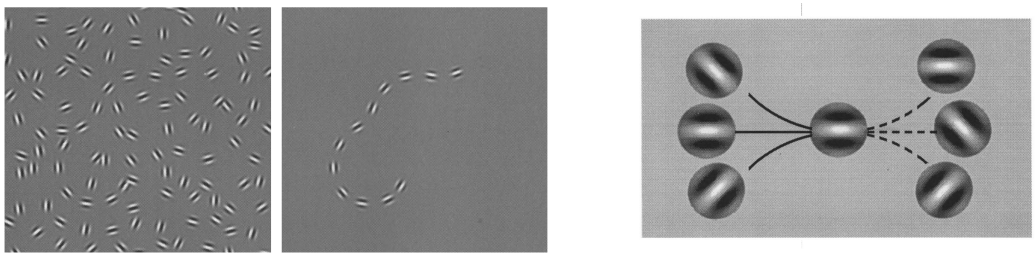


Figure 1.11 (a) Path segregation. (b) A schematic representation of the notion of association field. Images taken from [61].

tested in their ability to identify a set of oriented local elements (Gabor patches) forming a path, among other items not organized in any recognizable pattern (see Figure 1.11a), with different conditions of spacing and alignment of the elements. The results of the study are synthesized in the schematic concept of *association field* [61], displayed in Figure 1.11b: this object characterizes the geometry of the mutual influences between local edge elements depending on their orientation and reciprocal position. In other words, the perception of an edge element is strengthened by the presence of surrounding segments with certain relative positions and orientations with respect to it. In particular, the strongest correlation takes place between those segments that are either *collinear* or *co-circular*: in Figure 1.11b, filled lines indicate the correlation between the central horizontal element and the ones on its left, while dotted lines connect it with elements uncorrelated with it, such as the ones on its right. The psychophysical analysis performed in [61] revealed that association fields and horizontal

connections display a comparable spatial extent: this fact, together with their shared orientation specificity, makes the lateral connectivity a potential anatomical implementation of this perceptual phenomenon [71, 121].

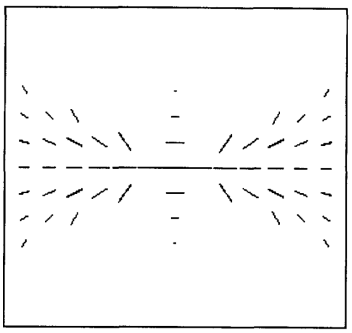


Figure 1.12 The connectivity pattern *found in the model of [152] includes both co-axial and trans-axial contributions. The length of the lines indicates connection strength.*

The "classical" association field focuses on the reciprocal influences between *co-axial* oriented elements. However, an analogous correlation has also been observed between *parallel* segments arranged along the *trans-axial* direction, i. e. the one orthogonal to the central oriented element: this phenomenon is referred to as *ladder effect* [106, 61, 152]; see Figure 1.12.

1.3 Mathematical models of V1

In this section, we give an overview on some previous mathematical models describing the connectivity of the visual cortex and the laws of perceptual organization. We then briefly outline the model constructed by G. Citti and A. Sarti [41] in $\mathbb{R}^2 \times S^1$, which we will later recover as a limit case of our model, when applied to a bank of Gabor filters.

1.3.1 Overview

Neuromathematical models of V1 often adopt a differential geometry approach to modeling the functional architecture of this area. A breakthrough idea has been that of representing V1 as a fiber bundle with the space of retinal locations as a basis. This differential approach first appeared in the works of J. Koenderink [89] and W. Hoffman [79]. It was then further developed by J. Petitot and Y. Tondut [121] with the idea that, in the cortical processing, illusory contours on the retinal plane are lifted to curves satisfying a geodesic condition in

the total space. A more complete description, allowing for non equi-oriented boundaries, was given in [41] by writing their model in the Lie group $SE(2) = \mathbb{R}^2 \times S^1$ by requiring the invariance under roto-translations. This construction is employed to formulate a model of perceptual completion and formation of subjective surfaces which can be seen as the lifting in $SE(2)$ of the variational models for inpainting, based on elastica functionals [5, 15, 21]. The local invariance of the architecture of V1 w.r.t. $SE(2)$ has been exploited in a number of other works. In [16], cortical orientation maps are modeled as minimizers of an uncertainty principle stated in this setting, see also [49]. In [18], orientation maps are generated by also taking into account the scale variable. An extension of [41] in the context of spatio-temporal analysis of visual stimuli was developed in [17]. On the other hand, a semi-discrete variant of [41] was proposed in [26], where only a finite number of angles is taken into consideration. Moreover, a study on the relation between association field curves and sub-Riemannian geodesics in $\mathbb{R}^2 \times S^1$ has been carried out in [53]. Perceptual grouping of spatial visual features is modeled in [128], where the authors develop a mean field neural theory and use harmonic analysis tools to account for the constitution of perceptual units in presence of a visual stimulus. In [45], a model for the grouping of spatio-temporal visual features is proposed. In [42], the authors address the problem of modal completion through a gauge field Lagrangian: this couples the neurogeometrical model in [41] with a retinex term [95, 68, 22] implementing the perceptual invariance w.r.t. contrast. Moreover, a number of models have been proposed where perceptual grouping and texture segmentation tasks are addressed through a combination of long-range interactions and recurrent processing, involving the so-called bipole model [76, 115, 77].

In [127], a Fokker-Planck equation was considered instead of a sub-Riemannian heat equation, resulting from a study of statistical kernels of edge co-occurrence in natural images (see also [112] and [129]). This corresponds to assuming that the propagation has a deterministic component along the first horizontal vector field and a stochastic component in the direction of the second.

Linear as well as non-linear diffusion equations in $\mathbb{R}^2 \times S^1$ were also employed in [55, 56] for medical image processing; here, the lifting process of the image is based on a so-called orientation score, essentially providing a local orientation representation of an image through a generalized unitary wavelet transform [54]. In [1], such a lifting was complemented with the definition of a sub-Riemannian structure in a 5-dimensional space obtained by introducing the features of curvature and intensity, for a blood vessel tracking task. In [57], the approach of evolution equations was extended to the range of Gabor transforms, again with applications to the processing of images.

The functional architecture of the visual cortex has also been described through structures

different from the Lie group $SE(2)$, such as the affine group [130], the Galilei group [17], the hyperbolic space [132].
Another model of V1 horizontal connections based on differential geometry has been proposed in [20], where a crucial role is again given to curvatures, providing a directional rate of change of orientation. In this work, the relationship between nearby tangents is analyzed for curves as well as textures. An analysis of other possible relationships between the lateral connectivity and visual function beyond contour integration had been carried out in [19] as well.

1.3.2 The Citti-Sarti model

The model proposed by G. Citti and A. Sarti in [41] focuses on the representation of a gray level image I through its level lines. Simple cells over each point (x,y) are sensitive to the orientation of the level lines of I, thus detecting an angle as engrafted variable. The cell giving the maximal response is assumed to have $\Theta(x,y) = -\arctan\left(\frac{I_y}{I_x}\right)$ as its preferred orientation. Therefore, the vector field

$$Y_\Theta = -\sin(\Theta(x,y))\partial_x + \cos(\Theta(x,y))\partial_y$$

on $\mathbb{R}^2$ is tangent to the level lines of I at the point (x,y). The images of these lines through the map $(x,y) \mapsto (x,y,\Theta(x,y))$ are called *lifted level lines*, and their tangent vector at every point can be written as a linear combination of the vector fields

$$Y_1 = -\sin\theta\,\partial_x + \cos\theta\,\partial_y\,, \quad Y_2 = \partial_\theta\,. \tag{1.4}$$

These vector fields define a bi-dimensional sub-bundle of the tangent bundle to $\mathbb{R}^2 \times S^1$, referred to as the *horizontal tangent bundle*. One can define a scalar product on this sub-bundle by imposing the orthonormality of Y_1 and Y_2: this determines a sub-Riemannian structure on $\mathbb{R}^2 \times S^1$. The Lie algebra generated by Y_1 and Y_2 through the bracket operation between vector fields is the whole Euclidean tangent plane, since

$$[Y_1, Y_2] = -\cos(\theta)\,\partial_x - \sin(\theta)\,\partial_y =: -Y_3. \tag{1.5}$$

In other words, Y_1 and Y_2 satisfy the Hörmander rank condition. This leads, by the Chow theorem, to the so-called *connectivity property*: any couple of points in $\mathbb{R}^2 \times S^1$ can be connected through a *horizontal curve*, i.e. an integral curve of a section of the horizontal tangent bundle. See [80, 38, 107] as references on these topics.

Such sub-Riemannian structure has been shown to be naturally induced by the action of a bank of Gabor filters on a visual stimulus. In particular, consider the bank of filters $\{\psi_{x,y,\theta}\}_{x,y,\theta}$ defined in (1.3), and denote $O_{(x,y,\theta)}(I) := O_{\psi_{x,y,\theta}}(I)$. That is, $O_{(x,y,\theta)}(I)$ is the outcome of filtering an image I with the profile $\psi_{x,y,\theta}$. This can be locally approximated as

$$O_{(x,y,\theta)}(I) \approx -Y_3(\theta)\, I_\sigma(x,y), \tag{1.6}$$

where I_σ is a smoothed version of I, obtained by convolving it with a Gaussian kernel:

$$I_\sigma(x,y) = \int \exp\left(\frac{(x-u)^2+(y-v)^2}{\sigma^2}\right) I(u,v)\,du\,dv.$$

This once more defines a lifting of the 2D image domain to the 3D space $\mathbb{R}^2 \times S^1$: each point $(x,y) \in \mathbb{R}^2$ is sent to a point $(x,y,\overline{\theta}) \in \mathbb{R}^2 \times S^1$ such that $\overline{\theta}$ is a local maximum point of $\theta \mapsto O_{(x,y,\theta)}(I)$, so that the whole image domain is lifted to the set

$$\left\{(x,y,\overline{\theta}) \;:\; O_{(x,y,\overline{\theta})}(I) = \max_{\theta} O_{(x,y,\theta)}(I)\right\} \subseteq \mathbb{R}^2 \times S^1.$$

This "non-maximal suppression" principle is based on experimental evidence on the sharp orientation tuning of V1 neurons. The lifting of the level lines of I through this procedure yields curves that are tangent to the planes generated by Y_1 and Y_2 [43].

The lateral propagation of neural activity in the cortical space is described in [41] through the sub-Riemannian heat equation $\partial_t u = \Delta u$, where $\Delta = Y_1^2 + Y_2^2$, and the association field around a point $(x_0,y_0,\theta_0) \in \mathbb{R}^2 \times S^1$ is characterized as a family of integral curves of Y_1 and Y_2 starting at this point. Namely, $\gamma' = Y_{1\,|\gamma} + kY_{2\,|\gamma}$ and $\gamma(0) = (x_0,y_0,\theta_0)$, where k varies in $\mathbb{R}$.

The evolution of the activity of V1 neurons is influenced by a combination of intra-columnar and lateral connections. In [41], the sub-Riemannian diffusion modeling the horizontal connections and the mechanism of selection of maxima implemented by the short-range connectivity have been combined by alternating their action iteratively: precisely, each iteration consists of a first step of diffusion in a finite time interval and a second step of non maximal suppression. The time interval is then sent to zero. See also [30] and [31], where the connections between each couple of neurons are represented by a weight function which is decomposed as the sum of two terms modeling these two mechanisms.

In the next chapter, we will propose a new model of the functional architecture of V1 that does not require any differential or group structures as most previous models. Our construction is based on the definition of a metric structure onto the feature space associated to a family of filters modeling the RPs of simple cells. Such a geometry is induced by the shape of the RPs themselves. In particular, when the family of RPs is represented by a bank of Gabor filters indexed by $\mathbb{R}^2 \times S^1$, the induced distance is locally equivalent to a Riemannian approximation to the sub-Riemannian structure described in [41]. Still, our model can be applied to a wide range of filter banks, and it is able to recover some geometrical properties compatible with both neurophysiological and psychophysical experimental evidence, even when applied to a randomly ordered family of learned filters.

Chapter 2

A metric model for the functional architecture of V1

In this chapter, after recalling some theoretical notions on distances and measures (Section 2.1), we provide the main original contribution of this thesis, consisting of a novel technique for modeling the connectivity of V1 based on the feature selectivity of simple cells. We first introduce a distance onto the feature space associated to a family of RPs, which is naturally coupled with a local notion of correlation between the profiles (Section 2.2). Along the lines of the differential approach described in Section 1.3, we will then characterize the long range lateral connectivity through a propagation with respect to this metric structure (Section 2.3). Finally, we will show the results obtained by applying our construction to some different banks of filters and examining the geometric properties emerging from the computed connectivity (Section 2.4). We have recently outlined the above-mentioned methods and results in [109, 110]. Here, we want to give a self-contained and unitary presentation of these studies.

2.1 Theoretical background

2.1.1 Metric measure spaces

In this section, we recall some notions about distances, length spaces and measures. We refer to [33] as an introductory text on these topics.

Definition 2.1. *Given an arbitrary set X, a function $d : X \times X \to \mathbb{R} \cup \{+\infty\}$ is a* distance *(or* metric*) on X if the following conditions are satisfied for all $p, q, q' \in X$.*

(i) $d(p,q) > 0$ *if* $p \neq q$*, and* $d(p,p) = 0$ *(positiveness).*

(ii) $d(p,q) = d(q,p)$ *(symmetry).*

(iii) $d(p,q) \leq d(p,q') + d(q',q)$ *(triangle inequality).*

The pair (X,d) *is called a* metric space*.*

Remark 2.1. Every metric space (X,d) has a natural structure of topological space: a basis for the topology on X is given by the set of *open balls* $B_\varepsilon(p) := \{q \in X : d(p,q) < \varepsilon\}$ for all $\varepsilon > 0$ and $p \in X$.

Definition 2.2. *A metric space is* separable *if it admits a dense countable subset.*

Definition 2.3. *A metric space* X *is* locally compact *if every point of* X *has a compact neighborhood.*

A distance on a set X can be induced by a *length structure*, which consists of a class of *admissible paths* for which we can define a length, and a *length function* assigning a nonnegative number to every admissible path. Specifically, we call a *path* any continuous map $\gamma : [a,b] \to X$, and we define:

Definition 2.4. *A* length structure *on a topological space* X *is a couple* $(A(X),L)$*, where* $A(X)$ *is a set of paths on* X *which is closed under restrictions, concatenations and linear reparameterizations of paths; and* $L : A(X) \to \mathbb{R}$ *is a nonnegative function such that:*

(i) $L\left(\gamma_{|_{[a,b]}}\right) = L\left(\gamma_{|_{[a,c]}}\right) + L\left(\gamma_{|_{[c,b]}}\right)$ *for any* $c \in [a,b]$*;*

(ii) $t \mapsto L\left(\gamma_{|_{[a,t]}}\right)$ *is continuous on* $[a,b]$*;*

(iii) $L(\gamma \circ \varphi) = L(\gamma)$ *for any linear homeomorphism* φ*;*

(iv) $\inf\{L(\gamma) : \gamma(a) = p,\, \gamma(b) \in X \setminus U_p\} > 0$ *for all* $p \in X$ *and any neighborhood* U_p *of* p*.*

A path $\gamma \in A(X)$ *is called* admissible*, and* $L(\gamma)$ *is called the* length *of* γ*.*

Remark 2.2. Property (iv) expresses a condition of compatibility with the underlying topology of the space.

We can now define a distance function induced by a length structure as follows:

Definition 2.5. *Given a length structure* $(A(X),L)$ *we define, for all* $p,q \in X$*,*

$$d_L(p,q) := \inf\{L(\gamma) \;\; s.t. \;\; \gamma : [a,b] \to X,\; \gamma \in A(X),\; \gamma(a) = p,\; \gamma(b) = q\}. \tag{2.1}$$

Remark 2.3. It is easy to check that (X,d_L) is a metric space.

Definition 2.6. *A length structure is* complete *if for all $p,q \in X$ there exists an admissible path joining them whose length is equal to $d_L(p,q)$; that is, if there exists a shortest path between every two points. In this case, the distance d_L is said to be* strictly intrinsic.

Definition 2.7. *A metric space (X,d) is called a* length space *if d can be obtained as the distance function d_L associated to a length structure; it is a* geodesic space *if d_L is strictly intrinsic.*

Example 2.1. Consider a metric space (X,d) and define the length of a path $\gamma : [a,b] \to X$ as

$$L_d(\gamma) := \sup \left\{ \sum_{i=1}^{n} d(\gamma(t_i), \gamma(t_{i-1})) \ : \ n \in \mathbb{N},\ a \leq t_0 \leq \ldots \leq t_n \leq b \right\}. \tag{2.2}$$

This defines a length structure, where the class of admissible paths contains all continuous paths on X parameterized by closed intervals.

Remark 2.4. The length function L_d of the previous example induces a distance $\hat{d} = d_{L_d}$ as in (2.1). $\hat{d}$ is called the *intrinsic* distance induced by d, and it does not necessarily coincide with d. If it does, then (X,d) is clearly a length space. On the other hand, it is possible to prove that for any length space (X,d) the intrinsic distance $\hat{d}$ induced by d coincides with d itself. This brings to the following equivalent definition of length space.

Definition 2.8. *Let d be a distance on X, and let $\hat{d}$ be the intrinsic metric induced by d. If $d = \hat{d}$, then the metric space (X,d) is said to be a* length space.

Example 2.2 (Riemannian distance). A *Riemannian* structure on a smooth manifold M is determined by a smooth section g of the positive-definite quadratic forms on the tangent bundle TM; that is, a scalar product $g_p(\cdot,\cdot)$ on the tangent space T_pM at each point p, varying smoothly w.r.t. p. This allows to define the *length* of a smooth curve $\gamma : [a,b] \to M$ as

$$L_g(\gamma) := \int_a^b \sqrt{g_{\gamma(t)}(\gamma'(t),\gamma'(t))}dt. \tag{2.3}$$

The *Riemannian distance* d_g on (M,g) is the one obtained from L_g as in (2.1). The metric space (M,d_g) is a length space, and it is also geodesic if the manifold is complete.

Example 2.3 (Carnot-Carathéodory distance). Consider now a smooth horizontal subbundle of TM, that is a collection HM of subspaces $H_pM \subseteq T_pM$, for any $p \in M$, that is locally generated by a set of smooth vector fields. A smooth curve $\gamma : [a,b] \to M$ is *horizontal* if $\gamma'(t) \in H_{\gamma(t)}M$ for a.e. $t \in [a,b]$. A smooth connected manifold M with horizontal subbundle

HM is said to be *H-connected* if any two points of M can be joined by a horizontal curve. By the Chow-Rashevsky Theorem, a sufficient condition for M to be H-connected is that the Lie algebra generated by H_pM through the Lie product of vector fields coincides with T_pM (*Chow condition*).
Now, a *sub-Riemannian metric* on M is given by a smoothly varying scalar product g defined on H_pM for each p. The triplet (M, HM, g) is called a *sub-Riemannian manifold*. Again, the metric g induces a length function as in (2.3); however, in this case L_g is only defined for horizontal curves, which are the admissible paths for this length structure. The distance induced on M by this length function via (2.1) is called the *Carnot-Carathéodory distance*. We refer to [107] for a complete review of these topics.

We now recall the definition of *measure* on a set X. If a metric space (X,d) is endowed with a measure, this is in general unrelated to the topological structure induced by the distance. However, as we will briefly point out, these two structures can be linked by some notion of compatibility.

Definition 2.9. *Let X be a set. A* σ-algebra *on X is a set $\mathscr{A}$ of subsets of X such that:*

(i) $\mathscr{A} \ni \emptyset, X$;

(ii) $A, B \in \mathscr{A} \Rightarrow A \setminus B \in \mathscr{A}$;

(iii) if $\{A_i\}_{i\in I} \subseteq \mathscr{A}$ *is a finite or countable collection, then* $\bigcup_i A_i \in \mathscr{A}$.

Remark 2.5. Given an arbitrary collection $\mathscr{G}$ of subsets of X, there exists a unique minimal σ-algebra containing $\mathscr{G}$; it is called the σ-algebra *generated* by $\mathscr{G}$.

Definition 2.10. *A* measure *on a σ-algebra $\mathscr{A}$ is a function $\mu : \mathscr{A} \to \mathbb{R}_+ \cup \{+\infty\}$ such that:*

(i) $\mu(\emptyset) = 0$;

(ii) if $\{A_i\}_{i\in I} \subseteq \mathscr{A}$ *is a finite or countable collection of disjoint sets, then* $\mu\left(\bigcup_i A_i\right) = \sum_i \mu(A_i)$.

A subset of X is said to be measurable *if it belongs to $\mathscr{A}$. We say that μ is* finite *if $\mu(X) < +\infty$, and* σ-finite *if X is the countable union of measurable sets with finite measure.*

When the set X is endowed with a topology, it is possible to define a σ-algebra determined by the open sets of X.

Definition 2.11. *If X is a topological space, then the σ-algebra generated by the set of all its open sets is called the* Borel σ-algebra *of X. A measure defined on the Borel σ-algebra is called a* Borel measure *over X.*

When the topological structure is induced by a distance, a Borel measure over X defines what is usually called a *metric measure space*:

Definition 2.12. *A* metric measure space *is a triple (X,d,μ) where d is a distance on X and μ is a measure on the Borel σ-algebra of (X,d).*

One may relate the properties of a Borel measure with the underlying topology of the space. The following is a rather natural "compatibility" condition between the measure and the topology.

Definition 2.13. *A measure μ on the σ-algebra of Borel sets of a Hausdorff topological space X is called a* Radon measure *if it has the following properties.*

(i) *μ is* inner regular*: for any open set A, $\mu(A)$ is the supremum of $\mu(K)$ over all compact sets $K \subseteq A$.*

(ii) *μ is* outer regular*: for any Borel set B, $\mu(B)$ is the infimum of $\mu(A)$ over all open sets $A \supseteq B$.*

(iii) *μ is* locally finite*: every point of X has a neighborhood of finite measure.*

We conclude this section by briefly introducing the (spherical) Hausdorff measure induced by a distance. We cite [78, 150, 60] as references for this topic.

Definition 2.14. *Let (X,d) a metric space and $s \geq 0$. For an $\varepsilon > 0$ define*

$$\mathscr{H}^s_\varepsilon(X) := \inf\left\{ \sum_i \big(diam(S_i)\big)^s \mid diam(S_i) < \varepsilon \ \forall i \right\}, \tag{2.4}$$

where the infimum is taken over all finite or countable coverings $\{S_i\}_{i\in I}$ of X, and we take $\inf\emptyset = +\infty$*. The* s-dimensional Hausdorff measure *of X is defined by*

$$\mathscr{H}^s(X) := \lim_{\varepsilon\to 0} \mathscr{H}^s_\varepsilon(X). \tag{2.5}$$

When the infimum in (2.4) is only taken over coverings of balls of the distance d, we denote it by $\mathscr{S}^s_\varepsilon(X)$ and the limit

$$\mathscr{S}^s(X) := \lim_{\varepsilon\to 0} \mathscr{S}^s_\varepsilon(X) \tag{2.6}$$

is called the s-dimensional spherical *Hausdorff measure of X*.

Remark 2.6. Since $\mathscr{H}^s_\varepsilon(X)$ and $\mathscr{S}^s_\varepsilon(X)$ are nonincreasing functions of ε, their (possibly infinite) limit as $\varepsilon \to 0$ always exists.

Remark 2.7. For subsets $A \subseteq X$, the (spherical) Hausdorff measure is defined by considering A as a metric space with the restricted metric.

Theorem 2.1. For any metric space (X,d) and any $s \geq 0$, $\mathscr{S}^s$ and $\mathscr{H}^s$ are Borel measures on X.

Example 2.4. The 0-dimensional (spherical) Hausdorff measure of a set is its cardinality.

Remark 2.8. The measures $\mathscr{H}^s$ and $\mathscr{S}^s$ are comparable, but not equal in general.

The Hausdorff measure of a set is strictly linked with the concept of dimension. Specifically, there is a "critical dimension" s_0, below which the measure is infinity and above which the measure is zero. This allows to extend the notion of dimension to metric spaces. The precise statement is enunciated below.

Theorem 2.2. For a metric space (X,d), there exists a real number $s_0 \in [0,+\infty]$ such that $\mathscr{H}^s(X) = 0$ for all $s > s_0$ and $\mathscr{H}^s(X) = +\infty$ for all $s < s_0$. The value s_0 is called the *Hausdorff dimension* of X.

Remark 2.9. The Hausdorff dimension s_0 is not necessarily an integer, and the measure $\mathscr{H}^{s_0}$ can be zero, a positive number or infinity. The theorem also holds for the spherical Hausdorff measure, with the same s_0.

Example 2.5. Countable sets have Hausdorff dimension 0; n-dimensional manifolds have Hausdorff dimension n.

In the following, "the (spherical) Hausdorff measure" on a metric space X refers to the s_0-dimensional measure, where s_0 is the Hausdorff dimension of X.

2.1.2 Dirichlet forms and associated operators

We now give some basic definitions on Dirichlet forms. The reader may refer to [64] for a comprehensive description.

Definition 2.15. *Given a real Hilbert space H, a* symmetric form *on H is a non-negative definite symmetric bilinear form densely defined on H.*

Remark 2.10. A symmetric form E induces a distance d on its domain by $d^2(u,v) := E(u-v,u-v)$.

Now, given a symmetric form $E : \mathscr{D}(E) \times \mathscr{D}(E) \to \mathbb{R}$ on H, a new symmetric form is defined on $\mathscr{D}(E)$ by

$$E_1(u,v) := E(u,v) + \langle u,v\rangle,$$

where $\langle\cdot,\cdot\rangle$ is the scalar product of H.

Definition 2.16. *A symmetric form E is* closed *if $\mathscr{D}(E)$ is complete w.r.t. the metric induced by E_1. It is* closable *if*

$$\{u_n\}_n \subseteq \mathscr{D}(E),\ E(u_n-u_m,u_n-u_m)\to 0,\ \langle u_n,u_n\rangle \to 0 \quad \Rightarrow \quad E(u_n,u_n)\to 0.$$

Proposition 2.1. A symmetric form is *closable* if and only if it admits a closed extension.

Theorem 2.3. For any closed symmetric form E on H, there exists a unique non-positive definite self-adjoint operator A on H such that

$$\begin{cases} \mathscr{D}(E) = \mathscr{D}(\sqrt{-A}) \\ E(u,v) = \langle -Au, v\rangle,\ u\in\mathscr{D}(A),\ v\in\mathscr{D}(E). \end{cases} \tag{2.7}$$

We now fix a measure space (X,μ), where μ is a σ-finite Borel measure, and we consider symmetric forms on the Hilbert space $H = L^2(X,\mu)$.

Definition 2.17. *A symmetric form on $L^2(X,\mu)$ is* Markovian *if for each $\varepsilon > 0$ there exists a function $\phi_\varepsilon : \mathbb{R}\to\mathbb{R}$ such that*

(i) $\phi_\varepsilon(t) = t$ *for all* $t\in[0,1]$, $-\varepsilon \le \phi_\varepsilon(t) \le 1+\varepsilon$ *for all* $t\in\mathbb{R}$, *and*

$$0 \le \phi_\varepsilon(t') - \phi_\varepsilon(t) \le t'-t \quad \forall t < t';$$

(ii) $u\in\mathscr{D}(E) \Rightarrow \phi_\varepsilon\circ u \in \mathscr{D}(E)$ *and* $E(\phi_\varepsilon\circ u, \phi_\varepsilon\circ u) \le E(u,u)$.

A closed Markovian symmetric form on $L^2(X,\mu)$ is called a Dirichlet form. *A Dirichlet form E is said to be* strongly local *if $E(u,v)=0$ whenever $u\in\mathscr{D}(E)$ is constant on a neighborhood of the support of $v\in\mathscr{D}(E)$.*

Example 2.6. Let (M,g) be a complete Riemannian manifold, equipped with the measure $d\mu_g(p) = \sqrt{\det g_p}\,dp$ induced by g. One can define a strongly local Dirichlet form by

$$E(u,v) := \frac{1}{2}\int_M g(\nabla u, \nabla v) d\mu_g, \tag{2.8}$$

where ∇u denotes the Riemannian gradient of u, defined as the unique vector field satisfying $g(\nabla u, X) = df(X)$ for any vector field X. The non-positive definite self-adjoint operator associated to E (as in Theorem 2.3) coincides with $\frac{1}{2}\Delta$, where Δ is the *Laplace-Beltrami* operator on M, defined by

$$\Delta u := div(\nabla u), \quad u \in C_0^\infty(M).$$

Indeed, for any $u, v \in C_0^\infty(M)$,

$$\langle -\Delta u, v\rangle = -\int_M div(\nabla u)\, v\, d\mu_g = \int_M g(\nabla u, \nabla v) d\mu_g.$$

Remark 2.11. A symmetric form E is often identified with its associated quadratic form, i.e. the one obtained by evaluating E on the diagonal. Conversely, any quadratic form Q defines a symmetric form $E(u,v) := \frac{1}{4}(Q(u+v) - Q(u-v))$ on $\mathscr{D}(Q) \times \mathscr{D}(Q)$. The same symbol E is often used to denote both the symmetric form and the quadratic form.

2.1.3 Diffusion processes on metric measure spaces

In this section, we provide a brief review on diffusion processes on metric measure spaces, containing some concepts that will be employed in the next sections. As mentioned before, we aim at extending to our context the diffusion-based approach adopted in differential models (see Section 1.3) to describe the horizontal propagation of neural activity in V1. As such, we need appropriate extensions of some important concepts and tools from smooth to non-smooth structures. In this section, we recall some results concerning the construction of diffusion processes on metric measure spaces and the properties of their associated Laplacian operators and heat kernels.

We shall mainly focus on the classical approach introduced in 1998 by K.-T. Sturm [137], which provides a general method to define Dirichlet forms and diffusion processes on metric measure spaces (X, d, μ), under a crucial assumption called the *Measure Contraction Property* (MCP). This property essentially gives a bound for distortions of the measure μ under contractions of X along quasi-geodesics w.r.t. the metric d. As a first example, any Riemannian manifold equipped with its geodesic distance and the Riemannian volume form satisfies this requirement, as a consequence of the Bishop volume comparison theorem.

Slightly different versions of the MCP were introduced in later years [138, 116]. These may be regarded as a generalization of the lower Ricci curvature bound on Riemannian manifolds: in fact, the (K,n)-MCP proposed in [116] is shown to be equivalent, for an n-dimensional Riemannian manifold, to its Ricci curvature being bounded from below by $(n-1)K$. Another

kind of synthetic treatment of the lower Ricci curvature bound on metric measure spaces is given by so-called *curvature-dimension conditions* CD(K,N) [138, 102] expressed in terms of the convexity of certain functions on the associated Wasserstein space. Here, N plays in some sense the role of an upper bound for the dimension. All these properties are associated to a uniform lower bound for the Ricci curvature, encoded in the constant K. This is not needed in the present context, where we are interested in defining a well-behaved diffusion process on our metric measure space, which is guaranteed by the MCP introduced in [137]. Indeed, under this assumption it is possible to obtain generalizations of the Laplacian operator and of its heat kernel, and the latter can also be shown to admit Gaussian estimates in terms of the distance. It is worth mentioning that self-adjoint Laplace operators on metric measure spaces are also constructed in [35], by a different method based on the concept of upper gradients. We also refer to [4] for a review on calculus in metric measure spaces.

We now enunciate some definitions and results contained in [137]. Let (X,d,μ) be a metric measure space, such that (X,d) is a locally compact separable metric space and μ is a Radon measure on X, strictly positive on nonempty open sets. One can then construct a Dirichlet form E on $L^2(X,\mu)$ as the Γ-limit of a sequence of forms, defined in analogy with the Dirichlet form of Example 2.6. Specifically, one defines

$$E^r(u) = \frac{1}{2}\int_X \mathscr{N}(p) \int_{B_r(p)\setminus\{p\}} \left(\frac{u(q)-u(p)}{d(q,p)}\right)^2 \frac{d\mu(q)}{\sqrt{\mu(B_r(q))}} \frac{d\mu(p)}{\sqrt{\mu(B_r(p))}}, \tag{2.9}$$

where $\mathscr{N}$ is a normalization function, and lets $E = \Gamma\text{-}\lim_{r\to 0} E^r$. This does always exist, provided that (X,d,μ) satisfies the following property.

Definition 2.18. *A metric measure space (X,d,μ) satisfies the* (weak) Measure Contraction Property *(MCP)* with exceptional set *if there exists a closed set $Z \subseteq X$ with $\mu(Z) = 0$ such that for every compact set $Y \subseteq X \setminus Z$ there are numbers $R > 0, \zeta < \infty$ and $\vartheta < \infty$, and μ^2-measurable maps $\Phi_t : X \times X \to X$ (for all $t \in [0,1]$), with the following properties.*

(i) for μ-a.e. $p,q \in Y$ with $d(p,q) < R$, and for all $s,t \in [0,1]$,

$$\Phi_0(p,q) = p, \quad \Phi_t(p,q) = \Phi_{1-t}(q,p), \quad \Phi_s(p,\Phi_t(p,q)) = \Phi_{st}(p,q), \tag{2.10}$$

$$d(\Phi_s(p,q),\Phi_t(p,q)) \le \vartheta|s-t|d(p,q). \tag{2.11}$$

(ii) Define, for $r<0$, the measures $d\mu_r(p)=\frac{d\mu(p)}{\sqrt{\mu(B_r(p))}}$. Then, for all $r<R$, μ-a.e. $p\in Y$, all μ-measurable $A\subseteq B_r(p)\cap Y$ and all $t\in[0,1]$,

$$\frac{\mu_r(A)}{\sqrt{\mu(B_r(p))}}\leq\zeta\,\frac{\mu_{rt}(\Phi_t(p,A))}{\sqrt{\mu(B_{rt}(p))}}. \tag{2.12}$$

The space (X,d,μ) is said to verify the strong MCP *if:*

- *the constants ζ and ϑ can be taken arbitrarily close to 1;*
- *for every $\zeta'>1$ there exists some $\vartheta'>1$ such that, for μ-a.e. $p\in Y$ and for all $r<R$ with $B_r(p)\subseteq Y$,*

$$\mu(B_{r\vartheta'}(p))\leq\zeta'\mu(B_r(p)).$$

In this case, there is no restriction in taking always $\zeta=\zeta'$ and $\vartheta<\vartheta'$.
For both the weak and strong MCP, one says without exceptional set *if $Z=\emptyset$.*

Remark 2.12. For fixed p and q, the map $\Phi_\cdot(p,q):[0,1]\to X,\ \ t\mapsto\Phi_t(p,q)$ is a *quasi-geodesic* joining p and q. Moreover, if (X,d) is a geodesic space such that geodesics joining p and q can be chosen in such a way that they depend in a measurable way on p and q, property (ii) simplifies to

$$\frac{\mu(A)}{\mu(B_r(p))}\leq\zeta\,\frac{\mu(\Phi_t(p,A))}{\mu(B_{rt}(p))}.$$

Example 2.7. Let (X,g) be a Riemannian manifold. If d is the Riemannian distance and μ is the Riemannian volume on X, (X,d,μ) is a metric measure space satisfying the initial requests on d and μ. Furthermore, for such a space the strong MCP without exceptional set is verified. This example is central in our setting, since the basic case which we have in mind as a prototype will be the metric space induced by a family of Gabor filters, whose distance is estimated by a Riemannian distance (see Section 2.2.2).
More examples are given by manifolds with corners or by gluing together of manifolds not necessarily of the same dimension.

The MCP implies some important facts, among which the volume doubling property on $X\setminus Z$. That is, for each compact set Y there exist constants M and $R>0$ such that

$$\mu\left(B_{2r}(p)\right)\leq M\cdot\mu\left(B_r(p)\right)$$

for all $r\in]0,R[$ and μ-a.e. $p\in Y$ (see Proposition 4.5 in [137]). Moreover, for each $u\in C_0^{Lip}(X)$, the Γ-limit and the point-wise limit of $E^r(u)$ exist and coincide. Such a limit defines a strongly local, regular Dirichlet form, whose associated intrinsic metric is

locally equivalent to the original distance d. A Poincaré inequality is shown to hold as well. Finally, the corresponding positive self-adjoint operator A has a Hölder continuous heat kernel h_t (see Theorem 7.4 in [137]):

Theorem 2.4 (Sturm, 1998)**.** There exists a measurable function

$$H :]0,\infty[\times X\times X \longrightarrow [0,\infty], \quad (t,p,q)\longmapsto H(t,p,q)\equiv h_t(p,q) \tag{2.13}$$

with the following properties.

(i) For every $t>0$, every $u\in L^2(X,\mu)$ and μ-a.e. $p\in X$,

$$e^{-At}u(p) = \int_X h_t(p,q)u(q)d\mu(q). \tag{2.14}$$

(ii) The function H is locally Hölder continuous on $]0,\infty[\times(X\setminus Z)\times(X\setminus Z)$ and identically zero on its complement in $]0,\infty[\times X\times X$.

(iii) For all $s,t>0$ and all $p,q\in X$,

$$h_t(p,q)=h_t(q,p) \quad \text{and} \quad h_{t+s}(p,q)=\int_X h_s(p,q')h_t(q',q)d\mu(q'). \tag{2.15}$$

The function H is defined pointwise uniquely by these properties and is called *heat kernel* for A.

Furthermore, this heat kernel admits upper and lower Gaussian estimates. More precisely (see Theorems 7.7 and 7.9 of [137]), we have the following result.

Theorem 2.5 (Sturm, 1998)**.** Let (X,d,μ) verify the strong MCP with some exceptional set, and let Z be the exceptional set for the weak MCP. Then, for every compact $Y\subseteq X\setminus Z$ and every $\varepsilon>0$, there exists a constant C such that

$$\begin{aligned}\frac{1}{C\,\mu(B_{\sqrt{t}\wedge R}(p))}\exp\left(-C\frac{d^2(p,q)}{2t}\right)\exp\left(-\frac{Ct}{R^2}\right) &\le h_t(p,q)\\ &\le \frac{C}{\mu(B_{\sqrt{t_0}}(p))}\exp\left(-\frac{d^2(p,q)}{(2+\varepsilon)t}\right)\exp\left(-(1+\varepsilon)\Lambda t\right),\end{aligned}$$

for each p,q which are joined by an arc γ in Y of arc length $d(p,q)$. Here $R=d(\gamma,X\setminus Y)$, $t_0=\inf\{t,d^2(p,X\setminus Y),d^2(q,X\setminus Y)\}$ and Λ is the bottom of the spectrum of the operator A on $L^2(X,\mu)$.

2.2 A functional architecture defined by RPs

In this section, we first give the main definitions at the basis of our model, whose basic idea is the construction of a metric space, encoding the local geometry of the cortex, induced by the RPs of simple cells. The main new feature of this model is that the cortical geometry is entirely defined by the profiles: this means that any bank of filters can induce a connectivity pattern through this technique. The space on which the distance function will be defined is the *feature space* $\mathscr{G}$ indexing a family of filters $\{\psi_p\}_{p\in\mathscr{G}}$ chosen to model the RPs of V1 simple cells. We shall provide details on the behavior of this *cortical distance* in the classical case of a family of Gabor filters of fixed scale. As remarked in Section 1.1.2, in this case the feature space is $\mathbb{R}^2\times S^1$: in effect, we will show that the distance function induced by Gabor filters on this space is locally equivalent to a Riemannian distance on $\mathbb{R}^2\times S^1$; this in turn approximates the Carnot-Carathéodory distance associated to the sub-Riemannian structure defined in [41]. We will then show that the application of our construction to a surface in $\mathbb{R}^2\times S^1$ (defined e.g. by an orientation map $\Theta(x,y)$) provides, as a limit case, an example of a cortical metric which cannot be described through differential structures.

2.2.1 The space of features as a metric space

We start by fixing a bank of linear filters $\{\psi_p\}_{p\in\mathscr{G}}\subseteq L^2(\mathbb{R}^2)$. In the following, we define a metric structure on the set of parameters $\mathscr{G}$ associated to $\{\psi_p\}_p$.

Definition 2.19. *Let $\{\psi_p\}_{p\in\mathscr{G}}$ be a family of real- or complex-valued functions in $L^2(\mathbb{R}^2)$. We call $\mathscr{G}$ the* feature space *associated to the family $\{\psi_p\}$.*
We then define the distance function $d:\mathscr{G}\times\mathscr{G}\longrightarrow\mathbb{R}$,

$$d(p,p_0):=\|\psi_p-\psi_{p_0}\|_{L^2(\mathbb{R}^2)}, \tag{2.16}$$

and the generating kernel *$K:\mathscr{G}\times\mathscr{G}\longrightarrow\mathbb{R}$,*

$$K(p,p_0):=Re\langle\psi_p,\psi_{p_0}\rangle_{L^2}. \tag{2.17}$$

This kernel is the reproducing kernel [10] on the closure of the linear span of $\{\psi_p\}_p$ in $L^2(\mathbb{R}^2)$ (see also [52]); it extends to a general family of filters the familiar idea of the reproducing kernel associated to a family of wavelets [9, 50]. In the special case where $\mathscr{G}$ has a group structure and the filters are given by $\psi_p=\mathscr{U}_p\psi$ for a fixed *mother filter* ψ through a

unitary group representation $\mathscr{U}$, one has

$$K(p,q) = K(p^{-1}q, \mathbf{1}). \tag{2.18}$$

This is indeed the case for the example of a bank of Gabor filters that will be examined later. Note that

$$d^2(p,p_0) = \|\psi_p - \psi_{p_0}\|^2_{L^2} = \|\psi_p\|^2_{L^2} + \|\psi_{p_0}\|^2_{L^2} - 2Re\langle\psi_p, \psi_{p_0}\rangle_{L^2}.$$

If we assume the filters to be normalized to have L^2-norm equal to t, the above expression only depends on the real part of the inner product between the two filters, that is on the kernel K:

$$d^2(p,p_0) = 2\big(t - K(p,p_0)\big). \tag{2.19}$$

The value of K at a couple of points increases as they get closer according to the distance d: therefore, $K(p,p_0)$ can be thought of as a measure of *correlation* between p and p_0 with respect to d.

The function we defined is obviously a distance on $\mathscr{G}$, since it is a restriction of the L^2 distance function. However, one may want to introduce some constraints on which filters can directly interact with one another in determining the geometry of the space – for instance, this can be done to inspect the behavior of the connectivity w.r.t. certain features encoded in the RPs. We will see a concrete example of this situation in the case of Gabor filters, where we will be able to *isolate* the spreading of neural activity along the axis of the preferred orientation of the starting RP, while discarding the contributions along the orthogonal axis. Imposing such constraints corresponds to defining around each point $p_0 \in \mathscr{G}$ a *local patch* $\mathscr{P}(p_0) \subseteq \mathscr{G}$, and to restrict the definition of d to this set. Given a local distance defined around each point, one may ask whether it is possible to glue all these distances together to obtain a global distance function on the feature space. We first define a new function $\tilde{d}$ as follows.

Definition 2.20. *For every $p, p_0 \in \mathscr{G}$, if there exists a sequence $\{q_j\}_{j=1,\dots,N}$ such that $q_0 = p_0$, $q_N = p$ and $q_j \in \mathscr{P}(q_{j-1})\,\forall j = 1,\dots,N$, we define*

$$\tilde{d}(p,p_0) := \inf\left\{ \sum_{j=1}^{N} d(q_{j-1},q_j) \,:\, N \in \mathbb{N},\, q_0 = p_0,\, q_N = p,\, q_j \in \mathscr{P}(q_{j-1})\,\forall j \right\}. \tag{2.20}$$

Otherwise, we set $\tilde{d}(p,p_0) := +\infty$.

Note that, in general, the existence of a sequence $\{q_j\}_{j=1,\dots,N}$ such that $q_0 = p_0$, $q_N = p$ and $q_j \in \mathscr{P}(q_{j-1}) \forall j = 1,\dots,N$ is not guaranteed for any couple of points (p, p_0). However, this would be a "degenerate" case where there are isolated points or regions of the feature space, corresponding to neurons whose activations are mutually independent.

Proposition 2.2. Given a set $\mathscr{G}$, define around each point p_0 a *patch* $\mathscr{P}(p_0) \subseteq \mathscr{G}$ such that

$$\forall p_0 \in \mathscr{G} \quad \exists \varepsilon > 0 \,:\, B_\varepsilon(p_0) := \{p \in \mathscr{G} \,:\, d(p,p_0) < \varepsilon\} \subseteq \mathscr{P}(p_0). \tag{2.21}$$

Then $\tilde{d} : \mathscr{G} \times \mathscr{G} \longrightarrow \mathbb{R}$ defined as above satisfies:

(i) $\tilde{d}(p,q) \geq 0 \quad \forall p,q \in \mathscr{G}$,

(ii) $\tilde{d}(p,s) + \tilde{d}(s,q) \geq \tilde{d}(p,q) \quad \forall p,s,q \in \mathscr{G}$,

(iii) $\forall p,q \in \mathscr{G}$, $\tilde{d}(p,q) = 0 \Leftrightarrow p = q$.

Proof. First, $\tilde{d}$ is well-defined. This means verifying that the local distance functions coincide on overlapping patches. Indeed, this happens by construction, since $d(p,p_0)$ is always equal to the L^2 distance between ψ_p and ψ_{p_0}.
Second, $\tilde{d}$ verifies the properties.

(i) $\tilde{d}$ is obviously non negative.

(ii) As for the triangle inequality, we have:

$$\begin{aligned}
&\tilde{d}(p,s) + \tilde{d}(s,q) \\
&= \inf \left\{ \sum_{j=1}^{N} d(q_{j-1},q_j) \,|\, N \in \mathbb{N},\, q_0 = q,\, q_N = p,\, q_j \in \mathscr{P}(q_{j-1}) \forall j, \exists j : q_j = s \right\} \\
&\geq \inf \left\{ \sum_{j=1}^{N} d(q_{j-1},q_j) \,|\, N \in \mathbb{N},\, q_0 = q,\, q_N = p,\, q_j \in \mathscr{P}(q_{j-1}) \forall j \right\} = \tilde{d}(p,q).
\end{aligned}$$

(iii) Lastly, we have to prove that $\tilde{d}(p,p_0) = 0 \Leftrightarrow p = p_0$. Suppose $p \neq p_0$. From (2.21), there exists an $\varepsilon > 0$ such that $B_\varepsilon(p_0) \subseteq \mathscr{P}(p_0)$. Now,

- if $p \notin \mathscr{P}(p_0)$, then $p \notin B_\varepsilon(p_0)$ and consequently $\tilde{d}(p,p_0) \neq 0$;
- on the other hand, if p is in $\mathscr{P}(p_0)$, then $\tilde{d}(p,p_0) \neq 0$ for the properties of d, which is a distance on $\mathscr{P}(p_0)$.

□

Note that, given a sequence $p_0 = q_0, q_1, \dots, q_N = p$, the condition $q_j \in \mathscr{P}(q_{j-1})$ does not imply having $q_{j-1} \in \mathscr{P}(q_j)$. Therefore, in general, (2.20) yields a *quasimetric* $\tilde{d}$, i.e. an asymmetric distance. This intuitively means that getting from p to p_0 may be harder than getting from p_0 to p, i.e. $\tilde{d}(p, p_0) > \tilde{d}(p_0, p)$. A simple practical example of a quasimetric $\tilde{d}$ is given by the walking times between points on a mountain: when p_0 is uphill w.r.t. p, $\tilde{d}(p, p_0)$ is greater that $\tilde{d}(p_0, p)$. See [148] as a reference on quasimetric spaces.
However, recall that the distance we are defining should model the lateral connectivity in V1. Due to the evidence that horizontal connections are largely reciprocal [88], it is reasonable to model this phenomenon through a symmetric distance. Since the construction of the patches $\mathscr{P}(\cdot)$ was meant to restrict which cells can interact with one another, it is natural to define them so that p is connected to q if and only if q is connected to p. This means requiring that

$$q \in \mathscr{P}(p) \Leftrightarrow p \in \mathscr{P}(q),$$

which implies the symmetry of $\tilde{d}$ by considering for each sequence $p = q_0, q_1, \dots, q_N = q$ the reversed sequence $\{q_{N-j}\}_{j=0,\dots,N}$. In the following, the symmetry is taken as an assumption.

To sum up, the kernel distance defined in (2.16) may be treated as a local object by restricting it to suitable patches defined around each point. In order to have a meaningful distance on the whole feature space taking into account these constraints, the local distance functions must be glued together: the above Proposition states that, under reasonable conditions on the choice of the patches, this yields a well-defined global distance on $\mathscr{G}$.

2.2.2 The case of Gabor filters

As a first example, we show the results of applying the model described above to the classical case of a bank of Gabor filters. We then prove that the distance obtained in this case is locally equivalent to a Riemannian approximation to the sub-Riemannian metric introduced in [41].
Let us consider the set $\{\psi_{x,y,\theta}\}_{x,y,\theta}$ of Gabor filters introduced in (1.2). For each value of $\lambda > 0$ and $\sigma > 0$, one obtains a family of filters parameterized by $p = (x, y, \theta) \in \mathbb{R}^2 \times S^1$ where each of the filters $\psi_{x,y,\theta}$ has wavelength λ and scale σ.

The distance function.

Fix $\lambda, \sigma > 0$ and denote $p = (x, y, \theta)$ and $p_0 = (x_0, y_0, \theta_0)$. We first note that the kernel $K(p, p_0) = Re\langle \psi_p, \psi_{p_0} \rangle_{L^2(\mathbb{R}^2)}$ shows certain invariances in this case. Specifically, as pointed

out before, the explicit expression for $K(\cdot, p_0)$ for $p_0 \neq (0,0,0)$ can be obtained from the one for $K(\cdot,0)$ through the group operation. We have:

$$K\big((x,y,\theta),(x_0,y_0,\theta_0)\big) = K\big((R_{\theta_0}T_{(x_0,y_0)}(x,y),\theta-\theta_0),\ (0,0,0)\big).$$

It is therefore sufficient to compute explicitly the expression of $K(p,p_0)$ for $p_0=(0,0,0)$. We have:

$$\langle \psi_p, \psi_0 \rangle_{L^2(\mathbb{R}^2)} = \sigma^2\pi \exp\left(-\frac{x^2}{4\sigma^2} - \frac{y^2}{4\sigma^2} - \frac{2\sigma^2\pi^2(1-\cos\theta)}{\lambda^2}\right) \exp\left(-i\pi\,\frac{x(1+\cos\theta)+y\sin\theta}{\lambda}\right).$$

The real part of this scalar product gives the kernel K. Of course, the same invariance holds for the distance d. Since the squared L^2-norm of each of the filters (1.2) is equal to $\sigma^2\pi$, we have:

$$d^2(p,0) = 2\sigma^2\pi - 2\sigma^2\pi \exp\left(-\frac{x^2}{4\sigma^2} - \frac{y^2}{4\sigma^2} - \frac{2\sigma^2\pi^2(1-\cos\theta)}{\lambda^2}\right)\cdot\cos\left(\pi\,\frac{x(1+\cos\theta)+y\sin\theta}{\lambda}\right). \tag{2.22}$$

Note that the distance d depends on σ and λ, since the scale and wavelength of the filters naturally influence its spatial extent and oscillatory behavior respectively.

Local patches.

In the case of Gabor filters, the level sets of the distance d (or equivalently, of the kernel K) in the feature space are in general not connected (see Figure 2.1). This is due to the oscillations of the periodic factor. In terms of the interactions between RPs, the central lobe of the level sets corresponds to the *co-axial* effect, i.e. the one along the axis of the preferred orientation of the starting RP ψ_{p_0}: along this axis, the strongest interactions are the ones with filters collinear with ψ_{p_0} (see Figure 2.2b), and the value of K decreases as the orientation of the filters varies from θ_0. On the other hand, the smaller lobes developing along the

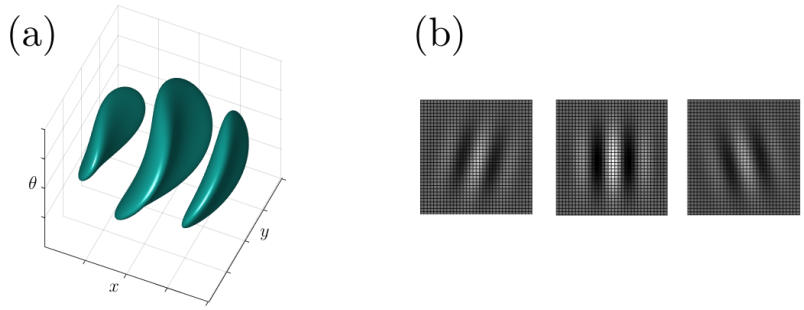


Figure 2.1 (a) A level set of $K\big((x,y,\theta),(0,0,0)\big)$. (b) Three slices for $\theta = -\frac{\pi}{6}, 0, \frac{\pi}{6}$.

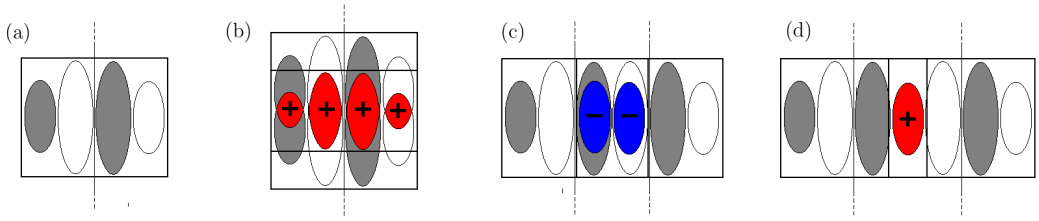


Figure 2.2 A schematic representation of the interactions between odd RPs sharing the same orientation, depending on their reciprocal position. (a) An example odd filter, with white ON areas and gray OFF areas. The axis of its preferred orientation is displayed. The next pictures show different reciprocal configurations of two such filters. Overlapping areas where the sign of the product is positive are displayed in red and marked by a plus symbol, while zones where the product is negative are blue and marked by a minus symbol. (b) Collinear filters bring to high values of K along their common axis. (c) Parallel filters with overlapping regions of opposite signs yield negative values of K. (d) Parallel filters with overlapping regions of the same sign yield positive values of K.

orthogonal axis result from the "periodic" correlation of the central RP with filters parallel to ψ_{p_0} (i.e. filters with orientation θ_0 but centered on points of the orthogonal axis): the kernel takes positive values when areas of the two RPs with the same sign overlap (as in Figure 2.2d), and negative values when areas of opposite signs overlap (as in Figure 2.2c). In a first stage we thought it best to restrict ourselves to the effect along the axis of the preferred orientation, i.e. to consider only the central connected component of the level sets. This only takes a straightforward analytical operation in the Gabor case, and it makes it easier to compare our model with the ones obtained through real-valued diffusion equations. We shall therefore restrict the distance to local patches defined around each point $p_0 = (x_0, y_0, \theta_0)$ of $\mathbb{R}^2 \times S^1$ such that the distance is truncated where it reaches its maximum, thus eliminating the periodicity of the cosine in Eq. (2.22):

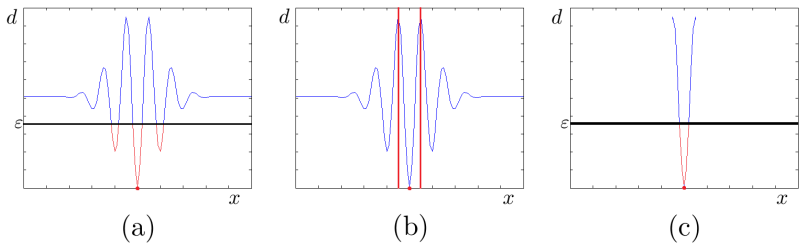


Figure 2.3 (a) For fixed $y = 0$ and $\theta = 0$, a plot of $x \mapsto d((x,0,0),(0,0,0))$. In red, the corresponding slice of a neighborhood $B_\varepsilon\big((0,0,0)\big) = \{(x,y,\theta) \in \mathbb{R}^2 \times S^1 : d((x,y,\theta),(0,0,0)) < \varepsilon\}$, which is not connected. (b) We truncate the distance function at its maximum. (b) The neighborhood of the same radius as before, with the truncated distance, turns out to be connected. (d) The non-connected ball $B_\varepsilon\big((0,0,0)\big)$ (dark blue) displayed in the 3D space $\mathbb{R}^2 \times S^1$. The patch $\mathscr{P}(0,0,0)$ is represented by the volume between the two light blue surfaces. After truncating the distance function, only the central lobe remains. In this example we set $\lambda = 1$ and $\sigma = 1$.

$$\mathscr{P}(p_0) := \{(x,y,\theta) \,:\, |a(1+\cos\delta)+b\sin\delta| < \lambda\},$$

with $(a,b,\delta) = \big(R_{\theta_0}T_{(x_0,y_0)}(x,y),\, \theta-\theta_0\big)$. Of course, the thickness of the patch depends on the frequency of the oscillations of the distance, ruled by the wavelength λ of the filters. Figure 2.3 schematically displays this operation on a plot of the distance function with respect to x, for fixed values of y and θ. The shape of the patches is shown in Figure 2.4.
For each $p, p_0 \in \mathbb{R}^2 \times S^1$ we then consider the distance $\tilde{d}(p,p_0)$ as defined in (2.20). Note that those neighborhoods which are "small enough" are connected even without truncating the distance function (see Figure 2.3a). In other words, there always exists an $\varepsilon > 0$ such that $B_\varepsilon(p_0) \subseteq \mathscr{P}(p_0)$, i.e. the local structure is preserved despite the clipping. In terms of the correlation kernel, this is equivalent to saying that the reciprocal influence of collinear filters is greater that the one between parallel filters, since all the values of $K(\cdot,p_0)$ above a sufficiently high threshold are confined inside $\mathscr{P}(p_0)$. This property, together with the symmetry of the patches, makes (2.20) a global distance on $\mathbb{R}^2 \times S^1$ (see Proposition 2.2 and the following remark). Moreover, note that a finite sequence $\{q_j\}_{j=0,\dots,N}$ connecting two points always exists. For $p_0 = (0,0,0)$ and $p = (x,y,\theta)$, take for example:

$$q_0 = (0,0,0) = p_0, \quad q_1 = (0,y,0), \quad q_2 = \left(0,y,\frac{\pi}{2}\right), \quad q_3 = \left(x,y,\frac{\pi}{2}\right) \quad q_4 = (x,y,\theta) = p.$$

The distance is therefore finite.

This definition provides an example of adjustment that can be introduced on the original kernel in order to define some restriction on which filters are influenced by one another. Nonetheless, we believe that the whole kernel is interesting for a further analysis, since its

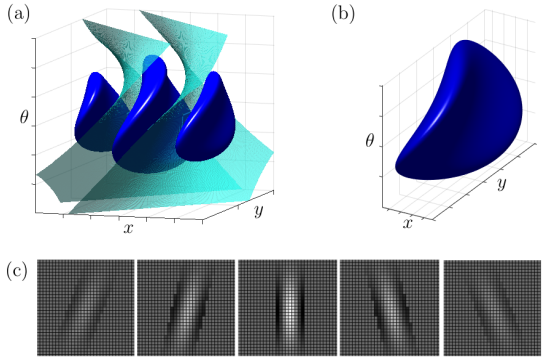


Figure 2.4 (a) In dark blue, a level set of $K\big((x,y,\theta),(0,0,0)\big)$. The patch $\mathscr{P}(0,0,0)$ is the volume between the two surfaces displayed in light blue. (b) The same level set, after truncating. Here, $\lambda = 1$. (c) Horizontal slices of the truncated kernel for $\theta = -0.75, -0.45, 0, 0.45, 0.75$. The white regions represent positive values, the black ones represent negative values. Note that the kernel has been truncated at its minimum.

oscillatory behavior in the orthogonal direction seems to account for the *trans-axial* "ladder" effect which has indeed been observed in both neurophysiological and psychophysical studies [106, 61, 152] (see also Figure 1.12 in Section 1.2.2), thus naturally including in the model the Gestalt perceptual principle of parallelism.

Local estimate of d^2.

Let us study the *local* behavior of the distance function d induced by the filters (1.2) on $\mathbb{R}^2 \times S^1$. The following proposition states that d is locally estimated by a Riemannian distance on $\mathbb{R}^2 \times S^1$.

Proposition 2.3. There exists a Riemannian metric g on $\mathbb{R}^2 \times S^1$ whose induced distance d_g satisfies:

$$\frac{d(p,p_0)}{d_g(p,p_0)} \to 1 \quad \text{as} \quad d_g(p,p_0) \to 0.$$

Moreover, the Riemannian volume form on $(\mathbb{R}^2 \times S^1, g)$ is a constant multiple of the Lebesgue measure.

Proof. Fix $p = (x,y,\theta) \in \mathbb{R}^2 \times S^1$, and let $x,y,\theta \to 0$. We have:

- $\exp\left(-\frac{x^2}{4\sigma^2} - \frac{y^2}{4\sigma^2} - \frac{2\sigma^2\pi^2(1-\cos\theta)}{\lambda^2}\right) \approx 1 - \frac{x^2}{4\sigma^2} - \frac{y^2}{4\sigma^2} - \frac{2\sigma^2\pi^2}{\lambda^2}\frac{\theta^2}{2}$.
- $\cos\left(\pi\frac{(x(1+\cos\theta)+y\sin\theta)}{\lambda}\right) \approx 1 - \frac{2\pi^2}{\lambda^2}x^2$.

Then

$$d^2(p,0) \approx 2\sigma^2\pi\left(\left(\frac{1}{4\sigma^2} + \frac{2\pi^2}{\lambda^2}\right)x^2 + \frac{y^2}{4\sigma^2} + \frac{\sigma^2\pi^2}{\lambda^2}\theta^2\right).$$

More generally, for $p = (x,y,\theta) \to (x_0,y_0,\theta_0) = p_0$,

$$d^2(p,p_0) \approx 2\sigma^2\pi\left(\left(\frac{1}{4\sigma^2} + \frac{2\pi^2}{\lambda^2}\right)a^2 + \frac{1}{4\sigma^2}b^2 + \frac{\sigma^2\pi^2}{\lambda^2}(\theta-\theta_0)^2\right),$$

where $(a,b) = R_{\theta_0}T_{(x_0,y_0)}(x,y)$. Equivalently,

$$d^2(p,p_0) \approx (x-x_0,\, y-y_0,\, \theta-\theta_0)\cdot g(p_0)\cdot \begin{pmatrix} x-x_0 \\ y-y_0 \\ \theta-\theta_0 \end{pmatrix}$$

where

$$g(p_0) = 2\sigma^2\pi \begin{pmatrix} \left(\frac{1}{4\sigma^2} + \frac{2\pi^2}{\lambda^2}\right)\cos^2\theta_0 + \frac{1}{4\sigma^2}\sin^2\theta_0 & \frac{2\pi^2}{\lambda^2}\cos\theta_0\sin\theta_0 & 0 \\ \frac{2\pi^2}{\lambda^2}\cos\theta_0\sin\theta_0 & \left(\frac{1}{4\sigma^2} + \frac{2\pi^2}{\lambda^2}\right)\sin^2\theta_0 + \frac{1}{4\sigma^2}\cos^2\theta_0 & 0 \\ 0 & 0 & \frac{\sigma^2\pi^2}{\lambda^2} \end{pmatrix}. \tag{2.23}$$

Finally, for every point p_0,

$$\det g(p_0) = 8\sigma^6\pi^3\left(\frac{1}{4\sigma^2} + \frac{2\pi^2}{\lambda^2}\right)\frac{1}{4\sigma^2}\frac{\sigma^2\pi^2}{\lambda^2}.$$

This concludes the proof. □

Convergence to a sub-Riemannian metric.

Finally, we show that the metric g computed above is a Riemannian approximation to a sub-Riemannian structure on $\mathbb{R}^2 \times S^1$ which is, up to constants, the same as the one defined in [41]. More precisely, we prove:

Theorem 2.6. Let $\sigma^2 = A\lambda$ for some $A > 0$. Then the distance d_g induced by the metric g in (2.23) converges uniformly on the compact sets of $\mathbb{R}^2 \times S^1$, as $\lambda \longrightarrow 0$, to the Carnot-Carathéodory distance induced by the vector fields

$$\tilde{Y}_1 = \sqrt{\frac{2A}{\pi}}\left(-\sin\theta\,\partial_x + \cos\theta\,\partial_y\right) \quad \text{and} \quad \tilde{Y}_2 = \frac{1}{\sqrt{2A\pi^3}}\,\partial_\theta, \tag{2.24}$$

with the horizontal norm that makes them orthonormal.

Proof. We first plug the condition $\sigma^2 = A\lambda$ into the expression of the Riemannian metric g. The distance induced by g is the Carnot-Carathéodory distance associated to the vector fields $\tilde{Y}_1, \tilde{Y}_2$ and

$$\tilde{Y}_3 := \frac{1}{L}\cdot\left(\cos\theta\,\partial_x + \sin\theta\,\partial_y\right) = \frac{1}{L}\cdot Y_3,$$

where $L := \sqrt{\frac{\pi}{2} + \frac{4A\pi^3}{\lambda}}$. The associated norm on the tangent space $T_p\left(\mathbb{R}^2 \times S^1\right)$ at each point p is

$$|v|^2 = v_1^2 + v_2^2 + v_3^2, \qquad \text{where } \sum_{i=1}^{3} v_i\tilde{Y}_i = v \in T_p\left(\mathbb{R}^2 \times S^1\right).$$

Note that $L \to +\infty$ as $\lambda \to 0$. The vector fields $(\tilde{Y}_1, \tilde{Y}_2, \tilde{Y}_3) = (\tilde{Y}_1, \tilde{Y}_2, \frac{1}{L}Y_3)$ define a Riemannian approximation to the sub-Riemannian metric defined by $(\tilde{Y}_1, \tilde{Y}_2)$, and the thesis follows from

the approximation theorem in [75], Section 1.4.D.

□

Remark 2.13. Note that the vector fields $\tilde{Y}_1$ and $\tilde{Y}_2$ generate the horizontal distribution of the sub-Riemannian structure on $\mathbb{R}^2 \times S^1$ introduced in [41]. In particular, $\tilde{Y}_1 = \sqrt{\frac{2A}{\pi}} \cdot Y_1$ and $\tilde{Y}_2 = \sqrt{\frac{1}{2A\pi^3}} \cdot Y_2$, where Y_1 and Y_2 are as in (1.4). Moreover, $\tilde{Y}_3 := \frac{1}{L} \cdot Y_3$, where $-Y_3 = [Y_1, Y_2]$.

Remark 2.14. The constraint $\sigma^2 = A\lambda$ implies, for $\lambda \to 0$, that the support of the filters shrinks and the number of oscillations under the Gaussian bell goes to infinity.

2.2.3 A non-differential example

The example that we are about to introduce is a relevant one since it represents an instance of feature space whose metric cannot be described through a differential structure, thus motivating our work in the more general setting of metric spaces.
Let us consider a surface

$$\Sigma = \big\{(x,y,\theta) \in \mathbb{R}^2 \times S^1 \ : \ \theta = \Theta(x,y)\big\}, \tag{2.25}$$

and the corresponding subset $\{\psi_{x,y,\Theta(x,y)}\}_{x,y}$ of the above-mentioned family of Gabor filters. This yields a feature space $\mathscr{G} = \mathbb{R}^2$, endowed with the metric structure defined by this subfamily of filters, i.e.

$$d\big((x,y),(x_0,y_0)\big) := \|\psi_{x,y,\Theta(x,y)} - \psi_{x_0,y_0,\Theta(x_0,y_0)}\|_{L^2(\mathbb{R}^2)}. \tag{2.26}$$

The restriction to Σ of a distance which is estimated by a Riemannian one is still estimated by the induced Riemannian metric on the surface. However, setting $\sigma^2 = A\lambda \to 0$ as before yields a sub-Riemannian structure on $\mathbb{R}^2 \times S^1$, determined by the vector fields $\tilde{Y}_1$ and $\tilde{Y}_2$ of Theorem 2.6 with the following norm on the horizontal planes:

$$|v|_H^2 = v_1^2 + v_2^2, \quad \text{where } \ v_1\tilde{Y}_1 + v_2\tilde{Y}_2 = v \ \in H_p. \tag{2.27}$$

For each p the horizontal plane H_p is the subspace of $T_p\big(\mathbb{R}^2 \times S^1\big)$ generated by Y_1 and Y_2. We work on the domain (x,y) of the function Θ defining the surface. Now consider, as in [40],

the projections $V := Y_{1\,\Theta}$ and $W := Y_{3\,\Theta}$ of the vector fields Y_1, Y_3 on the plane $P = \{(x,y,0)\}$:

$$V_{(x,y)} = Y_{1\,\Theta\,(x,y)} = -\sin(\Theta(x,y))\partial_x + \cos(\Theta(x,y))\partial_y$$
$$W_{(x,y)} = Y_{3\,\Theta\,(x,y)} = \cos(\Theta(x,y))\partial_x + \sin(\Theta(x,y))\partial_y.$$

The vector fields V and W span the plane P. Note that the surface Σ is foliated by integral curves of V, and the restriction of the horizontal norm (2.27) onto this surface would yield a degenerate distance whose balls are segments of curves.
The distance we want to consider on Σ is instead the one whose balls are obtained by intersecting Σ with the balls of the sub-Riemannian metric on $\mathbb{R}^2 \times S^1$ – i.e. the distance induced on Σ as a metric subspace of $\mathbb{R}^2 \times S^1$ with the Carnot-Carathéodory distance. At each point $p_0 \in \mathbb{R}^2 \times S^1$, the exponential mapping $\exp_{p_0} : \mathfrak{g} \to \mathbb{R}^2 \times S^1$ is defined by $\exp_{p_0}(X) = \gamma_X(1, p_0)$, where $\mathfrak{g}$ is the Lie algebra associated to $\mathbb{R}^2 \times S^1 = SE(2)$ as a Lie group (see [145]) and $\gamma_X(\cdot, p_0)$ is the unique solution to the Cauchy problem

$$\begin{cases} \dot{\gamma}(t) = X_{|\gamma(t)} \\ \gamma(0) = p_0. \end{cases}$$

For sufficiently small t, $\exp_{p_0}(tX) = \gamma_{tX}(1, p_0) = \gamma_X(t, p_0)$ is always well defined. Moreover, $\exp_{p_0}$ is a local diffeomorphism [145]. We can thus define *locally* on $\mathbb{R}^2 \times S^1$ the distance

$$d_Y^2(p, p_0) = v_1^2 + v_2^2 + |v_3|,$$

where $v_1, v_2, v_3 \in \mathbb{R}$ are such that $p = \exp_{p_0}\left(\sum_{i=1}^3 v_i Y_i\right)$. This distance is locally equivalent to the Carnot-Carathéodory distance d_{cc} on $\mathbb{R}^2 \times S^1$ [113]. Restricted on the domain of Θ, this becomes

$$d_\Sigma^2\big((x,y),(x_0,y_0)\big) = e_1^2 + |e_2|, \tag{2.28}$$

where $(x,y) = \exp_{(x_0,y_0)}\left(e_1 V + e_2 W\right)$ (see [40]). Note that the balls of this distance are indeed open sets of the surface.

Surfaces play a key role in modeling the visual cortex. A first example is given by a surface of maxima such as the one introduced in [41]. Another important instance is represented by the surface defined through an orientation map Θ of V1, see Section 1.1.2. We shall return to this example in the next Section, whose main subject will be the horizontal connectivity of V1. As already mentioned in Section 1.3, a possible way to represent this connectivity is by means of a diffusion process: in some differential models of V1, this

diffusion is expressed through second order operators associated to the sub-Riemannian structure taken into consideration. In order to still be able to use this approach in non-differential cases such as the one described above, we aim at extending it to the context of metric measure spaces.

2.3 Connectivity

A central aspect in modeling the visual cortex is the characterization of how the activity of a neuron is influenced by the surrounding cells. As outlined in Section 1.3, in many existing mathematical models of the visual cortex, the feature space associated to V1 simple cells is equipped with a sub-Riemannian structure. Starting from this local geometry, the idea is to describe the spreading of horizontal connections around each neuron through a diffusion equation (e.g. the sub-Riemannian heat equation [41] or the Fokker-Planck equation [127]) associated to the geometry of the space. Such constructions inspired us to give an analogous description of the lateral connectivity through a suitable concept of diffusion linked to the geometric structure of our space.
In our model, the feature space is equipped with a metric structure defined by the receptive profiles themselves. Starting from a general family of filters, we cannot expect the distance obtained to be compatible with some differential structure. We shall then address the issue in the much more general setting of *metric measure spaces*, following the classical approach of K.-T. Sturm [136, 137] recalled in Section 2.1.

2.3.1 The cortical metric measure space

Let us recall our setting: we have defined a metric space $(\mathscr{G}, d)$, where $\mathscr{G}$ is the feature space indexing a family $\{\psi_p\}_{p\in\mathscr{G}}$ of linear filters on the plane, and d is the distance function of Definition 2.20. The first step in order to be able to do some analysis on $(\mathscr{G}, d)$ is to equip it with a suitable measure. This has to be related to some notion of *density* of the filters with respect to the distance d. Moreover, the MCP of [137] (see Section 2.1) expresses a link between the metric balls and the measure. Therefore, a quite natural choice is the spherical Hausdorff measure (see Section 2.1) associated to the distance d. We shall denote it by μ. Suppose now that $(\mathscr{G}, d)$ is a locally compact separable metric space, that μ is a Radon measure with full support on X and that the metric measure space (X, d, μ) satisfies the MCP. This yields Gaussian estimates for the heat kernel h_t associated to the diffusion process defined by the Dirichlet form E, meaning that in this case one has an approximate

version of h_t expressed *explicitly* in terms of the cortical distance d.

The first comment to be made is about the nice behavior of the spherical Hausdorff measure and of the MCP in the event of equivalent (or locally equivalent) distances. Indeed, the following holds.

Proposition 2.4. The weak MCP for the spherical Hausdorff measure is invariant under local equivalence of distances.

Proof. If d and d' are two distances defined on X with corresponding spherical Hausdorff measures μ and μ', then we have [60]:

$$\exists \kappa > 0 \ : \kappa^{-1} d(x,y) \leq d'(x,y) \leq \kappa\, d(x,y) \quad \forall x,y \in X \tag{2.29}$$

$$\Longrightarrow \quad \kappa^{-s} \mu(A) \leq \mu'(A) \leq \kappa^{s}\, \mu(A) \quad \text{for any Borel set } A \subseteq X, \tag{2.30}$$

where s is the Hausdorff dimension. The same holds locally if the distances are only locally equivalent.
Suppose now that the MCP is verified for (X,d,μ). The exceptional set Z is obviously still a null set with respect to μ'. Fix a compact set $Y \subseteq X \setminus Z$. Note that, even if the equivalence is just local, the compactness of Y allows to have (2.29) with the same κ over all Y. The maps Φ_t are still measurable and verify the properties (i). In particular (2.11) holds thanks to the equivalence of the distances. Recall that the doubling property holds for d, μ and let M be a doubling constant:

$$\mu(B_{\kappa r}(x)) \leq M \mu(B_r(x)).$$

Now, (2.30) implies:

$$(\kappa^s M)^{-1} \mu(B_r(x)) \leq \mu'(B'_r(x)) \leq \kappa^s M\, \mu(B_r(x)) \quad \text{and} \quad (\kappa^{\frac{3s}{2}} M)^{-1} \mu_r(A) \leq \mu'_r(A) \leq \kappa^{\frac{3s}{2}} M\, \mu_r(A).$$

Finally, by these inequalities and the property (2.12) for d and μ, we have:

$$\frac{\mu'_r(A)}{\sqrt{\mu'(B'_r(x))}} \leq \frac{\kappa^{2s} M^{\frac{3}{2}}\, \mu_r(A)}{\sqrt{\mu(B_r(x))}} \leq \frac{\zeta\, \kappa^{2s} M^{\frac{3}{2}}\, \mu_{rt}(\Phi_t(x,A))}{\sqrt{\mu(B_{rt}(x))}} \leq \frac{\zeta\, \kappa^{4s} M^3 \mu'_{rt}(\Phi_t(x,A))}{\sqrt{\mu'(B'_{rt}(x))}},$$

i.e.

$$\frac{\mu'_r(A)}{\sqrt{\mu'(B'_r(x))}} \leq C\, \frac{\mu'_{rt}(\Phi_t(x,A))}{\sqrt{\mu'(B'_{rt}(x))}}.$$

□

In fact, if the (local) equivalence constant κ of the two distances can be locally chosen to be arbitrarily close to 1, then even the strong MCP is preserved. These facts are of particular importance for our purposes: by Proposition 2.3, the cortical distance arising from a set of Gabor filters is locally equivalent to a Riemannian distance on $\mathbb{R}^2 \times S^1$, with a local equivalence constant approaching 1. Recall that all Riemannian manifolds (M, g) are locally compact as metric spaces with the geodesic distance d_g [2], and that this property is preserved in metric spaces under local equivalence of distances by Proposition 2.4. Moreover, the MCP holds [137] on (M, d_g, μ_g) where μ_g is the Riemannian measure, which coincides uo to a constant with the spherical Hausdorff measure associated to d_g. This immediately leads to the following result.

Theorem 2.7. The cortical metric measure space $(\mathbb{R}^2 \times S^1, d, \mu)$ defined by the bank of Gabor filters (1.2) satisfies the MCP.

2.3.2 The MCP for a sub-Riemannian surface in $\mathbb{R}^2 \times S^1$

We now go back to the example, introduced in Section 2.2.3, of a sub-Riemannian surface in $\mathbb{R}^2 \times S^1$. We show that such a space satisfies the MCP, thus providing an example of a non-differential feature space on which the horizontal connectivity can still be represented through a suitable diffusion process. Specifically, we prove the following theorem.

Theorem 2.8. Consider a surface Σ as in (2.25), whose defining function $\Theta(x, y)$ is C^1, except possibly for a discrete set $\Pi \subset \mathbb{R}^2$. Denote $Z := \{(x, y, \Theta(x, y)) : (x, y) \in \Pi\} \subseteq \Sigma$. Then Σ verifies the MCP with exceptional set Z.

Proof. We first verify that (Σ, d_Σ) is locally compact.

(i) $\mathbb{R}^2 \times S^1$ with the Carnot-Carathéodory distance d_{cc} is a locally compact space [2]. Then each closed subset of (Σ, d_Σ) away from Z is locally compact because it is a closed subspace of $(\mathbb{R}^2 \times S^1, d_{cc})$ by the continuity of Θ.

(ii) Now, given $\zeta \in Z$, we need to construct a compact neighborhood of ζ in Σ. Consider a closed ball $\overline{B_\varepsilon^{cc}(\zeta)}$ of d_{cc} in $\mathbb{R}^2 \times S^1$ such that $\overline{B_\varepsilon^{cc}(\zeta)}$ does not contain any other point of Z; then define $B := \overline{B_\varepsilon^{cc}(\zeta)} \cap \Sigma$. This is a neighborhood of ζ in the induced metric d_Σ. We now prove that B is compact.
Given a sequence $\{p_n\}_n \subseteq B \subseteq \overline{B_\varepsilon^{cc}(\zeta)}$, by the compactness of $\overline{B_\varepsilon^{cc}(\zeta)}$ there exists a subsequence $\{p_{n_k}\}_k$ converging to a point $p \in \overline{B_\varepsilon^{cc}(\zeta)}$. If $\zeta \notin \{p_{n_k}\}_k$, then p belongs to B by (i). If $\zeta \in \{p_{n_k}\}_k$, either $p = \zeta \in B$ or $p_{n_k} \neq \zeta$ for $k > \bar{k}$ and the truncated sequence falls into the preceding case.

The measure μ that we consider on Σ is the one given by the sub-Riemannian area [66, 36], since this coincides up to a constant with the spherical Hausdorff measure on (Σ, d_Σ) (see [65, 62]). Specifically, given a subset $S \subseteq \Sigma$,

$$\mu(S) = \int_S |N_h| d\Sigma. \tag{2.31}$$

Here, N_h is the orthogonal projection of a unit vector field normal to Σ onto the horizontal distribution, and $d\Sigma$ is the Riemannian measure of Σ induced by the projected vector fields V and W defined in Section 2.2.3, i.e.

$$d\Sigma(x,y) = \sqrt{\det g_\Sigma(x,y)} dx dy.$$

Now denote $\xi = (x,y)$. We define

$$\Phi_t(\xi, A) = \exp_\xi(t \cdot \exp_\xi^{-1}(A)), \tag{2.32}$$

where $\cdot$ denotes the dilation

$$t \cdot v = \left(te_1, t^2 e_2\right) \quad \forall v = (e_1, e_2) \in P.$$

Given a compact set Y of Σ and $A \subseteq B_r(\xi) \cap Y$, we have:

$$\begin{aligned}
\mu(\Phi_t(\xi, A)) &= \int_{\exp_\xi(t \cdot \exp_\xi^{-1}(A))} d\mu = \int_{\exp_\xi(t \cdot \exp_\xi^{-1}(A))} |N_h(\xi')| \sqrt{\det(g_\Sigma(\xi'))} d\xi' \\
&= \int_{t \cdot \exp_\xi^{-1}(A)} \left|J_\xi(v)\right| |N_h(\exp_\xi(v))| \sqrt{\det(g_\Sigma(\exp_\xi(v)))} dv \\
&= t^3 \int_{\exp_\xi^{-1}(A)} \left|J_\xi(tu)\right| |N_h(\exp_\xi(tu))| \sqrt{\det(g_\Sigma(\exp_\xi(tu)))} du \\
&= t^3 \int_A \frac{\left|J_\xi(t \cdot \exp_\xi^{-1}(\xi'))\right|}{\left|J_\xi(\exp_\xi^{-1}(\xi'))\right|} |N_h(\Phi_t(\xi, \xi'))| \sqrt{\det(g_\Sigma(\Phi_t(\xi, \xi')))} d\xi',
\end{aligned}$$

where $d\xi' = dx'dy'$ and $J_\xi(v)$ denotes the Jacobian determinant of $\exp_\xi$. Then

$$\frac{\mu(\Phi_t(\xi, A))}{\mu(A)} = \frac{\int_A \left|J_\xi(t \cdot \exp_\xi^{-1}(\xi'))\right| \left|J_\xi(\exp_\xi^{-1}(\xi'))\right|^{-1} f(\Phi_t(\xi, \xi')) d\xi'}{\int_A f(\xi') d\xi'} t^3, \tag{2.33}$$

where we have denoted $f = |N_h|\sqrt{\det(g_\Sigma)}$. Now, one has [40, 113]:

$$(1+O(|v|))^{-1} \leq |J_\xi(v)| \leq 1+O(|v|).$$

This yields

$$\frac{(1+t^2O(|\exp_\xi^{-1}(\xi')|))^{-1}}{1+O(|\exp_\xi^{-1}(\xi')|)} \leq \frac{\left|J_\xi(t\cdot\exp_\xi^{-1}(\xi'))\right|}{\left|J_\xi(\exp_\xi^{-1}(\xi'))\right|} \leq \frac{1+t^2O(|\exp_\xi^{-1}(\xi')|)}{(1+O(|\exp_\xi^{-1}(\xi')|))^{-1}}$$

Since $\xi' \in B_r(\xi)$ and $t \in [0,1]$, the initial calculation leads to

$$\frac{\mu(\Phi_t(\xi,A))}{\mu(A)} \geq \frac{\int_A f(\Phi_t(\xi,\xi'))d\xi'}{\int_A f(\xi')d\xi'}\frac{t^3}{1+O(r)}.$$

Finally, $f(\Phi_t(\xi,\xi')) = f(\xi') + O(d_\Sigma(\Phi_t(\xi,\xi'),\xi'))$ and both $\Phi_t(\xi,\xi')$ and ξ' are in $B_r(\xi)$. Then,

$$\frac{\int_A f(\Phi_t(\xi,\xi'))d\xi'}{\int_A f(\xi')d\xi'} = \frac{\int_A f(\xi')d\xi' + O(r)\int_A d\xi'}{\int_A f(\xi')d\xi'} \quad \Rightarrow \quad \frac{\mu(\Phi_t(\xi,A))}{\mu(A)} \geq \frac{t^3}{1+O(r)}.$$

On the other hand, since $B_{rt}(\xi) = \Phi_t(\xi, B_r(\xi))$ and since we have estimates for J_ξ and f both from above and from below, we have:

$$\frac{\mu(B_{rt}(\xi))}{\mu(B_r(\xi))} \leq (1+O(r))\,\frac{\mu(\Phi_t(\xi,A))}{\mu(A)}.$$

□

Remark 2.15. Note that, in the case of a pinwheel surface (see Section 1.1.2), the exceptional set Z of the MCP is represented by the singularities at the center of each pinwheel arrangement.

2.3.3 Propagation through a connectivity kernel

The propagation of neural activity starting from a single neuron may be modeled through the diffusion process described above with a Dirac delta ψ_{p_0} as an initial datum. Thanks to Theorem 2.4, under the hypothesis that the MCP holds on $(\mathscr{G}, d, \mu)$, this is equivalent to considering the heat kernel $h_t(p, p_0)$, which also admits upper and lower Gaussian estimates.

Moreover, in the special case of a compact Riemannian submanifold of the Euclidean space, an arbitrary good approximation of the heat kernel can be provided by iterating the

Gaussian kernel for small t. In [47], an approximating kernel is defined as follows: given an exponentially decaying function h and a parameter $\alpha \in \mathbb{R}$,

$$k_t(p,q) = h\left(\frac{\|p-q\|^2}{t}\right); \quad k_t^{(\alpha)}(p,q) = \frac{k_t(p,q)}{Q_t^\alpha(p)Q_t^\alpha(q)}, \text{ where } Q_t(p) = \int k_t(p,q)Q(q)d\mu(q).$$

Here, Q is a density function expressing the distribution of points in a dataset. A new kernel, depending on the choice of α, is then defined via a normalization:

$$N^{(\alpha)}[k_t](p,q) = \frac{k_t^{(\alpha)}(p,q)}{\int k_t^{(\alpha)}(p,q')Q(q')d\mu(q')}. \tag{2.34}$$

The following theorem is proved (see Proposition 3 in [47]).

Theorem 2.9 (Coifman and Lafon, 2006)**.** Define the operators

$$H_t^{(\alpha)} f(p) := \int N^{(\alpha)}([k_t](p,q)f(q)d\mu(q) \quad \text{and} \quad L_{t,\alpha} f = \frac{1}{t}\left(f - H_t^{(\alpha)} f\right). \tag{2.35}$$

For $\alpha = 1$ and for any fixed m, the operator $L_{t,1}$ converges to the Laplace-Beltrami operator onto the linear span of its first m eigenfunctions, and the kernel

$$k_{t,n} := \left(H_{\frac{t}{n}}\right)^{n-1} N^{(\alpha)}\left[k_{\frac{t}{n}}\right]$$

converges to the Neumann heat kernel on the manifold as n goes to infinity.

As a matter of fact, the result proved in [47] is more general. For each value of α, the generator converges to a specific operator (see Theorem 2 in [47]). In particular, an interesting fact is that, for $\alpha = \frac{1}{2}$, the process approximates the diffusion of a Fokker-Planck equation depending on the density function Q. This result implies that different normalizations of the same Gaussian kernel may be used to define a generalization of other diffusion processes proposed in differential cortex models. As already mentioned in Section 1.3, the Fokker-Planck equation has been taken into consideration in various works to describe the lateral connectivity of V1 [112, 127].

We model the expansion of the activity starting from the stimulation of one specific profile (i.e. one point p_0 of the feature space $\mathscr{G}$) through an operator analogous to (2.35). First, we adapt the normalization operation proposed in [47] to our setting by taking the integrals w.r.t. the spherical Hausdorff measure associated to the cortical distance. For $\alpha = 1$

and $Q \equiv 1$, we obtain the operator N applied to kernels $\mathscr{K} : \mathscr{G} \times \mathscr{G} \to \mathbb{R}$ as follows:

$$N[\mathscr{K}](p,q) = \frac{\mathscr{K}^{(1)}(p,q)}{\int \mathscr{K}^{(1)}(p,q')d\mu(q')},$$

where

$$\mathscr{K}^{(1)}(p,q) = \frac{\mathscr{K}(p,q)}{\int \mathscr{K}(p,q')d\mu(q') \int \mathscr{K}(p',q)d\mu(p')}.$$

We then define the propagation operator

$$H_t f(p) := \int N[\nu(K_t)](p,q)\, f(q)\, d\mu(q), \tag{2.36}$$

where N is applied to the kernel K_t (we hereafter make explicit the dependence of K on the squared norm t of the filters) after passing it through a sigmoidal activation function

$$\nu(z) = \frac{1}{1+\exp(-z)}.$$

Note that, for $d(p,q) \to +\infty$, the term $\nu(K_t)$ is an exponentially decaying function of $\frac{d_t^2(p,q)}{2}$:

$$\nu(K_t(p,q)) = \frac{\exp\left(-\frac{d_t^2(p,q)}{2}\right)}{\frac{1}{e}+\exp\left(-\frac{d_t^2(p,q)}{2}\right)} \sim e \cdot \exp\left(-\frac{d_t^2(p,q)}{2}\right).$$

We finally provide a description of the propagation of neural activity around a point p_0 by defining

$$K_{t,n}^{p_0} := H_t^{n-1} K_t^{p_0}, \tag{2.37}$$

where $K_t^{p_0}(p) \equiv K_{t,1}^{p_0}(p) := N[\nu(K_t)](p,p_0)$.

By this iterative procedure, the spatial extent of the kernel widens at each step, and this width depends on the size of the RPs, i.e. on the diameter of their RFs (RPs are local objects, and they are always modeled through compactly supported or rapidly decaying functions). There is neurophysiological evidence [7] that the extension of horizontal connections departing from a V1 cell matches the size of its so-called *low-contrast summation field*; this is identified as the area measured by presenting low-contrast bars or gratings of increasing sizes at the RF center, and keeping the size of peak response. The low-contrast summation field has in turn been shown [85, 7] to be on average 2.2-fold greater than the "classical" RF. Therefore, since a relationship between the RF size and the spatial extent of horizontal connections seems to hold from a biological point of view, we believe that the optimal number

of iterations may not depend on the RP size. As for the choice of this number, a first stopping criterion is this average ratio between the size of low-contrast summation fields and classical RFs. Another possible quantitative framework for fitting the kernel to the neurophysiological data was proposed in [59] to fit a Fokker-Planck kernel to the data from [27] and [7]: the kernel was evaluated on a pinwheel map and compared with the measured distribution of the tracer by comparing their densities onto the rectangles of a grid (whose sampling size was chosen to match the pinwheel scale).

Remark 2.16. The kernel K_t can be locally approximated through an exponentially decaying function of the squared distance. By Taylor expansion, we have: $e^{-\frac{d_t^2(p,q)}{2t}} = \left(1 - \frac{d_t^2(p,q)}{2t}\right) + o\left(\frac{d_t^2(p,q)}{t}\right)$, where d_t depends on the squared norm t of the filters – also note that $d_t = t \cdot d_1$, where d_1 is the cortical distance obtained from the same bank of filters, but normalized to have L^2 norm equal to 1. Recall that K_t can be expressed in terms of d_t as follows:

$$K_t(p,q) = t - \frac{d_t^2(p,q)}{2},$$

where t is the squared L^2 norm of the filters. If we fix t, when $d_t(p,q)$ is small we then get

$$e^{-\frac{d_t^2(p,q)}{2}} \approx \left(1 - \frac{d_t^2(p,q)}{2}\right) = K_t(p,q). \tag{2.38}$$

This suggests that, with a much more rough approximation, one can even think of modeling the cortical connectivity as an iteration of (a proper normalization of) K_t itself instead of a Gaussian kernel. In this case, one may consider an activation function of the type $s(z) = \max(z - T, 0)$, which simply puts to zero all values below a certain threshold T.

Finally, note that this idea of repeated convolutions can be applied to model the evolution of the response of the layers of V1 to a visual stimulus I applied to the retina, in the presence of horizontal connections. In general, given a function of the cortical coordinates

$$F_0 : \mathscr{G} \longrightarrow \mathbb{R},$$

the action of the *propagated* kernel onto F can be expressed by

$$H_n[F](p_0) := \int_{\mathscr{G}} K_n^{p_0}(p)\, F_0(p) d\mu(p). \tag{2.39}$$

Now, note that

$$H_n[F](p_0) = \int_{\mathscr{G}} \left(\int_{\mathscr{G}} N[s(K)](p,q)\, F_0(p) d\mu(p) \right) K_{n-1}^{p_0}(q) d\mu(q)$$
$$= \int_{\mathscr{G}} K_{n-1}^{p_0}(q)\, H_1[F](q) d\mu(q) = H_{n-1}[H_1[F]](p_0) = \ldots = \underbrace{H_1 \ldots H_1}_{n}[F](p_0).$$

This means that applying the n-th step kernel to F_0 is equivalent to applying n times the local kernel to F_0. Now, one can take as F_0 the activation computed by lifting the image I to the cortical coordinates. This can be written as follows:

$$I_0(p) := h\left(\int I(x,y) \psi_p(x,y) dxdy \right),$$

where h is an activation function. Therefore, one can compute the "evolved" response $I_n(p_0) = H_n[I](p_0)$ by taking n steps of propagation through the local kernel. We will come back on this in Chapter 3, where an analogous form of propagation will be applied to the activations of the layers of a convolutional neural network.

2.4 Experiments

In this section we show, through some numerical simulations, the geometrical properties of the connectivity kernels obtained by applying our model to some different banks of filters. We first provide some general information about the numerical scheme used throughout the simulations. As a first example, we resume the classical model of Gabor filters presented in Section 2.2.2 and display the association fields generated by the connectivity in this case, as well as in the case of the surface generated by an orientation map. The propagated kernel will be computed both through the mechanism of repeated integrations described in Section 2.3.3, and by suitably approximating the diffusion process introduced in Section 2.1. We then recover curvature selectivity properties of the association fields emerging from endstopped simple cells. We also take into account the spatiotemporal behavior of the RPs by considering a family of three-dimensional Gabor filters including a time parameter. Finally, we show that it is still possible to recover a coherent "bow-tie" pattern starting from a family of RPs obtained through a learning algorithm, whose parameterization carries no a priori geometric information.

2.4.1 Numerical scheme

The first distinction to be made in the treatment of these examples from the numerical point of view is that between the *continuous* case and the *discrete* one. This can refer either to the feature space $\mathscr{G}$ or to the spatial domain of the single filters. For instance, in the case of a Gabor system, both the feature space $\mathscr{G} = \mathbb{R}^2 \times S^1$ and the spatial variable $(u,v) \in \mathbb{R}^2$ need to be sampled. On the contrary, when dealing with a bank of learned filters, the family of indices $\mathscr{G}$ and the filter domains are finite sets. Of course, one could also consider "hybrid" cases, e.g. a bank of filters defined on $\mathbb{R}^2$ but indexed by a discrete set.

For what concerns the computation of the initial kernel, the numerical setup is very basic. The filters are either defined on $\mathbb{R}^2$ or numerically known onto a set of the type $\{(\frac{i}{n},\frac{j}{n})\}_{i,j=-n,\dots,n} \subseteq \mathbb{R}^2$; computing the generating kernel $K(p,q)$ simply involves taking the integral of the function $\psi_p \cdot \overline{\psi_q} \in L^1(\mathbb{R}^2)$, which is either compactly supported or exponentially decaying. Thus, the filter domains can safely be sampled (when needed) through a bounded square grid

$$\{(x_i,y_j)\}_{i,j=0,\dots,N} \subseteq [-W,W] \times [-H,H],$$

for some $N,W,H > 0$, with

$$x_{i+1} - x_i = y_{j+1} - y_j =: \delta > 0 \quad \forall i,j.$$

The kernel $K(p,q)$ is then standardly calculated as

$$\frac{1}{\delta^2} \sum_{i,j} \psi_p(x_i,y_j)\overline{\psi_q(x_i,y_j)}.$$

In the Gabor case, as shown in Section 2.2.1, this integral can even be computed analytically, thus avoiding this approximation.

As for the propagation, first note that, in the case of a finite feature space $\mathscr{G} = \{p_1,\dots,p_M\}$, the Hausdorff measure reduces to the counting measure: integrals over $\mathscr{G}$ are finite unweighted sums over its elements. Although seemingly trivial, this indeed captures the distribution of the features throughout the data thanks to the inhomogeneity of the distance d. Loosely speaking, given a certain set of filters, some features may be "more densely" represented than others w.r.t. d. As a basic example, suppose that the filter ψ_{p_1} is highly

correlated with 8 other filters, i.e. for a fixed ε

$$\left|\left\{i \in \{2,\dots,M\} \,:\, d(p_i,p_1) < \varepsilon\right\}\right| = 8.$$

On the other hand, suppose that ψ_{p_2} has only 2 filters highly correlated with it. As a consequence, $\mu(B^d_\varepsilon(p_1)) = 9$ and $\mu(B^d_\varepsilon(p_2)) = 3$, i.e. the balls of d of the *same* radius ε centered at p_1 and p_2 have different measures. For a non-finite metric space, the spherical Hausdorff measure extends this concept, although in general it is not explicitly computed; however, the distance d in the Gabor case turns out to be estimated by a Riemannian distance (see Proposition 2.3), and the spherical Hausdorff measure on a Riemannian manifold coincides up to a constant factor with the Riemannian volume form [60]. This also makes it possible to approximate the positive self-adjoint operator associate to the diffusion process described in Sections 2.1 and 2.3.1 through a *graph Laplacian* operator, as in [34]. We will go into more detail on the discretization of the Gabor feature space and on the definition of this discrete operator in Section 2.4.2.
A last point to be mentioned is the role of the normalization applied to the kernel when computing the propagation through the iterative procedure described in Section 2.3.3. Namely, for each n we have:

$$0 \le K_n^{p_0}(p) \le \|K_{n-1}^{p_0}\|_\infty \underbrace{\int_{\mathscr{G}} N\left[s\left(K_t\right)\right](p,q)d\mu(q)}_{=1}.$$

Therefore, $\|K_n^{p_0}\|_\infty$ is non-increasing w.r.t. n, which prevents the values of $K_n^{p_0}$ from exploding as $n \to \infty$.

2.4.2 Gabor filters

We now go back to the bank of Gabor filters $\{\psi_{x,y,\theta}\}$ considered in Section 2.2.2. Since the generating kernel in this case is known analytically, no discretization of the filter domain $\mathbb{R}^2$ is needed for its computation. As for the feature space $\mathbb{R}^2 \times S^1$, in the following we consider a sampling of the type

$$\{-\overline{x},\dots,\overline{x}\} \times \{-\overline{y},\dots,\overline{y}\} \times \{-\overline{\theta},\dots,\overline{\theta}\}$$

where $\overline{x} = \overline{y} = 1$ with discretization step .0075 and and $\overline{\theta} = 1.5$ with step .015 for the visualizations of the generating kernel; $\overline{x} = 1.5$ and $\overline{y} = 2$ with step .075, and $\overline{\theta} = 1.5$ with step .15

for the propagation.

The generating kernel K computed around a point $(x_0, y_0, \theta_0) \in \mathscr{G}$ is a three-dimensional function

$$(x, y, \theta) \mapsto K\big((x, y, \theta), (x_0, y_0, \theta_0)\big).$$

Since the feature space for this family is of the form *position* $\times$ *orientation*, it is possible to display a projection of it on the retinal plane. Specifically, by taking the maximum in the

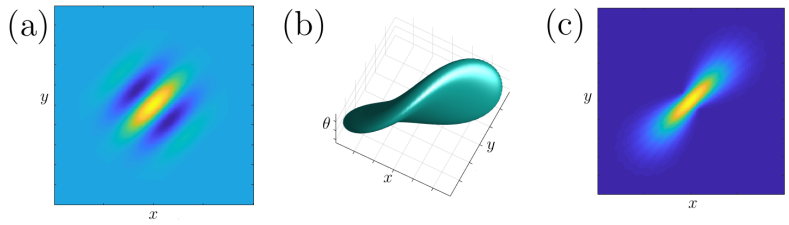


Figure 2.5 The behavior of $(x, y, \theta) \mapsto K\big((x, y, \theta), (0, 0, \frac{\pi}{4})\big)$ *in the case of Gabor filters. The kernel has been truncated as explained above. In particular, the images show: (a) the receptive profile* ψ_{p_0} *corresponding to* $p_0 = (0, 0, \frac{\pi}{4})$*; (b) a level set in* $\mathbb{R}^2 \times S^1$*; (c) the projection on the* (x, y) *plane obtained taking the maximum in* θ*.*

variable θ and projecting onto the (x, y) plane, we obtain a 2D function concentrated around (x_0, y_0), as displayed in Figure 2.5c.

We now display some stages of the iterative process described in Section 2.3.3, for the truncated kernel K. Figure 2.6a displays the real part of the filter $\psi_{(0,0,0)}$, chosen as starting

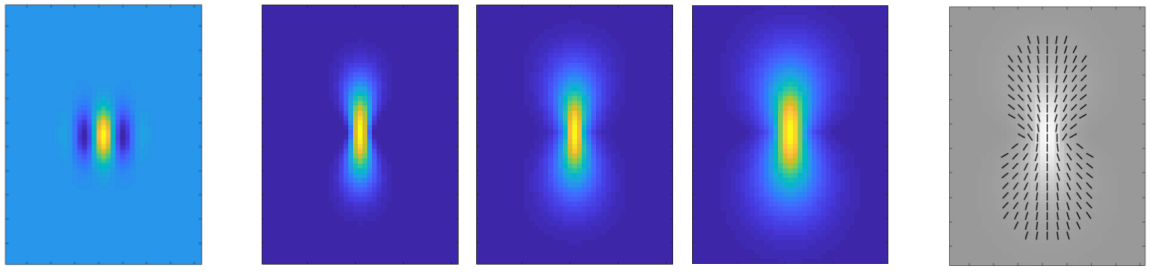


Figure 2.6 Propagation of the neural activity in $\mathbb{R}^2 \times S^1$ *through repeated integrations of the kernel. (a) The starting filter* $\psi_{(0,0,0)}$ *(real part). (b) The kernel* $K_n^{(0,0,0)}(x, y, \theta)$ *for* $n = 1, 2, 3$*, projected down onto the* (x, y) *plane by taking the maximum over* θ*. (c) The corresponding maximizing orientations* $\overline{\theta}(x, y)$*: at every location* (x, y)*, an oriented segment with angle* $\overline{\theta}(x, y)$ *is displayed – only where* $K_3^{(0,0,0)}\big(x, y, \overline{\theta}(x, y)\big)$ *is over a threshold.*

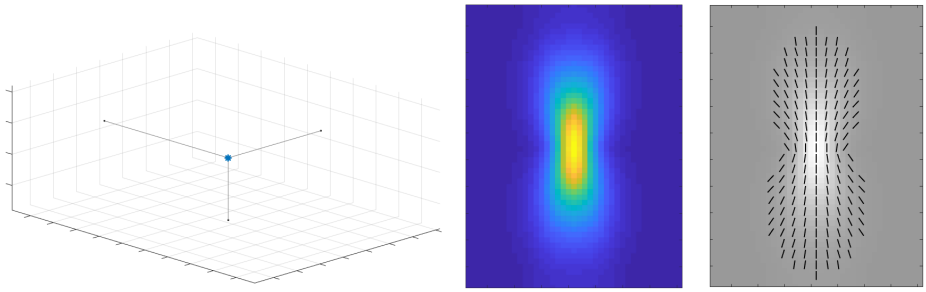
point. In the Gabor case, the connectivity kernel lives in $\mathbb{R}^2 \times S^1$. The functions

$$(x, y, \theta) \mapsto K_n^{(0,0,0)}(x, y, \theta), \quad n = 1, 2, 3, \tag{2.40}$$

obtained through three subsequent steps of the iterative rule (2.37), have been projected onto the (x, y) plane by taking the maximum over the variable θ, as shown in Figure 2.6b. Finally, Figure 2.6c shows the orientation $\overline{\theta}(x, y)$ maximizing the value of $K_3^{(0,0,0)}$, at each location (x, y) where this value exceeds a threshold. Specifically,

$$\overline{\theta}(x, y) := \arg\max_{\theta} K_3^{(0,0,0)}(x, y, \theta).$$

Now, as shown in Section 2.2.2, the cortical distance d obtained from the family of filters



(a) (b) (c)

Figure 2.7 Propagation of the neural activity through the discretized heat equation associated to the graph Laplacian in $\mathbb{R}^2 \times S^1$. (a) The starting point $(0,0,0)$, displayed as a blue asterisk in this 3D space. (b) The updated kernel, projected down onto the (x,y) plane by taking the maximum over θ. (c) The corresponding maximizing orientations $\overline{\theta}(x,y)$, as in Figure 2.6.

(1.2) is locally equivalent to a Riemannian distance on the space $\mathbb{R}^2 \times S^1$. In such a case, it is possible to discretize the Laplace-Beltrami operator by means of a *graph Laplacian operator* associated to the distance. Specifically, given a simple undirected weighted graph Γ with vertices $X = \{p_i\}_i$ equipped with weights $\{\mu_i\}_i$, and edges $E = \{e_{ij}\}_{i,j}$ equipped with weights $\{w_{ij}\}_{i,j}$, one can define for any function f on V the Laplacian operator onto the graph as

$$Lf(p_i) := \frac{1}{\mu_i} \sum_{j\,:\,p_i \sim p_j} w_{ij} \left(f(p_j) - f(p_j) \right), \tag{2.41}$$

where $p_i \sim p_j$ means that there is an edge connecting p_i and p_j. This operator, possibly with slightly different definitions from time to time, is widely used in shape analysis (see e.g. [100, 123]) to construct algorithms that keep trace of the geometry of the data, by means of parameterizations obtained through the eigenfunctions of L. In [34], a graph approximation of a Riemannian manifold M is constructed by taking the set of vertices X to be an ε-net in M

with an associated discrete measure $\tilde{\mu} = \sum_i \mu_i \delta_{p_i}$ which approximates the volume μ of M. In the Gabor case, this allows to consider a simple rectangular grid as set of vertices, provided that the discretization step is sufficiently small. Moreover, this choice yields uniform weights μ_i. The set of edges with relative weights is then defined depending on the distance. Namely, for $\rho \gg \varepsilon$, two vertices $p_i, p_j \in X$ are connected by an edge iff $d_{ij} \equiv d(p_i, p_j) < \rho$, and in this case one defines the edge weight $w_{ij} := \kappa \mu_i \mu_j$, where κ is a normalization constant depending on the dimension of the manifold. Note that the vertices can be chosen to be any ε-net, since the geometry of the manifold is encoded in the definition of the edges, i.e. in the choice of the neighborhood over which the sum (2.41) is taken. We implemented the graph Laplacian associated to this approximating graph, in order to obtain a discretized heat equation on the same sampling of $\mathbb{R}^2 \times S^1$ as before, with initial datum the Dirac delta $f_0 = \delta_{(0,0,0)}$ in this three-dimensional space, see Figure 2.7a. We took 100 iterations of the discretized differential equation with a time step of 0.01. We then projected the updated datum $f(x, y, \theta)$ onto the image plane, again by taking the maximum over θ, and displayed the maximizing orientations as in the preceding case. See Figure 2.7b-c.
The results obtained are compatible with the geometrical properties of V1 lateral connections, and the pattern of the maximizing orientations turns out to be consistent with the perceptual principles of association fields.

Similar results have been obtained in several works concerning second-order statistics on the distribution of edges in natural images. In [12], the statistical measurement of the organization of local edge elements into curves led to the introduction of an oriented filtering operation, expressed by a kernel on $\mathbb{R}^2 \times S^1$ (see Figure 15.3 in [12]); its orientation specificity and the negative lateral lobes recall some characteristics of our connectivity kernel, especially after clipping it at its minimum as explained above (cf. the slices in Figure 2.4c). In [67], association field-like structures comparable to our Figure 2.6c emerge from both a statistical analysis of edge co-occurrence in natural images, and a Bayesian study analyzing the probability of two edge elements belonging to the same extended contour. A similar analysis is carried out in [58], except that here contours are described as *ordered* sequences of tangent elements; in this work a stronger role of the parallelism cue emerges, i.e. there is a marked correlation between parallel edge elements. See also [94] and [134] as further references on the emergence of the collinearity, cocircularity and parallelism cues from statistics on natural images.
These studies are not to be regarded as a separate kind of analysis leading to results consistent with ours. Indeed, it is believed that the cortical processing is influenced by environmental statistics and adapts to best process those signals that are most likely to occur.

Propagation on the pinwheel surface.

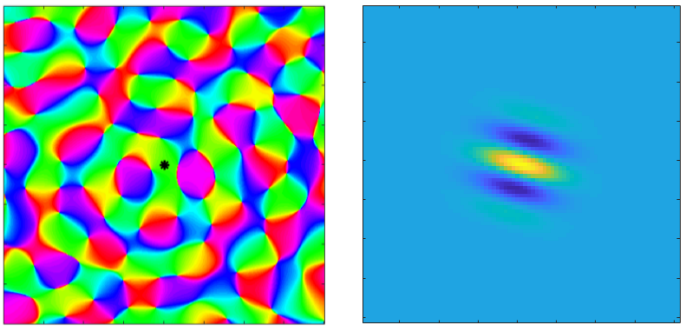


Figure 2.8 (a) The orientation map Θ, generated through superposition of plane waves. The point $(0,0)$ is highlighted in black. (b) The starting filter $\tilde{\psi}_{0,0}$ (real part).

We now consider the sub-family of Gabor filters $\{\psi_{x,y,\Theta(x,y)}\}_{x,y}$ determined by an orientation map Θ and the corresponding metric structure onto $\mathscr{G}_\Theta = \mathbb{R}^2$. We generated an orientation map Θ through superposition of plane waves with random phases, as described in [120], and we chose its central point $(0,0)$ as a starting point (see Figure 2.8a). The corresponding filter $\psi_{(0,0)}$ is displayed in Figure 2.8b. Again, we implemented the propagation of neural activity through iteration of the kernel onto the 2D feature space, and we displayed the updated kernel with color-coded intensity (Figure 2.9a), as well as the orientations $\Theta(x,y)$

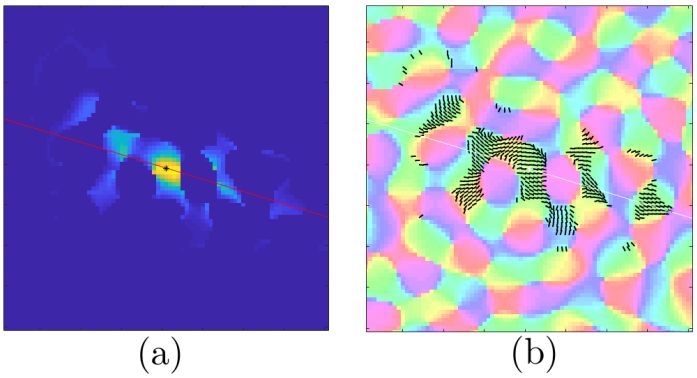


Figure 2.9 Propagation of the neural activity onto $\mathscr{G}_\Theta$ through repeated integrations of the kernel. (a) The propagated kernel around $(0,0)$, obtained after four iterations. A black asterisk shows the starting point $(0,0)$, and the orientation $\Theta(0,0)$ is highlighted by a red line superposed onto the image. (b) At each point (x,y) where the kernel exceeds a threshold, the corresponding orientation $\Theta(x,y)$ is displayed through an oriented segment superposed onto the orientation map.

corresponding to points (x,y) where the kernel exceeds a threshold (Figure 2.9b). As a sampling of the feature space, we took $]-2,2[\times]-2,2[\subseteq \mathbb{R}^2 = \mathscr{G}_\Theta$, discretized with step

0.05 for both x and y. Finally, recall that the cortical distance on $\mathscr{G}_\Theta$ can be seen as the

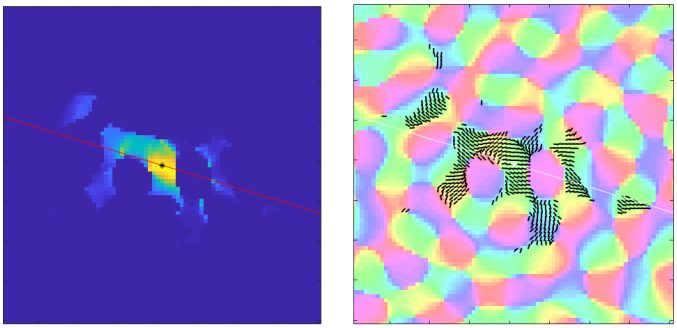


Figure 2.10 Propagation of the neural activity through the discretized heat equation associated to the graph Laplacian on $\mathscr{G}_\Theta$ with initial datum $\delta_{(0,0)}$. (a) The updated 2D kernel around $(0,0)$, with a black asterisk showing the starting point $(0,0)$, and a red line highlighting the orientation $\Theta(0,0)$. (b) At each point (x,y) where the kernel exceeds a threshold, the corresponding orientation $\Theta(x,y)$, as in 2.9.

restriction of the Gabor distance to $\Sigma = \{(x,y,\Theta(x,y))\}_{x,y} \subseteq \mathbb{R}^2 \times S^1$. This is still locally equivalent to a Riemannian metric on $\mathscr{G}_\Theta$. Note that, for $\sigma^2 = A\lambda \ll 1$, this approximates the distance d_Σ of Section 2.2.3. We implemented the graph Laplacian operator associated to this metric on $\mathscr{G}_\Theta$, and the corresponding discretized heat equation with initial datum $\delta_{(0,0)}$. The results for 150 iterations of the discretized equation, with a time step of 0.01, are displayed in Figure 2.10.

Note that in this case we do not need to project the connectivity kernels onto the image plane to visualize them, since the whole propagation already lives in a bidimensional space.

Again, the computed kernel spreads along the axis of the orientation $\Theta(0,0)$; moreover, it propagates in a *patchy* way, with peaks in the regions of the map whose orientation values are close to $\Theta(0,0)$. This behavior has been observed experimentally by tracking the spreading of neural activity through biocytin injections, and by comparing it with the underlying orientation preference map [27].

2.4.3 Endstopped simple cells

We will now focus on the connectivity pattern emerging from a particular class of neurons, namely *endstopped simple* (ES) cells, showing *simple* RPs equipped with end zones along the preferred orientation (see Section 1.1.2). Such profiles can be modeled by linear combination of two similarly positioned and oriented simple RPs of different sizes, as proposed in [51]. Specifically, one such RP ψ can be written as the (weighted) sum of the positive contribution

of a smaller simple RP ψ^S and the negative contribution of a larger simple RP ψ^L, as follows:

$$\psi := c_S\,\psi^S - c_L\,\psi^L, \tag{2.42}$$

where $c_S > c_L > 0$. Figure 2.11 shows an example of even and odd RPs generated as in

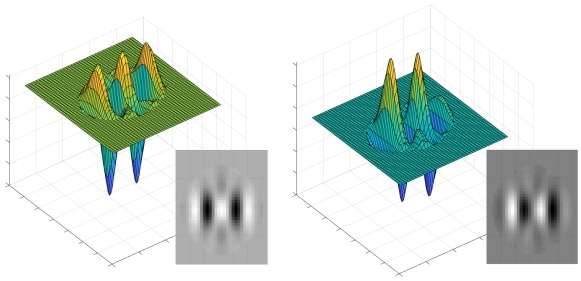


Figure 2.11 ES profiles obtained as in Eq. (2.42).

(2.42), where the simple RPs ψ^S and ψ^L are modeled by Gabor filters. To be more precise, the model in [51] proposes to represent the response R of ψ to some contrast pattern as

$$R = h\big(c_S\,h(R_S) - c_L\,h(R_L)\big), \tag{2.43}$$

where R_S and R_L are the responses of ψ^S and ψ^L to such pattern, and h is the rectifying function $h(z) = \max(0,z)$. This increases the stimulus specificity compared to directly computing the response from (2.42). Nonetheless, the two models coincide when the responses of both ψ^S and ψ^L to a visual stimulus are nonnegative, i.e. for those stimuli in the "preferred range" of orientation and curvature of the ES cell.

We computed the kernels associated to some families of ES profiles of different lengths and displayed the resulting association fields, obtained as before by selecting the orientation that maximizes the intensity at each location. This experiment shows that our model allows to recover the link between the length of ES cells and curvature [82, 51]. The left column of Figure 2.12 shows the connectivity patterns obtained for a set of non-endstopped simple cells (top image), corresponding to zero curvature, and for three families of ES cells of decreasing lengths, corresponding to increasing absolute values of curvature. In each of the cases displayed, we considered 21 RPs with orientation values equally spaced in $[-\pi,\pi]$. We then shifted them spatially to generate the whole filter bank. Our results are compared to the curvature-based connection fields obtained in [20] through differential tools, displayed in the right column of Figure 2.12. Since the sign of the curvature is taken into consideration in [20], each of our association fields corresponds to a couple of theirs (with curvatures κ and $-\kappa$).

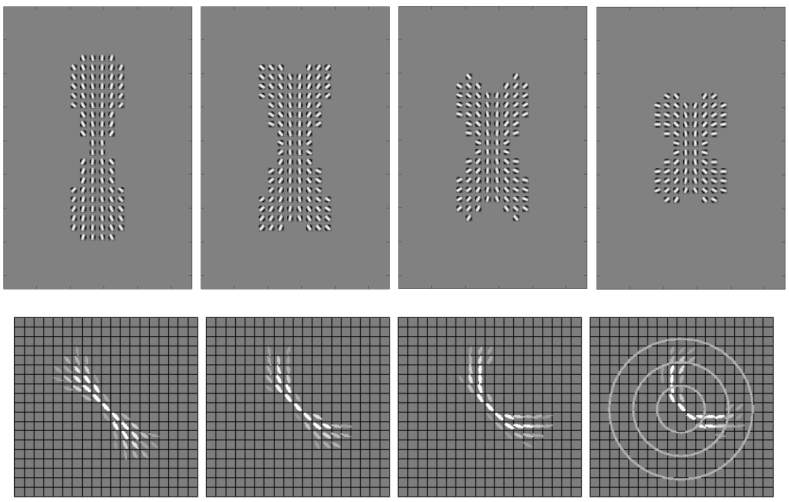


Figure 2.12 Top row: association fields emerging from the connectivity kernels associated to a simple cell (top) and to three ES profiles of varying length. Bottom row: the connection fields of curves obtained in [20] relative to curvature $\kappa = 0$ (top) and three increasing positive values of κ respectively.

2.4.4 Spatiotemporal Gabor filters

Let us consider a family of filters still of Gabor type, but taking into account the movement of the stimulus. Cocci et al. [46] fitted the RPs of a set of V1 neurons showing velocity-selective behaviours with a three dimensional Gabor model. That is we have, in addition to the two spatial dimensions, a third temporal dimension in the domain of these filters, which in fact form a subset of $L^2(\mathbb{R}^3)$. A convenient visualization of such filters is as a temporal

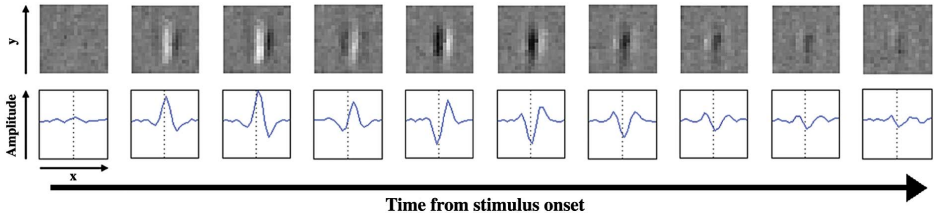


Figure 2.13 The time course of the recording of a simple cell's RP. Source: [46].

sequence of spatial maps (see Figure 2.13, taken from [46]). In [17], the authors also develop a differential model of functional architecture based on such 3D Gabor family.
Now, in order to mantain the notations as similar as possible to the one that we used for 2D Gabor filters, we shall write the general *inseparable* element of the family of filters as

$$\psi_{x,y,\theta,t,\alpha}(u,v,s) = \exp\left(-2\pi i\left(\frac{X}{\lambda}+\alpha(s-t)\right)\right)\cdot\exp\left(-\frac{X^2+Y^2}{2\sigma^2}-\frac{(s-t)^2}{2\beta^2}\right),$$

where

$$\begin{cases} X = (u-x)\cos\theta + (v-y)\sin\theta \\ Y = -(u-x)\sin\theta + (v-y)\cos\theta. \end{cases}$$

The set of filters is indexed by the position $(x,y) \in \mathbb{R}^2$, the orientation $\theta \in S^1$, the time $t \in \mathbb{R}^+$ at which the response of the filter is maximum, and the velocity $\alpha \in \mathbb{R}$. Parameters λ and β are fixed.

Let us compute the generating kernel K in this setting. The feature space in this case is $\mathbb{R}^2 \times S^1 \times \mathbb{R} \times \mathbb{R}$. We compute $K\big((x,y,\theta,t,\alpha),(x_0,y_0,\theta_0,t_0,\alpha_0)\big)$ with $t = t_0$: this means considering two cells whose activation peaks at the same time. In other words, we fix t and we take $\mathscr{G} = \mathbb{R}^2 \times S^1 \times \mathbb{R}$ as our feature space. We restrict to this case in order to be able to better interpret the results in terms of the orientation and velocity parameters. Denote $p = (x,y,\theta,\alpha)$ and $p_0 = (x_0,y_0,t_0,\alpha_0)$. We obtain the following expression.

$$K(p,p_0) = K^{\text{spatial}}\big((x,y,\theta),(x_0,y_0,\theta_0)\big) \cdot \beta^2\sqrt{\pi}\,\exp\left(-\frac{\beta^2(\alpha-\alpha_0)^2}{4}\right),$$

where K^{spatial} denotes the generator obtained in the time-independent case discussed above. Through repeated integrations against K, we obtain the connectivity kernels $K_n^{(x_0,y_0,\theta_0,\alpha_0)}$ associated to this family of filters. Since we are considering a 4-dimensional feature space,

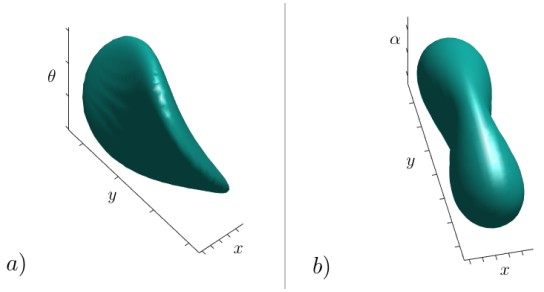


Figure 2.14 Visualizations of the spatiotemporal generating kernel computed around the point $(x_0,y_0,\theta_0,\alpha_0) = (0,0,0,0)$. *a) A level set of its projection onto the* (x,y,θ) *space, obtained by taking the maximum in* α. *b) A level set of its projection onto the* (x,y,α) *space, obtained by taking the maximum in* θ.

we visualize the generating kernel by projecting it onto the 3-dimensional spaces $\mathbb{R}^2 \times S^1$ (position and orientation) and $\mathbb{R}^2 \times \mathbb{R}$ (position and velocity). Figure 2.14 displays the kernel around $p_0 = (0,0,0,0)$ projected (a) onto $\mathbb{R}^2 \times S^1$ by taking the maximum in the variable α, and (b) onto $\mathbb{R}^2 \times \mathbb{R}$ by taking the maximum in the variable θ.

Note that the filters $\psi_{x,y,\theta,t,\alpha}$ represent cells that respond maximally to only one direction of movement (depending on the sign of α): such profiles are called *inseparable*. However, there exist also cells which are equally sensitive to both directions, whose profiles are called *separable* (since they can be obtained by the product of two *real* functions of space and time respectively), as well as cells sensitive to both directions but to a different extent. The family of all these profiles can be obtained through weighted sums of inseparable profiles [46]:

$$\psi^C_{x,y,\theta,t,\alpha}(u,v,s) = C\psi_{x,y,\theta,t,\alpha}(u,v,s) + (1-C)\psi_{x,y,\theta,t,-\alpha}(u,v,s), \tag{2.44}$$

where $C \in [0,1]$ is the *separability index*, weighing the contribution of sensitivity to the two opposite directions of movement, expressed by the velocity parameters α and $-\alpha$. Note that, if we introduce C as a parameter, then α can be taken to be nonnegative as an index.
The generator $\tilde{K}$ for the complete family

$$\{\psi^C_{x,y,\theta,t,\alpha}\}_{x,y,\theta,t,\alpha,C}$$

is easily obtained from the generator K for the inseparable family above. Denote

$$\begin{aligned} q &= (x,y,\theta,t,\alpha,C), \\ q_0 &= (x_0,y_0,\theta_0,t_0,\alpha_0,C_0). \end{aligned}$$

We have:

$$\begin{aligned} \tilde{K}(q,q_0) = {} & CC_0K(p^+,p_0^+) + C(1-C_0)K(p^+,p_0^-) \\ & + (1-C)C_0K(p^-,p_0^+) + (1-C)(1-C_0)K(p^-,p_0^-), \end{aligned}$$

where $p^+ = (x,y,\theta,t,\alpha)$, $p^- = (x,y,\theta,t,-\alpha)$ and p_0^+, p_0^- are defined similarly.

2.4.5 A family of learned filters

We now test our model in the case of a family of numerically-known filters indexed by a discrete subset of $\mathbb{R}^2 \times \mathscr{F}$, where the *fiber* $\mathscr{F}$ is just a set of indices with no a priori geometric structures. The aim of this example is to show that the metric structure defined on this feature space by the kernel K still generates "bow-tie" patterns onto the retinal plane.
Specifically, we chose a bank of filters obtained through an unsupervised learning algorithm which maximizes sparseness; this procedure was first proposed by Olshausen and Field

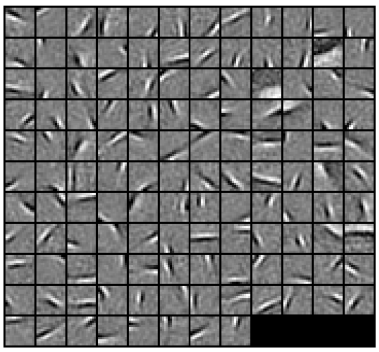


Figure 2.15 A bank of 128 filters obtained by training a set of basis functions on natural images (see [117]). The algorithm used is described in [98].

in 1996 [117] in order to find efficient linear codes for natural scenes, as an attempt to understand the response properties of visual neurons in terms of the statistical structure of natural images. Their algorithm generates a family of localized, oriented, bandpass RPs, as the 128 shown in Figure 2.15 (which were generated using a later version of the algorithm, provided by Lee et al. in 2007 [98]).

We then "centered" the support of each filter ψ by identifying the spatial location around which ψ is concentrated and cropping its domain (16×16 pixels originally) symmetrically around this point to obtain a 11×11 pixel support. The central location was simply chosen to be the point where the function ψ reaches its maximum. To manage the cases in which the maximum point was near the border, we added a 5-pixel padding of zeros around the initial filters before cropping.

At this point, we have a set $\{\psi_f\}_{f=1,\dots,128}$ of functions centered at zero: by shifting them spatially, we obtain a family of filters $\psi_{x,y,f}$ centered at (x,y), where $x,y \in \{-W,\dots,W\}$. Therefore, the feature space in this case can be written as

$$\mathscr{G} = \{-W,\dots,W\}^2 \times \mathscr{F},$$

where $\mathscr{F} = \{1,\dots,128\}$. The width W depends on the size of the images one considers onto the retinal plane. For example, in [117], 512×512-pixel images where considered to generate the bank of filters: in this case, one may take $W = 256$.

We are now able to compute K. Note that, since the filters have an 11×11 support, the support of the generating kernel K projected onto the (x,y)-plane will be of size 21×21 and the supports of the kernels obtained by repeated integrations will widen progressively. Figure 2.16 displays the projection onto the retinal plane of the kernel around a filter, for four different filters of the family. As in the Gabor example, the first projection (central

column) shows at each (x,y) the maximum value (color-coded) of the kernel over all $f \in \mathscr{F}$; the second projection (right column) displays, at the points (x,y) where this value is above a threshold, a tiny version of the filter $\psi_{x,y,f}$ where the maximum is achieved. Even in

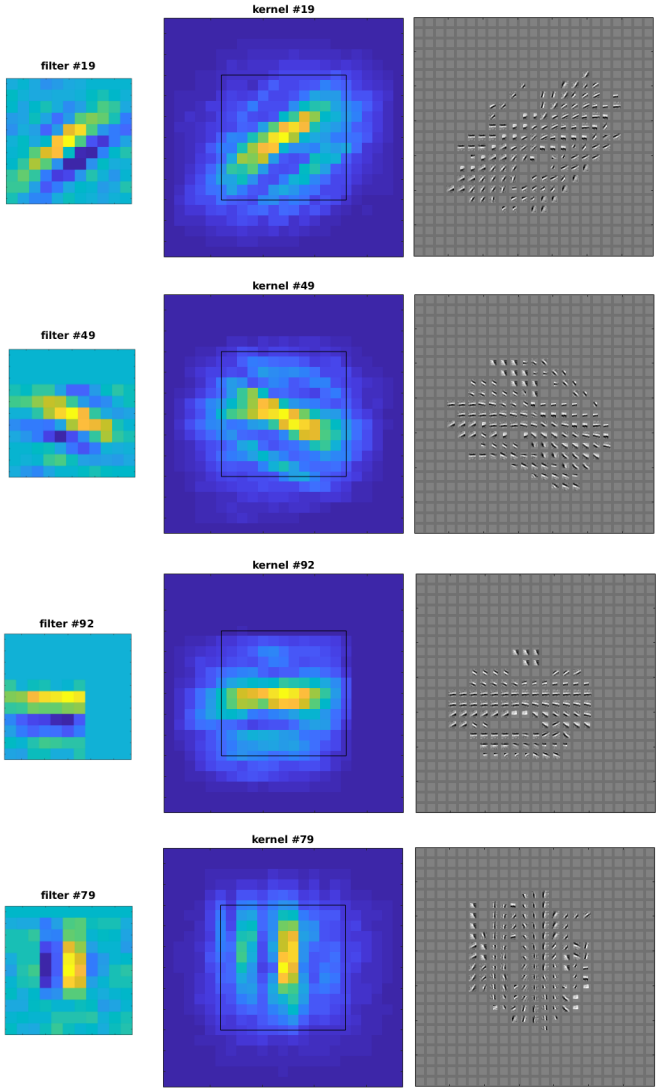


Figure 2.16 Visualization of the kernel K computed around four filters $\psi_{0,0,f_0}$ ($f_0 = 19,49,92,79$). In each of the four cases the image displays, from left to right: the original filter (already cropped around its center); the projection of the kernel on the (x,y)-plane obtained by taking the maximum in the variable f; the same projection, where instead of the intensity value we displayed at each pixel the filter $\psi_{x,y,f}$ maximizing $K\big((x,y,f),(0,0,f_0)\big)$.

this non-structured case, a "bow-tie" pattern along the preferred axis of the starting filter develops, with the orientations of the maximizing filters organizing across space in a way consistent with the association fields of [61]. However, in this example the "side lobes"

emerging from the interaction of parallel RPs were not discarded. A clipping similar to the one performed for Gabor filters may be reproduced e.g. by adopting techniques of connected component extraction; nonetheless, since the pattern obtained turns out to be consistent with the "ladder" effect described in [152], truncating the kernel might mean ignoring some relevant information.

Note that in this case we only displayed bidimensional representations of the connectivity kernel. This is still possible since we constructed a parameterization of the family of filters in order to have a feature space of the form *position* $\times$ *features*, so it makes sense to project the results on the (x,y)-plane. However, a visualization of the kernels in the space $\mathscr{G} = \{-W,\dots,W\}^2 \times \mathscr{F}$ would no longer be meaningful, since the set of indices $\mathscr{F} = \{1,\dots,128\}$ is not ordered with respect to the distance.

Chapter 3

A metric model for lateral connections in CNNs

Convolutional Neural Networks (CNNs) are a powerful algorithmic framework that provides outstanding performances on image classification tasks. However, there is still little insight into how the learning process of these algorithms develops and how exactly image information is coded in CNNs, and notably *how these mechanisms are related to human object processing*. Indeed, although CNN models were initially inspired [63, 96] by Hubel and Wiesel's hierarchical model of the visual system [82], they display critical discrepancies w.r.t. biological vision in both structure and feature analysis.

In [13] the authors show that, unlike in human vision, global shapes have surprisingly little impact on the classification output of the net: on the contrary, CNNs turn out to learn mostly from local features. As such, CNN architectures are very unstable to small local perturbations, even when the global structure of the image is preserved and its content is still easily recognizable by a human observer. Along the same lines, it has been recently shown [29] that a very good classification accuracy on the ImageNet benchmark dataset can be reached through a model that only relies on the occurrences of local features, with no information on their spatial location in the image.

Besides, although the overall convolutional architecture has much in common with the process of feature extraction carried out in the visual pathways, its structure implements a *purely feedforward* mechanism. On the other hand, the human visual system is well known to rely on both *lateral* (intra-layer) and *feedback* (top-down) recurrent connections for processes that are critical for object recognition, such as contour integration or figure-ground segregation (see Chapter 1). In recent years, several models have been proposed in which CNN architectures are enriched with some recurrent mechanism inspired by biological visual systems. In [139], pre-trained feedforward models were augmented with a Hopfield-like

recurrent mechanism acting on the activation of the last layer, to improve their performance in pattern completion: partially visible objects converge to fixed attractor points dictated by the original whole objects. In [101] a "Recurrent CNN" architecture is introduced, where lateral connections of convolutional type are inserted in a regular feedforward CNN. We shall briefly outline this architecture, henceforth referred to as RecCNN, in Section 3.1.3. A systematic analysis of the effect of adding lateral and/or top-down connections has been carried out in [135], where the resulting architectures are tested on a task of classification of cluttered digits. In RecCNNs, lateral connections are learned, and no geometrical prior (apart from the convolutional structure) is inserted. As such, these connections are determined by additional parameters that are completely independent of the feedforward architecture.

In the following, we propose to modify the classical CNN architecture by inserting a biologically plausible geometric prior on the structure of lateral connections, in terms of a measure of *correlation* between neurons inspired by our connectivity model described in Chapter 2. The main point that we wish to make is that this information allows the networks to *spontaneously* implement perceptual mechanisms of global shape analysis and completion. Therefore, we shall examine the ability of the models to *generalize* an image classification task to data corrupted by a variery of different perturbations: these include occlusions (as in [139]), local contour disruption (as in [13]) and adversarial attacks via Fast Gradient Sign Method (FGSM) [74]. We stress that the data perturbations are only inserted in the testing phase – that is, the models are not trained to classify corrupted images.
As a preliminary experiment, we examine the effect of an *implicit* constraint given by the introduction of a suitable regularization term in the loss function. We then improve this method by directly modifying the architecture of a base 2-layer CNN through lateral connections similar to the ones in RecCNNs, but defined by a structured kernel encoding specific geometric information. This new architecture will be referred to as KerCNN. We will compare the performance of these new models to the one of the base CNN, for the generalization task mentioned above.

The chapter is organized as follows: in Section 3.1, we provide the necessary background about some deep learning architectures; in Section 3.2 we outline our new models and define the task; finally, Section 3.3 is devoted to the results. The analysis presented in Sections 3.2 and 3.3 roughly corresponds to the content of [108].

3.1 Feedforward and recurrent CNN architectures

In this section we give some general background on deep neural networks, with a focus on the specific architectures mentioned before, namely feedforward and recurrent CNNs.

3.1.1 Deep Neural Networks

The term *deep learning* indicates a class of machine learning algorithms, inspired by information processing in biological nervous systems, designed to recognize patterns and learn data representations. Its diverse applications include computer vision, speech recognition, social network filtering, machine translation, drug design, medical image analysis. The main object at the basis of this framework is the *artificial neural network*, which may be defined as a directed graph whose vertices are called nodes (or artificial neurons, or simply neurons) and whose edges represent the connections between these nodes. The *inputs* of a neuron are all the nodes from which it receives incoming connections; its *output* is typically defined as some nonlinear function of a weighted sum of these inputs. The nodes of such a network can be aggregated into groups, called *layers*, which are connected with each other in a hierarchical way, that is they are ordered; the presence of one or more *hidden* layers interposed between the first (input) and last (output) layer characterizes a sub-family of algorithms called *deep neural networks* (DNNs). The most basic DNN architecture is a purely feedforward one, where the neurons of each layers send outgoing connections only to neurons of the next layer. The network is said to be *fully connected* if all the nodes of a layer are connected to all the nodes of the next one.

A feedforward DNN implements an operator F which is determined through a *minimization process*, to approximate a given operator

$$Z : \mathscr{H}_0 \to \mathscr{H}_L. \tag{3.1}$$

The approximating functional F is obtained as the composition $F_0 \circ \ldots \circ F_L$ of mappings

$$F_l : \mathscr{H}_l \to \mathscr{H}_{l+1}$$

acting between suitable functional spaces and representing the network's layers. We hereafter outline this process through the following steps.

- We first describe the approximating operator F, determining the so-called *architecture* of the feedforward DNN.

- We then introduce the *loss function*, i.e. the function to be minimized, essentially expressing a measure of dissimilarity between the approximating operator F and the "true" operator Z.
- Finally, we focus on the optimization process, which is typically implemented via a gradient descent method.

Feedforward DNNs

The unknown operator $F = F_0 \circ \ldots \circ F_L$ takes as input a function $I \equiv h_0 : \mathscr{G}_0 \to \mathbb{R}$ (first layer) and produces an output $h_L = Fh_0 : \mathscr{G}_L \to \mathbb{R}$ (last layer). For $l \in \{0, \ldots, L\}$, the mapping $F_l : \mathscr{H}_l \to \mathscr{H}_{l+1}$ encodes the operations performed between the l-th layer and the $(l+1)$-th layer. This yields an alternance of linear and nonlinear operations: F_l applies to its input $h_l \in \mathscr{H}_l$ a linear operation followed by a nonlinear one. The *activation* $h_{l+1} : \mathscr{G}_{l+1} \to \mathbb{R}$ of the $(l+1)$-th layer is obtained as a function of the preceding activation $h_l : \mathscr{G}_l \to \mathbb{R}$ as follows:

$$h_{l+1} := F_l(h_l) = s_{l+1}(A_l h_l + b_l),$$

where:

- $s_{l+1} : \mathscr{H}_{l+1} \to \mathscr{H}_{l+1}$ is a nonlinear *activation function*;
- $A_l : \mathscr{H}_l \to \mathscr{H}_{l+1}$ is a linear operator;
- $b_l \in \mathscr{H}_{l+1}$ is a further additive element, often referred to as *bias* term.

In most cases, the activation functions s_l are obtained by pointwise applying a real function $s_l : \mathbb{R} \to \mathbb{R}$, which we still denote by the same name – i.e. with a slight abuse of notation we write $s_l(f) := s_l \circ f$. A typical choice for it in recent literature is the so-called *Rectified Linear Unit* (ReLU), defined as $s_l(z) = \max(0, z)$ [114, 93]. Another common example is the sigmoidal activation function

$$s_l(z) = \frac{1}{1 + e^{-z}}. \tag{3.2}$$

In practical implementations, the domains $\mathscr{G}_l$ are discrete sets $\{1, \ldots, N_l\}$ and we generally denote $h_l = \{h_l(i)\}_{i \in \mathscr{G}_l}$. In this case, the linear operators A_l at each layer can be represented as $N_{l+1} \times N_l$ matrices $\{w_l(j,i)\}_{j \in \mathscr{G}_{l+1}, i \in \mathscr{G}_l}$, and the bias terms are vectors $b_l = \{b_l(j)\}_{j \in \mathscr{G}_{l+1}}$. In these terms, the network is fully connected when at every layer the entries of W_l are all independent of each other and $w_l(j,i) \neq 0$ for all $i \in \mathscr{G}_l, j \in \mathscr{G}_{l+1}$. According to this formulation, for a fixed $j \in \mathscr{G}_{l+1}$, $\{w_l(j,i)\}_{i \in \mathscr{G}_l}$ contains the *weights* of the incoming edges of $h_{l+1}(j)$ in the graph structure. See Figure 3.1 for a schematic depiction.

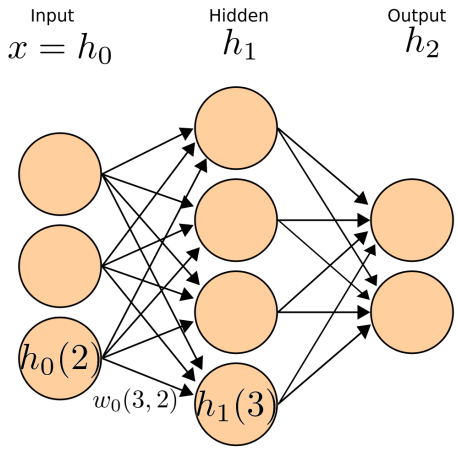


Figure 3.1 A feedforward, fully connected DNN with one hidden layer. As an example, we displayed the neuron $h_0(2) = I(2)$ of the input layer and the neuron $h_1(3)$ of the first hidden layer, as well as the weight $w_0(3,2)$ connecting them.

The loss function

Recall that the functionals F_l are to be determined in order for F to best approximate the true operator Z: this is done by minimizing a *loss function* $\mathscr{L}$, usually composed by a "fiducial term" P, and a "regularization term" R. The latter is generally introduced to enforce some restriction on the complexity of the function space in which the approximating function F is sought. For instance, smoothness conditions or bounds on the vector space norm may be imposed. In fact, the most common regularization term in deep learning is defined as

$$R(F) = \lambda \sum_{l=0}^{L} \sum_{\substack{j \in \mathscr{G}_{l+1} \\ i \in \mathscr{G}_l}} |w_l(j,i)|^2. \tag{3.3}$$

That is, the squared 2-norms of the matrices $\{w_l(j,i)\}_{i,j}$ for each l are penalized. The term is weighted by a factor λ, called the *regularization hyperparameter*. This method, usually referred to as L^2 regularization, enforces small values of the coefficients. Alternatively, one can use an L^1 term (or *lasso* [140]) penalizing the absolute values of the weights, yielding a sparse representation – i.e. one involving few non-zero coefficients. The role of the regularization term R is linked to the concept of generalization in learning problems: we will come back on this later.

On the other hand, the term P quantifies how much F differs from Z. In a so-called *supervised learning* framework, Z is known on a given subset $\mathscr{X} \subseteq \mathscr{H}_0$ and the fiducial term takes the form

$$P(F(I), Z(I)), \tag{3.4}$$

where $I \in \mathscr{X}$. Then the whole loss function is obtained in this case as

$$\mathscr{L}(F,Z,\mathscr{X}) = \frac{1}{|\mathscr{X}|}\sum_{I\in\mathscr{X}} P(F(I),Z(I)) + R(F). \tag{3.5}$$

As for the form of P, the main distinction to be made is between *regression* and *classification* tasks.

1. The cases in which the outputs $Z(I)$ express a *continuous* quantity (e.g. the height of a population) are referred to as regression tasks. In these cases, a typical choice for P is the classical mean squared error (MSE):

$$P(F(I),Z(I)) := \|F(I)-Z(I)\|_2^2.$$

2. On the other hand, we talk of classification when we aim at separating the input data into M categorical classes identified by *labels* $\{0,\dots,M-1\}$ (e.g. classify images of handwritten digits with the correct number in $\{0,\dots,9\}$). In this case, the problem is generally formulated in probabilistic terms by choosing an output space of probability vectors:

$$\mathscr{H}_{L+1} = \left\{ p \in [0,1]^M \,:\, \sum_{n=0}^{M-1} p_n = 1 \right\}. \tag{3.6}$$

In order for the values of F to be contained in $\mathscr{H}_{L+1}$, the last activation function s_L is taken to be the *softmax* function

$$s_{L+1} : \mathbb{R}^M \to \mathbb{R}^M, \quad s_L(v)_n = \frac{e^{v_n}}{\sum_{k=0}^{M-1} e^{v_k}} \quad \forall\, n = 0,\dots,M-1. \tag{3.7}$$

The softmax function produces a vector whose entries are real numbers between 0 and 1 that sum to 1: this can be interpreted as a vector of probabilities. Note that this provides an example of activation function which is not obtained by pointwise applying a real function.
If an input $I \in \mathscr{X}$ belongs to the $\overline{n}$-th class, the exact function Z maps it to the vector

$$e^n = (e_0,\dots,e_{M-1}), \quad e^n_n = \begin{cases} 1 & \text{if } n = \overline{n} \\ 0 & \text{otherwise,} \end{cases} \tag{3.8}$$

which assigns probability 1 to the n-th class, and 0 to all the other classes. The most common loss function used for multiclass classification is given by the *cross entropy* between the output $F(I)$ and the target vector $Z(I)$ containing the "true" probabilities

associated to the input I:

$$P(F(I),Z(I)) = -\sum_{n=1}^{M} Z(I)_n \log(F(I)_n) = -\log(F(I)_{n(I)}), \tag{3.9}$$

where $n(I)$ is the correct class for I. The last equality holds since $Z(I)_{n(I)} = 1$ and $Z(I)_n = 0$ for each $n \neq n(I)$.

Optimization process

The supervised learning process amounts to determining the linear functionals A_l and the bias terms b_l through minimization of the loss function $\mathscr{L}$. Note that, in a discrete setting, this consists in determining all the entries of the matrices $\{w_l(j,i)\}_{i,j}$ and of the bias vectors $\{b_l(j)\}_j$, which are referred to as the *parameters* of the network.
The typical minimization scheme employed to optimize a DNN's parameters is the *gradient descent* (or *steepest descent*) method. Gradient descent is a first-order iterative algorithm based on the construction of a minimizing sequence for a functional J defined on a Hilbert space. The main idea is to consider a flow in the opposite direction of the gradient of the functional, which is orthogonal to the level lines of J: this means considering a Cauchy problem

$$\begin{cases} \partial_t u(\xi,t) = -\nabla J(u(\xi,t)) \\ u(\xi,0) = u_0. \end{cases} \tag{3.10}$$

By varying the initial point u_0, the solutions of (3.10) form the *steepest descent flow* of J. By discretizing these differential equations, one obtains sequences $(u^{[t]})_{t\in\mathbb{N}} = \big(u(\cdot,t)\big)_{t\in\mathbb{N}}$ along which the value of J decreases. These also converge to minima of J provided that a Palais-Smale compactness condition on the level sets of J is verified. We refer to [3] for a complete discussion on this subject.
An advantage of this method is computational efficiency, due to its formulation through a discretized Cauchy problem, yielding a very simple iteration step

$$u^{[t+1]} = u^{[t]} - \varepsilon\, \nabla J(u^{[t]}).$$

In the present setting, we have $u = (w,b) = \big(\{w_l(j,i)\}_{i,j}, \{b_l(j)\}_j\big)$ and J is given by

$$J(w,b) := \mathscr{L}\big(F^{(w,b)}, Z, \mathscr{X}\big), \tag{3.11}$$

where the dependence of F on (w,b) has been made explicit in the right-hand side for clarity.

Backpropagation

According to the steepest descent method described above, the network parameters are updated through

$$w_l(i,j)^{[t+1]} = w_l(i,j)^{[t]} - \varepsilon \, \frac{\partial J(w,b)}{\partial w_l(i,j)^{[t]}}, \tag{3.12}$$

$$b_l(i)^{[t+1]} = b_l(i)^{[t]} - \varepsilon \, \frac{\partial J(w,b)}{\partial b_l(i)^{[t]}}. \tag{3.13}$$

Each optimization step t is commonly referred to as an *epoch*. The *learning rate* ε is often not kept fixed during the whole learning process: a typical scheme consists in starting with a higher value (yielding larger changes at every iteration) and gradully decreasing it.

The gradients $\nabla_w J = \left(\frac{\partial J}{\partial w_l(i,j)}\right)_{l,i,j}$ and $\nabla_b J = \left(\frac{\partial J}{\partial b_l(i)}\right)_{l,i}$ can be analytically computed through a routine commonly referred to as *backpropagation* [146]. This name is due to the fact that the partial derivatives of J w.r.t. the parameters of the internal layers are computed through an iterative formula starting from the *last* layer, essentially obtained by iteratively applying the chain rule. We denote

$$J_I(w,b) := P\big(F^{(w,b)}(I), Z(I)\big),$$

so that

$$J(w,b) = \frac{1}{|\mathscr{X}|} \sum_{I \in \mathscr{X}} J_I(w,b) + R\big(F^{(w,b)}\big).$$

The backpropagation algorithm is actually only needed for the fiducial part of the loss function, which is affected by the chain of operations performed on the input. We hereby show how to compute the derivative of the fiducial terms $J_I(w,b) = P\big(F^{(w,b)}(I), Z(I)\big)$ w.r.t. one of the linear weights $w_l(i,j)$. A similar, simpler argument can be applied for the bias weights $b_l(i)$.

Denote for convenience $\hat{h}_l(i) = \sum_j w_l(i,j) h_{l-1}(j) + b_l(i)$. Then:

$$h_l(i) = s_l(\hat{h}_l)_i, \tag{3.14}$$

$$\frac{\partial \hat{h}_l(i)}{\partial w_l(i,j)} = h_{l-1}(j). \tag{3.15}$$

By (3.15), we have:

$$\frac{\partial J_I(w,b)}{\partial w_l(i,j)} = \frac{\partial J_I}{\partial \hat{h}_l(i)} \frac{\partial \hat{h}_l(i)}{\partial w_l(i,j)} = \frac{\partial J_I}{\partial \hat{h}_l(i)} h_{l-1}(j) =: \delta_i^l \, h_{l-1}(j).$$

Therefore, computing the whole derivative amounts to determining the quantity $\delta_i^l = \frac{\partial J_I}{\partial \hat{h}_l(i)}$. We distinguish the following two cases.

- $l = L+1$. In this case, $w_{L+1}(i,j)$ is a parameter of the *last* layer before applying the loss function:

  $$I \to \ldots \to h_L(j) \xrightarrow{w_{L+1}(i,j)} h_{L+1}(i) \to P(h_{L+1}, Z(I)).$$

  Then δ_i^{L+1} is easily computed as follows.

  $$\delta_i^{L+1} = \frac{\partial J_I}{\partial h_{L+1}(i)} \frac{\partial h_{L+1}(i)}{\partial \hat{h}_{L+1}(i)} = \partial_i P\big(h_{L+1}\big) \cdot \partial_i \big(s_{L+1}(\hat{h}_{L+1})\big)_i.$$

  Here, $\partial_i P$ denotes the derivative of $P(F,Z) = P\big((f_1,\ldots,f_N),(z_1,\ldots,z_N)\big)$ w.r.t. f_i, and the second factor contains the derivative of the i-th component of $s_{L+1}(\cdot)$ w.r.t. its i-th argument, by Eq. (3.14). For instance, if s_{L+1} is the softmax function (3.7),

  $$\partial_i \big(s_{L+1}(\hat{h}_{L+1})\big)_i = \big(s_{L+1}(\hat{h}_{L+1}(i))\big)_i - \big(s_{L+1}(\hat{h}_{L+1}(i))\big)_i^2.$$

- $l < L+1$. That is, $w_l(i,j)$ is a parameter of an *internal* layer:

  $$I \to \ldots \to h_{l-1}(j) \xrightarrow{w_l(i,j)} h_l(i) \xrightarrow{w_{l+1}(m,i)} h_{l+1}(m) \to \ldots \to P\big(F_{L+1} \circ \ldots \circ F_l(h_{l-1}), Z(I)\big).$$

  In this case we have:

  $$\begin{aligned} \delta_i^l &= \frac{\partial J_I}{\partial h_l(i)} \frac{\partial h_l(i)}{\partial \hat{h}_l(i)} = \frac{\partial J_I}{\partial h_l(i)} s_l'(\hat{h}_l(i)) = \left(\sum_m \frac{\partial J_I}{\partial \hat{h}_{l+1}(m)} \frac{\partial \hat{h}_{l+1}(m)}{\partial h_l(i)} \right) s_l'(\hat{h}_l(i)) \\ &= \left(\sum_m \delta_m^{l+1} w_{l+1}(m,i) \right) s_l'(\hat{h}_l(i)). \end{aligned}$$

  Note that, for internal layers, the nonlinear activation functions are real functions s_l applied pointwise: therefore the derivatives $\partial_i \big(s_l(\cdot)\big)_i$ are simply given by s_l'. The non-differentiability of the ReLU activation function is handled in practice by setting its derivative at zero to be either 0 or 1.

To sum up, the derivative in (3.12) is given by the following *backward* iterative rule:

$$\frac{\partial J_I(w,b)}{\partial w_l(i,j)} = \delta_i^l h_{l-1}(j), \quad \text{with} \quad \delta_i^l = \begin{cases} \partial_1 P\big(h_{L+1}(i)\big) s'_{L+1}(\hat{h}_{L+1}(i)) & \text{if } l = L+1 \\ s'_l(\hat{h}_l(i)) \sum_m \delta_m^{l+1} w_{l+1}(m,i) & \text{otherwise.} \end{cases}$$

The gradients $\nabla_w J(w,b)$ and $\nabla_b J(w,b)$ may alternatively be computed numerically through a finite difference scheme; however, in addition to being approximate, this would also be very computationally expensive. All the experiments that will be presented in Section 3.3 have been carried out using PyTorch [118], a popular deep learning framework providing a powerful automatic differentiation tool which keeps track of the network graph: this allows to "backpropagate" the gradients of all the internal operations by applying the chain rule to their *analytically computed* derivatives, as outlined above.

Training, validation and testing

A typical learning routine involves splitting the dataset $\mathscr{X}$ into a larger *training set* $\mathscr{X}^{train}$ and a smaller *testing set* $\mathscr{X}^{test}$, and only using the former to determine A and b in the optimization process. That is, the sum in (3.5) is only taken over $\mathscr{X}^{train}$. Precisely, the quantity to be minimized w.r.t. the parameters w and b of the network is

$$J(w,b) = \mathscr{L}\big(F^{(w,b)}, Z, \mathscr{X}^{train}\big) = \frac{1}{|\mathscr{X}^{train}|} \sum_{I \in \mathscr{X}^{train}} P\big(F^{(w,b)}(I), Z(I)\big) + R\big(F^{(w,b)}\big).$$

This procedure is referred to as the *training* phase. The ability of the learned functional to *generalize* the approximation of Z to *unseen* examples $I \in \mathscr{H}_0 \setminus \mathscr{X}^{train}$ can then be *tested* onto $\mathscr{X}^{test}$, which is contained in $\mathscr{H}_0 \setminus \mathscr{X}^{train}$ and on whose elements the correct value of Z is known.

As mentioned before, a critical role in the generalization ability of the network is played by the regularization term R. Since the exact function Z is only known on a subset of its domain, if no bounds are imposed on the complexity of the space, an approximating function F can be learned for which the loss function is zero on the training data; however, loosely speaking, this may be the result of learning the data "by heart" rather than constructing meaningful representations for pattern recognition. This phenomenon is commonly referred to as *overfitting*, and leads to poor performances on the testing data. We cite [24] as a solid reference on these topics.

As a matter of fact, the training problem of most neural networks produces error surfaces that are non-convex; it has been shown that even a single-neuron network can have exponentially many distinct local minima [11]. A common approach for "empirically" addressing this issue is repeat the optimization process multiple times with different random initialization seeds. On the other hand, finding a global minimum on the training set might actually lead to overfitting. A popular technique to avoid overfitting is *validation-based early stopping* [111, 122], guided by "verification" steps taken during the training phase. That is, the original dataset is split into $\mathscr{X} = \mathscr{X}^{train} \bigcup \mathscr{X}^{val} \bigcup \mathscr{X}^{test}$; at every fixed number of training epochs (on $\mathscr{X}^{train}$) the loss function is evaluated onto the *validation set* $\mathscr{X}^{val}$. Since the network parameters are updated independently of the elements of $\mathscr{X}^{val}$, evaluating the loss function on this set allows to track the generalization ability of the network during the optimization: even if the loss function computed onto the training set is decreasing during the minimization process, this may not be the case for the same function computed on the "unseen" examples; in fact, it may even start increasing. The early stopping method consists in choosing the number of training epochs according to the behavior of the loss function on the validation data – i.e. the optimization algorithm is stopped as soon as the validation loss function stops decreasing.
From the theoretical point of view, many recent works have tried to characterize the error surfaces of DNNs. Although local minima are in general not equivalent, the probability of finding a local minimum with a high value appears to decrease with network size [37]. It is also worth mentioning that several recent works have focused on the (apparently much more critical) issue of saddle points in the error surfaces (see e.g. [86]).

According to the classical (also called "vanilla") gradient descent algorithm, the sum in (3.11) is taken over the whole training set $\mathscr{X}^{train}$. In most cases, however, a variant called *stochastic gradient descent* (SGD) [124] is employed when training neural networks: this performs separate update steps for a sequence of "mini-batches" of fixed size, $\mathscr{X}_1, \dots, \mathscr{X}_{\overline{m}} \subseteq \mathscr{X}^{train}$. That is, the training set is split into $\mathscr{X}^{train} = \bigcup_m \mathscr{X}_m$; for each $m \leq \overline{m}$, a functional J_m is defined as in (3.11) by only summing over $\mathscr{X}_m$, and the parameters are modified by applying one step of the update rule in (3.12) for J_m: each epoch implies looping over mini-batches until the whole dataset has been used. This method allows to employ highly optimized matrix operations that lead to a very efficient computation of the gradient w.r.t. a mini-batch. Note that taking a small batch size may lead to heavy fluctuations in the values of the functional across iterations: this makes it possible to possibly jump out of the current basin to reach potentially better local minima. Morever, it has been shown that SGD displays the same convergence behavior as vanilla gradient descent for slowly decreasing learning

rates [28].
Further, several "adaptive" variants of SGD have recently been introduced that adapt the learning rate to each parameter by enforcing smaller updates for parameters associated with frequently occurring features. One of the most popular among these methods is *Adaptive Moment Estimation* (Adam) [87], which also takes into account a momentum estimation for updating the learning rate. We refer to [126] for a useful review on gradient descent algorithms.

Image data

The inputs $I \in \mathscr{H}_0$ of a DNN can represent a variety of different data: they can encode images, text, audio signals, chemical features, just to name a few. In the present context, we wish to focus on computer vision tasks, i.e. on situations where the input space contains image data. According to the notations introduced above, an image can be represented as a function

$$I : \mathscr{G}_0 \to \mathbb{R},$$

where $\mathscr{G}_0 = [\alpha_1, \alpha_2] \times [\beta_1, \beta_2] \subseteq \mathbb{R}^2$ for grayscale images and $\mathscr{G}_0 = [\alpha_1, \alpha_2] \times [\beta_1, \beta_2] \times \{1,2,3\}$ for RGB images. In the discrete setting, this translates into a $H_0 \times W_0 \times n_0$ matrix with $n_0 = 1$ or $n_0 = 3$ respectively. In the following, we shall generally denote input images by $I(u,v,c)$ for the general formulation, and by $I(i,j,c)$ when focusing on the discrete implementation.

From now on, we will focus on image classification tasks. That is, the networks' inputs will be images I, and we will adopt the cross entropy as a fiducial term in the loss function. In this case, the generalization ability of the network can be quantified through its *accuracy* on the testing set, simply defined as the percentage of elements of $\mathscr{X}^{test}$ that get correctly classified by the trained network.

3.1.2 CNNs for image classification

A breakthrough in deep learning for computer vision was reached with the introduction of a particular class of DNN architectures, namely Convolutional Neural Networks (CNNs). The characterizing feature of these networks is the presence of layers that are not fully connected as in classical deep architectures: instead, translation invariance is enforced by local convolutional windows shifting over the spatial domain, thus allowing for the weights to be shared across different locations of the image. The analogy with localized receptive profiles in biological vision is strong. The processing of early visual areas is classically

modeled as the *lifting* of an image to a feature space through a bank of filters with a localized support (see also Section 1.1.2); the first convolutional layer of a CNN implements an analogous mechanism, i.e. the linear functional W_0 is a *convolution* operator

$$\begin{aligned} A_0 I(u,v,k) := \psi_k^0 * I(u,v) &= \int_{\mathscr{G}_0} \psi_k^1(u',v',c) I(u-u',v-v',c) du'dv'dc \\ &= \sum_{c\in\{1,2,3\}} \int_{\beta_1}^{\beta_2} \int_{\alpha_1}^{\alpha_2} \psi_k^1(u',v',c) I(u-u',v-v',c) du'dv', \end{aligned}$$

where $\{\psi_k^1\}_{k=1,\dots,n_1}$ is a bank of filters acting on images $I(u,v,c)$. The subsequent convolutional layers are defined similarly: for each $l \in \{0,\dots,L-1\}$ we have

$$A_l h_l(u,v,k) := \psi_k^{l+1} * h_l(u,v),$$

where $\{\psi_k^{l+1}\}_{k=1,\dots,n_{l+1}}$ is the bank of filters associated to the $(l+1)$-th layer. The activation of the $(l+1)$-th layer is then obtained as

$$h_{l+1}(u,v,k) = F_l h_l(u,v,k) = s_{l+1}\left(A_l h_l(u,v,k) + b_l(k)\right) = s_{l+1}\left(\psi_k^{l+1} * h_l(u,v) + b_l(k)\right). \tag{3.16}$$

Each activation $h_l(u,v,k)$ is defined onto a *feature space* $\mathscr{G}_l \subseteq \mathbb{R}^2 \times \{1,\dots,n_l\}$ encoding a *spatial* 2D component and a *feature* index $k \in \{1,\dots,n_l\}$.

The spatial shifting of the filters introduces a critical constraint in the architecture, thus significantly reducing the degrees of freedom (corresponding to the number of trainable parameters) w.r.t. a regular DNN. Note that a smaller constraint is also applied to the bias terms, which do not depend on the spatial coordinates (u,v) but only on the feature index k: that is, $b_l(u,v,k) = b_l(k)$.

The final layer of the network is typically not convolutional, i.e. the operator A_L is not translation invariant. A_L maps to an output space $\mathscr{H}_{L+1}$ whose shape depends on the task: in the case of multiclass classification, $\mathscr{H}_{L+1}$ is as in (3.6), where M must match the number of classes, and the last activation function s_{L+1} is the softmax function (3.7). It is also not uncommon to have more than one fully connected layers, with pointwise nonlinear activation functions (e.g. ReLUs) interposed between them.

We now focus on the discrete formulation. In this setting, the update rule (3.16) reads:

$$h_{l+1}(i,j,k) = s_{l+1}\left(\sum_{i',j',k'} \psi_k^{l+1}(i',j',k') \cdot h_l(i-i',j-j',k') \; + \; b_l(k)\right).$$

Let us recall the shapes of the tensors involved in these operations and fix some notations. First, the input image $I = h_0$ has size $H_0 \times W_0 \times n_0$, where n_0 equals 1 or 3, and the filters must have either 1 or 3 channels accordingly: the bank of filters $\{\psi_k^1\}_{k=1,\dots,n_1}$ is a $d_1 \times d_1 \times n_1 \times n_0$ tensor, where n_1 is the number of filters of the first layer. The convolution between I and the filters ψ_k^1 gives an $H_1 \times W_1 \times n_1$ tensor, to which a *bias* vector $b_1 \in \mathbb{R}^{n_1}$ is added along the third component, to obtain the output h_1 of the layer. Likewise, for each l a bank of filters $\{\psi_k^l\}_{k=1,\dots,n_l}$ of size $d_l \times d_l \times n_l \times n_{l-1}$ is convolved with a tensor h_l of size $H_l \times W_l \times n_l$, and a bias vector $b_l \in \mathbb{R}^{n_l}$ is added to the result.
Another layer that can optionally be interposed between convolutional layers consists of the application of a *pooling* operation $\mathscr{P}$: this performs a downsampling of its input over the *spatial variables* (i.e. the "depth" dimension remains unchanged), typically by taking the maximum or by averaging over small neighborhoods. For instance, if a pooling layer is applied to an activation h_l of size $H_l \times W_l \times n_1$ over 2×2 squares, then the output $\mathscr{P}(h_l)$ will be an $\frac{H_l}{2} \times \frac{W_l}{2} \times n_l$ tensor. This downsampling operation reduces the dimensionality and introduces invariance to small shifts and distortions. The insertion of pooling layers has a neural motivation as well: the receptive fields of visual neurons tend to get wider and wider moving towards higher cortical layers, and subsampling the spatial dimension of a feature space is equivalent to taking filters with a wider support in the next layer.
In these discrete terms, the non-invariance of the last linear operator A_L yields a *fully connected* layer. In practice, the output h_L of the last convolutional layer is "flattened" to a vector of length $S = H_L \cdot W_L \cdot n_L$ and transformed through a matrix $\{w_L(j,i)\}_{i,j}$ of size $S \times M$ and a bias vector of length M (where M is the number of classes), thus yielding a vector with M entries.

3.1.3 Recurrent CNNs (RecCNNs)

The human visual system, as outlined in the preceding chapters, relies not only on a hierarchical transmission of signals, but also on a *lateral* spreading of information. On the other hand, the sequential structure of a CNN implements a purely feedforward mechanism: the output of each layer only depends on the activation of the preceding layer. Recurrent CNNs [101, 135] (RecCNNs) are a modification of this kind of architecture, where *lateral* connections of convolutional type are added to a regular CNN. This means that the network includes not only connections from one layer to the next one, but also connections from a layer to itself: this can be described by introducing a time parameter t. The activation of the l-th hidden layer at time t, which we denote by h_l^t, is a function of:

- h_{l-1}^t (the output of the preceding layer at the same time step t);

- h_l^{t-1} (the output of the same layer at time $t-1$).

Specifically, following the notations introduced for CNNs, we have:

$$h_l^t = s_l\big(\underbrace{\phi^l * h_l^{t-1}}_{\text{lateral}} + \underbrace{\psi^l * h_{l-1}^t}_{\text{feedforward}} + b_l\big) \tag{3.17}$$

for all $t, l > 0$, where $h_0^t = I$ for all $t > 0$, and $h_l^0 \equiv 0$ for all $l > 0$. Here, $\phi^l = \{\phi_k^l\}_k$ denotes the convolutional filters defining the lateral connections at the l-th layer. Since the coefficients of the net are kept the same at each time step, the only additional parameters w.r.t. a standard CNN are the recurrent weights ϕ^l. In a representation of the network as a graph, the recurrent

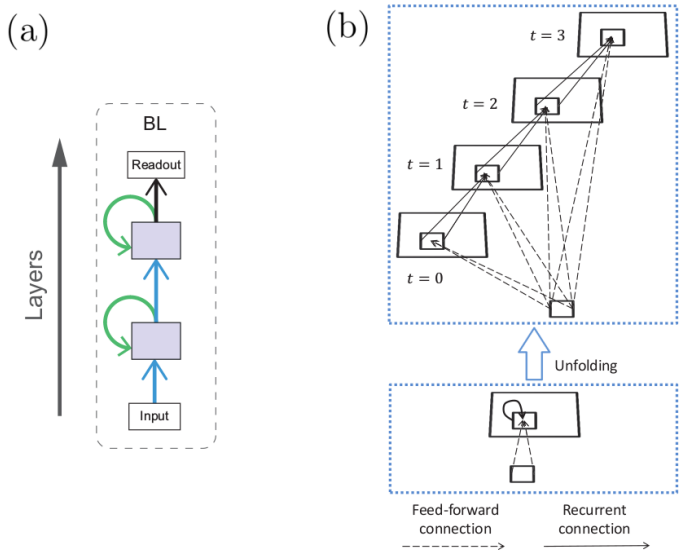


Figure 3.2 (a) Schematic representation of an RecCNN, from [135]. (b) A recurrent convolutional layer from [101], unfolded for $T = 3$ time steps.

connections can be represented in a compact way as arrows going from a layer to itself, see Figure 3.2(a). In order to represent the "unfolded" network, one needs to make explicit the time component, as displayed on the top in Figure 3.2b (for 3 time steps).
Recurrent neural networks (RNNs) are often employed to process sequential inputs, e.g. audio recordings, video or text. In such cases, a new input $I^t = h_0^t$ is fed into the network at each time step. On the contrary, in RecCNNs the input image is *static*, i.e. it is kept fixed at each time step: the time variable only affects the processing.

3.2 Kernel CNNs

The lateral connections in RecCNNs are completely learned and independent of the feedforward ones. Moreover, the inclusion of these connections in a CNN increases its complexity in terms of trainable parameters.
Our main contribution is to induce the metric structure presented in Chapter 2 on the layers of a CNN. Accordingly, we will first modify the loss function with a regularizing term defined by means of a gradient in the metric. After that, we will use the kernel defined in Chapter 2 to introduce convolutional lateral connections in the net. This means that the recurrent kernels are obtained as a function of the feedforward ones. As such, their inclusion in the structure of a CNN does not introduce any new parameters. Moreover, this construction allows to enrich the model with biologically inspired prior knowledge on the geometrical structure of lateral connections.

In the following, we first show how to define connectivity kernels associated to each convolutional layer and outline the proposed network architecture; we then introduce a testing framework to analyze the performance of the networks in a task of classification of corrupted images, focusing on types of image degradation where mechanisms of perceptual completion and global object analysis are required for correct classification.

3.2.1 The feature space as a metric space and the connectivity kernels

Applying the model defined in Chapter 2, we first endow the sets $\mathscr{G}_l$ of features with a structure of metric space. Note that the filters $\{\psi_k^l\}_k$ are all centered at zero; they are then shifted over the spatial domain in the convolution operation. Similarly to the case of the bank of learned filters of Section 2.4.5, we can define a family of translation-invariant filters by $\{\psi_k^l(\cdot - i, \cdot - j)\}_{(i,j,k)\in\mathscr{G}_l}$. The distance $d_l : \mathscr{G}_l \times \mathscr{G}_l \to \mathbb{R}$ associated to the l-th convolutional layer is nothing but the L^2 distance between such filters:

$$d_l\big((i,j,k),(i_0,j_0,k_0)\big)^2 = \sum_{i',j',c} |\psi_k^l(i'-i,j'-j,c) - \psi_{k_0}^l(i'-i_0,j'-j_0,c)|^2. \tag{3.18}$$

The invariance of d_l w.r.t. translations yields

$$d_l\big((i,j,k),(i_0,j_0,k_0)\big) = d_l\big((i-i_0,j-j_0,k),(0,0,k_0)\big).$$

Therefore, the distance between all couple of points is completely characterized by the distance from the points of the type $(0,0,k_0)$. The same holds for the associated *correlation*

kernel, computed by taking the L^2 scalar product between the filters ψ^l. For the sake of brevity, we denote

$$\tilde{K}_l(i,j,k_0,k) \equiv \tilde{K}_l\big((i,j,k),(0,0,k_0)\big) = \nu\Big(\sum_{i',j',c} \psi^l_{k_0}(i',j',c)\cdot\psi^l_k(i-i',j-j',c)\Big). \quad (3.19)$$

Here, ν is the sigmoidal activation function

$$\nu(z) = \frac{1}{1+e^{-z}}.$$

The indices are let vary so that the product $\psi^l_{k_0}(i',j',c)\cdot\psi^l_k(i-i',j-j',c)$ is computed whenever the supports of the two filters overlap. Therefore, if the size of the bank of filters ψ^l is $d_l\times d_l\times n_l\times n_{l-1}$, then the size of the kernel is obtained as $(2d_l-1)\times(2d_l-1)\times n_l\times n_l$. The final kernel is then given by

$$K_l(i,j,k_0,k) = N[\tilde{K}_l](i,j,k_0,k), \quad (3.20)$$

where N is the same normalization operator introduced in [47] and employed to normalize the kernel in Chapter 2. Specifically, in the current case of a discrete, translation-invariant kernel $\tilde{K}_l\big((i,j,k_0),(0,0,k)\big) = \tilde{K}_l(i,j,k_0,k)$, the operator reads:

$$N[\tilde{K}_l](i,j,k_0,k) := \frac{\tilde{K}_l{}^{(1)}(i,j,k_0,k)}{\sum_{i',j',k'}\tilde{K}_l{}^{(1)}(i',j',k_0,k')},$$

where

$$\tilde{K}_l{}^{(1)}(i,j,k_0,k) = \frac{\tilde{K}_l(i,j,k_0,k)}{\sum_{i',j',k'_0}\tilde{K}_l(i',j',k'_0,k)\sum_{i',j',k'}\tilde{K}_l(i',j',k_0,k')}.$$

3.2.2 A loss function with a metric gradient term

The definition of a distance on the feature space allows us to add to the loss function a new regularization term R depending on the *metric gradient* of the filters. In this way, the loss function will take the expression of a classical functional of calculus of variation. As already outlined in Section 2.1, a typical way to define a gradient norm in a metric setting is by averaging a finite difference taken w.r.t. the distance onto a ball, see Eq. (2.9): the pointwise value is obtained as the radius of the ball goes to zero. Here, we take the same

approach. We start by defining, for any $(u,v,k) \neq (u_0,v_0,k_0)$,

$$\nabla\psi^l(u_0,v_0,k_0) = \left\{\nabla_{(u,v,k)}\psi^l(u_0,v_0,k_0)\right\}_{u,v,k} := \left\{\frac{\psi^l_k(u,v)-\psi^l_{k_0}(u_0,v_0)}{d^l\big((u,v,k),(u_0,v_0,k_0)\big)}\right\}_{u,v,k}. \quad (3.21)$$

Note that, thanks to the translation invariance of the spatial variables due to the convolutional structure, for these coordinates we may adopt the usual finite difference scheme to define our gradient, i.e. consider a perturbation (u_0+h,v_0+h) of (u_0,v_0) by a fixed small step h. However, the notion of "small step" depends on the metric taken into consideration. Since the metric d^l is defined onto the feature space $\mathscr{G}_l$, we say that (u,v,k) is a "small perturbation" of (u_0,v_0,k_0) when $d^l\big((u,v,k),(u_0,v_0,k_0)\big)$ is small. In this case, for fixed (u_0,v_0,k_0), one may regard $\nabla_{(u,v,k)}\psi^l(u_0,v_0,k_0)$ as a directional derivative. We then define

$$|\nabla\psi^l(u_0,v_0,k_0)|^2 := \frac{1}{|B_\varepsilon(u_0,v_0,k_0)|}\int_{B_\varepsilon(u_0,v_0,k_0)\setminus\{(u_0,v_0,k_0)\}}\big(\nabla_{(u,v,k)}\psi^l(u_0,v_0,k_0)\big)^2 dudvdk,$$

where $B_\varepsilon(u_0,v_0,k_0)$ is the ball of d^l of radius ε around (u_0,v_0,k_0). On the other hand, since $\nabla_{(u,v,k)}\psi^l(u_0,v_0,k_0) = \nabla_{(u-u_0,v-v_0,k)}\psi^l(0,0,k_0)$, we can simply consider the gradient computed at the central point of the filter $\psi^l_{k_0}$. That is, we can set $(u_0,v_0)=(0,0)$ and define the regularization term

$$R_l := \lambda\sum_{k_0}|\nabla\psi^l(0,0,k_0)|^2, \quad (3.22)$$

where λ is a parameter weighting the contribution of the term in the loss function. For the sake of readability, we omitted in (3.21) the dependence on the *channel* component $c\in\{1,\dots,n_{l-1}\}$: for each c, we are actually taking $\psi^l_k(u,v,c)$ and $\psi^l_{k_0}(u_0,v_0,c)$ in (3.21). This yields a different gradient for each channel c. If $n_{l-1}>1$, we also sum over c in (3.22).

3.2.3 The KerCNN architecture

We now transpose the notion of connectivity introduced in Chapter 2, and notably the propagation of Eq. (2.39), directly into the structure of a CNN. We shall refer to the resulting network architecture as KerCNN. The update rule of a KerCNN layer is inspired by the iterative procedure outlined in Chapter 2 to describe the propagation of neural activity in V1. Specifically,

$$\begin{cases} h^1_l = s_l(\psi^l * h^{T_{l-1}}_{l-1} + b_l) \\ h^t_l = \frac{1}{2}\Big(K_l * h^{t-1}_l + h^{t-1}_l\Big) \quad \text{for } 1<t\leq T_l, \end{cases} \quad (3.23)$$

where the kernels K_l are defined from the filters ψ^l as in (3.20).
The output of the $(l-1)$-th layer is first lifted to the l-th feature space through a feedforward step, yielding an activation h_l^1, which is then updated through convolution with the kernel K_l, as in (2.39). The new output h_l^2 is defined by averaging between this updated activation $K_l * h_l^1$ and the original activation h_l^1. The same procedure is repeated, yielding a sequence of activations h_l^t, until a fixed stopping time T_l is reached. Note that each layer has its own stopping time. We remark that convolutions with the kernel K_l are taken with appropriate padding, so that the size of h_l^t is preserved at every iteration.
The intuitive idea here is that K_l behaves like a "transition kernel" on the feature space of the l-th layer, slightly modifying its output according to the correlation between its filters: the activation of a filter encourages the activation of other filters highly correlated with it.

3.2.4 Task: stability to corrupted images

Our guess is that the insertion of structured lateral connections may improve the performance of a CNN in tasks related to perceptual mechanisms of global shape analysis and integration. In particular, we focus on a task of classification of *corrupted* images.
We start by considering a fixed CNN architecture with L convolutional layers as a base model, and we modify it by inserting the structured lateral connections as described in Section 3.2.3. Note that this yields a different KerCNN architecture for each combination of the layers' stopping times $(T_1,\dots,T_L)$. If all stopping times are 1, the model coincides with the base CNN.
Given a labeled image dataset, each model is trained in a supervised way to perform classification. No corruption is applied to the images during the training phase: the training examples are the original images from the dataset (up to some basic pre-processing). The actual experiment consists in analyzing the ability of the models to *generalize* the classification to the degraded images, by comparing their classification accuracy on *corrupted testing images*. We examine different kinds of image corruption:

1. Gaussian patches occluding the image, similar to the ones in [139];

2. Disruption of local contours, in analogy with the study presented in [13], obtained by subdividing the image into horizontal or vertical strips and by shifting each of these strips by a random number of pixels $d \in \{0,\dots,D\}$;

3. Adversarial attacks through the Fast Gradient Sign Method (FGSM) [74]. FGSM, one of the most popular attack methods, simply adjusts the input image by taking a *gradient ascent* step to *maximize* the loss function. Precisely, the perturbed image $\tilde{I}$ is

obtained as

$$\tilde{I} = I + \varepsilon \cdot sign\left[\nabla_I \mathscr{L}\left(F, Z, \mathscr{X}^{test}\right)\right] \tag{3.24}$$

where $\mathscr{L}$ is the loss function (3.5).

In all three cases, the amount of degradation can be quantified by a parameter: the standard deviation γ of the Gaussian patches, the maximum displacement D of the strips, and the step ε of the FGSM. We expect the classification accuracy of the models to decrease as the amount of degradation increases: the more stable a model is to these perturbations, the slower the drop in performance w.r.t. the degradation parameter.

Lipschitz stability

The analysis of stability of a network to perturbations of its inputs is closely linked to its Lipschitz properties. If $\Phi(I)$ is the feature vector associated to the input I at a certain layer, analyzing the stability of this representation w.r.t. some type of degradation means making sure that a small change in the input provokes a small change in the feature vector. This can be expressed through a Lipschitz condition

$$\|\Phi(I') - \Phi(I)\| \leq C_\Phi \|I' - I\|$$

with suitable norms on the input and output spaces, where I' is a perturbation of I. For instance, I' can be obtained as a deformation of I through a diffeomorphism $\tau : \mathbb{R}^d \to \mathbb{R}^d$, i.e. $I'(u) := I(u - \tau(u))$. Another kind of perturbation may be given by the addition of a noise term, i.e. $I'(u) = I(u) + \varepsilon(u)$.
In special cases such as S. Mallat's scattering networks [104, 32], where the filters are pre-defined, the stability properties of the model can be established analytically. A similar analysis is carried out in [23] with learned filters, but for an architecture constrained by the projection of the activation at each layer onto a *reproducing kernel Hilbert space* (see also [103]). For general network architectures, several algorithms to estimate the Lipschitz constant have been developed [151, 153, 14, 142].

We now take Φ to be the activation h_l^1 of a certain layer after a standard convolutional step. We write $h_l^1(I)$ to make explicit the dependence on the input. We can examine the stability properties of taking the subsequent iterative steps in (3.23) by comparing the norms $\|h_l^{t+1}(I') - h_l^{t+1}(I)\|_2$ and $\|h_l^t(I') - h_l^t(I)\|_2$. Precisely, we have the following estimate in terms of the kernels K_l.

Proposition 3.1. Fix $l \in \{1,\dots,L\}$ and I, I' in the input space $\mathscr{H}_0$. Let $h_l^t(I), h_l^t(I') \in \mathscr{H}_l$ be defined as in (3.23) for each t. Then we have

$$\|h_l^{t+1}(I') - h_l^{t+1}(I)\|_2 \leq \frac{1}{2}\left(1+\sqrt{S_l}\right)\|h_l^t(I') - h_l^t(I)\|_2, \tag{3.25}$$

where $S_l := \sup_q \int_{\mathscr{G}_l} K_l(p,q)dp$.

Proof. Comparing the two norms in (3.25) means estimating the Lipschitz constant of the mapping $\mathscr{C}_l : \mathscr{H}_l \to \mathscr{H}_l$ defined by $\mathscr{C}_l[h] := \frac{1}{2}(h + K_l * h)$. We have:

$$\begin{aligned}\|K_l * h\|_2^2 &\leq \int_{\mathscr{G}_l}\left(\int_{\mathscr{G}_l} K_l(p,q)h(q)^2 dq\right)\underbrace{\left(\int_{\mathscr{G}_l} K_l(p,q')dq'\right)}_{=1}dp \\ &= \int_{\mathscr{G}_l} h(q)^2\left(\int_{\mathscr{G}_l} K_l(p,q)dp\right)dq \leq \left(\sup_q \int_{\mathscr{G}_l} K_l(p,q)dp\right)\|h\|_2^2 = S_l\|h\|_2^2.\end{aligned}$$

Therefore, thanks to the linearity of the mapping $\mathscr{C}_l$, an upper bound for its Lipschitz constant is given by $\frac{1}{2}\left(1+\sqrt{S_l}\right)$. That is,

$$\|\mathscr{C}_l[h'] - \mathscr{C}_l[h]\| \leq \frac{1}{2}\left(1+\sqrt{S_l}\right)\|h' - h\| \quad \forall h, h' \in \mathscr{H}_l.$$

The thesis follows from the fact that $h_l^{t+1} = \mathscr{C}_l[h_l^t]$. □

Remark 3.1. Note that the above argument is not specific to a particular type of perturbation of I. In fact, the three types of image degradation described before may be expressed as different kinds of transformations.

1. A Gaussian patch centered at $(\bar{u}_1, \bar{u}_2)$ superposed onto an image I can be expressed as the multiplication of $I(u)$ by a function $1 - \exp\left(\frac{(u_1-\bar{u}_1)^2+(u_2-\bar{u}_2)^2}{\sigma^2}\right)$.
2. The local edge disruption may be approximated by the action of a diffeomorphism τ as seen before.
3. The FGSM method modifies the image through an additive "adversarial" term, see Eq. (3.24), and can therefore be included in the additive noise perturbations.

Remark 3.2. The kernel K_l is not normalized w.r.t. the first variable p, and the bound on the Lipschitz constant of $\mathscr{C}_l$ depends on the integral of K_l w.r.t. p. Therefore, S_l cannot be replaced by 1 in general. This can be done if the (symmetric) unnormalized kernel

$\tilde{K}_l = \nu\left(\psi^l * \psi^l\right)$ computed around any point of the feature space has constant L^1 norm. Indeed, in terms of $\tilde{K}_l$ we have:

$$S_l \leq \frac{\sup_p \int_{\mathscr{G}_l} \tilde{K}_l(p,q)dq}{\inf_p \int_{\mathscr{G}_l} \tilde{K}_l(p,q)dq}.$$

Therefore, the stability of the operator in the Lipschitz sense depends on how close the integral $\int_{\mathscr{G}_l} \tilde{K}_l(p,q)dq$ is to being constant in p. This in turn revolves around the "homogeneity" of the bank of filters ψ^l. In particular, if the bank of filters is obtained through the action of a group, the corresponding kernel $\tilde{K}$ is invariant w.r.t. the group law in the sense of (2.18): as a consequence, the integral $\int_{\mathscr{G}_l} \tilde{K}(p,q)dq$ is constant in p and $S_l \leq 1$ in this case.

3.3 Results

In this section, we provide a complete analysis of the results obtained on MNIST [97], a popular database of images of handwritten digits for classification. We compare the performance of a 2-layer CNN model with the ones of the corresponding "regularized" network introduced in Section 3.2.2, as well as the KerCNN and RecCNN models, for varying stopping times T_1 and T_2. The models will be trained on the MNIST dataset, and tested for the task outlined in Section 3.2.4, for different types and amounts of image degradation. We will finally give a synthetic report on the same study carried out on the Kuzushiji-MNIST [44], Fashion-MNIST [149] and CIFAR-10 [92] datasets. All the experiments were carried out using PyTorch [118].

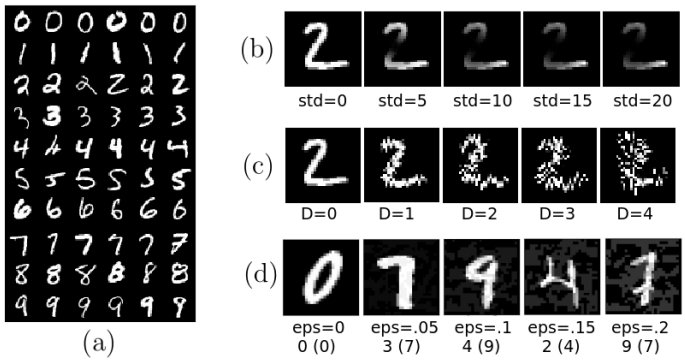


Figure 3.3 (a) A sample from the MNIST dataset. (b) A testing image from the MNIST dataset corrupted by a Gaussian patch of increasing standard deviation. (c) A testing image from the MNIST dataset corrupted by an increasing amount of local contour disruption D. (d) Testing images from different classes, perturbed by applying the FGSM to the base CNN with increasing values of ε. Below each image, we display the classified label, as well as the correct label (in brackets). Apart from the unperturbed image ($\varepsilon = 0$), all the images are misclassified by the CNN.

3.3.1 Implementation

The MNIST dataset

The MNIST dataset [97] consists in 70000 labeled 28×28 grayscale images of handwritten digits from 0 to 9: see the sample in Figure 3.3a. We trained the networks on the 60000 training images, and we tested them on the 10000 testing images corrupted by the three types of degradation mentioned above. Some examples are displayed in Figure 3.3b-d.

Base model

Our base model is a CNN with 2 hidden layers. We take 16 filters of size 5×5 in the first convolutional layer and 16 filters of size $5 \times 5 \times 16$ in the second convolutional layer, each followed by ReLU activation and max pooling, and a fully connected last layer followed by softmax activation. The total number of trainable parameters is 7482. We then compare this

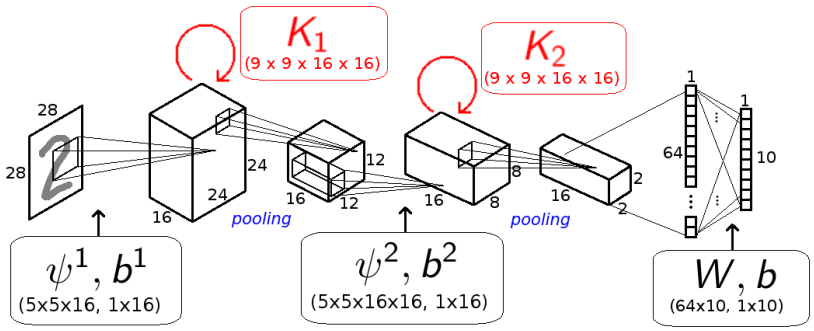


Figure 3.4 Our KerCNN model with structured lateral connections defined by kernels K_1 and K_2.

model with the one obtained from it by inserting the structured lateral connections. See Figure 3.4 for a description of the model. The lateral kernels in this case have size $9 \times 9 \times 16 \times 16$. We also analyze the performance of the model obtained from the CNN by inserting recurrent connections according to the RecCNN model, i.e. through the update rule (3.17). As said before, lateral connections given by the kernels K_l do not introduce new parameters in the starting CNN. On the other hand, the insertion of *learned* lateral connections results in a model with more parameters than the base CNN: for example, the introduction of learned kernels of size $4 \times 4 \times 16 \times 16$ in the first layer of the base model would add 4096 new parameters to the original 7482. In the following, we consider a 7482-parameter version of the RecCNN, obtained by decreasing the size of feedforward filters in order to compensate for the extra recurrent parameters, as in [135].

Training details

All the models were trained to minimize the cross entropy classification loss (3.9), for 150 epochs with validation-based early stopping: to this end, the training set was split into 50000 training images and 10000 validation images. Adam optimizer was employed with the standard parameters indicated in [87], a batch size of 50 and the Xavier initialization scheme [73]; L^2 regularization with $\lambda = .0005$ was used. In the models including lateral connections (of any kind), recurrent dropout [133] with .2 probability was applied to the "horizontal" contributions. In RecCNNs, local response normalization (LRN) was applied after recurrent convolutional layers as in [101, 135]. The training and testing images were z-score normalized according to the mean and standard deviation computed across the whole training set. For each architecture (i.e. each combination of stopping times T_1 and T_2), 10 nets initialized with different random seeds were trained. The testing results displayed in the following are obtained by testing all 10 nets and averaging the classification accuracy over trials. Error bars (95% confidence intervals) are shown in the plots to keep track of the variability across initialization seeds.

Image perturbations

- We first consider testing images corrupted by *occlusions* in the form of Gaussian "bubbles" at random locations over the image, similar to the ones considered in [139]. Specifically, the image I' obtained by modifying the original input I through a patch centered at $(\overline{u}_1, \overline{u}_2)$ was implemented as:

$$I'(u) = (I(u) - b) \cdot (1 - g(u)) + b,$$

  where $g(u) := \frac{1}{2\pi\gamma^2} \exp\left(\frac{(u_1-\overline{u}_1)^2+(u_2-\overline{u}_2)^2}{2\gamma^2}\right)$ and b is the "background color". For the MNIST dataset, b was chosen to be the value at the upper left angle of each image. See Figure 3.3b. The number of patches per image was kept fixed to 4. The level of degradation is represented by the standard deviation γ of the Gaussian bubbles, expressed in pixels and varying in $\{0, 5, 10, 15, 25, 30\}$.

- In [13], evidence is provided that the feature extraction performed by deep CNNs mostly relies on local edge relations, rather than on global object shapes. Their experiments showed that, conversely to human vision, the networks' performance was much more robust to global shape changes preserving local features, than to a disruption of local contours preserving the global infomation. We hypothesized that the insertion of structured lateral connections in CNNs could make the models more

robust to these local perturbations. To automatically create a "local scrambling" of pixel information analogous to the one considered in [13], we proceeded as follows. For a fixed $D \in \mathbb{N}$, we subdivided the images into horizontal strips and shifted each of these strips by a number of pixels d_n, where $|d_n|$ was randomly chosen in $\{0,\dots,D\}$:

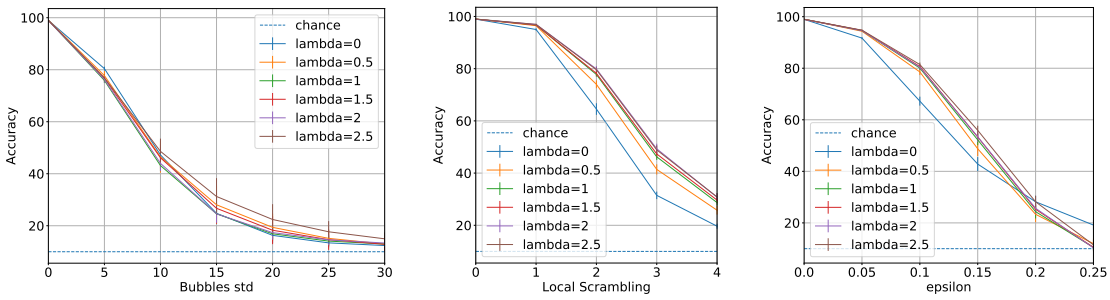
$$\forall u_2 \in [s_n, s_{n+1}),\ \forall u_1 \quad I'(u_1 + d_n, u_2) = I(u_1, u_2),$$

where $[s_n, s_{n+1})$ is the interval of u_2 in the n-th strip and d_n is the corresponding shift. We then repeated the procedure by subdividing the modified image into vertical strips and by shifting them as well. Some examples are displayed in Figure 3.3c. The amount of degradation is given in this case by the maximum displacement D expressed in pixels, which was kept the same for both horizontal and vertical strips. In the following experiments, D varies in $\{0,1,2,3,4\}$.

- Figure 3.3d shows some examples of images obtained through (3.24) applied to the base CNN for MNIST, for increasing values of ε. For sufficiently small ε, this perturbation results in an image that is almost identical to the original one to the human eye; however, these images are misclassified by the network. The corruption of the image is expressed by the parameter ε, varying in $\{0,.05,.1,.15,.2,.25\}$.

3.3.2 Results for the loss regularization

We start by displaying the results obtained for the above-mentioned tasks by simply considering the base CNN architecture and adding a regularization term R_1 associated to the connectivity kernel of the first layer, as in (3.22). We let the regularization parameter λ vary



(a) (b) (c)

Figure 3.5 Results for the "regularized CNN", with MNIST testing images corrupted through Gaussian patches (a), local edge disruption (b) and FGSM (c). In all three plots, the classification accuracy (y-axis) is plotted against the amount of degradation (x-axis). Each curve refers to a value of the regularization parameter λ.

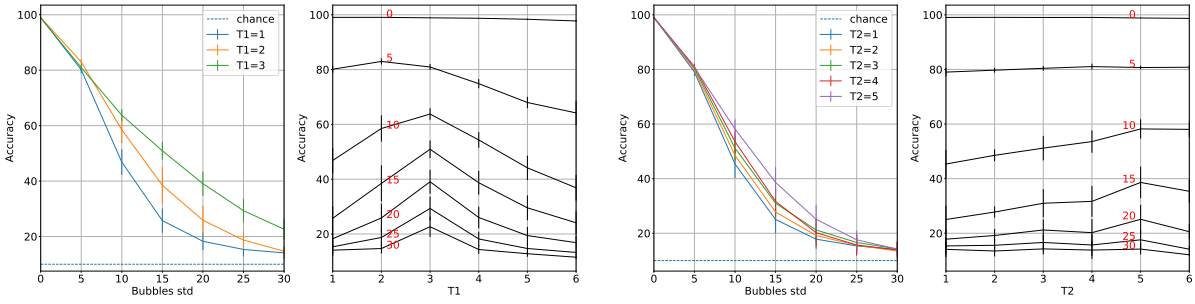
in $\{0,.5,1,1.5,2,2.5\}$, and we compared the resulting networks. Note that the base CNN is obtained for $\lambda = 0$. For all three generalization tasks we plotted the classification accuracy (y-axis) of the models for varying levels of degradation (x-axis). Each curve corresponds to a value of λ, and the dashed blue line displays the chance level accuracy (10%). Figures 3.5b and 3.5c show that the performance on images corrupted by local edge disruption and FGSM attacks (for most values of ε) is improved as the weight λ increases: this suggests that the constraint inserted through the regularization term indeed helps integrate the perturbed local features into global shapes. However, this is not the case for images occluded by Gaussian patches (see Figure 3.5a): this prior does not allow to "fill" in the occlusions. Indeed, this constraint is defined in terms of the *local* correlation measure encoded in the initial kernel, and it does not implement a proper propagation process defining a long-range connectivity. In the following, we directly modify the architecture according to (3.23), where the activations of the layers are *iteratively* updated through the kernel.

3.3.3 Results for KerCNNs

We hereafter compare the classification accuracy of the base CNN with the one of the corresponding KerCNN model, defined as described in Section 3.2.3, for varying amounts of image degradation and for different stopping times of KerCNN.

Gaussian patches

We first examine the KerCNN defined by inserting lateral connections in the first layer of the base CNN. Figure 3.6a(left) shows its classification accuracy for varying values of standard deviation γ of the Gaussian patches. The three graphs displayed are referred to different stopping times $T_1 = 1,2,3$, and the chance level accuracy is displayed as well (dashed blue line). Note that, for $T_1 = 1$, the model is the standard CNN with no lateral connections. The mean performance of these three nets on the original testing set (0 degradation) is almost identical ($99.0 \pm 0.1\%$). On the other hand, for increasingly degraded images the performance drops dramatically for the CNN ($T_1 = 1$, blue curve), while decaying much more slowly for increasing values of T_1. Note that the gap in classification accuracy between the CNN and the best KerCNN reaches ~ 25 points. After reaching its optimal value ($T_1 = 2$ for $\gamma \leq 5$ and $T_1 = 3$ for greater values), the performance drops again by taking further steps. For the sake of legibility, we displayed in the left plot only the curves up to the optimal value of T_1. The behavior of classification accuracy w.r.t. $T_1 \in \{1,\dots,6\}$ can be best appreciated in the right plot of Figure 3.6a, displaying a curve for each value of standard deviation.
We now analyze the performance of the KerCNN models with lateral connections:



(a) (b)

Figure 3.6 Results for MNIST testing images corrupted through Gaussian patches, for KerCNN with lateral connections in the first (a), resp. second layer (b). Left plots: classification accuracy (y-axis) at increasing values of γ (x-axis), displayed for stopping time $T_1 = 1,2,3$ (a), resp. $T_2 = 1,\dots,5$ (b). Right plots: classification accuracy (y-axis) for increasing values (x-axis) of T_1 (a), resp. T_2 (b), displayed for different values of degradation. Each curve refers to a value of std, displayed in red in correspondence of the curve.

- only in the second layer;
- in both layers.

Analogous to the preceding case, the optimal stopping time for the net with lateral connections in the second layer is $T_2 = 2$ for the original images, $T_2 = 4$ for a small degradation ($\gamma = 5$) and $T_2 = 5$ for greater values of standard deviation. Figure 3.6b(left) plots the accuracy against the level of degradation: we display the curves for $T_2 = 1,\dots,5$; the accuracy w.r.t. stopping times $T_2 \in \{1,\dots,6\}$ is plotted in Figure 3.6b(right), where each curve corresponds to a level of image degradation. The results show the same pattern as before, although with a smaller improvement (up to ~ 15 points between the base CNN and the model with optimal T_2).

It is interesting to note that the optimal number of iterations shifts towards higher values (for both layers) as the size of the occlusions increases. As mentioned before, the kernel K_l can be thought of as an *anisotropic* transition kernel on the space of activations of the l-th layer. As such, the repeated application of the lateral contribution given by these kernels may be interpreted as a spreading of activation, around each spatial location, along those orientations that are most activated at that point. Intuitively, this "compensates" for the gaps in the activation caused by the occlusions: the wider the gap, the higher the number of iterations of the kernel needed for the image to be consistently completed.

We finally study the combinatorics of stopping times $T_1, T_2 \in \{1,\dots,6\}$ in the two layers: Figure 3.7 displays the results for different levels of image degradation. For each combination of T_1 (x-axis) and T_2 (y-axis), the mean accuracy over all trials (color-coded) is displayed.

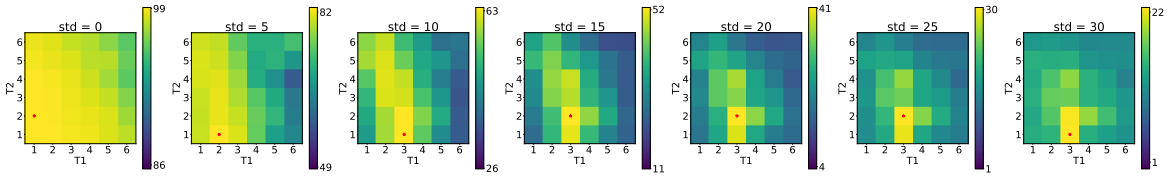
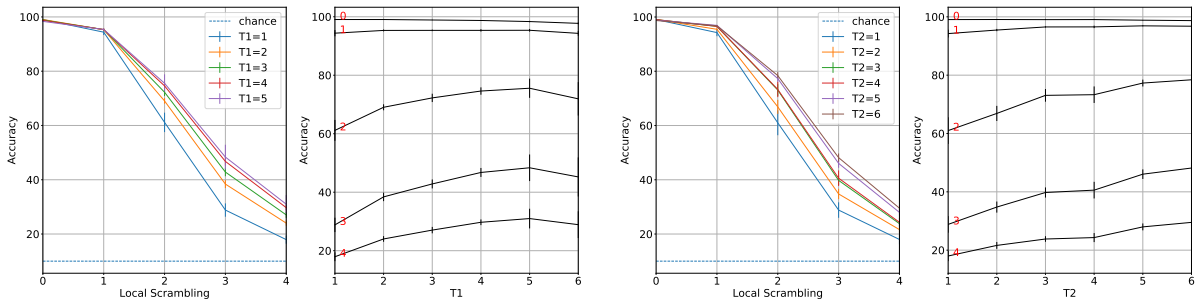


Figure 3.7 Classification accuracy (color-coded) for KerCNN for all combinations of $T_1, T_2 \in \{1,\dots,6\}$, displayed for $\gamma = 0,\dots,30$. The maximum value of accuracy is marked by a red star onto the corresponding cell.

Note that the highest values of accuracy lie on a diagonal that shifts towards higher values of both T_1 and T_2 as the level of degradation increases. It is interesting to observe that, for $\gamma = 15, 20, 25$, the optimal couple (T_1, T_2), highlighted by a red star, is one involving lateral connections in both layers.

Local contour disruption



(a) (b)

Figure 3.8 Results for MNIST testing images corrupted through local contour disruption, for KerCNN with lateral connections in the first (a), resp. second layer (b). Left plots: classification accuracy at increasing values of displacement D, displayed for stopping time $T_1 = 1,\dots,5$ (a), resp. $T_2 = 1,\dots,5$ (b). Right plots: classification accuracy for increasing values of T_1 (a), resp. T_2 (b), displayed for different values of degradation. Each curve refers to a value of D, displayed in red in correspondence of the curve.

As before, we compare the classification accuracy of the models for an increasing amount of degradation, given by the maximum displacement D. In this case, the performance of the models turns out to rise for increasing stopping times up to $T_2 = 6$ for the models with lateral connections in the second layer, while there is a peak in performance at $T_1 = 5$ for the ones with lateral connections in the first layer: see Figure 3.8. A similar situation is observed when analyzing the combinatorics of stopping times for the first and second layers, as shown in Figure 3.9: the optimal couple of values (T_1, T_2) shifts towards the maximum as

the displacement D increases, and the best accuracy is reached at $(T_1, T_2) = (6, 6)$ above a certain amount of degradation.

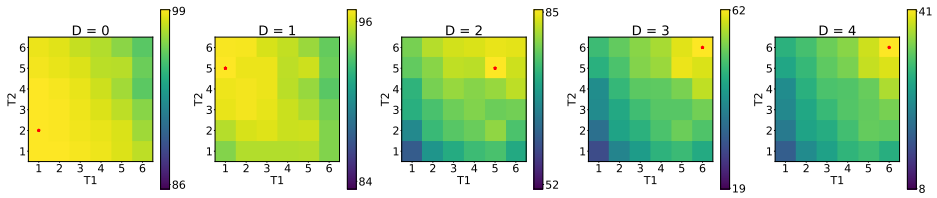
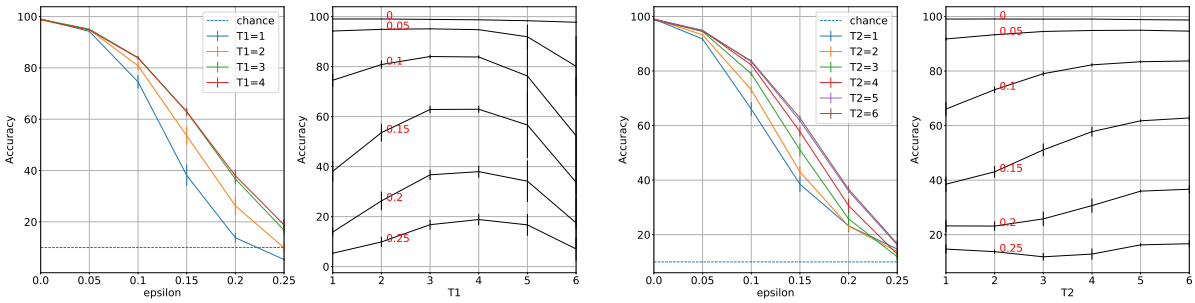


Figure 3.9 Classification accuracy (color-coded) for KerCNN for all combinations of $T_1, T_2 \in \{1, \dots, 6\}$, displayed for $D = 0, \dots, 4$. The maximum value of accuracy is marked by a red star onto the corresponding cell.

Adversarial attacks

Finally, we test our model's robustness to adversarial attacks via FGSM. Again, we first examine the performance of the models with lateral connections in one layer at a time, for varying T_1 and T_2 respectively. Figure 3.10 displays the classification accuracies of these



(a) (b)

Figure 3.10 Results for MNIST testing images perturbed via FGSM, for KerCNN with lateral connections in the first (a), resp. second layer (b). Left plots: classification accuracy at increasing values of ε, displayed for stopping time $T_1 = 1, \dots, 6$ (a), resp. $T_2 = 1, \dots, 6$ (b). Right plots: classification accuracy for increasing values of T_1 (a), resp. T_2 (b), displayed for different values of degradation. Each curve refers to a value of ε, displayed in red in correspondence of the curve.

models for $T_1 \in \{1, \dots, 6\}$ (a) and $T_2 \in \{1, \dots, 6\}$ (b). As before, the left figure plots the accuracy against the amount of degradation, with a curve for each stopping time T_i, while the right figure plots the accuracy against the stopping time T_i, with a curve for each value of ε. Finally, Figure 3.11 displays the analysis of the combinatorics of T_1 and T_2. Similarly to the case of Gaussian patches, the highest accuracy values lie on a diagonal. However, while in that case the optimal combination was clearly located around a single spot, two peaks

develop in the current case, corresponding to either high values of T_1 and low values of T_2, or viceversa.

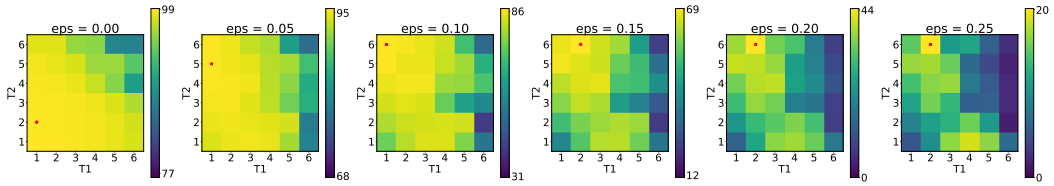


Figure 3.11 Classification accuracy (color-coded) for KerCNN for all combinations of $T_1, T_2 \in \{1,\dots,6\}$, displayed for $\varepsilon = 0,\dots,0.25$. The maximum value of accuracy is marked by a red star onto the corresponding cell.

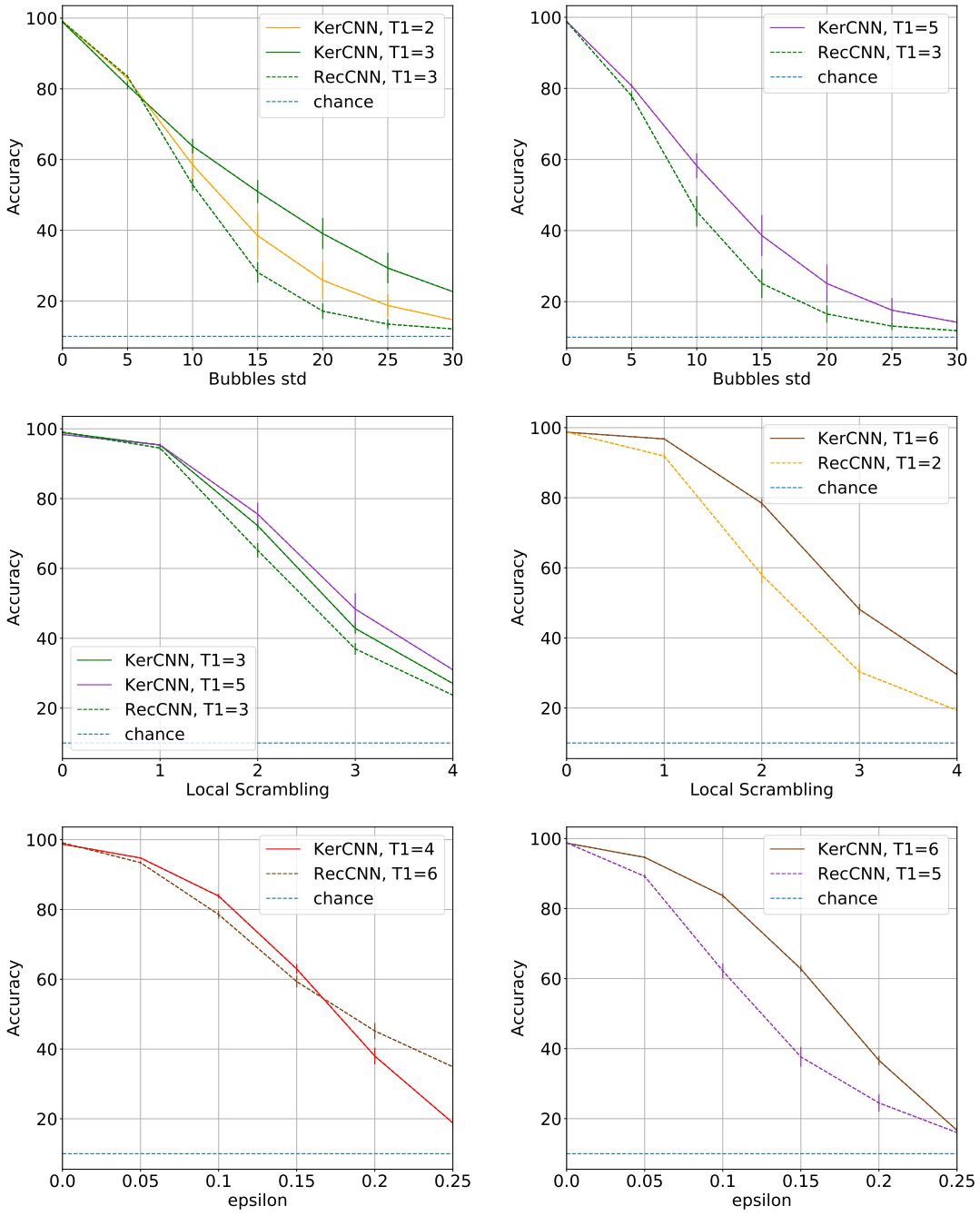
We summarize the main results of this section in Table 3.1. For each type of degradation and each level of corruption, the difference in mean % accuracy between the base CNN and the optimal KerCNN model is displayed, as well as the corresponding combination of stopping times (T_1, T_2).

Table 3.1 Overview on the best KerCNN performances resulting from the analysis of the combinatorics of stopping times (T_1, T_2), for the MNIST dataset.

			MNIST				
std γ	**0**	**5**	**10**	**15**	**20**	**25**	**30**
best (T_1, T_2)	(1,2)	(2,1)	(3,1)	(3,2)	(3,2)	(3,2)	(3,1)
% gap	+0.03%	+2.83%	+16.94%	**+26.74%**	+22.90%	+15.01%	+8.55%
Shift D	**0**	**1**	**2**	**3**	**4**		
best (T_1, T_2)	(1,2)	(1,5)	(5,5)	(6,6)	(6,6)		
% gap	+0.03%	+2.51%	+24.32%	**+33.29%**	+23.35%		
FGSM ε	**0**	**.05**	**.1**	**.15**	**.2**	**.25**	
best (T_1, T_2)	(1,2)	(1,5)	(1,6)	(2,6)	(2,6)	(2,6)	
% gap	+0.04%	+1.47%	+12.35%	**+31.08%**	+30.55%	+5.33%	

3.3.4 Comparison between KerCNNs and RecCNNs

We now compare our model with the RecCNN architectures described above. Here, recurrent convolutional connections as described in Section 3.1.3, with weights ϕ^l of size $4 \times 4 \times 16 \times 16$, have been added in the first (resp. second) layer; the size of the feedforward weights of the second layer has been decreased to $3 \times 3 \times 16 \times 16$ to make the number of



(a) (b)

Figure 3.12 Comparison between optimal KerCNN and optimal RecCNN. Top: Gaussian patches. (a) KerCNN with $T_1 = 2$ (orange, filled) and $T_1 = 3$ (green, filled); RecCNN with $T_1 = 3$ (green, dashed). (b) KerCNN with $T_2 = 5$ (violet, filled) and RecCNN with $T_2 = 3$ (green, dashed). Middle: local edge disruption. (a) KerCNN with $T_1 = 3$ (green, filled) and $T_1 = 5$ (violet, filled); RecCNN with $T_1 = 3$ (green, dashed). (b) KerCNN with $T_2 = 6$ (brown, filled) and RecCNN with $T_1 = 2$ (orange, dashed). Bottom: adversarial attacks. (a) KerCNN with $T_1 = 4$ (red, filled) and RecCNN with $T_1 = 6$ (brown, dashed). (b) KerCNN with $T_2 = 6$ (brown, filled) and RecCNN with $T_1 = 5$ (violet, dashed).

parameters match with the base CNN (as in [135]). The performance of these RecCNN models on the tasks examined before has been compared to the one of the base CNN, as well as with the corresponding KerCNNs. In most experiments, the RecCNN model did not reach better accuracies than the base CNN, although in some cases a pattern similar to the one seen for KerCNNs could be observed: in such cases, the performance increased until an optimal stopping time. However, the improvement in accuracy turned out to be much smaller in general than the one obtained by KerCNN models. Moreover, the geometric content of these learned lateral kernels is not evident and the iterative steps taken according to (3.17) do not seem to implement a kind of propagation – a hint of this lies in the fact that the optimal stopping time for RecCNNs never depends on the amount of degradation of the testing images.

In Figure 3.12, we compare the accuracies of the KerCNN and RecCNN architectures for the corresponding optimal stopping times for each task. In all plots, the filled curves refer to KerCNN models, while the accuracy of RecCNNs is displayed by dashed curves. The color of each curve matches the one used for the corresponding stopping time in all the plots throughout the paper.

Note that, in Figure 3.12a(top), curves for KerCNN with both $T_1 = 2$ and $T_1 = 3$ are displayed. Although the KerCNN model with stopping time $T_1 = 3$ (orange curve) widely outperforms the optimal RecCNN for all values of standard deviation above 10, the RecCNN displays a higher accuracy with small occlusions. However, for these smaller patches the optimal stopping time for KerCNN is $T_1 = 2$ (green curve), and this model outperforms the best RecCNN for *all values* of degradation. A similar situation can be observed in Figure 3.12a(middle) for local edge disruption, where both $T_1 = 3$ and $T_1 = 5$ curves are displayed for the KerCNN model.

To sum up, the KerCNN model clearly outperforms the corresponding RecCNN architecture, when comparing the two for their respective best stopping times, for almost all tasks examined. It is interesting to note that the only case in which RecCNNs show a higher accuracy than KerCNNs for some values of degradation (only for lateral connections in the first layer) is when the images are perturbed via FGSM for $\varepsilon > 0.2$. This suggests that, although the recurrent structure of RecCNNs may help improve the stability to "noise-like" perturbations, the absence of a geometric prior prevents them from implementing any mechanism of completion or contour integration. It is worth noting that, in the study carried out in [135], the networks were *trained and tested* to recognize cluttered digits: in their experiments, RecCNNs significantly outperform the purely convolutional architectures, thus showing the benefits of recurrence in *learning* challenging tasks. On the other hand, our study shows that this does not extend to the case where the networks are facing nuisances for which they were

not specifically optimized. For such generalization task, our structured lateral connections inducing a geometric prior turn out to be much more effective.

3.3.5 Other datasets

In this last section, we provide a synthetic report of our results on three different datasets. In order to analyze the effect of our lateral connections on images different from digits while keeping most of our settings unchanged, we first examined two MNIST-like datasets: the Kuzushiji-MNIST dataset [44], containing 10 phonetic letters of hiragana, one of the components of the Japanese writing system; and the Fashion-MNIST dataset [149], consisting of Zalando's article images subdivided into 10 item categories (T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot). Both datasets are made up of 70000

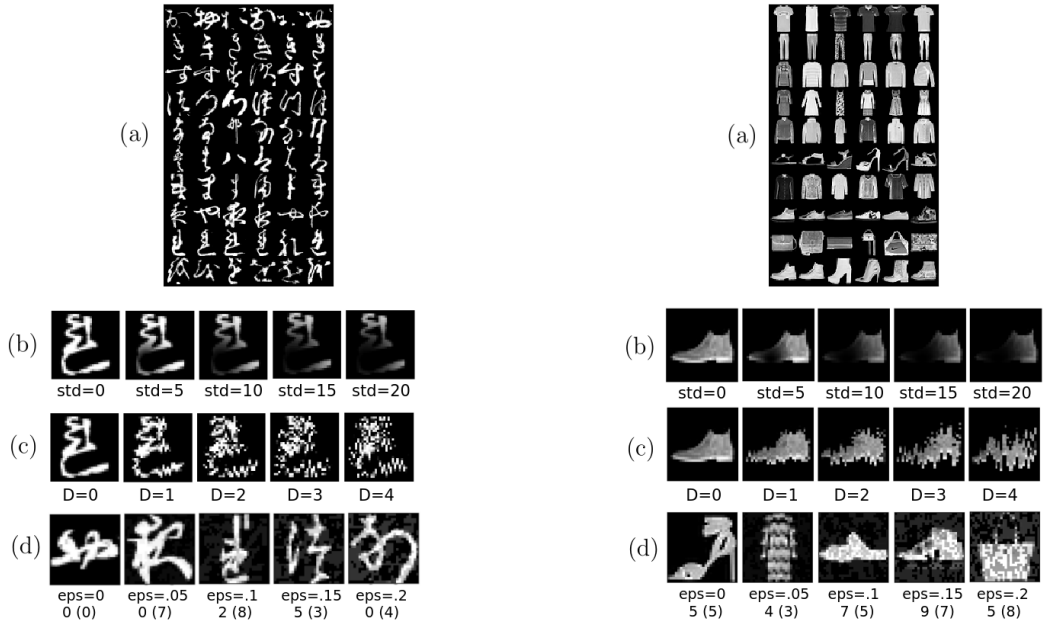


Figure 3.13 Examples from the Kuzushiji-MNIST (left) and Fashion-MNIST (right) datasets. (a) A sample from the dataset. Each row corresponds to a class. (b) A testing image corrupted by a Gaussian patch of increasing standard deviation. (c) A testing image corrupted by an increasing amount of local contour disruption D. (d) Testing images from different classes, perturbed by applying the FGSM to the base CNN with increasing values of ε*. Below each image, we display the classified label, as well as the correct label (in brackets). Apart from the unperturbed image (*$\varepsilon = 0$*), all the images are misclassified by the CNN.*

images of size 28×28, and we used the same training-validation-testing split as in MNIST. Figure 3.13 displays, for each of these two datasets, some representatives of their 10 classes, as well as some testing images corrupted by the three types of degradation examined. Finally,

we tested our model on the CIFAR-10 dataset [92], consisting of 60000 32×32 colour images in 10 classes (0:Airplane, 1:Automobile, 2:Bird, 3:Cat, 4:Deer, 5:Dog, 6:Frog, 7:Horse, 8:Ship, 9:Truck). Differently from MNIST-like datasets, CIFAR-10 poses the significantly harder problem of recognizing objects in natural scene images. The dataset includes 50000 training images and 10000 test images. We extracted 10000 images from the training set to use for validation-based early stopping – so that in our experiments the models were trained on 40000 samples, validated on 10000 samples and tested on 10000 samples. Figure 3.14

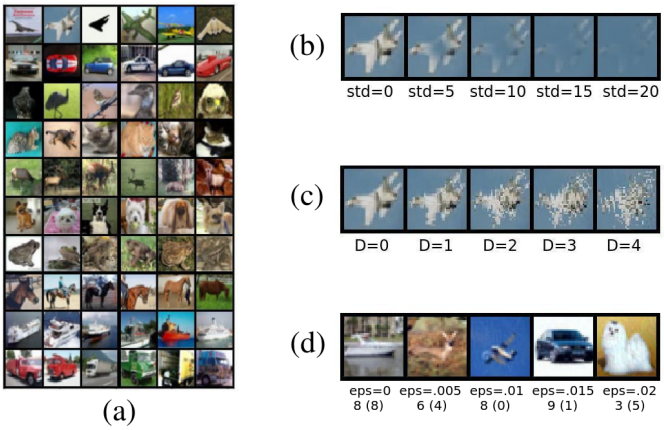


Figure 3.14 (a) A sample from the CIFAR-10 dataset. Each row corresponds to a class. (b) A testing image corrupted by a Gaussian patch of increasing standard deviation. (c) A testing image corrupted by an increasing amount of local contour disruption D. (d) Testing images from different classes, perturbed by applying the FGSM to the base CNN with increasing values of ε. Below each image, we display the classified label, as well as the correct label (in brackets). Apart from the unperturbed image ($\varepsilon = 0$), all the images are misclassified by the CNN.

shows some examples of (original as well as perturbed) testing images from CIFAR-10.

For all three datasets, we considered a CNN with 2 hidden layers as a base model.

- For what concerns Kuzushiji-MNIST and Fashion-MNIST, the architecture was kept the same as for MNIST, except for the number of filters of the second layer which was set to 32 instead of 16. The training options were kept the same as before, except for the L^2 regularization parameter for Kuzushiji-MNIST which was set to $\lambda = .001$.

- As for CIFAR-10, 64 and 128 filters were employed respectively in the first and second convolutional layers. Moreover, since the images are RGB, the filters of the first layer have three channels in this case. The networks were trained with early stopping for a maximum of 300 epochs. Stochastic gradient descent was employed with an initial learning rate of .01, which was automatically decreased by 1/10 when validation accuracy stopped increasing for 10 epochs. We used a batch size of 64 samples

and an L^2 regularization parameter $\lambda = .001$. Also, dropout with .5 probability was employed in the last layer. The rest of the settings were kept the same as for the other datasets. Due to the longer training times, the results displayed for each architecture are obtained by averaging over 3 networks, instead of 10, trained with different random seeds. Moreover, we let vary the stopping times T_i only in $\{1,2,3,4\}$.

Tables 3.2, 3.3 and 3.4 summarize the results obtained on these less easy and less overused databases. In Table 3.1, presented as a final report for MNIST, we displayed the best improvement in mean % accuracy w.r.t. the base CNN as an index of performance of our KerCNNs. We now summarize our results on these new datasets by reproposing a similar overview: for each degradation type and corruption degree, we display the mean % accuracies of the base CNN and the best KerCNN, as well as their difference.
For what concerns Kuzushiji-MNIST (Table 3.2), the best performance gap for images occluded by Gaussian patches is comparable to the one obtained for MNIST. However,

Table 3.2 Overview on the best KerCNN performances resulting from the analysis of the combinatorics of stopping times (T_1,T_2), for the Kuzushiji-MNIST dataset.

				Kuzushiji-MNIST			
std γ	**0**	**5**	**10**	**15**	**20**	**25**	**30**
best (T_1,T_2)	(1,2)	(2,1)	(3,3)	(3,3)	(3,3)	(3,3)	(3,3)
CNN	93.13%	74.20%	39.67%	21.22%	14.81%	36.79%	30.38%
KerCNN	93.13%	75.72%	59.96%	51.44%	43.69%	12.39%	11.41%
% gap	+0.00%	+1.53%	+20.29%	**+30.22%**	+28.89%	+24.40%	+18.97%
Shift D	**0**	**1**	**2**	**3**	**4**		
best (T_1,T_2)	(1,2)	(1,4)	(3,4)	(5,2)	(5,3)		
CNN	93.13%	85.06%	61.62%	42.15%	31.49%		
KerCNN	93.13%	87.91%	73.60%	59.15%	47.48%		
% gap	+0.00%	+2.85%	+11.99%	**+17.00%**	+16.00%		
FGSM ε	**0**	**.05**	**.1**	**.15**	**.2**	**.25**	
best (T_1,T_2)	(1,2)	(1,5)	(1,6)	(1,6)	(5,5)	(5,6)	
CNN	93.13%	65.03%	28.15%	11.28%	6.36%	3.95%	
KerCNN	93.13%	74.08%	48.91%	25.63%	13.74%	7.76%	
% gap	+0.00%	+9.05%	**+20.76%**	+14.35%	+7.38%	+3.81%	

a greater contribution of the second layer's kernel can be observed: that is, the optimal combinations of stopping times display larger values of T_2 for this type of degradation. This may be due to the more frequent occurrence of complex patterns requiring a "higher order" analysis (such as crossings and loops) w.r.t. MNIST. On the other hand, on images subject to

Table 3.3 Overview on the best KerCNN performances resulting from the analysis of the combinatorics of stopping times (T_1, T_2), for the Fashion-MNIST dataset.

				Fashion-MNIST			
std γ	**0**	**5**	**10**	**15**	**20**	**25**	**30**
best (T_1, T_2)	(1, 3)	(3, 1)	(3, 2)	(3, 2)	(3, 2)	(4, 4)	(4, 4)
CNN	89.86%	72.03%	49.07%	32.47%	22.43%	17.13%	14.18%
KerCNN	90.02%	73.37%	55.44%	43.03%	31.71%	25.55%	22.57%
% gap	+0.16%	+1.33%	+6.37%	**+10.55%**	+9.28%	+8.42%	+8.39%
Shift D	**0**	**1**	**2**	**3**	**4**		
best (T_1, T_2)	(1, 3)	(4, 3)	(5, 4)	(6, 6)	(6, 6)		
CNN	89.86%	77.27%	58.58%	44.33%	34.81%		
KerCNN	90.02%	83.69%	72.18%	66.43%	60.87%		
% gap	+0.16%	+6.42%	+13.61%	+22.10%	**+26.06%**		
FGSM ε	**0**	**.02**	**.04**	**.06**	**.08**	**.1**	
best (T_1, T_2)	(1, 3)	(2, 6)	(2, 6)	(2, 6)	(2, 6)	(2, 6)	
CNN	89.86%	53.81%	31.49%	18.53%	13.01%	10.13%	
KerCNN	90.02%	70.48%	54.83%	42.78%	32.84%	25.57%	
% gap	+0.16%	+16.67%	+23.34%	**+24.25%**	+19.82%	+15.45%	

local displacement, smaller overall values of T_1 and T_2 bring to the best accuracy, also leading to significantly smaller gaps in performance relative to MNIST. In fact, the abundance of small details in such characters makes this kind of perturbation far more disruptive than it is for images like MNIST's digits: even a small displacement may completely destroy some tiny yet characterizing features. Finally, the results for adversarial attacks with small values of ε are analogous to the ones obtained for digits, although with a faster decay in accuracy. On the other hand, although a configuration different from MNIST is observed for $\varepsilon \geq .2$, the accuracy values are around (or even below) chance level in these cases, which makes somewhat pointless to speculate about them.

Let us now examine the results obtained for the Fashion-MNIST dataset, displayed in Table 3.3. As for the images occluded by Gaussian patches, the slightly increased contribution of the second layer w.r.t. MNIST is again probably due to the heterogeneity of features characterizing these images, including both extended contours and tiny, intricate line patterns. For this type of perturbation, the improvement provided by our lateral connections is more moderate than it is for the preceding datasets, reaching a maximum accuracy gap of $\sim$10%. This may depend upon such images being largely composed by solid color areas rather than lines. Intuitively, when an occlusion falls in the middle of one such area, it does not interrupt a curve or a contour: therefore, the activation values of filters sensitive to local orientation is

very low at these locations and consequently the action of the kernel on them is less relevant. On the other hand, the perturbation obtained by shifting horizontal and vertical strips does not affect constant areas, while it consistently disrupts the image edges. Moreover, differently from Kuzushiji-MNIST's characters, global shapes rather than local details are markedly characterizing for discriminating between Fashion-MNIST classes. This makes our lateral connections particularly suited to manage this kind of perturbation. Indeed, a far greater

Table 3.4 Overview on the best KerCNN performances resulting from the analysis of the combinatorics of stopping times (T_1, T_2), for the CIFAR-10 dataset.

				CIFAR-10			
std γ	**0**	**5**	**10**	**15**	**20**	**25**	**30**
best (T_1, T_2)	(2, 1)	(2, 1)	(2, 1)	(2, 2)	(3, 2)	(3, 2)	(4, 1)
CNN	75.64%	58.22%	32.84%	22.89%	19.27%	17.97%	17.40%
KerCNN	75.57%	58.08%	32.90%	23.57%	20.53%	19.33%	18.89%
% gap	- 0.07%	- 0.14%	+0.06%	+0.67%	+1.26%	+1.36%	**+1.49%**
Shift D	**0**	**1**	**2**	**3**	**4**		
best (T_1, T_2)	(2, 1)	(4, 4)	(4, 4)	(4, 4)	(4, 4)		
CNN	75.64%	41.7%	27.70%	23.71%	21.91%		
KerCNN	75.57%	52.97%	43.33%	36.72%	31.99%		
% gap	- 0.07%	+11.27%	**+15.63%**	+13.02%	+10.08%		
FGSM ε	**0**	**.005**	**.01**	**.015**	**.02**	**.025**	
best (T_1, T_2)	(2, 1)	(2, 3)	(3, 4)	(4, 4)	(4, 4)	(4, 4)	
CNN	75.64%	42.9%	21.80%	10.83%	5.32%	2.86%	
KerCNN	75.57%	51.25%	35.55%	25.58%	18.66%	13.53%	
% gap	- 0.07%	+8.35%	+13.75%	**+14.75%**	+13.33%	+10.67%	

improvement from the CNN performance can be observed w.r.t. Kuzushiji-MNIST in this case, especially for large values of the displacement D: note that, for $D = 4$, the $\sim$35% accuracy obtained by the base CNN rises to $\sim$60% with the optimal KerCNN model. Finally, for what concerns adversarial attacks, we considered values of ε varying in a smaller range, since the decay in performance for this dataset turned out to be much faster; namely, we took $\varepsilon \in \{0, .02, .04, .06, .08, .1\}$. Again, up to this rescaling, the results are analogous to the other datasets.

As for CIFAR-10 (Table 3.4), the performance of CNNs and KerCNNs on images corrupted by Gaussian patches is comparable for all values of γ, with a slight advantage for KerCNNs for occlusions large enough ($\gamma > 5$). In our view, such "insensitivity" of lateral kernels to this type of perturbation may be linked to the increased difficulty of dealing with color images – indeed, this aspect certainly requires further investigation. On the other hand, the

improvement obtained by KerCNNs w.r.t. CNNs for images subject to edge disruption and adversarial attacks is still consistent (up to $\sim$15%). Note that the value of ε for adversarial attacks in this case was let vary in $\{0, .005, .01, .015, .02, .025\}$ (again due to the faster decay in accuracy w.r.t. ε).

Overall, we believe that the global results are very promising, both for what concerns the effectiveness of the model for image recognition under challenging conditions, and from the point of view of its interpretation linked to biological vision.

Conclusion

In this work, we introduced a novel technique for describing the properties of the horizontal connectivity in the primary visual cortex, by means of a metric structure which is directly induced by the geometry of feedforward connections. Our construction is very flexible, since it does not rely on any invariance in the parameterization of the family of profiles: this makes it possible to define the connectivity pattern associated to any bank of filters $\{\psi_p\}_p \subseteq L^2(\mathbb{R}^2)$. Our results show that various different properties consistent with neurophysiological and psychophysical data can indeed be recovered by applying this technique to different sets of profiles.
The flexibility of this construction leads to the hypothesis that a similar approach may also be employed for higher cortical layers, in order to give a geometric characterization of their connectivity.

We further proposed a new deep learning architecture obtained by inserting biologically plausible lateral connections in Convolutional Neural Networks, which turned out to improve their generalization ability in image classification tasks under challenging conditions: the testing images were subject to perturbations designed to undermine the ability of the networks to recognize the objects via a local feature extraction – thus requiring an integration of context information, which in biological vision is critically linked to lateral connectivity.
As a future development, we intend to extend our results to richer datasets and different tasks, and to examine the connectivity kernels obtained in these cases. Moreover, in connection with the hypothesis mentioned above, it would be interesting to acquire a more precise understanding of the different feature information encoded in the kernels associated to each layer of the convolutional architecture: this may help gain better insight into the analysis carried out by the networks at each stage of their processing.

Bibliography

[1] S. Abbasi-Sureshjani, M. Favali, G. Citti, A. Sarti, and B. M. ter Haar Romeny. Curvature integration in a 5d kernel for extracting vessel connections in retinal images. *IEEE Trans Image Process*, 27:606–621, 2018.

[2] A. A. Agrachev, D. Barilari, and U. Boscain. *A Comprehensive Introduction to sub-Riemannian Geometry*. Cambridge University Press, Cambridge, 2019.

[3] A. Ambrosetti and A. Malchiodi. *Nonlinear Analysis and Semilinear Elliptic Problems*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2007.

[4] L. Ambrosio. Calculus, heat flow and curvature-dimension bounds in metric measure spaces, 2018.

[5] L. Ambrosio and S. Masnou. A direct variational approach to a problem arising in image reconstruction. *Interfaces and Free Boundaries*, 5(1):63–81, 2003.

[6] A. Angelucci and J. Bullier. Reaching beyond the classical receptive field of V1 neurons: Horizontal or feedback axons? *Journal of Physiology Paris*, 97:141–154, 2003.

[7] A. Angelucci, J.B. Levitt, E. Walton, J. M. Hupé, J. Bullier, and J. S. Lund. Circuits for local and global signal integration in primary visual cortex. *J Neurosci*, 22:8633–8646, 2002.

[8] F. Anselmi and T. Poggio. Representation learning in sensory cortex: a theory. *CBMM memo*, 26, 2014.

[9] J.-P. Antoine and R. Murenzi. Two-dimensional directional wavelets and the scale-angle representation. *Signal Processing*, 52:241–272, 1996.

[10] N. Aronszajn. Theory of reproducing kernels. *Trans Amer Math Soc*, 66:937–404, 1950.

[11] P. Auer, M. Herbster, and M. K. Warmuth. Exponentially many local minima for single neurons. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 316–322. MIT Press, 1996.

[12] J. August and S. W. Zucker. The curve indicator random field: Curve organization via edge correlation. In K. Boyer and S. Sarkar, editors, *Perceptual Organization for Artificial Vision Systems*, volume 546 of *The Kluwer International Series in Engineering and Computer Science*. Springer, Boston, MA, 2000.

[13] N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman. Deep convolutional networks do not classify based on global object shape. *PLOS Computational Biology*, 14:1–43, 2018.

[14] Radu Balan, Maneesh Singh, and Dongmian Zou. *Lipschitz properties for deep convolutional networks*, volume 706 of *Contemporary Mathematics*, pages 129–151. American Mathematical Society, United States, 2018.

[15] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 2001.

[16] D. Barbieri, G. Citti, G. Sanguinetti, and A. Sarti. An uncertainty principle underlying the functional architecture of V1. *Journal of Physiology Paris*, 106:183–193, 2014.

[17] D. Barbieri, G. Cocci, G. Citti, and A. Sarti. A cortical-inspired geometry for contour perception and motion integration. *J Math Imaging Vis*, 49(3):511–529, 2014.

[18] E. Baspinar, G. Citti, and A. Sarti. A geometric model of multi-scale orientation preference maps via gabor functions. *Journal of Mathematical Imaging and Vision*, 60:900–912, 2017.

[19] O. Ben-Shahar, P. Huggins, T. Izo, and S.W. Zucker. Cortical connections and early visual function: intra- and inter-columnar processing. *J Physiol Paris*, 97:191–208, 2003.

[20] O. Ben-Shahar and S. Zucker. Geometrical computations explain projection patterns of long-range horizontal connections in visual cortex. *Neural Computation*, 16:445–476, 2004.

[21] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*,

SIGGRAPH '00, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[22] M. Bertalmío and J. D. Cowan. Implementing the retinex algorithm with Wilson–Cowan equations. *Journal of Physiology-Paris*, 103(1):69–72, 2009. Neuromathematics of Vision.

[23] A. Bietti and J. Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *Journal of Machine Learning Research*, 20(25):1–49, 2019.

[24] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[25] T. Bonhoeffer and A. Grinvald. Iso-orientation domains in cat visual cortex are arranged in pinwheel-like patterns. *Nature*, 353(6343):429–431, 1991.

[26] U. Boscain, R. Chertovskih, J.P. Gauthier, and A. Remizov. Hypoelliptic diffusion and human vision: a semi-discrete new twist. *SIAM Journal on Imaging Sciences*, 7:669–695, 2014.

[27] W. Bosking, Y. Zhang, B. Schoenfield, and D. Fitzpatrick. Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex. *J Neurosci*, 17:2112–2127, 1997.

[28] L. Bottou. On-line learning in neural networks. chapter On-line Learning and Stochastic Approximations, pages 9–42. Cambridge University Press, New York, NY, USA, 1998.

[29] W. Brendel and M. Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2019.

[30] P. C. Bressloff and J. D. Cowan. The functional geometry of local and long-range connections in a model of V1. *J Physiol Paris*, 97(2-3):221–236, 2003.

[31] P.C. Bressloff, J.D. Cowan, M. Golubitsky, P.J. Thomas, and M.C. Wiener. What geometric visual hallucinations tell us about the visual cortex. *Neural Computation*, 14:473–491, 2002.

[32] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, 2013.

[33] D. Burago, Y. Burago, and S. Ivanov. *A course in metric geometry*, volume 33 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2001.

[34] D. Burago, S. Ivanov, and Y. Kurylev. A graph discretization of the Laplace-Beltrami operator. *J. Spectr. Theory*, 4(4):675–714, 2014.

[35] J. Cheeger. Differentiability of lipschitz functions on metric measure spaces. *Geom. Funct. Anal.*, 9(3):428–517, 1999.

[36] J.-H. Cheng, J.-F. Hwang, A. Malchiodi, and P. Yang. Minimal surfaces in pseudohermitian geometry. *Ann. Sc. Norm. Super. Pisa Cl. Sci. (5)*, 4(1):129–177, 2005.

[37] A. Choromanska, M. Henaff, M. Mathieu, B. Gerard, and Y. LeCun. The loss surfaces of multilayer networks. *Journal of Machine Learning Research*, 38:192–204, 2015.

[38] W.-L. Chow. Über Systeme von linearen partiellen Differentialgleichungen erster Ordnung. *Math. Ann.*, 117:98–105, 1939.

[39] G. Citti and A. Sarti (eds.). *Neuromathematics of Vision*. Lecture Notes in Morphogenesis. Springer, 2014.

[40] G. Citti and M. Manfredini. Implicit function theorem in Carnot-Carathéodory spaces. *Commun. Contemp. Math.*, 8(5):657–680, 2006.

[41] G. Citti and A. Sarti. A cortical based model of perceptual completion in the roto-translation space. *Journal of Mathematical Imaging and Vision archive*, 24:307–326, 2006.

[42] G. Citti and A. Sarti. A gauge field model of modal completion. *Journal of Mathematical Imaging and Vision*, 52:267–284, 2013.

[43] G. Citti and A. Sarti. Models of the visual cortex in lie groups. In *Harmonic and Geometric Analysis*, Advanced Courses in Mathematics–CRM Barcelona. Birkhäuser, Basel, 2015.

[44] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep learning for classical japanese literature. In *Workshop on Machine Learning for Creativity and Design*. NIPS, 2018.

[45] G. Cocci, D. Barbieri, G. Citti, and A. Sarti. Cortical spatiotemporal dimensionality reduction for visual grouping. *Neural Comput.*, 27(6):1252–1293, 2015.

[46] G. Cocci, D. Barbieri, and A. Sarti. Spatiotemporal receptive fields of cells in V1 are optimally shaped for stimulus velocity estimation. *J Opt Soc Am A*, 29, 2012.

[47] R. R. Coifman and S. Lafon. Diffusion maps. *Appl Comput Harmon Anal*, 21, 2006.

[48] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J Opt Soc Am A2*, 2:1160–1169, 1985.

[49] D.Barbieri, G.Citti, and A.Sarti. How uncertainty bounds the shape index of simple cells. *The Journal of Mathematical Neuroscience*, 4(5), 2014.

[50] C.-X. Deng, S. Li, and Z.-X. Fu. The reproducing kernel hilbert space based on wavelet transform. In *Proceedings of the 2010 International Conference on Wavelet Analysis and Pattern Recognition, Qingdao*, pages 370–374, 2010.

[51] A. Dobbins, S. Zucker, and M. Cynader. Endstopped neurons in the visual cortex as a substrate for calculating curvature. *Nature*, 329:438–441, 1987.

[52] R. Duits. *Perceptual Organization in Image Analysis: A mathematical approach based on scale, orientation and curvature*. PhD thesis, Eindhoven University of Technology, 2005. Phd thesis.

[53] R. Duits, U. Boscain, F. Rossi, and Y. Sachkov. Association fields via cuspless sub-riemannian geodesics in se(2). *Journal of Mathematical Imaging and Vision*, 49:384–417, 2014.

[54] R. Duits, M. Felsberg, G. Granlund, and B.M. ter Haar Romeny. Image analysis and reconstruction using a wavelet transform constructed from a reducible representation of the euclidean motion group. *International Journal of Computer Vision*, 72:79–102, 2007.

[55] R. Duits and E.M. Franken. Left-invariant parabolic evolutions on se(2) and contour enhancement via invertible orientation scores, part i: Linear left-invariant diffusion equations on se(2). *Quarterly of Applied Mathematics*, 68:293–331, 2010.

[56] R. Duits and E.M. Franken. Left-invariant parabolic evolutions on se(2) and contour enhancement via invertible orientation scores, part ii: Nonlinear left-invariant diffusions on invertible orientation scores. *Quarterly of Applied Mathematics*, 68:255–292, 2010.

[57] R. Duits, H. Führ, B. Janssen, M. Bruurmijn, L. Florack, and H. van Assen. Evolution equations on gabor transforms and their applications. *Applied and Computational Harmonic Analysis*, 35:483–526, 2013.

[58] J. H. Elder and R. M. Goldberg. Ecological statistics of gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2:324–353, 2002.

[59] M. Favali, G. Citti, and A. Sarti. Local and global gestalt laws: A neurally based spectral approach. *Neural Computation*, 29:394–422, 2017.

[60] H. Federer. *Geometric Measure Theory*. Springer-Verlag, 1969.

[61] D. J. Field, A. Hayes, and R. F. Hess. Contour integration by the human visual system: evidence for a local association field. *Vision Res*, 33:173–193, 1993.

[62] B. Franchi, R. Serapioni, and F. Serra Cassano. Rectifiability and perimeter in the Heisenberg group. *Math. Ann.*, 321(3):479–531, 2001.

[63] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[64] M. Fukushima, Y. Oshima, and M. Takeda. *Dirichlet forms and symmetric Markov processes*, volume 19 of *De Gruyter Studies in Mathematics*. Walter de Gruyter & Co., Berlin, extended edition, 2011.

[65] M. Galli and M. Ritoré. Existence of isoperimetric regions in contact sub-Riemannian manifolds. *J. Math. Anal. Appl.*, 397(2):697–714, 2013.

[66] N. Garofalo and D.-M. Nhieu. Isoperimetric and Sobolev inequalities for Carnot-Carathéodory spaces and the existence of minimal surfaces. *Comm. Pure Appl. Math.*, 49(10):1081–1144, 1996.

[67] W. S. Geisler, J. S. Perry, B. J. Super, and D. P. Gallogly. Edge co-occurrence in natural images predicts contour grouping performance. *Vision Research*, 41:711–724, 2001.

[68] T. Georgiev. Covariant derivatives and vision. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, pages 56–69, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[69] Z. Gigus and J. Malik. Detecting curvilinear structure in images. Technical report, Berkeley, CA, USA, 1991.

[70] C. D. Gilbert. Laminar differences in receptive field properties of cells in cat primary visual cortex. *J Physiol*, 268:391–421, 1977.

[71] C. D. Gilbert, A. Das, M. Ito, M. Kapadia, and G. Westheimer. Spatial integration and cortical dynamics. In *Proceedings of the National Academy of Sciences USA*, volume 93, pages 615–622, 1996.

[72] C. D. Gilbert and T. N. Wiesel. Morphology and intracortical projections of functionally identified neurons in cat visual cortex. *Nature*, 280:120–125, 1979.

[73] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.

[74] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the ICLR*, 2015.

[75] M. Gromov. Carnot-Carathéodory spaces seen from within. In *Sub-Riemannian geometry*, volume 144 of *Progr. Math.*, pages 79–323. Birkhäuser, Basel, 1996.

[76] S. Grossberg and E. Mingolla. Neural dynamics of perceptual grouping: Textures, boundaries, and emergent segmentations. *Perception & Psychophysics*, 38:141–171, 1985.

[77] T. Hansen and H. Neumann. A recurrent model of contour integration in primary visual cortex. *J of Vision*, 8:1–25, 2008.

[78] F. Hausdorff. Dimension und äusseres mass. *Mathematische Annalen*, 79:157–179, 1918.

[79] W.C. Hoffman. The visual cortex is a contact bundle. *Appl Math Comput*, 32:137–167, 1989.

[80] L. Hörmander. Hypoelliptic second order differential equations. *Acta Math.*, 119:147–171, 1967.

[81] D. H. Hubel. *Eye, brain, and vision*. New York, WH Freeman (Scientific American Library), 1987.

[82] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat visual cortex. *J Physiol (London)*, 160:106–154, 1962.

[83] R. Ivry J. Beck, A. Rosenfeld. Line segregation. *Spatial Vision*, 4:75–101, 1989.

[84] J. P. Jones and L. A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *J Neurophysiol*, 58:1233–1258, 1987.

[85] M. K. Kapadia, G. Westheimer, and C. D. Gilbert. Dynamics of spatial summation in primary visual cortex of alert monkeys. In *Proc Natl Acad Sci USA*, volume 96, pages 12073–12078, 1999.

[86] K. Kawaguchi. Deep learning without poor local minima. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 586–594. Curran Associates, Inc., 2016.

[87] D. P. Kingma and J. L. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, pages 1–13, 2015.

[88] Z. F. Kisvarday and U. T. Eysel. Cellular organization of reciprocal patchy networks in layer iii of cat visual cortex (area 17). *Neuroscience*, 46:275–286, 1992.

[89] J. J. Koenderink and A. J. van Doom. Representation of local geometry in the visual system. *Biol. Cybern.*, 55(6):367–375, 1987.

[90] K. Koffka. *Principles of Gestalt Psychology*. New York: Harcourt, Brace, 1935.

[91] W. Köhler. *Gestalt Psychology*. New York: H. Liveright, 1929.

[92] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[93] A. Krizhevsky, I. Sutskever, and G E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[94] N. Kruger. Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8:117–129, 1998.

[95] E. H. Land and J. J. McCann. Lightness and retinex theory. *J. Opt. Soc. Am.*, 61(1):1–11, 1971.

[96] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput*, 1(4):541–551, 1989.

[97] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[98] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms, proceedings of the 19th annual conference on neural information processing systems. *Cambridge MA: MIT Press*, pages 801–808, 2007.

[99] T. S. Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18, 1996.

[100] B. Levy. Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13, 2006.

[101] M. Liang and X. Hu. Recurrent convolutional neural network for object recognition. *CVPR*, 2015.

[102] J. Lott and C. Villani. Ricci curvature for metric-measure spaces via optimal transport. *Ann. of Math. (2)*, 169(3):903–991, 2009.

[103] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks. In *NIPS*, 2014.

[104] S. Mallat. Group invariant scattering. *Comm. Pure Appl. Math.*, 65(10):1331–1398, 2012.

[105] L. M. Martinez and J.-M. Alonso. Complex receptive fields in primary visual cortex. *Neuroscientist*, 9:317–331, 2003.

[106] G. Mitchison and F. Crick. Long axons within the striate cortex: their distribution, orientation, and patterns of connection. In *Proceedings of National Academy of Sciences USA*, volume 79, pages 3661–3665, 1982.

[107] R. Montgomery. *A tour of subriemannian geometries, their geodesics and applications*, volume 91 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2002.

[108] N. Montobbio, L. Bonnasse-Gahot, G. Citti, and A. Sarti. KerCNNs: biologically inspired lateral connections for classification of corrupted images. *submitted*, 2019.

[109] N. Montobbio, G. Citti, and A. Sarti. From receptive profiles to a metric model of V1. *J Comput Neurosci*, 46(3):257–277, 2019.

[110] N. Montobbio, A. Sarti, and G. Citti. A metric model for the functional architecture of the visual cortex. *under revision*, 2019.

[111] N. Morgan and H.Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 630–637. Morgan-Kaufmann, 1990.

[112] D. Mumford. In C. Bajaj, editor, *Elastica and computer vision*, pages 507–518. Springer-Verlag, 1993.

[113] A. Nagel, E. Stein, and S. Wainger. Balls and metrics defined by vector fields. I. Basic properties. *Acta Math.*, 155(1-2):103–147, 1985.

[114] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[115] H. Neumann and E. Mingolla. Computational neural models of spatial integration in perceptual grouping. In T. F. Shipley and P. J. Kellman, editors, *Advances in psychology*, volume 130, chapter From fragments to objects: Segmentation and grouping in vision, pages 353–400. 2001.

[116] S. Ohta. On the measure contraction property of metric measure spaces. *Comment. Math. Helv.*, 82(4):805–828, 2007.

[117] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[118] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[119] A. Peters and E. G. Jones. Primary visual cortex in primates (volume 10). In *Cerebral Cortex*, Science+Business Media New York. Springer, 1994.

[120] J. Petitot. *Neurogéométrie de la vision - Modèles mathématiques et physiques des architectures fonctionnelles*. Éditions de l'École Polytechnique, 2008.

[121] J. Petitot and Y. Tondut. Vers une neuro-géométrie. fibrations corticales, structures de contact et contours subjectifs modaux. In *Mathématiques , Informatique et Sciences Humaines*, volume 145, pages 5–101. CAMS, EHESS, 1999.

[122] L. Prechelt. Early stopping - but when? In *Neural Networks: Tricks of the Trade, volume 1524 of LNCS, chapter 2*, pages 55–69. Springer-Verlag, 1997.

[123] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo. Discrete laplace-beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009.

[124] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[125] K. S. Rockland and J. S. Lund. Intrinsic laminar lattice connections in primate visual cortex. *Journal of Comparative Neurology*, 216:303–318, 1983.

[126] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747*, 2016.

[127] G. Sanguinetti, G. Citti, and A. Sarti. A model of natural image edge co-occurrence in the rototranslation group. *J Vis*, 10, 2010.

[128] A. Sarti and G. Citti. The constitution of visual perceptual units in the functional architecture of V1. *Journal of Computational Neuroscience*, 38:285–300, 2014.

[129] A. Sarti and G. Citti. The constitution of visual perceptual units in the functional architecture of V1. *Journal of Computational Neuroscience*, 38:285–300, 2015.

[130] A. Sarti, G. Citti, and J. Petitot. The symplectic structure of the primary visual cortex. *Biol. Cybern.*, 98(1):33–48, 2008.

[131] A. Sarti, G. Citti, and J. Petitot. The symplectic structure of the visual cortex. *Biological Cybernetics*, 98:33–48, 2008.

[132] Alessandro Sarti, Giovanna Citti, and Jean Petitot. Functional geometry of the horizontal connectivity in the primary visual cortex. *Journal of Physiology-Paris*, 103(1):37–45, 2009. Neuromathematics of Vision.

[133] S. Semeniuta, A. Severyn, and E. Barth. Recurrent dropout without memory loss. In *Proceedings of COLING 2016, the 26th International Conference on Computational*

Linguistics: Technical Papers, pages 1757–1766, Osaka, Japan, 2016. The COLING 2016 Organizing Committee.

[134] M. Sigman, G.A. Cecchi, C. D. Gilbert, and M. O. Magnasco. On a common circle: Natural scenes and gestalt rules. In *Proceedings of the National Academy of Sciences*, volume 98, pages 1935–1940, 2001.

[135] C.J. Spoerer, P. McClure, and N. Kriegeskorte. Recurrent convolutional neural networks: a better model of biological object recognition. *Frontiers in psychology*, 8:1551, 2017.

[136] K.-T. Sturm. On the geometry defined by dirichlet forms. In E. Bolthausen et al., editor, *Seminar on Stochastic Analysis, Random Fields and Applications*, pages 231–242. Birkhäuser, Boston, 1995.

[137] K.-T. Sturm. Diffusion processes and heat kernels on metric spaces. *Ann Probab*, 26:1–55, 1998.

[138] K.-T. Sturm. On the geometry of metric measure spaces. ii. *Acta Math.*, 196(1):133–177, 2006.

[139] H. Tang, M. Schrimpf, W. Lotter, C. Moerman, A. Paredes, J. O. Caro, W. Hardesty, D. Cox, and G. Kreiman. Recurrent computations for visual pattern completion. *PNAS*, 115:8835–8840, 2018.

[140] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the royal statistical society, series B*, 58:267–288, 1994.

[141] R. B. Tootell, E. Switkes, M. S. Silverman, and S. L. Hamilton. Functional anatomy of macaque striate cortex. ii. retinotopic organization. *The Journal of Neuroscience*, 8(5):1531–1568, 1988.

[142] A. Virmaux and K. Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3835–3844. Curran Associates, Inc., 2018.

[143] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt. A century of gestalt psychology in visual perception: I. perceptual grouping and figure-ground organization. *Psychological Bulletin*, 138(6):1172–1217, 2012.

[144] B. Wandell. *Foundations of Vision: Behavior, Neuroscience and Computation*. Sinauer Associates Inc., 1995.

[145] F. W. Warner. *Foundations of differentiable manifolds and Lie groups*, volume 94 of *Graduate Texts in Mathematics*. Springer-Verlag, New York-Berlin, 1983. Corrected reprint of the 1971 edition.

[146] P. J. Werbos. *Beyond regression: new tools for prediction and analysis in the behavioural sciences*. PhD thesis, Harvard University, Boston, MA, 1974.

[147] M. Wertheimer. Laws of organization in perceptual forms. *Psycologische Forschung*, 4:301–350, 1923.

[148] W.A. Wilson. On quasi-metric spaces. *American Journal of Mathematics*, 53:675, 1931.

[149] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv:cs.LG/1708.07747*, 2017.

[150] J. Yeh. *Real analysis*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, third edition, 2014. Theory of measure and integration.

[151] R. Yeh, M. Hasegawa-Johnson, and M. N. Do. Stable and symmetric filter convolutional neural network. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2652–2656, 2016.

[152] S. C. Yen and L. H. Finkel. Extraction of perceptually salient contours by striate cortical networks. *Vision Res*, 38:719–741, 1998.

[153] D. Zou, R. Balan, and M. K. Singh. On lipschitz bounds of general convolutional neural networks. *CoRR*, abs/1808.01415, 2018.

[154] S.W. Zucker. Differential geometry from the frenet point of view: boundary detection, stereo, texture and color. In O. D. Faugeras N. Paragios, Y. Chen, editor, *Handbook of Mathematical Models in Computer Vision*, pages 357–373. Springer, US, 2006.