Field Computation: A Theoretical Framework for Massively Parallel Analog Computation Parts I – IV

Bruce MacLennan

Department of Computer Science
University of Tennessee

CS-90-100
February 1990

Revised

December 4, 1991

Abstract

This report comprises the first four parts of a systematic presentation of *field computation*, a theoretical framework for understanding and designing massively parallel analog computers. This theory treats computation as the continuous transformation of *fields*: continuous ensembles of continuous-valued data. This theory is technology-independent in that it can be realized through optical and molecular processes, as well as through large neural networks.

Part I is an overview of the goals and assumptions of field computation. Part II presents relevant ideas and results from functional analysis, including theorems concerning the field-computation of linear and multilinear operators. Part III is devoted to examples of the field computation of a number of useful linear and multilinear operators, including integrals, derivatives, Fourier transforms, convolutions and correlations. Part IV discusses the field computation of nonlinear operators, including: a theoretical basis for universal (general purpose) field computers, ways of replacing field polynomials by sigmoid transformations, and ways of avoiding higher-dimensional fields (since they may be difficult to represent in physical media).

Contents

Part I
Field Computation

1 Introduction

1.1 Truly Massive Parallelism

AI is moving into a new phase characterized by a broadened understanding of the nature of knowledge, and by the use of new computational paradigms.[1] A sign of this transition is the growing interest in neurocomputers, optical computers, molecular computers and other massively parallel analog computers. We have argued elsewhere [5, 6, 7, 8] that the new AI will augment the traditional deep, narrow computation with shallow, wide computation. That is, the new AI will exploit *massive parallelism*, but this means different things to different people; massive parallelism may begin with a hundred, a thousand, or a million processors. Biological evidence suggests that skillful behavior requires a very large number of processors, so many in fact that it is infeasible to treat them individually; they must be treated *en masse.* This has motivated us to propose [5] the following definition of massive parallelism: *A computational system is* **massively parallel** *if the number of processing elements is so large that it may conveniently be considered a continuous quantity.* That is, a system is *massively* parallel if the processing elements can be considered a *continuous mass* rather than a *discrete ensemble* [9].

How large a number is large enough to be considered a continuous quantity? That depends on the purpose at hand. A hundred is probably never large enough; a million is probably always large enough; a thousand or ten thousand may be enough. One of the determining factors will be whether the number is large enough to permit the application of continuous mathematics (see below).

We propose this definition of massive parallelism for a number of reasons. First, skillful behavior seems to require significant neural *mass.*[2] Second, we are interested in computers, such as optical computers and molecular computers, for which the number of processing elements *is* effectively continuous. Third, continuous mathematics is generally easier than discrete mathematics. And fourth, we want to encourage a new style of thinking about parallelism. Currently, we try to apply to parallel machines the thought habits we have acquired from thinking about sequential machines. This strategy works fairly well when the degree of parallelism is low, but it will not scale up. One cannot think individually about the 10^{20} processors of a molecular computer. Rather than postpone the inevitable, we think that it is time to develop a theoretical

[1]Research reported herein has been supported in part by Faculty Research Awards (1988, 1989) from the University of Tennessee, Knoxville, and in part by the Mathematical Sciences Section of Oak Ridge National Laboratory.

[2]Even a bee has some 10^6 neurons [2, p. 33].

framework for understanding massively parallel analog computers. The principal goal of this paper is to outline such a theory.

1.2 Field Transformation

Our aim then is to develop a way of looking at massive parallelism that encompasses a variety of implementation technologies, including neural networks, optical computers, molecular computers and other massively parallel analog computers. What these all have in common is the ability to process in parallel amounts of data so massive as to be considered a continuous quantity. This suggests that we structure our theory around the idea of a *field,* i.e. a continuous (dense) ensemble of data. We have in mind both scalar fields (such as potential fields) and vector fields (such as gradient fields). Any operation on such a field, either to produce another field or to produce a new state of the field, can be considered massively parallel, if it operates on all the elements of the field in parallel. Indeed, it would not be feasible to serialize the processing of the field; modest degrees of parallelism cannot cope with an infinite (or nearly infinite) number of field elements.

In the remainder of this paper we explore *field transformation computers,* that is, computers characterized by the ability to perform (in parallel) transformations on scalar and vector fields. This does not mean that field computers are unable to perform scalar calculations; in fact many field transformation computers have the scalar capabilities of conventional digital and analog computers. Scalars have many uses in field computation. For example, we may want to use a scalar parameter to control the rate at which a field transformation takes place (e.g., a reaction rate in a molecular computer). Similarly, we may use a scalar representing the average intensity of a field to control the contrast enhancement of that field. A scalar threshold value may be used to suppress low level noise, and so forth.

It must be stressed that there are many field computers already in existence, for example, large neurocomputers and many optical computers. What we are proposing is:

1. a name for the class of such computers
2. a theoretical framework for understanding massively parallel analog computers
3. a basis for constructing general purpose computers of this type.

1.3 Classes of Field Transformations

Field transformations, like filters, can be divided into two classes: *nonrecursive* and *recursive.* A nonrecursive transformation is simply a functional composition of more elementary transformations. The output of a nonrecursive transformation depends only on its input. A recursive transformation involves some kind of feedback. Hence, its output depends both on its input and on its prior state. Recursive transformations are ideal for simulating the temporal behavior of physical systems, as in simulated annealing, genetic algorithms and Boltzmann machines [9].

1.4 General Purpose Field Computers

Field computers in the past have been designed for special purposes, and we expect it to be the case in the future. In these computers, devices implementing field transformations (such as filters and convolutions) are assembled to solve a small class of problems (e.g., edge detection). On the other hand, our experience with digital computation has shown the value of *general purpose* or *programmable* computers. This architectural feature permits one computer to perform a variety of digital computations, which often eliminates the need to construct special purpose devices, and often speeds implementation of digital algorithms.

The foregoing observations suggest that general purpose *field* computers will be similarly valuable. In these the connections between field transformation units and field storage units are programmable, thus facilitating their reconnection for a variety of purposes. In fact, we may want to make better use of our resources by *multiplexing* the use of field transformation units under the control of a program. Thus, a program for a general purpose field computer might look very much like a conventional program, except that the basic operations are field transformations rather than scalar arithmetic.

We cannot build into a general purpose field computer every transformation we might need. Instead we must choose a set of primitive operations that permit the programming of most others. How can such a set of primitive operations be chosen? How can we be guaranteed that we have provided all the necessary facilities, at least for a given class of problems? For digital computers this question is answered in part by computability theory. For example, this theory shows us how to construct a *universal Turing machine,* which, given an appropriate program, can emulate any Turing machine. Although the universal Turing machine is hardly a practical general purpose computer, consideration of it and other universal machines shows us the kinds of facilities a computer must have in order to be universal. There follows the hard engineering job of going from the theoretically sufficient architecture to the practically necessary architecture.

Can the same be accomplished for field computers? Is there a *universal field computer* that can emulate any field computer? If there is such a thing, then we can expect that it may form a basis for practical general purpose field computers in much the same way that Turing machines do for digital computers. Fortunately there is a set of primitive operations that can implement any field transformation in a very wide class. We briefly discuss the results below; the details can be found in Section 9.

1.5 Universal Machines and Finite Resources

Later (Section 9), we show that with a certain set of built-in field transformations we can implement (within a stated limit) any field transformation in a very wide class. This is analogous to the result from Turing machine theory: The universal Turing machine allows us to implement (within a stated limit) any function in a wide class (now known as the *Turing computable functions*).

The phrase 'within a stated limit' appears in both of the preceding statements. What does it mean? For the Turing machine it means that a stated limit (e.g., precision or range of argument) can be achieved by providing a long enough tape. For the digital computer it means that computations are normally performed to a given precision (e.g., the word length), and that finite increments in the desired precision require finite increments in the resources required (e.g., additional registers and memory cells for double and multiple precision results, or stack space for recursion). The case is much the same for the universal field computer. Finite increments in the desired accuracy of a field transformation will require finite increments in the resources used (such as field transformation and storage units).

There are a number of theoretical bases for a universal field computer. We have investigated designs based on Fourier analysis, interpolation theory and Taylor's theorem, all generalized for field transformations [5, 6]. In this paper we present the basis for a design based on Taylor's theorem as well as a design based on a functional analog of polynomial approximation. There are no doubt as many principles upon which universal field computers can be based as there are bases for universal digital computers. Engineering considerations must determine which of these models can form a feasible basis for practical general purpose field computers.

2 Assumptions

2.1 Fields

2.1.1 Assumed Properties of Fields

Physical Realizability By *physical realizability* we mean that all fields must ultimately be represented in some physical medium (e.g., electrical potential, light intensity, chemical concentration). This places certain constraints on the fields with which we must deal. For example, fields must occupy a finite amount of space, otherwise they will not fit in our field computers. Second, the dynamic range of the fields' values are limited; concentrations, intensities, and so forth cannot be arbitrarily large. Third, a field's values vary continuously with their location in the field. Physical media will not support an infinite gradient, that is, a discontinuous change in value.

As we will see, these physical constraints translate into mathematical properties that helpfully limit the class of fields with with we must deal. On the other hand, the analysis is sometimes simplified by assuming the existence of unrealizable fields (e.g., Dirac deltas). In these cases we must be careful that the results of such analysis apply to realizable fields (see Sections 6.2.4 and 6.2.6 for examples).

Fields are Functions We treat fields as functions ϕ from a domain Ω to a range K, that is, $\phi : \Omega \to K$. Thus, at each point $t \in \Omega$ the field ϕ has a value $\phi(t) \in K$, which we will often write ϕ_t. The domain Ω is a metric space. The range K is a subset of an algebraic field. In this paper it will always be some closed interval of the real numbers, although fields whose values are bounded subsets of the complex numbers might be useful in some applications.

In the remainder of this section we discuss constraints on the allowable domains Ω, on the ranges K, and on the functions $\phi : \Omega \to K$. Our goal will be to define $\Phi_K(\Omega)$, the space of all K-valued fields over a domain Ω.[3]

Fields Belong to Linear Spaces Fields belong to *linear spaces,* which means that we can define an addition $\phi + \psi$ and scalar multiplication $a\phi$ on them that satisfy the usual properties:

1. $(\phi + \psi) + \chi = \phi + (\psi + \chi)$
2. $\phi + \mathbf{0} = \mathbf{0} + \phi = \phi$, for some $\mathbf{0}$ in the space
3. There is a $-\phi$ such that $\phi + -\phi = -\phi + \phi = \mathbf{0}$
4. $\phi + \psi = \psi + \phi$
5. $a(\phi + \psi) = a\phi + a\psi$
6. $(ab)\phi = a(b\phi)$
7. $(a + b)\phi = a\phi + b\phi$
8. $1\phi = \phi$, where 1 is the unity element of K.

These properties are satisfied by obvious definition of the operations:

$$\begin{aligned} (\phi + \psi)_t &= \phi_t + \psi_t \\ (a\phi)_t &= a(\phi_t) \end{aligned}$$

Notice, however, that a space of fields $\Phi(\Omega)$ cannot be closed under these operations, since that would violate our physical realizability constraints (specifically, that dynamic range be bounded).

[3]We suppress K when it is clear from context, and write $\Phi(\Omega)$.

2.1.2 Domains and Ranges

Field Domains are Measure Spaces The domain Ω over which a field is defined is assumed to be a measure space. In most cases the domain will in fact be some closed and bounded subset of a Euclidean space E_n. For example, it might be a finite line segment to represent the frequency of a sound, or a closed disk to represent the light intensity over a retina. Why then have we gone to the extra generality of measure spaces?

There are two reasons. The first is that by keeping the theory general, we ensure that it applies to a wide variety of fields. At this stage it is difficult to anticipate the "shapes" we may need for our fields.

The second reason is that a finite set (when provided with a "weight" function) is a measure space. Thus, although it is our goal to encourage thinking of fields as continuously varying structures, most of the theory in fact applies to fields composed of discrete elements (such as pixel arrays and neural networks).

For our purposes, one of the most important properties of measure spaces is that we can define integration operations over them. For example, if $\phi : \Omega \rightarrow K$, then we can define the (Lebesgue) integral $\int_\Omega \phi_t \mathrm{d}\mu(t)$. Here μ is the measure of the measure space. Since it is usually clear from context, we will write integrals more simply: $\int_\Omega \phi_t \mathrm{d}t$. Also, note that for finite Ω and equal weighting, integration reduces to summation:

$$\int_\Omega \phi_t \mathrm{d}t = \sum_{t \in \Omega} \phi_t \tag{1}$$

The Domain of a Measure Space is Bounded As discussed in Section 2.1.1 we require that fields occupy a finite amount of space. How can this be expressed mathematically? We simply require that the "area" (volume, length) of the space be bounded, by stating:

$$\int_\Omega 1 \mathrm{d}t < \infty \tag{2}$$

Later (Section 4.1) we will see that this can be expressed in the more compact form $\|1\| < \infty$. For convenience we define the "size" of the space Ω by:

$$|\Omega| = \int_\Omega 1 \mathrm{d}t \tag{3}$$

Hence we require that $|\Omega| < \infty$.

The Range of a Field is Bounded We assume that every field $\phi : \Omega \rightarrow K$ has a bound β_ϕ such that $|\phi_t| \leq \beta_\phi$ for all $t \in \Omega$. Furthermore, as mentioned above (Section 2.1.1), we require that a space of fields have limited dynamic range. Thus, for each space $\Phi_K(\Omega)$ there is a β such that $\beta_\phi \leq \beta$ for all $\phi \in \Phi(\Omega)$. Such a β is given by $\beta = \sup_{x \in K} |x| < \infty$.

2.1.3 Bounded Gradient

Fields are Continuous As discussed under Physical Realizability (Section 2.1.1) we assume that all fields are continuous. That is, for all $t, t' \in \Omega$ and every $\epsilon > 0$ there is a $\delta > 0$ such that $|\phi_t - \phi_{t'}| < \epsilon$ whenever $d(t, t') < \delta$. Here $d(t, t')$ is the distance between the points t and t' (recall, Section 2.1.1, that Ω is a metric space).

Uniform Continuity As discussed in Section 2.1.1 we assume that there is a maximum gradient that physical fields can sustain. This implies, first of all, that fields are *uniformly continuous,* which means that for all $\epsilon > 0$ there is a δ such that $|\phi_t - \phi_{t'}| < \epsilon$ whenever $d(t, t') < \delta$. That is, for any change δ in the field's domain, there is a maximum amount ϵ that the field's value can change.

Lipschitz Condition In fact, most fields satisfy a stronger condition than uniform continuity. Specifically, there is a bounded ratio of the change of the field's value to a change in position in the field:

$$\frac{|\phi_t - \phi'_t|}{d(t, t')} \leq \gamma \tag{4}$$

This means that such fields satisfy a *Lipschitz condition (of order 1).*

Fields are Band Limited It is generally reasonable to assume that fields are band limited, that is that they have a frequency limit beyond which there is no information. Higher frequency variation is either physically impossible, or represents noise that should be ignored. This means that field transformations may be assumed to have a low-pass filter on their inputs that does not affect their operation. This has important consequences for the theoretical development (Section 5.5.2).

2.2 Field Transformations

Continuity The noise that accompanies physical processes will cause slight variations in the inputs to field transformations. It is undesirable if this noise has a major effect on the output of the transformation. Therefore, we require that field transformations be *continuous*; this will ensure that small changes in the input will cause at most small changes in the output.

Using the norm that will be defined in Section 4.1, the continuity of field transformations can be expressed:

$$\lim_{n \to 0} \|\phi_n - \phi\| = 0 \ \text{ implies } \ \lim_{n \to 0} \|T(\phi_n) - T(\phi)\| = 0 \tag{5}$$

We can generally make the stronger assumption of uniform continuity: for every $\epsilon > 0$ there is a $\delta > 0$ such that $\|T(\phi) - T(\psi)\| < \epsilon$ whenever $\|\phi - \psi\| < \delta$. In effect, we are assuming that a field transformation is "uniformly insensitive" to a given quantity of noise, no matter what the input field.

Input Filters As noted above (Section 2.1.3), noise often manifests itself as high frequency variation in the field. Therefore, to decrease the effects of noise, field transformation units will often be constructed with a low-pass filter on their input. In other cases, the implementation will naturally suppress high frequencies, and thus behave as though the input is filtered. The foregoing permits us to assume the presence of low pass input filters on field transformations when this simplifies the mathematical analysis (see for example Sections 4.7.2 and 4.7.3).

3 Summary

Since fields have bounded domain (Section 2.1.2) and range (Section 2.1.2), the integral $\int_\Omega {\phi_t}^2 \mathrm{d}t$ exists:

$$\int_\Omega {\phi_t}^2 \mathrm{d}t \le \beta_\phi^2 \int_\Omega 1 \mathrm{d}t \le \beta_\phi^2 |\Omega| \tag{6}$$

We will see below (Section 4.1) that this is equivalent to saying that all fields have finite norms.

Since fields are also continuous (Section 2.1.3), we know that fields over Ω belong to the function space $L_2(\Omega)$. This is very important, because L_2 function spaces are Hilbert spaces, which means that a large body of powerful mathematics can be brought to bear on the problems of field computation (see also Section 4.2).

We cannot conclude, however, that $\Phi(\Omega) = L_2(\Omega)$, since $L_2(\Omega)$ contains many functions that do not satisfy our other constraints (e.g., Sections 2.1.2 and 2.1.3). In most cases these additional constraints will help us. Nevertheless, we must be careful that in applying properties of L_2 spaces we do not violate these other constraints.

Part II
Functional Analysis Preliminaries

In Part II we present those concepts from functional analysis that are required to understand the theory of field computation.

4 Hilbert Spaces

4.1 Definitions

Inner Product For fields $\phi, \psi \in \Phi(\Omega)$ we define the inner product in the usual way:

$$\phi \cdot \psi = \int_{\Omega} \phi_t \psi_t \mathrm{d}t \tag{7}$$

This operation satisfies the properties of a (real) inner product:

1) $\phi \cdot \phi \geq 0$, with $\phi \cdot \phi = 0$ if and only if $\phi = \mathbf{0}$. Here and elsewhere $\mathbf{0}$ represents the zero field, $\mathbf{0}_t = 0$.

2) $\phi \cdot \psi = \psi \cdot \phi$.

3) $(a\phi + b\psi) \cdot \chi = a(\phi \cdot \chi) + b(\psi \cdot \chi)$.

From this it follows that inner products are *bilinear* (linear in both of their arguments).

Orthogonality The orthogonality of two fields is defined, in the usual way, in terms of the inner product: ϕ is *orthogonal* to ψ if and only if $\phi \cdot \psi = 0$.

Norm of Field The norm of a field is a measure of its size. As usual, we use the inner product norm:

$$\|\phi\|^2 = \phi \cdot \phi \tag{8}$$

This is the usual norm for L_2 function spaces (Section 3).[4] Note that since fields have bounded domain and range (Section 2.1.2), we know that $\|\phi\| \leq \beta_\phi |\Omega|^{1/2}$ (see Section 3). That is, all fields are finite in size. As noted before (Section 3), this means that fields belong to Hilbert spaces (Section 4.2).

4.2 Fields Belong to Hilbert Spaces

A Hilbert space is an inner product space that has the additional property of being *complete* with respect to the inner product norm (Section 4.1). That is, if $\phi_1, \phi_2, \ldots$ is a sequence of functions, then $\|\phi_m - \phi_n\| \to 0$ implies that the limit $\lim \phi_n$ exists.

[4]Although it is not the only norm upon which a theory of field computation can be based. In [5, 6] we used the L_1 norm. There seems to be little practical difference, but the L_2 is mathematically more convenient.

The set of continuous real-valued fields satisfying $\|\phi\| < \infty$ is not complete as it stands. But it can be completed in a well-known way to yield the function space $L_2(\Omega)$. We will have to be careful though, since some of the fields created by this completion process are not physically realizable (Section 2.1.1).

A Hilbert space is a special kind of *Banach space* (a complete normed linear space). This fact will be used when we define derivatives of field transformations (Section 5.1).

4.3 Orthonormal Bases

A *unit field* $e \in \Phi(\Omega)$ is a field whose norm is 1, $\|e\| = 1$.

A sequence $e_0, e_1, e_2, \ldots$ of unit fields is called *orthonormal* if its elements are mutually orthogonal: $e_i \cdot e_j = 0$ for $i \neq j$. Of course, $e_i \cdot e_i = \|e_i\|^2 = 1$. Any sequence of mutually orthogonal fields $x_1, x_2, \ldots$ can be *normalized* by $e_i = x_i / \|x_i\|$.

An orthonormal sequence is *complete* if there is no nonzero field that is orthogonal to every element of the sequence. That is, if $\phi \neq \mathbf{0}$, then there is an e_k such that $\phi \cdot e_k \neq 0$.

An *orthonormal (ON) basis* is a complete ON sequence.

The function space $L_2(\Omega)$, which includes all the fields over Ω, has a number of well-known orthonormal bases. For example, over the reals we have the trigonometric basis:

$$\begin{aligned} (e_0)_t &= 1/\sqrt{2\pi} && (9) \\ (e_k)_t &= \cos t/\sqrt{\pi}, k > 0 && (10) \\ (e_{-k})_t &= \sin t/\sqrt{\pi}, -k < 0 && (11) \end{aligned}$$

and various systems of orthonormal polynomials.

Therefore, we will generally assume that spaces of fields have ON bases; we will generally not have to specify the basis that we are using however.

We will sometimes refer to basis elements as being *higher order* or *lower order.* This refers to their location in the sequence e_0, $e_1, \ldots$ Of course this ordering is somewhat arbitrary, and may be changed without affecting the sequence's status as an ON basis. Nevertheless, there is usually a *natural* order, such as increasing frequency for the trigonometric basis, and increasing degree for polynomial bases.

4.4 Generalized Fourier Series

Define $c_k = \phi \cdot e_k$; the real numbers $c_0, c_1, \ldots$ are called the *generalized Fourier coefficients* of ϕ (with respect to the basis $e_0, e_1, \ldots$). It can be shown that every $\phi \in L_2(\Omega)$ can be expanded in a *Fourier series:*

$$\phi = \sum_{k=0}^{\infty} (\phi \cdot e_k) e_k \qquad (12)$$

Other useful properties are *Parseval's equality:*

$$\|\phi\|^2 = \sum_k (\phi \cdot e_k)^2 \tag{13}$$

and the *Parseval relation:*

$$\phi \cdot \psi = \sum_k (\phi \cdot e_k)(\psi \cdot e_k) \tag{14}$$

4.5 Definition of General Field Product

It is useful to define a generalization of the inner product that is continuous-dimensional analogue of products between vectors and matrices. The reason is that these products seem to be the kind of operations that we can expect general purpose field computers to compute. For example, it is well known that linear neural networks compute vector-matrix products ([15], Chapter 9; see also [11]); similarly, these operations correspond to some optical phenomena (see for example [3]). These considerations, together with the theoretical developments described later, lead us to base general purpose field computation on these general field products.[5]

If $\phi \in \Phi(\Omega_2)$ and $\Psi \in \Phi(\Omega_1 \times \Omega_2)$, then the product $\Psi\phi \in \Phi(\Omega_1)$ is defined:

$$(\Psi\phi)_s = \int_{\Omega_2} \Psi_{st}\phi_t \mathrm{d}t \tag{15}$$

If we let Ψ_s be the field $(\Psi_s)_t = \Psi_{st}$, then the product can be defined:

$$(\Psi\phi)_s = \Psi_s \cdot \phi \tag{16}$$

If $\phi \in \Phi(\Omega_1)$ and $\Psi \in \Phi(\Omega_1 \times \Omega_2)$, then the product $\phi\Psi \in \Phi(\Omega_2)$ is defined:

$$(\phi\Psi)_t = \int_{\Omega_1} \phi_s \Psi_{st} \mathrm{d}s \tag{17}$$

If we define the transpose $(\Psi^{\mathrm{T}})_{ts} = \Psi_{st}$, then we can write

$$(\phi\Psi)_t = \phi \cdot {\Psi^{\mathrm{T}}}_t \tag{18}$$

Finally, if $\Psi \in \Phi(\Omega_1 \times \Omega_2)$ and $\mathrm{X} \in \Phi(\Omega_2 \times \Omega_3)$, then we define the product $\Psi\mathrm{X} \in \Phi(\Omega_1 \times \Omega_3)$ as follows:

$$(\Psi\mathrm{X})_{su} = \Psi_s \cdot {\mathrm{X}^{\mathrm{T}}}_u = \int_{\Omega_2} \Psi_{st}\mathrm{X}_{tu} \mathrm{d}t \tag{19}$$

Any linear operator L that can be expressed in the form $L\phi = \Psi\phi$, for some field Ψ, is called an *integral operator.* Indeed, since we require $\|\Psi\| < \infty$, it is an integral operator *of Hilbert-Schmidt type.* We will be interested in expressing derivatives and other linear transformations as integral operators (see Sections 4.7 and 5.5).

[5]Note that the scalar, inner and outer products (see Sect. 4.5) can all be considered degenerate general products.

4.6 The Outer Product

4.6.1 Definition

For two fields $\phi \in \Phi(\Omega_1)$, $\psi \in \Phi(\Omega_2)$ the *outer product* $\phi \wedge \psi \in \Phi(\Omega_1 \times \Omega_2)$ is defined:

$$(\phi \wedge \psi)_{st} = \phi_s \psi_t \tag{20}$$

Since physically realizable fields are bounded (Section 2.1.2), their outer product always exists.

4.6.2 Useful Properties

We present a few simple properties of the outer product.

Theorem 1 *If $\phi \in \Phi(\Omega_1)$ and $\psi, \zeta \in \Phi(\Omega_2)$, then*

$$(\phi \wedge \psi)\zeta = \phi(\psi \cdot \zeta) \tag{21}$$

In other words, ϕ is "weighted" by $\psi \cdot \zeta$.

Proof: Simply expand the product.

□

Corollary 1 *If $\phi, \zeta \in \Phi(\Omega_1)$ and $\eta \in \Phi(\Omega_2)$, then*

$$\phi(\zeta \wedge \eta) = (\phi \cdot \zeta)\eta \tag{22}$$

In other words, η is "weighted" by $\phi \cdot \zeta$.

Theorem 2 *If $\phi \in \Phi(\Omega_0 \times \Omega_1)$, $\zeta \in \Phi(\Omega_1)$ and $\psi, \eta \in \Phi(\Omega_2)$, then*

$$(\phi \wedge \psi)(\zeta \wedge \eta) = \phi\zeta(\psi \cdot \eta) \tag{23}$$

Proof: First expand the product as an integral over the direct product measure space:

$$\begin{aligned}
[(\phi \wedge \psi)(\zeta \wedge \eta)]_r &= \int_{\Omega_1 \times \Omega_2} (\phi \wedge \psi)_{rst}(\zeta \wedge \eta)_{st} \mathrm{d}\mu(s,t) \\
&= \int_{\Omega_1 \times \Omega_2} \phi_{rs}\psi_t\zeta_s\eta_t \mathrm{d}\mu(s,t)
\end{aligned}$$

Next apply Fubini's theorem, which says that the multiple integral equals the iterated integral:

$$\int_{\Omega_1 \times \Omega_2} \phi_{rs}\psi_t\zeta_s\eta_t \mathrm{d}\mu(s,t) = \int_{\Omega_2} \left[\int_{\Omega_1} \phi_{rs}\psi_t\zeta_s\eta_t \mathrm{d}\mu(s) \right] \mathrm{d}\mu(t) \tag{24}$$

Finally, regroup and remove from under the integral sign factors not dependent on the variable of integration:

$$\begin{aligned}
&= \int_{\Omega_1} \phi_{rs}\zeta_s \mathrm{d}\mu(s) \int_{\Omega_2} \psi_t\eta_t \mathrm{d}\mu(t) \\
&= (\phi\zeta)_r(\psi \cdot \eta) \\
&= [\phi\zeta(\psi \cdot \eta)]_r
\end{aligned}$$

□

Corollary 2 *If $\phi, \zeta \in \Phi(\Omega_1)$ and $\psi, \eta \in \Phi(\Omega_2)$, then*

$$(\phi \wedge \psi) \cdot (\zeta \wedge \eta) = (\phi \cdot \zeta)(\psi \cdot \eta) \tag{25}$$

Proof: Since $\phi \cdot \psi = \phi\psi$ (the right-hand side denoting a general field product) the result follows immediately.

□

Corollary 3 $\|\phi \wedge \psi\| = \|\phi\|\|\psi\|$.

Proof:

$$\begin{aligned} \|\phi \wedge \psi\|^2 &= (\phi \wedge \psi) \cdot (\phi \wedge \psi) \\ &= (\phi \cdot \phi)(\psi \cdot \psi) \\ &= \|\phi\|^2\|\psi\|^2 \end{aligned}$$

□

Corollary 4 *If $\phi_k, \psi_k \in \Phi(\Omega_k)$, then*

$$(\phi_0 \wedge \phi_1 \wedge \cdots \wedge \phi_n)(\psi_1 \wedge \cdots \wedge \psi_n) = \phi_0(\phi_1 \cdot \psi_1) \cdots (\phi_n \cdot \psi_n) \tag{26}$$

Proof: Apply the theorem inductively:

$$\begin{aligned} &(\phi_0 \wedge \phi_1 \wedge \cdots \wedge \phi_n)(\psi_1 \wedge \cdots \wedge \psi_n) \\ &\quad= (\phi_0 \wedge \phi_1 \wedge \cdots \wedge \phi_{n-1})(\psi_1 \wedge \cdots \wedge \psi_{n-1})(\phi_n \cdot \psi_n) \\ &\quad\vdots \\ &\quad= (\phi_0 \wedge \phi_1)\psi_1(\phi_2 \cdot \psi_2) \cdots (\phi_n \cdot \psi_n) \\ &\quad= \phi_0(\phi_1 \cdot \psi_1) \cdots (\phi_n \cdot \psi_n) \end{aligned}$$

□

The following theorem and its corollary show the relation between outer products and iterated general products. Note that we take the general product to be left associative: $K\phi\psi = (K\phi)\psi$.

Theorem 3 *Suppose $K \in \Phi(\Omega \times \Omega_2 \times \Omega_1)$, $\phi \in \Omega_1$ and $\psi \in \Omega_2$. Then:*

$$K\phi\psi = K(\psi \wedge \phi) \tag{27}$$

Proof: Simply expand the general products as integrals:

$$\begin{aligned}
(K\phi\psi)_r &= \int_{\Omega_2} (K\phi)_{rs}\psi_s \mathrm{d}s \\
&= \int_{\Omega_2}\int_{\Omega_1} K_{rst}\phi_t \mathrm{d}t\ \psi_s \mathrm{d}s \\
&= \int_{\Omega_2}\int_{\Omega_1} K_{rst}\psi_s\phi_t \mathrm{d}s\mathrm{d}t \\
&= \int_{\Omega_2}\int_{\Omega_1} K_{rst}(\psi \wedge \phi)_{st} \mathrm{d}s\mathrm{d}t
\end{aligned}$$

By Fubini's theorem (see proof of Theorem 4.6.2) the iterated integral may be replaced by the multiple integral over the direct product space:

$$\begin{aligned}
&= \int_{\Omega_2\times\Omega_1} K_{rst}(\psi \wedge \phi)_{st} \mathrm{d}(s,t) \\
&= [K(\psi \wedge \phi)]_r
\end{aligned}$$

□

Corollary 5 *Suppose $K \in \Phi(\Omega \times \Omega_n \times \cdots \times \Omega_1)$ and $\phi_k \in \Phi(\Omega_k)$, for $1 \leq k \leq n$. Then:*

$$K\phi_1\phi_2\cdots\phi_n = K(\phi_n \wedge \cdots \wedge \phi_2 \wedge \phi_1) \tag{28}$$

Proof: An inductive application of the theorem.

□

4.7 Kernels of Linear and Multilinear Operators

4.7.1 Need for Kernels

We explore the conditions under which a linear field transformation $L : \Phi(\Omega_1) \to \Phi(\Omega_2)$ can be expressed as a general field product, $L(\phi) = K\phi$, for some $K \in \Phi(\Omega_2 \times \Omega_1)$. This condition is equivalent to saying that L is an *integral operator (of Hilbert-Schmidt type) with kernel K*. It is an important condition, because field products can be computed by neural networks and other massively parallel devices. The kernel is the analog of the matrix representing a linear transformation in the finite-dimensional case.

After the linear case (Section 4.7.2), we explore the multilinear case (i.e., the case for multi-argument operators that are linear in each of their arguments; see Section 4.7.3).

4.7.2 Kernels of Linear Operators

Theorem 4 *Let*

$$K = \sum_{k=0}^{\infty} L(e_k) \wedge e_k \tag{29}$$

If this field exists, then it is the kernel of L, $L(\phi) = K\phi$.

Proof: To see this, expand the Fourier series for ϕ and make use of linearity of L:

$$\begin{aligned} L(\phi) &= L\left[\sum_{k=0}^{\infty}(\phi \cdot e_k)e_k\right] \\ &= \sum_k L[(\phi \cdot e_k)e_k] \\ &= \sum_k (\phi \cdot e_k)L(e_k) \\ &= \sum_k L(e_k)(e_k \cdot \phi) \\ &= \sum_k [L(e_k) \wedge e_k]\phi \\ &= \left[\sum_k L(e_k) \wedge e_k\right]\phi \\ &= K\phi \end{aligned}$$

This completes the proof.

□

The preceding result assumes that the kernel exists. Sufficient conditions are established next. It will be useful to have the following definitions:

Definition 1 *A linear transformation $F : \Phi(\Omega) \to \Phi(\Omega)$ is called a* filter *if the basis elements e_k are eigenvectors of the transformation. That is, there are λ_k such that $F(e_k) = \lambda_k e_k$. The sequence $\lambda_0, \lambda_1, \ldots$ is called the* transfer function *of the filter.*

Definition 2 *A sequence λ_k is called* absolutely summable *if*

$$\sum_{k=0}^{\infty} |\lambda_k| < \infty \tag{30}$$

.

Another way of saying this is that the sequence belongs to the space l_1.

Theorem 5 *Suppose L can be written in the form $L = G \circ F$, where F is a filter whose transfer function is absolutely summable, and G is a continuous linear transformation. Then the kernel of L exists and is*

$$K = \sum_{k=0}^{\infty} L(e_k) \wedge e_k \tag{31}$$

Proof: We show that the norm of the kernel is finite.

$$\begin{aligned}
\|\sum L(e_k) \wedge e_k\| &\leq \sum \|L(e_k) \wedge e_k\| \\
&= \sum \|L(e_k)\| \|e_k\|, \quad \text{by Section4.6.1} \\
&= \sum \|L(e_k)\|, \quad \text{since the } e_k \text{ are normalized} \\
&= \sum \|G[F(e_k)]\|
\end{aligned}$$

Since F is a filter we know $F(e_k) = \lambda_k e_k$. Hence,

$$\begin{aligned}
&= \sum \|G(\lambda_k e_k)\| \\
&= \sum |\lambda_k| \|G(e_k)\|, \quad \text{since } G \text{ is linear} \\
&= \sum |\lambda_k| \beta \|e_k\|, \quad \text{since } G \text{ is continuous and hence bounded} \\
&= \beta \sum |\lambda_k| \\
&< \infty, \quad \text{since the } \lambda_k \text{ are absolutely summable}
\end{aligned}$$

□

Hence, if the filter sufficiently suppresses the higher-order components of its argument, the kernel will exist. This is certainly the case when F has a sharp cutoff: $\lambda_k = 0$ for all k greater than some N.

In practice the conditions on the preceding theorem are not a problem. Since most fields are band-limited (Section 2.1.3) they can be written $\phi = F(\phi)$ for a filter F with a sharp cutoff. Also, since we generally want our field transformations to be insensitive to higher-order noise, it is useful to express them in the form $G \circ F$, where G is the "ideal" or "goal" transformation and F is an appropriate filter.

Finally, we show that if L can be written as a field product, then the sum-of-outer-products series converges.

Theorem 6 *Suppose that $L : \Phi(\Omega_1) \to \Phi(\Omega_2)$ satisfies $L(\phi) = K\phi$ and $\|K\| < \infty$. Then $\|\sum_k L(e_k) \wedge e_k\| < \infty$.*

Proof: The hypothesis amounts to supposing that L is an integral operator with Hilbert-Schmidt kernel K. It is well known[6] that any such operator on a separable Hilbert space (such as a space of fields) can be represented by an infinite matrix

$$M_{mn} = f_m \cdot L(e_n) \tag{32}$$

[6]See for example [13, p. 68].

where the f_m are an ON basis for $\Phi(\Omega_2)$. In addition, this matrix satisfies

$$\sum_m \sum_n M_{mn}^2 < \infty \tag{33}$$

Now compute a bound on the sum-of-outer-products field:

$$\begin{aligned}
\|\sum_m L(e_m) \wedge e_m\|^2 &= \sum_m \sum_n [L(e_m) \wedge e_m] \cdot [L(e_n) \wedge e_n] \\
&= \sum_m \sum_n L(e_m) \cdot L(e_n)(e_m \cdot e_n) \\
&= \sum_n \|L(e_n)\|^2
\end{aligned}$$

Next expand $L(e_n)$ in a Fourier series and apply Parseval's equality (Section 4.4):

$$\begin{aligned}
&= \sum_n \|\sum_m f_m \cdot L(e_n) f_m\|^2 \\
&= \sum_n \sum_m [f_m \cdot L(e_n)]^2 \\
&= \sum_m \sum_n M_{mn}^2 \\
&< \infty
\end{aligned}$$

□

4.7.3 Kernels of Multilinear Operators

The results in Section 4.7.2 are easily extended to multilinear operators. An operator

$$M : \Phi(\Omega_1) \times \cdots \times \Phi(\Omega_n) \to \Phi(\Omega) \tag{34}$$

is *multilinear* if it is linear in each of its arguments.

Our goal is to find a field

$$K \in \Phi(\Omega \times \Omega_n \times \cdots \times \Omega_1) \tag{35}$$

such that

$$M(\phi_1, \ldots, \phi_n) = K\phi_1 \cdots \phi_n = K(\phi_n \wedge \cdots \wedge \phi_1) \tag{36}$$

That the two products above are equivalent is established in Section 4.6.1.

Theorem 7 *If M is a multilinear operator, then its kernel is*

$$K = \sum_{k_1=0}^{\infty} \cdots \sum_{k_n=0}^{\infty} M(e_{k_1}, \ldots, e_{k_n}) \wedge e_{k_n} \wedge \cdots \wedge e_{k_1} \tag{37}$$

if the sum exists.

Proof: Suppose $\phi_j \in \Phi(\Omega_j)$, $j = 1, \ldots, n$, and $c_{jk} = \phi_j \cdot e_k$, $k = 0, \ldots$ That is, c_{jk} is the k-th generalized Fourier coefficient of ϕ_j. Then, since M is multilinear we can expand:

$$\begin{aligned} M(\phi_1, \ldots, \phi_n) &= M(\sum c_{1k} e_k, \ldots, \sum c_{nk} e_k) \\ &= \sum_{k_1} c_{1k_1} M(e_{k_1}, \ldots, \sum c_{nk} e_k) \\ &= \sum_{k_1} \cdots \sum_{k_n} c_{1k_1} \cdots c_{nk_n} M(e_{k_1}, \ldots, e_{k_n}) \end{aligned}$$

At this point it will be convenient to work from the other side:

$$\begin{aligned} &K(\phi_n \wedge \cdots \wedge \phi_1) \\ &= \left[\sum_{k_1} \cdots \sum_{k_n} M(e_{k_1}, \ldots, e_{k_n}) \wedge e_{k_n} \wedge \cdots \wedge e_{k_1} \right] (\phi_n \wedge \cdots \wedge \phi_1) \\ &= \sum_{k_1} \cdots \sum_{k_n} [M(e_{k_1}, \ldots, e_{k_n}) \wedge (e_{k_n} \wedge \cdots \wedge e_{k_1})](\phi_n \wedge \cdots \wedge \phi_1) \\ &= \sum_{k_1} \cdots \sum_{k_n} [M(e_{k_1}, \ldots, e_{k_n})(e_{k_n} \cdot \phi_n) \cdots (e_{k_1} \cdot \phi_1)], \\ &\quad \text{by Cor. 4 in Sec. 4.6.1} \\ &= \sum_{k_1} \cdots \sum_{k_n} M(e_{k_1}, \ldots, e_{k_n}) c_{nk_n} \cdots c_{1k_1} \end{aligned}$$

The two expansions can be seen to be equal.

□

Theorem 8 *If M is a continuous multilinear operator whose inputs are filtered:*

$$M(\phi_1, \ldots, \phi_n) = G(F_1\phi_1, \ldots, F_n\phi_n) \tag{38}$$

then

$$K = \sum_{k_1=0}^{\infty} \cdots \sum_{k_n=0}^{\infty} M(e_{k_1}, \ldots, e_{k_n}) \wedge e_{k_n} \wedge \cdots \wedge e_{k_1} \tag{39}$$

exists and is the kernel of M, provided the transfer functions of the filters F_k are absolutely summable.

Proof: Let $\lambda^{(i)}$ be the transfer function of the i-th filter F_i; hence ${\lambda^{(i)}}_1$, ${\lambda^{(i)}}_2, \ldots$ are its eigenvalues. As in Theorem 5 we compute a bound on $\|K\|$.

$$\begin{aligned} \|K\| &= \| \sum_{k_1} \cdots \sum_{k_n} M(e_{k_1}, \ldots, e_{k_n}) \wedge e_{k_n} \wedge \cdots \wedge e_{k_1} \| \\ &\le \sum_{k_1} \cdots \sum_{k_n} \| M(e_{k_1}, \ldots, e_{k_n}) \wedge e_{k_n} \wedge \cdots \wedge e_{k_1} \| \\ &= \sum_{k_1} \cdots \sum_{k_n} \| M(e_{k_1}, \ldots, e_{k_n}) \| \end{aligned}$$

Now note that

$$\begin{aligned} M(e_{k_1}, \ldots, e_{k_n}) &= G(F_1 e_{k_1}, \ldots, F_n e_{k_n}) \\ &= G({\lambda^{(1)}}_{k_1} e_{k_1}, \ldots, {\lambda^{(n)}}_{k_n} e_{k_n}) \\ &= {\lambda^{(1)}}_{k_1} \cdots {\lambda^{(n)}}_{k_n} G(e_{k_1}, \ldots, e_{k_n}) \end{aligned}$$

Continuing the derivation of the bound:

$$\begin{aligned} \|K\| &\leq \sum_{k_1} \cdots \sum_{k_n} |{\lambda^{(1)}}_{k_1} \cdots {\lambda^{(n)}}_{k_n}| \, \|G(e_{k_1}, \ldots, e_{k_n})\| \\ &\leq \sum_{k_1} \cdots \sum_{k_n} |{\lambda^{(1)}}_{k_1} \cdots {\lambda^{(n)}}_{k_n}| \end{aligned}$$

The last step follows from the assumption that G is bounded (continuous). Continuing:

$$\begin{aligned} &= \sum_{k_1} \cdots \sum_{k_n} |{\lambda^{(1)}}_{k_1}| \cdots |{\lambda^{(n)}}_{k_n}| \\ &= \prod_{i=1}^{n} \sum_{k} |{\lambda^{(i)}}_{k}| \end{aligned}$$

This will be finite if each of the $\sum_k |{\lambda^{(i)}}_k|$ are finite.

□

4.7.4 Kernels in Terms of Generalized Functions

In this section we derive an alternative formula for the kernel that is often easier to use than those given in Sections 4.7.2 and 4.7.3. The new formula is easily understood through the analogy with finite-dimensional spaces. Recall that a linear transformation $L : \Re^m \to \Re^n$ can be represented by a matrix-vector product $L(x) = Mx$ in which the kth column of M is $L(\delta_k)$. That is, $(M^{\mathrm{T}})_k = L(\delta_k)$. Here the δ_k represent the basis vectors defined by the *Kronecker delta functions*:

$$(\delta_k)_i = \begin{cases} 1, & \text{if } k = i \\ 0, & \text{if } k \neq i \end{cases} \tag{40}$$

Now consider the infinite dimensional case and suppose L is a Hilbert-Schmidt operator, $L(\phi) = K\phi$, where K is given in Section 4.7.2. We claim that the tth "column" of K is $L(\delta_t)$, that is, $(K^{\mathrm{T}})_t = L(\delta_t)$. Here the δ_t are the *Dirac delta functions*:

$$\delta_t(s) = \begin{cases} \infty, & \text{if } s = t \\ 0, & \text{if } s \neq t \end{cases} \tag{41}$$

Although "generalized functions" such as the Dirac delta are not physically realizable, their use often simplifies the derivation of physically realizable fields. See Sections 6.2.4 and 6.2.6 for a further discussion.

To establish our claim we need to make use of the well-known "sifting property" of the Dirac delta:

$$\delta_t \cdot \phi = \int_\Omega \delta_t(s)\phi_s \mathrm{d}s = \phi_t \tag{42}$$

With this in hand it is easy to show that $(K^\mathrm{T})_t = L(\delta_t)$:

$$\begin{aligned}
(K^\mathrm{T})_t &= \left[\sum_k L(e_k) \wedge e_k\right]_t^\mathrm{T} \\
&= \left[\sum_k e_k \wedge L(e_k)\right]_t \\
&= \sum_k e_k(t) L(e_k) \\
&= L\left[\sum_k e_k(t) e_k\right] \\
&= L\left[\sum_k (\delta_t \cdot e_k) e_k\right]
\end{aligned}$$

The expression in brackets is the generalized Fourier series for δ_t, so we conclude $(K^\mathrm{T})_t = L(\delta_t)$. We state the formula for K in two ways, also using an alternate notation for δ_t (explained in Section 6.2.3).

$$(K^\mathrm{T})_t = L(\delta_t) = L(\Delta_t^1) \tag{43}$$

$$K_{st} = [L(\delta_t)]_s = [L(\Delta_t^1)]_s \tag{44}$$

We state without proof the analogous formulas for the kernel of a multilinear operator:

$$K_{st_n \cdots t_1} = [M(\delta_{t_1}, \ldots, \delta_{t_n})]_s \tag{45}$$

$$(K^\mathrm{T})_{t_n \cdots t_1} = M(\delta_{t_1}, \ldots, \delta_{t_n}) \tag{46}$$

In the second equation the transpose must be interpreted as being around the first dimension.

5 Derivatives

5.1 Derivatives of Field Transformations

Our goal is to find ways of approximating field transformations. The notion of a derivative is important for understanding approximations of real functions (for instance, in understanding the Taylor series). For many of the same reasons, it is necessary to investigate the derivatives of field transformations. However, since fields are functions (Section 2.1.1), we need the derivative of an *operator* on a function space. There are two kinds of derivatives that may be defined, the Fréchet (Section 5.2) and Gâteaux (Section 5.3). For fields they happen to be equivalent (Section 5.4).

It will be noted that these derivatives are defined on functions between *Banach spaces* (complete normed linear spaces). Since Hilbert spaces are Banach spaces, we can apply these results (Section 4.2).

5.2 Fréchet Derivatives

Suppose X and Y are two Banach spaces and U is an open subset of X. Then $T : U \to Y$ is *Fréchet differentiable* at ϕ if there is a bounded linear operator $D : X \to Y$ such that the following holds. For all $\alpha \in X$ such that $\phi + \alpha \in U$, there is an $E : X \to Y$ such that

$$T(\phi + \alpha) = T(\phi) + D(\alpha) + E(\alpha) \tag{47}$$

and

$$\lim_{\|\alpha\| \to 0} \frac{\|E(\alpha)\|}{\|\alpha\|} = 0 \tag{48}$$

Under these circumstances, D is called the Fréchet derivative of T at ϕ; it is denoted by $T'(\phi)$. The Fréchet derivative is a locally linear approximation to T; $T'(\phi)(\alpha) = D(\alpha)$ is called the *Fréchet differential* of T.

Since a linear operator is continuous if and only if it is bounded, Fréchet derivatives are (by definition) continuous.

Note that $T' : X \to \mathcal{L}(X, Y)$, where $\mathcal{L}(X, Y)$ is the space of all continuous (bounded) linear operators from X to Y.

5.3 Gâteaux Derivatives

Suppose X and Y are Banach spaces, $U \subseteq X$ is open, and $T : U \to Y$. Then T has a Gâteaux derivative at $\phi \in U$ if, for all $\alpha \in U$ the following limit exists:

$$\mathrm{d}T(\phi, \alpha) = \lim_{t \to 0} \frac{T(\phi + t\alpha) - T(\phi)}{t} = \frac{\mathrm{d}}{\mathrm{d}t} T(\phi + t\alpha)|_{t=0} \tag{49}$$

We write $\mathrm{d}T(\phi, \alpha)$ for the Gâteaux derivative of T at ϕ in the "direction" α. The Gâteaux derivative, if it exists, is unique.

5.4 Some Useful Properties of Derivatives on Function Spaces

1. Every Fréchet derivative is a Gâteaux derivative. Since the Gâteaux derivative is unique, the two derivatives are identical if the Fréchet exists.

2. The derivative of a linear operator is that operator: $L'(\phi) = L$.

3. Higher order derivatives are defined in the obvious way. Suppose $T : X \to Y$. Since $T' : X \to \mathcal{L}(X, Y)$, it is easy to see that the higher derivatives have the types:

$$\begin{array}{l} T'' : X \to \mathcal{L}(X, \mathcal{L}(X, Y)) \\ T^{(3)} : X \to \mathcal{L}(X, \mathcal{L}(X, \mathcal{L}(X, Y))) \end{array} \tag{50}$$

   and so forth. Note that each successive derivative is of "higher type" than its predecessor. We will see below that this leads to the *problem of higher dimensional gradients* (Section 11). [See Section 5.2 for the notation $\mathcal{L}(X, Y)$.]

4. The spaces

$$\mathcal{L}(X, \mathcal{L}(X, \cdots \mathcal{L}(X, \mathcal{L}(X, Y)) \cdots)) \tag{51}$$

   are isomorphic to the spaces $\mathcal{L}(X^k, Y)$, and we will often make use of this fact. We use $\mathrm{d}^k T$ to denote that k-th order "uncurried" derivative of T:

$$\begin{array}{l} \mathrm{d}^k T : X \to \mathcal{L}(X^k, Y) \\ \mathrm{d}^k T(\phi)(\alpha_1, \ldots, \alpha_k) = T^{(k)}(\phi)(\alpha_1) \cdots (\alpha_k) \end{array} \tag{52}$$

5. The derivative of a composition is given by the following equation (shown in both curried and uncurried forms):

$$(T \circ U)'(\phi)(\alpha) = T'[U(\phi)][U'(\phi)(\alpha)] \tag{53}$$

$$\mathrm{d}(T \circ U)(\phi, \alpha) = \mathrm{d}T[U(\phi), \mathrm{d}U(\phi, \alpha)] \tag{54}$$

5.5 Gradients of Operators

5.5.1 Definition of Gradient

Based on the analogy with finite-dimensional spaces [5] we define the *gradient* of a field transformation $T : \Phi(\Omega_1) \to \Phi(\Omega_2)$ at a point $\phi \in \Phi(\Omega_1)$ to be the field $K \in \Phi(\Omega_2 \times \Omega_1)$ satisfying the following property:

$$T'(\phi)(\alpha) = K\alpha, \text{ for all } \alpha \in \Phi(\Omega_1) \tag{55}$$

In other words, the derivative T' is an integral operator with (Hilbert-Schmidt) kernel K. Sufficient conditions for the existence of a gradient are discussed in Section 5.5.2.

The notation $\nabla T(\phi)$ denotes the gradient of T at ϕ. Thus,

$$T'(\phi)(\alpha) = [\nabla T(\phi)]\alpha \tag{56}$$

It will also be convenient to use the notation $\nabla_\alpha T(\phi)$ for the *directional derivative* of T in the "direction" α:

$$\nabla_\alpha T(\phi) = \nabla T(\phi)\alpha = T'(\phi)(\alpha) \tag{57}$$

This permits ∇_α to be treated as an operator; operator techniques are exploited in [5, 6].

It will be useful to consider the form taken by higher order directional derivatives. Supposing that all the gradients exist, observe:

$$\begin{aligned} T^{(k)}(\phi)(\alpha_1)\cdots(\alpha_k) &= \nabla^k T(\phi)\alpha_1\cdots\alpha_k \\ &= \nabla^k T(\phi)(\alpha_k \wedge \cdots \wedge \alpha_1) \\ &= \nabla_{\alpha_k}\cdots\nabla_{\alpha_1} T(\phi) \end{aligned}$$

Thus, a k-th order differential can be expressed as a k-fold product with the k-th order gradient (if it exists), or as a product between the gradient and the k-fold outer product. These relationships depend on the properties of outer products discussed in Section 4.6.2.

5.5.2 When Can a Derivative Be Expressed as a Product?

Another way of asking this is "When do gradients exist?" Yet another way of asking it is, "When are derivatives integral operators (of Hilbert-Schmidt type)?"

In this section we present practical sufficient conditions for the existence of gradients. Since derivatives are linear (or multilinear) these conditions are direct applications of the results in Sections 4.7.2 and 4.7.3.

For practical applications it seems reasonable to assume that field transformations have filtered inputs (see Section 4.7.2 for the definition of a filter). Typically there are limitations on the gradients sustainable in the media used to represent the input fields to a transformation (Section 2.1.3). Also, higher order components typically represent noise, and we do not want our transformations to be excessively sensitive to noise (Section 2.2).

Therefore, we consider the form of derivatives of transformations that can be written $T = G \circ F$, where F is a filter (Theorem 5) and G is the "ideal" operator to be computed. By the formula for the derivative of a composition (Eq. 53) we have:

$$\begin{aligned} T'(\phi)(\alpha) &= (G \circ F)'(\phi)(\alpha) \\ &= G'[F(\phi)][F'(\phi)(\alpha)] \\ &= (G' \circ F)(\phi)[F'(\phi)(\alpha)] \end{aligned}$$

Notice that the input to G' is also filtered by F: $(G' \circ F)(\phi)$. Also note that since F is a filter it is linear, and so by Section 5.4 $F'(\phi) = F'$. Hence,

$$T'(\phi)(\alpha) = (G' \circ F)(\phi)[F(\alpha)] \tag{58}$$

That is,

$$T'(\phi) = (G' \circ F)(\phi) \circ F \tag{59}$$

Hence, $T'(\phi)$ can be written in the form $H \circ F$, so the gradient $\nabla T(\phi)$ exists. Since the gradient has the appropriate form $(H \circ F)$, the higher order gradients also exist.

The existence of gradients of multilinear operators exist under similar circumstances.

Summarizing, gradients exist for operators that are sufficiently insensitive to higher-order (typically, higher frequency) components of their inputs. In particular, if the operators are band-limited (insensitive to all components beyond a certain order) then the gradients exist. However, they also exist if there response rolls off as an absolutely summable sequence.

Part III
Linear and Multilinear Operators

6 Linear Operators

6.1 Introduction

Many of the most important field transformations are linear and multilinear operators. However, in their ideal forms they often do not satisfy the conditions in Theorems 5 and 8 for the existence of their kernels. In these cases we have to consider the approximation of the ideal operation by a field product.

6.2 Examples

6.2.1 Definite Integral

Definition The definite integral operator defint : $\Phi(\Omega) \to \Re$ simply computes the total value of the field:

$$\text{defint}\,\phi = \int_\Omega \phi_t \mathrm{d}t \tag{60}$$

If we let $\mathbf{1}_t = 1$ be the constant 1 field, then we can define the definite integral:

$$\text{defint}\,\phi = \phi \cdot \mathbf{1} \tag{61}$$

The definite integral is often useful, especially for computing the mean of a field. For example, by subtracting from a field its mean, we may get maximum use of the dynamic range of a field storage unit. The definite integral is also useful for automatic gain control.

Formula

$$\text{defint}\,\phi = \mathbf{1} \cdot \phi \tag{62}$$

Since $\mathbf{1}$ is a physically realizable field, no approximation is involved in this field computation of the definite integral.

Proof Simply observe:

$$\begin{aligned} \mathbf{1} \cdot \phi &= \int_\Omega \mathbf{1}_t \phi_t \mathrm{d}t \\ &= \int_\Omega \phi_t \mathrm{d}t \\ &= \text{defint}\,\phi \end{aligned}$$

Alternately, we compute the kernel according to Equation 29:

$$\begin{aligned} K &= \sum_{k=0}^{\infty} (\text{defint}\, e_k) \wedge e_k \\ &= \sum_{k=0}^{\infty} (\mathbf{1} \cdot e_k) \wedge e_k \end{aligned}$$

Now note that $\mathbf{1} \cdot e_k$ is a scalar, and that an outer product with a scalar is the same as a scalar product. That is, if a is a scalar, then:

$$a \wedge \phi = \phi \wedge a = a\phi \tag{63}$$

Therefore, the formula for the kernel is:

$$K = \sum_{k=0}^{\infty} (\mathbf{1} \cdot e_k) e_k \tag{64}$$

But this is just the Fourier expansion (Section 4.4) of $\mathbf{1}$, so $K = \mathbf{1}$.

6.2.2 Indefinite Integral

Definition For illustrative purposes we take $\Omega = [0, 1]$. The indefinite integral $\int \phi$ of a field ϕ is then defined:

$$(\int \phi)_s = \int_0^s \phi_t \mathrm{d}t \tag{65}$$

Formula

$$\int \phi = \Delta^0 \phi \tag{66}$$

where

$$\Delta^0{}_{st} = \begin{cases} 1, & \text{if } s \geq t \\ 0, & \text{if } s < t \end{cases} \tag{67}$$

The *unit step field* (or *Heaviside field*) Δ^0 can be visualized as follows: it is 1 above the $s = t$ diagonal and zero below it. Although Δ^0 is discontinuous (and thus violates our physical realizability constraints, Section 2.1.1), it can be approximated arbitrarily closely by continuous functions.

The point of the Δ^0 notation will become apparent later (Sections 6.2.3 and 6.2.5). An alternative formula for the indefinite integral will be derived in Section 7.2.4.

Proof To see that Δ^0 is the kernel, observe:

$$\begin{aligned} (\int \phi)_s &= \int_0^s \phi_t \mathrm{d}t \\ &= \int_0^1 \Delta^0{}_{st} \phi_t \mathrm{d}t \\ &= \Delta^0{}_s \cdot \phi \\ &= (\Delta^0 \phi)_s \end{aligned}$$

To see that Δ^0 exists, observe:

$$\begin{aligned}
\|\Delta^0\|^2 &= \int_0^1 \int_0^1 (\Delta^0_{st})^2 \mathrm{d}t\mathrm{d}s \\
&= \int_0^1 \int_0^s 1 \ \mathrm{d}t\mathrm{d}s \\
&= \int_0^1 s \ \mathrm{d}s \\
&= 1/2 \\
&< \infty
\end{aligned}$$

6.2.3 Product Mask

Definition The product mask $\mu \times \phi$ computes a point-wise product between the given field ϕ and a fixed field μ:

$$(\mu \times \phi)_t = \mu_t \phi_t \tag{68}$$

This is obviously a linear operator. In addition to its obvious use for masking out part of a field, it may also be used with defint to compute weighted averages of fields.

Formula

$$\mu \times \phi = K\phi \tag{69}$$

where

$$K_{st} = \mu_s \delta(s-t) \tag{70}$$

Here we have made use of the *Dirac delta function* (or *unit impulse function*). This "generalized function" has the value $+\infty$ at the origin, and the value 0 everywhere else. An alternative notation for K is:

$$K_{st} = \mu_s {\Delta^1}_{st} \tag{71}$$

where

$${\Delta^1}_{st} = \delta(s-t) \tag{72}$$

We prefer this notation because Δ^1 is the derivative along the second coordinate of Δ^0, which was defined in Section 6.2.2.

$${\Delta^1}_{st} = \mathrm{d}{\Delta^0}_{st}/\mathrm{d}t \tag{73}$$

Since the kernel K is defined in terms of the *unit impulse field* Δ^1, it is not physically realizable, and we must use an approximation; see Section 6.2.4.

Proof This formula for the kernel can be established by Equation 44:

$$K_{st} = (\mu \times \Delta^1_t)_s = \mu_s \Delta^1_{ts} = \mu_s \Delta^1_{st} \tag{74}$$

(since Δ^1 is symmetric). Notice that K is not physically realizable; indeed, it is not even a Hilbert-Schmidt kernel (since $\|K\| = \infty$). We turn to this problem next.

6.2.4 Delta Functions and Physical Realizability

The field K, defined as it is in terms of the Dirac delta function, violates several of our physical realizability conditions (Section 2.1.1), since it is infinite valued at the origin and discontinuous. However, we can approximate it by various realizable functions. For example, we can approximate it by a square wave, $\delta(x) \approx S_\epsilon(x)$, where:

$$\begin{aligned} S_\epsilon(x) &= 1/\epsilon, \text{ if } -\epsilon/2 < x < +\epsilon/2 \\ S_\epsilon(x) &= 0, \text{ otherwise} \end{aligned}$$

Clearly, $\delta = \lim_{\epsilon \to 0} S_\epsilon$. Similarly, we could approximate δ by a triangular wave, or a Gaussian distribution, or any number of other standard functions.

All of these approximations have the effect of "smearing out" the product $\mu \times \phi$. For example, with S_ϵ:

$$\begin{aligned} (K\phi)_s &= \int_\Omega \mu_s S_\epsilon(t-s)\phi_t \mathrm{d}t \\ &= \mu_s \int_{s-\epsilon/2}^{s+\epsilon/2} \phi_t \mathrm{d}t \end{aligned}$$

Notice that μ_s is multiplied by the average of the values of ϕ in an interval of width ϵ centered on s.

From time to time we will make use of singularity functions such as the Dirac delta function. Although they are convenient for the theoretical development, keep in mind that physical realizability requires them to be approximated. This is not very different from the familiar situation of numerical approximation on digital computers. On the other hand, delta functions are significant in the theoretical development, since they show us when an operator is *local,* that is, the value of the output field at a point depends on the value of the input field at only one or a few points. This is important, because local operators can be implemented with very sparse interconnections between layers in a neural network (see [11] for more information).

6.2.5 Derivative

Definition There are of course many derivative and derivative-like operators that can be defined on spaces of fields. In this case we take $\Omega = [a, b]$ and consider the derivative operator on this closed interval: $D\phi = \phi'$; that is, $(D\phi)_t = \mathrm{d}\phi_t/\mathrm{d}t$.

Formula

$$D\phi = \Delta^2\phi \tag{75}$$

where

$$\Delta^2{}_{st} = -\delta'(s-t) \tag{76}$$

Here we make use of the *doublet* δ', which is the derivative of the *Dirac delta function* or *unit impulse function* δ (see Section 6.2.3). The doublet is a very unusual function. Its value is zero everywhere, except "just to the left" of 0, where its value is $+\infty$, and "just to the right" of 0, where its value is $-\infty$. Therefore the *doublet field* Δ^2 has the value $-\infty$ "just below" the $s = t$ line, and the value $+\infty$ "just above" it. Since the field Δ^2 is defined in terms of this "generalized function," it is not physically realizable; this issue is discussed later (Section 6.2.6).

Note that Δ^2 is the derivative of Δ^1 along its second coordinate:

$$\Delta^2{}_s = D\Delta^1{}_s \tag{77}$$

See Section 7.2.4 for an alternative approach to the field computation of the derivative.

Proof The simplest proof of this result uses the following "sifting" property of the doublet:

$$-\phi'_s = \int_\Omega \delta'(s-t)\phi_t \mathrm{d}t \tag{78}$$

Hence, letting $\Delta^2{}_{st} = -\delta'(s-t)$, it is immediate that

$$\phi'_s = \Delta^2{}_s \cdot \phi \tag{79}$$

Alternately, use the formula for the kernel given in Section 4.7.4:

$$K_{ts} = (D\Delta^1_s)_t = \Delta^2_{st} \tag{80}$$

Note however that Δ^2 is not a Hilbert-Schmidt kernel, since $\|K\| = \infty$.

It may be instructive to consider the special case of the trigonometric basis. Suppose $a = -\pi$, $b = \pi$ and:

$$\begin{aligned} e_0(t) &= 1/\sqrt{2\pi} \\ e_{2n-1}(t) &= \cos nt/\sqrt{2\pi}, \quad n = 1, 2, \cdots \\ e_{2n}(t) &= \sin nt/\sqrt{2\pi}, \quad n = 1, 2, \cdots \end{aligned}$$

First apply the formula for the kernel (Eq. 29):

$$\begin{aligned} K &= \sum_k De_k \wedge e_k \\ &= \sum_k e'_k \wedge e_k \end{aligned}$$

For the trigonometric basis, observe that:

$$\begin{aligned} e_0'(t) &= 0 \\ e_{2n-1}'(t) &= -n \sin nt/\sqrt{2\pi} = -ne_{2n}(t) \\ e_{2n}'(t) &= n \cos nt/\sqrt{2\pi} = ne_{2n-1}(t) \end{aligned}$$

Hence,

$$\begin{aligned} e_0' &= \mathbf{0} \\ e_{2n-1}' &= -ne_{2n} \\ e_{2n}' &= ne_{2n-1} \end{aligned}$$

Hence, the kernel is

$$\begin{aligned} K &= \sum_{n=1}^{\infty}(e_{2n-1}' \wedge e_{2n-1} + e_{2n}' \wedge e_{2n}) \\ &= \sum_{n=1}^{\infty} n(e_{2n-1} \wedge e_{2n} - e_{2n} \wedge e_{2n-1}) \end{aligned}$$

Now note that

$$\begin{aligned} (e_{2n-1} \wedge e_{2n})_{st} &= e_{2n-1}(s)e_{2n}(t) \\ &= (\cos ns/\sqrt{2\pi})(\sin nt/\sqrt{2\pi}) \\ &= \sin nt \cos ns/2\pi \end{aligned}$$

Similarly,

$$(e_{2n} \wedge e_{2n-1})_{st} = \cos nt \sin ns/2\pi \qquad (81)$$

Therefore,

$$\begin{aligned} (e_{2n-1} \wedge e_{2n} - e_{2n} \wedge e_{2n-1})_{st} &= (\sin nt \cos ns - \cos nt \sin ns)/2\pi \\ &= \sin(nt - ns)/2\pi \\ &= \sin n(t-s)/2\pi \end{aligned}$$

Therefore the kernel is defined by:

$$K_{st} = \frac{1}{2\pi}\sum_{n=1}^{\infty} n \sin n(t-s) \qquad (82)$$

This is the Fourier series for $-\delta'(t-s)$ as can be seen by noting that the sines "pile up" to $-\infty$ on the negative side of 0, and to $+\infty$ on the positive side of 0. Clearly K is not physically realizable; see Section 6.2.4 as well as the following section.

6.2.6 Doublets and Physical Realizability

The field Δ^2, defined as it is in terms of the doublet, violates several of our physical realizability conditions (Section 2.1.1), since it is infinite valued at the origin and discontinuous. However, just as we did for Δ^1 (Section 6.2.4), we can approximate Δ^2 by various realizable functions. For example, we can truncate the Fourier series given above. More directly, we can approximate it by a square wave, ${\Delta^2}_{st} \approx S_\epsilon(s,t)$, where:

$$\begin{aligned} S_\epsilon(s,t) &= -2/\epsilon^2, \ \text{ if } -\epsilon/2 < s-t < 0 \\ S_\epsilon(s,t) &= +2/\epsilon^2, \ \text{ if } 0 < s-t < +\epsilon/2 \\ S_\epsilon(s,t) &= 0, \ \text{ otherwise} \end{aligned}$$

Clearly, $\Delta^2 = \lim_{\epsilon\to 0} S_\epsilon$. By using S_ϵ we are approximating the derivative by the "difference":

$$\phi'_t \approx \frac{\int_0^{\epsilon/2} \phi_t \mathrm{d}t - \int_{-\epsilon/2}^{0} \phi_t \mathrm{d}t}{\epsilon} \tag{83}$$

Notice that in field computation the difference is computed between two "average" values, whereas in digital computation it is computed between the values at two points.

6.2.7 Discrete Fourier Transform

Definition Let f_0, f_1, $f_2, \ldots$ be an orthogonal basis for $\Phi(\Omega_2)$ and suppose $\Omega_1 = 0, 1, 2, .., N$. The result of the discrete Fourier transform $\mathsf{F}_N\phi$ is an $N+1$ element discrete field σ, such that σ_n is the nth Fourier coefficient of ϕ with respect to the f_n. That is,

$$(\mathsf{F}_N\phi)_n = \phi \cdot f_n \tag{84}$$

Note that we extract only the first $N+1$ coefficients. There are two reasons for this. First, extracting all the coefficients would require Ω_1 to be infinite (i.e. the natural numbers), which violates our condition that the domains of fields be bounded (Section 2.1.2). The second reason is that higher order coefficients frequently represent noise, and so should be suppressed. Nevertheless, later in this section we show a way to capture the full spectrum in a physically realizable field.

Formula

$$\mathsf{F}_N\phi = K\phi \tag{85}$$

where

$$K = \sum_{n=0}^{N} \delta_n \wedge f_n \tag{86}$$

Here δ_n is the *Kronecker delta function* defined by:

$$\begin{aligned}\delta_n(n) &= 1\\ \delta_n(m) &= 0, n \neq m\end{aligned}$$

Alternately we can define the Kronecker delta in a way analogous to the Dirac delta (Section 6.2.3). Thus $\delta(k)$ is 1 if $k = 0$, but 0 if $k \neq 0$. Then K is defined:

$$K_{mt} = \sum_{n=0}^{N} \delta(m-n) f_n(t) \tag{87}$$

Observe that K is in effect an "array" of the basis functions $f_0, f_1, \ldots, f_N$; $K_m = f_m$. Alternately, we may say that the kernel *is* the (truncated) series of basis functions.

Proof Although it is straight-forward to show the correctness of the formula for K by expanding the product $K\phi$, we will show that the formula can be derived directly from the definition, $(\mathsf{F}_N\phi)_n = \phi \cdot f_n$. First note that any $N+1$-element "array" (discrete field) A satisfies

$$A = \sum_{n=0}^{N} \delta_n A_n \tag{88}$$

since

$$\begin{aligned}A_k &= (\sum_n \delta_n A_n)_k\\ &= \sum_n \delta_n(k) A_n\\ &= \delta_k(k) A_k\\ &= A_k\end{aligned}$$

Therefore, the Fourier transform can be represented:

$$\begin{aligned}\mathsf{F}_N\phi &= \sum_{n=0}^{N} \delta_n (\mathsf{F}_N\phi)_n\\ &= \sum_{n=0}^{N} \delta_n(\phi \cdot f_n)\\ &= \sum_{n=0}^{N} \delta_n(f_n \cdot \phi)\\ &= \sum_{n=0}^{N} (\delta_n \wedge f_n)\phi\\ &= \left(\sum_{n=0}^{N} \delta_n \wedge f_n\right)\phi\\ &= K\phi\end{aligned}$$

6.2.8 Representing the Full Spectrum

To represent the full spectrum, we must find some way of fitting the infinite set of natural numbers into finite space. Although there are a number of ways of doing this, we choose a representation that may have some practical applications. We define the following family of "regressive pulse functions":

$$\begin{aligned} r_n(x) &= 1, \ \text{if } 2^{-n} < x < 2^{-n-1} \\ r_n(x) &= 0, \ \text{otherwise} \end{aligned}$$

In other words, r_0, $r_1, \ldots$ are a series of exponentially narrower contiguous pulses of unit amplitude. We will use the nonzero portion of each pulse to represent a Fourier coefficient; therefore higher order coefficients will occupy exponentially less space. It is easy to show that this transformation is the integral operator:

$$\mathsf{F}_\infty \phi = K\phi = \left(\sum_{n=0}^{\infty} r_n \wedge f_n \right) \phi \tag{89}$$

provided that the infinite sum exists. To show that it does, we assume that the f_n are orthonormal (vice merely orthogonal) and observe that $\|r_n\| = 2^{-n-1}$. Then, derive:

$$\begin{aligned} \|K\| &\leq \sum_n \|r_n \wedge f_n\| \\ &= \sum_n \|r_n\| \|f_n\| \\ &= \sum_n 2^{-n-1} \\ &= 1 < \infty \end{aligned}$$

6.2.9 Continuous Fourier Transform

Definition In this case we replace the discrete series of basis functions f_0, $f_1, \ldots$ by a family of functions that depends continuously on a parameter ω. For example, we may replace the complex exponential basis $f_n(t) = e^{-in\omega_0 t}$ by the family:

$$f_\omega(t) = e^{-i\omega t} \tag{90}$$

If we are concerned with the spectrum in only a finite interval $\Omega = [a, b]$, then there is no difficulty extending the discrete case to the continuous case. Here we define:

$$(\mathsf{F}_{[a,b]}\phi)_\omega = \phi \cdot f_\omega \tag{91}$$

If we want the full spectrum, then we have a representation problem, since the domain of the resulting field would be $[0, \infty)$, which is not physically realizable. In this case we choose some continuous monotonic mapping ρ from $[0, \infty)$ into a bounded domain. An example ρ is

$$\rho(\omega) = 1/(\omega + 1) \tag{92}$$

The resulting definition of the transform function is:

$$(\mathsf{F}_\rho \phi)_{\rho(\omega)} = \phi \cdot f_\omega \tag{93}$$

The field returned by $\mathsf{F}_\rho \phi$ is physically realizable, since the image of $[0, \infty)$ under ρ is $[0, 1]$.

Formula In the band-limited case we use the analogous formula to that in Section 6.2.7:

$$\mathsf{F}_{[a,b]}\phi = K\phi \tag{94}$$

where

$$K_{\omega t} = f_\omega(t) \tag{95}$$

To compute the full spectrum, a different kernel is required:

$$K_{st} = f_{\rho^{-1}(s)}(t) \tag{96}$$

Note that ρ^{-1} exists, since ρ is monotonic.

Also note that, as before, the kernel *is* the orthonormal basis (possibly reindexed to ensure a bounded domain).

Proof First consider the band-limited case. Derive:

$$\begin{aligned} (K\phi)_\omega &= K_\omega \cdot \phi \\ &= f_\omega \cdot \phi \\ &= (\mathsf{F}_{[a,b]}\phi)_\omega \end{aligned}$$

Further, the field K exists, since f_ω depends continuously on ω.

For the full-spectrum case derive:

$$\begin{aligned} (K\phi)_s &= K_s \cdot \phi \\ &= f_{\rho^{-1}(s)} \cdot \phi \\ &= (\mathsf{F}_\rho \phi)_s \end{aligned}$$

Hence $(K\phi)_{\rho(\omega)} = f_\omega \cdot \phi$. It remains to show that K exists. Assume an orthonormal basis, $\|f_\omega\| = 1$. Then:

$$\begin{aligned} \|K\|^2 &= \int_{\Omega_1 \times \Omega_2} K_{st}^2 \mathrm{d}(s,t) \\ &= \int_{\Omega_1} \int_{\Omega_2} f_{\rho^{-1}(s)}^2(t) \mathrm{d}t \mathrm{d}s \\ &= \int_{\Omega_1} \|f_{\rho^{-1}(s)}\|^2 \mathrm{d}s \\ &= \int_{\Omega_1} 1 \mathrm{d}s \\ &= |\Omega_1| < \infty \end{aligned}$$

6.2.10 Inverse Fourier Transform

Definition We consider the case of the finite-spectrum discrete Fourier transform, although it will be apparent that the solution can be simply extended to the full-spectrum or continuous cases. Our goal is as follows. Suppose that the generalized coefficients of ϕ beyond the Nth are zero, then we want F_N^{-1} such that:

$$\mathsf{F}_N^{-1}(\mathsf{F}_N\phi) = \phi \tag{97}$$

We are given a discrete field c such that c_n is the nth generalized Fourier coefficient, $n = 0, 1, \ldots, N$. Then F_N^{-1} takes the simple form:

$$\mathsf{F}_N^{-1}c = \sum_{n=0}^{N} c_n f_n \tag{98}$$

Formula

$$\mathsf{F}_N^{-1}c = cK = K^{\mathrm{T}}c \tag{99}$$

where K is the kernel of the discrete Fourier transform (Section 6.2.7), which is simply the "array" of basis functions. If we use F_N for this kernel, then the inverse transform is expressed more obviously by:

$$\mathsf{F}_N^{-1}c = c\mathsf{F}_N = \mathsf{F}_N^{\mathrm{T}}c \tag{100}$$

Proof Recall (Section 6.2.7) that the kernel of the discrete Fourier transform is defined:

$$K_{nt} = f_n(t) = f_{nt} \tag{101}$$

That is, $K_n = f_n$. Therefore,

$$\begin{aligned}(\mathsf{F}_N^{-1}c)_t &= \sum_n c_n f_{nt} \\ &= \sum_n c_n K_{nt} \\ &= (cK)_t\end{aligned}$$

Hence $\mathsf{F}_N^{-1}c = cK = K^{\mathrm{T}}c$.

6.2.11 Using Fourier Methods to Compute Linear Operators

The Fourier transform permits a way computing linear operators that is particularly suitable for neural implementation [11]. To see this observe:

$$\begin{aligned}L\phi &= L(\sum_n c_n f_n) \\ &= \sum_n c_n L(f_n) \\ &= \sum_n c_n \lambda_n\end{aligned}$$

Here the c_n are the Fourier coefficients of ϕ, $c = \mathsf{F}_N\phi$, and the λ_n are the values of L on the basis functions (i.e., λ_0, $\lambda_1, \ldots$ is the *transfer function* of L). Then

$$L(\phi) = c\Lambda = (\mathsf{F}_N\phi)\Lambda = \Lambda^{\mathrm{T}}(\mathsf{F}_N\phi) \tag{102}$$

where

$$\Lambda_n = L(f_n) \tag{103}$$

In other words, any linear operator can be computed by extracting the Fourier coefficients of its argument and using these to weight the values of the operator on the basis fields. The advantage of this for neural implementation is that if the input is band limited, then there are only a finite number of coefficients. These can be represented by the "hidden units" between the neural layers that compute F_N and Λ^{T}. The kernel of the linear operator is of course $\Lambda^{\mathrm{T}}\mathsf{F}_N$.

The foregoing suggests a generalization based on the continuous Fourier transform. We may compute any linear operator by:

1. Taking its Fourier transform (discrete or continuous).
2. Multiplying the result by a (discrete or continuous) product mask representing the operator's transfer function.
3. Taking the corresponding inverse Fourier transform.

This suggests that a general purpose field computer could be structured around Fourier transforms.

7 Multilinear Operators

7.1 Introduction

Like the linear operators discussed in Section 6, many of the multilinear operators discussed here are not implementable by physically realizable field products. They can, however, be approximated in a straight-forward way.

7.2 Examples

7.2.1 Local Product

Definition The local product $\phi \times \psi$ of two fields over the same domain is defined:

$$(\phi \times \psi)_s = \phi_s\psi_s \tag{104}$$

Formula

$$\phi \times \psi = M\phi\psi = M(\psi \wedge \phi) \tag{105}$$

where

$$M_s = \Delta^1{}_s \wedge \Delta^1{}_s \tag{106}$$

The unit impulse field Δ^1 is defined in Section 6.2.3. Since Δ^1 is not physically realizable, M will have to be approximated; see Section 6.2.4. Notice that the field product $M\phi\psi$ is a very inefficient way to compute the local product $\phi \times \psi$. The field product brings together all the possible combinations (ψ_s, ϕ_t), but the field M ignores all except those for which $s = t$. It is wasteful to use the power of global computation where only local computation is required. This suggests that most general-purpose field computers will have the local product operation built in.

Proof It is easy to check that the formula for M is correct:

$$\begin{aligned} [M(\psi \wedge \phi)]_s &= M_s \cdot (\psi \wedge \phi) \\ &= (\Delta^1{}_s \wedge \Delta^1{}_s) \cdot (\psi \wedge \phi) \\ &= (\Delta^1{}_s \cdot \psi)(\Delta^1{}_s \cdot \phi) \\ &= \psi_s \phi_s \end{aligned}$$

The last two steps follow from Cor. 2 (Section 4.6.1) and the sifting property of Δ^1 (Eq. 42).

It is also easy to derive the formula for M directly from the formula for the kernel of a multilinear operator (Section 4.7.3):

$$M = \sum_k \sum_l (e_k \times e_l) \wedge e_l \wedge e_k \tag{107}$$

Hence,

$$\begin{aligned} M_{stu} &= \sum_k \sum_l (e_k \times e_l)_s e_l(t) e_k(u) \\ &= \sum_k \sum_l e_l(s) e_l(t) e_k(s) e_k(u) \\ &= \left[\sum_l e_l(s) e_l(t)\right] \left[\sum_k e_k(s) e_k(u)\right] \\ &= \left[\sum_l (\Delta^1{}_s \cdot e_l) e_l(t)\right] \left[\sum_k (\Delta^1{}_s \cdot e_k) e_k(u)\right] \end{aligned}$$

The last step is by the sifting property. Now observe that in the brackets we have the Fourier series for Δ^1, hence:

$$\begin{aligned} M_{stu} &= \Delta^1{}_s(t) \Delta^1{}_s(u) \\ &= (\Delta^1{}_s \wedge \Delta^1{}_s)_{tu} \end{aligned}$$

7.2.2 Linear Measure Spaces

In order to define the convolution and correlation of arbitrary fields we need a subtraction operation on the domains of fields. That is, for any $\phi \in \Phi(\Omega)$ for which we might want a convolution or correlation, we need $s - t \in \Omega$ to be defined for all $s, t \in \Omega$.

To this end, we define a *linear measure space* to be a measure space that is also a linear space. Therefore, it has addition, subtraction and scalar multiplication operations satisfying the usual properties.

There is one difficulty with this definition. Since a linear space must be closed with respect to its operations (addition, subtraction and scalar multiplication), it cannot be bounded, since it must contain at for every real number a and every $t \in \Omega$. Thus linear measure spaces violate one of our physical realizability constraints, namely, that the domains of fields be bounded (Section 2.1.2). This is a problem that is commonly faced in the analysis of linear, shift-invariant systems, since most implementable systems are bounded, and hence not completely shift-invariant. We shall take the same pragmatic approach here that is commonly applied in that analysis: apply the theory based on linear spaces, but be careful of "edge effects."

7.2.3 Convolution

Definition If Ω is a linear measure space (Section 7.2.2), then the convolution $\phi \star \psi$ of two fields $\phi, \psi \in \Phi(\Omega)$ is defined:

$$(\phi \star \psi)_s = \int_\Omega \phi_{s-t} \psi_t \mathrm{d}t \tag{108}$$

Formula

$$\phi \star \psi = M\phi\psi = M(\psi \wedge \phi) \tag{109}$$

where

$$M_{st} = \Delta^1{}_{st} \tag{110}$$

Since the field M is defined in terms of the unit impulse field Δ_1 (Section 6.2.4), it must in practice be approximated.

Proof It is easiest to establish this result by direct expansion of the product:

$$\begin{aligned}
(M\phi\psi)_s &= (M\phi)_s \cdot \psi \\
&= \left(\int_\Omega M_{st} \phi_t \mathrm{d}t \right) \cdot \psi \\
&= \left(\int_\Omega \Delta^1{}_{st} \phi_t \mathrm{d}t \right) \cdot \psi \\
&= \int_\Omega (\Delta^1{}_{st} \cdot \psi) \phi_t \mathrm{d}t
\end{aligned}$$

$$\begin{aligned} &= \int_\Omega \psi_{s-t}\phi_t \mathrm{d}t \\ &= (\phi \star \psi)_s \end{aligned}$$

The second to last step follows from the sifting property of Δ^1.

7.2.4 Use of Convolution to Implement Linear, Shift-Invariant Operators

Many linear, shift-invariant operators can be implemented more efficiently by convolution than by general field product. For example, the derivative operator (Section 6.2.5) can be implemented by:

$$D\phi = -\delta' \star \phi \tag{111}$$

where δ' is the unit doublet (Section 6.2.5). To see this, observe

$$(-\delta' \star \phi)_s = -\int \delta'_{s-t}\phi_t \mathrm{d}t = \phi'_s \tag{112}$$

by the sifting property of the doublet (Section 6.2.5). This formula should be compared with that in Section 6.2.5:

$$D\phi = \Delta^2\phi \tag{113}$$

Although neither δ' nor Δ^2 is physically realizable, δ' has the advantage that it is of lower dimension: $\delta' \in \Phi(\Omega)$, but $\Delta^2 \in \Phi(\Omega \times \Omega)$. Therefore, δ' will generally be easier to represent in field computers.

For another example of the use of convolution, consider the indefinite integral (Section 6.2.2), which is also shift invariant. Define the *Heaviside field* υ to be a slice through Δ^0:

$$\upsilon_s = \Delta^0_{s0} \tag{114}$$

It has the property:

$$\upsilon_s = \begin{cases} 1, & \text{if } s \geq 0 \\ 0, & \text{if } s < 0 \end{cases} \tag{115}$$

This field can be convolved with an arbitrary field to compute its indefinite integral:

$$\int \phi = \upsilon \star \phi \tag{116}$$

To see this, observe:

$$\begin{aligned} (\upsilon \star \phi)_s &= \int_\Omega \upsilon_{s-t}\phi_t \mathrm{d}t \\ &= \int_\Omega \Delta^0_{s-t,0}\ \phi_t \mathrm{d}t \\ &= \int_\Omega \Delta^0_{st}\ \phi_t \mathrm{d}t \\ &= (\textstyle\int \phi)_s \end{aligned}$$

The foregoing examples suggest that convolution is a useful operation to include in general purpose field computers.

7.2.5 Correlation

Definition If Ω is a linear measure space (Section 7.2.2), then the correlation $\phi \otimes \psi$ of two fields $\phi, \psi \in \Phi(\Omega)$ is defined:

$$(\phi \otimes \psi)_s = \int_\Omega \phi_{t-s} \psi_t \mathrm{d}t \tag{117}$$

Notice that this differs from convolution only in having '$t - s$' where the latter has '$s - t$'.

The correlation of two different fields is usually called *crosscorrelation,* whereas the correlation of a field with itself is called *autocorrelation.*

Formula

$$\phi \otimes \psi = M\phi\psi = M(\psi \wedge \phi) \tag{118}$$

where

$$M_{st} = {\Delta^1}_{ts} \tag{119}$$

Since the field M is defined in terms of the unit impulse field Δ^1 (Section 6.2.4), it must in practice be approximated.

Proof It is easiest to establish this result by direct expansion of the product:

$$\begin{aligned}
(M\phi\psi)_s &= (M\phi)_s \cdot \psi \\
&= \left(\int_\Omega M_{st}\phi_t \mathrm{d}t\right) \cdot \psi \\
&= \left(\int_\Omega {\Delta^1}_{ts}\phi_t \mathrm{d}t\right) \cdot \psi \\
&= \int_\Omega ({\Delta^1}_{ts} \cdot \psi)\phi_t \mathrm{d}t \\
&= \int_\Omega \psi_{t-s}\phi_t \mathrm{d}t \\
&= (\phi \otimes \psi)_s
\end{aligned}$$

The second to last step follows from the sifting property of Δ^1.

Part IV
Nonlinear Operators

8 Introduction

In Sections 1.4 and 1.5 we discussed the utility of general purpose field computers, and claimed that there exists a *universal set* of field operations that permits the approximation of any field transformation in a large and useful class. In this section we present one such universal set and give several examples of the resulting approximations.

9 Approximation Based on Taylor Series

9.1 Taylor's Theorem

9.1.1 Taylor's Expansion in Derivatives

We state the standard Taylor theorem from functional analysis. Like the more familiar Taylor theorem from real analysis, it permits the expansion of a function in an infinite series about a point. The difference is that in the present case the function is a field transformation, and the point is a field.

Theorem 9 *Suppose U is any open subset of $\Phi(\Omega_1)$ and $T : \Phi(\Omega_1) \to \Phi(\Omega_2)$ is a map which is C^n in U (that is, the first n derivatives of T exist). Let $\phi \in U$ and $\alpha \in \Phi(\Omega_1)$ be such that $\phi + \theta\alpha \in U$ for all $\theta \in [0,1]$. Then:*

$$T(\phi+\alpha) = \sum_{k=0}^{n-1} \frac{T^{(k)}(\phi)(\alpha)^k}{k!} + R_n(\phi,\alpha) \tag{120}$$

where

$$R_n(\phi,\alpha) = \int_0^1 \frac{(1-\theta)^{n-1} T^{(n)}(\phi+\theta\alpha)(\alpha)^n}{(n-1)!} \mathrm{d}\theta \tag{121}$$

Here '$(\alpha)^k$' denotes k occurrences of the argument α. Also note that $T^{(0)} = T$.

In uncurried form the Taylor expansion is:

$$T(\phi+\alpha) = \sum_{k=0}^{n-1} \frac{\mathrm{d}^k T(\phi,\alpha,\ldots,\alpha)}{k!} + R_n(\phi,\alpha) \tag{122}$$

where

$$R_n(\phi,\alpha) = \int_0^1 \frac{(1-\theta)^{n-1} \mathrm{d}^n T(\phi+\theta\alpha,\alpha,\ldots,\alpha)}{(n-1)!} \mathrm{d}\theta \tag{123}$$

and the appropriate number of α arguments (zero or more) must be supplied for $\mathrm{d}^k T$.

9.1.2 Taylor's Expansion in Gradients

If the first n gradients of T are defined (Section 5.5.2), then its Taylor expansion is:

$$T(\phi+\alpha) = T(\phi) + \sum_{k=1}^{n} \frac{\nabla_\alpha^k T(\phi)}{k!} + R_n(\phi,\alpha) \tag{124}$$

As shown in Section 5.5.1, the kth term can be written in any of the following forms:

$$\begin{aligned} \frac{1}{k!}\nabla_\alpha^k T(\phi) &= \frac{1}{k!}\nabla^k T(\phi)\alpha\cdots\alpha \\ &= \frac{1}{k!}\nabla^k T(\phi)\alpha^{(k)} \end{aligned}$$

where by $\alpha^{(k)}$ we mean the k-fold outer product $\alpha\wedge\alpha\wedge\cdots\wedge\alpha$. The two forms on the right are especially useful, since they separate the part of the term which is fixed by the point of expansion, $\nabla^k T(\phi)$, from the part which is variable, α.

The remainder term is:

$$R_n(\phi,\alpha) = \int_0^1 \frac{(1-\theta)^{n-1}\nabla_\alpha^n(\phi+\theta\alpha)}{(n-1)!}\mathrm{d}\theta \tag{125}$$

9.1.3 Horner's Rule Expansion

As is done for conventional polynomials, we can eliminate the need to compute higher (outer product) powers of α by using a form of "Horner's Rule." Consider the 3-term Taylor expansion:

$$T(\phi+\alpha) \approx T(\phi) + \nabla T(\phi)\alpha + \frac{1}{2}\nabla^2 T(\phi)\alpha^{(2)} \tag{126}$$

This can be written

$$T(\phi+\alpha) \approx T(\phi) + [\nabla T(\phi) + \frac{1}{2}\nabla^2 T(\phi)\alpha]\alpha \tag{127}$$

In general, define

$$Q_k(\phi,\alpha) = \nabla^k T(\phi) + \frac{1}{k+1} Q_{k+1}(\phi,\alpha)\alpha \tag{128}$$

for $k \geq 0$ where $\nabla^0 T = T$. Then the infinite Taylor expansion is given by

$$T(\phi+\alpha) = Q_0(\phi,\alpha) \tag{129}$$

The Horner's Rule expansion has direct relevance to implementation of field transformations by neural networks incorporating conjunctive synapses (sigma-pi units); see [11].

9.2 Summary of Taylor Series Approximation

We have already seen (Part III) that any reasonable linear or multilinear field transformation can be approximated by a general field product. The results of this section show that reasonable (i.e. sufficiently differentiable) nonlinear transformations can be approximated by a *field polynomials*, that is, by a local sum of general products. We conclude that the following constitute a *universal set* of operators:

1. Local sum: $(\phi + \psi)_t = \phi_t + \psi_t$
2. General product: $(\Psi \mathrm{X})_{su} = \int_\Omega \Psi_{st} \mathrm{X}_{tu} \mathrm{d}t$

(Note that the scalar and outer products, which also appear in field polynomials, are degenerate cases of the general product, and so are not strictly necessary.) Of course, practical general purpose field computers will implement a larger set of primitive operations.

9.3 General Polynomial Approximation

The Taylor series approximation of field transformations suffers from a limitation similar to that of the Taylor series approximation of real functions, namely, the approximation is only *locally good.* That is, since the Taylor series extrapolates from a fixed point, the accuracy tends to fall off rapidly with the distance from that point. This is not an important limitation if our only purpose is to establish a universal set of operations. If, however, we are interested in practical field computation, then the limitation is significant. In this case we require polynomials that satisfy some *global* criterion of goodness.

The problem of the polynomial approximation of field transformations can be put as follows. Consider the nth degree polynomial:

$$P_n(\phi) = K_0 + K_1\phi + K_2\phi^{(2)} + \cdots + K_n\phi^{(n)} \tag{130}$$

How can we choose the fields $K_0, \ldots, K_n$ so as to minimize the "distance" between P_n and the desired transformation T? The difficulty is to define an appropriate distance between field transformations. The usual development of an approximation theory presumes an inner product norm and a basis. Unfortunately, we have not found a suitable way to define an inner product on field transformations.

One way to compare field transformations is to compare their values on a finite set of input fields:

$$\delta(T, U) = \sum_{k=1}^{m} \|T(\phi_k) - U(\phi_k)\| \tag{131}$$

Although this measure is only a pseudo-metric,[7] it is nevertheless useful for a number of purposes (see [11]).

[7]For a pseudo-metric $\delta(x, y) = 0$ need not imply $x = y$.

10 Local Transformations

10.1 Theory

In this section we consider the special case of (nonlinear) *local* transformations. These are transformations in which each point of the output field is a function of the corresponding point of the input field:

$$[T(\phi)]_t = F_t(\phi_t) \tag{132}$$

We write $\overline{F}$ for the local transformation that applies F_t at each point t; thus:

$$[\overline{F}(\phi)]_t = F_t(\phi_t) \tag{133}$$

Suppose that for all $t \in \Omega$, $F_t : [a, b] \to K$. Then the type of F is

$$F : \Omega \to [a, b] \to K \tag{134}$$

and hence the type of $\overline{F}$ is:

$$\overline{F} : \Phi_{[a,b]}(\Omega) \to \Phi_K(\Omega) \tag{135}$$

Although Taylor's theorem can be used to derive the power series of a local transformation [6], a more general result is just as easy to obtain:

Theorem 10 *Suppose that $\overline{F} : \Phi_{[a,b]}(\Omega) \to \Phi_K(\Omega)$ and that the series*

$$F_t(x) = \sum_{k=0}^{\infty} \alpha_{kt} x^k \tag{136}$$

converges uniformly with respect to t and x. Then $\overline{F}$ is given by the following (L_2) convergent series:

$$\overline{F}(\phi) = \sum_{k=0}^{\infty} \alpha_k \times \phi^k \tag{137}$$

Here ϕ^k denotes the k-fold local product:

$$\begin{aligned} \phi^0 &= \mathbf{1} \\ \phi^{k+1} &= \phi \times \phi^k, \quad k \geq 0 \end{aligned}$$

We call transformations such as Eq. 137, *local* field polynomials or *local* power series, since the powers are computed by local products.

Proof: Let $\epsilon > 0$ be chosen; we must show that there is an N such that

$$\|\overline{F}(\phi) - \sum_{k=0}^{n} \alpha_k \times \phi^k\| < \epsilon \tag{138}$$

whenever $n \geq N$. Let $\zeta = \epsilon|\Omega|^{-1/2}$. Since the series for $F_t(x)$ converges uniformly, we know that there is an N, independent of t and x, such that

$$|F_t(x) - \sum_{k=0}^{n} \alpha_{kt} x^k| < \zeta \tag{139}$$

whenever $n > N$. Now consider:

$$\begin{aligned}
\|\overline{F}(\phi) - \sum_{k=0}^{n} \alpha_k \times \phi^k\|^2 &= \int_\Omega \left[\overline{F}(\phi) - \sum_{k=0}^{n} \alpha_k \times \phi^k\right]_t^2 \mathrm{d}t \\
&= \int_\Omega \left[F_t(\phi_t) - \sum_{k=0}^{n} \alpha_{kt} \phi_t^k\right]^2 \mathrm{d}t \\
&\leq \int_\Omega \zeta^2 \mathrm{d}t \\
&= \zeta^2 |\Omega| \\
&= \epsilon^2
\end{aligned}$$

Hence,

$$\|\overline{F}(\phi) - \sum_{k=0}^{n} \alpha_k \times \phi^k\| < \epsilon \tag{140}$$

□

The common case where F is a constant function, that is, $F_t = f$ for all t, is especially useful:

Corollary 6 *If $\overline{f} : \Phi_I(\Omega) \to \Phi_K(\Omega)$ and I is in the interval of convergence of*

$$f(x) = \sum_{k=0}^{\infty} a_k x^k \tag{141}$$

then this series converges:

$$\overline{f}(\phi) = \sum_{k=0}^{\infty} a_k \phi^k \tag{142}$$

Here $a_k \phi^k$ denotes a scaling *of ϕ^k by $a_k \in \Re$. If this is not a primitive operation, then it can be accomplished by a local product with the constant field $a_k\mathbf{1}$.*

Proof: It is well-known that if I is in the interval of convergence of $\sum_{k=0}^{\infty} a_k x^k$ then the series converges uniformly in I. Hence the theorem applies.

□

10.2 Examples

10.2.1 Importance of Sigmoid Nonlinearities

We begin by exploring the field computation of two local transformations with a "sigmoid" shape. Such transformations are important because of their applications in neural networks. The functional behavior of the most common artificial neurons are defined by the equation

$$y_i = \sigma\left(\sum_{k=1}^{N} W_{ij} x_j\right) \tag{143}$$

This defines the activity level y_i of an output neuron i $(1 \leq i \leq M)$ in terms of the activities x_j of the input neurons j $(1 \leq j \leq N)$. The "weights" W_{ij} reflect the strength and polarity (excitatory or inhibitory) of the synapses between neurons j and i. The sigmoid function σ is a nonlinear function whose effect is to "sharpen up" the value computed by the summation; it acts as a "soft threshold." Notice that if we think of x, y and W as finite fields, then the preceding equation can be expressed as a field computation:

$$y = \overline{\sigma}(Wx) \tag{144}$$

We are of course most interested in the case where x, y and W are continuous fields, but the mathematics is the same. In the following subsections we discuss the field computation of two common sigmoid transformations.[8]

10.2.2 Hyperbolic Tangent

Definition The hyperbolic tangent sigmoid transformation is

$$\overline{\tanh} : \Phi_{[a,b]}(\Omega) \rightarrow \Phi_{[-1,1]}(\Omega) \tag{145}$$

where $-\pi/2 < a < b < \pi/2$. Its effect as a soft threshold can be seen in its continuous variation between these values:

$$\begin{aligned} \tanh(-\infty) &= -1 \\ \tanh(0) &= 0 \\ \tanh(+\infty) &= +1 \end{aligned}$$

The tanh function is most useful when we want the nonlinearity to preserve the sign of the input. It also arises naturally in analog VLSI implementations of neural networks [12, p. 69 and passim].

[8]For neural networks there is little practical difference between different sigmoids, since the effect of scaling can be accomplished by modifying the weights, and the effect of translation by applying a constant bias to the neuron; see [11].

Formula

$$\overline{\tanh}(\phi) = \phi - \frac{1}{3}\phi^3 + \frac{2}{15}\phi^5 - \frac{17}{315}\phi^7 + \cdots + \frac{2^{2n}(2^{2n}-1)B_{2n}}{(2n)!}\phi^{2n-1} + \cdots \tag{146}$$

where B_{2n} is the $2n$-th Bernoulli number.

Proof This follows directly from Cor. 6 and the Maclauren series for tanh [14], whose interval of convergence is $(-\pi/2, \pi/2)$.

10.2.3 Logistic Function

Definition The sigmoid function most commonly used in neural networks is the *logistic function*, given by

$$\mathrm{lgst}(x) = \frac{1}{1+e^{-x/T}} \tag{147}$$

It can also be defined as a translated, scaled hyperbolic tangent:

$$\mathrm{lgst}(x) = \frac{1}{2}\tanh\left(\frac{x}{2T}\right) + \frac{1}{2} \tag{148}$$

It is a soft threshold, as can be seen from its behavior:

$$\begin{aligned} \mathrm{lgst}(-\infty) &= 0 \\ \mathrm{lgst}(0) &= 1/2 \\ \mathrm{lgst}(+\infty) &= 1 \end{aligned}$$

This simoid is most useful when the output is to be interpreted as a probability. The parameter T, commonly called "computational temperature," adjusts the slope of the sigmoid at the origin (which is $1/T$). At $T = 0$ it becomes a step function (threshold). We consider here the corresponding local transformation $\overline{\mathrm{lgst}} : \Phi_{[a,b]}(\Omega) \to \Phi_{[0,1]}(\Omega)$, where $-\pi T < a < b < \pi T$.

Formula

$$\overline{\mathrm{lgst}}(\phi) = \frac{1}{2} + \frac{1}{4T}\phi - \frac{1}{48T^3}\phi^3 + \frac{1}{480T^5}\phi^5 - \frac{17}{80640T^7}\phi^7 + \cdots \tag{149}$$

Proof The series can be derived by direct differentiation, or from the series for tanh by the relation $\mathrm{lgst}(x) = \frac{1}{2}\tanh(x/2T) + \frac{1}{2}$. Since the radius of convergence for the tanh series is $\pi/2$, the radius for lgst will be πT.

10.2.4 Replacing Local Field Polynomials by Sigmoid Functions

Sections 10.2.2 and 10.2.3 showed how a local sigmoid transformation could be computed by a local polynomial. On the other hand, artificial neurons often have a built in sigmoid function, and we can expect many field computers will provide a local sigmoid transformation. For this reason it is important to ask the converse question: When can a field polynomial be computed by a local sigmoid transformation? The following equations characterize a sigmoid centered at the origin, but with a bias of b, a slope m at the origin, and asymptotic values $b \pm \mu$:

$$\begin{aligned}
\sigma(0) &= b && (150)\\
\sigma(x) - b &= b - \sigma(-x) && (151)\\
\sigma(+\infty) &= b + \mu && (152)\\
\sigma'(+\infty) &= 0 && (153)\\
\sigma'(0) &= m && (154)\\
\sigma'(x) &\geq 0 && (155)\\
\sigma'(x) &< m, \quad x \neq 0 && (156)
\end{aligned}$$

Now consider a power series for σ:

$$\sigma(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots \tag{157}$$

Equation 150 tells us $a_0 = b$. Also, Eq. 151 tells us $\sigma(x) + \sigma(-x) = 2b$. But,

$$\sigma(-x) = a_0 - a_1 x + a_2 x^2 - a_3 x^3 + \cdots \tag{158}$$

Therefore,

$$2b = \sigma(x) + \sigma(-x) = 2b + 2a_2 x^2 + 2a_4 x^4 + \cdots \tag{159}$$

and we conclude that the power series contains only odd powers. Furthermore, by Eq. 154 and

$$\sigma'(x) = a_1 + 3a_3 x^2 + \cdots \tag{160}$$

we know $a_1 = m$. Thus the power series for a sigmoid must look like this:

$$\sigma(x) = b + mx + a_3 x^3 + \cdots \tag{161}$$

This suggests that we ask when a cubic of the form

$$f(x) = b + mx + a_3 x^3 \tag{162}$$

can be approximated by a sigmoid function. This cubic can be easily seen to satisfy Eqs. 150, 151 and 154. Since $f'(x) = m + 3a_3 x^2$, it will satisfy Eq. 156 only if $a_3 < 0$. Thus we rewrite it

$$f(x) = b + mx - ax^3 \tag{163}$$

where $a > 0$. The cubic differs from the sigmoid in its asymptotic properties (Eqs. 152 and 153), so we must restrict our attention to the portion of the cubic between its extema. To determine their location, set $f'(r) = 0$ and then since $m - 3ar^2 = 0$ we find

$$r = \pm\sqrt{m/3a} \tag{164}$$

Therefore, we will be able to approximate $f(x)$ by $\sigma(x)$ only if $|x| \leq r$, the *radius* of the cubic sigmoid. Equation 155 is satisfied within this radius.

For a specific example, suppose we want to approximate Eq. 163 by a scaled, translated hyperbolic tangent:

$$\sigma(x) = b + \mu \tanh(\nu x) \tag{165}$$

First, observe that

$$\begin{aligned} \sigma(0) &= b & (166) \\ \sigma'(0) &= \mu\nu & (167) \end{aligned}$$

Therefore, if we let $m = \mu\nu$, then f and σ will agree at 0 in their value and derivative. We need an additional condition to determine the values of μ and ν. Since the argument of x is restricted to $[-r, r]$ we require $f(r) = \sigma(r)$. Substituting into the power series for tanh yields

$$b + mr - ar^3 = b + \mu[\nu r - (\nu r)^3/3 + \cdots] \tag{168}$$

Therefore,

$$mr - ar^3 = \mu\nu r - (\mu\nu^3/3)r^3 + \cdots \tag{169}$$

If we neglect higher order terms, and let $m = \mu\nu$, then we have

$$mr - ar^3 = mr - (m\nu^2/3)r^3 \tag{170}$$

Hence, $a = m\nu^2/3$ and so $\nu = \sqrt{3a/m} = r^{-1}$. Thus, the sigmoid approximates the cubic if

$$\begin{aligned} \nu &\approx r^{-1} & (171) \\ \mu &\approx mr & (172) \end{aligned}$$

That is, the cubic

$$f(x) = b + mx - ax^3, \quad |x| \leq r = \sqrt{m/3a} \tag{173}$$

may be approximated by the sigmoid

$$\sigma(x) = b + mr \tanh(x/r) \tag{174}$$

Therefore, if we must compute the local field polynomial

$$T(\phi) = b\mathbf{1} + mL\phi - a(L\phi)^3 \tag{175}$$

then this can be accomplished by a composition of a local sigmoid and a linear transformation:

$$T(\phi) \approx \overline{\sigma}(L\phi) \tag{176}$$

This is especially important for neural networks, which generally have the effect of a linear transformation followed by a sigmoid nonlinearity. In this case the input weight matrix is $r^{-1}L$ and the output is scaled by mr and biased by b.[9]

11 Problem of High Dimensional Gradients

11.1 The Problem

Consider the kth term in the Taylor expansion:

$$\frac{1}{k!}\nabla^k T(\phi)\alpha^{(k)} \tag{177}$$

If $\alpha \in \Phi[a,b]$ is a field defined over a closed segment of the real line, then $\alpha^{(k)}$ is a field defined over a k-dimensional hypercube. Unfortunately, fields of dimension higher than 3 are, seemingly, unrealizable.[10] In other words, there seems to be no way to build a field computer that can store fields like $\alpha^{(k)}$, for $k > 3$. The immediate problem of raising α to a high power can be eliminated by alternate forms of the Taylor expansion, such as that given by Horner's Rule (Section 9.1.3), but we are left with the problem that the gradients are of high dimension. In particular, if $T : \Phi(\Omega_1) \to \Phi(\Omega_2)$, then

$$\nabla^k T(\phi) \in \Phi(\Omega_2 \times \Omega_1^k) \tag{178}$$

In general, for any field polynomial $\sum_{k=0}^{N} K_k \phi^{(k)}$ the coefficient fields are of successively higher dimension, $K_k \in \Phi(\Omega_2 \times \Omega^k)$.

There are several ways to avoid higher dimensional fields. One, which is discussed in detail in [11], is to discretize the field. A second solution is to represent one dimension by time. For example, if the field Ψ_{st} is represented by a time-varying field $(\Psi_s)_t$, then in the field product

$$(\Psi\phi)_s = \int_\Omega \Psi_{st}\phi_t \mathrm{d}t \tag{179}$$

the integral can be computed by accumulating the product $\Psi_{st}\phi_t$ over an interval of time $\Omega = [t_o, t_f]$. Of course this approach buys only one additional dimension. A

[9] Of course this is not the only way to approximate a local cubic by a sigmoid. We could for example pick μ and ν to minimize the L_2 error: $\|\sigma - f\|_2$.

[10] See [1] for an exception, however.

third solution is applicable when a higher dimensional field is *sparse*, that is, zero over most of its domain. For example, if $\Psi \in \Phi(\Omega \times \Omega)$ satisfies

$$\Psi_{st} = 0, \quad \text{if } |s - t| > \epsilon \tag{180}$$

then it may be approximately represented by the lower dimensional $\psi \in \Phi(\Omega)$:

$$\phi_s = \int_{s-\epsilon}^{s+\epsilon} \Psi_{st} \mathrm{d}t \tag{181}$$

We can replace a higher dimensional general product $\Psi\phi$ by a lower dimensional local product $2\epsilon(\psi \times \phi)$ as shown by this derivation:

$$\begin{aligned}
(\Psi\phi)_s &= \int_\Omega \Psi_{st}\phi_t \mathrm{d}t \\
&= \int_{s-\epsilon}^{s+\epsilon} \Psi_{st}\phi_t \mathrm{d}t \\
&\approx \int_{s-\epsilon}^{s+\epsilon} \psi_s\phi_t \mathrm{d}t \\
&= \psi_s \int_{s-\epsilon}^{s+\epsilon} \phi_t \mathrm{d}t \\
&\approx \psi_s \int_{s-\epsilon}^{s+\epsilon} \phi_s \mathrm{d}t \\
&= 2\epsilon\psi_s\phi_s
\end{aligned}$$

There are also various mixed strategies, and no doubt other solutions that may be useful in various situations.

11.2 Field Computation in Alternate Domains

11.2.1 Introduction

Another approach to the problem of higher dimensional gradients is to map the gradient into a lower dimensional space, and do the corresponding computations in this lower dimensional space. For example, suppose $\Omega \subset E^2$ is an appropriate subset of Euclidean two-space. Further suppose that $\Psi, \mathrm{X} \in \Phi(\Omega^2)$ are two higher dimensional (in fact four-dimensional) fields. It seems that there ought to be a continuous function $R : \Phi(\Omega^2) \rightarrow \Phi(\Omega')$, with $\Omega' \subset E^2$, that maps four-dimensional spaces into two-dimensional spaces in such a way that we can find an operation $\mathrm{IP} : \Phi(\Omega') \times \Phi(\Omega') \rightarrow \Re$ that does a four-dimensional inner product on the two-dimensional surrogates of the fields:

$$F \cdot G = \mathrm{IP}[R(\Psi), R(\mathrm{X})] \tag{182}$$

In fact, such continuous maps R exist; they are based on space-filling curves, such as Peano curves (see Section 11.3). There remains the problem of whether lower dimensional correspondents of the product operation (IP in this example) exist. In this section we show that they do.

11.2.2 Measure Preserving Change of Domain

Suppose we have a field $\Psi \in \Phi(\Omega)$ and we want to represent it by a field $\psi \in \Phi(\Omega')$. A typical motivation for this would be that Ω' is of lower dimension that Ω. Thus we want $R : \Phi(\Omega) \to \Phi(\Omega')$ such that $\psi = R(\Psi)$. The change of domain is accomplished by letting

$$R(\Psi) = \Psi \circ C \tag{183}$$

where $C : \Omega' \to \Omega$ is bijective.[11] The transformation R has the correct domain and range, and loses no information (since C is bijective). It is also easy to establish that R is linear:

$$\begin{aligned}
[R(a\Psi + b\mathrm{X})]_x &= [(a\Psi + b\mathrm{X}) \circ C]_x \\
&= (a\Psi + b\mathrm{X})_{C_x} \\
&= a\Psi_{C_x} + b\mathrm{X}_{C_x} \\
&= a(\Psi \circ C)_x + b(\mathrm{X} \circ C)_x \\
&= a[R(\Psi)]_x + b[R(\mathrm{X})]_x
\end{aligned}$$

Hence $R(a\Psi + b\mathrm{X}) = aR(\Psi) + bR(\mathrm{X})$.

We now impose an additional condition on C: we require it to be *measure-preserving*. That is, for and $S \subseteq \Omega$ and $S' = C^{-1}[S]$ (i.e., S' is the inverse image of S under C) we have

$$\int_S f_t \mathrm{d}t = \int_{S'} f(C_s) \mathrm{d}s \tag{184}$$

That is,

$$\int_S \Phi_t \mathrm{d}t = \int_{S'} [R(\Phi)]_s \mathrm{d}s \tag{185}$$

Under this assumption we find that R is an *isometry* (isometric transformation), since

$$\begin{aligned}
\|R(\Psi) - R(\mathrm{X})\|^2 &= \|R(\Psi - \mathrm{X})\|^2 \\
&= \|(\Psi - \mathrm{X}) \circ C\|^2 \\
&= \int_{\Omega'} (\Psi - \mathrm{X})^2_{C_s} \mathrm{d}s \\
&= \int_{\Omega} (\Psi - \mathrm{X})^2_t \mathrm{d}t \\
&= \|\Psi - \mathrm{X}\|^2
\end{aligned}$$

An isometry is a homeomorphism, so we've shown that under these assumptions the spaces $\Phi(\Omega)$ and $\Phi(\Omega')$ are homeomorphic (and in fact isometric). Further, since an isometry is necessarily continuous, we've shown that the transformation R is continuous.

[11] We will impose additional constraints shortly.

11.2.3 Transformations in the Alternate Domain

To complete the replacement of a domain Ω by a more convenient domain Ω', we must also replace transformations T on $\Phi(\Omega)$ by corresponding transformations T' on $\Phi(\Omega')$ so that

$$T(\Psi) = T'[R(\Psi)] \tag{186}$$

(In some cases it is more convenient to have $R[T(\Psi)] = T'[R(\Psi)]$.)

Inner Product We consider some examples, of which the simplest is the inner product. We want $\mathrm{IP} : \Phi(\Omega') \times \Phi(\Omega') \to \Re$ such that

$$\Psi \cdot \mathrm{X} = \mathrm{IP}[R(\Psi), R(\mathrm{X})] \tag{187}$$

We proceed to derive IP:

$$\begin{aligned}
\Psi \cdot \mathrm{X} &= \int_\Omega \Psi_t \mathrm{X}_t \mathrm{d}t \\
&= \int_\Omega (\Psi \circ C \circ C^{-1})_t (\mathrm{X} \circ C \circ C^{-1})_t \mathrm{d}t \\
&= \int_\Omega [R(\Psi)]_{C_t^{-1}} [R(\mathrm{X})]_{C_t^{-1}} \mathrm{d}t \\
&= \int_{\Omega'} [R(\Psi)]_s [R(\Psi)]_s \mathrm{d}s \\
&= R(\Psi) \cdot R(\mathrm{X})
\end{aligned}$$

Hence, let

$$\mathrm{IP}(\psi, \chi) = \psi \cdot \chi \tag{188}$$

and then $\Psi \cdot \mathrm{X} = \mathrm{IP}[R(\Psi), R(\mathrm{X})]$, as we might expect.

Local Transformations By an analogous derivation it is easy to see that if $\overline{f}(\Psi, \mathrm{X})$ is any local binary operation, then

$$R[\overline{f}(\Psi, \mathrm{X})] = \overline{f}[R(\Psi), R(\mathrm{X})] \tag{189}$$

Simply observe that

$$\begin{aligned}
\{R[\overline{f}(\Psi, \mathrm{X})]\}_s &= [\overline{f}(\Psi, \mathrm{X})]_{C_s} \\
&= f(\Psi_{C_s}, \mathrm{X}_{C_s}) \\
&= f\{[R(\Psi)]_s, [R(\mathrm{X})]_s\} \\
&= \{\overline{f}[R(\Psi), R(\mathrm{X})]\}_s
\end{aligned}$$

That is R commutes with local transformations.

General Product Now we consider a more useful example, the general product. Suppose

$$\begin{aligned} \Psi &\in \Psi(\Omega'' \times \Omega) \\ \mathrm{X} &\in \Psi(\Omega) \end{aligned}$$

We want to replace the product $\Psi\mathrm{X}$ by a product $\psi\chi$ in which

$$\begin{aligned} \psi &\in \Phi(\Omega'' \times \Omega') \\ \chi &\in \Phi(\Omega') \end{aligned}$$

Derive:

$$\begin{aligned} (\Psi\mathrm{X})_s &= \int_\Omega \Psi_{st}\mathrm{X}_t \mathrm{d}t \\ &= \int_\Omega [\Psi \circ (\mathrm{I}\times C) \circ (\mathrm{I}\times C)^{-1}]_{st}(\mathrm{X} \circ C \circ C^{-1})_t \mathrm{d}t \end{aligned}$$

where the direct product $(\mathrm{I}\times C)_{st} = (s, C_t)$. Notice that $(\mathrm{I}\times C)$ is an isomorphism, and let

$$S(\Psi) = \Psi \circ (\mathrm{I}\times C) \tag{190}$$

Then continue the derivation:

$$\begin{aligned} (\Psi\mathrm{X})_s &= \int_\Omega [S(\Psi)]_{s,C_t^{-1}}[R(\mathrm{X})]_{C_t^{-1}}\mathrm{d}t \\ &= \int_{\Omega'} [S(\Psi)]_{su}[R(\mathrm{X})]_u \mathrm{d}u \\ &= [S(\Psi)R(\mathrm{X})]_s \end{aligned}$$

Hence,

$$\Psi\mathrm{X} = S(\Psi)R(\mathrm{X}) \tag{191}$$

Outer Product Next we consider the computation of an altered outer product:

$$\mathrm{OP}(\phi, \psi) = R(\phi \wedge \psi) \tag{192}$$

Our goal is to compute $\mathrm{OP}(\phi, \psi)$ without generating the higher-dimensional field $\phi \wedge \psi$. The outer product is a bilinear operator, but can be expressed as a local product of two linear operators, as follows:

$$\phi \wedge \psi = (\phi \wedge \mathbf{1}) \times (\mathbf{1} \wedge \psi) \tag{193}$$

This is convenient, since R commutes with local transformations:

$$R(\phi \wedge \psi) = R(\phi \wedge \mathbf{1}) \times R(\mathbf{1} \wedge \psi) \tag{194}$$

Each of the two factors are linear operators, so they have kernels which we compute as follows (Eq. 44):

$$K_{st} = [(\delta_t \wedge \mathbf{1}) \circ C]_s = (\delta_t \wedge \mathbf{1})_{C_s} \tag{195}$$

$$K'_{st} = [(\mathbf{1} \wedge \delta_t) \circ C]_s = (\mathbf{1} \wedge \delta_t)_{C_s} \tag{196}$$

Then,

$$\mathrm{OP}(\phi, \psi) = K\phi \times K'\psi \tag{197}$$

This formula involves no higher-dimensional fields (except K and K', which can instead be provided as primitive operators).

Outer Product Power Finally, we define an altered outer-product power:

$$P_n(\phi) = R[\phi^{(n)}] \tag{198}$$

To accomplish this it will be useful to have two domain bijections:

$$C \;:\; \Omega' \to \Omega \times \Omega' \tag{199}$$

$$C_0 \;:\; \Omega' \to \Omega \tag{200}$$

and the corresponding representation transformations:

$$R(\Psi) \;=\; \Psi \circ C \tag{201}$$

$$R_0(\Psi) \;=\; \Psi \circ C_0 \tag{202}$$

Note that $R : \Phi(\Omega \times \Omega') \to \Phi(\Omega')$ and $R_0 : \Phi(\Omega) \to \Phi(\Omega')$. Then we define the powers recursively:

$$P_1(\phi) \;=\; R_0(\phi) \tag{203}$$

$$P_{n+1}(\phi) \;=\; \mathrm{OP}[\phi, P_n(\phi)], \quad n \geq 1 \tag{204}$$

where $\mathrm{OP}(\phi, \psi) = R(\phi \wedge \psi)$. Note that the computation of $P_n(\phi)$ makes use of no fields of dimension greater than $\Omega \times \Omega'$.

Field Polynomials The preceding derivations show that an arbitrary field polynomial can be implemented by products over an alternative domain as follows:

$$\sum_{n=0}^{N} K_n \phi^{(n)} = K_0 + \sum_{n=1}^{N} S(K_n) P_n(\phi) \tag{205}$$

Notice that all the fields that constitute this polynomial belong to $\Phi(\Omega)$, $\Phi(\Omega')$ or $\Phi(\Omega \times \Omega')$. In particular:

$$K_0 \;\in\; \Phi(\Omega) \tag{206}$$

$$S(K_n) \;\in\; \Phi(\Omega \times \Omega'), \quad n \geq 1 \tag{207}$$

$$P_n(\phi) \;\in\; \Phi(\Omega') \tag{208}$$

11.3 Reduction of Dimension

We now return to the problem of the reduction of dimension. We are given Ω and Ω', where Ω' is assumed to be of lower dimension than Ω. To apply the preceding results we need a measure-preserving bijection $C : \Omega' \to \Omega$. Fortunately, such functions exist, although they are a little unusual. For example, suppose that $\Omega = [0,1]^2$ and $\Omega' = [0,1]$. Then there are various space-filling curves (such as the Peano curve) $C : [0,1] \to [0,1]^2$. These functions are continuous, because they are curves, and bijections, because they are space-filling. (They are not, however, homeomorphisms, because their inverses are not continuous, a result of Brouwer's [4, Section 36, pp. 228–236].

We also require that the space-filling curve C be measure-preserving. This presents no difficulty if we take a curve C that is the limit of a sequence C_n of functions $C_n : \Omega \to \Omega_n$ that divide Ω into narrower and narrower "strips" and preserve the measure. For example, we may have $\Omega_n = [0, 1/2^n] \times [0, 2^n]$. In practice, we are not much concerned about what happens in the limit, since we have to use one of the finite approximations C_n anyway.

The foregoing discussion suggests that space-filling curves and, more generally, fractal curves may be important in the representation of higher dimensional fields. Thus it is especially intriguing that some structures in the brain seem to have a fractal geometry.

References

[1] H. John Caulfield, "Parallel N^4 Weighted Optical Interconnections," *Applied Optics 26*, 19 (1 October 1987), pp. 4039–4040.

[2] *DARPA Neural Network Study*, AFCEA International Press, 1988.

[3] Nabil H. Farhat, Demetri Psaltis, Aluizio Prata and Eung Paek, "Optical Implementation of the Hopfield Model," *Applied Optics 24* (1985), pp. 1469–1475.

[4] Felix Hausdorff, *Set Theory*, transl. John R. Aumann et al., New York: Chelsea, 1957.

[5] B. J. MacLennan, "Technology-Independent Design of Neurocomputers: The Universal Field Computer," *Proceedings, IEEE First International Conference on Neural Networks* (June 21–24, 1987), Vol. III, pp. 39–49.

[6] B. J. MacLennan, "Field Computation and Nonpropositional Knowledge," Naval Postgraduate School Technical Report *NPS52-87-040,* September 1987.

[7] B. J. MacLennan, "Field Computation: A Model of Massively Parallel Computation in Electronic, Optical, Molecular and Biological Systems," extended abstract in Proceedings of AAAI Spring Symposium, *Parallel Models of Intelligence: How Can Slow Components Think So Fast?,* Stanford, March 22–24, 1988, pp. 180–183.

[8] B. J. MacLennan, "Logic for the New AI," *Aspects of Artificial Intelligence,* J. H. Fetzer (ed.), Kluwer, Dordrecht, 1988, pp. 163–192.

[9] B. J. MacLennan, "Continuous Computation: Taking Massive Parallelism Seriously," poster presentation, Los Alamos National Laboratory Center for Nonlinear Studies 9th Annual International Conference, *Emergent Computation,* Los Alamos, NM, May 22–26, 1989. See also University of Tennessee, Knoxville, Department of Computer Science Technical Report CS-89-83, June 1989, 13 pages.

[10] B. J. MacLennan, "Outline of a Theory of Massively Parallel Analog Computation," poster presentation at IEEE/INNS *International Joint Conference on Neural Networks,* Washington, D.C., June 18–22, 1989. Abstract in proceedings, Vol II, p. 596. For full text see University of Tennessee, Knoxville, Department of Computer Science Technical Report CS-89-84, June 1989, 23 pages.

[11] B. J. MacLennan, "Field Computation: A Theoretical Framework for Massively Parallel Analog Computation, Parts V – VI, University of Tennessee, Knoxville, Department of Computer Science Technical Report, forthcoming.

[12] Carver Mead, *Analog VLSI and Neural Systems*, Addison-Wesley, 1989.

[13] R. E. Moore, *Computational Functional Analysis*, Ellis Harwood Ltd.; John Wiley & Sons, 1985.

[14] National Bureau of Standards, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, U.S. Government Printing Office, 1965.

[15] David E. Rumelhart, James L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations,* MIT Press, 1986.