Field Computation in Natural and Artificial Intelligence

Summary*

Bruce MacLennan
Computer Science Department
University of Tennessee, Knoxville
MacLennan@cs.utk.edu

I. Motivation for Field Computation

In this paper we discuss the applications of *field computation* to natural and artificial intelligence. (More detailed discussions of field computation can be found in prior publications, e.g. MacLennan 1987, 1990, 1993b, 1997.) For this purpose, a *field* is defined to be a spatially continuous arrangement of continuous data. Examples of fields include two-dimensional visual images, one-dimensional continuous spectra, two- or three-dimensional spatial maps, as well as ordinary physical fields, both scalar and vector. A *field transformation* operates in parallel on one or more fields to yield an output field. Examples include summations (linear superpositions), convolutions, correlations, Laplacians, Fourier transforms and wavelet transforms. Field computation may be nonrecurrent (entirely feed-forward), in which a field passes through a fixed series of transformations, or it may be recurrent (including feedback), in which one or more fields are iteratively transformed, either continuously or in discrete steps. Finally, in field computation, the topology of the field (that is, of the space over which it is extended) is generally significant, either in terms of the information it represents (e.g. the dimensions of the field correspond to significant dimensions of the stimulus), or in terms of the permitted interactions (e.g. only local interactions).

Field computation is a theoretical model of information representation and processing in natural and artificial systems. As a model, it is useful for describing certain natural systems and for designing certain artificial systems. The theory may be applied regardless of whether the system is actually discrete or continuous in structure, so long as it is approximately continuous. We may make an analogy to hydrodynamics: although we know that a fluid is composed of discrete particles, it is nevertheless worthwhile to treat it as a continuum for most purposes. So also in field computation, an array of data may be treated as a field so long as the number of data elements is sufficiently large to be treated as a continuum, and the quanta by which an element varies are small enough so that it can be treated as a continuous variable.

Physicists sometimes distinguish between *structural fields*, which describe phenomena that are physically continuous (such as gravitational fields), and *phenomenological fields*, which are approximate descriptions of discontinuous phenomena (e.g. velocity fields of fluids). Field computation deals with phenomenological fields in the sense that it doesn't matter whether their realizations are spatially discrete or continuous, so long as the continuum limit is a good mathematical approximation to the computational process. Thus, we have a sort of "Complementarity Principle," which permits the computation to be treated as discrete or continuous as convenient to the situation (MacLennan 1993a).

Neural computation follows different principles than conventional, digital computing. Digital computation functions by long series of high-speed, high-precision discrete operations. The degree of parallelism is quite modest, even in the latest "massively parallel" computers. We may say that conventional computation is *deep but narrow*. Neural computation, in contrast, functions by the massively parallel application of low-speed, low-precision continuous (analog) operations. The sequential length of computations is typically short (the "100 Step Rule"), as dictated by the real-time response requirements of animals. Thus, neural computation is *broad but shallow*. As a consequence of these differences we find that neural computation typically requires very large numbers of neurons to fulfill its purpose. In most of these cases the neural mass is sufficiently large (15 million neurons/cm^2) that it is useful to treat it as a continuum.

To achieve by artificial intelligence the levels of skillful behavior that we observe in animals, it is not unreasonable to suppose that we will need a similar computational architecture, comprising very large numbers of comparatively slow, low precision analog devices. Our current

VLSI technology, which is oriented toward the fabrication of only moderately large numbers of precisely-wired, fast, high-precision digital devices, makes the wrong tradeoffs for efficient, economical neurocomputers; it is unlikely to lead to neurocomputers approximating the 15 million neurons/cm^2 density of mammalian cortex. Fortunately, the brain shows what can be achieved with large numbers of slow, low-precision analog devices, which are (initially) imprecisely connected. This style of computation opens up new computing technologies, which make different tradeoffs from conventional VLSI. The theory of field computation shows us how to exploit relatively homogeneous masses of computational materials (e.g. thin films), such as may be generated by chemical manufacturing processes. We need such a theory to guide our design and use of such radically different computers.

II. Overview of Field Computation

A field is treated mathematically as a continuous function over a bounded set representing the spatial extent of the field. Typically, the value of the function is restricted to some bounded subset of the real numbers, but complex- and vector-valued fields are also useful.

Fields are required to be *physically realizable*, which places restrictions on the allowable functions. I have already mentioned that fields are continuous functions over a bounded domain that take their values in a bounded subset of a linear space. Furthermore, it is generally reasonable to assume that fields are uniformly continuous finite-energy (i.e. square integrable) functions. Among other things, these assumptions imply that fields belong to a Hilbert space of functions. (See Pribram 1991 and MacLennan 1990, 1993b for more on Hilbert spaces as models of continuous knowledge representation in the brain; see MacLennan 1990 for more on the physical realizability of fields.)

A *field transformation* is any continuous (linear or nonlinear) function that maps one or more input fields into one or more output fields. Since a field comprises an uncountable infinity of points, the elements of a field cannot be processed individually in a finite number of discrete steps, but a field can be processed sequentially by a continuous process, which sweeps over the input field and generates the corresponding output sequentially. Normally, however, a field transformation operates in parallel on the entire input field and generates all elements of the output at once. Many useful information processing tasks can be implemented by a composition of field transformations, which feeds the field(s) through a fixed series of processing stages. (One might expect sensory systems to be implemented by such feed-forward processes, but in fact we find feedback at almost every stage of sensory processing, so they are better treated as recurrent computations, discussed next.)

In many cases we are interested in the dynamical properties of fields: how they change in time. The changes are usually continuous, defined by differential equations, but may also proceed by discrete steps. As with the fields treated in physics, we are often most interested in dynamics defined by local interaction processes, although nonlocal interactions are also used in field computation (several examples are considered later). One reason for dynamic fields is that the field may be converging to some solution by a recurrent field computation; for example, the field might be relaxing into the most coherent interpretation of perceptual data, or into an optimal solution of some other problem. Alternately, the time-varying field may be used for some kind of real-time control, such as the motor control.

An interesting question is whether there can be a universal field computer, that is, a general purpose device (analogous to a universal Turing machine) that can be programmed to compute any field transformation (in a large, important class of transformations, analogous to the Turing-computable functions). In fact, we have shown (Wolpert & MacLennan submitted) that any Turing machine, including a universal Turing machine, can be emulated by a corresponding field computer, but this does not seem to be the concept of universality that is most relevant to field computation. Another notion of universality is provided by an analog of Taylor's theorem for Hilbert spaces. It shows how arbitrary field transformations can be approximated by a kind of "field polynomial" computed by a series of products between the input field and fixed "coefficient" fields (MacLennan 1987, 1990).

Adaptation and learning can be accomplished by field computation versions of many of the common neural network learning algorithms, although some are more appropriate to field computation than others. Learning typically operates by computing or modifying "coefficient fields" or connection fields in a computational structure of fixed architecture.

III. Field Computation in the Brain

A. Realization in the Brain

Computational maps are ubiquitous in the brain. For example, there are the well-known maps in somatosensory and motor cortex, in which the neurons form a topological image of the body. There are also the *retinotopic* maps in the vision areas, in which locations in the map mirror locations on the retina, as well as other properties, such as the orientation of edges. Auditory cortex contains *tonotopic* maps, with locations in the map systematically representing frequencies in the manner of a spectrum. Auditory areas in the bat's brain provide further examples, with systematic representations of Doppler shift and time delay, among other significant quantities.

In the presence of multiple stimuli, such maps typically represent the presence of all the stimuli. For example, if several tones are present in a sound, then a tonotopic map will show corresponding peaks of activity. Similarly, if there are patches of light (or other visual microfeatures, such as oriented grating patches) at many locations in the visual field, then a retinotopic map will have peaks of activity corresponding to all of these microfeatures. In this way the *form* of the stimulus may be represented as a superposition of microfeatures.

Computational maps such as these are reasonably treated as fields, and it is useful to treat the information processing in them as field computation. Indeed, since the cortex is estimated to contain at least 146,000 neurons per square millimeter (Changeux 1985, p. 51), even a square millimeter has enough neurons to be treated as a continuum, and in fact there are computational maps in the brain of this size and smaller (Knudsen et al. 1987). Even one tenth of a square millimeter contains sufficient neurons to be treated as a field for many purposes. The larger maps are directly observable by noninvasive imaging technique, such as NMR.

We refer to these fields as *axonal fields*, because the field's value at each location corresponds to the axonal spiking (e.g. rate and/or phase) of the neuron at that location. If only the rate is significant, then it is appropriate to treat the field as real-valued. If both rate and phase are significant (Hopfield 1995), then it is more appropriate to treat it as complex-valued.

Another place where field computation occurs in the brain is in the dendritic trees of neurons (MacLennan 1993a). The tree of a single pyramidal cell may have several hundred thousand inputs, and signals propagate down the tree by passive electrical processes (resistive and capacitive). Therefore, the dendritic tree acts as a large analog filter operating on the neuron's input field, which may be significant in dendritic information processing. In this case, the field values are represented by neurotransmitter concentrations, electrical charges and currents in the dendritic tree; such fields are called *dendritic fields*. They may have a complicated topology, since it is determined by the morphology of the dendritic tree over which it's spread.

Axonal and dendritic fields are comparatively dynamic, since their patterns of activity change on millisecond or faster time scales. There are also more static fields in the brain, which change on slower time scale or not at all. Examples include *connection fields* that describe patterns of connection between brain regions and *synaptic fields* that describe the transmission efficacies of masses of synapses. In the former case, we often find that the pattern of connections computes a convolution $\rho \otimes \phi$ with the input field ϕ, where the field ρ describes the common receptive field profile of all the output neurons. More generally, the connections may compute a linear transformation $L\phi = \int L(u,v)\ \phi(v)\ \mathrm{d}v$, where the kernel L of the operation is a connection field. In the case of synaptic fields, the transmitted signal is given by a pointwise product $\sigma(u)\phi(u)$ between the synaptic field σ and the input field ϕ. Connection fields and synaptic fields change comparatively slowly under the control of neurological development and learning.

B. Gabor Wavelets

There is considerable evidence (reviewed in MacLennan 1991) that images in primary visual cortex (V1) are represented in terms of Gabor wavelets, that is, hierarchically arranged, Gaussian-modulated sinusoids (equivalent to the pure states of quantum mechanics). The Gabor-wavelet transform of a two-dimensional visual field generates a four-dimensional field: two of the dimensions are spatial, the other two represent spatial frequency and orientation. To represent this four-dimensional field in two-dimensional cortex, it is necessary to "slice" the field, which gives rise to the columns and stripes of striate cortex. The representation is nearly optimal, as defined by the Gabor Uncertainty Principle (a generalization of the Heisenberg Uncertainty Principle to information representation and transmission). Time-varying two-dimensional visual images may be viewed as three-dimensional functions of space-time, and it is possible that time-varying images are represented in vision areas by a three-dimensional Gabor-wavelet transform, which generates a time-varying five-dimensional field (representing two spatial dimensions, spatial frequency, spatial orientation and temporal frequency). The effect is to represent the "optic flow" of images in terms of spatially fixed, oriented grating patches with moving fringes. (See MacLennan 1991 for more details.) Finally, Pribram provides evidence that Gabor representations are also used for controlling the generation of motor fields (see citations in MacLennan 1997, p.64).

C. Direction Fields

Another example of field computation in the brain is provided by direction fields, in which a direction in space is encoded in the activity pattern over a brain region (Georgopoulos 1995). Such a region is characterized by a vector field $\mathbf{D}$ in which the vector value at each neural location gives the preferred direction encoded by the neuron at that location. The population code for a direction $\mathbf{r}$ is proportional to the scalar field given by the inner product of $\mathbf{r}$ at each point of $\mathbf{D}$. It will have a peak at the location corresponding to $\mathbf{r}$, which falls off as the cosine of the angle between this vector and the surrounding neurons' preferred directions. (See MacLennan 1997, section 6.2, for a more detailed discussion.)

Field computation is also used in the brain for modifying direction fields. For example, a direction field representing a remembered location, relative to the retina, must be updated when the eye moves (Droulez & Berthoz 1991a, 1991b), and the peak of the direction field must move in a direction given by the velocity vector of the eye motion. The change in the direction field is given by a differential field equation, in which the change in the value of the direction field is given by the inner product of the eye velocity vector and the gradient of the direction field: $\mathrm{d}\phi/\mathrm{d}t = \mathbf{v}\cdot\nabla\phi$. Each component (x and y) of the gradient is approximated by a convolution between the direction field and a "derivative of Gaussian" (DoG) field, which is implemented by the DoG shape of the receptive fields of the neurons. (See MacLennan 1997, section 6.3, for a more detailed discussion.)

Other examples of field computation in motor control include the control of frog leg position by the linear superposition of convergent force fields generated by spinal neurons (Bizzi & Mussa-Ivaldi 1995), and the computation of convergent vector fields, defining motions to positions in head-centered space, from positions in retina-centered space, as represented by products of simple receptive fields and linear gain fields (Andersen 1995). (See MacLennan 1997, section 6, for more details.)

D. RBF Networks

One kind of field transformation, which is very useful and may be quite common in the brain, is similar to a *radial basis function (RBF) neural network*. The input field is a computational map, which encodes significant stimulus values by the location of peak activity within the field (similar to the direction fields already discussed). The transformation has two stages. The first stage is a convolution between the input field and a local field (such as a Gaussian); this "coarse codes" the stimulus as a pattern of activity. (We do not require the local field to be strictly radial, although it commonly is.) This stage is implemented by a layer of neurons with identical receptive field profiles given by the local field.

The second stage is a linear transformation of the coarse-coded field, which yields the output field; it is also implemented by a single layer of neurons. Thus the transformation is given by $L(\rho \otimes \phi)$, where ϕ is the input, ρ is the local field, and L is the linear transformation.

Notice that this transformation is linear in its input field (which does not imply, however, that it is a linear function of the stimulus values). Since, if there are several significant stimuli, the input field will be a superposition of the fields of the individual stimuli, the output will likewise be a superposition of the corresponding individual outputs. Thus this transformation supports a limited kind of parallel computation in superposition. This is especially useful when the output, like the input, is a computational map.

It has been shown (Lowe 1991, Moody & Darken 1989, Wettschereck & Dietterich 1992) that simple networks of this form are universal in an important sense, and can adapt through a simple learning algorithm. For example, as we saw for direction fields, and input vector $\mathbf{r}$ can be coded by a vector field $\mathbf{D}$ to yield a scalar field $\mathbf{r}\cdot\mathbf{D}$, which is linearly transformed $L(\mathbf{r}\cdot\mathbf{D})$. Learning proceeds by slow adaptation of the encoding vector field $\mathbf{D}$ and by fast adaptation of the kernel field L.

E. Diffusion Processes

Diffusion processes can be implemented by the spreading activation of neurons, and they can be used for important tasks, such as path planning (Steinbeck & al. 1995) and other kinds of optimization (Miller & al. 1991, Ting & Iltis 1994). In a diffusion process the rate of change of a field is directly proportional to the Laplacian of the field, $\mathrm{d}\psi/\mathrm{d}t \propto \nabla^2\psi$. The Laplacian of the field can be approximated in terms of the convolution of a Gaussian with the field, which is implemented by a simple pattern of connections with nearby neurons: $\mathrm{d}\psi/\mathrm{d}t \propto \gamma\otimes\psi - \psi$, where γ is a Gaussian field of appropriate dimension. (See MacLennan 1997 for more details.)

F. Information Fields

As previously remarked, Hopfield (1995) has proposed that in some cases the information content of a spike train is encoded in the *phase* of the impulses relative to some global or local clock, whereas the impulse *rate* reflects pragmatic factors, such as the importance of the information. Phase-encoded fields of this sort are a special case of what may be termed *information fields*. An information field represents by virtue of its form, that is, the relative magnitude and disposition of its parts; its significance is a holistic property of the field. The overall magnitude of the field does not contribute to its meaning, but may reflect the strength of the signal and thereby influence the confidence or urgency with which it is used. Thus a physical field ϕ may be factored $\phi = m\,\nu$, where $m = \|\phi\|$ is its magnitude and ν is the (normalized) information field, representing its meaning. Information fields can be identified in the brain wherever we find information processing that depends on the form of a field, but not its absolute magnitude, or where the form is processed differently from the magnitude. Information is inherently idempotent: repeating a signal does not affect its semantics, although it may affect its reliability, urgency and other pragmatic factors; the idempotency of information was recognized by Boole in his *Laws of Thought*. Of course, this independence of magnitude also characteristic of the quantum field, which has led Bohm & Hiley (1993) to characterize this field as *active information.*

IV. Field Computing Hardware

Field computation can, of course, be performed on conventional digital computers or by special-purpose, but conventional digital hardware. However, as noted previously, neural computation and field computation are based on very different tradeoffs from traditional computation, which creates the opportunity for new computing technologies better suited for neural computation and field computation (which is broad but shallow). The ability to use slow, low precision analog devices, imprecisely connected, compensates for the need for very large numbers of computing elements. These characteristics suggest optical information transmission and processing, in which fields are represented by optical wavefronts. They also suggest molecular processes, in which fields are represented by spatial distributions of molecules of different kinds or in different states (e.g. bacteriorhodopsin). Practical field computers of this kind will probably combine optical, molecular and electrical processes for various computing purposes.

Mills (1995) has designed and implemented *Kirkhoff machines*, which operate by diffusion of charge carriers in bulk silicon. This is a special purpose field computer which finds the steady state defined by the diffusion equation with given boundary conditions. Mills has applied it to a number of problems, but its full range of application remains to be discovered.

To date, much of the work on quantum computing has focused on quantum mechanical implementation of binary digital computing. However, field computation seems to be a more natural model for quantum computation, since it makes better use of the full representational potential of the wave function. Indeed, field computation is expressed in terms of Hilbert spaces, which also provides the basic vocabulary of quantum mechanics. Therefore, since many field computations are described by the same mathematics as quantum phenomena, we expect that quantum computers may provide direct, efficient implementations of these computations. Conversely, the mathematics of some quantum-mechanical processes (such as computation in linear superposition) can be transferred to classical systems, where they can be implemented without resorting to quantum phenomena. This can be called *quantum-style computing*, and it may be quite important in the brain (Pribram 1991).

V. Concluding Remarks

In this summary I have attempted to provide a brief overview of field computation, presenting it as a model of massively parallel analog computation, which can be applied to natural intelligence, implemented by brains, as well as to artificial intelligence, implemented by suitable field computers. It is my hope that this summary will entice the reader to look at the more detailed presentations listed in the references and perhaps to explore the field computation perspective.

VI. References

Andersen, R. A. (1995). Coordinate transformation and motor planning in posterior parietal cortex. In M. S. Gazzaniga (ed.), *The Cognitive Neurosciences*, MIT Press, pp. 519-532.

Bizzi, E. & Mussa-Ivaldi, F. A. (1995). Toward a neurobiology of coordinate transformation. In M. S. Gazzaniga (ed.), *The Cognitive Neurosciences*, MIT Press, pp. 495-506.

Bohm, D. & Hiley, B. J. (1993). *The undivided universe: An ontological interpretation of quantum theory*. Routledge.

Changeux, J.-P. (1985). *Neuronal Man: The Biology of Mind*, tr. by L. Garey. Oxford.

Droulez, J. & Berthoz, A. (1991a). The concept of dynamic memory in sensorimotor control. In D. R. Humphrey & H.-J. Freund (eds.), *Motor Control: Concepts and Issues*, Wiley, pp. 137-161.

Droulez, J. & Berthoz, A. (1991b). A neural network model of sensoritopic maps with predictive short-term memory properties. *Proc. National Acad. Science USA* **88**: 9653-9657.

Georgopoulos, A. P. (1995). Motor cortex and cognitive processing. In M. S. Gazzaniga (ed.), *The Cognitive Neurosciences*, MIT Press, pp. 507-517.

Hopfield, J. J. (1995). Pattern recognition computation using action potential timing for stimulus representation. *Nature* **376**: 33-36.

Knudsen, E. J., du Lac, S. & Esterly, S. D. (1987). Computational maps in the brain. *Ann. Rev. of Neuroscience* **10**: 41-65.

Lowe, D. (1991). What have neural networks to offer statistical pattern processing? *Proc. SPIE Conf. on Adaptive Signal Processing*, San Diego, pp. 460-471.

MacLennan, B. J. (1987). Technology-independent design of neurocomputers: The universal field computer. In M. Caudill & C. Butler (eds.), *Proc. IEEE First Intl. Conf. on Neural Networks*, IEEE Press, Vol. 3, pp. 39-49.

MacLennan, B. J. (1990). Field computation: A theoretical framework for massively parallel analog computation, parts I-IV. *Tech. Rep. CS-90-100*. Comp. Sci. Dept., Univ. of Tennessee, Knoxville. Accessible via URL `http://www.cs.utk.edu/~mclennan`.

MacLennan, B. J. (1991). Gabor representations of spatiotemporal visual images. *Tech. Rep. CS-91-144*. Comp. Sci. Dept., Univ. of Tennessee, Knoxville. Accessible via URL `http://www.cs.utk.edu/~mclennan`.

MacLennan, B. J. (1993a). Information processing in the dendritic net. In K. Pribram (ed.), *Rethinking Neural Networks: Quantum Fields and Biological Data*, Lawrence Erlbaum, pp. 161-197.

MacLennan, B. J. (1993b). Field computation in the brain. In K. Pribram (ed.), *Rethinking Neural Networks: Quantum Fields and Biological Data*, Lawrence Erlbaum, pp. 199-232.

MacLennan, B. J. (1997). Field computation in motor control. In P. Morasso & V. Sanguineti (eds.), *Self-Organization, Computational Maps, and Motor Control,* Elsevier, pp. 37-73.

Miller, M.I., Roysam, B., Smith, K. R. & O'Sullivan, J. A. (1991). Representing and computing regular languages on massively parallel networks. *IEEE Trans. Neural Networks* **2**: 56-72.

Mills, J. W. (1995). Kirkhoff machines. Technical report, Comp. Sci. Dept., Indiana Univ., Bloomington.

Moody, J. E. & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation* **1**: 281-294.

Pribram, K. H. (1991). *Brain and perception: Holonomy and structure in figural processing*. Lawrence Erlbaum.

Steinbeck, O., Tóth, A. & Showalter, K. (1995). Navigating complex labyrinths: Optimal paths from chemical waves. *Science* **267**: 868-871.

Ting, P.-Y. & Iltis, R. A. (1994). Diffusion network architectures for implementation of Gibbs samplers with applications to assignment problems. *IEEE Trans. Neural Networks* **5**: 622-638.

Wettscherick, D. & Dietterich, T. (1992). Improving the performance of radial basis function networks by learning center locations. In J. E. Moody, S. J. Hanson & R. P. Lippmann (eds.), *Advances in Neural Information Processing Systems 4*, Morgan Kaufmann, pp. 1133-1140.

Wolpert, D. H. & MacLennan, B. J. (submitted). A computationally universal field computer which is purely linear. See also Santa Fe Institute TR 93-09-056 and Univ. of Tennessee, Knoxville, Comp. Sci. Dept. TR CS-93-206. Accessible via URL `http://www.cs.utk.edu/~mclennan`.