Neural, Parallel & Scientific Computations 6 (1998) 297-336

# A LAGRANGIAN FORMULATION OF NEURAL NETWORKS I: THEORY AND ANALOG DYNAMICS

Eric Mjolsness[1] and Willard L. Miranker[2]

[1]Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena CA 91109-8099

[2]Department of Computer Science
and Neuroengineering and Neuroscience Center
Yale University
New Haven CT 06520

and
Research Staff Member, Emeritus,
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

**ABSTRACT:** We expand the mathematical apparatus for relaxation networks, which conventionally consists of an objective function $E$ and a dynamics given by a system of differential equations along whose trajectories $E$ is diminished. Instead we (1) retain the objective function $E$, in a standard neural network form, as the measure of the network's computational *functionality*; (2) derive the dynamics from a Lagrangian function $L$ which depends on both $E$ and a measure of computational *cost*; and (3) tune the form of the Lagrangian according to a *meta-objective* $\mathcal{M}$ which may involve measuring cost and functionality over many runs of the network. The key new features are the Lagrangian, which specifies an objective function that depends on the neural network's state over all times (analogous to Lagrangians which play a similar fundamental role in physics), and its associated *greedy functional derivative* from which neural-net relaxation dynamics can be derived. It is the greedy variation which requires the dissipation critical to optimization with neural dynamics.

With these methods we are able to analyze the approximate optimality of Hopfield/Grossberg dynamics, the generic emergence of sub-problems involving learning and scheduling as aspects of relaxation-based neural computation, the integration

Received November 08, 1997 1061-5369/96 $03.50 

of relaxation-based and feed-forward neural networks, and the control of computational *attention mechanisms* using priority queues, coarse-scale blocks of neurons, default-valued neurons, and other special-case optimization algorithms. Some of these applications are the subject of part II of this work.

In part II of this work we show that the combination of Lagrangian and meta-objective suffice to derive and provide an interpretation for so-called *clocked objective functions*, a notation useful for the algebraic formulation and design of ramified neural network applications. Clocked objectives thus generalize the original static objective function $E$ and furnish a practical neural network specification language.

# 1 INTRODUCTION

Optimization is a prominent way to bring mathematical methods to bear on the design of neural networks. Often the connection is made [Hop84, Gro88, HT85] by specifying the attractors of a neural network's dynamics by means of a static objective function (or *objective*) to be optimized, provided that the optimization problem can be put in a standard neural-net form (which is not too restrictive a requirement [MG90]). In this way it has proven possible to design neural networks for applications in image processing [KMY86], combinatorial optimization [DW87], clustering [RGF90, BK93], particle tracking in accelerators [YHP91], object recognition [Tre91] and other applications. It is also customary (albeit limiting) to introduce a generic steepest-descent dynamics to optimize or "relax" the objective, without further regard to computational constraints. The resulting equations of motion generally contain gradients of the static objective, but are otherwise ad hoc and not particularly suited to elaboration or refinement in response to varied computational constraints. We shall develop a more general approach, starting from basic principles, to formulating the dynamics of a relaxation-based neural network.

Here *we start from fundamental computational considerations which, we hypothesise, constrain all dynamical systems that compute.* Specifically, the cost and functionality (efficacy) of a computation are fundamental to its design, and in general each must be traded off against the other in the course of optimizing that design. (Here the "design" is all the information which directly specifies the structure or configuration of the dynamical system that performs a computation.) In the context of neural computations, we will find measures of cost and functionality and combine them into *dynamical objective functions* from which one may derive the entire dynamics of a neural network. This dynamics includes not only the (fixed point) attractors but also the equations of motion governing convergence to an attractor, i.e. a mathematical model or specification of the network itself.

Our dynamical objective functions can be specialized in many ways that correspond to the wide variety of goals and constraints that may be imposed on a computation. We will also relate the dynamical objective functions to a so-called *Lagrangian functional.* Our Lagrangian is analogous to one which plays a similar and fundamental role in physics. A basic constraint which we impose on our approach is that such a dynamical objective function or Lagrangian is optimized in a special way, by means of *greedy algorithms* which don't look ahead in time. This constraint allows our algorithms to be implemented in physical hardware, and also allows us to derive nonconservative, irreversible dynamics which can lead to a desired fixed point.

We will derive these algorithms by means of a novel *greedy variation* applied to the Lagrangian functional.

Generally we will accept the limited type of optimization that results, but sometimes we can do better by introducing another level of optimization: a *meta-optimization* problem in which the (analytic) form of the dynamic objective (the Lagrangian functional) is itself varied so as to optimize another objective function. This latter optimization may involve measuring cost and functionality over many runs of the network. This meta-optimization problem determines the choice of the exact algebraic form of the Lagrangian and hence of the computational dynamics for a whole class of applications. So for a meta-objective function, cost and functionality are measured over a class of computational problems rather than over a single instance of that class as would be the case for a Lagrangian functional. In practice the computational cost or analytic effort required to perform the meta-optimization is to be amortized over many problem instances. One example of this approach will be a (meta-) optimality objective for Hopfield/Grossberg dynamics [Hop84, Gro88], for which we provide a proof that the associated Lagrangian is optimal in an approximate sense.

## 1.1 Cost and Functionality

Consider a physical system capable of nontrivial computation. More abstractly, consider a discrete, continuous or mixed dynamical system which computes, in the sense that it models a computational device or framework. Examples include a general-purpose computer equipped with suitable programs, a discrete data structure implemented by means of such a program, an individual silicon chip, or an animal brain. Such devices have detailed dynamics, often approximable as large sparsely coupled systems of ordinary differential equations, which have been designed (or evolved in the case of a brain) to serve some set of computational purposes at feasible cost. So we refer to these dynamical systems as *computational systems* and hypothesize very broadly that *fundamentally, a computational system is designed (or evolved) to optimize two things:* its *cost* and its *functionality*. *Functionality* means what the system can do, and *cost* means how cheaply or quickly it can do it.

For example, the design of silicon chips is largely constrained by the use of chip area and cycle time as the measures of cost, and the need to attain at least a minimal level of functionality to make the chip generally useful (e.g. to implement an adequate instruction set in a CPU chip); tradeoffs between minimization of chip area and maximization of detailed functionality are frequent in the design process. For another example we refer to the implementation of abstract data structures such as priority queues, for which a functionality specification requires that a small set of operations (such as adding a prioritized element to a queue and removing the element with highest priority from the queue) must be supported, and cost is conventionally characterized by an asymptotic scaling rule for the time-cost of performing a worst-case mix of these operations on a very large queue.

For a relaxation-based neural net which is programmed or designed to optimize a static objective function $E(\mathbf{x})$ from an arbitrary starting point $\mathbf{x}_{\text{initial}}$, typical expressions for cost $C$ and functionality $F$ might be

$$C = \text{4-Volume of the Net} = \text{Space} \times \text{Time} \tag{1}$$

and

$$F = E(\mathbf{x}_{\text{final}}) - E(\mathbf{x}_{\text{initial}}). \tag{2}$$

The space-time product is familiar in computer science as an important measure of cost, in which the Space term is a volumetric measure of hardware usage such as chip area (including on-chip wires) or memory usage, and the Time term is likewise a computational version of physical time such as the number of clock cycles required to complete a computation. (A specific volumetric measure of wiring cost for circuit implementations of neural nets has been proposed in [Mjo85].) As to functionality, the use of an objective function $E$ is a common way to measure progress (hence functionality) in a wide variety of computational problems. For example, one can fit a piecewise-constant model to a 2-d image given by the data $\{d_{ij}\}$, segmenting it into roughly constant regions, with the objective function [KMY86]

$$\begin{aligned} E(f, s^h, s^v) &= \frac{A}{2}\sum_{ij}(f_{ij} - d_{ij})^2 + \frac{B}{2}\sum_{ij}(f_{i+1,j} - f_{ij})^2(1 - s^v_{ij}) \\ &\quad + \frac{B}{2}\sum_{ij}(f_{i,j+1} - f_{ij})^2(1 - s^h_{ij}) + \mu\sum_{ij}(s^h_{ij} + s^v_{ij}), \end{aligned} \tag{3}$$

where $f_{ij} \in \Re$ is a reconstructed version of the image, and $s^{h,v}_{ij} \in \{0, 1\}$ represent discrete decisions concerning the probable presence or absence of horizontal and vertical edges. $f$ and $s$ together constitute the vector x appearing in equation (2). This kind of objective has been used to derive functional neural networks for large-scale problems ($10^5$ neurons with $10^6$ connections) as required for image-processing [RC91, KMY86].

## 1.2 Outline

We (a) introduce a three-level optimization framework, concentrating on Lagrangians (of a type relevant to computation) and their specialization to clocked objective functions (section 2); (b) apply the framework to derive analog circuits such as those modeled by the Hopfield/Grossberg dynamics for optimization (section 3); and (c) apply the framework to incorporate computational attention mechanisms (similar to saccading and foveation in biological vision) into various dynamical systems which are designed to solve optimization problems (section 2 of Part II).

Section 2 introduces the three-level optimization framework, beginning with the general form of a Lagrangian suitable for use in attractor dynamics for optimization problems. The greedy functional derivative is defined and calculated for such Lagrangians (sections 2.1 and 2.2). The strategy used to design circuit-implementable Lagrangians is one of *refinement* (section 2.3), in which cost and functionality measures are first defined at a coarse temporal scale and then refined for use at finer time scales, down to the infinitesimal time scale suitable for dynamical systems that model analog circuits. The validity of the transformations required during refinement is ultimately specified by a meta-objective function which measures network performance. One circuit-implementable form of Lagrangian is introduced in sections 2.2 and 2.3, though not completely derived until section 3.2, and it is illustrated by the concrete example of Hopfield/Grossberg dynamics for a region-segmentation neural network. A more general circuit-implementable form of Lagrangian, which allows network dynamics to be controlled by a repeating cycle of objective functions rather than a single objective function, is introduced in section 2.1 of Part II.

where it is illustrated by an algorithm similar to line minimization. This type of Lagrangian gives rise to the practical *clocked objective function* and *clocked sum*

notation of sections 2.1.2 and 2.1.3 of Part II, whose theoretical justification requires all three levels of optimization: the objective $E$, the Lagrangian $L$, and the meta-objective $\mathcal{M}$.

Section 3 is devoted to the study of circuit-level Lagrangians with continuous time dynamics and analog-valued neurons. Two novel possibilities for such Lagrangians are discussed in sections 3.1.1 and 3.1.2. In section 3.2 a simple meta-optimality criterion for a limited class of analog circuit Lagrangians is presented. Since this constrained meta-objective function $\mathcal{M}_\tau$ is a function of the fastest and slowest physical time scales in various circuits, it is invariant with respect to monotonic, coordinatewise reparameterizations (changes of variable) of the circuit.

In sections 3.2.1, 3.2.2, and 3.2.3 we prove Theorem 1, which asserts that the Lagrangian $L$ corresponding to Hopfield/Grossberg dynamics yields a value of $\mathcal{M}_\tau[L]$ which is within a factor of two of the optimal value of $\mathcal{M}_\tau$. This means, roughly, that the worst-case time constant for this Lagrangian $L$ is at most twice that of the optimal Lagrangian $L^*$, whatever that is. The proof exploits a sharp global optimality result for Hopfield/Grossberg dynamics (Lemma 1 of section 3.2.2). Unlike $\mathcal{M}_\tau$, the optimized functional of Lemma 1 does depend on the coordinate system chosen. A number of limitations of Theorem 1 are discussed. The resulting Lagrangian for analog circuits can be generalized to clocked objective functions, as discussed in section 2.1.5 of Part II. Section 2.1.6 of Part II provides an instructive example: a clocked objective function which incorporates one or more general feed-forward neural networks (for which relatively efficient learning algorithms are available) inside a general relaxation neural network.

In section 2 of Part II we show how simple cost constraints can lead to a variety of computational *attention mechanisms* analogous to virtual memory protocols in present-day computers, and an associated Lagrangian or clocked objective function to control each attention mechanism. Examples of possible foci of attention include a subset of the $n$ (out of $N$) neurons with highest estimated improvement in functionality $|\Delta E|$, which may be tracked efficiently by means of a priority queue data structure (section 4.1 of Part II ); a subset of course-scale blocks in a minimal partition of the neurons, scheduled by their estimated individual and pairwise contributions to $|\Delta E|$ (section 4.2 of Part II ); a set of rectangular windows in a two-dimensional network, each of which can either "jump" or "roll" to a new location (section 4.3 of Part II ); a subset of neurons in a sparsely active network including all neurons which don't have prescribed default values and hence do require storage space (section 4.4 of Part II ); and a subset of neurons determined as the Cartesian product of several simpler foci of attention (section 4.5 of Part II ). The designs presented in section 2 of Part II are theoretically well-motivated but may need to be revised in the light of subsequent experimentation, which is beyond the scope of the present paper.

Finally, a brief summary of our work is given in the concluding section 4.

## 2 DYNAMICAL OBJECTIVE FUNCTIONS AND LAGRANGIANS

We have argued that fundamentally, a computing system is designed by trading off two competing utilities: its *cost* of operation and its *functionality*. We may specify a fixed allowable cost and seek to obtain maximal functionality, or we may specify a

fixed functionality and seek to obtain a minimal cost, or we may seek a specified trade-off between cost and functionality. We may specify further dynamical *constraints* required for implementability. With Lagrange multipliers and/or penalty terms we may reduce all these cases to extremizing

$$S = AC_{\text{cost}} + BF_{\text{functionality}}, \tag{4}$$

where the system is more *functional* for lower values of $F$, and where any dynamical constraints have been absorbed into the $C_{\text{cost}}$ term. Now the designer's problem is to find functions $C$ and $F$ (perhaps based on equations (1) and (2)) which depend on the trajectory of some vector of state variables $\mathbf{x}(t)$ over time, such that the global optimization of $S$ can be reduced to a collection of *local decisions* about how to change the individual components of the state vector $\mathbf{x}$ at a given small time step from time $t - \Delta t$ to time $t$. (A local decision could be viewed as the choice of the value of a variable (e.g. a control variable).) These decisions must however be made by very simple physical devices such as transistor circuits containing only a few transistors. Such local decisions will prove to be analogous, in a physical system, to a differential or difference equation formulation of dynamics that follows from the principle of least action for the same system.

For example, it would be advantageous if $C$ and $F$ were each sums (or integrals) over a collection of decisions spread out over space and time. To express this summation, let us index the components of the state vector $\mathbf{x}$ by an index $s$. Since $s$ indexes all the variables present at a fixed time, those variables could be viewed as being embedded in one fixed-time slice of a space-time volume, in which case $s$ may also be viewed as indexing spatial locations in the system. So we refer to $s$ as the *spatial index* and $t$ as the *temporal index*; the entire trajectory of a computation is specified by $\{x(s,t)\}$. Then the sum over decisions would be

$$S = A \sum_{\text{decisions}(s,t)} C_{s,t}(\{x(s',t')\}) + B \sum_{\text{decisions}(s,t)} F_{s,t}(\{x(s',t')\}) \tag{5}$$

where each function $C_{s,t}$ or $F_{s,t}$ may depend on only a few of its arguments $\{x(s',t')\}$ and hence on only a small part of the trajectory near $(s,t)$. In equation (5) we may introduce a continuous time axis by replacing the temporal sums by integrals; we can do this by integrating over $t$ and summing over $s$. Following the analogy with physics, $S$ is referred to as the "action". The decomposition (5) would be a useful first step towards enforcing spatial and temporal *locality* on the dynamics of our computation, since the decomposition distributes $S$ over a sum of terms which pertain to particular spatial and temporal locations. Unlike space, time has an intrinsic directionality, and we will also need to enforce *causality* in the optimization of $S$. Before seeking specific forms for $C_{s,t}$ and $F_{s,t}$, we will discuss locality and especially causality.

A pattern of *communication* is implicit in the dependence of $C_{s,t}$ and $F_{s,t}$ on $x(s',t')$. If $C_{s,t}$ and $F_{s,t}$ were each a function only of $x_{s,t}$, rather than a functional of the entire state vector $\mathbf{x}(t')$ at many different times $t'$, then every decision term could be optimized independently, and the associated computation would proceed without any communication. This is a trivial case, however, and generally we will have quite a bit of interaction (via specific $C$ and $F$ terms) between variables defined at different times and places. (For a non-trivial example see the region-segmentation Lagrangian of section 2.1.2.) The pattern of communication is defined by a communication graph

whose nodes are space-time sites $(s,t)$ and whose links record the presence or absence of functional dependencies of $C_{s,t}$ or $F_{s,t}$ on trajectory variables $x$ defined at other space-time sites $(s',t')$. We want to keep this implicit pattern of communication relatively local, and we insist that it be *causal*.

The effect of causality on the communication pattern is twofold. (i) Causality favors the adoption of a *convention* in which interactions between variables indexed by different times are entirely incorporated in the $C$ and $F$ terms indexed by the later of the two times, and do not enter into the $C$ and $F$ terms defined at the earlier of the two times. That way, every $C_t$ or $F_t$ term depends only on variables indexed by times $t' \leq t$. This is called the *retarded interaction form* of $S$. (ii) If we introduce computational dynamics by sequential optimization, at successive time steps $t'$ of sets of variables indexed by $t'$, then causality denies a computation the possibility of optimizing all terms of $S$ with respect to any one variable $x(s',t')$. Instead, each variable $x(s',t')$ can only be varied under an objective involving those terms of $S$ all of whose variables $x(s'',t'')$ are optimized at the same time as $x(s',t')$ or earlier. The values of all other variables (those indexed by $t'' > t'$) are as yet undetermined. Which terms of $S$ are eligible to participate in the variation of $x(s',t')$? Any $C_t$ or $F_t$ term for which $t > t'$ depends on variables (such as $x(s,t)$) which have unknown values at time step $t'$ and are not being varied at that time step. Such a term is is ineligible; so we are restricted to those terms of $S$ indexed by time $t \leq t'$.

Note that the eligible terms of $S$ with $t \leq t'$ are mostly irrelevant to the optimization of $x(s',t')$, since point (i) imples that the $t < t'$ terms do not contain the variable $x(s',t')$. This leaves only the $t = t'$ terms of $S$ to determine $x(s',t')$.

Of course, an acausal optimizer could achieve a better value for $S$ by being less "greedy" (increasing present $C_t + F_t$ terms to decrease future ones by a greater amount), but as argued above *causality forces our dynamics to be greedy*. In other words, the causality constraint only permits a partial or *greedy optimization* of $S$, and the nature of the partial optimization depends on the decomposition of $S$ into a sum over decisions of causally constrained terms. This basic limitation to causal or *greedy dynamics* will be more or less severe depending on which of many possible decompositions of $C$ and $F$ over time is chosen.

We shall define the *greedy derivative* of $S$ with respect to $x(s',t')$ as being the ordinary derivative of the sum of such eligible $(t \leq t')$ terms of $S$, and use that derivative to define optimality of $x(s',t')$. But this greedy derivative immediately simplifies due to the retarded interaction form of $C$ and $F$:

$$\frac{\partial_G}{\partial_G x(s',t')} \sum_t (AC_t + BF_t) \equiv \frac{\partial}{\partial x(s',t')} \sum_{t \leq t'} (AC_t + BF_t) = \frac{\partial}{\partial x(s',t')} (AC_{t'} + BF_{t'}). \quad (6)$$

How can we find functions $C(\mathrm{x}\{t'\})$ and $F(\mathrm{x}\{t'\})$ that specify (via optimization of $S$) an entire computational task and yet break up into a sum over easily computed decisions? This is a statement of the problem of algorithm design, for which there is no general answer, but we can still invent some fairly general techniques. The cost function can be regarded as some kind of space-time volume to be minimized (e.g. circuit size times the duration of its use) and can be decomposed into a sum of space-time volumes for the many elementary decisions or state changes, at individual

locations and times, that comprise the associated computation:

$$C = \text{Vol} = \sum_{s,t} \delta\text{Vol}_{s,t}. \tag{7}$$

Also the functionality $F(\mathbf{x}\{t'\})$ is often measured by some definite objective function $E(\mathbf{x})$, such as total tour length in a traveling salesman problem [HT85], and this can be decomposed over time as (cf. equation (2))

$$F(\mathbf{x}_{\text{final}}) = E(\mathbf{x}_{\text{final}}) - E(\mathbf{x}_{\text{initial}}) = \sum_t \Delta E\Big|_{t-\Delta t}^{t}. \tag{8}$$

For example, a standard form for analog neural networks' objectives $E$ is [MG90]:

$$E(\mathbf{v}) = -\frac{1}{6}\sum_{ijk} T_{ijk} v_i v_j v_k - \frac{1}{2}\sum_{ij} T_{ij} v_i v_j - \sum_i h_i v_i + \sum_i \phi_i(v_i), \tag{9}$$

which encompasses many network designs including equation (3). Here $v$ takes the place of $x$, and the indices $i$, $j$, and $k$ take the place of $s$. In equation (9), $v_i$ is the output value of neuron $i$; $T_{ij}$ and $T_{ijk}$ are connection weights between two and three neurons, respectively; $h_i$ is a bias input to neuron $i$; and $\phi(v_i)$ is the potential function for neuron $i$ and determines the transfer function $g_i$ (e.g. a sigmoid function) through

$$v_i = g_i(u_i) \quad \text{and} \quad u_i = \phi'(v_i). \tag{10}$$

Often equation (9) is further specialized by setting $T_{ijk} = 0$.

As a complete example of a dynamical objective function we present, in the following equation (11), a dynamical objective for the Hopfield/Grossberg dynamics of an analog circuit. This dynamical objective will be derived in sections 2.1 and 3.2, using the fact (to be established in section 2.2) that, for a continuous-time analog circuit model, a condition for the greedy optimization takes the form of a (functional) derivative $\delta/\delta\dot{v}$ (where $\dot{v}_i = dv_i/dt$). The dynamical objective is

$$S[\dot{\mathbf{v}}(t), \mathbf{v}(t)] = \int dt \sum_i \left( K[\dot{v}_i, v_i] + \frac{\partial E}{\partial v_i}\dot{v}_i \right), \tag{11}$$

where $K[\dot{v}, v]$ is a cost-of-movement term to be derived in section 3 (see Theorem 1). Varying with respect to $\dot{v}_i$ and making use of the form of $E$ given by equation (9), we will find analog neural-net equations of motion as expected:

$$\tau_H \dot{u}_i = \sum_{jk} T_{ijk} v_j v_k + \sum_j T_{ij} v_j + h_i \quad \text{and} \quad v_i = g(u_i) \ . \tag{12}$$

Here $\tau_H$ is a time constant. The dynamical objective function $S$ of equation (11) can be recognized as an instance of (5) by identifying the neuron index $i$ with the space index (i.e. component index) $s$ and the time integral $\int dt$ with the temporal sum $\sum_t$; also $C_{st} \to K[\dot{v}_i(t), v_i(t)]$ and $F_{st} \to (\partial E[\mathbf{v}(t)]/\partial v_i)\dot{v}_i(t)$.

There is a close analogy between equation (5) and standard ideas and terminology in physics. The *action*, $S$, can be decomposed into the temporal sum (in physics, an

integral) of a Lagrangian $L(t)$ which in turn is a spatial sum of a *Lagrangian density* $L_{s,t} = C_{s,t} + F_{s,t}$:

$$\begin{aligned} S &= \sum_t L(t) \\ &= \sum_{(s,t)} L_{s,t}(\{\mathbf{x}(t')\}) = \sum_{(s,t)} (C_{s,t} + F_{s,t}) \ . \end{aligned} \tag{13}$$

(Note that the sum over time may become an integral when we consider time steps of infinitesimal duration, since the extra factor of $\Delta t$ required to get an integral is just a constant that doesn't affect the solution to an optimization problem.) For our neural network design purposes the Lagrangian $L$ is generally the most useful of these alternative notations, particularly for algebraic manipulation, because the temporal sum has the same algebraic form from one problem to the next (and hence is uninformative), but the spatial sum does not.

Extremization of such functions (or functionals) provides a foundation for the study of many dynamical systems including quantum field theories. $F$ and $C$ might with lower confidence be identified as classical kinetic energy and potential energy terms respectively, but as we will see, many details are different. These differences prevent a literal-minded mapping of our ideas and constructs onto the formalism of physics. In particular, causality is not built into physical theories by means of the partial optimization of $S$, but in a completely different way that is inconvenenient for treating irreversible dynamics such as our computations; therefore neither the dynamics nor the Lagrangians of physics can be called "greedy" in the sense we use the term.

There are a number of other ways to derive dissipative dynamics from Lagrangians, as summarized in [VJ89]. Allowing explicit time dependence, such as an overall factor of $e^{\mu t}$, in a conventional Lagrangian permits physically damped second-order dynamics to be derived. The strategy of the approach is to start with a differential equation, derive an associated Lagrangian (this is called the inverse problem of the calculus of variations, and it may have many solutions), and use that Lagrangian to analyse or approximate the solutions of the differential equation. Our strategy and methods differ, since the Lagrangians are obtained from cost and functionality considerations and hence are known before the differential equations are known. Moreover these Lagrangians require an unconventional variational principle (the greedy variation) to produce acceptable differential equations. Nevertheless there may exist some deeper relationships between our greedy Lagrangians and previous approaches discussed in [VJ89].

## 2.1 Cost and Functionality Terms

Equation (8) for $F$ is particularly appropriate for a net whose dynamics is intended to converge to fixed points that encode the answer to a static optimization problem, such as the standard neural network form of (9). Equation (8) represents a substantial specialization from the general set of functions $F_t(\{x(s',t')\}) = \sum_s F_{s,t}(\{x(s',t')\})$ that appears in (5). For in equation (8), $F_t$ depends on $t$ only through its arguments and not through its subscript, so that the algebraic form of $F_t$ is independent of time (i.e. $F_t$ is autonomous):

$$F_t(\{x(s',t')|t' \leq t\}) = E[\mathbf{x}(t)] - E[\mathbf{x}(t-\Delta t)]. \tag{14}$$

In the simplest case of static special-purpose neural circuitry the computational cost is just a constant $N$, reflecting the hardware committed (neurons and connections), times the length of time it is used:

$$C = ANt_{\text{total}} \tag{15}$$

for fixed hardware, or the more general

$$C = A \int dt N(t) \tag{16}$$

if the amount of hardware devoted to the network can vary over time (a possibility we will consider in detail in section 2 of Part II. Once $N$ is allowed to vary with time, it becomes relevant to consider the details of how much node and wire volume is required to implement dynamically a given pattern of connections.

Equations (14) and (15) go part of the way towards defining a computational system, but they are not yet detailed enough to specify a parallel algorithm or analog circuit that optimizes $E$. Our main line of development will be from these equations towards an analog circuit. But first we note an alternative strategy for generating parallel algorithms which will be developed in sections 2.1 and 2 of Part II.

### 2.1.1 Remarks on Some Generalizations

It is by no means necessary to specialize the expression for $S$ in (5) all the way to the form in (14), if some other way to minimize the original action in (4) can be found. Most alternative sets of $F$ functions would pertain only to one particular objective function $E$, but there are also systematic methods for deriving $F_t$ from $E$ in which $F_t$ benefits from retaining an explicit time dependence. For example, $F_t$ might take the form of $\Delta E_{\alpha(t)}$ for one of $p$ possible objectives $E_\alpha$, where the choice of objective as a function of time (given by $\alpha(t) \in \{1, 2, \ldots p\}$) is made in a cyclic fashion. Then (14) is replaced by

$$F_t(\{x(s', t') | t' \leq t\}) = \sum_\alpha \psi_\alpha(t) \Delta E_\alpha[\mathbf{x}(t), \mathbf{x}(t - \Delta t); \mathbf{x}(t^{\text{old}})], \tag{17}$$

where $\psi_\alpha(t) = 1$ if $\alpha = \alpha(t)$ and 0 otherwise, and where

$$\Delta E_\alpha[\mathbf{x}(t), \mathbf{x}(t - \Delta t); \mathbf{x}(t^{\text{old}})] = E_\alpha[\mathbf{x}(t); \mathbf{x}(t^{\text{old}})] - E_\alpha[\mathbf{x}(t - \Delta t); \mathbf{x}(t^{\text{old}})]. \tag{18}$$

Here we assumed that $t'$ takes only the values $t, t - \Delta t$ and $t^{\text{old}}$, where $t - \Delta t$ is the previous time step in the current $\alpha$ phase of the cycle and $t^{\text{old}}$ is the final time step of the previous phase $\alpha - 1$ in the cycle. Because of its explicit dependence on a cyclic *clock signal* $\alpha(t)$, $E_\alpha$ is called a *clocked objective function*. It must be fundamentally connected to the original objective function $E$ if the resulting cyclic Lagrangian is to have the correct functionality, but there are several ways of making such a connection. This possibility is explored further in section 2.1 of Part II and applied extensively in section 2 of Part II.

It is troubling that there exists a wide variety of different local and causal Lagrangians (cf. (5)) each of whose dynamics will partially optimize the original dynamical objective function or action given by (4). How do we choose one over another, and what are the minimal criteria for any to be acceptable? In other words, what are the rules of the game for proposing distributed cost and functionality terms in (5)? The answers must ultimately be related to algorithmic performance in minimizing the action itself (see (4)). We begin our work on these questions in section 2.3.2.

### 2.1.2 Refinement to Continuous Dynamics

For the moment, let us assume that (14) and (15) describe an acceptable Lagrangian, which is a decomposition of (1) and (2) to finite-sized time steps, and try to further refine them to a dynamics with infinitesimal time steps, i.e. continuous time and continuous-valued (i.e. analog) variables.

A standard form for analog neural networks' objectives $E$ is given in (9). The corresponding functionality term $F$ may be derived with a series of three design transformations. Start with an objective function $E[\mathrm{v}]$ of continuous variables $v_1 \ldots v_m$ and discrete 0/1-valued variables $v_{m+1} \ldots v_n$, with $\phi_i(v_i) = 0$ for the latter (where $\phi$ is defined in (9)). The first transformation is to reformulate the discrete variables as continuous variables each with the *constraints* that $0 \leq v_i \leq 1$. This step may introduce new local minima at the intermediate values of $v_i$; if this possibility can be analyzed away, or designed away by adding a "bump term" such as the penalty term $\sum_i c_i v_i(1 - v_i)$ to $E$, then we have a valid transformation. The second transformation is to replace the constraints with penalty or barrier terms $\phi_i(v_i)$ added to $E$ for unconstrained, continuous-valued optimization. Steps 1 and 2 together may sometimes be replaced by the one-step Mean Field Theory derivation of continuous-valued objectives for discrete-valued variables (first discussed in [Hop84] and extended by others including [Sim90, PS89, GY91]) with improved control over local minima. But in section 2 of Part II we will have occasion to separate the two steps.

As an example of these first two steps, the image region segmentation objective (3) can be refined to an analog neural net with discrete variables $s \in \{0, 1\}$ replaced by continuous variables $l \in [0, 1]$:

$$\begin{aligned} E(f, l^h, l^v) \;=\; & \frac{A}{2} \sum_{ij} (f_{ij} - d_{ij})^2 + \frac{B}{2} \sum_{ij} (f_{i+1,j} - f_{ij})^2 (1 - l^v_{ij}) \\ & + \frac{B}{2} \sum_{ij} (f_{i,j+1} - f_{ij})^2 (1 - l^h_{ij}) + \mu \sum_{ij} (l^h_{ij} + l^v_{ij}) \\ & + \sum_{ij} \phi_i(l^h_{ij}) + \sum_{ij} \phi_i(l^v_{ij}). \end{aligned} \tag{19}$$

Finally, we must refine the global objective $E$ into an arbitrarily large number of infinitesimal-step $\Delta E$ terms for use in the simplest continuous-time dynamics. Using Taylor's theorem for small $\Delta t$,

$$F_{\text{coarse}} = \Delta E \approx \Delta t \sum_i E_{,i} \dot{v}_i \equiv \Delta t F_{\text{fine}}[\dot{\mathrm{v}}] \tag{20}$$

(so that $\sum_t F_{\text{coarse}} \approx \int dt F_{\text{fine}}$), where

$$E_{,i} \equiv \frac{\partial E}{\partial v_i} = -\frac{1}{2} \sum_{ij} T_{ijk} v_j v_k - \sum_j T_{ij} v_j - h_i + \phi'(v_i), \tag{21}$$

and v is a vector of variables comprised of all the $f$, $l^v$, and $l^h$ variables of (19). This third transformation step does not yet specify the associated transformations of the fine-scale cost term $C_{\text{fine}}[\{v_{s,t}\}]$ which we will work out in section 3. The result will be of the form $C_{\text{fine}}[\dot{\mathrm{v}}] = \sum_i K[\dot{v}_i, v_i]$ (cf. (117) of section 3.2). Together with (20), this gives us the Lagrangian

$$LF_{\text{fine}}[[\dot{\mathrm{v}}, \mathrm{v}] = \sum_i \left( K[\dot{v}_i, v_i] + \frac{\partial E}{\partial v_i} \dot{v}_i \right) \tag{22}$$

and the action functional

$$S = \int dt L_{\text{fine}}. \tag{23}$$

This action is in agreement with equation (11). For the region segmentation example, $\partial E/\partial v_i$ is given by (21).

In summary, we have *transformed* the problem three times along the way to the circuit-level functionality term in (20) and an associated Lagrangian. The transformations are intended to preserve (approximately) the fixed points of the equations of motion, while making the dynamics progressively more implementable as an analog neural network. Both the transformations' validity (as measured by the functionality term of the original coarse-scale action (4)) and their efficiency (as measured by the cost term of (4)) must still be demonstrated, since the finer-scale versions of this action functional are only partially optimized. The three transformations used to obtain equation (20) were: (1) discrete variables $\rightarrow$ continuous variables, constrained to intervals; (2) constraints $\rightarrow$ penalty or barrier terms in unconstrained, continous optimization; and (3) temporal refinement: $F_i = \Delta E \approx \int dt \dot{E}$. (The refinement of $C$ must still be worked out before we have a derivation of the fine-scale Lagrangian. See section 3.)

## 2.2 Greedy Functional Derivatives

Based on the foregoing work, we seek to derive continuous-time dynamics from suitable Lagrangians. This requires formulating the greedy derivative of (6) for use with continuous-time dynamics, hence formulating it as a functional derivative.

Following equation (5), we argued that the local cost and functionality terms $F_{s,t}$ and $C_{s,t}$ in a Lagrangian should depend on variables $x_{s',t'}$ only for $t' \leq t$, and that only variables with $t' = t$ should be varied in the optimization of $F_{s,t} + C_{s,t}$; all values of earlier variables are held fixed. Then $F$ and $C$ are said to be in *retarded interaction form*. These constraints can be imposed on any continuous-time Lagrangian in differential form,

$$L(\mathbf{x}(t), \dot{\mathbf{x}}(t), \ddot{\mathbf{x}}(t), \ldots), \tag{24}$$

as follows. First we replace the derivatives by difference expressions $(\mathbf{x}(t) - \mathbf{x}(t - \Delta t))/\Delta t$, and so on, taking care that the largest time $t'$ to appear is $t$. This yields an approximate discrete-time Lagrangian, which we then optimize with respect to $\mathbf{x}(t)$ by differentiating to find the dynamics. Then we take the limit as $\Delta t \rightarrow 0$. In that way we ensure that $t' \leq t$ (retarded interaction form) and that only variables for which $t' = t$ are actually optimized at time $t$, as required.

This procedure for finding the continuous-time dynamics for a Lagrangian in differential form (24) may be formalized by means of the *greedy functional derivative* introduced in [MG90, MM91]. Here we provide a new formal derivation of the greedy functional derivative $\delta_G$ which exploits the retarded interaction form of a Lagrangian.

Let $N$ be a normal form operator on derivative expressions:

$$\begin{aligned}
N[x(t)] &= x(t), \\
N[\dot{x}(t)] &= (x(t) - x(t - \Delta t))/\Delta t, \\
N[\ddot{x}(t)] &= (x(t) - 2x(t - \Delta t) + x(t - 2\Delta t))/(\Delta t)^2, \\
\ldots &\quad \text{and so on. Also} \\
N[F[y(t)]] &= F[N[y(t)]], \quad y = x(t), \dot{x}(t), \ddot{x}(t), \ldots \text{ if } F \text{ is autonomous.}
\end{aligned} \tag{25}$$

So $N$ serves to replace time derivatives by temporal difference expressions for which $t' \leq t$, which we can differentiate with respect to $x(t)$. In other words, it suffices to put a Lagrangian $L$ into retarded interaction form, so that a greedy variation can be taken while preserving its value in the $\Delta t \to 0$ limit. ($N$ is known in numerical analysis as the "backward divided difference operator".) Then the greedy functional derivative may be defined, even on Lagrangians $L$ not yet in retarded interaction form, so as to agree with (6): For any small $\Delta t > 0$,

$$\begin{aligned}
\frac{\delta_G}{\delta_G x(t)} \int d\hat{t} L(x(\hat{t}), \dot{x}(\hat{t}), \ldots) &\equiv \int d\hat{t} \delta(\hat{t} - t) \frac{\partial}{\partial x(\hat{t})} NL(x(\hat{t}), \dot{x}(\hat{t}), \ldots) \\
&= \frac{\partial}{\partial x(t)} NL(x(t), \dot{x}(t), \ldots) \qquad \text{(as in (6))} \\
&= \frac{\partial}{\partial x(t)} L(N[x(t)], N[\dot{x}(t)], \ldots) \\
&= \frac{\partial}{\partial x(t)} L(x(t), \frac{x(t) - x(t - \Delta t)}{\Delta t}, \ldots),
\end{aligned} \tag{26}$$

where the last step used (25). Continuing,

$$\begin{aligned}
\frac{\delta_G}{\delta_G x(t)} \int d\hat{t} L(x(\hat{t}), \dot{x}(\hat{t}), \ldots) &= \frac{\partial}{\partial x(t)} L(x(t), \frac{x(t) - x(t - \Delta t)}{\Delta t}, \ldots) \\
&= \Big( \sum_{n=0}^{\infty} \frac{1}{(\Delta t)^n} \frac{\partial}{\partial (d^n x(t)/dt^n)(t)} \Big) L(x(t), \dot{x}(t), \ldots) \\
&\qquad \text{(by the chain rule)} \\
&= \int d\hat{t} \delta(\hat{t} - t) \Big( \sum_{n=0}^{\infty} \frac{1}{(\Delta t)^n} \frac{\partial}{\partial (d^n x(t)/dt^n)(\hat{t})} \Big) L(x(\hat{t}), \dot{x}(\hat{t}), \ldots) \\
&= \Big( \sum_{n=0}^{\infty} \frac{1}{(\Delta t)^n} \frac{\delta}{\delta (d^n x(t)/dt^n)(t)} \Big) \int d\hat{t} L(x(\hat{t}), \dot{x}(\hat{t}), \ldots).
\end{aligned} \tag{27}$$

Here the functional derivatives $\delta / \delta (d^n x(t)/dt^n)$ are taken to be independent of one another as partial functional derivatives (so for example $\delta \dot{x}(\hat{t}) / \delta x(t) = 0$, rather than $\delta \dot{x}(\hat{t}) / \delta x(t) = d\delta(\hat{t} - t)/d\hat{t}$ as would be the case for total functional derivatives).

So the greedy functional derivative $\delta_G / \delta_G x(t)$ is given by the operator equation

$$\frac{\delta_G}{\delta_G x(t)} = \sum_{n=0}^{\infty} \frac{1}{(\Delta t)^n} \frac{\delta}{\delta (d^n x(t)/dt^n)(t)}, \tag{28}$$

where $\Delta t$ is infinitesimal. Again, the conventional functional derivatives are independent of one another (they are partial functional derivatives). Needless to say, the highest powers of $(1/\Delta t)$ will dominate all others in the limit $\Delta t \to 0$. For example if $L$ depends on $\mathrm{v}$ and $\dot{\mathrm{v}}$, but not on higher time derivatives, then the greedy functional derivative will be $(1/\Delta t)\delta / \delta \dot{\mathrm{v}}$. This will generally be the case for our circuit Lagrangians.

We can derive analog, continuous-time network dynamics by applying the greedy functional derivative to the continous-time Lagrangian (22). Since the highest time-derivative in the Lagrangian is $\dot{v}_i$ for each variable $v_i$, the greedy functional derivative is proportional to $\delta / \delta \dot{\mathrm{v}}$. Then the equations of motion become

$$\frac{\delta S}{\delta \dot{v}_i} = K_{,\dot{v}}[\dot{v}_i, v_i] + \frac{\partial E}{\partial v_i} = 0. \tag{29}$$

For $K[\dot{v}, v] = (1/2)\tau_H \dot{v}_i^2 / g'(g^{-1}(v_i))$, the circuit-level cost term which will be derived in section 3.2.3, and for an objective function $E$ given by equations (9) and (10), the greedy variation equations become Hopfield/Grossberg dynamics:

$$\tau_H \dot{u}_i + u_i = \sum_{jk} T_{ijk} v_j v_k + \sum_j T_{ij} v_j + h_i \text{ and } v_i = g(u_i). \tag{30}$$

This type of dynamical system describes an analog neural network, and we will make no distinction between such a dynamical system and the neural network itself.

As an example, we may work out the dynamics for the region segmentation Lagrangian given by (22) and (19). Specializing the dynamics of (30) to the region segmentation objective (19), we can expand the first term of the objective to find a potential term $(A/2)f_{ij}^2$ for the $f_{ij}$ variables. Then we find the standard Hopfield/Grossberg equations of motion for this analog network, which are

$$\begin{array}{rclcrcl} \tau_f \dot{e}_{ij} + e_{ij} & = & Ad_{ij} - B(f_{ij} - f_{i+1,j})(1 - l^v_{ij}) - B(f_{ij} - f_{i,j+1})(1 - l^v_{ij}), & & f_{ij} & = & (1/A)e_{ij}, \\ \tau_k \dot{k}^v_{ij} + k^v_{ij} & = & B/2 \sum_{ij} (f_{i+1,j} - f_{ij})^2 - \mu, & & l^v_{ij} & = & g(k^v_{ij}), \\ \tau_k \dot{k}^h_{ij} + k^h_{ij}, & = & B/2 \sum_{ij} (f_{i,j+1} - f_{ij})^2 - \mu, & & l^h_{ij} & = & g(k^h_{ij}). \end{array} \tag{31}$$

## 2.3 Theory for Refinement to Circuit Lagrangians

We have found a path of argument from computational first principles to specific neural networks, but the status of some of the steps along the path is still unclear. The basic problem is that various transformations of the original action functional (4) are required to get an implementable dynamical system, and limitations of causality and the simplicity of elementary processing devices require that the spatially and temporally distributed Lagrangian functional (such as (5) or (11)) be optimized only partially (as in the discussion following (5)).

Our approach to this basic problem is to catalog a variety of useful transformations that lead towards circuits or parallel algorithms, and to re-use the fundamental dynamical objective function (4), or closely related quantities, as a measure (i.e. a criterion) for judging the success of such transformations. Such a criterion may be called a *meta-objective* since it is an objective function used to select a dynamical objective function for the neural network dynamics.

This approach may be thought of as a symbolic search procedure to be carried out by human designers, who select the likely transformation sequences, with machine assistance in evaluating them and perhaps also performing them. On occasion it may be possible to eliminate the search procedure by proving the (meta-) optimality of a given Lagrangian, but we do not think that this will be possible in most cases.

### 2.3.1 Transformations of Lagrangians

Recall the three transformations leading to circuit-level Lagrangians in section 2.1.2:

**T1.** discrete variables $\rightarrow$ continuous variables constrained to intervals

**T2.** constraints $\rightarrow$ penalty or barrier terms in unconstrained continous optimization

**T3.** refinement: $F_t = \Delta E \approx \int dt \dot{E}$. (The refinement of $C$ will be worked out in section 3.)

We comment on each of these transformations.

T1 and T3 are required to achieve a circuit implementation, but more generally they serve the purpose of making a parallel algorithm. Discrete-time update schemes may be introduced instead, but some care is required so that the updates of independent variables done in parallel don't have the joint effect of increasing rather than decreasing $E$. For example, for some networks it is possible to "color" the variables with a small number of colors so that no two connected variables ($x_i$ and $x_j$ such that $T_{ij} \neq 0$) have the same color; then different colors can be updated at different times in a clocked objective function, and all the variables of the same color can be updated at once (even by discrete jumps) without interference in $E$. (Interference would mean that several variables would each, if updated alone, diminish $E$, but if the same updates were done together then $E$ could increase.) Such (fairly standard) parallel update schemes are not so important for continous-time and analog-valued networks, whose descent dynamics are explicitly parallel.

Transformations like T2, which incorporate static constraints into the static optimization problem, may change the nature of the optimization problem significantly. Penalty and barrier terms on constraints that involve many variables destroy locality, unless they are further transformed to a local form by methods such as those described in [MG90]. In this case a minimization problem is replaced by a saddle-point problem. Alternatively one can introduce Lagrange multipliers, but that also changes the static optimization into a saddle point problem [PB87]. Either way, the dynamics associated with the Lagrangian functional loses its obvious convergence properties (because limit cycles around a saddle point become possible), and it may be necessary to engage in *meta-optimization* of some kind in order to secure convergence for a local circuit implementation. Another alternative, which requires clocked objective functions but does not explicitly introduce saddle points, is to use an algorithm similar to the "gradient projection algorithm" or "scaled gradient projection algorithms" [BT89] to repeatedly reestablish the constraints as the dynamics proceed. Such an alternative will be employed in section 2.1.6 of Part II.

In previous work [MG90] it has been demonstrated that static neural network objective functions may be transformed in a variety of ways in order to acheive design goals such as reduced wiring cost or attaining an implementable form while preserving the functionality (the fixed points) of an optimizing neural network. Likewise, in this paper we will introduce a number of transformations from one Lagrangian to another that satisfy design constraints while preserving or improving the functionality of a computation.

A fundamental aspect of (5) is that, due to its linearity, it naturally supports the hierarchical decomposition of computational dynamics into large state changes (or decisions), each achieved through many smaller state changes or decisions. This is in analogy to multiscale or multigrid algorithms from numerical analysis, or to renormalization group ideas in statistical physics, or to the idea of stepwise refinement in the design of computer programs. As in (5), the action $S$ can be decomposed into a sum over state-change decisions. But if each of these decisions is in turn made by a dynamical system consisting of a sequence of sub-decisions at a finer time scale (which may also involve a finer spatial scale), then we can relate the two time scales ("big" decisions and "sub-decisions") and reexpress the action in terms of the fine-scale

decisions alone ("small" decisions):

$$\begin{aligned} S &= A \sum_{\text{big decisions}(\tilde{s},\tilde{t})} \tilde{C}_{\tilde{s},\tilde{t}}(\{\mathbf{x}(t')\}) + B \sum_{\text{big decisions}(\tilde{s},\tilde{t})} \tilde{F}_{\tilde{s},\tilde{t}}(\{\mathbf{x}(t')\}) \\ &= A \sum_{\text{big decisions}(\tilde{s},\tilde{t})} \left[ \sum_{\text{sub-decisions}(s,t)} C_{s,t}(\{\mathbf{x}(t')\}) \right] \\ &\quad + B \sum_{\text{big decisions}(\tilde{s},\tilde{t})} \left[ \sum_{\text{sub-decisions}(s,t)} F_{s,t}(\{\mathbf{x}(t')\}) \right] \\ &= A \sum_{\text{small decisions}(s,t)} C_{s,t}(\{\mathbf{x}(t')\}) + B \sum_{\text{small decisions}(s,t)} F_{s,t}(\{\mathbf{x}(t')\}). \end{aligned} \tag{32}$$

Notice that the step from equation (4) to equation (5), or more specifically to (7) and (8), can be given a similar hierarchical interpretation: we are expressing a single quantity, optimized over the entire circuit convergence time, as a sum of quantities to be optimized more locally in time or space. The further refinement to infinitesimal time steps, (23), is another example. Then equation (32) subsumes all these examples of hierarchical design.

### 2.3.2 Meta-Optimization

We have discussed the necessity for some criterion or figure of merit by which to compare alternative Lagrangians and the dynamical systems to which they give rise. Generally we start with some global objective function such as $S$ in (4), then transform it though a series of spatially and temporally localized Lagrangians of the form (5) to a final circuit-level Lagrangian $L$, which is only partially optimized (i.e. is greedily optimized) by the dynamics. Finally we wish to quantify the performance of the resulting dynamical system, i.e. to evaluate the quality of the associated computation, for example by computing the value of $S$ at the end of a run. The meta-optimization problem is to optimize the resulting evaluation, treating it as a functional of the exact form of $L$.

An obvious way to do that is by means of a retrospective (a posteriori) evaluation of the original objective $S$ of (4). But *optimizing* with respect to this protocol of retrospective evaluation of $S_{coarse}$ seems out of the question, since that involves many repeated tests of the neural network dynamics with different values of the parameters that specify the (transformed) Lagrangian and is therefore far more expensive than one relaxation run of the network. (The parameterization of $L$ may involve real-valued parameters or may simply be the discrete choice of a sequence of transformations to derive $L$ from $S_{coarse}$.)

Fortunately the cost of optimizing $S_{coarse}$ as a function of the form of $L$ (i.e. the cost of meta-optimization) may by *amortized* over many inputs $\mathbf{h}$ (cf. (9)) to one network, drawn according to some probability distribution, or even over many network connection matrices $T$ drawn according to another probability distribution. Optimizing $\mathcal{M}$ may be very expensive but the expense is amortized by using the resulting dynamics to improve the performance of many different computations. An apparent obstacle is that different $\mathbf{h}$ vectors and $T$ matrices will in general have unrelated meta-objectives $\mathcal{M}$, so amortization may be difficult to accomplish.

Such amortization may still be achieved if the meta-objective function $\mathcal{M}[L]$ is

altered to become an average-case measure of $S_{coarse}$:

$$\mathcal{M}[L] =< S_{coarse}[L] >_{h,T} . \tag{33}$$

Just as in neural network learning procedures, the distribution average would be sampled by a finite sum over a *training set*; this sum would be optimized, and then a further sampling could be made to test *generalization* from the training set to a testing set. If such generalization is to be expected, either on experimental evidence or according to theoretical criteria such as the Vapnik-Chervonenkis dimension [Vap82, BH89], then amortization will be possible. For the cost of computing (hence of optimizing) $\mathcal{M}[L]$ is multiplied by the size of the training set, but that large initial cost is then effectively divided by the number of times that $L$ is used subsequently, which may be far larger than the training set. This gives the desired amortization.

Alternatively, one could amortize the cost of optimizing $\mathcal{M}$ by taking $\mathcal{M}$ to be a worst-case measure of $S_{coarse}$ which can be optimized analytically. The worst case performance is very hard to evaluate experimentally, but it may be more easily subject to analysis than the average-case performance, at least if we are allowed to alter the form of $S_{coarse}$ somewhat. That will be our approach in section 3.2.

# 3 CIRCUIT DYNAMICS

## 3.1 Refinement to a Circuit

Upon refinement, the Lagrangian $L = C + F$ becomes

$$L = AN\Delta t + B\Delta E. \tag{34}$$

We would like to take the limit $\Delta t \to 0$, refining to infinitesimally small time steps in a continuous analog circuit. We expect this to be both simpler than a discrete-time (finite $\Delta t$) dynamics, and also more relevant to neural network implementations. But performing the greedy optimization of such a Lagrangian presents some surprising problems.

For instance, a first-order expansion of $\Delta E(\Delta t)$ yields a Lagrangian proportional to $\Delta t$: $L[\dot{\mathbf{v}}, \Delta t] = \Delta t(C + B\sum_i E_{,i}[\mathbf{v}]\dot{v}_i)$, which cannot be optimized with respect to $\Delta t \geq 0$ without going outside the expansion's domain of validity. To avoid this problem $\Delta t$ might be taken to be a small constant, but that would make the entire cost term $C = AN\Delta t$ constant and therefore irrelevant to the dynamic optimization problem. More seriously, partial optimization can only affect $\dot{\mathbf{v}}$ which appears linearly in this Lagrangian; $\dot{v}_i = \pm\infty$ will be the optimum, which would not only invalidate the expansion of $E(t)$ again, but would violate physical limits on circuits as well. A somewhat more physical dynamics would result if we arbitrarily followed the analogy from the Lagrangians of physics and changed the cost term to a kinetic energy $(1/2)\sum_i \dot{v}_i^2$, but we have no computational justification for doing so.

On the other hand, not expanding $\Delta E(\Delta t)$ at all leaves a fine-scale optimization problem which is equivalent to optimizing the full coarse-scale objective $E$ in much less time. This is simply not possible. And even a second-order finite Taylor expansion of $\Delta E(\Delta t)$ is problematic, since the optimized values of $\Delta t$ and $\dot{v}$ are likely to lie outside the expansion's small domain of suitability as an aproximation.

The essential problem here is that each fine-scale optimization, to be implementable as a circuit, must be *more constrained* than the coarse-scale optimization. We must stay within the domain of convergence of a Taylor expansion of $\Delta E(\Delta t)$, and we must not violate physical speed limits (e.g. for physical implementability we must prevent circuit time constants from becoming too small), and so on. Such constraints are either (a) direct physical limits on circuit implementations, or (b) computational limits on what can be achieved with a small amount of physical computing (computation which occurs in a physical medium) in time $\Delta t$. These constraints are generally too complex to state exactly in a simple Lagrangian.

We identify two general approaches to formulating such circuit constraints and the corresponding fine-scale Lagrangians. In the "underconstrained" approach, we impose simplified, loose versions of the physical and computational constraints on the optimization of $L_{\text{coarse}}$, in the hopes that the resulting dynamics will be constrained enough for a genuine physical implementation (perhaps at an even finer time scale). These loose constraints can be tightened up for analytic or computational convenience, and then expressed as penalty or barrier functions which are added to $L$ to form $L_{\text{fine}}$, the fine-scale Lagrangian. By contrast, the "overconstrained" approach stays within the realm of physical implementation by hypothesising a parameterized class of fine-scale Lagrangians known to be implementable, which can be thought of as alternative strategies, and optimizing some measure of their relationship to the original coarse-scale Lagrangian $L$. In particular, the cost terms of $L_{\text{fine}}$ may be optimized while the functionality term is taken to be $\Delta E \approx \Delta t \dot{E}$ as in the coarse-scale Lagrangian. Thus the underconstrained approach applies looser constraints than implementability may actually require, and the overconstrained approach applies tighter constraints than are actually required. We give examples of each.

### 3.1.1 Underconstrained Refinement #1

We will require $\Delta \mathbf{v}$ be small enough so that $\Delta E[\Delta \mathbf{v}]$ can be expanded to first (or second) order in a Taylor series, and that each $|\dot{v}_i|$ be bounded by a physical speed limitation. So we must optimize

$$L[\dot{\mathbf{v}}, \Delta t] = AN\Delta t + B\Delta E[\Delta \mathbf{v}] \tag{35}$$

subject to

$$||\dot{\mathbf{v}}||_\infty = \max_i |\dot{v}_i| \leq s \tag{36}$$

(where $\dot{\mathbf{v}} \approx \Delta \mathbf{v}/\Delta t$) and

$$||\Delta \mathbf{v}||_2 \leq r(v), \tag{37}$$

where $r(v)$ is chosen to ensure that a first (or second) order expansion of $\Delta E[\Delta \mathbf{v}]$ is sufficiently accurate. Also, there are two approaches to varying $\Delta t$. If we let $\Delta t$ be optimized (subject to $\Delta t \geq 0$), the cost term in the Lagrangian will keep it small but not necessarily drive it to the continuum limit $\Delta t \to 0$. Or, we can let $\Delta t = \chi\tau$, where $\chi \in \{0, 1\}$ is a discrete dynamical variable which "stops" the neural network when $\chi$ is optimized to zero, and where $\tau$ is a small constant which we can analytically drive towards zero to extract continuum dynamics.

In the latter case, $||\Delta \mathbf{v}||_2 \approx \chi\tau||\dot{\mathbf{v}}||_2 \leq \chi\tau\sqrt{n}s$ is more restrictive in the limit

both constraints become irrelevant. So we can drop constraint (37). If we express constraint (36) as a barrier function $\sum_i \phi_{\pm 1}(\dot{v}_i/s)$, the fine-scale Lagrangian becomes unconstrained:

$$L_{\text{fine}}[\dot{\mathbf{v}}, \chi] = \sum_i \phi_{\pm 1}(\dot{v}_i/s) + \chi\tau[AN + B\sum_i E_{,i}\dot{v}_i]. \tag{38}$$

Except for the new $\chi$ variable, this is the same form of Lagrangian for neural networks that we have proposed in [MG90, Mjo87]. The corresponding dynamics are (varying $\dot{\mathbf{v}}$, cf. (28))

$$\dot{v}_i = -sg_{\pm 1}(E_{,i}), \tag{39}$$

and varying $\chi$ to get the stopping criterion, we find the optimal values of $\chi$ occur only at the boundaries of the allowed domain of $\chi$:

$$\chi = \Theta[s\sum_i E_{,i}g_{\pm 1}(E_{,i}) - AN]. \tag{40}$$

Here $\Theta(x)$ is the Heaviside function (1 for $x \geq 0$; 0 for $x < 0$).

### 3.1.2 Underconstrained Refinement #2

If, on the other hand, we let $\Delta t$ be optimized freely, then we are taking a computational step that requires a small but nonzero amount of time to change the state by $\Delta\mathbf{v}$, which is constrained by both (36) and (37), which in turn are related by $\dot{\mathbf{v}} \approx \Delta\mathbf{v}/\Delta t$. We will express constraint (36) as $||\Delta\mathbf{v}||_\infty \leq s\Delta t$, which can be tightened to the more tractable

$$(1/s)\sum_i |\Delta v_i| \leq \Delta t. \tag{41}$$

Also we can tighten constraint (37) to

$$||\Delta\mathbf{v}||_\infty \leq \frac{r(v)}{\sqrt{n}} \equiv \hat{r}(v) \tag{42}$$

(which implies (37)). Optimizing $L[\dot{\mathbf{v}}, \Delta t]$ of (35) with respect to $\Delta t$, which occurs linearly in (35), as constrained by (41) just saturates the constraint: $\Delta t = (1/s)\sum_i |\Delta v_i|$.

The remaining constrained optimization is with respect to $\Delta\mathbf{v}$. Using barrier functions, we find an unconstrained Lagrangian

$$\hat{L}[\Delta\mathbf{v}] = \frac{AN}{s}\sum_i |\Delta v_i| + \sum_i E_{,i}\Delta v_i + \sum_i \phi_{\pm 1}\Big(\frac{\Delta v_i}{\hat{r}(v)}\Big), \tag{43}$$

or

$$\hat{L}[\Delta\mathbf{v}] = \sum_i E_{,i}\Delta v_i + \frac{AN\hat{r}(v)}{s}\sum_i \phi_{+/0/-}\Big(\frac{\Delta v_i}{\hat{r}(v)}\Big), \tag{44}$$

where $\phi_{+/0/-}(x) \simeq \phi_{\pm 1}(x) + |x|$. Also

$$\Delta v_i/\hat{r}(v) = \begin{cases} -1 & \text{if } E_{,i} - AN/s > 0 \\ 0 & \text{if } E_{,i} - AN/s < 0 \text{ and } E_{i,i} + AN/s > 0 \\ +1 & \text{if } E_{,i} + AN/s < 0 \end{cases} \tag{45}$$

A number of calculations of bounding expressions $\hat{r}(v)$ are possible, but we will not pursue this approach further here.

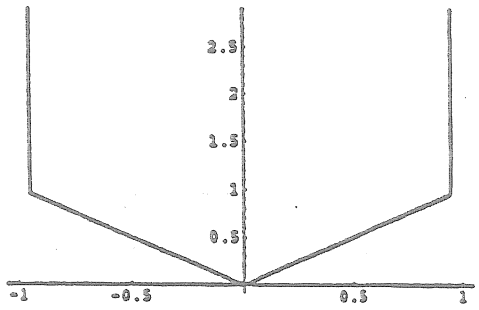


Figure 1: Potential $\phi_{+/0/-}(x)$ incorporates automatic stopping criterion. When other terms fail to alter the ordering among $\phi(-1)$, $\phi(0)$, and $\phi(1)$, then $\Delta v = 0$ is favored and neuron $v_i$ stops.

### 3.1.3 Stopping Criterion

Lagrangians (38) and (43) each have intrinsic stopping criteria which compare the expected improvement in functionality $\Delta E$ with a cost of movement, and allow movement only when it is sufficiently beneficial. But $E$ may not always be the right function for this purpose. A monotonic function $b(E)$ may be used in place of $E$ in (8) and may likewise be decomposed into a sum of $\Delta b$ terms. The latter would alter the tradeoff with the cost term for incomplete optimizations and therefore the stopping criterion (the point at which a further decrease in F is smaller than the expected cost of obtaining it).

One major drawback of using a monotonic function $b(E)$ in place of $E$ in a Lagrangian is that if $E$ is of the standard neural network form (9), it is already a sum of local terms and therefore close to neural implementation. By contrast, direct optimization of $b(E)$ requires a global calculation of $E$ even to get the gradient $\nabla b = b' \nabla E$ needed for the dynamics of every variable. One can circumvent this problem by transforming the objective function with a particular type of Legendre transformation [MG90]:

$$\chi b(E) \rightarrow -\sigma E + \chi\tau + \sigma b^{-1}(\tau). \tag{46}$$

In the resulting gradient dynamics, only the one variable $\sigma$ requires computation of the objective function $E$. Unfortunately this transformation replaces a static minimization objective with a static saddle-point objective, since some of the new variables are to maximize rather than minimize the transformed objective. To find a Lagrangian which always converges, rather than cycling around the saddle point, may then require an appeal to meta-optimization (e.g. either experiment or deeper analysis) of the saddle-point-seeking Lagrangian.

## 3.2 Overconstrained Refinement: Meta-Optimization of $K$

A second, more systematic way to overcome the problems with refining the Lagrangian through expansion of $E(\Delta t)$ is to define a class of Lagrangians which are known to be physically implementable and mathematically tractable, though they are not the only physically implementable expressions for a circuit-level Lagrangian, and to pick the

best member of the class based on a meta-optimization criterion. So we overconstrain the set of allowed Lagrangians and optimize. We will be able to do this theoretically for a meta-objective that measures worst-case performance of a Lagrangian for minimizing an especially simple class of neural network objective functions.

The allowable class of objective functions will be those of the form $E[\mathbf{v}] = -(1/2)\sum_{ij} T_{ij}v_iv_j - \sum_i h_iv_i + \sum_i \phi(v_i)$, in which the matrix $T$ is negative semi-definite and has eigenvalues whose absolute values are bounded above by some number $t_{\max}$. An example of such an objective function is the hysteresis-free version of the common winner-take-all network objective [HT85] $E = (A/2)(\sum_i v_i - 1)^2 - \sum_i h_iv_i + \sum_i \phi(v_i)$. There is a straightforward generalization to the case in which different neurons $v_i$ have different potential functions $\phi_i(v_i)$, but we won't work that out here. The negative-definite restriction on $T$ is severe because it means that $E$ must be unimodal (since each $\phi_i$ is unimodal too). Unimodal objectives have some computational uses, such as in the winner-take-all network or the "invisible hand" algorithm for matching [KY91], but our meta-optimization results will not be widely applicable until they are generalized to multimodal objective functions. Nevertheless we can present the unimodal analysis as an example of the meta-optimization of a circuit-level Lagrangian.

What mathematical conditions would make a Lagrangian physically implementable, so the associated dynamics can be implemented with a circuit, and also result in good performance? The essential limiting factors for circuit speed are the *time constants* (such as resistance-capacitance products in an electrical circuit) that govern the approach to any stable state of any one- or two-element subcircuit. These time constants must be larger than some physical lower bound, say $\tau_{\text{fast}}$. We also want the stable fixed points to be minima of some neural network objective $E$. Subject to these constraints, we want to minimize the slowest time constant for the full circuit (which as we will show is also larger than $\tau_{\text{fast}}$). Of course time constants are only defined for a local linearization of a dynamical system, so we must constrain them in the neighborhood of each attainable configuration, and we may optimize the worst case time constant over all such configurations.

With these points in mind, we define a constrained optimization problem over a limited class of Lagrangians of the form

$$L[\dot{\mathbf{v}}] = \sum_i K[\dot{v}_i, v_i] + \sum_i E_{,i}\dot{v}_i, \tag{47}$$

where the objective takes the form

$$E[\mathbf{v}] = -\frac{1}{2}\sum_{ij} T_{ij}v_iv_j - \sum_i h_iv_i + \sum_i \phi(v_i), \tag{48}$$

and $\mathbf{h}$ includes the input to the network. Note that the cost term in (47) is a sum over kinetic-energy terms each pertaining to only one neuron; this is a form of locality. Also the equivalence of stable fixed points and local minima of $E$ can be ensured by simple constraints on $K$. (47) together with the time constant constraints and $K$ constraints to be introduced specify the class of Lagrangians that we will call "circuit-implementable". This class is parameterized by the kinetic-energy function $K$ from $\Re^2$ to $\Re$, suitably constrained.

One important property of equation (47) is that it retains its form under componentwise *reparameterizations* $v_i = f_i(x_i)$, where $f_i$ is monotonically increasing, differentiable, and its inverse is differentiable. (Note that such reparameterizations form

a continuous group under composition.) That is, under such a reparameterization the $dE/dt$ term is invariant, and the $K$ term, while not invariant, becomes another function $\hat{K}[\dot{x}_i, x_i]$ of the corresponding new variables. So the problem of optimizing with respect to $K$ can be solved equivalently in any such parameterization we choose, if only the objective and constraints are also chosen to be parameterization-invariant in this sense. We will insure that condition by deriving them from physical circuit time-constants for exponential convergence to fixed points.

The greedy functional derivative was derived in section 2.2. We use that result to find the greedy optimum of the action $\int dtL$ with respect to the trajectory $\mathbf{v}(t)$. The dynamical system that results from calculating the greedy variation of $L$ with respect to $\mathbf{v}$ (i.e. the regular variation with respect to $\dot{\mathbf{v}}$) and setting it to zero is

$$\dot{v}_i = \tilde{K}[-E_{,i}, v_i], \tag{49}$$

where $\tilde{K}[w, v]$ is the inverse of $K[\dot{v}, v]_{\dot{v}}$ on its first argument. This forces us to constrain $K$ to be monotonic in its first argument. Here we introduce the notation

$$w_i[\mathbf{v}] = -E_{,i} = \sum_j T_{ij} v_j + h_i - \phi'(v_i). \tag{50}$$

For stable fixed points to correspond to local minima of $E$ (for which $\mathbf{w} = 0$), it suffices to assume that

$$\tilde{K}[0, v] = 0 \text{ and } \tilde{K}[w, v]_{,w} \geq 0 \tag{51}$$

for all $w$ and $v$. The linearization of this dynamical system at $\mathbf{v}$ is

$$\dot{\Delta v}_i = \tilde{K}[w_i, v_i] + \sum_j A_{ij} \Delta v_j, \tag{52}$$

where

$$\begin{aligned} A_{ij} &= \frac{\partial}{\partial v_j} \tilde{K}[w_i, v_i] \\ &= \tilde{K}_{,w}[w_i, v_i]\big(T_{ij} - \delta_{ij}\phi''(v_i)\big) + \tilde{K}_{,v}\delta_{ij}. \end{aligned} \tag{53}$$

Now we are in a position to derive the constraints on the function $K$ that result from considering the time-constants of the dynamics specified by $A = (A_{ij})$. We want the circuit elements and their connections to be physically implementable, so we'll constrain one- and two-element subcircuits of the linearized system (52) to be slower than $\tau_{\text{fast}}$. We do this by setting all elements of $A$ to zero except for $A_{ii}$ (for a one-element subcircuit) or $\{A_{ii}, A_{ij}, A_{ji}, A_{jj}\}$ (for a two-element subcircuit), to get a $1 \times 1$ or $2 \times 2$ matrix $\hat{A}(i)$ or $\hat{A}(i, j)$. Furthermore, we may arbitrarily pick the subcircuit's fixed point $\mathbf{v}^*$ by adjusting the input vector $\mathbf{h}$; this does not alter any element of $A$ or $\hat{A}$. In that case $\tilde{K}[w_i, v_i] = 0$, and the linearized dynamics (52) converges exponentially to $\mathbf{v}$ with a time constant determined by the largest eigenvalue $\{\lambda_i\}$ of the matrix $\hat{A}$, i.e. by its matrix norm $||\hat{A}||_2$. So the physical constraint would be

$$\max_{\hat{A} \subset A} ||\hat{A}||_2 \leq 1/\tau_{\text{fast}}, \tag{54}$$

where $\hat{A} \subset A$ means that $\hat{A}$ is varied over all $1 \times 1$ and $2 \times 2$ submatrices of $A$ and over all state vectors $\mathbf{v}$.

The constraint (54) is parameterization-invariant. Invariance follows for any $\hat{A}$ by applying Taylor's theorem at a fixed point $\mathbf{v}^*$ of $\mathbf{v}$, to get the linearized dynamics in a new coordinate system $\{x_i = f_i(v_i)\}$. The new matrix $\tilde{A}$ is just a similarity transform $J\hat{A}J^{-1}$ of $\hat{A}$, where $J$ is the (nonsingular) Jacobian of the change of coordinates. Therefore $\tilde{A}$ and $\hat{A}$ have the same eigenvalues (cf. [Ner70], Theorem 5.2 or 5.3) and $||\hat{A}||_2$ is parameterization-invariant as long as the Jacobian $J$ is not singular (which ours never are). Furthermore, the identity of the $1 \times 1$ and $2 \times 2$ submatrices of $A$ are invariant under our coordinate-wise reparameterizations $\{x_i = f_i(v_i)\}$. So the whole constraint (54) is parameterization-invariant. This invariance confirms the intuition that exponential convergence to a fixed point in one coordinate system $\{v_i\}$ ( i.e. $\mathbf{v} - \mathbf{v}^* \approx \mathbf{c}\exp -\lambda t$) does not change its convergence exponent $\lambda$ in another coordinate system $\{x_i = f_i(v_i)\}$.

Note that because each $f_i$ is assumed to be monotonic, differentiable, and to have a differentiable inverse, constraints (51) are also parameterization-invariant. That's because each $w_i = -E_{,i}$ is multiplied by $f_i'(v_i)$ in reparameterization $\{x_i = f_i(v_i)\}$, where $0 < f_i'(v_i) < \infty$.

Constraint (54) is not a sufficiently convenient form for all our subsequent analysis, so we will relate the constraint to something more tractable. The matrix norm of each $\hat{A} \subset A$ is bounded above and below by multiples of $\max_{ab} |\hat{A}_{ab}|$ (cf. [GL83], p. 15):

$$\max_{ab} |\hat{A}_{ab}| \leq ||\hat{A}||_2 \leq \dim(\hat{A}) \max_{ab} |\hat{A}_{ab}|, \tag{55}$$

whence

$$\max_{ij} |A_{ij}| \leq \max_{\hat{A} \subset A} ||\hat{A}||_2 \leq 2 \max_{ij} |A_{ij}|, \tag{56}$$

where as before $\hat{A}$ ranges over all $1 \times 1$ and $2 \times 2$ submatrices of $A$. So a closely related but more tractable constraint may be formulated:

$$\max_{\mathbf{v}} \max_{ij} |A_{ij}(\mathbf{v})| \leq 1/\tau_{\text{fast}}. \tag{57}$$

Of course, the bounds of (56) hold regardless of what coordinate system is used to derive $A$, so long as $\hat{A}$ is expressed in the same coordinate system. Still, constraint (57) is *not* parameterization-invariant, since similarity transformations do not preserve the elements of a matrix. We will have occasion to use both (54) and (57) in what follows.

Since one $K$ is to apply to many connection matrices $T$ and state vectors $\mathbf{v}$, we will also constrain a worst-case estimate of the circuit speed over all $T$ in some allowable class $\mathcal{T}$ in the formula for $A$, and over all state vectors $\mathbf{v}$ for each connection matrix:

$$\max_{\mathbf{v}} \max_{T \in \mathcal{T}} \max_{\hat{A} \subset A} ||\hat{A}||_2 \leq 1/\tau_{\text{fast}}. \tag{58}$$

As previously mentioned, we take $\mathcal{T}$ to be the set of negative-semidefinite connection matrices $T$, such that the absolute values of the $T$'s eigenvalues (i.e. $T$'s singular values) are bounded above by $t_{\max}$. Constraint (58) is parameterization-invariant but not as analytically tractable as the alternative,

$$\max_{\mathbf{v}} \max_{T \in \mathcal{T}} \max_{ij} |A_{ij}(\mathbf{v}, T)| \leq 1/\tau_{\text{fast}}. \tag{59}$$

which will enter into the following analysis even though it is not parameterization-invariant.

The invariance of constraint (58) is one reason to prefer the time-constant constraint (58) over the "speed limit" imposed in sections (3.1.1) and (3.1.2), which explicitly depends on the choice of variables. On the other hand the speed-limit constraints take into account the entirety of each trajectory, rather than just the behavior near (all possible) fixed points.

Next we must formulate the objective function, which will be a worst-case estimate of the much slower time constant for convergence of the full circuit (as opposed to 2 $\times$ 2 subcircuits). We want to minimize $\tau_{\text{slow}}$, where

$$\begin{aligned} \tau_{\text{slow}} &= \max_{\mathbf{v}} \max_{T \in \mathcal{T}} \max_i 1/|\lambda_i(A(\mathbf{v}, T))| \\ &= \max_{\mathbf{v}} \max_{T \in \mathcal{T}} \max_i |\lambda_i(A^{-1}(\mathbf{v}, T))| \\ &= \max_{\mathbf{v}} \max_{T \in \mathcal{T}} ||A^{-1}(\mathbf{v}, T)||_2. \end{aligned} \tag{60}$$

Equivalently we want to maximize

$$\min_{\mathbf{v}} \min_{T \in \mathcal{T}} ||A^{-1}(\mathbf{v}, T)||_2^{-1}. \tag{61}$$

Again, the objective (60) will be parameterization-invariant because the time-constants are invariant under similarity transformations.

Because the optimization of (60) with respect to $K[\dot{v}, v]$ subject to (58) is invariant under reparameterizations $x_i = f_i(v_i)$, we may change variables to $u_i = \phi_i'(v_i)$, calculate $A$ for the linearized $u$ variables, restate the optimization problem, and find the optimizing $K$. The functions $\phi$ are the single-variable potentials appearing in equation (48), so each $\phi_i'$ is monotonic, differentiable, and has a differentiable inverse. The variables $u_i$ were introduced in equation (10). Using the u variables, one may express the dynamics by means of the Lagrangian

$$L = \sum_i \tilde{K}[\dot{u}_i, u_i] + \sum_i \frac{\partial E}{\partial u_i} \dot{u}_i, \tag{62}$$

whence the equation of motion

$$\dot{u}_i = \tilde{K}_{\dot{u}}^{-1}[-\frac{\partial E}{\partial u_i}, u_i] \tag{63}$$

(where the function inverse concerns only the first argument, $\dot{u}_i$, of $\tilde{K}_{,\dot{u}_i}$). This may be rewritten in terms of $w_i$ from equation (50):

$$w_i = -\frac{\partial E}{\partial v_i} = -\frac{1}{g'(u_i)} \frac{\partial E}{\partial u_i} \tag{64}$$

which enables us to define

$$\hat{K}[w_i, u_i] = \tilde{K}_{,\dot{u}}^{-1}[w_i g'(u_i), u_i] \tag{65}$$

and reexpress the $\dot{u}_i$ dynamics as

$$\dot{u}_i = \hat{K}[w_i, u_i]. \tag{66}$$

Then the linearized dynamics is

$$\dot{\Delta u}_i = \hat{K}[w_i, u_i] + \sum_j A_{ij}\Delta u_j, \tag{67}$$

where $A_{ij} = \partial \hat{K}[w_i, u_i]/\partial u_j$, i.e.

$$-A_{ij} = \hat{K}_{,w}[w_i, v_i]\left(\tilde{T}_{ij}g'(u_i) + \delta_{ij}\right) - \hat{K}_{,u}[w_i, u_i]\delta_{ij}. \tag{68}$$

(We have defined $\tilde{T} \equiv -T$.)

So our optimization problem is to find $\hat{K}$ which solves the following optimization problem:

$$\begin{aligned}
&\text{Maximize} \\
&\hat{\mathcal{O}} \;=\; \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} ||A^{-1}(\mathrm{u},\tilde{T})||_2^{-1} \\
&\text{with respect to (w.r.t.)} \\
&\hat{K},\ \text{subject to} \\
&\hat{\mathcal{C}} \;=\; \Big(\max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}}\ \max_{\hat{A}\subset A} ||\hat{A}||_2 \le 1/\tau_{\text{fast}} \\
&\qquad \text{and}\ \ \hat{K}_{,wu} = \hat{K}_{,uw}\ \ \text{and}\ \ \hat{K}_{,w} \ge 0\ \ \text{and}\ \ \hat{K}[0,u] = 0\Big) \\
&\qquad \text{where} \\
&\qquad \tilde{\mathcal{T}} = \{\tilde{T} | \sigma_1(\tilde{T}) \le t_{\max}\ \text{and}\ \tilde{T}\ \text{is positive semi-definite}\},\ \text{and} \\
&\qquad -A_{ij} = \hat{K}_{,w}[w_i, v_i]\left(\tilde{T}_{ij}g'(u_i) + \delta_{ij}\right) - \hat{K}_{,u}[w_i, u_i]\delta_{ij}
\end{aligned} \tag{69}$$

and $\sigma_1(\tilde{T})$ is the largest singular value of $\tilde{T}$, i.e. the largest absolute value of any eigenvalue of $T$.

By introducing new notation

$$\begin{aligned}
\mu[w,u] &= \hat{K}_{,w}[w,v] \\
\nu[w,u] &= -\hat{K}_{,u}[w,v]
\end{aligned} \tag{70}$$

and translating the constraints appropriately, we can treat $\mu$ and $\nu$ as *independent functions* except for the constraint on the mixed partial derivatives. Then the problem (69) is equivalent to the following optimization problem:

$$\begin{aligned}
&\text{Maximize} \\
&\hat{\mathcal{O}} \;=\; \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} ||A^{-1}(\mathrm{u},\tilde{T})||_2^{-1} \\
&\text{w.r.t. } (\mu,\nu), \\
&\text{subject to} \\
&\hat{\mathcal{C}} \;=\; \Big(\max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}}\ \max_{\hat{A}\subset A} ||\hat{A}||_2 \le 1/\tau_{\text{fast}} \\
&\qquad \text{and}\ \ \mu_{,u} = -\nu_{,w}\ \ \text{and}\ \ \mu \ge 0\ \ \text{and}\ \ \nu[0,u] = 0\Big) \\
&\qquad \text{where} \\
&\qquad \tilde{\mathcal{T}} = \{\tilde{T} | \sigma_1(\tilde{T}) \le t_{\max}\ \text{and}\ \tilde{T}\ \text{is positive semi-definite}\},\ \text{and} \\
&\qquad -A_{ij} = \mu[w_i, v_i]\left(\tilde{T}_{ij}g'(u_i) + \delta_{ij}\right) + \nu[w_i, u_i]\delta_{ij}.
\end{aligned} \tag{71}$$

In the next section we will establish an approximate solution to this optimization problem: a $(\mu, \nu)$ pair that satisfies all the constraints and comes within a factor of 2

of the globally optimal value of $\hat{\mathcal{O}}$. Here we simply make several observations about the optimization problem (71).

First, one of the most important questions about this problem, and our solution to it, is whether the restriction to positive semi-definite $\tilde{T}$'s can be removed. Connection matrices appearing in real applications can have bounded singular values, but rarely are all the eigenvalues of the same sign. Second, we note the close relation of this problem to a worst-case minimization of the condition number of $A$, $\kappa(A) = ||A||_2||A^{-1}||_2$. Since $\max_{ij}|a_{ij}| \leq ||A||_2$ and $\mu$ and $\nu$ can easily be rescaled by a constant while preserving their constraints, the two problems look quite similar. Indeed, maximizing $\kappa(A)$ over all $u, w, \tilde{T} \in \tilde{T}$ subject to the $\mu$ and $\nu$ constraints would yield an upper bound of $\tau_{\text{fast}}\kappa_{\max}$ for $\mathcal{O}_{\max}$. But our problem is more difficult because the extremization over $u, w, \tilde{T} \in \tilde{T}$ is performed separately for the constraint and the objective.

### 3.2.1 Optimization of $\mu$ and $\nu$

A useful auxiliary problem to (71) is obtained by replacing (54) with the non-invariant expression (57):

$$\begin{aligned}
&\text{Maximize} \\
&\qquad \mathcal{O} \;=\; \min_{u,w,\tilde{T}\in\tilde{T}} ||A^{-1}(\mathrm{u},\tilde{T})||_2^{-1} \\
&\text{w.r.t. } (\mu,\nu), \\
&\text{subject to} \\
&\qquad \mathcal{C}(c) \;=\; \Big(c \max_{u,w,\tilde{T}\in\tilde{T}} \max_{ij} |A_{ij}(\mathrm{u},\tilde{T})| \leq 1/\tau_{\text{fast}} \\
&\qquad\qquad \text{and } \mu_{,u} = -\nu_{,w} \text{ and } \mu \geq 0 \text{ and } \nu[0,u] = 0\Big) \\
&\qquad \text{where} \\
&\qquad\qquad \tilde{T} = \{\tilde{T} | \sigma_1(\tilde{T}) \leq t_{\max} \text{ and } \tilde{T} \text{ is positive semi-definite}\}, \text{and} \\
&\qquad\qquad -A_{ij} = \mu[w_i, v_i]\Big(\tilde{T}_{ij} g'(u_i) + \delta_{ij}\Big) + \nu[w_i, u_i]\delta_{ij}.
\end{aligned} \tag{72}$$

Unlike the original problem (71), we will be able to solve this auxiliary problem exactly.

To solve the constrained maximization problem (72) and others like it, we will use the following proof strategy. Given objective $\mathcal{O}$ and constraints $\mathcal{C}$, we will maximize some lower bound objective $\mathcal{O}_-$ such that $\mathcal{O}_-[\mu,\nu] \leq \mathcal{O}[\mu,\nu]$, subject to *tightened* constraints $\mathcal{C}_-$ such that $\mathcal{C}_-[\mu,\nu] \Rightarrow \mathcal{C}[\mu,\nu]$. In this way we ensure that $\max(\mathcal{O}_-|\mathcal{C}_-) \leq \max(\mathcal{O}|\mathcal{C})$. Likewise we will maximize some upper bound objective $\mathcal{O}_+$ such that $\mathcal{O}[\mu,\nu] \leq \mathcal{O}_+[\mu,\nu]$, subject to *loosened* constraints $\mathcal{C}_+$ such that $\mathcal{C}[\mu,\nu] \Rightarrow \mathcal{C}_+[\mu,\nu]$; this combination ensures that $\max(\mathcal{O}|\mathcal{C}) \leq \max(\mathcal{O}_+|\mathcal{C}_+)$. Having solved both constrained optimizations, we will see that both give the same value for the objective:

$$\max(\mathcal{O}_+|\mathcal{C}_+) = \max(\mathcal{O}_-|\mathcal{C}_-) \tag{73}$$

which implies that all the extremal values are the same:

$$\max(\mathcal{O}|\mathcal{C}) = \max(\mathcal{O}_-|\mathcal{C}_-) = \max(\mathcal{O}_+|\mathcal{C}_+). \tag{74}$$

Furthermore, the extremal values $\mu_-^*$ and $\nu_-^*$ of $\max(\mathcal{O}_-[\mu,\nu]|\mathcal{C}_-[\mu,\nu])$ all satisfy constraints $\mathcal{C}$ (since they satisfy $\mathcal{C}_-$) and thus constitute extremal values of $\max(\mathcal{O}[\mu,\nu])$

as well. Thus we will have solved the original constrained optimization problem of maximizing $\mathcal{O}$ with respect to $\mathcal{C}$, by finding the maximal value and arguments $\mu^*, \nu^*$) at which the maximum is attained.

In the next section we will use this proof strategy to solve the auxiliary optimization problem (72). A variant of the same argument can then be used to conclude that the $(\mu, \nu)$ pair for the $c = 2$ auxiliary problem comes within a factor of two of solving the original optimization problem (71).

In fact, using (56), we see that the $c = 1$ version of (72) is a upper bound for (71) and the $c = 2$ version is an lower bound. In other words,

$$\max(\mathcal{O}|\mathcal{C}(c=2)) \leq \max(\hat{\mathcal{O}}|\hat{\mathcal{C}}) \leq \max(\mathcal{O}|\mathcal{C}(c=1)). \tag{75}$$

Furthermore, $\mathcal{C}(c = 2)$ implies $\hat{\mathcal{C}}$ so that the extremal $(\mu^*, \nu^*)$ for $\max(\mathcal{O}|\mathcal{C}(c = 2))$ are in the constraint set for $\max(\hat{\mathcal{O}}|\hat{\mathcal{C}})$. As it will turn out, $\max(\mathcal{O}|\mathcal{C}(c))$ is proportional to $1/c$, so $\hat{\mathcal{O}}(\mu^*, \nu^*) = \mathcal{O}(\mu^*, \nu^*)$ is proven to be within a factor of two of its optimal value, $\max(\hat{\mathcal{O}}|\hat{\mathcal{C}})$. In other words,

$$\mathcal{O}(\mu^*, \nu^*) \equiv \max(\mathcal{O}|\mathcal{C}(c=2)) \leq \max(\hat{\mathcal{O}}|\hat{\mathcal{C}}) = 2\max(\mathcal{O}|\mathcal{C}(c=2)) \tag{76}$$

which implies

$$(1/2)\max(\hat{\mathcal{O}}|\hat{\mathcal{C}}) \leq \mathcal{O}(\mu^*, \nu^*), \equiv \max(\mathcal{O}|\mathcal{C}(c=2)) \tag{77}$$

and $(\mu^*, \nu^*)$ is an approximate solution (satisfying the constraints and optimizing the objective to within a factor of two) of the meta-optimization problem (71) or equivalently (69).

### 3.2.2 Solution of the Auxiliary Problem

We may solve the auxiliary problem for $c = 1$, then scale it to any other $c$ by scaling $\tau_{\text{fast}}$ appropriately. So we'll assume $c = 1$ in the following solution of (72). The basic strategy will be to obtain upper bounds by restricting consideration to diagonal connection matrices $T$, and to compare these upper bounds with lower bounds that follow from matrix theory. In some cases, we will find it useful to repeat the above reasoning to solve the bounding constrained optimization problems themselves. For example, $\max(\mathcal{O}_-|\mathcal{C}_-)$ will be found by way of $\max(\mathcal{O}_{--}|\mathcal{C}_{--})$ and $\max(\mathcal{O}_{-+}|\mathcal{C}_{-+})$. But first we will treat the upper bound $\max(\mathcal{O}_+|\mathcal{C}_+)$.

By simply restricting the class $\tilde{T}$ in problem (72) to the subset $\tilde{T}_+$ of $\tilde{T}$ matrices which are also diagonal, we simultaneously increase the value of $\mathcal{O}[\mu, \nu]$ (since it's a minimum over a proper subset of $\tilde{T} \in \tilde{T}$) and loosen the constraint $\mathcal{C}[\mu, \nu]$. So one

lower bound optimization problem is:

$$
\begin{aligned}
&\text{Maximize-}\\
&\quad \mathcal{O}_{+1} \;=\; \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} ||A^{-1}(\mathrm{u},\tilde{T})||_2^{-1}\\
&\text{w.r.t. } (\mu,\nu),\\
&\text{subject to}\\
&\quad \mathcal{C}_{+1} \;=\; \Big( \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} \max_{ij} |A_{ij}(\mathrm{u},\tilde{T})| \le 1/\tau_{\text{fast}}\\
&\qquad\qquad \text{and } \mu_{,u} = -\nu_{,w} \text{ and } \mu \ge 0 \text{ and } \nu[0,u] = 0\Big)\\
&\qquad \text{where}\\
&\qquad\qquad \tilde{\mathcal{T}}_+ = \{\tilde{T} | \tilde{T} \text{ is diagonal and } \sigma_1(\tilde{T}) \le t_{\max} \text{ and } \tilde{T} \text{ is positive semi-definite}\}, \text{ and}\\
&\qquad\qquad -A_{ij} = \mu[w_i,v_i]\Big(\tilde{T}_{ij}g'(u_i) + \delta_{ij}\Big) + \nu[w_i,u_i]\delta_{ij}.
\end{aligned}
\tag{78}
$$

This will not be the sought-after $\mathcal{O}_+$ and $\mathcal{C}_+$, but it moves in the right direction since $\mathcal{O} \le \mathcal{O}_{+1}$ and $\mathcal{C} \Rightarrow \mathcal{C}_{+1}$.

If $\tilde{T}$ is diagonal then so is $A$. For a diagonal matrix $A = \mathrm{diag}(a_i)$, $||A^{-1}||^{-1} = \min_i |a_i|$ and $\max_{ij} |A_{ij}| = \max_i |a_i|$. So we can calculate more detailed bounds:

$$
\begin{aligned}
\mathcal{O}_{+1} &= \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} \min_i \Big|\mu_i(\tilde{T}_{ii}g_i' + 1) + \nu_i\Big| \\
&\le \min_{u,\tilde{T}\in\tilde{\mathcal{T}}_+} \min_i \Big|\mu_i(\tilde{T}_{ii}g_i' + 1) + \nu_i\Big|_{w=0} \\
&= \min_{u,\tilde{T}\in\tilde{\mathcal{T}}_+} \min_i \Big|\mu_i(\tilde{T}_{ii}g_i' + 1)\Big|_{w=0} && \text{(since } \nu[0,u] = 0\text{)} \\
&= \min_{u,\tilde{T}\in\tilde{\mathcal{T}}_+} \min_i \mu_i(|\tilde{T}_{ii}|g_i' + 1)\Big|_{w=0} \\
&= \min_u \min_i \mu_i\Big((\min_{\tilde{T}\in\tilde{\mathcal{T}}_+} |\tilde{T}_{ii}|)g_i' + 1\Big)\Big|_{w=0} && \text{(since } \mu \ge 0 \text{ and } g_i' \ge 0\text{)} \\
&= \min_u \min_i \mu[0,u_i] && \text{(since } \min_{\tilde{T}\in\tilde{\mathcal{T}}_+} |\tilde{T}_{ii}| = 0\text{)} \\
&= \min_u \mu[0,u] \\
&\equiv \mathcal{O}_+[\mu,\nu].
\end{aligned}
\tag{79}
$$

Likewise we can bound the main constraint of $\mathcal{C}_{+1}$, which is that $\hat{\mathcal{C}}_{+1} \le 1/\tau_{\text{fast}}$, where

$$
\begin{aligned}
\hat{\mathcal{C}}_{+1} &= \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} \max_i \Big|\mu_i(\tilde{T}_{ii}g_i' + 1) + \nu_i\Big| \\
&\ge \max_{u,\tilde{T}\in\tilde{\mathcal{T}}_+} \max_i \Big|\mu_i(\tilde{T}_{ii}g_i' + 1) + \nu_i\Big|_{w=0} \\
&= \max_{u,\tilde{T}\in\tilde{\mathcal{T}}_+} \max_i \Big|\mu_i(\tilde{T}_{ii}g_i' + 1)\Big|_{w=0} && \text{(since } \nu[0,u] = 0\text{)} \\
&= \max_{u,\tilde{T}\in\tilde{\mathcal{T}}_+} \max_i \mu_i(|\tilde{T}_{ii}|g_i' + 1)\Big|_{w=0} \\
&= \max_u \max_i \mu_i\Big((\max_{\tilde{T}\in\tilde{\mathcal{T}}_+} |\tilde{T}_{ii}|)g_i' + 1\Big)\Big|_{w=0} && \text{(since } \mu \ge 0 \text{ and } g_i' \ge 0\text{)} \\
&= \max_u \max_i \mu[0,u_i]\Big(t_{\max}g'(u_i) + 1\Big) && \text{(since } \max_{\tilde{T}\in\tilde{\mathcal{T}}_+} |\tilde{T}_{ii}| = t_{\max}\text{)} \\
&= \max_u \mu[0,u]\Big(t_{\max}g'(u) + 1\Big) \\
&\equiv \hat{\mathcal{C}}_+[\mu,\nu].
\end{aligned}
\tag{80}
$$

So the upper bound optimization problem becomes

$$\begin{array}{rcl} \text{Maximize} & & \\ \mathcal{O}_+ & = & \min_u \mu[0,u] \\ \text{w.r.t. } (\mu,\nu), & & \\ \text{subject to} & & \\ \mathcal{C}_+ & = & \Big(\max_u \mu[0,u]\Big(t_{\max} g'(u)+1\Big) \le 1/\tau_{\text{fast}} \\ & & \quad \text{and } \mu_{,u} = -\nu_{,w} \text{ and } \mu \ge 0 \text{ and } \nu[0,u]=0\Big) \end{array} \tag{81}$$

To this optimization problem we propose the solution $(\mu_+^*, \nu_+^*)$:

$$\begin{array}{rcl} \mu_+^*[w,u] & = & 1/\tau_{\text{fast}}(t_{\max} g_0 + 1) \\ \nu_+^*[w,u] & = & 0, \end{array} \tag{82}$$

where $g_0 \equiv \max_u g'(u)$. These values for $\mu$ and $\nu$ are *constant*, i.e. independent of $w$ and $u$, so the mixed partial derivative constraint of problem (81) is trivially satisified. Clearly also $\mu_+^* \ge 0$ and $\nu_+^*[0,u] = 0$ are satisfied. The $\hat{C}_+ \le 1/\tau_{\text{fast}}$ constraint can also be verified:

$$\hat{C}_+[\mu_+^*, \nu_+^*] = \max_u \mu_+^*[0,u]\Big(t_{\max} g'(u)+1\Big) = \frac{\max_u \Big(t_{\max} g'(u)+1\Big)}{\tau_{\text{fast}}(t_{\max} g_0 + 1)} = 1/\tau_{\text{fast}}. \tag{83}$$

So $(\mu_+^*, \nu_+^*)$ satsifies the desired constraints. The objective is $\mathcal{O}_+[\mu_+^*, \nu_+^*] = \min_u \mu_+^*[0,u] = 1/\tau_{\text{fast}}(t_{\max} g_0 + 1)$. But from the constraints we see this value is also an upper bound for $\mathcal{O}_+[\mu,\nu]$ as follows:

$$\begin{array}{rcl} 1/\tau_{\text{fast}} & \ge & \hat{C}_+[\mu,\nu] \\ & = & \max_u \mu[0,u]\Big(t_{\max} g'(u)+1\Big) \\ & \ge & (\min_u \mu[0,u])\Big(\max_u (t_{\max} g'(u)+1)\Big) \\ & = & \mathcal{O}_+[\mu,\nu]\Big(t_{\max} g_0 + 1\Big), \end{array} \tag{84}$$

which implies $\mathcal{O}_+ \le 1/\tau_{\text{fast}}(t_{\max} g_0 + 1)$. So $(\mu_+^*, \nu_+^*)$ in (82) solves problem (81).

Next, we use matrix theory to find and solve a constrained optimization problem $\max(\mathcal{O}_-|\mathcal{C}_-)$ which can serve as a lower bound for $\max(\mathcal{O}|\mathcal{C})$.

To bound $\mathcal{O}$ below (in problem (72)), we must simplify $||A^{-1}||_2^{-1}$. In matrix notation, $||A^{-1}||_2^{-1}$ is just $\sigma_n(A)$, the smallest singular value of $A$. Also $A$ is given by the matrix expression

$$A = \text{diag}(\mu)(\tilde{T}\text{diag}(g') + 1) + \text{diag}(\nu). \tag{85}$$

The smallest singular value $\sigma_n(M+N)$ of a sum of matrices $M$ and $N$ is bounded below by $\sigma_n(M) - \sigma_1(N)$, as shown for example in [GL83] (Cor. 8.3-2, p.286). We will take $A = M + N$ with $N = \text{diag}(\nu)$ and use $\sigma_1(\text{diag}(\nu)) = \max_i |\nu_i|$ to find a lower bound $\mathcal{O}_-$ for $\mathcal{O}$:

$$\mathcal{O} \ge \mathcal{O}_- \equiv \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \Big[\sigma_n\Big(\text{diag}(\mu)(\tilde{T}\text{diag}(g') + 1)\Big) - \max_i |\nu_i|\Big]. \tag{86}$$

We can also bound the main constraint of $\mathcal{C}$, which is that $\hat{\mathcal{C}} \leq 1/\tau_{\text{fast}}$. We will use the fact that $\sigma_1(M+N) \leq \sigma_1(M) + \sigma_1(N)$, which is also shown in [GL83] (Cor. 8.3-2, p.286). The bound is as follows:

$$\begin{aligned}
\hat{\mathcal{C}}[\mu,\nu] &= \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \max_{ij} |A_{ij}| \\
&\leq \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} ||A||_2 \\
&\quad \text{(standard matrix norm bounds, e.g. [GL83], 2.2-10, p. 15)} \\
&= \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \sigma_1(A) \\
&\leq \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \left[\sigma_1\Big(\text{diag}(\mu)(\tilde{T}\,\text{diag}(g') + 1)\Big) + \max_i |\nu_i|\right] \\
&\equiv \hat{\mathcal{C}}_-[\mu,\nu].
\end{aligned} \tag{87}$$

So the lower bound optimization problem becomes

$$\begin{aligned}
&\text{Maximize} \\
&\qquad \mathcal{O}_- = \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \left[\sigma_n\Big(\text{diag}(\mu)(\tilde{T}\,\text{diag}(g') + 1)\Big) - \max_i |\nu_i|\right] \\
&\text{w.r.t. } (\mu,\nu), \\
&\text{subject to} \\
&\qquad \mathcal{C}_- = \Big(\max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \left[\sigma_1\Big(\text{diag}(\mu)(\tilde{T}\,\text{diag}(g') + 1)\Big) + \max_i |\nu_i|\right] \leq 1/\tau_{\text{fast}} \\
&\qquad\qquad \text{and } \mu_{,u} = -\nu_{,w} \text{ and } \mu \geq 0 \text{ and } \nu[0,u] = 0\Big).
\end{aligned} \tag{88}$$

Consider the related optimization problem

$$\begin{aligned}
&\text{Maximize} \\
&\qquad \mathcal{O}_{-+1} = \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \left[\sigma_n\Big(\text{diag}(\mu)(\tilde{T}\,\text{diag}(g') + 1)\Big) - \max_i |\nu_i|\right] \\
&\text{w.r.t. } (\mu,\nu), \\
&\text{subject to} \\
&\qquad \mathcal{C}_{-+1} = \Big(\max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \left[\sigma_1\Big(\text{diag}(\mu)(\tilde{T}\,\text{diag}(g') + 1)\Big) + \max_i |\nu_i|\right] \leq 1/\tau_{\text{fast}} \\
&\qquad\qquad \text{and } \mu \geq 0 \text{ and } \nu[0,u] = 0\Big),
\end{aligned} \tag{89}$$

which differs from (88) by removing the partial derivative constraint that relates $\mu$ and $\nu$. Clearly if we solve this problem and find a solution that also obeys the partial derivative constraint, then we will have solved the original problem. That is what we will do. But the new problem (89) can be further simplified by observing that the optimal $\nu^*_{-+1}$ must be identically zero; otherwise, an optimal $(\mu^*_{-+1}, \nu^*_{-+1} \neq 0)$ would have a lower value of the objective than $(\mu^*_{-+1}, 0)$ which equally well satisfies the constraint $\mathcal{C}_{-+1}$; that would contradict the assumed optimality of $(\mu^*_{-+1}, \nu^*_{-+1} \neq 0)$.

So to solve $\max(\mathcal{O}_-|\mathcal{C}_-)$, i.e. problem (88), it suffices to (a) solve problem (89) assuming $\nu = 0$, i.e. to solve:

$$\begin{aligned}
&\text{Maximize} \\
&\qquad \mathcal{O}_{-+} = \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \sigma_n\Big(\text{diag}(\mu)(\tilde{T}\,\text{diag}(g') + 1)\Big) \\
&\text{w.r.t. } (\mu,\nu), \\
&\text{subject to} \\
&\qquad \mathcal{C}_{-+} = \Big(\max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \sigma_1\Big(\text{diag}(\mu)(\tilde{T}\,\text{diag}(g') + 1)\Big) \leq 1/\tau_{\text{fast}} \\
&\qquad\qquad \text{and } \mu \geq 0\Big),
\end{aligned} \tag{90}$$

and then (b) verify that the mixed derivative constraint $\mu_{,u} = -\nu_{,w} (= 0)$ is satisfied by the solution $(\mu^*_{-+}, 0)$ to (90). Furthermore, the optimizing values $(\mu^*_-, \nu^*_-)$ will just be $(\mu^*_{-+}, 0)$.

We will solve $\max(\mathcal{O}_{-+}|\mathcal{C}_{-+})$ using the same strategy as for $\max(\mathcal{O}|\mathcal{C})$ itself: by constructing an upper bound problems by restricting to diagonal connection matrices $\tilde{T} \in \tilde{\mathcal{T}}_+ = \tilde{\mathcal{T}} \cap \{\text{diagonal matrices}\}$, and a lower bound problem using more matrix theory, and showing that they have a common solution.

The upper bound for $\mathcal{O}_{-+}$ is calculated as follows:

$$\begin{aligned}
\mathcal{O}_{-+} &= \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \sigma_n\Big(\mathrm{diag}(\mu)(\tilde{T}\,\mathrm{diag}(g') + 1)\Big) \\
&\leq \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} \sigma_n\Big(\mathrm{diag}(\mu)(\tilde{T}\,\mathrm{diag}(g') + 1)\Big) \\
&= \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} \min_i \Big|\mu_i(\tilde{T}_{ii}g'_i + 1)\Big| \\
&\leq \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} \min_i \mu_i(|\tilde{T}_{ii}|g'_i + 1) \\
&= \min_{u,w} \min_i \mu_i\Big(\min_{\tilde{T}\in\tilde{\mathcal{T}}_+} |\tilde{T}_{ii}|g'_i + 1\Big) \\
&= \min_{u,w} \min_i \mu[w_i, u_i] \qquad (\text{since } \min_{\tilde{T}\in\tilde{\mathcal{T}}_+} |\tilde{T}_{ii}| = 0) \\
&= \min_{u,w} \mu[w, u] \\
&\equiv \mathcal{O}_{-++}.
\end{aligned} \tag{91}$$

The corresponding (lower) bound for $\hat{\mathcal{C}}_{-+}$ is calculated as follows:

$$\begin{aligned}
\hat{\mathcal{C}}_{-+} &= \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \sigma_1\Big(\mathrm{diag}(\mu)(\tilde{T}\,\mathrm{diag}(g') + 1)\Big) \\
&\geq \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} \sigma_1\Big(\mathrm{diag}(\mu)(\tilde{T}\,\mathrm{diag}(g') + 1)\Big) \\
&= \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} \max_i \Big|\mu_i(\tilde{T}_{ii}g'_i + 1)\Big| \\
&= \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}_+} \min_i \mu_i(\tilde{T}_{ii}g'_i + 1) \qquad (\text{since } \tilde{T}_{ii} \geq 0) \\
&= \max_{u,w} \min_i \mu_i\Big(\max_{\tilde{T}\in\tilde{\mathcal{T}}_+} \tilde{T}_{ii}g'_i + 1\Big) \\
&= \max_{u,w} \min_i \mu[w_i, u_i]\Big(t_{\max}g'(u_i) + 1\Big) \qquad (\text{since } \max_{\tilde{T}\in\tilde{\mathcal{T}}_+} \tilde{T}_{ii} = t_{\max}) \\
&= \max_{u,w} \mu[w, u]\Big(t_{\max}g'(u) + 1\Big) \\
&\equiv \hat{\mathcal{C}}_{-++}.
\end{aligned} \tag{92}$$

So the upper bound optimization problem becomes similar to problem (81):

$$\begin{aligned}
&\text{Maximize} \\
&\qquad \mathcal{O}_{-++} = \min_{u,w} \mu[w, u] \\
&\text{w.r.t. } (\mu, \nu), \\
&\text{subject to} \\
&\qquad \mathcal{C}_{-++} = \Big(\max_{u,w} \mu[w, u]\Big(t_{\max}g'(u) + 1\Big) \leq 1/\tau_{\text{fast}} \\
&\qquad\qquad \text{and } \mu \geq 0\Big).
\end{aligned} \tag{93}$$

To this optimization problem we again propose the solution (cf. equation (82))

$$\mu^*_{-++}[w, u] = 1/\tau_{\text{fast}}(t_{\max}g_0 + 1), \tag{94}$$

where $g_0 \equiv \max_u g'(u)$. The proof for this solution is the same as that of the solution of problem (81) by equation (82), except that now $w$ must be optimized everywhere $u$ is. This establishes the solution of problem (93) by equation (94).

We must now find a lower bound $\max(\mathcal{O}_{-+-}|\mathcal{C}_{-+-})$ for $\max(\mathcal{O}_{-+}|\mathcal{C}_{-+})$, and to do so, we require another matrix theory result: that for positive semi-definite matrices $M$ and $N$, $\sigma_n(M+N) \geq \sigma_n(M) + \sigma_n(M)$ [SgS90].

(Note on the proof so far: We could not use this result earlier since $\mathrm{diag}(\nu)$ was not positive semi-definite. Also the use of this result and equation (92) are the only places in the proof that depend on the assumption that $\tilde{T}$ is positive semi-definite.)

Thus,

$$\begin{aligned}
\mathcal{O}_{-+} &= \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \sigma_n\Big(\mathrm{diag}(\mu)\tilde{T}\mathrm{diag}(g') + \mathrm{diag}(\mu)\Big) \\
&\geq \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \Big[\sigma_n\Big(\mathrm{diag}(\mu)\tilde{T}\mathrm{diag}(g')\Big) + \sigma_n(\mathrm{diag}(\mu))\Big] \\
&\geq \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \Big[\sigma_n(\mathrm{diag}(\mu))\sigma_n(\tilde{T})\sigma_n(\mathrm{diag}(g')) + \sigma_n(\mathrm{diag}(\mu))\Big] \\
&\qquad \text{(since } ||MN||_2 \leq ||M||_2||N||_2, \text{ [GL83] p.16)} \\
&= \min_{u,w} \Big[\sigma_n(\mathrm{diag}(\mu))\Big(\min_{\tilde{T}\in\tilde{\mathcal{T}}} \sigma_n(\tilde{T})\Big)\sigma_n(\mathrm{diag}(g')) + \sigma_n(\mathrm{diag}(\mu))\Big] \\
&= \min_{u,w} \min_i \mu[w_i, u_i] \\
&\qquad \text{(since } \min_{\tilde{T}\in\tilde{\mathcal{T}}} \sigma_n(\tilde{T}) = 0) \\
&= \min_{u,w} \mu[w, u] \\
&\equiv \mathcal{O}_{-+-}[\mu].
\end{aligned} \tag{95}$$

Likewise,

$$\begin{aligned}
\hat{\mathcal{C}}_{-+} &= \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \sigma_1\Big(\mathrm{diag}(\mu)\tilde{T}\mathrm{diag}(g') + \mathrm{diag}(\mu)\Big) \\
&\leq \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \Big[\sigma_1\Big(\mathrm{diag}(\mu)\tilde{T}\mathrm{diag}(g')\Big) + \sigma_1(\mathrm{diag}(\mu))\Big] \\
&\qquad \text{(since } ||M+N||_2 \leq ||M||_2 + ||N||_2, \text{ [GL83] Cor 8.3-2)} \\
&\leq \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} \Big[\sigma_1(\mathrm{diag}(\mu))\sigma_1(\tilde{T})\sigma_1(\mathrm{diag}(g')) + \sigma_1(\mathrm{diag}(\mu))\Big] \\
&\qquad \text{(since } ||MN||_2 \leq ||M||_2||N||_2, \text{ [GL83] p.16)} \\
&= \max_{u,w} \sigma_1(\mathrm{diag}(\mu))\Big(\Big(\max_{\tilde{T}\in\tilde{\mathcal{T}}} \sigma_1(\tilde{T})\Big)\sigma_1(\mathrm{diag}(g')) + 1\Big) \\
&= \max_{u,w}(\max_i \mu[w_i, u_i])\Big(t_{\max}(\max_i g'(u_i) + 1\Big) \\
&\qquad \text{(since } \max_{\tilde{T}\in\tilde{\mathcal{T}}} \sigma_1(\tilde{T}) = t_{\max}) \\
&\equiv \hat{\mathcal{C}}_{-+-}[\mu].
\end{aligned} \tag{96}$$

We can assemble these bounds into the constrained optimization problem

$$\begin{aligned}
&\text{Maximize} \\
&\qquad \mathcal{O}_{-+-} = \min_{u,w} \mu[w, u] \\
&\text{w.r.t. } (\mu, \nu), \\
&\text{subject to} \\
&\qquad \mathcal{C}_{-+-} = \Big(\max_{u,w}(\max_i \mu[w_i, u_i])\Big(t_{\max}(\max_i g'(u_i) + 1\Big) \leq 1/\tau_{\text{fast}} \\
&\qquad\qquad \text{and } \mu \geq 0\Big).
\end{aligned} \tag{97}$$

To this optimization problem we once again propose the constant solution (cf. equation (82))

$$\mu^*_{-+-}[w,u] = 1/\tau_{\text{fast}}(t_{\max}g_0 + 1), \tag{98}$$

where $g_0 \equiv \max_u g'(u)$. Clearly the constraint $\mu^*_{-+-} \geq 0$ is satisfied. The $\hat{C}_{-+-} \leq 1/\tau_{\text{fast}}$ constraint can also be verified:

$$\hat{C}_{-+-}[\mu^*_{-+-}] = \max_{w,u} \mu^*_{-+-}[w,u]\Big(t_{\max}g'(u)+1\Big) = \frac{\max_u \Big(t_{\max}g'(u)+1\Big)}{\tau_{\text{fast}}(t_{\max}g_0+1)} = 1/\tau_{\text{fast}}. \tag{99}$$

So $\mu^*_{-+-}$ satsifies the desired constraints. The objective is $\mathcal{O}_{-+-}[\mu^*_{-+-}] = \min_{w,u} \mu^*_{-+-}[w,u] = 1/\tau_{\text{fast}}(t_{\max}g_0+1)$. But, once again, from the constraints we know this value is also an upper bound for $\mathcal{O}_{-+-}[\mu]$:

$$\begin{aligned} 1/\tau_{\text{fast}} &\geq \hat{C}_{-+-}[\mu] \\ &= \max_{w,u} \mu[w,u]\Big(t_{\max}g'(u)+1\Big) \\ &\geq \big(\min_{w,u} \mu[w,u]\big)\Big(\max_{w,u}(t_{\max}g'(u)+1)\Big) \\ &= \mathcal{O}_{-+-}[\mu,\nu]\Big(t_{\max}g_0+1\Big), \end{aligned} \tag{100}$$

which implies $\mathcal{O}_{-+-} \leq 1/\tau_{\text{fast}}(t_{\max}g_0+1)$. So equation $\mu^*_{-+-}$ in (98) solves problem (97).

We have previously solved problem $\mu^*_{-++}$ in (93) with equation (94). The resulting maximal values of $\mathcal{O}$ are the same for the two problems (97) and (93) ($\max(\mathcal{O}_{-+-}|\mathcal{C}_{-+-}) = \max(\mathcal{O}_{-++}|\mathcal{C}_{-++}) = 1/\tau_{\text{fast}}(t_{\max}g_0+1)$), and they are attained by the same $\mu^* =$ constant functions. Since these were lower and upper bounds for $\max(\mathcal{O}_{-+}|\mathcal{C}_{-+})$, we conclude that the same $\mu^*$ and maximal value of $\mathcal{O}$ also solve problem (90), namely the calculation of $\max(\mathcal{O}_{-+}|\mathcal{C}_{-+})$. But in the discussion of problem (90) we pointed out that, if $\mu^*_{-+,u} = 0$ (as it certainly is, since $\mu^*_{-+}$ is a constant independent of both $u$ and $w$), then $(\mu^*_{-+}, \nu = 0)$ is also a solution $(\mu^*_-, \nu^*_-)$ of problem (88). This result is the sought-after lower bound for the original problem (72), and may be joined with the solution of (81) (an upper bound for (72)) by (82) to finish the entire problem: $\max(\mathcal{O}_-|\mathcal{C}_-) = \max(\mathcal{O}|\mathcal{C}) = \max(\mathcal{O}_+|\mathcal{C}_+) = 1/\tau_{\text{fast}}(t_{\max}g_0+1)$; and the optimum is attained at $(\mu^*,\nu^*) = (\mu^*_-,\nu^*_-) = (\mu^*_+,\nu^*_+)$, i.e.

$$\begin{aligned} \mu^*[w,u] &= 1/\tau_{\text{fast}}(t_{\max}g_0+1) \\ \nu^*[w,u] &= 0 \end{aligned} \tag{101}$$

is shown to be a solution of (72) for $c = 1$. Other values of $c$ may be absorbed into the definition of $\tau_{\text{fast}}$. So we have established Lemma 1:

**Lemma 1.** The optimization problem

$$\begin{aligned}
&\text{Maximize} \\
&\qquad \mathcal{O} \;=\; \min_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} ||A^{-1}(\mathbf{u},\tilde{T})||_2^{-1} \\
&\text{w.r.t } (\mu,\nu), \\
&\text{subject to} \\
&\qquad \mathcal{C}(c) \;=\; \Big( \max_{u,w,\tilde{T}\in\tilde{\mathcal{T}}} c \max_{ij} |A_{ij}(\mathbf{u},\tilde{T})| \le 1/\tau_{\text{fast}} \\
&\qquad\qquad \text{and } \mu_{,u} = -\nu_{,w} \text{ and } \mu \ge 0 \text{ and } \nu[0,u] = 0\Big) \\
&\qquad \text{where} \\
&\qquad\qquad \tilde{\mathcal{T}} = \{\tilde{T} | \sigma_1(\tilde{T}) \le t_{\max} \text{ and } \tilde{T} \text{ is positive semi-definite}\}, \text{ and} \\
&\qquad\qquad -A_{ij} = \mu[w_i,v_i]\Big(\tilde{T}_{ij} g'(u_i) + \delta_{ij}\Big) + \nu[w_i,u_i]\delta_{ij}
\end{aligned} \tag{102}$$

has as one solution

$$\begin{aligned}
\mu^*[w,u] &= 1/(c\tau_{\text{fast}}(t_{\max} g_0 + 1)) \\
\nu^*[w,u] &= 0.
\end{aligned} \tag{103}$$

It remains only to translate this solution for $\mu[w,u]$ and $\nu[w,u]$ back into a function $\hat{K}$ (as called for in (69)) and thence to the desired "kinetic energy" or "cost of movement" function $\tilde{K}[\dot{u},u]$ or its equivalent, $K[\dot{v},v]$.

### 3.2.3 Approximate Solution of the Meta-Optimization Problem

¿From equation (77), we can apply Lemma 1 with $c = 2$ to find a $(\mu^*,\nu^*)$ pair which comes within a factor of 2 of solving the meta-optimization problem (71) or equivalently (69). (Note that (77) was derived assuming that $\max(\mathcal{O}|\mathcal{C}(c))$ is proportional to $1/c$, which has now been established in Lemma 1.) Changing back to $\hat{K}$ notation,

$$\hat{K}_{,w} = 1/\tau_H, \quad \hat{K}_{,u} = 0, \tag{104}$$

where

$$\tau_H = 2\tau_{\text{fast}}(t_{\max} g_0 + 1) \tag{105}$$

is a constant. (The factor of 2 comes from $c = 2$.) The general solution of these partial differential equations is $\hat{K}[w,u] = w/\tau_H + c_1$, but from the statement of problem (69) we must take $\hat{K}[0,u] = c_1 = 0$. Then

$$\hat{K}[w,u] = w/\tau_H. \tag{106}$$

Using (65),

$$\tilde{K}_{,\dot{u}}[\dot{u},u] = \hat{K}^{-1}[\dot{u},u] g'(u) = \tau_H \dot{u} g'(u). \tag{107}$$

This has the solution $\tilde{K}[\dot{u},u] = (\tau_H/2)\dot{u}^2 g'(u) + c_2(u)$. But the term $c_2(u)$ has no effect on the dynamics, since its greedy derivative is zero, and without loss of generality we can take $c_2(u) = 0$. Then

$$\tilde{K}[\dot{u},u] = \frac{\tau_H}{2}\dot{u}^2 g'(u). \tag{108}$$

This is the sought-after kinetic energy or cost term for $\dot{u}$, and the associated equation of motion is (from equation (63))

$$\begin{aligned}
\dot{u}_i &= \frac{1}{\tau_H}\Big(\sum_j T_{ij} v_j + h_i - u_i\Big), \\
v_i &= g(u_i).
\end{aligned} \tag{109}$$

This $\tilde{K}$ may also be translated back to a Lagrangian expressed directly in terms of $\dot{v}_i$, using $K[\dot{v}, v] = \tilde{K}[\dot{u}(v), u(v)]$:

$$K[\dot{v}, v] = \frac{\tau_H}{2}\dot{v}^2 / g'(g^{-1}(v)), \tag{110}$$

or equivalently

$$K[\dot{v}, v] = \frac{\tau_H}{2}\dot{v}^2 \phi''(v). \tag{111}$$

If $g(u)$ is linear (i.e. if $\phi(v)$ is quadratic), this kinetic energy expression is proportional to the conventional $(m/2)\dot{v}^2$ expression encountered in physics, but for nonlinear $g$ this expression is different from a kinetic energy in physics. (110) is the circuit cost-of-movement (or kinetic energy) term used elsewhere in this paper, and a greedy variation of the associated action functional yields equations of motion equivalent to the Hopfield/Grossberg dynamics of (109).

Assembling Lemma 1 and (47), (48), (52), (58), (61), (77), (105), and (111), we have demonstrated the following theorem:

**Theorem 1.** The linearized dynamics determined by a greedy variation of the Lagrangian

$$\begin{array}{rcl} L[\dot{\mathrm{v}}] & = & \sum_i K[\dot{v}_i, v_i] + \sum_i E_{,i}\dot{v}_i, \text{ with} \\ E[\mathrm{v}] & = & -\frac{1}{2}\sum_{ij} T_{ij} v_i v_j - \sum_i h_i v_i + \sum_i \phi(v_i), \text{ and} \\ & & \phi'(v) = g^{-1}(v) \text{ and } g_0 = \max_u |g'(u)| \end{array} \tag{112}$$

may be computed to be

$$\begin{array}{rcl} \dot{\Delta v}_i & = & \tilde{K}[w_i, v_i] + \sum_j A_{ij}\Delta v_j \\ \text{where} & & \\ A_{ij} & = & \tilde{K}_{,w}[w_i, v_i]\Big(T_{ij} - \delta_{ij}\phi''(v_i)\Big) + \tilde{K}_{,v}\delta_{ij}, \text{ and} \\ & & w_i \equiv -E_{,i}, \text{ and } \tilde{K}[K[\dot{v}, v], v] = \dot{v}. \end{array} \tag{113}$$

If we define the objective

$$\mathcal{M}_\tau(K) = \min_{\mathrm{v}} \min_{T \in \mathcal{T}} ||A^{-1}(\mathrm{v}, T)||_2^{-1}, \tag{114}$$

where

$$\mathcal{T} = \{T | \sigma_1(T) \leq t_{\max} \text{ and } T \text{ is negative semi-definite}\}, \tag{115}$$

and if we impose the constraints on $K$ that

$$\begin{array}{ll} (a) & \max_{\mathrm{v}} \max_{T \in \mathcal{T}} \max_{\hat{A} \subset A} ||\hat{A}||_2 \leq 1/\tau_{\text{fast}}, \\ & \text{(where } \hat{A} \text{ runs over } 1 \times 1 \text{ and } 2 \times 2 \text{ submatrices of } A\text{), and} \\ (b) & K \text{ is continuous in its first and second derivatives,} \\ (c) & \tilde{K}[0, v] = 0, \text{ and} \\ (d) & \tilde{K}[w, v]_{,w} \leq 0, \end{array} \tag{116}$$

then the function

$$\begin{array}{rcl} K[\dot{v}, v] & = & (\tau_H/2)\dot{v}^2\phi''(v) \text{ where} \\ \tau_H & = & 2\tau_{\text{fast}}(t_{\max} g_0 + 1) \end{array} \tag{117}$$

satisfies the constraints and comes within a factor of two of the globally maximal value of $\mathcal{M}(K)$ subject to these constraints. Furthermore, the objective $\mathcal{M}_\tau$ and the constraints $(a) - (d)$ in (116), with definitions of $A$, $\tilde{K}$ and $w$ as in (113) are invariant with respect to coordinatewise reparameterizations $x_i = f_i(v_i)$ in which each $f_i$ is monotonically increasing, differentiable, and has a differentiable inverse.

### 3.2.4 Notes on the Solution

If $\phi$ differs from one neuron to the next, and is indexed by $i$ as $\phi_i$, then the optimal $K$ term will still have the above form if it too is allowed to depend on $i$. The proof in section 3.2.2 can easily be altered to establish this generalization of the result.

Note that (105) relates the fastest physical time scale $\tau_{\text{fast}}$ in a circuit to an optimal value of the neural time scale $\tau_H$ appearing in Hopfield's version of the analog neural network [Hop84], and the two are not the same. The best value for the neural time constant is the slowest time constant in the system. The ratio of the latter to the fastest time constant is roughly the product of the neural gain $g_0$ and largest eigenvalue of $\tilde{T}$.

We note a change of variable which simplifies the kinetic energy term in the above dynamics, for use in the next section:

$$\begin{aligned} &L[\dot{\mathbf{w}}] = \textstyle\sum_i \frac{1}{2}\dot{w}_i^2 + \sum_i \frac{\partial E}{\partial w_i}\dot{w}_i, \\ &\partial L/\partial \dot{w}_i(t) = 0 \Rightarrow \dot{w}_i + \partial E/\partial w_i = 0, \text{ i.e.} \\ &\dot{w}_i = -\partial E/\partial w_i \end{aligned} \tag{118}$$

which is supposed to be identical to $\dot{u}_i = -\partial E/\partial v_i$, $v_i = g(u_i)$ (cf. (12)). This can be arranged by choosing $w$:

$$\begin{aligned} &\tfrac{dw_i}{du_i}\dot{u}_i = -\tfrac{\partial E}{\partial v_i}\tfrac{dv_i}{dw_i} \\ &\Rightarrow \tfrac{dw_i}{du_i} = \tfrac{dv_i}{dw_i} = \tfrac{dv_i/du_i}{dw_i/dv_i} \\ &\Rightarrow \tfrac{dw_i}{du_i} = \sqrt{g'(u_i)} \end{aligned} \tag{119}$$

i.e.

$$w_i = \int^{u_i} du\sqrt{g'(u)} \text{ and } v_i = \int^{w_i} dw\sqrt{g'(u(w))}. \tag{120}$$

## 4 Discussion and Conclusions

We introduced a Lagrangian formulation of the relaxation dynamics of neural networks which compute by optimizing an objective function in a standard neural network form. This optimization involves a trading-off cost and functionality in the formulation of optimization problems. The Lagrangian formulation makes novel use of a *greedy functional derivative*, which we defined and computed. With these tools we demonstrated the use of three levels of optimization in the design of relaxation neural network dynamics: the original objective $E$, the Lagrangian $L$, and a meta-objective $\mathcal{M}$ which measures cost and functionality over many trials of the network.

Applications of the Lagrangian formulation were divided into two broad groups: analog circuit Lagrangians, and Lagrangians that require a hidden switching mechanism to implement as a circuit. At the circuit level, we showed that a limited meta-optimality criterion is nearly optimized (within a factor of two of the global optimum) by a Lagrangian corresponding to the conventional Hopfield-Grossberg continuous-time analog neural network dynamics; we also provided several alternative Lagrangians which might be preferable under less analytically tractable meta-optimality criteria. In part II of this work we shall introduce a generalization of such relaxation Lagrangians to cyclic Lagrangians with clocked objective functions,

which have a simple circuit implementation involvi
present suitable algebraic notation including a *clock*
use the notation concisely to express neural netwo
minimization and for relaxation networks that con

## Acknowledgements

We wish to acknowledge Charles Garrett and K. Sri
also discussions with Roger Smith, Chien-Ping Lu,
Stan Eisenstat and Alain Martin; also the hospitali
Physics at Santa Barbara. This research was sup
AFOSR-88-0240 and by ONR grant N00014-92-J-4

# References

[BH89] Eric B. Baum and David Haussler. What s
*Neural Computation*, 1(1):151–160, Sprin

[BK93] Joachim Buhmann and Hans Kuhnel. Complexity optimized data clustering by competitive neural networks. *Neural Computation*, 5(1):75–88, January 1993.

[BT89] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation*, chapter 3, pages 210–217. Prentice Hall, 1989.

[DW87] R. Durbin and D. Willshaw. An analog approach to the travelling salesman problem using an elastic net method. *Nature*, 326:689–691, 1987.

[GL83] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1983.

[Gro88] Stephen Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1:17–61, 1988.

[GY91] D. Geiger and A. L. Yuille. A common framework for image segmentation. *International Journal of Computer Vision*, 6(3):227–243, August 1991.

[Hop84] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences USA*, vol. 81:3088–3092, May 1984.

[HT85] J. J. Hopfield and D. W. Tank. 'Neural' computation of decisions in optimization problems. *Biological Cybernetics*, vol. 52:141–152, 1985.

[KMY86] Christof Koch, Jose Marroquin, and Alan Yuille. Analog "neuronal" networks in early vision. *Proceedings of the National Acadamy of Sciences USA*, 83, June 1986.

[KY91] J. J. Kosowsky and A. L. Yuille. The invisible hand algorithm: Solving the assignment problem with statistical physics. Technical Report 91-1, Harvard Robotics Laboratory, 1991.

[MG90] Eric Mjolsness and Charles Garrett. Algebraic transformations of objective functions. *Neural Networks*, 3:651–669, 1990.

[Mjo85] Eric Mjolsness. *Neural Networks, Pattern Recognition, and Fingerprint Hallucination.* PhD thesis, California Institute of Technology, 1985. See section II.3.1 for wiring cost.

[Mjo87] Eric Mjolsness. Control of attention in neural networks. In *Proc. of First International Conference on Neural Networks*, volume vol. II, pages 567–574. IEEE, 1987.

[MM91] Eric Mjolsness and Willard L. Miranker. A Lagrangian approach to fixed points. In Richard P. Lippmann, John E. Moody, and David S. Touretzky, editors, *Neural Information Processing Systems 3.* Morgan Kaufmann, 1991.

[Ner70] Evar D. Nering. *Linear Algebra and Matrix Theory.* John Wiley & Sons, Inc, second edition, 1970.

[PB87] John C. Platt and Alan H. Barr. Constrained differential optimization. In Dana Z. Anderson, editor, *Neural Information Processing Systems.* American Institute of Physics, 1987.

[PS89] C. Peterson and B. Soderberg. A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, 1(3), 1989.

[RC91] A. Rangarajan and R. Chellappa. A continuation method for image estimation and segmentation. Technical Report CAR-TR-586, Center for Automation Research, University of Maryland, October 1991.

[RGF90] K. Rose, E. Gurewitz, and G. Fox. Statistical mechanics and phase transitions in clustering. *Phys. Rev. Lett*, 65(8), 1990.

[SgS90] G. W. Stewart and Ji guaang Sun. *Matrix Perturbation Theory.* Academic Press, 1990. Theorem on p.25-26, using definitions on p.3.

[Sim90] Petar D. Simic. Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimization. *Network: Computation in Neural Systems*, 1(1):89–103, January 1990.

[Tre91] Volker Tresp. A neural network approach for three-dimensional object recognition. In *Neural Information Processing Systems 3.* Morgan Kaufmann, 1991.

[Vap82] V. N. Vapnik. *Estimation of Dependences Based on Emprirical Data.* Springer Verlag, 1982.

[VJ89] B. D. Vujanovic and S. E. Jones. *Variational Methods in Nonconservative Phenomena.* Academic Press, 1989.

[YHP91] A. Yuille, K. Honda, and C. Peterson. Particle tracking by deformable templates. In *International Joint Conference on Neural Networks*, pages I–7 to I–12. IEEE, July 1991.