Defining the Neural Code

Thomas Bangert[*], **Ebroul Izquierdo**
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, England
`thomas.bangert@qmul.ac.uk`

Abstract

While the cellular physiology of neurons has been studied in great detail and is reasonably well defined, the underlying nature of their activity is far from understood and continues to be much debated [2] [12, 8]. In the same way that Boolean logic and theoretical Von Neumann machines [13] have provided the theoretical basis for computation and helped stimulate the development of practical computing, we suggest that it may be helpful to study the nature of neural function more abstractly; specifically, without taking into consideration the inherent complexity of the underlying cellular mechanics. It is commonly assumed by those studying the functionality of complex natural neural systems that their activity of is equivalent to computation [8]. It is asserted (by Church's thesis) that the underlying principles of computation are fundamentally universal and equivalent, and therefore that everything a neural system (no matter how complex) is able to do can be done by any other system capable of general computation [5]. The way computation is done by neural systems might well be very different from well-known models of computation (such as Von Neumann machines) but by Church's thesis they may be considered equivalent. If the core activity of complex neural systems may be considered computation and we accept that the neuron is the atomic unit through which this computation is expressed then an understanding of neural activity must be preceded by a formal understanding of the neuron itself and of the means by which the neuron communicates with other neurons. We suggest that to effectively study and develop a greater understanding of neural function it may be helpful to design minimal but fully functional artificial organisms in which the complex cellular physiology is abstracted and simplified but where the need to function in a complex environment is retained.

1 Introduction

Neural function is often described from the point of view of cellular physiology in elaborate detail but with very little insight into overall function [2, 11, 8]. On the other hand, in the field of *machine learning*, models of artificial simplified neurons are used which pay little attention to the constraints of empirical observation of the neural networks of natural organisms [4, 6]. The primary focus of artificial neural networks is to replicate high level functionality observed from studying the behaviour of complex natural organisms such as *learning* and *recognition* (of images and patterns). Despite great advances in the understanding of the cellular mechanics of neurons (including the complete mapping of the neural structure of some natural organisms [8]) and claims made for artificial neural networks (in respect of certain toy applications [6]) there is currently no working model of even the simplest of natural neural systems [8] and no consensus on the most fundamental aspect of neural function – the neural code [11]. The neural code is simply the nature of the information passed between neurons

---

[*] corresponding author; however, both authors are members of the MMV Research Group

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

that are physically linked to each other. To understand and define this information we suggest it might be helpful to consider a biologically plausible but abstracted hypothetical organism [2], particularly with respect to its evolutionary development from a simple fixed function system to a more general system capable of higher level function. Since mapping the neural connectivity of even the simplest of natural organisms has not led to an understanding of its function [8] we suggest that designing minimal but fully functional artificial organisms might be helpful in the development of a better more formal understanding of the neural code.

2 The Simplest Organism

All natural neural systems irrespective of their complexity must serve the underlying needs of the organism in which they exist. Since the neural systems of even the simplest organisms are extraordinarily complex (as a result of a billion years of evolution) it may be helpful to deduce the fundamental principles of these systems by designing a simplified artificial organism; an artificial organism whose behaviour is the expression of a set of well-defined rules [2]. The expression of these rules will be implemented using a small number of atomic units to form the inter-connected neural system; the function of these units is abstracted rather than being dependant on the complexities of cell biology. Nevertheless, they must remain biologically plausible. It may be assumed that abstracted functionality may readily be implemented by cell biology, in the same way that abstract architectural design is translated to physical structure. Conversely, it is often the case that determining the abstract design and functionality from a physical implementation is much more difficult (or indeed impossible). Further, it might be useful to observe how the components of the simplest organisms might develop over time to meet the challenges presented by the environment. All complex natural organisms have simple single celled ancestors and development of complexity is by necessity a step by step process where each step must have utility to the intermediary organisms. Neurons themselves must develop and differentiate step by step from non-specialised cells in the earliest of multi-cellular organism, and each step on this path to differentiation must have utility to the host organism.

The fundamental role of any practical neural system has its roots in motor control. The flagellum is the simplest of natural motor systems, which has its origins in single celled organisms that existed more than a billion years ago [10]. It consists of a simple rotary motor connected to a whip like filament structure whose primary role is to generate thrust (and thereby movement) when rotated in a liquid environment, allowing the organism to *swim*. The power of the flagellum, and associated speed of rotation, may readily be controlled by modulating the proton motive force that drives the motor. This is generally implemented by an interface that applies an electrical potential, with the strength of the potential modulating the application of motive force by the motor. Practical control of an organism may therefore be expressed as the application of a potential.

Motors are generally not useful without control and for a natural organism it is critically important to be able to direct its motors appropriately in respect of the conditions found in the environment. Organisms that are able to respond to the presence of food or signs of a predator have a great advantage over those that lack this ability. To determine the conditions of the environment requires the ability to measure (or sense) relevant properties of the external environment, such as atmospheric carbon dioxide or electromagnetic radiation. The simplest organism with the ability to measure the external environment and act in response to those measurements is a purely sensory-motor organism; in which a sensor is directly linked to a motor (see Figure 1a). The sensor produces a potential based on some measure of the environment and this potential actuates the motor directly. This arrangement may be implemented within a single celled organism or a very small multi-celled organism.

2.1 Communicating between Sensor and Motor

As an organism evolves from a single cell organism to a multi-cell organism, individual cells will begin to specialize; forming specialist motor cells and specialist sensor cells. The sensor cells found in multi-cell natural organisms express their measurement as a membrane potential, which varies in strength as a function of the strength of the stimulus which activates the sensor. Almost all eukaryotic cells (that is, cells of more complex organisms) actively maintain a non-zero membrane potential, and therefore the developmental path for specialist cells that use this membrane potential as a means for communication is straight forward. The establishment of specialist cells whose primary purpose is communication allows subsequent specialization that adds greater functionality.

When the sensor and motor cells (which perhaps in the past existed within a single cell organism) are separated by an insulating cell wall, an interface needs to exist which allows the potential produced by the sensor to be passed to the motor cell that it actuates. One such mechanism that allows adjacent cells to communicate by allowing a potential to pass is a simple gap junction. The design of a neural control system for the simplest of multi-cellular organisms may therefore consist of a sensor cell directly linked to a motor cell. The sensor cell will produce a continuous potential which in turn passes through the gap junction to directly drive the motor cell; the continuous potential from the sensor cell actuates the power the motor cell applies to the motor.

Size is often a critical factor in the success of an organism within a given evolutionary landscape. This has a fundamental impact on how a simple sensor-motor mechanism is organized. As an organism becomes larger there will be increasing pressure to separate motor from sensor. A motor is often best located at the rear of an organism whereas a sensor is sometimes most useful at the front of an organism. An interim solution would be for the sensor and motor cells to stretch and become elongated to preserve the gap junction (and indeed the cells of many natural organisms do just this). This allows an organism some flexibility in respect of size, but this solution is quite limited; there is a practical limit on how long individual sensor cells can become. Once this limit is reached, the organism can only become larger by introducing intermediary cells between the sensor and motor cells. It is possible to pass a potential from sensor to motor via an intermediary cell that relays potential, but it is impractical as the potentials produced by sensor cells are typically very small and easily dissipated. An increase in an organism's size therefore makes necessary a more general long distance sensor-motor communication mechanism. One simple solution is to take the energy of a weak but continuous potential and to concentrate it into very brief but much more powerful potentials; these spikes in potential (or action potentials) have the ability to travel much greater distances than a continuous potential. In natural organisms, action potentials developed alongside a new mechanism that allows communication between cells without a physical connection; these are special chemical junctions (synapses) which allow cells to communicate while remaining electrically insulated from each other. This prevents the action potential produced by the cell from being dissipated by the act of transmission.

A cell that has the ability to transmit a potential without dissipating it, makes possible multiple junctions that are able to accurately transmit the same potential. A direct electrical connection, which inherently dissipates the potential it transmits, cannot accommodate multiple junctions. While greater insulation and more powerful potentials allow greater distances to be bridged, action potentials inherently dissipate over distance. Larger distances can be bridged by adding insulation to the part of the cell that transmits the action potential. Insulation is found in more complex natural organisms in the form of specialist glial cells which flatten and wrap around elongated neural structures that propagate the action potential. Irrespective of the degree of insulation, however, all action potentials dissipate as a function of distance. Since synaptic distance is an unknown, all action potentials will dissipate by an amount not known to the means of transmission and therefore the power (strength) of an action potential cannot be used to carry information. Reception and transmission of action potentials therefore becomes inherently an all-or-none function. The synaptic transmission mechanism is all-or-none and does not transmit any information about the strength of the action potential sent across the synapse. The receiver cannot therefore know the strength of an action potential that triggered the synapse; it can know only that it has received an action potential. The cell's only function is to decide whether or not to produce an action potential in return, something which is also all-or-none. Action potentials may be considerably attenuated by distance (or other factors) but as long as the outgoing synaptic link is triggered, action potentials will fully regenerate as they are relayed, without any loss of information. These specialist cells (referred to as neurons) are divided into two general regions: one region which senses and integrates action potentials received from incoming synapses and a second region which produces and directs action potentials toward outgoing synapses.

The simplest neuron (for the hypothetical sensor-motor organism) is simply a relay neuron. When a relay neuron receives an action potential it will simply produce a single action potential with as little delay as possible. In classical artificial neural network terms this type of neuron may be described as having a threshold of one and synaptic weights of one. Action potentials are however not produced directly by a sensor cell and they cannot actuate a motor cell. The potentials produced by a sensor cell must first be coded into action potentials, which are then relayed to the motor cell, which decodes an action potential sequence back to a continuous potential that can actuate the motor. Action potentials do not by themselves represent a continuous (or graded) potential and therefore the potential produced by a sensor cell must first be encoded for transmission over the network of relay neurons and then decoded and translated back to a continuous potential for the motor cell. This encoding and decoding requires specialist neurons: sensor neurons and motor neurons.

2.2 Coding Sensor Information with Action Potentials

In the simplest case, an organism's neural architecture consists simply of a sensor cell communicating directly with a motor cell. This can be implemented by linking a sensor neuron directly to a motor neuron with a single synapse (see Figure 1d). As the hypothetical organism increases further in size the distance between sensor and motor can be bridged by any arbitrary number of intermediary relay neurons.

While the development of action potentials and synapses is complex, a step by step transition from neurons that produce a continuous potential to those that produce action potentials is straight forward. Once the transition is complete, the direct physical link that is required for communication between sensor and motor cells will be broken. An electrical synapse (implemented by a gap junction) transmits potential directly, but once the direct link between individual cells is broken, the link becomes something that transmits only information. If a physical potential produced by a sensor can be reproduced from the code, the transmission of the potential can be abstracted purely as the transmission of information. The initial sensor potential must be coded by the sensor neuron and at the destination the information must be decoded by the motor neuron, which produces the potential that drives the motor. This information that must be coded and decoded is well defined (real numbers between 0 and 1 for the hypothetical organism). What is not defined is the nature of the code itself; how the information is coded.

There are a variety of ways in which information may be encoded using action potentials. The simplest is a rate code where the strength of a continuous sensor potential is coded by the rate at which action potentials are sent. A fundamental observation of the sensor neurons found in natural organisms shows that the firing rate of sensor neurons is a function of stimulus strength [1]. However, a rate code for action potentials must count action potentials over times and is therefore an integer function and lacks the necessary degree of precision required to accurately code the continuous potential produced by the sensors. An action potential rate code is fundamentally a unary code. If a sensor neuron can fire an action potential every 4 ms then, for a time period of 1 second, only numbers from 0 to 250 can be coded for. Natural organisms are known to have a much lower latency than 1 second, and indeed a much higher data rate [11]. A more complex way to code information is by a temporal code. A temporal code might for example be a binary code where a spike indicates a 1 and lack of a spike within a set time period indicates a 0. A 40 ms time period could therefore transmit a 10-bit integer using 4 ms neurons. It is unlikely, however, that neurons are sufficiently reliable to support a time based binary code, particularly without a master clock that synchronizes the precise timing required [2]. Furthermore, as most neurons of natural organisms can fire at most every 4 ms, the data rate is also insufficiently high.

The advantage of designing a simple artificial organism is that it allows the Gordian knot of a century of conflicting observations and theories of increasing complexity about neurons and their function to be cut by the expedient necessity of designing practical neural circuitry that lives or dies based on its utility. The sensors of natural organisms are known to be very precise and a sensor potential is inherently an analogue value. This indicates a high data rate, which must be coded by a small number of spikes. The simplest way to code a continuous stream of real numbers would be to code sensor values as the distance between spikes; a high potential is coded by a small temporal distance between spikes and a low potential is coded by a large temporal distance between spikes. As long as neurons can accurately code by timing spikes and then preserve this timing when the action potentials are relayed then the sensor information can be preserved.

Information coded by a sequence of spikes must ultimately be accurately decoded back to the original potential that it represents. The fundamental property of a neuron (assuming they developed from sensors) is the ability to measure a physical property. The distance between spikes is a property that can readily be measured, and a sensor that originally developed to measure a property external to an organism might be repurposed to measure the interval between the spikes of a neuron to which it is connected.

The code for the hypothetical organism therefore has features of both a rate code and a temporal code. The distance between spikes codes the information and the data rate will be a function of the degree of precision with which neurons are able to time the spikes. One feature of this code is that large values may be transmitted much faster than small values. If a neuron is able to time its spikes with a precision of 30 bits (that is, reliably represent up to 2^{30} different sensor values) then a 4 ms neuron with a minimum firing rate of 100 ms (distance between spikes) can send a large 30 bit number in about 4 ms, but a small number would take up to 100 ms.

A temporal rate code for the hypothetical simple artificial organism is therefore a design choice, which might or might not reflect how neurons in natural organisms code information. For a simple organism where neurons only repeat action potentials between sensor and motor neurons such a code is sufficient for the precision, latency and data rate required. The sensor values that are most important (indicating food or danger) will have an inverse relation between size of the number and importance; large numbers will generally be more important and therefore can be acted upon more quickly. The limit on latency is just a single action potential (plus transmission delay and inter-neuron relay delay), which is consistent with the latency observed in a wide variety of natural organisms [11]. The code may therefore be considered useful for meeting the needs of very simple natural organisms. With sensor neurons able to accurately code potentials produced by sensors, with motor neurons able to produce precise potentials from an action potential code and with relay neurons able to bridge any distance between sensor and motor neurons, the hypothetical organism will have lifted any restriction on its ability to increase in size.

2.3 Building a Practical Organism

In respect of a practical implementation we will assume the neural system drives a surface dwelling (in effect 2-dimensional) organism that lives simply by photosynthesis. The primary directives of this organism are therefore to seek out the sun (and to bask in it once found) and to avoid predators and other dangers. The sensors of this organism will be two photoreceptors and it will be driven by two motor driven flagella (Figure 1). The fact that the organism lives by photosynthesis is important only in that this allows for the simplest of all practical organisms, on organism which is also straight forward to simulate and which can be built using conventional electronic devices as well as being similar to very simple organisms found in nature.

By default, photoreceptor sensors have a reverse bias and therefore sensor neurons that are driven by photoreceptor neurons will fire at a higher rate when light levels are low. Connecting the sensors of the photosynthetic organism directly to the motors will therefore by default lead to an organism that will swim toward the light, swim slowly or stop when it has found optimal illumination and which will turn to evade dangerous shadows (which obscure one sensor but not the other). For sake of completeness we may assume that the two sensors are linked by a circuit breaker when sensor values become too high. This will allow the organism to sleep when it is dark. This simple organism whose nervous system consists of two sensor neurons and two motor neurons (Figure 1d) can readily be shown to function effectively.

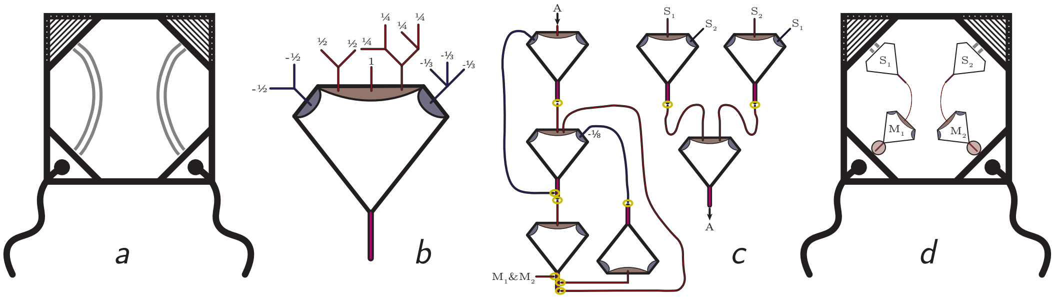

Figure 1: The neural structures of the simplified hypothetical organism: (a) without sensor or motor neurons (sensor and motor directly linked by gap junction); (b) abstract representation of the basic neuron, with activation, inhibition and dendrite structure showing integer division; (c) design of evader circuit (storage and decrement circuit and sensor subtraction circuit); (d) with sensor and motor neurons (directly linked but using action potentials to communicate).

3 Basic Arithmetic

Simple repeater neurons will have a single input synapse and a single output synapse. However, if a repeater neuron were to have two output synapses each of which connect to two separate neurons, then the neurons that link to each synapse would both repeat the action potentials produced by a single neuron – duplicating the action potential sequence being received. In a simple organism a duplicator neuron might arise from the need to drive two motors rather than a single motor from a single sensor. Conversely, a neuron with two input synapses will lead to two separate incoming action potential sequences being combined: it will integrate the action potentials of both neurons that link to it into a single action potential sequence. Integration is, however, more complex than duplication. If an action potential from one synapse overlaps with an action potential from the second synapse (including the resting period) then this will lead to the overlapping action potential either being delayed or not repeated. When two action potential sequences that are to be integrated are not identical and not appositionally synchronized then this will produce an imbalanced output sequence. Only when the two action potential sequences are identical but shifted equidistantly in phase will the integration of both sequences result in the output of a correct action potential sequence, with each action potential equidistant from the others. In this limited case, the result will be equivalent to addition. Although very limited, this form of addition might be of practical use in summing the output of two sensor neurons if the organism sought to double its sensors.

If the utility of dual sensors outweighs the drawbacks of incorrect addition, then evolutionary pressure might over time lead to a neuron that would be increasingly more capable of correct addition, with the side effect of forking the design of a simple repeater neuron to a multi-input multi-purpose neuron. Adjacent sensors measuring the same physical property will have an identical output potential most of the time and if they are connected by adjacent sensor neurons it may be that a first step in the development of a more complex neuron would be that adjacent neurons producing the same action potential sequence will develop the ability to synchronize their firing to be out of phase, allowing correct addition by a simple repeater neuron with dual synaptic input. Addition is of such fundamental importance, however, that any incremental improvement toward a more complete solution to the problem of being able to correctly add two separate action potential sequences would be of direct advantage to the organism. The transition from repetition to a more general implementation of addition is non-trivial and therefore a general solution might lead to considerable internal complexity. A neuron capable of general addition remains, however, backwards compatible with simple repetition (repetition is the simplest case of addition).

Addition is much more useful if it can be complemented with subtraction. Synaptic links which lead to repetition may be said to activate the receiving neuron. Subtraction is inherently the opposite of activation and may therefore be said to require some form of inhibition. Since it cannot be known from an action potential whether it is meant to activate or inhibit, dedicated synapses would be needed for inhibition. Dedicated inhibitory synapses would therefore be a necessary precursor to make subtraction possible. Developmentally, an inhibitory synapse may have had its origins in a defective synapse which dissipates an accumulating potential rather than adding to it. Just like addition might have had its origins with the much simpler function of inserting action potentials into an action potential sequence, so too subtraction might have had its orgin with the ability to remove individual action potentials from an action potential sequence. This simple precursor to subtraction would require an action potential to arrive at the precise time an action potential is about to be produced. This initially very limited ability to reduce the number of spikes in an action potential sequence might in time have led to the development of a more generalized arithmetical subtraction.

Once a neuron is capable of addition and subtraction, multiplication and division can easily be implemented by repeated addition or subtraction using multiple synapses; action potential sequences can be duplicated and then added or subtracted. However, general multiplication or division require the use of increment or decrement operators and a comparison operator (end at zero). It may well be that incremented or decremented repetition with the use of a comparison is too difficult for a single neuron to implement. The difficulty a neuron whose basic operation is repeating sensor measurement faces is that is that sensor measurements are inherently real numbers of arbitrary precision, whereas increments or decrements are integer operations.

When a function is too complex to implement directly in hardware it is often useful to implement important components that are useful toward a general solution of the problem. Specifically there are often component functions which can be solved much more quickly when implemented directly in hardware and there are other component functions that have a large hardware footprint if implemented generically but which can be implemented much more compactly with custom hardware. When attempting to implement multiplication and division from the point of a neuron it might be useful to first implement simplified integer multiplication and division functions ($\times 2$, $\times\frac{1}{2}$, $\times 3$, $\times\frac{1}{3}$, $\times 4$, $\times\frac{1}{4}$,$\times 5$, $\times\frac{1}{5}$ ...). This simplified form of multiplication and division is attractive because it it can be implemented very simply by inserting action potentials or removing actions potentials. For example, to double any action potential sequence ($\times 2$) an action potential can simply be inserted at the midpoints between the action potentials of an action potential sequence. Division can be implemented just as easily by simply removing action potentials from a sequence. For example, to divide by two ($\times\frac{1}{2}$) remove every second action potentia, to divide by three ($\times\frac{1}{3}$) remove two out of every three action potentials.

While both simplified multiplication and division are easy to implement in hardware, more abstractly though we see multiplication as repeated addition and division as repeated subtraction, for which it is necessary to know the subtractive or additive element. With multiplication the additive element is simply the number itself. To multiply any number A by one of the constants ($\times 2$,$\times 3$, ... ,$\times$n) it is necessary only to know the number A which is then added to itself the required number of times (A+A, A+A+A, ... A+A+...+A). For division, however, the subtractive element is the multiplicative inverse of the number and therefore not known. Just as with using British imperial units to measure size or distance, the key is understanding fractions. With the 'inch' unit in particular, integer multiples of *inches* are straight forward but most commonly one would deal with smaller sizes which were measured in fractions of an *inch*, and this required special skills to cope with efficiently. A bolt might for example be $\frac{3}{4}$*inch* or a socket might be $\frac{3}{16}$'ths of an *inch*. Just as a carpenter or automobile repairman must be fluent in these arithmetic fractions, so too when designing the neural systems of simple hypothetical organisms these common fractions must be expressed directly in hardware to allow designs where the overall number of neurons must be kept to a minimum.

Because integer fractions are so fundamental to the core activity of neurons we suggest that it be implemented directly by individual neurons themselves. We suggest that the mechanism for this is the complex branching (dendrite) structure of the input region of neurons. A repeater neuron therefore requires no dendrite branching at all since it simply repeats any action potential it receives. On the other hand, a neuron that operates on fractions of $\frac{1}{16}$'ths would need a more complex dendrite tree that divides four times, and a neuron that operates on fractions of $\frac{1}{32}$'nds would need a tree with a depth of five. The complexity of the dendrite tree would therefore be a function of the degree of precision that is needed. The drawback of such an arrangement is that when approaching the degree of precision needed by the natural world (tree depths of 10, 16 and 32) the dendrite tree structure needed becomes impractically vast. A major benefit of such a structure would be all the fractions of the specific depth can be readily calculated simply by connecting to the correct multiple of of dendrites (for example a spike sequence can be decreased by the fraction of $\frac{5}{32}$ simply by connecting to the synapses of 5 out of 32 branches of the dendrite tree). Such *deep* fractions also instruct us as to how increment or decrement operations may be defined, which are instrumental to being able to support general computation. The dendrite structure of a neuron may therefore also be seen as a way to impose an integer structure upon the real number sensor values which the action potentials represent.

4 The Rules of Complexity

While simple relay neurons are sufficient to design an organism that will exhibit very simple rule based behaviour (swim towards the light and evade danger), a neuron capable of arithmetically integrating multiple inputs allows the design of an organism with much more complex behaviour. Neurons capable of integrating multiple inputs and general arithmetic allow for the implementation in neural circuitry of arbitrary general directives. We shall set four directives:

1. swim toward light at a speed inversely proportional to its brightness
2. at optimal light levels circle slowly
3. at minimal light levels spiral slowly outwards
4. swim away from danger

The first directive is identical to the previous organism with the sensors directly linked to the motors. The core design of the organism is therefore the same as the previous organism (Figure 1d), and may therefore be considered backward compatible, with the new directives being in effect *plug-ins*; neural circuitry that is added in to supplement or modify the basic functionality. Being able to plug in additional neural components is made possible by neurons capable of integrating multiple inputs. The ability to plug in additional new components with more advanced ability also shows how over time greater complexity may develop within a simple organism.

The second directive allows the organism to continue moving in conditions of maximum luminance (with reverse biased sensors at minimum and sensor neurons producing no action potentials). The ability to move at a minimal speed, as well as allowing a difference in levels of propulsion per side which leads the organism to circle slowly, maximizes the organism's ability to detect danger at the earliest opportunity while at the same time maintaining a safe degree of movement. To allow the organism to move while all sensors are at zero requires action potentials to be produced that do not immediately originate from sensor input. This can either be from a specialist clock-like neuron which always fires at a constant rate or it can be from a subsystem that stores a previous sensor value. The ability to internally store action potential sequences or to self-generate action potentials is important in the transition from a system that is simply *feed forward* to one capable of more general function. The simplest implementation is to use a modified sensor that simply produces a constant value irrespective of any measurement (that is, the ability to measure is disabled). This constant is divided to obtain the correct motor values (to make the organism slowly circle) and then added to the motor neurons. This input is inhibited when there is any sensor input (using a multiplier so that even small values inhibit the circuit). With the clock neuron set to produce an action potential sequence at 50% (the maximum and minimum firing rate) the organism can be set to slowly circle by reducing the clock to 2% for one motor and 1% for the second motor output. This can be calculated by dividing the clock by 50 and 25 for each of the motors respectively.

The third directive is nominally implementation by the use of a crude on-off non-neuronal circuit breaker, which conserves energy in conditions of darkness but leaves the organism in a vulnerable immobile state. Inhibition, which allows for subtraction as well as addition, lets the simple circuit breaker to be replaced by a neural sub-system that will produce increasingly large negative values when the sensor potential is too high. These negative values will inhibit the motor neurons. Inhibition must not, however, reduce the motor neuron's output to zero. When in complete darkness (with sensors at maximum membrane potential) the organism must continue to move at a slow steady continuous pace. If a motor neuron on one side of the organism is inhibited a fraction more than the motor neuron on the other side, then the organism will circle. Slowly circling may be implemented by inhibiting high input values with a small difference between the two motors. When using a known sensor input, the ability to drive the motors to circle accurately derives from the ability to divide the sensor input value accurately. To circle at 5% and 2.5% of maximum requires division by 20 and 40, respectively. The third directive, however, requires the organism not only to circle but to spiral outwards, which is a modified circling motion which may be implemented by incrementing the lesser value over time until it approaches the larger value. This requires the output to the low power motor to be increased slowly over time until it approaches the output of the high power motor. To implement this, a timer circuit is needed, which increments a stored value over time. Timing (equivalent to counting with a constant time interval) and incrementing are important general abilities.

Timing requires a system with the ability to count and counting requires the ability to store a value, which is then incremented at each time interval. A stored value may be implemented by a circular circuit of relay neurons, where the output of a neuron is sent back to its own input – indirectly by a second neuron. Action potential sequences are self-perpetuating on this circular circuit as they travel over a set of relay neurons linked to form a circle. The circle must be primed until the initial action potential completes the circuit, at which point the action potential is self-sustaining and the priming must be inhibited. This circular arrangement allows neural circuits to store information. It is the building block of a memory store, with each neuron circle being able to store one number of arbitrary precision. Being able to store information is of fundamental importance for neural circuits being able to implement complex systems. The input is added to (incremented) by dividing the same input by an appropriate high integer and adding it. The number of secondary neurons of the circle determine the increment delay. If integer division for a single neuron is limited (to 120 for example) then two neurons in sequence may be used (to achieve 14,400). If the delay for an action potential to circumnavigate the circle is 1 second, then a 1/14,400 increment would produce a 4-hour timer. A one second delay can be implemented either with a very large circle or with a specialist neuron which delays action potentials. Relay neurons are under selective pressure to repeat action potentials with minimal delay. Older variants of neurons might therefore have greater delay, and it is these older slower neurons that might have found utility in applications where delay is useful.

The fourth directive is partially implemented by the first directive. The organism will turn away from any shadow that blocks the light from one of the sensors, but it will fail to follow this up by escaping from the danger. A shadow indicates danger and the organism should turn away and then swim away at maximum speed. The simple sensor-motor circuit is therefore insufficient as it will not remember the danger once the shadow is out of sight and it will continue at its normal pace once it has turned away from the shadow. The organism cannot know when it has escaped from the danger so the best it can do is to swim at maximum speed for a fixed period of time. A shadow should therefore trigger a timer and while the timer is active the motors should be actuated to maximum power. This can be implemented with a memory circuit, in the same way as directive 3 but with the timer count being decremented rather than incremented. Figure 1c shows a basic design for the memory circuit and the sensor subtraction circuit.

5 Defining the Neuron as The Atomic Unit of Calculation

For neurons to collectively implement functions of higher level ability (as exhibited by natural organisms), the atomic units of the circuitry implementing that functionality should be simple. The first simplifying assumption is that the underlying neural code is numerical. The second simplifying assumption is that the core functionality of neurons is to carry out simple numerical division and addition. Equation 1 shows the output function given k inputs. The integer ratio of $\frac{a_k}{b_k}$ is equivalent to a weight but both a_k and $b_k$must be integers within the division range of the dendrite tree. The integer b_k represents the number of divisions of the dendrite tree and a_k is the number of those branches the input neuron connects to. The output is simply the sum of the input, without any transfer function.

$$o = \sum_{k=1}^{n} \frac{a_k}{b_k} I_k \tag{1}$$

6 Analog and Digital: equivalency when faced with limits on precision

Sensor neurons typically do not signal their state by firing action potentials but rather present a graded potential to inter-neurons that are linked directly by a gap-junction. The underlying complexity of this sensor function may be abstracted simply as a sensor measurement, which may be coded as a real number of arbitrary precision. In the case of photoreceptors there are hard physical limits on precision. Photoreceptors measure light by transducing the energy from light into an electrical difference (by means of pumping ions). Light is absorbed by proteins that have evolved to absorb only within a specific range of wavelengths. When light is absorbed the energy (or *force*) carried by the electro-magnetic wave-particle can be measured. The precision of this measurment is limited by the particle nature of electromagnetic radiation; it can only be absorbed in discrete packets of energy (called *photons*). Measuring light can therefore be made no more precise than counting discrete photons.

In direct sunlight a human foveal cone photoreceptor will receive approximately 2.5×10^9 photons (with a simplifying assumptions that all photons have a wavelength of 500nm, that illumination of the natural environment for humans is 1000 watts per m^2, that the surface area of the human fovea is 3mm^2 and foveal sensor density is 100,000 sensors per mm^2). Assuming that a neuron can fire at most every 4ms (and that all available photons are accurately transduced), counting photons within that period of time requires integer numbers that can be coded by 27 bits. If glial cells focus and direct specific wavelengths and thereby increase the effective sensor surface area by a factor of at most 10 then this will increase the requirements to 30 bit integers. This value corresponds well with what we know about colour vision where older standards for colour reproduction require three 8 bit integers (for a total of 24 bits) whereas newer standards require 10 bit colour (with each pixel being coded by 30 bits).

7 Neurons and Logic Gates

The neural network of our hypothetical organism does not *learn*, *habituate*, *recognize* or change its function in any way. Indeed, for our simple hypothetical organism most change to the core of the neural system is dangerous and will in almost all cases cause it to function incorrectly (that is, with poor fitness). As an example, if the relay neurons of a memory circuit to *learn* to fire together rather than to fire in sequence then the information stored by the memory sub-circuit would be lost. If the circuit that carries out the timed increments for the ability to swim in a spiral were to lose its very small increment this would lead to a failure of the spiral. Neural circuits are inherently very flexible in that functionality may be incrementally added but the core function must not change. Indeed, a practical circuit would incorporate error correction to resist *change* (in most cases, damage) to core functionality. The neural circuitry is also not self-organizing in any way. It is purely the implementation of an abstract design, which is the expression of some *builder* mechanism. Any mechanism with a general ability to build complex systems would require the design information in the form of a code from which it will carry out its work. Because the neural system cannot function before being built, the code that expresses the design of neural circuitry must be external to the neural system itself. Natural organisms have non-neural means to code design information, but this will be left to future work to explore.

Almost all conventional abstract models of neurons assume the central dogma that the functionality of neurons is linked to the action potential; that action potentials are counted and this count may be modified or *weighted* in some way depending on where the action potentials orginate from. Neurons are also assumed to have a central threshold, which may also be seen as a type of weight. There are many variations of this, such as the leaky bucket threshold count. The feature that unites these models is the hypothesis that synapses themselves or the dendrites are weighted in some way to moderate the value of an incoming action potential, and that the degree of moderation can be modified and therefore is a type of stored value. It is assumed that these stored values are used by neural systems to store information. The neurons of natural organisms often have a very complex and intricate dendrite (tree like) structure which hold large numbers of synapses. The structural complexity of these synapses is often assumed to indicate some higher ability. The many synapses may for example be used to connect to an array of sensors and by modifying the weights of the synapses the neuron may used as a kind of detector. Taken to the most extreme, it is sometimes suggested for higher organisms which have complex visual systems that they have individual neurons that act as *grandmother* detectors [7]. When such neural systems are simulated, the function of these neurons may be expressed as a search space comprised of the synpatic weights. What is implicit in such simulations is that these stored values are addressed. Generally, the threshold and weights are set by training the neuron, which amounts to an exhaustive search through all possible weights and then a central threshold which responds best when presented with the stimulus.

Very little consideration is given to the problem of addressing synaptic weights since addressing appears trivial when neurons are simulated using a Von Neumann machine. This assumption of addressing ignores the complexity of computation and the history of the development of practical computation. With a Von Neumann machine each word of memory can be addressed (that is, it can be read and modified if the address is known). Turing machines provide an alternative model of computation where the memory store is not addressed. Turing machines can be designed to organize their memory store so that is can be addressed but this is not a trivial step. A Turing machine has a store of instructions which are able to systematically modify the memory (the stored information), but only the current memory location can be directly accessed. All other memory locations can be accessed only by first locating them by means of a linear search. Neurons on the other hand are much simpler devices, which in the case of a relay neuron simply fires an action potential on receipt of an action potential. More complex neurons integrate incoming action potentials in a more complex way, but their sole ability is to fire an action potential. There can be no specially coded action potential sequence that modifies a threshold or changes the synaptic weights. It is sometimes suggested that synaptic weights can be changed by general principles such as 'neurons that fire together wire together' [9]. This is a fundamental error. Systems capable of general computation must have a facility to store information and be able to systematically read and write that information. If neurons are unable to modify their weights and thresholds systematically (by use of their action potentials) then the hypothesis that weights and thresholds are in some way used as a store of information must be wrong. So too is the view that neurons detect or perceive higher level aspects of the environment. This paradigm is often expressed implicitly in terms of the neural code preserving an external signal, with the expectation that given a neural code some external signal may be reconstructed. Individual neurons do not detect or propagate signals from the environment and they do not detect high level constructs (such as *bugs* or *grandmothers*). This is a category error, conflating the high level abilities of complex neural systems with the abilities of individual neurons.

The model of the neuron presented here does not use thresholds and synaptic weights; it does not store values and is not self-modifying in any way. It is an unchanging unit capable of only very basic arithmetic; a calculation which is performed on coded information as it is relayed. A neuron which performs addition and integer multiplication is superficially very similar to a threshold synaptic weight neuron. This makes it consistent with empirical observations of neural activity. The primary difference is the role of the complex dendrite structure found in the neurons of many natural organisms. It is often assumed that dendrite complexity belies some important and complex function. We suggest the contrary; that function of the complexities of the dendrite structure is very simple – that it is a complex and ornate solution to the very simple problem of integer division. Neurons branch not to store increasing amounts of information but to achieve greater precision. It would be predicted therefore that higher organisms, which have specialist cells with a very high degree of branching (such as the Purkinje cells, which have upwards of 100,000 branches), need those cells to make precision calculations and if those cells are degraded this would lead to a degradation of abilities for which precision calculations are needed.

Using the needs of a simple hypothetical organism to define the neuron and to design simple but practical neural systems has therefore led us to a simple but effective neural code and a simple neural model in which the primary function of a neuron is basic arithmetic. We have shown that these simple atomic units of calculation can be used to design more complex neural circuits such as memory stores and increment/decrement. These are the fundamental elements necessary for universal computation. Universal computation also requires the ability to code a set of instructions, to read, write and modify stored values and to execute a set of coded instructions in sequence. It therefore requires the ability to abstract the simple fundamental operations. The neural systems of an organism capable of general computation do not directly express the function of the organism hard-coded in the neural circuitry but by a more abstract code which use the generic elements of the neural circuitry systematically to achieve the same thing.

The design of the practical organism using simple neurons shows a noteworthy similarity with the design of digital circuitry. The fundamental unit of digital circuits is the logic gate, which are the simplest units capable of arithmetic (binary/boolean arithmetic) and which are usually implemented using transistors (and other electronic components such as resistors and capacitors). The AND gate carries out binary addition, but on its own it is insufficient to support general computation. However, when joined with the NOT gate this leads to a single device capable of universal function (the NAND gate). The advantage of this is that any digital device no matter how complex can be designed using just this single atomic unit.

Logic gates are not inherently capable of storing information, but when the output of two NAND gates are connected to their inputs they are able to maintain their state and this property leads to the ability to store one bit of information. Information coded by digital circuits is therefore binary, and all devices capable of calculation and computation designed using digital circuits classed as digital; in which number are coded using binary digits. The hypothetical organism presented here shows a potential alternative to digital computation. The values coded using a binary code are inherently integers; indeed a single binary digit can only code for the integers 0 and 1. A neural spike code (which codes by distance between spikes) is inherently an analog code which represents sensor measurements. This suggests, therefore, that natural organisms use an analog model of computation rather than the digital.

Digital computation using single binary bits is generally not very useful, and therefore the digital circuitry used to support general computation is commonly organized into words of 32 or 64 bits. The underlying implementation is less important (indeed NOR gates can as readily be used as NAND gates) than the abstract design of using the underlying hardware to code numbers and being able to perform basic mathematical operations on the numbers. Equally, if the underlying function of neurons is also in the same way simply to code and relay numbers with the ability to perform addition and integer division then the underlying physical complexities of neural mechanics are removed. The most important element which would be removed is the central role of the action potential. A simulation could remove the discontinuity of the action potential and replace it with simple numerical continuity. A practical system that uses analog numbers is limited in precision to the underlying mechanics of the system and therefore these numbers can be represented using a digital representation (provided the number of digits are sufficient). Since a digital circuit with the ability to carry out division and addition is trivial neural circuitry and digital circuitry may be considered equivalent.

8 Discussion

Studying the needs of a simple hypothetical organism suggests that the most important functions a neural network needs are repetition, duplication (addition) and being able to divide by an integer fraction. A general neuron integrating all three functions into a single unit greatly simplifies neural network design and allows for complex behaviour to be implemented with very small numbers of neurons. The proposed neuron does not by itself store information and neither does it have any higher level ability (such as *learning*) . It is an unchanging unit capable of only the most basic arithmetic. Its function is superficially very similar to a conventional artificial threshold synaptic weight model and this makes it consistent with empirical observations of actual neural activity. The difference is primarily in formally defining the neural code. The most visible difference is the role the complex dendrite structure plays. It is often assumed that dendrite complexity must belie some important and complex function [3, 8, 14]. We suggest the contrary; that the underlying function of what appears to be very complex dendrite structure is very simple – that it is a complex and ornate solution to the very simple problem of integer division. Complex branching is not the product of *learning*, *recognition* or storing information but simply a by-product of the need to achieve greater arithmetic precision. An empirical prediction from this would be that natural organisms, some of which have specialist cells with a very high degree of branching (such as the Purkinje cells with upwards of 100,000 branches), use neurons with a complex dendrite structure simply to calculate more precisely.

9 Conclusion

We have used a simplified abstract model of a neuron, which does not rely on any stored information such as thresholds or synaptic weights, to design the neural circuitry of simple organisms in order to illustrate utility. Organisms that exhibit useful and complex behaviour may be designed using very small numbers of the proposed general purpose neuron. The proposed neurons are capable only of simple arithmetic calculation but they may readily be combined to achieve more complex functionality. We have demonstrated a *network* of neurons that implement a memory store and the increment/decrement functions. Noted is the similarity of the memory store implementation when compared to a Von Neumann machine implementation using logic gates. We will leave it to further work to demonstrate conditional branching and to implement the ability to store programs. This is expected to show that Von Neumann machines and neural networks have a very similar approach to practical computation: that the core function of both is simple processing of numbers.

References

[1] Edward D. Adrian. The impulses produced by sensory nerve endings. *The Journal of Physiology*, 61(1):49–72, 1926.

[2] Valentino Braitenberg. *Vehicles, Experiments in Synthetic Psychology*. The MIT Press, 1984.

[3] Tiago Branco and Michael Häusser. The single dendritic branch as a fundamental functional unit in the nervous system. *Current Opinion in Neurobiology*, 20(4):494–502, 2010.

[4] Francis Crick. The recent excitement about neural networks. *Nature*, 337(6203):129–132, 1989.

[5] Martin D. Davis and Elaine J. Weyuker. *Computability, Complexity, and Languages*. Academic Press, 1983.

[6] Alexander K. Dewdney. *Yes, We Have No Neutrons: An Eye-Opening Tour through the Twists and Turns of Bad Science*. Wiley, 1997.

[7] Charles G. Gross. Genealogy of the “grandmother cell”. *The Neuroscientist*, 8(5):512–518, 2002.

[8] Christof Koch and Gilles Laurent. Complexity and the nervous system. *Science*, 284(5411):96–98, 1999.

[9] S Lowel and W Singer. Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. *Science*, 255(5041):209–212, 1992.

[10] Jonathan Moran, Paul G. McKean, and Michael L. Ginger. Eukaryotic flagella: Variations in form, function, and composition during evolution. *BioScience*, 64(12):1103–1114, 2014.

[11] Fred Rieke, David Warland, Rob de Ruyter van Steveninck, and William Bialek. *Spikes: exploring the neural code*. The MIT Press, 1997.

[12] Richard B Stein, E Roderich Gossen, and Kelvin E Jones. Neuronal variability: noise or part of the signal? *Nature Reviews Neuroscience*, 6(5):389–397, 2005.

[13] John von Neumann. First draft of a report on the edvac. *IEEE Annals of the History of Computing*, 15(4):27–75, 1993.

[14] Rafael Yuste and Winfried Denk. Dendritic spines as basic functional units of neuronal integration. *Nature*, 375(6533):682–684, 1995.