CAMBRIDGE
UNIVERSITY PRESS

ORIGINAL ARTICLE

Logic and learning in network cascades

Galen J. Wilkerson* and Sotiris Moschoyiannis

University of Surrey, Guildford, UK GU2 7XH, UK (e-mail: s.moschoyiannis@surrey.ac.uk)
*Corresponding author. Email: g.wilkerson@surrey.ac.uk

Action Editor: Hocine Cherifi

Abstract

Critical cascades are found in many self-organizing systems. Here, we examine critical cascades as a design paradigm for logic and learning under the linear threshold model (LTM), and simple biologically inspired variants of it as sources of computational power, learning efficiency, and robustness. First, we show that the LTM can compute logic, and with a small modification, universal Boolean logic, examining its stability and cascade frequency. We then frame it formally as a binary classifier and remark on implications for accuracy. Second, we examine the LTM as a statistical learning model, studying benefits of spatial constraints and criticality to efficiency. We also discuss implications for robustness in information encoding. Our experiments show that spatial constraints can greatly increase efficiency. Theoretical investigation and initial experimental results also indicate that criticality can result in a sudden increase in accuracy.

Keywords: percolation; critical cascades; information cascade; information diffusion; influence maximization; network rewiring; neuronal avalanches; statistical learning; linear threshold model; universal Boolean logic

1. Introduction

Critical network cascades are ubiquitous, found in brain function (neuronal avalanches), social and biological systems (information cascades), epidemics (SIR model), and many other self-organizing systems where there is interaction between entities (Bak *et al.*, 1987; Easley *et al.*, 2010; Newman, 2018). A very simple Boolean model that undergoes critical cascades (also known as avalanches) as a function of network topology is available (Watts, 2002) as a networked version of the linear threshold model (LTM) (Sakoda, 1971; Schelling, 1971; Granovetter, 1978), which we use as a basis for simple, biologically motivated learning. This is a departure from usual studies of the LTM, since we treat it not only as a model of critical cascades in complex systems, but as a model for classification and statistical learning.

Here our focus is on the benefits of certain elementary and biologically motivated modifications to this simple LTM. Of particular interest are logic computation, spatial (geometric) constraints, criticality, lack of type I error, and information encoding. This is a preliminary investigation of these simple modifications, how they scale over large networks, and their effect on the network's ability to solve problems. Logic is useful to understand if comparing the action of natural networks to various types of artificial learning and computation. Presence or absence of types of error may contribute to robustness of classification or computation actions of the network (Rubin *et al.*, 2017). Criticality occurs as percolation (see below) in many naturally occurring networks and may be a key to efficient learning. A geometric constraint (distance) is motivated by energetic or metabolic costs of edges in many real-world networks (Lynn & Bassett, 2019). Information coding may also be important as an efficient way for the network to take in information or express its action (Dayan & Abbott, 2001).

In the fields of complex systems and closely related network science, information cascades have been studied as a form of *information diffusion* in recent years (Jalili & Perc, 2017; Easley *et al.*, 2010). Generally speaking, this phenomenon describes how information is passed between entities over a network and may describe how the information is processed by the network, subject to parameters such as topology. Of particular interest is identification of important nodes in these networks by various network *centrality* measures.

While it is impossible to model all of the brain's or a social network's dynamical operations with exactness, here we investigate fundamentals of computational power, efficiency, and robustness in one of the simplest *learning models* of cascades sufficient for computation and classification (Vapnik, 1999). We therefore attempt to maintain a policy of simplicity throughout this work, using basic models or modifications to them, since complex behavior can emerge as these models are scaled in size. We study whether actions or small modifications of the LTM are sufficient to perform complex information processing and learning.

Here we must mention the McCulloch–Pitts (MP) neuron (McCulloch & Pitts, 1943), as the origin of neural networks and other areas of machine learning (Murphy, 2012, pp. 568-569), but also bearing some similarity to action of a single node in the LTM (described below), since both are Boolean threshold units. Like the LTM, the MP neuron has Boolean inputs and outputs and a threshold value, but unlike the LTM, it contains a vector of real weights, as well as inhibitory inputs which can block the firing of the neuron. It then computes a dot product of the inputs with the weights, firing if this dot product attains the threshold and no inhibitory inputs are present. Subsequent to the MP neuron, seen especially in backpropagation and gradient descent, a major focus in machine learning has been on adjusting the weights of these networked neurons to achieve learning or passing the neuron output through an activation function of various types to improve learning (Murphy, 2012). The mechanism we investigate here with the LTM is to continue to use simple thresholding at the neuron level, but rather than making the neuron more complex, to look at ways *learning emerges from simple Boolean neuron connectivity*. Then the activity of the network becomes synthesis of information and decisions from micro-scale to macro-scale (Lynn & Bassett, 2019). This is exactly the action of critical state change observed in many naturally occurring networks undergoing cascades (Beggs & Plenz, 2003; Easley *et al.*, 2010; Newman, 2018).

We motivate our biological perspective and simplicity by information cascades and information flow in social networks (Kempe *et al.*, 2003), similar neuronal avalanches in brain networks (Hesse & Gross, 2014), along with the other related fields mentioned above (Easley *et al.*, 2010). Arguably, cascades and avalanches are a complex activity that results from very simple interaction, hence our interest here. Of course, much more complex and dynamic activity can also be found in biological networks (Bohte *et al.*, 2002), not our focus in this article.

Critical cascades are a kind of *percolation*, a model found in theoretical physics and graph theory (Stauffer & Aharony, 2018). A key feature of cascade percolation is a sudden *phase transition* of the cascade size or frequency (macro-state) *order parameter* as the individual edge or node (micro-state) *control parameter* reaches a *critical* value. In biological systems, it has been shown that these critical avalanches are fundamental and can lead to optimal information processing (the "criticality hypothesis") (Hesse & Gross, 2014; Shew & Plenz, 2013). It has been shown that the LTM on Erdos–Renyi graphs undergoes a phase transition in cascade frequency as a function of the control parameter connectivity, expressed as the mean degree z, or graph edge probability p (Watts, 2002).

The problem of learning in networks undergoing cascades lies at the intersection of social and computer science (information diffusion and influence maximization) (Domingos & Richardson, 2001; Kempe *et al.*, 2003; Khalil *et al.*, 2014), neuroscience (neuronal avalanches, neural plasticity) (Lynn & Bassett, 2019; Shew & Plenz, 2013), percolation theory (statistical mechanics and graph theory) (Stauffer & Aharony, 2018), and learning theory (cognitive science and statistical learning) (Coolen, 1998; Hastie *et al.*, 2009; Rojas, 2013; Vapnik, 1999).

Brain-motivated origins and recent rapid advances notwithstanding neither modern computing nor machine learning adopt critical cascades as a design paradigm, although they suffer significant deficits in efficiency when compared to naturally occurring information processing systems (Coolen, 1998; Minsky & Papert, 1969; Rojas, 2013) and seem to be reaching physical limits (Conte *et al.*, 2019).

This last statement means that the area of critical cascades as a computational or statistical learning model remains largely unexplored. As stated, our present work originates from the simple model of critical percolation in the form of global cascades (Newman, 2018; Watts, 2002). There has been work on identifying influencers and minimizing epidemics in the SIR model, another kind of cascade (Chen *et al.*, 2012; Newman, 2002). Also there is work on various attacks to disrupt or disconnect networks, which relates to percolation because of connectivity (Callaway *et al.*, 2000). The nascent field of guided self-organization also has some connections, since a cascade can emerge from node-level connection choices in adaptive rewiring (Jarman *et al.*, 2017; Prokopenko, 2009). In fact, we view our present work as fitting into this context, since we discuss criticality here as well as constraints which could result in efficient individual edge choices and emergent learning. Also similar to the present work is that on *influence maximization* (Domingos & Richardson, 2001; Kempe *et al.*, 2003; Khalil *et al.*, 2014). However, again there the focus is not identical to ours, since that work is framed as an maximization problem, and usually focuses on approximation algorithms and sub- or super-modular activation in a social influence context rather than criticality or percolation and learning.

In machine and deep learning, there has been some work on phase transitions in the Hopfield model, Markov random fields, and restricted Boltzmann machines. These have theoretical links to the Ising model, another basic physical model that can undergo phase transitions (Barra *et al.*, 2017; Bruce *et al.*, 1987; Häggström, 2000; Wang *et al.*, 2013), so this has some closer connections to our work. Here however, we examine an arguably simpler model and some biologically motivated variants. Deep learning generally has some connections here as a form of network-based statistical learning, but as we have stated these methods do not focus on criticality or cascades, but rather on edge weight optimization (Rojas, 2013). In the field of brain research, there has been work on the physics of neuronal networks and their ability to learn and process information, as well as the benefits of criticality, but again this is largely not framed as statistical learning (Bassett *et al.*, 2011; Beggs & Plenz, 2003; Hesse & Gross, 2014; Lynn & Bassett, 2019; Shew & Plenz, 2013). There has been development at the intersection of brain research and machine learning in areas of spike timing dynamics in networks, particularly in applying techniques such as backpropagation and gradient descent (Bohte *et al.*, 2002). While these are very exciting, they are also complex and dynamical models, investigating temporal patterns over networks. It has also come into question whether backpropagation is biologically plausible (Bengio *et al.*, 2015) and seems that self-organization may be a more tenable mechanism (Hesse & Gross, 2014). Our interest is again in the *simplest possible biologically motivated model of cascades* and what it can achieve. Finally, there has been work on percolation in spatial networks, but also not as learning (Barthélemy, 2011; Gao *et al.*, 2015; Gray *et al.*, 2018; Penrose, 2003).

In the first section of the present work, we introduce the LTM and show how it can compute monotone logical operations. We then develop an antagonistic linear threshold model (ALTM), by taking the complement of the original labeling rule, and show that it can compute (functionally complete) universal logic. We also observe that the ALTM introduces order dependence and compare its cascade frequency to that of the LTM. Finally, we introduce a formalism of the LTM as a binary statistical classifier and discuss classification accuracy and error.

In the second section, we develop a formalism for the original LTM as a statistical learning model. We then experimentally investigate the biologically motivated effect of spatial distance and criticality on learning in the LTM (Wilkerson, 2021). Finally, we examine how the LTM robustly encodes information.

2. Methods

2.1 Linear Threshold Model (LTM)

The LTM, which can undergo critical cascades, is defined as follows (Watts, 2002):

---

1 Linear threshold model (LTM)

1: $G \leftarrow G(N, p)$ ▷ create Erdos–Renyi network
2: $\bar{\phi} \leftarrow \sim U[0, 1]^N$ ▷ set random threshold values
3: mark all nodes *unlabeled*
4: set A such that $|A| \ll N$ ▷ randomly set a few seeds
5: **while** $\sigma(A)$ changes **do**   ▷ while node labels change
6:     randomly examine all unlabeled nodes u, applying the labeling function

$$g(u, v, \phi) = \begin{cases} \textbf{label u} & v \geq \phi \\ \textbf{no change} & v < \phi \end{cases} \tag{1}$$

for $v = \frac{L(u)}{deg(u)}$, where v is the fraction: $L(u)$ number of node u's neighbors labeled over $deg(u)$.

We can think of the labeling of the seed nodes as a shock or perturbation of the system, which was otherwise at stable equilibrium (Watts, 2002). The *cascade size* (or *activation*) $\sigma(A) \in [0, 1]$ on seed set A is the resulting fraction of nodes that are labeled after the cascade of the LTM above is run in steps 5–6, $\sigma(A) = \frac{num\ labeled}{N}$. A *global cascade* is said to occur if the cascade size attains a predetermined fixed fraction of the network ($\sigma(A) \geq \Phi$), where $\Phi \in [0, 1]$ is a constant. For example, in prior work, a value of $\Phi = 0.1$ has been used (Gray *et al.*, 2018; Watts, 2002), so we would be evaluating $\sigma(A) \geq 0.1$ for the global cascade (Gray *et al.*, 2018; Kempe *et al.*, 2003; Watts, 2002).

The *cascade frequency* is then defined over a number of trials as the fraction of times the cascade size reaches Φ:

$$f_c = \frac{|\sigma(A) \geq \Phi|}{num\ trials} \tag{2}$$

Here, graph edges are considered to be undirected and unweighted. Note that in some formulations of the LTM, edges are directed and have weights (Kempe *et al.*, 2003).

2.2 Cascades and logic

2.2.1 Equivalence of LTM to AND, OR

The threshold function in the LTM can compute logical AND or OR, depending on the threshold value. Low-threshold nodes are easily influenced, corresponding to OR, while high-threshold nodes are more difficult to influence and correspond to AND. Formally, we see that for a node having two neighbors and a threshold $\phi \leq 1/2$, its activation is exactly equivalent to a logical OR. Similarly, when the node has threshold $\phi > 1/2$, it is equivalent to AND (Figure 1) (Wilkerson & Moschoyiannis, 2019).

Generally, for k neighbors, a node behaves like multi-input OR for any $\phi \leq \frac{1}{k}$, and multi-input AND for any $\phi > \frac{k-1}{k}$. For $\frac{1}{k} < \phi \leq \frac{k-1}{k}$, a node behaves like a threshold logic unit (Rojas, 2013). Note that these are monotone Boolean functions (increasing the number of true inputs cannot decrease the number of true outputs).

A	B	ν	C
0	0	0/2	0
0	1	1/2	1
1	0	1/2	1
1	1	2/2	1

≡ **OR**

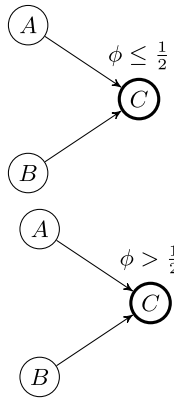| A	B	ν	C
0	0	0/2	0
0	1	1/2	0
1	0	1/2	0
1	1	2/2	1

≡ **AND**

Figure 1. Cascades compute logic: A graph focusing on node *C*'s behavior from neighbors (inputs) *A* and *B* under the original labeling rule (Equation (1)). Top: Truth table for **OR** and corresponding cascade network when node *C*'s threshold $\phi \leq \frac{1}{2}$. Bottom: Truth table for **AND** and cascade network when *C*'s $\phi > \frac{1}{2}$. Networks drawn as directed to show flow of information.
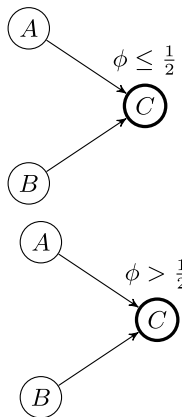
A	B	ν	C
0	0	0/2	1
0	1	1/2	0
1	0	1/2	0
1	1	2/2	0

≡ **NOR**

A	B	ν	C
0	0	0/2	1
0	1	1/2	1
1	0	1/2	1
1	1	2/2	0

≡ **NAND**

Figure 2. Logic under the complementary labeling rule (Equation (3)): Top: Truth table for **NOR** and corresponding cascade network whenever node *C*'s $\phi \leq \frac{1}{2}$. Bottom: Truth table for **NAND** and cascade network whenever *C*'s $\phi > \frac{1}{2}$.
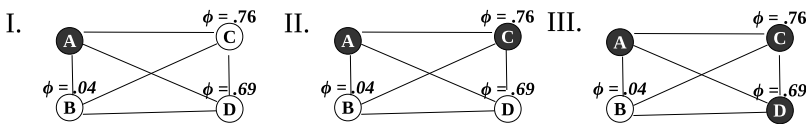

Figure 3. An example of a cascade using the antagonistic model's (ALTM's) rule (Equation (3)), starting at seed node A (step I) and proceeding to nodes C (step II) and D (step III).

2.2.2 Antagonistic cascades and NAND/NOR

Making a single modification to the LTM, we construct a new *antagonistic labeling rule* simply by taking the complement of Equation (1), reversing the inequalities:

$$\neg g(u, v, \phi) = \begin{cases} \textbf{label u} & v < \phi \\ \textbf{no change} & v \geq \phi \end{cases} \qquad (3)$$

again for node *u* and *v* the fraction of neighbors labeled. The result of this rule is that each node computes NAND or NOR. Also note that this function is monotonically decreasing (Figure 2). Antagonistic interactions have been investigated in social networks (Leskovec *et al.*, 2010; Altafini, 2012) and can also be constructed from connected MP neurons using excitatory and inhibitory connections (Rojas, 2013, pp. 34).

For convenience, we will call the LTM with this antagonistic labeling rule (Equation (3)) the ALTM. The ALTM can also undergo cascades. An example of the cascade operation in the ALTM can be seen in (Figure 3).
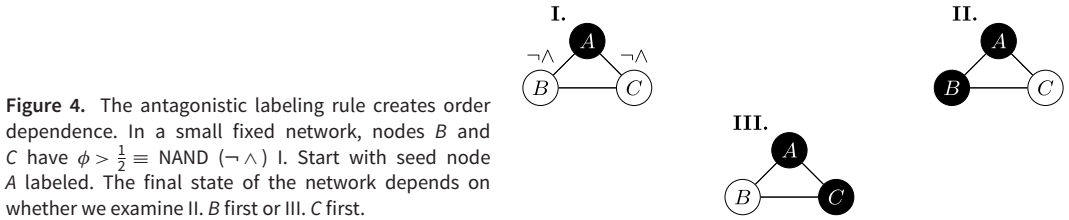
Figure 4. The antagonistic labeling rule creates order dependence. In a small fixed network, nodes B and C have $\phi > \frac{1}{2} \equiv$ NAND ($\neg \wedge$) I. Start with seed node A labeled. The final state of the network depends on whether we examine II. B first or III. C first.

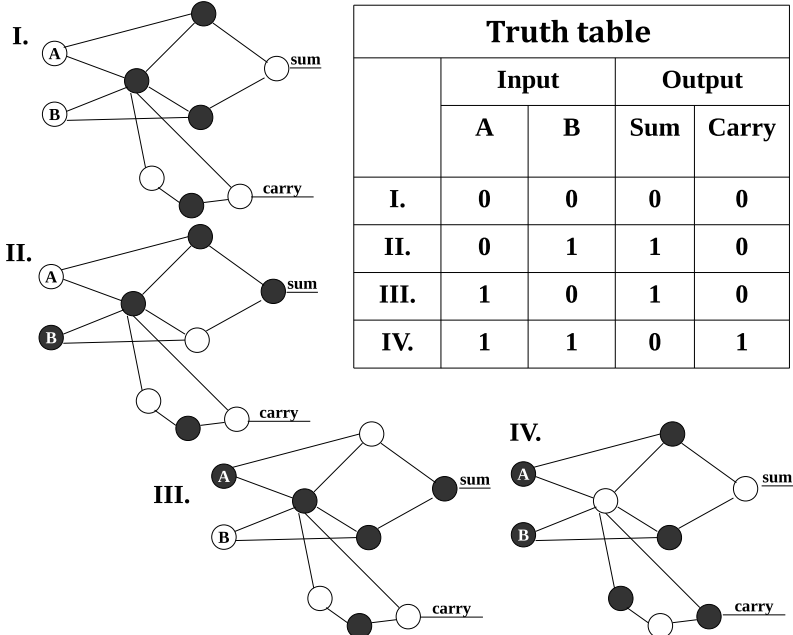| Truth table				
	Input		**Output**	
	A	**B**	**Sum**	**Carry**
I.	0	0	0	0
II.	0	1	1	0
III.	1	0	1	0
IV.	1	1	0	1

Figure 5. A half-adder implemented by an ALTM network having the antagonistic labeling rule (Equation (3)). Here all nodes have been assigned ϕ values equivalent to NAND. This is a very small example of the power of the functionally complete Boolean logic that can be computed by these networks.

Note that while the LTM labeling is stable with respect to the order of node examination and converges to equilibrium, the ALTM is very unstable and order-dependent (Demirel, 2013, p. 13). Chosen in another order, the ALTM's cascade may obtain a different final state of the network labeling as in Figure 4.

2.2.3 Universal Boolean logic and the XOR problem
As stated, the original LTM, computing a monotone Boolean function at each node, can only compute monotonically increasing functions on the network. (There are no negative weights or inhibitory connections.) Therefore, it is unable to compose functions to compute (non-monotone) XOR, a problem faced by early neural networks (Minsky & Papert, 1969; Rojas, 2013).

However, the ALTM's antagonistic labeling rule (Equation (3)) computes NAND or NOR so when connected in a logic circuit can compute universal Boolean logic, including NOT, NAND, XOR, and much more complex Boolean functions. The universality of NAND or NOR composition is well known and is called *functional completeness* (Savage, 1998). An example of this can be seen in a half-adder (Figure 5), a common component in modern integrated logic circuits (Mano, 1993, pp. 19-20).
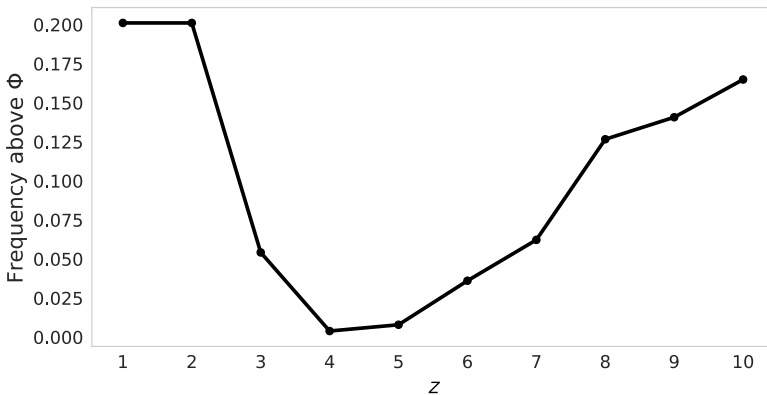
Figure 6. Bimodality in the cascade frequency f_c in the ALTM, here on a network with 10,000 nodes over 100 realizations per average degree z, a fixed $\phi_* = 0.18$ for all nodes, and one labeled seed node. A cascade is called *global* when its activation size is above the median cascade size ($\Phi = 0.62$) of the entire graph.

2.2.4 Cascade frequency

Frequency of global cascades under the ALTM (Figure 6) approximately complements that of the LTM ((Watts, 2002), Figure 2b). We calculate the frequency that the cascade size exceeds the global threshold (Equation (2)) of median cascade size ($\Phi = 0.62$), a fraction of the entire graph, over 100 samples, with $N = 10,000$. Here a fixed node threshold $\phi_* = 0.18$ is assigned to all nodes (as in the original work (Watts, 2002)). The minimum cascade frequency occurs in the same mean degree (z) region where frequency was maximized in the LTM (Watts, 2002).

A *vulnerable node u* is a node having degree k and threshold ϕ small enough that, if only one of u's neighbors becomes labeled, u will become labeled. Near mean degree $z = 5$, we see low cascade frequency (Figure 6). Here the graph may still be tree-like, having a very small clustering coefficient ($\langle C \rangle \propto \frac{z}{N-1} \approx 0.0005$), so that seeds largely reach nodes from only one neighbor. However, since we have chosen $\phi_* = 0.18$, nodes are not vulnerable, since the fraction of neighbors labeled $\nu = 1/5 > 0.18 = \phi_*$ (Watts, 2002).

In the original LTM model, as z increases, densely connected unlabeled nodes tend to dilute or block the effects of incoming labeled nodes, which are often unable to overcome thresholds ϕ, driving down the cascade frequency. Just as there is a "blocking" effect of edge density in the LTM, in the ALTM there is an expected "anti-blocking" effect as z increases (Watts, 2002). In the right-hand side ($z \geq 8$) of the ALTM cascade frequency (Figure 6), unlabeled dense networks tend to be easily influenced by incoming labels, increasing labeling until negative feedback (antagonism) discourages further labeling, as seems indicated by the lower peak in the right-hand mode.

2.3 Cascade as classification

The activation σ of a cascade from a set of labeled seed nodes A can be considered a binary classifier. Here, we mainly consider the original LTM.

We note that for the LTM we can choose a set of nodes to be an *input set*, and rather than choosing any random seed in the graph, only choose seed nodes out of that input set without loss of generality to previous results on cascade percolation (Watts, 2002). Therefore, starting with one possible seed node $a \in A$, so that if $a = node_1$(a labeled seed), $|A| = 1$, or if $a = \emptyset$ (no seed labeled), $|A| = 0$, we can restrict the seed set A to be either that particular node ($A = \{a\}$), or the null set ($A = \emptyset$). We remind the reader that, when we run a cascade on a set of seed nodes, we measure the output as in previous work (Watts, 2002), calling the cascade *global* if it attains some predetermined global threshold Φ, a fraction of the graph's nodes (Equation (2)). We can then consider the cascade action as a *binary indicator* function (*binary classifier*) (Murphy, 2012, p. 3),
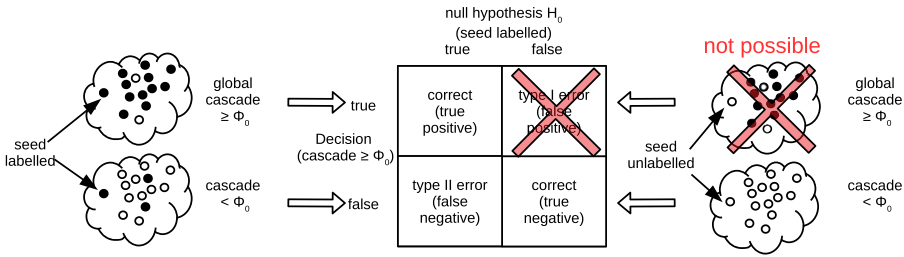
Figure 7. The confusion matrix of seed node classification. The LTM cannot undergo a cascade when there is no seed labeled (false positive); therefore, there is no type I error. Clouds are abstract representations of LTM networks.

where *global* or *not global* cascade equate to True or False, respectively, as we see in statistical learning (e.g., medical testing). The function $\hat{y}$ estimates the classification of the activation $\sigma(A)$ on seed node set A, evident in the confusion matrix (Figure 7).

The classification is therefore,

$$\hat{y}(G, A, \bar{\phi}) = \begin{cases} \textbf{True} & \sigma(A) \geq \Phi \\ \textbf{False} & \sigma(A) < \Phi \end{cases} \tag{4}$$

for graph G, seed node set A, vector of node thresholds $\bar{\phi} \in [0, 1]^N$, activation (cascade) function $\sigma : X \to Y, X \subseteq A, Y \in \{False, True\}$, and global threshold $\Phi \in [0, 1]$. Intuitively, we can then see $\hat{y}$ as classifying the truth or falsity of the input seed set A via the cascade activation σ. From a popular classification example, for a particular data point (e.g., patient), we can then consider labeling or not labeling of LTM seeds as presence or absence of particular features (e.g., fever, cough, etc.) and the hypothesis as ill (True) or not ill (False). If only one seed is present, only one feature is being used for classification.

This has some interesting implications for accuracy, particularly sensitivity and specificity, as follows:

When a data point is "classified" via its input feature (seed) by $\hat{y}$ (Equation (4)), it is possible to measure *sensitivity* or *true positive rate* (TPR) in the usual way ($TPR = \frac{TP}{N^+}$), whereby $N^+ = TP + FN$, the true positives together with false negatives make up the number of positive results also seen below (Equation (5), LHS) (Murphy, 2012, pp. 181–182).

However, as we have seen, the LTM composes monotone functions; therefore, *cascades cannot occur with no seed labeled* (Figure 7). Thus, it is not useful or informative to run cascades without seed nodes, since trivially true negatives equal number of negative cases, $TN = N^-$, so here we do not run them. Therefore, here $N_{trials} = N^+$, the number of trials equals the number of positive results. *Cascade frequency* measures the number of times *global cascades* (true positives) occur over the total number of trials (Equation (2)), and thus (Equation (5), RHS). Letting TP be number of true positives ($\sigma(A) \geq \Phi$) and FN number of false negatives, $TP + FN = N_{trials}$ (Figure 7), we thus obtain that *cascade frequency is equivalent to sensitivity*:

$$TPR = 1 - \beta = \frac{TP}{N^+} = \frac{TP}{(TP + FN)} = \frac{TP}{N_{trials}} = f_c, \tag{5}$$

where $1 - \beta$ is sensitivity, β is type II error, and f_c is cascade frequency.

Since it is impossible to obtain a cascade when no seed is labeled ($A = \emptyset$), the number of false positives is always zero (Figure 7), and type I error is absent, giving us

$$TNR = 1 - \alpha = \frac{TN}{(TN + FP)} = \frac{TN}{(TN + 0)} = 1, \tag{6}$$

where TNR is true negative rate, equivalent to specificity ($1 - \alpha$) (Murphy, 2012, p. 181), α is type I error, TN are number of true negatives, and FP are number of false positives.

However, we briefly note that the ALTM, since it computes universal logic, is able to undergo cascades both when seeds are labeled and unlabeled, as we see from the truth tables for NAND/NOR (Figure 2), and in the half-adder (Figure 5). As a result, both type I and type II error can be obtained.

2.4 Learning in the LTM

For the remainder of this paper, we will discuss basics of learning in the original LTM (Watts, 2002). As this model is already well known, and holding to our policy of simplicity, we believe it is valuable to understand as the first step in the context of statistical learning. Here we take some initial experimental and theoretical steps toward sources of efficiency in biologically motivated model of learning in cascades.

We, therefore, begin with the seemingly trivial problem of classifying data points having a single feature, that is, a single seed node, evaluating $\hat{y}$ (Equation (4)) on a seed set A when $|A| = 1$. (Intuitively, we can think of this as classification of a set of data points only having one feature.) This is already interesting, since in the LTM, type II error is possible (Figure 7).

Classifying a data point having a single feature (seed) using a cascade is also interesting because it is a nontrivial NP-hard problem, as can be seen in the theoretical work we have mentioned on *influence maximization*, maximizing activation of a network (cascade size) from a set of seed nodes (Domingos & Richardson, 2001; Kempe *et al.*, 2003). Formally, the problem is: *Given a Linear Threshold Model, which k edges maximize the activation for a fixed set of seeds?* (Khalil *et al.*, 2014).

As the first step, it is sufficient to simply maximize the cascade frequency when the seed node is labeled. Doing so will minimize type II error (false negatives) and thereby increase sensitivity (Equation (5)), as described below.

2.4.1 The LTM as a statistical learning model

Although we have formulated it above as a classifier, it is not apparent that the LTM can be considered formally as a learning model (Vapnik, 1999, pp. 17–19), so first we frame it as such. In this context, we can consider the edge probability p of graph G and the node thresholds vector $\bar{\phi}$ as making up the parameter set Λ of the model, such that $\Lambda = \{p, \bar{\phi}\}$. In learning, we need to find the best Λ that minimizes the loss L between the true seed value y, and the $\hat{y}$, the response of the learning model (Equation (4)). Here the simple misclassification (or "0–1") loss (Murphy, 2012, pp. 177) is used, thus

$$L(y, \hat{y}(x, \alpha)) = \begin{cases} \mathbf{1} & y \neq \hat{y}(x, \alpha) \\ \mathbf{0} & y = \hat{y}(x, \alpha) \end{cases} \tag{7}$$

where $\alpha \in \Lambda$, the set of parameters $\{p, \bar{\phi}\}$, and choice of seed node $x \in A$ is in the input vector, for the seed set A.

The meaning of the above formalism is that we can randomly add edges or choose values of ϕ to minimize the loss. In the present work, we are interested in the parameter of topology (edges), so do not study values of ϕ. This means *we are learning the value of p that minimizes the loss*, so here $\Lambda = \{p\}$, the set of all possible edge probabilities, and $\alpha = p_i$, a particular edge probability.

Since, as shown above, type I error does not occur (Figure 7), the only case where $L = 1$ in the loss (Equation (7)) is the false negatives. Thus, the loss to be minimized is simply the type II error, or false negative rate:

$$FNR = \beta = \frac{FN}{(TP + FN)} = \frac{FN}{N_{trials}} = 1 - f_c \tag{8}$$
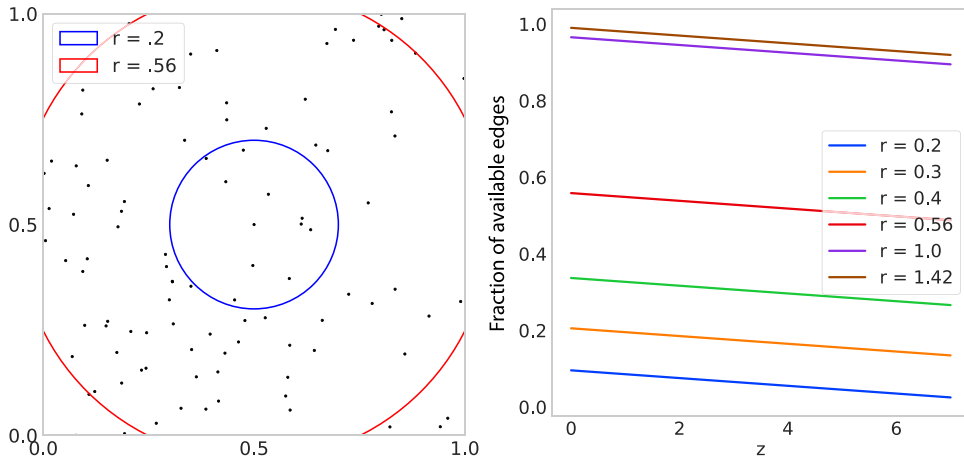
Figure 8. (Left) Examples showing circles of possible neighbors for node centered at (0.5, 0.5) within the unit square, having radii = 0.2 and 0.56 (areas 0.13 and 1.0, respectively). (Right) Fraction of available edges under radius r over complete graph edges $e_r = |E_a|/|E_{max}|$ for $|E_{max}| = N(N-1)/2$. This fraction behaves as expected, gradually decreasing as random edges are added, increasing mean degree z. Note that few edges are excluded by $r = 1.0$, although the maximum distance in the unit square is $1.42 \approx \sqrt{2}$.

where β is commonly known as the *false negative rate* (*FNR*) (Murphy, 2012, p. 181) and f_c is again the cascade frequency. Thus, minimizing the loss is equivalent to maximizing the cascade frequency, or sensitivity (Equation (5)).

As we are at the initial stage of this investigation, it is important to understand this rudimentary behavior from a single seed and the basic connectivity p before trying more sophisticated learning algorithms.

2.4.2 Spatial networks and cascade efficiency

As stated in the introduction, our eventual goal is to understand the efficiency of biologically inspired learning compared to artificial learning. Here, we examine the benefits of spatial distance for maximizing the cascade size and frequency, thereby minimizing loss while examining a restricted number of edges to do so. Since in neuronal networks, connections have a metabolic cost (Lynn & Bassett, 2019), this is a simple study of the effect of spatial constraints.

For this reason, we initially choose the random geometric graph—a kind of spatial Erdos–Renyi topology (Penrose, 2003, pp. 1–2). We lay out the nodes of the random geometric graph distributed randomly and uniformly on the unit square in positions $x, y \sim U[0, 1]^2$. For all experiments, and due to computational constraints, we use $N = 100$ nodes over 200 trials.

We study the effect of spatial distance when creating edges in these graphs. That is, we impose a radius r and only allow nodes within that radius to connect. This is motivated by several factors. In the random geometric graph case, it has been observed that smaller connection distance results in a lattice-like topology, while a larger distance results in an Erdos–Renyi topology (Lynn & Bassett, 2019). It has also been observed that in spatial graphs, a smaller connection distance can reduce the "blocking" effect of graph density (Gray *et al.*, 2018; Watts, 2002).

Therefore, the parameters for these experiments are the distance (radius) at radius values $r = 0.2, 0.3, 0.4, 0.56, 1.0,$ and 1.42, within the unit square (Figure 8). Below $r = 0.2$, very few edges are available, approaching 0 as edges are added (Figure 8, right). At $r = 0.56$, the circle around a node has area approximately 1 (Figure 8, left), and $r = 1.42 \approx \sqrt{2}$ is the maximum distance (along the diagonal) of the unit square. Few radius values are shown between $r = 0.56$ and $r = 1.42$, for clarity, and because cascade effects are not significantly different (Figure 9).
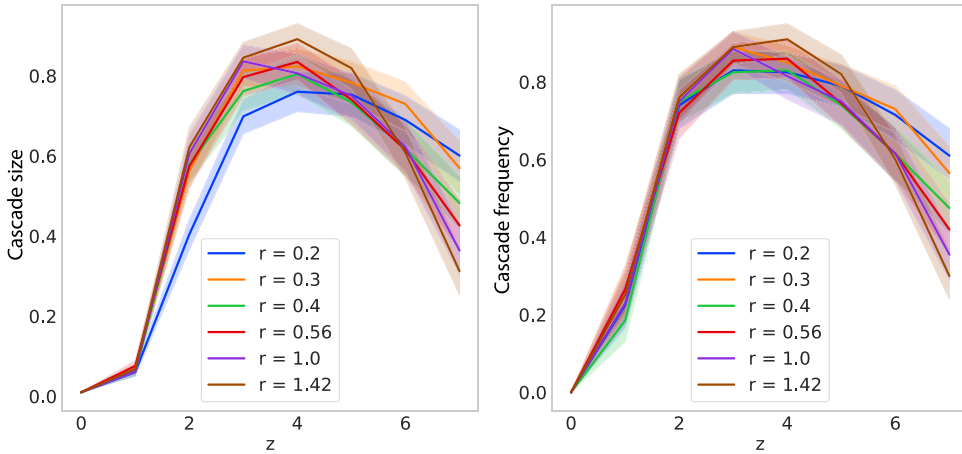
Figure 9. (Left) Mean cascade size versus mean degree z shows similar behavior for $r = 0.2, 0.3, 0.4, 0.56, 1.0,$ and 1.42, near $z = 1$. Notably, the small $r = 0.2$ (blue) still achieves large cascade size. However, near $z = 7$, there is significantly less "blocking" effect for smaller radius (blue) than larger (brown). Shading shows 95% confidence interval. (Right) Cascade frequency shows similar results ($\Phi = 0.1$). Most notably, sudden increase in cascade frequency implies explosive increase in sensitivity (true positive rate) (Equation (5)). Networks have 100 nodes over 200 trials.

In preparation to understanding cascade frequency versus the number of available edges due to radius, we define *cascade efficiency* e_c to be the ratio of cascade frequency f_c to fraction of available edges:

$$e_c = \frac{f_c}{e_r} \tag{9}$$

where fraction of available edges under radius r is $e_r = |E_a|/|E_{max}|$, such that $|E_a|$ is the number of available edges in the graph within a given radius r, and $|E_{max}| = \frac{N(N-1)}{2}$, the maximum number of edges for the Erdos–Renyi graph of size N (Figure 8, right). Intuitively, e_c is a measure of the accuracy contribution per available edge in the graph.

We start each experiment with an empty graph and add random edges one at a time, so the average degree (z) ranges from 0 to 7 (recalling that the average degree relates to the edge probability in Erdos–Renyi graphs by $z = p(N - 1)$). Since the edges added are random and $p = \frac{2E}{N(N-1)}$, this remains a random geometric graph (Penrose, 2003).

2.4.3 Hypotheses

We hypothesize that cascades under a smaller radius will not experience the 'blocking' effect of edge density as edges are added (Gray *et al.*, 2018; Watts, 2002).

We also expect that, as the radius approaches r=0.56 and above, where the circle area ≥ 1 (equal to the area of the unit square in which the nodes are positioned), the cascade size and frequency will be similar that of larger radii.

Furthermore, as a smaller radius reduces the number of available edges and therefore available nodes due to sparseness, we expect maximum cascade size to increase as radius increases. Subsequently, we therefore also expect maximum cascade frequency to increase with radius.

As maximum cascade size and frequency remain relatively high, even for small radii that utilize smaller numbers of edges, we expect cascade efficiency for small radii to be significantly higher than for large radii.

Due the above results, we therefore expect an inverse relationship between maximum cascade frequency and maximum efficiency.

2.4.4 Experiment

The steps for our experiment are the following (Wilkerson, 2021):

---

2 Experiment

1: $r \leftarrow [0, 1.42]$ and $\Phi \leftarrow [0, 1]$ ▷ set radius and global threshold
2: **for** 200 trials **do**
3:      $G \leftarrow G(N, p)$ ▷ create Erdos–Renyi network
4:      $\bar{\phi} \leftarrow [0.18, ..., 0.18]^T$ ▷ use same threshold value for all nodes
5:      $x, y \sim U[0, 1)^2$ ▷ lay out nodes randomly in the unit square
6:      $A \leftarrow \{a = node_0\}$ ▷ assign one seed node
7:      **while** mean degree $z \leq 7$ **do**
8:          reset labeling of all nodes except seeds (A)
9:          run cascade $\sigma(A)$
10:        record cascade size and number of edges
11:        add 50 random edges (nodes with mutual distance $\leq r$)

12: $f_c = |\sigma(A) \geq \Phi|/\text{num trials}$ ▷ calculate cascade frequency

---

2.4.5 Experimental results

The first hypothesis, that under a smaller radius there will be a lower blocking effect as density increases ($z \geq 4$), is supported by the results (Figure 9, right), as we see that for $r \leq 3$. Cascade size and frequency are significantly higher than for larger radii.

The second hypothesis seems to be confirmed, experimental results for $r \geq 0.56$ are not significantly different from one another (Figure 9, right). Surprisingly, $r = 4$ (circle Area ≈ 0.5) also has a similar behavior to larger radii.

Cascade size and frequency for smaller radii ($r \leq 0.3$) are significantly lower than that of the largest radius ($r = 1.42$), in the domain $z \geq 3$ (Figure 9). Also cascade size for $r = 0.2$ is significantly smaller in $2 \leq z \leq 4$. Thus, there is some increase in cascade size and frequency as radius increases. However, we note that near $z = 3$, cascade size and frequency for $r = 0.3$ and $r = 1.0$ are not significantly different; therefore, it does not always hold that larger radius leads to larger cascade size or frequency.

Results also indicate that cascade efficiency is indeed significantly higher for small radii than large radii (Figure 10, left), especially near maximum cascade size and frequency $z = 4$ (Figure 9), indicating more efficient use of edges to obtain cascade frequency. We also note that for the smallest radius $r = 0.2$, the efficiency gap grows with z.

Finally, a plot of the Pareto front of maximum cascade efficiency versus maximum cascade frequency seems to indicate that there is a inverse relationship between these (Figure 10, right). Pearson's correlation between these factors of -0.57 with p-value 0.24 confirms this.

2.4.6 Criticality, loss, and cascade frequency

As mentioned above, in many self-organizing systems, cascades are *critical* phenomena. That is, the cascade size or frequency may suddenly—continuously or discontinuously ("explosively")—percolate when the micro-scale control parameter nears a particular critical value (Watts, 2002). This implies that a small change in the control parameter can lead to a sudden change in the cascade frequency, which by (Equation (8)) is equivalent to a *sudden or implosive reduction in the loss function*. Also, in the LTM, maximum cascade frequency has previously been found to be maximized at connectivity far below the maximum degree ($z \approx 3.7 \ll N - 1$) (Watts, 2002).
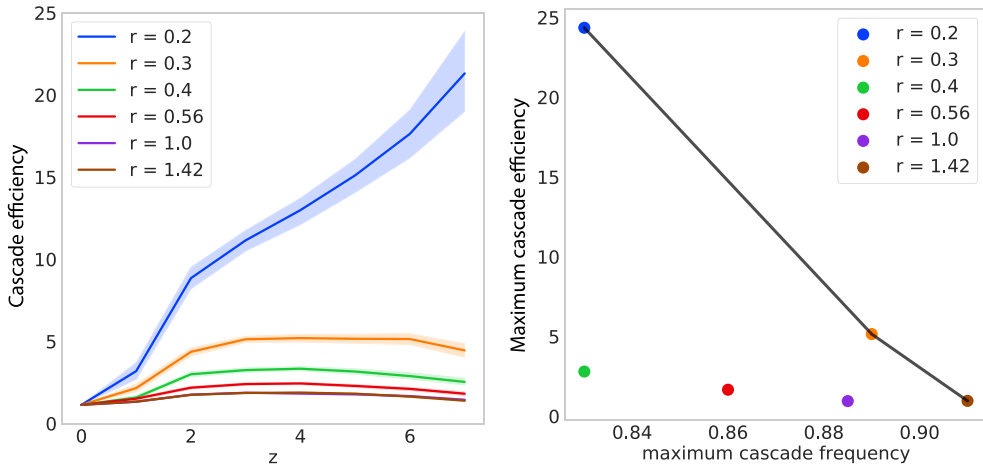
Figure 10. (Left) Cascade efficiency e_c (Equation (9)) versus mean degree z. Shading shows 95% confidence interval. (Right) Pareto front of maximum cascade efficiency (maximum cascade frequency/available edges) versus maximum cascade frequency. Note that $r = 0.2$ and $r = 0.3$ (area = 0.126 and 0.283) lie on the front and achieve frequency > 0.8, despite only covering a small fraction of the unit square.
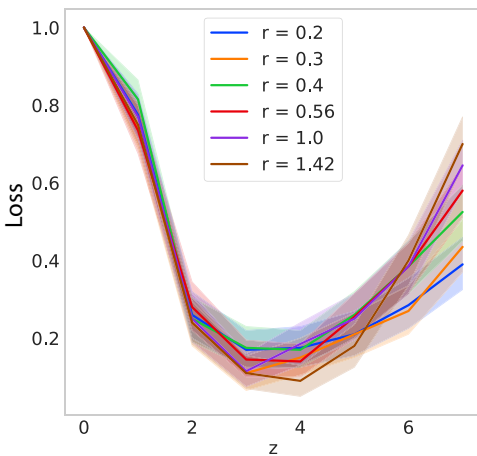

Figure 11. Loss: The false negative rate β is 1—sensitivity, therefore 1—cascade frequency (Equation (8)). As there is no type I error, the false negative rate is our loss (Equation (7)) and is minimized when cascade frequency is maximized. Therefore, percolation of cascade frequency can mean sudden reduction in loss. The graph of $r = 1.42$, where distance plays no role, illustrates previous theoretical results for criticality (Watts, 2002). Other distances are shown for comparison. Shading shows 95% confidence interval.

Therefore, this criticality and percolation of cascade frequency can be another source of efficiency in learning, as the loss gradient may be sudden and steep for such systems, and the minimum attained quickly. This means that the amount of edge rewiring may be significantly reduced in order to reach a near-efficient solution to cascade size.

As a preliminary result, one can already observe this critical behavior at $N = 100$ for $r = 1.42$ near $z = 1$ and $z = 4$ (Figure 11). The random geometric graph having radius $r = 1.42 > \sqrt{2}$ (the maximum distance in the unit square) is an Erdos–Renyi graph as used in the LTM, since radius no longer plays a role in connections. It has been shown previously that a critical phase transition in cascade frequency occurs in the LTM as a function of average node degree z (Watts, 2002).

For example, here for $r = 1.42$ (Figure 11) observe that the loss is already approaching its minimum near $z = 2$, having $\sim 1/2$ the edges than at $z = 4$ (recall $\langle E \rangle = \frac{N(N-1)p}{2} = \frac{Nz}{2}$). For a large graph, this could mean a very significant connection savings. Thus, this is another potential source of efficiency in this kind of learning model. It remains to be shown theoretically and experimentally for larger N that LTM cascades in random geometric graphs are subject to criticality, although
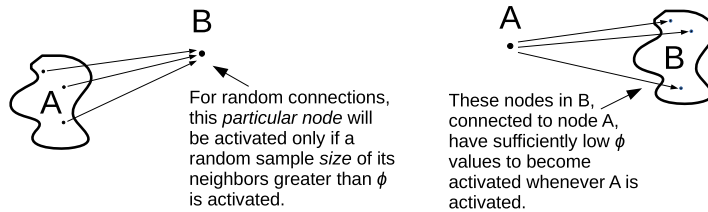
Figure 12. The LTM maps between two information representations in networks, choice of *particular nodes labeled* (seed node choice), and *number of nodes labeled* (seed size or cascade size). Left: Node *B* with random connections to nodes in network *A* can be seen to randomly sample the activation in *A* and become activated if the cascade in *A* is sufficiently large ($\sigma(A) \geq \Phi$), mapping the cascade size in *A* to the particular node *B*. Right: Nodes in network *B* having sufficiently low ϕ thresholds to be vulnerable to the activation of node *A* map activation of *A* to the seed size of *B*. Edges are drawn as directed to show information flow.

theoretical results for percolation on random geometric graphs (Penrose, 2003) and experimental results for cascades in similar Waxman graphs have been demonstrated (Gray *et al.*, 2018).

Treating the LTM as a learning model, loss as a function of mean degree *z* is a kind of *learning curve* (Figure 11). Training is simple addition of random edges, and loss decreases suddenly until around the point $z = 4$, where it begins to increase again. One could call this increase *over-fitting* in this context (Murphy, 2012, p. 22)—we have continued to present the same input to the model as we train it, adding edges and increasing its complexity, but the classification performance deteriorates. Preliminary results on small networks seem to indicate that the rate of over-fitting is also reduced for smaller radii as $z \geq 5$ (Figure 11).

2.5 Input/output and information representation

We have shown how the LTM can act as a classifier and statistical learning model. However, we must pose a basic question: How can or should information be represented in an LTM model? Particularly, if considering this as a learning algorithm, how should input and output be passed to the LTM network?

Choice of particular nodes activated and *number of nodes activated* are two basically different ways of representing or encoding information in the LTM (in the brain corresponding to sparse or population coding, respectively (Dayan & Abbott, 2001, pp. 97, 379)), which seems a problem when we are choosing how to consider the LTM as a learning model in real-world contexts. The choice between these two representations can be seen in the influence maximization and global cascade model literature, discussing seed node selection (particular nodes) (Kempe *et al.*, 2003), or seed and cascade size (number of nodes) (Gleeson & Cahalane, 2007; Watts, 2002).

Biological or social networks are often organized in a modular fashion, so that the *output of one (sub-) network may become the input of another*. This is especially evident in the brain, where cortices or other modular structures process information and pass the results to other structures (Lynn & Bassett, 2019). This is a question of *information coding* of these networks, and since we are also interested in biologically inspired algorithms, relates to *neural coding*, how information is passed between neuronal avalanches, here occurring in the LTM, before and after it undergoes a cascade (Dayan & Abbott, 2001).

Therefore, the question remains—how should input or output information be encoded in our LTM learning model, by choosing particular nodes (e.g., binary encoding) or by number of nodes (e.g., unary encoding) (Sayood, 2012, pp. 75, 102)? Fortunately, the LTM solves this dilemma naturally. If we consider an LTM network *A*, randomly connected to a particular node *B* in another network (Figure 12, left), it is exactly the LTM threshold labeling rule (Equation (1)) which tells us that, if sufficient (random) nodes in *A* are activated, *B* will be activated. Since the connections are random, this is a random sampling of cascade size within *A*. Thus, *cascade size is mapped to*

activation of a particular node. On the other hand, if we consider a particular node in network *A*, connected to a set of nodes in network *B*, having sufficiently low-threshold (φ) values (Figure 12, right), if node *A* is activated, the set of nodes in *B* will be activated. *This maps activation of a particular node A to seed size in B*. Of course, this mapping between node choice and number of nodes occurs within the LTM network itself as well. Therefore, the LTM can map both ways between number of nodes and particular node encodings.

The implications of this are that, in a learning or classification context, the LTM may be robust to a variety of input or output representations. That is, it may be possible to use either seed size or choice of particular seeds, or some combination of the two (patterns, clusters of activated nodes), as input or output of an LTM, enabling us to pass complex input (e.g., an image) to the network or out of it.

3. Discussion and conclusions

3.1 Discussion

Here, we have explored several sources of computational power and efficiency in the simple, biologically motivated LTM, both as a logic computer and as a statistical learning model.

Universal Boolean logic of the ALTM means that any logical function can be computed if the network is sufficiently large. Given the simplicity of this model, this gives some indication that further theoretical connections can be made between percolation in networks and theory of Boolean circuits.

One may view the lack of false positives in the LTM as a form of efficiency or increased accuracy as well. It may be possible to take advantage of this, along with the universality of the ALTM, by constructing a hybrid model containing a balance of nodes having threshold (LTM) and antagonistic (ALTM) rules. This balance, together with an appropriate global cascade threshold, may act as a noise filter (Smith *et al.*, 1997; Rubin *et al.*, 2017), whereby cascades compute universal logic but only occur for sufficiently large activation.

Spatial organization is another substantial qualitative difference between naturally occurring networks—for example, neurological networks subject to physical processes and metabolic constraints—and theoretical networks (Lynn & Bassett, 2019). Spatial characteristics can greatly benefit self-organization for cascades in the LTM in several ways: first, with spatial information, the search space for edges or nodes is greatly reduced as we have seen (Figure 8), significantly increasing the efficiency of information processing. Second, by only searching and connecting to local nodes, the average degree of the graph is reduced, preserving node vulnerability to cascades. Therefore, reducing the possible maximum connections may preserve or increase the cascade probability and prevent any "blocking" that would normally result from network density (Gray *et al.*, 2018; Watts, 2002). Thus, the average cascade size and frequency per number of nodes examined tends to increase another source of efficiency in the LTM when spatially organized. Third, spatial organization can lead to different topologies and information mixing as discussed below.

Criticality has been discussed here as a possible way to greatly speed up learning. As we have seen, the loss function may decrease rapidly as random edges are added when percolation occurs. We see this as important, as there can be a *phase transition in the loss* in response to the connectivity control parameter (average number of edges, or equivalently edge probability *p*). It has been shown that there are other information processing benefits to critical cascades in the "criticality hypothesis" (Beggs & Plenz, 2003; Shew & Plenz, 2013), and it appears that the brain tends to remain at homeostasis near criticality exactly for these benefits (Hesse & Gross, 2014).

We have also seen that the LTM may be robust to several ways of representing information. The way data are passed into, out of, and within the LTM network seems to tolerate variation in the encoding of information between number of nodes and particular nodes activated.

3.2 Conclusion

The simple LTM and its biologically inspired variants yield computational power, learning efficiency, and robustness. The LTM computes logical AND and OR, and replacing the threshold rule with its complement allows the ALTM to compute universal logic. The LTM can be framed as a classification model that does not produce type I error.

The LTM can also be formulated as a model of statistical learning, trainable by adjusting its edges. Spatial constraints on edge formation can significantly increase the efficiency of cascades, requiring examination of fewer edges to achieve similar cascade frequency, meaning efficient increases in accuracy, supported by our experimental results. Criticality is also a potential source of learning efficiency, with loss decreasing as an abrupt phase transition as percolation occurs. We have shown the theoretical relationship between well-known cascade criticality and accuracy. Finally, the LTM seems to naturally and robustly map between apparently different information representations. These examinations therefore begin to bring together theories of percolation, Boolean logic, and statistical, and biological learning in this basic ubiquitous model.

4. Future work

There remains much to be done in future:

Brain cortices can have modular organization, where networks are arranged into subnetworks. Therefore, it is interesting to consider how learning can occur in modular topologies such as the stochastic block model (Karrer & Newman, 2011). Similarly, layered topology can be found both in naturally occurring and artificial neural networks, so should be investigated (Lynn & Bassett, 2019).

It is also usual that one would make trials on scale-free (preferential attachment) graphs, as well as small-world graphs (Newman, 2018), since these have different properties of centrality and modularity, and can have different computational implications in terms of integration or segregation of information (Bassett *et al.*, 2011; Lynn & Bassett, 2019). Similar to decision trees, it may be that large categorical decisions embodied in long, small-world connections lead to more efficient learning (Hastie *et al.*, 2009; Watts & Strogatz, 1998), so that rather than forcing all information to pass through all nodes or layers, it can be efficiently categorized (Rojas, 2013). Therefore, more investigation comparing lattice-like or layered topologies versus other topologies may be fruitful in this context. Along with this, the interaction of topology with the information contained in multi-class and multi-dimensional input data may lead to insights in efficiency. Intuitively, the question here is: How should the network topology or learning (rewiring) method correspond to the shape of the data?

The latter experiments will also require practical trials in the encoding of information as it is passed into and out of the LTM. Therefore, the robustness of the LTM to these representations can be studied in more detail.

Also, it remains to be seen whether other training schemes can be more effective. Neurobiology has given us a rich literature on topics such as neuroplasticity and long-term potentiation vis-a-vis learning in the brain (Bassett *et al.*, 2011; Lynn & Bassett, 2019). Much of this originated with the ideas of Hebb, and versions of it have been adapted to machine learning for many years (Coolen, 1998; Hebb, 2005; Kato & Ikeguchi, 2008; Rojas, 2013). Therefore, rewiring methods based on more recent research of biological learning may also be investigated. More advanced and biologically motivated algorithms could involve edge creation or deletion methods based on a combination of activation, proximity, and degree. For example, in future, we may also investigate edge creation between labeled and unlabeled nodes based on spatial proximity.

Together with the discussion of criticality above, this also motivates investigation into guided self-organization (Prokopenko, 2009), whereby nodes make their own connection decisions, leading to emergence of optimal global states (e.g., minimal loss). This can perhaps reap other time efficiency rewards if rewiring is performed concurrently at the micro-scale.

We intend to continue studying the simplest possible models demonstrating biologically motivated and efficient learning behavior, as these help us make theoretical progress, identify basic sufficient conditions, and understand learning in the light of emergent complexity.

Conflict of interest. None.

References

Altafini, C. (2012). Consensus problems on networks with antagonistic interactions. *IEEE Transactions on Automatic Control*, *58*(4), 935–946.

Bak, P., Tang, C., & Wiesenfeld, K. (1987). Self-organized criticality: An explanation of the 1/f noise. *Physical Review Letters*, *59*(4), 381.

Barra, A., Genovese, G., Sollich, P., & Tantari, D. (2017). Phase transitions in restricted boltzmann machines with generic priors. *Physical Review E*, *96*(4), 042156.

Barthélemy, M. (2011). Spatial networks. *Physics Reports*, *499*(1–3), 1–101.

Bassett, D. S., Wymbs, N. F., Porter, M. A., Mucha, P. J., Carlson, J. M., & Grafton, S. T. (2011). Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, *108*(18), 7641–7646.

Beggs, J. M., & Plenz, D. (2003). Neuronal avalanches in neocortical circuits. *Journal of Neuroscience*, *23*(35), 11167–11177.

Bengio, Y., Lee, D.-H., Bornschein, J., Mesnard, T., & Lin, Z. (2015). Towards biologically plausible deep learning. *arxiv preprint arxiv:1502.04156*.

Bohte, S. M, Kok, J. N., & La Poutre, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, *48*(1–4), 17–37.

Bruce, A. D., Gardner, E. J., & Wallace, D. J. (1987). Dynamics and statistical mechanics of the hopfield model. *Journal of Physics A: Mathematical and General*, *20*(10), 2909.

Callaway, D. S., Newman, M. E. J., Strogatz, S. H., & Watts, D. J. (2000). Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, *85*(25), 5468.

Chen, D., Lü, L., Shang, M.-S., Zhang, Y.-C., & Zhou, T. (2012). Identifying influential nodes in complex networks. *Physica A: Statistical Mechanics and Its Applications*, *391*(4), 1777–1787.

Conte, T., DeBenedictis, E., Ganesh, N., Hylton, T., Still, S., Strachan, J. P., Williams, R. S., Alemi, A., Altenberg, L., Crooks, G., *et al.* (2019). Thermodynamic computing. *arxiv preprint arxiv:1911.01968*.

Coolen, A. C. C. (1998). A beginner's guide to the mathematics of neural networks. In *Concepts for neural networks* (pp. 13–70). Springer.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational Neuroscience Series.

Demirel, Y. (2013). *Nonequilibrium thermodynamics: transport and rate processes in physical, chemical and biological systems*. Newnes.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 57–66).

Easley, D., Kleinberg, J, *et al.* (2010). *Networks, crowds, and markets*. Vol. 8. Cambridge: Cambridge University Press.

Gao, J., Zhou, T., & Hu, Y. (2015). Bootstrap percolation on spatial networks. *Scientific Reports*, *5*, 14662.

Gleeson, J. P., & Cahalane, D. J. (2007). Seed size strongly affects cascades on random networks. *Physical Review E*, *75*(5), 056103.

Granovetter, M. (1978). Threshold models of collective behavior. *American Journal of Sociology*, *83*(6), 1420–1443.

Gray, C., Mitchell, L., & Roughan, M. (2018). Super-blockers and the effect of network structure on information cascades. In*Companion proceedings of the the web conference 2018* (pp. 1435–1441).

Häggström, O. (2000). Markov random fields and percolation on general graphs. *Advances in Applied Probability*, *32*(1), 39–66.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology Press.

Hesse, J., & Gross, T. (2014). Self-organized criticality as a fundamental property of neural systems. *Frontiers in Systems Neuroscience*, *8*, 166.

Jalili, M., & Perc, M. (2017). Information cascades in complex networks. *Journal of Complex Networks*, *5*(5), 665–693.

Jarman, N., Steur, E., Trengove, C., Tyukin, I. Y., & van Leeuwen, C. (2017). Self-organisation of small-world networks by adaptive rewiring in response to graph diffusion. *Scientific Reports*, *7*(1), 1–9.

Karrer, B., & Newman, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, *83*(1), 016107.

Kato, H., & Ikeguchi, T. (2008). Self-organized complex neural networks through nonlinear temporally asymmetric hebbian plasticity. In *International conference on artificial neural networks* (pp. 623–631). Springer.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 137–146).

Khalil, E. B., Dilkina, B., & Song, L. (2014). Scalable diffusion-aware optimization of network topology. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1226–1235).

Leskovec, J., Huttenlocher, D., & Kleinberg, J. (2010). Signed networks in social media. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1361–1370). ACM.

Lynn, C. W., & Bassett, D. S. (2019). The physics of brain network structure, function and control. *Nature Reviews Physics*, *1*(5), 318.

Mano, M. M. (1993). *Computer system architecture*. Prentice-Hall, Inc.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133.

Minsky, M. L., & Papert, S. (1969). *Perceptrons; an introduction to computational geometry*. MIT Press.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Newman, M. (2018). *Networks*. Oxford university press.

Newman, M. E. J. (2002). Spread of epidemic disease on networks. *Physical Review E*, *66*(1), 016128.

Penrose, M. (2003). *Random geometric graphs*. Vol. 5. Oxford University Press.

Prokopenko, M. (2009). *Guided self-organization*. Taylor & Francis.

Rojas, R. (2013). *Neural networks: a systematic introduction*. Springer Science & Business Media.

Rubin, R., Abbott, L. F., & Sompolinsky, H. (2017). Balanced excitation and inhibition are required for high-capacity, noise-robust neuronal selectivity. *Proceedings of the National Academy of Sciences*, *114*(44), E9366–E9375.

Sakoda, J. M. (1971). The checkerboard model of social interaction. *The Journal of Mathematical Sociology*, *1*(1), 119–132.

Savage, J. E. (1998). *Models of computation*. Vol. 136. Addison-Wesley Reading, MA.

Sayood, K. (2012). *Introduction to data compression*. Newnes.

Schelling, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology*, *1*(2), 143–186.

Shew, W. L., & Plenz, D. (2013). The functional benefits of criticality in the cortex. *The Neuroscientist*, *19*(1), 88–100.

Smith, S. W., *et al.* . (1997). *The scientist and engineer's guide to digital signal processing*. California Technical Pub. San Diego.

Stauffer, D., & Aharony, A. (2018). *Introduction to percolation theory*. CRC press.

Vapnik, V. (1999). *The nature of statistical learning theory*. Information Science and Statistics. Springer New York.

Wang, C., Komodakis, N., & Paragios, N. (2013). Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, *117*(11), 1610–1627.

Watts, D. J. (2002). A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, *99*(9), 5766–5771.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440.

Wilkerson, G., & Moschoyiannis, S. (2019). Universal boolean logic in cascading networks. In *International conference on complex networks and their applications* (pp. 601–611). Springer.

Wilkerson, G. J. (2021). *Logic and learning in network cascades*. https://github.com/galenwilkerson/Logic-and-Learning-in-Network-Cascades