Learning Hierarchically Structured Concepts

Nancy Lynch
CSAIL, MIT

Frederik Mallmann-Trenn
King's College London*

February 11, 2020

Abstract

We use a recently developed synchronous Spiking Neural Network (SNN) model to study the problem of learning hierarchically structured concepts. Specifically, we introduce a data model that describes simple hierarchical concepts. We define a biologically plausible layered SNN model, with learning modeled using Oja's local learning rule, a well known biologically plausible rule for adjusting synapse weights. We define what it means for such a network to recognize hierarchical concepts; our notion of recognition is robust, in that it tolerates a bounded amount of noise.

Then, we present two unsupervised learning algorithms by which a layered network may learn to recognize hierarchical concepts according to our noise-tolerant definition. We analyze correctness and performance formally; the amount of time required to learn each concept, after learning all of the sub-concepts, is approximately $O\left(\frac{1}{\eta k}\left(\ell_{\max}\log(k)+\frac{1}{\epsilon}\right)+\log(k)\right)$, where k is the number of sub-concepts per concept, $\ell_{\max}$ is the maximum hierarchical depth, η is the learning rate, and ϵ describes the amount of noisiness tolerated. An interesting feature of these algorithms is that they allow the networks to learn sub-concepts in a highly interleaved manner. We complement our learning results with lower bounds, which say that, in order to recognize concepts with hierarchical depth ℓ with noise-tolerance, a neural network must have at least ℓ layers.

The results in this paper represent a first step in studying learning of hierarchical concepts using SNNs. The case studied here is basic, but the results suggest many directions for extensions to more elaborate and realistic cases.

Contents

*Some of the work was conducted in affiliation with CSAIL, MIT.

1 Introduction

We are interested in the general question of *how concepts that have structure are represented in the brain*. What do these representations look like? How are they learned, and how do the concepts get recognized after they are learned? We draw inspiration from intriguing research in experimental computer vision on "network dissection" by Zhou, et al. [21] showing how deep convolutional neural networks learn structure using unsupervised learning of visual concepts: the lower layers of the network learn very basic concepts and the higher layers learn higher-level concepts. Our general thesis is that the structure that is naturally present in the concepts gets mirrored in its brain representation, in some natural way that facilitates both learning and recognition. This appears to be consistent with neuroscience research, which indicates that visual processing in mammalian brains is performed in a hierarchical way, starting from primitive notions such as position, light level, etc., and building toward complex objects; see, e.g., [8, 7, 4].

We approach this problem using ideas and techniques from theoretical computer science, distributed computing theory, and in particular, from recent work by Lynch, Musco, Parter, and others on synchronous Spiking Neural Networks (SNNs) [13, 12, 19, 6], These papers began the development of an algorithmic theory of SNNs, developing a formal foundations, and using them to study problems of attention, focus, and neural coding. Here we continue that general development, by initiating the study of learning within the same framework. We focus on learning hierarchically-structured concepts.

Specifically, we focus on learning of *concept hierarchies*, in which concepts are built from other lower-level concepts, which in turn are built from other still-lower-level concepts, etc. Such structure is natural, for example, for physical objects that are learned and recognized during human or computer visual processing. An example of such a hierarchy might be the following model of a *human*: A human consists of a *body*, a *head*, a *left leg*, a *right leg*, a *left arm*, and a *right arm*. Each of these concepts consists in turn consists of concepts; e.g., the head consists of two *eyes*, a *nose*, a *mouth*, etc. Again, each of these concepts may consist of more concepts, allowing us to model a human to an arbitrary degree of granularity. Most concepts in real life have additional structure, e.g., arms and legs are positioned symmetrically; we ignore such additional information here and simply assume that each concept consists of sub-concepts.

For this initial theoretical study, we make some simplifications: we fix a maximum level $\ell_{\max}$ for our concept hierarchies, we assume that all non-primitive concepts have the same number k of "child concepts", and we assume that our concept hierarchies are trees, i.e., there is no overlap in the composition of different concepts at the same level of a hierarchy. We expect that all of these assumptions can be removed or weakened, but we think it is useful to start with the simpler case.

This paper is intended to demonstrate theoretically, at least for this special case, how hierarchically structured data can be represented, learned, and recognized in biologically plausible layered SNNs. To that end, we provide general definitions for *concept hierarchies* and *layered neural networks*. We define precisely what it means for a layered neural network to *recognize* a particular

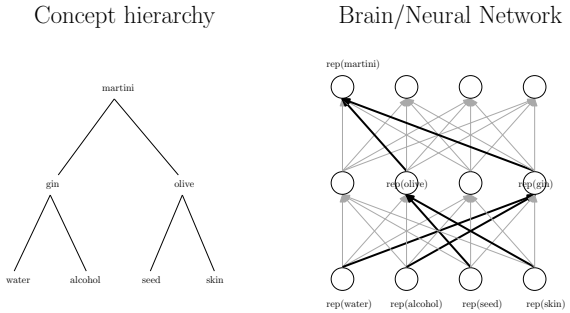

Figure 1: The figure shows the concept *martini*, which consists of two concepts, etc. The right-hand side shows a network that has "learned" the concept "martini" in the sense that, when the neurons representing the basic parts *water, alcohol, seed, skin* are excited, then exactly one neuron on the top layer will fire. This neuron should also fire when "most" of the basic parts are excited, and it should not fire when few of the basic parts are excited. The network accomplishes this by strengthening synapses (bold edges) and weakening others (thin edges).

concept hierarchy; our notion of recognition is robust, in that it tolerates a bounded amount of noise. We also define what it means for a layered neural network to *learn to recognize* a concept hierarchy, according to our noise-tolerant definition of recognition.

Next, we present two simple layered networks that can learn efficiently to recognize concept hierarchies; the first assumes reliability during the learning process, whereas the second tolerates some random noise. An example of such learning is shown in Figure 1. We also provide lower bounds, showing that, in order to recognize concepts with hierarchical depth ℓ with noise-tolerance, a neural network must have at least ℓ layers. We end with many directions for extending this work.

In more detail: We assume a fixed number $\ell_{\max}$ of levels in our concept hierarchies. Each concept hierarchy $\mathcal{C}$ has a fixed set C of concepts, organized into levels ℓ, $0 \leq \ell \leq \ell_{\max}$. These are chosen

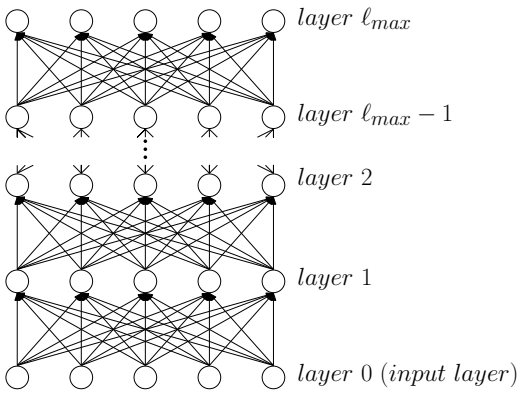

Figure 2: The figure depicts the general structure of the feed-forward network.

from some universal sets D_ℓ, $0 \leq \ell \leq \ell_{\max}$ of *concepts*. Each concept at each level ℓ, $1 \leq \ell \leq \ell_{\max}$ has precisely k child level $\ell - 1$ concepts. We assume that each concept hierarchy is a tree, that is, there is no overlap among the sets of children of different concepts. Each individual concept hierarchy represents the concepts and child relationships that arise in a particular execution of the network (or lifetime of an organism). The chosen concepts and their relationships may be different in different concept hierarchies.

We then define a synchronous Spiking Neural Network model, similar to the model in [13, 12], but with additional structure to support learning. Most importantly, it incorporates edge weights (representing synapse strengths) into the neuron states; this provides a convenient formal way to describe how those weights change during learning. We model learning using *Oja's rule*, a biologically inspired rule that was first introduced in [16] and has since received considerable attention due its connections with dimensionality reduction; see e.g., [17, 5]. Oja's rule is a mathematical formalization of Hebbian learning [10].

Even though there is no direct experimental evidence yet that Oja's precise rule is used in the brain, its core characteristics such as long-term potentiation, long-term depression, and normalization are known to occur in brain networks, and have been studied thoroughly (e.g., [2, 1]).

Interestingly, to the best of our knowledge, Oja's rule has so far been studied only in "flat" settings, where the network has only one layer. Moreover, so far the literature on learning (e.g., [16]) has assumed that the learning parameter η is time-dependent, in proving convergence. In this paper, we study the hierarchical (multilayer) setting, and we show convergence with a fixed learning rate.

We next define what it means for an SNN of our type to correctly recognize a concept hierarchy, including both positive and negative requirements, that is, situations in which the network is required to recognize the concepts and situations where it is required not to do so. These conditions include noise-tolerance requirements: Not all of the children of a concept c need to be recognized in order for c to be recognized—a sufficiently large fraction is enough. On the other hand, if too few children of c are recognized, then c should not be recognized. We also define what it means for an SNN to correctly learn to recognize a concept hierarchy.

Then we present algorithms that allow a network, starting from a default configuration, to recognize and to learn the concepts in a particular concept hierarchy. Our algorithms are efficient, in terms of network size and running time. Namely, a network with max layer $\ell_{\max}$ suffices to recognize a concept hierarchy with max level $\ell_{\max}$. For recognition, we get extremely short recognition time, corresponding to the number of layers in the network. For learning, we obtain fairly short convergence time and large stability time. Our time bound results for learning appear in Theorem 5.3 and Theorem 6.1. Our results require the examples to be shown several times and in a constrained order: roughly speaking, we require the network to "learn" the children of a concept c first, before examples of c are shown; Thus, in our running example, we require enough examples of "head", "body", etc. to be able to learn the concepts before the network sees them all together as "human". Except for this constraint, concepts may be shown in an arbitrarily interleaved manner.

In Theorem 6.1, we consider "noisy learning", where the examples we see are perturbed by noise. This requires the network to see more examples in comparison to the noise-free case (Theorem 5.3). The learning process requires multiple examples to be shown to the network (as inputs).

Once we see that a network with max layer $\ell_{\max}$ can easily learn and recognize any concept hierarchy with max level $\ell_{\max}$, it is natural to ask whether $\ell_{\max}$ layers are actually necessary. Certainly they yield a natural and efficient representation. But it is interesting to ask the theoretical question of whether shallower networks could accomplish the same thing. For this, we give some lower bound results. These results use some simple assumptions about how data is represented. First, in Theorem 7.1, we show that a two-layer concept hierarchy requires a two-layer network in order to solve the recognition problem. Then we show (Theorem 7.2) that a three-layer concept hierarchy

requires a three-layer network, and (Theorem 7.3) that an $\ell_{\max}$-level concept hierarchy requires an $\ell_{\max}$-layer network. The assumptions describe the way data is represented in the networks, namely, that only one neuron is used to represent any particular concept, and that no neuron is used to represent a mixture of concepts. These assumptions match what arises 'naturally' from our learning algorithms.

This paper is intended to show, using theoretical techniques, how structured concepts can be represented, learned, and recognized in a biologically plausible neural network. We give fundamental definitions, algorithms, and lower bounds, for particular types of concept hierarchies and networks. This represents a first step towards a theory of learning for hierarchically structured concepts in SNNs; it opens up a huge number of follow-on questions, which we discuss in Section 8.

Related work: Some of the inspiration from this work comes from intriguing new experimental computer vision research on "network dissection" by Zhou, et al. [21], which was, in turn, inspired by neuroscience vision research such as that by Quiroga, et al [18]. The authors of [21] describes experiments that show that unsupervised learning of visual concepts in deep convolutional neural networks results in "disentangled" representations. These include neural representations, not just for the main concepts of interest, but also for their components and sub-components, etc., throughout a concept hierarchy. As in this paper, they consider individual neurons as representations for individual concepts. They find that the representations that arise are generally arranged in layers so that more primitive concepts (colors, textures,...) appear at lower layers whereas more complex concepts (parts, objects, scenes) appear at higher layers. They find that deeper networks have higher capacity to represent concepts with larger visual complexity, and that wider networks can increase the number of represented concepts.

The Quiroga paper [18] is example of a neuroscience paper that explores experimentally the notion that individual neurons in the brain act as "concept cells", representing individual visual concepts. Their focus was on higher-level concepts, such as pictures of famous individuals, and representations by neurons in the medial temporal lobe (MTL).

In the neuroscience vision research community, researchers generally agree that visual processing in mammalian brains is performed in a hierarchical way, starting from primitive notions such as position, light level, etc., and building toward complex objects; see, e.g., [8, 7, 4]. Some of this work indicates that the network includes feedback edges in addition to forward edges; the function of the feedback edges seems to be to solidify representations of lower-level objects based on context [9, 14]. While we do not address feedback edges in this paper, that is one of our intended future directions. The learning rule we study, Oja's rule, was introduced by [16] and is also used for dimensionality reduction; see e.g., [17, 5]. As mentioned earlier, to the best of our knowledge, Oja's rule has so far only been studied in "flat" settings with only one-layered networks and with time-dependent learning rates ([16, 17, 5].

Work by Mhaskar et al. [15] is related to ours in that they also consider embedding a tree-structured concept hierarchy in a layered network. They also prove results saying that deep neural networks are better than shallow networks at representing a deep concept hierarchy, However, their concept hierarchies are different mathematically from ours, formalized as compositional functions. Also, their notion of representation corresponds to function approximation, and their proofs are based on approximation theory, rather than the limitations or noise-tolerance in recognizing hierarchical concepts. Other results along the same lines appear in [20].

There is also an interesting connection to circuit complexity (e.g., [11]) with respect to the question of how many layers are required to solve the recognition problem (Section 4.1). The models studied are slightly different as neurons have the power of threshold gates.

Nonetheless, understanding the trade-off between the number of layers and the number of neurons

per layer would be a very interesting question for future work.

Acknowledgments: We thank Brabeeba Wang for helpful conversations and suggestions.

2 Data Model

We define some general parameters and notation, and then define the notion of a "concept hierarchy". A concept hierarchy is supposed to represent all the concepts that arise in some particular "lifetime" of an organism, together with hierarchical relationships between them. We follow this with notions of "support" that say which lowest-level concepts are sufficient to support the recognition of higher-level concepts.

Preliminaries We start by fixing some constants: $\ell_{\max}$, a positive integer, representing the maximum level number for the concepts we consider. n, a positive integer, representing the total number of lowest-level concepts. k, a positive integer, representing the number of sub-concepts for each concept that is not at the lowest level, in any concept hierarchy. r_1, r_2, reals in $[0,1]$ with $r_1 \leq r_2$; these represent thresholds for noisy recognition. We assume a universal set D of *concepts*, partitioned into disjoint sets $D_\ell, 0 \leq \ell \leq \ell_{\max}$. We refer to any particular concept $c \in D_\ell$ as a *level ℓ concept*, and write $level(c) = \ell$. Here, D_0 represents the most basic concepts and $D_{\ell_{\max}}$ the highest-level concepts. We assume that $|D_0| = n$.

2.1 Concept hierarchies

A *concept hierarchy* $\mathcal{C}$ consists of a subset C of D, together with a *children* function. For each ℓ, $0 \leq \ell \leq \ell_{\max}$, we define C_ℓ to be $C \cap D_\ell$, that is, the set of level ℓ concepts in $\mathcal{C}$.

For each concept $c \in C_\ell$, $1 \leq \ell \leq \ell_{max}$, we designate a nonempty set $children(c) \subseteq C_{\ell-1}$. We call each $c' \in children(c)$ a *child* of c. We require the following three properties. First, $|C_{\ell_{\max}}| = k$. Second, For any $c \in C_\ell$, where $1 \leq \ell \leq \ell_{\max}$, we have that $|children(c)| = k$; that is, the degree of any internal node in the concept hierarchy is exactly k. Finally, For any two distinct concepts c and c' in C_ℓ, where $1 \leq \ell \leq \ell_{\max}$, we have that $children(c) \cap children(c') = \emptyset$; that is, the sets of children of different concepts at the same level are disjoint.

It follows that $\mathcal{C}$ is a forest with k roots, and that it has height $\ell_{\max}$. Also, for any $\ell, 1 \leq \ell \leq \ell_{\max}$, $|C_\ell| = k^{\ell_{\max} - \ell + 1}$. We extend the *children* notation recursively, namely, we define concept c' to be a *descendant* of a concept c if either $c' = c$, or c' is a child of a descendant of c. We write $descendants(c)$ for the set of descendants of c. Let $leaves(c) = descendants(c) \cap C_0$, that is, all the level 0 descendants of c. Note that our notion of concept hierarchies is quite restrictive, in that we allow no overlap between the sets of children of different concepts. Allowing overlap is an interesting research direction itself.

2.2 Support

In this subsection, we fix a particular concept hierarchy $\mathcal{C}$, with its concept set C. For any given subset B of the general set D_0 of level 0 concepts, and any real number $r \in [0,1]$, we define a set $supported_r(B)$ of concepts in C. This represents the set of concepts $c \in C$, at any level, that have enough of their leaves present in B to support recognition of c. The notion of "enough" here is defined recursively, based on having an r-fraction of children present at every level.

Definition 2.1 (Support)**.** *Given $B \subseteq D_0$, define the following sets of concepts at all levels, recursively: $B_0 = B \cap C_0$. B_1 is the set of all concepts $c \in C_1$ such that $|children(c) \cap B_0| \geq rk$. In*

general, for $1 \leq \ell \leq \ell_{max}$, B_ℓ *is the set of all concepts* $c \in C_\ell$ *such that* $|children(c) \cap B_{\ell-1}| \geq rk$. *Define* $supported_r(B)$ *to be* $\bigcup_{0 \leq \ell \leq \ell_{\max}} B_\ell$. *We sometimes also write* $supported_r(B, \ell)$ *for* B_ℓ.

We now give an example. Consider Figure 1. If $B = \{\text{seed, skin,water}\}$ and $r = 1$, then $supported_1(B) = \{olive\}$. The special case $r = 1$ is important as it corresponds to a "noise-free" notion of support, in which all the leaves of a concept have to be present:

Lemma 2.2. *For any* $B \subseteq D_0$, $supported_1(B)$ *is the set of all concepts* $c \in C$ *(at all levels) such that* $leaves(c) \subseteq B$.

3 Network Model

We first describe the network structure, then the individual neurons, and finally the operation of the overall network. Before we do so, we introduce four constants: ℓ'_{max}, a positive integer, representing the maximum number of a layer in the network. n, a positive integer, representing the number of distinct inputs the network can handle; this is the same n as in the data model. τ, a real number, representing the firing threshold for neurons. η, a positive real, representing the learning rate.

3.1 Network structure

Our network $\mathcal{N}$ consists of a set N of neurons, partitioned into disjoint sets $N_\ell, 0 \leq \ell \leq \ell'_{max}$, which we call *layers*. We assume that each layer contains exactly n neurons, i.e., $|N_\ell| = n$ for all ℓ. We refer to the n neurons in layer 0 as *input neurons* and to all other neurons as *non-input neurons*. We assume total connectivity between successive layers, i.e., each neuron in N_ℓ, $0 \leq \ell \leq \ell'_{max} - 1$ has an outgoing edge to each neuron in $N_{\ell+1}$, and these are the only edges.

We assume a one-to-one mapping $rep : D_0 \to N_0$, where $rep(c)$ is the neuron corresponding to concept c. That is, rep is a one-to-one mapping from the full set of level 0 concepts D_0, to N_0, the set of layer 0 neurons, This will allow the network to receive an input corresponding to any level 0 concept. See Figure 2 for a depiction.

We "lift" the definition of rep to sets of level 0 concepts as follows: For any $B \subseteq D_0$, define $rep(B) = \{rep(b) | b \in B\}$. That is, $rep(B)$ is the set of all *reps* of concepts in B. We use analogous "lifting" definitions to extend other functions to sets. Since we know that $|C_0| = k^{\ell_{\max}+1}$, $C_0 \subseteq D_0$, and all elements of D_0 have *reps* among the n neurons of N_0, it follows that $n \geq k^{\ell_{\max}+1}$.

3.2 Neuron states

We distinguish between input and non-input neurons. Each input neuron $u \in N_0$ has just one state component: $firing, with values in \{0,1\}; this indicates whether or not the neuron is firing$. We denote the *firing* component of neuron u at time t by $firing^u(t)$; we sometimes abbreviate this in math formulas as just $y^u(t)$. Each non-input neuron $u \in N_\ell$, $1 \leq \ell \leq \ell'_{max}$, has three state components:

- *firing*, with values in $\{0, 1\}$, indicating whether the neuron is firing.
- *weight*, a real-valued vector in $[0, 1]^n$ representing current weights on incoming edges.
- *engaged*, with values in $\{0, 1\}$; indicating whether the neuron is currently prepared to learn.

We denote the three components of neuron u at time t by $firing^u(t)$, $weight^u(t)$, and $engaged^u(t)$, respectively, and abbreviate these by $y^u(t)$, $w^u(t)$, and $e^u(t)$. We also use the notation $x^u(t)$ to denote the column vector of *firing* flags of u's incoming neighbor neurons at time t.

That is, $x^u(t) = \begin{bmatrix} y^{v_1}(t) \\ y^{v_2}(t) \\ \vdots \\ y^{v_n}(t) \end{bmatrix}$, where $\{v_i\}_{i \leq n}$ are the incoming neighbors of u, which are exactly all the nodes in the layer below u.

3.3 Neuron transitions

Now we describe neuron behavior, specifically, we describe how to determine the values of the state components of each neuron u at time $t \geq 1$ based on values of state components at the previous time $t-1$ and on external inputs.

Input neurons: If u is an input neuron, then it has only one state component, the *firing* flag. We assume that the value of the *firing* flag is controlled by the network's environment and not by the network itself, i.e., the value of $y^u(t)$ is set by some external input signal.

Non-input neurons: If u is a non-input neuron, then it has three state components, *firing*, *weight*, and *engaged*. Whether or not neuron u fires at time t, that is, the value of $y^u(t)$, is determined by its *potential* and its *activation function*. The potential at time t, which we denote as $pot^u(t)$ is given by the dot product of the weights and inputs at neuron u at time $t-1$, that is, $pot^u(t) = w^u(t-1)^T \cdot x^u(t-1) = \sum_{j=1}^{n} w_j^u(t-1)x_j^u(t-1)$. The activation function, which defines whether or not neuron u fires at time t, is then defined by: $y^u(t) = \begin{cases} 1 & \text{if } pot^u(t) \geq \tau, \\ 0 & \text{otherwise} \end{cases}$. Here, τ is the assumed firing threshold. We assume that the value of the *engaged* flag of u is controlled by u's environment, that is, for every t, the value of $e^u(t)$ is set by some input signal, which may arise from the environment or from another part of the network.

Finally, for the weights, we assume that each neuron that is engaged at time t determines its weights at time t according to Oja's learning rule. That is, if $e^u(t) = 1$, then

$$\textit{Oja's rule: } w^u(t) = w^u(t-1) + \eta z(t-1) \cdot (x^u(t-1) - z(t-1) \cdot w^u(t-1)), \tag{1}$$

where η is the assumed learning rate. and $z(t-1) = pot^u(t)$. For convenience, we will sometimes drop the neuron superscripts and time arguments, abbreviate $w^u(t-1)$ by w, $w^u(t)$ by w', etc. and write simply $w' = w + \eta z \cdot (x - z \cdot w)$.

3.4 Network operation

During execution, the network proceeds through a series of *configurations* $C(0), C(1), \ldots$, where $C(t)$ describes the configuration at time t. Each configuration specifies a state for every neuron in the network, that is, values for all the state components of all the neurons.

As described above, the y values for the input neurons are specified by some external source. The y, w, and e values for the non-input neurons are defined by the network specification at time $t = 0$. For times $t > 0$, the y and w values are determined by the activation and learning functions described above. The e values will be determined by special inputs arriving from other subnetworks, which we describe later.

4 Problem Statements

In this section we define two problems: *recognizing concept hierarchies*, and *learning to recognize concept hierarchies*. In both cases, we assume that each item is represented by exactly one neuron.

Throughout this section, we fix a concept hierarchy $\mathcal{C}$, with concept set C and maximum level $\ell_{\max}$, partitioned into $C_0, C_1, \ldots, C_{\ell_{\max}}$. We fix constants n, k, r_1 and r_2 according to the definitions for a concept hierarchy. We also fix a network $\mathcal{N}$, with constants ℓ'_{max}, n, τ, and η as in the definitions for a network. Thus, the maximum layer number ℓ'_{max} for $\mathcal{N}$ may be different from the maximum level number $\ell_{\max}$ for $\mathcal{C}$, but the number n of input neurons is the same as the number of level 0 items in $\mathcal{C}$. The following definition will be useful in defining the recognition and learning problems. It defines what it means for a particular subset B of the level 0 concepts to be "presented" as input to the network, at a certain time t.

Definition 4.1. Presented: *If $B \subseteq D_0$ and t is a non-negative integer, then we say that B is* presented at time t *(in some particular execution) if, for every layer 0 neuron u, the following hold. If $u \in rep(B)$ then $y^u(t) = 1$. Furthermore, if $u \notin rep(B)$ then $y^u(t) = 0$. That is, all of the layer 0 neurons in $rep(B)$ fire at time t and no other layer 0 neuron fires at that time.*

4.1 Recognition

Here we define what it means for network $\mathcal{N}$ to recognize concept hierarchy $\mathcal{C}$. We assume that every concept $c \in C$, at every level, has a unique representing neuron $rep(c)$.

We also assume that, during the entire recognition process, the *engaged* flags of all neurons are off, i.e., for every neuron u at a layer > 0, $e^u(t) = 0$.

The following definition uses the two "ratio" values $r_1, r_2 \in [0, 1]$, with $r_1 \leq r_2$. Ratio r_2 represents the fraction of children of a concept c at any level that should be sufficient to support firing of $rep(c)$. r_1 is a fraction below which $rep(c)$ should not fire.

Definition 4.2. Recognition problem: *Network $\mathcal{N}$* (r_1, r_2)-recognizes *a concept c in a concept hierarchy $\mathcal{C}$ provided that $\mathcal{N}$ contains a unique neuron $rep(c)$ such that the following holds. Assume that $B \subseteq C_0$ is presented at time t. Then:*

1. When $rep(c)$ must fire: *If $c \in supported_{r_2}(B)$, then $rep(c)$ fires at time $t + layer(rep(c))$.*
2. When $rep(c)$ must not fire: *If $c \notin supported_{r_1}(B)$, then $rep(c)$ does not fire at time $t + layer(rep(c))$.*

We say that $\mathcal{N}$ (r_1, r_2)-recognizes $\mathcal{C}$ *provided that it* (r_1, r_2)-recognizes *each concept c in $\mathcal{C}$.*

The special case of $(1, 1)$-recognition is interesting and summarized in the following lemma. The proof follows from the definitions and Lemma 2.2.

Lemma 4.3. *Network $\mathcal{N}$ $(1, 1)$-recognizes a concept c in a concept hierarchy $\mathcal{C}$ if and only if $\mathcal{N}$ contains a unique neuron $rep(c)$ such that the following holds. If $B \subseteq C_0$ is presented at time t, then $rep(c)$ fires at time $t + layer(rep(c))$ if and only if $leaves(c) \subseteq B$.*

4.2 Learning

In the learning problem, the network does not know ahead of time which particular concept hierarchy might be presented in a particular execution. It must be capable of learning *any* concept hierarchy. In order for the network to learn a concept hierarchy $\mathcal{C}$, it must receive inputs corresponding to all the concepts in C. We describe below how individual concepts are "shown" to the network, and then give constraints on the order in which the concepts are shown. Then we state the learning guarantees, assuming an allowable presentation schedule for $\mathcal{C}$.

We begin by describing how an individual concept c is "shown" to the network, in order to help in learning c. This involves identifying a subset B of D_0 which should be presented to the network. Here we define B to be simply $leaves(c)$ $(= descendants(c) \cap C_0)$.

Definition 4.4 ("Showing" of concepts)**.** *Concept c is* shown *when, for every input neuron u, u fires if and only if $u \in rep(leaves(c))$.*

Learning a concept hierarchy will involve showing all the concepts in the hierarchy to the network. Informally speaking, we assume that the concepts are shown "bottom-up", e.g., before learning the concept of a head, the network learns the lower-level concepts of mouth, eye, etc. And before learning the concept of a human, the network first learns the lower-level concepts of head, body, legs, etc. More precisely, to enable network $\mathcal{N}$ to learn the concept hierarchy $\mathcal{C}$, we assume that every concept in its concept set C is shown to the network at least σ times (where σ is a parameter to be specified by a learning algorithm). Furthermore, we assume that any concept $c \in C$ is shown only after each child of c has been shown at least σ times. We allow the concepts to be shown in an arbitrary order and in an interleaved manner provided that theses constraints are observed.

Definition 4.5. Bottom-up presentation schedule: *A* presentation schedule *for $\mathcal{C}$ is any finite list $c_0, c_1, \ldots, c_m$ of concepts in C, possibly with repeats. A presentation is* σ-bottom-up*, where σ is a positive integer, provided that each concept in C appears in the list at least σ times, and no concept in C appears before each of its children has appeared at least σ times.*

A presentation schedule $c_0, c_1, \ldots, c_m$ generates a corresponding sequence $B_0, B_1, \ldots, B_m$ of sets of level 0 concepts to be presented in a learning algorithm. Namely, B_i is defined to be $rep(leaves(c_i))$.

Definition 4.6. Learning problem: *Network $\mathcal{N}$* (r_1, r_2)-learns *concept hierarchy $\mathcal{C}$* with σ repeats *provided that the following holds. After a training phase in which $\mathcal{C}$ is shown to the network according to a σ-bottom-up presentation schedule, network $\mathcal{N}$ (r_1, r_2)-recognizes $\mathcal{C}$.*

5 Algorithms (Neural Networks) for Recognition and Noise-Free Learning

Fix a concept hierarchy $\mathcal{C}$ with concept set C. Recognition can be achieved by simply embedding the digraph induced by $\mathcal{C}$ in the network $\mathcal{N}$. See Figure 1 for an illustration. For every ℓ and for every level ℓ concept c of $\mathcal{C}$, we designate a unique representative $rep(c)$ in layer ℓ of the network. Let R be the set of all representatives, that is, $R = rep(C) = \{rep(c) \mid c \in C\}$. We use rep^{-1} with support R to denote the corresponding inverse function that gives, for every $u \in R$, the unique concept c with $rep(c) = u$. We define the edge weight $weight(u, v)$ from a neuron u at layer ℓ to a neuron v at layer $\ell + 1$ to be $weight(u, v) = \begin{cases} 1 & \text{if } rep^{-1}(v) \in children(rep^{-1}(u)), \\ 0 & \text{otherwise.} \end{cases}$. Finally, for any particular choice of $r_1, r_2 \in [0, 1]$, with $r_1 \leq r_2$, we set the threshold for every non-input neuron to be $\tau = \frac{(r_1+r_2)k}{2}$. It should be clear that the resulting network $\mathcal{N}$ solves the (r_1, r_2)-recognition problem.

Theorem 5.1. *Network $\mathcal{N}$* (r_1, r_2)-recognizes *$\mathcal{C}$. Moreover, the required time is $\ell_{\max}$.*

In the remainder of this section we move from the recognition problem to the harder problem of learning. We start with the following module which we require.

Winner-Take-All modules: Our algorithm will use Winner-Take-All modules to select certain neurons to be prepared to learn. We abstract from these modules by describing how the *engaged* flags should be set during learning; we give the precise requirements in Assumption 5.2.

While the network is being trained, example concepts are "shown" to the network, one example at each time t, according to a σ-bottom-up schedule as defined above. We assume that, for every

example concept c that is shown, exactly one neuron at the appropriate layer will be engaged; this layer is the one with the same number as the level of c in the concept hierarchy. Furthermore, the neuron on that layer that is engaged is the one that has the largest potential pot^u. More precisely, in terms of timing, we assume:

Assumption 5.2 (Winner-Take-All Assumption)**.** *If a level ℓ concept c is "shown" at time t, then at time $t+\ell$, exactly one neuron u in layer ℓ has its engaged state component equal to 1, that is, it has $e^u(t+\ell)=1$. Moreover, u is chosen so that $pot^u(t+\ell)$ is the highest potential at time $t+\ell$ among all the layer ℓ neurons.*

Main result: We assume that the network starts in a clean state in which, for every neuron u in layer 1 or higher, $w^u(0)=\frac{1}{k^{\ell_{\max}}}\mathbf{1}$, where $\mathbf{1}$ is the n-dimensional all-one vector. We set the threshold τ for all neurons to be $\tau=\frac{(r_1+r_2)\sqrt{k}}{2}$. The initial condition and threshold, Assumption 5.2, and the general model conventions for activation and learning suffice to determine how the network behaves. Our main result is:

Theorem 5.3 ((r_1,r_2)-Learning)**.** *Let $\mathcal{N}$ be the network described above, with maximum layer ℓ'_{max}, and with learning rate $\eta=\frac{1}{4k}$. Let r_1,r_2 be reals in $[0,1]$ with $r_1\le r_2$. Let $\epsilon=\frac{r_2-r_1}{r_1+r_2}$. Let $\mathcal{C}$ be any concept hierarchy, with maximum level $\ell_{\max}\le\ell'_{max}$. Assume that the concepts in $\mathcal{C}$ are presented according to a σ-bottom-up presentation schedule as defined in Section 4.2, where σ is $O\left(\frac{1}{\eta k}\left(\ell_{\max}\log(k)+\frac{1}{\epsilon}\right)+b\log(k)\right)$. Then, $\mathcal{N}$ (r_1,r_2)-learns $\mathcal{C}$.*

A rigorous analysis can be found in Appendix A; the main idea of the analysis is as follows. We first identify structural properties of Oja's rule (Lemma A.1, Lemma A.2 and Lemma A.3), in which we quantify the weight changes of a single concept provided that all sub-concepts have been learned already. Among other properties, we show that the weights quickly to converge to either $1/\sqrt{k}$ or 0 depending on whether the weights belong to neurons that represent the sub-concepts or not.

We then use the aforementioned properties to describe in Lemma A.6 the learning (and weight changes) throughout the network. What makes this challenging is that we allow concept to be shown in an interleaved manner in addition to allowing concepts of different levels to be shown (as long as sub-concepts are shown before the concept is shown). In order to prove that all concepts are learned correctly despite these challenges, we use an involved yet elegant induction. The inductive hypothesis has five different parts that hold at every time step. Finally, in Section A.3 we put everything together and show that the network successfully (r_1,r_2)-learns the concept hierarchy.

6 Extension: Noisy Learning

We extend our model, algorithm, and analysis to noisy learning. The idea is that we should be able to learn concepts even if we do not see all the child concepts all the time. For example, we could expect to learn the concept of a human even if we sometimes only see "legs" and a "body" and other times we see the "head" and "legs" etc.

6.1 Noisy Learning Algorithm

Formally, our model is as follows. Recall that in Definition 4.4, we assumed that when a concept c is shown, that *all reps* of the leaves of c fire. We now weaken this assumption, as follows.

We redefine the notion of *showing* concept c, by first executing the procedure $mark(c,k,p)$ to determine a subset of $leaves(c)$, which is defined as follows. Fix an arbitrary $p\in(0,1]$.

If c is a level 0 concept, then *mark* c. Otherwise, select $\lceil p\cdot k\rceil$ children of c uniformly at random and recurse on these children, i.e., execute $mark(c',k,p)$ for each of the selected children c'.

Then we define B to be $\{c' \in leaves(c) \mid c' \text{ is marked}\}$. Thus, concept c is shown when, for every input neuron u, we have that u fires if and only if $u \in \{rep(c') \mid c' \in leaves(c) \text{ and } c' \text{ is marked}\}$. We call this *noisy showing*. Our result is as follows.

In the following we state our main theorem in the noisy learning setting. The main difference to Theorem 5.3 is that we only guarantee for the first $O(n^6)$ rounds of learning that the weights are 'correct' meaning that the network (r_1, r_2)-learns the concept hierarchy. This is natural since if we consider a number of rounds T that is of order exponential in n, then at some point $t \leq T$ it is very likely that the weights will be unfavorable for recognition. [1]

Theorem 6.1 ((r_1, r_2)-Noisy-Learning)**.** *Let $\mathcal{N}$ be the network described in Section 3, with maximum layer ℓ'_{max}, and with learning rate $\eta \leq c_\eta \frac{r_2^6}{k^4 \log n}$, for some small enough constant c_η. Let r_1, r_2 be reals in $[0, 1]$ with $r_1 \leq r_2$. Consider the noisy learning algorithm defined in Section 6.1 with $p \in [r_2, 1]$. Let $\mathcal{C}$ be any concept hierarchy, with maximum level $\ell_{\max} \leq \ell'_{max}$. Assume that the concepts in $\mathcal{C}$ are presented according to a σ-bottom-up presentation schedule as defined in Section 4.2, with $\sigma = c' \frac{1}{\eta k} \left(\ell_{\max} \log(k) + \frac{r_2 k + 1 - r_2}{\eta r_2^{3/2}(r_2 - r_1)} \right) + \log(k)$, for some large enough constant c'.*

Then, w.h.p., $\mathcal{N}$ (r_1, r_2)-learns $\mathcal{C}$ for at least $O(n^6)$ rounds.

6.2 Proof idea

We extend our model, algorithm, and analysis to noisy learning. The idea is that we should be able to learn concepts even if we do not see all the child concepts all the time. For example, we could expect to learn the concept of a "human" even if we sometimes only see "legs" and a "body" and other times we see the "head" and "legs" etc. Informally speaking, we assume that when a concept is shown, a random p-fraction of the sub-concepts are shown.

In contrast, in the noise-free case, all sub-concepts are shown. See Section 6 for a precise definition of the model.

In the presence of noise, many of the properties of the noise-free case no longer hold, rendering the proof significantly more involved. Here we give a rough outline of our proof. First, we bound the change of potential during a period of T rounds. We then derive very rough bounds on the change of a single weight during such a period. Using these rough results, we are able to prove much more precise bounds on the change of the weights in a given interval of length T. It turns out that the way the weights change depends highly on the other weights, which makes the analysis non-trivial. The way we show that weights converge, is by using the following potential function ψ. Fix an arbitrary time t and let $w_{min}(t)$ and $w_{max}(t)$ be the minimum and maximum weights among $w_1(t), w_k(t), \ldots, w_k(t)$, respectively. Let $\psi(t') = \max\left\{\frac{w_{max}(t')}{\bar{w}}, \frac{\bar{w}}{w_{min}(t')}\right\}$, where $\bar{w} = \frac{1}{\sqrt{pk+1-p}}$. We show that, in contrast to the noise-free case, weights belonging to representatives of sub-concepts converge to $\bar{w}$ instead to $1/\sqrt{k}$.

Our goal is to show that the above potential decreases quickly until it is very close to 1. Showing that the potential decreases is involved, since one cannot simply use a worst case approach, due to the terms in Oja's rule being non-linear and potentially having a high variance, depending on the distribution of weights. The key to showing that ψ decreases is to carefully use the randomness over the input vector and to carefully bound the non-linear terms. Bounding these non-linear terms tightly presents a major challenge which we overcome using two techniques. First, we consider a

[1] This can happen since in such a large time frame, it's very likely that there will be a long sequence of runs in which the same representatives are simply (due to bad luck) not shown. The network will forget about their importance. This is also partly the reason why the learning rate in the following theorem is smaller than the one of the noise-free counterpart: the smaller learning rate guarantees that during the first $O(n^6)$ rounds no unlikely sequence occurs that is very 'bad'.

process P' which is almost as the original process P, with the difference that the weights only change marginally in each period of T rounds. If they change by more, then we assume that the weights are simply reset to the value at the beginning of the T rounds. As we will see later, we can couple the processes P and P' with high probability. This coupling allows us to avoid a conditioning that would otherwise change the probability space and prevent us from using second technique. Second, we show that the changes of the weights form a Doob maringale allowing us to use Azuma-Hoeffding inequality to get asymptotically almost tight bounds on the change of the weights during the T rounds. We present the precise definition, results and proofs in Appendix B.

7 Results on Lower Bounds

In this section, we give three lower bound theorems describing limitations on the number of layers needed to recognize concept hierarchies with particular numbers of levels. The first theorem, Theorem 7.1, simply says that a network $\mathcal{N}$ with maximum layer 1 cannot recognize a concept hierarchy $\mathcal{C}$ with maximum level 2. This bound depends only on the requirement that $\mathcal{N}$ should recognize $\mathcal{C}$ according to our definition for noisy recognition in Definition 4.2. That definition says that the network must tolerate bounded noise, as expressed by the ratio parameters r_1 and r_2 in the definition of noisy recognition. Our result assumes reasonable constraints on the values of r_1 and r_2.

The second theorem, Theorem 7.2, extends Theorem 7.1 by showing that a network $\mathcal{N}$ with maximum layer 2 cannot recognize a concept hierarchy $\mathcal{C}$ with maximum level 3. In addition to the basic definition of noisy recognition, this result requires a new "non-interference" assumption for concept representations in the network. This assumption seems to be reasonable, in that it is guaranteed by our learning algorithms in Section 5.

The third theorem, Theorem 7.3, generalizes the first two to arbitrary numbers of levels and layers. The heart of the proof of Theorem 7.3 is a lemma, Lemma 7.4 that considers a single network $\mathcal{N}$ and a concept hierarchy $\mathcal{C}$, and assumes that $\mathcal{N}$ recognizes $\mathcal{C}$. The lemma says that, for any ℓ, the representation of any level ℓ concept must be in a layer $\geq \ell$. The proof is by induction on ℓ. As for Theorem 7.2, these results requires the new "non-interference" assumption for concept representations.

7.1 Assumptions for the lower bounds

Here we list the assumptions that we use for our lower bounds. These assumptions are for a particular concept hierarchy $\mathcal{C}$, with concept set C, to be (r_1, r_2)-recognized by a particular network $\mathcal{N}$. The first three of these properties are already required by the definition of the noisy recognition problem. These three properties are enough to prove our first lower bound theorem, Theorem 7.1.

1. Every concept $c \in C$ has a unique designated neuron $rep(c)$ in the network. (In general, it might be in any layer, regardless of the level of c.)

2. Let B be any subset of C_0. If $c \in supported_{r_2}(B)$, then presentation of B results in firing of $rep(c)$ at time $t + layer(rep(c))$.

3. Let B be any subset of C_0. If $c \notin supported_{r_1}(B)$, then presentation of B does not result in firing of $rep(c)$ at time $t + layer(rep(c))$.

The fourth property is new here, and used only for the second and third lower bound theorems, Theorem 7.2 and Theorem 7.3. This property is an attempt to rule out extraneous firing caused by mixing inputs that belong to different higher-level concepts. Stated informally: presenting the sets

of leaves of several concepts simultaneously does not trigger any additional firing at higher layers of the network than what we would get by presenting the sets one at a time.

4. *Non-interference:* Fix any $\ell \geq 1$. Consider any subset B of the level ℓ concepts in $\mathcal{C}$. For each $b \in B$, let $N(b)$ be the set of neurons (at all layers) whose firing is triggered by presenting $leaves(b)$.[2] Let N be the set of neurons (at all layers) whose firing is triggered by presenting all the leaves of all the concepts in B at the same time. Then $N = \bigcup_{b \in B} N(b)$.

Note that all of the above assumptions arise from our learning algorithms. While they are natural, it is nonetheless conceivable that there exists other learning algorithms that violate these assumptions.

Throughout this section, we assume the model presented in Section 2 and Section 3. Furthermore, since we are considering recognition only, and not learning, we assume that the *engaged* state components are always equal to 0.

Also throughout this section, we assume that r_1 and r_2 satisfy the following constraints:

1. $0 \leq r_1 \leq r_2 \leq 1$.
2. $r_1 k$ is not an integer; define r_1' so that $r_1' k = \lfloor r_1 k \rfloor$.
3. Define r_2' so that $r_2' k = \lceil r_2 k \rceil$.
4. $(r_2')^2 \leq 2r_1' - (r_1')^2$.

For example, for $k = 10$, $r_1 = .51$ and $r_2 = .8$ satisfy these conditions.

7.2 Impossibility for recognition for two levels and one layer

We consider an arbitrary concept hierarchy $\mathcal{C}$ with maximum level 2 and concept set C.

We assume a (static) network $\mathcal{N}$ with maximum layer 1, and total connectivity from the layer 0 neurons to the layer 1 neurons. For such a network and concept hierarchy, we get a contradiction to the noisy recognition problem definition from Section 4.1, for any values of r_1 and r_2 that satisfy the constraints given in Section 7.1. For the problem requirements in this section, we use only Assumptions 1-3 from Section 7.1.

Theorem 7.1. *Assume that $\mathcal{C}$ has maximum level 2 and $\mathcal{N}$ has maximum layer 1. Assume that r_1, r_2, r_1', r_2' satisfy the constraints in Section 7.1. Then $\mathcal{N}$ does not recognize $\mathcal{C}$, according to Assumptions 1-3.*

The proof can be found in Appendix C

7.3 Impossibility for recognition for three levels and two layers

Now we consider another special case result, for three levels and two layers, before moving on in the next section to the general case of $\ell_{\max}$ levels and $\ell_{\max} - 1$ layers. Now we assume an arbitrary concept hierarchy $\mathcal{C}$ with maximum level 3, and a network $\mathcal{N}$ with maximum layer 2 and total connectivity between each consecutive pair of layers. For such a network and concept hierarchy, we get a contradiction to the noisy recognition problem definition, for any values of r_1 and r_2 that satisfy the constraints given in Section 7.1. For the problem requirements here, we use Assumptions 1-4 from Section 7.1.

[2] A note about timing: When we say that the firing of a neuron u is triggered by presenting a set of level 0 concepts at time t, we mean that this firing occurs at time exactly $t + layer(u)$.

Theorem 7.2. *Assume that $\mathcal{C}$ has maximum level 3 and $\mathcal{N}$ has maximum layer 2. Assume that r_1, r_2, r_1', r_2' satisfy the constraints in Section 7.1. Then $\mathcal{N}$ does not recognize $\mathcal{C}$, according to Assumptions 1-4.*

The proof can be found in Appendix C.

7.4 Impossibility for recognition for $\ell_{\max}$ levels and $\ell_{\max} - 1$ layers

We now generalize the results of Sections 7.2 and 7.3 to arbitrary numbers of levels. Now we assume an arbitrary concept hierarchy $\mathcal{C}$ with maximum level $\ell_{\max}$, and a network $\mathcal{N}$ with maximum layer ℓ'_{max} and total connectivity between consecutive layers. Again we get a contradiction to the noisy recognition problem definition, for any values of r_1 and r_2 satisfying the constraints given in Section 7.1. For the problem requirements here, we use Assumptions 1-4 from Section 7.1.

Theorem 7.3. *Assume that $\ell'_{max} < \ell_{\max}$, that is, that the network $\mathcal{N}$ has fewer layers than the number of levels in the concept hierarchy $\mathcal{C}$. Assume that r_1, r_2, r_1', r_2' satisfy the constraints in Section 7.1. Then $\mathcal{N}$ does not recognize $\mathcal{C}$, according to Assumptions 1-4.*

The proof of Theorem 7.3 borrows ideas from that of Theorem 7.2. However, instead of a case analysis as in the earlier proof, we would like to use induction on the level numbers. For this, we extract a lemma about where the *reps* of different concepts may appear in a single network. For this lemma (unlike in the theorem), we do not assume any particular relationship between $\ell_{\max}$ and ℓ'_{max}.

Lemma 7.4. *Suppose that $\mathcal{N}$ recognizes $\mathcal{C}$, according to Assumptions 1-4. Then for every $\ell, 1 \leq \ell \leq \ell_{\max}$, and for any level ℓ concept $c \in C$, $rep(c)$ is in a layer $\geq \ell$.*

The proof can be found in Appendix C. Using Lemma 7.4, we can now prove the main theorem:

Proof of Theorem 7.3. Assume that $\ell'_{max} < \ell_{\max}$, that is, that $\mathcal{N}$ has fewer layers than the number of levels in $\mathcal{C}$. Assume for contradiction that $\mathcal{N}$ (r_1, r_2)-recognizes $\mathcal{C}$.

Fix any level $\ell_{\max}$ concept c. Then by Lemma 7.4, $rep(c)$ must be in a layer $\geq \ell_{\max}$. But this is impossible since $\mathcal{N}$ has strictly fewer than $\ell_{\max}$ layers. □

8 Conclusions and Future Work

In this paper, we have proposed a theoretical model for recognition and learning of hierarchically structured concepts in synchronous Spiking Neural Networks. Using this model, we have given algorithms and lower bound results that appear to be consistent with experimental results for learning of structured concepts. These results suggest numerous directions for future research.

Extensions to our results: We can consider different orders in which concepts in a hierarchy can be learned. Is it possible to learn higher-level concepts before learning low-level concepts? How does the order of learning affect the time required to learn? Another interesting issue is robustness of the network, e.g., to noise in calculating potentials, or to failures of neurons or synapses.

Variations in the network model: Our networks have a simple structure; it would be interesting to consider natural variations. For example, instead of all-to-all connections between consecutive layers, what happens if we assume a smaller number of randomly-determined connections between layers? Also, in our networks, all edges go from one layer ℓ to the next higher layer $\ell + 1$. What if we allow edges to go from layer ℓ to any higher layer? What if we allow feedback edges from each

layer ℓ to the next-lower layer $\ell - 1$? How does this aid in recognizing or learning concepts based on feedback from representations of higher-level concepts? What would be the effect of using other variants of Hebbian learning rules besides Oja's rule?

Learning different kinds of structure: It would be interesting to understand more generally what kinds of structures can be learned by synchronous SNNs. In our concept hierarchies, each level $\ell + 1$ concept corresponds to the "and" of several level ℓ concepts. What if we allow concepts that correspond to "ors", or even "nors", of other concepts? Similar questions were suggested by Valiant [], in terms of a different model.

In computer vision, it appears that a concept may be first recognized in outline and then details get filled in later. How can we fit this type of recognition into our model? Also, in addition to learning individual concepts, it would be interesting to learn relationships between concepts, such as association or causality. We might also consider other types of concept structure, such as sequences (as in music), or geometric or topological structure.

References

[1] Alain Artola, S Bröcher, and Wolf Singer. Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature*, 347(6288):69, 1990.

[2] Alain Artola and Wolf Singer. Long-term depression of excitatory synaptic transmission and its relationship to long-term potentiation. *Trends in neurosciences*, 16(11):480–487, 1993.

[3] Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

[4] Daniel J Felleman and DC Essen Van. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.

[5] Peter Földiák and Peter Fdilr. Adaptive network for optimal linear feature extraction. 1989.

[6] Yael Hitron, Nancy A. Lynch, Cameron Musco, and Merav Parter. Random sketching, clustering, and short-term memory in spiking neural networks. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, pages 23:1–23:31, 2020. URL: https://doi.org/10.4230/LIPIcs.ITCS.2020.23, doi:10.4230/LIPIcs.ITCS.2020.23.

[7] D. Hubel and T. Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

[8] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3):574–591, 1959. URL: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1959.sp006308, arXiv:https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1959.sp006308, doi:10.1113/jphysiol.1959.sp006308.

[9] JM Hupé, AC James, BR Payne, SG Lomber, P Girard, and J Bullier. Cortical feedback improves discrimination between figure and background by v1, v2 and v3 neurons. *Nature*, 394(6695):784, 1998.

[10] Richard Kempter, Wulfram Gerstner, and J Leo Van Hemmen. Hebbian learning and spiking neurons. *Physical Review E*, 59(4):4498, 1999.

[11] Swastik Kopparty and Srikanth Srinivasan. Certifying polynomials for ac^ 0 (parity) circuits, with applications. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[12] Nancy A. Lynch and Cameron Musco. A basic compositional model for spiking neural networks. *CoRR*, abs/1808.03884, 2018. URL: http://arxiv.org/abs/1808.03884, arXiv:1808.03884.

[13] Nancy A. Lynch, Cameron Musco, and Merav Parter. Computational tradeoffs in biological neural networks: Self-stabilizing winner-take-all networks. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 15:1–15:44, 2017. URL: https://doi.org/10.4230/LIPIcs.ITCS.2017.15, doi:10.4230/LIPIcs.ITCS.2017.15.

[14] Nikola T Markov, Julien Vezoli, Pascal Chameau, Arnaud Falchier, René Quilodran, Cyril Huissoud, Camille Lamy, Pierre Misery, Pascale Giroud, Shimon Ullman, et al. Anatomy of hierarchy: feedforward and feedback pathways in macaque visual cortex. *Journal of Comparative Neurology*, 522(1):225–259, 2014.

[15] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. Learning functions: when is deep better than shallow. *arXiv preprint arXiv:1603.00988*, 2016.

[16] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

[17] Erkki Oja. Principal components, minor components, and linear neural networks. *Neural networks*, 5(6):927–935, 1992.

[18] R. Quian Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried. Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045):1102–1107, 2005. URL: https://doi.org/10.1038/nature03687, doi:10.1038/nature03687.

[19] Lili Su, Chia-Jung Chang, and Nancy A. Lynch. Spike-based winner-take-all computation: Fundamental limits and order-optimal circuits. *CoRR*, abs/1904.10399, 2019. URL: http://arxiv.org/abs/1904.10399, arXiv:1904.10399.

[20] Matus Telgarsky. Benefits of depth in neural networks. *CoRR*, abs/1602.04485, 2016. URL: http://arxiv.org/abs/1602.04485, arXiv:1602.04485.

[21] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *CoRR*, abs/1711.05611, 2017. URL: http://arxiv.org/abs/1711.05611, arXiv:1711.05611.

A Analysis of Noise-free Learning

We present our analysis for the noise-free learning algorithm in this section. In Section A.1, we describe how incoming weights change for a particular neuron when it is presented with a consistent input vector. In Section A.2, we prove our main invariant, saying how neurons get bound to concepts, when neuron firing occurs, and how weights change, during the time when the network is learning. In Section A.3, we use that invariant to prove Theorem 5.3.

A.1 Weight Change for Individual Neurons

In this subsection we give a series of lemmas that describe how incoming weights change for a particular neuron when it is presented with a consistent input vector during execution of our noise-free learning network. Throughout this subsection, we consider a single neuron u in a layer ≥ 1.

We begin by considering how weights change in a single round. Lemma A.1 describes how the weights change for firing neighbors, and for non-firing neighbors. In this lemma, we consider a neuron u with weight vector $w(t-1)$ and input vector $x(t-1)$, both at time $t-1 \geq 0$. Write $z(t-1)$ for the dot produce of $w(t-1)$ and $x(t-1)$. We assume that the engaged component, $e(t)$, is equal to 1. We give bounds on the new weights for u at time t, given by $w(t)$.

Lemma A.1. *Let $F \subseteq \{1, \ldots, n\}$, with $|F| = k$. Assume that:*

1. *$x_i(t-1) = 1$ for every $i \in F$ and $x_i(t-1) = 0$ for every $i \notin F$. That is, exactly the incoming neighbors in F fire at time $t-1$.*
2. *All weights $w_i(t-1), i \in F$ are equal, and all weights $w_i(t-1), i \notin F$ are equal.*
3. *$w_i(t-1) < \frac{1}{\sqrt{k}}$ for every $i \in F$.*
4. *$w_i(t-1) > 0$ for every $i \notin F$.*
5. *$0 < \eta \leq \frac{1}{4k}$.*

Then:

1. *All weights $w_i(t), i \in F$ are equal, and all weights $w_i(t), i \notin F$ are equal.*
2. *For any $i \in F$, $w_i(t) > w_i(t-1)$.*
3. *For any $i \in F$, $w_i(t) < \frac{1}{\sqrt{k}}$.*
4. *For any $i \notin F$, $w_i(t) < w_i(t-1)$.*
5. *For any $i \notin F$, $w_i(t) > 0$.*

Proof. Part 1 is immediate by symmetry—all components for $i \in F$ are changed by the same rule, based on the same information.

Note that $z(t-1) < k\frac{1}{\sqrt{k}} = \sqrt{k}$, because of the assumed upper bound for each $w_j(t-1)$ and the fact that $|F| = k$. Similarly, we have that $z(t-1) > 0$.

For Part 2, consider any $i \in F$. Since $z(t-1) < \sqrt{k}$ and $w_i(t-1) \leq \frac{1}{\sqrt{k}}$, the product $z(t-1)w_i(t-1) < 1$. Then by Oja's rule:

$$w_i(t) = w_i(t-1) + \eta z(t-1)(1 - z(t-1)w_i(t-1)) > w_i(t-1) + \eta z(t-1) \cdot 0 = w_i(t-1),$$

as needed.

For Part 3, again consider any $i \in F$. Since $w_i(t-1) < \frac{1}{\sqrt{k}}$, we may write $w_i(t-1) = \frac{1}{\sqrt{k}} - \lambda$ for some $\lambda > 0$. Then by symmetry, for every $j \in F$, we also have $w_i(t-1) = \frac{1}{\sqrt{k}} - \lambda$. We thus have

that

$$\begin{aligned}
w_i(t) &= w_i(t-1) + \eta z(t-1)(1 - z(t-1)w_i(t-1)) \\
&= w_i(t-1) + \eta k \cdot \left(\frac{1}{\sqrt{k}} - \lambda\right)\left(1 - k\left(\frac{1}{\sqrt{k}} - \lambda\right)^2\right) \\
&= w_i(t-1) + \eta k \cdot \left(\frac{1}{\sqrt{k}} - \lambda\right)\left(1 - k\left(\frac{1}{k} - \frac{2\lambda}{\sqrt{k}} + \lambda^2\right)\right) \\
&< w_i(t-1) + \eta k \cdot \left(\frac{1}{\sqrt{k}}\right) 2\lambda\sqrt{k} \\
&\leq w_i(t-1) + \frac{\lambda}{2} \\
&< 1/\sqrt{k},
\end{aligned}$$

as needed.

For Part 4, consider any $i \notin F$. We have

$$\begin{aligned}
w_i(t) &= w_i(t-1) + \eta z(t-1)(0 - z(t-1)w_i(t-1)) \\
&= w_i(t-1) + \eta z(t-1)(0 - z(t-1)w_i(t-1)) \\
&= w_i(t-1)(1 - \eta z(t-1)^2) \\
&< w_i(t-1),
\end{aligned}$$

as needed.

Finally, for Part 5, again consider any $i \notin F$. We then have:

$$\begin{aligned}
w_i(t) &= w_i(t-1) + \eta z(t-1)(0 - z(t-1)w_i(t-1)) \\
&= w_i(t-1) - \eta z(t-1)^2 w_i(t-1) \\
&= w_i(t-1)(1 - \eta z(t-1)^2) \\
&> w_i(t-1)(1 - \eta k) \\
&\geq w_i(t-1)(1 - \frac{k}{4k}) \\
&= \frac{3}{4} w_i(t-1) \\
&> 0,
\end{aligned}$$

as needed. □

The following lemma, Lemma A.2, extends Lemma A.1 to any number of steps. This lemma assumes that the same x inputs are given to the given neuron u at every time. When we apply this later, in the proof of Lemma A.6, it will be in a context where these inputs may occur at separated times, namely, the particular times at which u is actually engaged in learning. At the intervening times, u will not be engaged in learning and will not change its weights.

Lemma A.2. *Let $F \subseteq \{1, \ldots, n\}$, with $|F| = k$. Assume that:*

1. *For every $t \geq 0$, $x_i(t) = 1$ for every $i \in F$ and $x_i(t) = 0$ for every $i \notin F$.*
2. *All weights $w_i(0)$ are equal.*
3. *$0 < w_i(0) < \frac{1}{\sqrt{k}}$ for every i.*

4. $0 < \eta \leq \frac{1}{4k}$.

Then for any $t \geq 1$:

1. *All weights $w_i(t), i \in F$ are equal, and all weights $w_i(t), i \notin F$ are equal.*
2. $0 < w_i(t) < \frac{1}{\sqrt{k}}$ *for every i.*
3. *For any $i \in F$, $w_i(t) > w_i(0)$.*
4. *For any $i \notin F$, $w_i(t) > w_i(0)$.*

The next lemma, Lemma A.3 gives quantitative bounds on the amount of weight increase and weight decrease, again for a single neuron u involved in learning a single concept. This lemma describes what happens in multiple rounds, not just a single round. We use notation $w(t), x(t), z(t)$ as before. We assume that $x(t)$ is the same at all times $0, 1, \ldots$, and assume that the engaged component $e(t)$ is equal to 1 at all times.

Lemma A.3 (Learning Properties)**.** *Let $F \subseteq \{1, \ldots, n\}$ with $|F| = k$. Let $\varepsilon \in (0, 1]$. Let b be a positive integer. Let $\sigma = \frac{4}{3\eta k}(\ell_{\max} \log(k)) + \frac{3}{\eta k \varepsilon} + \frac{b \log(k)}{\log(\frac{16}{15})}$. Thus, σ is $O\left(\frac{1}{\eta k}\left(\ell_{\max} \log(k) + \frac{1}{\varepsilon}\right) + b \log(k)\right)$. Assume that:*

1. *For every $t \geq 0$, $x_i(t) = 1$ for every $i \in F$, $x_i(t) = 0$ for every $i \notin F$, and $e(t) = 1$.*
2. *All weights $w_i(0)$ are equal to $\frac{1}{k^{\ell_{\max}}}$.*
3. $\eta = \frac{1}{4k}$.[3]

Then for every $t \geq \sigma$, the following hold:

1. *For any $i \in F$, we have $w_i(t) \in [\frac{1}{(1+\varepsilon)\sqrt{k}}, \frac{1}{\sqrt{k}}]$.*
2. *For any $i \notin F$, we have $w_i(t) \leq \frac{1}{k^{\ell_{\max}+b}}$.*

Proof. We first show Part 1. We begin with a Claim that bounds the time to double the weight w_i for $i \in F$, when w_i is not "too close" to the target weight $\frac{1}{\sqrt{k}}$.
Claim 1: Assume that $i \in F$. For $j \geq 1$, the number of rounds needed to increase w_i from $\frac{1}{2^{j+1}\sqrt{k}}$ to $\frac{1}{2^j\sqrt{k}}$ is at most $\frac{4}{3\eta k}$.
Proof of Claim 1: Using the assumption that all the weights are the same, and the assumption that $w_i \leq \frac{1}{2\sqrt{k}}$, we get:

$$\begin{aligned} w_i(t) &= w_i(t-1) + \eta z(t-1) \cdot (1 - z(t-1) \cdot w_i(t-1)) \\ &= w_i(t-1) + \eta k w_i(t-1)(1 - k w_i^2(t-1)) \\ &\geq w_i(t-1) + \frac{\eta k}{2^{j+1}\sqrt{k}}(1 - k\frac{1}{4k}) \\ &= w_i(t-1) + \frac{\eta k}{2^{j+1}\sqrt{k}}(3/4). \end{aligned}$$

Increasing w_i from $\frac{1}{2^{j+1}\sqrt{k}}$ to $\frac{1}{2^j\sqrt{k}}$ means we must increase it by $\frac{1}{2^{j+1}\sqrt{k}}$. Each round increases w_i by at least $\eta k \frac{1}{2^{j+1}\sqrt{k}}(3/4)$. Thus, the time to double w_i from $\frac{1}{2^{j+1}\sqrt{k}}$ to $\frac{1}{2^j\sqrt{k}}$ is at most $\frac{1}{2^{j+1}\sqrt{k}}$ divided

[3] This is a very precise assumption but it could be weakened, at a corresponding cost in run time.

by $\eta k \frac{1}{2^{j+1}\sqrt{k}}(3/4)$, which is $\frac{4}{3\eta k}$.
End of proof of Claim 1.
Claim 2: For $i \in F$, the total time to increase w_i from the starting value $\frac{1}{k^{\ell_{\max}}}$ to the target value $\frac{1}{2\sqrt{k}}$ is at most $\frac{4}{3\eta k}(\ell_{\max}\log(k))$.
Proof of Claim 2: Claim 1 implies that the total time required to increase w_i from $\frac{1}{k^{\ell_{\max}}}$ to the target value $\frac{1}{2\sqrt{k}}$ is at most $\frac{4}{3\eta k}(\ell_{\max}\log(k))$.
End of Proof of Claim 2.

Next, we bound the time required to increase w_i from $\frac{1}{2\sqrt{k}}$ to $\frac{1}{(1+\varepsilon)\sqrt{k}}$. This time, of course, depends on ε.
Claim 3: For $i \in F$, the time to increase w_i from $\frac{1}{2\sqrt{k}}$ to $\frac{1}{(1+\varepsilon)\sqrt{k}}$ is at most $\frac{3}{\eta k\varepsilon}$.
Proof of Claim 3: The argument is generally similar to that for Claim 1:

$$\begin{aligned}
w_i(t) &= w_i(t-1) + \eta z(t-1)(1 - z(t-1)w_i(t-1)) \\
&= w_i(t-1) + \eta k w_i(t-1)(1 - k w_i^2(t-1)) \\
&\geq w_i(t-1) + \frac{\eta k}{2\sqrt{k}}\left(1 - \frac{1}{(1+\varepsilon)^2}\right) \\
&= w_i(t-1) + \frac{\eta\sqrt{k}}{2}\left(1 - \frac{1}{(1+\varepsilon)^2}\right) \\
&\geq w_i(t-1) + \frac{\eta\sqrt{k}}{2}\frac{\varepsilon}{3}, \\
&= w_i(t-1) + \frac{\eta\sqrt{k}\varepsilon}{6},
\end{aligned}$$

where we used the fact that $(1 - 1/(1+x)^2) \geq x/3$ for $x \leq 1$. It follows that the total time to increase w_i from its initial value $\frac{1}{2\sqrt{k}}$ to the target value $\frac{1}{(1+\varepsilon)\sqrt{k}}$ is at most

$$\left(\frac{1}{(1+\varepsilon)\sqrt{k}} - \frac{1}{2\sqrt{k}}\right) \cdot \frac{6}{\eta\sqrt{k}\varepsilon} = \frac{1-\varepsilon}{2(1+\varepsilon)\sqrt{k}} \cdot \frac{6}{\eta\sqrt{k}\varepsilon} = \frac{6(1-\varepsilon)}{2(1+\varepsilon)\eta k\varepsilon} \leq \frac{3}{\eta k\varepsilon}.$$

End of Proof of Claim 3.

It follows that the total time for Part 1 is at most

$$\frac{4}{3\eta k}\left(\ell_{\max}\log(k)\right) + \frac{3}{\eta k\varepsilon}.$$

Note that once the weights for indices in F reach their target values, they never decrease below those values. This follows from strict monotonicity shown in Lemma A.2.

We now turn to proving Part 2. We consider what happens after the increasing weights (for indices in F) in reach the level $\frac{1}{2\sqrt{k}}$, and then bound the number of rounds for the decreasing weights to decrease to the desired target $\frac{1}{k^{\ell_{\max}+b}}$. The reason we choose the level $\frac{1}{2\sqrt{k}}$ for the increasing weights is that this is enough to guarantee that z is "large enough" to produce enough decrease.

For this part, we use our assumed lower bound on η.
Claim 4: For $i \notin F$, the time to decrease w_i from the starting weight $\frac{1}{k^{\ell_{\max}}}$ to $\frac{1}{k^{\ell_{\max}+b}}$ is at most $\frac{b\log_2 k}{\log_2\frac{16}{15}}$, which is $O(b\log(k))$.

Proof of Claim 4: Considering a single round, we get:

$$\begin{aligned} w_i(t) &= w_i(t-1)(1-\eta z(t-1)^2) \\ &\leq w_i(t-1)\left(1-\frac{1}{4k}\left(\frac{\sqrt{k}}{2}\right)^2\right) \\ &= w_i(t-1)\left(1-\frac{1}{16}\right) = w_i(t-1)\frac{15}{16}. \end{aligned}$$

The inequality uses the facts that $\eta \geq \frac{1}{4k}$ and $z(t-1) \geq k(\frac{1}{2\sqrt{k}}) = \frac{\sqrt{k}}{2}$.

Thus, the weight decreases by a factor of 15/16 at each round.

Now consider the number of rounds needed to reduce from $\frac{1}{k^{\ell_{\max}}}$ to the target weight $\frac{1}{k^{\ell_{\max}+b}}$.

This number is bounded by $\frac{b\log_2 k}{\log_2 \frac{16}{15}}$, which is $O(b\log(k))$, as claimed. *End of Proof of Claim 4.*

Summing up the three bounds yields the result. □

A.2 Invariants About Engagement, Weights, and Firing

We use some assumptions about the various parameters:

1. The concept hierarchy consists of $\ell_{\max}$ levels.
2. r_1, r_2 satisfy $0 < r_1 < r_2 \leq 1$.
3. $\varepsilon = \frac{r_2 - r_1}{r_1 + r_2}$.
4. $\eta = \frac{1}{4k}$.
5. b is an arbitrary positive integer.
6. σ, for the σ-bottom-up presentation definition, satisfies $\sigma = \frac{4}{3\eta k}(\ell_{\max}\log(k)) + \frac{3}{\eta k\varepsilon} + \frac{b\log(k)}{\log(\frac{16}{15})}$.

   Thus, σ is $O\left(\frac{1}{\eta k}\left(\ell_{\max}\log(k) + \frac{1}{\varepsilon}\right) + b\log(k)\right)$. Here, η, ε, and b are as described just above.

We start with an assumption about the engagement flags.

Assumption A.4. *For every time t and layer ℓ, a neuron u on layer $\ell \geq 1$ is engaged (i.e., $u.engaged = 1$) at time t, if and only if both of the following hold:*

1. *A level ℓ concept was shown at time $t - \ell$.*
2. *Neuron u is selected by the WTA at time t.*

Note that, by Assumption 5.2, the WTA selects exactly one neuron, which together with the assumption above implies that exactly one neuron will be engaged on layer ℓ at time t.

We say that a layer ℓ neuron u, $\ell \geq 1$, "binds" to a level ℓ concept c at time t if c is presented for the first time at time $t - \ell$, and u is the neuron that is engaged at time t. At that point, we define $rep(c) = u$.

Here is an auxiliary lemma, about unbound neurons.

Lemma A.5. *Let u be a neuron with layer$(u) \geq 1$. Then for every $t \geq 0$, the following hold:*

1. *If u is unbound at time t, then all of u's incoming weights at time t are the initial weight $\frac{1}{k^{\ell_{\max}}}$.*

2. *If u is unbound at time t, then u does not fire at time t.*

We are now ready to prove our main invariant.

Lemma A.6. *Consider any particular execution of the network in which inputs follow a σ-bottom-up schedule. For any $t \geq 0$, the following hold.*

1. *The $rep()$ mapping from the set of concepts to the set of neurons a is one-to-one mapping; that is, for any two distinct concepts c and c' for which $rep(c)$ and $rep(c')$ are both defined, we have $rep(c) \neq rep(c')$.*

2. *For every concept c with $level(c) \geq 1$, every showing of c at a time $\leq t - level(c)$, leads to the same neuron $u = rep(c)$ becoming engaged.*

3. *For every concept c with $level(c) \geq 1$, and any $t' \geq 1$, if c is shown at time $t - level(c)$ for the t'-th time, then the following are true at time t:*

   (a) *If $t' \geq 1$, then u with $u = rep(c)$ has weights in $\left(\frac{1}{k^{\ell_{\max}}}, \frac{1}{\sqrt{k}}\right)$ for all neurons in $rep(children(c))$, and weights in $\left(0, \frac{1}{k^{\ell_{\max}}}\right)$ for all other neurons.*

   (b) *If $t' \geq \sigma$, then u with $u = rep(c)$ has weights in $\left[\frac{1}{(1+\varepsilon)\sqrt{k}}, \frac{1}{\sqrt{k}}\right]$ for all neurons in $rep(children(c))$, and weights in $\left[0, \frac{1}{k^{\ell_{\max}+b}}\right]$ for all other neurons.*

4. *For every concept c, if a proper ancestor of c is shown at time $t - level(c)$, then $rep(c)$ is defined by time t, and fires at time t.*

5. *For any neuron u, the following holds. If u fires at time t, then there exists c such that $u = rep(c)$ at time t, and an ancestor of c is shown at time $t - layer(u)$. (This ancestor could be c or a proper ancestor of c.)*

Proof. First observe that, by Assumption A.4, every representative $rep(c)$ is on the layer equal to $level(c)$. We prove the five-part statement of the lemma by induction on t.

Base: $t = 0$.
Then for Part 1, the only concepts for which *reps* are defined at time 0 are level 0 concepts, and these all have distinct *reps* by assumption. For Parts 2 and 3, note that $level(c) \geq 1$ implies that the times in question are negative, which is impossible; so these are trivially true. For Part 4, it must be that $level(c) = 0$ (to avoid negative times), and a proper ancestor of c is shown at time 0. Then the layer 0 neuron $rep(c)$ fires at time 0, by the definition of "showing".

For Part 5, first note that at time 0 no neurons at layers ≥ 1 are bound, so by Lemma A.5, they cannot fire at time 0. Since we assume that u fires at time 0, it must be that $layer(u) = 0$, which implies that $u = rep(c)$ for some level 0 concept c. Then, since u fires at time 0, by definition of "showing, an ancestor of c must be shown at time 0.
Inductive step: Assume the five-part claim holds for time $t - 1$ and consider time t.
We prove the five parts one by one.

1. For Part 1, let c and c' be any two distinct concepts for which $rep(c)$ and $rep(c')$ are both defined at time t. We must show that $rep(c) \neq rep(c')$.

   If both $rep(c)$ and $rep(c')$ are defined at time $t - 1$, then by the I.H., Part 1, $rep(c) \neq rep(c')$ at time $t - 1$. Since the *reps* do not change, this is still true at time t, as needed. So the only possibility for conflict is that one of these two concepts, say c', already has its *rep* defined at

time $t-1$ but the other concept, say c, does not, and $rep(c)$ becomes defined at time t. Let $u = rep(c)$; we show that u is unbound at time $t-1$.

By Assumption A.4, the *engaged* flag gets set at time t for u, and for no other neurons. Since c is shown at time $t-\ell$, by the σ-bottom-up assumption, each child of c must have been shown at least σ times prior to time $t-\ell$. Then by the I.H., Parts 4 and 5, the layer $\ell-1$ neurons "fire correctly" at time $t-1$, i.e., all neurons in the set $rep(children(c))$ fire and no other layer $\ell-1$ neuron fires, at time $t-1$.

This implies that every neuron that is already bound prior to time t has incoming potential in round t strictly less than k times the initial weight, by I.H. 3.(a) and by the disjointness of the concepts. On the other hand, every unbound neuron has incoming potential equal to k times the initial weight, by Lemma A.5. By assumption, there is at least one unbound neuron available. It follows that the neuron u that is chosen by the WTA is unbound, and so cannot be equal to $rep(c)$.

2. For Part 2, let c be any concept with $level(c) \geq 1$, and write $\ell = level(c)$. We must prove that any showing of c at any time $\leq t-\ell$ leads to the same neuron $u = rep(c)$ becoming engaged.

   If c is not shown at time precisely $t-\ell$, then the claim follows directly from the I.H., Part 2.

   So assume that c is shown at time $t-\ell$. If $t-\ell$ is the first time that c is shown, then $rep(c)$ first gets defined at time t, so the conclusion is trivially true (since there is only one showing to consider).

   It remains to consider the case where $rep(c)$ is already defined by time $t-1$. Then, by the I.H. Part 2, we know that any showing of c at a time $\leq t-1-\ell$ leads to neuron $rep(c)$ becoming engaged.

   We now argue that the same $rep(c)$ is also selected at time t. As in the proof of Part 1, the *engaged* flag is set at time t for exactly one layer ℓ neuron; we claim that this chosen neuron is in fact the previously-defined $rep(c)$. As in the proof for Part 1, we claim that all neurons in the set $rep(children(c))$ fire and no other layer $\ell-1$ neuron fires at time $t-1$. Then $rep(c)$ has incoming potential in round t that is strictly greater than k times the initial weight, by I.H. Part 3(a). On other hand, every other layer ℓ neuron has incoming potential that is at most k times the initial weight, again by I.H. Part 3(a). It follows that $rep(c)$ has a strictly higher incoming potential in round t than any other layer ℓ neuron, and so is the chosen neuron at time t.

3. For Part 3, let c be any concept with $level(c) \geq 1$, and write $\ell = level(c)$. Let $t' \geq 1$. Assume that c is shown at time $t-\ell$ for the t'-th time. We must show:

   (a) If $t' \geq 1$, then u with $u = rep(c)$ has weights in $\left(\frac{1}{k^{\ell_{\max}}}, \frac{1}{\sqrt{k}}\right)$ for all neurons in $rep(children(c))$, and weights in $\left(0, \frac{1}{k^{\ell_{\max}}}\right)$ for all other neurons.

   (b) If $t' \geq \sigma$, then u with $u = rep(c)$ has weights in $\left[\frac{1}{(1+\varepsilon)\sqrt{k}}, \frac{1}{\sqrt{k}}\right]$ for all neurons in $rep(children(c))$, and weights in $\left[0, \frac{1}{k^{\ell_{\max}+b}}\right]$ for all other neurons.

   For both parts, we use Part 2 (for t, not $t-1$) to infer that every showing of c at a time $\leq t - level(c)$ leads to the same neuron $u = rep(c)$ being engaged. Thus, neuron u has been engaged t' times as a result of showing c, up to time t.

   For Part (a), assume that $t' \geq 1$. Then we may apply Lemma A.2, with $F = rep(children(c))$, to conclude that the incoming weights for u are in the claimed intervals. For Part (b), assume that $t' \geq \sigma$. Then we may apply Lemma A.3, with $F = rep(children(c))$, to conclude that the incoming weights for u are in the claimed intervals.

4. For Part 4, let c be any concept, and assume that c^*, a proper ancestor of c, is shown at time $t - level(c)$. We must show that $rep(c)$ is defined by time t, and that it fires at time t.

   Since c^* is shown at time $t - level(c)$, by the definition of a σ-bottom-up schedule, that means c was shown at least σ times by time $t - level(c) - 1$. This implies that $rep(c)$ is defined by time $t - 1$, and so, by time t.

   Moreover, since c was shown at least σ times by time $t - level(c) - 1$, by the I.H., Part 3(b), at time $t - 1$, $rep(c)$ has incoming weights at least $\frac{1}{(1+\varepsilon)\sqrt{k}}$ for all neurons in $rep(children(c))$. By the I.H. Part 4, the neurons in $rep(children(c))$ fire at time $t - 1$ since c^* is also a proper ancestor of all children of c. Therefore, at time t, the potential of $rep(c)$ is at least $k \cdot \frac{1}{(1+\varepsilon)\sqrt{k}}$, which by definition means that u fires at time t.

5. For Part 5, fix an arbitrary neuron u and suppose that u fires at time t. We must show that there is some concept c such that $u = rep(c)$ at time t, and an ancestor of c is shown at time $t - layer(c)$.

   Since u fires at time t, by Lemma A.5, we know that u is bound at time t; let c be the (unique) concept such that $u = rep(c)$. The firing of u at time t is due to the showing of some concept, say c^*, at time $t - layer(u)$.

   Let R be the subset of $rep(children(u))$ that fire at time $t - 1$. We claim that $|R| \geq 2$; that is, at least two *reps* of children of c must fire at time $t - 1$. For, if at most one $rep(c')$ for a child of c fires at time $t - 1$, then by the I.H., Part 3(a), the total potential incoming to u at step t would be at most $\frac{1}{\sqrt{k}} + \frac{k^{\ell_{\max}}}{k^{\ell_{\max}}} < \frac{1}{\sqrt{k}} + 1$, which is smaller than the threshold for firing.

   So $|R'| \geq 2$; let u' and u'' be any two distinct elements of R'. Since u' and u'' fire at time $t-1$, by Lemma A.5, we know that both are bound at time $t - 1$; let c' and c'' be the respective concepts such that $u' = rep(c')$ and $u'' = rep(c'')$. We know that $c' \neq c''$ because each concept gets only one *rep* neuron, by the way that *rep* is defined. Note that the firing of both u' and u'' must be due to the showing of concept c^* at time $(t - 1) - (layer(u) - 1) = t - layer(u)$. Then by the I.H., Part 5, applied to both u' and u'', we see that c^* must be an ancestor of both c' and c''. Therefore, c^* must be an ancestor of the common parent c of c' and c'', as needed.

This completes the proof. □

A.3 Proof of Theorem 5.3

Now we use Lemma A.6 to prove our main theorem, Theorem 5.3. The proof is straightforward, but we include it for the sake of completeness.

Proof. By assumption, all the concepts in the hierarchy have been shown according to a σ-bottom-up schedule.

This implies, by Assumption A.4, that all the concepts in the hierarchy have *reps* in the corresponding layers. Also, by Lemma A.6, Part 3(b), the weights are set as as follows: For every concept c with $level(c) \geq 1$, all incoming weights of $rep(c)$ from the *reps* of its children, i.e., the neurons in $rep(children(c)$, are in the range $[\frac{1}{(1+\varepsilon)\sqrt{k}}, \frac{1}{\sqrt{k}}]$, and weights from all other neurons (on layer $level(c) - 1$) are at most $\frac{1}{k^{\ell_{\max}+b}}$.

Now we must argue that the resulting network (r_1, r_2)-recognizes the concept hierarchy. This has two directions. First, consider any subset $B \subseteq C_0$. We argue by induction on levels ℓ, $1 \leq \ell \leq \ell_{\max}$, that the *rep* of any level ℓ concept in $supported_{r_2}(B)$ fires (see Definition 2.1 for the definition of $supported_{r_2}$).

For the base case, consider a level 1 concept $c \in \mathit{supported}_{r_2}(B)$. Since $c \in \mathit{supported}_{r_2}(B)$, it means that $|\mathit{children}(c) \cap B| \geq r_2 k$. As noted above, the *rep* of each of these children is connected to $\mathit{rep}(c)$ by an edge with weight at least $\frac{1}{(1+\varepsilon)\sqrt{k}}$, which yields a total incoming potential for $\mathit{rep}(c)$ of at least

$$\frac{r_2 k}{(1+\varepsilon)\sqrt{k}} = \frac{r_2\sqrt{k}}{1+\varepsilon}.$$

We must show that the right-hand side is at least as large as the firing threshold $\tau = \frac{r_1+r_2}{2}\sqrt{k}$. That is, we must show that $\frac{r_2}{1+\varepsilon} \geq \frac{r_1+r_2}{2}$. Plugging in the expression for ε, we get that:

$$\frac{r_2}{1+\varepsilon} = \frac{r_2}{1+\frac{r_2-r_1}{r_1+r_2}} = \frac{r_1+r_2}{2},$$

as needed.

For the inductive step, consider $\ell \geq 2$ and suppose that the *rep* of any level $\ell-1$ concept in $\mathit{supported}_{r_2}(B)$ fires. Consider a level ℓ concept $c \in \mathit{supported}_{r_2}(B)$. This means that $|\mathit{children}(c) \cap B_{\ell-1}| \geq r_2 k$, using notation from the definition of "supported" (Definition 2.1), that is, at least $r_2 k$ children of c are in $\mathit{supported}_{r_2}(B)$. By Lemma A.6, Part 3(b), the *rep* of each of these children is connected to $\mathit{rep}(c)$ by an edge with weight at least $\frac{1}{(1+\varepsilon)\sqrt{k}}$, which yields a total incoming potential for $\mathit{rep}(c)$ of at least

$$\frac{r_2 k}{(1+\varepsilon)\sqrt{k}} = \frac{r_2\sqrt{k}}{1+\varepsilon}.$$

Arguing as in the base case, this is at least as large as the firing threshold τ, as needed.

For the other direction, again consider any subset $B \subseteq C_0$. First note that, by Lemma A.5, unbound neurons never fire. We now argue that for any *rep* of any concept c in the hierarchy such that $c \notin \mathit{supported}_{r_1}(B)$ does not fire. Recognizing that levels correspond to layers, this time we proceed by induction on layers ℓ, $1 \leq \ell \leq \ell_{\max}$.

For the base case, consider a level 1 concept $c \notin \mathit{supported}_{r_1}(B)$; $\mathit{rep}(c)$ is at layer 1. Since $c \notin \mathit{supported}_{r_1}(B)$, it means that $|\mathit{children}(c) \cap B| < r_1 k$. By Lemma A.6, Part 3(b), the *rep* of each of these children is connected to $\mathit{rep}(c)$ by an edge with weight $\leq \frac{1}{\sqrt{k}}$

and all other incoming weights of $\mathit{rep}(c)$ are at most $\frac{1}{k^{\ell_{\max}+b}}$.

We add a technical assumption here: that $r_1 k$ is not an integer, and the difference between $r_1 k$ and the next-smaller integer (which is $r_1 k - \lceil r_1 k - 1 \rceil$) is at least $\frac{\sqrt{k}}{k^b}$.

Writing a as an abbreviation for $r_1 k - \lceil r_1 k - 1 \rceil$, we get that the total incoming potential for $\mathit{rep}(c)$ is at most

$$\frac{r_1 k - a}{\sqrt{k}} + \frac{k^{\ell_{\max}}}{k^{\ell_{\max}+b}} = r_1\sqrt{k} - \frac{a}{\sqrt{k}} + \frac{1}{k^b} \leq r_1\sqrt{k} < \frac{r_1+r_2}{2}\sqrt{k} = \tau,$$

which implies that $\mathit{rep}(c)$ does not fire.

For the inductive step, consider $\ell \geq 2$ and suppose that the *rep* of any level $\ell-1$ concept c such that $c \notin \mathit{supported}_{r_1}(B)$ does not fire. Consider a level ℓ concept $c \notin \mathit{supported}_{r_1}(B)$. This implies that $|\mathit{children}(c) \cap B_{\ell-1}| < r_1 k$, because if this cardinality were $\geq r_1 k$, then by the recursive definition of "supported", c would be in $\mathit{supported}_{r_1}(B)$.

By the inductive hypothesis, only the *reps* of concepts in $\mathit{children}(c) \cap B_{\ell-1}$ contribute to the incoming potential for $\mathit{rep}(c)$. By Lemma A.6, each such *rep* is connected to $\mathit{rep}(c)$ by an edge with weight at most $\frac{1}{\sqrt{k}}$ and all other neurons have weight at most $\frac{1}{k^{\ell_{\max}+b}}$. Thus, the total incoming potential for $\mathit{rep}(c)$ is at most

$$\frac{r_1 k - a}{\sqrt{k}} + \frac{k^{\ell_{\max}}}{k^{\ell_{\max}+b}},$$

which is strictly less than τ as in the base case. Thus $rep(c)$ does not fire.

□

B Analysis of Noisy Learning

We start by giving a proof overview.

B.1 Proof Overview

The overall analysis of Theorem 6.1 is at its core similar to the analysis of Theorem 5.3 presented in Appendix A.

The main difference is that the weights of the neurons after learning are slightly different: Following the notation of Lemma A.1, Lemma A.2 and Lemma A.3, we show that for all $i \in F$ the weights will eventually approximate

$$\bar{w} = \frac{1}{\sqrt{pk + 1 - p}}$$

and for $i \notin F$, the weights are bounded by $1/k^{2\,\ell_{\max}}$. Note that, in this section, we set the parameter b, governing the desired decrease of unrelated weights, to be $b = \ell_{\max}$. Also note that we recover the noise-free case by setting $p = 1$.[4]

The main difficulty in the noisy case is to establish a noisy version of Lemma A.3, which we do in Lemma B.1.

Due to the noise, main structural properties, such as weights of neurons in F changing monotonically, do not hold anymore. To make matters worse, we cannot simply use Chernoff bounds and assume the worst-case distribution of the weight changes. Instead, we work with a fine-grained potential analysis, which we describe in more detail in the paragraph "Proof idea" below.

After establishing Lemma B.1, proving the main theorem is then analogous to the noise-free case. This is because the overall structure of the network follows exactly the noise-free case, with the small exception that the weights are slightly different. Nonetheless, the same arguments as in the proof Lemma A.6 still hold.

Proof idea. First we bound the change of the potential during a period of T rounds (Lemma B.3). We then derive very rough bounds on the change of a single weight during such a period (Lemma B.4). Using these rough results, we are able to prove much more precise bounds on the change of the weights in a given interval of length T. It turns out that the way the weights change depends highly on the other weights, which makes the analysis non-trivial. The way we show that weights converge, is by using the following potential ψ. Fix an arbitrary time t and let $w_{min}(t)$ and $w_{max}(t)$ be the minimum and maximum weights among $w_1(t), w_k(t), \ldots, w_k(t)$, respectively. Let

$$\psi(t') = \max\left\{\frac{w_{max}(t')}{\bar{w}}, \frac{\bar{w}}{w_{min}(t')}\right\}.$$

Our goal is to show that this potential decreases quickly until it is very close to 1. Showing that the potential decreases is involved, since one cannot simply use a worst case approach, due to the terms in Oja's rule being non-linear and potentially having a high variance, depending on the distribution of weights. The key to showing that ψ decreases is to carefully use the randomness over the input vector x together with the rough bounds (established before) on the worst-case values of w_i and z during the interval. Bounding these non-linear terms tightly presents a major challenge which we overcome using two techniques. First, we consider a process P' which is almost as the

[4] In this case the probabilistic guarantees become deterministic guarantees.

original process P, with the difference that the weights only change marginally in each period of T rounds. If they change by more, then we assume that the weights are simply reset to the value at the beginning of the T rounds. As we will see later, we can couple the processes P and P' with high probability. This coupling allows us to avoid a conditioning that would otherwise change the probability space and prevent as from using the following second technique. Second, we show that the changes of the weights form a Doob maringales allowing us to use Azuma-Hoeffding inequality to get asymptotically almost tight bounds on the change of the weights during the T rounds starting at time t. To this end we define

$$X(t') = z(t+t') \cdot \big(x_{i^*}(t+t') - z(t+t') \cdot w_{i^*}(t+t')\big)$$

and

$$S = \sum_{t' \le T} X(t'). \tag{2}$$

We bound these quantities in Lemma B.7 and Lemma B.8. We then show that these imply Lemma B.1. Finally, at the end of the section, we prove Theorem 6.1.

B.2 Weight Change for Individual Neurons

We consider the first $O(n^6)$ rounds. Consider any interval of length T. Note that the number of intervals is bounded by $O(n^6)$. Define the event $\mathcal{E}$ to be the event that for each interval and each input neuron u, the number times u fires is in $[(1-\delta)pT, (1+\delta)pT]$.

Using a Chernoff bound and a union bound,

we get that $\mathcal{E}$ holds w.h.p. In the following we will often condition on event $\mathcal{E}$.

We use some assumptions about the various parameters:

1. $\delta = \frac{r_2 - r_1}{r_2}/50$,
2. $b = \frac{100}{\delta}$,
3. $T = \frac{2^{10}k^4 \log n}{p^6\delta^2}$,
4. The learning rate $\eta = \frac{1}{4Tk^4}$.
5. The firing threshold $\tau = \frac{r_2 k \bar{w}}{(1+10\delta)}$

Lemma B.1 (Learning Properties, Noisy Case)**.** *Let $F \subseteq \{1, \dots, n\}$ with $|F| = k$. Let $\varepsilon \in (0, 1]$. Let b be a positive integer.*

Let $\sigma = c' \frac{1}{\eta k}\left(\ell_{\max}\log(k) + \frac{r_2 k + 1 - r_2}{\eta r_2^{3/2}(r_2 - r_1)}\right) + \log(k)$, for some large enough constant c'.

Assume that:

1. *For every $t \ge 0$, $x_i(t) = 0$ for every $i \notin F$, and $e(t) = 1$.*
2. *All weights $w_i(0)$ are equal to $\frac{1}{k^{\ell_{\max}}}$.*
3. *$\eta = \frac{1}{4Tk^4}$.*[5]
4. *$\mathcal{E}$ holds.*

Then for every $t \ge \sigma$, the following hold:

[5] This is a very precise assumption but it could be weakened, at a corresponding cost in run time.

1. *For any $i \in F$, we have $w_i(t) \in [\frac{\bar{w}}{(1+20\delta)}, (1+20\delta)\bar{w}]$.*

2. *For any $i \notin F$, we have $w_i(t) \leq \frac{1}{k^{2\,\ell_{\max}}}$.*

Let

$$\phi(t) = \sum_{i \leq k} w_i.$$

We now give some structural properties.

Observation B.2. *For every i and t,*

$$w_i(t) \in [0, 1]. \tag{3}$$

Moreover, for all $i \notin F$, w_i is monotonically decreasing.

Proof. We can show this by induction. Using the inductive hypothesis $w_i(t-1) \in [0,1]$. we have $w_i(t-1) \leq z(t-1) \leq \sum_{j \in F} w_j(t-1) \leq k$. Thus $w_i(t) = w_i(t-1) + \eta z(t-1)(1 - z(t-1)w_i(t-1)) \leq w_i(t-1) + \eta k(1 - w_i(t-1)w_i(t-1)) \leq 1$. Thus set $w_i(t-1) = 1 - \lambda$ for some $\lambda > 0$ and we get

$$w_i(t) \leq 1 - \lambda + \eta k(1 - (1-\lambda)^2) \leq 1 - \lambda + \frac{1}{2}(2\lambda - \lambda^2) \leq 1.$$

Similarly, $w_i(t) \geq w_i(t-1) - \eta k z(t-1) w_i(t-1) \geq w_i(t-1)(1 - \eta k^2) \geq 0$.
It remains to argue that for all $i \notin F$, w_i is monotonically decreasing. This holds since,

$$w_i(t) \leq w_i(t-1) - \eta k z(t-1) w_i(t-1) \leq w_i(t-1).$$

□

The following lemma show that $\phi(\cdot)$ does not change too much an interval of length T.

Lemma B.3. *Assume $\mathcal{E}$ holds. For every t and $t' \leq T$ we have*

$$\phi(t)\left(1 - \frac{8}{b}\right) \leq \phi(t + t') \leq \phi(t)\left(1 + \frac{8}{b}\right).$$

Proof. First note that $z(t) \leq \phi(t) \leq k$. Observe that $\phi(t+1) \leq \phi(t) + \eta\phi(t)k = \phi(t)(1 + \frac{1}{Tbk})$. Thus,

$$\phi(t+t') \leq \phi(t)(1 + \frac{1}{Tbk})^T \leq \phi(t) e^{\frac{1}{bk}} \leq \phi(t) e^{\frac{1}{b}} \leq \phi(t)(1 + \frac{8}{b})$$

for $b \geq 1$. Thus, for $b \geq 8$,

$$\phi(t+t') \geq \phi(t) - T\eta \max_{t'' \leq T} \phi(t+t'')^2 k \geq \phi(t) - T\eta\phi(t)^2(1 + \frac{8}{b})^2 k \geq \phi(t) - \frac{8}{b}\phi(t) = \phi(t)(1 - \frac{8}{b}).$$

□

From this we derive the following two corollaries bounding the change of the weights (Lemma B.4) as well as the change of the first two moments of z (Lemma B.5).

Lemma B.4. *Assume $\mathcal{E}$ holds. For every t and $t' \leq T$ we have*

$$-\eta T(1 + 8/b)^2 \phi(t)^2 w_i(t) \leq w_i(t+t') - w_i(t) \leq \eta T(1 + 8/b)\phi(t) \tag{4}$$

From this it follows that

$$(1 - \delta) w_i(t) \leq w_i(t + t') \leq (1 + \delta) w_i(t) \tag{5}$$

Lemma B.5. *Assume $\mathcal{E}$ holds. For every t and $t' \le T$ we have*

$$(1-2\delta)pT\phi(t) \le \sum_{t'\le T} z(t+t') \le (1+2\delta)pT\phi(t). \tag{6}$$

and

$$(1-2\delta)pT\sum_{i\le k} w_i^2(t) \le \sum_{t'\le T} z^2(t+t'). \tag{7}$$

Proof. By Lemma B.4 and assuming that $\mathcal{E}$ holds,

$$\sum_{t'\le T} z(t+t') \ge (1-\delta)pT\sum_{i\le k}\left(w_i(t) - \eta T(1+8/b)^2\phi(t)^2 w_i(t)\right).$$

Using $(1+8/b)^2 \le 2$ and $\phi(\cdot) \le k$, we get

$$\sum_{t'\le T} z(t+t') \ge (1-\delta)pT\phi(t)(1-2/b) \ge (1-2\delta)pT\phi(t)$$

To show the r.h.s. of (6), we apply Lemma B.3, and derive

$$\sum_{t'\le T} z(t+t') \le (1+\delta)pT\left(\phi(t) + \sum_{i\le k}\eta T(1+8/b)\phi(t)\right) \le (1+2\delta)pT\phi(t).$$

We now prove (7). By Lemma B.4 and assuming that $\mathcal{E}$ holds,

$$\begin{aligned}
\sum_{t'\le T} z^2(t+t') &\ge (1-\delta)pT\sum_{i\le k}\left(w_i(t) - \eta T(1+8/b)^2\phi(t)^2 w_i(t)\right)^2 \\
&\ge (1-\delta)pT\sum_{i\le k}\left(w_i^2(t) - 2w_i(t)\eta T(1+8/b)^2\phi(t)^2 w_i(t)\right) \\
&\ge (1-\delta)pT\sum_{i\le k} w_i^2(t)\left(1 - 2\eta T(1+8/b)^2\phi(t)^2\right) \\
&\ge (1-\delta)pT\sum_{i\le k} w_i^2(t)\left(1 - 2\eta T 2k^2\right) \\
&\ge (1-2\delta)pT\sum_{i\le k} w_i^2(t),
\end{aligned}$$

where we used that $(1+8/b)^2 \le 2$ and $\phi(\cdot) \le k$.

□

The following lemma shows that the potential increases exponentially until it is of order $\sqrt{p}$

Lemma B.6. *Assume $\mathcal{E}$ holds. Fix any t and $t' \le T$. Let $\phi(t) = \sum_{i\le k} w_i$. For every t with $\phi(t) \le \sqrt{p}/8$ we have that*

$$\phi(t+T) \ge \left(1 + \frac{\eta T p}{4}\right)\phi(t).$$

Furthermore, Once $\phi(t) \ge \sqrt{p}/8$ we have for all $t' \in [t, O(n^6)]$ that $\phi(t') \ge \sqrt{p}/16$.

Proof. Note that $z(t') \leq \phi(t') \leq \sqrt{p}/4$. Note that when $x_i(t') = 1$, the potential increases at least by $\eta w_i(t') \geq (1-\delta)\eta w_i(t)$. This will happen at least $pT(1-\delta)$ times. Summing over all $i \leq k$ gives an increase of at least

$$pT\eta(1-\delta)^2 \sum_{i \leq k} w_i(t) = pT\eta(1-\delta)^2\phi(t).$$

Otherwise, for $x_i(t') = 0$, (which only happens for at most $T(1-p(1-\delta))$ rounds), the potential decreases by

$$\eta \sum_{i \leq k} z(t')^2 w_i(t') \leq \eta \sum_{i \leq k} z(t)^2 w_i(t)(1+\delta)^3 \leq \eta \left(\frac{\sqrt{p}}{4}\right)^2 \phi(t)(1+\delta)^3.$$

Note that $(1-p(1-\delta))(1+\delta)^3 \leq 4$ and $(1-\delta)^2 \geq 3/4$. Thus

$$\phi(t+T) \geq \eta pT(1-\delta)^2\phi(t) - T(1-p(1-\delta))\eta \left(\frac{\sqrt{p}}{4}\right)^2 \phi(t)(1+\delta)^3 \geq T\eta p\phi(t)/4.$$

The second part follows from Lemma B.3.

□

Lemma B.7. *Consider the process P', which, after every round, resets the weights of all weights that exceed the bounds of Lemma B.3 and the second part of Lemma B.4. The exceeding weights are set back to the weights at the beginning of the round. For any $t' \leq T$. Let $\mathcal{F}_t$ denote the filtration up to time t, which informally speaking fixes all the random decisions during the first t rounds.*

$$\mathbb{E}\left[\, z(t+t') \mid \mathcal{F}_t \,\right] \geq (1-\delta)p\phi(t+t')$$

and

$$\mathbb{E}\left[\, z(t+t')^2 w_{i^*}(t+t') \mid \mathcal{F}_t \,\right] \leq (1+\delta)^3 p\phi(t)\left((1-p)w_{max}(t)w_{i^*}(t) + pw_{i^*}(t)\phi(t)\right)$$

Proof. In the following, the randomness is over $x_i(t+t')$. We have

$$\mathbb{E}\left[\, z(t+t') \mid w(t+t'), \mathcal{F}_t \,\right] = p\phi(t')m.$$

Moreover,

$$\begin{aligned}
\mathbb{E}\left[\, z(t+t')^2 \mid w(t+t'), \mathcal{F}_t \,\right] &= \sum_{i \leq k} \left(pw_i(t+t')^2 + p^2 w_i(t+t') \sum_{j \leq k, j \neq i} w_j(t+t') \right) \\
&= \sum_{i \leq k} \left(pw_i(t+t')^2 - p^2 w_i(t+t')^2 + p^2 w_i(t+t')\phi(t+t') \right) \\
&= (p-p^2) \sum_{i \leq k} w_i(t+t')^2 + p^2\phi(t+t')^2.
\end{aligned}$$

Combining this with the assumption of the process P', in which the weights do not diverge too much, we get

$$\begin{aligned}
\mathbb{E}\left[\, z(t+t')^2 w_{i^*}(t+t') \mid \mathcal{F}_t \,\right] &\le w_{i^*}(t+t')(1+\delta)^2 \left((p-p^2) \sum_{i \le k} w_i(t)^2 + p^2 \phi(t)^2 \right) \\
&\le w_{i^*}(t+t')(1+\delta)^2 \left((p-p^2) \sum_{i \le k} w_i(t)^2 + p^2 \phi(t)^2 \right) \\
&\le w_{i^*}(t)(1+\delta)^3 \left((p-p^2) w_{max}(t) \phi(t) + p^2 \phi(t)^2 \right) \\
&\le (1+\delta)^3 p \phi(t) \left((1-p) w_{max}(t) w_{i^*}(t) + p w_{i^*}(t) \phi(t) \right).
\end{aligned}$$

□

Lemma B.8. *Consider the process P', which, after every round, resets the weights of all weights that exceed the bounds of Lemma B.3 and the second part of Lemma B.4. The exceeding weights are set back to the weights at the beginning of the round. Fix an arbitrary time t. We have, with high probability,*

$$\psi(t+T) \le \max \left\{ \psi(t) - \eta T \phi(t) p^2 \bar{w} \frac{\bar{w} - w_{i^*}}{4}, (1+10\delta)\bar{w} \right\}.$$

Proof. By Lemma B.4, each weight increases throughout $[t, t+T]$ at most by a factor $(1+\delta)$ and decrease by at most a factor $(1-\delta)$.

W.l.o.g. assume

$$\frac{\bar{w}}{w_{min}(t)} \ge (1-2\delta) \frac{w_{max}(t)}{\bar{w}}. \tag{8}$$

Note that for all $i \le k$ with $w_i(t) \ge (1+2\delta) w_{min}$, we have $w_i(t+T) \ge (1+\delta/50) w_{min}$. Thus, we only consider the neurons i^* with $w_{i^*}(t) \in [w_{min}, (1+2\delta) w_{min}]$. By the second part of Lemma B.7, for $t' \le T$

$$\mathbb{E}\left[\, z(t+t')^2 w_{i^*}(t+t') \,\right] \le (1+\delta)^3 p \phi(t) \left((1-p) w_{max}(t) w_{i^*}(t) + p w_{i^*}(t) \phi(t) \right).$$

We now bound the terms in the parentheses. First note that

$$w_{i^*}(t) w_{max}(t) \le (1+2\delta) w_{min}(t) w_{max}(t) \le (1+6\delta) \bar{w}^2$$

and furthermore,

$$\begin{aligned}
w_{i^*}(t) \phi(t) &\le (k-1)(1+\delta) w_{i^*}(t) w_{max} + (1+\delta) w_{i^*}(t) w_{i^*}(t) \\
&\le (1+8\delta) \left((k-1)\bar{w}^2 + w_{i^*}(t)^2 \right) \\
&= (1+8\delta) \left(k\bar{w}^2 + w_{i^*}(t)^2 - \bar{w}^2 \right)
\end{aligned}$$

$$\begin{aligned}
(1-p) w_{max}(t) w_{i^*}(t) + p w_{i^*}(t) \phi(t) &\le (1+8\delta) \left((1-p)\bar{w}^2 + pk\bar{w}^2 + p(w_{i^*}(t)^2 - \bar{w}^2) \right) \\
&= (1+8\delta) \left(1 - p(\bar{w}^2 - w_{i^*}(t)^2) \right)
\end{aligned}$$

Therefore,

$$\mathbb{E}\left[\, z(t+t')^2 w_{i^*}(t+t') \,\right] \le (1+13\delta) p \phi(t) \left(1 - p(\bar{w}^2 - w_{i^*}(t)^2) \right)$$

$(1+8x)(1+x)^3 \le (1+13x)$ for $x \le 0.69$. Finally, using the definition of S (Equation 2) and combining the above with the first part of Lemma B.7,

$$\begin{aligned}\mathbb{E}[S] &\ge T\left(\mathbb{E}\left[z(t+t')\right] - \mathbb{E}\left[z(t+t')^2 w_{i^*}(t+t')\right]\right)\\ &\ge T\phi(t)p\left((1-\delta) - (1+13\delta)\left(1-p(\bar{w}^2 - w_{i^*}(t)^2)\right)\right)\\ &\ge T\phi(t)p^2\frac{\bar{w}^2 - w_{i^*}(t)^2}{2},\end{aligned}$$

where we used that $(1-y)(1-13y)(1-x) \ge x\cdot y/2$ for $x \le 0.9$ and $y \le 1/20$. We seek to apply Theorem D.1 to S. Let X_i be the random choices of the pk children that fire in round i (in the definition of the noisy learning). Recall that $S = \sum_{t' \le T} z(t+t')\cdot(x_{i^*}(t+t') - z(t+t')\cdot w_{i^*}(t+t'))$. We define $Y_i = \mathbb{E}[S \mid X_i, \dots, X_1]$ and observe that $\bar{S} = \mathbb{E}[S \mid X_T, \dots, X_1] = Y_T$. We seek to show that $Y_0, Y_1, \dots, Y_T$ is a (Doob) martingale with respect to the sequence $X_0, X_1, \dots X_T$. Indeed we have, using the tower rule,

$$\mathbb{E}[Y_i \mid X_{i-1}, \dots, X_1] = \mathbb{E}[\mathbb{E}[S \mid X_i, \dots, X_1] \mid X_{i-1}, \dots, X_1] = \mathbb{E}[S \mid X_{i-1}, \dots, X_1] = Y_{i-1}.$$

Thus, we can apply Theorem D.1 to the Doob martingale $Y_T, Y_{T-1}, \dots, Y_1$ with $|Y_{i+1} - Y_i| \le k^2$ for all i. we derive

$$\begin{aligned}\mathbb{P}\left[|S - \mathbb{E}[S]| \ge \frac{\mathbb{E}[S]}{2}\right] &\le 2\exp\left(-\frac{2\left(\frac{\mathbb{E}[S]}{2}\right)^2}{4Tk^2}\right) \le 2\exp\left(-\frac{\left(T\phi(t)p^2\frac{\bar{w}^2 - w_{i^*}(t)^2}{2}\right)^2}{8Tk^2}\right)\\ &\le 2\exp\left(-\frac{T\left(\phi(t)p^2\frac{\bar{w}^2 - w_{i^*}(t)^2}{2}\right)^2}{32k^2}\right) \le 2\exp\left(-10\log n\right) \le \frac{1}{n^5},\end{aligned}$$

where the last inequality follows from

$$\begin{aligned}T\left(\phi(t)p^2\frac{\bar{w}^2 - w_{i^*}(t)^2}{2}\right)^2 &\ge T\left(\phi(t)p^2\frac{\bar{w}^2(1-(1-\delta)^2)}{2}\right)^2 \ge T\left(\phi(t)p^2\frac{\delta\bar{w}^2}{2}\right)^2\\ &\ge k^2\frac{2^{10}k^2\log n}{4p^6\delta^2}\phi(t)^2p^4\frac{\delta^2}{2k^2} \ge 320k^2\log n,\end{aligned}$$

where we used that $w_{i^*}(t) \le (1-2\delta)\bar{w}$. □

Thus, we have that

$$w_{i^*}(t+T) \ge w_{i^*} + \eta S \ge w_{i^*} + \eta\frac{\mathbb{E}[S]}{2} \ge w_{i^*} + \eta T\phi(t)p^2\frac{\bar{w}^2 - w_{i^*}(t)^2}{4}.$$

Proving that values close to $w_{max}(t)$ decrease in a similar fashion, is analogous.

Proof of Lemma B.1. By Lemma B.6, it takes at most $T\cdot\frac{4}{\eta T}$ examples to be shown for the potential ϕ to double. The the required number of samples for the potential to reach a value of $\Omega(\sqrt{p})$ is at most $O\left(\ell_{\max}\frac{4\log(k)}{\eta}\right)$. From there on, by Lemma B.7, we have for small $w_{i^*}(t)$,

$$w_{i^*}(t+T) - w_{i^*} \ge \eta T\phi(t)p^2\frac{\bar{w}^2 - w_{i^*}(t)^2}{4} \ge \eta T\phi(t)p^2\bar{w}\frac{\bar{w} - w_{i^*}(t)}{4} \ge \frac{\eta T\phi(t)p^2\bar{w}^2\delta}{4} \ge \frac{\eta T p^{2.5}\bar{w}^2\delta}{50}.$$

Hence, after showing another $\frac{50}{\eta p^{2.5}\bar{w}^2\delta}$ examples all $w_i, i \le k$ are $\bar{w}$ up to an additive error of 10δ. Note that $1/\bar{w}^2 = pk + 1 - p$. Using $p \ge r_2$ gives the desired bound on the number of examples required per concepts. Note that once the potential is within a multiplicative factor of $(1+10\delta)$, it will always remain (for $O(n^6)$ rounds) within a multiplicative factor of $(1+20\delta)$. □

B.3 Proof of Theorem 6.1

We are ready to prove the main theorem.

Proof of Theorem 6.1. As mentioned at the beginning of the section, it suffices to consider the learning of one concept. Generalizing to a concept hierarchy is analogous to the noise-free case.

We now argue how the learning of one concept follows from Lemma B.1. By Lemma B.1, all weights in F are at least $\frac{\bar{w}}{(1+20\delta)}$ and most $(1+20\delta)\bar{w}$. Hence, if $c \in supported_{r_2}(B)$, then we can show by a similar induction as in the proof of Theorem 5.3 that each rep fires since, the potential is at least $r_2 k \frac{\bar{w}}{(1+20\delta)} = \tau$, which means that the corresponding rep fires. On other other hand, if $c \notin supported_{r_1}(B)$, then there will be a neuron that does not fire since all weights are, by Lemma B.1, at most $(1+20\delta)\bar{w}$. Therefore, the potential for $rep(c)$ will be at most $r_1 k(1+20\delta)\bar{w} < r_2 k \frac{\bar{w}}{(1+20\delta)} = \tau$.

$$r_1 k(1+10\delta)\bar{w} + k^{\ell_{\max}} \frac{1}{k^{2\,\ell_{\max}}} < r_1 k(1+10\delta)\bar{w} - \frac{1}{\sqrt{k}} + \frac{1}{\sqrt{k}} \leq r_2 k \frac{\bar{w}}{(1+10\delta)} = \tau.$$

We note again that the results we used were for the process P' with bounded weight change, but as mentioned earlier, the event $\mathcal{E}$ implies a successful coupling of the original process P with P' holding with high probability. Thus, all the results also apply to the original process with high probability. Finally, applying union bound over all intervals yields the result.

□

C Lower Bounds Proofs

Proof of Theorem 7.1. Assume for contradiction that $\mathcal{N}$ recognizes $\mathcal{C}$. Let c denote any one of the concepts in C_2, i.e., a level 2 concept in C. Then c has k children, each of which has k children of its own, for a total of k^2 grandchildren.

Each of the k^2 grandchildren must have a rep in layer 0, but neither c nor any of its k children do, because layer 0 is reserved for level 0 concepts. So in particular, $rep(c)$ is a layer 1 neuron. By the structure of the network, this means that the only inputs to $rep(c)$ are from layer 0 neurons. Since we assume total connectivity, we have an edge from each layer 0 neuron to $rep(c)$. We define:

- $W(b)$, for each child b of c in the concept hierarchy: The total weight of all edges $(u, rep(c))$, where u is a layer 0 neuron that is the rep of a child of b.
- W: The total weight of all the edges $(u, rep(c))$, where u is a layer 0 neuron that is a rep of a grandchild of c. In other words, $W = \Sigma_{b \in children(c)} W(b)$.

We consider two scenarios. In Scenario A (the "must-fire scenario"), we choose input set B to consist of enough leaves of c to force $rep(c)$ to fire, that is, we ensure that $c \in supported_{r_2}(B)$, while trying to minimize the total weight incoming to $rep(c)$. Specifically, we choose the $r_2' k \geq r_2 k$ children b of c with the smallest values of $W(b)$. And for each such b, we choose its $r_2' k$ children with the smallest weights. Let B be the union of all of these $r_2' k$ sets of $r_2' k$ grandchildren of c. Since $r_2' \geq r_2 k$, it follows that $c \in supported_{r_2}(B)$.

Claim 1: In Scenario A, the total incoming potential to $rep(c)$ is at most $(r_2')^2 W$.

In Scenario B (the "can't-fire scenario"), we choose input set B to consist of leaves of c that force $rep(c)$ not to fire, that is, we ensure that $c \notin supported_{r_1}(B)$, while trying to maximize the total weight incoming to $rep(c)$. Specifically, we choose the $r_1' k < r_1 k$ children b of c with the largest values of $W(b)$, and we include all of their children in B. For each of the remaining $(1-r_1')k$ children of c, we choose its $r_1' k < r_1 k$ children with the largest weights and include them all in B. Since $r_1' k$ is strictly less than $r_1 k$, it follows that $c \notin supported_{r_1}(B)$.

Claim 2: In Scenario B, the total incoming potential to $rep(c)$ is at least $(r'_1)W + (1 - r'_1)r'_1W = 2r'_1W - (r'_1)^2W$.
Proof of Claim 2: We define:

- W_1: The total of the weights $W(b)$ for the r'_1k children b of c with the largest values of $W(b)$.
- $W_2 = W - W_1$: The total of the weights $W(b)$ for the remaining $(1 - r'_1)k$ children of c.
- W_3: We know that $W_1 \geq r'_1W$, since W_1 gives the total weight for the r'_1k children of c with the largest weights, out of k children. Define $W_3 = W_1 - r'_1W$; then W_3 must be nonnegative.

Then the total incoming potential to $rep(c)$ is

$$\begin{aligned}
&\geq W_1 + r'_1W_2, \\
&= r'_1W + W_3 + r'_1(W - W_1), \\
&= r'_1W + W_3 + r'_1(W - W_3 - r'_1W), \\
&= 2r'_1W - (r'_1)^2W + (1 - r'_1)W_3, \\
&\geq 2r'_1W - (r'_1)^2W,
\end{aligned}$$

as needed.
End of proof of Claim 2

Now, Claim 1 implies that the threshold τ of neuron $rep(c)$ must be at most $(r'_2)^2W$, since it must be small enough to permit the given B to trigger firing of $rep(c)$. On the other hand, Claim 2 implies that the threshold must be strictly greater than $2r'_1W - (r'_1)^2W$, since it must be large enough to prevent the given B from triggering firing of $rep(c)$. So we must have

$$2r'_1W - (r'_1)^2W < \tau \leq (r'_2)^2W.$$

But this contradicts our assumption that $(r'_2)^2 \leq 2r'_1 - (r'_1)^2$. □

Proof of Theorem 7.2. Assume for contradiction that $\mathcal{N}$ recognizes $\mathcal{C}$. Let c denote any one of the concepts in C_3, i.e., a level 3 concept in C. Then c has k children (level 2 concepts); each of which has k children of its own (level 1 concepts), so c has k^2 grandchildren. Moreover, each of these k^2 grandchildren also has k children of its own (level 0 concepts).

The proof involves a case analysis, based on which layers can contain *reps* of c, its children, and its grandchildren. First, because layer 0 is reserved for inputs, we have:
Claim 1: None of c, its children, or its grandchildren has its *rep* at layer 0.

In the next two claims, we use arguments similar to those in the proof of Theorem 7.1 to show that neither c nor any of its children can have its *rep* at level 1.
Claim 2: For each child b of c, $rep(b)$ is not in layer 1.
Proof of Claim 2: Suppose it is; then we can get a contradiction by arguing as in the proof of Theorem 7.1. In the modified proof, we argue about $rep(b)$, which is in layer 1, and the *reps* of its grandchildren, which are in layer 0. The two scenarios and the calculations are the same as before.
End of proof of Claim 2
Claim 3: $rep(c)$ is not in layer 1.
Proof of Claim 3: Suppose it is; then we can again get a contradiction by an argument similar to that for Theorem 7.1. But now we argue about $rep(c)$, which is in layer 1, and the *reps* of its great-grandchildren, which are in layer 0.

A difference is that this time, for each grandchild b' of c, we consider all of its children as a group. Thus, we define a "weight" for b', $W(b')$, equal to the sum of the weights of the edges from the (layer 0) neurons that are *reps* of children of b' to $rep(c)$, that is, the sum of the

weights of all edges $(u, rep(c))$ where $u \in rep(children(b'))$. Then for each child b of c, we define $W(b) = \Sigma_{b' \in children(b)} W(b')$, and define $W = \Sigma_{b \in children(c)} W(b)$. The rest of the proof proceeds as in Theorem 7.1.

End of proof of Claim 3

Having ruled out all other possibilities, the only remaining case is that $rep(c)$ is in layer 2, along with the *reps* of all its (level 2) children; so assume that this is the case. This means that $rep(c)$ cannot be affected by the firing of the *reps* of its children, but only (possibly) by the *reps* of its grandchildren and great-grandchildren.

Now the argument becomes somewhat different from before. For each (level 1) grandchild b' of c, consider what happens when exactly the the set of level 0 concepts $children(b')$ (which are great-grandchildren of c) is presented. This firing triggers some layer 1 neurons, say $N(b')$, to fire. These are not necessarily representations of anything in particular, just layer 1 neurons that happen to be triggered by this particular presentation. They might include $rep(b')$, and possibly other neurons.

Now define a weight that the grandchild b' contributes as input to $rep(c)$:

$$W(b') = \Sigma_{u \in N(b')} weight(u, rep(c)).$$

This is the total weight from all the layer 1 neurons that are triggered by the presentation of $children(b')$.

Next, consider any child b of c. Define a weight for b based on the total weight that its children (which are grandchildren of c) contribute to $rep(c)$:

$$W(b) = \Sigma_{b' \in children(b)} W(b').$$

We claim that this sum is the correct total weight incoming to $rep(c)$ when all of the concepts in $leaves(b)$ (the grandchildren of b) are presented at the same time. This follows from Assumption 4, the Noninterference Assumption, which says that no new layer 1 neurons are triggered to fire, other than those that are triggered by the presentation of separate sets $children(b')$.

Now define a weight for c based on the total weight that its children contribute to $rep(c)$:

$$W = \Sigma_{b \in children(c)} W(b).$$

We claim, again by Assumption 4, that this sum is the correct total weight incoming to $rep(c)$ when all of the concepts in $leaves(c)$ (the great-grandchildren of c) are presented at the same time.

The rest of the argument is analogous to the argument for Theorem 7.1. For the "must-fire" scenario, choose the $r'_2 k$ children b of c with the smallest values of $W(b)$, and for each of these, the $r'_2 k$ children b' with the smallest values of $W(b')$. Let the presented set B be the set of all the children b'' of these chosen b'. Then $c \in supported_{r_2}(B)$. Again using Assumption 4, we get:

Claim 4: The total weight of edges incoming to c that is triggered by presenting B is at most $(r'_2)^2 W$.

For the "can't-fire" scenario, choose the $r'_1 k < rk$ children b of c with the largest values of $W(b)$ and for each of these, all their children b'. And for the remaining $(1 - r'_1)k$ children b of c, choose their $r'_1 k < rk$ children b' with the largest weights. Let the presented set B be the set of all the children b'' of these chosen b'. Then $c \notin supported_{r_1}(B)$. We get:

Claim 5: The total weight of edges incoming to c that is triggered by presenting B is at least $(2r'_1 - (r'_1)^2)W$.

Combining Claims 4 and 5 yields, as before, that the threshold τ for c satisfies the extended inequality

$$2r'_1 W - (r'_1)^2 W < \tau \leq (r'_2)^2 W,$$

which contradicts our assumption that $(r'_2)^2 \leq 2r_1 - (r'_1)^2$. □

Proof of Lemma 7.4. We prove this by induction on ℓ. We use two base cases. For $\ell = 1$, the result is obvious because layer 0 is reserved for the level 0 concepts. For $\ell = 2$, suppose for the sake of contradiction that the claim in the theorem statement doesn't hold, that is, some level 2 concept c has $rep(c)$ at layer 1. Then again arguing as in the proof of Theorem 7.1, we get a contradiction.

For the inductive step, assume that $\ell \geq 3$, and assume the inductive claim holds for all levels $\leq \ell - 1$. Assume for contradiction that, for some c, $rep(c)$ is in some layer ℓ', $1 \leq \ell' \leq \ell - 1$. The inductive hypothesis implies that each concept $b \in children(c)$ has its rep at some layer $\geq \ell - 1$. This means that there are no connections from the $reps$ of children of c to $rep(c)$. So again, we consider c's grandchildren.

Thus, for each (level $\ell - 2 \geq 1$) grandchild b' of c, consider what happens when exactly the set $leaves(b')$ is presented. This firing triggers some layer $\ell' - 1$ neurons, say $N(b')$, to fire. Now define a weight that the grandchild b' contributes as input to $rep(c)$:

$$W(b') = \Sigma_{u \in N(b')} weight(u, rep(c)).$$

Next, consider any child b of c. Define a weight for b based on the total weight that its children (which are grandchildren of c) contribute to $rep(c)$:

$$W(b) = \Sigma_{b' \in children(b)} W(b').$$

We claim that this sum is the correct total weight incoming to $rep(c)$ when all of the concepts in $leaves(b)$ are presented at the same time. This follows from Assumption 4, the Noninterference Assumption.

Now define a weight for c based on the total weight that its children contribute to $rep(c)$:

$$W = \Sigma_{b \in children(c)} W(b).$$

Again by Assumption 4, this sum is the correct total weight incoming to $rep(c)$ when all of the concepts in $leaves(c)$ are presented at the same time.

The rest of the argument is analogous to those for Theorem 7.1 and Theorem 7.2. For the "must-fire" scenario, choose the $r_2'k$ children b of c with the smallest values of $W(b)$, and for each of these, the $r_2'k$ children b' with the smallest values of $W(b')$. Let the presented set B be the set of all the leaves of these chosen b'. Then $c \in supported_{r_2}(B)$, and the total weight of edges incoming to c that is triggered by presenting B is at most $(r_2')^2 W$ (using Assumption 4 once again). For the "can't-fire" scenario, choose the $r_1'k < r_1k$ children b of c with the largest values of $W(b)$ and for each of these, all their children b'. And for the remaining $(1 - r_1')k$ children b of c, choose their $r_1'k < r_1k$ children b' with the largest weights. Let the presented set B be the set of all the leaves of all these chosen b'. Then $c \notin supported_{r_1}(B)$, and the total weight of edges incoming to c that is triggered by presenting B is at least $(2r_1' - (r_1')^2)W$.

Combining these two claims yields, as before, that the threshold τ for c satisfies

$$2r_1'W - (r_1')^2 W < \tau \leq (r_2')^2 W,$$

which contradicts our assumption that $(r_2')^2 \leq 2r_1' - (r_1')^2$. □

D Auxiliary Claims

The following is a slight modification from Theorem 5.2 in [3].

Theorem D.1 (Azuma-Hoeffding inequality - general version [3]). *Let $Y_0, Y_1, \ldots$ be a martingale with respect ot the sequence $X_0, X_1, \ldots$. Suppose also that Y_i satisfies $a_i \leq Y_i - Y_{i-1} \leq b_i$ for all i. Then, for any t and n*

$$\mathbb{P}\left[\,|Y_n - Y_0| \geq t\,\right] \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$