

## MIT Open Access Articles

### *Learning hierarchically-structured concepts*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

**Citation:** Lynch, Nancy and Mallmann-Trenn, Frederik. 2021. "Learning hierarchically-structured concepts." *Neural Networks*, 143.

**As Published:** 10.1016/J.NEUNET.2021.07.033

**Publisher:** Elsevier BV

**Persistent URL:** <https://hdl.handle.net/1721.1/144297>

**Version:** Original manuscript: author's manuscript prior to formal peer review

**Terms of use:** Creative Commons Attribution-NonCommercial-NoDerivs License



---

# Learning Hierarchically-Structured Concepts

---

Nancy Lynch<sup>1</sup> and Frederik Mallmann-Trenn<sup>2</sup>

<sup>1</sup>Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

<sup>2</sup>King's College London, London, England

## Abstract

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

8 Then, we present a learning algorithm by which a layered network may learn  
9 to recognize hierarchical concepts according to our robust definition. We an-  
10alyze correctness and performance rigorously; the amount of time required to  
11 learn each concept, after learning all of the sub-concepts, is approximately  
12  $O\left(\frac{1}{\eta k} (\ell_{\max} \log(k) + \frac{1}{\varepsilon}) + b \log(k)\right)$ , where  $k$  is the number of sub-concepts  
13 per concept,  $\ell_{\max}$  is the maximum hierarchical depth,  $\eta$  is the learning rate,  $\varepsilon$   
14 describes the amount of uncertainty allowed in robust recognition, and  $b$  describes  
15 the amount of weight decrease for "irrelevant" edges. An interesting feature of this  
16 algorithm is that it allows the network to learn sub-concepts in a highly interleaved  
17 manner. This algorithm assumes that the concepts are presented in a noise-free  
18 way; we also extend these results to accommodate noise in the learning process.  
19 Finally, we give a simple lower bound saying that, in order to recognize concepts  
20 with hierarchical depth two with noise-tolerance, a neural network should have at  
21 least two layers.

22 The results in this paper represent first steps in the theoretical study of hierarchical  
23 concepts using SNNs. The cases studied here are basic, but they suggest many  
24 directions for extensions to more elaborate and realistic cases.

25 **Keywords:** Hierarchical Concepts, Representing Hierarchical Concepts, Recognizing Hierarchical  
26 Concepts, Learning Hierarchical Concepts, Spiking Neural Networks, Brain-Inspired Algorithms

## 27 1 Introduction

28 We are interested in the general problem of *how concepts that have structure are represented in the*  
29 *brain*. What do these representations look like? How are they learned, and how do the concepts  
30 get recognized after they are learned? We draw inspiration from recent experimental research on  
31 computer vision in convolutional neural networks (CNNs) by Zeiler and Fergus [54] and Zhou, et  
32 al. [55]. This research shows that CNNs learn to represent structure in visual concepts: lower layers  
33 of the network represent basic concepts and higher layers represent successively higher-level concepts.  
34 This observation is consistent with neuroscience research, which indicates that visual processing  
35 in mammalian brains is performed in a hierarchical way, starting from primitive notions such as  
36 position, light level, etc., and building toward complex objects; see, e.g., [15, 14, 7]. More generally,

37 we consider the thesis that *the structure that is naturally present in real-world concepts get mirrored*  
38 *in their brain representations, in some natural way that facilitates both learning and recognition.*

39 We approach this problem using ideas and techniques from theoretical computer science, distributed  
40 computing theory, and in particular, from recent work by Lynch, et al. on synchronous Spiking  
41 Neural Networks (SNNs) [28, 25, 27, 45, 13]. These papers began the development of an algorithmic  
42 theory of SNNs, developing formal foundations, and using them to study problems of attention and  
43 focus, neural representation, and short-term learning. Here we continue that general development, by  
44 initiating the study of long-term learning within the same framework.

45 We focus here on learning hierarchically-structured concepts. We capture these formally in terms of  
46 abstract *concept hierarchies*, in which concepts are built from lower-level concepts, which in turn are  
47 built from still-lower-level concepts, etc. Such structure is natural, e.g., for physical objects that are  
48 learned and recognized during human or computer visual processing. An example of such a hierarchy  
49 might be the following model of a *human*: A human consists of a *body*, a *head*, a *left leg*, a *right leg*,  
50 a *left arm*, and a *right arm*. Each of these concepts may consist of more concepts, allowing us to  
51 model a human to an arbitrary degree of granularity. Most concepts in the real world have additional  
52 structure, e.g., arms and legs are positioned symmetrically; however, we ignore such information for  
53 now and assume simply that each concept consists of sub-concepts. For this initial theoretical study,  
54 we make some additional simplifications: we fix a maximum level  $\ell_{\max}$  for concept hierarchies, we  
55 assume that all non-primitive concepts have the same number  $k$  of "child concepts", and we assume  
56 that our concept hierarchies are trees, i.e., there is no overlap in the composition of different concepts  
57 at the same level of a hierarchy. We expect that these assumptions can be removed or weakened, but  
58 it seems useful to start with the simplest case.

59 This paper demonstrates theoretically, in terms of simple hierarchies, how hierarchically-structured  
60 data can be represented, learned, and recognized in feed-forward layered Spiking Neural Networks.  
61 Specifically, we provide formal definitions for *concept hierarchies* and *layered neural networks*. We  
62 define precisely what it means for a layered neural network to *recognize* a particular concept in a  
63 concept hierarchy. Our notion of recognition is *robust*: a concept is required to be recognized if the  
64 input is close to the ideal concept, and is required not to be recognized if the input is far from the  
65 ideal. We also define what it means for a layered neural network to *learn to recognize* a concept  
66 hierarchy, according to our robust definition of recognition.

67 Next, we present two simple, efficient algorithms (layered networks) that learn to recognize concept  
68 hierarchies; the first assumes reliability during the learning process, whereas the second tolerates a  
69 bounded amount of noise. An example of such learning is shown in Figure 1. We also provide a  
70 preliminary lower bound, saying that, in order to robustly recognize concepts with hierarchical depth  
71 2, a neural network should have at least 2 layers. We discuss possible extensions of this bound to  
72 concepts with larger depth. We end with many directions for extending this work.

73 *Note:* We view this work as the first step in a general project to produce a theory for how logical  
74 concepts are represented, and learned, in the brain. Our general approach is to start with the simplest  
75 case, working out basic definitions, algorithms, and limitations for that case, and then to extend in  
76 many directions, step-by-step. We think such a stepwise approach will be effective in developing the  
77 theory. In addition, we hope that this first step, besides being of interest on its own, will provide a  
78 useful blueprint for later extensions.

79 **In more detail:** We describe our data model in Section 2. We assume a fixed maximum number  
80  $\ell_{\max}$  of levels in our concept hierarchies. Each concept hierarchy  $\mathcal{C}$  has a fixed set  $C$  of concepts,  
81 organized into levels  $\ell$ ,  $0 \leq \ell \leq \ell_{\max}$ . These are chosen from some universal set  $D$  of *concepts*.  
82 Each concept at each level  $\ell$ ,  $1 \leq \ell \leq \ell_{\max}$  has precisely  $k$  children, which are level  $\ell - 1$  concepts.  
83 We assume here that each concept hierarchy is a tree, that is, there is no overlap among the sets of  
84 children of different concepts. Each individual concept hierarchy represents the concepts and child  
85 relationships that arise in a particular execution of the network (or lifetime of an organism). However,  
86 the chosen concepts and their relationships may be different in different concept hierarchies. Again  
87 we note that these assumptions are a considerable simplification of reality, but we regard them as a  
88 good starting point.

---

<sup>1</sup>© Universal Color Slide Company.

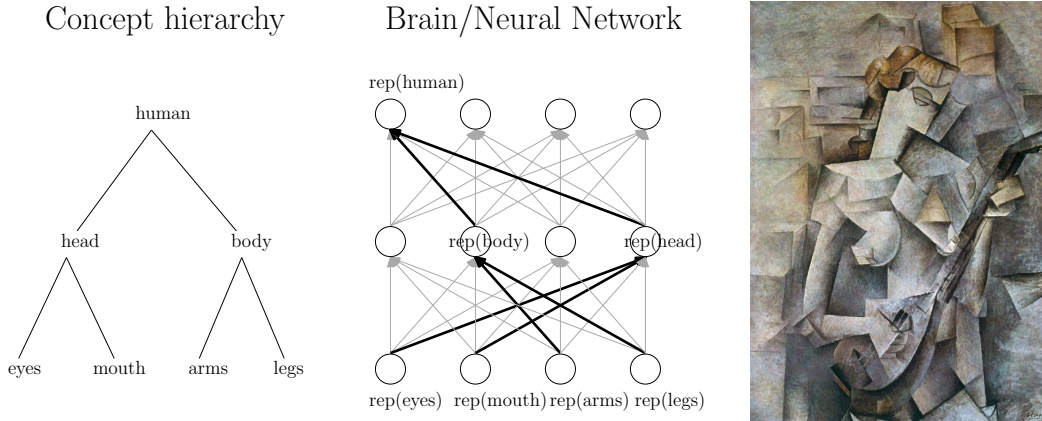


Figure 1: The leftmost figure shows the concept *human*, which consists of two sub-concepts, and so on. The second figure shows a network that has "learned" the concept "human" in the sense that, when the neurons representing the basic parts *eyes*, *mouth*, *arms*, *legs* are excited, then exactly one neuron  $u$  on the top layer will fire. Neuron  $u$  should also fire when "most" of the basic parts are excited, and  $u$  should not fire when few of the basic parts are excited. For example, the painting “Girl with a Mandolin” by Picasso<sup>1</sup> should cause  $u$  to fire despite the lack of a mouth and legs. The network accomplishes this by strengthening relevant synapses (bold edges) and weakening others (thin edges).

89 Next, in Section 3, we define a synchronous Spiking Neural Network model<sup>2</sup>, derived from the one  
 90 in [28, 27], but with additional structure to support learning. Namely, the new model incorporates edge  
 91 weights (representing synapse strengths) into neuron states; this provides a convenient way to describe  
 92 how those weights change during learning. We model learning using *Oja’s rule*, a biologically-  
 93 inspired rule that can be regarded as a mathematical formalization of Hebbian learning [18]. Oja’s  
 94 rule was first introduced in [35], and has since received considerable attention due its connections  
 95 with dimensionality reduction; see, for example, [36, 8]. Although there is no direct experimental  
 96 evidence yet that Oja’s precise rule is used in the brain, its core characteristics such as long-term  
 97 potentiation, long-term depression, and normalization are known to occur in brain networks, and  
 98 have been studied thoroughly (e.g., [2, 1]). Interestingly, to the best of our knowledge, Oja’s rule has  
 99 so far been studied only in "flat" settings, where the network has only one layer. Moreover, previous  
 100 work (e.g., [35]) has allowed the learning parameter  $\eta$  to be time-dependent, in order to achieve  
 101 convergence. In this paper, we consider the multilayer setting, and we show convergence with a fixed  
 102 learning rate.

103 In Section 4, we present our definitions for the robust recognition and noise-free learning problems.  
 104 Thus, we define how an SNN represents a concept hierarchy; here we use the simplifying assumption  
 105 that each concept is represented by just one neuron. We define what it means for an SNN to correctly  
 106 recognize a concept hierarchy, including situations in which the network is required to recognize a  
 107 concept  $c$  and situations where it is required not to do so. In particular, if a sufficiently large fraction  
 108  $r_2$  of the children of concept  $c$  are recognized, then  $c$  should be recognized, whereas if fewer than  
 109 a smaller fraction  $r_1$  of the children of  $c$  are recognized, then  $c$  should not be recognized. We also  
 110 define what it means for an SNN to learn to recognize a concept hierarchy, in the noise-free setting.

111 Then, in Section 5, we present algorithms that allow a network, starting from a default configuration,  
 112 to recognize and to learn the concepts in a particular concept hierarchy. Our algorithms are efficient,

<sup>2</sup>A word about our use of the Spiking Neural Network terminology: Our model here is simpler than typical SNN models, in that neuron actions depend just on the previous state and not on a longer history. In some of our prior work, such as [45], we use a more elaborate version of the model in which neurons actions can depend on bounded history. This is useful for capturing aspects of neuron processing such as accumulating potential. In future extensions of the present work, we expect to use such elaborations. We use the SNN terminology here in an attempt to keep the terminology consistent across our papers.

113 in terms of network size and running time. In particular, a network with max layer  $\ell_{\max}$  suffices to  
114 recognize a concept hierarchy with max level  $\ell_{\max}$ . Recognition happens within a very short time,  
115 proportional to the number of layers in the network. For learning, our algorithm converges reasonably  
116 quickly to a configuration that supports robust recognition. Our convergence time bound result for  
117 noise-free learning is [Theorem 5.3](#). Our algorithms require the examples to be shown several times  
118 and in a constrained order: roughly speaking, we require the network to "learn" the children of a  
119 concept  $c$  first, before examples of  $c$  are shown. Thus, in our running example, we require enough  
120 examples of "head", "body", etc. to be able to learn those concepts before the network sees them all  
121 together as "human". Except for this constraint, concepts may be shown in an arbitrarily interleaved  
122 manner. In [Section 6](#), we adapt our problem definitions and learning algorithm to a setting where  
123 the examples presented may be perturbed by noise. The modified algorithm still works, but now  
124 convergence requires the network to see more examples, compared to the noise-free case, as we show  
125 in [Theorem 6.4](#). The detailed analysis needed to prove [Theorems 5.3](#) and [6.4](#) appears in [Sections A](#)  
126 and [B](#), respectively.

127 Once we see that a network with max layer  $\ell_{\max}$  can easily learn and recognize any concept hierarchy  
128 with max level  $\ell_{\max}$ , it is natural to ask whether  $\ell_{\max}$  layers are actually necessary. Certainly these  
129 networks yield natural and efficient representations, but it is still interesting to ask the theoretical  
130 question of whether shallower networks could accomplish the same thing. In [Section 7](#), we give  
131 a preliminary lower bound result, showing that a two-layer concept hierarchy requires a two-layer  
132 network in order to solve the noisy recognition problem. We also discuss the possibility of extending  
133 this result to more levels and layers.

134 In summary, this paper is intended to show, using theoretical techniques, how structured concepts  
135 can be represented, recognized, and learned in biologically plausible neural networks. We give  
136 fundamental definitions and algorithms for particular types of concept hierarchies and networks. This  
137 represents a first step towards a theory of representation and learning for hierarchically-structured  
138 concepts in SNNs; it opens up many follow-on questions, which we discuss in [Section 8](#).

139 **Related work:** Immediate inspiration from this work came from experimental computer vision  
140 research on "network dissection" by Zhou, et al. [[55](#)]. This work describes experiments that show  
141 that unsupervised learning of visual concepts in deep convolutional neural networks results in  
142 "disentangled" representations. These include neural representations, not just for the main concepts  
143 of interest, but also for their components and sub-components, etc., throughout a concept hierarchy.  
144 As in this paper, they consider individual neurons as representations for individual concepts. They  
145 find that the representations that arise are generally arranged in layers so that more primitive concepts  
146 (colors, textures,...) appear at lower layers whereas more complex concepts (parts, objects, scenes)  
147 appear at higher layers. Earlier work by Zeiler and Fergus [[54](#)] made similar observations. As we  
148 described earlier, this work is consistent with neuroscience research, which indicates that visual  
149 processing in mammalian brains is performed hierarchically [[15](#), [14](#), [7](#)]. Some of this work indicates  
150 that the network includes feedback edges in addition to forward edges; the function of the feedback  
151 edges seems to be to solidify representations of lower-level objects based on context [[16](#), [33](#)]. While  
152 we do not yet address feedback edges in this paper, that is one of our main intended future directions.

153 Brain-like hierarchical models have been studied before (e.g., [[43](#)] and [[44](#)]). The authors of [[43](#)]  
154 propose a model consisting of different kinds of cells to model image recognition in the brain. Another  
155 biologically-motivated line of research concerns synfire chains, which are essentially a feed-forward  
156 network of neurons. These networks are a predecessor of spiking neural networks (SNNs). An  
157 interesting work in this field is [[44](#)], which studies a hierarchical organization of synfire chains.

158 The SNN model [[29](#), [30](#), [9](#), [17](#), [11](#)], upon which all of our neural algorithms research is based, is a  
159 model for neural computation that balances biological plausibility with theoretical tractability. Our  
160 work is influenced by research of Maass et al. [[30](#), [31](#), [32](#)] on the computational power of SNNs, and  
161 by that of Valiant [[47](#), [48](#), [49](#), [50](#)] on learning in the *neuroidal model* of brain computation. Recent  
162 research by Papadimitriou, et al. [[40](#), [42](#), [22](#), [41](#)] on problems of learning and association of concepts  
163 is another source of inspiration.

164 Oja's learning rule [[35](#), [36](#)]. is a biologically plausible local rule for adjusting synapse weights during  
165 learning. As mentioned earlier, to the best of our knowledge, Oja's rule has so far been studied only  
166 in single-layered networks and with time-dependent learning rates ([[35](#), [36](#), [8](#)]). Other related learning  
167 rules include Hebbian variants [[12](#), [23](#)] or BCM learning [[3](#)].

168 The learning algorithms in this paper utilize a *Winner-Take-All* sub-network [21, 53, 46, 4, 32, 51,  
 169 37, 24], to help in selecting which neurons to engage in learning. Winner-Take-All is an important  
 170 primitive in neural computation that is used to model visual attention and competitive learning.  
 171 Maybe "Note that such engagement of a neuron to learn is also known in some of neuroscience  
 172 literature eligibility traces (or synaptic flags); see [10] for experimental evidence of the existence of  
 173 eligibility traces.

174 Work by Mhaskar et al. [34] is related to ours in that they also consider embedding a tree-structured  
 175 concept hierarchy in a layered network. They also prove results saying that deep neural networks  
 176 are better than shallow networks at representing a deep concept hierarchy, However, their concept  
 177 hierarchies differ mathematically from ours, since they are formalized as compositional functions.  
 178 Also, their notion of representation is different, corresponding to function approximation, and their  
 179 proofs are based on approximation theory. Other related work appears in papers by Knoblauch  
 180 and collaborators, e.g., [6, 19, 39]. These papers describe experimental work involving hierarchical  
 181 concepts that are more general than ours (e.g., allowing overlap), networks that are more general  
 182 (e.g., allowing feedback), and more robust types of representations (cell assemblies). They present  
 183 this work in the context of an integrated robot system combining processing of visual and language  
 184 input, decisions, and action). For us, this provides good inspiration for future theoretical work.

185 **Acknowledgments:** We thank Brabebe Wang for helpful conversations and suggestions. we also  
 186 thank an anonymous referee for much constructive feedback, and many suggestions for interesting  
 187 extensions. The authors were supported in part by NSF Award Numbers CCF-1810758, CCF-  
 188 0939370, CCF-1461559, and CCF-2003830.

## 189 2 Data Model

190 In this section, we define an abstract notion of a *concept hierarchy*, which represents all the concepts  
 191 that arise in some particular "lifetime" of an organism, together with hierarchical relationships  
 192 between them. As noted above, our definition is restricted to tree-structured hierarchical relationships;  
 193 extensions are left for future work. We follow this with a definition for the notion of *support*, which  
 194 indicates which lowest-level concepts are sufficient to trigger the recognition of higher-level concepts.

### 195 2.1 Preliminaries

196 We begin by defining some general notation. First, we fix four constants:

- 197 •  $\ell_{\max}$ , a positive integer, representing the maximum level number for the concepts we  
 198 consider.
- 199 •  $n$ , a positive integer, representing the total number of lowest-level concepts.
- 200 •  $k$ , a positive integer, representing the number of top-level concepts in any concept hierarchy,  
 201 and also the number of sub-concepts for each concept that is not at the lowest level.<sup>3</sup>
- 202 •  $r_1, r_2$ , reals in  $[0, 1]$  with  $r_1 \leq r_2$ ; these represent thresholds for noisy recognition.

203 We assume a predetermined universal set  $D$  of *concepts*, partitioned into disjoint sets  $D_\ell, 0 \leq \ell \leq$   
 204  $\ell_{\max}$ . We refer to any particular concept  $c \in D_\ell$  as a *level  $\ell$  concept*, and write  $level(c) = \ell$ .  
 205 Here,  $D_0$  represents the most basic concepts and  $D_{\ell_{\max}}$  the highest-level concepts. We assume that  
 206  $|D_0| = n$ .

### 207 2.2 Concept hierarchies

208 A *concept hierarchy*  $\mathcal{C}$  consists of a subset  $C$  of  $D$ , together with a *children* function. For each  
 209  $\ell, 0 \leq \ell \leq \ell_{\max}$ , we define  $C_\ell$  to be  $C \cap D_\ell$ , that is, the set of level  $\ell$  concepts in  $\mathcal{C}$ . For each  
 210 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  
 211  $c' \in children(c)$  a *child* of  $c$ . We require the following three properties.

- 212 1.  $|C_{\ell_{\max}}| = k$ .

---

<sup>3</sup>Assuming the same number  $k$  throughout is a simplification of what would be needed for applications; it should be easy to generalize this.

213 2. For any  $c \in C_\ell$ , where  $1 \leq \ell \leq \ell_{\max}$ , we have that  $|\text{children}(c)| = k$ ; that is, the degree of  
 214 any internal node in the concept hierarchy is exactly  $k$ .

215 3. For any two distinct concepts  $c$  and  $c'$  in  $C_\ell$ , where  $1 \leq \ell \leq \ell_{\max}$ , we have that  
 216  $\text{children}(c) \cap \text{children}(c') = \emptyset$ ; that is, the sets of children of different concepts at  
 217 the same level are disjoint.

218 It follows that  $C$  is a forest with  $k$  roots and height  $\ell_{\max}$ . Also, for any  $\ell, 0 \leq \ell \leq \ell_{\max}$ ,  $|C_\ell| =$   
 219  $k^{\ell_{\max} - \ell + 1}$ . Note that our notion of concept hierarchies is quite restrictive, in that we allow no overlap  
 220 between the sets of children of different concepts. Allowing overlap is an important next direction for  
 221 future work.

222 We extend the *children* notation recursively by defining a concept  $c'$  to be a *descendant* of a  
 223 concept  $c$  if either  $c' = c$ , or  $c'$  is a child of a descendant of  $c$ . We write  $\text{descendants}(c)$  for the set  
 224 of descendants of  $c$ . Let  $\text{leaves}(c) = \text{descendants}(c) \cap C_0$ , that is, all the level 0 descendants of  $c$ .

### 225 2.3 Support

226 Now we give a key definition that indicates which lowest-level concepts should be sufficient to trigger  
 227 recognition of higher-level concepts.

228 We fix a particular concept hierarchy  $C$ , with its concept set  $C$  partitioned into  $C_0, \dots, C_{\ell_{\max}}$ . For  
 229 any given subset  $B$  of the general set  $D_0$  of level 0 concepts, and any real number  $r \in [0, 1]$ , we  
 230 define a set  $\text{supported}_r(B)$  of concepts in  $C$ . This represents the set of concepts  $c \in C$ , at all levels,  
 231 that have enough of their leaves present in  $B$  to support recognition of  $c$ . The notion of "enough"  
 232 here is defined recursively, based on having an  $r$ -fraction of children supported at every level.

233 **Definition 2.1 (Supported).** Given  $B \subseteq D_0$ , define the following sets of concepts at all levels,  
 234 recursively:

- 235 1.  $B_0 = B \cap C_0$ . That is, we restrict attention to just the level 0 concepts in  $C$ .
- 236 2.  $B_1$  is the set of all concepts  $c \in C_1$  such that  $|\text{children}(c) \cap B_0| \geq rk$ . That is, we consider  
 237 the level 1 concepts in  $C$  for which at least an  $r$ -fraction of their children appear in  $B_0$ .
- 238 3. For  $2 \leq \ell \leq \ell_{\max}$ ,  $B_\ell$  is the set of all concepts  $c \in C_\ell$  such that  $|\text{children}(c) \cap B_{\ell-1}| \geq rk$ .  
 239 That is, we consider the level  $\ell$  concepts in  $C$  for which at least an  $r$ -fraction of their children  
 240 appear in  $B_{\ell-1}$ .

241 Define  $\text{supported}_r(B)$  to be  $\bigcup_{0 \leq \ell \leq \ell_{\max}} B_\ell$ . We sometimes also write  $\text{supported}_r(B, \ell)$  for  $B_\ell$ .

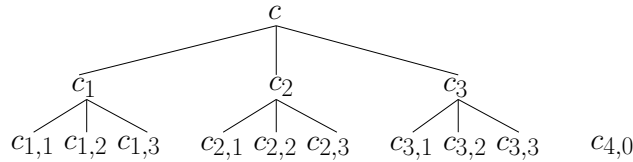


Figure 2: This example illustrates the  $\text{supported}_r(B)$  definition, with  $k = 3$  and  $r = \frac{2}{3}$ . We depict just a single level 2 concept  $c$  with children  $c_1, c_2, c_3$  and grandchildren  $c_{1,1}, c_{1,2}, c_{1,3}, c_{2,1}, c_{2,2}, c_{2,3}, c_{3,1}, c_{3,2}, c_{3,3}$ . The set  $B$  consists of concepts  $c_{1,1}, c_{1,2}, c_{3,1}, c_{3,3}$  plus an "extra" concept  $c_{4,0}$  that is not a descendant of  $c$ . Then  $B_0 = \{c_{1,1}, c_{1,2}, c_{3,1}, c_{3,3}\}$ ,  $B_1 = \{c_1, c_3\}$ , and  $B_2 = \{c\}$ .

242 The special case  $r = 1$  is important as it corresponds to a "noise-free" notion of support, in which all  
 243 the leaves of a concept must be present. That is:

244 **Lemma 2.2.** For any  $B \subseteq D_0$ ,  $\text{supported}_1(B)$  is the set of all concepts  $c \in C$  (at all levels) such  
 245 that  $\text{leaves}(c) \subseteq B$ .

246 **3 Network Model**

247 In this section, we define our network model. We first describe the network structure, then the  
 248 individual neurons, and finally the operation of the overall network.

249 **3.1 Preliminaries**

250 We introduce four constants:

- 251 •  $\ell'_{max}$ , a positive integer, representing the maximum number of a layer in the network.
- 252 •  $n$ , a positive integer, representing the number of distinct inputs the network can handle. This  
 253 is the same  $n$  as in the data model, where it represents the total number of level 0 concepts  
 254 in a concept hierarchy.
- 255 •  $\tau$ , a real number, representing the firing threshold for neurons.
- 256 •  $\eta$ , a positive real, representing the learning rate for our learning rule.

257 **3.2 Network structure**

258 Our networks are directed graphs consisting of neurons arranged in layers, with edges directed from  
 259 each layer to the next-higher layer; thus, they are feed-forward layered neural networks.

260 Specifically, a network  $\mathcal{N}$  consists of a set  $N$  of neurons, partitioned into disjoint sets  $N_\ell, 0 \leq \ell \leq$   
 261  $\ell'_{max}$ , which we call *layers*. We refer to any particular neuron  $u \in N_\ell$  as a *layer  $\ell$  neuron*, and  
 262 write  $layer(u) = \ell$ . We assume (for simplicity) that each layer contains exactly  $n$  neurons, that is,  
 263  $|N_\ell| = n$  for every  $\ell$ . We refer to the  $n$  layer 0 neurons as *input neurons* and to all other neurons as  
 264 *non-input neurons*. We assume total connectivity between successive layers, that is, each neuron in  
 265  $N_\ell, 0 \leq \ell \leq \ell'_{max} - 1$  has an outgoing edge to each neuron in  $N_{\ell+1}$ , and these are the only edges.

266 We assume a one-to-one mapping  $rep : D_0 \rightarrow N_0$ , where  $rep(c)$  is the neuron corresponding to  
 267 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  
 268 set of layer 0 neurons. This will allow the network to receive an input corresponding to any level 0  
 concept. See Figure 3 for a depiction.

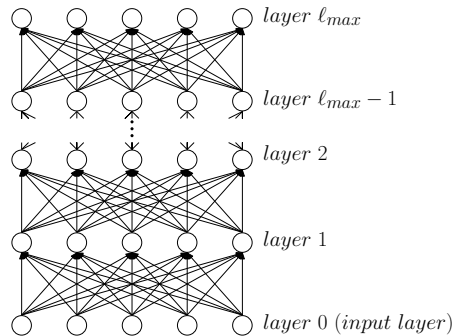


Figure 3: The figure depicts the general structure of a feed-forward network.

269

270 We "lift" the definition of  $rep$  to sets of level 0 concepts as follows: For any  $B \subseteq D_0$ , we define  
 271  $rep(B) = \{rep(b)|b \in B\}$ . That is,  $rep(B)$  is the set of all *reps* of concepts in  $B$ . (We will use  
 272 analogous "lifting" definitions to extend other functions to sets.)

273 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  $n$   
 274 neurons of  $N_0$ , it follows that  $n \geq k^{\ell_{max} + 1}$ . However, we imagine that  $n$  is much larger than this,  
 275 because we imagine that the total number of possible level 0 concepts is much larger than the number  
 276 that will arise in any particular execution of the network.

277 In Section 4, we will consider extensions of the  $rep()$  function from level 0 concepts to higher-level  
 278 concepts. Establishing such higher-level *reps* will be the job of a learning algorithm.



279 **3.3 Neuron states**

280 We assume that the state of each neuron consists of several *state components*. Here we distinguish  
 281 between input neurons and non-input neurons. Namely, each input neuron  $u \in N_0$  has just one state  
 282 component:

- 283 • *firing*, with values in  $\{0, 1\}$ ; this indicates whether or not the input neuron is currently firing.

284 We denote the *firing* component of input neuron  $u$  at integer time  $t$  by  $firing^u(t)$ ; we will sometimes  
 285 abbreviate this in mathematical formulas as just  $y^u(t)$ .

286 Each non-input neuron  $u \in N_\ell$ ,  $1 \leq \ell \leq \ell'_{max}$ , has three state components:

- 287 • *firing*, with values in  $\{0, 1\}$ , indicating whether the neuron is currently firing.
- 288 • *weight*, a real-valued column vector in  $[0, 1]^n$  representing current weights on incoming  
 289 edges.
- 290 • *engaged*, with values in  $\{0, 1\}$ ; indicating whether the neuron is currently prepared to learn.  
 291 As discussed in the intro, these model eligibility traces (see [10]).

292 We denote the three components of non-input neuron  $u$  at time  $t$  by  $firing^u(t)$ ,  $weight^u(t)$ , and  
 293  $engaged^u(t)$ , respectively, and abbreviate these by  $y^u(t)$ ,  $w^u(t)$ , and  $e^u(t)$ .

294 We also use the notation  $x^u(t)$  to denote the column vector of *firing* flags of  $u$ 's incoming neighbor  
 295 neurons at time  $t$ . That is,  $x^u(t) = [y^{v_1}(t)y^{v_2}(t) \dots y^{v_n}(t)]^T$ , where  $\{v_i\}_{i \leq n}$  are the incoming  
 296 neighbors of  $u$ , which are exactly all the nodes in the layer below  $u$ .

297 **3.4 Neuron transitions**

298 Now we describe neuron behavior, specifically, we describe how to determine the values of the state  
 299 components of each neuron  $u$  at time  $t \geq 1$  based on values of state components at the previous time  
 300  $t - 1$  and on external inputs. Again, we distinguish between input neurons and non-input neurons.

301 **Input neurons:** If  $u$  is an *input neuron*, then it has only one state component, the *firing* flag.  
 302 Since  $u$  is an input neuron, we assume that the value of the *firing* flag is controlled by the network's  
 303 environment and not by the network itself, that is, the value of  $y^u(t)$  is set by some external input  
 304 signal, which we do not model explicitly.

305 **Non-input neurons:** If  $u$  is a *non-input neuron*, then it has three state components, *firing*, *weight*,  
 306 and *engaged*. Whether or not neuron  $u$  fires at time  $t$ , that is, the value of  $y^u(t)$ , is determined by its  
 307 incoming *potential* and its *activation function*.

308 The potential at time  $t$ , which we denote by  $pot^u(t)$  is given by the dot product of the weights and  
 309 inputs at neuron  $u$  at time  $t - 1$ , that is,

$$pot^u(t) = w^u(t-1)^T \cdot x^u(t-1) = \sum_{j=1}^n w_j^u(t-1)x_j^u(t-1).$$

310 The activation function, which defines whether or not neuron  $u$  fires at time  $t$ , is then defined by:

$$y^u(t) = \begin{cases} 1 & \text{if } pot^u(t) \geq \tau, \\ 0 & \text{otherwise,} \end{cases}.$$

311 where  $\tau$  is the assumed firing threshold.

312 We assume that the value of the *engaged* flag of  $u$  is controlled by  $u$ 's environment, that is, for  
 313 every  $t$ , the value of  $e^u(t)$  is set by some input signal, which may arise from outside the network  
 314 or from another part of the network. For example, the *engaged* flag could be used to ensure that,  
 315 in any round, only one neuron is prepared to learn.<sup>4</sup> This neuron might be selected by a separate  
 316 "Winner-Take-All" sub-network.

<sup>4</sup>We use the term "round" to represent the activity between two consecutive times. In particular, "round  $t$ " refers to the activity that takes the system from time  $t - 1$  to time  $t$ . Thus, the potential in round  $t$  means the same thing as the potential at time  $t$ , captured by  $pot^u(t)$ .

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

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

319 where  $\eta$  is the assumed learning rate and  $z(t-1) = \text{pot}^u(t)$ .<sup>5</sup> Thus, the weight vector is adjusted  
 320 by an additive amount that is proportional to the learning rate and the potential, and depends on the  
 321 input firing pattern, with a negative adjustment that depends on the potential and the prior weights.

### 322 3.5 Network operation

323 During execution, the network proceeds through a sequence of *configurations*,  
 324  $\text{Con}(0), \text{Con}(1), \text{Con}(2), \dots$ , where  $\text{Con}(t)$  describes the configuration at nonnegative inte-  
 325 ger time  $t$ . Each configuration specifies a state for every neuron in the network, that is, values for all  
 326 the state components of every neuron.

327 As described above, the  $y$  values for the input neurons are specified by some external source. The  $y$ ,  
 328  $w$ , and  $e$  values for the non-input neurons are defined by the network specification at time  $t = 0$ . For  
 329 times  $t > 0$ , the  $y$  and  $w$  values are determined by the activation and learning functions described  
 330 above. The  $e$  values (engagement flags) are determined by special inputs arriving from outside the  
 331 network or from other sub-networks. In our algorithms in Sections 5.2 and 6.2, they will arrive from  
 332 Winner-Take-All sub-networks.

## 333 4 Problem Statements

334 In this section we define our two main problems: *recognizing concept hierarchies*, and *learning to*  
 335 *recognize concept hierarchies*. Our notion of recognition is robust to a bounded amount of noise. The  
 336 notion of learning we define in this section corresponds to noise-free learning; we extend this to noisy  
 337 learning in Section 6. In all cases, we assume that each item is represented by exactly one neuron;  
 338 considering more elaborate representations is another direction for future work.

### 339 4.1 Preliminaries

340 Throughout this section, we fix constants  $\ell_{\max}$ ,  $n$ ,  $k$ ,  $r_1$ , and  $r_2$  according to the definitions for a  
 341 concept hierarchy in Section 2. We consider a concept hierarchy  $\mathcal{C}$ , with concept set  $C$  and maximum  
 342 level  $\ell_{\max}$ , partitioned as usual into  $C_0, C_1, \dots, C_{\ell_{\max}}$ . We also fix constants  $\ell'_{\max}$ ,  $n$ ,  $\tau$ , and  $\eta$  as in  
 343 the definitions for a network in Section 3, and consider a network  $\mathcal{N}$  as described earlier. Thus, we  
 344 allow the maximum layer number  $\ell'_{\max}$  for  $\mathcal{N}$  to be different from the maximum level number  $\ell_{\max}$   
 345 for  $\mathcal{C}$ , but the number  $n$  of input neurons is the same as the number of level 0 items in  $\mathcal{C}$ .

346 The following definition will be useful in defining our recognition and learning problems. It expresses  
 347 what it means for a particular subset  $B$  of the level 0 concepts to be "presented" as input to the  
 348 network, at a certain time  $t$ .

349 **Definition 4.1 (Presented).** *If  $B \subseteq D_0$  and  $t$  is a non-negative integer, then we say that  $B$  is*  
 350 *presented at time  $t$  (in some particular execution) if, for every layer 0 neuron  $u$ , the following hold:*

- 351 1. *If  $u \in \text{rep}(B)$  then  $y^u(t) = 1$ .*
- 352 2. *If  $u \notin \text{rep}(B)$  then  $y^u(t) = 0$ .*

353 *That is, all of the layer 0 neurons in  $\text{rep}(B)$  fire at time  $t$ , and no other layer 0 neuron fires at time  $t$ .*

### 354 4.2 Robust recognition

355 Here we define what it means for network  $\mathcal{N}$  to recognize concept hierarchy  $\mathcal{C}$ . We assume that  
 356 every concept  $c \in C$ , at every level, has a unique representing neuron,  $\text{rep}(c)$ ; this extends the  $\text{rep}()$   
 357 function from level 0 concepts to higher-level concepts. For this definition, we also assume that,

---

<sup>5</sup>The  $z(t-1)$  notation is standard for Oja’s rule, so we use that in the rest of this paper when we analyze network behavior based on this rule.

358 during the entire recognition process, the *engaged* flags of all neurons are off, i.e., for every neuron  
 359  $u$  with  $layer(u) > 0$ , and every  $t$ ,  $e^u(t) = 0$ .

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

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

366 Then:

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

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

371 The special case of  $(1, 1)$ -recognition is interesting, since it is equivalent to the requirement that all  
 372 level 0 descendants of a concept must be present for recognition:

373 **Lemma 4.3.** Network  $\mathcal{N}(1, 1)$ -recognizes a concept  $c$  in concept hierarchy  $\mathcal{C}$  if and only if  $\mathcal{N}$   
 374 contains a unique neuron  $rep(c)$  such that the following holds. If  $B \subseteq D_0$  is presented at time  $t$ , then  
 375  $rep(c)$  fires at time  $t + layer(rep(c))$  if and only if  $leaves(c) \subseteq B$ .

376 *Proof.* By the definition of the robust recognition problem and [Lemma 2.2](#). □

### 377 4.3 Noise-free learning

378 In the learning problem, the network does not know ahead of time which particular concept hierarchy  
 379 might be presented in a particular execution. It must be capable of learning *any* concept hierarchy.

380 In our algorithm in [Section 5.2](#), in order for the network to learn a concept hierarchy  $\mathcal{C}$ , it must receive  
 381 inputs corresponding to all the concepts in  $\mathcal{C}$ . Here we define how individual concepts are "shown" to  
 382 the network, and then give constraints on the order in which the concepts are shown. Such constraints  
 383 are captured by the notion of a *bottom-up training schedule*. Then we state our learning guarantees,  
 384 assuming a bottom-up training schedule for  $\mathcal{C}$ .

385 We begin by describing how an individual concept  $c$  is "shown" to the network. Recall that  $leaves(c)$   
 386 is defined to be  $descendants(c) \cap C_0$ .

387 **Definition 4.4 (Showing a concept).** Concept  $c$  is shown at time  $t$  provided that the set  $B =$   
 388  $leaves(c)$  is presented at time  $t$ . That is, for every input neuron  $u$ ,  $y^u(t) = 1$  if and only if  
 389  $u \in rep(leaves(c))$ .

390 Learning a concept hierarchy will involve showing all the concepts in the hierarchy. Informally  
 391 speaking, we assume that the concepts are shown "bottom-up". For example, before the network is  
 392 shown the concept of a head, it is shown the lower-level concepts of mouth, eye, etc. And before  
 393 it is shown the concept of a human, it is shown the lower-level concepts of head, body, legs, etc.  
 394 More precisely, to enable network  $\mathcal{N}$  to learn the concept hierarchy  $\mathcal{C}$ , we assume that every concept  
 395 in its concept set  $\mathcal{C}$  is shown at least  $\sigma$  times, where  $\sigma$  is a parameter to be specified by a learning  
 396 algorithm. Furthermore, we assume that any concept  $c \in \mathcal{C}$  is shown only after each child of  $c$  has  
 397 been shown at least  $\sigma$  times. We allow the concepts to be shown in an arbitrary order and in an  
 398 interleaved manner, provided that these constraints are observed.

399 **Definition 4.5 ( $\sigma$ -bottom-up training schedule).** A training schedule for  $\mathcal{C}$  is any finite list  
 400  $c_0, c_1, \dots, c_m$  of concepts in  $\mathcal{C}$ , possibly with repeats. A training schedule is  $\sigma$ -bottom-up, where  
 401  $\sigma$  is a positive integer, provided that each concept in  $\mathcal{C}$  appears in the list at least  $\sigma$  times, and no  
 402 concept in  $\mathcal{C}$  appears before each of its children has appeared at least  $\sigma$  times.

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

405 **Definition 4.6** ( $(r_1, r_2, \sigma)$ -learning). Network  $\mathcal{N}(r_1, r_2, \sigma)$ -learns concept hierarchy  $\mathcal{C}$  provided  
 406 that the following holds. At any time after a training phase in which all the concepts of  $\mathcal{C}$  are shown  
 407 according to a  $\sigma$ -bottom-up training schedule, network  $\mathcal{N}(r_1, r_2)$ -recognizes  $\mathcal{C}$ .

## 408 5 Algorithms for Recognition and Noise-Free Learning

409 We give algorithms for both of the problems described in [Section 4](#).

### 410 5.1 Recognition

411 Fix a concept hierarchy  $\mathcal{C}$  with concept set  $C$ , and  $r_1, r_2 \in [0, 1]$ , with  $r_1 \leq r_2$ . Recognition can  
 412 be achieved by simply embedding the digraph induced by  $\mathcal{C}$  in the network  $\mathcal{N}$ . See [Figure 1](#) for an  
 413 illustration. For every  $\ell$  and for every level  $\ell$  concept  $c$  of  $\mathcal{C}$ , we designate a unique representative  
 414  $rep(c)$  in layer  $\ell$  of the network. Let  $R$  be the set of all representatives, that is,  $R = rep(\mathcal{C}) =$   
 415  $\{rep(c) \mid c \in C\}$ . We use  $rep^{-1}$  with support  $R$  to denote the corresponding inverse function that  
 416 gives, for every  $u \in R$ , the unique concept  $c \in C$  with  $rep(c) = u$ .

417 If  $u$  is a layer  $\ell$  neuron and  $v$  is a layer  $\ell + 1$  neuron, then we define the edge weight  $weight(u, v)$  by:

$$weight(u, v) = \begin{cases} 1 & \text{if } rep^{-1}(v) \in children(rep^{-1}(u)), \\ 0 & \text{otherwise.} \end{cases}$$

418 That is, we define the weights of edges corresponding to child relationships in the concept hierarchy  
 419 to be 1, and the weights of other edges to be 0.

420 Finally, we set the threshold  $\tau$  for every non-input neuron to be  $\frac{(r_1+r_2)k}{2}$ . It should be clear that the  
 421 resulting network  $\mathcal{N}$  solves the  $(r_1, r_2)$ -recognition problem:

422 **Theorem 5.1.** Network  $\mathcal{N}(r_1, r_2)$ -recognizes  $\mathcal{C}$ .

423 Recall that the definition of recognition, [Definition 4.2](#) says that each individual concepts  $c$  in  
 424 the hierarchy is recognized. For a level  $\ell$  concept  $c$ , the definition includes a time bound of  
 425  $layer(rep(c)) = level(c) = \ell$  for recognizing concept  $c$ .

426 We note that our choice of weights in  $\{0, 1\}$  here is for simplicity. Other combinations are possible,  
 427 and in fact, our learning algorithm below results in different weights, approximating  $\frac{1}{\sqrt{k}}$  and 0.

### 428 5.2 Noise-free learning

429 Now we move from the simple recognition problem to the harder problem of learning. Now we  
 430 must design a network  $\mathcal{N}$  that can learn an arbitrary concept hierarchy  $\mathcal{C}$  with parameters as listed  
 431 in [Section 2](#) and [Section 3](#), and with  $\ell_{\max} \leq \ell'_{\max}$ . Our algorithm utilizes *Winner-Take-All (WTA)*  
 432 sub-networks [[21](#), [53](#), [46](#), [4](#), [32](#), [51](#), [37](#), [24](#)].

433 **Winner-Take-All sub-networks:** Our algorithm uses Winner-Take-All sub-networks to select  
 434 which neurons are prepared to learn at different points during the learning process. In this paper,  
 435 we abstract from these sub-networks by simply describing their effects on the *engaged* flags in the  
 436 non-input neurons. We give the precise requirements in [Assumption 5.2](#).

437 While the network is being trained, example concepts are "shown" to the network, one example at  
 438 each time  $t$ , according to a  $\sigma$ -bottom-up training schedule as defined in [Section 4.3](#). We assume  
 439 that, for every example concept  $c$  that is shown, exactly one neuron at the appropriate layer will  
 440 be engaged; this layer is the one with the same number as the level of  $c$  in the concept hierarchy.  
 441 Furthermore, the neuron on that layer that is engaged is the one that has the largest potential  $pot^u$ .  
 442 More precisely, in terms of timing, we assume:

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

447 **Main algorithm:** We assume that the network  $\mathcal{N}$  starts in a clean state in which, for every neuron  
448  $u$  in layer 1 or higher,  $w^u(0) = \frac{1}{k^{\ell_{\max}+1}} \mathbf{1}$ , where  $\mathbf{1}$  is the  $n$ -dimensional all-one vector. We set the  
449 threshold  $\tau$  for all neurons to be  $\frac{(r_1+r_2)\sqrt{k}}{2}$ , and the learning rate  $\eta$  to be  $\frac{1}{4k}$ . The initial condition,  
450 threshold, learning rate, [Assumption 5.2](#), and the general model conventions for activation and  
451 learning suffice to determine how the network behaves, when shown a particular series of concepts.  
452 Our main result is:

453 **Theorem 5.3 (Noise-Free Learning Theorem).** *Let  $\mathcal{N}$  be the network described above, with maxi-*  
454 *imum layer  $\ell'_{\max}$ . Let  $b$  be an arbitrary positive real  $\geq 2$ . Let  $r_1, r_2$  be reals with  $0 < r_1 < r_2 \leq 1$ ;*  
455 *assume that  $r_1 k$  is not an integer, and  $r_1 k - \lfloor r_1 k \rfloor \geq \frac{\sqrt{k}}{k^{b-1}}$ . Also assume that  $r_2$  and  $k$  satisfy the*  
456 *inequality  $\frac{1}{\sqrt{k}} + \frac{1}{k} \leq \frac{r_2 \sqrt{k}}{2}$ .<sup>6</sup> Let  $\varepsilon = \frac{r_2 - r_1}{r_1 + r_2}$ .*

457 *Let  $\mathcal{C}$  be any concept hierarchy, with maximum level  $\ell_{\max} \leq \ell'_{\max}$ . Let  $\sigma = \frac{4}{3\eta k} ((\ell_{\max} + 1) \log(k)) +$   
458  $\frac{3}{\eta k \varepsilon} + \frac{b \log(k)}{\log(\frac{10}{9})}$ . Thus,  $\sigma$  is  $O\left(\frac{1}{\eta k} (\ell_{\max} \log(k) + \frac{1}{\varepsilon}) + b \log(k)\right)$ .*

459 *Then  $\mathcal{N}(r_1, r_2, \sigma)$ -learns concept hierarchy  $\mathcal{C}$ .*

460 That is, unwinding the definition of  $(r_1, r_2, \sigma)$ -learning, at any time after a training phase in which  
461 all the concepts of  $\mathcal{C}$  are shown according to a  $\sigma$ -bottom-up training schedule, network  $\mathcal{N}(r_1, r_2)$ -  
462 recognizes  $\mathcal{C}$ .

463 A rigorous analysis can be found in [Appendix A](#); the main idea of the analysis is as follows. We first  
464 prove some direct consequences of Oja's rule ([Lemma A.1](#), [Lemma A.2](#), and [Lemma A.3](#)). These  
465 quantify the weight changes for a single neuron involved in learning a single concept, assuming  
466 that all of its child concepts have already been learned. In particular, we show that the weights  
467 change quickly so that they approximate either  $1/\sqrt{k}$  or 0, depending on whether or not the weights  
468 correspond to neurons that represent child concepts.

469 We next build on these lemmas to describe, in [Lemma A.6](#), the learning (i.e., weight changes) that  
470 occur throughout the network in the course of the entire execution. What makes this challenging is  
471 that we allow "incomparable" concepts to be shown in an interleaved manner; the only constraint is  
472 that, for every concept  $c$ , child concepts of a concept  $c$  must be shown sufficiently many times before  
473  $c$  is shown. In order to prove that all concepts are learned correctly despite these challenges, we use  
474 an involved yet elegant five-part induction. Finally, in [Section A.3](#) we put everything together and  
475 show that the network successfully  $(r_1, r_2, \sigma)$ -learns the concept hierarchy.

## 476 6 Extension to Noisy Learning

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

481 To model this, we assume that, in order to show a concept  $c$ , we show a random  $p$ -fraction of its  
482 sub-concepts. Formally, we use the following recursive marking procedure to determine which inputs  
483 should be presented to the network: We begin by marking  $c$ . Then, proceeding recursively, for any  
484 marked concept, we mark a random  $p$ -fraction of the sub-concepts. The recursion terminates when a  
485 subset of the leaves of  $c$  are marked. The inputs presented to the network are the representations of  
486 the marked leaves of  $c$ .

### 487 6.1 Modifications to the model

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

490 **Definition 6.1 ( $p$ -noisy-showing a concept).** *Concept  $c$  is  $p$ -noisy-shown at time  $t$ , where  $p \in (0, 1]$ ,*  
491 *provided that a subset  $B \subseteq \text{leaves}(c)$  produced by the random function  $\text{mark}(c, p)$  is presented at*

<sup>6</sup>This last assumption can be satisfied by a variety of different combinations of assumptions on  $r_2$  and  $k$  individually, such as  $r_2 \geq \frac{1}{2}$  and  $k \geq 6$ , or  $r_2 \geq \frac{1}{4}$  and  $k \geq 11$ .

492 time  $t$ .

493 Random function  $\text{mark}(c, p)$  is defined recursively based on the level of  $c$ : If  $\text{level}(c) = 0$ , then  
494  $\text{mark}(c, p) = \{c\}$ . If  $\text{level}(c) \geq 1$ , then choose a subset  $C'$  consisting of exactly  $\lceil pk \rceil$  children of  $c$ ,  
495 uniformly at random, and let  $\text{mark}(c, p) = \bigcup_{c' \in C'} \text{mark}(c', p)$ .

496 In the noisy case, we need an upper bound ( $\sigma_2$  in the following definition) on the number of times a  
497 concept is noisy-shown. See the discussion in the footnote before [Theorem 6.4](#) for more details.

498 **Definition 6.2** ( $(\sigma_1, \sigma_2)$ -**bottom-up training schedule**). A training schedule is  $(\sigma_1, \sigma_2)$ -bottom-up,  
499 where  $\sigma_1$  and  $\sigma_2$  are positive integers,  $\sigma_1 \leq \sigma_2$ , provided that each concept in  $C$  appears in the list  
500 at least  $\sigma_1$  times and no more than  $\sigma_2$  times, and no concept in  $C$  appears before each of its children  
501 has appeared at least  $\sigma_1$  times.

502 **Definition 6.3** ( $(r_1, r_2, \sigma_1, \sigma_2, p)$ -**noisy learning**). Network  $\mathcal{N}(r_1, r_2, \sigma_1, \sigma_2, p)$ -noisy-learns con-  
503 cept hierarchy  $\mathcal{C}$  provided that the following holds. At any time after a training phase in which all the  
504 concepts of  $\mathcal{C}$  are  $p$ -noisy-shown according to a  $(\sigma_1, \sigma_2)$ -bottom-up training schedule, network  $\mathcal{N}$   
505  $(r_1, r_2)$ -recognizes  $\mathcal{C}$ .

## 506 6.2 Noisy Learning Algorithm

507 The algorithm is exactly the same as in [Section 5.2](#), except that here we use  $p$ -noisy showing  
508 ([Definition 6.1](#)) instead of ordinary showing ([Definition 4.4](#)). We prove that our modified algorithm  
509 is robust in that it works even for our notions of noisy showing and noisy learning.

510 Our theorem for noisy learning, [Theorem 6.4](#), differs from [Theorem 5.3](#) in that we guarantee  
511 "correctness" only in cases where each concept is noisy-shown at most  $n^6$  times, that is, in cases  
512 where the network  $(r_1, r_2, \sigma, n^6, p)$ -noisy learns the concept hierarchy.<sup>7</sup> Let  $\bar{w} = 1/\sqrt{pk + 1 - p}$ .

513 Our algorithm uses the learning rate  $\eta = \frac{(\frac{\delta p \bar{w}}{20})^3}{64Tk^2p^3}$  and the firing threshold  $\tau = r_2k(\bar{w} - 2\delta)$ , where  
514  $\delta = \bar{w}(r_2 - r_1)/50$ .

515 We now state our main theorem in the noisy-learning setting.

516 **Theorem 6.4 (Noisy-Learning Theorem)**. Let  $\mathcal{N}$  be the network described in [Section 3](#), with  
517 maximum layer  $\ell'_{\max}$ . Let  $r_1, r_2$  be reals with  $0 < r_1 < r_2 \leq 1$ ; assume that  $r_2 - r_1 \geq 1/k$  and  
518  $k \geq 2$ . Let  $\mathcal{C}$  be any concept hierarchy, with maximum level  $\ell_{\max} \leq \ell'_{\max}$  and a total of  $|C|$  concepts.  
519 Let  $\sigma = c' \frac{k^6}{p^6 \delta^3} (\ell_{\max} \log(k) + \log(|C|n/\delta))$ , for some large enough constant  $c'$ .

520 Then, w.h.p.,  $\mathcal{N}(r_1, r_2, \sigma, n^6, p)$ -noisy-learns concept hierarchy  $\mathcal{C}$ .<sup>8</sup>

## 521 6.3 Proof idea

522 In the presence of noise, many of the properties of the noise-free case no longer hold, rendering  
523 the proof significantly more involved. Here we give a rough outline of our proof; details appear in  
524 [Appendix B](#).

525 In the analysis we only consider the learning of one concept, as the interleaving of different concepts  
526 is no different than in the noise-free case and hence we do not repeat that analysis. Therefore, in the  
527 reminder we fix one concept.

528 First, we bound the worst-case change of potential during a period of  $T$  rounds (where the concept is  
529 shown), provided it is initially within certain bounds. We later show that it will stay throughout the  
530 first  $n^6$  rounds where the concept is shown.

531 We aim to derive bounds on the change of the weight of a single edge during such a period. It  
532 turns out that the way the weights change depends highly on the other weights, which makes

---

<sup>7</sup>Note that we assume that every concept is shown at most  $n^6$  times. This is natural since if we consider a number  $T$  of rounds 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. 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  $n^6$  rounds no unlikely sequence occurs that is very 'bad'.

<sup>8</sup>We define w.h.p in this paper to be  $1 - \frac{1}{n}$ .

533 the analysis non-trivial. For this reason, we refrain from showing convergence of each weight  
534 separately. Instead we use the following potential function  $\psi$ . to show that the max and min weight  
535 convergence towards  $\bar{w} = \frac{1}{\sqrt{pk+1-p}}$  and 0 respectively. Fix an arbitrary time  $t$  and let  $w_{min}(t)$  and  
536  $w_{max}(t)$  be the minimum and maximum weights among  $w_1(t), w_k(t), \dots, w_k(t)$ , respectively. Let  
537  $\psi(t) = \max \left\{ \frac{w_{max}(t)}{\bar{w}}, \frac{\bar{w}}{w_{min}(t)} \right\}$ .

538 Note that, in contrast to the noise-free case, weights belonging to representatives of sub-concepts  
539 converge to  $\bar{w}$  instead to  $1/\sqrt{k}$ .

540 Our goal is to show that the above potential decreases quickly until it is very close to 1. Showing  
541 that the potential decreases is involved, since one cannot simply use a worst-case approach, due to  
542 the terms in Oja’s rule being non-linear and potentially having a high variance, depending on the  
543 distribution of weights. Instead, the key to showing that  $\psi$  decreases is to carefully use the randomness  
544 over the input vector and to carefully bound the non-linear terms. Bounding these non-linear terms  
545 tightly presents a major challenge. To overcome it, we show that the changes of the weights form a  
546 Doob martingale allowing us to use Azuma-Hoeffding inequality to get asymptotically almost tight  
547 bounds on the change of the weights during the  $T$  rounds. The proof can be found in [Appendix B](#).

## 548 7 A Lower Bound

549 Our results so far demonstrate how concept hierarchies with  $\ell_{max}$  levels can be represented robustly  
550 by networks with the same number of layers, and how such representations can be learned, even in  
551 the presence of noise. We would also like lower bound theorems saying that  $\ell_{max}$  layers are necessary  
552 for robust representation, under suitable restrictions.

553 In this section, we give a first step toward such a result, [Theorem 7.1](#). It says that a network  $\mathcal{N}$   
554 with maximum layer 1 cannot recognize a concept hierarchy  $\mathcal{C}$  with maximum level 2. This bound  
555 depends only on the requirement that  $\mathcal{N}$  should recognize  $\mathcal{C}$  according to our definition for noisy  
556 recognition in [Definition 4.2](#). That definition says that the network must tolerate bounded noise, as  
557 expressed by the ratio parameters  $r_1$  and  $r_2$ . Our result assumes reasonable constraints on the values  
558 of  $r_1$  and  $r_2$ . Note that the bound does not involve learning, only recognition.

559 A preliminary generalization of this result to more levels and layers appears in [\[26\]](#). However,  
560 in addition to the basic definition of noisy recognition, this generalization uses a strong technical  
561 assumption about disjointness of certain sets of triggered neurons. This assumption might be  
562 reasonable, in that it is guaranteed by our learning algorithms in [Section 5.2](#); however, we think it is  
563 too strong and would prefer to weaken it to, say, a simple limitation on the number of neurons at each  
564 layer in the network. We leave this task for future work.

### 565 7.1 Assumptions for the lower bound

566 Here we list explicitly the assumptions that we use for our lower bound result, [Theorem 7.1](#). We  
567 state these assumptions in a general way, in terms of a particular concept hierarchy  $\mathcal{C}$  with concept  
568 set  $C$  and any number  $\ell_{max}$  of levels, and an arbitrary network  $\mathcal{N}$  with any number  $\ell'_{max}$  of layers.  
569 However, our lower bound result, [Theorem 7.1](#), refers to just the special case of two levels and one  
570 layer. These assumptions capture the idea that concept hierarchy  $\mathcal{C}$  is  $(r_1, r_2)$ -recognized by network  
571  $\mathcal{N}$ .

- 572 1. Every concept  $c \in C$  has a unique designated neuron  $rep(c)$  in the network. (In general, it  
573 might be in any layer, regardless of the level of  $c$ .)
- 574 2. Let  $B$  be any subset of  $C_0$ . If  $c \in supported_{r_2}(B)$ , then presentation of  $B$  at time  $t$  results  
575 in firing of  $rep(c)$  at time  $t + layer(rep(c))$ .
- 576 3. Let  $B$  be any subset of  $C_0$ . If  $c \notin supported_{r_1}(B)$ , then presentation of  $B$  at time  $t$  does  
577 not result in firing of  $rep(c)$  at time  $t + layer(rep(c))$ .

578 Throughout this section, we assume the model presented in [Section 2](#) and [Section 3](#). Furthermore,  
579 since we are considering recognition only, and not learning, we assume that the *engaged* state  
580 components are always equal to 0. Also throughout this section, we assume that  $r_1$  and  $r_2$  satisfy  
581 the following constraints:

- 582 1.  $0 \leq r_1 \leq r_2 \leq 1$ .  
583 2.  $r_1 k$  is not an integer; define  $r'_1$  so that  $r'_1 k = \lfloor r_1 k \rfloor$ .  
584 3. Define  $r'_2$  so that  $r'_2 k = \lceil r_2 k \rceil$ .  
585 4.  $(r'_2)^2 \leq 2r'_1 - (r'_1)^2$ .

586 we think these constraints are reasonable. For example, for  $k = 10$ ,  $r_1 = .51$  and  $r_2 = .8$  satisfy  
587 these conditions. Or  $r_1 = \frac{1}{3}$  and  $r_2 = \frac{2}{3}$ .

## 588 7.2 Impossibility for recognition for two levels and one layer

589 We consider an arbitrary concept hierarchy  $\mathcal{C}$  with maximum level 2 and concept set  $C$ . We assume  
590 a (static) network  $\mathcal{N}$  with maximum layer 1, and total connectivity from layer 0 neurons to layer 1  
591 neurons. For such a network and concept hierarchy, we get a contradiction to the noisy recognition  
592 problem in Section 4.2, for any values of  $r_1$  and  $r_2$  that satisfy the constraints given in Section 7.1.  
593 For the problem requirements, we use only Assumptions 1-3 from Section 7.1.

594 **Theorem 7.1.** *Assume that  $\mathcal{C}$  has maximum level 2 and  $\mathcal{N}$  has maximum layer 1. Assume that*  
595  *$r_1, r_2, r'_1, r'_2$  satisfy the constraints in Section 7.1. Then  $\mathcal{N}$  does not recognize  $\mathcal{C}$ , according to*  
596 *Assumptions 1-3.*

597 *Proof.* Assume for contradiction that  $\mathcal{N}$  recognizes  $\mathcal{C}$ . Let  $c$  denote any one of the concepts in  $C_2$ ,  
598 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  
599 of  $k^2$  grandchildren.

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

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

608 We consider two scenarios. In Scenario A (the "must-fire scenario"), we choose input set  $B$  to consist  
609 of enough leaves of  $c$  to force  $rep(c)$  to fire, that is, we ensure that  $c \in supported_{r_2}(B)$ , while trying  
610 to minimize the total weight incoming to  $rep(c)$ . Specifically, we choose the  $r'_2 k \geq r_2 k$  children  $b$  of  
611  $c$  with the smallest values of  $W(b)$ . And for each such  $b$ , we choose its  $r'_2 k$  children with the smallest  
612 weights. Let  $B$  be the union of all of these  $r'_2 k$  sets of  $r'_2 k$  grandchildren of  $c$ . Since  $r'_2 k \geq r_2 k$ , it  
613 follows that  $c \in supported_{r_2}(B)$ .

614 *Claim 1:* In Scenario A, the total incoming potential to  $rep(c)$  is at most  $(r'_2)^2 W$ .

615 In Scenario B (the "can't-fire scenario"), we choose input set  $B$  to consist of leaves of  $c$  that force  
616  $rep(c)$  not to fire, that is, we ensure that  $c \notin supported_{r_1}(B)$ , while trying to maximize the total  
617 weight incoming to  $rep(c)$ . Specifically, we choose the  $r'_1 k < r_1 k$  children  $b$  of  $c$  with the largest  
618 values of  $W(b)$ , and we include all of their children in  $B$ . For each of the remaining  $(1 - r'_1)k$   
619 children of  $c$ , we choose its  $r'_1 k < r_1 k$  children with the largest weights and include them all in  $B$ .  
620 Since  $r'_1 k$  is strictly less than  $r_1 k$ , it follows that  $c \notin supported_{r_1}(B)$ .

621 *Claim 2:* In Scenario B, the total incoming potential to  $rep(c)$  is at least  $(r'_1)W + (1 - r'_1)r'_1 W =$   
622  $(2r'_1 - (r'_1)^2)W$ .

623 *Proof of Claim 2:* We define:

- 624 •  $W_1$ : The total of the weights  $W(b)$  for the  $r'_1 k$  children  $b$  of  $c$  with the largest values of  
625  $W(b)$ .
- 626 •  $W_2 = W - W_1$ : The total of the weights  $W(b)$  for the remaining  $(1 - r'_1)k$  children of  $c$ .



- $W_3$ : We know that  $W_1 \geq r'_1 W$ , since  $W_1$  gives the total weight for the  $r'_1 k$  children of  $c$  with the largest weights, out of  $k$  children. Define  $W_3 = W_1 - r'_1 W$ ; then  $W_3$  must be nonnegative.

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

$$\begin{aligned}
&\geq W_1 + r'_1 W_2, \\
&= r'_1 W + W_3 + r'_1 (W - W_1), \\
&= r'_1 W + W_3 + r'_1 (W - W_3 - r'_1 W), \\
&= 2r'_1 W - (r'_1)^2 W + (1 - r'_1) W_3, \\
&\geq 2r'_1 W - (r'_1)^2 W, \\
&= (2r'_1 - (r'_1)^2) W,
\end{aligned}$$

as needed.

*End of proof of Claim 2*

Now, Claim 1 implies that the threshold  $\tau$  of neuron  $rep(c)$  must be at most  $(r'_2)^2 W$ , 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  $(2r'_1 - (r'_1)^2) W$ , since it must be large enough to prevent the given  $B$  from triggering firing of  $rep(c)$ . So we must have

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

which implies that

$$2r'_1 - (r'_1)^2 < (r'_2)^2.$$

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

## 8 Conclusions and Future Work

In this paper, we have proposed a theoretical model for recognizing and learning hierarchically-structured concepts in synchronous, feed-forward layered Spiking Neural Networks. Our networks use Oja's learning rule for adjusting synapse weights. Based on this model, we have presented two learning algorithms, one for noise-free learning and one that allows bounded noise. Both algorithms learn concepts in a bottom-up manner, but allow arbitrary interleaving in learning of incomparable concepts. We have analyzed both algorithms in detail.

The representations produced by these algorithms are certain types of embeddings of the hierarchical concept structure in the neural network. These representations support robust concept recognition, even when some of the inputs are missing. We have also provided a preliminary lower bound on the number of layers, saying that two-level concepts cannot be recognized robustly in one-level networks.

This paper represents a first step towards a theory of representation and learning for hierarchically-structured concepts in SNNs. In the longer term, we are interested in theoretical models that capture key features of real computer vision algorithms and brain networks. Our current model is highly abstract and makes many simplifying assumptions: for instance, we assume that concepts are strictly tree-structured, that every concept has the same number of children, that the number of network layers is at least as large as the number of concept levels, that the networks are feed-forward, and that the learning rule is applied without error. To make the results more realistic, one should loosen all of these these assumptions, systematically.

The results in this paper suggest numerous directions for future research:

**Extensions to our results:** One can consider more flexible orders in which concepts in a hierarchy can be learned, based on a larger class of training schedules. 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 networks, for example, to presentation of a few "extraneous" inputs that are not part of the concept being shown, to noise in calculating potentials, or to failures of neurons or synapses.

664 Also, our algorithms use some auxiliary capabilities, such as Winner-Take-All, in order to select  
665 neurons for learning; it would be interesting to combine our algorithms with network implementations  
666 of these auxiliary capabilities in order to obtain complete, self-contained networks that solve the  
667 learning problem "from scratch". Finally, we would like to strengthen the lower bound results to  
668 apply to many levels and layers.

669 **Variations in the network model:** Our networks have a simple layered structure; it would be  
670 interesting to consider some natural variations. For example, instead of all-to-all connections between  
671 consecutive layers, what happens to the results if one assumes a smaller number of randomly-  
672 determined connections between layers? Also, in our networks, all edges go from one layer  $\ell$  to the  
673 next higher layer  $\ell + 1$ . How do the results change if one allows edges to go from layer  $\ell$  to any  
674 higher layer?

675 What would be the impact on the results of allowing feedback edges from each layer  $\ell$  to the next-  
676 lower layer  $\ell - 1$ ? How would the costs of recognizing and learning concepts change based on  
677 feedback from representations of higher-level concepts?

678 What would be the effect on the results of using other incremental learning rules besides Oja's  
679 rule? In an extreme case, what happens to the results if learning occurs all at once, rather than  
680 incrementally? In general, how can we compare the computational power of incremental learning  
681 models vs. one-shot learning models such as the one in [52, 20]?

682 **Variations in the data model:** Another interesting research direction is to consider variations on  
683 the structure of concept hierarchies. How do the results change if we allow different numbers of  
684 children for different concepts? It is not clear how one can set the firing thresholds in this case.  
685 Perhaps these thresholds could be 'learned'. Another interesting extension is to allow a level  $\ell$   
686 concept to have children at any level smaller than  $\ell$ , rather than just level  $\ell - 1$ ? What happens if a  
687 concept hierarchy need not be a tree, but may include a bounded amount of overlap between the sets  
688 of children of different concepts?

689 It would be interesting to understand more generally what kinds of logical structures can be learned  
690 by synchronous SNNs. In our concept hierarchies, each level  $\ell + 1$  concept corresponds to the "and"  
691 of several level  $\ell$  concepts. What if we allow concepts that correspond to "ors", or "nors", of other  
692 concepts? Similar questions were suggested by Valiant [47], in terms of a different model. Also, in  
693 addition to learning individual concepts, it would be interesting to consider learning relationships  
694 between concepts, such as association, causality, or sequential order.

695 **Different forms of representation:** In this paper, each concept  $c$  is represented by just one neuron  
696  $rep(c)$ . An interesting extension, which may be more biologically plausible, would be to allow  
697 the representation of each concept  $c$  to be a more elaborate "code" consisting of a particular set of  
698 neurons that fire. Important examples here are representations based on "cell assemblies" [38, 39].  
699 What are the theoretical advantages and costs of such codes, compared to simpler single-neuron  
700 representations? Another type of extension would be to "time-share" the network, allowing the same  
701 layer of the network to represent different levels of the concept hierarchy at different times. Ideas  
702 from [6, 19] on state machine simulations in neural networks may be useful here.

703 **Experimental work:** All the ideas we have presented in this paper are purely theoretical. It would  
704 be valuable to complement this work with experiments to evaluate the performance and robustness of  
705 the algorithms presented here, as well as future algorithms.

## 706 References

- 707 [1] Alain Artola, S Bröcher, and Wolf Singer. Different voltage-dependent thresholds for induc-  
708 ing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature*,  
709 347(6288):69, 1990.
- 710 [2] Alain Artola and Wolf Singer. Long-term depression of excitatory synaptic transmission and its  
711 relationship to long-term potentiation. *Trends in neurosciences*, 16(11):480–487, 1993.
- 712 [3] Elie L Bienenstock, Leon N Cooper, and Paul W Munro. Theory for the development of  
713 neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of*  
714 *Neuroscience*, 2(1):32–48, 1982.

- 715 [4] Robert Coultrip, Richard Granger, and Gary Lynch. A cortical model of winner-take-all  
716 competition via lateral inhibition. *Neural Networks*, 5(1):47–54, 1992.
- 717 [5] Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of*  
718 *randomized algorithms*. Cambridge University Press, 2009.
- 719 [6] Rebecca Fay, Ulrich Kaufmann, Andreas Knoblauch, Heiner Markert, and Günther Palm.  
720 Combining visual attention, object recognition and associative information processing in a  
721 neurobotic system. In *Biomimetic neural learning for intelligent robots*, pages 118–143.  
722 Springer, 2005.
- 723 [7] Daniel J Felleman and DC Essen Van. Distributed hierarchical processing in the primate cerebral  
724 cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.
- 725 [8] Peter Földiák and Peter Fdíl. Adaptive network for optimal linear feature extraction. 1989.
- 726 [9] Wulfram Gerstner and Werner M. Kistler. *Spiking neuron models: Single neurons, populations,*  
727 *plasticity*. Cambridge University Press, 2002.
- 728 [10] Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eli-  
729 gibility traces and plasticity on behavioral time scales: experimental support of neohebbian  
730 three-factor learning rules. *Frontiers in neural circuits*, 12:53, 2018.
- 731 [11] Stefan Habenschuss, Zeno Jonke, and Wolfgang Maass. Stochastic computations in cortical  
732 microcircuit models. *PLoS Computational Biology*, 9(11):e1003311, 2013.
- 733 [12] D. O. Hebb. *The Organization of Behavior*. Wiley and Sons, New York, 1949.
- 734 [13] Yael Hitron, Nancy A. Lynch, Cameron Musco, and Merav Parter. Random sketching, clustering,  
735 and short-term memory in spiking neural networks. In *11th Innovations in Theoretical Computer*  
736 *Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, pages 23:1–  
737 23:31, 2020. URL: <https://doi.org/10.4230/LIPIcs.ITCS.2020.23>, doi:10.4230/  
738 [LIPIcs.ITCS.2020.23](https://doi.org/10.4230/LIPIcs.ITCS.2020.23).
- 739 [14] D. Hubel and T. Wiesel. Receptive fields, binocular interaction, and functional architecture in  
740 the cat’s visual cortex. *Journal of Physiology*, 160:106–154, 1962.
- 741 [15] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat’s  
742 striate cortex. *The Journal of Physiology*, 148(3):574–591, 1959. URL: <https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1959.sp006308>,  
743 [arXiv:https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.](https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1959.sp006308)  
744 [1959.sp006308](https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1959.sp006308), doi:10.1113/jphysiol.1959.sp006308.
- 745 [16] JM Hupé, AC James, BR Payne, SG Lomber, P Girard, and J Bullier. Cortical feedback  
746 improves discrimination between figure and background by v1, v2 and v3 neurons. *Nature*,  
747 394(6695):784, 1998.
- 748 [17] Eugene M. Izhikevich. Which model to use for cortical spiking neurons? *IEEE Transactions on*  
749 *Neural Networks*, 15(5):1063–1070, 2004.
- 750 [18] Richard Kempter, Wulfram Gerstner, and J Leo Van Hemmen. Hebbian learning and spiking  
751 neurons. *Physical Review E*, 59(4):4498, 1999.
- 752 [19] Andreas Knoblauch, Heiner Markert, and Günther Palm. An associative cortical model of  
753 language understanding and action planning. In *International work-conference on the interplay*  
754 *between natural and artificial computation*, pages 405–414. Springer, 2005.
- 755 [20] Andreas Knoblauch, Günther Palm, and Friedrich T Sommer. Memory capacities for synaptic  
756 and structural plasticity. *Neural Computation*, 22(2):289–341, 2010.
- 757 [21] John Lazzaro, Sylvie Ryckebusch, Misha Anne Mahowald, and Carver A. Mead. Winner-take-  
758 all networks of  $o(n)$  complexity. Technical report, DTIC Document, 1988.
- 759 [22] Robert A. Legenstein, Wolfgang Maass, Christos H. Papadimitriou, and Santosh S. Vempala.  
760 Long term memory and the densest  $k$ -subgraph problem. In *9th Innovations in Theoretical*  
761 *Computer Science (ITCS 2018)*, pages 57:1–57:15, Cambridge, MA, January 2018.
- 762 [23] S. Lowel and W. Singer. Selection of intrinsic horizontal connections in the visual cortex by  
763 correlated neuronal activity. *Science Magazine*, 255(5041):209–212, January 1992.
- 764 [24] Nancy Lynch, Cameron Musco, and Merav Parter. Computational tradeoffs in biological neural  
765 networks: Self-stabilizing winner-take-all networks. In *ITCS 2017*, 2017. Full version available  
766 at <https://arxiv.org/abs/1610.02084>.  
767

- 768 [25] Nancy Lynch, Cameron Musco, and Merav Parter. Winner-take-all computation in spiking  
769 neural networks, April 2019. arXiv:1904.12591.
- 770 [26] Nancy A. Lynch and Frederik Mallmann-Trenn. Learning hierarchically structured concepts.  
771 *CoRR*, abs/1909.04559v3, 2019. URL: <http://arxiv.org/abs/1909.04559v3>, arXiv:  
772 [1909.04559v3](http://arxiv.org/abs/1909.04559v3).
- 773 [27] Nancy A. Lynch and Cameron Musco. A basic compositional model for spiking neural networks.  
774 *CoRR*, abs/1808.03884, 2018. URL: <http://arxiv.org/abs/1808.03884>, arXiv:1808.  
775 [03884](http://arxiv.org/abs/1808.03884).
- 776 [28] Nancy A. Lynch, Cameron Musco, and Merav Parter. Computational tradeoffs in biological  
777 neural networks: Self-stabilizing winner-take-all networks. In *8th Innovations in Theoretical  
778 Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 15:1–  
779 15:44, 2017. URL: <https://doi.org/10.4230/LIPIcs.ITCS.2017.15>, doi:10.4230/  
780 [LIPIcs.ITCS.2017.15](https://doi.org/10.4230/LIPIcs.ITCS.2017.15).
- 781 [29] Wolfgang Maass. On the computational power of noisy spiking neurons. In *NIPS1996*, 1996.
- 782 [30] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models.  
783 *Neural Networks*, 10(9):1659–1671, 1997.
- 784 [31] Wolfgang Maass. Neural computation with winner-take-all as the only nonlinear operation. In  
785 *NIPS 1999*, pages 293–299, 1999.
- 786 [32] Wolfgang Maass. On the computational power of winner-take-all. *Neural Computation*, 2000.
- 787 [33] Nikola T Markov, Julien Vezoli, Pascal Chameau, Arnaud Falchier, René Quilodran, Cyril  
788 Huissoud, Camille Lamy, Pierre Misery, Pascale Giroud, Shimon Ullman, et al. Anatomy  
789 of hierarchy: feedforward and feedback pathways in macaque visual cortex. *Journal of  
790 Comparative Neurology*, 522(1):225–259, 2014.
- 791 [34] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. Learning functions: when is deep better  
792 than shallow. *arXiv preprint arXiv:1603.00988*, 2016.
- 793 [35] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical  
794 biology*, 15(3):267–273, 1982.
- 795 [36] Erkki Oja. Principal components, minor components, and linear neural networks. *Neural  
796 networks*, 5(6):927–935, 1992.
- 797 [37] Matthias Oster and Shih-Chii Liu. Spiking inputs to a winner-take-all network. In *NIPS 2006*,  
798 page 1051, 2006.
- 799 [38] Günther Palm. *Neural assemblies: An alternative approach to artificial intelligence*, volume 7.  
800 Springer Science & Business Media, 2012.
- 801 [39] Günther Palm, Andreas Knoblauch, Florian Hauser, and Almut Schüz. Cell assemblies in the  
802 cerebral cortex. *Biological cybernetics*, 108(5):559–572, 2014.
- 803 [40] Christos H. Papadimitriou and Santosh S. Vempala. Cortical learning via prediction. *Proceedings  
804 of Machine Learning Research (PMLR)*, 40:1402–1422, 2015.
- 805 [41] Christos H. Papadimitriou and Santosh S. Vempala. Random projection in the brain and  
806 computation with assemblies of neurons. In *10th Innovation in Theoretical Computer Science  
807 (ITCS 2019)*, pages 57:1–57:19, San Diego, CA, January 2019.
- 808 [42] Christos H. Papadimitriou, Santosh S. Vempala, Daniel Mitropolsky, Michael Collins,  
809 and Wolfgang Maass. Brain computation by assemblies of neurons, December 2019.  
810 bioRxiv:10.1101/869156v1.
- 811 [43] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in  
812 cortex. *Nature neuroscience*, 2(11):1019–1025, 1999.
- 813 [44] Sven Schrader, Markus Diesmann, and Abigail Morrison. A compositionality machine realized  
814 by a hierarchic architecture of synfire chains. *Frontiers in Computational Neuroscience*, 4:154,  
815 2011. URL: <https://www.frontiersin.org/article/10.3389/fncom.2010.00154>,  
816 doi:10.3389/fncom.2010.00154.
- 817 [45] Lili Su and Chia-Jung Chang and Nancy Lynch. Spike-based winner-take-all computation:  
818 Fundamental limits and order-optimal circuits. *Neural Computation*, 31(12), December 2019.  
819 Published online. Also, arXiv:1904.10399.

- 820 [46] Simon J. Thorpe. Spike arrival times: A highly efficient coding scheme for neural networks.  
821 *Parallel Processing in Neural Systems*, pages 91–94, 1990.
- 822 [47] Leslie G. Valiant. *Circuits of the Mind*. Oxford University Press on Demand, 2000.
- 823 [48] Leslie G. Valiant. A neuroidal architecture for cognitive computation. *Journal of the ACM*  
824 (*JACM*), 47(5):854–882, 2000.
- 825 [49] Leslie G. Valiant. Memorization and association on a realistic neural model. *Neural Computa-*  
826 *tion*, 17(3):527–555, 2005.
- 827 [50] Leslie G. Valiant. The hippocampus as a stable memory allocator for cortex. *Neural Computa-*  
828 *tion*, 24(11):2873–2899, 2012.
- 829 [51] Wei Wang and Jean-Jacques E. Slotine.  $k$ -winners-take-all computation with neural oscillators.  
830 *arXiv preprint q-bio/0401001*, 2003.
- 831 [52] David J Willshaw, O Peter Buneman, and Hugh Christopher Longuet-Higgins. Non-holographic  
832 associative memory. *Nature*, 222(5197):960–962, 1969.
- 833 [53] Alan L. Yuille and Norberto M. Grzywacz. A winner-take-all mechanism based on presynaptic  
834 inhibition feedback. *Neural Computation*, 1(3):334–347, 1989.
- 835 [54] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks.  
836 In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision –*  
837 *ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.
- 838 [55] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representa-  
839 tions via network dissection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,  
840 41(9):2131–2145, September 2019.

## 841 A Analysis of Noise-free Learning

842 Here we present our analysis for the noise-free learning algorithm in [Section 5](#). In [Section A.1](#), we  
843 describe how incoming weights change for a particular neuron when it is presented with a consistent  
844 input vector. In [Section A.2](#), we prove our main invariant, saying how neurons get bound to concepts,  
845 when neuron firing occurs, and how weights change, during the time when the network is learning. In  
846 [Section A.3](#), we use that invariant to prove [Theorem 5.3](#).

### 847 A.1 Weight Change for Individual Neurons

848 In this subsection we give a series of three lemmas that describe how incoming weights change for a  
849 particular neuron when it is presented with a consistent input vector during execution of our noise-free  
850 learning network. Throughout this subsection, we consider a single neuron  $u$  with  $\text{layer}(u) \geq 1$ .

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

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

- 858 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  
859 incoming neighbors in  $F$  fire at time  $t-1$ .
- 860 2. All weights  $w_i(t-1)$ ,  $i \in F$  are equal, and all weights  $w_i(t-1)$ ,  $i \notin F$  are equal.
- 861 3. For every  $i \in F$ ,  $0 < w_i(t-1) < \frac{1}{\sqrt{k}}$ .
- 862 4. For every  $i \notin F$ ,  $w_i(t-1) > 0$ .
- 863 5.  $0 < \eta \leq \frac{1}{4k}$ .

864 *Then:*

865 1. All weights  $w_i(t), i \in F$  are equal, and all weights  $w_i(t), i \notin F$  are equal.

866 2. For every  $i \in F, w_i(t) > w_i(t - 1)$ .

867 3. For every  $i \in F, w_i(t) < \frac{1}{\sqrt{k}}$ .

868 4. For every  $i \notin F, w_i(t) < w_i(t - 1)$ .

869 5. For every  $i \notin F, w_i(t) > 0$ .

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

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

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

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

876 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 have  $w_j(t - 1) = \frac{1}{\sqrt{k}} - \lambda$ . We thus have that

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

877 as needed.

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

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

878 as needed.

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

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

879 as needed. □

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

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

- 886 1. For every  $t \geq 0$ ,  $x_i(t) = 1$  for every  $i \in F$  and  $x_i(t) = 0$  for every  $i \notin F$ .
- 887 2. All weights  $w_i(0)$  are equal.
- 888 3.  $0 < w_i(0) < \frac{1}{\sqrt{k}}$  for every  $i$ .
- 889 4.  $0 < \eta \leq \frac{1}{4k}$ .

890 Then for any  $t \geq 1$ :

- 891 1. All weights  $w_i(t)$ ,  $i \in F$  are equal, and all weights  $w_i(t)$ ,  $i \notin F$  are equal.
- 892 2.  $0 < w_i(t) < \frac{1}{\sqrt{k}}$  for every  $i$ .
- 893 3. For every  $i \in F$ ,  $w_i(t) > w_i(0)$ .
- 894 4. For every  $i \notin F$ ,  $w_i(t) < w_i(0)$ .

895 **Lemma A.3** gives quantitative bounds on the amount of weight increase and weight decrease over  
896 many rounds, again for a single neuron  $u$  involved in learning a single concept. We use notation  
897  $w(t)$ ,  $x(t)$ ,  $z(t)$  as before. We assume that  $x(t)$  is the same at all times  $t = 0, 1, \dots$ , and assume that  
898 the engaged component  $e(t)$  is equal to 1 at all times  $t$ .

899 **Lemma A.3** (Learning Properties). Let  $F \subseteq \{1, \dots, n\}$  with  $|F| = k$ . Let  $\varepsilon \in (0, 1]$ .  
900 Let  $b$  be a positive integer. Let  $\sigma = \frac{4}{3\eta k}((\ell_{\max} + 1) \log(k)) + \frac{3}{\eta k \varepsilon} + \frac{b \log(k)}{\log(\frac{10}{15})}$ . Thus,  $\sigma$  is  
901  $O\left(\frac{1}{\eta k} (\ell_{\max} \log(k) + \frac{1}{\varepsilon}) + b \log(k)\right)$ . Assume that:

- 902 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  $e(t) = 1$ .
- 903 2. All weights  $w_i(0)$  are equal to  $\frac{1}{k^{\ell_{\max}}}$ .
- 904 3.  $\eta = \frac{1}{4k}$ .<sup>9</sup>

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

- 906 1. For any  $i \in F$ , we have  $w_i(t) \in [\frac{1}{(1+\varepsilon)\sqrt{k}}, \frac{1}{\sqrt{k}}]$ .
- 907 2. For any  $i \notin F$ , we have  $w_i(t) \in [0, \frac{1}{k^{\ell_{\max} + b}}]$ .

908 *Proof.* We first show Part 1. **Lemma A.2** implies the upper bound of  $\frac{1}{\sqrt{k}}$ , so it remains to show the  
909 lower bound. We do this in two steps, first increasing the weight to an intermediate target value  $\frac{1}{2\sqrt{k}}$   
910 and then to the real target value  $\frac{1}{(1+\varepsilon)\sqrt{k}}$ . These two steps use different arguments.

911 For the first step, we begin with Claim 1, which bounds the number of rounds required to double the  
912 weight  $w_i$ , for  $i \in F$ , when  $w_i$  is not "too close" to the target weight  $\frac{1}{\sqrt{k}}$ .

913 *Claim 1:* Assume that  $i \in F$ . For any positive integer  $j$ , the number of rounds needed to increase  $w_i$   
914 from  $\frac{1}{2^{j+1}\sqrt{k}}$  to  $\frac{1}{2^j\sqrt{k}}$  is at most  $\frac{4}{3\eta k}$ .

<sup>9</sup>This is a very precise assumption but it could be weakened, at a corresponding cost in running time.

*Proof of Claim 1:* Since all the weights are the same and  $\frac{1}{2^{j+1}\sqrt{k}} \leq w_i(t-1) \leq \frac{1}{2\sqrt{k}}$ , we get:

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

915 Increasing  $w_i$  from  $\frac{1}{2^{j+1}\sqrt{k}}$  to  $\frac{1}{2^j\sqrt{k}}$  means we must increase it by an additive amount of  $\frac{1}{2^{j+1}\sqrt{k}}$ . We  
916 have just shown that each round increases  $w_i$  by at least  $\eta k \frac{1}{2^{j+1}\sqrt{k}} (3/4)$ . Thus, the number of rounds  
917 required 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 by  $\eta k \frac{1}{2^{j+1}\sqrt{k}} (3/4)$ , which is  
918  $\frac{4}{3\eta k}$ .

919 *End of proof of Claim 1.*

920 Now we can prove the first step, bounding the number of rounds required for the weight to reach at  
921 least  $\frac{1}{2\sqrt{k}}$ :

922 *Claim 2:* For  $i \in F$ , the number of rounds required to increase  $w_i$  from the starting value  $\frac{1}{k^{\ell_{\max}}}$  to  
923 the intermediate target value  $\frac{1}{2\sqrt{k}}$  is at most  $\frac{4}{3\eta k} ((\ell_{\max} + 1) \log(k))$ .

924 *Proof of Claim 2:* By applying Claim 1  $(\ell_{\max} + 1) \log(k)$  times.

925 *End of Proof of Claim 2.*

926 Next, for the second step, we bound the number of rounds required to increase  $w_i, i \in F$ , from  $\frac{1}{2\sqrt{k}}$   
927 to  $\frac{1}{(1+\varepsilon)\sqrt{k}}$ . This time, of course, depends on  $\varepsilon$ .

928 *Claim 3:* For  $i \in F$ , the number of rounds required to increase  $w_i$  from the intermediate target value  
929  $\frac{1}{2\sqrt{k}}$  to the final target value  $\frac{1}{(1+\varepsilon)\sqrt{k}}$  is at most  $\frac{3}{\eta k \varepsilon}$ .

930 *Proof of Claim 3:* The argument is generally similar to that for Claim 1, but now using the fact that  
931  $\frac{1}{2\sqrt{k}} \leq w_i(t-1) \leq \frac{1}{(1+\varepsilon)\sqrt{k}}$ :

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

932 where we used the fact that  $(1 - 1/(1+x)^2) \geq x/3$  for  $0 \leq x \leq 1$ . It follows that the total time to  
933 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

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

934 *End of Proof of Claim 3.*



935 It follows that the total number of rounds for Part 1 is at most the sum of the bounds from Claims 2  
 936 and 3, or

$$\frac{4}{3\eta k} ((\ell_{\max} + 1) \log(k)) + \frac{3}{\eta k \varepsilon},$$

937 which is  $O\left(\frac{1}{\eta k} (\ell_{\max} \log(k) + \frac{1}{\varepsilon})\right)$ .

938 Note that once the weights for indices in  $F$  reach their target values, they never decrease below those  
 939 values. This follows from strict monotonicity shown in [Lemma A.2](#).

940 We now turn to proving Part 2. [Lemma A.2](#) implies the lower bound, so it remains to show the upper  
 941 bound.

942 We consider what happens after the increasing weights (for indices in  $F$ ) have already reached the  
 943 level  $\frac{1}{2\sqrt{k}}$ , and then bound the number of rounds for the decreasing weights to decrease to the desired  
 944 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  
 945 to guarantee that  $z$  is "large enough" to produce a sufficient amount of decrease. For this part, we use  
 946 our assumed lower bound on  $\eta$ .

947 *Claim 4:* For  $i \notin F$ , the number of rounds required to decrease  $w_i$  from the starting weight  $\frac{1}{k^{\ell_{\max} + 1}}$   
 948 to  $\frac{1}{k^{\ell_{\max} + b}}$  is at most  $\frac{b \log_2 k}{\log_2 \frac{15}{16}}$ , which is  $O(b \log(k))$ .

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

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

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

950 Thus, the weight decreases by a factor of 15/16 at each round. Now consider the number of rounds  
 951 needed to reduce from  $\frac{1}{k^{\ell_{\max} + 1}}$  to the target weight  $\frac{1}{k^{\ell_{\max} + b}}$ . This number is bounded by  $\frac{b \log_2 k}{\log_2 \frac{15}{16}}$ ,  
 952 which is  $O(b \log(k))$ , as claimed.

953 *End of Proof of Claim 4.*

954 Summing the bounds for Part 1 (increasing) and Part 2 (decreasing), we see that the total number of  
 955 rounds to complete all the needed increases and decreases is at most

$$\frac{4}{3\eta k} ((\ell_{\max} + 1) \log(k)) + \frac{3}{\eta k \varepsilon} + \frac{b \log_2 k}{\log_2 \frac{15}{16}},$$

956 which is  $O\left(\frac{1}{\eta k} (\ell_{\max} \log(k) + \frac{1}{\varepsilon}) + b \log(k)\right)$ , as needed.

957 □

## 958 A.2 Main Invariants

959 In this section, we give a key lemma, [Lemma A.6](#), which describes key properties of the algorithm  
 960 with respect to engagement, weight settings, and firing. This lemma deals with the network as a  
 961 whole, and draws upon the lemmas in [Section A.1](#) for properties involving learning by individual  
 962 neurons. [Lemma A.6](#) relies on assumptions about the input, captured by our  $\sigma$ -bottom-up training  
 963 definition, and also about the settings of *engagement* flags.

964 For the rest of [Appendix A](#), we use the following assumptions about the various parameter settings:

- 965 1. The concept hierarchy consists of  $\ell_{\max}$  levels.
- 966 2. The network consists of  $\ell'_{\max}$  levels, with  $\ell_{\max} \leq \ell'_{\max}$ .

- 967 3.  $b$  is a positive real  $\geq 2$ .
- 968 4.  $r_1, r_2$  satisfy  $0 < r_1 < r_2 \leq 1$ , and  $r_1 k$  is not an integer; more strongly, we assume the  
969 technical condition that  $r_1 k - \lfloor r_1 k \rfloor \geq \frac{\sqrt{k}}{k^{b-1}}$ . Furthermore, we assume that  $\frac{1}{\sqrt{k}} + \frac{1}{k} \leq \frac{r_2 \sqrt{k}}{2}$ .
- 970 5.  $\varepsilon = \frac{r_2 - r_1}{r_1 + r_2}$ .
- 971 6.  $\tau = \frac{(r_1 + r_2)\sqrt{k}}{2}$ .
- 972 7.  $\eta = \frac{1}{4k}$ .
- 973 8.  $\sigma$ , for the  $\sigma$ -bottom-up training schedule definition, is equal to  $\frac{4}{3\eta k} ((\ell_{\max} + 1) \log(k)) +$   
974  $\frac{3}{\eta k \varepsilon} + \frac{b \log(k)}{\log(\frac{16}{15})}$ . Thus,  $\sigma$  is  $O\left(\frac{1}{\eta k} (\ell_{\max} \log(k) + \frac{1}{\varepsilon}) + b \log(k)\right)$ .

975 We use the following assumption about the settings of the engagement flags.

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

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

980 Recall that, by [Assumption 5.2](#), the WTA selects exactly one layer  $\ell$  neuron at time  $t$ . This, together  
981 with [Assumption 5.2](#), implies that exactly one layer  $\ell$  neuron will be engaged at time  $t$ .

982 We also define the point at which a particular layer  $\ell$  neuron  $u$  gets "bound" to a particular level  $\ell$   
983 concept  $c$ . Namely, we say that a layer  $\ell$  neuron  $u$ ,  $\ell \geq 1$ , "binds" to a level  $\ell$  concept  $c$  at time  $t$  if  $c$   
984 is presented for the first time at time  $t - \ell$ , and  $u$  is the neuron that is engaged at time  $t$ . At that point,  
985 we define  $\text{rep}(c) = u$ .

986 Here is a simple auxiliary lemma, about unbound neurons.

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

- 988 1. If  $u$  is unbound at time  $t$ , then all of  $u$ 's incoming weights at time  $t$  are the initial weight  
989  $\frac{1}{k^{\ell_{\max} + 1}}$ .
- 990 2. If  $u$  is unbound at time  $t$ , then  $u$  does not fire at time  $t$ .

991 We are now ready to prove our main lemma. It has five parts, whose proofs are intertwined.

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

- 994 1. The  $\text{rep}()$  mapping from the set  $C$  of concepts to the set  $N$  of neurons  $a$  is one-to-one  
995 mapping; that is, for any two distinct concepts  $c$  and  $c'$  for which  $\text{rep}(c)$  and  $\text{rep}(c')$  are  
996 both defined by time  $t$ , we have  $\text{rep}(c) \neq \text{rep}(c')$ .
- 997 2. For every concept  $c$  with  $\text{level}(c) \geq 1$ , every showing of  $c$  at a time  $\leq t - \text{level}(c)$ , leads to  
998 the same neuron  $u = \text{rep}(c)$  becoming engaged at time  $t$ .
- 999 3. For every concept  $c$  with  $\text{level}(c) \geq 1$ , and any  $t' \geq 1$ , if  $c$  is shown at time  $t - \text{level}(c)$  for  
1000 the  $t'$ -th time, then the following are true at time  $t$ :

- 1001 (a) Neuron  $u = \text{rep}(c)$  has weights in  $\left(\frac{1}{k^{\ell_{\max} + 1}}, \frac{1}{\sqrt{k}}\right)$  for all neurons in  $\text{rep}(\text{children}(c))$ ,  
1002 and weights in  $\left(0, \frac{1}{k^{\ell_{\max} + 1}}\right)$  for all other neurons.
- 1003 (b) If  $t' \geq \sigma$ , then  $u$  with  $u = \text{rep}(c)$  has weights in  $\left[\frac{1}{(1+\varepsilon)\sqrt{k}}, \frac{1}{\sqrt{k}}\right]$  for all neurons in  
1004  $\text{rep}(\text{children}(c))$ , and weights in  $\left[0, \frac{1}{k^{\ell_{\max} + b}}\right]$  for all other neurons.

1005 4. For every concept  $c$ , if a proper ancestor of  $c$  is shown at time  $t - \text{level}(c)$ , then  $\text{rep}(c)$  is  
1006 defined by time  $t$ , and fires at time  $t$ .

1007 5. For any neuron  $u$ , the following holds. If  $u$  fires at time  $t$ , then there exists  $c$  such that  
1008  $u = \text{rep}(c)$  at time  $t$ , and an ancestor of  $c$  is shown at time  $t - \text{layer}(u)$ . (This ancestor  
1009 could be  $c$  or a proper ancestor of  $c$ .)

1010 *Proof.* First observe that, by [Assumption A.4](#), every representative  $rep(c)$  is on the layer equal to  
1011  $level(c)$ . We prove the five-part statement of the lemma by induction on  $t$ .

1012 *Base:*  $t = 0$ .

1013 For Part 1, the only concepts for which *reps* are defined at time 0 are level 0 concepts, and these all  
1014 have distinct *reps* by assumption. For Parts 2 and 3, note that  $level(c) \geq 1$  implies that the times  
1015 in question are negative, which is impossible; so these are trivially true. For Part 4, it must be that  
1016  $level(c) = 0$  (to avoid negative times), and a proper ancestor of  $c$  is shown at time 0. Then the layer  
1017 0 neuron  $rep(c)$  fires at time 0, by the definition of "showing".

1018 For Part 5, first note that at time 0 no neurons at layers  $\geq 1$  are bound, so by [Lemma A.5](#), they cannot  
1019 fire at time 0. Since we assume that  $u$  fires at time 0, it must be that  $layer(u) = 0$ , which implies  
1020 that  $u = rep(c)$  for some level 0 concept  $c$ . Then, since  $u$  fires at time 0, by definition of "showing",  
1021 an ancestor of  $c$  must be shown at time 0.

1022 *Inductive step:* Assume the five-part claim holds for time  $t - 1$  and consider time  $t$ . We prove the  
1023 five parts one by one.

1024 For Part 1, let  $c$  and  $c'$  be any two distinct concepts for which  $rep(c)$  and  $rep(c')$  are both defined  
1025 by time  $t$ . We must show that  $rep(c) \neq rep(c')$ . If both  $rep(c)$  and  $rep(c')$  are defined by time  
1026  $t - 1$ , then by the inductive hypothesis, Part 1,  $rep(c) \neq rep(c')$  at time  $t - 1$ . Since the *reps* do  
1027 not change, this is still true at time  $t$ , as needed. So the only remaining possibility for conflict is that  
1028 one of these two concepts, say  $c'$ , already has its *rep* defined by time  $t - 1$  and the other concept,  $c$ ,  
1029 does not, and  $rep(c)$  becomes defined at time  $t$ , to be the same neuron as  $rep(c')$ . But we claim that,  
1030 because of the weight settings,  $rep(c)$  must be defined at time  $t$  to be a neuron that is unbound at  
1031 time  $t - 1$ .

1032 So suppose that  $u$  is the neuron that gets defined to be  $rep(c)$  at time  $t$ ; we argue that  $u$  must be  
1033 unbound at time  $t - 1$ . Write  $\ell = level(c)$ ; then also  $layer(u) = \ell$ . By [Assumption A.4](#), the  
1034 *engaged* flag gets set at time  $t$  for  $u$ , and for no other layer  $\ell$  neurons. Since  $c$  is shown at time  $t - \ell$ ,  
1035 by the  $\sigma$ -bottom-up assumption, each child of  $c$  must have been shown at least  $\sigma$  times prior to time  
1036  $t - \ell$ . Then by the inductive hypothesis, Parts 4 and 5, the layer  $\ell - 1$  neurons "fire correctly" at time  
1037  $t - 1$ , that is, all neurons in the set  $rep(children(c))$  fire and no other layer  $\ell - 1$  neuron fires, at  
1038 time  $t - 1$ . This firing pattern implies that every layer  $\ell$  neuron that is already bound strictly prior to  
1039 time  $t$  has incoming potential in round  $t$  that is strictly less than  $k$  times the initial weight, by the  
1040 inductive hypothesis Part 3(a) and by the disjointness of the concepts. On the other hand, every layer  
1041  $\ell$  neuron that is unbound at time  $t - 1$  has incoming potential equal to  $k$  times the initial weight, by  
1042 [Lemma A.5](#). By assumption, there must be at least one unbound neuron available. It follows that  
1043 the neuron  $u$  that is chosen by the WTA is unbound at time  $t - 1$ , and so cannot be the same as the  
1044 already-bound neuron  $rep(c')$ .

1045

1046 For Part 2, let  $c$  be any concept with  $level(c) \geq 1$ , and write  $\ell = level(c)$ . We must prove that any  
1047 showing of  $c$  at any time  $\leq t - \ell$  leads to the same neuron  $u = rep(c)$  becoming engaged. If  $c$  is not  
1048 shown at time precisely  $t - \ell$ , then the claim follows directly from the inductive hypothesis, Part 2.  
1049 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  
1050 defined at time  $t$ , so the conclusion is trivially true (since there is only one showing to consider).

1051 It remains to consider the case where  $rep(c)$  is already defined by time  $t - 1$ . Then, by the inductive  
1052 hypothesis, Part 2, we know that any showing of  $c$  at a time  $\leq t - 1 - \ell$  leads to neuron  $rep(c)$   
1053 becoming engaged. We now argue that the same  $rep(c)$  is also selected at time  $t$ . As in the proof of  
1054 Part 1, the *engaged* flag is set at time  $t$  for exactly one layer  $\ell$  neuron; we claim that this chosen  
1055 neuron is in fact the previously-defined  $rep(c)$ . As in the proof for Part 1, we claim that all neurons  
1056 in the set  $rep(children(c))$  fire and no other layer  $\ell - 1$  neuron fires at time  $t - 1$ . Then  $rep(c)$  has  
1057 incoming potential in round  $t$  that is strictly greater than  $k$  times the initial weight, by the inductive  
1058 hypothesis, Part 3(a). On other hand, every other layer  $\ell$  neuron has incoming potential that is at  
1059 most  $k$  times the initial weight, again by the inductive hypothesis, Part 3(a). It follows that  $rep(c)$   
1060 has a strictly higher incoming potential in round  $t$  than any other layer  $\ell$  neuron, and so is the chosen  
1061 neuron at time  $t$ .

1062

1063 For Part 3, let  $c$  be any concept with  $level(c) \geq 1$ , and write  $\ell = level(c)$ . Let  $t' \geq 1$ . Assume that  $c$   
 1064 is shown at time  $t - \ell$  for the  $t'$ -th time. We must show:

1065 (a) Neuron  $u = rep(c)$  has weights in  $\left(\frac{1}{k^{\ell_{\max} + 1}}, \frac{1}{\sqrt{k}}\right)$  for all neurons in  $rep(children(c))$ , and  
 1066 weights in  $\left(0, \frac{1}{k^{\ell_{\max} + 1}}\right)$  for all other neurons.

1067 (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  
 1068  $rep(children(c))$ , and weights in  $\left[0, \frac{1}{k^{\ell_{\max} + b}}\right]$  for all other neurons.

1069 For both parts, we use Part 2 (for  $t$ , not  $t - 1$ ) to infer that every showing of  $c$  at a time  $\leq t - level(c)$   
 1070 leads to the same neuron  $u = rep(c)$  being engaged. Thus, neuron  $u$  has been engaged  $t'$  times as a  
 1071 result of showing  $c$ , up to time  $t$ .

1072 For Part (a), fix any  $t' \geq 1$ . Then we may apply [Lemma A.2](#), with  $F = rep(children(c))$ , to  
 1073 conclude that the incoming weights for  $u$  are in the claimed intervals. Here we use the fact that the  
 1074 initial settings  $w_i(0)$  are equal to  $\frac{1}{k^{\ell_{\max} + 1}}$ . For Part (b), assume that  $t' \geq \sigma$ . Then we may apply  
 1075 [Lemma A.3](#), with  $F = rep(children(c))$ , to conclude that the incoming weights for  $u$  are in the  
 1076 claimed intervals.  
 1077

1078 For Part 4, let  $c$  be any concept, and assume that  $c^*$ , a proper ancestor of  $c$ , is shown at time  
 1079  $t - level(c)$ . We must show that  $rep(c)$  is defined by time  $t$ , and that it fires at time  $t$ .

1080 Since  $c^*$  is shown at time  $t - level(c)$ , by the definition of a  $\sigma$ -bottom-up schedule, that means  $c$   
 1081 was shown at least  $\sigma$  times by time  $t - level(c) - 1$ . This implies that  $rep(c)$  is defined by time  
 1082  $t - 1$ , and so, by time  $t$ . Moreover, since  $c$  was shown at least  $\sigma$  times by time  $t - level(c) - 1$ ,  
 1083 by the inductive hypothesis, Part 3(b), at time  $t - 1$ ,  $rep(c)$  has incoming weights at least  
 1084  $\frac{1}{(1+\varepsilon)\sqrt{k}}$  for all neurons in  $rep(children(c))$ . By the inductive hypothesis, Part 4, the neurons  
 1085 in  $rep(children(c))$  fire at time  $t - 1$  since  $c^*$  is also a proper ancestor of all children of  $c$ .  
 1086 Therefore, in round  $t$ , the potential of  $rep(c)$  is at least  $k \cdot \frac{1}{(1+\varepsilon)\sqrt{k}}$ , which by our assumptions  
 1087 on the values of the parameters means that the potential is at least  $\tau$ , which implies that  $u$  fires at time  $t$ .  
 1088

1089 For Part 5, fix an arbitrary neuron  $u$  and suppose that  $u$  fires at time  $t$ . We must show that there is  
 1090 some concept  $c$  such that  $u = rep(c)$  at time  $t$ , and a (not necessarily proper) ancestor of  $c$  is shown  
 1091 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  $c$   
 1092 be the (unique) concept such that  $u = rep(c)$ . The firing of  $u$  at time  $t$  is due to the showing of some  
 1093 concept, say  $c^*$ , at time  $t - layer(u)$ .

1094 Let  $R$  be the subset of  $rep(children(c))$  that fire at time  $t - 1$ . We claim that  $|R| \geq 2$ ; that is, at  
 1095 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  
 1096 at time  $t - 1$ , then by the inductive hypothesis, Part 3(a), the total potential incoming to  $u$  in round  $t$   
 1097 would be at most

$$\frac{1}{\sqrt{k}} + \frac{k^{\ell_{\max}}}{k^{\ell_{\max} + 1}} = \frac{1}{\sqrt{k}} + \frac{1}{k} \leq \frac{r_2\sqrt{k}}{2} \leq \tau,$$

1098 where  $\tau$  is the threshold for firing.

1099 Therefore,  $|R| \geq 2$ ; let  $u'$  and  $u''$  be any two distinct elements of  $R$ . Since  $u'$  and  $u''$  fire at time  
 1100  $t - 1$ , by [Lemma A.5](#), we know that both are bound at time  $t - 1$ ; let  $c'$  and  $c''$  be the respective  
 1101 concepts such that  $u' = rep(c')$  and  $u'' = rep(c'')$ . We know that  $c' \neq c''$  because each concept gets  
 1102 only one  $rep$  neuron, by the way that  $rep$  is defined. Note that the firing of both  $u'$  and  $u''$  must be  
 1103 due to the showing of the same concept  $c^*$  at time  $(t - 1) - (layer(u) - 1) = t - layer(u)$ . Then  
 1104 by the inductive hypothesis, Part 5, applied to both  $u'$  and  $u''$ , we see that  $c^*$  must be an ancestor of  
 1105 both  $c'$  and  $c''$ . Therefore,  $c^*$  must be an ancestor of the common parent  $c$  of  $c'$  and  $c''$ , as needed.

1106 This completes the overall proof of the lemma. □

### 1107 A.3 Proof of [Theorem 5.3](#)

1108 Now we use [Lemma A.6](#) to prove our main theorem about noise-free learning, [Theorem 5.3](#).

1109 *Proof.* By assumption, all the concepts in the hierarchy are shown according to a  $\sigma$ -bottom-up  
 1110 training schedule. This implies, by [Assumption A.4](#), that after the schedule, all the concepts in the  
 1111 hierarchy have *reps* in the corresponding layers, that is, for each  $c \in C$ ,  $\text{layer}(\text{rep}(c)) = \text{level}(c)$ .  
 1112 Also, by [Lemma A.6](#), Part 3(b), the weights after the schedule are set as follows: For every concept  
 1113  $c$  with  $\text{level}(c) \geq 1$ , all incoming weights of  $\text{rep}(c)$  from the *reps* of its children, i.e., the neurons  
 1114 in  $\text{rep}(\text{children}(c))$ , are in the range  $[\frac{1}{(1+\varepsilon)\sqrt{k}}, \frac{1}{\sqrt{k}}]$ , and weights from all other neurons (on layer  
 1115  $\text{level}(c) - 1$ ) are in the range  $[0, \frac{1}{k^{\ell_{\max} + b}}]$ .

1116 We must argue that the resulting network  $\mathcal{N}(r_1, r_2)$ -recognizes the concept hierarchy  $\mathcal{C}$ , according to  
 1117 [Definition 4.2](#). This has two directions, saying that certain neurons must fire and certain neurons must  
 1118 not fire, at certain times, when a particular subset  $B \subseteq C_0$  is presented. So suppose that a particular  
 1119 subset  $B \subseteq C_0$  is presented at time  $t$ .

1120 *Neurons that must fire:* We must show that the *rep* of any concept  $c$  in  $\text{supported}_{r_2}(B)$  fires at time  
 1121  $t + \text{level}(c)$  (see [Definition 2.1](#) for the definition of *supported*). We prove this by induction on the  
 1122 level number  $\ell$ ,  $1 \leq \ell \leq \ell_{\max}$ , showing that the *rep* of each level  $\ell$  concept in  $\text{supported}_{r_2}(B)$  fires  
 1123 at time  $t + \text{level}(c)$ .

1124 For the base case, consider a level 1 concept  $c \in \text{supported}_{r_2}(B)$ ; then  $\text{rep}(c)$  is in *layer* 1. Since  
 1125  $c \in \text{supported}_{r_2}(B)$ , it means that  $|\text{children}(c) \cap B| \geq r_2 k$ , that is, at least  $r_2 k$  children of  $c$  are in  
 1126  $B$ . As noted above, the *rep* of each of these children is connected to  $\text{rep}(c)$  by an edge with weight  
 1127 at least  $\frac{1}{(1+\varepsilon)\sqrt{k}}$ , which yields a total incoming potential for  $\text{rep}(c)$  in round 1 of at least

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

1128 To show that  $\text{rep}(c)$  fires at time  $t + 1$ , it suffices to show that the right-hand side is at least as large  
 1129 as the firing threshold  $\tau = \frac{(r_1+r_2)\sqrt{k}}{2}$ . That is, we must show that  $\frac{r_2}{1+\varepsilon} \geq \frac{r_1+r_2}{2}$ . Plugging in the  
 1130 expression for  $\varepsilon$ , we get that:

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

1131 as needed.

1132 For the inductive step, consider  $\ell \geq 2$  and assume by induction that the *rep* of any level  $\ell - 1$  concept  
 1133 in  $\text{supported}_{r_2}(B)$  fires at time  $t + \ell - 1$ . Consider a level  $\ell$  concept  $c \in \text{supported}_{r_2}(B)$ . Since  
 1134  $c \in \text{supported}_{r_2}(B)$ , it means that  $|\text{children}(c) \cap B_{\ell-1}| \geq r_2 k$ , using notation from [Definition 2.1](#),  
 1135 that is, at least  $r_2 k$  children of  $c$  are in  $\text{supported}_{r_2}(B)$ . By the inductive hypothesis, the *reps* of  
 1136 all of these children of  $c$  fire at time  $t + \ell - 1$ . As noted above, the *rep* of each of these children is  
 1137 connected to  $\text{rep}(c)$  by an edge with weight at least  $\frac{1}{(1+\varepsilon)\sqrt{k}}$ , which yields a total incoming potential  
 1138 for  $\text{rep}(c)$  in round  $t + \ell$  of at least

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

1139 Arguing as in the base case, this is at least as large as the firing threshold  $\tau$ , as needed to guarantee  
 1140 that  $\text{rep}(c)$  fires at time  $t + \ell$ .

1141 *Neurons that must not fire:* We must show that the *rep* of any concept  $c$  that is not in  $\text{supported}_{r_1}(B)$   
 1142 does not fire at time  $t + \text{level}(c)$ . Again we prove this by induction on the level number  $\ell$ ,  $1 \leq \ell \leq$   
 1143  $\ell_{\max}$ , showing that the *rep* of each level  $\ell$  concept that is not in  $\text{supported}_{r_1}(B)$  does not fire at time  
 1144  $t + \text{level}(c)$ .

1145 For the base case, consider a level 1 concept  $c \notin \text{supported}_{r_1}(B)$ ; then  $\text{rep}(c)$  is in *layer* 1. Since  
 1146  $c \notin \text{supported}_{r_1}(B)$ , it means that  $|\text{children}(c) \cap B| < r_1 k$ , which implies that  $|\text{children}(c) \cap B| \leq$   
 1147  $\lfloor r_1 k \rfloor$ . As noted above, the *rep* of each of these children is connected to  $\text{rep}(c)$  by an edge with  
 1148 weight at most  $\frac{1}{\sqrt{k}}$ . Also, there are at most  $k^{\ell_{\max} + 1}$  other level 0 firing neurons, since  $B \subseteq C_0$ ,  
 1149 and all the weights on edges connecting these to  $\text{rep}(c)$  are at most  $\frac{1}{k^{\ell_{\max} + b}}$ . Therefore, the total  
 1150 incoming potential for  $\text{rep}(c)$  in round  $t + 1$  is at most

$$\frac{\lfloor r_1 k \rfloor}{\sqrt{k}} + \frac{k^{\ell_{\max} + 1}}{k^{\ell_{\max} + b}} = \frac{\lfloor r_1 k \rfloor}{\sqrt{k}} + \frac{1}{k^{b-1}}.$$

1151 Now we use the technical assumption that  $r_1 k - \lfloor r_1 k \rfloor \geq \frac{\sqrt{k}}{k^{b-1}}$ . Then the right hand side of the last  
 1152 inequality is at most

$$\frac{r_1 k - \frac{\sqrt{k}}{k^{b-1}}}{\sqrt{k}} + \frac{1}{k^{b-1}} = r_1 \sqrt{k} < \frac{(r_1 + r_2) \sqrt{k}}{2} = \tau,$$

1153 which implies that  $\text{rep}(c)$  does not fire.

1154 For the inductive step, consider  $\ell \geq 2$  and assume by induction that the  $\text{rep}$  of any level  $\ell - 1$   
 1155 concept that is not in  $\text{supported}_{r_1}(B)$  does not fire at time  $t + \ell - 1$ . Consider a level  $\ell$  concept  
 1156  $c \notin \text{supported}_{r_1}(B)$ . Since  $c \notin \text{supported}_{r_1}(B)$ , it means that  $|\text{children}(c) \cap B_{\ell-1}| < r_1 k$ , that  
 1157 is, the number of children of  $c$  that are in  $\text{supported}_{r_1}(B)$  is less than  $r_1 k$ . As noted above, the  $\text{rep}$   
 1158 of each of these children is connected to  $\text{rep}(c)$  by an edge with weight at most  $\frac{1}{\sqrt{k}}$ .

1159 Now consider the rest of the incoming edges to  $\text{rep}(c)$ . They may come from the  $\text{reps}$  of children of  
 1160  $c$  that are not in  $\text{supported}_{r_1}(B)$ , from layer  $\ell - 1$  neurons that are bound to concepts that are not  
 1161 children of  $c$ , and from unbound layer  $\ell - 1$  neurons. However, the  $\text{reps}$  of children of  $c$  that are not  
 1162 in  $\text{supported}_{r_1}(B)$  do not fire, by the inductive hypothesis, and the unbound neurons do not fire, by  
 1163 [Lemma A.5](#). So that leaves us to consider the layer  $\ell - 1$  neurons that are bound to concepts in  $C$   
 1164 that are not children of  $c$ . There are at most  $k^{\ell_{\max}} + 1$  such neurons. Since the weights of the edges  
 1165 connecting them to  $\text{rep}(c)$  are at most  $\frac{1}{k^{\ell_{\max} + b}}$ , the total incoming potential for  $\text{rep}(c)$  in round  $t + \ell$   
 1166 is at most

$$\frac{\lfloor r_1 k \rfloor}{\sqrt{k}} + \frac{k^{\ell_{\max} + 1}}{k^{\ell_{\max} + b}} = \frac{\lfloor r_1 k \rfloor}{\sqrt{k}} + \frac{1}{k^{b-1}}.$$

1167 As in the base case, this is strictly less than  $\tau$ . Therefore,  $\text{rep}(c)$  does not fire at time  $t + \text{level}(c)$ .  $\square$

## 1168 B Analysis of Noisy Learning

1169 Here we present our analysis for the noisy learning algorithm in [Section 6](#). In [Lemma B.1](#), we  
 1170 describe how incoming weights change for a particular neuron when it is noisy-shown. The proof can  
 1171 be found in [Section B.4](#). Once we understand the weight changes of one neuron, we are able to use  
 1172 essentially the same invariants as in the noise-free case ([Lemma A.6](#)), describing how neurons get  
 1173 bound to concepts, when neuron firing occurs, and how weights change, during the time when the  
 1174 network is learning. In [Section B.3](#), we put everything together to prove [Theorem 6.4](#).

1175 We start by giving a slightly more detailed proof overview than the one in [Section 6.3](#).

### 1176 B.1 Proof Overview

1177 The overall proof of [Theorem 6.4](#) is at its core similar to the proof of [Theorem 5.3](#) presented in  
 1178 [Appendix A](#). The main difference is that the weights of the neurons after learning are slightly different:  
 1179 following the notation of [Lemma A.1](#), [Lemma A.2](#) and [Lemma A.3](#), we show that, for every  $i \in F$ ,  
 1180 the weight will eventually approximate

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

1181 and for every  $i \notin F$ , the weight will eventually be in the interval  $[0, 1/k^{2\ell_{\max}}]$ . Note that, in this  
 1182 section, we set the parameter  $b$ , governing the desired decrease of unrelated weights, to be  $b = \ell_{\max}$ .  
 1183 Also note that we can recover the noise-free case by setting  $p = 1$ .<sup>10</sup>

1184 The main difficulty in the noisy case is to establish a noisy version of [Lemma A.3](#), which we do in  
 1185 [Lemma B.1](#). Then, proving the main theorem is analogous to the noise-free case. This is because the  
 1186 behavior of this network is the same as that of the noise-free algorithm, except for how the weights of  
 1187 individual neurons are updated. Nonetheless, the same arguments as in the proof [Lemma A.6](#) still  
 1188 hold. Therefore, the core of this section is to prove [Lemma B.1](#). Due to the noise, main structural  
 1189 properties of the noise-free case, such as weights of neurons in  $F$  changing monotonically, do not  
 1190 hold anymore. To make matters worse, we cannot simply use Chernoff bounds and assume the

<sup>10</sup>In this case the probabilistic guarantees become deterministic guarantees.

1191 worst-case distribution of the weight changes, since assuming worst-case in each round prevents the  
 1192 weights from converging. Instead, we use a fine-grained potential analysis.

1193 We first bound the worst-case change of any weight  $w_i$  during a period of  $T$  rounds ([Lemma B.2](#)),  
 1194 assuming that the weight at the beginning of the period,  $w_i(t)$ , is in the interval  $[\frac{\sqrt{p}}{4k}, \frac{4}{\sqrt{p}}]$ . Namely,  
 1195 we show that for some small  $\delta_1$  (defined in [Section B.2](#)), we have  $(1 - \delta_1)w_i(t) \leq w_i(t + T) \leq$   
 1196  $(1 + \delta_1)w_i(t)$ . We later show that this assumption holds w.h.p. throughout the first  $n^6$  rounds. It  
 1197 turns out that the way an individual weight changes depends strongly on the other weights in  $F$  and  
 1198 on the neurons of the previous layer that fire. More precisely, it depends on  $z(t)$ , which can change  
 1199 dramatically between rounds, rendering the analysis non-trivial. In order to show that the weights  
 1200 converge to  $\bar{w}$ , we use the potential function  $\psi(\cdot)$ . For any time  $t$ , let  $w_{min}(t)$  and  $w_{max}(t)$  be the  
 1201 minimum and maximum weight, respectively, among  $\{w_i(t) \mid i \in F\}$ . Let

$$\psi(t) = \max \left\{ \frac{w_{max}(t)}{\bar{w}}, \frac{\bar{w}}{w_{min}(t)} \right\}.$$

1202 Our goal is to show that this potential decreases quickly until it is very close to 1. Showing that the  
 1203 potential decreases is involved, since one cannot simply use a worst-case approach, due to the terms  
 1204 in Oja's rule being non-linear and potentially having a high variance, depending on the distribution of  
 1205 weights. Instead, we consider the terms  $\bar{w}/w_{min}(t)$  and  $w_{max}(t)/\bar{w}$  of the potential and consider  
 1206 four cases depending on whether these terms are small or large.

1207 First, if the term  $\bar{w}/w_{min}(t)$  is large and the term  $w_{max}(t)/\bar{w}$  is small, then the minimum weight  
 1208  $w_{min}$  increases and since the maximum weight  $w_{max}$  increases by at most a factor of  $(1 + \delta)$ , the  
 1209 potential decreases. The second case, where the term  $w_{max}(t)/\bar{w}$  is large and the term  $\bar{w}/w_{min}(t)$  is  
 1210 small, can be bounded analogously. Finally, if  $\bar{w}/w_{min}(t)$  and  $w_{max}(t)/\bar{w}$  are both large and close  
 1211 to each other, then we show that both terms decrease. Note that if both terms are small, then the  
 1212 potential is small and we are done.

1213 For example, to prove the first case, we first show that, for every  $i \in F$  with  $w_i(t) \geq (1 + 2\delta_1)w_{min}$ ,  
 1214 we have  $w_i(t + T) \geq (1 + \delta/2)w_{min}$ , using the previously established bounds. As mentioned before,  
 1215 in order to prove that any such neuron  $i^*$  increases its weight, we cannot use worst-case bounds.  
 1216 Instead, we carefully use the randomness over the input vector  $x$ . To this end we define, for every  
 1217  $t' \geq 0$ ,

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

1218 and

$$S = \sum_{t'=1}^T X(t'). \quad (2)$$

1219 Based on these terms we construct a Doob martingale ([Lemma B.4](#)), which allows us to get asymptot-  
 1220 ically almost tight bounds on  $S$ . To do this, we use the Azuma-Hoeffding inequality ([Theorem C.1](#)).  
 1221 Putting everything together, we see that  $\psi(\cdot)$  decreases. This then allows us to prove [Theorem 6.4](#).

## 1222 **B.2 Convergence of the Weights**

1223 We use the following assumptions about the various parameters:

- 1224 1.  $\delta = \bar{w}(r_2 - r_1)/50$ ,
- 1225 2.  $\delta_1 = \frac{\delta p \bar{w}}{20}$ ,
- 1226 3.  $T = \frac{7 \log(|C|n)}{100 p^3 \delta_1^2}$ ,
- 1227 4. The learning rate  $\eta = \frac{\delta_1^3}{64 T k^2 p}$ .
- 1228 5. The firing threshold  $\tau = r_2 k (\bar{w} - 2\delta)$
- 1229 6.  $b = \ell_{\max}$ .

1230 The following lemma is the noisy counterpart to [Lemma A.3](#).

1231 **Lemma B.1** (Learning Properties, Noisy Case). Let  $F \subseteq \{1, \dots, n\}$  with  $|F| = k$ . Let  $\varepsilon \in (0, 1]$ .

1232 Let  $\sigma = c' \frac{k^6}{p^6 \delta^3} (\ell_{\max} \log(k) + \log(|C|n/\delta))$ , for some large enough constant  $c'$ .

1233 Assume that:

1234 1. For every  $t \geq 0$ ,  $x_i(t) = 0$  for every  $i \notin F$ , and  $e(t) = 1$ .

1235 2. All weights  $w_i(0)$  are equal to  $\frac{1}{k}$ .

1236 3.  $\eta$  is defined above.<sup>11</sup>

1237 Then for every  $t \in [\sigma, n^6]$ , the following with high probability:

1238 1. For any  $i \in F$ , we have  $w_i(t) \in [\bar{w} - 2\delta, \bar{w} + 2\delta]$ .

1239 2. For any  $i \notin F$ , we have  $w_i(t) \leq \frac{1}{k^2 \ell_{\max}}$ .

1240 Proving [Lemma B.1](#) is the main goal of the section and we need a series of properties to prove it.  
1241 We give the proof in [Section B.5](#). We now proceed by showing how [Theorem 6.4](#) follows from this  
1242 lemma.

### 1243 **B.3 Proof of [Theorem 6.4](#), assuming [Lemma B.1](#)**

1244 As mentioned at the beginning of this section, it suffices to consider the learning of one concept.  
1245 Generalizing to a concept hierarchy is analogous to the noise-free case (in particular the proof of  
1246 [Lemma A.6](#)).

1247 We now argue how the learning of one concept follows from [Lemma B.1](#). By [Lemma B.1](#), all  
1248 weights in  $F$  are at least  $\bar{w} - 2\delta$  and most  $\bar{w} + 2\delta$ . Hence, if  $c \in \text{supported}_{r_2}(B)$ , then we can  
1249 show by a similar induction as in the proof of [Theorem 5.3](#) that each  $\text{rep}$  fires since, the potential  
1250 is at least  $r_2 k (\bar{w} - 2\delta) = \tau$ , which means that the corresponding  $\text{rep}$  fires. On the other hand,  
1251 if  $c \notin \text{supported}_{r_1}(B)$ , then there will be a neuron that does not fire since all weights are, by  
1252 [Lemma B.1](#), at most  $\bar{w} + 2\delta$ .

Note that, by definition of  $\delta$ ,

$$\begin{aligned} r_1(\bar{w} + 2\delta) &= (r_2 - 50\delta/\bar{w})(\bar{w} + 2\delta) \\ &\leq r_2 \bar{w} + 2\delta r_2 - 50\delta \\ &\leq r_2 \bar{w} - 2\delta r_2 - 46\delta, \end{aligned}$$

1253 since  $r_2 \leq 1$ . Therefore, the potential for  $\text{rep}(c)$  will be at most

$$r_1 k (\bar{w} + 2\delta) + k^{\ell_{\max}} \frac{1}{k^2 \ell_{\max}} < r_2 k \left( \bar{w} - 2\delta - \frac{46\delta}{r_2} \right) + \frac{1}{k} \leq r_2 k (\bar{w} - 2\delta) = \tau,$$

1254 since  $k46\delta = k \frac{46}{50} \bar{w} (r_2 - r_1) \geq \frac{46}{50\sqrt{k}} \geq 1/k$ , due to  $\bar{w} \geq 1/\sqrt{k}$ ,  $r_2 - r_1 \geq 1/k$  and  $k \geq 2$ . Thus,  
1255 the neuron does not fire.

### 1256 **B.4 Towards [Lemma B.1](#)**

1257 In this subsection, we define a key property  $\mathcal{E}_t$  that says that the weights remain within certain  
1258 multiplicative bounds, for during the interval  $[t, t + T]$  rounds. We show in [Lemma B.2](#) that  $\mathcal{E}_t$  holds  
1259 with probability 1. Then we assume  $\mathcal{E}$  and show [Lemma B.3](#), which bounds the expected change  
1260 of the terms in Oja's rule. To derive bounds on the actual change we first show how the changes  
1261 form a Doob-martingale ([Lemma B.4](#)). Using this, we are finally able to show in in [Lemma B.5](#) and  
1262 [Lemma B.6](#) that the potential decreases.

1263 Let  $\mathcal{E}_t$  be the event that for every  $t' \in [t, t + T]$ , we have

$$(1 - \delta_1) w_i(t) \leq w_i(t') \leq (1 + \delta_1) w_i(t).$$

<sup>11</sup>This is a very precise assumption but it could be weakened, at a corresponding cost in run time.



1264 **Lemma B.2.** Assume  $w_i(t) \in [\frac{\sqrt{p}}{4k}, \frac{4}{\sqrt{p}}]$ . Then,  $\mathcal{E}_t$  holds.

1265 *Proof.* Let  $w_{max}(t)$  denote the maximum weight at time  $t$ . We have  $w_{max}(t+1) \leq$   
 1266  $w_{max}(t) + \eta z(t) \leq w_{max}(t) + \eta w_{max}(t)kp$ . Thus,  $w_{max}(t+t') \leq w_{max}(t)(1 + \eta kp)^T =$   
 1267  $w_{max}(t) \left(1 + \frac{\eta kpT}{T}\right)^T = w_{max}(t)e^x$  for  $x = \eta kpT$ . Since  $p \geq 1/k$ , we have  $x < 1$ , we have

$$w_{max}(t+t') \leq w_{max}(t)e^x \leq w_{max}(t)(1 + x + x^2) \leq w_{max}(t)(1 + 2x).$$

1268 this completes the upper bound of  $\mathcal{E}_t$  since  $2\eta kpT \leq \delta_1$ .

1269 We now consider the lower bound of  $\mathcal{E}_t$ . Similarly, if  $w_{min}(t)$  denotes the minimum weight at time  $t$ ,  
 1270 then

1271  $w_{min}(t+1) \geq w_{min}(t) - \eta z^2(t) \geq w_{min}(t) - \eta w_{max}^2(t)k^2p^2 \geq w_{min}(t) - \eta 16k^2p$ . Thus  $w_{min}(t+1) \geq$   
 1272  $w_{min}(t) - T\eta 16k^2p \geq w_{min}(t) - T\eta 16k^2p \frac{1}{\sqrt{p}/(4k)} w_{min}(t) \geq w_{min}(t) (1 - 64\eta Tk^2\sqrt{p}) \geq$   
 1273  $w_{min}(t)(1 - \delta_1)$ , since  $w_{min}(t) \geq \frac{\sqrt{p}}{4k}$ .  $\square$

1274

1275 We define the following potential function

$$\phi(t) = \sum_{i \in F} w_i(t).$$

1276 The following bounds the expected change of the weights.

1277 **Lemma B.3.** Suppose  $\mathcal{E}_t$  holds. Then, we have

- 1278 1.  $\mathbb{E} [z(t+t') | w(t+t'), \mathcal{F}_t] = p\phi(t+t')$   
 1279 2.  $\mathbb{E} [z(t+t')^2 w_{i^*}(t+t') | \mathcal{F}_t] \leq (1 + \delta_1)^3 p\phi(t) ((1-p)w_{max}(t)w_{i^*}(t) + pw_{i^*}(t)\phi(t))$   
 1280  
 1281 3.  $\mathbb{E} [z(t+t')^2 w_{i^*}(t+t') | \mathcal{F}_t] \geq (1 - \delta_1)^3 p\phi(t) ((1-p)w_{min}(t)w_{i^*}(t) + pw_{i^*}(t)\phi(t)).$

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

$$\mathbb{E} [z(t+t') | w(t+t'), \mathcal{F}_t] = p \sum_{i \in F} \mathbb{E} [x_i(t+t')] w_i(t+t') = p \sum_{i \in F} w_i(t+t') = p\phi(t+t').$$

Moreover,

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

1282 We suppose  $\mathcal{E}_t$  holds, thus in every obtainable configuration it must hold that  $(1 - \delta_1)w_i(t) \leq$   
 1283  $w_i(t+t') \leq (1 + \delta_1)w_i(t)$ . Therefore,  $(1 - \delta_1)\phi(t) \leq \phi(t+t') \leq (1 + \delta_1)\phi(t)$ . Thus,

$$\begin{aligned}
& \mathbb{E} \left[ z(t+t')^2 w_{i^*}(t+t') \mid \mathcal{F}_t \right] = \\
& = \sum_{w' \wedge w' \text{ obtainable}} \mathbb{E} \left[ z(t+t')^2 w_{i^*}(t+t') \mid w(t+t') = w', \mathcal{F}_t \right] \mathbb{P} [w(t+t') = w'] \\
& = \sum_{w' \wedge w' \text{ obtainable}} w'_{i^*}(t+t') \mathbb{E} \left[ z(t+t')^2 \mid w(t+t') = w', \mathcal{F}_t \right] \mathbb{P} [w(t+t') = w'] \\
& \leq (1 + \delta_1) w_{i^*}(t) \sum_{w' \wedge w' \text{ obtainable}} \left( (p - p^2) \sum_{i \in F} w'_i(t+t')^2 + p^2 \phi(t+t')^2 \right) \mathbb{P} [w(t+t') = w'] \\
& \leq (1 + \delta_1)^3 w_{i^*}(t) \left( (p - p^2) \sum_{i \in F} w_i(t)^2 + p^2 \phi(t)^2 \right) \\
& \leq w_{i^*}(t) (1 + \delta_1)^3 \left( (p - p^2) w_{max}(t) \phi(t) + p^2 \phi(t)^2 \right) \\
& \leq (1 + \delta_1)^3 p \phi(t) \left( (1 - p) w_{max}(t) w_{i^*}(t) + p w_{i^*}(t) \phi(t) \right).
\end{aligned}$$

1284 Similarly,

$$\mathbb{E} \left[ z(t+t')^2 w_{i^*}(t+t') \mid \mathcal{F}_t \right] \geq (1 - \delta_1)^3 p \phi(t) \left( (1 - p) w_{min}(t) w_{i^*}(t) + p w_{i^*}(t) \phi(t) \right).$$

1285

□

1286 In the following, we define a sequence of random variables  $Y_1, Y_2, \dots$  and show it forms a Doob  
1287 martingale.

1288 **Lemma B.4.** *Fix neuron  $i^*$ . Let  $X_i$  be the random choices of the  $pk$  children that fire*  
1289 *in round  $i$  (in the definition of the noisy learning). Recall that  $S = \sum_{t' \leq T} z(t+t')$ .*  
1290  *$(X_{i^*}(t+t') - z(t+t')) \cdot w_{i^*}(t+t')$ . Let  $Y_i = \mathbb{E} [S \mid X_i, \dots, X_1]$ . Then the following holds*

1291 1. *The sequence  $Y_0, Y_1, \dots, Y_T$  is a (Doob) martingale with respect to the sequence*  
1292  *$X_0, X_1, \dots, X_T$ .*

1293 2. *For all  $i$ ,  $|Y_i - Y_{i+1}| \leq 8k^2 \sqrt{p}$ .*

1294 3.  *$S = \mathbb{E} [S \mid X_T, \dots, X_1] = Y_T$ .*

1295 *Proof.* For the first part, we have, using the tower rule,

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

1296 For the second part, note that  $w_i \leq 2/\sqrt{p}$ . Thus,  $|Y_i - Y_{i+1}| \leq z_{t+i}^2 w_{i^*} \leq k^2 p^2 2^3 / \sqrt{p}^3$ .

1297 The third part follows trivially.

1298

□

1299 Let

$$\delta_2 = \left( k \frac{\sqrt{p}}{2k} \right) p^2 \left( \frac{20\delta_1}{p} \right) = 10p^{3/2} \delta_1$$

1300 The following lemma shows that if the potential is large due to  $w_{min}$  being small, then the weight of  
1301 the smallest neurons increases.

**Lemma B.5.** *Suppose  $\mathcal{E}_t$  holds. Consider the neurons  $i^*$  with  $w_{i^*}(t) \in [w_{min}, (1 + 2\delta_1)w_{min}]$  and  
 $\bar{w} - w_{i^*}(t) \geq \delta$ . Assume*

$$\frac{\bar{w}}{w_{min}(t)} \geq (1 - 2\delta_1) \frac{w_{max}(t)}{\bar{w}}. \tag{3}$$

1302 *Then, with probability at least  $1 - 1/n^6$ ,*

$$w_{i^*}(t+T) \geq w_{i^*}(t) + T\eta\delta_2/2$$

*Proof.* By the second part of [Lemma B.3](#), for  $t' \leq T$

$$\mathbb{E} [z(t+t')^2 w_{i^*}(t+t')] \leq (1+\delta_1)^3 p \phi(t) ((1-p)w_{max}(t)w_{i^*}(t) + pw_{i^*}(t)\phi(t)).$$

1303 We now bound the terms in the parentheses. First note that

$$w_{i^*}(t)w_{max}(t) \leq (1+2\delta_1)w_{min}(t)w_{max}(t) \leq \frac{1+2\delta_1}{1-2\delta_1}\bar{w}^2 \leq (1+4.5\delta_1)\bar{w}^2,$$

since  $\delta_1 \in [0, 1/18]$ . Furthermore, for  $\delta_1 \in [0, 1/9]$  we have  $(1+4.5\delta_1)(1+\delta_1) \leq (1+6\delta_1)$ . Thus,

$$\begin{aligned} w_{i^*}(t)\phi(t) &\leq (k-1)(1+\delta_1)w_{i^*}(t)w_{max}(t) + (1+\delta_1)w_{i^*}(t)w_{i^*}(t) \\ &\leq (k-1)(1+\delta_1)(1+4.5\delta_1)\bar{w}^2 + (1+\delta_1)w_{i^*}(t)w_{i^*}(t) \\ &\leq (1+6\delta_1) \left( (k-1)\bar{w}^2 + w_{i^*}(t)^2 \right) \\ &= (1+6\delta_1) (k\bar{w}^2 + w_{i^*}(t)^2 - \bar{w}^2). \end{aligned}$$

Note that  $(1-p)\bar{w}^2 + pk\bar{w}^2 = 1$ . Thus,

$$\begin{aligned} (1-p)w_{max}(t)w_{i^*}(t) + pw_{i^*}(t)\phi(t) &\leq (1+6\delta_1) \left( (1-p)\bar{w}^2 + pk\bar{w}^2 + p(w_{i^*}(t)^2 - \bar{w}^2) \right) \\ &= (1+6\delta_1) (1-p(\bar{w}^2 - w_{i^*}(t)^2)). \end{aligned}$$

Therefore,

$$\mathbb{E} [z(t+t')^2 w_{i^*}(t+t')] \leq (1+10\delta_1)p\phi(t) (1-p(\bar{w}^2 - w_{i^*}(t)^2)),$$

1304 where we used that  $(1+6x)(1+x)^3 \leq (1+10x)$  for  $x \leq 0.045$ .

Note that

$$\bar{w}^2 - w_{i^*}(t)^2 \geq \bar{w}^2 - w_{i^*}(t)\bar{w} = \bar{w}(\bar{w} - w_{i^*}(t)) \geq \bar{w}\delta = \bar{w}\frac{20}{\bar{w}p}\delta_1. \quad (4)$$

Finally, using the definition of  $S$  ([Equation 2](#)) and combining the above with the first part of [Lemma B.3](#),

$$\begin{aligned} \mathbb{E} [S] &\geq T (\mathbb{E} [z(t+t')] - \mathbb{E} [z(t+t')^2 w_{i^*}(t+t')]) \\ &\geq T\phi(t)p (1 - (1+10\delta_1) (1-p(\bar{w}^2 - w_{i^*}(t)^2))) \\ &\geq T\phi(t)p^2 \frac{\bar{w}^2 - w_{i^*}(t)^2}{2}, \end{aligned}$$

1305 where we used that  $1 - (1+z)(1-x) = 1 - (1-x+z-zx) = x - z + zx \geq x/2$  for  $z \leq x/2$ . We  
1306 define the sequence  $Y_1, Y_2, \dots$  of variabls as defined in [Lemma B.4](#). By [Lemma B.4](#), this sequence  
1307 is a Doob martingale. Thus, we can apply [Theorem C.1](#) to the Doob martingale  $Y_T, Y_{T-1}, \dots, Y_1$   
1308 with  $|Y_i - Y_{i+1}| \leq \delta_3$  for  $\delta_3 = 8k^2\sqrt{p}$ .

We derive using the lower bounds on the weights and [Equation 4](#).

$$\begin{aligned} \mathbb{P} \left[ |S - \mathbb{E} [S]| \geq \frac{\mathbb{E} [S]}{2} \right] &\leq 2 \exp \left( -\frac{2 \left( \frac{\mathbb{E} [S]}{2} \right)^2}{T\delta_3^2} \right) \leq 2 \exp \left( -\frac{2 \left( T\phi(t)p^2 \frac{\bar{w}^2 - w_{i^*}(t)^2}{2} \right)^2}{T\delta_3^2} \right) \\ &\leq 2 \exp \left( -\frac{T(\phi(t)p^2(\bar{w}^2 - w_{i^*}(t)^2))^2}{4\delta_3^2} \right) \leq 2 \exp(-7\ell_{\max} \log(|C|n)) \leq \frac{1}{|C|n^6}, \end{aligned}$$

where the last inequality follows from

$$T(\phi(t)p^2(\bar{w}^2 - w_{i^*}(t)^2))^2 \geq T\delta_2^2 = 100Tp^3\delta_1^2 = 7\log(|C|n).$$

Thus

$$w_{i^*}(t+T) \geq w_{i^*}(t) + \eta S \geq w_{i^*}(t) + \eta \mathbb{E} [S] / 2 \geq w_{i^*}(t) + T\eta\delta_2/2$$

1309

□

1310 The following lemma is analogous to the previous one, with the difference that we analyse the case  
 1311 where  $\psi$  is dominated by large weights (rather than small) and show that these large weights decrease.

**Lemma B.6.** *Suppose  $\mathcal{E}_t$  holds. Consider the neurons  $i^*$  with  $w_{i^*}(t) \in [w_{max}(1 - 2\delta_1), w_{max}]$  and  $w_{i^*}(t) - \bar{w} \geq \delta$ . Assume*

$$\frac{w_{max}(t)}{\bar{w}} \geq (1 - 2\delta_1) \frac{\bar{w}}{w_{min}(t)} \quad (5)$$

1312 Then, with probability at least  $1 - 1/n^6$ ,

$$w_{i^*}(t + T) \leq w_{i^*}(t) - T\eta\delta_2/2$$

*Proof.* We have for all  $i \in F$  with  $w_i(t) \geq (1 + 2\delta_1)w_{min}$ , we have  $w_i(t + T) \geq (1 + \delta_1/2)w_{min}$ , since each weight can only decrease by a factor of  $(1 - \delta_1)$  and since  $(1 + 2\delta_1)(1 - \delta_1) = 1 + \delta_1 - 2\delta_1 \geq (1 + \delta/2)$ . Thus, we only consider the neurons  $i^*$  with  $w_{i^*}(t) \in [w_{min}, (1 + 2\delta_1)w_{min}]$ . By the third part of [Lemma B.3](#), for  $t' \leq T$

$$\mathbb{E} [z(t + t')^2 w_{i^*}(t + t')] \geq (1 - \delta_1)^3 p\phi(t) ((1 - p)w_{min}(t)w_{i^*}(t) + pw_{i^*}(t)\phi(t)).$$

1313 We now bound the terms in the parentheses. First note that

$$w_{i^*}(t)w_{min}(t) \geq (1 - 2\delta_1)w_{min}(t)w_{max}(t) \geq (1 - 2\delta_1)^2 \bar{w}^2 \geq (1 - 4\delta_1)\bar{w}^2,$$

1314 since  $\delta_1 \geq 0$ .

Thus,

$$\begin{aligned} (1 - p)w_{min}(t)w_{i^*}(t) + pw_{i^*}(t)\phi(t) &\geq (1 - 4\delta_1) ((1 - p)\bar{w}^2 + pk\bar{w}^2 + p(w_{i^*}(t)^2 - \bar{w}^2)) \\ &= (1 - 4\delta_1) (1 - p(\bar{w}^2 - w_{i^*}(t)^2)) \end{aligned}$$

Therefore,

$$\mathbb{E} [z(t + t')^2 w_{i^*}(t + t')] \geq (1 - 10\delta_1)p\phi(t) (1 - p(\bar{w}^2 - w_{i^*}(t)^2)),$$

1315 where we used that  $(1 - 4x)(1 - x)^3 \geq (1 - 10x)$  for  $x \geq 0$ .

Note that

$$\bar{w}(w_{i^*}(t) - \bar{w}) \geq \bar{w}\delta = \bar{w} \frac{20}{wp} \delta_1 \quad (6)$$

Finally, using the definition of  $S$  ([Equation 2](#)) and combining the above with the first part of [Lemma B.3](#),

$$\begin{aligned} \mathbb{E} [S] &\leq T (\mathbb{E} [z(t + t')] - \mathbb{E} [z(t + t')^2 w_{i^*}(t + t')]) \\ &\leq T\phi(t)p (1 - (1 - 10\delta_1) (1 - p(\bar{w}^2 - w_{i^*}(t)^2))) \\ &\leq 2T\phi(t)p^2 \bar{w}^2 - w_{i^*}(t)^2 = -2T\phi(t)p^2 (w_{i^*}(t)^2 - \bar{w}^2) \\ &\leq -2T\phi(t)p^2 \bar{w}(w_{i^*}(t) - \bar{w}), \end{aligned}$$

1316 where we used that  $1 - (1 - z)(1 - x) = 1 - (1 - x - z + zx) = x - z + zx \leq 2x$  for  $z \leq 1$ .

1317 This allows us to apply [Theorem C.1](#) and the rest is analogous.

Thus

$$w_{i^*}(t + T) \leq w_{i^*}(t) + \eta S \leq w_{i^*}(t) + \eta \mathbb{E} [S] / 2 \leq w_{i^*}(t) - T\eta\delta_2/2$$

1318 □

1319 We have for all  $i \in F$  with  $w_i(t) \geq (1 + 2\delta_1)w_{min}$ , we have  $w_i(t + T) \geq (1 + \delta_1/2)w_{min}$ , since each  
 1320 weight can only decrease by a factor of  $(1 - \delta_1)$  and since  $(1 + 2\delta_1)(1 - \delta_1) = 1 + \delta_1 - 2\delta_1 \geq (1 + \delta/2)$ .

1321 Note that if neither [Equation 3](#) nor [Equation 5](#) applies, then both  $w_{min}(t)$  and  $w_{max}(t)$  must be close  
 1322 to  $\bar{w}$  and the claim follows easily.

1323 **B.5 Proof of Lemma B.1**

1324 We argue by induction on  $j$ , that  $\psi(j \cdot T) \leq \max(\psi(0) - jT\eta\delta_2/2, \bar{w} + 2\delta)$  with probability at  
 1325 least  $1 - j/(|C|n^6)$ . The base case is trivial. Assume the claim holds up to  $j - 1$ . We have

1326  $w_i((j - 1)T) \in [\frac{\sqrt{p}}{4k}, \frac{4}{\sqrt{p}}]$ . Therefore, by Lemma B.2  $\mathcal{E}_{(j-1)T, T}$  holds. This allows us to apply  
 1327 Lemma B.5 and Lemma B.6.

Consider the following equations

$$\frac{\bar{w}}{w_{min}(t)} \geq (1 - 2\delta_1) \frac{w_{max}(t)}{\bar{w}}. \quad (7)$$

$$\frac{w_{max}(t)}{\bar{w}} \geq (1 - 2\delta_1) \frac{\bar{w}}{w_{min}(t)} \quad (8)$$

1328 We consider four cases based on whether or not the two equations Equation 7 and Equation 8 hold.  
 1329 In the first case Equation 7 holds and Equation 8 does not. In this case we can bound the drop of  $\psi()$   
 1330 by considering the the increase of  $w_{min}()$  and we can disregard the increase of  $w_{max}()$ , since even if  
 1331 it increases by a factor of  $(1 + \delta_1)$ , we have

$$\frac{w_{max}(jT)}{\bar{w}} \leq (1 + \delta_1) \frac{w_{max}((j - 1)T)}{\bar{w}} \leq (1 + \delta_1)(1 - 2\delta_1) \frac{\bar{w}}{w_{min}((j - 1)T)} \leq (1 - \delta_1) \frac{\bar{w}}{w_{min}((j - 1)T)}.$$

1332 In the second case Equation 8 holds and Equation 7 does not. This case is analogous to the first case.

1333 In the third case Equation 7 and Equation 8 hold. Here, one can show that both the minimum weight  
 1334 increases, and the maximum weight decreases.

1335 In the fourth case, none of the equations hold. This yields a contradiction

$$\frac{\bar{w}}{w_{min}(t)} < (1 - 2\delta_1) \frac{w_{max}(t)}{\bar{w}} < (1 - 2\delta_1)^2 \frac{\bar{w}}{w_{min}(t)}.$$

1336 Thus we can disregard this case.

1337 W.l.o.g. we assume the first case holds.

1338 Consider the neurons  $i^*$  with  $w_{i^*}(t) \in [w_{min}, (1 + 2\delta_1)w_{min}]$  and  $\bar{w} - w_{i^*}(t) \geq \delta$ . Then, by  
 1339 Lemma B.5, with probability at least  $1 - 1/n^6$ ,

$$w_{i^*}(t + T) \geq w_{i^*}(t) + T\eta\delta_2/2 \geq w_{i^*}(t) + w_{i^*}(t) \frac{T\eta\delta_2}{2(4\sqrt{p})}.$$

1340 Note that in the analogous cases two and three we have for any neurons  $i^*$  with  $w_{i^*}(t) \in [w_{max}(1 -$   
 1341  $2\delta_1), w_{max}]$  that

$$w_{i^*}(t + T) \leq w_{i^*}(t) - T\eta\delta_2/2 \leq w_{i^*}(t) - w_{i^*}(t) \frac{T\eta\delta_2}{2(4\sqrt{p})}.$$

1342 Let  $\delta_4 = T\eta\delta_2/(8\sqrt{p})$ . Thus, either way

$$\psi(jT) \leq (1 - \delta_4)\psi((j - 1)T).$$

1343 Using the fact that  $\log(1 + x) \geq 2x$  for  $x \in (-1/2, 0)$ , we get that after

$$j^* = \log_{1-\delta_4}(\delta/\psi(0)) = \frac{\log(\delta/\psi(0))}{\log(1 - \delta_4)} \leq \frac{\log(\delta/\psi(0))}{-2\delta_4} = \frac{\log(\psi(0)/\delta)}{2\delta_4}$$

1344 intervals of length  $T$  the  $\psi()$  is within an error of at most  $2\delta$  and stays there by assumption for  $n^6$   
 1345 rounds. Thus the total number of rounds is  $Tj^*$ . The bound from the claim follows by observing that  
 1346 term  $\eta T/\delta_4$  is a small polynomial in  $p$  and  $w$  and  $\delta$ .

1347 Finally, we consider the time required for weights  $i \notin F$  to decrease below  $k^{-2\ell_{\max}}$ . After the  
 1348 weights in  $F$  are close to their target, we have that  $z(t) \geq pk\bar{w}/2$ . Thus at this point, the weights  
 1349 decrease changes as follows every round

$$w_i(t) = w_i(t-1)(1 - \eta z(t-1)^2) \geq w_i(t-1)(1 - \eta p^2 k^2 \bar{w}^2 / 4).$$

1350 Thus, the potential halves every  $20/(\eta p^2 k^2 \bar{w}^2)$  rounds. Since the potential only needs to drop by a  
 1351 factor of  $k^{2\ell_{\max}}$ , the bound follows.

## 1352 C Auxiliary Content

1353 The following is a slightly modified version of Theorem 5.2 in [5], which we use in [Lemma B.5](#) and  
 1354 [Lemma B.6](#).

1355 **Theorem C.1** (Azuma-Hoeffding inequality - general version [5]). *Let  $Y_0, Y_1, \dots$  be a martingale  
 1356 with respect to the sequence  $X_0, X_1, \dots$ . Suppose also that  $Y_i$  satisfies  $a_i \leq Y_i - Y_{i-1} \leq b_i$  for all  
 1357  $i$ . As an example, the engaged flag could be used to ensure that, in any round, only one neuron in the  
 1358 network is prepared to learn.*

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