A Theory on the Completeness of the DN Logic Capability

Juyang Weng
Department of Computer Science and Engineering, Neuroscience Program, and Cognitive Science Program
Michigan State University
East Lansing, MI, 48824 USA
Email: http://www.cse.msu.edu/~weng/

***Abstract*—While the brain develops through interactions with the external environment, it almost never experiences an exact physical event twice. Furthermore, the attended objects and events do not appear exactly the same as before and the background settings are not the exactly same either. However, the brain can abstract and generalize to deal with those environmental variations. This paper discusses a theory on the completeness of the logic capability of the Developmental Networks (DN) as a simplified brain-like model from such environmental variations. Various abstractions and generalizations are emergent properties of a DN through its incremental lifetime learning experience. While this process takes place, the network appears to have an increasing amount of logic capability in the eyes of human observers. Since it seems impossible to explain all possible kinds of logic capabilities that a human can acquire through his lifetime, I propose a general task-nonspecific formulation about logic capability in a DN. I prove that a DN incrementally generates and updates an internal, emergent, Finite Automaton (FA), whose complexity becomes increasingly high under the teacher's scaffolding scheme. It seems that such a highly complex FA can implement any practical logic. This is a theoretical paper but it discusses and cites supporting experimental results.**

I. Introduction

Logic capabilities require an emergent solution for two conflicting criteria — specificity and transfer.

After a child has learned that the object that it sees in his left hand is called "apple", the child should call the same object in his left hand "apple" when he is asked again. However, he should not call a pear in his left hand "apple". This is called specificity. *Specificity* means that the brain must maintain a discrimination power for specific examples it has learned under the same context setting. In our example, the child distinguish "apple" from a "pear" in the same context setting (in his left hand).

When a new apple is now placed in his right hand, the child should still call it "apple". This is called transfer a skill across different instances of the same class. In general, *transfer* means that skills acquired from one environmental setting are transferred to applicable but different environmental settings without a need for explicit learning for those new settings. In the above example, the change in the environmental setting is the location (from left to right hand). Transfer is a particular type of generalization. The environmental setting can change in many aspects, such as location, viewing angle, lighting, background, temporal context, and task.

Let us look at the example of Where-What Networks [6], which model the brain's dorsal pathways (where or how information) and the ventral pathways (what information). After training, the Type Motor area (TM) in the WWN is location invariant, meaning that the same object can be recognized correctly at any locations. However, this is not transfer yet, since the same object appearance has been learned at all locations. Suppose that a new apple is learned at a particular location. If the WWN can report "apple" for the new apple at different locations without a need for learning the new apple at all other locations, then the WWN does transfer for the new apple across these successfully tested locations. This is what WNNs have accomplished, in all their disjoint tests [6].

We propose that skill transfer is a key for the brain to have a general logic capability. Skill transfer is a notation extensively studied in psychology [2] but in machine learning this notion is still novel and challenging. Daniel Oblinger, the Director of the DAPPA Transfer Learning Program, wrote 2011 [9]: "Creating a formal theory of transfer remains a critical, yet difficult, direction for future work." As a part of network logic capabilities, I attempt a theory of transfer here and explain its power in dealing with logic.

Let us first discuss how *handcrafted* Symbolic Networks (SN) do logic, but we want emergent networks to do logic.

II. Symbolic Networks

Each element in the environment is attended sequentially by the agent to reach a new state, let us start with the framework of (deterministic) Finite Automaton (FA) as a special case of Symbolic Networks (SNs).

A. Finite automata

An FA example is shown in Fig. 1(a). At each time instance, the FA is at a state. At the beginning, our example is at state z_1. Each time, it receives a label as input (e.g., "young"). Depending on its current state and the next input, it transits to another state. For example, if it is at z_1 and receives label "young", it transits to "z_2", meaning "I got 'young'." All other inputs from z_1 leads back to z_1 meaning "start over". The states have the following meanings: z_1: start; z_2: "young"; z_3: "kitten" or equivalent; z_4: "kitten looks" or equivalent. An FA can abstract. For example, our FA example treats "young cat" and "kitten" the same in its state output.

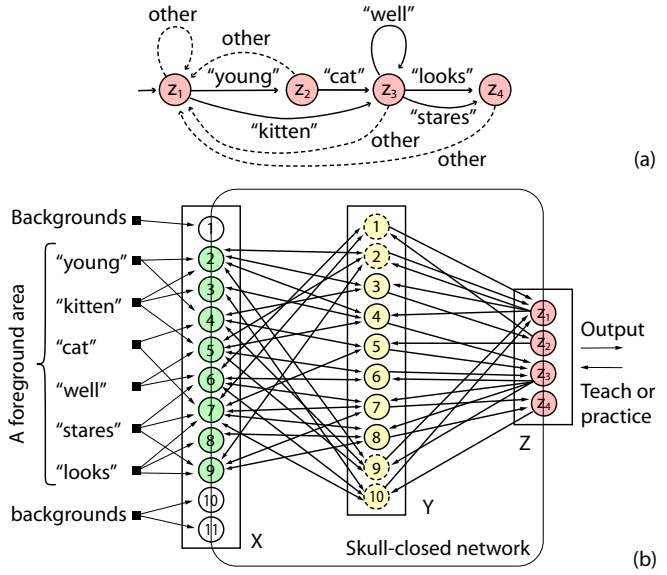

Fig. 1. Conceptual correspondence between an Finite Automaton (FA) with the corresponding DN. (a) An FA, handcrafted and static. (b) A corresponding DN that simulates the FA. It was taught to produce the same input-out relations as the FA in (a). A state symbol (e.g., z_2) in (a) corresponds to a Z image (e.g., $(z_1, z_2, z_3, z_4) = (0, 1, 0, 0)$) in (b). Each state symbol in (a) is abstract and so is each Z image in (b).

The FA framework has been often defined as a language acceptor in the traditional automata theory [3]. To model an agent, it is desirable to extend the definition of the FA as a language acceptor to an agent FA. An agent FA (AFA) M for a finite symbolic world is the same as a language acceptor FA, except that it outputs its current state, instead of an action (accept or not accept) associated with the state. In the following, an FA means an AFA by default.

The input space of an FA is denoted as $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_l\}$, where each σ_i representing an input symbol, whose meaning is only expressed in the design document (e.g., that of CYC [5]), not something that the FA is aware of. The set of states can be denoted as $Q = \{q_1, q_2, ..., q_n\}$.

It is important to note that the meanings of input symbols σ's and the meanings of states q's are in the design document of the FA, but the FA is not "aware" of such meanings. Fig. 1(a) gives a simple example of FA.

B. Completeness of FA

Let Σ^* denote the set of all possible strings of any finite $n \geq 0$ number of symbols from Σ. All possible input sequences that lead to the same state q are equivalent as far as the FA is concerned. It has been proved that an FA with n states partitions all the strings in Σ^* into n sets. Each set is called equivalence class, consisting of strings that are equivalent. Since these strings are equivalent, any string x in the same set can be used to denote the equivalent class, denoted as $[x]$. Let Λ denote an empty string. Consider the example in Fig. 1(a). The FA partitions all possible strings into 4 equivalent classes. All the strings in the equivalent class $[\Lambda]$ end in z_1. All strings in the equivalent class ["kitten" "looks"] end in z_4, etc.

The completeness of agent FA can be described as follows. When the number of states is sufficiently large, a properly designed FA can sufficiently characterize the cognition and behaviors of an agent living in the symbolic world of input set Σ. But Weng stated FA is insufficient for perception [14].

C. Other types of automata

There are four well-known types of automata: FA, Push-down Automata, Linear Bounded Automata, and Turing machines. Automata have been used to model the syntax of a language, but syntax does not give much information about semantics. As argued by linguisticists [11], [4], semantics is primary in language acquisition, understanding and production, while syntax is secondary.

The DN theory below enables the semantics to emerge implicitly in its connection weights in the network. In particular, it treats syntax as part of the emergent semantics, at all possible "levels". It does not separately treat syntax as in the Chomskys hierarchy of automata. Therefore, FA seems sufficient for a state-based symbolic agent in terms of logic.

III. Developmental Networks

The new class of DN, different from all traditional networks, is a model for the brain [14]. Its has demonstrated perception, attention, cognition, and language acquisition [6], [8], [14].

A. DN architecture

A basic DN has three areas, the sensory area X, the internal (brain) area Y and the motor area Z. An example of DN is shown in Fig. 1(b). The internal neurons in Y have two-way connection with both X and Z. In principle, the X area can model any sensory modality (e.g., vision, audition, and touch). The motor area Z serves both input and output. When the environment supervises Z, Z is the input to the network. Otherwise, Z gives an output vector to drive effectors (muscles) which act on the real world. The order of areas from low to high is: X, Y, Z. As shown in Fig. 1, X and Y provide bottom-up input $\mathbf{b}$ to Y and Z, respectively; while Z and Y give top-down input $\mathbf{t}$ to Y and X, respectively.

B. DN algorithm

Algorithm 1 (DN): Input areas: X and Z. Output areas: X and Z. The dimension and representation of X and Y areas are hand designed based on the sensors and effectors of the species (or from evolution in biology). Y is the skull-closed (inside the brain), not directly accessible by the outside.

1) At time $t = 0$, for each area A in $\{X, Y, Z\}$, initialize its adaptive part $N = (V, G)$ and the response vector $\mathbf{r}$, where V contains all the synaptic weight vectors and G stores all the neuronal ages. For example, use the generative DN method discussed below.
2) At time $t = 1, 2, ...$, for each A in $\{X, Y, Z\}$ repeat:
   a) Every area A performs neurogenesis if it is needed, using its bottom-up and top-down inputs $\mathbf{b}$ and $\mathbf{t}$, respectively.

b) Every area A computes its area function f, described below,

$$(\mathbf{r}', N') = f(\mathbf{b}, \mathbf{t}, N)$$

where $\mathbf{r}'$ is its response vector.

c) For every area A in $\{X, Y, Z\}$, A replaces: $N \leftarrow N'$ and $\mathbf{r} \leftarrow \mathbf{r}'$.

This is a Generative DN in the sense that new neurons are generated autonomously. In the following by DN, we mean a GDN by default.

C. Unified DN area function

It is desirable that each brain area uses the same area function f, which can develop area specific representation and generate area specific responses. Each area A has a weight vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$. Its pre-response value is:

$$r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \dot{\mathbf{v}} \cdot \dot{\mathbf{p}} \tag{1}$$

where $\dot{\mathbf{v}}$ is the unit vector of the normalized synaptic vector $\mathbf{v} = (\dot{\mathbf{v}}_b, \dot{\mathbf{v}}_t)$, and $\dot{\mathbf{p}}$ is the unit vector of the normalized input vector $\mathbf{p} = (\dot{\mathbf{b}}, \dot{\mathbf{t}})$. The inner product measures the degree of match between these two directions $\dot{\mathbf{v}}$ and $\dot{\mathbf{p}}$, because $r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \cos(\theta)$ where θ is the angle between two unit vectors $\dot{\mathbf{v}}$ and $\dot{\mathbf{p}}$. This enables a match between two vectors of different magnitudes (e.g., a weight vector from an object viewed indoor to match the same object when it is viewed outdoor). The pre-response value ranges in $[-1, 1]$.

To simulate lateral inhibitions (winner-take-all) within each area A, top k winners fire. Considering $k = 1$, the winner neuron j is identified by:

$$j = \arg \max_{1 \le i \le c} r(\mathbf{v}_{bi}, \mathbf{b}, \mathbf{v}_{ti}, \mathbf{t}). \tag{2}$$

The area dynamically scale top-k winners so that the top-k respond with response values r' in $(0, 1]$. For $k = 1$, only the single winner fires with response value $r'_j = 1$ (a pike) and all other neurons in A do not fire. The response value r'_j approximates the probability for $\dot{\mathbf{p}}$ to fall into the Voronoi region of its $\dot{\mathbf{v}}_j$ where the "nearness" is $r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t})$.

D. DN learning: Hebbian

All the connections in a DN are learned incrementally based on Hebbian learning — cofiring of the pre-synaptic activity $\dot{\mathbf{p}}$ and the post-synaptic activity y of the firing neuron. When a neuron j fires, its firing age is incremented $n_j \leftarrow n_j + 1$ and then its synapse vector is updated by a Hebbian-like mechanism:

$$\mathbf{v}_j \leftarrow w_1(n_j)\mathbf{v}_j + w_2(n_j) r'_j \dot{\mathbf{p}} \tag{3}$$

where $w_2(n_j)$ is the learning rate depending on the firing age (counts) n_j of the neuron j and $w_1(n_j)$ is the retention rate with $w_1(n_j) + w_2(n_j) \equiv 1$. The simplest version of $w_2(n_j)$ is $w_2(n_j) = 1/n_j$ which corresponds to:

$$\mathbf{v}_j^{(i)} = \frac{i-1}{i} \mathbf{v}_j^{(i-1)} + \frac{1}{i} 1 \dot{\mathbf{p}}(t_i), i = 1, 2, ..., n_j, \tag{4}$$

where t_i is the firing time of the post-synaptic neuron j.

IV. DN Can Perfectly Simulate Any Given FA

It is a long, extensively debated question (e.g., see Minsky 1991 [7]) whether a neural network can abstract as well as a symbolic network. Weng 2011 [16] provided three theorems, which provide properties about how well a DN can abstract, using FA as a basis. The proofs for the three theorems are available as a report [15]. Since this paper is meant to discuss the general logic capability of DN, let us have an informal explanation of the three theorems. For more detail about their importance, the reader is referred to [17].

Theorem 1: The developmental program (DP) of DN can incrementally grow a Generative DN (GDN) to simulate any given FA on the fly, so that the performance of the DP is immediate and error-free, provided that the Z area of the DN is supervised when the DN observes each new state transition from the FA. The learning for each state transition completes within two network updates. There is no need for a second supervision for the same state transition to reach error-free future performance. The number of Y neurons in the DN is the number of state transitions in the FA. However, the DN generalizes with 0% action error for infinitely many equivalent input sequences that it has not observed from the FA but are intended by the human FA designer.

As a sketch of the proof, Fig. 2 illustrates how the DN simulates each new state transition of FA by creating a new Y neuron that immediately initializes with the image code of the state $q(t-1)$ and the image code of the input $\sigma(t-1)$ through the first network update (see the Y area at time $t-0.5$). During the next network update, the Z area is supervised as the image code of the desired state $q(t)$ and the links from the uniquely firing new Y neuron to the firing Z neurons are created through a Hebbian mechanism. Since the match of the new Y neuron is exact and only one Y neuron fires at any time, the Z output is always error-free if all image codes for Z are known to be binary (spikes).

The other two theorems in Weng 2011 [16] establish that when the learned DN takes infinitely many inputs, DN is optimal in the sense of maximum likelihood, whether the DN is frozen (the 2nd theorem) or allowed to continue to adapt (the 3rd theorem).

In general, an FA can be designed for any symbolic function, such as functions of the propositional logic, first-order logic, and higher order logic. For example., consider a two variable logic function $f(A, B) = A \wedge B$, where A and B are logic variables and $\wedge$ denotes logic AND. Table I is the state transition table implementing this function, starting with state q_0. In the table, at the row q and column σ, the entry is the state q' in $q \xrightarrow{\sigma} q'$. Thus, FA can model any logic functions.

The next problem to solve is, typically, the FA is not known or given *a priori*. Instead, the DN must automatically construct an FA on the fly (i.e., development), through interactions with the environment, which typically includes a human teacher.

We then turn the question to a teacher who is part of the environment which the GDN interacts with. What are the conditions for the teacher to successfully teach a GDN for a task?

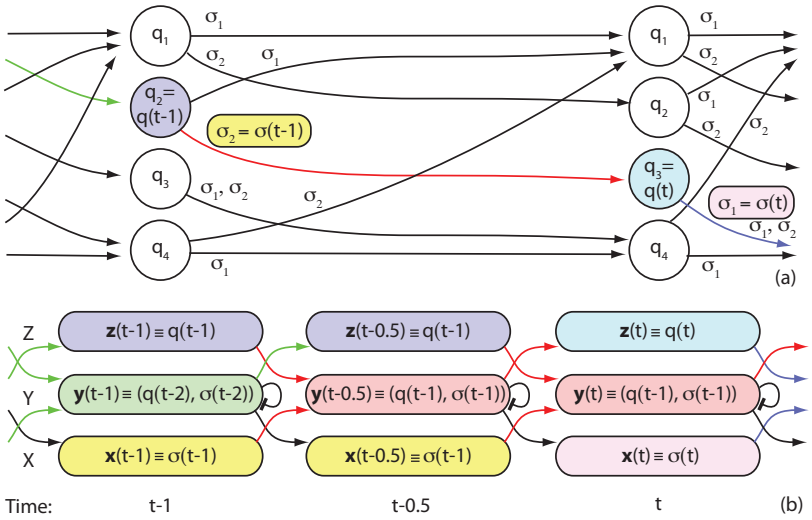

Fig. 2. Model the brain mapping, DN, and SN. In general, the brain performs external mapping $b(t) : X(t-1) \times Z(t-1) \mapsto X(t) \times Z(t)$ on the fly. (a) An NS samples the vector space Z using symbolic set Q and X using Σ, to compute symbolic mapping $Q(t-1) \times \Sigma(t-1) \mapsto Q(t)$. This example has four states $Q = \{q_1, q_2, q_3, q_4\}$, with two input symbols $\Sigma = \{\sigma_1, \sigma_2\}$. Two conditions (q, σ) (e.g., $q = q_2$ and $\sigma = \sigma_2$) identify the active outgoing arrow (e.g., red). $q_3 = \delta(q_2, \sigma_2)$ is the target state pointed to by the (red) arrow. (b) The grounded DN generates the internal brain area Y as a bridge, its two-way connections with its two banks X and Z, the inner-product distance, and adaptation, to realize the external brain mapping. It performs at least two network updates during each unit time. To show how the DN learns a SN, the colors between (a) and (b) match. The sign $\equiv$ means "image code for". In (b), the two red paths from $q(t-1)$ and $\sigma(t-1)$ show the condition $(\mathbf{z}(t-1), \mathbf{x}(t-1)) \equiv (q(t-1), \sigma(t-1))$. At $t-0.5$, they link to $\mathbf{y}(t-0.5)$ as internal representation, corresponding to the identification of the outgoing arrow (red) in (a) but a DN does not have any internal representation. At time t, $\mathbf{z}(t) \equiv q(t) = \delta(q(t-1), \sigma(t-1))$ predicts the action. But the DN uses internal $\mathbf{y}(t-0.5)$ to predict both state $\mathbf{z}(t)$ and input $\mathbf{x}(t)$. The same color between two neighboring horizontal boxes in (b) shows the retention of (q, σ) image in (a) within each unit time, but the retention should be replaced by temporal sampling in general. The black arrows in (b) are for predicting X. Each arrow link in (b) represents many connections. When it is shown by a non-black color, the color indicates the corresponding transition in (a). Each arrow link represents excitatory connections. Each bar link is inhibitory, representing top-k competition among Y neurons.

TABLE I
FA THAT IMPLEMENTS LOGIC AND

$q \setminus \sigma$	T	F	$\wedge$
q_0	q_T	q_F	-
q_T	-	-	$q_{T\wedge}$
q_F	-	-	$q_{F\wedge}$
$q_{T\wedge}$	q_T	q_F	-
$q_{F\wedge}$	q_F	q_F	-

V. IN THE HUMAN EYES

In order to define a task condition, the teacher specifies a condition set C applied to an agent-centerd external environment $E(t)$ at time t. If $C(E(t)) =$ True, meaning that the condition set C checks the spatiotemporal context of environment $E(t)$ up to time t and his conclusion is true. We say "spatiotemporal context" instead of directly $E(t)$ at time t because certain conditions involve temporal context (e.g., the teacher says to the agent: "start!" which lasts a second).

Suppose that a teacher (or caregiver) needs to teach a DN to perform a task whose goal is to change the external environment $E(t)$ from $E(t_0)$ to $E(t_n)$ through n time steps so that an initial condition set C_0 on $E(t_0)$ and an target condition set C_n on $E(t_n)$ are both satisfied: $C_0(E(t_0)) =$ True and $C_n(E(t_n)) =$ True, in the eyes of the teacher.

For example, for a transportation problem, C_0 checks that the agent and the cargo are at the starting location L_0. C_n checks that the agent and the cargo are at the target location L_n. If the agent moves from L_0 to L_n without taking the cargo, the condition set C_n is not true. If somebody else takes the cargo from L_0 to L_n but not the agent, C_0 can be false or C_n can be false. Such condition verifications are done typically by humans, although it is desirable for the agent also to check such conditions so that it can know whether it is doing correctly.

We can also consider a series of tasks. Each task i is a condition pair (C_{i-1}, C_i), where C_{i-1} is the task starting condition set and C_i is the task ending condition set. It is important to note that C_{i-1} should include also task specification, so that the agent can react to perform the task i. The task specification can be implicit (e.g., the teacher extends her arm forward means "shake hands with me"), and explicit (e.g., simply state "shake hands with me").

Suppose that an agent DN successfully performs a task (C_0, C_n) from environment $E(t_0)$ and reaches an target environment $E(t_n)$ so that $C_0(E(t_0)) = C_n(E(t_n)) =$ True. Then, the DN can be modeled by an emergent FA which learns and performs state transitions through discrete times but in a task nonspecific way:

Algorithm 2 (DN grounded learning and execution): For $i = 1, 2, ..., n$ do

1) Take sensory input $X(t_{i-1})$ from environment $E(t_{i-1})$.
2) In the first DN update, Y does motor attention from $Z(t_{i-1})$ to get the attended state q_{i-1}, and sensory

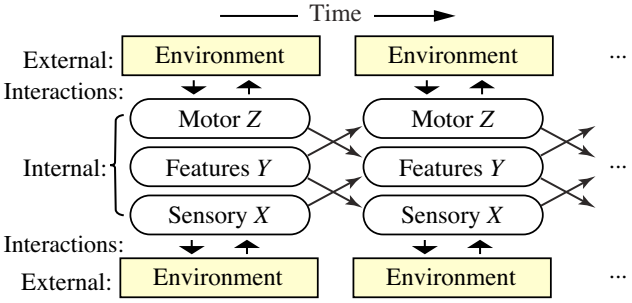

Fig. 3. Grounded task learning and execution by the DN. The internal attention by Y determines which part of the sensory X and motor Z are attended. The teacher, as part of the external environment, interacts (teaches) the DN agent.

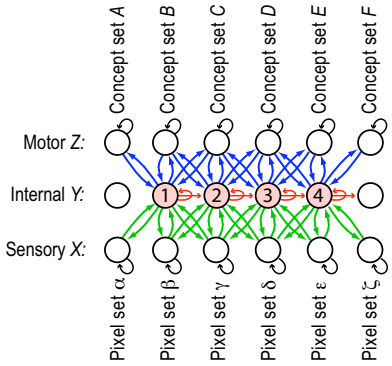

Fig. 4. How a DN simulates an autonomously generated NFA.

attention from $X(t_{i-1})$ to get the attended input σ_{i-1}:

$$(q_{i-1}, \sigma_{i-1}) \longrightarrow Y$$

3) In the 2nd DN update, Z and X perform prediction

$$Y \longrightarrow (\sigma_i, q_i).$$

where q_i predicts $Z(t_i)$ which interacts with (can be overridden by) the environment to give $E(t_i)$, and σ_i predicts $X(t_i)$ which interacts with (can be overridden by) the environment in step 1).

In the above algorithm, the teacher must teach DN to generate motor $Z(t_i)$ in each step i. This is too tedious if the teach immediately teach a baby to read a novel.

In practice, the teacher designs a teaching schedule, from simple tasks to complex ones. The simple tasks serve as scaffolding for performing more complex tasks, as described originally by Vygotsky [11] now well known in developmental psychology [18]. For example, early tasks are eating, crawling and playing. Later tasks are reading, practice and planning.

A teacher designs a teaching schedule first, based on what an age group can typically do. Suppose $S = (C_0, C_1, ...C_n)$ is a teaching schedule for an age group where (C_{i-1}, C_i) is the task i, $i = 1, 2, ..., n$. The skills acquired by the age group enable the agent to learn each task i in a semi supervised way, so that the teacher does not need to supervise the agent for every time frame during the teaching.

Variations of environments are important. So, we need the mechanism of transfer: State-equivalence in the agent perception enables it to map two different environment contexts E and E' to the same state: q, so that the FA in the brain applies the skill $q \xrightarrow{\sigma} q'$ learned in E to new context E' without learning E' explicitly. If the agent successfully does the task through a variety of environments (starting, intermediate, and target environments), the teacher gives a "pass".

The following theorem gives a sufficient condition for a DN to successfully carry out a task in the eyes of a teacher.

Theorem 2 (Task success): Suppose that DN has acquired a concept set C, which consists of symbolic symbols of all the actions that DN can generate. If a teacher trains the DN additionally by expanding teaching schedule S so that at each step i, the attended DN state q_i contains concepts $q_i = \{c_{i,1}, c_{i,2}, ..., c_{i,n_i}\}$, where $c_{i,j} \in C$, $1 \le j \le n_i$, so that

1) $C_0(E(t_0)) =$ True,
2) DN behaves as an emergent FA for the given task, to perform state transition $q_{i-1} \xrightarrow{\sigma_i} q_i, i = 1, 2, ..., n$.
3) $C_n(E(t_n)) =$ True.

Then the DN is an emergent FA that has successfully executed the task in the eyes of the teacher.

Proof: The proof follows from the proof of Theorem 1 in [15]. It has been proved that a GDN can simulate any given FA sequentially, perfectly, and error free, based on the observed state transition $q_{i-1} \xrightarrow{\sigma_i} q_i$. The differences here are: (1) We denote the DN as a non-symbolic (emergent) FA. (2) The sensory attention from X is incorporated to give σ_i. (3) The motor attention from Z is incorporated to give q_i. (4) Represent the symbolic state q_i by a subset of concept set C, such that each state q_i emerges from Z through real time instead of handcrafted upfront by a programmer. (5) The actions of DN are vectors checked by the teacher using her intuition about task conditions C_0 and C_n, so the term "in the eyes of the teacher". For example, the teacher thinks that a baby's action "eat" does not have to be perfect. ■

It is important to note that the DN is not a symbolic FA, since handcrafting such an FA for a non-trivial task is typically intractable for a large task (e.g., the symbolic CYC [5]). If $C = \{\text{eat}, \text{craw}, \text{play}\}$, then $q_1 = \{\text{eat}, \text{craw}\} \subset C$. However, the number of all the actually emerged states from DN is very small compared with all the possible subsets of C, 2^C.

The importance of this theorem includes: (1) A subset $q \in C$, as a new concept (can be indicated by a new word), from the set of learned concepts C could emerge from the DN — an important strength of scaffolding. (2) This theorem is task non-specific, applicable to any task at lest in principle. Because of the task non-specificity, it has established as a special case that DN is logically complete, not only in terms of FA, but also in the human eyes.

VI. Bi-Directional Excitations

To examine more closely how DN enables states to emerge, let us consider a simple case, without loss of generality, as illustrated in Fig. 4. Greek letters α, β, γ, etc. denote the firing

neurons in X, while the letters, A, B, C, etc. denote the firing neurons in Z. During the first time block T_1, $t \in T_1$, the set of firing pixels are $X(t) = \alpha \cup \beta \cup \gamma$ the union of three sets of firing pixels α, β and γ, and the set of fitting motor neurons are $Z(t) = A \cup B \cup C$ the union of three sets of firing motor neurons A, B, and C, and the Y neuron 1 fires. This situation does not significantly change during the time block T_1. That is why each of the neuronal sets A through C, α through γ, and Y neuron 1 has an arrow loop that points back to the same set as time sequence. The Y neuron 1 is the current best-matched neuron among all the neurons in the Y area for both bottom-up and top-down inputs. Because the Y neurons co-fires with the corresponding X and Z neurons in time block T_1, their connections with the Y neuron 1 are all bi-directional.

Consider the next time block T_2, $t \in T_2$, the firing neurons in X and Z are $\beta \cup \gamma \cup \delta$ and $B \cup C \cup D$, respectively. Suppose that the Y neuron 2 becomes the best-matched neuron. Similar to time block T_1, the connections with the Y neuron 2 are also bi-directional. This process continues on. In general, because of the limited neuronal resource in Y, a slight change in either X and Z does not necessarily change the firing Y neuron, only a considerable change does.

The two-way excitatory connections result in the prediction from partial stimuli. This is a major difference from a symbolic FA which is rigid. For example, in Fig. 4, suppose that only the neurons in the set α is firing. The learned bi-directionally connections in Fig. 4 quickly cause X to have $\alpha \cup \beta \cup \gamma$ firing, and Z to have $A \cup B \cup C$ firing, because the Y neuron 1 fires as the top-1 match.

In the following, we use the theorem 2 and the general connection pattern in Fig. 4 to discuss a series of tasks.

VII. Realizing Psychological Learning Models

An FA is exact. DN can go beyond an FA, to learn and act like a brain. So, we discuss psychological learning models.

There are many types of associative learning formulated by psychology, including classical conditioning, instrumental conditioning and the very large class of more sophisticated learning, called cognitive learning [2] which includes transfer, shown in Fig. 5, as the basic mechanism to scaffold skill complexity.

We use the original word (e.g., Tone, Food) for the sensory stimulus sensed by X and the quoted word (e.g., "Tone", "Food") to represent the concepts in Z. For simplicity, we assume that each input symbol is exclusive, in the sense that the agent can take one input symbol at a time, not conjunctively.

A. Classical conditioning

Let us first consider a task, classical conditioning. For this task, the concepts learned by the Z neurons are "Tone", "Food" and "Salivation", 3 concepts. The number of all possible actions from 3 concepts is $2^3 = 8$, but not all will appear. We use Λ to represent that none of the concept is present. According to the Theorem 2, the state $q \in 2^C$ emerges from Z.

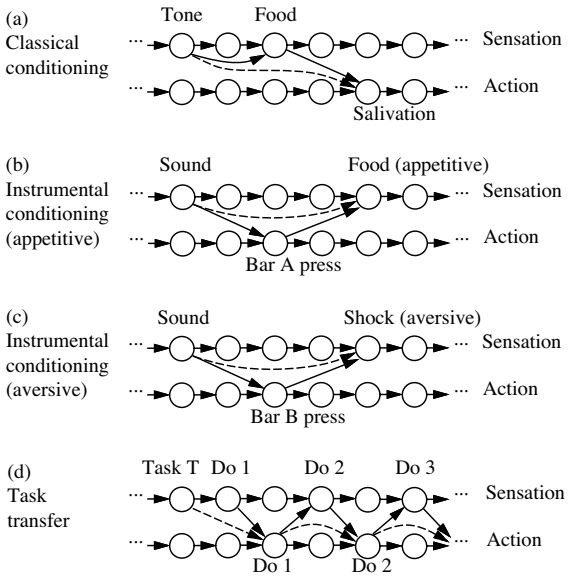

Fig. 5. A few learning models in psychology: (a) classical conditioning, (b-c) instrumental conditioning, and (d) task transfer. Each circle in the figure indicates a time instance. From left to right is the passing events cross time. Solid arrows denote short-time-separation associations. Dashed arrows denote prediction through learned experience.

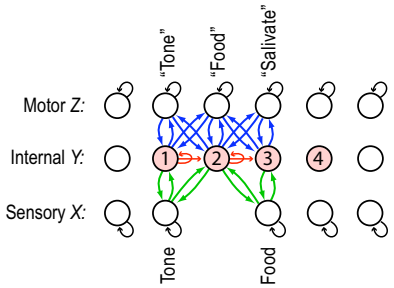

Fig. 6. A DN from an environment as classical conditioning.

Suppose that our agent is a dog. The owner of the dog always make a tone before feeding the dog with its favorite food. The DN is a model for this simple dog brain. After experiencing this environmental events repeatedly, the developed DN is illustrated in Fig. 6 which corresponds to the FA in Table II. In each table entry, e.g., $\{1\}$, {T, F}, the first item (e.g., $\{1\}$) is the set of Y neurons that fire under the corresponding row-column condition; and the second item (e.g., {T, F}) is the next Z state. "-" means never happened. The state of the DN is determined by all the firing neurons in Z, not by a single Z neuron.

Modeled as a task, classical conditioning, repeated sequences of events are learned: a tone is conditionally followed by food which is unconditionally followed by salivation. "Conditionally" means if the environment satisfies the condition. "Unconditionally" means that the agent will develop (early in life) such a behavior under typical, normal environmental conditions. (This does not mean that the behavior is innate without a need for environment.) After the dog is trained in such an environment, the dog will salivate after it hear the

TABLE II
DN FOR CLASSICAL CONDITIONING REPRESENTED AS AN FA: T: "TONE"; F:"FOOD"; S: "SALIVATE"; OTHER: OTHER Y NEURONS.

Motor Z \ Sensory X	Λ	Tone	Food
State A	other, A	{1}, {T}	-
State B: {T}	{1}, {T}	{1}, {T, F}	{2}, {T, F, S}
State C: {T, F}	{1}, {T, F}	{2}, {T, F, S}	{2}, {T, F, S}
State D: {T, F, S}	{2}, {T, F, S}	{2}, {T, F, S}	{2}, {T, F, S}

Tone, even without the presentation of Food. How does the DN accomplish its learning for classical conditioning?

As the tone will last for multiple time frames, the firing Y neuron 1 has two-way excitatory connections with the Z neuron "Tone". When the stimulus Food is presented next, the Z neuron for generating "Food" fires through motor-supervised learning. Just like the first stimulus Tone, Y neuron 2 fires.

In order to enable DN to make predictions, we require $k > 1$ in top-k competition of the Y area. Furthermore, there are sufficient number of Y neurons so that a single Y neuron in Fig. 6 denotes more than one neuron.

B. Training in classical conditioning

When the Food stimulus follows the Tone stimulus, the Z neuron "Tone" does not stop firing immediately, at last lasting for a few time frames. This is because of the two-way connections between the Z neuron "Tone" and the Y neuron 1: Although the X neuron Tone stopped firing, the Y neuron 1 still gives excitation from the Z neuron "Tone" and will be a top-k winner for a short while.

When the Y neuron 2 fires as the top winner, the Z neuron "Food" is supervised to fire. This sustained firing enables the Y neuron 2 to detect the co-occurring of "Food" bottom-up input, the "Tone" top-down input, and the "Food" top-down input. While Z neuron "Food" fires, the "Salivate" also fires (via motor supervised learning to simulate unconditioned response). Food is the stimulus for the unconditional response "Salivate". Eventually, the Y neuron 2 represents the co-firing of Tone, Food, "Tone", "Food", and "Salivate".

When the memory of "Tone" abates, the Y neuron 2 does not match well. The Y neuron 3 takes over as the firing Y neuron, representing the co-firing of Food stimulus, "Food" action and "Salivate" action.

In general, mutual excitatory connections enable the corresponding Y and Z neurons to prolong their firing. Such a sustained firing enables a series of Y and Z neurons to form a chain, connected by bidirectional excitatory synaptic links.

C. Testing in classical conditioning

Next, suppose that the Tone stimulus is presented without the follow-up Food stimulus. The network responds through the established connections in Fig. 6: While the Tone stimulus is present, the Y neuron 1 fires which causes the Z neuron "Tone" to fire. The firing of Y neuron 1 causes the Z neuron "Food" also fires. Thus, although the Food stimulus is not present, the DN predicts the Food stimulus to follow. In other words, the DN "thinks" about "Food". Furthermore, the firing Z neuron "Food" predicts the firing of Z neuron "Salivate" via the Y neuron 2.

In general, an absent sensory stimulus (e.g., Food) is bridged by the corresponding Z neuron (e.g., "Food") through $Y - Z$ excitatory loops. Similarly, the X-Y loops allow Y neurons to predict sensory X, as a predicted mental image. Such predictions for Z and X through Y realize all the dashed links in Fig. 5. This is a form of "brain" internal prediction, which is different from the real sensory experience during training because of the absence of the real stimuli sensed by X.

D. Instrumental conditioning

Instrumental conditioning is another psychological learning model, illustrated in Fig. 5, that involves reinforcers.

The Q-learning algorithm [13] is a well-known machine-learning algorithm that uses a symbolic network. The Q-learning algorithm deals with the time-delayed problem using a time-discount mechanism. A Q-value $Q(s, a)$ is estimated for each state-action pair (s, a), where s and a are symbolic state and symbolic action, respectively.

The Q-learning algorithm uses an expression that updates the Q value of the current state-action pair (s, a) using all the experienced next state-action pair (s', a') using a time-discount parameter $0 < \gamma < 1$ to discount the Q value at (s', a') and the next reward. The Q-learning mechanism has an advantage of simplicity, but has three major disadvantages: (1) The time discount parameter γ causes the system to favor immediate rewards to more remote future rewards which is not always desirable (e.g., working hard every day for a long-term career goal). (2) The representation of the Q-learning network is symbolic, where each state s is a symbol with pre-defined extra-body meanings. Thus, it is unable to learn concepts beyond those handcrafted. (3) Its number of potential states is exponential in the number of concepts that are needed to define a state.

From Theorem 2, a DN agent needs to learn concepts, sound, bar A press, bar B press, food, and shock. Instrumental conditioning is a special case of planning based on the predicted value of alternative actions. It is more convenient for us to discuss this subject in Section VIII.

E. Language acquisition

Semantics is the study of how linguistic elements carry meaning. Acquisition of semantics [12] includes word meaning, argument structure (the linguistic structure that a word, such as a verb, projects into the syntax by virtue of its

meaning), tense and aspects (e.g., how an event happened and how it unfolds through time), quantification and scope (e.g., every, some).

The primary challenge for acquiring word meanings is the problem of reference — how word forms are linked to specific concepts of the environment. Attention plays a major role in the establishment of such a link.

As discussed in [8], suppose a DN need to learn the following sentence: I gave you a key yesterday. We can draw a diagram where a node denotes on word, a label indicates the related properties, and a link from a word to another word represents a relationship. For example "I" have the property "subject, singular, the first person", etc. Then each state in DN can learn the firing of multiple Z neurons, where each Z neuron represents an acquired concept. The relationship is also a concept, represented as a neuron in Z as explained in [8], except that a relationship needs the co-firing of the corresponding multiple concepts (e.g., I gave you, where "gave" is a relationship with "I" and "you" the associated object concepts). From the knowledge diagrams developed autonomously by DN, as discussed in [8], we can see that DN is of general purpose in terms of language acquisition.

VIII. Autonomous Planning

Among many more complex tasks enabled by DN, we address *autonomous planning* as an example.

Theorem 3: A DN allows internal reasoning to realize autonomous planning.

Proof: Autonomous planning requires first an accumulation of experiences so that alternative condition-action pairs are learned. Let us use a notation: the pair (l, p) denotes a pair of the last context $l = (q, \sigma)$ and the predicted context $p = (q', \sigma')$ learned by the DN.

Suppose that there are two plans according to the experiences: The execution path of the plan (a) is recalled as:

$$(l_1, p_{a,1}), (l_{a,2}, p_{a,2}), ..., (l_{a,i}, p_{a,i}))$$

and that of the plan (b) is recalled as:

$$(l_1, p_{b,1}), (l_{b,2}, p_{b,2}), ..., (l_{b,j}, p_{b,j})).$$

Both lead to a completion of the task. Both plans are recalled sequentially using only the internal part of DN in Fig. 3. Finally the value of $p_{a,i}$ is compared with that of $p_{b,j}$. The modulatory system is the value system of DN [10], [1], which decides which value is better and so chooses the corresponding plan (a) or (b). The association of a to the predicted action in $p_{a,1}$ and b with that in $p_{b,1}$ is DN learned prediction. At the end of the plan (a), the selected plan in $p_{a,i}$, as part of the last context in l, predicts the first action in $p_{a,1}$. The similar process takes place for plan (b). ■

Zhang & Weng [19], [20] has successfully enabled the agent to recall the context-action sequence as discussed in the proof, through a process called scaffolding: Artificially arranged setting to enable quick learning via real-time online human-agent interactions. However, the theoretical result here for autonomous planning has yet to be demonstrated in the future studies. I expect that non-trivial demonstration of autonomous planning is possible only within a relatively mature DN.

IX. Conclusions

A DN network has a potential to learn and perform any practical logic as observed in the eyes of human observer, but it requires the facilitation of a well designed teaching sequence for scaffolding. Internally, the developmental mechanisms of DN are not based on any logic. Any such a logic is limited.

We have passed "neural networks are scruffy" in that they are not logical, as Minsky put [7]. A DN is meant to scale up, as its development is autonomous. Guided of the presented theory, the future work is to expand the demonstrated DN experimental results gradually to human level performance.

References

[1] J. Daly, J. Brown, and J. Weng. Neuromorphic motivated systems. In *Proc. Int'l Joint Conference on Neural Networks*, pages 2917–2914, San Jose, CA, July 31 - August 5 2011.

[2] M. Domjan. *The Principles of Learning and Behavior*. Brooks/Cole, Belmont, California, fourth edition, 1998.

[3] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Boston, MA, 2006.

[4] J. M. Iverson. Developing language in a developing body: the relationship between motor development and language development. *Journal of child language*, 37(2):229–261, 2010.

[5] D. B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[6] M. Luciw and J. Weng. Where What Network 3: Developmental top-down attention with multiple meaningful foregrounds. In *Proc. IEEE Int'l Joint Conference on Neural Networks*, pages 4233–4240, Barcelona, Spain, July 18-23 2010.

[7] M. Minsky. Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI Magazine*, 12(2):34–51, 1991.

[8] K. Miyan and J. Weng. WWN-Text: Cortex-like language acquisition with What and Where. In *Proc. IEEE 9th Int'l Conference on Development and Learning*, pages 280–285, Ann Arbor, August 18-21 2010.

[9] D. Oblinger. Toward a computational model of transfer. *AI Magazine*, 32(2):126–128, 2011.

[10] S. Paslaski, C. VanDam, and J. Weng. Modeling dopamine and serotonin systems in a visual recognition network. In *Proc. Int'l Joint Conference on Neural Networks*, pages 3016–3023, San Jose, CA, July 31 - August 5 2011.

[11] L. S. Vygotsky. *Thought and language*. MIT Press, Cambridge, Massachussetts, 1962. trans. E. Hanfmann & G. Vakar.

[12] L. Wagner. Acquisition of semantics. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(4):519–526, 2010.

[13] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[14] J. Weng. A 5-chunk developmental brain-mind network model for multiple events in complex backgrounds. In *Proc. Int'l Joint Conf. Neural Networks*, pages 1–8, Barcelona, Spain, July 18-23 2010.

[15] J. Weng. Three theorems about developmental networks and the proofs. Technical Report MSU-CSE-11-9, Department of Computer Science, Michigan State University, East Lansing, Michigan, May,12 2011.

[16] J. Weng. Three theorems: Brain-like networks logically reason and optimally generalize. In *Proc. Int'l Joint Conference on Neural Networks*, pages 2983–2990, San Jose, CA, July 31 - August 5 2011.

[17] J. Weng. Why have we passed "neural networks do not abstract well"? *Natural Intelligence: the INNS Magazine*, 1(1):13–22, 2011.

[18] D. J. Wood, J. S. Bruner, and G. Ross. The role of tutoring in problem-solving. *Journal of Child Psychology and Psychiatry*, pages 89–100, 1976.

[19] Y. Zhang and J. Weng. Action chaining by a developmental robot with a value system. In *Proc. IEEE 2nd Int'l Conf. on Development and Learning (ICDL 2002)*, pages 53–60, MIT, Cambridge, Massachusetts, June 12-15 2002.

[20] Y. Zhang and J. Weng. Task transfer by a developmental robot. *IEEE Transactions on Evolutionary Computation*, 11(2):226–248, 2007.