2021 IEEE International Conference on Development and Learning (ICDL) | 978-1-7281-6242-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDL49984.2021.9515627

Optimal Developmental Learning for Multisensory and Multi-Teaching Modalities

J. Arden Knoll*†, Jacob Honer*‡, Samuel Church*‡, Juyang Weng*†§

*GENISAMA LLC, Okemos, MI 48864 USA
†Department of Computer Science and Engineering, ‡Department of Electrical and Computer Engineering,
§Cognitive Science Program and Neuroscience Program, Michigan State University, East Lansing, MI, 48824 USA

***Abstract*—Most intelligent systems require the use of multiple sensors and motors, but training these systems becomes costly as the number and complexity of sensors and motors increase. Reinforcement learning promises to make the training process easier on human trainers but can take significantly more time. While many learning systems are either directly supervised or supervised through reinforcement, few can learn through multiple teaching modalities. Furthermore, many neural network systems do not allow for frame-incremental learning. This work introduces a real-time, frame-incremental framework that uses multiple sensors and multiple teaching modes (motor-supervision, reinforcement, or self-practice). The Developmental Network (DN) used in this work is optimal in the sense of maximum likelihood throughout a "lifetime", under three conditions: (1) an incremental learning framework, (2) a training experience and (3) a limited amount of computational resources. Because the DN is free from the local minima problem, all DNs are performance-equivalent and we record "lifetime" errors of a single trained network, removing a need for post-selection—post-selecting the luckiest network from many randomly initialized and trained neural networks according to their performances on test sets.**

I. Introduction

The two major novelties of this work involve the multisensory and multi-teaching capabilities of our system. Our system is able to integrate information from multiple sensors (two cameras) by learning from multiple teaching modes (motor-supervision, reinforcement, and practice).

A. Multisensory Systems

The general consensus in biology is that coarse multisensory circuits are available early in life and are refined later on [1]. In other words, both single-modality and multi-modality skills are present early in life.

Consistent with the biological data, our model does not assume that single modalities are developed first and integrated later in life to give rise to multisensory skills. Instead, our model provides coarse connections to multi-modal areas early in life, which are then refined concurrently over time through neuronal competitions. Monocular neurons eventually result from the refinement of coarse binocular neurons in early life, similar to what was reported in [2] for unilateral-eye-closure kittens. In the case of two cameras, motor neurons that require monocular (single sensor) information learn to interpret the same neuronal input pattern differently from those motor neurons that require binocular (multiple sensors) information.

The early integration of multisensory information is not only biologically inspired, but seems to be consistent with maximal likelihood optimality, where the system must minimize the average error over its "lifetime". The first novelty of this work in multisensory learning is the theory and method for integrating monocular information with binocular information. However, the learning process is tedious and takes large amounts of data. Thus, we developed a system that can also learn from multiple teaching modalities.

B. Multi-teaching Systems: Eight Types

Traditionally, the frameworks for supervised learning and reinforcement learning were treated as different systems. Biologically, however, these two learning types should be considered two special cases of a multi-teaching system.

Weng 2012 [3] proposed a biologically inspired framework which unifies supervised learning, reinforcement learning, and a new kind of learning called communicative learning. This general framework considers three attributes: representation type (symbolic or emergent), effectors (teacher imposed or learner self-generated), and biased sensors (reinforcement or no reinforcement). These three attributes are combined to give rise to a total of eight learning types.

A 3-tuple (s, e, b) is used to classify training types. For example, $s = 1$ means symbolic internal representation, $e = 1$ means effector-imposed, and $b = 1$ means biased sensors (pain and sweet) are activated. Therefore, the binary code $seb = 000, 001, ..., 111$ results in eight types of teaching. A system is not fixed in supervised learning or reinforcement learning, but can conduct any type of learning at any time.

Many methods in traditional AI, fussy systems, evolutional computation, and neural networks (e.g., Neocognitron [4], HMAX [5], and LSTM [6]) use an open-skull approach where a human programmer selects features or controls the hidden connections. Thus, traditional supervised learning corresponds to type $seb = 110$. However, supervised learning using a skull-closed DN, used in an earlier version of our system (3Deye) [7], corresponds to type $seb = 010$. Our motor-supervised learning, $seb = 010$, is different from traditional motor-supervised learning, $seb = 110$, because we require everything inside the skull to be off-limit to human access and unsupervised. Motor signals cannot directly dictate any hidden neuron, only indirectly affect it.

Authorized licensed use limited to: Michigan State University. Downloaded on December 15,2021 at 01:49:12 UTC from IEEE Xplore. Restrictions apply.

In this second version of 3Deye [7], we investigate types $seb = 010$, $seb = 001$, and $seb = 000$.

Real-time and interactive teaching of type $seb = 010$ is challenging because it requires the teacher to continuously provide the motor actions (e.g., stereo disparity, where, what, and navigation actions) on time. Providing reinforcement signals (i.e., bad or good) for the desired motor actions is easier than providing actual values.

In past work, our DN was able to integrate motor-supervised learning $seb = 010$ and reinforcement learning $seb = 001$ [8], [9] by allowing the environment to conduct motor-supervision and reinforcement.

DNs avoid using any symbolic model or time-discount model that causes a greedy behavior without a long-term value system. Instead of a value r in Q-learning [10], punishments and rewards are modeled by biologically inspired biased sensors. Pain sensors cause a release of "serotonin" and sweet sensors cause a release of "dopamine". The biased neural modulators are diffusely transmitted to the entire brain [11].

In this work, we also conduct $seb = 000$ type learning, called *communicative learning*, during which the system generates its own actions and observes the effects without direct motor or reinforcement supervision from the human trainer. If there are any interactions from the teacher, the interactions are from "unbiased sensors" such as video cameras and microphones.

We believe that the future of machine learning is in type $seb = 000$, which is likely the most powerful since this mode seems to occur most commonly in human learning. However, we do not expect great performance improvements for this first attempt at communicative learning, other than creating more smooth clusters in hidden neurons which tend to improve the feature detection and attention.

The novelties of the multi-teaching aspects of this work are the $seb = 000$ communicative learning, and the integration of the three types of learning: motor supervised, reinforcement, and communicative.

II. System Architecture

We ran 3DEye on a Sony G8142 mobile phone attached to a GENISAMA VCML-100 to satisfy the following mutually conflicting constraints: (1) real-time speed, (2) low cost, and (3) high mobility. See [7] for details of design considerations, DN architecture, and the graphical user interface.

A. DN Algorithm

The DN, detailed in [12], takes Z as the state/action and X as input. It has a hidden area Y in which each hidden neuron has different receptive fields in X and in Z. We should consider Z as a state in a finite automaton (FA).

The receptive field from Z to each Y neuron (effective receptive field) is globally connected so that each hidden neuron receives inputs from all Z neurons, as shown in Fig. 1.

Each hidden neuron in Y takes $\mathbf{x}(t) = (\mathbf{x}_l(t), \mathbf{x}_r(t))$ where $\mathbf{x}_l \in X_l$ and $\mathbf{x}_r \in X_r$ denote the left image and right image, respectively, and X_l and X_r are the space of left images and

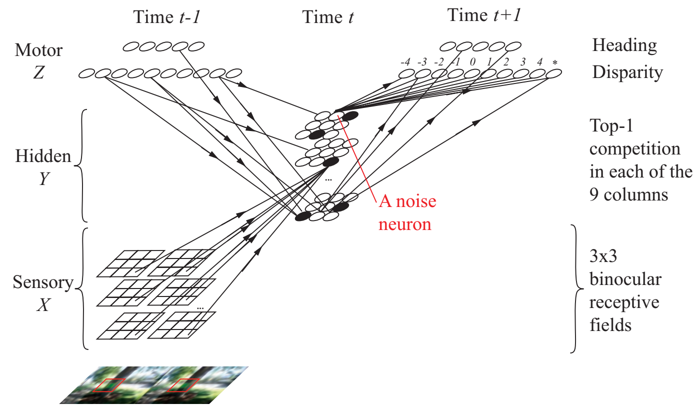

Fig. 1. The architecture of the 3DEye network as DN-1 with motor neurons needing monocular and binocular cues. All hidden Y neurons emerge ML-optimally.

the space of right images, respectively. Although not explicitly stated, each hidden neuron only takes a small patch of $\mathbf{x}(t) = (\mathbf{x}_l(t), \mathbf{x}_r(t))$.

The DN algorithm is presented below:

Algorithm 1 (DN Algorithm): Areas: X: sensory; Y: hidden (internal); Z: motor
Input areas: X and Z.
Output area: Z.
The dimension and representation of the X and Z areas are based on the sensors and effectors of the agent.

1) At time $t = 0$, supervise initial state $\mathbf{z}_0$. Grab the first sensory input $\mathbf{x}_0$, all the weights are random with neuronal ages being zero. $\mathbf{y}_0$ being a zero vector.
2) At time $t = 1, 2, 3, ...$, grab $\mathbf{x}_t$. Each neuron in Y and Z areas update in parallel using dually optimal LCA [13].

B. DN Reinforcement Learning

The effects of serotonin and dopamine on motor neurons and hidden neurons are different.

For motor neurons, the effects are to weaken or strengthen the pre-action potential of the target motor neurons, which determines their likelihood to fire. Computationally, each motor neuron has a learned serotonin weight w_p and a learned dopamine weight w_s, that are the age-dependent amnesic averages of past serotonin and dopamine quantities. The firing neuron i, in motor zone j, learns its pain and sweet weights (w_{ip} and w_{is}) from its serotonin r_{jp} and dopamine r_{js}:

$$w_{ip} \leftarrow w_1 w_{ip} + w_2 r_{jp}$$

$$w_{is} \leftarrow w_1 w_{is} + w_2 r_{js}$$

where w_1 and w_2 are the age-dependent amnesic retention rate and learning rate, respectively in Eq. (1) [13].

The "unbiased" pre-action potential r_{iu} of a firing neuron i is relatively weakened by serotonin $0 \leq w_{ip} \leq 1$ and is relatively strengthened by dopamine $0 \leq w_{is} \leq 1$ to become the biased pre-action potential (z_i) that participates in the top-k competition for motor-neuron firing in the future:

$$z_i = r_{iu}\gamma(1 - \alpha w_{ip} + \beta w_{is})$$

Authorized licensed use limited to: Michigan State University. Downloaded on December 15,2021 at 01:49:12 UTC from IEEE Xplore. Restrictions apply.

(e.g., $\alpha = 0.9$, $\beta = 0.3$ and $\gamma = 1$) [8].

For hidden neurons, the effects are to increase the learning rate of the target neurons so that they can better memorize such important events—punishments and rewards [9] relative to mundane unbiased events. Computationally, the learning rate w_2 and retention rate w_1 of the firing hidden neuron are modified by $w_{is} + w_{ip}$ in the following way:

$$w_2 = \min\left\{(1 + w_{is} + w_{ip})\frac{1+\mu(a_i)}{a_i}, 1\right\}; w_1 = 1 - w_2. \tag{1}$$

where $\mu(a_i)$ is the amnesic function [13] depending on the firing age of neuron i.

III. Theory

The lack of *autonomous development* in most learning systems has lead to multiple problems in standard AI research. By autonomous development, we mean that the brain inside the skull is inaccessible to human teachers.

A. PSUTS

Post Selection Using Test Sets (PSUTS) is the experimental practice of using test sets in post selection of systems and is one problem propagated by the lack of autonomous development in intelligent systems.

Suppose all data is denoted as set D. It is divided into three disjoint sets, training set T, validation set V, and test set T', $D = T \cup V \cup T'$. The training set is used to train a network, the validation set is the exam used for validating whether the trained network does well, and the test set is the real exam. The test set T' should not be available to the system developers.

If the system developers want to reduce the bias in dividing between T and V, they may use a validation method, such as cross-validation, where they divide all the data in T and V into f folds and do f experiments. In the i-th experiment, $i = 1, 2, ..., f$, they use the i-th fold as the validation set V and the remaining data as the training set T, reporting the average error of these f networks as the validation error.

However, the developers must choose one of their networks with only the validation error, and without its performance on the test set T' because T' is secret.

If some competition organizers made the test set available to all competitors, allowing competitors to see test set errors, then competitors could use those results to find the network with the smallest error among their large group of networks. This would mean the competition has fake performance because the method for determining the performance is improper.

In a typical neural network publication, the test set T' is self-collected or downloaded from a public site. This, unfortunately, causes the improper research practice of PSUTS.

All traditional neural networks, trained through gradient-based methods such as error back-propagation, suffer from the local minima problem. The DN, however, is unique in that it does not suffer from this problem and is ML-optimal.

B. Developmental errors

A traditional network is a classifier that approximates a static mapping from a domain X of vectors to a set S of symbols, where each symbol corresponds to a class. The DN, on the other hand, considers two entities concurrently, all the sensors form the sensory space X and all the effectors for the motor space Z. The hidden area Y, which emerges automatically from X and Z, is considered a part of the temporal context, defined by

$$\mathbf{c}_t \triangleq (\mathbf{x}_t, \mathbf{y}_t, \mathbf{z}_t)), t = 0, 1, 2, ... \tag{2}$$

where $\mathbf{x} \in X$, $y \in Y$ and $\mathbf{z} \in Z$, indexed by discrete time t.

Since the resulting network is extremely recurrent, we must unfold time t for the network as the following Table shows:

TABLE I
Unfolding Time in the Development of DN

Time sample index	0	1	2	...
Actable world W_z	$W_z(0)$	$W_z(1)$	$W_z(2)$	...
Motor Z	$Z(0)$	$Z(1)$	$Z(2)$	...
Skull-closed brain Y	$Y(0)$	$Y(1)$	$Y(2)$	...
Sensor X	$X(0)$	$X(1)$	$X(2)$	...
Sensible world W_x	$W_x(0)$	$W_x(1)$	$W_x(2)$	...

Unlike many neural networks that suffer from the so-called wait-for-convergence problems, each column in the table can use only the information in the immediately left column so the network responds in real time.

We treat X and Z in Table I as external because they can be "supervised" by the world as well as "self-supervised" by the network itself. We require the area Y to be internal and hidden from external teachers.

Since Weng 2015 [14], we require a DN to be ML-optimal under three conditions: (1) the system restrictions (i.e, the body's sensors and effectors); (2) the teaching experience; (3) the computational resources including the number of hidden neurons. For performance evaluation, we record all the errors that take place throughout the lifetime:

Definition 1 (Developmental error): The developmental errors of a developmental network $N = (X, Y, Z, M)$ with sensory area X, hidden area Y, motor area Z, and memory M, run through lifetime by sampling at discrete time indices $t = 0, 1, 2, ...$ as $N(t) = (X(t), Y(t), Z(t), M(t))$. Starting at $t = 0$ with supervised sensory input $\mathbf{x}_0 \in X(0)$, initial state $\mathbf{z}_0 \in Z(0)$, random weight vector $\mathbf{y}_0 \in Y(0)$, and initial memory $\mathbf{m}_0 \in M(0)$, the network N recursively and incrementally updates at each time $t = 1, 2, ...$:

$$(\mathbf{x}_t, \mathbf{y}_t, \mathbf{z}_t, \mathbf{m}_t) = f(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{z}_{t-1}, \mathbf{m}_{t-1}) \tag{3}$$

where f is the Developmental Program (DP) of N. If $\mathbf{z}_t \in Z(t)$ is supervised by the teacher, the network complies and the error $e_t = 0$ is recorded. Otherwise, N generates a motor vector $\mathbf{z}_t$ and its deviation from the desired $\mathbf{z}_t^*$ is recorded as error e_t. The lifetime average error from time 0 to time t is

$$\bar{e}(t) \triangleq \frac{1}{t}\sum_{i=0}^{t} e_i = \frac{t-1}{t}\bar{e}(t-1) + \frac{1}{t}e_t \tag{4}$$

Authorized licensed use limited to: Michigan State University. Downloaded on December 15,2021 at 01:49:12 UTC from IEEE Xplore. Restrictions apply.

where the second equation gives an incremental way to compute the average.

Under the above Three Conditions, DN-1 in [14] and DN-2 in [12] were proven to be optimal in the sense of maximum likelihood. This is independent of the initial weights because a different set of weights may affect which neurons fire first but the resulting network is performance equivalent to all other DNs given the same data. Thus, there is no need for PSUTS.

C. Motor Neurons that Need Binocular Cues

There are two types of motor neuron: those that need binocular cues and those that need monocular cues.

Binocular cues are useful for near-range depth perception [15]. The farther an object is from the viewer, the less the binocular disparity is observable.

Our LCA (Lobe Component Analysis) theory indicates that "phase" is a resulting phenomenon caused by statistics-based competition. Such a "phase" phenomenon has already been reported by our earlier work [16] using a large number of locally shifted monocular natural images as inputs to simulate stereo effects. This earlier work reached a sub-pixel accuracy for developmental stereo-disparity detection and was the first to use this biologically inspired approach.

For stereo-disparity detection, the output actions are binocular disparities. However, because the DN is task-nonspecific, following the idea of AMD (Autonomous Mental Development) [17], its output can be trained to generate actions other than disparity (navigation, object/sound recognition, etc) [18].

An important mechanism for dealing with weak-texture is *subwindow voting*. Within a mask of 3×3 subwindows, each subwindow is analyzed by a column of many hidden neurons. The output disparity is the voting result from the subwindows, weighted according to each subwindow's analyzed certainty.

It is also important that the absolute disparity value between the left and right receptive fields accounts for a small proportion of the subwindow size so that a given neuron's input from the left and right subwindows overlaps. In contrast to human eyes, the binocular cameras of 3DEye are fixed and parallel. This means that infinity has a zero disparity.

As infinity is not a common fixation point, we select a 3D fixation point that is near the center of the most likely fixation region. We simulate the effect of the human convergence motor by globally shifting both images horizontally so that the calibrated 3D point has a zero disparity. The 3D points should then have smaller absolute values of disparities because the disparities values now take both negative and positive values.

D. Motor Neurons that Need Monocular Cues

Unlike laser scanners, information from a single video camera is rich. The cues that human monocular vision [19] utilizes to perceive depths include accommodation, linear perspective, interposition, shading and lighting, texture gradient, relative size, aerial perspective and motion parallax [15].

Consider that we teach 3DEye to navigate a walkway bounded by lawns and bushes with possible obstacles such as pedestrians. Although disparity information is useful for detecting obstacles, disparities do not provide information about the walkway. Monocular information is vital in this situation and in similar applications like autonomous driving.

The general-purpose nature of a DN does not require a user to handpick features inside the skull. All he needs to do is to link the motor to the real navigation effectors, such as the steering wheel, the gas pedal, and the brake.

E. Multisensory Integration

We have the following theorem.

Theorem 1 (Multisensory abstraction): Suppose that a DN has two types of motor neurons, disparities and heading direction. Then, the disparity motor neurons automatically establish connections from those hidden neurons that are tuned to the specific disparity values regardless (invariant to) monocular cues. Similarly, the heading-direction motor neurons automatically establish connections from those hidden neurons that are tuned to the specific monocular cues regardless (invariant to) disparity values.

Proof: Without loss of generality we assume $k = 1$ in top-k competition within each column. According to the proof in [14], each of the 9 columns has 1 neuron firing that corresponds to the best match between the LCA weight vector and the binocular image patch at the corresponding receptive field. Let us consider two cases, (a) disparity neuron and (b) heading neuron. (a) When a disparity neuron fires, 9 hidden neurons fire but the supervised motor neuron corresponds to the nearest receptive field per learning rule to detect the nearest object. Thus, the hidden neurons corresponding to the smallest disparity (nearest) are linked with the motor neuron per Hebb's rule. The value of the connection corresponds to the probability for the hidden neurons to fire, conditioned on the motor neuron is firing. The firing neurons in the hidden areas also establish their connections with the firing motor neurons, to update the conditional probability per [14]. Because a disparity motor neuron is supervised to fire only when the disparity value is observed, the motor neuron pulls all such hidden neurons tuned to this disparity regardless of monocular cues. The proof for (b) is similar and therefore is omitted. ■

IV. Experimental Results

The purpose of the following experiments was to illustrate how the DN integrates monocular and binocular cues to perform a difficult task and test the effects of the $seb = 010$, $seb = 001$, and $seb = 000$ teaching modes.

The data consisted of three stereo image sequences (Nav-1, Nav-2, Nav-3) collected in real-time from an outdoor walkway setting like the one in Fig. 1. Nav-1, Nav-2, and Nav-3 contained 6502, 6686 and 5508 frames, respectively.

We let the DN live through 14 sessions, as shown in Table II. The DN has a growth rate with which about half of the 900 neurons per column ($750 \times 9 = 6750$ total hidden neurons) are activated in session 1 and half session 3. The hidden area used $k = 8$ for top-k in each of the 9 columns, the disparity motor area used $k = 3$ ($k = 1$ for heading and stop/go), and the top down weight was set to 0.1 like that in [7].

Authorized licensed use limited to: Michigan State University. Downloaded on December 15,2021 at 01:49:12 UTC from IEEE Xplore. Restrictions apply.

TABLE II
AVERAGE ERRORS THROUGH "LIFETIME" SESSIONS WITHOUT REINFORCEMENT

Session	Nav	Mode	Disparity	Heading	Stop/Go
1	1-even	Motor-S	0.00 px	0%	0%
2	1-odd	Frozen	1.27 px	10%	2%
3	2-even	Motor-S	0.00 px	0%	0%
4	2-odd	Frozen	1.43 px	15%	3%
5	1-odd	Frozen	2.19 px	17%	2%
6	3-odd	Frozen	2.74 px	20%	3%
7	3-even	Practice	2.71 px	21%	3%
8	3-odd	Frozen	2.70 px	23%	4%
9	3-even	Practice	2.67 px	24%	3%
10	3-odd	Frozen	2.70 px	23%	4%
11	3-even	Practice	2.68 px	26%	3%
12	2-odd	Frozen	1.65 px	15%	2%
13	1-odd	Frozen	2.31 px	21%	3%
14	3-odd	Frozen	2.67 px	26%	3%

TABLE III
AVERAGE ERRORS THROUGH "LIFETIME" SESSIONS WITH REINFORCEMENT

Session	Nav	Mode	Disparity	Heading	Stop/Go
7	3-even	Reinforce	2.79 px	49%	39%
8	3-odd	Frozen	3.06 px	14%	4%
9	3-even	Reinforce	2.73 px	65%	53%
10	3-odd	Frozen	2.60 px	15%	4%
11	3-even	Reinforce	2.43 px	15%	3%
12	2-odd	Frozen	2.74 px	20%	3%
13	1-odd	Frozen	2.26 px	13%	4%
14	3-odd	Frozen	1.27 px	10%	2%

We defined three motor concepts for our DN to learn and labeled each frame of the image sequences for each motor. The disparity is the same as defined in [7] with a range of -5 to 9. The heading is the direction in which the navigation system should move: left, straight, or right. The stop/go motor concept contains two neurons: stop and go.

While all three navigation sequences contained a range of disparity values between -4 and 5, Nav-1 exclusively contained the value of -5 disparity and Nav-2 exclusively contained the range 5 to 9. Since all motor neurons must be linked before reinforcement learning to have a nonzero chance of firing, we split each navigation sequence into odd frames and even frames to create six total sequences. Combining both Nav-1-odd with Nav-2-odd and Nav-1-even with Nav-2-even gave the full range of disparity values, and we used the two combinations as disjoint sets.

To begin our experiments we first trained with motor-supervision (seb=010) on Nav-1-even (session 1), tested on Nav-1-odd (session 2), and then trained on Nav-2-even (session 3). After session 3 we tested the DN in frozen mode on Nav-2-odd and Nav-1-odd again (sessions 4 and 5). To get the DN's performance on an unseen sequence, we ran the frozen DN on Nav-3-odd (session 6).

Since DNs are performance-equivalent, we took the DN that ran on sessions 1-6 and created two identical copies to be trained differently from that point forward. This is why the errors for sessions 1-6 are only reported once in Table II and not in Table III and why there is a horizontal line separating sessions 1-6 in Table II from the rest of the sessions.

We let one DN learn on Nav-3-even with reinforcement (seb=001), and then tested its performance on Nav-3-odd in frozen mode (sessions 7-8 in Table III). Since reinforcement may need to take multiple epochs, we repeated sessions 7-8 two more times (sessions 9-14). In session 7, we punished the disparity motor if the error was above 4 and rewarded it if the error was less than 2. We then decreased these thresholds by one for each following reinforcement session. The heading and stop/go motors, on the other hand, were always punished if their prediction was incorrect and rewarded if it was correct. We set $\alpha = 0.9$, $\beta = 0.3$, and $\gamma = 1$ so that reinforcement quickly changes the DN's behavior.

To compare the effects of reinforcement against self-practice (seb=000), sessions 7-14 in Table II ran without reinforcement. The hidden neurons were updated with LCA and all other motor neurons were left to update freely without reinforcement.

For both DNs, we repeated test sessions 4-6 to see if the DN forgot its earlier experience. In both Tables, "Motor-S" means "motor-supervision", "Frozen" means the DN does not learn. "Practice" means the DN is running freely and updating using its own actions, and "Reinforce" means the DN motors are being reinforced. The session-wise average disparity error is shown under the Disparity column and the classification error (percent of incorrect classifications) for the heading and stop/go motors is shown under those respective columns.

A. Performance

In Table II, the DN that underwent only practice sessions had slightly improved disparity errors after the practice sessions. This improvement showed that practice mode is effective; the neurons were able to update their weights to become more generalized as seen in Fig. 3.

There are three important conclusions from Table III: (1) The motor-supervised training (seb=010) is effective in disjoint tests. (2) The reinforcement training (seb=001) on Nav-3-even improved disjoint tests on Nav-3-odd. (3) Such reinforcement training has negligable effects on the performance of other sequences (1 and 2) meaning their memory is not tangibly damaged. Overall, the multisensory and multi-teaching scheme is experimentally confirmed.

B. Visualization

We would like to provide a visualization for the features the DN has automatically developed. For visualization purposes, $k = 1$ for top-k competition in the Y and Z areas. Nav-1 was used for training and then Nav-2 was used for practice.

Fig. 2 shows how some stereo road images participated in the updates of a bottom-up feature, which is averaged with more sensory inputs from age 1, to 101. This specific neuron fired once while training and then about 100 times while in the practice mode. This shows that the neuron was able to teach itself through passive updates. The disparity values of the Z neurons that are linked to this hidden neuron show that it trained on values that were either correct or nearly correct.

Fig. 3 shows a neuron that was relatively noisy (concrete surface) until after age 60. Then, the neuron matched a few

Authorized licensed use limited to: Michigan State University. Downloaded on December 15,2021 at 01:49:12 UTC from IEEE Xplore. Restrictions apply.

Fig. 2. Some road features developed.

Fig. 3. Some more general features developed.

images with a larger disparity well enough that it learned those weights and became tuned to that new disparity value.

Fig. 4 shows the bottom up weights of the motor neuron that corresponds to disparity -4 (center subwindow).

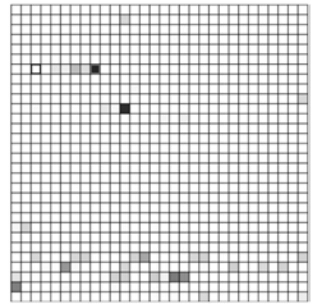

Fig. 4. Visualization of the bottom up weights of the motor neuron for disparity -4. A darker intensity means a larger value.

V. Conclusions

As far as we are aware, this paper presented the first multi-sensory learning system that uses multiple teaching modalities. This work represents a major step towards enabling systems to not only be motor-supervised, but also to be self-taught through real-time practice, with or without reinforcers.

Our results also showed that the DN is able to take advantage of the rich data that cameras provide by seamlessly integrating monocular and binocular visual information. Thus, the DN could be a robust and inexpensive perception system, making it competitive against laser-based systems.

Future work will involve fully autonomous learning (seb=000 mode) where motor-supervision is self-generated.

References

[1] D. J. Lewkowicz and A. A. Ghazanfar. The emergence of multisensory systems through perceptual narrowing. *Trends in Cognitive Sciences*, 13(11):470–478, 2009.

[2] D. H. Hubel and T. N. Wiesel. The period of susceptibility to the physiological effects of unilateral eye closure in kittens. *Journal of Physiology*, 206(2):419–436, 1970.

[3] J. Weng. *Natural and Artificial Intelligence: Introduction to Computational Brain-Mind*. BMI Press, Okemos, Michigan, 2012.

[4] K. Fukushima, S. Miyake, and T. Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Trans. Systems, Man and Cybernetics*, 13(5):826–834, 1983.

[5] T. Serre, T. Poggio, M. Riesenhuber, L. Wolf, and S. Bileschi. High-performance vision system exploiting key features of visual cortex. *US Patent*, US7606777B2, Sept. 1 2006.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[7] J. A. Knoll, V. N. Hoang, J. Honer, S. Church, T. H. Tran, and J. Weng. Fast developmental stereo-disparity detectors. In *Proc. IEEE International Conference on Development and Learning and Epigenetic Robotics*, pages 1–6, Valparaiso, Chile, Oct. 26-27, 2020.

[8] Z. Zheng, K. Qian, J. Weng, and Z. Zhang. WWN: Integration with coarse-to-fine, supervised and reinforcement learning. In *Proc. International Joint Conference on Neural Networks*, pages +1–8, Beijing, China, July 7-13 2014.

[9] Z. Zheng, K. Qian, J. Weng, and Z. Zhang. Modeling the effects of neuromodulation on internal brain areas: Serotonin and dopamine. In *Proc. International Joint Conference on Neural Networks*, pages +1–8, Dallas, TX, 2013.

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

[11] J. Weng. *Natural and Artificial Intelligence: Introduction to Computational Brain-Mind*. BMI Press, Okemos, Michigan, second edition, 2019.

[12] J. Weng, Z. Zheng, and X. Wu. Developmental Network Two, its optimality, and emergent Turing machines. U.S. Provisional Patent Application Serial Number: 62/624,898, Feb. 1 2018. Published.

[13] J. Weng and M. Luciw. Dually optimal neuronal layers: Lobe component analysis. *IEEE Trans. Autonomous Mental Development*, 1(1):68–85, 2009.

[14] J. Weng. Brain as an emergent finite automaton: A theory and three theorems. *International Journal of Intelligent Science*, 5(2):112–131, 2015.

[15] M. Kalloniatis and C. Luu. The perception of depth. In H. Kolb, E. Fernandez, and R. Nelson, editors, *Webvision: The Organization of the Retina and Visual System*, pages 1139–1148. University of Utah Health Sciences Center, Salt Lake City, UT, 2007.

[16] M. Solgi and J. Weng. Developmental stereo: Emergence of disparity preference in models of visual cortex. *IEEE Trans. Autonomous Mental Development*, 1(4):238–252, 2009.

[17] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599–600, 2001.

[18] J. Weng, Zejia Zheng, Xiang Wu, and Juan Castro-Garcia. Auto-programming for general purposes: Theory and experiments. In *Proc. International Joint Conference on Neural Networks*, pages 1–8, Glasgow, UK, July 19-24 2020.

[19] M. Dattilo, V. Biousse, B.B. Bruce, and N.J. Newman. Functional and simulated visual loss. In M. Hallett, J. Stone, and A. Carson, editors, *Handbook of Clinical Neurology*, volume 139, pages 229–341. Elsevier, Amsterdam, Netherlands, 2016.

Authorized licensed use limited to: Michigan State University. Downloaded on December 15,2021 at 01:49:12 UTC from IEEE Xplore. Restrictions apply.