Motor Initiated Expectation through Top-Down Connections as Abstract Context in a Physical World

Matthew D. Luciw and Juyang Weng
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan 48824
Email: {luciwmat, weng}@cse.msu.edu

Shuqing Zeng
Electrical Controls and Integration Laboratory
R&D Center, General Motors Inc.,
30500 Mound Road, Warren, MI 48090
Email: shuqing.zeng@gm.com

***Abstract*—Recently, it has been shown that top-down connections improve recognition in supervised learning. In the work presented here, we show how top-down connections represent temporal context as expectation and how such expectation assists perception in a continuously changing physical world, with which an agent interacts during its developmental learning. In experiments in object recognition and vehicle recognition using two types of networks (which derive either global or local features), it is shown how expectation greatly improves performance, to nearly 100% after the transition periods. We also analyze why expectation will improve performance in such real world contexts.**

I. Introduction

We use temporal context to guide visual recognition. Normal human adults do not experience the world as a disjointed set of moments. Instead, each moment contributes to the context by which the next is evaluated. Is our ability to utilize temporal context in decision making innate or developed? There is evidence that the ability may be developed from a child's genetic programming so that it emerges several months after birth. From much experimental evidence, Piaget [7] stated that before they are around 10 months old, children do not visually experience objects in sequences, but instead as disassociated images. Many of Piaget's tests measured overt behavior and could not measure internal decisions, however. Baillargeon [1] later found evidence of object permanence, an awareness of object existence even when the objects are hidden from view, in children as young as 3.5 months. It illustrated an aspect of the covert (internal) process behind recognition. Evidence of this awareness in significantly younger infants is not supported. How does this ability to predict emerge? Does it emerge from interactions with the environment (i.e., would a child in an environment obeying different laws of physics learn to predict differently?) or is it genetically programmed to emerge at a specific time?

From the above, it can be seen that, after sufficient developmental experience, children can generate an internal expectation signal that is useful for biasing prediction. The network basis for generating this signal is not clear. Object permanence, specifically the drawbridge experiment, may be a special case that can be solved through a set of "occlusion detectors", such as those found in the superior central sulcus in the ventral visual pathway [2]. How the brain creates prediction signals in general relates to the fundamental question of how the brain represents time. Buonomano [4] discussed the two prevalent views of how this may be – "labeled lines", in which each neuron's firing can represent events on different timescales, or "population clocks", where the temporal information is represented by the overall population dynamics of local neural circuits. In the latter, each individual neuron carries little timing information. These local circuits have many feedforward and feedback internal connections, but also connections from other areas, where external events arise and perturb this network's (circuit's) state.

The network models we present here are similar to the types of local circuits discussed by Buonomano. A key difference is the external stimulus originates from the external environment, as they are images. And the output of the networks controls an agent's behavior. But internally, they are guided by Hebbian rules and competitive inhibition, arranged in multiple layers, and connected through feedforward, lateral, and feedback connections. Feedback connections[1] are prevalent throughout all areas of the cortex, and there is short-range and long-range feedback. This paper introduces the new mechanism of *expectation* in the context of these cortex-inspired neural network circuits, which are connected to external sensors (e.g., camera) and motors (e.g., output tokens, or labels). Expectation is carried out by delayed feedback connections. The application area is classification problems in object recognition. However, these networks are general purpose – not task-specific to object recognition – and can handle any type of input signal. The power of expectation to lead to a better recognition rate is

This work is supported in part by General Motors Research and Development.

[1]These are also called *top-down* connections, since we formulate with the outputs (called motors, after robot motor control and biological motor cortex) at the top and inputs (called sensors, after robot input sensors and biological sensory cortex) at the bottom.

978-1-4244-2662-1/08/$25.00 ©2008 IEEE

studied and shown through experiments.

Computational mechanisms to develop object permanence were presented in [11], using a incremental hierarchical regressor (not biologically plausible). That work introduced the *priming* mechanism, by which the next sensory context and action was predicted by the robot, and this prediction could be compared to the actual next events as a way to measure novelty. Priming was formulated as a one-frame prediction regression problem, with a prototype updating queue used for longer prediction. In [5], a decay function was applied to neuron firing rate, thereby imposing continuity from one frame to the next. In [3], a local interconnected circuit model that learned to represent different timescales was presented. A key aspect to their ability to learn time was their short-term synaptic plasticity.

This is the first time where the effect of internally generated *expectation* has been studied for a biologically-plausible (e.g., each neuron adapts via Hebbian rules) end-to-end network, applied to a practical engineering problem – object recognition. The top-down connections are naturally, within the context of cortical-like processing, extended to provide an expectation signal for the next output. This paper shows how this improves performance over networks not utilizing expectation when the input streams follows the laws of temporal continuity. Networks that develop both global (entire image) and local (image part) features are tested, and it is shown that expectation in both cases greatly improves the performance.

This paper is organized as follows. Section II presents the network architecture, highlighting the new expectation mechanism. Section III provides analysis. Section IV describes experiments and presents results. Section V gives some broader perspective for how to utilize this mechanism in mentally developing agents. Section VI concludes the paper.

II. Network Function and Architecture

The networks applied here (Fig. 1) are modeled after those developed in previous work [8]. It adds the new expectation capability to the network type from [6]. That network is called monolithic, or global, since each neuron that is connected to the sensory layer takes input from every single sensory component. A second network type, called local, is presented in section IV, also with expectation capability. Both these types of networks are motivated by and are for use in Autonomous Mental Development [9] (AMD).

Structurally, there is a set of layers, based on the layered architectures found in cortex. Each layer is a set of neurons, arranged in a grid on a 2D plane, with equal distance between any two adjacent (non-diagonal) neurons. The system is end-to-end – from sensors to motors (output labels in the case of classification). The number of layers in between the sensors and motors depends on the implementation. Here we deal with networks having three layers, with a sensor (pixel) layer, a layer-one that performs feature extraction and detection from images, guided by the top-down response from layer-two, and a layer-two that performs feature extraction and detection from the response from layer-one. The firing rate of layer-two

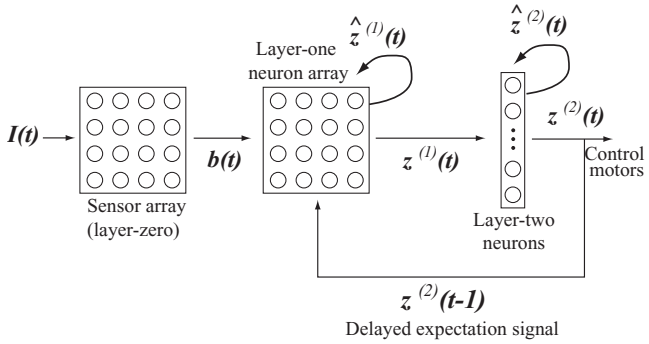

Fig. 1. Architecture diagram of the networks studied in this paper, which also introduces some notation. Applied to visual class recognition, $\mathbf{I}(t)$ is an image from a discrete video sequence, represented as bottom-up activity $\mathbf{b}(t)$ to layer-one. The layer-one neural grid performs top-down controlled feature extraction and detection, while the output of layer two (non topographic) is used in control (e.g., class recognized). The pre competition responses are denoted as $\hat{\mathbf{z}}^{(1)}(t)$ and $\hat{\mathbf{z}}^{(2)}(t)$. After lateral inhibition and smoothing, the layers' firing rates are $\mathbf{z}^{(1)}(t)$ and $\mathbf{z}^{(2)}(t)$. The delayed expectation signal is the layer-two response is fed back to layer-one after a one-frame delay $\mathbf{z}^{(2)}(t-1)$.

neurons controls the overall output – the current classification decision.

Based on cortical neurons, neurons in this model have bottom-up, lateral, and top-down input projections. We do not use explicit lateral connections in this paper, but instead make use of approximate methods for ease of use and computational reasons. The explicit input to these neurons can be divided into two parts: bottom-up input activity from the previous layer and the top-down input activity from the next layer. Each neuron's sensitivity is modeled by a weight for each "axonal" input line from the bottom-up and top-down.

A network operates at discrete times $t = 0, 1, ...$[2]. The outside stimulus is an image stream. So, at each time, there is an image $\mathbf{I}(t)$, which becomes represented as an activity vector $\mathbf{b}(t)$ from the sensor array (pixels), considered as layer zero. The following is the algorithm for computation in the three-layer (sensors to topographic layer-one to non-topographic layer-two) and globally connected version of the networks.

1. Pre-response – Neuron i's on layer-one computes its pre-competitive response $\hat{z}_i$ – called *pre-response*, linearly from the bottom-up part and top-down part

$$\begin{aligned} \hat{z}_i(t) &= (1-\alpha) \cdot \frac{\mathbf{b}(t) \cdot \mathbf{w}_{b,i}^{(1)}(t)}{\|\mathbf{b}(t)\| \, \|\mathbf{w}_{b,i}^{(1)}(t)\|} \\ &+ \alpha \cdot \frac{\mathbf{z}^{(2)}(t-1) \cdot \mathbf{w}_{e,i}^{(1)}(t)}{\|\mathbf{z}^{(2)}(t-1)\| \|\mathbf{w}_{e,i}^{(1)}(t)\|} \end{aligned} \quad (1)$$

where $\mathbf{w}_{b,i}^{(1)}(t)$ and $\mathbf{w}_{e,i}^{(1)}(t)$ are this neuron's bottom-up and

[2]Unless it is necessary, we will not include time such as (t) in all the following equations.

top-down weight vectors, respectively, and $\mathbf{z}^{(2)}(t-1)$ is the firing rates of layer-two neurons from the last time step. The expectation parameter α is discussed in detail later[3].

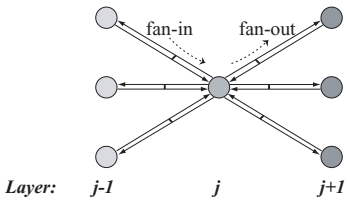

Fig. 2. In a general multi-layer network, each neuron has bottom-up and top-down connections to and from itself. The fan-in weight vector deals with bottom-up sensitivity while the fan-out weight deals with top-down sensitivity. In the model presented here, the weights are shared among each two-way weight pairs.

2. Competition via Lateral Inhibition – A neuron's pre-response is used for intra-level competition as follows. The neurons with the $k^{(l)}$ highest pre-responses, on layer l, are considered winners. The others are inhibited. The top pre-response $\eta = \text{argmax}_{\hat{z}_i} \hat{z}_i^{(l)}$. Rank the pre-responses $r_i = \text{rank}(\hat{z}_i^{(l)})$ where the rank function is defined such that $r_\eta = 1$. Then, the response $z_i^{(l)}$ of a neuron is $z_i^{(l)} = \text{s}(r_i)\,\hat{z}_i$ where

$$\text{s}(r_i) = \begin{cases} 1/\eta & \text{if } 1 \le r_i \le k^{(l)}, \\ 0 & \text{if } r_i > k^{(l)}, \end{cases} \tag{2}$$

and this ensures the maximum response is one and the next highest $k-1$ are scaled appropriately.

3. Smoothing via Lateral Excitation – Lateral excitation means that when a neuron fires, the nearby neurons in its local area are more likely to fire. This leads to a smoother representational map. The topographic map can be realized by not only considering a nonzero-responding neuron i as a winner, but also its 3×3 neighbors, which are the neurons with the shortest distances from i (less than two). To realize lateral excitation, take neuron j that is a neighbor of a neuron i with a non-zero response, then set the response of neuron j to $z_j^{(1)} = h(d_{ij})$ where d_{ij} is the distance between the neurons and h is a neighborhood response function. However, do not do this if this will actually decrease $z_j^{(1)}$. Layer-one uses this mechanism but layer-two does not.

4. Hebbian Updating with LCA – After inhibition, all neurons with non-zero response – these are called the winners – are allowed to fire and update their synapses. For a winner cell i, update the weights to be more like the stimulus using the lobe component updating principle, described in [10] and [8].

Updating Top-Down Weights – How are the layer-one top-down weights updated? Here, the layer-two bottom-up weights are copied directly to become the layer-one top-down weights. Let the layer-two weight matrix be $\mathbf{W}^{(2)}$ of dimensions $n \times c$, where c is the number of classes and therefore number of layer-two neurons. The columns of this matrix (layer-two fan-in) are the bottom-up synaptic weights of the layer-two neurons. Then the rows (layer-one fan-out) gives the top-down synaptic weights of the layer-one neurons. Figure 2 may provide some elucidation.

Motor layer – Layer two develops using steps 1,2, and 4 above, but there is not top-down input, so Eq. 1 does not have a top-down part. The response $\mathbf{z}^{(2)}$ is computed in the same way otherwise, with its own parameter $k^{(2)}$ controlling the number of non-inhibited neurons.

When the network is being trained, $\mathbf{z}^{(2)}$ is imposed originating from outside (e.g., by a teacher). In a classification problem, there are c layer-two neurons and c possible object classes. The true class being viewed is known by the teacher, who communicates this to the system. Through an internal process, the firing rate of the neuron corresponding to the true class is set to one, and all others set to zero. Note that this will effect the next frame's training, so it is essential to have a significant amount of class repetition frame-by-frame in the video sequence.

III. Analysis

A. Expectation

Expectation is realized by $\mathbf{z}^{(2)}(t)$ affecting the layer-one response at time $t+1$. In Eq. 1, α controls the maximum influence of top-down versus the bottom-up part. This bottom-up, top-down coupling is not new [6]. The novelty for this paper is twofold: first, the top-down activation originates from the *previous* time step $(t-1)$ and second, nonzero top-down parameter ($\alpha > 0$) is used in the testing phase. These simple modifications create a temporally sensitive network. It is important to understand the inter-layer dynamics that differentiate this method from e.g., temporal smoothing of the output. The key mechanism is the delayed top-down response of the classification output vector. The high responding layer-two neurons (e.g., the correct classes neuron) will boost the pre-responses of the layer-one feature neurons correlated with those neurons. It will boost these neurons more than the feature neurons mainly correlated with other classes. This in effect biases select feature neurons to fire (which would indicate the actual detection of these features in a purely bottom-up network). These neurons' firing leads to a stronger chance of firing of certain layer-two neurons (indicating output decision) without taking into account the actual next image. This top-down signal is thus generated as an expectation of the next frame's class. The actual next image also stimulates feature neurons to fire from the bottom up. These two effects interact in ways we will study here.

How does the expectation-enabled network take into account past data? Layer-one maps an image $\mathbf{b}(t) \in \mathcal{X}$ to a response vector $\mathbf{z}^{(1)}(t) \in \mathcal{Z}$. The function to do this (layer-one) is $\mathbf{z}^{(1)}(t) = L_1(\mathbf{b}(t), M(t))$ where $M(t)$ is the "mental state" of the network. In expectation networks, old data gets integrated into the current state, *even if the weights are frozen*[4],

[3]We do not utilize linear or non-linear function g, such as a sigmoidal, on each firing rate in this paper since we found it unnecessary in this implementation.

[4]Weight freezing is often done when it is known the system is going into "testing phase"

thereby "hashing" temporal information into spatial. For a network without expectation and frozen weights, the mental state is constant $M(t) = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}\}$. In this case, L_1 always maps $\mathbf{b}_i$ to the *same* point in $\mathcal{Z}$. For a network with expectation, the mental state is $M(t) = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{z}^{(2)}(t-1)\}$. The effect of an image at $t-1$ persists within the network, transformed to $\mathbf{z}^{(2)}(t-1)$. It contributes to the next response at a level α. Therefore, for each new image frame, the effect of the old one decays multiplied by α. That leads to the following property:

Property 1: The effect on the current network state of an image encountered at time $t-n$ is measured as α^n.

When $\alpha = 1$, the network state disregards subsequent image frames entirely. When $\alpha = 0$, the network operates in a frame-independent way[5].

B. Centered single-object recognition

Here, we assume there is a single object centered within the field of view of the agent. Scenes with multiple objects or an object that greatly varies its location through time (within the scene) are out of the scope of this paper. The true state the agent is attempting to determine is what type of object it is, e.g., the true label $\mathbf{y}^*(t)$. This label is known to a nearby teacher, who can interact with the system to provide it (teaching). However, it is desirable the system learn to accurately identify the objects itself.

Why should expectation improve the performance? Each layer-two neuron defines a hyperconal region in $\mathcal{Z}$, centered on the lobe components that are the columns of $\mathbf{W}^{(2)}$ (see Appendix). A layer-one response $\mathbf{z}^{(1)}$ is simply classified based on which region it falls into. But these decision boundaries in the response space may not be accurate due to several factors such as too few neurons, or poor features. This means that the response distribution from class i versus class j may not be linearly separable.

From Eq. 1 it can be seen that, when $\alpha = 1$, the response $\mathbf{z}^{(1)}(t)$ will be a linear combination of the layer-two lobe component directions, where the coefficients come from $\mathbf{z}^{(2)}(t-1)$. This is due to the shared weights between layer-one top-down and layer-two bottom up weights. If coefficient j is one and the rest zero, for example, the layer-one response will be exactly the lobe component direction corresponding to class j.

The two parts of Eq. 1 each lead to different points in $\mathcal{Z}$. The bottom-up part (if $\alpha = 0$) gives the networks response without considering past images – unit vector $\mathbf{z}_B^{(1)}(t)$. The top-down part (if $\alpha = 1$) gives a linear combination of class centers in response space – unit vector $\mathbf{z}_T^{(1)}(t)$.

Assume expectation is biased for a particular class. Then it "pulls" $\mathbf{z}_B^{(1)}(t)$ towards the layer-two lobe component representing that class. Since expectation takes into account the recent class history the most, this is advantageous when the probability that the next image contains the same class is high. This is called the temporal continuity principle. Expectation will reduce the effect of outlying points across the wrong decision boundary by pulling them back towards the class center, leading to more correct classifications under temporal continuity.

How can this system transition from one output class to the next without external intervention? First, the layer-one features derived must provide class distributions in $\mathcal{Z}$ that are mostly separable. Second, α must not be set too high. After a class transition in the video, the system should transition to output the correct class with a few errors immediately after the transition as it adjusts (e.g., the boosted layer-one features from the now-incorrect class are not supported from the bottom-up image and gradually stop firing). If the class distributions are too mixed up, or α is set too high, there will be errors, and perhaps "hallucinations", where the system never adapts to output the correct class.

IV. Experiments

A. MSU 25-Object Data Set

Fig. 3. Sample(s) from each of the 25 objects classes, also showing some rotation. In the experiments, the training images were 56×56 and grayscale.

25 small objects were selected (see Fig. 3) and captured in 3D rotation. 200 images of 56 rows and 56 pixels were taken in sequence for each object. At the operator's rate of rotation, the 200 images covered about two complete rotations of 360 degrees for each object. The capture process was intentionally not too controlled, so an object varies slightly in position and size throughout its sequence. The background was controlled[6] by placement of a uniform color by a sheet of gray fabric. Including an additional "empty" (no object) class, there were $200 \times 25 + 1 = 5001$ images total. Every fifth image in each sequence was set aside for testing, so the other 80% were used to train the networks. Grayscale images were used[7].

We trained five different networks with 20×20 neurons over ten epochs using $\alpha = 0.3$ in the training phase. The images were trained in different class sequences, with a few empty (no object) frames in between each sequence to mimic

[5]Note the above depends on a large enough $k^{(1)}$ and $k^{(2)}$. Further study is needed to precisely quantify the effect of these inhibition parameters.

[6]Due to automatic adjustment of the overall image intensity by the camera's capture software, later background color normalization was done.

[7]These images are available at http://www.cse.msu.edu/ei/datasets.htm.

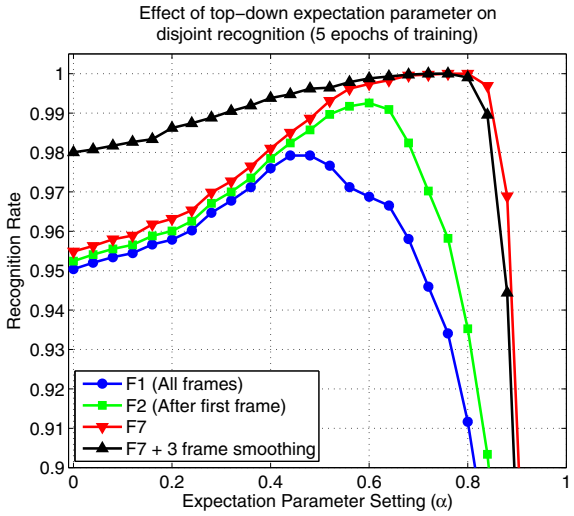

Fig. 4. The effect of different expectation parameter values on performance. The number following F indicates the frame to start measuring performance after a sequence shift to a new rotating object sequence. Three frame smoothing means the latest three outputs are summed and the maximum class output used to get the current output.

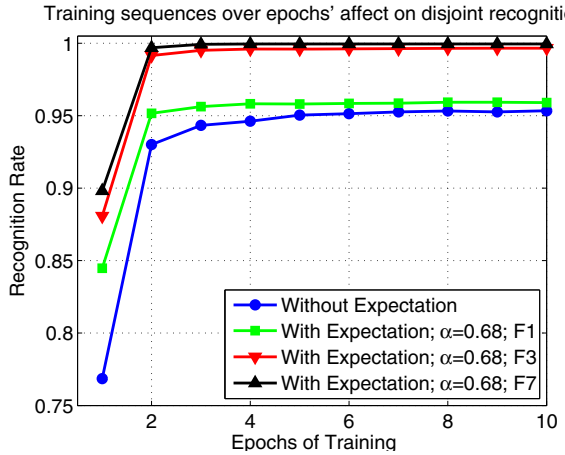

Fig. 5. How performance improves over epochs through all object sequences.

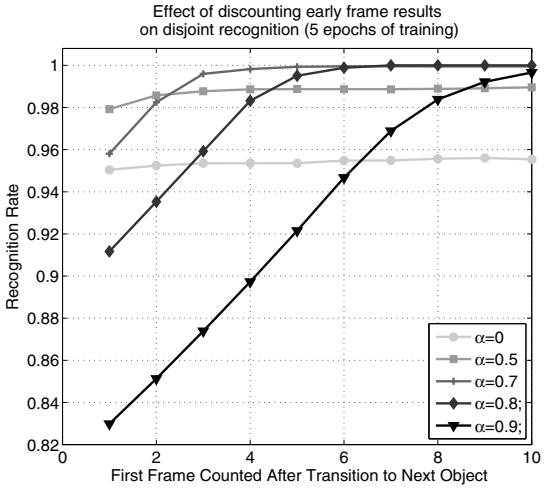

Fig. 6. Increased α shifts errors from bottom-up misclassifications to errors within the sequence transition periods.

an object being placed and taken away. The disjoint images were tested after each epoch, also presented in sequences each in between a few empty frames, under varying network conditions[8]. Figure 4 shows the effect of different α in testing after 5 epochs. It can be seen that expectation leads to near-perfect performance except for some errors immediately after a class-transition. Figure 5 measures how training the same sequences over and over again can help performance. It helps a lot to see the same sequence at least twice. Figure 6 shows how the transition period is affected by α. Increasing expectation eventually leads to no errors except in the transition periods. But higher α will have longer transitions. It would be allowable for there to be a brief period of confusion on transitions for robotic agents.

B. Vehicle data using local features

As could be seen in Section III, the power of expectation depends on the features derived from the training data. If the features derived allow layer-two to nearly linearly separate the classes, expectation becomes very powerful. In this section we test global features versus local features on a vehicle / non-vehicle discrimination problem. We deal with classes that are decomposable into visible parts (e.g., headlight and license plate on a car). Local features could become tuned to parts, which should improve the generalization power of the network, meaning the performance will be better with less data.

Fig. 7. Six examples of the vehicles class and six from the "not-a-vehicle" class.

In a global feature network, each neuron on layer-one is sensitive to every element within the sensor matrix, which has d elements (pixels). In the local network used here, each neuron on layer-one has a $r \times r$ receptive field, where r^2 is less than d. The number of neurons is equal to the number of pixels, and each neuron's receptive field is centered on a particular pixel[9]. The competition step is also local. A neuron competes with its local region of $l \times l$ neurons. The *local* top-$k^{(1)}$ responding neurons are called winners, in the local version. The local network used here did not utilize smoothing, and α was gradually increased in training from 0 to 0.3 after 40 training samples.

There were 225 vehicle images and 225 not-vehicles images, each sized 32×32. Some examples can be seen in Fig. 7.

[8]We found it was best to set $k^{(1)} = k^{(2)} = 1$ in training, but to increase them in testing. They were held constant at $k^{(1)} = 15$, $k^{(2)} = 8$ for the tests.

[9]The image boundaries are extended with black pixels

We perform a generalization test, comparing a global network (10×10 neurons), a local network ($r = 11$, and $l = 5$), and another local network using expectation after it matures. The networks were initialized and trained with only 5 samples from each class. Then all samples were tested, in two sequences (cars $\rightarrow$ not cars). Next, the next 5 samples were trained, and so on, until all samples were trained. We did this 10 times for each network type, using a different training order, for statistical reasons. Results are summarized in Fig. 8. The local network does indeed do better with less data, however it eventually only does just as well as the global network. If expectation is enabled however, the performance becomes nearly perfect.

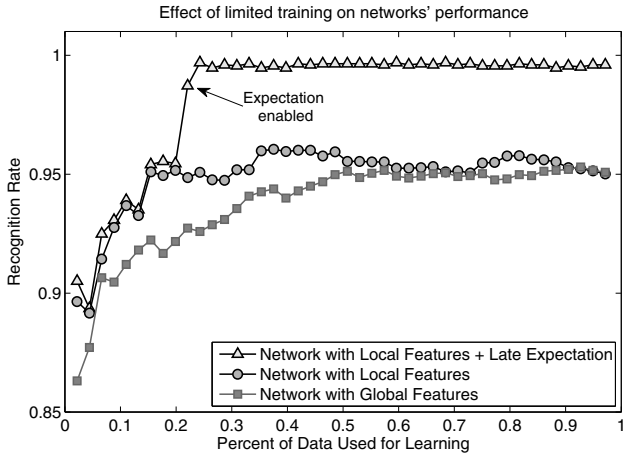

Fig. 8. Performance of globally and locally connected networks when data is limited. The locally connected network performs better with less data for this dataset. This may be because these vehicle images can be made up of several interchangeable parts. Once training is mature, the expectation mechanism can be enabled in testing, and performance shoots up to a nearly perfect level.

V. AMD Applications

From the perspective of developmental robotics and autonomous mental development, the mechanism described in this paper can lead to more realistic learning ability for mentally developing machines. Supervised machine learning methods that can be applied to visual recognition of objects are formulated at a per-frame level where each training sample (frame) must be accompanied by a label. In classical supervised learning, each x (e.g., input frame) is associated with a y (e.g., output label). However, in realistic communicative training, the training signal is sparse in time – the teacher may only speak the name of the viewed object once as it rotates,for example. When a child is taught, say to recognize numerals and letters in the classroom, there is not a continual stream of repetitive speech from the teacher speaking the names of the characters over and over. The teacher will 1. direct the child's visual attention and 2. speak the name of the character. The direction of attention hopefully ensures that the child is looking at the character continuously.

Using this paper's architecture, a semi-supervised learning mode can be utilized, which can significantly reduce the training load of the teacher. In this mode, expectation is enabled and the weights are not frozen. This will be very useful, since the common distinctions of "training phase" from "testing phase" cannot be made so easily in a real time, developmental system. Then, the purpose of the teacher is to sparsely provide labels, and correct errors. The error correction can operate as follows. A signal giving the true class from an external source will impose strong supervision at the motor end (a vector of all zeros and a single one corresponding to the correct class). This, in effect, interrupts the internally generated top-down expectation and replaces it with the correct expectation in the strongest form.

VI. Conclusions

Although temporal context has been used in many task-specific models and in artificial neural networks, this is the first time where context is used as motor initiated expectation through top-down connections in a biologically plausible general-purpose developmental network. If the motor action represents abstract meaning (e.g., recognized class) these mechanisms enable meaningful abstract expectation by such networks. We showed the effect in improving recognition rate to nearly perfect performance after a transition period when the class has first changed. Expectation's effectiveness in improving performance is linked to the capacity to develop discriminating features. Expectation was shown in both globally and locally connected networks, and the local features are better for generalization (better performance with less training). For future work, we will utilize this in a real world developmental system to "train itself" to reduce the training load of the teacher.

References

[1] R. Baillargeon. Object permanance in 3.5 and 4.5-month-old infants. *Developmental Psychology*, 23:655–664, 1987.

[2] C.I. Baker, C. Keysers, J. Jellema, B. Wicker, and D. I Perrett. Neuronal representation of disappearing and hidden objects in temporal cortex of macaque. *Exp. Brain Res.*, 140:375–381, 2001.

[3] D. V. Buonomano. Decoding temporal information: a model based on short-term synaptic plasticity. *J. Neuroscience*, 20(3):1129–1141, 2000.

[4] D. V. Buonomano and U.R. Karmarkar. How do we tell time? *Neuroscientist*, 8:42–51, 2002.

[5] J. Krichmar and G. Edelman. Machine psychology: autonomous behavior, perceptual categorization and conditioning in a brain-based device. *Cerebral Cortex*, 12:818C830, 2002.

[6] M.D. Luciw and J. Weng. Topographic class grouping with applications to 3d object recognition. In *Proc. International Joint Conf. on Neural Networks*, Hong Kong, June 2008. accepted and to appear.

[7] J. Piaget. *The Construction of Reality in the Child*. Basic Books, New York, 1954.

[8] J. Weng, H. Lu, T. Luwang, and X. Xue. A multilayer in-place learning network for development of general invariances. *International Journal of Humanoid Robotics*, 4(2), 2007.

[9] 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.

[10] J. Weng and N. Zhang. Optimal in-place learning and the lobe component analysis. In *Proc. World Congress on Computational Intelligence*, Vancouver, Canada, July 16-21 2006.

[11] J. Weng, Y. Zhang, and Y. Chen. Developing early senses about the world: 'object permanence' and visuoauditory real-time learning. In *Proc. International Joint Conf. on Neural Networks*, Portland, Oregon, July 20-24 2003.