

Unsupervised Learning in Hybrid Cognitive Architectures

Yury Vinokurov and Christian Lebiere

Carnegie Mellon University
Department of Psychology
5000 Forbes Ave
Pittsburgh, PA 15212

**Dean Wyatte, Seth Herd,
and Randall O'Reilly**

University of Colorado, Boulder
Department of Psychology
Muenzinger D251C
345 UCB
Boulder, CO 80309

Abstract

We present a model of unsupervised learning in the hybrid SAL (Synthesis of ACT-R and Leabra) architecture. This model follows the hypothesis that higher evaluative cognitive mechanisms can serve to provide training signals for perceptual learning. This addresses the problem that supervised learning seems necessary for strong perceptual performance, but explicit feedback is rare in the real world and difficult to provide for artificial learning systems. The hybrid model couples the perceptual strengths of Leabra with ACT-R's cognitive mechanisms, specifically its declarative memory, to evolve its own symbolic representations of objects encountered in the world. This is accomplished by presenting the objects to the Leabra visual system and committing the resulting representation to ACT-R's declarative memory. Subsequent presentations are either recalled as instances of a previous object category, in which case the positive association with the representation is rehearsed by Leabra, or they cause ACT-R to generate new category labels, which are also subject to the same rehearsal. The rehearsals drive the network's representations to convergence for a given category; at the same time, rehearsals on the ACT-R side reinforce the chunks that encode the associations between representation and label. In this way, the hybrid model bootstraps itself into learning new categories and their associated features; this framework provides a potential approach to solving the symbol grounding problem. We outline the operations of the hybrid model, evaluate its performance on the CU3D-100 (cu3d.colorado.edu) image set, and discuss further potential improvements to the model, including the integration of motor functions as a way of providing an internal feedback signal to augment and guide a purely bottom-up unsupervised system.

Introduction

Dynamic environments present a major challenge for object recognition systems trained on static datasets. Because traditional object recognition techniques depend on a training regime in which the ground truth is known, they do not allow the system to dynamically develop its own categories of classification based on environmental information. As an

Copyright © 2012, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

alternative to classifiers that employ traditional supervised machine learning algorithms, we present an approach based on a hybrid framework that incorporates two separate cognitive architectures: ACT-R and Leabra. This framework, which we call SAL (Synthesis of ACT-R and Leabra) combines Leabra's strength as a perceptual neural network with ACT-R's symbolic and sub-symbolic theory of declarative memory and procedural action. Crucially, we demonstrate mutually reinforcing learning in the hybrid SAL framework, which allows the SAL model to dynamically discover and learn representations of novel objects.

ACT-R

ACT-R is a production rule-based cognitive architecture which implements the rational analysis theory developed by Anderson (Anderson and Lebiere 1990), (Anderson et al. 2004). ACT-R separates functional elements such as declarative memory, visual processing, and motor control into modules, which in turn expose constructs called "buffers" to the core procedural module. Event execution is dictated by the consequents of production rules, which fire when their antecedent conditions represented over the set of buffers are met. In the event that more than one rule's antecedents are satisfied, the production with the highest utility is selected. The utility of a production is learned according to whether a production helps the model achieve its goals, and is back-propagated through time up the production chain, so that rules closer to the goal are rewarded more, while rules more temporally distant from the goal receive proportionately less reward (Anderson and Fu 2004).

Declarative memory

ACT-R stores its declarative memories as "chunks," structures containing places ("slots") to which values (which can be data objects of any kind, including other chunks) can be assigned. A request to the declarative memory module specifies the conditions required for chunk retrieval to occur, and the module either returns a chunk that corresponds to those specifications or signals a retrieval failure.

The main quantity which controls retrieval from declarative memory is the chunk's "activation," a measure of both its relevance and activity in memory. The activation A_i is computed via the equation $A_i = B_i + S_i + P_i + \epsilon_i$, where i is the index of the given chunk, and B_i , S_i , P_i , and ϵ_i , are the

base-level activation, the spreading activation, partial match activation, and noise, respectively. Spreading activation encodes the strength of association between chunks and is active only when chunks have other chunks as slot values; in this model, we do not use spreading activation, so $S_i = 0$ uniformly. Base-level learning encodes the decay of the chunk’s activation as a function of time according to the formula $B_i = \ln\left(\sum_{j=1}^n t_j^{-d}\right)$. Here t_j is the time since the j^{th} presentation of the i^{th} chunk, and d is the decay parameter, which is set to 0.5. This equation captures the power law dynamics of practice and forgetting. The noise parameter controls the width of a Gaussian zero-centered distribution from which the noise is sampled and added to the activation. In order for a chunk to be successfully retrieved, its activation must exceed the global retrieval threshold (RT).

ACT-R supports several retrieval modalities, including exact matching and partial matching. Exact matching will only retrieve a chunk if the exact specifications requested are matched. In many tasks involving continuous values we may wish to retrieve simply the chunk best matched to the request. For this purpose, ACT-R implements a partial matching mechanism. To use partial matching, a metric that gives the similarity between two chunks as a real number between -1 and 0 (with 0 being most similar and -1 being maximally dissimilar) must be defined on the relevant set of chunks. Then the partial match activation is computed as $P_i = \sum_k MP \times M_{ik}$, where MP is a “penalty” paid by chunks dissimilar to the request specifications, which weights the similarity metric M_{ik} . The metric gives the similarity scaled to between 0 and -1 between the value k in the retrieval specification and the corresponding slot of chunk i .

Leabra

As described in (Vinokurov et al. 2011), Leabra is a set of algorithms for simulating neural networks (O’Reilly and Munakata 2000) which combine into an integrative theory of brain (and cognitive) function. These algorithms serve as a computational theory of brain (primarily cortical) function, as the algorithms and their parameterization have been closely guided by a wide variety of data. Leabra operates at an intermediate level of detail (e.g., firing-rate coded, point neurons) to allow modeling effects both down to the molecular effects (e.g., increased sodium conductances from pharmacological manipulation) and up to the cognitive level (e.g., differentially worse performance in the incongruent condition of the Stroop task).

Leabra simulations have explored numerous neuroscientific processes including cognitive control (Herd and O’Reilly 2006) (O’Reilly and Pauli 2010), visual search (Herd and O’Reilly 2005), self-directed learning (Herd, Mingus, and O’Reilly 2010), working memory and executive function in the prefrontal cortex (Hazy, Frank, and O’Reilly 2006), and cortical learning (O’Reilly and Munakata 2002).

Here we used the LVIS (Leabra Vision) model (O’Reilly

et al. 2011). A V1 layer is activated according to a set of filters for line/edge orientation, length sums, and end stopping. This layer is sparsely connected in a convergent manner to a V4 layer, which in turn is similarly connected to an IT layer. Connections from V1 to V4, V4 to IT, and IT to output are allowed to learn using Leabra’s combination of self-organizing and biologically-realistic error driven learning. The IT layer plays a key role in shape-based object recognition which is leveraged by the LVIS model (Riesenhuber and Poggio 1999). The model learns to categorize 100 types of objects and performs at about 93% on generalizing novel exemplars into learned categories. For the present experiments, it was trained on 50 of those object classes (the TRAIN items in the hybrid model), with the remaining 50 reserved as novel (TEST) items.

The SAL Hybrid

The SAL hybrid framework unites Leabra’s detailed account of bottom-up perception with the high-level cognitive functionality modeled by ACT-R (for a full list of applications of ACT-R, see <http://act-r.psy.cmu.edu/publications/index.php>). Previous use of the SAL hybrid is described in (Jilk et al. 2008), in which an ACT-R/Leabra hybrid model was embedded in an Unreal Tournament world.

In the current hybrid architecture, we have wrapped the interaction between ACT-R and Leabra within a *leabra-visual* buffer, which uses a socket-based interface for data exchange. Data received from Leabra is transformed into ACT-R chunks that become available in the buffer, and requests to the buffer control Leabra’s operations. The current implementation of SAL focuses only on vision; however, the architecture is modular and can be extended easily to other functionalities such as motor actions.

In the context of the SAL framework, we define a partial match metric on the distributed representation generated by Leabra’s model of the inferotemporal cortex (IT). The IT representation is encoded as a 200-dimensional vector and the similarity between two vectors is defined using the cosine metric. That is,

$$M_{ik} = \left(\frac{\vec{V}_i \cdot \vec{V}_k}{\|\vec{V}_i\| \|\vec{V}_k\|} \right) - 1 = \cos^2(\theta) - 1 \quad (1)$$

where V_i and V_k represent vectors corresponding to two different IT representations. This metric produces a value between 0 and -1 as required by ACT-R.

The Label Generator Model

Setup

Here we present the label generator model implemented in SAL. The Leabra side of the label generator model consists of a network trained on 50 (out of 100) randomly chosen object classes, with the other 50 being held out of training. The purpose of this method is to establish basic non-random feature detectors within the network and ensure a reasonable signal-to-noise ratio in the IT representations. On the ACT-R side, a model that interfaces with Leabra implements

the cognitive decision procedures involved in learning object categories. The integrated model proceeds according to the following logic:

- ACT-R directs Leabra’s visual attention to an image.
- Leabra processes the image and returns the IT representation.
- ACT-R uses the IT representation as a cue for retrieving an association between the representation and a previously known label.
- If a known label is successfully retrieved, ACT-R directs Leabra to train that representation with the retrieved label.
- If retrieval has failed, ACT-R assumes this is a previously unseen object and generates a label for it. Leabra is then directed to train on that label as above.

This control flow represents a single iteration of the model, after which the next image is presented. The parameters that control the ease of retrieval within ACT-R are the mismatch penalty (MP) and the retrieval threshold (RT). For this series of experiments, we set $MP = 7$ and $RT = -1$, and the noise parameter (ANS) was set to $ANS = 0.25$. These values of RT and ANS are typical for many ACT-R models. The parameters MP and RT, represent a tradeoff between the penalty paid by dissimilar items and the reduced difficulty of retrieval. In a sense, MP controls the size of the neighborhood within which similar representations will converge to a single attractor label, while RT controls the depth of the “well” of memory out of which potential matches have to climb to be recalled. In this case, the value $MP = 7$ represents a fairly discriminating model that heavily penalizes dissimilar representations on recall. Setting $RT = -1$ represents a fairly low barrier to retrieval. Correspondingly, the noise parameter ANS may be seen as characterizing the “fuzziness” of the neighborhood boundary.

Model Evaluation

The challenge of evaluating the performance of the hybrid model consists of the fact that no single metric adequately reflects its activity. Partially this is because within the context of the model there is no real notion of ground truth; although we know the object classes presented to the model, the model dynamically learns their features independently of any external feedback. Furthermore, during the presentation itself, the model can do one of four things: correctly recall a previously seen object (true positive), correctly assign a new label to a previously unseen object (true negative), incorrectly recall a previously seen item (false positive), and incorrectly generate a new label for a previously seen object (false negative).

In general, for the purposes of evaluating the model, we rely on two metrics: the confusion matrix, which gives the fraction of the time that a given response was recalled for a given stimulus, and the overall label histogram, which shows the various labels generated by the model for each object class. Additionally, we present the trained network with the original item set on training to measure whether

the Leabra network (standing apart from the ACT-R model) generates the labels it has been trained on for given stimuli.

Learning in ACT-R With a Fully Trained Network

To test the basic concept of label generation using the hybrid model, we began with a fully trained (to the 93% accuracy on the 100-way test mentioned above) Leabra network and allowed the ACT-R model to develop its own labels for Leabra’s representations. Since the IT representations had converged through network training, they were essentially identical for all exemplars, so only one exemplar per class (selected randomly) was used. The model began with a block of 5 object classes and the entire block was presented 25 times, for a total of 125 presentations per item; presentation order of exemplars within the block was random. At that point, another block of 5 items was added, and the presentations were repeated with the 10-item block for another 25 trials, with the result that every item in the first block would now have been seen 250 times, while every item in the new block would have been seen 125 times. This incremental presentation strategy was repeated until 25 object classes had been added, for a total of $125 + 250 + 375 + 500 + 625 = 1875$ trials. The resulting confusion matrix in Fig. 1 demonstrates that when the IT representation has converged, ACT-R is extremely good at sorting objects into self-created categories. Although there are some shape-based confusions, e.g. *globe* and *skull*, they are very reasonable (IT is sensitive only to shape, not scale, texture, or orientation). Overall, 18 of 25 object classes are perfectly recalled, while the rest are mapped identically onto a pre-existing category.

Simple Random Presentation Schedule

In SAL, the learning process is influenced by both the global parameters and the presentation schedule. The subset of items to be presented can be chosen and modified during the model run time. Initially, we tested the model by running it on a subset of three randomly chosen categories from the set of 50 TEST categories. There were 5 exemplars per category and the model was run for a total of 500 iterations, with each presentation being of a randomly chosen exemplar of a randomly chosen class. All classes and exemplars had an equal probability of being chosen.

The confusion matrix in Fig. 2 demonstrates that for the 3-item case, after 500 iterations the network has converged on a distinct representation for all three items. Slight confusions may occasionally arise between object representations; for example, the *sailboat* is confused for the *scissors* about 2% of the time. The shapes of the three objects (*sailboat*, *scissors*, and *tropicaltree*) in this particular instance are sufficiently dissimilar that a model with a high value for MP will discriminate between them relatively easily. From Fig. 3, we can also see the labels most commonly assigned by the integrated model to the three object categories; although one object category may be assigned multiple labels, all that matters in the end is that the representation and the label assignment both converge

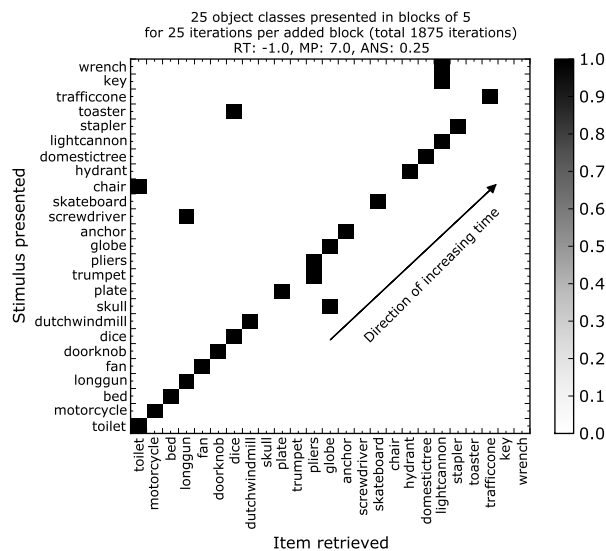


Figure 1: The confusion matrix for the progressive presentation of 25 object classes. Square shade represents fraction of recalls of object on the x -axis given the object on the y axis as a stimulus. All recall values in this case are either 0 (white) or 1 (black). Time of first encounter of object increases uniformly along both x and y axes.

on a stable attractor. In the 3-item case shown in Fig. 4, the network has converged on a set of stable representations.

Staggered Presentation Schedule

To simulate a sort of real dynamic environment in which agents encounter novel objects in a serial fashion, rather than all at once, we broke up a set of 15 object classes into blocks of 3. Each object class was represented by 5 exemplars, and each block of object classes was presented for 500 iterations, as above. Presentation blocks did not overlap, and neither declarative memory nor network weights were reset after each block. As a consequence of this, object classes presented earlier in time tended to form more stable attractors in representation space, while classes presented later in time tended to fall into those attractors if they already possessed an underlying similarity. For example in Fig.5, it can be seen that the *trafficcone* has fallen into the well-established *mailbox* representation, owing to their superficial shape similarity.

Of the 15 items presented in this fashion, 6 are recalled with 89% accuracy or better upon presentation; furthermore, the *rolltopdesk* and the *simpledesk* are so similar that *rolltopdesk* naturally gets folded completely into the *simpledesk* object class. Three further items (*mailbox*, *microwave*, and *toaster*) are recalled correctly 78%, 74%, and 68% of the time. This accounts for 9 of the 15 items. Of the other 6, *trafficcone* and *turntable* collapse completely onto pre-existing categories, *mailbox* and *motorcycle* respectively.

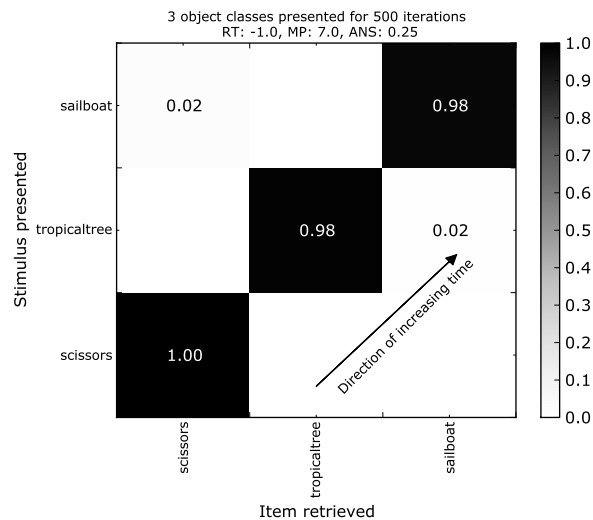


Figure 2: The confusion matrix for a simple presentation schedule in which a block of 3 items is presented for a total of 500 iterations. The arrow shows the increasing direction of time of introduction from left to right and top to bottom, i.e. the first object class encountered is *scissiors*, followed by *tropicaltree* and *sailboat*. The model converges quickly on a stable label assignment for each object class, and recall is at least 98% accurate.

In both cases, this is due to strong shape similarities; most *mailbox* exemplars have a long leg supporting a flared shape (the box itself), which causes resemblance to the triangular shape of the *trafficcone*, while both *turntable* and *motorcycle* exhibit a prominent circular feature. Several other object classes, including the *umbrella* and *trafficlight* never converge on stable representations.

Fig. 6 shows the labels that were assigned to the various object classes. As the confusion matrix shows, some of the object classes create strong attractors into which other object classes fall; for example, the *turntable* and the *motorcycle* were both assigned the label 53. Fig. 7 illustrates the post-test performance of the standalone network. Many of the representations have converged strongly to a single label.

Discussion

The performance of the hybrid model is strongly impacted by the order of presentation and the structure of the IT representation. As the IT representation encodes only shape features in a rotation- and scale- invariant way, similar-looking objects will collapse onto identical self-generated labels. This is all the more true as the training regime features no presentations of negative exemplars; only positive exemplars are possible under this setup. The order of presentation primarily affects which object classes will serve as the initial attractors during training.

Nevertheless, the model is able to learn stable representa-

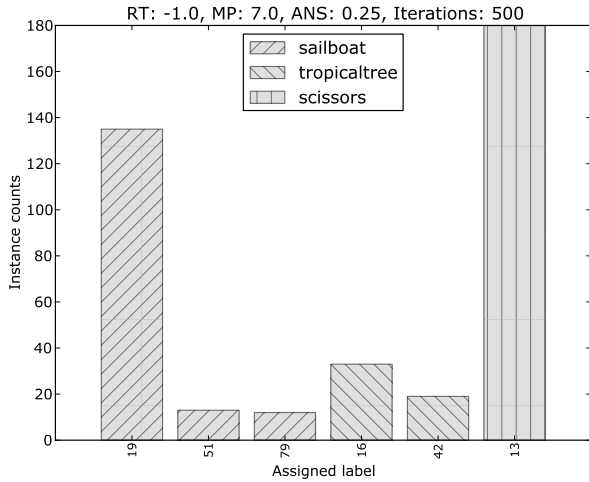


Figure 3: A distribution of the top labels assigned by the model to the presented items and subsequently recalled. The model has converged predominantly on label 19 for *sailboat*, on labels 16 and 42 for *tropicaltree*, and on label 13 for *scissors*. Only those labels which have counts greater than 5 times the minimum number of counts for any label associated with the object class are shown.

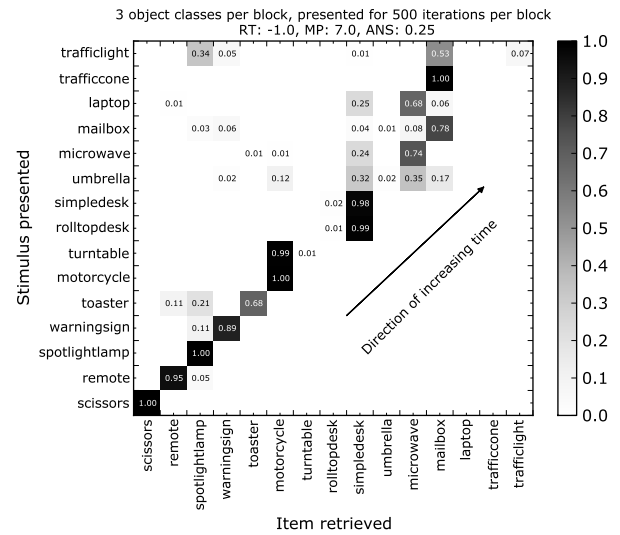


Figure 5: The confusion matrix for a staggered presentation schedule. The arrow shows the increasing direction of time of introduction from left to right and top to bottom, i.e. the first object class encountered is *scissors*, followed by *remote*, etc., and *trafficlight* is encountered last. Presentations within a single block of 3 object classes are random. Only non-zero confusion values are shown.

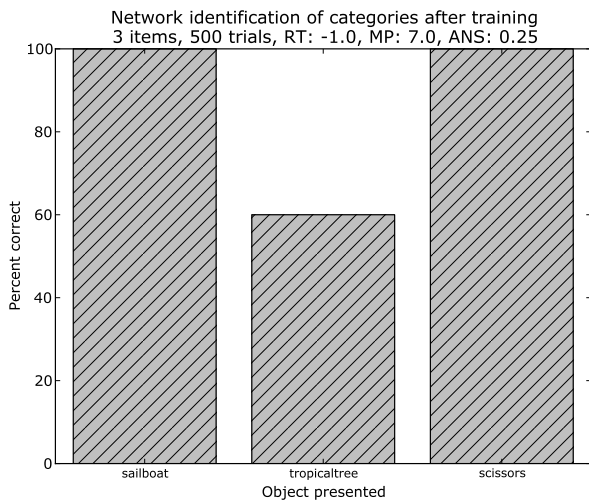


Figure 4: The correctness of the model is evaluated by whether the trained network responds with any of the dominant labels (those having counts at least 5 times greater than the minimum counts associated with the object class) which ACT-R associated with the object class. In this simple 3-item case, the network always correctly identifies the *sailboat* and *scissors*, and correctly identifies the *tropicaltree* 60% of the time. During the random presentations, the *tropicaltree* was seen fewer times, so its label was not as well established.

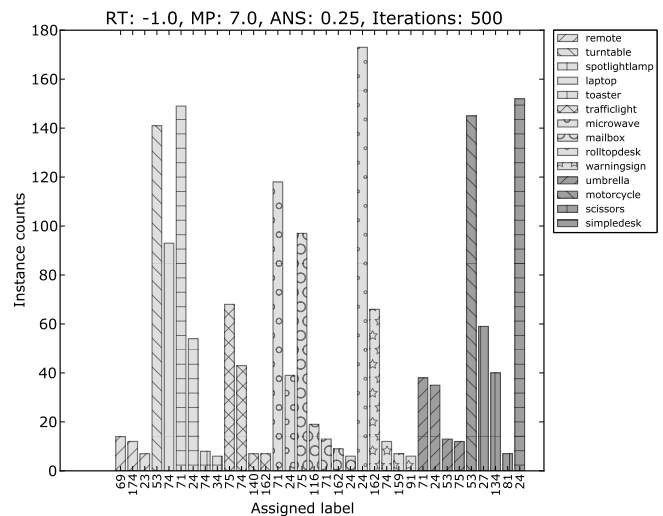


Figure 6: The top labels assigned to the object classes by the integrated model and their counts. Only those labels which exceed the floor threshold of 5 times the minimum number of counts assigned to a label for a given object class are shown.

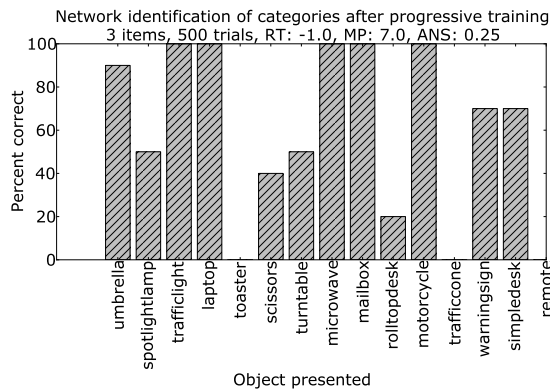


Figure 7: After 500 iterations of training per block of 3 object classes, for a total of 15 categories, the network correctly associates the assigned label with 5 categories 100% of the time, with another 90% of the time, and with two others 70% of the time.

tions for a number of successively presented stimuli, providing evidence that object learning can be bootstrapped from a basic set of features to a richer one. The representations are learned without any supervision, allowing the model to generate its own associations between percept and label. It would be possible to achieve excellent performance on a static data set via traditional machine learning techniques which leverage both a plus phase and a minus phase for training, but such controlled experimental conditions are not typically available to agents attempting to navigate a world. For such autonomous agents within a complex environment full of possibly unknown object classes, the ability to ground labels in perceptions of the external world is a great advantage.

One way of breaking the degeneracy between similar objects involves generating IT representations that contained additional information, such as size and texture. Another improvement would integrate motor information and allow the model to manipulate presented objects; the success or failure of manipulation would provide additional information about the object class. This would even allow for a discrimination of various subclasses of larger classes, as for example the class of doorhandles might be differentiated into ones that can be twisted and ones that can be pushed down. Additionally, metacognitive parameters can be employed to aid in the judgment of object familiarity (Vinokurov et al. 2011).

Conclusions

We have presented a model for self-supervised learning of novel object classes within the modular and extensible SAL framework, which unites the perceptual capabilities of Leabra's neural models with the cognitive functionality of ACT-R. This model provides a basis for implementing unsupervised learning and symbol-grounding in dynamic circumstances by agents navigating through and operating in such environments.

Acknowledgments

This work was conducted through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016.

References

- Anderson, J., and Fu, W.-T. 2004. Extending the computational abilities of the procedural learning mechanism in act-r. In *26th Annual Conference of the Cognitive Science Society*.
- Anderson, J. R., and Lebiere, C. 1990. *The Adaptive Character of Thought*. Hillsdale, New Jersey: Erlbaum.
- Anderson, J. R.; Bothell, D.; Byrne, M. D.; Douglass, S.; Lebiere, C.; and Qin, Y. 2004. An integrated theory of mind. *Psychological Review* 111(4):1036–1060.
- Hazy, T.; Frank, M.; and O'Reilly, R. 2006. Banishing the homunculus: Making working memory work. *Neuroscience* 139(1):105 – 118.
- Herd, S., and O'Reilly, R. 2005. Serial visual search from a parallel model. *Vision Research* 45:2987–2992.
- Herd, S.A.; Banich, M., and O'Reilly, R. 2006. Neural mechanisms of cognitive control: An integrative model of stroop task performance and fmri data. *Journal of Cognitive Neuroscience* 18:22–32.
- Herd, S.; Mingus, B.; and O'Reilly, R. 2010. Dopamine and self-directed learning. In *Proceeding of the 2010 conference on Biologically Inspired Cognitive Architectures 2010: Proceedings of the First Annual Meeting of the BICA Society*, 58–63. Amsterdam, The Netherlands, The Netherlands: IOS Press.
- Jilk, D. J.; Lebiere, C.; O'Reilly, R. C.; and Anderson, J. R. 2008. Sal: An explicitly pluralistic cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence* 20:197–218.
- O'Reilly, R. C., and Munakata, Y. 2000. *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. MIT Press.
- O'Reilly, R. C., and Munakata, Y. 2002. Psychological function in computational models of neural networks. In Michela Gallagher, Randy J. Nelson, I. B. W., ed., *Handbook of Psychology*, volume 3. Wiley. chapter 22.
- O'Reilly, R.C.; Herd, S., and Pauli, W. 2010. Computational models of cognitive control. *Current opinion in neurobiology* 20:257–261.
- O'Reilly, R. C.; Herd, S. A.; Wyatt, D.; Mingus, B.; and Jilk, D. 2011. Bidirectional biological object recognition. *in preparation*.
- Riesenhuber, and Poggio, T. 1999. Hierarchical models of object recognition in cortex. *Nature Neuroscience* 2(11):1019–1025.
- Vinokurov, Y.; Lebiere, C.; Herd, S.; and O'Reilly, R. 2011. A metacognitive classifier using a hybrid act-r/leabra architecture. In *AAAI Workshops*.