State Interpretation for Social Hierarchical Learning

George Wildridge
Social Robotics Lab
Yale University
51 Prospect St, New Haven, Connecticut 06520, USA
gwildridge17@choate.edu

***Abstract*— As robots begin to transition into our everyday lives, a great potential opens for robots to collaboratively aid humans in the completion of everyday tasks. Social Hierarchical Learning is an area of active research within machine learning that aims to enable effective human robot collaboration. This is accomplished through extending the capabilities of hierarchical learning. The long-term goal of this research is to enable a robot to transition from student to peer to teacher. In other words, from having no knowledge about a task, to understanding a task well enough to function as a peer, and finally for a robot to understand a task enough to function as a teacher. Building towards this goal, first and foremost, requires the robot to be able to understand and interpret the environment it is placed in so as to help guide its own actions and to gather information about the task. Utilizing machine learning, both the task and the environment can be understood by the robot visually. Results suggest that utilizing a support vector machine on top of image processing techniques will yield the robot the greatest understanding of the environment.**

***Keywords*—Image Classification; Deep Learning; Softmax Classifier; Support Vector Machine.**

I. INTRODUCTION

As of right now, robots most often execute tasks in rigidly defined state spaces with very small degrees of unpredictability from behind barriers that serve to protect human workers. Social Hierarchical Learning(SHL) aims to bring robots into more collaborative environments. However, doing so requires a robot to have the ability to interact with an unpredictably changing environment and have a high level of self-awareness. In order for a robot to collaborate effectively, it should be able to reduce the physical or cognitive workload of fellow teammates. It sets about these tasks through use of various methods within Machine Learning(ML) including: Markov Decision Processes, Reinforcement Learning, Hierarchical Learning and Learning from Demonstration. For more on these topics, please see [1]. The awareness aspect specifically will be talked about in this paper and it is achieved through the use of deep learning and support vector machines.

Submitted 11/18/16

II. BACKGROUND

A. Image Classification

Image classification uses ML to separate images into different categories, or classes. Three different ML techniques will be explored in this paper: Support Vector Machines(SVM), Feed Forward Neural Networks(FFNN) and Convolutional Neural Networks(CNNs). The basis for all three is a linear classifier. A linear classifier involves an input (an image in this instance) and a set of initially arbitrary parameters known as weights:

$$f = Wx \tag{1}$$

Where x is the image's pixels (stretched into rows) and W is a matrix with as many rows as there are dimensions in the images and as many columns as there are classes. When the two are multiplied together we get a score for each class. Basically we are computing a weighted sum of all the pixel values for each score. Ideally the optimal class would correspond with the highest score value. However, especially in initial stages, this is often not the case. To account for this, a classifier like an SVM or a softmax classifier are introduced to quantify how poorly the linear classifier is working. The value of these classifiers can then be used to gradually shift the weights using gradient descent so as to improve the scores in another pass.

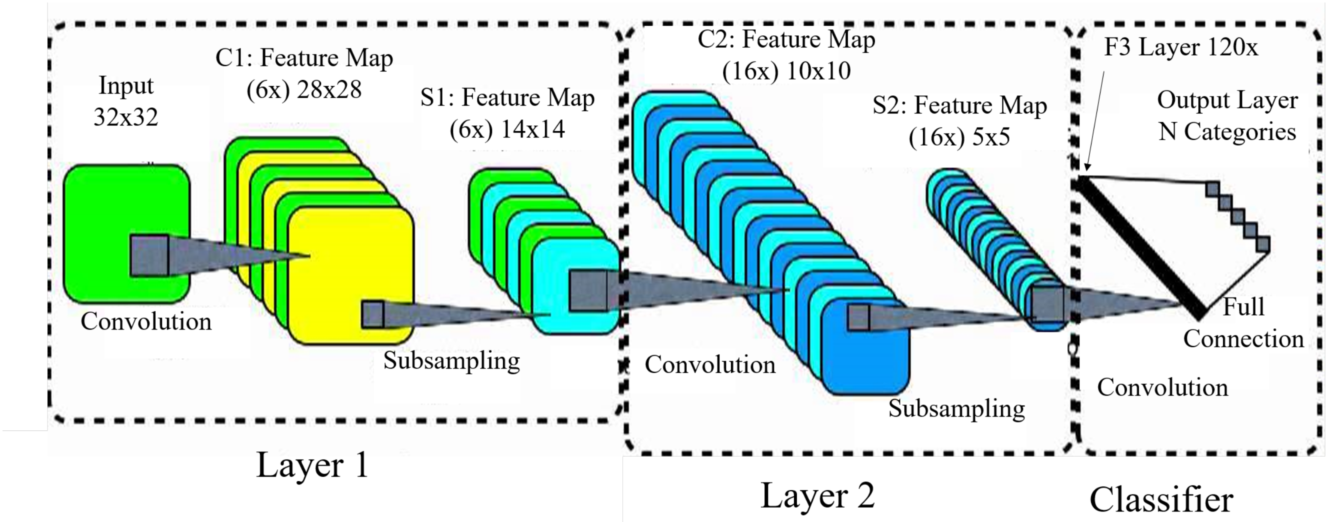

Fig. 1. A graphical model of a convolutional neural network with a fully connected classifier. Full connection represents a feed forward neural network. In this study, the classifier makes reference to the aforementioned softmax classifier. The model gives an example of a two layer convolutional neural network; however, depending on the problem, more or less layers can be used.

After many iterations over the dataset, optimal values for the weights can be found.

B. Deep Learning

The driving force behind deep learning is to recognize and learn feature hierarchies within data. This is most helpful in fields like computer vision.

To understand what deep learning is doing, a great place to start is with a linear classifier, as seen in Eq. 1. The difference between a linear classifier and deep learning are a couple of layers of complexity.Deep leaning, also called neural networks, embeds linear classifiers within each other. To understand this, it is easiest to look at the equation of a three-layer FFNN:

$$f = W_3 \max(0, W_2 \max(0, W_1 x)) \tag{2}$$

and then a two-layer neural network:

$$f = W_2 \max(0, W_1 x) \tag{3}$$

and, again, a linear classifier:

$$f = Wx \tag{4}$$

As more layers are added to the network, so too are more sets of weights. This gives the ability for more detail to be stored about the features within an world state or an image, and to an extent leads to greater precision in classification. Optimizing the weights of neural networks is still equally as important as optimizing the weights of the linear classifier, though it too becomes more complex. Through the recursive application of chain rule, the agent is capable of backpropagating through the neural network in an effort to determine which weights have the most effect on the output. After, these weights can be incremented using gradient descent to help improve the accuracy of the neural network on the next pass. Through repeated iteration, neural networks can become extraordinarily good at classifying certain tasks.

1) Convolutional Neural Networks: A convolutional neural network separates itself from a feed forward neural network in that the images are initially subsampled before being passed through the neural network. This process allows for more data to be obtained from each image and can be seen in Fig. 1.

C. Support Vector Machines

Using the output of a linear classifier given random weights, the intent of the SVM is to gauge the degree of how wrong the classification is. It does this by repeatedly finding the difference between the lowest score and the correct score for each image. With the output we can identify how to change the linear classifiers weights in order to improve classification on its next pass. This can be represented by this equation:

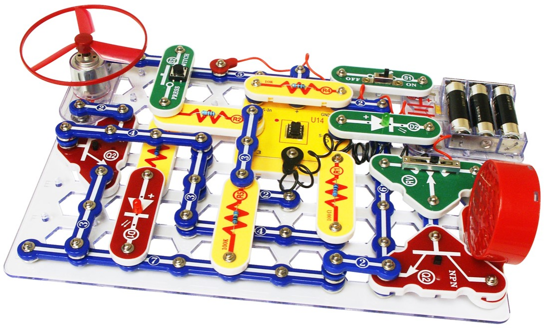

Fig. 2. An image of the snap circuits board. This game allows for the complexity and abstraction of a given task to be modulated.

$$L_i = \sum_{j \neq y_i} [\max(0, w_j^T - w_{y_i}^T + \Delta)] \tag{5}$$

Where Li is the loss for a single training example, x is a vector of images, w is a vector of weights, y is a vector of labels and delta represents the safety margin.

The max is taken in order to clamp possible loss at zero so negative numbers will not have a bearing on the overall loss of the function. If negative loss was allowed to pass through, it would decrease the overall loss which can distort the overall loss and make it seem like we are closer to identifying the correct scores even if we are not. For the overall loss we must average the loss over all the images:

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i \tag{6}$$

Where N is the number of training examples. For optimization, the equation for the gradient with respect to the row of W that corresponds with the correct class:

$$\nabla_{w_{y_i}} L_i = -(\sum_{j \neq y_i} 1(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0)) x_i \tag{7}$$

Where J does not equal y_i the gradient is:

$$\nabla_{w_j} L_i = 1(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0) x_i \tag{8}$$

D. Softmax Classifier

Again using the scores outputted from the linear classifier given random weights, the softmax classifier interprets the scores as the unnormalized log probabilities of the classes. So scores are exponentiated, and normalized before the log is taken to retrieve the probabilities of the classes. We then attempt to maximize the log likelihood and minimize the -log likelihood of the true class. This can be represented by the following equation where Li is the loss for a single example, f is the array of class scores for a single example:

$$L_i = -\log \frac{e^{f_n}}{\sum_j e^{f_j}} \tag{9}$$

The gradient of which would be:

$$\frac{\vartheta L_i}{\vartheta f_k} = p_k - 1(y_i = k) \tag{10}$$

Where P is a vector of the normalized probabilities.

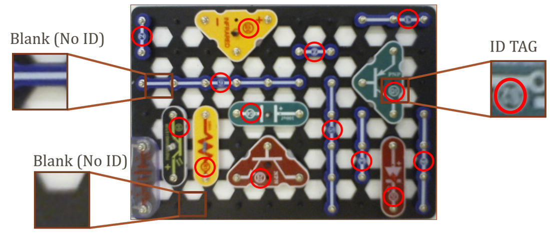

Fig. 3. The image of the snap circuit board would be broken up into smaller images as shown above before being fed to the classifiers.

III. Task Structure

The chosen task for measuring a robots transition between learner, peer and teacher is a childrens game called snap-circuits. Snap-circuits, as seen in Fig. 2, allow for varying levels of task complexity and abstraction as they afford all of the capabilities of regular circuits, but on a larger scale that robots are capable of manipulating. Further, the multiplicity of colors and the labels associated with each peace provides a strong framework for visual recognition. In this study, goals can be set that a human and robot would be asked to work towards, such as make the light bulb light up or make the fan spin.

A robot's understanding of the task environment can be defined as understanding which of the 32 distinct pieces are on the board, where on the board they are located and each pieces respective orientation (north, south, east, west). Obtaining this information from a picture is challenging as it requires a robot to learn to differentiate pieces based on their shape and labels. A feat generally accomplished through extended training on tens of thousands of images.

TABLE I

RESULTS UTILIZING THE APPROACH DESCRIBED IN STAGE ONE WITH OUR DATASET

Stage 1	Correct	Labeled Correct	% of Labeled Predicted to be None
SVM Vertical	86%	0%	100%
SVM Horizontal	90%	50%	0%
Softmax Vertical	90%	0%	100%
Softmax Horizontal	90%	0%	100%
Conv Net Vertical	90%	0%	100%
Conv Net Horizontal	90%	0%	100%

TABLE II

RESULTS UTILIZING THE APPROACH DESCRIBED IN STAGE TWO WITH OUR DATASET.

Stage 2	Classifier	Correct	Labeled Correct	Strayed From Average
SVM Vertical	Label	5%	100%	0%
	Identity	0%	0%	100%
SVM Horizontal	Label	3%	100%	0%
	Identity	50%	50%	100%
Softmax Vertical	Label	90%	0%	0%
	Identity	0%	0%	0%
Softmax Horizontal	Label	90%	0%	0%
	Identity	17%	0%	0%
Conv Net Vertical	Label	90%	0%	0%
	Identity	0%	0%	0%
Conv Net Horizontal	Label	90%	0%	0%
	Identity	0%	0%	0%

IV. APPROACH

After building a dataset, three different approaches were taken in an attempt to yield the best understanding of the board state.

A. Dataset

As no previous dataset existed, the first step was to build a dataset. In all, 390 pictures were taken of the board, with each piece pictured in every possible position. These pictures looked similar to the larger image seen in Fig. 3. These larger pictures were then broken down into 42,120 smaller pictures that consisted of the dataset. The goal of the ML was to label the pictures that contained the physical ID tag as in Fig. 3. These smaller pictures were then labeled with one of the 32 distinct classes and with the pieces orientation. In this study there were 32 distinct classes (one for each part).

B. Stage 1

In the first stage, the data was passed through a SVM, a FFNN with a softmax classifier and a CNN respectively. In total, there were 32 separate output classes, one for each piece and an additional one designating no ID tag.

C. Stage 2

In the second stage, the data was initially relabeled as either having an ID tag or not having an ID tag and passed through a SVM, a FFNN with a softmax classifier or a CNN. Ideally, this separates the ID tagged pieces from the rest, to then be passed on to another respective SVM, FFNN with a softmax classifier or CNN for each pieces respective classification.

D. Stage 3

In the third stage, each larger picture is first run through the standard image processing techniques of contour and color detection in an effort to identify distinct pieces. The length, color and surface area of these distinct pieces could then aid a respective SVM, FFNN with a softmax classifier or CNN in recognition of the pieces.

V. RESULTS

In testing, separate classifiers were run on both the vertical and the horizontal pieces.

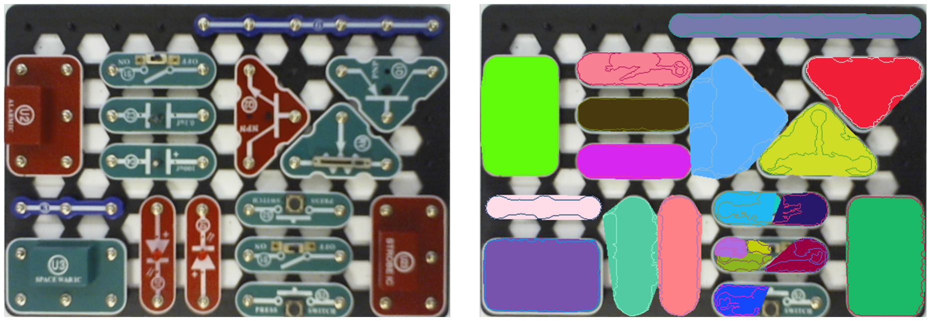

Fig. 4. A demonstration of the use of image processing techniques such as color and contour detection in the third stage of testing.

A. *Stage 1*

In the first stage of testing, an SVM, a FFNN with a softmax classifier, and CNN were run directly on the images as seen on Table 1. Notably, the data shows that the classifiers performed around 90% accuracy on all the data. To gauge how well the classifiers did at giving the ID'ed pictures the correct ID, the third and fourth columns of the table must be looked at. Disappointingly, most of the classifiers performed around 0% in classifying the labeled data. Further most of the classifiers predicted that all of the data. Further, most of the classifiers predicted that all of the data was unlabeled. Although this is certainly not the ideal result, the data shows that the horizontal SVM made significantly more progress in classifying the images then either the FFNN with a softmax classifier or the CNN.

B. *Stage 2*

During stage two of testing, an effort was made to further increase accuracy by creating a classifier whose classifier whose sole purpose is to separate a labeled image from an unlabeled image unlabeled image (shown as the label classifier in Table 2). This then allowed a separate classifier to only distinguish between labeled images(shown as identity classifier in Table 2). Mixed results were again received. In looking at how well the classifiers performed across the SVM, FFNN with a softmax classifier and CNN, the classifiers routinely labeled all the data one class as shown by most of the classifiers never straying from the average. The results of stage two again point to the SVM as being the most promising classifier to solve this problem. However, the lack of significant improvement points to a necessity to either incorporate many thousands more images into the dataset, or take a different approach to labeling through more fundamental image processing techniques like contour and color detection. The problem boils down to there being too few labeled images, and too many unlabeled images for the classifier to make meaningful strides in accurately classifying the dataset.

C. *Stage 3*

Initial steps have been taken in incorporating the more fundamental image processing techniques through the incorporation of contour and color detection. This is best seen in Fig. 4.

VI. Future Work

Future work will focus on further improving the robots visual understanding of the environment using a support vector machine on top of image processing techniques such as contour and color detection. Accuracy on the ID'ed images is expected to rise to around 80% in this context. Focus will then shift to applying artificial intelligence techniques like learning from demonstration and hierarchical learning to enable a robot to shift from learner to peer to teacher.

Acknowledgments

I would like to thank the Social Robotics Lab at Yale University, Dr. Alessandro Roncone, Dr. Olivier Manguin, Dr. Bradley Hayes and Prof. Brian Scassellati for both the tremendous opportunity and their assistance in understanding the material. I would also like to acknowledge Dr. Christopher Hogue and the Science Research Program for their support.

References

[1] Wildridge, G. (2016) A Review of Social Hierarchical Learning.