MARVEL: Modeling Animal Representations

Via Error-Driven Learning

Alejandro Romero

Professor Frank

Brown University, CLPS 1492

Abstract

Most computer models designed for discriminating between animals do so by iterating over visual information from image data with dedicated binary classifiers. This technique results in computer vision networks that learn to solve the object recognition and categorization problem via a prioritization of contour and structural feature information over color and texture information. This is a similar strategy as what studies suggest is how infants discriminate between animals, specifically cats and dogs. It is commonly accepted in the literature that a mental prototype is formed which humans can use to help them assess the similarity of a novel input's perceptual features to determine how to properly categorize it. However, whether computer networks generate such prototypes is unclear from solely assessing their performance on the categorization of cats versus dogs. This paper explores the formation of such "mental" representations in a neural network by analyzing which features are prioritized over others in the network's determination of animal categories.

Modeling Animal Representations Via Error-Driven Learning

Motivation and Background

The object categorization problem is one that is explored in a wide array of literature and computer models in the areas of computer vision and deep learning research. Training models that can reliably glean semantic meaning from observable object features in the environment is a fundamental stepping stone in the fields of virtual reality, autonomous vehicles, machine learning frameworks, and more, as it allows for the kind of object discrimination in real-world scenes that is required to make computers mimic certain human tasks. One such subproblem in the object recognition literature is that of distinguishing cats from dogs (Parkhi et al, 2012). Current models approach this problem by learning categorical differences between the animals via contour and structural feature detection, with texture and color information being of lesser importance (Fleuret & German, 2008; Zhang et al., 2008).

This strategy of object recognition and subsequent categorization is similar to that of infants, as has been proposed in the developmental cognitive science literature. Studies therein have utilized the familiarization-novelty preference procedure to generate empirical data suggesting that, using perceptual information from animal silhouettes, 3- and 4-month old human infants have the ability to form categorical representations for cats and dogs and can discriminate between members of either family (Quinn et al., 2001; Quinn & Eimas, 1996a; Quinn et al. 1993). Though there is debate in the literature, one school of thought suggests that humans make use of mental prototypes for representing object categories and make categorization decisions based on feature similarity between novel perceptual input objects and this prototype (Smith et al., 2016). Although neural networks are able to discriminate between cats and dogs in input images, it is unclear what their version of a "mental prototype" for a cat or dog might look like. Therefore, the motivation for this project was to qualitatively explore what kinds of representations a neural network might make and which features it prioritizes to come to such categorical conclusions.

Approach

To tackle the problem of modeling a network's representations of animal categories, a classic cats and dogs categorization task was designed. The task was created with the purpose of showing how learning can make use of hidden layers of neurons to recode input features to capture the functional similarity of the items in the training environment. The structure of this environment was composed of members of the animal families Canidae (dogs) and Felidae (cats). These families were made up of purely domestic breeds in Project 1, but included exotic members in Project 2 to test the network's performance on more diverse and varied data (though admittedly, the training data in both projects discussed in this paper are quite simplistic and limited in comparison to those of large-scale object recognition models (O'Reilly et al., 2013)).

The feature information for each animal specimen was fed to the network via a TSV (tab-separated values) file, where each input pattern was coded via a binary string. The presence of a feature was denoted by a 1 character in the string, whereas the absence of a feature was denoted by a 0 (***Figure 1***). The list of features that the network had to categorize based on were: animal size, tail size, fur color, ear shape, animal ID, and family name. The animal ID in this case was simply a non-zero integer corresponding to which animal in the sequence was being presented in the input. These values were thus unique for each animal. This animal ID value, as well as the animal family parameter, were also used as the network's outputs to assess correctness during training.

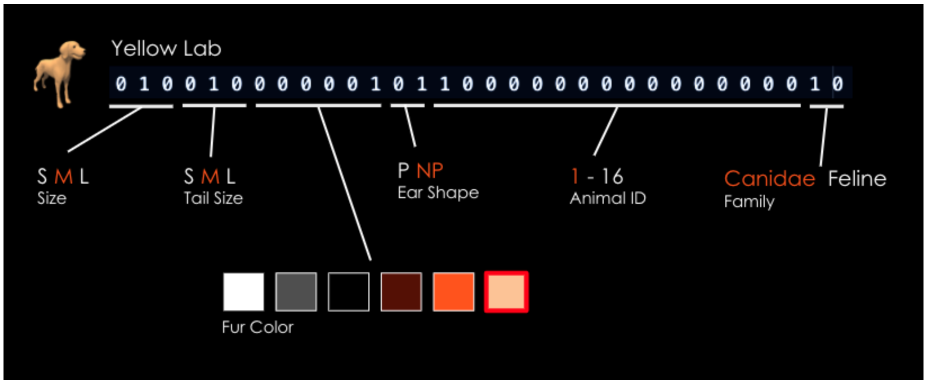


Figure 1: Example binary string syntax for input data, as formatted in the TSV file fed to the network. In this example, the string corresponds to a yellow lab, which exhibits medium size, beige fur, medium tail size, non-pointy ears, an ID of 1, and is a member of the Canidae family.

The network itself was created using the Emergent software with a Go backend, which allowed for a self-contained environment capable of training, testing, and performing the representational analyses that would be needed to explore the network's behavior. The base code that was used for the projects was originally developed to solve the family trees task, and focuses on categorical representations in networks (O'Reilly et al., 2012). It was heavily modified herein to address the specific issue of animal categorization. The network's architecture consisted of six input layers, six intermediary hidden layers, and one main hidden layer (***Figure 2***). The six input layers corresponded to the six feature categories for each animal as discussed above. Each of these input layers was of the appropriate size to accommodate for all possible input patterns. For instance, the animal ID layer consisted of neural units with dimension 4x4 to account for all 16 animal input IDs. These input layers had localist representations of the 16 different animals and each fed into their own hidden layers to provide the network with the ability to re-represent said localist representations into richer distributed patterns. The purpose of these distributed patterns was to help the model learn by emphasizing relevant feature distinctions in the input streams. The main hidden layer, of dimension 7x15, had the purpose of managing the mapping between these rich pattern representations and the output of the model.

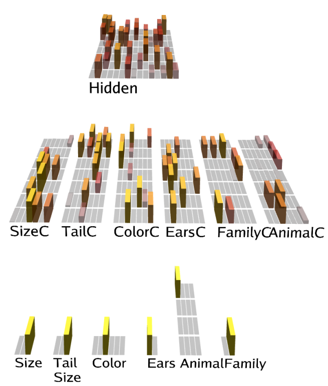


Figure 2: The base neural network used in both projects, with each layer sporting connections to other layers to encode information from inputs and generate correct outputs.

Each intermediary 7x7 hidden layer in the network (denoted as a *Code* layer by the letter C following the parameter name in ***Figure 2***) was bidirectionally connected to the main hidden layer to facilitate learning between minus and plus phases. The outputs of the network were the animal ID code and family name, such that, during training, the network would settle in the minus/expectation phase and then receive plus/outcome activation via these two layers, which were also bidirectionally connected to their respective code layers. The network was trained using a combination of error-driven learning and BCM Hebbian learning as per XCAL (eXtended Contrastive Attractor Learning Model). The efficiency of this learning rule is comparatively explored in Project 1, where the network is tested with both pure Hebbian and pure error-driven learning individually to assess the effects of each.

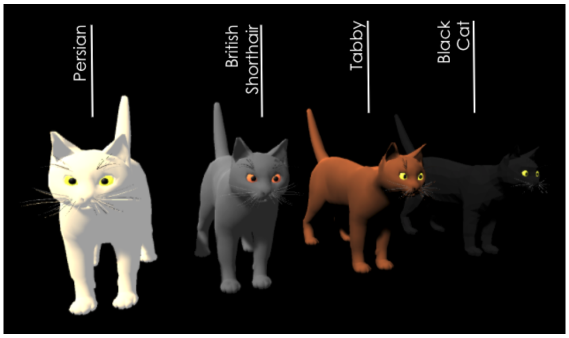
The network was tested on the same data with which it was trained, since the purpose of this exploration was not to critically assess network performance, but rather to analyze the underlying representations used by the model to complete the task. This testing environment facilitated the execution of representational analyses, which allowed for the network's hidden layer activity to be visualized via similarity matrices and cluster plots, which, in turn, provided a means by which to qualitatively assess the network's representational strategies.

PROJECT 1

Methods

The network was first trained on a selection of domestic cat and dog breeds to mirror the type of inputs typically included in datasets used to train neural networks to solve the classic cat versus dog categorization problem (Zhang et al., 2008; Parkhi et al., 2012). The Felidae family of inputs consisted of four specimens: Persian cat, British Shorthair, Tabby cat, and Black cat. The Canidae family consisted of six input breeds: Corgi, Chocolate Lab, Maltese, Yellow Lab, Rottweiler, and German Shepherd. These breeds were selected based on their status as relatively common housepets, as well as their overlapping physical features. Each of these breeds was 3D modeled and textured in the Blender 3D engine to provide visual models by which to compare the model's feature preference output (***Figures 3 & 4***).

Each of these specimens were presented to the network via a TSV file of binary strings representing their individual characteristics (***Figure 1***). This input information was then used to train the network with ten maximum runs and 100 maximum epochs per trial. By default, the



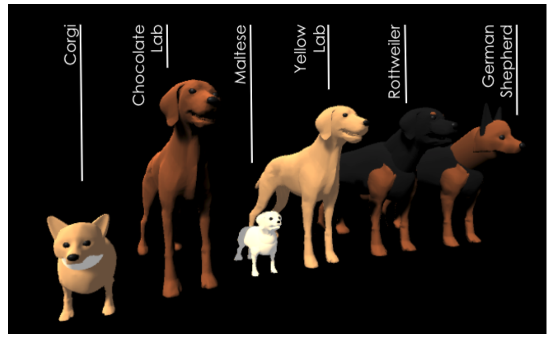


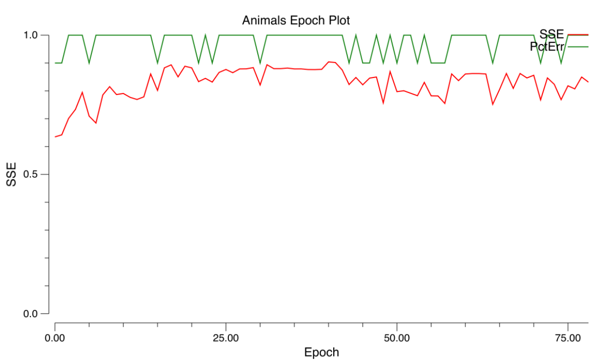
Figure 3 (Left): Stimuli for Felidae family in Project 1. Composed of the following domestic cat types: Persian, British Shorthair, Tabby, and Black cat.

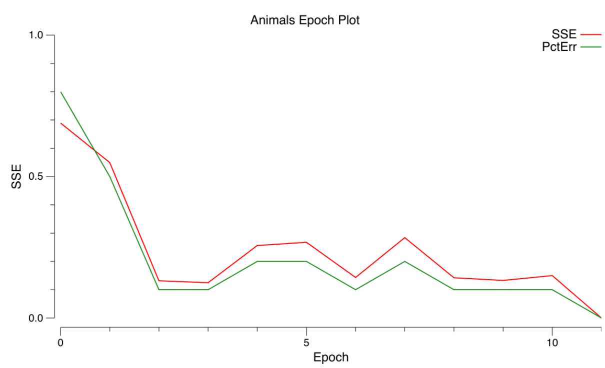
Figure 4 (Right): Stimuli for Canidae family in Project 1. Composed of the following domestic dog breeds: Corgi, Chocolate Lab, Maltese, Yellow Lab, Rottweiler, and German Shepherd.

network was trained using a combination of error-driven and BCM Hebbian learning. However, it was also trained and tested using pure error-driven and pure Hebbian learning to analyze the differences in the network's performance under these different conditions.

In the case of pure Hebbian learning, the model was never able to learn the task across runs. Because there was no explicit error-driven learning in the network, it was impossible for the network to learn how to properly represent the features in the given input data. The network had an SSE > 50 for all runs, as well as 75% - 100% error in most runs, even after training for 80 or more epochs (***Figure 5***). Therefore, these observations support the basis that Hebbian/self-organizing learning alone is insufficient to drive learning on this animal representation task. The average learning trajectory in the pure Hebbian learning trials demonstrate increasingly worse performance as learning progresses, which is caused by the "rich-get-richer" representations in the network (O'Reilly et al., 2012). Therefore, the localist focus of pure Hebbian learning, and its dynamics focused on self-organizing positive feedback, cannot drive the network to solve the task properly.

Next, the network was trained and tested using pure error-driven learning rules. The training results showed that, under these conditions, the network was now able to learn in little over 10 epochs on average, with both its SSE and percent error values reaching 0 (***Figure 6***). This data therefore suggests that error-driven learning is critical in driving the model toward learning to represent the input data in such a way that allows for correct outputs in testing.





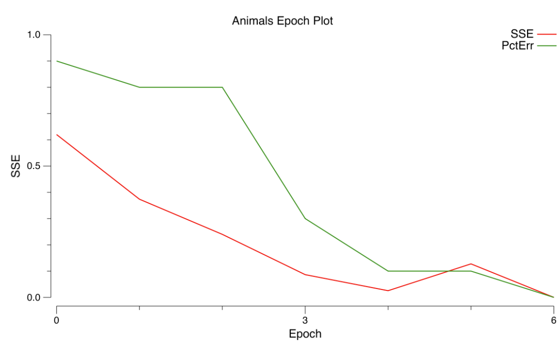


Figure 5 (Top Left): Plot for a sample epoch using pure Hebbian/Self-Organizing learning. SSE is viewable in red and Percent Error in green.

Figure 6 (Top Right): Plot for a sample epoch using pure Error-Driven learning. SSE is viewable in red and Percent Error in green.

Figure 7 (Bottom Middle): Plot for a sample epoch using the combined Hebbian/Self-Organizing and Error-Driven learning model. SSE is viewable in red and Percent Error in green.

The network was also tested using the XCAL combination of both error-driven and Hebbian learning. Under this learning rule, the network was now able to reliably learn in less than 10 epochs per trial (within the range of 5 - 10), with both an SSE and percent error of 0 (***Figure 7***). This learning was markedly faster than that of the pure Hebbian model and somewhat faster than the pure error-driven learning model. The purpose of integrating both learning types when training the network lies in the subtle benefits of Hebbian learning. In deeper, more complex networks, Hebbian learning bolsters learning by means of its characteristic self-organizing constraints, which ensure the steady drive of learning even after the majority of learning has taken place during error-driven learning. As error signals weaken during learning, Hebbian learning can provide a means for more efficiently learning subtle patterns and generalizing to new ones. Though these effects were relatively weak in this project due to the limited scope of the input patterns, they still provide a basis for using the combination of these two learning types instead of any individual one. The performance of all three learning rules were assessed in Project 1 via representational analyses.

Finally, one last manipulation was made to test the influence of the quantity of hidden layer units on pattern representation and overall task performance. The intermediary code hidden layer dimensions were changed from 7x7 to 3x3, and the dimension of the main hidden layer was changed from 7x15 to 3x15. The network was then trained using the joint Hebbian and error-driven learning model to gauge its performance with decreased hidden unit volume.

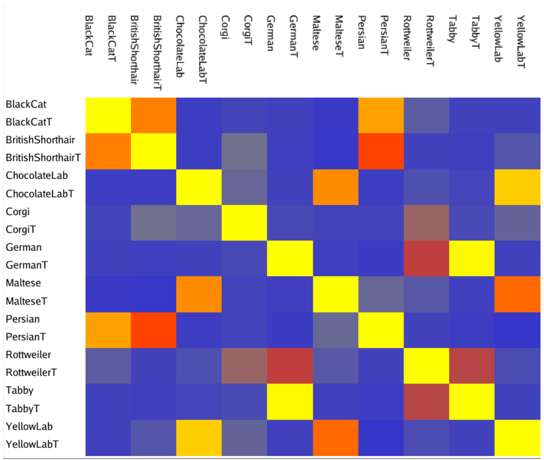
Results and Discussion

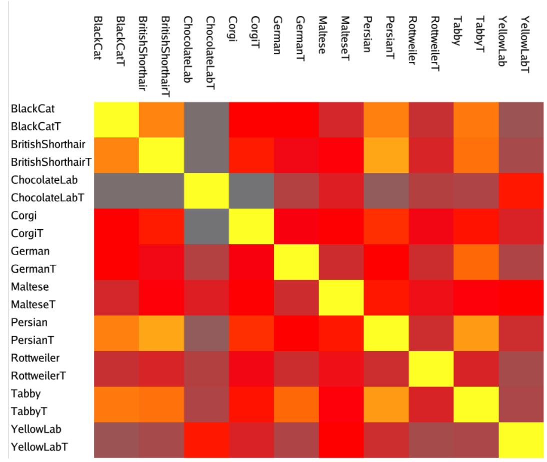
Performance of the model was assessed via representational analyses, in which similarity matrices and cluster plots were generated to visualize the activity in the hidden layer for mapping the rich pattern representations of the intermediary hidden layers and the output of the model.

Similarity matrices were used to examine relationships between different activation patterns in the main hidden layer. This allowed for the visualization of the pairwise similarities between the given input patterns using a correlation similarity measure, where 1 (yellow) corresponds to identical patterns, 0 (grey) corresponds to no relation among the patterns, and -1 (blue) corresponds to anticorrelated patterns. Therefore, in all cases, the input patterns are yellow along the main diagonal, as they are perfectly identical to themselves. Furthermore, the training and testing inputs (testing inputs are designated a T after their specimen names) show overlap in the matrices due to their identical patterns.

For the pure Hebbian runs, some similarity was represented in the hidden layer, presumably due to the inherent similarity of these features in the input patterns to begin with. However, the network was unable to represent the similarities between the remainder of the input patterns (***Figure 8***) due to the lack of error-driven learning. This resulted in the network being unable to learn across input patterns, prompting the network's poor performance in the categorization task (***Figure 5***). The "rich-get-richer" dynamics of Hebbian learning resulted in certain pattern associations becoming stronger over time and others becoming worse over time. This is reflected in worse performance as time went on during learning, and the network never being able to fully solve the task.

In the case of pure error-driven learning, the network was able to represent the similarities across input patterns, with somewhat related input patterns resulting in moderate similarity





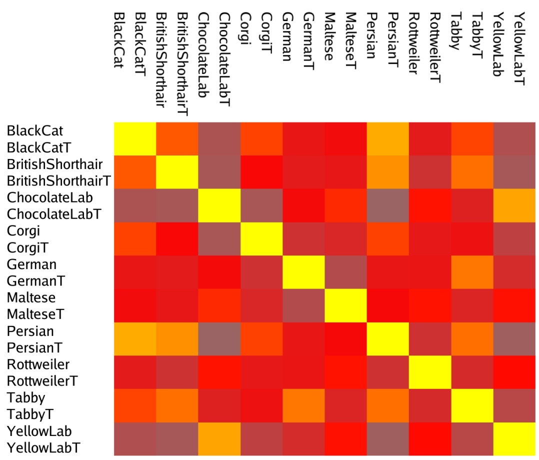


Figure 8 (Top Left): Similarity matrix for the main hidden layer with pure Hebbian/Self-Organizing learning. Yellow denotes identical patterns, grey denotes unrelated patterns, and blue represents anticorrelated patterns.

Figure 9 (Top Right): Similarity matrix for the main hidden layer with pure Error-Driven learning. Yellow denotes identical patterns, grey denotes unrelated patterns, and blue represents anticorrelated patterns.

Figure 10 (Bottom Middle): Similarity matrix for the main hidden layer using the combined Hebbian/Self-Organizing and Error-Driven learning model. Yellow denotes identical patterns, grey denotes unrelated patterns, and blue represents anticorrelated patterns.

coding in the hidden layer, as denoted by red squares (***Figure 9***). This re-coding of information meant that the network had formed sensible representations for each of the input patterns, such that it was able to use these representations to perform effectively on the categorization task.

The joint Hebbian and error-driven learning model produced results similar to those of the pure error-driven learning model. Because of the influence of Hebbian learning, which provides a steady drive on learning even after the majority of the input patterns have been represented via error-driven learning, pattern representations that were slightly richer than those in the pure error-driven condition were generated, resulting in a more inclusive similarity matrix (***Figure 10***).

These differences in similarity matrices had repercussions for the network's representat-ional cluster plots, which were also generated for each learning style. They acted as visualizations for the structure of some of the more complex patterns represented in the

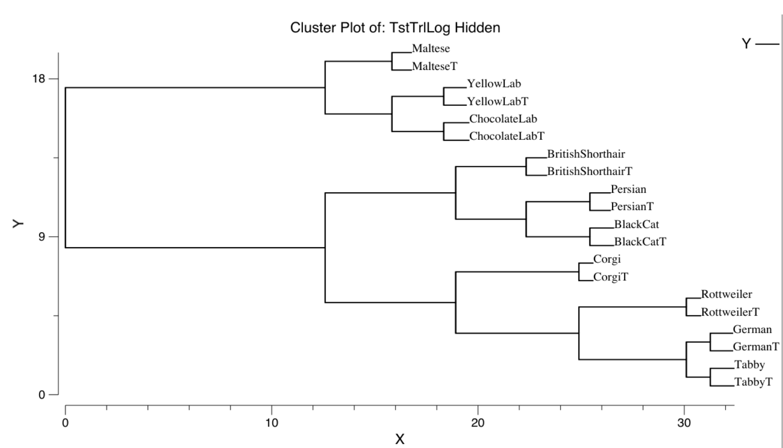


Figure 11: Cluster plot for the main hidden layer with pure Hebbian/Self-Organizing learning.

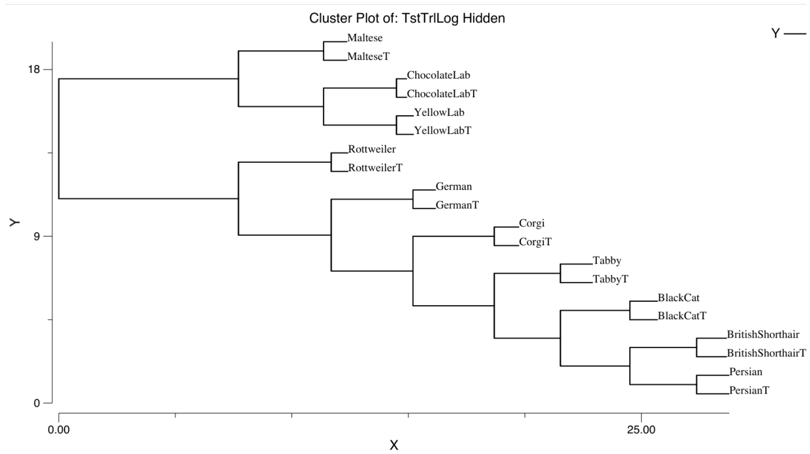


Figure 12: Cluster plot for the main hidden layer with pure Error-Driven learning.

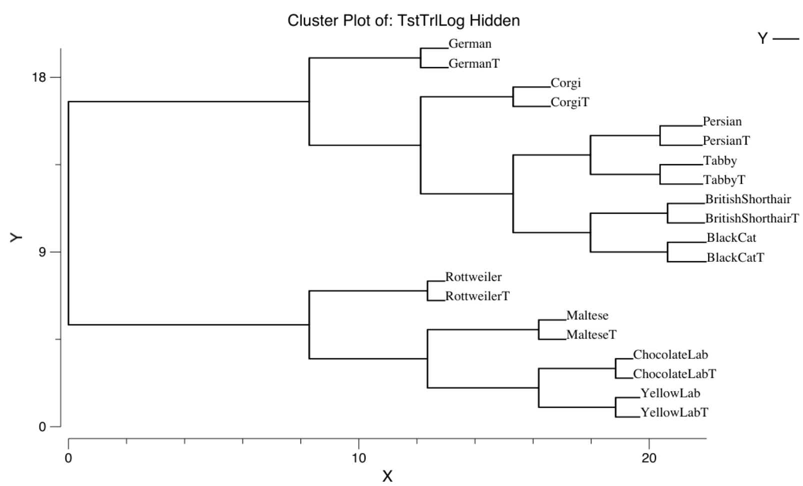
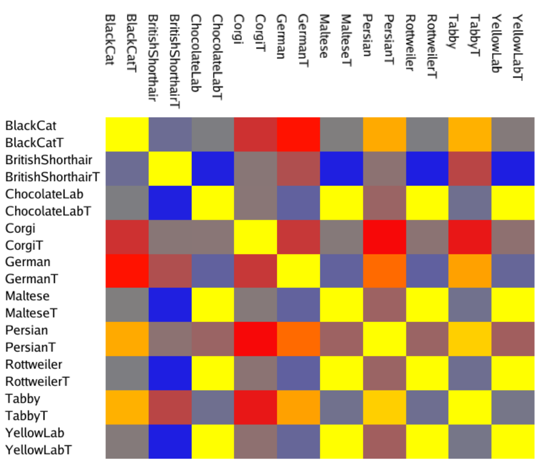


Figure 13: Cluster plot for the main hidden layer using the combined Hebbian/Self-Organizing and Error-Driven learning model.

similarity matrices. As expected, the Hebbian-only model was unable to categorize the input specimens along any metrics other than the feature similarities that it was able to parse just from the inputs themselves (which required no true learning). All other similarity representations, which required error-driven learning to properly code in the hidden layer, were not accounted for by this model. The resulting cluster plot was therefore unorganized and unsuccessful in correctly discriminating between the Canidae and Felidae input families (***Figure 11***). With pure error-driven learning, the network came closer to correctly categorizing the inputs, but committed some representational errors in classifying some of the Canidae breeds that overlapped significantly with specimens of the Felidae family (***Figure 12***). The combination of both learning strategies, however, resulted in the model being able to generate a cluster plot with more accurate categorical distinctions among the specimens (***Figure 13***). The overall plot had similar structure to that of pure error-driven learning, but also included slight differences. For instance, the four cat specimens were grouped closer together and there was cleaner discrimination for the dog breeds. These changes can be credited to the increased representational power of the network with Hebbian learning in addition to error-driven learning, as subtle corrections in the hidden layer representations can be executed after the bulk of error-driven learning has occurred.

These subtle differences thus resulted in the manifestation of more humanlike categorical strategies in the cluster plot. For instance, the Rottweiler was grouped with the other Canidae members due to the shape of its ears. All of the representations for the Felidae specimens were grouped closer to one another to reflect the sizable overlap among their features. The network seemed to use ear shape as the main differentiating feature across specimens, as all dogs with non-pointy ears were grouped together, but dogs with pointy ears (the German Shepherd and Corgi) were grouped closer to the inputs corresponding to cats. However, these dogs were still more proximal to the other dogs in the cluster plot than the cat specimens, suggesting that tail length and animal size were also important features used by the network to discriminate between members of either family. The Corgi's closer proximity to the cat breeds over that of the German Shepherd is illustrative of this observation (***Figure 13***). It was also apparent that fur color was deemed unimportant in categorizing specimens into either family, as there was a great deal of overlap in this input parameter between the two families. Because of this, inputs that exhibited low fur color similarity but high similarity in other features were grouped closely together.

One final exploration executed in Project 1 was that of measuring network performance using hidden layers of decreased size. The intermediary code hidden layers were altered from dimension 7x7 to 3x3, and the main hidden layer in charge of managing pattern representation was changed from dimension 7x15 to 3x5. Changing the concentration of units in the hidden layers resulted in excessive overlap between representations in these layers, as the similarity matrix for the network under these conditions showed high correlation between specimens that otherwise should have had lower pairwise similarities (***Figure 14***). The



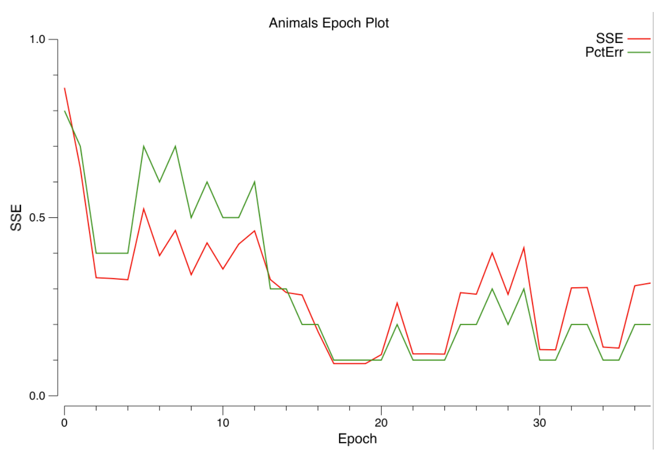


Figure 14 (Left): Similarity matrix for the smaller main hidden layer (3x5) with joint Hebbian and Error-Driven learning. Yellow denotes identical patterns, grey denotes unrelated patterns, and blue represents anticorrelated patterns.

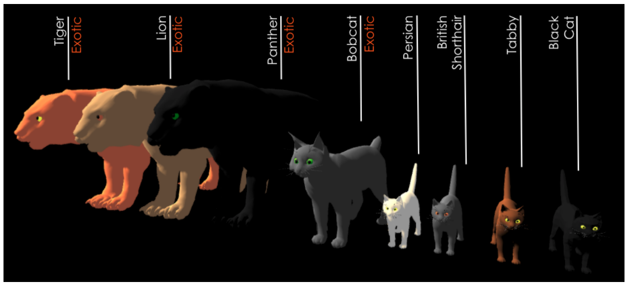
Figure 15 (Right): Plot for a sample epoch using joint Hebbian and Error-Driven learning. SSE is viewable in red and Percent Error in green.

decrease in hidden units available for encoding stored representations caused the network to struggle during learning, despite using the joint Hebbian and error-driven learning model that performed effectively with higher hidden layer sizes earlier in Project 1. On average, even after over 35 epochs, the network struggled to attain SSE and percent error values of 0 on the categorization task (***Figure 15***). These data demonstrated that the sizes of the hidden layers were pivotal to the performance of the network, since the units in these layers were directly used to re-represent localist pattern representations to facilitate learning.

PROJECT 2

Methods

Project 2 acted as an expansion of the joint Hebbian and error-driven learning trials from Project 1, with the default hidden layer sizes of 7x7 for intermediary code layers and 7x15 for the main hidden layer being used, as these parameters were determined to be optimal from the results of Project 1. In Project 2, however, the network was trained on a selection of members of the Felidae and Canidae families that included both domestic and exotic species, allowing for increased variation in the input patterns as well as providing an opportunity to observe any alterations in the network's learning strategies within a novel training environment. The Felidae family was now made up of the following specimens: Persian cat, British Shorthair, Tabby cat, Black cat, Tiger, Lion, Panther, and Bobcat (***Figure 16***). The Canidae family consisted of the same breeds as Project 1, with the addition of two exotic specimens: Corgi, Chocolate Lab, Maltese, Yellow Lab, Rottweiler, German Shepherd, Fox, and Wolf (***Figure 17***).



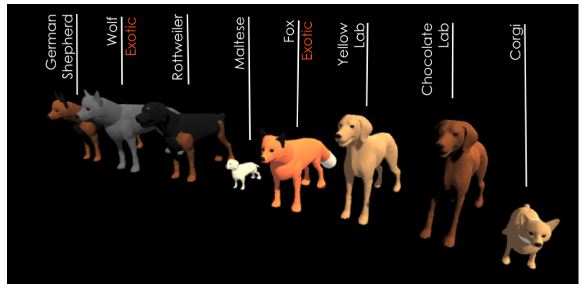


Figure 16 (Left): Stimuli for Felidae family in experiment 1. Composed of the following domestic cat types: Persian, British Shorthair, Tabby, and Black cat.

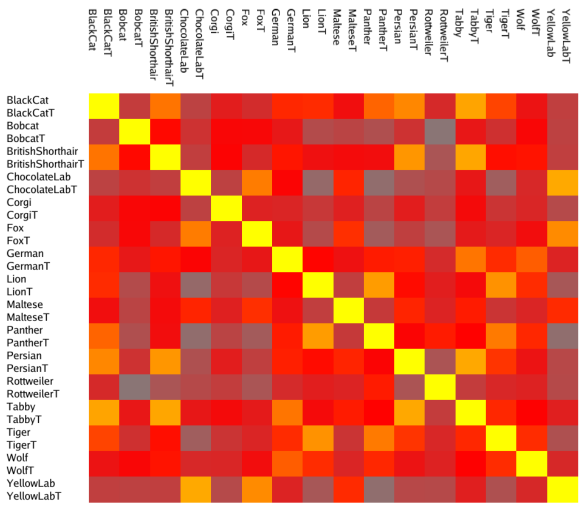
Figure 17 (Right): Stimuli for Canidae family in experiment 1. Composed of the following domestic dog breeds: Corgi, Chocolate Lab, Maltese, Yellow Lab, Rottweiler, and German Shepherd.

Training and testing were identical to the joint Hebbian and error-driven learning trials of Project 1, with the addition of more input patterns to accommodate the novel stimuli. Performance of the model on the categorization task was assessed using the same representational analyses as in Project 1 (similarity matrices and cluster plots).

Results and Discussion

As in Project 1, performance of the model was assessed via representational analyses. With the six novel specimen input patterns, the network had to solve the same categorization problem and learn to form representations with an even wider array of inputs than previously. The network was able to learn to correctly identify the input in around 10 - 15 epochs per trial (***Figure 19***). This learning was about 10 epochs slower than for the same learning model in Project 1, due to the 60% increase in the input size. Unlike the pure Hebbian and smaller-hidden-layer trials from Project 1, the network was able to consistently reach 0 SSE and percent error values, suggesting an effective representational strategy was taking place in the network.

The similarity matrix of the network was reminiscent of the matrix produced by the network in Project 1 under the same learning rule (***Figure 18***). It suggested that the hidden layer was able to recognize similarities between input patterns to a sensible degree. This was



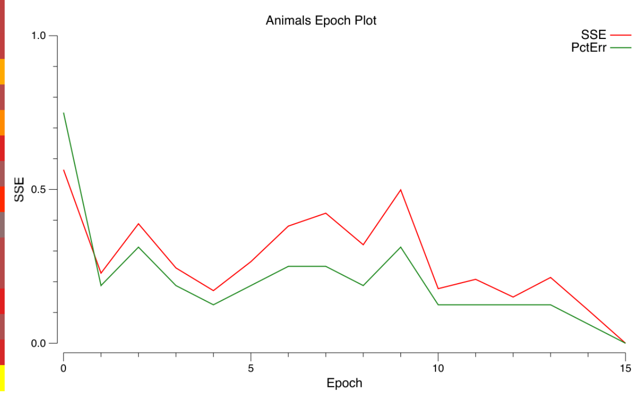


Figure 18 (Left): Similarity matrix for the smaller main hidden layer (3x5) with joint Hebbian and Error-Driven learning. Yellow denotes identical patterns, grey denotes unrelated patterns, and blue represents anticorrelated patterns.

Figure 19 (Right): Plot for a sample epoch using joint Hebbian and Error-Driven learning with added exotic input patterns. SSE is viewable in red and Percent Error in green.

supported by the cluster plot generated by the representational analysis, which showed sensible categorization decisions in terms of the network's hierarchy for prioritizing features to determine a given animal's family (***Figure 20***). The cluster plot carried some structural similarity with that of the one produced from the same learning rule in Project 1, as the majority of Canidae specimens were grouped on one side of the plot, and the Felidae specimens were grouped in their own subtree in the plot. However, because of the novel feature information provided by the additional animal specimen inputs, the cluster plot appeared to show that the network used new rules for category representation.

This time around, it seemed that tail length was of much more importance to the network than it had been previously, as it was used in conjunction with the ear shape parameter to do the majority of the sorting in the cluster plot. This can be observed in a number of features of the cluster plot. The Bobcat and Corgi were grouped together between the clusters for Canidae and Felidae, since they both had short tails and pointy ears. Though the Rottweiler also had a short tail, it was grouped closer to the other members of the Canidae family due to its non-pointy ear shape. In addition, the Fox was grouped well within the Canidae cluster despite its pointy ear shape, since it shared medium tail length with many of the dog breeds therein. It is also notable that the Wolf and German Shepherd were grouped together, but closer to the large, exotic Felidae specimens. When considering the prioritization of ear

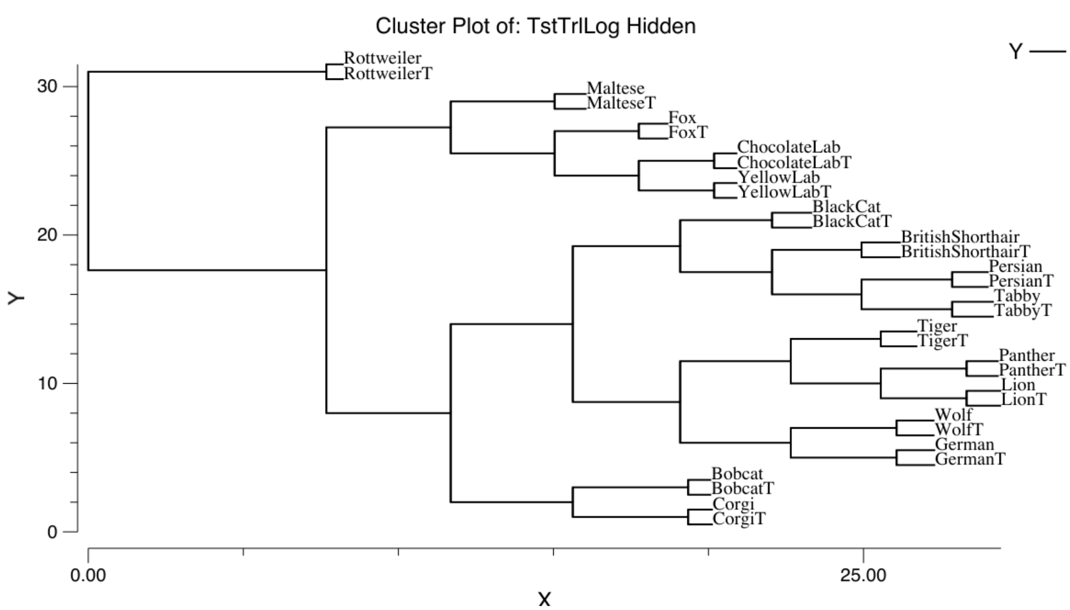


Figure 20: Cluster plot for the main hidden layer using the combined Hebbian/Self-Organizing and Error-Driven learning model with both domestic and exotic animal input patterns.

shape and tail length in representing specimen categories, it makes sense that they would be grouped in such a way, due to their pointy ears and long tails, which are traits shared by the large cats. They also shared the overall large size trait, suggesting that their features most closely aligned to the network's representation of large felines and that size is a feature beyond ear shape and tail size that the network also uses to categorize animals. As in Project 1, the cluster plot showed no evidence suggesting that color was of importance to the network.

General Discussion

Contemporary neural networks that are designed for discriminating between cats and dogs do so via the identification of structural features, similarly to the strategy employed by human infants faced with the same task (Quinn et al., 2001). This is possible due the presence of hidden layers that allow the network to re-code complex patterns to form representations allowing it to properly identify members of either animal family. The data produced in these projects, while rather crude, provides some functional insight into this underlying process.

Overall, it seems that the neural network used in this study made use of more structure- and shape-based features for categorizing animals, rather than using color or texture information. This allowed the network to categorize the majority of inputs correctly, forming representations for families of animals based on the most common identifiers for each one. The network learned to prioritize features based on those it "saw" in the environment, using underlying trends in the data to find the optimal way of representing Canidae and Felidae. For instance, although there were small dogs present in the input patterns, the model also learned that dogs could be of medium or large sizes, and therefore gravitated towards intermediary medium sizes for representing dogs. Although cats could be any color, and were typically small if domestic and large if exotic, the network learned to give more weight to the features of ear shape and tail length in determining what inputs were characteristically feline.

Because the network's learned representations were focused on structural similarity between animals, the general "mental prototype" that the network holds in memory when discriminating between cats and dogs can be modeled visually (***Figure 21***). Though these representations are biased by the limited initial input patterns given to the network, they allow for the qualitative evaluation of how a network, and humans, can come to model such mental prototypes. This network learned to represent dogs as having medium size, medium tails, and non-pointy ears, whereas cats could be either small or large, with pointy ears, and long tails. In both cases, the network learned to ignore color information, so the colors of these prototypes are trivial to the network.

This experimental exploration into the representational strategies of neural networks faced with categorization tasks provides initial insight as to how such processes are executed. It not only shows that phenomena of human cognition can be modeled using computer networks, but that the mental shortcuts and strategies used by us may be used by computers too.

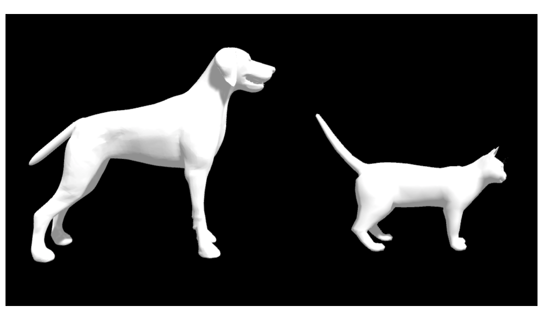


Figure 21: 3D models generated from the network's prioritized features for representing members of the Canidae and Felidae families.

References

Fleuret, F. & Geman, D. (2008). Stationary Features and Cat Detection. Journal of Machine Learning Research (JMLR). 9.

Hayward, W. G. (1998). Effects of outline shape in object recognition. Journal of Experimental Psychology: Human Perception and Performance, 24, 427–440.

Kurbat, M. A., Smith, E. E., & Medin, D. L. (1994). Categorization, typicality, and shape similarity. In Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Erlbaum.

Landau, B., Smith, L. B., & Jones, S. S. (1988). The importance of shape in early lexical learning. Cognitive Development, 3, 299–321.

Malt, B. C., & Smith, E. E. (1984). Correlated properties in natural categories. Journal of Verbal Learning & Verbal Behavior, 23, 250–269.

Murphy, G. L., & Smith, E. E. (1982). Basic-level superiority in picture categorization. Journal of Verbal Learning and Verbal Behavior, 21, 1–20.

O'Reilly, R. C., Munakata, Y., Frank, M. J., Hazy, T. E., and Contributors (2012). Computational Cognitive Neuroscience. Wiki Book, 4th Edition (2020). URL: https://github.com/CompCogNeuro/ed4

O'Reilly, R. C., Wyatte, D., Herd, S., Mingus, B., & Jilk, D. J. (2013). Recurrent processing during object recognition. Frontiers in Psychology, 4, 124. 10.3389/fpsyg.2013.00124

Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V. (2012). Cats and dogs. Proceedings /

CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 3498-3505. 10.1109/CVPR.2012.6248092.

Quinn, P. C., Eimas, P. D. (1996a). Perceptual cues that permit categorical differentiation of animal species by infants. Journal of Experimental Child Psychology, 63, 189–211.

Quinn, P. C., Eimas, P. D., Tarr, M.J. (2001). Perceptual categorization of cat and dog silhouettes by 3- to 4-month-old-infants. Journal of Experimental Child Psychology 79, 78–94.

Quinn, P. C., Eimas, P. D., & Rosenkrantz, S. L. (1993). Evidence for representations of perceptually similar natural categories by 3-month-old and 4-month-old infants. Perception, 22, 463–475.

Rakison, D. H., & Butterworth, G. E. (1998). Infants' use of object parts in early categorization. Developmental Psychology, 34, 49–62.

Smith, J. D., Zakrzewski, A. C., Johnson, J. M., Valleau, J. C., & Church, B. A. (2016). Categorization: The View from Animal Cognition. *Behavioral sciences (Basel, Switzerland)*, *6*(2), 12. https://doi.org/10.3390/bs6020012

Tversky, B., & Hemenway, K. (1984). Objects, parts, and categories. Journal of Experimental Psychology: General, 113, 169–193.

Ullman S. (1996). High-level vision: Object recognition and visual cognition. Cambridge, MA: MIT Press.

Zhang, W., Sun, J., & Tang, X. (2008). Cat Head Detection - How to Effectively Exploit Shape and Texture Features. 5305. 802-816.10.1007/978-3-540-88693-8_59.