A Convolutional Neural Network Approach to Semantic Cognition

Benjamin Berkman
Center for Data Science
New York University
bjb433@nyu.edu

Atharv Bhat
Courant Institute of Mathematical Sciences
New York University
arb881@nyu.edu

Stephen Spivack
Center for Data Science
New York University
ss7726@nyu.edu

Abstract

In this paper, we report on a series of experiments based on the neural network model of semantic cognition by Rogers and McClelland. We extend this framework to the domain of images and use a convolutional neural network to extract features that are passed to a modified version of the original model. We also test how well our model performs on images that are perturbed with random Gaussian noise and observe the development patterns. We find that when we test our model on natural images in which no noise is added, we are able to recover the hierarchical clustering as observed in Rogers and McClelland. However, when we test our model on images that are perturbed with random Gaussian noise, we are unable to fully recover that same hierarchical classification structure.

Keywords: cognitive science; computer vision; deep learning; learning and memory

1. Introduction

One of the fundamental questions at the intersection of cognitive science and machine learning is how we can train computers to behave intelligently like humans. One such approach with a deep, rich intellectual history that dates back to the mid-twentieth century is connectionism. This approach posits the mind as a parallel, distributed information-processing machine where activity is spread across individual artificial neurons whose connections are modified by learned experience. With a tradition rooted in the learning theories of Hebb [4] and the perceptron models of Rosenblatt [12], this approach has several key advantages pertaining to how mind and brain actually behave at a biological level.

In this paper, we will expand the semantic cognition models of Rogers and McClelland [8], which was adapted from the pioneering work of Rumelhart and Todd [13]. Semantic cognition is a key component of executive function and is central to object recognition, language, and problem solving [11]. This uniquely human ability allows us to apply our knowledge about the structure of the world to execute behavior. Critically, semantic cognition relies on frontal and temporoparietal cortical networks in the brain [10], and damage to these networks have been shown to impair performance in neuropsychological tests of patients with semantic dementia [1].

Rogers and McClelland first modeled semantic cognition using a multi-layer neural network. Seeking to answer, "How do we know what properties something has, and which of its properties should be generalized to other objects", the authors applied gradient descent via backpropagation to map plants and animals to their respective semantic properties (e.g., "green" and "tall"). This contrasted with more traditional symbolic approaches. Rogers and McClelland also applied noise to each concept to "resemble the loss of neurons in semantic dementia."

The overall goal of this paper is to report our findings from a series of experiments in which we train a convolutional neural network on images belonging to the object classes of Rogers and McClelland and see how well this model performs when this same set of images is perturbed with random Gaussian noise.

2. Related work

2.1. Computer vision and deep learning

Computer vision is a multidisciplinary field concerned with how machines extract meaningful information from digital images or video [2]. One such important area of active research is object recognition and image classification. The basic idea here is that a supervised learning model is inputted with an image of an object and produces an output corresponding to a label for that object. The applications of this are widespread and exist within many of today's technologies, including video surveillance, task automation, and even self-driving cars [15]. With the recent advent of graphics processing units (GPUs) and increasing interest in deep learning, the field of computer vision has seen a revolution dominated by neural network models, specifically convolutional neural network architectures.

2.2. Convolutional neural networks

Convolutional neural networks (CNNs) are an architecture of neural networks well-suited for object recognition and image classification. A convolution layer applies a filter to an image to create various feature maps that embed contextual information (e.g., "does this image contain leaves?") while

a pooling layer reduces the positional dependency of each feature. The CNN architecture can be traced back to LeCun et al. (1998) [6] and is generally surveyed in LeCun at al. (2015) [5]. In our series of experiments, we use a pre-trained 18-layer Residual Network (ResNet) which uses an identity shortcut to bypass the vanishing gradient problem [3] from which deep neural networks often suffer.

3. Methods

3.1. Data

To obtain the data for our experiments, we used Fatkun Batch Download Image – a useful image batch download extension freely available through Google – to extract 100 images for each of the eight classes in Rogers and McClelland. Each image class – canary, daisy, oak, pine, robin, rose, salmon and sunfish – comes from the domain of living things. For image classes with multiple colors, we partitioned by color into two subsets of equal size so that each class contains 50 images of one color, and 50 images of another color. For example, canaries can be either red or yellow, so we downloaded 50 images for each color. Pines are only green, so we downloaded 100 images of green pines. See Figure 1 for an example of four stimuli used in our experiments.

Figure 1. Example images used in our experiments. Top left panel is red rose. Top right panel is green pine. Bottom left panel is yellow canary. Bottom right panel is gray salmon.

For preprocessing we serialized the data using pickle. This formats the data into a byte stream, where each data point can be represented as a dictionary containing the matrix representation of the image (expressed as height by width by channels), the item of the image (e.g., "robin"), its color (e.g., "red"), its relation (e.g., "can") represented as a one-hot vector, and its attribute (e.g., "grow) also represented as a one-hot vector. For each item and color, we assign five images randomly to a validation set, and the rest to a training set. This returns approximately 1,200 images to the training set and 70 images to the validation set. From this pickle object, we create a PyTorch Dataset which applies transformations such as converting each image to a 224 by 224 tensor. We form Python DataLoaders for the training and validation data sets with 24 samples per batch.

3.2. Models

Consistent with Rogers and McClelland, we implemented a CNN with hidden and representation units. While Rogers and McClelland used eight hidden units in the representation layer and 15 hidden units, we trained a model with 32 hidden units in the representation layer and 128 hidden units. We used the pretrained 18-layer ResNet [3] as a feature extractor instead of the one-hot encoded inputs from Rogers and McClelland. See Figure 2 for a schematic of the model used in our experiments. In the forward pass, we use the ReLu activation function for both the representation and hidden layers, and the sigmoid activation function for the output layer. We trained the model for 20 epochs optimizing against binary cross entropy as the loss function, which can be defined as:

$$-\sum_{c=1}^{M} y_{i,c} \log(p_{i,c})$$

where M represents the number of classes, y is a binary indicator of correct classification into class c for observation i, and p is the predicted probability of i in c.

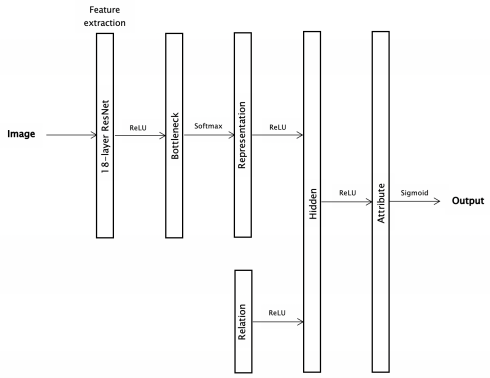

Figure 2. CNN Architecture. Image is inputted to 18-layer ResNet extractor before being passed to the model from Rogers and McClelland (2003).

3.3. Experiments

In our first experiment we trained and tested our CNN for object recognition on the set of images belonging to the classes described above. In our second experiment we perturbed those same images with random Gaussian noise using the procedure described above and observed the development patterns by visualizing each epoch using a dendrogram.

To generate noisy images for our second set of experiments, for each image and each pixel, we randomly sampled from a Gaussian distribution across four ranges of values for the standard deviation, and added that sample to the original pixel value. All image values were clipped within a [0,1] range. This resulted in four sets of images with σ = [0.0, 0.05, 0.1, 0.2], where $\mu = 0$ for all sets. See Figure 3 for an

example of this procedure applied to a single image from our dataset.

Figure 3. Example image of a red robin. Top left panel is original image. Top right panel has random Gaussian noise sampled from σ = 0.05. Bottom left panel has random Gaussian noise sampled from σ = 0.1. Bottom right panel has random Gaussian noise sampled from σ = 0.2.

4. Results

We first evaluated the performance of our CNN on classifying images. Training the the network over 20 epochs and evaluating the training and validation loss at each epoch, we observed decreasing training and out-of-sample loss at each progressive epoch. The binary cross entropy loss decreases sharply for the first two epochs, then levels out, decreasing at a steady rate for the remaining 18 epochs. After 20 epochs, we observed a training loss of 0.081 and a validation loss of 0.074. Validation loss is consistently lower than training loss, suggesting we did not overfit the data. Figure 4 displays the learning curve of the data over the 20 epochs.

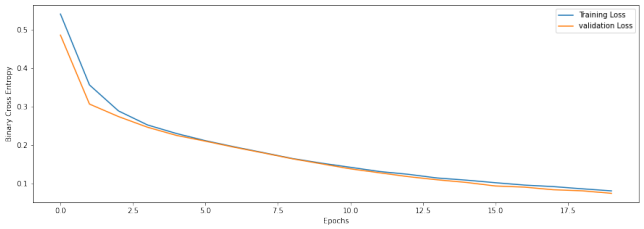

Figure 4. CNN learning curve for training and validation sets over 20 epochs.

We then extracted the activation pattern of the representation layer for each item and visualized these patterns. We did this for the natural images (no added noise) as well as the noisy images with σ = [0.0, 0.05, 0.1, 0.2]. We display the representations of each item after epoch 4, 8, 12, 16, and 20 for σ = 0.0 (no noise) and σ = 0.2 (most noise) in Figure 5 and Figure 6. In both cases, we observe that the representations move from broad to specific as the epochs progress. In the no noise environment, there is a bit more pronounced shift from broad to specific representation than in the noisy environment. This is noticeably visible in red canaries, for example. Without noise, the network learned strong representations after 20 epochs. With noise, the representations are as uniform after 20 epochs as they are without noise after only four epochs.

Next, we studied the hierarchical clustering of the items through dendrogram plots after epoch 10, 15, and 20. These plots provide insight into items the model identifies as most similar to each other. Much like the representation plots, we studied how these plots differ as we vary noise over σ = [0.0, 0.05, 0.1, 0.2]. Figure 7 displays the hierarchy with σ = 0 (no noise). After the first 10 epochs, fish (sunfish, gray salmon, red salmon) are clustered together and separated from a cluster of flowers and trees, while birds (yellow robin, yellow canary, red canary, and red robin) formed their own cluster. At this point, the model still made some errors, such as not classifying the yellow rose with the other flowers. After 20 epochs, however, the model formed nearly perfect hierarchies. For example, yellow robins and yellow canaries are grouped together at the base cluster, then merged with red robins and red canaries to form a bird cluster. This suggests that colors are more relevant than item – the model is more likely to group two birds of different type (robin and canary) but of the same color, than two birds of the same type but different color.

Figure 8 displays the dendrogram for noisy images with σ = 0.2. The model took longer to learn in this case. After 10 epochs, the model was unable to discern strong hierarchies as nearly all items appear roughly equidistant from each other. After 20 epochs, the model is able to create a hierarchy, but it is not as accurate as the items without noise. For example, animals and plants formed two distinct clusters in the non-noisy environment. With noise, we observe a cluster composed of plants (oaks and daisies), but the other cluster contains animals (birds and fishes) as well as additional plants (roses and daisies). The model also clustered gray salmons as more similar to red canaries and red robins than a red salmon or sunfish. This suggests noise does affect the ability of the model to form accurate hierarchies.

5. Discussion

In this paper, we extended the Rogers and McClelland neural network model of semantic cognition to a CNN model trained on a set of images belonging to the same object classes. Broadly, we found that for natural images in which no noise was added, our model was able to recover the same hierarchical classification structure reported in Rogers and McClelland. However, when we perturbed the image set using random Gaussian noise, our model failed to recover this hierarchical structure.

When our image set is perturbed with random Gaussian noise, the classification behavior of our CNN model seems

Figure 5. Representations for images with $\sigma = 0.0$.

to capture some of the neuropsychological deficits seen in patients with semantic dementia [8]. At a biological level, neurodegenerative disorders result from the loss of CA3 hippocampal cells [9], which are critically involved for both the encoding and retrieval of semantic memory from distributed sensory cortical representations [14]. Moreover, the hippocampus is strongly connected to the frontal and temporoparietal networks [16]. Perturbations in this system have been shown to lead to the behavioral deficits observed in patients with semantic dementia [11]. Therefore, it seems that our model is able to recapitulate these biological findings at both a computational and behavioral level. As Rogers and McClelland wrote, "increasing degrees of perturbation degrade the network's ability, first to activate specific information about the item (specific name, object-specific properties) and later to activate more general properties, recapitulating the pattern of progressive deterioration of conceptual knowledge seen in semantic dementia" [8]. We see this through our experiments: as we increased noise to the images, the hierarchical structures appeared flatter, suggesting an inability to identify object-specific properties. Further, at the end of training, noisy images still make incorrect connections (e.g., grouping gray salmon with canaries before other salmon), a phenomenon Rogers and McClelland describe as well [8].

Figure 6. Representations for images with $\sigma = 0.2$.

Future research could include altering the way in which we add noise to the images. In our experiments, we randomly sampled from a Gaussian distribution with varying levels for the standard devitation, where each sample is added to each pixel. Instead, we could apply a Gaussian filter with a specific kernel size and standard deviation to blur the images instead of adding noise to them, which would allow us to observe if the trends reported in this paper replicate to other image preprocessing methods.

Another line of future research would be to run behavioral experiments and record data from healthy controls as well as patients with semantic dementia during a visual classification task. As we did not record any behavioral data in this paper we have no way to compare how well our model would perform compared to the behavioral data from both groups of participants. Therefore, having access to such data would hopefully allow us to close the explanatory gap between the behavior of our CNN model and the cognitive and behavioral deficits observed in semantic dementia. We believe that having data across all three levels of Marr's description of information processing systems [7] would allow us to draw stronger conclusions regarding the performance of our model and how it relates to human cognition as a whole.

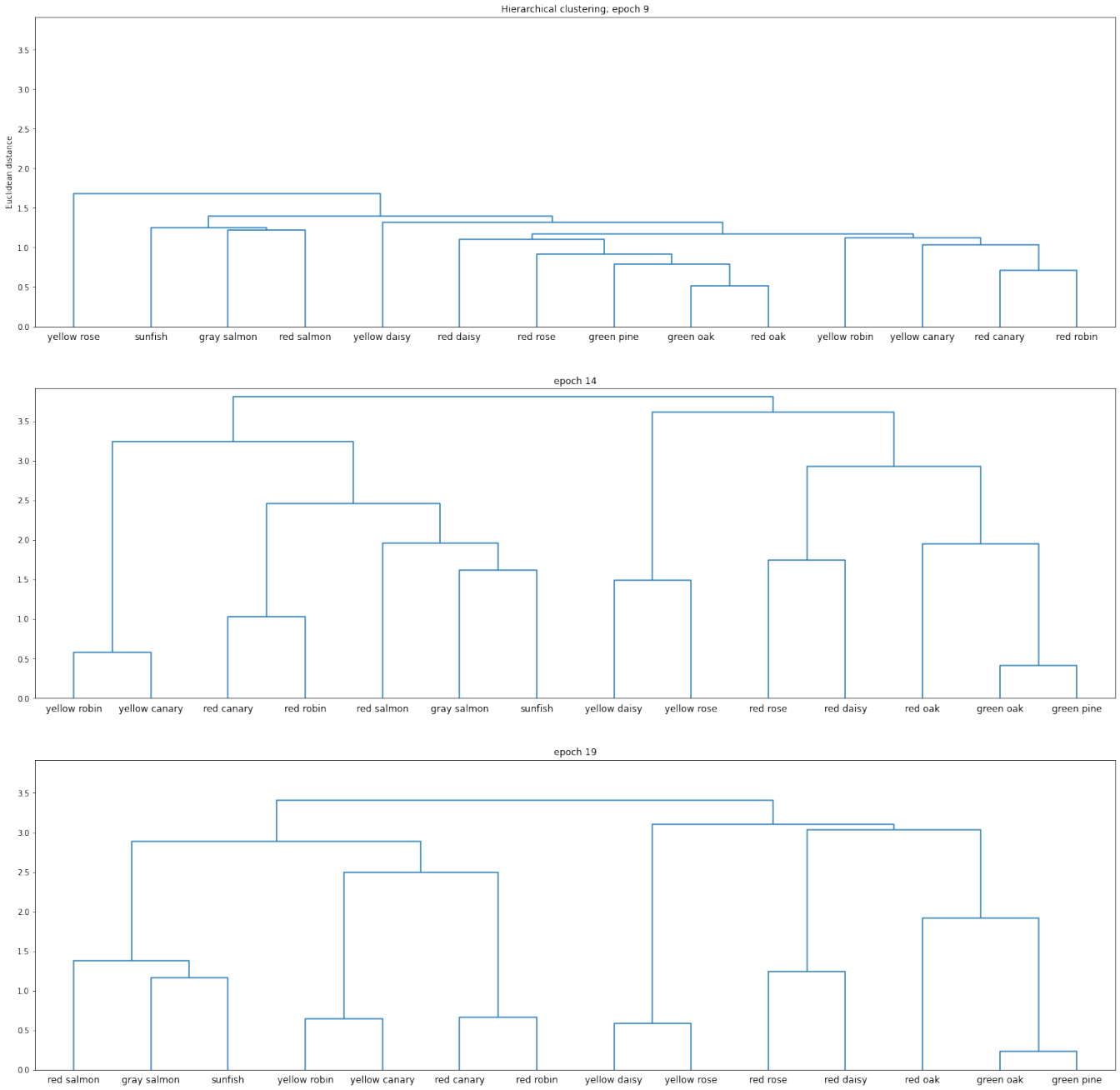

Figure 7. Dendrogram with $\sigma = 0.0$.

References

[1] Sasha Bozeat, Matthew A Lambon Ralph, Karalyn Patterson, Peter Garrard, and John R Hodges. Non-verbal semantic impairment in semantic dementia. *Neuropsychologia*, 38(9):1207–1215, 2000. 1

[2] David Forsyth and Jean Ponce. *Computer vision: A modern approach.* Prentice hall, 2011. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 2

[4] Donald O. Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, June 1949. 1

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. 2

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998. 2

[7] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010. 5

[8] James L. McClelland and Timothy T. Rogers. Nature reviews neuroscience. *The parallel distributed processing approach to semantic cognition*, 4, Apr 2003. 1, 4

[9] Manuela Padurariu, Alin Ciobica, Ioannis Mavroudis, Dim-

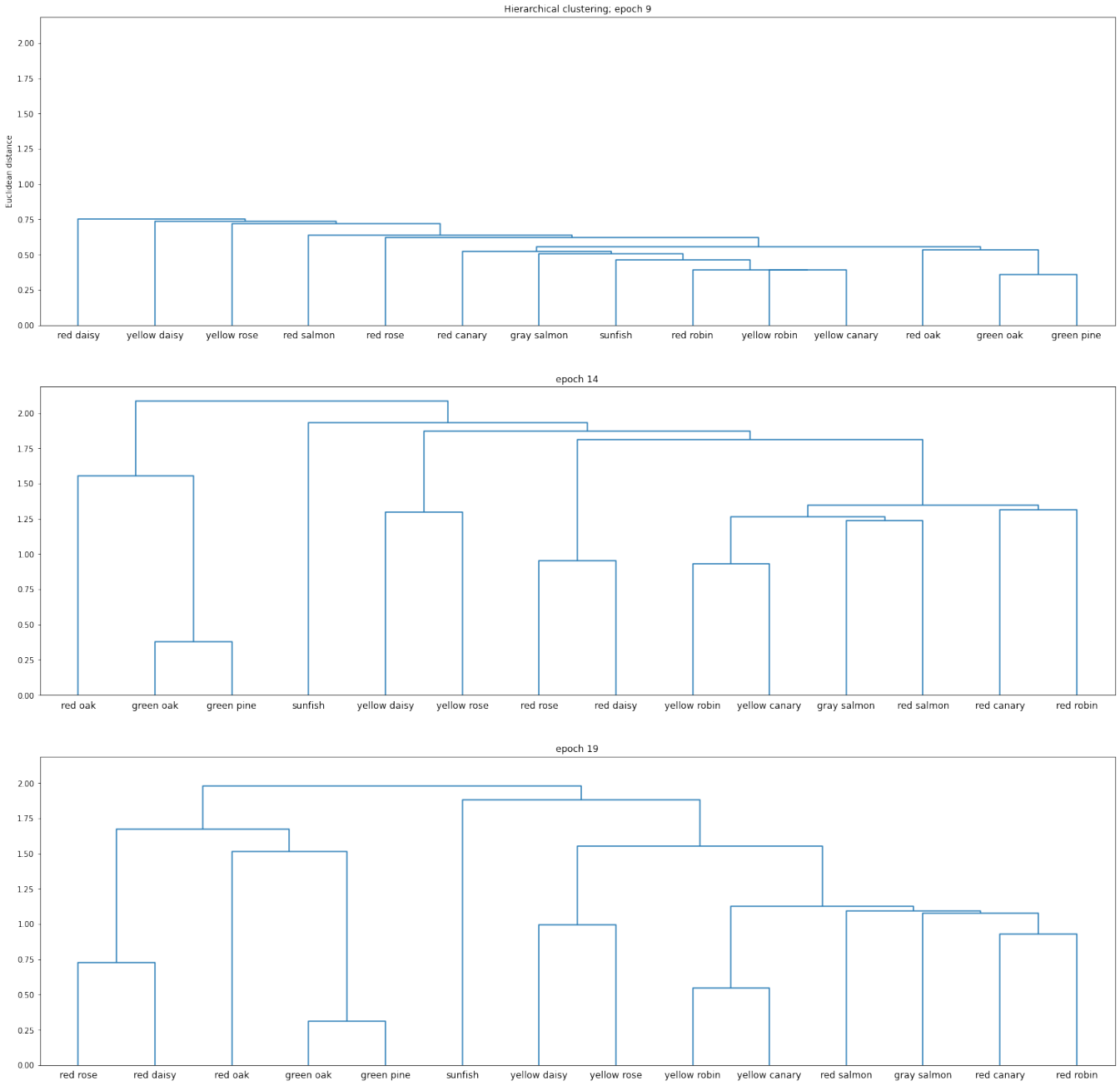

Figure 8. Dendrogram with $\sigma = 0.2$.

itrios Fotiou, and Stavros Baloyannis. Hippocampal neuronal loss in the ca1 and ca3 areas of alzheimer's disease patients. *Psychiatria Danubina*, 24(2.):152–158, 2012. 4

[10] Matthew A Lambon Ralph, Elizabeth Jefferies, Karalyn Patterson, and Timothy T Rogers. The neural and computational bases of semantic cognition. *Nature Reviews Neuroscience*, 18(1):42–55, 2017. 1

[11] Timothy T Rogers, James L McClelland, et al. *Semantic cognition: A parallel distributed processing approach*. MIT press, 2004. 1, 4

[12] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958. 1

[13] David E. Rumelhart and Peter M. Todd. Learning and connectionist representations. In David E. Meyer and Sylvan Kornblum, editors, *Attention and Performance Xiv*, pages 3–30. MIT Press, 1993. 1

[14] Larry R Squire and Barbara J Knowlton. Memory, hippocampus, and brain systems. 1995. 4

[15] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 1

[16] Andrew M Ward, Aaron P Schultz, Willem Huijbers, Koene RA Van Dijk, Trey Hedden, and Reisa A Sperling. The parahippocampal gyrus links the default-mode cortical network with the medial temporal lobe memory system. *Human brain mapping*, 35(3):1061–1073, 2014. 4