Chunking in Neural Networks

Simin Li
Advisor: Tai-Sing Lee
Carnegie Mellon University School of Computer Science

Abstract

Object recognition has been increasingly used in real world applications. One way to advance object detection is through changing how objects are represented in neural networks [6]. **Chunking**, or organizing information into meaningful chunks, makes it easier for humans to encode and retain [7][8]. However, chunking in neural networks has not been explored extensively.

- **Does chunking occur in neural networks? What does it look like?**
- **What effect does training with hard-to-discriminate images in a search task have on the formation of chunks?**

To answer these questions, we trained a modified version of Resnet18 on a simple search task with Chinese characters and found what we believe to be evidence of chunking in neural networks:
The representation of trained targets in the network became sparser with training.
Networks trained on Chinese characters with higher discrimination difficulty created more distinct representations of characters which aligns with results in a human study [4].

Background

Chunking is the combination of lower level concepts to form higher level concepts, which can help push the limits of human memory capacity [1]. Hebb believes that the representation of chunks should be **sparse**: number of neurons involved should be small [9].
In a human study researchers found that higher similarity between target and distractors in search task led to higher performance at the end of training. This is because attention effects caused by false alarms in the similar case led to stronger and more holistic chunks [4].

Methods and Materials

Materials:
The full dataset was generated based on an existing human study [4][7]. It consists of 142283 200x300 gray scale images where each image contains a single Chinese character on top ("the target") chosen from 30 groups of similar characters (Figure 1), and 3-5 Chinese characters on the bottom ("the composition"). The goal is to judge whether the target is in the composition. An example of this is shown in Figure 2.
Method:
We modify the fully connected layer in a pretrained Resnet18 to have 1 output. We train 2 networks on the same 5 targets with 250 examples each for 100 epochs but *M_hard* with compositions with high target distractor similarity (TD similarity) and *M_easy* with low TD similarity. Finally, we looked at the accuracy, sparsity, and cosine distance within the characters in the same similarity group at each stage of training.

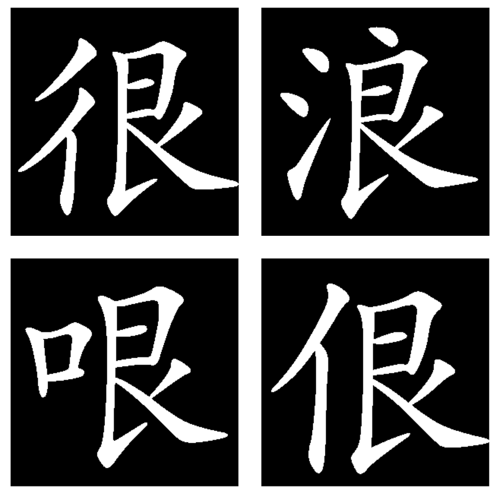

Figure 1. Example of characters in the same similarity set

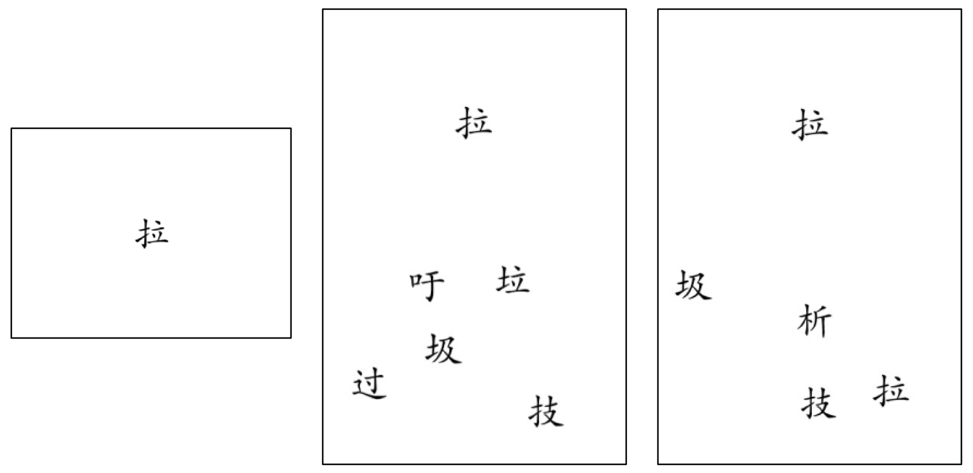

Figure 2. Example of training data

Results & Contribution

1. We found that character representations became sparser in the fully connected layer for both models, where sparsity is measured by the ratio of zero activations. Additionally, sparsity for M_easy is higher than that of M_hard (Figure 3, Figure 4).
2. We found that cosine distance between characters and other characters in the same similarity group decrease for both models, but M_hard has higher distance than M_easy (Figure 5).

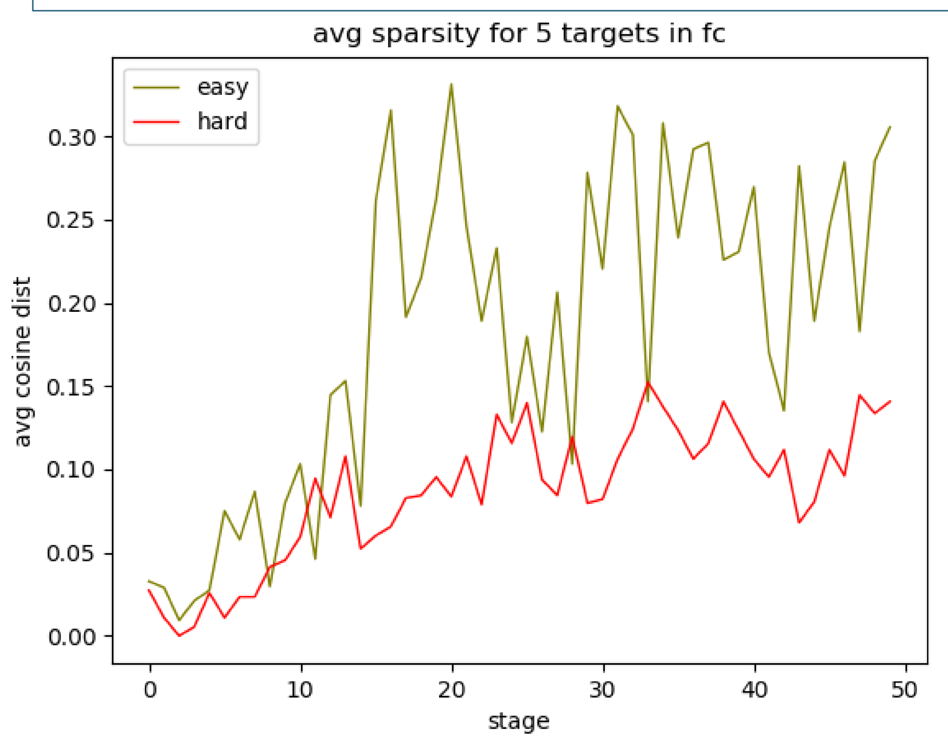

Figure 3. Sparsity of characters in the fc layer at each stage of training

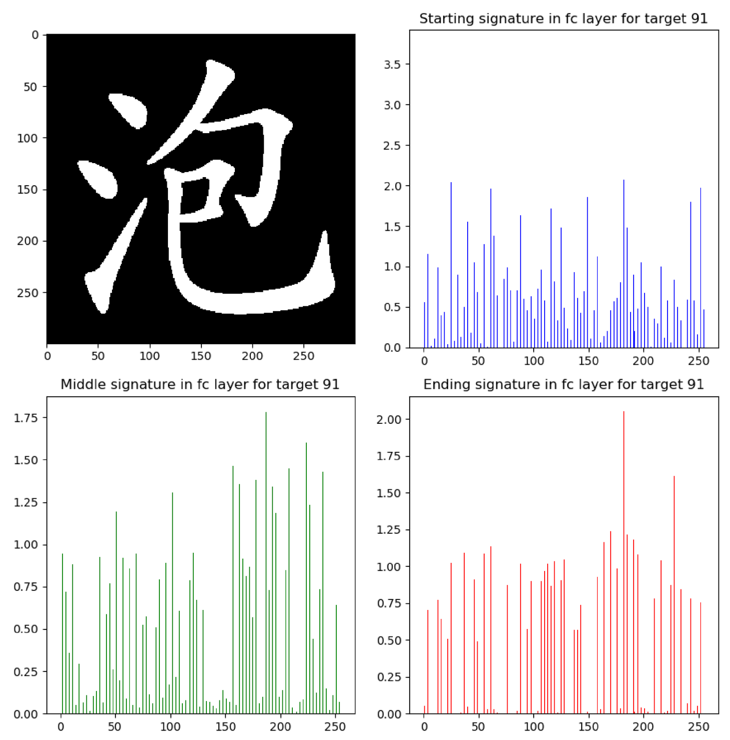

Figure 4. Representation of target91 in each stage of training.

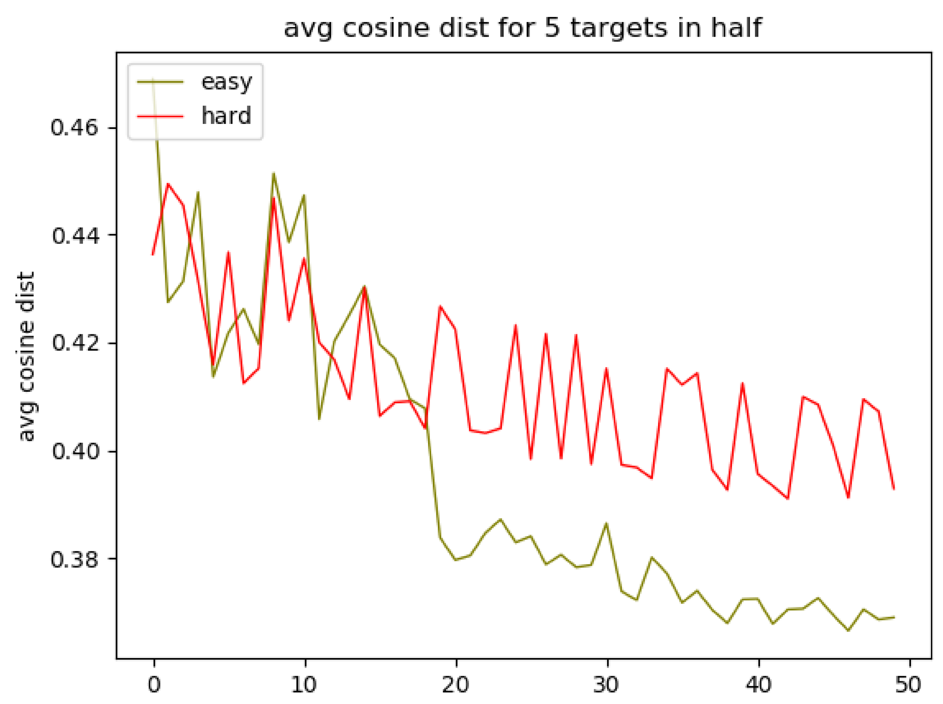

Figure 5. Average cosine distance between Targets and other characters in the same similarity set.

3. We found that accuracy in M_easy was higher than M_hard when tested on examples with higher TD similarity and we found the reverse when tested on examples with lower TD similarity (Figure 6, Figure 7).

We are unsure why M_easy produces sparser representations than M_hard. We believe that the decrease in cosine similarity overall for both M_easy and M_hard is due to more neurons having zero activation as we saw from the previous result. Comparing between M_easy and M_hard at each stage of training, we propose that M_hard has larger values because more distinct representations were formed just like in the study [4]. The accuracy for M_easy is higher in test1 because some distractors in the training set for M_easy might show up more frequently in test1 which led to better generalization. The same reasoning can be used to explain why accuracy is higher for M_hard in test2.

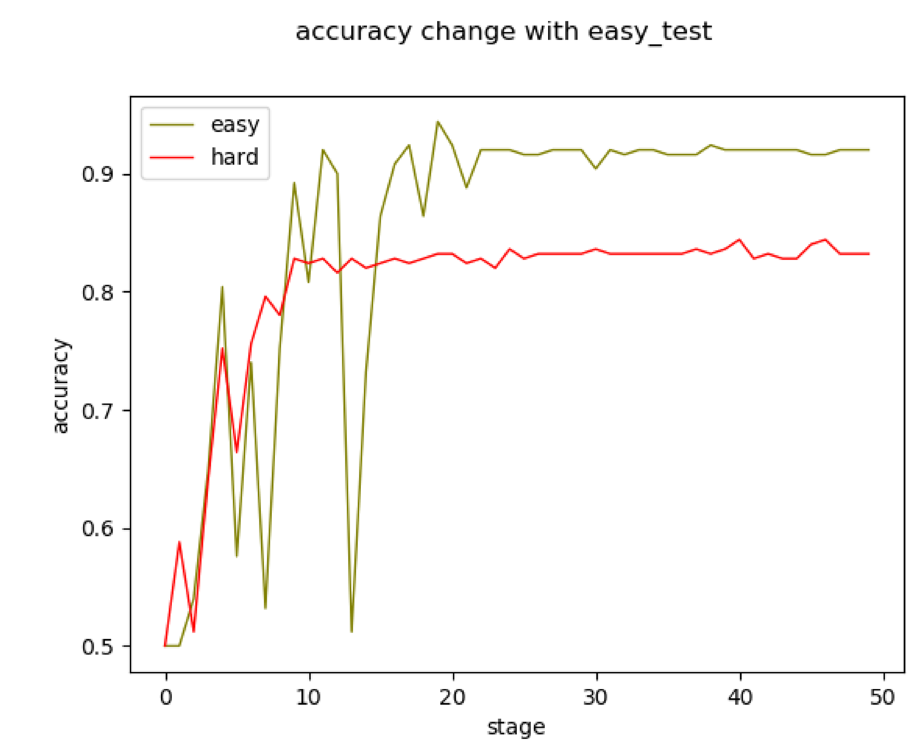

Figure 6. Accuracy on test 1

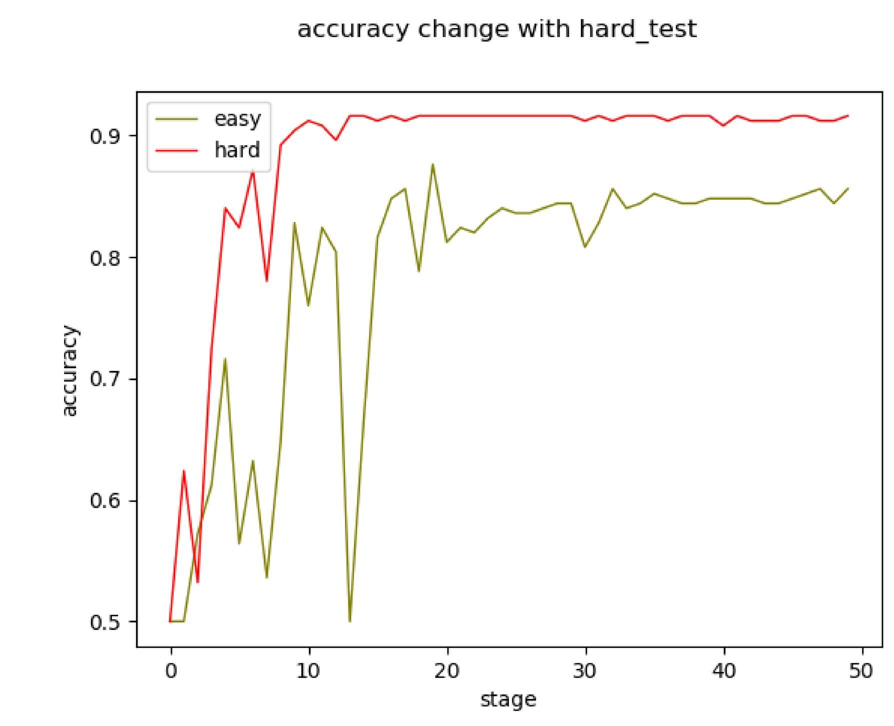

Figure 7. Accuracy on test 2

Future Work

We would like to see whether or not the trained network can still perform well on the original dataset.

Contact

Simin Li
Carnegie Mellon University School of Computer Science
happysimin@
4125196379

References

3.ul Hassan, M. (2018). VGG16–Convolutional Network for Classification and Detection. Neurohive. Dostopno na: https://neurohive. io/en/popular-networks/vgg16/[10. 4. 2019].
4.Popov, V., & Reder, L. (2019). Greater discrimination difficulty during perceptual learning leads to stronger and more distinct representations.
5.Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling* (Vol. 11). Sage.
6. Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. In *Advances in neural information processing systems* (pp. 2553-2561).
7. Reder, L. M., Liu, X. L., Keinath, A., & Popov, V. (2016). Building knowledge requires bricks, not sand: The critical role of familiar constituents in learning. *Psychonomic bulletin & review*, *23*(1), 271-277.
8. Gobet, F., Lane, P. C., Croker, S., Cheng, P. C., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in cognitive sciences*, *5*(6), 236-243.
9. Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology Press.