Extensions to Rogers and McClelland: An interesting exploration

Tong Li
tl2204@nyu.edu

Wenjie Shao
ws2237@nyu.edu

Shixuan Song
ss14381@nyu.edu

Abstract

In this project, we implement two separate methods based on R&M [3] for understanding human cognition by 1) adding a Convolutional Neural Networks frontend to the cognition network and 2) training an Recurrent Neural Network after converting the original dataset to text. We then study the differentiation of learning and simulate the effects of knowledge degradation, or dementia, by perturbing the network representations.

1. Introduction

Semantic cognition is vital to human learning process and can be thought of building blocks for our understanding of the world. Various models have been experimented to capture the underlying process of learning and acquisition of semantic knowledge. Since the start of the new millennium, computational methods have emerged and been implemented, such as the neural network model for cognitive development, adopted by R&M [3], with beliefs that the gradual learning process in such model mimics the underlying cognitive progress.

However, such obtained knowledge is not guaranteed to exist indefinitely. Rather, it is subject to be lost throughout our lifetime interactions with the world. Such degradation may be even more apparent for humans with semantic dementia. Thus, in addition to investigating the learning process, it is also worth studying the loss process of acquired semantic knowledge.

In our project, we construct two extensions to this paper, each of which is independent to the other. The first extension we experiment is by adding a CNN front-end to the cognition networks to replace the clean (perfect) input pattern in the original dataset. Another is to convert the original dataset to text and construct an Question Answering (QA) dataset and train an RNN on it.

2. Method and Experiment

In R&M [3], the original dataset consists of three parts: 8 items, 4 relations, and 36 attributes. The data vector is one-hot encoded of length 48 for each item and relation combination (as input), and its corresponding attributes (as output). Based on this, we construct two datasets for our models.

2.1. Dataset Preparation

Image Dataset To train our visual front-end (CNN) model, we scrape 400 RGB images for each item (3200 in total) for training CNN and 10 (80 in total) for the combined semantic cognition model. We apply transformations such as Resize and Crop to ensure they are in the same dimension.

QA Dataset To train the RNN model, we reverse the one-hot encoded vector to text and separate the attributes to make each input of length 3, such as (Canary Can Sing). We then permute all combinations of (Item Relation Attribute) and assign a label of 1 to represent the positive class, i.e.

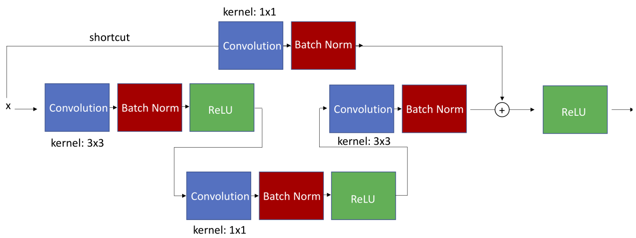

Figure 1. ResNet Skip Connection Illustration

the fact provided by the original dataset, and 0 to those not supported by the data, such as (Canary Has Leaves). This vector of words is then mapped to vector of indices using a dictionary.

2.2. Visual Frontend (CNN) Model

The original dataset [3] uses clean input for each item as represented by one-hot encoding. They assume a very high-level of categorization and knowledge about the item. However, this is not usually what we obtain, especially during the initial learning phase. The stimulus we receive is not a perfect category. It could be direct visual stimulus such as images. During initial learning process, we do not know what the image represents and thus cannot extract its properties. To infer the attribute from what we see, we need to first either categorize it into a known object class, or learn it as a new category.

Thus, we plan to build a CNN with PyTorch to mimic this categorization process. We adopt the Residual Net 101 model architecture. Developed by He et. al. [2], this deep model allows skip connections as illustrated in Fig 1 and reduces the training time. We train this model from scratch on 3200 images (400 for each item) and are able to achieve a 97.5% accuracy on the test set.

We then combine the CNN with the semantic cognition model. Instead of providing a clean input for the item, we loop through the images in the test set (10 for each item), and one-hot encode this prediction for each image. That is, we are using CNN as a visual front-end to the R&M architecture. The internal process of making the prediction for images in CNN can be considered as the categorization process of human. The encoded item input is then combined with the relation to form input patterns to the feed-forward network which outputs a list of attributes.

2.2.1 Degradation Effects

To study the effects of degradation in semantic dementia, we adopt two approaches to perturb the representation layer. The first approach is that after fully training the cognition network, we add a random noise $\epsilon \in N(0,\sigma^2),\ \ \epsilon \in \mathbf{R}^8,\ \ \sigma^2 \in [0,8]$. σ^2 that represents the magnitude of noise. We then use this perturbed representation to run the cognition model for dozens of times more and extract the mean of representation layer weights (activation values) under noise from these runs as the final output. Another approach is to apply dropout to the representation layer, which is previously used in neural net to avoid overfitting. As in the previous case, we perturb the weights from the fully trained network and repeat dozens of times and take the average. We select different levels of dropout probabilities for the representation layer. Each dimension has the same probability of its layer being set to 0 for each level. We then plot the layer activation values after these two kinds of perturbations for different levels of specificity of different relations.

2.3. Question Answering with RNN

Besides the R&M architecture, we would like to find an alternative approach to mimic how human digest information and constantly shaping their cognition based on previous knowledge. One potential method proposed by Dezfouli et al. [1] is to represent the complex learning and cognitive processes by humans through RNNs. We design a question answering model with RNN using Keras framework which achieves a training accuracy of 98.74% with an epoch number of 2500.

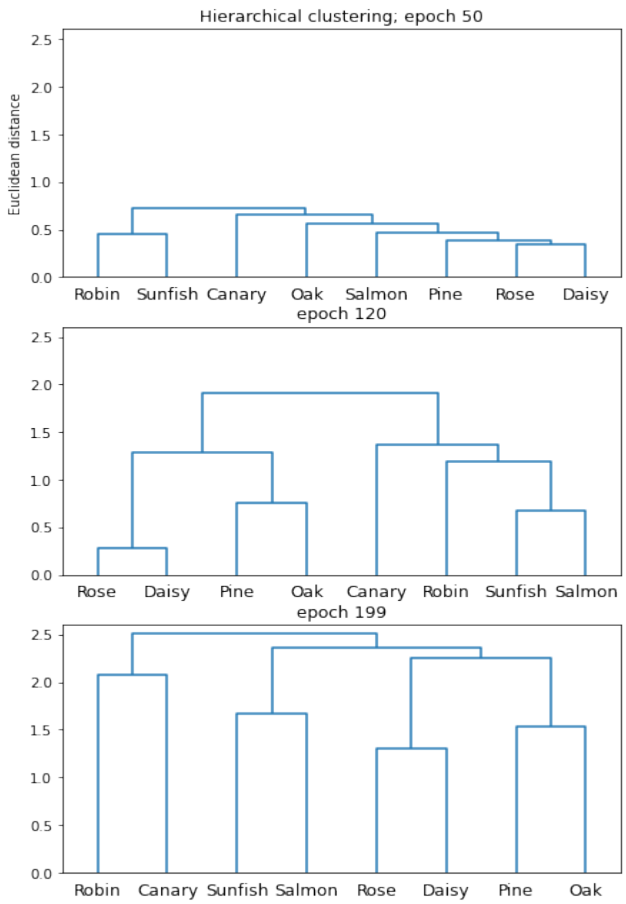

Figure 2. CNN Frontend Differentiation

3. Results for Visual Frontend

3.1. Differentiation after learning

After training the model for 50, 120, and 200 epochs (each item is eventually trained 2000 times since we have 10 images for each item), we extract the layer weights for item representations and create a hierarchical plot as shown in Fig 2. We can see that the internal weights follow a broad-to-specific learning pattern as in the human cognitive development process. At 50 epochs it is not differentiating much and has roughly the same weights for all items. At 120 epochs we see that plants have been differentiated from animals. At 200 epochs, we can tell that all items have been differentiated very specifically.

3.2. Degradation When Noise is Added

We plot the mean layer activation values for different levels of noise (variance and dropout probability). For random Normal noise, Fig 3 shows the effects on relations 'ISA' and 'Has'. Fig 4 shows the effects when we use dropout. For each approach, we combine the facts to four categories: General, Narrowed, Specific and False. The first three represent the specificity of item-relation-attribute, while false means a wrong item-relation-attribute bridge. By looking at the graphs, we can find that with the increasing level of noise, layer activation values begin to decrease which mimics the semantic dementia symptoms (progressive brain damage). We also notice that the degradation effects on specific and narrowed attributes are much larger than the general attributes. We can hypothesize that the loss of knowledge and semantic cognition follows a specific-to-general pattern, which is in reverse order to the learning process. In addition, we find that False combinations now receive higher activation values when the noise is intensified, illustrating that dementia could lead to wrong inference between items and attributes.

4. Results for Question Answering with RNN

To study the process of differentiation in the context of RNN with question answering, we extract the weights from the embedding layer for the items. The results for 500, 1000, and 2500 epochs are shown in Fig 5. From these plots, we can barely find improvements between different stages as we compare the model trained with different numbers of epochs, despite the high training accuracy. In the plot with 2500 epochs, we notice several obvious categorization disparities. For example, rose and salmon are categorized together indicating they have been identified as most similar under the model. Some animals like robin, canary and salmon are clearly different from the neighboring items. As a re-

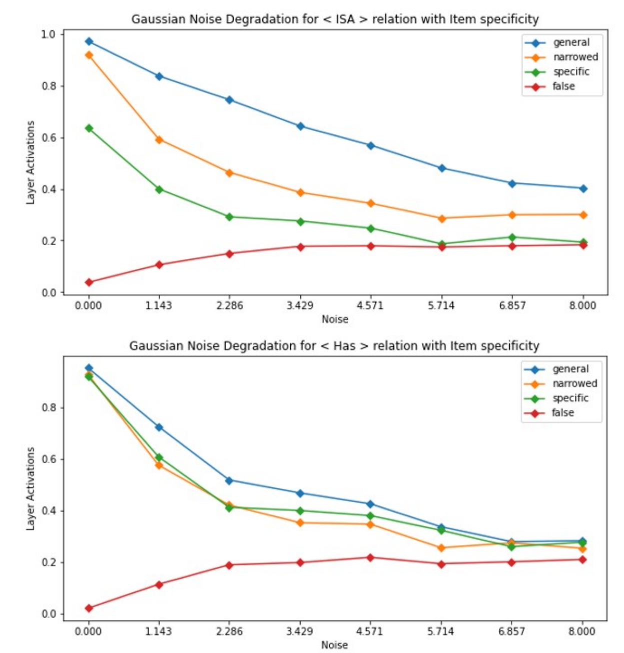

Figure 3. Degradation with Gaussian Noise for ISA and HAS

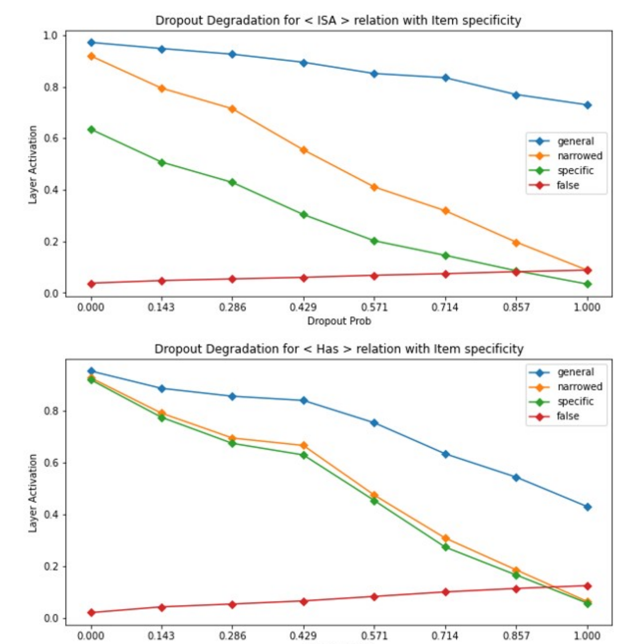

Figure 4. Degradation with Dropout Noise for ISA and HAS

sult, these clustering plots cannot represent the differentiation very well. One potential reason for such result is that since the size of our vocabulary

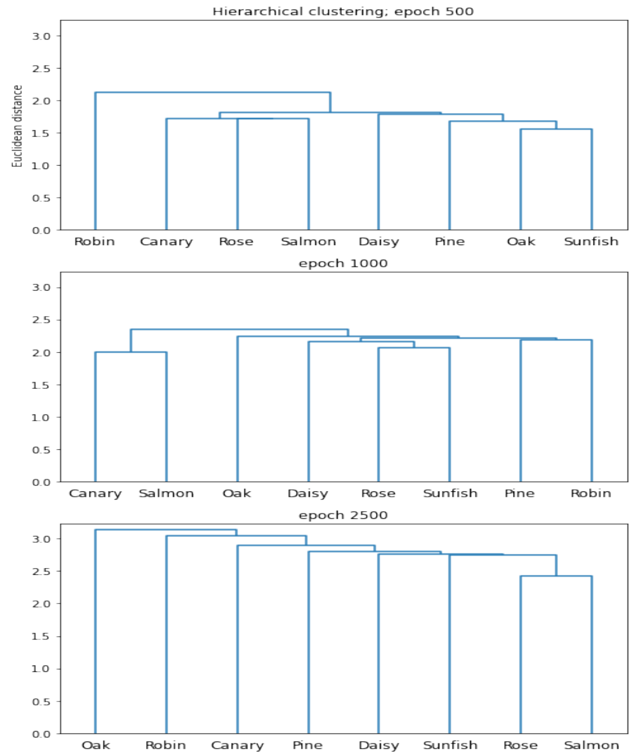

Figure 5. RNN weights differentiation

is small, the embedding does not need to differ much to make predictions and therefore we do not observe obvious differentiation between items.

5. Conclusion

In our project, we build two models based on R&M to further study cognitive learning process and degradation. We start by eliminating the assumption of a very high level knowledge and thus a clean input as in R&M by taking images (visual stimulus) as inputs to pass through a CNN frontend, the output of which is then combined with the R&M architecture. In this model, we are able to produce similar patterns (broad to specific) of differentiation during learning as in the original experiment, even though we use the image as input instead of a clean one. It is worth noting humans can typically learn a category and its properties in a few images while CNN requires to be trained on a relatively large dataset (3200 images) in order to achieve human performance. Realizing that acquired knowledge may not be permanent, we then study the effects of degradation during semantic dementia. We find that the pattern of knowledge loss is in the reverse order of learning, i.e. the cognition is lost from specific to general. We also find that with the increasing level of noise, false item-relation-attribute bridge receives more activation, which is similar to the dementia where wrong inference is made between item and properties.

We then build an RNN on QA that takes in firm statements and corresponding answers (Yes or No) as training set (instead of the original one-hot encoder vector) to study differentiation. However, after 2500 epochs, we are not able to obtain a similar pattern as in R&M.

6. Future Work

A future work could be to reconstruct the RNN to obtain a more meaningful structure for extracting item representations. It is also interesting to combine two models together and create a visual question answering net.

7. Acknowledgement

We build our models based on PyTorch and Keras. The skeleton for implementing degradation to the model is based on a GitHub Repository. [4]

References

[1] A. Dezfouli, K. Griffiths, F. Ramos, P. Dayan, and B. Balleine. Models that learn how humans learn: The case of decision-making and its disorders. *PLOS Computational Biology*, 6, 2019.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[3] J. McClelland and T. Rogers. The parallel distributed processing approach to semantic cognition. *Nature Reviews Neuroscience*, 2003.

[4] petersim. Github repo. `https://github.com/petersim1/CCM_Final_Project`.