Semantic concept learning with an RNN

Guillermo Benjamín Grande & Jasper Wilson

1 Introduction

Rumelhart designed a neural network that used items and relations as inputs, and output the attributes that were true for that item relation pair. This model had two hidden layers: a representation layer that received activations from the item layer, and a hidden layer that received activations from this representation layer and the relation layer. This model is important for several reasons. The first is that it accurately learns the correct attributes for each item relation pair. The second is that the representation layer separates the items by category in a way that seems intuitive. The third is that adding random noise to the weights from the item layer to the representation layer mimicked the expected effect of destruction of analogous neurons in a human brain.

This paper seeks to extend this model by interpreting it as a question answering model that receives items, relations, and attributes as inputs and returns a true-false distinction as the output. Our new model has two other key differences from the Rumelhart model.

The first key difference is that because our model uses attributes as an input instead of an output, we can investigate how attributes are represented by adding a hidden layer that receives activations from the attribute layer. We expect attributes to either be clustered by semantics or grammar. If they are clustered by semantics we would expect to see attributes that relate to specific items to be clustered together. For example, "plant" and "leaves" should be clustered together because the items they relate to overlap. If they are clustered by grammar we would expect to see attributes that are related to different items in the same way to cluster together. For example, "leaves" will be close to "scales" since they both use the "HAS" relation. This could possibly give us some insight on whether we

represent attributes with an abstract understanding of the meaning of that attribute (we would expect to see clusters by grammar), or if we represent attributes with exemplars possessing them (we would expect to see clusters by item relatedness).

The second key difference is that we can include a temporal component by converting this into a recurrent neural network, which more closely reflects how we actually cognize questions. McClelland suggests in his 2008 article that many cognitive processes are temporal by nature, and the way this model presents question answering might be one of these processes. Because the item is presented first in a question like "Does a salmon have scales?", we expect that this should have a priming effect on relevant attributes (Collins & Loftus 1975). This priming effect is why we expect that accounting for time with our new model will improve performance.

2 Methods

Being a derivative of the Rumelhart model, our model analyzed in this paper uses the same items, relations, and attributes as in the original. However, due to the model focusing instead on outputting a single bit (1 for yes, 0 for no) rather than a subset of attributes, the training data had to be re-initialized. Both the forward pass and backpropagation treat each time step differently in terms of what inputs are being activated and thus the computation had to be tweaked. Finally, an optimal learning rate was chosen and representation analyses, PCA, and noise analyses were all implemented to see how the two representation layers evolve.

2.1 Initializing data

From the Rumelhart model there are 8 items, 4 relations, and 36 attributes, yielding a total of 1152 possible questions this new model can be asked. As such, the training data was initialized in blocks of 8×1152 for items, 4×1152 for relations, 36×1152

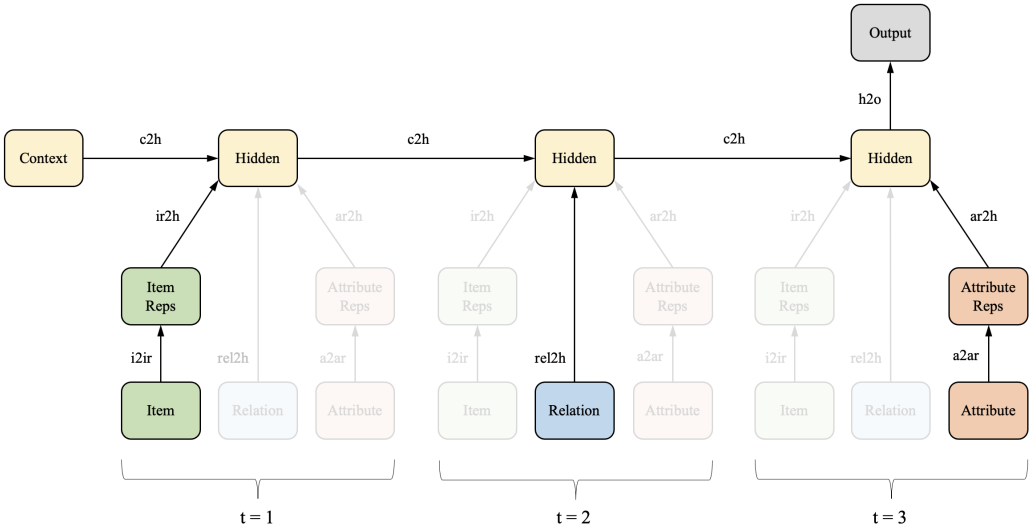


Figure 1: The RNN model for concept learning (transparent items are not active)

for attributes, and 1×1152 for outputs. Each column of these blocks thus represents a training input/output sample. The samples were organized in a way such that, in order, every item was paired with all relations one-by-one and the first attribute, and then again for the second attribute and so on. Finally, it was decided that the model would output 0 for questions that did not follow the grammar of the original model (for example, "Pine is a red?") and thus the output block would only contain 1's in the columns corresponding to input sample 3-tuples (item, relation, attribute) that are both grammatically valid and have "yes" for an answer. This initialization and an accompanying spreadsheet can be found in initData.

2.2 RNN structure

Cognitively speaking, when a question is asked it is plausible for concept representations to be activated before the asking is done. This model accounts for this priming possibility by introducing a temporal dimension. Questions that can serve as inputs to this model are of the form "item relation attribute?" with each word being presented in a distinct time step. As such, each run of the model for one sample 4-tuple (item,

relation, attribute, output) takes the same input sample 3-tuple at every time step with the necessary inputs zeroed. So, for $t = 1$, relations and attributes will be zeroed but not items and so on. The forward pass can be found in forwardPass.m within the rnnModel folder.

For times 1 and 2 the model does not output anything as the question is not finalized. Due to this, the forward pass for these time steps only calculates the hidden activations, which will serve as the recurrent factor and context activations for the next time step. The final output activation will be the result of the forward pass at $t = 3$, which will then be tested against the true sample output to calculate the loss for that pass of the model.

For backpropagation, each weight will be updated via a single backpropagation. Since only one input is active at any time step, the weights corresponding to attribute representations and attributes for example, (ar2h and a2ar in Figure 1) will be updated by a simple backprop without worrying about time. Other weights, including those corresponding to relations, items, and contexts, will be updated via one backpropagation traveling back through time. These computations can be found in trainModel.m.

2.3 Hyperparameters and analyses

The only two specifiable hyper-parameters for training the model are the number of epochs and the learning rate. The number of epochs was chosen to be 2500 to follow in the steps of McClelland's analysis, whilst the learning rate was chosen to be 0.01 after plotting training errors with varying learning rates and picking the error minimizer.

With regards to post-training analysis, we looked at how the item and attribute representation layer activations evolved over time (epochs 250, 1000, and 2500). We then passed these activations through matlab's built-in PCA function to get a clearer perspective on how the item and attribute representations clustered initially and at the end of training. It is important to notice, particularly for attribute representations, that PCA is vulnerable to some information loss; nonetheless, adding a third component to the

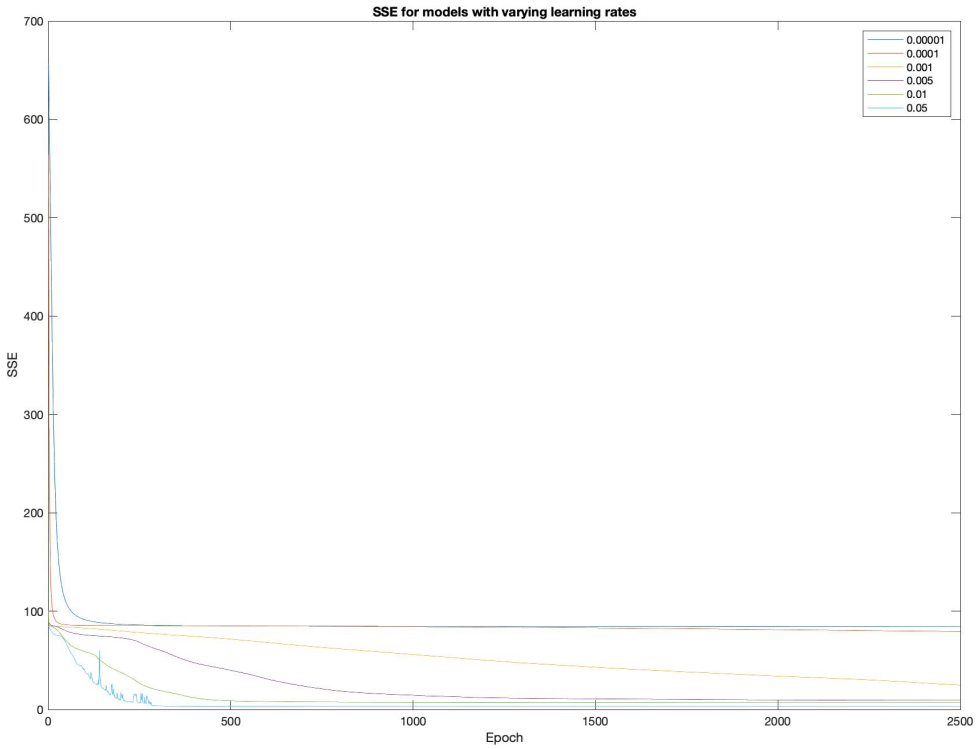


Figure 2: Evolution of SSE in model training for varying learning rates

analysis did not further aid in visualizing clusters. The analyses can be found in the rnnWorkspace.m pipeline.

Furthermore, we looked at how adding increasing noise terms to the weights connecting items to their representations (i2ir) and attributes to their representations (a2ar) affected output activations by adding to the weights random normal matrices with mean 0 and increasing standard deviations $(0, 1, 2, 3, 4, 5, 6)$.

3 Results

With a learning rate of 0.01 the model's error dropped rapidly until 500 epochs and slowed down dramatically for the remaining 2000 epochs. Training this model ten times gave a mean SSE of 6.298 and a mean classification error of 1.74%. The remaining results in this section notedly highlight what we believe to be our findings, it should be noted

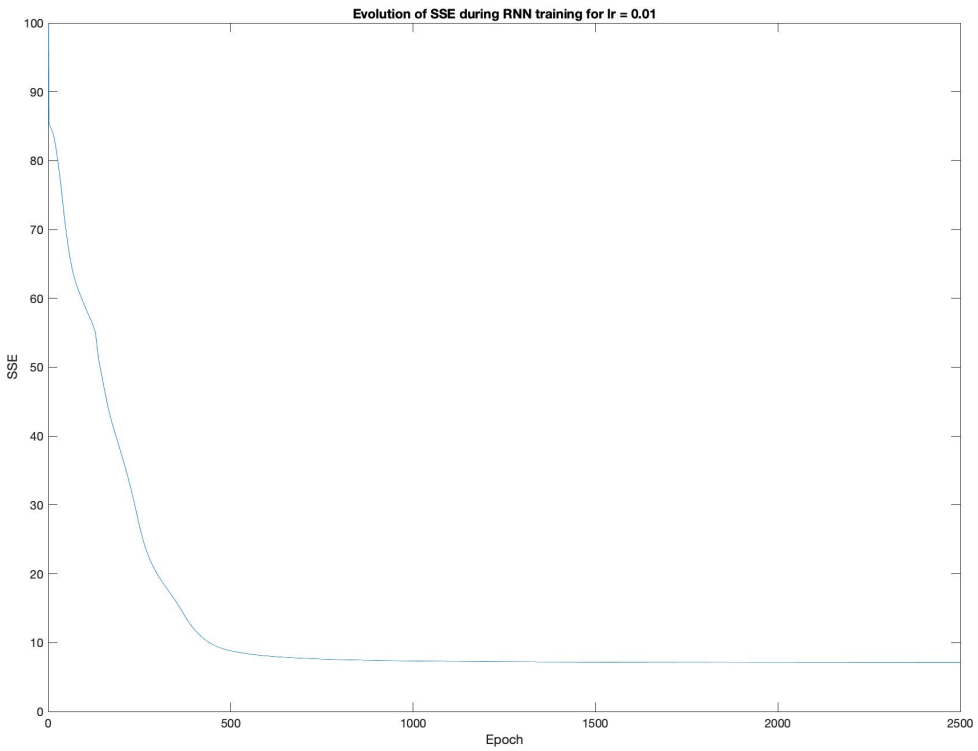


Figure 3: Evolution of SSE in model training over 2500 epochs with learning rate of 0.01

that these results were not always consistent.

The PCA for item representations at initial stages of training vs. final stages show a separation between animal and plant representations, as well as vague pair groupings. The PCA for attribute representations at initial stages of training v.s final stages show differing cluster formations.

Finally, adding random noise terms with increasing standard deviations to the item to item representation and attribute to attribute representation weights (this was the same method of adding noise described in Rogers & McClelland 2003), yielded decreases in output activation for questions with "yes" as an answer and increases in output activation for a question with "no" as an answer.

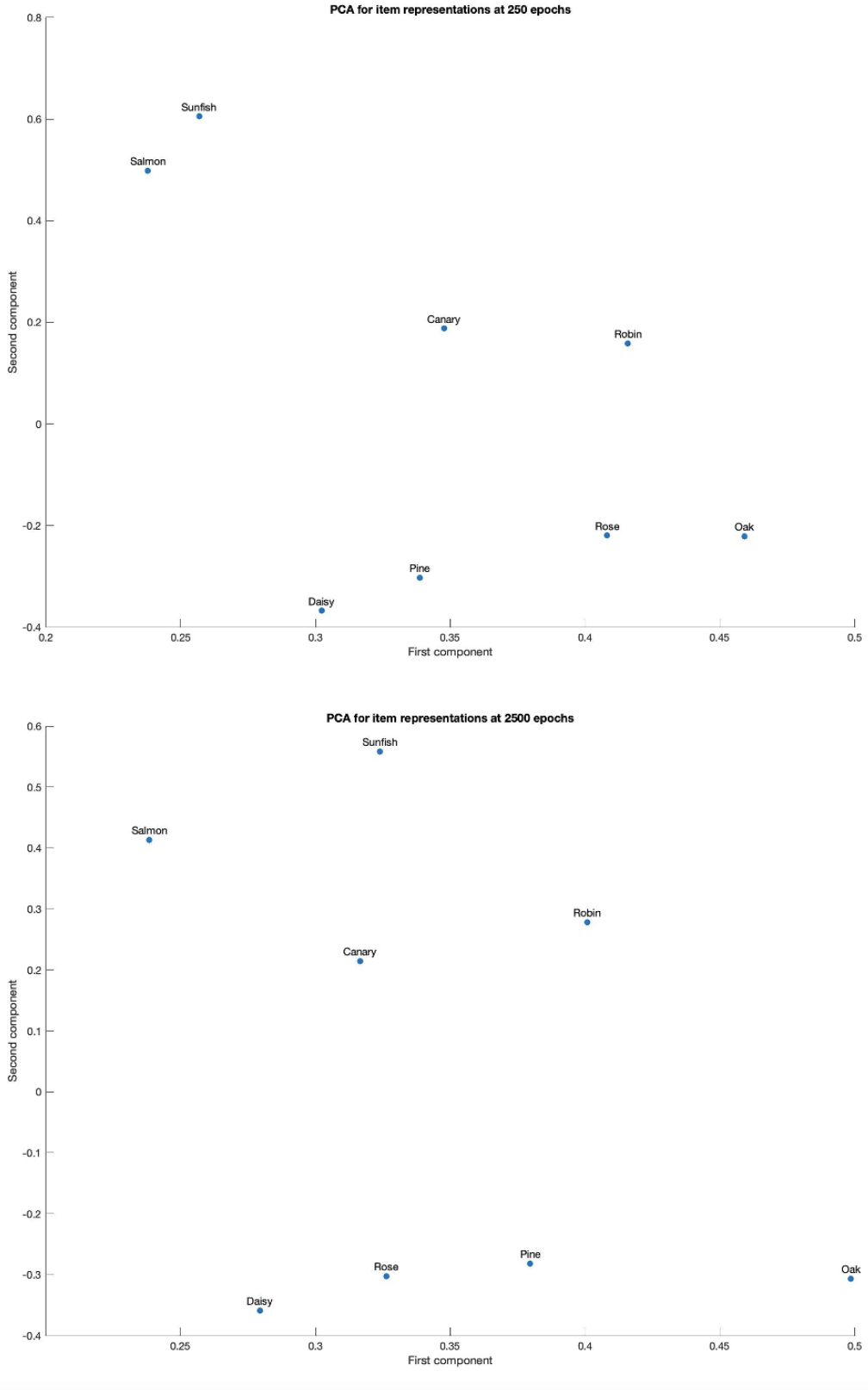
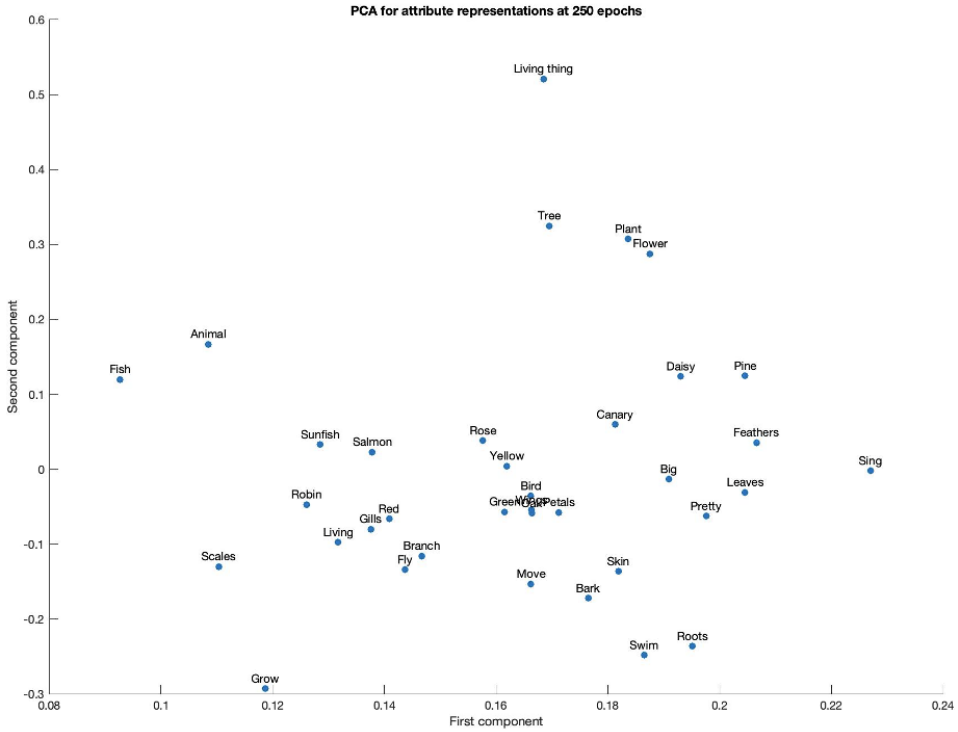


Figure 4: Item representation activations at 250 and 2500 epochs, respectively



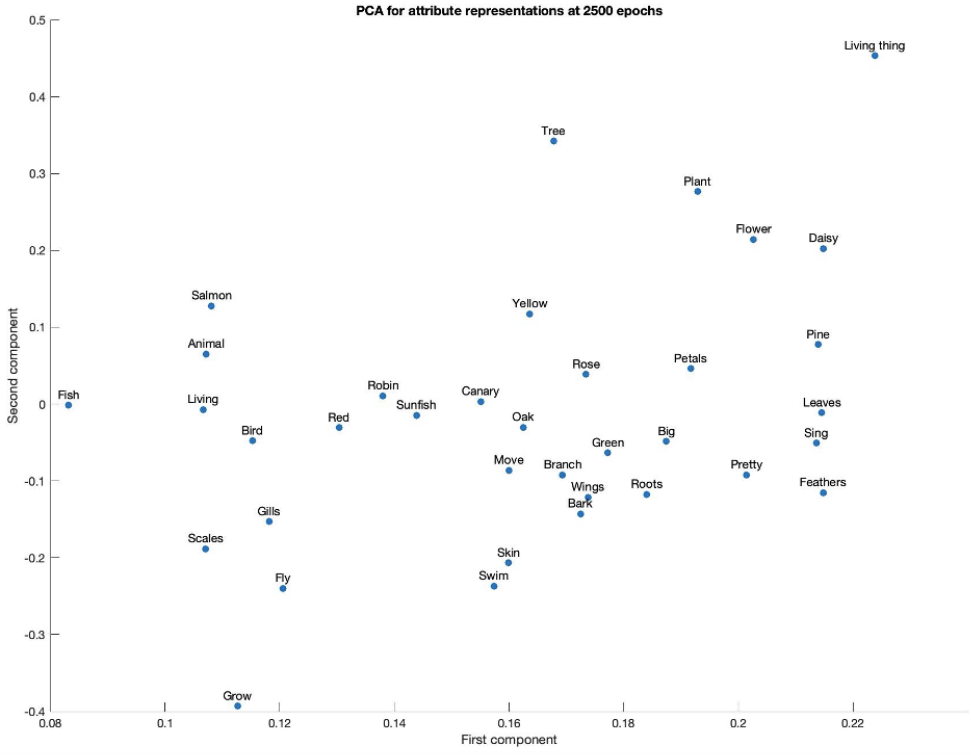


Figure 5: Attribute representation activations at 250 and 2500 epochs, respectively

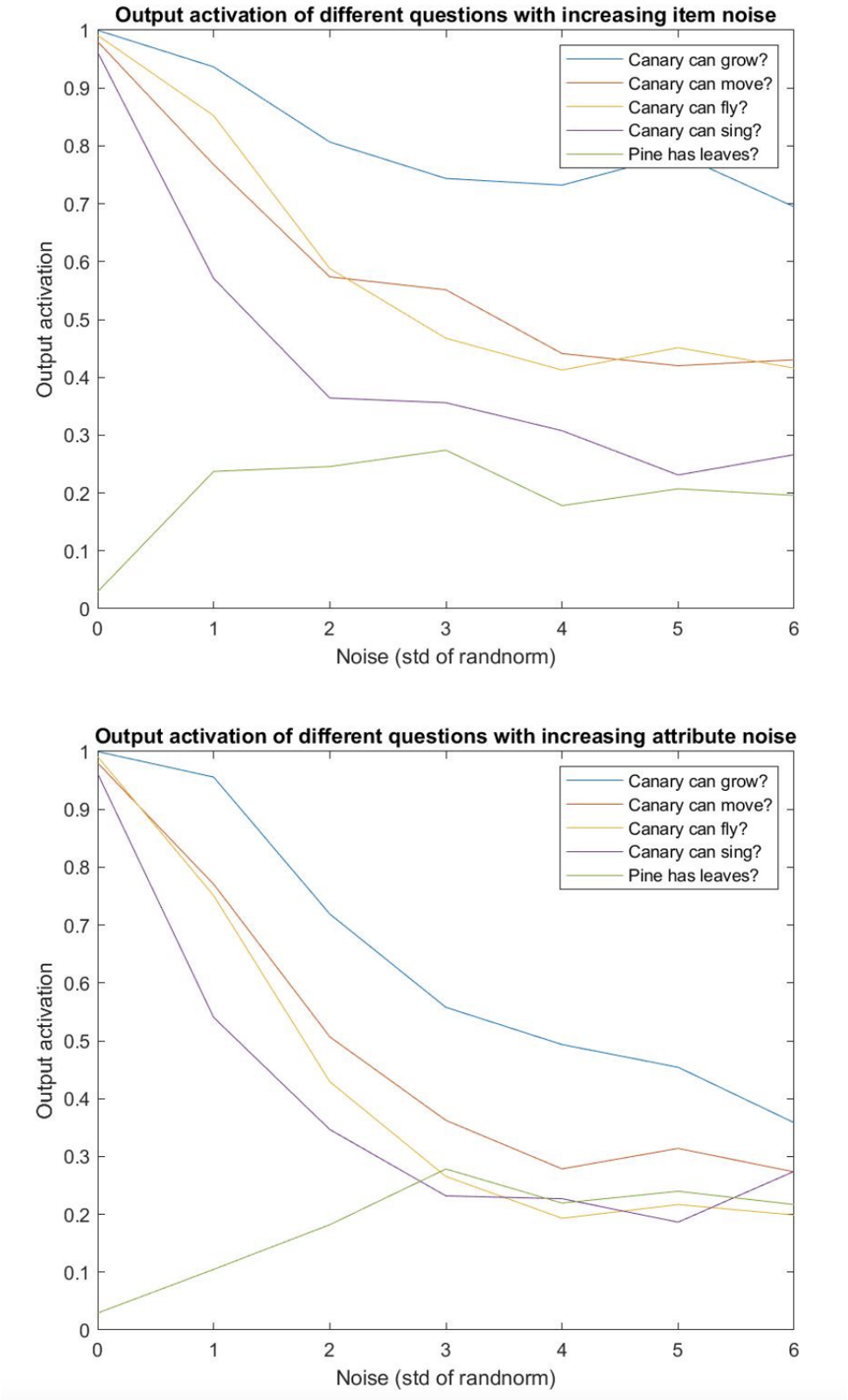


Figure 6: Output activations for different questions with increasing item and attribute noise, respectively

4 Discussion

This model performed worse than the Rumelhart model which consistently had an SSE below one and a classification error of zero. This could be because while this model more accurately represents the process of hearing and interpreting questions, the original model might be more analogous to the process of learning about objects. This suggests that while learning, we ask the question "What does a tree have?" rather than asking "Does a tree have . . . ?" for each attribute.

4.1 Analyses

The PCA clustered the items in the same two major intuitive categories (plants and animals) four intuitive sub-categories (birds, fish, trees, flowers) that the original Rumelhart model did (Figure 4). However, these clusters were not as clear as the clusters described in the Rogers and McClelland paper, and occasionally the model would cluster incorrectly. This faulty clustering was not associated with especially high SSE or classification error. It is also worth noting that these clusters often became less distinct as the model iterated (Figure 4). This suggests that the model might be overemphasizing individual differences. This overemphasis might be due to the items to representation weights being optimized faster than the rest of the model. So while the model is improving on the whole, these quickly optimized weights could be plateauing or even worsening.

The PCA did not consistently cluster the attributes in either of the ways we suggested it might in the introduction. It seems instead that the clusters were a mixture of the two. The top left corner is dominated by attributes that relate to plants with the "IS A" relation. On the left there are attributes that relate to the items in the "animal" category, and in the lower part of the center "branch", "wings", "bark", and "roots" all share the "HAS" relation and are very close to another. PCAs of other trials of the same model also had mixed methods of clustering. Unfortunately none of these clusters are

very distinct, so while there is some indication that both grammar and semantics are important, it seems more reasonable to accept that the PCA does not tell us anything meaningful about how attributes are represented.

Adding increasing amounts of noise to the weights between the item and attribute layers and their respective representation layers created the same disruptions in activation that were reported in the Rogers & McClelland paper. This result suggests that the issues caused by semantic dementia could be related to the deterioration of either the representation of items or the representation of attributes.

4.2 Limitations

An important hurdle we had to jump over in this project was implementing the RNN with sparse data. Given that a majority of sample input 3-tuples form either ungrammatical questions or questions with "no" for an answer, a lot of outputs were being compared to 0, with only $< 10\%$ of compared outputs being 1. This made learning somewhat difficult, and initial versions of the model would favor outputting values very close to 0 a majority of times to minimize error. It was figured out that this was being caused by a domination of the hidden activations (all 1's) serving as context activations for the next time step of forward pass. Having all hidden layer neurons output 1 due to a logistic activation function and large context activations meant that a majority of the backpropagation would be rendered null (the derivative would yield a term very close to 0, which then had to be multiplied to a majority of backprop calculations). To deal with this issue, we allowed the initialization of all weights and biases to be uniformly random within $(-1, 1)$; also, we used a tanh activation for the hidden layer to allow hidden activations to also possibly be negative and within $(-1, 1)$. This might not be analogous to how synapses function in the brain, and thus it is possible that the issues seen in the analysis could be caused by these tweaks.

Another limitation faced was the model's high variance, also potentially due to this

tweaked initialization. It is quite possible that in two independent training sessions of the same length, the first model will correctly answer "Canary can sing?" while the second will not. It seems like there are certain representations that the model has a harder time correctly portraying over others; this might be due to scarcity of inputs, and thus negative sample 4-tuples (ones that are ungrammatical and/or have 0 outputs) dominating how representations change in the backpropagation instead of positive sample 4-tuples.

4.3 Prospects

This model has varying inconsistencies that could be ameliorated. However, it could also be the case that the model is missing something. It is plausible, for example, that representations of relations play a role when connecting items to valid attributes; it would make sense to test the model with an added relation representations layer and see how they evolve over time, and whether these representations intersect with item or attribute representations and help the training of the model.

Further time must also be put in regards to the PCAs to see what variables are causing certain clusterings, specifically the unexpected ones. It might be that separating the attribute representation PCA into relations or overarching concepts of "color" for example, will create a sturdier foundation for analysis.

References

Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. Psychological Review, 82(6), 407–428. https://doi-org.libproxy.wustl.edu/10.1037/0033-295X.82.6.407

Rogers, T. T., & McClelland, J. L. (2003). Semantic Cognition: A parallel distributed processing approach.

Thomas, M. S. C., & McClelland, J. L. (2008). Connectionist models of cognition.