Learn to Integrate Diagram and Text in AI Geometry Reasoning

Authors: Xuefei Yuan, Te-Lin Wu
Mentor TA: Bryan McCann

Abstract In this project, we simulate geometry concepts acquisition with unsupervised learning in deep neural network. Specifically, we build a model that learns to combine diagrams and text descriptions for geometry problems. The results show that the network is able to learn several geometry concepts, including line lengths, angle sizes, line orientations, similar triangles, etc. To understand what are learned during the unsupervised learning, we analyze the response profiles of units in hidden layers and find some units resembling cortical neurons in primate brain. We also test whether the model can generalize the learned geometry concepts to new shapes, and we find that the generalization is very poor. Using unsupervised deep learning, the current study provides a modeling framework to simulate how people learn to combine diagrams and text descriptions in geometry reasoning, which has significant implications for cognitive psychology and computational neuroscience.

Introduction

In the past few decades, many algorithms in machine learning and artificial intelligence have been applied to cognitive sciences to simulate human behavior. Among them, artificial neural network modeling has been very helpful in understanding human cognitive functions [1]. In this project, we want to use this connectionist approach to simulate geometry concept acquisition with unsupervised learning in deep neural network. Specifically, we aim to build a model that learns to combine diagrams and text descriptions for geometry problems. The motivation for the current research is multi-folds.

First, we want to understand if unsupervised learning can naturally extract essential geometry information. A previous research done by Staniov and Zorzi showed that visual numerosity spontaneously emerges in ‘deep networks’ that learn a hierarchical generative model of the sensory input. We hope to know whether the same results could be found in geometry cognition.

Therefore, we adopt a training procedure similar to their study. At the first stage, the network learns to efficiently encode the geometry figures, without any labels. At the second stage, the efficient representation of the input is used to train several classifiers to generate proper predicates for geometry figures, such as comparing the sizes of two angles, deciding whether two triangles are similar or not. We are curious to know if unsupervised learning at the first stage could give rise to hidden units that extract geometry properties of the visual input.

Second, if we successfully train a deep neural network to combine geometry images with their corresponding text descriptions, we want to know how it performs the task. The ultimate goal of this project is not achieving the highest accuracy, but to understand how the network learns to perform those tasks. By examining the response profiles of the hidden layer units, we hope to gain some insights into how people learn geometry from a computational neuroscience perspective. For instance, after we train the network to efficiently encode the visual inputs, can we find neurons in the hidden layer that are selective to distance between two parallel lines? If so, these neurons resemble the magnitude-sensitive neurons in parietal cortex [2] and they may contribute to magnitude representation.

Third, we want to compare several potential read-out mechanisms of neural codes. A common practice in neuroscience is to train a linear classifier fed with the recorded firing rates to infer which neuron has stimulus selectivity. However, this method might be problematic if the neural codes are not linear. In addition, converging evidence shows that the brain might use a generative model rather than a discriminative model to make inferences [3]. Therefore, the neural codes should be decoded using Bayesian inference [4]. To test these ideas in our model, we implement three classifiers, i.e., logistic regression (discriminative model), multilayer perceptron (non-linear) and Naïve Bayes (generative model), to read-out the neural codes in our deep network. Comparison between the behaviors of these three classifiers will deepen our understanding on these potential read-out mechanisms of neural codes.

Finally, we want to understand to what extent the learned geometry concepts generalize to new shapes. This would provide insights into human analogical reasoning.

The rest of this paper is structured as follows. We first explain how we define the tasks, generate the input dataset, and train the model. Next we present the results and discussion for each task, followed by the main conclusions.

Method

Input We generated a parameterized dataset including 2112 figures in total. Each input figure is a 51-by-51 binary matrix. The figure type and their corresponding parameters can be found in the online supplementary material (web.stanford.edu/~xfyuan/CS229Project/).

Training Procedure The unsupervised learning stage is implemented by a stack of restricted Boltzmann machines (RBMs). Due to the limit of space, we are unable to present the method in detail here. We used the same model architecture and training procedure as described in [5], except that the hidden layer size is 400-225-100-49-4 and no fine-tuning is involved. To speed up the training, we use the GPU implementation of the deep learning network provided in [6]. We modified their codes to train our model.

After the unsupervised learning stage, we use the activities of the each hidden layer to separately train classifiers to perform different tasks. We have 3 classifiers: logistic regression, Naïve Bayes, multilayer perceptron. The multilayer perceptron contains one hidden layer of 100 units. Accuracies reported in the results session are computed by 5-fold cross validation.

For the purpose of model comparison, at the first stage we also implement two other forms of dimension reduction. The first one is to directly resize the image to match the dimension of the top layer (4 pixels). Another one is to apply principle component analysis (PCA) to the visual input and use the first 4 principle components to represent the visual input. Also, we use the top layer representation to reconstruct the visual input. At the second stage, we trained classifiers fed with these four types of codes, i.e., "resized image", "PCA reduced image", "reconstructed image", "raw visual input" to perform various tasks described later.

Finally, we train a new network with hidden layer size [400, 225, 100, 49, 2] to visualize how well the network learns geometry concepts.

Experiments

Predicates learning. We examined if the network can compare the properties of different figures. We first replicate the learned network to construct a second network. We then ask if the input figure in Network 1 is "*" than the input figure in Network 2, where "*" can be different predicates depending on the type of the input figure. For instance, if the input is "line", then the value of "*" are "longer", "shorter" or "equal". Under this framework, our model learns to perform 3 tasks. (**A**) Magnitude comparison, i.e., the comparison of the lengths of two lines, the size of two acute angles and the distances between two parallel lines. (**B**) Spatial relationship regarding rotation. We compare the orientation of two lines to determine whether the one in Network 1 is clockwise or counter-clockwise relative to the one in Network 2. Note that the model needs to extract the orientation feature and ignore the length feature. (**C**) Similar triangles. We ask if the classifier could correctly predict whether two triangles are similar.

Generalization of concepts. To probe the degree to which the model integrates the semantics of the text description and its visual input, we use 3 tasks to test its ability to generalize the learned text descriptions across different input types. (**A**) The generalization of magnitude comparison. In this task, the classifiers are trained to compare line lengths. We test whether they can generalize to circle size comparison. (**B**) The generalization of similar triangles. The model is trained by triangles. We test if this concept can generalize to "similar quadrilaterals". Currently, we only use non-square rhombuses and squares to represent quadrilaterals. (**C**) The generalization of rotation. The model is trained by comparing two figures of a single line to predict whether the line in figure A is clockwise" or "counter-clockwise" relative to the line in figure B. We test whether it can generalize to comparing the orientation of figures containing two lines.

Results and Discussions

The results show that the deep neural network successfully learned the features that are needed to reconstruct the figure, as shown in the comparison between the raw data and the reconstructed images (see Fig. 2 in *Supplementary material*). Overall, the best performance is achieved by multilayer perceptron, followed by multi-class logistic Regression, and Naïve Bayes has the worst performance. In addition, the performance decreases in higher level of the network. Next we report and discuss the results for individual task.

Magnitude Comparison

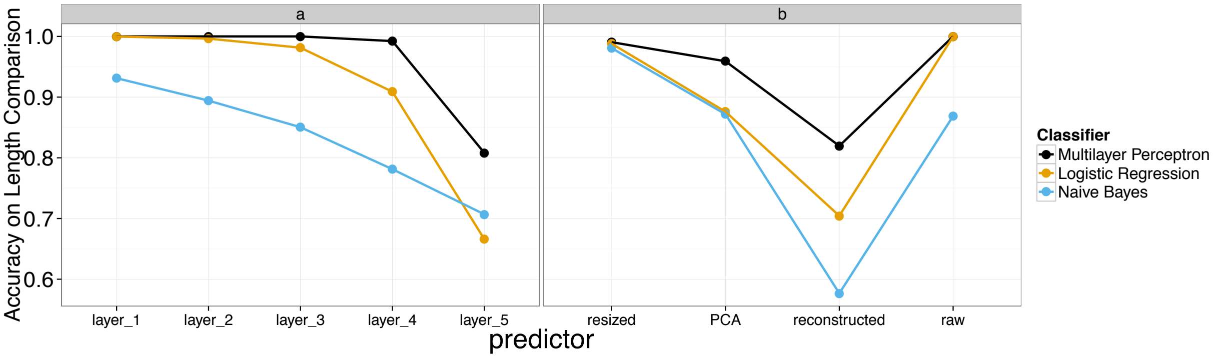

Fig. 1 Accuracy on comparison of line length

We find that classifiers fed with activities of top layer have lower performance than those fed with activities of raw input (Fig. 1). Comparing angle sizes and distances between two lines yield accuracy patterns similar to Fig. 1. This contradicts Stoianov and Zorzi's paper, in which the raw data cannot support the performance, but the top layer can. The main reason is that our top layer size is 4, which is much less than their top layer size (400). Information is at a great loss in our top layer. Error-analysis confirms that the small layer size of top layer is largely the reason for poor performance, and increasing its layer size reliably increase the top layer's performance.

Spatial relationship regarding rotation.

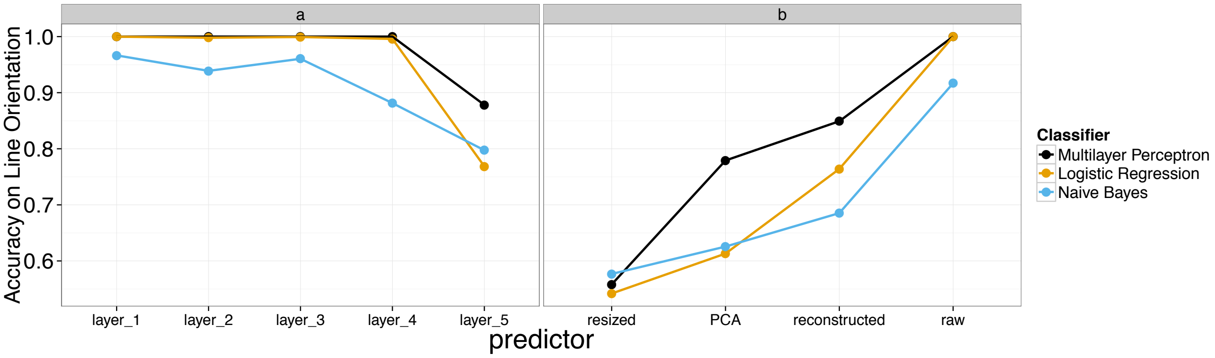

Fig. 2 Accuracy on comparison of line orientation

Although the performance of top layer is not as good as the other layers, it outperforms the resized image and the PCA reduced images (Fig. 2). In addition, the performances of all the first 4 layers are perfect, indicating that even after we reduce the input dimension by a factor of 7, the information of stimulus orientation is still well preserved. This finding suggests that the orientation-tuning found in human visual cortex might emerge from exposure to natural images.

Similar triangles

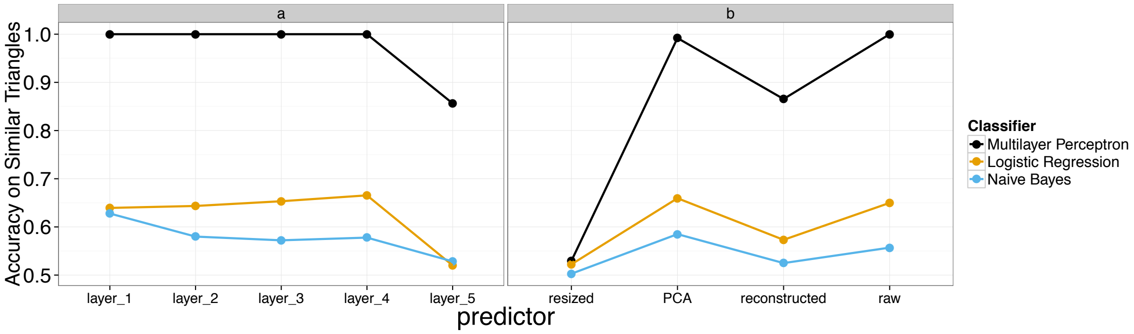

Fig. 3 Accuracy on "Similar Triangles"

The performance drops only when the top layer codes are used to train the classifiers, indicating that all the previous layers preserve the shape information of the triangles (Fig. 3). On the other hand, we find that "PCA reduced image" demonstrates higher performance than the top layer. This contradicts to the original Hinton's paper. The reason might be that they used fine-tuning to optimize reconstruction, but we did not (for the purpose of mimicking neuronal noises). Though the performance of PCA is very appealing, no extant studies showed that neural system perform PCA. Therefore, it might only serve as an upper bound for the optimality of neural system that lacks biological reality.

Across all the tasks, we find that multilayer perceptron outperforms logistic regression and Naïve Bayes. This indicates traditional read-out method in neuroscience might be insufficient to recover the stimulus selectivity in the neural codes, and introducing an extra hidden layer will help decode non-linear neural codes. Our results suggest a potential improvement of neural activity decoding to maximally recover the neurons' stimulus selectivity.

The generalization of magnitude comparison.

We find that classifiers trained to decide which line is longer do not generalize to circle size comparison (Fig. 4). This indicates that the model does not simply rely on the total number of active units in the input to compare line lengths. If it does, the classifier should generalize to circle size comparison. Therefore, the model must have used other tricks to compare line length. For instance, the network might first align one line with the other and check if the pixels composing line A is a subset of the pixels composing line B. In this case, it can answer the question “which line is longer”, but cannot compare the sizes of two circles.

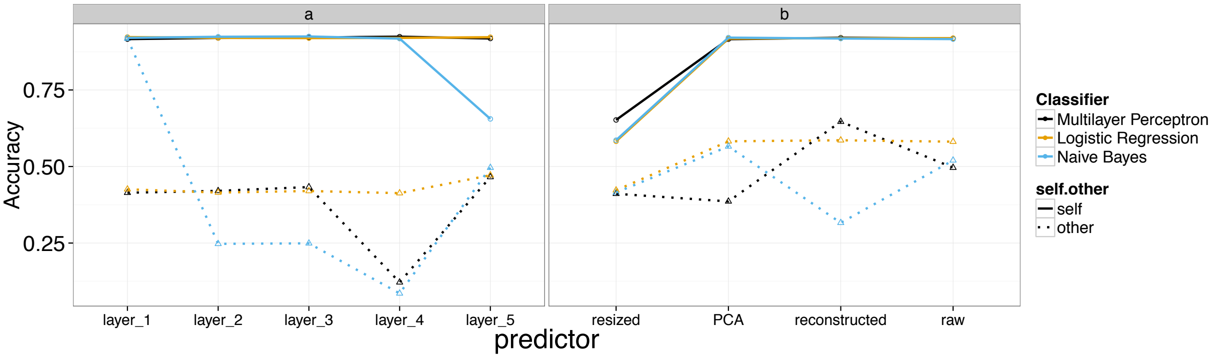

Fig. 4 Accuracy on circle size comparison. Self: classifiers trained by images of circles. Other: classifiers trained by images of lines.

The generalization of similar triangles

We found that the classifiers trained to decide whether two triangles are similar or not does not generalize to shape comparison of quadrilaterals. In other words, the classifier is able to judge whether a right triangle is similar to an equilateral triangle, but it is not able to differentiate between a non-square rhombus and a square. The only possible reason is that the model does not rely on the presence of right angle to judge the similarity of triangles. Instead, it may use other tricks, such as the relationships between the lengths of the three sides of the triangles. If this is the case, it is still able to perform the “similar triangle” task, but it will not be able to perform the “similar quadrilateral” task (since both the square and non-square rhombus are equilateral).

The generalization of rotation

Again, the model successfully learns to name the spatial relationship between two lines (whether the line in image 1 is clockwise or counter-clockwise relative to the one in image 2). This ability does not generalize to the judgement of orientations of two parallel lines. One possible explanation is that all the images are centered. Therefore, a single-line image contains a line going through the center and a double-line image contains two parallel lines located on each side of the center. The classifiers trained with single-line images might learn to focus on central information. When they are used to judge the orientation of two lines, they fail to process the peripheral information.

Visualization of response properties and learned concepts

We have shown that the model can learn various geometry concepts. We next probe the mechanisms underlying its performance. We analyze the units in the 4th hidden layer and test if there are neurons sensitive to the distances between two parallel lines. Below we present the activation of all units whose activations significantly correlate with distance ($p<.01$). We found that some neurons are more active when the distance increases, whereas others respond in the opposite way (Fig. 5.a). These units have the similar response profiles as the magnitude-sensitive parietal neurons found in electrophysiology experiments [2].

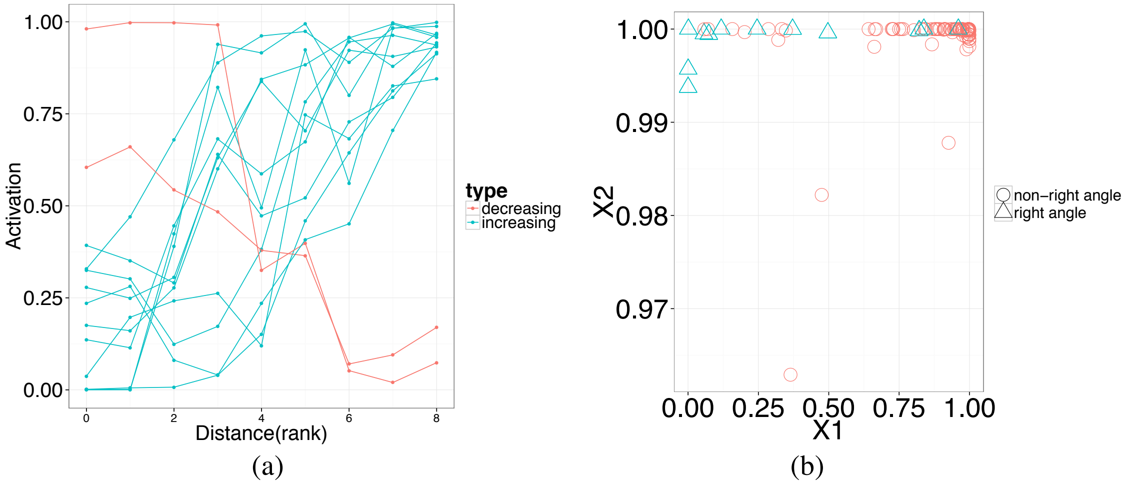

Fig. 5 (a). Distance-sensitive units in the 4th hidden layer. (b) Representations of “right angles” in the top layer

In Fig 5.b, we visualize the representation of “right angles” and “non-right angle” in the top layer. We see that the network already extracts some features of “right angles” and “non-right angle”, although the representation is very noisy. Note that this is before any supervised learning occurs. From the visualization we see that some visual features of geometry figures can be extracted without training to perform specific tasks.

Visualization of Connection Weights

To further understand how the model performs the task, we plot the learned weight between the visual input and the first hidden layer.

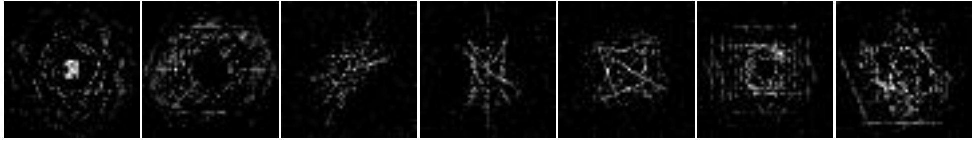

Fig. 6 Examples of connection weights between the visual input and the 1st hidden layer units.

We observe some on-center neuron (e.g., #132) and on-surround neuron (e.g., #122). These neurons may contribute to the comparison of line length, since longer lines would activate on-surround neurons more.

In addition, we found neurons that are selective to different orientations (e.g., #64 and # 83). These neurons might be useful for the comparison of orientations.

Finally, We find neurons that are selective to different angle sizes. Fig. 6 shows that neuron #99 is selective to acute angles, neuron #295 is selective to rights angles and neuron #151 is selective to obtuse angles. These neurons may contribute to the comparison of angles sizes.

Conclusions:

In this project, we train a deep neural network to encode geometry figures and find that geometry features are spontaneously extracted in the hidden layers without supervised learning. These extracted features can be used to answer various geometry questions, such as “Which angle is bigger?”, “Are these two triangles similar?”, “Is this line clock-wise or counter-clockwise relative to the other line?”. This finding suggests that the acquisition of geometry concepts may have a strong perceptual basis.

In addition, the poor generalization of learned geometry concepts reveals the challenge for geometry concepts acquisition, which has significant implication for mathematical education. When kids learn to generate the correct verbal descriptions for some geometry figures, they may not learn it in the way we expect, so the learned concepts do not spontaneously generalize to new shapes. Future study should investigate how to add extra constraints to facilitate generalization.

To summarize, the current project provides a computational framework to simulate geometry cognition with deep neural network, which bridges machine learning, cognitive psychology and computational neuroscience.

Notes: This project is done jointly with our CS221 class project. All the results presented here are counted for CS229. Results that are counted for CS221 are not included in the current report.

.

References

[1] M. Zorzi, A. Testolin, and I. P. Stoianov, “Modeling language and cognition with deep unsupervised learning: a tutorial overview.,” *Front. Psychol.*, vol. 4, no. August, p. 515, Jan. 2013.

[2] J. F. Cantlon, M. L. Platt, and E. M. Brannon, “Beyond the number domain.,” *Trends Cogn. Sci.*, vol. 13, no. 2, pp. 83–91, Feb. 2009.

[3] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman, “How to grow a mind: statistics, structure, and abstraction.,” *Science (80-.).*, vol. 331, no. 6022, pp. 1279–1285, Mar. 2011.

[4] J. M. Beck, W. J. Ma, R. Kiani, T. Hanks, A. K. Churchland, J. Roitman, M. N. Shadlen, P. E. Latham, and A. Pouget, “Probabilistic population codes for Bayesian decision making.,” *Neuron*, vol. 60, no. 6, pp. 1142–52, Dec. 2008.

[5] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks.,” *Science* , vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[6] A. Testolin, I. Stoianov, M. De Filippo De Grazia, and M. Zorzi, “Deep Unsupervised Learning on a Desktop PC: A Primer for Cognitive Scientists.,” *Front. Psychol.*, vol. 4, no. May, p. 251, Jan. 2013.