Developing a Natural Method to Measure the Complexity of Language and Applications to Color Naming

Mentors: Amrut Nadgir, Jeremy Yeung

Students: Harshika Jalan, Emilyne Kim, Jiaxun (Aaron) Li, Ethan Qiu, Srinath Rangan

Abstract

We propose a method to apply neural network models to measure the complexity of different aspects of language. This method is applied to color naming systems in languages across the globe and compared to an existing method of measuring the complexity of color naming systems using mutual information. In particular, we measure the complexity of a language by finding the smallest possible neural network that can adequately represent the naming system within a given error bound. The size of a neural network is quantified by counting the number of hidden nodes in the network, and is also known as neural complexity. Surprisingly, we discovered that languages with high mutual information required fewer nodes to accurately represent.

I. Measuring Complexity Using Neural Networks

Information Theory provides a universal measure of complexity over probability distributions[1]. However, in many real-world applications, it is often difficult to define explicit probability distributions for the information in the task at hand. In particular, there is no current measure of the complexity of various behaviors and decisions made by the brain, such as the complexity of different languages.

In neuroscience and linguistics, neural networks are commonly used to model the relevant structure and dynamics in the brain when performing certain tasks. It is often insightful to compare and classify the "neural resources" used by the brain to perform various tasks by training a neural network model of the brain and analyzing how it solves the problem. To this end, various measures of task complexity have been sought out by researchers in the field. In dynamical models of the brain, such as Recurrent Neural Networks and Spiking Neural Networks, a promising direction of research consists of estimating task complexity by training a neural network to perform the task and analyzing the dynamics of the system while it is performing the task[6]. In standard perceptron networks, where there are no such dynamics to analyze, it is often more insightful to look at the structure of the trained networks.

To build on the recent overwhelming successes of neural network models in the space of linguistics, such as GPT3 by OpenAI, we propose a method to apply neural network models to measure the complexity of different aspects of language. This method is applied to color naming systems in languages across the globe and compared to an existing method of measuring the complexity of color naming systems using the information bottleneck principle[3]. In particular, we characterize the complexity of an aspect of language in a given language by finding the smallest possible neural network that can adequately represent the system within a given error bound. The size of a neural network is quantified by counting the number of hidden nodes in the network and is also called the neural complexity of the model[5].

II. Color Naming Systems

The dataset used in this study is downloaded from the World Color Survey[3] Palette, where each color chip is uniquely identified by its row (lightness value) and column (Munsell hue). These color chips are distributed in a 3D CIELAB[8] color space, where the similarity between two colors are inversely correlated with the distance between them (Figure 1). The distribution of colors in this space is a joint Gaussian distribution between the color chips which can be decomposed into multiple independent individual Gaussian distributions centered at the coordinates of each color chip. These distributions have a standard deviation of eight, which represents how dissimilar two colors need to be before they can be distinguished by the naked eye.

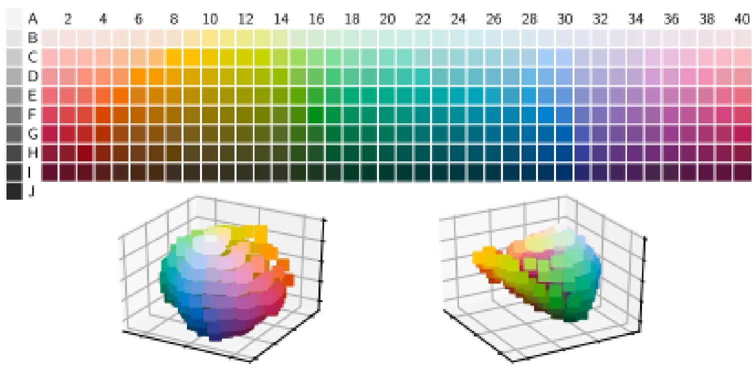

Figure 1: (Top) The various color chips organized by lightness and hue. (Bottom) The same color chips embedded in a 3D CIELAB color space. This figure was taken from *Efficient compression in color naming and its evolution* by Zaslavsky et. al.[4]

With this color space and data on what terms different languages use to describe each color chip, we are able to construct the relevant probability distributions as follows: in a particular language, there's a many-to-many relationship between the color chips and the terms used describe each color. Based on the number of people in the survey who associated a color chip with a given color name, we define a conditional probability distribution over the color names that describes the probability a speaker will use a color name to describe a given color chip. In mathematical terms, if W is a random variable that represents color names and M similarly represents the color chip, we are able to define $p(W|M)$ using survey data. After calculating the distributions within this mapping, the complexity of a language is typically defined as the mutual information in bits between the distribution of color chips in the similarity space and their associated names[3]:

$$I(W;M) = \sum_{m,w} p(M=m)p(W=w|M=m)\log_2 \frac{p(W=w|M=m)}{p(W=w)}$$ and

$$p(W=w) = \sum_{m} p(W|M=m)$$.

III. Our Methodology

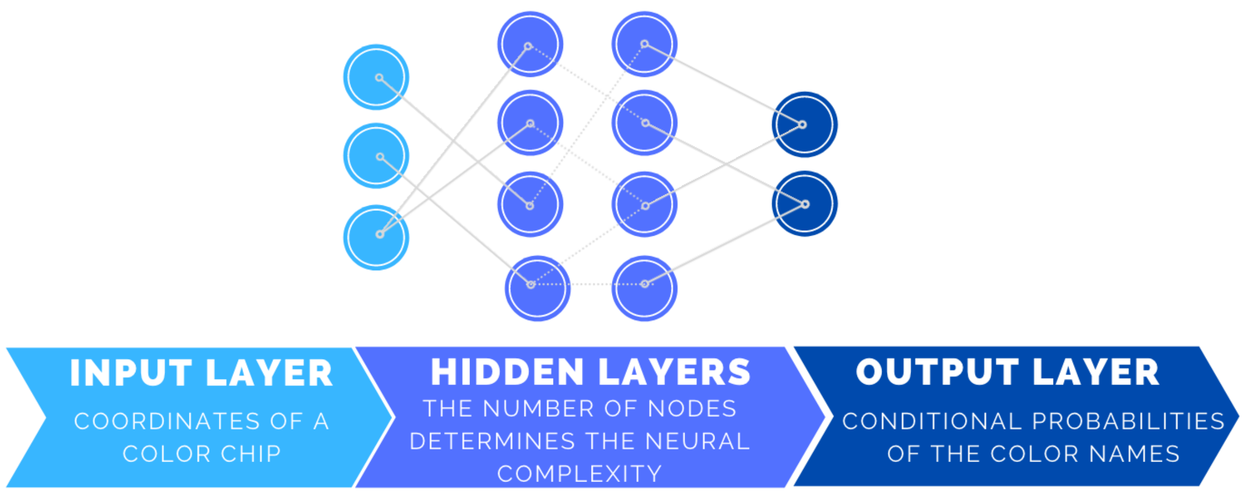

Figure 2: A visual representation of the inputs and outputs to the neural network models.

To model various color naming systems using neural networks, we trained them to learn the conditional probabilities, $p(W|M)$. A paradigm was chosen where the networks were trained to take the coordinates of a color chip in a normalized 3D CIELAB color space as inputs and were asked to correctly output the conditional probability of each color name given the chip input (Figure 2). The input coordinates were chosen from the normalized axes of a 3D CIELAB color space because similar colors will have similar coordinates which will make the learning process more efficient.

The networks that were trained vary in size from one hidden node to 24, inclusive, and can have between one and three hidden layers. They have fully connected layers with a bias and utilize a ReLU activation function. It is trained using gradient descent based methods by using the Adam optimizer in the PyTorch library. The World Color Survey data was randomly split into a training and test set for each network trained, where the training set consisted of 80% of the color chips.

We define the term, *shape*, to describe the number of layers and nodes per layer of a network. For each network shape that was tested, ten neural networks were trained of that given shape with random weight initializations and different training/test sets. The final error of a network was defined by its $l1$ error on the test set. For a given network shape, the associated error was an average of that of the ten networks trained and tended to have a standard deviation one-to-two orders of magnitude smaller than the associated error. For a given network size, the error was the minimum of that of the different shapes with that particular number of total hidden nodes. The standard deviations associated with the errors

To estimate the complexity of a particular color naming system, we defined error thresholds that ranged from zero to one with a spacing of 0.001. No network size had an associated error greater than one and the majority were around the 0.05 range. For each threshold, we computed the smallest network size which had an error below that threshold. If no such network existed, then the size was set to 24. Our measure of complexity was defined as the average of these sizes. In more intuitive terms, the estimated complexity of a color naming system is associated with the number of hidden nodes required to learn that system with error below a certain threshold.

IV. Results

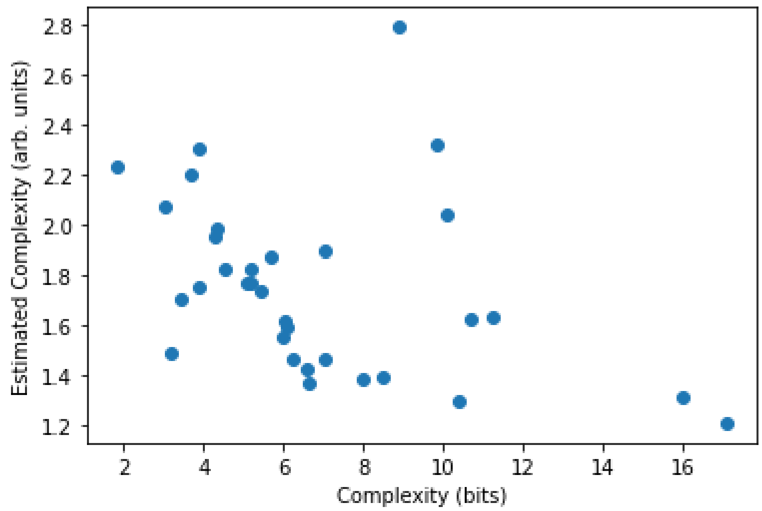

Figure 3: Scatterplot of the estimated complexity using neural networks and the complexity measured by mutual information in bits. Both measures are inversely correlated.

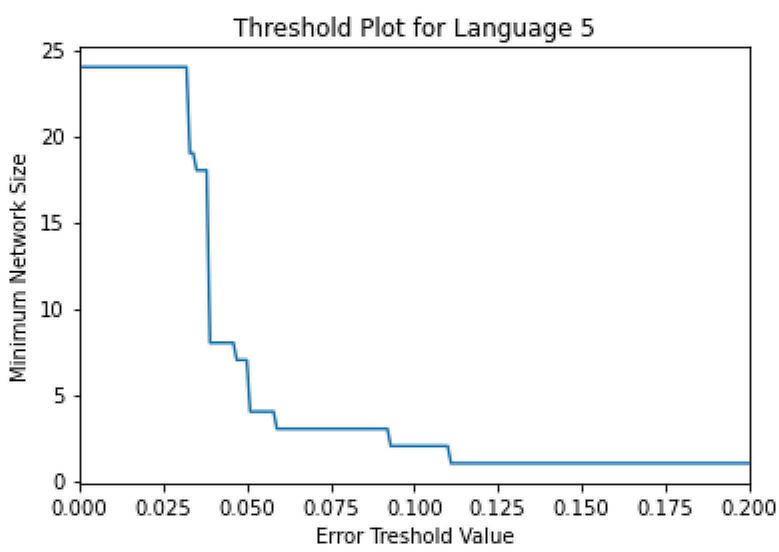

Figure 4: As the threshold error value increases, the smallest network that can model the color naming system of language 5 decreases in size. As expected, larger networks are required for lower error.

When plotting the estimated complexity of various color naming systems using our neural network method against their complexities in bits, we surprisingly found an overall downwards trend with some expected scatter (Figure 3). In addition, there are a few outlier languages which also follow this downward trend but have a much steeper decline. This result indicates the languages that had a higher mutual information between the color chips and color names actually required fewer nodes to learn, on average, than those that had a lower mutual information.

As a sanity check, we confirmed that the minimum network size required to model a given language below a certain error threshold will decrease as the threshold increases. This relationship is visualized for language 5 in Figure 4.

We initially hypothesized that if a color naming system is complex in the sense that it has a high mutual information, a larger neural network would be required to learn the associated conditional probabilities within a given error threshold. The reasoning is that it is unlikely for complex linguistic rules to be translated into extremely simple neural networks. However, we found the opposite to be true. The more complex languages were actually more learnable!

V. Conclusion and Future Work

The major striking result from this study is that more complex color naming systems actually require fewer nodes to learn! There are a few hypotheses that might explain why. One is the prevalence of soft versus hard category boundaries in the naming systems. While softer boundaries and somewhat inconsistent naming systems tend to have a lower complexity, as defined by mutual information, than harder boundaries, it could be the case that hard boundaries require less resources for a neural network to learn. To test this hypothesis, one could imagine constructing artificial languages with different types of boundaries and comparing the two complexity measures applied to these languages. In addition, it could very well be possible that the more complex man-made languages share a different property that make them easier to learn. To tease out what such a property could be, one could also apply these conflicting complexity measures to artificial languages.

In our proposed method of estimating complexity, we chose to count the number of hidden nodes in the network for simplicity. However, this method leaves out crucial information about the structure of the particular networks that were successful. To improve our measure, one could perform a more thorough analysis of the various networks. In particular, the complexity of deep neural networks can actually be analyzed using the information bottleneck principle[7].

Our training method uses gradient descent, a method that is useful but may get stuck on local minimums. Professors Pilanci and Ergen's method published in *Neural Networks are Convex Regularizers: Exact Polynomial-time Convex Optimization Formulations for Two-layer Networks*[9] offers a method to find the global minimum for fitness functions for two-layer networks by transforming the training process into a convex optimization problem. The authors also claim the algorithm can be generalized to an arbitrary number of hidden layers. If we are able to apply this algorithm to our training process, we can be more certain that the smallest networks for a given threshold are not overparameterized.

In addition to estimating the complexity of individual languages, our technique can be applied to evaluate the similarity of unique languages by training minimal complexity networks to recognize color naming conventions in multiple languages simultaneously. The difference between the summed network complexity required to model each language individually and the network complexity required to model both simultaneously can act as a measure of how similar two languages are. Such a measure can be used to trace the evolution of language as well as the spread of culture across the globe.

Broadly speaking, we have introduced a method to measure complexity in linguistics that is based on the human brain. We found that this method conflicts with existing measures and proposed hypotheses as to why.

References

1. A Mathematical Theory of Communication—Shannon—1948—Bell System Technical Journal—Wiley Online Library. (n.d.). Retrieved December 2, 2020, from https://onlinelibrary.wiley.com/doi/10.1002/j.1538-7305.1948.tb01338.x
2. Regier, T., Kemp, C., & Kay, P. (n.d.). Word meanings across languages support efficient communication. 22.
3. Kay, Berlin, Maffi, Merrifield, Cook (2009). The World Color Survey. Stanford: CSLI (ISBN (Cloth): 9781575864150).
4. Zaslavsky, N., Kemp, C., Regier, T., & Tishby, N. (2018). Efficient compression in color naming and its evolution. Proceedings Of The National Academy Of Sciences, 115(31), 7937-7942. doi: 10.1073/pnas.1800521115.
5. Kon M. A., Plaskota L. (2000). Information complexity of neural networks. http://math.bu.edu/people/mkon/nn30.pdf.
6. Maheswaranathan N., Williams A. H., Gollub M. D., Ganguli S., Sussillo D. (2019). Universality and individuality in neural dynamics across large populations of recurrent networks. https://arxiv.org/pdf/1907.08549.pdf.
7. Tishby N., Zaslavsky N. (2015). Deep Learning and the Information Bottleneck Principle. https://arxiv.org/pdf/1503.02406.pdf.
8. Durmus D. (2020). CIELAB color space boundaries under theoretical spectra and 99 test color samples. Wiley Online Library. https://onlinelibrary.wiley.com/doi/abs/10.1002/col.22521.
9. Pilanci M., Ergen T. (2020). Neural Networks are Convex Regularizers: Exact Polynomial-time Convex Optimization Formulations for Two-Layer Networks. https://stanford.edu/~pilanci/papers/NNConvex.pdf.