JOURNAL OF
EMERGING INVESTIGATORS

Article

Identifying Neural Networks that Implement a Simple Spatial Concept

Rayhan Zirvi[1], Kenneth Kay[2]
[1]River Hill High School, Clarksville, Maryland
[2]Center for Theoretical Neuroscience, Columbia University, New York City, New York

SUMMARY

Modern artificial neural networks have been remarkably successful in various applications, from speech recognition to computer vision. However, it remains less clear whether they can implement abstract concepts, which are essential to generalization and understanding. To address this problem, we investigated the above vs. below task, a simple concept-based task that honeybees can solve. We hypothesized that neural networks would successfully solve this task, and that performance would vary substantially between network architectures. Specifically, we predicted that the convolutional neural network (CNN), a prototypical architecture well known for its ability to classify objects accurately, would perform better than the single-layer and multi-layer perceptrons (SLP and MLP, respectively). In the first task (Experiment 1), a visual target was presented above or below a black bar; in the second (Experiment 2), a visual target was presented above or below a reference shape. We found that all networks achieved 100% testing accuracy on Experiment 1. In contrast, the networks' accuracy differed substantially in Experiment 2. The SLP had only a 50% testing accuracy, and the CNN outperformed the MLP (98% vs. 81% accuracy, respectively). Further analysis of the connection weights and distances between shapes suggested that the MLP may not evaluate relative spatial relationships in Experiment 2. Instead, the network seemed to partition the image into upper and lower zones, which appears inconsistent with the concept of relative locations. These findings indicate different capacities of network architectures, offer insight into their mechanisms, and motivate work on how neural systems implement conceptual knowledge.

INTRODUCTION

The use of neural networks to implement and model intelligence appears promising but may be limited by the narrow scope of tasks known to be solvable by these systems (1). That is, when tested for generalization on data outside of the training distribution, these networks often fail or even are logically bound to fail (1). For example, an artificial system programmed or trained to play tic-tac-toe will not be able to transfer its knowledge and experience to another game, such as chess. Importantly, a possible reason for this limitation is that such systems do not understand or effectively implement concepts (2).

The advantage of studying artificial neural networks, in contrast to the biological neural networks in the brain, is that these systems are mathematically formalized (i.e., made up of a collection of interacting neuron-like units) and thus can be implemented, simulated, and monitored on a computer. Artificial neural networks can consist of billions of neurons and can be structured into layers (2). Typically, there is an input layer, a variable number of hidden layers, and an output layer (3). These hidden layers perform typically non-linear calculations on the inputs and then yield output activity (3). Each neuron has an activation value and each connection from a given layer to the next layer has a weight parameter corresponding to the connection strength (3). These parameters determine how neurons contribute to the downstream connected neurons, and so on. For many artificial networks, the network learns in a process by which the error at the output layer is calculated and then back-tracked through the network to determine how to change parameters at each layer and unit, an algorithm called backpropagation (3, 4). Generally, a neural network is trained on an initial dataset and then given a test dataset on which it has not previously been trained in order to simulate a real-world problem requiring a general form of knowledge. The definition of the network's performance is the ability to respond correctly to these test instances, an ability otherwise known as "generalization" (4).

Toward addressing the limitations of artificial neural networks, one approach is to identify how actual brains, made up of biological neural networks, achieve cognitive abilities. By studying how the human brain achieves cognition, we can develop models for how neural networks might accomplish such abilities. We took this approach by developing artificial neural networks that model the conceptual ability of honeybees as a simple system.

A recent experimental study of honeybees found that they can identify the above vs. below relationship in a set of familiar stimuli and generalize this knowledge to new stimuli (5). In the study, Avarguès-Weber *et al*. tested honeybees on an above vs. below concept task (5). In the first version of the task, honeybees were required to recognize whether a shape is above or below a horizontal black bar, and in the second version of the task, honeybees were required to recognize whether a shape is above or below another shape. In both versions, each honeybee was presented

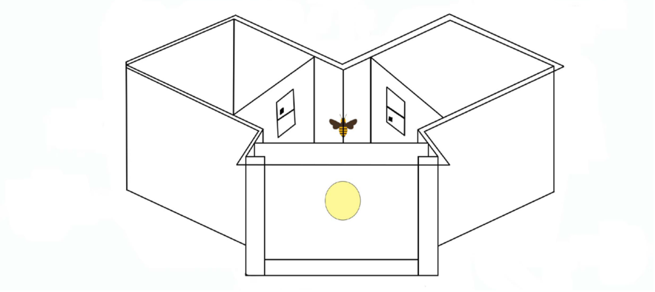

Figure 1. Y-maze (decision maze) with a honeybee presented with two images. The left image demonstrates the 'above' relationship and the right image demonstrates the 'below' relationship. The honeybee then chooses an image and follows that path in the maze. From there, it is either rewarded with a sucrose (table sugar) solution or penalized with a quinine (bitter compound) solution.

with two images in a y-maze (**Figure 1**). The honeybee then chose an image and followed the corresponding path in the maze. After many training trials, the bees performed a transfer test with the same tasks, but different shapes in the images. In the transfer tests, bees chose the correct image with 70% accuracy (5). This finding was important because it was an instance of conceptual learning in a small brain (only about 1 million neurons, compared to 86 billion neurons in the human brain) (6). In this respect, bees may be distinctive among invertebrates, which usually have less complex, centralized, and specialized nervous systems than vertebrates (7). Indeed, the fact that honeybees were able to solve this complex conceptual task challenges the view that a large brain is needed for abstract cognition (8). The bees not only needed to recognize the shapes on the images but also understand how their positions are related. This suggests that neural networks, which have a similar number of neurons to honeybees, could potentially be capable of learning and categorizing abstract concepts as well.

The authors of the above vs. below honeybee study claimed that conceptual ability was indicated by the bees' performance (5). Our study expanded upon this work by determining whether it is possible to construct a neural network capable of performing the above vs. below task. As with the honeybee study, we sought to determine whether classic neural network architectures can effectively solve the above vs. below task and to understand the networks' strategy for solving the task. We tested single-layer perceptrons (SLP), multi-layer perceptrons (MLP), and convolutional neural networks (CNN). MLPs have hidden layers, but SLPs do not. CNNs have several convolutional layers (containing filters) and pooling layers that scan for features at various locations in the image.

We hypothesized that the neural network architectures will differ significantly in testing accuracy in the above vs. below concept task. In particular, we predicted the CNN, which is designed to process visual data and recognize images, will perform the best out of all the networks. Our reasoning was that the convolutional and pooling layers of CNNs could plausibly help these networks identify spatial relations between objects (9). Thus, we predicted the known advantage of CNNs in image recognition will extend to this task of spatial relationship recognition. In the first task (Experiment 1), we displayed images of a visual target above or below a constant black bar; in the second (Experiment 2), we displayed images of a visual target above or below a reference shape. After training the neural networks, we found that the CNN had the highest performance with 100% testing accuracy and 98% testing accuracy on the two tasks respectively, consistent with this hypothesis. The MLP achieved 100% testing accuracy on the first task, but only 81% testing accuracy on the second. These findings revealed differences between neural network models and suggested further analysis of the networks' mechanisms to clarify these differences.

RESULTS

As with the previously published experimental study about honeybees that tested their ability to learn and apply concepts, we performed two experiments to test the ability of neural networks to conceptualize above vs. below (5). Experiment 1 investigated the ability of neural networks to identify a target shape above or below a constant black bar, while Experiment 2 investigated the ability to identify a target shape above or below a reference shape (**Figure 2**). 60,000 training images and 10,000 testing images displaying these relationships were randomly generated for both experiments. Each image with a target shape displaying the 'above' spatial relationship was paired with a corresponding image with the same target shape displaying the 'below' spatial relationship.

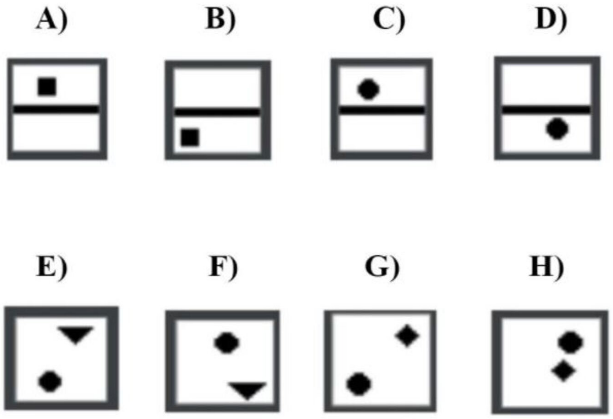

Figure 2. Representative training and testing images for Experiment 1 and 2. 60,000 training images and 10,000 testing images were randomly generated for both experiments. Figures 2A–2D are for Experiment 1. Reference is the black bar for all images in Experiment 1. A) Training dataset target (square) above reference. B) Training dataset target (square) below reference. C) Test dataset target (circle) above reference. D) Test dataset target (circle) below reference. Figures 2E–2H are for Experiment 2. Reference is the circle for all images in Experiment 2. E) Training dataset target (upside-down triangle) above reference. F) Training dataset target (upside-down triangle) below reference. G) Test dataset target (diamond) above reference. H) Test dataset target (diamond) below reference.

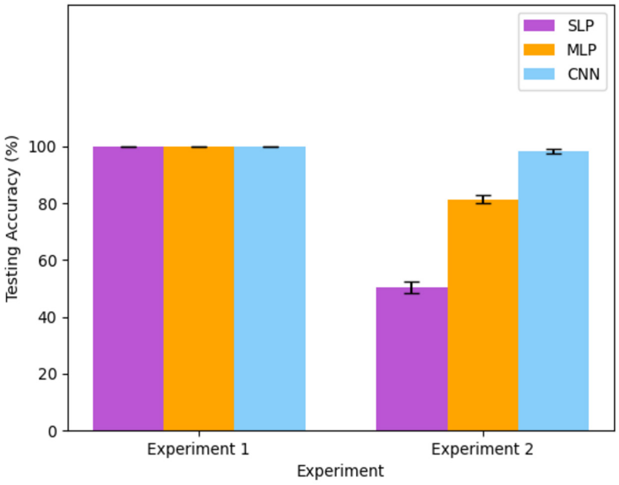

Figure 3. Testing accuracies for different neural networks in Experiment 1 and 2 (n=10). The mean testing accuracy over 10 trials for each neural network is presented. Error bars present the standard error. SLP = single-layer perceptrons, MLP = multi-layer perceptrons, CNN = convolutional neural network.

We tested three network architectures: SLP, MLP, and CNN. Each network was run for 10 iterations of training and testing in both experiments.

Experiment 1: All networks perform with full accuracy

In Experiment 1, the MLP, CNN, and SLP neural networks were all able to successfully solve the task with perfect testing accuracies of 100% (**Figure 3**). Since the SLP was able to solve the task, only a single linear transformation of the inputs was needed for separating the two choices of spatial relation (above vs. below). To understand how these networks solved the task, we visualized the connection weights (parameters representing the strength of connections between units) from input layer neurons (pixels) to single neurons within each network (**Figure 4**). Plots of connection weights in the MLP showed clustering of positive and negative weight in the upper vs. lower zones (**Figure 4A**). A similar pattern could be seen in the SLP with a more well-defined clustering of weights (**Figure 4B**).

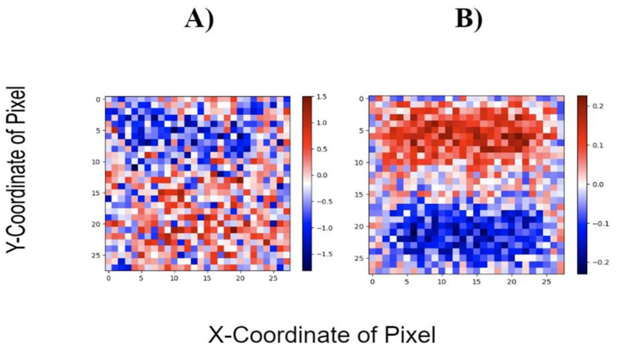

Figure 4. Examples of neural network connections in Experiment 1. The connection weights are shown for each pixel in the input images. A) Map of connection weights from the input (pixels) to one hidden layer neuron for the multi-layer perceptrons. B) Map of the connection weights from the input (pixels) to one output neuron for the single-layer perceptrons.

Experiment 2: CNN outperforms MLP

In Experiment 2, the CNN outperformed the MLP, achieving approximately 16% higher testing performance (MLP: 81.4%, CNN: 98.3%) (**Figure 3**). Importantly, the SLP had only random chance performance (50.3%), further suggesting a difference between task versions. The performances across instances of each neural network architecture differed significantly (Kruskal-Wallis test, $p < 0.0001$), and the testing accuracies of the SLP vs. MLP ($p = 0.008$), SLP vs. CNN ($p = 0.0000004$), and MLP vs. CNN ($p = 0.009$) were all significantly different (post-hoc Dunn's test, $p < 0.01$).

To gain insight into how the networks performed the task, we analyzed the connection weights of the MLP, which performed both task versions successfully. Similar to the first task version, the connection weights indicated that at least several of the hidden layer neurons with strong output weights divided the image into above vs. below zones (**Figure 5**). To investigate further, we analyzed the absolute y-distances (vertical differences in pixels) between the target and reference shapes of incorrectly identified images vs. correctly identified images. A histogram of the distances showed that correctly identified images had generally larger y-distances than incorrectly identified images (**Figure 6**).

DISCUSSION

We investigated the capacity of neural networks to perform simple conceptual tasks, focusing on the above vs. below relationship, a concept that can be understood by honeybees. Our results demonstrated that CNNs and MLPs could correctly identify the spatial relationship of above vs. below by solving the tasks outlined in Experiment 1 and 2. Interestingly, in Experiment 2, the CNN performed with a considerably higher accuracy than the MLP. The SLP was also able to fully solve Experiment 1 but performed at random chance levels in Experiment 2.

This study suggested two insights. First, the ability to tell whether a visual object is located above vs. below a constant bar can be simple, as demonstrated by the SLP model achieving a perfect performance in Experiment 1. This indicated that the challenge of simplistic image classification in a biological setting may not be in the analysis of the image array. Instead, the challenge may be for a subject (e.g., a honeybee) to approach and attend to the relevant visual objects (in a world full of other stimuli). Once a subject does this, solving the spatial concept can become much simpler.

Second, important architectural differences between models were indicated in the more difficult version of the task (Experiment 2), in which CNNs outperformed MLPs. Additional analyses performed on the MLP suggested that the network at least partially relied on the absolute positions of visual stimuli (both the reference and target shapes) to solve the task. In contrast to MLPs, CNNs and deeper networks in general

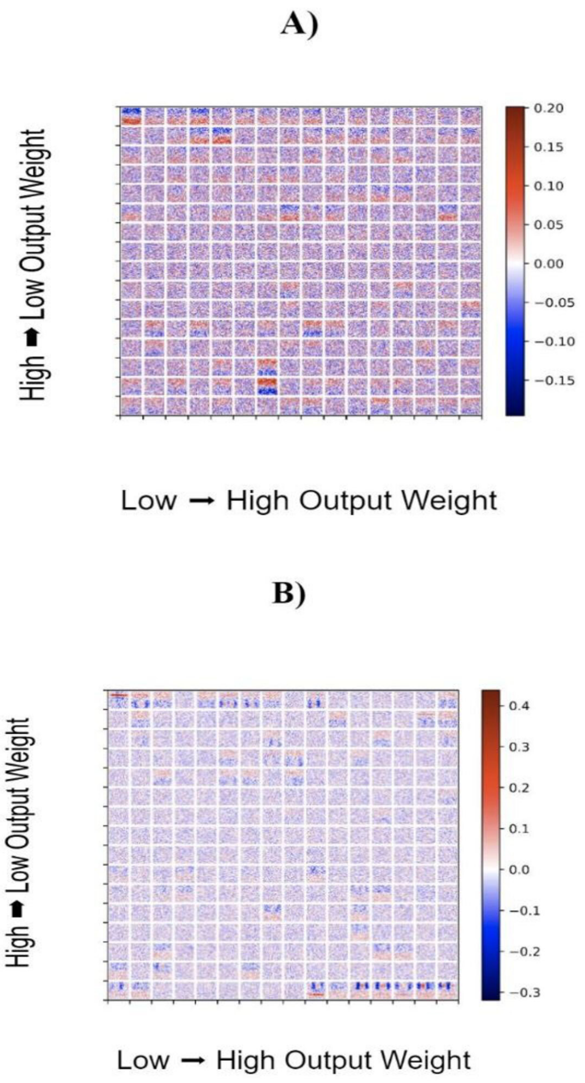

Figure 5. Collection of 256 (28x28 grid) heatmaps of multi-layer perceptrons connection weights from input layer neurons (pixels) to a single hidden layer neuron. Heatmaps are sorted in increasing order (by row, then by column) by the weight from the hidden layer neuron to the output neuron. The heatmaps near the top left represent the smallest (most negative) weights, and the heatmaps near the bottom right represent the largest (most positive) weights. Each individual heatmap represents the connection weights from the input layer neurons (pixels) to a single hidden layer neuron. A) Heatmaps from Experiment 1. B) Heatmaps from Experiment 2.

are known to be able to distinguish between visual objects irrespective of position in the input image (9). This is made possible by the specialized architecture of the CNNs (e.g., convolution and pooling layers) and additional layers in these networks. These same components of CNNs may plausibly be used to identify spatial relationships between objects. This possibility is consistent with the present findings, though the task tested here is not equivalent to object recognition tasks typically tested on CNNs.

Perhaps the most important caveat in these experiments is that the neural networks might not fully need to determine relative position (the concept tested by the task), and instead rely on absolute positions. This was clearly the case in Experiment 1 since the simple SLP, which has no hidden layers, was able to achieve a high accuracy. Plots of connection weights further revealed that the SLP used a simple strategy in which images were analyzed by upper vs. lower zones (**Figure 4B**). More generally, neural networks can overfit training data, leading to diminished performance on test data; thus, simple alternative interpretations of networks' success on a task should be considered.

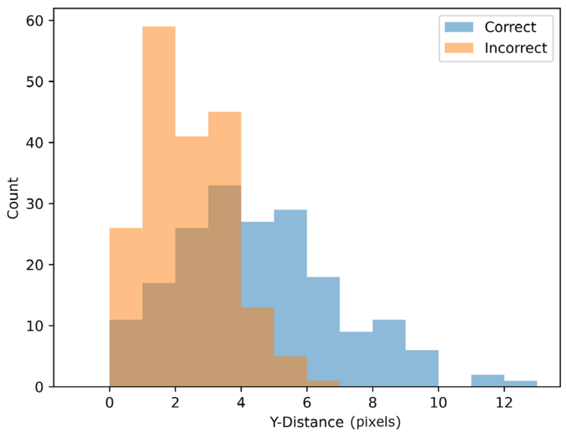

Figure 6. Absolute y-distances (vertical differences in pixels) between the centers of target and reference shapes for the multi-layer perceptrons in Experiment 2. Note that y-distances are larger for correctly identified images compared to incorrectly identified images.

Future investigation of these networks can clarify their underlying mechanisms and thus their differences in performance. Furthermore, these results suggest that other simple conceptual tasks can be meticulously studied using artificial networks. Better understanding of these systems can potentially shed light on how actual brains work. Modeling and understanding the brain may guide the future of artificial neural networks, ultimately generating opportunities for further insights.

MATERIALS AND METHODS

Three neural network architectures were trained in these experiments: SLP, MLP, and CNN. The SLP model contained a single linear transformation with no hidden layers. The MLP model contained a single hidden layer with 256 neurons with a Rectified Linear Unit (ReLU) activation function. Both models had an output layer of two neurons with a Softmax activation function. The SLP and MLP had 1,570 and 100,738 parameters, respectively. As such, the MLP model represented a significant increase in complexity over the SLP. The CNN model contained three convolutional layers, each of which used a 3x3 kernel or filter, and applied a ReLU activation function. Between each pair of convolution layers, a simple 2x2 max pooling layer was used to reduce size. This was flattened and fed to a dense layer with 256 neurons and a ReLU activation function. Finally, the output layer used Softmax activation on two neuron outputs, as in the other models. Effectively, the CNN model utilized a small convolutional stack to process and reduced the input before feeding it into a model form equivalent to the MLP. However, because the max pooling reduced the size of the tensor representation, the final CNN model had 169,625 parameters. Each model was trained over 10 epochs. In both experiments, the neural networks were programmed in Python using the PyCharm Integrated Development Environment (IDE). The open-source software library TensorFlow with the Keras API was used to build, train, and test the neural networks (10, 11).

Each training and testing image consisted of a 28x28 pixel grid with a grayscale color range. The reference placement on the image was randomized as well as the target's location. Each image with a given target above the reference was paired with a corresponding image with the same target below the reference (**Figure 2**). The images were randomly shuffled during training to ensure the model focused on generalizable learning instead of learning tailored to specific images. Each model for both experiments included training on 4–5 target shapes and testing on 1 novel target shape. The experiment utilized a training set of 60,000 images and a testing set of 10,000 images. Each model was trained and tested 10 times on randomized data.

In Experiment 1, each image consisted of a target shape placed above or below a reference black bar. The three neural networks were trained to identify the correct relation based on two types of images, one with 'target above bar' spatial relation and the other with 'target below bar' spatial relation (**Figure 2A–2D**). The target was randomly selected out of 6 different shapes (square, diamond, 0° normal triangle, 180° upside-down triangle, rectangle, circle) with 30-36 pixels in area, whereas the reference was kept constant as a black bar.

In Experiment 2, each image consisted of a target shape placed above or below a constant reference shape. Like in Experiment 1, the three networks were trained to identify the correct relation based on two images, one with 'target above reference' spatial relation and the other with 'target below reference' spatial relation (**Figure 2E–2H**). The target varied among 5 different shapes (square, diamond, 0° normal triangle, 180° upside-down triangle, rectangle) with 30-36 pixels in area, whereas the reference's constant shape was a circle.

We applied a Kruskal-Wallis test on Experiment 2 with a significance level (α) of 0.01 to determine if there was a significant difference in the testing accuracies of the different neural networks ($df = 2$, $H = 25.8$, $\eta^2 = 0.88$). Then, we ran a post-hoc Dunn's test with a Bonferroni correction and significance level (α) of .01 to account for multiple comparisons between the SLP vs. MLP, SLP vs. CNN, and MLP vs. CNN pairs. To perform these statistical tests, we used the open-source scientific computation library *SciPy* (12).

The code used in this study can be found at: https://github.com/rzirvi1665/Neural-Networks-Above-Vs-Below.git

Received: February 7, 2022
Accepted: July 17, 2022
Published: September 13, 2022

REFERENCES

1. Garnelo, Marta and Murray Shanahan. "Reconciling Deep Learning with Symbolic Artificial Intelligence: Representing Objects and Relations." *Current Opinion in Behavioral Sciences*, vol. 29, Oct. 2019, pp. 17–23. doi:10.1016/j.cobeha.2018.12.010.
2. LeCun, Yann, et al. "Deep Learning." *Nature*, vol. 521, no. 7553, May 2015, pp. 436–44. doi:10.1038/nature14539.
3. Yang, Guangyu Robert and Xiao-Jing Wang. "Artificial Neural Networks for Neuroscientists: A Primer." *Neuron*, vol. 109, no. 4, Feb. 2021, p. 739. doi:10.1016/j.neuron.2021.01.022.
4. Rumelhart, David E., et al. "Learning Representations by Back-Propagating Errors." *Nature*, vol. 323, no. 6088, Oct. 1986, pp. 533–36. doi:10.1038/323533a0.
5. Avarguès-Weber, Aurore, et al. "Conceptualization of Above and Below Relationships by an Insect." *Proceedings of the Royal Society B: Biological Sciences*, vol. 278, no. 1707, Mar. 2011, pp. 898–905. doi:10.1098/rspb.2010.1891.
6. Giurfa, M., et al. "The Concepts of 'sameness' and 'Difference' in an Insect." *Nature*, vol. 410, no. 6831, Apr. 2001, pp. 930–33. doi:10.1038/35073582.
7. Cope, Alex J., et al. "Abstract Concept Learning in a Simple Neural Network Inspired by the Insect Brain." *PLOS Computational Biology*, vol. 14, no. 9, Sept. 2018, p. e1006435. doi:10.1371/journal.pcbi.1006435.
8. Saxe, Andrew M., et al. "A Mathematical Theory of Semantic Development in Deep Neural Networks." *Proceedings of the National Academy of Sciences*, vol. 116, no. 23, June 2019, pp. 11537–46. doi:10.1073/pnas.1820226116.
9. Haldekar, Mandar, et al. "Identifying Spatial Relations in Images Using Convolutional Neural Networks." *ArXiv:1706.04215 [Cs]*, June 2017. doi:10.48550/arXiv.1706.04215.
10. Abadi, Martín, et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." *ArXiv:1603.04467 [Cs]*, Mar. 2016. doi:10.48550/arXiv.1603.04467.
11. Chollet, François, et al. *Keras*, 2015. GitHub, github.com/keras-team/keras.
12. Virtanen, Pauli, et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." *Nature Methods*, vol. 17, no. 3, 3, Mar. 2020, pp. 261–72. doi:10.1038/s41592-019-0686-2.