Transferable Learning in Convolutional Neural Networks

Psych 1401: Computational Cognitive Neuroscience

Gabe Grand, Joe Kahn, Tomás Reimers

May 2016

Honor Code Statement
We affirm our awareness of the standards of the Harvard College Honor Code.

1 Introduction

Recent advances in deep learning have catapulted multi-layer neural networks to the forefront of the machine learning and computer vision communities. In the past few years, deep neural networks have made remarkable strides in multiple domains, including object recognition, speech recognition, and control (reviewed in LeCun et al., 2015 and Lake et al., 2016). In particular, convolutional neural networks ("convnets," LeCun et al., 1989) have now reached human-level performance on object-recognition tasks (He et al., 2015; Russakovsky et al., 2015).

Despite the rapid pace of progress in deep learning, researchers have only recently begun to consider the question of whether neural networks constitute reasonable models of human cognition beyond limited, superficial similarities to biological neural networks. One key component of human perceptual learning is the ability to rapidly generalize acquired knowledge across categorical domains. Neurological and psychophysical studies confirm that humans quickly learn to recognize novel stimuli, in some cases after just a single exposure (Rutishauser et al., 2006; Carey and Bartlett, 1978; Pinker, 1999). In contrast, modern neural networks require thousands of training iterations on vast datasets in order to match human capacity (Russakovsky et al., 2015).

This discrepancy raises an interesting dilemma. Given that randomly-initialized neural networks must start from zero previous visual experience, while humans typically draw on an extensive body of perceptual knowledge, is it fair to directly compare the performance of humans and neural nets on the same recognition tasks?

In order to shed light on this issue, we conducted a systematic study of neural nets' ability to generalize perceptual knowledge from one object classification task to another. Our research was guided by three main experimental questions:

- To what extent does learning transfer from one classification task domain to another?
- At what layer(s) of the network is transferable knowledge stored?
- What are the layerwise temporal dynamics of learning? In other words, does learning occur in a bottom-up or top-down manner?

For each of these questions, we designed and performed a computational experiment. In Section 3.1, we found that the process of learning can be greatly accelerated through pre-training. These results suggest that the network stores some degree of generalizable perceptual knowledge that can be transferred across image domains. In Section 3.2, we "lesioned" different layers of the net in order to determine where in the network transferable knowledge is stored. We found that lesioning higher layers of the network impaired performance,

while lesioning low layers generally had minimal impact. Finally, in Section 3.3, we measured the layerwise evolution of the values of the network's weights and biases in an attempt to quantify the temporal dynamics of learning. We found that learning occurs in a top-down manner, where higher layers undergo more and earlier changes than lower layers. Taken together, our results support a reverse-hierarchical account of deep learning that is consistent with theories of perceptual learning in the brain.

2 Methods

2.1 Model

We used an AlexNet model (Krizhevsky et al., 2012) implemented with Google's TensorFlow library (Abadi et al., 2015).[1] The input layer contained $40 \cdot 40 = 1600$ units, reflecting the dimensions of the input images. The learning rate was set to 0.0001, and the batch size was set to 50. Dropout probability was set to 0.8 (meaning there was an 20 percent chance that a given unit would be discarded). The neural network model, topology, and hyperparameters were kept constant across all three procedures.

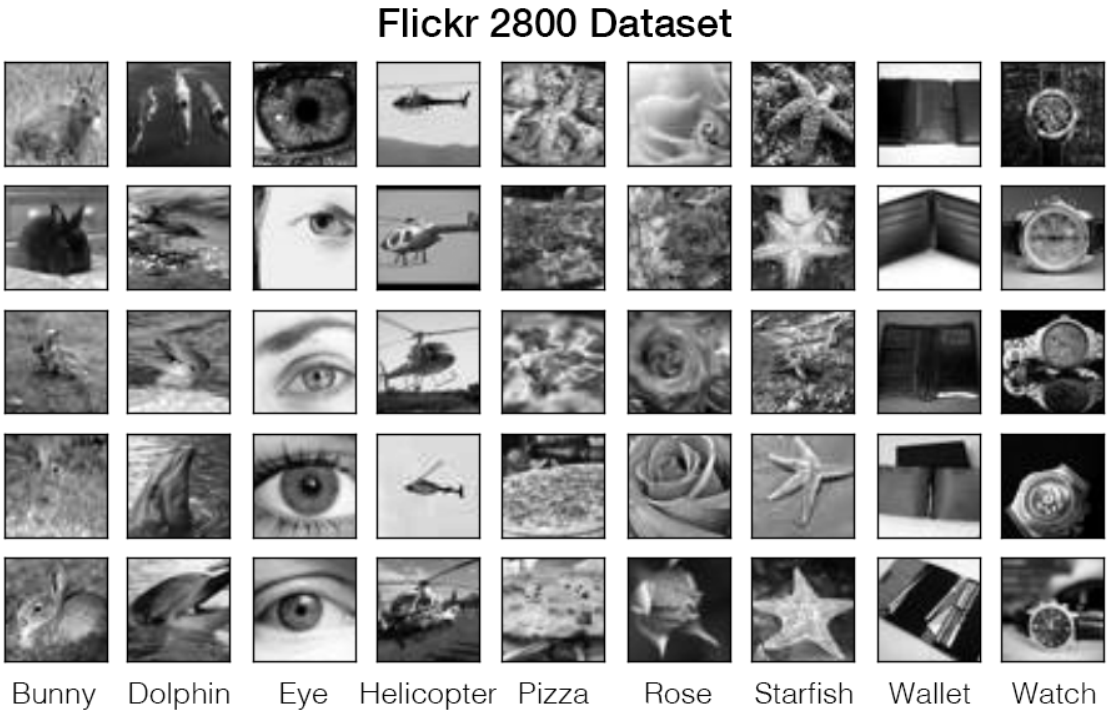

Figure 1: **Samples from Flickr 2800 dataset.** The dataset consists of 10 image categories, each containing 2,800 images. Each category was manually selected for consistency.

2.2 Flickr 2800

For this project, we considered using several common machine learning datasets, including the Caltech-101 dataset and ImageNet. However, most categories of Caltech-101 contain only approximately 50 images, which we deemed was not sufficient for training. Meanwhile, ImageNet contained more than enough images. However, the approval process necessary to get access to ImageNet would have have shortened the window during which we could perform our experiments. Ultimately, we opted to assemble our own "Flickr 2800"

[1] Code can be found at: http://github.com/aymericdamien/TensorFlow-Examples/

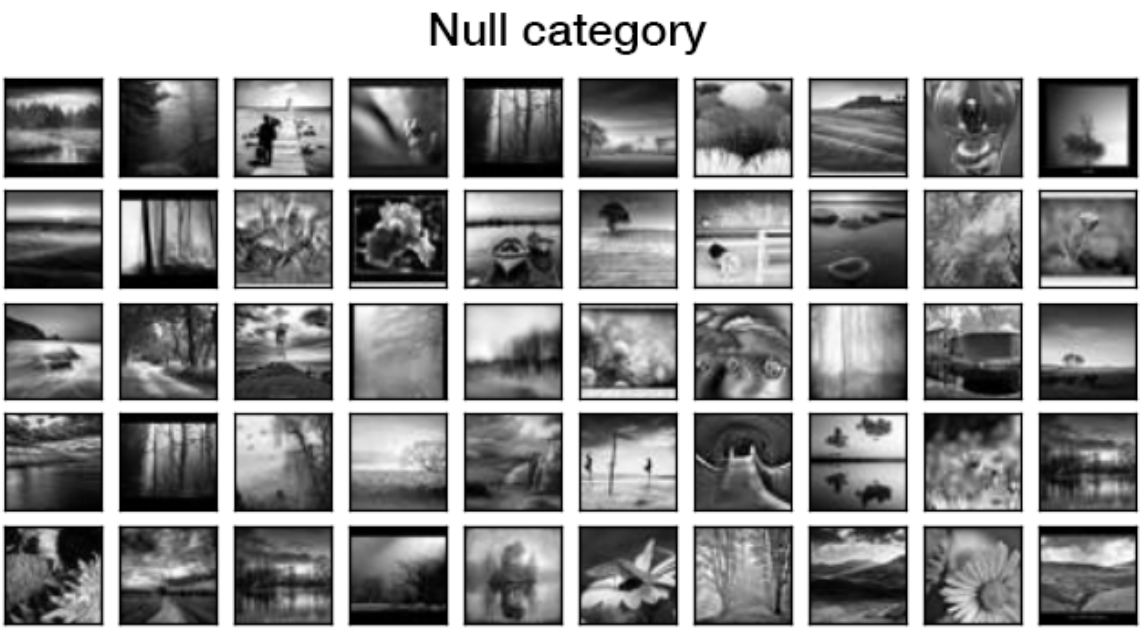

Figure 2: **Null category.** An additional null category was formed by querying the Flickr API for 2,800 images that contained none of the 10 object tags.

dataset, which provided us with additional experimental flexibility while still allowing for a high degree of control over the content of the images.

Flickr 2800 consisted of approximately 30,800 images scraped from Flickr. The dataset was composed of 10 categories of common objects: bunny, dog, dolphin, eye, helicopter, pizza, rose, starfish, wallet, and watch. Each category was manually selected for consistency and contained 2,800 images. An additional category of 2,800 images was formed by querying the Flickr API for images that contained none of the 10 object tags. This category contained a variety of images (landscapes, people, etc.) and was used as the negative training class during all supervised learning procedures. All images were scaled to 40x40 pixels and converted to grayscale.

3 Results

3.1 Cross-domain transfer learning

To test our initial hypothesis, we trained five separate AlexNets on various classes of images from Flickr 2800 (each class has 2800 distinct images): eye, pizza, wallet, rose, watch. Each net was trained as a binary classifier between the selected class and the null class. At the start of training, the test and null set were shuffled and divided into training and test sets (80% and 20% respectively). We proceeded to train the net using batches of 50 images, evaluating performance (both loss and accuracy) against the whole test set every 250 iterations. We have reproduced both the accuracy and loss curves for all five classes below.

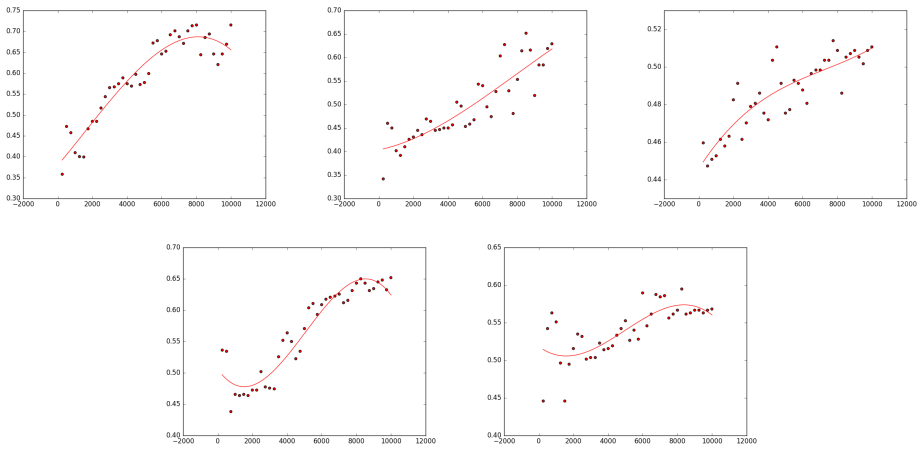

Figure 3: **Accuracy curves for nets initialized with random weights.** Top row: eye, pizza, rose. Bottom row: wallet, watch.

The distinct upward trend on all the accuracy curves demonstrates that the performance of the nets on the classification task improved over time. It should be noted that, while none of the nets appear to have reached convergence, they appear to have learned some features, which is sufficient for our experiment. The loss curves (below) paint a similar picture as the accuracy curves. To avoid redundancy, for the remainder of this paper, we elect to focus on accuracy as the principal metric of performance.

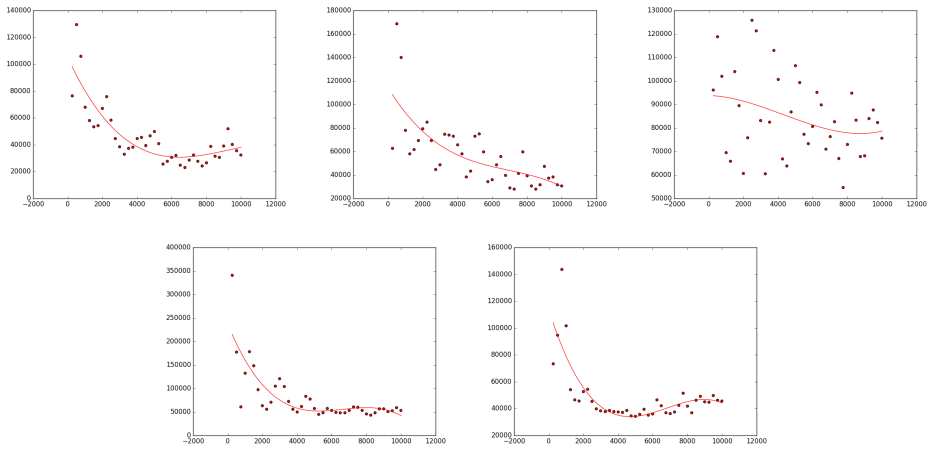

Figure 4: **Loss curves for nets initialized with random weights.** Top row: eye, pizza, rose. Bottom row: wallet, watch.

After confirming that the nets were indeed learning under our particularly constrained conditions (short training period, relatively small dataset, etc), we shifted our attention to quantifying the transfer of learning. To do this, we transferred all the weights and biases from the one net into a brand new one, and trained it on a different data set.

Terminology: Networks that use transferred weights and biases are named `class1_class2`, where `class1` indicates the class we transferred the weights and biases from, and `class2` refers to the class we trained the net to recognize.

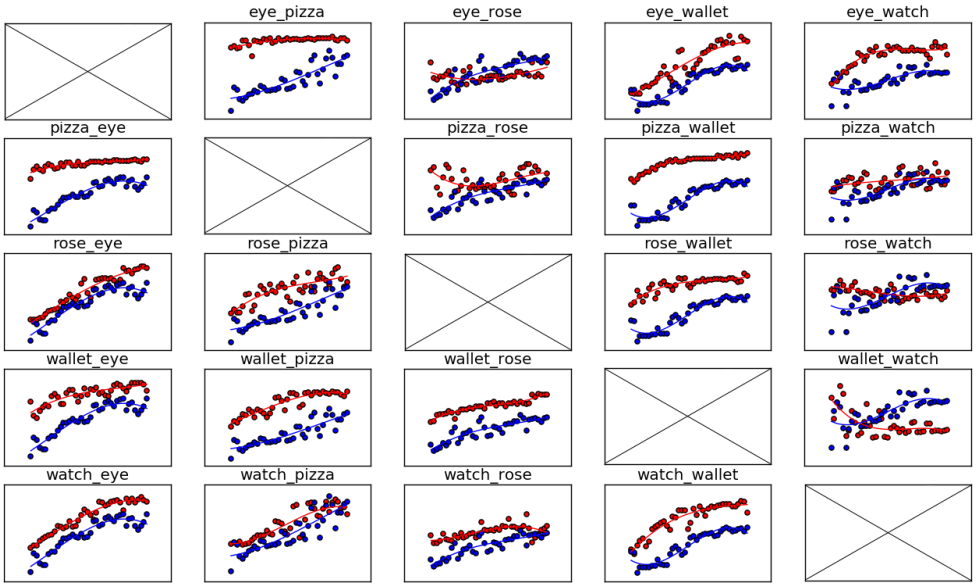

Figure 5: **Matrix of transfer learning.** Each graph is titled `class1_class2`. The red plot demonstrates the learning curve `class2` after starting with `class1`, while the blue plot shows the learning curve for `class1` when initialized with randomized weights. In almost every single case, the net with transferred weights clearly outperformed the randomly-initialized net.

As illustrated by the above matrix, all but approximately three curves show that networks with transferred weights will both start and end with higher accuracy than their randomly-initialized counterparts. This was surprising to us; while we expected that the nets might train faster, we did not expect that they would end with higher accuracy.

On average, nets initialized with random weights had an final accuracy of 0.615, while those initialized with transferred weights had a final accuracy of 0.664. Transferred nets were able to match the final accuracy of their untransferred counterparts in an average of 2105.263 iterations. In cases where the initial accuracy of a transferred net exceeded the final accuracy of its counterpart, the minimum value of 250 iterations was used. The sole case where the transferred net failed to reach the accuracy of its counterpart was excluded.

To measure the difference in learning curves, we used the difference in the discrete integral (right-hand Riemann sum) between the two curves. This metric captures both higher accuracy and faster learning. The average difference for all of the graphs we tested was 908.480.

3.2 Hierarchical lesioning

Given that networks with transferred weights appear to outperform randomly initialized ones, do certain hidden layers contain more transferable knowledge than others? In other words, is it possible to detect where the transferable knowledge is stored? To investigate this question we "lesioned" various permutations of hidden layers.

Terminology: Throughout this section, a 6-bit binary string is used to indicate which hidden layers were lesioned. A 0 indicates that the weights and biases for a given layer were sampled from a random normal (i.e., "lesioned") rather than transplanted from a previous model. The first three bits in the string refer to the lowest three layers, the next two to the fully connected layers, and the last bit to the hidden output layer. For instance, a bit string of `111111` is a non-lesioned network, where all layers contained transferred weights. Meanwhile, `000000` is equivalent to not transplanting in weights from a previous model at all (i.e. training a "blank" net).

Figure 6: **Compared performance of multiple lesion permutations.** There appear to be two distinct performance groups. Nets with high-level lesions tend to show impaired performance, while nets with low-level lesions demonstrate comparable performance to an unlesioned network. The curves demonstrate test accuracy over 10,000 training iterations.

The lesion permutation `111000` results in slightly better results than starting from a blank net and is exceeded slightly by `110000`. This seems to suggest that knowledge stored in the lower layers is less suitable for transfer than knowledge stored in higher layers. This difference in learning performance and accuracy is evident in both the higher overall accuracy for nets with high-level lesions (after ten-thousand iterations), as well as their higher initial accuracy. It should also be noted that no transplants (with the exception of `000001`) appeared to inhibit learning during any stage of the training process, and all outperformed the `blank_eye` after a maximum of 1000 training iterations.

Interestingly, while `111000` yields slightly better performance than beginning from a blank net, not lesioning the final layer of the net `(111001)` boosts performance to beyond the performance of even the normally transplanted `wallet_net`. This may suggest that transplanting the fully-connected layers is sub-optimal (and by implication that some layers, while still useful for knowledge transfer, may undermine the efficacy of transplanting other layers).

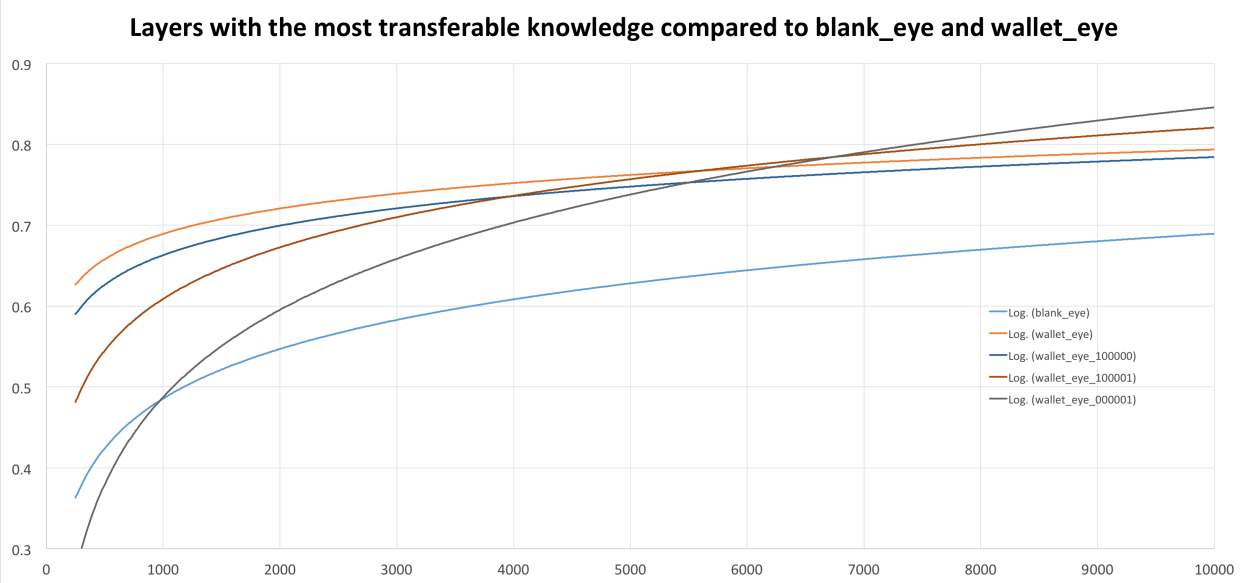

Figure 7: **Isolating layers with the most transferable knowledge.** All layers except for the top (Dense 2) and bottom (Conv 1) were lesioned.

Fig. 7 shows the most minimal transplants possible which still yielded and/or exceeded the performance boost of transplanting all weights. Interestingly enough, `100000` and `100001` performed roughly equivalently to `000001`, perhaps challenging the notion that only higher level features are transferred. This may be evidence that the lowest and highest level features are transferable, and that "middle" features are less so and perhaps even inhibitory.

Fig. 8 is a summary of these results. The estimated area under each learning curve is used as a proximate metric for the absolute amount of learning which takes place, as well as rate.

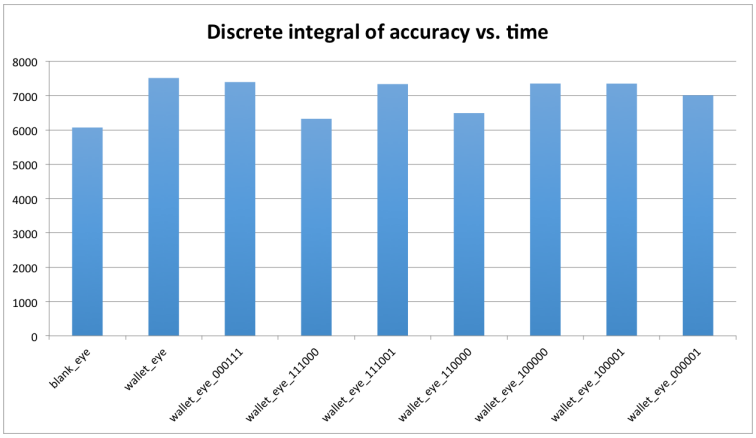

Figure 8: **Discrete integral under the accuracy curve.** The unlesioned transfer net performs best. However, several other lesioned nets demonstrated comparable performance.

*A final note: All of the above figures relate to the specific case of training a net on the `eye` class after already learning to recognize `wallet`. Similar results were found for other class pairs, but are excluded for the sake of brevity.

3.3 Layerwise temporal dynamics of learning

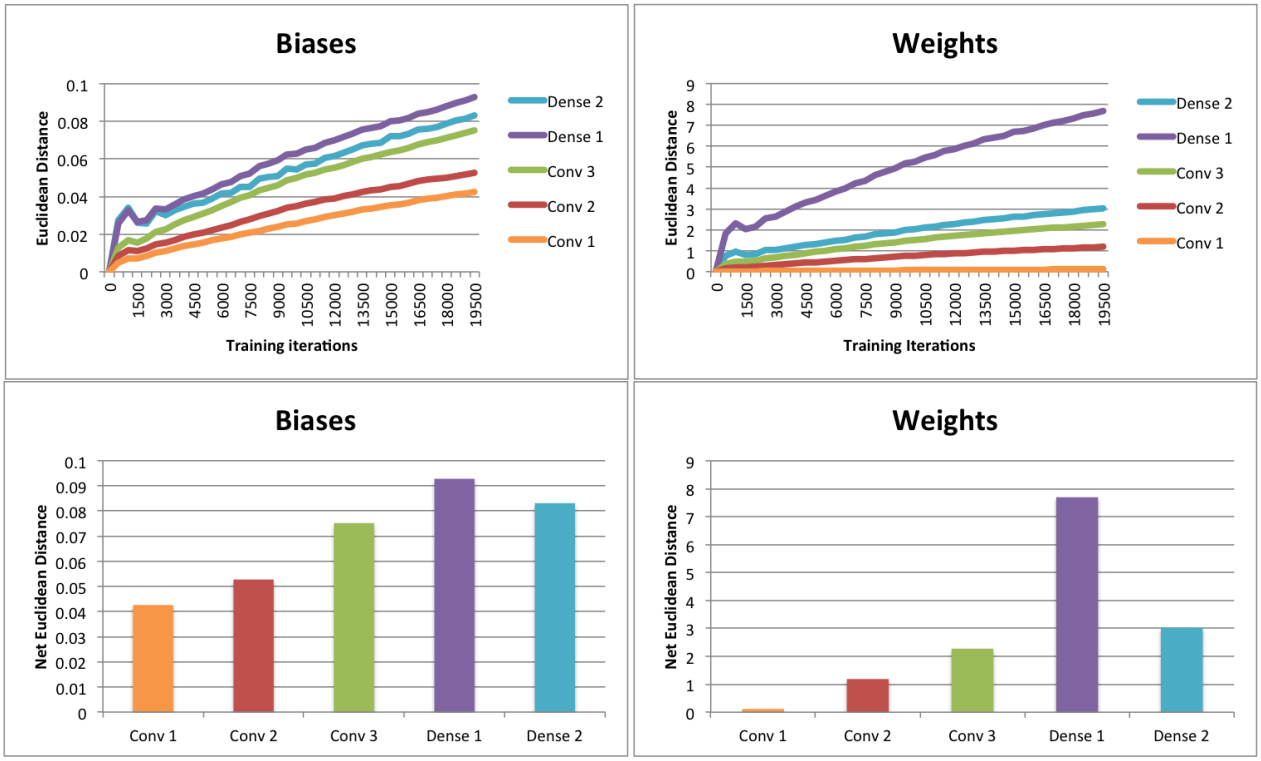

Figure 9: **Layerwise evolution of biases and weights.** Top row: Each line represents a hidden layer of the network as it changed during the course of training. In accordance with the Reverse Hierarchy Theory, higher layers of the network underwent more change than lower layers. (For convenience, layers the legend are ordered according to their placement in the network. "Conv" = convolutional layer; "Dense" = fully connected layer.) Bottom row: net euclidean distance over 10,000 training iterations for each layer.

We trained a randomly-initialized network for 20,000 iterations on the "eye" category. Every 500 iterations, the biases and weights of each layer were recorded as vectors. (The weight matrices of each layer were flattened to form vectors, while the biases were given as vectors in the first place.) This produced a time-series of bias vectors $\vec{b_0}, \vec{b_1}...\vec{b_t}$ and weight vectors $\vec{w_0}, \vec{w_1}...\vec{w_t}$.

For each layer's time series, we calculated the euclidean distance between the bias and weight vectors at the nth iteration and the initial bias and weight vectors:

$$distance = \sum_{i=0}^{n} \sqrt{(v_t)_i - (v_0)_i}$$

We chose euclidean distance as a convenient metric because it allows for the comparison of distances across vector spaces of different dimensions. In our case, this is especially crucial because different layers of the network contain different numbers of units, and therefore produce bias and weight vectors of different dimensions. Nevertheless, it is important to note that euclidean distance as a metric does not shed any light on the nature of the changes taking place in the network across time. This method only verifies that such changes do occur, and provides a convenient way to quantify the degree of change in each layer over time.

With the exception of the top two (fully connected) layers, which are switched, the temporal dynamics during training exactly mirror the network's structure. In other words, both in their weights and biases, higher layers of the network exhibit higher rates of change than lower layers. The pattern we observed was remarkably consistent across all ten of the image categories.

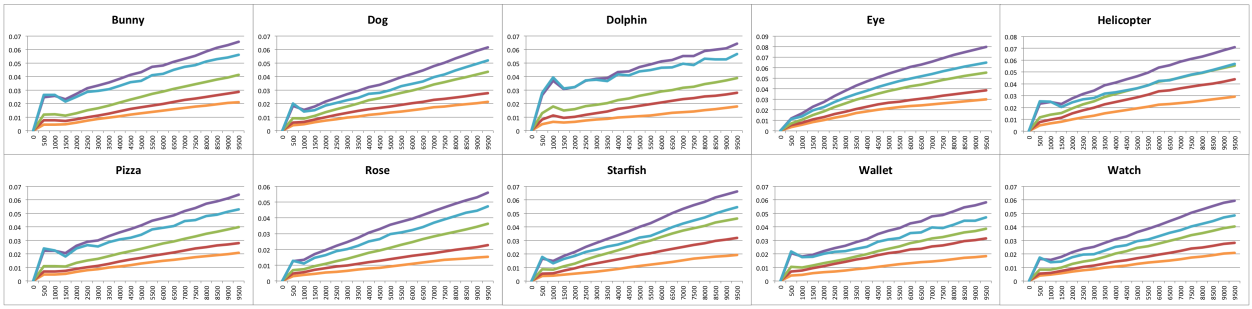

Figure 10: **Layerwise evolution of biases across 10 categories.** Patterns of learning were highly uniform across all 10 image categories.

As expected, networks with transferred weights exhibited less change across all layers compared to randomly-initialized networks. Again, this result was remarkably general across domain: nets with transferred biases and weights uniformly exhibited less overall change in biases and weights compared to randomly-initialized nets, regardless of which category the weights and biases were transferred from.

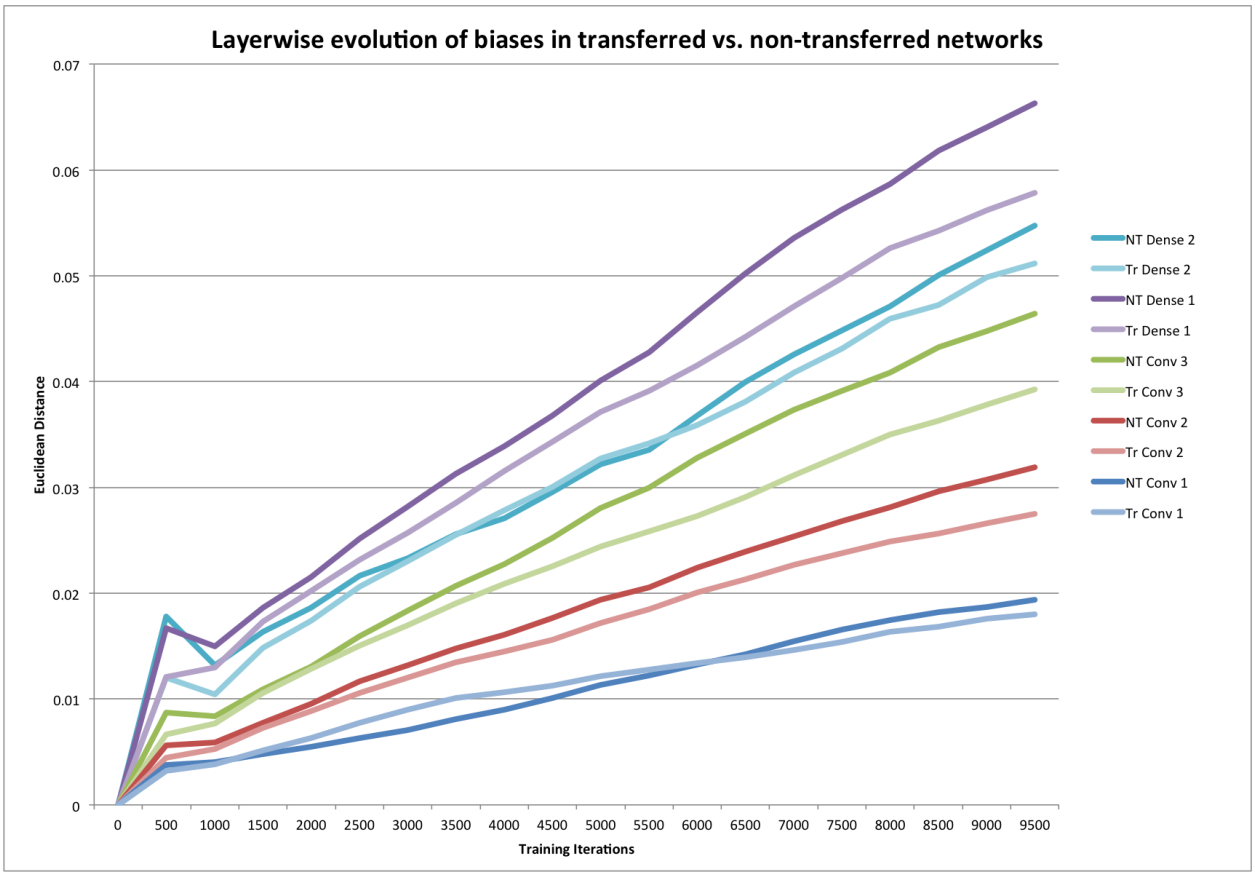

Figure 11: **Layerwise evolution of biases in transferred vs. non-transferred networks.** Networks with transferred biases and weights exhibit less change in each layer during training when compared to networks without pre-training. A similar pattern was observed with the layerwise evolution of weights. (Tr = transferred; NT = non-transferred.)

4 Discussion

4.1 Applications

From an engineering perspective, our results help to illustrate the significant potential benefits of transfer learning in many application settings. In section 3.1, we found that using transferred weights dramatically decreased initial loss, increased initial accuracy, and consistently decreased the number of training iterations necessary to reach threshold performance. Deep learning is both computationally- and data- intensive. By expediting the training process, transfer learning offers the opportunity to avoid thousands of unnecessary CPU cycles in which low-level features are learned.

Additionally, supervised learning necessitates the collection of large amounts of labeled data, a process that often involves many man-hours of work. Moreover, sufficient training data may not be available in some domains, or may be in a different feature space or follow a different distribution (Pan and Yang, 2010). In these settings, transfer learning would greatly expedite the learning process by allowing labeled data to be imported from a different domain.

Indeed, researchers and industry professionals have already begun to apply transfer learning to a wide variety of domains, including medical image analysis (Shie et al., 2015), local climate modeling (Hu et al., 2016), action recognition (Giel and Diaz, 2015), and even video games (Tessler et al., 2016). Thus, transfer learning offers an attractive alternative to initializing the network with random initial weights and biases.

4.2 Comparing transfer learning in artificial and biological neural networks

Though transfer learning results in a performance boost for deep neural networks, it would be naive to conclude from this fact alone that neural nets exhibit "learning-to-learn" in the same way that humans do. So long as neural nets require thousands of training instances to accurately recognize a certain class of object, we will be hard-pressed to claim that they learn in a manner comparable to the brain.[2] Nevertheless, it is still useful to develop a better understanding of exactly how artificial neural networks models learn in relation to their biological counterparts.

One interesting account of perceptual learning in the brain is Reverse Hierarchy Theory (RHT) (Ahissar and Hochstein, 2004; Hochstein and Ahissar, 2002). RHT asserts that learning is a top-down guided process, which begins in high levels of the cortical hierarchy, and gradually cascades down to lower layers when top-level modifications do not suffice (see Fig. 12, left). RHT is supported by behavioral studies that showed that on an orientation-discrimination task, easy task conditions were learned early and were easily generalized to new stimuli. Meanwhile, hard task conditions were learned later and performance improvements had higher task-specificity (Ahissar and Hochstein, 1997). Electrophysiological studies of primates also found that changes in IT receptive-field properties exhibited a high degree of orientation-generalization, while neurons in V4 and lower level visual areas were both orientation- and position-specific (Sigala and Logothetis, 2002; Vogels and Orban, 1994; Yang and Maunsell, 2004).

With respect to neural network models, RHT predicts that higher layers of the net should contain more generalizable knowledge than lower layers. Indeed, our lesion experiments showed that in all but one case, "lesioning" the upper layers of the net by resetting their weights and biases causes a marked performance drop compared to non-lesioned nets. Conversely, lesioning the lower layers of the net did not cause significant impairment, and in some cases actually resulted in a slight improvement to accuracy. Layerwise analysis of the evolution of weights and biases provides further support for the notion that higher layers of the network undergo more and earlier change than lower layers.

Intriguingly, these findings seem to be at odds with a piece of common wisdom in machine learning: namely, that lower layers of convnets learn abstract features like Gabor filters and color blobs, while higher layers learn highly-specific representations of individual objects (Yosinski et al., 2014). According to this logic, low-level features should be much more transferable across disparate object domains, so lesioning low levels should theoretically have a negative impact on performance. Furthermore, if low-level features were relatively consistent across different object domains, then the lower layers of networks with transferred weights should have "less to learn" in a new domain. If this were the case, then when comparing the layerwise evolution of transferred vs. non-transferred nets, we should have observed a greater difference between net

[2]Using Bayesian approaches, other models have achieved one-shot learning. See (Fei-Fei et al., 2006).

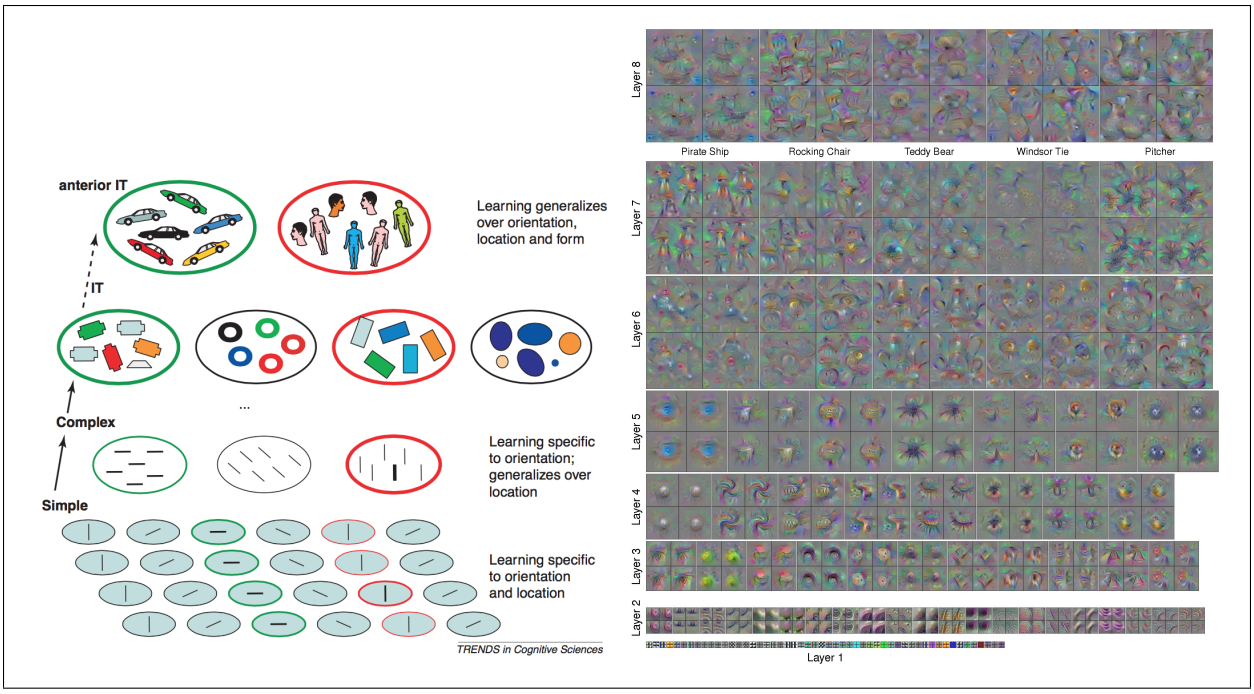

Figure 12: **Reverse Hierarchy Theory.** Left: According to RHT, a hierarchy of cortical areas is responsible for processing visual information. Lower levels represent specific details of color, orientation, motion, 2D position and stereoscopic disparity. Meanwhile, higher level representations of shapes, categories, objects, and concepts generalize to new learning conditions (Ahissar and Hochstein, 2004). Right: Visualization of features in each layer of a trained convnet. In accordance with RHT, representations learned at each layer increase in complexity as we move up the hierarchy (Zeiler and Fergus, 2014)

.

learning (as quantified by euclidean distance) in low-levels than in high levels. Instead, however, we observed the opposite phenomenon (See Fig. 11).

One possible computational explanation for our observations is known as the "vanishing gradient problem." It is well-documented that stochastic gradient descent tends to be unstable in deep neural networks (Hochreiter et al., 2001; Bengio et al., 1994). In some cases, this can lead to a scenario where lower layers of the network learn orders of magnitude less information than higher layers. Vanishing gradient would help to explain why lower layers of the network consistently evolved less than higher layers. Additionally, it would also explain why lesioning lower layers had a minimal effect on the network's performance.

On one level, this explanation appears to be purely a consequence of the statistical properties of deep neural nets. However, if the visual system operated under similar principles, then we might also expect to see a similar effect in the brain. Currently, he question of whether deep learning in general, and stochastic gradient descent in particular, is biologically plausible is the subject of much debate. Indeed, machine learning researchers have often attempted to convince their skeptical neuroscientist colleagues that such a feat is indeed possible with neural circuitry (Bengio et al., 2015; Hinton, 2007). If some neurobiological mechanism for backpropagation, such as spike-timing dependent plasticity, were indeed discovered, then it would allow us to apply our existing knowledge about the statistical dynamics of deep learning to the brain. In turn, this connection could help to provide a theoretical basis for the layerwise learning dynamics put forward by the Reverse Hierarchy Theory.

4.3 Directions for further exploration

With our three experiments, we have only scratched the surface when it comes to understanding how and why transfer learning works in deep neural networks. Using these results as a starting point, there are many possible directions that future research could take. One approach that falls in line with the current *modus operandi* of machine learning research would be to simply apply more data and computational power to the same methodology. Resource constraints had a significant influence on almost every aspect of our research, from the selection of the dataset, to the use of low-resolution grayscale images, to the amount of training we provided our networks. Perhaps more training iterations on more data could reveal new patterns in the learning process that occur over larger timescales.

Aside from simply addressing resource limitations, research in transfer learning could benefit from more sophisticated computational methods of investigation. As discussed, the use of euclidean distance as a metric for learning is not ideal, because it does not reveal how the network's internal representations evolve over time. One solution would be to apply Representational Similarity Analysis (Kriegeskorte et al., 2008) to identify what exactly the network learns during training. Similarly, use of layerwise feature visualization techniques (Zeiler and Fergus, 2014; Yosinski et al., 2015) could provide a better subjective understanding of the kinds of features learned at each layer. Alternatively, a different approach would be to use the features of each layer to train a logistic regression model, and then apply this model to a classification task. This procedure would provide a more powerful metric of the richness of the feature representations learned at each layer.

Our research considered perceptual learning only in the domain of object recognition. However, deep neural networks have been successfully applied to many other domains, including audition and speech processing (Graves et al., 2013; Weng et al., 2014). A fruitful direction for future research would be to examine whether perceptual learning transfers across different sensory modalities.

5 Conclusion

Our findings suggest that deep neural networks learn internal representations that possess a remarkable degree of domain generality. They also provide support for the notion that deep learning is consistent with top-down theories of perceptual learning in the brain, such as the Reverse Hierarchy Theory.

Nevertheless, these results do not lead us to believe that deep learning is the be-all-end-all of artificial intelligence. In particular, deep neural networks lack the kinds of compositional, hierarchical, and causal forms of representation that allow humans to generate new examples of a class of objects, or devise temporally-extended planning strategies (Lake et al., 2016). Even with extensive pre-training on related images, it is unlikely that a current AlexNet will ever be able to learn to recognize a new object class from a single instance, regardless of the amount of training data available. Ultimately, in order to create AI models capable of true "learning-to-learn," we will need to transfer our own knowledge from cognitive neuroscience to our future endeavors in the field of machine learning.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2015). Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow.org*.

Ahissar, M. and Hochstein, S. (1997). Task difficulty and the specificity of perceptual learning. *Nature*, 387(6631):401–406.

Ahissar, M. and Hochstein, S. (2004). The reverse hierarchy theory of visual perceptual learning. *Trends in Cognitive Sciences*, 8(10):457–464.

Bengio, Y., Lee, D.-H., Bornschein, J., and Lin, Z. (2015). Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

Carey, S. and Bartlett, E. (1978). Acquiring a single new word.

Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):594–611.

Giel, A. and Diaz, R. (2015). Recurrent neural networks and transfer learning for action recognition.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034.

Hinton, G. (2007). How to do backpropagation in a brain. In *Invited talk at the NIPS'2007 Deep Learning Workshop*.

Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Hochstein, S. and Ahissar, M. (2002). View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 36(5):791–804.

Hu, Q., Zhang, R., and Zhou, Y. (2016). Transfer learning for short-term wind speed prediction with deep neural networks. *Renewable Energy*, 85:83–95.

Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2016). Building machines that learn and think like people. *arXiv preprint arXiv:1604.00289*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359.

Pinker, S. (1999). How the mind works. *Annals of the New York Academy of Sciences*, 882(1):119–127.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Rutishauser, U., Mamelak, A. N., and Schuman, E. M. (2006). Single-trial learning of novel stimuli by individual neurons of the human hippocampus-amygdala complex. *Neuron*, 49(6):805–813.

Shie, C.-K., Chuang, C.-H., Chou, C.-N., Wu, M.-H., and Chang, E. Y. (2015). Transfer representation learning for medical image analysis. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 711–714. IEEE.

Sigala, N. and Logothetis, N. K. (2002). Visual categorization shapes feature selectivity in the primate temporal cortex. *Nature*, 415(6869):318–320.

Tessler, C., Givony, S., Zahavy, T., Mankowitz, D. J., and Mannor, S. (2016). A deep hierarchical approach to lifelong learning in minecraft. *arXiv preprint arXiv:1604.07255*.

Vogels, R. and Orban, G. A. (1994). Does practice in orientation discrimination lead to changes in the response properties of macaque inferior temporal neurons? *European Journal of Neuroscience*, 6(11):1680–1690.

Weng, C., Yu, D., Watanabe, S., and Juang, B.-H. F. (2014). Recurrent deep neural networks for robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 5532–5536. IEEE.

Yang, T. and Maunsell, J. H. (2004). The effect of perceptual learning on neuronal responses in monkey visual area v4. *The Journal of Neuroscience*, 24(7):1617–1626.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer vision–ECCV 2014*, pages 818–833. Springer.