The Emergence of Semantics in Neural Network Representations of Visual Information

Dhanush Dharmaretnam
University of Victoria
Department of Computer Science,
3800 Finnerty Rd, Victoria, Canada
dhanushd@uvic.ca

Alona Fyshe
University of Victoria
Department of Computer Science,
3800 Finnerty Rd, Victoria, Canada
afyshe@uvic.ca

Abstract

Word vector models learn about semantics through corpora. Convolutional Neural Networks (CNNs) can learn about semantics through images. At the most abstract level, some of the information in these models must be shared, as they model the same real-world phenomena. Here we employ techniques previously used to detect semantic representations in the human brain to detect semantic representations in CNNs. We show the accumulation of semantic information in the layers of the CNN, and discover that, for misclassified images, the correct class can be recovered in intermediate layers of a CNN.

1 Introduction

As we study semantics through the lens of a corpus, it can be easy to forget that concepts exist independently of language. Animals with no language system still form representations of concepts, based on their interactions with the world (e.g. vision, touch, taste). In this paper, we bridge the gap between the study of semantics in computer vision and computational linguistics. We take inspiration from previous work on brain-based representations of meaning, and measure the semantic information through the layers of a Convolutional Neural Network (CNN) trained to detect objects in images.

Our work is not the first to draw connections between computer vision and distributional semantics. Indeed, joint models of semantics based on images and text have been developed, and word vectors have been grounded in the visual space (Silberer and Lapata, 2012, 2014; Bruni and Baroni, 2013). However, we believe that our work is the first to explore the convergence of semantic embeddings built from text and images, specifically studying the hidden representations of CNNs, and how the accumulation of semantic evidence builds as a function of network depth.

2 Methods

The representations created by CNNs have differing numbers of dimensions, and even within a CNN, the size of layers may differ (see Section 2.2 for more details on CNNs). In addition, the size of word vectors also varies. We need a method to detect similarities across embedding spaces, regardless of their dimensionality.

A solution, *similarity-encoding*, was proposed in parallel by two recent papers (Anderson et al., 2016; Xu et al., 2016), which calculates the similarity of embedding spaces using the *correlation* of elements within a space, rather than directly comparing the embeddings between spaces.

Similarity-encoding compares the correlation matrices of embedding spaces. For example, if the word vectors have dimension p, our matrix of word vectors will be $\mathbb{R}^{k \times p}$(where k is the number of concepts). The resulting correlation matrix will be $W \in \mathbb{R}^{k \times k}$, and will represent the pairwise correlation of all concepts in word space. Similarly, for a given layer of a CNN with dimension q, we can create a matrix of embeddings for the same k concepts ($\mathbb{R}^{k \times q}$), and compute a correlation matrix in image space: $I \in \mathbb{R}^{k \times k}$. Thus, we have taken embedding spaces for the same set of k concepts and created new spaces of dimension k based on the correlation between concepts. Because the correlation spaces share the same dimension, we can directly compare the correlation matrices of word vectors (W) with the correlation matrices of image embeddings (I).

To compare the correlation spaces, we could simply calculate the correlation between matrices W and I, which would be a Representational Similarity Analysis (RSA), as proposed by Kriegeskorte et al. (2008). RSA has the advantage of being a

Proceedings of NAACL-HLT 2018, pages 776–780
New Orleans, Louisiana, June 1 - 6, 2018.

fairly simple and straightforward method for comparing the representations across two embedding spaces. However, RSA has the disadvantage that it produces one aggregate number for a pair of correlation matrices, and does not show which concepts contributed most to a high or low score. We will need this added flexibility to explore misclassified images in Section 3.3.

Both Anderson et al. (2016) and Xu et al. (2016) extend RSA with an additional analysis step, inspired by some of the first work searching for semantics in the brain (Mitchell et al., 2008). Anderson et al. (2016) present a pictorial overview of the procedure in Figure 2 of their paper.

In similarity-encoding, we choose two elements from the concept list (c_1 and c_2), and calculate the correlation of their embeddings to every other $k-2$ concept in each embedding space (word or image). This creates a vector for each of the two held out concepts with length $k-2$, and is equivalent to selecting the corresponding rows of the full correlation matrices W and I, but omitting the columns which correspond to c_1 and c_2. We then compare the correlation patterns of c_1 and c_2 in word space (vectors w_{c_1} and w_{c_2}) to the correlations in image space (vectors i_{c_1} and i_{c_2}) by checking if:

$$\text{corr}(w_{c_1}, i_{c_1}) + \text{corr}(w_{c_2}, i_{c_2}) \quad (1)$$

(the correlation of correctly matched concepts: c_1 to c_1 and c_2 to c_2) is greater than:

$$\text{corr}(w_{c_1}, i_{c_2}) + \text{corr}(w_{c_2}, i_{c_1}) \quad (2)$$

(the correlation of incorrectly matched concepts). Xu et al. (2016) call this the **2 vs. 2 test**. If the correctly matched vectors are more correlated than the incorrectly matched vectors, then the test is considered to have passed. We perform the 2 vs. 2 test for all possible pairs of concepts, 13,695 tests in total. The **2 vs. 2 accuracy** is the percentage of 2 vs. 2 tests passed, and chance 2 vs. 2 accuracy is 50%. We compute significance for the 2 vs. 2 test by performing a permutation test: permuting the rows of embeddings in one space, and re-running the full similarity-encoding methodology.

2.1 Word Vectors

We chose four word vector models from recent work. **SkipGram** vectors are from a neural network trained to predict co-occurring words. We used the 300 dimensional model trained on Google news (English) (Mikolov et al., 2013). **RNN** is a recurrent neural network trained to predict the next word in a sequence. It has 640 dimensions, and was trained on transcriptions of English broadcast news(Mikolov et al., 2011). **Glove** is a regression-based model that incorporates both local and global co-occurrence information. This 300-dimensional model was trained on the English Wikipedia and Gigaword 5 corpora combined (Pennington et al., 2014). **Cross-lingual** word vectors project embeddings from multiple languages into a shared space. We used the German-English model (512 dimensions), trained on WMT-2011 (Faruqui and Dyer, 2014).

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) can be trained to perform image classification (Goodfellow et al., 2016). In general, the input is three continuous valued matrices, representing the RGB values of an image. The network convolves the input with a set of learned filters (typically 2D). The output of the convolution is fed to another layer of the network, which performs additional operations (e.g. more convolutions, pooling). Each layer of the network produces a hidden representation that is used by subsequent layers, ending in a final classification layer.

There has been a proliferation of neural network architectures for image classification; we explore three CNNs: VGG 16, ResNet 50 and Inception V3. We chose these networks based on availability (pre-trained models are readily downloadable), for their performance on image classification tasks, and for their diversity in structural complexity. All CNNs used here were trained on ImageNet (Deng et al., 2009).

VGG 16 is one of the two deepest networks described in Simonyan and Zisserman (2014). It has a very simple linear architecture. We measured the 2 vs. 2 accuracy against all convolutional and dense layers of the network. **Inception V3** (Szegedy et al., 2015), has a more complicated architecture, including Inception blocks which act as multi-resolution feature extractors, applying differently sized filters in parallel. We measured the 2 vs. 2 accuracy at the concat (mixed) layers, which appear at the end of inception blocks. **ResNet 50** (He et al., 2015) uses residual modules, which use linear shortcut connections to allow earlier representations to filter up as required. Residual modules allow the networks to become very deep without the typical problems in training associated with deep networks. We measured the 2 vs. 2 accuracy at the activation layers both within and at the end of residual blocks.

Illustrations of the architectures for each of these networks appear in the supplementary material (Figures 1-3), annotated to show the layers we used for our experiments. We used the Keras implementation of all networks (Chollet et al., 2015).

2.3 Concept Selection

Each of the CNNs described in Section 2.2 was trained on ImageNet, a collection of over a million images, each annotated for the presence of one of 1000 concepts. These concepts can be fairly high-level single words (e.g. stove, sandwich) or extremely specific multi-word concepts (e.g. German short-haired pointer, tobacco shop). We selected all concepts for which a match was found in all four of the word vector models from Section 2.1, resulting in 166 concepts. From this set of 166 concepts, we randomly chose 5 images annotated with the given concept in the ImageNet validation set, for a total of 830 images.

We then computed each network's activation on each of the 830 images. These images are divided into 5 groups, such that each of the 166 concepts occurs exactly once per group. We ran the 2 vs. 2 test separately for each of the 5 groups, and report the average across the 5 runs to account for variability across images.

3 Results

3.1 Word Vector Comparison

Figure 1 shows the performance of several word vector models against the layers of VGG 16. The first point represents the performance using correlation at the pixel level only, and no CNN-derived representation. The performance of SkipGram, Glove and the Cross-lingual vectors are very similar, and are within a percentage or two across all layers. The RNN model we tested did not perform as well, on average about 5% lower in the first convolutional layers, and 10% lower in the highest hidden layers. These results are similar to those seen in Xu et al. (2016) when comparing word vectors to brain activity. Because the performance is very similar for the three top performing word vectors, our analyses proceeds with SkipGram vectors only.

Note that the 2 vs. 2 accuracy improves as we move up the layers of the CNN. This is evidence that, though trained on very different data sources, the semantic representations in CNNs and word vectors are quite similar, and the similarities grow stronger as the CNN gets closer to its final classification layer. Though the starting points are different (text vs. images) the final result of the CNN is similar to word vectors built from corpora, implying that a shared embedding space can emerge from each data source independently.

The very early layers of CNNs have been shown to represent low level features like edges, curves and other simple shapes (Mahendran and Vedaldi, 2016), so we did not expect early layers to have any significant relation to word vectors. We were surprised to find that even the very first layer of the VGG 16 gives above chance 2 vs. 2 accuracy using SkipGram vectors ($p < 0.001$). Upon inspection, we found that the correlation for matched vectors (the value for Eq. 1) was just slightly larger than the correlation for mismatched vectors (the value of Eq. 2), implying that the network has only weak evidence for semantic relationships at the early layers of the CNN. Figure 4 in the supplementary material shows this effect in greater detail. Note that the first layer of VGG 16 improves upon the pixel level accuracy, implying that even simple CNN features provide useful signal.

We also noted that some macroscopic distinctions between the concepts can likely be inferred from the low level features of images alone. For example, man made objects tend to have more straight lines and natural objects are more curved. Thus, it is logical that the early layers of a CNN could distinguish between some pairs of objects using only the most basic of visual features.

3.2 CNN Comparison

Figures 1-3 show 2 vs. 2 accuracy for SkipGram vectors against the layers of VGG 16, ResNet 50 and Inception V3 networks. In general, the pattern is similar: as the depth of the layers increases, so too does the 2 vs. 2 accuracy. However, there are a few interesting exceptions to this pattern. In ResNet 50 we see a drop in accuracy in several places, most notably between activation layers 16 and 17. Upon inspection of the architecture diagram, we noted that several of the early residual blocks were not improving 2 vs. 2 accuracy, and relied mostly on residual connections (see Supp. Figure 3), implying some of the depth in ResNet 50 may be unnecessary.

We see maximum 2 vs. 2 accuracy in later layers of CNNs, but not always at the last layer. Incep-

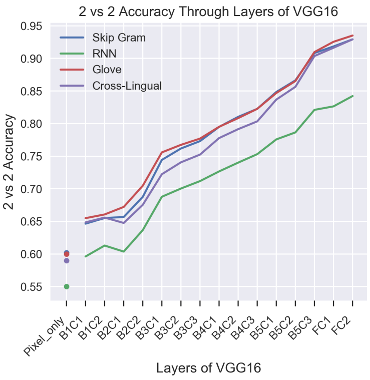

Figure 1: 2 vs. 2 accuracy for 4 word vector models and layers of VGG 16. BnCm: mth conv. layer of the nth block.

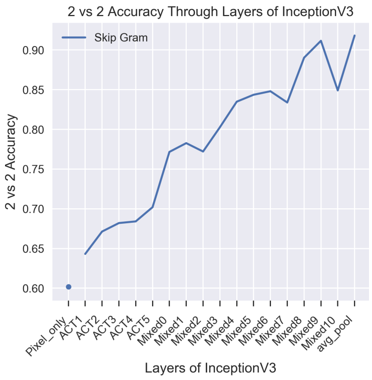

Figure 2: 2 vs. 2 accuracy for SkipGram and layers of Inception V3.

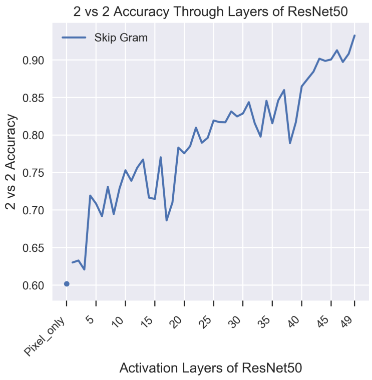

Figure 3: 2 vs. 2 accuracy for SkipGram and layers of ResNet 50.

tion V3 shows maximum accuracy of 0.90 at layer Mixed9, several layers before the final classification layer. ResNet 50 has maximum accuracy of 0.93 at the last layer, and VGG 16 peaks at the final layer with 0.94 accuracy. This implies that there may be a way to improve Inception V3 using, for example, a skip connection from the highest scoring layer to the final classification layer.

3.3 Misclassifications

We wondered if mistakes made in the ImageNet classification task could be detected, or even compensated for, using word vectors. Within one set of 166 images, we selected those images misclassified by VGG 16 such that they were misclassified into classes with a matching word vector (36 images). Could word vectors determine where in the network these misclassifications emerge? For this, we developed a variant of the 2 vs. 2 test: the **1 vs. 2 test**. For every misclassified image, there is a true and a predicted concept class (c_{true} and $c_{predicted}$, guaranteed to be different, since the image was misclassified). We selected the word vectors corresponding to c_{true} and $c_{predicted}$ and compute their correlation to all word vectors for which there was a corresponding correctly classified image. This creates vectors w_{true} and $w_{predicted}$ which represent correlations in word space. We also compute the correlations of the hidden representations for the misclassified image to the hidden representations of the correctly classified images to create a vector $i_{misclassified}$. The 1 vs. 2 test is considered to have passed if $i_{misclassified}$ is more correlated to w_{true} than to $w_{predicted}$. The 1 vs. 2 accuracy is the fraction of 1 vs. 2 tests passed, and chance is again 50%. The 1 vs. 2 test allows us to test if the classification mistake made at the final layer of the CNN is present through all of the hidden layers of the CNN.

Figure 4 shows the results for this experiment using layers from VGG 16. We see that in B4C1 and B5C1, the correlation of the hidden representations are, on average, significantly closer to the correlations of the correct rather than the predicted word vector ($p = 0.048$). But, during the last fully connected layer, this difference disappears, leading to the misclassification of the image. This implies that, for at least some of the misclassified images, the information required to make the correct prediction exists in the hidden representations, but the classification layer is not using it for the final prediction.

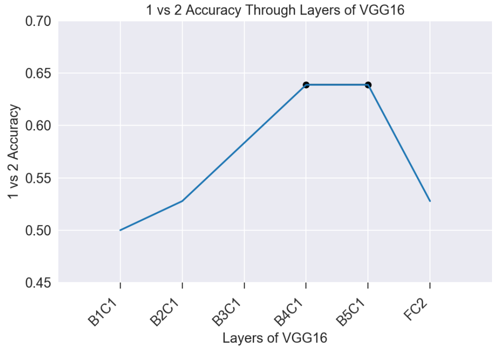

Figure 4: Results for the 1 vs. 2 test. For some layers of VGG 16, the correct class of misclassified images can be recovered (black dots).

4 Conclusion and Future Work

In this paper, we used methodology originally developed to analyze brain images to study semantic representations in CNNs. Our results point to several interesting possibilities for future work. The techniques explored here could be use to combat adversarial attacks on CNNs, detect misclassifications, or possibly guide the improvement of CNN architectures, and eventually help to unite the study of semantics in computer vision and computational linguistics.

5 Acknowledgments

This research was supported by CIFAR (Canadian Institute for Advanced Research) and NSERC (Natural Sciences and Engineering Research Council). This research was enabled in part by support provided by WestGrid (https://www.westgrid.ca/) and Compute Canada (www.computecanada.ca).

References

Andrew James Anderson, Benjamin D. Zinszer, and Rajeev D.S. Raizada. 2016. Representational similarity encoding for fMRI: Pattern-based synthesis to predict brain activity using stimulus-model-similarities. *NeuroImage* 128:44–53.

Elia Bruni and Marco Baroni. 2013. Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research* 48.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition* pages 2–9.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. *Proceedings of the European Association for Computational Linguistics* pages 462–471.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Kaiming He, Xiangyu Zhang, Shaoquing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385* pages 1–17.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. 2008. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in systems neuroscience* 2(November):4.

Aravindh Mahendran and Andrea Vedaldi. 2016. Visualizing Deep Convolutional Neural Networks Using Natural Pre-images. *International Journal of Computer Vision* 120(3):233–255.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)* pages 1–12.

Tomáš Mikolov, Stefan Kombrink, Anoop Deoras, Lukáš Burget, and Jan Černocký. 2011. RNNLM — Recurrent Neural Network Language Modeling Toolkit. In *Proceedings of Automatic Speech Recognition and Understanding (ASRU)*. pages 1–4.

Tom M Mitchell, Svetlana V Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L Malave, Robert A Mason, and Marcel Adam Just. 2008. Predicting human brain activity associated with the meanings of nouns. *Science (New York, N.Y.)* 320(5880):1191–5.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe : Global Vectors for Word Representation. In *Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar.

Carina Silberer and Mirella Lapata. 2012. Grounded models of semantic representation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pages 1423–1433.

Carina Silberer and Mirella Lapata. 2014. Learning Grounded Meaning Representations with Autoencoders. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pages 721–732.

Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint* pages 1–14.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. *arXiv preprint* .

Haoyan Xu, Brian Murphy, and Alona Fyshe. 2016. BrainBench : A Brain-Image Test Suite for Distributional Semantic Models. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)* pages 2017–2021.