CS230

Categorization of seen images from brain activity using sequence models

Arash Jamalian
Department of Computer Science
Stanford University
arashj@stanford.edu

Rafi Ayub
Department of Psychiatry
Stanford University
rafiayub@stanford.edu

Faraz Fadavi
Department of Computer Science
Stanford University
fadavi@stanford.edu

Abstract

How do different visual regions of the brain react to an image and are these activities any different when looking at an image of a dog versus a mountain? Are there specific voxels that are more receptive to living species than to objects? With the release of BOLD5000 data set, we have access to human functional MRI (fMRI) mapped to 5000 distinct images from Scene UNderstanding (SUN), COCO, and ImageNet datasets. Relative to prior datasets, BOLD5000 has significantly higher image count and diversity, providing a great opportunity to better understand neuron actives related to vision. In this paper we present an LSTM model for identifying what image a person is looking at from their brain activities over time. First we identify 3 super categories for our problem: Animal, Artifact, and Scene. Then we train an LSTM model on visual region voxel activities over time to predict these super categories and achieve an accuracy of 68%. Our results demonstrate an important step towards better understanding the visual processing systems of the human brain. [1]

1 Introduction

The human brain exhibits coordinated and hierarchical dynamics that gives rise to complex behaviors and high-order cognitive processes. It does so by encoding high-dimensional sensory information into the low-dimensional activity of neuronal populations. Understanding this encoding is critical in order to reverse engineer many of the brain's sensory systems. The visual system is a prime example of this. Visual processing in the brain is performed hierarchically at distinct parts of the cortex [1] [2]. Models of the human visual system perform optimally when this hierarchy is captured [1], and the success of hierarchical deep learning models such as ResNet [3] serve as a testimony to the importance of understanding this neural encoding.

Brain activity in response to visual stimuli can be measured using functional magnetic resonance imaging, or fMRI. Functional MRI exploits the difference in magnetic properties of oxygenated and deoxygenated blood to measure when neurons are more active. Thus, each voxel in an fMRI image with three spatial dimensions and one temporal dimension is a timeseries of the fluctuations of blood oxygenation as nearby neurons consume more or less oxygen. This is known as a BOLD timeseries, or blood oxygen level dependent timeseries. Images of the whole brain activity over time of participants that are shown visual stimuli while scanning can be used to elucidate the underlying low-level neural encoding. Many groups have categorized seen objects and even generated seen images directly from fMRI using deep neural networks, with varying degrees of success [4] [5] [6]. While the results are impressive, all of these studies either use the amplitude of BOLD activity at a certain timepoint or average across all timepoints, eliminating any temporal information. BOLD activity is know to have a slow, low frequency response and can peak 5-6 seconds after the onset of a stimulus [7]. Thus, compressing in the temporal dimension could remove important decoded information about the presented stimuli, especially for a complex stimulus such as an image. Sequential deep learning models can retain this information about the timecourse of the BOLD activity.

Here, we apply an LSTM classifier, a commonly used sequence model, to categorize images shown to subjects in an fMRI scanner from their fMRI timeseries data directly. Subjects were shown images from three computer vision datasets - ImageNet, COCO, and SUN - while their brain activity was scanned. Regional activity of brain areas involved in visual processing were

[1] Code for this paper can be found here (clickable)

CS230: Deep Learning, Fall 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

extracted and used as timeseries inputs to the model. We used the model to classify three categories - animal, artifact, and scene - and achieved an accuracy of 68%. Taken together, these results represent an important step forward in better understanding how the brain encodes diverse visual stimuli.

2 Related work

The original authors of the BOLD5000 dataset have initially explored the dimensionality of the fMRI data and how it compares to the visual stimuli. Chang et al. [8] also ran tSNE on each of the ROI's BOLD timseries data and found that the three databases were difficult to separate, indicating that either the fMRI data is noisy or brain activity response profiles to various visual stimuli are similar. They also compared ROI activations to individual layer activations in a pretrained AlexNet, demonstrating that there is some level of encoding occurring in each region. While this exploratory analysis is interesting, they do not compare ROI data with more biologically plausible deep learning models. Their analysis is also just the tip of the iceberg of understanding how each region is encoding visual information. We aim to extend their exploratory analysis and attempt to categorize the visual stimuli by further analyzing this encoding.

Two studies have applied similar deep learning approaches to categorize the visual stimuli. In Qiao et al. [9], the investigators used a bidirectional LSTM to represent bidirectional information flow in the cortex for visual processing. They achieved a maximum of about 60% accuracy with their model, which outperformed most other conventional machine learning classifier. While incorporating physiological information flow into a model may be useful, they only include visual areas that are involved in low-level processing of visual features, which may make distinguishing object categories challenging. Another study does use higher level visual areas and predicts hierarchical convolutional neural network features from ROI activations [10]. They achieve high accuracies of almost 90%. However, one aspect that we wish to focus more - compared to this study - is if the temporal information present in the BOLD timeseries is important for object identification.

Two other studies have gone a step further and attempted to generate images directly from fMRI data. Han et al. [4] demonstrates impressive results with a variational autoencoder (VAE) by using a linear regression model to map fMRI data to the latent variables in the VAE. They then can use the decoder of the VAE to generate an image from the latent variable representation of the fMRI data. This approach is straightforward and can generate images with less computational power as the other aforementioned studies, but the authors also discard temporal information and assume a linear relationship between the regional activity of the brain areas and the latent variables. The other study by Shen et al. [5] used a GAN approach to generate the image by mapping fMRI data to different layers of the discriminator network. This approach utilizes the strengths of adversarial training to better predict the seen image, but again they discard temporal information when encoding the fMRI data.

3 Dataset and Features

We used the public dataset BOLD5000. BOLD5000 consists of fMRI scans collected from 4 participants while viewing images from three widely used computer vision databases: ImageNet, COCO, and SUN. In total 5254 images were presented to each participant in 15 sessions. 4916 images were presented once, and 113 images were presented 3 times. The image distribution is 1916 from Imagenet, 2000 from COCO, and 1338 from SUN.

Images from ImageNet typically contain single, in-focus objects. Images from COCO contain one or multiple objects in a more naturalistic setting. Images from SUN are images of indoor and outdoor spaces, landscapes, cityscapes, and other scenery-like locations [8]. The BOLD5000 dataset had sub-category labels for each image. From ImageNet, two images from 958 sub-categories were shown to participants. We used wordnet [11] to map each of these sub-categories to one of the following super-categories: Artifact, Animal, Food, Plant, Scene, Communication, and Person. For COCO images, we used category IDs to map each image to one of COCO super-categories [12]: Animal, Person, Vehicle, Furniture, Kitchen, Sports, Indoor, Food, Electronics, Accessory, Outdoor, Appliance. All SUN images were put under single Scene category.

Images were shown to participants in ten second trials. The TR, or period, of the fMRI scan was two seconds, so each image presentation is associated with five timepoints. The fMRI data for every scan was preprocessed using fmriprep, an automated pipeline. Nuisance signals was additionally regressed out of the fMRI data according to a 16 parameter model; six motion parameters, average CSF signal, average white matter signal, and all their derivatives were regressed out. TRs or frames with excessive motion (greater than a framewise displacement of 0.5 mm) were censored. These steps are taken to ensure that the retained blood-oxygenation level dependent (BOLD) fMRI signal is solely due to gray matter neuronal activity and not due to motion or other biological noise.

Regions of interest that play low-level and high-level roles for visual processing were extracted from the fMRI data. Voxel timeseries were extracted from the parahippocampal place area, retrosplenial complex, occipital place area, lateral occipital complex, and early visual area in each hemisphere, resulting in ten total ROIs per subject. Each ROI timeseries was flattened from a 4D tensor to a 2D matrix where each row is a timepoint and each column is a voxel from that ROI. Since each subject's

ROIs had a different number of voxels, all ROIs were zero padded to the max number of a voxels in a single subject's ROI. For the LSTM classifier, the fMRI data was split into trials of ten seconds each, where each trial is associated with a visual stimuli or lack thereof. With a TR of two seconds - meaning the fMRI activity was sampled every two seconds - each sample was of dimensions 5 x number of ROI voxels. Extracting all the trials from all four subjects and separating into the three super-categories - animal, artifact, and scene - yielded a dataset of 13136 samples.

4 Methods

In order to better capture high-dimensional temporal information from the BOLD timeseries data, we utilized a specific type of recurrent neural network called the Long Short Term Memory (LSTM) model. LSTMs are widely used with sequential data since their unique architecture prevents vanishing gradients over long sequences, allowing it to retain information over lengthy time intervals [13].

Voxel activities in 10 visual regions (LHPPA, RHPPA, LHRSC, RHRSC, LHOPA, RHOPA, LHEarlyVis, RHEarlyVis, LHLOC, RHLOC were used as dataset. For each time step, the corresponding brain activities in these regions were concatenated together for input to LSTM. Our model had a total of 5 time steps, mapping brain activities to the stimuli image shown in the 10-second time window. The LSTM cells thus accept samples of dimensions 5x6960.

To interpret the encoding of sequential information by the LSTM cells, we used a single dense layer with ReLU activation after applying dropout and a final softmax layer to predict the three classes. The final architecture can been seen in Figure 1.

Weighted categorical cross entropy loss function was used for training the model. The weight factors were chosen to balance the number of samples in each subcategories: $w_{animal} = 1.16, w_{artifact} = 1.41, w_{scene} = 1$

$$Loss = -\sum_{c=1}^{M} w_c y_{o,c} log(\widehat{y}_{o,c})$$

Our final model was trained using mini-batch gradient descent with the Adam optimizer, learning rate of 0.0016, learning decay of 1e-3, minibatch size of 208, for 11 epochs. We also used early stopping callback, monitoring minimum validation loss. The LSTM layer used 12 hidden units, input dropout of 49% and recurrent dropout of 55% used. The dataset was shuffled and split 80-20 giving the 10508 training samples and 2628 dev samples.

5 Experiments/Results/Discussion

First, we sought to understand how separable the visual stimuli are in each of their categories. To achieve this, we applied a ResNet architecture pre-trained on ImageNet to extract hierarchical features from the images in our dataset. We forward propagated every image through the model and kept the weights from the last global average pooling layer. This yielded a 4096 element feature vector for each image. Then, t-distributed stochastic neighborhood embedding, or t-SNE, was applied to these feature vectors. t-SNE is a probabilistic dimensionality reduction technique that has been previously applied to top layers of convolutional neural networks to visualize the image space deep learning models are classifying [14] [15].

Next, we wanted to see how well our categories separated in this lower-dimensional embedding to assess the difficulty of the classification task. We colored each image by their category to visualize this separation (Figure 2). When the embedding is colored by the three super-categories, images seemed to form distinct clusters, though artifacts and scenes seem to have some overlap. However, when we added more categories, clusters become more difficult to distinguish. This could potentially explain the decreased accuracy of our classifier with a higher number of classes. For this reason, we decided to use the three

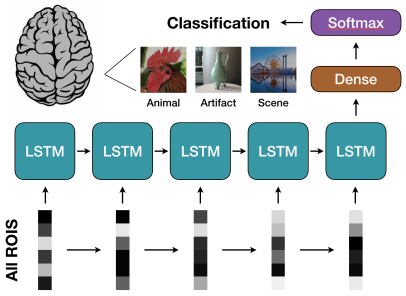

Figure 1: LSTM classifier model.

super-categories due to their ease of separation in a lower-dimensional embedding. Table 1b lists the super categories along with their image counts used for developing our model.

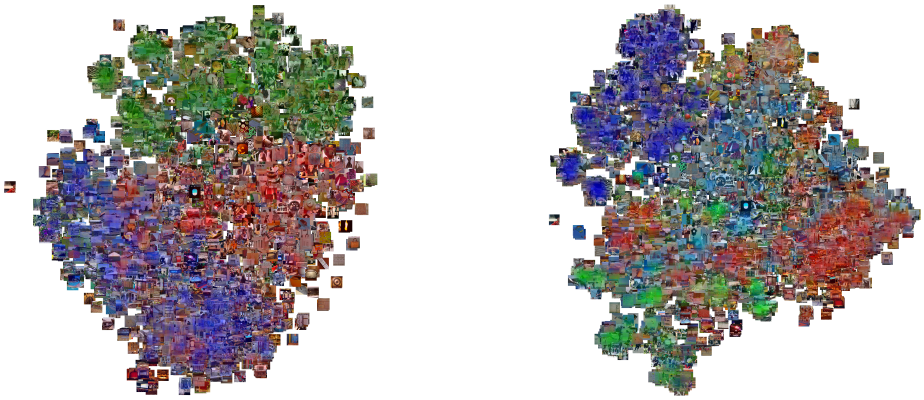

Figure 2: t-SNE embedding of images coloured by super categories (artifacts - red, animals - green, scenes - blue) (left) and sub categories (artifacts - teal, animals - blue, scenes - red) (right)

We used an AWS p2.xlarge EC2 instance with 61GiB RAM for training our model. When using ROIs for training, the data set size was 4GiB therefore we could pass the whole data to keras fit method. We coded the model in Python on a Jupyter Notebook using Keras [16], TensorFlow [17], Numpy [18], Matplotlib [19], and Scikit-learn [20].

Given the small data set of 13136 fMRI samples, our model would quickly overfit to the training data, causing the validation accuracy to start decreasing after few epochs. To help the model generalize, we used dropout on both the input and the recurrent states of the LSTM layer. In addition, we ran hyper parameter search by randomly sampling learning rate (log scale), recurrent dropout rate, input dropout rate, batch size, and number of LSTM hidden units (log scale) in each iteration. Table2 shows notable iterations in our hyper parameter search. We observed that lower learning rates and high dropout help our model achieve better accuracy. With high dropout rates, number of hidden units seem to have less influence in our model performance.

Figure3 shows the performance achieved with this model. Our LSTM model achieved a modest accuracy of 68% across all four subjects. We also ran the classifier on each subject individually, and found that they all performed relatively similar, except for an unusually high performance for subject one of 75% (Table3). Given the same images were shown to all participants, this indicates that the fMRI datasets are influenced by subject's physiological traits.

	Animal	Artifact	Scene	Subject Total
Subject 1	1227	1014	1420	3661
Subject 2	1227	1014	1420	3661
Subject 3	1227	1014	1420	3661
Subject 4	726	585	842	2153
Total	**4407**	**3627**	**5102**	**13136**

Table 1: Image count per selected super categories

Iteration	learning rate	Input dropout	Recurrent dropout	Batch size	LSTM hidden units	Val loss	Val accuracy
16	0.2317	0.54	0.27	234	4	1.14	0.581
28	0.0031	0.6	0.38	94	25	0.918	0.673
37	**0.0016**	**0.49**	**0.55**	**208**	**12**	**0.906**	**0.675**
39	0.0618	0.63	0.04	238	43	1.037	0.605
60	0.1296	0.27	0.39	62	155	1.29	0.325
81	0.0034	0.59	0.01	236	202	0.918	0.677
83	0.0980	0.18	0.33	57	248	1.297	0.396

Table 2: Hyper parameter tuning

Table 4 illustrates how using weighted loss influences the model performance for different categories. We observed that giving more weight to the artifact category slightly increases the artifact category recall but drops the recall in the other two categories. Animal and Scene categories were relatively easier to classify compared to artifact category. This could indicate that brain is processing living and scene objects differently from objects. We also ran experiments to classify five super categories (animal, artifact, scene, person, and food). With only 341 person samples and 230 food samples present, most samples from these two categories got miscategorized bringing down the model performance to 58%.

When training our model on higher level visual regions only, we could only achieve 62% validation accuracy. Including the lower level visual regions helped our model achieve 68% accuracy. This indicates that the higher level regions contain most of the relevant features, however including lower level regions still has a positive impact and helped out model achieve better accuracy.

6 Conclusion/Future Work

In this study, we applied an LSTM architecture to classify the category of visual stimuli shown to subjects in an fMRI scanner using only the activity of their visual brain areas. We achieved modest results according to our classification metrics. We explored how the fMRI data itself is not well separable in a lower-dimensional embedding despite the ease of class identification in a lower-dimensional embedding of the visual stimuli themselves. Our model had more difficulty categorizing artifacts and scenes than animals, and performed worse when more categories were introduced. While the performance is suboptimal, this work demonstrates that temporal information in fMRI data can still be important for visual stimuli categorization and represents one of the first approaches using temporal information.

Many of the difficulties we faced with the multiple approaches that we tried was the ease of the model to overfit the data. While typical approaches to reduce variance somewhat ameliorated this issue for our best model, most other approaches did not cooperate. The best solution for overfitting would be to acquire more data. Had we had more time, we would be able to process additional datasets, namely many of the ones used in the studies cited in the Related Work section, and significantly increase our training data set size. Our small training set was one of the key drawbacks of this study.

Beyond reducing overfitting, a next step would be to alter the architecture of the LSTM classifier, by perhaps using it in tandem with an autoencoder, to generate images from the encoded sequential information. This would be a unique approach in that it use temporal information to encode features from fMRI data that are needed to generate an image, which other studies have not attempted. This would be an interesting direction to pursue as well.

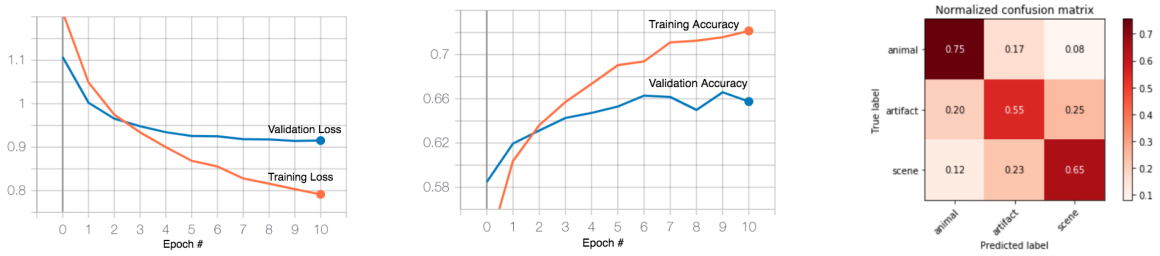

Figure 3: Training/validation loss, accuracy and confusion matrix of the final model

	Subject 1	Subject 2	Subject 3	Subject 4	All Subjects
Training Size	2928	2928	2928	1722	**10508**
Validation Size	733	733	733	431	**2628**
Training Loss	0.720	0.749	0.747	0.689	**0.793**
Validation Loss	0.777	0.983	0.928	0.943	**0.912**
Training Accuracy	0.756	0.747	0.749	0.773	**0.719**
Validation Accuracy	0.746	0.648	0.664	0.654	**0.675**

Table 3: Performance comparison when using dataset from individual subject vs. datasets from all subjects

	Weighted Loss			Normal Loss		
	Precision	Recall	F_1 Score	Precision	Recall	F_1 Score
Animal	0.70	**0.75**	0.73	0.70	**0.79**	0.74
Artifact	0.52	**0.55**	0.53	0.55	**0.51**	0.53
Scene	0.73	**0.65**	0.69	0.73	**0.69**	0.71

Table 4: Effect of weighted vs. normal categorical cross entropy loss on Precision, Accuracy and F_1 Score

7 Contributions

Arash Jamalian worked on labeling the images, developing the model architecture, and running hyper parameter search. Rafi Ayub researched the related work, pre-processed the fMRI data, and helped with with the model architecture and experiments. Faraz Fadavi did not contribute.

References

[1] Daniel L.K. Yamins et al. "Performance-optimized hierarchical models predict neural responses in higher visual cortex". In: *Proceedings of the National Academy of Sciences of the United States of America* 111.23 (2014), pp. 8619–8624. ISSN: 10916490. DOI: 10.1073/pnas.1403112111.

[2] Martin N. Hebart and Guido Hesselmann. "What visual information is processed in the human dorsal stream?" In: *Journal of Neuroscience* 32.24 (2012), pp. 8107–8109. ISSN: 02706474. DOI: 10.1523/JNEUROSCI.1462-12.2012.

[3] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (2016), pp. 770–778. ISSN: 10636919. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

[4] Kuan Han et al. "Variational autoencoder: An unsupervised model for encoding and decoding fMRI activity in visual cortex". In: *NeuroImage* 198.January 2018 (2019), pp. 125–136. ISSN: 10959572. DOI: 10.1016/j.neuroimage.2019.05.039.

[5] Guohua Shen et al. "Deep image reconstruction from human brain activity". In: *PLoS Computational Biology* 15.1 (2019), pp. 1–23. ISSN: 15537358. DOI: 10.1371/journal.pcbi.1006633.

[6] Guohua Shen et al. "End-to-end deep image reconstruction from human brain activity". In: *Frontiers in Computational Neuroscience* 13.April (2019). ISSN: 16625188. DOI: 10.3389/fncom.2019.00021.

[7] Gary H. Glover. "Overview of functional magnetic resonance imaging". In: *Neurosurgery Clinics of North America* 22.2 (2011), pp. 133–139. ISSN: 10423680. DOI: 10.1016/j.nec.2010.11.001.

[8] Nadine Chang et al. "BOLD5000, a public fMRI dataset while viewing 5000 visual images". In: *Scientific data* 6.1 (2019), p. 49.

[9] Kai Qiao et al. "Category Decoding of Visual Stimuli From Human Brain Activity Using a Bidirectional Recurrent Neural Network to Simulate Bidirectional Information Flows in Human Visual Cortices". In: *Frontiers in Neuroscience* 13.July (2019), pp. 1–15. DOI: 10.3389/fnins.2019.00692.

[10] Tomoyasu Horikawa and Yukiyasu Kamitani. "Generic decoding of seen and imagined objects using hierarchical visual features". In: *Nature Communications* 8.May (2017), pp. 1–15. ISSN: 20411723. DOI: 10.1038/ncomms15037. arXiv: 1510.06479. URL: http://dx.doi.org/10.1038/ncomms15037.

[11] George A. Miller. "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11 (Nov. 1995), pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: http://doi.acm.org/10.1145/219717.219748.

[12] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[13] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. ISSN: 08997667. DOI: 10.1162/neco.1997.9.8.1735.

[14] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605. ISSN: 02624079.

[15] Leon Yao and John Miller. "Tiny ImageNet Classification with Convolutional Neural Networks". In: (2015). URL: http://cs231n.stanford.edu/reports/leonyao%7B%5C_%7Dfinal.pdf.

[16] François Chollet et al. *Keras*. https://github.com/fchollet/keras. 2015.

[17] Martın Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[18] S. van der Walt, S. C. Colbert, and G. Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation". In: *Computing in Science Engineering* 13.2 (Mar. 2011), pp. 22–30. ISSN: 1558-366X. DOI: 10.1109/MCSE.2011.37.

[19] J. D. Hunter. "Matplotlib: A 2D Graphics Environment". In: *Computing in Science Engineering* 9.3 (May 2007), pp. 90–95. ISSN: 1558-366X. DOI: 10.1109/MCSE.2007.55.

[20] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.