NX414 - Mini-project Report

Kolly Florian, Mikami Sarah, Montlahuc Louise
Team 1

***Abstract*—This report presents our approach to predicting neural activity from IT neurons given visual stimuli. We explore various models, ranging from simple regression on pixel data to more sophisticated task- and data-driven neural network models. Our goal is to develop the most accurate model for predicting IT neural activity. By finetuning a pretrained ResNeXt-50 model on the neural data, we obtain a final R^2 score of 0.45.**

I. Regression on stimuli

Before trying complex models, we analysed the possibility of predicting neural activity directly from pixels. We use two simple family of models: linear regression and ridge regression. Due to the high-dimensional space of images, we also compute the result of those models after using PCA on the images, retaining the first 1000 principal components (PC). Note that we start by normalizing the images. The results are presented in Table I.

Regression	Param.	R^2 on train	R^2 on valid.
Linear	-	0.384	-0.041
Linear	PCA	0.466	-0.082
Ridge	$\alpha = 1$	0.999	-0.929
Ridge	PCA, $\alpha = 1$	0.466	-0.082
Ridge	$\alpha = \alpha_0$	0.168	0.087
Ridge	PCA, $\alpha = \alpha_0$	0.159	0.086

Table I
Results when doing regression on the stimuli (pixels)

We obtain $\alpha_0 \approx 615848$ by doing a grid search while keeping the same distribution of classes between the train and validation set. There is a clear overfitting of these methods on the training data, leading to an overall bad generalisation and hence bad scores on the validation set.

II. Regression on pretrained networks

To obtain better scores, we then try to predict the neural activity with a task-driven modeling approach. The hypothesis is that training a network to perform relevant behavioral task makes it learn representations that resemble those of the brain. Our selection of models is inspired by the leaderboard of Brain-Score [1], [2]. We chose six models, including some of the best performing ones, also taking into account the limited compute ressources at disposal: ResNet-18, ResNet-50 [3], ConvNeXt (base) [4], ViT (base) [5], ResNeXt-50 [6] and DinoV2 [7][1].

For each model, we select the last three layers (before the fully connected head) and save the activations when we pass the images through the model. We perform three types of probing on the activations in order to predict the IT neural activity: a simple linear regression, a ridge regression and a two-layers MLP. Table II lists the best R^2 score on the validation data per model, indicating which layer and probing method yield this score.

Model	Layer	Probing	R^2
ResNet-18	layer3	Ridge	0.269
ResNet-50	layer3	Ridge	0.369
ResNeXt	layer3	Ridge	0.391
ConvNeXt	layer7	Ridge	0.199
ViT	encoder	Ridge	0.306
DinoV2	norm	Ridge	0.319

Table II
Best result of probing pretrained layers

The decision of selecting the last three layers of each model was made after an analysis of the distribution of the explained variance per neuron with respect to the layer of a pretrained ResNet-50 network. Figure 1 clearly shows that the explained variance distribution improves as the depth of the model increases.

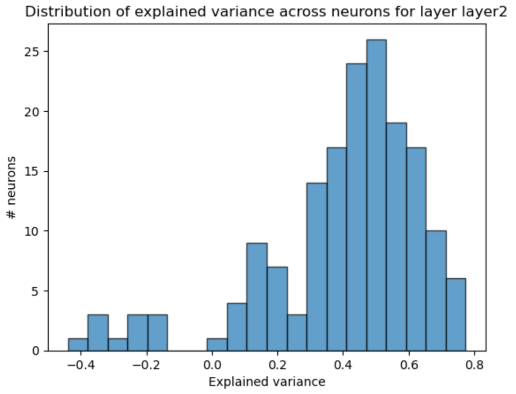

Figure 1. Distribution of explained variance across ResNet-50 depth

III. Finetuning pretrained models

The previous results indicate that ResNet-50 and ResNeXt are good candidates for finding the best model. Taking these models as backbone, we finetune them either I) a task-driven task (classify objects in images) or II) a data-driven task (regress neural activity given images). Specifically, we cut the network after a given layer and add either a classification or regression head. After experimenting with various training scheme and heads, we found our best result with an R^2 score of 0.45 using a data-driven approach. The backbone is given by ResNeXt-50 cut before layer 4, with a two-layers MLP preceded by an adaptive average pooling [8]. The training scheme consists of 30 epochs with a linear learning rate warmup during the first 5 epochs up to 10^{-6} followed by a cosine annealing scheduler down to 0.

[1] Thanks to the teaching team for this recommandation

References

[1] M. Schrimpf, J. Kubilius, H. Hong, N. J. Majaj, R. Rajalingham, E. B. Issa, K. Kar, P. Bashivan, J. Prescott-Roy, F. Geiger, K. Schmidt, D. L. K. Yamins, and J. J. DiCarlo, "Brain-score: Which artificial neural network for object recognition is most brain-like?" *bioRxiv preprint*, 2018. [Online]. Available: https://www.biorxiv.org/content/10.1101/407007v2

[2] M. Schrimpf, J. Kubilius, M. J. Lee, N. A. R. Murty, R. Ajemian, and J. J. DiCarlo, "Integrative benchmarking to advance neurally mechanistic models of human intelligence," *Neuron*, 2020. [Online]. Available: https://www.cell.com/neuron/fulltext/S0896-6273(20)30605-X

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[4] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *CoRR*, vol. abs/2201.03545, 2022. [Online]. Available: https://arxiv.org/abs/2201.03545

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[6] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *CoRR*, vol. abs/1611.05431, 2016. [Online]. Available: http://arxiv.org/abs/1611.05431

[7] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2024. [Online]. Available: https://arxiv.org/abs/2304.07193

[8] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," *CoRR*, vol. abs/1803.01534, 2018. [Online]. Available: http://arxiv.org/abs/1803.01534