

Decoding Brain-Like Representations with a Generative Adversarial Network

Submission ID 3000257

Submission Type Poster

Topic Neuroscience

Status Submitted

Submitter Ghislain St-Yves

Affiliation Medical University of South Carolina, Department of Neurosciences

SUBMISSION DETAILS

Presentation Type Poster Presentation

Presentation Abstract Summary We consider the inference problem of reconstructing a visual stimulus from a set of neural activity (e.g. fMRI) that encode this stimulus. Recovering a complete image is complicated by the fact that neural representations are noisy, high-dimensional, and contain incomplete information about image details. Thus, reconstructions have to rely on a strong prior over the class of images being reconstructed. Here we propose to train generative adversarial networks (GANs) to learn a generative model of images that is conditioned on observations of brain activity. We consider two challenges of this approach: 1) Given that GANs require far more data to train than is typically collected in an fMRI experiment, how do we obtain enough voxel measurements to train a GAN that is conditioned on brain activity? 2) How do we ensure that the GAN is robust against noise present in fMRI data? Using a synthetic example we show that these challenges can be surmounted using a sufficiently accurate encoding model. We find that we are able to control the samples generated by the GAN to reconstruct images, and show that variability of the GAN samples is dependent on signal-to-noise, receptive field sizes and tuning to low and high-level visual features.

Paper Upload (PDF) [ccn_styvesg_short_paper.pdf](#)

Co-author Information

* Presenting Author

First Name	Last Name	Affiliation	E-mail
Ghislain *	St-Yves *	Medical University of South Carolina, Department of Neurosciences	stayves@musc.edu

Thomas	Naselaris	Medical University of South Carolina, Department of Neurosciences	tnaselar@musc.edu
--------	-----------	--	-------------------

Keywords

Keywords
fMRI
Stimulus inference
generative model