Explorations in Feature Visualization with Optimization

Dawn Finzi
CS 229 – Fall 2019

Motivation

Feature visualization has recently become a popular area of research in both machine learning [1,2] and neuroscience [3,4]

Machine Learning: Motivated by a need for improved interpretability of neural networks. Progress toward refuting the "black box" critique.

Neuroscience: Motivated by a desire to understand and control how the brain works. In the last year or so, a flurry of papers have come out which probe the nature of neural computations in visual cortex by using convolutional neural networks to optimize images for neurons in macaque visual cortex [3,4].

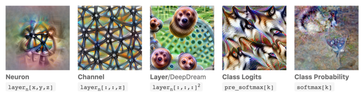


Fig. 1) Feature visualization using optimization for different aspects of a networks (from https://distill.pub/2017/feature-visualization/)

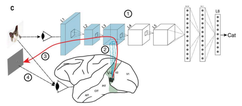


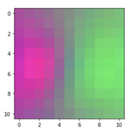
Fig. 2) Broad outline of the method used to drive neuronal responses in [2]

However, to date no one has leveraged this approach in humans where we would aim to target downstream object-selective cortical regions using noisier data (fMRI BOLD signal). Before that can even be attempted, we need to explore the space of potential visualizations, namely:

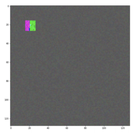
1. How do design decisions impact the results? Particularly, how do different loss functions and preprocessing choices alter the end visualization?
2. How reliable across iterations are the images generated by this partly stochastic process?
3. So far neuroscience research has focused on AlexNet but more biologically plausible networks have started to emerge (TNN) [5]. How do the features compare across these two networks, AlexNet and TNN, which have different architectures but are both trained on the same task (ImageNet)?

Sanity checks with Alexnet

To begin with, we implemented a simple version of gradient ascent feature visualization to use on single units in the first convolutional layer of AlexNet (pre-trained on ImageNet). Since there are no non-linearities in this layer, we have ground truth and the loss decreases linearly over iterations.



(a) True filter



(b) Optimized filter

(c) Loss over time

Implementation

- Create a random noise image tensor to optimize
- Instantiate pre-trained CNN of choice from checkpoint
- Create a loss tensor based on the *negative* of the model output to the image tensor for the layer and channel of interest, with added L2 regularization and total variation loss as follows:

$$\mathcal{L}(x, F) = -\frac{1}{m}\sum_{i=1}^{m}(F^{[i]}(x)) + \lambda\|x\|_2^2 + \sum_{i,j}\sqrt{(x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2}$$

where F represents the feature activation and x is the input image, where $x \in \mathbb{R}^{H \times W}$

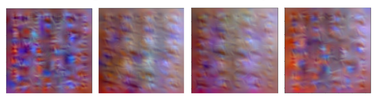
- Optimize the input image iteratively using gradient descent (Adam optimizer)

Output: An image that has been optimized to maximally activate a particular aspect of the network

Comparisons within and across networks

Repeated optimizations of the same channel and layer look visually similar:

Four different optimizations for AlexNet layer conv5, channel 6

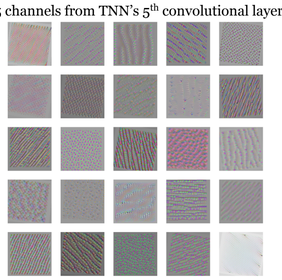


"pick" "black widow" "hotpot" "airliner"

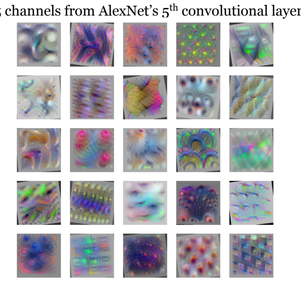
However, they are classified extremely differently if fed back into a pretrained network

Visualizations are very different for different networks:
ex) 'TNN': a biologically inspired neural network with 10 convolutional layers compared to AlexNet's 5 convolutional layers

25 channels from TNN's 5th convolutional layer



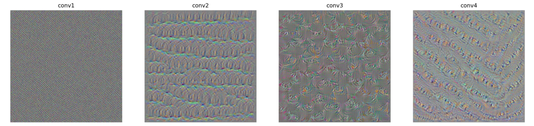
25 channels from AlexNet's 5th convolutional layer



More layers mean that more complex representations don't develop until later layers of the network

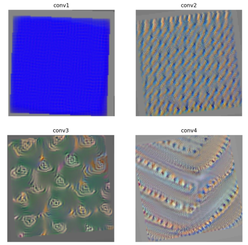
Large effects of preprocessing and loss

Feature visualization in AlexNet without preprocessing

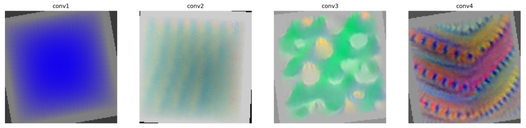


The exact same channels and layers using the Lucid [2] preprocessing recipe:

- Padding by 12 pixels to avoid edge artifacts
- Jittering by 8 pixels
- Randomly scaling by a factor of one of [0.9, 0.92, 0.94, 0.96, 0.98, 1.0, 1.02, 1.04, 1.06, 1.08, 1.1]
- Rotating by an angle of one of [-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] degrees
- Jittering for a second time for 4 pixels



And again with both preprocessing and total variation loss:



Future work

- Determine the distinct effects of individual current preprocessing steps, as well as other preprocessing options
- Examine features from more specialized networks, such as FaceNet

References:
1. Mordvintsev, A., Olah, C., & Tyka, M. (2015). Inceptionism: Going deeper into neural networks.
2. Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. *Distill*, *2*(11), e7.
3. Bashivan, P., Kar, K., & DiCarlo, J. J. (2019). Neural population control via deep image synthesis. *Science*, *364*(6439)
4. Ponce, C. R., Xiao, W., Schade, P. F., Hartmann, T. S., Kreiman, G., & Livingstone, M. S. (2019). Evolving images for visual neurons using a deep generative network reveals coding principles and neuronal preferences. *Cell*, *177*(4), 999-1009.
5. Nayebi, A., Bear, D., Kubilius, J., Kar, K., Ganguli, S., Sussillo, D., ... & Yamins, D. L. (2018). Task-Driven convolutional recurrent models of the visual system. In *Advances in Neural Information Processing Systems* (pp. 5290-5301).