SAMPLING THE "INVERSE SET" OF A NEURON

Suryabhan Singh Hada and **Miguel Á. Carreira-Perpiñán**,
EECS, UC Merced

BayLearn
Bay Area Machine Learning Symposium

1 Motivation and summary

With the recent success of deep neural networks in computer vision, it is important to understand the internal working of these networks. What does a given neuron represent? The concepts captured by a neuron may be hard to understand or express in simple terms. The approach we propose in this paper is to characterize the region of input space that excites a given neuron to a certain level; we call this the *inverse set*. This inverse set is a complicated high dimensional object that we explore by an optimization-based sampling approach. Inspection of samples of this set by a human can reveal regularities that help to understand the neuron. This goes beyond approaches which were limited to finding an image which maximally activates the neuron or using Markov chain Monte Carlo to sample images [2], but this is very slow, generates samples with little diversity and lacks control over the activation value of the generated samples. Our approach also allows us to explore the intersection of inverse sets of several neurons and other variations.

Work supported by NSF award IIS–1423515, by a UC Merced Faculty Research Grant, by computing in the MERCED cluster (NSF grant ACI–1429783), and by NVIDIA GPU donation.

2 The inverse set of a neuron, and how to sample it

We say an input $\mathbf{x}$ is in the inverse set of a given neuron having a real-valued activation function f if it satisfies the following two properties:

$$z_1 \leq f(\mathbf{x}) \leq z_2 \qquad \mathbf{x} \text{ is a valid input} \qquad (1)$$

where z_1, $z_2 \in \mathbb{R}$ are activation values of the neuron. $\mathbf{x}$ being a valid input means the image features are in the valid range (say, pixel values in [0,1]) and it is a realistic image.

For a simple model, the inverse set can be calculated analytically. For example, consider a linear model with logistic activation function $\sigma(\mathbf{w}^T\mathbf{x} + c)$ and all valid inputs to have pixel values between [0,1]. For $z_2 = 1$ (maximum activation value) and $0 < z_1 < z_2$, the inverse set will be the intersection of the half space $\mathbf{w}^T\mathbf{x} + c \geq \sigma^{-1}(z_1)$ and the [0,1] hypercube.

In general for deep neural networks we approximate the inverse set with a sample that covers it in a representative way. A simple way to do this is to select all the images in the training set that satisfy eq. (1), but this may rule out all images. A neuron may "like" certain aspects of a training image without being sufficiently activated by it. Therefore, we need an efficient algorithm to sample the inverse set.

3 Sampling the inverse set of a neuron: an optimization approach

To create a sample $\mathbf{x}_1, \dots, \mathbf{x}_n$ that covers the inverse set, we transform eq. (1) into a constrained optimization problem:

$$\operatorname*{arg\,max}_{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n} \sum_{i,j=1}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \quad \text{s.t.} \quad z_1 \leq f(\mathbf{x}_1), \dots, f(\mathbf{x}_n) \leq z_2.$$

The objective function makes sure that the samples are different from each other but also satisfy eq. (1).

However, this generates noisy-looking samples. To make them realistic we use an image generator network **G**, which has been empirically shown to produce realistic images when a code vector **c** is passed as an input. We also observe that using Euclidean distances directly on the generated images is very sensitive to small changes in their pixels. Instead, we compute distances on a low-dimensional encoding $\mathbf{E}(\mathbf{G}(\mathbf{c}))$ of the generated images constructed by an Encoder **E** .Then we have our final formulation of the optimization problem over the n samples $\mathbf{G}(\mathbf{c}_1), \dots, \mathbf{G}(\mathbf{c}_n)$:

$$\operatorname*{arg\,max}_{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_n} \sum_{i,j=1}^{n} \|\mathbf{E}(\mathbf{G}(\mathbf{c}_i)) - \mathbf{E}(\mathbf{G}(\mathbf{c}_j))\|_2^2$$
$$\text{s.t.} \quad z_1 \leq f(\mathbf{G}(\mathbf{c}_1)), \dots, f(\mathbf{G}(\mathbf{c}_n)) \leq z_2$$

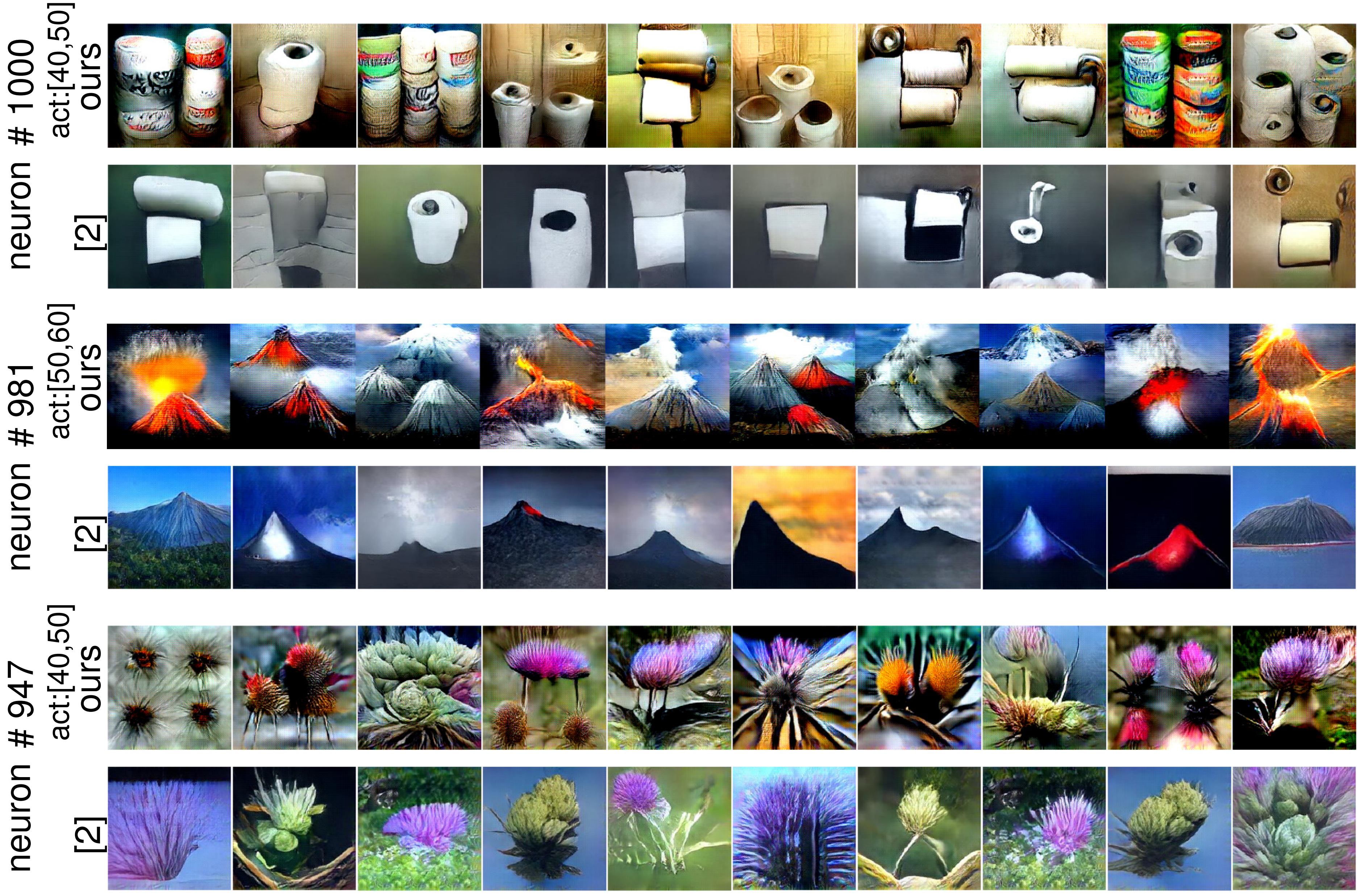
4 Computational solution of the optimization problem

We optimize the final objective function using the augmented Lagrangian method. Because of the quadratic complexity of the objective function over the number of samples n, it is computationally expensive to generate many samples. We apply two approximations to speed up the sampling process.

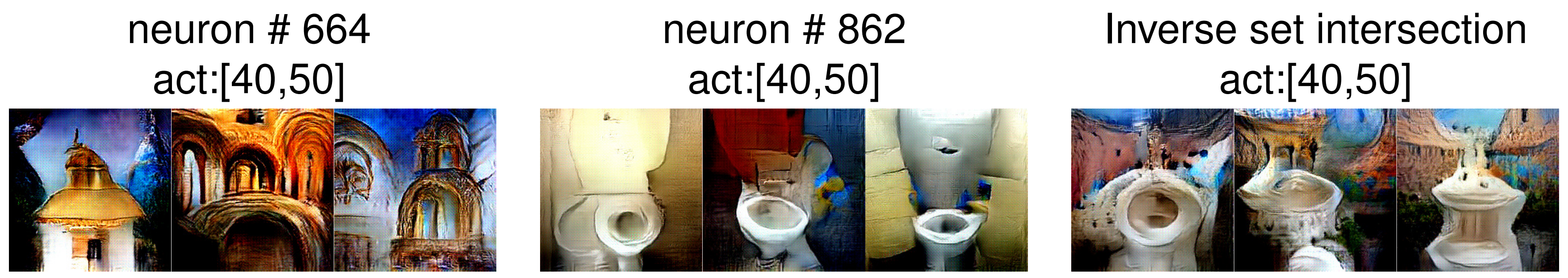
- Firstly. we solve the problem in an inexact but good enough way. The sum-of-all-pairs objective is not a strict necessity, it is really a mechanism to ensure diversity of the samples and coverage of the inverse set. We observe that this is already achieved by stopping the optimization algorithm once the samples enter the feasible set, by which time they already are sufficiently separated.
- Second, we create the samples incrementally, K samples at a time (with $K \ll n$). For the first K samples we optimize objective function, initializing the code vectors **c** with random values and stopping as soon as all K samples are in the feasible region. These samples are then fixed. The next K samples use as objective plus their distances to the seeds. We initialize them to the previous K samples and take a single gradient step in the augmented Lagrangian optimization, so that the new samples move further away from both the seeds and each other while staying feasible. These gives K new samples which we fix, and the process is repeated until we generate the desired n samples.

5 Experiments

Below we show some results of our sampling approach to create the inverse set for different neurons from CaffeNet [1]. Compared to the samples from [2], our samples are much more diverse as shown in below figure.



Rows 1, 3 and 5: 10 samples picked from 500 samples generated by our sampling approach to cover the inverse set for neuron number 1000 (class: toilet paper) in the activation range [40,50] and neuron number 981 (class: volcano) in the activation range [50,60] and neuron number 947 (class: cardoon) in the activation range [40,50], respectively . All three neurons are from layer fc8 of CaffeNet [1]. **Rows 2, 4 and 6**: samples generated for the same neurons by the sampling approach from [2]. Their activation values are not guaranteed to be in any fixed range.



Sampling the intersection of two inverse sets. The sample images from left to right are from the inverse set of neuron 664 (class: monastery), of neuron 862 (class: toilet seat) and of their intersection, all in the activation range [40,50]. Both neurons are from layer fc8 of CaffeNet [1].

7 References

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Dar- rell: Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093 [cs.CV]*, July 20 2014.

[2] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. *(CVPR 2017)*