Sampling the "Inverse Set" of a Neuron

Suryabhan Singh Hada   Miguel Á. Carreira-Perpiñán
{shada,mcarreira-perpinan}@ucmerced.edu

Electrical Engineering and Computer Science,
University of California, Merced
http://eecs.ucmerced.edu

Introduction

- Deep neural nets are accurate black-box models. They have shown much success in many applications such as computer vision and natural language processing.
- This makes it necessary to understand the internal working of these networks. What does a given neuron represent?
- We solve this by characterizing 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 using an optimization-based sampling approach. Inspection of samples of this set by a human can reveal regularities that help to understand the neuron.

Inverse set definition

- 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:

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

  where $z_1,\ z_2 \in \mathbb{R}$ are activation values of the neuron.

- For example, consider a linear model with weight vector $(\mathbf{w})$, bias (b), logistic activation function $\sigma(\mathbf{w}^T\mathbf{x} + b)$ 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.

Inverse set for a neuron in a deep neural network

- 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.
- Therefore, we need an efficient algorithm to sample the inverse set.

Sampling the inverse set: an optimization approach

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

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

- The objective function ensures that the samples are different from each other and satisfy eq. (1).
- It has two issues. The generated images are noisy and are very sensitive to small changes in their pixels.

Sampling the inverse set: an optimization approach

- We solve the issues in following way:
  - To counter the noisy image issue, we use generator network $\mathbf{G}$ to generate images from a code vector $\mathbf{c}$.
  - For the second issue, we compute distances on a low-dimensional encoding $\mathbf{E}(\mathbf{G}(\mathbf{c}))$ of the generated images constructed by an encoder $\mathbf{E}$.
- Our final formulation for generating n samples.

$$\arg\max_{\mathbf{c}_1,\mathbf{c}_2,\cdots,\mathbf{c}_n} \sum_{i,j=1}^{n} \|\mathbf{E}(\mathbf{G}(\mathbf{c}_i)) - \mathbf{E}(\mathbf{G}(\mathbf{c}_j))\|_2^2$$

$$\text{s.t.} \quad z_1 \leq f(\mathbf{G}(\mathbf{c}_1)), \ldots, f(\mathbf{G}(\mathbf{c}_n)) \leq z_2.$$

Computation constraints

- Because of the quadratic complexity of the objective function over the number of samples n, it is computationally expensive to generate many samples.
- It involves optimizing all code vectors $(\mathbf{c})$ together; for larger n, it is not possible to fit all in the GPU memory.
- Two approximation:
  - Stop the optimization algorithm once the samples enter the feasible set, as, by that time, the samples are already separated.
  - Create the samples incrementally, K samples at a time (with $K \ll n$).

Faster sampling approach

- Optimize the objective function for the first K samples, initializing the code vectors $\mathbf{c}$ with random values. We stop the optimization once the samples are in the feasible set. These samples are then fixed (called seeds $\mathbf{C}_0$).
- The next K samples are generated by the following equation:

$$\underset{\mathbf{c}_1,\mathbf{c}_2,\cdots,\mathbf{c}_K}{\arg\max} \sum_{i,j=1}^{K} \|\mathbf{E}(\mathbf{G}(\mathbf{c}_i)) - \mathbf{E}(\mathbf{G}(\mathbf{c}_j))\|_2^2 +$$

$$\sum_{i=1}^{K} \sum_{y=1}^{|\mathbf{C}_0|} \|\mathbf{E}(\mathbf{G}(\mathbf{c}_i)) - \mathbf{E}(\mathbf{G}(\mathbf{c}_y))\|_2^2$$

$$\text{s.t.} \quad z_1 \leq f(\mathbf{G}(\mathbf{c}_1)), \ldots, f(\mathbf{G}(\mathbf{c}_K)) \leq z_2 \text{ and } \mathbf{c}_y \in \mathbf{C}_0.$$

- We initialize them with the previous K samples and take a single gradient step in the feasible region. The resultant samples are the new K samples.

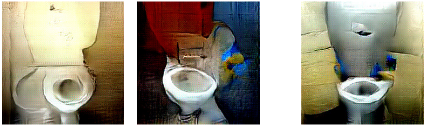
Experiments

- neuron # 981 volcano class.

Inverse set Intersection

neuron #664 (monastery), [50,60]

neuron #862 (toilet seat), [50,60]



Inverse set Intersection

Conclusion

- The goal of understanding what a neuron in a deep neural network may be representing is not a well-defined problem.
- For some neurons, their preferred response does correlate well with intuitive concepts or classes, such as the example of volcano class.
- By characterizing a neuron's preference by a diverse set of examples, we can explain this preference in a more holistic way.
- Our sampling method also has more general applicability; just by modifying the constraints, it can also be used for high dimensional sampling in other domains.

Thank You !