Title: Biologically Inspired Robust Perception Approximating Sparse Coding

Author(s): Rego, Jocelyn Rose
Kim, Edward
Kenyon, Garrett

Intended for: ISTI final presentation

Issued: 2021-09-14

Biologically Inspired Robust Perception
Approximating Sparse Coding

Jocelyn Rego, ISTI Student Fellow

Garrett Kenyon PhD, LANL PI

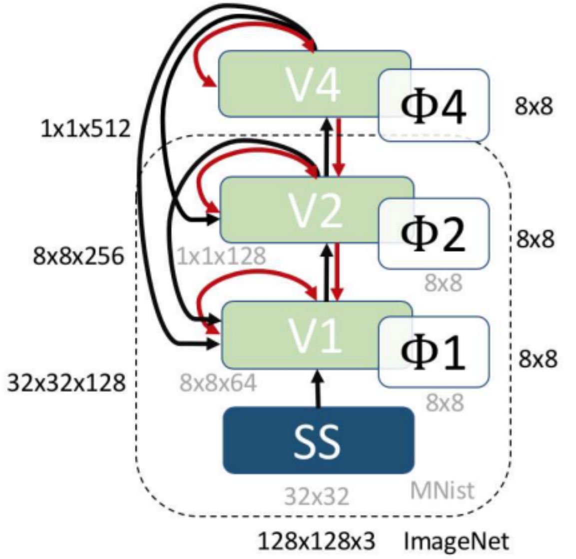
Edward Kim PhD, Drexel University PI

Motivation and Recent Findings

- Sparse coding has a variety of applications, such as image denoising and data compression.
- Recent work has shown the how sparse coding and biologically inspired architectures can mitigate adversarial attacks on images.

Modeling Biological Immunity to Adversarial Examples

Edward Kim[1], Jocelyn Rego[1], Yijing Watkins[2], Garrett T. Kenyon[2]
[1]Department of Computer Science, Drexel University, PA
[2]Los Alamos National Laboratory, Los Alamos, NM



(c) Visual Cortex Model

Sparse Coding

- Sparse coding: the process of finding a sparse subset of overcomplete dictionary elements along with corresponding coefficients to effectively reconstruct a signal
- Essentially, the goal is to find a good reconstruction that is also sparse.

$$E = \frac{1}{2} \sum_{x,y} \left[I(x,y) - \sum_i a_i \, \phi_i(x,y) \right]^2 + \lambda \sum_i C(a_i)$$

- Biological sensory systems, like V1, seem to employ sparse representations with population codes, allowing for complex stimuli to be encoded in the activity of a few neurons.

Sparse Coding with LCA

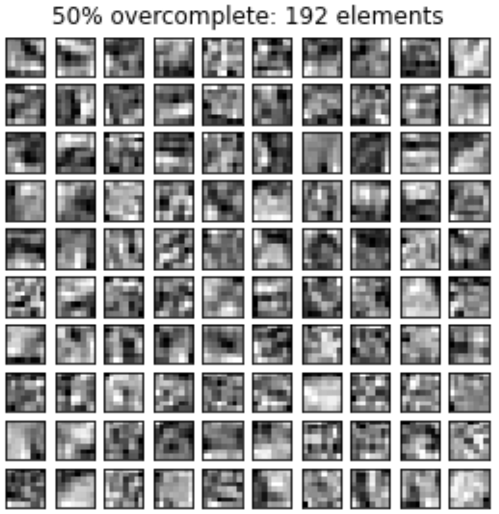
- “Locally Competitive Algorithms” solve the sparse coding problem with a dynamical system of nonlinear ordinary differential equations, through local competition and thresholding.

$$\dot{\boldsymbol{u}}(t) = f(\boldsymbol{u}(t)) = \frac{1}{\tau}\left[\boldsymbol{b}(t) - \boldsymbol{u}(t) - \left(\Phi^t \Phi - I\right) \boldsymbol{a}(t)\right],$$

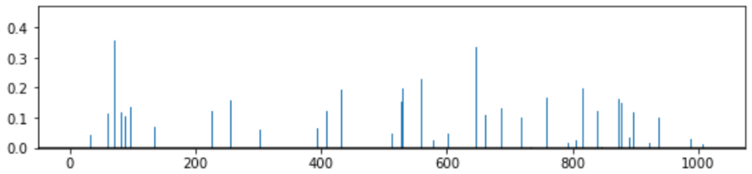
$$\boldsymbol{a}(t) = T_\lambda(\boldsymbol{u}(t)) .$$

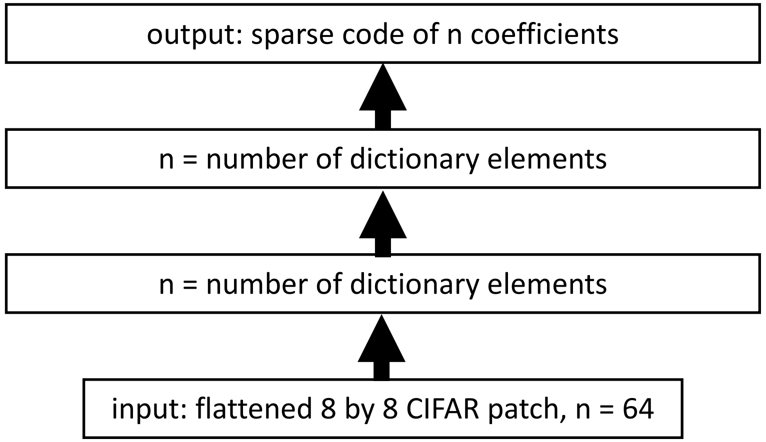
- Thresholding functions limit the lateral inhibition by only allowing highly active units to suppress others and forcing most coefficients to be identically zero.
  - Here, we use a soft-thresholding, corresponding to an L1 sparsity penalty.
- LCAs display properties necessary for a neurally plausible sparse coding algorithm, such as stability, sparsity, and the ability to handle time varying stimuli.

Fast LCA Approximation



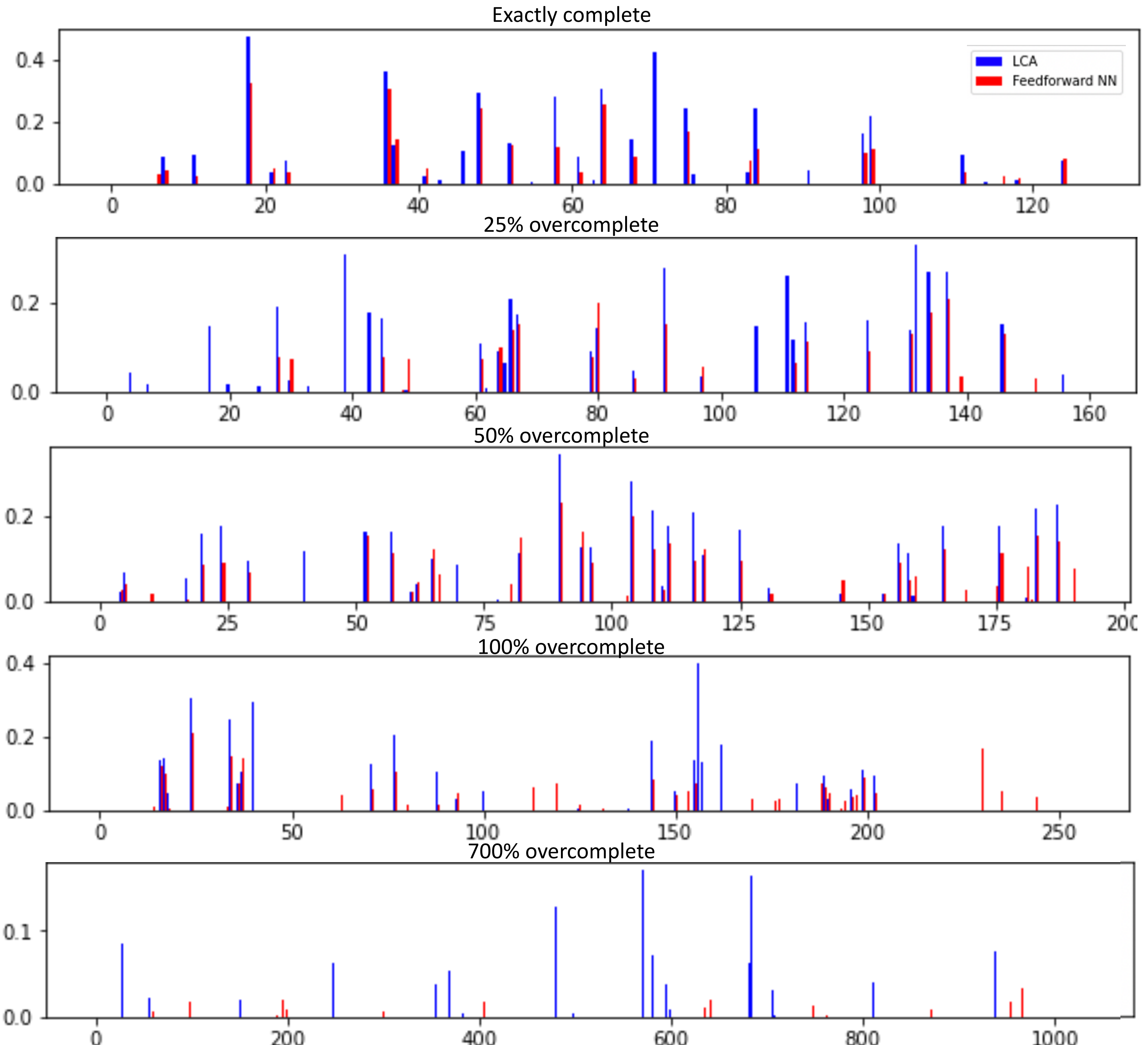
- The problem: sparse coding is slow
- Proposed solution: estimate sparse codes more quickly with feedforward neural networks, trained on pre-computed sparse code examples
- Implementation
  - Learned 5 dictionaries of increasing overcompleteness
    - 128, 160, 192, 256, 1024 elements
  - Inferred sparse codes of 8 by 8 grayscale CIFAR image patches via positive LCA with each dictionary
  - Trained 5, 2 hidden-layer feedforward neural networks on these LCA computed codes
    - Nodes in each hidden layer = number of output dictionary elements



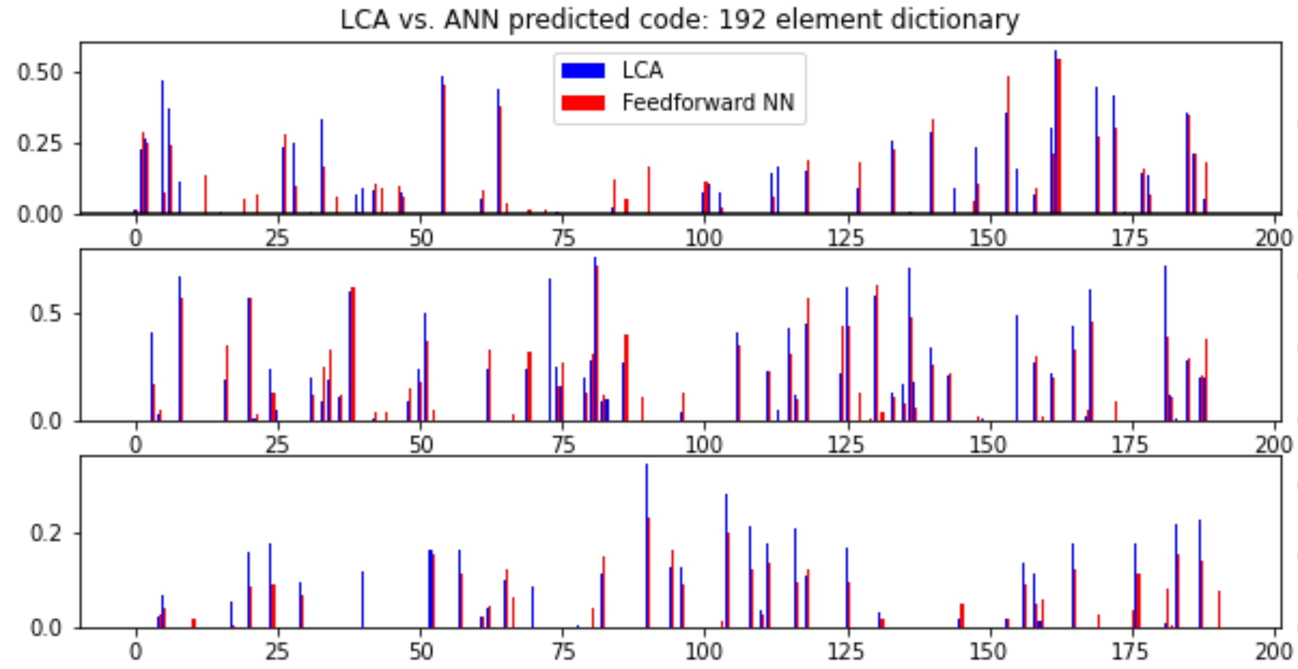


ANN predicted LCA

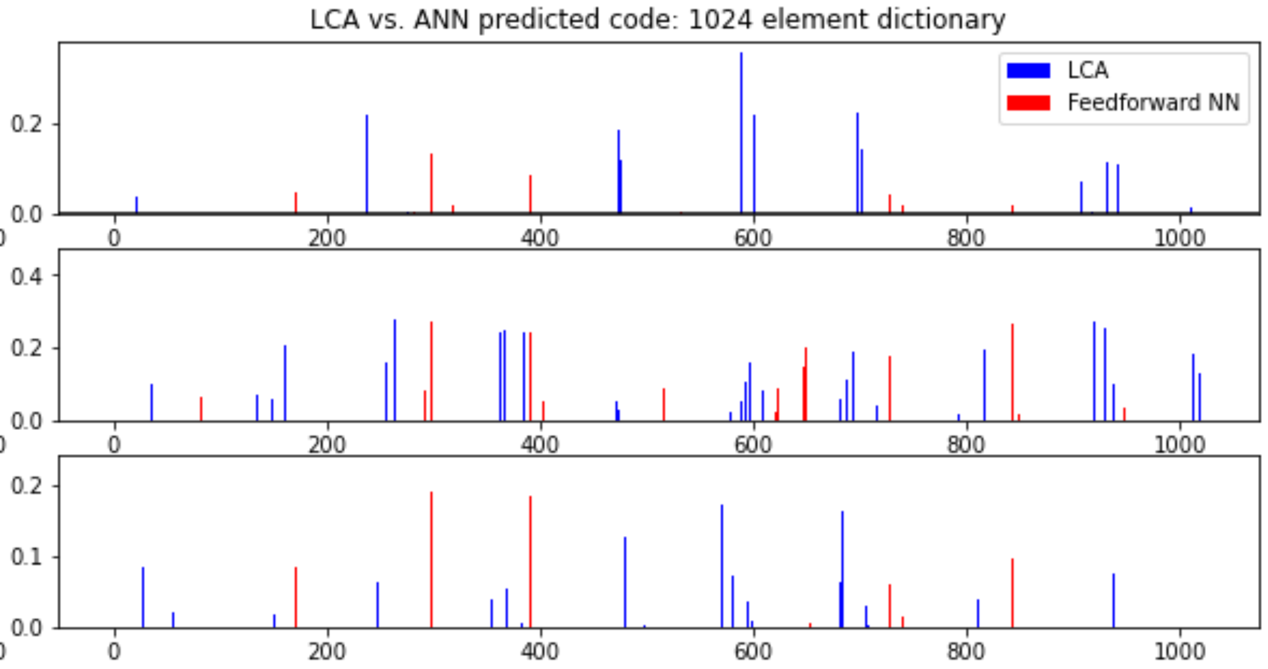
- The ability of a feedforward, fully connected neural network to estimate the solution to a sparse coding approximation (LCA), diminishes as the dictionary used to reconstruct signals becomes more overcomplete.



ANN predicted LCA



50% overcomplete dictionary (1.5 times complete)



700% overcomplete dictionary (8 times complete)

Future Work

- Can neural networks reach a more accurate LCA approximation by learning fewer inference steps at a time, progressively reaching the solution?
  - If so, how much does the problem have to be broken down to be solved?
- Approximating sparse coding with neural networks has the potential to greatly speed up an approximation of sparse code inference.

Thank you!