

Poster presentation

Open Access

## Sparse coding with population sketches

Eva L Dyer\*, Don H Johnson and Richard G Baraniuk

Address: Electrical and Computer Engineering Department, Rice University, TX, USA

Email: Eva L Dyer\* - e.dyer@rice.edu

\* Corresponding author

from Eighteenth Annual Computational Neuroscience Meeting: CNS\*2009  
Berlin, Germany. 18–23 July 2009

Published: 13 July 2009

BMC Neuroscience 2009, **10**(Suppl 1):P132 doi:10.1186/1471-2202-10-S1-P132

This abstract is available from: <http://www.biomedcentral.com/1471-2202/10/S1/P132>

© 2009 Dyer et al; licensee BioMed Central Ltd.

Sparsity plays two very important roles in sensory neural information processing. At the cellular level, the sparse activation of cells within a population aids in minimizing the total metabolic cost required to represent sensory stimuli. At the population level, sparse codes provide a concise representation of complex stimuli, enabling efficient learning and storage/recall of memories. In models for sparse coding in the primary visual cortex (V1), the receptive fields of simple cells in the input layer of V1 serve as the basis from which a sparse representation of the visual environment is constructed. Recently, a number of neurally plausible mechanisms for sparse coding were developed [1], yielding testable predictions of cortico-cortical interactions of cells within a population by inducing local competition to minimize the total number of active cells in the network. Although this algorithmic framework provides an explicit mechanism for simple cell sparse coding, many still contend that current models for sparse coding are not plausible due to the overhead required to transmit both the locations and spike rates of all active cells within the population to higher areas in the cortex.

With this in mind, we set out to develop a novel model for information processing in V1 that encourages sparse responses at the cellular level while also producing a sparse distributed representation that may be easily passed on to higher cortical areas. Taking a number of studies of information flow and connectivity into account, we have developed a new mathematical framework that solves connectivity constrained sparse coding problems; in this setting, a number of distinct computational units are presented the same input and must inde-

pendently approximate the incoming signal and then efficiently minimize the number of these units needed to produce the final sparse representation. Our proposed sparse coding framework is inspired by the organization of simple cells with common receptive field features into densely connected networks known as minicolumns. In addition to minimizing long-range connections, we hypothesized that neural systems may take advantage of columnar connectivity constraints by computing a high-level sketch of each microcircuit's response to an impending stimulus. Any model that describes the computation of this distributed *population sketch* must ensure that: "information" needed for complex inference tasks is preserved and that these tasks be performed without requiring knowledge of the single cell responses underlying the observed population activity.

The most natural choice for the structures that will serve as the basis of our new representation, are the orientation minicolumns in V1. These densely connected networks consist of a number of simple cells tuned to the same orientation as well as other types of excitatory cells and interneurons. Connecting these distributed microcircuits is a small network of cells in layers II/III that we will refer to as a *fusion network*. The function of this network is still not known, however, studies of information flow in the cortex suggest that cells in these upper layers may provide a means for competition to occur amongst neighboring columns. To obtain a sketch from our model of this hypercolumn, we employed locally competitive algorithms for sparse coding [1] within each orientation minicolumn; each of these approximations are then fed

forward to the *fusion network* which solves a sparse approximation problem to determine the minimum number of minicolumns needed to effectively represent the stimulus. We call the sparse weight vector that emerges from this computation our *population sketch*.

To show the viability of this approach, we considered the approximation error in light of the sparsity and the number of wires required by our model and the fully connected model in [1]. We found that for a specified approximation error, the total sparsity of the solution (total number of cells active amongst all columns) is either equal to or only slightly higher than those obtained from the fully connected network. Despite the fact that we pay a slight penalty in the sparsity of the resulting code, a major advantage of our connectivity-constrained approach is that it minimizes the number of long-range wires in the network. To show that the *population sketch* our model produces could in fact be used in high-level inference tasks, we demonstrate that the sketch is invariant under translations and changes in scale and illumination. Finally, we compare the quality of the reconstructions from sparse codes obtained with our method with other standard methods for sparse approximation.

## References

1. Rozell CJ, Johnson DH, Baraniuk RG, Olshausen BA: **Sparse coding via thresholding and local competition in neural circuits.** *Neural Computation* 2008, **20**:2526-2563.

Publish with **BioMed Central** and every scientist can read your work free of charge

*"BioMed Central will be the most significant development for disseminating the results of biomedical research in our lifetime."*

Sir Paul Nurse, Cancer Research UK

Your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:  
[http://www.biomedcentral.com/info/publishing\\_adv.asp](http://www.biomedcentral.com/info/publishing_adv.asp)

