On the Origins of Hierarchy in Visual Processing

Angelo Frisocinis, Victor Boutin and Laurent U Perrinet
INT, Aix Marseille Univ, CNRS, Marseille, France

Motivation

It is widely assumed that visual processing follows a forward sequence of processing steps along a hierarchy of laminar sub-populations of the neural system. Taking the example of the early visual system of mammals, most models are consequently organized in layers from the retina to visual cortical areas, until a decision is taken using the representation that is formed in the highest layer. Typically, features of higher complexity (position, orientation, size, curvature, ...) are successively extracted in distinct layers [2, 3]. This is prevalent in most deep learning algorithms and stems from a long history of feed-forward architectures [2]. One of the most successful paradigm to achieve such a representation relies on algorithms performing ultimately sparse coding and dictionary learning. As shown in previous studies [4, 6], such an algorithm converges to a set of kernels that has strong analogies with the receptive fields of simple cells located in the Primary Visual Cortex of mammals (V1). Using the Multilayer Convolutional Sparse Coding (ML-CSC) from [8] we unsupervisedly trained a simple two-layer convolutional neural network on a set of natural images with a growing number of neurons in the second layer. By doing this, we could quantitatively manipulate the complexity of the representation emerging from such learning and analyze the sub-populations by the combination of the simple-cell-like oriented kernels found in the first layer.

Within the ML-CSC, [8] gives theoretical guarantees of stability and recovery for the learning and coding problem. Given an input signal \( y \in \mathbb{R}^N \), this problem consists in finding a set of sparse maps \( \{v_i\} \) and dictionaries \( \{D_j\}_{j=1}^8 \) that fit the Lasso formulation:

\[
\min_{\{v_i\}, \{D_j\}} \frac{1}{2} \sum_i (y_i - \langle v_i, D_j \rangle)^2 + \lambda \|v_i\|_1,
\]

subject to

\[
\forall \ i, \ j, \ |v_i|, \ |D_j| \leq 1.
\]

where \( d_i^j \) is the \( i^{\text{th}} \) atom of the \( j^{\text{th}} \) dictionary.

ML-CSC algorithm

Input: training set \( \{y_i\}_{i=1}^N \); initial dictionaries \( \{D_j\}_{j=1}^8 \)

1. for \( k = 1 \) to \( K \) do
   1.1. \( D_1^{(k+1)} = D_1^{(k)} - \eta \frac{\partial \mathcal{E} }{\partial D_1^{(k)}} \)
   1.2. \( D_2^{(k+1)} = D_2^{(k)} - \eta \frac{\partial \mathcal{E} }{\partial D_2^{(k)}} \)
   1.3. \( D_3^{(k+1)} = D_3^{(k)} - \eta \frac{\partial \mathcal{E} }{\partial D_3^{(k)}} \)
   2. end

Output: \( \{D_j\}_{j=1}^8 \)

Why sparse coding?

Sparse coding is a powerful tool for compressing raw data and has been successfully applied to a wide range of applications, including image and video processing, natural language processing, and machine learning. It works by finding a set of atoms \( \{v_i\} \) and a set of coefficients \( \{a_{ij}\} \) such that \( y = \sum_i a_{ij} v_i \). The goal is to find a sparse representation, i.e., one where most of the coefficients are zero.

Quantitative analysis

Scatter plot of orientation co-occurrences: The analysis of co-occurrences was performed using two measures of angular distance between each pair of edges. It is observed that the most co-linear features are dominant in all the 4 tested architectures (C), (D), (E) and (F) corresponding to second layer dictionaries composed of respectively 32, 64, 128 and 256 atoms \( d_i^j \). The architecture (C) shows almost uniquely co-linear configurations, right angles and parallel configurations start emerging for higher second layer dimensions (D), (E) and (F).

Learning on natural images

Results for different architectures with increasing complexity. We show the result of the optimization function (1) applied to a dataset of patches from the natural images dataset [7]. For simplicity, we assume a dictionary \( D_1 \) composed of 8 convolutional kernels. We trained 4 different networks with an increasing number of atoms \( d_i^j \) in the second layer dictionary. We then observed that the fitting of the optimization function (1) applied to a dataset of patches (64 x 64 pixels) extracted from a dataset of natural with a level of sparsity \( \lambda \) = 0.1. All the networks were trained with a first layer dictionary \( D_1 \) composed of 8 convolutional kernels (atoms \( a_{ij} \) of 8 x 8 pixels \( L_1 \)). We trained different networks with an increasing number of atoms \( d_i^j \) in the second layer dictionary. (A), (B), (C) and (D) corresponding to second layer dictionaries composed of respectively 32, 64, 128 and 256 atoms \( d_i^j \). A single atom \( d_i^j \) was always composed of 9 x 9 pixels and 8 channels, with an effective dimension of 16 x 16 pixels (as in Fig. 1, (2)).

References