Fire Together – Wire Together – Come Together
or
Neuronal Tension May Co-Shape V1 Orientation Maps

Cornelius Weber & Jochen Triesch

Frankfurt Institute for Advanced Studies, Johann Wolfgang Goethe University Frankfurt. Email: c.weber@fias.uni-frankfurt.de

Introduction

The structure of orientation maps, has been shown to minimize the length of horizontal connections in V1, given certain connection patterns as a function of orientation difference. We take a V1 model network with horizontal connections.

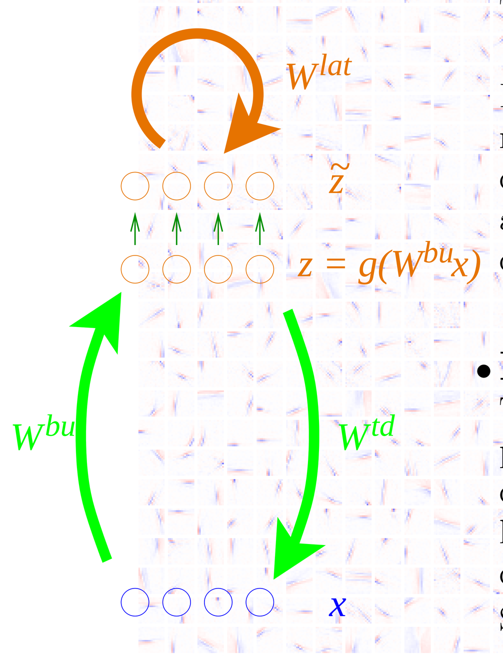
Neural Activations are maintained in this network by recurrent computations constituting an associator network.

Weight Learning has been performed for the purpose of memorizing the network's internal representation of natural image patches.

Neuronal Shifting is performed here to assess whether minimizing the lengths of the learnt connections leads to a realistic orientation map. After convergence, horizontally directed tension forces are in balance. The results with 1024 neurons and 16×16 pixel retinal input show that the neurons arrange topographically and form an orientation map similar to one hypercolumn in V1.

The image sequences below show neurons shifting to these positions.

Neural Activation



- **Top level: horizontal V1 connections**
  The attractor network with weights W^{lat} learns to memorize the input z in its continuous activations $\tilde{z}$. The learning rule is given in the next right box.

  Here we will regard their physical forces and a possible relation to topographic mappings. In the following, 2-D computational indices that define a cell's position on a grid are considered as continuous 2-D positions on a cortical sheet and will be modified by cell-shifting.

- **Lower level: visual edge detectors**
  They are models for V1 simple cells and supply pre-processed visual input to the top level. The simple cells (their recsptive fields seen here as background) were learnt by a generative model via overcomplete, sparse coding from grey-scale natural image patches.
  Symbols: x = visual input; W^{bu} = recognition weights; z = hidden code; g = logistic transfer function; W^{td} = generative weights, used only during learning.

The hidden units' activations z are determined by the data and the lower level network. If any two units tend to fire together, this will in the following influence W^{lat} via learning.

Weight Learning

- **Activation initialization**
  $\tilde{z}_i(t^0) = z_i(t^0)$
  The attractor network activations are initialized with the output of the model's simple cells.

- **Activation update**
  $\tilde{z}_i(t+1) = g(\vec{w}_i^{lat} \cdot \tilde{z}(t))$
  Recurrent relaxation for a few iterations ...

- **Lateral weight learning**
  $\Delta w_{ij}^{lat} \approx (z_i(t^{end}) - \tilde{z}_i(t^{end})) \cdot \tilde{z}_j(t^{end-1})$

  Learning uses the difference between the bottom-up input and the attractor network activations. The attractor network tries to remember the bottom-up input as good as possible. (If during relaxation time the bottom-up input changes slightly, then invariances can be built into the attractor network.)
  The background behind these lines shows the learnt lateral weights.

The learnt horizontal attractor network weights are in the following interpreted as physical connections which exert a force between any two mutually connected units.

Neuronal Shifting

The neuron's indices which define them on a computational grid are now interpreted as 2-D positions on a cortical sheet. The neuron's positions are unsorted so far. Pulled by the lateral weights, we can shift the neurons' positions until all forces are in balance.

- **Position shifts**

$$\Delta\vec{x}_i \approx \sum_j \Big(\underbrace{(|w_{ij}| + |w_{ji}|)}_{\text{attraction}} - \underbrace{\frac{\eta}{\|\vec{x}_i - \vec{x}_j\|}}_{\text{repulsion}} \Big) \cdot \frac{\vec{x}_i - \vec{x}_j}{\|\vec{x}_i - \vec{x}_j\|}$$

  – an attractive force which pulls neurons together is proportional to the absolute values of the weights between neurons
  – a repulsive force which prevents the neurons to collapse into one point is inversely proportional to the distance between any two neurons (scale parameter η)

- **Cost function**

$$E = \frac{1}{2}\Big(\sum_{i,j} (|w_{ij}| + |w_{ji}|)\, d_{ij} - \eta \sum_{i,j} \ln d_{ij} \Big)$$

$$\text{where} \quad d_{ij} = \|\vec{x}_i - \vec{x}_j\| = \sqrt{\textstyle\sum_r (x_{ir} - x_{jr})^2}$$

Hence we have $\Delta\vec{x}_i = -\nabla_{\vec{x}_i} E$. In shifting the neurons' positions, we minimize this cost function by gradient descent.

Discussion

- In order to minimize horizontal weight length, similar neurons shift together. Neurons are similar w.r.t. receptive field position and orientation tuning.
- The number of neurons, and the receptive field size correspond to a structure no larger than a "hyper-column" in V1. Therefore we are not able to see larger structures such as patterned orientation maps.
- Whether the resulting map structure is bioloigally realistic depends on *(i)* the lateral weights and *(ii)* the forces. Since the forces are straightforward to implement, rather the lateral weights are critical, and how they are learned.
- Therefore, this method can be used to check whether lateral weights, and their learning rules, may be realistic.
- Neurons are not known to do far horizontal movements though. However, we assume that minute movements would be possible in order to balance forces, and that in the biological system, all forces must be in balance.

Acknowledgments

We acknowledge financial support by the European Union through project MEXT-CT-2006-042484 ("PLICON") and by the Hertie Foundation.

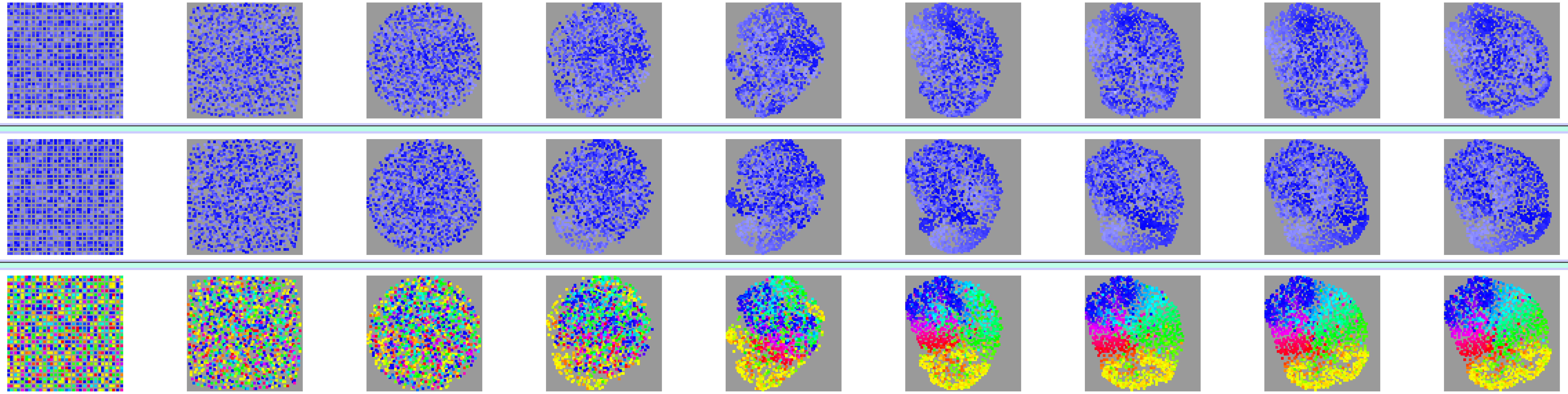
The blue color of each neuron denotes the
X-position
of the receptive field

left right

The blue color of each neuron denotes the
Y-position
of the receptive field

top bottom

The color of each neuron denotes its
orientation tuning



iteration 1 10 30 60 100 240 700 2400 4900