

Visual processing

- Photo receptors. Mainly:
 - Rods (> 10⁸)
 - Cones ($\approx 6 \cdot 10^6$, most in fovea)
- $\blacktriangleright \rightarrow$ Retinal ganglion cells
 - Multiple different types ($\approx 1.5 \cdot 10^6$ in total)
- $\blacktriangleright \rightarrow Optic nerve$
- \blacktriangleright \rightarrow Lateral geniculate nucleus (LGN)
- $\blacktriangleright \rightarrow Visual \ Cortex$
 - \blacktriangleright Ventral stream (V1 \rightarrow V2 \rightarrow V4)
 - \blacktriangleright Dorsal stream (V1 \rightarrow V2 \rightarrow MT)

Roland Memisevic

・ロト・(回ト・ミト・ミト)・ロト・(回ト・ミト)

Spikes and firing rate

This makes it possible to abstract away the spikes and think of real numbers being sent along wires.

Machine learning for vision

- Because of phenomena like *refractory periods* firing rates saturate at a maximum value.
- Some neurons have non-zero minimum firing rate called *spontaneous firing rate*.
- The input collected from other neurons depends on synaptic efficacies, which can be modeled as a weighted sum.



- Neurons communicate information via series of action potentials travelling along the axon.
- The *firing rate* depends on the total input collected from other neurons via its dendrites.
- Action potentials and firing rates are complex chemical processes.
- They can be modeled with differential equations whose solutions yield highly idealized models of the depency of firing rate on total input.

Roland Memisevic Machine learning for vision

Idealized input-output behaviour



 Together this yields the common idealized input-output model



 Other sigmoidal functions can be used.

・ロト ・回 ・ ・ ヨト ・ ヨ ・ つへの

 (Or completely other non-linearities...)

・ロト・西ト・ヨト・ヨー うくの

Idealized input-output behaviour



- The use of an element-wise nonlinearity which is applied to a linear combination of inputs is sometimes referred to as *linear/nonlinear model*.
- It is the building block of most practical neural network models.
- And it is key to building multi-layer systems.

Machine learning for vision

Hebbian learning

- The fact that synaptic efficacies can change is the basis for *learning*.
- The most famous learning rule is *Hebbian learning*, introduced by Donalb Hebb in 1949:

"When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

• "Fire together, wire together".

Hebbian learning

 Hebbian learning is *local* so it doesn't need information to travel far to perform weight updates.

Roland Memisevic

- Interestingly, Hebbian-type local learning rules (with some simple twists) can implement most of the modules needed in visual recognition, such as:
- Linear regression
- Logistic regression
- PCA
- Sparse coding, feature learning

Roland Memisevic

Roland Memisevic Machine learning for vision

▲■▶ ▲ 臣▶ ▲ 臣▶ 三臣 - わえぐ

ロト (日) (三) (三) (三) (三) (0)

PCA via Hebbian learning

► Hebbian learning with linear neurons:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})\boldsymbol{x}$$

Average weight change:

$$\begin{split} \left\langle \Delta \boldsymbol{w} \right\rangle &= \left\langle \eta(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})\boldsymbol{x} \right\rangle \\ &= \left\langle \eta \boldsymbol{x} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w} \right\rangle \\ &= \eta \boldsymbol{C} \boldsymbol{w} \end{split}$$

where C is the covariance matrix of \boldsymbol{x} .

Weight dynamics as differential equation:

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}\boldsymbol{t}} = \eta \boldsymbol{C}\boldsymbol{w}$$

Solution:

|▲□ ▶ ▲ 臣 ▶ ▲ 臣 ▶ ○ 臣 ● りへの

 $\boldsymbol{w}(t) = \boldsymbol{e}^{\eta C t} \boldsymbol{w}(0)$

PCA via Hebbian learning

- Consider discrete time for simplicity (but this works analogously for continuous time):
- Write w(t) as a linear combination of eigenvectors of C:

$$\boldsymbol{w}(t) = \sum_{k=1}^{D} a_k(t) \boldsymbol{v}_k$$

• Repeated multiplication of w(0) by C yields

Roland Memisevic

$$\boldsymbol{w}(t) = \boldsymbol{C} \cdots \boldsymbol{C} \boldsymbol{w}(0) = \boldsymbol{C}^{t} \boldsymbol{w}(0)$$
$$= \boldsymbol{C}^{t} \sum_{k} a_{k}(0) \boldsymbol{v}_{k}$$
$$= \sum_{k} a_{k}(0) \boldsymbol{C}^{t} \boldsymbol{v}_{k}$$
$$= \sum_{k} a_{k}(0) \lambda_{k}^{t} \boldsymbol{v}_{k}$$

Machine learning for vision

PCA via Hebbian learning

- "But ||w|| will grow."
 Solution: Renormalize after every step (eg. Oja's rule).
- "What about the other eigenvectors v₂,..., v_D?"
 Solution: Orthogonalize after each step (eg. Sanger's rule).

PCA via Hebbian learning

- Assume $\lambda_1 > \ldots > \lambda_D$
- We get

$$\boldsymbol{w}(t) = \sum_{k=1}^{D} \boldsymbol{a}_{k}(0) \lambda_{k}^{t} \boldsymbol{v}_{k}$$
$$= \lambda_{1}^{t} \left[\boldsymbol{a}_{1}(0) \boldsymbol{v}_{1} + \sum_{k=2}^{D} \boldsymbol{a}_{k}(0) \left(\frac{\lambda_{k}}{\lambda_{1}} \right)^{t} \boldsymbol{v}_{k} \right]$$
$$\rightarrow \boldsymbol{a}_{1}(0) \lambda_{1}^{t} \boldsymbol{v}_{1} \quad \text{for large } t$$

 $\lim_{t\to\infty}\frac{\boldsymbol{w}(t)}{\|\boldsymbol{w}(t)\|}=\boldsymbol{v}_1$

So

Receptive fields

The firing rate of a neuron can be highly selective wrt. the input.

Roland Memisevic

The set of stimuli parameters that make a neuron fire is called receptive field.



・ロケ・雪ケ・叫ケ・山 ひゃつ

Receptive fields



 a: Typical receptive fields of retinal ganglion cells and of cells in LGN.

Machine learning for vision

 b: Typical receptive fields of cells in V1. These are also known as simple cells.

Roland Memisevic

Receptive fields



Receptive fields



Retinotopy

The receptive fields of neighboring neurons in LGN and V1 tend to cover neighboring areas of the visual field.

Machine learning for vision

> This is known as retinotopic organization.

Roland Memisevic

- It means that their joint responses yield the "visual analog of an STFT".
- A convolution layer could be used to simulate this (But this is not the only organizing principle that can be found in visual cortex.)

◆□ > ◆□ > ◆三 > ◆三 > ・三 のへの



Roland Memisevic

the visual analog of an

filters, (d,e) response maps, (e) amplitude map (more on this later).

Gabor filter response



To compute local Fourier amplitudes (the spectrogram, (f)) it is necessary to sum over the squares of two filter responses.

Machine learning for vision

In practice, this requires convolving the image with a *non-linear* function.

◆□> ◆□> ◆目> ◆目> ◆目> ● ○○

Introspective evidence for frequency channels

Roland Memisevic



Complex cells



Machine learning for vision

⇒ + ≡ + ≡ + 900

- There do seem to be cells in V1 whose responses are phase independent. They are called **complex** cells (also proposed by Hubel and Wiesel)
- A simple mathematical model for these (called "energy mechanism") is to sum over squared simple-cell responses.
- A Fourier feature pair with 90 deg phase difference is known as quadrature pair. (Better than saying "sin/cos", because phasors in the Fourier expansion may have arbitrary different phase.)

Receptive fields

Machine learning for visior

Topography

- Another organizing principle, besides retinotopy, is spatial organization according to *frequency and* orientation of filters.
- This is often called topographic organization
- ► The phase of the filters, in contrast, is random.
- All attempts to model topography seem to require complex cell like mechanisms.
- Topographic organization is ubiquitous across the cortex.

Machine learning for vision

Roland Memisevic

・ロ・・聞・・聞・・聞・・日・

Topography (synthetic)

Roland Memisevic



from: Natural Image Statistics (Hyvarinen, Hurri, Hoyer; 2009)

▲□▶ ▲□▶ ▲臣▶ ▲臣▶ 三臣 - のへで

▲ 臣 ▶ ▲ 臣 ▶ 臣 ● � � �

Topography (synthetic)



(a) x-position, (b) y-position,(c) orientation, (d) frequency,(e) phase

from: Natural Image Statistics (Hyvarinen, Hurri, Hoyer; 2009)

