Unsupervised learning
*Apprentissage non supervisé*

Jean-Pierre Nadal

Laboratoire de Physique Statistique
Ecole Normale Supérieure

and

Centre d’Analyse et Mathématique Sociales
Ecole des Hautes Etudes en Sciences Sociales

nadal@lps.ens.fr
Unsupervised Hebbian learning and neural coding:
the Oja model

*a reasonably non technical introduction*

Biblio:

Model:
- \( I_j(t), j = 1, \ldots, N \) input/stimulus at time \( t \) (e.g. intensities of \( N \) photoreceptors)
- single cell, activity \( V \)
  - simplest model: linear computation
  \[
  V(t) = \sum_j W_j I_j(t)
  \]
- \( W_j \) ~ synaptic weights

Usefull remark: output = scalar product of two vectors,
- \( W = \text{vector of components } W_j, j=1,\ldots,N \)
- \( I = \text{vector of components } I_j, j=1,\ldots,N \)

\[
V(t) = W \cdot I(t)
\]
Unsupervised Hebbian learning

Environment

• The typical stimuli 'seen' by the cell are characterized by some probability distribution $P$ (e.g. $P$ is given by the statistics of 'natural images')

(more exactly, the environment is defined by a measure on the N-dimensional input space)

Auto-organization / unsupervised learning

• at each time $t$, a stimulus $\mathbf{I}(t) = \{I_j(t), j = 1, \ldots, N\}$ is drawn from the distribution $P(\mathbf{I})$

• this produce an output $V(t) = \sum_j W_j I_j(t)$

• the weights are updated with an Hebbian rule: for each $j$, $W_j \rightarrow W_j + \Delta W_j$

  $\Delta W_j = \sim \text{product input } I_j(t) \text{ by output } V(t)$
Unsupervised Hebbian learning

‘pure’ Hebbian adaptation would be

$$\Delta W_j = \varepsilon \ V(t) \ I_j(t) \quad (\varepsilon = \text{learning rate})$$

actually the interesting rule to consider is the covariance rule:

$$\Delta W_j = \varepsilon \ [V(t) - < V >] \ [I_j(t) - < I_j >]$$

where $< I_j >$ denotes the average over the input distribution $P$
(hence $< V > = \sum_j W_j < I_j >$)

To simplify the discussion, let’s first subtract the mean to the input,
or, equivalently, assume $< I_j > = 0$ for every $j$
- and then we can consider the pure Hebbian rule.
Unsupervised Hebbian learning

Thus at each time $t$ we have
\[ \Delta W_j = \varepsilon \, V(t) \, I_j(t) \]
that is
\[ W(t+1) = W(t) + \varepsilon \, V(t) \, I(t) \]
where
\[ V(t) = W(t) \cdot I(t) \]

In geometrical terms (illustration for $N=2$):

- Case $V(t) > 0$
Unsupervised Hebbian learning

Thus at each time $t$ we have $\Delta W_j = \varepsilon V(t) I_j(t)$

that is $W(t+1) = W(t) + \varepsilon V(t) I(t)$

where $V(t) = W(t) \cdot I(t)$

In geometrical terms (illustration for $N=2$):
Unsupervised Hebbian learning

What to expect on the long run depends on the input distribution (the environment)
P: density of stimuli higher where darker green → higher probability to have an input from a darker domain

$W(t)$ will tend to align with the direction of greatest spread
Unsupervised Hebbian learning

From these simple drawings one would expect that:
• the direction of the weight vector $W$ will tend to align with the direction of largest spread of the stimuli
• the length of the vector $W$ will increase infinitely

Both statements are correct. Let’s first cure the growth problem, and then study the interesting result.

Modification of the learning rule:

$$\Delta W_j = \varepsilon \{ V(t) I_j(t) - V(t)^2 W_j(t) \}$$

This is Oja’s learning rule (1982). One can also write it as:

$$W(t+1) = W(t) + \varepsilon \{ V(t) I(t) - V(t)^2 W(t) \}$$

$$= (1 - \varepsilon V(t)^2) W(t) + \varepsilon V(t) I(t)$$

(hence renormalization of the scale of $W$ + Hebbian modification).

As before, $V(t) = W(t) \cdot I(t)$

(and recall that we assume $<I_j> = 0$ for every $j$).
Unsupervised Hebbian learning

With this Oja’s learning rule, one can show that, at large times, \( \mathbf{W} \) converges to a unit vector,

\[
\sum_j W_j^2 = 1,
\]

with its direction aligned with the direction of largest spread (see the Appendix for some mathematical details).

Remark: one can easily check this normalization, assuming convergence.

From the definition of the learning rule, compute \( \sum_j W_j \Delta W_j \).

One gets

\[
\sum_j W_j \Delta W_j = \varepsilon V^2 (1 - \sum_j W_j^2 )
\]

Convergence means (in average) \( \Delta W_j = 0 \), hence \( 1 - \sum_j W_j^2 = 0 \).

Of course the term added to the pure Hebbian learning term has been precisely chosen in order to impose such constraint - that is to impose (at least in the long run) \( \sum_j W_j^2 = \) given constant.
Unsupervised Hebbian learning
Principal Component Analysis

Assume that $W$ is indeed a unit vector with direction aligned with the direction of largest spread of the input distribution $P$. This direction is called the first principal component of $P$.

Why is this a good choice for this neuron? or, otherwise stated, What is this choice of weights good for?
What is the output $V$ for a given input $I$? $V = W$. $W$ being of unit length, this means that $V$ is the coordinate of the input on the axis $(O, W)$ - the projection of $I$ onto this axis.
Principal Component Analysis

all the inputs on this line have the same projection $V$ as the input $I$ - they are all seen by the neuron as a same stimulus.
Principal Component Analysis

all the inputs on this line have the same projection $V$ as the input $I$ - they are all seen as a same stimulus by the neuron.

taking any other axis would be worst
Principal Component Analysis

having at our disposal a single cell to ‘see’ the stimuli, the ability to discriminate between stimuli is maximized by taking $V$ as the projection onto the first principal component.
**Principal Component Analysis (PCA)**

Beyond the 1rst Principal Component:
in dimension $N$, there are $N$ Principal Components (PCs)

(project the inputs in the space orthogonal to the 1rst PC. Compute the 1rst PC in
that space: this is the 2nd PC of the original input space. Iterate... See also the
Appendix).

One can have a network with $K$ (linear) output units
computing the $K$ first PCs

$\rightarrow$ generalizations of Oja’s learning rule
which require **competition** and/or
inhibitory lateral connections between units
(otherwise each unit would compute the 1rst PC)

Refs:
Oja 1983, 1992;
Sanger, 1989; Földiák, 1989
The Oja model can be considered as the simplest version of a family of models aiming at describing information processing by the retina, and more generally the first stages of visual (or other sensory) processing. The unsupervised learning rules allow to discuss the epigenetic development of the visual system – how it structures itself just after birth while being under a flow of visual stimuli – and even before birth, while under a flow of spontaneous activity.

In these analysis one is lead to consider:

• the role of the normalization (and other) constraints (e.g. the development of occular dominance can de shown to result from a similar Hebbian rule but under a constraint $\Sigma_j W_j = 1$ instead of $\Sigma_j W_j^2 = 1$).

• the noise which is present at every stage of processing

• more realistic input statistics

In most cases this requires to go beyond coding solely based on the PCA, and other notions related to optimal coding have to be introduced - information maximization (« infomax »), redundancy reduction, Independent Component Analysis (ICA).
We consider Oja's learning rule (1982)

$$\Delta W_j = \varepsilon \{ V(t) I_j(t) - V(t)^2 W_j(t) \}$$

that is:

$$W(t+1) = W(t) + \varepsilon \{ V(t) I(t) - V(t)^2 W(t) \}$$

where $V(t) = W(t) \cdot I(t)$

and $I(t)$ is drawn from some distribution $P$

(and we assume $\langle I_j \rangle = 0$ for every $j$).

What follows: some notes on

• the convergence of this learning dynamics
• the link with PCA
• what is PCA
Convergence

The rigorous proof of convergence is not that easy, because it is not true that at each time $t$ the direction of the weight vector becomes closer to the direction of the first PC: there are (rarer and rarer) events where the randomly chosen stimulus induces the opposite effect. Actually the convergence has been proved under an additional assumption, that the learning rate $\varepsilon$ is taken time dependent, decreasing with time: $\varepsilon \sim 1/t$.

The easy way to understand what’s going on: one makes an approximation, replacing the true dynamics by its average over the stimulus distribution. Let us do that now.

*Recall that we denote by $< . >$ the average over the stimulus ensemble.*
Appendix: some mathematical details

The average learning dynamics is
\[ W_j(t+1) = W_j(t) + \varepsilon \{ <V I_j> - <V^2> W_j(t) \} \]

One has \( V = \sum_k W_k I_k \), hence

\[ <I_j V> = <I_j \sum_k W_k(t) I_k> = \sum_k C_{jk} W_k(t) \]

where \( C_{jk} = <I_j I_k> \).\( C \) = covariance matrix of the input distribution \( (C_{jk} = <I_j I_k> - <I_j> <I_k> , \text{but here we have assumed} <I_k> = 0) \).

Assume convergence. Then one has for every \( j \), \( <V I_j> - <V^2> W_j = 0 \), that is

\[ \sum_k C_{jk} W_k = \lambda W_j \]

with \( \lambda = <V^2> \).

In matrix form,

\[ C W = \lambda W \]

That is, \( W \) is an eigenvector of the covariance matrix \( C \), for the eigenvalue \( \lambda \).
The covariance matrix is real symmetric ($C_{jk} = C_{kj}$), it can be diagonalized in an orthogonal basis: this is precisely doing the Principal Component Analysis.

There are $N$ orthogonal eigenvectors $W^a$ of unit length, $a = 1, \ldots, N$, each one solution of

$$ C W^a = \lambda_a W^a $$

with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$. The eigenvalues have a geometrical meaning: $\lambda_a$ is the variance of the projections onto $W^a$, the $a$th PC.

Let $V_a = W^a \cdot I$, then

$$ \langle (V_a)^2 \rangle = \lambda_a $$

Indeed,

$$ \langle (V_a)^2 \rangle = \langle \sum_j W^a_j I_j \sum_k W^a_k I_k \rangle = W^a^T C W^a = \lambda_a W^a W^a^T $$

and $W^a W^a^T = \sum_j (W^a_j)^2 = 1$.

The first principal component is the one associated to the largest eigenvalue, that is with the largest variance - the direction of largest spread.
Principal Component Analysis (PCA) is a standard tool in data analysis and signal processing. It allows to project data, initially living in a high dimensional space, in a space of small dimension (the space of the first principal components) where most of the data really lives, and in such a way that discriminability between data points is maximized.

Back to our learning dynamics:
If $\lambda_1 > \lambda_2 \geq ... \geq \lambda_N$, one can check that $W(t)$, under the average learning dynamics, converges to the eigenvector associated to the largest eigenvalue, the unique stable fixed point of this dynamics (the other PCs, that is the other eigenvectors of the correlation matrix, are also fixed points, but they are unstable fixed points).