ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Unsupervised and reinforcement learning in neural networks
Week 2

Lecturer: Marc-Oliver Gewaltig

Solutions: PCA & Oja's rule

Exercise 1

By looking at the formulation of the following learning rules, classify them to Hebbian-type or non-Hebbian rules.

Variable definitions. x: stimulus (neuronal input), y: neuronal activity, w: synaptic weight, ϵ: small positive number (learning rate), r : reward. Unless otherwise stated, x,y are mean firing rates.

Hebbian-type rules are local rules that exploit presynaptic-postsynaptic correlations, i. e. a term $+\epsilon xy$ is present in the formulation. According the lecture notation $x = \nu_j^{pre}$, $y = \nu_i^{post}$ and the Hebbian component is more generally defined as $\alpha_2^{corr}(w)\,\nu_i^{post}\nu_j^{pre}$, with $\alpha_2^{corr}(w) \geq 0$.

1. Rescorla-Wanger (reward predicting) rule. $w \to w + \epsilon\delta x$, where $\delta = r - y$, $x \in \{0, 1\}$. It is an anti-Hebbian rule, because of the term $-\epsilon xy$.

2. Homeostatic rule. $w \to w - \epsilon\,(y - \theta)$, where θ is the homeostatic threshold. Non-Hebbian, no dependence on the presynaptic activity x.

3. Node perturbation. $w \to w + \epsilon ry\xi$, where ξ represents noise (random variable drawn from a Gaussian distribution). Non-Hebbian, no dependence on the presynaptic activity x.

4. Sejnowski rule. $w \to w + \epsilon\,(y - \langle y\rangle)\,(x - \langle x\rangle)$, with $\langle .\rangle$ being the mean activity of the signals y and x across time. Hebbian, subtracting the mean simply centers the signals around 0.

5. Oja rule. $w \to w + \epsilon y\,(x - yw) = w + \epsilon\left(xy - wy^2\right)$. Hebbian with an additional term to achieve stability. The second term comes from the Taylor expansion, if we define $\alpha_2^{post}(w) = -w$.

6. Learning in Hopfield networks. $w \to w + \epsilon yx$, with $x, y \in \{-1, 1\}$. Hebbian.

7. Delta rule. $w \to w + \epsilon\delta x$, where $\delta = t - y$, with t being the desired output for input x. Similar to rule 1 but under a different framework. This one is supervised. Anti-Hebbian, because of the term $-\epsilon xy$.

8. Associative Reward Inaction $w \to w + \epsilon r\,(y - P(y))\,x$, where $P(y)$ the probability of y to be active, $y \in \{0, 1\}$. Hebbian-type for reward defined $r \geq 0$, $P(y)$ is a quantity that can be known to the synapse.

Reward is a global signal that can be locally known to the synapse, and modulate the synaptic changes. If reward is strictly defined as $r \in [0, 1]$ then for rules 3 and 8 it induces synaptic modifications in rewarded cases only. If reward is also negative, e.g. in the case of a punishment (an electric shock) then for rule 8 it induces anti-Hebbian learning (Hebbian with a negative sign).

Exercise 2

Use principal component analysis to reduce the dimensionality of the dataset shown in Fig. 1.

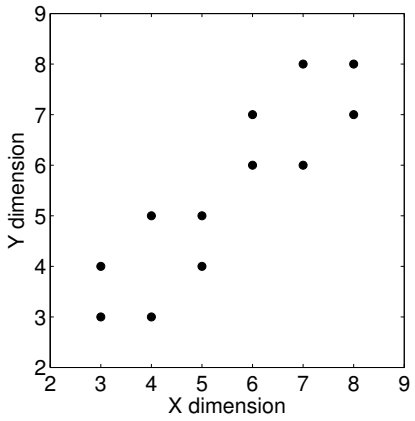

Figure 1: Original Dataset

X	3	3	4	4	5	5	6	6	7	7	8	8
Y	3	4	3	5	4	5	6	7	6	8	7	8

1. *Center the data by subtracting their mean.*

   Calculation of the mean:

$$\overline{X} = \sum_{i=1}^{p} \frac{X_i}{p} = \frac{6 \cdot 11}{12} = \frac{11}{2} \tag{1}$$

   Similarly, $\overline{Y} = \frac{11}{2}$.

$\tilde{X}$	-2.5	-2.5	-1.5	-1.5	-0.5	-0.5	0.5	0.5	1.5	1.5	2.5	2.5
$\tilde{Y}$	-2.5	-1.5	-2.5	-0.5	-1.5	-0.5	0.5	1.5	0.5	2.5	1.5	2.5

2. *Calculate the covariance matrix of the data.*

   The covariance matrix takes the form:

$$C := \frac{1}{p} \sum_{\mu=1}^{p} (\mathbf{x}^{\mu} - \overline{\mathbf{x}})(\mathbf{x}^{\mu} - \overline{\mathbf{x}})^T, \tag{2}$$

where $\mathbf{x} = \begin{pmatrix} X \\ Y \end{pmatrix}$ and $\overline{\mathbf{x}} = \begin{pmatrix} \overline{X} \\ \overline{Y} \end{pmatrix}$.

Thus

$$C = \frac{1}{12} \begin{pmatrix} 35 & 31 \\ 31 & 35 \end{pmatrix}. \tag{3}$$

3. *Find the eigenvalues & eigenvectors of the covariance matrix end explain their meaning in the context of PCA.*

   The characteristic equation of the covariance matrix C is

$$det(C - \lambda I) = det \begin{pmatrix} 35/12 - \lambda & 31/12 \\ 31/12 & 35/12 - \lambda \end{pmatrix} = 0. \tag{4}$$

   Solving the equation, we get two solutions: $\lambda_1 = 11/2$ and $\lambda_2 = 1/3$. The larger eigenvalue corresponds to the most important eigenvector.

   Further, we solve the equations $CV_1 = \lambda_1 V_1$ and $CV_2 = \lambda_2 V_2$ and we find the two (normalized) eigenvectors:

$$V_1 = \begin{pmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix} \tag{5}$$

   and

$$V_2 = \begin{pmatrix} -\sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix}. \tag{6}$$

   V_1 corresponds to direction 45^o and V_2 to 135^o. These are the principle components; the new axes for describing the data sets.

4. *Calculate the output data of PCA and discard the less significant component. What are the principal axes in the original coordinate system? Could you obtain the new dataset without making any calculations?*

   The feature matrix is composed of the eigenvectors (column-wise) in the order of larger to smaller corresponding eigenvalue:

$$F = \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix} \tag{7}$$

   The new data are obtained by multiplying the transposed feature matrix (i.e. with the most significant eigenvector on top) by a matrix D_c whose columns are the mean-centered data $\mathbf{x} - \overline{\mathbf{x}}$:

$$D_n = F^T D_c. \tag{8}$$

   This will give us the data shown in Fig. 2.

   In practice we want to reduce the dimensions of the dataset to the eigenvectors with the larger eigenvalues, in our case V_1. The feature vector becomes:

$$F^T = \begin{pmatrix} \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix} \tag{9}$$

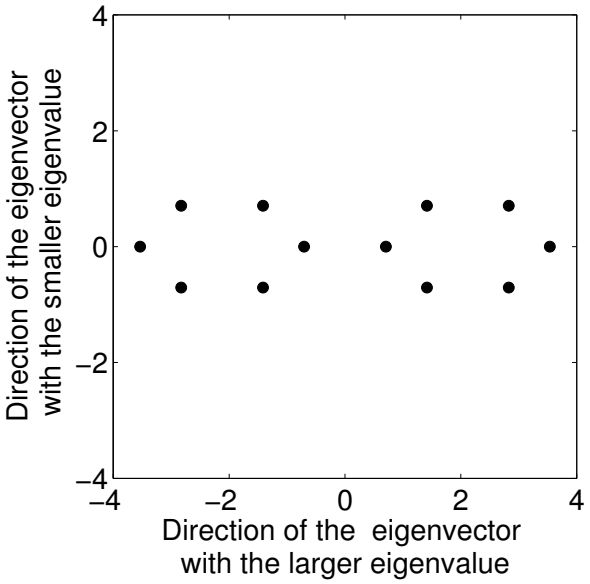

Figure 2: Dataset plotted on the new axes formed by V_1 and V_2.

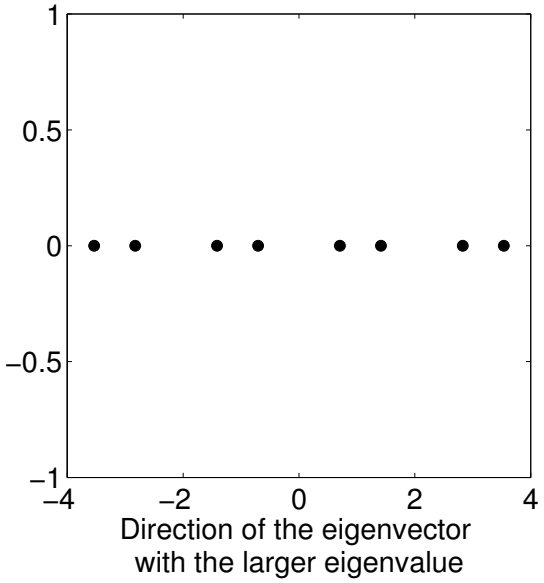

Figure 3: Reduced dataset plotted on the main principal component V_1.

and the dataset is shown in Fig. 3.

We could have easily foreseen how the reduced dataset would look like. Simply by looking Fig.1, we can see that the two axes are at 45 ° and 135 °. The projection of the dataset on the axis 45 ° gives us Fig. 3.

5. *Can you obtain back the original data? How?* Assuming that I have used the whole feature matrix for calculating the new dataset, we simply need to invert the transformation (8):
$$D_c = F^{-T} D_n = F D_n, \tag{10}$$
and then add to the data the means we originally calculated.

For additional information you may read the simple PCA tutorial by Lindsay I Smith.

Exercise 3

3.1 *Let us consider a neuron that receives an N-dimensional input. Its weight dynamics is given by:*
$$\frac{d\vec{w}}{dt} = C\vec{w} \tag{11}$$
with
$$C = \begin{pmatrix} 1 & 0.5 & 0 & 0 & \dots & 0 & 0.5 \\ 0.5 & 1 & 0.5 & 0 & \dots & 0 & 0 \\ 0 & 0.5 & 1 & 0.5 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.5 & 0 & 0 & 0 & \dots & 0.5 & 1 \end{pmatrix}. \tag{12}$$
Show that the complex vectors $w_k = \exp\left(\frac{2\pi i k}{N} m\right)$*, with* $k = 1 \dots N$ *and* $m \in \mathbb{Z}$ *are eigenvectors of C. Assume cyclic boundary conditions.*

We need to show that $Cw = \lambda w$, where λ is the eigenvalue. Considering the kth element, we have
$$(Cw)_k = \lambda \exp\left(\frac{2\pi i k}{N} m\right) \tag{13}$$
For elements with $k = 2, \dots, N-1$, the term $(Cw)_k$ is
$$(Cw)_k = 0.5 \exp\left(\frac{2\pi i (k-1)}{N} m\right) + \exp\left(\frac{2\pi i k}{N} m\right) + 0.5 \exp\left(\frac{2\pi i (k+1)}{N} m\right). \tag{14}$$
Since $\exp(2\pi i n) = \exp(2\pi i (n+1))$, this is actually also true for $k = 1, N$. Putting (14) into (13), we get
$$\begin{aligned} 0 &= 0.5 \exp\left(\frac{2\pi i (k-1)}{N} m\right) + (1-\lambda) \exp\left(\frac{2\pi i k}{N} m\right) + 0.5 \exp\left(\frac{2\pi i (k+1)}{N} m\right) && (15) \\ &= \exp\left(\frac{2\pi i k}{N} m\right) \left(0.5 \exp\left(\frac{-2\pi i}{N} m\right) + (1-\lambda) + 0.5 \exp\left(\frac{2\pi i}{N} m\right)\right) && (16) \\ &= \exp\left(\frac{2\pi i k}{N} m\right) \left(\cos\left(\frac{2\pi m}{N}\right) + 1 - \lambda\right) && (17) \end{aligned}$$

Here we have used the fact that $e^{ix} = \cos(x) + i\sin(x)$ (and thus $e^{ix} + e^{-ix} = 2\cos(x)$). The equation above only holds if

$$\lambda = 1 + \cos\left(\frac{2\pi m}{N}\right) \tag{18}$$

This defines N eigenvalues.

3.2 *Assume that the neuron receives N input patterns* $\vec{\xi}^{\mu} = (\xi_1^{\mu}, \xi_2^{\mu}, \ldots, \xi_N^{\mu})^T$ *with* $\xi_k^{\mu} = \sqrt{\frac{N}{2}}\left(\delta_k^{\mu} + \delta_k^{(\mu \bmod N)+1}\right)$. *Here,* δ_k^{μ} *denotes the Kronecker symbol, which is 1 if* $\mu = k$ *and 0 otherwise. Show that the matrix C is produced by:*

$$C_{kj} = \left\langle \xi_k^{\mu} \xi_j^{\mu} \right\rangle = \frac{1}{N} \sum_{\mu=1}^{N} \xi_k^{\mu} \xi_j^{\mu}. \tag{19}$$

Let's calulate the element C_{kj} of the matrix

$$C_{kj} = \frac{2}{N} \sum_{\mu=1}^{N} \xi_k^{\mu} \xi_j^{\mu} \tag{20}$$

$$= \frac{1}{N} \sum_{\mu=1}^{N} \sqrt{\frac{N}{2}}\left(\delta_k^{\mu} + \delta_k^{(\mu \bmod N)+1}\right) \sqrt{\frac{N}{2}}\left(\delta_j^{\mu} + \delta_j^{(\mu \bmod N)+1}\right) \tag{21}$$

$$= \frac{1}{2} \sum_{\mu=1}^{N} \delta_k^{\mu}\delta_j^{\mu} + \delta_k^{\mu}\delta_j^{(\mu \bmod N)+1} + \delta_k^{(\mu \bmod N)+1}\delta_j^{\mu} + \delta_j^{\mu}\delta_k^{\mu} \tag{22}$$

$$= \frac{1}{2}\left(\underbrace{\sum_{\mu=1}^{N} \delta_k^{\mu}\delta_j^{\mu}}_{\delta_j^k} + \underbrace{\sum_{\mu=1}^{N} \delta_k^{\mu}\delta_j^{(\mu \bmod N)+1}}_{\delta^j_{(k \bmod N)+1}} + \underbrace{\sum_{\mu=1}^{N} \delta_k^{(\mu \bmod N)+1}\delta_j^{\mu}}_{\delta^k_{(j \bmod N)+1}} + \underbrace{\sum_{\mu=1}^{N} \delta_j^{\mu}\delta_k^{\mu}}_{\delta_k^j} \right) \tag{23}$$

$$= \delta_j^k + 0.5\,\delta^k_{(j \bmod N)+1} + 0.5\,\delta^j_{(k \bmod N)+1} \tag{24}$$

This is indeed the the C matrix. The δ_j^k is the diagonal and the $\delta^k_{(j \bmod N)+1}$ is the lower off-diagonal term: they are 1 only if $j = k-1$ or $j = N$ and $k = 0$. The $\delta^j_{(k \bmod N)+1}$ element is the upper off-diagonal term.

Comment on how the weights will evolve given the nature of the input patterns.

Since all eigenvalues of C are positive (from (18)), the weight vector w will grow exponentially. The only exception is if N is even and $m = N/2$, in which case $\lambda = 0$, which means we have a fixed point of the rule. This corresponds to the weight vector with components

$$w_k = \exp(\pi i k) = cos(k\pi) + isin(k\pi) = \pm 1. \tag{25}$$

Exercise 4

4.1 *Show that the fixed points of this equation are eigenvectors of the C matrix.*

By definition, the fixed points if the equation are the vectors solutions of

$$\frac{d}{dt}w = 0 = Cw - (w^T Cw)w. \tag{26}$$

Noticing that $(w^T Cw)$ is a scalar and defining $\lambda(w) := (w^T Cw)$, this becomes

$$Cw = \lambda(w)w. \tag{27}$$

This is an eigenvalue equation, with an eigenvalue dependent on w. Thus solutions of the differential equation are also eigenvectors of C.

4.2 *Show that the eigenvector e_k associated with the largest eigenvalue of C is a stable fixed point.*
Hint: *Assume that the weight is almost the eigenvector e_k, but slightly perturbed in the direction of a different eigenvector e_j: $w(t) = \alpha(t)e_k + \epsilon(t)e_j$, with $\epsilon \ll 1$ and $\epsilon^2 + \alpha^2 = 1$.*

Let's rewrite Oja's rule with our ansatz. The left hand side becomes

$$\frac{d}{dt}w = \left(\frac{d}{dt}\alpha\right)e_k + \left(\frac{d}{dt}\epsilon\right)e_j, \tag{28}$$

and the two right hand side terms become

$$Cw = C(\alpha e_k + \epsilon e_j) = \alpha C e_k + \epsilon C e_j = \alpha\lambda_k e_k + \epsilon\lambda_j e_j \tag{29}$$

and

$$\begin{aligned}
(w^T Cw)w &= (\alpha e_k^T + \epsilon e_j^T)C(\alpha e_k + \epsilon e_j)w && (30)\\
&= (\alpha e_k^T + \epsilon e_j^T)(\alpha\lambda_k e_k + \epsilon\lambda_j e_j)w && (31)\\
&= (\alpha^2\lambda_k \underbrace{e_k^T e_k}_{=1} + \alpha\epsilon\lambda_j \underbrace{e_k^T e_j}_{=0} + \alpha\epsilon\lambda_k \underbrace{e_j^T e_k}_{=0} + \epsilon^2\lambda_j \underbrace{e_j^T e_j}_{=1})w && (32)\\
&= (\alpha^2\lambda_k + \epsilon^2\lambda_j)(\alpha e_k + \epsilon e_j). && (33)
\end{aligned}$$

We used the fact that C being a covariance matrix, it is symmetric so that its eigenvalues are orthogonal. We also assumed that the eigenvectors are normalized (thus $e_j^T e_k = \delta_{jk}$). Remembering the ansatz $\epsilon^2 + \alpha^2 = 1$, we further simplify

$$\begin{aligned}
(w^T Cw)w &= ((1-\epsilon^2)\lambda_k + \epsilon^2\lambda_j)(\alpha e_k + \epsilon e_j) && (34)\\
&= \alpha((1-\epsilon^2)\lambda_k + \epsilon^2\lambda_j)e_k + (\lambda_k\epsilon - (\lambda_k - \lambda_j)\epsilon^3)e_j. && (35)
\end{aligned}$$

Notice that the terms 28, 29 and 35 are all of the form $\ldots e_k + \ldots e_j$. Since these two vectors are orthogonal, we can project our rewritten Oja's rule to one of those. Projecting to e_j yields

$$\frac{d}{dt}\epsilon = \epsilon\lambda_j - (\lambda_k\epsilon - (\lambda_k - \lambda_j)\epsilon^3) = -(\lambda_k - \lambda_j)(\epsilon - \epsilon^3) \tag{36}$$

This differential equation has 3 fixed points: $\epsilon = 0$ and $\epsilon = \pm 1$. We consider what happens when w deviates slightly from the e_k eigenvector, i.e. $|\epsilon| \ll 1$. In that case, $\frac{d}{dt}\epsilon > 0$ when $\epsilon < 0$ and $\frac{d}{dt}\epsilon < 0$ when $\epsilon > 0$ (Remember that $\lambda_k > \lambda_j$). Thus the dynamics will

bring ϵ back to zero, and thus α to 1, making the e_k vector a stable fix point of Oja's rule.

Note that to complete the proof, one should also ensure that

$$\epsilon(t_0)^2 + \alpha(t_0)^2 = 1 \Rightarrow \epsilon(t)^2 + \alpha(t)^2 = 1, \forall t > t_0. \tag{37}$$

This equivalent to prove that $\frac{d}{dt}(\epsilon^2 + \alpha^2) = 0$. This straightforward by noting that

1. $\frac{d}{dt}(\epsilon^2 + \alpha^2) = 2\epsilon\frac{d\epsilon}{dt} + 2\alpha\frac{d\alpha}{dt}$,
2. we know $\frac{d\epsilon}{dt}$ from (36) and
3. we can calculate $\frac{d\alpha}{dt}$ the same way we obtained (36), but projecting on e_k.

The actual calculation is left as an exercise.