Theoretical Neuroscience II
Teacher: Prof. Jochen Braun
Assistant Teacher: Ehsan Kakaei
Name: Viviana Villafaña Barraza & Tasnuba Tazeen & Lee Po Shing
Date: 17th May, 2018

Exercise 4: Development of Ocular Dominance

Introduction

In order to understand the development of ocular dominance in visual cortex, one may to apply the concept of Hebbian plasticity which the synchronisation of pre- and post- synaptic activities determines the development of synaptic strength and thus the propagation of activity. In visual cortex, initially neurons receive projections from both eyes with random synaptic strength; however neuronal connections are progressively weaken or strengthen due to the synaptic activity, and eventually pruned to receive solely input from a dominant eye and ocular dominance is therefore formed. To model the development of ocular dominance, the simplest way is to consider a feed-forward network with two upstream inputs u_L and u_R as the inputs for two separate eyes and multiple downstream outputs $\boldsymbol{v}$ as different pyramidal neurons in primary visual cortex. The synaptic strength can then be considered as the following equation, which $\boldsymbol{C}$ denoted the covariance matrix of the inputs:

$$\tau_w \frac{d\boldsymbol{w}}{dt} = \boldsymbol{C}\,\boldsymbol{w}$$

In addition, to prevent such a system from an exponential growth, there are some methods such as weight saturation, synaptic normalization and the Oja's rule. Among them, weight saturation does not provide a good biological model, since inputs do not have an important role in this model. Similarly, synaptic normalization strengthens synapses only when other synapses are weakened. This means that constant interactions between neurons are required, which biologically implausible. The Oja's rule is a dynamic normalization in which the size of weights determines the rate of change.

In this assignment, the model described above was computed using MatLab. In particular, using the simple covariance method and Oja's rule, the effect of input statistics and initial values on the development of synaptic strength was calculated and plotted.

Method

In this exercise, a network consisting two outputs, which represents the two LGN afferents from the left and right eye respectively (u_L and u_R), and corresponding synaptic weight (w_L and w_R, w in the code) was used to model the development of ocular dominance. For simplicity, the values of u_L and u_R were either 1 or 0 with equal probability respectively (*Table. 1*).

$P(u_L \cap u_R)$	$u_R = 1$	$u_R = 0$	$P(u_L)$
$u_L = 1$	$\gamma/4$	$1/2 - \gamma/4$	$1/2$
$u_L = 0$	$1/2 - \gamma/4$	$\gamma/4$	$1/2$
$P(u_R)$	$1/2$	$1/2$	

Table. 1 Joint probability of different input combinations With two inputs u_L and u_R which were either 1 or 0 with equal probability, the joint probability of every combination were calculated as above, with γ ranged from 0 to 2.

To see the effect of input statistics, three different values of γ (gamma in the code) of 0.46, 1 and 1.72 were used in the code, to create inputs with negative, independent and positive correlation respectively.

In order to compute the above covariance matrix for each value of gamme, a function file called *covarianceLR.m* was created. The number of inputs (input_trial in the code) was set as 1000. With an input of γ, four different joint probabilities (P_11, P_10, P_01, P_00 in the code) were calculated. For the LGN afferents (u_11, u_10, u_01, u_00 in the code), matrice were generated with trials number in accord to their corresponding probabilities. Then the four matrices were combined with the sequence of input randomised (temp_u in the code). In order to compute the covariance matrix, the formula used was as below:

$$\boldsymbol{C} = \begin{pmatrix} c_S & c_D \\ c_D & c_S \end{pmatrix}, \begin{cases} c_S = \sum_{u_R=0}^{1} P(u_R){u_R}^2 - (\sum_{u_R=0}^{1} P(u_R)u_R)^2 = \sum_{u_L=0}^{1} P(u_L){u_L}^2 - (\sum_{u_L=0}^{1} P(u_L)u_L)^2 \\ c_D = \sum_{u_R=0}^{1}\sum_{u_L=0}^{1} P(u_L \cap u_R)u_L u_R - \sum_{u_R=0}^{1} P(u_R)u_R * \sum_{u_L=0}^{1} P(u_L)u_L \end{cases}$$

Using the modified Matlab function file *ShowEigen.m* originally provided, the eigenvectors and the eigenvalues which expressed in form of multiplication with the respective eigenvector were plotted.

Next, in order to see the development with weight saturation of the two LGN afferents, another function file called *plotweight.m* was created. The time (t in the code) was set from 0 to 25 with stepsize of 0.01.The dynamic equation of the input weight was modelled as below:

$$\frac{d\boldsymbol{w}}{dt} = \boldsymbol{C}\,\boldsymbol{w}, \text{i.e.} \frac{d}{dt}\begin{pmatrix} w_R \\ w_L \end{pmatrix} = \begin{pmatrix} c_S & c_D \\ c_D & c_S \end{pmatrix}\begin{pmatrix} w_R \\ w_L \end{pmatrix}$$

$$\frac{d}{dt}\begin{pmatrix} w_R \\ w_L \end{pmatrix} = \begin{pmatrix} c_S w_R + c_D w_L \\ c_D w_R + c_S w_L \end{pmatrix}$$

And hence,

$$\frac{d}{dt} w_R = c_S w_R + c_D w_L \qquad \frac{d}{dt} w_L = c_D w_R + c_S w_L$$

Using a for-loop, the development of the weights of different initial values were calculated. The initial values were randomly generated ranged from 0 to 0.5. With another for-loop inside, the development was calculated iteratively using the two equations as above and plotted on the weight-space.

Similarly, the development was done with Oja's rule applied in the function file *plotweightoja.m*. Similar structure was used as *plotweight.m* except the dynamic equation was modelled as below:

$$\frac{d\boldsymbol{w}}{dt} = \boldsymbol{C}\,\boldsymbol{w} - \frac{1}{2}(\boldsymbol{w}^T\,\boldsymbol{C}\,\boldsymbol{w})\,\boldsymbol{w}, \text{i.e.} \frac{d}{dt}\begin{pmatrix} w_R \\ w_L \end{pmatrix} = \begin{pmatrix} c_S & c_D \\ c_D & c_S \end{pmatrix}\begin{pmatrix} w_R \\ w_L \end{pmatrix} - \frac{1}{2}[(w_R \quad w_L)\begin{pmatrix} c_S & c_D \\ c_D & c_S \end{pmatrix}\begin{pmatrix} w_R \\ w_L \end{pmatrix}]\begin{pmatrix} w_R \\ w_L \end{pmatrix}$$

$$\frac{d}{dt}\begin{pmatrix} w_R \\ w_L \end{pmatrix} = \begin{pmatrix} c_S w_R + c_D w_L \\ c_D w_R + c_S w_L \end{pmatrix} - \frac{1}{2}[(w_R \quad w_L)\begin{pmatrix} c_S w_R + c_D w_L \\ c_D w_R + c_S w_L \end{pmatrix}]\begin{pmatrix} w_R \\ w_L \end{pmatrix}$$

$$\frac{d}{dt}\begin{pmatrix} w_R \\ w_L \end{pmatrix} = \begin{pmatrix} c_S w_R + c_D w_L \\ c_D w_R + c_S w_L \end{pmatrix} - \frac{1}{2}\begin{pmatrix} c_S {w_R}^2 + 2c_D w_R w_L \\ c_D {w_R}^2 + 2c_S w_R w_L \end{pmatrix}\begin{pmatrix} w_R \\ w_L \end{pmatrix}$$

And hence,

$$\frac{d}{dt} w_R = (c_S w_R + c_D w_L) - \frac{1}{2}(c_S {w_R}^2 + 2c_D w_R w_L) w_R \qquad \frac{d}{dt} w_L = (c_D w_R + c_S w_L) - \frac{1}{2}(c_S {w_R}^2 + 2c_D w_R w_L) w_R$$

Again, the development was done iteratively and plotted in the weight-space.

Results & Discussion

Covariance matrices were successfully obtained after training

For γ = 0.46,

$P(u_L \cap u_R)$	$u_R = 1$	$u_R = 0$	$P(u_L)$
$u_L = 1$	0.115	0.385	$1/2$
$u_L = 0$	0.385	0.115	$1/2$
$P(u_R)$	$1/2$	$1/2$	

$$c_S = \sum_{u_R=0}^{1} P(u_R){u_R}^2 - (\sum_{u_R=0}^{1} P(u_R)u_R)^2 = 0.5 - 0.5^2 = 0.25$$

$$c_D = \sum_{u_R=0}^{1}\sum_{u_L=0}^{1} P(u_L \cap u_R)u_L u_R - \sum_{u_R=0}^{1} P(u_R)u_R * \sum_{u_L=0}^{1} P(u_L)u_L = 0.115 - 0.5^2 = -0.135$$

$$\boldsymbol{C} = \begin{pmatrix} c_S & c_D \\ c_D & c_S \end{pmatrix} = \begin{pmatrix} 0.25 & -0.135 \\ -0.135 & 0.25 \end{pmatrix}$$

To find eigenvalues,

$$\begin{pmatrix} 0.25 & -0.135 \\ -0.135 & 0.25 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.25-\lambda & -0.135 \\ -0.135 & 0.25-\lambda \end{pmatrix}$$
$$\det \begin{pmatrix} 0.25-\lambda & -0.135 \\ -0.135 & 0.25-\lambda \end{pmatrix} = \lambda^2 - 0.5\lambda + 0.044275 = 0$$

$$\lambda = 0.115;\ \lambda = 0.385$$

Eigenvectors are therefore $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ with the latter one as the dominating one.

For γ = 1,

$P(u_L \cap u_R)$	$u_R = 1$	$u_R = 0$	$P(u_L)$
$u_L = 1$	0.25	0.25	$1/2$
$u_L = 0$	0.25	0.25	$1/2$
$P(u_R)$	$1/2$	$1/2$	

$$c_S = \sum_{u_R=0}^{1} P(u_R){u_R}^2 - (\sum_{u_R=0}^{1} P(u_R)u_R)^2 = 0.5 - 0.5^2 = 0.25$$

$$c_D = \sum_{u_R=0}^{1}\sum_{u_L=0}^{1} P(u_L \cap u_R)u_L u_R - \sum_{u_R=0}^{1} P(u_R)u_R * \sum_{u_L=0}^{1} P(u_L)u_L = 0.25 - 0.5^2 = 0$$

$$\boldsymbol{C} = \begin{pmatrix} c_S & c_D \\ c_D & c_S \end{pmatrix} = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix}$$

To find eigenvalues,

$$\begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.25-\lambda & 0 \\ 0 & 0.25-\lambda \end{pmatrix}$$

$$\det \begin{pmatrix} 0.25-\lambda & 0 \\ 0 & 0.25-\lambda \end{pmatrix} = (-\lambda + 0.25)^2 = 0$$

$$\lambda = 0.25$$

Eigenvectors are therefore $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ which both of them were not correlated.

For γ = 1.72,

$P(u_L \cap u_R)$	$u_R = 1$	$u_R = 0$	$P(u_L)$
$u_L = 1$	0.43	0.07	$1/2$
$u_L = 0$	0.07	0.43	$1/2$
$P(u_R)$	$1/2$	$1/2$	

$$c_S = \sum_{u_R=0}^{1} P(u_R){u_R}^2 - (\sum_{u_R=0}^{1} P(u_R)u_R)^2 = 0.5 - 0.5^2 = 0.25$$

$$c_D = \sum_{u_R=0}^{1}\sum_{u_L=0}^{1} P(u_L \cap u_R)u_L u_R - \sum_{u_R=0}^{1} P(u_R)u_R * \sum_{u_L=0}^{1} P(u_L)u_L = 0.43 - 0.5^2 = 0.18$$

$$\boldsymbol{C} = \begin{pmatrix} c_S & c_D \\ c_D & c_S \end{pmatrix} = \begin{pmatrix} 0.25 & 0.18 \\ 0.18 & 0.25 \end{pmatrix}$$

To find eigenvalues,

$$\begin{pmatrix} 0.25 & 0.18 \\ 0.18 & 0.25 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.25-\lambda & 0.185 \\ 0.18 & 0.25-\lambda \end{pmatrix}$$
$$\det \begin{pmatrix} 0.25-\lambda & 0.18 \\ 0.18 & 0.25-\lambda \end{pmatrix} = \lambda^2 - 0.5\lambda + 0.0301 = 0$$

$$\lambda = 0.43;\ \lambda = 0.07$$

Eigenvectors are therefore $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ with the former one as dominating.

The covariance matrices obtained above were the same as calculated in MatLab. Correspondingly, the calculated eigenvectors and eigenvalues expressed as the multiplication of the eigenvalue and the respective eigenvector were plotted as in *Fig. 1*.

(a)

(b)

(c)

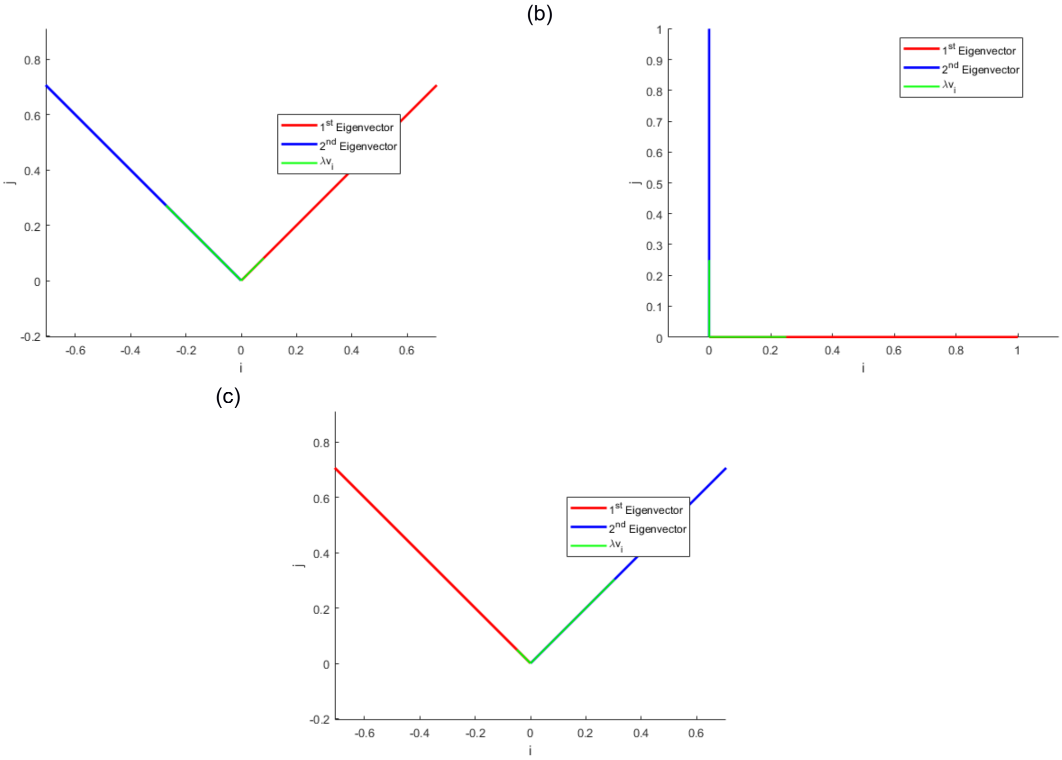

Fig. 1 Eigenvectors and eigenvalues of different covariance matrices. Using inputs with different statistics, the eigenvectors and corresponding eigenvalues were calculated as above. Notably, for cases where $\gamma \neq 0$, the eigenvectors were $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$, but with values more than 1 the first one dominates, vice versa. For $\gamma = 1$, the eigenvectors were $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ with equal significance. (a) γ = 0.46, (b) $\gamma = 1$ and (c) $\gamma = 1.72$.

Development of synaptic weight based on anti-correlated inputs could reach monocular or binocular

In *Fig. 2(a)*, since the dominating eigenvector, i.e. with the higher eigenvalue, was $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$(*Fig. 1(a)*), therefore the rate of change of weights would have a higher projection on this eigenvector, or grow along this eigenvector, and therefore the two weights competed with each other. As a result, for most of the weight developed to be monocular, i.e. the synaptic weight for one eye is saturated but the other one diminished, depending on the initial weight. The only exceptions are the initial values that both weights were close to 0.5 and they grew along the other eigenvector and developed into binocular with both synaptic weights saturated. But with Oja's rule applied, all different initial weights developed into monocular, again depending on the initial weight whether which one was larger.

(a) (b)

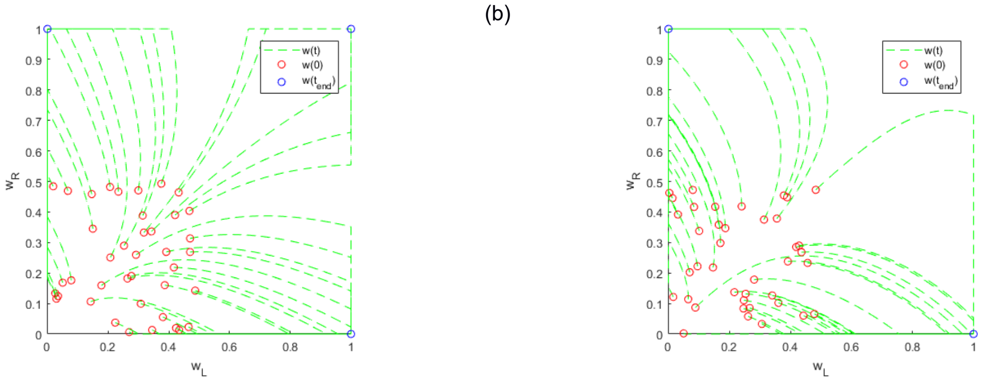

Fig. 2 Development of synaptic weights based on the covariance matrix with anti-correlated inputs. In (a), the development solely based on covariance matrix showed dependence of initial weight. When the initial values were both close to 0.5, the final weight became both 1, which means the neurons became binocular. Otherwise, the values became 1 for w_R and 0 for w_L or vice versa, i.e. the neurons became monocular, depending on the initial weight. In (b), with Oja's rule applied, the final weight became always monocular.

Development of synaptic weight based on uncorrelated inputs could only reach binocular

In *Fig. 3(a)*, since the eigenvectors were$\begin{pmatrix}0\\1\end{pmatrix}$ and $\begin{pmatrix}1\\0\end{pmatrix}$(*Fig. 1(b)*) and both of them are equally dominating, therefore the rate of change of weights would have same projection on both eigenvectors and therefore the two weights grew independent to each other. Therefore, the weights didn't grew along the principal eigenvectors but in a way dependent on their initial values. But due to the limitation of growth in weight, the weight eventually converged to binocular. Similar patterns were observed even with Oja's rule applied.

(a) (b)

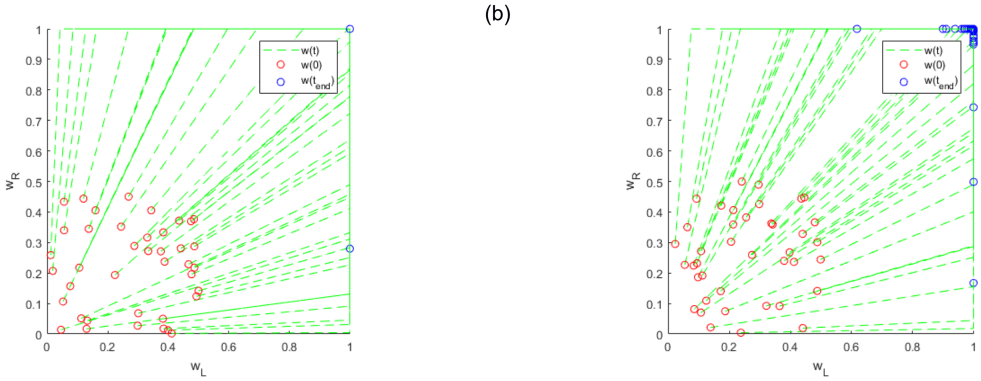

Fig. 3 Development of synaptic weights based on the covariance matrix with uncorrelated inputs. In (a), the development solely based on covariance matrix showed independence of initial weight and grew eventually to binocular. which means the neurons became binocular. Otherwise, the values became 1 for w_R and 0 for w_L or vice versa, i.e. the neurons became monocular, depending on the initial weight. In (b), with Oja's rule applied, the same result was found.

Development of synaptic weight based on correlated inputs could only reach binocular

In *Fig. 4(a)*, since the dominating eigenvector was$\begin{pmatrix}1\\1\end{pmatrix}$ (*Fig. 1(c)*), therefore the rate of change of weights would have higher projection on it and therefore the weights grew along with this eigenvector independent of their initial values. Eventually, binocular neurons were developed. Similar patterns were observed even with Oja's rule applied.

(a) (b)

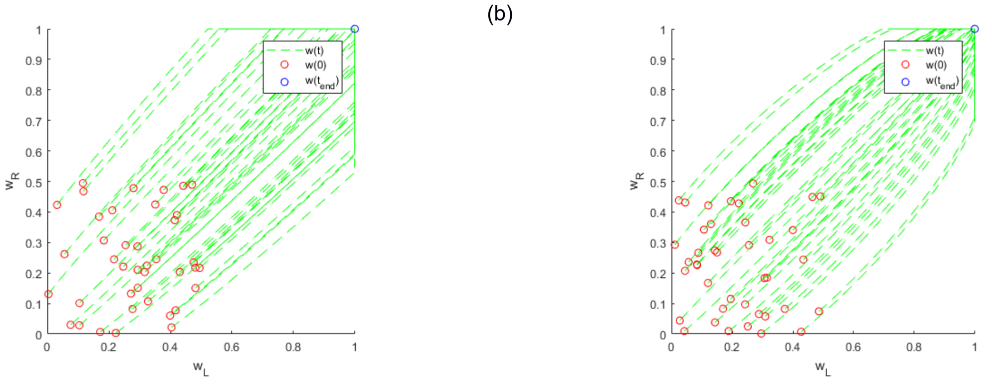

Fig. 4 Development of synaptic weights based on the covariance matrix with correlated inputs. In (a), the development solely based on covariance matrix showed independence of initial weight and grew along the dominating eigenvector. The final weight became both 1, which means the neurons became binocular. In (b), with Oja's rule applied, the final weight became also always binocular.

Conclusion

For monocular dominance to occur, the input statistics had to be similar to the first case, i.e. anti-correlated, and also the initial weight had to be not too high. In other words, the necessary condition is an initial weight not too close to 0.5 and the sufficient condition is an anti-correlated input statistics.