Computational Cognitive Neuroscience
Exercise Set 1

Due: 28.01.2016, before class

Guidelines for Submission

- Submissions can be made via email to hande.celikkanat@helsinki.fi before the deadline, or by delivering a hard copy at the beginning of the class.
- In case you are submitting both written exercises and programming exercises, please prepare two separate files respectively for submission.
- Please do submit programming exercises only via email, and as python scripts with .py extension. Also please kindly indicate with a comment which exercise you are attempting before related code.

1 NIS Ch. 19

1. Suppose $I(x, y)$ is a black-and-white image given as a vector with values 0 (black) and 1 (white) with dimensions 20×20. Design a feature detector $W(x, y)$ such that $\langle W, I \rangle$ gives the average luminance, i.e., how many white pixels there are compared to black ones. See notation of NIS Chapter 19.1.
2. Find a solution to the following equation:
$$\mathbf{Mx} = \mathbf{y}$$
where $\mathbf{x}$ is the unknown 2-dimensional vector, $\mathbf{y} = [0\ 1]^T$ and
$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$$
3. Find all possible solutions to the following system of equations:
$$\begin{cases} \mathbf{Mx} & = \mathbf{y} \\ \lambda \mathbf{x} & = \mathbf{y} \\ \|\mathbf{x}\| & = 1 \end{cases}$$
where $\mathbf{x} \in \mathbb{R}^2$ and $\lambda \in \mathbb{R}_+$ are the unknown, and (a)
$$\mathbf{M} = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}$$
(b)
$$\mathbf{M} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$
If such solutions exist, then $\mathbf{x}$ is called an *eigenvector* of $\mathbf{M}$ and λ the corresponding *eigenvalue*.

4. Let M be the 2×1 matrix $M = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$. Compute its pseudoinverse M^+ using the formula (19.25). Show that this M^+ indeed minimizes the error $\|y - MM^+y\|$ for all $y \in \mathbb{R}^2$.

5. Let $\sigma(\alpha) = \frac{d}{1+e^{-\alpha}}$ for some $d \in \mathbb{R}_+$. This is called the *sigmoid* function. Plot this function (for some constant d).

2 NIS Ch. 4

1. Show that the two expressions for variance are equivalent, i.e., for any random variable z_1 we have:
$$E\{z_1^2\} - (E\{z_1\})^2 = E\{(z_1 - E\{z_1\})^2\}.$$

2. Similarly for co-variance: Show that for random variables z_1 and z_2 we have:
$$E\{z_1 z_2\} - E\{z_1\}E\{z_2\} = E\{(z_1 - E\{z_1\})(z_2 - E\{z_2\})\}.$$

3. Prove (4.42): Show that for independent variables z_1 and z_2 we have:
$$E\{g_1(z_1)g_2(z_2)\} = E\{g_1(z_1)\}E\{g_2(z_2)\}$$
for any (measurable) functions g_1 and g_2.

4. NIS Exercise 4.10.2.

5. NIS Exercise 4.10.6.

3 Roj Ch. 1

1. How does the cell membrane potential remain constant at around $-70\ mV$ even though it is above the equilibrium for potassium ($-80\ mV$) and below the equilibrium for sodium ($58\ mV$)? (Note that the exact numbers vary across sources!)

2. In Figures 1.9 and 1.10 the x-axis is the position on the axon left-to-right being the direction of propagation, and the time point is fixed. How would the curve look like if the position on the axon was fixed and the x-axis would be time going from left to right?

3. Why doesn't the action potential change direction in the middle of the axon?

4. The original formulation of the Hebb rule by Hebb was: "When an axon of cell A is near enough to excite a 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." Look at the passage in the end of page 21 of Rojas and explain what is the difference between how Rojas puts it and the original formulation. The original formulation is more in agreement with experimental results. Rojas' formulation, however, resembles better what is done in AI, in particular this course.

5. Re-write the output of the neuron from Figure 1.14 using a dot-product.

6. Let $\mathbf{M}$ be the 2×2 matrix
$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$
Design a neural network which outputs 1 for those inputs $\mathbf{x} = [x_1\ x_2]^T$ for which $\mathbf{Mx}$ lies in the upper half of the plane including the x-axis and otherwise outputs 0.

7. Combine the idea of the previous exercise with the very first exercise of this sheet and design a neural network which takes an 20×20 image as input and outputs its luminance.

4 Python

Please use numpy/scipy/matplotlib packages for the programming exercises as a preparation step for the projects. In particular, please use the array or matrix types of numpy for representing matrices, though you may use whichever you would like, and even switch between them as you wish. Relevant packages can be imported as:

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from scipy import linalg,stats
```

1. Implement the computation of the last exercise: a python script which takes a 20×20-matrix as an input, transforms it into a 400-dimensional vector and outputs the average luminance using the feature detector.

2. Let $\mathbf{M}$ be the 4×3 matrix
$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

   - Compute its transpose $\mathbf{M}^T$.
   - Compute its pseudoinverse $\mathbf{M}^+$ using NIS formula (19.25).
   - Left-multiply the matrix by its pseudoinverse ($\mathbf{M}^+\mathbf{M}$) to acquire the 3×3 identity matrix $\mathbf{I_3}$.

3. Implement a program that outputs the initial segment of the Fibonacci sequence of a given length. The Fibonacci sequence is a sequence of natural numbers $(a_n)_{n=0}^{\infty}$ with $a_0 = a_1 = 1$ and $a_n = a_{n-1} + a_{n-2}$ for $n > 1$.

4. Implement a program that samples 100 values from each of the following distributions and plots the histograms of each of them:

   - Gaussian,
   - Uniform on $[0, 1]$,
   - Poisson.

5. Generate a 2×100-matrix with each row (an array of length 100) is an independent sample of the standard normal distribution. In other words, this matrix can be seen as a sample of 2-dimensional Gaussian vector $\mathbf{x}$ with independent entries. Calculate the covariance matrix $\text{cov}(\mathbf{x})$. Let $\mathbf{M}$ be any 2×2-matrix (generate a random one). Calculate the covariance matrix $\text{cov}(\mathbf{Mx})$ and compare the outcome with $\mathbf{MM}^T$.