**Lab 1: Introduction to Python and One-Dimensional (1-D) Convolution**

**Part A: Introduction**

1. We perform dot-products and 1-D convolutions in neural networks.

A discrete-time signal can be considered as a vector, an ordered collection of real or

complex numbers with two or more dimensions. In Python’s Numpy, vectors are 1-by-n or n-by-1 arrays, where n is the number of elements in the sequence.

First open python by typing “python”+Enter in the shell (or command prompt). You should see something like this on screen:

Python 3.5.2 (default, Nov 12 2018, 13:43:14)

[GCC 5.4.0 20160609] on linux

The output will depend on your python version and operating system.

We will now create a vector in python and display. For this purpose, we need to import numpy and matplotlib libraries. In order to do so, write the following lines:

import numpy as np

import matplotlib.pyplot as plt

Now we create a vector using numpy as follows:

x = np.array([0.,0.,1.,1.,1.,1.,1.,1.,1.,1.])

In the previous line we created an 8-by-1 vector from a python list. In order to print the vector, type “print(x)”. We now want to display the vector, to do so we write:

fig1 = plt.figure(1)

plt.stem(x)

plt.pause(0.0001)

Now we want to create a vector of range -2:7, we do so by using the following line:

k = np.arange(-2.,8.,1.)

Now we write

fig2 = plt.figure(2)

plt.stem(k,x)

plt.show()

We have two figures on screen. One is u[n] and the other is the

time-shift u[n-d]. In the printouts, indicate which one is u[n] and which one is u[n-d], and

what is the value of d?

2)

Using the following statement:

x = np.hstack([np.zeros(shape=(N)), np.ones(shape=(M+1))])

and choosing suitable vector k, display the graphics of u[n] and u[n-5] for –10≤n≤20,

respectively. Print your codes and the output figures.

3)  
Let the signal x[n]=7cos(0.1n)+cos(0.95n). Print the graphics of x[n] and x[n-20] for

-40≤n≤80, respectively.

hint: what will happen if you run cos(0.1\*k) for k=[-40:80]?

*Note: use np.cos() function to create a cosine signal*

Part B: 1-D Convolution

1. Look at the documentation of the convolution function in numpy: <https://docs.scipy.org/doc/numpy/reference/generated/numpy.convolve.html>
2. Consider the moving average system h=1./5.\*np.array([1.,1.,1.,1.,1.]). If the input signal is x, then the output signal will be y=np.convolve(x,h). For the following input signal x, print the graphic of the output.

(a) x=1 for k=[0:20]

(b). x=cos(0.1k) for k=[-40:80]

(c). x=cos(0.95k) for k=[-40:80]

(d). x is the signal obtained in Part A.3

1. Consider the difference system h=[1,-1]. For the following input signal x, print the graphic of the output.

(a). x=1 for k=[0:20]

(b). x=cos(0.1k) for k=[-40:80]

(c). x=cos(0.95k) for k=[-40:80]

(d). x is the signal obtained in Part A.3

4) According to the results you obtained in step 2 and 3, what can you say for the average system and the difference system?

Note: For non-blocking display of the figures, use the command “plt.pause(0.0001)” after each time you want use the display command

----

**Appendix 1: Report Format and Instructions:**

1. Prepare a report (including your answers/plots) to be uploaded on BB.

2. The report should be typeset (no handwriting allowed except for lengthy derivations,

which may be scanned and embedded into the report).

3. Show all steps of your work clearly.

4. Unclear presentation of results will be penalized heavily.

5. No partial credits for unjustified answers.

6. Return all Matlab/Python code that you wrote in a single .m/.py file. In Lab 1 and 2 you should return Python code.

7. Code should be commented, code for different questions should be clearly sepa-

rated.

8. The code file should NOT return an error during runtime.

9. If the code returns an error at any point, the remaining part of your code will not be

evaluated (i.e., 0 points).