PSYCH-GA.2211/NEURL-GA.2201 – Fall 2020
Mathematical Tools for Neural and Cognitive Science

Homework 1

Due: 24 Sep 2020
(late homeworks penalized 10% per day)

See the course web site for submission details. Please: don't wait until the day before the due date... start *now*!

1. **Inner product with a unit vector.** Given unit vector $\hat{u}$, and an arbitrary vector $\vec{v}$, write (MATLAB) expressions for computing:
   (a) the component of $\vec{v}$ lying along the direction $\hat{u}$,
   (b) the component of $\vec{v}$ that is orthogonal (perpendicular) to $\hat{u}$, and
   (c) the distance from $\vec{v}$ to the component that lies along direction $\hat{u}$.

   Verify that your code is working by testing it on random vectors $\hat{u}$ and $\vec{v}$ (generate these using `randn`, and don't forget to re-scale $\hat{u}$ so that it has unit length). First, do this visually with 2-dimensional vectors, by plotting $\hat{u}$, $\vec{v}$, and the two components described in (a) and (b). (hint: execute "axis equal" to ensure that the horizontal and vertical axes have the same units). Then test it numerically in higher dimensions (e.g., 4) by writing expressions to verify each of the following, and executing them on a few randomly drawn vectors $\vec{v}$:

   - the vector in (a) lies in the same (or opposite) direction as $\hat{u}$.
   - the vector in (a) is orthogonal to the vector in (b).
   - the sum of the vectors in (a) and (b) is equal to $\vec{v}$.
   - the sum of squared lengths of the vectors in (a) and (b) is equal to $||\vec{v}||^2$.

2. **Testing for (non)linearity.** Suppose, for each of the systems below, you observe some example input/output pairs of vectors (or scalars). Determine whether each system could possibly be a *linear* system. If not, explain why. If so, provide an example of a matrix that is consistent with the observed input/output pairs, and state whether you think that matrix

is unique (i.e., the *only* matrix that is consistent with the observations).

System 1:	[2, 4]	$\longrightarrow$	1
	[-2, 0]	$\longrightarrow$	3
System 2:	1	$\longrightarrow$	[4, 6]
	0.5	$\longrightarrow$	[2, 6]
System 3:	[3, 1.5]	$\longrightarrow$	[-6, -6]
	[-2, -1]	$\longrightarrow$	[4, 4]
System 4:	[1, 2]	$\longrightarrow$	[3, 1]
	[1, -1]	$\longrightarrow$	[-1, 4]
	[3, 3]	$\longrightarrow$	[5, 8]
System 5:	0	$\longrightarrow$	[0, 11]

3. **Geometry of linear transformations.** The files `sysN.p` (where `N=1,2,3,4`) each provide a function that implements a linear system whose input and output are both 2-dimensional vectors. For each of these:

   (a) Generate 30 random 2D inputs using randn. Compute the corresponding outputs. Plot 30 line segments from each input to output, labeling input and output with different symbols or colors (use hold on, and plot points at start and end of each segment). Describe, in words, what the system is doing to the input space.

   (b) Characterize the system, by measuring its response to impulses, and embedding these in a matrix. Compute the SVD of this matrix, and explain how the components of the SVD relate to the description you provided in the previous part.

4. **A simple visual neuron.** Suppose a retinal neuron in a particular species of toad generates responses that are a weighted sum of the (positive-valued) intensities of light that is sensed at 5 localized regions of the retina. The weight vector is $[1, 4, 5, 4, 1]$. (a) Is this system linear? If so, how do you know? If not, provide a counterexample. (b) What unit-length stimulus vector (i.e., vector of light intensities) elicits the largest response in this neuron? Explain how you arrived at your answer. (c) What physically-realizable unit-length stimulus vector produces the smallest response in this neuron? Explain your reasoning. [hint: visualize a simpler version of the problem, in 2 dimensions]

5. **Gram-Schmidt.** A classic method for iterative construction of an orthonormal basis is known as *Gram-Schmidt orthogonalization.* First, one generates an arbitrary unit vector (typically, by normalizing a vector created with `randn`). Each subsequent basis vector is created by generating another arbitrary vector, subtracting off the projections of that vector along each of the previously created basis vectors, and normalizing the remaining vector.

   Write a MATLAB function `gramSchmidt` that takes a single argument, N, specifying the dimensionality of the basis. It should then generate an $N \times N$ matrix whose columns contain a set of orthogonal normalized unit vectors. Try your function for $N = 3$, and plot the basis vectors (you can use MATLAB's `rotate3d` to interactively examine these). Check your function numerically by calling it for an N much larger than 3 (e.g. 1000) and verifying that the resulting matrix is orthonormal (hint: you should be able to do this without using loops). bf Extra credit: make your function *recursive* – instead of using a `for` loop, have the function call itself, each time adding a new column to the matrix of previously created orthogonal columns. To do this, you'll probably need to write two functions (a main function that initializes the problem, and a helper function that is called with a matrix containing the current set of orthogonal columns and adds a new column until the number of column equals the number of rows).

6. **Null and Range spaces.** Imagine you have a linear system characterized by matrix M, which takes as input a vector, $\vec{v}$, and outputs a vector, $\vec{y}$, such that $\vec{y} = M\vec{v}$. Explain in a few sentences what the null and range spaces of the matrix are. Imagine a creature that has a linear sensory system: it takes a vector input (e.g., an array of light intensities) and produces a vector output (a set of responses that represent attributes). If the system has as non-zero null space, what does this tell you about the creature's perceptual capabilities? If the system has a range space that does not cover the full output space, what might a neuroscientist expect to see in experimental measurements of the system's responses?

   Load the file `mtxExamples.mat` into your MATLAB world. You'll find a set of matrices named `mtxN`, with $N = 1, 2....$ For each matrix, **use the SVD** to: (a) determine if there are non-trivial (i.e., non-zero) vectors in the input space that the matrix maps to zero (i.e., determine if there's a nullspace). If so, write a MATLAB expression that generates a random example of such a vector, and verify that the matrix maps it to the zero vector; (b) write a MATLAB expression to generate a random vector y that lies in the range space of the matrix, and then verify that it's in the range space by finding an input vector, x, such that $Mx = y$.