Homework 02

Due: January 28, 9PM
Point total: 60

Instructions:

- Submit your PDF and .py file to Blackboard by the due date and time. Please do not zip your files together, as this interferes with Blackboard's preview functionality. Always show all your work, and for full credit, you must use the method that the problem instructs you to use (unless none is mentioned). Handwritten or typeset solutions are both acceptable, but unreadable submissions will be penalized. You may discuss problems with other students, but you may not write up solutions together, copy solutions from a common whiteboard, or otherwise share your written work or code. Do not use code or language that is copied from the Internet or other students; attribute the ideas *and* rephrase in your own words.

Problem 1 (12 points, 4 each)

In each case, find the matrix that performs the transformation described. (Note that you are looking for a *single* matrix, which you can find by multiplying out matrices that perform each individual operation. Be careful about which matrix goes first or last.) You can assume the direction of rotation is the one that is most convenient for our formulas.

i. In two dimensions: Swap the x and y coordinates, then rotate 30 degrees about the origin.

ii. In three dimensions: double the length, then rotate 45 degrees around the x axis, then 45 degrees around the z axis

iii. In four dimensions: Project down to three dimensions (xyz) by removing the w coordinate (and producing a three-dimensional vector), then rotate 60 degrees around the y-axis. (For the projection matrix, notice that the matrix needs to "shift" the elements to occupy new places in the smaller vector, since w came first but is now dropped.)

Problem 2 (10 points)

In each subproblem, you are given three vectors, $\vec{q}, \vec{r}$, and $\vec{s}$. Determine whether the three vectors are linearly independent. If they are not, find the dimension of their span.

i. $\vec{q} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \vec{r} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}, \vec{s} = \begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix}$

ii. $\vec{q} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \vec{r} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}, \vec{s} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$

iii. $\vec{q} = \begin{bmatrix} \pi \\ 1 \\ 1 \end{bmatrix}, \vec{r} = \begin{bmatrix} -1 \\ \pi \\ 0 \end{bmatrix}, \vec{s} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$

Problem 3 (8 pts, 4 each)

i. Find an example of four 4-dimensional vectors such that any three of them are linearly independent, but all four together are not. (Hint: You can do this with vectors of just zeros and ones.) Explain how you know how all four are linearly dependent, and explain how you know that any three of them are linearly independent.

ii. A student has proposed a strange variant of a perceptron in which the weights are always altered by $\alpha\vec{1}$ or $-\alpha\vec{1}$, where $\vec{1}$ is the all ones vector and α is a small constant. (This is instead of using the input $\vec{x}$ to calculate the adjustment to the weights.) What is the dimension of the span of these vectors? How does the dimension of the span lead us to conclude that this perceptron can't possibly learn all possible decision boundaries for n-dimensional input?

Problem 4 (9 points, 3 each)

Try going through the perceptron tutorial (`perceptron_fish_tutorial.ipynb`) before answering these questions.

i. In the provided demo code (with 20 epochs), how many total times is a perceptron classification compared to a true classification?

ii. Why do we see no decision boundary after the first epoch (and before the second)? Justify your explanation with an equation for the decision boundary in $y = mx + b$ form (round to two digits after the decimal place).

iii. If a perceptron has no bias term (weight for a constant input), is it possible that an incorrect classification of a point could result in no update to the weights? If so, describe what input(s) lead to that situation. If not, explain why not.

Problem 5 (21 points)

This problem is contained in `hw2_perceptron_transform.ipynb`. Please submit your work as a .py file along with a PDF for the other problems. (Don't submit a whole Python notebook, please.)