Homework 8 **NSC 3270 / 6270**
Due Thu Dec 9 at start of class **Fall 2021**
30 points

This is kind of two homework assignments in one (I combined these in part because I did not want an assignment due right after Thanksgiving break). Both parts are due the last day of class (Thu Dec 9). I am posting Part One now (and will discuss it in class on Tue Nov 16 and Thu Nov 18). I will post Part Two later (and we will discuss the week after Thanksgiving).

PART ONE (20 point)

Q1. This assignment asks you to use Keras to carry out a set of pattern classification analyses on one sample participant from the Haxby et al. (2001) study.

Note that this problem will require more programming (especially in terms of organizing code into multiple functions) than other assignments.

In previous assignments, you used the Keras toolbox to create, train, and test error-driven classification networks. Now you will use the same Keras framework to analyze functional neuroimaging data. This gives you hands-on experience with multivariate pattern analysis (MVPA), a common tool for analyzing neuroimaging data (with the Haxby et al. paper one of the first examples of the technique).

Recall that participants in the Haxby experiment viewed images from 8 different categories (faces, houses, cats, shoes, scissors, bottles, scrambled images, and chairs) while the fMRI scans were recorded (see class slides for details). You will create classification networks using Keras where the training patterns are the fMRI voxels recorded while participants viewed the images, and the target output values indicate which category is being viewed (using one hot coding). In other words, your networks will attempt to decode the participant's perceptions from the pattern of blood-flow changes across the ventral temporal lobe.

Lecture slides provide details regarding how to carry out certain parts of this assignment.

You will find the following data files and Jupyter notebook on Brightspace:

`haxby_vt_patterns.npy`	contains the pre-processed functional imaging data
`labels.txt`	contains the category labels and run numbers
`Homework8.ipynb`	beginning code for the assignment

Q1a (15 points). Write code to run an MVPA analysis using leave-one-out cross validation on the supplied Haxby et al. data. The class slides will explain cross-validation and provide a template for this code. As noted in class, you must write each section of the cross-validation analysis as a separate function.

Recall that a cross-validation analysis involves selecting a portion of the data for training a classifier and another portion for testing that classifier. You will be carrying out cross-validation at the level of the individual runs. Because there are 12 runs, this means there will be 12 folds of the cross validation. For each fold, one run's worth of data become the test patterns, and the rest of the data become the training patterns.

You will use a linear classifier for this part of the assignment. The class slides will note that you should not use validation items (we discuss why) and you should use a regularizer (to help prevent overfitting). You will need to select the number of training epochs as discussed in class.

You will report the classification accuracy across the 12 folds. Chance performance is 12.5%. You should be able to get classification performance somewhere in the 30-50% range. Remember that the classifier is classifying based on patterns of neural data, so you that is about as good as an MVPA classifier can get. As noted in class, sometimes MVPA analyses report classification accuracy that is only a few percentage points greater than chance.

For the assignment, you should iterate over the full leave-one-out cross-validation 5 times (in a real application, you might run it 100 times). We do this because the classifier itself is "noisy" (relying on stochastic gradient descent over a network initialized randomly). Save the classification for each of the 12 folds and note the classification performance on each of the 8 categories and for each of the 5 iterations. First, report the average classification accuracy overall. Next, examine and report the average classification for each object category. Are certain categories classified more accurately? How much variability is there across categories and across folds?

Again, details of how to implement the cross-validation will be provided in the class slides (those details supersede what is written here).

Q1b (5 points). Is the reported classifier accuracy greater than what you would expect by chance? For this part of the assignment, I want you to write the code for doing a permutation test (I am not asking you to run the full permutation test because that can take a long long time to run, especially on a laptop).

The class slides will provide details of how to do a permutation test.

Recall that the essence of a permutation test is to scramble the labels associated with the fMRI data (in a particular way, as noted on the class slides). Since by scrambling we've broken any correspondence between what is being measured with fMRI and what category the subject was looking at, any level of performance by a classifier for this "permuted" is a measure of the level of classification performance you would expect by chance.

`Homework8.ipynb` provides code for scrambling the labels (as will explained on class slides).

In a full permutation test, you would scramble the data 1000 or even 5000 times. For each permutation, you would record the level of accuracy from a classifier trained on this scrambled data. The distribution of 1000 or 5000 levels of accuracy provides a distribution of classification performance you would expect by chance. From this large number of classifications of scrambled data, you can find the critical level of accuracy (upper 5th percentile) whereby levels of classification accuracy (on the original data) would be considered "significantly greater than chance".

PART TWO (10 point)

For Part Two of this final assignment, I want you to generate some RDMs and MDS solutions for the images and network representations from the CNN model you created for Homework 7. (We will talk about RDMs and MDS the week after Thanksgiving in class.)

I have given you a large amount of code to start with in the Jupyter notebook file `Homework8PartTwoMNIST.ipynb` on Brightspace. This code illustrates the approach to use for this part of the assignment with the original MNIST dataset (and an example CNN model). You will need to adapt this code for use with the Fashion-MNIST dataset and the CNN model you submitted for Homework 7. Recall that much of the coding for MNIST and Fashion-MNIST are nearly identical to one another, so the code I gave you provides a lot of what you need. You just need to understand the code and adapt it appropriately.

You will need to refer to class slides for additional details. Much of this part of the assignment will be discussed on Thursday the week after Thanksgiving.

Q2a (1 point). You will create RDMs and MDS solutions using five examples of each of the 10 clothing types in the Fashion-MNIST dataset (feel free to adapt the code I gave you for the MNIST dataset).

Q2b (3 points). Calculate and display the RDM based on the raw pixel values of the images (the resulting RDM will be 50x50) and display a 2D MDS solution from the RDM. I gave you much of the code to get started on this. Make sure the RDM and MDS are labeled with the names of the clothing types.

Q2c (4 points). Calculate and display the RDM based on an early or intermediate layer of your trained CNN model from Homework 7 (based on the discussion from class and using the approach outlined in the ipynb file provided) and display a 2D MDS solution from this RDM. Make sure the RDM and MDS are labeled with the names of the clothing types.

Q2d (2 points). Do the same as Q2c but for the penultimate layer (the layer before the classification layer) of your CNN model from Homework 7.

Unexcused late assignments will be penalized 10% for every 24 hours late, starting from the time class ends, for a maximum of two days, after which they will earn a 0.