Neuroscience 299: Computing with High-Dimensional Vectors
Assignment 8: Locality-preserving encodings
Due October 27, 1pm

Reminder: Please do *either* the writing assignment *or* the programming assignment. Expected length for the writing assignment is approximately 250-500 words, but there is no strict minimum or maximum.

Writing assignment:

Pick either option A or option B (you need **not** and should **not** answer both).

Option A: Designing a similarity kernel. The focus paper describes how the kernel shape resulting from a Fractional Power Encoding (FPE) is determined from the phase distribution. Your response should address the following questions:

- Can you describe why, in mathematical terms, this relationship between phase distribution and kernel shape holds?
- Suppose one wanted to design a particular similarity kernel to encode with an HD vector. How would one select the phase distribution? In what situations would this not be possible?
- In what situations are some kernel shapes better than others? Give an example.

Option B: Read one of the following papers and suggest two directions for further research related to VFA (explaining which experiments you would perform in this case):

- Weiss, Cheung, & Olshausen (2016): A neural architecture for representing and reasoning about spatial relationships.
- Komer & Eliasmith (2020): Efficient navigation using a scalable, biologically inspired spatial representation.
- Frady, Kanerva, & Sommer (2018): A framework for linking computations and rhythmbased timing patterns in neural firing, such as phase precession in hippocampal place cells.

Programming assignment:

You will implement fractional power encoding using Fourier Holographic Reduced Representations (FHRR). Your assignment should include each of the following steps:

1. Define the binding, bundling, and similarity (inner/dot product) metrics for FHRR. You may have these already from Assignment 2. It is okay to use someone else's code from the Notion page, under the condition that you properly cite. In any case, it is your responsibility to make sure that the implementation you use is correct.
2. Generate a random HD vector, **z** (with phases chosen uniformly from [-pi, pi]). This is your base vector for a Fractional Power Encoding (FPE).
3. Use the base vector **z** to generate FPEs for a scalar *x* taking values in the following range [10, 50] with step 0.1 and plot the similarity kernel relative to a center of 30 (y-axis: normalized[1] inner product with $\mathbf{z}^{30}$, x-axis: x). Also plot the normalized sinc function (shifted by 30), $\sin[\pi*(x\text{-}30)]/(\pi*(x\text{-}30))$.[2] Choose the dimensionality of the base vector relatively high enough such that these plots look qualitatively similar.
4. Plot how the convergence to the normalized sinc function changes with the dimensionality of **z**. To do this, you should find a way to measure the divergence between the normalized sinc function and your approximation via the FPE, and plot the average divergence (across different random base vectors) as a function of dimensionality.
5. Next, form a 2-dimensional normalized sinc function using the FPE method. Consider two versions: one formed via the binding operation and another with the bundling operation. Visualize the similarity kernels obtained in each case. Can you tell which functions describe the shapes of the obtained kernels? Note that you should use random base vectors to encode different dimensions in your 2-D input data.
6. Generate a periodic kernel (see the paper for examples, you may generate a 1D or 2D kernel), plot its result, and describe qualitatively the kernel that you generate.
7. The sinc functions you generated in Part 5 are not circularly symmetric. Using FPE, generate a 2-D normalized sinc function that is circularly symmetric. (*Hint*: choose a phase distribution that is circularly symmetric.)

[1] Normalize the inner product by $1/n$ – where n is the dimension of **z**.

[2] Using definition that sinc(x) = 0.