SYDE 556/750
Simulating Neurobiological Systems
Lecture 14: Spatial Semantic Pointers

Chris Eliasmith

November 20, 2024

UNIVERSITY OF WATERLOO

Accompanying Readings: Dumont & Eliasmith, 2020. See here.

Contents

1 Introduction

Note: These notes are very under constructions, and mostly mathematical background. Much of the material is directly from Dumont and Eliasmith, 2020.

We define circular convolution exponentiation as:

$$B^k = \underbrace{B \circledast B \circledast \cdots \circledast B.}_{\text{k times}} \tag{1}$$

This can be written:

$$B^k = \mathcal{F}^{-1}\{\mathcal{F}\{B\}^k\}, \quad k \in \mathbb{R} \tag{2}$$

where exponentiation in the fourier domain is regular exponentiation. Where:

$$\begin{aligned}
\mathcal{F}\{X\} &= r_x e^{i\theta_x} && (3)\\
\text{So,} & && (4)\\
\mathcal{F}\{X\}^x &= \left(r_x e^{i\theta_x}\right)^x && (5)\\
&= r_x^x e^{i\theta_x x} && (6)
\end{aligned}$$

Note that because we are working with exclusively unitary vectors, $r_x = 1$, and so the θ phases completely determine the SSP that is being used. If that were not the case, the magnitude would grow exponentially or collapse to zero with repeated binding.

We can write a 2D spatial representation as:

$$S(x,y) = X^x \circledast Y^y = \mathcal{F}^{-1}\{\mathcal{F}\{X\}^x \odot \mathcal{F}\{Y\}^y\}, \tag{7}$$

where $\odot$ is the Hadamard (element-wise) product. In the frequency domain:

$$\begin{aligned}
S(x,y) &= (X^x \circledast Y^y) && (8)\\
&= \mathcal{F}^{-1}(\mathcal{F}\{X\}^x \odot \mathcal{F}\{Y\}^y) && (9)\\
&= \mathcal{F}^{-1}(r_x^x e^{i\theta_x x} r_y^y e^{i\theta_y y}) && (10)\\
&= \mathcal{F}^{-1}(r_x^x r_y^y e^{i(\theta_x x + \theta_y y)}) \qquad = \mathcal{F}^{-1}(e^{i(\theta_x x + \theta_y y)}) && (11)
\end{aligned}$$

where we have left off the vector indexes for clarity, but note that the θ for each dimension can be different.

2 Grid cells

In the above, and initial characterization of SSPs, we do not enforce any particular structure on the SSP axis vectors. However, we can choose the axis vectors to have a specific structure to get grid-like representations. To begin, let us consider an argument for why grid cells are useful for representing space.

Suppose ideal place cells are Gaussian bumps, and that they evenly cover a space to be represented. We would like to find the hidden layer activations $G \in \mathbb{R}^{n_x \times n_g}$ (where n_g is the number of hidden neurons and $n_g < n_p$, and n_p is the number of place cells) and the matrix of read-out weights $W \in \mathbb{R}^{n_g \times n_p}$ that minimize the reconstruction error of the place cell responses.

$$\min_{G,W} ||P - \hat{P}||_F^2, \tag{12}$$

$$\hat{P} = GW \tag{13}$$

The optimal W for a fixed G is given by

$$W^* = (G^T G)^{-1} G^T P. \tag{14}$$

This W should be thought of as the connection weights between the final two layers of some deep neural network. The input to the full network would be low level sensory information and the output would be the place cell activity, P. The hidden layer with activations G is the last layer before the place cells, and, since $n_g < n_p$, it creates an information bottleneck. We are interested in finding the optimal G - a compressed representation of spatial position that is optimal for reconstructing P in a single layer.

As stated in [1], if the number of place cells is large and their receptive fields uniformly cover space (and space has periodic boundary conditions) then PP^T will approximately be a circulant matrix and its eigenvectors will be Fourier modes.

Thus, the optimal responses of hidden neurons will be linear combinations of plane waves. This will produce hidden neurons with grid-like spatial responses. Adding a non-negativity constraint to this optimization problem will result in the activity of an individual hidden neuron being proportional to a sum of three plane waves whose wave vectors are 120^o degrees apart. Specifically, a column of G will have entries,

$$\sum_{j=1}^{3} e^{i\mathbf{k}_j \cdot \mathbf{x}_n} + e^{-i\mathbf{k}_j \cdot \mathbf{x}_n} \tag{15}$$

$$\text{where} \quad |\mathbf{k}_j| = |\mathbf{k}_i| \quad \forall i, j \tag{16}$$

$$\sum_{j=1}^{3} \mathbf{k}_j = 0 \tag{17}$$

The interference pattern of these waves will have a hexagonal grid pattern, like grid cells. Note that this is also real as the imaginary parts cancel. The first equation will show the interference patterns of the choice of **k** vectors – i.e. be the grid cells in G.

It might be helpful to recall Euler's formula:

$$e^{ix} = \cos(x) + i\sin(x) \tag{18}$$

to show that equation 15 is:

$$\sum_{j=1}^{3} e^{i\mathbf{k}_j\cdot\mathbf{x}_n} + e^{-i\mathbf{k}_j\cdot\mathbf{x}_n} \tag{19}$$

$$=\sum_{j=1}^{3} \cos(\mathbf{k}_j\cdot\mathbf{x}_n) + i\sin(\mathbf{k}_j\cdot\mathbf{x}_n) + \cos(\mathbf{k}_j\cdot\mathbf{x}_n) - i\sin(\mathbf{k}_j\cdot\mathbf{x}_n) \tag{20}$$

$$=2\sum_{j=1}^{3} \cos(\mathbf{k}_j\cdot\mathbf{x}_n) \tag{21}$$

which defines the interference pattern we see as grid-like for the appropriate choice of **k**. Changing the orientation of the **k** will rotate the grid, and changing the length of the **k** will change the spatial frequency (i.e., spacing) of the grid.

Returning to SSPs, we might ask if there is a way to choose the axis vectors, X and Y such that the resulting spatial representation is grid-like. That is, can we impose specific structure on the axis vectors to get grid cells? First, note that we can write multi-dimensional SSPs as:

$$S(\mathbf{x}) = \mathcal{F}^{-1}\left\{\mathrm{e}^{\mathrm{i}(A\mathbf{x})}\right\} \tag{22}$$

where A is a matrix whose columns are the axis vectors, and **x** is the 2D space we are representing. Until now, we just pick random axis-vectors, which would be random phases in the two columns of A, chosen from an even distribution around the unit circle. We call these RandSSPs. Instead, we can pick the columns and rows carefully, such that the resulting spatial representation is grid-like. We call these HexSSPs. Specifically, if we pick the rows of A to be the vertices of an equilateral triangle (in 2-dimensions, or more generally, the m-dimensional simplex), i.e. the **k** vectors above, we will get grid-like representations (see Figure 1A).

We can then rotate and scale those base **k** vectors to get different grid spacing and sizes. As shown in Figure 1A, the number of rotated vectors will determine the spacing of rings in the representation, and the number of scaled vectors will determine the smoothness of the representation. As shown in Figure 1B, the HexSSP is much better than the RandSSP for estimating a smooth point function.

3 Mathematical properties

The most critical mathematical property that SSPs have is that they preserve Euclidean relations in a much higher dimensional space. That is,

$$S(x_1, y_1) \circledast S(x_2, y_2) = S(x_1 + x_2, y_1 + y_2) \tag{23}$$

$$= X^{x_1+x_2} \circledast Y^{y_1+y_2} \tag{24}$$

So it's easy to shift a current spatial representation around without decoding the representation. We can use this to implement basic differential equations [3]:

$$S_{t+\Delta t} = \left(X^{\Delta x_t} \circledast Y^{\Delta y_t}\right) \circledast S_t, \tag{25}$$

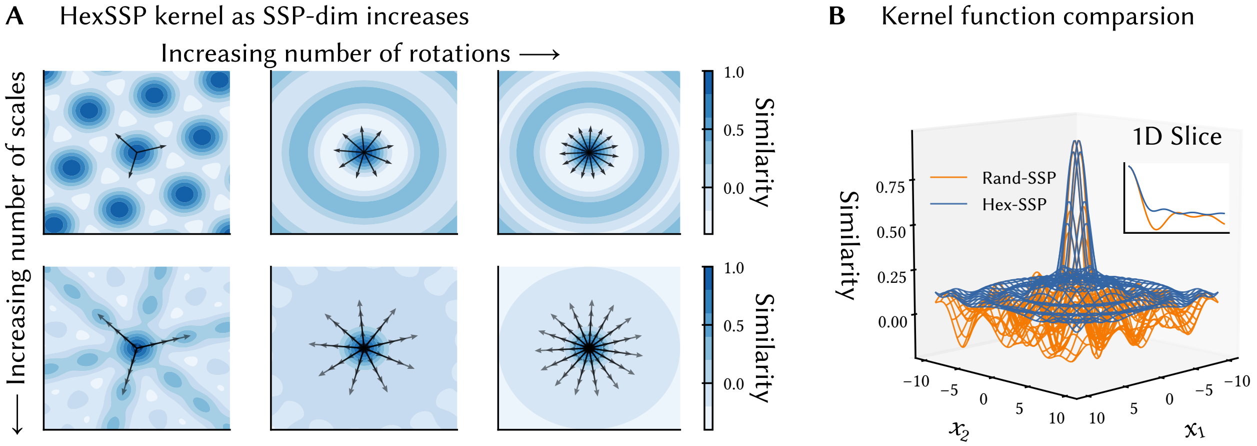

Figure 1: HexSSPs. **(A)** Interference patterns induced by HexSSPs representing points in a 2D space as a heat map. The initial 3-vector structure (top left) produces hexagonal interference patterns like those found in grid cell neurons. Adding rotated versions (top right) produces a surface with a central peak surrounded by rings. Increasing the number of rotated vectors (bottom left) results in a starburst pattern. Adding many rotated and scaled vectors (bottom right) reduces the magnitude of the surrounding rings and increases smoothness and robustness to noise. **(B)** RandSSPs result in a sinc kernel function, while HexSSPs produce a sum of sinc functions, making local maxima and minima more shallow. Figure from [2].

where Δx_t and Δy_t are derived from differential equations that relate x and y to t. Assuming $S_t = X^{x_t} \circledast Y^{y_t}$, then the algebraic properties of SSPs ensure that:

$$X^{x_t} \circledast Y^{y_t} \circledast X^{\Delta x_t} \circledast Y^{\Delta y_t} \tag{26}$$

$$= X^{x_t + \Delta x_t} \circledast Y^{y_t + \Delta y_t} \tag{27}$$

It's also possible to write a similar equation without discretizing time. The result of doing so gives:

$$\frac{dS}{dt} = \left(\frac{dx}{dt} \ln X + \frac{dy}{dt} \ln Y \right) \circledast S. \tag{28}$$

Of relevance to using SSPs as probability representations, it's important to note that as the dimensionality becomes sufficiently high, the expected similiarity approaches:

$$X^{x_1} \cdot X^{x_2} = \mathrm{sinc}\,(x_2 - x_1) \tag{29}$$

for SSPs [4].

4 Probabilities

This section is based on [5]. Assume a fixed dataset, $\mathcal{D} = \{x_1, \ldots, x_n \mid x_i \in \mathbb{R}^m\}$ of n samples of m-dimensional data.

We use a length scale parameter, h, so when we write $\mathrm{X}^{x/h}$ we mean $\mathcal{F}^{-1}\left\{ \mathrm{e}^{\mathrm{i}\theta_X x/h} \right\}$, for $x \in \mathbb{R}^m$. This parameter essentially normalizes the SSPs over the appropriate domain given the number of samples. You can find the optimal length scale for the estimator we discuss below, but it is beyond our scope.

We define our estimator as:

$$\hat{f}(x \mid \mathcal{D}) = \mathrm{X}^{x/h} \cdot \frac{1}{nh} \sum_{x_i \in \mathcal{D}} \mathrm{X}^{x_i/h} \tag{30}$$

For any domain space $x \in X \subseteq \mathbb{R}^m$, we will denote the normalized sum as:

$$M_{X,n} = \frac{1}{nh} \sum_{x_i \in \mathcal{D}} \mathrm{X}^{x_i/h} \tag{31}$$

Recall that we know that the dot product between SSPs induces a sinc function. This is a 'quasi'-kernel because it takes on negative values. So our estimator is not a Kernel Density Estimator (KDE; a common kind of density estimatory), but a special-case Fourier Integral Estimator (FIE). FIEs can be converted to probability density estimators with a correction.

The particular correction for the FIE is:

$$f_X(x) \approx \max\left\{0, \hat{f}_{\text{FIE}}\left(x \mid \mathcal{D}\right) - \xi\right\} \tag{32}$$

$\xi \in \mathbb{R}$ is selected so $\int_{-\infty}^{\infty} \max\left\{0, \hat{f}_{\text{FIE}}\left(x \mid \mathcal{D}\right) - \xi\right\} dx = 1$. The particular correction for our estimator is:

$$f_X(x) \approx \max\left\{0, \mathrm{X}^{x/h} \cdot M_{X,n} - \xi\right\} \tag{33}$$

Which looks like a ReLU with a bias of ξ. Interestingly, we can think of either the X or M as connection weights (and the other as activities). Which we choose will lead to different implementation architectures.

Interestingly, unbinding this kind of representation can be thought of as computing a conditional distribution. Briefly,

$$g(X) = f(X, Y = y) \overset{C}{\approx} \mathrm{X}^{x/h} \circledast Y^0 \cdot \sum_{x_i, y_i \in \mathcal{D}} \mathrm{X}^{x_i/h} \circledast Y^{\frac{y_i - y}{h}} \tag{34}$$

Recognizing that

$$f(X \mid Y = y) = \frac{1}{\eta} f(X, Y = y) \tag{35}$$

means unbinding can be seen as a non-normalized conditioned distribution. There are various ways we might compute the normalization constant $\eta \approx \int_{-\infty}^{\infty} \|\mathrm{X}^{x/h} \cdot M_{X|Y,n}\|^2 dx$.

We can similarly perform marginalization with SSP operations:

$$\begin{aligned}
f_X(x) &= \int_{\mathcal{Y}} f_{XY}(x, y) dy && (36) \\
&\overset{C}{\approx} \int_{\mathcal{Y}} \mathrm{X}^{x/h} \circledast \mathrm{Y}^{y/h} \cdot \left(\sum_{(x_i, y_i) \in \mathcal{D}} \mathrm{X}^{x_i/h} \circledast \mathrm{Y}^{y_i/h} \right) dy && (37) \\
&\overset{C}{\approx} \left(\mathrm{X}^{x/h} \circledast \int_{\mathcal{Y}} \mathrm{Y}^{y/h} dy \right) \cdot \left(\sum_{(x_i, y_i) \in \mathcal{D}} \mathrm{X}^{x_i/h} \circledast \mathrm{Y}^{y_i/h} \right) && (38)
\end{aligned}$$

The integral over y is a vector we can approximate by sampling Y. If we let:

$$\Phi_Y = \int_{\mathcal{Y}} \mathrm{Y}^{y/h} dy, \tag{39}$$

then

$$f_X(\mathrm{x}) \overset{C}{\approx} \left(\mathrm{X}^{x/h} \circledast \Phi_Y\right) \cdot M_{XY,n}. \tag{40}$$

Noting that circular convolution can be written as a matrix-vector product between one argument and the circulant matrix, Circ$(\cdot)$, of the other argument, we can make the following simplification:

$$\begin{aligned} \left(\mathrm{X}^{x/h} \circledast \Phi_Y\right) \cdot M_{XY,n} &= \left(\mathrm{Circ}(\Phi_Y)\mathrm{X}^{x/h}\right)^T M_{XY,n} \qquad (41) \\ &= \mathrm{X}^{x/h} \cdot \left(\mathrm{Circ}(\Phi_Y)^T M_{XY,n}\right) \qquad (42) \end{aligned}$$

So the circulant is a linear map that marginalizes M.

See the original paper for examples pertaining to entropy estimation and mutual information calculation. The latter is what is used to solve the search problem shown in the slides.

References

[1] Ben Sorscher et al. “A unified theory for the origin of grid cells through the lens of pattern formation”. In: *Advances in Neural Information Processing Systems*. 2019.

[2] P. Michael Furlong et al. “Compositional Neurosymbolic Representations Enable Efficient Active Exploration”. In: *Nature Communications* (2024 under review).

[3] Aaron R. Voelker et al. “Simulating and Predicting Dynamical Systems With Spatial Semantic Pointers”. In: *Neural Computation* 33.8 (July 2021), pp. 2033–2067. DOI: `10.1162/neco_a_01410`. URL: `https://doi.org/10.1162/neco_a_01410`.

[4] Aaron R Voelker. “A short letter on the dot product between rotated Fourier transforms”. In: *arXiv preprint arXiv:2007.13462* (2020).

[5] P. Michael Furlong and Chris Eliasmith. “Fractional Binding in Vector Symbolic Architectures as Quasi-Probability Statements”. In: *44th Annual Meeting of the Cognitive Science Society*. Cognitive Science Society, 2022. URL: `http://compneuro.uwaterloo.ca/files/publications/furlong.2022.pdf`.