Toroidal Grid Optimization via Gradient Descent

Jason Ken A

February 8, 2020

Contents

1 Problem Statement

1.1 Toroidal Graph

According to AZSPCS, a toroidal graph is defined as an $N \times N$ grid of unique tokens – here presented as IJ, where I and J represent the alphabetic indices (eg. A corresponds to 1) of the rows and columns respectively – whose edges wrap around. For $N = 4$, the grid is shown in Figure 1. The tokens outside of the square grid represent tokens which "wrap around" the grid (hence the toroidal grid).

	DA	DB	DC	DD	
AD	AA	AB	AC	AD	AA
BD	BA	BB	BC	BD	BA
CD	CA	CB	CC	CD	CA
DD	DA	DB	DC	DD	DA
	DA	DB	DC	DD	

Figure 1: A 4×4 toroidal grid

1.2 Evaluation Function

The goal of the challenge (and hence this essay) is to create a new grid – with tokens from the original grid to be arranged in any order – which minimizes a loss function (which quantifies failure) defined as follows:

1. For each pair of tokens, calculate the squared distance between them in the new grid,
2. multiply it with its squared distance within the original grid
3. The sum of these multiplications is the value of the loss

As is with all loss functions, the goal is to minimize the loss function.

1.2.1 Distance Metric

If 2 two-dimensional coordinates are defined as $s_1 = (x_1, y_1)$ and $s_2 = (x_2, y_2)$, the Euclidean distance d is defined as

$$d(s_1, s_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{1}$$

and the squared Euclidean distance d^2 evaluates to

$$d^2(s_1, s_2) = (\Delta x)^2 + (\Delta y)^2 \tag{2}$$

simplifying much of the computation.

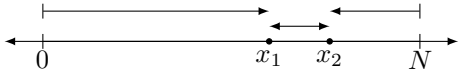


Figure 2: One-dimensional diagram of toroidal distance

But with a toroidal surface, both Δx and Δy have 2 possible values. A one-dimensional case with toroidal surface of length N is shown in Figure 2. These two distances can be written as

$$\Delta_1(x) = x_2 - x_1$$
$$\Delta_2(x) = (x_1 - 0) + (N - x_2) = x_1 + N - x_2$$

To obtain a general equation which works with both $x_1 > x_2$ and $x_1 < x_2$, we can write

$$\Delta_1(x) = |x_2 - x_1|$$
$$\Delta_2(x) = \min(x_1, x_2) + N - \max(x_1, x_2)$$

where $|a|$ is the absolute value of a, $\min(a, b)$ and $\max(a, b)$ are defined as the minimum and maximum of a and b respectively. min and max are used to determine the "left-most" and the "right-most" numbers.

Since, the distance can only have 1 value, it is defined as

$$\Delta x = \min(\Delta_1(x), \Delta_2(x))$$

Generalizing it to 2 dimensions, we get

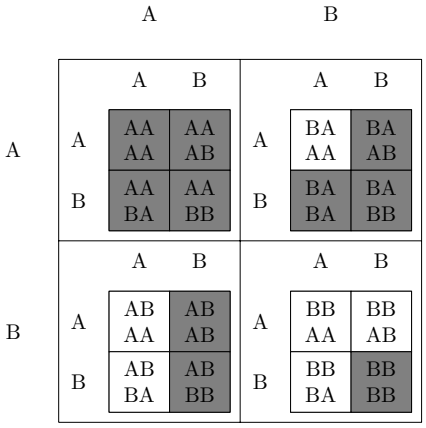
$$d(s_1, s_2) = \sqrt{(\Delta x)^2 + (\Delta y)^2}$$
$$d^2(s_1, s_2) = (\Delta x)^2 + (\Delta y)^2$$

where d^2 is the distance metric to be used in our calculations.

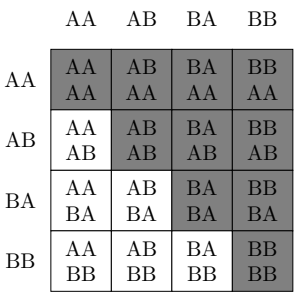
1.2.2 Token Comparisons

To be able to perform comparisons for every possible pair of tokens, a grid of comparisons (between two tokens to obtain distances) is required. A viable approach would be to to represent the comparison in a 4-dimensional grid (or tensor), where for every token in the toroidal grid, we compare it to every token. An example of the structure of the comparison for a 2×2 grid is shown in Figure 3a. The white boxes denote duplicate comparisons (eg. $\begin{smallmatrix} BA \\ AA \end{smallmatrix}$ is identical to $\begin{smallmatrix} AA \\ BA \end{smallmatrix}$), because the distance is invariant to the order of the tokens.

But since removing the irregularly shaped duplicate entries will be a hassle, we can simplify the problem by reducing the dimensionality of the grid: from a 4-dimensional grid A_{ijkl} into a 2-dimensional grid $A_{ij,kl}$. An example is shown in Figure 3b. By doing so, we can easily remove the duplicate entries by multiplying it with the upper triangular matrix, see subsubsection 1.2.3. Let $C(X)$ denote a function mapping, returning this comparison grid for an input toroidal grid X. Note that if X is a grid of the form $N \times N$, $C(X)$ has the form $N^2 \times N^2$



(a) A_{ijkl}, a 4-dimensional comparison



(b) $A_{ij,kl}$ a 2-dimensional comparison

Figure 3: Tensors C representing comparison grids of a 2×2 toroidal grid, where every element represents the toroidal distance between T_{ij} and T_{kl}, where T is the toroidal grid. Shaded cells denote unique comparisons.

1.2.3 Loss Function as Matrix Multiplications

Let $\odot$ denote the element-wise matrix multiplication, and O denotes the original grid. The loss function for toroidal grid X is equal to

$$L(x) = \sum C(X) \odot C(O) \odot U \tag{3}$$

where $C(X), C(O), U$ are matrices of the form $N^2 \times N^2$, and U_{ij} is defined with the piece-wise defined function:

$$U_{ij} = \begin{cases} 1, & j \geq i \\ 0, & j < i \end{cases} \tag{4}$$

an example the matrix U with shape 4×4 is as follows:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

2 Superposition

Since the entries into the toroidal grid are discrete (eg. AA can only be in one position), it is not possible to optimize the loss function. Therefore, relaxing the constraints to enable superposition – here defined as having a token being in multiple positions at once each with its own "probabilities" – is essential.

A simple method of allowing superposition, is by allowing any position to have any token value