Neural noise improves path representation in a simulated network of grid, place, and time cells

David M. Schwartz1 and O. Ozan Kaynaklı1,2,3
1. Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ
2. Dept. of Electrical and Computer Engineering, U.C. Berkeley, CA
3. Human R&D, Santa Clara, CA

Abstract
• The joint activity of grid, place, and time cell populations forms a neural code for paths.
• We measure the performance of a network of these populations, as well as interneurons, which implement biologically realizable denoising algorithms.
• Simulations demonstrate that representation improves when activity of a small fraction of the population is corrupted by noise.

A code for paths in space and time

De-noising network

Subspace learning
• Before denoising is possible, this network must learn (i.e., adapt its weights) the hybrid code.
• Code subspace learning is complete when the interneurons may be used to determine if the states of the pattern neurons map to a valid codeword, i.e., when the network has developed a connectivity matrix \( W \), whose rows are approximately perpendicular to the code space.
• (anti)Hebbian learning update rule:

\[
-w \rightarrow w - a (y - \langle y \rangle) x - \theta \left( y - \langle y \rangle \right) x, \quad \theta = \text{sparsity threshold}
\]

• \( a = 0.01 \) is the learning rate at iteration \( t \).
• \( y \rightarrow y' \) is the scalar projection of \( x \) onto \( w \).
• \( \theta \) is a sparsity threshold.
• \( y \) is a penalty coefficient.
• \( \Gamma \) is a sparsity enforcing function, approximating the gradient of a penalty function, \( \langle y \rangle = \sum \langle y(mw) \rangle \), which, for appropriate choices of \( \eta \), penalizes non-sparse solutions only in the learning procedure.

De-noising algorithms
• Goals: Recovers the correct pattern of activity, \( x \), from the noisy state, \( X_0 = x + n \), where \( n \) is this noise pattern.
• \( x, W \) reveals inconsistencies in \( x_0 \) that the de-noising algorithm seeks to correct in the feedback stage. To see this, consider that \( x_0 = \langle x | W \rangle = x^* - \theta x^* + \eta x^* \).

Algorithm 1: Sequential denoising

1. Initialize the denoised pattern, \( \hat{x} \rightarrow \hat{x} = x \).
2. While \( \Delta x = \| \hat{x} - x \| / \| x \| > \varepsilon \)
3. \( \hat{x} \rightarrow \hat{x} = \hat{x} - \theta x^* + \eta x^* \).
4. \( \varepsilon \rightarrow \varepsilon = \text{a decreasing function of iteration} \).
5. Repeat until stopping criteria is satisfied.

Algorithm 2: Modular Hebbian

1. Initialize the denoised pattern, \( \hat{x} \rightarrow \hat{x} = x \).
2. For each noise \( n \) in the noisy pattern, \( \hat{x} \rightarrow \hat{x} - \theta n + \eta n \).
3. \( \varepsilon \rightarrow \varepsilon = \text{a decreasing function of iteration} \).
4. Repeat until stopping criteria is satisfied.

De-noising results
• Error rate (fraction of incorrect neurons after denoising) for E random neurons before de-noising for different noise frequencies
• De-noising reduces error: \( \log(2) > \log({P_m}) \)

Discussion
• Readily de-noising codes including all cell types may be constructed by proper choice of population parameters.
• Specific accuracy of decoding position or time alone decreases with increasing frequency and ubiquity of noise.
• Average strength of connection from place cells to grid modules decreases with increasing width of place field.
• A proportion of time cells in which \( r \) is positively correlated with \( n \) exhibits the opposite trend. Surprisingly, without this correlation, the trend disappears.
• Accuracy of path representation is maximized when a small number of participating cells are subject to noise with intermediate intensity.

References

Acknowledgments
We thank Sam Gaunt, Elaine Harper, and Jean-Marc Ferreol of CNSL (Dept. of Psychology University of Arizona) for curating position and time data.
This work was supported by NSF grant no. 1464140.