Vector-based Navigation Using (Semi) Grid-Like Representations

Sean Sullivan

April 30, 2019

Introduction

Representations similar to grid cells can emerge in supervised training of rat-like motion models used for navigation within an enclosure. Besides being an efficient coding mechanism for the information in place and head direction cells, it is unclear why this phenomenon arises and whether the emergence of these representations are robust to alterations of the model. In particular, the robustness of grid-like representations in vector based navigation tasks has not been shown for variations in the motion model used for training In this work, I replicate previous models that demonstrate the emergence of grid-like neurons but train them under experimental alterations to the training data to observe and better understand the boundaries of the grid cells' emergence. Recent work by has shown that grid-like representations emerge naturally when training an agent to navigate small enclosures under a random motion model (Banino et al. 2018; Cueva and Wei 2018). To better understand the results of this paper, when they hold and to what degree, further experiments on a similar model must be run. I hypothesize that, consistent with the results of previous authors, these phenomena will emerge, under variations to the trajectories used to train the model.

Methods

The model used is based off of a rat-like motion model, inspired by Raudies and Hasselmo (2012). This model simulates a rat as an agent in a square or circular environment, or scene. Within the scene, the rat moves forward each time step with a random velocity sampled from a Rayleigh distribution, for which I used a scale parameter of 0.13 m/s. The rat also rotates randomly with an angular velocity randomly sampled from a normal distribution with a standard deviation of 330° per second and a mean of 0°. When the rat is within 0.03 meters of a wall, it turns an additional 90° per second away from its current facing direction.

These distributions and measurements are based on both those from Raudies, et al. as well as Banino et al., who both used similar models in their research based on empirical phenomena of rat movement (Raudies and Hasselmo 2012). To generate the trajectories, this simulation process was run for 750 time steps with a delta of 0.02s each time step to generate 15s worth of movement. At each step, the velocities were randomly generated as specified and then used to calculate the next position and head direction. The initial position was selected uniformly at random from within the scene and the initial head direction was sampled uniformly at random from around the unit circle. From the trajectories of 750 points, 100 linearly spaced points are taken as a point in our data set. Each of these points consists of 5 components, the velocities for each time step (angular and forward), the position at each time step, the head direction at each time step, the initial position, and the initial head direction. An example of trajectories generated by this algorithm are shown for a square and circular cage below. The square has side length of 2.2 meters and the circle has diameter of 2.2 meters.

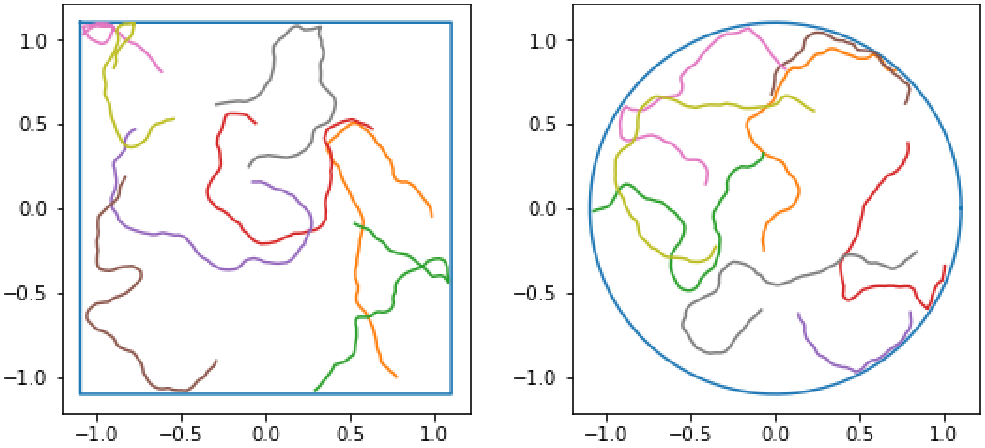

Figure 1: Square and circular scenes depicting generated trajectories. Trajectories cover the space uniformly and at random, with some bias towards staying near the walls. Each colored line represents a different trajectory, with 8 trajectories shown in each scene.

These trajectories are then used to train a model similar to that from Banino et al. We use a supervised model that takes as input a sequence of velocities, and outputs predicted place cell and head direction cell activations. We use an ensemble of 256 place cells that we model as a two dimensional mixture of Guassians:

$$\boldsymbol{c}(x) = \text{softmax}\left(-\frac{||x - \boldsymbol{\mu}||_2^2}{2\sigma^2}\right)$$

where $\boldsymbol{c} : \mathbb{R}^2 \to [0,1]^{256}$ maps a position to a probability distribution over the place cells with $\boldsymbol{\mu} \in \mathbb{R}^{256\times2}$ the place cell centers which are fixed throughout an experiment after being sampled uniformly at random over the scene. The standard deviation of the Guassians, σ is also fixed as the scale of the place cells for each experiment at a value of 0.01. Similarly, we model the head-direction cells with an ensemble of 12 cells, again with a probability distribution over the ensemble with a mixture of Von Mises distributions:

$$\boldsymbol{h}(\phi) = \mathrm{softmax}\left(\kappa \cos(\phi - \boldsymbol{\mu})\right)$$

with $\boldsymbol{h} : \mathbb{R}^2 \to [0,1]^{12}$ takes a facing angle gives a distribution over the cells with centers at $\boldsymbol{\mu} \in [-\pi, \pi]^{12}$. The centers are again sampled uniformly at random for each experiment. The concentration parameter κ is fixed to a value of 20.

Given a trajectory with head directions ϕ and positions x at each time step, $\boldsymbol{c}(x)$ and $\boldsymbol{h}(\phi)$ give us output sequence of place cell and head direction cell activations. These will be used as the supervised output for our model. The input to the model will egocentric velocities at each time step of a trajectory. For a single trajectory this is 100 time steps with 3 components at each step: $[v_t, \cos(\dot{\phi_t}), \sin(\dot{\phi_t})]$, the forward velocity and the sine and cosine of the angular velocity.

Given the velocity inputs, the model will predict the activations of the place and head direction cells, effectively performing path integration that can be recovered from an argmax over the place cell distribution at each time step.

The network architecture is as follows: the 3×100 vector is fed into an LSTM (Long Short Term Memory) layer with hidden dimension 128 (Hochreiter and Schmidhuber 1997). The initial cell and hidden state of the LSTM are computed as linear combinations of the initial place cell and head direction cell activations, $\boldsymbol{c}(x_0)$ and $\boldsymbol{h}(\phi_0)$. Thus we have the inital states of the LSTM are:

$$\begin{aligned} cs &= W_1 \boldsymbol{c}(x_0) + W_2 \boldsymbol{h}(\phi_0) + b_1 \\ hs &= W_3 \boldsymbol{c}(x_0) + W_4 \boldsymbol{h}(\phi_0) + b_2 \end{aligned}$$

We learn the weight matrices and biases through backprop during training. After passing the sequence of velocities through the recurrent LSTM layer we have a vector of size 128 at each time step. This sequence is passed through a linear layer with output size 256 to obtain a vector $g \in \mathbb{R}^{256}$ for each time step. This is the layer in which we see grid like representations emerge. Dropout is applied to the output of this linear layer with probability 0.5 (Srivastava et al. 2014). Finally, this output is passed into two further linear layers, one to the head direction cells with output size 12 and the other to the place cells with output size 256. Each of these outputs is normalized by applying a softmax function to obtain a predicted distribution $\boldsymbol{y}$ over place cells and $\boldsymbol{z}$ over head direction cells at each time step.

We minimize the cross entropy loss between these predictions and the ground truth values. Weight decay of 1×10^{-5} was applied to the final 3 linear layers and gradient clipping to the range $[-1 \times 10^{-5}, 1 \times 10^{-5}]$. This clipping prevents gradients from growing too large during backpropagation through the recurrent LSTM model. A visualization of the model architecture can be seen in figure 2.

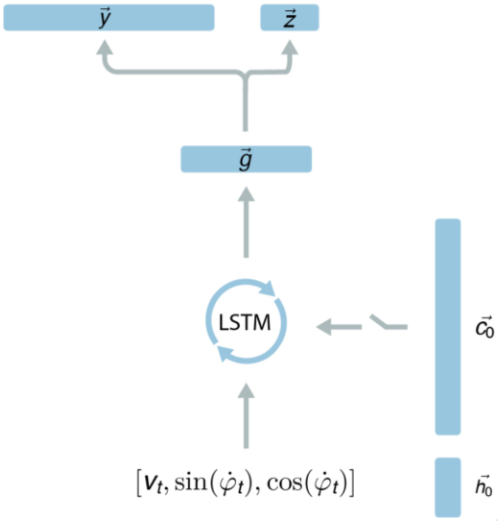

Figure 2: As in Banino et al., linear layers project initial head cell and place cell distributions to the initial state of the LSTM, which is fed egocentric velocities as input. The output of the recurrent layer is fed through three linear layers to make prediction of head and place cell activations through an intermediary grid layer.

This supervised model is inspired by the development of the entorhinal cortex and hippocampus. There is evidence that in rodents, grid cells emerge after the appearance of place cells and head direction cells (Wills et al. 2010). Furthermore, both of these cell ensembles sit very close to and could impact the development of grid cells (Barry and Burgess 2014). This is demonstrated in our model as the place and head direction cell activation are used directly in the loss function to optimize the weights in the linear layer that gives rise to the vector g. As it is in this layer that we expect to see emergent grid cells, the relationship roughly parallels the biological development of these cells. Finally, in developed rodents, grid cells have been observed to influence the activity of place cells via connections to the hippocampus (Zhang et al. 2013). This model, then, has a reasonable biological basis related to the mechanisms known to help animals path integrate.

Experiment and Results

The model implemented by Banino et al. produces grid cells given the model architecture described when trained on the trajectories from their own rat model. These trajectories, however, have different statistical properties than those found in Raudies et al. To demonstrate this difference, I measured, over 5000 trajectories, the distribution of position over all time steps in the trajectory. While the model proposed by Raudies et al. produces a somewhat uniform distribution over the space, the trajectories from Banino et al. tend to hug the walls and especially the corners. Example distributions of the two are shown in figure 3. Note that, while the model from Banino et al. has a much higher density at the walls, when this is excluded, the distribution is mostly uniformly random, much like Raudies et al. While this behavior is to be expected at the edge of an enclosure to some degree, I wanted to investigate if the emergence of grid cells was dependent on this distribution.

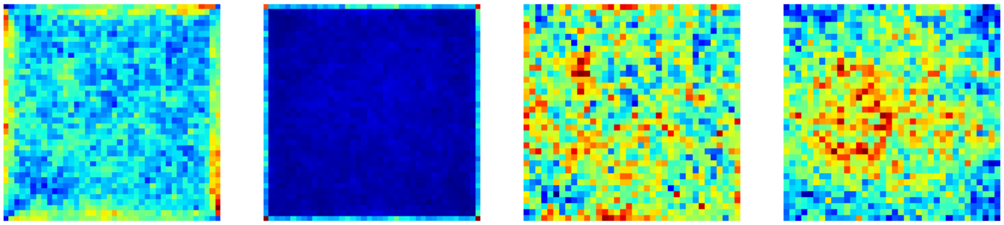

Figure 3: Distribution of position across 5000 trajectories. Red is most frequent while blue is least frequent. From left to right, distribution generated by Raudies et al., distribution generated by Banino et al. cropped distribution by Raudies et al., and cropped distribution by Banino et al. Note that the distributions are much more similar (both are near uniformly random) when the walls are cropped out. With walls included, the corners in particular of the second distribution are dark red, indicating on average the rat spends many more time steps in the corner.

To test the model's robustness to the distribution of the trajectories, I first trained the model on Deepmind's trajectories. This replicated their results and after 40 epochs of 1000 trajectories each, grid-like cells were clearly present in layer g.

The spatial autocorrelation of the ratemap is a good determinant of how grid like a cell is and can pick up on grid-like structure that is not apparent in the ratemap (Hafting et al. 2005). Here we see that many of the cells are grid-like after only 40 epochs (in the original paper this model was trained for 1000 epochs). About 30 of 256 of the units in g appear somewhat grid like after this many epochs. These grid units did not appear, however, in this quantity at earlier epochs. At epoch 24, for example, there were no grid cells as can be seen in figure 5.

In training on the more uniform trajectories, generated from my own model inspired by Raudies et al., the results are starkly different when we have trained for only 24 epochs. Here we see that 10 of the 256 units already exibit grid-like features. This result was replicable across 5 trials of these trajectories, with $8 - 13$ grid like units arising around 24-28 epochs

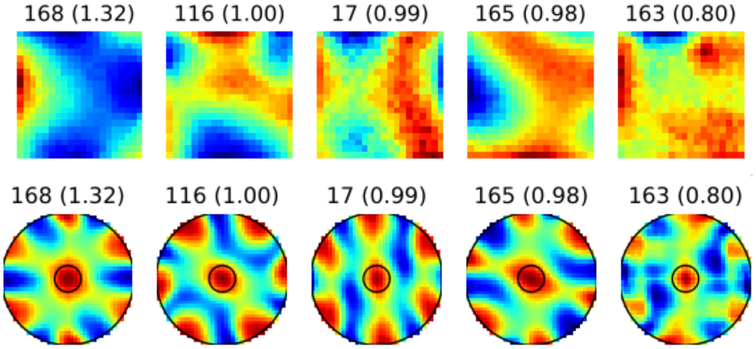

Figure 4: Spatial activation map of 5 grid cells units after 40 epochs trained on the Deepmind trajectories. This upper plots represent where in the grid each cell was active, red being most active and blue being least active. The bottom plots show the spatial autocorrelation of the maps.

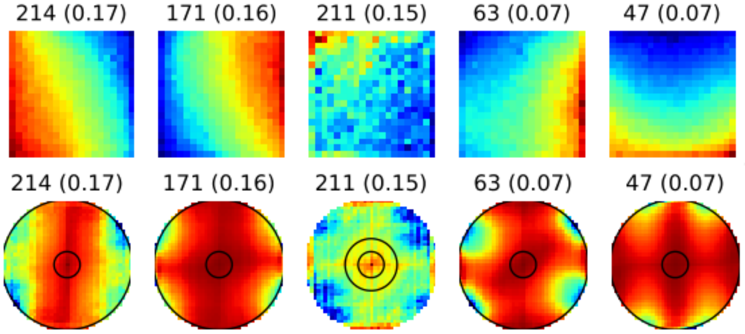

Figure 5: Spatial activation map of 5 grid cells units after 24 epochs trained on the Deepmind trajectories. None of the 256 units showed any grid patterns in either the map or the autocorrelograms after this epoch.

in. This suggests there could be some dependence on the statistics of the trajectories in the speed at which grid cells emerge. Another interesting finding, however, is that after these first grid cells emerged, the model did not improve much beyond this point. The loss stayed constant and no more grid cells emerged. In contrast, the Deepmind trajectories produced many more grid cells as reported in the original paper, with as many as 25% of the units showing grid like properties, though this took up to 1000 epochs.

This phenomenon makes sense given that the model will try to store information in the hidden layers as efficiently as possible. If there is a higher density and specificity of information uniformly around the enclosure then the model could learn a different set of rules. These results ultimately show that the model is not robust to changes in trajectory generation. With further exploration it could be possible to find a motion model that does not produce any grid cells under this model.

Another interesting phenomena that might help explain this result is the following

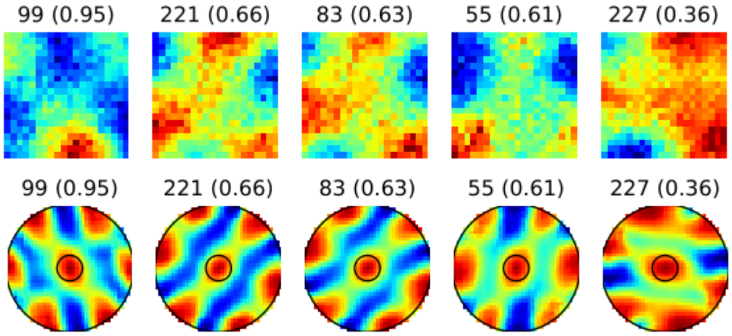

Figure 6: Spatial activation map of 5 grid cells units after 24 epochs trained on the generated trajectories. Despite the promising look of these results so early, the cells do not improve beyond this point.

examination of the data. I plotted an example trajectory and overlaid on each point in the trajectory, a vector in the rat's current facing direction. The magnitude of this vector is directly proportional to the rat's forward velocity. In the figure below we can see the comparison for the Deepmind trajectories and my own generated trajectories.

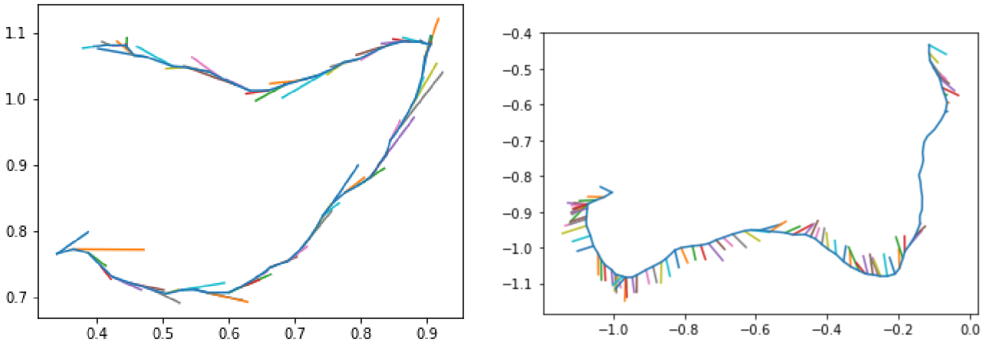

Figure 7: The left image shows a trajectory with facing directions at each point for the generated trajectories while the right shows the same for the trajectories made available publicly by Deepmind. Note how some of the facing directions in the latter are perpendicular to the path the rat is taking.

We would expect that the facing direction of the rat is tangent or nearly tangent to the curve at all points, as this is the direction in which we add the velocity to compute the next position of the rat. For the publicly available Deepmind trajectories, however, this is not the case. Here, the facing direction is at times orthogonal to the direction the rat is moving in. It makes sense, then, that the model is able to utilize the information to more rapidly generate grid cells and path integrate with the generated trajectories. There must be some information encoded in the Deepmind trajectories, however, as this model ultimately outperforms the generated trajectories. Further transparency in how Deepmind generated the trajectories would be useful in investigating this topic more.

Discussion and Conclusions

Because the model is a neural network, probing the model for causal or interpretable results is not feasible. By experimenting with inputs and seeing when things break, we can learn more about the emergent phenomena. In this case, altering the input to the model resulted in a stark contrast in output. Grid cells appeared earlier but there were fewer of them and ultimately, the performance of the model suffered. The model used to generate trajectories was realistic, with perhaps more intuitive statistics than the trajectories used in the source paper. If the trajectories are realistic and the model is capable of grid cells emerging, but they do not emerge with these trajectories, it is reasonable to ask why they don't emerge or whether a new model ought to be introduced to expand upon this.

Other directions for future work include extending this investigation to the unsupervised components of the same paper with Deepmind Labs. This, however, might be rendered moot by the instability of the model when provided alternate motion models. While the model draws some inspiration from neuroscience, it would be interesting to see how it behaves under further biological constraints. Applying competitive inhibition learning rules or using an algorithm other than backpropagation such as feedback alignment could result in a more biologically feasible model that is more stable with alternative motion. With a learning rule that is more opaque the interpretability of the model would also be improved. Rather than observing the "emergence" of grid-like representations, we could understand why this phenomena occurs. Testing the boundaries of when the model breaks is only the start of this much deeper investigation.

Acknowledgements

I would like to thank Dr. Kenneth Blum for his extensive help and insight in this project. Without him I would have lacked the inspiration to push through the many roadblocks we have encountered over the course of the semester with this project. Thank you!

The code developed for this project can be found at https://github.com/sullivan-sean/vector-nav

Bibliography

Banino, Andrea, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, et al. 2018. "Vector-Based Navigation Using Grid-Like

Representations in Artificial Agents." *Nature* 557 (7705): 429–33. https://doi.org/10.1038/s41586-018-0102-6.

Barry, C., and N. Burgess. 2014. "Neural Mechanisms of Self-Location." *Current Biology* 24 (8): R330–R339. https://doi.org/10.1016/j.cub.2014.02.049.

Cueva, Christopher J, and Xue-Xin Wei. 2018. "Emergence of Grid-Like Representations by Training Recurrent Neural Networks to Perform Spatial Localization." *arXiv Preprint arXiv:1803.07770*.

Hafting, Torkel, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I. Moser. 2005. "Microstructure of a Spatial Map in the Entorhinal Cortex." *Nature* 436 (7052): 801–6. http://search.proquest.com.ezp-prod1.hul.harvard.edu/docview/204577700?accountid=11311.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "LSTM Can Solve Hard Long Time Lag Problems." In *Advances in Neural Information Processing Systems*, 473–79.

Raudies, Florian, and Michael E. Hasselmo. 2012. "Modeling Boundary Vector Cell Firing Given Optic Flow as a Cue." Edited by Laurence T. Maloney. *PLoS Computational Biology* 8 (6): e1002553. https://doi.org/10.1371/journal.pcbi.1002553.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15: 1929–58. http://jmlr.org/papers/v15/srivastava14a.html.

Wills, T. J., F. Cacucci, N. Burgess, and J. O'Keefe. 2010. "Development of the Hippocampal Cognitive Map in Preweanling Rats." *Science* 328 (5985): 1573–6. https://doi.org/10.1126/science.1188224.

Zhang, Sheng-Jia, Jing Ye, Chenglin Miao, Albert Tsao, Ignas Cerniauskas, Debora Ledergerber, May-Britt Moser, and Edvard I. Moser. 2013. "Optogenetic Dissection of Entorhinal-Hippocampal Functional Connectivity." *Science* 340 (6128): 1232627. https://doi.org/10.1126/science.1232627.