Lecture 11: Models of the Hippocampus

2018.11.12
The Hippocampus
The Hippocampus

Latin for “seahorse”
The Hippocampus

Latin for “seahorse”

“Cortex” = archicortex (hippocampus) + neocortex (PFC, visual, etc)

archi = ancient, b/c earlier evolutionarily
HPC at the top.

The Felleman-vanEssen Diagram
The Felleman-vanEssen Diagram

HPC at the top.
Entorhinal cortex just below HPC, and above IT
Anatomy of the Tri-synaptic circuit

- CA1
- CA3
- DG (Dentate Gyrus)
- EC (Entorhinal cortex)
- PRC (Perirhinal cortex)
- IT (Inferior temporal cortex)

Regions: Optic tracts, Fimbria, Caudate nucleus, Hippocampal sulcus, Subiculum, Para-hippocampal gyrus, Fimbriodentate sulcus, Infrahinal, Dentate gyrus, CA1, CA2, CA3, CA4.
Anatomy of the Tri-synaptic circuit discovered in 1911 by the usual suspect: Ramon y Cajal
Functions of the hippocampus

1. Behavioral inhibition theory ("slam on the breaks")

Jeffrey Gray
Functions of the hippocampus

1. Behavioral inhibition theory ("slam on the breaks")

2. Memory
The Hippocampus

patient H.M.
patient H.M.

Temporal lobectomy (to treat epilepsy)

• resolved his epilepsy, but....
The Hippocampus

patient H.M.

Temporal lobectomy (to treat epilepsy)

• resolved his epilepsy, but….
• could no longer form memories (though cognitive capabilities intact)
The Hippocampus

patient H.M.

Temporal lobectomy (to treat epilepsy)

• resolved his epilepsy, but….  
• could no longer form memories (though cognitive capabilities intact)

Hippocampal dysfunction leaves old/semantic knowledge intact, but disrupts recent memory formation and new learning
The Hippocampus

Temporal lobectomy (to treat epilepsy)

- resolved his epilepsy, but....
- could no longer form memories (though cognitive capabilities intact)

Hippocampal dysfunction leaves old/semantic knowledge intact, but disrupts recent memory formation and new learning.

Consolidated old memory

Anterograde amnesia

Not-yet-consolidated
Functions of the hippocampus

1. Behavioral inhibition theory ("slam on the breaks")

2. Memory (Milner & Scoville from HM)
SIMPLE MEMORY: A THEORY FOR ARCHICORTEX

By D. MARR
Trinity College, Cambridge

(Communicated by G. S. Brindley, F.R.S.—Received 27 July 1970—Revised 12 November 1970)
SIMPLE MEMORY

(Communicated by G. S. Brindley,}

CONTENTS

0. INTRODUCTION 24
  0.1. Notation 25

1. GENERAL CONSTRAINTS 25
  1.0. Introduction 25
  1.1. Simple memory 26
  1.2. Numerical constraints 27
  1.3. The form of the analysis 29
  1.4. The consequences of the numerical constraints 30

2. THE BASIC MODEL FOR ARCHICORTEX 32
  2.0. Introduction 32
  2.1. Codon formation 32
  2.2. Diagnosis in simple memory 35
  2.3. The basic equation, and various constraints 38
  2.4. The collateral effect 40

3. CAPACITY CALCULATIONS 41
  3.0. Introduction 41
  3.1. Establishing and recovering a simple representation 41
  3.2. Justifying the model of §3.1 50
  3.3. Remarks concerning threshold setting 52
  3.4. The return from the memory 53
  3.5. Scanning during recall 54

4. A THEORY OF HIPPOCAMPAL CORTEX 54
  4.0. Introduction 54
  4.1. The morphology of the hippocampal formation 55
  4.2. The hippocampal pyramidal cells 63
  4.3. Short-axon cells in the cornu ammonis 66
  4.4. The fascia dentata 69
  4.5. Collaterals and their synapses in the hippocampus 71
  4.6. A brief functional classification of cell types 73
  4.7. The histology of various hippocampal areas 74

5. NEUROPHYSIOLOGICAL PREDICTIONS OF THE THEORY 77
  5.0. Introduction 77
  5.1. The general model for archicortex 77
  5.2. The hippocampal cortex 78

REFERENCES 80

[Published: July 1971]
HPC stores patterns immediately, w/o further analysis

Neocortex later picks out important features, might take a while (‘consolidation’)

Can support sensory -> HPC construction of “codons” then, recovery of pattern from partial input

HPC as “medium-term storage for training” deep cortex.

Two-layer recurrent model
HPC as “medium-term storage for training” deep cortex.

HPC stores patterns immediately, w/o further analysis.

Neocortex later picks out important features, might take a while (‘consolidation’).

Can support sensory -> HPC construction of “codons” then, recovery of pattern from partial input.

Three-layer recurrent model.
Models: Marr

HPC stores patterns immediately, w/o further analysis

HPC as “medium-term storage for training” deep cortex.

Figure 6. The recall problem. $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$ are the populations of cells defined in table 1. Shading represents the parts of these populations involved in the storage of an event $E_\alpha$. A newscue $\mathcal{X}$ is presented to one block of $\mathcal{P}_1$, $\mathcal{A}_0$ of whose cells were involved in $E_\alpha$ and $\mathcal{A}_1$ of which were not. This produces activity in one block of $\mathcal{P}_2$, and in $\mathcal{P}_3$, $B_i$ of the active cells in $\mathcal{P}_2$ were active in $E_\alpha$ and $B_2$ were not; $C_\alpha$ of the active cells in $\mathcal{P}_3$ were also active in $E_\alpha$, and $C_1$ were not. The numbers $A_0$, $B_i$, $C_i$ ($i = 1, 2$) are computed in the text.

- ○ = partial cue
- • = noisy cue
- ---- = two-layer
- ——— = three-layer

Third layer basically irrelevant
Why There Are Complementary Learning Systems in the Hippocampus and Neocortex: Insights From the Successes and Failures of Connectionist Models of Learning and Memory

James L. McClelland
Carnegie Mellon University
and the Center for the Neural Basis of Cognition

Bruce L. McNaughton
University of Arizona

Randall C. O’Reilly
Carnegie Mellon University
and the Center for the Neural Basis of Cognition

HPC as “medium-term storage for training” deep cortex. . .

Q: What happens if you don’t randomize ImageNet before training?
Why There Are Complementary Learning Systems in the Hippocampus and Neocortex: Insights From the Successes and Failures of Connectionist Models of Learning and Memory

James L. McClelland
Carnegie Mellon University
and the Center for the Neural Basis of Cognition

Bruce L. McNaughton
University of Arizona

Randall C. O’Reilly
Carnegie Mellon University
and the Center for the Neural Basis of Cognition

HPC as “medium-term storage for training” deep cortex. . .

Q: What happens if you don’t randomize ImageNet before training?

A: catastrophic forgetting.

. . . because you want to avoid catastrophic forgetting.
Models: Complementary Learning Systems ("CLS")

Aquisition of New Information

Interference with Existing Memories

from McClelland 2013
Models: Complementary Learning Systems ("CLS")

from McClelland 2013
Complementary Learning Systems (CLS) and their Interactions.

Connections within and among neocortical areas (green) support gradual acquisition of structured knowledge through interleaved learning.

Bidirectional connections (blue) link neocortical representations to the hippocampus/MTL for storage, retrieval, and replay.

Rapid learning in connections within hippocampus (red) supports initial learning of arbitrary new information.
Models: Complementary Learning Systems ("CLS")

Diagram showing connections between concepts such as action, name, motion, form, color, and valence, all related to the Medial Temporal Lobe.
HPC as “medium-term storage for training” deep cortex. …

… because you want to avoid catastrophic forgetting.

via experience replay and interleaving.

from Kumaran & McClelland (2012)
Models: Complementary Learning Systems ("CLS")

- Input from neocortex comes into EC; EC projects to DG, CA3, and CA1
- Drastic pattern separation occurs in DG
- Downsampling in CA3 assigns an arbitrary code
- Invertable somewhat sparsified representation in CA1
- Fewish-shot learning in DG, CA3, CA3->CA1 allows reconstruction of ERC pattern from partial input.
- Other connections shown in black are part of the slow-learning neocortical network.
- Recurrence within CA3, through the hippocampal circuit shown, and through the outer loop also involving the rest of the neocortex.

from Kumaran, Hassabis & McClelland (2016)
Models: RNNs

Simple (unrestricted) RNNs

Unlike feedforward networks, recurrent networks can **store state**.
Models: RNNs

Simple (unrestricted) RNNs

Recurrent network

input layer

hidden layers: “deep” if > 1

output layer (class/target)
Remember the context of convnets with feedback:
All RNNs executed by unrolling in time
Models: RNNs

Long-Short Term Memory (LSTMs)
Models: RNNs

Long-Short Term Memory (LSTMs)

Jürgen Schmidhuber
Models: RNNs

Long-Short Term Memory (LSTMs)

Gate:

sigmoid + pointwise multiplication

Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy
Models: RNNs

Long-Short Term Memory (LSTMs)
Models: RNNs

Long-Short Term Memory (LSTMs)

what to throw away:

\[ f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \]
Models: RNNs

Long-Short Term Memory (LSTMs)

what to store:

\[ i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \]
\[ \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \]
Models: RNNs

Long-Short Term Memory (LSTMs)

update old cell state:

\[ C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \]
Models: RNNs

Long-Short Term Memory (LSTMs)

what to actually output:

\[ o_t = \sigma (W_o [h_{t-1}, x_t] + b_o) \]
\[ h_t = o_t \times \tanh(C_t) \]
Models: RNNs

Gated Recurrent Unit (GRU) combines forget and input gate:

\[ z_t = \sigma \left( W_z \cdot [h_{t-1}, x_t] \right) \]
\[ r_t = \sigma \left( W_r \cdot [h_{t-1}, x_t] \right) \]
\[ \tilde{h}_t = \tanh \left( W \cdot [r_t \ast h_{t-1}, x_t] \right) \]
\[ h_t = (1 - z_t) \ast h_{t-1} + z_t \ast \tilde{h}_t \]
LSTMs and GRUs very useful in speech recognition and a variety of other problems with long time-scale dependencies.

... but have some severe limitations.
Models: Deep Networks with External Memory Stores

Approach: Learn parameters (weights on read / write heads) via gradient descent

Neural Turing Machines

Alex Graves, gravesa@google.com
Greg Wayne, gregwayne@google.com
Ivo Danihelka, danihelka@google.com

Google DeepMind, London, UK
Models: Deep Networks with External Memory Stores

Controller = e.g. DNN
Models: Deep Networks with External Memory Stores

**Reading**: getting / interpretation memories from memory bank (RAM)

**Writing**: changing the contents of the memory bank as a function of what's in the active focus

**Addressing**: changing the connection between the active focus and the memory bank
Models: Deep Networks with External Memory Stores

1. Reading

- \( n \) = number of memory vectors
- \( m \) = memory vector length

\[ M_t = n \times m \text{ memory matrix at time } t \]
Models: Deep Networks with External Memory Stores

1. Reading

\[ n = \text{number of memory vectors} \]
\[ m = \text{memory vector length} \]

\[ w_t = \text{read weight vector of length } n \]
\[ w_t(i) \geq 0 \quad \sum_i w_t(i) = 1 \]

\[ M_t = n \times m \text{ memory matrix at time } t \]
1. Reading

- $n$ = number of memory vectors
- $m$ = memory vector length

The read weight vector $w_t$ is defined as:

- $w_t = \text{read weight vector of length } n$
  - $w_t(i) \geq 0$
  - $\sum_i w_t(i) = 1$

The memory matrix at time $t$, $M_t$, is:

- $M_t = n \times m$ memory matrix at time $t$

The read head $R_t$ is calculated as:

- $R_t = M_t \cdot w_t$
Models: Deep Networks with External Memory Stores

1. Reading

\[ M_t = n \times m \text{ memory matrix at time } t \]

\[ n = \text{number of memory vectors} \]

\[ m = \text{memory vector length} \]

\[ w_t = \text{read weight vector of length } n \]

\[ w_t(i) \geq 0 \quad \sum_i w_t(i) = 1 \]

\[ R_t = M_t \cdot w_t \]

NB: differentiable w.r.t. \( M \) and \( w \)
2. Writing

Models: Deep Networks with External Memory Stores

- Memory bank
- Write head

$n \times m$
2. Writing

$w_t = \text{write weights of length } n$

$e_t = \text{erase vector of length } m$

$a_t = \text{add vector of length } m$

$e_t(j), a_j(t) \in [0, 1], 0 \leq j < m - 1$
Models: Deep Networks with External Memory Stores

2. Writing

- $w_t = \text{write weights of length } n$
- $e_t = \text{erase vector of length } m$
- $a_t = \text{add vector of length } m$
- $e_t(j), a_t(j) \in [0, 1], 0 \leq j < m - 1$

Update equation, separately for each $i$:

$$M_{t+1}(i) = M_t(i) \cdot (1 - w_t(i)e_t) + w_t(i)a_t$$

- $M$ memory bank
- $w_t$ write weights
- $e_t$ erase vector
- $a_t$ add vector
- $i$ index
- $m$ length of $e_t, a_t$
- $n$ length of $w_t$
2. Writing

- $w_t = \text{write weights of length } n$
- $e_t = \text{erase vector of length } m$
- $a_t = \text{add vector of length } m$
- $e_t(j), a_j(t) \in [0, 1], 0 \leq j < m - 1$

Update equation, separately for each $i$:

$$M_{t+1}(i) = M_t(i) \cdot (1 - w_t(i)e_t) + w_t(i)a_t$$

"correct" behavior in limit:

If $e_t = 1$ AND $w_t = 1$
memory is erased
Models: Deep Networks with External Memory Stores

2. Writing

\[ w_t = \text{write weights of length } n \]
\[ e_t = \text{erase vector of length } m \]
\[ a_t = \text{add vector of length } m \]

\[ e_t(j), a_t(j) \in [0, 1], 0 \leq j < m - 1 \]

Update equation, separately for each \( i \):

\[ M_{t+1}(i) = M_t(i) \cdot (1 - w_t(i) e_t) + w_t(i) a_t \]

"correct" behavior in limit:

If \( e_t = 1 \) AND \( w_t = 1 \) memory is erased

If \( e_t, a_t = 0 \) OR \( w_t = 0 \) memory is unchanged
Models: Deep Networks with External Memory Stores

2. Writing

- $w_t =$ write weights of length $n$
- $e_t =$ erase vector of length $m$
- $a_t =$ add vector of length $m$

If $e_t = 1$ and $w_t = 1$, memory is erased.
If $e_t, a_t = 0$ or $w_t = 0$, memory is unchanged.

Update equation, separately for each $i$:

$$M_{t+1}(i) = M_t(i) \cdot (1 - w_t(i)e_t) + w_t(i)a_t$$

“Correct” behavior in limit:

- If $e_t = 1$ AND $w_t = 1$, memory is erased.
- If $e_t, a_t = 0$ OR $w_t = 0$, memory is unchanged.

Formula smoothly interpolates two cases above.
But where do the $w_t$ come from?  

3. Addressing
Models: Deep Networks with External Memory Stores

But where do the $w_t$ come from? **3. Addressing**

1. **Content Addressing**: Hopfield Networks (Hopfield, 1982)  
   Shiffrin, Hintzman (Minerva II, 1984)

   address $== \text{similarity between controller (DNN) and memory}$
But where do the $w_t$ come from? 3. **Addressing**

1. **Content Addressing:** Hopfield Networks (Hopfield, 1982)
   Shiffrin, Hintzman (Minerva II, 1984)

address $== \text{similarity between}$
controller (DNN)
and
memory

$$w^c_t(i) = \beta_t \cdot \langle k_t, M_t(i) \rangle$$

$\beta_t$ = coupling strength

$k_t$ = output from the controller (DNN)

$\langle \cdot , \cdot \rangle$ = similarity measure e.g. cosine distance
But where do the $w_t$ come from?  

### 3. Addressing

1. Content Addressing:  Hopfield Networks (Hopfield, 1982)  
   Shiffrin, Hintzman (Minerva II, 1984)

   $\text{address} \equiv \text{similarity between}$
   
   controller (DNN)
   
   and
   
   memory

   $$w_t^c(i) = \text{Softmax}(\beta_t \cdot \langle k_t, M_t(i) \rangle)$$

   $\beta_t$ = coupling strength

   $k_t$ = output from the controller (DNN)

   $\langle \cdot, \cdot \rangle$ = similarity measure e.g. cosine distance
But where do the $w_t$ come from? **3. Addressing**

1. Content Addressing: Hopfield Networks (Hopfield, 1982)
   Shiffrin, Hintzman (Minerva II, 1984)

   address $\equiv$ similarity between
   controller (DNN)
   and
   memory

   $w^c_t(i) = \frac{\exp(\beta_t \cdot \langle k_t, M_t(i) \rangle)}{\sum_j \exp(\beta_t \cdot \langle k_t, M_t(j) \rangle)}$

   $\beta_t$ = coupling strength
   $k_t$ = output from the controller (DNN)
   $\langle \cdot, \cdot \rangle$ = similarity measure e.g. cosine distance
Models: Deep Networks with External Memory Stores

But where do the $w_t$ come from?  

3. Addressing

2. Location-based addressing:

convolution over vertical axis, separately for each column

$$w_t(i) = \sum_{j=0}^{n-1} w_t^c(j) \cdot s_t(i - j)$$

$s_t = \text{learned probability distribution over locations}$
But where do the $w_t$ come from?

2. Location-based addressing:

Technically:

$$w_t^g(i) = g_t \cdot w_t^c(i) + (1 - g_t)w_{t-1}(i)$$

(gated interpolation)

$$g_t \in [0, 1]$$

$$\tilde{w}_t(i) = \sum_{j=0}^{n-1} w_t^g(j) \cdot s_t(i - j)$$

(addressing)

$$w_t(i) = \frac{\tilde{w}_t(i) \gamma_t}{\sum_j \tilde{w}_t(j) \gamma_t}$$

(sharpening)

$$\gamma_t \geq 1$$
Models: Deep Networks with External Memory Stores

NB: Location-based addressing is a special case of content addressing but easier to have explicit rather than learned.
Models: Deep Networks with External Memory Stores

Previous State
\[ \begin{bmatrix} w_{t-1} \\ M_t \end{bmatrix} \]

Controller Outputs
\[ \begin{bmatrix} k_t \\ \beta_t \\ g_t \\ s_t \\ \gamma_t \end{bmatrix} \]

Content Addressing

Interpolation
\[ W_t^g \]

Convolutional Shift
\[ W_t \rightarrow W_t' \]

Sharpening
\[ W_t \rightarrow W_t' \]

params to learn + controller (DNN) params)
Models: Deep Networks with External Memory Stores

Exp I: Copy

**input:** random binary vectors $V$ (varying lengths) .... then delimiter

**output:** $V$

![Graph showing cost per sequence (bits) versus sequence number (thousands)]
Models: Deep Networks with External Memory Stores

Exp 1: Copy

Memory usage patterns

- Inputs
- Outputs
- Adds
- Reads

Location

Time

Write Weightings

Time

Read Weightings
Models: Deep Networks with External Memory Stores

Exp 2: Copy Generalization

**input:** random binary vectors $V$ (varying lengths $\leq N$) ... then delimiter

**output:** $V$, but in testing lengths $\geq N$
Models: Deep Networks with External Memory Stores

**NTM**

<table>
<thead>
<tr>
<th>Targets</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Targets" /></td>
<td><img src="image2" alt="Outputs" /></td>
</tr>
</tbody>
</table>

**LSTM**

<table>
<thead>
<tr>
<th>Targets</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3" alt="Targets" /></td>
<td><img src="image4" alt="Outputs" /></td>
</tr>
</tbody>
</table>

Time
Exp 3: Repeat Copy

**input:** random binary vectors $V$ (varying lengths $\leq N$), integer $k$

**output:** $V$ repeated $k$ times, then delimiter

*It’s been hard to get LSTMs to do well on this sort of task.*

![Figure 7: Repeat Copy Learning Curves.](image)
Models: Deep Networks with External Memory Stores

**NTM**
Length 10, Repeat 20

Targets

Outputs

Length 20, Repeat 10

Targets

Outputs

**LSTM**
Length 10, Repeat 20

Targets

Outputs

Length 20, Repeat 10

Targets

Outputs
Exp 4: Associative Recall

**input**: start codon + sequence of items (random binary) + stop codon

**testing**: random element of sequence, **output**: next element

*Figure 10: Associative Recall Learning Curves for NTM and LSTM.*
Exp 4: Associative Recall

**input**: start codon + sequence of items (random binary) + stop codon

**testing**: random element of sequence, **output**: next element

![Figure 11: Generalisation Performance on Associative Recall for Longer Item Sequences.](image)

The NTM with either a feedforward or LSTM controller generalises to much longer sequences of items than the LSTM alone. In particular, the NTM with a feedforward controller is nearly perfect for item sequences of twice the length of sequences in its training set.
Models: Deep Networks with External Memory Stores

Exp 4: Associative Recall

Memory usage patterns

Inputs
Outputs

Add:

Reads

Location

Time → Write Weightings

Time → Read Weightings
Exp 5: Dynamic N-grams

**input**: stream generated by binary n-gram model

**output**: next element of sequence

**specifically**: 6-gram transition matrices (2x5) generated from beta(1/2, 1/2) distribution of 2x5s

Beta distribution chosen because it describes the statistics of a nice special case of how sequences of temporally-interrelated data often arise (e.g. the Chinese restaurant process, dirichlet distribution)
Exp 5: Dynamic N-grams

**input:** stream generated by binary n-gram model

**output:** next element of sequence

*Figure 13: Dynamic N-Gram Learning Curves.*
Exp 5: Dynamic N-grams

Memory usage patterns

- Add Vectors
- Write Weights
- Predictions
- Inputs
- Read Weights

Time
Exp 6: Sorting

**input:** sequence of binary vectors + scalar priority list

**output:** vectors sorted by priority
Exp 6: Sorting

**input**: sequence of binary vectors + scalar priority list

**output**: vectors sorted by priority
NTM experiment details:

<table>
<thead>
<tr>
<th>Task</th>
<th>#Heads</th>
<th>Controller Size</th>
<th>Memory Size</th>
<th>Learning Rate</th>
<th>#Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>1</td>
<td>100</td>
<td>$128 \times 20$</td>
<td>$10^{-4}$</td>
<td>17,162</td>
</tr>
<tr>
<td>Repeat Copy</td>
<td>1</td>
<td>100</td>
<td>$128 \times 20$</td>
<td>$10^{-4}$</td>
<td>16,712</td>
</tr>
<tr>
<td>Associative</td>
<td>4</td>
<td>256</td>
<td>$128 \times 20$</td>
<td>$10^{-4}$</td>
<td>146,845</td>
</tr>
<tr>
<td>N-Grams</td>
<td>1</td>
<td>100</td>
<td>$128 \times 20$</td>
<td>$3 \times 10^{-5}$</td>
<td>14,656</td>
</tr>
<tr>
<td>Priority Sort</td>
<td>8</td>
<td>512</td>
<td>$128 \times 20$</td>
<td>$3 \times 10^{-5}$</td>
<td>508,305</td>
</tr>
</tbody>
</table>

Table 1: NTM with Feedforward Controller Experimental Settings
Symbolic Reasoning with Differentiable Neural Computers


More recent (somewhat more powerful) version:
Models: Deep Networks with External Memory Stores

More recent (somewhat more powerful) version:

Symbolic Reasoning with Differentiable Neural Computers


key additional feature: Temporal addressing
Functions of the hippocampus

1. Behavioral inhibition theory ("slam on the breaks")

2. Memory (Milner & Scoville from HM)

3. Spatial cognition
Hippocampus as a cognitive map

John O'keefe
Hippocampus as a cognitive map

https://www.youtube.com/watch?v=lfNVv0A8Qvl
Hippocampus as a cognitive map

Hippocampal Place Cells (*O’keefe & Nadel, 1970s*)
Hippocampal Place Cells

Get formed quickly and just a quickly remapped
Entorhinal Grid Cells (Mozers, 2005)

Entorhinal Grid Cells (Mozers, 2005)

- Grids are hexagonal and independent of arena size
- Maintain alignment with visual landmarks.
Entorhinal Grid Cells (Mozers, 2005)

There are multiple maps of different grid spacings.

Hafting et al., 2005
Entorhinal Grid Cells (Mozers, 2005)
Boundary cells also found in subiculum (part of HPC) and ERC.

Firing of a boundary cell recorded in rat subiculum in 1 x 1 metre square-walled box with 50 cm-high walls. A 50 cm-long barrier inserted into box elicits second field along north side of barrier in addition to original field along south wall.
Hippocampus as a cognitive map
Hippocampus as a cognitive map
Models: Spatial Attractor of Path Integrator

Path integration and the neural basis of the ‘cognitive map’

Bruce L. McNaughton*, Francesco P. Battaglia, Ole Jensen†, Edvard I. Moser‡
and May-Britt Moser†
Models: Spatial Attractor of Path Integrator

Path integration and the neural basis of the ‘cognitive map’

Bruce L. McNaughton*, Francesco P. Battaglia§, Ole Jensen¶, Edvard I. Moser¶ and May-Britt Moser†
Models: Spatial Attractor of Path Integrator

One-d attractor map

One ring of cells for clockwise, one ring for counterclockwise
Models: Spatial Attractor of Path Integrator

Path integration and the neural basis of the ‘cognitive map’

Bruce L. McNaughton*, Francesco P. Battaglia*, Ole Jensen†, Edvard I. Moser†
and May-Britt Moser†

Two-D grid version
but *weirdness* at boundary

Two-D grid version
Models: Spatial Attractor of Path Integrator
Models: Spatial Attractor of Path Integrator
Models: Spatial Attractor of Path Integrator
Models: Spatial Attractor of Path Integrator

Accurate Path Integration in Continuous Attractor Network Models of Grid Cells

Yoram Burak¹ ², Ila R. Fiete³ ⁴
1 Center for Brain Science, Harvard University, Cambridge, Massachusetts, United States of America, 2 Kavli Institute for Theoretical Physics, University of California Santa Barbara, Santa Barbara, California, United States of America, 3 Computer Science and Neural Systems, Division of Biology, California Institute of Technology, Pasadena, California, United States of America

Square grid (128x128), with toroidal wraparound

inhibitory input from surround ring of neurons

coupling to velocity (from subiculum?)

dependence of emergent pattern on strength of inhibition
Models: Spatial Attractor of Path Integrator

Accurate Path Integration in Continuous Attractor Network Models of Grid Cells

Yoram Burak\textsuperscript{1,2,\ast}, Ila R. Fiete\textsuperscript{2,3}

1 Center for Brain Science, Harvard University, Cambridge, Massachusetts, United States of America, 2 Kavli Institute for Theoretical Physics, University of California Santa Barbara, Santa Barbara, California, United States of America, 3 Computation and Neural Systems, Division of Biology, California Institute of Technology, Pasadena, California, United States of America

periodic boundary

aperiodic boundary
Accurate Path Integration in Continuous Attractor Network Models of Grid Cells

Yoram Burak¹,², Ila R. Fiete²,³

¹ Center for Brain Science, Harvard University, Cambridge, Massachusetts, United States of America, ² Kavli Institute for Theoretical Physics, University of California Santa Barbara, Santa Barbara, California, United States of America, ³ Computation and Neural Systems, Division of Biology, California Institute of Technology, Pasadena, California, United States of America

Now driven by real rat motion data

A. Instantaneous activity of model units

B. Time-average grid-cell response of real rat

C. Drift
EMERGENCE OF GRID-LIKE REPRESENTATIONS BY TRAINING RECURRENT NEURAL NETWORKS TO PERFORM SPATIAL LOCALIZATION

Christopher J. Cueva; Xue-Xin Wei
Columbia University
New York, NY 10027, USA
{ccueva, weixxpku}@gmail.com

ABSTRACT

Decades of research on the neural code underlying spatial navigation have revealed a diverse set of neural response properties. The Entorhinal Cortex (EC) of the mammalian brain contains a rich set of spatial correlates, including grid cells which encode space using tessellating patterns. However, the mechanisms and functional significance of these spatial representations remain largely mysterious. As a new way to understand these neural representations, we trained recurrent neural networks (RNNs) to perform navigation tasks in 2D arenas based on velocity inputs. Surprisingly, we find that grid-like spatial response patterns emerge in trained networks, along with units that exhibit other spatial correlates, including border cells and band-like cells. All these different functional types of neurons have been observed experimentally. The order of the emergence of grid-like and border cells is also consistent with observations from developmental studies. Together, our results suggest that grid cells, border cells and others as observed in EC may be a natural solution for representing space efficiently given the predominant recurrent connections in the neural circuits.
**EMERGENCE OF GRID-LIKE REPRESENTATIONS BY TRAINING RECURRENT NEURAL NETWORKS TO PERFORM SPATIAL LOCALIZATION**

Christopher J. Cueva, Xue-Xin Wei
Columbia University
New York, NY 10027, USA
{ccueva, weixx}gmail.com
RNN governing equation:

\[
\frac{dx_i}{dt} = -x_i(t) + \sum_{j=1}^{N_{rec}} W_{ij}^{rec} \tanh(x_i(t)) + \sum_{k=1}^{N_{inp}} W_{ik}^{inp} I_k(t) + \xi_i(t)
\]

\(x_i\)'s are the internal neurons

\[
N_{rec} = 100
\]

weights and biases trained to produce desired output.

\(y_j\)'s are the desired readouts

\[
y_j(t) = \sum_{i=1}^{N_{rec}} W_{ji}^{out} \tanh(x_i(t))
\]
RNN governing equation:

\[
\tau \frac{dx_i}{dt} = -x_i(t) + \sum_{j=1}^{N_{\text{rec}}} W_{ij}^{\text{rec}} \tanh(x_i(t)) + \sum_{k=1}^{N_{\text{inp}}} W_{ik}^{\text{inp}} I_k(t) + \xi_i(t)
\]

\(x_i\)'s are the internal neurons
\(N_{\text{rec}} = 100\)

weights and biases trained to produced desired output → \(y_j\)'s are the desired readouts

\[
y_j(t) = \sum_{i=1}^{N_{\text{rec}}} W_{ji}^{\text{out}} \tanh(x_i(t))
\]

generated as modified Brownian motion
Emergence of grid-like representations by training recurrent neural networks to perform spatial localization

Christopher J. Cueva, Xue-Xin Wei
Columbia University
New York, NY 10027, USA
{ccueva, weixx}@gmail.com

Figure 2: Different types of spatial selective responses of units in the trained RNN. Example simulation results for three different environments (square, triangular, hexagon) are presented. Blue (yellow) represents low (high) activity. a) Grid-like responses. b) Band-like responses; c) Border-related responses; d) Spatially irregular responses. These responses can be spatially selective but they do not form a regular pattern defined in the conventional sense.
Emergence of grid-like representations by training recurrent neural networks to perform spatial localization

Christopher J. Cueva, Xue-Xin Wei
Columbia University
New York, NY 10027, USA
{ccueva, weixxpku}@gmail.com

Figure 5: Complete set of spatial response profiles for 100 neurons in a RNN trained in a square environment. a) Without proper regularization, complex and periodic spatial response patterns do not emerge. b) With proper regularization, a rich set of periodic response patterns emerge, including grid-like responses. Regularization can also be adjusted to achieve spatial profiles intermediate between these two examples.
Goal-Driven Models

Vector-based navigation using grid-like representations in artificial agents

Similar results from another group about the same time.
Many units in model are irregular — neither grid-like nor band-like nor border-like …
Many units in model are irregular — neither grid-like nor band-like nor border-like ...

... but this is actually true in real entorhinal cortex as well. According to Lisa Giacomo, perhaps 70% of ERC cells are “irregular”.

Figure 2: Different types of spatial selective responses of units in the trained RNN. Example simulation results for three different environments (square, triangular, hexagon) are presented. Blue (yellow) represents low (high) activity. a) Grid-like responses. b) Band-like responses; c) Border-like responses; d) Spatially irregular responses. These responses can be spatially selective but do not form a regular pattern defined in the conventional sense.
Analogy to visual system results

In both cases, striking qualitative features of “characteristic neurons” that neuroscientists feel are important can be shown to just “emerge” from the system achieving down-stream computational goal …

grid-cell-like tuning in navigation-based NN model of ERC

Gabor-like and center-surround tuning in early layer of categorization-based NN model of ventral stream
Analogy to visual system results

In both cases, striking qualitative features of “characteristic neurons” that neuroscientists feel are important can be shown to just “emerge” from the system achieving down-stream computational goal …

… but actually, just as interesting, these goal-driven NN models have many “non-characteristic” units that differ from the characteristic neurons — and in fact, so do the real brain areas. So, perhaps the goal-driven models go substantially beyond the intuitions of neuroscientists in a way that is brain-like.
Hippocampus as a cognitive non-spatial map

**Mapping of a non-spatial dimension by the hippocampal-entorhinal circuit**

Dmitriy Aronov\(^1\), Rhino Nevers\(^1\) & David W. Tank\(^1\)

“During spatial navigation, neural activity in the hippocampus and the medial entorhinal cortex (MEC) is correlated to navigational variables such as location, head direction, speed, and proximity to boundaries. These activity patterns are thought to provide a maplike representation of physical space. However, the hippocampal-entorhinal circuit is involved not only in spatial navigation, but also in a variety of memory-guided behaviours.

A conceptual framework reconciling these views is that spatial representation is just one example of a more general mechanism for encoding continuous, task-relevant variables.”
Hippocampus as a cognitive non-spatial map

LETTER

doi:10.1038/nature21692

Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit

Dmitriy Aronov, Rhino Nevers & David W. Tank

“During spatial navigation, neural activity in the hippocampus and the medial entorhinal cortex (MEC) is correlated to navigational variables such as location, head direction, speed, and proximity to boundaries. These activity patterns are thought to provide a maplike representation of physical space. However, the hippocampal–entorhinal circuit is involved not only in spatial navigation, but also in a variety of memory-guided behaviours.

A conceptual framework reconciling these views is that spatial representation is just one example of a more general mechanism for encoding continuous, task-relevant variables.”
Hippocampus as a cognitive non-spatial map

Mapping of a non-spatial dimension by the hippocampal-entorhinal circuit

Dmitriy Aronov, Rhino Nevers & David W. Tank

Figure 1 | Sound modulation task. a, Schematic of the SMT. Rat deflects a joystick to increase sound frequency and must release it in a target zone. J, joystick; L, lick tube; N, nosepoke; S, speaker. b, For a single session, frequencies at which the joystick was released on individual trials (bottom), and the distribution of these frequencies across trials (top). Most releases occurred early in the target zone (green). c, Same data as in b, but plotted as a function of time. The COV indicates a bigger spread of the distribution. d, COV values of frequencies and times at the joystick release across all 189 sessions from 9 rats (blue). Red circles, median values across sessions for each of the rats.
Functions of the hippocampus

1. Behavioral inhibition theory (“slam on the breaks”)

2. Memory (Milner & Scoville from HM)

3. Spatial cognition

2.5: memory as map of conceptual space.
The neural map is the agent's internal memory storage that can be read from and written to during interaction with its environment, but where the write operator is selectively limited to affect only the part of the neural map that represents the area where the agent is currently located.
A Hard Memory Task?

Montezuma's Revenge

Discussion about the content page [ctrl-c]