Modeling Hippocampus in context of language

Philipp Rost

Friedrich-Alexander-Universität Erlangen-Nürnberg

May 18, 2022

Agenda

Hippocampus

Hippocampus

- Has the shape of a seahorse
- Key factor in forming memories (not preserving them!)
- Related to emotions
- Responsible for any kind of navigation (i.e. ranking stuff like danger of animals)
  - Crafts a **cognitive room** of the “surroundings” by using **place cells** and **grid cells**
  - Place cell: irregular arranged, fires at specific positions in space
  - Grid cell: lattice-like arranged, fires continuously

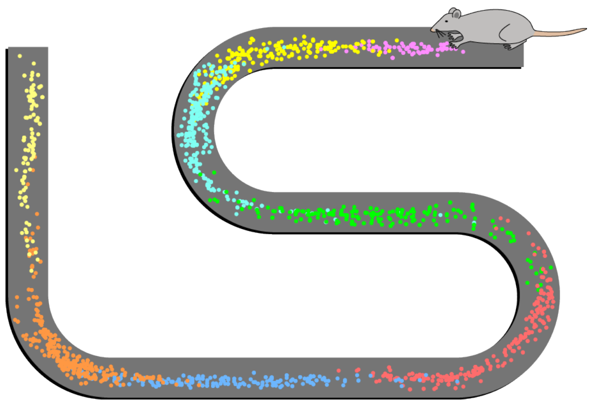


Figure: Different colors represent different place cells (= states)

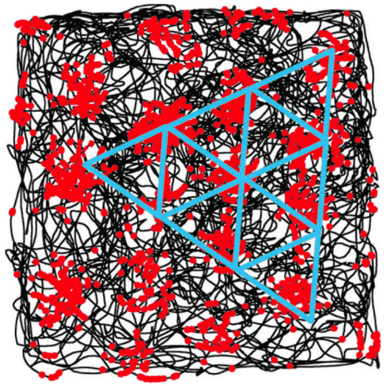
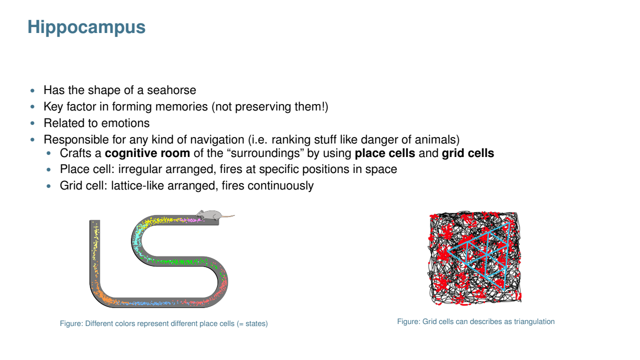


Figure: Grid cells can describes as triangulation



- for classical navigational tasks we have “place cells” and “grid cells”
- “specific position”: snippet of pathway or in the staircase
- Beispiel für surroundings: These cells map pitches (Original: Orts- und Rasterzellen kartieren akustische Tonhöhen; deutschlandfunk-Artikel)

Referenzen:

- Neue Erinnerungen formen: Pauls Masterarbeit
- Emotionen: Wikipedia-Artikel (dort stehen Quellen)
- Navigation: Steht sicher auch etwas bei Wikipedia, sonst PMA oder SR-Paper

Successor Representation

- Claim: Hippocampus learns a map representing each state as a successor state which are encoded via place cell
  - Where does the claim originates? The proposed technique works fine for spatial navigation.
- Mathematical principle roots in reinforcement learning (and transition probability matrices) and can be computed like

$$M_a = \sum_{t=0}^{a} \gamma^t T^t,$$

  with discount factor $\gamma \in (0, 1)$ and transition probability matrix T
- The policy is encoded in matrix $T \implies$ the SR is policy-dependent (it is based on RL)
- Row i: resembles (all) successor states of state i (the higher the value the higher the chance to be in this state after the next step)
- Column j: High value in component i means this state is visited often after starting at state i

Successor Representation

- Claim: Hippocampus learns a map representing each state as a successor state which are encoded via place cell
  - Where does the claim originates? The proposed technique works fine for spatial navigation.
- Mathematical principle roots in reinforcement learning (and transition probability matrices) and can be computed like

$$M_a = \sum_{t=0}^{a} \gamma^t T^t,$$

  with discount factor $\gamma \in (0, 1)$ and transition probability matrix T
- The policy is encoded in matrix $T \implies$ the SR is policy-dependent (it is based on RL)
- Row i: resembles (all) successor states of state i (the higher the value the higher the chance to be in this state after the next step)
- Column j: High value in component i means this state is visited often after starting at state i

– You can also calculate the SR up to infinity

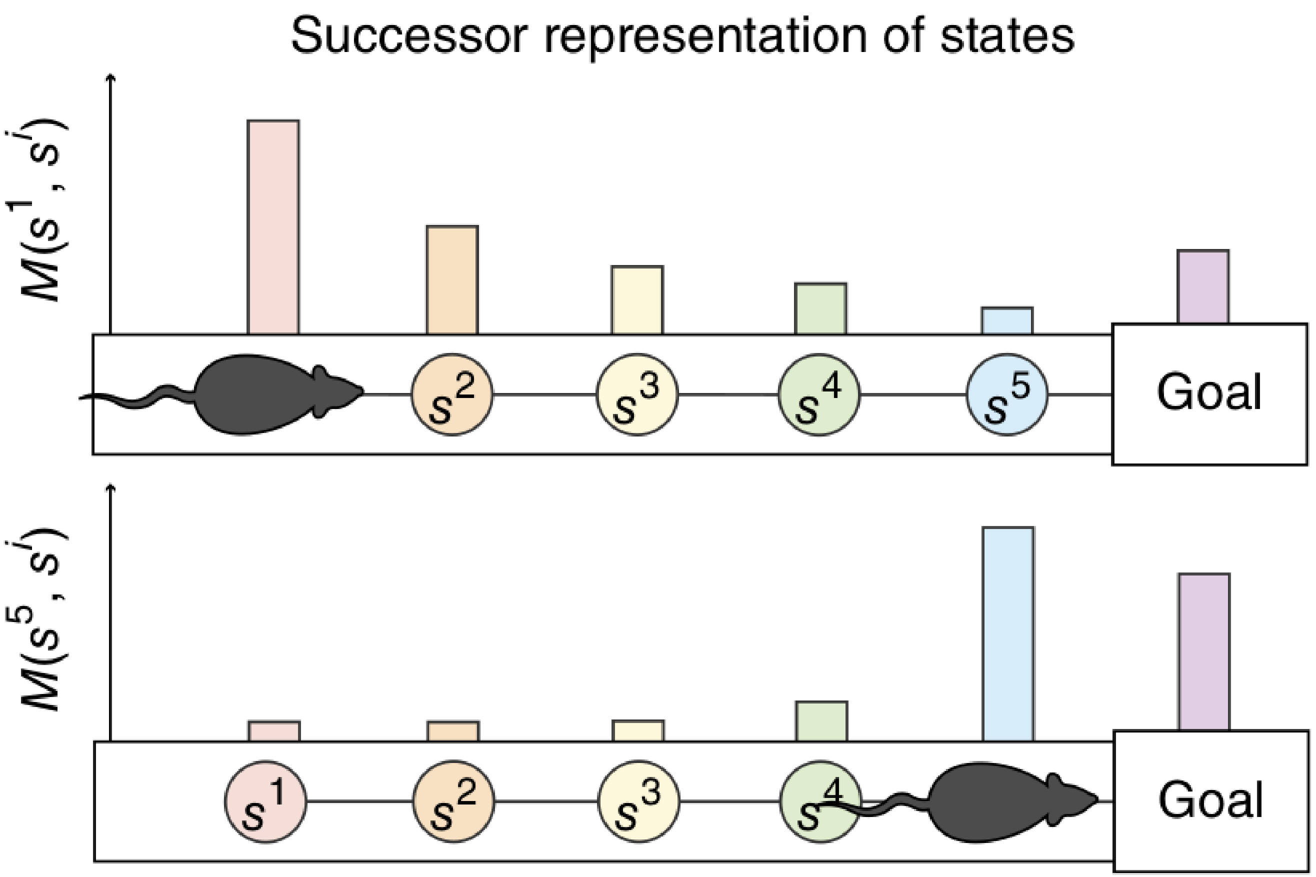
– By γ you can control how influential further apart states are

– By multiplying T with itself n-times you receive the probabilities of being in an arbitrary position after n steps. So, the SR is just a “layerd” transition probability matrix T.

– Dadurch dass die vielen Übergangsmatrizen „übereinander“ gelegt werden, erhöhen sich die Werte in jedem Feld mit jedem Summanden. Ist ein Wert nun hoch, bedeutet das, man war „in vielen Zwischenständen“ dort.

Successor Representation

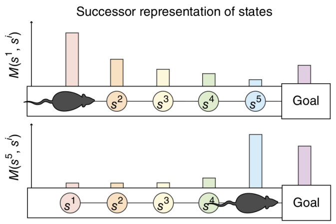
Example 1/2 of interpreting a SR matrix *M*



Superscript means index not power

Successor Representation

Example 1/2 of interpreting a SR matrix M

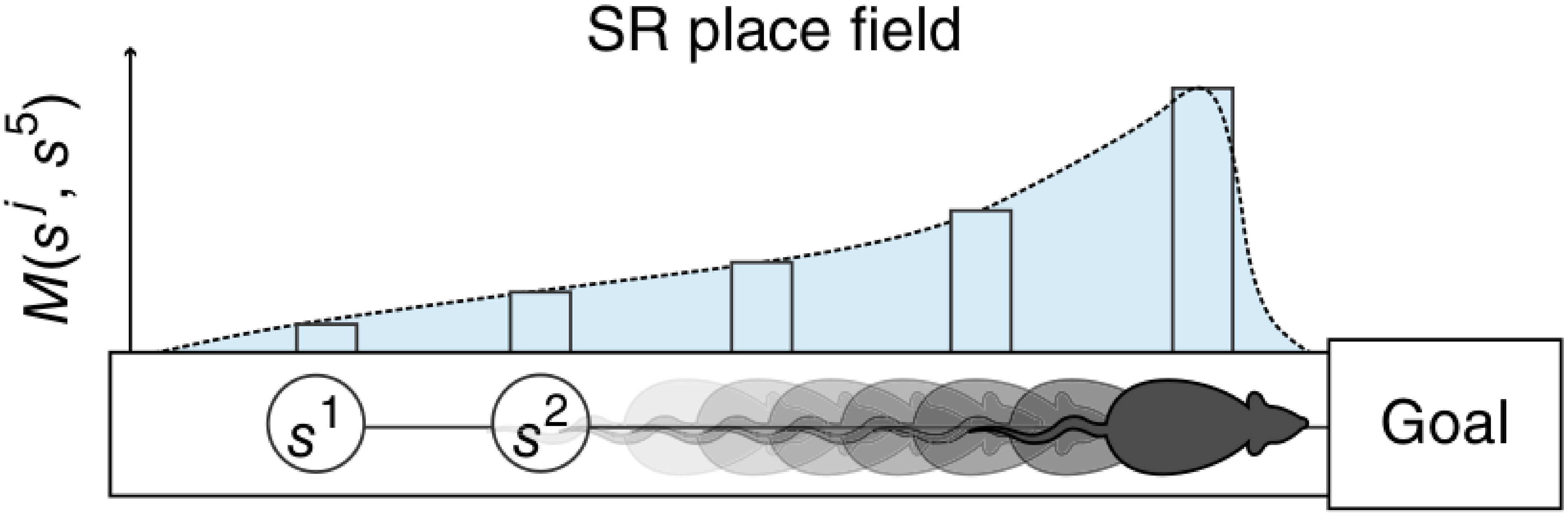


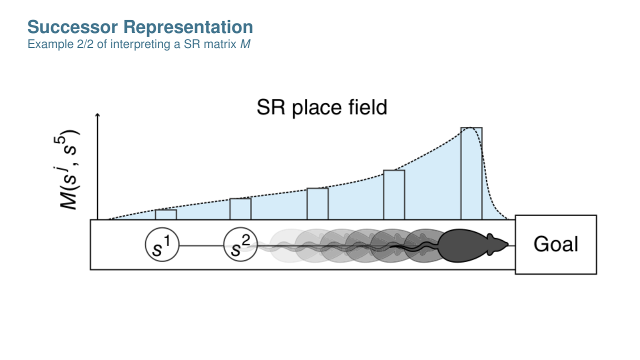
Superscript means index not power

- 6 states
- Policy: going to right or staying in current location
- $M(s^1, s')$ describes row of M and implies the following: Due to the “stay or go to the right”-policy is the probability for the states s^1, s^2 the largest

Successor Representation

Example 2/2 of interpreting a SR matrix *M*





– You can recognize the development of the policy: The more often the rat/mice goes to the right, the more likely it becomes reaching state s^4 or s^5

Goal of the thesis

Goal of the thesis

We try to answer the question: *Does the Hippocampus, i.e. place (and grid) cells, generate a cognitive map of languages?*

Approach(es): Training a (shallow, supervised) neural network by...

1. building a graphical model based on word classes
   - Word classes with made up rules like “Pronoun $\rightarrow$ Verb”
   - Building a **cognitive room**: List of all words
   - For each rule two 1 hot encoded vectors related to the cognitive room are used for training (first vector as input, second as preferable output)

   $\Rightarrow$ Works well due to graph environment

Goal of the thesis

We try to answer the question: *Does the Hippocampus, i.e. place (and grid) cells, generate a cognitive map of languages?*

Approach(es): Training a (shallow, supervised) neural network by...

1. building a graphical model based on word classes
   - Word classes with made up rules like “Pronoun → Verb”
   - Building a **cognitive room**: List of all words
   - For each rule two 1 hot encoded vectors related to the cognitive room are used for training (first vector as input, second as preferable output)
   - ⇒ Works well due to graph environment

– **training data**: first word is input, second word is supervised output. So, the network should learn to predict the second one based on the primary word

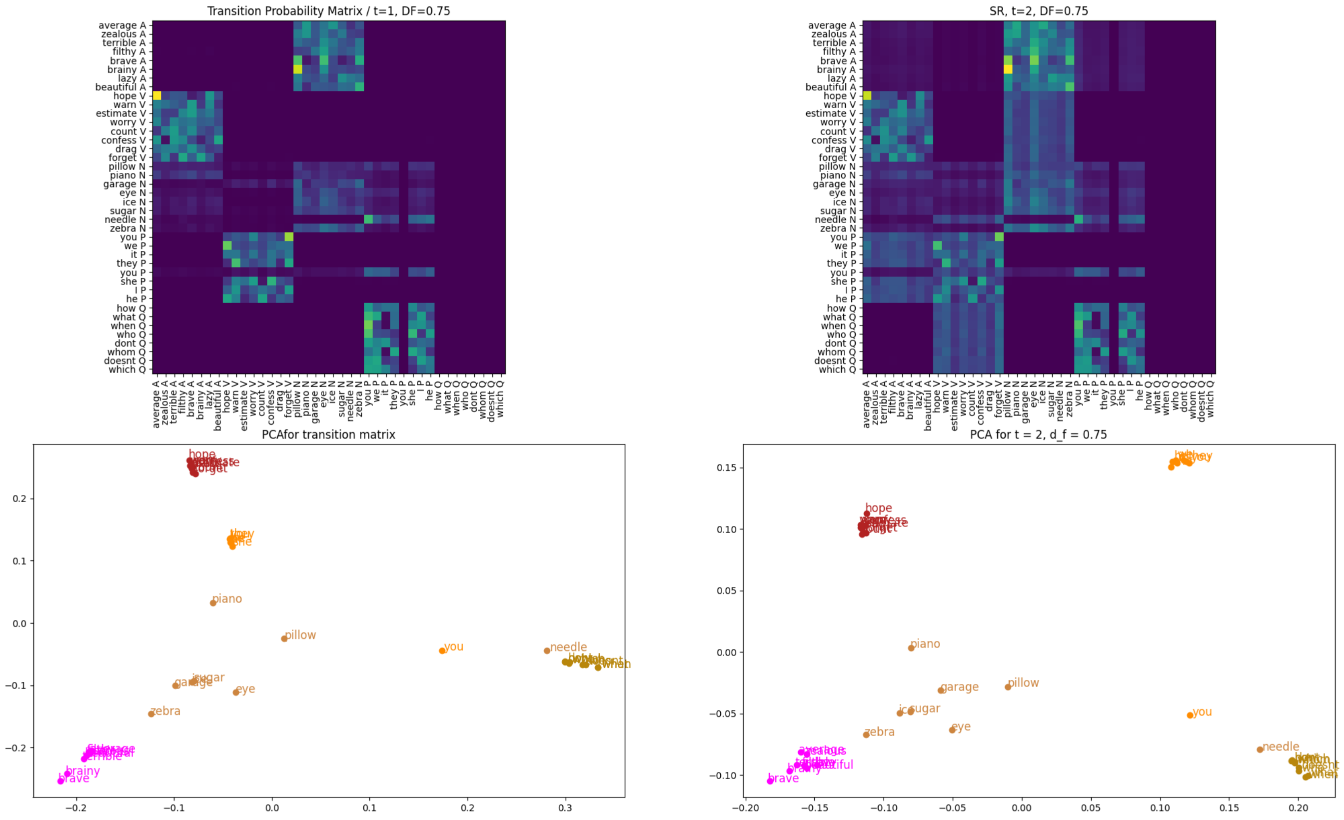
– **1-hot-encoded vector**: Imagine having a list of all words, then we generate a 0-vector with equal length and note a 1 at the position of the word in the list.

– **word vectors**: For predicting/evaluation we invert the calculating of a word vector, which means receiving a word

Example results for graphical model

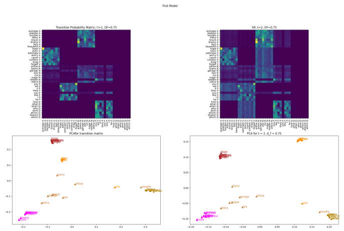
Rules used: Question Word → Personal Pronoun, Adjective → Noun, Personal Pronoun → Verb & Verb → Adjective

First Model



Example results for graphical model

Rules used: Question Word → Personal Pronoun, Adjective → Noun, Personal Pronoun → Verb & Verb → Adjective



- “noun”-row smeary because no rule with first argument “noun” exists
- each color represents a word class
- clustering obvious $\Longrightarrow$ Word classes recognizable

Goal of the thesis

We try to answer the question: *Does the Hippocampus, i.e. place (and grid) cells, generate a cognitive map of languages?*

Approach(es): Training a (shallow, supervised) neural network by...

1. building a graphical model based on word classes
   - Word classes with made up rules like "Pronoun $\rightarrow$ Verb"
   - Building a **cognitive room**: List of all words
   - For each rule two **1 hot encoded vectors** related to the cognitive room are used for training (first vector as input, second as preferable output)

   $\Rightarrow$ Works well due to graph environment
2. Parsing text to use two following words as training data by...
   - Using 1 hot encoded vectors but without made up rules. Input & output are determined by running order
   - Calculating word vectors using `spacy`
     - `spacy` can (bidirectionally) transfer between words and vectors

   $\Rightarrow$ Works ...not really

Goal of the thesis

We try to answer the question: *Does the Hippocampus, i.e. place (and grid) cells, generate a cognitive map of languages?*

Approach(es): Training a (shallow, supervised) neural network by...

1. building a graphical model based on word classes
   - Word classes with made up rules like “Pronoun → Verb”
   - Building a **cognitive room**: List of all words
   - For each rule two **1 hot encoded vectors** related to the cognitive room are used for training (first vector as input, second as preferable output)

   ⇒ Works well due to graph environment
2. Parsing text to use two following words as training data by...
   - Using 1 hot encoded vectors but without made up rules. Input & output are determined by running order
   - Calculating word vectors using `spacy`
     - `spacy` can (bidirectionally) transfer between words and vectors

   ⇒ Works ...not really

– **training data**: first word is input, second word is supervised output. So, the network should learn to predict the second one based on the primary word

– **1-hot-encoded vector**: Imagine having a list of all words, then we generate a 0-vector with equal length and note a 1 at the position of the word in the list.

– **word vectors**: For predicting/evaluation we invert the calculating of a word vector, which means receiving a word

Example of word vector model