From sequences to schemas: How recurrent neural networks learn temporal abstractions

Abstract

The world, despite its complexity, harbors patterns and regularities crucial for animals, with numerous real-life processes evolving over time into structured sequences of events. Brains have evolved to learn and exploit these sequential regularities, by forming knowledge at different degrees of abstraction: from simple transition and timing, to chunking, ordinal knowledge, algebraic patterns, and finally nested tree structures. How regularities expressed in algebraic patterns or abstract schemas (e.g., AAB or ABA) are encoded in the brain is still an open question. Here, we study whether and how neural circuits acquire, organize and use such an abstract code. We first build a computational framework to generate sequences with abstract temporal patterns. Next, we propose Recurrent Neural Networks (RNN) models performing different tasks requiring learning and predicting such sequences, and study the conditions under which learning is possible. We study the internal representations formed by the network models, and the extent to which these representations might be abstract, allowing to generalize to novel sequences and tasks.

Keywords: abstract sequential pattern; generalisation; abstraction; classification; prediction; transfer learning; RNN

The ability to generalize from specific experiences is fundamental to how we understand and interact with the world. Yet, the precise brain mechanisms that support this ability remain largely elusive (Murphy, Mondragón, & Murphy, 2008; Santolin & Saffran, 2018). A key aspect of this process involves learning temporal rules–such as algebraic patterns–from sequences of events. These patterns, like “AAB” or “ABA”, represent abstract relational structures between elements, independent of the specific stimuli used (e.g., any tokens of sounds, colors, shapes, etc.) (Marcus, 2003). Unlike simple repetition or alternation, these patterns require recognizing underlying rules that govern sequence structure. How do neural circuits learn and represent such abstract temporal schemas (Dehaene, Meyniel, Wacongne, Wang, & Pallier, 2015)? What computational principles enable the brain to detect, process and exploit temporal regularities?

Recurrent Neural Networks (RNNs), which are well-suited for modeling time-evolving processes such as working memory and decision making (Machens, Romo, & Brody, 2005), have emerged as powerful tools in neuroscience for linking activity to behavior (Barak, 2017; Yang & Molano-Mazón, 2021). In this study, we use RNNs to investigate how abstract temporal structures can be learned and represented.

We first systematically generate algebraic sequences (that may be of various lengths, but here of length $L = 6$) using a binary branching tree, where terminal nodes define abstract sequence classes. Each distinct letter is drawn from an alphabet of a given size (Fig. 1A). We then train RNNs to perform various tasks on these sequences: classification (Fig. 1B), next-item prediction (Fig. 3A) or reconstruction (Fig. 3B) of such sequences via an autoencoder. We then analyze the computational and representational properties that support abstraction across tasks.

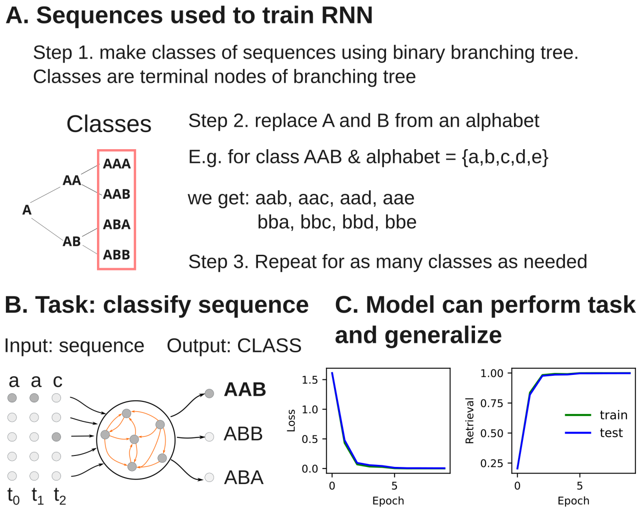
A. Sequences used to train RNN
Step 1. make classes of sequences using binary branching tree.
Classes are terminal nodes of branching tree
Classes
A
AA
AB
AAA
AAB
ABA
ABB
Step 2. replace A and B from an alphabet
E.g. for class AAB & alphabet = {a,b,c,d,e}
we get: aab, aac, aad, aae
bba, bbc, bbd, bbe
Step 3. Repeat for as many classes as needed
B. Task: classify sequence
Input: sequence
Output: CLASS
a a c
t0 t1 t2
AAB
ABB
ABA
C. Model can perform task and generalize
Loss
1.5
1.0
0.5
0.0
0
5
Epoch
Retrieval
1.00
0.75
0.50
0.25
0
5
Epoch
train
test

Figure 1: Learning temporal regularities via sequence classification. **(A)** Sequences used to train all models are generated via a binary branching tree. Classes correspond to the terminal nodes of this tree. To generate sequences of a given class, we replace A and B from an alphabet of a given size. **(B)** We train a recurrent neural network (RNN) to classify sequences into one of multiple classes based on the underlying temporal pattern (for clarity sequences of length $L = 3$ are shown but all figures use $L = 6$). **(C)** The model can perform the task and generalize the abstract classes to unseen sequences.

Our results show that a discrete-time RNN trained on the classification task (e.g., aac and bba to AAB; aca and ada to ABA, Fig. 1B) learns to generalize well to unseen sequences (Fig. 1C). Principal Component Analysis (PCA) of the hidden representations reveals low-dimensional, linearly separable representations that cluster according to abstract sequence classes (Fig. 2A). These clusters are organized such that transitions between items (e.g. same item AA vs. different item AB) are geometrically separable. Further analysis of hidden unit selectivity, as in (Yang et al., 2019), shows neurons tuned to abstract class identity (Fig. 2B, left) but not specific items (Fig. 2B, right). Ablating class-selective neurons during testing impairs classification performance specifically for the corre-

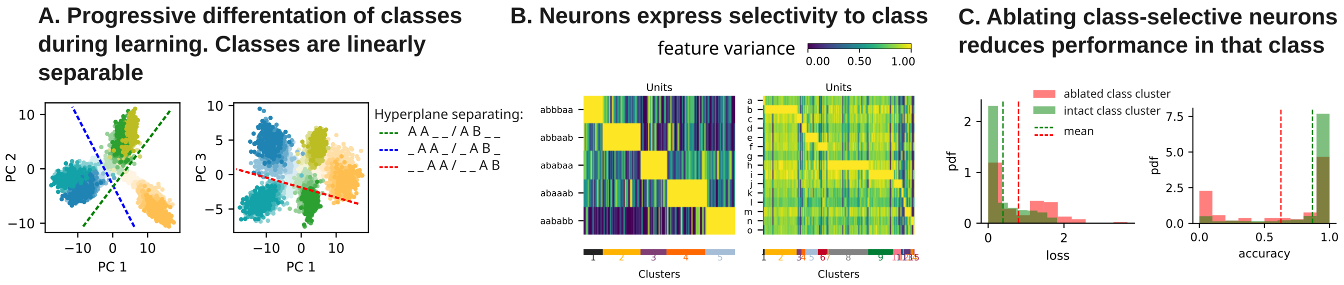

Figure 2: Learning temporal regularities via sequence classification. **(A)** PCA applied to representations reveals progressive differentiation of sequence classes during learning. At the end of learning, hyperplanes associated with each transition in the sequence separate classes according to whether a letter is repeated or changed. **(B)** We compute the selectivity of a neuron at the end of the sequence to a given feature by computing its normalized variance (Yang et al., 2019). **(C)** Ablating clusters of units selective for a class during testing reduces the performance on that class of sequences.

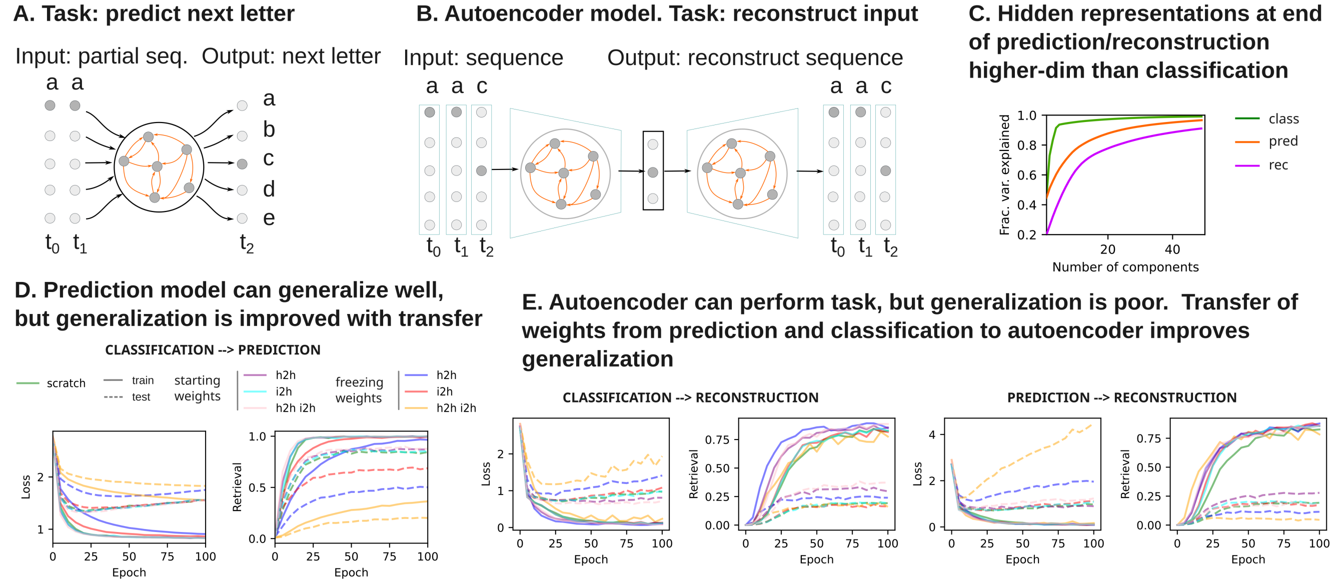

Figure 3: Learning temporal regularities via next-token prediction and reconstruction. **(A)** We train an RNN model to predict the next letter in the sequence after receiving a partial sequence as a cue. The network can learn the training set as well as generalize. **(B)** We train an autoencoder model to reconstruct the same sequences as in Fig. 1A. **(C)** PCA analysis on the hidden activity at the end of the sequence reveals higher-dimensional representations for the prediction and reconstruction models. **(D)** The prediction model's ability to generalize can slightly improve when transferring weights (input, recurrent or both) from the classification task. **(E)** The autoencoder model's ability to generalize is good for sequences of $L = 3$ (not shown) but poor for $L = 6$ (shown). However, setting the initial values of the (input, recurrent or both) weights from a better performing model (prediction - left and classification - right) improves generalization performance.

sponding class, confirming their role in representing abstract temporal structure (Fig. 2C).

Next, we study two other networks: an RNN tasked with predicting the next item in the sequence (Fig. 3A), and an auto-encoder tasked with reconstructing the input sequence (Fig. 3B). While both achieve good performance, only the prediction network generalizes well (Fig. 3D and E). PCA of hidden states shows that both prediction and reconstruction networks use higher-dimensional representations relative to the classifier (Fig. 3C).

Given that all of these models must exploit the temporal patterns present in the input to perform the tasks, we next apply transfer learning to assess whether these networks share representational structure: we initialize or freeze weights from one task to train on another. Transfer from classification to prediction yields marginal gains in generalization, while transfer (from both classification or prediction tasks) to the reconstruction task improves performance (Fig. 3D and E).

Our findings demonstrate how task constraints shape the emergence of abstract sequence representations in RNNs and provide insight into computational principles that might underlie abstraction in artificial and biological neural systems.

References

Barak, O. (2017). Recurrent neural networks as versatile tools of neuroscience research. *Curr. Opinion in Neurobio.*, *46*, 1–6.

Dehaene, S., Meyniel, F., Wacongne, C., Wang, L., & Pallier, C. (2015). The neural representation of sequences: from transition probabilities to algebraic patterns and linguistic trees. *Neuron*, *88*(1), 2–19.

Machens, C. K., Romo, R., & Brody, C. D. (2005). Flexible control of mutual inhibition: a neural model of two-interval discrimination. *Science*, *307*(5712), 1121–1124.

Marcus, G. F. (2003). *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

Murphy, R. A., Mondragón, E., & Murphy, V. A. (2008). Rule learning by rats. *Science*, *319*(5871), 1849–1851.

Santolin, C., & Saffran, J. R. (2018). Constraints on statistical learning across species. *Trends in Cog. Sciences*, *22*(1), 52–63.

Yang, R., Joglekar, M. R., Song, H. F., Newsome, W. T., & Wang, X.-J. (2019). Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, *22*(2), 297–306.

Yang, R., & Molano-Mazón, M. (2021). Towards the next generation of recurrent network models for cognitive neuroscience. *Curr. Opinion in Neurobio.*, *70*, 182–192.

From sequences to schemas: How recurrent neural networks learn temporal abstractions

Author Information

Vezha Boboeva (v.boboeva@ucl.ac.uk)[1],

Alberto Pezzotta (a.pezzotta@ucl.ac.uk)[2],

George Dimitriadis (g.dimitriadis@ucl.ac.uk)[1,2],

Athena Akrami (athena.akrami@ucl.ac.uk)[1],

[1]Sainsbury Wellcome Centre
25 Howland St, London W1T 4JG

[2]Gatsby Computational Neuroscience Unit
25 Howland St, London W1T 4JG