Notes on two very different papers studying grokking mechanism

I came across two ICLR/ICML papers studying grokking [1] and a very similar phenomenon to grokking [2], characterized by a sudden decrease in generalization error when neural networks are extensively trained on a simple algebraic task. It is notable that the former introduces an empirical method called mechanistic interpretability, investigating what exactly the learned neural network calculates, while the latter views learning as a (linearized) dynamical system to deduce conditions at which its hidden states merge to result in the sudden generalization. I found the distinct, empirical and theoretical approaches interesting and decided to study their details.

Progress Measures for Grokking via Mechanistic Interpretability (2023)

Set Up

The paper studied a 1-lyr transformer trained on a modular addition task, a setting identical to the work that first reported grokking. [3]

- task (modular addition): given $a, b \in \{0, \ldots, P-1\}$ where P is prime, predict $c \equiv a + b \pmod{P}$. a, b are encoded as P-dimensional one-hot vectors, $=$ as a special token
- training: train on 30% of all $P \cdot P$ pairs, test on all pairs not used for training
- model: 1-lyr Transformer with ReLU MLP without layernorm, tied embed/unembed, negligible skip connection around MLP. For notations, embedding matrix $W_E \in \mathbb{R}^{d \times P}$, MLP output map $W_{\text{out}} \in \mathbb{R}^{d \times n}$, unembedding $W_U \in \mathbb{R}^{P \times d}$, neuron-logit map $W_L := W_U W_{\text{out}}$; $\text{Logits}(a, b) \approx W_U W_{\text{out}} \, \text{MLP}(a, b) = W_L \, \text{MLP}(a, b)$
- observations: training loss decreases to ~ 0 while validation loss stays at > 0 → training loss remains the same but validation loss suddenly drops to ~ 0 indicating perfect generalization

Claim

"Transformer uses Fourier transforms and trig identities to convert addition to rotation about a circle". This claim is rather natural because on a cyclic group, addition is phase multiplication which corresponds to the trig rotation identities in real coordinates.

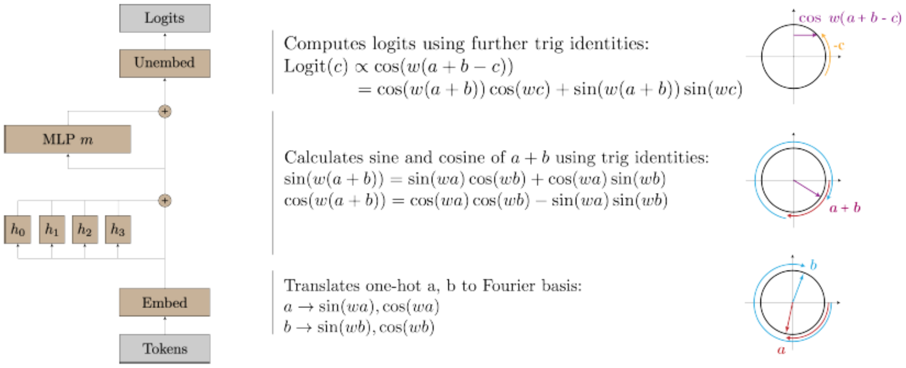

Logits
Unembed
MLP m
h_0
h_1
h_2
h_3
Embed
Tokens
Computes logits using further trig identities:
$\text{Logit}(c) \propto \cos(w(a+b-c))$
$= \cos(w(a+b))\cos(wc) + \sin(w(a+b))\sin(wc)$
$\cos\ w(a+b-c)$
$-c$
Calculates sine and cosine of $a+b$ using trig identities:
$\sin(w(a+b)) = \sin(wa)\cos(wb) + \cos(wa)\sin(wb)$
$\cos(w(a+b)) = \cos(wa)\cos(wb) - \sin(wa)\sin(wb)$
$a+b$
Translates one-hot a, b to Fourier basis:
$a \to \sin(wa), \cos(wa)$
$b \to \sin(wb), \cos(wb)$
b
a

Figure 1: The algorithm implemented by the one-layer transformer for modular addition. Given two numbers a and b, the model projects each point onto a corresponding rotation using its embedding matrix. Using its attention and MLP layers, it then composes the rotations to get a representation of $a + b \mod P$. Finally, it "reads off" the logits for each $c \in \{0, 1, \ldots, P-1\}$, by rotating by $-c$ to get $\cos(w(a+b-c))$, which is maximized when $a + b \equiv c \mod P$ (since w is a multiple of $\frac{2\pi}{P}$).

Evidence

The paper provided suggestive, mechanistic and causal (via ablation) evidence at the level of weights and activations. After reading more mechanistic interpretability work, I realized this kind of thoroughness is standard in the community, hence the term mechanistic.

- **suggestive: periodicity in weights, activations and logits**
  - took DFT along the input dimension of W_E, looked at ℓ_2 norms of Fourier components, observed sparsity and identified 5 key frequencies
  - similar periodic structure in attention, activations and logits, with logits showing $\sim$ 20 significant components in 2D Fourier basis over (a, b), aligning with products of sin and cos at key frequencies
- **mechanistic: low-rank decomposition of neuron-logit map W_L.**
  - showed
  $$W_L \approx \sum_{k \in \{14,35,41,42,52\}} \cos(w_k)\, u_k^\top + \sin(w_k)\, v_k^\top,$$
  where $\cos(w_k), \sin(w_k) \in \mathbb{R}^P$ are vectors over c with entries $\cos(w_k c)$, $\sin(w_k c)$, and $u_k, v_k \in \mathbb{R}^n$
  - projected MLP(a, b) onto u_k, v_k and observed that the resulting scalars are approximately multiples of $\cos(w_k(a + b))$ and $\sin(w_k(a + b))$ with high fraction variance explained
- **zooming in: neuron-level approximation with sin and cos**
  - fit a degree-2 polynomial in $\sin(w_k a), \cos(w_k a), \sin(w_k b), \cos(w_k b)$ for a single key frequency to observe most MLP neurons have high FVE with just one frequency *polynomial neurons
  - for neuron groups assigned to frequency k, the corresponding columns of W_L had only $\sin(w_k c)$ and $\cos(w_k c)$ components
- **correctness checks: ablations**
  - replaced the activations of the polynomial neurons by their fitted polynomial predictions → negligible loss change
  - restricted each neuron activation to the polynomial components corresponding to $\cos(w_k(a + b))$ and $\sin(w_k(a + b))$ at key frequencies → loss improvement
  - computed a 2D DFT of the full logit tensor over (a, b) to ablate selected frequencies
    - Ablating key frequency increased loss, ablating other frequencies didn't
    - Ablating all non-key Fourier components improved loss

Progress measures

The paper calculated and observed the ablation-based opposite metrics during training to segment it into phases: memorization → circuit formation → cleanup, suggesting the use of progress measures for observing circuit formation and verifying that grokking is indeed progressive with circuit forming gradually. (it was the cleanup that made it look sudden by manifesting the circuit at a certain point)

- restricted loss: take 2D DFT of logits over inputs (a, b) and keep only constant term + the 20 terms corresponding to $\cos(w_k(a + b))$ and $\sin(w_k(a + b))$ for the five key frequencies and measure loss
- excluded loss: ablate only the key-frequency components and measure loss

Algorithm Development in Neural Networks (2025)

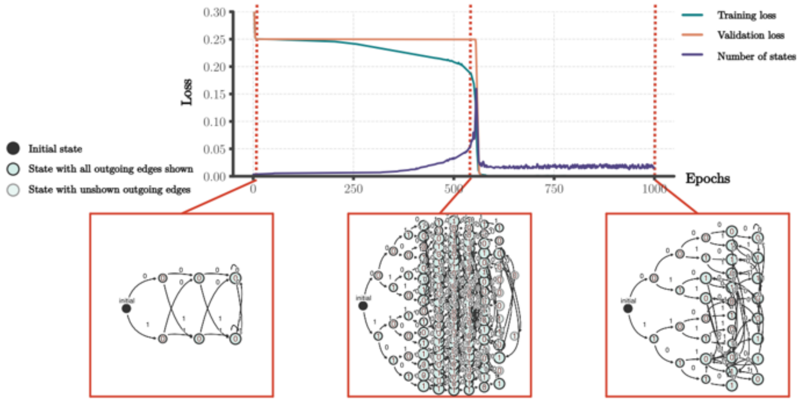
Set Up

The paper studied a 1-lyr RNN trained on a streaming parity task. Note the recurring algebraic theme as in the modular addition task.

- task (streaming parity): given streaming inputs, output 0 if the number of 1s seen so far is even, 1 otherwise
- model: 1-lyr RNN then MLP
- training and testing: train on all sequences up to length 10 and test on sequences of length larger than 10
- observation: constant training and validation loss → sudden drop in both, indicating perfect generalization

Observation

The paper introduced the deterministic finite automata (DFA) view of the neural network's hidden state evolution and observed that RNN (i) expands automata into a complete tree fitting the training data, followed by (ii) the merger of states in the automaton inducing the observed training and validation loss drop and perfect generalization to arbitrarily long sequences.



Initially few states and
random outputs
(due to random small weights)

Phase I Automaton expands into
a complete tree
fitting the training data

Phase II States in the automaton
merge until it becomes finite
(this is when training loss = 0,
generalization on all data happens)

Explanation

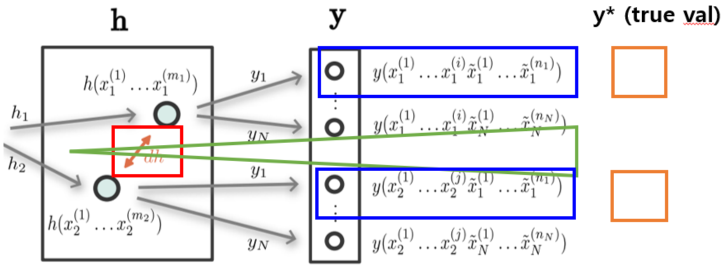
In order to understand the mechanism behind such state merger, the paper introduced Interaction theory, a theory on behaviour of representations as hidden representations of two input sequences get near during training in a highly expressible arbitrary neural networks (where smooth continuous mapping is feasible due to the universal approximation theorem). This gives insights into how the neural network output y is computed from the interaction between the hidden states. I found it interesting how the paper shifts between continuous (RNN) to discrete (DFA) and back to continuous (dynamics) concepts; framing the hidden states as interacting objects in a dynamical system also felt like a novel perspective. Denote x as input, h as hidden state, y as NN output and y* as true value, with suitable subscripts.

- expand y_i around the midpoint $\bar{h} = \frac{h_1+h_2}{2}$ to deduce linear approximation of y

$$y_i(x_{\alpha,i}) \approx \bar{y}_i + \frac{1}{2}\,Dy_i(\bar{h})\,(h_\alpha - h_{\neg\alpha}), \quad \bar{y}_i := \frac{y_i(x_{1,i}) + y_i(x_{2,i})}{2}, \ \alpha \in \{1,2\}. \tag{1}$$

- note that the task is in a supervised learning setting, meaning the MSE loss between y (now locally linearly approximated) and y* is forced to decrease

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\sum_{\alpha\in\{1,2\}} \left\| y_i(x_{\alpha,i}) - y^*_{\alpha,i} \right\|^2. \tag{2}$$


h
y
y* (true val)
$h(x_1^{(1)} \dots x_1^{(m_1)})$
h_1
h_2
dh
$h(x_2^{(1)} \dots x_2^{(m_2)})$
y_1
y_N
y_1
y_N
$y(x_1^{(1)} \dots x_1^{(i)} \tilde{x}_1^{(1)} \dots \tilde{x}_1^{(n_1)})$
$y(x_1^{(1)} \dots x_1^{(i)} \tilde{x}_N^{(1)} \dots \tilde{x}_N^{(n_N)})$
$y(x_2^{(1)} \dots x_2^{(j)} \tilde{x}_1^{(1)} \dots \tilde{x}_1^{(n_1)})$
$y(x_2^{(1)} \dots x_2^{(j)} \tilde{x}_N^{(1)} \dots \tilde{x}_N^{(n_N)})$

- applying gradient descent to the above loss function as well as Anzats representational movement and a bit of algebra leads to a self-contained 3-scalar system. Note the representational difference $d_h := h_2 - h_1$, the output differences $dy_i := Dy_i(\bar{h})\, d_h$ and the alignment between output and prediction $w_i = \|dy_i|^2 - dy_i^T(y^*_{2,i} - y^*_{1,i})$ are the variables of interest

$$\dot{d}_h = -\frac{1}{2}\sum_{i=1}^{N} w_i\, d_h, \quad \dot{dy}_i = -w_i\, dy_i, \quad \dot{w}_i = -w_i^2 + \frac{1}{2}\big(\tilde{y}_i dy_i\big)^2, \quad \tilde{y}_i := y^*_{2,i} - y^*_{1,i}. \tag{3}$$

- the system can be further reduced to 2D in $(\|d_h\|, w_i))$ with three fixed points only one of which is stable, being a closed-form expression for the final representational distance $\|d_h(\infty)\|^2$.

$$A_{\text{low}} := \left\langle \|\tilde{y}_i\|^2 \right\rangle_i - \|\langle \tilde{y}_i \rangle_i\|^2 \tag{4}$$

$$A_{\text{high}} := \|\langle \tilde{y}_i \rangle_i\|^2 - \left\langle \frac{\|y_i(x_{2,i}) - y_i(x_{1,i})\|^2}{\|d_h\|^2} \right\rangle_i^{-1} \tag{5}$$

  Then

$$\|d_h(\infty)\|^2 = \frac{1}{2}\left(A_{\text{high}} + \sqrt{A_{\text{high}}^2 + 4A_{\text{low}}}\right). \tag{6}$$

- recall we are interested in the case where representations merge, i.e. $\|d_h(\infty)\|^2 = 0$, and this occurs when

$$A_{\text{low}} = 0 \quad \text{and} \quad A_{\text{high}} < 0. \tag{7}$$

- $A_{\text{low}} = 0$ iff $y^*_{1,i} = y^*_{2,i}$ and $A_{\text{high}} < 0$ (loosely, under certain conditions) iff $C < N \cdot G^{\,n-m}$ where C denotes an unknown constant determined by the architecture, G denotes average decrease of representational distance when applying the recurrent map, m denotes the minimal sequential length for representations and n denotes that for potential subsequences. i.e. mergers are favored by larger training samples N, smaller initial scale $G < 1$ and longer sequential length m
- and these merger conditions were empirically verified

References

[1] Nanda, Neel, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. “Progress Measures for Grokking via Mechanistic Interpretability.” ICLR 2023

[2] van Rossem, Loek, and Andrew M. Saxe. “Algorithm Development in Neural Networks: Insights from the Streaming Parity Task.” ICML 2025

[3] Power, Alethea, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. “Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets.” ICLR 2021 Workshop

*GPT 5.2 for the write-up