Neural Networks as Automata

William Merrill

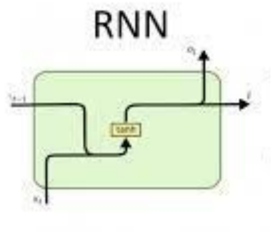
Jan 31, 2022

Motivation

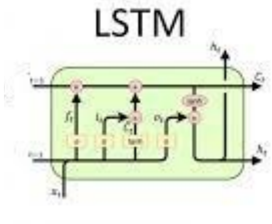
- In NLP, neural networks seem to learn a lot about the structure of language

- What kinds of patterns can they represent?
  - What kinds of formal languages do they recognize?

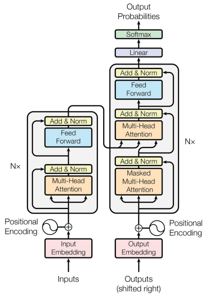
Contributions



≈ Finite automata



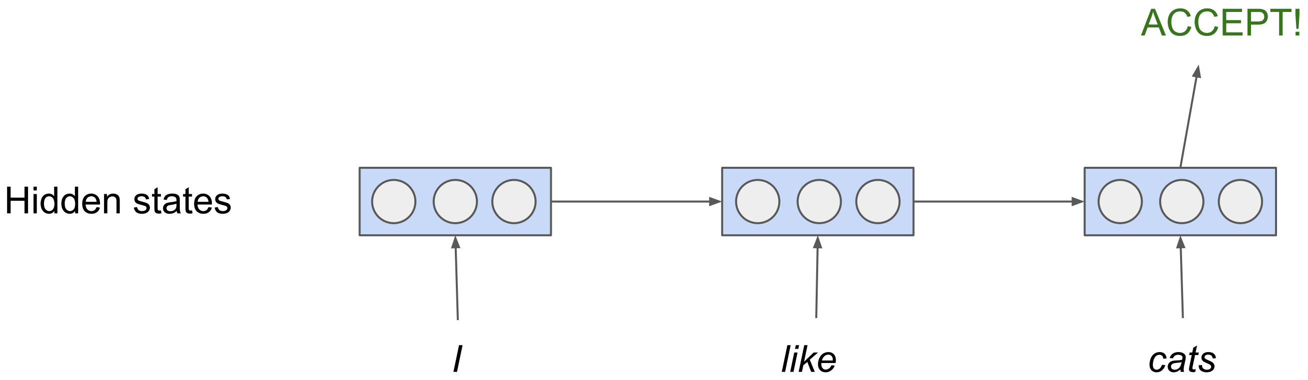
≈ Counter automata



≈ Threshold circuits

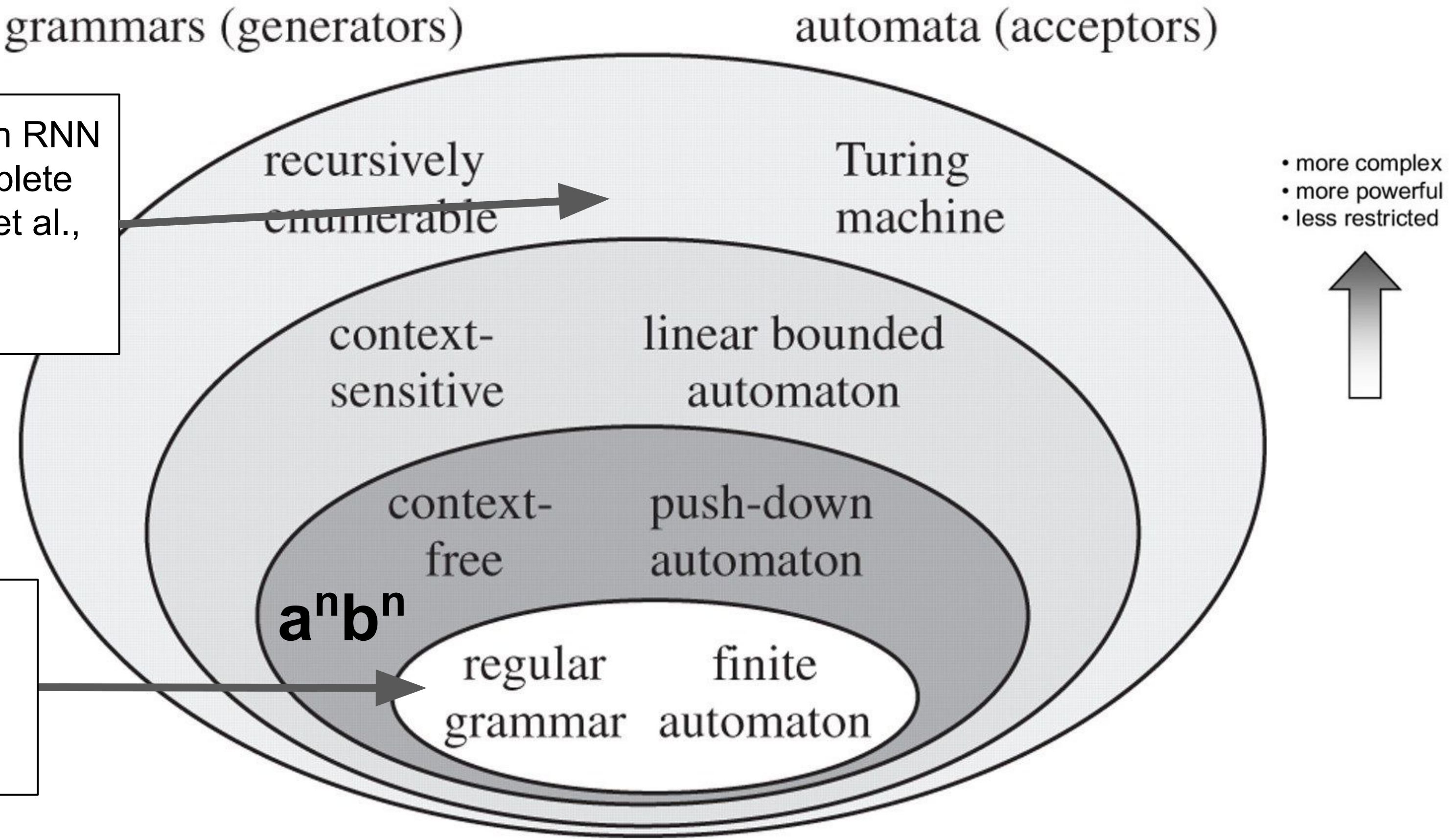
RNNs

Basic RNNs



$$h_t = \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{(t-1)} + b_{hh})$$

RNNs

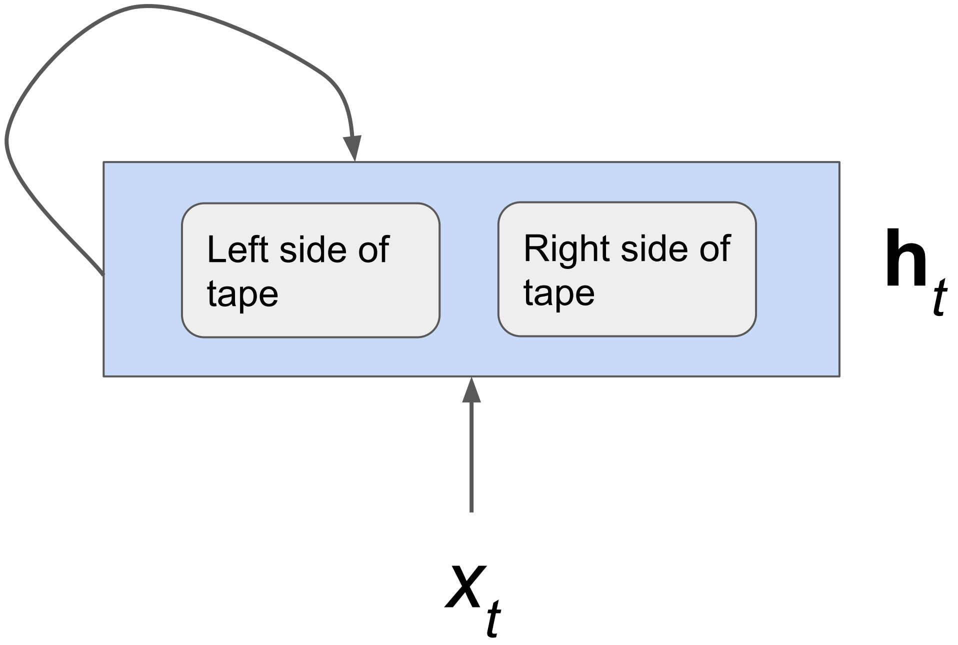


Ideal 2-neuron RNN is Turing complete (Siegelmann et al., 1991)

But… in practice, RNNs can't learn $a^n b^n$ (Weiss et al., 2018)

Proof Sketch: 2-Neuron RNN is Turing-complete

Use neurons to simulate 2-stack Turing machine construction



(Neurons also need to encode finite control)

Problems with “RNNs are Turing-complete”

1. Relies on **infinite precision**
2. Relies on **unbounded runtime**
3. Doesn’t take **training** into account

Definition: saturation operator

Analyze capacity of saturated network, not original network

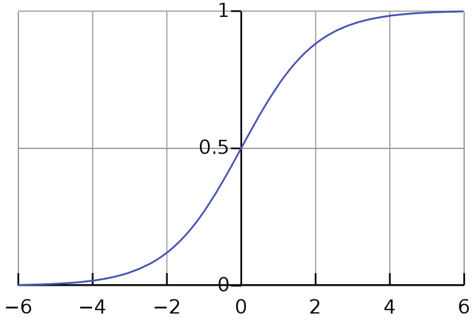
Definition 11 (Saturated network) Let $f(x;\theta)$ be a neural network parameterized by θ. We define the saturated network $\mathrm{s}f(x;\theta)$ as

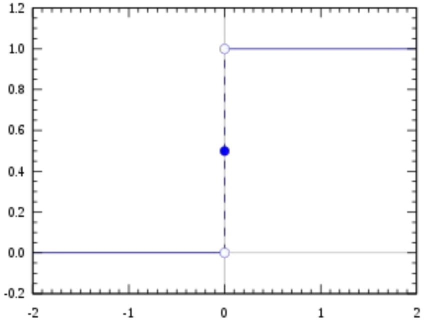
$$\mathrm{s}f(x;\theta) = \lim_{\rho\to\infty} f(x;\rho\theta).$$

[Sequential Neural Networks as Automata](#)
William Merrill, 2019
[Formal Language Theory Meets Modern NLP](#)
William Merrill, 2021

Saturation: function space view

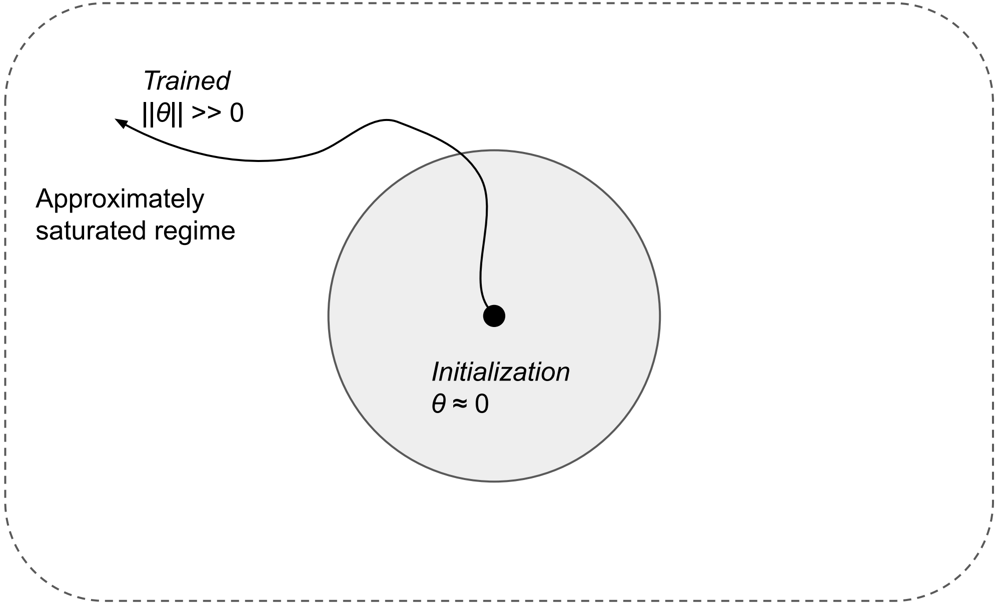
- Activation functions become step functions
- Precision of individual neurons is bounded



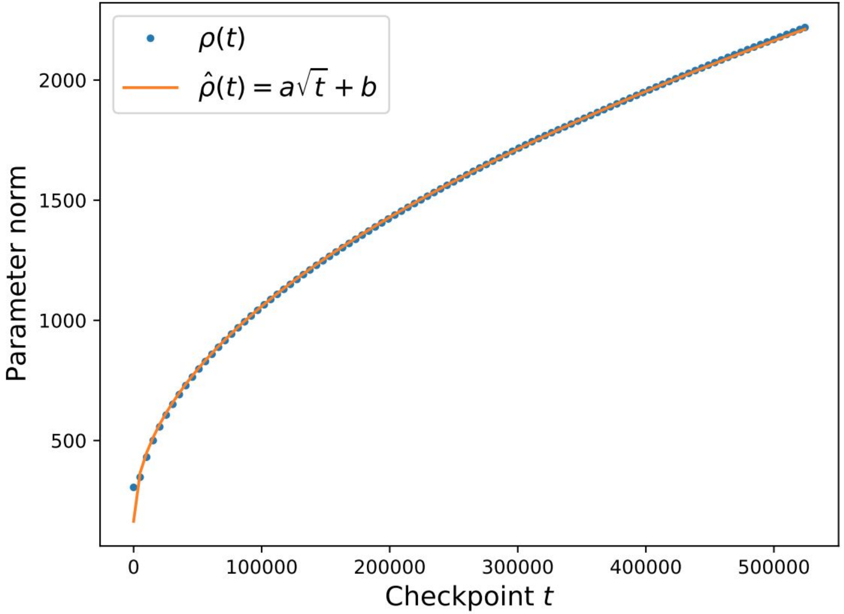


Saturation: parameter space view

Under minimal assumptions, highly trained networks are ~saturated



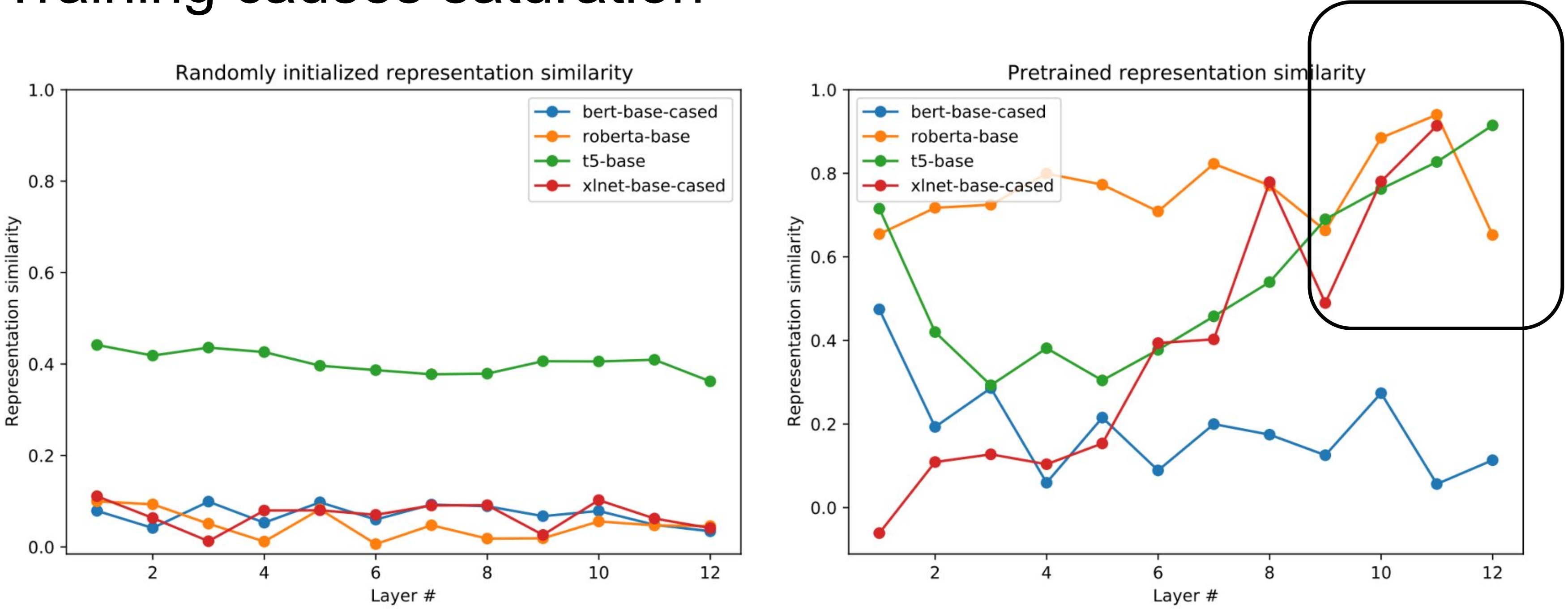
Norm growth in network training*



*Paper uses transformers, not RNNs

[Effects of Parameter Norm Growth During Transformer Training: Inductive Bias from Gradient Descent](#)
William Merrill, Vivek Ramanujan, Yoav Goldberg, Roy Schwartz, Noah A. Smith, 2021

Training causes saturation

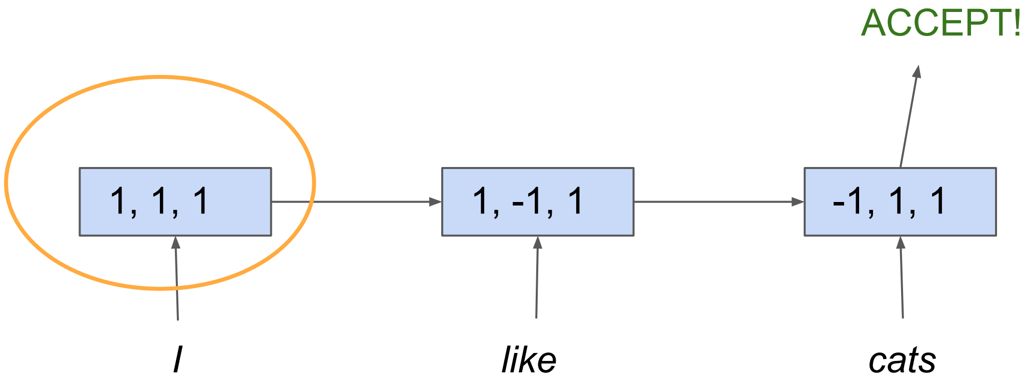


[Effects of Parameter Norm Growth During Transformer Training: Inductive Bias from Gradient Descent]()
William Merrill, Vivek Ramanujan, Yoav Goldberg, Roy Schwartz, Noah A. Smith, 2021

Saturated RNNs = finite automata

Saturated nonlinearity → step function

Saturated hidden state is fixed-length binary vector



[Sequential Neural Networks as Automata]

William Merrill, 2019

LSTMs

Generalized RNNs

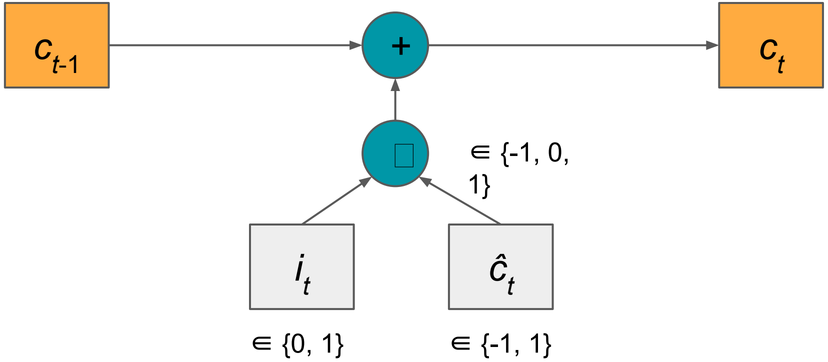
$h_t = \boldsymbol{f}(h_{t-1}, x_t)$

f = “gating function”

Saturated LSTMs ⊆ counter automata

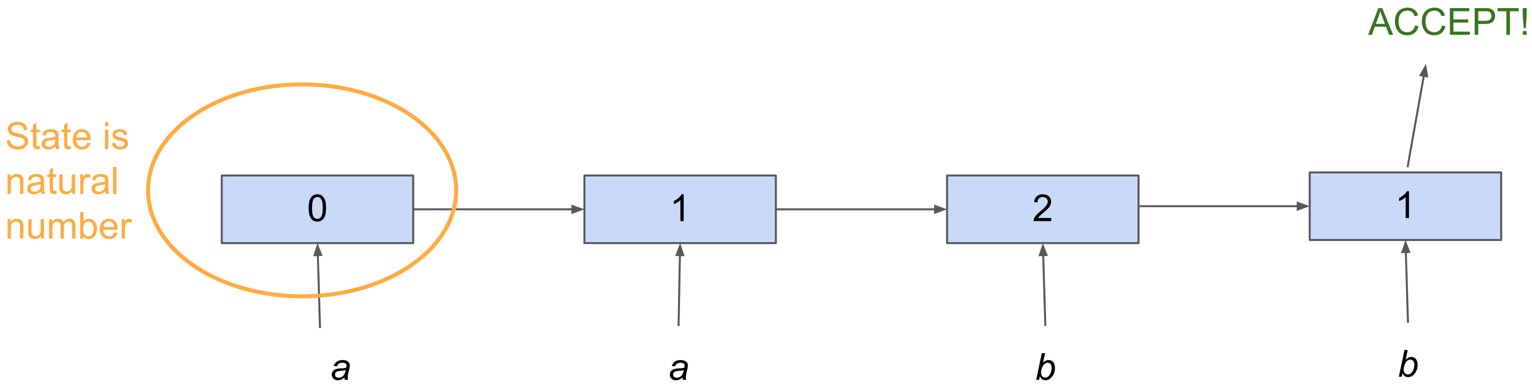
A (formerly popular) generalized RNN

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
&\boxed{c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t} \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$



In saturated LSTM, the memory can grow with the sequence length!

Example: LSTM Recognizing $a^n b^n$



Sequential Neural Networks as Automata
William Merrill, 2019

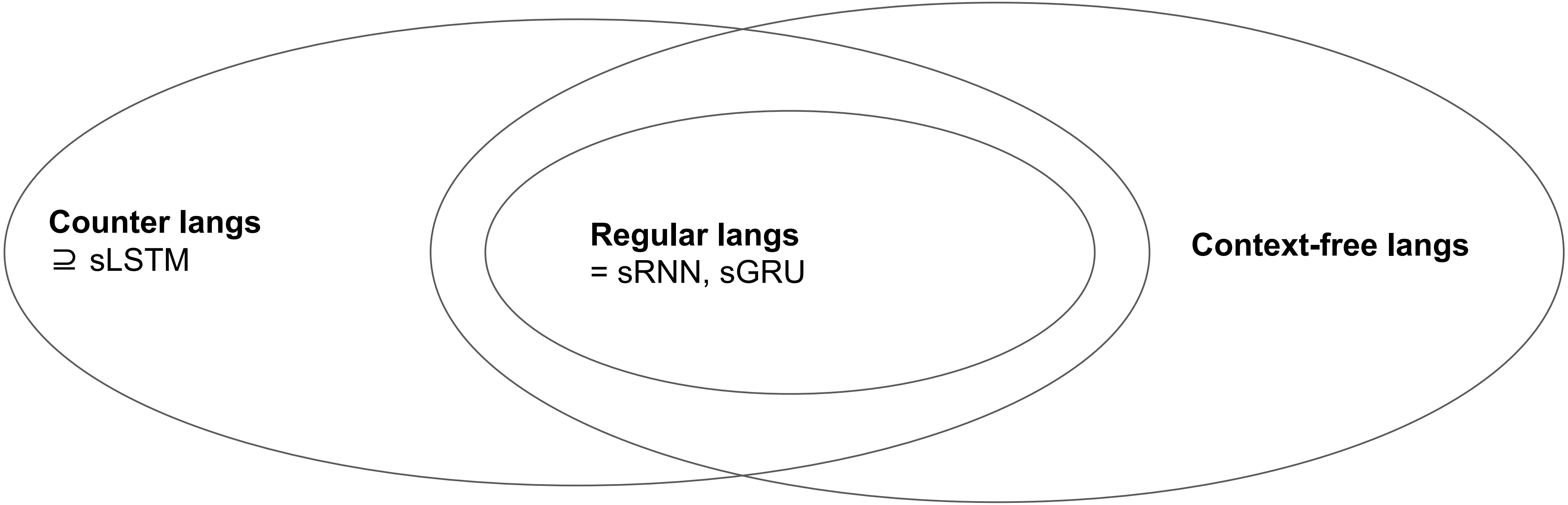
Gated recurrent units (GRUs)

Another, superficially similar generalized RNN

$$z = \sigma(W_z \cdot x_t + U_z \cdot h_{(t-1)} + b_z)$$
$$r = \sigma(W_r \cdot x_t + U_r \cdot h_{(t-1)} + b_r)$$
$$\tilde{h} = tanh(W_h \cdot x_t + r * U_h \cdot h_{(t-1)} + b_z)$$
$$\boxed{h = z * h_{(t-1)} + (1 - z) * \tilde{h}}$$

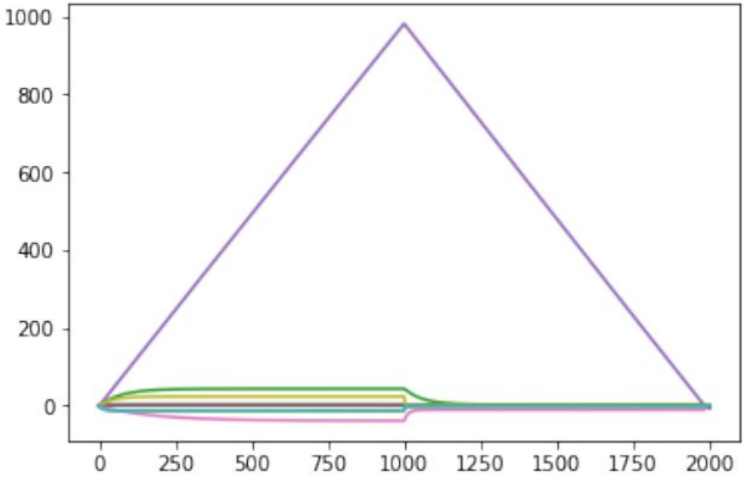
Saturated GRUs are still finite state, because of the different gating!

Hierarchy of saturated RNNs

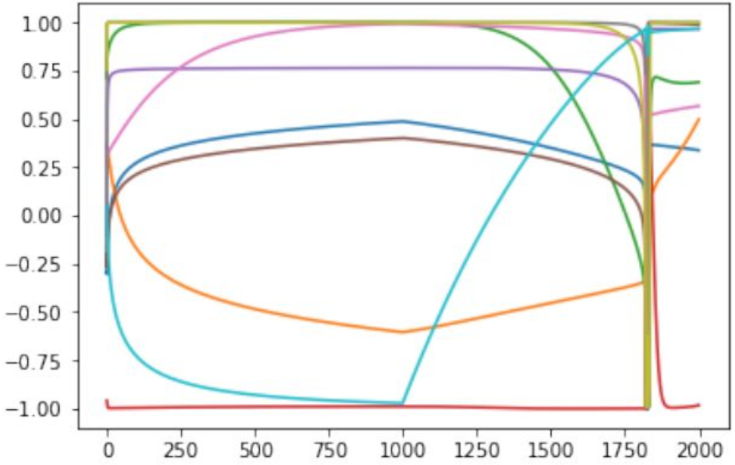


Sequential Neural Networks as Automata
William Merrill, 2019

Saturated nets predict empirical nets’ behaviors



(a) a^nb^n-LSTM on $a^{1000}b^{1000}$



(c) a^nb^n-GRU on $a^{1000}b^{1000}$

[On the Practical Computational Power of Finite Precision RNNs for Language Recognition]
Gail Weiss, Yoav Goldberg, Eran Yahav, 2018

LSTMs vs. QRNNs

QRNNs proposed as replacement for LSTMs

$$\mathbf{Z} = \tanh(\mathbf{W}_z * \mathbf{X})$$
$$\mathbf{F} = \sigma(\mathbf{W}_f * \mathbf{X})$$
$$\mathbf{O} = \sigma(\mathbf{W}_o * \mathbf{X}),$$

vs.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

- Much faster than LSTMs
- Known to be computable as weighted finite automaton (Peng et al., 2018)

But is QRNN as expressive as LSTM?

Quasi-Recurrent Neural Networks
James Bradbury, Stephen Merity, Caiming Xiong, Richard Socher, 2016
Rational Recurrences
Hao Peng, Roy Schwartz, Sam Thomson, Noah A. Smith, 2018

Result: saturated LSTMs are more powerful than QRNNs

Proof.

1. Saturated LSTMs can compute $$f_0 : x \mapsto \begin{cases} \#_{a-b}(x) & \text{if } \#_{a-b}(x) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

   aab → 1
   bba → 0

2. This function has Hankel matrix $$\begin{pmatrix} 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & \cdots \\ 2 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

3. Which has infinite rank ⇒ cannot be computed by WFA (and thus QRNN)

[A Formal Hierarchy of RNN Architectures](#)
William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, 2020

A different typology of RNNs

- Y axis: memory of hidden state
- X axis: is it WFA-computable?

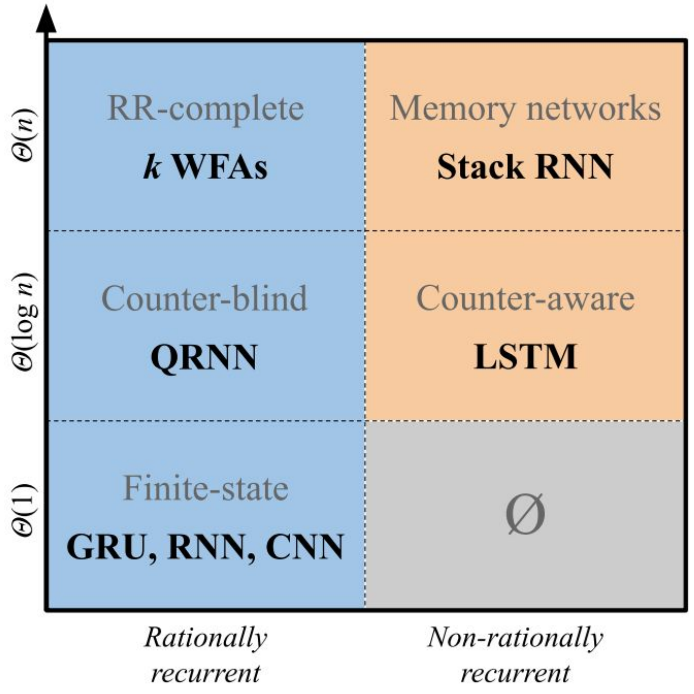
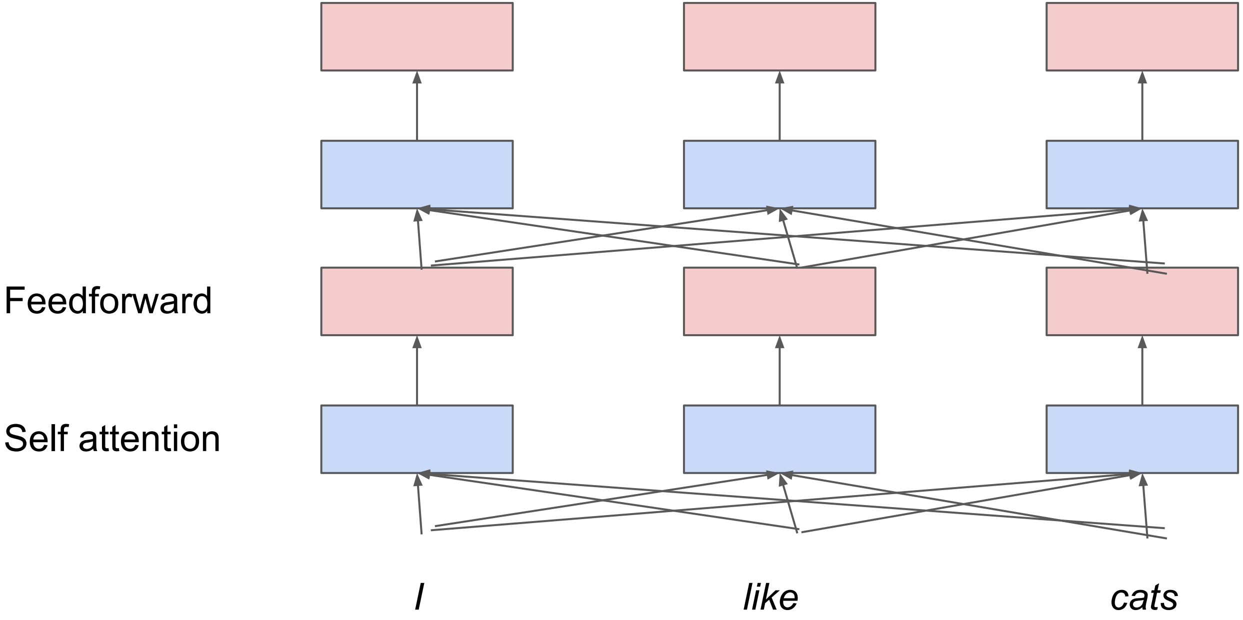


Figure 1: Hierarchy of state expressiveness for saturated RNNs and related models. The y axis represents increasing space complexity. $\emptyset$ means provably empty. Models are in bold with qualitative descriptions in gray.

[A Formal Hierarchy of RNN Architectures](#)
William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, 2020

Transformers

Transformers



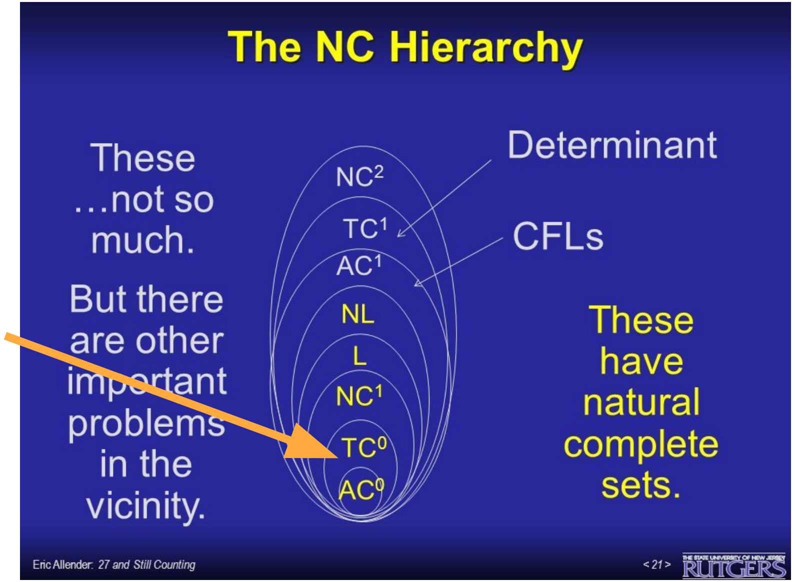
Theoretical power of saturated transformers

$\subseteq$ *constant-depth threshold circuits*

(with floating point datatype)

	$\mathbb{F}$	$\mathbb{Q}$
Hard	$\subset \mathsf{AC}^0$	$\subseteq \mathsf{AC}^0$
Saturated	$\boxed{\subseteq \mathsf{TC}^0}$	$= \mathsf{ALL}$

Saturated transformers



[Hard Attention isn't All You Need: The Power of Saturated Transformers](#)

William Merrill, Ashish Sabharwal, Noah A. Smith, 2021

Transformers can count (like LSTMs)

- Counter languages like $a^n b^n$ are in TC^0

Language	Model	Bin-1 Accuracy [1, 50]↑	Bin-2 Accuracy [51, 100]↑	Bin-3 Accuracy [101, 150]↑
Shuffle-2	LSTM (Baseline)	**100.0**	**100.0**	**100.0**
	Transformer (Absolute Positional Encodings)	**100.0**	85.2	63.3
	Transformer (Relative Positional Encodings)	**100.0**	51.6	3.8
	Transformer (Only Positional Masking)	**100.0**	**100.0**	93.0
BoolExp-3	LSTM (Baseline)	**100.0**	**100.0**	99.7
	Transformer (Absolute Positional Encodings)	**100.0**	90.6	51.3
	Transformer (Relative Positional Encodings)	**100.0**	96.0	68.4
	Transformer (Only Positional Masking)	**100.0**	**100.0**	**99.8**
$a^n b^n c^n$	LSTM (Baseline)	**100.0**	**100.0**	97.8
	Transformer (Absolute Positional Encodings)	**100.0**	62.1	5.3
	Transformer (Relative Positional Encodings)	**100.0**	31.3	22.0
	Transformer (Only Positional Masking)	**100.0**	**100.0**	**100.0**

[On the Ability and Limitations of Transformers to Recognize Formal Languages]
Satwik Bhattamishra, Kabir Ahuja, Navin Goyal, 2020

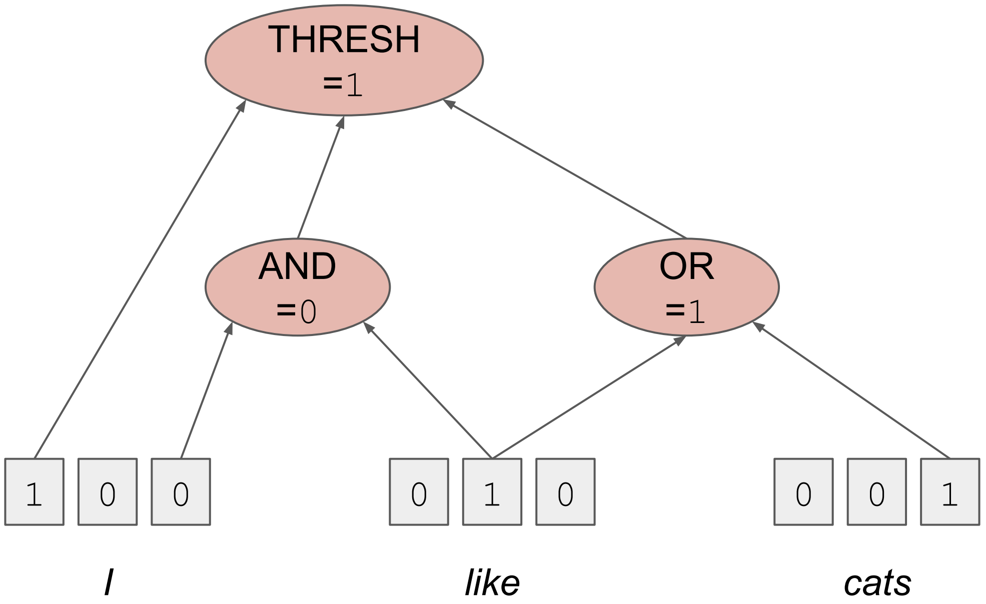
Conclusion

- Saturation as a means of comparing different neural net architectures
- LSTMs can count, unlike other RNNs
- So can transformers

Can “counting” explain the success, or limitations, of neural nets in NLP?

Thanks to Dana Angluin, Bob Frank, Noah Smith, Roy Schwartz, Yoav Goldberg, and Tal Linzen

What are threshold circuits, though?



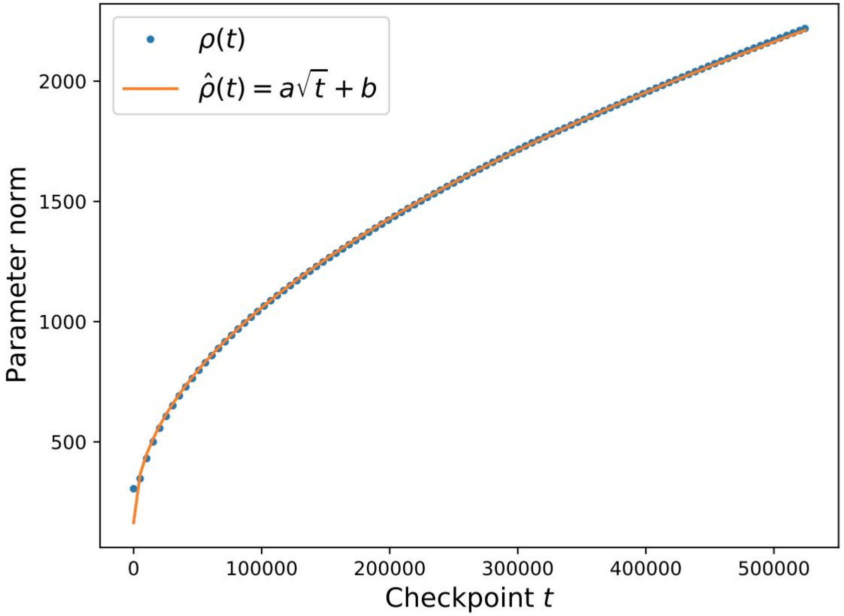
Saturated self attention

In a saturated transformer, all attention heads must be *uniform* over a subset of indices

Saturation as an inductive bias

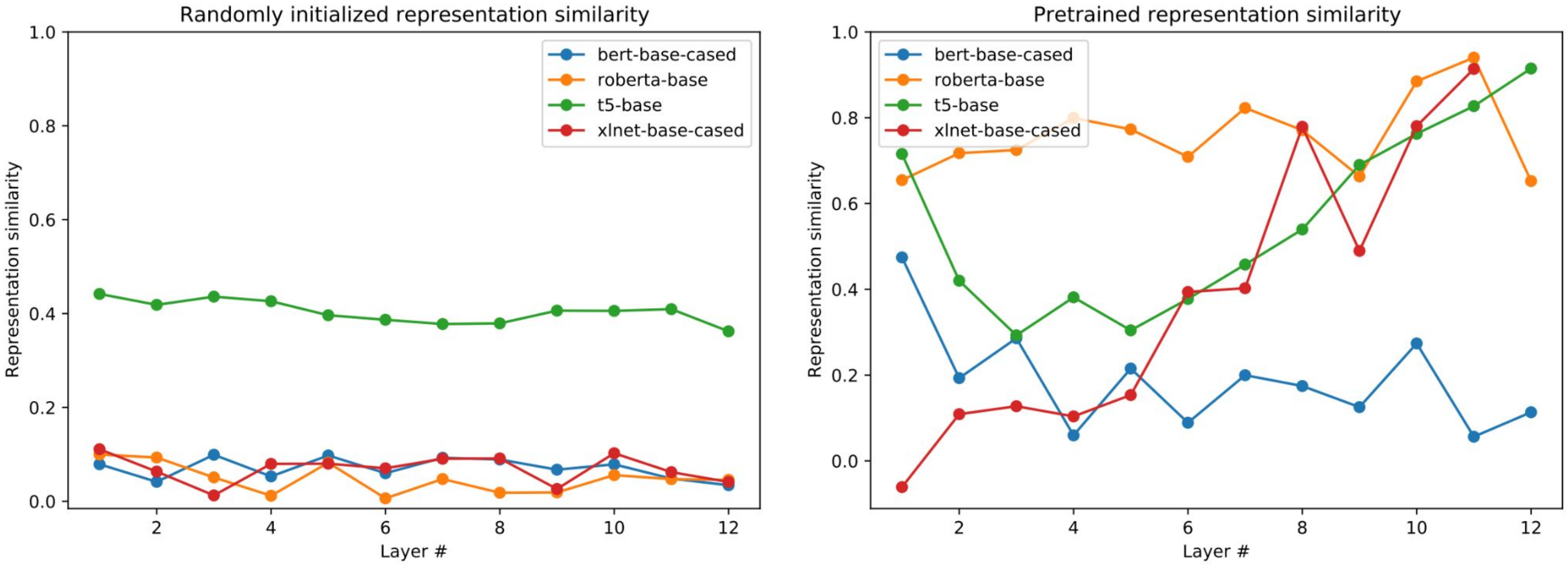
- Trained transformers approximate saturated transformers

- Because of *norm growth* during training



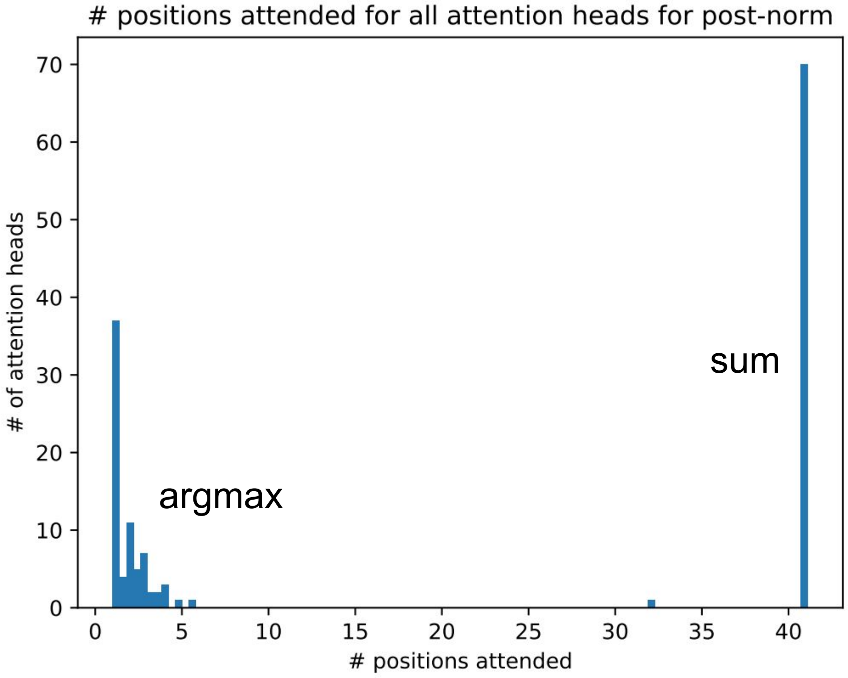
Effects of Parameter Norm Growth During Transformer Training: Inductive Bias from Gradient Descent
William Merrill, Vivek Ramanujan, Yoav Goldberg, Roy Schwartz, Noah A. Smith, 2021

Saturated representations in (trained) transformers



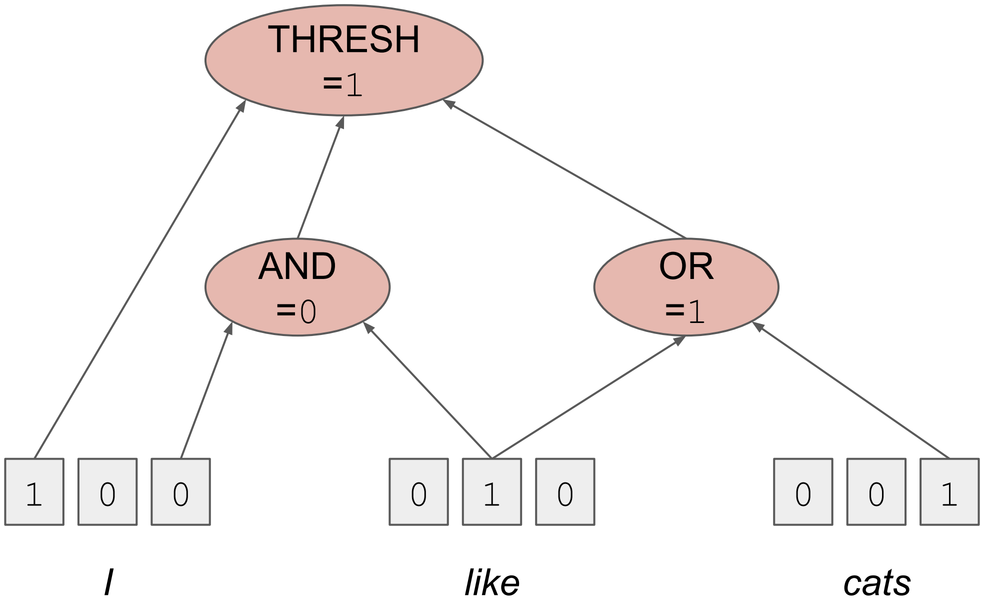
[Effects of Parameter Norm Growth During Transformer Training: Inductive Bias from Gradient Descent]
William Merrill, Vivek Ramanujan, Yoav Goldberg, Roy Schwartz, Noah A. Smith, 2021

Saturated attention heads are “sums” and “argmaxes”



[Effects of Parameter Norm Growth During Transformer Training: Inductive Bias from Gradient Descent](#)
William Merrill, Vivek Ramanujan, Yoav Goldberg, Roy Schwartz, Noah A. Smith, 2021

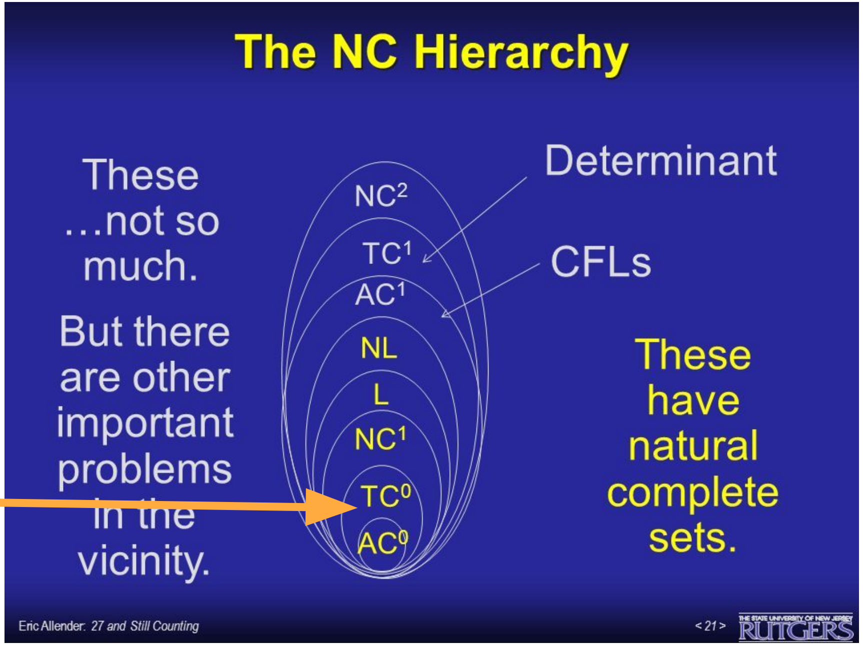
What are threshold circuits, though?



Circuit complexity classes ⇒ languages

- Established connections between regular languages, context-free languages, etc.
- TODO: How to relate circuit results to language?

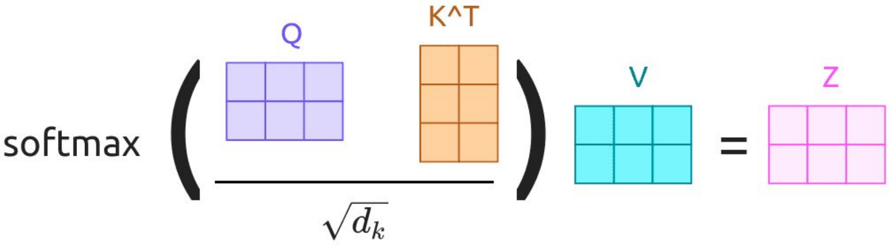
Saturated transformers



Summary

1. Saturated RNNs are finite-state
2. Saturated LSTMs can count
3. Saturated transformers are threshold circuits

Self attention



Probability-weighted sum