# Sequential neural networks as automata

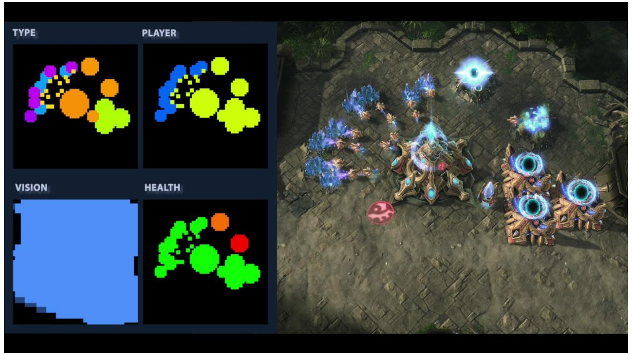
William Merrill

*Advised by*
Dana Angluin
Robert Frank

# Neural Networks

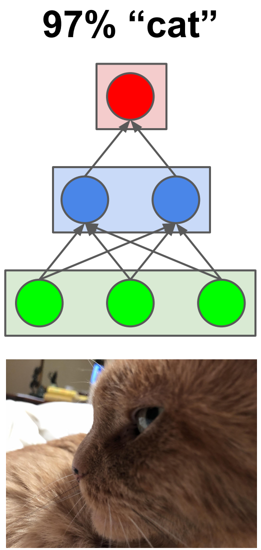
# Modern Artificial Intelligence

- Most recent advances in AI use **neural networks**
- Especially true for language (NLP)

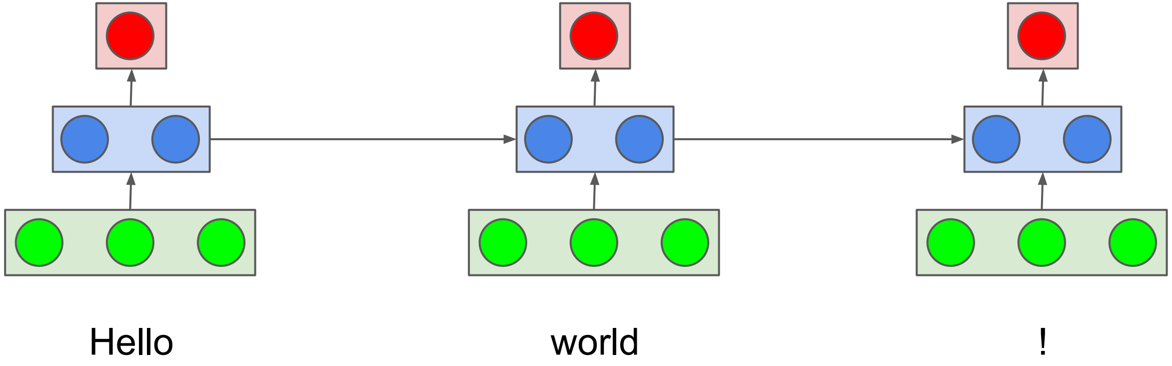
# What is a Neural Network?

- A network of artificial cells which send information to each other
- Learn the weights for cell connections from data



# Sequential Neural Networks

- For language, we use networks that can read variable-length sequences



# Interpretability of Neural Networks

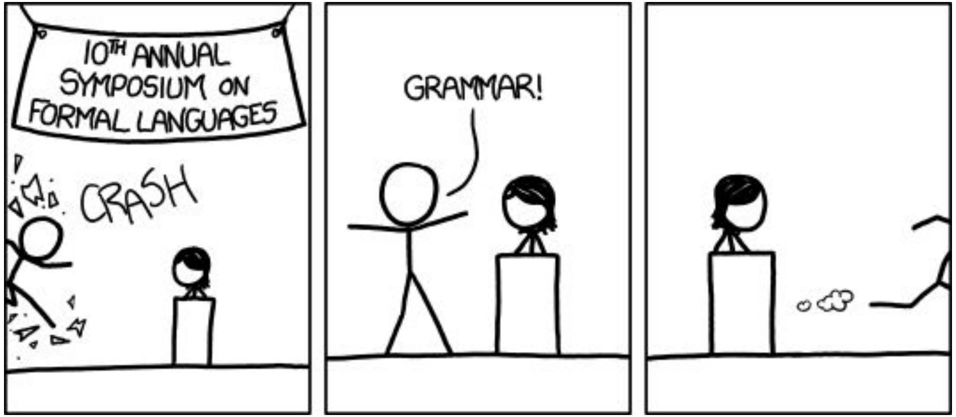
- Neural networks are good at translation, classification, summarization, etc.
- But, *how* and *why* they work is still an open question
- Cell connections must encode some kind of grammar

# Why is Interpretability Important?

- Guiding research
- Social accountability
- Intellectual value

# My Method

- Build off of **formal language theory**
- Prove what kinds of linguistic structure neural networks can model



# Formal Language Theory

# Formal Languages

- Potentially infinite sets of valid sentences

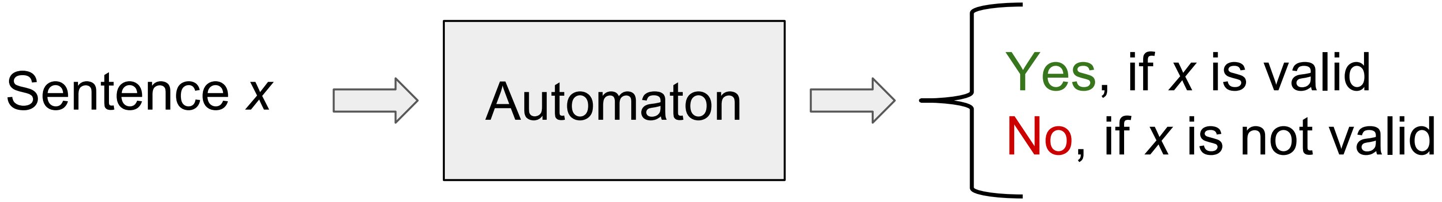
english = {“*I am Will.*”, “*I like AI!*”, ..}

íslenska = {“*Ég heiti Will.*”, “*Mér líkar við gervigreind!*”, ..}

palindromes = {“*aa*”, “*aba*”, “*abba*”, ..}

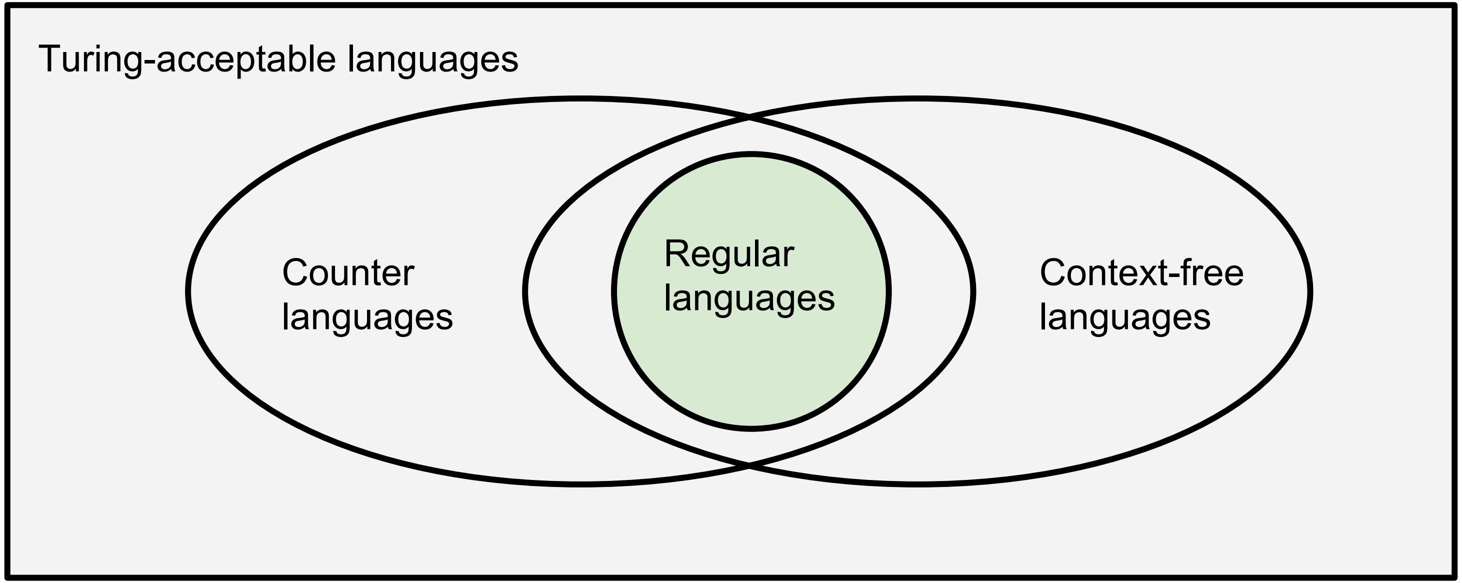
# Automata

- **Grammar/Automaton**: Computational device that decides whether a sentence is in a language (says yes/no)



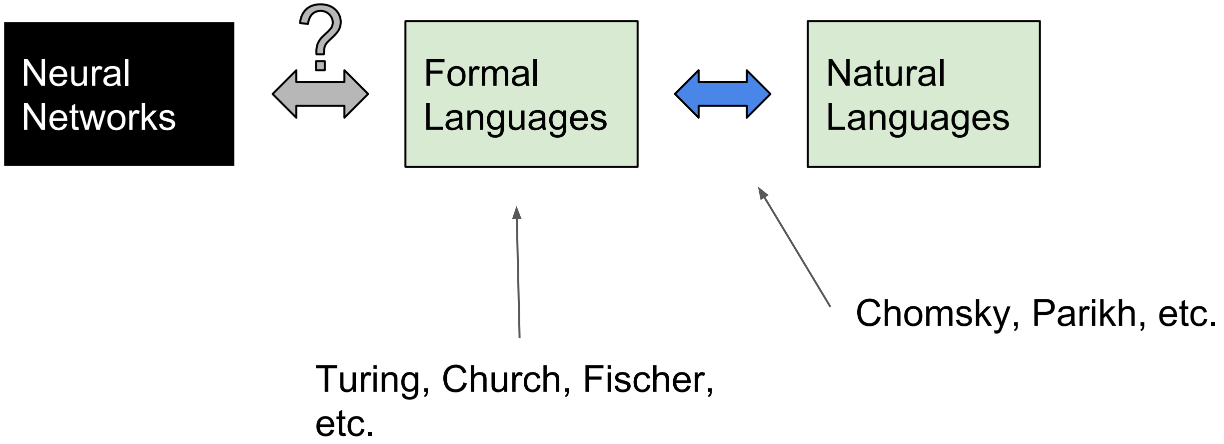
# Types of Automata

- More computationally complex automata can accept more languages



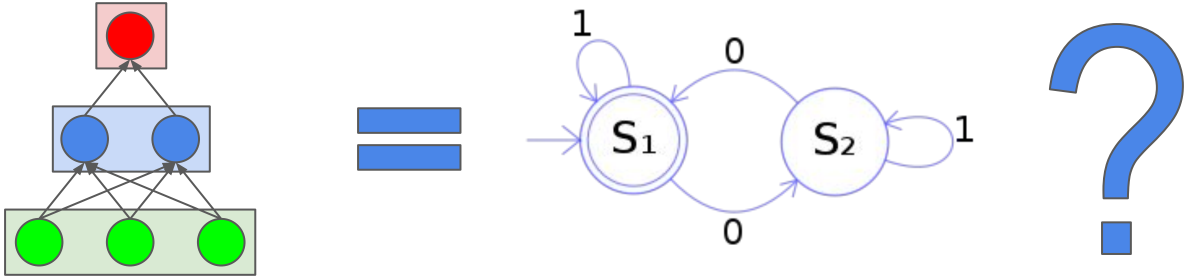
# Formal and Natural Languages

- Formal languages and automata are well studied (since 1930s)
- Formal languages model structures in natural language



# Research Questions

1. What kinds of formal languages can neural networks accept?
2. How do these languages relate to formal models of natural language?



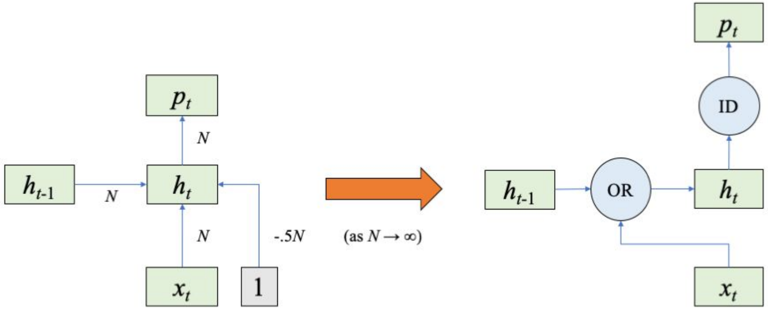
# My Contributions

1. Definitions
    a. Language acceptance for neural networks
    b. Measure of network’s memory
2. Results
    a. SRNs
    b. LSTMs
    c. Attention
    d. CNNs
3. Experiments

# Definitions

# Asymptotic Acceptance

- Networks output a probability (not yes/no)
- Need to make network say yes or no




**Definition 1.2.2** (Asymptotic acceptance). Let $L$ be a language with indicator function $\mathbb{1}_L$. A neural sequence acceptor $\hat{\mathbb{1}}$ with weights $\theta$ asymptotically accepts $L$ if

$$\lim_{N \to \infty} \hat{\mathbb{1}}^{N\theta} = \mathbb{1}_L.$$

# State Complexity

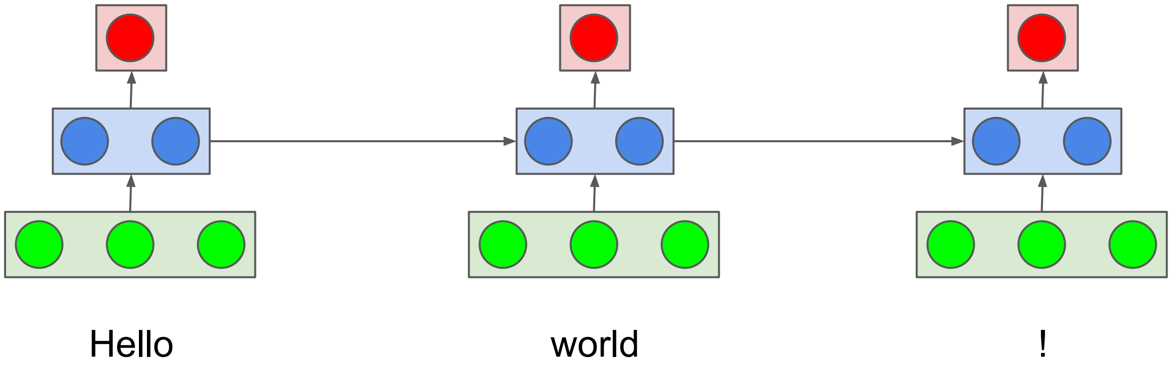
- Measure of network's memory (as function of sentence length)
- How many states can network be in after reading *n* words?

$$\text{memory} = \log_2(\text{state complexity})$$

# Theoretical Results

# Simple Recurrent Networks (SRNs)

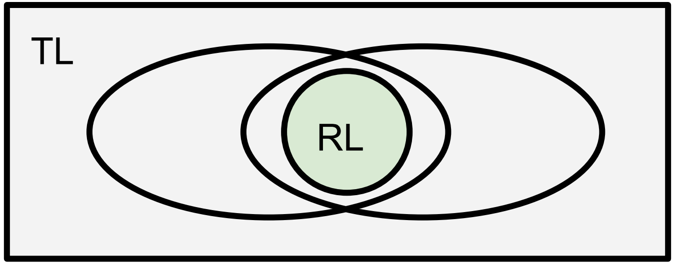
- Simplest architecture for recurrent neural networks
- Turing-complete under unconstrained definition of acceptance (Siegelmann, 1995)



# SRNs as Automata

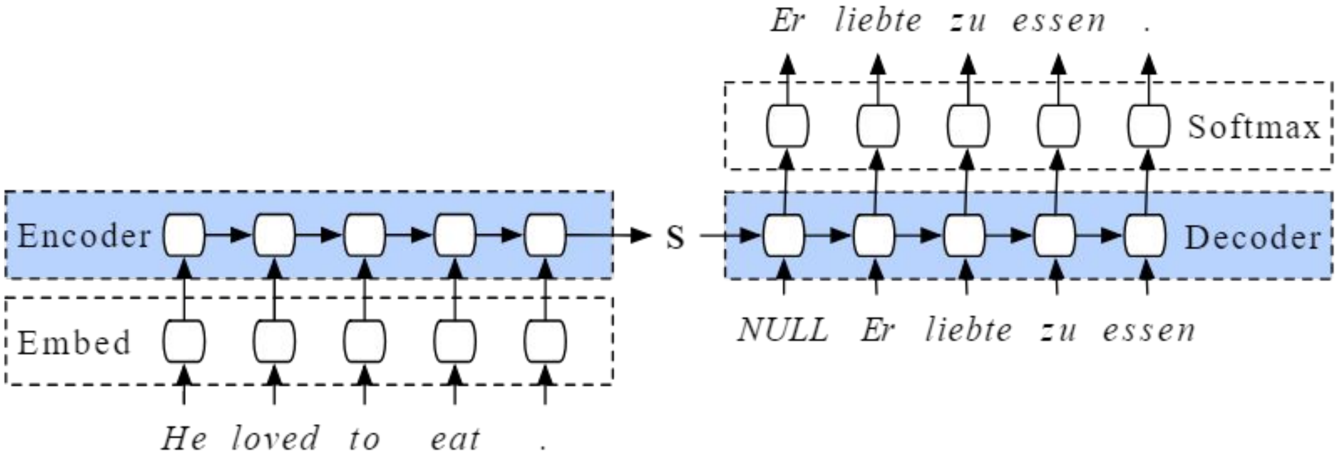
- ***Thm 2.1.2*: SRNs accept exactly the regular languages**
- State complexity: O(1) *(Constant)*

- Reduced characterization is more accurate than Siegelmann (1995)'s
- Similar result for gated recurrent units (GRUs)



# Long Short-Term Memory Networks (LSTMs)

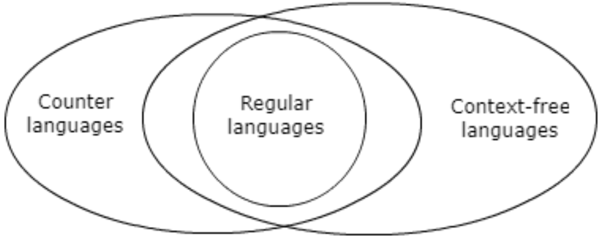
- More complicated recurrent neural network
- Used for machine translation and other tasks requiring syntax

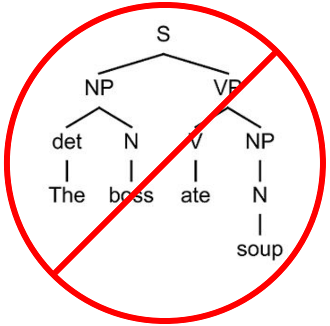


# LSTMs as Automata

- ***Thm 2.2.2*: LSTMs accept a subclass of the counter languages**
- State complexity: $O(n^k)$ *(Polynomial in sentence length)*

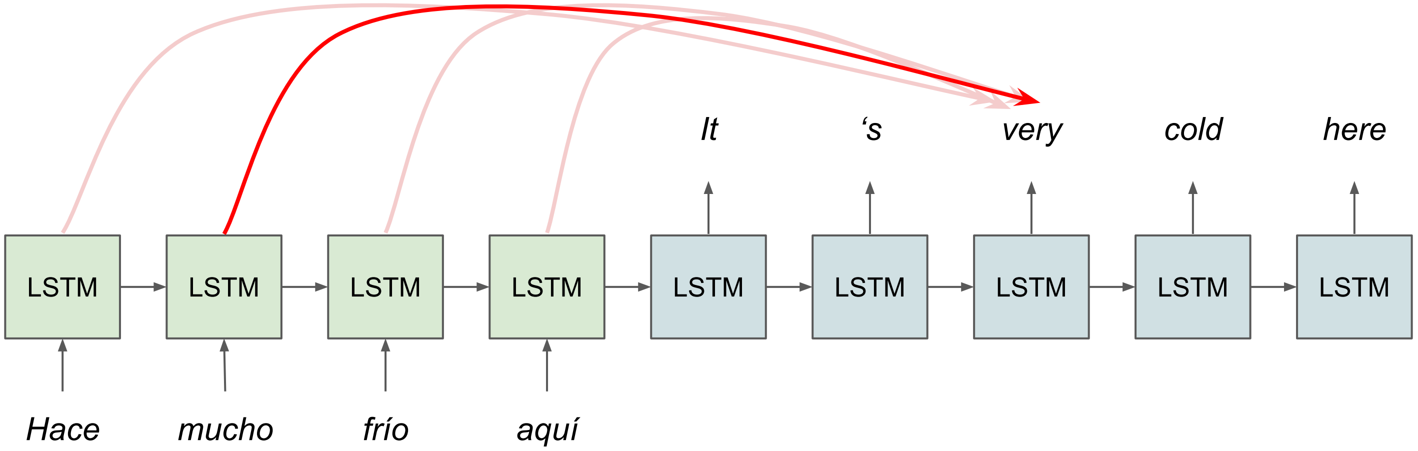
- More powerful than other recurrent networks
- But not powerful enough to model complex tree structure



# Attention

- Modern machine translation uses **attention**
- Focus on specific input words at different steps

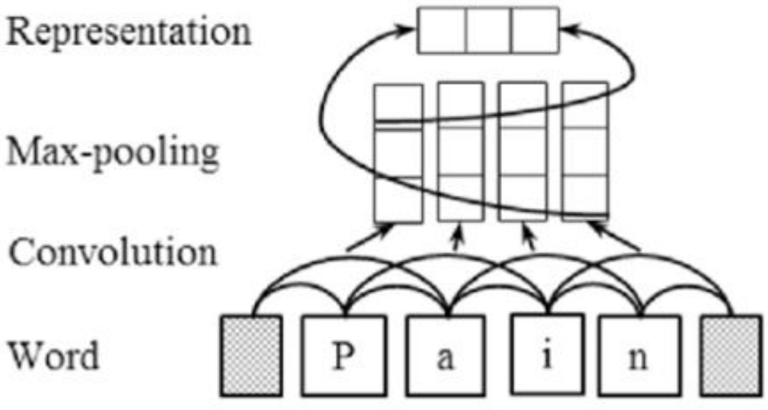


# Attention Results

- State complexity: $2^{O(n)}$ *(Exponential in sentence length)*
- Additional memory allows:
  - Copying a sequence (primitive translation)
  - More complex hierarchical representations
- Supports claim "attention is all you need" (Vaswani, 2017)

# Convolutional Neural Networks (CNNs)

- CNNs model words at the character level
- Deal with phonology, morphology
  - *pain* versus *pains*



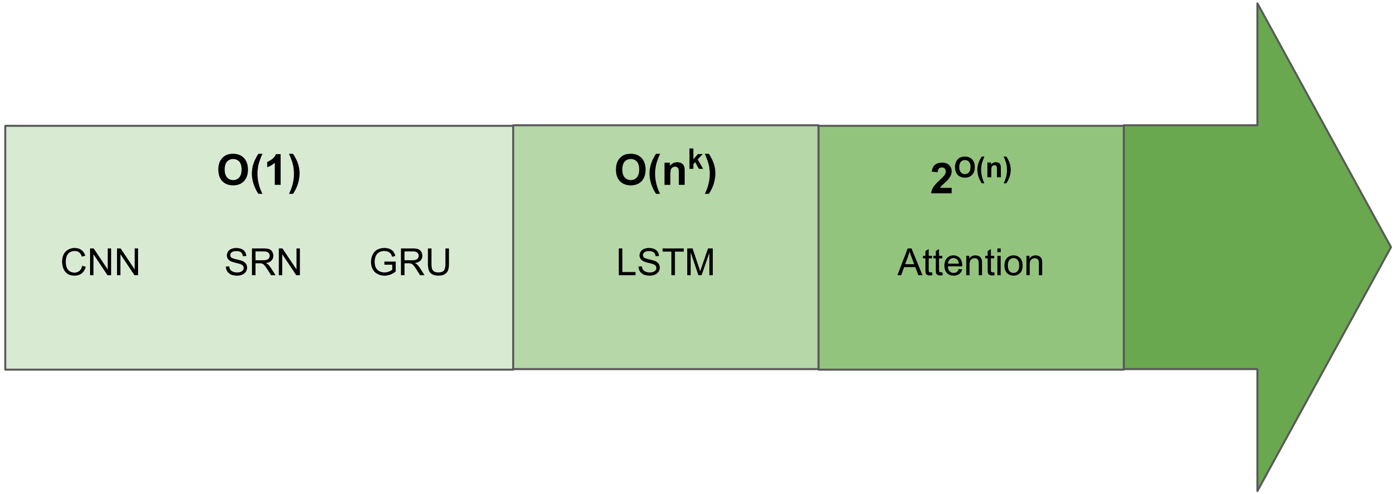
# CNNs as Automata

- ***Thm 3.1.1*: CNNs accept the strictly local languages**

- Explains success of character-level CNNs
- Strictly local languages* are good model of phonological grammar (Heinz et al., 2011)

*Tier-based strictly local languages

# State Complexity Hierarchy

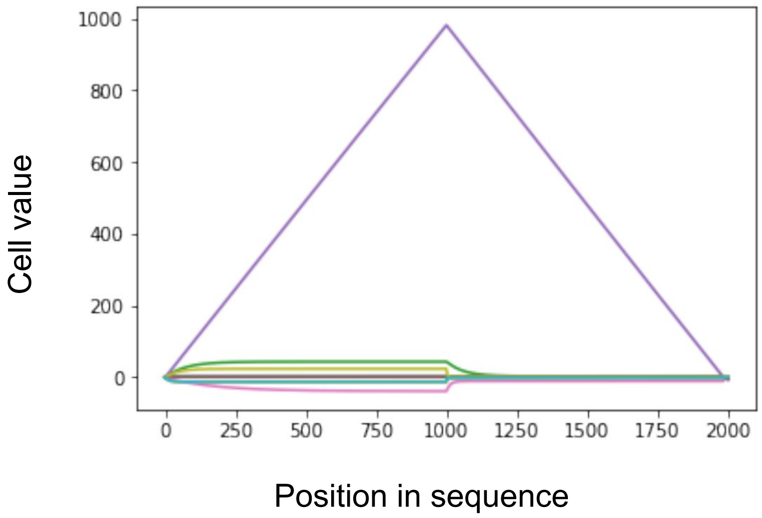


# Experiments

# LSTMs as Counter Automata

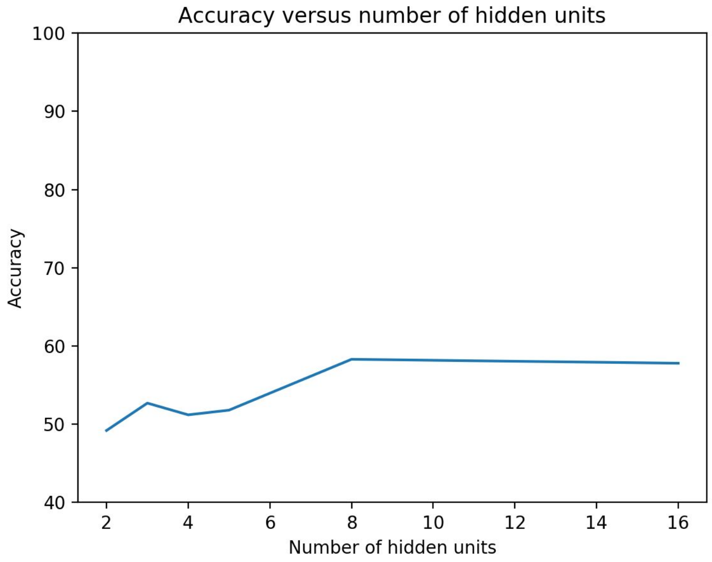
- Prediction: LSTMs are equivalent to counter machines
- LSTMs use memory to “count” (Weiss et al, 2018)

LSTM cell values on $a^n b^n$



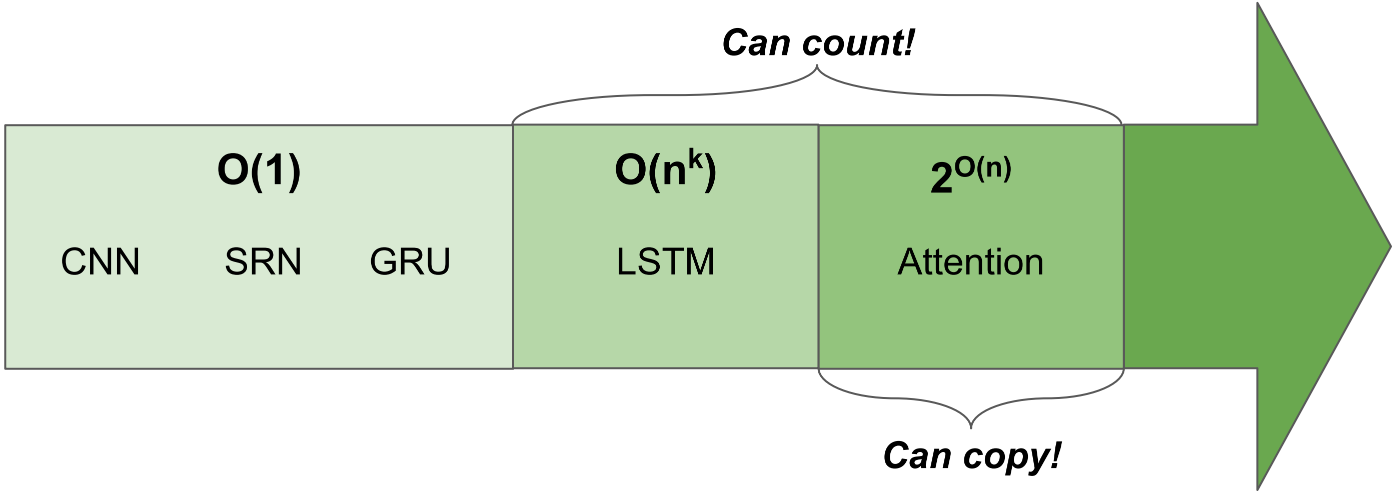
# Memory Constraints of LSTMs

- Prediction: LSTMs don’t have enough memory to reverse sentences
- LSTM cannot reverse long sentences!



# Validating State Complexity

- Counting requires $O(n^k)$ complexity
- Copying requires $2^{O(n)}$ complexity

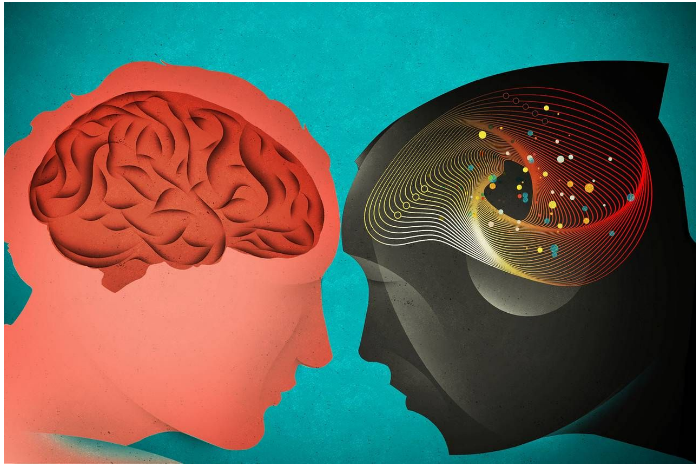


# Summary

- Theoretical tools
  - Language acceptance
  - Formalizing memory
- Results about types of networks
- Experiments

# Conclusion

- Step towards understanding the "black box" of neural networks
  - Extendable to other architectures
- Related neural networks to mental grammar
  - LSTM *can't* do complex trees
  - CNN *can* do phonology



# Acknowledgements

- My advisors, Bob and Dana
- Computational Linguistics at Yale:
  - Yiding, David, Noah, Andrew, Annie, Yong, Simon, Aarohi, Yi Chern, Sarah, Rachel
- Linguistics Senior Seminar:
  - Anelisa, James, Jay, Jisu, Magda, Noah, Rose, Hadas, Raffaella
- Advanced Natural Language Processing Seminar:
  - Michi, Suyi, Davey, John, Yavuz, Gaurav, Tianwei, Tomoe, Rui, Danny, Angus, Brian, Yong, Garrett, Noah, Alex, Talley, Ishita, Bo, Jack, Tao, Yi Chern, Irene, Drago
- Vidur Joshi and others @ Allen Institute for Artificial Intelligence