Number Sequence Prediction Problems for Evaluating Computational Powers of Neural Networks

Hyoungwook Nam
College of Liberal Studies
Seoul National University

Segwang Kim
Department of Electrical and Computer Engineering
Seoul National University

Kyomin Jung
Department of Electrical and Computer Engineering
Seoul National University

MOTIVATION

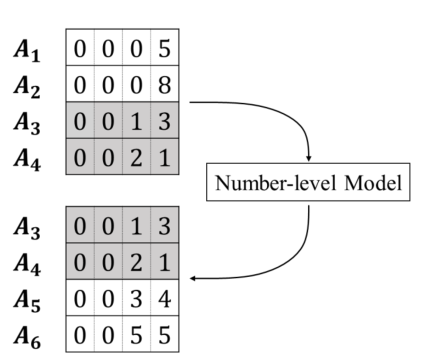
- Neural networks have been very successful for learning tasks on various data types including **image data and phonetic data**
- But it is hard to find well-defined **discrete and algorithmic** tasks where neural networks have been successfully applied
- Inspired by **number series tests for human intelligence**, number sequence prediction tasks can **assess computational powers** of neural networks

OBJECTIVES

- Define a set of algorithmic machine learning tasks with **numerical sequences**
- Quantify the **complexities** of simulating the sequence generation rules
- Evaluate **computational powers** of current deep learning models

TWO TYPES OF PROBLEMS

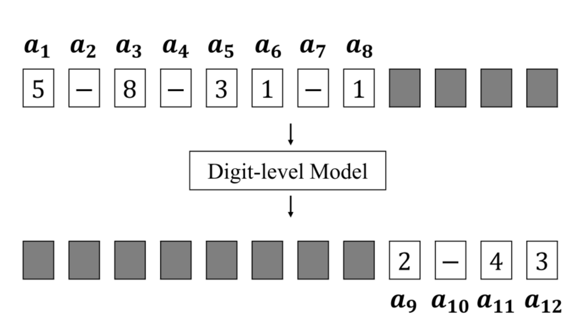
Number-level (CNN)



Digits in a 2-dimensional grid

Predict the **numbers in parallel**

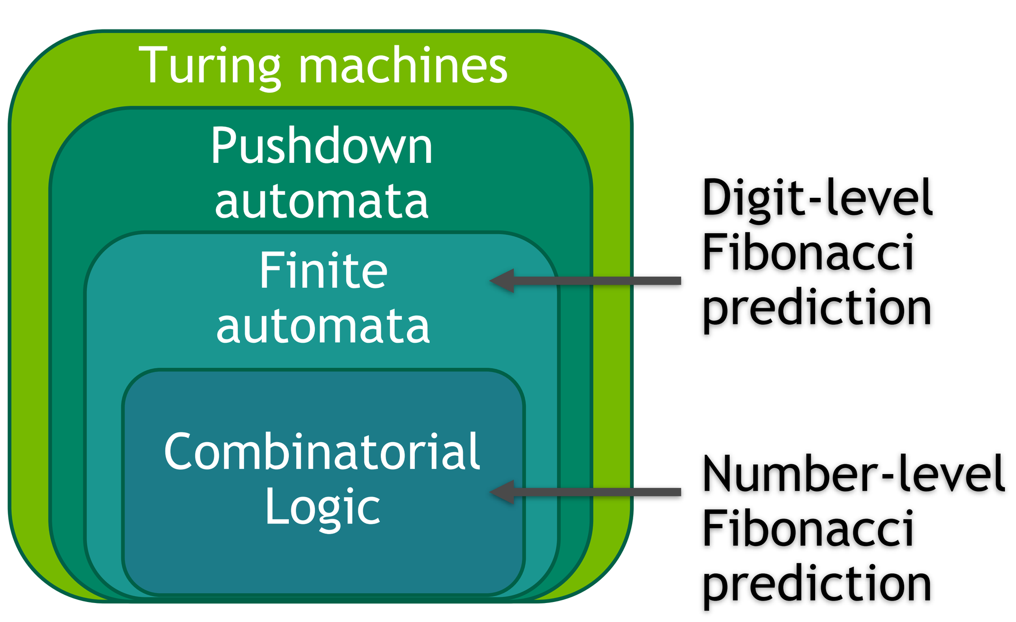
Digit-level (RNN)



A digit input per a time step

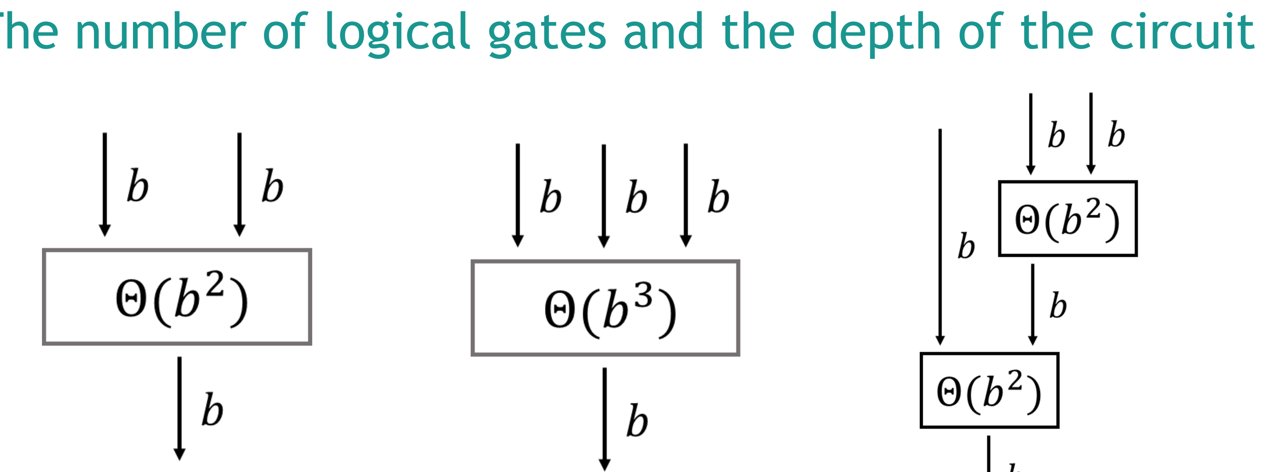
Predict the **digits sequentially**

COMPUTATIONAL POWERS



DIFFICULTY AND COMPLEXITY

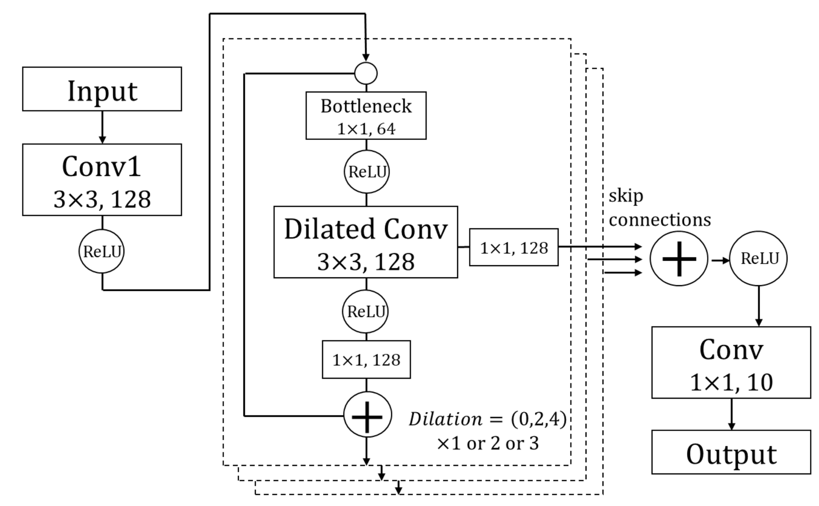
The number of logical gates and the depth of the circuit



Order-2	Order-3	Order-2
Width = $\theta(b^2)$	Width = $\theta(b^3)$	Width = $\theta(b^2)$
Depth = 1	Depth = 1	Depth = 2

NUMBER-LEVEL SEQUENCE PREDICTION

Models



Sequences

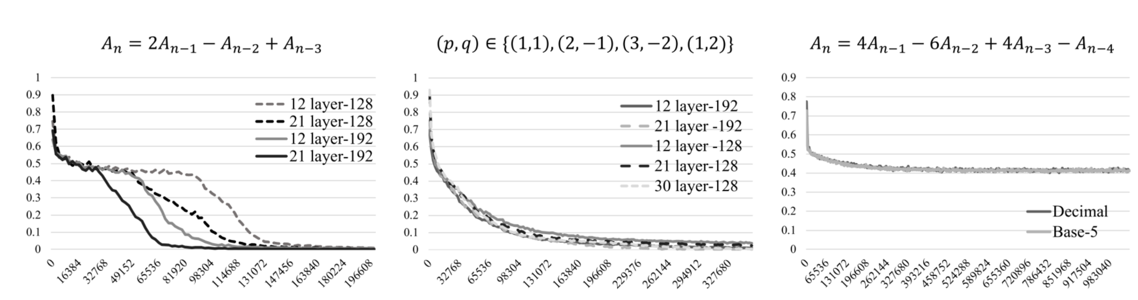
Order-2 Relations

- Fibonacci: $A_{n+2} = A_{n+1} + A_n$
- Arithmetic: $A_{n+2} = 2A_{n+1} - A_n$

Order-3 Relations

- Progression: $A_{n+3} = 3A_{n+2} - 3A_{n+1} + A_n$
- Jumping Fibonacci: $A_{n+3} = A_{n+2} + A_n$

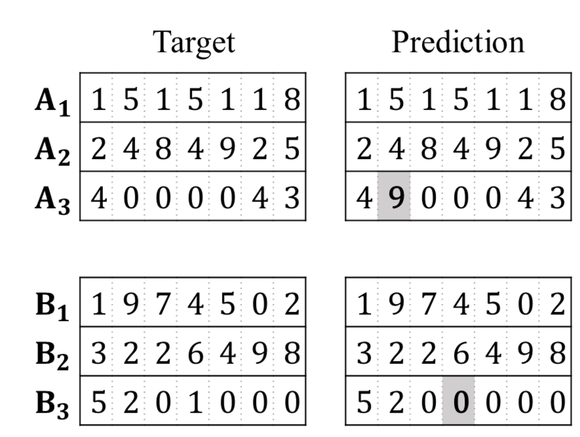
Results



Y-axis: error rates / X-axis: # of training examples

Depth is a better indicator for the complexity

CNNs tend to learn deep but narrow rules

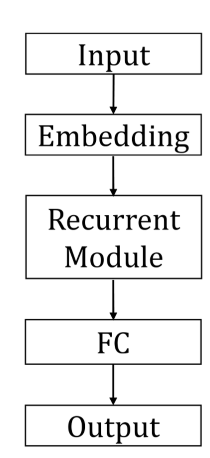


Error examples from number-level Fibonacci prediction

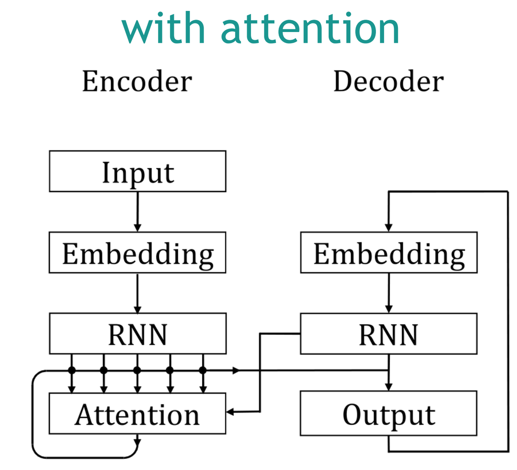
DIGIT-LEVEL SEQUENCE PREDICTION

Models

Recurrent model



Encoder-decoder model with attention



Sequences

Counting sequences ($A_{n+1} = A_n + c$)

- **Finite** automata

Palindromes (e.g. abcd_dcba)

- **Pushdown** automata

Fibonacci/Arithmetic/Geometric

- **Queue** automata (= Turing)

Results

Tasks	Reverse-order (training)	Geometric	Arithmetic	Fibonacci
LSTM	28.4% (1.2%)	79.4%	77.1%	80.5%
GRU	51.9% (0.9%)	69.0%	77.1%	79.3%
Attention(unidirectional)	42.0% (8.8%)	62.8%	77.0%	69.3%
Attention(bidirectional)	0.0% (0.0%)	51.0%	72.9%	60.9%
Stack-RNN	**0.0%** (0.0%)	64.1%	63.8%	69.4%
NTM	**0.0%** (0.0%)	57.1%	65.7%	68.1%

- Reverse-order (palindrome) training errors suggest that RNNs can simulate **finite automata**
- **Memory-augmented models** could simulate up to **pushdown automata**

TAKEAWAYS

- **Number sequence predictions** effectively evaluate computational powers of neural networks
- **Complexity** of a number-level problem can be defined with the combinatorial logic
- **Computational powers** of current recurrent models are **limited** up to those of pushdown automata