Deep learning: Challenges in learning and generalization

Tomas Mikolov, Facebook AI

What is generalization?

Memorization

- to remember all training examples (Universal Approximator)
- 2 + 3 = 5, 3 + 2 = 5, ...

Generalization

- to infer novel conclusions
- 123 + 234 = 357, ...

How much do deep neural networks generalize?

- Often less than we would expect (or hope)
- It is easy to make wrong conclusions when using deep networks without understanding how they work
- In this talk: examples of limits of learning in recurrent neural networks

Language Modeling for strong AI

- Language models assign probability to sentences

- P("Capital city of Czech is Prague") > P("Capital city of Czech is Barcelona")

AI-complete problem:

- *A bit of progress in language modeling*, Joshua Goodman 2001
- Hutter prize compression challenge

AI-completeness of language modeling

We could build intelligent question answering systems and chatbots using perfect language models:

- P(“Is the value of of Pi larger than 3.14? Yes.”) >
  P(“Is the value of of Pi larger than 3.14? No.”)

Language models can generate novel text:

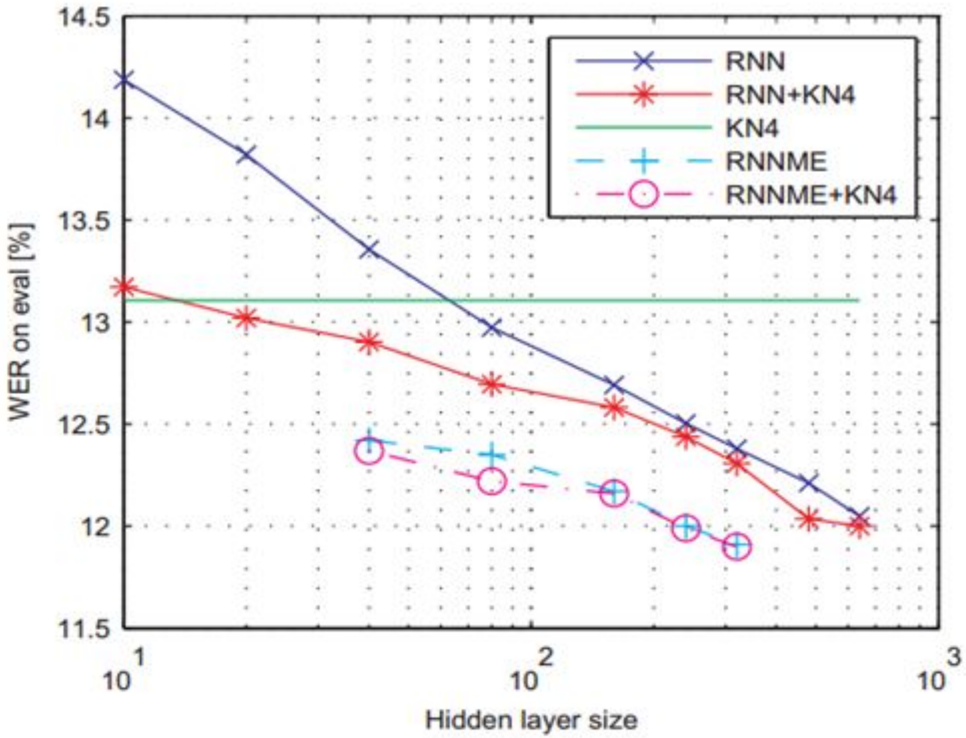
- better language models generate significantly better text (RNNLM, 2010)

Recurrent neural language models

- Breakthrough after 30 years of dominance of n-grams

- The bigger, the better!
    - This continues to be the mainstream even today

- Can this lead to AGI?



Strategies for training large scale neural network language models, Mikolov et al, 2011

End-to-end Machine Translation with RNNLM (2012)

Simple idea - create a training set from pairs of sentences in different languages:

1. Today it is Sunday. Hoy es domingo.
   It was sunny yesterday. Ayer estaba soleado.
   …

2. Train RNNLM

3. Generate continuation of text given only the English sentence: translation!

End-to-end Machine Translation with RNNLM (2012)

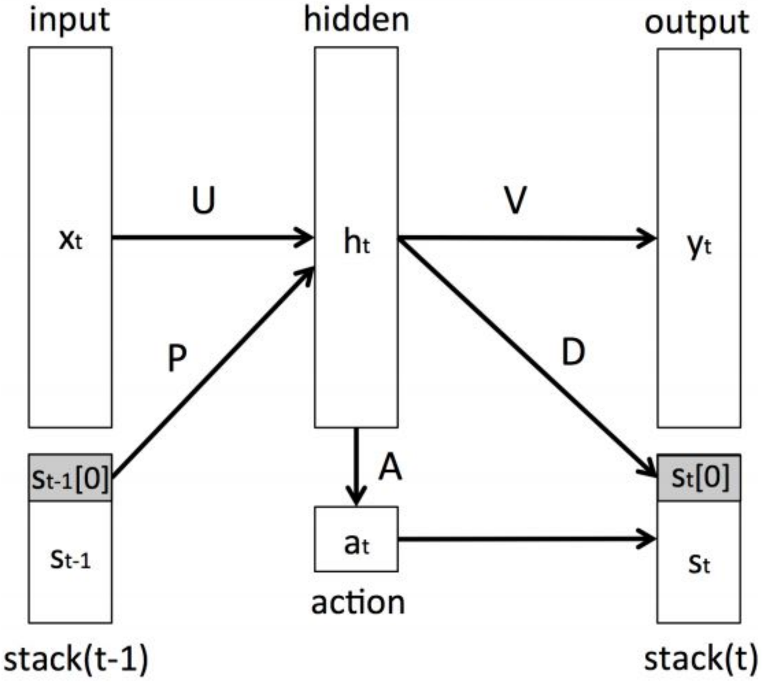
- Problem: the performance drops for long sentences

- Even worse: cannot **learn** identity!
  - Today it is Sunday. Today it is Sunday.
    It was sunny yesterday. It was sunny yesterday.
    …
  - Can perfectly memorize training examples, but fails when test data contain longer sequences

Towards RNNs that learn algorithms

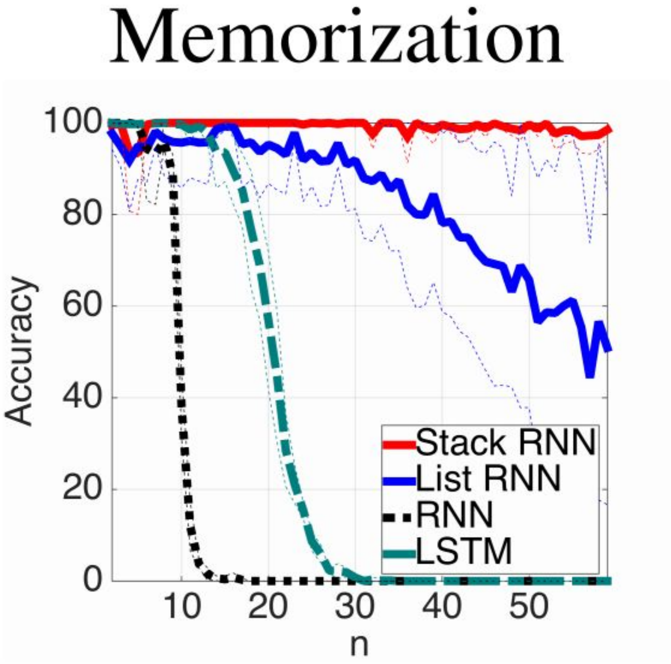
- RNNs trained with stochastic gradient descent usually do not learn algorithms
  - just memorize training examples
  - does not matter how many hidden layers we use, and how big the hidden layers are
- This does not have to be a serious problem for applied machine learning
  - memorization is often just fine
- A critical issue for achieving strong AI / AGI

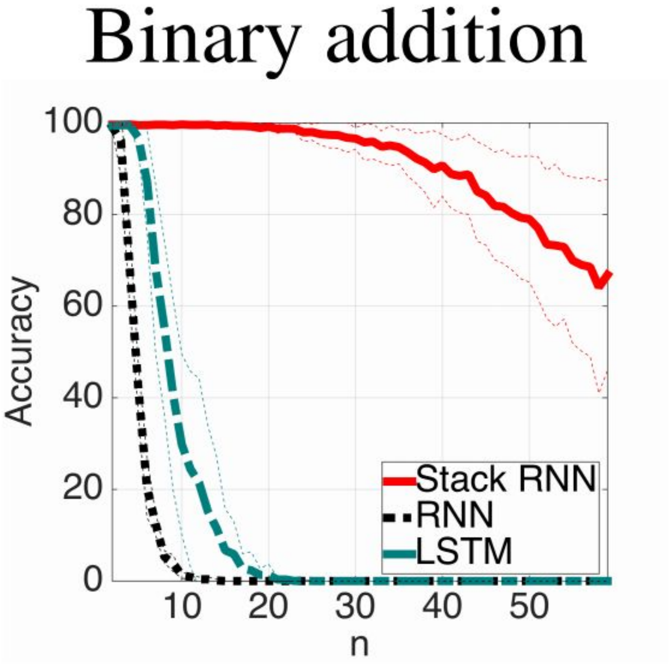
Stack-augmented RNN



Inferring algorithmic patterns with stack-augmented recurrent nets, Joulin & Mikolov, 2015

Generalization in RNNs





Binary addition learned with no supervision

Inputs:	.	1	0	0	0	1	1	+	1	1	1	0	=	**1**	**0**	**0**	**0**	**1**	**1**	**.**	
Predictions:	0	0	.	0	1	0	1	0	1	1	1	1	**1**	**0**	**0**	**0**	**1**	**1**	**.**	0	
Stack 1:	0							1					1							0	Counter
Stack 2:	1	-1											1			0				1	End of number 2
Stack 3:	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	Number 2
Stack 4:								1	0	0	0	0	0	0	0	1					Length of number 2
Stack 5:	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	Carry
Stack 6:			1	0	0	0	1	1					0	1	0	0	0	1	-1		Number 1
Stack 7:																					Junk
Stack 8:																					Junk
Stack 9:																					Junk
Stack 10:																					Junk

Future research - algorithmic transfer learning

- current machine learning models are usually bad at high-level transfer learning

- the “solution” that is learned is often closer to look up table than minimum description length solution

- teaching an RNN to solve a slightly more complex version of already solved task thus mostly fails
  A roadmap towards machine intelligence, Mikolov et al, 2015

Future research - different approach to learning

- we need much less supervision

- probably no SGD, no convergence (learning never ends)

- maybe more fundamental (basic) model than RNN?
  - are memory, learning, tasks, rewards etc. just emergent properties in a more general system?