Recurrent Neural Language Models and Weighted Automata

Extraction and Approximation

Reda Marzouk and Colin de la Higuera

Nantes University

April 2021

Motivation

Deep Learning for language modeling tasks:

Empirical success vs. Poor Theory

Theoretical issues

- **Expressiveness power:** Formalization of the class of languages a given model architecture can *represent*,

Motivation

Deep Learning for language modeling tasks:

Empirical success vs. Poor Theory

Theoretical issues

- **Expressiveness power:** Formalization of the class of languages a given model architecture can *represent*,
- **The learning inductive bias:** The class of languages a given pair (architecture, learning algorithm) can *learn*,

Motivation

Deep Learning for language modeling tasks:

Empirical success vs. Poor Theory

Theoretical issues

- **Expressiveness power:** Formalization of the class of languages a given model architecture can *represent*,
- **The learning inductive bias:** The class of languages a given pair (architecture, learning algorithm) can *learn*,
- **Semantics of the distributional representation of neural language models**

Why answers are important?

- More Principled design architectures/learning algorithms,

Motivation

Deep Learning for language modeling tasks:

Empirical success vs. Poor Theory

Theoretical issues

- **Expressiveness power:** Formalization of the class of languages a given model architecture can *represent*,
- **The learning inductive bias:** The class of languages a given pair (architecture, learning algorithm) can *learn*,
- **Semantics of the distributional representation of neural language models**

Why answers are important?

- More Principled design architectures/learning algorithms,
- Interpretability of models,

Motivation

Deep Learning for language modeling tasks:

Empirical success vs. Poor Theory

Theoretical issues

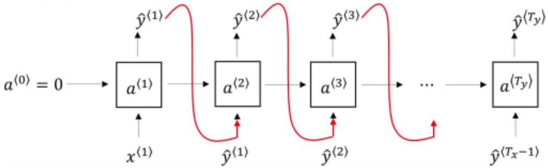
- **Expressiveness power:** Formalization of the class of languages a given model architecture can *represent*,
- **The learning inductive bias:** The class of languages a given pair (architecture, learning algorithm) can *learn*,
- **Semantics of the distributional representation of neural language models**

Why answers are important?

- More Principled design architectures/learning algorithms,
- Interpretability of models,
- Property Checkability of models

Why is a general theory of RNNs hard to develop?

• **Example 1:** RNN Language models with ReLu activation function:

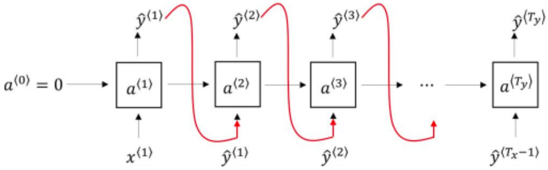


Basic question

- **Interpretation:** The output is the next symbol probability given a prefix sequence,

Why is a general theory of RNNs hard to develop?

• **Example 1:** RNN Language models with ReLu activation function:



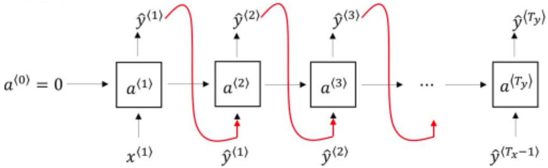
Basic question

- **Interpretation:** The output is the next symbol probability given a prefix sequence,
- **Property:** Is the model consistent? (i.e. $\sum_{w\in\Sigma^*} \mathbb{P}(w) = 1$)

Not necessarily (Chen et al. 2018 [1])

Why is a general theory of RNNs hard to develop?

• **Example 1:** RNN Language models with ReLu activation function:



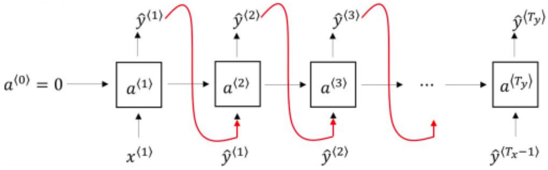
Basic question

- **Interpretation:** The output is the next symbol probability given a prefix sequence,
- **Property:** Is the model consistent? (i.e. $\sum_{w\in\Sigma^*} \mathbb{P}(w) = 1$)

  Not necessarily (Chen et al. 2018 [1])
- Even worse, deciding consistency is an undecidable problem,

Why is a general theory of RNNs hard to develop?

- **Example 1:** LSTMs, GRUs language models



Basic question

- **Interpretation:** The output is the next symbol probability given a prefix sequence,
- **Property:** Is the model consistent? (i.e. $\sum_{w \in \Sigma^*} \mathbb{P}(w) = 1$)

LSTM, GRUs language models are consistent
(Welleck et al., 2020 [2])

Building a bridge between RNN-LMs and Weighted Automata:

Extraction and Approximation

RNN-LMs and Weighted Automata

Problem: Approximating RNN-LMs with Finite automata

• Given a target RNN-LM R, a class of finite state automata $\mathcal{C}$, Find a finite state automaton $A \in \mathcal{C}$ with R smallest description size that approximates well R

Motivation

- Model compression,
- Model checking,
- Advanced decoding and pattern queries,
- Adversial attacks through model stealing,

RNN LMs and Weighted Automata: Extraction

Problem: Approximating RNN-LMs with Finite automata

• Given a target (consistent) RNN-LM R, a class of finite state automata $\mathcal{C}$,
Find a finite state automaton $A \in \mathcal{C}$ with the smallest description size that approximates well R

Questions

- Architecture-independent algorithm?

RNN LMs and Weighted Automata: Extraction

Problem: Approximating RNN-LMs with Finite automata

• Given a target (consistent) RNN-LM R, a class of finite state automata $\mathcal{C}$,
Find a finite state automaton $A \in \mathcal{C}$ with the smallest description size that approximates well R

Questions

- Architecture-independent algorithm?
- Agnostic vs. Exact case :Presence of non-linearities in RNN state transitions (**e.g.** RNNs with ReLu can represent irrational languages)

RNN LMs and Weighted Automata: Extraction

Problem: Approximating RNN-LMs with Finite automata

• Given a target (consistent) RNN-LM R, a class of finite state automata $\mathcal{C}$,
Find a finite state automaton $A \in \mathcal{C}$ with the smallest description size that approximates well R

Questions

- Architecture-independent algorithm?
- Agnostic vs. Exact case :Presence of non-linearities in RNN state transitions (**e.g.** RNNs with ReLu can represent irrational languages)
- How to measure the quality of approximation?

RNN LMs and Weighted Automata: Extraction

Problem: Approximating RNN-LMs with Finite automata

• Given a target (consistent) RNN-LM R, a class of finite state automata $\mathcal{C}$,
Find a finite state automaton $A \in \mathcal{C}$ with the smallest description size that approximates well R

Questions

- Architecture-independent algorithm?
- Agnostic vs. Exact case :Presence of non-linearities in RNN state transitions (**e.g.** RNNs with ReLu can represent irrational languages)
- How to measure the quality of approximation?
- Computational complexity issues?

RNN LMs and Weighted Automata: Extraction

Problem: Approximating RNN-LMs with Finite automata

• Given a target (consistent) RNN-LM R, a class of finite state automata $\mathcal{C}$,
Find a finite state automaton $A \in \mathcal{C}$ with the smallest description size that approximates well R

Questions

- Architecture-independent algorithm?
- Agnostic vs. Exact case :Presence of non-linearities in RNN state transitions (**e.g.** RNNs with ReLu can represent irrational languages)
- How to measure the quality of approximation?
- Computational complexity issues?
- Which class of weighted automata to approximate RNN-LMs?

Which type of weighted automata to approximate RNN-LMs with?

Weighted automata

Weighted Automata (WA): Algebraic Characterization

A weighted automata (WA) over an alphabet Σ is a parametrized model $\{\alpha, \{A_\sigma\}_{\sigma\in\Sigma}, \beta\}\}$ where $\alpha, \beta \in \mathbb{R}^n$, $A_\sigma \in \mathbb{R}^{n\times n}$. The weight of a string $w = \sigma_1..\sigma_{|w|} \in \Sigma^*$ is given by: $f(w) = \alpha^T \prod_{i=1}^{|w|} A_{\sigma_i}\beta$

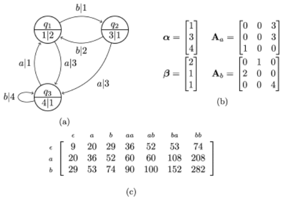


Figure: A graphical representation of a WFA (Balle et al. [3])

Weighted automata

Advantages and drawbacks

Advantages:

• High expressiveness power (as compared to other classes of weighted automata),

• Noise Robustness of Spectral approaches for extracting WA,

Drawbacks:

• Not a generative model (Important for text generation)

Proposed approach

- Spectral approach (Ayache et al., 2018 [4])
- Regression in state space (Okudono et al., 2020 [5])

Probabilistic Nondeterministic Finite Automata(PFA)

Definition: Probabilistic finite automata

A probabilistic finite automaton (PFA) is a weighted automaton where α defines a probability distribution (the initial probability distribution), and $\forall \sigma \in \Sigma : \ A_\sigma(i, j)$ represents the probability of emitting symbol σ and transitioning to state j, when we are at state i

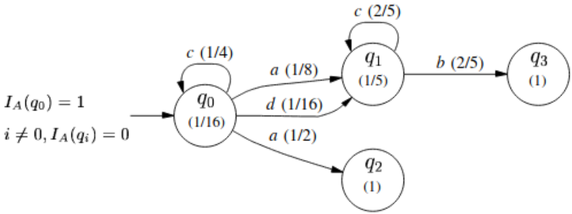


Figure: A graphical representation of a PFA (Vidal et al. [6])

Probabilistic Nondeterministic Finite Automata(PFA)

Definition: Probabilistic finite automata

A probabilistic finite automaton (PFA) is a weighted automaton where α defines a probability distribution (the initial probability distribution), and $\forall \sigma \in \Sigma : \ A_\sigma(i, j)$ represents the probability of emitting symbol σ and transitioning to state j, when we are at state i

Advantages and drawbacks

Advantages:

- Suitable for text generation tasks,
- Can be learnt using spectral approaches

Drawbacks:

- Though, the ouput of a spectral algorithm is given as an observable operator model (loss of weight interpretability)

Deterministic PFA

Definition: Deterministic PFA

A deterministic PFA (DPFA) is a PFA such that:

- There is only one initial state,
- for each state $q \in Q$, for each symbol $\sigma \in \Sigma$, there is at most one transition,

Advantages and drawbacks

Advantages:

• Transparent and readily Interpretable,

• Can be used as a generative model

Drawbacks:

• Low expressiveness power,

Proposed approach

L* variant for extracting PDFAs from RNN-LMs (Weiss et al. [7])

The complexity of comparing RNN Language models and Weighted Automata

RNNs and FSMs: Equivalence and quality of approximation

Equivalence problem between a PDFA and consistent RNN-LMs with ReLu activation function

• **Instance:** A *consistent* RNN-LM with ReLu activation function R, a PDFA $\mathcal{A}$,

• **Problem:** Are they equivalent?

Theorem (Marzouk, de la Higuera, 2020)

The equivalence problem between PDFA and consistent RNN-LMs with ReLu as an activation function is undecidable.

RNNs and FSMs: Equivalence and quality of approximation

Equivalence problem between a PDFA and consistent RNN-LMs with ReLu activation function

• **Instance:** A *consistent* RNN-LM with ReLu activation function R, a PDFA $\mathcal{A}$,

• **Problem:** Are they equivalent?

Theorem (Marzouk, de la Higuera, 2020)

The equivalence problem between PDFA and consistent RNN-LMs with ReLu as an activation function is undecidable.

- The proof is a reduction from the Halting Turing Machine problem.

RNNs and FSMs: Equivalence and quality of approximation

Equivalence problem between a PDFA and consistent RNN-LMs with ReLu activation function

• **Instance:** A *consistent* RNN-LM with ReLu activation function R, a PDFA $\mathcal{A}$,

• **Problem:** Are they equivalent?

Theorem (Marzouk, de la Higuera, 2020)

The equivalence problem between PDFA and consistent RNN-LMs with ReLu as an activation function is undecidable.

- The proof is a reduction from the Halting Turing Machine problem.
- As a corollary, same undecidability result holds for WFA/PFAs.

RNNs and FSMs: Equivalence and quality of approximation

Equivalence problem between a PDFA and consistent RNN-LMs with ReLu activation function

• **Instance:** A *consistent* RNN-LM with ReLu activation function R, a PDFA $\mathcal{A}$,

• **Problem:** Are they equivalent?

Theorem (Marzouk, de la Higuera, 2020)

The equivalence problem between PDFA and consistent RNN-LMs with ReLu as an activation function is undecidable.

- The proof is a reduction from the Halting Turing Machine problem.
- As a corollary, same undecidability result holds for WFA/PFAs.
- The equivalence problem in a bounded support is EXP-Hard.

RNNs and FSMs: Equivalence and quality of approximation

• Results on equivalence are negative. What about the approximation problem?

Approximation between a PFA and consistent RNN-LMs with ReLu activation function

• **Instance:** A consistent RNN-LM with ReLu activation function, a PFA $\mathcal{A}$, $c > 0$,

• **Problem:** Does there exist a word $w \in \Sigma^*$ such that $|R(w) - \mathcal{A}(w)| > c$?

Theorem (Marzouk, de la Higeura, 2020)

The approximation problem between a PFA and consistent RNN-LMs is decidable.

RNNs and FSMs: Equivalence and quality of approximation

Approximation between a PFA and consistent RNN-LMs with ReLu activation function in bounded support

• **Instance:** A consistent RNN-LM with ReLu activation function, a PFA $\mathcal{A}$, $c > 0$, $N > 0$

• **Problem:** Does there exist a word $w \in \Sigma^{\leq N}$ such that $|R(w) - \mathcal{A}(w)| > c$?

Theorem (Marzouk, de la Higuera, 2020)

The approximation problem in a bounded support is NP-Hard.

• **Proof.** Reduction from the 3-SAT problem.

Open questions and perspectives

- Weighted Automata Extraction algorithms from RNN language models with theoretical garantees,

Open questions and perspectives

- Weighted Automata Extraction algorithms from RNN language models with theoretical garantees,
- Generalization of weighted automata to families of non-linear WAs with nice expressiveness and learnability properties,

Open questions and perspectives

- Weighted Automata Extraction algorithms from RNN language models with theoretical garantees,
- Generalization of weighted automata to families of non-linear WAs with nice expressiveness and learnability properties,
- Expressiveness power of RNNs trained with Backprop:
  - Vanishing gradient regime,
  - Exploding gradient regime,
  - With additional components (e.g. attention mechanism etc.)

Thanks for your attention

Bibliography I

[1] Y. Chen, S. Gilroy, A. Maletti, J. May, and K. Knight, "Recurrent neural networks as weighted language recognizers," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2261–2271.

[2] S. Welleck, I. Kulikov, J. Kim, R. Y. Pang, and K. Cho, *Consistency of a recurrent language model with respect to incomplete decoding*, 2020. arXiv: `2002.02492 [cs.LG]`.

[3] B. Balle and M. Mohri, "Generalization bounds for learning weighted automata," *Theor. Comput. Sci.*, vol. 716, no. C, pp. 89–106, Mar. 2018, ISSN: 0304-3975.

Bibliography II

[4] S. Ayache, R. Eyraud, and N. Goudian, "Explaining black boxes on sequential data using weighted automata," in *ICGI*, 2018.

[5] T. Okudono, M. Waga, T. Sekiyama, and I. Hasuo, "Weighted automata extraction from recurrent neural networks via regression on state spaces," in *AAAI*, 2020.

[6] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. C. Carrasco, "Probabilistic finite-state machines-part ii," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 7, pp. 1026–1039, Jul. 2005.

[7] G. Weiss, Y. Goldberg, and E. Yahav, "Learning deterministic weighted automata with queries and counterexamples," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/d3f93e7766e8e1b7ef66dfdd9a8be93b-Paper.pdf.