Phonetic Normalization using Recurrent Neural Networks

Jasper Ginn

Universiteit Leiden

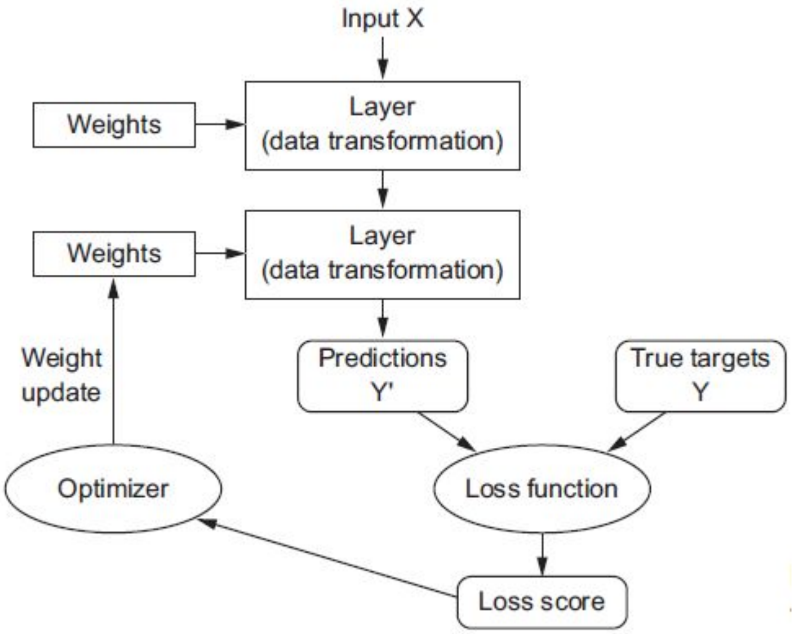
CENTRE FOR INNOVATION
Leiden University

Overview

- Neural Network Refresher
- Problem context
- Data collection
- Evaluation
- Modeling

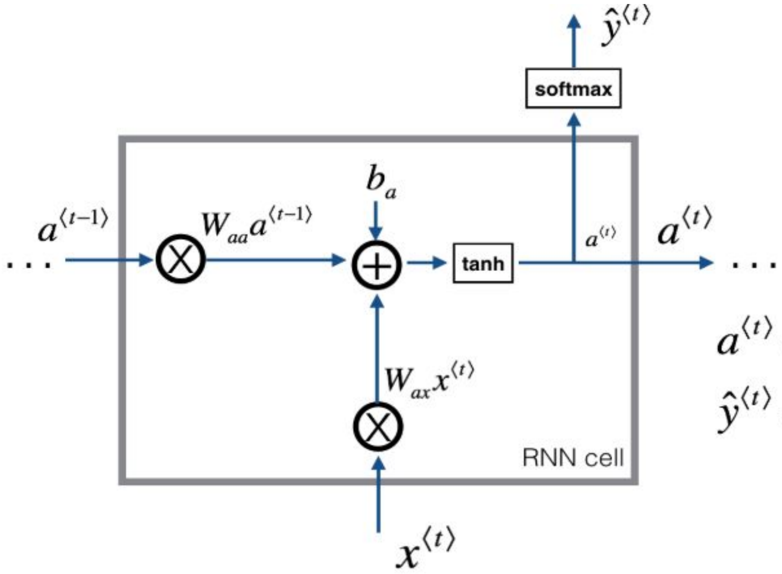
Neural Network Refresher

Vanilla Neural Network



From: Cholet, F. “Deep Learning with Python” (Manning), p.11

Recurrent Neural Network



$$a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$$

$$\hat{y}^{\langle t \rangle} = soft\max(W_{ya}a^{\langle t \rangle} + b_y)$$

Andrew Ng, "Sequence Models" (week 1) on Coursera

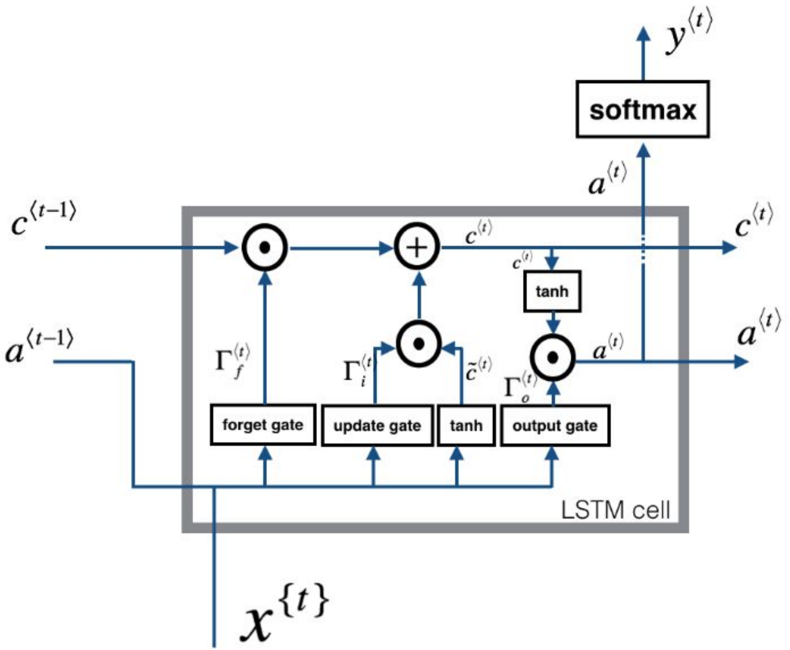
Looooooong-term dependencies

- RNN bad at capturing long-term dependencies

  “Boaz, who was, ... , went to work”

- Who does the verb refer to?
- ‘Vanishing gradients’
- Introduce memory cell

Long Short-Term Memory (LSTM)



...

$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$

$$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$

$$\tilde{c}^{\langle t \rangle} = \tanh(W_C[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_C)$$

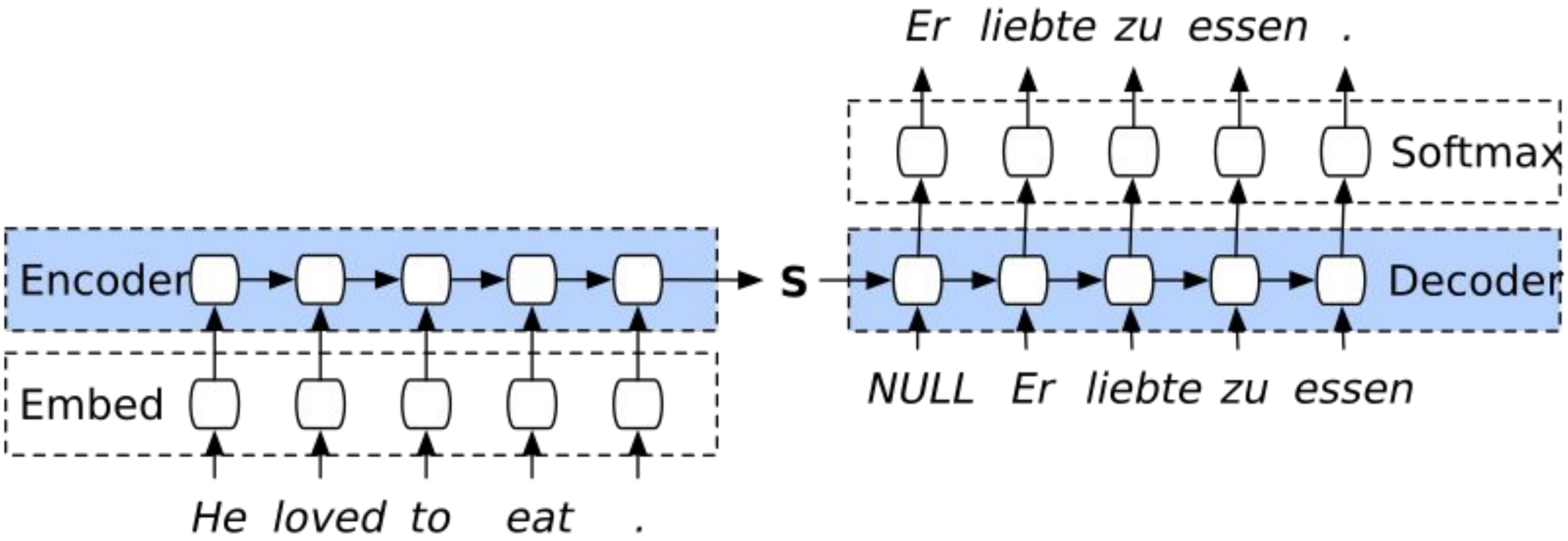
$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \circ c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \circ \tilde{c}^{\langle t \rangle}$$

$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$

$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \circ \tanh(c^{\langle t \rangle})$$

Andrew Ng, “Sequence Models” (week 1) on Coursera

Encoder / Decoder Architecture



Source: https://tinyurl.com/ydcdlbun

Teacher Forcing

Source: https://tinyurl.com/yavyy4pp

Problem Context

Goal

- Normalize written text on the ChitChat chatbot
- A lot of this text is ‘ritten’ like it is ‘spooken’
- Solution: find a way to map text to ‘spoken’ words (phonetics)

Spoken versus written words

- {‘owpen’ = ‘open’} → [OW P AH N]
- {‘bard’ = ‘barred’} → [B AA R D]
- {‘aksiom’ = ‘axiom’} → [AE K S IY AH M]
- {‘effekt’ = ‘effect’} → [IH F EH K T]
- {‘prophet’ = ‘profit’} → [P R AA F AH T]
- {‘raise’ = ‘raze’} → [R EY Z]
- …

Model Requirements

- Normalize written text on the ChitChat chatbot
- A lot of this text is ‘ritten’ like it is ‘spooken’
- Solution: find a way to map text to ‘spoken’ words (phonetics)

Goal

- Similar-sounding words should map to the same output
- Non similar-sounding words should not map to the same output

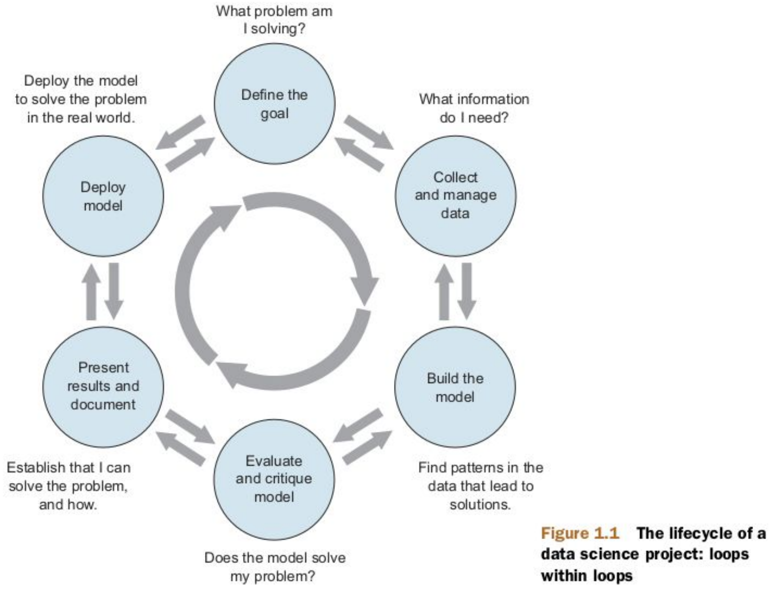
Data Collection, Models & Evaluation

Data Sources

- **Wikipedia**
  - Extracted using the wikt2pron module
  - Unreliable (many pronunciations, we don't know which belong to which dialect)
- **Carnegie Mellon CMUDict**
  - ARPAbet phonemes (see slide 14)

Measuring Performance

- BLEU / model accuracy
- **Do models map homophones and non-homophones?**
- Misspelled words from wikipedia



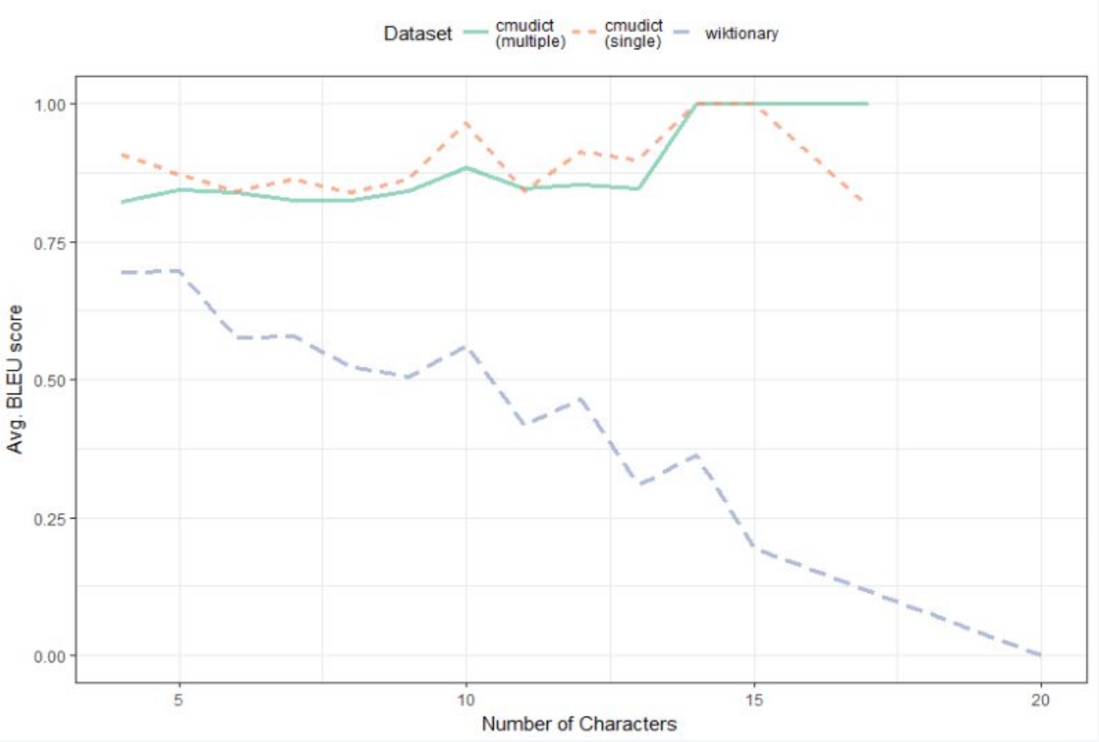
Zumel, N. and Mount, J. "Practical Data Science with R", p.6

Universiteit Leiden
CENTRE FOR INNOVATION Leiden University

Three models, two datasets

- Models
  - cmudict (single)
    - Disregard phonemes
    - [OW P AH N] ⇒ {O, W, P, A, H, N}
  - cmudict (multiple)
    - Phonemes as characters
    - [OW P AH N] ⇒ {OW, P, AH, N}
  - Wiktionary
    - Use XSAMPA-style phonetics

Model performance



	All equal	Largest subgroup	Equal to reference
cmudict (single)	77.33%	91.15%	86.94%
cmudict (multiple)	63.56%	86.58%	76.87%
wiktionary	42.11%	80.2%	51.02%

	cmudict (single)	cmudict (multiple)	wiktionary
Accuracy	0.94	0.83	0.76
Sensitivity	0.87	0.77	0.69
Specificity	0.92	0.89	0.87
F1	0.90	0.83	0.77

Model Performance (2)

- On all metrics, cmudict (single) performs best
  - Bleu, homophones, misspelled words

Takeaways

Which framework to use?

- **Keras:**
  - Abundance of documentation
  - Can be used in R/Python
- **PyTorch:**
  - Flexible + Pythonic

The fall of RNN / LSTM

Eugenio Culurciello Follow
Apr 13 · 9 min read

We fell for Recurrent neural networks (RNN), Long-short term memory (LSTM), and all their variants. **Now it is time to drop them!**