University of Zurich^UZH

Institute of Computational Linguistics

Machine Translation

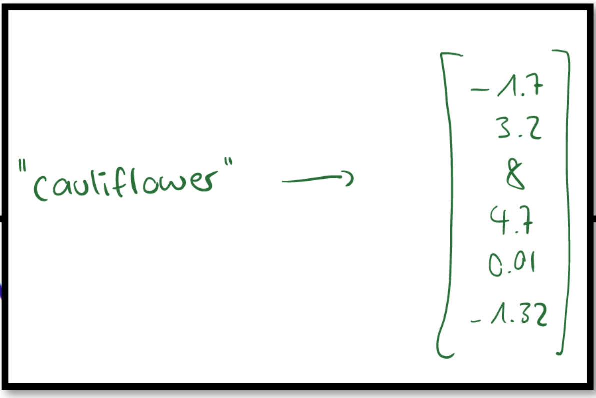
9 ~~RNNs as Language Models~~ Tensorflow

Mathias Müller
Samuel Läubli

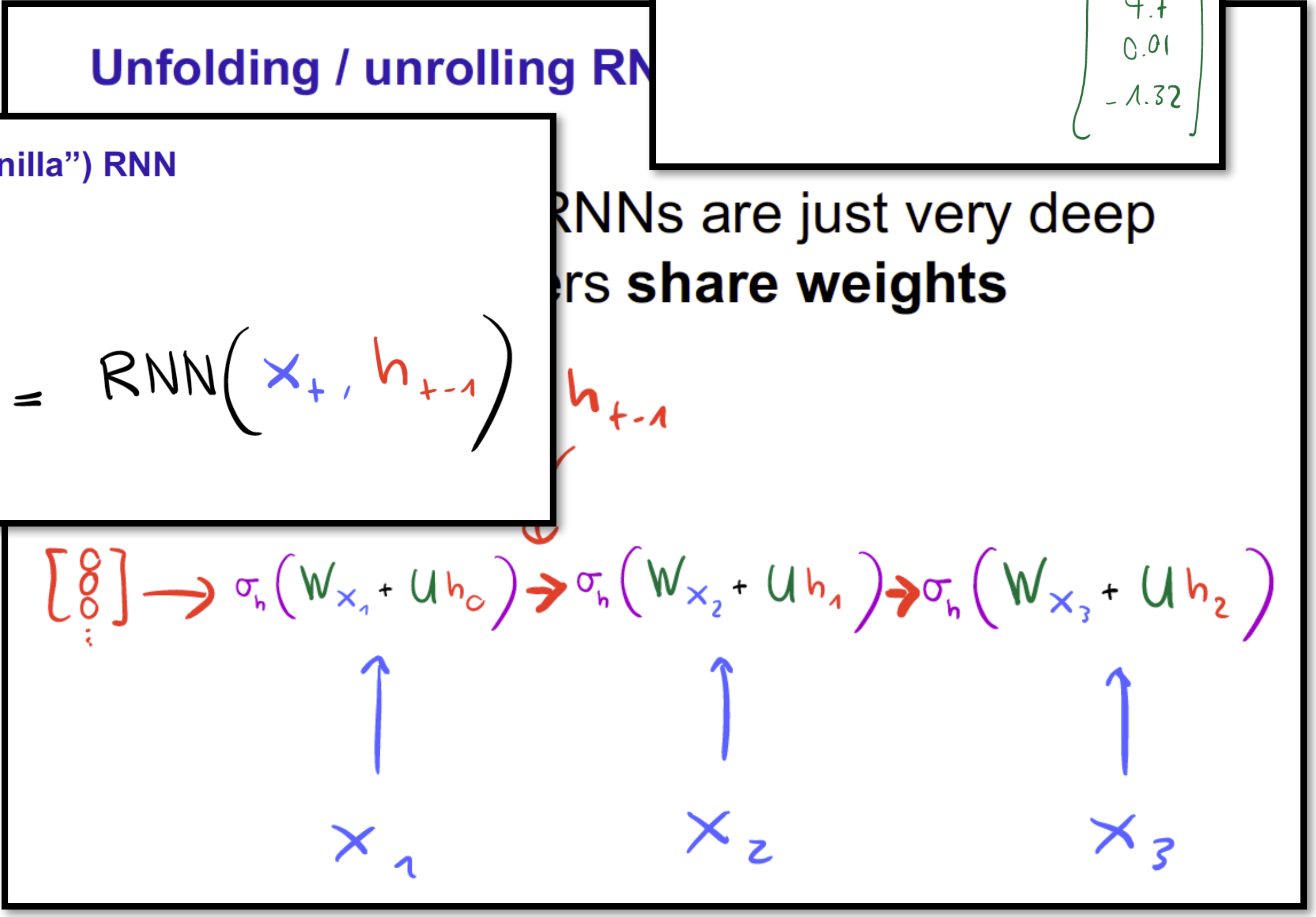
thanks!

Please download the slides from OLAT.

Last time



$$\text{"cauliflower"} \longrightarrow \begin{bmatrix} -1.7 \\ 3.2 \\ 8 \\ 4.7 \\ 0.01 \\ -1.32 \end{bmatrix}$$



Unfolding / unrolling RN

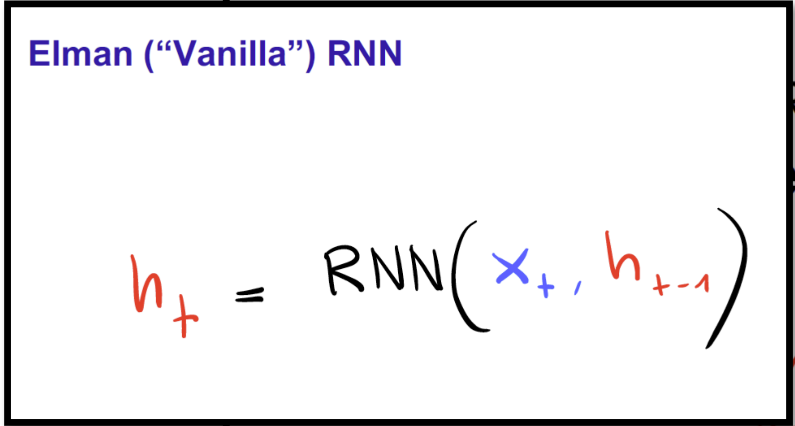
RNNs are just very deep

rs **share weights**

h_{t-1}

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \rightarrow \sigma_h(Wx_1 + Uh_0) \rightarrow \sigma_h(Wx_2 + Uh_1) \rightarrow \sigma_h(Wx_3 + Uh_2)$$

$x_1 \quad x_2 \quad x_3$

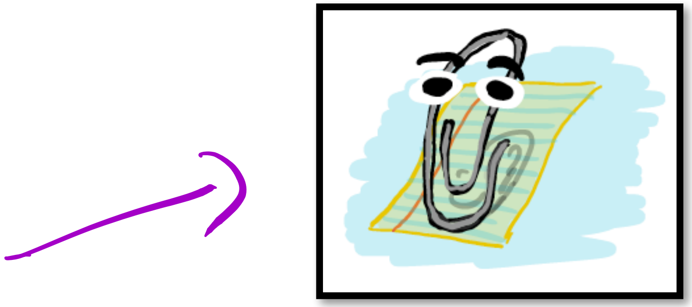


Elman ("Vanilla") RNN

$$h_t = \mathrm{RNN}(x_t, h_{t-1})$$

Topics of today

- Tensorflow



- RNN Language Models

- Romanesco: our RNNLM built with Tensorflow

University of Zurich^UZH

Institute of Computational Linguistics

Tensorflow

Tensorflow

- An open-source Deep Learning library

- Development directed by Google

- Tensorflow ≈ **Numpy++** ++++

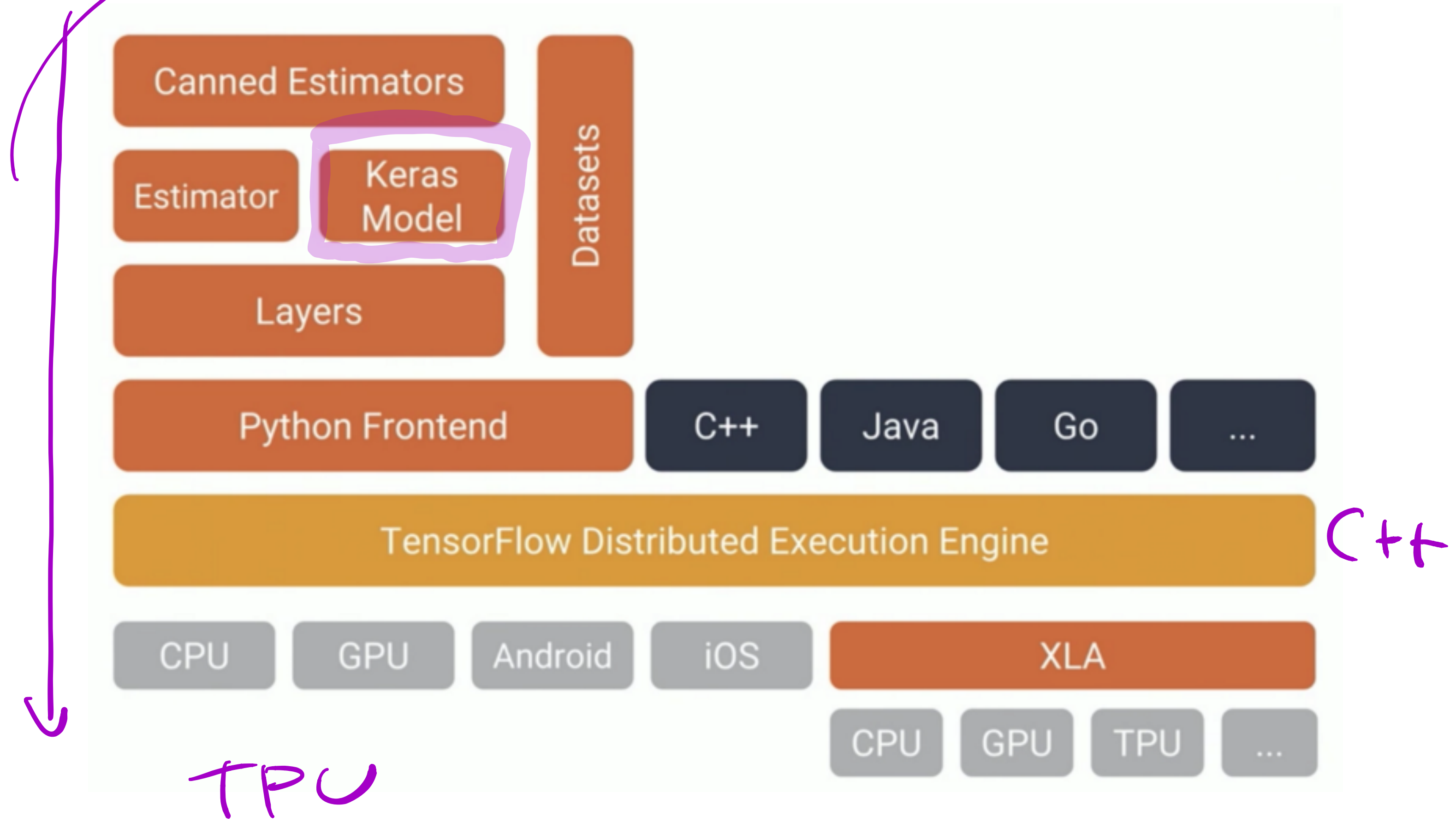
+ X platform

+ auto-differentiation

Why would we need a library like TF?

Tensorflow

+ "complexity"



Do you know Colab?

- Go to

  https://colab.research.google.com/drive/104SOHE0myVlxGqpHDPkaTG9TR4v-1apF

- Click “File”, ”Save a copy in Drive”
- Click “Connect” in the upper right corner

Summary Tensorflow Notebook

- TF has auto differentiation
- TF has **symbolic graphs**, with “variables” and “ops”
- Deferred execution vs. eager execution
- Custom coding possible on many levels
- Keras is available from within TF, high level of abstraction (this is where you should start)

Doing some research about Tensorflow

- How to use a fully-connected feed-forward layer?
- How to use cross-entropy as a loss function, a) in tf.keras, b) custom TF code?
- Are RNNs implemented in TF?

Summary Tensorflow

- Popular DL library with auto-differentiation, distributed training, bindings for many languages etc. etc.
- **Numpy++:** many numpy operations have the same name and behaviour in TF
- **Symbolic graphs**: graphs are populated with abstract variables, only later actual values are filled in

University of Zurich^UZH

Institute of Computational Linguistics

RNN Language Models

RNN Language Models

- Review: what language models do:
  - given some prefix text, predict the next word

"best car in the world: ______" $\longrightarrow$ Tesla

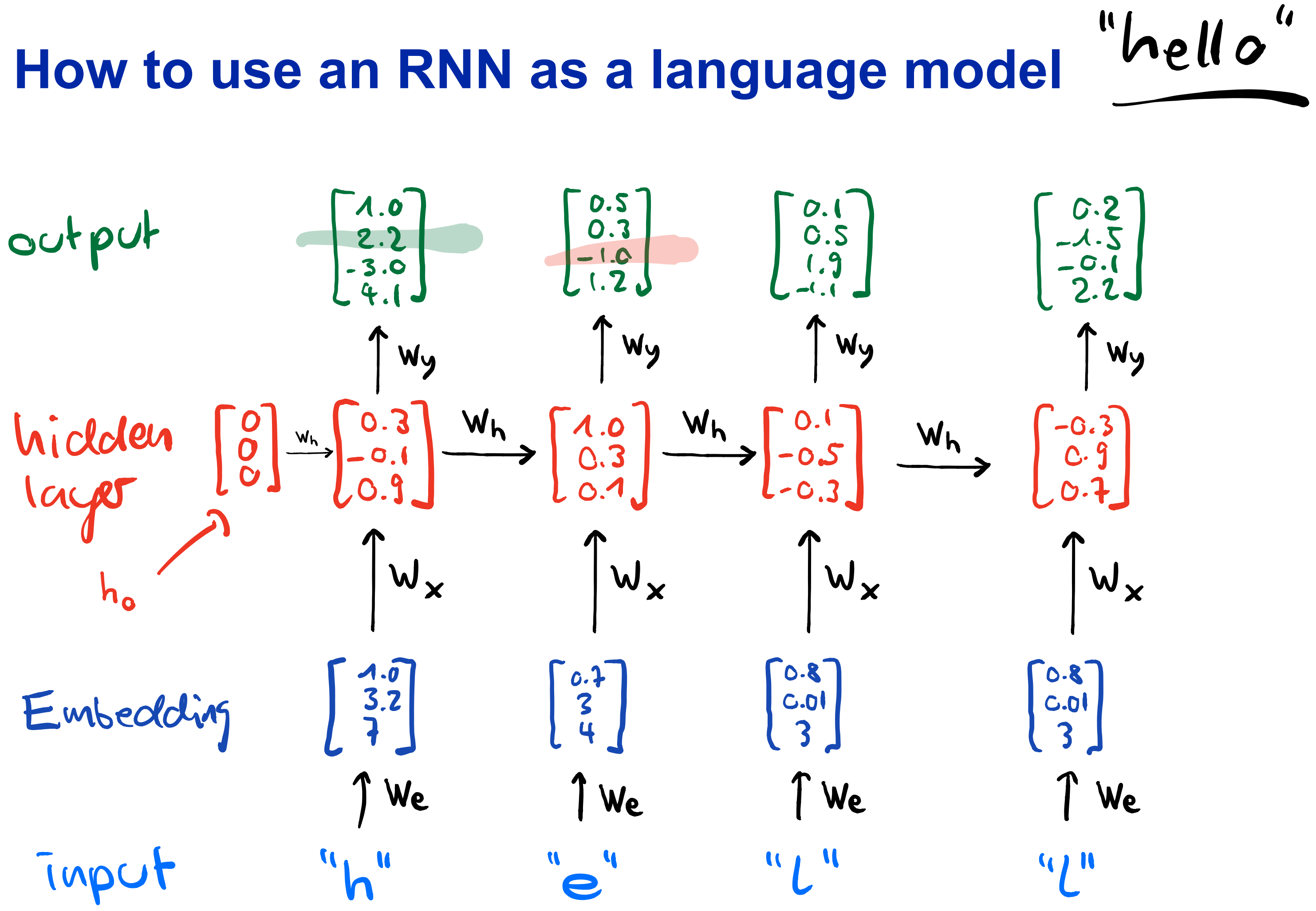
  - given some text, compute its probability

"BMWs are the best cars in the world." $\longrightarrow$ 0.000017

How to use an RNN as a language model

Vocab :	ID	one-hot
"h"	1	$\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$
"e"	2	$\boxed{\begin{bmatrix}0\\1\\0\\0\end{bmatrix}}$
"l"	3	$\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$
"o"	4	$\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

How to use an RNN as a language model



Romanesco

- Simple (= educational!) RNNLM written in Tensorflow
- Main author: Samuel Läubli
- Serves as basis for Exercise 4
- https://github.com/ZurichNLP/romanesco

Supported actions

- Train a model

```
romanesco train data.txt
```

- Score text

```
cat input.txt | romanesco score
```

- Generate text

```
romanesco sample 50
```

Vocabulary

'<unk>'

```
>>> from romanesco.vocab import Vocabulary

>>> my_vocab = Vocabulary()
>>> my_vocab.build('train.txt', max_size=10000) # 10'000

>>> my_vocab.size
7982

>>> my_vocab.get_id('the')
6
```

$$\longrightarrow \begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\\\vdots\end{bmatrix}$$

```
>>> my_vocab.get_words([9, 832, 6, 1036, 7, 5607, 12])
['I', 'love', 'the', 'people', 'of', 'Iowa', '.']
```

Reading text: steps

shakerpeare.txt

1) • Replace ‘\n’ with ‘<eos>’
2) • Split at whitespaces to make entire input a giant list of symbols

"the" 6

3) • List of symbols $\longrightarrow$ list of vocab IDs
• List of vocab IDs $\longrightarrow$ Batches

• Each batch contains **batch_size** sequences of length **NUM_STEPS**

[7, 5, 3, 4, 6, 8]

batch_size = 2
seq_length = 3

"I love the people

[[7, 5, 3]

[illegible]

[[4,6,8]]

Creating batches from list of vocab IDs

[7,5,3...] 3 2

```
def iterate(raw_data, batch_size: int, num_steps: int):

    data_len = len(raw_data)
    num_batches = data_len // batch_size
15
    data = raw_data[0 : batch_size * num_batches]
    data = np.reshape(data, [batch_size, num_batches])
                                    3      ,  5

    num_batches_in_epoch = (num_batches - 1) // num_steps

    for i in range(num_batches_in_epoch):
        s = i * num_steps # start
        e = s + num_steps # end
        yield data[:, s : e], data[:, s + 1 : e + 1]
```

Computation graph definition

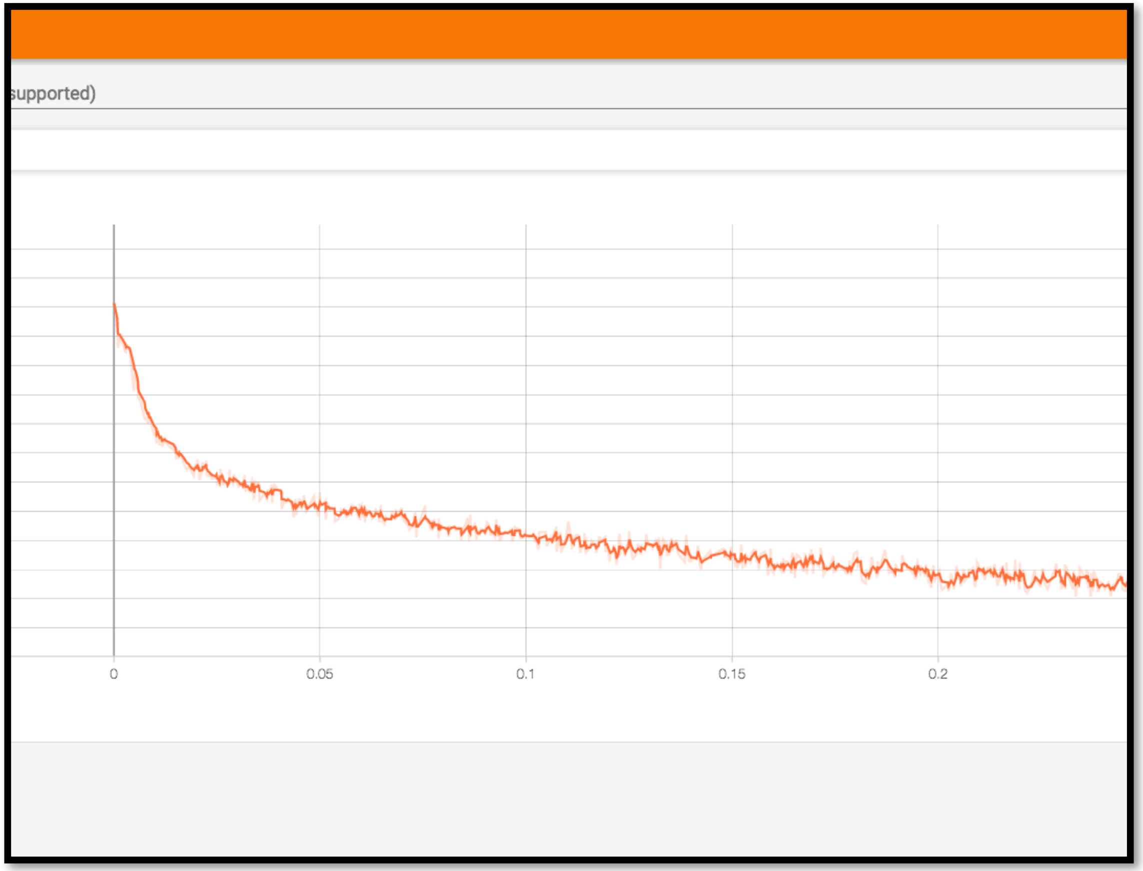
https://github.com/ZurichNLP/romanesco/blob/master/romanesco/compgraph.py

Training

↓

https://github.com/ZurichNLP/romanesco/blob/master/romanesco/train.py

Tensorboard in your local browser



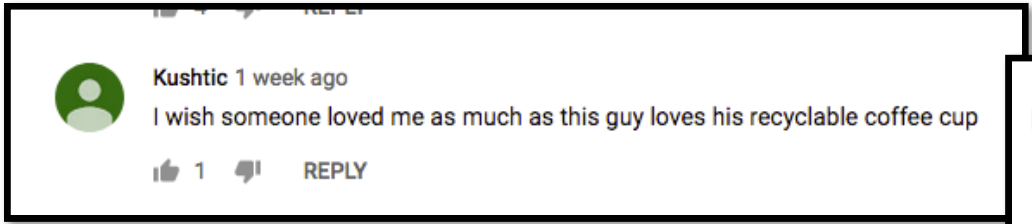
next: let's try romanesco!

Summary

- **Tensorflow**: DL Library that greatly facilitates training and using NN models
- **RNNLMs**: Recurrent NNs can be used as language models
- **Romanesco**: educational example of an RNNLM written in Tensorflow

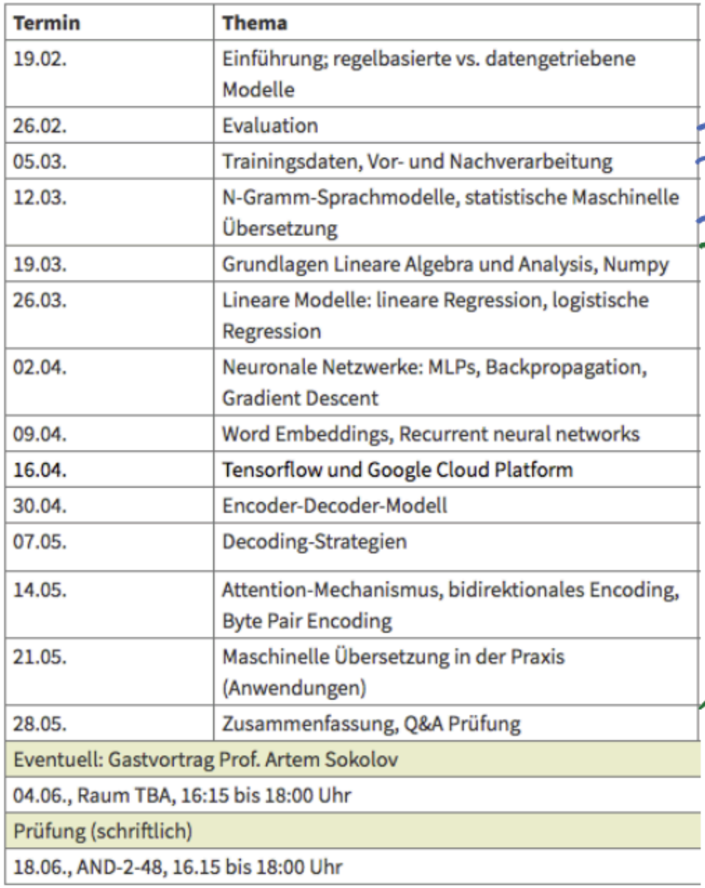
Further Reading / Links

- **Colab**, run Jupyter notebooks in the cloud with 1 free GPU and tensorflow pre-installed: https://colab.research.google.com/
- Romanesco: https://github.com/ZurichNLP/romanesco
- Good basic tf.keras tutorials: https://github.com/tensorflow/docs/blob/master/site/en/tutorials/keras/index.md
- Out of hundreds, one useful TF tutorial / code collection: https://github.com/aymericdamien/TensorFlow-Examples
- Good tutorial with a likeable presenter, **Josh Gordon**: https://www.youtube.com/watch?v=tYYVSEHq-io


Kushtic 1 week ago
I wish someone loved me as much as this guy loves his recyclable coffee cup

Carter Ellsworth 7 months ago
This guy likes to smile

Next time



Termin	Thema
19.02.	Einführung; regelbasierte vs. datengetriebene Modelle
26.02.	Evaluation
05.03.	Trainingsdaten, Vor- und Nachverarbeitung
12.03.	N-Gramm-Sprachmodelle, statistische Maschinelle Übersetzung
19.03.	Grundlagen Lineare Algebra und Analysis, Numpy
26.03.	Lineare Modelle: lineare Regression, logistische Regression
02.04.	Neuronale Netzwerke: MLPs, Backpropagation, Gradient Descent
09.04.	Word Embeddings, Recurrent neural networks
16.04.	Tensorflow und Google Cloud Platform
30.04.	Encoder-Decoder-Modell
07.05.	Decoding-Strategien
14.05.	Attention-Mechanismus, bidirektionales Encoding, Byte Pair Encoding
21.05.	Maschinelle Übersetzung in der Praxis (Anwendungen)
28.05.	Zusammenfassung, Q&A Prüfung
Eventuell: Gastvortrag Prof. Artem Sokolov	
04.06., Raum TBA, 16:15 bis 18:00 Uhr	
Prüfung (schriftlich)	
18.06., AND-2-48, 16.15 bis 18:00 Uhr	

EVALUATION

TRAINING DATA

SMT

NMT

this is kinda important