Deep Learning for NLP

Universität Bielefeld

Lecture 10 – Seq2Seq, Text Generation, Evaluation Metrics

- **Dr. Steffen Eger**

- Natural Language Learning Group (NLLG)

Last lectures

- Learning to represent words
- ConvNets for sentence classification
- RNNs for sequence tagging

Today

1. Auto-encoders
2. Encoder-Decoder architectures
3. Evaluation Metrics
4. Text Generation

- Lecture based on:
  - Chapter 14, www.deeplearningbook.org
  - Stanford tutorials:
    https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf
    http://ufldl.stanford.edu/wiki/index.php/Stacked_Autoencoders
  - NLP publications (as referenced)

Auto-encoders

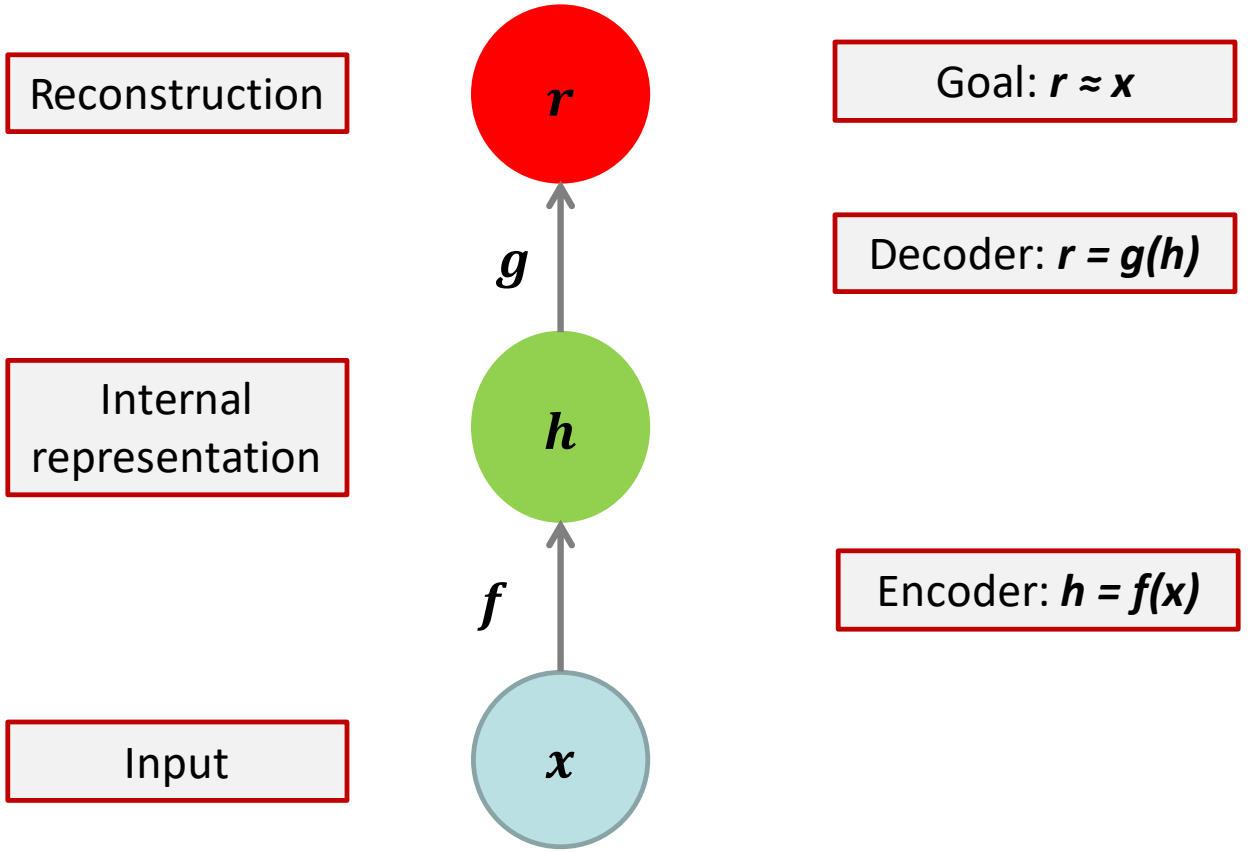
- A neural network that is trained to attempt to copy its input to its output
  - → I.e., learn the identity function, $F(\boldsymbol{x}) = \boldsymbol{x}$
  - → That is too easy (under which conditions?)

- Restrict the auto-encoder so that it can only copy approximately
  → need a bottleneck in order to learn successfully

- Traditionally used for dimensionality reduction or representation learning

Architecture

- Simple feed-forward network / MLP



Learning

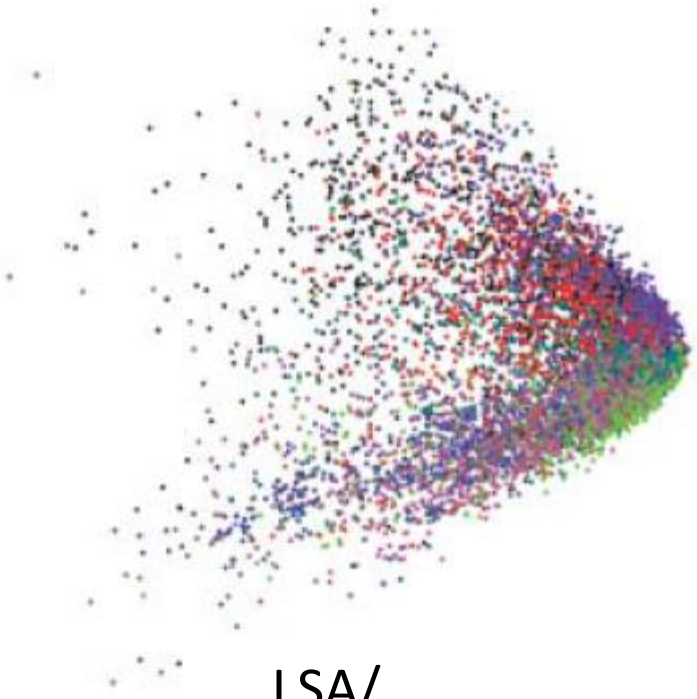
- An autoencoder may be seen as situated in the **unsupervised** learning scenario
- Alternatively: supervised with self-defined auxiliary task (similarly to how we learn word embeddings)
  - The training data consists of pairs $(\boldsymbol{x}, \boldsymbol{x})$ or variants thereof

Undercomplete auto-encoder

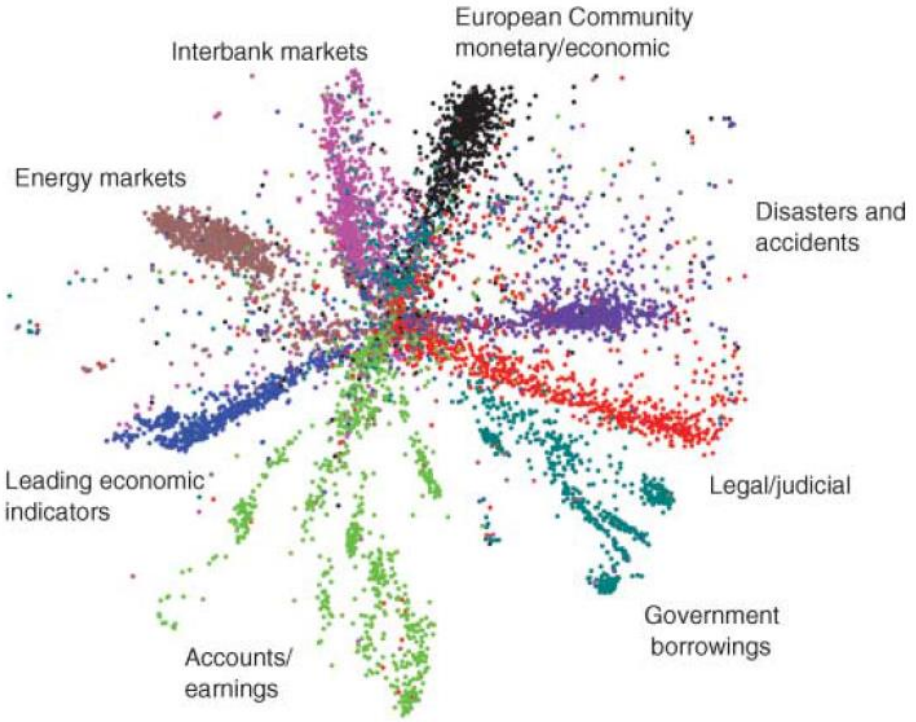
- **Undercomplete**: The hidden dimensionality is smaller than the input dimensionality
- With linear activation functions and square error loss, an undercomplete autoencoder learns [the same as] PCA.
- Auto-encoders with non-linear encoder and decoder functions can be seen as a (more powerful) generalization of PCA.

Auto-encoders vs. PCA

- Articles from the Reuter corpus were mapped to a 2000 dimensional vector, using the most common word stems



LSA/
PCA



Deep Autoencoder

Hinton et al., Reducing the Dimensionality of Data with Neural Networks

Denoising auto-encoders

- Add some random noise to the input

$$\tilde{\boldsymbol{x}} \leftarrow \mathbf{noise}(\boldsymbol{x})$$

- The auto-encoder should learn to “remove the noise”, i.e., reconstruct input from a corrupted version
  - This forces the network to actually learn something even when hidden dimensionality is $\geq$ input dimensionality

DISCUSS

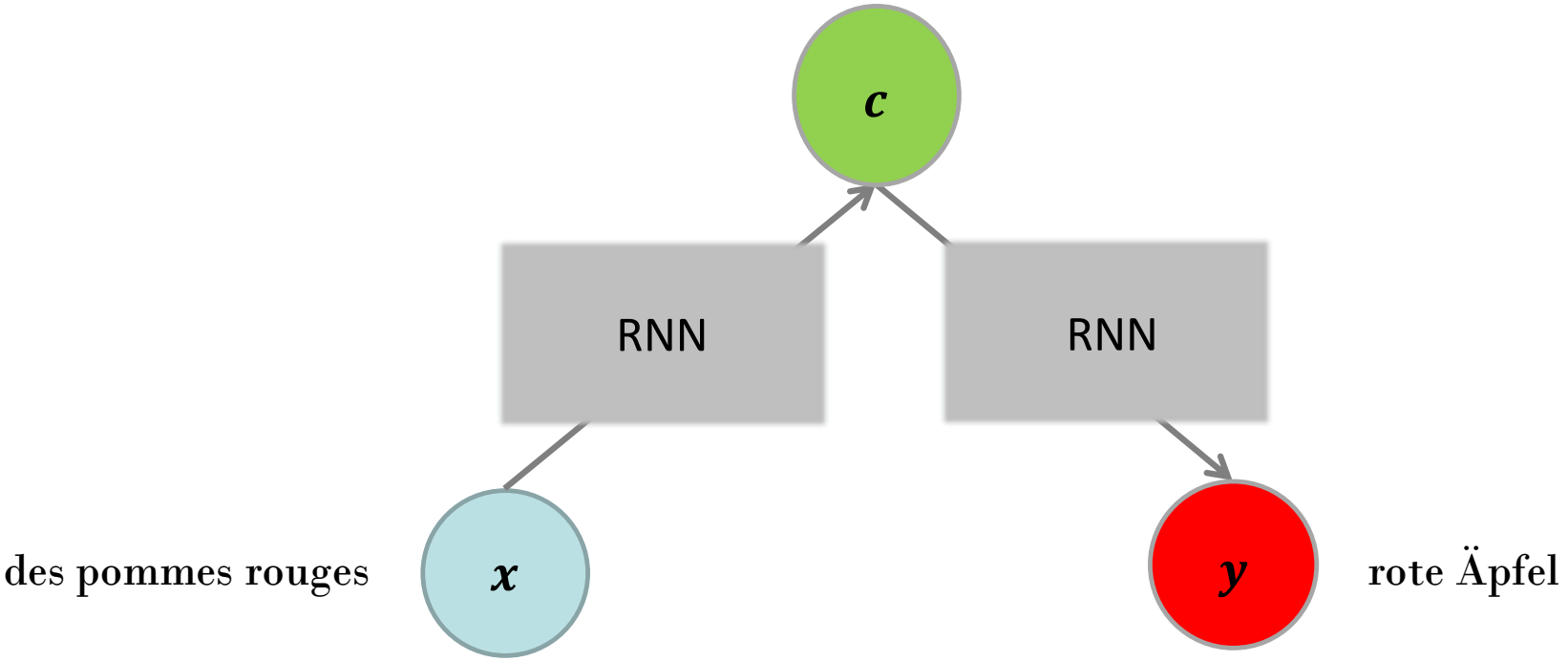
Assume your input vectors are 1-hot of size 2^n and your hidden layer has size n. What efficient representation could the auto-encoder learn?

Agenda

1. Autoencoder
2. **Encoder-Decoder architectures**
   - **Machine translation (MT)**
     - Cho, K., Gulcehre, B. V. M. C., Bahdanau, D., Schwenk, F. B. H., & Bengio, Y. (2014): Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. EMNLP.
     - Sutskever et al. 2014, Sequence-to-Sequence Learning with Neural Nets
     - Bahdanau, Dzmitry, Kyunghyun Cho, Yoshua Bengio (2015): Neural Machine Translation by Jointly Learning to Align and Translate. ICLR. http://arxiv.org/abs/1409.0473
3. Extensions & Further applications

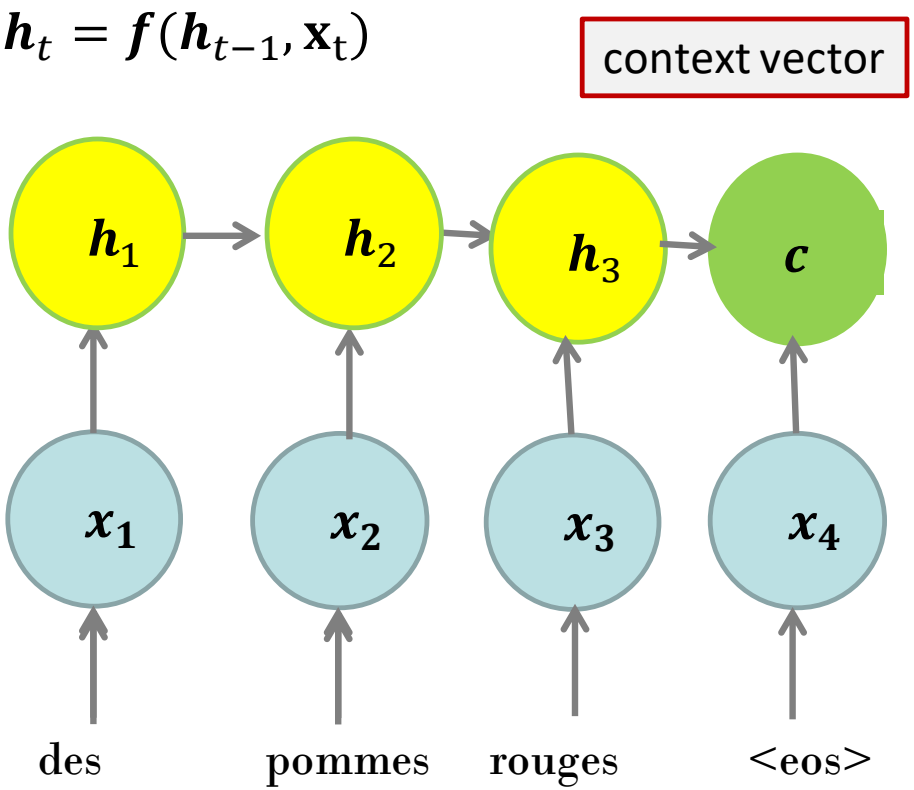
Encoder-Decoder architecture

- Instead of feedforward networks we can also have more complex architectures for the encoder and the decoder e.g. recurrent neural networks or LSTMs



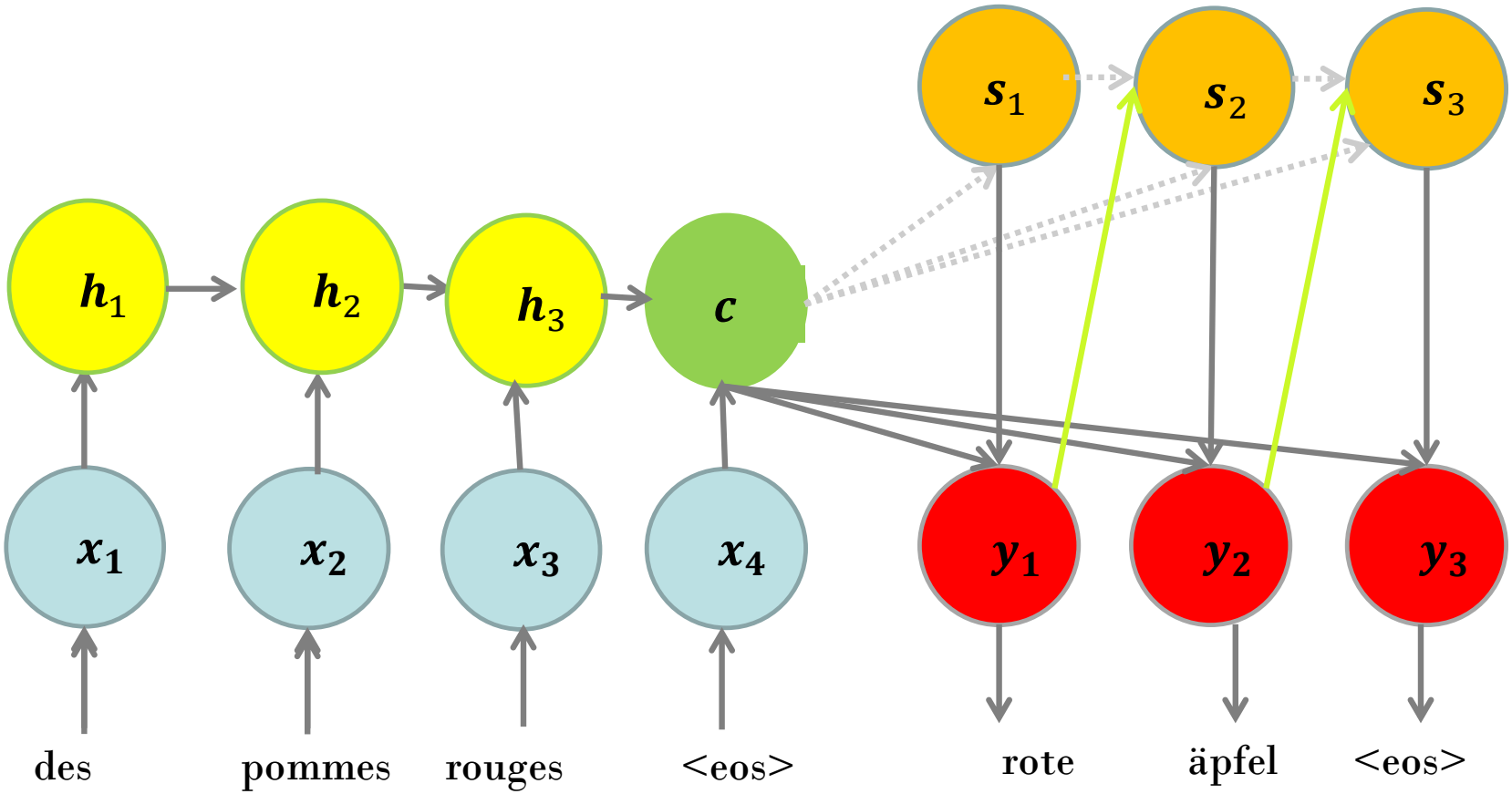
RNN Encoder

- Read in the whole input sequence.
- Learn a context vector $\boldsymbol{c}$
- $\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{h}_{t-1}, \mathbf{x}_t)$



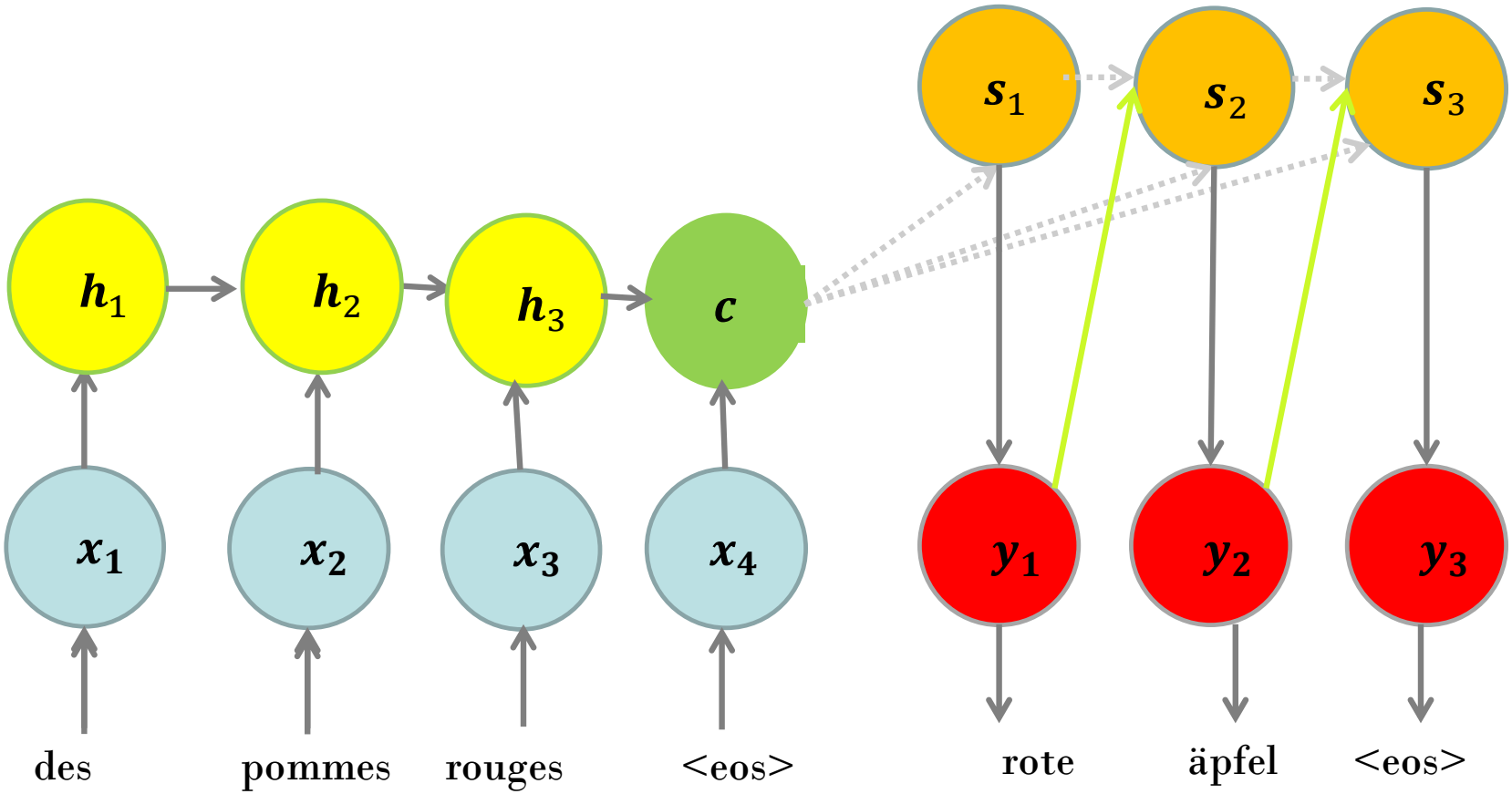
RNN Encoder-Decoder architecture

- Predict one output word at a time.



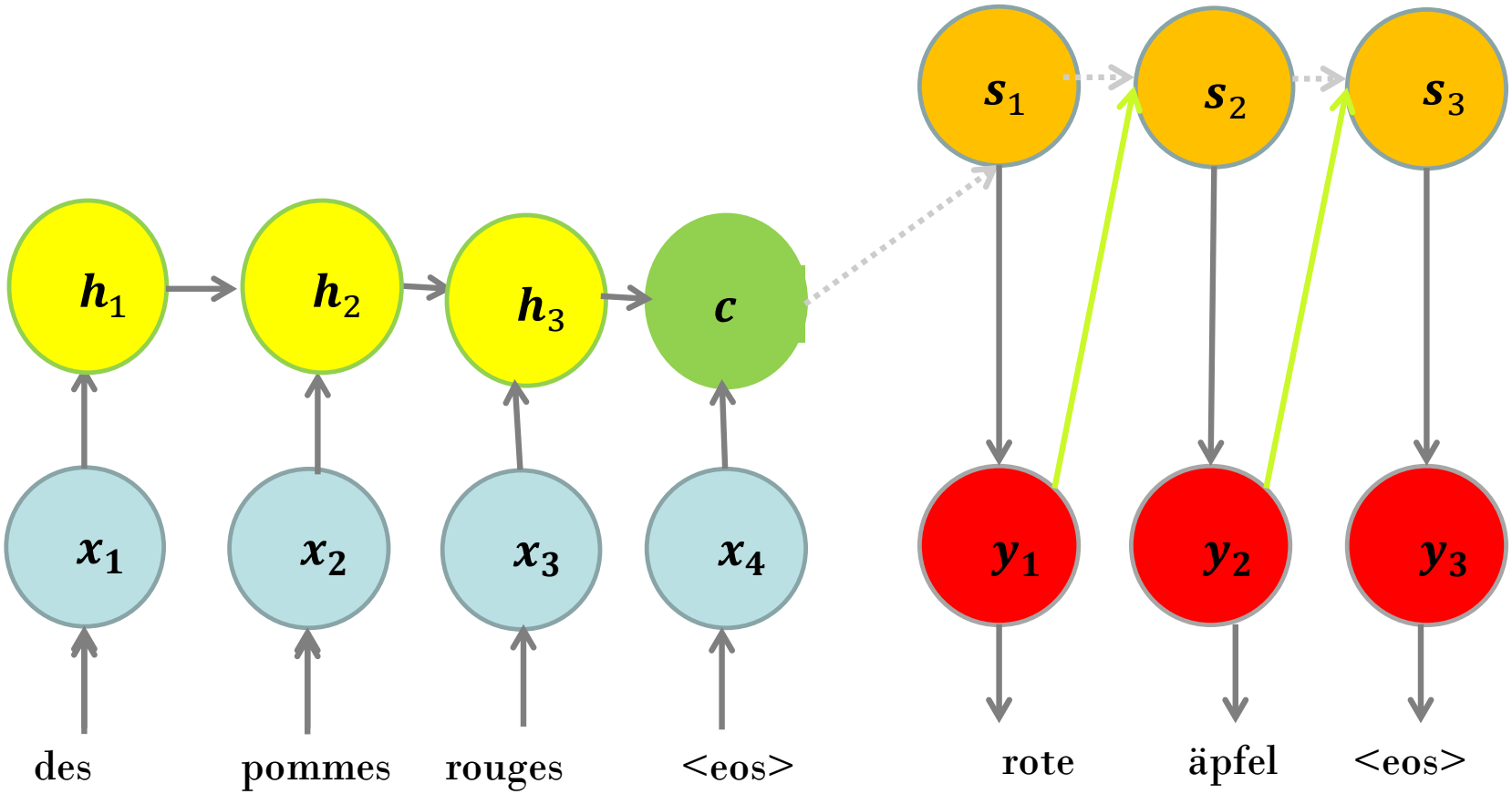
RNN Encoder-Decoder architecture

- Different variants



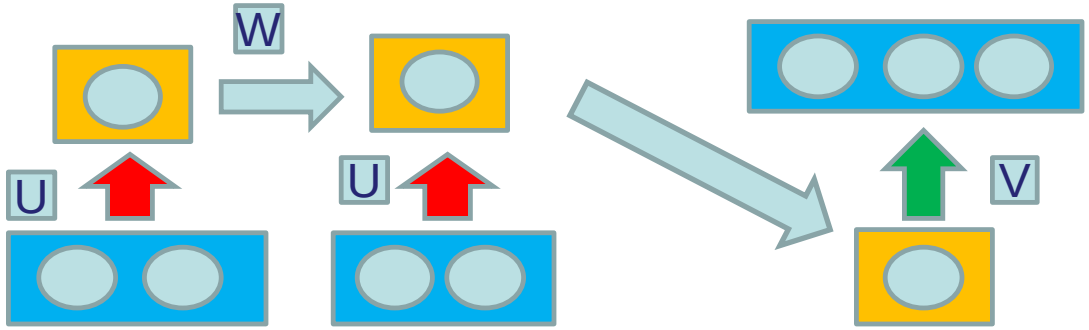
RNN Encoder-Decoder architecture

- Different variants



EXERCISE

Assume we got 2 input symbols (a,b) and the following architecture:



Assume that:

embedding of a = (1,1)

embedding of b = (-1,0.5)

U = [1/4; 2/3]

W = [0.8]

V = [1;0;-1]

Calculate: **Output layer activation for input b,a**

What else do you need?

Quiz

What‘s the result before softmax?

A: (5/6, 1, -1/3)
B: (118/120, 0, -118/120)
C: (1,0,-1)
D: (99/40, 3, 70/75)
E: (0,0,0)
F: none of the above

Encoder-Decoder architecture

- Summary of basic idea:
  - Two RNNs (could be other architectures as well, like Transformer)
  - The one encodes the input sentence (can think of it as a sentence embedding)
  - The other decodes the “sentence embedding” (i.e., the context vector retrieved by the encoder)
    - The decoder is like a language model
    - It typically chooses the **max** element at each time step
      - Rather than sampling probabilistically → since we want the best output given the fixed input

Encoder-Decoder architecture

Problems

- Long range dependencies

→ Problems especially with long input sequences

- Context vector ***c*** has a fixed length
- Input size is variable

→ Problems especially with long input sequences

Encoder-Decoder architecture

- Sutskever et al. 2014, Sequence-to-Sequence Learning with Neural Nets

- Their core contribution is to **reverse** the *input sequence* during training and testing

  - The cat sat on the mat → mat the on sat cat The

Encoder-Decoder architecture

- Sutskever et al. 2014, Sequence-to-Sequence Learning with Neural Nets
- Their core contribution is to **reverse** the *input sequence* during training and testing
  - The cat sat on the mat → mat the on sat cat The
- On average, the dependencies have the same length whether we reverse or not

Encoder-Decoder architecture

- Sutskever et al. 2014, Sequence-to-Sequence Learning with Neural Nets

- The cat sat on the mat vs. Die Katze saß auf der Matte
  - Dist(The,Die) = 6
  - Dist(cat,Katze) = 6 ….

- mat the on sat cat The vs. Die Katze saß auf der Matte
  - Dist(The,Die) = 1
  - Dist(cat,Katze) = 2 ….
  - Dist(mat,Matte) = 12

Encoder-Decoder architecture

- Sutskever et al. 2014, Sequence-to-Sequence Learning with Neural Nets

- While the effect is not entirely clear, Sutskever et al. speculate that

  - Minimal time lag is now greatly reduce (not the average one)

  - This way, it's easier to "establish communication" between source and target sentences

Encoder-Decoder architecture

- Sutskever et al. 2014, Sequence-to-Sequence Learning with Neural Nets

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

Problems

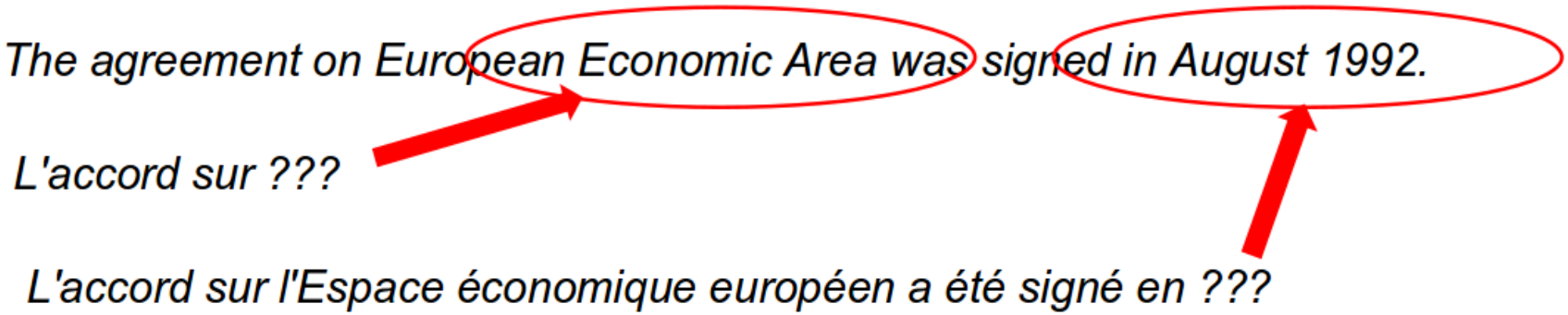
- A major problem with the solutions so far is the global vector $\boldsymbol{c}$:
  - All the input is encoded in a single (big) vector that summarizes the whole sentence
  - However, when we translate a word, it typically only depends on a *local* neighborhood in the source sentence, not on the complete source sentence

Die Katze saß auf der Matte: depends only on cat

Die Katze saß auf der Matte: depends only on The and cat

Problems & an Idea

- Only pay attention to the elements of the input that are relevant for producing the **current** output.



Problems & an Idea

- A major problem with all presented solutions so far is the global vector $\boldsymbol{c}$:
  - Ideally, we would make use of some sort of **alignment** (based on attention) information during training and testing
  - That's the idea of Bahdanau, Dzmitry, Kyunghyun Cho, Yoshua Bengio (2015): Neural Machine Translation by Jointly Learning to Align and Translate. ICLR. http://arxiv.org/abs/1409.0473

> *"The decoder decides parts of the source sentence to pay attention to. By letting the decoder have an attention mechanism, we relieve the encoder from the burden of having to encode all information in the source sentence into a fixed- length vector."*

http://www.iclr.cc/lib/exe/fetch.php?media=iclr2015:bahdanau-iclr2015.pdf

Joint Alignment and Translation

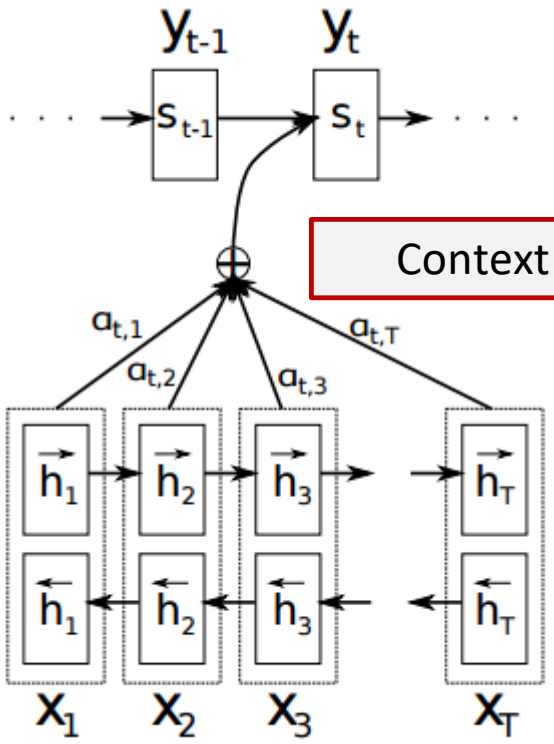
$$\boldsymbol{y}_t = \mathbf{g}(\boldsymbol{s}_t, \boldsymbol{y}_{t-1})$$

$$\boldsymbol{s}_t = \boldsymbol{f}(\boldsymbol{s}_{t-1}, \boldsymbol{y}_{t-1}, \boldsymbol{c}_t)$$

"It should be noted that unlike the existing encoder–decoder approach, here the probability is conditioned on a distinct context vector $\boldsymbol{c}_t$ for each target word y_t."

Learning the context vector

Generate t-th target word y_t give source sentence (x_1, …, x_T)



Context vector $\boldsymbol{c}_t$

Concat the two hidden states of the bidirectional RNN

http://arxiv.org/pdf/1409.0473v7.pdf

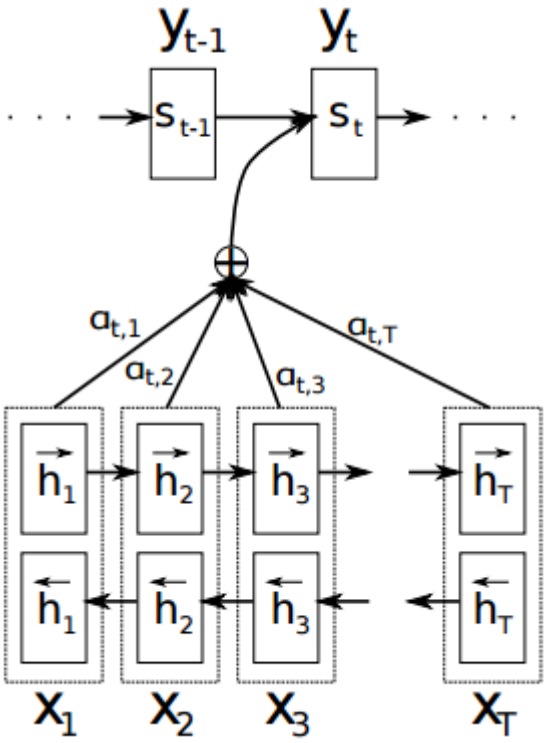
Context vector:

$$\boldsymbol{c}_t = \sum_{j=1}^{T} \alpha_{tj} \boldsymbol{h}_j$$

The context vector varies depending on the relation α_{tj} between the current output y_t and the input words x_j.

Learning the context vector with attention

What is α_{tj}?
Has become popular as "attention".



http://arxiv.org/pdf/1409.0473v7.pdf

Context vector:

$$\boldsymbol{c}_t = \sum_{j=1}^{T} \alpha_{tj} \boldsymbol{h}_j$$

How is the attention vector computed?

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_k \exp(e_{tk})}$$

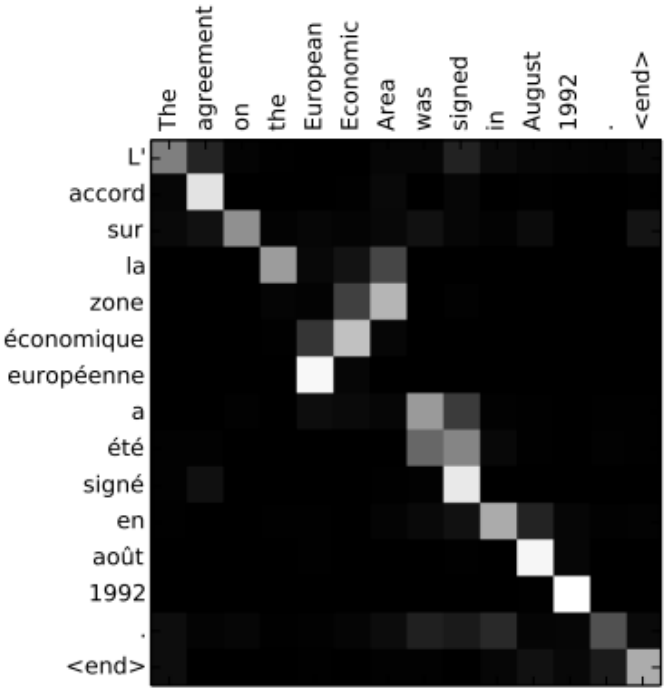
- where $e_{tj} = a(\boldsymbol{s}_{t-1}, \boldsymbol{h}_j)$ indicates how likely target word t (which we want to produce at the current time step) is aligned to input word j
- "We parametrize the alignment model a as a feedforward network which is **jointly** trained with the other components of the proposed system"
- α_{tj} (and e_{tj}) are also termed **soft alignments**

Illustration: Soft alignments

- Soft alignments: probability distribution over tokens that a current token aligns to
- Hard alignments: “degenerate prob. distribution” with 0/1 values (NB may still align to **many** tokens)

The English words

- **Hard alignment** example:
  - L’ → (1,0,0,0,0,…)
  - été → (0,0,0,1,1,0,0,….)



Soft alignment

Agenda

1. Autoencoder
2. Encoder-Decoder architectures
   - Noisy-channel model
   - Machine translation (MT)
     - Cho, K., Gulcehre, B. V. M. C., Bahdanau, D., Schwenk, F. B. H., & Bengio, Y. (2014): Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. EMNLP.
     - Sutskever et al. 2014, Sequence-to-Sequence Learning with Neural Nets
     - Bahdanau, Dzmitry, Kyunghyun Cho, Yoshua Bengio (2015): Neural Machine Translation by Jointly Learning to Align and Translate. ICLR. http://arxiv.org/abs/1409.0473
3. **Extensions & Further applications**

Seq2Seq tasks

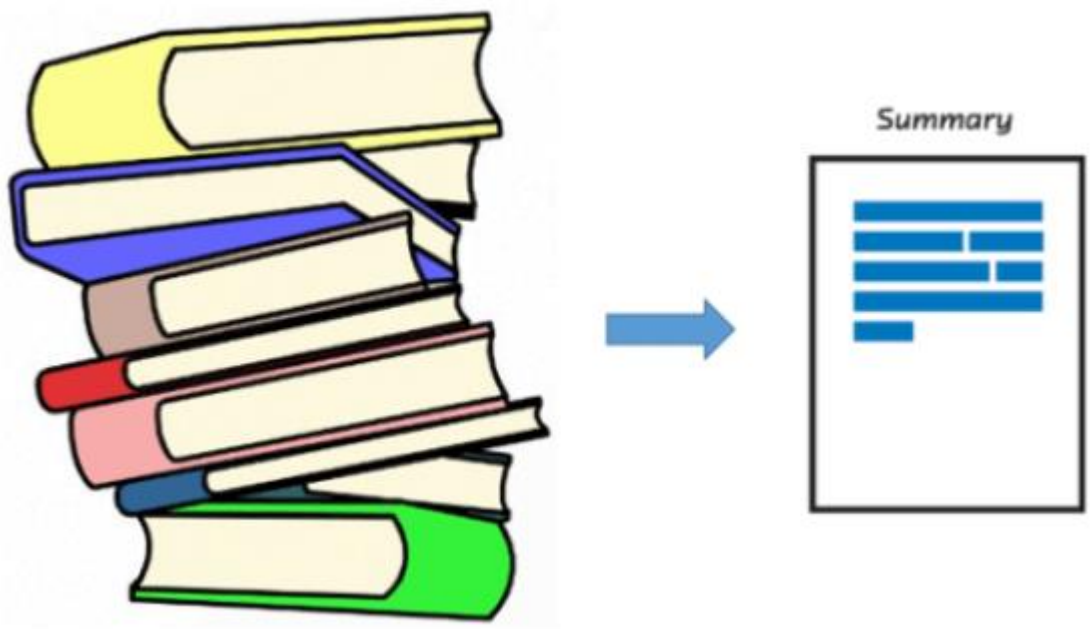
- Encoder-Decoder models are extremely general
  - Cannot only translate from English to German
  - But also many other “Sequence-to-Sequence” tasks (Seq2Seq) such as
    - Lemmatization, Grapheme-to-Phoneme conversion, Spelling correction, Summarization, etc.
  - Of course, you could also POS tag with them, etc.

Seq2Seq tasks

TASK	EXAMPLE
Lemmatization	g e s p i e l t → s p i e l e n
G2P	s c h u h → S u:
Spelling correction	I _ l v o e _ u → I _ l o v e _ y o u

Sequence-to-sequence tasks: Summarization

- Typically on document-level



DISCUSS

Which other tasks can you think of that fit into the Seq2Seq paradigm? (Name 3)

Encoder-Decoder Models from Transformers

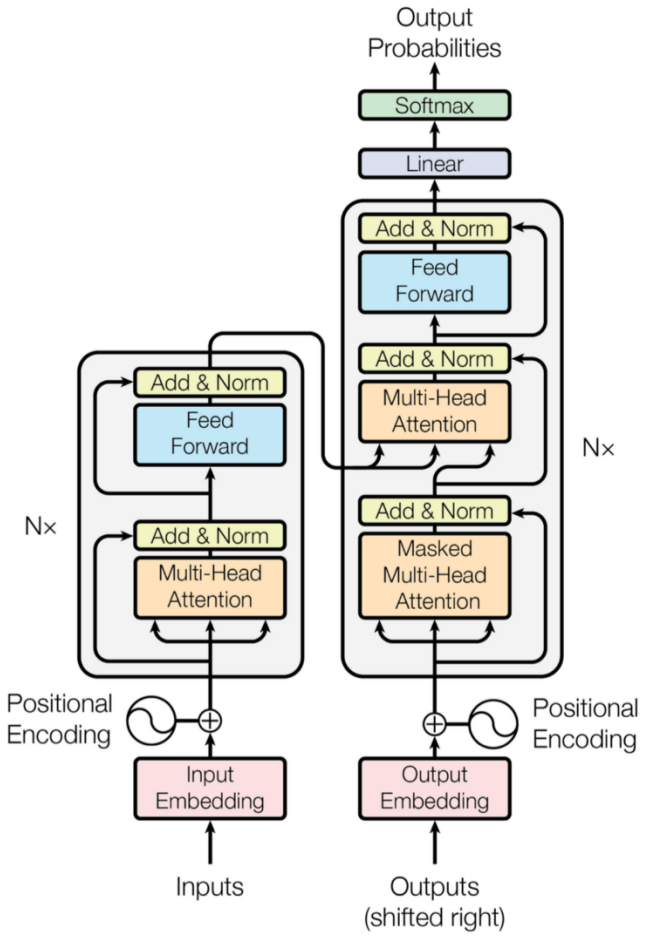
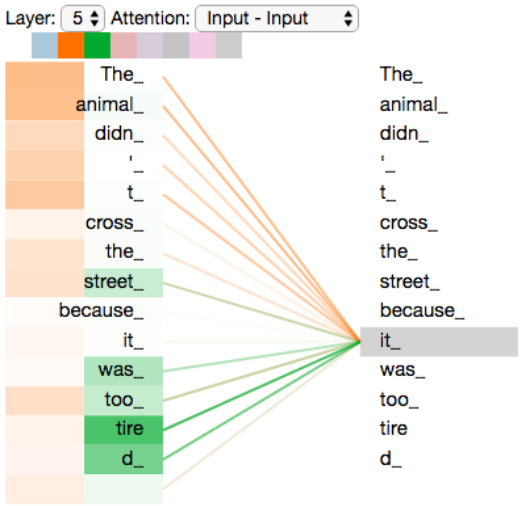


Figure 1: The Transformer - model architecture.

- No recurrence
- Allow efficient parallelization
- Can be pre-trained on a lot of data
- Better handling of long-range dependencies
- Use self-attention
- More details: see the accompanying video



Evaluation Metrics

How to evaluate the quality of text generation systems?

(see google slides)

Text Generation Systems

- Generating text with neural models
- In the following, we present popular „decoder-only“ models
  - GPT (2018)
  - GPT-2 (2019)
  - GPT-3 (2020)
  - PaLM (2022)

- All these LMs are „left-to-right“ LMs, they model
  - $p(y_t \mid y_1, \dots, y_{t-1})$

- Even though these models do not have an encoder, they can still be used for Seq2Seq tasks

GPT

- Published in 2018, by OpenAI
- Improving Language Understanding by Generative Pre-Training

- Uses left-to-right language modeling
- Pre-training on massive amounts of data
- Fine-tuning on task specific data (e.g. sentiment)

- Main message:
  - Pre-training on massive amounts of data helps
  - Only little task-specific fine-tuning is necessary
  - Similar story as BERT

GPT-2

- Published in 2019, by OpenAI
- Language Models are Unsupervised Multitask Learners

- Main message:
  - Models learn tasks themselves from data
  - No need for fine-tuning
  - More data helps more

GPT-2

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”

“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

GPT-3

- Published in 2020, by OpenAI
- Language Models are Few-Shot Learners

- Main message:
  - Language Models can learn like humans
  - You show them a few examples and they will learn even **new tasks**
  - No need for task-specific fine-tuning at all

GPT-3

Zero-shot

The model predicts the answer given only a natu
description of the task. No gradient updates are

```
1  Translate English to French:
2  cheese =>
```

One-shot

In addition to the task description, the model see
example of the task. No gradient updates are pe

```
1  Translate English to French:
2  sea otter => loutre de mer
3  cheese =>
```

task description

example

prompt

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:
2  sea otter => loutre de mer
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese =>
```

task description

examples

prompt

GPT-3

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

Figure 3.14: The GPT-3 generated news article that humans had the greatest difficulty distinguishing from a human written article (accuracy: 12%).

PaLM

- Published in 2022, by Google
- PaLM: Scaling Language Modeling with Pathways

- Main message:
  - More data helps even more
  - Trained on 6000+ TPUs
  - „Breakthrough performances“ and discontinuous improvements
  - Performance near human or better on 150+ (language understanding) tasks
    - Including commonsense reasoning, mathematical reasoning, code completion, and natural language understanding

PaLM

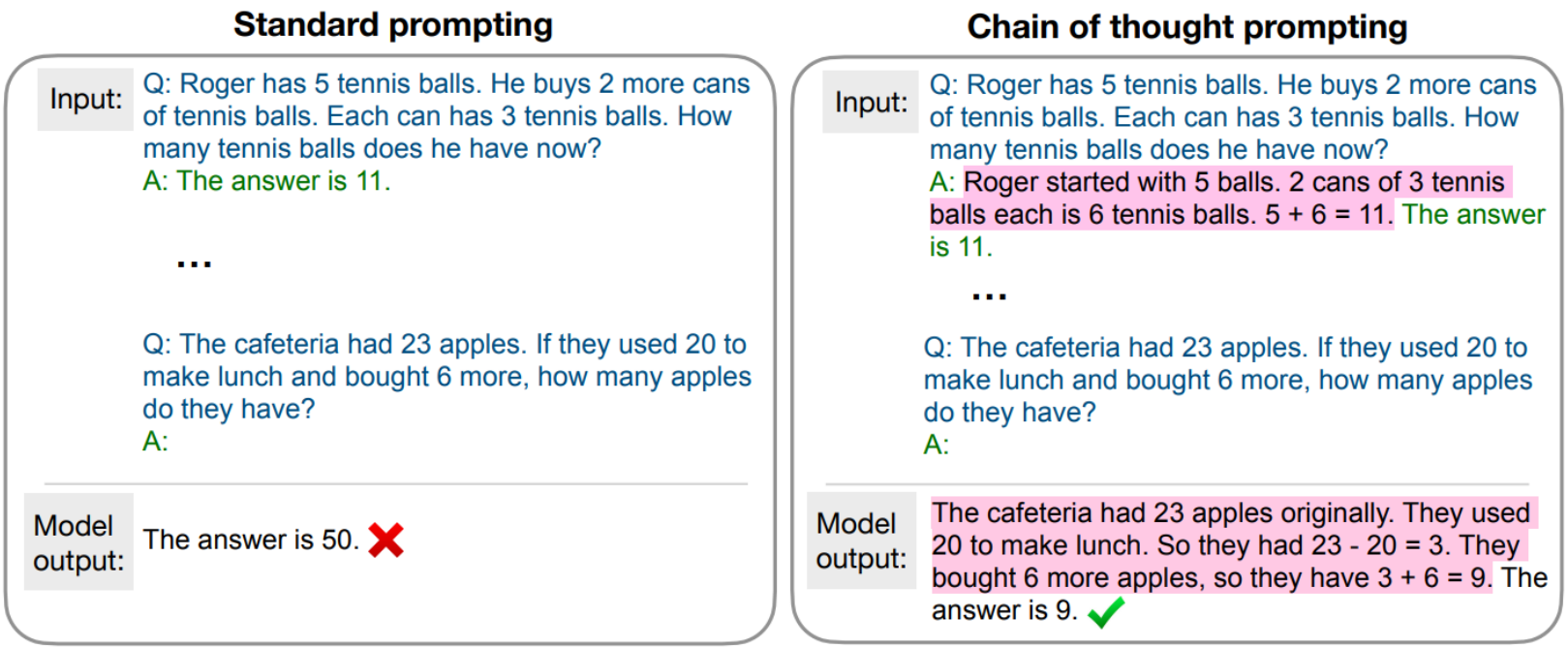


Figure 8: Chain of thought prompting allows language models to better perform multi-step reasoning tasks such as math word problems.

DISCUSS

Try out: Writing with transformers

https://app.inferkit.com/demo

DISCUSS

What are major limitations of the presented text generation models?

DISCUSS

Compare GPT, GPT-2, GPT-3 and PaLM regarding:

(i) Number of co-authors

(ii) Number of pages

What do you conclude?

Summary

- Auto-Encoder
  - Learn to “copy” the input to the output with restrictions
  - Yields good (efficient, low-dimensional) representations
- Encoder-Decoder
  - Use one (RNN, Transformer) for encoding and one for decoding
  - **Attention-based approaches**
- Text generation models (decoder-only)
  - Near human performances
- Evaluation Metrics
  - Difficult task
  - Use transformers to evaluate transformers

- Major breakthroughs in 10+ years

References (1)

- Cho, K., Gulcehre, B. V. M. C., Bahdanau, D., Schwenk, F. B. H., & Bengio, Y. (2014): Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. EMNLP.
- Bahdanau, Dzmitry, Kyunghyun Cho, Yoshua Bengio (2015): Neural Machine Translation by Jointly Learning to Align and Translate. ICLR. http://arxiv.org/abs/1409.0473
- Minh-Thang Luong, Hieu Pham, Christopher D. Manning (2015): Effective Approaches to Attention-based Neural Machine Translation, EMNLP.
- Li Dong, Mirella Lapata (2016): Language to Logical Form with Neural Attention, http://arxiv.org/abs/1601.01280
- Alexander M. Rush, Sumit Chopra, Jason Weston (2015): A Neural Attention Model for Abstractive Sentence Summarization, ACL, http://www.aclweb.org/anthology/D15-1044
- Tai, Kai Sheng, Richard Socher, and Christopher D. Manning (2015). Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1556–1566.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio (2016): Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, http://arxiv.org/abs/1502.03044

References (2)

- Faruqui et al. (2016), Morphological inflection generation using character sequence to sequence learning.
- Schnober, Eger, Do Dinh, and Gurevych (2016). Still not there? Comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks
- Aharoni and Goldberg (2017). Morphological inflection generation with hard monotonic attention.
- Chung, Cho, and Bengio (2016). A character-level decoder without explicit segmentation for neural machine translation.
- Sutskever et al. (2014), Sequence to sequence learning with neural networks