Transformers for text processing

- Text is a sequence
- Position of words matter
- Order of the words matter

How to design a model architecture
that captures this?

Option 1: Recurrent neural network

- Put words into the network one by one
- The network will keep the previous words in "memory"
- Position of the word = how long ago it got into memory

Problem:

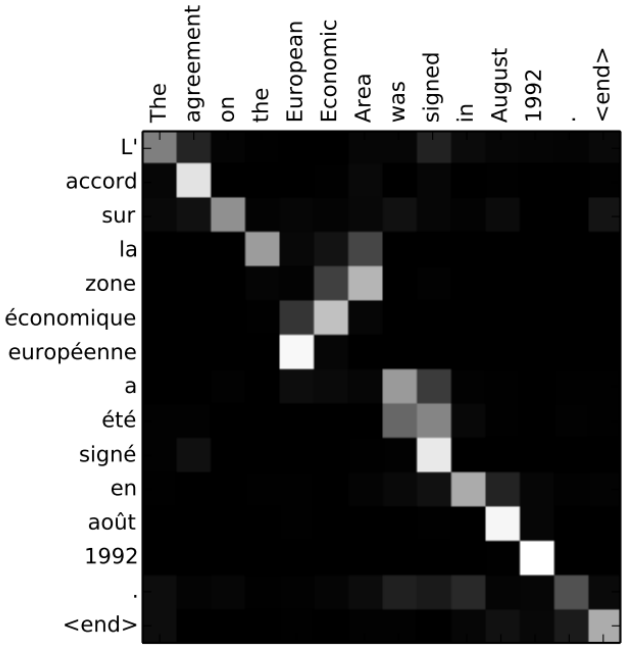
- Can't train in parallel
- It's hard to make networks with good memory -> hard to remember words that were far ago -> hard to capture relationships between words far from each other

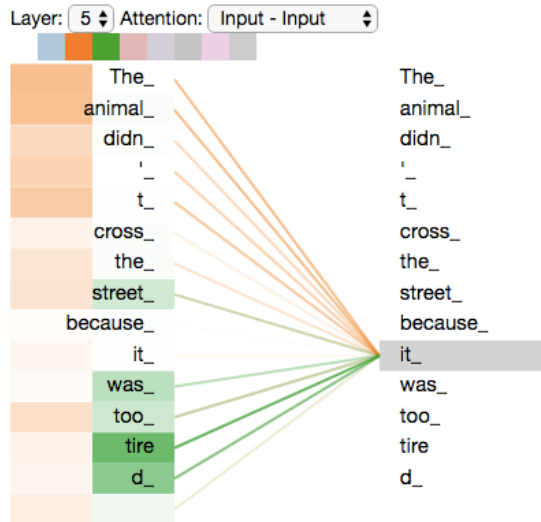
Option 2: Attention

- Give all of the words to the network in parallel
- Add to each word a vector that marks its position
- Compute attention between words

Attention = "to which other words should the network pay attention when processing this word"

Option 2: Attention





https://jalammar.github.io/illustrated-transformer/

Transformers

- Large Neural networks that use attention
- Famous language model: GPT-3, BERT, ALBERT, ELECTRA...
- Starting to be used also for image processing and time-series processing

Code time!

https://github.com/janbogar/czechitas-transformers/blob/main/czechitas-BERT.ipynb

Transfer learning

We want to use the network for something it wasn't trained for.

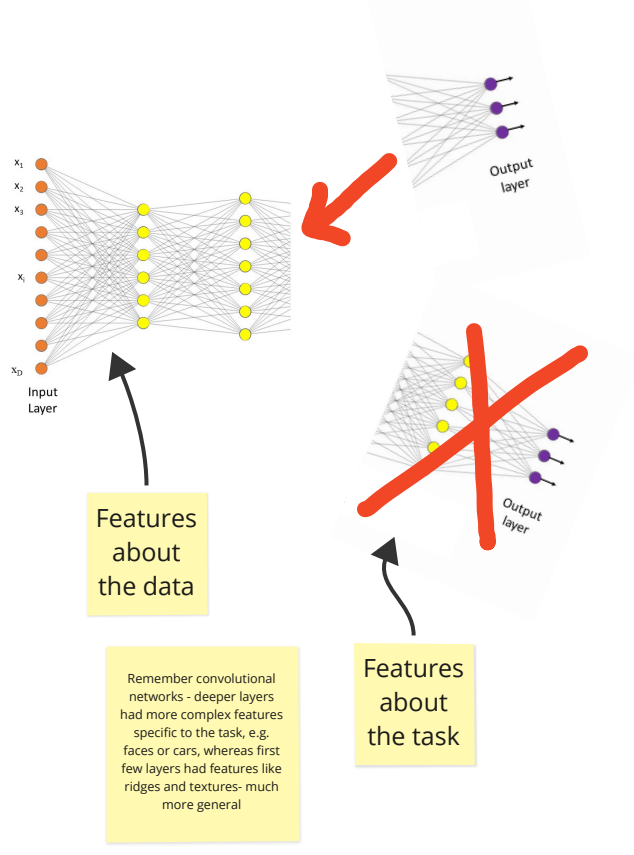
Why?

- somebody else already trained it
- I have enough data to train it, but for a different task

How?

Finetuning

1. Train on a lot of data
2. If needed:
    a. Cut off the end of the network
    b. Replace the end of the network
3. Train on a small amount of data



Positives:

- best results

Negatives:

- difficult- you train a huge network

Embeddings

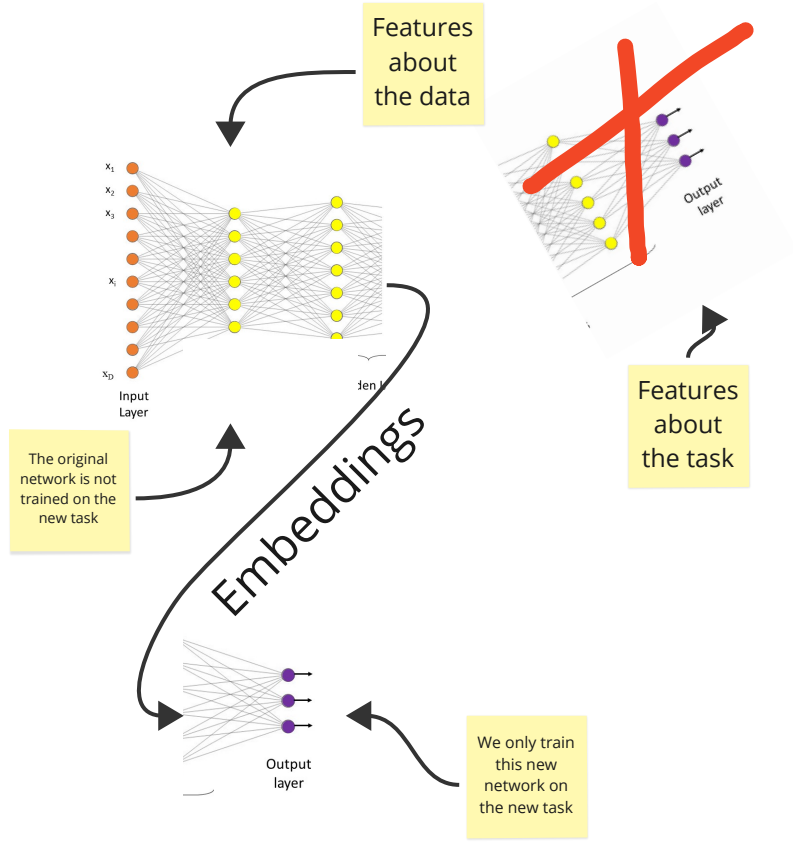
1. Train on a lot of data
2. Cut off the end of the network
3. Process with the network
4. Train a new network on the output of small network on a small amount of data

Positives:

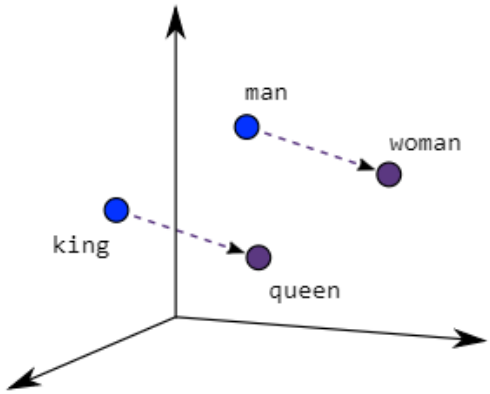
- Embeddings represent data - usefull
- Small network is easy to train

Negatives:

- Not as good results as finetuning



Embeddings



King - Man + Woman = Queen

$$\begin{bmatrix} -4 \\ 0 \\ -2 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -3 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -3 \\ -1 \\ -2 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

actual vectors would have more dimensions, this is just illustratory

Embeddings can express relevant properties of data as vectors, which makes it very useful. Here they obviously capture some meaning of the words.

https://en.wikipedia.org/wiki/Word2vec