National Technical
University of Ukraine
"Igor Sikorsky
Kyiv Polytechnic Institute"

Institute of
Physics and
Technology

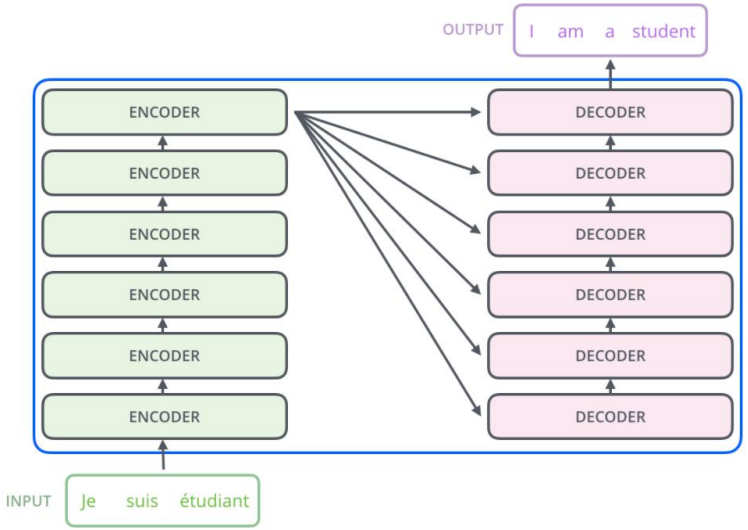
Intellectual Data Analysis

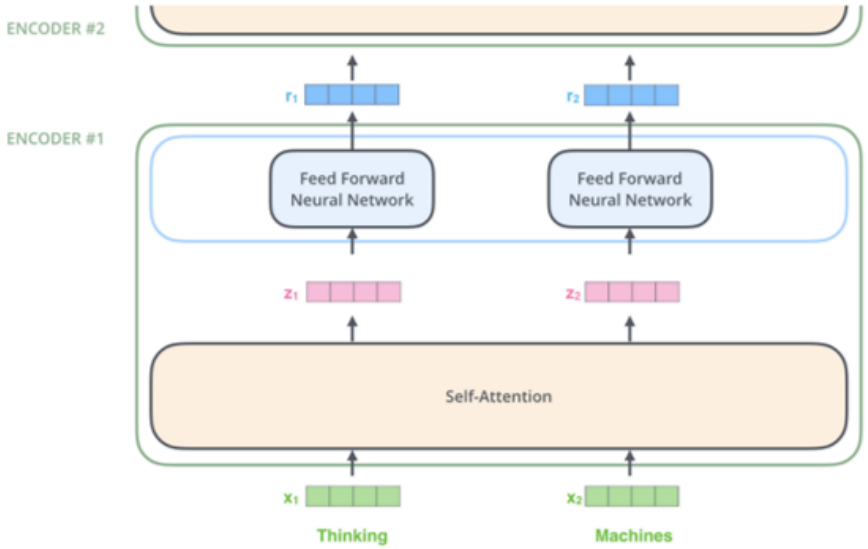
Practice 7: Transformers

Dr. Nataliya K. Sakhnenko

Please review Lecture 11 before this practical

Transformer





6 Dec 2017

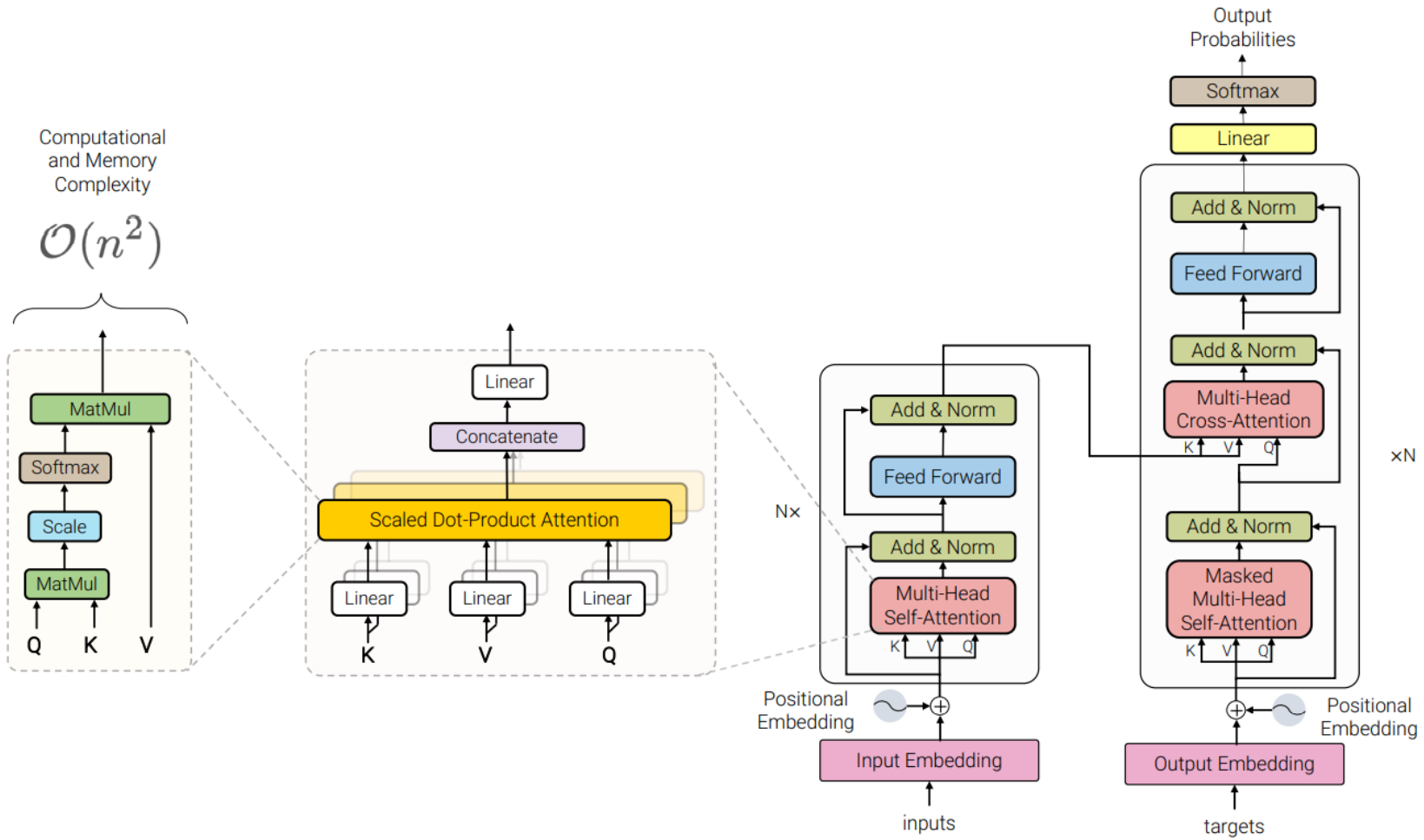
Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$

Attention(Q, K, V)

Transformer



Positional Encoding

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Hugging Face

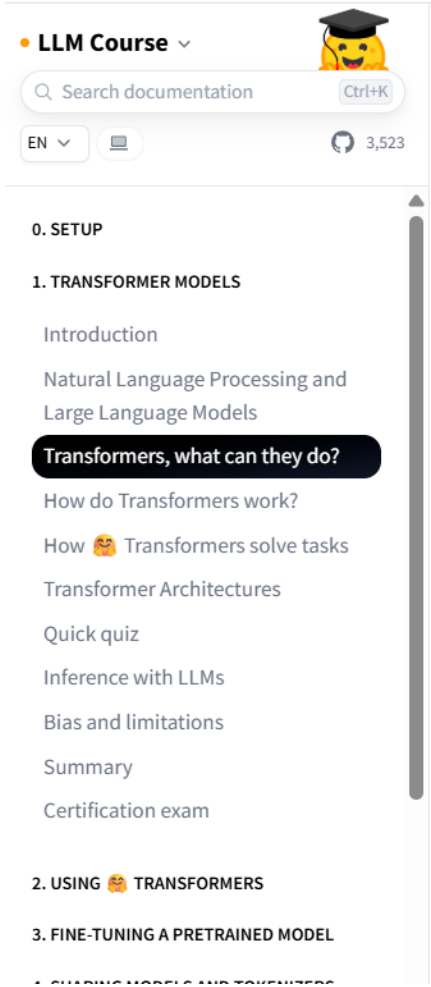
What is Hugging Face?

- A leading open-source ecosystem for modern AI and NLP.
- Provides thousands of **pretrained models** (Transformers, Diffusion, Speech, Vision).
- Offers large **public datasets** and tools for easy data loading.
- Includes powerful libraries: **Transformers**, **Datasets**, **Tokenizers**, **Accelerate**.
- Supports **fine-tuning**, evaluation, and deployment of AI models.
- Hosts **Hugging Face Hub** — a collaborative platform for sharing models, datasets, and demos.
- Enables easy experimentation via **pipelines**, Spaces, and inference widgets.
- A community-driven platform fostering **open, transparent, and reproducible AI**.

https://huggingface.co/blog/proflead/hugging-face-tutorial

https://huggingface.co/learn/llm-course/en/

Hugging Face



The most basic object in the 🤗 Transformers library is the `pipeline()` function. It connects a model with its necessary preprocessing and postprocessing steps, allowing us to directly input any text and get an intelligible answer:

```
from transformers import pipeline

classifier = pipeline("sentiment-analysis")
classifier("I've been waiting for a HuggingFace course my whole life.")
```

```
[{'label': 'POSITIVE', 'score': 0.9598047137260437}]
```

We can even pass several sentences!

```
classifier(
    ["I've been waiting for a HuggingFace course my whole life.", "I hate this so much!"]
)
```

```
[{'label': 'POSITIVE', 'score': 0.9598047137260437},
 {'label': 'NEGATIVE', 'score': 0.9994558095932007}]
```

By default, this pipeline selects a particular pretrained model that has been fine-tuned for sentiment analysis in English. The model is downloaded and cached when you create the `classifier` object. If you rerun the command, the cached model will be used instead and there is no need to download the model again.

https://huggingface.co/docs/transformers/main_classes/pipelines

https://huggingface.co/learn/llm-course/en/chapter1/3

Hugging Face Tasks page: huggingface.co/tasks

Model Hub (filter by Task): huggingface.co/models