Deep Learning

Übung WS 23/24

Judith Nester (nester@uni-koeln.de)

01-02-2024

Studienleistung

Wer jetzt noch im Klips-Kurs ist, bekommt die Studienleistung. Bitte bei mir melden, falls es Redebedarf gibt!

Recap

» Encoder-Decoder

- Separation of the processing of input data from the generation of the output
- Encoder -> maps input data to an internal representation
- Decoder -> maps from internal representation to the output
- Advatages:
  - Flexibility for various tasks
  - Ability to learn complex mappings between input and output
- Challenges:
  - Limited processing of long sequences
  - Limited parallelization leading to slower training times

Recap

- Transformer
  - Uses Encoder-Decoder Architecture with Attention mechanism
  - Uses Attention mechanism to allow the network to learn what to focus on
  - Processing of input data in parallel -> fast!
  - Make it possible to divide Encoder and Decoder depending on the task
  - Good scalability for larger datasets and models
- Transfer Learning
  - Enables domain adaption by further "specialising" a pre-trained language model by finetuning it on a smaller, more specific data set
- Hugging Face
  - A Python library for transformer models

Today

BERT - Bidirectional Encoder Representation from Transformers

GermanBERT

Domain-specific BERT Models

GPT - Generative Pre-trained Transformer

We did it!

Section 1

BERT - Bidirectional Encoder Representation from Transformers

Introduction

» BERT (Bidirectional Encoder Representations from Transformers) has outperformed the state of the art in many NLP tasks

» General idea

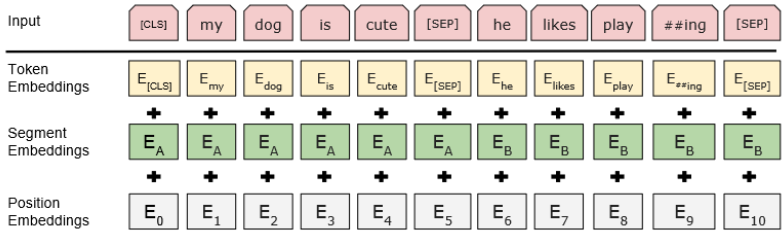
- Encoder-Attention-Decoder architecture (= transformer)
- Stacked Encoders (BASE: 12, LARGE: 24)
- Process whole input at once (max. 512 tokens)
- Bidirectional Context Modeling:
  - Consideration of entire context of a word within a sentence (both left and right)
- Pre-training and fine-tuning on different tasks

Jacob Devlin et al. (2019). »BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding«. *In: Proceedings of NAACL. Minneapolis, Minnesota: ACL, pp. 4171–4186. doi: 10.18653/v1/N19-1423*

Pre-Training and Fine-Tuning

- BERT models are trained on huge data sets
  - "Original" Training Data for BERT:
    - BookCorpus, Wikipedia, Web Data (not labeled!)
    - BERT-Base: $>$ 3 Billion Tokens
    - BERT-Large: even more
- Training one from scratch requires significant resources (time/money)
- Pre-trained models are shared freely (for example on Huggingface)
- Recipe: Take a pre-trained model and fine-tune it on your task
  - Pre-trained model contains an abstract language representation
- Fine-tuning
  - Any language-related task!

Input Representation with BERT



- Token Embeddings: WordPiece-Embeddings
- Segment Embeddings: indicates whether a token belongs to sentence A or B
- Position Embeddings: position of tokens in a sentence; [CLS] = begin token; [SEP] = end token;

BERT Training Tasks

Masked Language Modeling (MLM)

- Sentence-wise
- 15% of the tokens are »masked« by a special token
- Model predicts these, having access to all other tokens

BERT Training Tasks

Masked Language Modeling (MLM)

- » Sentence-wise
- » 15% of the tokens are »masked« by a special token
- » Model predicts these, having access to all other tokens

Next sentence prediction (NSP)

- » Two sentences are concatenated
- » Model has to predict wether second sentence follows on the first or not

Section 2

GermanBERT

GermanBERT

- Trained by Deepset AI
  - https://www.deepset.ai/german-bert
  - https://huggingface.co/bert-base-german-cased
- BERT Model für deutsche Sprache
- Data:
  - The latest German Wikipedia dump (2019) (6GB of raw txt files), the OpenLegalData dump (2.4 GB), and news articles (3.6 GB)
- Outperforms multilingual versions of the original BERT Model
- Also checkout GBERT (https://aclanthology.org/2020.coling-main.598/)

Model	germEval18Fine	germEval18Coarse	germEval14	CONLL03	10kGNAD
multilingual cased	0.441	0.71	0.834	0.792	0.888
multilingual uncased	0.461	0.731	0.823	0.786	0.901
German BERT cased (**ours**)	**0.488**	**0.747**	**0.84**	**0.804**	**0.905**

Section 3

Domain-specific BERT Models

HateBERT

- Trained by GroNLP (Natural Language Processing and Computational Linguistics group at the University of Groningen)
- domain-specific for abusive language phenomena
- Data:
  - RAL-E ("a large-scale dataset of Reddit comments in English from communities banned for being offensive, abusive, or hateful" - 43,379,350 tokens)

Tommaso Caselli et al. (2021). »HateBERT: Retraining BERT for Abusive Language Detection in English«. *In: Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021), pp. 17–25. doi: 10.18653/v1/2021.woah-1.3*

HateBERT

- Data was used to re-train BERT base-uncased on the MLM-Task
- Result was a BERT Model with a shifted domain
- HateBERT outperforms original BERT models in hate speech and abusive language tasks

BERT	HateBERT
"women"	
excluded (.075)	stu**d (.188)
encouraged (.032)	du*b (.128)
included (.027)	id***s (.075)

Table 1: MLM top 3 candidates for the templates "Women are [MASK.]".

Other examples

- LEGAL-BERT
  - Legal Texts
  - `https://huggingface.co/nlpaueb/legal-bert-base-uncased`
- BioBERT
  - Biomedical Texts
  - `https://huggingface.co/dmis-lab/biobert-v1.1`
- FinBERT
  - Financial Texts
  - `https://huggingface.co/ProsusAI/finbert`

Section 4

GPT - Generative Pre-trained Transformer

GPT - Generative Pre-trained Transformer

- » Trained by OpenAI
  - Commercial Large Language Model behind ChatGPT (`https://chat.openai.com`)
  - Open Source up to Version 3.5
- » Uses only Decoders in its architecture
- » Trained on unsupervised task of next-word prediction
  - Prediction of the next word in a sentence

GPT-2: Alex Radford et al. (2019). »Language Models are Unsupervised Multitask Learners«. *url: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf*
GPT-3: Tom B. Brown et al. (2020). »Language Models are Few-Shot Learners«. *doi: https://doi.org/10.48550/arXiv.2005.14165*

GPT - Generative Pre-trained Transformer

- Uses only Decoders in its architecture
- Trained on data set of 300 billion tokens
- Trained on unsupervised task of Next-word Prediction
  - Prediction of the next word in a sentence
  - Error is calculated and the model can be improved
  - Output is added to the input

GPT 3

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

Latest Exercise

Have you found any special Transformer models? Have you been inspired for a possible module exam topic?

Section 5

We did it!

Ihr habt viel gelernt

- Git basics
- Python basics, installation issues
- Logistic regression / gradient descent
- Feed-forward neural networks
- Training (Hyperparameter, Trouble Shooting)
- Embeddings
- Sequential Data
- Recurrent and LSTM networks
- Encoder/Decoder networks, Attention
- Transformer

Figure: Dall.E2 ("University students enjoying their lecture-free time, pop art")