Large Language Models (LLM)

PA154 Language Modeling (11.1)

Pavel Rychlý

pary@fi.muni.cz

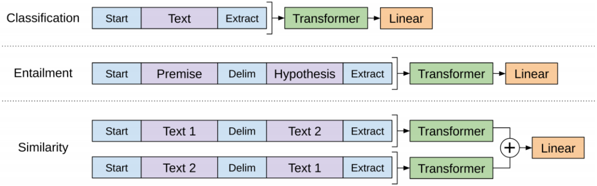
April 27, 2023

Large Models

- bigger is better
- many layers
- need big machines
- using advanced hardware: GPU, TPU on multiple servers

Usage of Large Models

- training of big models on huge data is expensive (long training time)
- fine tuning on small data of target task
- combining language model with additional NN/layer, training only new layer
  - big model is frozen, only used



Pre-trained models

- word2vec, fastText: pre-trained word embeddings
- transformers: BERT
- transformer modifications:
    - RoBerta, Albert, ...
- language specific models
- multilingual models

Pre-trained fastText

- 157 languages
- word vectores with dimension 300
- up to 1 or 2 mil. words
- Czech:
  - 2 mil. words
  - text format: 1.2 GB, binary format 4.2 GB
- Breton:
  - 602k words
  - text format: 340 MB, binary format 4.2 GB

Pre-trained fastText

Czech embeddings trained on Common Crawl: `cc.cs.300.vec.gz`

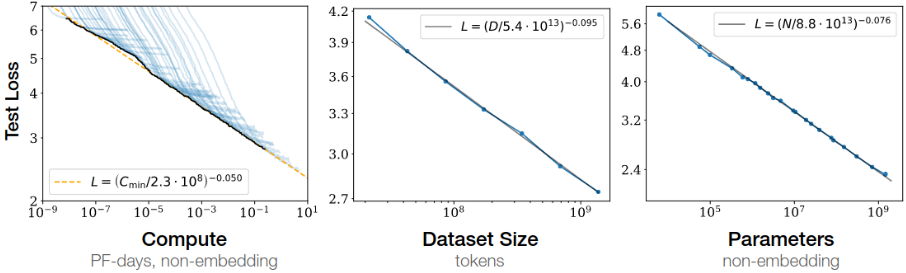
```
2000000 300
, 0.0052 0.1646 0.0675 0.0577 0.2342 0.0089 0.1601 0.0240 -0
. 0.0485 0.0674 0.0261 0.0220 -0.0779 -0.0309 -0.2006 0.0100
a 0.1253 0.0177 0.0770 -0.0103 0.0687 0.0175 0.0171 0.0013 -
</s> 0.0251 -0.0350 0.0364 0.0349 0.0159 -0.0586 -0.4607 0.0
: -0.0715 -0.0175 -0.0210 0.0818 -0.0174 -0.0204 0.0574 0.00
v 0.1013 0.1792 -0.0174 0.0365 0.0920 0.0802 -0.1830 0.0271
na -0.1200 0.2000 0.2071 0.0144 0.3272 -0.0145 -0.1196 0.080
) 0.0614 -0.1514 0.0203 0.1658 0.0958 -0.0628 -0.0841 -0.064
se -0.1456 0.1170 0.0285 -0.0062 -0.0890 -0.0042 -0.0969 -0.
( 0.0671 -0.1871 0.0332 0.1324 0.1774 -0.0685 0.0082 -0.0666
" 0.1381 -0.2536 0.0805 0.0379 0.2684 -0.0038 0.0437 -0.0905
je 0.0170 -0.1937 0.0388 -0.0084 0.1255 -0.0953 -0.0267 -0.0
- -0.2497 -0.0093 0.1759 -0.0839 0.1842 -0.0276 0.1605 -0.08
s 0.0081 -0.0854 -0.0566 0.0116 -0.5178 -0.0091 -0.2048 0.05
```

Scaling transformers

- main factors:
  - number of model parameters N
  - size of the dataset D
  - amount of compute operations C
- evaluation on test loss (cross-entropy)
- there is a capacity limit for a fix N, D, or C
- performance improves predictably as long as we scale up N and D in tandem

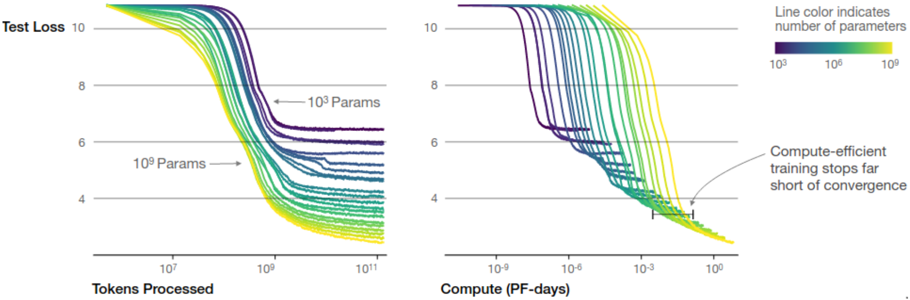
Scaling transformers

- number of model parameters N
- size of the dataset D
- amount of compute operations C



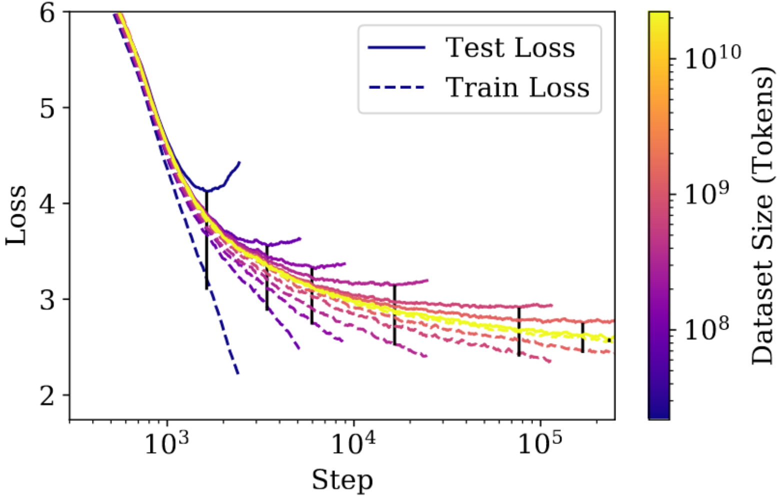
Scaling transformers

- larger models require **fewer samples** to reach the same performance
- larger models are **much slower** per sample
- smaller models **reach same performance faster**



Scaling transformers

- bigger dataset reduces overfitting
- N = 300M parameters



BERT

- *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*
- encoder only (not language modelling)
- pre-training on raw text
- masking tokens, is-next-sentence
- big pre-trained models available
- domain (task) adaptation

Input: `The man went to the [MASK]`$_1$ `. He bought a [MASK]`$_2$ `of milk .`
Labels: `[MASK]`$_1$ `= store; [MASK]`$_2$ `= gallon`

Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`

BERT's sizes

- BASE
  - L=12, H=768, A=12
  - Total Parameters=110M
- LARGE
  - L=24, H=1024, A=16
  - Total Parameters=340M

ALBERT

- A Lite BERT
- factorized embedding parameters
- cross-layer parameter sharing
- inter-sentence coherence loss
  Next Sentence Prediction → Sentence-Order Prediction
- much smaller: No. parameters: 108M → 12M (base)

Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`

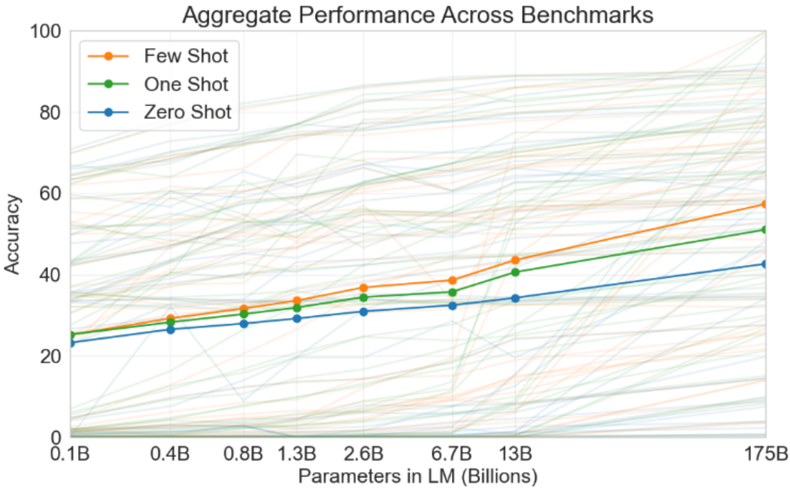
GPT

- Open AI
- GPT-2: 1.5 billion parameters
- GPT-3: 175 billion parameters
- very good text generation
  $\rightarrow$ potentially harmful applications
- Misuse of Language Models
- bias – generate stereotyped or prejudiced content: gender, race, religion
- Sep 2020: Microsoft have ”exclusive” use of GPT-3

GPT3's sizes

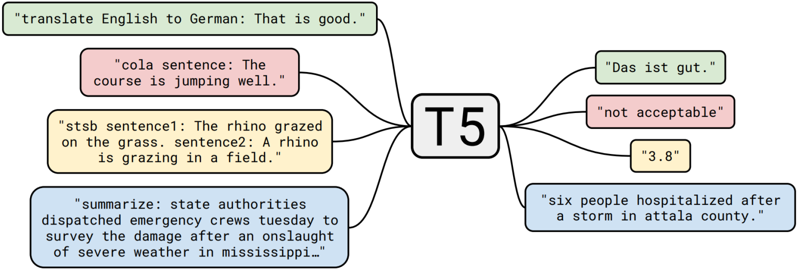
Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	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}

GPT3 performance



T5: Text-To-Text Transfer Transformer

- Google AI
- transfer learning
- C4: Colossal Clean Crawled Corpus



Subword tokenizers

- universal tokenization: subword units
  - Byte-Pair Encoding (BPE)
  - WordPiece
  - SentencePiece

Intrinsic evaluation

- direct evaluation of word embeddings
- semantic similarity (WordSim-353, SimLex-999, ...)
- word analogy (Google Analogy, BATS (Bigger Analogy Test Set))
- concept categorization (ESSLLI-2008)

Extrinsic evaluation

- using the model in a downstream NLP task
- Part-of-Speech Tagging, Noun Phrase Chunking, Named Entity Recognition, Shallow Syntax Parsing, Semantic Role Labeling, Sentiment Analysis, Text Classification, Paraphrase Detection, Textual Entailment Detection

Multi-task benchmarks

- GLUE (`https://gluebenchmark.com`)
  nine sentence- or sentence-pair language understanding tasks
- SuperGLUE (`https://super.gluebenchmark.com`)
  more difficult language understanding tasks
- XTREME – Cross-Lingual Transfer Evaluation of Multilingual Encoders
  (`https://sites.research.google/xtreme`)
  40 typologically diverse languages, 9 tasks

Libraries and Frameworks

- Dive into Deep Learning: online book
  `https://d2l.ai`
- Hugging Face Transformers: many ready to use models
  `https://huggingface.co/transformers`
- jiant: library, many tasks for evaluation
  `https://jiant.info`
- GluonNLP: reproduction of latest research results
  `https://nlp.gluon.ai`
- low level libraries: NumPy, **PyTorch**, TensorFlow, MXNet