DLP - Week - 3 Notes

L1 Intro to LLMs

Initialize → Train the model

import transformers

- how to train transformer based models using the HF transformers module.
- over the years, the dataset keeps growing, & also the model size.
- transformers based models can be of 3 types — Encoder only, decoder only, & encoder & decoder models. But these days decoder only models are emerging for the tasks of lang. modelling. (GPT based)
- all the LLMs go thru pretraining phase.
- we try to use unlabelled data to train the LLM, the entire language. After that, model can perform anything like sentiment, NLP, etc.

L2 Causal Language Modelling

$$P(x_1, \dots x_T) = \prod_{i=1}^{T} P(x_i | x_1, \dots x_{i-1})$$

$P(x_i | x_1 \dots x_{i-1}) = f_\theta(x_i | x_1 \dots x_{i-1})$
can f_θ be a transformer?

- so we had 3 types of transformers?
  ↳ we look into decoder only model.

→ Feed forward NN → input is the seq. of words
↑
Masked Multi Head (Self Attenⁿ) → we want the model to see only the present & past inputs.

→ we achieve this by masking. So we ∴, we need the multi-head attenⁿ layer is required. The output represents each term in the chain rule —

$$= P(x_1)\ P(x_2 | x_1)\ P(x_3 | x_2, x_1)$$

- the probabs. are determined by the parameters of the model → θ.

We want to maximize the likelihood of $L(\theta)$

L3 Generative Pre Trained Transformer

- stack of n modified decoder layers.

X is input seq.

$h_0 = X \in R^{T \times d_{model}}$

$h_l = transformer\_block(h_{l-1}), \quad \forall l \in [1, n]$

$P(x_i) = softmax(h_n[i] W_v)$

$L = \sum_{i=1}^{T} \log(P(x_i | x_1, \ldots, x_n))$

- Book corpus data → 7000 unique books, 74 million sentences, 1 billion words 16 genres, uses long-range contiguous text. BPE Tokenizer. Vocab size = 40478 Embedding dim → 768
- Model – 12 decoder layers (~~transformer~~ blocks)
  Context size = 512
  Attention heads = 12
  FFN Hidden Layer size = $768 \times 4 = 3072$
  GELU Activation (Gaussian) → FFNN
  Multi head attention mask

concatenate

Matmul

softmax

$\cdot \frac{1}{\sqrt{d_k}}$

$QK^T + M$

all work parallelly

12 such attention mask heads

- after that a residual connect^{n} (here a dropout is used).
- FFNN

—— 768

——————→ 3072

—— 768

- no. of params.

token embedding = V x embedding

= 40478 x 768 = 31 M

Position embedding = context len x em.

= 512 x 768 = 0.3M

attn^{n} params per block

$W_Q = W_K = W_V = 768 \times 64 \times 3 \times 12$

$= 1.7$ M

linear layer = $768 \times 768 \approx 0.6$

$2.3\text{M} \times 12 = 27.6$ M for attn^{n} block.

FFN per block

$2 \times (768 \times 3072) + 3072 + 768 = 4.7$M

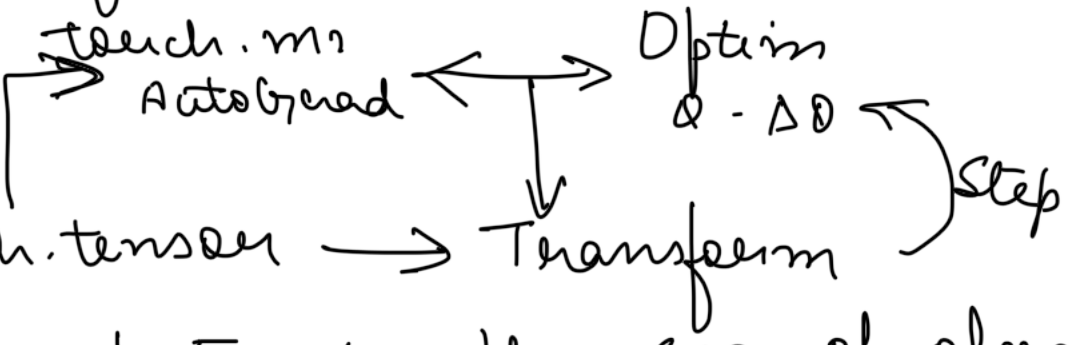
x12

= 56.4 M

Total = 117 million params

64 HF Transformers

import transformers



- the module abstracts the seq. of operⁿ
Use the data in batches.
- Please beware of shuffling of data.
Because of Python GIL.
↓ to avoid this
use built-in data loader