Introduction video

Wednesday, April 19, 2023 10:40 AM

1-4 Intro. to main concepts

5-8 basics of dataset and tokenizer.

9-12 Transformer models in CV and speech recognition

Gradio –

NLP – field of linguistic and ML, understanding everything related to human language.

NLP tasks

1. Classifying whole sentences :– SA, Spam, grammatically correct
2. " each word :– gram. components, NER
3. Generating text content :–
4.

Why challenging?

Pipeline ( ) fn :– Selects a model pretrained for a particular tasks.

3 main steps

① text preprocessed in a way that model can understand

② preprocessed i/p ⟶ to the model.

③ predictions are processed, so you can make sense.

Available pipelines :–

- feature-extraction
- fill-mask
- ner
- qa
- SA
- summarization
- text-generation
- translation

text-generation
translation
zero-shot classification.

Mask-filling:- is used to train BERT models
zero-shot learning:- X need for fine-tuning model on data to use.
Text-generation:- provide a prompt, model will auto-complete it.
Grouped_entities = True, regroup together parts of the sentence
Hugging Face into one

How Transformer Works?
June 2018
GPT-family:- Auto-regressive
BERT-family:- Auto-encoding
BART/T-5 :- seq-to-seq
Trained as lang. models
large amount of raw texts
self-supervised fashion.
Objectives automatically computed
X humans to label the data.
Statistical understanding of the language.
Transfer learning
fine-tuning in a supervised way.
Caused language modeling
O/p depends on present and past i/p
Masked language modeling:-
To achieve better performance by ↑ size and amount of data
in which models are pretrained.

TgT
CO_2 footprint of LLMs:-

CO_2 footprint of LLMs:—

Type of energy:— (non)-renewable

Training Time

H/W

I/O

Data

CO_2 emission per kWh wise vary করে।

Mumbai তে 920 gm

Montreal এ 20 gm

almost 40X

Other elements:—

pretraining → recycling

fine-tuning

Starting with smaller exp. & Debugging

Lit. reviews for selecting hyperparameter range

Random search Vs Grid search

ML emissions calculator

Code carbon

Emissions tracker

large corpus

Pretraining:— Training from scratch.

Weights initialized randomly

training w/o any prior knowledge.

Large dataset

Training Time Weeks.

Fine-tuning:— Training after pretraining

" on dataset specific task

" on dataset specific task

Why m directly train on specific tasks (fine-tuning):—
advantage of taking knowledge acquired during pretraining.
req. less data for decent result.
Time & resources ↓

Encoder:— Self-Attention, Bi-dir.
Decoder:— Uni-dir., Auto-regressive, Masked Self-Attention

i/p → Encoder → rep. (features)

encoder rep.'s + Other i/p → Decoder → Target seq.

Encoder only models:— req. understanding of i/p.
Sentence classification
NER

Decoder " ":— generative tasks
Text generation.

En-Decoder model / seq-seq:— generative tasks req. an i/p
Translation, summarization.

Attention layers:—

Tell the model to give more attention to certain words.
Word এর meaning আগের বা পরের word এর depend করে।
Context

Original architecture of Transformers:—
Designed for Translation.

আলাদা lang. → Encoder

Same Target lang. → Decoder →

Same target ... → Decoder →

Encoder এর word দেখতে পারে
Decoder seq. বাস্তব করতে। future words training এর সময়ে দেখতে পারবে না।

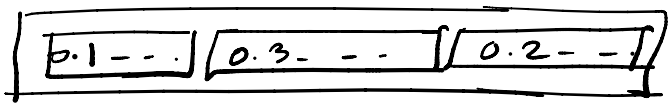
1st Attn layers in decoder pays attn. to all past i/p to decoder.
2nd " " " " can access o/p of encoder. Thus whole i/p seq. to predict next word.

Architecture: — skeleton of the model.
def. of layers
op. happen w/i ; BERT

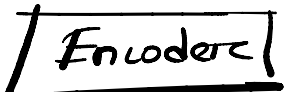
Checkpoints: — weights loaded in a archi., bert_base_cased

Model: — Architecture / checkpoint both.

When to use encoders:
Bi-directional
Good at extracting meaningful info.
Seq. classification, qa, MLM
NLU
examples — BERT, RoBERTa, ALBERT.



768 Dim Vectors



Welcome to NYC

Use encoders of Transformer
Attn. layers access all the words in initial sentence.

Use cases of transformers

Attn. layers access all the words in initial sentence.
Bi-dirc. attn. / auto-encoding
Pretraining by MLM.
Best suited for understanding full sentence.
sentence classification, NER, QA
examples:- (AL) BERT,
Distil "
ELECTRA
RoBERTa.

Decoder - archi:- GPT-2

same tasks can be performed as encoder-only models but performance decreased a little.

Masked self-attention mechanism:-

only see words on the left
right side words are hidden.

When to use?

Uni dirc.
Causal tasks; generating seq.
NLG
GPT-2/Neo.
1024

Only access words before it. / Auto-regressive models.
Text generation.

CTRL
GPT/-2
Transformer-XL

3. En-decoder / seq-to-seq models:-

encoder - all the words
decoder - access words positioned before a given word. } attn. layer access.

decoder – access words positioned before a given word.] access.

Pre-training more complex
T5 masks random spans of texts. & predicts to pretrain

Best suited for gen. new sentences on a given i/p
Summarization, Translation, Generative QA.
m(BART), Marian, T5.

Bias and Limitations:—

Dataset বড়ো করার জন্য internet যা আছে তাই দিয়ে pre-train করা হয়। এতে করে অনেক সময় bias চলে আসে।

This man/woman works as a _____.

এটার result gender-wise vary করতে।

pretrain model use করার সময় মনে রাখতে হবে যে model গুলা sexist, racist/ homophobic হতে পারে।

Fine-tuning করার দ্বারা এই Bias কে remove করা possible না।

Chapter2:— Using Transformers:—

Intro:— Transformers very large.
- Billions of parameters.

New models released regularly.

Transformer library to load, train & save models.

Features:—
1. Ease of use:— Two line code ব্যবহার করা যায় model গুলা
2. Flexibility:— ~~DNN~~ other NN model এর মতো framework ব্যবহার করে use করা যায়।
3. Simplicity:— Understandable & usable by using a single file.

Chapter 2:— Contents
model + Tokenizer = replicate pipeline fn.
model & Config. classes.
Load a model

... prediction.

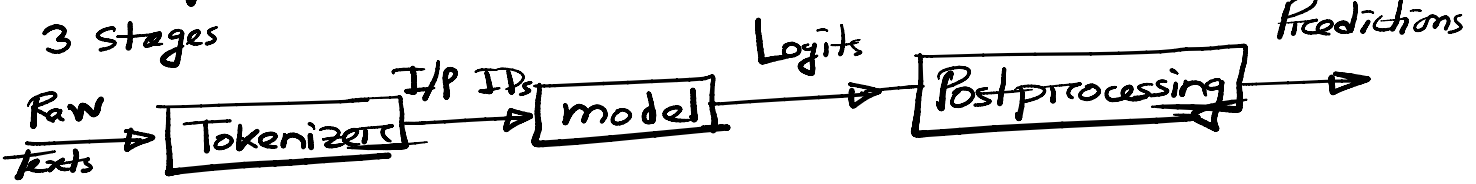
Load a model

Process numerical i/p to o/p prediction.

Tokenizer API:-

What happens inside pipeline fn:-

3 stages



Raw Texts → Tokens → Special Tokens → I/P IDs

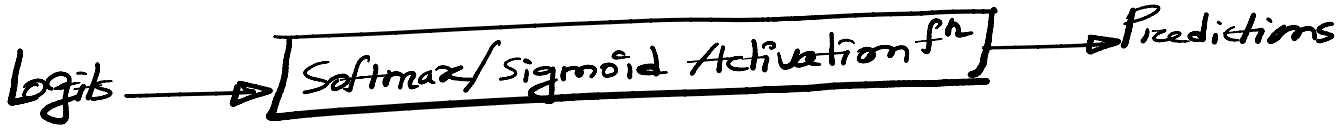
Tokenizer Flowchart.

Attn. mask = 0 মানে padding, এগুলো Attn. দিতে হবে না

= 1 " Text. Attn. দিতে হবে।

Automodel loads a model w/o pretraining head.

Logits:- Raw, unnormalized predictions of model



Preprocessing with a Tokenizer:-

Can't process Text directly } problem
Convert text i/p into no.

Soln → use Tokenizer

Tokenizer responsible for:-

Splitting i/ps into tokens

Tokens Mapping → Integer

+extra i/p

Model pretraining এর সময় যেভাবে preprocessing করা হয়েছে exactly সেভাবেই preprocessing করতে হবে।

1. Download info. from Hub.

from_pretrained(model_name)

1. Download info. from Hub.

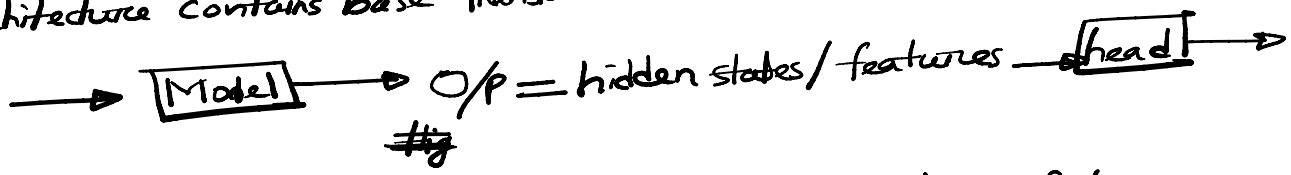
AutoTokenizer.from_pretrained(model_name)

Tokenizer returns dict. (I/P ID)
need to convert into Tensors

Model Download করা:-

AutoModel.from_pretrained (checkpoint)

Architecture contains base Transformer Module



Features = High dim. vector rep. contextual understanding of i/p.

Same architecture use করে Diff. Task করা যাবে যদি head diff. হয়।

Vector O/p of Transformer Model:-

Large dim. vector as O/p. Three dim.
Batch size:- At a time কতটা sentence process করবে
Seq. len:- Padding করার সময়ে সবচেয়ে sentence এর len. কত
Hidden size:- Vec. dim. of each model i/p। 768 থাকবে।

O/p behave like namedtuples / dictionaries

Access the elements by
by attributes op.last_hidden_state.shape
" key op["last " "]
" index op[0]