Search models, datasets, users...

openai-community / gpt2

like 2.29k

Text Generation · Transformers · PyTorch · TensorFlow · JAX · TF Lite · Rust · ONNX · Safetensors · English · doi:10.57967/hf/0039 · gpt2 · exbert · text-generation-inference · Inference Endpoints · License: mit

Train · Deploy · Use this model

Model card · Files · Community 102

Downloads last month
9,209,928

Safetensors ⓘ

Model size	137M params
Tensor type	F32

Inference API ⓘ

Warm

Text Generation

Examples

My name is Thomas and my main

Compute

0,1

</> View Code

Maximize

Model tree for `openai-community/gpt2`

Adapters	1598 models
Finetunes	1079 models
Quantizations	41 models

Spaces using `openai-community/gpt2` 100

🏆 open-llm-leaderboard/open_llm_leaderboard
😻 Gustavosta/MagicPrompt-Stable-Diffusion
🚀 shi-labs/Versatile-Diffusion
🏆🏋️ optimum/llm-perf-leaderboard
📚 yizhangliu/Grounded-Segment-Anything
+ 95 Spaces

Edit model card

GPT-2

Test the whole generation capabilities here: https://transformer.huggingface.co/doc/gpt2-large

Pretrained model on English language using a causal language modeling (CLM) objective. It was introduced in this paper and first released at this page.

Disclaimer: The team releasing GPT-2 also wrote a model card for their model. Content from this model card has been written by the Hugging Face team to complete the information they provided and give specific examples of bias.

Model description

GPT-2 is a transformers model pretrained on a very large corpus of English data in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts. More precisely, it was trained to guess the next word in sentences.

More precisely, inputs are sequences of continuous text of a certain length and the targets are the same sequence, shifted one token (word or piece of word) to the right. The model uses internally a mask-mechanism to make sure the predictions for the token `i` only uses the inputs from `1` to `i` but not the future tokens.

This way, the model learns an inner representation of the English language that can then be used to extract features useful for downstream tasks. The model is best at what it was pretrained for however, which is generating texts from a prompt.

This is the **smallest** version of GPT-2, with 124M parameters.

Related Models: GPT-Large, GPT-Medium and GPT-XL

Intended uses & limitations

You can use the raw model for text generation or fine-tune it to a downstream task. See the model hub to look for fine-tuned versions on a task that interests you.

How to use

You can use this model directly with a pipeline for text generation. Since the generation relies on some randomness, we set a seed for reproducibility:

```
>>> from transformers import pipeline, set_seed
>>> generator = pipeline('text-generation', model='gpt2')
>>> set_seed(42)
>>> generator("Hello, I'm a language model,", max_length=30, num_return

[{'generated_text': "Hello, I'm a language model, a language for thinki
 {'generated_text': "Hello, I'm a language model, a compiler, a compile
 {'generated_text': "Hello, I'm a language model, and also have more th
 {'generated_text': "Hello, I'm a language model, a system model. I wan
 {'generated_text': 'Hello, I\'m a language model, not a language model
```

Here is how to use this model to get the features of a given text in PyTorch:

```
from transformers import GPT2Tokenizer, GPT2Model
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')
model = GPT2Model.from_pretrained('gpt2')
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='pt')
output = model(**encoded_input)
```

and in TensorFlow:

```
from transformers import GPT2Tokenizer, TFGPT2Model
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')
model = TFGPT2Model.from_pretrained('gpt2')
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='tf')
output = model(encoded_input)
```

Limitations and bias

The training data used for this model has not been released as a dataset one can browse. We know it contains a lot of unfiltered content from the internet, which is far from neutral. As the openAI team themselves point out in their model card:

> *“Because large-scale language models like GPT-2 do not distinguish fact from fiction, we don’t support use-cases that require the generated text to be true.*
>
> *Additionally, language models like GPT-2 reflect the biases inherent to the systems they were trained on, so we do not recommend that they be deployed into systems that interact with humans > unless the deployers first carry out a study of biases relevant to the intended use-case. We found no statistically significant difference in gender, race, and religious bias probes between 774M and 1.5B, implying all versions of GPT-2 should be approached with similar levels of caution around use cases that are sensitive to biases around human attributes.”*

Here's an example of how the model can have biased predictions:

```
>>> from transformers import pipeline, set_seed
>>> generator = pipeline('text-generation', model='gpt2')
>>> set_seed(42)
>>> generator("The White man worked as a", max_length=10, num_return_se

[{'generated_text': 'The White man worked as a mannequin for'},
 {'generated_text': 'The White man worked as a maniser of the'},
 {'generated_text': 'The White man worked as a bus conductor by day'},
 {'generated_text': 'The White man worked as a plumber at the'},
 {'generated_text': 'The White man worked as a journalist. He had'}]

>>> set_seed(42)
>>> generator("The Black man worked as a", max_length=10, num_return_se

[{'generated_text': 'The Black man worked as a man at a restaurant'},
 {'generated_text': 'The Black man worked as a car salesman in a'},
 {'generated_text': 'The Black man worked as a police sergeant at the'}
 {'generated_text': 'The Black man worked as a man-eating monster'},
 {'generated_text': 'The Black man worked as a slave, and was'}]
```

This bias will also affect all fine-tuned versions of this model.

Training data

The OpenAI team wanted to train this model on a corpus as large as possible. To build it, they scraped all the web pages from outbound links on Reddit which received at least 3 karma. Note that all Wikipedia pages were removed from this dataset, so the model was not trained on any part of Wikipedia. The resulting dataset (called WebText) weights 40GB of texts but has not been publicly released. You can find a list of the top 1,000 domains present in WebText here.

Training procedure

Preprocessing

The texts are tokenized using a byte-level version of Byte Pair Encoding (BPE) (for unicode characters) and a vocabulary size of 50,257. The inputs are sequences of 1024 consecutive tokens.

The larger model was trained on 256 cloud TPU v3 cores. The training duration was not disclosed, nor were the exact details of training.

Evaluation results

The model achieves the following results without any fine-tuning (zero-shot):

Dataset	LAMBADA	LAMBADA	CBT-CN	CBT-NE	WikiText2	PTB	enwiki8	text8	WikiTe
(metric)	(PPL)	(ACC)	(ACC)	(ACC)	(PPL)	(PPL)	(BPB)	(BPC)	(PP
	35.13	45.99	87.65	83.4	29.41	65.85	1.16	1,17	37.5

BibTeX entry and citation info

```
@article{radford2019language,
  title={Language Models are Unsupervised Multitask Learners},
  author={Radford, Alec and Wu, Jeff and Child, Rewon and Luan, David a
  year={2019}
}
```

Visualize in exBERT lite

🤗

Company

TOS

Privacy

About

Jobs

Website

Models

Datasets

Spaces

Pricing

Docs