what is GPT2

Generative Pre-trained Transformer 2(GPT-2) is a language model. It can generate content based on the input. The GPT-2 is based on the Transformer and it has 4 model sizes, which are GPT-2 small, GPT-2 medium, GPT-2 large and GPT-2 extra large. It is pre-trained with a large scale of content dataset and can be used to predict the next word or generate articles based on the assigned input.

As GPT2 is a pre-trained model, the workflow could download the parameters and finetune it with your own workload. A quick example can be found at https://huggingface.co/distilgpt2

Install

For GPT2, there are two recommended ways to use it.

1. API from huggingface
   GPT2 API from huggingface is commonly used. In huggingface, GPT2 is included in the transformers package. You can use pip to install transformers.
   - pip install transformers
   - As the transformers package is compatible with both pytorch(https://pytorch.org/get-started/locally/) and tensorflow(https://www.tensorflow.org/install/pip), you can install either one based on your preference.
2. wrapper from gpt-2-simple(https://github.com/minimaxir/gpt-2-simple)
   This is a python package that wraps the GPT2 with easy-to-use interface. You can use pip to install it
   - pip install gpt-2-simple
   - You also need to install the tensorflow>=2.5.1.

The API from huggingface is more comprehensive as it allows you to switch between multiple language models. And there are also varietas of gpt models supported which can be more efficient than the original gpt2 for some scenarios. However, the gpt2-simple wrapper may be more beginner friendly.

Usage

1. API from huggingface
   a. fine-tune(Or training)

```
# install and import the necessary packages
# pip install numpy transformers datasets
from datasets import load_dataset, load_metric
from transformers import
AutoTokenizer,TrainingArguments,TrainingArguments, Trainer,
AutoModelForSequenceClassification
```

```python
import numpy as np

def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    return metric.compute(predictions=predictions,
references=labels)

def tokenize_function(examples):
    return tokenizer(examples["text"], padding="max_length",
truncation=True)

# set the fine tune dataset, REPLACE to your own dataset
dataset = load_dataset("yelp_review_full")
dataset["train"][100]


# Set the tokenizer. Use the gpt2 in this doc. Huggingface
support multiple model in transformer package and  you can
switch to other models.
# For gpt2, you also need to set the tokenizer.pad_token
tokenizer = AutoTokenizer.from_pretrained("gpt2")
tokenizer.pad_token = tokenizer.eos_token


# Set the training data set
tokenized_datasets = dataset.map(tokenize_function,
batched=True)
small_train_dataset =
tokenized_datasets["train"].shuffle(seed=42).select(range(10
00))
small_eval_dataset =
tokenized_datasets["test"].shuffle(seed=42).select(range(100
0))

# Assign the training model. You can also use other models.
model =
AutoModelForSequenceClassification.from_pretrained("gpt2",
num_labels=5)

# Set other training parameters
metric = load_metric("accuracy")
```

```
training_args = TrainingArguments(output_dir="test_trainer",
evaluation_strategy="epoch")

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=small_train_dataset,
    eval_dataset=small_eval_dataset,
    compute_metrics=compute_metrics,
)

# start training
trainer.train()
```

b. generate

```
>>> 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_sequences=5)
```

2. gpt-2-simple
    - import gpt-2-simple
    - gpt-2-simple.download_gpt2(model_name="124M") #download the pre-trained parameters
    - sess = gpt-2-simple.start_tf_sess()
    - gpt-2-simple.finetune() # finetune with your workload. For the details of parameters, please refer to the github documentation or the colab example attached later.
    - gpt-2-simple.generate(sess, run_name='run1') # start generate content

Useful links

- quick gpt-2-simple code example using google colab https://colab.research.google.com/drive/1VLG8e7YSEwypxU-noRNhsv5dW4NfTGce#scrollTo=4RNY6RBI9LmL
- get start with huggingface https://huggingface.co/docs/transformers/installation
- huggingface gpt2 doc https://huggingface.co/docs/transformers/model_doc/gpt2
- huggingface transformer fine-tune https://huggingface.co/docs/transformers/training &&https://colab.research.google.com/github/huggingface/notebooks/blob/main/transformers_doc/en/pytorch/training.ipynb