


📄 qwen.md 20.43 KB

 马建仓 AI 助手

一键复制

编辑

原始

niujunhao 提交于 1个月前 · [【r1.1.0】修复文档中版本配套关系](#)

• 通义千问

• 模型描述

• 仓库介绍

• 前期准备

• 环境要求

• 模型权重准备

• 从huggingface...

• 模型权重切分与合并

• 全参微调

• 数据集准备

• 启动微调

• 微调完成后

• lora微调

• 评测

• C-Eval 评测

• MindSpore推理

• 基于高阶接口推理

• 单卡推理

• 多卡推理

• Batch 推理

• 通过 model.gener...

• 补充说明

• BF16 支持

# 通义千问

## 模型描述

通义千问是阿里云研发的通义千问大模型系列。基于Transformer的大语言模型, 在超大规模的预训练数据上进行训练得类型多样, 覆盖广泛, 包括大量网络文本、专业书籍、代码等。

```
@article{qwen,
  title={Qwen Technical Report},
  author={Jinze Bai and Shuai Bai and Yunfei Chu and Zeyu Cui and Kai Dang and Xiaodong Deng and Yang Fan and
  journal={arXiv preprint arXiv:2309.16609},
  year={2023}
}
```

## 仓库介绍

Qwen 基于 MindFormers 实现, 主要涉及的文件有:

1. 模型具体实现:

```
qwen
├── qwen_tokenizer.py      # tokenizer
├── qwen_model.py         # 模型实现
└── qwen_chat.py          # Chat接口
```

2. 模型配置:

```
qwen
├── predict_qwen_7b.yaml      # 7B 推理启动配置
├── finetune_qwen_7b.yaml    # 7B 全参微调启动配置(8K)
├── finetune_qwen_7b_bf16.yaml # 7B 全参微调启动配置(bf16, 2K)
├── finetune_qwen_7b_lora.yaml # 7B lora微调启动配置
├── predict_qwen_14b.yaml    # 14B 推理启动配置
├── finetune_qwen_14b.yaml   # 14B 全参微调启动配置(8K)
├── finetune_qwen_14b_bf16.yaml # 14B 全参微调启动配置(bf16, 2K)
└── finetune_qwen_14b_lora.yaml # 14B lora微调启动配置
```

3. 环境准备和任务启动脚本:

```
qwen
├── alpaca_converter.py      # alpaca数据集格式转换脚本
├── qwen_preprocess.py       # 数据集预处理脚本
├── convert_weight.py        # 权重转换脚本
├── run_qwen.py              # Qwen高阶接口脚本
└── run_qwen_chat.py        # Chat功能启动运行脚本
```

## 前期准备

### 环境要求

1. 环境搭建请参考 [MindSpore官网](#)，安装MindSpore2.3.0 + CANN社区版7.2.0配套版本。
2. 因Qwen的tokenizer基于 tiktoken 实现，而 tiktoken 官方不支持 Python 3.8 以下的版本，所以运行Qwen需要Python 3.8或者更高版本。

### 模型权重准备

本仓库提供已经转换完成的预训练权重、词表文件用于训练/微调/推理，用户可自行从下方链接拉取后直接使用。

- [Qwen-7B-Base](#)
- [Qwen-14B-Base](#)
- [qwen.tiktoken](#)

### 从huggingface版本权重文件转换

也可选择从huggingface下载预训练权重后根据以下步骤进行权重转换，需要下载整个工程。huggingface权重的下载链接如下：

- [Qwen-7B-Base](#)
- [Qwen-7B-Chat](#)
- [Qwen-14B-Base](#)
- [Qwen-14B-Chat](#)

首先，请安装官方Qwen模型所需的依赖软件包：

```
pip install torch==2.0.1 transformers==4.32.0 transformers_stream_generator einops accelerate tiktoken
pip uninstall tokenizers
pip install tokenizers==0.13.0
```

然后运行 [Mindformers 的权重转换工具](#)，将huggingface的权重转换为 Mindspore 的ckpt格式。

注意：权重转换完成之后，注意重新根据本项目[requirements.txt](#)恢复 tokenizers 包的版本：

```
pip install -r requirements.txt
```

### 模型权重切分与合并

从hugging face或官方github仓库转换而来的权重通常是单卡权重，基于该权重进行多卡微调，评测，推理，涉及ckpt从单机策略到分布式策略的切换。

通常训练采用分布式训练，基于该权重进行评测，推理多采用单卡，涉及ckpt从分布式策略到单机策略的切换。

以上涉及到ckpt的单卡，多卡转换，详细教程请参考特性文档[模型权重切分与合并](#)

### 全参微调

全参微调性能（seq\_length=8192，global\_batch\_size=8）：

Model	tokens/s
Mindformers-Qwen-7B	1512
Mindformers-Qwen-14B	901

### 数据集准备

目前提供alpaca数据集的预处理脚本用于全参微调任务。

数据集下载链接如下：

- [alpaca\\_data](#)

执行 alpaca\_converter.py，将原始数据集转换为指定格式。

```
python research/qwen/alpaca_converter.py \
--data_path path/alpaca_data.json \
--output_path /path/alpaca-data-conversation.json
# 参数说明
# data_path: 存放alpaca数据的路径
# output_path: 输出转换后对话格式的数据路径
```

```
{
  "id": "1",
  "conversations": [
    {
      "from": "user",
      "value": "Give three tips for staying healthy."
    },
    {
      "from": "assistant",
      "value": "1. Eat a balanced diet and make sure to include plenty of fruits and vegetables. \n2. Exercise regularly to keep your body healthy."
    }
  ]
},
```

执行 `qwen_preprocess.py` , 进行数据预处理和Mindrecord数据生成。

```
python research/qwen/qwen_preprocess.py \
--input_glob /path/alpaca-data-conversation.json \
--model_file /path/qwen.tiktoken \
--seq_length 8192 \
--output_file /path/alpaca-8192.mindrecord
```

### 启动微调

1. 当前支持模型已提供yaml文件，下文以Qwen-7B为例，即使用 `run_qwen_7b.yaml` 配置文件进行介绍，请根据实际使用模型更改配置文件。  
当前模型已支持使用Flash Attention算法进行全参微调，请参考 [Flash Attention使用文档](#)

2. 设置如下环境变量：

```
export MS_ASCEND_CHECK_OVERFLOW_MODE=INFNAN_MODE
```

3. 修改 `finetune_qwen_7b.yaml` 中相关配置，默认开启自动权重转换，使用完整权重。

```
load_checkpoint: '/path/model_dir' # 使用完整权重，权重按照`model_dir/rank_0/xxx.ckpt`格式存放
auto_trans_ckpt: True              # 打开自动权重转换
use_parallel: True
run_mode: 'finetune'

model:
  model_config:
    seq_length: 8192 # 与数据集长度保持相同

train_dataset: &train_dataset
data_loader:
  type: MindDataset
  dataset_dir: "/path/alpaca-8192.mindrecord" # 配置训练数据集文件夹路径

processor:
  tokenizer:
    vocab_file: "/path/qwen.tiktoken" # 配置tiktoken文件夹路径
```

4. 启动微调任务。

```
cd mindformers/research/qwen
bash ../../scripts/msrun_launcher.sh "python run_qwen.py \
--config finetune_qwen_7b.yaml \
--run_mode finetune \
--load_checkpoint /path/to/ckpt \
--use_parallel True \
--auto_trans_ckpt: True \
--train_dataset /path/alpaca-8192.mindrecord"

# 参数说明
# config: 配置文件路径
# load_checkpoint:: 权重文件路径，权重按照`model_dir/rank_0/xxx.ckpt`格式存放
# auto_trans_ckpt:: 自动权重转换开关
```

训练的log日志路径: `./output/log`

checkpoint(含优化器参数)存储路径: `./output/checkpoint`

checkpoint(不含优化器参数)存储路径: `./output/checkpoint_network`

若想合并ckpt用于后续评估, 选择不含优化器参数的权重即可。

微调完成后

- 合并权重文件:
- 多卡微调后, 如果想单卡运行推理或者评估, 需要合并权重文件:

```
python mindformers/tools/transform_ckpt.py \
--src_ckpt_strategy {path}/output/strategy/ \
--src_ckpt_dir {path}/output/checkpoint_network/ \
--dst_ckpt_dir {path}/target_checkpoint/ \
--prefix qwen_7b_base

# 参数说明
# src_ckpt_strategy: 切分权重时生成的分布式策略文件所在目录
# src_ckpt_dir: 多卡训练出的权重文件所在目录
# dst_ckpt_dir: 存放合并后权重文件的路径
# prefix: ckpt文件前缀名
```

关于权重文件的切分、合并, 可参考详细教程: [权重切分与合并](#)

- 运行推理:
- 由于微调时使用了`chatml`格式来准备训练数据, 所以在训练后的权重上进行推理时 (尤其是与训练数据相关的问题时), 也需要以`chatml`格式 (可使用 `run_qwen_chat.py` 加载此权重进行推理以验证微调效果)。

lora微调

lora微调性能 (seq\_length=2048, global\_batch\_size=8) :

Model	tokens/s
Mindformers-Qwen-7B	2694.7
Mindformers-Qwen-14B	1429.2

请参照[数据集准备](#)章节获取mindrecord格式的alpaca数据集, 参照[模型权重准备](#)章节获取权重。

- 当前支持模型已提供yaml文件, 下文以Qwen-7B为例, 即使用 `finetune_qwen_7b_lora.yaml` 配置文件进行介绍, 请根据实际使用模型
- 修改 `finetune_qwen_7b_lora.yaml` 中相关配置, 配置权重和数据集路径。

```
load_checkpoint: 'model_dir' # 使用完整权重, 权重按照`model_dir/rank_0/xxx.ckpt`格式存放

train_dataset: &train_dataset
  data_loader:
    type: MindDataset
    dataset_dir: "dataset_dir" # 配置训练数据集文件夹路径
    shuffle: True

model:
  model_config:
    seq_length: 2048 # 与数据集长度保持相同
  pet_config:
    pet_type: lora
    lora_rank: 64
    lora_alpha: 16
    lora_dropout: 0.05
    target_modules: '.*wq|.*wk|.*wv|.*wo|.*w1|.*w2|.*w3'
    freeze_exclude: ["*wte*", "*lm_head*"] # 使用chat权重进行微调时删除该配置
```

- 启动Lora微调任务。

```
bash ./../scripts/mindformers_train.py python run_qwen.py \
--config finetune_qwen_7b_lora.yaml \
--load_checkpoint: /path/to/ckpt_file \
--use_parallel True \
--run_mode finetune \
--auto_trans_ckpt: True \
--seq_length 2048 \
--train_dataset: /path/alpaca-2048.mindrecord"

# 参数说明
# config: 配置文件路径
# load_checkpoint:: 权重文件夹路径, 权重按照'model_dir/rank_0/xxx.ckpt'格式存放
# auto_trans_ckpt:: 自动权重转换开关
# run_mode: 运行模式, 微调时设置为finetune
# train_dataset: 训练数据集文件夹路径
```

评测

评测脚本下载地址[评测脚本](#)，下载后，脚本解压到mindformers/research/qwen/下，权重文件 qwen\_7b\_base.ckpt 放在脚本同级目录下。

C-Eval 评测

C-Eval是全面的中文基础模型评估套件，涵盖了52个不同学科的13948个多项选择题。

评测结果对比：

Model	C-Eval
Qwen-7B	62.6
Mindformers-Qwen-7B	63.3
Qwen-14B	72.1
Mindformers-Qwen-14B	72.13

运行此评测集的方法：

```
wget https://huggingface.co/datasets/ceval/ceval-exam/resolve/main/ceval-exam.zip
mkdir -p data/ceval && cd data/ceval; unzip ../../ceval-exam.zip && cd ../../
python evaluate_ceval.py -d data/ceval/
```

MindSpore推理

注意事项：

1. 当前支持模型已提供yaml文件，下文以Qwen-7B为例，即使用 predict\_qwen\_7b.yaml 配置文件进行介绍，请根据实际使用模型更改配置。
2. 运行下面的代码需要在 research/qwen 目录下，或者先将 research/qwen 目录所在路径加入到 PYTHONPATH 环境变量中。

基于高阶接口推理

单卡推理

1. 主要参数配置参考

```
load_checkpoint: '/path/qwen_7b_base.ckpt' # 填写权重路径
auto_trans_ckpt: False # 关闭自动权重转换

model:
  model_config:
    use_past: True # 使用增量推理加快推理速度
    is_dynamic: True # 开启动态shape(可选)

processor:
  tokenizer:
    vocab_file: /path/qwen.tiktoken # 配置词表路径
  use_parallel: False # 关闭并行模式
```

2. 启动推理

```
cd /path/mindformers/research/qwen/
export PYTHONPATH=/path/mindformers:$PYTHONPATH
python run_qwen.py \
--config predict_qwen_7b.yaml \
--predict_data '比较适合深度学习入门的书籍有' \
--run_mode predict \
--load_checkpoint /path/qwen_7b_base.ckpt \
--seq_length 2048 \
--device_id 0
# 比较适合深度学习入门的书籍有《Python深度学习》、《深度学习入门》、《动手学深度学习》等。这些书籍都比较容易理解，适合初学者。
```

多卡推理

1. 主要参数配置参考：

以单机2卡，模型并行的多卡推理为例，请参照[RANK\\_TABLE\\_FILE准备](#)获取单机2卡的 RANK\_TABLE\_FILE 文件。

```
load_checkpoint: '/path/model_dir'          # 使用完整权重，权重存放格式为"model_dir/rank_0/xxx.ckpt"
auto_trans_ckpt: True                      # 打开自动权重转换

model:
  model_config:
    use_past: True                         # 使用增量推理加快推理速度
    is_dynamic: True                       # 开启动态shape

processor:
  tokenizer:
    vocab_file: /path/qwen.tiktoken        # 配置词表路径

use_parallel: True                         # 使用并行模式

# parallel of device num = 2
parallel_config:
  data_parallel: 1
  model_parallel: 2
  pipeline_stage: 1
  micro_batch_num: 1
  vocab_emb_dp: True
  gradient_aggregation_group: 4
```

注：可配置 model\_config:param\_init\_type 为 float32 提高推理精度，但同时会影响在线推理性能。

2. 启动推理：

```
cd mindformers/research/qwen
WORKER_COUNT=2
# 推理命令中参数会覆盖yaml文件中的相同参数
bash ../../scripts/msrun_launcher.sh "python run_qwen.py \
--config predict_qwen_14b.yaml \
--run_mode predict \
--use_parallel True \
--load_checkpoint /path/model_dir \
--auto_trans_ckpt True \
--seq_length 2048 \
--predict_data '比较适合深度学习入门的书籍有' $WORKER_COUNT

# 比较适合深度学习入门的书籍有《Python深度学习》、《深度学习入门》、《动手学深度学习》等。这些书籍都比较容易理解，适合初学者。
```

Batch 推理

run\_qwen.py 允许通过 --batch\_size 指定并行发起推理的数量，通过 --predict\_data 传入多个具体的问题：

```
python run_qwen.py --config predict_qwen_7b.yaml \
--batch_size 2 \
--predict_data '帮助我制定一份去上海旅游攻略' '比较适合深度学习入门的书籍有'
```

### 通过 model.generate() 推理

```
import sys

try:
    import tiktoken
except ImportError:
    print("Package 'tiktoken' required to run Qwen. please install it with pip.", file=sys.stderr)
    sys.exit()

import mindspore as ms
from mindformers.tools.register.config import MindFormerConfig

from qwen_model import QwenForCausalLM
from qwen_tokenizer import QwenTokenizer
from qwen_config import QwenConfig

config = MindFormerConfig("/path/predict_qwen_7b.yaml")
config.use_past = True

model_config = QwenConfig.from_pretrained("/path/predict_qwen_7b.yaml")
model_config.checkpoint_name_or_path = '/path/qwen_7b_base.ckpt'
model_config.seq_length = 512

tokenizer = QwenTokenizer(**config.processor.tokenizer)

ms.set_context(mode=ms.GRAPH_MODE, device_target='Ascend', device_id=0)
ms.set_context(ascend_config={"precision_mode": "must_keep_origin_dtype"})

batch_size = 16
model_config.batch_size = batch_size
model = QwenForCausalLM(model_config)

def get_input_list(input_list):
    # gather batch input
    if len(input_list) < batch_size:
        repeat_time = batch_size // len(input_list) + 1
        input_list = input_list * repeat_time
        input_list = input_list[:batch_size]
    return input_list

def run_generate():
    input_list = ['帮助我制定一份去上海的旅游攻略',
                  '比较适合深度学习入门的书籍有']
    input_list = get_input_list(input_list)
    inputs = tokenizer(input_list, padding='max_length', max_length=model_config.seq_length, add_special_tokens=False)

    output = model.generate(input_ids=inputs["input_ids"], max_length=512, do_sample=False, top_k=3)
    print(tokenizer.decode(output, skip_special_tokens=True))

run_generate()
# '帮助我制定一份去上海的旅游攻略。Assistant:好的，去上海旅游的话，您可以先去外滩欣赏夜景，然后去城隍庙感受老上海的风情，还可以去豫园
# '比较适合深度学习入门的书籍有《Python深度学习》、《深度学习入门》、《动手学深度学习》等。这些书籍都比较容易理解，适合初学者。'
```

### 补充说明

#### BF16 支持

当前版本仅支持 bf16 数据类型的训练，暂不支持推理。

- convert\_weight.py 脚本默认的数据类型已经改为与原始权重一致（对于通义千问而言，即 bfloat16）；
- 推理时可将YAML配置中的 compute\_dtype 和 param\_init\_type 改为 float16；
- 如用训练集 bfloat16 数据训练，建议用 bfloat16 格式的数据，以便小数据量训练时保持精度和速度提升。



深圳市奥思网络科技有限公司版权所有

- Git 大全

Git 命令学习

CopyCat 代码克隆

检测

APP与插件下载
- Gitee Reward

Gitee 封面人物

GVP 项目

Gitee 博客

Gitee 公益计划

Gitee 持续集成
- OpenAPI

帮助文档

在线自助服务

更新日志
- 关于我们

加入我们

使用条款

意见建议

合作伙伴
- client@oschina.cn

企业版在线使用：400-606-0201

专业版私有部署：13670252304

13352947997



技术交流

