Retrieval-based-Voice-Conversion-WebUI / **infer-web.py**

12 people chore(sync): merge dev into main (#1379) e9dd11b · 2 years ago

1529 lines (1450…

```python
import os
import sys

now_dir = os.getcwd()
sys.path.append(now_dir)
from infer.modules.vc.modules import VC
from infer.modules.uvr5.modules import uvr
from infer.lib.train.process_ckpt import (
    change_info,
    extract_small_model,
    merge,
    show_info,
)
from i18n.i18n import I18nAuto
from configs.config import Config
from sklearn.cluster import MiniBatchKMeans
from dotenv import load_dotenv
import torch

try:
    import intel_extension_for_pytorch as ipex  # pylint: disable=import-error, unused-import

    if torch.xpu.is_available():
        from infer.modules.ipex import ipex_init

        ipex_init()
except Exception:  # pylint: disable=broad-exception-caught
    pass
import numpy as np
import gradio as gr
import faiss
import fairseq
import pathlib
import json
from time import sleep
from subprocess import Popen
```

```
    from random import shuffle
    import warnings
    import traceback
    import threading
    import shutil
    import logging


    logging.getLogger("numba").setLevel(logging.WARNING)

    logger = logging.getLogger(__name__)

    tmp = os.path.join(now_dir, "TEMP")
    shutil.rmtree(tmp, ignore_errors=True)
    shutil.rmtree("%s/runtime/Lib/site-packages/infer_pack" % (now_dir), ignore_errors=True)
    shutil.rmtree("%s/runtime/Lib/site-packages/uvr5_pack" % (now_dir), ignore_errors=True)
    os.makedirs(tmp, exist_ok=True)
    os.makedirs(os.path.join(now_dir, "logs"), exist_ok=True)
    os.makedirs(os.path.join(now_dir, "assets/weights"), exist_ok=True)
    os.environ["TEMP"] = tmp
    warnings.filterwarnings("ignore")
    torch.manual_seed(114514)


    load_dotenv()
    config = Config()
    vc = VC(config)


    if config.dml == True:

        def forward_dml(ctx, x, scale):
            ctx.scale = scale
            res = x.clone().detach()
            return res

        fairseq.modules.grad_multiply.GradMultiply.forward = forward_dml
    i18n = I18nAuto()
    logger.info(i18n)
```

```
                    interactive=True,
                )
                sr2.change(
                    change_sr2,
                    [sr2, if_f0_3, version19],
                    [pretrained_G14, pretrained_D15],
                )
                version19.change(
                    change_version19,
                    [sr2, if_f0_3, version19],
                    [pretrained_G14, pretrained_D15, sr2],
                )
                if_f0_3.change(
                    change_f0,
                    [if_f0_3, sr2, version19],
                    [f0method8, pretrained_G14, pretrained_D15],
                )
                gpus16 = gr.Textbox(
                    label=i18n("以-分隔输入使用的卡号, 例如   0-1-2   使用卡0和卡1和卡2"),
                    value=gpus,
                    interactive=True,
                )
                but3 = gr.Button(i18n("训练模型"), variant="primary")
                but4 = gr.Button(i18n("训练特征索引"), variant="primary")
                but5 = gr.Button(i18n("一键训练"), variant="primary")
                info3 = gr.Textbox(label=i18n("输出信息"), value="", max_lines=10)
```

```
                    info3 = gr.Textbox(label=i18n("输出信息"), value="", max_lines=10)
                    but3.click(
                        click_train,
                        [
                            exp_dir1,
                            sr2,
                            if_f0_3,
                            spk_id5,
                            save_epoch10,
                            total_epoch11,
                            batch_size12,
                            if_save_latest13,
                            pretrained_G14,
                            pretrained_D15,
                            gpus16,
                            if_cache_gpu17,
                            if_save_every_weights18,
                            version19,
                        ],
                        info3,
                        api_name="train_start",
                    )
                    but4.click(train_index, [exp_dir1, version19], info3)
                    but5.click(
                        train1key,
                        [
                            exp_dir1,
                            sr2,
                            if_f0_3,
                            trainset_dir4,
                            spk_id5,
                            np7,
                            f0method8,
                            save_epoch10,
                            total_epoch11,
                            batch_size12,
                            if_save_latest13,
                            pretrained_G14,
                            pretrained_D15,
                            gpus16,
                            if_cache_gpu17,
                            if_save_every_weights18,
                            version19,
                            gpus_rmvpe,
                        ],
                        info3,
                        api_name="train_start_all",
                    )

            with gr.TabItem(i18n("ckpt处理")):
                with gr.Group():
                    gr.Markdown(value=i18n("模型融合, 可用于测试音色融合"))
                    with gr.Row():
```

```
1357                    ckpt_a = gr.Textbox(label=i18n("A模型路径"), value="", interactive=True)
1358                    ckpt_b = gr.Textbox(label=i18n("B模型路径"), value="", interactive=True)
1359                    alpha_a = gr.Slider(
1360                        minimum=0,
1361                        maximum=1,
1362                        label=i18n("A模型权重"),
1363                        value=0.5,
1364                        interactive=True,
1365                    )
1366                with gr.Row():
1367                    sr_ = gr.Radio(
1368                        label=i18n("目标采样率"),
1369                        choices=["40k", "48k"],
1370                        value="40k",
1371                        interactive=True,
1372                    )
1373                    if_f0_ = gr.Radio(
1374                        label=i18n("模型是否带音高指导"),
1375                        choices=[i18n("是"), i18n("否")],
1376                        value=i18n("是"),
1377                        interactive=True,
1378                    )
1379                    info__ = gr.Textbox(
```

Retrieval-based-Voice-Conversion-WebUI / **infer-web.py** ↑ Top

Code | Blame | Raw

```
1384                        value="",
1385                        max_lines=1,
1386                        interactive=True,
1387                    )
1388                    version_2 = gr.Radio(
1389                        label=i18n("模型版本型号"),
1390                        choices=["v1", "v2"],
1391                        value="v1",
1392                        interactive=True,
1393                    )
1394                with gr.Row():
1395                    but6 = gr.Button(i18n("融合"), variant="primary")
1396                    info4 = gr.Textbox(label=i18n("输出信息"), value="", max_lines=8)
1397                but6.click(
1398                    merge,
1399                    [
1400                        ckpt_a,
1401                        ckpt_b,
1402                        alpha_a,
1403                        sr_,
1404                        if_f0_,
1405                        info__,
1406                        name_to_save0,
1407                        version_2,
1408                    ],
1409                    info4,
```

```
                    api_name="ckpt_merge",
                )  # def merge(path1,path2,alpha1,sr,f0,info):
            with gr.Group():
                gr.Markdown(value=i18n("修改模型信息(仅支持weights文件夹下提取的小模型文件)"))
                with gr.Row():
                    ckpt_path0 = gr.Textbox(
                        label=i18n("模型路径"), value="", interactive=True
                    )
                    info_ = gr.Textbox(
                        label=i18n("要改的模型信息"), value="", max_lines=8, interactive=True
                    )
                    name_to_save1 = gr.Textbox(
                        label=i18n("保存的文件名, 默认空为和源文件同名"),
                        value="",
                        max_lines=8,
                        interactive=True,
                    )
                with gr.Row():
                    but7 = gr.Button(i18n("修改"), variant="primary")
                    info5 = gr.Textbox(label=i18n("输出信息"), value="", max_lines=8)
                but7.click(
                    change_info,
                    [ckpt_path0, info_, name_to_save1],
                    info5,
                    api_name="ckpt_modify",
                )
            with gr.Group():
                gr.Markdown(value=i18n("查看模型信息(仅支持weights文件夹下提取的小模型文件)"))
                with gr.Row():
                    ckpt_path1 = gr.Textbox(
                        label=i18n("模型路径"), value="", interactive=True
                    )
                    but8 = gr.Button(i18n("查看"), variant="primary")
                    info6 = gr.Textbox(label=i18n("输出信息"), value="", max_lines=8)
                but8.click(show_info, [ckpt_path1], info6, api_name="ckpt_show")
            with gr.Group():
                gr.Markdown(
                    value=i18n(
                        "模型提取(输入logs文件夹下大文件模型路径),适用于训一半不想训了模型没有自动提取保存小文
                    )
                )
                with gr.Row():
                    ckpt_path2 = gr.Textbox(
                        label=i18n("模型路径"),
                        value="E:\\codes\\py39\\logs\\mi-test_f0_48k\\G_23333.pth",
                        interactive=True,
                    )
                    save_name = gr.Textbox(
                        label=i18n("保存名"), value="", interactive=True
                    )
                    sr__ = gr.Radio(
                        label=i18n("目标采样率"),
                        choices=["32k", "40k", "48k"],
```

```
                        value="40k",
                        interactive=True,
                    )
                    if_f0__ = gr.Radio(
                        label=i18n("模型是否带音高指导,1是0否"),
                        choices=["1", "0"],
                        value="1",
                        interactive=True,
                    )
                    version_1 = gr.Radio(
                        label=i18n("模型版本型号"),
                        choices=["v1", "v2"],
                        value="v2",
                        interactive=True,
                    )
                    info___ = gr.Textbox(
                        label=i18n("要置入的模型信息"), value="", max_lines=8, interactive=True
                    )
                    but9 = gr.Button(i18n("提取"), variant="primary")
                    info7 = gr.Textbox(label=i18n("输出信息"), value="", max_lines=8)
                    ckpt_path2.change(
                        change_info_, [ckpt_path2], [sr__, if_f0__, version_1]
                    )
                but9.click(
                    extract_small_model,
                    [ckpt_path2, save_name, sr__, if_f0__, info___, version_1],
                    info7,
                    api_name="ckpt_extract",
                )

            with gr.TabItem(i18n("Onnx导出")):
                with gr.Row():
                    ckpt_dir = gr.Textbox(label=i18n("RVC模型路径"), value="", interactive=True)
                with gr.Row():
                    onnx_dir = gr.Textbox(
                        label=i18n("Onnx输出路径"), value="", interactive=True
                    )
                with gr.Row():
                    infoOnnx = gr.Label(label="info")
                with gr.Row():
                    butOnnx = gr.Button(i18n("导出Onnx模型"), variant="primary")
                butOnnx.click(
                    export_onnx, [ckpt_dir, onnx_dir], infoOnnx, api_name="export_onnx"
                )

            tab_faq = i18n("常见问题解答")
            with gr.TabItem(tab_faq):
                try:
                    if tab_faq == "常见问题解答":
                        with open("docs/cn/faq.md", "r", encoding="utf8") as f:
                            info = f.read()
                    else:
                        with open("docs/en/faq_en.md", "r", encoding="utf8") as f:
```

```
1516                            info = f.read()
1517                    gr.Markdown(value=info)
1518                except:
1519                    gr.Markdown(traceback.format_exc())
1520
1521        if config.iscolab:
1522            app.queue(concurrency_count=511, max_size=1022).launch(share=True)
1523        else:
1524            app.queue(concurrency_count=511, max_size=1022).launch(
1525                server_name="0.0.0.0",
1526                inbrowser=not config.noautoopen,
1527                server_port=config.listen_port,
1528                quiet=True,
1529            )
```