

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

...

 **github-actions[bot]** chore(format): run black on main (#1851) 6061f63 · last year 

1619 lines (1539...

```
1  import os
2  import sys
3  from dotenv import load_dotenv
4
5  now_dir = os.getcwd()
6  sys.path.append(now_dir)
7  load_dotenv()
8  from infer.modules.vc.modules import VC
9  from infer.modules.uvr5.modules import uvr
10 from infer.lib.train.process_ckpt import (
11     change_info,
12     extract_small_model,
13     merge,
14     show_info,
15 )
16 from i18n.i18n import I18nAuto
17 from configs.config import Config
18 from sklearn.cluster import MiniBatchKMeans
19 import torch, platform
20 import numpy as np
21 import gradio as gr
22 import faiss
23 import fairseq
24 import pathlib
25 import json
26 from time import sleep
27 from subprocess import Popen
28 from random import shuffle
29 import warnings
30 import traceback
31 import threading
32 import shutil
33 import logging
34
35
36 logging.getLogger("numba").setLevel(logging.WARNING)
37 logging.getLogger("librosa").setLevel(logging.WARNING)
```

```
37 logging.getLogger("httpx").setLevel(logging.WARNING)
38
39 logger = logging.getLogger(__name__)
40
41 tmp = os.path.join(now_dir, "TEMP")
42 shutil.rmtree(tmp, ignore_errors=True)
43 shutil.rmtree("%s/runtime/Lib/site-packages/infer_pack" % (now_dir), ignore_errors=True)
44 shutil.rmtree("%s/runtime/Lib/site-packages/uvr5_pack" % (now_dir), ignore_errors=True)
45 os.makedirs(tmp, exist_ok=True)
46 os.makedirs(os.path.join(now_dir, "logs"), exist_ok=True)
47 os.makedirs(os.path.join(now_dir, "assets/weights"), exist_ok=True)
48 os.environ["TEMP"] = tmp
49 warnings.filterwarnings("ignore")
50 torch.manual_seed(114514)
51
52
53 config = Config()
54 vc = VC(config)
55
56
57 if config.dml == True:
58
59     def forward_dml(ctx, x, scale):
60         ctx.scale = scale
61         res = x.clone().detach()
62         return res
63
64     fairseq.modules.grad_multiply.GradMultiply.forward = forward_dml
65 i18n = I18nAuto()
66 logger.info(i18n)
67 # 判断是否有能用来训练和加速推理的N卡
68 ngpu = torch.cuda.device_count()
69 gpu_infos = []
70 mem = []
71 if_gpu_ok = False
72
73 if torch.cuda.is_available() or ngpu != 0:
74     for i in range(ngpu):
75         gpu_name = torch.cuda.get_device_name(i)
```



```

491     f0_dir = "%s/2a_f0" % (exp_dir)
492     f0nsf_dir = "%s/2b-f0nsf" % (exp_dir)
493     names = (
494         set([name.split(".")[0] for name in os.listdir(gt_wavs_dir)])
495         & set([name.split(".")[0] for name in os.listdir(feature_dir)])
496         & set([name.split(".")[0] for name in os.listdir(f0_dir)])
497         & set([name.split(".")[0] for name in os.listdir(f0nsf_dir)])
498     )
499     else:
500         names = set([name.split(".")[0] for name in os.listdir(gt_wavs_dir)]) & set(
501             [name.split(".")[0] for name in os.listdir(feature_dir)]
502         )
503     opt = []
504     for name in names:
505         if if_f0_3:
506             opt.append(
507                 "%s/%s.wav|%s/%s.npy|%s/%s.wav.npy|%s/%s.wav.npy|%s"
508                 % (
509                     gt_wavs_dir.replace("\\", "\\\\"),
510                     name,
511                     feature_dir.replace("\\", "\\\\"),
512                     name

```

```

512         name,
513         f0_dir.replace("\\", "\\\\"),
514         name,
515         f0nsf_dir.replace("\\", "\\\\"),
516         name,
517         spk_id5,
518     )
519 )
520 else:
521     opt.append(
522         "%s/%s.wav|%s/%s.npy|%s"
523         % (
524             gt_wavs_dir.replace("\\", "\\\\"),
525             name,
526             feature_dir.replace("\\", "\\\\"),
527             name,
528             spk_id5,
529         )
530     )
531 fea_dim = 256 if version19 == "v1" else 768
532 if if_f0_3:
533     for _ in range(2):
534         opt.append(
535             "%s/logs/mute/0_gt_wavs/mute%s.wav|%s/logs/mute/3_feature%s/mute.npy|%s/logs/mut
536             % (now_dir, sr2, now_dir, fea_dim, now_dir, now_dir, spk_id5)
537         )
538 else:
539     for _ in range(2):
540         opt.append(
541             "%s/logs/mute/0_gt_wavs/mute%s.wav|%s/logs/mute/3_feature%s/mute.npy|%s"
542             % (now_dir, sr2, now_dir, fea_dim, spk_id5)
543         )
544 shuffle(opt)
545 with open("%s/filelist.txt" % exp_dir, "w") as f:
546     f.write("\n".join(opt))
547 logger.debug("Write filelist done")
548 # 生成config#无需生成config
549 # cmd = python_cmd + " train_nsf_sim_cache_sid_load_pretrain.py -e mi-test -sr 40k -f0 1 -bs
550 logger.info("Use gpus: %s", str(gpus16))
551 if pretrained_G14 == "":
552     logger.info("No pretrained Generator")
553 if pretrained_D15 == "":
554     logger.info("No pretrained Discriminator")
555 if version19 == "v1" or sr2 == "40k":
556     config_path = "v1/%s.json" % sr2
557 else:
558     config_path = "v2/%s.json" % sr2
559 config_save_path = os.path.join(exp_dir, "config.json")
560 if not pathlib.Path(config_save_path).exists():
561     with open(config_save_path, "w", encoding="utf-8") as f:
562         json.dump(
563             config.json_config[config_path],
564             f,

```

```

565         ensure_ascii=False,
566         indent=4,
567         sort_keys=True,
568     )
569     f.write("\n")
570     if gpus16:
571         cmd = (
572             "%s" infer/modules/train/train.py -e "%s" -sr %s -f0 %s -bs %s -g %s -te %s -se %s
573             % (
574                 config.python_cmd,
575                 exp_dir1,
576                 sr2,
577                 1 if if_f0_3 else 0,
578                 batch_size12,
579                 gpus16,
580                 total_epoch11,
581                 save_epoch10,
582                 "-pg %s" % pretrained_G14 if pretrained_G14 != "" else "",
583                 "-pd %s" % pretrained_D15 if pretrained_D15 != "" else "",
584                 1 if if_save_latest13 == i18n("是") else 0,
585                 1 if if_cache_gpu17 == i18n("是") else 0,
586                 1 if if_save_every_weights18 == i18n("是") else 0,
587                 version19,
588             )
589         )
590     else:
591         cmd = (

```

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

↑ Top

Code

Blame

Raw



```

596         sr2,
597         1 if if_f0_3 else 0,
598         batch_size12,
599         total_epoch11,
600         save_epoch10,
601         "-pg %s" % pretrained_G14 if pretrained_G14 != "" else "",
602         "-pd %s" % pretrained_D15 if pretrained_D15 != "" else "",
603         1 if if_save_latest13 == i18n("是") else 0,
604         1 if if_cache_gpu17 == i18n("是") else 0,
605         1 if if_save_every_weights18 == i18n("是") else 0,
606         version19,
607     )
608 )
609 logger.info("Execute: " + cmd)
610 p = Popen(cmd, shell=True, cwd=now_dir)
611 p.wait()
612 return "训练结束， 您可查看控制台训练日志或实验文件夹下的train.log"
613
614
615 # but4.click(train_index, [exp_dir1], info3)
616 def train_index(exp_dir1, version19):
617     # exp_dir = "%s/logs/%s" % (now_dir, exp_dir1)

```

...

```

618 exp_dir = "logs/%s" % (exp_dir1)
619 os.makedirs(exp_dir, exist_ok=True)
620 feature_dir = (
621     "%s/3_feature256" % (exp_dir)
622     if version19 == "v1"
623     else "%s/3_feature768" % (exp_dir)
624 )
625 if not os.path.exists(feature_dir):
626     return "请先进行特征提取!"
627 listdir_res = list(os.listdir(feature_dir))
628 if len(listdir_res) == 0:
629     return "请先进行特征提取!"
630 infos = []
631 npys = []
632 for name in sorted(listdir_res):
633     phone = np.load("%s/%s" % (feature_dir, name))
634     npys.append(phone)
635 big_npy = np.concatenate(npys, 0)
636 big_npy_idx = np.arange(big_npy.shape[0])
637 np.random.shuffle(big_npy_idx)
638 big_npy = big_npy[big_npy_idx]
639 if big_npy.shape[0] > 2e5:
640     infos.append("Trying doing kmeans %s shape to 10k centers." % big_npy.shape[0])
641     yield "\n".join(infos)
642     return
643     big_npy = (
644         MiniBatchKMeans(
645             n_clusters=10000,
646             verbose=True,
647             batch_size=256 * config.n_cpu,
648             compute_labels=False,
649             init="random",
650         )
651         .fit(big_npy)
652         .cluster_centers_
653     )
654 except:
655     info = traceback.format_exc()
656     logger.info(info)
657     infos.append(info)
658     yield "\n".join(infos)
659
660 np.save("%s/total_fea.npy" % exp_dir, big_npy)
661 n_ivf = min(int(16 * np.sqrt(big_npy.shape[0])), big_npy.shape[0] // 39)
662 infos.append("%s,%s" % (big_npy.shape, n_ivf))
663 yield "\n".join(infos)
664 index = faiss.index_factory(256 if version19 == "v1" else 768, "IVF%s,Flat" % n_ivf)
665 # index = faiss.index_factory(256 if version19 == "v1" else 768, "IVF%s,PQ128x4fs,RFlat"%n_ivf)
666 infos.append("training")
667 yield "\n".join(infos)
668 index_ivf = faiss.extract_index_ivf(index) #
669 index_ivf.nprobe = 1
670 index.train(big_npy)

```

```

671     faiss.write_index(
672         index,
673         "%s/trained_IVF%s_Flat_nprobe_%s_%s_%s.index"
674         % (exp_dir, n_ivf, index_ivf.nprobe, exp_dir1, version19),
675     )
676     infos.append("adding")
677     yield "\n".join(infos)
678     batch_size_add = 8192
679     for i in range(0, big_npy.shape[0], batch_size_add):
680         index.add(big_npy[i : i + batch_size_add])
681     faiss.write_index(
682         index,
683         "%s/added_IVF%s_Flat_nprobe_%s_%s_%s.index"
684         % (exp_dir, n_ivf, index_ivf.nprobe, exp_dir1, version19),
685     )
686     infos.append(
687         "成功构建索引 added_IVF%s_Flat_nprobe_%s_%s_%s.index"
688         % (n_ivf, index_ivf.nprobe, exp_dir1, version19)
689     )
690     try:
691         link = os.link if platform.system() == "Windows" else os.symlink
692         link(
693             "%s/added_IVF%s_Flat_nprobe_%s_%s_%s.index"
694             % (exp_dir, n_ivf, index_ivf.nprobe, exp_dir1, version19),
695             "%s/%s_IVF%s_Flat_nprobe_%s_%s_%s.index"
696             % (
697                 outside_index_root,
698                 exp_dir1,
699                 n_ivf,
700                 index_ivf.nprobe,
701                 exp_dir1,
702                 version19,
703             ),
704         )
705         infos.append("链接索引到外部-%s" % (outside_index_root))
706     except:
707         infos.append("链接索引到外部-%s失败" % (outside_index_root))
708
709     # faiss.write_index(index, '%s/added_IVF%s_Flat_FastScan_%s.index'%(exp_dir,n_ivf,version19))
710     # infos.append("成功构建索引, added_IVF%s_Flat_FastScan_%s.index"%(n_ivf,version19))
711     yield "\n".join(infos)
712
713
714     # but5.click(train1key, [exp_dir1, sr2, if_f0_3, trainset_dir4, spk_id5, gpus6, np7, f0method8,
715     ✓ def train1key(
716         exp_dir1,
717         sr2,
718         if_f0_3,
719         trainset_dir4,
720         spk_id5,
721         np7,
722         f0method8,
723         save_epoch10,

```

```
724         total_epoch11,  
725         batch_size12,  
726         if_save_latest13,  
727         pretrained_G14,  
728         pretrained_D15,  
729         gpus16,  
730         if_cache_gpu17,  
731         if_save_every_weights18,  
732         version19,  
733         gpus_rmvpe,  
734     ):  
735         infos = []  
736  
737         def get_info_str(strr):  
738             infos.append(strr)  
739             return "\n".join(infos)  
740  
741         # step1:处理数据  
742         yield get_info_str(i18n("step1:正在处理数据"))  
743         [get_info_str(_) for _ in preprocess_dataset(trainset_dir4, exp_dir1, sr2, np7)]
```



```
1546         label=i18n("目标采样率"),
1547         choices=["32k", "40k", "48k"],
1548         value="40k",
1549         interactive=True,
1550     )
1551     if_f0__ = gr.Radio(
1552         label=i18n("模型是否带音高指导,1是0否"),
1553         choices=["1", "0"],
1554         value="1",
1555         interactive=True,
1556     )
1557     version_1 = gr.Radio(
1558         label=i18n("模型版本型号"),
1559         choices=["v1", "v2"],
1560         value="v2",
1561         interactive=True,
1562     )
1563     info__ = gr.Textbox(
1564         label=i18n("要置入的模型信息"),
1565         value="",
1566         max_lines=8,
1567         interactive=True,
1568     )
```

```

1568         but9 = gr.Button(i18n("提取"), variant="primary")
1569         info7 = gr.Textbox(label=i18n("输出信息"), value="", max_lines=8)
1570         ckpt_path2.change(
1571             change_info_, [ckpt_path2], [sr__, if_f0__, version_1]
1572         )
1573     but9.click(
1574         extract_small_model,
1575         [ckpt_path2, save_name, sr__, if_f0__, info____, version_1],
1576         info7,
1577         api_name="ckpt_extract",
1578     )
1579
1580 with gr.TabItem(i18n("Onnx导出")):
1581     with gr.Row():
1582         ckpt_dir = gr.Textbox(
1583             label=i18n("RVC模型路径"), value="", interactive=True
1584         )
1585     with gr.Row():
1586         onnx_dir = gr.Textbox(
1587             label=i18n("Onnx输出路径"), value="", interactive=True
1588         )
1589     with gr.Row():
1590         infoOnnx = gr.Label(label="info")
1591     with gr.Row():
1592         butOnnx = gr.Button(i18n("导出Onnx模型"), variant="primary")
1593         butOnnx.click(
1594             export_onnx, [ckpt_dir, onnx_dir], infoOnnx, api_name="export_onnx"
1595         )
1596
1597 tab_faq = i18n("常见问题解答")
1598 with gr.TabItem(tab_faq):
1599     try:
1600         if tab_faq == "常见问题解答":
1601             with open("docs/cn/faq.md", "r", encoding="utf8") as f:
1602                 info = f.read()
1603         else:
1604             with open("docs/en/faq_en.md", "r", encoding="utf8") as f:
1605                 info = f.read()
1606         gr.Markdown(value=info)
1607     except:
1608         gr.Markdown(traceback.format_exc())
1609
1610 if config.iscolab:
1611     app.queue(concurrency_count=511, max_size=1022).launch(share=True)
1612 else:
1613     app.queue(concurrency_count=511, max_size=1022).launch(
1614         server_name="0.0.0.0",
1615         inbrowser=not config.noautoopen,
1616         server_port=config.listen_port,
1617         quiet=True,
1618     )
1619

```

